AMERICAN UNIVERSITY OF BEIRUT

Unmanned Aerial Vehicle Path Planning for Surveillance Missions with Regions of Interest

by
Mohammad Tuqan

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Mechanical Engineering
of the Maroun Semaan Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
July 2019

AMERICAN UNIVERSITY OF BEIRUT

Unmanned Aerial Vehicle Path Planning for Surveillance Missions with Regions of Interest

by

Mohammad Tuqan

Approved by:

Dr. Naseem Daher, Assistant Professor — Advisor
Electrical and Computer Engineering

Dr. Elie Shammas, Associate Professor — Co-Advisor
Mechanical Engineering

Dr. Daniel Asmar, Associate Professor — Member of Committee
Mechanical Engineering

Dr. Imad Elhajj, Professor — Member of Committee
Electrical and Computer Engineering

Date of thesis defense: July 18, 2019

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: Tuqan Mohammad Marwan

Last First Middle

☑ Master's Thesis ○ Master's Project ○ Doctoral Dissertation

☑ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

□ I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after: **One ___ year from the date of submission of my thesis, dissertation or project.**
Two ___ years from the date of submission of my thesis , dissertation or project.
Three ___ years from the date of submission of my thesis , dissertation or project.

M. Tuqan 05/08/2019

Signature Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

Acknowledgements

I want to thank my advisors Dr. Naseem Daher and Dr. Elie Shammas for their guidance and input throughout this thesis and in making this experience a truly beneficial and exceptional one.

I would also like to mention how grateful I am for having the constant support of my parents, Marwan and Muneera, and my siblings, Muneer, Mohannad, and Masa. Thank you for always believing in me.

An Abstract of the Thesis of

Mohammad Tuqan for Master of Engineering
Major: Mechanical Engineering

Title: **Unmanned Aerial Vehicle Path Planning for Surveillance Missions with Regions of Interest**

Surveillance missions entail visiting defined sites of interest in an environment to perform a desired task. The core of the decision making lies in planning the path and trajectory to achieve the goals assigned with the task; often modeled as a Traveling Salesman Problem (TSP). Several algorithms exist in literature for tackling the TSP and its variants. However, the problem is often accompanied with heavy computational effort. As a result, in several distinct approaches in the literature, the optimality of the solution comes at the expense of computational efficiency. In addition, each of the existing algorithms addresses a specific version of the problem and may not be applicable under different constraints or conditions imposed on the system. In this thesis, a path planning algorithm is proposed for surveillance missions that do not only require visiting certain known specified regions, but also exploring these regions. A planar kinematic model with bounded steering rate is set as a motion model, and the region exploration is imposed as traversing an arc with a nonzero arc-length around a point. Using the above two assumptions, a time-optimal control problem is defined to ensure completely visiting and exploring all of the sites, formulated as a Dubins Traveling Salesman Problem with Neighborhoods. This study builds on previous work done on the convex optimization solution to derive a geometrical representation of the local minima for path headings. Hence, solving for the optimal local Dubins maneuver between every pair of consecutive regions along the visiting sequence that corresponds to the global shortest path. The solution is validated to converge to that of the brute force approach with simulation results demonstrating the high computational efficiency of the proposed algorithm for maps with up to 500 regions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Unmanned Aerial Vehicles (UAVs) have become an active area of research in robotics and control engineering for the wide range of applications within several industries. Due to its flexible mobility and the recent drop in the purchase price, this platform has boomed in the field of engineering technologies. Furthermore, the rise of autonomous systems has also played a role in this boom. Whether for surveillance, exploration or search tasks, autonomy has been proven to increase the efficiency and further ease the completion of such tasks. By installing a camera and an actuated mechanism, such as a robotic arm (Figure 1.1), the already wide range of applications is further expanded; Figure. 1.2 shows a UAV used in rescue missions and fire combat.

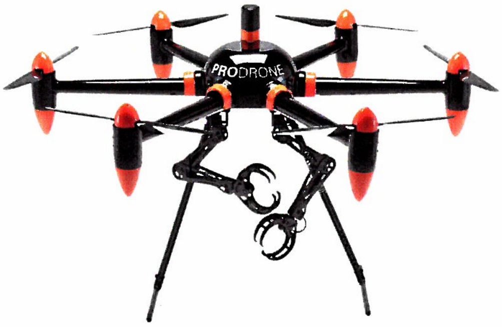


Figure 1.1: A robot arm drone product by Prodrone [1]

Adding perception features (mainly localization and mapping) to such platforms can further facilitate autonomy and intelligence integration. Perception collects data points that maps the surrounding environment and accordingly localize the robot or vehicle within that environment (awareness). This data is processed by decision making algorithms to specify actions for the robot to execute in order to achieve a desired goal or task. The core of this decision making lies in defining the path that must be followed during a task. The path is built

Figure 1.2: Rescue tasks performed by a drone firefighter [2]

and defined based on the task deliverable and the constraints imposed by the environment and the dynamics of the robot/vehicle itself [5, 6, 7]. Furthermore, autonomy provides the flexibility in conducting cooperative missions by multiple vehicles (robotic swarms) to perform more complex tasks [8].

Path planning involves finding the optimal path, relative to a defined cost function, to perform a specific task. However, before planning the path, the problem to be solved by the planner must be properly defined. First, a cost function that is generally based on time, energy, or distance must be defined. This cost/objective function must be optimized by the planner in order to determine the desired solution. Second, a vehicle model is adopted to describe the states of the dynamic system and to define its constraints. The constraints on the dynamics are involved in planning the motion of the vehicle over the path solved by the path planner. Consider a non-holonomic vehicle with a minimum turning radius, ρ, that is operated with an A* planner. The planner solution must be translated into a sequence of Dubins, Curve-Curve-Curve (CCC) or Curve-Straight-Curve (CSC), paths to be rendered as a feasible maneuver [9, 10]. This often requires refining the A* solution to a smoother waypoint sequence [10]. In addition, the choice of the model representing the vehicle may differ under different operating conditions. For instance, modeling a ground vehicle at lower speeds, where the dynamics are less dominant, can be achieved with a kinematic model without much compromise on fidelity. At higher speeds on the other hand, where a kinematic model might fail, a dynamic model will demonstrate a more accurate representation of the vehicle [5, 11]. Finally, a configuration or search space that closely represents the environment in which the vehicle is expected to operate must be defined. A simple example of a configuration space is the 2D occupancy grid in Euclidean space, as used in A* algorithm, where every point is defined by a 2D pose in space. A cost function may also be defined in a configuration space to model obstacles, altitude, and other features in a given environment. Note, however, that the configuration space, cost function, and vehicle model are often based on the characteristics and the deliverable of the desired objective, which range from finding a feasible obstacle-free path connecting two waypoints in a

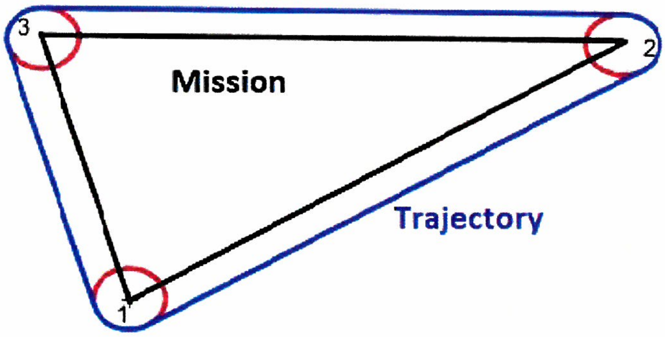


Figure 1.3: Mission vs trajectory planning

configuration space to more complex surveillance and exploration missions.

This thesis tackles a problem that is motivated by search and inspection tasks for infected trees in dense forests. The trees are modeled as neighborhoods or regions of interest, hence classifying the problem as a Dubins Traveling Salesman Problem with Neighborhoods (DTSPN) [12, 13, 14], as will be explained later. In order to search the regions of interest, the exploration factor is accounted for by traversing a non-zero arc around the region. As such, the problem is broken down into two sub-problems: mission and trajectory planning. As demonstrated in Figure 1.3, mission planning, often referred to as the travelling salesman problem (TSP) in literature, involves solving for the optimal sequence for the path to visit the regions of interest. On the other hand, trajectory planning invokes the Dubins Touring Regions Problem (DTRP) that is concerned with the optimal path maneuver to visit and explore regions of interest along an optimal sequence.

This work aims at developing a novel algorithm that solves for the optimal path to visit a set of regions of interest while traversing a non-zero arc around every region along the path [15]. We build upon the complex optimization approach utilized in solving the DTSPN [13] to define the local minima heading between every pair of regions. These local minima of the Dubins maneuver are then used to solve for the global optimal path. The main contribution of this thesis is the proposal of a computationally efficient and feasible approach to solve for the constrained DTSPN to visit, and explore, regions of interest in a given map. This is done by redefining the configuration space based on the 2D coordinates of the region center instead of the 2D position along the path, as is the case of the classical Dubins maneuvers. It is assumed that the regions are of equal radii, r_{region}, and satisfy the D_4 condition, which stipulates that the distance between any pair of regions in the map must be greater than $4r_{region}$. A further extension to the problem is introduced for regions of non-equal radii, where the same approach is adopted to develop a planner that solves for a feasible path that visits and explores every region of interest.

The thesis is organized as follows: chapter 2 provides an overview of prior work in the fields of optimal path planning the travelling salesman problem, along with solutions and algorithms that are currently in hand. The problem is formulated in chapter 3 by first defining the configuration space and the initial and final configurations of every local maneuver, and then analyzing the brute force approach to motivate the proposed approach. Chapter 4 demonstrates the efficiency and feasibility of the proposed approach by starting with the parameterization of Dubins CSC paths, then arriving at a geometrical derivation of the local minima heading, and finally validating the proposed algorithm against the brute force solution. In chapter 5, the problem is extended to regions of non-equal radii and a proposed planner is demonstrated. Finally, chapter 6 presents a conclusion and provides an outlook of the work presented in this thesis.

Chapter 2

Literature Review

The objective of this work is to develop an algorithm that solves for the optimal path that visits, and explores, regions of interest in a given map. Surveillance missions that involve visiting cities or points of interest is often modeled as a TSP [16]. Since the regions of interests are modeled as circular regions of equal radii, the problem can be expanded to DTSPN. In DTSPN, the problem is divided into two sub-problems: mission and trajectory planning. The proposed problem, however, imposes an exploration constraint on the desired trajectory plan that involves traversing a non-zero arc around every region. As such, the nature of Dubins maneuvers (arc and straight line segments) is utilized to plan this constrained path. In this chapter, we present some background material, starting with the typical Dubins vehicle and its associated optimal path, followed by an overview of the literature on path planning in general. The chapter ends with a more specific overview on surveillance missions and the TSP.

2.1 Dubins Optimal Curves

For a given map with defined circular regions, and provided a path sequence to visit all the regions, the shortest global path maneuver can be defined as a concatenation of local maneuvers between every pair of consecutive regions along that sequence. In this study, the local maneuver is considered as a Dubins optimal maneuver solution connecting an initial and final configuration states. Dubins considered a graphical approach to build his minimum path synthesis for vehicles with constrained curvature, or with minimum turning radius [9]. The synthesis proves that for a given initial and final pose, the shortest path belongs to one of the six following CSC and CCC segments (Figure 2.1): RSR, RSL, LSR, LSL, LRL, and RLR, where R denotes turning right, L denotes turning left, and S denotes moving in a straight line. Figure 2.2 demonstrates the four possible CSC paths between a pair of configuration states. This conclusion arises from the fact that the shortest path is found to be a concatenation of straight line segments or arcs with radii that are equal to the minimum turning radius of the vehicle. This was later validated in 1992 by Boissonnat [17] using Pontryagin's maximum principle. It is important to note that Dubins vehicle is assumed to be travelling at constant speed between the pair of poses, hence having a constant minimum turning radius along the entire Dubins path segment. A first-order model of a unicycle with limited steering rate and a minimum turning radius, ρ, referred to as a Dubins vehicle [9] is considered for the motion model. The state of the vehicle model can then be defined by the configuration $(x, y, \psi) \in SE(2)$, where (x, y) denote the position of the vehicle in $\mathbb{R}^2$ and the heading angle of the vehicle is denoted by $\psi \in \mathbb{S}^1$. Furthermore, if the vehicle is assumed to be travelling at a constant velocity, v, then the model can be formulated as follows:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{pmatrix} = v \begin{pmatrix} \cos\psi \\ \sin\psi \\ \frac{u}{\rho} \end{pmatrix}, \mid u \mid \leq 1. \tag{2.1}$$

The yaw rate, $\dot{\psi}$, is bounded by the control input, u, and the sign of u denotes the direction of the turn: clockwise (turning right) and counter-clockwise (turning left). Control input, u_{max}, denotes the maximum steering effort corresponding to the minimum turning radius of the vehicle traveling at constant velocity, v.

