Consensus-Based Rendezvous
Carlos H. Caicedo-Núñez and Miloš Žefran

Abstract—We present a distributed algorithm for solving the rendezvous problem based on the so-called consensus protocols. By using the properties of the consensus protocols we show that our algorithm drives the agents in the formation to a common configuration without the need for the agents to communicate with each other, and with no need for the agents to work in a common reference frame. We also show that several existing rendezvous algorithms can be viewed as special cases of the proposed algorithm. We include simulation results to illustrate the approach.

I. INTRODUCTION

In robotic networks, rendezvous refers to the task of driving each of the agents in a robotic network equipped with sensors detecting other agents towards a common configuration without the need for the agents to communicate with each other.

Several distributed algorithms for solving the rendezvous problem have been proposed. Originally presented in [1], the problem has been extended to both synchronous [2] and asynchronous [3], [4] cases. The proposed algorithms are all distributed in the sense that each robot takes decisions based only on the information it can gather from a subset of the agents in the formation.

In recent years, another class of algorithms, called consensus protocols, attracted a great deal of attention in the control community. Starting with the works by Jadbabaie et al. [5], Olfati-Saber and Murray [6] and Moreau [7], the protocols have been applied to a number of problems. In addition to the developments in robotics and control communities [5], [8]–[10], consensus protocols have been studied in parallel computing, the setting in which they were originally devised [11]–[13]. Variations of these protocols, for instance gossip algorithms [14] and aggregation protocols [15] have been also developed in the computer network community.

Observe that if the agents can move freely in $\mathbb{R}^n$, achieving rendezvous is equivalent to achieving consensus in $\mathbb{R}^n$. This relationship between consensus and rendezvous has not been unnoticed [16]–[18]. Nonetheless, an in-depth investigation of this relationship has not been done before so our paper fills this gap. We propose a consensus protocol that achieves rendezvous. Thus, the convergence properties of the latter are inherited from the former. We also show that the rendezvous problem is easy to solve and analyze if it is treated as a consensus problem.

Our results establish that for any point that each agent chooses in a certain set defined by its neighbors, it can move $\delta$-close to it for arbitrary $\delta$, and obtain a solution to the rendezvous problem under some mild restrictions that are presented in the paper. We show, for instance, that our algorithm will be independent of the coordinate frame, removing one of the requirements in [4], and that other popular rendezvous algorithms, such as the circucenter algorithm [1], [2] can be viewed either as a particular implementation of our approach, or as a limit point for valid solutions with our algorithm.

II. PRELIMINARIES AND NOTATION

A. Consensus algorithms

Consensus protocols were introduced in [11], and then re-discovered in the control community [5]. Subsequent research inspired by this work led to the continuous time version [6] and its generalization [7]. The reader interested in consensus algorithms is referred to the survey [19] and the references therein. In this paper, we will focus on the discrete time consensus algorithm.

Let $x_0 \in \mathbb{R}^n$ be a vector, and let $A \in \mathbb{R}^{n \times n}$ be a square matrix with the following properties:

1) $A$ is primitive, i.e. there is a positive integer $k$ such that $A^k$ has all its entries positive, and

2) $A$ is stochastic, i.e. all its entries are non-negative, and the sum of the entries in each column is equal to 1.

It follows from the Geršgorin’s circle theorem [20] and the Perron-Frobenius Theorem for primitive matrices [21] that $A$ has all but one of its eigenvalues in the interior of the complex unit circle, that the remaining eigenvalue is equal to 1 and has 1, the vector in $\mathbb{R}^n$ which has all its entries equal to 1, as its associated eigenvector. From here it follows that the discrete time linear system given by

$$x_m = A^n x_0 \quad (1)$$

is stable, and converges to an equilibrium point which happens to be a scalar multiple of 1, the eigenvector associated to the eigenvalue 1 [22]. We will call such matrix $A$ a consensus matrix.

