ROUTE PLANNING FOR RELAY UAV

Anders Holmberg* **Per-Magnus Olsson**
SAAB AB **Linköping University**

Keywords: *relay UAV, payload UAV, preplanning*

Abstract

To expand the operative area for surveillance UAV, we propose the use of a relay UAV. The relay UAV is used as an intermediary node in a communication network: the surveillance UAV transmits data to the relay UAV, which sends it back to a ground station. In this exploratory report, we calculate the route for a relay UAV, to ensure communication at certain time points, given the route of the surveillance UAV. The results presented here are preliminary and may be considered a first iteration of ideas and methods.

1 Introduction

Saab Aerosystems have been developing manned aircrafts, both civilian and military, since 1937. During the last decade two demonstrator programs have resulted in two unmanned aircraft vehicle (UAV) prototypes: SHARC and FILUR. The interest has grown into a business area and SAAB is currently involved in the NEURON project designing an autonomous fixed wing aircraft. Saab's first UAV product, the helicopter Skeldar, is under construction. Skeldar is a fully autonomous unmanned aerial vehicle (UAV) with vertical take off and landing (VTOL) capacity with a maximum takeoff weight of 150 kg. The payload capacity is 30 kg which makes it suitable for surveillance with its electro optical and infra red sensors. [7].

The Artificial Intelligence and Integrated Computer Systems division at Department of Computer and Information Science at the Linköping University has carried out research and demonstrations with their autonomous Yamaha RMAX UAV helicopters since 1997 [1].

Through Linklab, a joint venture with Saab and the Linköping University, both parties together perform research in the area of future aviation systems [6]. One of several research areas is autonomous systems, where knowledge gathered from Linköping University's UAVs' are transferred into Saab projects such as the Skeldar platform.

The problem of interest in this paper is to extend the area of operability for a payload UAV to go beyond visual line of sight from the ground station. This is done by introducing one extra UAV carrying information relay equipment, and calculating its route.

2 Problem Formulation

In this chapter we describe the problem and discuss some ways to represent the environment.

2.1 Problem Setup

Consider the task of a UAV to perform some surveillance mission. There are many different applications; both peaceful and military, for this, some of them are photogrammetry, search and rescue operations, surveillance of areas, forest fire control and more. It is required that the UAV must transmit information back to some ground station in real time, or at least at specified time points, and that the transmission requires line of sight between the UAV and the ground station to be performed. In this case, the operating area of the UAV is limited to the range of the communications equipment, and the UAV may not move behind hills, mountains, buildings or other solid objects. This would severely limit the usefulness of the UAV and to

anders.holmberg@saabgroup.com SAAB AB. *Corresponding author.
perol@ida.liu.se Department of Information and Computer Science. Linköping University, Sweden.

remedy this we want to add a second UAV, acting as a relay between some ground station and the first UAV. The task of the relay UAV is to forward information from the payload UAV back to the ground station in real time, and to do so it must place itself in such a position that it can receive transmissions from the payload UAV and in turn transmit them to the ground station. At the same time, it must stay within transmission range of the payload UAV as well as the ground station.

Fig 1 shows how a UAV, from now on referred to as a payload UAV, sends information to a ground station using a relay UAV. The problem that we investigate here is where to place the relay UAV to ensure that it can receive transmissions from the payload UAV and send these to the ground station. An obvious solution is to place the relay UAV at very high altitude, as suggested in [5]. A drawback with doing so is that payload UAV would have to transmit the information with higher power to reach the relay UAV. There might also be other limitations such as limits on air space from aviation authorities on where we may or may not fly.

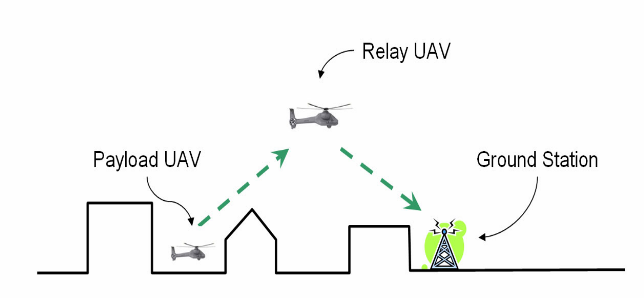

Fig 1 Through the use of a relay UAV, a free flow of information from the payload UAV to the ground station is assured.

To keep the power consumption at a minimum we focus on calculating positions within a limited range from the payload UAV, where the range is dependent on the range of the UAVs' communication equipment. The relay UAV is instructed to stay fairly close to the ground, while still maintaining line of sight to the payload UAV and the ground station. This implies that we must take the terrain and buildings into consideration, as these will otherwise block the line of sight. A similar problem is investigated in [2] where the goal is guide a small UAV to some survey position, while at the same time maintaining communication with a ground station, using a another UAV as a relay. This is performed using extensive pre processing and mixed integer linear programming as an on line solution method.

