MOTION PLANNING WITH DISCRETE GEODESICS

CHARACTERISTICS AND ALGORITHMS

A Dissertation

by

THOMAS JAI LOPEZ

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Eduardo Gildin
Committee Members,	David Staack
	Sam Noynaert
	Michael Pate
Head of Department,	Timothy Jacobs

December 2021

Major Subject: Multidisciplinary Engineering

ABSTRACT

Several civil and military applications such as surveillance and reconnaissance require unmanned vehicles to visit targets while being optimized for travel time, fuel, and other communication and kinematic constraints. Unmanned vehicles could also have other limitations set by the size, the design, and the environment. This thesis considers discrete path planning problems for unmanned vehicles.

The shortest continuous path for a constant speed vehicle from any arbitrary initial location and orientation to any arbitrary final location and orientation in the absence of obstacles takes the form of a Dubin's path. While these paths are continuous and smooth, this thesis explores the case when the paths could be discrete instead. Dubin's path is always one of the six kinds which could be an LRL. RLR, RSR, LSL, RSL, LSR; were L and R, are left and right turn arc segments respectively, and S is a straight-line segment. This thesis is concerned with similar paths except that the L and R, are left and right turn polygonal arc segments instead. Our work will prove that the discrete paths are shorter than their continuous counterparts. Moreover, the discrete paths are a more general form since they could generate the same results as the continuous counterpart when the polygonal arc chord length limits to zero. This is because a circular arc could be imagined as a polygonal arc with infinite edges.

These paths are particularly relevant when we consider environments to travel with limited communication abilities. In GPS denied or GPS limited environments, performing discrete straight segment maneuvers are more reliable than traversing through circular arc segments especially since we do not have GPS feedback. Every vertex of the polygonal path

could be imagined as ‘position’ (or ‘time’ when traveling in a constant speed) where the GPS location could be pinged for instead of relying on GPS feedback for the entire tour.

The thesis characterizes the Discrete Geodesics with both non-inflection and inflection segments and presents explores algorithms. The work explores a more generic and beneficial version of the Dubin’s paths extended to study traveling salesmen problems.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a dissertation committee consisting of Dr. Eduardo Gildin and Dr. Sam Noynaert of the Petroleum Engineering Department and Dr. David Staack and Dr. Michael Pate of the Mechanical Engineering Department. Research guidance was also given by Dr. Swaroop Darbha and Dr. Rathinam Sivakumar of the Mechanical Engineering Department, and Dr. David Casbeer of the Air Force Research Laboratories. All other work conducted for the dissertation was completed by the student independently.

Funding Sources

Graduate study was supported by research assistantship from the Air Force Research Laboratories, teaching assistantship from the Mechanical Engineering Department, fellowship by the Gildin Graduate Fellowship and research fund from the School of Public Health. The contents of the work are solely the responsibility of the author and do not necessarily represent the official views of the funding offices.

TABLE OF CONTENTS

LIST OF FIGURES

1. Introduction

1.1 Motivation

Path planning for unmanned vehicles has gained significant attention with the development of advanced technologies in aerial, ground, marine, submarine vehicles as well as in directional drilling applications. Path planning in a broader content is currently relevant especially for autonomous guidance needed for various civil and military applications. It has become even more important to have better lower bounds in path/route planning and optimization. Moreover, having autonomous guidance in environments with limited location measurements (such as GPS) makes exploring discrete geodesics invaluable.

The need to find shortest paths spatially and in the presence of other constraints is the primary motivation of this paper. The work done by L.E Dubin's in 1964 showed that the geodesics (synonym for shortest path) from an initial location and orientation to final location and orientation will be of one of the following six types in the absence of obstacles. an LRL. RLR, RSR, LSL, RSL, LSR; were L and R, are left and right turn arc segments respectively, and S is a straight-line segment. The location and orientation information together constitute a configuration. So, we could rephrase Dubin's path as the shortest path from an initial configuration to a final configuration in the absence of obstacles.

This thesis is devoted to studying path planning using a discrete version of Dubin's paths where we consider polygonal arcs instead of circular arc segments. The characteristics and algorithms of these Discrete Dubin's Geodesics would presented in Section 3 of this dissertation. It would be seen that paths with discrete polygonal arcs and shorter in length in comparison with the original continuous Dubin's paths for a given initial and final configuration. By characterizing discrete

Dubin's path and developing algorithms for the same, we present better geodesics that could be incorporated into path planning algorithms where Dubin's continuous geodesics are currently used.

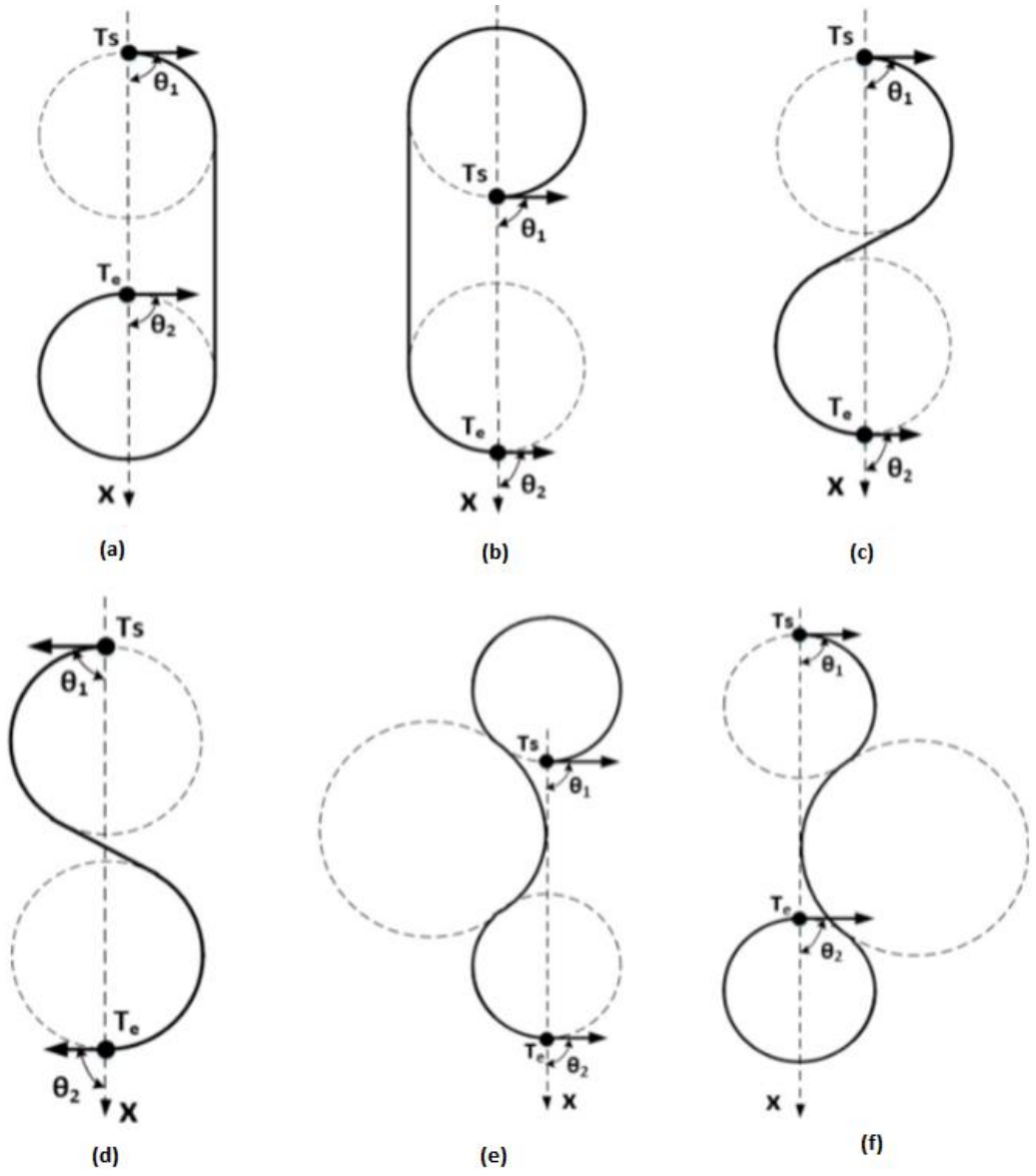

Figure 1.1 Six possible Dubin's continuous paths. With T_s the initial (start) location, T_e the final (exit) location, θ_1 the initial orientation, θ_2 the final orientation, the figure shows the six Dubin's continuous curves (a) RSR (b) LSL (c) RSL (d) LSR (e) LRL and (f) RLR; where R, S, L represents "Right turn section", "Straight section" and "Left turn section" respectively. Figure adapted from L.E Dubins [9]

1.1.1 Studying Discrete Dubin's Travelling Salesman Problems

Since, discrete Dubin's paths segments are shorter in comparison to continuous Dubin's paths for a given initial and final configuration, it is worth exploring how they perform when studying tours with multiple targets called as the Discrete Dubin's Travelling Salesman Problems (DDTSPs). The continuous Dubin's paths makes it possible to find shortest paths given multiple targets (location and orientation given in each of the targets) and a given order of visit for the targets. When the order of visit of the targets is also unknown, there has been work done by [1,2] to study the Dubin's Travelling Salesman problem. Lagrangian based relaxation techniques have been studied [3,4] to compute tighter lower bounds for multiple target tour problems with one unmanned vehicle. This work has also been extended to the case with multiple unmanned vehicles. This thesis would also explore how the Discrete Dubin's paths fair in optimizing tours with multiple targets.

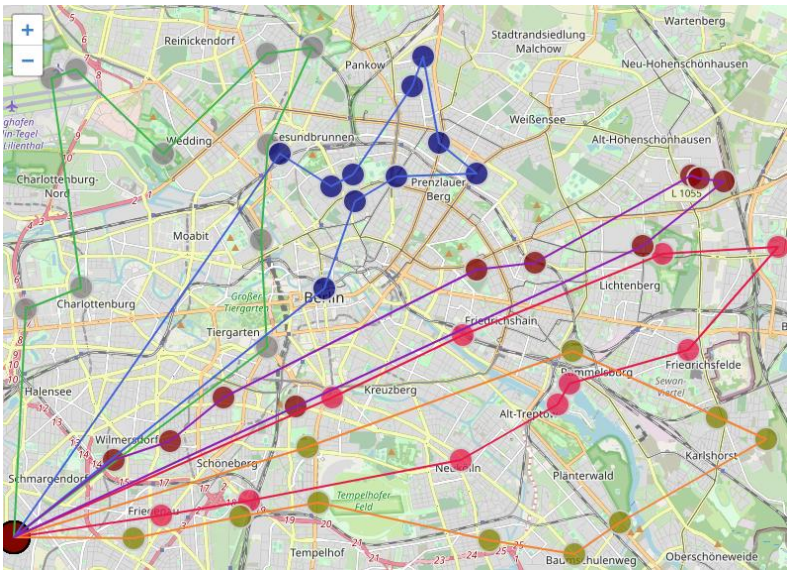

Figure 1.2 TSP paths of multiple vehicles. Each vehicle path represented by different color. The nodes show the target locations that the vehicles need to go to. The vehicles, in this case return the depot after the tour and hence each path ties back to the initial location. Adapted from source: statisticshowto.com

1.1.2 Applications in UAV routing

With the rise of UAVs used by the armed forces of various countries, it becomes important to develop better optimization algorithms. UAV's have a significant advantage over traditional fighter planes because they do not need a pilot and they are portable. They could be deployed in terrains that are hard to reach. Optimizing UAVS, for missions involving reconnaissance and patrolling the border have been studied in [3,4,5]. Civil applications of UAVs include monitoring wildfire, monitoring traffic, search, and rescue missions etc. [6,7,8]. It is also becoming common to use UAVs to collect data from sensors dispersed in the ground at various locations [9,10]. The sensor network nodes need to be periodically visited by the UAVs to transfer data.

Figure 1.3 U.S. Airforce MQ-9A Reaper Airforce UAV in action. The MQ-9 is the first hunter-killer class UAV designed for long endurance and high-altitude surveillance. It has an endurance of about 42 hours at an altitude of 50,000 feet at max. speed of 250 mph with payload of 3000 pounds. Source: Wikipedia

1.1.3 Better lower bounds for battery and fuel limitations

Another challenge is the fact that UAVs have limited fuel capacity. This is a very significant problem in planning missions as it becomes necessary to launch the UAV reasonably close to the mission targets considering how long the UAV needs to be on the tour and how far it needs to travel. Coordinating these tours require planning very efficient routes to make sure the fuel

resources are used in the most efficient manner while also incorporating contingencies in case of emergencies and other uncertainties. The battery capacity for onboard electronics is another limitation on the range of the UAVs like the fuel capacity constraints.

Figure 1.4 Volumetric-Specific energy Vs Mass-Specific energy. For various drone energy sources. The figure shows shaded zones for electrical, electrochemical, and chemical power sources for drones. Li-Po and Li-Ion are the most common sources for drones. Lithium-Thionyl-Chloride (Li-SOCL$_2$) and Lithium-Air have 2 times and 7 times higher energy density compared to Lithium Li-Po. Liquid Hydrogen has a much higher energy density at a factor of 150 times more energy density compared to Li-Po. Source: Drone Industry Insights.

1.1.4 Suited for quadcopter related path planning

Though fixed wing UAVs cannot change their direction heading instantaneously, with the advent of quadcopter like drones, this is no longer the case. It would be more natural to follow discrete polygonal type paths instead of continuous Dubin's paths. It is also easier to maintain accuracy physically when maneuvering UAV's though sections of straight paths instead of paths that include curvatures.

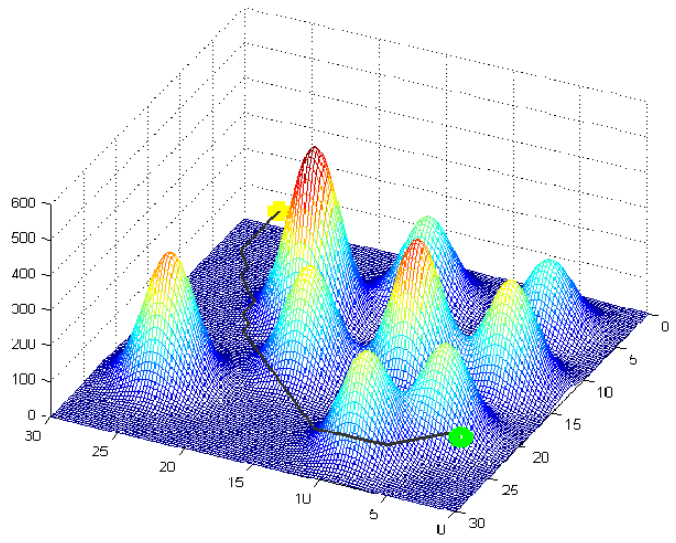

Figure 1.5 Discrete polygonal paths through a mountainous region. The figure shows a three-dimensional discrete Dubin's path for a UAV to efficiently traverse through the mountains while maintaining a low altitude to prevent detection.

1.1.5 UAV routing in GPS denied/limited environments

Another key motivation to explore Discrete Dubin's Paths is the fact that UAVs rely heavily on GPS signals for navigation feedback. The less dependent we could make the UAVs on GPS, the better it could be especially in enemy territory where the GPS signals could be prone to spoofing and jamming. In such hostile environments, access to GPS could be nil or very limited [24,25]. In scenario's where GPS signals are limited, it is common practice to deploy Unattended Ground Sensors (UGSs). If the UAV, or a group of UAVs need to be routed through, the UGSs need to be pre-deployed in the zone. With the range sensors in the UAVs, as well as the strength of the communication link (wireless), the distance between the UAV(s) and the UGS could be estimated. Deployment of the UGSs enroute the UAV's path doesn't come intuitively in the continuous Dubin's paths in comparison to the discrete Dubin's paths. Moreover, since for both the UGS and the UAVs, the batteries that power the wireless signal also have limited capacity, it requires that

both are within the range of each other to communicate. Communication Constrained UAV Routing Problem will be referred to CCURP in this dissertation.

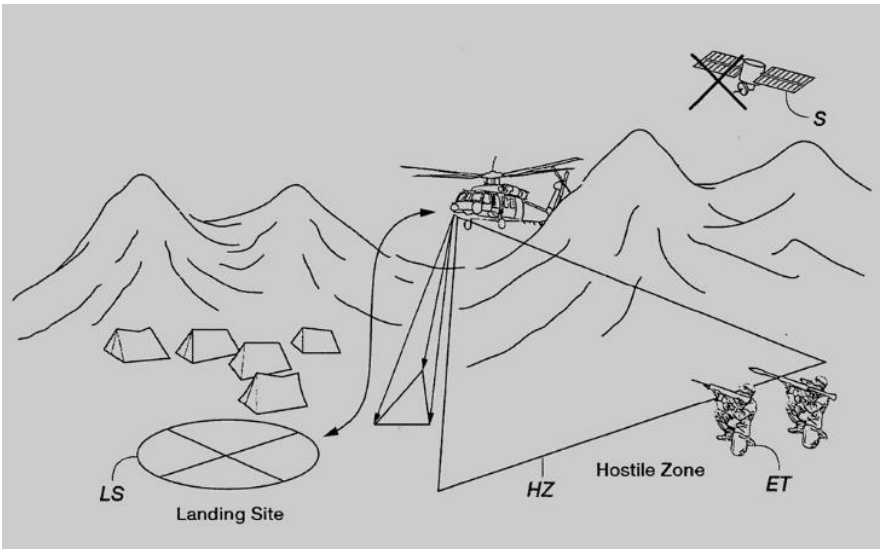

Figure 1.6 Hostile zone, GPS denied zone and Landing zone. Figure shows a typical environment where the satellite coverage is denied and hence GPS signals cannot be used to communicate. The UAV will have to traverse through mountainous regions as also shown in Figure 1.5; however, making it more challenging because of the additional constraints imposed by mapping out hostile zones that the UAV needs to avoid when heading towards the Landing site. Adapted from source: Pioneering Minds

1.1.6 Petroleum Engineering application

In Petroleum Engineering, Rotary Steerable Systems (RSS) are used to steer the drill bit downhole to drill directional wells. Depending on the mechanism used in the RSS, directional drilling systems are broadly classified into point-the-bit and push-the-bit systems. Point the bit systems direct the drill-bit tip, while push the bit systems push the drill collar to achieve the desired change in direction downhole. Even though trajectory plans to follow are continuous curves and smooth, the RSS moves in discrete angle and angle increments. It is thus, more natural for developing trajectory plans in a discrete Dubin's sense rather than a continuous Dubin's path.

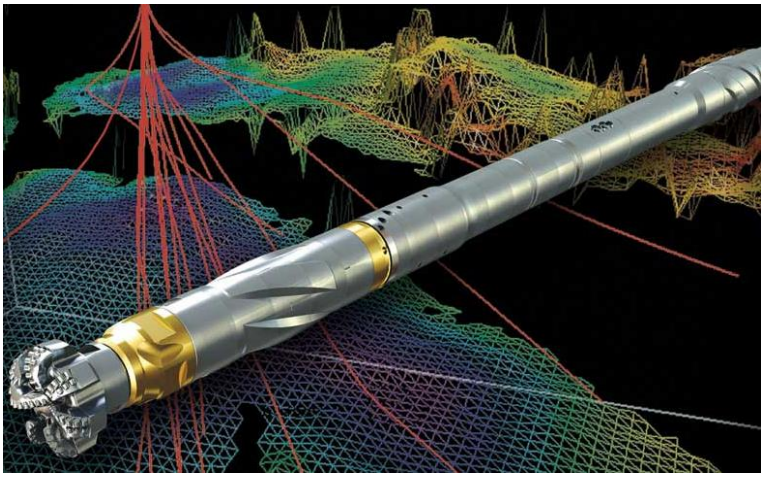

Figure 1.7 Rotary Steerable System – Schlumberger. The figure shows a 3-dimensional surface mesh in the Mesh in the Petrel Software used to perform CFD analysis subsurface. Shown in red are typical well bore path plans that are studied to compare effectiveness in production from a particular region. The figure is inlayed with the Schlumberger's Rotary Steerable System that could be used to drill the directional well once planned. Adapted from source: slb.com

Two key optimizations that is possible with using discrete Dubin's path in trajectory planning for directional drilling are : (1) we could optimize the trajectory between the drilling site to the completion interval, and 2) we could optimize the drilling site location itself. Traditional well trajectory planning methods rely heavily on projected 2D profile of the wellbore trajectory. This is usually coupled with empirical knowledge as well trial and error methods to select a drilling site. With the results of this thesis, petroleum engineering practitioners could use these curves to develop efficient optimization methods. Using gradient descent methods have been proposed in the literature [3] based on 3D Dubins curves for optimal wellbore trajectory planning. However, practically drilling a curved wellbore section is more costly than the straight sections. This makes it more advantageous to plan trajectories based on discrete polygonal arcs or bounded curvature rather than arcs with bounded curvature alone.

Another important application is in subsea or onshore field layout optimization to minimize the development cost. The objective in this sense would be to find the optimal drilling site to drill multiple wells for a drill site so that they could reach all the completion wells, minimizing the total cost of the trajectories while meeting the dogleg severity constraints.

