43rd IEEE Conference on Decision and Control
December 14-17, 2004
Atlantis, Paradise Island, Bahamas

The Multi-Agent Rendezvous Problem - The Asynchronous Case

J. Lin and A.S. Morse
Yale University
jie.lin, as.morse@yale.edu

B. D. O. Anderson
Australian National University and
National ICT Australia
brian.anderson@anu.edu.au

***Abstract*— This paper is concerned with the collective behavior of a group of $n > 1$ mobile autonomous agents, labelled 1 through n, which can all move in the plane. Each agent is able to continuously track the positions of all other agents currently within its "sensing region" where by an agent's *sensing region* is meant a closed disk of positive radius r centered at the agent's current position. The *multi-agent rendezvous problem* is to devise "local" control strategies, one for each agent, which without any active communication between agents, cause all members of the group to eventually rendezvous at single unspecified location. This paper describes a family of asynchronously functioning strategies for solving the problem. Correctness is established appealing to the concept of "analytic synchronization."**

Current interest in cooperative control has led to a number of distributed control algorithms capable of causing large groups of mobile autonomous agents to perform useful tasks. Of particular interest here are provably correct algorithms which solve what we shall refer to as the "multi-agent rendezvous problem." This problem, which was posed in [1], is concerned with the collective behavior of a group of $n > 1$ mobile autonomous agents, labelled 1 through n, which can all move in the plane. Each agent is able to continuously track the positions of all other agents currently within its "sensing region" where by an agent's *sensing region* is meant a closed disk of positive radius r centered at the agent's current position. The *multi-agent rendezvous problem* is to devise "local" control strategies, one for each agent, which without any active communication between agents, cause all members of the group to eventually rendezvous at single unspecified location.

In this paper, as in [1], [2], we consider distributed strategies which guide each agent toward rendezvous by performing a sequence of "stop-and-go" maneuvers. A *stop-and-go maneuver* takes place within a time interval consisting of two consecutive sub-intervals. The first, called a *sensing period*, is an interval of fixed length during which the agent is stationary. The second, called a *maneuvering period*, is an interval of variable length during which the agent moves from its current position to its next 'way-point' and again come to rest. Successive way-points for each agent are chosen to be within r_M units of each other where r_M is a pre-specified positive distance no larger than r. It is assumed that for each agent i, a positive number τ_{M_i}, called a *maneuver time*, is chosen to be large enough so that the required maneuver for agent i from any one way-point to the next can be accomplished in at most τ_{M_i} seconds. Since our interest here is exclusively with devising of *high level* strategies which dictate when and where agents are to move, we will use point models for agents and shall not deal with how maneuvers are actually carried out or with how vehicle collisions are to be avoided.

In the synchronous case treated in [1], [2], the kth maneuvering periods of all n agents begin at the same time $\bar{t}_k$. The kth way-point of each agent is a function of the positions of its " registered neighbors" at time $\bar{t}_k$. Agent i's registered neighbors at time $\bar{t}_k$ are those other agents positioned within its sensing region at time $\bar{t}_k$. This notion of a neighbor induces a *symmetric* relation on the group since agent j is a registered neighbor of agent i at time $\bar{t}_k$ just in case agent i is a registered neighbor of agent j at the same time. Because of this it is possible to characterize neighbor relationships at time $\bar{t}_k$ with a simple graph whose vertices represent agents and whose edges represent existing neighbor relationships. Although the neighbor relation is symmetric, it is clearly not transitive. On the other hand if agent i is at the same position as neighbor j at time $\bar{t}_k$, then any registered neighbor of agent j at time $\bar{t}_k$ must certainly be a registered neighbor of agent i at the same time. It is precisely because of this *weak transitivity* property that one can infer a *global* condition of the entire agent group from a *local* condition of one agent and

Proofs and references will appear in the full length version of this paper. This research was supported by the National Science Foundation and by the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council.

0-7803-8682-5/04/$20.00 ©2004 IEEE

its neighbors. In particular, if the graph characterizing neighbor relationships at time $\bar{t}_k$ is connected, and any one agent is at the same position as all of its neighbors, then the weak transitivity property guarantees that all n agents have rendezvoused at time $\bar{t}_k$.

