Swarm of UAVs: Search & Rescue Operation in Chaotic Ship Wakes

April, 2014

Master Thesis
by
Ehsan Ebrahimi-Oskoei

Supervisors:
John R. Page (University of New South Wales)
Arne Karlsson (Royal Institute of Technology)

School of Engineering Sciences (SCI), Aeronautical and Vehicle Engineering
Royal Institute of Technology
100 44 STOCKHOLM
Abstract

A person falling of a ship can be difficult to locate since the wake behind the ship forms a chaotic field, making it extremely difficult to predict the location of the victim even if the time when they fell overboard is known. Survivability for humans immersed at sea is very dependent on the time spent in the water, and varies significantly with sea temperature; this makes it imperative that the victim is retrieved rapidly. Our current research is aimed at reducing this time using several UAV’s searching the ships wake simultaneously, as a swarm. Since the wake is chaotic, a simulation was developed to model different random motions of a victim based on a chaotic equation. Our current research is making use of an established simulator environment and developing it further to investigate how different platforms may affect rescue time, varying on the size of the ship, the weather conditions and whether the search is operated during day or night. Two different search strategies were implemented in the developed simulator; these are Expanding Square search and Parallel search. An overall conclusion based on the results obtained is that the expanding square search tends to be a more rigid and reliable search strategy. Also the results show that for any scenario, the sought person is detected within minutes.
This page intentionally left blank
PART I  Overview and Summary

1  INTRODUCTION ................................................................................................................ 1

1.1  Background and some basics on Unmanned Aerial Vehicles and Systems .......... 2

    Unmanned Aerial Vehicle (UAV) ................................................................................. 2

    UAV classification and configurations ................................................................. 3

    UAV swarm ........................................................................................................... 5

    Centralized control architecture ................................................................... 5

    Decentralized control architecture .............................................................. 6

    Applications of UAVs ............................................................................................ 7

1.2  Standard search strategies .......................................................................................... 9

    Parallel Track and Creeping line Search ............................................................. 9

    Expanding Square Search ................................................................................ 10

    Sector Search ...................................................................................................... 11

    Track line Search ................................................................................................ 11

    Contour Search .................................................................................................... 12

2  SUPPLEMENTARY CLARIFICATIONS FOR THE APPENDED PAPERS ............. 13

3  REFERENCES .................................................................................................................. 17

PART II  Papers

    Paper 1: UAV Swarm Search Strategy Applied To Chaotic Ship Wakes

    Paper 2: UAV Swarm Search and Rescue Simulator
This page intentionally left blank
PART I

Overview and Summary
1 \hspace{1em} \textbf{INTRODUCTION}

It is very problematic to find a person who has fallen overboard a ship. The difficulty of locating a person inherits from the chaotic nature of the wake region downstream of the ship stern, i.e. the wake behind the ship forms a chaotic field. This Chaotic field should not be misjudged as a region with purely stochastic nature; in fact chaotic systems are deterministic. However such systems are sensitive to initial conditions, making it extremely difficult to predict the location of the victim even if the time when they fell overboard is known.

One area where Unmanned Aerial Vehicles (UAVs) could find a significant application is for the search element of Search and Rescue (SAR) operations. These operations are often tedious, often operated under extreme conditions and when searching for a missing person the sensors have a higher success rate at low altitudes which is too risky for manned flight particularly over water.

In this thesis the utilization of a group of UAVs (UAV swarm) contribution to the search element of a SAR operation resulting from a person falling overboard into the chaotic wake region behind the ship is investigated. The reason for using a UAV swarm is that this greatly reduces the time to fully cover the significant wake region. Also as each UAV will be able to cover a specific area repeatedly over a short period of time, a more precise search is possible reducing the risk of failing to detect the person.

Several aspects need to be considered, such as speed and extent of the chaotic wake region (which is related to the wake source), UAV speeds, weather conditions (sensor resolution), arrival time and search strategies. Thus a MATLAB based simulator was developed in order to model the environment and generate the search element of different SAR operations taking these aspects into account. The appended paper “UAV Swarm Search and Rescue Simulator” provides a detailed description on the modelling and functioning of the developed simulator.

The search strategies implemented in the simulator are based on Parallel/creeping line search and Expanding square search, as these two are among the most common strategies for maritime SAR operations. The specific details of each of these two search strategies are provided in the appended papers “UAV Swarm Search Strategy Applied To Chaotic Ship Wakes” and “UAV Swarm Search and Rescue Simulator”.

The outline of the thesis is as follow:

\textbf{Part I}: Section one covers background and basics on Unmanned Aerial Vehicles and Systems. In section two a brief overview of standard search strategies recommended by the International Maritime Organization is also discussed (including the strategies implemented in the simulator).

\textbf{Part II}: Section one contains the Conference paper “UAV Swarm Search Strategy Applied To Chaotic Ship Wakes”, which was presented by the author at the 15th Australian International Aerospace Congress held in Melbourne, Australia. Section two consists of the paper “UAV Swarm Search and Rescue Simulator”, which functions as a handbook for the developed simulator.
1.1 Background and some basics on Unmanned Aerial Vehicles and Systems

In this section a brief overview of terminology and types of Unmanned Aerial Vehicles (UAVs), UAV swarms and possible applications are presented. The focus of the potential applications will be on missions intended for civilian usage.

Before proceeding, a UAV is merely one component/subsystem of a distributed system. Thus the correct term when describing the whole system is Unmanned Aerial System (UAS), which has been acknowledged by most organizations and among the scientific community during the last years.

**Unmanned Aerial System (UAS)**

An over simplistic view of an Unmanned Aerial System (UAS) may be regarded as a system that consists of three major components:

1. **Ground** - such as the Ground Control station (GCS), Ground Data Terminal (GDT). Ground Control Station being the most important of this subsystem.
2. **Communication** – such as control, command and Payload data link.
3. **Aerial** – the aerial platform (UAV) and the payloads for its purpose, such as sensors and cameras.

**Unmanned Aerial Vehicle (UAV)**

An Unmanned aerial vehicle (UAV) is an aircraft designed to operate without a human pilot onboard, the aircraft can be autonomous, semi-autonomous or controlled remotely by an operator from a control center (CS).

Usage of unmanned aerial vehicles can be dated as far back as to World War I. In 1917 Peter Cooper and Elmer A. Sperry (inventors of the gyroscopic compass) converted a U.S. Navy Curtiss N-9 aircraft into the first radio-controlled UAV [1].

A modern UAV is an unmanned aerial vehicle that is equipped with data processor units, sensors, automatic control, and communication systems and capable of performing autonomous flight missions without any interaction with a human pilot. A common deception is to confuse Unmanned Aerial Vehicles with “drones”. Although a drone is unmanned, it has no level of autonomously and can only be launched into a pre-programmed path and return to the base. Drones unlike UAVs do not have any means of communication and any result from a mission, such as surveillance photos are unattainable until it has returned back to the base.

The autonomy of a UAV may be described by a 10-point scale proposed by Parasuraman, Sheridan and Wickens [2] as illustrated in Table 1. The scaling is based on the assistance/workload the automated system requires from a human operator to decide and perform a certain task, i.e. the ability of the system in terms of decision making and authority. Modern UAVs have gained rapid development and growth during the last two-three decades, due to the technological advancements during this period. For example, powerful processor
units, lightweight materials, sophisticated navigation and communication capabilities have enabled development of advanced cost effective UAVs.

Table 1. Automation levels of Unmanned Aerial Systems (UAS)

<table>
<thead>
<tr>
<th>Level</th>
<th>Decisions/Actions of the Automated system</th>
<th>Authority</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Makes all decisions, human operator not informed.</td>
<td>Completely autonomous</td>
</tr>
<tr>
<td>9</td>
<td>Decides whether to inform the human operator of decisions and actions taken.</td>
<td>Completely autonomous</td>
</tr>
<tr>
<td>8</td>
<td>Informs the human operator about decisions and actions taken only if requested.</td>
<td>Completely autonomous</td>
</tr>
<tr>
<td>7</td>
<td>Executes automatically, but the human operator is continuously informed about decisions and actions taken.</td>
<td>Completely autonomous</td>
</tr>
<tr>
<td>6</td>
<td>Informs the human operator and executes automatically if the human operator does not disapprove within a restricted time.</td>
<td>Autonomous with consent</td>
</tr>
<tr>
<td>5</td>
<td>Selects actions, but does not execute until approved by the human operator.</td>
<td>Semi-Autonomous</td>
</tr>
<tr>
<td>4</td>
<td>Suggests one alternative action to the human operator and the human operator decides whether to reject or approve the suggested alternative</td>
<td>Advisory</td>
</tr>
<tr>
<td>3</td>
<td>Narrows down the number of alternative actions to a few from which the human operator decides which alternative is to be executed.</td>
<td>Advisory</td>
</tr>
<tr>
<td>2</td>
<td>Complete set of alternative actions is presented, from which the human operator decides which alternative is to be executed.</td>
<td>Advisory</td>
</tr>
<tr>
<td>1</td>
<td>None, The human operator has to make all decisions and actions.</td>
<td>None</td>
</tr>
</tbody>
</table>

**UAV classification and configurations**

There is currently a lack of universal classification standard for UAVs and a wide variety of organizations, research communities, governments and industries have employed different terms and scaling. To address this issue, the leading recommendation is that the UAV classifications can be emerged into the already established criteria’s and classifications that are set for manned aircraft. For a wider coverage of this subject the reader is referred to [3].

The wide varieties of classification of UAVs extend from grouping the aircrafts into different classes based on one or several of the following characteristics:

- Mass/Maximum take-off weight (MTOW)
- Operational altitude
- Range
- Endurance
- Speed
- The maximum kinetic energy, which is a function of the speed and weight of the aircraft and is essentially a classification based on impact damage.
An approach introduced by Roland Weibel and John Hansman [4] is to group the UAVs into five categories, two more categories have since been added and are marked with a (*) in Table 2. These categories are to collect UAVs of common size and operating characteristics into a specific group/class of UAVs as listed in Table 2. The operating features in Table 2 are compiled from several sources in order to provide an overall view of capabilities of the different UAV classes [4]-[5].

