Receding Horizon Planning for Dubins Traveling Salesman Problems

Xiang Ma and David A. Castañón

Abstract— **In this paper we study the problem of finding the shortest global path of non-holonomic vehicles given a set of ordered waypoints. We represent the nonholonomic motion constraints using Dubins' model. Many approaches for this problem piece together smooth trajectories by considering the solution of consecutive two-point optimal trajectories, enforcing smoothness at waypoints by requiring continuity of the vehicle state between segments. In this paper, we extend the two point optimal Dubins path solution to consider three consecutive waypoints, with several variations. We use this solution in a receding horizon approach to compute a smooth path through the sequence of waypoints. We compare our approach to alternative algorithms based on two- and three-point solutions, and show the advantage of our receding horizon algorithm over these alternatives. We also consider the problem when the set of waypoints is specified, but not the order. For these problems, we compute the order using an approximate traveling salesman problem solution based on straight line distances, and then solve for a smooth path using the receding horizon algorithm. Our experimental results show that this is superior to alternative approaches proposed in the literature.**

I. Introduction

Cooperative control for unmanned air vehicles is often based on a hierarchical control strategy, where the higher level control selects and assigns activities to individual vehicles. At the lower level of control, each vehicle must select specific routes and sequences of activities in order to complete their activities in minimum time, accounting for the vehicles kinematic constraints.

When vehicles are agile and can change directions quickly relative to the inter-activity travel times, the time between two activities can be described approximately in terms of the distances between the activities. The resulting sequencing problem becomes a Traveling Salesman Problems (TSP) [2], [3]. TSPs are specified in terms of points and distances to travel between pairs of points. The goal of a TSP is to find a closed path that visits each point exactly once and incurs the least cost, consisting of the sum of the distances along the path. Distances can be defined in different ways, one of which is Euclidean distance leading to the Euclidean TSP (ETSP). Exact algorithms, heuristics as well as constant factor approximation algorithms with polynomial time requirements are available for the ETSP, as described in [4], [5]. However, when vehicles have significant kinematic constraints such as limited turning radius, the paths obtained from ETSP solutions are hard to approximate with flyable trajectories. Thus, the ETSP solution provides poor estimates of actual travel time and vehicle location.

Dept. Electrical & Computer Eng., Boston University, Boston, MA dac@bu.edu

A classical model for two-dimensional motion of vehicles with kinematic constraints is Dubins' model [6]; we refer to these models as Dubins vehicles. The solution of TSP problems with Dubins vehicles (DTSP) was recently considered in [11], [12]. The presence of kinematic constraints in Dubins vehicles implies that the distance between pairs of nodes depends on the incoming and outgoing directions of the trajectory through the node pair. As such, the distances cannot be precomputed considering only the location of the nodes. Extensions of the TSP formulation for Dubins vehicles are possible by creating multiple nodes for each physical waypoint representing possible discrete travel orientations, but these extensions result in significantly larger TSPs, making the real-time solution of the path planning problem impractical.

An alternative approach for DTSP proposed in [11] is to use a hierarchical approach: First, determine the sequence of waypoints by solving an an approximate TSP that relaxes the non-holonomic vehicle constraints. Second, find a path through the sequence of points that satisfies the nonholonomic kinematic constraints. In [11], this second problem is solved by piecing together the solution of several two-point shortest path problems with Dubins vehicles.

In this paper, we extend the approach of [11] to consider a receding horizon approach to solving for the shortest path through a sequence of points. We extend the classical solution of Dubins [6] for the shortest path between two points to the shortest path through three consecutive points using a Dubins vehicle. We subsequently use this solution in receding horizon algorithms to determine smooth paths for Dubins vehicles through a sequence of points.

We compare the performance of our receding horizon algorithms with a lower bound obtained relaxing the turn radius constraints and with the Alternating Algorithm proposed in [11]. Our experiments show that our receding horizon algorithms yield significantly better solutions than the alternative algorithms evaluated. We also compare our DTSP solutions with the approach proposed in [12] on problems where the task sequence is not specified. Our experiments show that our proposed DTSP algorithms yield significant shorter paths when compared to the random direction sampling algorithm of [12].