It is shown, among others in [7], [11], that if $\{A_i\}_{i \in \mathbb{N}} \subseteq \mathbb{R}^{n \times n}$ are all consensus matrices, and the matrix $A_i$ is such that its positive entries are bounded below by a certain $\alpha$, then

$$\lim_{m \to \infty} \prod_{i=1}^{m} A_i = \frac{1}{n} 11^T. \quad (2)$$

The following lemma is taken from [21].

Lemma 1: Let $A, B \in \mathbb{R}^{n \times n}$ be two non-negative matrices. If both $A$ and $B$ have its zero and positive entries in the same positions, then either both matrices are primitive,
or none of them is. In other words, for non-negative matrices the condition of being primitive depends only on the profile of the matrix.

B. Proximity graphs

The concept of proximity graph will be useful to introduce the notion of neighbor which will be fundamental for our algorithm. We assume the reader is familiar with the basic concepts of graph theory, as presented for instance in [23], [24]. During this section we will be following the presentation in [2].

Let \( F(\mathbb{R}^n) \) be the set of finite point sets in \( \mathbb{R}^n \). We denote by \( \mathcal{P} = \{p_1, \ldots, p_m\} \subset \mathbb{R}^n \) a typical element of \( F(\mathbb{R}^n) \), where \( p_1, \ldots, p_m \) are distinct points. Let \( \mathcal{G}(\mathbb{R}^n) \) be the set of undirected graphs whose vertex set belongs to \( F(\mathbb{R}^n) \).

A proximity graph function \( G : F(\mathbb{R}^n) \to \mathcal{G}(\mathbb{R}^n) \) is a map that associates to each element \( \mathcal{P} \in F(\mathbb{R}^n) \) an undirected graph with vertices given by the elements of \( \mathcal{P} \), and with the set of edges \( \mathcal{E} \) defined by the function \( \mathcal{E}_G : F(\mathbb{R}^n) \to F(\mathbb{R}^n \times \mathbb{R}^n) \) contained in \( \mathcal{P} \times \mathcal{P} \setminus \text{diag}(\mathcal{P}) \), where \( \text{diag}(\mathcal{P}) = \{(p, p) | p \in \mathcal{P}\} \).

We say that \( \mathcal{A}_G \in \mathcal{G}(\mathbb{R}^n \times \mathbb{R}^n) \) is the matrix induced by the proximity graph \( G(\mathcal{P}) \) if its entries are non-negative, has non-zero diagonal terms, and the entry \( a_{ij} \neq 0 \) if and only if \( (p_i, p_j) \in \mathcal{E} \). In case that the proximity graph is undirected, then \( (p_i, p_j) \in \mathcal{E} \Rightarrow (p_j, p_i) \in \mathcal{E} \). In an abuse of notation, we will denote the edges of the graph \( G \) by either \( \mathcal{E} \) or the function \( \mathcal{E}_G \) that defines the set.

**Lemma 2:** If \( G(\mathcal{P}) \) is connected, then \( \mathcal{A}_G \) is primitive.

**Proof:** By Lemma 1 it is enough to show the result when the positive entries of the matrix \( \mathcal{A}_G \) are equal to 1. Observe that the \((i, j)\) entry in the product \( \mathcal{A}_G^k \) is given by

\[
\sum_{\{i_1, i_2, \ldots, i_k\} \subseteq \{1, \ldots, n\}} a_{i_1i_2}\ldots a_{i_{k-1}i_j},
\]

which will be positive if and only if it is possible to arrive from position \( p_i \) to position \( p_j \) passing through at most \( k \) different vertices (in fact, the entry \((i, j)\) corresponds to the number of ways of doing so). Since \( G(\mathcal{P}) \) is connected, the result follows.

III. Model

Let \( \mathcal{R} \) be a robotic network as defined in [25]. Let \( \{a_i\}_{i=1}^N \) denote the set of agents and let \( \{q_i\}_{i=1}^N \in F(\mathbb{R}^n) \) be their positions with respect to a fixed coordinate frame \( \mathcal{Q} \). For simplicity we will denote the set of points with respect to the frame \( \mathcal{Q} \) simply as \( \mathcal{R} \). We assume that the agents have no knowledge about \( \mathcal{Q} \).