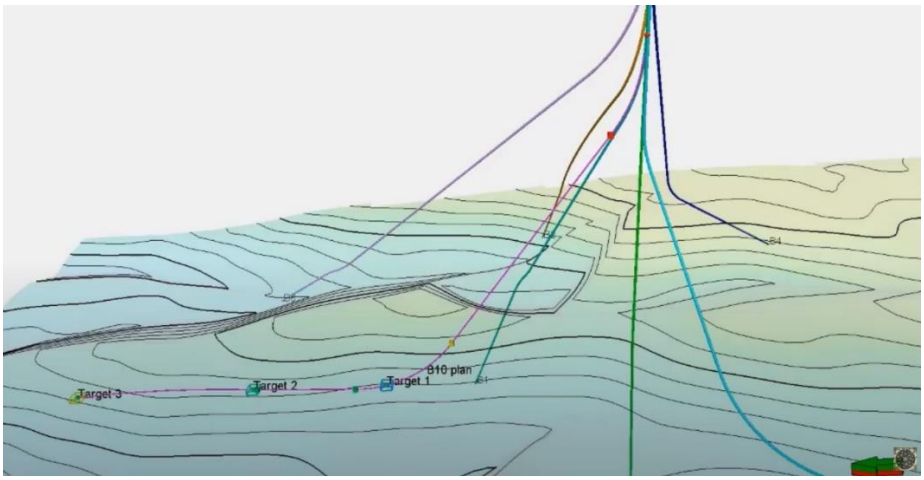

Figure 1.8 Borehole Trajectory Planning – Petrel Software by Schlumberger. The figure shows three targets at different locations that the Rotary Steerable System needs to traverse through. The wellbore trajectories are shown over isosurfaces of similar depths. Adapted from source: slb.com

Various other applications for discrete Dubin's paths in the oil and gas industry includes path planning for sidetracks, inaccessible locations, salt dome drilling, fault controlling, relief well drilling, drainage drilling, drilling multilaterals, offshore platform drilling etc.

1.2 Related work

Curvature-constrained path planning has a long history that predates motion planning for robots. In the work done (1889) by Andrey Andreevich Markov used for the construction of railway segments, he considers four essential mathematical problems critical to the design of railways. His work was published in “Soobscenija Charkovskogo Matematiceskogo Obscestva”. The first problem in which he defined the problem statement could be cited as one of the first works in line of our research. In this problem, his objective was to “find a minimum length curve between two points in the plane provided that the curvature radius of the curve should be less than a given quantity and the tangent to the curve should have a given direction at the initial point”.

In 1951, differential game theory was developed by Rufus Philip Isaacs in his work with Rand Corporation. In this work, he defined and outlined the “homicidal chauffeur” problem. In this work, the car (Chauffeur) pursues a pedestrian to hit and kill. The car travels at a constant preset speed and is bound by the turn radius. The pedestrian tries to avoid being attacked by the car. The pedestrian is not bounded by the turn radius but is bounded by a maximum attainable speed. The pedestrian is not modelled based on fatigue, but both the car and the pedestrian are assumed to have infinity energy to keep on going. The problem also has the direction of the car specified at the initial time.

Lester Eli Dubin’s in 1957 presented a problem in the American Journal of Mathematics. In his work, he considered 2-dimension smooth curves of a bounded curvature. His work characterized that given an initial location and initial direction; as well as final location and final direction, the shortest path between the initial location and final location would be one of the six types of curves. His work lays the foundation of the work we pursue in this research.

Rufus' homicidal chauffeur problem with the pedestrian made immovable is the same problem as the problem considered by Markov. Dubin's problem on the other hand also considers both the incoming and outgoing directions at both the final and initial locations respectively. Even though Markov's remaining three problems that is presented in the published work in 1889 considers fixing the initial and final directions, the problems also consider some additional constraints that is inherent to the railway problem.

Dubin's, Markov's and Rufus's work considers the motion of the car only in the forward direction. James Alexander Reeds and Lawrence Alan Shepp considered the optimization problem given that the car can move both in the forward and the reverse direction. Their work was published in the Pacific Journal of Mathematics in 1990. The car in their problem was bounded by the turning radius, moved with a constant magnitude of linear velocity and could instantaneously change the direction of motion. Study of this class of vehicles was particularly interesting because this is the way carts move in storage rooms.

With this mainly 2 classes were studied thereafter. The 1st class is for vehicles that only move in one direction and the other class for vehicles that move in both directions. The study of motion with bounded curvature has given rise to numerous literatures for optimization problems considering more sophisticated and realistic problems. These include problems with controlled wheels problems, bicycles, vehicle with trailer, vehicle with dual chassis etc. These complicated tasks give rise to complex optimization problems that become extremely difficult to solve. Nonetheless, it is beneficial to study the simplest models since they serve as a base line to study situations and problems that are complex. In other words, they could serve as a bound (upper or lower) when dealing with problems with additional complexity.

Previous work on discrete Dubin's path was only done by Bique et.al [12, 13], In their initial work [12], they put the first and foremost constraint on the bounded-curvature polygonal path by defining a path that turns "too sharply". Though they rightly pointed out that just have a turn restriction at every vertex wouldn't suffice the turn constraint. In other words, as can be seen in Figure 1.9 below, requiring that the at every vertex, the allowed turn is by at most θ wouldn't suffice. This is because having a discrete polygonal arc with more than two short edges at each of whose vertices, the turn θ is made would simulate sharp turn.

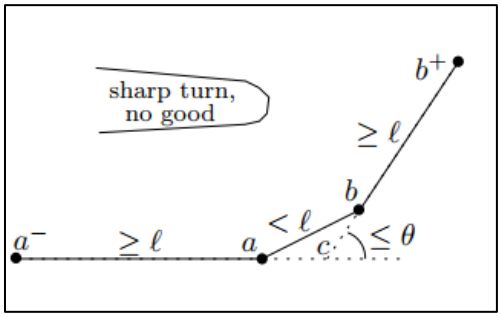

Figure 1.9 Turn constraints as defined by Bique et.al. Let a^-a and bb^+ be edges adjacent to ab. Let the turn constraint be such that the vehicle should not turn by more that θ at a vertex. Let path a^-cb^+ be such that the turn constraint is violated c. For the path $a^-a\ bb^+$ though we do not violate the turn angles at a or at b, we could see that the having two turns that do not violate the turn constraints (turn at a and at b) close to each is equivalent to the path a^-cb^+ that violates the turn constraint (Since both paths begin at a^-a and end at bb^+). Thus, it is important to not just have a restriction of the turn itself but also how close the turns vertices are to each other. Figure adapted from Bique et.al [12]

In their definition, in addition to the turn constraint, they also introduced a length constraint. There were some deficiencies fundamental deficiencies in these definitions which they resolved in the subsequent paper [13]. The second paper dealt with the inflection free case. Some of the definitions for the characteristics where redefined using the new definition for bounded curvature. But not all where addressed.

1.3 Objectives of the research

In this thesis, we aim to take the work started by Bique et.al [12, 13] and to redefine the characteristics of the inflection free discrete Dubin's paths because of the inadequacies mentioned in the way they have defined the bounded-curvature polygonal path in their initial work [11]. Though some fundamental deficiencies are resolved in the subsequent paper, not all where addressed. We aim to characterize both inflection and inflection-free discrete Dubin's paths. The inflection cases have not been studied in the past. We also aim to develop an algorithm for the CCC discrete Dubins paths which have not been attempted before. Once we have developed algorithms that could give us the shortest paths from any arbitrary initial and final configurations, we aim to extend these results to develop algorithms to traverse through multiple targets. We will also explore the Euclidean asymmetric travelling salesman multiple vehicle multiple target allocation problem.

1.4 Thesis Contributions

We aim to redefine the Discrete Dubins path based on 6 fundamental characteristics. Proofs of which are also present in our research. Moreover, we address the inflection case too which includes the CCC paths.

We present a novel algorithm to generate the discrete Dubin's path for a given initial and final configuration. The algorithm is compared to a brute force algorithm that computes all possible solution without taking into consideration the constraint set by out characteristics. We compare the computational time between the two algorithms.

We also show how the discrete Dubin's result gives a better geodesic compared to the Dubin's continuous path especially because traveling through bounded discrete polygonal arcs is shorter than traversing through bounds circular arcs.

Finally, we extend our path planning algorithm to include multiple targets. We provide an approximation constant for the Single-Vehicle Euclidean traveling salesman problem. This constant is beneficial because we can compare how well our algorithm fares with other path planning algorithms and the reader can use this measure to decide if the algorithm would meet their performance needs for optimal trajectory generation. The Single Vehicle case is also extended to multiple vehicles by using Prim's algorithm for calculating the minimum spanning tree. Using this, we eliminate the zero cost edges to find the path the individual needs to travel to get to the targets in the optimal sequence.

1.5 Thesis Organization

The thesis will be organized in the following manner. Chapter 2 will introduce the reader to routing motion using continuous geodesics like Dubins, Reed-Shepp and differential drives, as well as discuss how local planning methods are used in continuous path planning algorithms.

In Chapter 3 we present the characteristics of the discrete Dubin's paths and presents the proofs for the same. The chapter also presents the novel algorithms for computing discrete Dubin's paths as well as compares the algorithm in performance to a brute force algorithm.

In Chapter 4, we extend the algorithm to the single vehicle multiple target case and show how the constant approximation factor was arrived at. This factor is an important performance measure to compare our algorithm to other algorithms. The single vehicle case is then extended to multiple vehicles, multiple target case using Prim's algorithms to divide the targets among the vehicles. In

Chapter 5, we present our contributions and conclusions of our work in discrete Dubin's path characteristics and algorithms in Chapter 3 as well as on the combinatorial motion planning algorithms presented in Chapter 4. We will also discuss our efforts to develop the hardware for a lab-bench scale Rotary Steerable System (RSS) that could be used as a versatile testbed to study various trajectory planning algorithms for directional drilling including discrete Dubin's algorithms. We also present the reader with recommendations for future work in implementing our algorithms and to develop them further for more complex cases.

2. ROUTING MOTION WITH CONTINUOUS GEODESICS

2.1 Introduction

In his work in 1957 on "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents" [9], L.E Dubins came up with characteristics of the kinematic constraints that need to be implemented in vehicles as follows.

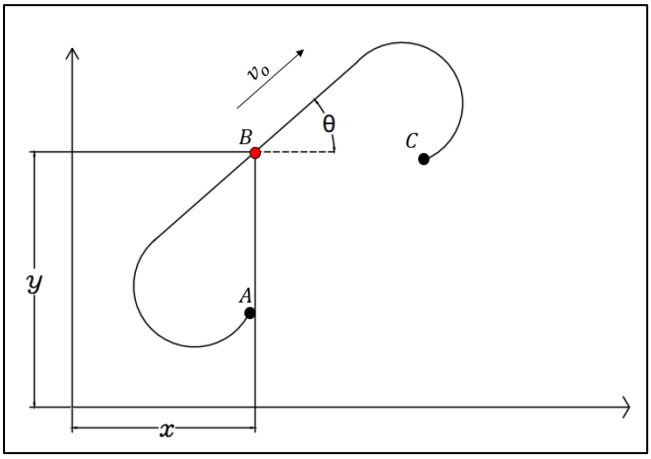

Figure 2.1 Continuous Dubin's vehicle parameters. The figure shows the initial vehicle location at *A* the final vehicle location at *C* and the current vehicle location at *B*. The current orientation is given as θ, location as (x, y) and speed as v_o

$$\frac{dx(v_i,t)}{dt} = v_o \cos\big(\theta(v_i,t)\big)$$

$$\frac{dy(v_i,t)}{dt} = v_o \sin\big(\theta(v_i,t)\big)$$

$$\frac{d\theta(v_i,t)}{dt} = \delta \text{ where } \delta\epsilon[-\omega,+\omega]$$

Here, v_o denotes the speed, ω represents the bound on the yaw rate and $r = \frac{v_o}{\omega}$ is the minimum turning radius of each vehicle. These results by Dubin's formed the foundational motivation for our work on developing a discrete version of these continuous Dubin's paths. Dubins results gives

the optimal path a vehicle must take from a given initial configuration to a given final configuration. The configuration includes information of both the position and the orientation information of the vehicle. In Figure 2.2 and Figure 2.3 (x_1, y_1, θ_1) represents the initial configuration and (x_2, y_2, θ_2) represents the final configuration. Paths in Figure 2.2 and Figure 2.3 have a curved segment of radius r which is the minimum turning radius. If we denote the curved segment of radius r along which the vehicle turns in the anticlockwise (clockwise) rotational motion a L(R). The straight segment the vehicle travels is denoted by S. Dubin's result states that the optimal path joining an initial configuration (x_1, y_1, θ_1) to a final configuration (x_2, y_2, θ_2) subject to the kinematic constraints shown above will always be of the one of the following RSL, LSR, RSR, LSL, LRL or RLR form. Here, RSL, LSR, LSL and RSR are of the CSC path for where C represents the curved segment, whereas LRL and RLR are of the CCC path. If the distance between the initial and final configuration is at least $2r$, the path would be a CCC path, otherwise it would be CSC path.

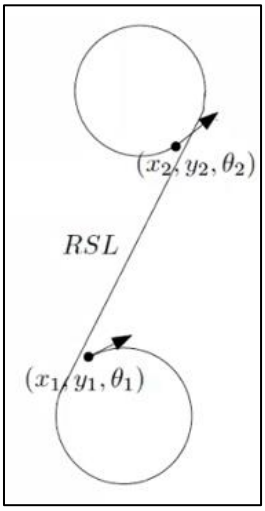

Figure 2.2. Curved-Straight-Curved (CSC) Paths. Path above is a typical CSC path in which the first curved segment is a right-turn segment, and the last curved segment is a left-turn segment. The figure shows the initial configuration (x_1, y_1, θ_1) and the final configuration (x_1, y_1, θ_1)

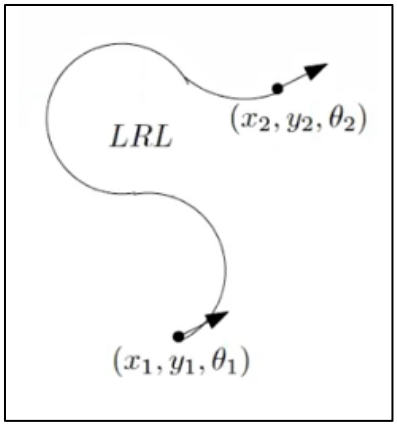

Figure 2.3. Curved-Curved-Curved (CCC) Paths. Path above is a typical CCC path in which the first curved segment is a left-turn segment, the second curved segment is a right-turn-segment and the third curved segment is a left-turn segment. The figure shows the initial configuration (x_1, y_1, θ_1) and the final configuration (x_1, y_1, θ_1)

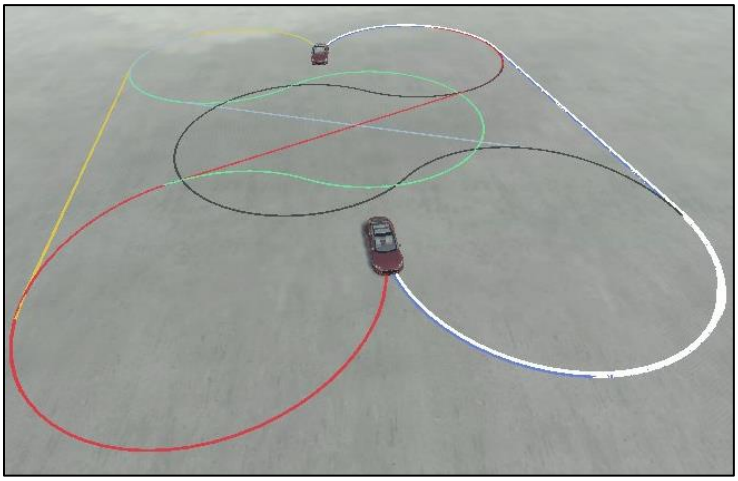

Figure 2.4 CSC and CCC Paths. The picture shows the car closer to the camera at the initial configuration and the car farther at the final configuration. The red (RSL), purple (LSR), yellow (RSR), white (LSL), green (RLR), and black (LRL) paths are all the six possible Dubin's paths

2.2 Dubin's characteristics

The Dubin's car can be imagined as a rigid body that moves in the plane.

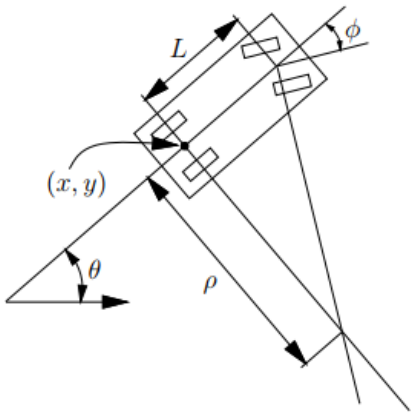

Figure 2.5 Dubin's Car. The figure is similar to Figure 2.1 except that the Dubin's car has coordinates located at the center of the rear axle; and in addition, there are 2 more parameters (compared to Dubin's point vehicle in Figure 2.1); which are: (1) the length between the axles L and (2) the steering angle $\emptyset$. Figure adapted from Lavalle et.al [34]

The Figure 2.5 above shows the parameters associated with the car. A configuration of the Dubin's car can be denoted as $p = (x, y, \theta)$.

The reference frame on the cars body is fixed such that the origin is at the center of the axle for the rear wheels. The x axis is pointing towards the main axis of the car while the y axis coincides with the axis of rotation of the rear wheels. The signed speed of the car is denoted as s. The steering angle is denoted as $\emptyset$. For the wheel orientation as shown in the figure above, it is negative. The length between the rear axle and the front axle is denoted as L. With a fixed steering angle, we can imagine the car to be traveling in a circular motion with a radius of ρ. From figure it could also be noted that ρ can be determined from the intersection of the steering axes with the rear wheel axes. The motion of the car could be represented as a system of equations of the following form

$$\dot{x} = q_1(x, y, \theta, \emptyset, s)$$

$$\dot{y} = q_2(x, y, \theta, \emptyset, s)$$

$$\dot{\theta} = q_3(x, y, \theta, \emptyset, s)$$

For a small time-interval Δt tends to zero, the car must be moving in the direction pointed by the rear wheels. This implies

$$\frac{dy}{dx} = \tan\theta$$

$$\frac{dy}{dx} = \frac{\dot{y}}{\dot{x}}$$

$$\tan\theta = \sin\theta / \cos\theta$$

Therefore,

$$-\dot{x}\sin\theta + \dot{y}\cos\theta = 0$$

The above constraint is met if $\dot{x} = \cos\theta$ and if $\dot{y} = \sin\theta$. Also, any scalar multiple of this is also a solution. The scaling factor directly corresponds to the speed of the car.

The first two components of the configuration transition equations are $\dot{x} = s\cos\theta$ and $\dot{y} = s\sin\theta$.

Now we will derive the equation for $\dot{\theta}$. Let v be the distance travelled by the Dubin's car.

$$dv = p\, d\theta$$

From Trigonometry,

$$\rho = L / \tan\emptyset$$

This implies

$$d\theta = \frac{\tan\emptyset}{L}\, dv$$

Using $\dot{v} = s$ and dividing both the sides by dt, we get

$$\dot{\theta} = \frac{s}{L}\tan \emptyset$$

Now considering action variable w_s, corresponding to speed and the action variable $w_\emptyset$, corresponding to steering angle, a two-dimensional action vector $w = (w_s, w_\emptyset)$, is obtained. Thus, the configuration transition equation for the simple car is

$$\dot{x} = w_s \cos\theta$$

$$\dot{y} = w_s \sin\theta$$

$$\dot{\theta} = \frac{w_s}{L}\tan w_\emptyset$$

The steering angles possible are in the interval $[-\pi/2, \pi/2]$ is also sufficiently large as any other value is equivalent to the value between $-\pi/2$ and $\pi/2$. Note that it was assumed that the car moves in the direction where the rear wheels are pointing to derive the expressions for $\dot{x}$ and $\dot{y}$. If we imagine sitting on a tricycle with the front wheel perpendicular to the back wheels. (That is $w_\emptyset = \pi/2$). The tricycle with this steering angle would rotate in place when we pedal. In other words, $\dot{x} = 0$, and $\dot{y} = 0$, because the center of the back axle does not translate.

This behavior is not possible in a standard automobile because a car that is rear-wheel drive would probably skid across the pavement and a car with a front wheel drive could potentially behave like a tricycle; however, it is impossible because the front wheel would physically collide with the front axle when turned to $\pi/2$. Thus, for a simple Dubin's car, the steering angle should be $< \pi/2$.

Note that the maximum steering angle also implies a minimum turning radius. We might have encountered the problem of having a minimum turn radius especially when attempting an illegal U-turn. It would be difficult to turn the car without driving it off the road.

Another dynamic we need to consider is the speed. If we want to drive the car at the maximum steering angle (or the minimum turning radius), we need to be driving at considerably low speeds

to ensure that we are driving safe and that the car does not tip over. This implies that the maximum steering angle should be less at higher speeds.

For the sake of simplicity, it is assumed that the Dubin's car is moving slowly to neglect the effect of other dynamics. We can place a bound (like $|w_s| \leq 1$) on the speed so that the dynamics won't affect the configurations we could reach. It is important to remember that the Dubin's car moves in the forward direction only. It cannot drive in reverse. If the Dubin's car is a facing a wall, it will be forced to hit the wall and thus this clearly affects the set of configurations reachable.

2.3 Other continuous geodesics

The differential drive

Many industrial and domestic robots do not move like the simple car or the other vehicle models we have considered above. A typical one is the differential drive robot which has two wheels that can be independently controlled by two motors. Sometimes a third idle wheel is present to prevent the robot from tipping over. A fourth idle wheel opposite to the third idle wheel could also be provided for the same purpose. The kind in which the robot uses only two wheels and utilizes the inclination of the robot for movement is common in Segway robots. Here however, we are interested on only the transition equations for the wheels of the differential drive robot and not in particular about how the robot is actuated.

