On Efficient Computation of Shortest Dubins Paths Through Three Consecutive Points

Armin Sadeghi Stephen L. Smith

***Abstract*— In this paper, we address the problem of computing optimal paths through three consecutive points for the curvature-constrained forward moving Dubins vehicle. Given initial and final configurations of the Dubins vehicle, and a midpoint with an unconstrained heading, the objective is to compute the midpoint heading that minimizes the total Dubins path length. We provide a novel geometrical analysis of the optimal path, and establish new properties of the optimal Dubins' path through three points. We then show how our method can be used to quickly refine Dubins TSP tours produced using state-of-the-art techniques. We also provide extensive simulation results showing the improvement of the proposed approach in both runtime and solution quality over the conventional method of uniform discretization of the heading at the mid-point, followed by solving the minimum Dubins path for each discrete heading.**

I. Introduction

Routing problems for non-holonomic vehicles have been studied extensively in the fields of robotics and autonomous systems [1], [2], [3], [4]. The non-holonomic motion of a forward-moving Dubins vehicle with bounded turning radius [5] is commonly studied as a model for fixed-wing aerial vehicles. A configuration of a Dubins vehicle consists of a location (x, y) in the Euclidean plane and a heading $\alpha \in [0, 2\pi)$. The motion of the Dubins' vehicle with minimum-turning radius $R_{\min}$ and control input $u \in [-1/R_{\min}, 1/R_{\min}]$ is governed by the following equations:

$$\dot{x} = \cos\alpha, \quad \dot{y} = \sin\alpha, \quad \dot{\alpha} = u.$$

Dubins [5] provided the set of candidate optimal paths between pair-wise configurations of the Dubins vehicle.

In this paper, we focus on the Dubins path problem between three consecutive points, where headings at only the initial and final point are fixed. Our interest in this problem stems from two applications. First, given a Dubins path through a set of points, a fast solution to this problem provides a method for inserting a new point into the Dubins path with minimum additional cost. Second, we show how it can be used as a tool to perform repeated local optimizations on a Dubins path through a set of points.

Related work: Ma *et al.* [6] study the optimal Dubins paths for three consecutive points where the initial heading is fixed and the midpoint and final point have free headings, building on the optimal control results in [7]. Under the assumption that the pairwise Euclidean distance between all points is at least $2R_{\min}$, the authors provide a sufficient condition for the optimal path between the points. In addition, a receding horizon algorithm is proposed to construct feasible Dubins path on an ordered set of points.

The authors of [8] formulate a family of convex optimization sub-problems to address the problem of the optimal Dubins path between a set of n ordered points with distance at least $4R_{\min}$ apart. The drawback of the approach is that the number of convex optimization sub-problems can grow to $2^{(n-2)}$ in the worst case. Their approach provides a solution to the three-point Dubins problem, but it requires solving several convex optimization problems. A heuristic was recently proposed [9] to extend this method to the problem of Dubins paths through neighborhoods.

Another closely related problem is the Dubins TSP, where given n points, the objective is to sequence the points and choose a heading at each point such that the resulting Dubins tour length is minimum. In [10], [11], [12], approximation algorithms are proposed to assign headings to each point when the points are ordered according to the Euclidean TSP tour. In [12], the headings are assigned by a heuristic solution to the three-point Dubins problem considered in this paper. This heuristic approach is adopted in [13] to insert points into the tours of multiple-Dubins vehicles.

In [14], the continuous interval of headings at each point is approximated by a finite number of samples. Each sample, along with the position of the corresponding point forms a configuration, and the problem reduces to computing a generalized traveling salesman problem (GTSP) tour that visits one configuration for each point. The authors in [15] present an experimental comparison of Dubins TSP algorithms including the GTSP approach. Recently and built on the results for the pairwise optimal Dubins interval path, Manyam and Rathinam [16] proposed a Dubins TSP algorithm based on uniform discretization of the headings at each point to intervals.

Contributions: The focus of this paper is to provide an efficient method for computing the optimal Dubins path between three consecutive points. We present a novel analysis of the problem that relies on inversive geometry, and results in a set of equations defining the optimal heading at the mid-point. We provide a simple method to approximate the optimal heading, and give bounds on its worst-case deviation from optimal. We then present an iterative method that is guaranteed to converge to the optimal solution. In simulation, we compare our approach to the uniform discretization method of [14] in both solution quality and computation time. Finally, we show that a Dubins TSP can be solved using a coarse heading discretization followed by repeated

This research is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

The authors are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo ON, N2L 3G1 Canada (a6sadeghiyengejeh@uwaterloo.ca; stephen.smith@uwaterloo.ca)

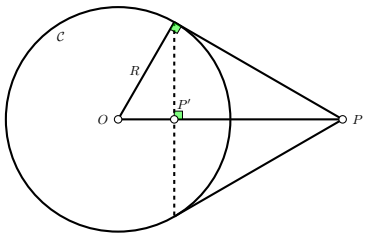

Fig. 1: Point P' is the inverse of point P with respect to circle $\mathcal{C} = \text{circle}(O, R)$.

heading optimization using our technique to achieve high-quality tours in approximately 8% of the computation time.