This paper is organized as follows. In Section II we review basic facts about Dubins' problem, and discuss extensions that we will use in our development. In Section III, we discuss the problem of finding an optimal path for a Dubins vehicle through three consecutive points, and provide characterization of the optimal paths through three points with free terminal point orientation. In Section IV, we describe several tour planning algorithms that exploit the results in the previous sections. Section V contains experimental performance comparisons of the different algorithms. Section VI contains our conclusions.

II. Problem Statement and Background

We consider a vehicle that moves at constant speed in the plane, that can execute turns with a bound on maximum curvature. Dubins' model [6] for the motion of such vehicles is described by a state vector $X(t) = [x(t), y(t), \theta(t)]^T \in \mathbb{R}^2 \times \mathbb{S}^1$, where (x, y) is the current position in the plane, and θ is the orientation angle of the vehicle. Assume without loss of generality that the absolute speed is normalized to 1, and the maximum curvature is bounded by $1/r$. The resulting system's dynamics are

$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ \theta \end{pmatrix} = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ \frac{u}{r} \end{pmatrix}, \quad |u| \leq 1 \tag{1}$$

In this paper, we are given a sequence of waypoint positions $\{(x_0, y_0), \ldots, (x_N, y_N)\}$, and an initial orientation θ_0 at time 0. The problem of interest is to find the optimal control $u(t)$ that generates a minimum time (shortest) path through the ordered sequence of points while satisfying the kinematic constraints in (1).

A. Optimal Trajectory between Two Points

Dubins' original work [6] derived conditions that characterized the optimal path between two points when both initial and terminal orientations were specified. Subsequently, Sussmann and Tang [7] rederived Dubins' results as an application of *Pontryagin's Maximum Principle*. Subsequent analyses of Dubins' shortest path problem can be found in [8], [10]. The main results are that the shortest path for a Dubins vehicle between two states (referred to as a Dubins path) must be one of six types, consisting of combinations of straight line segments and maximum curvature arcs:

- Family CCC: types RLR and LRL
- Family CSC: types RSR, LSL, RSL, LSR

Here C denotes an arc of a circle with radius r; when this arc turns clockwise (resp. counterclockwise), it will be an R (resp. L) arc. S denotes a straight line segment. The characterization reduces the problem of examining at most only 6 possibilities for determining the shortest path.

Efficient algorithms are available for determining the shortest path. Graphically, the algorithm starts by drawing two maximum curvature circles that are tangential to the initial state vector, and two maximum curvature circles that are tangential to the terminal state vector. Dubins' results indicate that the optimal trajectory selects an arc on one of the two initial circles, and connects tangentially to an arc on one of the two terminal circles. If the separation between the initial and end points is sufficient, this can only be accomplished by a line segment. There are at most four such line segments, and computation of the travel distances is straightforward, thereby specifying the optimal path.

B. Two-point Path with Free Terminal Orientation

Consider a variation of Dubins' shortest path problem where the orientation at the terminal state is free. This problem was discussed in [9]. In this case, the types of shortest paths are reduced, because the last terminal arc is not needed to match the terminal orientation. The problem is to find a minimum time trajectory with objective

$$\min_{T, |u| \leq 1} \int_0^T 1 dt \tag{2}$$

subject to constraints (1) and boundary conditions

$$X(0) = \begin{pmatrix} x_0 & y_0 & \lambda_0 \end{pmatrix}^T, x(T) = x_f, y(T) = y_f \tag{3}$$

Let λ_x, λ_y and λ_θ denote the costate variables associated with the minimum time problem. The Hamiltonian for this variational problem is

$$H(X, \Lambda, u) = 1 + \lambda_x \cos(\theta) + \lambda_y \sin(\theta) + \lambda_\theta u/r \tag{4}$$

Pontryagin's Maximum Principle [1] yields necessary conditions for the optimal trajectory:

$$\frac{d}{dt}\begin{pmatrix} \lambda_x \\ \lambda_y \\ \lambda_\theta \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \lambda_x \sin(\theta) - \lambda_y \cos(\theta) \end{pmatrix} \tag{5}$$

with boundary conditions (3) and $\lambda_\theta(T) = 0$.

