2015

Aerial vehicle trajectory design for spatio-temporal task satisfaction and aggregation based on utility metric

Amarender Reddy Mekala

Recommended Citation

Mekala, Amarender Reddy, "Aerial vehicle trajectory design for spatio-temporal task satisfaction and aggregation based on utility metric" (2015). *Masters Theses*. 7700.
https://scholarsmine.mst.edu/masters_theses/7700

AERIAL VEHICLE TRAJECTORY DESIGN FOR SPATIO-TEMPORAL TASK SATISFACTION AND AGGREGATION BASED ON UTILITY METRIC

by

AMARENDER REDDY MEKALA

A THESIS

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

in

COMPUTER SCIENCE

2015

Approved by

Dr. Sanjay K. Madria, Advisor
Dr. Chaman Sabharwal
Dr. Dan Lin

PUBLICATION THESIS OPTION

This thesis consists of the below article that has been submitted for publication as follows:

Paper 1. Pages 6-39 has been submitted as ”Aerial Vehicle Trajectory Design for Spatio-Temporal Task Satisfaction and Aggregation Based on Utility Metric”, In the 9th ACM International Conference on Distributed Event-Based Systems (DEBS 2015).

ABSTRACT

Flight trajectories are mainly designed in order to make sure that the aerial vehicle reaches the destination point from the start point. In addition to that, the flight path is designed in such a way that the flight avoids dangerous zones and the maneuvers required for the path are practically feasible for the flight.

This thesis focuses on the aerial vehicle trajectory generation passing through a set of pre-defined way points while satisfying the task points sent by the ground base station in between the way points. Each specified waypoint is reached exactly at the specified location and time. Intermediate waypoints are generated in such a way that they satisfy the task points, which give high benefit measure, i.e. satisfying tasks with high task priority and QoS (Quality of Service) priority. Generated waypoints are points from where imagery data of the tasks are collected. Task points are aggregated by the TABUM (Task Aggregation Based on Utility Metric) approach, which takes into consideration factors such as task points' priorities, sensory capability and deviation required from the shortest path to satisfy that waypoint.

We generate a 4D flight trajectory, which is a collection of predefined waypoints and generated waypoints by taking the velocities, maneuverability of the aerial vehicle into consideration while ensuring that the vehicle avoids the no-fly zones. We finally frame the problem of trajectory generation in a constrained environment as an optimization problem and solve it by increasing the benefit measure and decreasing the cost measure (deviation from line-of-sight path). We perform experiments to show the effect of the utility metric threshold value and compare the performance of the vehicles' trajectory with flight maneuverability and helicopter maneuverability capabilities. We also show how the performance of the sensor/camera attached to the flight will effect the benefit measure.

ACKNOWLEDGMENTS

I would like to express my gratitude to all those who have helped me with this research. First, I would like to thank my advisor Dr. Sanjay Kumar Madria who gave me the opportunity to work on this research topic. His suggestions and encouragement carried me through difficult times and his valuable feedback contributed greatly to this thesis. Secondly, I would like to thank Dr. Chaman Sabharwal and Dr. Dan Lin for being part of my thesis committee and taking time to review this work.

I cannot miss to mention my family and friends whose constant support and encouragement has always been invaluable for me. This project is partially funded by the Air Force Research Laboratory (AFRL) in Rome, New York and I would like to express my gratitude to Mark Linderman from AFRL as well.

TABLE OF CONTENTS

Page

SECTION

PAPER

SECTION

LIST OF ILLUSTRATIONS

Figure Page

PAPER I

SECTION

LIST OF TABLES

Table Page

PAPER I

1. INTRODUCTION

An aerial vehicle's path is a collection of waypoints through which the aerial vehicle must pass through in order to reach the destination point from the start point. Flight paths are generally designed and pre-defined for unmanned aerial vehicles so that they can pass through waypoints safely. Manned aerial vehicles have a possibility to change the path during the journey depending upon various factors like maneuvering, weather conditions, no-fly zones etc.

In our paper, we try to satisfy the task points which lie near the path of the flight when it is flying through the pre-defined waypoints. We modify the trajectory which deviates from the shortest line-of-sight path so that it can take images of the task points which fall under the scope of the camera attached to the flight. So, an efficient trajectory design approach is needed in order to satisfy maximum number of task points while passing through the pre-defined waypoints. We provided a trajectory design approach in order to solve the problem of task satisfaction by taking the real-time constraints into consideration such as aerial vehicle maneuverability, no-fly zone constraints, time deadline constraints. We try to satisfy the task points with high priority while minimizing the total distance travelled by the vehicle which leads to minimum fuel consumption.

The common applications of the trajectory design and its optimization are in the field of aerospace navigation where the aerial vehicle may be manned or unmanned (robotic). It is also applicable in applications related to the collection of imagery data from aerial vehicles and military applications like battlefield surveillance. They are also used for spying, monitoring purposes, architectural mapping and beneficial to dropping or picking up light-weighted items.

1.1. MOTIVATION

Many papers have been published to generate the aerial vehicle trajectory passing through a given set of pre-defined waypoints. However, they did not address the problem of generating new waypoints in between the pre-defined waypoints in case there are few tasks to be satisfied when the vehicle is moving from one waypoint to other. So, we address the aspect of task satisfaction while passing through the specified waypoints based on the task's priority and extra fuel consumed when deviating from the original path. This is the main motivation for this work. As part of this thesis work, we addressed this issue by aggregating the task points based on swath width (scope of camera) and satisfying the task points which fall in this swath width. While doing this, we also consider other real-time constraints such as no-fly zone avoidance, camera/sensory capability, aerial vehicle maneuvering capabilities (helicopter maneuvering and aircraft maneuvering).

1.2. ORGANIZATION OF THESIS

The rest of the thesis is organized as follows. Section 2 is the related work section where the papers designed the aerial vehicle trajectories based on specified waypoints. Section 2.4 is the submitted research paper in which the Task Aggregation Based on Utility Metric (TABUM) approach is proposed which is the core part of research. It also contains the implementation details of the real-time constraints discussed above. Section 3 contains the step-by-step approach of the technical implementation and the experimental results for the simulation. It also contains the screenshots of the simulation tool developed to perform experiments and validate our approach. Finally, Section 4 concludes the thesis.

2. RELATED WORK

A lot of research has been done on the flight trajectory design in the past few years. Most of the papers addressed the issue of flight trajectory design passing through specified way points while taking a safe path away from the obstacles and by considering the kinematic constraints of the vehicle.

2.1. 4D TRAJECTORY GENERATION AND TRACKING FOR WAYPOINT-BASED AERIAL NAVIGATION

The basic operation for any unmanned aerial vehicles is the autonomous navigation where the vehicle passes through a given set of waypoints or along a predefined trajectory. Bousson and Machado [1] proposed a method for generation of the 4D navigation trajectories and tracking of unmanned aerial vehicles passing through given sequence of waypoints with time constraints at each of the waypoints. They used a twofold approach, where they used the pseudo-spectral (variable representation methods) based trajectory optimization method to generate an optimal continuous trajectory, which passes through a sequence of waypoints, and then a trajectory with minimum deviation is generated using a predictive control law. The complexity of the computations required to generate the trajectory in this approach is low because they neither used slack variables to deal with waypoints nor included the flight dynamics equations. In this work the authors described a trajectory generation method based on chebyshev pseudo-spectral approximation over the former methods like collocation methods which produce much smoother curves. A one-step ahead predictive control method was presented and used for controlling the aircraft along the trajectories that were generated which is simple and robust and lends itself to real time control.

2.2. OPTIMUM FLIGHT PATH DESIGN PASSING THROUGH WAYPOINTS FOR AUTONOMOUS FLIGHT CONTROL SYSTEM

In this paper [2], the authors proposed a trajectory generation method which passes through specified waypoints. They have taken an auxiliary variable to deal with the unspecified time for passing through the waypoint. They also do not have a strict condition that the flight must pass through the specified waypoint exactly. So the flight can pass within an acceptable range from the specified waypoint. The problem is then solved by using sequential quadratic programming (non-linear optimization method) algorithm. The trajectory optimization problem passing through the given waypoints has the equality constraint equations on the state variables at the unspecified time, as well as inequality constraint equations on the control inputs. The considered trajectory optimization problem has several unspecified intermediate times because the passing time of each waypoint is not specified.

2.3. RADAR-ASSISTED COLLISION AVOIDANCE/GUIDANCE STRATEGY FOR PLANAR FLIGHT

Ajith and Ghose [3] proposed a radar-assisted collision avoidance/guidance strategy (RACAGS) for planar flight, which generates a 3D navigation trajectory using the RACAGS algorithm. The trajectory ensures that it avoids the collision with minimum possible deviation from the original path. They mainly addressed the Nap-of-the-Earth flight (NOE) model where it is carried out close to the surface of the earth with the distinguishing feature that mainly uses lateral maneuvers to avoid obstacles. The collision avoidance system is part of the guidance system which generates maneuver commands necessary to guide the vehicle towards its goal while avoiding obstacles on its path. In this paper, the collision avoidance problem deals with the near-field mission where it treats the nominal trajectory as reference input to the innermost guidance loop and avoids obstacles by using information provided by the obstacle-detection system. In this approach, the radar collects all the information about the obstacle and generates two acceleration commands- the guidance command and the avoidance command – to trace a collision-free trajectory from the start position to a goal position. They also dealt with the construction of the obstacles virtually and designed the three-dimensional collision avoidance/guidance algorithm.

2.4. ADAPTIVE PATH PLANNING FOR UNMANNED AERIAL VEHICLES BASED ON BI-LEVEL PROGRAMMING AND VARIABLE PLANNING TIME INTERVAL

In this paper [4], the authors present an adaptive path planner for unmanned aerial vehicles (UAVs) to adapt a real-time path search procedure to variations and fluctuations of UAVs' relevant performances, with respect to sensory capability, maneuverability, and flight velocity limit. They introduced bi-level programming and variable planning step techniques to model the path planning components and then an adaptive path planner is developed for purpose of adaptation and optimization. Moreover, they used obstacle avoidance and performance limits as path search constraints to guarantee path safety and navigability. Using the bi-level programming, the approach generates waypoints and control inputs at variable planning time intervals. It can address the performance variations and adaptively plan smooth flight paths only when necessary. It also takes the flight maneuvering capabilities into consideration and checks whether it is feasible for the flight to reach the next waypoint.

PAPER

I. AERIAL VEHICLE TRAJECTORY DESIGN FOR SPATIO-TEMPORAL TASK SATISFACTION AND AGGREGATION BASED ON UTILITY METRIC

ABSTRACT

An optimal design method for generating a flight path passing through specified waypoints while satisfying the task points sent by the ground base station is proposed. Each specified waypoint is reached exactly at the specified location and time while intermediate waypoints are generated in such a way that they satisfy the task points, which give high benefit measure, i.e. satisfying tasks with high task priority and QoS priority. Generated waypoints are points from where imagery data of the tasks are collected. Task points are aggregated by the TABUM (Task Aggregation Based on Utility Metric) approach, which takes into consideration factors such as task points' priorities, sensory capability and deviation required from the shortest path to satisfy that waypoint. We generate a 4D flight trajectory, which is a collection of predefined waypoints and generated waypoints by taking the velocities, maneuverability of the aerial vehicle into consideration while ensuring that the vehicle avoids the no-fly zones. We finally frame the problem of trajectory generation in a constrained environment as an optimization problem and solve it by increasing the benefit measure and decreasing the cost measure (deviation). We perform experiments to show the effect of the utility metric threshold value and compare the performance of the vehicles' trajectory with flight maneuverability and quad copter maneuverability capabilities.

1.1. INTRODUCTION

Flight trajectory design of the aerial vehicles is done for Unmanned Aerial Vehicles (UAVs) to fly from one point to other point while passing through intermediate waypoints. The generation of waypoints without the consideration of the dynamic constraints of the vehicle is easy but is not practically feasible for the vehicle to pass along those waypoints. Usually, aircrafts face difficulty in passing through close neighboring points due to its limited maneuverability such as minimum radius of turn, maximum climb gradient, minimum stability required for the flight etc. Also the flight trajectories must be designed in such a way that it takes factors such as fuel consumption, arrival delay, distance travelled, no-fly zones into consideration. The common applications of the trajectory design and its optimization are in the field of aerospace navigation where the aerial vehicle may be manned or unmanned (robotic). It is also applicable in applications related to the collection of imagery data from aerial vehicles and military applications like battlefield surveillance. They are also used for spying, monitoring purposes, architectural mapping and beneficial to drop or pick up light weighted items. Figure 1.1 shows a flight passing through waypoints and satisfying task points (collecting imagery data) located near the path. Bousson and Machado [1] proposed a method for generation of the 4D navigation trajectories and tracking of unmanned aerial vehicles passing through given sequence of waypoints with time constraints at each of the waypoints. They used a twofold approach, where they used the pseudo-spectral based trajectory optimization method to generate an optimal continuous trajectory, which passes through a sequence of waypoints, and then a trajectory with minimum deviation is generated using a predictive control law.