2.2 Scenario Description

In this report, we consider the case of a payload UAV, a relay UAV and a static ground station. Transmissions between the payload and relay UAV takes place when the payload UAV is at certain predetermined positions. There are no aerodynamic limitations except maximum acceleration, retardation and speed. There is no wind.

We assume that a valid route for the payload UAV is given. It contains a number of positions that the payload UAV must visit to perform its surveillance mission. The task of the relay UAV is to position itself so that it can receive transmissions and forward these to the ground station immediately after they are received. This creates a hierarchical relationship between the UAVs as the relay UAV must adjust its path depending on the path of the payload UAV and has no other task.

The relay UAV uses the route of the payload UAV to determine its own route, given suitable mission restrictions. After the route calculation is performed, it is displayed to the operator. If the operator accepts the route, the relay UAV is ready and the mission can start. Otherwise the operator may manually replan the route for the relay UAV.

2.3 Environment Representation

One important factor to consider is how to represent the environment. There are two basic representations: continuous and discrete. These are treated in this chapter.

2.3.1 Continuous Representation

A plane can be used to limit an area in a three dimensional space. Each plane divides the space in two parts and the plane equation tells which part belongs to the area of interest. Several planes can be used to form complex shapes such as pyramids, cubes etc. An advantage with this representation is that arbitrary shapes can be modeled if the number if planes is high enough. A disadvantage is that only convex shapes can be easily modeled. If non-convex shapes are to be modeled, they must be modeled as several convex shapes, each consisting of several planes.

One advantage of continuous representation is that it is easy to check whether a point is inside a volume defined by the planes. It is quite simple to check if the point fulfills the plane equations. However if the number of planes is large, this is also a large disadvantage as the time needed for intersection checks is increased. It is possible to speed this up to some extent using spatial partitioning algorithms such as quad trees and k-d trees [3].

2.3.2 Discrete Representation

In a discrete representation, the world is divided into different cells, normally squares and rectangles. Commonly squares are used and in each square its height is given by a singular value.

There are some implications of choosing a discrete representation of the environment. Some of the most obvious are the size of each cell. If the size of the cell is too large, there might be large variations in terrain height within the square, leading to large discrepancies between the representation and the true terrain. On the other hand, if the cell size is small, the number of cells will be very large. Also, there are many different methods to determine what height value should be in the cell: the most common is based on sampling some terrain data base to determine the height in each cell. This leaves a lot of choices: should a single sample determine the height or several? Should the highest/lowest or the average value be used? Regardless of what method is used to determine the height, a tessellated height map is created. The use of a discretized representation has the advantage of fast calculation of cell number from coordinates.

This representation is quite similar to the representation used in computer graphics, and this makes computer graphics algorithms easier to use.

2.4 Implemented Representation

In this application, we chose to store the terrain as planes, i.e. a continuous representation. This is based on the fact that there will be quite few objects in the environment and the disadvantage of long execution time is thus not a problem.

The air above the terrain is discretized into volumes, giving us a hybrid representation. These volumes are used to store the common fields of view, as described in section 2.5. At each position, there will be a vertical pillar of volumes from the lowest to the highest point in common field of view. However, even if the common field of view would be infinitely high, we will limit the maximum value at some point to represent limitations of where we may fly from aviation authorities.

2.5 Common Field Of View

To determine to which positions it is possible to send messages from some position, such as from a UAV or from the ground station, we use a ray casting algorithm. In such an algorithm a ray is shot from the start position to the target position, checking for intersecting objects along the way. If no object is intersected there is free line of sight between the start and end position. From a potential UAV position, rays are cast to all volumes within the communication range of the UAV. The set of volumes is then associated with the UAV's position. The same thing is done from the ground station's position. When performing these calculations, we assume that all transmission is equally strong in all directions.

If there are no obstacles in the way, a sphere with radius equal to that of the UAV's communication range is formed. If there are objects within the radius, these must be removed from the sphere. This can be done for example using constructive solid geometry, as described in [4]. **Fig 2** shows two sets, where the volumes ground station and the payload UAV can transmit to are shown as shaded areas. The set of volumes that are common to the payload UAV and the ground station is shown in darker color. The common field of view is from now on abbreviated CFOV. If a relay UAV is placed anywhere in the CFOV, it is guaranteed that it can receive transmissions from the payload UAV and forward the information to the ground station. Mathematically, the CFOV is the intersection between the fields of view of the payload UAV and the ground station.