The reader is advised to consider the operational characteristics as typical values for UAVs of a certain category and not as absolute criteria’s. Especially in the field of smaller sized UAVs, where there is a continuous rapid improvement in operational characteristics. Mini UAVs (MUAVs) are by far the most established UAV type, according to UAS yearbook 2010-2011 [6] nearly 30 percent of existing UAV platforms are of this type.

Table 2. UAV classifications and corresponding operational characteristics

<table>
<thead>
<tr>
<th>UAV Category</th>
<th>Range (km)</th>
<th>Flight Altitude (m)</th>
<th>Endurance (hours)</th>
<th>MTOW (kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nano* (NAV)</td>
<td>&lt; 1</td>
<td>100</td>
<td>&lt; 1</td>
<td>&lt; 0.025</td>
</tr>
<tr>
<td>Micro (MAV)</td>
<td>&lt; 10</td>
<td>250</td>
<td>1</td>
<td>&lt; 5</td>
</tr>
<tr>
<td>Mini (MUAV)</td>
<td>&lt; 10</td>
<td>150-300</td>
<td>&lt; 2</td>
<td>&lt; 30</td>
</tr>
<tr>
<td>Close Range*</td>
<td>10-30</td>
<td>3000</td>
<td>2-4</td>
<td>150</td>
</tr>
<tr>
<td>Tactical (TUAV)</td>
<td>&lt; 200</td>
<td>300-5000</td>
<td>5-10</td>
<td>150-500</td>
</tr>
<tr>
<td>Medium Altitude-Long Endurance (MALE)</td>
<td>&gt; 500</td>
<td>5000-15 000</td>
<td>10-48</td>
<td>1500-7000</td>
</tr>
<tr>
<td>High Altitude-Long Endurance (HALE)</td>
<td>&gt; 20 000</td>
<td>&gt;15000</td>
<td>10-48</td>
<td>4500-15 000</td>
</tr>
</tbody>
</table>

There is a widespread misinterpretation regarding the distinction between NAV and MAV UAVs. It is illustrated in numerous papers where standard’s defined for a NAV is mistakenly selected to describe the criteria’s(typical characteristics) of a MAV. To authors knowledge the only set criteria’s for a NAV is done by the U.S Defense Advanced Research Projects Agency (DARPA). The main difference between these two categories is that in order to be classified as a NAV, a very strict size and mass limit must be followed. The wingspan of a NAV must be below 15 cm besides the differences in operational characteristics between the two categories displayed in Table 2.

**UAV configurations**

The conventional UAV platforms are normally classified into two main categories:

- Fixed-wing UAVs, also called horizontal take-off and landing (HTOL) UAVs.
- Rotary-wing UAVs, also called vertical take-off and landing (VTOL) UAVs.

The differences between these two categories of platforms can be regarded in terms of
Structural Complexity and Performance capabilities. Fixed-wing configurations have the advantage of requiring less structural complexity as well as efficient aerodynamics (low thrust-to-weight ratio). Such features enable better performance in terms of endurance, range and speed compared to rotary-wing configurations. The majority of current UAVs are based on fixed-wing configuration [7], due to the less complexity in design and the performance characteristics.

The rotary-wing configurations are on the other hand capable of vertical take-off and landing, hovering and providing a higher level of maneuverability. Platforms based on Rotary-wing configuration range from NAVs (Nano UAVs) up to TUAVs (Tactical UAVs). The absence of rotary-wing platforms in the two last UAV categories (MALE and HALE) is due to less efficient aerodynamics and rotor propulsion which limits the maximum operating altitude. There are also various unconventional UAV platform configurations such as, flapping-wing UAVs, Hybrid/Convertible UAVs, Lighter than air (LTA) UAVs.

**UAV swarm**

Progressions in technology (such as sensors, processing and communication units) has enabled smaller UAVs to accomplish the same functions as larger UAVs. However there are many scenarios in which a single UAV lacks in capability due to limitations (such as endurance, payload and long range communication bandwidth). An area of interest for many years has been to address multiple UAVs to perform missions rather than a single UAV. It has not been feasible in reality though until recently, mainly due to improvements in collision and Path finding [8], [9].

UAV swarm is in this context expressed as multiple UAVs collaborating in order to perform a certain task/mission, either controlled by operator(s) or autonomously. The multiple UAVs may be identical or differ in characteristics such as payload (sensors), level of automation or even platform configurations. For this reason, UAV swarms are generally classified as either Homogenous or Heterogeneous swarms in order to specify the similarities of the UAVs within the swarm.

Both approaches have advantages and disadvantages, a group of identical UAVs require a less complex system, but are limited in terms of payload, information processing and flexibility. A group of UAVs with different characteristics (heterogeneous swarm) requires a more complex system, while beneficial in circumstances where diversity of UAV characteristics is crucial. For an example, scenarios that require a higher level of sensor capability or several sensors that a single UAV cannot carry due to payload limitations [10], [11].

A comprehensive set of literature has been published (such as [11] - [13]) with a wide variety of approaches designed for controlling (path planning and task allocating) a multi-UAV system. These approaches are typically based on either centralized or decentralized control architectures [14], [15] Provides a detailed description on the differences between these control architectures.

**Centralized control architecture**

A centralized control approach permits a low level of autonomy of the UAV system, without any communication between each individual UAV. The operator(s) receives information from
each UAV and coordinates and prescribes task assignments to each individual UAV. This approach have the tendency of being simpler theoretically (low level autonomy) and easier to optimize, but less redundant and robust to failure of a UAV or communication.

**Decentralized control architecture**

A Decentralized (distributed) control approach requires a high level of autonomy of the UAV system and communication between each individual UAV. Each UAV must be able to communicate, share and receive information and make necessary decisions, thus shifts the role of the operator to a supervisory level.

Systems based on decentralized approach has not been utilized until the last years as it is far more complicated and requires a high level of autonomy. The significance of a decentralized approach is that the system becomes more robust, flexible and redundant, as the tasks and information is distributed on a UAV-to-UAV basis. Moreover, the benefits of the decentralized approach allow the UAVs to adapt to dynamic environments as the overall behavior of the system is constructed on local collaborations between the UAVs. The collaborative nature of the decentralized approach is thus less sensitive to a temporary or permanent failure of individual UAVs [16].

It should be noted that UAV swarm in its right denotation is merely based on decentralized control architecture; the term “multi-UAV” would be more suitable when describing both control strategies. Various control strategies based on decentralized approach has been developed, these are generally constructed on one or a combination [17], [18] of the following strategies:

- **Baseline strategy.** The simplest decentralized control method. The UAVs fly in a straight line until reaching a boundary of the search area, then turn to a certain degree to avoid leaving the search area.

- **Random strategy.** Essentially a slightly altered version of the baseline strategy, modified to allow the UAV to change its direction by a small amount at each time step.

- **Repulsion strategy.** This strategy is based on each UAV having a particular repulsion radius, i.e. the UAVs maneuver in a way to repel whenever within a given radius of another UAV at each time step.

- **Pheromone strategy.** Inspired from social insects. The UAVs leave a marker (digital, chemical, heat etc.) along its flight path as a trail to indicate which cells that has been visited (repulsive cells). The other UAVs are able to sense the pheromone level in their local surroundings and thus adjust their flight pattern by covering a cell that has not been visited.

**To summarize, the advantages of UAV swarms over single-UAVs are primarily:**

- **Distributed workload**
  - Larger area coverage during a set time (dividing the workload geometrically)
  - Less complex sensors
  - Quicker network setup independent of ground infrastructure
Increased network coverage (ad-hoc communication)

- **Coordinated workload**
  - Continuous coverage of an area (covering several locations at the same time)

- **Redundancy**
  - If an individual UAV is disabled others are still in function and can continue the mission.

**Applications of UAVs**

Nearly all countries currently have access to UAVs, out of which more than 52 countries are producing 1435 different types of UAVs for civil/commercial, military and research purposes [9]. The main argument for the use of UAVs over manned aircrafts is that these aircrafts can be operated in all environments and conditions without endangering human life (pilot). For example, at altitudes unapproachable or dangerous for a manned aircraft during an extended period of time. The scenarios where UAVs may be more suitable than manned aircrafts are commonly phrased as the “dull, dirty and dangerous” missions. This terms are adopted from a military point of view and could be extended to include covert, diplomatic, research, environmentally and economics [19].

UAVs have predominantly been used for military purposes, however as UAVs have become more affordable so has the interest into the potentials of UAVs for civilian purposes. This has led to a broad set of literature, addressing issues such as the absence of regulations, classification as well as potential civilian applications [20]-[22]. In Figure 1 some of the potential applications are displayed in order to present the broad-spectrum of civilian UAV applications.

The significance of UAVs has already been demonstrated in a variety of civilian missions “actual situation”. For example, searching for trapped survivors during hurricane Katrina in 2005 [23]. Assisting firefighters by identifying location and direction of fires during the California rim fire in 2013 [24]. Measuring radiation levels of the Fukushima Daiichi nuclear power plant [25], [26]. United Nations (UN) launched its first ever UAV to monitor armed groups in Democratic Republic of Congo since 3 Dec 2013 [27], [28]. Organizations and research centers are also conducting UAV competitions as to further boost the development (crowd sourcing) as well as promoting the utility of UAVs for civilian applications.

Examples of such competitions are:

- **The UAV Challenge - Outback Rescue** [29]. Annual competition organized by Australian Research Centre for Aerospace Automation (ARCAA), CSIRO (Commonwealth Scientific and Industrial Research Organization) and Queensland University of Technology


- **AUVSI** (Association for Unmanned Vehicle System International) [31]. AUVSI organizes several competitions annually.
- **NASA Centennial Challenges** [32]. Annual competition organized by National Aeronautics and Space Administration (NASA)

To this date, UAV applications in “actual situations” have been restricted to single-UAVs and most UAV swarm research is conducted by universities and research institutes. The experiments and projects done by researchers are to replicate the “actual situations” either in a smaller scale or more commonly in simulators. The ideal scenarios for application of UAV swarms are however the same as for single-UAVs, above all when a broader area has to be covered during a short period of time. For example search and rescue missions, forest/bushfires and communication setup after a disaster.