In [2], the condition for the flight to pass through a set of specified waypoints is relaxed such that the vehicle passes through a certain acceptable range from the waypoints and time at which the flight must pass through the waypoint is unspecified, due to which the accuracy of the temporal dimension of the waypoint is compromised. Ajith and Ghose [3] proposed a radar-assisted collision avoidance/guidance strategy (RACAGS) for planar flight, which generates a 3D navigation trajectory using the RACAGS algorithm. The trajectory ensures that it avoids the collision with minimum possible deviation from the original path. In [5], a real–time dynamic trajectory smoothing method for aerial vehicles that satisfies the kinematic input constraints of the aerial vehicle is proposed. It results in shortest possible path (line-of-sight path which is a straight line passing through the waypoints) while passing through waypoints.

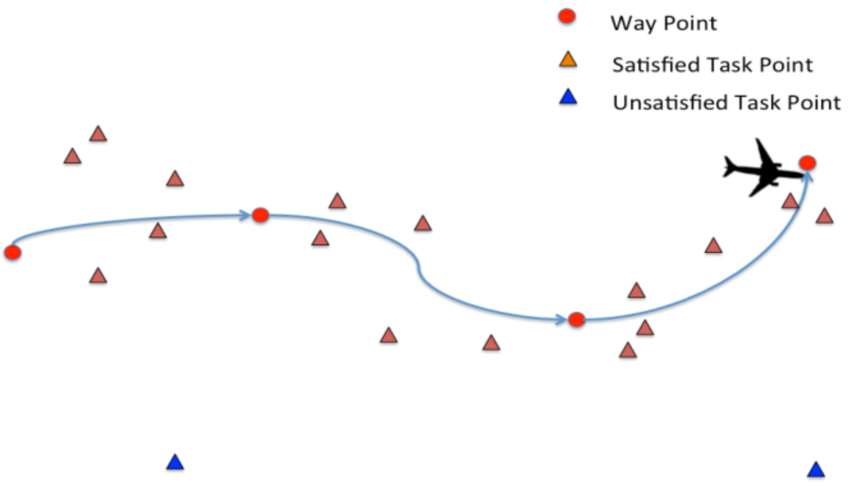


Figure 1.1. Flight passing through waypoints and satisfying task points

In this paper [4], the authors present an adaptive path planner for unmanned aerial vehicles (UAVs) to adapt a real-time path search procedure to variations and fluctuations of UAVs' relevant performances, with respect to sensory capability, maneuverability, and flight velocity limit. They introduced bi-level programming and variable planning step techniques to model the path planning components and then an adaptive path planner is developed for purpose of adaptation and optimization. Moreover, they used obstacle avoidance and performance limits as path search constraints to guarantee path safety and navigability. The flight trajectory design is converted into a parameterized optimization problem where the UAV kinematic constraints is one of the parameter while trying to constrain the problem by avoiding collision zones [6].

In [7], they proposed a method for generating multiple possible paths between the waypoints while avoiding the threat zones and the user will have the access to choose any of the paths according to user's priorities. In [8], the authors proposed a trajectory planning approach which passes through specified waypoints by taking into account the maneuverability of air vehicle and forbidden zones. The model is based on conversion of the airspace into a network which contains feasible trajectories. Every path in the network represents a flyable trajectory so that trajectory is within the performance limits of the air vehicle and the generation of network is done non-deterministically.

We try to approach the problem of flight navigation trajectory design passing through the waypoints while taking the maneuverability, no-fly zones avoidance constraints into consideration and also satisfying the tasks assigned by the user (Ground Base Station) using TABUM (Task Aggregation Based on Utility Metric) approach. One of the major differences between our approach and those specified in the literature is the emphasis we give on the strategic aspect of satisfying the task points specified by the user rather than merely focusing on the trajectory aspects i.e. passing through specified waypoints. The other major difference is that we address the problem of satisfying the task points using a benefit function and cost function while taking the no-fly zone avoidance, dynamic characteristics of the vehicle, sensory capability into consideration.

We design an optimal trajectory which is least deviated from the original shortest path (line-of-sight path between the waypoints) resulting in less total distance travelled, less fuel consumption and leads to less arrival delay. In our scenario, we have task points to satisfy by taking photographs of those points by flying over them. New waypoints are generated in such a way that the flight can cover the task points with maximum priority (task point numerical weight assigned by the user) with minimum deviation from the original path. Our new approach TABUM aggregates the task points specified by the user. We assume in our approach that flight must not take loops to satisfy the waypoints. In the remaining paper, we have used the terms aerial vehicle and vehicle interchangeably.

The paper is organized as follows: section 1.2 states the nomenclature used in the rest of the paper. Section 1.3 states the problem statement we are trying to address in a formal manner. Section 1.4 explains the flight path design using the TABUM approach and considering all the constraints. The implementation procedure and algorithms are discussed in Section 1.5. Section 1.7 describes the entire trajectory generation problem as an optimization problem. The experiments section contains the experiments performed by taking a sample dataset to check the performance of the vehicle's trajectory with different maneuvers and different camera capabilities at different utility metric threshold values.

1.2. NOMENCLATURE

The basic terminology used in the paper is described below:

1.2.1. Footprint Diameter. The area on the ground, which can be covered by the camera attached to an aerial vehicle at any given instance of time. This area is generally circular in shape or rectangular. The size of this area is known as the footprint diameter, which varies for each camera. This area is also called as the Instantaneous Field of View (IFOV) of the camera. The circles shown in figure 1.2 are the footprints of the camera.

1.2.2. Swath Width. The strip of the earth's surface, which can be covered by the lateral movement of a camera attached to an aerial vehicle. This entire region is called the Field of View (FOV) of the camera. This swath width is basically a collection of footprints captured by the camera. Figure 1.2 explains about the swath width and footprint.

As you can see in the figure 1.2 that footprints are small at the center and gets larger while moving to the horizon due to which the quality of images at the horizon is low. So the quality of images are high only if the task points lie within the QoS region and images have a low quality if the task points lie near the horizon. We assumed that the flight could take images perpendicular to the flight path only and there is only one camera attached to the flight.

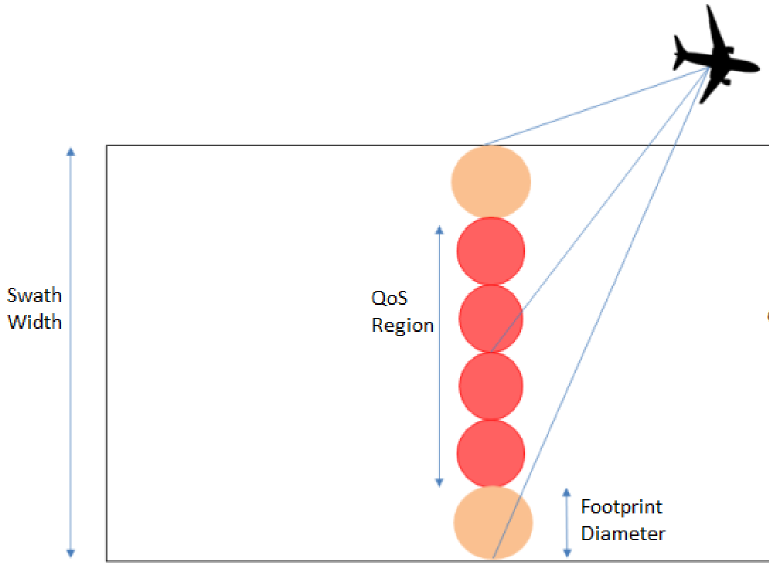


Figure 1.2. Swath width and Footprint of the flight camera

This swath width varies with the altitude of the flight. We can calculate the swath width, which depends on the flight's altitude and FOV angle of the camera, which is specific for each camera. We can derive the formula from figure 1.3 as shown in Eq. 1.1. Swath is a 2D area on the ground as shown below.

$$SwathWidth = 2 * height * \tan(\theta/2) \tag{1.1}$$

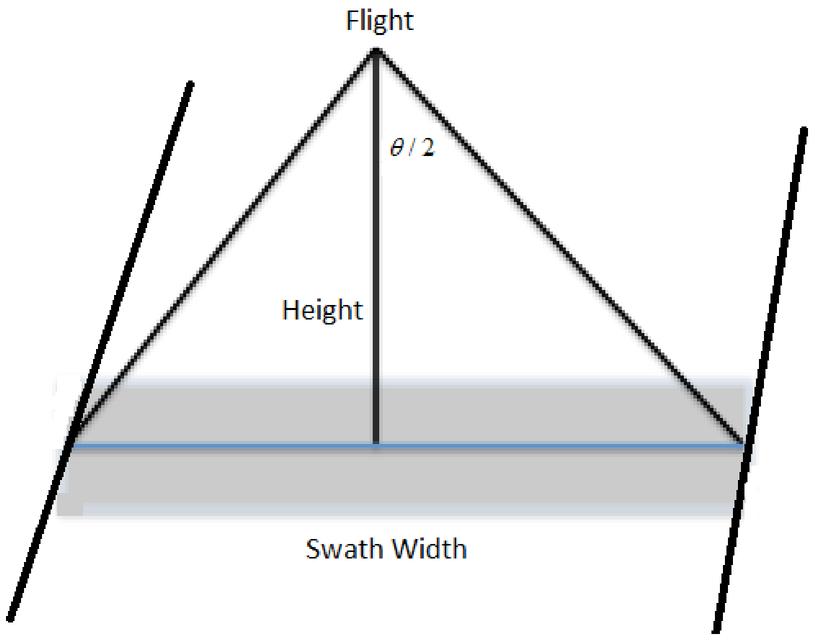


Figure 1.3. Calculation of Swath Width

where, height = flight altitude, θ = camera scan angle

1.3. PROBLEM STATEMENT

Given pre-defined waypoints and task points, we need to generate a trajectory passing through all the pre-defined waypoints exactly at given position and time and cover as many task points as possible while travelling in the shortest path possible. We need to generate a new set of intermediate waypoints from where images must be taken by taking all the following factors in to consideration:

1. Total Distance Travelled (Fuel consumption)
2. Benefit Measure (Satisfied task's priorities sum)
3. Navigational Constraints (Maneuverability)
4. Avoiding specific dangerous zones (No-fly zones)

5. Utility Metric Threshold Value

Finally, we would be framing the problem of generating the flight path passing through the pre-defined waypoints and the newly generated waypoints as an optimization problem.

1.4. OUR APPROACH (TASK AGGREGATION BASED ON UTILITY METRIC)

Task aggregation is most important step in the process of designing the aerial vehicle's trajectory. All the task points that fall in the swath width of the sensor camera are aggregated as one waypoint from where the image/video is taken. We assume the vehicle cannot turn back or take loops during the journey, due to which we need to select the best set of task points that can be covered from a point even by deviating from the line-of-sight path through the pre-defined waypoints. We need to select a waypoint in such a way that the point deviates from the shortest path only if it satisfies the task points whose sum of priority weights is worth the cost of deviation.

To ensure whether the flight deviates from the path only if the task points satisfied are worth enough, we introduce a metric called Utility Metric. We then ask the user or flight agent to set a utility metric threshold value. Only if the utility metric of the intermediate waypoint (deviated point) is greater than the threshold value, we try to reach that point. Also the new waypoint selected must not be located in the no-fly zones.

1.4.1. Input. Let $W = \{W_1, W_2, W_3..., W_i, ...W_n\}$ be the set of 'n' pre-defined waypoints through which the flight must pass at any cost and $T = \{T_1, T_2, T_3..., T_i, ...T_m\}$ be the set of 'm' task points which must be satisfied by taking an image or video. Each task has a different task priority, QoS (Quality of Service) priority and time priority specified by the user (requestor). Waypoints are given in the format as shown in Eq. 1.2.

$$W_i = [id, \alpha, \beta, \Pi, t] \tag{1.2}$$

where, id = waypoint id, α = latitude, β = longitude, Π = altitude, t = timestamp.Task points are given in the format as shown in Eq. 1.3:

$$T_i = [id, \alpha, \beta, TP, QP, TiP, TiD, t, Req] \tag{1.3}$$

where, id = task id, α = latitude, β = longitude, TP = task priority, QP = QoS Priority, TiP = Time Priority, TiD = Time Deadline, t = Timestamp ,Req = Type of Request (Image or Video) / Feedback Desired. Let $C = \{C_1, C_2, C_3, ...C_i, ...C_p\}$ be a set of 'p' no-fly zones provided to the flight by user or the flight agent. Each no-fly zone is represented as shown in Eq. 1.4

$$C_i = [id, \alpha, \beta, radius] \tag{1.4}$$

where, id = no-fly zone, α = latitude of zone center, β = longitude of zone center, radius = radius of no fly zone. We also get the input of the sensor capabilities as swath width or camera scan angle.

