Technical Notes and Correspondence

Asynchronous Consensus Protocols Using Nonlinear Paracontractions Theory

Lei Fang and Panos J. Antsaklis, *Fellow, IEEE*

***Abstract*—Several consensus protocols have been proposed in the literature and their convergence properties studied via a variety of methods. In all these methods, the communication topologies play a key role in the convergence of consensus processes. In this note, based on asynchronous iteration methods for nonlinear paracontractions, we establish a new result which shows that consensus is reachable under directional and time-varying topologies by using asynchronous nonlinear protocols. Our result makes use of the confluent iteration graph which unifies various communication assumptions and contributes to a fundamental understanding of convergent consensus processes. This result extends existing ones in the literature and has many potential applications. As an illustration, we consider a special case of our model and discuss the robot rendezvous problem via a center-of-gravity algorithm.**

***Index Terms*—Asynchronous iterations, consensus, nonlinear paracontractions.**

I. Introduction

In recent years, there has been growing interest in the coordinated control of multiagent systems. Coordination normally implies synchronizing agent actions and exchanging information among the agents. Consensus seeking among agents therefore becomes one of the fundamental problems in control theory, namely to reach an agreement regarding a certain quantity of interest that depends on the state of all agents [1]–[5]. This need stems from the fact that in order for agents to coordinate their behaviors, they need to use some shared knowledge about variables such as direction, speed, time-to-target etc.

Recent research reveals two trends beyond the fundamental convergence analyses of consensus algorithms. One trend concerns the performance issues of such algorithms. Xiao and Boyd showed that the convergence speed of a consensus process can be increased by changing the weights on communication links [6]. For large or random networks, rewiring only a small portions of existing links would result in a fast consensus seeking [7]. A multihop relay protocol was designed to achieve a better convergence performance without changing the network topology [8]. Another trend is to study the problem in the presence of unreliable (e.g., delay, dropout, and noise effects) and/or dynamically changing communication topologies. Such setups render the problem under consideration more realistic. The effect of communication delays was addressed in [2], [9], [10] for fixed topologies and in [11], [12] for time-varying topologies. Consensus in the presence of noise was considered in [13], [14]. Instead of addressing various communication imperfections separately, the asynchronous framework [4], [15]–[19] provides another way to solve the problem where a number of agents update their states asynchronously by using (possibly outdated) information from their neighbors. Under asynchronous communications, heterogeneous agents, time-varying communication delays and packet dropout can all be taken into account in the same framework.

In this note, we study the consensus problem from a nonlinear paracontracting operator point of view [20]. Specifically, we show that consensus is reachable under directional and time-varying topologies with asynchronous nonlinear protocols. The confluent iteration graph is introduced to incorporate various communication assumptions and it proves to be fundamental in understanding the convergence of consensus processes.

The general convergence result on asynchronous nonlinear protocols is new and has not been presented before. This result is more general compared to asynchronous linear cases considered in [4], [19] and synchronous nonlinear cases in [21]. Another side contribution is that we formulate and address the consensus problem in a geometric framework, where the problem is typically addressed using matrix theory [12], [22], Lyapunov functions [11], [21], or graph theory [19]. In this framework, the convergence of asynchronous consensus protocols is studied with respect to the properties of a pool of paracontracting operators from which a common fixed point is extracted.

II. Problem Formulation

A. Synchronous vs. Asynchronous Consensus Protocols

We consider a set $\mathcal{N} = \{1, 2, \ldots, m\}$ of agents embedded, at each discrete time t, in a directed graph $G(t) = (\mathcal{N}, \mathcal{E}(t))$. The directed graph $G(t)$ is used to model the interaction topology among a group of agents, where every graph node corresponds to an agent and a directed edge e_{ij} represents a unidirectional information exchange link from agent i to agent j. That is, agent j can receive information from agent i. The interaction graph represents the communication pattern at a particular time and it is time-dependent.

Each agent i starts with a initial state $x_i(0) \in \mathbb{R}^n$. The (linear) consensus protocol updates $x_i(t)$ according to

$$x_i(t+1) = \sum_{j=1}^{m} f_{ij}(t) x_j(t) \tag{1}$$

where $f_{ij}(t) \geq 0$ and $\sum_{j=1}^{m} f_{ij}(t) = 1$, for all i, t. Whenever $f_{ij}(t) > 0$, agent j communicates its current state $x_j(t)$ to agent i (the directed edge $(j,\ i) \in \mathcal{E}(t)$). Each agent i updates its own state, by forming a weighted average of its own state and the states it has just received from other agents. Notice that each agent has access to its own state, i.e., $(i,\ i) \in \mathcal{E}(t)$ for all t. We say that *consensus* (or *agreement*) is reached if $\|x_i(t) - x_j(t)\| \to 0$ as $t \to \infty$ for all $i,\ j$.