---

**Figure 1.** *Source: Unmanned Aircraft Systems for Civilian Missions [20].*
1.2 **Standard search strategies**

Several search strategies have been recognized and set as standards by the International Maritime Organization and International Civil Aviation Organization. These are published annually in IAMSAR (International Aeronautical and Maritime Search and Rescue) manual [33]. These are designed to provide simple and effective visual search patterns that can be applied in various situations.

The standard search patterns are as follow:

- Parallel Track and Creeping line Search
- Expanding Square Search
- Sector Search
- Track Line Search
- Contour Search

The main characteristics of these search patterns and conditions under which they are being utilized are discussed below.

**Parallel Track and Creeping line Search**

Both parallel track and creeping line search are based on covering (sweeping) the search area by maintaining parallel tracks as shown in Figure 2. The only difference between these two is in the orientation of the search legs, i.e. if the search legs are parallel to the long sides (major axis) or short sides (minor axis) of the designated search area.

Parallel track or creeping line search strategy are generally used when one or a combination of the following conditions exists:

- The search area is large.
- Uniform coverage is required, i.e. the search area has to be divided into subareas for assigning individual search units to search for the target (sought person) simultaneously.
- Search is conducted over water or flat terrain.
- The location of the target is uncertain, i.e. equal probability of the target being anywhere in the search area.
Expanding Square Search

This pattern begins at the center of the designated search area and expands outward in concentric squares as shown in Figure 3.

Expanding Square search strategy is generally used when one or a combination of the following conditions exists:

- The search area or search section is relatively small.
- Uniform coverage is required, i.e. search area has to be divided into subareas for assigning individual search units to search for the target simultaneously.
- Search is conducted over water.
- Most effective when the location of the target is known/predicted within close limits, i.e. a more concentrated search strategy.
**Sector Search**

The pattern employed in this search strategy is shown in Figure 4; essentially the idea in this search pattern is to cover the entire search area from different angles so terrain and lighting problems can be minimized.

Sector search strategy is generally used when one or a combination of the following conditions exists:

- The search area is small and circular.
- Most effective when the location of the target is highly accurate (such as when the target has been sighted and then lost).

![Figure 4. Sector Search pattern.](image)

**Track line Search**

The pattern of this search strategy is to search along the projected track line (route) of a distressed craft (ship or aircraft) as shown in Figure 5. This strategy is normally used when an aircraft or ship has disappeared without a trace and it is believed that the target is in close surroundings of the crafts known route.

![Figure 5. Track line search pattern.](image)
**Contour Search**

The search pattern of this search strategy may be described as a downward spiral motion as shown in Figure 6. This search strategy is applied around mountains or valleys with sharp changes in elevation as the other search patterns are not practical in these circumstances.

Figure 6. Contour Search pattern. *Source: IAMSAR Manual 2010 [33]*
2 SUPPLEMENTARY CLARIFICATIONS FOR THE APPENDED PAPERS

In this section some supplementary material and clarifications to the two appended papers “UAV Swarm Search Strategy Applied To Chaotic Ship Wakes” and “UAV Swarm Search and Rescue Simulator” are given.

Constructing a model based on the ship parameters

Deterministic chaos is found in many dynamic systems of different nature, an excellent published book discussing a variety of these systems is “Evolutionary Algorithms and Chaotic Systems” [34]. The approach in finding a suitable chaotic equation (Equation (1)) for modelling the ship wake was to investigate different bifurcation diagrams in [34]. The equation chosen is based on the bifurcation diagram that best resembles the nature of a ship wake [35], [36]. The control parameter in our setup is in the longitudinal direction ($x$-direction) downstream of the stern of the ship and the corresponding growth is in the lateral direction ($y$-direction) with respect to the midsection of the stern.

In order to model the wake region using Equation (1), it is necessary to have the initial values of the control parameter $x$ and the lateral position $y$ as well as the range of the control parameter. It was accomplished by setting the initial values and range of the control parameter to correspond to the same values depicted in the selected bifurcation diagram [35]. The parameters $x_{\text{norm}}$ and $y_{\text{norm}}$ in Equation (1) are based on these initial values and range of the control parameter in order to obtain a normalized wake region that is independent of the ship parameters. Once these values are known, the following $y$-position along the longitudinal axis is obtained as shown in Equation (1), where index $i$ indicates current position.

$$y_{\text{norm}}(i + 1) = \frac{(x_{\text{norm}}(i)y_{\text{norm}}(i) - y_{\text{norm}}(i)) \cdot (x_{\text{norm}}(i)y_{\text{norm}}(i) - (2x_{\text{norm}}(i) + y_{\text{norm}}(i)))}{x_{\text{norm}}(i) + y_{\text{norm}}(i)} + y_{\text{norm}}(i)^2$$ (1)

It can be seen that in order to create a complete region, the increment between each step along the longitudinal axis has to be extremely small. The reason is that only one possible $y$-position can be obtained at each step, thus the step has to be small enough to create a large number of data points in the lateral direction. The extremes of these data’s are then used in order to establish the outer boundaries (lateral extent) of the search area at each instance along the longitudinal axis.

Furthermore the range of the control parameter corresponds to the significant part of the wake region that can be considered chaotic, which is 20 ship lengths [36]. As a result this procedure enables establishing the search area (wake region) for ships of different lengths, since the growth of the wake in $y$-direction is obtained from the control parameter.
The track pattern of the person in the wake region

The definition of the block coefficient $C_B$ is the ratio of volume displacement of the ship to the volume of a rectangular block having length $L$, width $B$ and height $T$. Where $L = LBP$ (acronym for Length Between Perpendiculars which is a measure of the length of the ship that is in contact with the water), $B$ the breadth of the hull and $T$ the draught of the ship.

Numerous estimate methods exist in order to evaluate the approximate value of the block coefficient; these are normally based on statistical regressions with information composed from existing ships. An estimate method proposed by Barrass B. relates the value of $C_B$ by a direct link between the service speed $v_s$ and $LBP$ [37]. This latter relation was used in calculation of the speed of the ship wake, which is a function of the service speed and the block coefficient.

By combining these relations, the speed of the ship wake can be expressed as a function of the service speed $v_s$ and $LBP$ [38] as shown in Equation (2). However the speed of the ship wake is replaced with the speed of the sought person $v_{person}$, which is assumed to be the same as the wake speed.

\[
C_B = 1.2 - 0.39 \left( \frac{v_s}{\sqrt{LBP}} \right) \quad \rightarrow \quad v_{person} = \left( \frac{C_B}{2} - 0.05 \right) \cdot v_s
\]

$v_s$ - Service speed [knots]

$v_{person}$ - Speed of the person [knots]

$LBP$ - Length between perpendiculars [m]

Arrival to the scene (Results)

The arrival of the UAV swarm to the scene is largely dependent on the circumstances (distance, availability etc.) and the platforms used. For example a number of small UAVs may be deployed from larger UAVs, manned aircraft and ships, thus depending on the distance different procedures in deployment will largely affect the arrival time. The approach was to imbed diverse deployment and arrivals to the scene by requiring user input on estimated arrival time and investigating to what extent it will affect the choice of search strategy and success rate.

The specified arrival time enables to set the initial conditions of the search in the simulator environment as it indicates the initial position of the sought person upon the arrival of the UAVs. Thus the time parameter of significance is the elapsed time of the search, hence the parameter considered in the simulator in establishing if detection of the person has occurred.
Final comments and summary

This chapter has covered specific parts of the appended papers in which the author believes additional explanations may be useful. Such as, the approach and details of the chaotic equation used in the modelling of the ship wake based on ship parameters. The method used in estimating the sought persons speed and track pattern as well as the definition of the ship parameters $c_b$ and $LBP$ is also described. The approach taken in the arrival of the UAVs to the scene and how different deployment and arrival times are imbedded in the simulator environment is also discussed briefly. However these are merely supplementary clarifications and a more detailed description of the overall methodology is available in “UAV Swarm Search Strategy Applied To Chaotic Ship Wakes”. The obtained results and discussions are also found in the paper “UAV Swarm Search Strategy Applied To Chaotic Ship Wakes”. The process and implementation of the suggested methods is discussed in detail in the paper “UAV Swarm Search and Rescue Simulator”, which is a handbook for the developed simulator.
Acknowledgements

I would like to express my deep gratitude to Mr. John Randall Page, my research supervisor at University of New South Wales, for his guidance, enthusiastic encouragement and support throughout the research work. I would also like to extend my thanks to the fellow students at the simulation laboratory for much appreciated feedback and valuable discussions.

I would also like to acknowledge Mr. Arne Karlsson, my research supervisor at Royal Institute of Technology, for his guidance and willingness to help despite being overwhelmed with additional work.

Finally, I wish to thank my parents for their support and encouragement throughout my study.
3 REFERENCES


[36] Mr. John Randall Page, “personal communication”.


This page intentionally left blank
PART II

Papers
Paper 1

UAV Swarm Search Strategy Applied To Chaotic Ship Wakes

Conference paper presented by the author at the 15th Australian International Aerospace Congress, Melbourne, Australia. 25 - 28 February 2013
This page intentionally left blank
UAV Swarm Search Strategy Applied To Chaotic Ship Wakes

Ehsan Ebrahimi ¹, John Page ²

¹ University of New South Wales
School of Mechanical & Manufacturing Engineering

Abstract
A person falling of a ship can be difficult to locate since the wake behind the ship forms a chaotic field, making it extremely difficult to predict the location of the victim even if the time when they fell overboard is known. Survivability for humans immersed at sea is very dependent on the time spent in the water, and varies significantly with sea temperature, this makes it imperative that the victim is retrieved rapidly. Our current research is aimed at reducing this time using several UAV’s searching the ships wake simultaneously, as a swarm. Since the wake is chaotic, a simulation was developed to model different random motions of a victim based on a chaotic equation. Our current research is making use of an established simulator environment and developing it further to investigate how different platforms may affect rescue time, varying on the size of the ship, the weather conditions and whether the search is operated during day or night.