Figure 1.4 shows few of the possible swath widths from which an image can be taken in order to satisfy the task points. The swath width with highest benefit measure is selected based on the algorithm specified above.

1.4.2. Utility Metric. Utility metric is the calculated as the ratio of the benefit measure to the cost measure for a given waypoint. The algorithm and procedure for calculating the utility metric is given below:

Alg. 1 Utility Metric Calculation Algorithm

Input: Waypoint W_i; Task points Set in that swath width T ;
All no-fly zones Region C
Output: Returns highest utility measure for W_i

```
1    begin
2    maxUtilityMeasure = 0;
3        for each swathWidth S in AllPossibleSwathWidths
4            benefitMeasure = 0, costMeasure = 0, utilityMeasure = 0;
5            if center of swath width $\notin$ C
6                for each task point t $\in$ T
7                    if t $\in$ S
8                        if t $\in$ QoS region of S
9                            benefitMeasure += TP * QP;
10                       else
11                           benefitMeasure += TP;
12           horizontalDeviation = perpendicular distance between waypoint
13                                          and line-of-sight path
14           costMeasure = horizontalDeviation
15           utilityMeasure = benefitMeasure / costMeasure
16           if (utilityMeasure $\geq$ maxUtilityMeasure)
17               maxUtilityMeasure = utilityMeasure;
18   return maxUtilityMeasure;
19   end
```

1. Calculating Benefit Measure: For every possible swath width (figure 1.4) utility measure is calculated as in lines 3-15. Line 5 explains that the waypoint must not lie in no-fly zone. Lines 7,8 explain that if the task point lies within the swath width then check whether the task point lies in the QoS region of the swath width. If yes, product of task priority and QoS priority is added to the benefit measure as shown in Line 9 or else only task priority is added as shown in Line 11.

   Let TP (task priority) be High, Med, Low and QP (QoS priority) values be given as high, med, low. But the QP value for each task priority is different and can

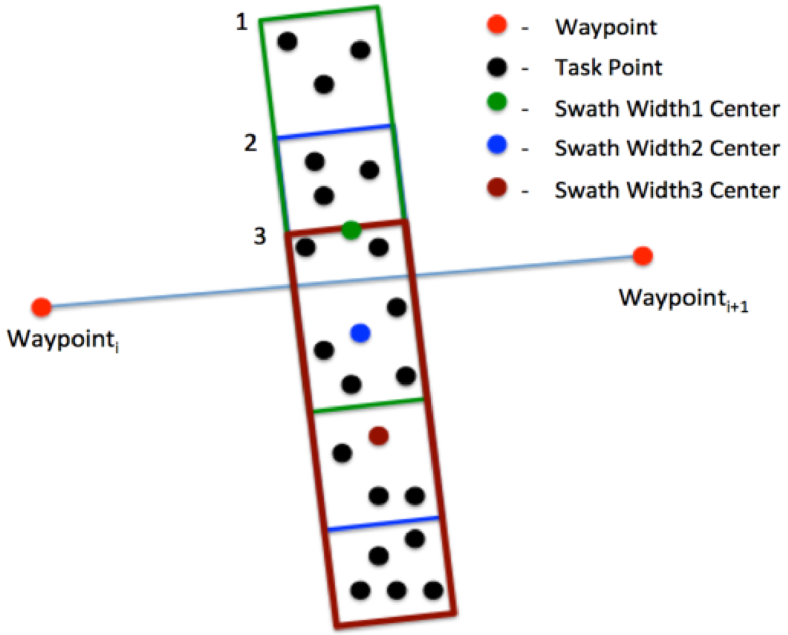


Figure 1.4. Swath widths with different possible combinations of task points

be shown like the QP matrix in Eq. 1.5 where each row is the QP values for task priority H, M, L starting from above respectively.

$$TPMatrix = \begin{bmatrix} H \\ M \\ L \end{bmatrix} QPMatrix = \begin{bmatrix} h' & m' & l' \\ h'' & m'' & l'' \\ h''' & m''' & l''' \end{bmatrix} \tag{1.5}$$

Priority weights are calculated as shown in Table 1.1.

So, in line 9 of the algorithm QP doesn't mean the weight specified by the value of the QP, but it rather depends on the value of the task priority as mentioned in the above table. For example, if a task point has QP as h and TP as M its value should be equal to the value of Mh" position in the table. The values in the table can be given by the user, pilot or may be constant, depending upon the application and the mission.

Table 1.1. Priority Weight Calculation Matrix

	High QoS Priority (h)	Med QoS Priority (m)	Low QoS Priority (l)
High Task Priority (H)	Hh′	Hm′	Hl′
Med Task Priority (M)	Mh″	Mm″	Ml″
Low Task Priority (L)	Lh‴	Lm‴	Ll‴

2. Calculating Cost Measure: Cost measure is the total deviation required for the waypoint to be satisfied with specific location and altitude. The deviation is calculated as the perpendicular distance of the waypoint from the line-of-sight path. Figure 1.5 shows the deviation of the waypoint from the line-sight path.

3. Calculating Utility Metric Value: If the flight needs to pass through the waypoint which is deviated from the original flight path by some distance as shown in figure 1.5, we calculate the utility metric as shown in Eq. 1.6.

$$UM = \frac{\sum_{i=1}^{6} TP_i * QP_i}{Deviation} \tag{1.6}$$

where, TP_i and QP_i are i'th task's Task priority and QoS priority respectively.

According to the algorithm, we add only TP_i if the task point doesn't lie in the QoS region of the swath width or else we add product of TP_i and QP_i. So the utility metric of the generated waypoint in figure 1.5 would be calculated as shown in Eq. 1.7. As task point 1 and 6 do not lie in the QoS region, we add only TP_1 and TP_6.

$$UM = \frac{TP_1 + TP_2 * QP_2 + TP_3 * QP_3 + TP_4 * QP_4 + TP_5 * QP_5 + TP_6}{Deviation} \tag{1.7}$$

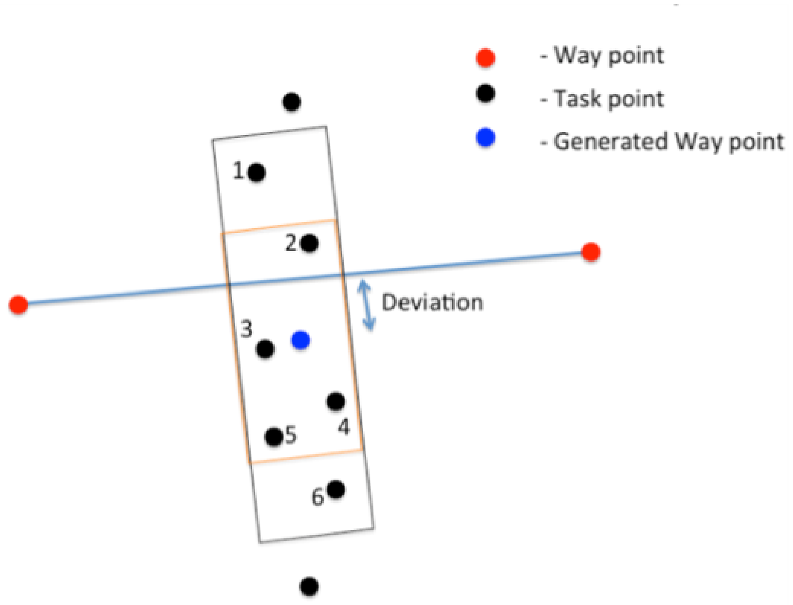


Figure 1.5. Utility Metric Calculations

After calculating the UM value at the points from which we get different combination of task points i.e. different swath widths, the point which gives the highest benefit measure (numerator in above formula) is selected, if the UM value of that point is above the threshold value specified by the user. If needed by the flight agent or the user, we can specify a maximum deviation threshold which means even if the UM value of a point is high enough resulting in high benefit measure, that point will not be selected as intermediate waypoint if large deviation from the original path is required. We have also taken care of task points sent from the ground base station when the flight is on the go. So any user from the ground can send task points dynamically and we use the same utility metric to check its value.

For calculation of the utility metric, we take the values of the TP matrix and QP matrix dynamically from the user as shown in Eq. 1.8 and Eq. 1.9. In the simulation, we have used the following sample values for the matrices.

TP Matrix =

$$\begin{bmatrix} H \\ M \\ L \end{bmatrix} = \begin{bmatrix} 9 \\ 4 \\ 1 \end{bmatrix} \tag{1.8}$$

QP Matrix =

$$\begin{bmatrix} h' & m' & l' \\ h'' & m'' & l'' \\ h''' & m''' & l''' \end{bmatrix} = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 1.5 & 1 \\ 3 & 2 & 1 \end{bmatrix} \tag{1.9}$$

So, if a high task priority with high QoS priority is satisfied by the flight, the benefit measure would be 27 (9 * 3), similarly if a med task priority with med QoS priority is satisfied, benefit measure would be 6 (4 * 1.5).

1.5. IMPLEMENTATION AND ALGORITHMS

We select the waypoints in such a way that task points with high priorities are satisfied based on the utility metric threshold value (TABUM approach). After generating the waypoints as discussed in above section, we try to calculate the time dimension of the waypoints using the task's time priority, time deadline and time stamp to calculate the 4D waypoint.

1.5.1. Waypoint Time Calculation. We choose the timestamp of the new waypoint in such a way that most of the tasks' time constraints are satisfied and is feasible for flight to reach at that time considering the velocity and dynamic constraints of the vehicle. There can be three possible values for the time deadline (TiD) field which are specified below:

1. GE (Greater than or equal): Time priority of the task is satisfied when the image is taken on or after the timestamp specified.

2. LE (Less than or equal): Time priority of the task is satisfied when the image is taken on or before the timestamp specified.

3. E (Equal): Time priority of the task is satisfied only when the image is taken exactly at the timestamp specified.

If no task point's time priority is satisfied or if the flight can't reach that waypoint at that specific time, we take the time weight sum of that waypoint as zero and try to reach that waypoint without any time constraint. We need to satisfy the waypoint at selected timestamp only if the vehicle can reach that point with its velocity constraints and also only if the pre-defined waypoints can be satisfied at exact time even after satisfying the generated waypoints at particular time stamp. We calculate the time benefit measure in the same way as benefit measure as shown in Eq. 1.10.

$$Time\,benefit\,measure = \sum_{i=1}^{n} TP_i * TiP_i \tag{1.10}$$

where, TP_i is task priority and TiP_i is task time priority

The timestamps that gives the highest sum of time benefit measure for all the waypoints are selected for each waypoint.

Lines 3-5 explain that for each task point that can be covered from the waypoint, we select that as the waypoint time. Lines 8-13 explain the calculation of the time benefit measure from the formula shown in Eq. 1.10. To calculate the time benefit measure we also take the time deadline into consideration.

For example, we calculate the time benefit measure of the waypoint W by selecting the time of task point T1. Let tasks T1, T2, T3, T4 be satisfied by a waypoint W. Let the values of task priority, time priority, time deadline and timestamp for tasks T1, T2, T3, and T4 be as follows respectively.

T1 – HIGH, HIGH, GE, 9:15:00

Alg. 2 Waypoint Timestamp Calculation Algorithm

Input: Waypoint W_i; Task points Set in that swath width T
Output: Returns highest time benefit measure for W_i

```
begin
maxTimeBenefitMeasure = 0;
    for each task point t ∈ T
        timeBenefitMeasure = 0;
        currTime = $t_{time}$ ;
        if vehicle can reach waypoint at currTime
            for each task point t ∈ T
                if ( ($t_{TiD}$ = GE and currTime ≥ $t_{time}$) or
                    ($t_{TiD}$ = LE and currTime ≤ $t_{time}$) or
                    ($t_{TiD}$ = E and currTime = $t_{time}$))
                        timeBenefitMeasure += $t_{TP}$ * $t_{TiP}$
        if (timeBenefitMeasure ≥ maxTimeBenefitMeasure)
            maxTimeBenefitMeasure = timeBenefitMeasure
return maxTimeBenefitMeasure;
end
```

T2 – MED, LOW, E, 9:18:00

T3 – HIGH, LOW, LE, 9:22:00

T4 – LOW, HIGH, GE, 9:30:00

Taking an image any time on or after 9:15:00 would satisfy task Point T1's time priority. Taking an image exactly at 9:18:00 would satisfy T2's time priority and so on. So, from the algorithm we can calculate the time benefit measure of waypoint W if the image is taken at 9:15:00 as follows. Time Benefit Measure = $TP_1 * TiP_1 + TP_3 * TiP_3$ = 9 * 3 + 9 * 1 = 36. We then choose the way point time which gives the highest time benefit measure.