II. Preliminaries

Here we provide a brief background on circle inversion [17]. In two dimensional geometry, circle inversion is a mapping of a geometric object Q with respect to a circle $\mathcal{C} = \text{circle}(O, R)$ to another object $\text{inv}(Q, \mathcal{C})$. The inverse of a point P with respect to $\mathcal{C}$ is a point on the segment $\overline{OP}$ with distance $\frac{R^2}{|OP|}$ from P. The inverse of a line (resp. circle) with respect to circle $\mathcal{C}$ is a circle, unless the line (resp. circle) contains O, in which case the inverse is a line. The inverse of a line (resp. circle) is obtained by inverting three points on the line (resp. circle). With a slight abuse of terminology, we define inverse of a line segment S with respect to $\mathcal{C}$ to be the inverse of the infinite line containing the line segment S with respect to $\mathcal{C}$.

The angle between a circle and an intersecting line is defined as the angle between the line and the tangent to the circle at the intersection point. The angles between the intersecting lines and circles are preserved under the circle inversion operation.

III. Problem Formulation

We now formulate the problem of finding an optimal path for a Dubins vehicle between three consecutive points.

A. Three-Point Dubins Path

Let the tuple $X_i = (\mathbf{x}_i, \alpha_i)$ denote a Dubins vehicle configuration, consisting of a point $\mathbf{x}_i$ in the Euclidean plane, and a heading $\alpha_i \in [0, 2\pi)$ at $\mathbf{x}_i$. An alternative representation of the heading at $\mathbf{x}_i$ is two circular arcs (left and right turns) containing $\mathbf{x}_i$ and tangent to the heading. Given initial and final configurations X_i and X_f, along with a midpoint $\mathbf{x}_m$ with free heading, the three-point problem is defined by the tuple $(X_i, \mathbf{x}_m, X_f)$ and stated as follows.

Problem III.1 (Three-point Dubins path)**.** Given a tuple $(X_i, \mathbf{x}_m, X_f)$, with pairwise Euclidean distances between the points $\mathbf{x}_i$, $\mathbf{x}_m$, $\mathbf{x}_f$ of at least $4R_{\min}$, find a heading α_m at $\mathbf{x}_m$ such that the length of an optimal Dubins path starting at X_i, passing through $X_m = (\mathbf{x}_m, \alpha_m)$, and arriving at X_f is minimum.

From the Bellman's principle of optimality [18], the optimal Dubins path through three configurations is obtained by concatenating two optimal Dubins paths between the pairs. Given two configurations X_1 and X_2, the optimal Dubins path from X_1 to X_2 can be computed in constant time [5]. The optimal Dubins paths between two configurations is in the set $\{CCC, CSC\}$ where S is a straight line segment and C is a circular turn with minimum turning radius in either left L or right R direction. Therefore, in general the optimal path through three points is obtained by concatenating two Dubins paths as follows.

$$\{(C_1C_2C_3)_1(C_4C_5C_6)_2, (C_1C_2C_3)_1(C_4S_5C_6)_2,$$
$$(C_1S_2C_3)_1(C_4C_5C_6)_2, (C_1S_2C_3)_1(C_4S_5C_6)_2\}.$$

From [8] the set of optimal Dubins paths under $4R_{\min}$ distance assumption of Problem III.1 is reduced to $(C_1S_2C_3)_1(C_4S_5C_6)_2$. The $4R_{\min}$ distance constraint is relaxed further in Section VII.

B. Properties of Three-Point Dubins Path

In a path of type $(C_1S_2C_3)_1(C_4S_5C_6)_2$, the arc segments C_3 and C_4 are the two incident path segments to the mid-point. In the optimal solution to Problem III.1, the two arcs incident to the mid-point have equal lengths and both are in the same turning direction i.e., left turn or right turn [8]. Thus for simplicity we represent the path as $C_1S_2C_3S_4C_5$. We summarize the properties of the optimal Dubins path through three consecutive points, provided in [8], as follows.