Keywords: UAV, Swarm, Chaotic Field, Simulator, Ship Wake, Search Strategy, Search and Rescue, Deterministic.

Introduction
The use of Unmanned Aerial Vehicles is most known for military purposes, due to the fact that they can operate under extreme conditions without risking any casualties, but it is being used in more and more non-military purposes. As the technology advances and the reliability of these unmanned vehicles grows their demand growth is resulting in them being cheaper to produce and operate.
One area where Unmanned Aerial Vehicles (UAV’s) could find a significant application is for the search element of Search and Rescue (SAR) operations. These operations are often tedious, often operated under extreme conditions and when searching for a missing person the sensors have a higher success rate at low altitudes which is too risky for manned flight particularly over water. In this paper it is proposed to investigate the utilization of a group of UAVs (UAV swarm) contribution to a search and rescue operation resulting from a person falling overboard into the chaotic wake region behind the ship. The reasons for using a UAV swarm is that this greatly reduce the time to fully cover the significant wake region. Also as each UAV will be able to cover a specific area repeatedly over a short period of time, a more precise search is possible reducing the risk of failing to detect the person.

SIMULATOR
The simulator that has been developed is a MATLAB based tool. It can be divided into two sections, where section one consisted of constructing a model based on the ship parameters and UAV speeds, while in section two it simulates a Search and Rescue operation applied to the previous generated model.
Constructing a model based on the ship parameters

The simulated environment was created using a chaotic equation to model the wake behind the ship. Within this region a large number of discreet points were setup for the UAV’s to follow for different search patterns always ensuring they remained within the wake region. This is possible since the equation defines the upper and lower boundaries of the wake, which is then used by the simulator to limit the search area of individual UAVs. A major advantage of the large number of discreet points is that they can be easily generated, so simulation is applicable for large or small ships. The user inputs the size of the ship then the simulator calculates the size of the wake and the number of discreet points each UAV will pass during each step as well as determining the ideal number of UAV’s and their search pattern. The first step in building the model was to apply a suitable chaotic equation [1]. Eqn 1 was applied to form a normalized wake region that is applicable for all ship sizes.

\[ x_{\text{norm}}(i) = \text{current normalized } x\text{-position} \]
\[ y_{\text{norm}}(i) = \text{current normalized } y\text{-position} \]

\[ y_{\text{norm}}(i + 1) = \frac{- (x_{\text{norm}}(i) y_{\text{norm}}(i) + y_{\text{norm}}(i)) \left( x_{\text{norm}}(i) y_{\text{norm}}(i) - 2 x_{\text{norm}}(i) y_{\text{norm}}(i) \right)}{x_{\text{norm}}(i)^2 + y_{\text{norm}}(i)^2 + y_{\text{norm}}(i)^2} \]  \hspace{1cm} (1)

Once this is completed, the simulator requests user input Fig. 1 about the ship speed and length as well as the velocity of the UAVs that will be engaged in the Search and Rescue operation.

![User input dialog box](image)

**Fig. 1** User input dialog box

The significant part of a ship wake that can be considered as chaotic is roughly 20 ship lengths, even though it might vary slightly depending on the speed and shape of the ship. This area is significant even when the victim has washed out of the wake, as the initial point for a problematic search is determined by behavior in this region though this is beyond the scope of this paper. Knowing the ship length and the speed of the UAVs and using 20 ship lengths as a reference for the horizontal range of the wake, allows the simulator to mesh the wake region into a size that is equal to the distance the UAVs can fly per time step, this is currently set to 0.01 s Eqn 2.

\[ v_{\text{uav, step}} = v_{\text{uav}} \cdot \text{step time} \]
\[ L_w = 20 \cdot LBP \]
\[ \text{mesh}_x = \frac{L_w}{v_{\text{uav, step}}} \]
\[ \text{mesh size} = \frac{\text{range } x_{\text{norm}}}{\text{mesh}_x} \]  \hspace{1cm} (2)
where \( \text{mesh}_x \) and \( \text{mesh}_y \) is the mesh co-ordinates in the \( x \) and \( y \) direction. The program then models an upper and lower boundary in the \( y \)-direction for each \( x \) coordinate. These boundaries are later applied to the control algorithm to ensure that the UAVs never search outside of the wake region.

**The pattern of the person in the wake region**

The predicted movement of the person in the wake region is also based on the chaotic equation. The simulator randomly selects some points, then goes on to create new points between each point based on the expected velocity of the person, which is assumed to be the same speed as the wake. The new points between each previous point are allowed to change direction in both \( x \) and \( y \)-direction with respect to the speed of the victim. It is assumed the victim can move in any direction during a time step, this is to simulate a real environment where currents, winds and waves will make the motion of the victim move back and forth in the \( x \) and \( y \)-direction. When the motion of the victim has been established Fig. 2, the program will use the previous (\( x, y \)) co-ordinates and the new (\( x, y \)) co-ordinates to calculate the distance and time between the two points as well as the time elapsed from the time the victim fell overboard. This procedure gives more continuity to the motion since it is based on the step time and gives the opportunity to synchronize with the time of the UAV by using the same step time.

![Pattern of the sought person in a ship wake of 6000 m.](image)

**Fig. 2 Pattern of the sought person in a ship wake of 6000 m.**

As mentioned earlier, the speed of the victim is assumed to be the same as the velocity of the ship wake, knowing the service speed as well as the length between perpendiculaires LBP of the ship (the length of the ship that is in contact with the water), can be calculated from Eqn 3.

\[
v_w = \left( \frac{C_B}{2} - 0.05 \right) \cdot v_s
\]

\[
C_B = 1.2 - 0.39 \left( \frac{v_s}{\sqrt{\text{LBP}}} \right)
\]

\( C_B = \) the block coefficient of the ship.
Search and rescue (SAR) operation

There are several common factors that are all linked and have to be considered for a successful SAR operation, such as the time until arrival at the scene, the UAV speed, altitude and endurance and the sensors ability to detect the victim under the prevalent conditions. The sensor’s ability to detect a victim in the prevalent conditions will significantly affect the maximum altitude at which the UAVs can operate, since the sensors ability to detect a person immersed in water is very dependent on line of sight. Flying at a higher altitude will result in a greater area of sensor cover, but as consequence reduce the probability of detection.

![Sensor coverage](image)

In this paper two different search strategies will be investigated within a chaotic ship wake. The strategies selected, as these are the most commonly used currently, are expanding square search with intersecting sections and a parallel search, these two strategies are presented below. The detection algorithm that is implemented in the simulator lowers the sensor range coverage that is set by a value based on the sought persons velocity, this reduction in range is based on the distance the person can move in any direction within 0.1 s. The reason for this range reduction is to enable a time interval in the detection algorithm since if the person is within this new sensor range, any movement during a time interval of 0.1 s will still be in the range of the set sensor range.

**Expanding square search with intersecting sections.**

This strategy is based on each UAV following an expanding square pattern as shown in Fig. 4. The sections that each UAV will cover are time dependent i.e. the area that the UAV will cover is broader, where the sought person is less likely to be, such as in the beginning of the ship wake. The justification for this approach is due to the fact that until the UAVs have arrived to the scene the sought person will have moved some distance away from the stern of the ship. While this area cannot be neglected owing to the complexity of a chaotic region and the difficulty to estimate the position of the sought person, it is still a reasonable assumption that this region is of low priority, if the sought person is in this area and hasn’t been detected from the ship. Other assumptions can also be used to improve the efficiency of the search. The region further away from the origin of the wake is more critical to finding the victim in a good condition. Hence the search sections of the UAVs will become narrower and allow coverage of these critical areas repeatedly over a short time period to ensure that the emerged person is detected and rescued as quickly as possible.
The UAV swarms behavior will be such that each UAV will intersect the adjacent UAVs section by flying to some extent into that region, to ensure a full coverage of the wake area as shown in Fig. 5. In order to avoid the risk of collision, the adjacent UAVs will fly at a slightly different altitude, without losing any major accuracy in sensor resolution. Altitude change can in fact be a strategy by itself, where the low priority regions can be covered quicker by having a higher altitude but with less accuracy, hence releasing the UAV quicker to join the search in the more critical regions of the wake.

Parallel search

This strategy is based on a parallel “sweeping” motion as shown in Fig. 6 where the black arrow represents the forward motion and red arrow the backward motion when the UAV has reached the end of its section. The wake region is divided into a lower and upper region and the UAVs will cover a specific section in either the lower region or the upper without any intersecting between the adjacent sections and all UAVs flying at the same altitude and velocity. The idea behind this strategy is to quickly cover the section from side to side but it is less precise. Especially for the cases when the wake length is large and hence each UAV has a larger area to cover thus the time it takes for the UAV to revisit a previous point will be longer. This increases the risk of missing detection of the sought person if the person was just outside of the sensor range in the previous run. The purpose of this strategy is to determine whether if a group of UAVs carrying out a Search mission completely independent of each of the other UAVs will have a major effect of the detection time.
Results

The results presented are based on a number of simulations where a group of 10 UAVs performed a simulated search and rescue for different lengths of a ship wake. Then varying the time taken for arrival to the wake region is also considered in order to investigate the elapsed time from arrival to detection of the sought person. It should also be noted that the time for arrival is chosen to be within the time interval that the sought person remains within the significant part of the wake region, which is set to 20 ship lengths. The sought persons movement will still be affected by the chaotic wake outside the set boundary of 20 ship lengths, but in order to reduce the computation requirement on the simulator, this upper boundary has been set. This problem can be solved, in later work, outside the wake region by implementing a side shift section change after a specific set time interval. As time passes the search area will shift downstream of the wake, hence the area of interest will still be 20 ship lengths but further away from the stern of the ship. The average time to location are then compared between the two strategies to find if either strategy provides better in performance or if either is more suitable for certain wake lengths and environmental conditions Fig. 7. The arrival time is also considered to see if arriving at the location when the sought person is in the first sections of the wake or if when it has passed more than approximately 60 percent of the wake has a significant effect on which strategy is most suitable Fig. 8 and Fig. 9 respectively. The speed of the UAVs, speed of the ship as well as the sensor range are set to constant values and do not differ between the search strategies.

