Path Planning for a UAV in an Agricultural Environment to Tour and Cover Multiple Neighborhoods

Koel Sinha

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Kevin K Kochersberger
Pratap Tokekar
Michael Roan

Sep 28, 2017
Blacksburg, Virginia

Copyright 2017, Koel Sinha

Path Planning for a UAV in an Agricultural Environment to Tour and Cover Multiple Neighborhoods

Koel Sinha

ABSTRACT

This work focuses on path planning for an autonomous UAV to tour and cover multiple regions in the shortest time. The three challenges to be solved are - finding the right optimal order to tour the neighborhoods, determining entry and exit points to neighborhoods, and covering each neighborhood. Two approaches have been explored and compared to achieve this goal - a TSP - Greedy and TSP - Dijkstra's. Both of them use a TSP solution to determine the optimal order of touring. They also use the same back and forth motion to cover each region. However, while the first approach uses a brute force to determine the the next closest node of entry or exit, the second approach utilizes the Dijkstra's algorithm to compute all possible paths to every node in the graph, and therefore choose the shortest pairs of entry and exit for each region, that would generate the shorter path, overall. The main contribution of this work is to implement an existing algorithm to combine the touring and covering problem, and propose a new algorithm to perform the same. Empirical results comparing performances of both approaches are included. Hardware experiments are performed on a spraying hexacopter, using the TSP - Greedy approach. Unique system characteristics are studied to make conclusions about stability of the platform. Future directions are identified to improve both software and hardware performance.

This work received support from DuPont USA.

Path Planning for a UAV in an Agricultural Environment to Tour and Cover Multiple Neighborhoods

Koel Sinha

GENRAL AUDIENCE ABSTRACT

In a world with a rapidly growing population and resources depleting faster, increasing efficiency and productivity has become paramount. Until now, automation has helped cope with the world's increasing demand for food. However, studies have shown that automation in itself will be insufficient in improving crop output in the coming years. Fortunately, another technology that is taking big leaps in terms of technological advances - Information Technology, when combined with automation, presents itself as a viable option. This takes agriculture towards a site-specific approach for all crop monitoring, growth and protection activities and is know as Precision Agriculture. Spraying fluids on crops using a UAV is on of the prominent problems being researched in this field. This work presents two approaches - TSP - Greedy and TSP - Dijkstra's to tour or visit and spray multiple regions that have been previously identified in the shortest time. While the TSP - Greedy algorithm is an implementation of an existing approach, the TSP - Dijkstra's algorithm is a new approach proposed in this work. The solution to TSP or Traveling Salesman Problem generates the optimal order to visit the regions. The 'Greedy' or 'Dijsktra's' approaches define entry and exit points for each region, that gives the shortest path overall. Images of areas with weed afflicted regions marked on them are used as the input for this algorithm. The TSP-Greedy approach is used in performing hardware experiments. Data collected from these experiments are used to analyze performance of an Unmanned Aerial Vehicle (UAV) platform. Water has been used as the spraying fluid for testing the sprayer assembly. GPS or Global Positioning System is used for navigation of the UAV. Future directions are identified to improve both software and hardware performance.

To my father - my hero, my inspiration.

Acknowledgments

I would like to thank Dr. Kevin Kochersberger for giving me the opportunity to be a part of the Unmanned Systems Lab at Virgnia Tech and work on a challenging and relevant research area such as this. His belief in my potential has inspired me to push the envelop during my time here.

I would like to thank Dr Pratap Tokekar for always welcoming my questions and giving me insights that helped me make my work more holistic. I thank Dr Mike Roan for being a part of my committee and his valuable comments.

Dr John Bird joined the lab during a very crucial period of my master's studies and even in the short time, has always been a guiding voice with his experiences and useful anecdotes. I would also like extend my appreciation to Drew and Haseeb from the lab for their help with conducting the numerous field tests required to complete my thesis.

I want to thank my roommates Amruta, for never loosing faith in my abilities, even when I was in doubt and Shriya, for never letting me loose sight of my goal. Amith, Arpit and Aravind deserve a mention here for never letting me procrastinate. I would also like to thank Varun for being a steady pillar of support during every single moment of despair that I had during the last few months.

This section would be incomplete without thanking my little sister, Urmika for always being my silver lining in any situation and for making me laugh even during the most stressful times. Lastly, I want to take this opportunity to express my gratitude to my father, who has been unwavering with his faith and support for as long as I can remember.

Contents

List of Figures

Chapter 1

Introduction

Advancements in robotics are percolating into every facet of our lives. It is, therefore natural for it to touch agriculture, a field that is a major contributor to our economy in terms of producing food, money and employment.

In [1], Thomas Malthus observed that the rapidly growing human population would soon surpass the available food leading to food scarcity and famine. Thanks to advancements in mechanization and automation - from ploughs to tractors to autonomous robots - man kind has moved from an estimated population of 980 million in the 1800s to 7.5 billion today. It has been predicted that in order to meet the needs of a projected population of 9 billion in 2050, agricultural productions must double [2]. Our resources are not limitless and we must therefore rely on scientific innovations to increase our total factor productivity of global agriculture from 1.4 to the suggested 1.75[3][4][5]. In simpler words, the efficiency of agricultural processes must be improved to utilize available resources better.

Automation has brought about a substantial increase in agricultural productivity and has also reduced the need for manpower[6][7][8]. In recent times, intelligent machines have proved to be helpful in tapping our resources further [9]. Robotics, therefore presents itself as a promising and - thanks to the ever-growing interest and research in it - a viable answer.

Another, more recent technology that has been augmented by the rapid growth in different means of data collection is 'Precision Agriculture'. Also referred to as precision farming, prescription farming etc in different literatures, it represents the use of electronic and spatial information technology for crop monitoring and guide site-specific crop management (water, chemicals,seeds, harvesting etc) . It also reduces waste production and minimizes negative effects on the environment.

Use of different chemicals is a component central to growth of crops. Spraying of these chemicals in agricultural fields generally involve either aerial or terrestrial applicators. While terrestrial applicators can achieve better distribution of product, they require designated paths within the field. This makes them susceptible to disturbances due to unevenness of terrain and reduces the area for crop production. Aerial applicators can fly over crops for the various agricultural actions and therefore do not suffer the disadvantages of terrestrial applicators. They are also are faster and more flexible. But, the increased distance of the applicator from the crops leads to a drift in the spray[10]. However, research has shown that in case of multi-rotor UAVs, downwash of the rotors can act as a tunnel preventing product drift , presenting themselves as solutions to requirements, both of maneuverability and high precision[11].

The goal of this research was to develop a path planning algorithm for an autonomous UAV to tour previously identified regions and spots and spray them precisely

Chapter 2

Literature Review

The literature review is organized into four broad categories: UAVs in aerial spraying, navigation and path planning.

2.1 UAVs in Aerial Spraying

The advent of UAVs as aerial sprayers has been fairly recent and its adoption commercially is still limited. Japan's Yamaha Corporation was the first to produce an unmanned aerial vehicle for spraying crop protection product. [11] describes the development of a low volume spray system for a UAV for precise spray application. [12] looks at developing a PWM controller for a UAV precision sprayer and [13] explores an electrostatic spraying system. [14] develops a UAV to activate a spray release controlled by an electronic control system based on pre-programmed spray locations in terms of GPS coordinates.

2.2 Navigation

Automation in agriculture cannot be achieved without appropriate navigation techniques. Navigation can be (i) Manual, (ii) Human assisted or (iii) Fully autonomous. Fully autonomous navigation can be further categorized into two. The first method determines the path off-line with the help of local and global positioning systems. The target area is generally a known environment in this case. The second method uses inputs from different sensors and references such as crop lines to establish a course of navigation for the robot.

Compared to the different sensors that could be used for navigation, GPS is easier to integrate. Not only that, recent advancements in GPS technology have achieved accuracies of the order of 1 cm [15].[16] notes an increase in yield gains by incorporating the use of GPS in the planting process. It is concluded in [17] that the use of GPS has resulted in reduced fuel consumption. Stanford University developed an automatic steering system based on RTK-GPS (Real time kinematic-GPS) for a medium sized John Deere 7800 and achieved a typical positioning error average of almost zero under full load as opposed to the 40 - 60 mm achieved using manual steering [18]. RTK-GPS makes use of multiple satellite constellations and depends on the phase of the signal as opposed to the payload of the signal. This results in a better resolution of pseudo-range and makes it one among the most advanced and accurate GPS technologies [15].

In this work, we have used images with afflicted areas identified and marked, to plan a path off-line and use RTK-GPS for autonomous navigation of the UAV.

2.3 Path Planning

Although path planning in robotics is an old and extensively studied problem, path planning for agricultural robots is a more recent research topic. Commercial adoption of autonomous guidance systems has been slow with crop-row guidance systems achieving some success [19],[20]. However, with an increase in agricultural productivity being the need of the hour, Drone manufacturing companies such as DGI and others are working towards making crop production and monitoring UAVs commercially available.

The planning problem in our work can be broken down into (i) a coverage problem and (ii) a touring problem

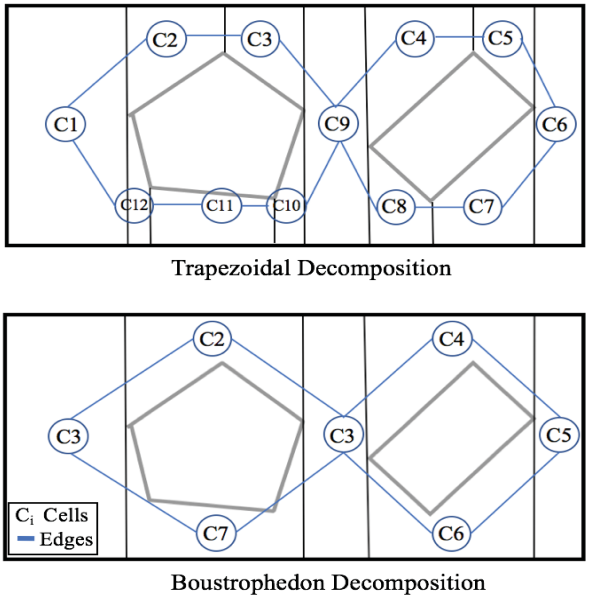

Figure 2.1: Cell Relation

A lot of research has been conducted on complete coverage of a given area. While a randomized approach can work for robots used for domestic purposes such as cleaning, agricultural applications require a more optimized solution. [21] and [22] review different methods of breaking down a region into sub-regions or cells. In the classical cellular decomposition, cell

boundaries are obtained by events encountered while sweeping a line across the the space from left right. This information is used to create an adjacency graph where the nodes are the decomposed cells and the edges are the relationship between two cells. Subsequently, a path is computed to visit every node in the adjacency graph exactly once. This work is further extended to develop the trapezoidal method for cell decomposition to ensure complete coverage.

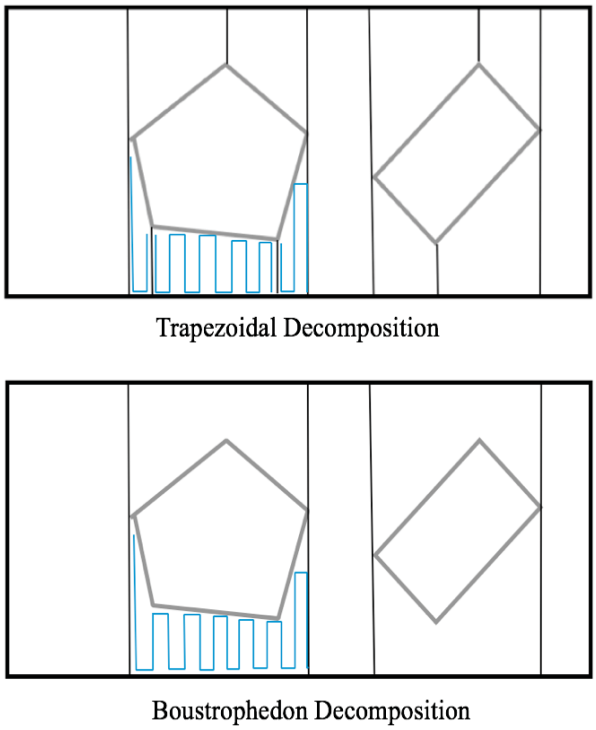

Figure 2.2: Merging smaller cells to form a bigger cell reduces number of trips to scan the region

By restricting each cell to a trapezoid shape, it is seen that every cell could be covered using a back and forth motion. However, results obtained using this method show that some of the smaller cells can be merged to form a bigger cell which would reduce the number of back and forth laps required to cover them. [21] explores sweeping the space while varying the angle of the sweep line at 30° intervals. Three directions, which give the biggest cellular decompositions are chosen and the whole process is repeated adding angles 15° lesser and

greater than the filtered ones. Once the improvement per step meets set threshold requirements, the largest cell so far is removed and the process continued until the entire space is covered. Building on this, [22] and [23] achieve similar cell decomposition for both convex and non-convex cells. This technique, termed as Boustrodephon decomposition takes into account only those vertices from which vertical lines could be drawn both upwards and downwards. Both of these methods are off-line methods. [24] achieves cellular decomposition for non-polygons and n dimensional space by sweeping a slice through the space and denoting points where the surface normal of the obstacle is perpendicular to the slice as cell boundaries [24][25][26].