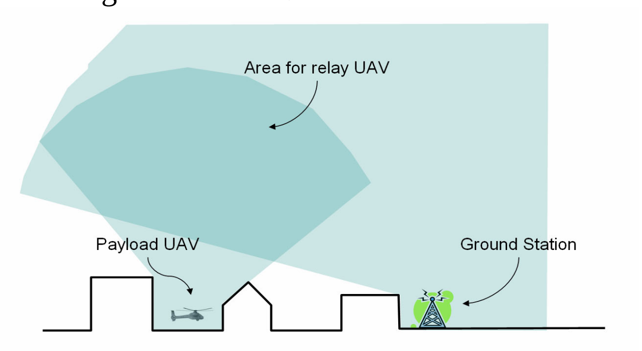

Fig 2 By adding the fields of view for the payload UAV and the ground station, we can see the common field of view. Only from the common field of view there is free line of sight to both the payload UAV and the ground station. The relay UAV must thus stay in the common field of view to be able to relay the information from the relay UAV to the ground station.

3 Route Planning

The problem of interest in this article is consequently to offline calculate a route for relay UAV when the route of the payload UAV is known in advance. Given the route of the payload UAV and points where link coverage is needed calculation of CFOVs for these points is started.

Route planning problem: To pick a point in each CFOV in such a way that the relay UAV manages to reach every point on the same or less time that the payload UAV reaches its corresponding points?

To solve the problem further definitions are needed. The number of points where payload has requested link coverage is denoted N. These points are denoted Point(n) n=1…N. Corresponding CFOV for each point n=1…N is denoted CFOV(n). Corresponding time for each point is denoted t(n). Point of times are denoted on a time line such that t(0)=0 at the starting point and t(n)<t(n+1) for all n=0…N

Definition 1: Point(n+1) is *reachable* from Point(n) if there are no obstacles on the straight line between the two points and the time for the payload UAV to move between two points is less or equal to the time for the relay UAV to move between its corresponding points, t(n+1)-t(n).

Definition 2: Point (n) *reaches* Point(n+1) if Point(n+1) is reachable from Point(n).

Note that points that require a detour are not considered to be reachable, even if the detour is short enough to be finished on time. This is a simplification which speeds up the calculation workload in this initial study, in future work, chapter 5, we mention how to move on with a better definition.

Definition 3: A route is *valid* if Point(n) reaches Point(n+1) for all points n = 0…N-1 in the route.

The two following obvious methods show why the problem does not have an obvious solution:

- While standing in point n, calculate reachable points in following CFOV(n+1). Pick a point in CFOV(n+1) and calculate reachable points in the successive CFOV(n+2). n=1…N-2. In the case that the point we pick is close to an obstacle, we can not guarantee that any point in the following CFOV is reachable. In **Fig 3** we see that CFOV(2) is not reachable from the chosen point in CFOV(1) due to the obstacle.

The problem occurs with regression as well:

- While standing in point N, the last pointing the sequence, calculate reachable points in the previous CFOV(N-1). Pick a point in CFOV(N-1) and calculate reachable points in the preceding CFOV(N-2). **Fig 4** shows that CFOV(1) is not reachable from the chosen point in CFOV(2) due to the obstacle.

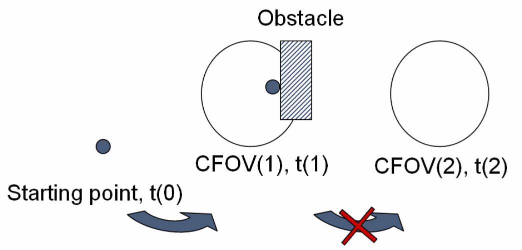

Fig 3. Progression from starting point towards increasing number of CFOV. Problem when no valid route between chosen point and successive CFOV exist.

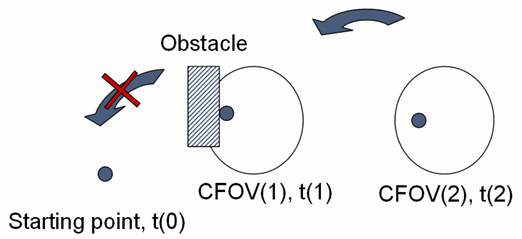

Fig 4. Regression from CFOV(N) towards decreasing number of CFOV. Problem when no valid route between chosen point and preceding CFOV exist.

The conclusion is that before a point is chosen it must be validated that it is reachable from the previous point *and* that it reaches at least one point in the following CFOV. To prevent dead ends later on in the route it is not enough to validate only one CFOV ahead when choosing point, it must be validated that all following CFOVS are reachable. Otherwise the problem is just temporarily ignored, but may occur again later on in the route.