Lemma III.2 (Three-point Dubins)**.** *Given $(X_i, \mathbf{x}_m, X_f)$, in a shortest path of type $C_1S_2C_3S_4C_5$, the line segment between $\mathbf{x}_m$ and the center of the circle associated with the optimal heading bisects the angle between the line segments S_2 and S_4.*

Substituting the left L and right R turns for each C_i in the path $C_1S_2C_3S_4C_5$, we obtain the set of 8 candidate optimal path types for Problem III.1.

IV. Optimal Path and Inversive Geometry

In this section we use inversive geometry to establish properties of optimal paths of type $C_1S_2C_3S_4C_5$, that form the basis of our solution approach to Problem III.1.

A. Inversive Geometry in Dubins Paths

Figure 2 shows the optimal path for the case $R_1S_2R_3S_4L_5$. The points A, B and $\mathbf{x}_c$ are the centers of the circles associated with the headings at the points $\mathbf{x}_i, \mathbf{x}_f$ and $\mathbf{x}_m$ respectively. In Figure 2, the common tangent of the circles centered at A and $\mathbf{x}_c$ is an *outer-common tangent* and the common tangent of the circles centered at $\mathbf{x}_c$ and B is an *inner-common tangent*.

Figure 3 shows the inverse of the components of the path with respect to the circle $\mathcal{C}$ centered at $\mathbf{x}_m$ with radius $R_{\min}$. Each S segment in Figure 2 is shown as a line in Figure 3. The circle inversion operation on each line generates a circle containing the mid-point $\mathbf{x}_m$, shown in Figure 3 in the same color. The inverse of the circle associated with the heading at $\mathbf{x}_m$ is a line passing through the two intersection points of $\text{circle}(\mathbf{x}_m, R_{\min})$ and $\text{circle}(\mathbf{x}_c, R_{\min})$.

The following lemma provides a sufficient condition for optimality of a $C_1S_2C_3S_4C_5$ path based on Lemma III.2. The proof of the lemma is given in [19].

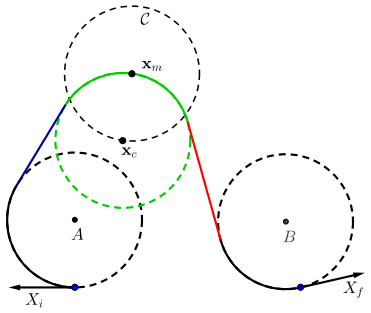

Fig. 2: An optimal path of type $R_1S_2R_3S_4L_5$. Each component of the path is sketched in different colors.

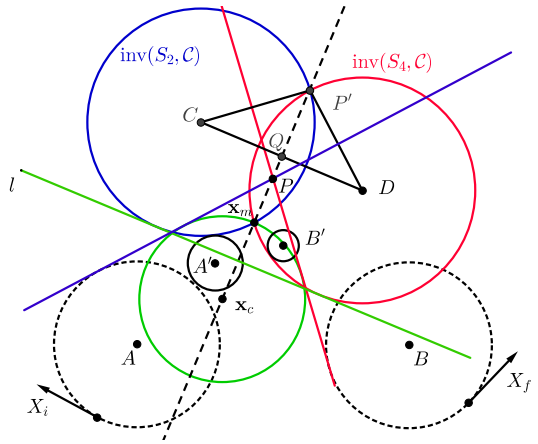

Fig. 3: Path $R_1S_2R_3S_4L_5$ and inverse of the path components. The optimal path given in the figure with initial state X_i, final state X_f and point $\mathbf{x}_m$ to visit.

Lemma IV.1 (Radius of inverted circles in an optimal path)**.** *In any optimal path of type $C_1S_2C_3S_4C_5$, the inverses of the line segments, S_2 and S_4, with respect to a circle centered at $\mathbf{x}_m$ with radius $R_{\min}$ are two circles of equal radius.*

B. Optimality Condition

Without loss of generality we set $R_{\min}$ to 1 in the rest of the paper, otherwise we scale the location of the points to satisfy the assumption. In addition, we rotate the coordinate system such that the centers A and B lie on the x-axis. Then, the optimal heading at $\mathbf{x}_m$ equals the angle between the line tangent to $\text{circle}(\mathbf{x}_c, R_{\min})$ at $\mathbf{x}_m$ and the x-axis.

Due to the $4R_{\min}$ distance constraint on the points, $\text{circle}(A, R_{\min})$ does not contain $\mathbf{x}_m$. Therefore, the inverse of $\text{circle}(A, R_{\min})$, i.e.,$\text{inv}(\text{circle}(A, R_{\min}), \mathcal{C})$, is a circle centered at point A' with radius $r_{A'}$ (see Figure 3). The point A' and radius $r_{A'}$ are defined as follows:

$$r_{A'} = \frac{1}{|\overline{A\mathbf{x}_m}|^2 - 1}, \quad |\overline{A'\mathbf{x}_m}| = |\overline{A\mathbf{x}_m}| r_{A'}. \tag{1}$$

Substituting A, A' and $r_{A'}$ with B, B' and $r_{B'}$, respectively, we can define B' and $r_{B'}$.