We also assume that each of the robots is capable of identifying the agents that satisfy a certain criterion \( \mathcal{C} \). This induces the relationship \( \sim \) that defines the edges in the proximity graph \( G(\mathcal{R}) \). In other words, \( (p_i, p_j) \in \mathcal{E}_G(\mathcal{R}) \) if and only if \( p_i \sim p_j \). Observe that \( \sim \) is not necessarily a symmetric relation. For each agent \( a_i \), we define the set of its neighbors \( \mathcal{N}_i \) as

\[
\mathcal{N}_i = \{a_j \in \mathcal{R} | a_i \sim a_j, a_i \neq a_j\}.
\]

Some examples of \( \mathcal{C} \) are being closer than \( d \) or being Voronoi neighbors. We assume that each agent \( a_i \) can correctly estimate the positions of its neighbors with respect to its local coordinate frame \( \mathcal{Q}_i \). While this paper focuses on deterministic case, the case when the estimates of the positions are corrupted by noise is currently under investigation. Simulations show that our algorithm is quite robust. For example, we have achieved rendezvous in formations of 80 agents deployed uniformly in a square of side 12 when the measurement noise is distributed as \( \mathcal{N}(0, 100) \).

If rather than observing and estimating the position of the agents (which can be done, for instance, using laser or ultrasonic sensors) we allow the agents to exchange their locations, we have to impose the existence of a universal frame \( \mathcal{Q} \) known to all the robots. In what follows, we assume that there is no communication between the robots, and hence each agent observes and correctly estimates the location of its neighbors relative to its local coordinate frame. The goal of the network is to achieve rendezvous: all the agents should converge to a common configuration.

Based on the information the agents gather about their neighbors they update their position, with respect to the coordinate frame \( \mathcal{Q}_i \), as

\[
q_i[m + 1] = q_i[m] + u_i[m].
\]

We will introduce a control law \( u_i \) for the motion of agent \( a_i \) based on distributed consensus algorithms. As will be shown in the next section, under the assumption of connectivity of \( G(\mathcal{R}) \), the convergence of the formation to rendezvous will be guaranteed.

Although, as discussed in [26], in order to achieve average consensus it is necessary to ensure synchrony in the network, consensus protocols that do not solve the averaging problem (which is typically the case with the rendezvous) can be implemented in an asynchronous scenario. Our algorithm, thus, will work in both synchronous and asynchronous settings; synchrony will be imposed only if the point to which the agents converge should be some sort of average or not. Alternative rendezvous algorithms for both the synchronous and the asynchronous cases can be found in the cited literature.

IV. CONSENSUS-BASED RENDEZVOUS

Algorithm 1 describes Consensus-Based Rendezvous (CBR). The constant \( \rho \), introduced in Line 4 of Algorithm 1 is assumed to be a common constant for all the agents. As we will discuss later, the algorithm will also converge if each agent chooses its own \( \rho_i \) independently from the other robots.

Although it appears that the control input \( u_i \), and therefore the next position \( q_i[m + 1] \), will be dependent on the coordinate frame \( \mathcal{Q}_i \) that agent \( a_i \) chooses, this is not the case. The following lemma formalizes this point.

**Lemma 3:** The evolution of each agent is independent of the local frame \( \mathcal{Q}_i \) it chooses to implement the CBR algorithm.
Algorithm 1 Consensus-Based Rendezvous

Require: Agent $a_i$ at time $m$
1: Identify neighbors $N_i = \{a_{i1}, a_{i2}, \ldots, a_{ir_i(m)}\}$.
2: Evaluate the position $p_{i, i_j}, 1 \leq j \leq r_i(m)$ of each neighbor, and its own position $p_{i, i_0}$ with respect to a local coordinate frame $Q_i$.
3: Compute $p_i = \sum_{j=0}^{r_i(m)} \lambda_{i, j} p_{i, j}$, where $\lambda_{i, j} > \epsilon > 0$ and $\sum_{j=0}^{r_i(m)} \lambda_{i, j} = 1$.
4: Set $u_i[m] = \rho (p_i - p_{i, i_0})$, where $0 < \rho < 1$.