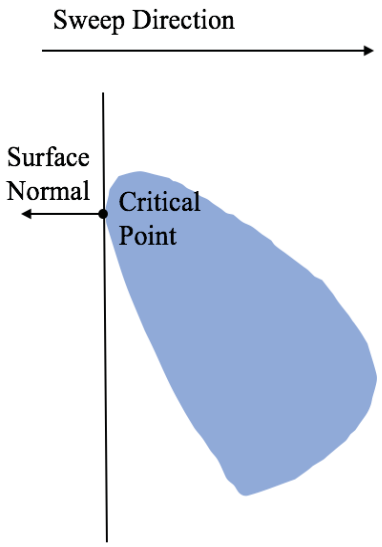

Figure 2.3: Morse Decompostion

[25] also surveys other, more recent coverage methods such as landmark-based, contact sensor-based, grid-based, graph-based etc. [27] proposes an on-line approach using spiral paths that give 62-67% coverage and [28] draws comparison between the back and forth and the spiral search method. Artificial Intelligence has also been explored as a possible solution

to the problem. [29] and [30] try to solve the problem by representing the destination as an attractive force and the obstacles as a repulsive force. [31] applies the concept to an agricultural field to steer through from one point to another. A non-learning neural network is used in [32] to generate collision free complete coverage solution for a dynamic and unstructured environment. Here each neuron represents a cell. The running time of this algorithm is $O(N^2)$ for a grid of N x N dimension. The resultant path is similar to that generated by a Boustrodephon decomposition. [33] extends the work done in [34] to get from one point to another by combining genetic and neural network algorithms to apply it on the coverage problem for an agricultural environment. The target space is converted to a 2D environment and each coordinate is a chromosome. Although this approach strives to achieve 90% coverage, the path generated is not the most efficient path.

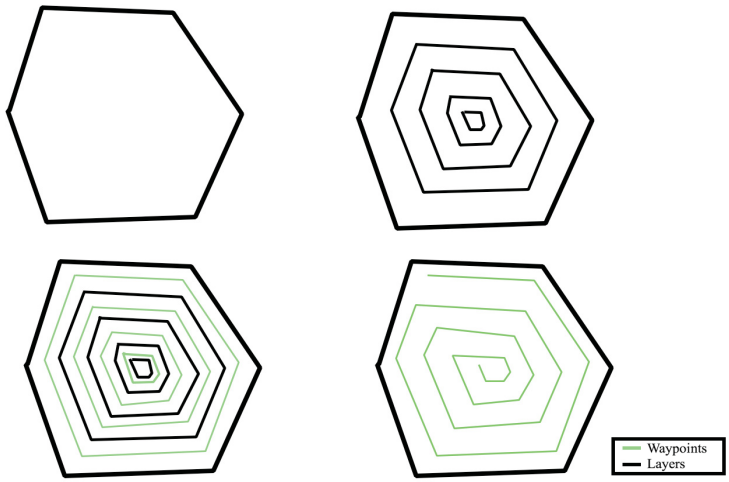

Figure 2.4: Spiral scanning patterns

The coverage problem can also be modeled as a TSP problem. The center of each cell in the grid is a node. Once this node is visited, the cell is considered visited. [35] and [36] use swarm-based algorithms to solve it. [37] combines cellular decomposition with TSP to reduce the size of the problem. Once the region is broken down into cells and a Boustrodephon

path determined to cover each cell, TSP is used to compute the order in which these cells should be visited.

Just like the coverage problem, the touring problem is a well researched area too. Algorithms such as the Knigsberg Bridges problem, Chinese Postman problem, Rural Postman problem try to find the shortest path to visit all or a set of edges in a graph. The Knigsberg Bridges problem seeks to find out a path to cover seven bridges connected in a certain way that visits each of them exactly once. Leonard Euler determined the conditions necessary for such a path to exist and called such a tour the 'Euler Cycle' [38]. [39][40] present algorithms to solve this for undirected, directed and mixed graphs. The Chinese Postman problem (CPP) refers o the problem of finding a subset of edges with minimum length that can be added to a graph to make it Eulerian [41]. The Rural Postman Problem is an extension of CPP where only a set of the edges must be visited [42]. The Generalized Directed Rural Postman Problem deals with clusters of arcs or edges, and the goal is to find the shortest path that traverses at least one edge from each cluster at least once [43], [44] . There are different variations to the above problems and various kinds of formulations have been adopted to solve them.

Another class of such problems involves visiting vertices. The Hamiltonian circuit problem, TSP and its variants like TSPN, GTSP etc focus on finding the shortest tour to visit all or a set of nodes in a graphs. The Hamiltonian circuit problem tries to find a path to visit all vertices in a given non-weighted graph exactly once. This is an NP complete problem. The objective of a TSP problem is to find the shortest path to visit a set of given nodes exactly once in the shortest possible time. This is an NP hard problem even for the special case where the nodes in the graph represent points on the euclidean plane [45]. For the Euclidean version of the problem, there exist polynomial time approximation schemes [45][46] i.e., for any $\epsilon > 0$ there exists a polynomial time algorithm which guarantees an approximation

factor of $1 + \epsilon$. A generalized version of the classic TSP is introduced in [47] to visit a set of neighborhoods instead of nodes. This is commonly known as the Traveling Salesman Problem with neighborhoods (TSPN). [48] gives the first constant-factor approximation algorithm for TSPN. Although the constant-factor is not exactly computed or optimized, it is hypothesized that it can be reduced to $(2 + \epsilon)$. Similar to the Generalized RPP, there exists a Generalized TSP Problem where there are a number of clusters of nodes and the goal is to find the shortest path to visit exactly one node from each cluster [49]. A problem of interest to this work is the Zookeeper Route problem, as it deals with computing the shortest path to visit a set of polygons exactly once, without entering them [50].

Although coverage problems and touring problems have been studied extensively separately, not a lot of literature is available on optimal path planning to visit and cover neighborhoods. [51] uses a constant factor approximation algorithm that combines Boustrophedon cell decomposition and TSPN to solve the problem addressed in this work.

Chapter 3

Algorithm Description

This chapter presents the approach adopted for solving the path planning problem for this application. It details an algorithm combining the routing problem and the coverage problem. It also attempts to define boundaries to this solution in terms of the optimal solution. The following assumptions have been made about the environment:

1. The environment does not have obstacles
2. The terrain does not have any significant variations in height, making this a 2D problem.

The first step in this algorithm is to discretize each region into cells based on the dimension and resolution of the spray applicator. Since every region needs to be covered completely, the entry and exit points have been restricted to only the vertices of these rectangles. The next step is to determine the optimal order in which these regions are to be visited. These two are combined to construct the final tour.

3.1 Coverage

The coverage of these areas is achieved by breaking down the regions into cells. These cells are then traversed in a back and forth motion to cover the entire region. [52] and [25] shows some of the work done to determine the optimal direction to scan a polygon. A factor that affects the time to complete a coverage scan is the number of turns. Every turn involves deceleration, turning, and acceleration. Setting the scan direction, parallel to the longer edge of the rectangle reduces the number of turns, thereby reducing the time to cover these regions. There might also be platform specific instabilities associated with these turns (some of which have been observed for the UAV in this work), which are reduced using this constraint. The entry and exit points to the regions are restricted to the vertices of the rectangles. Depending on the dimension of the region, there will be an even or odd number of scans.

3.2 Touring

In order to compute an optimal tour to visit the given regions, a TSP is first solved to visit the centroids of each region. This gives us the order in which to visit the regions. We now have a path to cover each region, possible entry and exit points and the optimal order. Next, the shortest tour to visit all regions is computed using the following approaches:

(i) TSP - Greedy

(ii) TSP - Dijkstra's - an approach combining algorithms for TSP and Dijkstra's shortest path calculation.

3.2.1 TSP-Greedy

This is an algorithm based on the dynamic programming approach as used in [51]. A cost function $C(i, s_i)$ is formulated to calculate the cost of entering each region at a certain vertex.

$$C(i, s_i) = \min_{t_i-1}[\min_{s_i-1}(C(i-1, s_i-1)) + d(t_i-1, s_i)], \tag{3.1}$$

where $s_i - 1$ and $t_i - 1$ are the entry and exit respectively to the previous region $R_i - 1$ and d(x,y) is the Euclidean distance between points x and y. Using the optimal order of visiting the regions, as obtained at the beginning of this section, the cost of entering any other regions R_i is calculated as the sum of the minimum cost of entering region at $s_i - 1$ and the shortest distance from the its exit $t_i - 1$ to nearest vertex of the next region, acting as its entry point s_i.

3.2.2 TSP-Dijkstra's

In this approach, the edge weight of every polygon represents the cost of entering the the region at one end of the edge, covering the region and exiting through the other end of the edge. For example, the cost of the edge AB of the rectangle in Fig 3.1 (a) is originally 4 . In the new representation, the weight of edge AB is 16 - the sum of the cost to cover the region (Fig 3.1 (b)) and the original weight of edge CB (Fig 3.1 (c)). Since for every vertex of entry there exists 4 vertices of exit, every rectangle can be broken down into four separate representations, signifying 16 possible ways to enter and exit a rectangle or a quadrangle. Fig 3.1 (d) shows the the 4 representations.

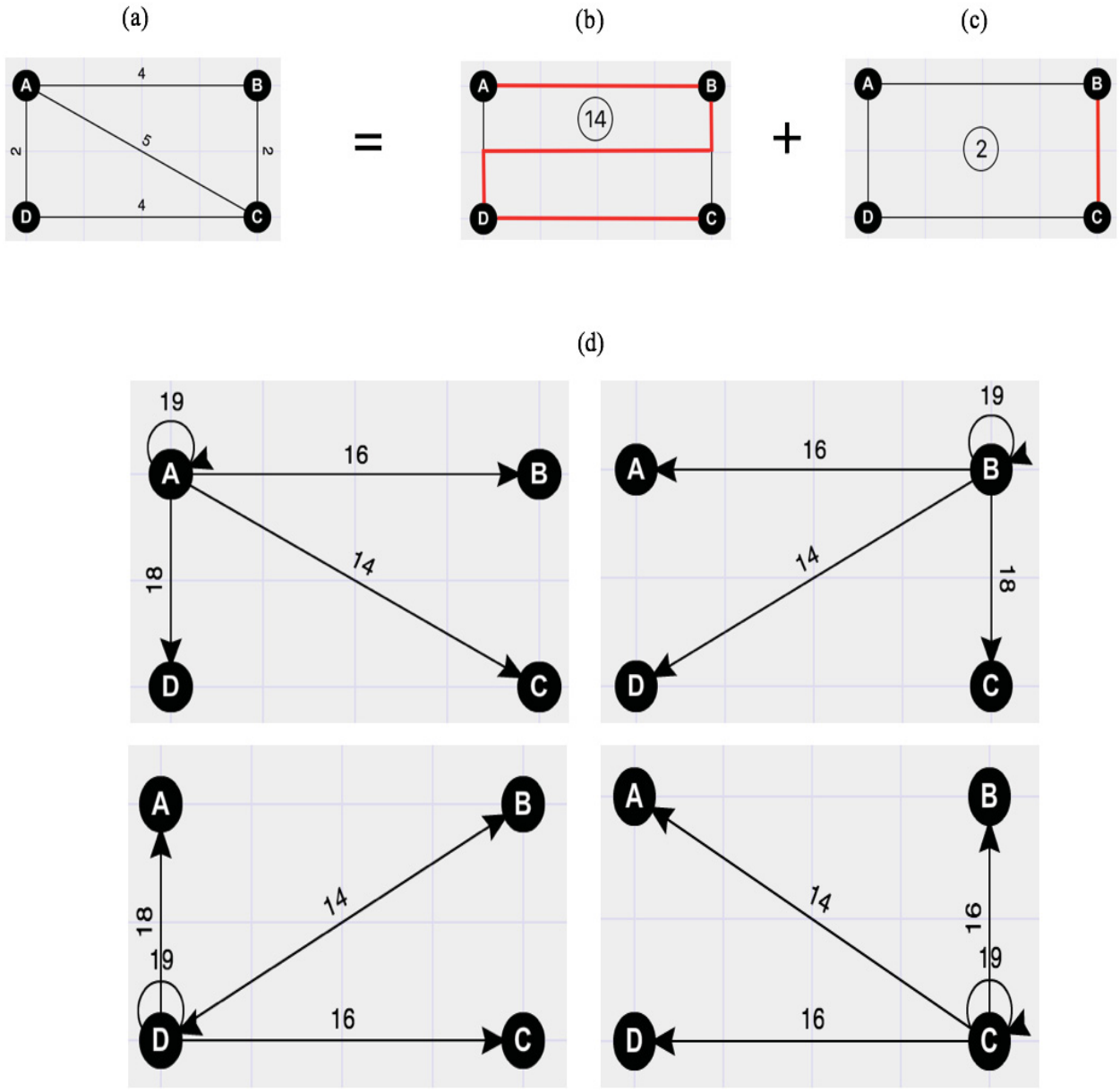

Figure 3.1: Calculation of new edge weights

Once the regions are converted to the new representation, the following constraints are implemented.

