NTNU

Norwegian University of
Science and Technology

UAV/UAS path planning for ice management information gathering

Artsiom Stalmakou

Master of Science in Engineering Cybernetics
Submission date: June 2011
Supervisor: Lars Imsland, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem description

Performing Dynamic Positioning (DP) operations in ice-infested regions require reliable information about the incoming ice conditions. Severe ice features such as ice ridges and icebergs must be detected, tracked and forecasted in due time to ensure operational success and safety. Furthermore, quantization of important ice properties will aid the accuracy of ice-load models and thus the performance of the DP system. To obtain this information, it is envisioned that Unmanned Aerial Vehicles (UAV) will support the information gathering. Several aspects regarding the development of such Unmanned Aircraft Systems (UAS) need to be addressed. Among these challenges, a purposeful path planning strategy is a key objective.

The path planning should be formulated as an optimization problem and take into account the constraints imposed by the dynamics of the aircraft and possibly regulations and airspace restrictions. Furthermore, the representation of the surveillance area (SA) is an important issue. Consequently, method / approach for representation of the SA should be proposed and constraints imposed by the SA is expected to be included in the path planning optimization problem. Also, the ice flow within the SA should be taken into account. Additionally, it can be assumed that the UAV operates at a constant altitude so that only a planar motion of the UAV is considered.

Work description:

- Describe what ice management is, and the possible role of UAS in the ice management.
- Discuss and propose UAS operational strategies, based on requirements and restrictions.
- Based on the above, formulate appropriate path planning optimization problems for UAVs in ice management.
- Test / illustrate the strategies / planning in appropriate ways (e.g simulations).

Supervisor: Lars Imsland, NTNU
Co-advisor(s): Joakim Haugen og Esten Grøtli, NTNU

Abstract

The key objective of this work is the proposition of the path planning strategy for unmanned aerial vehicle (UAV) intended for information gathering in Arctic environments / ice-infested regions. Two different path planning strategies are considered; one for analysis of the surveillance area defined as a grid and the other for analysis of the surveillance area defined as a sector. The mixed integer linear programming (MILP), YALMIP modeling language and GUROBI optimizer are used for formulation and solution of the path planning optimization problems. Furthermore, both path planning strategies are tested for the cases of constant and variable ice flow, respectively; the following are investigated in each simulation case: flight path of the UAV, coverage of the surveillance area, speed and acceleration of the UAV and the solver-time. Moreover, throughout this work only a planar motion is considered, with one single UAV used in each simulation case.

ACKNOWLEDGMENTS

I would like to tank my supervisor Lars Imsland for his important support throughout this work. Also, I would like to thank my co-advisors Esten Grøtli and Joakim Haugen for their advices and practical help related to this work.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter gives an introduction to this thesis.

1.1 Background

It is a well known fact that offshore drilling operations are challenging, where precise positioning of the drilling vessel is essential for operational success. Several strategies have been developed for this purpose. One of these is referred to as dynamic positioning (DP), where the horizontal movement of the drilling vessel is controlled in surge, sway and yaw (Fossen, 2010). However, it takes more than precise positioning of the drilling vessel when it comes to drilling operations in ice-covered waters. It is desirable to avoid collisions with potential ice threats in order to ensure operational success and safety. Hence the information about the ice conditions is required; it is envisioned that the unmanned aerial vehicles (UAVs) can be used to support the process of information gathering. Consequently, appropriate path planning strategy for UAVs is of great importance.

1.2 Ice management

The Arctic has become an interesting field for oil- and gas production for many countries. According to (Eik, 2008), 25 % of world's hydrocarbon resources are located in Arctic. It is however difficult to carry out drilling operations in Arctic due to ice-covered waters (Stalmakou, 2010). Ice in the form of ice floes / ice ridges / icebergs leads to several problems related to drilling operations. For example, ice forced under the hull can make great damage to the vessel (Jenssen et al., 2009). The ice management is a key factor for the operational success; however, there is no unambiguous definition of the ice management (Eik, 2008). Following definition is proposed by (Eik, 2008):

> *Ice management is the sum of all acitivities where the objective is to reduce or avoid actions from any kind of ice features. This will include, but is not limited to:*
>
> - *Detection, tracking, and forecasting of sea ice, ice ridges and icebergs.*
> - *Threat evaluation.*
> - *Physical ice management such as ice breaking and iceberg towing.*
> - *Procedures for disconnection of offshore structures applied in search for or production of hydrocarbons.*

Generally speaking, the ice management is a combination of a wide range of activities with primary objective to ensure safety and efficiency of the offshore operations in ice-covered waters.

1.3 UAV/UAS in ice management

1.3.1 UAV vs. UAS

Let us start with a brief explanation of what unmanned aircraft system (UAS) is, and look at the difference between a unmanned aerial vehicle (UAV) and UAS. According to (Dalamagkidis et al., 2008), for several years it has been common to use the term UAV, which is an areal vehicle without a pilot on board. However, now it has become more and more common to use the term UAS (Dalamagkidis et al., 2008). The major difference between a UAV and UAS is that a UAS is a *system*, which includes the unmanned aircraft / UAV, the launch and retrieval system, the ground control station (GCS) and the communication channel between the unmanned aircraft and GCS (Dalamagkidis et al., 2008).

1.3.2 Role of UAS in ice management

It is a well known fact that the Arctic is one of the most challenging regions in the world when it comes to offshore operations. The combination of such factors as wind, waves, darkness and low visibility, low temperature, presence of the ice and a number of other factors related to the equipment itself, leads to that both efficiency and safety related to these offshore operations are significantly reduced (Stalmakou, 2010). Therefore, the information gathering in the area of operation is of great importance, and this is where the application of the UAS may be appropriate. Use of UAS in ice management is attractive due to its characteristics. For example, the fact of that the applied aircraft is unmanned, opens for new opportunities related to operations in hazardous environments. As already mentioned, the Arctic is characterized as a challenging area of operation. Use of UAV in such environments provides the opportunity to push the limits when it comes to information gathering without putting human lives at risk. The physical dimensions of this type of aircraft are variable due to wide range of applications (Stalmakou, 2010). Appropriate choice of such aircraft makes it possible to perform operations that would be impossible to perform for ordinary manned aircraft. For example, direct landing on any ice type. Furthermore, it is a great flexibility in the choice of instrumentation. It is up to the user of the system to determine what information is relevant for a given operation. When this choice is made it is relatively easy to select the appropriate instruments and attach these to the UAV. There is clearly a limitation related to how much and how heavy equipment a UAV can carry. However, by using a smart integration of the instruments one can reduce the load of the UAV and gain the ability to perform many different measurements (Stalmakou, 2010). Consequently, with the right instrumentation the UAV can provide critical information about the area of operation needed for successful operational outcome. Moreover, it is possible to implement several levels of control, which in turn provides great flexibility related to control of the UAV. According to (Hobbs, 2010), there are three levels of control: manual, supervisory and autonomous. Each level sets clear limits for the operator's actions under the control of the UAV. For example, manual control is direct control of the UAV performed by the operator via radio control box, while under autonomous control all controlling actions are completely autonomous (Hobbs, 2010). Hence the control of unmanned aircraft can be performed with variable degree of human intervention, which subsequently opens for operations in more complex environments and execution of more complex tasks (Mettler et al., 2003). Based on what has already been said, there is no doubt that a UAS has a great potential within ice management. By selecting appropriate instrumentation in combination with appropriate control strategy, a UAS can be adapted to operate in the most demanding environments, without putting human lives at risk. It is likely to expect that application of UASs in ice management can improve already existing ice management strategies by making them more efficient and reliable.

1.4 Objective

The main goal of this work is to propose and test path planning strategies for UAV intended for operation in Arctic environments / ice-infested regions. A more detailed work description can be divided into the following steps:

- A brief introduction to the existing national and international regulations, restrictions and requirements related to the operation of UASs.
- A formulation of path planning optimization problems based on the proposed path planning strategies for the UAV.
- Implementation and simulation of the formulated path planning optimization problems.

1.5 Structure of the thesis

Chapter 2 gives a brief introduction to existing national and international regulations, restrictions and requirements related to operation of UASs. Furthermore, Chapter 3 proposes and discusses the operational strategies for UAV where two cases are considered; grid shaped and sector shaped surveillance area. Formulation of the optimization problems for proposed strategies are presented in Chapter 4, where MILP is used for this purpose. The simulation results are presented in Chapter 5. Chapter 6 contains conclusion and recommendations for further work.

Chapter 2

UAS regulations

This chapter discusses requirements and restrictions for UASs defined by the Civil Aviation Authority of Norway and international regulations.

2.1 National and international UAS regulations

2.1.1 Civil Aviation Authority of Norway

National regulations for use of unmanned aircraft systems (UASs) are of great importance for the safety of the operation of such systems. There are however no regulations currently governing the application of UASs within Norwegian airspace (Civil Aviation Authority - Norway). The main reason for this is that the use of UASs involves a number of issues that have not previously existed, and a lot of pioneering work is still done (Civil Aviation Authority - Norway). Following paragraph is take from the (Civil Aviation Authority - Norway):

> *There are currently no national regulations governing the use of unmanned aircraft systems for either commercial / utility operations or air sports / recreation. Civil Aviation Authority (Norway) has initiated the work to develop a regulatory framework for this activity, and has also strengthened its staff who will work with this.*

However, few guidelines for the development of such regulations are already defined. Regulations for UASs with aircraft that have take-off mass exceeding 150 kg will be prepared by the European Aviation Safety

Agency (EASA), and this will be the basis for the development of the national regulations for UASs with a maximum take-off mass up to 150 kg (Civil Aviation Authority - Norway). Furthermore, Civil Aviation Authority of Norway will set requirements for reporting system for incidents and accidents with UASs, as for manned aviation; also the requirements for UAS equipment, operations and personnel qualifications shall be set (Civil Aviation Authority - Norway). Operations of unmanned aircraft in the Norwegian airspace shall also be restricted to segregated airspace, unless the unmanned aircraft can be observed without aids such as binoculars and camera (Civil Aviation Authority - Norway).

To summarize, Norway does not have any regulations for application of UASs in Norwegian airspace. However, work related to preparation of these regulations have already been initiated, where regulations to be prepared by the EASA will be used as basis for formulation of national regulations.

2.1.2 International airspace - UAS in Arctic airspace

Let us now take a look at the rules for the application of UASs within the international airspace. Since UASs operations in Arctic are considered, it is obvious to look at regulations that applies within the Arctic airspace.

According to (Marshall, 2009), the operation of the UASs within international airspace depends on:

- Understanding of what an aircraft is from a regulatory perspective.
- International Civil Aviation Organization (ICAO).
- Understanding of which regulations, rules or laws that control the operation of a particular type of aircraft.

Considering the literal definition provided by ICAO there is no distinction between manned and unmanned aircraft; no specifications are provided that can serve as a basis for creating a distinction between manned and unmanned aircraft (Marshall, 2009). However, the Contracting States (or simply a union of cooperating States) can develop their own definitions and categories of aircraft, which in turn have influence on which laws and regulations do apply despite the operations in international airspace (Marshall, 2009). When it comes to so-called *Rules of the Road* within the international airspace, the ICAO has defined three categories of such rules which do apply without exception in international airspace (Marshall, 2009):

- General rules.
- Visual flight rules.
- Instrumentation flight rules.

To summarize, operations of UASs in international airspace are regulated based on both ICAO regulations and Contracting States own regulations (Marshall, 2009). Any operation of such systems in international airspace must comply with what is stated in these regulations. As long as this requirement is fulfilled it is natural to assume that operation of UASs will be allowed (Marshall, 2009).

2.1.3 Summary

In order to be able to operate UASs in international airspace, and therefor also the Arctic airspace, it is required that the regulations of the Contracting State (which is Norway in this case) and the regulations of the ICAO are followed. However, the current situation is that Norway lacks national regulations governing the use of systems with unmanned aircraft. The work directed towards the development of these regulations has already been initiated, with EASA regulations as basis. As a consequence, it is natural to assume that regulations prepared by EASA and ICAO is a good starting point for definition of requirements and restrictions related to use of UASs in international airspace. For clarification purpose one can say that UASs does not fly in law-empty room, but current regulations may not always fit for desired applications of such systems (Johansen, 2008).

2.2 Restrictions and requirements for UAS

This section discusses the requirements and restrictions related to the application of UASs in Arctic airspace. As already mentioned, Norway does not provide national regulations governing the use of UASs. Consequently, the ICAO and EASA regulations will be used instead, combining available guiding documentation provided by the Civil Aviation Authority of Norway. It is further natural to assume that some of the requirements and restrictions to be presented in this section will differ from those requirements and restrictions which is to be defined by the national regulations in the future. Therefore, the material to be presented in this section, up to a certain extent, is assumed to be of supervising character.

2.2.1 Guidance on special conditions related to airworthiness certification of UAS

According to (Haddon et al., 2009), it is required to have special conditions in order to address the characteristics of UAS. Some of these conditions will be presented in this section.

Emergency recovery capability

Emergency recovery capability is an important property of UASs. As specified in (Haddon et al., 2009), such a capability will normally consist of either a flight termination system or emergency recovery procedures. Following definitions are taken from (Haddon et al., 2009):

- A flight termination system - *Immediately end the flight and reduce the kinetic energy at impact, but does not necessary ensure the crash / impact point location.*
- Emergency recovery procedures - *Functions that could be implemented through UAS flight crew command or through an automatic pre-programmed course of action, that are intended to navigate the unmanned aircraft to a pre-selected emergency site and then to make a safe landing or terminate the flight.*

Command and control link

Based on (Haddon et al., 2009), there are several requirements related to command and control link. The one which has its primary influence on path planning for the UAS is:

- *The UAS flight crew should be provided with a continuos indication of the command and control link signal strength together with the maximum link range.*

A full list of airworthiness factors related to command and control link can be found in (Haddon et al., 2009).

System safety assessment

According to (Haddon et al., 2009), it is required to perform UAS safety assessment i order to demonstrate that the UAS meets safety objectives. This safety assessment is related to for example (Haddon et al., 2009):

- Risk of an uncontrolled crash.
- UAS design viewed as a whole.
- Emergency recovery capability.

The total level of risk associated with use of UAS should not be worse than the equivalent use of manned aircraft (Civil Aviation Authority - Norway).

Level of autonomy

The level of UAS autonomy varies; at the lowest level of autonomy the operator performs direct control of the unmanned aircraft, while at the highest level of autonomy it is not necessary to have permanent control link and the operator intervene the unmanned aircraft flight in exceptional cases (JAA/EUROCONTROL, 2004). Therefore there is a need to investigate how these levels of autonomy affect the UAS airworthiness criteria (JAA/EUROCONTROL, 2004).

Human machine interface

Human machine interface is an important part of the UAS. It is important to show that any operation of such systems is as safe as ordinary operations of manned aircraft (JAA/EUROCONTROL, 2004). The following paragraph is taken from the (JAA/EUROCONTROL, 2004):

> *The UAV operations concept implicitly requires that a specific Human Machine interface design can adequately and safely control the vehicle, be cognizant of the environment and air traffic around the vehicle, and respond to emergency conditions and situations in order for continued safe flight and landing of the vehicle.*

Control station

The superior requirement of safety results in certain requirements for the UAS control station. The following paragraph is taken from (Haddon et al., 2009):

> *The design of any element of the control station that could, due to failure, prejudice the safe control of the aircraft, must be approved as part of the type-certification.*

2.2.2 Airspace restrictions

Operation of UASs in international airspace is an important issue. There is still much work that remains related to definition of recommended practice and standards for such operations. Nevertheless, few guidelines are available. According to (Marshall, 2009), no ICAO-imposed barriers prohibit UAS operations within international airspace as long as following requirements are met:

- The unmanned aircraft is operated at altitudes near the surface of the ocean or the ice caps. It is assumed that «near the surface» is referred to an operational altitude that does not require a permission. According to (Civil Aviation Authority - Norway), this altitude is limited to 400 ft.
- The operation of the unmanned aircraft do not interfere with traditional commercial operations.

Furthermore, the Contracting States which provide services in a specific part of the international airspace have the right to define additional rules for the same area (Marshall, 2009). However, currently there is no regional agreements that specifically address flights of unmanned aircraft in their control or information areas (Marshall, 2009). Additionally, based on Article 8 in (ICAO, 2006), any aircraft without a pilot is required to have special authorization to fly over the territory of a Contracting State, which in turn is given by the Contracting State. When it comes to registration marks, the Article 20 in (ICAO, 2006) is applied:

> *Every aircraft engaged in international air navigation shall bear its appropriate nationality and registration marks.*

Finally, any aircraft is required to have certificate of airworthiness (ICAO, 2006). Following paragraph is taken from Article 31 in (ICAO, 2006):

> *Every aircraft engaged in international airspace shall be provided with a certificate of airworthiness issued or rendered valid by the State in which it is registered.*

2.2.3 Safety margins

When it comes to the safety margins, according to (Civil Aviation Authority - Norway), any operations of unmanned aircraft can only take place in segregated airspace (unless the unmanned aircraft can be observed without any aids). Additionally, the manual control of the unmanned aircraft must be possible at any given time to make sure that collisions with other aircraft, persons, vessels, vehicles and structures on the ground can be prevented (Civil Aviation Authority - Norway). In practice it means that there are currently no precisely defined safety margins; it is the superior requirement of safety that apply, with primary objective to avoid collisions at all cost. When it comes to ICAO safety margins, the situation is even more complicated. As already mentioned, the ICAO makes no distinction between manned and unmanned aircraft. Consequently, any rule applicable to manned aircraft is also applicable to unmanned aircraft. So if one intend to comply with safety margins, one have to follow the *Rules of the air*.

2.2.4 Operator certification

Training of the UAS operators has become an important requirement. It turns out that a significant amount of accidents related to UAS operations occurs because of inexperience among the operators (Dalamagkidis et al., 2008). Consequently, the requirement of operator certification has great importance for successful operation of such systems; especially if UAS safe-system is based on manual pilot override (Dalamagkidis et al., 2008).

2.2.5 Summary

Requirements and restrictions associated with use of UASs are essential to ensure the superior requirement of safety. Unmanned aircraft is a unique class of aircraft that require specially adapted rules. Requirements / restrictions related to airworthiness certification, airspace restrictions, safety margins and operator certification are assumed to be the key elements for this superior requirement of safety.

Chapter 3

Operational strategies for UAV

This chapter proposes and discusses the operational strategies for UAV used for information gathering in Arctic environments.