1.5.2. Video Monitoring Tasks. We may have the input of the task points from the user which request for video surveillance of a specific region for a certain amount of time. Below figure explains to calculate how long a point can be under video surveillance. Let be the angle, which the sensor camera can move, in forward, backward direction i.e. along the path of the flight. So the total distance that a point can be under the camera surveillance can be calculated as L1 + L2 +L3 from figure 1.6.

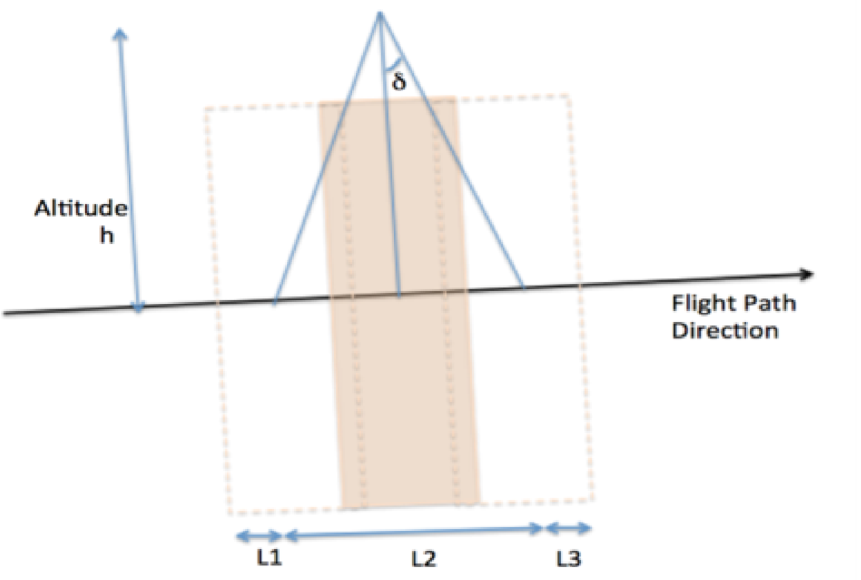


Figure 1.6. Maximum video surveillance calculation time

Total time (Time) a point can be under the surveillance of the camera is calculated as shown in Eq. 1.11.

$$Time = (L1 + L2 + L3)/V_{min} \tag{1.11}$$

where, L1=L3= footprint diameter / 2, L2 = 2 * h * tan δ, V_{min} = minimum velocity attainable by flight, δ = camera forward-backward movement angle.

Here, δ is not the camera scan angle but the angle by which it can move forward and backward. Then, we find the points (start and end) on the path of the flight between which video of the task point must be taken. This is done only if the video task priority is high than the sum of the neglected task priorities and we check that the least task priorities sum is neglected when video task point is satisfied.

From the Alg. 3, we calculate the start and end point between which videos must be taken in order to satisfy a video task point if the weight of the neglected task points is less than the video task point priority. For example, if task points T1, T2, T3, T4 fall in the region where the flight needs to capture the video of the task point T5 by focusing only on T5. Then, T5 is satisfied only if the task priority of T5 is more than the sum of the task priorities of T1, T2, T3 and T4. By checking this condition, we achieve high benefit measure by satisfying tasks with higher priority.

1.5.3. No-Fly Zones Avoidance. There may be no-fly zones in the path of the flight, which is passing through all the pre-defined waypoints and the newly generated waypoints. We need to design the flight path in such a way that the flight does not pass through these regions but passes through the waypoints. We get the input of the no-fly zones as a point (latitude, longitude) and a radius, which would be a circular region around that point in the format of (latitude, longitude, radius-of-no-fly-zone). When we take the input of the no-fly zones, each region may interfere with the path in the two possible cases described below:

1. No-Fly Zone in the path between the waypoints: If the no-fly zone is interfering with the path between any of the two waypoints, the vehicle's path is designed to avoid the no-fly zone and reach the next waypoint. If the vehicle cannot reach the next waypoint due to distance of the next waypoint from the no-fly zone or dynamic constraints of the aerial vehicle, next better waypoint is calculated from where by taking an image will satisfy tasks, which account for next best benefit measure. If the waypoint can be reached, then the path remains the same except that it deviates away from the no-fly zone. In Section 1.6, we will see how avoiding the no-fly zone is taken care and its effects on the total benefit

Alg. 3 Video Monitoring Task Points Algorithm

Input: Waypoint W; Task points T, Max Surveillance Time,
Vehicle Min Velocity (V_{min})
Output: Returns true if video of the task point can be taken

```
begin
possible = false;
    if videoTimeInterval-of-videoPoint $\leq$ Max Surveillance Time
        minDistance(D) = $V_{min}$ * videoTimeInterval-of-videoPoint;
        for each $W_i$ between L1 and L3
            benefitNeglected = 0;
            startPoint(SP) = $W_i$;
            endPoint(EP) = $W_i$ + D;
            sum = sum of task priorities between SP and EP;
            if (sum $\leq TP_{video}$ and sum $\leq$ benefitNeglected)
                benefitNeglected = sum;
                possible = true;
return possible;
end
```

measure achieved. Figure 1.7 shows a basic example of how flights deviate from the no-fly zone. r_{min} is the minimum possible turn radius of the vehicle.

2. No-Fly Zone which contains the waypoints: If the no-fly zone contains any waypoint, another waypoint is calculated in the same swath width region such that it covers task points which result in high benefit measure (second best point). For example, in figure 1.8 point 1 or 2 can be chosen as the waypoint, which are outside the no-fly zone and can cover tasks. The other best waypoint is calculated using the TABUM approach i.e. point with highest benefit measure and utility metric greater than the specified utility metric threshold value.

1.5.4. Aerial Vehicle Maneuvering.

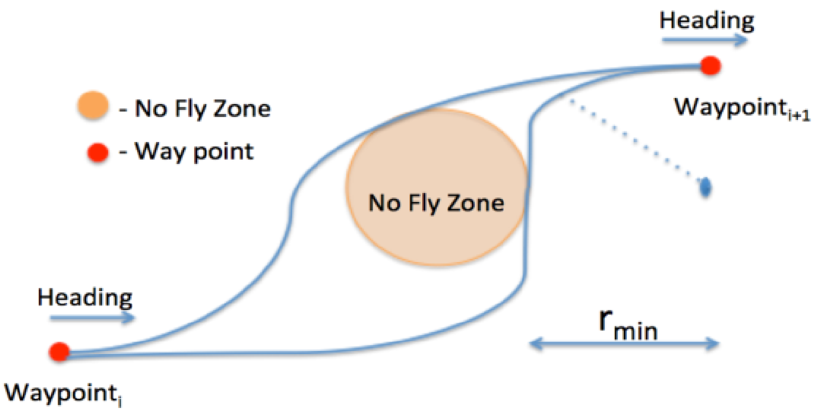


Figure 1.7. Possible paths avoiding no-fly zones

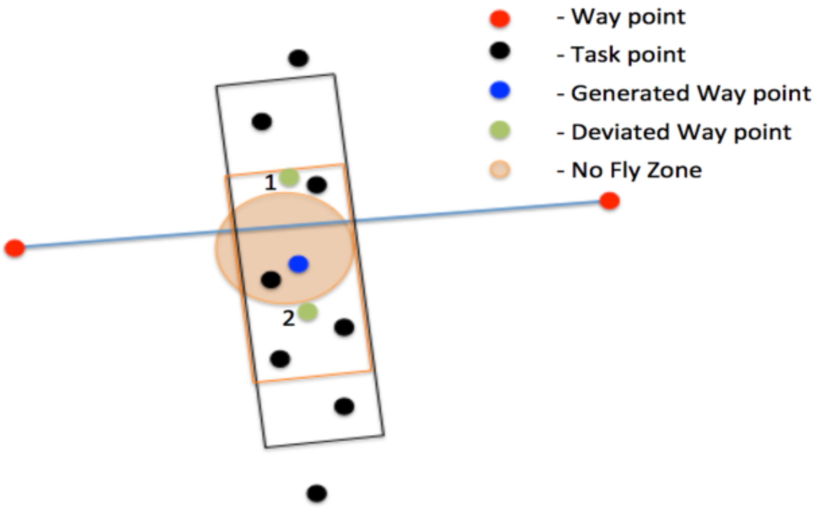


Figure 1.8. No-fly zone containing waypoints

1. Flight Maneuvering: Even high performance aircrafts face difficulties in passing through the close neighboring waypoints due to the dynamic constraints of the vehicle such as radius of turn, climb rate or descent rate etc. So while satisfying the waypoints we must make sure that the flight can reach that point with required minimum stability so that image can be taken from that point. For example, in figure 1.9 if the flight is at a location shown by the point 'current location' then the flight can either take a turn by banking at an angle towards the left or right. It can take a turn with minimum radius of turn making the center as R or L, so it can't reach points within circle. So point 3 can't be reached, points 1 & 4 can be reached. Point 2 can reached but is against the assumption that flight must not turn back or take loops to satisfy the point, so we consider point 2 also as unreachable. To reach the next waypoint by a flight, the distance between the two waypoints must be at least greater than the minimum radius or turn possible by the flight. Also, the altitude difference must not be more than the rate of climb or rate of descent possible by the flight. But for the flight to reach the waypoints with a specific required heading at all the points may not feasible all the times. The conditions for the flight to reach the points with required heading is discussed in Section 1.5.6.

2. For Helicopter-Maneuvering vehicle: Most of the UAV's such as Quad copters, Hex copters etc. have helicopter-maneuvering systems, which can move in all six directions (Six DOF model) without any dynamic constraints of the vehicle. For example, in figure 1.9 all the points can be reached from the current location.

1.5.5. 360° **Camera Rotation.** We have also performed experiments when the camera attached to the aerial vehicle can rotate 360° and can take images of any task point, which falls, in the circular region of radius equal to half the swath width. Figure 1.10 explains the camera rotation possible by the camera.

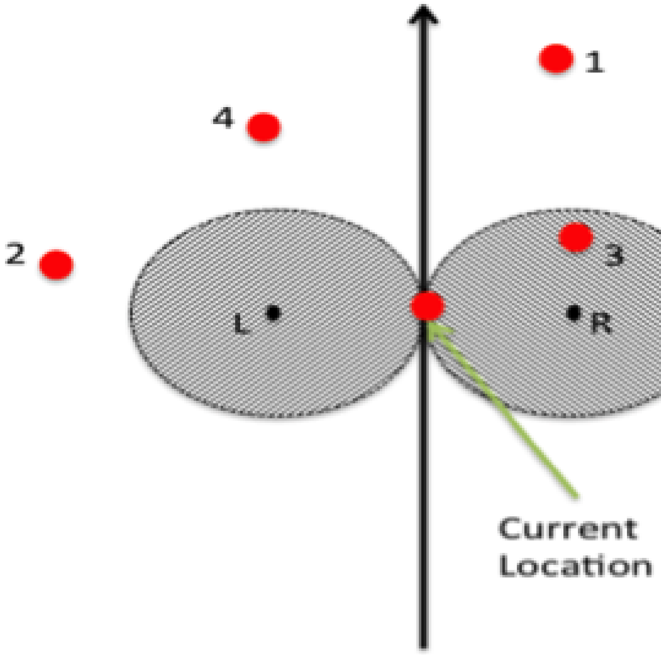


Figure 1.9. Flight maneuvering at a point

Initially we considered that the camera could take images only perpendicular to the flight due to which we need to make sure that flight heading is always parallel to the line-of-sight path. However, if the camera has this capability of taking images by turning 360° then the constraint of the flight heading can be compromised. It means the task aggregation would be the same but the flight can be in any direction and satisfy the task points. In section 1.6, we will see the vehicle's performance change with this capability.

1.5.6. Conditions To Check For Flight To Reach Waypoints. An aerial vehicle cannot reach the waypoints, which are very close to the vehicle position as shown in figure 1.9. If the vehicle has a camera which has the capability of the rotating 360° then we can check that the vehicle can reach the next waypoint and take the images of the task points if the distance between the waypoints is greater than the minimum turn radius as shown in figure 1.9.