A more sophisticated method consists of two parts: First part starts at the final CFOV(N). Calculate the set of points in CFOV(N-1) that reaches any point in CFOV(N). Call this set of points BCFOV(N-1) as Backward cropped CFOV. The set of points in CFOV(N-2) that reaches BCFOV(N-1) are called BCFOV(N-2) and the method iterates backwards towards the starting point. **Fig 5** shows that BCFOV(2) is created by calculating the set of points that reaches CFOV(3). The red circle gives a rough figure of surrounding points that reaches CFOV(3). When obstacles are intersecting the path between CFOV(n-1) and CFOV(n), points are cut out from BCFOV(n-1) as shown in **Fig 6.**

The method has parallels to constraint programming since points not fulfilling the constraints are removed and constraints are nested on each other. This way all CFOVS are processed. There exists a valid route from start point till end point if BCFOV(1) is non-empty. If BCFOV(1) turns out to be the empty set, the payload route has to be replanned in space or/and time. Replanning the route for the payload UAV is done by the human operator and is not covered in this study.

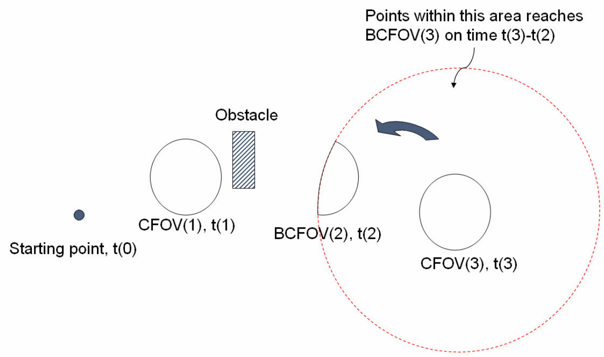

Fig 5. BCFOV(2) are calculated by sorting out points that reaches CFOV(3).

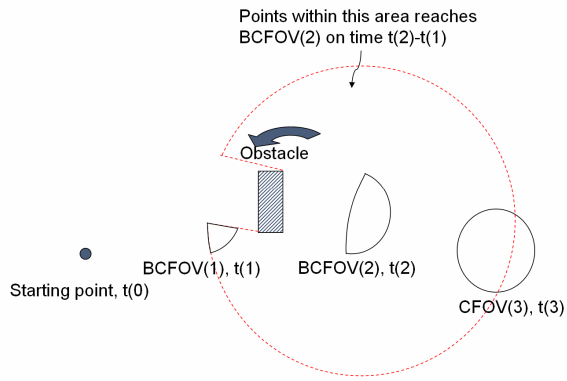

Fig 6. BCFOV(1) are calculated by sorting out points that reaches BFCFOV(2). Note that several points are left out due to the obstacle.

A drawback with the method is its complexity. To get an accurate BCFOV(n) in the worst case, each point in CFOV(n) has to be checked against each point in CFOV(n+1), which gives high complexity and long computation time. To speed up the calculation accuracy has to be balanced against time. By using fewer calculation points we get a pessimistic, over cut, estimate of BCFOV, which sometimes ends up empty when it in fact should be non-empty.

The second part of the method checks reachability before each waypoint is chosen. From the starting point we cut out not reachable points in BCFOV(1) and save the remaining points in Forward and Backward cropped CFOV, FBCFOV(1). This way we know that all points in FBCFOV is reachable and can reach all the remaining CFOVs. **Fig 7** shows how the FBCFOV(1) is calculated and influenced by the obstacle. The blue circle gives a rough figure of surrounding reachable points. In **Fig 8** a point in FBCFOV(1) is chosen.

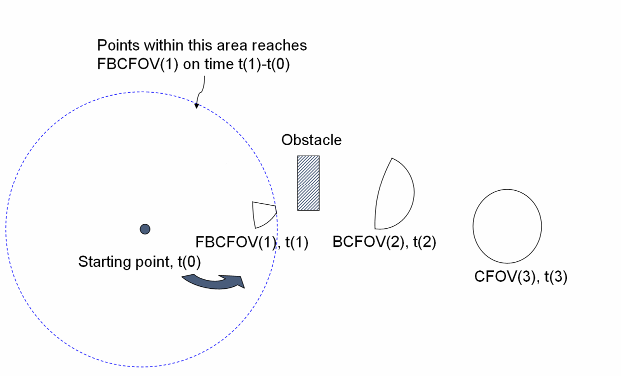

Fig 7. FBCFOV(1) is calculated to be the set of points in BCFOV(1) reachable from the starting point.