(i) The vertices of the first region (as obtained by using algorithm for TSP) are the neighbors to the start node.

(ii) Once a region is entered using a vertex, it must be exited before proceeding to the next region.

(iii) The neighbors of a vertex used for entering a region are the vertices of the same region.

(iv) The neighbors of a vertex used for exiting a region are the vertices of the next region.

(v) The goal node is the neighbor to the vertices of the last region.

Next, Dijkstra's algorithm is implemented, beginning at the start node, to reach the goal node while entering, covering and exiting each region using the shortest path. The algorithm is described in Algorithm 1.

Algorithm 1 Modified Dijkstra's Algorithm

```
1: entryFlag = 1
2: exitFlag = 1
3: for each node n in graph do
4:     n.distance = Infinity
5:     n.parent = Undefined
6: end for
7: List = Neighbors of Start node
8: Set start node to be parent node of all its neighbors
9: while List not empty do
10:    Current = Node in list with smallest distance, remove current from list
11:    if exit region condition then
12:        entryFlag = entryFlag'
13:        exitFlag = exitFlag'
14:    end if
15:    for each neighbor n of current do
16:        if n.distance ≥ current.distance + length of edge from n to current then
17:            n.distance = current.distance + length of edge from n to current
18:            n.parent = current
19:            Add n to list if it isn't already there
20:        end if
21:    end for
22: end while
```

3.3 Performance

[58] proves that a route to cover multiple convex polygons can be generated, if these are attached to the boundary of an enclosing simple polygon.

Lemma 1. *Let R = R1,R2, . . . ,Ri, . . . ,Rn be a set of convex regions located along the perimeter of a simply connected polygon P. There exists an optimal solution for visiting the regions in R which visits them in the order they appear along the boundary of P.*

Therefore, for a special case of this problem where the identified areas in the target field can be enclosed in a simple polygon, this algorithm gives us an α-approximate solution with $\alpha = 1$.

Now, let's assume that an optimal tour OPT exists that visits and covers each of these regions in minimum time. Let τ_R^* be the optimal tour that visits all the regions.

$$|OPT| \geq |\tau_R^*| \tag{3.2}$$

Now, let $|C_{R_i}^*|$ be the optimal coverage tour to cover region R_i. Again,

$$|OPT| \geq \sum_{R_i \in R} |C_{R_i}^*| \tag{3.3}$$

Assuming that the algorithm for computing τ_R is an α-approximation and that for computing the coverage tour within the regions is a $\beta - approximation$, then

$$\tau \leq \alpha|\tau_R^*| + \sum_{R_i \in R} \beta|C_{R_i}^*| \tag{3.4}$$

From equations, (3.2), (3.3) and (3.4)

$$|\tau| \leq \alpha|\tau_R^*| + \sum_{R_i \in R} |C_{R_i}^*| \leq \alpha|OPT| + \beta|OPT| \tag{3.5}$$

$$\therefore |\tau| \leq |(\alpha + \beta)OPT| \tag{3.6}$$

$$i.e, |\tau| \leq |2OPT| \tag{3.7}$$

Chapter 4

System Description

Our system consists of a hexacopter, a spray tank, sprayer assembly (pipes, valves, nozzles etc), RTK-GPS and a ground Control station.

4.1 Hexacopter

A hexacopter was chosen for carrying out our spraying applications. The dimensions of the copter are 19" x 38" x 38" . The copter weighs 19 pounds and can carry a payload of approximately 3.3 pounds.

The flight controller on the hexacopter is a Pixhawk 2.0. It has six KDE 35 A electronic speed controllers for each of the six Tarot 5008 340kv Brushless Motor. The propellers have a dimension of 18" x 6.5" and are from Xoar.

Figure 4.1: Hexacopter Platform

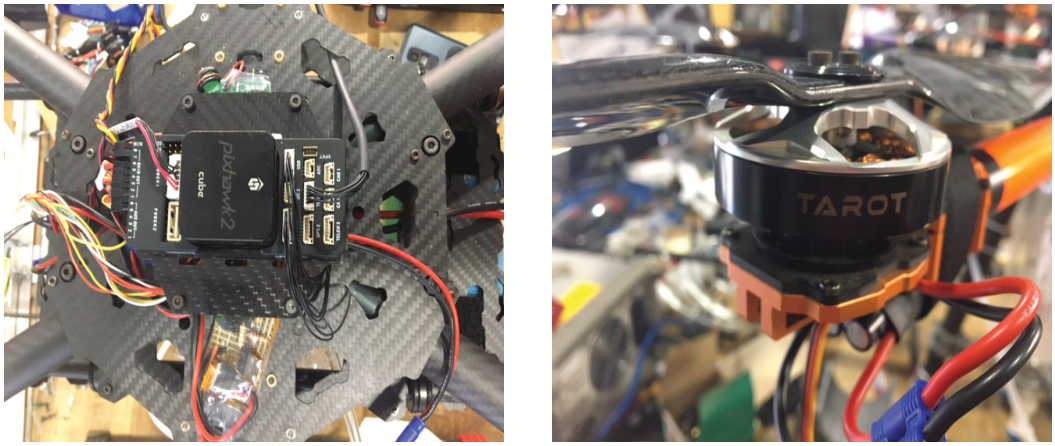

Figure 4.2: On-board hardware(a) Pixhawk Flight Controller (b) Tarot Motors and Xaro popellers

4.2 Sprayer Assembly

The sprayer assembly consists of (i) a spray tank (ii) spray applicator (nozzles, pipes and valves). The assembly is attached to the hexacopter at the bottom-center, with a metal rod with extensions on both sides to form slots to hold six 500 ml bottles. This design allows for the bottles to be locked in once inserted into the slots facing the ground. We use water in the spray tank for our experiments. The water in the tank is pressurized to 85 psi and the valves give us a pressure of 30 psi at the nozzles. The outside diameter of the pipes carrying

the water from the tank to the nozzle is 1/4". A relay output from the pixhawk controller powers a solenoid to operate the valves connected to the nozzle.

Figure 4.3: Sprayer sssembly (a) Spray Tank (b) Spray Applicator

4.3 RTK-GPS

The nature of the application requires high accuracy positioning systems. RTK-GPS has proven to give accuracies of the order of cm. We use the Swift Nav Piksi Multi which claims to give a horizontal accuracy of 2 cm and a vertical accuracy of 6 cm. The accuracy degrades at a rate of 1mm horizontal and 3mm vertical for each km between the base and rover. One set of the Piksi Multi receiver and radio is used as stationary base station and another set is set on-board the copter. RTK solution is achieved by transmitting GPS corrections from the base station receiver to the copter receiver using the radio links.

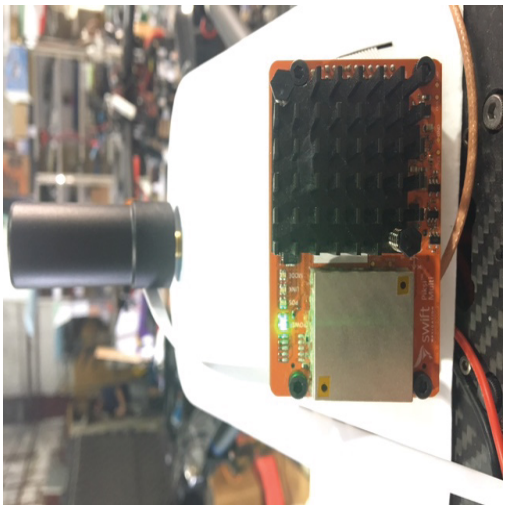

Figure 4.4: On-board RTK receiver

4.4 Ground Station

The ground station is a laptop with the APM mission planner running on it. The path planning is done offline. It is safe to assume the terrain to be a flat surface while converting the image co-ordinates to GPS coordinates. A file in the '.waypoint' format is generated with commands for (i) Take-off and Land, (ii) Go to Waypoints, and (iii) Operating the sprayer while scanning a region and uploaded to Mission Planner. This is then used to fly the hexacopter autonomously.

Chapter 5

Experiments and Results

This chapter first describes empirical experiments conducted to compare the path planning approaches from Section 3.2.2. It then presents experiments and results from the hardware platform described in Chapter 4 using the software developed.

5.1 Empirical comparison of TSP-Greedy and TSP-Dijkstra's Algorithms

An empirical comparison of the two approaches to solve the problem statement is presented in this section. Both approaches are implemented on Matlab. Five test cases are prepared and results for both approaches are generated. It is assumed for these test cases, that the number of back-and-forth movement required to cover each region is odd, i.e, the coverage path for any region extends from one vertex of the region (i.e, rectangle) to the diagonally opposite vertex. Additionally, all definitions described in section 3.2.2 apply here. Figure 5.1 illustrates the first test case. Fig 5.1 (a) is the original graph. Figure 5.1 (b) is the new

representation of of the same, following the rules described in section 3.2.2. Figure 5.1 (c) shows the path generated by the TSP-Greedy algorithm (TSP-G) and the TSP-Dijkstra's algorithm (TSP-D). Figure 5.2 and 5.3 show more test cases with column (a) showing the new representations of the graphs, and column (b) showing their respective paths for TSP - Greedy and TSP - Dijkstra's.

Figure 5.4 presents a table comparing the total distance of the path generated by both approaches. It is observed, that as predicted theoretically, the path generated by TSP-Dijkstra's is less than or equal to the the path generated by TSP - Greedy in all five cases.