Proof: Let $q_1, q_2, \ldots, q_{r_i(m)}$ be the positions of the neighbors of $a_i$ with respect to the fixed coordinate frame $Q$. Let $p_{i, i_1}, p_{i, i_2}, \ldots, p_{i, i_{r_i(m)}}$ be those positions when expressed with respect to the coordinate frame $Q_i$. We will denote the point $p_{i, i_j}$ as $p_{i, j}$ for simplicity. The convex combination in line 3 of the CBR is given by

$$p_i = \lambda_{i, 0} p_{i, 0} + \lambda_{i, 1} p_{i, 1} + \cdots + \lambda_{i, r_i(m)} p_{i, r_i(m)} \tag{4}$$

Recall that the change on the coordinate system between $Q_i$ and $Q$ is given by a translation $q'_i$ and a rotation $O$. The expression for (4) in the coordinate frame $Q$ is thus given by:

$$\lambda_{i, 0} (q'_i + O p_{i, 0}) + \cdots + \lambda_{i, r_i(m)} (q'_i + O p_{i, r_i(m)}) = \sum_{j=0}^{r_i(m)} \lambda_{i, j} q'_i + O p_{i, j} \tag{5}$$

which means that the position to which $a_i$ moves depends only on the coefficients $\lambda_{i, j}$ and is independent of the choice of the coordinate frame.

Remark: Note that we did not use the orthonormality of the matrix $O$ at any point. In fact, the result is still valid for any transformation induced by an invertible matrix $O'$.

Observe that, due to Lemma 3, we can stack together the equations for each agent, and write the evolution of the system in matrix form as $q[m+1] = I q[m] + U[m]$, where $I \in \mathbb{R}^{N \times N}$ is the identity matrix, $q \in \mathbb{R}^{N \times n}$ and $U \in \mathbb{R}^{N \times n}$. Under the assumption that the proximity graph $G(R)$ is connected, the induced matrix $A_G$, where $\lambda_{i, j} = \lambda_{j, i}$, is a consensus matrix. This makes $U[m] = \rho (A_G - I) q[m], \rho \in (0, 1)$ and thus we can rewrite the discrete time system as

$$q[m+1] = [(1 - \rho) I + \rho A_G] q[m]. \tag{6}$$

Lemma 4: Let $C \subseteq \mathbb{R}^{N \times N}$ be the set of all consensus matrices in $\mathbb{R}^{N \times N}$. $C$ is closed under finite convex combinations.

Proof: It is enough to show that if $A, B \in C$, then $C = \lambda A + (1 - \lambda) B \in C$ when $\lambda \in (0, 1)$. Since $C$ has non-negative entries, and $C1 = (\lambda A + (1 - \lambda) B)1 = 1$, it follows that $C$ is stochastic. To show that it is also primitive, observe that since $A$ and $B$ are both primitive, there is a positive integer $k$ such that both $A^k$ and $B^k$ have only positive entries. Since $C^k = (A + B)^k = A^k + B^k + \cdots$ and the entries of both $A$ and $B$ are non-negative, the entries in each term of the product will also be non-negative. Since both $A^k$ and $B^k$ both have only positive entries, then $C^k$ will have only positive entries, and hence it will be primitive.

Remark: Observe that even if only one of the matrices $A, B$ is a consensus matrix, then $C \in C$ as long as both matrices are stochastic.

Applying Lemma 4, under the assumption that $G(R)$ is connected and considering the previous remark, we conclude that we can write (6) as

$$q[m+1] = C[m] q[m], \tag{7}$$

where $C = [(1 - \rho) I + \rho A_G]$ is a consensus matrix.

Lemma 5: If agent $a_i$, chooses its own $\rho_i$, the evolution of the network can be written as in (7), where the resulting matrix $C$ is a consensus matrix.