To derive a set of equations for the optimal heading in the path $C_1S_2C_3S_4C_5$, we require the following definitions:

- μ_A is 1 if $C_1 = C_3$ and -1 otherwise,
- μ_B is 1 if $C_5 = C_3$ and -1 otherwise,

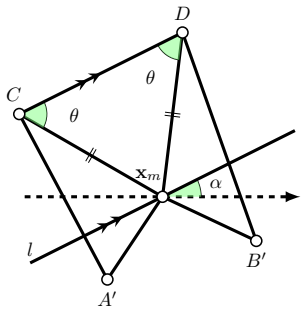

Fig. 4: Triangles $\triangle CA'\mathbf{x}_m$ and $\triangle DB'\mathbf{x}_m$ of Figure 3. Line l contains $\mathbf{x}_m$ and is parallel to the direction of the heading at $\mathbf{x}_m$.

- R is the radius of the circles centered at C and D in the optimal path,
- $\theta = \angle \mathbf{x}_m CD = \angle \mathbf{x}_m DC$ (see Figure 3),
- $\beta_1 = \angle \mathbf{x}_m AB$ and $\beta_2 = \angle \mathbf{x}_m BA$.

The following proposition (proof given in [19]) provides the set of equations to obtain the optimal heading.

Proposition IV.2. *The optimal heading α^* at $\mathbf{x}_m$ is the unique solution to the following set of equations:*

$$\frac{1}{2(\mu_A + |\overline{A\mathbf{x}_m}| \cos(\beta_1 + \theta - \alpha^*))} = R, \tag{2}$$

$$\frac{1}{2(\mu_B + |\overline{B\mathbf{x}_m}| \cos(\beta_2 + \theta + \alpha^*))} = R, \tag{3}$$

$$\frac{1}{2(1 - \sin(\theta))} = R. \tag{4}$$

The set of unknowns in Equations (2), (3) and (4) are R, α and θ, where α is the optimal heading at $\mathbf{x}_m$. Unfortunately, we have been unsuccessful in obtaining a closed form solution to these set of trigonometric equations. In Section V.I, we leverage Proposition IV.2 to bound the optimal heading at the the mid-point. Moreover, we propose a geometric method to approximate the heading, followed by an iterative procedure to converge to the optimal.

In the analysis of the inverse geometry for the $C_1S_2C_3S_4C_5$ paths, there are two special cases that must be considered in which the line segments S_2 and S_4 are parallel. Due to lack of space we refer the reader to [19] for a detailed analysis. When the lines S_2 and S_4 are parallel and non-intersecting, the result in Lemma IV.1 still holds. Moreover, all the parameters in Equations (2), (3) and (4). When S_2 and S_4 are parallel and intersecting, the analysis in [19] shows that the corresponding path $C_1S_2C_3S_4C_5$ must be the optimal path of its type.

V. Three-point Dubins Algorithm

In this section we propose a simple method to find the optimal path in the problem instance $(X_i, \mathbf{x}_m, X_f)$. First, leveraging the properties in Section IV, we propose a method to find an approximate midpoint heading.

A. Approximation Method

In this section we propose an approximation of the optimal heading at the mid-point $\mathbf{x}_m$. We assume that the pairwise distances of $\mathbf{x}_i$, $\mathbf{x}_m$ and $\mathbf{x}_f$ go to infinity. Then, the

length of segments $\overline{A\mathbf{x}_m}$ and $\overline{B\mathbf{x}_m}$ go to infinity which, by Equation (1), implies $|\overline{A'\mathbf{x}_m}|$ and $|\overline{B'\mathbf{x}_m}|$ approach zero. From Lemma I.1 in [19], the radius of the circles $\text{inv}(S_2, \mathcal{C})$ and $\text{inv}(S_4, \mathcal{C})$ is bounded from below by $\frac{1}{4}R_{\min}$. Therefore, the angles $\angle \mathbf{x}_m CA'$ and $\angle \mathbf{x}_m DB'$ (see Figure 4) approach zero and the angles $\angle C\mathbf{x}_m A$ and $\angle B\mathbf{x}_m D$ go to $\frac{\pi}{2}$.