3.1 Surveillance area

During the drilling operations in Arctic environments it is important to have a clear definition of the area which poses a potential danger for the drilling vessel. According to (Edmond et al., 2011), this is the first step of the ice management process. Surveillance area (SA) is the description that will be used in further referencing to this area. This area is assumed to contain the sea ice in the form of ice floes, ice ridges and icebergs which are moving towards the drilling vessel. The main objective is to perform appropriate path planning for the unmanned aircraft within SA in order to gather necessary information for the ice management system. However, such factors as sea currents, waves, tides and wind will influence the direction and speed of the ice movement. This makes the path planning for the unmanned aircraft challenging, especially if it is desired to cover the same area several times. Furthermore, it is important to consider the location of the SA. As proposed in (Edmond et al., 2011), one possible approach is to divide the area of interest in surveillance- and management zones. The idea is illustrated in Figure 3.1, with following zones defined around the drilling vessel (Edmond et al., 2011):

- The General Surveillance Zone
- The Threat Assessment Zone

- The Ice Drift Corridor
- The Physical Management Zone
- The Emergency Disconnection Zone

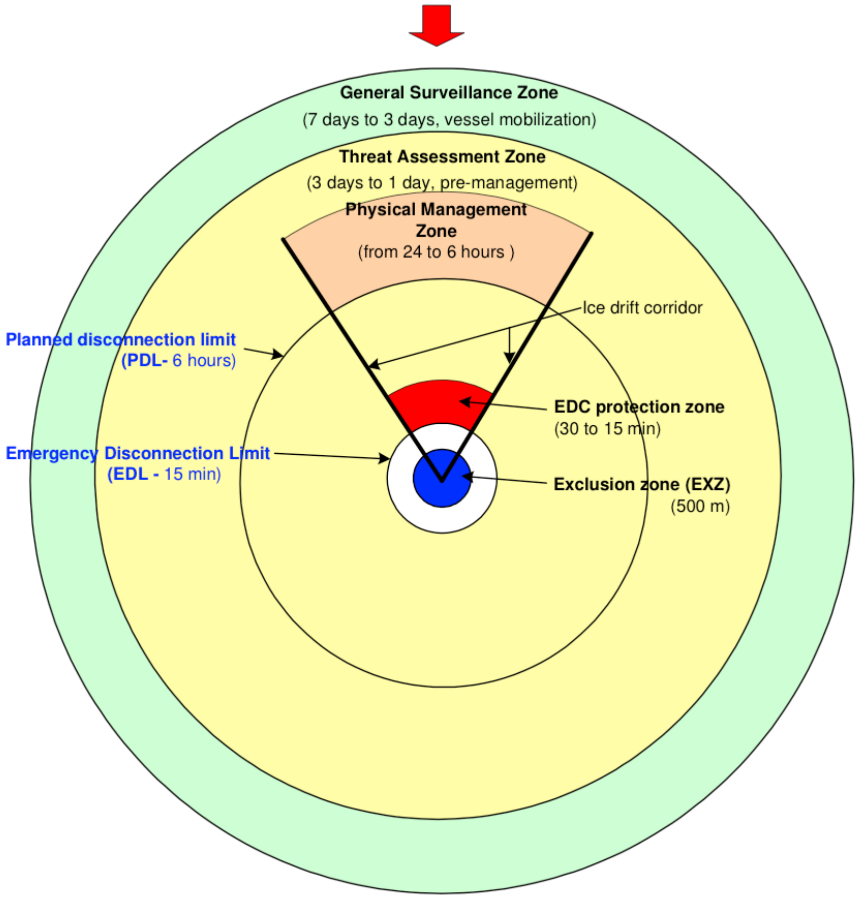

Figure 3.1: Surveillance- and management zones (Edmond et al., 2011).

Based on the presented approach, it is reasonable to assume that the location of the SA can be selected by observing the direction of motion of the ice. Consequently, the most obvious choice of such location is where the ice is moving toward the drilling vessel. The question is: how to find the direction of motion of the ice? One possible approach is to use weather / environmental forecast in combination with satellite images to retrieve needed information (Edmond et al., 2011), given that these sources area precise and reliable. Furthermore, the distance between the drilling vessel and the location of the SA is also significant. So, how distant should the SA be from the drilling vessel? As stated in (Edmond et al., 2011), it depends on the scale of the operation. For example, if long-term forecasts of the sea ice are of interest, then it is meaningful to locate the SA far away from the drilling vessel.

So, further in this work it will be assumed that the SA is the area which poses potential danger for the drilling vessel containing the sea ice drifting towards the drilling vessel. Furthermore it is also assumed that the appropriate location of such area can be chosen based on information provided by the weather / environmental forecasts and satellite images.

3.2 Strategies

Appropriate strategy for analysis of the SA is essential to achieve successful operation of the drilling vessel in Arctic environments. The safety of the drilling vessel and the crew is highly prioritized. Therefore, it is of great importance that the analysis of the SA is performed such that potential ice threats can be detected in time for physical ice management or safe evacuation (Edmond et al., 2011). The strategies for performing such analysis will be proposed and discussed in this section.

3.2.1 Analysis of the surveillance area defined as a grid

Let us look at a situation where it is desirable to achieve purposeful coverage of a specific area (in the ice-covered waters of the Arctic basin) for ice threat identification and forecasting of the ice drift. In this case the SA can be defined as a grid, where each element of this grid will be referred to as a cell of the SA. Figure 3.2 illustrates this idea, where presented SA consists of sixteen cells. However, it is reasonable to assume that the number of cells can vary depending on the size of the SA, size of the cells and the scale of the operation. Furthermore, the representation of the cells is an important issue. This leads to the following question: how can one represent the cells within the SA? One possibility is to use the approach presented in (Reinl and von Stryk, 2007); according to this approach one can represent each cell as a coordinate **C** (center of the cell) combined with a distance **d** form a given coordinate to the edge of the corresponding cell. Consequently, this method gives us a quadratic SA which is illustrated in Figure 3.3.

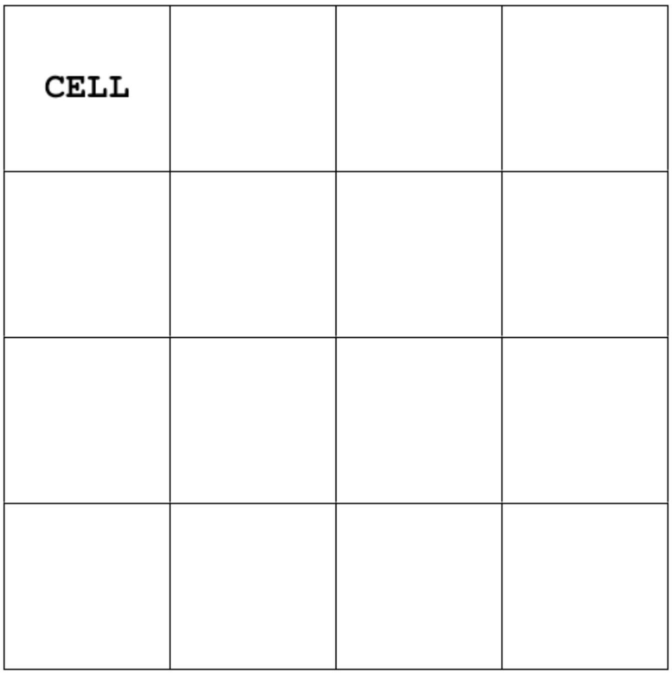

Figure 3.2: Surveillance area defined as a grid.

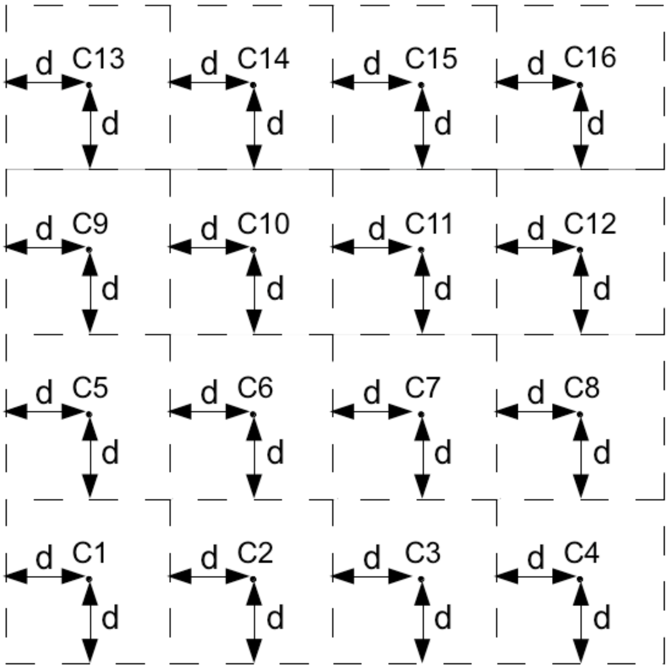

Figure 3.3: Representation of the cells within the surveillance area.

Furthermore, assume that each cell of the SA contains a smaller cell. These cells will be referred to as sub-cells. The sub-cells are basically represented in the same way as all of the cells within the SA. The main difference is that the center of each cell is assumed to be fixed relative to the position of the drilling vessel, while the center of each sub-cell may move in any direction replicating the ice flow within the corresponding cell. Assume that the ice flow can be represented as illustrated in Figure 3.4, and that the f_x and f_y are the velocities of the ice flow along the x- and y-axis given in m/s.

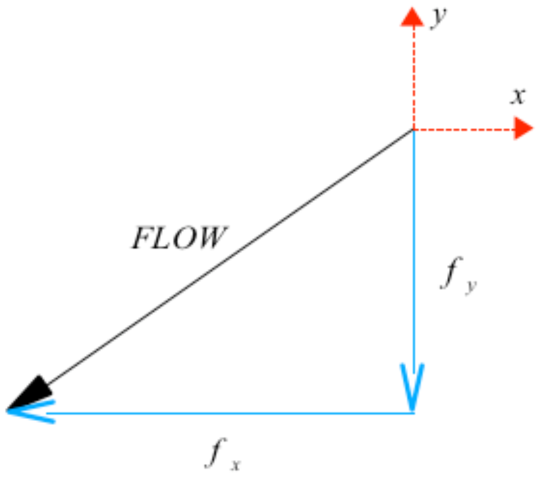

Figure 3.4: Representation of the ice flow.

This means that the motion of the ice can be described by two dimensional vector defined in (3.1), where information about the ice flow is assumed to be taken from the weather / environmental forecasts and satellite images.

$$U_F = \begin{bmatrix} f_x \\ f_y \end{bmatrix} \tag{3.1}$$

Resulting SA is illustrated in Figure 3.5, where each cell contains a sub-cell marked with a red dotted line. The main motivation for introducing these sub-cells is the desire to replicate the ice movement within the SA.

The next step is to perform appropriate path planning for the unmanned aircraft within the SA. As already mentioned, the coverage is prioritized. Consequently, it is reasonable to require that all of the cells within the SA should be visited during the flight of the UAV. Additionally, one can require that each cell and the corresponding sub-cell are visited simultaneously. As a result, the flight path of the unmanned aircraft is expected to be corrected depending on the ice movement within the SA. Subsequently, this will probably make it possible to keep track of the ice threats within the SA.

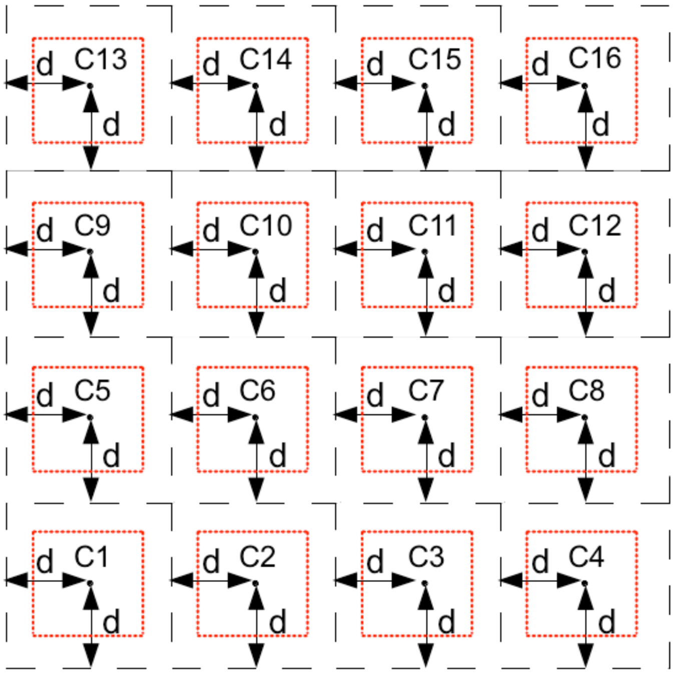

Figure 3.5: SA with sub-cell.

Advantages and disadvantages

There are several advantages and disadvantages with the proposed strategy. Let us start with brief discussion of the advantages.

As already mentioned all of the cells within the SA is to be visited during the flight of the unmanned aircraft. This gives the possibility to cover the SA entirely, and subsequently gain the overview of the sea ice that may pose potential risk for the drilling vessel. Furthermore, the sub-cell of each cell is intended to move in the same way as the sea ice drifts in the corresponding cell. This will probably provide the information about how the sea ice (within a specific cell) changes its position as time goes by. This is advantageous especially if it is desirable to fly over the same area several times. In practice this means that the flight path of the unmanned aircraft is to be corrected depending on the motion of the ice within the SA.

Let us take a look at the potential disadvantages of this strategy. It is clear that the area to be covered can vary in size, depending on the scale of the operation. If the SA is large, then it may require long period of time to achieve entire coverage of the area of interest. This can be a problem, especially if there is a possibility of rapid changes in the environments over a short period of time. This puts a limit on the size of the SA. Furthermore, the process of visiting all cells of the SA can potentially be a time consuming. It is also reasonable to assume that the operational speed of the unmanned aircraft may be affected by the size of the SA. Given that the SA is large it may be motivating to fly faster to minimize the time needed to achieve appropriate coverage. On the other hand this can affect the cost of the flight in terms of fuel consumption, which in turn can reduce the duration of the flight for the unmanned aircraft.

3.2.2 Analysis of the surveillance area defined as a sector

Let us look at the situation where it is desirable to observe the ice flow-line which goes through the position of the drilling vessel. This means that the ice which is on the collision course with the drilling vessel is to be observed. Furthermore, assume once more that the direction and speed of the ice movement can be found from weather / environmental forecast and satellite images. In this case it is attractive to use similar approach as in (Edmond et al., 2011), where it is chosen to introduce the area referred to as ice drift corridor (IDC). This approach is illustrated in Figure 3.6. The ice threats within the IDC introduce significant collision risk for the drilling vessel (Edmond et al., 2011). However, the IDC constitutes a minor part of the SA illustrated in Figure 3.6. Besides, the cells of the SA that are located outside the IDC are less relevant in this case since none of these cells are to be visited. Consequently, instead of having a grid shaped SA, it is now chosen to shape the SA as a sector. This is illustrated in Figure 3.7. Additionally, Figure 3.7 includes a flow map of the sea ice within the sector. It is easy to determine that the flow-line which goes through the position of the drilling vessel (marked as a red dashed line) constitutes a large risk for the drilling vessel being hit by the sea ice. This leads to the following question: how can one represent the sector such that the flow-line which goes through the position of the drilling vessel stays within the sector?

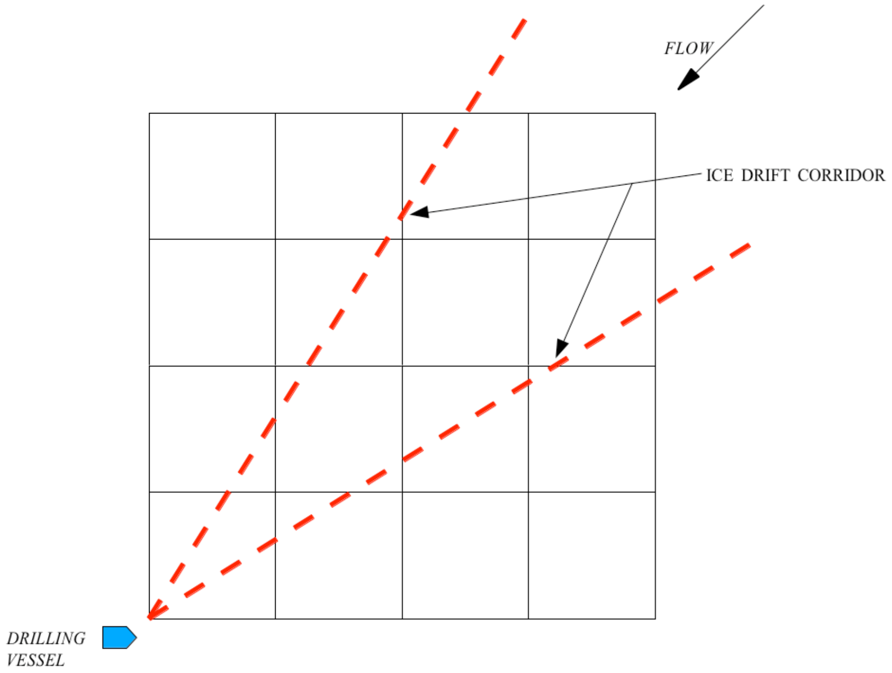

Figure 3.6: Ice drift corridor.

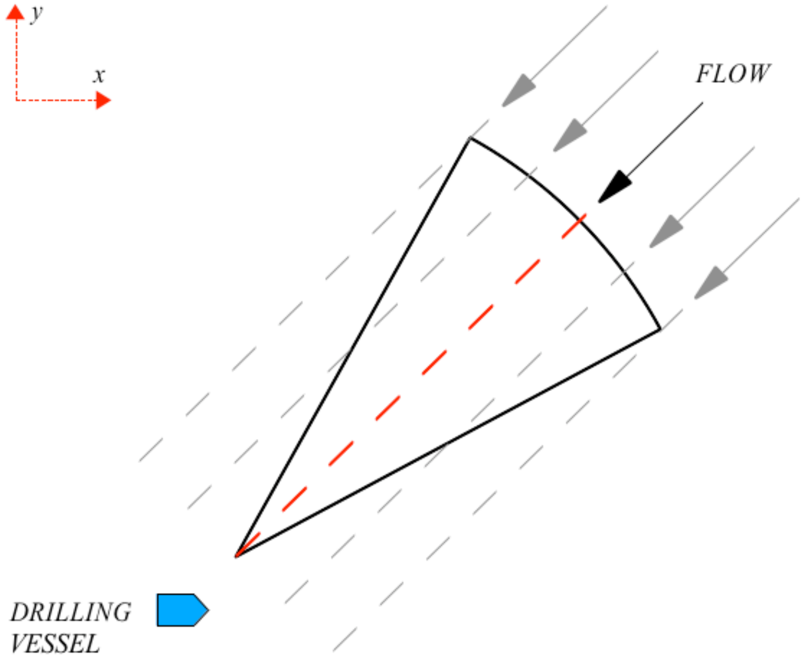

Figure 3.7: Sector shaped SA.

One possible approach is to consider the activation radius of the unmanned aircraft, the ice flow direction and the width of the sector. Figure 3.8 illustrates the idea. Variables presented in Figure 3.8 are defined in the following way:

- R - Activation radius of the unmanned aircraft.
- θ - Ice flow direction.
- 2β - Width of the sector.

Precession of the ice flow direction (θ) is critical. Error in this variable can lead to wrong location of the sector. As a consequence, important information about the ice threats approaching the drilling vessel may be lost, especially if the sector is chosen to be narrow.

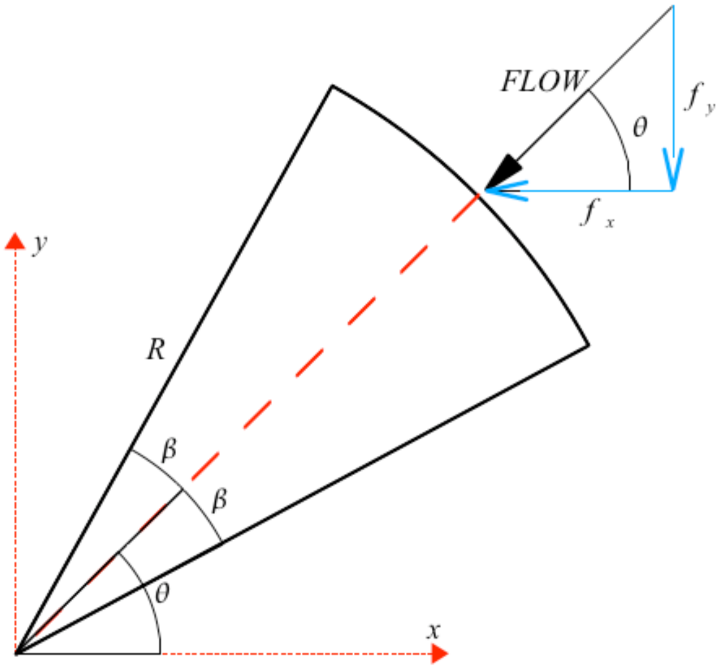

Figure 3.8: Representation of the sector.

The next step is appropriate choice of the cells to be visited within the sector. One possible approach is to use the polar coordinates for this purpose. An example of the arrangement of the cells within a sector is presented in Figure 3.9. The cells are placed along three different lines, which are referred to as L_1, L_2 and L_3 in Figure 3.9. It is reasonable to assume that the arrangement of the cells within a sector can be varied depending on for example the width of the sector and the activation radius of the unmanned aircraft.

