Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Distributed minimum time servicing for a team of Dubins vehicles

Giuseppe Notarstefano   Paola Pedone

Abstract— **In this paper we study the following problem for a team of Dubins vehicles, i.e. nonholonomic vehicles moving at constant longitudinal speed along planar paths with bounded curvature. Given the initial configurations of the vehicles, find the point in the plane that minimizes the time to be reached by all vehicles. We call it *minimum-time servicing problem*. We show that this problem can be approximated by an abstract linear program, namely a generalized version of linear programming, that can be solved in a distributed way over a network. We provide a control and communication law for a wireless network of Dubins vehicles to compute and reach the *minimum-time servicing point* while maintaining the network connected.**

I. Introduction

Motion coordination for mobile robots has received great attention in the last years. In order to capture the high level nature of the problem, simplified models for the vehicle dynamics have been chosen. Widely used dynamics are single and double integrators. Although very simple, such models give rise to challenging problems while capturing many interesting features of the real scenario. A necessary feature for real vehicles to fit in single and double integrator dynamic models is the capacity of hovering. An interesting scenario of motion coordination is the team coordination of Uninhabited Aerial Vehicles (UAVs) for diverse applications as search and rescue operations in hostile environments or monitoring and surveillance of protect areas. If we want to deal with UAVs, we need a model that captures two important features: i) vehicles cannot hover, ii) they have a minimum turning radius. Dubins model takes into account these two constraints. It is basically a nonholonomic vehicle in the plane that moves at constant speed and with a lower bound on the turning radius. In this paper we deal with the following problem. We search for the point that can be reached in minimum time by all the vehicles and study how the agents may compute this point in a distributed way. We call this problem "minimum-time servicing problem". One can imagine, in fact, a scenario in which a team of UAVs is operating in a general configuration and needs to get a service (e.g. fueling) from a "service vehicle". We want the UAVs to compute in a distributed way the position that the station must occupy in order to minimize the time to be reached by the all agents. The problem we want to deal with is strongly related to the rendezvous problem for a network of mobile agents that can hover. In fact, if vehicles can stop, the minimum time servicing point is also the minimum time rendezvous point, since we may imagine that each vehicle simply stops once it has reached that position.

The rendezvous problem has been introduced in [1] for a network of single integrators and extended to various synchronous and asynchronous stop-and-go strategies in [2]. After that, other control and communication strategies have been proposed to achieve the desired task. Recently the problem was studied in the case of measurement noise on the position of the neighboring agents [3]. The problem of minimum time rendezvous was studied in [4] where a control and communication law was proposed. The law is based on a distributed algorithm for the computation of the circumcenter of a set of points in the plane.

Motion coordination problems for UAVs modeled as Dubins vehicles have received considerable attention recently. In [5] authors study the so called Traveling Salesperson Problem (TSP) for Dubins vehicles. That is, design a closed tour through a set of given points for a team of Dubins vehicles. In [6] the same authors have studied a coverage problem, that is to minimize the traveling time from any vehicle to any point in the operating region. Variations of these problems may be found in references therein. Another relevant problem for teams of Dubins vehicles is decentralized collision avoidance that has strong implications in air traffic management. A decentralized strategy, based on a hybrid control formalism, was proposed in [7]. In this paper we do not consider the problem of avoiding collisions between agents.

The main contributions of the paper are as follows. We introduce the minimum time servicing problem, that is, the problem of finding the point in the plane that can be reached in minimum time by a team of Dubins vehicles from given configurations. We characterize the solution in terms of the reachable sets of the Dubins vehicles. We show that the solution can be found by solving the following problem. Find the smallest intersection of the family of reachable sets parametrized by the time to reach the points in the plane. Then, we show that an approximated version of this problem is an abstract linear program, namely a generalized version of linear programming, that can be solved in a distributed way by using an algorithm introduced in [8]. Finally, we design a control and communication law for a wireless network of Dubins vehicles in order to compute and reach the minimum-time servicing point. We develop a strategy to maintain connectivity of a subgraph of the communication graph. This strategy is necessary for the control and communication law to work correctly.

The paper is organized as follows. In Section II we describe the connection between abstract linear programming and Helly-type theorems. In Section III we introduce the Dubins model and the network of Dubins vehicles. Then, we set up the minimum-time servicing problem. In Section IV we show that the minimum time servicing problem may be approximated by a suitable abstract linear program (through a parametrized Helly system) and provide a distributed algorithm to solve it. Finally, in Section V we design a control and communication law to compute and reach the minimum-time servicing point while maintaining the network connected. Due to the lack of space we postpone the

Giuseppe Notarstefano and Paola Pedone are with the Department of Engineering, University of Lecce, Via per Monteroni, 73100 Lecce, Italy, {giuseppe.notarstefano, paola.pedone}@unile.it

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

proofs to a forthcoming technical report.