One way to ensure that a neighbor graph is connected at time $\bar{t}_k$, assuming it is connected when the rendezvousing process begins, is to constrain each agent's way points to be positioned in such a way so that no agent can lose any of its registered neighbors when it moves from one way point to the next. This can be accomplished using a clever idea proposed in [1]. An immediate consequence is that each agent's set of registered neighbors is non-decreasing and, because of this, ultimately converges to a fixed neighbor set for $\bar{t}_k$ sufficiently large.

A second local constraint is to require the way-point of each agent i at the beginning of its kth maneuvering period to lie in the "local" convex hull $\mathcal{H}_i(k)$ of agent i's own position and the sensed positions of its registered neighbors at time $\bar{t}_k$. It is easy to prove that the global convex hull $\mathcal{H}(k+1)$ of all n agent positions at time $\bar{t}_{k+1}$ to be non-increasing under this constraint.

A third constraint is to stipulate that for each i, the only condition under which agent i's kth way point can be positioned at a corner of $\mathcal{H}_i(k)$, is when $\mathcal{H}_i(k)$ is a single point. The global implication of doing this is that the diameter of $\mathcal{H}(k+1)$ must either be strictly smaller than the diameter of $\mathcal{H}(k)$ or every agent must be at the same position as all of its registered neighbors at time $\bar{t}_k$ – and this is true whether or not the graph characterizing neighbor relationships at time $\bar{t}_k$ is connected.

In [2], a more or less standard Lyapunov based argument is used to prove that if the preceding constraints are adopted by all agents and if the graph characterizing initial neighbor positions is connected, then all n agents will eventually rendezvous at a single point. Not surprisingly, the Lyapunov function used is the diameter of the global convex hull. However, although connectivity of the graph characterizing initial neighbor positions is sufficient for rendezvousing, it is not necessary. An example illustrating this is given in [2].

The strategies described in [1], [2] cannot be regarded as truly distributed because each agent's decisions must be synchronized to a common clock shared by all other agents in the group. In this paper we redefine the strategies so that a common clock is not required. To do this it is necessary to modify somewhat what is meant by a registered neighbor of agent i at time $\bar{t}_{ik}$ where $\bar{t}_{ik}$ is the time at which agent i's kth maneuvering period begins. Our definition is guided by considerations discussed above for the synchronous case. For example, the new definition is crafted to retain versions of the symmetry and weak transitivity properties of the neighbor relation inherent in the synchronous case. Doing this is challenging, because unlike the synchronous case, the time each agent registers its neighbors and its neighbor's positions is not synchronized with one another.

The same way-point update rules considered in the synchronous case are adopted for the asynchronous case. Thus the only functional differences between the two cases are the definitions of registered neighbors and their positions. Of course in the asynchronous case, way-point updates are computed asynchronously, whereas in the synchronous case they are not.

Not surprisingly, the analysis of the asynchronous version of the problem is considerably more challenging than that of the synchronous version. For example, while it is more or less obvious in the synchronous case that agents retain their neighbors as the system evolves, proving that this is also true in the asynchronous case involves a number of steps.

A single ordered time set can be formed by merging the distinct "event times" $\bar{t}_{ik}$, $i \in \{1, 2, \ldots, n\}$, $k \geq 1$ generated by all n agents. The elements of this set are relabelled as $t_1, t_2, \cdots$ in such a way that $t_j < t_{j+1}, j \in \{1, 2, \ldots\}$. With this notation, agent i's registered neighbors at its kth event time $\bar{t}_{ik}$ are its registered neighbors at time $t_{P_i(k)}$ where $P_i(k)$ denotes that value of p for which $t_p = \bar{t}_{ik}$. For each $i \in \{1, 2, \ldots, n\}$, the domain of definition of agent i's registered neighbors is then extended from the set $\{t_{P_i(k)} : k \geq 1\}$ to the set $\{t_p : p \geq P_i(1)\}$ by stipulating that for values of t_p which are between two successive event times of agent i, agent i's registered neighbors remain the same. This means that registered neighbors of each agent are defined at each time $t_p \geq t_{\bar{p}}$ where $\bar{p} \overset{\Delta}{=} \max\{P_1(1), P_2(1), \ldots P_n(1)\}$. Because of this, it is possible to describe neighbor relationships with a directed graph with vertex set $\{1, 2, \ldots, n\}$ and directed edge set defined so that (i, j) is a directed edge from vertex i to j just in case agent j is a registered neighbor of i at event time t_s. The main result of this paper {Corollary 1} is that if this graph is ever strongly connected, then all n agents will eventually rendezvous.