Proof: Recall that $a_i$ updates its position according to (3). By replacing $u_i$ with the equivalent expression in Line 4 of Algorithm 1, we obtain

$$q_i[m+1] = q_i[m] + \rho_i \left( \lambda_{i, 0} q_i[m] + \sum_{j \in N_i} \lambda_{i, j} q_j[m] - q_i[m] \right), \tag{8}$$

which can be rewritten as

$$q_i[m+1] = (1 - \rho_i) q_i[m] + \rho_i \left( \lambda_{i, 0} q_i[m] + \sum_{j \in N_i} \lambda_{i, j} q_j[m] \right) \tag{9}$$

Then, if we stack together this expressions for all the agents, we can write the evolution of the system in matrix form as in (7). Furthermore, the resulting matrix $C$ is stochastic, and since $G(R)$ is connected, Lemma 2 ensures it is also primitive. Hence, $C$ is a consensus matrix.

Now, we state the main result of this section.

Theorem 6 (Convergence of the CBR): A discrete time system evolving according to (7) reaches consensus.

Proof: Since the matrix $C[m]$ is a consensus matrix, the result follows from the properties of consensus algorithms. For details we refer the reader to either [11] or [7].

Observe that Theorem 6 implies that all the robots in $R$ converge towards a common configuration so they achieve rendezvous.

Remark: The constant $\epsilon$ in Line 3 is needed to guarantee the convergence of the algorithm. For practical implementation this condition always holds due to the finite numerical precision of the processor performing computations.

A. Robustness of the algorithm

So far we assumed that the proximity graph $G(R)$, induced by the formation at time $m$, $m \geq 0$, is always connected and that each agent is capable of correctly detecting all its neighbors. As mentioned above, in this paper we do not consider location estimation errors. So we focus on what
happens when the agent misses one (or more) of its neighbors as the system evolves.

As above, we denote for simplicity the state of the formation $\mathcal{R}$ at time $m$ by $\mathcal{R}[m]$. We denote the proximity graph induced at time $m$ by $\mathcal{G}(\mathcal{R})[m]$, and its set of edges by $\mathcal{E}_{\mathcal{G}}[m]$.

We define the real proximity graph of the formation $\mathcal{R}$ at time $m$ as the graph whose vertices are the agents in the formation, and whose edge set $\mathcal{E}_{\mathcal{G}}[m] \subseteq \mathcal{E}_{\mathcal{G}}[m]$ contains the pair $(a_i, a_j)$ if and only if the agent $a_i$ does identify the agent $a_j$ as one of its neighbors. Since communication is not allowed between the agents, the fact that agent $a_i$ misses $a_j$ does not imply that $a_j$ also misses $a_i$ even when $\sim$ is symmetric. Observe that when no identification errors occur, the real proximity graph coincides with $\mathcal{G}(\mathcal{R})[m]$.

We say that the formation is strongly connected if there exists a positive integer $T$ such that, for every $m > 0$, the graph with vertices induced by $\mathcal{R}$, and edge set defined by the union $\mathcal{E} = \bigcup_{i=0}^{T-1} \mathcal{E}_{\mathcal{G}}[m + i]$, is connected.

Note that since the matrix $C[m]$ has entries $c_{i,j} = \lambda_{i,j}$, the weight assigned by agent $a_i$ to the position of agent $a_j$ is always stochastic. Under the assumption that the underlying graph for $\mathcal{R}$ (i.e., the real proximity graph) is strongly connected, we can state the following theorem which follows directly from the theory of consensus algorithms. The proof can be found for example in [5], [7], [11].

Theorem 7: Under the assumption that the real proximity graph induced by the formation is strongly connected, Algorithm 1 is robust to failures by the agents to detecting some neighbors and the system achieves rendezvous.

Remark: Observe that the condition on strong connectivity is reasonable since without it we would be considering disconnected components in the formation, making it impossible to achieve rendezvous.

V. AN ALTERNATIVE FORMULATION FOR THE CBR ALGORITHM

Recall that the agents are updating their positions according to an arbitrary convex combination of neighbors’ positions with $\lambda_{i,i}$ strictly positive and all the $\lambda_{i,j}$ bounded below by some $\epsilon$. We now present an alternative formulation for the CBR Algorithm 1.

Lemma 8: Let $C$ be a convex polytope in $\mathbb{R}^n$, and let $p \in C$ be any interior point. Then $p$ can be written as a convex combination of some of the vertices of $C$.