The ships that are being considered for the simulation are cruise ships and at current time, the largest ships are approximately 300 m in length, hence the wake lengths will be up to 6000 m, with the assumption that the chaotic wake region is up to 20 ship lengths.
Fig. 7 Overall search time from arrival to the location until detecting the sought person.

Fig. 8 Search time while the sought person is still in the first sections of the wake when the UAVs arrive.

Fig. 9 Search time when the sought person has reached the critical regions (>50% off the wake length).
Discussion of results

From Fig. 7 it can be seen that the overall search time is not affected between the two strategies for shorter wake lengths and in fact the parallel search strategy has a marginally better performance this is despite the lack of cooperation with the other adjacent UAVs. Fig. 8 shows an investigation as to whether an early arrival time at the location has a significant influence on the suitability of search strategy. It indicates in such cases the expanding square strategy has lower detection time for all wake lengths, although it does not vary significantly with the parallel strategy up to larger ship lengths. Whilst in Fig. 9 the arrival time that is considered is when the sought person is beyond 50 percent of the ship’s wake which is set as a critical region. In this case the parallel search strategy has search time down to half of the time required for the expanding square search for wake length of 2000 m and whilst it has higher search time for larger ship wakes, it still shows the best results compared to the other scenario.

Conclusions

Based on the results from Fig. 7 it appears as when the wake area is smaller, the time it takes for the UAVs operating with the parallel strategy to do each run and come back to their initial position will be less, with risk of missing detection. The expanding square search has a more redundant behavior and searches each part of its section more systematically. Consequently has a lower performance than the parallel strategy when the risk of missing detection is lower. It can also be observed from the same figure that the slope for the parallel strategy is higher and quite constant. A logical explanation is that the redundant behavior of the expanding square search in combination with the communication between the adjacent UAVs reduces the number of times they have to cover the section while it takes more runs for the parallel strategy to detect the person. An overall conclusion based on these results is that the expanding square search tends to be a more rigid and reliable search strategy for all the scenarios, whilst the parallel strategy results have more variability. Also the results show that for any scenario, the sought person is detected within minutes. A further area of interest would be how a combination of parallel strategy and expanding square would compare where a broader search area is considered investigating the straightforwardness of the parallel search compared with the redundant behavior of the expanding square search.

References

1. Ivan Zelinka, et al., Evolutionary Algorithms and Chaotic System. Springer.2010, pp. 368
3. Sqn Ldr Rajesh Kumar, ” Tactical Reconnaissance : UAVS Versus Manned Aircraft”. The Research Department Air Command and Staff College, 1997, pp. 29-38
5. Afzal Ahmed, Dr. Masahiko Nagai. and Dr. Chen Tianen., ” UAV Based Monitoring System and Object Detection Technique Development for a Disaster Area”. The International


Paper 2
UAV Swarm Search and Rescue Simulator
This page intentionally left blank
UAV swarm Search and Rescue simulator

Thesis, University of New South Wales, 2012
Ehsan Ebrahimi-Oskoei
ehsaneo@kth.se

Thesis supervisor:
John Randall Page, University of New South Wales
Acknowledgements

I would like to express my deep gratitude to Mr. John Randall Page, my research supervisor, for his guidance, enthusiastic encouragement and support throughout the research work. I would also like to extend my thanks to the fellow students at the simulation laboratory for much appreciated feedback and valuable discussions. Finally, I wish to thank my parents for their support and encouragement throughout my study.
# Table of Contents

1 Constructing a model based on the ship parameters .............................................................................................. 1  
   1.1 Normalized Wake region ................................................................................................................................ 1  
   1.2 Un-normalized wake region ......................................................................................................................... 2  
   1.3 The pattern of the sought person in the wake region ....................................................................................... 3  
2 Simulation of a Search and Rescue operation ....................................................................................................... 4  
   2.1 Parallel search strategy .................................................................................................................................... 5  
      2.1.1 Background ................................................................................................................................................. 5  
      2.1.2 Predefined rules ............................................................................................................................................ 5  
      2.1.3 Section allocation .......................................................................................................................................... 6  
      2.1.4 Control algorithm ....................................................................................................................................... 7  
         2.1.4.1 Control vectors ...................................................................................................................................... 7  
         2.1.4.2 Flight path of each UAV ...................................................................................................................... 8  
   2.2 Expanding square search with intersecting sections .......................................................................................... 10  
      2.2.1 Background ................................................................................................................................................. 10  
      2.2.2 Predefined rules ............................................................................................................................................ 10  
      2.2.3 Section allocation .......................................................................................................................................... 11  
      2.2.4 Control algorithm ....................................................................................................................................... 12  
         2.2.4.1 Control vectors ...................................................................................................................................... 12  
         2.2.4.2 Flight path of each UAV ...................................................................................................................... 15  
   2.3 Detection Algorithm ........................................................................................................................................... 16  
      2.3.1 Pre-search ...................................................................................................................................................... 16  
         2.3.1.1 Synchronizing time of each individual UAV with the time of the sought person ............................... 16  
         2.3.1.2 Control the search time of each UAV ................................................................................................. 17  
         2.3.1.3 Creating a time interval based on speed difference of the person and the UAVs ............................... 18  
      2.3.2 Search and detection ................................................................................................................................... 20  
      2.3.3 Post-detection .............................................................................................................................................. 22  
3 Future Work .......................................................................................................................................................... 24
NOMENCLATURE

$c_B$ The block coefficient of the ship
$d_{red}$ Reduced sensor range diameter
$d_{true}$ True sensor range diameter
$D_{uav,step}$ Distance the UAVs cover in one time step
$L_w$ Horizontal range of the wake
$n$ Number of times a section has been searched
$t_{\text{elapsed}}$ The elapsed time from when UAVs arrived to the scene
$T_{\text{flight path}}$ The total time of a flight path
$t_{n+1}$ The elapsed time from last position of the UAV at its initial position
$t_{\text{step}}$ time step
$v_{\text{uav}}$ Speed of the UAVs
$v_{\text{person}}$ Speed of the sought person
$v_s$ Service speed (average speed of the ship)
$x_{\text{UAV},i}$ x-position of the UAV allocated to section $i$
$x_{\text{UAV,initial}}$ Initial x-position of the UAV allocated to section $i$
$y_{\text{UAV},i}$ y-position of the UAV allocated to section $i$
$y_{\text{UAV,initial}}$ Initial y-position of the UAV allocated to section $i$

$LBP$ Length between perpendiculars of the ship
$\text{mesh}_x$ Number of discreet points in $x$-direction
$\text{mesh}_y$ Number of discreet points in $y$-direction
UAV swarm Search and Rescue simulator

The simulator is a MATLAB based tool. It can be divided into two sections, where section one consists of constructing a model based on the ship parameters and the speed of the UAVs while in section two it simulates a Search and Rescue operation applied to the previous generated model. Two different search strategies have been developed for the Search and Rescue operations, these are Parallel and Intersecting expanding square search strategy.

1 Constructing a model based on the ship parameters

1.1 Normalized Wake region

The simulated environment is created using a suitable chaotic equation (1) to model the wake behind the ship. Equation (1) is applied to form a normalized wake region that is applicable for all ship sizes Figure 1. The next step in the construction of the model is to specify the characteristic parameters that are necessary for the simulator to calculate the size of the wake. Furthermore, discreet points are created for the UAVs to follow for different search patterns.

\[ x_{norm}(i) = \text{current normalized } x - \text{position} \]

\[ y_{norm}(i) = \text{current normalized } y - \text{position} \]

\[
y_{norm}(i + 1) = \frac{\left(x_{norm}(i)y_{norm}(i) - y_{norm}(i)\right) \cdot \left(x_{norm}(i)y_{norm}(i) - \left(2x_{norm}(i) + y_{norm}(i)\right)\right)}{x_{norm}(i) + y_{norm}(i)} + y_{norm}(i)^2
\]

(1)

Figure 1 Normalized wake region
1.2 **Un-normalized wake region**

The simulator requires specifications about the size of the ship, in order to calculate the size of the wake as well as the speed of the ship wake. In addition to the ship parameters, the sensor range and the speed of the UAVs that will be engaged in the Search and Rescue operation also has to be specified. In order to calculate the size and speed of the wake, the simulator requires the user to specify the speed and overall length of the ship, the speed of the UAVs and the range of the sensors at prevalent conditions as shown in Figure 2.

![Image](image.png)

*Figure 2 User input dialog for model construction*

Once these parameters are known, the simulator creates discreet points by meshing the wake region into a size that is equal to the distance the UAVs can fly per time step. The default setting for the time step is set to 0.01 s. The number of discreet points is hence a function of the length of the wake, the speed of the UAVs as well as the time step shown in Equation (2).

\[
D_{uav,step} = v_{uav} \cdot t_{step}
\]

\[
LBP \approx 0.9 \cdot length \ of \ the \ ship
\]

\[
L_w = 20 \cdot LBP
\]

\[
mesh_x = \frac{L_w}{D_{uav,step}}
\]

\[
mesh \ size = \frac{range_{x,\ norm}}{mesh_x}
\]

\[
mesh_y = \frac{range_{y,\ norm}}{mesh \ size}
\]  

(2)
Due to the shape of the wake region, which expands in the y-direction downstream from the origin, it is necessary for the program to model an upper and lower boundary in the y-direction for each x-coordinate. The program stores the maximum and minimum values of the wake region in y-direction at each x-coordinate. Since the mesh size is now known, it uses these upper and lower boundary values to calculate the number of allowed discrete points in the y-direction. These points specify how the UAVs can cover the wake region in y-direction in the flight path coordinate system at any given point. These values are stored and applied to the control algorithm to ensure that the UAVs can fully cover the wake region, but never search outside of the wake region at any point. The size of the mesh (distance between each discrete point) is of great importance, as any data’s about the position of the UAVs will only be known at these points. It is hence required to have a low time step to ensure that the distance between discrete points is much lower than the sensors range of the UAVs.