Let L be the distance (see Figure 2.6 below) between the two wheels and r be the radius of the wheels. Let (u_l, u_r) be the action vector that actuates the angular wheel velocities. Depending on direction of rotation and the speed of rotations, the robot can move in different ways (see Figure 2.7 and Figure 2.8 below). If both $u_l = u_r > 0$, then the robot can move in a straight direction. To

switch from forward direction to reverse direction, $u_l = u_r$, but the sign needs to be opposite of what it was earlier. It is important to not that the speed is also proportional to the radius of the wheel. There are some robots, which can actively change the radius of the wheels to accomplish this speed change effect.

If $-u_l = u_r \neq 0$, then the robot rotates in one direction (say clockwise). If the wheels are made to rotate in the opposite direction compared to before then the robot rotates in the opposite direction (say counterclockwise). Because of these dynamics, it is good to have the body frame of the robot placed at the center point of the axle between both the wheels. This is particularly useful as there would be no translation when the clockwise or counterclockwise rotation of the robot takes place. The following are the transition equation of the robot

$$\dot{x} = \left(\frac{r}{2}\right)(u_r + u_l)\cos\theta$$

$$\dot{y} = \left(\frac{r}{2}\right)(u_r + u_l)\sin\theta$$

$$\dot{\theta} = \left(\frac{r}{L}\right)(u_r - u_l)$$

Note that the $\cos\theta$ and $\sin\theta$ terms are present here is like the simple car since the differential drive robot also moves in the direction the wheel point.

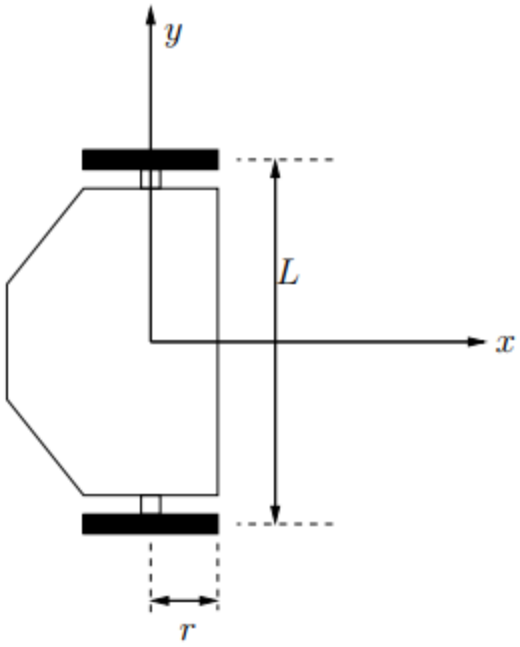

Figure 2.6. Differential Drive Dimensions. The length *L* is the distance between the center of the two wheels and *r* is the radius of the wheels. The *x* and *y* coordinates of the vehicle are measured at the center of the axle.

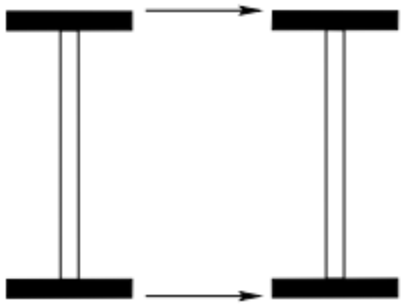

Figure 2.7. Differential Drive Lateral Motion. When both wheels spin in the same direction the vehicle drives forward or backward depending on the direction of spin.

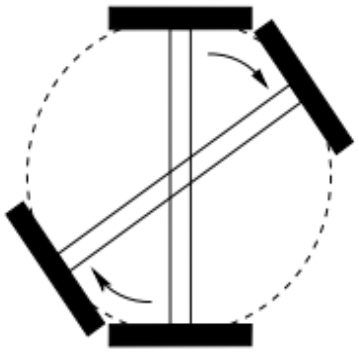

Figure 2.8. Differential Drive Rotation Motion. When both wheels spin in the opposite direction with respect to each other, the vehicle rotates towards the left or towards the right depending on the direction of spin.

We get the translational speed of the robot by averaging the angular velocities of both the wheels. If one when was rotating, and the other wheel was fixed, the translational speed is ½ the speed if both the wheel where rotating. The rotational speed of the robot on the other hand is proportional to the difference in the angular velocities of both the wheels. Thus, we have the following two equations for the robots translational and rotational velocities. They are:

$$u_{robot\_trans} = \frac{(u_l + u_r)}{2}$$

$$u_{robot\_rot} = u_l - u_r$$

Here, u_{robot_trans} and u_{robot_rot} are the action variables which makes the robot translate and rotate respectively. Using these action variables, the transition equation for the robot configuration can be written as

$$\dot{x} = r.u_{robot\_trans} \cos\theta$$

$$\dot{y} = r.u_{robot\_trans} \sin\theta$$

$$\dot{\theta} = \frac{r}{L}.u_{robot\_rot}$$

Recall that these equations are like the transition equations of a simple car. The $\dot{\theta}$ expression of a simple car has w_s appearing, in other words, the rotation rate of the simple car depends on the translational velocity of the car which is true.

The advantage the differential drive has over the simple car is that for an initial configuration and final configuration in $\mathbb{R}^2$, the distance traveled by the frame (which is in the center of the rear axle) is always the Euclidean distance between the initial and final configuration. This is to say that the differential drive can make the body frame follow any continuous path in a plane.

Airplane

The simple car and the derived Dubin's car model developed earlier could be used as a basic to extend to the three-dimensional world and hence to understand the development of the transition equations for an airplane, A simple aircraft model is more than necessary for air traffic control and analysis, but a more detailed model might be necessary when studying the flight stability of an aircraft.

To begin with, assume the aircraft maintain a steady altitude and is only capable of performing yawing rotations. In this case, we can assume the Dubin's car equivalent to work in this scenario, but this is not possible because while $s = 1$ (Some positive speed is possible); $s = 0$ (complete stop) to instantaneously stop in midair is impossible. A more precise model thus becomes necessary. We might also need to consider roll rotations, dynamic effects, disturbances, and pitch effects when studying possible aircraft maneuvers in details. Here, for the sake of simplicity, we are neglecting these effects.

We are only considering a Dubin's like car that can change its altitude also. Nonetheless, the fact that we are dealing with a three-dimensional space puts us in $C = \mathbb{R}^3 \times \mathbb{S}^1$. A configuration here

Is represented as $r = (x, y, z, \theta)$. Let u_z be the action that causes the change in altitude (i.e., $u_z = \dot{z}$). Now the transition equations for the aircraft configuration are

$$\dot{x} = \cos\theta$$

$$\dot{y} = \sin\theta$$

$$\dot{z} = u_z$$

$$\dot{\theta} = u_w$$

With a fixed value for $\dot{z}$ and $\dot{\theta}$, we could get a helical path. We could have constraints over the maximum possible yaw rate and altitude by restricting u_w and u_z, respectively.

2.4 Local Planning Method approach

Suppose the task at hand is to find a path for a Dubin's vehicle as shown in Figure 2.9

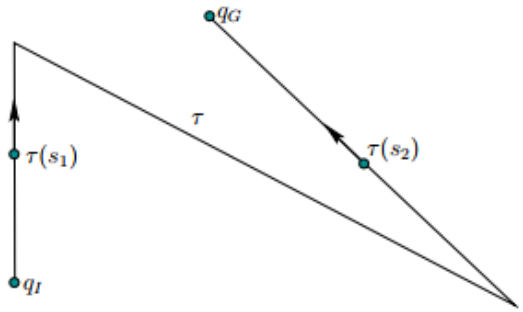

Figure 2.9 Local Planning Method path problem. The figure shows the initial location of the

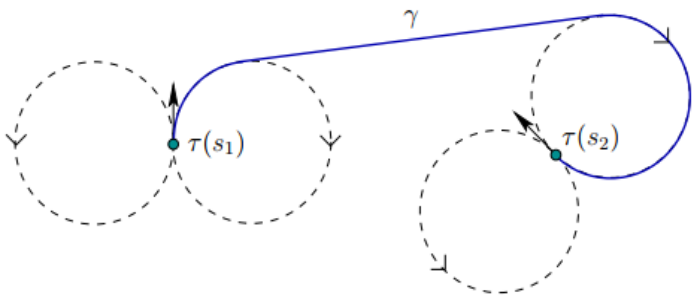

Figure 2.10 Dubins Path without LPM

The path τ shown in the Figure 2.8 is computed by a trajectory planning algorithm that ignores the turn constraints for the vehicle. As can be seen, the vehicle makes to sharp turns. These two sharp turns cannot be traversed by the Dubin's car. The parametric function $\tau(s_1)$ and $\tau(s_2)$ are chosen at random for the first case and appears at the locations as can be seen in the Figure 2.9.

The path τ between $\tau(s_1)$ and $\tau(s_2)$ needs to be replaced by another path that can be executed by the Dubin's vehicle. The orientation of $\tau(s_1)$ and $\tau(s_2)$ are important in this problem because they are part of the initial and final configurations.

The replacement path as shown in the Figure 2.10 is found by using the following local planning method. For the Dubin's vehicle, using the initial configuration, we can draw the left-hand and right-hand circles tangent to the initial heading. Similarly, we can draw the left-hand and right-hand circles tangent to the final heading as shown in the figure. The left and right circles correspond to the maximum possible sharpest turn be it in the left or right directions respectively. It is easy to find a line that in tangent to one circle from the initial configuration and connect it tangent to one circle from the final configuration. It needs to be ensured that the line also matches the direction of the flow of the car depending on the heading direction. In other words, that means to consider the direction of movement as a one-way street.

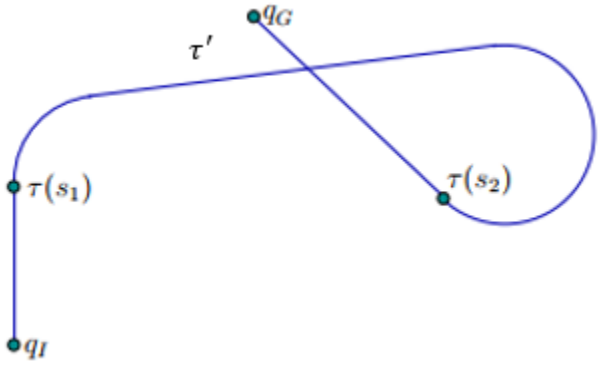

Figure 2.11 Dubin's Path after LPM

Using γ in Figure 2.10, the path τ in Figure 2.9 is updated to obtain τ' in Figure 2.11. The path τ' meets the differential turn constraints for the Dubin's car. The problem was a simple one though in practice, dozens of iterations might be needed to replace path segments. This problem was easy also because there were no obstacles in the path. If we imagine a τ, wherein the path is a narrow road with many zigzag turns, it could be possible that a feasible solution might not even exist. Meaning that it would be impossible to have a path with the Dubin's car could travel. In the case we had considered, if there was an obstacle at the loop in Figure 2.11, then the replacement path would not be possible. In general, with this approach, we can have no guarantee that the replacement portions would be collision-free. This is the principle of the local planning method approach where in we identify those sections locally where the turn constraints are not met and plan trajectories between the initial and final sections such that the Dubin's turn constraints are met and we have the shortest path. In the Local planning method approach, it is important to construct the new path segments as close to the original path as possible.

2.5 Remarks and Discussions

The Dubin's vehicle, the Reeds-Shepp vehicle and the differential-drive vehicle shortest path covered in this section can be build combining some motion primitives. Because of symmetries, it is enough to consider the optimal paths from an initial configuration $p_I = (0,0,0)$ to a goal configuration $p_G \in C$. Where C is the configuration space (All possible configurations between the initial and final configuration). If the optimal that is desired is different from p_I, then the optimal path can be recovered by using rigid body transformations applied to p_I and p_G. In other words, the whole path can be easily rotated and translated without affecting its optimality.

Alternatively, instead of fixing p_I, one can fix p_G and we can consider all the optimal paths from all the possible p_I.

Once the p_I is fixed, the configuration space can be partitioned in various cells that sets of placements. Inside each of the cell, the optimal curve is described by parameterized motion primitives. As an example, the cell for Dubin's vehicle would indicate "left turn", "straight", "right turn". These curves are ideally suited for using as a local planning method in a sampling-based planning algorithm. The idea is that each distinct situation (in our case, left turn, straight, right turn) is called a state and is denoted as x. The set of all states is called a state space denoted by X. In discrete planning, it is important that this set would be countable. In most practical cases, this set would be finite.

It is also important that for every discrete planning application, the state space would only have the relevant information and the irrelevant information would not be encoded in the state. For example: a Discrete planning problem for moving a robot in India should not have the states represent information about whether lights are on in France). The problem with including irrelevant information in the states is that it makes the problem from being easily amenable to one that is completely intractable. However, it is important that the state space X is large to include all the relevant information to solve the task. The planner chooses what action (in the context of Dubin's car the action can be turning the steering to the desired steering angle) need to perform to transform the system. Each action, denoted as u, when it is applied to the current state x, would produce a new state x_{new} . This is specified by the transition function of the state as denoted by f. Thus, the state transition equation would look like:

$$x_{new} = f(x, u)$$

The set of all actions that could be applied for x is denoted as U (in the context of the Dubin's car, this set represents all the possible steering angles the car can take). It is important to note that for distinct states x, x_{new} which are elements of X, the action spaces $U(x)$ and $U(x_{new})$ needn't be disjoint. In other words, multiple states can have the same action space.

Thus, it is now convenient to define U which is the set of all the possible actions over all the states.

$$U = \bigcup_{x \in X} U(x)$$

To summarize, we have the nonempty state space set X which is a finite set (or a countably infinite set). There is an action space $U(x)$ for each state x. The transition function of the state f produces a new state x_{new} by the action u on x. There is a goal set for the states denoted by X_G.

Now, regarding the steering angle and the speed, several interesting variations are:

Simple Car: $U = [-1,\ 1] \times [-\emptyset_{max}, \emptyset_{max}]$. Here we require that $|u_{\emptyset}| \leq \emptyset_{max} < \frac{\pi}{2}$

Tricycle: $U = [-1,\ 1] \times [-\frac{\pi}{2}, \frac{\pi}{2}]$. Here we assume that the tricycle can rotate in one place. This is however unrealistic for a simple car.

Reeds-Shepp Car: This car restricts the Simple Car to have only the $u_s \epsilon \{-1,0,1\}$ speeds. Thus, the car can be assumed to have only three discrete gears. The forward, the parked and the reverse gear.

Dubin's Car: This car restricts the Reed-Shepp car to have only the $u_s \epsilon \{0,1\}$ speeds. Thus, the car can be assumed to have only two discrete gears. The forward, the parked gear.

3. TRAJECTORY PLANNING WITH DISCRETE GEODESICS

3.1 Introduction

A configuration in a continuous Dubin's path is defined as a pair containing both the location and the heading information. Depending on if we are considering path planning for motion in a plane or in space, the location information required would be two dimensional or three dimensional. Apart from this, assuming a constant speed vehicle, the bounded radius needs to be specified. In comparison, a discrete Dubin's path require additional parameters to specify a configuration. The configuration is still defined as a pair (location, heading). The bounded radius and the number of discretization, i.e., number of sides of the polygon need to be specified. In the discrete setting, we also require that where the robot starts, there is a pre-edge. This is to require that, where the robots start, the robot does not start too sharply. This requirement can be relaxed determining the length of the pre-edge. The length of the pre-edge determines the orientation of the left-turn and right-turn polygons inscribed within the left-turn and right-turn circles respectively. Like we have a pre-edge for the initial configuration, we need to define a post-edge for the final configuration as shown in Figure 3.1

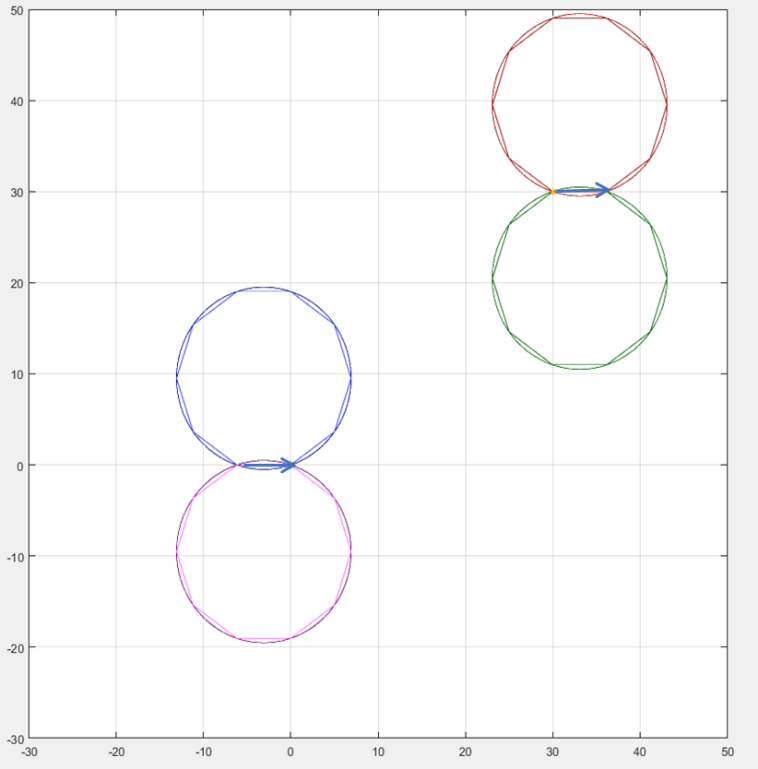

Figure 3.1. Example - Initial and Final Configurations for Discrete Dubins Paths. The initial location is at $(0, 0)$ and the initial orientation is 0 degrees. The final location is at $(30, 30)$ and the final orientation is 0 degrees. Once we know the initial and final configurations, the number of discretization (sides of the polygon) and the minumum turn radius (radius of the circle that incribes the polygon), we have the information to fully define the blue, purple polygons (they both represent the maximum turn limits at the initial configuration); as well as the red and green polygons (they both represent the maximum turn limits at the final configuration).

Similar to the continuous Dubin's results, if the distance between the initial configuration and the final configuration is greater than or equal to $2R\cos\frac{\emptyset}{2}$ where R is the bounded radius for the constant speed vehicle, then the shortest path would not be of the CCC form, the same result is true for the discrete Dubin's case and the proof is similar to Dubin's seminal work in 1974 [1]. Figure 3.2 shows a case where the initial and final configuration greater than or equal to $2R\cos\frac{\emptyset}{2}$ and shows all the Feasible CSC paths. A CSC path would not have a short edge (This proof is shown in Section 3.2). Thus, a simple brute force algorithm shows all the possible paths connecting

the vertices without violating the maximum turn angle. If the maximum turn angle is violated in this case, we would see the paths run through the inside of any of the initial or final configuration polygons.

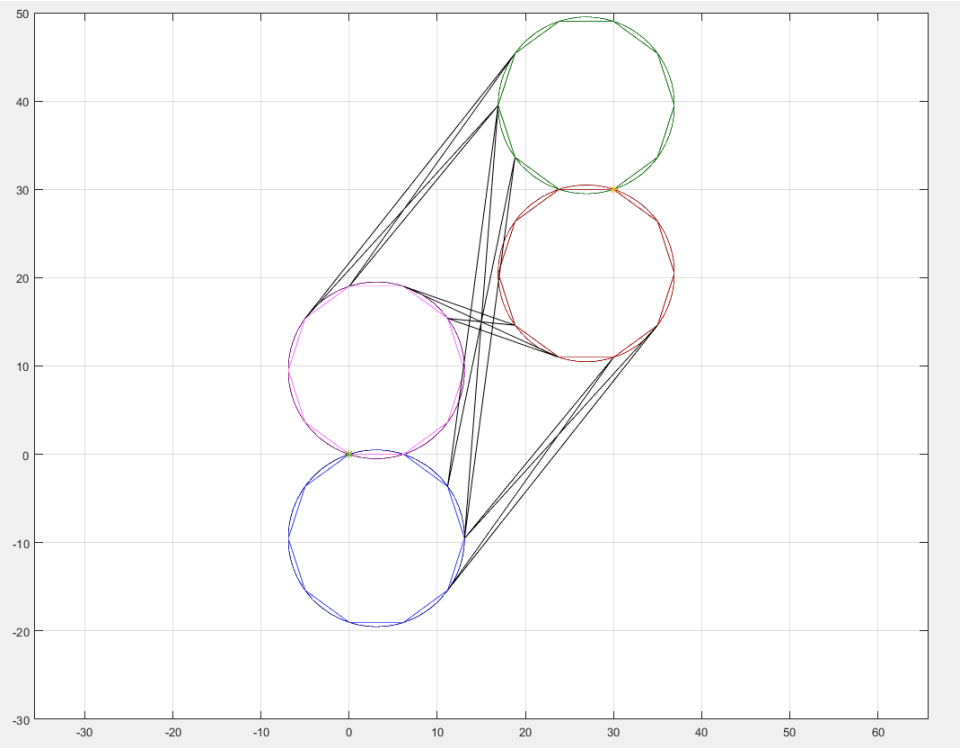

Figure 3.2 Example - All Feasible Discrete Dubins CSC Paths. Running the Brute force algorithm give us all the feasible paths. It could be seen that none of the paths are inside the initial configuration polygons as well as the final configuration polygons.

For a CCC path, Figure 3.3 shows the parameters of relevance that need to be considered in our optimization algorithm. The pre-edge for the initial configuration is shown in red and the post edge for the final configuration is shown in blue. The path considered in the case is a LRL path. Given the initial and final configuration, it is straight forward to draw the circle on the left and the circle on the right. Within these circles, the discrete polygonal arc segments from the initial heading to the final heading is inscribed respectively. Proofs shows in Section 3.2 would also show that the top circle which represents the circle for the second C in the CCC path would have the center

located within the distance of 2R the upper-bound and $2R\cos\left(\frac{\emptyset}{2}\right)$ the lower-bound from the center of C1 and C3. The problem at hand would be to find where with these bounds the center of C2 lies as well as what the orientation of the polygon would be within C1, C2 and C3. Section 3.2 also would prove that a CCC path have at most 2 short edges. For a particular location of C2 within the bounds, we would already know the angles $\theta_1, \theta_2, \theta_3, \beta_1, \beta_2, \zeta_1$ and ζ_2 geometrically (see Figure 3.3 below to see definitions).