Therefore, in terms of the angles $\beta_1 = \angle \mathbf{x}_m AB$, $\beta_2 = \angle \mathbf{x}_m BA$, $\theta = \angle \mathbf{x}_m CD$, we have $\beta_1 - \alpha + \theta = \frac{\pi}{2}$, and $\beta_2 + \alpha + \theta = \frac{\pi}{2}$. From these equations we can approximate the heading α at the mid-point α by

$$\bar{\alpha} = \frac{\beta_1 - \beta_2}{2}. \tag{5}$$

The following result (proof given in [19]) establishes the maximum error between $\bar{\alpha}$ and the optimal heading α^*.

Proposition V.1 (Maximum error of approximated heading)**.** *For the optimal path of Problem III.1, the following holds for the optimal heading at $\mathbf{x}_m$:*

$$\left| \alpha^* - \frac{\beta_1 - \beta_2}{2} \right| \leq \zeta.$$

For the optimal path P^*, the bound ζ is defined as

(i) $\zeta = 0$ *if* $|\overline{A\mathbf{x}_m}| = |\overline{B\mathbf{x}_m}|$ *and P^* is $RSRSR$, $LSLSL$, $RSLSR$, or $LSRSL$;*
(ii) $\zeta = \frac{\pi}{9}$ *if P^* is $RSRSR$ or $LSLSL$;*
(iii) $\zeta = \frac{\pi}{5}$ *if P^* is $RSLSR$ or $LSRSL$; and*
(iv) $\zeta = \frac{11\pi}{36}$ *if P^* is $RSRSL$, $LSRSR$, $RSLSL$, or $LSLSR$.*

Note that the ζ values in Proposition V.1 are the worst-case bounds. In addition, these worst-case bounds improve as the distance between the points increase. Given these approximation of the optimal heading at the mid-point, we can initialize an iterative method to converge to the optimal.

B. Iterative Method

Starting from the heading given in Section IV as the initial heading, we propose the following method for iteratively improving the heading. The method converges to the optimal heading by iteratively correcting the angle between the bisector of the two line segments of the path $C_1S_2C_3S_4C_5$ and the vector between the mid-point and the center of the circle associated with the heading (see Figure 5). Without loss of generality, assume that the center of the first curve is located at the origin and the center of the final curve is located at $(x_f, 0)$ and let $\mathbf{x}_m = (x_m, y_m)$ be the mid-point. We define vectors $\vec{v_i}$ and $\vec{v_f}$ parallel to the first and second line segments the path $C_1S_2C_3S_4C_5$. Let $\mathbf{x}_c = (x_c, y_c)$ be the center of the circle associated with a heading at the mid-point. Let Rot_θ be the rotation matrix with angle θ and $\vec{e}_v$ be the unit length vector in the direction of $\vec{v}$. We have,

$$\vec{v_i} = \text{Rot}_{\theta_i}[x_c, y_c], \quad \vec{v_f} = \text{Rot}_{\theta_f}[x_c - x_f, y_c],$$
$$\vec{v_m} = [x_m - x_c, y_m - y_c], \quad \vec{v} = \vec{e}_v + \vec{e}_{v_f}.$$

The angle θ_i is the angle of a common tangent of two circles $\text{circle}(A, 1)$ and $\text{circle}(\mathbf{x}_c, 1)$ from the line connecting the centers A and $\mathbf{x}_c$. The angle θ_i equals zero if the line segment is an outer-common tangent and $\sin^{-1}(2/|v_i|)$, otherwise. The algorithm for each path of type $C_1S_2C_3S_4C_5$ is as follows:

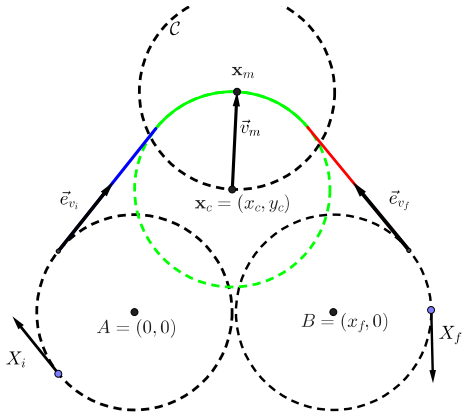

Fig. 5: The vectors $\vec{e}_{v_i}, \vec{e}_{v_f}$ and $\vec{v_m}$ for a $R_1S_2R_3S_4R_5$ path.

(i) Find the approximated heading $\bar{\alpha}$ (Equation (5)),
(ii) Compute vectors $\vec{v_i}$, $\vec{v_f}$ and $\vec{v_m}$,
(iii) Compute the vector $\vec{v}$ bisecting the angle between $\vec{v_i}$ and $\vec{v_f}$,
(iv) Return if vectors $\vec{v}$ and $\vec{v_m}$ are aligned,
(v) compute the angle γ between $\vec{v_i}$ and $\vec{v_m}$,
(vi) Rotate (x_c, y_c) about $\mathbf{x}_m$ by γ,
(vii) continue from step (ii).