II. PRELIMINARIES

In this section we discuss a class of optimization problems and a class of properties that will play an important role in the distributed computation of the minimum time servicing point.

A. Abstract linear programming

We consider optimization problems specified by a pair (H, ω), where H is a finite set with cardinality $\operatorname{card} H$, and $\omega : 2^H \to \Omega$ is a function with values in a linearly ordered set $(\Omega, \leq)$; we assume that Ω has a minimum value $-\infty$. The elements of H are called *constraints*, and for $G \subset H$, $\omega(G)$ is called the *value* of G. Intuitively, $\omega(G)$ is the smallest value attainable by a certain objective function while satisfying the constraints of G. An optimization problem of this sort is called *abstract linear program* (ALP) (or generalized linear program (GLP)) if the following two axioms are satisfied:

(i) *Monotonicity*: if $F \subset G \subset H$, then $\omega(F) \leq \omega(G)$;
(ii) *Locality*: if $F \subset G \subset H$ with $-\infty < \omega(F) = \omega(G)$, then, for all $h \in H$,

$$\omega(G) < \omega(G \cup \{h\}) \implies w(F) < w(F \cup \{h\}).$$

A set $B \subset H$ is *minimal* if $\omega(B) > \omega(B')$ for all proper subsets B' of B. A minimal set B with $-\infty < \omega(B)$ is a *basis*. Given $G \subset H$, a *basis of* G is a minimal subset $B \subset G$, such that $-\infty < \omega(B) = \omega(G)$. A constraint h is said to be *violated* by G, if $\omega(G) < \omega(G \cup \{h\})$.

The *solution* of an abstract linear program (H, ω) is a minimal set $B_H \subset H$ with the property that $\omega(B_H) = \omega(H)$. The *combinatorial dimension* δ of (H, ω) is the maximum cardinality of any basis.

An ALP algorithm takes an ALP problem (H, ω) and returns a basis B for H. In [9], Matoušek, Sharir and Welzl provided a randomized ALP algorithm which uses two primitive operations. A *basis computation* takes a family G of at most $\delta + 1$ constraints and finds a basis for G. A *violation test* takes a basis B and a constraint h, and returns true if B is a basis of $B \cup \{h\}$. Assuming that the time required for a violation test and for a basis computation are polynomial in δ, their algorithm runs in expected time linear in the number of constraints n and subexponential in d.

In the scenarios we are interested in, δ will always be much smaller than n.

Remark 2.1: Beyond linear programming, numerous geometric optimization problems can be cast as abstract linear programs. Examples include computing the smallest enclosing ball and annulus of a set of points, the largest ellipsoid in a polytope, the smallest enclosing orthotope, the distance between convex polytopes and others. More examples are discussed in [9], [10] and references therein. □

B. Helly-type theorems and abstract linear programs

In this subsection we describe the class of Helly-type theorems which play an important role in combinatorial geometry. Then, we recall the important connection between Helly-type theorems and abstract linear programs proved by Amenta in [11].

We start stating Helly's theorem.

Theorem*(Helly's Theorem)* Let K be a family of at least $d+1$ convex sets in E^d, and assume K is finite or that every member of K is compact. If every $d+1$ members of K have a point in common, then there is a point common to all the members of K. □

Theorems with the same logical structure, for objects other than convex sets, for properties other than intersection, or for cases in which $d+1$ is replaced by some other constant k are called *Helly-type theorems*. A version of Helly-type theorem is the following.

Theorem*(Helly-type Theorem)* Let C be a family of objects, and $\mathcal{P}$ a predicate on subsets of C. There is a constant $k \in \mathbb{N}$ such that for all finite $H \subset C$, H has property $\mathcal{P}$ if and only if every $B \subset H$ with $\operatorname{card} B \leq k$ has property $\mathcal{P}$. □

The constant k is called the *Helly number* of $(H, \mathcal{P})$ and a pair $(C, \mathcal{P})$ satisfying a Helly-type theorem is called a *Helly system*.

In [11] Amenta has established that a parametrized family of Helly systems gives rise, under suitable assumptions, to an ALP. From now on, we assume that the property $\mathcal{P}$ is the intersection $\cap$ as in the original Helly theorem. Also, we let X be a set and C be the family of subsets of X.