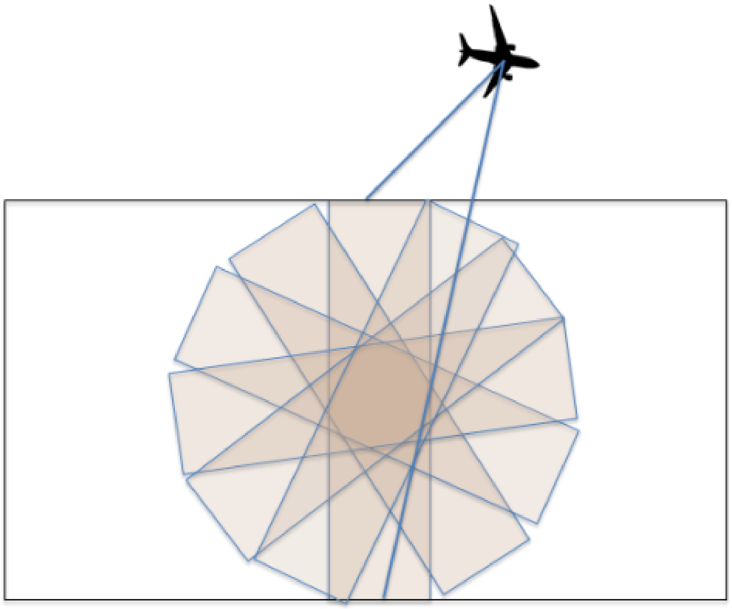


Figure 1.10. 360° Camera Rotation Capability

But if the camera cannot rotate 360° then the flight must reach the next waypoint with heading parallel to the line-of-sight path in order to take the images. Figure 1.12 explains how these conditions are derived. We can check these constraints by the methods shown below:

- If $\Delta y \geq 2 * r_{min}$, then the next waypoint can be satisfied only if $\Delta x \geq 2 * r_{min}$. If the vertical distance between the two waypoints is greater than twice the minimum turn radius then the flight must take turn a 90° turn from the first waypoint to reach the second waypoint. Figure 1.12 explains this trajectory required by the vehicle to satisfy the waypoint with this condition.

- Let θ be the angle at which the flight leaves the turn circle. From figure 1.12, θ can be calculated by making the tangent length in Eq. 1.13 equal to zero i.e. $\Delta y - 2 * r_{min} * (1 - \cos\theta) = 0$. From the above equation we can derive that $\theta = \cos^{-1}(1 - \frac{\Delta y}{2 * r_{min}})$. Then, the next waypoint can be reached with the required heading, if $\Delta x \geq 2 * r_{min} * \sin\theta$. Because if $\Delta x \leq 2 * r_{min} * \sin\theta$, it means that the next way point is very close to the previous way point and cannot be reached with same required heading.

1.6. PERFORMANCE CHANGE BY MANEUVERABILITY CONSTRAINTS

There will be difference in the performance of satisfying the task points with flight maneuvers and helicopter maneuvers which are described below. Also we will check the performance changes when the flight has different camera capabilities.

1.6.1. Total Distance Travelled. The total distance travelled by the vehicle with and without navigational constraints is not equal. Because when the vehicle has flight maneuverability it will take turns to reach the next point where as if it has a helicopter maneuverability it always reaches the next waypoint through the shortest distance possible i.e. line-of-sight through the waypoints as shown in figure 1.11.

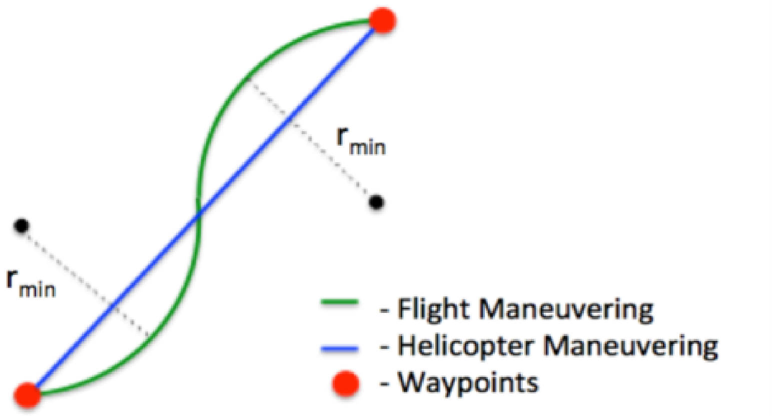


Figure 1.11. Different maneuvers between waypoints

Figure 1.11 shows that the flight trajectory is longer than the helicopter trajectory. In figure 1.11, we represent the common maneuvering of the flight in general case. We try to calculate the length of the flight path it travels to reach one point from other.

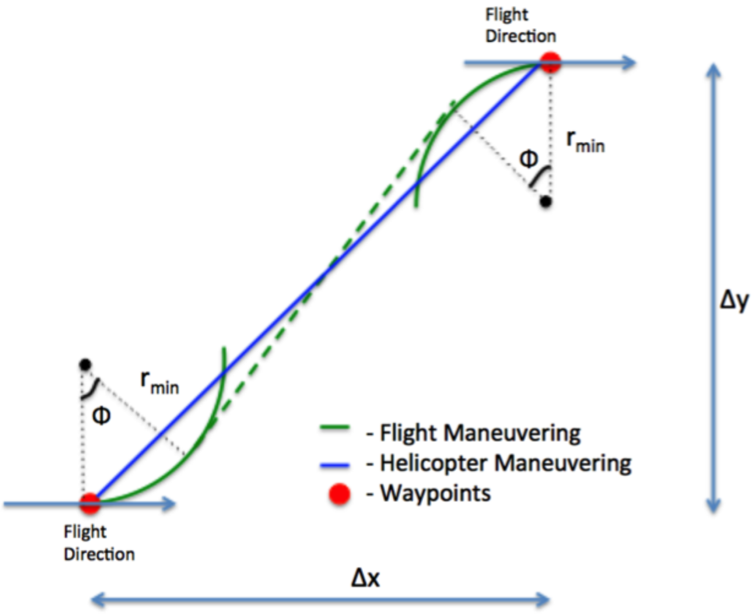


Figure 1.12. Flight and Helicopter distance calculation

From simple geometry, we can calculate the distance travelled by helicopter maneuvering vehicles and flight maneuvering vehicles as shown in Eq. 1.12 and Eq. 1.13 respectively. For helicopter maneuvering:

$$\sqrt{(\Delta x)^2 + (\Delta y)^2} \tag{1.12}$$

For flight maneuvering: 2 * arc length + tangent length =

$$\phi r + \phi r + \sqrt{(\Delta x - 2 * r_{min} * \sin\phi)^2 + (\Delta y - 2 * r_{min} * (1 - \cos\phi))^2} \tag{1.13}$$

where, $\phi \leq 90^\circ$,so that flight will not take loops.

1.6.2. Benefit Measure Achieved. As shown in figure 1.8, when there is a waypoint with highest benefit measure in a no-fly zone then we select the waypoint with next highest benefit measure lying outside the no-fly zone, which is the same procedure for both helicopter and flight maneuvering systems.

If there is no-fly zone is interfering the vehicle path, a helicopter-maneuvering vehicle can avoid the no-fly zone and satisfy the next waypoint even if the no-fly zone is so close to the waypoint. Figure 1.13 shows the helicopter-maneuvering path avoiding the no-fly zone. But for a flight-maneuvering vehicle, there must be a minimum distance between the waypoint and the no-fly zone, which can be checked by the condition shown in the figure 1.13. Also, we can infer from the diagram that with increase in minimum radius of turn of the vehicle it cannot satisfy waypoints closer to the no-fly zones. This results in low benefit measure for the flights with bigger radius of turn. But if the camera can take images by rotating 360° then the flight can reach the closer points by turning before the waypoint arrives. Figure 1.14 shows the required flight heading for different camera capabilities. In figure 1.14, the flight cannot reach the best waypoint without avoiding the no-fly zone so the flight reaches the next best waypoint, which reduces the benefit measure. In figure 1.14, the flight can reach the best waypoint and avoid the no-fly zone by turning much before the no-fly zone, so benefit measure will be high. This comparison is shown in the experiments section where the flight with turn radius greater than the footprint radius can achieve the same performance of the flight with less turn radius.

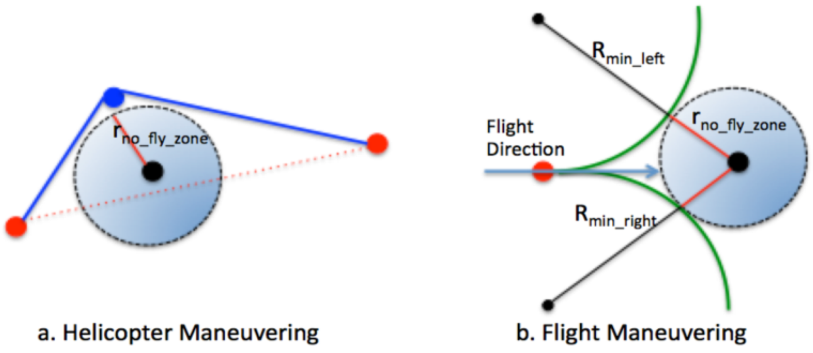
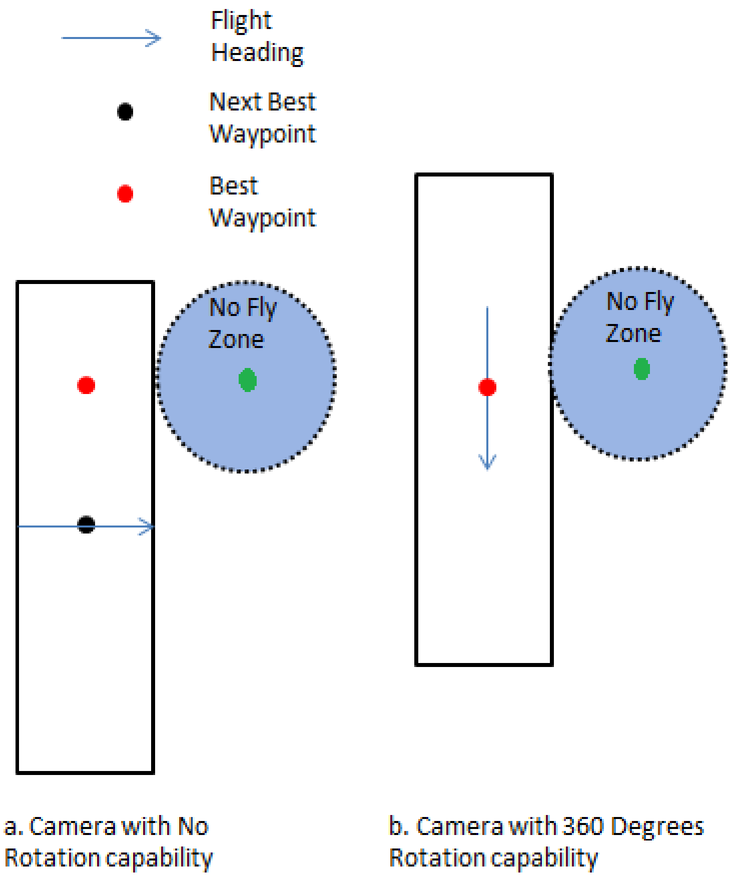


Figure 1.13. No-fly zones avoidance in flight and helicopter maneuvering vehicles

As shown in figure 1.13, the waypoint must be in such a distance so that distance between the points of turn either on right or left and the no-fly zone center must be greater than or equal to sum of minimum radius of turn and the radius of no-fly zone. So, the distance between the black points shown in the figure must be greater than $(r_{min} + r_{no-fly-zone})$ in order to avoid passing through the no-fly zone. If not, we need to change the waypoint to other point that gives next better benefit measure while avoiding the no-fly zone. In figure 1.15, we show a screen shot of the application of how the waypoints are changed to avoid no-fly zones, which are closer to the waypoints or in the path. In figure 1.15 , yellow points are waypoints and pink points are points from where the waypoints are deviated to avoid these zones.

The same waypoints in yellow are satisfied by the helicopter-maneuvering system where as it can't be attained by the flight-maneuvering system due to the navigational constraints of the vehicle. So, only by using the helicopter maneuvering system we can satisfy the waypoints with high benefit measure or else we need to satisfy the next better waypoint by which we can state that high benefit measure and more number of task points can be satisfied by helicopter-maneuvering systems rather than flight-maneuvering systems.