The problem of finding the optimal heading at the mid-point is defined as the following:

$$\min_{x_c, y_c} \cos^{-1}(\vec{e}_v \cdot \vec{v}_m) \tag{6}$$

The minimum of (6) occurs when the vectors $\vec{v}$ and $\vec{v_m}$ are parallel. Note that the derivative of the right hand side of objective function (6) is not defined where the vectors $\vec{e}_v$ and $\vec{v}_m$ are parallel. However, minimizing (6) is equivalent to the following maximization:

$$\max_{x_c, y_c} \vec{e}_v \cdot \vec{v}_m \tag{7}$$

To prove correctness of the iterative method, it suffices to show that all local maxima (x_c, y_c) of (7) are globally maximal. The following lemma validates the iterative method. The detailed proof of the lemma is given in [19].

Lemma V.2. *The center of the circle associated with the optimal heading is the unique maximizer of* (7)*.*

An immediate consequence of Lemma V.2 is the convergence of the iterative method.

Corollary V.3. *The iterative method converges to the optimal heading at the mid-point.*

Remark V.4 (Eliminating the distance constraint). The $4R_{\min}$ distance constraint in Problem III.1 ensures that the path types are of type $C_1S_2C_3S_4C_5$. Eliminating the distance constraint introduces additional path types with CC and CCC segments. The iterative method is applicable to any $C_1S_2C_3S_4C_5$ path type even when $4R_{\min}$ is not satisfied. Implementing the method for computing CC paths in [3] alongside our iterative method for $C_1S_2C_3S_4C_5$ paths, we

obtain a method to optimally find the heading at the mid point for all path types between three consecutive points with exception of $\{C_1C_2C_3S_4C_5, C_1S_2C_3C_4C_5, C_1C_2C_3C_4C_5\}$. Although these path types are not considered in our method, simulations in Section VII show the paths generated by our method are within 0.1 percent of the optimal path. •

VI. Locally Optimizing a Dubins TSP Tour

The solution to Problem III.1 provides a method for locally optimizing a Dubins TSP tour in a post-processing phase. Given a set of n points in the Euclidean plane, a solution to the Dubins TSP is an ordering of the n points, along with a heading at each point that minimizes the total path length. Let T be a Dubins tour such that T_i is the ith configuration $(\mathbf{x}_i, \alpha_i)$. Now we define our post-processing method as follows:

(i) For every T_i, solve the problem $(T_{i-1}, \mathbf{x}_i, T_{i+1})$ and update α_i,

(ii) Randomly delete a configuration T_i in T and re-insert to a position in the tour with minimum additional cost.

Note that every segment of three consecutive vertices on the tour is a $(X_i, \mathbf{x}_m, X_f)$ problem instance. Therefore, in a tour of length n, finding the position to insert a point with minimum additional cost requires solving $n-1$ problem instances of type $(X_i, \mathbf{x}_m, X_f)$. The steps (i) and (ii) of refinements terminates if there is no improvement in the path.

VII. Simulation Results

We evaluate the performance of the proposed approach on both randomly generated $(X_i, \mathbf{x}_m, X_f)$ instances and in post-processing Dubins TSP tours as in Section VI. The point-to-point Dubins path [20] and the three-point Dubins method are implemented in Python and the experiments are conducted on an Intel Corei5 @2.5Ghz processor. The experiments in this section consider a Dubins vehicle with $R_{\min} = 1$.

A. Three-Point Dubins

In this section we compare performance if the iterative method to discretizing the heading at $\mathbf{x}_m$ with 360 equally-spaced headings. Let α_d be a heading among the discretized headings. The discretization method creates the configuration $X_m = (\mathbf{x}_m, \alpha_d)$, and solves two Dubins path problems, namely (X_i, X_m) and (X_m, X_f). The discretization method returns the minimum path among the headings.

Figure 6 shows the deviation of the path length produced by the iterative method to that of the discretized heading. The experiments are conducted on 50000 random $(X_i, \mathbf{x}_m, X_f)$ instances, where the points are uniformly randomly selected in a 10×10 environment. The x-axis in Figure 6 is the rounded minimum distance of the three points. For example, 1 on the x-axis represent the instances where the minimum distance between the points is in interval $[1, 1.5)$. The negative values represent instances in which the proposed methods outperform the discretization method. The distribution shows that even when points are less than $4R_{\min}$ apart from each other, the iterative method generates shorter paths.