The time-optimal problem for a start to an end configuration has been solved for the Dubins vehicle and in fact, the time optimal paths are the well-known Dubins paths [9]. It was proven in [9] that the shortest path connecting two configurations in a plane is one of six possible maneuvers that can be divided into two types: CSC type (LSL, LSR, RSR, and RSL) and CCC type (RLR and LRL), where C is an arc with a minimum turning radius and S is a straight line with no steering, R and L indicate whether the C section is turning clockwise or counter-clockwise (Table 2.1).

The emergence of the Dubins vehicle model have made the Dubins path approach a convenient motion planning tool for non-holonomic vehicles in wide ranges of operation: ground, underwater, and aerial [18, 19, 20, 21, 22]. In [10], the authors use 3D Dubins maneuvers to translate the optimal path solved for by the A* planner to velocity and orientation commands for non-holonomic ve-

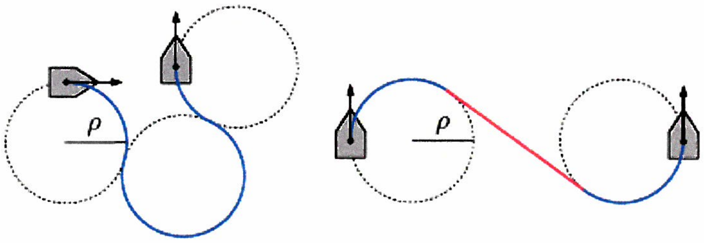


Figure 2.1: Dubins CCC and CSC maneuver [3].

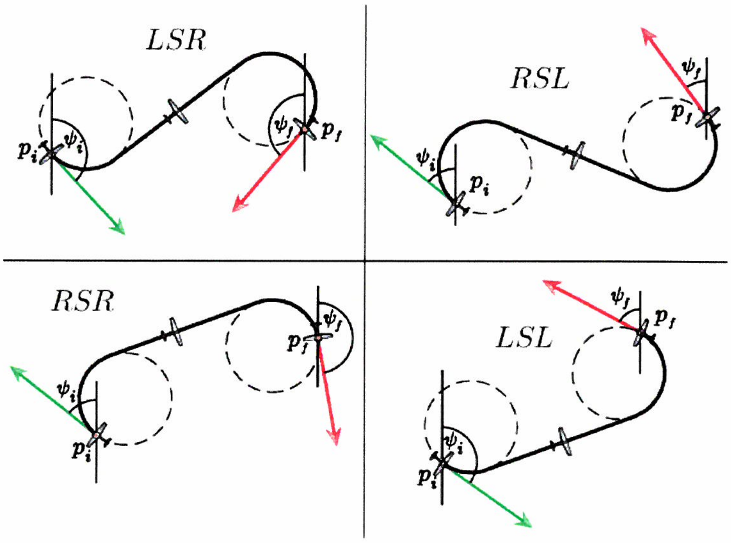


Figure 2.2: Demonstration of CSC Dubins maneuver [4].

Symbol	u
L	u_{max}
R	$-u_{max}$
S	0

Table 2.1: The path segments from which all optimal Dubins paths can be constructed

hicles. The nature of Dubins solution, straight lines and arcs, have rendered it a tempting tool for surveillance missions involving regions of interests modeled as disks (DTSPN) [16]. In this thesis, the path is planned based on the shortest sequence of Dubins paths to visit and explore all the regions of interest in the environment. In addition, Dubins maneuvers are leveraged to abide the planner to the exploration factor imposed on the desired solution of having to traverse a non-zero arc around every region. The classical Dubins problem, however, is extended by redefining the initial and final configuration for each local maneuver; the configuration is defined for every region as $(x, y, \theta) \in SE(2)$, where (x, y) denote the position of the center of the region in $\mathbb{R}^2$, and the heading angle of the vehicle is denoted by $\theta \in \mathbb{S}^1$. For every local maneuver, the initial and final headings are set according to the position of the initial and departure (the region from which the path departs the final regions towards) regions with respect to the final region.

2.2 Optimal Path Planning

In the context of robotic path planning, several parameters must be defined by the task or application deliverable in order to adopt or develop the appropriate planner. First, a defined vehicle model is utilized to represent the maneuvering capability of a robot or vehicle. Accordingly, a configuration space is defined to represent the states of the vehicle in the operating environment. Finally, a cost function is adopted based on a defined cost metric to optimize the path according to the constraints imposed on the system. Path planning algorithms are thereby developed based on the availability of these three elements: vehicle model, configuration space, and cost function.

Path planning algorithms can be divided into two main categories: node-based and sampling-based algorithms [5, 11]. Optimal node-based algorithms represent the configuration space in cost assigned nodes. The cost can be calculated based on a specified heuristic, and the path to the goal node is planned accordingly. One of the early developed path planning algorithms is the Dijkstra algorithm, which searches for the shortest path based on costs of edges that are previously known. An extension of Dijkstra's algorithm is given by the A* algorithm, which post-calculates the cost towards the initial state and a heuristic estimation of the cost towards the goal state. Due to their relative simplicity on implementation, A* algorithms have been used on multiple platforms including UAVs [10]. Several algorithms branched out of A* to offer more degrees of freedom on the configuration space. For instance, the D* algorithm, also referred to as Dynamic A* in literature, uses sensor data to detect dynamic obstacles in the environment, hence adding a temporal element in the configuration space [5, 11]. The D* algorithm then reallocates the cost distribution in the space and accordingly modifies the path to take into account dynamic obstacle avoidance. Other extensions to the A* planners were developed to solve for smoother optimal paths such as the anytime A* or phi* algorithms. Phi* checks for a line of sight between every new node and its parent node of the current node; in the absence of any obstacles between the two nodes, the current node is removed from the path and a straight line is extended between the parent and the new node. By repeating this algorithm along the path, the solution converges to a more optimal and smoother version of the A* solution [5, 11].

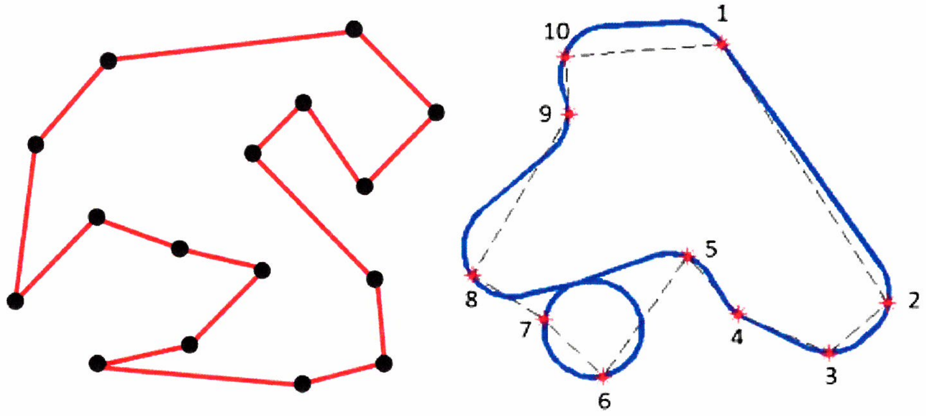
Sampling-based algorithms require prior knowledge of the workspace, as they involve representing the environment or the workspace as a set of nodes where the path is built by tracing it from one node to the other. The node selection is often performed through a random search until a feasible desired path is found. One of the traditional sampling-based methods is the Rapidly Exploring Random Trees (RRT), which works as follows. The path segment propagating from node to node in the tree is traced by taking the 3D Dubins path between the two nodes. After defining an initial and goal node/region in a configuration space, the tree starts exploring the search space by choosing a point at random, and if the selected node is obstacle-free, a line segment of specified length is extended from the node in the tree closest towards the selected node. The algorithm repeats until the tree reaches the goal node or region [5, 11, 23, 24]. The main drawback of this method is that it does not guarantee convergence to an optimal or shortest path. Despite that, RRT was successfully leveraged in [23] to design a path planner for an UAV to achieve both dynamic and static obstacle avoidance based on RRT. However, due to its high computational efficiency, RRT has remained an active area of research, specifically in solving for a near optimal solution. Several modified RRT algorithms have been developed to solve for near-optimal solutions [25, 24, 26]. RRT* provides a modified version of RRT to guarantee asymptotic convergence to an optimal solution. The algorithm selects a random sample in the free space and extends a path from the closest node on the generated tree to the sample. If the path is obstacle-free, the algorithm compares the cost of reaching the sample from a set of potential parent nodes on the tree within the vicinity of the random sample [26]. Other modifications can be on the definition of the configuration space; for instance, in the TRRT algorithm suggested in [27], a cost function is integrated in the configuration space prioritizes downhill over uphill movement for a surface vehicle [24].

While traditional path planning algorithms have shown excellent results for finding obstacle-free feasible paths to connect two configuration states in an environment, further measures must be taken in the context of surveillance missions and search tasks. In such applications, it is desirable to solve for a path that visits a set of predefined sites or regions of interest in a given map. In this study, for circular regions of equal radii, a Dubins vehicle with a predefined configuration space is utilized to plan the trajectory of the shortest path maneuver along a path sequence. The problem is hence extended from a TSP to a DTSPN.

2.3 The Travelling Salesman Problem

As mentioned earlier, traditional path planning algorithms have desirable performance in finding an obstacle-free path connecting two configurations, limitations arise in surveillance missions where further constraints on the mission and the path are imposed. For a given map with a defined set of sites of interest, the problem transforms into solving for the path that visits all of these sites. The TSP has been utilized to model and yield time-optimal solutions to visiting known locations in a map [28, 16, 29, 30]. Typically, the motion models are trivial, zero dynamics, where the edge costs are linked to the Euclidean distance between two sites. The TSP can even handle asymmetric costs [28], nonetheless, integrating the cost with the motion model has not been fully explored.

Hints of one such integration of the edge-cost and the motion model can be seen in the DTSPN in [14]. This problem branches out from the standard TSP and the Dubins Travelling Salesman Problem (DTSP). In TSP, the shortest



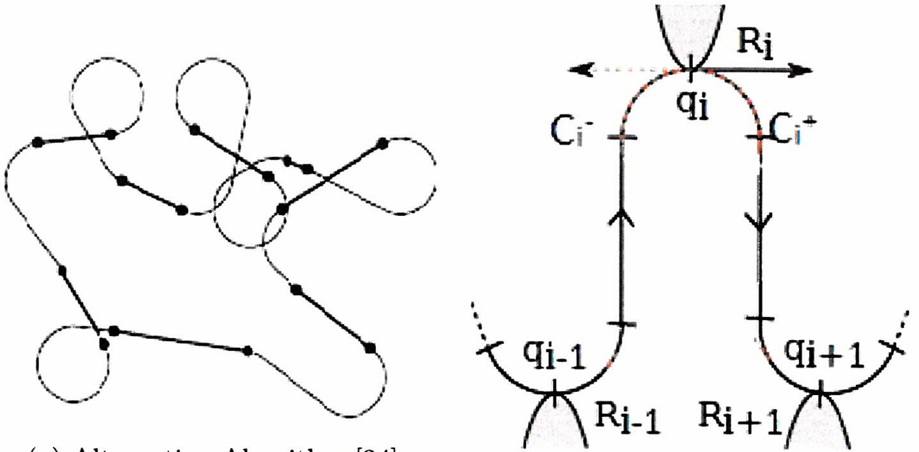
(a) TSP

(b) DTSP

Figure 2.3: TSP vs DTSP

path to visit all waypoints is a series of straight lines connecting each pair of waypoints along the path (Figure 2.3a). On the other hand, the DTSP considers the Dubins maneuvers to connect every pair of waypoints. In fact, the DTSP corresponds to the TSP with an additional minimum turning radius condition of the Dubins vehicle, demonstrated in Figure 2.3b. DTSPN becomes an extension to DTSP where regions or neighborhoods, modelled as disks instead of points, are to be visited along the path. Other variants of the DTSPN arise when further constraints are imposed on the maneuver around the region. Such constraints may be imposed by the dynamics of the vehicle [31], or the nature of the task deliverable [32]. Note that for the DTSPN, the headings of the vehicle along the path are not known a priori, which results in adding infinite possibilities that further complicate the problem. As a result, several approaches exist in the literature that can be categorized into three classes: decoupled approach, transformation methods, and the evolutionary approach.