Let $\mathcal{I} \subset \mathbb{R}$ be an interval. A nested family $\bar{h}$ is defined as $\{h_\lambda \mid \lambda \in \mathcal{I}\}$, where $h_\lambda \subset X$ for each λ, and $h_\alpha \subset h_\beta$ for $\alpha < \beta$. Now, consider a collection $\bar{H}$ of nested families $\bar{h}$, all indexed by the same parameter λ. If $\bar{H}$ has the property that $(H_\lambda, \mathcal{P})$ is a Helly system of dimension k, for every $\lambda \in \mathcal{I}$, $(\bar{H}, \mathcal{P})$ is a *parametrized Helly system* with Helly number k. Notice that if H_{λ_1} has an empty intersection (i.e. does not satisfy property $\mathcal{P}$), then the same holds for H_{λ_2}, for $\lambda_2 < \lambda_1$. Viceversa, if H_{λ_1} has a nonempty intersection, then the same holds for H_{λ_2}, for $\lambda_2 > \lambda_1$.

A parameterized Helly system may be turned into a constrained optimization problem by considering the following objective function. For $\bar{G} \subset \bar{H}$, let $\omega(\bar{G})$ be the minimum value λ^* such that G_{λ^*} intersects and $+\infty$ if it does not intersect at any λ. If furthermore for any $\bar{G} \subset \bar{H}$ the intersection at λ^* is a unique point, then the optimization problem $(\bar{H}, \omega)$ can be shown to be an abstract linear program. The result is stated formally in the next theorem.

Theorem 2.2 (Helly type theorems and ALP [11]): Let $(\bar{H}, \mathcal{P})$ be a parametrized Helly system with Helly number k such that, for all $\bar{G} \subset \bar{H}$, $\lambda^* = \omega(\bar{G})$ exists and the intersection $\cap G_{\lambda^*}$ (for λ^* finite) is a unique point. Then $(\bar{H}, \omega)$ is an abstract linear program of combinatorial dimension k. □

This theorem will pay a key role in the solution of the minimum time servicing problem.

III. SCENARIO AND PROBLEM SET UP

In this section we first introduce the scenario of work. That is, we describe the Dubins vehicle dynamics with its reachability properties and the mathematical model for a network of Dubins vehicles. Then we define the minimum time servicing problem.

A. The Dubins vehicle and its reachable set

A Dubins vehicle is described by a configuration $g \in SE(2)$, where $SE(2)$ is the special Euclidean group of dimension 2, that may move with unit longitudinal speed along $\mathcal{C}^2$ curves that are twice differentiable and whose curvature is bounded above by $\frac{1}{\rho}$, with $\rho > 0$ being the minimum turning radius. Such curves are also known as Dubins paths. Given a Dubins path $t \mapsto \gamma(t)$, $t \in [0, T]$, with $\|\gamma'(t)\| = 1$, the evolution $g_\gamma(t) \in SE(2)$ is given by

$$g_\gamma(t) = (\gamma(t), \arctan(\gamma'(t))).$$

In this paper we will restrict to Dubins paths that minimize the time (equivalently the length) to reach a final configuration from an initial one. In [12] it was shown that optimal paths are the combination of straight lines and/or arcs of circle, so that the curvature assumes only the values $\pm\frac{1}{\rho}$ or zero. From now on we will assume that the vehicle evolves only along optimal Dubins paths so that the control input to decide is the path to follow in the given time interval. Here we are tacitly assuming that there is a lower level control feedback that is able to perfectly track such path. We use the following notation. Given two instants t_1 and t_2, with $t_1 \leq t_2$, and two configurations $g(t_1)$ and $g(t_2)$, we have the following update law, for $t \in [t_1, t_2]$,

$$g(t) = \phi(g(t_1), u(g(t_1), g(t_2), t)), \tag{1}$$

where $g(t)$ is the configuration on the optimal Dubins path that connects $g(t_1)$ to $g(t_2)$ reached at the instant t and $u(g(t_1), g(t_2), t)$ the control input needed to follow the path. Consistently, $g(t) = \phi(g(t_1), u(g(t_1), g(t), t))$ and $g(t)g(t_1)^{-1} = \phi(\mathbf{I}, u(g(t_1), g(t), t))$, where $\mathbf{I} \in SE(2)$ is the identity element in the group and g^{-1} is the inverse of g according to the inverse operation defined in the group. Here t_1, t and t_2 may be either continuous time instants (points of the positive real line) or discrete time ones (natural numbers).

Next, we describe the (planar) reachable set of Dubins vehicles, that is, the set of points in the plane that can be reached, starting from an initial configuration, in time $t \geq 0$ (or, equivalently, by paths of length less than or equal to $l \geq 0$). The problem was originally solved in [13]. The paper provides a detailed analysis of the all possible shapes of the reachable set wavefronts depending on the length of the path. In [6] the reachable set of Dubins vehicles was studied to solve a coverage problem for teams of UAVs.