The average computation time may vary based on the distances of the points due to considering additional path types

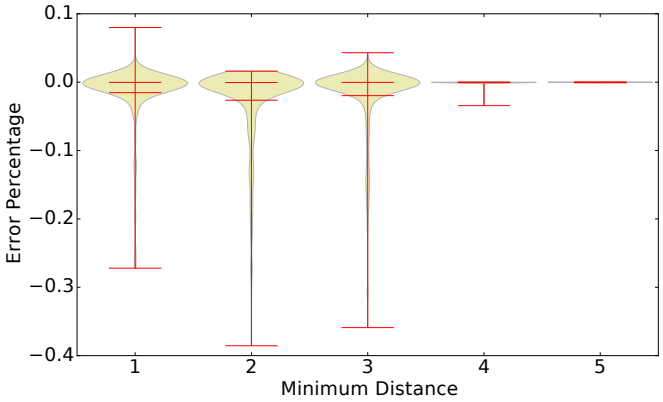

Fig. 6: The percentage deviation of path length produced by the iterative method relative to the discretization method with 360 equally spaced headings. The width of the distributions represent the probability an instance lying in the corresponding error percentile.

	$2R_{\min}$	$3R_{\min}$	$4R_{\min}$
Approx. heading	65.3	67.1	74.2
Iterative method	5.2	6.8	13.6

TABLE I: The factor of improvement in runtime of the iterative and approximation method over 360 discrete headings. The average of solver time on 10000 instances for the discretization method with 360 equally spaced headings is 0.01728 seconds.

mentioned in Remark V.4. The iterative method improves the runtime of computing a three-point Dubins path, under $4R_{\min}$ distance constraint, compared to 360 discretization by a factor of 13.65. However, this factor of improvement is 5.21 for the instances with points less than $2R_{\min}$ apart. Table I shows the factor of improvement in runtime of the iterative and approximation method when compared to discretization with 360 headings.

B. Post-processing on Dubins Tour

In this experiment, we implement the GTSP method [14] on random instances with various discretization levels followed by our post-processing method in Section VI. Given a Dubins TSP on n points, and a discretization level of d at each point, the GTSP instance will have nd vertices. The results show the advantages of the local optimization on GTSP solutions with coarse discretization over solving GTSP with fine discretization.

To characterize the performance of our algorithm, we conduct experiments on low and high-density Dubins TSP instances. Table II shows the results on uniformly randomly generated instances. Each row of the table is a class of 20 random instances with the same problem parameters: that is, the environment size $W \times W$, the number of points N, and the minimum pair-wise distance D.

The GTSP instances are solved using the state-of-the-art GTSP solver, GLKH [21] which is implemented in `C`. In Table II the abbreviations G. Len and G. Time represent the average tour length and solver time, in seconds, for the GTSP solver. Similarly, P. Len represents the average tour length after post-processing and P. Time represents the time required

		P. Time	GTSP 1-discretization					GTSP 10-discretization				
			G. Time	G. Len	P. Len	P. Len/ref	Time/ref	G. Time	G. Len	P. Len	P. Len/ref	Time/ref
Low Density	N10W15D4.0	0.1	0.0	75.5	54.6	1.000	0.02	0.8	54.8	54.6	1.000	0.17
	N10W10D3.0	0.1	0.0	66.0	38.2	1.003	0.03	0.6	38.4	38.1	1.000	0.18
	N20W20D3.0	0.3	0.0	142.5	94.5	1.002	0.01	9.5	94.7	94.5	1.002	0.26
	N30W20D2.0	0.5	0.4	187.3	110.3	1.037	0.01	18.4	107.1	106.4	1.000	0.18
	N30W30D3.0	0.7	0.1	229.0	157.6	1.006	0.01	20.5	157.1	156.7	1.000	0.15
	N40W30D4.0	1.6	0.3	289.7	205.3	1.022	0.01	45.8	201.3	200.7	1.000	0.16

		P. Time	GTSP 5-discretization			GTSP 10-discretization			GTSP 20-discretization		
			G. Time	G. Len	P. Len	G. Time	G. Len	P. Len	G. Time	G. Len	P. Len
High Density	N10W5D0.0	0.3	1.5	28.0	25.1	2.4	24.6	22.9	6.5	21.4	21.2
	N20W5D0.0	0.6	7.5	44.6	41.8	16.5	38.6	36.4	46.2	35.1	34.8
	N30W5D0.0	2.3	24.5	58.2	54.5	39.2	52.5	50.9	121.4	46.6	46.0
	N30W20D0.0	0.7	3.6	103.0	97.5	16.0	97.0	93.2	67.3	95.4	92.3
	N40W5D0.0	2.1	39.0	71.8	70.2	77.9	65.5	63.2	222.4	61.4	61.1
	N50W20D0.0	0.8	23.2	153.5	145.2	63.3	141.7	137.9	318.0	137.7	136.9