In the decoupled approach, the mission or the path sequence is determined independently of the headings of the path. The mission is often solved for using a polynomial time approximation scheme for the Euclidean Traveling Salesman Problem (ETSP) presented in [33]. One of the most common algorithms in this approach is the Alternating Algorithm (AA), demonstrated in Figure 2.4a [34], [35]. The sequence of visits is determined by the solution of the ETSP and the odd segments of the solution are replaced with the shortest Dubins maneuver connecting a pair of straight line segments. One of the variants of the AA approach is the receding horizon algorithm, which determines the heading by assigning a look-ahead distance between a pair of points along the sequence [36]. An alternative approach is proposed in [13] where a local optimization technique is employed to determine the waypoints and the headings to the goal, also motivated by the work done in [37] in formulating the convex optimization problem to DTSPN for



(a) Alternating Algorithm [34].

(b) Convex optimization solution to DTSPN [13].

Figure 2.4: Decoupled approach

free initial and final headings at every local maneuver. However, the solution only visits regions of interest at a point on the boundary as shown in Figure 2.4b.

Transformation methods, on the other hand, take assumptions on the headings and compute the length of the shortest maneuver between all pairs of regions [38]. The maneuvers complete the graph that demonstrates the problem and the graph is fed to an Asymmetric Travelling Salesman Problem (ATSP) solver. Genetic algorithm and randomized sampling-based resolution complete approach are proposed by [28] and [39], respectively, to solve the DTSPN. The latter transforms DTSPN into ATSP that is solved in [40] by LKH algorithm.

The shortcoming of the more traditional algorithms, such as the AA and transformation methods, is that they can only guarantee multiplicative factor approximation of the optimal solution. As a result, evolutionary optimization to generate machine learning through automated discovery was proposed to solve for the DTSPN [3], [41]. In the memetic algorithm [41], for instance, the authors propose boundary-based encoding scheme to determine the visiting points of every neighborhood, and an evolutionary algorithm to derive the optimal Dubins tour.

The approach in [13] provides a solution that competes with the results from the evolutionary approach with relatively less computational expense. However, the solution only visits the regions of interest at one point on the boundary before heading towards the next region along the sequence, hence neglecting the exploration factor imposed in the problem of having to traverse a non-zero arc around every region. This study, just like in [13], builds on the definition of shortest local and global Dubins maneuver used in formulating the convex optimization problem proposed and solved in [37]. Furthermore, the approach leverages Dubins' nature of straight lines and curves to abide by the exploration constraint on the path. The initial and final headings for every local maneuver along the sequence is based on the positions of the initial, final, and departure regions with respect to eachother. Dubins problem is then extended by redefining the initial and final configuration of the maneuver, where each configuration is defined by 2D Euclidean coordinates of the region center and the vehicle heading on the boundary of the region with radius r_{region}. The defined parameterization of Dubins CSC maneuvers is then used to derive a geometrical representation of the local minima. By taking the appropriate assumptions on the headings for the local Dubins maneuver between every pair of regions along the path sequence, the number of possible paths is reduced to one optimal solution by imposing a condition on the local final heading angle θ_f corresponding to two different tangential positions, clockwise and counter-clockwise, around the region (disk). This expands the solution of [13] to solve for the constrained path problem addressed in this work. The proposed algorithm is validated by the classical brute force approach, where all possible paths for every sequence of visits is analyzed, and follows a similar approach [15].

Chapter 3

Problem Formulation

Since the problem is modeled as a DTSPN, a first-order unicycle model, also referred to as a Dubins car model given by equation (2.1), is adopted as a vehicle model. Note, however, that this model is used to plan a trajectory that traverses around the regions of interest and does not necessarily represent the model of the UAV itself. The classical Dubins problem, however, is extended by redefining the initial and final configuration for each local maneuver; the configuration, q, is defined for every region as $(x, y, \theta) \in SE(2)$, where (x, y) denote the position of the center of the region in $\mathbb{R}^2$ and the heading angle of the vehicle is denoted by $\theta \in \mathbb{S}^1$. In order to complete the definition of the problem tackled in this study, several assumptions are considered on the regions of interest and the dynamics of the maneuver. Consider a map of mutually exclusive regions modeled as disks with radius, r_{region}. Furthermore, assume that every pair of regions along a path sequences are separated by a distance greater than $4r_{region}$. For every sequence of visits, the path is expected to traverse a non-zero arc around every region. As a result, the local maneuver between every pair of regions can be considered a Dubins maneuver with a minimum turning radius equal to r_{region}. The objective becomes minimizing the global Dubins maneuver along the path sequence.

The optimization problem, for n regions of interest, can now be formulated as:

$$\Gamma = \Gamma(q_n, q_1) + \sum_{i=1}^{n-1} \Gamma(q_i, q_{i+1}), \tag{3.1}$$

where Γ is the sum of lengths of the local Dubins maneuvers along the path sequence (between every pair of regions). Minimizing equation (3.1) can be achieved by solving for the optimal configurations, q, that correspond to the optimal heading for each local Dubins maneuver, $\Theta = [\theta_1, \theta_2, .., \theta n]$. Note that the configuration of the end points of the maneuver, q's, belong to a set of infinite possibilities. In addition, the continuity of the global path depends on the direction of the headings, θ in q, about the region. Hence, the problem at hand becomes setting the appropriate heading at initial and final regions, and accord-

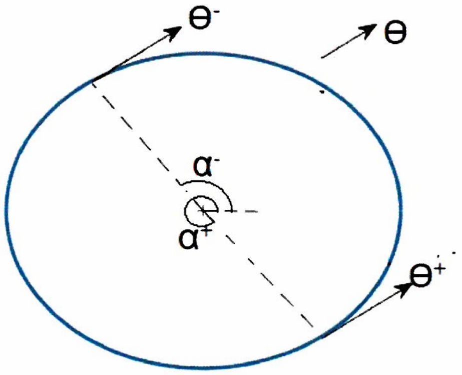


Figure 3.1: Heading configuration

ingly solving for the global optimal path while taking into consideration the path continuity.

3.1 Defining the Initial and Final Configurations

In a 2D Euclidean space, each region is modeled as a circle or a disk with a radius r_{region}, 2D position of its centers, and the heading angle, θ. The configuration of the vehicle around each region can be redefined as follows:

$$q = [x_{center}, y_{center}, \theta]. \tag{3.2}$$

If θ is the heading of the vehicle on the boundary of the region, then the angular position of the vehicle about the center of the region, α, can be calculated, depending on the direction of the turn about the region, as follows:

$$\alpha^{+} = \theta - \frac{\pi}{2}, \tag{3.3}$$

$$\alpha^{-} = \theta + \frac{\pi}{2}, \tag{3.4}$$

with (+) denoting the left or counter-clockwise direction and (-) denoting the right or clockwise direction (Figure 3.1). α and θ are used to calculate the position of the vehicle along boundary of the target region in the global coordinate frame, (x, y), as follow:

$$x = x_{region} + r_{region} \cos \alpha, \tag{3.5}$$

$$y = y_{region} + r_{region} \sin \alpha. \tag{3.6}$$

It is assumed that all regions are of the same radius, r_{region}, and the distance between every pair of regions satisfies D_4 conditions; the distance between every pair of regions is greater than $4r_{region}$. The candidate Dubins maneuver between every pair of regions can then, by elimination, be limited to the Dubins CSC maneuvers; RSR, RSL, LSL, LSR. Each of the maneuvers can take the vehicle from one configuration state around an initial region to a desired configuration state around the target or final region in the local maneuver. Consider a LSL maneuver between a pair of regions to take the vehicle to a position of angle, θ_f, at the final region. This heading can be achieved at the counter-clockwise and clockwise direction around the final region (Figure 3.2). But since the interest of the path is to explore every visited region, the path candidates are narrowed down to the path segments that travel around the region and hence the constraints on the direction. Figure 3.3 shows the possible CSC Dubins paths between a pair of regions of equal radius, r_{region}, subject to the direction constraint.

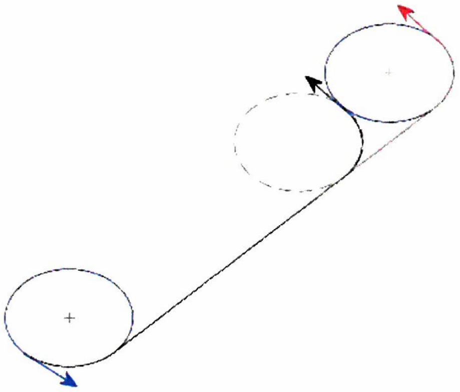


Figure 3.2: The LSL Dubins paths to take the vehicle to a desired direction around the target region (black in the clockwise direction and red in the counter-clockwise direction).

As mentioned in the previous section, the problem can be divided into two sub-problems; mission planning (sequence of visits) and trajectory planning (maneuver along the sequence of visits). In the approach proposed in this study, for every path sequence, the local CSC Dubins maneuver between every pair of regions is calculated by defining an assumption on the headings based on the angle, β; the angle the line connecting the centers of the initial, current, region and the final region, along the sequence, makes with the horizontal. In other words, β represents the locations of a pair of consecutive regions along a sequence with respect to eachother in the global frame. Consider Figure 3.4a with three regions, C_i, C_f, and C_d, along the sequence where the initial configuration for the local Dubins maneuver from every region defined as, $q_i \in C_i$, $q_f \in C_f$, and $q_d \in C_d$, with the corresponding headings assumptions defined as follows:

$$\theta_i = \beta_i, \tag{3.7}$$

$$\theta_f = \beta_f, \tag{3.8}$$

$$\theta_d = \beta_d. \tag{3.9}$$

β_i denotes the position of the initial region with respect to the final region, while β_f denotes the position of the departure region with respect to the final region. As in Figure 3.4a, if the vehicle follows a RSL Dubins maneuver from C_i to C_f, following the precedent assumption on the headings, the problem is refined to solving for the angle α_d that corresponds to θ_d heading at the optimal orientation about the center of the region C_d; clockwise or counter clockwise orientation. Due to the path continuity limitation, the paths RSL and RSR are eliminated (since the vehicle arrives at C_f in the L, counter-clockwise, direction) and the path candidates become LSL and LSR.

Hence, the problem can be expanded as follows. For a given map with n regions of interest, each modeled as a circle or a disk, C, with equal radius, r_{region}, and assuming that the desired path to visit all regions starts and ends at the same region, it is sought after to find the sequence that corresponds to the shortest path to visit all regions. In addition, since the desired path along the sequence is constrained by the condition imposed in exploring the regions, traversing a non-zero arc around every region, while taking into account the path continuity limitation, solving for the optimal maneuver along that path sequence becomes a necessity.The brute force approach is considered to tackle this problem where the length of every path for every visiting sequence is calculated to find the shortest global path corresponding to the shortest visiting sequence.