1.6.3. Task Points and QoS Points Satisfied. Moreover the number of task points and benefit measure reduces with increases in utility metric threshold value for flight maneuvering system also because the next better waypoint might have the utility metric value less than the threshold value. For example, there may be two waypoints with high benefit measure, the point which is close enough can be satisfied by the helicopter maneuvering system but not with the flight, then waypoint with next better benefit measure value is selected if and only if the utility metric of that point is above the threshold value. If not, both the waypoints are removed i.e. flight takes the shortest path possible directly to the next waypoint skipping this one. In addition to this, the flight with camera rotation capability can satisfy more task points and QoS points due



a. Camera with No Rotation capability

b. Camera with 360 Degrees Rotation capability

Figure 1.14. Flight heading for different camera capabilities

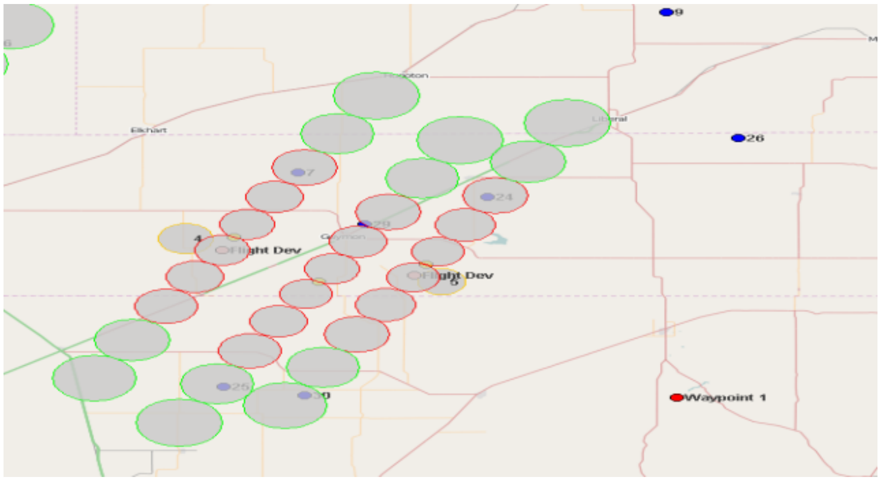


Figure 1.15. No-fly zone avoidance screenshot of sample data

to the relaxation given to the required flight heading. We will explain the performance difference in the experiments section.

1.7. OPTIMIZATION PROBLEM

We frame the problem of path designing for the aircraft passing through specific waypoints while satisfying certain user specific constraints as a conventional optimization problem.

Finally, we state the problem of flight path designing as follows. Let W = $\{W_1, W_2, W_3, ..., W_i, ...W_j\}$ be set of 'j' waypoints which the flight must pass through any cost where W_i is given as format $\{id, \alpha, \beta, \Pi, t\}$. Let T = $\{T_1, T_2, T_3, .., T_i, ...T_k\}$ be set of 'k' task points given by the user where T_i is given in the format $\{id, \alpha, \beta, TP, QP, TiP, TiD, t, Req\}$. The objective of flight is to satisfy the task points Ti if the utility metric is above the specified threshold value. New waypoints are generated from where an image or a video of the task points must be taken based on the constraint shown in Eq. 1.14. n is the number of satisfied task points from any single waypoint.

$$UM = \frac{\sum_{i=1}^{n} TP_i * QP_i}{Deviation} \, and \, UM \geq UM_{Threshold_{v}alue} \tag{1.14}$$

UM value of the generated waypoint is calculated at all possible combinations of the task points, which come under the swath width when an image is taken at that waypoint. The waypoint with highest benefit measure whose UM value is above the threshold value is chosen as the final generated waypoint from where an image must be taken. By increasing the UM threshold value, the flight will travel less distance (less deviation from the original path) and thus satisfies only few tasks i.e. it leads to less benefit measure. So it is up to the user and flight agent to agree upon the utility metric threshold value.

1.7.1. No-Fly Zone Constraints. Let C = $\{C_1, C_2, C_3, ...C_i, ..C_n\}$ be a set of no-fly zones provided to the flight by user of flight agent where C_i is in format of id, latitude, longitude, no-fly-zone radius. While designing the path of the flight which satisfies task points while passing through the waypoints, we must make sure that the flight doesn't pass through no-fly zone C_i. For a flight not to pass through the Ci, we also consider the navigation problems faced by the flight due to its limited maneuverability. The flight avoids the no-fly zones by satisfying the constraint shown in Eq. 1.15.

$$Distance\,between\,any\,point\,on\,the\,path\,and\,C_i \geq C_i' s\,radius + Size\,of\,vehicle \tag{1.15}$$

1.7.2. Navigational Constraints.

1. Helicopter Maneuvering Vehicle: In this case, the aerial vehicle can reach any waypoint regardless of the location of the waypoints where it lies. But it must check a single condition that the vehicle is always moving towards the destination i.e. vehicle should not move backwards to satisfy the next waypoint.

2. Flight Maneuvering Vehicle: In this case, the minimum radius of turn is calculated by the formula based on the bank angle and velocity of the vehicle which is shown in Eq. 1.16.

$$R = \frac{V^2}{g * \tan\theta} \tag{1.16}$$

The flight also has to satisfy the constraints shown in Eq. 1.17 and Eq. 1.18 so that it is practically feasible in the real time scenario.

$$V_{min} \leq V \leq V_{max} \tag{1.17}$$

$$0 \leq \theta \leq \theta_{max} \tag{1.18}$$

Where, V = True airspeed (speed) of the aircraft

V_{min} = Minimum airspeed possible by the aircraft

V_{max} = Maximum airspeed possible by the aircraft

θ = Bank Angle

θ_{max} = Max bank angle

Or the minimum radius of turn of the flight can also be calculated when the load factor that can be maintained by the flight is specified by the user. The minimum radius of the flight is satisfied calculated by the Eq. 1.19.

$$R = \frac{V^2}{g * \sqrt{l^2 - 1}} \tag{1.19}$$

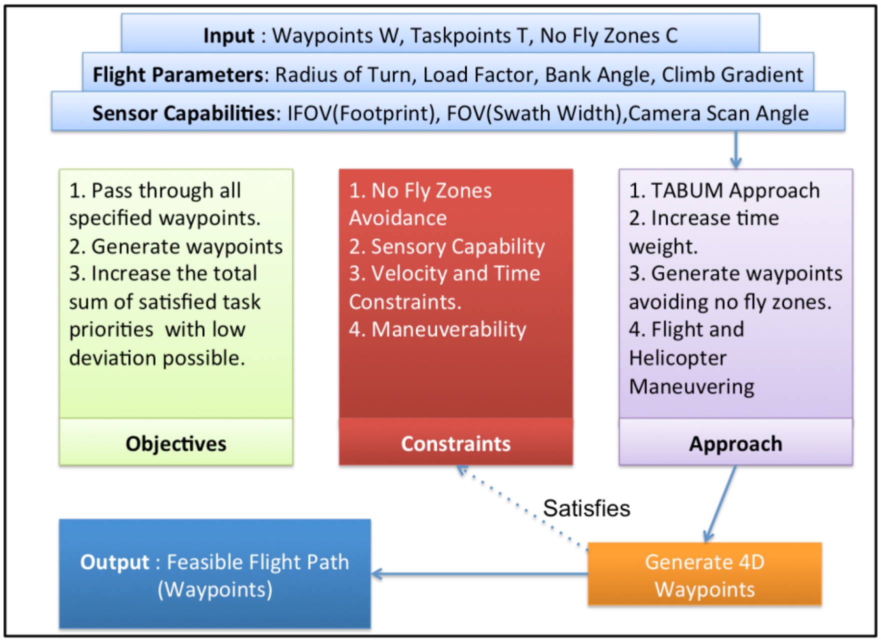
where, load factor l = L/W, L is load and W is weight of the aircraft. The load factor must always be within the limits as specified in Eq. 1.20 i.e. it must not exceed the maximum load factor attainable by the flight.

$$1 \leq l \leq l_{max} \tag{1.20}$$

Figure 1.16 summarizes the entire trajectory designing approach where it specifies the input to the algorithm which includes waypoints, task points, no-fly zones, flight parameters and sensor capabilities. We can also see the approach followed to achieve the objectives while satisfying the constraints in order to generate a feasible flight path.

1.8. CONCLUSION AND FUTURE WORK

We proposed a 4D trajectory generation method, which passes through the specified waypoints while taking the limited maneuverability of the aerial vehicle into


Input : Waypoints W, Taskpoints T, No Fly Zones C
Flight Parameters: Radius of Turn, Load Factor, Bank Angle, Climb Gradient
Sensor Capabilities: IFOV(Footprint), FOV(Swath Width),Camera Scan Angle
1. Pass through all specified waypoints.
2. Generate waypoints
3. Increase the total sum of satisfied task priorities with low deviation possible.
Objectives
1. No Fly Zones Avoidance
2. Sensory Capability
3. Velocity and Time Constraints.
4. Maneuverability
Constraints
1. TABUM Approach
2. Increase time weight.
3. Generate waypoints avoiding no fly zones.
4. Flight and Helicopter Maneuvering
Approach
Satisfies
Output : Feasible Flight Path (Waypoints)
Generate 4D Waypoints

Path Planner Process Diagram

Figure 1.16. Path planning algorithm requirements, constraints and approach

consideration. We generate new waypoints between the specified waypoints where an image must be taken so that tasks with higher priority weights are covered. To generate these waypoints we use the utility metric, which is based on the benefit and cost measure. We try to satisfy the task points from the generated waypoints as long as the utility metric is above the threshold value. The flight path is designed in such a way that it ensures that it never passes through the no-fly zones. Different paths are generated based on the utility metric threshold from which the pilot can choose the best suitable path for the mission.

More research needs to be done in order to satisfy the task points when the flight can take loops between the waypoints. This way the flight can satisfy more task points by reaching the waypoints, which are not possible to reach without loops. Also, better results can be attained if the capability of the camera increases i.e. if it can take images far away from the flight position. This way many task points can be satisfied without any deviation from the line-of-sight path and also quality of the images can be improved.

SECTION

3. EXPERIMENTAL EVALUATION

In this Section we first describe about the technical implementation of the TABUM procedure, further a detailed discussion of experimental setup is described and finally a detailed analysis of the results concludes the section.

3.1. TECHNICAL IMPLEMENTATION

Requests are sent by the users in the form of the task points to the aerial vehicle which flies from starting waypoint to destination waypoint through the other pre-defined waypoints. As discussed, task points from the different base stations on the ground come in the format of $[id,\alpha,\beta,TP,QP,TiP,TiD,t,Req]$ and waypoints are in the form of $[\alpha,\beta,\Pi,t]$. We need to generate the new waypoints where the flight must take an image or a video, maximizing the utility metric with least deviation from the original path. Finally, the path of the flight would be the passing through all of the pre-defined waypoints and generated waypoints from where images/videos should be taken.

In our scenario, we assume that the sensor/camera attached to the aerial vehicle can take images/videos only perpendicular to the flight path and it can only move the camera to an extent using the lateral movement of the camera (camera moved along camera scan angle to cover the entire swath width which is collection of footprints) as shown in 3.1. So, the flight must reach the waypoints at the same heading as the path angle to satisfy the requests with minimum number of images i.e. the flight can cover as many task points as possible with least number of images (optimum utilization).

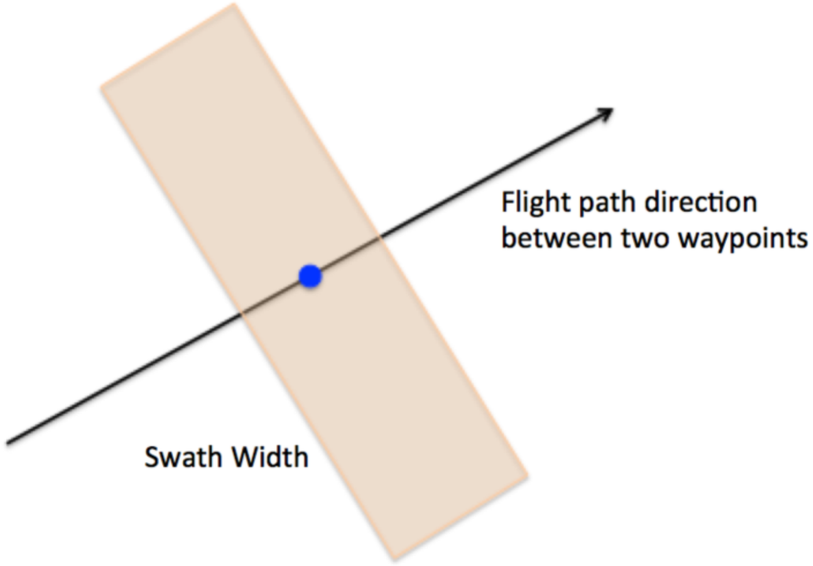


Figure 3.1. Swath Width Direction Perpendicular to Flight Path

Step 1: Given the input of waypoints and task points, task points are divided logically such that they fall between any of the two pre-defined waypoints as shown in 3.2. Here, the task points are represented by triangles and waypoints are represented by red dots. Different colors for the task points are assigned based on which group they fall into. If any of the task points falls under more than one division then that task point is added to the division which is closer to the path and feasible for the flight to take an image of it.