TABLE II: Average tour length and time of the GTSP approach compared to the post-processing method on random instances with low-density of points (top table) and high-density (bottom table). The instance names consist of the environment size $W \times W$, the number of points N, and the minimum pair-wise distance D.

for the post-processing of the GTSP tour. The total time of the GTSP approach and the post-processing is denoted by Time. Table II (top) shows the performance of the post-processing technique on the GTSP tours with a discretization level of 1 and 10 in low-density Dubins TSP instances. The time and the tour length of the GTSP solution with discretization level 20 is the reference, denoted by ref, for evaluating the performance of the post-processing method. The table (top) includes the ratios of the total time and post-processed tour length to the reference. In the class of instances N30W20D2.0, the deviation of the post-processed tour length from the reference is 3.7% and the total time of solving the GTSP with 1-discretization followed by the post-processing technique is just 1% of the solver-time of the GTSP approach with discretization level 20.

In an environment with high density of points, the discretization level has larger impact on the ordering of the points in a GTSP solution. Table II (bottom) shows the results of the GTSP tour with post-processing on high-density instances. For example, the results on the class of instances N50W20D0.0 show that the tour length of the post-processed GTSP tour with discretization level 5 is 5.3% longer than the GTSP tour with discretization level 20. However, the runtime is improved by a factor of 13.26.

REFERENCES

[1] J. T. Isaacs, D. J. Klein, and J. P. Hespanha, "Algorithms for the traveling salesman problem with neighborhoods involving a Dubins vehicle," in *American Control Conference*, 2011, pp. 1704–1709.

[2] 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.

[3] P. Isaiah and T. Shima, "Motion planning algorithms for the Dubins travelling salesperson problem," *Automatica*, vol. 53, pp. 247–255, 2015.

[4] B. Hérissé and R. Pepy, "Shortest paths for the Dubins' vehicle in heterogeneous environments," in *IEEE Conference on Decision and Control*, 2013, pp. 4504–4509.

[5] 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*, pp. 497–516, 1957.

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

[7] H. J. Sussmann and G. Tang, "Shortest paths for the reeds-shepp car: a worked out example of the use of geometric techniques in nonlinear optimal control," *Rutgers Center for Systems and Control Technical Report*, vol. 10, pp. 1–71, 1991.

[8] X. Goaoc, H.-S. 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.

[9] P. Vana and J. Faigl, "On the Dubins traveling salesman problem with neighborhoods," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 4029–4034.

[10] K. Savla, E. Frazzoli, and F. Bullo, "Traveling salesperson problems for the Dubins vehicle," *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1378–1391, 2008.

[11] 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, no. 1, pp. 98–104, 2007.

[12] D. G. Macharet and M. F. Campos, "An orientation assignment heuristic to the Dubins traveling salesman problem," in *Advances in Artificial Intelligence–IBERAMIA 2014*. Springer, 2014, pp. 457–468.

[13] D. G. Macharet, A. Alves Neto, V. F. da Camara Neto, and M. F. Campos, "Efficient target visiting path planning for multiple vehicles with bounded curvature," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 3830–3836.

[14] J. Le 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.

[15] M. S. Cons, T. Shima, and C. Domshlak, "Integrating task and motion planning for unmanned aerial vehicles," *Unmanned Systems*, vol. 2, no. 01, pp. 19–38, 2014.

[16] S. Manyam and S. Rathinam, "On tightly bounding the Dubins traveling salesmans optimum," *arXiv preprint arXiv:1506.08752*, 2015.

[17] D. W. Henderson and D. Taimina, *Experiencing geometry*. Prentice Hall, 2000.

[18] R. Bellman, "Dynamic programming and lagrange multipliers," *Proceedings of the National Academy of Sciences*, vol. 42, no. 10, pp. 767–769, 1956.

[19] A. Sadeghi and S. L. Smith, "On efficient computation of shortest Dubins paths through three consecutive points," Sep. 2016, arXiv preprint arXiv:1609.06662.

[20] A. M. Shkel and V. Lumelsky, "Classification of the Dubins set," *Robotics and Autonomous Systems*, vol. 34, no. 4, pp. 179–202, 2001.

[21] K. Helsgaun, "Solving the equality generalized traveling salesman problem using the Lin–Kernighan–Helsgaun algorithm," *Mathematical Programming Computation*, vol. 7, no. 3, pp. 269–287, 2015.