Proof: We proceed by induction on the dimension $n$. If $n = 1$, then any point in the segment defined by $C$ can be written as a convex combination of its extremes. Suppose the result holds for dimension $k + 1$. Take any line that passes through $p$. Let $q$ and $r$ be the intersections of such line with the boundary of $C$. Since each one of this two new points lies on a face of a convex polytope in $\mathbb{R}^{k+1}$, the respective face can be embedded in $\mathbb{R}^k$. By induction hypothesis, both $q$ and $r$ can be written as a convex combination of the vertices that define such faces. Finally, since $p$ can be written as a convex combination of $q$ and $r$, the result follows.

Lemma 9: Let $\mathcal{P} = \{p_1, \ldots, p_m\} \in \mathbb{F}(\mathbb{R}^n)$, and let $\mathcal{H}(\mathcal{P})$ be the convex hull of $\mathcal{P}$. Let $p$ be any point either in the interior or in the boundary of $\mathcal{H}(\mathcal{P})$, and let $q$ be an interior point of $\mathcal{H}(\mathcal{P})$. Then, there exists $\lambda_0, \lambda_1, \ldots, \lambda_m$ such that $\lambda_0 > 0$, $\lambda_i \geq 0$, $1 \leq i \leq m$, $\sum_{i=0}^{m} \lambda_i = 1$ and $q = \lambda_0 p + \cdots + \lambda_m p_m$.

Proof: If $p = q$, then let $\lambda_0 = 1$, $\lambda_i = 0$, $1 \leq j \leq m$. Suppose then $p \neq q$. Let $t$ be the intersection of the ray $pq$, starting at $p$, with $\mathcal{H}(\mathcal{P})$. Denote by $T \subset \mathcal{P}$ the vertices of the face on which $t$ lies. Since $T$ defines a face of a convex polytope in $\mathbb{R}^{k+1}$, such face can be embedded in $\mathbb{R}^k$. By Lemma 8, the point $t$ can be written as a convex combination of points in $T$. Since $q$ lies in the segment $pt$, then $q$ can be written as a convex combination of $p$ and $t$, with a non-zero coefficient of $p$.

Lemma 10: Let $\mathcal{P} \in \mathbb{F}(\mathbb{R}^n)$. For any given $\delta > 0$, there exists $\epsilon > 0$ such that for every $p_0, q \in \mathcal{H}(\mathcal{P})$, there is a subset $\mathcal{P}' = \{p_1, \ldots, p_r\} \subseteq \mathcal{P}$, possibly empty, such that if $q = \sum_{i=0}^{r} \lambda_i p_i$ with $\lambda_i > 0$ and $\sum_{i=0}^{r} \lambda_i = 1$, there is $q^* = \sum_{i=0}^{r} \lambda_i^* p_i$ with $\lambda_i^* \geq 0$ such that $\sum_{i=0}^{r} \lambda_i^* = 1$, $\|q^* - q\| < \delta$, and if $\lambda_i^* > 0$ then $\lambda_i^* \geq \varepsilon$ for $0 \leq i \leq r$.

Proof: Since $\mathcal{P}$ is finite, it is bounded. Therefore, there is a positive real number $k$ such that if $x \in \mathcal{P}$, then $|x| < k$. From the proof of Lemmas 8 and 9, $\mathcal{P} \subset \mathbb{R}^n$ implies that $r = |\mathcal{P}'| \leq n$ (i.e. we can write the convex combination with at most $n$ elements). Let $\delta^* = \min(\delta/2, k/2)$. We will show that $\delta < \delta^*/(nk)$ works. Fix one such $\delta$. If $p_0 = q$, then $\mathcal{P}' = \emptyset$, $\lambda_0 = 1$ and we are done. Suppose then that $p_0 \neq q$. Construct a convex combination as in Lemma 9:

$$q = \sum_{i=0}^{r} \lambda_i p_i.$$

Let $\Lambda_0 = \{j | \lambda_j < \epsilon\}$ and let $\Lambda_1 = \{i | \lambda_i \geq \epsilon\}$. Since $ne < 2nc < 2\delta^*/k \leq 1$, then $\Lambda_1$ is non-empty. We will construct the new coefficients $\lambda^*$ by modifying the coefficients $\lambda$ in (10). Rearrange (10) as

$$q = \sum_{j \in \Lambda_0} \lambda_j^* p_j + \sum_{i \in \Lambda_1} \lambda_i p_i.$$