Given a configuration $g \in SE(2)$ and a point $q \in \mathbb{R}^2$, we let $L_\rho(g, q)$ be the length of the shortest Dubins path (with minimum turning radius ρ) from the initial configuration g to the point q. Given $t \geq 0$ and a configuration $g \in SE(2)$, let $\mathcal{R}_g(t)$ denote the reachable set of the Dubins vehicle in time t starting from state g, i.e.

$$\mathcal{R}_g(t) = \{q \in \mathbb{R}^2 \mid L_\rho(g, q) \leq t\}.$$

Reachable sets for Dubins vehicles are shown in Figure 1.

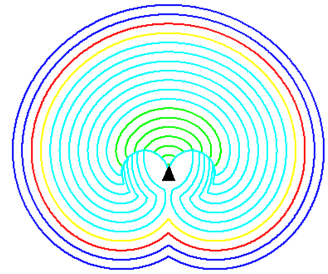

Fig. 1. Reachable sets for the Dubins vehicle for increasing values of t

The following important properties may be proven for the reachable sets.

P1 $\mathcal{R}_g(t)$ is a monotonic function in t, i.e., $\mathcal{R}_g(t_1) \subseteq \mathcal{R}_g(t_2)$ for $t_1 \leq t_2$.

P2 there exists a constant $k_0 > 0$ such that $\mathcal{R}_g(t)$ is a simply connected set for all $t \geq k_0\rho$.

B. Network of Dubins vehicles

We consider a synchronous network of mobile robots (robotic network) adopting a specific version of the formal model introduced in [14]. We have a network of Dubins vehicles (physical agents) labeled by a set of identifiers $I = \{1, \ldots, n\}$, $n \in \mathbb{N}$. The agents live in the state space $SE(2)$, move according to the Dubins model introduced above and communicate according to a *communication edge map* $E_{\text{cmm}} : SE(2)^n \to 2^{I \times I}$ with the following property: an edge (i, j) belongs to $E_{\text{cmm}}((g^{[1]}, \ldots, g^{[n]}))$, $(g^{[1]}, \ldots, g^{[n]}) \in SE(2)^n$, if and only if agents i and j (with configuration respectively $g^{[i]}$ and $g^{[j]}$) can communicate. We denote $G = (I, E_{\text{cmm}})$ the associated undirected communication graph. In this paper we use the disk-graph as communication graph. For $(g_1, \ldots, g_n) = ((p_1, \theta_1), \ldots, (p_n, \theta_n)) \in SE(2)^n$, with $p_i \in \mathbb{R}^2$ and $\theta_i \in SO(2)$, $i \in \{1, \ldots, n\}$, the pair (i, j) (associated to the configurations g_i and g_j) is an edge in $G_{\text{disk}}(g_1, \ldots, g_n)$ if and only if $\|p_i - p_j\| \leq r_{\text{cmm}}$, where $r_{\text{cmm}} > 0$ is the communication radius. The robotic network evolves according to a discrete-time communication and motion model. For all $i \in I$, to the ith physical agent corresponds a processor, labeled i, that performs the following actions. First, at each communication round the ith processor sends to each of its outgoing neighbors in the communication graph a *message* (possibly the `null` message) computed by applying a *message-generation function* (msg) to the current values of $g^{[i]}$ and $w^{[i]}$, where $w^{[i]}$ is its *logical state* (namely a set of variables stored in its memory). After a negligible period of time, the ith processor updates the value of its logical state $w^{[i]}$ by applying a *state-transition function* (stf) to the current value of $w^{[i]}$, and to the messages received at time t. Between communication instants, the motion of the ith agent is determined by applying a *control function* (*ctl*) to the current value of $g^{[i]}$, and the current value of $w^{[i]}$. The three functions (message-generation, state-transition and control) equipped with the sets of messages and logical variables define a *control and communication law* $\mathcal{CC}$. This idea is formalized as follows.

Definition 3.1 (Evolution of a robotic network): Let $\mathcal{S}$ be a robotic network and $\mathcal{CC}$ be a control and communication law for $\mathcal{S}$. Let also W be the set of logical states, $W_0 \subset W$ the subset allowable initial values and M the set of messages. The *evolution* of $(\mathcal{S}, \mathcal{CC})$ from initial conditions $g_0^{[i]} \in SE(2)$ and $w_0^{[i]} \in W_0$, $i \in I$, is the set of curves $g^{[i]} : \mathbb{N} \to SE(2)$ and $w^{[i]} : \mathbb{N} \to W$, $i \in I$, satisfying

$$g^{[i]}(t+1) = \phi\big(g^{[i]}(t),\, \text{ctl}(g^{[i]}(t), w^{[i]}(t), y^{[i]}(t))\big),$$

where, for $i \in I$,

$$w^{[i]}(t) = \text{stf}(w^{[i]}(t-1), y^{[i]}(t))\,,$$

with the conventions that $g^{[i]}(0) = g_0^{[i]}$ and $w^{[i]}(-1) = w_0^{[i]}$. Here, the function $y^{[i]} : \mathbb{N} \to M^n$ (describing the messages received by agent i) has components $y_j^{[i]}(t) = \text{msg}(g^{[j]}(t), w^{[j]}(t), i)$, if $(i,j) \in E_{\text{cmm}}$ and `null` otherwise.