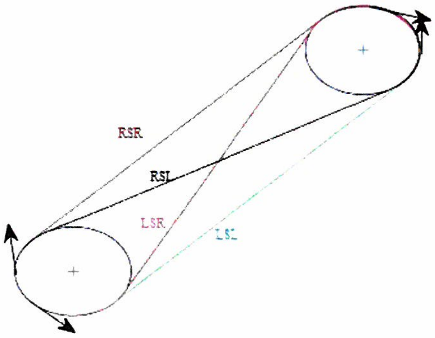


Figure 3.3: CSC Dubins paths between a pair of regions of equal radius, r_{region}

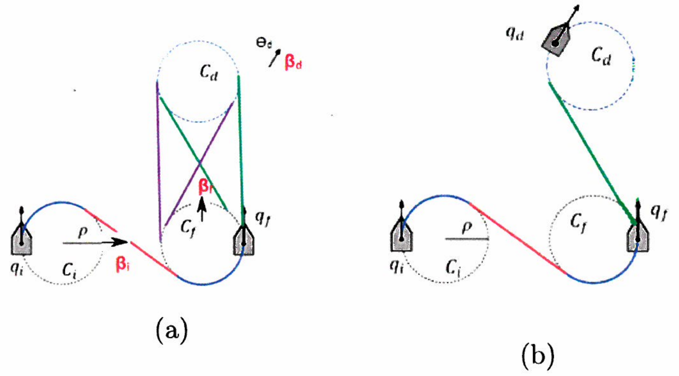


Figure 3.4: Assumptions are taken on the headings based on the locations of a pair of consecutive regions along a sequence with respect to each other

3.2 Brute Force Approach

If the total number of regions is n and all paths start and end at the same region (last region along every sequence is the starting region along a sequence), then the total number of path sequences, s, is:

$$s = n! \tag{3.10}$$

Assume that the map satisfies D_4 condition and the assumption on the headings of the local maneuver discussed in the previous section holds. In this case, the candidate Dubins maneuvers between a pair of regions are; LSL, RSL, RSR, and LSR (Figure 3.3). However for a path traversing along a sequence, the local path between every pair of regions further restricts the Dubins candidates as shown in Figure 3.5. In other words, the continuity in the overall path limits the path candidates on every local maneuver between a pair of regions. Hence, for n regions, the total number of paths per sequence, p, is given as:

$$p_i = 2^{n+1}. \tag{3.11}$$

Note that the increment in the exponent of equation (3.11) arises from the condition on starting and ending the visiting sequence at the same region. The approach calculates every path by summing the length of the Dubins maneuvers between every pair of regions along every possible visiting sequence. From equations (3.10) and (3.11) the total number of paths, T, for a given map of n regions of interest can be calculated as follows:

$$T = n!(2^{n+1}). \tag{3.12}$$

Consider a map with four regions of interest, each with radius $r_{region} = 4$, with the 2D coordinates of the center of every region, C, given as:

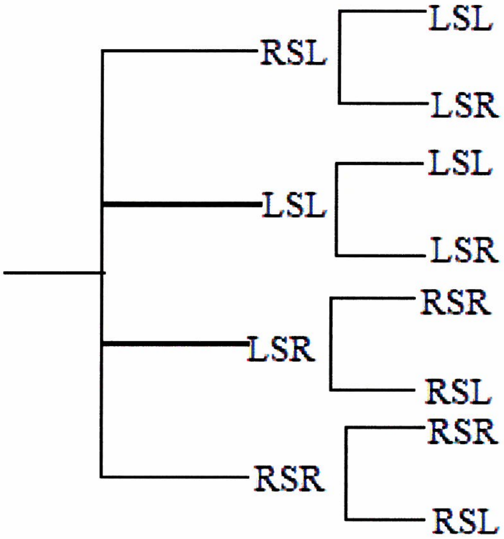
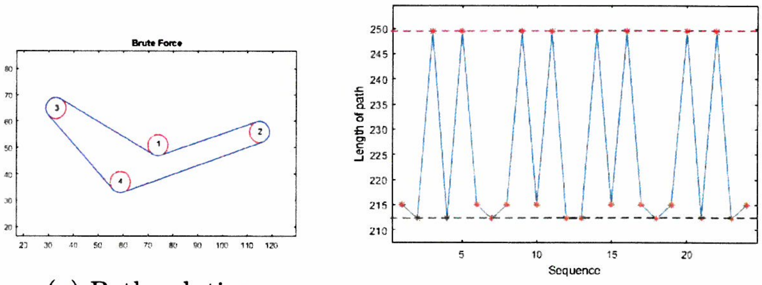


Figure 3.5: Path continuity along a sequence of visits

$$C = \begin{pmatrix} 74,51 \\ 115,56 \\ 33,65 \\ 59,37 \end{pmatrix}. \tag{3.13}$$

For a map of four regions, there exist 24 visiting sequence and 32 paths per sequence and, hence, 768 global path candidates. The length of every path for every path sequence is calculated and the shortest path along every sequence is then extracted, demonstrated in Figure 3.6a (also in Table 3.1). The shortest path is found to belong to a family of combinations of visit sequence and path maneuver (Figure 3.6b). This hints that the final solution can be solved for by eliminating possible global path candidates. For instance, every visit sequence belonging to the same visiting loop converges to the same solution; visiting sequences $[4, 3, 1, 2]$ and $[4, 2, 1, 3]$ both belong to the same visiting loop and hence converge to the same solution (Table 3.1).

While it is guaranteed that this approach obtains the optimal solution, solving for every path becomes computationally unfeasible in more populated or dense maps. As a result, a smarter and more computationally feasible approach is required to tackle the problem addressed in this study. Since the objective is to minimize (3.1) using the appropriate assumption on the headings and the corresponding optimal direction around the regions of interest, the Dubins maneuver for a pair of regions is parameterized to geometrically find the local minima head-



(a) Path solution

(b) Plot of the shortest path for every sequence

Figure 3.6: Brute force: Simulation results

ing and solve for the optimal solution. The local minima heading is used to define an optimization condition based on the assumed headings to solve for the local maneuver, between every pair of regions along the sequence, corresponding to the global shortest path.

Sequence	Shortest path	Length of shortest path
4 3 2 1	’RSRRSRRSLLSR’	215.0513
4 3 1 2	’RSRRSLLSRRSR’	212.2274
4 2 3 1	’LSLLSLLSRRSL’	249.4172
4 2 1 3	’LSLLSRRSLLSL’	212.2274
4 1 3 2	’RSLLSRRSRRSR’	249.4172
4 1 2 3	’LSRRSLLSLLSL’	215.0513
3 4 2 1	’LSLLSLLSRRSL’	212.2274
3 4 1 2	’LSLLSRRSLLSL’	215.0513
3 2 4 1	’RSRRSRRSLLSR’	249.4172
3 2 1 4	’RSRRSLLSRRSR’	215.0513
3 1 4 2	’LSRRSLLSLLSL’	249.4172
3 1 2 4	’RSLLSRRSRRSR’	212.2274
2 4 3 1	’RSRRSRRSLLSR’	212.2274
2 4 1 3	’RSRRSLLSRRSR’	249.4172
2 3 4 1	’LSLLSLLSRRSL’	215.0513
2 3 1 4	’LSLLSRRSLLSL’	249.4172
2 1 4 3	’RSLLSRRSRRSR’	215.0513
2 1 3 4	’LSRRSLLSLLSL’	212.2274
1 4 3 2	’LSRRSRRSRRSL’	215.0513
1 4 2 3	’RSLLSLLSLLSR’	249.4172
1 3 4 2	’RSLLSLLSLLSR’	212.2274
1 3 2 4	’LSRRSRRSRRSL’	249.4172
1 2 4 3	’LSRRSRRSRRSL’	212.2274
1 2 3 4	’RSLLSLLSLLSR’	215.0513

Table 3.1: Shortest path for every sequence

Chapter 4

Proposed Algorithm for DTSPN

In a given map with defined regions of interests modeled as disks of equal radius, r_{region}, it is desired to find the shortest path to visit all regions while traversing a non-zero arc around every region. As such, the problem is modeled as a DTSPN. While the brute force approach solves for the optimal solution, due to its high computational inefficiency, it is rendered feasible only for low populated maps. The convex optimization approach in [13] guarantees convergence to the optimal solution with high computational efficiency for densely populated maps. The solution, however, only visits the regions at one point on the boundary; corresponding to the local minima position and heading, hence it does not factor in the exploration factor in the constrained DTSPN proposed in this study. In this chapter, the Dubins CSC paths are parameterized using the initial and final configurations defined in the problem formulation. Based on this parameterization, the optimization problem is redefined by expressing the difference between the length of candidate local maneuvers in terms of the initial and final configurations. An expression of the local minima is then derived upon which an algorithm is proposed to solve for the constrained DTSPN presented in this study [15].

4.1 Dubins Path Parameterization

Notation	Definition
L_C	distance between the centers of a pair of regions.
L_s	length of the straight path segment.
γ_1	arc segment angle around the initial region.
γ_2	arc segment angle around the target region.
β	angle the line L_C makes with the horizontal axis.

Table 4.1: Notation

Using the assumptions taken in formulating the problem, it is sought after to derive a geometrical representation of the local minima heading to define a decision making criteria on the local maneuver. Dubins CSC paths are parameterized in this chapter to build up for that purpose. As per the problem formulation, unlike the case with the classical Dubins curves, the initial and final configurations are defined by the 2D euclidean coordinates of the center, and the heading angle on the boundary, of the corresponding region. Since it is assumed that all regions are at distances greater than $4r_{region}$ from eachother, only the Dubins CSC paths are considered.

4.1.1 RSR Path

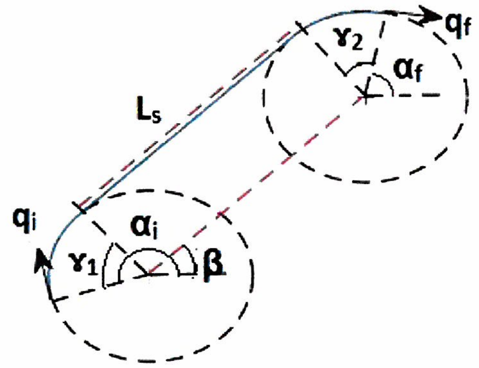


Figure 4.1: RSR Path

Let the initial region with radius r, be centered at $p_i = (x_i, y_i)$ with the initial heading angle θ_i, then the path configuration at the initial point can be defined as $q_i = [p_i, \theta_i]$ where $q \in \mathbb{R}^2 \times S^1$. Similarly, the path configuration at the final region is $q_f = [p_f, \theta_f]$. If the angle of the initial and final curved path segments are denoted as γ_1 and γ_2 respectively, and the straight line segment as L_s (Figure 4.1), then the length of the Dubins maneuver is $L = L_s + r(\gamma_1 + \gamma_2)$. The parameters can now be derived in terms of the line connecting the two regions, L_c, and the angle that it makes with the horizontal, β, as follows:

$$L_s = L_c, \tag{4.1}$$

$$\gamma_1 = mod(\theta_i - \beta, 2\pi), \tag{4.2}$$

$$\gamma_2 = mod(\beta - \theta_f, 2\pi), \tag{4.3}$$

where

$$\beta = mod(\tan^{-1}(\frac{y_f - y_i}{x_f - x_i}), 2\pi), \tag{4.4}$$

$$L_c = ((y_f - y_i)^2 + (x_f - x_i)^2)^{\frac{1}{2}}. \tag{4.5}$$

Since the directions of the path about both regions are in the clockwise, right direction, then the angular positions of the initial and final point are $\alpha_i = \theta_i + \frac{\pi}{2}$ and $\alpha_f = \theta_f + \frac{\pi}{2}$, respectively.