Let $\sum_{j \in \Lambda_0} \lambda_j = \lambda = ne + l$ where $m \in \mathbb{Z}$ and $0 \leq l < \epsilon$. Suppose $\lambda \geq \epsilon$. This implies that $|\Lambda_0| > m + 1$. Choose a set with $m$ elements $J = \{j_0, j_1, \ldots, j_m\} \subset \Lambda_0$ such that if $0 \in \Lambda_0$ then $0 \in J$. For $j \in J$ let $\lambda_j^* = \epsilon$. For $j' \in \Lambda_0 \setminus J$, we set $\lambda_{j'}^* = 0$. Choose $i \in \Lambda_1$, set $\lambda_i^* = \lambda_i + l$ and for $i' \in \Lambda_1 \setminus \{i\}$ set $\lambda_{i'}^* = \lambda_{i'}$. Observe that $\sum_{j=1}^{m} \lambda_j^* = 1$. Let $q^* = \sum_{i=0}^{r} \lambda_i p_i$. We will show that $\|q^* - q\| < \delta$. Note that

$$q^* - q = \left( \sum_{j \in J} \lambda_j^* p_j + \sum_{i \in \Lambda_1 \setminus \{i\}} \lambda_i^* p_i \right) - \left( \sum_{j \in \Lambda_0} \lambda_j p_j + \sum_{i \in \Lambda_1} \lambda_i p_i \right)$$

$$= \sum_{j \in J} \lambda_j p_j - \sum_{j \in \Lambda_0} \lambda_j p_j + \sum_{i \in \Lambda_1} \lambda_i p_i.$$
and we can bound its norm as

$$
\|q^*-q\| = \left\| \sum_{j \in J} \epsilon p_j - \sum_{j \in \Lambda_0} \lambda_j p_j + r p_i \right\|
\leq \left\| \sum_{j \in J} \epsilon p_j \right\| + \left\| \sum_{j \in \Lambda_0} \lambda_j p_j \right\| + \|r p_i\|
\leq \epsilon \left\| p_j \right\| + \epsilon \left\| p_j \right\| + \epsilon \|p_i\|
< (2 \left| \Lambda_0 \right| + 1) k \epsilon \leq (2(r - 1) + 1) k \epsilon
< 2n k \epsilon < 2\delta^* \leq \delta.
$$

We thus have a convex combination for a point \(q^*\) which is \(\delta\)-close to our original point \(q\) and has the desired properties.

We denote the constructed \(q^*\) by \(\lambda (p_0, q)\). The case when \(\lambda < \epsilon\) is treated similarly, and due to space limitations we omit the details.

Algorithm 2 is an alternative formulation for the CBR algorithm. For simplicity, we will denote \(\mathcal{H}(N_i \cup \{a_i\})\) as \(\mathcal{H}(N_i)\). Our algorithm gives the agent \(a_i\) the freedom to choose any point inside \(\mathcal{H}(N_i)\), and then move to a suitable location \(\delta\)-close to it. The reason we have to impose such condition is to guarantee the convergence of the formation to rendezvous, since for (2) to hold, the existence of a positive lower bound for all the positive entries of all the matrices \(A_i\) is a sufficient condition. Furthermore, if such a condition is not imposed, counterexamples can be constructed as shown in [13].

Algorithm 2: Alternative Consensus-Based Rendezvous

**Require:** Knowledge of \(k, \delta\) and \(n\).

**Require:** Agent \(a_i\) at time \(m\).

1. Identify \(N_i = \{a_{i1}, a_{i2}, \ldots, a_{ir_i(m)}\}\).
2. Evaluate \(\mathcal{H}(N_i)\).
3. Set \(p_i^*\) to be any arbitrary interior point in \(\mathcal{H}(N_i)\). Choose \(\lambda (a_i, p_i^*)\) as in Lemma 10.
4. Set \(u_i[m] = p_i - (p_i - p_i, 0)\), where \(p_i, 0\) is the position of \(a_i\) with respect to the same coordinate frame the agent used to define \(p_i\), and \(p_i \in (0, 1)\).