When it comes to the path planning strategy for the unmanned aircraft, it is important to remember that the flow-line which goes through the position of the drilling vessel poses great risk for the drilling vessel being hit by the sea ice. In this particular case the flow-line is coincident with L_2 in Figure 3.9. Consequently, it is attractive to visit all of the cells within the sector where the flight path of the UAV is such that those cells which are placed along the L_2 are prioritized to be visited.

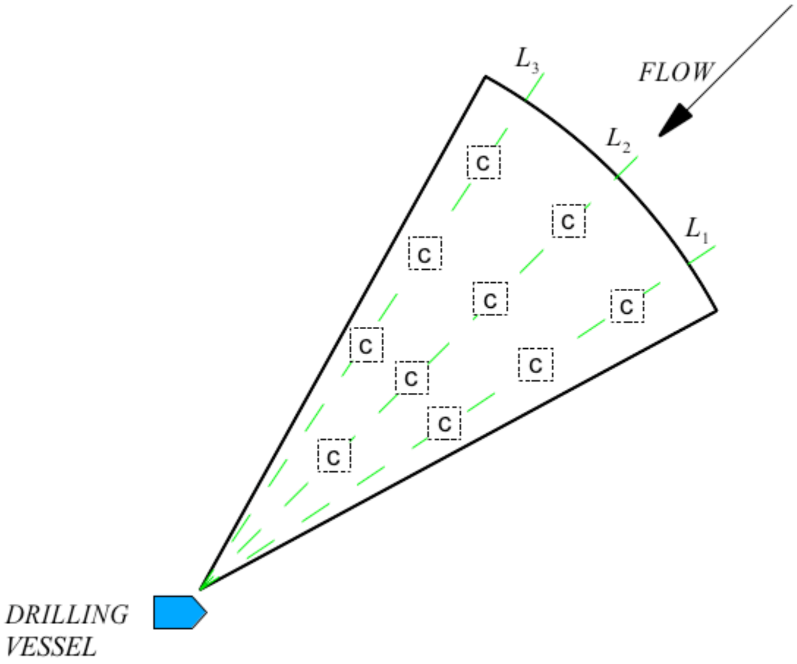

Figure 3.9: Arrangement of the cell within the sector.

Advantages and disadvantages

One of the advantages is that this strategy makes it possible to perform continuous monitoring of the sea ice approaching the drilling vessel. This information is of great importance for planning and execution of the maneuvers related to collision avoidance between the drilling vessel and the sea ice. On the other hand, the coverage provided by this strategy is expected to be less efficient compared to the coverage of strategy presented in Section 3.2.1. This strategy is expected to provide excellent knowledge about what is on the collision course with the drilling vessel, but limited information about the other parts of the potentially unsafe area. Furthermore, it is reasonable to assume that the time needed for coverage of the sector can be limited to an appropriate value. But this will be linked to the dimensions of the sector. Consequently, if the time is a critical factor, it may be tempting to choose a sector as narrow as possible. On the other hand, this can affect the efficiency of the strategy. More precisely, if the sector is too narrow then there is a great chance that important information about the drifting ice will be lost.

3.3 Summary

The path planning strategy for unmanned aircraft, which is intended for operation in Arctic environments, depends on the objectives set for the information gathering in the area of interest. Two strategies were proposed and discussed in this chapter. The first one is referred to as analysis of the SA defined as a grid, where each cell of this grid is to be visited during the flight of the unmanned aircraft (with primary objective to achieve efficient coverage of the SA). The second one is referred to as analysis of the SA defined as a sector. The arrangement of the cells within the sector is such that the collision path between the drilling vessel and the sea ice is properly investigated.

Chapter 4

Optimized flight path for UAV

This chapter contains the formulation of the path planning optimization problems based on strategies presented in Chapter 3.

4.1 Mixed Integer Linear Programming (MILP)

Mixed integer linear programming, or simply MILP, is chosen to be used in path planning optimization for unmanned aircraft. According to (Magatão, 2005), MILP provides a general framework for modeling a large variety of problems. Some of these problems are planning, scheduling and network design (Magatão, 2005). Despite a wide range of applications, it is still the case that the MILP models are computationally difficult to handle (Magatão, 2005). The reason for this is the lack of general and efficient algorithm for solution of the MILP problems, which are also referred to as NP-hard (Culligan, 2006; Magatão, 2005). Furthermore, MILP is intended for optimization problems which have integer variables in the cost function and / or the constraints, where use of the binary variables is also allowed in problem formulation (Kamal et al., 2005; Culligan, 2006). This opens for new possibilities related to enabling non-convex and logical constraints to be included in the problem formulation (Culligan, 2006). The standard form of the MILP can be written in the following way (Culligan, 2006):

$$\begin{aligned} &\min && f(x) = c^T x \\ &\text{subject to:} && b_L \leq Ax \leq b_U \\ & && x_L \leq x \leq x_U \end{aligned} \tag{4.1}$$

where $A \in \mathbb{R}^{m \times n}$; c, x, x_L, $x_U \in \mathbb{R}^n$; b_L, $b_U \in \mathbb{R}^m$. Furthermore, a subset variables $x_I \in x$ is defined as $x_I \in \mathbb{Z}^{n_I}$, and a set of binary variables $x_B \in \{0, 1\}$ is also included in x_I (Culligan, 2006). In this work it is chosen to use YALMIP as a modeling language, implemented as a free toolbox in MATLAB, which is primary used for advanced modeling and solution of convex and non-convex optimization problems (YALMIP Wiki). However, YALMIP is relying on the external solvers for the actual computations. Consequently, it is chosen to use GUROBI optimizer which is applicable for linear programming (LP), quadratic programming (QP) and MILP (Gurobi Optimization).

4.2 YALMIP

As already mentioned YALMIP is the modeling language that will be used in this work. This section gives a brief introduction to this modeling language.

4.2.1 Declaration of variables

YALMIP syntax is identical to the MATLAB syntax (Podhradský, 2010). The declaration of the variables is performed in the same way as in MATLAB. Two types of variables is to be used, these are presented in the following list:

1. `sdpvar()` - command used to define the symbolic decision variables (YALMIP Wiki).

2. `binvar()` - command used to define the decision variables constrained to be binary (0 or 1) (YALMIP Wiki).

More information about declaration of variables can be found in (YALMIP Wiki).

4.2.2 Declaration of constraints

Declaration of constraints can be performed by writing an easy and economic code (Podhradský, 2010). To types of the constraints are to be used; more precisely, equality constraints and inequality constraints. Also, YALMIP makes it possible to handle double inequality in one constraint (Podhradský, 2010). Short examples of declaration of such constraints are presented in the following list:

1. `X = sdpvar(n,1); % X is a variable of dimension n × 1`
   `C = []; % Empty constraint matrix`
   `C = C + [X(:,1) == 10]; % All elements of X are constrained to be 10 (equality constraint)`

2. `X = sdpvar(n,1); % X is a variable of dimension n × 1`
   `C = []; % Empty constraint matrix`
   `C = C + [0 ≤ X(:,1) ≤ 10]; % All elements of X are constrained to stay between 0 and 10 (double inequality constraint)`

More information about declaration of constraints can be found in (YALMIP Wiki).

4.2.3 Objective function

An objective function, is the function which is to be minimized (alternatively maximized) during the solution of the optimization problem. In this work the notation `J` will be used to identify the objective function.

4.2.4 Solver

Once the problem has been formulated it can be solved in the following way (Podhradský, 2010):

- `solvesdp(Constraints, Objective, Solver);`

As already mentioned it was chosen to use GUROBI optimizer for the actual computations. Consequently, formulated problem(s) will be solved in the following way:

1. `options = sdpsettings('solver','gurobi');`
2. `solvesdp(Constraints, Objective, options);`

4.3 Problem formulation

This section contains formulation of the path planning optimization problems based on strategies presented in Sections 3.2.1 and 3.2.2. The UAV model used in both cases is taken from (Grøtli and Johansen, 2011).

It is assumed that the UAV can be described by the discrete time model presented in (4.2):

$$\mathbf{p}_{(i+1)} = I_2\mathbf{p}_i + \Delta t I_2 \mathbf{v}_i \ , \tag{4.2}$$

where I_2 is an identity matrix of dimension 2, Δt is the sample time, while $\mathbf{p}_i := (x_i, y_i)^T$ and $\mathbf{v}_i := (v_{x,i}, v_{y,i})^T$ represents the position and velocity of the UAV at time-step i, respectively (Grøtli and Johansen, 2011). This model can also be written in the following way:

$$\begin{bmatrix} x \\ y \end{bmatrix}_{i+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_i + \begin{bmatrix} \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}_i \ . \tag{4.3}$$

Moreover, only a planar motion of the unmanned aircraft is considered. This means that the unmanned aircraft is assumed to be operated at constant altitude above the ice-covered waters of the Arctic basin. According to (Richards and How, 2002), this is a common restriction in air traffic models.

4.3.1 Problem formulation - Analysis of the surveillance area defined as a grid

Dynamics constraints of the aircraft

According to (Richards and How, 2002; Grøtli and Johansen, 2011), the dynamics constraints of the aircraft can be written in the linear form:

$$\mathbf{p}_{(i+1)} = I_2\mathbf{p}_i + \Delta t I_2 \mathbf{v}_i \ , \tag{4.4}$$

$$\mathbf{p}_0 = \mathbf{p}_{\text{init}} \ , \tag{4.5}$$

$\forall i \in \{0, \ldots, N-1\}$, where $\mathbf{p}_{\text{init}}$ is the initial state of the aircraft while $N := T/\Delta t$ represents the number of time-steps and T is the optimization horizon. Furthermore, it was chosen to use both velocity and acceleration constraints. According to (Richards and How, 2002; Culligan, 2006), the acceleration constraints are given by:

$$a_{x,i} \sin\left(\frac{2\pi m}{M_{\text{vel}}}\right) + a_{y,i} \cos\left(\frac{2\pi m}{M_{\text{vel}}}\right) \leq a_{\max} \cos\left(\frac{\pi}{M_{\text{vel}}}\right) \ , \tag{4.6}$$

$\forall i \in \{0, \ldots, N-1\}$, $\forall m \in \{1, \ldots, M_{\text{vel}}\}$, where M_{vel} is an arbitrary number of the constraints used in polygonal approximation and $a_{\max}$ is the maximum acceleration of the unmanned aircraft. The larger the M_{vel} is, the better is approximation. When it comes to velocity constraints, the same approach is used as for the acceleration constraints. However, in this particular case it is chosen to implement both upper and lover velocity constraints. According to (Grøtli and Johansen, 2011; Culligan, 2006), this can be done by:

$$v_{x,i} \sin\left(\frac{2\pi m}{M_{\text{vel}}}\right) + v_{y,i} \cos\left(\frac{2\pi m}{M_{\text{vel}}}\right) \leq S_i^* \ , \tag{4.7}$$

$$\left(v_{x,i} \sin\left(\frac{2\pi m}{M_{\text{vel}}}\right) + v_{y,i} \cos\left(\frac{2\pi m}{M_{\text{vel}}}\right)\right) \alpha \geq S_i^* - L\left(1 - b_{i,m}^{\text{vel}}\right) \ , \tag{4.8}$$

$$S_i^* = S_i \cos\left(\frac{\pi}{M_{\text{vel}}}\right) \ , \tag{4.9}$$

$$v_{\min} \leq S_i \leq v_{\max} \ , \tag{4.10}$$

$\forall i \in \{0, \ldots, N-1\}$, $\forall m \in \{1, \ldots, M_{\text{vel}}\}$, where L is a constant which can be chosen arbitrary large. Furthermore, α is a constant which is used to adjust the accuracy of the approximation (Grøtli and Johansen, 2011). Finally, it is also required that binary variable $b_{i,m}^{\text{vel}}$ is constrained in the following way (Grøtli and Johansen, 2011):

$$\sum_{m=1}^{M_{\text{vel}}} b_{i,m}^{\text{vel}} = 1 \ , \tag{4.11}$$

$\forall i \in \{0, \ldots, N-1\}$, which means that the speed of the unmanned aircraft can not be lower that $v_{\min}$.

Grid constraints

As already mentioned in Section 3.2.1, the SA is defined as a grid. Each element of this grid is referred to as a cell. Furthermore, each cell contains an sub-cell which replicates the motion of the ice within the corresponding cell. Let us start with representation of the cells within the SA. The idea for doing this is taken from (Richards and How, 2002; Reinl and von Stryk, 2007). Each cell of the SA can be represented by introducing the constraints:

$$\begin{array}{rcl} x_i - (x_{C,k} + d_{\text{cell}}) & \leq & M_{\text{big}}(1 - b_{i,k}^{\text{cell}}) \ , \\ x_i - (x_{C,k} - d_{\text{cell}}) & \geq & -\ M_{\text{big}}(1 - b_{i,k}^{\text{cell}}) \ , \\ y_i - (y_{C,k} + d_{\text{cell}}) & \leq & M_{\text{big}}(1 - b_{i,k}^{\text{cell}}) \ , \\ y_i - (y_{C,k} - d_{\text{cell}}) & \geq & -\ M_{\text{big}}(1 - b_{i,k}^{\text{cell}}) \ , \end{array} \tag{4.12}$$

$\forall i \in \{0, \dots, N-1\}$, $\forall k \in \{1, \dots, W\}$, where W is the number of cells within the SA, M_{big} is a large positive number (which must be larger than any position / velocity to be reached in the problem), while $b_{i,k}^{\text{cell}}$ is a binary variable which is true when cell k is visited and false otherwise (Richards and How, 2002). Furthermore, d_{cell} represents the distance from the center of a cell k to the edge of the corresponding cell. Each cell-center k is described by:

$$C_k = \begin{bmatrix} x_{C,k} \\ y_{C,k} \end{bmatrix} \ . \tag{4.13}$$

When it comes to representation of the sub-cells, the same approach is used. However, in this case the motion of the ice is taken into account. As already mentioned in Section 3.2.1, the motion of the ice is described in the following way:

$$U_F = \begin{bmatrix} f_x \\ f_y \end{bmatrix} \ . \tag{4.14}$$

Consequently, each sub-cell can be represented by introducing the constraints:

$$\begin{array}{rcl} x_i - (x_{C,k} + d_{\text{s-cell}} + f_x \cdot \Delta t \cdot \text{c}) & \leq & M_{\text{big}}(1 - b_{i,k}^{\text{cell}}) \ , \\ x_i - (x_{C,k} - d_{\text{s-cell}} + f_x \cdot \Delta t \cdot \text{c}) & \geq & -\ M_{\text{big}}(1 - b_{i,k}^{\text{cell}}) \ , \\ y_i - (y_{C,k} + d_{\text{s-cell}} + f_y \cdot \Delta t \cdot \text{c}) & \leq & M_{\text{big}}(1 - b_{i,k}^{\text{cell}}) \ , \\ y_i - (y_{C,k} - d_{\text{s-cell}} + f_y \cdot \Delta t \cdot \text{c}) & \geq & -\ M_{\text{big}}(1 - b_{i,k}^{\text{cell}}) \ , \end{array} \tag{4.15}$$

$\forall i \in \{0, \dots, N-1\}$, $\forall k \in \{1, \dots, W\}$, where $d_{\text{s-cell}}$ represents the distance from the center to the edge of a sub-cell k, while c is a counter which resets each time the center of the sub-cell k reaches the edge of the corresponding cell. Finally, it is also required that:

$$\sum_{i=1}^{N} b_{i,k}^{\text{cell}} = 1 \ , \tag{4.16}$$

$\forall k \in \{1, \dots, W\}$. The idea is taken from (Richards and How, 2002), which means that each cell and corresponding sub-cell is to be visited during the flight of the unmanned aircraft.

Objective function

The next step is the appropriate choice of the objective function. Nevertheless, let us first repeat once more the purpose of the proposed strategy; it is desirable to achieve the coverage of a specific area, in the ice-covered waters of Arctic basin, for ice threat identification and forecasting of the ice drift. As already mentioned, the area to be investigated is defined as a grid, where each element of this grid is referred to as a cell (which additionally contains a sub-cell). In this context it is reasonable to assume that appropriate coverage of such area may be achieved if each cell of this grid (and the corresponding sub-cell) is visited during the flight of the unmanned aircraft. This requirement is already implemented in (4.16). However, it is also attractive that the time needed to achieve full coverage of the SA is minimized; since it is advantageous to avoid the situation where the collected information is partially outdated at the time the full coverage of the SA is complete. Based on (Richards and How, 2002) and what is presented above, the proposed objective function takes the following form:

$$J_{TIME} = \beta \sum_{k=1}^{W} \sum_{i=0}^{N-1} i \cdot \Delta t \cdot b_{i,k}^{\text{cell}} = \beta \sum_{k=1}^{W} \sum_{i=0}^{N-1} T_i \cdot b_{i,k}^{\text{cell}} \ , \tag{4.17}$$

where β is a tuning parameter, while T_i is the time elapsed at time-step i. However, it is likely to expect that the J_{TIME} alone is insufficient to achieve realistic path planning for the unmanned aircraft. The reason for this is that the cost related to the acceleration of the unmanned aircraft is not included in J_{TIME}. It is reasonable to assume that the indiscriminate acceleration use will lead to a flight path which is not consistent with reality. Additionally, fuel consumption is assumed to be proportional to the acceleration (Grøtli and Johansen, 2011). This is one more reason to include a cost related to the acceleration of the unmanned aircraft. According to (Grøtli and Johansen, 2011; Richards and How, 2002), this can be done as follows:

$$J_{ACC} = \epsilon \sum_{i=0}^{N-2} \left(|a_{x,i}| + |a_{y,i}| \right) \ . \tag{4.18}$$

The absolute value of the acceleration components, $a_{x,i}$ and $a_{y,i}$, can be found based on the additional constraints (Grøtli and Johansen, 2011):

$$\begin{aligned} (v_{j,k} - v_{j,i}) &\leq w^{\text{acc}}_{j,i} \ , \\ -\ (v_{j,k} - v_{j,i}) &\leq w^{\text{acc}}_{j,i} \ , \end{aligned} \tag{4.19}$$

$k = i+1$, $\forall i \in \{0, \ldots, N-2\}$, $\forall j \in \{x, y\}$, where $\mathbf{w}^{\text{acc}}_i = \left(w^{\text{acc}}_{x,i}, w^{\text{acc}}_{y,i}\right)^T = \left(|a_{x,i}|, |a_{y,i}|\right)^T$, while ϵ is tuning parameter. The cost function in (4.18) can now be written as:

$$J_{ACC} = \epsilon \sum_{i=0}^{N-2} \left(w^{\text{acc}}_{x,i} + w^{\text{acc}}_{y,i}\right) \ . \tag{4.20}$$

The complete problem is then to minimize J_{TOTAL}, which is defined as:

$$J_{TOTAL} = J_{ACC} + J_{TIME} \ , \tag{4.21}$$

subject to the constraints in (4.4), (4.5), (4.6), (4.7), (4.8), (4.9), (4.10), (4.11), (4.12), (4.15), (4.16) and (4.19).

4.3.2 Problem formulation - Analysis of the surveillance area defined as a sector

Dynamics constraints of the aircraft

Dynamics constraints used in this case are exactly the same as for the previous case. More precisely, following equations are used: (4.4), (4.5), (4.6), (4.7), (4.8), (4.9), (4.10) and (4.11).

Sector constraints

As already mentioned in Section 3.2.2, the SA is defined as a sector. Proposed approach for representation of this sector is divided in two steps:

1. Definition of the potential area of operation for the unmanned aircraft.

2. Definition of the upper and lower borders of the sector.

Let us start with the first step. The area of operation is assumed to be a circular area with origin which coincides with the position of the drilling vessel. The distance from the origin to the edge of this area is referred to as the activation radius of the unmanned aircraft. One possible approach for representation of this area is to use polygonal approximation, which means that the area of operation can be represented by using a set of linear constraints. This idea is taken from (Richards and How, 2002). The Figure 4.1 illustrates the idea.