Step 2: After step 1, we find the intersection point(image) of each task point on the flight path between the pair of waypoints whichever that task point belongs to, based on the division done in Step 1. Figure 3.3 explains this step.

Step 3: Then, these intersection points are grouped together based on the foot print diameter i.e. group all those points which fall under a same foot print region as shown in 3.4. The midpoint of the footprint is taken as the new waypoint which covers all the task points of the grouped images (respective task points) of its respective group.

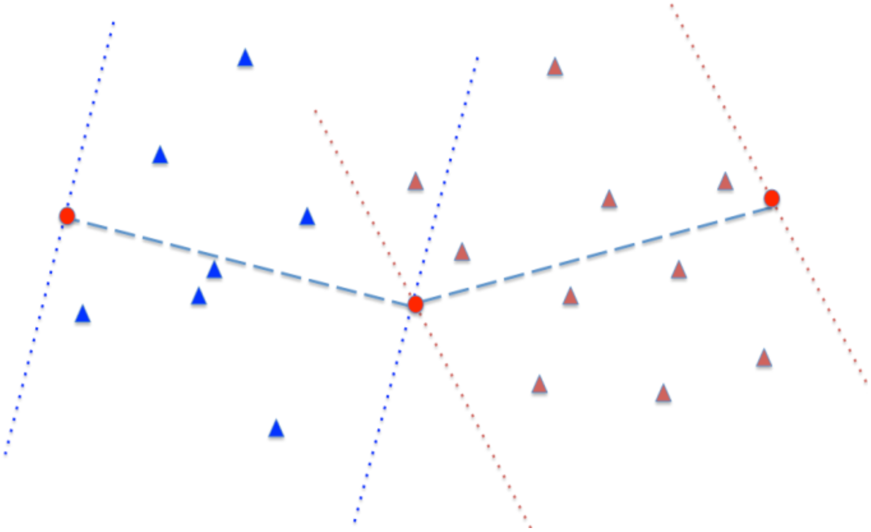


Figure 3.2. Task points logically divided into groups based on their location

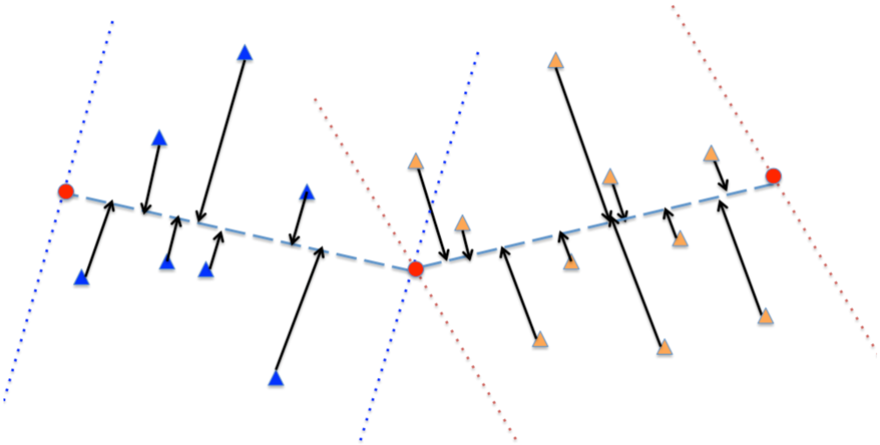


Figure 3.3. Images of task points calculated on flight original path

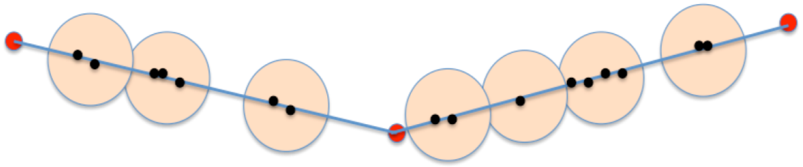


Figure 3.4. Grouping based on Footprint Diameter

Step 4: Then, we check whether the generated waypoint can cover all the task points of its group or not i.e. whether those task points fall under the swath width of that waypoint. If yes, that generated waypoint would be the final waypoint, which the flight must pass through, or else we need to use the utility metric to find another waypoint and select the waypoint which would give us higher utility metric i.e. higher benefit measure and lower cost measure. Also the utility metric must be above the threshold value given by the user, if not we select the waypoint on the path itself as the final waypoint.

Step 5: We then use the TABUM approach described in the paper to determine the final waypoint. Also, we implemented all the algorithms described in the paper such as no-fly zone avoidance, consideration of aerial vehicle maneuvering capabilities, 360° camera rotation.

3.2. EXPERIMENTAL SETUP

We performed the experiments by taking a sample dataset of 7 waypoints where waypoint 0 is the departure point and waypoint 6 is the destination point for the aerial vehicle. A pre-defined set of circular regions with different radii is taken as the no-fly zones through which the aerial vehicle should not pass and a set of 125 task points is taken as the input from the users in the format shown in Eq. 1.3 and the flight path is designed using them. A sample waypoint, task point and no-fly zone formats are shown below respectively:

T - [1, 37.611512, -103.897705, 10000, 06/08/2015 12:45:18]

W - [1, 37.652704, 103.191833, HIGH, MED, HIGH, GE, 06/08/2015 12:54:08, Image/Video]

C - [1, 37.044572, -102.564919, 6.2]

Where, the first value is the id of the respective inputs.

Tasks will be satisfied according to this utility metric value threshold and one with highest benefit measure. We will be calculating the total benefit measure and total distance travelled for changing utility metric threshold values.

To simulate the waypoints, task points, no-fly zones and flight path, we used the OpenStreetMaps (OSM) [9] data to download the maps and access the locations by latitude, longitude. We integrated our algorithms (TABUM, flight maneuverability, and No-fly zones avoidance) for generating flight path into this implementation tool, which used OSM data.

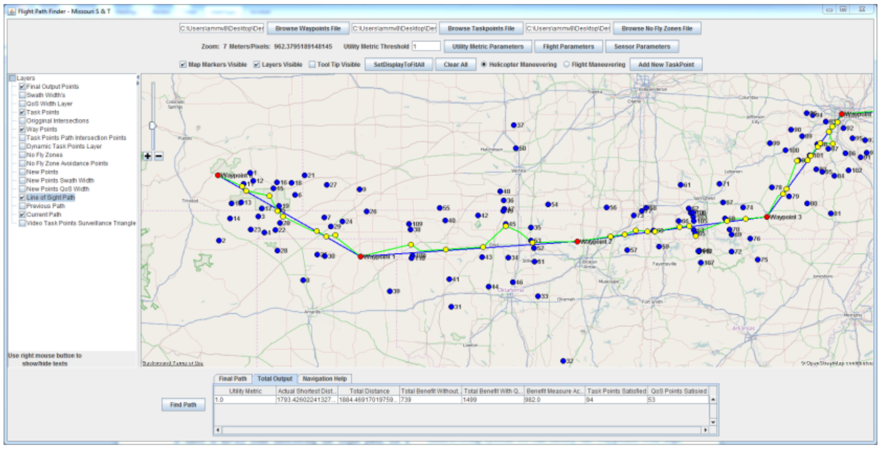
3.3. RESULTS

The data given has 7 waypoints from which we can calculate that the shortest distance i.e. line-of-sight through the waypoints is 968.37 nautical miles and total benefit measure (including QoS) that needs to satisfy is 1499 for the 125 task points. All this data (task points) is generated using a random point generator within a specified region [10]. Figure 3.5 shows the screenshot of the application, which is used to perform the experiments. Figure 3.5 shows the screenshot of the application with different flight paths varying UM values.

In Figure 3.6, waypoints are colored red and generated waypoints are colored yellow. The red line is path when UM threshold is 0, green line is path when UM = 1, and blue line is line-of-sight path (for high UM threshold). From figure 3.6, we can see that generated waypoints tend to deviate less from the path as UM threshold value increases. This means that as the utility metric threshold increases the flight deviates only if there is a high benefit measure compared to the cost measure.

We try to evaluate the importance of utility metric value in the process of trajectory generation by trying to analyze the flight path changes by change in utility metric threshold value and also for the two different maneuvers. We also compare the performance of the flight for different turn radius capabilities.

As shown in the figure 3.6, we can state that as utility metric threshold value increases, the flight travels less distance i.e. deviates less from the shortest path (line-of-sight) passing through the waypoints. If UM = 0, the flight travels the maximum because the flight can satisfy maximum tasks with any deviation possible. As UM threshold increases, the total distance travelled also decreases. At one value (UM = 50) of utility metric threshold value, the total distance travelled reaches the minimum value i.e. line-of-sight path distance. So the flight travels along the line-of-sight of waypoints if the utility metric is high. But for the flight maneuver trajectory the shortest distance is little more, as it needs to take turns at the pre-defined waypoints.



C:\Users\ammv8\Desktop\De[
C:\Users\ammv8\Desktop\De[
C:\Users\ammv8\Desktop\De[

Zoom: 7 Meters/Pixels: 962.3795189148145 Utility Metric Threshold 1

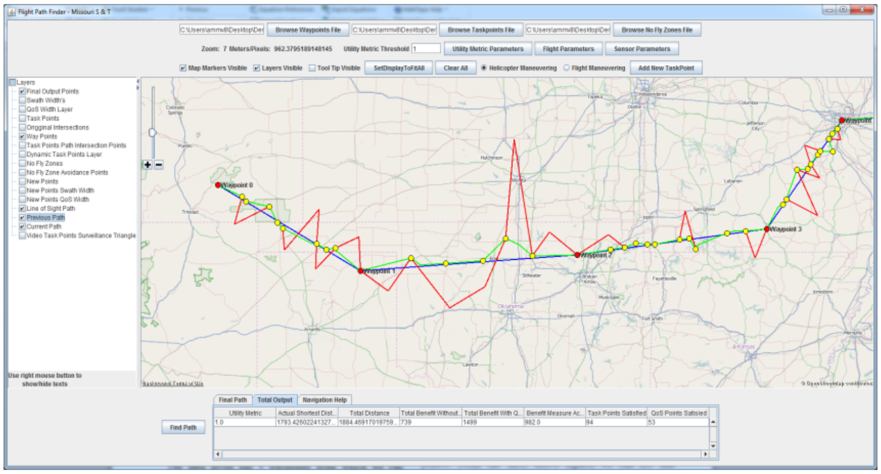
☑ Map Markers Visible ☑ Layers Visible □ Tool Tip Visible ● Helicopter Maneuvering ○ Flight Maneuvering

☑ Final Output Points
□ Swath Width's
□ QoS Width Layer
☑ Task Points
□ Origginal Intersections
☑ Way Points
□ Task Points Path Intersection Points
□ Dynamic Task Points Layer
□ No Fly Zones
□ No Fly Zone Avoidance Points
□ New Points
□ New Points Swath Width
□ New Points QoS Width
☑ Line of Sight Path
□ Previous Path
☑ Current Path
□ Video Task Points Surveillance Triangle

Utility Metric	Actual Shortest Dist...	Total Distance	Total Benefit Without...	Total Benefit With Q...	Benefit Measure Ac...	Task Points Satisfied	QoS Points Satisied
1.0	1793.42602241327...	1884.46917019759...	739	1499	982.0	94	53

Figure 3.5. Flight Path Finder Application

In figure 3.7, we can see that the distance travelled is least for the helicopter maneuvering vehicle for all the utility metric thresholds. When the threshold value is 0.1, we can see that flight with less turn radius travels more because it is capable to take sharp turns and reach the waypoints by deviating from the line-of-sight path but the flight with high turn radius cannot reach few waypoints and takes the line-of-sight path. Similarly, the flight with a camera which has 360° rotation capability can cover more task points thus travels more distance than the flight which doesn't have a camera



C:\Users\ammv8\Desktop\Der C:\Users\ammv8\Desktop\Der C:\Users\ammv8\Desktop\Der

Zoom: 7 Meters/Pixels: 962.3795189148145 Utility Metric Threshold 1

☑ Map Markers Visible ☑ Layers Visible ☐ Tool Tip Visible ● Helicopter Maneuvering ○ Flight Maneuvering

☐ Layers
- ☑ Final Output Points
- ☐ Swath Width's
- ☐ QoS Width Layer
- ☐ Task Points
- ☐ Origginal Intersections
- ☑ Way Points
- ☐ Task Points Path Intersection Points
- ☐ Dynamic Task Points Layer
- ☐ No Fly Zones
- ☐ No Fly Zone Avoidance Points
- ☐ New Points
- ☐ New Points Swath Width
- ☐ New Points QoS Width
- ☑ Line of Sight Path
- ☑ Previous Path
- ☑ Current Path
- ☐ Video Task Points Surveillance Triangle

Utility Metric	Actual Shortest Dist...	Total Distance	Total Benefit Without...	Total Benefit With Q...	Benefit Measure Ac...	Task Points Satisfied	QoS Points Satisied
1.0	1793.42602241327...	1884.46917019759...	739	1499	982.0	94	53

Figure 3.6. Flight paths for different utility metric thresholds

with 360° rotatable camera. From the graph shown in figure 3.7, we can derive that by introducing the utility metric threshold value, we can reduce the total distance travelled to satisfy the task points according to the user's interests and vehicle's mission.