Since the Hamiltonian is linear in the control u, the optimal controls are of two types: $u^* = -sign(\lambda_\theta)$ if $\lambda_\theta \neq 0$, or $u^* \in [-1, 1]$ if $\lambda_\theta = 0$ in an interval. The last condition corresponds to a singular arc, which in this case is a straight line trajectory ($u = 0$) with orientation $\tan\theta = \frac{\lambda_y}{\lambda_x}$, and the first condition correspond to maximum curvature arcs of radius r.

Note the following consequence of the necessary conditions: along optimal trajectories,

$$\frac{d}{dt}\lambda_\theta = \lambda_x \frac{d}{dt}y - \lambda_y \frac{d}{dt}x$$

which integrates to

$$\lambda_\theta = \lambda_x y - \lambda_y x + C \tag{6}$$

for some constants C, λ_x and λ_y. In particular, this implies that all points along optimal trajectories at which $\lambda_\theta = 0$ lie on the same straight line in the plane. Such points describe the straight line singular arcs as well as points of transition from one maximum curvature arc to a different maximum curvature arc, and it includes the destination point (x_f, y_f). As in [9], this characterization yields the results that the optimal trajectories must be one of two types:

- Family CS: types RS and LS
- Family CC_v: types RL and LR, with $v > \pi$

In particular, the final condition specifies that, when two arcs follow back to back, the final arc must be longer than π radians. It is also straightforward to characterize the regions where family CC_v is optimal: the terminal destination must be inside of a maximum curvature circle tangential to the initial position and orientation. This leads to fast algorithms

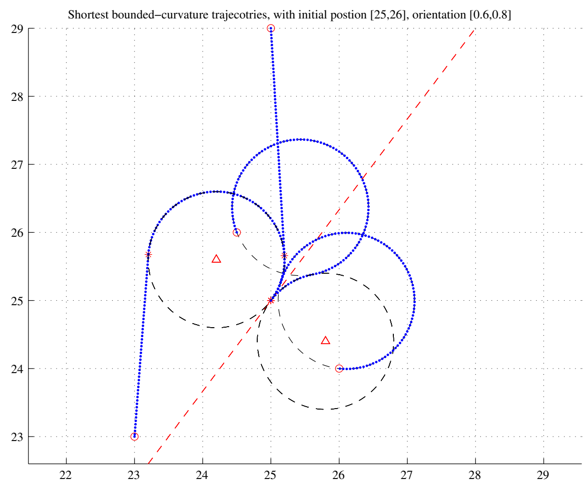

Fig. 1. Sample Paths with Free Terminal Orientation for 3 different destinations

for computing the optimal trajectory given the initial state and final position [9]. Typical sample paths are shown in 1.

III. Three-point Dubins Paths with Free Terminal Orientation

In this section, we analyze the problem of finding the shortest path for a Dubins vehicle starting from initial state (x_0, y_0, θ_0), passing the fixed intermediate point (x_m, y_m) and reaching the final point (x_f, y_f). We follow the analysis in [7], solving a minimum time problem with objective 2 and constraints (1)(3), plus the visiting constraints

$$x(t_1) = x_m, y(t_1) = y_m \text{ for some } t_1 \in (0, T) \quad (7)$$

The Hamiltonian is again given by (4). Pontryagin's Maximum Principle yields necessary conditions for optimality, with the evolution of the costates as in (5). However, the boundary conditions are more complex due to the presence of the intermediate point constraints (7). At the terminal time T, $\lambda_\theta(T) = 0$. At the intermediate time t_1, which is a free variable,

$$\begin{aligned} \lambda_x(t_1^+) &= \lambda_x(t_1^-) + k_1 \\ \lambda_y(t_1^+) &= \lambda_y(t_1^-) + k_2 \\ \lambda_\theta(t_1^+) &= \lambda_\theta(t_1^-) \\ x(t_1) = x_1,\ y(t_1) = y_1,\ H(t_1^+) &= H(t_1^-) \end{aligned} \quad (8)$$

Along the optimal trajectory, the optimal control satisfies

$$\min_{\|u\|\le 1}[H] = \min_{\|u\|\le 1}[1 + \lambda_x \cos\theta + \lambda_y \sin\theta + \lambda_\theta u] = 0 \quad (9)$$

From (5), $\lambda_x(t)$ and $\lambda_y(t)$ are constant in the intervals $t \in (0, t_1)$ and $t \in (t_1, T)$, with a jump discontinuity at t_1 as indicated in (8). For clarity, in the subsequent development, we denote constants $\lambda_x(t_1^-) = \lambda_x, \lambda_y(t_1^-) = \lambda_y$ and $\lambda_x(t_1^+) = \lambda_x + k_1, \lambda_y(t_1^+) = \lambda_y + k_2$.