4.1.2 RSL Path

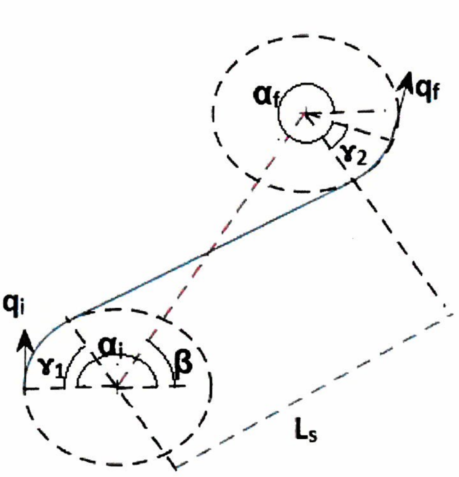


Figure 4.2: RSL Path

Similarly, for the RSL parameterization, the initial point on the path at the initial circle is defined as q_i and the final point as q_f. However, since the final point is in the counter-clockwise, left direction, the angular position of the final point about the final circle is $\alpha_f = \theta_f - \frac{\pi}{2}$.

For L_c and β as previously defined, the length of the straight line segment of the RSL path can be derived as follows:

$$L_S = (L_C^2 - 4r^2)^{\frac{1}{2}}. \tag{4.6}$$

Equation (4.6) is derived by leveraging the fact that the straight line segment connects two tangent points, thus forming two right-angled triangles with L_c as their hypotenuse.

In addition, with μ_1 and μ_2 defined as shown in Figure 4.2, the arc angles, γ_1 and γ_2, are expressed as:

$$\mu_2 = mod(\cos^{-1}(\frac{L_S}{L_C}), 2\pi), \tag{4.7}$$

$$\mu_1 = mod(\beta - \mu_2, 2\pi), \tag{4.8}$$

$$\gamma_1 = mod(\theta_i - \mu_1, 2\pi), \tag{4.9}$$

$$\gamma_2 = mod(\theta_f - \mu_1, 2\pi). \tag{4.10}$$

4.1.3 LSL Path

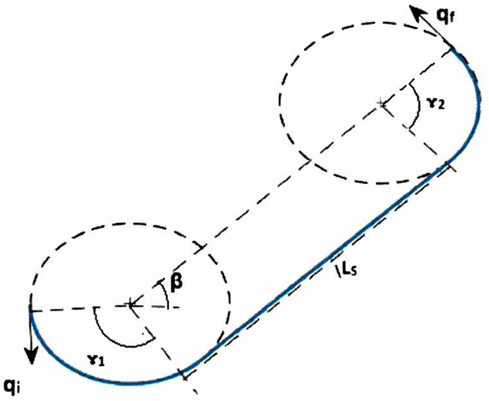


Figure 4.3: LSL Path

Let the initial region with radius r, be centered at $p_i = (x_i, y_i)$ with the initial heading angle θ_i, then the path configuration at the initial point can be defined as $q_i = [p_i, \theta_i]$ where $q \in \mathbb{R}^2 \times S^1$. Similarly, the path configuration at the final region is $q_f = [p_f, \theta_f]$. If the angle of the initial and final curved path segments are denoted as γ_1 and γ_2 respectively, and the straight line segment as L_s (Figure 4.1), then the length of the Dubins maneuver is $L = L_s + r(\gamma_1 + \gamma_2)$. The parameters can now be derived in terms of the line connecting the two regions, L_c, and the angle that it makes with the horizontal, β, as follows:

$$L_s = L_c, \tag{4.11}$$

$$\gamma_1 = mod(\beta - \theta_i, 2\pi), \tag{4.12}$$

$$\gamma_2 = mod(\theta_f - \beta, 2\pi), \tag{4.13}$$

where

$$\beta = mod(\tan^{-1}(\frac{y_f - y_i}{x_f - x_i}), 2\pi), \tag{4.14}$$

$$L_S = L_c = ((y_f - y_i)^2 + (x_f - x_i)^2)^{\frac{1}{2}}. \tag{4.15}$$

Since the directions of the path about both regions are in the counter-clockwise, left direction, then the angular positions of the initial and final point are $\alpha_i = \theta_i + \frac{\pi}{2}$ and $\alpha_f = \theta_f - \frac{\pi}{2}$, respectively.

4.1.4 LSR Path

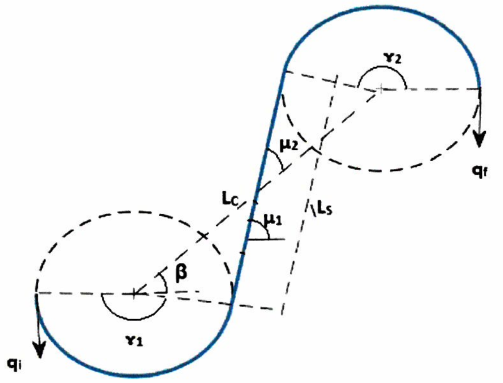


Figure 4.4: LSR Path

Similarly, for the LSR parameterization, the initial point on the path at the initial region is defined as q_i and the final point as q_f. However, since the final point is in the clockwise, right direction, the angular position of the final point about the final region is $\alpha_f = \theta_f + \frac{\pi}{2}$.

For L_c and β as previously defined, the length of the straight line segment of the RSL path can be derived as follows:

$$L_S = (L_C^2 - 4r^2)^{\frac{1}{2}}. \tag{4.16}$$

Equation (4.6) is derived by leveraging the fact that the straight line segment connects two tangent points, thus forming two right-angled triangles with L_c as their hypotenuse.

In addition, with μ_1 and μ_2 defined as shown in Figure 4.4, the arc angles, γ_1 and γ_2, are expressed as:

$$\mu_2 = mod(\cos^{-1}(\frac{L_S}{L_C}), 2\pi), \tag{4.17}$$

$$\mu_1 = mod(\beta - \mu_2, 2\pi), \tag{4.18}$$

$$\gamma_1 = mod(\mu_1 - \theta_i, 2\pi), \tag{4.19}$$

$$\gamma_2 = mod(\mu_1 - \theta_f, 2\pi). \tag{4.20}$$

Given an initial and final configuration, the presented parameterization of Dubins paths is used to reformulate the optimization problem to solve for the local minima heading of the difference in the length of the candidate Dubins maneuvers. As such, the shortest global maneuver is obtained by solving for the local minima headings for every region based on its position along the path sequence.

4.2 Geometrical Representation of the Local Minima

Considering the brute force approach to DTSPN, and given a map with n regions of interest, the total number of paths is equal to $n!(2^{n+1})$. The optimal path sequence is solved for using the polynomial time approximation scheme to the ETSP in [33]. However, there exists an optimal global maneuver along the sequence among the 2^{n+1} possible maneuvers (also proven in [42]). This can be demonstrated by considering the Dubins maneuver candidates between a pair of regions, where the start and end do not belong to the initial and final circles, respectively, as shown in Figure 4.5. At least two maneuvers, LSL and RSL, can be eliminated since they result in arc lengths greater than π, which hints that the optimal maneuver can be solved for via elimination.

Consider the RSR and RSL paths presented in Figure 4.6, that correspond to the local maneuvers between two regions with the initial region in the clockwise orientation. Using the path parameterization presented in the previous section, the difference between L_{RSR} and L_{RSL} can be expressed as:

$$(\triangle L)_{RSR-RSL} = A - 2r_{region}\theta_f, \tag{4.21}$$

where

$$A = \frac{L_c - L_s - 2r_{region}\cos^{-1}(\frac{L_S}{L_C})}{2r_{region}}. \tag{4.22}$$

Note that the expression in equation (4.21) does not depend on θ_i, since both maneuvers follow the same path between any arbitrary θ_i to $\theta = \beta$ on the initial circle, where β is the angle that the line connecting the initial region to the

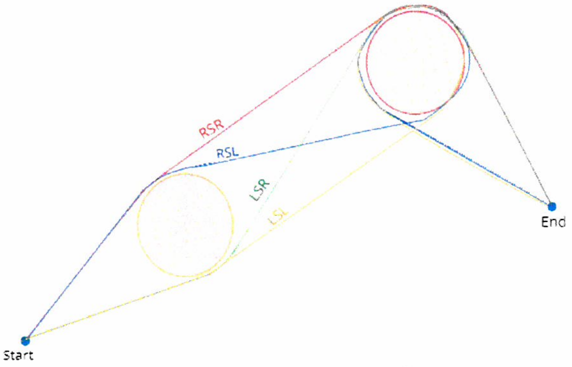


Figure 4.5: Local Dubins maneuver candidates between a pair of regions with the start and end points not belonging to the initial and final circles, respectively. Note that the paths are not overlapping for visual illustration purposes only.

final region makes with the horizontal. Consider the case when $(\triangle L) = 0$, then $(\theta_f)_{critical}$ can be defined as follows starting from equation (4.21):

$$(\theta_f)_{critical} = mod(\frac{A}{2r_{region}} + \beta, 2\pi). \tag{4.23}$$

This heading, however, corresponds to two tangential points on the boundary of the final region in the local maneuver; clockwise (denoted by superscript '+') and counterclockwise (denoted by a superscript '-') direction, demonstrated in Figure 4.7. The angular positions of both points about the final region can now be expressed as:

$$(\alpha_{f,critical})^+ = mod((\theta_{f,critical})^+ - \frac{\pi}{2}, 2\pi), \tag{4.24}$$

$$(\alpha_{f,critical})^- = mod((\theta_{f,critical})^- + \frac{\pi}{2}, 2\pi). \tag{4.25}$$

A conditional statement on the maneuver can now be formulated as follows: given a pair of regions with a heading angle θ_f, the angular position about the center of the region as, $\alpha_f^+ = \theta_f - \frac{\pi}{2}$ (Figure 4.8).

- If $(\alpha_{f,critical})^+ \leq \alpha_f^+ \leq mod((\alpha_{f,critical})^+ + \pi, 2\pi)$, then the optimal maneuver is RSL.

- If $(\alpha_{f,critical})^+ > \alpha_f^+ > mod((\alpha_{f,critical})^+ + \pi, 2\pi)$, then the optimal maneuver is RSR.

The same approach holds for LSL and LSR paths being symmetrical to RSR and RSL paths, respectively, with $(\theta_f)_{critical}$ defined as:

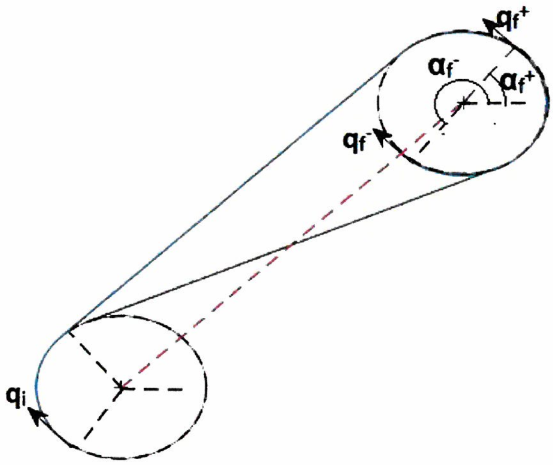


Figure 4.6: RSR versus RSL maneuver

$$(\theta_f)_{critical} = mod(\frac{-A}{2r_{region}} + \beta, 2\pi). \tag{4.26}$$

Similarly, the condition on the maneuver imposed on θ_f is established by defining $(\alpha_{f,critical})^+$ as in equation (4.24):

- If $(\alpha_{f,critical})^+ < \alpha_f^+ < mod((\alpha_{f,critical})^+ + \pi, 2\pi)$, then the optimal maneuver is LSL.
- If $(\alpha_{f,critical})^+ \geq \alpha_f^+ \geq mod((\alpha_{f,critical})^+ + \pi, 2\pi)$, then the optimal maneuver is LSR.

