2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Decentralized algorithm for minimum-time rendezvous of Dubins vehicles

Amit Bhatia Emilio Frazzoli

Abstract— **In this paper, we consider the problem of minimum-time rendezvous of a team of Dubins vehicle at a pre-assigned destination point starting from arbitrary initial configurations. We impose an additional constraint that the separation between arrival angles of successive team members at the destination be equal. We propose a decentralized algorithm that solves the problem up to a desired level of accuracy in finite time. The communication complexity of the algorithm is quadratic and the space complexity is constant in team size. The proposed algorithm is proved to be correct by establishing some important facts for the shortest path length of a Dubins vehicle as a function of arrival angle at the destination point.**

I. INTRODUCTION

In this paper, we consider the problem of *minimum-time rendezvous* of a team of Dubins vehicles under geometric constraints. A Dubins vehicle can be considered to be a simplified model of a mobile robot or an airplane moving in 2 dimensions with a bounded curvature and constant speed [1], [2]. The team members are required to reach a specified location (the *rendezvous point*) at the same time, and do so as quickly as possible. In addition, it is desired that the arrival angles of team members are equally spaced at the rendezvous point. This version of the rendezvous problem is especially relevant for planning attack missions in hostile environments using a team of UAVs, where maximum spread of the vehicles in terms of arrival angles at the destination can improve the chances of successful mission completion. In Fig. 1, we show one such scenario, where team members are spaced apart by $\pi/2$ radians at the rendezvous point.

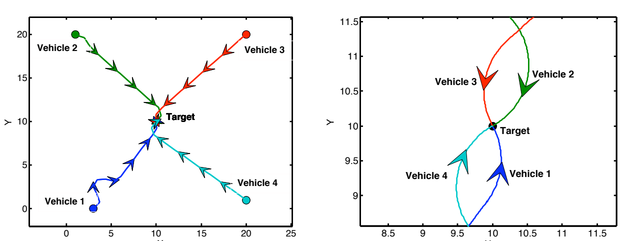

Fig. 1. Rendezvous of Dubins vehicles with maximal separation

Consensus problems have been studied by the controls community, both in discrete-time systems framework, e.g., in [3]–[8], and in continuous-time systems framework, e.g., in [9]–[16]. Provably-correct *stop-and-go* policies (both synchronous and asynchronous) that are local in nature and guarantee convergence of the agents (assumed to be point masses) to an unspecified or a specified point have been proposed in [4], [5]. In the continuous-time systems framework, time-optimal rendezvous problem for the case of two distinct linear time varying systems with amplitude constraints on the input and with unspecified destination point has been considered in [9]. Rendezvous problem for linear systems with unspecified destination point and time has also been examined in the framework of cone invariance problems in [10]. For agent models with nonlinear dynamics but no actuator constraints (e.g., unicycles), policies have been proposed in [11]–[14].

However, the problem becomes very hard to analyze, even for a simplified model of an airplane (e.g., a Dubins vehicle model), where the speed is constrained to be constant, and the curvature is bounded. To effect cooperation between a team of UAVs, the notion of *coordination functions and variables* has been recently proposed in [16]. The notion is used for executing cooperative timing missions under the presence of threats. To solve the optimization problem, a point mass model is assumed for each vehicle and the space of actions for each vehicle is discretized to a finite set. The consensus value of the coordination variables is then used to generate a feasible path based on actual dynamics.

In our work, we use the Dubins model for the agents, and address the minimum-time rendez-vous problem, without discretizing the set of inputs to arrive at the solution. Moreover, we impose an additional geometric constraint in the problem. The main contribution of this paper is a decentralized ϵ-approximate algorithm to solve the rendezvous problem[1] with accuracy ϵ in finite time. The communication complexity of the algorithm is quadratic and the space complexity is constant in team size. In order to prove correctness of the proposed algorithm, we also establish the following facts for a Dubins vehicle. First, we establish that for a given pair of initial configuration and final position, the length of the shortest path between the two is a continuous function of the vehicle heading at the final position. This result helps in proving continuity of the cost function for the team. Second, it is established by construction that a Dubins shortest paths between two given configurations can be extended to feasible paths of arbitrary lengths. This ensures that the solution satisfies feasibility condition for every vehicle.

The paper is organized as follows. In Section II, we define the problem and the notation used in the paper. In Section III, we discuss some important properties of the cost functions for a single Dubins vehicle and the team.

Amit Bhatia is with the Department of Mechanical and Aerospace Engineering, University of California at Los Angeles, Los Angeles, California 90095, abhatia@ucla.edu

Emilio Frazzoli is with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139, frazzoli@mit.edu

[1]From now on, we will refer to our problem as the rendezvous problem.

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

In Section IV, we present a way of extending the Dubins path between two configurations to arbitrary lengths. A decentralized ϵ-approximate algorithm for solving the problem approximately is presented in Section V. Simulation results are presented in Section VI and the paper is concluded in Section VII. Proofs of all the stated results are presented in the Appendix.

II. Preliminaries

We denote by $\mathcal{C}$ the configuration space for a Dubins vehicle, where, $\mathcal{C}$ is the Special Euclidean group $SE(2) = \mathbb{R}^2 \times S^1$. $C = (P, \psi), P \in \mathbb{R}^2, \psi \in [0, 2\pi)$ denotes a configuration of the vehicle. We assume that the vehicle is moving with unit speed and has a minimum turning radius of unity. Given the initial configuration $\mathcal{C}_0 = (P_0, \alpha)$ and the final configuration $\mathcal{C}_f = (P_f, \beta)$, a Dubins path is a curve (twice differentiable almost everywhere) $\Gamma : [0, T] \to \mathbb{R}^2$, starting at P_0 with direction α and ending at P_f with direction β and respecting the constraint that the curvature along the path is bounded above by 1. Let $|\Gamma|$ denote the length of Dubins path Γ. For a given pair of initial and final configuration $\mathcal{C}_0, \mathcal{C}_f$, we choose the local coordinate system (fixed at vehicle's initial position) such that $\mathcal{C}_0 = (0, 0, \alpha), \alpha \in [0, 2\pi]$, $\mathcal{C}_f = (d, 0, \beta), \beta \in [0, 2\pi]$, with all the angles being measured counter clockwise (shown in Fig. 2).