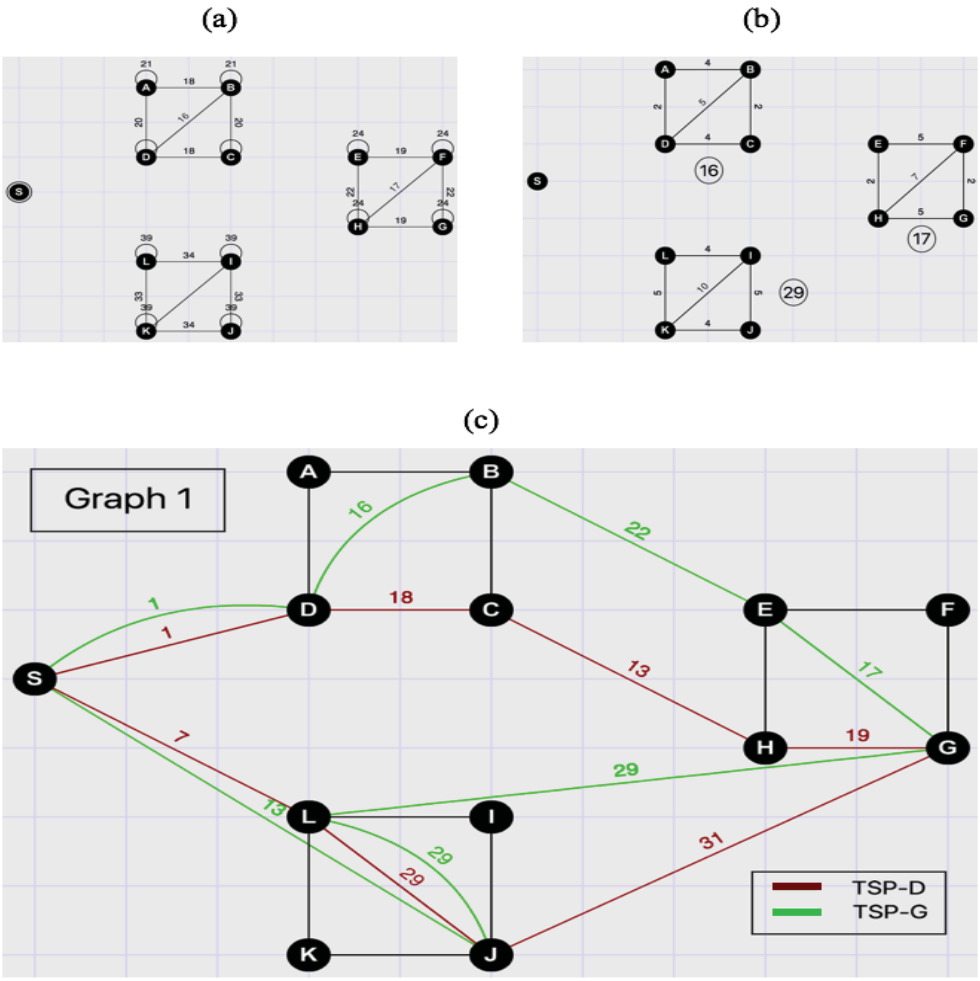

Figure 5.1: TSP - Greedy and TSP - Dijkstra's comparison - 1

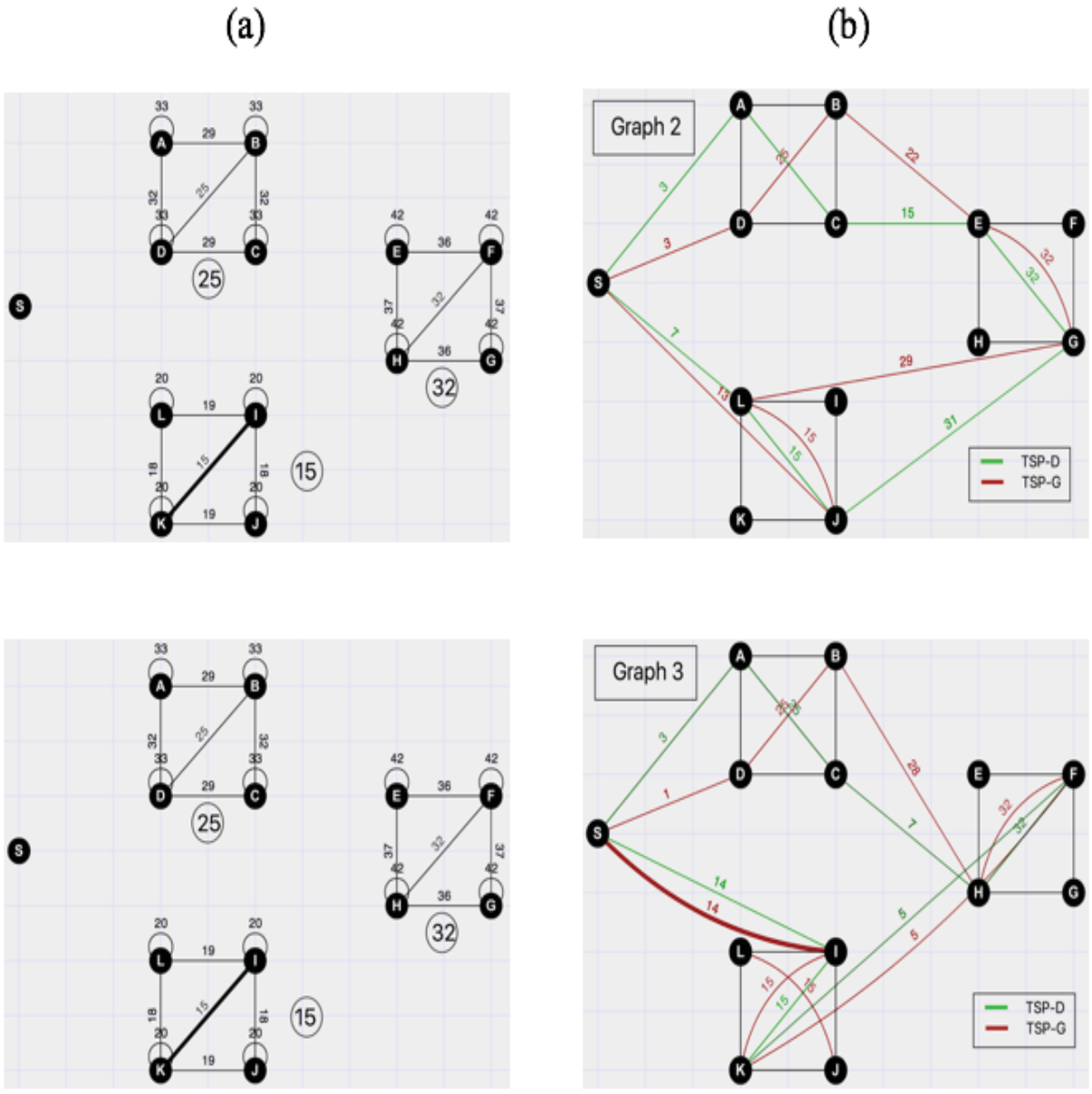

Figure 5.2: TSP - Greedy and TSP - Dijkstra's comparison - 2

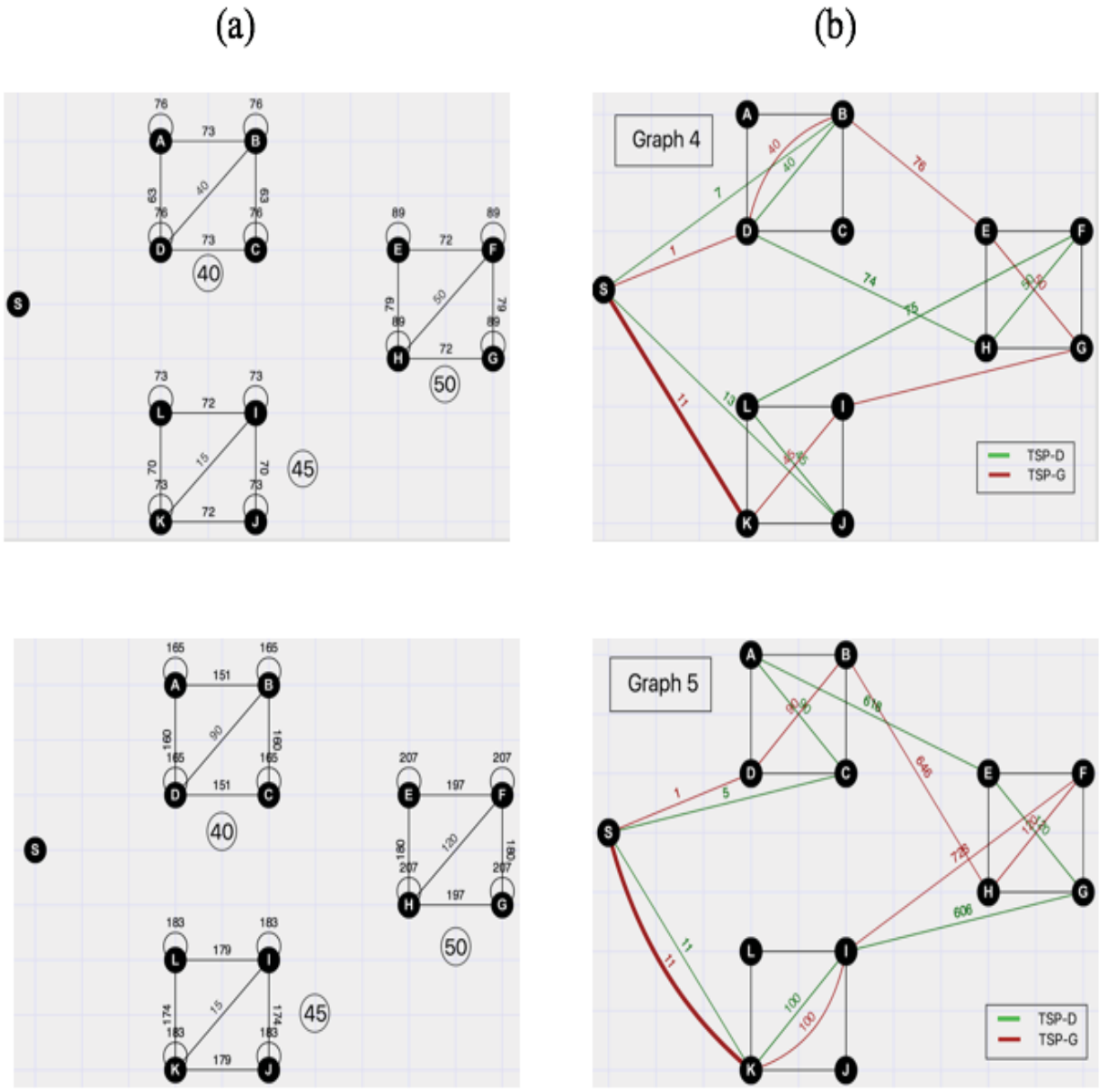

Figure 5.3: TSP - Greedy and TSP - Dijkstra's comparison - 3

Graph	TSP-Dijkstra's	TSP-Greedy
1	118	127
2	128	138
3	101	120
4	304	304
5	1550	1694

Figure 5.4: Comparison of distance of path obtained from TSP - Greedy and TSP - Dijkstra's

5.2 Hardware experiments and results

The TSP-Greedy algorithm is modified to interface with Mission Planner to control and navigate a spraying hexacopter (Section 5.2). This section described results from test cases prepared to test

(i) Path planning algorithm.

(ii) Waypoint file generation.

(iii) Path planning using results from weed detection

5.2.1 Phase 1: Path Planning Algorithm

This test case investigates path planning by the TSP-Greedy algorithm. The algorithm is implemented on Matlab. The inputs used are (i) Two images with three arbitrary convex simple polygons marked (Figure 5.5) and (ii) Vertices of the polygons . The code works as expected and produces the shortest path within bounds of optimality. Figure 5.6 shows the path generated.

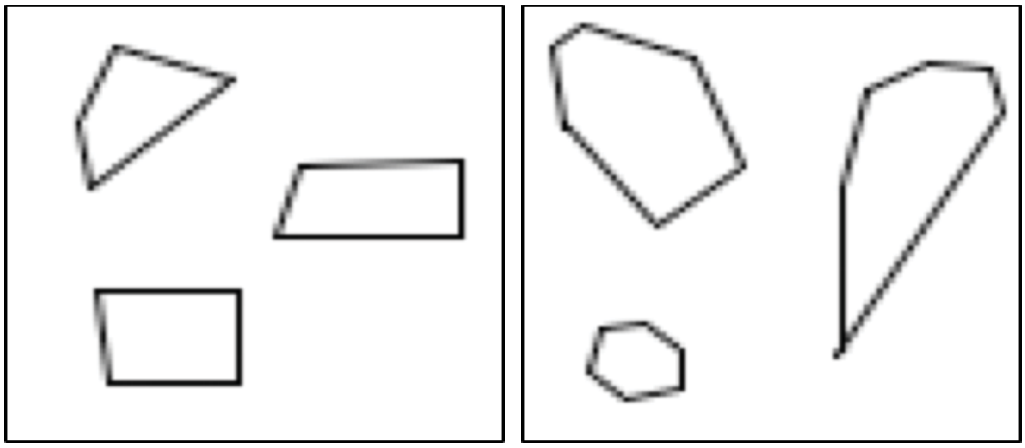

Figure 5.5: Images with arbitrary convex polygons

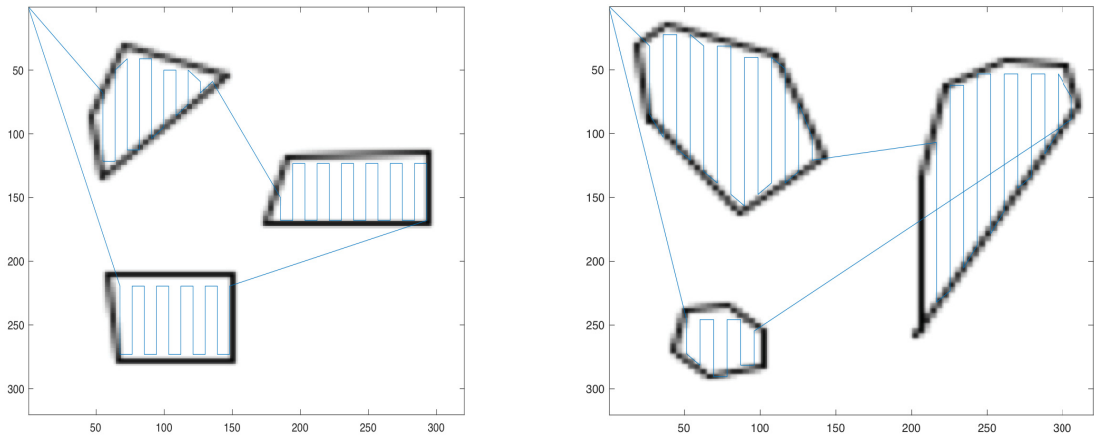

Figure 5.6: Path generated for polygons in Fig 5.5

5.2.2 Phase 2: Waypoint File Generation

This test case investigates the path planning for polygons marked on actual satellite images and the interfacing with Mission Planner. The input for this test are (i) Satellite image of target areas marked as polygons (Figure 5.7) and (ii) Vertices of the polygons. As described in 4.4, a waypoint file is generated using Matlab, which is then used by Mission Planner to control and navigate the UAV.