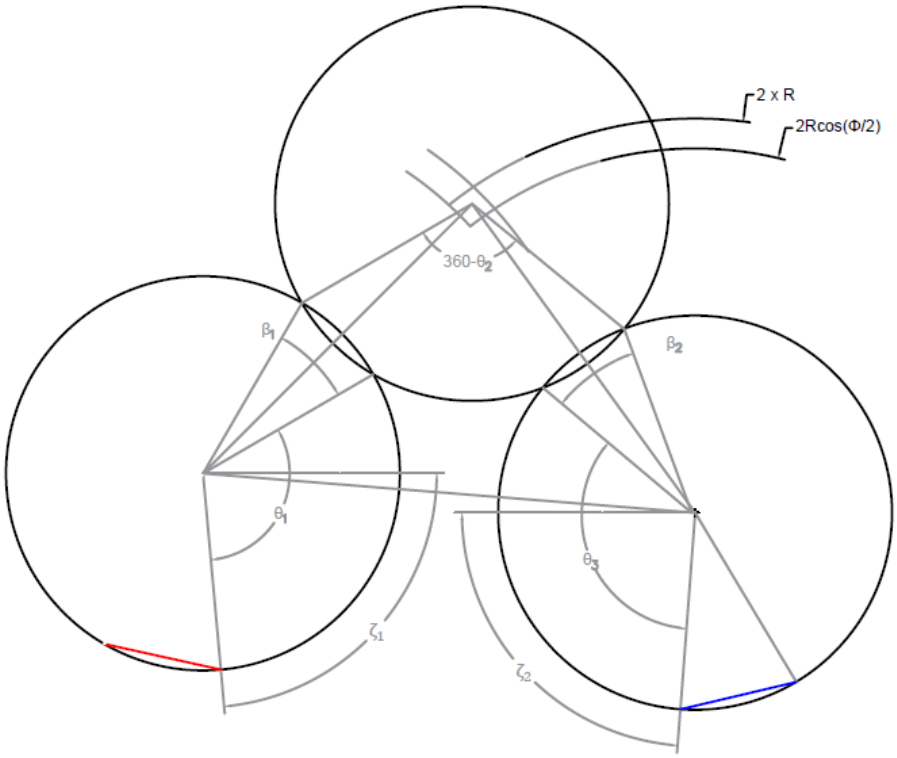

Figure 3.3. CCC Path Parameters. The red edge represents the initial configuration, and the blue edge represents the final configuration. θ_1 is the angle subtended in C1 by the initial location and the first point of intersection between C1 and C2. $360 - \theta_2$ is the angle subtended in C2 by the last point of intersection between C1 and C2 and the first point of intersection between C2 and C3. θ_3 is the angle subtended in C3 by the final location and the last point of intersection between C2 and C3. β_1 is the angle subtended by the two points of intersection between C1 and C2. β_2 is the angle subtended by the two points of intersection between C2 and C3. ζ_1 is the angle subtended by the short edge in C1 and ζ_2 is the angle subtended by the short edge in C2.

3.2 Discrete geodesics characteristics

3.2.1 Characteristic 1

A discrete circular arc of bounded curvature R with 2 edges would have a maximum turn angle α between the pre-edge and post-edge, given by

$$\alpha = sin^{-1}\left(\frac{pre-edge}{2}\right) + sin^{-1}\left(\frac{post-edge}{2}\right)$$

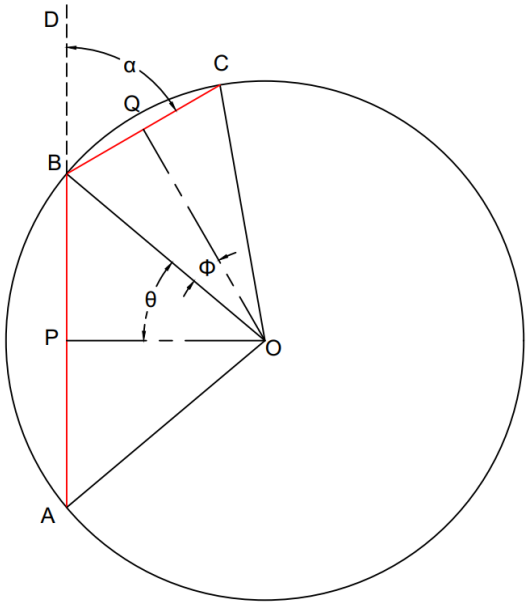

Figure 3.4.Maximum Turn Angle. With the initial location at A and the final location at C, the figure shows the turn maximum turn angle α (∠ DBC). The maximum turn angle is a function of the pre-edge length (AB) and the post-edge length (BC)

The maximum turn angle is derived using simple geometric equation as shown below.

Since ABD is a line $\angle \mathrm{ABD} = 180$

Line ABD is composed of three angles $\angle \mathrm{ABD} = \angle \mathrm{DBC} + \angle \mathrm{CBO} + \angle \mathrm{OBA}$

Wherein $\angle \mathrm{DBC} = \alpha$, $\angle \mathrm{CBO} = (90 - \Phi)$, and $\angle \mathrm{OBA} = (90 - \theta)$

$$\angle \mathrm{ABD} = \alpha + (90 - \Phi) + (90 - \theta)$$

$$180 = \alpha + 90 - \Phi + 90 - \theta$$

$$\alpha = \Phi + \theta$$

$$\alpha = \sin^{-1}\left(\frac{\mathrm{AB}}{2}\right) + \sin^{-1}\left(\frac{\mathrm{BC}}{2}\right)$$

Thus, we get a relationship between the turn angle, the current edge AB and the leading-edge BC

3.2.2 Characteristic 2

In CCC discrete Dubin's paths, an inflection edge between two polygonal paths cannot be shorter than the distance between points of intersection of the two circles that inscribe the polygonal paths

In the CCC path shown in Figure 3.5, the path is annotated with initial location: a, final location: z, pre-edge: $a'a$; and post-edge: zz'. The Polygonal paths $P1$, $P2$ and $P3$ are inscribed within circles $C1$, $C2$ and $C3$ respectively. The path has 2 inflection edges. One that lies between $P1$ and $P2$ and the other that lies between $P2$ and $P3$.

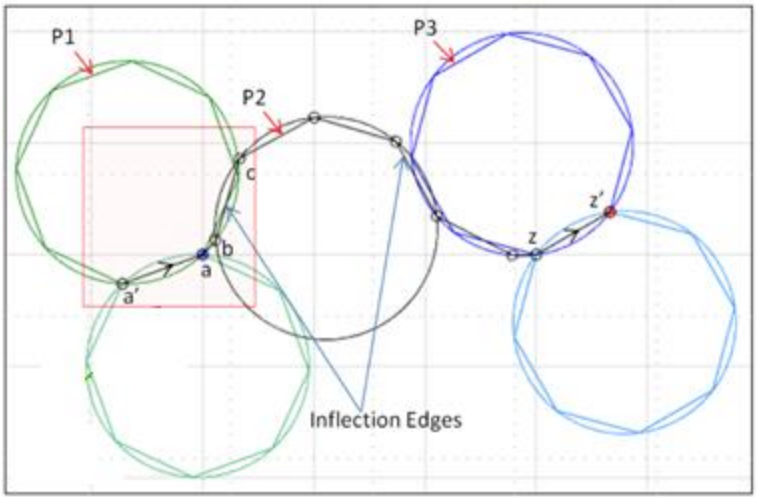

Figure 3.5 Discrete Dubin's Inflection Edges. The figure shows the inflection bc between the pre-edge $a'a$ and the post-edge bc. Since the total path is a LRL path we can expect two inflection edges because of the two change in directions as can be seen.

The polygonal paths with vertices that lie inside the circle that inscribes the polygon violates the turn angle constrain as defined in Characteristic 1. As shown in Figure 3.5 if the inflection edge bc was such that the inflection vertex c lied inside either of the circles, the turn at the inflection vertex b would be too sharp and violate the turn constraint.

3.2.3 Characteristic 3

In CCC discrete Dubin's path, the two points of intersections between the adjacent circles that inscribe the polygonal paths must be $\leq d_\theta$.

This means that for a *C1-C2-C3* discrete-Dubin's path,

(1) the distance between the two points of intersection between *C1* and *C2* must be $\leq d_\theta$

(2) The distance between the two points of intersection between *C2* and *C3* must be $\leq d_\theta$

If the distance between the 2 points of intersection between *C1* and *C2* (or *C2* and *C3*) is $\geq d_\theta$, then by characteristic 1, the inflection edge must be $\geq d_\theta$. In a CCC path, if the inflection edge is greater than d_θ it would violate the turn angle constraint as defined above.

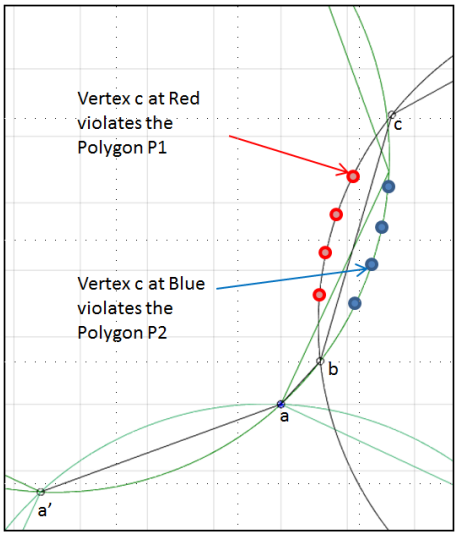

Figure 3.6 Violating Maximum Turn Angle. The red dots show typical location to have path vertices when the turn constraint is violated in the C1 polygon, whereas the blue dots show typical location to have path vertices when the turn constraint is violated in the C2 polygon.

3.2.4 Characteristic 4

A θ-discrete – geodesic circular arc can have at most one short-edge and the rest normal edges.

An edge in a θ-discrete–geodesic circular arc can be either a long edge or a normal edge or a short edge. For a θ-discrete–geodesic, the chord that subtends the angle θ is a normal edge. Any edge longer than a normal edge is a long edge; and any edge shorter than a normal edge is a short edge. It would suffice to prove the following to prove the theorem.

(1) We cannot have a long edge in a θ-discrete–geodesic circular arc

(2) We cannot have more than 1 short edge in a θ-discrete–geodesic circular arc.

To prove that we cannot have a long edge; by definition, a θ-discrete circular arc by Bique et.al [9], "a polygon chain forms a θ-discrete circular arc of unit- bounded curvature if

(i) Its vertices belong in sequence to a common circle of radius one, and

(ii) Edges have a length at most, $d_\theta = 2\sin\frac{\theta}{2}$ (i.e., the edge subtends a circular arc angle almost theta)."

Thus, by definition, we cannot have an edge $>d_\theta$; i.e., we cannot have a long edge.

To prove that θ-discrete–geodesic circular arc can have at most one short-edge and the remaining normal edges

Consider the θ-discrete–circular arc path from point A to B. The angle subtended by AO and BO is φ

φ can be either $< \theta, = \theta,$ or $> \theta$

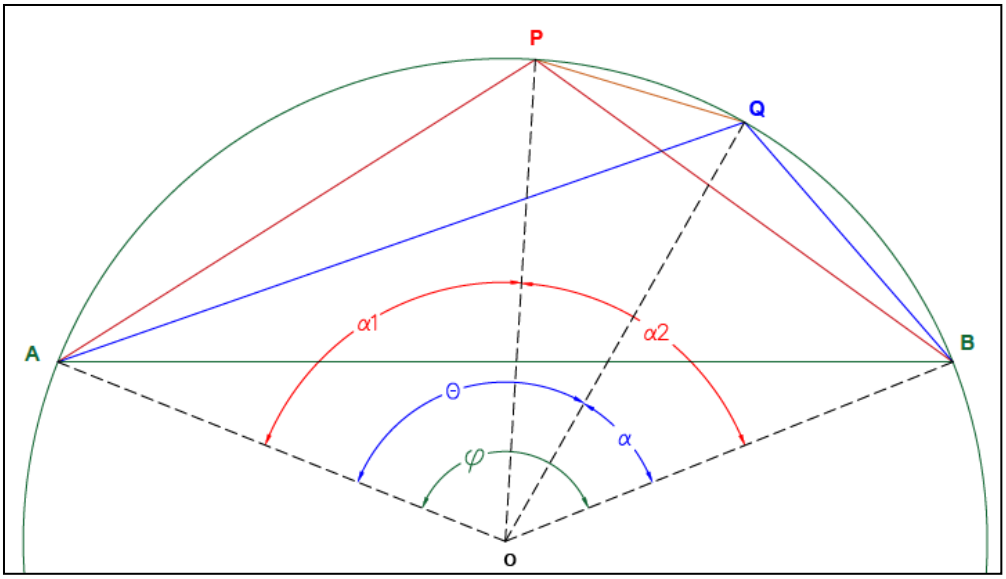

Figure 3.7. Proof of Characteristic 4. With A and the initial location and B and the final location, we could take two alternate paths APB (red-edges) or AQB (blue-edges). The edge AQ is normal edge (see angle subtended by chord AQ is θ, which is the discretization angle for the θ-discrete–geodesic circular arc). Thus, AQB has a normal edge AQ and a short edge QB, while APB has two short edges AP and PB. As can be seen from figure, AP, PB, AQ, QB and AB subtends the angles α_1, α_1, θ, α and φ. By triangular inequality we prove that AQB is shorter than APB and hence it is shorter to take the path with one short edges rather than two short edges and hence proving that a θ-discrete–geodesic circular arc cannot have more than one short edges.

Consider the case of ϕ> θ; For the path (call as path "D1") with just one short edge and remaining normal edges, we have

$$(n \times \theta) + \alpha = \varphi \tag{1}$$

where θ is the angle subtended by the normal edge with the center of the circular arc. α is the angle subtended by the short edge with the center of the circular arc. φ is the angle subtended by the initial and final position (A and B respectively) with the center of the circular arc

For the path (call as path "D2") with two short edges and remaining normal edges, we have

$$\{(n-1) \times \theta)\} + \alpha_1 + \alpha_2 = \varphi \tag{2}$$

Equation (2) – (1) implies

$$\theta + \alpha = \alpha_1 + \alpha_2 \tag{3}$$

Since $\theta \geq \max\{\alpha_1, \alpha_2\}$ and $\alpha \leq min\{\alpha_1, \alpha_2\}$, so we can write

$$\theta - \alpha \geq \max\{\alpha_1, \alpha_2\} - min\{\alpha_1, \alpha_2\}$$

$$|\theta - \alpha| \geq |\alpha_2 - \alpha_1| \tag{4}$$

From equations (1) and (2), comparing the distance traversed in D1 versus D2, is equivalently to comparing the distance 'd1' traversing the short edge (subtending α with O) and the normal edge (subtending θ with O) versus the distance 'd2' traversing the short edge (subtending α_1 with O) and the short edge (subtending α_2 with O)

$$\mathrm{d1} = 2\sin\left(\frac{\varphi}{2}\right) + 2\sin\left(\frac{\propto}{2}\right)$$

$$\mathrm{d2} = 2\left[\sin\left\{\frac{\frac{\varphi}{2} + \frac{\propto}{2}}{2}\right\}\cos\left\{\frac{\frac{\varphi}{2} - \frac{\propto}{2}}{2}\right\}\right]$$

$$\mathrm{d3} = 2\sin\left(\frac{\propto_1}{2}\right) + 2\sin\left(\frac{\propto_2}{2}\right)$$

$$\mathrm{d4} = 2\left[\sin\left\{\frac{\frac{\propto_1}{2} + \frac{\propto_2}{2}}{2}\right\}\cos\left\{\frac{\frac{\propto_1}{2} - \frac{\propto_2}{2}}{2}\right\}\right]$$

From equation (4) since $|\theta - \alpha| \geq |\alpha_2 - \alpha_1|$;

$$\cos\left\{\frac{\frac{\varphi}{2} - \frac{\propto}{2}}{2}\right\} \leq \cos\left\{\frac{\frac{\propto_1}{2} - \frac{\propto_2}{2}}{2}\right\}$$

From equation (3) since $\theta + \alpha = \alpha_2 + \alpha_1$;

$$\sin\left\{\frac{\frac{\varphi}{2} + \frac{\propto}{2}}{2}\right\} = \sin\left\{\frac{\frac{\propto_1}{2} + \frac{\propto_2}{2}}{2}\right\}$$

Therefore $d1 \leq d2$. This means that 2 short edges can be combined as a normal edge and a short edge to have a shorter path.

Consider the case for $\varphi = \theta$. We can substitute $\alpha = 0$ in the above proof to get the result that 2 short edges can be combined as a single normal edge have a shorter path.

Consider the case for $\varphi < \theta$. We need to prove that the distance of the path 'd3' with a single short edge connecting point A to point B is shorter than the distance of path 'd4' with more than 1 short edges connecting point A to point B

In Figure 3.7, consider the path with 2 short edges from point A to point B, by triangular inequality in triangle AA'B we can see that AB is shorter than AA'+ A'B. Hence any 2 short edges can be combined to form a single short edge from A to B. This can be naturally extended to a more general case of having n short edges between A and B can be combined to have a single short edge between A and B.

Combining the results for $\phi > \theta$, $\phi = \theta$, $\phi < \theta$, any path with more than 1 short edge can be combined to have shorter paths with at most 1 short edge and remaining normal edges.

An Example: For the Path $A - A1 - A2 - A3 - A4 - B$ in Figure 3.8, Path $(A - A1 - A2)$ with 2 short edges can be combined to be $(A - A2)$ with a single short edge (result of case for $\phi < \theta$)

Path $(A3 - A4 - B)$ with 2 short edges and be combined to be $(A3 - B)$ with a single short edge (result of case for $\phi < \theta$).The updated path now is Path $(A - A2 - A3 - B)$

Path $(A - A2 - A3)$ with 2 short edges can be combined to be $(A - Q - A3)$ with a single normal edge and a short edge (result of case for $\phi > \theta$). The updated path now is Path $(A - Q - A3 - B)$

The path (Q-A3-B) with 2 short edges can be combined to $(Q - B)$ with a single short edge (result of case for $\phi < \theta$) The updated path now is Path $(A - Q - B)$ which is the shortest and has a single short edge and has remaining 1 normal edge as the theorem required.

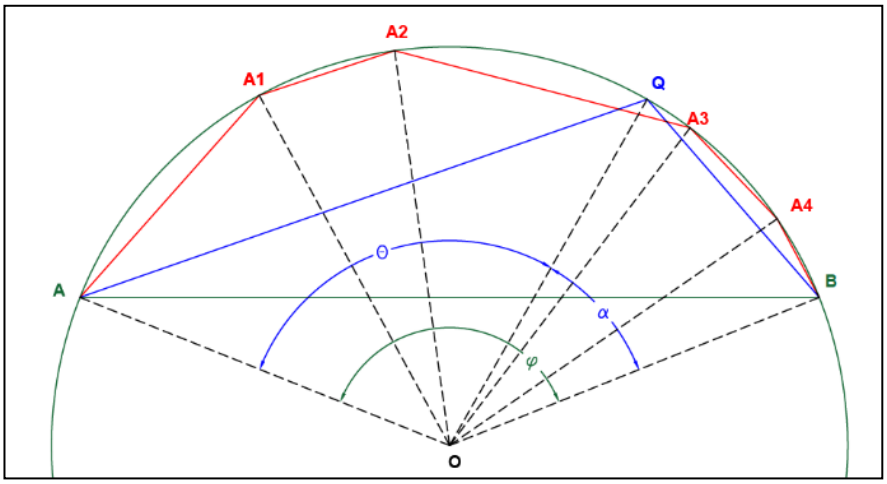

Figure 3.8 Example for Characteristic 4. This is an example that extends the fact that - *it is shorter to have the path with one short edge rather than and equivalent path with two short edges* (shown in Figure 3.7) to help in optimizing the path $A - A1 - A2 - A3 - A4 - B$ which has five short edges to begin with. The procedure used the same triangular inequality principle to combine two short edges into one and thereby finally arriving at a path AQB with just one short edge.

3.2.5 Characteristic 5

In CCC discrete Dubin's path, the distance between the centers of the adjacent circles that inscribe the polygonal paths must be $\leq 2R$

Suppose the shortest path CCC path with C2 at 2R+z distance from C1 is as shown in Figure 3.10 and as shown in Figure 3.10,

Ψ_1 angle of inflection edge 1 in C1

Ψ_2 angle of inflection edge 1 in C2

Ψ_3 angle of inflection edge 2 in C2

Ψ_4 angle of inflection edge 2 in C3.