It is known that between any two configurations, the shortest path for a Dubins vehicle belongs the the Dubins set $\mathcal{D}$ which contains six *admissible paths* $\mathcal{D} = \{\mathrm{LSL}, \mathrm{RSR}, \mathrm{RSL}, \mathrm{LSR}, \mathrm{RLR}, \mathrm{LRL}\}$ (proved in [1]). Here L denotes a left turn of radius 1, R denotes a right turn of radius 1 and S denotes a straight line segment. For a given vehicle, for a fixed P_0, P_f, α, we denote by $\mathcal{L}_\Lambda(\beta)$ the length of a path $\Lambda \in \mathcal{D}$ for a given arrival angle β. Using the notation of [2], for a path $\Lambda \in \mathcal{D}_{\mathrm{long}}$ $\mathcal{L}_\Lambda(\beta) = t(\beta) + p(\beta) + q(\beta)$, where $t(\beta)$ is the length of the first segment, $p(\beta)$ is the length of the second segment, and $q(\beta)$ is the length of the last segment of the path (shown in Fig. 2). $\mathcal{L}^*(\beta) = \min_{\Lambda \in \mathcal{D}} \mathcal{L}_\Lambda(\beta)$ denotes the length of the shortest path for given angle β. $\mathcal{L}_\Lambda(\cdot)$ will be called as *vehicle cost function for path* Λ and $\mathcal{L}^*(\cdot)$ as the *vehicle optimal cost function*.

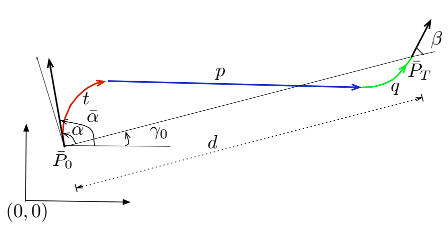

Fig. 2. Example of Dubins shortest path

Let $\mathcal{T} = \{V_j\}_{j=1}^N$ denote the team of vehicles V_j, participating in the rendezvous at location $\bar{P}_T$, where $\bar{P}$ denotes the coordinates of a point in a stationary global reference frame. $\bar{\mathcal{C}}_{0,j} = (\bar{P}_{0,j}, \bar{\alpha}_j)$ denotes the initial configuration of vehicle V_j in global coordinate system. $\gamma_{0,j}$ is the angular orientation of the local coordinate frame of vehicle j in the global reference frame (shown in Fig. 2). For a given arrival angle β for a vehicle $V_j \in \mathcal{T}$, ψ denote the phase of arrival angle and is given by $\psi = \beta + \gamma_{0,j} - 2\pi j/N$. The phase indicates the deviation of vehicle arrival angle from the one based on its position in the team. With slight abuse of notation we denote by $\mathcal{L}_j^*(\psi)$ the vehicle optimal cost for vehicle V_j for a given phase angle ψ . The cost function for the team is given by $\mathcal{L}_\mathcal{T}(\cdot) = \max_{V_j \in \mathcal{T}} \mathcal{L}_j^*(\cdot)$. The $\max$ here denotes the fact that the team cannot achieve rendezvous if all the team members cannot arrive at same time. The rendezvous problem can now be stated as:

Definition 1 (Rendezvous Problem): For a given team of Dubins vehicles $\mathcal{T} = \{V_j\}_{j=1}^N$ with initial configurations $\bar{\mathcal{C}}_{0,j}$ and destination point $\bar{P}_T$, find the optimal phase angle ψ^* and the cost $\mathcal{L}_\mathcal{T}^*$ for the team such that $\mathcal{L}_\mathcal{T}^* = \mathcal{L}_\mathcal{T}(\psi^*) = \min_{\psi \in [0, 2\pi]} \mathcal{L}_\mathcal{T}(\cdot)$, and plan paths Γ_j for each vehicle $V_j \in \mathcal{T}$, with phase angle ψ^* and $|\Gamma_j| = \mathcal{L}_\mathcal{T}^*$.

We now introduce the notion of ϵ-approximate algorithms that are guaranteed to solve the rendezvous problem in finite time up to desired level of accuracy ϵ.

Definition 2 (ϵ-approximate algorithm): An algorithm is ϵ-approximate for a given rendezvous problem if it finds a phase angle ψ^ϵ and the corresponding team cost $\mathcal{L}_\mathcal{T}^\epsilon = \mathcal{L}_\mathcal{T}(\psi^\epsilon)$ such that $\mathcal{L}_\mathcal{T}^\epsilon \leq \mathcal{L}_\mathcal{T}^* + \epsilon$ and plans paths Γ_j for each vehicle $V_j \in \mathcal{T}$, with phase angle ψ^ϵ and $|\Gamma_j| = \mathcal{L}_\mathcal{T}^\epsilon$, in finite time.

III. Cost functions

In this section we first discuss properties of $\mathcal{L}^*(\cdot)$ and then those of $\mathcal{L}_\mathcal{T}(\cdot)$. We will show that the function $\mathcal{L}^*(\cdot)$ is continuous and moreover its derivative does not change sign more than 16 times. An arrival angle where the derivative of $\mathcal{L}^*(\cdot)$ changes its sign will be denoted by β_{crit} and the set of all such arrival angels will be denoted by $\{\beta\}_{\mathrm{crit}}$. Note that at such arrival angles the left and the right derivatives of $\mathcal{L}^*(\cdot)$ will not be equal. To compute this set, we will discuss some of the important properties of the vehicle cost curves that will make the computations easier. We then discuss the properties of cost function $\mathcal{L}_\mathcal{T}(\cdot)$ for the team $\mathcal{T}$ and then discuss the computation of the optimal phase angle ψ^* and the team cost $\mathcal{L}_\mathcal{T}^*$ for the rendezvous problem.

A. Properties of vehicle cost functions

We first begin by stating a lemma that narrows down the candidate Dubins paths for the shortest path between two configurations when $d \geq 4$. The configurations where $d \geq 4$ are known as the *long path configurations*.