To establish the correctness of Corollary 1 requires the analysis of the asymptotic behavior of the *asynchronous* process which describe the n-agent system. Despite the apparent complexity of this process, it is possible to capture its salient features using a suitably defined *synchronous* discrete-time, hybrid dynamical system $\mathbb{S}$. We call the sequence of steps involved in defining $\mathbb{S}$ *analytic synchronization*. Analytic synchronization is applicable to any finite family of continuous or discrete time dynamical processes $\{\mathbb{P}_1, \mathbb{P}_2, \dots, \dots, \mathbb{P}_n\}$ under the following conditions. First, each process $\mathbb{P}_i$ must be a dynamical system whose inputs consist of functions of the states of the other processes as well as signals which are exogenous to the entire family. Second, each process $\mathbb{P}_i$ must have associated with it an ordered sequence of event times $\{t_{i1}, t_{i2}, \dots\}$ defined in such a way that the state of $\mathbb{P}_i$ at event time $t_{i(k_i+1)}$ is uniquely determined by values of the exogenous signals and states of the $\mathbb{P}_j, \;\; j \in \{1, 2, \dots, n\}$ at event times t_{jk_j} which occur prior to $t_{i(k_i+1)}$ but in the finite past. Event time sequences for different processes need not be synchronized. Analytic synchronization is a straight forward procedure for creating a single synchronous process for purposes of analysis which captures the salient features of the original asynchronous processes. As a first step, all n event time sequences are merged into a single ordered sequence of even times $\mathcal{T}$. The "synchronized" state of $\mathbb{P}_i$ is then defined to be the original of $\mathbb{P}_i$ at $\mathbb{P}_i$'s event times $\{t_{i1}, t_{i2}, \dots\}$; at values of $t \in \mathcal{T}$ between event times t_{ik_i} and $t_{i(k_i+1)}$, the synchronized state of $\mathbb{P}_i$ is taken to be the same as the value of its original state at time t_{ik}. What results is a synchronous dynamical system evolving on $\mathcal{T}$ with state composed of the synchronized states of the n individual processes. The definition of $\mathbb{S}$ in section §II illustrates the analytic synchronization procedure.

I. The Asynchronous Agent System

For each agent i, the real time axis can be partitioned into a sequence of time intervals $[0, t_{i1})$, $[t_{i1}, t_{i2})$, $\dots$, $[t_{i(k_i-1)}, t_{ik_i})$, $\dots$each of length at most $\tau_D + \tau_{M_i}$ where τ_D is a number greater than τ_{M_i} called a *dwell time*. Each interval $[t_{i(k_i-1)}, t_{ik_i})$ consists of a *sensing period* $[t_{i(k_i-1)}, \bar{t}_{ik_i})$ of fixed length τ_D during which agent i is stationary, followed by a maneuvering period $[\bar{t}_{ik_i}, t_{ik_i})$ of length at most τ_{M_i} during which agent i moves from its current position to its next way-point. Although all agents use the same dwell time, they operate asynchronously in the sense that the time sequences $t_{i1}, t_{i2}, \cdots, \;\; i \in \{1, 2, \dots, n\}$ are uncorrelated. Thus each agent's strategy can be implemented independent of the rest, without the need for a common clock.

Because of the asynchronous nature of the control strategies under consideration, care must be exercised in defining what is meant by a registered neighbor if one is to end up with something similar to the symmetry property of the neighbor relationship defined in the synchronous case. For the asynchronous case, agent i's *registered neighbors* at time $\bar{t}_{ik}$ are taken to be those agents which are fixed at one position within agent i's sensing region for at least $\tau_S > 0$ seconds during agent i's kth sensing period $\mathcal{S}_i(k) \overset{\Delta}{=} [t_{i(k-1)}, \bar{t}_{ik})$. Here τ_S is a positive number called a *sensing time*. For reasons to be made clear below, we shall require τ_S to satisfy

$$\tau_S \leq \frac{1}{2}(\tau_D - \tau_{M_i}) \qquad \forall i \in \{1, 2, \dots, n\} \qquad (1)$$