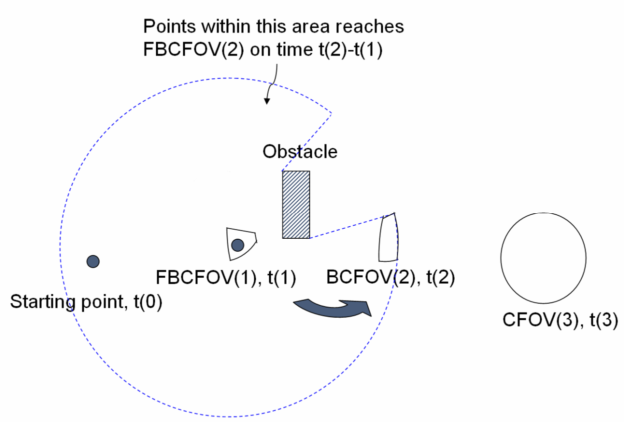

Fig 8. Waypoint has been chosen in FBCFOV(1). This enables the search of reachable points in BCFOV(2). The obstacle decrease the set of points that reaches BCFOV(2).

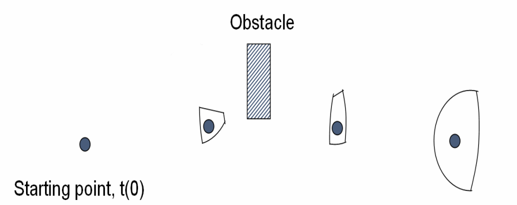

Fig 9. Final relay UAV route with three generated waypoints.

No special factors have been considered when determining the point choosing algorithm since criteria wary between different scenarios; you might want to stay at as low altitude as possible, fly as short distance as possible, use as less fuel as possible etc. In this report we chose the point in the center of FBCFOV. **Fig 9** shows the final payload UAV route with three generated waypoints.

3.1 Synchronization of Payload and Relay UAV

When planning the relay UAV route a rough payload UAV route is used together with estimates when the relay UAV will be able to attend its waypoints. Due to wind disturbances and discrepancies between the real world and modeled world, the payload UAV route might not be followed exactly and the estimates for the relay UAV will end up a bit from reality.

If no synchronization is used each UAV will try to follow its route as carefully as possible but take no actions when the other UAV is being late.

There are several ways to handle synchronization, two of them are:

- Alter the speed of the earliest UAV in order to reach corresponding waypoint at the same time.
- The earliest UAV hover (helicopter) or enter a holding orbit mode (fixed wing) at its waypoint and wait for latest UAV to reach its corresponding waypoint.

We chose the former alternative in our implementation. The UAVs continually transmits their position and estimated time of arrival in next waypoint. The UAV with the earlier time in its estimate has to slow down to get the same estimated time of arrival as the UAV with the latest estimated time. If the earlier UAV is a fixed wing it can go into a holding loop and fly by the waypoint at any given point of time.

This is of course a problem since it requires that UAVs are able to transmit information among each other though we earlier said that relay UAV only has to guarantee coverage in certain points. It is therefore the case UAVs will get unsynchronized when they are unable to communicate, and when they are able to do so, they get synchronized again.

An advantage with keeping calculated BFCFOVs in the UAVs is that when the waypoint is reached, the UAV can carry on within the BFCFOV and position itself in a way that it will reach next waypoint easier.

This way we use rough plans and update them online in each UAV. Another way of doing the synchronization is to have an execution monitor on the ground that calculates new improved routes for each UAV. If no improved routes are received from ground station the UAVs carry on with its old plans.

4 Implementation

A prototype of the algorithm was implemented in Matlab and run in our simulator environment developed in Java.

We used the following parameters for our run. The figures does not correspond to any given length unit, it can be interpreted as m, m/s and m/s^2 but the acceleration and retardation will be a bit unrealistic. Positions are given as x,y,z.

- Ground station range = 400
- Ground station position = 500, 500, 100

- Payload UAV start point = 500, 500, 50
- Payload UAV range = 300
- Payload UAV speed = 20
- Payload UAV acc = 20
- Payload UAV retardation = 5

- Relay UAV start point = 500, 500, 80
- Relay UAV range = 300
- Relay UAV speed = 3
- Relay UAV acc = 20
- Relay UAV retardation = 20

- Obstacle height = 400

The algorithm has been tested with obstacles of different heights. For the route calculated here we chose very high obstacles to force the UAVs to fly around them instead of over them.

Our test environment is shown in **Fig 10**. Yellow lines indicate the route of the payload UAV, green circles are waypoints where link coverage is required. Obstacles are marked with as red squares. Resolution in x-y-z axis is 1 unit of length. The letter G in the middle of the environment marks the position of ground station.