As before, the time-varying costate $\lambda_\theta(t)$ determines the optimal control: $u^*(t) = -sign(\lambda_\theta)$ whenever $\lambda_\theta \neq 0$. When $\lambda_\theta = 0$, the optimal trajectory is a singular arc corresponding to a straight line segment with fixed angles satisfying $\tan(\theta_0) = \frac{\lambda_y}{\lambda_x}$ if the segment is before t_1, or $\tan(\theta_1) = \frac{\lambda_y + k_2}{\lambda_x + k_1}$ if the segment is after t_1.

Following an argument similar to (6) or Lemma 2.1 of [9], the points where $\lambda_\theta(t) = 0, t > t_1$ must lie in a straight line

$$\lambda_x y + k_1 - \lambda_y x - k_2 + C = 0$$

This line includes the terminal point (x_f, y_f) because of the boundary condition $\lambda_\theta(T) = 0$. Similarly, the points where $\lambda_\theta(t) = 0, t < t_1$ lie on a straight line

$$\lambda_x y - \lambda_y x + C = 0$$

These lines contain the points where the optimal control may switch between the three type of optimal control: $u = \pm 1$ or $u = 0$. We refer to these switching points as inflection points.

We now characterize the optimal trajectories and the corresponding optimal controls. Note that, along the optimal trajectories, the Hamiltonian and λ_θ are continuous at the intermediate point (x_m, y_m).

Theorem 3.1: Consider the three point Dubins vehicle shortest path problem with free terminal point orientation. Then, the optimal trajectory must be one of four types $CSCS$, $CSCC$, $CCCS$ or $CCCC$ (or their shortened version), where C stands for a maximum curvature segment, and S stands for a straight line segment.

Proof: Let t_1, θ_m denote the time and orientation of an optimal trajectory when it crosses the midpoint (x_m, y_m). From Bellman's principle of optimality, this optimal trajectory must be a minimum-time Dubins path between states (x_0, y_0, θ_0) and (x_m, y_m, θ_m), as well as a minimum time path from (x_m, y_m, θ_m) to (x_f, y_f). It is straightforward to show that, if the initial minimum time Dubins path is of CCC or CSC type, the costate value $\lambda_\theta(t_1) \neq 0$ because the last arc moves the costate away from the line where $\lambda_\theta = 0$. Since this costate is continuous at t_1, this implies that the final maximum curvature turn C must continue after t_1. By the principle of optimality and the results of Section II-B, the minimum time trajectory from (x_m, y_m, θ_m) to (x_f, y_f) is of type CS or CC. Hence, for the full problem from time 0 to T, the optimal trajectory must be one of four types: $CCCC, CCCS, CSCS, CSCC$, or degenerate versions of these types. ■

Trajectories that end in CC can only happen when the midpoint and endpoint are separated by less than $2r$. When the initial condition is sufficiently separated from the last two points, one can rule out the existence of optimal trajectories of the form $CSCC, CCCS$ and $CCCC$. The following conditions are sufficient to guarantee that the optimal trajectory is of form $CSCS$.

Lemma 3.1: Assume that (x_m, y_m, θ_m), (x_f, y_f) are separated by a distance less than $2r$. If the two radius r circles tangent to the initial condition do not overlap the two radius r circles containing both (x_m, y_m, θ_m) and (x_f, y_f) the midpoint and endpoint, the optimal trajectory must be of type $CSCS$.