4.3 Proposed Algorithm

After establishing the local optimization criteria, the proposed algorithm is presented in Algorithm 1. The algorithm solves for the shortest global maneuver in

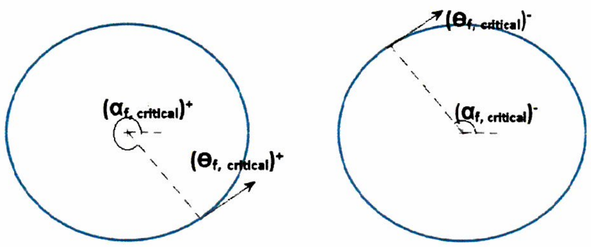


Figure 4.7: Heading orientation

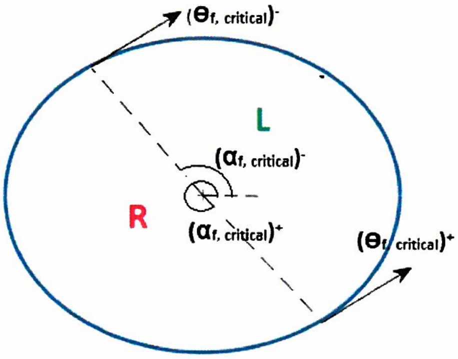


Figure 4.8: Direction

a given map of regions of interest from a given starting point on the map. The shortest visit sequence is solved for using the ETSP solution [33]. The headings ($Heading()$) on every region correspond to the initial point on the local Dubins maneuver; the angle, β, that the line connecting the current region with the next region makes with the horizontal axis. Hence, for the j-th pair of regions along the sequence, $\theta_{initial} = \beta_j$ and $\theta_{final} = \beta_{j+1}$.

```
Result: path
map = initialize();
sequence = ETSP(map);
$\Theta$ = Headings(sequence);
for every region along the sequence do
    $\theta_{initial}$ = currentHeading;
    $\theta_{final}$ = nextHeading;
    if first region then
        shortestManeuver($\theta_{initial}$, $\theta_{final}$);
    end
    else
        optimalManeuver($\theta_{initial}$, $\theta_{final}$);
    end
end
rewire();
```

Algorithm 1: Simplified Algorithm

After solving for the path sequence and the corresponding headings, the local path between every pair of consecutive regions along the sequence is solved for using the formulated conditions imposed on θ_{final} defined in the previous section. Initially, the orientation of the first region is chosen based on the shortest Dubins maneuver between the first pair of regions. However, this maneuver is re-routed by the optimal maneuver between the last and first region (rewire()), if the direction of the latter does not agree with the former. Special case arises when regions are aligned on the same line along the path sequence. In Figure 4.9 for instance, the shortest path to visit four regions along the sequence, $[1, 2, 3, 4]$, will result in a tangential zero arc maneuver around region 3. As such, a bais is imposed on the RSL and LSR paths in the inequality of the condition statement to force the global path to traverse a non-zero arc length around aligned regions.

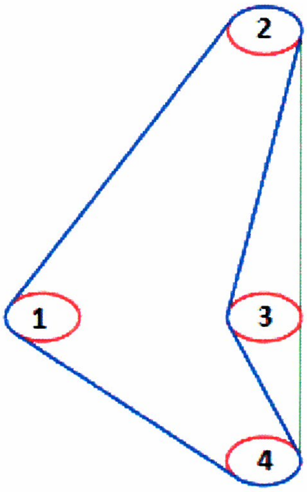


Figure 4.9: A bias in the equality is imposed on RSL and LSR paths

The work in [37] seeks to obtain the unique solution presented in [42] for free initial and final headings at entry and departure respectively. As a result, the path is parameterized and solved for the local minima between a pair of regions satisfying D_4 and $D_{2+2\sqrt{2}}$ conditions, which are regions of unity radius separated by a distance equal to 4 and $2 + 2\sqrt{2}$, respectively. Furthermore, the authors proved that the upper bound of the arc length around every region in a sequence along the global shortest path depends on the distance between the regions and the sharpness of the turn along the euclidean path. Similarly, this is implied in equation (4.23) and equation (4.26) as the heading corresponding to the local minima, $(\theta_f)_{critical}$ is expressed in terms of the length of the line connecting the two regions, L_c, and the angle that line makes with the horizontal, β.

The proposed approach emphasizes on differentiating between two different types of local paths connecting q_i and q_f: shortest and optimal. The shortest local path corresponds to the shortest Dubins maneuver between q_i and q_f. On the other hand, the optimal local path is the Dubins local maneuver between q_i and q_f corresponding to the global optimal path, which is the shortest global path to visit all regions. The implication of this differentiation is that the global shortest path does not necessarily mean that it is also the shortest path locally connecting the initial and final region. This difficulty arises from the fact that any heading around a region corresponds to two directions, and hence two different points around the region. As a result, $(\theta_f)_{critical}$ is expressed in terms of the angular position $(\alpha_f)_{critical}$ in both clockwise and counterclockwise direction to reach the two, equivalent, conditions imposed on the path headings.

As discussed earlier, since the algorithm is motivated by the objective of traversing a non-zero arc around every region and supported by the assumption that the regions are modelled as Dubins circles with radius, r_{region}, the appropriate initial heading in every local maneuver connecting two regions is β, which corresponds to the point around the region from which the two path candidates separate (RSR and RSL if clockwise, and LSL and LSR if counterclockwise). The algorithm is demonstrated in Figure 4.10 for seven regions of interest with unity radius. The path sequence is solved for using ETSP and the proposed algorithm. Because of the sharp turn from region-3 centered at $(27, 23)$ to region-4 centered at $(26, 28)$, with the heading in the initial circle in the counterclockwise direction, the local optimal maneuver is solved by the proposed algorithm to be a LSR maneuver (shown in blue in Figure 4.10). It is noted that despite the fact that the LSL maneuver (shown in red in Figure 4.10) is the shorter maneuver to reach region-4, however it will result in a greater arc length around the region as the path transverses to region-5. In addition, the arc length around region-4 will exceed π in the LSL case, which causes the path to intersect with itself, hence rendering it non-optimal [37]. On the other hand, the LSR maneuver corresponds to the global optimal solution since it will result in the shortest local and global arc length. The algorithm was also simulated at different region radius and greater number of regions (region density) for robustness. Figure 4.11a and Figure 4.11b show simulation results for 40 and 100 regions of interest respectively.

4.4 Discussion

The proposed solution to the constrained DTSPN problem in hand is validated with the brute force solution. Both approaches are implemented on MATLAB and the solution and the computational effort are compared. An existing TSP solver, based on [33], on MATLAB is used to solve for the shortest sequence throughout the analysis. Since the algorithm solves for a unique global path solution, the number of iterations in the "for loop" of Algorithm 1, where the core of the computation occurs, is equal to the number of regions, n. This further implies that the computational efficiency is independent of the radius of the regions, r_{region} and the density of the map, $\frac{n}{D}$ (where D is the grid width of the map). This is also evident following a 'Big-O' notation analysis on the approach proposed in Algorithm 1 where the "for loop" is of order $O(n)$. Similarly, the

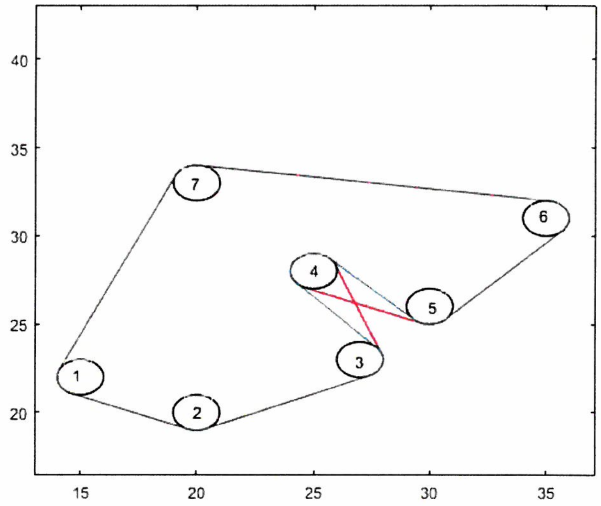
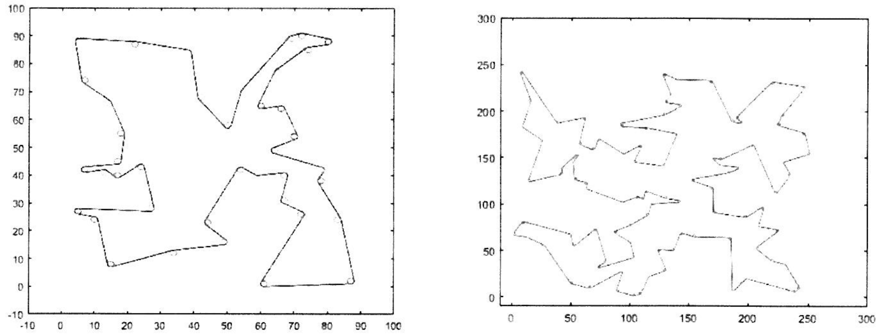


Figure 4.10: Simulation result for seven regions of interest showing the global optimal path (blue) and a local non-optimal path around a sharp turn (red).

computational efficiency of the brute force approach is only dependent on n with a computational complexity of order $O(2^{n+1})$.

For a map with 10 regions of interest, with the center of every region given in equation (4.27), the proposed algorithm is used to solve for the shortest path to visit all regions for different r_{region}. The solution is also compared to that of the brute force solution with the computational time recorded for each simulation (Table 4.2); where t_{PA} and t_{BF} are the simulation time for the proposed approach and the brute force approach respectively. The solution always converges to that of the brute force solution. While the proposed algorithm demonstrates faster performance, the simulation time did not change with the change in r_{region} for both approaches. The proposed algorithm is also tested for maps of different densities. For $r_{region} = 1$, the computational time for maps with different densities is recorded in Table 4.3. Similarly, the computational time remains in the same order for different map densities.



(a) Simulation result for a map with 40 regions of interest.

(b) Simulation result for a map with 100 regions of interest.

Figure 4.11: Simulation Results

$$C = \begin{pmatrix} 9,164 \\ 68,7 \\ 84,56 \\ 193,98 \\ 10,105 \\ 180.198 \\ 35,96 \\ 130,31 \\ 100,55 \\ 116,182 \end{pmatrix}. \tag{4.27}$$

Figure 4.12 demonstrates the results of the computational time taken to converge to the solution for different number of regions. The brute force approach takes significantly longer time to converge to a solution for more than 15 regions (converges to the shortest path solution in 88 seconds for a map of 15 regions). On the other hand, for the same map, the proposed algorithm converged to a solution in less than 0.09 seconds (Figure 4.12). The algorithm also demonstrated efficient performance for highly populated maps. As can be seen in Figure 4.12, the algorithm converged to the shortest path solution for a map of 500 regions in 0.2 seconds. The simulation time for the proposed algorithm is also analyzed for different orders of regions, n (Figure 4.13). The algorithm demonstrates consistent performance, $0.1-0.2$s for n in the 10^3 order of magnitude before it increases to 1.466s for n of order 10^4. Computational shortcomings of the proposed algorithm start to appear for n with order of magnitude of 10_4 and higher; where for for maps with 10_4 the algorithm converged to the solution in 211.5s and consumed more than 12 hours for maps with n of order 10_6. It is also important to reiterate that throughout this study, it is assumed that the regions in the map satisfy D_4

condition; the distance between any pair of regions is greater than $4r_{region}$. As a result, for maps with more closely packed regions, an alternative or modified approach has to be considered.

r_{region}, m	Length of path, m	t_{PA}, in s	t_{BF}, in s	$\frac{t_{PA}-t_{BF}}{t_{BF}} * 100$
0.5	659.84	0.0548	0.1938	-71
1	665.98	0.0552	0.1778	-69
1.5	672.20	0.0734	0.1708	-57
2	678.50	0.051	0.1467	-65
2.5	684.88	0.0476	0.1532	-69
3	691.35	0.0527	0.1898	-72
3.5	697.89	0.0554	0.1754	-68
4	704.51	0.0511	0.1615	-68
4.5	711.22	0.0508	0.1466	-65
5	718.01	0.0604	0.1296	-53
5.5	724.88	0.0569	0.1471	-61
6	731.83	0.0571	0.1586	-64
6.5	738.86	0.0469	0.1356	-65

Table 4.2: Computational time of the proposed algorithm (PA) verus Brute Force (BF) for maps with different r_{region} for $n = 10$ regions.

Density($\frac{n}{D}$)	t_{PA}, in s	t_{BF}, in s	$\frac{t_{PA}-t_{BF}}{t_{BF}} * 100$
0.0278	0.0492	0.1642	-70
0.0313	0.055	0.2331	-76
0.0358	0.048	0.281	-83
0.0417	0.0455	0.1481	-73
0.0500	0.0451	0.1429	-69
0.0625	0.0482	0.1555	-69
0.0833	0.0518	0.1492	-65
0.1250	0.066	0.1732	-62
0.2500	0.0452	0.1488	-70

Table 4.3: Computational time of the proposed algorithm (PA) versus Brute Force (BF) for maps with different densities.