1.3 The pattern of the sought person in the wake region

In order to simulate a Search and Rescue operation, it is necessary to model a predicted movement of the sought person in the wake region. The pattern of the sought person is modeled based on the same chaotic equation that was used to model the ship wake. The speed of the sought person is assumed to be the same as the speed of the ship wake. The speed of the ship wake can be expressed as a function of the service speed and the length of the ship that is in contact with the water \( LBP \) see Equation (3), both parameters are previously specified by the user in the modeling of the wake region.

\[
C_B = 1.2 - 0.39 \left( \frac{v_s}{\sqrt{LBP}} \right) \\
\nu_{\text{person}} = \left( \frac{C_B}{2} - 0.05 \right) \cdot v_s = \left( \frac{1.2 - 0.39 \left( \frac{v_s}{\sqrt{LBP}} \right)}{2} - 0.05 \right) \cdot v_s \quad (3)
\]

Once a feasible motion has been created, the simulator randomly selects some points as reference values. The simulator then creates new points between the selected reference values with respect to the speed of the sought person and then goes on to create new reference points between the newly created points. This procedure is repeated several times, where the points created are allowed to change direction in both x and y-direction. It is assumed that the person can move in any direction during a time step. This is in order to simulate a realistic environment where currents, winds and waves will cause the motion of the sought person to move back and forth in x and y-direction as shown in Figure 3. When the motion of the sought person has been created, the program calculates the distance and time the person has been moving along each point. By using the previous and the new coordinates, the program calculates the distance and time between the two points as well as the elapsed time from the moment the person fell overboard. This procedure ensures that the motion is following the chaotic behavior of the wake as well as a unique motion for the sought person every time a simulation is carried out. It also gives more continuity to the motion, since the motion is based on the time step, hence the position and time of the person will be known at each time step. This allows synchronization with the time of the UAVs. Knowing that it takes several time steps for the sought person to cover a distance that a UAV will cover during a time step will give means to take this into account. This is crucial for verifying if the sought person is located within the sensors range of a UAV at a given time and will be discussed in the Detection algorithm section.
2 Simulation of a Search and Rescue operation

The idea behind controlling the motion of the UAVs without the need of user interaction will make it necessary to have predefined rules that will obey the characteristics of the desired search strategy. It is also necessary to define initial conditions that will be eligible for different wake lengths. The developed simulator contains as stated earlier, two different strategies to search for the sought person within the chaotic ship wake. The development of each of the strategies can be divided into two main sections, namely control algorithm and detection algorithm.

The control algorithm ensures that the UAVs are following the characteristic pattern of the chosen search strategies, while the detection algorithm is used to verify if the sought person has been found. The parallel search strategy was initially implemented in the program by having the detection algorithm embedded within the control algorithm, i.e. to guide each UAV during the entire time that the search was performed. This required numerous calculations at each time step based on the boundaries, section properties etc. which was shown to be inefficient especially when a UAV had to search the same section repeatedly and hence had to repeat the same calculations. Thus a different approach was carried out in the construction of the intersecting expanding square search, mainly to reduce the computational time for larger wake lengths. The new approach was to separate the control and detection algorithm, where the control algorithm’s function is to create an appropriate flight path for each UAV to fully cover its section. By doing so, it will not eliminate the calculations based on boundaries, section properties etc., but it will eliminate the necessity of repeating the same calculations. This procedure will also enable the detection algorithm to perform several search and rescue operations for different arrival times to the scene without needing to go through the extensive calculations, as the flight paths are already established. Once the latter approach showed to be successful, the parallel search strategy was modified and constructed in the same manner. The development of the modified parallel search strategy and the intersecting expanding square search is discussed in the following sections. The detection algorithm

Figure 3 Pattern of the sought person in a ship wake of 10 000 m.
between the two strategies does not differ and is hence discussed after the sections explaining the implementation of the two search strategies.

2.1 Parallel search strategy

2.1.1 Background
This strategy is based on a parallel “sweeping” motion as shown in Figure 4, where the black arrow represents the forward motion and red arrow the backward motion when the UAV has reached the end of its section. The wake region is divided into a lower and upper region and the UAVs will cover a specific section in either the lower region or the upper without any intersecting between the adjacent sections and all UAVs flying at the same altitude and velocity.

![Figure 4 Search pattern of each UAV in the parallel search strategy.](image)

2.1.2 Predefined rules
Based on the characteristics of this search strategy, the following predefined rules were to be obeyed in order to guide the direction of a specific UAV.

1. The UAV has to move either upwards or downwards until it has reached the upper/lower boundary value.

2. When the boundary value has been reached, the UAV has to move sideward. The direction of this sideward movement is decided by whether the UAV is making a forward or backward motion.

3. The forward to backward motion should be decided by when the UAV has reached the end of its section and need to move back to the original start position and vice versa.

4. Once the UAV has reached either the end of its section in the forward movement or the beginning of its section in the backward movement, the new initial direction should be in a way to coincide with the previous covered area.
2.1.3 Section allocation

The section allocation for the UAVs is based on dividing the wake region into an upper and lower section; these sections are subsequently divided into five sections in the horizontal range. The section allocation scheme is shown in Figure 5.

![Figure 5 Section allocation scheme for the parallel search strategy.](image)

The chaotic behavior of the wake makes it extremely difficult to predict the location of the person even when the time the sought person has fallen overboard is known; hence each section of the wake has to be searched. Several assumptions can be made to justify the proposed section allocation strategy, for instance it is less likely that the sought person is still in the beginning of the wake by the time the UAV swarm has arrived to the scene and can be qualified as a low priority region. The second assumption is that it is less likely that the person is found in good condition at the end of the wake region and this area can be considered as a low priority region. This allows the search time for the remaining search sections to be more uniform, i.e. despite the growth of the wake region in y-direction; it ensures that these sections are fully covered within a time limit. The low priority regions are decided from the normalized wake region in order to be applied for all wake lengths as shown in Figure 6, hence will vary in size depending on the wake length.

![Figure 6 High/Low priority regions of the normalized wake region](image)
2.1.4 Control algorithm

The control algorithms functions such as following: to process the data’s obtained during the modeling of the wake and the section allocation procedure to guide each individual UAV. The UAVs are thus able to perform the desired search strategy pattern within the proposed sections and ensures that none of the UAVs ever searches outside of the wake region. The data’s of each UAV’s position at every time step is then stored and later provided to the detection algorithm. In order to control and guide each individual UAV, a large number of parameters are introduced, where the main parameters can be assembled into control vectors, such as Positional, Directional and Boundary vector.

The program also calculates the time for each UAV to fully cover its section based on the parallel search pattern from the data’s obtained when modeling the wake region. It is done by checking the section properties and at which points the x-coordinate will be constant along the flight path, i.e. when the UAV will move in the y-direction. The program then uses the stored data’s about the boundary values at each x-coordinate to calculate the number of discreet points in the flight path coordinate system the UAV will need to cover at each of the points where the x-coordinate is constant. It also takes into account that the sensors of the UAV will decrease the amount of discreet points necessary to fully cover the area. By summing the total number of necessary steps in y-direction as well as in x-direction, the total time can be calculated, as each step in the flight path coordinate system corresponds to a time step.

2.1.4.1 Control vectors

Positional vector:
The positional vector is the UAV’s position in the flight path coordinate system.

Positional vector: $[x_{UAV,i}, y_{UAV,i}]$, $i = \text{allocated section}$

Directional vector:
The initial direction of each UAV is set to begin its motion from the midsection towards the outer part of the wake region in y-direction, hence the initial direction depends on if the UAV is allocated to a section in the lower or the upper wake region.
The parameters in the directional vector for the parallel search strategy are:

Directional vector: $[\text{up}, \text{down}, \text{side up}, \text{side down}, \text{level}, \text{forward}, \text{backward}]$

These parameters are based on a binary system. It can be illustrated by the description of point 1-4 in Figure 7. This corresponds to a forward parallel movement:

Point 1: $[\text{up}, \text{down}, \text{side up}, \text{side down}, \text{level}, \text{forward}, \text{backward}] = [1, 0, 0, 0, 1, 0]$

Point 2: $[\text{up}, \text{down}, \text{side up}, \text{side down}, \text{level}, \text{forward}, \text{backward}] = [0, 0, 1, 0, 1, 0]$

Point 3: $[\text{up}, \text{down}, \text{side up}, \text{side down}, \text{level}, \text{forward}, \text{backward}] = [0, 1, 0, 0, 1, 0]$

Point 4: $[\text{up}, \text{down}, \text{side up}, \text{side down}, \text{level}, \text{forward}, \text{backward}] = [0, 0, 0, 1, 0, 0]$
Figure 7 Direction changes in the forward parallel movement.

Boundary vector:
The boundary vector will use the data’s of the upper/lower boundaries at each discreet point, the section properties as well as the positional and directional vector to prescribe each individual UAV with the following information:

- Initial position
- Final position
- Boundary and the positional vector will at each time step control whether the current direction (directional vector) is to be followed at the next time step(s). The number of steps in the flight path coordinate system before the direction should be altered (required \( x, y \)) is stored and used to maneuver each UAV.