Specifically, for any path with one inflection point after

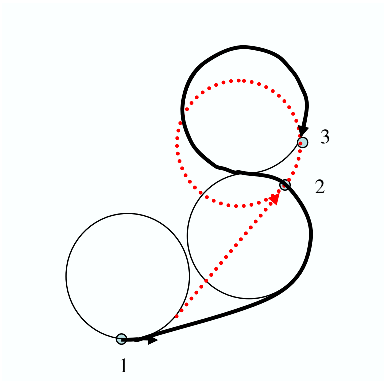

Fig. 2. Replacement of $CSCC$ path with shorter $CSCS$ path

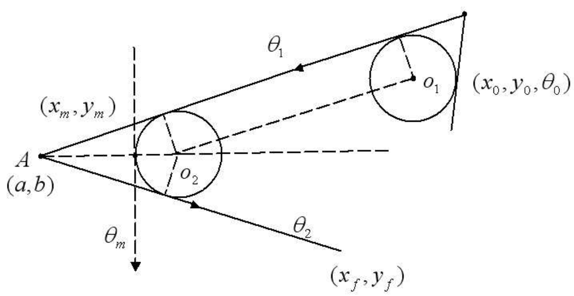

Fig. 3. Three-point Free Terminal Orientation Problem

(x_m, y_m) (thus ending in CC), one can find a shorter path that ends in CS. The proof is algebraic and tedious in nature, requiring enumeration of different geometric cases; instead, we illustrate this behavior graphically in Figure 2. This figure shows a $CSCC$ path, where the orientation chosen at the midpoint necessitates the CC ending to reach the terminal point. The dotted path is a much shorter path that chooses a different orientation at the midpoint, with the curvature heading towards the terminal point.

When the waypoints are spaced greater than twice the turning radius r of the vehicle, the shortest paths will be of type $CSCS$. These paths have two singular arcs (straight line trajectories), one before and one after the midpoint. For these cases, we can provide a further characterization of the optimal paths:

Lemma 3.2: Assume that the shortest Dubins path through three waypoints, with an initial given orientation, is of type $CSCS$. Then, on the second turn C, the midpoint (x_m, y_m) bisects the turning arc.

Proof: The lemma is illustrated in Figure 3. The claim is that a line from the midpoint (x_m, y_m) to the intersection of the two straight line segments bisects the intersecting angle. On line segments of the optimal path, we have $\lambda_\theta = 0$, $\frac{d}{dt}\lambda_\theta = 0$. Solving the set of simultaneous equations, and using the relationships in eqs. 8 yields the following:

$$\lambda_x = -\cos\theta_0; \qquad \lambda_y = -\sin\theta_0$$
$$\lambda_x + k_1 = -\cos\theta_1 \qquad \lambda_y + k_2 = -\sin\theta_1$$
$$\Rightarrow \frac{k_1}{k_2} = -\tan\frac{\theta_0+\theta_1}{2} \tag{10}$$

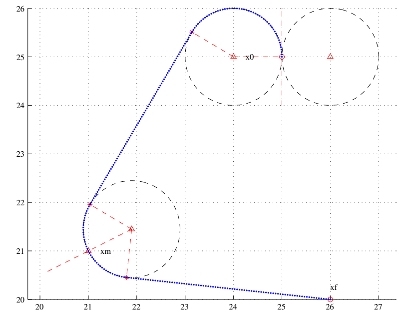

Fig. 4. Type '$LSLS$' in Three-point problem

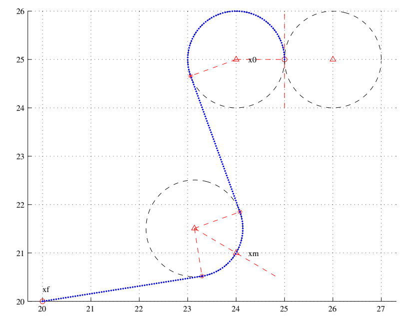

Fig. 5. Type '$LSRS$' in Three-point problem

Since the Hamiltonian is continuous on the optimal arc at (x_m, y_m), we have

$$\min_{\|u\|\le 1}[1 + \lambda_x\cos\theta_m + \lambda_y\sin\theta_m + \lambda_\theta(t_1)u] =$$
$$\min_{\|u\|\le 1}[1 + (\lambda_x + k_1)\cos\theta_m + (\lambda_y + k_2)\sin\theta_m + \lambda_\theta(t_1)u]$$

From the continuity of λ_θ at t_1, we have

$$k_1\cos\theta_m + k_2\sin\theta_m = 0 \tag{11}$$

Combining eqs. 10 and 11 yields

$$\tan\theta_m = \tan\frac{\theta_1+\theta_0}{2} \Rightarrow \theta_m = \frac{\theta_1+\theta_0}{2} \tag{12}$$

which establishes the Lemma by considering the complementary angles. ■

The above results restrict the class of possible optimal trajectories given a sequence of three waypoints, and provide simple characterizations of necessary conditions for optimality. However, we have not developed a complete characterization of all the possible geometric configurations of the three waypoints and initial orientation that can uniquely identify the optimal trajectory for any problem. The necessary conditions require the solution of coupled trigonometric equations, which can be done numerically, but are hard to describe in closed form. We have been developed an algorithm to compute the possible optimal solutions. For instance, optimal trajectories for cases satisfying Lemma 3.1 are shown in Figures 4, 5.