C. Minimum-time servicing: problem set up and centralized solution

Here we provide a formal definition for the minimum-time servicing problem and describe the solution in the centralized case.

Informally, the *minimum-time servicing problem* is the following. Given an initial configuration for the Dubins vehicles, find the point in the plane that minimizes the time to be reached by all the agents starting from the given configuration. We call such point *minimum-time servicing point*. Equivalently, find the point in the plane that minimizes the Dubins distance from each configuration to this point.

More formally we have the following definition.

Definition 3.2 (Minimum-time servicing problem): Let $(g_0^{[1]}, \dots, g_0^{[n]}) \in SE(2)^n$ be a given initial configuration for the network of Dubins vehicles. The minimum-time servicing problem is given by

$$\min_{q \in \mathbb{R}^2} \max_{i \in \{1,\dots,n\}} L_\rho(g_0^{[i]}, q). \tag{2}$$

The minimum-time servicing point is the point $q^* \in \mathbb{R}^2$ that solves the above minimization problem. □

The following lemma provides an equivalent formulation of the problem, which highlights a way to solve the problem in a distributed fashion, and proves existence of the solution.

Lemma 3.3: Let $(g_0^{[1]}, \dots, g_0^{[n]}) \in SE(2)^n$ be an arbitrary initial configuration. The minimum-time servicing problem is equivalent to

$$\begin{aligned} &\text{minimize}_{t \geq 0}\ t \\ &\qquad \text{subj. to } \cap_{i=1}^n \mathcal{R}_{g_0^{[i]}}(t) \neq \emptyset. \end{aligned} \tag{3}$$

A solution of the problem exists. □

IV. Distributed computation of the minimum-time servicing point

In this section we describe how we may compute an approximation of the minimum-time servicing point in a distributed way.

A. Approximating the minimum-time servicing problem by an abstract linear program

We construct an approximated version of the minimum-time servicing problem. We show that it is in fact an abstract linear program so that it can solved in a distributed way over the network.

We start doing the following assumption.

Assumption A0.
Let $(g_0^{[1]}, \dots, g_0^{[n]}) = ((p_0^{[1]}, \theta_0^{[1]}), \dots, (p_0^{[n]}, \theta_0^{[n]}) \in SE(2)^n$, with $p_0^{[i]} \in \mathbb{R}^2$ and $\theta_0^{[i]} \in SO(2)$ for $i \in \{1, \dots, n\}$, be given. Then

- $p_0^{[i]} \neq p_0^{[j]}$ for any $i \neq j$, $\{i,j\} \subset \{1,\dots,n\}$;
- $\cap_{i=1}^n \mathcal{R}_{g_0^{[i]}}(k_0\rho) = \emptyset$, where k_0 is the one defined in Property P2 in Section III. □

The assumption basically requires that at the initial time the vehicles are in different positions and that at least three of them are sufficiently far from each other. This allows us to deal with reachable sets that are disjoint and simply connected.

In order to set up an approximated version of the minimum-time servicing problem, we introduce an approximation of the reachable set for a Dubins vehicle. Given a configuration $g \in SE(2)$, we define $\widehat{\mathcal{R}}_g(t)$, for any $t \geq 0$, a set with the following properties:

(i) $\mathcal{R}_g(t) \subset \widehat{\mathcal{R}}_g(t)$;
(ii) $\widehat{\mathcal{R}}_g(t)$ is a *strictly convex* set.

Remark 4.1: A possible way to construct the set $\widehat{\mathcal{R}}_g(t)$ for given g and t is, for example, to approximate the non-convex portion of the set with an arc of circle.A detailed discussion of the construction of an approximation $\widehat{\mathcal{R}}_g(t)$ will be given in a forthcoming document. □

The following important result may be proven.

Lemma 4.2: Given any $(g_1, g_2, g_3) \in SE(2)^3$ satisfying Assumption A0 (for $n = 3$), let

$$\begin{aligned} \lambda^* = &\min_{\lambda \geq 0}\ \lambda \\ &\text{subj. to} \cap_{i=1}^3 \widehat{\mathcal{R}}_{g_i}(\lambda) \neq \emptyset. \end{aligned}$$

Then, the intersection $\cap_{i=1}^3 \widehat{\mathcal{R}}_{g_i}(\lambda^*)$ is a single point. □

We are ready to define the approximated minimum-time servicing problem. Given the initial configurations $(g_0^{[1]}, \dots, g_0^{[n]}) \in SE(2)^n$, we consider the following minimization problem.

$$\begin{aligned} &\text{minimize}_{t \geq 0}\ t \\ &\qquad \text{subj. to } \cap_{i=1}^n \widehat{\mathcal{R}}_{g_0^{[i]}}(t) \neq \emptyset. \end{aligned} \tag{4}$$

The following proposition may be proven.