Lemma 1: For $d \geq 4$, the shortest path belongs to the set $\mathcal{D}_{\mathrm{long}} = \{\mathrm{LSL}, \mathrm{RSR}, \mathrm{RSL}, \mathrm{LSR}\}$.

In Fig. 3, we have shown the cost curves for the long path cases as a function of arrival angle β for a fixed initial configuration and final position. The figure at the left shows the monotonic nature of $\mathcal{L}_\Lambda(\cdot)$ for the paths $\Lambda \in \mathcal{D}_{\mathrm{long}}$ and the figure at the right shows the continuity property of $\mathcal{L}^*(\cdot)$ together with critical arrival angles. Lemma 2 characterizes the points of discontinuity in $\mathcal{L}_\Lambda(\cdot)$ for $\Lambda \in \mathcal{D}_{\mathrm{long}}$, and

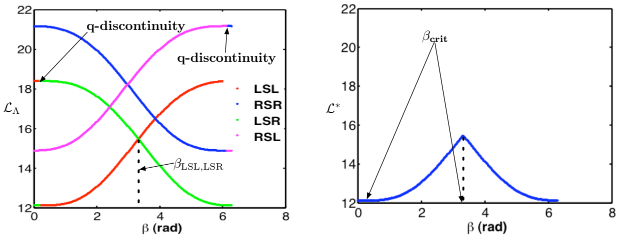

Fig. 3. Vehicle cost curves for the long path case

Lemma 3 talks about the increasing or decreasing nature of $\mathcal{L}_\Lambda(\cdot)$ for each path $\Lambda \in \mathcal{D}_{\text{long}}$.

Lemma 2: For $d \geq 4$, and for a given admissible path $\Lambda \in \mathcal{D}_{\text{long}}$, if $\mathcal{L}_\Lambda(\beta)$ is discontinuous at β, then this implies that either $t(\beta) = 0$ (or 2π) or $q(\beta) = 0$ (or 2π).

Lemma 3: $\mathcal{L}_{\text{LSL}}(\cdot), \mathcal{L}_{\text{RSL}}(\cdot)$ are monotonically increasing and $\mathcal{L}_{\text{RSR}}(\cdot), \mathcal{L}_{\text{LSR}}(\cdot)$ are monotonically decreasing functions with slope bounded in magnitude between 0 and 2 (modulo discontinuities).

In the lemmas below, $\beta_\Lambda(t, p, q)$ denotes the arrival angle for path Λ for which $\mathcal{L}_\Lambda(\beta) = t(\beta) + p(\beta) + q(\beta)$. Lemma 4, 5 show how the vehicle cost curves switch when there is discontinuity in $q(\cdot), t(\cdot)$ respectively (called as q, t-discontinuities respectively). In Fig. 3 at the left, we show an example of such discontinuties.

Lemma 4: For $d \geq 4, \beta_{\text{LSL}}(t, p, 0) = \beta_{\text{LSR}}(t, p, 0)$ and $\beta_{\text{RSL}}(t, p, 0) = \beta_{\text{RSR}}(t, p, 0)$.

Lemma 5: For $d \geq 4, \beta_{LSL}(0, p, q) = \beta_{RSL}(0, p, q)$ and $\beta_{\text{RSR}}(0, p, q) = \beta_{\text{LSR}}(0, p, q)$.

We now characterize the critical arrival angles β_{crit}, where the derivative of $\mathcal{L}^*(\cdot)$ changes its sign.

Proposition 1: The q-discontinuities for $\Lambda = \text{LSL}, \text{RSR}$ are respectively the same as those for $\Lambda = \text{LSR}, \text{RSL}$ and the t-discontinuities for $\Lambda = \text{LSL}, \text{RSR}$ are respectively the same as those for $\Lambda = \text{RSL}, \text{LSR}$. Moreover, $\mathcal{L}^*(\cdot)$ remains differentiable at t-discontinuities.

The next theorem proves the continuity property of the vehicle optimal cost curve $\mathcal{L}^*(\cdot)$.

Theorem 1: Given an initial configuration and a final location for a Dubins vehicle , the optimal cost function $\mathcal{L}^*(\cdot)$ is a continuous function of the arrival angle β if $d \geq 4$.

B. Computation of critical arrival angles

In this section, we first state two lemmas that will help us in computing the (q, t)-discontinuities.

Lemma 6: For a given Dubins admissible path $\Lambda \in \mathcal{D}_{\text{long}}$ and $d \geq 5$, $q_\Lambda(\cdot)$ is a monotonically increasing function for $\Lambda \in \{\text{LSL}, \text{RSL}\}$, and and monotonically decreasing for $\Lambda \in \{\text{RSR}, \text{LSR}\}$, modulo q-discontinuities. Moreover, $|q'_\Lambda| \leq 2$ (where defined), for $\Lambda \in \mathcal{D}_{\text{long}}$.

The above lemma will be used to compute q-discontinuities in $\mathcal{L}_\Lambda(\cdot)$ for $\Lambda \in \mathcal{D}_{\text{long}}$, in Section III-B.

Lemma 7: For a given Dubins admissible path $\Lambda \in \{\text{LSL}, \text{RSL}\}$ and $d \geq 2(1 + \sqrt{3})$, $t''_\Lambda(\beta) = 0$ iff $q(\beta) \in \{\pi/2, 3\pi/2\}$.

The above Lemma will be used to compute t-discontinuities in $\mathcal{L}_\Lambda(\cdot)$ for $\Lambda \in \mathcal{D}_{\text{long}}$, in Section III-B.

In the next proposition we upper bound the number of times the derivative of $\mathcal{L}^*(\cdot)$ can change its sign. We once again remark that the left and the right derivative may not be equal where the sign changes.

Proposition 2: The derivative of the function $\mathcal{L}^*(\cdot)$ can change its sign no more than 16 times.