IV. Dubins Tours

In this section, we develop algorithms for generating complete tours through a set of waypoints under kinematic motion constraints of (1) based on the results of the previous sections. We assume that a simpler model has already selected the ordering of the waypoints; such models can be based on the solution of traveling salesman problems that relax the kinematic motion constraints. Our goal is to generate a near-optimal kinematic-feasible tour of the waypoints preserving the original order.

Let $A = \{a_1, ..., a_n\}$ be an ordered set of waypoints in a compact region $Q \subset \mathbb{R}^2$ for the DTSP problem. If A is a tour, then $a_1 = a_n$. Let $\Theta = \{\theta_1, ..., \theta_n\}$ denote possible orientations of a Dubins vehicle at a_i, for $i = 1, ..., n$. If one specifies the sets A and Θ, there is a unique optimal trajectory, consisting of a sequence of two point Dubins shortest paths. That is, the optimal trajectory between waypoint a_k with orientation θ_k and waypoints a_{k+1} with orientation θ_{k+1} can be obtained by solving for the Dubins shortest path with those initial and terminal states. However, such a trajectory will be unnecessarily long, because the orientations $\theta_2, \ldots, \theta_n$ can be chosen as variables to minimize the tour length. Below, we describe four alternative algorithms that will generate better routes.

The first algorithm we discuss is the *Alternating Algorithm* (AA) described in [11]. The AA algorithm starts from a known initial position and orientation a_1, θ_1. It proceeds by fixing the orientation at every even-numbered waypoint $a_{2k}, k = 1, \ldots, n/2$ and its subsequent waypoint a_{2k+1} to be aligned with a line segment pointing from a_{2k} to a_{2k+1}. This means that the segment (a_{2k}, a_{2k+1}) is a straight line between the points. Furthermore, the orientations have been specified at all waypoints, so the minimum time paths for the remaining segments can be computed using a sequence of two-point Dubins paths. We use the AA algorithm as a reference for our subsequent algorithms.

An alternative approach that we propose is based on a greedy solution, denoted the Two-point Algorithm. The basic idea is to use our solution in subsection II-B for the shortest Dubins path from one state (position and orientation) to a second position, with free orientation. Since the initial orientation is known at waypoint a_1, we can solve the problem of finding the shortest path to a_2. This solution will specify a direction θ_2 at a_2. Using this direction, we extend the path by solving for the shortest path from (a_2, θ_2) to a_3, which specifies θ_3. The algorithm continues the iteration, extending the path until all the waypoints have been reached.

A third approach, denoted the Three Point Algorithm, uses the solution of the three point Dubins path with free midpoint and terminal orientation developed in Section III. In this algorithm, Dubins paths are computed only at every odd point $a_{2k-1}, k = 1, \ldots, (n+1)/2$. The solution of the three point Dubins path from $(a_{2k-1}, \theta_{2k-1})$ through a_{2k} to a_{2k+1} yields orientations θ_{2k} and θ_{2k+1}. Subsequently, we solve another three point Dubins path from $(a_{2k+1}, \theta_{2k+1})$ to the next two waypoints, and repeat the process until we have extended the path to cover all the waypoints. When the number of waypoints is even, the last segment will only be a two-point path, as in Section II-B.