Proposition 4.3: Let $(g_0^{[1]}, \dots, g_0^{[n]}) \in SE(2)^n$ be a collection of initial configurations satisfying Assumption A0. The approximated minimum-time servicing problem stated in (4) is an abstract linear program with combinatorial dimension $\delta = 3$. □

Remark 4.4: A question to investigate is how far the approximated minimum-time servicing point can be from the real one. Our conjecture is that the error depends on ρ but not on the minimum time t^*, so that the relative error goes to zero as t^* grows. It is worth noticing that the non-convex portion of the reachable set is a "small" portion of the set. Therefore, in many cases the solutions of the real and approximated problems will coincide. □

Having shown that the approximated minimum-time servicing problem is an abstract linear program, we may use the *FloodBasis* algorithm introduced in [8] to solve the problem over the network in a distributed way.

B. Distributed algorithm to solve a network abstract linear programming

We start reviewing the definition of *network abstract linear program*. Informally we can say that a network abstract linear program consists of three main elements: a network, an abstract linear program and a mapping that associates to each constraint of the abstract linear program a node of the network. Formally, a network abstract linear program (NALP) is a tuple $(\mathcal{G}, (H, \omega), \mathcal{B})$ consisting of a communication graph $\mathcal{G} = (I, E_{\text{cmm}})$, an abstract linear program (H, ω) and a bijective map $\mathcal{B} : H \to I$ called *constraint distribution map* that associates each constraint to a node. The *solution* of the network abstract linear program is attained when all processors in the network have computed a solution to the abstract linear program.

In [8] a distributed algorithm, called *FloodBasis* algorithm, was introduced to solve the network abstract linear program in a finite number of rounds under suitable assumptions on the communication graph. A sufficient condition is that the graph is undirected and connected at any communication round.

Here is an informal description of the *FloodBasis* algorithm:

> *[Informal description]* Each processor has a logical state of $\delta + 1$ variables taking values in H. The first δ components represent the current value of the basis to compute, while the last element is the constraint assigned to that node. At the start round the processor initializes every component of the basis to its constraint, then, at each communication round, performs the following tasks: (i) it acquires from its neighbors (a message consisting of) their current basis; (ii) it solves the abstract linear program for the constraint set given by the collection of its and its neighbors' basis and its constraint (that it maintains in memory), thus computing a new basis; (iii) it updates its logical state and message using the new basis obtained in (ii).

Remark 4.5: The time complexity of the *FloodBasis* algorithm, that is the number of rounds to compute the solution, is still under investigation. However, simulations and results from ALP literature let us conjecture that, in the average, it is of order $O(n)$. □

Assuming the communication graph remains connected, the Dubins vehicles can compute the minimum time servicing point by using the *FloodBasis* algorithm.

V. A CONTROL AND COMMUNICATION LAW TO COMPUTE AND REACH THE MINIMUM-TIME SERVICING POINT

A. Maintaining connectivity in a network of Dubins vehicles

In order to design a control and communication law for a wireless robotic network of Dubins vehicles, we need to solve a preliminary problem. That is, we have to ensure that the network does not become disconnected into subgroups that are not able to communicate among themselves. In this subsection we want to characterize the set of control inputs that ensure connectivity of the disk graph. We start with a negative result that says the disk graph cannot be kept connected.

Proposition 5.1 (Negative result for the disk graph): Let $\mathcal{S}$ be a robotic network of Dubins vehicles communicating according to the disk graph $G_{\text{disk}} = (I, E_{\text{disk}})$. For sufficiently small values of the communication interval, there exist initial configurations $\{g_0^{[i]}\}_{i\in\{1,\dots,n\}}$ such that the disk graph gets disconnected for any choice of the control inputs. □

Before introducing a suitable subgraph of the disk graph to maintain connected, we describe the strategy introduced in [1] to maintain the disk graph connected for a network of (discrete-time) single integrators. For $r > 0$ and $p \in \mathbb{R}^d$, we let $\overline{B}(p, r)$ denote the closed ball centered at p with radius r, i.e., $\overline{B}(p, r) = \{q \in \mathbb{R}^d \mid \|p - q\|_2 \leq r\}$. Network connectivity is maintained by restricting the allowable motion of each agent as follows. If agents i and j at positions $p^{[i]}(t) \in \mathbb{R}^2$ and $p^{[j]}(t) \in \mathbb{R}^2$ are neighbors in the disk graph $\mathcal{G}_{\text{disk}}$ at time t, then their positions at time $t + 1$ are required to belong to $\overline{B}\big(\frac{p^{[i]}(t)+p^{[j]}(t)}{2}, \frac{r_{\text{cmm}}}{2}\big)$. The constraint is illustrated in Figure 2.