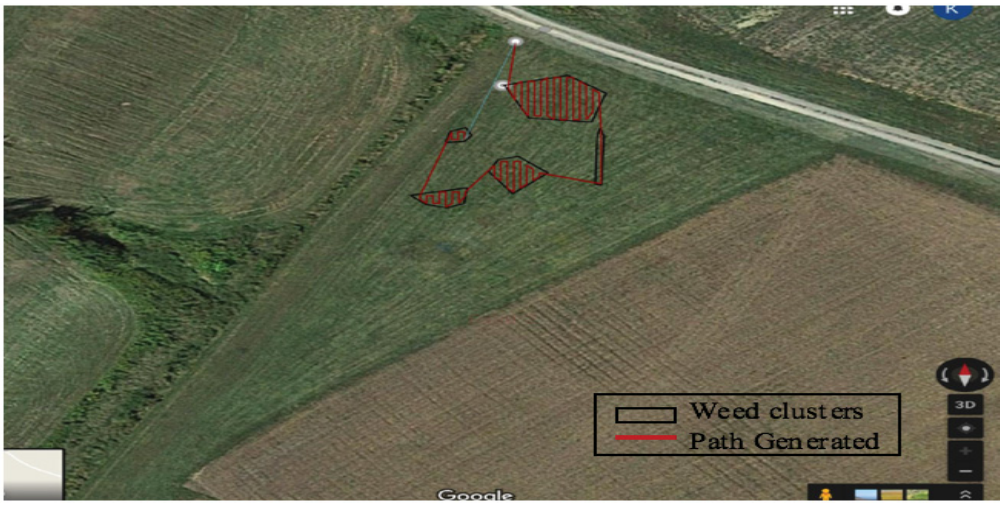

Figure 5.7: Path generated for satellite image of a patch of field in Kentland farms. Polygons to be scanned are marked in black. The red lines show the path generated on the image

The image coordinates from the path planning is shown in Figure 5.7. These are converted to GPS coordinates and a waypoint file is generated. Figure 5.8 shows the path generated by the said waypoint file on Mission Planner.

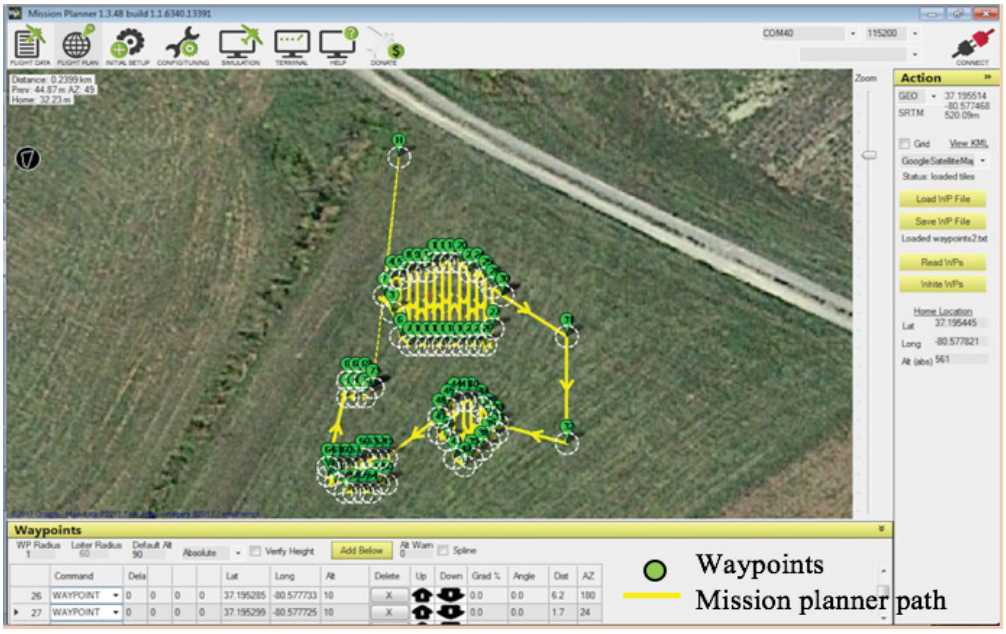

Figure 5.8: Waypoints on Mission Planner

5.2.3 Phase 3: Path Planning Using Results From Weed Detection

The flight test from phase displayed instabilities while approaching and leaving waypoints within the regions. For this phase, the number of turns are reduced by setting the scan direction parallel to the longest edge of the polygon. To further reduce the effect of such instabilities, the position and attitude outputs are studied for a series of position inputs. Setlling time, i.e., the time taken by the UAV to stabilize after reaching a waypoint is calculated. This process is further detailed in Chapter 6. This time is then used to construct a 'Buffer Zone' around each region, allowing the UAV to settle before spraying on the target regions. It is calculated that this settling time corresponds to approximately 0.5 meters which is 6 pixels on the image used. As a precautionary measure, the 'Buffer' is set to 10 pixels.

This test uses results of weed detection on an agricultural field obtained by another graduate student of the Unmanned Systems Lab at Virginia Tech. Figure 5.9 (a) is the image used for weed detection. Figure 5.9 (b) is the result of the weed detection algorithm. These weeds need to be clustered in order to identify areas in the form of polygons (rectangles in this work). The clusters are marked in red. These clusters are then fitted to form rectangles. This data is overlaid on a more conducive area for testing purposes. Figure 5.9 (b) shows the rectangles overlaid on an area in Kentland farms. The inputs for this test are (i) A satellite image of an area in Kentland farms (Figure 5.9 (c)) and (ii) Vertices of the polygons.

Figure 5.10 is the path generated by the path planning code and Figure 5.11 is the flight path on Mission Planner. Due to limits on flight time arising from battery and spray tank capacity, two of the bigger regions are ignored for this test. Results show that this software is capable of handling spraying necessities for big regions, spots and single rows. Figure 5.12 shows the actual path followed (purple lines).

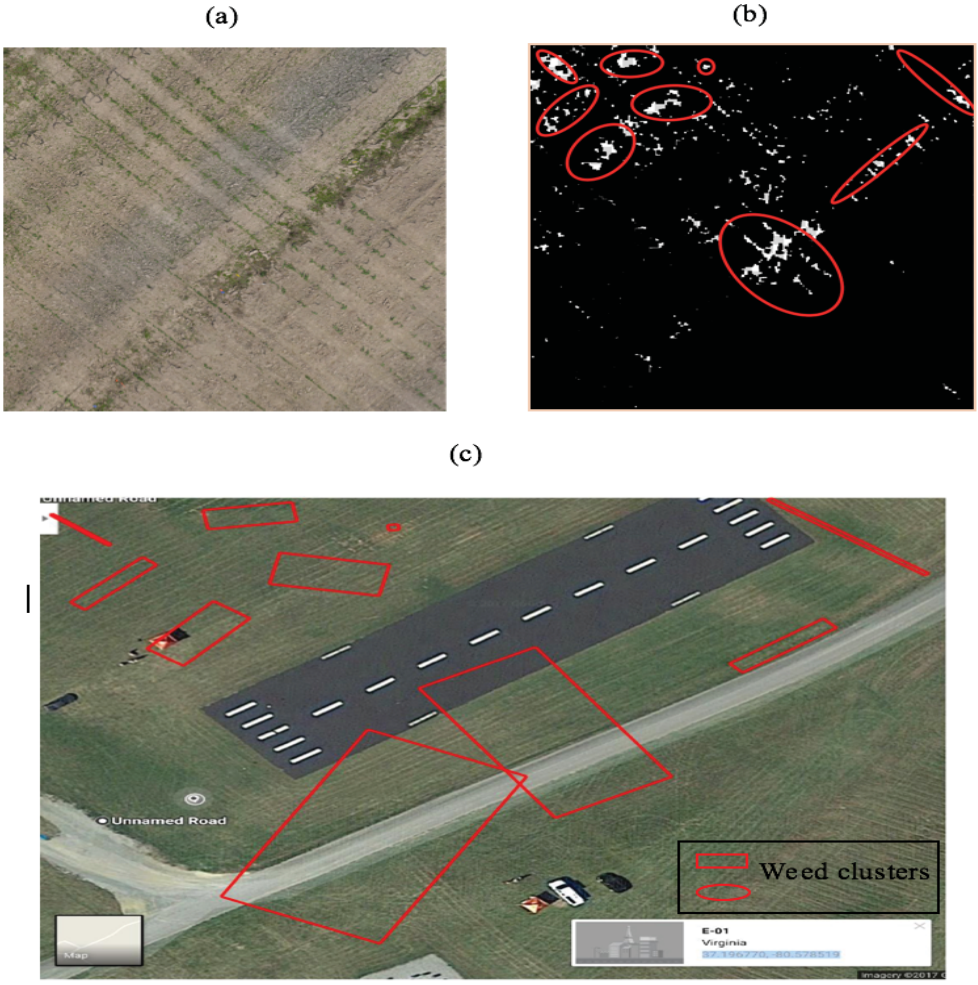

Figure 5.9: Weed detection and clustering

Figure 5.13 shows a comparison of the GPS coordinates as generated by the software (green) and the GPS coordinates actually followed (red) with an empty spray tank, and Figure 5.14 shows a comparison of the paths with a full spray tank and with commands for spraying the marked regions.