We now discuss how to use Lemma 6, 7 to compute the set $\{\beta_{\text{crit}}\}$. For the remainder of this section, assume that $d \geq 2(1 + \sqrt(6))$ so that we can use Lemmas 6 and 7 and Proposition 2. Proposition 1 and Theorem 1 imply that every β_{crit} will be either a q-discontinuity or a point of intersection between curves whose derivatives are of opposite signs. Let $\hat{q_\Lambda}(\cdot)$ denote the variant of $q_\Lambda(\cdot)$ without the $\mod$ function, and similarly $\hat{t_\Lambda}(\cdot)$ denote the variant of $t_\Lambda(\cdot)$ without the $\mod$ function. For the exact expressions, we refer the reader to [2]. $\beta_{\Lambda_1,\Lambda_2}$ denotes the point of intersection between $\mathcal{L}_{\Lambda_1}(\cdot)$ and $\mathcal{L}_{\Lambda_2}(\cdot)$ for $\Lambda_1, \Lambda_2 \in \mathcal{D}_{\text{long}}$. We refer the reader to Fig 3 at the left for one such example.

$\beta_\Lambda(t, p, 0)$ can be found using Lemma 6 and finding the $2k\pi$ crossings of $\hat{q}_\Lambda(\cdot)$, $k \in \mathbb{N}$.

To find $\beta_\Lambda(0, p, q)$, we first find $\beta_\Lambda(t, p, q \in \{\pi/2, 3\pi/2\})$. From Lemma 7 we know that between two such β, the function remains convex or concave. Hence in any such interval there can be at most two such crossings where $t = 0$ or 2π. We can find the $2k\pi$ crossings of $\hat{t}_\Lambda(\cdot)$, $k \in \mathbb{N}$ now using some kind of minimization routine on the bounded interval.

Let $\{\beta_{\text{disct}}\} = \{\beta : \hat{q}_\Lambda(\beta) = 2k\pi\} \cup \{\beta : \hat{t}_\Lambda(\beta) = 2l\pi\} \cup \{0, 2\pi\}, k, l \in \mathbb{N}, \Lambda \in \mathcal{D}_{\text{long}}$. Assume that $\{\beta_{\text{disct}}\}$ is a strictly increasing sequence. Now $\{\beta_{\text{crit}}\} \subseteq \{\beta_{\text{disct}}\} \cup \{\beta_{\text{inter}}\}$, where $\{\beta_{\text{inter}}\} = \{\beta_{\text{LSL,RSR}}, \beta_{\text{LSL,LSR}}, \beta_{\text{RSR,RSL}}, \beta_{\text{LSR,RSL}}\}$. An exhaustive search of the intersection points can be done between every two successive $\beta_1, \beta_2 \in \{\beta_{\text{disct}}\}$ since each of the $\mathcal{L}_\Lambda$ is guaranteed to be smooth in any such interval for $\Lambda \in \mathcal{D}_{\text{long}}$. This can be done for example by doing minimization of $|\mathcal{L}_{\Lambda_1}(\cdot) - \mathcal{L}_{\Lambda_2}(\cdot)|$ on the interval $[\beta_1, \beta_2]$.

Finally, the set $\{\beta_{\text{crit}}\}$ can be formed using $\{\beta_{\text{disct}}\}$ and $\{\beta_{\text{inter}}\}$ by examining the derivative of $\mathcal{L}^*(\cdot)$ around each $\beta \in \{\beta_{\text{disct}}\} \cup \{\beta_{\text{inter}}\}$.

C. Properties of the team cost function

Let $\{\psi_{\text{crit},j}\}$ denote the set of phases of critical arrival angles for vehicle V_j and $\psi_{i,j}$ denote the phase (possibly more than one) at which $\mathcal{L}_i^*(\cdot), \mathcal{L}_j^*(\cdot)$ intersect. In Fig. 4, we show one instance of the cost curves for each vehicle along with that for the team along with few important phase angles. The figure at the right indicates that the team cost curve $\mathcal{L}_\mathcal{T}(\cdot)$ is continuous which holds true under some assumptions. We state this in the next theorem.

Theorem 2: Given a team of Dubins vehicles $\mathcal{T} = \{V_j\}_{j=1}^N$ with fixed initial configurations and a final location, the team cost function $\mathcal{L}_\mathcal{T}(\cdot)$ is a continuous function of the phase if $d \geq 4$ for each vehicle.

The next theorem characterizes the optimal phase angle ψ^* for the rendezvous problem.

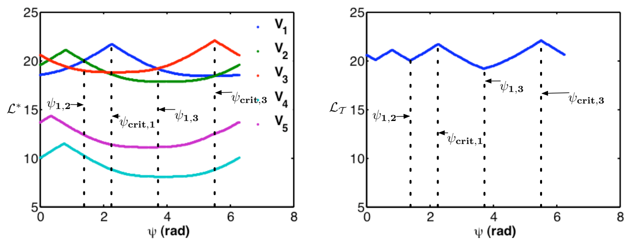

Fig. 4. Cost curves for the team members and the team

Theorem 3: The solution ψ^* to the rendezvous problem either belongs to the set $\{\psi_{\text{crit},j}\}$ for some $V_j \in \mathcal{T}$ or is a point of intersection between two vehicle optimal cost functions $\mathcal{L}_j^*, \mathcal{L}_i^*$ if $d \geq 4$ for each vehicle.

D. *Computation of optimal phase angle and team cost*

We now discuss how the results of Sections III-C can be used for solving the rendezvous problem approximately. From Theorem 2, we already know the function $\mathcal{L}_{\mathcal{T}}(\cdot)$ is continuous. We also know from Theorem 3, that ψ^* is an end point, or a point where the derivative of $\mathcal{L}_{\mathcal{T}}(\cdot)$ changes its sign. Note that we do not have any smoothness properties for $\mathcal{L}_{\mathcal{T}}(\cdot)$ beyond continuity. In Section V, a decentralized ϵ approximate algorithm that is based on golden section search is proposed that computes an approximation $\mathcal{L}_{\mathcal{T}}^{\epsilon}$ to the optimal cost $\mathcal{L}_{\mathcal{T}}^*$, up to the desired level of accuracy ϵ . The golden section search method is an iterative optimization scheme that is guaranteed to find the optimal for a continuous and unimodal function in a bounded interval up to a desired level of accuracy [17]. The scheme can compute an approximation to the optimum based on just the function evaluations without using any derivative information.

To actually achieve rendezvous, a Dubins vehicle should be able to plan a Dubins path of given length $\mathcal{L} = \mathcal{L}_{\mathcal{T}}(\psi^*) \geq \mathcal{L}_j^*(\psi^*)$. In the next section we propose a scheme of extending Dubins shortest path between two configurations to arbitrary lengths.