For any agent j, there may be more than one distinct interval of length at least τ_S within $\mathcal{S}_i(k)$ during which agent j is stationary. Let t^* denote the end time of the last of these. For purposes of calculation, agent i takes the *registered position* of agent j at the beginning of its kth maneuvering period, to be the actual position of agent j at *registration time* t^*. To attain a symmetry-like property for the asynchronous case, it is necessary to make sure that the *registration interval* $[t^* - \tau_S, \; t^*)$ lies within one of agent j's sensing periods. One way to guarantee that this is so is to require each agent to keep moving during each of its maneuvering periods except possibly for brief periods which are each shorter than τ_S. In the sequel we will assume that registration of each agent j during one of agent i's sensing periods always occurs at the end of a registration interval $[t^* - \tau_S, \; t^*)$ which also lies within one of agent j's sensing periods. Note that this and the requirement that agent j is stationary during its sensing periods together imply that agent j's registered position $x_j(t^*)$ is equal to $x_j(\bar{t}_{jk^*})$ where k^* is the sensing/maneuvering interval of agent j during which registration takes place.

A. Cooperation Assumption

Prompted by the preceding, we say that for each $i, j \in \{1, 2, \dots, n\}$, agent j's qth sensing period $\mathcal{S}_j(q)$ *strongly overlaps* agent i's kth sensing period $\mathcal{S}_i(k)$ if the overlap is a non-empty interval of length at least τ_S seconds. In the sequel we write $\Omega(i, k, j, q) \succ 0$ whenever $\mathcal{S}_i(k)$ and $\mathcal{S}_j(q)$ strongly overlap. The definition of a registered neighbor determines a relationship between agents similar to the symmetric relation determined by that of a registered neighbor in the synchronous case. Let $\lceil \bar{t}_{ik} \rceil_j$ denote the smallest integer q such that $\bar{t}_{jq} \geq \bar{t}_{ik}$.

Proposition 1: Suppose that agent j is a registered neighbor of agent i at the beginning of agent i's kth maneuvering period. Then agent i is a registered neighbor of agent j at the beginning of agent j's qth or $(q-1)st$ maneuvering period where $q = \lceil \bar{t}_{ik} \rceil_j$.

The notion of a pairwise motion constraint imposed in the synchronous case [2] can be replaced with the following constraint. Agent i is said to satisfy the *motion constraints induced by its neighbors*, if for each $j \in \{1, 2, \ldots, n\}$ for which $j \neq i$ and each $k \in \{1, 2, \ldots\}$ for which agent j is a registered neighbor of agent i at the beginning of maneuvering period k, the position to which agent i moves at the end of the period is within a closed disk of diameter r centered at the mean of agent i's position and the registered position of agent j both at the beginning of the period {i.e., at time $\bar{t}_{ik}$}. In the synchronous case, satisfaction of the pairwise motion constraint by agent i and neighbor j causes each to retain the other as a neighbor. The following proposition implies that essentially the same thing holds in the asynchronous case.

Proposition 2: Suppose that agents i and j satisfy the motion constraints induced by their registered neighbors. If agent j is a registered neighbor of agent i at the beginning of agent i's kth maneuvering period, then agent j is also a registered neighbor of agent i at the beginning of agent i's $k+1st$ maneuvering period.

We are interested in strategies which cause agents to retain their registered neighbors. We therefore make the following assumption.

Cooperation Assumption: *Each agent i satisfies the motion constraints induced by each of its registered neighbors.*

Suppose that the cooperation assumption is satisfied. Proposition 2 states that if agent j is a registered neighbor of agent i during maneuvering interval k then it will also be a registered neighbor of agent i during maneuvering interval $k+1$. In other words, if the cooperation assumption is satisfied, each agent retains all of its prior registered neighbors as the system evolves. Thus if $\mathcal{N}_i(k)$ denotes the sent of labels of agent i's neighbors at the beginning of its kth maneuvering period, then $\mathcal{N}_i(k) \subset \mathcal{N}_i(k+1), \ k \geq 1$.