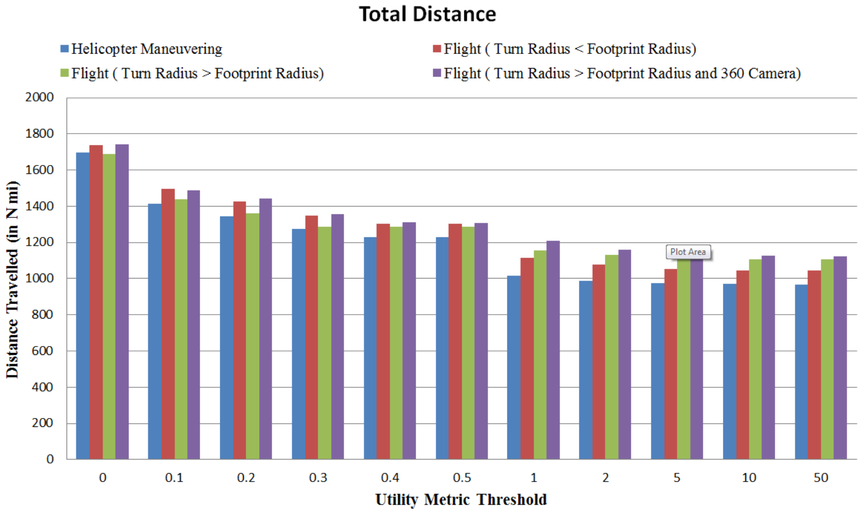


Figure 3.7. Total Distance Vs. Utility Metric Threshold Values

In figure 3.8, benefit measure satisfied at low utility metric threshold is higher than when it is at high threshold as the flight deviates less from the shortest path. So, when UM = 50, benefit measure = 734 i.e. the weight of the tasks that are covered just by passing through the waypoints along the line-of-sight. We can see that as utility metric increases the vehicle covers only few tasks by not deviating much from the line-of-sight path. The flight with bigger turn radius has less benefit measure compared with flight with small turn radius due to the maneuverability constraints. Also, the flight with 360° rotatable camera almost gives the same performance of the flight with smaller turn radius despite having a bigger turn radius because flight can reach the waypoints with any heading as explained in Section 1.6.2.

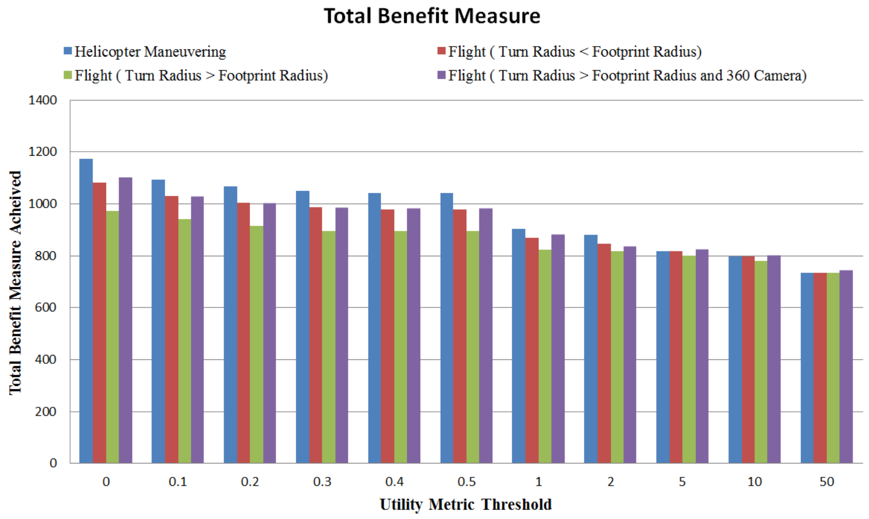


Figure 3.8. Total Benefit Measure Vs. Utility Metric Threshold Values

In figure 3.9 and 3.10, we can see that the fewer number of the task points and QoS points satisfied as the utility metric threshold value increases from 0 to 50. At few utility metric threshold values, we see that flight-maneuvering vehicle satisfies more points than helicopter maneuvering despite its maneuverability constraints, which depends on the distribution of the task points.

Also, the flight with bigger turn radius satisfies less number of task points when compared with one which has smaller turn radius. However, if a 360° rotatable camera is attached to the flight with larger turn radius it covers almost the same number of task

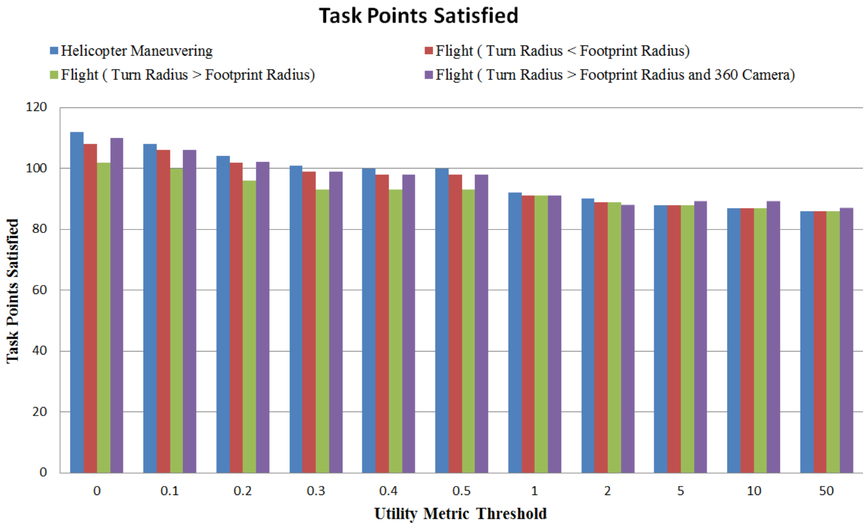


Figure 3.9. Number of Task points Satisfied Vs. Utility Metric Threshold Values

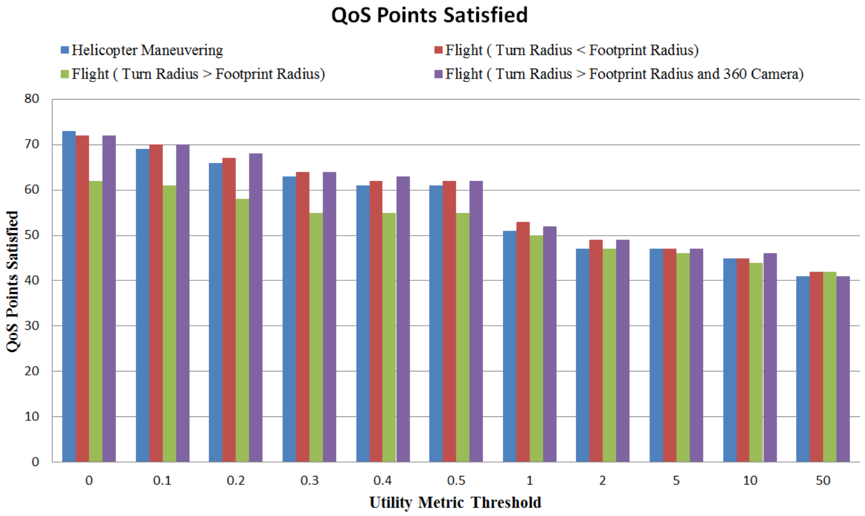


Figure 3.10. Number of QoS points Satisfied Vs. Utility Metric Threshold Values

points which the flight with smaller turn radius covers due to the relaxation in required flight heading.

From these experiments, we can derive that the helicopter maneuvering gives the best result in satisfying the task points due its flexible maneuverability. Also, based on the utility metric threshold value, the different task points with different benefit measures can be satisfied. So, the utility metric can be chosen based on the user's interests and the vehicle's mission. Also, we can see that flights with more flexible maneuverability and camera capabilities can satisfy more number of task points.

4. CONCLUSION AND FUTURE WORK

We are able to generate the aerial vehicle trajectories from the simulation tool which we developed by taking the input from the user. We have seen that constraints on the total distance travelled can be relaxed by giving a low utility metric threshold. Trajectories are generated which satisfies maximum number of task points if the flight is allowed to deviate more with less constraints on total distance travelled i.e. by giving a low utility metric threshold value. We have seen that as utility metric increases aerial vehicle deviates less from the line-of-sight path which results in less distance travelled at the cost of not satisfying few task points i.e. by achieving less benefit measure. So we need to select a utility metric threshold based on the requirements of the mission and bring a tradeoff between total distance travelled and total benefit measure achieved.

Also, we concluded that if the aerial vehicle has more flexible maneuverability like less turn radius, then it can satisfy more number of task points because the vehicle can take close turns and reach close neighboring points easily. So, the total benefit measure achieved is less if the aerial vehicle has larger turn radius. However, the task satisfaction can be increased with better camera capability like 360° rotation by which more tasks can be satisfied from a single point. So, an aerial vehicle with 360° camera rotation capability will achieve more benefit measure compared to the flight with same maneuverability characteristics but having a fixed camera.

More research needs to be done in order to satisfy the task points when the flight can take loops in its path between the waypoints. This way the flight can satisfy more task points by reaching the waypoints, which are not possible to reach without loops. Also, better results can be attained if the capability of the camera increases i.e. if it can take images far away from the flight position. This way many task points can be satisfied without any deviation from the line-of-sight path and also quality of the images can be improved.

BIBLIOGRAPHY

[1] K BOUSSON and PAULO FF MACHADO. 4d trajectory generation and tracking for waypoint-based aerial navigation. *WSEAS Transactions on Systems & Control*, 8(3), 2013.

[2] Gwanyoung Moon and Youdan Kim. Optimum flight path design passing through waypoints for autonomous flight control system. In *AIAA Guidance, Navigation, and Control Conference and Exhibit, Austin, TX, AIAA Paper*, volume 5334, pages 11–14, 2003.

[3] B Ajith Kumar and D Ghose. Radar-assisted collision avoidance/guidance strategy for planar flight. *Aerospace and Electronic Systems, IEEE Transactions on*, 37(1):77–90, 2001.

[4] Wei Liu, Zheng Zheng, and Kaiyuan Cai. Adaptive path planning for unmanned aerial vehicles based on bi-level programming and variable planning time interval. *Chinese Journal of Aeronautics*, 26(3):646–660, 2013.

[5] Erik P Anderson, Randal W Beard, and Timothy W McLain. Real-time dynamic trajectory smoothing for unmanned air vehicles. *Control Systems Technology, IEEE Transactions on*, 13(3):471–477, 2005.

[6] Guang Yang and Vikram Kapila. Optimal path planning for unmanned air vehicles with kinematic and tactical constraints. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 2, pages 1301–1306. IEEE, 2002.

[7] J Foo, Jared Knutzon, James Oliver, and Eliot Winer. Three-dimensional path planning of unmanned aerial vehicles using particle swarm optimization. In *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Portsmouth, Virginia*, 2006.

[8] Luitpold Babel. Trajectory planning for unmanned aerial vehicles: a network optimization approach. *Mathematical Methods of Operations Research*, 74(3):343–360, 2011.

[9] Open Street Maps. `http://www.openstreetmap.org`, 2014. . Open Street maps are used for simulation of the experiments, which implement the algorithms described, and to check their performance.

[10] Generating random task points. `http://www.geomidpoint.com/random/`, 2014. .We used this random point generator to generate the task points within the specified rectangular region.

VITA

Amarender Reddy Mekala was born on May 10th, 1991 in Hyderabad, India. He earned his Bachelor's degree in Computer Science and Engineering from Osmania University, India in 2012. He has been a graduate student in the Computer Science Department at Missouri University of Science and Technology since August 2013 and worked as a Graduate Research Assistant under Dr. Sanjay Kumar Madria from August 2013 to May 2015. He received his Masters in Computer Science at Missouri University of Science and Technology in May 2015.