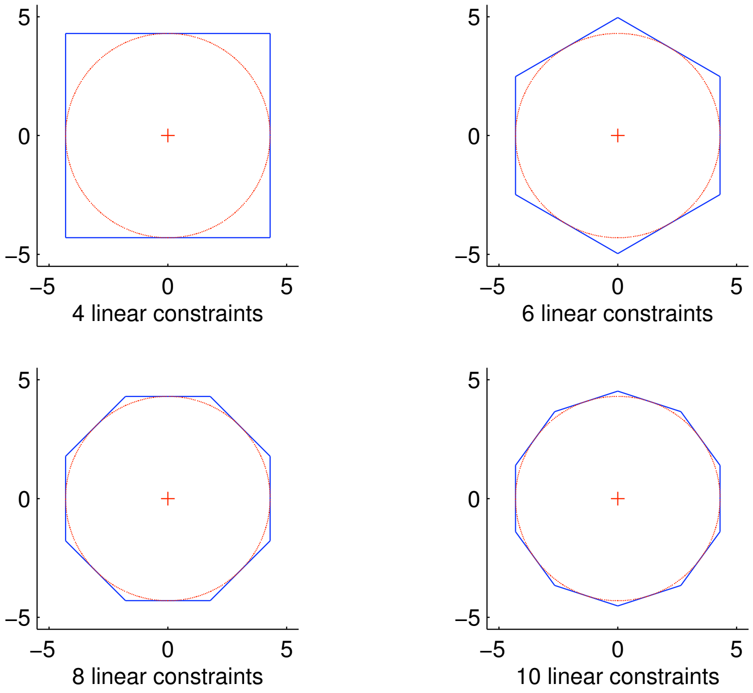

Figure 4.1: Polygonal approximation.

From Figure 4.1 it can be observed that the higher the number of linear constraints, the better approximation is achieved. Consequently, potential area of operation for the unmanned aircraft can be defined by:

$$x_i \sin\left(\frac{2\pi n}{M_{\text{pos}}}\right) + y_i \cos\left(\frac{2\pi n}{M_{\text{pos}}}\right) \leq R_a \ , \tag{4.22}$$

$\forall i \in \{0, \ldots, N-1\}$, $\forall n \in \{1, \ldots, M_{\text{pos}}\}$, where M_{pos} is the number of linear constraint used in polygonal approximation of the area of operation for the unmanned aircraft, while R_a is the activation radius. The next step is to define the upper and lower borders of the sector. Once again this can be done by using a linear constraints. Assuming that the flow direction of the ice (referred to as θ) and the width of the sector (referred to as 2β) are known, then the upper and lower borders of the sector can be represented in the following way:

$$x_i \tan(\theta - \beta) \leq y_i \leq x_i \tan(\theta + \beta) \ , \tag{4.23}$$

$\forall i \in \{0, \ldots, N-1\}$. Consequently, the combination of (4.22) and (4.23) gives us mathematic representation of the sector of interest. When it comes to representation of the cells within the sector of interest, it was chosen to use same approach as in Section 4.3.1. More precisely, constraints in (4.12) are used for this purpose.

Objective function

As already mentioned in Section 3.2.2, the purpose of the proposed strategy is the observation of the ice threats which are on the collision course with the drilling vessel. Consequently, in this case it is desirable that the flight path for the unmanned aircraft is chosen such that this collision course is properly investigated.

First of all, let us introduce a cost related to acceleration of the unmanned aircraft; the proposed cost function is equivalent with J_{ACC} in (4.20):

$$J_{ACC} = \epsilon \sum_{i=0}^{N-2} \left(w_{x,i}^{\text{acc}} + w_{y,i}^{\text{acc}} \right) \ , \tag{4.24}$$

where $w_{x,i}^{\text{acc}}$ and $w_{y,i}^{\text{acc}}$ are defined in (4.19). As already mentioned in Section 4.3.1, the main motivation for doing this is to achieve a flight path which is consistent with the reality. Furthermore, as for the previous case the cells within SA is to be visited during the flight of the unmanned aircraft. However, instead of that each cell is visited only once, it is now desirable that each cell is visited several times and that the visiting time interval is minimized. One possible approach for doing this is to introduce the following cost function:

$$J_{\theta} = \gamma \sum_{k=1}^{W} \sum_{i=0}^{N-1} \theta_{i,k} \ , \tag{4.25}$$

where γ is a tuning parameter, while $\theta_{i,k}$ represents how long ago the cell k was visited at the time-step i. More precisely, θ is a counter which resets each time a corresponding cell is visited. Moreover, representation of the θ is based on the additional constraints:

$$\begin{array}{rcl} \theta_{i,k} & \leq & \theta_{i-1,k} + \Delta t \cdot \left(1 + N \cdot b_{i,k}^{\text{cell}}\right) \ , \\ \theta_{i,k} & \geq & \theta_{i-1,k} + \Delta t \cdot \left(1 - N \cdot b_{i,k}^{\text{cell}}\right) \ , \\ \theta_{i,k} & \leq & N \cdot \Delta t \cdot \left(1 - b_{i,k}^{\text{cell}}\right) \ , \\ \theta_{i,k} & \geq & - \ N \cdot \Delta t \cdot \left(1 - b_{i,k}^{\text{cell}}\right) \ , \end{array} \tag{4.26}$$

$\forall i \in \{1, \ldots, N-1\}$ and $\forall k \in \{1, \ldots, W\}$, while

$$\theta_{0,k} = \Delta t \ , \tag{4.27}$$

$\forall k \in \{1, \ldots, W\}$. Therefore, it is reasonable to expect that by minimizing the value of the J_θ, the visiting interval will also be minimized. However, as already mentioned in Section 3.2.2, those cells which area placed along the flow-line which goes through the position of the drilling vessel is prioritized to be visited. This means that one needs to vary the profitability associated with the visiting order of the cells in the SA. One possible approach for doing this is to introduce a visiting cost for each cell in the SA; the cells which have lowest cost are prioritized to be visited. Proposed cost function in this case is:

$$J_{VC} = \sum_{k=1}^{W} \sum_{i=0}^{N-1} C_{\text{cost}}(k) \cdot b_{i,k}^{\text{cell}} \ , \tag{4.28}$$

where the $C_{\text{cost}}(k)$ represents the visiting cost related to cell k. It is logical to think that by minimizing the value of the J_{VC}, the cells associated with lowest cost will be visited first. The complete problem is then to minimize J_{TOTAL}, which is defined as:

$$J_{TOTAL} = J_{ACC} + J_\theta + J_{VC} \ , \tag{4.29}$$

subject to the constraints in (4.4), (4.5), (4.6), (4.7), (4.8), (4.9), (4.10), (4.11), (4.12), (4.19), (4.22), (4.23), (4.26) and (4.27).

4.4 Summary

Both strategies proposed in Sections 3.2.1 and 3.2.2 were formulated mathematically in this chapter. It was chosen to use MILP in combination with YALMIP modeling language for formulation of the optimization problems. More precisely, each of the strategies were formulated with a set of linear constraints in combination with a cost function to be minimized.

Chapter 5

Simulations and discussion

This chapter contains simulation results for the strategies proposed and mathematically formulated in Chapters 3 and 4. In each simulation case the following are investigated: flight path of the UAV, coverage of the SA, speed and acceleration of the UAV and the solver-time.

5.1 Introduction to simulations

The simulations are performed on a Dell Optiplex 780, with Intel Core 2 Duo CPU E8400 @3.00 GHz, 4 GB RAM and a Windows 7, 32-bit operating system. Moreover, MATLAB version 7.10.0 (R2010a) and YALMIP version 20101208, in combination with GUROBI optimizer, are used for formulation and solution of the optimization problems. Several simulation cases are to be presented. In all of these cases a single UAV is used. When it comes to initial conditions of the unmanned aircraft, the initial position was chosen to be as follows:

$$\mathbf{p}_{\text{init}} = [0, 0]^{\mathrm{T}} \ , \tag{5.1}$$

which is assumed to be coincident with the position of the drilling vessel. The initial velocity, on the other hand, was chosen such that:

$$|\mathbf{v}_0| \geq v_{\min} \ . \tag{5.2}$$

The main reason for this is the speed constraint given in (4.10). The inequality in (5.2) must be satisfied in order to avoid infeasible solution of the optimization problems. Important parameters describing the physical limitation of the unmanned aircraft are given in Table 5.1.

Parameter	Value
v_{min}	5 m/s
v_{max}	25 m/s
a_{max}	10 m/s^2

Table 5.1: Physical limitation of the UAV.

5.2 Surveillance area defined as a grid

This section gives a presentation of the simulation results where SA is defined as a grid. Two cases are investigated; constant and variable ice flow. As mentioned earlier, main purpose of this strategy is to achieve appropriate coverage of the SA. The superior requirement is to visit all parts of the SA and avoid blind zones during the flight of the UAV.

5.2.1 Constant ice flow

Let us start with the case where it is assumed that the ice flow is constant within the SA. Description of the ice flow used in this case is given in Table 5.2. Furthermore, information about the SA is summarized in Table 5.3. The optimization horizon is 90 s, and the discretization step is 3 s.

Parameter	Value	Parameter	Value
$U_F = [f_x, f_y]^{\mathrm{T}}$	$[-0.3 \text{ m/s} \;,\; -0.3 \text{ m/s} \;]^{\mathrm{T}}$	$\lvert U_F \rvert$	0.42 m/s

Table 5.2: Grid - Constant flow.

Parameter	Value
Number of cells	16
Cell size	100 m × 100 m
SA size	400 m × 400 m

Table 5.3: Grid - Summarized information.

The flight path of the UAV is illustrated in Figure 5.1, which is represented with a green dashed line. Furthermore, the ice flow is represented with a blue arrow in the center of each cell within the SA. One can see that each cell (and corresponding sub-cell) of the SA is visited during the flight of the UAV. This means that the superior requirement of visiting all parts of the SA is achieved. Moreover, the visiting order of the cells (and corresponding sub-cells) is assumed to be advantageous. From Figure 5.1 one can se that the flow-line which goes through the position of the drilling vessel (origin), also goes through the cells which are located on / close to the diagonal within the SA. The flight path of the UAV is such that the cells on / close to this diagonal are prioritized to be visited. Consequently, it is reasonable to assume that the potential ice threats approaching the drilling vessel will be discovered at an early stage of the flight.

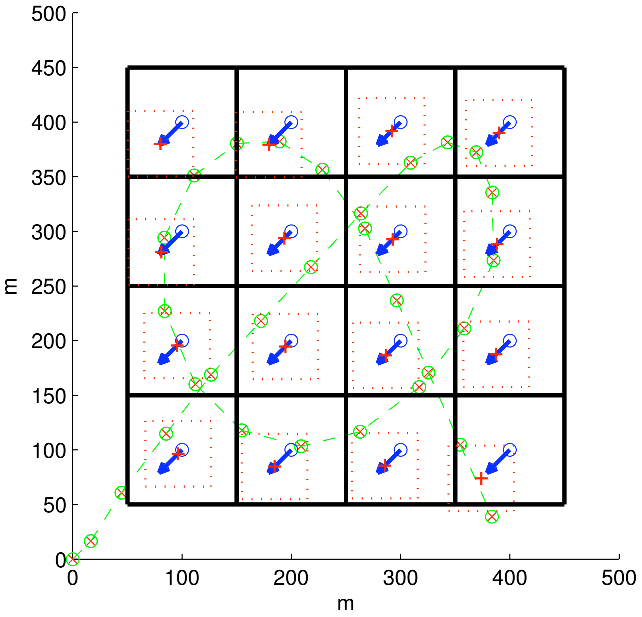

Figure 5.1: Position of the UAV within SA.

On the other hand, let us assume the situation where it is desirable to get image map of the SA. From Figure 5.1 one can see that some of the cells (and corresponding sub-cells) are visited several times, leading to more or less unsystematic overlap. Consequently, the process of merging of the gathered information may be challenging due to this unsystematic overlap. The coverage of the SA is illustrated in Figure 5.2. It is important to remember that achievable coverage depends on several factors, such as instrumentation of the UAV and its operational altitude. None of these factors are considered in this work; therefore, the Figure 5.2 has only demonstrative character. Anyway, it is assumed that the radius of coverage of the equipment attached to the UAV is limited to 60 m (illustrated with red circle). From Figure 5.2 one can see that most of the SA is covered during the flight.

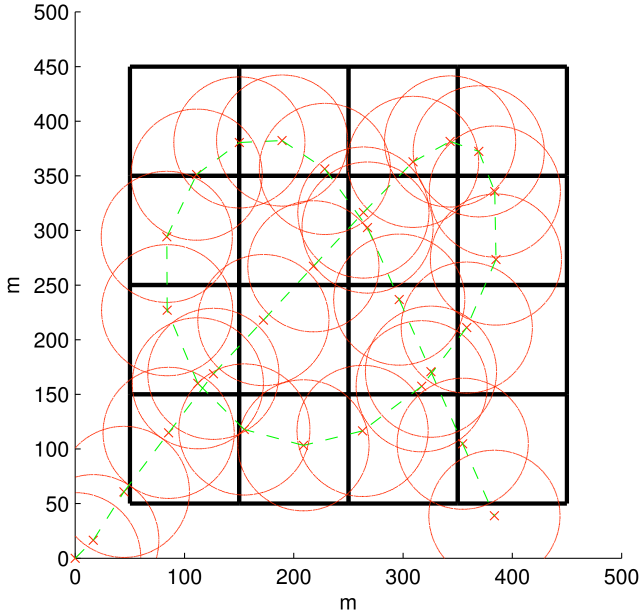

Figure 5.2: Coverage of the SA.

When it comes to the speed and acceleration of the UAV, one can see that these remain within the chosen limits. Figures 5.3 and 5.4 illustrate the speed and acceleration of the UAV, respectively. The speed of the UAV stays within the predefined range while the acceleration never exceeds its maximum limit. However, from Figure 5.4 one can see rapid changes in accelerations which occur within a short period of time. Finally, Table 5.4 shows the solver-time and achieved value of the objective function. The source code for this simulation case can be found in Appendix A.

Surveillance area	Optimal J	Solver-time
Grid (Constant flow)	660.2	1260 s

Table 5.4: Solver-time and objective function value.

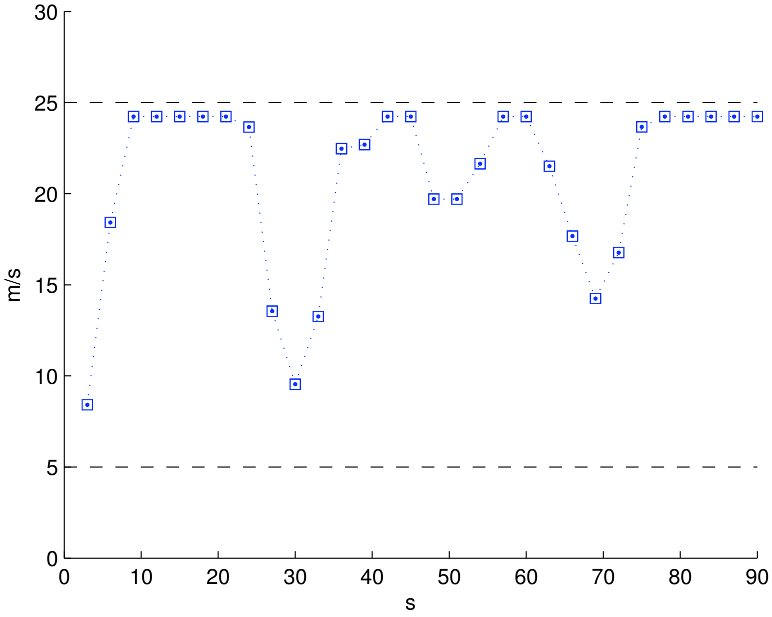

Figure 5.3: Speed of the UAV.

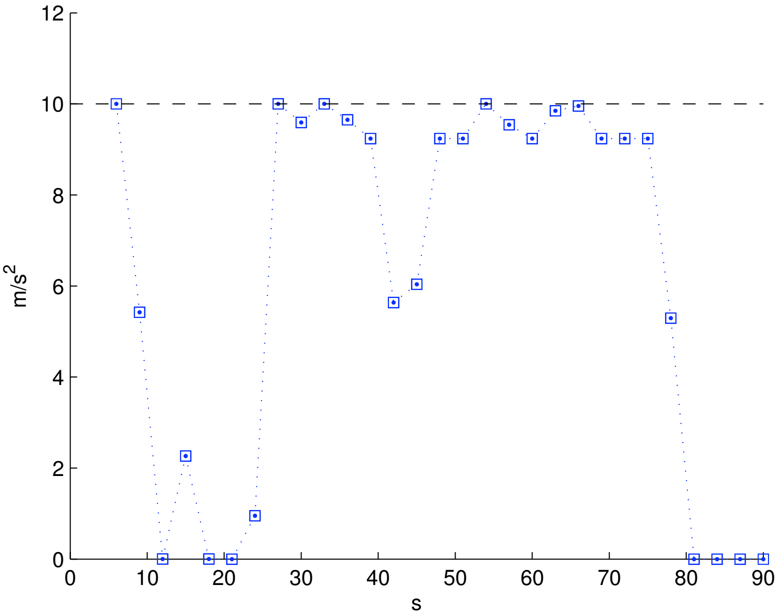

Figure 5.4: Acceleration of the UAV.

5.2.2 Variable ice flow

In this case a variable ice flow is used, which varies both in strength and direction for each cell of the SA; description of the ice flow is summarized in Table 5.5. Detailed information about the ice flow for each cell of the SA can be found in Appendix A, while information about the SA itself is presented in Table 5.3. As for the previous case the optimization horizon is 90 s, and discretization step is 3 s.

Parameter	Value	Parameter	Value
$\lvert U_F\rvert_{\min}$	0.42 m/s	Range of f_x	-0.6 m/s $\leq f_x \leq -0.3$ m/s
$\lvert U_F\rvert_{\max}$	0.85 m/s	Range of f_y	-0.6 m/s $\leq f_y \leq -0.3$ m/s

Table 5.5: Grid - Variable flow.

The flight path of the UAV is illustrated in Figure 5.5, which is represented with a green dashed line. Once again the flight path of the UAV is such that each cell (and the corresponding sub-cell) is visited during the flight. Blue arrows in the center of each cell of the SA, represents the ice flow at the corresponding location; the longer the arrow the stronger is the ice flow. From Figure 5.5 one can see that those cells which are placed on / close to the diagonal within SA are prioritized to be visited. As for the previous case these cells coincide with the flow-line which goes through the position of the drilling vessel (origin). Consequently, presented advantages and potential challenges for the case of the constant ice flow is also valid in this case. Furthermore, the coverage of the SA is illustrated in Figure 5.6. As for Figure 5.2, the result of Figure 5.6 is of demonstrative character. Assumed radius of coverage of the equipment attached to the UAV is 60 m (illustrated with red circle). One can see that most of the SA is covered during the flight. When it comes to the speed and acceleration of the UAV, one can see that none of the predefined limits are broken. Figures 5.7 and 5.8 illustrate the speed and acceleration of the UAV, respectively. However, as in the case of the constant ice flow, one can observe rapid changes in acceleration within a short period of time. Finally, Table 5.6 shows the solver-time and achieved value of the objective function. The source code for this simulation case can be found in Appendix A.