Agent i's kth *way-point* $\bar{x}_i(k)$ is the point to which agent i moves at the end of its kth maneuvering period. Thus if $x_i(t)$ denotes the position of agent i at time t represented in a world coordinate system, then $x_i(t_{ik})$ and agent i's kth way-point are one and the same. The rule which determines $\bar{x}_i(k)$ is essentially the same as considered previously for the synchronous case in [1], [2], except that now $\bar{x}_i(k)$ depend on agent i's its own position at the beginning of its kth maneuvering period and the registered {relative} positions of agent i's registered neighbors at the beginning of the period. In particular if agent i has m_{ik} registered neighbors at time $\bar{t}_{ik}$ with registered positions $z_1, z_2, \ldots, z_{m_{ik}}$ relative to agent i's, then agent i moves to the position $\bar{x}_i(k) = x_i(t_{i(k-1)}) + u_{m_{ik}}(z_1, \ldots, z_{m_{ik}})$ at the end of the period where

$$z_j = x_{ii_j}(\bar{t}_{ik}) - x_i(t_{i(k-1)}), \quad j \in \{1, 2, \ldots, m_{ik}\}, \tag{2}$$

and $x_{ii_j}(\bar{t}_{ik})$ is the registered position of neighbor i_j at time $\bar{t}_{ik}$. As in [2], $u_0 = 0$ and for $m \in \{1, \ldots, n-1\}$ u_m is a continuous control law mapping $\mathbb{D}^m$ into $\mathbb{D}_M$ where $\mathbb{D}$ and $\mathbb{D}_M$ are the closed disks of radii r and r_M respectively, centered at the origin in $\mathbb{R}^2$. For $m > 0$, u_m is defined so that the aforementioned neighbor motion constraint is satisfied and, in addition so that for each $\{z_1, z_2, \ldots, z_m\} \in \mathbb{D}^m$, $u_m(z_1, z_2, \ldots, z_n)$ is in the convex hull of $\{0, z_1, z_2, \ldots, z_m\}$, but not at a corner unless $z_1 = z_2 = \cdots = z_m = 0$. Examples satisfying these conditions can be found in [1], [2].

Since each agent is assumed to move to its kth way-point at the end of its kth maneuvering period, agent i's position at time $t_{ik}$is given by

$$x_i(t_{ik}) = x_i(t_{i(k-1)}) + u_{m_{ik}}(z_1, \ldots, z_{m_{ik}}) \tag{3}$$

where z_j is as in (2). In the full length version of this paper it is shown that $x_{ij}(\bar{t}_{ik})$ can be written explicitly as

$$x_{ij}(\bar{t}_{ik}) = \left\{ \begin{array}{ll} x_j(\bar{t}_{jq}) & \text{if } \Omega(i,k,j,q) \succ 0 \\ x_j(\bar{t}_{j(q-1)}) & \text{otherwise} \end{array} \right\} \tag{4}$$

where $j \in \mathcal{N}_i(k)$, $q = \lceil \bar{t}_{ik} \rceil_j$, and $\mathcal{N}_i(k) =$

$$\{j : ||x_i(\bar{t}_{ik}) - x_j(\bar{t}_{iq})|| \leq r, \ \ \Omega(i,k,j,q) \succ 0\} \bigcup$$
$$\{j : ||x_i(\bar{t}_{ik}) - x_j(\bar{t}_{i(q-1)})|| \leq r, \ \ \Omega(i,k,j,q-1) \succ 0\} \tag{5}$$

Of course the neighbor set $\mathcal{N}_i(k)$ and the registration positions $x_{ij}, \ j \in \mathcal{N}_i(k)$ all depend on i and k.

1) Main Results: Note that because agents do not move during sensing periods, for each $i \in \{1, 2, \ldots, n\}$ the positions of agent i at times $t_{i(k-1)}$ and t_{ik} are the same as at times $\bar{t}_{ik}$ and $\bar{t}_{i(k+1)}$ respectively. Thus (3) can also be written as

$$\begin{aligned} x_i(\bar{t}_{i(k+1)}) \quad = \quad & x_i(\bar{t}_{ik}) + u_{m_{ik}}(x_{ii_1}(\bar{t}_{ik}) - x_i(\bar{t}_{ik}), \\ & \ldots, x_{ii_{m_{ik}}}(\bar{t}_{ik}) - x_i(\bar{t}_{ik})) \end{aligned} \tag{6}$$