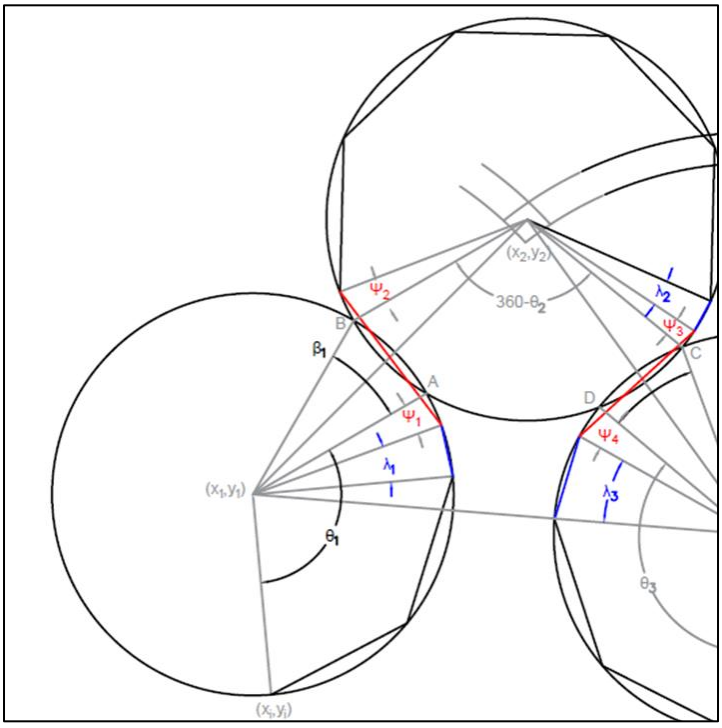

Figure 3.9 Parameters required for Characteristic 5. In addition to the paremeter shown in Figure 3.3, this figure has Ψ_1 angle of inflection edge 1 in C1, Ψ_2 angle of inflection edge 1 in C2, Ψ_3 angle of inflection edge 2 in C2, and Ψ_4 angle of inflection edge 2 in C3.

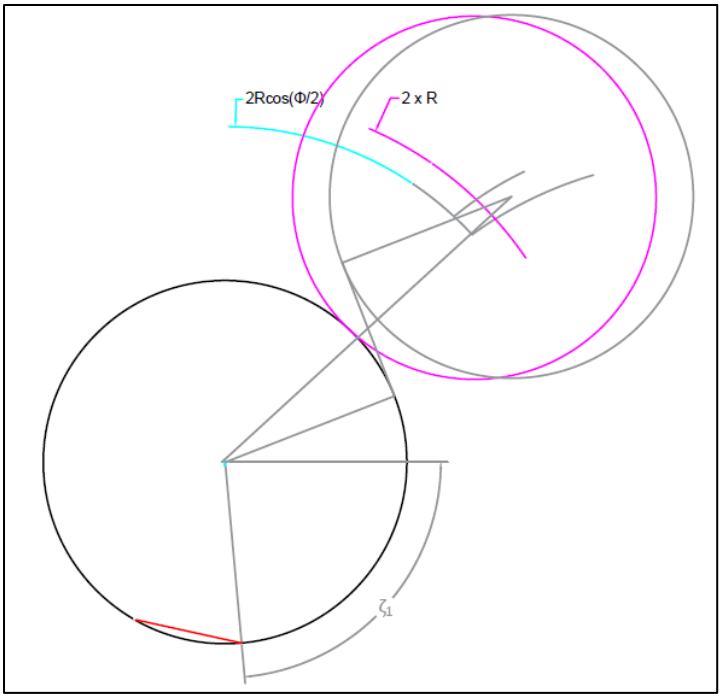

Figure 3.10. Showing bounds of the center of the second arc in a CCC path. As can be seen, if the center of C2 is situated farther to the center of C1 than $2R$, we could join the arcs at C1 and C2 with a tangential segment thus having a CSC path. If the center of C2 is situated closer to the center of C1 than $2R\cos\frac{\emptyset}{2}$, then it would be impossible to have an inflection edge that would not violate the turn angle constraint.

If the distance between the adjacent circles if greater than $2R\cos\frac{\emptyset}{2}$, as seen in figure, we would be able to insert straight segment between the two discrete arcs and there by have a shorter CSC path instead of a CCC path.

3.2.6 Characteristic 6

The path taken for a CSC inflection path with discrete polygonal arcs are always shorter than the path taken in with continuous circular arcs.

For a CSC path to be the shortest path, as seen earlier, the distance between the initial and final configuration must be greater than or equal to $2R\cos\frac{\emptyset}{2}$. In that case, as seen in Figure 3.11, it is possible to construct a line tangent to both the adjacent circles.

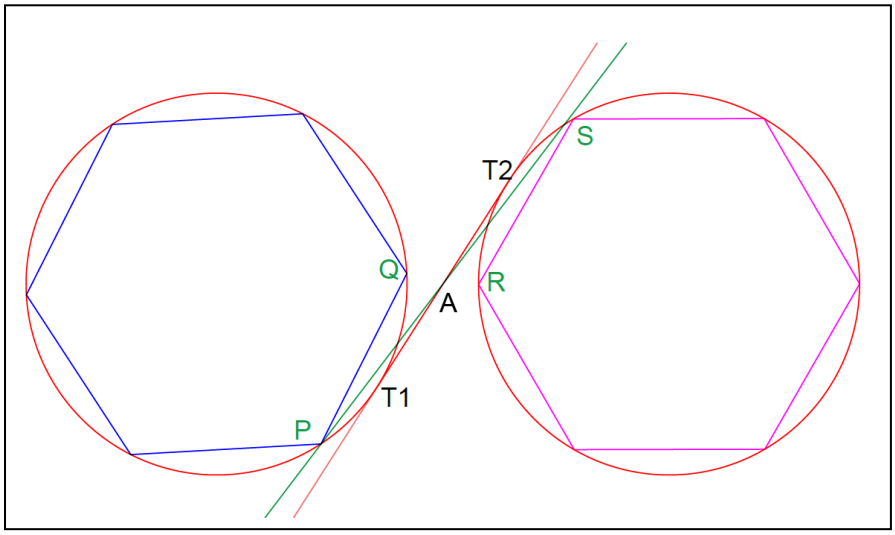

Figure 3.11. Initial Location of Tangent. We construct a tangent between the two circles that inscribe the polygonal arcs as a first step. The tangent is shown by the red line.

T1 and T2 are the points in the left and right circles respectively where the tangent line meets the circles. Point A is located at the midpoint of the line segment T1-T2. We can rotate the tangent line about A in the clockwise direction until the line meets the first vertex. For our case this is point S on the left polygon. With S as the point of rotation, we can rotate the line further in the clockwise direction to meet the first vertex of the other polygon.

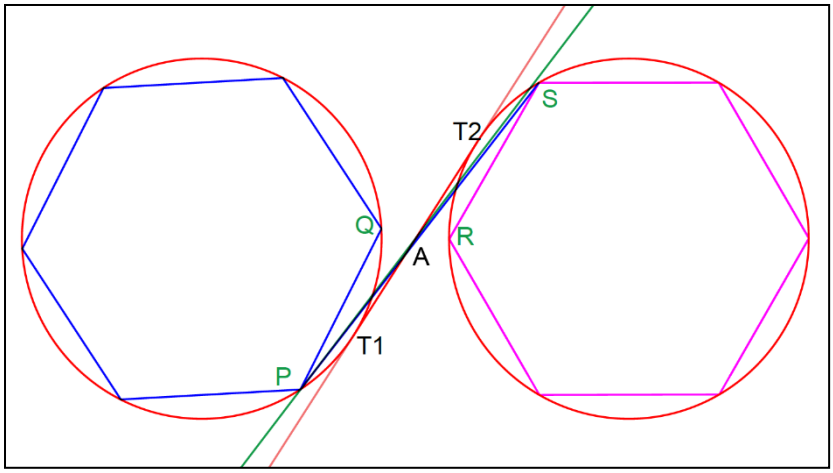

Figure 3.12. Final Location of Tangent. The tangent is made to rotate about its midpoint A clockwise to contact both the polygonal arcs at point P and S.

The green line is the resulting line after performing these operations. The green line segment between P and S in Figure 3.12 shows is the straight-line segment in the CSC discrete Dubin's path. As can be seen in Figure 3.13, in comparing the red path and the green path, clearly the green being a straight-line segment is shorter. This we could confirm that the CSC inflection path with discrete polygonal arcs is always shorter than their continuous counterparts.

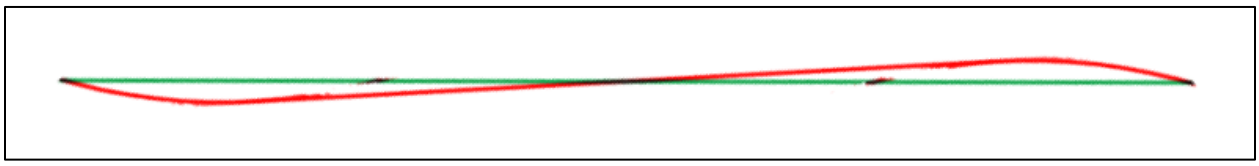

Figure 3.13. Example - Continuous Dubins being longer. After rotating the tangent as shown in Figure 3.11 and 3.12, we can compare the lengths of the results section in the Discrete and Continuous case and conclude that the discrete polygonal arcs are shorter as can be seen above

3.3 Discrete geodesics algorithm

3.3.1 Basic Algorithm

The Basic algorithm attempts to find every possible θ-discrete curvature constrained CSC and CCC paths. The algorithm utilizes a Brute force type algorithm taking into consideration the end point configuration and θ-discrete curvature constraints. The algorithm computes every feasible vertex combination to produce a CSC and CCC paths between the initial and final configuration. The algorithm has six main steps that will be described briefly here. Once the parameters and required initial and final configurations are read, the algorithm checks if the configurations are reasonable far away so as not to compute the CCC paths. The algorithm starts by computing the CSC paths. If the initial and final configurations are less than $4R\cos\frac{\emptyset}{2}$ then we compute the CCC paths. The lengths of all six paths are then compared to find the shortest of them all.

1) Read configuration inputs: Minimum Radius of Curvature, Discretization Steps, Initial Location coordinates and heading, Final location coordinates and heading

2) Compute the pre length and post length vectors

3) Draw configuration circles and configuration circles for the initial and final configurations. The configuration circles represent the turn bounds if we were to travel along a continuous path while the configuration polygons represent the turn bounds if we were to travel along the discrete path.

There are two configuration circles in the initial location. The configuration circle left (alternatively right) of the initial location represents the bounds on maximum turn if it were to take a left continuous turn from the initial location (respectively right turn). Similar configuration circles

are also drawn for the final configuration that shows the bounds of the approach turns towards the final locations.

4) Compute CSC paths

From any vertex of the initial configuration polygons (both left and right), there can possibly be a feasible straight path towards any vertex of the final configuration polygons (both left and right). The path would be feasible if the turn angle constraints for the θ-Discrete Curvature are satisfied at every vertex of the polygonal path. Compute the distances of every feasible path and save the shortest path. Identify the type of CSC path by determining the sign change in turn directions at the vertices

5) Compute CCC paths

The shortest path will take the C-C-C form only if the distance between the initial and final configuration is within $4R\cos\frac{\emptyset}{2}$ (where 'r' is the minimum radius of curvature) from each other. Compute the distance between the initial and final configuration and check if C-C-C paths exists.

- Compute L-R-L

For every vertex of the initial configuration 'Left' polygon and for every vertex of the final configuration 'Left' polygon, check if a 'Right' discrete circular arc of radius 'r' exists such that the turn angle constraints for the θ-Discrete Curvature is satisfied. Compute the distances of every feasible path and save the shortest path

- Compute R-L-R

For every vertex of the initial configuration 'Right polygon and for every vertex of the final configuration Right polygon, check if a 'Left' discrete circular arc of radius 'r' exists such that the

turn angle constraints for the θ-Discrete Curvature is satisfied. Compute the distances of every feasible path and save the shortest path

6) Compare the distances of the shortest CSC and CCC paths to identify the shortest discrete Dubin's path.

3.3.2 Novel Algorithm for Discrete Dubin's Path Generation

The novel algorithm reads the input in a similar manner to the basic algorithm. The algorithm does not use brute force methods to calculate the paths. Instead for CSC paths the algorithm utilizes principles derived in characteristic 6 to find the CSC path. We begin by computing the tangent to the circles that inscribe both the initial and final configurations. The tangent is rotated about its own midpoint to contact the vertices of both the polygonal arcs. The algorithm computes the CSC paths for the LSL, RSL, LSR, and RSR paths. For the CCC paths, the algorithms first checks if the distance between the initial and final configurations are within $4R\cos\frac{\emptyset}{2}$. If not we needn't compute the CCC paths because we can have a shorter CSC path. The algorithm uses the bounds from the characteristics to cut down on the computational effort and utilize principles from the characteristics to arrive at the shortest path.

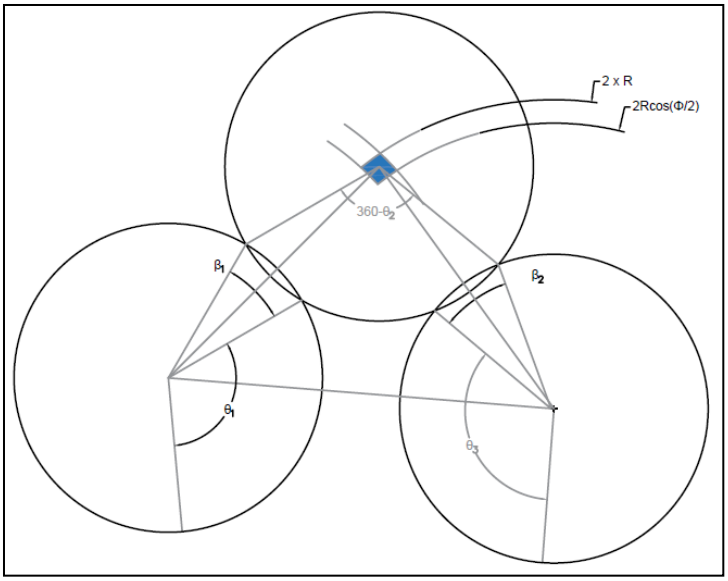

Figure 3.14. Bounds for center of C2 from C1 and C3. The figure shows that for CCC paths, the bounds for the center of C2 has to be in the shaded region as was discussed in Characteristic 2 and 5.

1) Read configuration inputs: Minimum Radius of Curvature, Discretization Steps, Initial Location coordinates and heading, Final location coordinates and heading

2) Compute the pre length and post length vectors

3) Draw configuration circles and configuration circles for the initial and final configurations. The configuration circles represent the turn bounds if we were to travel along a continuous path while the configuration polygons represent the turn bounds if we were to travel along the discrete path.

There are two configuration circles in the initial location. The configuration circle left (alternatively right) of the initial location represents the bounds on maximum turn if it were to take a left continuous turn from the initial location (respectively right turn). Similar configuration circles

are also drawn for the final configuration that shows the bounds of the approach turns towards the final locations.

4) Compute CSC paths

- Compute 4 tangents from the 2 initial configuration circles to the 2 final configuration circles.
- For each tangent, rotate the tangent to meet the corresponding polygonal arcs in the initial and final discrete arcs.
- Compute the total distance traversed by each of the CSC path.
- Identify the type of CSC path by determining the sign change in turn directions at the vertices and store the four path distances.

5) Compute CCC paths

- Refer to Figure 3.14 above. Compute the shaded region by calculating $2R$ and $2R\cos\frac{\emptyset}{2}$.
- Start with the base of the shaded region to start the algorithm. Since the sum of the total short edge angles is minimum at the base of the shaded region.
- Find the number of short edges in C1, C2 and C3.
- Using principles used in Characteristic 4, combine the short edges in C1, C2 and C3 separately to have at most one short edge in C1, C2 and C3 and remaining normal edges.
- Compare the short edge in C1 and C3. Let the angle subtended by the short edges in C1 and C3 be α_1 and α_3 respectively.
- Find the angle required to the nearest normal vertex in C1 and C3. Let the shorter angle be φ.

- Compute $Angle\ 1\ =\ min\ \{(\varphi - \alpha 1), (\alpha 1 - 0)];\ Angle 2 =\ min\ \{(\varphi - \alpha 2), (\alpha 2 - 0)];$

- If Angle 1< Angle2, Eliminate Short Edge 1 in C1 first. If Angle 2< Angle1, Eliminate Short Edge in C3 first.

- For the special case, number of short edges will be zero for a perfect discrete CCC path.

3.4 Remarks and Discussion

As an example, consider the discrete Dubin's path shown in Figure 3.15 has a discrete parameter $n = 8$. The path when overlayed with the continuous Dubins path as shown in Figure 3.16 shows a significant decrease in its length. This is attributed to the fact that the path in the discrete Dubin's sense traverses through discrete polygonal arcs as opposed to bounded circular arcs which are longer in lengths. The proof of this was discussed in Characteristic 6 in the previous section ad could be seen in Figure 3.12 and Figure 3.13. As we increase the number of sides of the polygon from n=8 to n=20, we could see that the polygon now even more closely resembles a circle and therefore the shortest path in comparison would be longer than the case with n=8 discretization. As the number of discretization tends to infinity, we would approach the limiting case of the continuous Dubin's result. Thus, we could see that the discrete Dubin's characterization, algorithm and results are generic in nature than the continuous Dubin's case.

The shortest bounded curvature polygonal paths have the same structure as the continuous Dubin's paths and the properties of continuous Dubin's paths follow from our result as a limiting case when the number of discretization tend to infinity.

We could also see from Figure 3.15 how the discrete Dubin's results are more practical in trajectory determination for UAVs compared to Figure 3.17 because we can create any number

of intermediate waypoints by increasing the number of discretization of the polygon and ask the closed loop control of the UAV to measure sensor reading in the same frequency. Figure 3.16 shows an overlay of the continuous Dubin's path results on the n=8 discrete Dubin's path. Seeing Figure 3.17 we could see that even when traversing through a smooth continuous curve, it is now more practical since we have a trajectory that exactly resembles the true nature of the discrete trajectory control we want in for the control systems in the UAV.

Finally, the distance of the shortest paths was directly proportional to the discretization in the polygonal path. Thus, we could infer that the length of the continuous Dubin's path could be used as an upper bound of the total length in formulating the discrete Dubin's problem as an optimization problem.

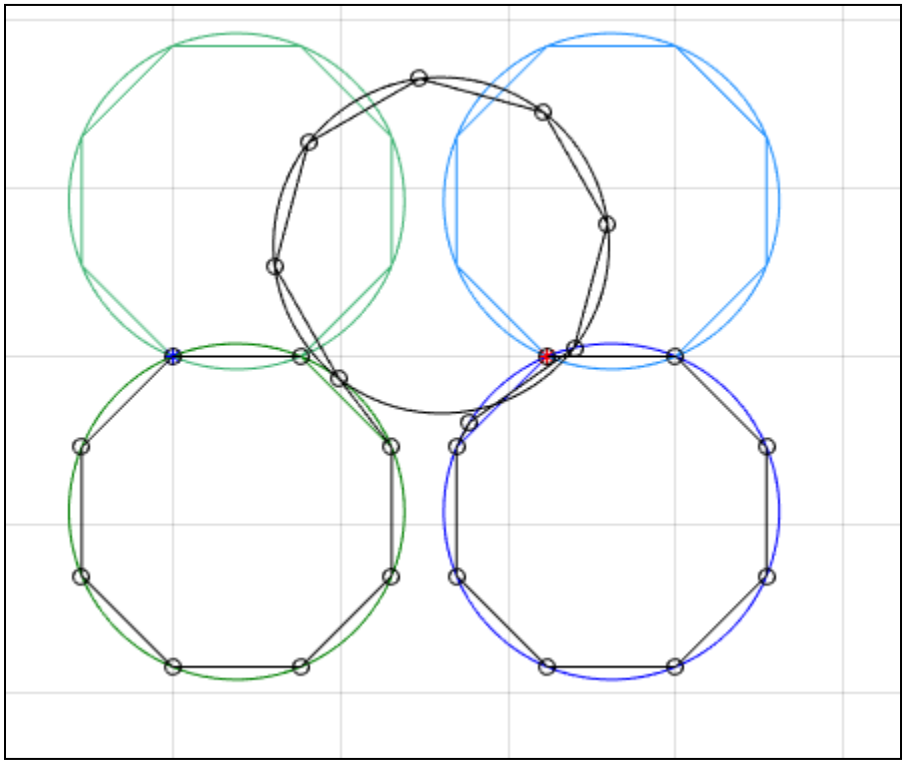

Figure 3.15. Example – CCC path with n=8. Figure shows how the complete CCC path have at most 2 short edges. The shortest edge between C1 and C2 is longer compared to the shortest edge between C2 and C3 for this particular problem.

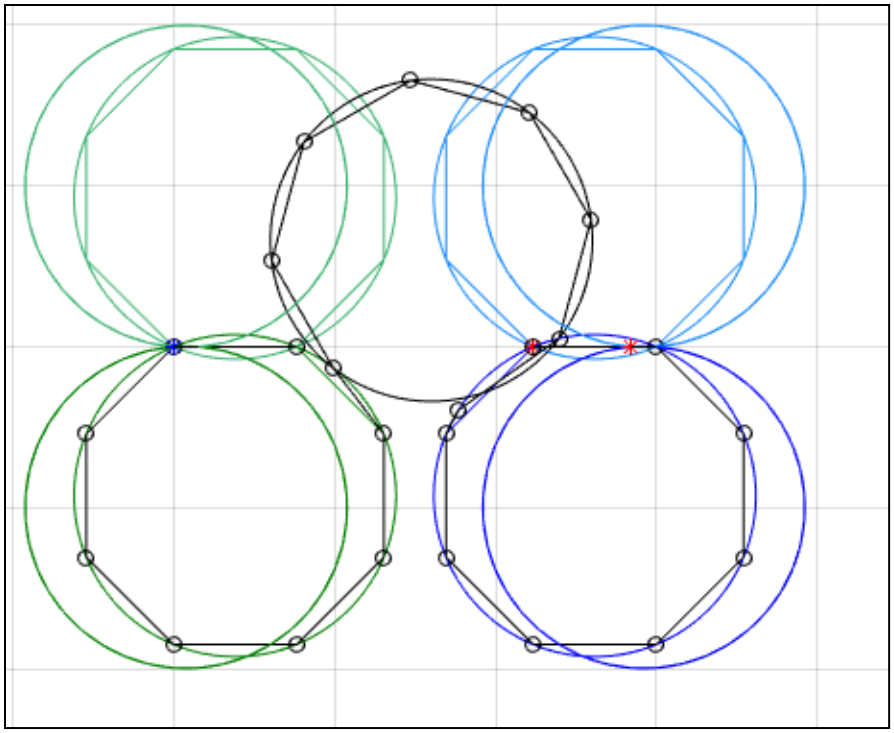

Figure 3.16. Comparison of CCC path Discrete vs Continuous. An overlay of the continuous Dubin's results over the n=8 discrete Dubins result show how the continuous case path is longer.