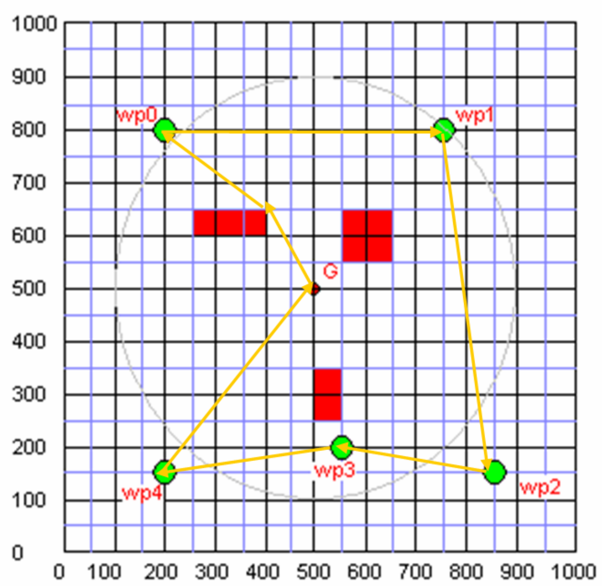

Fig 10. Overview of test environment. Yellow lines are the route of the payload UAV, green dots are waypoints where link coverage is requested. Red squares are obstacles and the letter G marks the position of the ground station. The range of the ground station is marked as the gray circle. The grid does not represent the resolution used in the simulation.

4.1 Calculate CFOVs

In Fig 11 the range of payload in each waypoint is marked with a red circle. A circle is placed, centered on the ground station to show in the area where the relay must be. The radius of this circle is the range of the relay since it is shorter than the range of the ground station.

Since the ground station is stationary it is a good idea to calculate the area in which relay is able to transmit to it. The left hand side of **Fig 12** shows the outline of this area. The right hand side shows a 3D visualization made in Matlab of the area, seen at an angle from above. In **Fig 13** the first CFOV has been calculated. We see to the left as well as to the right that the set of points is disjunctive and the algorithm has to decide in which of the two subsets to place the waypoint. The resulting relay route is shown in purple in **Fig 14**. Blue circles indicate synchronization points that correspond to green circles.

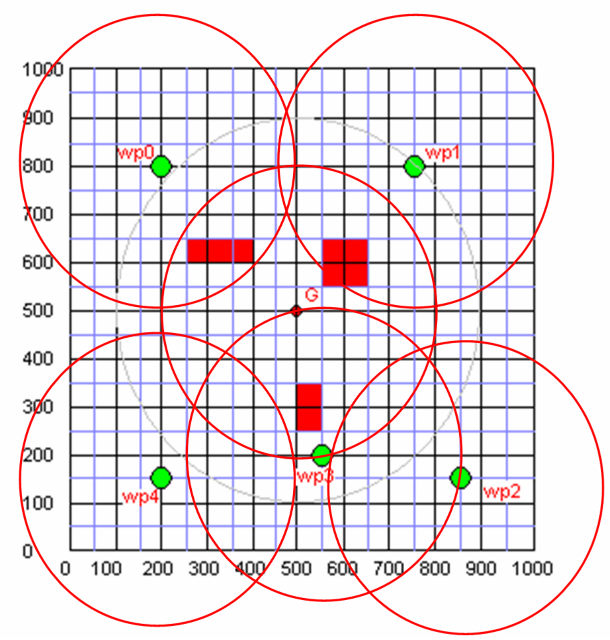

Fig 11. Transmission range is marked with red circles in payload waypoint0 to waypoint4. The inner circle, with the ‘G’ in its center, marks the range of the relay UAV placed on the ground station. Since relay UAV has shorter range than ground station, this circle is of interest instead of the range of the ground station.

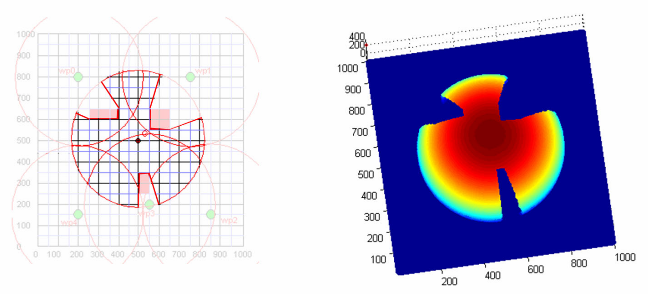

Fig 12. Left: Sketch of area for relay UAV.
Right: 3D visualization of the same area.