IV. Extension of Dubins Shortest Path

In this section we show how for any given pairs of configuration, the Dubins shortest path can be extended to arbitrary lengths.

Theorem 4: For a given initial configuration $\mathcal{C}_i = (0, 0, \alpha)$, and a final configuration $\mathcal{C}_f = (d, 0, \beta)$, it is possible to construct a Dubins path of length $\mathcal{L}(\beta) > \mathcal{L}^*(\beta)$ if $d \geq d^* = 2(1 + \sqrt{6})$.

Proof: For $d \geq d^*$, $p_\Lambda \geq 4$. Now let $\delta\mathcal{L} = \text{mod}\ (\mathcal{L}(\beta) - \mathcal{L}(\beta)^*, 2\pi)$. Extending the path length by $2\pi m, m \in \mathbb{N}$, is trivial (e.g. make m loops around a point on the original path). So let us assume that $\delta L < 2\pi$. Now first consider the case when $0 < \delta L \leq 2\pi - 4$. For this case the suggested path modification is shown in Fig. 5 at the left. We modify the linear portion of the original path (segment p). To extend the path length, we can make a left turn of length v followed by a right turn of length $2v$ and finally a left turn of length v. Let the distance between the centers of the circles C_1, C_2 be d_{12}. Then the length of curved portion is $4v$, with $v = \arcsin(d_{12}/4)$. Hence the increase in path length is given by $f(d_{12}) = 4 \arcsin(d_{12}/4) - d_{12}$. $f(\cdot)$ achieves its maximum value $f_{\max} = 2\pi - 4$ at $d_{12} = 4$ and is monotonically increasing in the interval $d_{12} \in [0, 4)$. Hence for $\delta\mathcal{L} \leq f_{\max}$ this maneuver suffices. For the case when $\delta\mathcal{L} \in (f_{\max}, 2\pi)$, the maneuver shown in Fig. 5 at the right can be used to achieve the required path length. It consists of a left turn of length $\pi/2$, followed by a straight line of length d_{14}, followed by a right turn of length π followed by a straight line of length $d_{24} = d_{14}$ followed by a left turn of length $\pi/2$. Hence to achieve the required increase in path length $\delta\mathcal{L}$, $d_{14} = d_{24} = (\delta\mathcal{L} - (2\pi - 4))/2$. ■

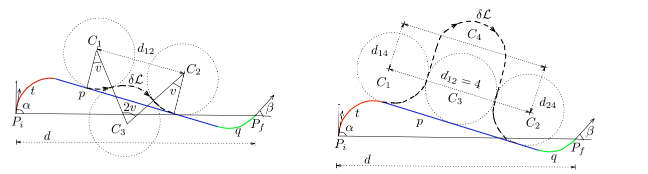

Fig. 5. Extension of Dubins shortest path

Remark 1: Note that d^* is not the tightest possible lower bound but one where extending the original path is fairly straightforward.

V. Algorithm

We now propose a decentralized algorithm (shown in Fig. 6, 8) which is an ϵ-approximate algorithm for the optimal rendezvous problem. We will call this algorithm as the Optimal Rendezvous algorithm. We first state the assumptions that are necessary for the algorithm to work correctly, then describe the algorithm in detail, followed by discussion of correctness and computational complexity of the algorithm.

A. *Assumptions*

Assumption 1: The communication between the agents is instantaneous, and the computation is fast so that the agents can be assumed to be at their initial position when the team arrives at the solution for the rendezvous problem.
The assumption guarantees that the solution computed by the algorithm is feasible for each vehicle $V_j \in \mathcal{T}$.

Assumption 2: The distance d between initial and final position for every vehicle satisfies the constraint $d \geq 7$.
This guarantees that the Dubins shortest paths can be extended to arbitrary lengths and that the results of Section III hold true.

B. *Description*

The algorithm uses a communication topology, in which each vehicle V_j receives the coordination variables from the vehicle V_{j-1}, makes changes to the variables (in function `Compute`) and then passes them along to V_{j+1} (in function `Com`). This is shown in Fig. 7 at the left for the case of 3 vehicles.

$\text{STATE}, \text{N}, \text{opt}, \text{limits}, \text{curr\_cons}, \text{last\_3\_cons}, \epsilon$ are the *coordination variables* of the algorithm which may be modified by each of the vehicles and then passed along

```
Optimal Rendezvous(T, N, ε)
  i ← 1
  STATE ← Start
  (STATE, opt, limits, curr_cons, last_3_cons) ← V_i.Init(STATE)
  while () do
     V_i.Compute(STATE, N, opt, limits, curr_cons, last_3_cons, ε)
     V_i.Com(V_{i+1}, STATE, N, opt, limits, curr_cons, last_3_cons, ε)
     i ← i + 1
     if (i > N) then
        i ← 1
  return
```

Fig. 6. Decentralized algorithm for rendezvous problem

to the next vehicle. We denote the set of coordination variables by $\mathcal{W}$. STATE stores the global state of the algorithm. N is the number of vehicles in the team $\mathcal{T}$. Each of the variables opt, curr_cons, last_3_cons stores the cost $\mathcal{L}_T(\psi)$ (denoted by field val) at a given offset angle ψ (denoted by field pt). opt is the currently known global optimal. curr_cons is the current variable on which the agents are trying to reach consensus. last_3_cons stores the last 3 consensus offset angles (last_3_cons.pts) and the corresponding values (last_3_cons.vals). The algorithm uses

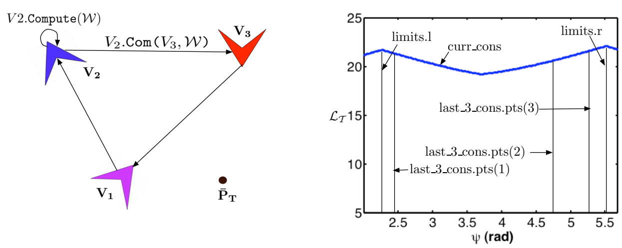

Fig. 7. Example of algorithm execution with 3 vehicles

golden section search [17] to compute approximations to local optimal iteratively in a given interval and traverses the interval $[0, 2\pi]$ in a piece-wise fashion. One such iteration is shown in Fig. 7 at the right. In any given iteration of the algorithm, all the vehicles except for a decision maker can only update the curr_cons.cost based on the currently proposed phase angle curr_cons.pt. Each vehicle V_j has two copies of coordination variables: one that it sent to vehicle V_{j+1} last time it communicated with it and a current copy that it just received from vehicle V_{j-1}. The bottleneck function `BNeck()` is a boolean function that tells a vehicle V_j if it is currently the bottleneck (and hence the decision maker) or not by comparing the 2 copies of the coordination variables available to the vehicle V_j. We denote by V_b the current bottleneck vehicle below. We now describe the function `Compute` for each vehicle in detail.