The n equations given by (6) for $i \in \{1, 2, \ldots, n\}$ together with (4) and (5) completely describes the evolution of the positions of the n agents under consideration as each maneuvers from way-point to way-point. Just as in the synchronous case, the analysis of these equations depends on the relationships between registered neighbors and how these relationships evolve with time. To characterize these relationships, we first extend the domain of definition of each agent's registered neighbors from its set of maneuvering period start times to a suitably defined set of "event times" common to all n agents. By an *event time* is meant any time $\bar{t}_{ik}$ at which any maneuvering period $[\bar{t}_{ik}, t_{ik})$ of any agent begins. Let $\{\bar{t}_{ik} :\ \ i \in \{1, 2, \ldots, n\},\ k \geq 1\}$ denote the set of all distinct event times. Label this set's elements as $t_1, t_2, \cdots, t_p, \cdots$ in such a way so that $t_p < t_{p+1},\ j \in \{1, 2, \ldots\}$. For $i \in \{1, 2, \ldots, n\}$, let P_i denote that strictly monotone function from the set of positive integers $\mathcal{I}$ to $\mathcal{I}$ which assigns to $k \in \mathcal{I}$ that value of $p \in \mathcal{I}$ for which $t_p = \bar{t}_{ik}$. Thus with this notation, $t_{P_i(k)} = \bar{t}_{ik}$ so agent i's registered neighbors at its kth event time $t_{P_i(k)}$, are its registered neighbors at time $\bar{t}_{ik}$. For each $i \in \{1, 2, \ldots, n\}$ we extend the domain of definition of agent i's registered neighbors from the set $\{t_{P_i(k)} : k \geq 1\}$ to the set $\{t_p : p \geq P_i(1)\}$ by stipulating that for values of t_p which are between two successive event times of agent i, say between t_{ik} and $t_{i(k+1)}$, agent i's registered neighbors are the same as its registered neighbors at time t_{ik}.

Let $\mathcal{T} \triangleq \{t_{\bar{p}}, t_{\bar{p}+1}, t_{\bar{p}+2} \ldots\}$ denote the set of all event times greater than or equal to $t_{\bar{p}}$ where $\bar{p} \triangleq \max\{P_1(1), P_2(1), \ldots P_n(1)\}$. Note that the registered neighbors of each agent are defined at each time in $\mathcal{T}$. For each $p \geq \bar{p}$, it is therefore possible to describe neighbor relationships using a directed graph $\mathbb{G}_p$ with vertex set $\{1, 2, \ldots, n\}$ and directed edge set defined so that (i, j) is a directed edge from vertex i to j just in case agent j is a registered neighbor of agent i at event time t_s.

Let us partially order the set of all directed graphs with vertex set $\{1, 2, \ldots, n\}$ by agreeing to say that $\mathbb{G}$ is contained in $\bar{\mathbb{G}}$ if the edge set of $\mathbb{G}$ is a subset on the edge set of $\bar{\mathbb{G}}$. It is natural then to define the *union* of a collection of such graphs to be the directed graph with vertex set $\{1, 2, \ldots, n\}$, and edge set equaling the union of the edge sets of all of the graphs in the collection. Because of the cooperation assumption and Proposition 2, we know that each agent keeps all of its registered neighbors as the system evolves. What this means is the sequence of graphs $\mathbb{G}_{\bar{p}}, \mathbb{G}_{\bar{p}+1}, \ldots, \mathbb{G}_p, \ldots$ forms the ascending chain

$$\mathbb{G}_{\bar{p}} \subset \mathbb{G}_{\bar{p}+1} \subset \cdots \mathbb{G}_p \cdots \tag{7}$$