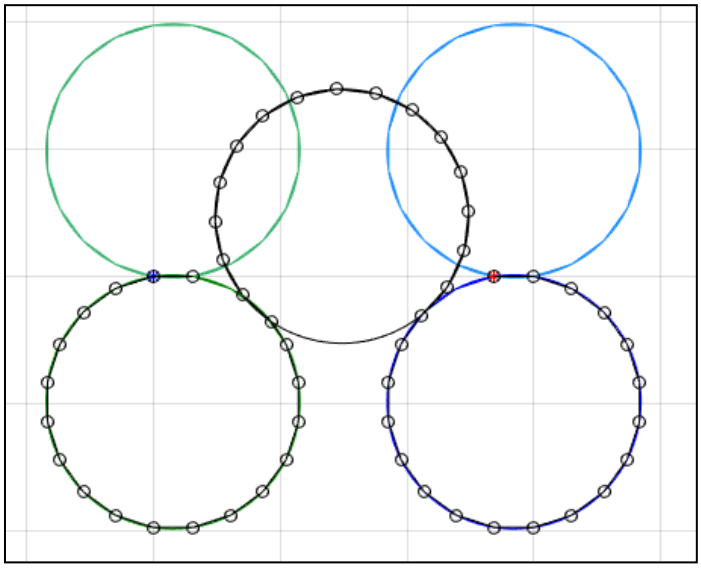

Figure 3.17 Showing CCC Discrete with 20 sides – Icosagon. Comparing this with the continuous Dubins results shown in figure 3.16 we can see how similar they are. Thus, reiterating that even continuous Dubin's path are better represented in a discrete Dubins sense because we can adjust how refined we want our discretization depending on the frequency at which we want to maintain our measurements and inputs to the control loop.

Discrete Dubin's path - As general case for continuous

A circular arc could be imagined as a discrete polygonal arc with infinite edges. In other words, a circle is a polygon with infinite edges with each edge of length limit to zero. Thus, the Discrete Dubin's results could be considered as a more generic form for the solutions to path planning problems. The continuous case results could be generated by setting the chord length parameter of the Discrete Dubin's close to zero or the number of edges parameter close to infinity. Moreover, since all sensors onboard an autonomous vehicle works based off the frequency of its internal clock, it is natural to code all trajectory planning and optimization algorithms in a discrete sense rather than a continuous one.

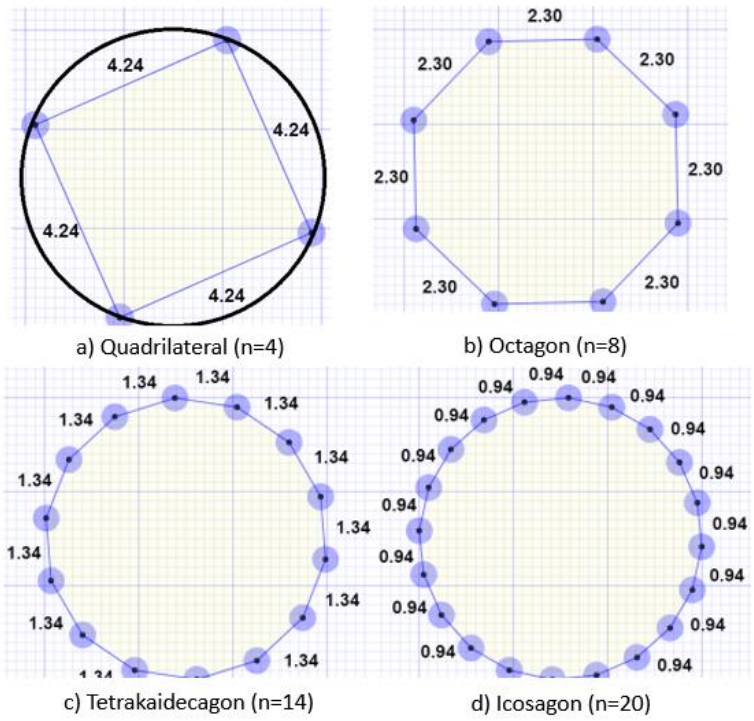

Figure 3.18 Polygons with various number of sides. (a) shows a quadrilateral with number of sides n=4, (b) shows an Octagon with n=8, (c) shows a Tetrakaidecagon with n=14 and (d) shows an Icosagon with n=20. All the four polygons are inscribed in the circle with the same radius of 3 units. As can be seen, the length of the edges decreases as the number of sides of the polygon increases for the same radius. As the number of sides tend to infinity, the length of the edges tent to zero. A circle can be thought of as polygon of infinite edges with infinitesimal length.

4. COMBINATORIAL MOTION PLANNING WITH DISCRETE GEODESICS

4.1 Introduction

Unlike the previous sections in which we dealt with a single vehicle; in this chapter we discuss the allocation of tour of q vehicles to p targets. The targets are also called interchangeable as waypoints. The initial location of the vehicle is also called as depot. For the tours that we plan in this section, we ask the vehicle to return to depot after the tour. This requirement is not necessary for our algorithms to work but is considered in order to extend the study by repeating through the same set of targets. For a time, t, let the position and heading of vehicle v_m be denoted by $\big(x(v_m,t), y(v_m,t), \theta(v_m,t)\big)$, where m is the index for the vehicle. Let each of the vehicles have an initial heading angle λ_m. The position of target c_n at time t is denoted by $(x(c_m,t), y(c_m,t))$. See Figure 4.1 which shows the initial location of the vehicle (depot) and the targets (waypoints). For this work, we assume the targets to be stationary. Therefore, $(\bar{x}(c_m), \bar{y}(c_m))$= $(x(c_m,t), y(c_m,t))$ for all t. Another assumption made is that the Euclidean distance between the initial position of each vehicle and a target, as well as between any two targets is greater that two times the minimum turning radius. The reason for this assumption is explained later.

The single vehicle algorithm is used to find the order of the targets to visit and the optimal trajectory for the vehicle to follow as shown in Figure 4.2. The multivehicle algorithm has the problem statement is as follows,

Given a set of targets $\{c_1, c_2 \ldots c_p\}$ and a set of vehicles $\{v_1, v_2 \ldots v_q\}$, (See Figure 4.3)

1) Find the sequence of targets for each vehicle to visit.
2) Find the path through the targets such that the discrete Dubin's constraints are met.

The objective is to minimize the total distance travelled by all the vehicles. Figure 4.4 shows how the distribute the waypoints among themselves and determines and order to visit the waypoints in the optimal trajectory. The problem is a Resource Allocation System RAS for q vehicles. RAS is not specific to our multitarget multivehicle problem, but rather is of a broader context in which the use of distributed resource allocation with energy efficient schemes are required in an optimized manner. For our case however, RAS applies to finding the optimal trajectory and targets for each vehicle to traverse through such that the total length travelled by all the vehicles is minimized.

If the discrete Dubin's constraints are not there, then the RAS is a multivehicle multidepot Euclidean traveling salesman problem. Additionally, for $q = 1$, the RAS is a single vehicle Euclidean traveling salesman problem.

Note that the problem is an asymmetric TSP since for a vehicle with two target A and B, the optimal path from A to B need not be the same as the optimal path from B to A. It needs to be noted that currently, there are no algorithms that given a constant approximation factor for solving an asymmetric travelling salesman problem (ATSP). This is true even if the costs of the path satisfy the triangular inequality. In this work, a constant approximation factor is presented for the discrete Dubin's RAS problem.

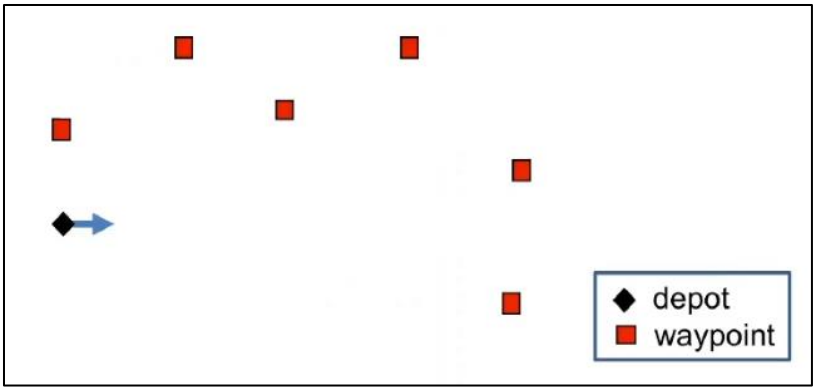

Figure 4.1 Multiple waypoints and single depot. The figure shows a single vehicle trajertory planning problem which has the depot location as well as the location information of the waypoints. The initial configuration of the vehicle is known. The final configuration location is known but the orientation is left free inorder that we can keep cycling through the targets if necessary.

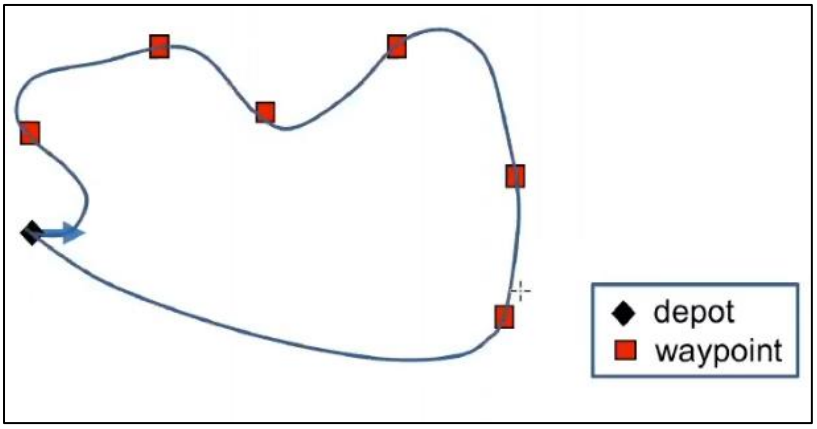

Figure 4.2 Single Vehicle Trajectory. The single vehicle algorithm is used to determine the order to visit the targets as well as to decide the optimal trajectory when traversing through.

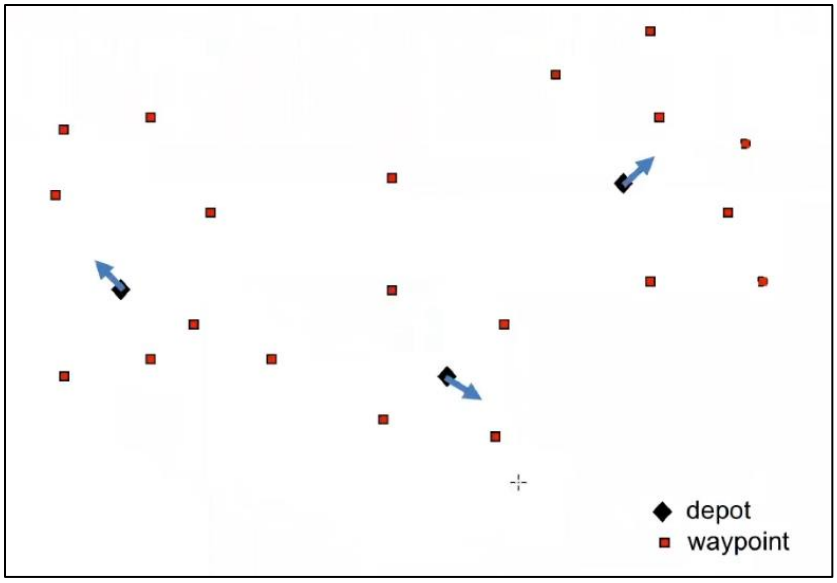

Figure 4.3 Multiple waypoints multiple vehicles. The figure shows how a typical Resource Allocation System (RAS) in the context of UAV path planning for multiple vehicles. The initial configuration is fully known. The vehicles are asked to return to depot after tour. The final configuration orientation is left free to allow for target cycling.

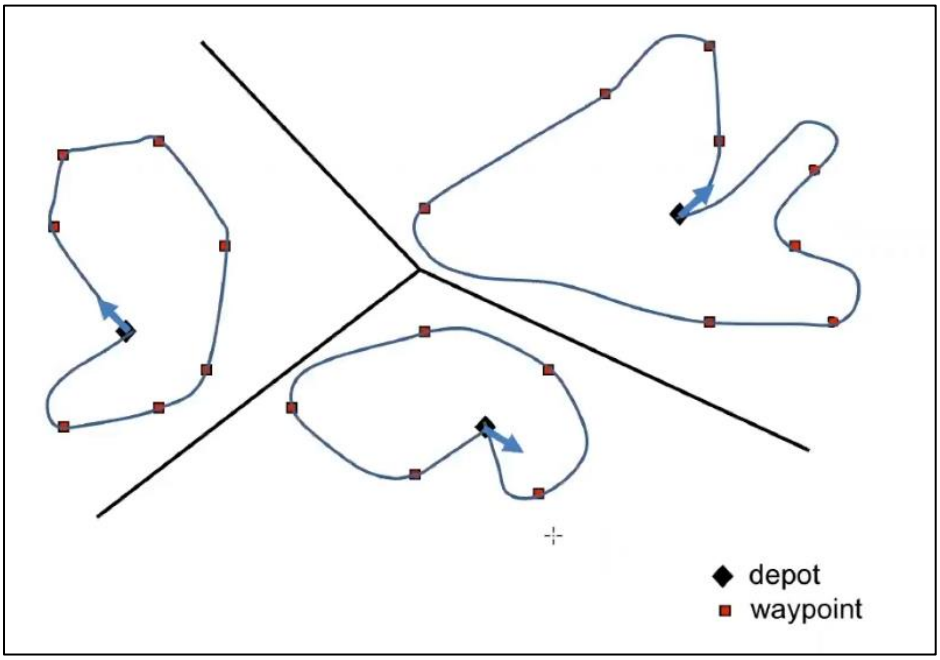

Figure 4.4 Multiple Vehicle Trajectory. The figure represents a typical solution showing how the vehicles distribute the targets among themselves and determine the order and the trajectory inorder to reduce the total distance travelled by all the vehicles.

4.2 Approximation algorithms

Single-Vehicle Discrete Dubin's Algorithm (SVDDA)

For a given set of targets and a single vehicle, we develop an algorithm called Single-Vehicle Discrete Dubin's algorithm (SVDDA) for the discrete Dubin's vehicle to find a path to travel from an initial configuration through all the targets in the shortest path.

A develop a subroutine called as the Discrete-Subroutine (D-sub) to find the shortest path between the vehicle and the next target is specified. The D-sub compares a left -turn path and a right turn path to the desired target in the required orientation and chooses the shorter path. The D-sub is explain after the SVDDA. The SVDDA is as follows

1) For a given set of targets and the initial configuration of the vehicle specified, check if the Euclidean distances between the targets themselves and with the vehicle are greater that two times the minimum turning radius.
2) Assume the kinematic constraints on the bounded curvature and the discrete Dubin's constraints are absent and solve the Euclidean traveling salesman problem using Christofides algorithm.
3) Store the output sequence of targets $\{c_1, c_2, c_3 \ldots c_p\}$.
4) With the initial configuration of the vehicle as the current configuration and the 1st target saved as the next target, use the D-sub to find the shortest path
5) Use the current vehicle configuration and proceed to the next target and repeat D-subroutine
6) Repeat Step 5 until path to the final target has been constructed.

D-subroutine

With both the current position and orientation information of the vehicle known and the next target position information known (the orientation at the target is free to be chosen), do the D-sub as follows

a) Find the distances of the two discrete Dubin's paths the vehicle could take by following the left-turn polygon (LP) and the right-turn polygon (RP).
b) Choose the path that has the smaller distance.
c) When the path is followed to target $(\bar{x}(c_m), \bar{y}(c_m),)$, the vehicle reaches the target at a particular orientation θ. Store $(\bar{x}(c_m), \bar{y}(c_m), \theta)$ as the vehicle configuration at the target.

Multiple-Vehicle Discrete Dubin's Algorithm (MVDDA)

For a given set of targets and multiple vehicles, we develop an algorithm MVDDA for the discrete Dubin's vehicles to select the targets they would visit, the sequence of targets for each vehicle to visit and the discrete Dubin's path for each vehicle to follow through their target sequence. The location of the targets and the initial location of the UAVs for a typical problem are shown in Figure 4.5 below. The MVDDA is as follows

1) Check if the Euclidean distances are $\geq 2r\ \cos(\emptyset/2)$. If yes, with the initial location of all the vehicles and targets known, construct a complete graph including all the vehicles and targets. The completed graph would look similar to Figure 4.6.

2) For any two vertices in the graph., assign the Euclidean distance between the vertices as the cost for the edge between them. For the edge that joins 2 vehicles, assign the cost to be zero (zero cost edges).
3) Use Prim's algorithm to find the minimum spanning tree of the graph. For n number of vehicles, the tree will have exactly $n-1$ edges zero cost edges. In the example considered we have two zero cost edges since we have three UAVs.
4) To get a tree for each vehicle, remove the zero cost edges. These edges are removed as shown in Figure 4.6.
5) For each vehicle and its corresponding tree, double the edges to get a Eulerian graph. See the corresponding step in Figure 4.7 with double edges.
6) Follow SVDDA to construct the tour for each vehicle. Figure 4.8 shows how the individual tours for the UAVs look like for the problem that we have considered.
7) Computed the Discrete Dubins optimal trajectory though the target sequence using the novel algorithms explained in Section 3.3.2. Figure 4.9 shows the optimal discrete Dubin's trajectory developed for our problem.

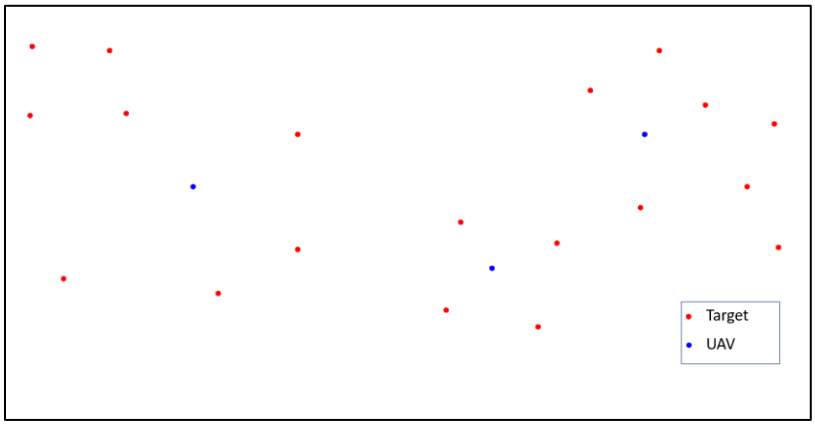

Figure 4.5 Given set of targets and initial locations of the UAV. The Euclidean distance between all the targets and UAVS need to be $\geq 2r\ \cos(\emptyset/2)$.

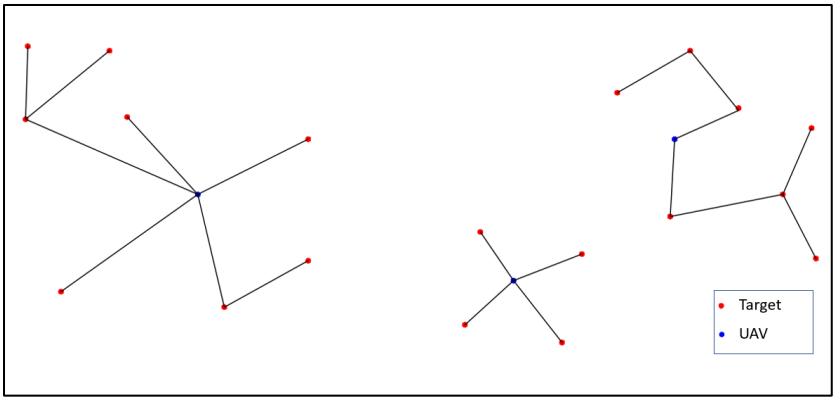

Figure 4.6 Construct complete graph including all vehicles and targets. The Prim's algorithm is used to arrive at this step. Since we have three UAVs, we will have two zero cost edges which need to be removed. Hence in this figure we do not see paths connecting any two UAVs to each other.

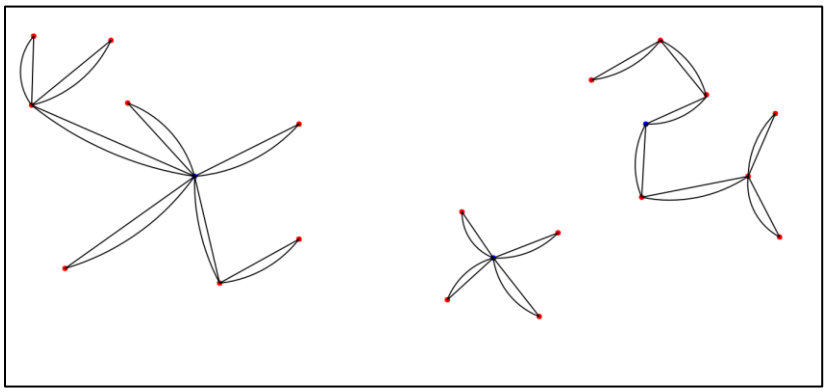

Figure 4.7 Double edges to get Eulerian graph. This is done to perform the Christofides algorithms to compute the optimal order or targets for each UAV.