Query Step: For a given left limit (limits.l), V_b sets the STATE $=$ Query and begins the polling process to decide the right hand limit. The function `Crit_Pt`(limits) updates the right hand limit to the smallest critical phase angle ψ_{crit} (if there exists one), st, $\psi_{\text{crit}} \in \{\psi_{\text{crit,j}}\}$, and, limits.l $< \psi_{\text{crit}} \leq$ limits.r.

Initialization Step: After the querying phase, V_b initializes the golden section search by setting the last_3_cons.pts as limits.l, $(\gamma^* - 1)$limits.l $+ (2 - \gamma^*)$limits.r, limits.r (where $\gamma^* = (1 + \sqrt{5})/2$ is the golden ratio) and polling for team

```
Compute(STATE, N, opt, limits, curr_cons, last_3_cons, ε)
if (BNeck() ∧ STATE = Query) then
   (curr_cons, last_3_cons) ← Init(curr_cons, last_3_cons)
   STATE ← Update_State()
else if (STATE = Query) then
   limits.r ← Crit_Pt(limits.l)
else if (BNeck() ∧ STATE = Init) then
   (curr_cons, last_3_cons) ← Init(curr_cons, last_3_cons)
   STATE ← Update_State()
else if (BNeck() ∧¬ Term(last_3_cons.pts, ε)) then
   last_3_cons ← Update_Bracket(last_3_cons, curr_cons)
   curr_cons.pt ← Probe_Pt(last_3_cons.pts)
else if (BNeck() ∧ Term(last_3_cons.pts, ε)) then
   opt ← Update_Opt(curr_cons)
   limits ← Update_Lims(limits)
   STATE ← Update_State()
else if (STATE = Optimal) then
   Plan_Path(opt)
   if (BNeck()) then
      break
else
   curr_cons.val ← max(Cost(curr_cons.pt), curr_cons.val)
return
```

Fig. 8. `Compute` function for each vehicle

cost $\mathcal{L}_T$ at these 3 points. Finally the vehicle proposes a new probe point before changing the STATE to Negotiate.

Search Step: After initialization and polling for $\mathcal{L}_T$ at the probe point curr_cons.pt, V_b updates the brackets in `Update_Bracket(curr_cons, last_3_cons)`. After this V_b proposes a new phase angle curr_cons.pt within the current bracketing interval last_3_cons in `Probe_Pt(last_3_cons)` using the golden ratio γ^*.

Update Step: At every search step V_b checks if the termination criteria is met in the function `Term`(last_3_cons.pts, ϵ). The termination criteria is based on the size of the bracketing interval and is given by last_3_cons.pts(3) $-$ last_3_cons.pts(1) $< \epsilon/2$. If the termination criteria is satisfied, V_b updates the global optimal opt in `Update_Opt(curr_cons)`. The limits are updated in `Update_Lims(limits)` according to the rule limits.l $\leftarrow$ limits.r, limits.r $\leftarrow 2\pi$. Finally based on the updated limits, V_b updates the STATE to Query or Optimal in `Update_State()`.

Path-Planning Step: Once the global optimal opt is found, every vehicle plans a path of length $\mathcal{L}_{\mathcal{T}}^{\epsilon} =$ opt.val with the phase of arrival angle $\psi^{\epsilon} =$ opt.pt and follows the path. V_b breaks from the infinite loop to stop any more inter vehicle communications.

C. *Correctness and Complexity*

We first state an important result regarding the correctness of the algorithm for the rendezvous problem.

Theorem 5: The Optimal Rendezvous algorithm is an ϵ-approximate algorithm for the rendezvous problem.

We now discuss the time and space complexity of the algorithm (depending on approximation ϵ) for the single vehicle case, and then consider the multi vehicle case.

Lemma 8: The Optimal Rendezvous algorithm has $O(\log(1/\epsilon))$ time complexity in ϵ when $N = 1$.

Next, we consider the communication complexity of the algorithm for multi-vehicle case when there is additional cost involved because of inter-vehicle communications and decentralized nature of the algorithm.

Theorem 6: The Optimal Rendezvous algorithm has communication complexity $O(N^2)$ in N and $O(\log(1/\epsilon))$ in ϵ.

Theorem 7: The Optimal Rendezvous algorithm has $O(1)$ space complexity.

VI. Simulations

In this section we present some simulation results for rendezvous problem, using our algorithm. We consider an example with 5 Dubins vehicles. The initial configuration of vehicles is: $\bar{\mathcal{C}}_{0,1} = ((3,3),\pi/3)$, $\bar{\mathcal{C}}_{0,2} = ((3,6),\pi/2)$, $\bar{\mathcal{C}}_{0,3} = ((3,14),5\pi/4)$, $\bar{\mathcal{C}}_{0,4} = ((27,6),2\pi/3)$, $\bar{\mathcal{C}}_{0,5} = ((27,14),\pi/9)$. The target point is $\bar{P}_T = (20,10)$. Fig. 9 shows the cost curves for each vehicle $\mathcal{L}_j(\cdot), j \in \{1,2,3,4,5\}$ and the team cost curve $\mathcal{L}_{\mathcal{T}}(\cdot)$. Using the algorithm described in Section V with $\epsilon = 0.0001$, the approximations to optimal phase and the optimal cost for the team are found to be $\psi^\epsilon = 3.7008$ radians, $\mathcal{L}_{\mathcal{T}}^\epsilon = 19.1840$. Fig. 10 shows the Dubins paths planned by each vehicle to achieve rendezvous at $\bar{P}_T$. Since vehicles V_4, V_5 were much closer to $\bar{P}_T$, they loop around their initial position to achieve the required path length $\mathcal{L}_{\mathcal{T}}^\epsilon$.