Because the set of directed graphs on vertices $\{1, 2, \ldots, n\}$ is a finite set, the chain must converge to the graph

$$\mathbb{G} \triangleq \bigcup_{p=\bar{p}}^{\infty} \mathbb{G}_p \tag{8}$$

in a finite number of steps. More is true. Suppose that agent i has agent j as a registered neighbor at the beginning of one of agent i's maneuvering periods. Then because of Proposition 1, agent i must be a registered neighbor of agent j at the beginning of one of agent j's maneuvering periods. These observations together with the cooperation assumption imply that agents i and j must both eventually become and remain registered neighbors of each other. As a consequence, there must be directed arcs in $\mathbb{G}$ from vertex i to vertex j as well as from vertex j to vertex i. Clearly $\mathbb{G}$ must be a directed graph with the property that for each distinct pair of vertices - say i and j - either there is no directed arc connecting one to the other or there are two directed arcs one from vertex i to vertex j and the other from vertex j to vertex i. Directed graphs with this property are usually regarded as simple graphs whose edges represent such pairs of directed arcs. In the sequel we shall adopt this viewpoint and refer to $\mathbb{G}$ as a simple graph. Our main result is as follows.

Theorem 1: Let $u_0 = 0 \in \mathbb{D}_M$ and for each $m \in \{1, 2, \ldots, n-1\}$, let $u_m : \mathbb{D}^m \to \mathbb{D}_M$ be any continuous function satisfying the aforementioned control law conditions. For each set of initial agent positions $x_1(0), x_2(0), \ldots, x_n(0)$, each agent's position $x_i(t)$ converges to a unique point $\pi_i \in \mathbb{R}^2$ such that for each $i, j \in \{1, 2, \ldots, n\}$, either $\pi_i = \pi_j$ or $||\pi_i - \pi_j|| > r$. Moreover, if agent j is a registered neighbor of agent i at the beginning of one of agent i's maneuvering periods, then $\pi_i = \pi_j$.

Theorem 1 states that the strategies under consideration cause all agents' positions to converge to points in the plane with the property that each pair of such points are either equal to each other, or separated by a distance greater than r units. The theorem further states that if one agent is ever a registered neighbor of another, then both converge to the same point. Thus all n agents position will converge to a single point if any one directed graph in the ascending chain is weakly connected. We are led to the following corollary.

Corollary 1: If at any event time $t_p \geq t_{\bar{p}}$, the directed graph $\mathbb{G}_p$ characterizing registered neighbors is strongly connected, then positions of all n agents converge to a common point in the plane.

II. A Synchronous Model of the Asynchronous Agent System

To establish the correctness of Theorem 1 requires the analysis of the asymptotic behavior of the *asynchronous* process described by (4) - (6) for $i \in \{1, 2, \ldots, n\}$. Despite the apparent complexity of this process, it is possible to capture its salient features for t_p sufficiently large using a suitably defined *synchronous* discrete-time, hybrid dynamical system $\mathbb{S}$. We will define $\mathbb{S}$ to be a synchronous discrete-time dynamical system evolving on the index set $\mathcal{P} = \{p; p \geq p^*\}$ where p^* is the smallest values of $p \geq \bar{p}$ for which the ascending chain shown in (7) has converged to the limit graph $\mathbb{G}$ in (8). Thus for $p \in \mathcal{P}$, each agent's registered neighbors of do not change. For simplicity, we will only deal with the case when each agent has at least one neighbor. Thus for $i \in \{1, 2, \ldots, n\}$, agent i's set of neighbor indices $\mathcal{N}_i$ is constant.

We will take as the state space of $\mathbb{S}$, the space $\mathcal{X}$ of all lists $\{y_1, y_2, \ldots y_n, w_1, w_2, \ldots, w_n\}$ satisfying

$$y_i, w_i \in \mathbb{R}^2, ||y_i - y_j|| \leq r, j \in \mathcal{N}_i, i \in \{1, \ldots, n\} \quad (9)$$

In the sequel we write y for $\{y_1, y_2, \ldots y_n\}$ and w for $\{w_1, w_2, \ldots w_n\}$. We refer to $\{y_i, w_i\}$ as the state of "node" i. For $i \in \{1, 2, \ldots, n\}$ let P_i^{-1} be a left inverse of P_i and let $\mathcal{P}_i = \mathcal{P} \cap \text{image } P_i$. We now define $\mathbb{S}$ to be a time-varying system with state $\{y, w\}$; for each $i \in \{1, 2, \ldots, n\}$, the state of node i evolves on $\mathcal{P}$ according to update equations defined for $p \in \mathcal{P}_i$ by

$$\begin{aligned} y_i(p+1) &= y_i(p) + u_{m_i}(v_{ii_1}(p) - y_i(p), \ldots, \\ &\quad v_{ii_{m_i}}(p) - y_i(p)) && (10) \\ w_i(p+1) &= y_i(p) && (11) \end{aligned}$$