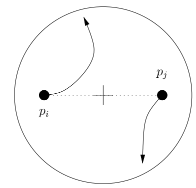

Fig. 2. Starting from p_i and p_j, the agents are restricted to move inside the disk centered at $\frac{p_i+p_j}{2}$ with radius $\frac{r_{\text{cmm}}}{2}$.

The above strategy suggests a way to ensure connectivity for Dubins vehicles. These vehicles cannot hover, however they have the possibility of loitering about a fixed rotation point. Therefore, instead of hovering, we let them loitering about a closed path, a *safety loitering path*, starting and ending at their current configuration. We assume that all the agents agree on the direction the loitering path is traversed. Without loss of generality we assume they agree on the clockwise direction. Then we may define a subgraph of the disk-graph as follows. There is an edge between two nodes i and j if and only if the safety loitering paths of agents i and j are contained in a circle of radius $\frac{r_{\text{cmm}}}{2}$ centered at the mean point of the (fixed) rotation points of the two loitering paths.

Formally, let $C_\rho(g_0) \subset \mathbb{R}^2$, $g_0 \in SE(2)$, be the image of the *circular loitering path* $\gamma_C : [0, 2\pi\rho] \to \mathbb{R}^2$ such that $g_{\gamma_C}(0) = g_{\gamma_C}(2\pi\rho) = g_0$ (with $\|\gamma_C'\| = 1$ and constant curvature $\frac{1}{\rho}$) and such that it is traversed in clockwise direction. We call such path *safety loitering path*. We denote $c_\rho(g_0) \in \mathbb{R}^2$ the center of the circle $C_\rho(g_0)$. The *Dubins disk graph* is the graph $G_{\text{Ddisk}} = (I, E_{\text{Ddisk}})$ such that $(i, j) \in E_{\text{Ddisk}}$ if and only if $\|c_\rho(g^{[i]}) - c_\rho(g^{[j]})\| \leq r_{\text{cmm}} - 2\rho$ or, equivalently, if and only if $C_\rho(g^{[k]}) \in \overline{B}\big(\frac{c_\rho(g^{[i]})+c_\rho(g^{[j]})}{2}, \frac{r_{\text{cmm}}}{2}\big)$, $k \in \{i, j\}$. Here we are assuming that $r_{\text{cmm}} > 2\rho$. Clearly, the Dubins disk graph is a subgraph of the disk graph. Network connectivity is maintained by restricting the allowable motion of each agent as follows. If agents i and j are neighbors in the Dubins disk graph $\mathcal{G}_{\text{Ddisk}}$ at time t, then their safety loitering paths at time $t + 1$ are required to belong to $\overline{B}\big(\frac{c_\rho(g^{[i]})+c_\rho(g^{[j]})}{2}, \frac{r_{\text{cmm}}}{2}\big)$. The constraint is illustrated in Figure 3 for an agent with three neighbors.

B. The move-toward estimate control and communication law

Next, we use the distributed algorithm that finds the minimum-time servicing point combined with the strategy to maintain connectivity to design a control and communication law that let the agents compute and reach the minimum-time servicing point.

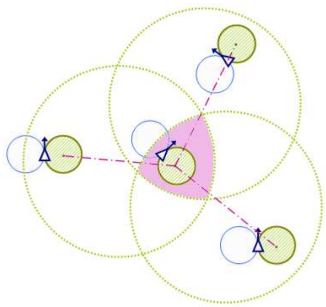

Fig. 3. The constraint set arising according to the Dubins disk graph for an agent with three neighbors.

The *move-toward-estimate* control and communication law may be summarized as follows. On the basis of their initial configurations, the agents run the *FloodBasis* algorithm for the problem of interest. A possibility could be to wait for the algorithm to end, then move toward the optimal servicing point. We propose a slightly different strategy. While the algorithm is running, each agent starts moving toward the point corresponding to its own current estimate of the solution. Everyone does it while maintaining connectivity with its current neighbors in the Dubins disk graph.

In Figure 4 and Figure 5 we illustrate two simulation runs of the above control and communication law for a team of 6 Dubins vehicles with radius of curvature $\rho = 1$, starting at randomly chosen initial configurations. In Figure 4 the communication interval has been set to 0.5s, whereas in Figure 5 it is 5s. In the first case the communication interval is sufficiently small and therefore the agents follow almost the centralized optimal paths to reach the minimum-time servicing point. In the second one, since the communication interval is larger, the agents tend to go towards the temporary estimates of the optimal point for a longer time, thus deviating consistently from the optimal path. Nevertheless they compute and reach the optimal point. In both the figures the path of the vehicles after they have reached the optimal point is not shown.