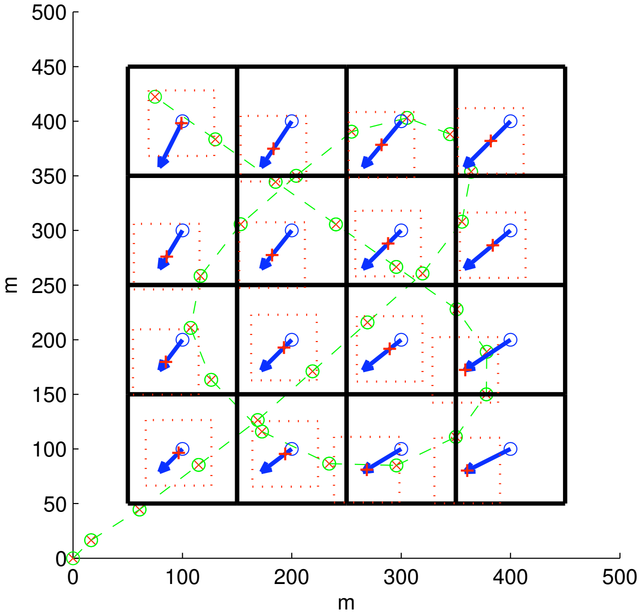

Figure 5.5: Position of the UAV within SA.

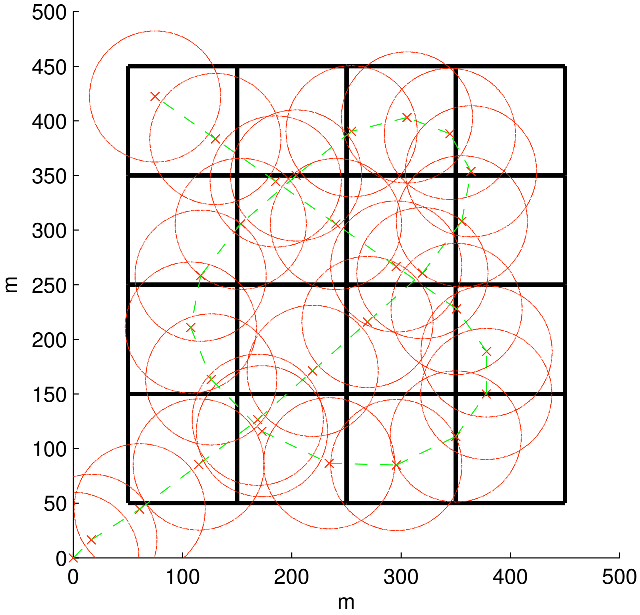

Figure 5.6: Coverage of the SA.

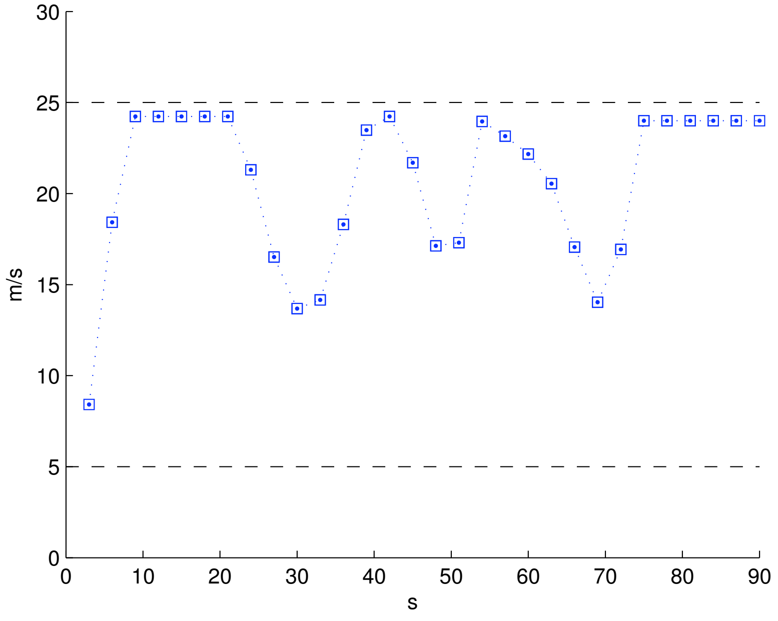

Figure 5.7: Speed of the UAV.

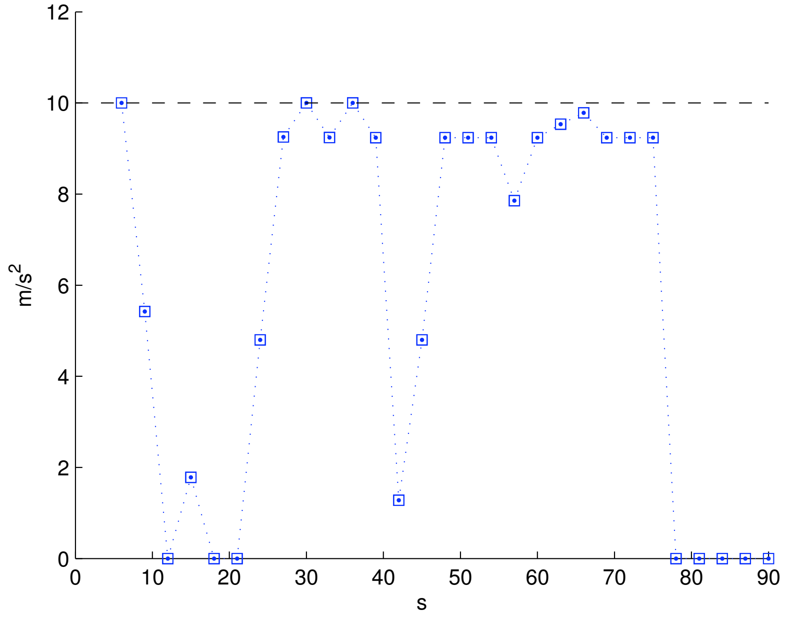

Figure 5.8: Acceleration of the UAV.

Surveillance area	Optimal J	Solver-time
Grid (Variable flow)	654.2	5865 s

Table 5.6: Solver-time and objective function value.

5.2.3 Comparison and discussion

Comparing the results presented in Sections 5.2.1 and 5.2.2 one can see a clear similarity. The path planning results, presented in Figures 5.1 and 5.5, shows that all cells (and corresponding sub-cells) of the SA are visited during the flight of the UAV. However, the visiting order of the cells (and corresponding sub-cells) is somewhat different in the case of constant ice flow compared to the case of variable ice flow. Also the coverage of the SA, presented in Figures 5.2 and 5.6, is very similar; nearly all of the blind zones that appear in both cases are located at the edges of the SA which are most distant from the location of the drilling vessel (origin). Furthermore, in both case the speed and acceleration of the UAV remain within the predefined limits; however, rapid changes in acceleration are observed. It is clearly a problem that should be avoided. One possible solution to this problem is to add jerk constraints to the optimization problems. When it comes to the solver-time, presented in Tables 5.4 and 5.6, it can be observed that it requires significantly more time to solve the optimization problem in the case of variable ice flow compared to the case of constant ice flow.

5.3 Surveillance area defined as a sector

This section gives a presentation of the simulation results where SA is defined as a sector. As in Section 5.2, two cases are investigated: constant and variable ice flow. The main purpose of tested strategy is the appropriate coverage of the sector during the flight of the UAV. The superior requirement in this case, is to choose a flight path of the unmanned aircraft such that the flow-line which goes through the position of the drilling vessel (origin) can be properly investigated.

5.3.1 Constant ice flow

We start with the situation where it is assumed that the ice flow is constant within the sector. Detailed information about the ice flow is given in Table 5.7, while physical dimensions of the sector and the cells within the sector are summarized in Table 5.8. The sector is defined based on the approach presented in Figure 3.8. Furthermore, optimization horizon is set to 90 s, and the discretization step is 3 s. The resulting flight path of the UAV is illustrated in Figure 5.9, marked with a green dashed line. It can be observed that all of the cells within the sector are visited several times during the flight of the UAV.

Parameter	Value	Parameter	Value
$U_F = [f_x, f_y]^{\mathrm{T}}$	$[-0.3$ m/s , -0.3 m/s $]^{\mathrm{T}}$	$\lvert U_F \rvert$	0.42 m/s

Table 5.7: Sector - Constant flow.

Parameter	Value
R	500 m
θ	45°
2β	11.5°
Number of cells	4
Cell size	20 m × 20 m

Table 5.8: Sector - Summarized information.

More precisely, the UAV continues circulating within the sector as expected. Consequently, the flow-line which goes through the position of the drilling vessel (origin) stays covered at any time. Based on the presented result, it is reasonable to assume that the ice threats which are on the collision course with the drilling vessel, will be observed in time for physical ice management / safe evacuation. Furthermore, the coverage of the sector is illustrated in Figure 5.10. Since neither the operational altitude or instrumentation of the UAV are considered in this work, the presented illustration is of demonstrative character; in this case it is assumed that radius of coverage of the equipment attached to the UAV is limited to 40 m (illustrated with a red circle). One can see that the sector is almost completely covered. Moreover, the resulting overlap is clearly systematic and predictable, which in turn makes the process of merging the gathered information straightforward. Figures 5.11 and 5.12 illustrate the speed and acceleration of the UAV, respectively; one can see that both speed and acceleration of the UAV remain within the predefined limits. However, from Figure 5.12 one can observe rapid changes in the acceleration within a short period of time. At last, the solver-time and achieved value of the objective function are presented in Table 5.9. The source code for this simulation case can be found in Appendix B.

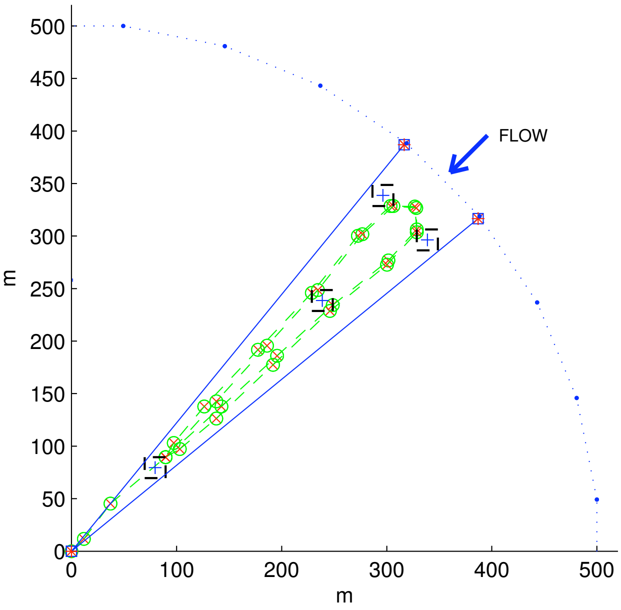

Figure 5.9: Position of the UAV within SA.

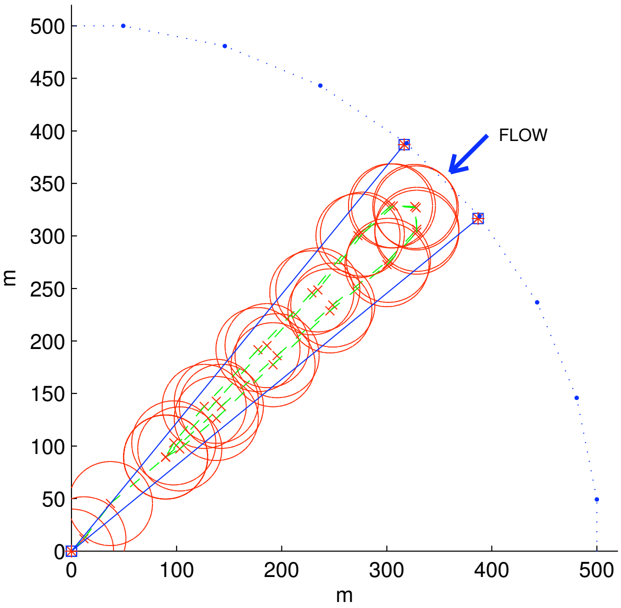

Figure 5.10: Coverage of the SA.

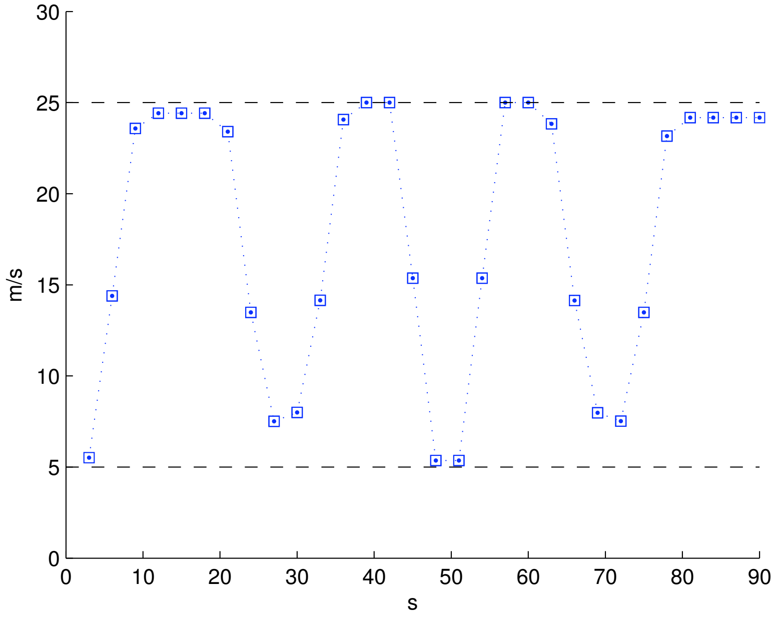

Figure 5.11: Speed of the UAV.

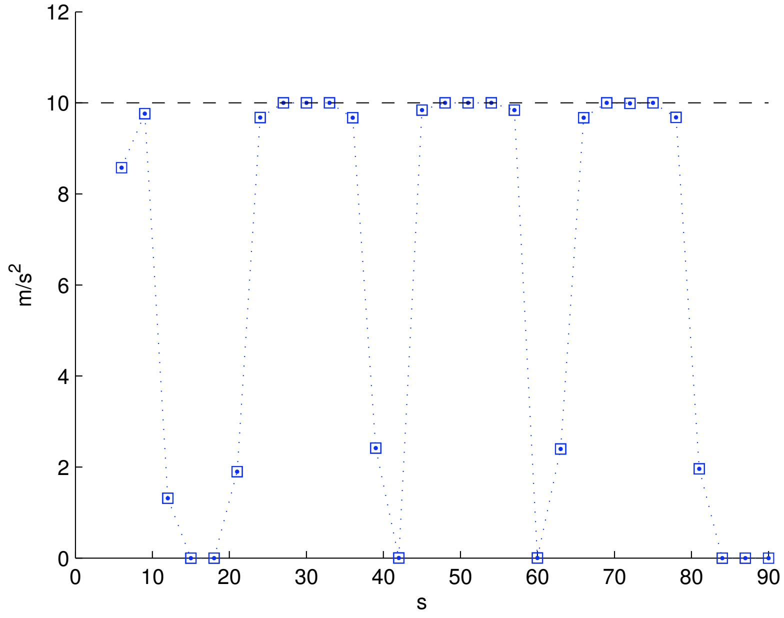

Figure 5.12: Acceleration of the UAV.

Surveillance area	Optimal J	Solver-time
Sector (Constant flow)	328.6	1487 s

Table 5.9: Solver-time and objective function value.

5.3.2 Variable ice flow

Let us consider the case where the ice flow changes as times goes by; Table 5.10 provides detailed information about the ice flow. Furthermore, two sectors are used in this case; one for the coverage of the flow-line described by $U_{F,1}$ and one for the coverage of the flow-line described by $U_{F,2}$. The physical dimensions of these sectors are described in Table 5.11. As for the previous case, the optimization horizon is 90 s and discretization step is 3 s. It is further assumed that after the first 45 s the ice flow changes from $U_{F,1}$ to $U_{F,2}$.

Parameter	Value	Parameter	Value
$U_{F,1} = [f_x, f_y]^{\mathrm{T}}$	$[-0.4$ m/s , -0.6 m/s $]^{\mathrm{T}}$	$\lvert U_{F,1}\rvert$	0.72 m/s
$U_{F,2} = [f_x, f_y]^{\mathrm{T}}$	$[-0.6$ m/s , -0.4 m/s $]^{\mathrm{T}}$	$\lvert U_{F,2}\rvert$	0.72 m/s

Table 5.10: Sector - Variable flow.

Upper sector		Lower sector	
Parameter	Value	Parameter	Value
R	500 m	R	500 m
θ	56°	θ	34°
2β	11.5°	2β	11.5°
Number of cells	4	Number of cells	4
Cell size	20 m × 20 m	Cell size	20 m × 20 m

Table 5.11: Sector - Summarized information.

The flight path of the UAV is illustrated in Figure 5.13, which is represented with a green dashed line. One can observe that the visiting order of the sectors is such that the upper sector is visited first and the lower sector is visited subsequently. Moreover, all of the cells within the corresponding sectors are visited during the flight. As expected, the UAV starts its journey through the cells of the upper sector and continues through the lower sector, where it circulates through the corresponding cells. Consequently, it is expected that both flow-lines, represented by

$U_{F,1}$ and $U_{F,2}$, will be properly covered. Furthermore, the coverage of the sectors is illustrated in Figure 5.14. As for Figure 5.10, the result of Figure 5.14 is of demonstrative character. It is assumed that radius of coverage of the equipment attached to the UAV is limited to 40 m (illustrated with a red circle). One can see that both of the sectors are almost completely covered during the flight. Moreover, the overlap which appears in the lower sector is systematic and predictable. When it comes to the speed and acceleration of the UAV, one can see that these remain within the predefined limits. Figures 5.15 and 5.16 illustrate the speed and acceleration of the UAV, respectively. However, as in the case of constant ice flow, one can observe rapid changes in acceleration within a short period of time. The solver-time and achieved value of the objective function are presented in Table 5.12. The source code for this simulation case can be found in Appendix B.

Surveillance area	Optimal J	Solver-time
Sector (Variable flow)	407	2395 s

Table 5.12: Solver-time and objective function value.

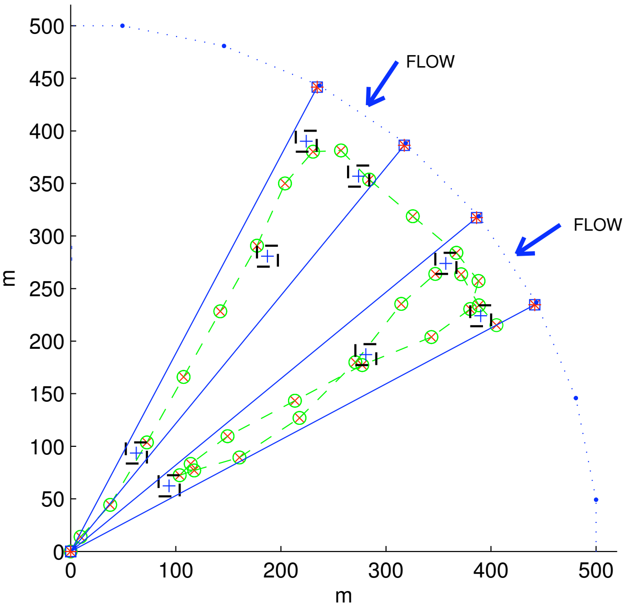

Figure 5.13: Position of the UAV within the SA.

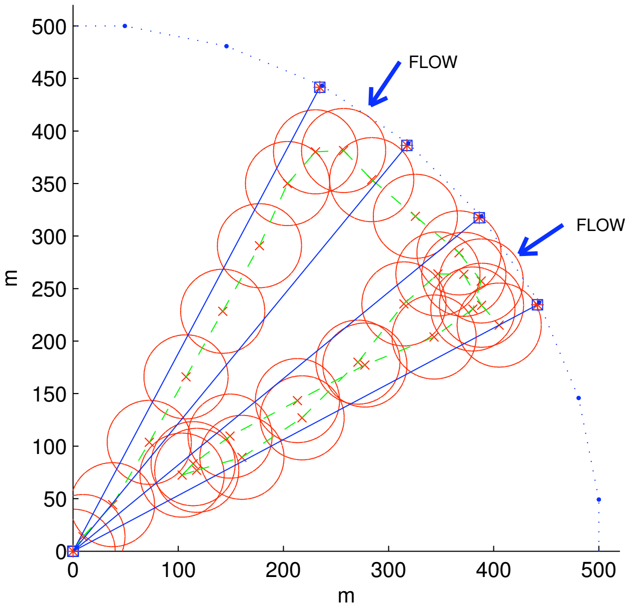

Figure 5.14: Coverage of the SA.