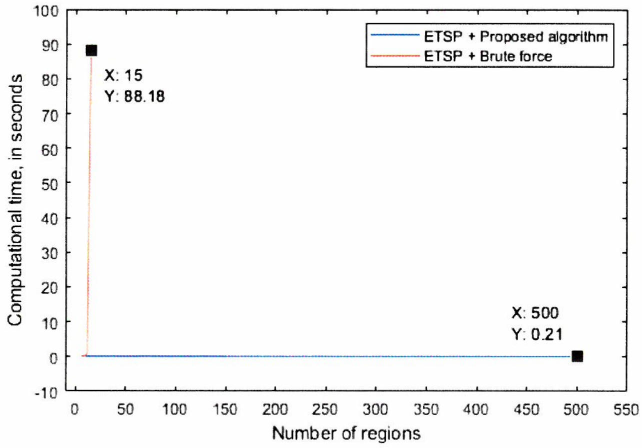


Figure 4.12: Comparison in the computational performance between the brute force solution and the proposed algorithm.

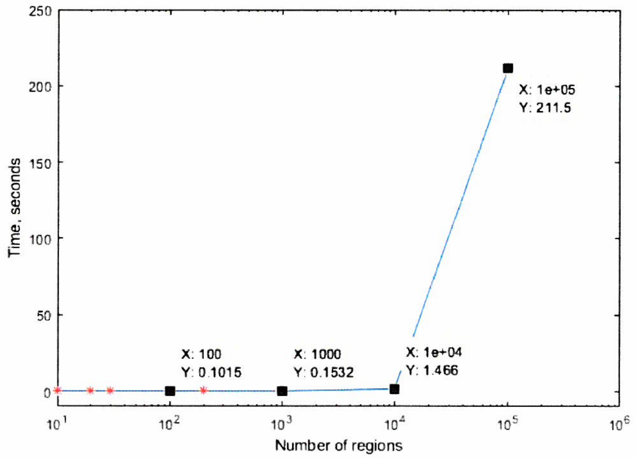


Figure 4.13: Simulation time of the proposed algorithm for different number of regions.

Chapter 5

Extension: Regions with Non-equal Radii

It has been, thus far, assumed that all regions are of equal radius. In certain applications, however, it might be of interest to model these regions each with its corresponding radius. While the proposed approach in the previous chapter will guarantee a feasible solution based on a concatenation of Dubins CSC maneuver, the waypoints tracing the overall global path have to be adjusted to the size of every corresponding region along the sequence. In this chapter, the proposed approach is used to build a path planner that solves for a feasible path to visit and explore regions of non-equal radii. The configuration of each region, q, in equation (3.2) is redefined as follows:

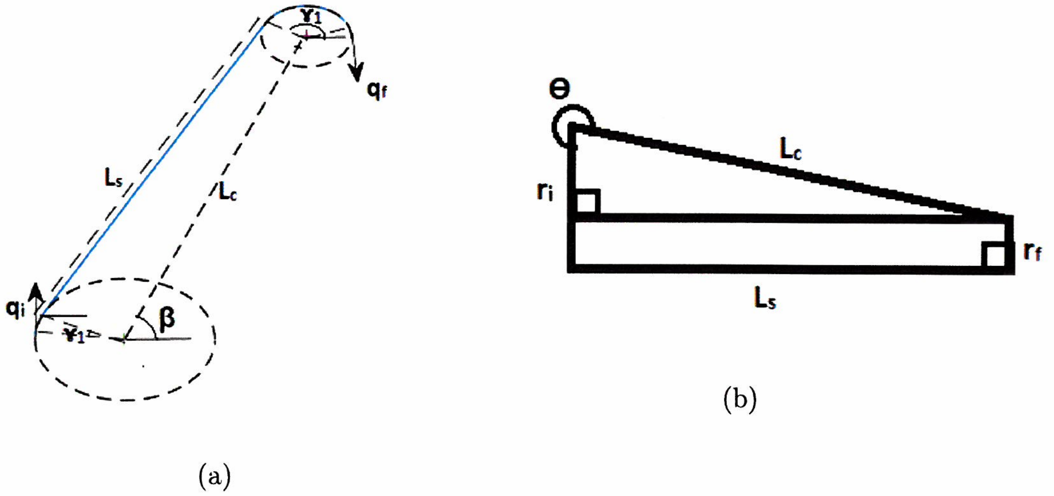
$$q = [x_{center}, y_{center}, \theta, r_{region}]. \tag{5.1}$$

where r_{region} is the radius of the region. Furthermore, it is assumed that the map satisfies D_4 condition with respect to the maximum r_{region}. This modification, however, imposes the necessity to re-parameterize the Dubins CSC paths between a pair of regions of non-equal radius to solve for the tangent points at arrival and departure for the modified path. Accordingly, the headings of the path are redefined based on these tangent points.

5.1 Dubins Path Re-parameterization

5.1.1 RSR Path

Let the initial region with radius r_i, be centered at $p_i = (x_i, y_i)$ with the initial heading angle θ_i, then the path configuration at the initial point can be defined as $q_i = [p_i, \theta_i, r_i]$. Similarly, the path configuration at the final region is $q_f = [p_f, \theta_f, r_f]$. If the angle of the initial and final curved path segments are denoted as γ_1 and γ_2 respectively, and the straight line segment as L_s (Figure 5.1a), then



(a)

(b)

Figure 5.1: RSR Path

the length of the Dubins maneuver is $L = L_s + r_i\gamma_1 + r_f\gamma_2$. The parameters can now be derived in terms of the line connecting the two regions, L_c, and the angle that it makes with the horizontal, β, as follows:

$$L_s = (L_c^2 - (r_i - r_f)^2)^{0.5}, \tag{5.2}$$

$$\theta = mod(\beta - \sin^{-1}(\frac{r_i - r_f}{L_c})), \tag{5.3}$$

$$\gamma_1 = mod(\theta_i - \theta, 2\pi), \tag{5.4}$$

$$\gamma_2 = mod(\theta - \theta_f, 2\pi), \tag{5.5}$$

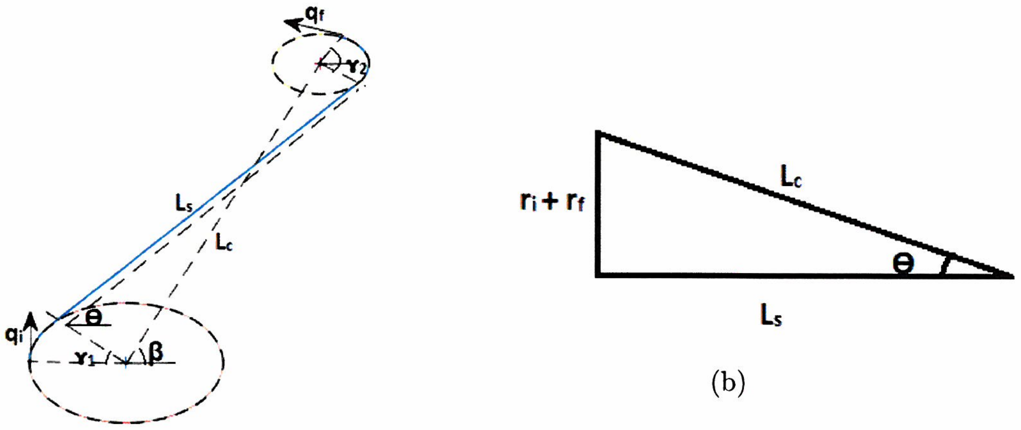
where

$$\beta = mod(\tan^{-1}(\frac{y_f - y_i}{x_f - x_i}), 2\pi), \tag{5.6}$$

$$L_c = ((y_f - y_i)^2 + (x_f - x_i)^2)^{\frac{1}{2}}. \tag{5.7}$$

5.1.2 RSL Path

Similarly, for the RSL parameterization, the initial point on the path at the initial circle is defined as q_i and the final point as q_f. However, since the final point is in the counter-clockwise, left direction, the angular position of the final point about the final circle is $\alpha_f = \theta_f - \frac{\pi}{2}$.



(a)

(b)

Figure 5.2: RSL Path

For L_c and β as previously defined, the length of the straight line segment of the RSL path can be derived as follows:

$$L_s = (L_c^2 - (r_i - r_f)^2)^{0.5}, \tag{5.8}$$

$$\theta = mod(\beta - \sin^{-1}(\frac{r_i - r_f}{L_c})), \tag{5.9}$$

$$\gamma_1 = mod(\theta_i - \theta, 2\pi), \tag{5.10}$$

$$\gamma_2 = mod(\theta - \theta_f, 2\pi), \tag{5.11}$$

$$L_S = (L_C^2 - (r_i + rf)^2)^{\frac{1}{2}}. \tag{5.12}$$

5.2 Path Planner for Maps with Regions of Non-equal Radii

Given a map of regions with non-equal radii, it is sought after to find a feasible path that visits and explores each region. It is assumed that the map satisfies D_4 condition with respect to the largest r_{region}. The methodology to tackling this extended problem entails using Algorithm 1, proposed in the previous chapter, to solve for the maneuver along the ETSP solution for an assumed map with all the regions having a radius equal to that of the largest region, $r_{region,max}$.

In the proposed approach presented in the previous chapter, the path parameterization used in defining the local minima, and hence the solution, is also used in planning the waypoint sequence for tracing the path. In this extended case, however, the the modified parameterization of Dubins paths for regions of non-equal radii is used to trace the solution of the algorithm for regions of radius $r_{region,max}$. In the case presented in the previous chapter, the same headings, β's, were used in tracing the path. For regions of non-equal radii, on the other hand, the departure heading of the local maneuver on the boundary of every region is set equal to the angle θ in Figure 5.1 and Figure 5.2.

Consider a map of 8 regions denoted as T in equation (5.13) where the first column is the x-coordinate of the center of every region, the second column is the y-coordinate of the center of every region, and the corresponding radius of every region in the third column.

$$T = \begin{pmatrix} 5,53,2 \\ 17,5,1 \\ 9,27,2 \\ 21,99,1 \\ 86,23,3 \\ 77,64,1 \\ 86,82,1 \\ 84,53,3 \end{pmatrix}. \tag{5.13}$$

The solution to the ETSP in this case as per [33] is $[4,7,6,8,5,2,3,1]$ and the global maneuver as per Algorithm 1, for a map with the same region centers and $r_{region} = r_{region,max} = 3$, is 'RSR-RSL-LSR-RSR-RSR-RSR-RSR-RSR'. Figure 5.3 shows the feasible path solution obtained by the planner following the ETSP solution.

Note that computational efficiency, as validated in the previous chapter, is independent of the radius of the region. In other words, this extended problem does not impose computational effort on the solver and as a result the planner can solve for densely populated maps provided it satisfies D_4 conditions. Figure 5.4a and Figure 5.4b show the feasible path solution for maps with 40 and 100 regions with $r_{region,max} = 3$.