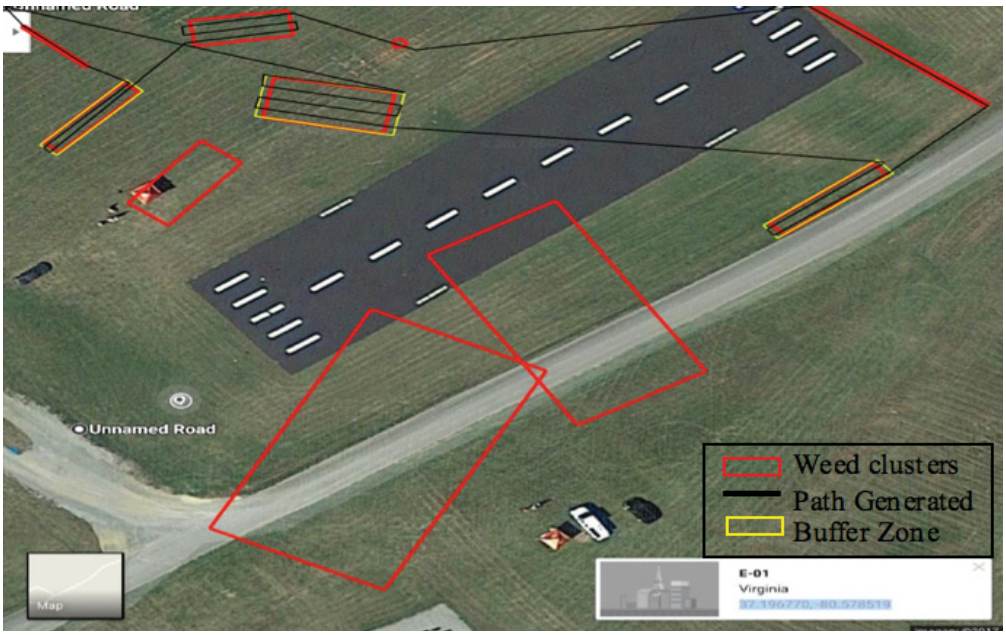

Figure 5.10: Path generated by software

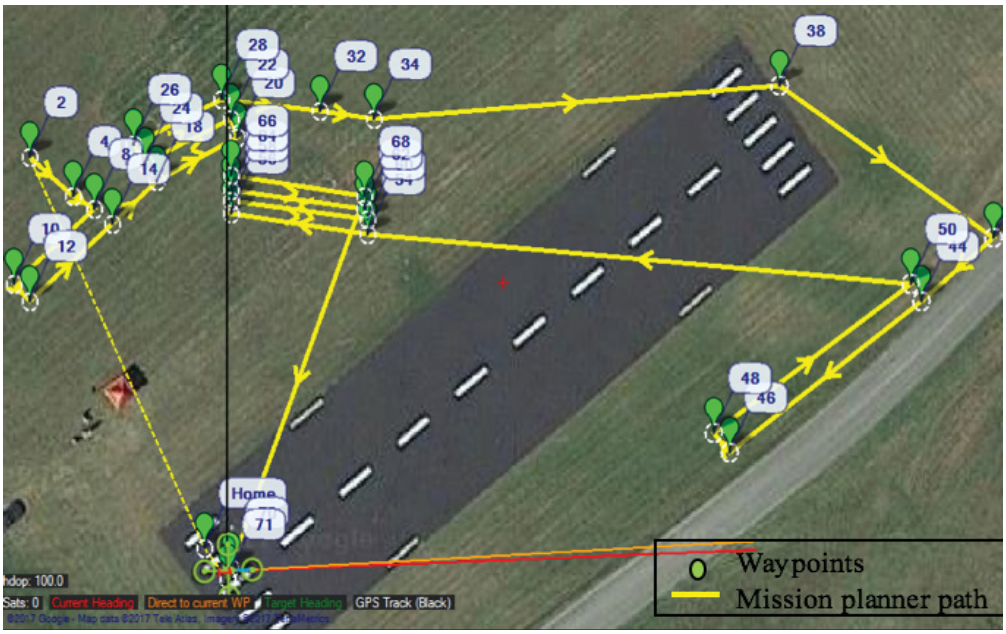

Figure 5.11: Path on mission planner

Figure 5.12: Actual Path followed

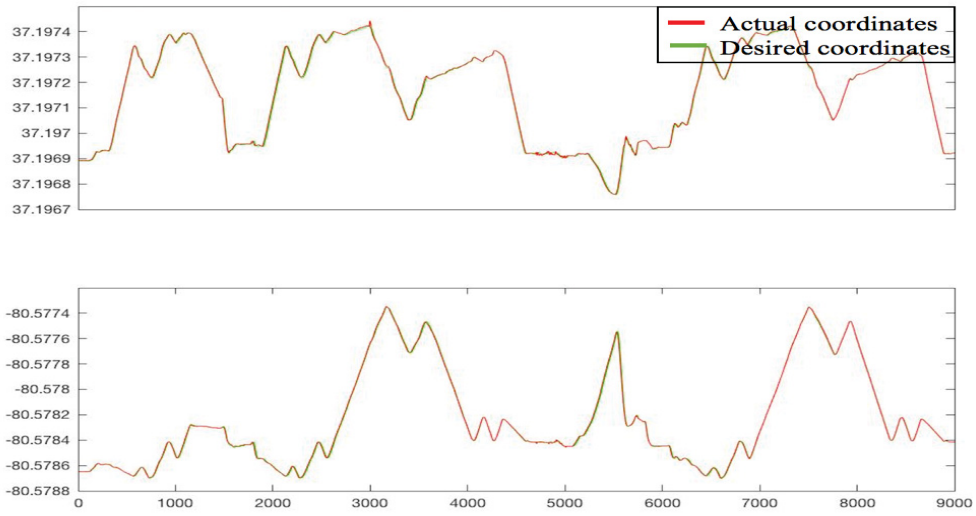

Figure 5.13: Comparison of GPS coordinates for a mission without payload and spraying action

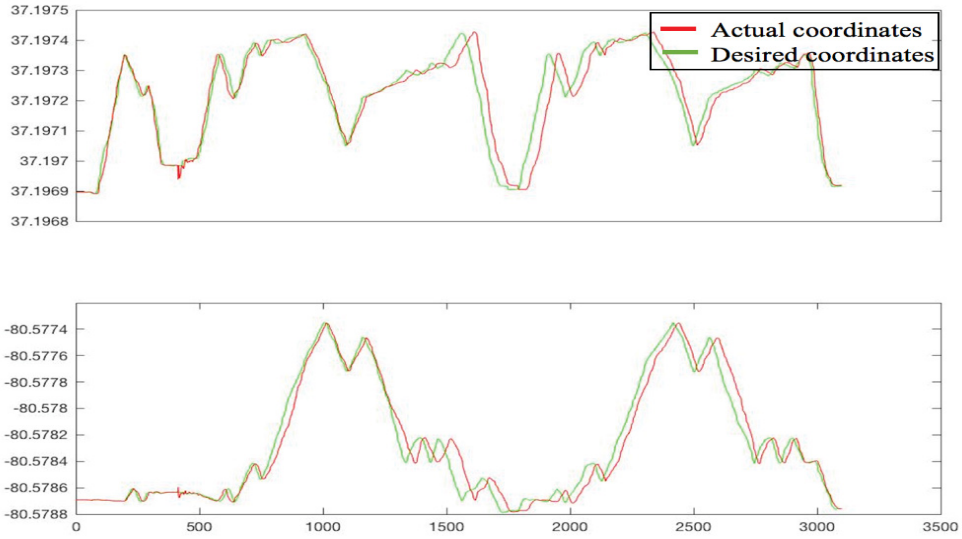

Figure 5.14: Comparison of GPS coordinates for a mission with payload and spraying action

Chapter 6

System Analysis

Test flights conducted for experiments detailed in Chapter 4 showed that the spraying hexacopter being used displayed instabilities while approaching and leaving waypoints. In other words, instabilities were observed during acceleration and deceleration actions, while navigating from one waypoint to another. This chapter presents an analysis of position, velocity and acceleration data to characterize the system and propose methods to stabilize the UAV.

6.1 Data Collection

For this analysis, the the system is given a series of position inputs as shown in Figure 6.1. A path is set for the UAV such that it tracks a loop in a shape of a square 4 times. Sixteen waypoints or sixteen position inputs are given for this path.

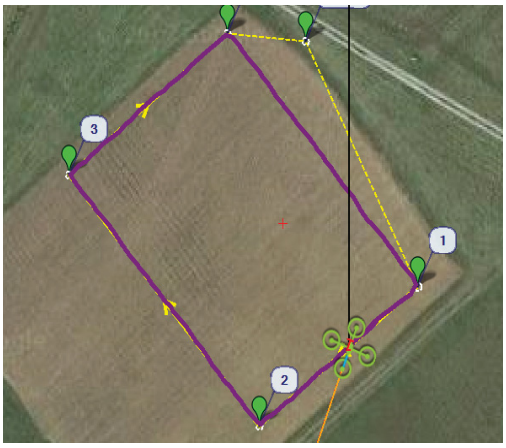

Figure 6.1: Flight path to study position and attitude outputs

6.2 Settling Time

Settling time is the time required for an output to reach and remain within a given error band following some input stimulus (Figure 6.3).

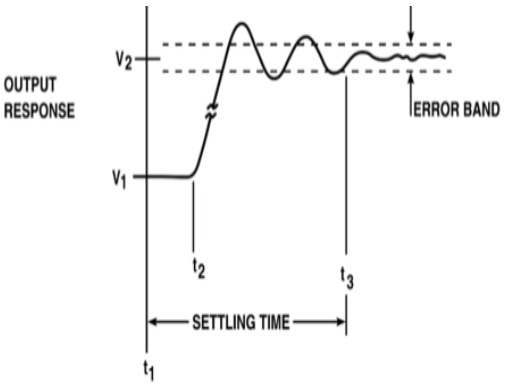

Figure 6.2: Settling time

Figure 6.3 shows the graph of (a) Latitude, (b) Longitude, (c) Roll angle and (d) Pitch angle. There a total of 16 troughs and peaks in the longitude and latitude plots account for the sixteen waypoints. As indicated by the blue dashed lines in the figure, every waypoint is associated with an an overshoot or undershoot in roll and pitch in the time axis.

From the data in Figure 6.3, some overshoots are chosen at random and a settling time

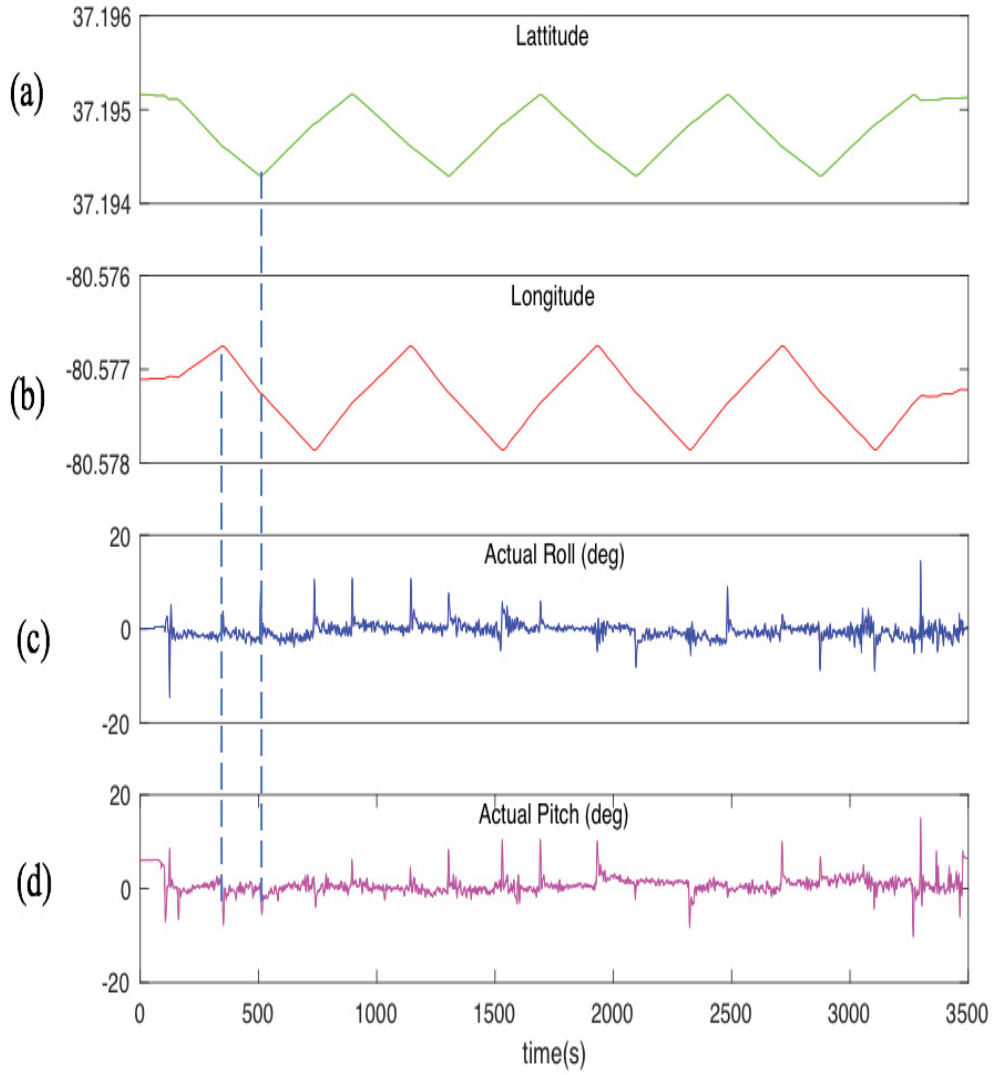

Figure 6.3: Navigation and orientation data - 1 (a) Latitude, (b) Longitude, (c) Roll angle and (d) Pitch angle

analysis is performed on them. It is observed that after every excitation or position input, the system takes approximately 0.25 s to stabilize (Figure 6.4). The settling time and the error band are marked in the figure.