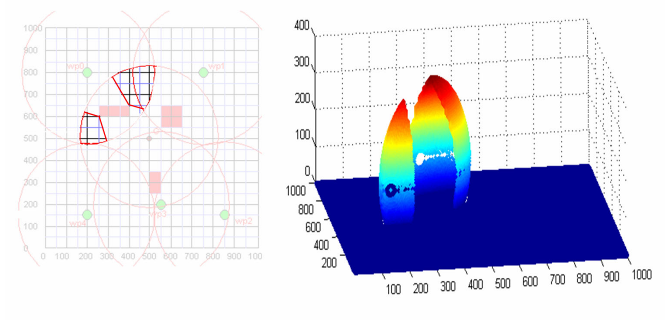

Fig 13. The first CFOV has been calculated.
Left: a sketch of the areas.
Right: A 3D visualization of the same areas. The plot is hollow since only zmin and zmax is shown..

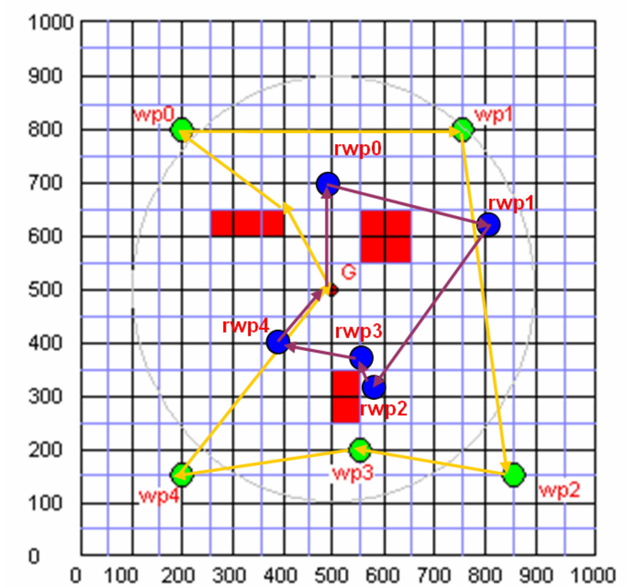

Fig 14. Final relay route planned by developed algorithm, shown in purple. Blue circles are synchronization points that correspond to the waypoints for the payload UAV, shown as green circles. Note that the relay-waypoint2 is placed close to the lower obstacle due to the simple point-choosing-algorithm we use.

4.2 Limitations

We only use two points when calculating backwards. The two points are chosen to be the center point of BCFOV(n+1) and the point in BCFOV(n+1) closest to BCFOV(n).

When choosing point in FBCFOV we go for a point close to the center of the following BCFOV. The reason for this is that we did not put focus on any criteria. It served well for our demonstration.

As seen in the definition of reachable and reaches we only investigate if the straight line between points are free from obstacles. It would be better to check whether we can fly between two points on the given time, regardless of the route. This would require a path planning algorithm to calculate paths around obstacles.

4.3 Time Consumption

Given an ordinary laptop computer of year 2006 and despite only using two points in BCFOV(n+1) for calculating BCFOV(n) the algorithm needs 30 seconds per waypoint. The use of two points gives an over cut BCFOV. A non verified estimate is that at least ten points should be used for the results to be satisfactory. However this is a balance between time and accuracy.

5 Future Work

If the algorithm is to be used in a commercial UAV all of the limitations mentioned in 4.2 have to be dealt with. What might be of more research interest is how to 1) lighten the constraints on the payload UAV, 2) distribute the calculations and 3) expand the scenario to involve several UAVs.

5.1 Payload UAV Within a Given Area

If we have a reactive payload UAV told to scan an area for objects with unknown position, it is more appropriate to give the UAV some freedom to fly anyhow within the area instead of demanding it to follow a predefined route. The relay UAV now has to guarantee link coverage for the entire area. This is can be done by simply gaining altitude. The transition between providing link coverage for points, as treated in this paper, to providing link coverage for entire areas are of interest.

5.2 Distributing Calculations

Although a UAV generally has several computers, power and battery life as well as physical constraints limits the computational power of a UAV. One possible solution is to use distributed calculations. In such a configuration, a UAV sends a request to the ground station or another UAV with a request that some calculation should be performed, together with the data needed to perform the calculations. When the calculation has been performed, the answer is sent back to the requesting UAV. Although this solves the problem with limited calculation power, it introduces problems with timing, communication delays and guaranteed delivery. Vital algorithms concerning the safety and wellbeing of the UAV may not be distributed, but some non-vital algorithms may be distributed to other parties in the same network. **Fig 15** shows an example when the payload and relay UAVs send their positions to the ground station which calculate the relay UAV's route.

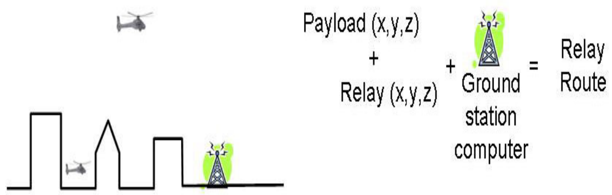

Fig 15. Payload and relay UAV send their positions to the ground station, which calculates the relay route.