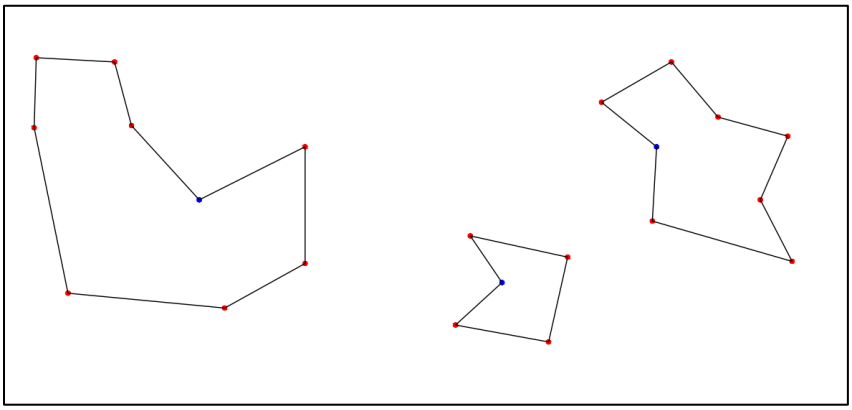

Figure 4.8 Showing tours constructed for each vehicle. This step considers the orientation at the target location free to be chosen.

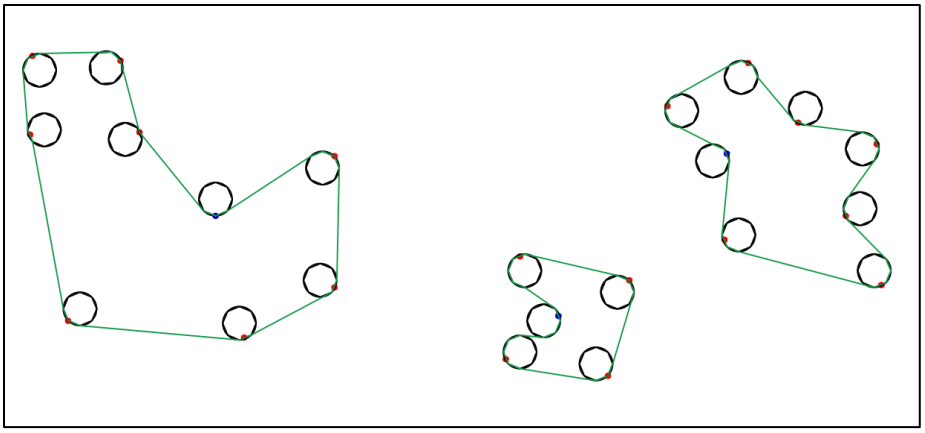

Figure 4.9 Showing tours constructed for each vehicle in Discrete Dubins. The Discrete dubins algorithm presented in Section 3.3.2 is used to compted optimal trajectories between the targets.

4.3 Discussions

The D-Subroutine, without the loss of generality is depicted in the figure below. The vehicle has a configuration of $\left(0,0,\frac{\pi}{2}\right)$ and the target is positioned in the X-Y plane at location T. The X-Y plane is divided into four quadrants. Assuming that the targets T and T' are placed at (x, y) such that $\sqrt{x^2 + y^2} \geq 2r$.

The path with the left-turn polygon LP is optimal if the $x > 0$, and the path with the right turn polygon RP is optimal if $x < 0$. Both LP and RP are optimal if $x = 0$ because the distances of both LP and RP are equal. Consider any point T or T' in the figure, the points T and T' are equidistant. The coordinates for T are $(x, -y)$, and the coordinates for T' are $(-x, -y)$. As shown in figure, the distance using path RP to T is less than using the LP. Likewise, the distance using the LP to T' is less than using the RP.

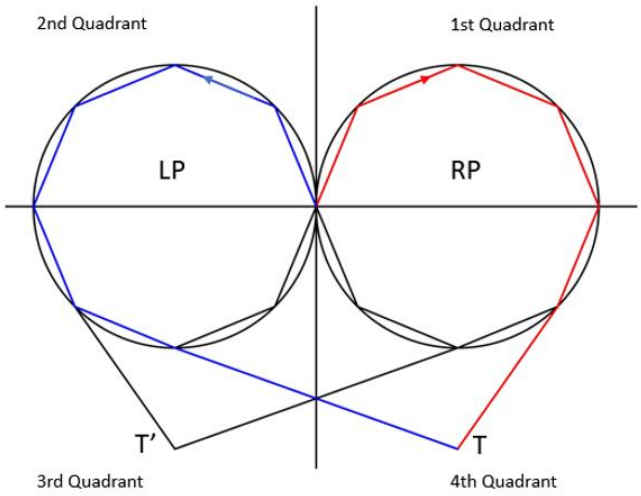

Figure 4.10 Example - D-Subroutine. Without the loss of generality, we can consider the location of the vehicle to be at $(0, 0)$ and oriented facing the north. If T is the target, we could see that it is shorter to traverse through the red path compared the blue path. The opposite is true is the target is located in the 3rd quadrant. This is the essence of the D-Sub

Since the distances are symmetric about the Y axis, we can consider only the RP path without the loss of generality. The RP path is in the first and fourth quadrant. Consider the set $\{(x, y): \sqrt{x^2 + y^2} = R\}$ in Figure 4.10 and $R \geq 2r$. Let $D(x, y)$ be the distance traveled by the discrete Dubin's path using the D-subroutine algorithm. $D(x, y)$ for Set S $\{(x, y): \sqrt{x^2 + y^2} = R\}$ is maximized when $x = 0$ and $y = -R$

This is because if you consider the point B with coordinates $(0, -R)$, also consider any other point A. A belongs to the Set S' $\{(x, y): x \geq 0, \sqrt{x^2 + y^2} = R\}$

The distance of the discrete Dubin's path in RP from point O to point B is greater than or equal to the distance of the discrete Dubin's path in RP from point O to point A.

Thus ,

$D(x, y)$ for Set S $\{(x, y): \sqrt{x^2 + y^2} = R\}$ is maximized when $x = 0$ and $y = -R$

The result along with the following discussion given us the constant approximation factor for the discrete Dubin's RAS problem.

Let R=$\sqrt{x^2+y^2} \geq 2r$. Then the ratio $\frac{D(x,y)}{\sqrt{x^2+y^2}}$ is maximized when x=0 and y= -2r. The maximum ratio is $\pi + 1 - \tan^{-1} 2$.

Since we know that $D(x,y)$ is maximized for when $x = 0$ and $y = -R$, it would be enough to consider the maximization ratio on the set of point $\{(0,-R): R \geq 2r\}$. Then from Figure 4.11,

$D(0,-R) = 2\pi r - 2\beta r + R$,

Where β is the angle OO'B.

$$\beta = \tan^{-1}(\frac{R}{r})$$

$$D(0,-R) = 2\pi r - 2(\tan^{-1}(\frac{R}{r}))r + R$$

If $R \geq 2r$, the ratio $\frac{D(x,y)}{\sqrt{x^2+y^2}}$ for $x = 0$ and $y = -R$ is

$$\frac{D(0,-R)}{R} = \frac{n(2\sin\frac{\emptyset}{2})}{R} + 1$$

Which is maximized at R=2r and thus the maximum ratio is $\pi + 1 - \tan^{-1} 2$. Now we can combine this bound with the bound of the Christofides algorithm to find the constant approximation factor for the discrete Dubin's single vehicle problem. The Christofides algorithm has a bound of 1.5 for the Euclidean traveling salesman problem. Combining this with our bound, we get the bound for the discrete Dubin's Euclidean travelling salesman problem with a single vehicle as $1.5 * \left[\frac{n(2\sin\frac{\emptyset}{2})}{R} + 1\right]$.

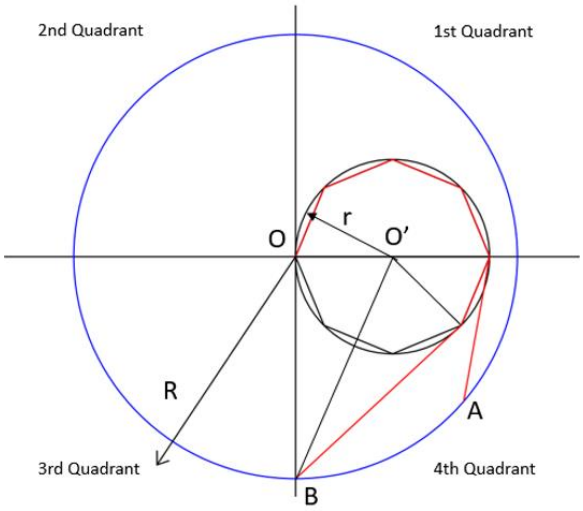

Figure 4.11 Example - Constant Approximation Factor. Without the loss of generality, we can consider the location of the vehicle to be at $(0, 0)$ and oriented facing the north. The constant approximation factor is the ratio of the distance traversed by the red path to the distance to reach the closest point on the blue circle (which is straight north equivalent to the length R).

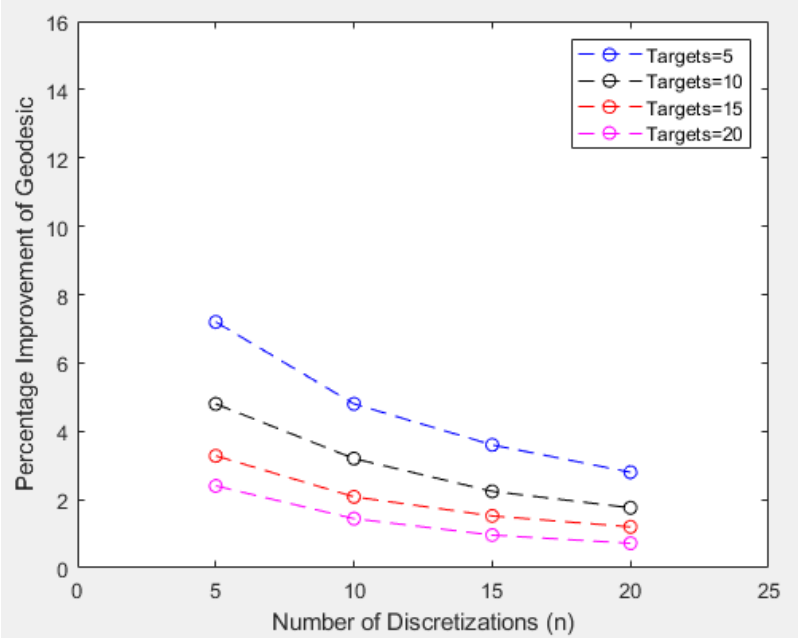

Figure 4.12 Percentage Improvement of Geodesics vs Discretizations. The figure shows that the Geodesic improves (Total distance traversed) gets shorter as the number of discretizations decreases. This is expected as we know that the discrete Dubin's paths are shorter than the continuous paths. We could also see that as the number of targets increase the percentage of improvement of the geodesic decrease. This can me attributed to the ratio of the minimum turn radius compared to the spatial location of the targets.

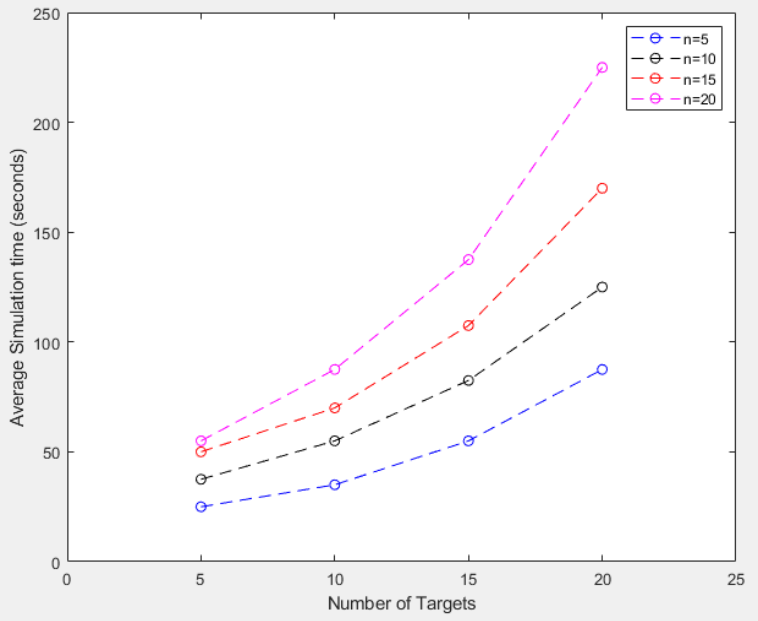

Figure 4.13 Average Simulation time vs Number of Targets. As expected the increase in the number of targets and the number of discretizations both increase the average simulation time because the algorithm subroutines need to be run as many more times.

Plots of the lower bound versus the number of discretization of the D-subroutine procedure for the various number of targets are shown in Figure 4.7. Both Figure 4.12 and 4.13 show the plots for different discretizations of the Discrete Dubin's polygonal. The percentage of improvement of lower bound is compared with the number of discretizations and the number of targets. As the number of discretization is increased, the lower bound increases. This is since the solution with larger number of vertices the solution would be closer to the results of the continuous Dubin's case and thus would have a greater lower bound. In other words, the geodesic worsens as we increase the discretization. We can also infer that the final value of the lower bound converges with the number of discretizations. Thus, we could approximate the continuous Dubin's results with reasonable accuracy without increasing the number of discretization to extremely large numbers. Clearly, after 15 iterations, the improvement of the lower bound is minimal thus giving as a precision bound for discretization. Figure 4.13 shows the computation time required for the

algorithm as a function of the size of the problem. Even though the computation time increases significantly as the number of discretization increases, the improvement in the lower bound is minimal after 15 discretization. Therefore, one can choose the discretization to have the computations done in a reasonable amount of time without compromising the bound on the accuracy. The computations were done on a AMD a10-6700 APU 3.70 GHz machine.

The Discrete Dubin's algorithms presented in this thesis gives us lower bounds that depend only on the number of discretization and the minimum turn radius. When the targets are closely spaced compared to the minimum turn radius (distance between the initial and final configuration is less than $2r$, we will need to consider CCC paths for the combinatorial motion planning problem. This thesis assumed the distance between the targets to be at least $2r$ and hence is concerned only with CSC discrete Dubin's paths. That said, about uncertainty, the solutions provided by the discrete Dubin's algorithm could provide equivalent geodesics wherein the vertex of any discrete polygonal arc could be replaced by the short edge that belongs to the arc and still not violate the turn and length constraints within the CSC paths. This would give us several alternate solutions to the same problem with the same shortest distance. Though the optimization problem gives us an exact alternate solutions, the measure of the quality the geodesic is given by the constant approximation factor. This makes the solution uncertain but the solution quality bounded. In other words, the solution quality is exact. The advantage of this is the fact that in the presence of obstacles in the path, or other constraints, we could switch from one solution to the other without compromising of the quality of the geodesic. The trajectory planning methodology used in this thesis deals with exactly solutions since the initial and final configurations are known as well as the locations of the targets with complete certainty.

5. CONCLUSIONS AND FUTURE WORK

This thesis is devoted for studying discrete Dubin's paths and its application in path planning missions. Discrete Dubin's path characteristics have been studied and algorithms have been presented to generate Discrete Dubin's paths for any arbitrary initial and final configurations in the absence of obstacles. We have addressed the path/route planning problem of single and multiple UAVs with multiple targets. The motion constraints have been induced to mandate the path of a UAV to have a discrete bounded curvature. This makes the UAV to travel along discrete arc-like paths to make turns. The shortest path between the two targets depends on the initial and final configuration of the UAVs. The configurations have both the location and orientation information. The orientation information at each of the targets to be visited needs to be known in addition to the sequence of the targets to be visited. Coupling both these set of decisions is what makes this planning problem hard to solve. We have developed a method to relax the orientation at each target and to find the bound for the routing problem with one UAV.

5.1 Contributions of section 3

Discrete Dubin's paths have been defined and characterized for both inflection-free and inflection edges. This includes both the CSC paths and CCC paths. The paths have been characterized with the following six characteristics :-

1) *A discrete circular arc of bounded curvature R with 2 edges would have a maximum turn angle α between the pre-edge and post-edge, given by*

$$\alpha = sin^{-1}\left(\frac{pre-edge}{2}\right) + sin^{-1}\left(\frac{post-edge}{2}\right)$$

2) *In CCC discrete Dubin's paths, an inflection edge between two polygonal paths cannot be shorter than the distance between points of intersection of the two circles that inscribe the polygonal paths*

3) *In CCC discrete Dubin's path, the two points of intersections between the adjacent circles that inscribe the polygonal paths must be* $\leq d_\theta$.

4) *A θ-discrete – geodesic circular arc can have at most one short-edge and the rest normal edges.*

5) *In CCC discrete Dubin's path, the distance between the centers of the adjacent circles that inscribe the polygonal paths must be* $\leq 2R$

6) *The path taken for a CSC inflection path with discrete polygonal arcs are always shorter than the path taken in with continuous circular arcs.*

Proofs of the six characteristics have been presented. A novel algorithm for computing the Discrete Dubin's path have been presented.

5.2 Contributions of section 4

The resource allocation problem of multiple targets with one UAV and multiple UAVs were studied with discrete Dubin's paths. The assumption that the distance between any two targets, or between the targets and the vehicles to be $\geq 2R\ cos\ (\emptyset/2)$ was made. This was a reasonable assumption in the context of UAVs. UAVs used in mapping and other tracking applications have a minimum turning radius of approximately 100m. If the vehicle is flying at a height of 150m with a 70-degree wide angle camera, the width of the area covered is at least 200m. We can see targets within 200m from the same UAV position. Thus, for planning paths applications as it applies to UAVs, is it reasonable to have distance assumption $\geq 2R\ cos\ (\emptyset/2)$.

With this assumption, we have developed a single vehicle algorithm based on relaxing the discrete Dubin's motion constraint as first to use Christofides algorithm to solve the Euclidean traveling salesman problem. Following which, we use the sequence of targets computed as the sequence for the vehicle to follow. We the introduced the discrete Dubin's paths between each target in the sequence found earlier. We also provide an approximation constant of $1.5 * \left[\frac{n(2\sin\frac{\emptyset}{2})}{R} + 1\right]$.

We extend this result to include multiple UAVs. With the initial location of all the vehicles and targets, we construct a graph. Using Prim's algorithm, we find the minimum spanning tree. Once the zero cost edges are taken out, we get a corresponding tree for each vehicle to traverse. Then for each vehicle we follow the single vehicle discrete Dubin's algorithm to develop the individual paths to follow.

5.3 Future work

5.31 Discrete Dubins Paths

A second class of constraints could be considered as a next immediate extension which includes considering the finite communication radius of the UAVs. There can be cases where the UAVs may be denied access to GPS signals. Due to their reliance on communication signals for successful navigation, it is important for UAVs to plan paths such the continuous and reliable signal range is ensured for the course of the tour. The radius of communication (R) being finite, the UAVs would not be able to receive or transmit signals beyond this range. It is common for deploying unattended ground sensors in the zone of interest. The path planning problem could be considered in two ways. Given the locations of the unattended ground sensors, how could we plan to path to ensure they vehicles are within the communication range. The alternate problem is to find out where to deploy the unattended ground sensors for a particular path on interest.

It would also be interesting to study how we could decentralize the implementation of the discrete Dubin's multiple vehicle algorithm. This is particularly useful in situation where not all vehicles know the information of all the targets. For this case, it would be useful to implement the minimum tree spanning algorithm in a decentralized manner. There has been literature to study the implementation of minimum spanning tree algorithm in a parallel manner [26]-[30]. Since each of the vehicle knows the information of only a particular set of targets. The vehicles would need to communicate to each other about the targets to come to a global spanning tree solution.

In this thesis, we have considered two dimensional discrete Dubin's paths. Extending our formulation to three dimensions is the next natural step. Three dimensional discrete Dubin's paths would be beneficial in flight plans considering the altitude dimension and related dynamics and constraints of a UAV. In rotary steering systems too three dimensional discrete Dubins paths are

needed to develop trajectories for directional wells. The challenge would however be in the visualization and verification of three dimensions polygonal spherical sections.

Another interesting path to work towards would be to develop the single and multiple vehicle algorithms if the distance between the targets, or the distance between the targets and the vehicles are $< 2R\ cos\ (\emptyset/2)$. In this case we would be considering CCC paths, but the challenge is that the Euclidean travelling salesman problem would no longer hold the triangular inequality between the nodes. Solving the asymmetric traveling salesman problem that do not satisfy the triangular inequality are a great challenge by itself. But the advantage to pursue research in this direction would be that in application other that for UAVs where the minimum turn radius is not as small compared to the range of sight the vehicle, implementing the CCC solutions into the ATSP problems with triangular inequality are the way forward.

5.31 Rotary Steerable System Hardware

We have designed a Rotary Steerable System (RSS) for directional drilling based on a push the bit system as shown in Figure 5.2. The tool is intended to be assembled on a collapsible rig system as shown in Figure 5.1. The rig system is unique in its ability to withstand all lateral, axial, and torsional reactional stress through the tensegrity structure that supports the drill assembly. The tensegrity structure was explored to study its effectiveness in supporting the system especially for application in Martian and Lunar surface where it would be advantageous to have a system that could collapse (for transport) and expand (when deployed). The Tensegrity structure is a path forward for transporting and deploying drilling systems. The focus of the thesis however was on the directional drilling capability of the tool as shown in Figure 5.2. The push the bit system is actuated through 4 pistons that can be individual controlled topside. Depending on the intended

direction of travel for the bit, the pistons are activated by pressurizing the appropriate line by manipulating a rotary valve topside which is not shown in the Figure 5.3. The pistons then, act against the surface of the steering collar which thereby pushes the bit to the desired direction.