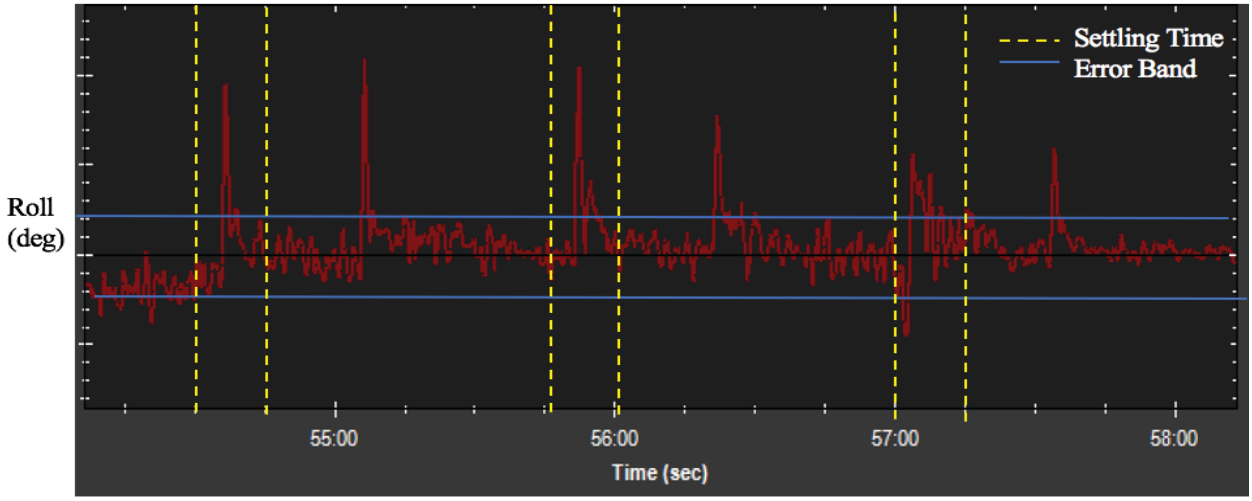

Figure 6.4: Settling time of the sprayer hexacopter

6.3 LQR Control

The on-board pixhawk controller is a two level cascade controller (Figure 6.5). The first level - the position controller, uses estimates obtained using sensor data and setpoints (position, velcity etc.) issued by Mission Planner to calculate desired position, velocity and acceleration values. The second level - the attitude controller uses inputs from the position controller to generate values for orientation (Figure 6.6).

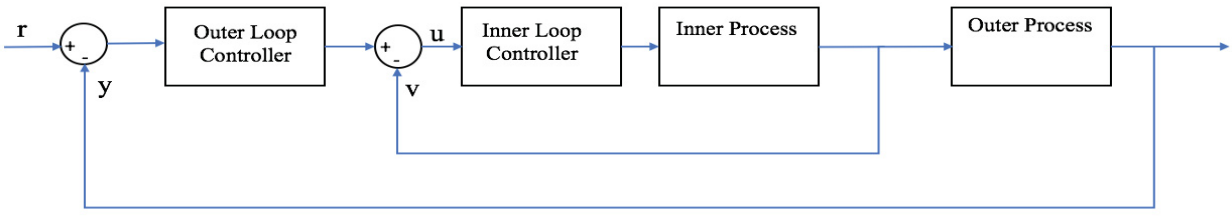

Figure 6.5: General block diagram of cascade control

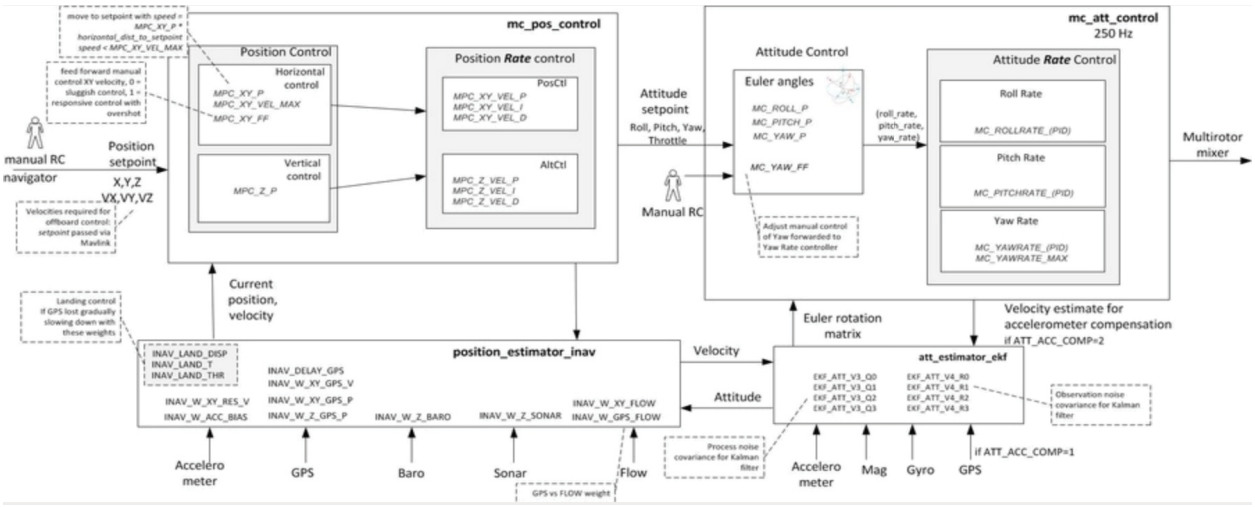

Figure 6.6: Pixhawk multirotor cascade control

Figure 6.7 shows graphs of (a) Latitude, (b) Longitude, (c) Roll, (d) Pitch, (e) Actual Velocity(x), (f) Actual Velocity(y), (g) Desired Acceleration(x) and (h) Desired Acceleration (y). As a waypoint is approached, the controller on the UAV prepares for the next waypoint. The controller calculates the values of all parameters required to effect the velocity change to reach the next waypoint. Since this application requires a constant velocity while spraying a region (for uniform distribution), the acceleration is a high impulse signal. This theory is corroborated by the data obtained from the test flight. As marked in Figure 6.7, every waypoint in Figures 6.7 (a) and 6.7 (b) is associated with a change in velocity and acceleration (in the form of an impulse signal) just before it. Intuitively, an impulse signal can be thought of as a blip (in time). The derivative of this impulse would therefore be two blips, just before and after the impulse signal. This could also be thought of as a 'Force couple' that results in rotation, but no translation. This is the phenomenon behind the sharp 'jerk' (derivative of acceleration) observed, just as a waypoint is approached.

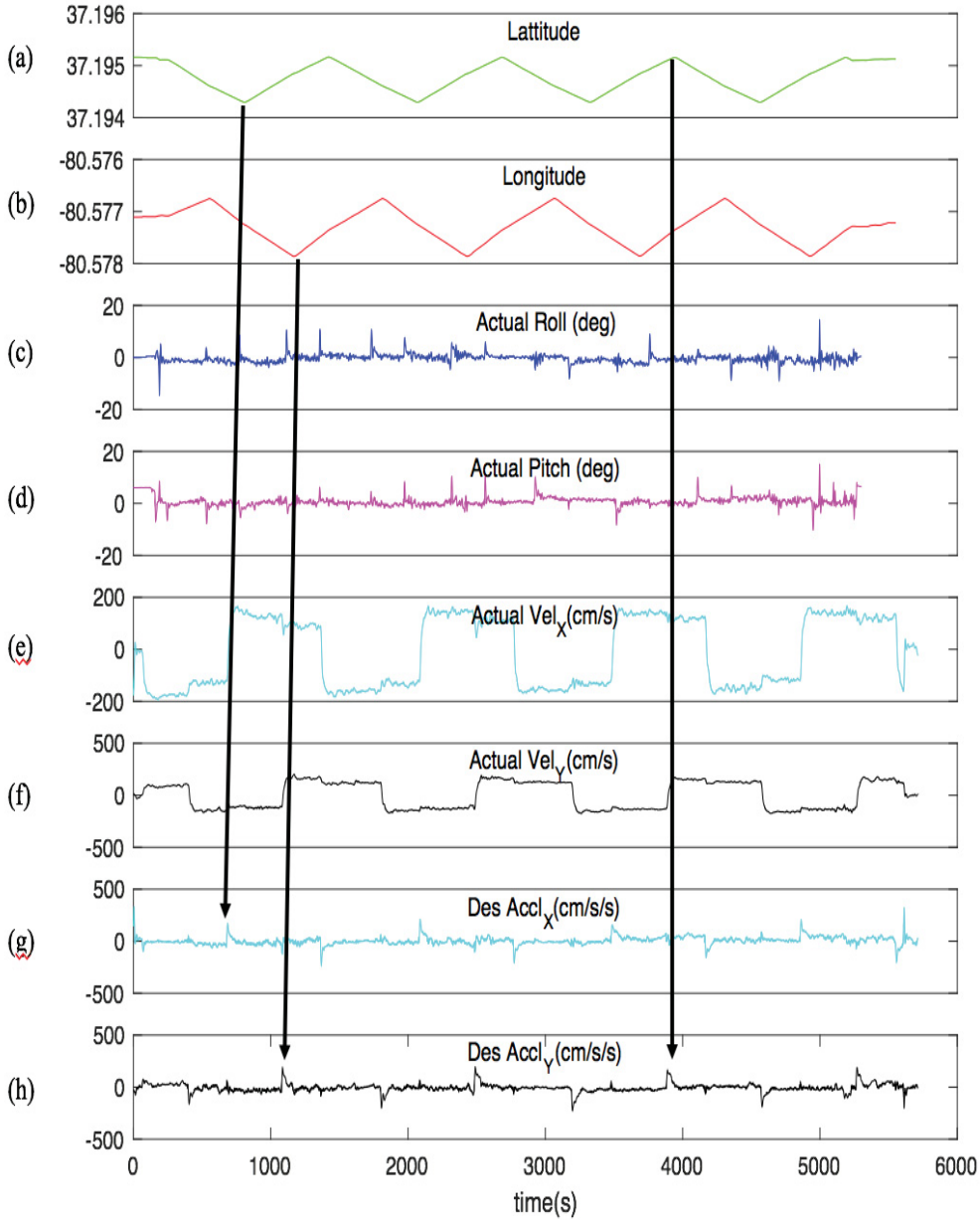

Figure 6.7: Navigation and orientation data - 2 (a) Latitude, (b) Longitude, (c) Roll, (d) Pitch, (e) Actual Velocity(x), (f) Actual Velocity(y), (g) Desired Acceleration(x) and (h) Desired Acceleration (y)

The desired acceleration values as shown in figures 6.7 (g) and (h) are generated by the PID controllers in place in the Pixhawk controller. The control parameters like the proportional, integral and differential gains can be tuned to reduce the magnitude of the observed over-

shoots. Additionally, the controller configurations can be improved to account for all values of payload. Although this would improve performance, it isn't a sufficient solution. The solution to this is an approach, that in addition to minimizing position, velocity, and yaw errors, regulates the roll and pitch values to a low setpoint.

A Linear Quadratic Regulator (LQR) is an approach in optimal control that can be applied to be a problem. The system in question should be linearized around the the desired configuration before applying LQR. Once linearized, the system can be represented as

$$\dot{\mathbf{X}} = \mathbf{AX} + \mathbf{BU} \tag{6.1}$$

$$\mathbf{Y} = \mathbf{CX} + \mathbf{DU} \tag{6.2}$$

where $\mathbf{X}$ is the state vector, $\mathbf{U}$ is the input, $\mathbf{A}$ is the system matrix and $\mathbf{B}$ is the input matrix. The LQR approach is based on minimizing the cost function

$$V(\mathbf{G}) = \int_0^\infty \mathbf{X}^T(\tau)\mathbf{Q}\mathbf{X}^T + \mathbf{U}^T(\tau)\mathbf{R}\mathbf{U}^T \, d\tau \tag{6.3}$$

where $\mathbf{Q}$ is the state weighting matrix and $\mathbf{R}$ is the control effort weighting matrix. When $Q \ll R$, the cost function minimizes the amount of control effort needed (small gain). When $R \ll Q$, the cost function minimizes the state response without considering the magnitude of control effort requires (large gain). $\mathbf{G}$ is the control gain which is computed to minimize the cost function $V(\mathbf{G})$ with the set values of Q and $\mathbf{R}$. The solution to the Lurie - Riccati equation gives

$$\mathbf{A^P + PA - PBR^{-1}B^TP + Q = 0} \tag{6.4}$$

$$\mathbf{G = R^{-1}B^TP} \tag{6.5}$$

where $\mathbf{P}$ is the Riccati solution. This representation minimizes the state response as per the $\mathbf{Q}$ matrix. This can be converted to a representation that regulates the outputs $\mathbf{Y}$ instead of the state response, such as in

$$\mathbf{Q} = q\mathbf{C^TC} \tag{6.6}$$

$$\mathbf{R} = r\mathbf{I} \tag{6.7}$$

converting the cost function to (assuming $\mathbf{D=0}$)

$$V(\mathbf{G}) = \int_0^\infty q\mathbf{X}^T(\tau)\mathbf{C^TCX}^T + r\mathbf{U}^T(\tau)\mathbf{U}^T\, d\tau \tag{6.8}$$

or,

$$V(\mathbf{G}) = r\int_0^\infty (q/r)\mathbf{Y}^T(\tau)\mathbf{Y}^T + \mathbf{U}^T(\tau)\mathbf{RU}^T\, d\tau \tag{6.9}$$

This representation now needs two scalar constants q and r or only q/r.