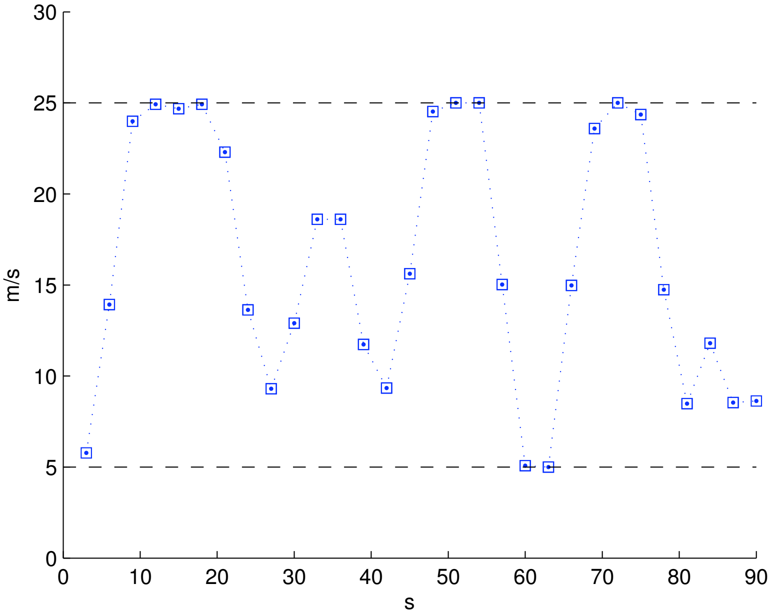

Figure 5.15: Speed of the UAV.

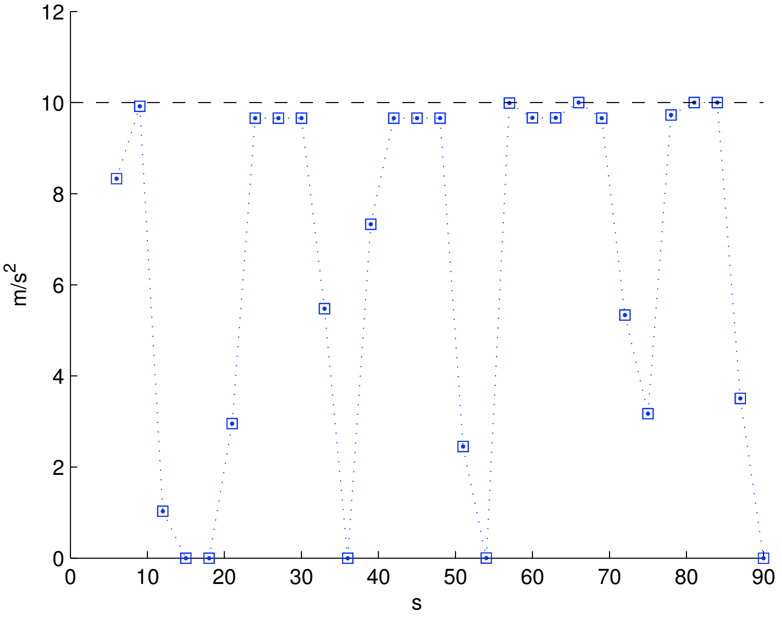

Figure 5.16: Acceleration of the UAV.

5.3.3 Comparison and discussion

Comparing the path planning results, presented in Figures 5.9 and 5.13, one can see that in both cases the waypoints of the UAV are chosen such that the cells of the corresponding sector(s) are visited during the flight. The main difference is that in the case of variable ice flow we have two sectors while in the case of constant ice flow we have one single sector. Furthermore, in the case of constant ice flow one can observe that the UAV circulates through the cells of the corresponding sector. However, in the case of variable ice flow the situation is slightly different; the visiting order of the cells varies with the ice flow direction. Moreover, in both cases, one can see that the UAV has the ability to make sharp turns. It is important to remember that in practice this will be possible for a rotary-wing UAV but not for a fixed-wing UAV. Consequently, for the case where fixed-wing UAV is used, the turning rate constraints should be included in the optimization problem. When it comes to the speed and acceleration of the UAV, one can see that these remain within the predefined limits in both cases; however, rapid changes in acceleration are observed. As already mentioned in Section 5.2.3, one possible solution to this problem is to add jerk constraints to the optimization problems. Regarding the coverage, presented in Figures 5.10 and 5.14, one can see that in both cases the blind zones are almost negligible in size. Comparing the solver-time, presented in Tables 5.9 and 5.12, once again one can observe that it requires significantly more time to solve the optimization problem in the case of variable ice flow compared to the case of constant ice flow.

5.4 Solver-time results

Surveillance area	Optimal J	Solver-time
Grid (Constant flow)	660.2	1260 s
Grid (Variable flow)	654.2	5865 s
Sector (Constant flow)	328.6	1487 s
Sector (Variable flow)	407	2395 s

Table 5.13: Solver-time and objective function value.

The solver-time results for the simulation cases in Sections 5.2 and 5.3 are summarized in Table 5.13. One can clearly see that it takes significantly more time to solve the optimization problems in the case of variable ice flow. Furthermore, during the simulation of the presented cases it was observed that the solver-time varies notably with the following factors:

- Number of linear constraints used for polygonal approximation of velocity and acceleration of the UAV. This parameter is also referred to as M_{vel} in Section 4.3.1.
- Number of cells within the SA (grid / sector). For the case where SA is defined as a sector, the width of the sector influences the solver-time.
- Tuning of the cost function for the case where the SA is defined as a sector.

When it comes to the number of linear constraints for polygonal approximation of velocity and acceleration, it was challenging to find an appropriate value for this parameter. Low value of the M_{vel} resulted in poor polygonal approximation, while high value of the M_{vel} resulted in poor solver-time. Consequently, it was made a compromise between polygonal approximation and the solver-time. Furthermore, the number of cells within SA (grid / sector) has a significant influence on the solver-time; the more cells, the higher is the solver-time. However, it is difficult to predict exactly how the solver-time will change when the number of cells changes; there is clearly no linear correspondence. For the case where SA is defined as a sector, it was observed that the wider the sector the higher is the solver-time; despite the fact that the number of cells within the sector remain unchanged. When it comes to the tuning of cost functions, it was observed that for the case of the sector shaped SA the tuning is critical for the solver-time. It is in fact the value of tuning parameter γ, in (4.25), that is of greatest importance. During the simulation, it was found that γ should be chosen in a range between 0 and 1. However, if γ is to close to 1 the solver-time becomes unacceptable. On the other hand, if γ is to close to 0 the path planning strategy fails. Consequently, one possible approach for performing appropriate tuning for γ, is to start with a value that is close to 0 and subsequently increase this value until appropriate flight path is achieved.

5.5 Summary

This chapter presented the simulation results for strategies proposed and formulated in Chapters 3 and 4. Each strategy was simulated for two cases; constant and variable ice flow. For each case, the following results were presented: flight path plot, SA coverage plot, speed plot, acceleration plot and the solver-value with the corresponding objective function value.

Chapter 6

Conclusion

Based on the presented flight path results it was concluded that the path planning has been successful for all simulation cases. Regarding the grid shaped SA, all of the cells and corresponding sub-cells are visited during the flight of the UAV, as expected. Moreover, the visiting order of the cells (and corresponding sub-cells) are not specified in advance. However, those parts of grid shaped SA where the ice is moving toward the drilling vessel are prioritized to be visited. When it comes to the sector shaped SA, the waypoints of the UAV are chosen such that continuous circulation through the cells of the corresponding sector is achieved, as expected. However, it was discovered that the UAV has the ability to make extremely sharp turns, almost on the spot. This is clearly a problem for the fixed-wing UAVs due to the limitations on the turning rate. Consequently, it was concluded that in the case where fixed-wing UAVs are used, the turning rate constraints should be included in the optimization problem. When it comes to the speed and acceleration of the UAV, it was observed that these remain within the predefined limits for all simulation cases; however, rapid changes in acceleration are observed in each simulation case. Hence it was concluded that jerk constraints should be included in the optimization problems.

When it comes to the coverage of the SA (grid / sector), it is important to remember that only a planar motion of the UAV is considered in this work; the operational altitude of the UAV is never taken into account. Additionally, the instrumentation of the UAV is beyond the scope of this work. Consequently, presented coverage of the SA (grid / sector) has only demonstrative character. Nevertheless, it was successfully demonstrated (for both grid and sector shaped SAs) that potentially achievable coverage is expected to be efficient. Moreover, it is worth noticing that assumed coverage radius of the UAV is drawn for each waypoint of the UAV, which represents significantly low (coverage) sampling frequency of the SA. In practice it is expected that this (coverage) sampling frequency will be much higher.

Regarding the solver-time, it was concluded that formulated optimization problems can not be used for real time path planning, and it is attractive to keep the complexity of the optimization problems as low as possible to reduce the solver-time. In all simulation cases presented in this work, the required solver-time is significantly longer compared to used optimization horizon. Moreover, there are several factors that can influence the solver-time; the optimality of the source code, computational challenges related to MILP and the optimality of the applied solver to name a few.

Recommendation for further work

The following list presents some suggestions for further work related to operational strategies for UAVs intended for information gathering in Arctic environments / ice-infested regions:

- **Motion in space**: Only a planar motion of the UAVs is considered in this work. It is therefore interesting to extend presented optimization problems to a 3D case, so that operational altitude of the UAVs can be taken into account. It is reasonable to expected that this will provide more realistic correspondence to practice.

- **Multiple UAVs**: In this work a single UAV is used in each simulation case. However, one can also imagine the situation where several cooperating UAVs are used instead; consequently, it is interesting to investigate potential advantages, disadvantages, possibilities and limitations of multiple cooperative UAVs.

- **Computational improvements**: The solver-time is an important issue. Consequently, it is attractive to compare different solvers and investigate their computational properties. Moreover, the optimality of the source code is in many cases a critical factor. Therefore, it is motivating to look at different methods that can be used for source code optimization. A more interesting idea is to design embedded computer system primary for solution of the path planning optimization problems for UAVs; in this case both the hardware and software can be optimized for solution of the particular class of optimization problems.

- **Communication infrastructure**: It is critical that the UAV stays within the range of the ground control station (GCS) for transmission of the gathered information; alternatively, the information must be stored for later delivery (ferrying). It is therefore attractive to propose / investigate operational strategies for UAV(s) where limitations related to communication infrastructure are taken into account, and possibly included in the path planning optimization.

Bibliography

Civil Aviation Authority - Norway (n.d.). N 25 Bruk av ubemannede luftfartøy i Norge, `http://www.luftfartstilsynet.no/regelverk/aic/n/article18535.ece`. Online; accessed January 19. 2011.

Culligan, K. F. (2006). *Online trajectory planning for UAVs using mixed integer linear programming*, Master's thesis, Massachusetts Institute of Technology.

Dalamagkidis, K., Valavanis, K. P. and Piegl, L. A. (2008). On unmanned aircraft system issues, challenges and operational restrictions preventing integration into the National Airspace System, *Progress in Aerospace Science* **44**: 503–519.

Edmond, C., Liferov, P. and Metge, M. (2011). Ice and iceberg management plans for Shtokman Field, *Offshore Technology Conference*.

Eik, K. (2008). Review of experiences within ice and iceberg management, *The Journal of Navigation* **61**(4): 557–571.

Fossen, T. I. (2010). *Guidance and Control of Marine Craft*, chapter 6, pp. 225–234.

Grøtli, E. I. and Johansen, T. A. (2011). Path planning for UAVs under communication constraints using SPLAT! and MILP, *Technical report*, Department of Engineering Cybernetics, Norwegian University of Science and Technology.

Gurobi Optimization (n.d.). Product overview, `http://www.gurobi.com/html/products.html`. Online; accessed February 7. 2011.

Haddon, D., Morier, Y., Lohl, N., Kneepkens, J., Magnier, V. and Goudou, P. (2009). Policy statement airworthiness certification of unmanned aircraft systems (UAS). EASA Doc: E.Y013-01.

Hobbs, A. (2010). Unmanned aircraft systems, *in* E. Salas and D. Maurino (eds), *Human Factors in Aviation*, Academic Press, pp. 505–531.

ICAO (2006). Convention on international civil aviation. Ninth edition, Doc: 7300/9.

JAA/EUROCONTROL (2004). UAV task-force: A concept for European regulations for civil unmanned aerial vehicles (UAVs), *Technical report*, Joint Aviation Authorities (JAA) and EUROCONTROL.

Jenssen, N. A., Muddesitti, S., Phillips, D. and Backstrom, K. (2009). Arctic, *Dynamic Positioning Conference*.

Johansen, K. S. (2008). *UAV - Ubemannede flygninger i lovtomt rom?*, Master's thesis, Universitet i Tromsø.

Kamal, W. A., Gu, D. W. and Postlethwaite, I. (2005). MILP and its application in flight path planning. Control and Instrumentation Group, Department of Engineering, University of Leicester.

Löfberg, J. (n.d.). Yalmip : A toolbox for modeling and optimization in matlab, `http://users.isy.liu.se/johanl/yalmip/pmwiki.php?n=Main.Download`. Downloaded software.

Magatão, L. (2005). *Mixed Integer Linear Programming and Constraint Logic Programming: Towards a Unified Modeling Framework*, PhD thesis, The Federal Center of Technological Education of Paraná.

Marshall, D. M. (2009). Regulation of UAS in Arctic airspace, *2009/2010 UAS Yearbook :* pp. 124–126.

Mettler, B., Schouwenaars, T., How, J., Paunicka, J. and Feron, E. (2003). Autonomous UAV guidance build-up: Flight-test demonstration and evaluation plan, *AIAA Guidance, Navigation and Control Conference*.

Podhradský, M. (2010). Modeling languages for optimization, *Technical report*, Faculty of Electrical Engineering, Czech Technical University.

Reinl, C. and von Stryk, O. (2007). Optimal control of multi-vehicle-systems under communication constraints using mixed-integer linear programming.

Richards, A. and How, J. P. (2002). Aircraft trajectory planning with collision avoidance using mixed integer linear programming.

Stalmakou, A. (2010). Use of UAVs for information gathering in ice management, *Technical report*, Department of Engineering Cybernetics, Norwegian University of Science and Technology.

YALMIP Wiki (n.d.). `http://users.isy.liu.se/johanl/yalmip/pmwiki.php?n=Main.HomePage`. Online; accessed April 17. 2011.

Appendix A

Source Code - Grid

run.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Used toolbox: YALMIP (Lofberg). %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

yalmip('clear');
clear all;
clc;

parameters;

variables;

model;

set_of_cells;

constraints;

%% Cost function %%

J_ACC       = 0;
J_TIME      = 0;

beta        = 1;        % Tuning parameter
epsilon     = 0.001;    % Tuning parameter

for k = 1:W
    for i = 1:N

            J_TIME = J_TIME + (beta*i*Td*b_k(k,1,i));

    end
end

for i = 1:(N-1)

   J_ACC = J_ACC + (epsilon*(w_s(1,1,i) + w_s(2,1,i)));

end

J_TOTAL = J_ACC + J_TIME;

%% SOLVE PROBLEM %%

diag = solvesdp(F,J_TOTAL,sdpsettings('solver','gurobi'));

%% Resulst %%

res.p       = double(p);
res.v       = double(v);
res.S       = double(S);
res.w_s     = double(w_s);
res.b_k     = double(b_k);
res.b_m     = double(b_m);

%% Draw %%

draw;

%% Save results %%

save('res.mat','res');
```

parameters.m

```
T       = 90;    % Horizon

Td      = 3;     % Sampling time

N       = (T - mod(T,Td))/Td;    % Steps

v_max   = 25;    % Max speed of the UAV

v_min   = 5;     % Min speed of the UAV

a_max   = 10;    % Max acceleration of the UAV
```

```

M_vel   = 8;    % Nr. of linear constr. speed/acceleration

L       = 45;   % Constant (arbitrary large)

M_big   = 500;  % Constant (arbitrary large)

W       = 16;   % Nr. of cells in surveillance area

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%% CONSTANT FLOW
flow    = zeros(2,1,W);

flow(:,1,1)  = [-0.3;-0.3]; % Flow within cell 1
flow(:,1,2)  = [-0.3;-0.3]; % Flow within cell 2
flow(:,1,3)  = [-0.3;-0.3]; % Flow within cell 3
flow(:,1,4)  = [-0.3;-0.3]; % Flow within cell 4

flow(:,1,5)  = [-0.3;-0.3]; % Flow within cell 5
flow(:,1,6)  = [-0.3;-0.3]; % Flow within cell 6
flow(:,1,7)  = [-0.3;-0.3]; % Flow within cell 7
flow(:,1,8)  = [-0.3;-0.3]; % Flow within cell 8

flow(:,1,9)  = [-0.3;-0.3]; % Flow within cell 9
flow(:,1,10) = [-0.3;-0.3]; % Flow within cell 10
flow(:,1,11) = [-0.3;-0.3]; % Flow within cell 11
flow(:,1,12) = [-0.3;-0.3]; % Flow within cell 12

flow(:,1,13) = [-0.3;-0.3]; % Flow within cell 13
flow(:,1,14) = [-0.3;-0.3]; % Flow within cell 14
flow(:,1,15) = [-0.3;-0.3]; % Flow within cell 15
flow(:,1,16) = [-0.3;-0.3]; % Flow within cell 16

%%%%%%% VARIABLE FLOW
% flow    = zeros(2,1,W);
%
% flow(:,1,1)  = [-0.3;-0.3]; % Flow within cell 1
% flow(:,1,2)  = [-0.4;-0.3]; % Flow within cell 2
% flow(:,1,3)  = [-0.5;-0.3]; % Flow within cell 3
% flow(:,1,4)  = [-0.6;-0.3]; % Flow within cell 4
%
% flow(:,1,5)  = [-0.3;-0.4]; % Flow within cell 5
% flow(:,1,6)  = [-0.4;-0.4]; % Flow within cell 6
% flow(:,1,7)  = [-0.5;-0.4]; % Flow within cell 7
% flow(:,1,8)  = [-0.6;-0.4]; % Flow within cell 8
%
% flow(:,1,9)  = [-0.3;-0.5]; % Flow within cell 9
% flow(:,1,10) = [-0.4;-0.5]; % Flow within cell 10
% flow(:,1,11) = [-0.5;-0.5]; % Flow within cell 11
% flow(:,1,12) = [-0.6;-0.5]; % Flow within cell 12
%
% flow(:,1,13) = [-0.3;-0.6]; % Flow within cell 13
% flow(:,1,14) = [-0.4;-0.6]; % Flow within cell 14
% flow(:,1,15) = [-0.5;-0.6]; % Flow within cell 15
% flow(:,1,16) = [-0.6;-0.6]; % Flow within cell 16

%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

alpha   = 1.01; % Constant (approximation accuracy)

d_c     = 50;   % d - cell

d_s_c   = 30;   % d - sub-cell
```

variables.m

```
p       = sdpvar(2,1,N);       % Position of the UAV

v       = sdpvar(2,1,N);       % Velocity of the UAV

S       = sdpvar(1,1,N);       % Speed of the UAV

w_s     = sdpvar(2,1,N-1);     % ABS(acceleration)

b_k     = binvar(W,1,N);       % 1 - cell visited
                               % 0 - cell not visited

b_m     = binvar(M_vel,1,N);   % 1 - S ≥ v_min
                               % 0 - S < v_min
```

model.m