The tool would serve as an effective test bed to study and implement discrete-Dubins trajectory following capabilities in directional drilling. The hydraulic schematic of the piston assembly can be seen in Figure 5.4. For this setup, two adjacent pistons are made to be actuated in tandem. Two pistons on the left of the schematic and controlled using the same valve signal while the two pistons on the right of the schematic are controlled by another valve signal. The responsiveness of the piston based on the pressure and flow inputs of the hydraulic fluid was modelled and studied. As could be seen from Figure 5.5, two sinusoidal input signals at a phase angle of 180 degrees are supplied to the piston pairs. The graph shows both (purple and magenta) the signals (having an amplitude of 1) and the position of the pistons (shown by green, red, blue, and yellow) varying between 0 and 0.5 inches. The input flow rate and pressure difference spikes are shown in Figure 5.6.

RSS systems that are used in real world application also have similar working principles as far as the end effector is concerned but would have different complex mechanisms depending on if the system is a point the bit, a push the bit, or a hybrid system. Given that we have a mechanism to steer the drilling bit instantaneous to the desired heading angle, we could now implement discrete Dubins algorithms and compare the costs with continuous Dubins trajectory. The Drilling Automation lab at Texas A&M University would be building the hardware for the RSS to study trajectory planning algorithms including discrete Dubin's algorithms.

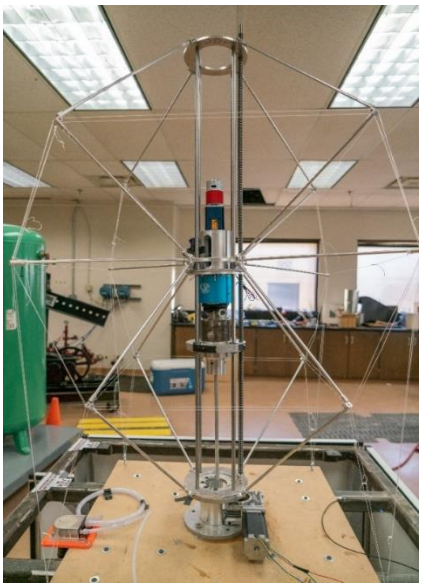

Figure 5.1. The Tensegrity Drilling Rig. We can see the tensegrity support structure holding the guide rods that support the Drill System shown in Figure 5.2 below.

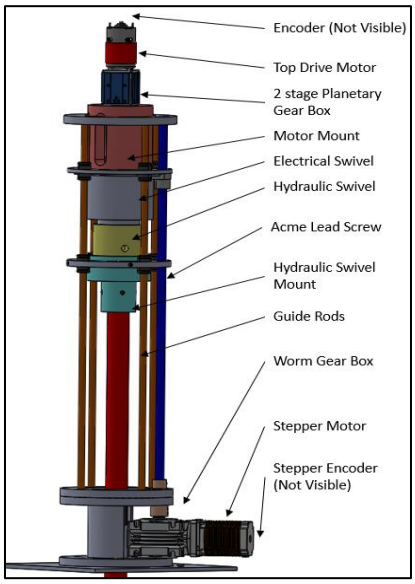

Figure 5.2. Drill system for the Tensegrity Drilling Rig. The figure shows the assembly drilling and actuation assembly inside the Tensegrity structure. The motor-planetary gearbox provides the drilling torque, while the stepper motor at the bottom of the assemble pulls the drilling assembly downwards into the formation by transmitting power to the motor mount through the acme lead screw. The guide rods help restrict motion to only the linear axis inline with the wellbore. The guide rods are supported by the Tensegrity Structure as shown in the Figure 5.1.

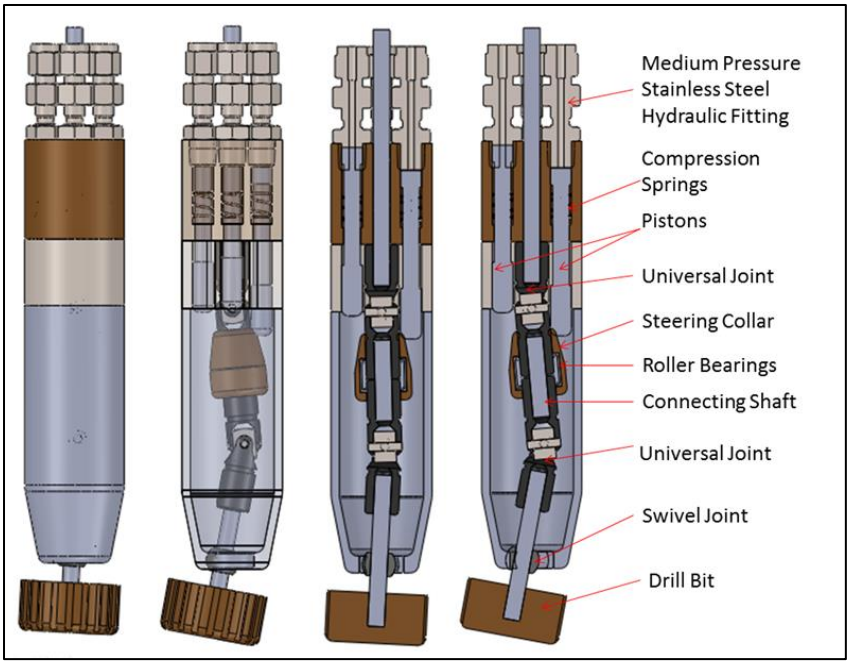

Figure 5.3. The Rotary Steerable System (RSS) to evaluate trajectory planning algorithms. This RSS assembly will be connected at the bottom of the drill collar in the Drill system shown in Figure 5.2. The system is a push-the-bit system but could perform as a point the bit system if the Steering Collar-Swivel Joint assembly can be made shorter.

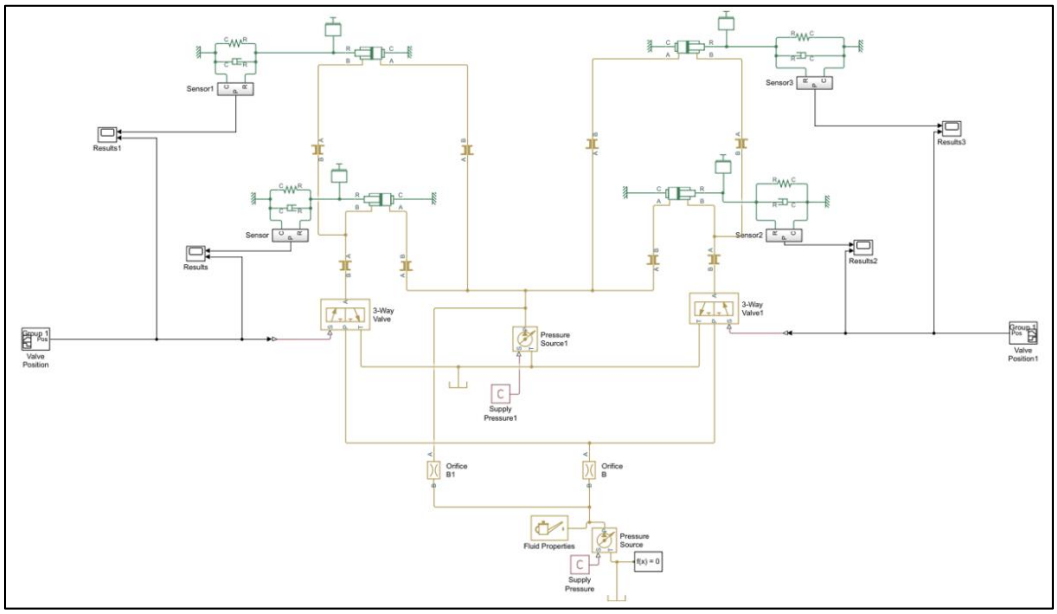

Figure 5.4. Hydraulic Schematic of the Rotary Steerable System. The schematic shows the four pistons being actuated by a hydraulic pump places upstream. There are two 3-way valves places inline upstream. Each valve controls two pistons so that maximum force could be transmitted to move the steering collar in the desired direction.

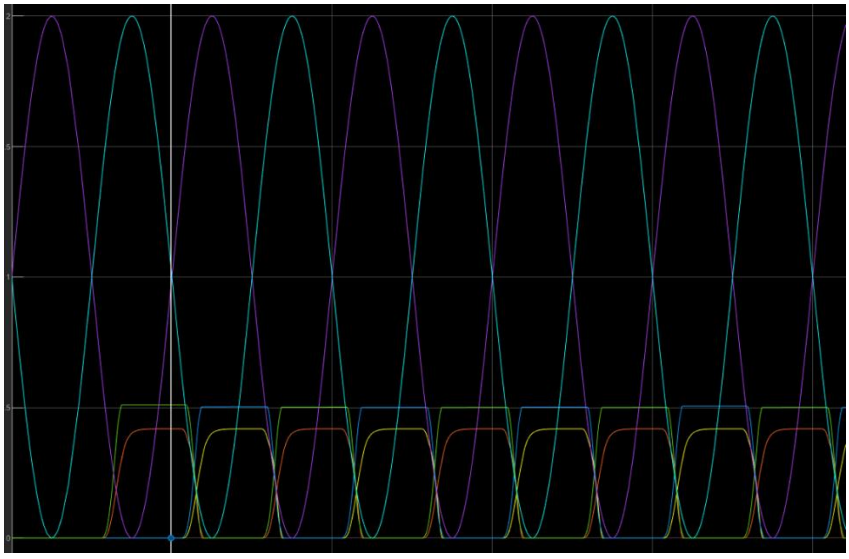

Figure 5.5. Valve signal and piston position graphs of the RSS pistons. Figure shows both the valve signals with the high amplitude (2 units) sinusoid waves in cyan and purple color. They are 180 degrees out-of-phase with each other to make the Drill Bit turn left – then right. This is to drill a zig-zag wellbore profile for testing the system performance. With smaller amplitude (0.5 and 0.375 inches), we could the displacement graphs of the pistons

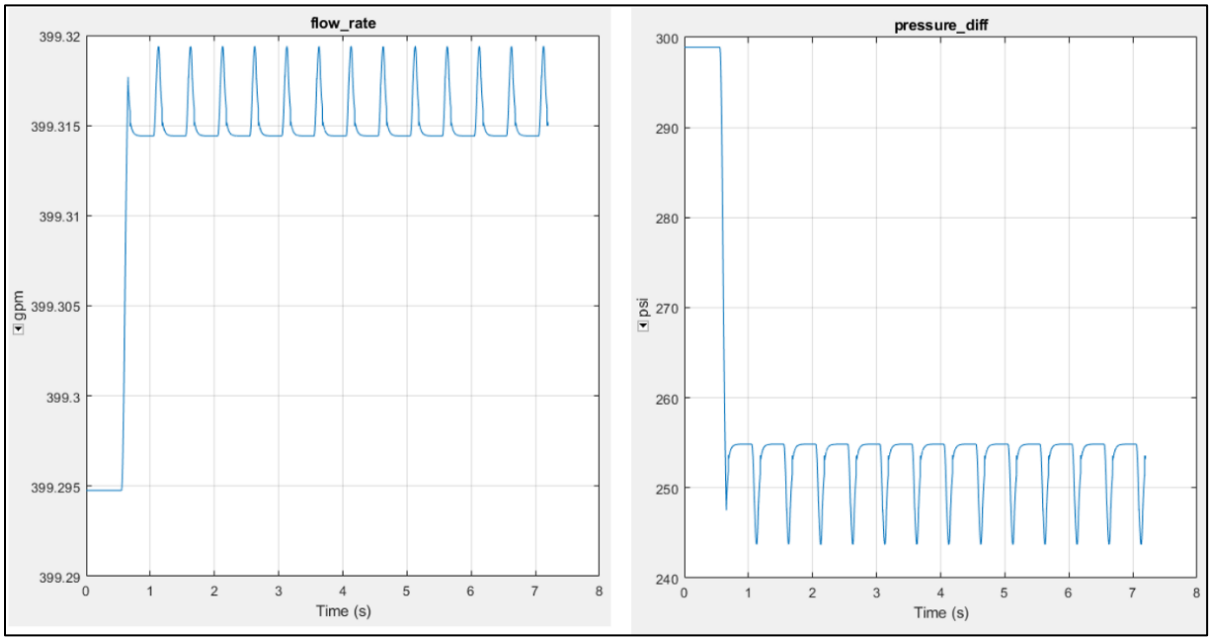

Figure 5.6. The input flow rate and pressure difference on the piston ports. We could see the input flow rate and pressure spikes at about 399 psi gpm and 255 psi. The spikes are created because of the flow dynamics and related inertia though the hydraulic flowlines from the pump up to the piston.

REFERENCES

[1] C. Schumacher, P. R. Chandler, and S. R. Rasmussen, "Task allocation for wide area search munitions," in *Proc. IEEE American Control Conference*, vol. 3, 2002, pp. 1917–1922.

[2] D. Gross, S. Rasmussen, P. Chandler, and G. Feitshans, "Cooperative operations in urban terrain (counter)," in *Proc. ISOP Defense and Security Symposium*, 2006, pp. 62 490G–62 490G.

[3] A. Girard, A. Howell, and J. Hedrick, "Border patrol and surveillance missions using multiple unmanned air vehicles," in *Proc. IEEE Conference on Decision and Control(CDC)*, vol. 1, 2004, pp. 620–625.

[4] S. Srinivasan, H. Latchman, J. Shea, T. Wong, and J. McNair, "Airborne traffic surveillance systems: Video surveillance of highway traffic," in *Proc. of the ACM International Workshop on Video Surveillance & Sensor Networks*, 2004, pp. 131–135.

[5] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, and C. Humphrey, "Supporting wilderness search and rescue using a camera-equipped mini uav," *Journal of Field Robotics*, vol. 25, no. 1-2, pp. 89–110, 2008.

[6] D. W. Murphy and J. Cycon, "Applications for mini vtol uav for law enforcement," in *Proc. ISOP Enabling Technologies for Law Enforcement and Security*, 1999, pp. 35–43.

[7] D. Jea, A. Somasundara, and M. Srivastava, "Multiple controlled mobile elements (data mules) for data collection in sensor networks," in *Distributed Com-*

puting in Sensor Systems. Springer, Berlin Heidelberg, 2005, pp. 244–257.

[8] G. Anastasi, M. Conti, and M. Di Francesco, "Data collection in sensor networks with data mules: An integrated simulation analysis," in *Proc. IEEE Symposium on Computers and Communications*, 2008, pp. 1096–1102.

[9] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, July 1957.

[10] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem," *Journal of the Operations Research Society of America*, vol. 2, no. 4, pp. 393–410, 1954.

[11] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," *Operations Research*, vol. 21, no. 2, pp. 498–516, 1973.

[12] E. L. Lawler, J. K. Lenstra, A. R. Kan, and D. B. Shmoys, *The traveling salesman problem: a guided tour of combinatorial optimization.* Wiley, New York, 1985, vol. 3.

[13] M. Padberg and G. Rinaldi, "A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems," *SIAM Review*, vol. 33, no. 1, pp. 60–100, 1991.

[14] G. Gutin and A. P. Punnen, *Travleing Salesman Problem and its Variations.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.

[15] D. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook, *The Traveling Salesman Problem - A Computational Study.* Princeton University press, Princeton, 2006.

[16] G. Reinelt, *The Traveling Salesman Problem - Computational Solutions for TSP Applications.* Lecture notes in Computer Science, No. 840, Springer-Verlag, Berlin Heidelberg, 1994.

[17] M. Held and R. M. Karp, "The traveling-salesman problem and minimum spanning trees," *Operations Research*, vol. 18, no. 6, pp. 1138–1162, 1970.

[18] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem," *Journal of the Operations Research Society of America*, vol. 2, no. 4, pp. 393–410, 1954.

[19] P. Oberlin, S. Rathinam, and S. Darbha, "A transformation for a multiple depot, multiple traveling salesman problem," in *Proc. IEEE American Control Conference*, 2009, pp. 2636–2641.

[20] X. Goaoc, H. Kim, and S. Lazard, "Bounded-curvature shortest paths through a sequence of points," INRIA, Tech. Rep., 2010.

[21] G. Warwick, "Lightsquared tests confirm GPS jamming," Aviation Week, July 2011.

[22] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via GPS spoofing," *Journal of Field Robotics*, vol. 31, no. 4, pp. 617–636, 2014.

[23] S. Rathinam, R. Sengupta, and S. Darbha, "Resource allocation algorithm for multivehicle systems with nonholonomic constraints," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 1, pp. 98–104, 2007.

[24] S. Yadlapalli, W. Malik, S. Darbha, and M. Pachter, "A lagrangian based algorithm for a multiple depot, multiple traveling salesman problem," *Nonlinear Analysis: Real World Applications*, vol. 10, no. 4, pp. 1990 – 1999, August 2009.

[25] P. Oberlin, S. Rathinam, and S. Darbha, "Today's traveling salesman problem," *IEEE Robotics and Automation Magazine*, vol. 17, no. 4, pp. 70 – 77, December 2010.

[26] W. Malik, S. Rathinam, and S. Darbha, "An approximation algorithm for a symmetric generalized multiple depot, multiple travelling salesman problem," *Operations Research Letters*, vol. 35, no. 6, pp. 747–753, 2007.

[27] S. Manyam, S. Rathinam, and S. Darbha, "Computation of lower bounds for a multiple depot, multiple vehicle routing problem with motion constraints," in *Proc. IEEE Conference on Decision and Control (CDC)*, 2013, pp. 2378–2383.

[28] W. Malik, S. Rathinam, S. Darbha, and D. Jeffcoat, "Combinatorial motion planning of multiple vehicle systems," in *Proc. IEEE Conference on Decision and Control*, 2006, pp. 5299 – 5304.

[29] K. Savla, E. Frazzoli, and F. Bullo, "On the point to point and traveling salesman problem for dubins' vehicle," in *Proc. IEEE American Control Conference*, 2005, pp. 786–791.

[30] K. Helsgaun, "An effective implementation of the lin-kernighan traveling salesman heuristic," *European Journal of Operations Research*, vol. 126, no. 1, pp. 106 – 130, 2000.

[31] Z. Tang and U. Ozguner, "Motion planning for multitarget surveillance with mobile sensor agents," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 898–908, 2005.

[32] H. Chitsaz and S. LaValle, "Time-optimal paths for a dubins airplane," in *Proc. IEEE Conference on Decision and Control*, 2007, pp. 2379–2384.

[33] C. Tomlin, I. Mitchell, and R. Ghosh, "Safety verification of conflict resolution manoeuvres," *IEEE Transactions on Intelligent Transportation Systems*, vol. 2, no. 2, pp. 110–120, 2001.

[34] S. M. LaValle, *Planning algorithms.* Cambridge University Press, New York, 2006.

[35] M. L. Fisher, "The lagrangian relaxation method for solving integer programming problems," *Management Science*, vol. 50, no. 12 supplement, pp. 1861–1871, 2004.

[36] G. Nemhauser and L. Wolsey, *Integer and Combinatorial Optimization.* Wiley-Interscience Publication, New York, 1988.

[37] C. E. Noon and J. C. Bean, "An efficient transformation of the generalized traveling salesman problem," *INFOR*, vol. 31, no. 1, pp. 39 – 44, 1993.

[38] A. Hashemi, Y. Cao, D. W. Casbeer, and G. Yin, "Unmanned aerial vehicle circumnavigation using noisy range-based measurements without global positioning system information," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, pp. 31 009–31 019, 2015.

[39] Y. Cao, J. Muse, D. Casbeer, and D. Kingston, "Circumnavigation of an unknown target using uavs with range and range rate measurements," in *Proc. IEEE Conference on Decision and Control (CDC)*, 2013, pp. 3617–3622.

[40] I. Shames, S. Dasgupta, B. Fidan, and B. Anderson, "Circumnavigation using distance measurements under slow drift," *IEEE Transactions on Automatic Control*, vol. 57, no. 4, pp. 889–903, April 2012.

[41] T. H. Summers, M. R. Akella, and M. J. Mears, "Coordinated standoff tracking of moving targets: control laws and information architectures," *Journal of*

Guidance, Control, and Dynamics, vol. 32, no. 1, pp. 56–69, 2009.

[42] K. Steiglitz and C. H. Papadimitriou, *Combinatorial optimization: Algorithms and complexity.* Prentice Hall, New Jersey, 1982.

[43] K. Helsgaun, "An effective implementation of the Lin-Kernighan traveling salesman heuristic," *European Journal of Operational Research*, vol. 126, no. 1, pp. 106–130, 2000.

[44] R. Sharma, R. Beard, C. Taylor, and S. Quebe, "Graph-based observability analysis of bearing-only cooperative localization," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 522–529, April 2012.

[45] R. Jonker and T. Volgenant, "Transforming asymmetric into symmetric traveling salesman problems," *Operations Research Letters*, vol. 2, no. 4, pp. 161–163, 1983.

[46] D. Applegate, R. Bixby, V. Chvatal, and W. Cook, "Concorde tsp solver," *See: http://www. tsp. gatech. edu/concorde. html*, 2005.

[47] G. Reinelt, "TSPLIB: A traveling salesman problem library," *ORSA Journal on Computing*, vol. 3, no. 4, pp. 376–384, 1991.