For this problem, the outputs to be regulated are (i) Position error, (ii) Velocity error, (iv) Heading or yaw error (iii)Acceleration errors. While the position, velocity and yaw errors can be calculated as the difference between the commanded (changing with every waypoint) and actual values of GPS coordinates, acceleration (in x and y) errors can be calculated as the difference between the actual values and the desired value set. An LQR controller can be implemented in conjunction with a PID controller as described in [53], [54] and [55].

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This work presents an algorithm to combine two well researched problems i.e the shortest path problem and the coverage problem to construct a route that visits and covers multiple regions in the shortest time. Here, it is developed with a focus on precision agriculture. Majority of the existing path planning research in this area focuses on covering a single area efficiently and can therefore benefit from this algorithm for its various monitoring, protection and production applications. The main contributions of this work are,

1. **Software**

   (i) Two approaches to plan a time optimized path to tour and cover multiple regions.

   (ii) Proof that, if the target regions appear along the boundary of a simply polygon, the path generated by TSP - Greedy will not exceed twice the time consumed by the optimal path.

(iii) Empirical results (from five data sets) that show that the results of TSP - Dijkstra's are less than or equal to TSP -Greedy.

2. **Hardware**

(i) A system analysis that shows overshoots or undershoots corresponding to jerks observed while approaching waypoints during test flights. Upon studying unique system parameters, it is found that this is due to a sharp acceleration impulse signal before every change in waypoint and velocity.

(ii) An LQR control approach to be implemented in conjunction with the existing PID controllers to control the instability.

(iii) A settling time analysis to construct a buffer zone around target regions to accommodate inaccuracies in spraying until the system stabilizes

(iv) Performance investigation of RTK-GPS for full and no payload.

7.2 Future Work

Research in this area of path planning is still developing with a new application emerging everyday. This work develops a solution with a set of defined assumptions and restrictions. A number of new problems can be explored by removing one or more of the constraints. Future work for this problem involves

1. **Software**

(i) Path planning for polygons, concave polygons, overlapping polygons, non-simple polygons, non-polygons etc. Concave polygons can be broken down into convex trapezoids using Trapezoidal decomposition. Holes in a non-simple polygon can

be seen as an obstacle and then decomposed to trapezoids. Similarly, over lapping polygons can be merged to form one big polygon and then broken down.

(ii) Modeling the problem as a Generalized Rural postman problem. It is required to visit exactly one edge from each cluster, that constructs the shortest path. In this approach every region is modeled as a cluster of edges. The weight of each edge is the new eight as calculated for TSP - Dijkstra's. in this work only four edge weights are considered for each pairs of entry and exit points. This can be extended to included costs for coverage in different orientations.

(iii) Solving this problem for uneven terrain. With the current assumption of flat surface, this is a 2D problem. Including variations in height converts this to a 3D problem to be solved.

2. **Hardware**

Implementation of optimal control approaches such as the LQR control detailed in Chapter 6 to achieve improved stability.

Bibliography

[1] Thomas Robert Malthus. *An essay on the principle of population: or, A view of its past and present effects on human happiness*. Reeves & Turner, 1888.

[2] United Nations. The world at six billionoff site, table 1, "world population from" year 0 to stabilization. page 5, 1999.

[3] Jonathan A Foley, Navin Ramankutty, Kate A Brauman, Emily S Cassidy, James S Gerber, Matt Johnston, Nathaniel D Mueller, Christine OConnell, Deepak K Ray, Paul C West, et al. Solutions for a cultivated planet. *Nature*, 478(7369):337–342, 2011.

[4] Global Harvest Initiative. Global harvest initiative: The 2012 global agricultural productivity report. 2012.

[5] Global Harvest Initiative. Editors note: Agriculture and information technology. 2012.

[6] Shimon Y Nof. *Springer handbook of automation*. Springer Science & Business Media, 2009.

[7] Qin Zhang. Opportunity of robotics in specialty crop production. *IFAC Proceedings Volumes*, 46(4):38–39, 2013.

[8] JK Schueller. Cigr handbook of agricultural engineering (vol. vi). *CIGR e The International Commission of Agricultural Engineering*, 2006.

[9] Yoshisada Nagasaka, Naonobu Umeda, Yutaka Kanetai, Ken Taniwaki, and Yasuhiro Sasaki. Autonomous guidance for rice transplanting using global positioning and gyroscopes. *Computers and electronics in agriculture*, 43(3):223–234, 2004.

[10] Bruno S Faiçal, Heitor Freitas, Pedro H Gomes, Leandro Y Mano, Gustavo Pessin, André CPLF de Carvalho, Bhaskar Krishnamachari, and Jó Ueyama. An adaptive approach for uav-based pesticide spraying in dynamic environments. *Computers and Electronics in Agriculture*, 138:210–223, 2017.

[11] Yanbo Huang, Wesley C Hoffmann, Yubin Lan, Wenfu Wu, and Bradley K Fritz. Development of a spray system for an unmanned aerial vehicle platform. *Applied Engineering in Agriculture*, 25(6):803–809, 2009.

[12] Hang Zhu, Yubin Lan, Wenfu Wu, W Clint Hoffmann, Yanbo Huang, Xinyu Xue, Jian Liang, and Brad Fritz. Development of a pwm precision spraying controller for unmanned aerial vehicles. *Journal of Bionic Engineering*, 7(3):276–283, 2010.

[13] Yu Ru, Lan Jin, Zhicheng Jia, Rui Bao, and Xiaodong Qian. Design and experiment on electrostatic spraying system for unmanned aerial vehicle. *Transactions of the Chinese Society of Agricultural Engineering*, 31(8):42–47, 2015.

[14] Xinyu Xue, Yubin Lan, Zhu Sun, Chun Chang, and W Clint Hoffmann. Develop an unmanned aerial vehicle based automatic aerial spraying system. *Computers and electronics in agriculture*, 128:58–66, 2016.

[15] Avital Bechar and Clément Vigneault. Agricultural robots for field operations. part 2: Operations and systems. *Biosystems Engineering*, 153:110–128, 2017.

[16] T Griffin, DM Lambert, and J Lowenberg-DeBoer. Economics of gps enabled navigation technologies. In *Proceedings of the 9th International Conference on Precision Agriculture. July*, pages 20–23, 2008.

[17] David Schimmelpfennig and Robert Ebel. On the doorstep of the information age: Recent adoption of precision agriculture. 2011.

[18] Thomas Bell. Automatic tractor guidance using carrier-phase differential gps. *Computers and electronics in agriculture*, 25(1):53–66, 2000.

[19] O Yekutieli and F Garbati Pegna. Automatic guidance of a tractor in a vineyard. In *Automation Technology for Off-Road Equipment Proceedings of the 2002 Conference*, page 252. American Society of Agricultural and Biological Engineers, 2002.

[20] DC Slaughter, DK Giles, and D Downey. Autonomous robotic weed control systems: A review. *Computers and electronics in agriculture*, 61(1):63–78, 2008.

[21] Timo Oksanen and Arto Visala. Coverage path planning algorithms for agricultural field machines. *Journal of Field Robotics*, 26(8):651–668, 2009.

[22] Howie Choset, Ercan Acar, Alfred A Rizzi, and Jonathan Luntz. Exact cellular decompositions in terms of critical points of morse functions. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 3, pages 2270–2277. IEEE, 2000.

[23] John Milnor. *Morse Theory.(AM-51)*, volume 51. Princeton university press, 2016.

[24] Ercan U Acar and Howie Choset. Robust sensor-based coverage of unstructured environments. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 1, pages 61–68. IEEE, 2001.

[25] Enric Galceran and Marc Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258–1276, 2013.

[26] Ercan U Acar and Howie Choset. Sensor-based coverage of unknown environments: Incremental construction of morse decompositions. *The International Journal of Robotics Research*, 21(4):345–366, 2002.

[27] Enrique Gonzalez, Oscar Alvarez, Yul Diaz, Carlos Parra, and Cesar Bustacara. Bsa: a complete coverage algorithm. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2040–2044. IEEE, 2005.

[28] Gustav Öst. Search path generation with uav applications using approximate convex decomposition, 2012.

[29] Jerome Barraquand, Bruno Langlois, and J-C Latombe. Numerical potential field techniques for robot path planning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(2):224–241, 1992.

[30] Alexander Zelinsky, Ray A Jarvis, JC Byrne, and Shinichi Yuta. Planning paths of complete coverage of an unstructured environment by a mobile robot. In *Proceedings of international conference on advanced robotics*, volume 13, pages 533–538, 1993.

[31] Mathias Jesper Sørensen. Artificial potential field approach to path tracking for a nonholonomic mobile robot. In *Artificial Potential Field Approach to Path Tracking for a Non-Holonomic Mobile Robot*, 2003.

[32] Simon X Yang and Chaomin Luo. A neural network approach to complete coverage path planning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1):718–724, 2004.

[33] AE F Ryerson and Q Zhang. Vehicle path planning for complete field coverage using genetic algorithms. *Agricultural Engineering International: CIGR Journal*, 2007.

[34] Du Xin, Chen Hua-hua, and Gu Wei-kang. Neural network and genetic algorithm based global path planning in a static environment. *Journal of Zhejiang University-Science A*, 6(6):549–554, 2005.

[35] Marco Dorigo and Luca Maria Gambardella. Ant colonies for the travelling salesman problem. *biosystems*, 43(2):73–81, 1997.

[36] Hamed Shah-Hosseini. The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm. *International Journal of Bio-Inspired Computation*, 1(1-2):71–79, 2009.

[37] Ryan J Meuth and Donald C Wunsch. Divide and conquer evolutionary tsp solution for vehicle path planning. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 676–681. IEEE, 2008.

[38] Leonhard Euler. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, 8:128–140, 1741.

[39] Carl Hierholzer and Chr Wiener. Über die möglichkeit, einen linienzug ohne wiederholung und ohne unterbrechung zu umfahren. *Mathematische Annalen*, 6(1):30–32, 1873.

[40] LR Ford and DR Fulkerson. Flows in networks princeton university press. *Princeton, New Jersey*, 276, 1962.

[41] Martin Grötschel and Ya-xiang Yuan. Euler, mei-ko kwan, königsberg, and a chinese postman. *Optimization Stories*, page 43, 2012.

[42] Horst A Eiselt, Michel Gendreau, and Gilbert Laporte. Arc routing problems, part ii: The rural postman problem. *Operations research*, 43(3):399–414, 1995.

[43] Michael Drexl. On the generalized directed rural postman problem. *Journal of the Operational Research Society*, 65(8):1143–1154, 2014.

[44] Michael Drexl and Hans-Jürgen Sebastian. On some generalized routing problems. Technical report, Deutsche Post Lehrstuhl für Optimierung von Distributionsnetzwerken (NN), 2007.

[45] Joseph SB Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric tsp, k-mst, and related problems. *SIAM Journal on computing*, 28(4):1298–1309, 1999.

[46] Sanjeev Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *Journal of the ACM (JACM)*, 45(5):753–782, 1998.

[47] Esther M Arkin and Refael Hassin. Approximation algorithms for the geometric covering salesman problem. *Discrete Applied Mathematics*, 55(3):197–218, 1994.

[48] Joseph SB Mitchell. A constant-factor approximation algorithm for tsp with pairwise-disjoint connected neighborhoods in the plane. In *Proceedings of the twenty-sixth annual symposium on Computational geometry*, pages 183–191. ACM, 2010.

[49] Gregory Gutin and Daniel Karapetyan. Generalized traveling salesman problem reduction algorithms. *arXiv preprint arXiv:0804.0735*, 2008.

[50] Chin Wei-Pang and Simeon Ntafos. The zookeeper route problem. *Information Sciences*, 63(3):245–259, 1992.

[51] Pratap Tokekar, Elliot Branson, Joshua Vander Hook, and Volkan Isler. Tracking aquatic invaders: Autonomous robots for monitoring invasive fish. *IEEE Robotics & Automation Magazine*, 20(3):33–41, 2013.

[52] Wesam H Al-Sabban, Luis F Gonzalez, and Ryan N Smith. Wind-energy based path planning for unmanned aerial vehicles using markov decision processes. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 784–789. IEEE, 2013.

[53] A Alaimo, V Artale, G Barbaraci, CLR Milazzo, C Orlando, and A Ricciardello. Lqr-pid control applied to hexacopter flight. *Issues*, 1(2):11.

[54] Emre Can Suicmez and Ali Turker Kutay. Attitude and altitude tracking of hexacopter via lqr with integral action. In *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*, pages 150–159. IEEE, 2017.

[55] V Artale, G Barbaraci, C Milazzo, C Orlando, and A Ricciardello. Dynamic analysis of a hexacopter controlled via lqr-pi. In *AIP Conference Proceedings*, volume 1558, pages 1212–1215. AIP, 2013.