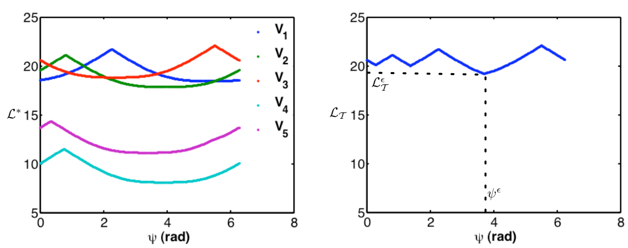

Fig. 9. Vehicle and team cost curves for the 5-vehicle example

VII. Conclusions

In this paper, we have considered the problem of rendezvous of a team of Dubins vehicles at a given destination point in minimum possible time, with the constraint that the arrival angles of successive team members are equally spaced at the destination. We have proposed a decentralized algorithm that solves the problem approximately in finite time up to desired level of accuracy. The proposed algorithm has quadratic communication complexity and constant space complexity in the size of the team. The correctness of the algorithm has been proved by establishing some facts for a single Dubins vehicle. Efforts are currently underway to examine the performance of the algorithm for rendezvous of a team of air vehicles under the presence of wind disturbances. We are also investigating possible modifications to the algorithm to make it robust with respect to the initial configuration of the vehicles.

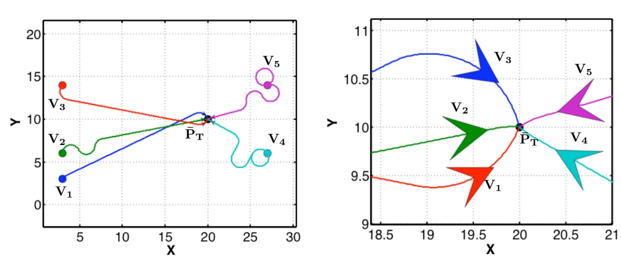

Fig. 10. Dubins paths for rendezvous of 5 vehicles at $\bar{P}_T$

VIII. Acknowledgements

We are grateful to C. Ebenbauer, N. Sabaghi and members of the ARES Lab at LIDS, MIT for fruitful discussions. The research leading to this work was supported by the National Science Foundation, through grants #0325716, #0705451, and #0705453. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the supporting organization.

References

[1] 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(3):497–516, 1957.

[2] Andrei M. Shkel and Vladimir J. Lumelsky. Classification of the dubins set. *Robotics and Autonomous Systems*, 34(4):179–202, 2001.

[3] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Transactions on Robotics and Automation*, 15(5):818–828, 1999.

[4] J. Lin, A.S. Morse, and B.D.O Anderson. The multi-agent rendezvous problem. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, volume 2, pages 1508–1513, 2003.

[5] J. Lin, A.S. Morse, and B.D.O Anderson. The multi-agent rendezvous problem- the asynchronous case. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, volume 2, pages 1926–1931, 2004.

[6] A. Jadbabaie, J. Lin, and A. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48:988– 1001, 2003.

[7] J. Cortes, S. Martinez, and F. Bullo. Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control*, 51(8):1289–1298, 2006.

[8] A. Ganguli, J. Cortes, and F. Bullo. Multirobot rendezvous with visibility sensors in nonconvex environments. *IEEE Transactions on Robotics Automation and Control*, November 2006. Submitted.

[9] P. Meschler. Time-optimal rendezvous strategies. *IEEE Transactions on Automatic Control*, 8(4):279–283, 1963.

[10] A. Tiwari and J. Fung. Polyhedral cone invariance applied to rendezvous of multiple agents. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, volume 1, pages 165–170, 2004.

[11] J. Marshall, M.E. Broucke, and B. A. Francis. Formations of vehicles in cyclic pursuit. *IEEE Transactions on Automatic Control*, 49(11):1963–1974, 2004.

[12] J. Marshall, M.E. Broucke, and B. A. Francis. A hierarchical cyclic pursuit scheme for vehicle networks. *Automatica*, 41(6):1045–1053, 2004.

[13] M. Pavone. and E. Frazzoli. Decentralized policies for geometric pattern formation and path coverage. *ASME Journal on Dynamic Systems, Measurement, and Control*, 2007. To appear.

[14] Dimos V. Dimarogonas and Kostas J. Kyriakopoulos1. On the rendezvous problem for multiple nonholonomic agents. *IEEE Transactions on Automatic Control*, 52:916–922, 2007.

[15] H. G. Tanner, A. Jadbabaie, and G. J. Pappas. Flocking in fixed and switching networks. *IEEE Transactions on Automatic Control*, 52:863–868, 2007.

[16] Timothy W. McLain and Randal W. Beard. Coordination variables, coordination functions, and cooperative timing missions. *Journal of Guidance, Control and Dynamics*, 28(1):150–161, 2005.

[17] W. H. Press. *Numerical Recipes in C.* Cambridge Universtiy Press, 2007.

Appendix

A. Proof of Lemma 1, 2, 4, 5 (Sketch)

The lemmas follow from the expressions for path lengths derived in [2].

B. Proof of Lemma 3

Using expressions for path lengths from [2], $\mathcal{L}'_{\text{LSL}} = (1 - \cos(q))$, $\mathcal{L}'_{\text{RSR}} = -(1 - \cos(q))$, $\mathcal{L}'_{\text{LSR}} = -(1 - \cos(q))$ and $\mathcal{L}'_{\text{RSL}} = (1 - \cos(q))$ (modulo discontinuities). This also shows that the slope is bounded between -2 and 2 (modulo discontinuities).

C. Proof of Proposition 1

The proof follows from Lemmas 3, 4, 5.