```
Ad = [1 0;0 1];     % Discretized A matrix
Bd = [Td 0;0 Td];   % Discretized B matrix
```

set_of_cells.m

```
wp_set    = zeros(2,1,W);    % Set of cell-centers

wp_set(:,1,1)  = [100;100]; % Cell 1
wp_set(:,1,2)  = [200;100]; % Cell 2
wp_set(:,1,3)  = [300;100]; % Cell 3
wp_set(:,1,4)  = [400;100]; % Cell 4

wp_set(:,1,5)  = [100;200]; % Cell 5
wp_set(:,1,6)  = [200;200]; % Cell 6
wp_set(:,1,7)  = [300;200]; % Cell 7
wp_set(:,1,8)  = [400;200]; % Cell 8

wp_set(:,1,9)  = [100;300]; % Cell 9
wp_set(:,1,10) = [200;300]; % Cell 10
wp_set(:,1,11) = [300;300]; % Cell 11
wp_set(:,1,12) = [400;300]; % Cell 12

wp_set(:,1,13) = [100;400]; % Cell 13
wp_set(:,1,14) = [200;400]; % Cell 14
wp_set(:,1,15) = [300;400]; % Cell 15
wp_set(:,1,16) = [400;400]; % Cell 16
```

constraints.m

```
F = []; % Empty constraint matrix

%% Model constraints %%

F = F + [p(1,1,1) == 0,...      % Init x-position
         p(2,1,1) == 0,...      % Init y-position
         v(1,1,1) == 5.5,...    % Init v_x velocity
         v(2,1,1) == 5.5];      % Init v_y velocity

for i = 1:(N-1)

    F = F + [p(:,1,i+1) == (Ad*p(:,1,i) + Bd*v(:,1,i))];

end

disp('Done adding model constraints...');

%% Velocity / acceleration constraints %%
```

```

for m = 1:M_vel
    for i = 1:N

        if i>1
           F = F + [((v(1,1,i) - v(1,1,i-1))*  ...
                       sin((2*pi*m)/M_vel)) +  ...
                    ((v(2,1,i) - v(2,1,i-1))*  ...
                       cos((2*pi*m)/M_vel)) ≤  ...
                      a_max*cos(pi/M_vel)];
        end

        F = F + [((v(1,1,i)*sin((2*pi*m)/M_vel)) +   ...
                  (v(2,1,i)*cos((2*pi*m)/M_vel))) ≤  ...
                   S(1,1,i)*cos(pi/M_vel)];

        F = F + [((v(1,1,i)*sin((2*pi*m)/M_vel)) + ...
                  (v(2,1,i)*cos((2*pi*m)/M_vel)))* ...
                   alpha ≥ ...
                 ((S(1,1,i)*cos(pi/M_vel)) ...
                   - L*(1 - b_m(m,1,i)))];

    end
end

for i = 1:N

    F = F + [sum(b_m(:,1,i)) == 1];
    F = F + [v_min ≤ S(1,1,i) ≤ v_max];

end

disp('Done adding velocity/acceleration constraints...');

%% Cell and sub-cell constraints %%

counter  = zeros(W,1,N);

for k = 1:W

    ct = 1;

    for i = 1:N

        if ( (abs(flow(1,1,k))*Td*(ct - 1) ≥ d_c) || ...
             (abs(flow(2,1,k))*Td*(ct - 1) ≥ d_c) )
            ct = 1;
        end

```

```
            F = F + [(p(1,1,i) - (wp_set(1,1,k) + d_c)) ...
                ≤   (M_big*(1 - b_k(k,1,i)))];

            F = F + [(p(1,1,i) - (wp_set(1,1,k) - d_c)) ...
                ≥ - (M_big*(1 - b_k(k,1,i)))];

            F = F + [(p(2,1,i) - (wp_set(2,1,k) + d_c)) ...
                ≤   (M_big*(1 - b_k(k,1,i)))];

            F = F + [(p(2,1,i) - (wp_set(2,1,k) - d_c)) ...
                ≥ - (M_big*(1 - b_k(k,1,i)))];


            F = F + [(p(1,1,i) - (wp_set(1,1,k) + d_s_c ...
                   + (flow(1,1,k)*Td)*(ct - 1))) ≤...
                     (M_big*(1 - b_k(k,1,i)))];

            F = F + [(p(1,1,i) - (wp_set(1,1,k) - d_s_c ...
                   + (flow(1,1,k)*Td)*(ct - 1))) ≥...
                   - (M_big*(1 - b_k(k,1,i)))];

            F = F + [(p(2,1,i) - (wp_set(2,1,k) + d_s_c ...
                   + (flow(2,1,k)*Td)*(ct - 1))) ≤...
                     (M_big*(1 - b_k(k,1,i)))];

            F = F + [(p(2,1,i) - (wp_set(2,1,k) - d_s_c ...
                   + (flow(2,1,k)*Td)*(ct - 1))) ≥...
                   - (M_big*(1 - b_k(k,1,i)))];

            counter(k,1,i) = ct;

            ct = ct + 1;

    end

    F = F + [(sum(b_k(k,1,:))) == 1];

end

disp('Done with waypoints constraints...');

%% Region constraints %%

% NB! Not required.
% Used to speed up the simulation

F = F + [0 ≤ p(1,1,:) ≤ 450];    % NB! Not required
F = F + [0 ≤ p(2,1,:) ≤ 450];    % NB! Not required

```

```
disp('Done with region constraints...');

%% ABS(acceleration) constraints %%

for i = 2:N

    F = F + [ (v(1,1,i) − v(1,1,i−1)) ≤ w_s(1,1,i−1)];
    F = F + [−(v(1,1,i) − v(1,1,i−1)) ≤ w_s(1,1,i−1)];
    F = F + [ (v(2,1,i) − v(2,1,i−1)) ≤ w_s(2,1,i−1)];
    F = F + [−(v(2,1,i) − v(2,1,i−1)) ≤ w_s(2,1,i−1)];

end

disp('Done with ABS(acceleration)');
```

draw.m

```
figure(1);   %Trajectory plot

lineCrd = [];

hold on;

% UAV position
for i = 1:N
    if i == 1

        plot(res.p(1,1,i),res.p(2,1,i),'xr');

        lineCrd = [res.p(1,1,i) res.p(2,1,i)];

    else

        plot(res.p(1,1,i),res.p(2,1,i),'xr');

        line([lineCrd(1,1) res.p(1,1,i)],...
             [lineCrd(1,2) res.p(2,1,i)],...
             'LineStyle','−−','Color',   ...
             'green','Marker','o');

        lineCrd = [res.p(1,1,i) res.p(2,1,i)];

    end
end

% Position of the cell−centers
```

```
for k = 1:W

    plot(wp_set(1,1,k),wp_set(2,1,k),'ob');

    flow_arrow(wp_set(:,1,k),flow(:,1,k));

end

% Cells
cl   = 'black';

for k = 1:W

    line([(wp_set(1,1,k) + d_c) (wp_set(1,1,k) - d_c)],...
         [(wp_set(2,1,k) + d_c) (wp_set(2,1,k) + d_c)],...
          'Color',cl,'LineWidth',2);

    line([(wp_set(1,1,k) - d_c) (wp_set(1,1,k) - d_c)],...
         [(wp_set(2,1,k) + d_c) (wp_set(2,1,k) - d_c)],...
          'Color',cl,'LineWidth',2);

    line([(wp_set(1,1,k) + d_c) (wp_set(1,1,k) + d_c)],...
         [(wp_set(2,1,k) - d_c) (wp_set(2,1,k) + d_c)],...
          'Color',cl,'LineWidth',2);

    line([(wp_set(1,1,k) - d_c) (wp_set(1,1,k) + d_c)],...
         [(wp_set(2,1,k) - d_c) (wp_set(2,1,k) - d_c)],...
          'Color',cl,'LineWidth',2);

end

% Reached sub-cells
cl = 'red';

for k = 1:W
    for i = 1:N
        if (res.b_k(k,1,i) == 1)

            plot(wp_set(1,1,k) + (flow(1,1,k)*...
                 Td*counter(k,1,i)),...
                 wp_set(2,1,k) + (flow(2,1,k)*...
                 Td*counter(k,1,i)),...
                 '+','Color','red','LineWidth',1);

            line([(wp_set(1,1,k) + (flow(1,1,k)*...
                   Td*counter(k,1,i)) + d_s_c)   ...
                  (wp_set(1,1,k) + (flow(1,1,k)*...
                   Td*counter(k,1,i)) - d_s_c)],...
                 [(wp_set(2,1,k) + (flow(2,1,k)*...
                   Td*counter(k,1,i)) + d_s_c)  ...
                  (wp_set(2,1,k) + (flow(2,1,k)*...
                   Td*counter(k,1,i)) + d_s_c)],...
                  'LineStyle',':','Color',cl,   ...
                  'LineWidth',1);

            line([(wp_set(1,1,k) + (flow(1,1,k)*...
                   Td*counter(k,1,i)) - d_s_c)  ...
                  (wp_set(1,1,k) + (flow(1,1,k)*...
                   Td*counter(k,1,i)) - d_s_c)],...
                 [(wp_set(2,1,k) + (flow(2,1,k)*...
                   Td*counter(k,1,i)) + d_s_c)  ...
                  (wp_set(2,1,k) + (flow(2,1,k)*...
                   Td*counter(k,1,i)) - d_s_c)],...
                  'LineStyle',':','Color',cl,   ...
                  'LineWidth',1);

            line([(wp_set(1,1,k) + (flow(1,1,k)*...
                   Td*counter(k,1,i)) + d_s_c)  ...
                  (wp_set(1,1,k) + (flow(1,1,k)*...
                   Td*counter(k,1,i)) + d_s_c)],...
                 [(wp_set(2,1,k) + (flow(2,1,k)*...
                   Td*counter(k,1,i)) - d_s_c)  ...
                  (wp_set(2,1,k) + (flow(2,1,k)*...
                   Td*counter(k,1,i)) + d_s_c)],...
                  'LineStyle',':','Color',cl,   ...
                  'LineWidth',1);

            line([(wp_set(1,1,k) + (flow(1,1,k)*...
                   Td*counter(k,1,i)) - d_s_c)  ...
                  (wp_set(1,1,k) + (flow(1,1,k)*...
                   Td*counter(k,1,i)) + d_s_c)],...
                 [(wp_set(2,1,k) + (flow(2,1,k)*...
                   Td*counter(k,1,i)) - d_s_c)  ...
                  (wp_set(2,1,k) + (flow(2,1,k)*...
                   Td*counter(k,1,i)) - d_s_c)],...
                  'LineStyle',':','Color',cl,   ...
                  'LineWidth',1);

        end
    end
end

xlabel('m');ylabel('m');
axis square;
axis([0 500 0 500]);

hold off;

```

```
figure(2);   % Velocity / speed plots

lineCrdtotal_2 = [];

hold on;
for i = 1:N
    if i == 1

        plot(i*Td,res.S(1,1,i),'.');

        lineCrdtotal_2 = [i*Td res.S(1,1,i)];

    else

        plot(i*Td,res.S(1,1,i),'.');

        line([lineCrdtotal_2(1,1) i*Td],        ...
             [lineCrdtotal_2(1,2) res.S(1,1,i)],...
             'LineStyle',':','Color','b','Marker','s');

        lineCrdtotal_2 = [i*Td res.S(1,1,i)];

    end
end

line([0 N*Td],[v_max v_max], ...
     'LineStyle','--','Color','black');
line([0 N*Td],[v_min v_min], ...
     'LineStyle','--','Color','black');

xlabel('s');ylabel('m/s');
axis([0 N*Td 0 30]);

hold off;

figure(3);   % Acceleration plot

lineCrdtotal = [];

hold on;
for i = 2:N
    if i == 2

        plot(i*Td,sqrt((res.v(1,1,i) -      ...
                        res.v(1,1,i-1))^2 + ...
                       (res.v(2,1,i) -      ...
                        res.v(2,1,i-1))^2),'.');

        lineCrdtotal = [i*Td sqrt((res.v(1,1,i) -     ...
                                   res.v(1,1,i-1))^2 + ...
                                  (res.v(2,1,i) -      ...
                                   res.v(2,1,i-1))^2)];

    else

        plot(i*Td,sqrt((res.v(1,1,i) -      ...
                        res.v(1,1,i-1))^2 + ...
                       (res.v(2,1,i) -      ...
                        res.v(2,1,i-1))^2),'.');

        line([lineCrdtotal(1,1) i*Td], ...
             [lineCrdtotal(1,2)        ...
              sqrt((res.v(1,1,i) - res.v(1,1,i-1))^2 + ...
                   (res.v(2,1,i) - res.v(2,1,i-1))^2)],...
             'LineStyle',':','Color','b','Marker','s');

        lineCrdtotal = [i*Td sqrt((res.v(1,1,i) -      ...
                                   res.v(1,1,i-1))^2 + ...
                                  (res.v(2,1,i) -      ...
                                   res.v(2,1,i-1))^2)];

    end
end

line([0 N*Td],[a_max a_max], ...
     'LineStyle','--','Color','black');

xlabel('s');ylabel('m/s^2');
hold off;

figure(4);

hold on;

% Cells
cl  = 'black';

for k = 1:W

    line([(wp_set(1,1,k) + d_c) (wp_set(1,1,k) - d_c)],...
         [(wp_set(2,1,k) + d_c) (wp_set(2,1,k) + d_c)],...
          'Color',cl,'LineWidth',2);

    line([(wp_set(1,1,k) - d_c) (wp_set(1,1,k) - d_c)],...
         [(wp_set(2,1,k) + d_c) (wp_set(2,1,k) - d_c)],...
          'Color',cl,'LineWidth',2);

    line([(wp_set(1,1,k) + d_c) (wp_set(1,1,k) + d_c)],...
         [(wp_set(2,1,k) - d_c) (wp_set(2,1,k) + d_c)],...
          'Color',cl,'LineWidth',2);

    line([(wp_set(1,1,k) - d_c) (wp_set(1,1,k) + d_c)],...
         [(wp_set(2,1,k) - d_c) (wp_set(2,1,k) - d_c)],...
          'Color',cl,'LineWidth',2);

end

uav_coverage = 60;

% UAV coverage
for i = 1:N
    if i == 1

        plot(res.p(1,1,i),res.p(2,1,i),'xr');
        coverage([res.p(1,1,i),res.p(2,1,i)],...
                  uav_coverage);

        lineCrd = [res.p(1,1,i) res.p(2,1,i)];

    else

        plot(res.p(1,1,i),res.p(2,1,i),'xr');
        coverage([res.p(1,1,i),res.p(2,1,i)],...
                  uav_coverage);

        line([lineCrd(1,1) res.p(1,1,i)],...
             [lineCrd(1,2) res.p(2,1,i)],...
             'LineStyle','--','Color','green');

        lineCrd = [res.p(1,1,i) res.p(2,1,i)];

    end
end

xlabel('m');ylabel('m');
axis square;
axis([0 500 0 500]);
hold off;
```

flow_arrow.m

```
function [] = flow_arrow(wp,flow)

theta = atan2(flow(2),flow(1)) + (2*pi);
```

```
beta   = atan2(-flow(2),-flow(1)) + (2*pi);
phi    = (25/180)*pi;

r      = 70*sqrt((flow(1)^2) + (flow(2)^2));
d      = 10;

hold on;

line([wp(1) (wp(1) + (r*cos(theta)))],...
     [wp(2) (wp(2) + (r*sin(theta)))],'LineWidth',2);

line([(wp(1) + (r*cos(theta))) (wp(1) +     ...
      (r*cos(theta)) + (d*cos(beta+phi)))],...
     [(wp(2) + (r*sin(theta))) (wp(2) +     ...
      (r*sin(theta)) + (d*sin(beta+phi)))],'LineWidth',2);

line([(wp(1) + (r*cos(theta))) (wp(1) +     ...
      (r*cos(theta)) + (d*cos(beta-phi)))],...
     [(wp(2) + (r*sin(theta))) (wp(2) +     ...
      (r*sin(theta)) + (d*sin(beta-phi)))],'LineWidth',2);

end
```

coverage.m

```
function [] = coverage(c,r)

theta    = 1:1:360;

hold on;
for i = 1:length(theta)

   plot(c(1) + r*cos(theta(i)),c(2) + ...
               r*sin(theta(i)),'Color','r');

end

end
```

Appendix B

Source Code - Sector

run.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Used toolbox: YALMIP (Lofberg). %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

yalmip('clear');
clear all;
clc;

parameters;

variables;

model;

set_of_cells;

constraints;

%% Cost function %%

J_ACC      = 0;
J_THETA    = 0;
J_VC       = 0;

%%%%%%%% NB! Used in the case of CONSTANT flow %%%%%%%%%%%

gamma      = 0.15;    % Tuning parameter

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
%%%%%%%% NB! Used in the case of VARIABLE flow %%%%%%%%%%%

% gamma       = 0.17;     % Tuning parameter

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

epsilon    = 0.001;     % Tuning parameter

for k = 1:W
    for i = 1:N

        %%%%%%% NB! Used in the case of CONSTANT flow %%%%

            J_THETA = J_THETA + gamma*theta(k,1,i);

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        %%%%%%% NB! Used in the case of VARIABLE flow %%%%

%         if ((k ≤ 4) && (i ≤ ((N-mod(N,2))/2)))
%             J_THETA = J_THETA + gamma*theta(k,1,i);
%         end
%
%         if ((k > 4) && (i > ((N-mod(N,2))/2)))
%             J_THETA = J_THETA + gamma*theta(k,1,i);
%         end

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    end
end

for k = 1:W
    for i = 1:N

            J_VC = J_VC + (C_cost(1,1,k)*b_k(k,1,i));

    end
end

for i = 1:(N-1)

        J_ACC = J_ACC + epsilon*(w_s(1,1,i) + w_s(2,1,i));

end

J = J_ACC + J_THETA + J_VC;

%% SOLVE PROBLEM %%
```

```

diag = solvesdp(F,J,sdpsettings('solver','gurobi'));

%% Results %%

res.p           = double(p);
res.v           = double(v);
res.S           = double(S);
res.w_s         = double(w_s);
res.theta       = double(theta);
res.b_k         = double(b_k);
res.b_m         = double(b_m);

%% Draw %%

draw;

%% Save results %%

save('res.mat','res');
```

parameters.m

```
T        = 90;    % Horizon

Td       = 3;     % Sample time

N        = (T − mod(T,Td))/Td;   % Steps

R_a      = 500;   % UAV activation radius

v_max    = 25;    % Max speed of the UAV

v_min    = 5;     % Min speed of the UAV

a_max    = 10;    % Max acceleration of the UAV

M_pos    = 32;    % Nr. of linear constr. feasible reagion

M_vel    = 12;    % Nr. of linear constr. speed/acceleration

L        = 55;    % Constant (arbitrary large)

M_big    = 550;   % Constant (arbitrary large)

%%%%%%%% NB! Used in the case of CONSTANT flow %%%%%%%%%%%
```

```

W       = 4;    % Nr. of cells in sector/SA

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%% NB! Used in the case of VARIABLE flow %%%%%%%%%%%

% W       = 8;    % Nr. of cells in sector/SA

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%% Used for calculation of polar coordinates
%%%%%%% of the cells within the sector(s)

diagonal    = 450;

side_S      = 30;

side_L      = 45;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%% CONSTANT FLOW %%%%%%%

flow        = [−0.3;−0.3];

%%%%%%% VARIABLE FLOW %%%%%%%

% flow        = [[−0.4;−0.6],[−0.6;−0.4]];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

alpha   = 1.01; % Constant (approximation accuracy)

d_c     = 10;   % d − cell
```

variables.m

```
p         = sdpvar(2,1,N);       % Position of the UAV

v         = sdpvar(2,1,N);       % Velocity of the UAV

S         = sdpvar(1,1,N);       % Speed of the UAV

w_s       = sdpvar(2,1,N−1);     % ABS(acceleration)

theta     = sdpvar(W,1,N);       % Counter
```

```

b_k        = binvar(W,1,N);        % 1 - cell visited
                                   % 0 - cell not visited

b_m        = binvar(M_vel,1,N);    % 1 - S ≥ v_min
                                   % 0 - S ≤ v_min
```

model.m

```
Ad = [1 0;0 1];      % Discretized A matrix
Bd = [Td 0;0 Td];    % Discretized B matrix
```

set_of_cells.m

```
wp_set       = zeros(2,1,W);    % Set of cell-centers

C_cost       = zeros(1,1,W);    % Set of cell-costs

%%%%%%%% NB! Used in the case of CONSTANT flow %%%%%%%%%%%

theta_angle = atan((-flow(2,1))/(-flow(1,1))); % THETA

beta_angle  = atan(side_L/diagonal);           % BETA

alpha_angle = atan(side_S/diagonal); % ALAPHA ≤ BETA:
                                     % ensures that all of
                                     % the cells are
                                     % within the sector

%%%%%%% Lineset - (L1, L2 and L3 in Figure 3.9)

l_x_line    = diagonal*cos(theta_angle - alpha_angle);
m_x_line    = diagonal*cos(theta_angle);
u_x_line    = diagonal*cos(theta_angle + alpha_angle);

l_y_line    = diagonal*sin(theta_angle - alpha_angle);
m_y_line    = diagonal*sin(theta_angle);
u_y_line    = diagonal*sin(theta_angle + alpha_angle);

line_set    = [[l_x_line;l_y_line], ...
               [m_x_line;m_y_line], ...
               [u_x_line;u_y_line]];