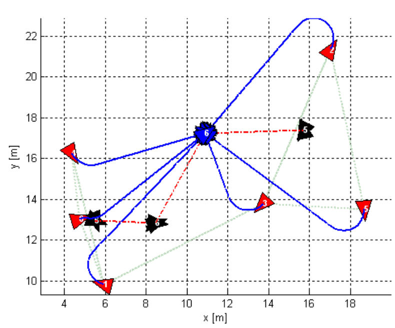

Fig. 4. Simulation of the move toward estimate control and communication law for a network of 6 Dubins vehicles with $\rho = 1$ and communication interval 0.5s.

VI. Conclusions

We have studied the following problem for a network of Dubins vehicles. Find the point in the plane that can be reached in minimum time by the all team of vehicles. We have shown that an approximation of this problem is equivalent to a class of optimization problems called abstract linear programming. Using results in [8] we have shown that the optimal point can be computed in a distributed way. We have also designed a control and communication law to compute and reach such point while maintaining the network connected.

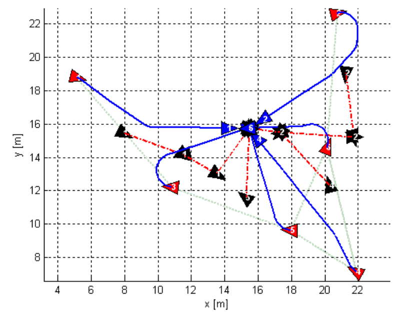

Fig. 5. Simulation of the move toward estimate control and communication law for a network of 6 Dubins vehicles with $\rho = 1$ and communication interval 5s.

References

[1] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita, "Distributed memoryless point convergence algorithm for mobile robots with limited visibility," *IEEE Trans Robotics & Automation*, vol. 15, no. 5, pp. 818–828, 1999.

[2] J. Lin, A. S. Morse, and B. D. O. Anderson, "The multi-agent rendezvous problem," in *Proc CDC*, Maui, HI, Dec. 2003, pp. 1508–1513.

[3] S. Martinez, "Practical rendezvous through modified circumcenter algorithms," in *Proc CDC*, New Orleans, LA, December 2007.

[4] G. Notarstefano and F. Bullo, "Distributed consensus on enclosing shapes and minimum time rendezvous," in *Proc CDC*, San Diego, CA, Dec. 2006, pp. 4295–4300.

[5] K. Savla, E. Frazzoli, and F. Bullo, "Traveling Salesperson Problems for the Dubins vehicle," *IEEE Trans Automatic Ctrl*, vol. 53, no. 9, 2008, (Submitted Jun 2006) To appear.

[6] K. Savla, F. Bullo, and E. Frazzoli, "The coverage problem for loitering Dubins vehicles," in *Proc CDC*, New Orleans, LA, Dec. 2007, pp. 1398–1403.

[7] L. Pallottino, V. G. Scordio, E. Frazzoli, and A. Bicchi, "Decentralized cooperative policy for conflict resolution in multi-vehicle systems," *IEEE Trans Robotics*, vol. 23, no. 6, pp. 1170–1183, 2007.

[8] G. Notarstefano and F. Bullo, "Network abstract linear programming with application to minimum-time formation control," in *Proc CDC*, New Orleans, LA, Dec. 2007, pp. 927–932.

[9] J. Matousek, M. Sharir, and E. Welzl, "A subexponential bound for linear programming," *Algorithmica*, vol. 16, no. 4/5, pp. 498–516, 1996.

[10] P. K. Agarwal and S. Sen, "Randomized algorithms for geometric optimization problems," in *Handbook of Randomization*, P. Pardalos, S. Rajasekaran, J. Reif, and J. Rolim, Eds. Dordrecht, The Netherlands: Kluwer, 2001.

[11] N. Amenta, "Helly-type theorems and generalized linear programming," *Descrete Comput. Geometry*, vol. 12, pp. 241–261, 1994.

[12] L. E. Dubins, "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, pp. 497–516, 1957.

[13] J. D. Boissonnat and X. N. Bui, "Accessibility region for a car that only moves forward along optimal paths," INRIA, Sophia-Antipolis, France, Tech. Rep. 2181, 1994.

[14] S. Martínez, F. Bullo, J. Cortés, and E. Frazzoli, "On synchronous robotic networks – Part I: Models, tasks and complexity," *IEEE Trans Automatic Ctrl*, vol. 52, no. 12, pp. 2199–2213, 2007.