The last algorithm we consider is a receding horizon algorithm based on the three-point Dubins path solution in

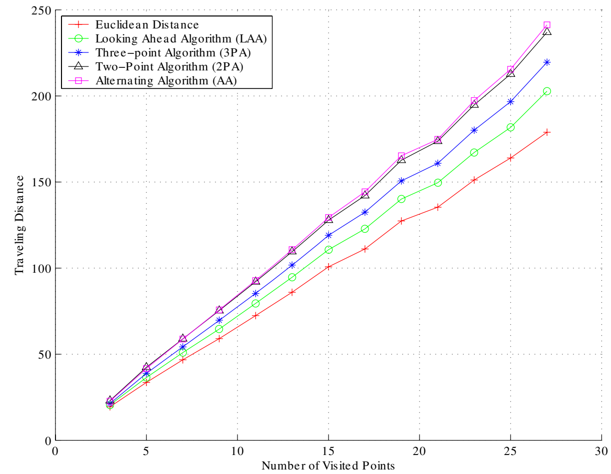

Fig. 6. Average length of tours for a Dubins vehicle generated by alterative algorithms when waypoint order is given.

III. The basic idea of this algorithm is to use this solution to determine only the path and orientation up to the middle waypoint. Thus, the solution for $(a_1, \theta_1), a_2, a_3$ is only used to determine θ_2. Note that the choice of θ_2 will be heavily influenced by the location of a_3 in the solution of the three-point Dubins path. Once θ_2 is known, the tour can be extended by solving another three point problem starting from (a_2, θ_2). We refer to this algorithm as the Look-Ahead (LA) algorithm, and it is based on receding horizon control principles.

V. Experiments

Each of the four algorithms highlighted in the previous section provides an approximate solution to the optimal tour problem with Dubins vehicles. In order to evaluate their performance, we conducted experiments on randomly generated sets of ordered waypoints. The waypoints were contained in a square of 10×10 units, where 1 unit was the minimum radius of curvature chosen for the Dubins vehicle. We varied the number of waypoints from 3 to 27 in increments of 2. For each number of waypoints, we generated 100 Monte Carlo samples and computed the average length of the tours generated by the different algorithms. Note that the order of the waypoints was selected randomly in these experiments. Specifically, no attempt was made to solve traveling salesman problems to find a good order without kinematic constraints.

Figure 6 shows the relative performance of the different algorithms, and also shows the sum of the lengths of the straight line segments between the points, which serves as a lower bound (infeasible, as kinematic turn radius constraints are not enforced) for the average performance of the algorithms. As the results show, the two algorithms based on the use of three-point Dubins paths yield shorter tours. The LA algorithm had the best performance in all cases, indicating the advantage of using a receding horizon controller that can modify current paths in anticipation of the next waypoint.

When the order of the waypoints is unknown, one must determine the optimal order and the path through the way-

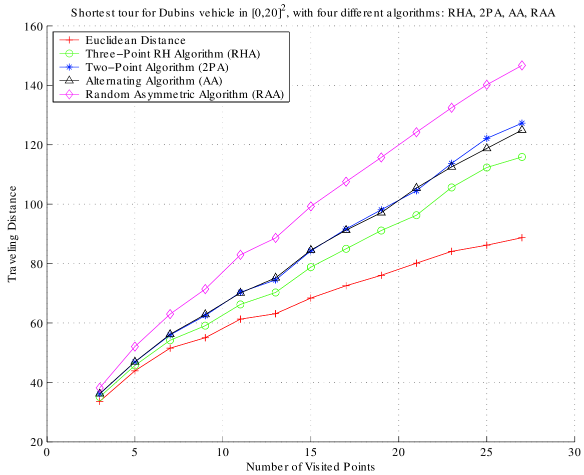

Fig. 7. Average length of tours for a Dubins vehicle generated by alternative algorithms when waypoint order is unknown.

points. One approach is to determine the order using a solution to a TSP problem based on Euclidean distances, neglecting the kinematic constraints of (1). An alternative algorithm proposed recently, the Random Asymmetric Algorithm (RAA) of [12], reverses the order of computation. First, one selects randomly the directions for the waypoints, and computes pairwise distances between all waypoints using Dubins two-point paths. The result is a TSP problem for the order which uses distances based on paths that satisfy (1). The solution of this TSP problem then selects the order, and consequently the paths. In [12], it is shown that, as the number of waypoints per unit area increases, the RAA algorithm computes better tours than the AA algorithm using the order given by a straight-line TSP formulation. In those experiments, the average spacing of waypoints was significantly smaller than the turning radius of the vehicles.