```

```
wp_set(:,1,1)   = [(4/4)*line_set(1,1); ...
                   (4/4)*line_set(2,1)];

wp_set(:,1,2)   = [(1/4)*line_set(1,2); ...
                   (1/4)*line_set(2,2)];

wp_set(:,1,3)   = [(3/4)*line_set(1,2); ...
                   (3/4)*line_set(2,2)];

wp_set(:,1,4)   = [(4/4)*line_set(1,3); ...
                   (4/4)*line_set(2,3)];

C_cost(1,1,1)     = 10;
C_cost(1,1,2)     = 5;
C_cost(1,1,3)     = 5;
C_cost(1,1,4)     = 10;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%% NB! Used in the case of VARIABLE flow %%%%%%%%%%%%
%
% theta_angle = [atan((-flow(2,1))/ ... % THETA1 & THETA2
%                     (-flow(1,1))),...
%                atan((-flow(2,2))/ ...
%                     (-flow(1,2)))];
%
%
% alpha_angle = atan(side_S/diagonal);  % ---||---
%
% beta_angle  = atan(side_L/diagonal);  % BETA
%
% %%%%%%% Lineset - (L1, L2 and L3 in Figure 3.9)
% %%%%%%% for both sectors
%
% line_set = [];
%
% for j = 1:length(flow(1,:))
%
%     l_x_line    = diagonal*cos(theta_angle(1,j) - ...
%                                    alpha_angle);
%
%     m_x_line    = diagonal*cos(theta_angle(1,j));
%
%     u_x_line    = diagonal*cos(theta_angle(1,j) + ...
%                                    alpha_angle);
%
%     l_y_line    = diagonal*sin(theta_angle(1,j) - ...
%                                    alpha_angle);
%
```

```
%     m_y_line     = diagonal*sin(theta_angle(1,j));
%
%     u_y_line     = diagonal*sin(theta_angle(1,j) + ...
%                                    alpha_angle);
%
%     line_set = [line_set,[[l_x_line;l_y_line], ...
%                            [m_x_line;m_y_line], ...
%                            [u_x_line;u_y_line]]];
% end
%
% wp_set(:,1,1)    = [(4/4)*line_set(1,1); ...
%                     (4/4)*line_set(2,1)];
%
% wp_set(:,1,2)    = [(1/4)*line_set(1,2); ...
%                     (1/4)*line_set(2,2)];
%
% wp_set(:,1,3)    = [(3/4)*line_set(1,2); ...
%                     (3/4)*line_set(2,2)];
%
% wp_set(:,1,4)    = [(4/4)*line_set(1,3); ...
%                     (4/4)*line_set(2,3)];
%
% wp_set(:,1,5)    = [(4/4)*line_set(1,4); ...
%                     (4/4)*line_set(2,4)];
%
% wp_set(:,1,6)    = [(1/4)*line_set(1,5); ...
%                     (1/4)*line_set(2,5)];
%
% wp_set(:,1,7)    = [(3/4)*line_set(1,5); ...
%                     (3/4)*line_set(2,5)];
%
% wp_set(:,1,8)    = [(4/4)*line_set(1,6); ...
%                     (4/4)*line_set(2,6)];
%
% C_cost(1,1,1)     = 10;
% C_cost(1,1,2)     = 5;
% C_cost(1,1,3)     = 5;
% C_cost(1,1,4)     = 10;
% C_cost(1,1,5)     = 10;
% C_cost(1,1,6)     = 5;
% C_cost(1,1,7)     = 5;
% C_cost(1,1,8)     = 10;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

constraints.m

```
F = []; % Empty constraint matrix

%% Model constraints %%

v_init = velocity(flow(1,1), ... % Function used for
                  flow(2,1));    % calculation of the
                                 % init velocity for
                                 % the UAV

F = F + [p(1,1,1) == 0,...             % Init x-position
         p(2,1,1) == 0,...             % Init y-position
         v(1,1,1) == v_init(1,1),...   % Init v_x velocity
         v(2,1,1) == v_init(2,1)];     % Init v_y velocity

for k = 1:(N-1)

    F = F + [p(:,1,k+1) == (Ad*p(:,1,k) + Bd*v(:,1,k))];

end

disp('Done adding model constraints...');

%% Velocity / acceleration constraints %%

for m = 1:M_vel
    for i = 1:N

        if i>1
           F = F + [((v(1,1,i) - v(1,1,i-1))* ...
                       sin((2*pi*m)/M_vel)) + ...
                    ((v(2,1,i) - v(2,1,i-1))* ...
                       cos((2*pi*m)/M_vel)) ≤ ...
                        a_max*cos(pi/M_vel)];
        end

        F = F + [((v(1,1,i)*sin((2*pi*m)/M_vel)) +   ...
                  (v(2,1,i)*cos((2*pi*m)/M_vel))) ≤  ...
                   S(1,1,i)*cos(pi/M_vel)];

        F = F + [((v(1,1,i)*sin((2*pi*m)/M_vel)) + ...
                  (v(2,1,i)*cos((2*pi*m)/M_vel)))* ...
                   alpha ≥ ...
                 ((S(1,1,i)*cos(pi/M_vel)) ...
                   - L*(1 - b_m(m,1,i)))];

    end
end

for i = 1:N

    F = F + [sum(b_m(:,1,i)) == 1];
    F = F + [v_min ≤ S(1,1,i) ≤ v_max];

end

disp('Done adding velocity/acceleration constraints...');

%% Cell constraints %%

for k = 1:W

    for i = 1:N

            F = F + [(p(1,1,i) - (wp_set(1,1,k) + d_c)) ...
                ≤   (M_big*(1 - b_k(k,1,i)))];

            F = F + [(p(1,1,i) - (wp_set(1,1,k) - d_c)) ...
                ≥ - (M_big*(1 - b_k(k,1,i)))];

            F = F + [(p(2,1,i) - (wp_set(2,1,k) + d_c)) ...
                ≤   (M_big*(1 - b_k(k,1,i)))];

            F = F + [(p(2,1,i) - (wp_set(2,1,k) - d_c)) ...
                ≥ - (M_big*(1 - b_k(k,1,i)))];

            if i>1
                F = F + [theta(k,1,i) ≤         ...
                        (theta(k,1,i-1) + Td + ...
                        (Td*N*b_k(k,1,i)))];

                F = F + [theta(k,1,i) ≥         ...
                        (theta(k,1,i-1) + Td - ...
                        (Td*N*b_k(k,1,i)))];

                F = F + [theta(k,1,i) ≤         ...
                        (N*Td*(1 - b_k(k,1,i)))];

                F = F + [theta(k,1,i) ≥ -       ...
                        (N*Td*(1 - b_k(k,1,i)))];
            end

    end

    F = F + [theta(k,1,1) == Td];  % Init value of
```

```
                                   % the counter k

end

disp('Done with cell constraints...');

%% Sector constraints %%

for m = 1:M_pos
    for i = 1:N

        F = F + [((p(1,1,i)*sin((2*pi*m)/M_pos)) +    ...
                  (p(2,1,i)*cos((2*pi*m)/M_pos))) ≤   ...
                   R_a];

    end
end

%%%%%%% NB! Used in the case of CONSTANT flow %%%%%%%%%%%%

for i = 1:N

    F = F + [((sin(theta_angle - beta_angle)/            ...
               cos(theta_angle - beta_angle))*p(1,1,i)) ...
               ≤ p(2,1,i) ≤                              ...
             ((sin(theta_angle + beta_angle)/            ...
               cos(theta_angle + beta_angle))*p(1,1,i))];

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%% NB! Used in the case of VARIABLE flow %%%%%%%%%%%%

% for i = 1:N
%
%     F = F + [((sin(theta_angle(1,2) - beta_angle)/  ...
%                cos(theta_angle(1,2) - beta_angle))* ...
%                p(1,1,i)) ≤ p(2,1,i) ≤               ...
%              ((sin(theta_angle(1,1) + beta_angle)/  ...
%                cos(theta_angle(1,1) + beta_angle))* ...
%                p(1,1,i))];
%
% end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('Done with sector constraints...');

```

```
%% ABS(acceleration) constraints %%

for i = 2:N

    F = F + [ (v(1,1,i) − v(1,1,i−1)) ≤ w_s(1,1,i−1)];
    F = F + [−(v(1,1,i) − v(1,1,i−1)) ≤ w_s(1,1,i−1)];
    F = F + [ (v(2,1,i) − v(2,1,i−1)) ≤ w_s(2,1,i−1)];
    F = F + [−(v(2,1,i) − v(2,1,i−1)) ≤ w_s(2,1,i−1)];

end

disp('Done with acceleration constraints...');
```

draw.m

```
figure(1);    %Trajectory plot

lineCrd = [];

hold on;

% UAV position
for i = 1:N
    if i == 1

        plot(res.p(1,1,i),res.p(2,1,i),'xr');

        lineCrd = [res.p(1,1,i) res.p(2,1,i)];

    else

        plot(res.p(1,1,i),res.p(2,1,i),'xr');

        line([lineCrd(1,1) res.p(1,1,i)], ...
             [lineCrd(1,2) res.p(2,1,i)], ...
             'LineStyle','−−','Color',    ...
             'green','Marker','o');

        lineCrd = [res.p(1,1,i) res.p(2,1,i)];

    end
end


% Cells
cl   = 'black';
```

```

for k = 1:W

    line([(wp_set(1,1,k) + d_c) (wp_set(1,1,k) - d_c)], ...
         [(wp_set(2,1,k) + d_c) (wp_set(2,1,k) + d_c)], ...
          'LineStyle','--','Color',cl,'LineWidth',1);

    line([(wp_set(1,1,k) - d_c) (wp_set(1,1,k) - d_c)], ...
         [(wp_set(2,1,k) + d_c) (wp_set(2,1,k) - d_c)], ...
          'LineStyle','--','Color',cl,'LineWidth',1);

    line([(wp_set(1,1,k) + d_c) (wp_set(1,1,k) + d_c)], ...
         [(wp_set(2,1,k) - d_c) (wp_set(2,1,k) + d_c)], ...
          'LineStyle','--','Color',cl,'LineWidth',1);

    line([(wp_set(1,1,k) - d_c) (wp_set(1,1,k) + d_c)], ...
         [(wp_set(2,1,k) - d_c) (wp_set(2,1,k) - d_c)], ...
          'LineStyle','--','Color',cl,'LineWidth',1);

    plot(wp_set(1,1,k),wp_set(2,1,k),'+b');

end

for q = 1:length(flow(1,:))

% Sector
line([0 R_a*cos(theta_angle(1,q) + beta_angle)], ...
     [0 R_a*sin(theta_angle(1,q) + beta_angle)], ...
     'Marker','s');

line([0 R_a*cos(theta_angle(1,q) - beta_angle)], ...
     [0 R_a*sin(theta_angle(1,q) - beta_angle)], ...
      'Marker','s');

plot(R_a*cos(theta_angle(1,q) - beta_angle),     ...
     R_a*sin(theta_angle(1,q) - beta_angle),     ...
    'Marker','*','Color','red');

plot(R_a*cos(theta_angle(1,q) + beta_angle),     ...
     R_a*sin(theta_angle(1,q) + beta_angle),     ...
    'Marker','*','Color','red');

    % Flow direction
    flow_direction(R_a,theta_angle(1,q));

end

% Polygonal approximation of the activation
% radius for the UAV
```

```

polygon(M_pos,R_a);

xlabel('m');ylabel('m');
axis square;
axis([0 520 0 520]);

hold off;

figure(2);   % Speed plot

lineCrdtotal_2 = [];

hold on;
for i = 1:N
    if i == 1

        plot(i*Td,res.S(1,1,i),'.');

        lineCrdtotal_2 = [i*Td res.S(1,1,i)];

    else

        plot(i*Td,res.S(1,1,i),'.');

        line([lineCrdtotal_2(1,1) i*Td],             ...
             [lineCrdtotal_2(1,2) res.S(1,1,i)],     ...
             'LineStyle',':','Color','b','Marker','s');

        lineCrdtotal_2 = [i*Td res.S(1,1,i)];

    end
end

line([0 N*Td],[v_max v_max], ...
     'LineStyle','--','Color','black');
line([0 N*Td],[v_min v_min], ...
     'LineStyle','--','Color','black');

xlabel('s');ylabel('m/s');
axis([0 N*Td 0 30]);

hold off;

figure(3);   % Acceleration plot

lineCrdtotal = [];

hold on;
```

```
for i = 2:N
    if i == 2

        plot(i*Td,sqrt((res.v(1,1,i) -      ...
                        res.v(1,1,i-1))^2 + ...
                       (res.v(2,1,i) -      ...
                        res.v(2,1,i-1))^2),'.');

        lineCrdtotal = [i*Td sqrt((res.v(1,1,i) -      ...
                                   res.v(1,1,i-1))^2 + ...
                                  (res.v(2,1,i) -      ...
                                   res.v(2,1,i-1))^2)];

    else

        plot(i*Td,sqrt((res.v(1,1,i) -      ...
                        res.v(1,1,i-1))^2 + ...
                       (res.v(2,1,i) -      ...
                        res.v(2,1,i-1))^2),'.');

        line([lineCrdtotal(1,1) i*Td],                 ...
             [lineCrdtotal(1,2)                        ...
              sqrt((res.v(1,1,i) - res.v(1,1,i-1))^2 + ...
                   (res.v(2,1,i) - res.v(2,1,i-1))^2)],...
             'LineStyle',':','Color','b','Marker','s');

        lineCrdtotal = [i*Td sqrt((res.v(1,1,i) -      ...
                                   res.v(1,1,i-1))^2 + ...
                                  (res.v(2,1,i) -      ...
                                   res.v(2,1,i-1))^2)];

    end
end

line([0 N*Td],[a_max a_max], ...
     'LineStyle','--','Color','black');

xlabel('s');ylabel('m/s^2');
axis([0 N*Td 0 12]);

hold off;

figure(4); % Coverage plot

hold on;

uav_coverage = 40;

% UAV coverage
```

```
for i = 1:N
    if i == 1

        plot(res.p(1,1,i),res.p(2,1,i),'xr');
        coverage([res.p(1,1,i),res.p(2,1,i)], ...
                  uav_coverage);

        lineCrd = [res.p(1,1,i) res.p(2,1,i)];

    else

        plot(res.p(1,1,i),res.p(2,1,i),'xr');
        coverage([res.p(1,1,i),res.p(2,1,i)], ...
                  uav_coverage);

        line([lineCrd(1,1) res.p(1,1,i)], ...
             [lineCrd(1,2) res.p(2,1,i)], ...
             'LineStyle','--','Color','green');

        lineCrd = [res.p(1,1,i) res.p(2,1,i)];

    end
end

for q = 1:length(flow(1,:))

% Sector
line([0 R_a*cos(theta_angle(1,q) + beta_angle)], ...
     [0 R_a*sin(theta_angle(1,q) + beta_angle)], ...
     'Marker','s');

line([0 R_a*cos(theta_angle(1,q) - beta_angle)], ...
     [0 R_a*sin(theta_angle(1,q) - beta_angle)], ...
     'Marker','s');

plot(R_a*cos(theta_angle(1,q) - beta_angle), ...
     R_a*sin(theta_angle(1,q) - beta_angle), ...
    'Marker','*','Color','red');

plot(R_a*cos(theta_angle(1,q) + beta_angle), ...
     R_a*sin(theta_angle(1,q) + beta_angle), ...
    'Marker','*','Color','red');

    % Flow direction
    flow_direction(R_a,theta_angle(1,q));

end

% Polygonal approximation of the activation
% radius for the UAV

polygon(M_pos,R_a);

xlabel('m');ylabel('m');
axis square;
axis([0 520 0 520]);

hold off;
```

flow_direction.m

```
function [] = flow_direction(R_a, theta)

space     = 10;
length    = 50;
length_s  = 15;
color     = 'blue';
lineStyle = '-';
lineWidth = 2;

f_d(1)   = line([(R_a + space)*cos(theta)            ...
                 (R_a + space + length)*cos(theta)], ...
                [(R_a + space)*sin(theta)            ...
                 (R_a + space + length)*sin(theta)], ...
                'LineWidth',lineWidth,'Color',color, ...
                'LineStyle',lineStyle);

f_d(2)   = line([(R_a + space)*cos(theta)            ...
                 (R_a + space + length_s)            ...
                 * cos(theta + (1/180*pi))],         ...
                [(R_a + space)*sin(theta)            ...
                 (R_a + space + length_s)            ...
                 * sin(theta + (1/180*pi))],         ...
                'LineWidth',lineWidth,'Color',color, ...
                'LineStyle',lineStyle);

f_d(3)   = line([(R_a + space)*cos(theta)            ...
                 (R_a + space + length_s)            ...
                 * cos(theta - (1/180*pi))],         ...
                [(R_a + space)*sin(theta)            ...
                 (R_a + space + length_s)            ...
                 * sin(theta - (1/180*pi))],         ...
                'LineWidth',lineWidth,'Color',color, ...
                'LineStyle',lineStyle);

```

```
hold on;

for k = 1:3
   plot(f_d(k));
end

text((R_a + space + length + length_s)*cos(theta), ...
     (R_a + space + length)*sin(theta),            ...
     'FLOW','FontSize',8);

end
```

coverage.m

```
function [] = coverage(c,r)

theta   = 1:1:360;

hold on;
for i = 1:length(theta)

   plot(c(1) + r*cos(theta(i)),c(2) + ...
               r*sin(theta(i)),'Color','r');

end

end
```

polygon.m

```
function [] = polygon(M_vel,R_a)

theta      = 2*pi/M_vel;

d           = R_a*tan(theta/2);

alpha       = atan(d/R_a);

l           = sqrt((d^2) + (R_a^2));

hold on;

for m = 1:M_vel
```

```

    line([l*cos((theta*m) − alpha),  ...
          l*cos((theta*m) + alpha)], ...
         [l*sin((theta*m) − alpha),  ...
          l*sin((theta*m) + alpha)], ...
         'LineStyle',':','Color','blue','Marker','.');

end

plot(0,0,'+r');
axis square;

end
```

velocity.m

```
function v = velocity(f_x, f_y)

flow    = [f_x;f_y];
flow_o  = −flow;

v       = zeros(2,1);

counter = 0;
step    = 0.1;

while (norm(v) ≤ 5.5)

    v(1,1)  = counter;
    v(2,1)  = (flow_o(2,1)/flow_o(1,1))*counter;

    counter = counter + step;

end

end
```