**Boundary vector:** \([\text{required } x (x_{\text{UAV}, i}, y_{\text{UAV}, i}), \text{required } y (x_{\text{UAV}, i}, y_{\text{UAV}, i}), \text{min } (x, y) \text{ position, max } (x, y) \text{ position}]\)

2.1.4.2 Flight path of each UAV

The collaboration of the control vectors will set the resources to create a flight path for each UAV. The created flight path for each UAV can be seen in Figure 8, where each color specifies the path of a UAV in the upper/lower region. The created flight paths are based on the sensors range of the UAVs, where the distance in the x-direction from point 1 to point 4 in Figure 7 will correspond to the sensors range in radius and the distance it will move towards a upper/lower limit is also reduced based on the range of sensors. However the sensor range considered for the flight path is reduced to some extent; this reduction is discussed more thoroughly in the detection algorithm section. In Figure 9, the area in the wake which corresponds to the upper region of the wake in section one and three is displayed with coordinates of the UAVs at different locations. The minimum distance between the UAVs allocated to the
and lower region will be two times the sensor radius, which is the distance the sensors of the UAVs will be able to cover. The UAVs will also always ensure that the outer parts of the wake region are always within the range of the sensors.

Figure 8 Flight paths for the UAV swarm.

Figure 9 Zoom in at the flight paths of UAV 1-4, Sensor radius = 4.2 m
2.2 Expanding square search with intersecting sections

2.2.1 Background
This strategy is based on each UAV following an expanding square pattern as shown in Figure 10. The UAV swarms behavior will be such that each UAV will intersect the adjacent UAVs section by flying to some extent into that region, to ensure a fuller coverage of the wake area. In order to avoid the risk of collision, the adjacent UAVs will fly at a slightly different altitude, without losing any major accuracy in sensor resolution. Altitude change can in fact be a strategy by itself, where the low priority regions can be covered quicker by having a higher altitude but with less accuracy, hence releasing the UAV quicker to join the search in the more critical regions of the wake.

![Figure 10 Search pattern of each UAV in the expanding square search](image)

2.2.2 Predefined rules
Based on the characteristic of this search strategy the following predefined rules were to be obeyed in order to guide the route of a specific UAV.

1. The expansion of the squares should be based on the sensor range, i.e. every segment of the enclosed area of current square should have been within the sensor range at some time.
2. The UAV must maintain its direction until it reaches the edges of current square.
3. When UAV has reached the edge of a current square, the new direction should be opposite of the last direction change.
4. When the UAV reaches the upper/lower part of the largest square route, it switches to parallel search strategy to search these segments.
5. Once completed the search of the outer segments of the section, the UAV should move back to the initial search position.
6. The UAV has to intersect the neighboring sections, without the risk of collision.
2.2.3 Section allocation

The sections that each UAV will cover are more time dependent than for the parallel search strategy i.e. the area that the UAV will cover is broader, where the sought person is less likely to be, such as in the beginning of the ship wake. As mentioned before this area cannot be neglected owing to the complexity of a chaotic region and the difficulty to estimate the position of the sought person, it is still a reasonable assumption that this region is of low priority, if the sought person is in this area and hasn’t been detected from the ship. Other assumptions can also be used to improve the efficiency of the search. The region further away from the origin of the wake is more critical to finding the victim in a good condition. Hence the search sections of the UAVs will become narrower and allow coverage of these critical areas repeatedly over a short time period to ensure that the emerged person is detected and rescued as quickly as possible, the proposed distribution of the group of UAVs can be seen in Figure 11.

![Figure 11 Section allocation scheme for the expanding square search strategy](image1)

Note that while for the parallel search strategy, the priority level was constant in the high priority region, in this strategy the priority level increases with distance from the low priority regions as shown in Figure 12.

![Figure 12 High/low priority level for the wake region.](image2)
2.2.4 Control algorithm

The control algorithm implemented has one function and that is to create a flight route for each UAV which obeys the predefined rules mentioned above. In order to do so a large set of parameters were introduced and assembled into vectors in the same manner as was done for the parallel search strategy. These vectors consist of boundary, expansion, direction, positional vector as for the control algorithm of the parallel search strategy.

2.2.4.1 Control vectors

**Boundary vector:**
The flights routes of the UAVs are to be intersecting as well as the section allocation are more priority based than the more straightforward approach that was undertaken with the parallel search. The new approach for section allocation is to create initial search position for a specific UAV with respect to the previous UAV’s initial search positions. This procedure allows one to change the priority of a region and the other sections will adapt, since the sections are not predetermined.

At each instant the program will check the previous sections extent and create a phase parameter, the phase parameter is going to be a set percentage of the magnitude of the previous section, hence decide to what degree the sections will intersect each other. With the phase known, it establishes the boundaries of the section, based on what priority level this section is to have. The higher priority the narrower will the section be, hence allowing a smaller time span to cover this section.

**Boundary vector:** \([\text{min x position, min y position, max x position, min y position}]\)

**Expansion vector:**
The expansion vector checks the properties of the wake within the established boundaries of the section and determines the appropriate growth rate in x and y direction with respect to the predefined rules.

The growth in y-direction is determined by first checking the distance from the start position to the outer boundaries in y direction at the minimum x position for the specific section, the reason for using the minimum x position is to ensure that the route created will at all times be within the wake. Once the distance has been determined, the number of allowed expansions based on the current sensor range is calculated, i.e. number of square patterns that are necessary. The growth in x-direction is then determined by the span of the section as well as the necessary number of square patterns. The growth rate parameters assembled into the expansion vector will together with the parameters of the boundary vector create means to guide each UAV within their specific section.

**Expansion vector:** \([\text{allowed number of squares, growth rate in x direction, growth rate in y direction}]\)

Once the expansion has been established, the appropriate position of switching to parallel search is calculated for the regions of the section that are at the outer boarders of the wake.

A switch between the two different search strategies, allows the outer boundaries to be searched without the UAVs ever searching outside of the wake region, which would have been the case if only the expanding square strategy would have been implemented, due to the convergent shape of the wake.
Positional vector:
The positional vector is as was described for the parallel search, with the only difference that it now also contains the initial positions of the specific UAV, due to that this positions are not determined until the phase parameter has established the scope of the section.

\[
\begin{align*}
\text{Positional vector: } [x_{\text{UAV, initial}}, y_{\text{UAV, initial}}, x_{\text{UAV, i}}, y_{\text{UAV, i}}], \quad i = \text{allocated section}
\end{align*}
\]

Directional vector
The directional vector is based on the same system that was applied for the parallel search. The parameters within the vector are nearly the same, with the exception of the parameters indicating movement towards/from the horizontal end of the section i.e. to research the section in the parallel search strategy.

\text{Directional vector: [side up, side down, level, forward, backward]}

The directional parameters will together with the expansion vector and positional vector, create the expanding square pattern and the initial position of the parallel search strategy for the outer boundaries of the wake. The flight path creation is divided into different steps, depending on the pattern the UAV is ought to follow and is illustrated in figure 13.

Step 1: creating the expanding square pattern for the specific section.

Step 2: switch to parallel search, when the UAV had reached the final position of the expanding square search.

Step 3: Stop the parallel search, when the UAV has reached the end of the horizontal range of the section and move towards the lower boundary limit.

Step 4: Switch back to parallel search for the lower boundary.

Step 5: Stop the parallel search at the position where the sensors of the UAV have covered the entire section and return to the initial position of the expanding square pattern.
Figure 13 The different steps in creating the flight path
2.2.4.2 Flight path of each UAV

As for the parallel search, the collaboration of the control vectors will enable to create the desired flight path, in this case a flight path of each UAV corresponding to an expanding square pattern, where the flight paths intersect the adjacent sections as shown in Figure 14. The rate in which the squares expand is set to half of the reduced sensor range as shown in Figure 15. It is done in order to enable detection of the person, at times when a person was just outside the range of the sensors during the previous square pattern.

Figure 14 Shows the collaboration of the UAVs in the intersecting expanding square search

Figure 15 Zoom in at the flight paths of section 1-4. Sensor radius = 4.2 m
2.3 Detection Algorithm

The final part of the simulation consists of the detection algorithm, where the established flight paths for each section are then incorporated in the detection algorithm as well as the data’s available from the model construction to set the necessary grounds to perform the actual search operation and to establish if the sought person will be detected. The construction of this algorithm can be divided into three phases:

i. pre-search

ii. search and detection

iii. post-detection

2.3.1 Pre-search

The three main conditions that need to be fulfilled before performing the search operation are:

a) Synchronize the time of each individual UAV with the time of the sought person and allow the UAVs to search the wake region simultaneously

b) Control the search time of each UAV.

c) Creating a time interval based on speed difference of the person and the UAVs.

2.3.1.1 Synchronizing time of each individual UAV with the time of the sought person

The time synchronization of each individual UAV with the time of the sought person is done by converting the arrival time of the UAVs to the scene to the number of time steps the person has been spending in the wake region by the time the UAVs arrive. The program uses the data’s regarding the sought person’s velocity and movement in the wake region to present several options to the user to specify the arrival time to the scene as shown in Figure 16.

---

![User input dialog for arrival time of the UAVs to the scene](image)

*Figure 16 User input dialog for arrival time of the UAVs to the scene*
The first option is in terms of the actual time [hours:mins:secs], where the user is informed about the upper limit, i.e., the total the person spends in the water before reaching the end of the wake region. The second option is in percentage of the upper limit and the last option is to allow the simulator to randomly choose an arrival time within the time interval that the person is still in the wake region, default setting is set to a random value of 20-80% of the upper limit. If option one is chosen, but the arrival time is higher than the upper limit, the user will be prompted to choose a lower value as shown in Figure 17.

![Figure 17 Warning dialog when arrival time corresponds to the time the person has left the wake region](image17)

It is also important, that the user only chooses one of the options, as the program will not be able to distinguish which of the values that is to be chosen, in such case the user is informed to only fill in one of the boxes as shown in Fig. 18 and the input window for arrival time will be displayed again.

![Figure 18 Warning dialog when two different arrival times are chosen](image18)

The user input for the arrival time permits to simulate how the arrival time to the scene will affect the elapsed time before the person is detected. As mentioned earlier, it is crucial that the elapsed time is lower as the arrival time to the scene increases (low/high priority section).