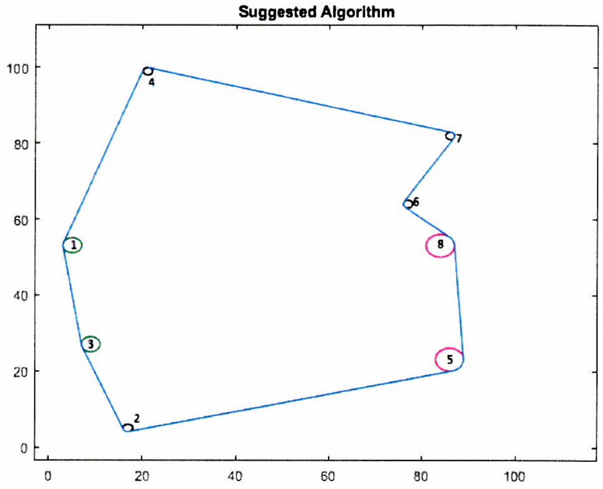
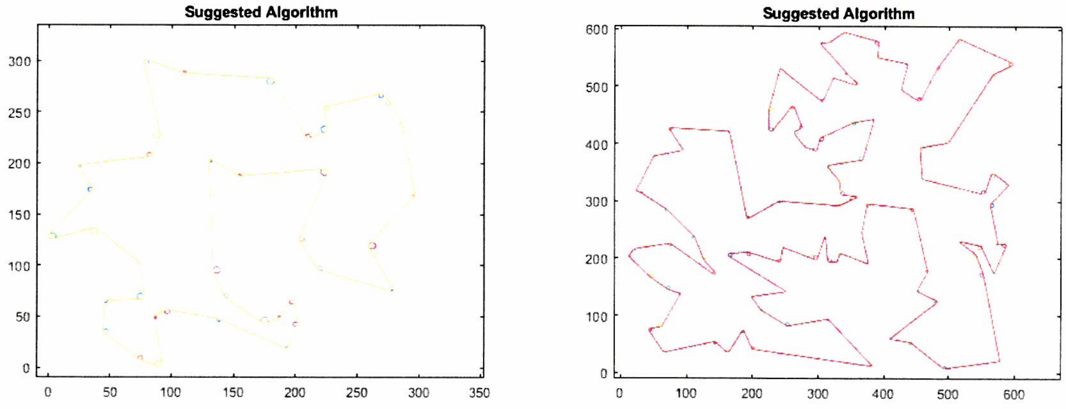


Figure 5.3: Map with 8 regions of non-equal radii.



(a) Map with 40 regions.

(b) Map with 100 regions.

Figure 5.4: Simulation results of the proposed algorithm for maps with different number of regions (40 vs. 100) but with same radius, $r_{region,max} = 3$.

Chapter 6

Conclusion and Outlook

In this study, a novel algorithm was developed that solves for the optimal path to the constrained DTSPN problem; visit and explore regions of interest by traversing a nonzero arc around each region. This is done by defining and solving for the local minima headings between every pair of regions along the path visiting sequence. For validating the proposed algorithm, the brute force approach was also investigated; solving for the length of every possible path to extract the shortest path. The proposed approach leverages the convex optimization problem to the classical DTSPN to solve for the optimal solution that converges to that of the brute force approach with minimal computational effort. While the brute force showed acceptable time performance for up to 15 regions, the proposed algorithm sustained a constant fast performance tested for maps with number of regions of order 10^4. In addition, an extension to the problem was introduced by considering regions with non-equal radii. The proposed algorithm was used to solve for a feasible path solution to visit and explore each of these regions while taking into account the variations in the turning radius along the path sequence in tracing the global path. This dictated re-parameterizing Dubins CSC paths for regions of non-equal radii.

One of the shortcomings of the proposed approach is that it offers no guarantee that the path will traverse a meaningful non-zero arc around the regions. In other words, the path might traverse a non-zero arc length around regions that might not be large enough to account for the exploration factor imposed by the problem proposed in this study. This is more likely to occur for regions located on a straight line along a path sequence. A minimum arc tolerance can be introduced to the optimization conditional statement in Chapter 5 to restrict the path to traverse a minimum arc length. Under such constraints, however, the proposed algorithm offers no guarantees in finding the shortest path solution. Furthermore, while the proposed approach demonstrated consistent, fast performance for maps of order up to 10^4, the computation time significantly increased for order 10^5 and higher.

The proposed approach assumes that regions in the map satisfy D_4 condition;

the distance between every pair of regions, of equal radius r_{region}, is greater than $4r_{region}$. This implies that the shortest path connecting an initial and final configuration is a Dubins maneuver of type CSC. In more populated maps where the D_4 condition is violated, CCC paths are considered as optimal local maneuvers in Dubins synthesis presented in [9], and as such the formulation of the proposed solution must be revisited. Future work may also entail revisiting the extended problem for regions with non-equal radii; while the planner solves for a feasible path to visit and explore every region, further work must be done in validating the optimality of the proposed solution.

Bibliography

[1] P. C. Ltd, 2019. https://www.prodrone.com/products/pd6b-aw-arm/.

[2] T. Frey, "Using drones to eliminate future forest fires," 2018. https://futuristspeaker.com/technology-trends/using-drones-to-eliminate-future-forest-fires/.

[3] D. G. Macharet, A. A. Neto, V. F. da Camara Neto, and M. M. Campos, "An evolutionary approach for the Dubins traveling salesman problem with neighborhoods, in Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation.," *ACM*, pp. 377–384, 2012.

[4] I. Lugo-Crdenas, G. R. Flores, S. Salazar, and R. Lozano, "Dubins path generation for a fixed wing UAV," *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014.

[5] L. Yang, J. Qi, D. Song, J. Xiao, J. Han, and Y. Xia, "Survey of robot 3d path planning alogrithms," *Journal of Control Science and Engineering*, 2016.

[6] L. Li, X. Wang, and M. Tan, "A Survey on Path Planning Algorithms in Robotic Fibre Placement," *The 27th Chinese Control and Decision Conference*, 2015.

[7] L. Yang, J. Qi, J. Xiao, J. Han, and Y. Xia, "Survey of Robots 3D Path Planning Algorithms," *Journal of Control Science and Engineering*, 2016.

[8] P. Yao, H. Wang, and Z. Su, "Cooperative path planning with applications to target tracking and obstacle avoidance for multi-UAVs," *Aerospace Science and Technology*, vol. 54, 2016.

[9] L. E. Dubins, "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial," *IEEE Robotics and Automation*, vol. 79, no. 3, pp. 497–516, 1957.

[10] W. Zhu, L. Li, L. Teng, Y. Chenglong, and K. Jiaxun, "Enhanced sparse a* search for uav path planning using dubins path estimation," *33rd Chinese Control Conference*, 2014.

[11] B. Paden, M. Cáp, S. Z. Yong, D. S. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *CoRR*, vol. abs/1604.07446, 2016.

[12] A. Dumitrescu and J. Mitchell, "Approximation algorithms for TSP with neighborhoods in the plane," *J. Algorithms*, vol. 48, no. 1, pp. 135–159, 2003.

[13] P. Vana and J. Faigl, "On the Dubins Travelling Salesman Problem with Neighborhoods," *International Conference on Intelligent Robots and Systems, IEEE*, pp. 4029–4034, 2015.

[14] J. Isaacs, D. Klein, and J. Hespanha, "Algorithms for the traveling salesman problem with neighborhoods involving a Dubins vehicle," *Proc. American Control Conf. (ACC)*, pp. 1704–1709, 2011.

[15] M. Tuqan, N. Daher, and E. Shammas, "A Simplified Path Planning Algorithm for Surveillance Missions ofUnmanned Aerial Vehicles," *IEEE Advanced Intelligent Mechatronics*, 2019.

[16] T. M. Cabreira, L. B. Brisolara, and F. P. R. Jr, "Survey on Coverage Path Planning with Unmanned Aerial Vehicles," *Drones*, 2019.

[17] J.-D. Boissonnat, A. Cerezo, and J. Leblond, "Shortest path of bounded curvature in the plane," *IEEE Robotics and Automation*, p. 23152320, 1992.

[18] B. Panomruttanarug, "Application of iterative learning control in tracking a Dubin's path in parallel parking," *International Journal of Automotive Technology*, vol. 18, 2017.

[19] W. Cai, M. Zhang, and Y. R. Zheng, "Task Assignment and Path Planning for Multiple Autonomous Underwater Vehicles Using 3D Dubins Curves," *Sensors*, vol. 17, 2017.

[20] W. Cai and M. Zhang, "Smooth 3D Dubins Curves Based Mobile Data Gathering in Sparse Underwater Sensor Networks," *Sensors*, vol. 18, 2018.

[21] R. Kikutis, J. Stanknas, D. Rudinskas, and T. Masiulionis, "Adaptation of Dubins Paths for UAV Ground Obstacle Avoidance When Using a Low Cost On-Board GNSS Sensor," *Sensors*, vol. 17, 2017.

[22] S. Rathinam, R. Sengupta, and S. Darbha, "A Resource Allocation Algorithm for Multivehicle Systems With Nonholonomic Constraints," *IEEE Transactions on Automation Science and Engineering*, vol. 4, 2007.

[23] Y. Lin and S. Saripalli, "Path Planning Using 3D Dubins Curve for Unmanned Ariel Vehicles," *International Conference on Unmanned Aircraft Systems*, 2014.

[24] J. Wang, S. Wu, H. Li, and J. Zou, "Path Planning Combining Improved Rapidly-Exploring Random Trees with Dynamic Window Approach in ROS," *IEEE*, 2018.

[25] D. Devaurs, T. Simeon, and J. Cortes, "Optimal Path Planning in Complex Cost Spaces With Sampling-Based Algorithms," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, 2016.

[26] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime Motion Planning using the RRT*," *International Conference on Robotics and Automation (ICRA)*, 2011.

[27] L. Jaillet, J. Cortes, and T. Simeon, "Sampling-Based Path Planning on Configuration-Space Costmaps," *IEEE Transactions on Robotics*, vol. 26, no. 4, 2010.

[28] K. J. Obermeyer, "Path planning for a uav performing reconnaissance of static ground targets in terrain," *AIAA Guidance, Navigation, and Control Conference*, pp. 10–13, 2009.

[29] N. Mathew, S. L. Smith, and S. L. Waslander, "Multirobot Rendezvous Planning for Recharging in Persistent Tasks," *IEEE TRANSACTIONS ON ROBOTICS*, vol. 31, 2015.

[30] M. Zia, Z. Cakir, and D. Z. Seker, "Spatial Transformation of Equality Generalized Travelling Salesman Problem to Travelling Salesman Problem," *ISPRS International Journal of Geo-Information*, 2018.

[31] W. Ghadiry, J. Habibi, A. G. Aghdam, and Y. M. Zhang, "Enhancements to patrolling operations based on Dubins' Traveling Salesman Problem," *World Automation Congress (WAC)*, 2016.

[32] P. Va and J. Faigl, "The Dubins Travelling Salesman Problem with Constrained Collecting Maneuvers," *Acta Polytechnica CTU Proceedings*, vol. 6, 2016.

[33] S. Arora, "Polynomial Time Approximation Schemes for Euclidean Traveling Salesman and Other Geometric Problems," *Princeton University*, 1998.

[34] K. Savla, E. Frazzoli, and F. Bullo, "On the point-to-point and travelling salesman problems Dubins' vehicle," *Preceedings of the American Control Conference, IEEE*, pp. 786–791, 2005.

[35] P. Isaiah and T. Shima, "Motion planning algorithms for the Dubins Travelling Salesperson Problem," *Automatica*, vol. 53, pp. 247–255, 2015.

[36] X. Ma and D. A. Castanon, "Receding horizon planning for Dubins traveling salesman problems,," *IEEE Conference on Decision and Control*, pp. 5453–5458, 2006.

[37] X. Goaoc, H. Kim, and S. Lazard, "Bounded-curvature shortest paths through a sequence of points using convex optimization," *SIAM Journal on Computing,*, vol. 42, no. 2, pp. 662–684, 2013.

[38] J. L. Ny, E. Feron, and E. Frazzoli, "On the Dubins Traveling Salesman Problem," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 265–270, 2012.

[39] K. Obermeyer, P. Oberlin, and S. Darbha, "Sampling-Based Roadmap Methods for a Visual Reconnaissance UAV," *AIAA Guidance, Navigation, and Control Conference*, 2010.

[40] 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.

[41] X. Zhang, J. Chen, B. Xin, and Z. Peng, "A memetic algorithm for path planning of curvature-constrained uavs performing surveillance of multiple ground targets," *Chinese Journal of Aeronautics*, vol. 27, no. 3, pp. 622–633, 2014.

[42] X. N. Bui, P. Soueres, J. D. Boissonnat, and J. P. Laumond, "The shortest paths synthesis for non-holonomic robots moving forwards," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2–7, 1994.