We conducted experiments on a 20×20 grid with numbers of waypoints from 3 to 27, corresponding to a sparser version of the test problems in [12]. The minimum turn radius r was 1 in these experiments. For each experiment, we computed the order generated by the solution of a TSP problem with straight line distances between waypoints, and then used our algorithms to evaluate the length of tours through these waypoints that satisfy the constraints of (1). We also implemented the RAA algorithm, and computed the corresponding length of the generated tour.

Figure 7 shows the average tour length of the different algorithms versus number of waypoints visited, averaged over 100 Monte Carlo runs where the location of the waypoints was randomly generated. It also shows the length of the shortest path tour assuming no curvature constraints, which serves as a lower bound. The results indicate that the LA algorithm computes significantly shorter paths in these experiments. Note that the average inter-waypoint distance in the denser experiments is around 5.5, which is larger than the experiments reported in [12].

VI. Conclusion

In this paper, we considered the problem of finding shortest path tours for problems with non-holonomic vehicle models that include a minimum turning radius constraint. We developed extensions of the classic two-point, known orientation shortest path problem solved by Dubins and others, to allow for unknown terminal orientation, and for determining the optimal path through three points, two of which have unknown orientation. We characterized necessary conditions for optimality of solutions, and extended these characterizations to obtain optimal paths in these extensions.

Our extensions to the results of Dubins provide the foundation for algorithms that can solve approximately the problem of finding shortest path tours, by building these tours incrementally. In particular, we developed a new receding horizon approach that uses our three-point Dubins path solution to extend a partial path by one additional waypoint, while incorporating knowledge of the location of the subsequent waypoint. Our experiments show that this receding horizon controller achieves superior performance to alternatives proposed in the recent literature.

An interesting extension of our results is to look for tours that visit areas instead of points. In the motivating unmanned air vehicle application, activities can be conducted remotely, so that a vehicle only needs to visit an area. We are developing extensions to our results that compute shortest tours for visiting sequences of areas.

VII. Acknowledgements

This work was supported in part by AFOSR grant FA9550-04-1-0133. The authors would like to thank the reviewers for their careful comments on the first draft of the paper.

References

[1] M. Athans and P. L. Falb, *Optimal Control,* McGraw-Hill, New York, NY 1966.

[2] J. Beardwood, J. Halton, and J. Hammersly, “The shortest path through many points,” *Proceedings of the Cambridge Philosophy Society,* vol.55,:299-327 1959.

[3] D. Applegate, R. Bixby, V. Chvàtal, and W. Cook, “On the solution of traveling salesman problems,” *Proceedings of the International Congress of Mathematicians,* Extra Volume ICM III, 1998.

[4] S. Arora, “Nearly linear time approximation scheme for Euclidean TSP and other geometric problems,” *Proc. 38th IEEE Annual Symposium on Foundations of Computer Sciences,* pp. 554-563, 1997.

[5] S. Lin and B.W. Kernighan, “An effective heuristic algorithm for the traveling-salesman problem,” *Operation Research,* vol.21: 498-516, 1973.

[6] 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,* 79: 497-516, 1957.

[7] H.J. Sussmann and W. Tang, “Shortest paths for the Reeds-Shepp Car: a worked out example of the use of geometric techniques in nonlinear optimal control,” *Report SYCON-91-10,* Rutgers University, 1991.

[8] J-D.Boissonnat, A.Cèrèzo, and J.Leblond, “Shortest Paths of Bounded Curvature in the Plane”, *IEEE Int. Conf. on Robotics and Automation,* Nice, France, 1992.

[9] X-N. Bui, P. Soures, J-D. Boissonnat and J-P Laumond, “The Shortest Path Synthesis for Non-Holonomic Robots Moving Forwards”, *INRIA Report,* No 2153, January, 1994.

[10] A.M. Shkel and V.J. Lumelsky, “Classification of the Dubins' set”, *Robotics and Autonomous Systems,* 34: 179-202, 2001.

[11] K. Savla, E. Frazzoli, and F. Bullo, “On the point-to-point and traveling salesperson problems for Dubins' vehicle”, *American Control Conference,* Portland, OR, 2005.

[12] J. Le Ny, E. Feron, “An Approximation Algorithm for the Curvature-Constrained Traveling Salesman Problem”, *Proceedings of the 43rd Annual Allerton Conference on Communications, Control and Computing,* September 2005.