2.3.1.2 Control the search time of each UAV

In order to simulate a real search operation, in which the UAVs only can perform the search operation under a limited time before refueling, it is also necessary to establish a time limit for the search procedure, an allowed search time. It is done by the program by presenting several options in deciding the time limit as shown in Figure 19. As mentioned earlier, the sections in the low priority regions are created to be broader in range than the sections in the high priority region. This is performed in order to ensure that it takes less time to cover a section in the high priority region. Therefore by presenting the options in terms of the average time it takes to cover each section, it enables to choose a low/high priority based value. The second option
is to instead choosing the number of times the largest section (low priority section) is to be covered, ensuring that all section will be fully covered. The last option is to allow the simulator to choose the time limit in terms of number of times the largest section is to be covered, default setting is set to three. The time limit is also dependent on when the UAVs arrive to the scene.

![User input dialog to specify the time limit of the search](image)

Figure 19 User input dialog to specify the time limit of the search

If the sum of arrival time and time limit is higher than the total time the person spends in the wake region, the user is informed that with current arrival time, the search can only be performed during a shorter period of time as shown in Figure 20.

![Information box, specifying the appropriate time limit for the search](image)

Figure 20 Information box, specifying the appropriate time limit for the search

2.3.1.3 Creating a time interval based on speed difference of the person and the UAVs

The simulator must be able to distinguish if the person is currently within the sensor range of the UAV, or if it has simply been at this position during a time when the UAV was located in another part of the section. Even though the time of each individual UAV is synchronized with the sought person’s time, there is another matter that must be taken into account; the UAV and the person do not move with the same speed. This issue together with the fact that the distance between each discreet point is based on the distance the UAVs can fly during a time step, makes it inevitable to create a time interval which takes into account that the person can be within the UAVs sensor coverage but not detected due the above mentioned reasons.
The approach for creating the time interval needed for the detection algorithm is to reduce the sensor range in a way that based on the person's speed, even if the person is at the edges of the reduced sensor range, it would take a minimum number of time steps (time increment) before the person would actually be outside of the true sensor range as shown in Figure 21. The time interval is a function of the time step and the desired time increment Equation (4). The next step is to take into account that the UAV is moving as well (sensor range coverage changes). This results in a decrement of the newly created time interval in such a way that the person during the new decreased time interval would not only be within the true sensor range but also within a range that is covered by the sensors during this time interval.

\[ t_{\text{interval}} = t_{\text{step}} \cdot \text{time increment} \]  \hspace{1cm} (4)

Figure 22 illustrates how the sensor range coverage changes at different time steps, as in this case during two time steps where t corresponds to the current time. It is clear from the figure that the reduced sensor range should be small enough to always be within the sensor range during the two time steps.

The time interval is created by looking at how much the sensor coverage changes in each direction at each time step.

As an example, with a UAV speed of 24 m/s, persons speed 4 m/s, a step time of 0.01 s and a true sensor area coverage of 10 x 10 m², after 2 time steps in forward motion the left part of the sensor area has moved about 0.48 m to the right. It is hence clear that creating a large time interval for the detection algorithm requires a high reduction in sensor range. Although the pattern in which the person moves has been altered to simulate direction changes due to wind, waves etc. (see section 1.3) it does not take into account that the speed of the person is not constant, hence the calculated speed can be considered as the average speed. Therefore the aim is to have as small time interval as possible that would correspond to an insignificant change of the person's movement despite if the actual speed is above the calculated average speed.

The default setting for the time interval is currently set to two time steps with reduced sensor range coverage of 8.6 x 8.6 m², which would correspond to a maximum movement of 0.08 m of the person in any direction and is well within the true sensor range coverage during the time interval. It should be noted that the program automatically reduces the sensor radius by
10%, before reducing the sensor range once again according to Equation (5). The reason for the first reduction is to take into account that the UAVs will need to fly at slightly different altitudes for the intersecting expanding square search to avoid collision. In order to enable comparison of the two search strategies under same conditions, the same reduction in sensor range is also done when performing the parallel search strategy. The second reduction is dependent on the speed of the UAVs and person as well as the chosen time interval as can be seen in Equation (1.5).

$$ (v_{\text{person}} + v_{\text{uav}}) \cdot \text{time interval} < d_{\text{true}} - d_{\text{red}} $$  \hspace{1cm} (5)

---

2.3.2 Search and detection

Once the arrival time has been specified, the search procedure is constructed according to the section allocation, where each section is searched one after another. By setting the initial search time of each UAV to the arrival time, it will simulate as the UAVs are searching the sections simultaneously, as a swarm of UAVs. The detection algorithm will then at each time step use the flight path for the UAV allocated to the specific section to check the current position of the UAV as time elapses. The current position of each UAV will be known at all times during the search, as each step within a specific flight path corresponds to one time step. This holds true even when a specific section has to be covered several times as shown in Equation (6) and Equation (7).

$$ t_{\text{elapsed}} = n \cdot T_{\text{flight path}} + t_{n+1} $$  \hspace{1cm} (6)

---

Figure 22 Sensor coverage at different time step.
\[
t_{n+1} = \begin{cases} 
  t_{n+1} & \text{if } 1 < t_{n+1} \leq T_{flight\ path} \\
  1 & \text{if } t_{n+1} > T_{flight\ path}
\end{cases}
\] (7)

The time variable \( t_{n+1} \) decides the position of the UAV along its corresponding flight path while \( t_{\text{elapsed}} \) corresponds to the elapsed time from when the search began. The current position of the person is thus also known during the search as the elapsed time of the person is the sum of \( t_{\text{elapsed}} \) and arrival time. The detection algorithm will at each time step during the time limit of the search, check if the person is within the reduced sensor range coverage of a UAV during the specified time interval. If the person is detected, the search will be aborted and the following will be displayed in the command window see Figure 23:

- Section and (x, y) coordinates of the detected person
- (x, y) coordinates of the UAV
- The elapsed time until person detected, \( t_{\text{elapsed}} \)
- Reduced sensor range radius
- Wake length

Since the parallel search strategy’s sections has been arranged to have two UAVs that will search within the same horizontal range of the wake without any intersection between any sections, the detection algorithm will only abort the search for the UAVs that are located in the sections further away from the origin of the wake. Thus, it takes into account that the person might be found quicker in the upper/lower part by the other UAV searching the same segment of the wake in x direction. If the person is found in both the upper and lower region of the same segment, both of these detections will be displayed in the command window.

```
>> detection_par
Searching the sections of the wake region for the sought person
(x,y)-coordinates of the sought person when detected
   1960.2   -12.348
Position of UAV:
   1959.9   -13.337
x-coordinate = distance from origin of the wake [m], y-coordinate = distance from the midsection of the wake
The elapsed time until person detected in section 8 (lower) [hours:mins:secs]
00:02:52.52
Reduced sensor range radius [m] :
   4.26
Horizontal range of the wake [m] :
   2700
```

*Figure 23 Command window, displaying the coordinates of the UAV, person and time when person detected*

While for the expanding square search strategy, where the sections do intersect, hence cannot be excluded that any of the UAVs allocated to the adjacent sections might found the person quicker. Thus the detection algorithm will not abort the search until all sections, has been covered and displays all the positions at which the person was detectable and then uses the lowest elapsed time to establish by which UAV the person was first detected.
The sensor range is by obvious reasons the main parameter deciding the feasibility of detecting the person, especially when time is of crucial matter. The sensor range is implemented in three different forms in the program; in terms of the actual range, in terms of the corresponding range within the normalized wake region and finally, in terms of number of discreet points in the flight path coordinate system. The first form is used during the reduction in sensor range procedure, while the two other forms are used during the construction of the flight path and detection algorithm respectively.

2.3.3 Post-detection

The post-detection part of the program is to visualize the search and rescue operation. The program uses the lowest value on $t_{\text{elapsed}}$, i.e. the value corresponding to the elapsed time of when the person first was detected and plays a short movie sequence. In Figure 24, a snapshot of the movie sequence illustrating the scenario of the parallel search strategy as according to the script in Figure 23. The sequence displays how the person and the group of UAVs are moving within the wake region before the person is detected. The arrival time, elapsed time of the search, each UAVs position as well as the person’s position (red letters/numbers) is displayed at each time step.

![Figure 24](image)

*Figure 24* Snapshot of the movie sequence before the person was detected by the UAV allocated in section 8 for the search and rescue operation based on parallel search strategy.

Figure 25, displays when performing a search and rescue operation based on expanding square search pattern on the same wake region, using the default settings.
Figure 25 Snapshot of the movie sequence before the person is detected by the UAV allocated in section 4 for the search and rescue operation based on expanding square search strategy.
3 Future Work

In this section a list of what the author believes is suitable for future work is presented:

- Extend the search area:
  The UAV swarm searches only within the wake region, a realistic study for future work is to expand the search area to allow detection even when the person has washed out from the wake region. It is feasible as the initial point for a problematic search downstream of the wake is determined by behavior in the wake region.

- Multiple persons:
  The current software is designed for a search and rescue of a single person, hence aborted once the person is detected. A continuation of this work would be to implement the possibility when several persons are within the wake region.

- GPS/Pattern of the person:
  An interesting investigation would be to establish an actual pattern of an object moving within a wake region, by dropping an object with GPS transmitter at the stern of a ship and recording the GPS coordinates as it moves within the wake region. The flight path coordinate system is designed as discreet points with origin at the stern of the ship, by implementing the position of the ship in GPS coordinates, a search and rescue operation under real conditions can be simulated.

- Section allocation:
  The UAVs are allocated a specific section, this section and hence the area that each UAV will cover is constant with respect to time, a further enhancement would be to modify the software to allow the UAVs to have a variable section, such as:

  - Sections moving downstream of the wake with time, especially for the case when an extended search area has been applied and the UAVs will need to search beyond the wake region.
  - The UAVs in the low priority region just behind the stern of the ship allowed searching the section during a specific time, to fully cover the section, but then join the UAVs in the high priority region.

- Visualization:
  Improving the visualization of the search and rescue simulation part, such as; incorporating the Matlab code into a Simulink model.