where for $j \in \mathcal{N}_i$

$$v_{ij}(p) = \left\{ \begin{array}{ll} y_j(p) & \text{if } \ \Omega(i, P_i^{-1}(p), j, \lceil t_p \rceil_j) \succ 0 \\ w_j(p) & \text{otherwise} \end{array} \right\}, \quad (12)$$

and for $p \notin \mathcal{P}_i$ by

$$\begin{aligned} y_i(p+1) &= y_i(p) && (13) \\ w_i(p+1) &= w_i(p) && (14) \end{aligned}$$

We require y_i satisfies the *neighbor constraints*

$$||y_i(p) - w_j(p)|| \leq r \quad \text{if} \quad \Omega(i, P_i^{-1}(p), j), \lceil t_p \rceil_j) \not\succ 0, \quad p \in \mathcal{P}_i, \quad j \in \mathcal{N}_i \quad (15)$$

Note that these constraints together with the definition of $\mathcal{X}$ and v_{ij} insure that $||v_{ij} - y_i(p)|| \leq r$ whenever $p \in \mathcal{P}_i$. This in turn is necessary for (10) to make sense because the domain of u_{m_i} is $\mathbb{D}^{m_i}$.

The preceding defines $\mathbb{S}$ to be a synchronous discrete-time dynamical system with state constraints given by (15). The definition depends on the $\mathcal{N}_i$ as well as the n event time sequences $\{\bar{t}_{ik}; k \geq 1\}$. We've assumed that the $\mathcal{N}_i$ are non-empty; in addition, $\mathcal{N}_i \subset \{1, 2, \ldots, i-1, i+1, \ldots, n\}$. It can be shown that the $\mathcal{N}_i$ all have the following *symmetry property:* If $j \in \mathcal{N}_i$ then $i \in \mathcal{N}_j$. Because of this we can associate with the $\mathcal{N}_i$ a simple graph $\mathbb{G}$ with vertex set $\{1, 2, \ldots, n\}$ and edge set defined such that (i, j) is in the edge set just in case $i \in \mathcal{N}_j$ and $j \in \mathcal{N}_i$. Note that this is the same as the simple graph mentioned just before theorem 1.

By a *trajectory* of $\mathbb{S}$ is meant a sequence of states $\{\{y(p), w(p)\} : p \in \mathcal{P}\}$ which satisfy (10) - (14) as well as the neighbor constraints (15). In the full length version of this paper it is proved that the family of such trajectories is non-empty and contains the trajectory which represents actual agent system under consideration. In particular it is shown that if we define

$$\left. \begin{aligned} y_i(p) &= x_i(\bar{t}_{ik}) \\ w_i(p) &= x_i(\bar{t}_{i(k-1)}) \end{aligned} \right\}, \quad \begin{aligned} P_i(k-1) < p \leq P_i(k), \\ k \in P_i^{-1}(\mathcal{P}) \end{aligned} \quad (16)$$

for $i \in \{1, 2, \ldots, n\}$, then $\{\{y(p), w(p)\} : p \in \mathcal{P}\}$ is a trajectory of $\mathbb{S}$. Note that y_i has been defined so that it is constant between agent i's event times and agrees with x_i whenever p is such that t_p is within one of agent i's sensing periods.

In the full length version of this paper it is proved that the trajectory of $\mathbb{S}$ defined by (16) converges to a point at which $y_1 = y_2 = \cdots y_n = w_1 = w_2 = \cdots = w_n$ provided $\mathbb{G}$ is connected. Theorem 1 is an immediate consequence.

III. Concluding Remarks

The approach taken in this paper appears to have much in common with the embedding process discussed in Chapter 7 of [3] for analyzing "partially asynchronous iterative algorithms." This suggests that the tools developed in [3] may be helpful in understanding the asynchronous system considered in this paper.

References

[1] 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*, pages 818–828, oct 1999.

[2] J. Lin, A. S. Morse, and B. D. O. Anderson. The multi-agent rendezvous problem. In *Proc 2003 CDC*, pages 1508–1513, dec 2003.

[3] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation*. Prentics Hall, 1989.