**Theorem 11:** Algorithm 2 is a special case of Algorithm 1.

**Proof:** Lemmas 8 and 9 guarantee that regardless of the point that is chosen by each agent, there exists a matrix \(C\) as in (7), such that it has non-zero entries on its main diagonal and reflects the decision made by each robot in \(R\). Since \(a_i\) is moving toward an interior point of \(\mathcal{H}(N_i)\) (hence in \(\mathcal{H}(R)\)) the bound \(k\) used in the proof of Lemma 10 holds for every step time in the evolution of \(R\). Hence Lemma 10 also guarantees that we can actually choose a point \(q^*\) that is \(\delta\)-close to the original \(q\), such that there is a lower bound for a consensus matrix \(C\) representing the evolution of \(R\), and such lower bound is common for every time \(m > 0\).

We showed that the agents achieve rendezvous if agent \(a_i\) chooses to move within distance \(\delta\) of any point that belongs to the interior of \(\mathcal{H}(N_i)\). However, we can actually allow the agents to choose a point on the boundary of \(\mathcal{H}(N_i)\) and then move to a point in the interior that is within distance \(\delta\). For instance, the circumcenter algorithm presented originally in [1] and then revisited in [2], in which the agents evolve towards the circumcenter of the set of neighbors, can be seen as a special case of Algorithm 2 where the agents either move towards the circumcenter or to a point which is at most \(\delta\) away from it.

Another particular realization of interest is what we call the center of mass algorithm, in which each agent \(a_i\) assigns the same weight to itself and each of its neighbors so that the algorithm converges at each time step to the center of mass of \(N_i\).

**VI. Simulations**

To test the algorithm, 30 agents were distributed uniformly in a square of side 10. At each time step the agents randomly choose a convex combination of the configurations of its neighbors. Agent \(a_i\) has a probability \(p_i, j\) of failing to detect neighbor \(a_j\). By adjusting the sensing radius of the agents the real proximity graph ranges from a complete graph to the case in which the graph is completely disconnected.

As expected, when \(G(R)\) is strongly connected the formation achieves rendezvous, as shown in Figure 1. When \(G(R)\) consists of connected components which are disconnected between them, the result coincides with the theoretical conclusion that in each connected component rendezvous is achieved. This situation is shown in Figure 2.

Since Algorithm 1 is a special case of consensus algorithm it has the convergence rate of consensus protocols. Therefore, if the proximity graph is fixed the convergence rate is exponential [22]. Partial results on the convergence rate of consensus algorithms when the topology of the graph changes were first proposed in [27], [28]. More recently, it was proved in [29] that even when the topology of the communication graph changes the convergence is still exponential. In our simulations, 60 to 80 iterations were needed for the agents to converge to rendezvous.
 formal family of algorithms where an agent $a_i$ moves towards a point within distance $\delta$ from a point in the interior of $H(N_i)$, the convex hull of the locations of all its neighbors. We showed that this algorithm is a particular realization of moving towards an arbitrary convex combination of the locations of the neighbors, which implies that this procedure is equivalent to a consensus algorithm. Hence the convergence of the agents is guaranteed under the assumptions needed by the consensus protocols. The proposed algorithm is fully distributed (hence scalable), stable, robust under failures and requires no communication between the agents as long as they are capable of estimating relative positions of their neighbors. The algorithm also generalizes previous rendezvous algorithms in the sense that they can be viewed either as a particular realization of our algorithm, or that the assumptions on the information each agent in the formation has are relaxed. The algorithm can be implemented in both synchronous or asynchronous model of communication. While we assumed that each agent can estimate the location of its neighbors exactly, simulations show that the algorithm is robust to sensor noise. Formal analysis of the algorithm with noisy estimates is currently under investigation.

**REFERENCES**