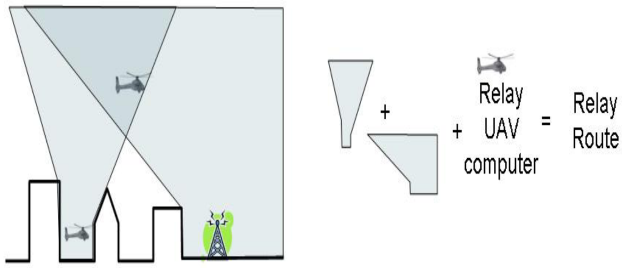

Fig 16. The payload UAV and the ground stations send their fields of view to the relay UAV, which calculates the CFOV and the route.

Another possible method is shown in **Fig 16**, where the payload UAV and the ground station send their fields of view to the relay UAV, which calculates the CFOVs and chooses the required positions, giving the route for the payload UAV.

5.3 Several UAVs

Different setups are possible if we add more UAVs to the scenario. By adding more payload UAVs we will get a quicker area scan and better situation awareness. By adding more relay UAVs we get a higher degree of link coverage. The upper part of **Fig 17** shows how several relay

UAVs are be used as a chain to extend the area of operability of the payload UAV. The lower part of **Fig 17** shows how two payload UAVs use the same relay UAV.

The effect on the algorithm by adding payload UAVs is that the common part of each UAVs field of view gets smaller. The risk of ending up with an empty BCFOV(1) (and even CFOV(1)!) is therefore higher, and we need to either coordinate the payloads or add more relay UAVs.

The algorithm presented here is not expandable to handle several relay UAVs. This is a difficult problem to solve but has some similarities to research in ad hoc networks, and this scenario will be further investigated in the future.

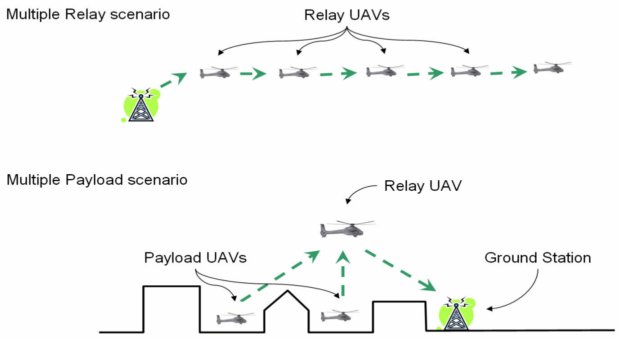

Fig 17. Upper: Several relay UAVs forming a chain to extend the working area of the payload UAV. Lower: Two payload UAVs using the same relay UAV.

6 Conclusions

This is a proof of concept for our initial algorithm of solving the relay UAV positioning and planning problem. In order to get an algorithm for use in commercial products, several refinements are needed. The scenario covers many different research areas such as representation of the environment, ray tracing, dynamic programming and synchronization between UAVs. The contribution of this report is to show how we join these diverse research areas to solve the problem of generating a valid relay UAV route as a solution to the problem.

With this report we hope to encourage other researchers to look into this area of 3D path planning under the given constraints.

Acknowledgments

Erik Skarman and Leif Axelsson at Saab Aerosystems, PhD students of Patrick Doherty at Department of Computer and Information Science at Linköping University and master students Mikael Call and Mikael Nordenroth have all contributed in the development of the algorithm. This research was funded in part by Vinnova through National Aviation Research Program NFFP, and in part by Linklab, a joint venture between SAAB AB and Linköping University.

References

[1] Doherty P, Haslum P, Heintz F, Merz T, Persson T, and Wingman B. (2004). A Distributed Architecture for Intelligent Unmanned Aerial Vehicle Experimentation. *Proceedings of the 7th International Symposium on Distributed Autonomous Robotic Systems*.

[2] Schouwenaars T, Feron E, How J. Multi-Vehicle Path Planning for Non-Line of Sight Communication. *Journal of Field Robotics*, 23(3/4), pp 269-290.

[3] Hearn D and Baker, P. *Computer Graphics with Open GL*. Prentice Hall, 2003.

[4] Lohmüller, Friedrich. http://www.f-lohmueller.de/pov_tut/csg/povcsg1e.htm

[5] Djuknic G M, Freidenfelds J and Okunev Y. Establish Wireless Communications Service Via High Altitude Aeronautical Platforms: A Concept Whose Time Has Come? *IEEE Communications Magazine*, Vol. 35, No. 9, pp 128-135.

[6] LinkLab. www.linklab.se

[7] SAAB AB. www.saabgroup.com

Copyright Statement