D. Proof of Theorem 1

We know from Lemma 3 that each of the $\mathcal{L}_\Lambda(\cdot)$ is monotonic and continuous function (modulo discontinuities due to modulus function). Lemmas 2, 4, 5 imply that the points of discontinuity will result only in change of the optimal path Λ^* but $\mathcal{L}^*(\beta)$ will still remain continuous at such points. Moreover, intersection of any two curves $\mathcal{L}_{\Lambda_1}, \mathcal{L}_{\Lambda_2}$, $\Lambda_1, \Lambda_2 \in \mathcal{D}_{\text{long}}$ will only affect the derivative but will not affect continuity of $\mathcal{L}^*(\cdot)$.

E. Proof of Lemma 6

Using expressions for path lengths from [2], $q'_{\text{LSL}} = 1 + \sin(q)/p$, $q'_{\text{RSR}} = -(1 + \sin(q)/p)$, $q'_{\text{LSR}} = -(1 + \sin(q)/p)$ and $q'_{\text{RSL}} = 1 + \sin(q)/p$, modulo q-discontinuities. Also for $d \geq 5$ and for $\Lambda \in \mathcal{D}_{\text{long}}$, $p \geq 1$.

F. Proof of Lemma 7

Using expressions for path lengths from [2] for $\Lambda \in \{\text{LSL}, \text{RSR}\}$, $t''_\Lambda = -\cos(q)(p + 2\sin(q))/p^2$ (modulo t-discontinuities). For $d \geq 2(1 + \sqrt{3})$, $t'' = 0 \Rightarrow q \in \{\pi/2, 3\pi/2\}$.

G. Proof of Proposition 2

The derivative of $\mathcal{L}^*(\cdot)$ changes its sign at an arrival angle β, if β is a q-discontinuity or vehicle cost curves of opposite monotonicity (e.g $\mathcal{L}_{\text{LSL}}(\cdot)$, $\mathcal{L}_{\text{RSR}}(\cdot)$) intersect at β. The number of q-discontinuities for a given path $\Lambda \in \mathcal{D}_{\text{long}}$ at which $\mathcal{L}_\Lambda(\cdot)$ is bounded by 2 (using Lemma 6 and the fact that $\beta \in [0, 2\pi]$). This together with Lemma 4 implies that there can be no more than 4 q-discontinuities in $\mathcal{L}^*(\cdot)$. As a result of the q-discontiuities, functions of opposite monotonicity can intersect more than once but still no more than 4 times. The points of t-discontinuities do not affect the answer since we are looking for sign change in derivative. There can be 4 such pairs of functions of opposite monotonicity. Hence, the derivative of $\mathcal{L}^*(\cdot)$ can change its sign no more than 16 times.

H. Proof of Theorem 2

The proof follows from the definition of $\mathcal{L}_\mathcal{T}(\cdot)$ and the fact that the vehicle optimal cost function $\mathcal{L}^*_j(\cdot)$ is continuous for each vehicle $V_j \in \mathcal{T}$ for $d \geq 4$.

I. Proof of Theorem 3

The proof follows from the fact that a continuous function has its global extremal at a point where the derivative of the function changes its sign or an end point.

J. Proof of Theorem 5

Consider the `Compute` function. Let the current interval for golden section search be $\text{last\_3\_cons.pts}(1)$, $\text{last\_3\_cons.pts}(3)$. The local optimal search terminates when $\text{last\_3\_cons.pts}(3) - \text{last\_3\_cons.pts}(1) < \epsilon/2$. We also know from Lemma 3 that the slope is bounded between -2 and 2. Hence the global optimal computed by the algorithm $\mathcal{L}^\epsilon_\mathcal{T} = \text{opt.val}$ satisfies the equation $\mathcal{L}^\epsilon_\mathcal{T} - \mathcal{L}^*_\mathcal{T} \leq \epsilon$.

K. Proof of Lemma 8

For the case $N = 1$, there is no inter-vehicle communication involved in the Optimal Rendezvous algorithm and the vehicle remains the bottleneck at all times. For a given interval limits, and tolerance ϵ, the golden section search will terminate when $\text{last\_3\_cons.pts}(3) - \text{last\_3\_cons.pts}(1) < \epsilon/2$. After m iterations of the golden section search, $\text{last\_3\_cons.pts}(3) - \text{last\_3\_cons.pts}(1) = (\gamma^* - 1)^m(\text{limits.r} - \text{limits.l})$. Hence, the number of iterations required for the given tolerance will be $m = \log(\epsilon/(2(\text{limits.r} - \text{limits.l})))/\log(\gamma^* - 1)$. Also note that $\text{limits.r} - \text{limits.l} \leq 2\pi$. Hence, $m \leq \log(\epsilon/(4\pi))/\log(\gamma^* - 1)$.

We also know that the number of critical phase angles cannot be more than 16 (since number of critical arrival angles is bounded by 16, using Theorem 2). Hence in the worst case the number of iterations taken by Optimal Rendezvous algorithm to terminate will be $(16+1)\log(\epsilon/(4\pi))/\log(\gamma^* - 1)$.

L. Proof of Theorem 6

For the multi-vehicle case ($N > 1$), the algorithm first polls the team members for deciding the interval (limits) to search for local optimal. The left end of the interval limits.l is always decided a priori without polling. The right end is guaranteed to be found in at most $2N$ iterations of the algorithm. The bracketing interval (and corresponding costs) of golden section search last_3_cons will be updated in at most every $2N$ iterations. The number of intervals where local optimal of $\mathcal{L}_\mathcal{T}(\cdot)$ is searched, can be no more than $17N$. Hence, in the worst case, the number of messages communicated within the team to compute the required approximation $\mathcal{L}^\epsilon_\mathcal{T} = \text{opt.val}$ (to the global optimal $\mathcal{L}^*_\mathcal{T}$) is given by $m = 17N(2N + 2N\log(\epsilon/(4\pi))/\log(\gamma^* - 1))$.

M. Proof of Theorem 7

The information communicated between successive team members $V_j, V_{j+1} \in \mathcal{T}$ is the set of *coordination variables* $\mathcal{W} = \{\text{STATE}, \text{N}, \text{opt}, \text{limits}, \text{curr\_cons}, \text{last\_3\_cons}, \epsilon\}$. This is the only information that each vehicle needs to store about the team $\mathcal{T}$. For every vehicle $V_j \in \mathcal{T}$, the space requirements for storing a local copy of these variables remain constant and independent of the team size.