The consensus protocol (1) is synchronous in the sense that all the agents update their states at the same time using the latest state values from their neighbors. From a practical point of view, since a central synchronizing clock may not exist and communication links are created and fail dynamically, it is of interest to consider asynchronous consensus protocols.

In the asynchronous setting the order in which states of agents are updated is not fixed and the selection of previous values of the states used in the updates is also not fixed. Now let $t_0 < t_1 < \cdots < t_k < \cdots$ be the time instants when the state of the multiagent system undergoes change. Let $x_i(k)$ denote the state of agent i at time t_k. The index k

Manuscript received October 10, 2006; revised August 21, 2007. Current version published November 05, 2008. Recommended by Associate Editor F. Bullo.

L. Fang is with Magnetar Capital LLC, Evanston, IL 60201 USA (e-mail: justinlfang@gmail.com).

P. J. Antsaklis is with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: antsaklis.1@nd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2008.2007146

is often called the *event-based discrete time index*. The dynamics of asynchronous systems can be written as

$$x_i(k+1) = \begin{cases} F^i(x_1(s^1(k)), \dots, x_m(s^m(k))) & \text{if } i = I(k) \\ x_i(k) & \text{if } i \neq I(k) \end{cases} \quad (2)$$

where $s^j(k)$, $k = 0, 1, \dots$, $j = 1, \dots, m$, are sequences from $\mathbb{N}_0$, with $s^j(k) \leq k$, $\forall j, k$, and $I(k)$, $k = 0, 1, \dots$ is a sequence from $\{1, \dots, Q\}$. Here, Q is the total number of possible interaction topologies observed during the consensus process. By properly arranging the index k, we can assume $I(k)$ is a singleton without loss of generality. We refer to $k - s^j(k)$ as *iteration delays* and $I(k)$ as *updating sets*. Notice that F^i can be any (possible nonlinear) paracontracting operators (see Def. 2 in Section III). In particular, linear consensus protocols take the following form:

$$F^i(x_1(s^1(k)), \dots, x_m(s^m(k))) = \sum_{j=1}^{m} f_{ij}(k) x_j(s^j(k)).$$

We shall see below how, at least in some specific set-ups, the same global behavior of the multiagent system in terms of convergence to a common equilibrium takes place also in the asynchronous case. To develop conditions under which all agents reach consensus requires the analysis of the asymptotic behavior of the asynchronous process (2). In spite of the complexity of this process, it is possible to capture its salient features using the theory of paracontractions and confluence developed in [20] and this is done in the next section.

III. Theory of Paracontractions and Confluence

A. Asynchronous Iteration

The consensus problem can be regarded as a special case of finding *common fixed points* (not necessarily unique) of a finite set of paracontracting multiple point operators. That is, all the operators are defined on (different) products of $\mathbb{R}^n$. To avoid divergent phenomena, asynchronous iterations which fulfill certain coupling assumptions called *confluence* are considered.

We next summarize some results from the theory of paracontractions and confluence which are useful in deriving sufficient conditions for consensus.

Definition 1: Let $\mathbb{I}$ be a set of indices, $m \in \mathbb{N}$ a fixed number, and $\mathcal{F} = \{F^i | i \in \mathbb{I}\}$ be a pool of operators $F^i : D^{m_i} \subset \mathbb{R}^{nm_i} \to D$, where $m_i \in \{1, \dots, m\}$, $\forall i \in \mathbb{I}$, and $D \subset \mathbb{R}^n$ is closed. Furthermore, let $\mathcal{X}_{\mathcal{O}} = \{x(0), \dots, x(-M)\} \subset D$ be a given set of vectors (M is the number of initial conditions). Then, for sequences $\mathcal{I} = I(k)$ ($k = 0, 1, \dots$) of elements in $\mathbb{I}$, $\mathcal{S} = \{s^1(k), \dots, s^{m_{I(k)}}(k)\}$, $k = 0, 1, \dots$, of m_i-tuple from $\mathbb{N}_0 \cup \{-1, \dots, -M\}$ with $s^l(k) \leq k$ for all $k \in \mathbb{N}_0$, $l = 1, \dots, m_{I(k)}$, we study the asynchronous iteration given by

$$x(k+1) = F^{I(k)}\left(x(s^1(k)), \dots, x(s^{m_{I(k)}}(k))\right), k = 0, 1, \dots. \quad (3)$$

An asynchronous iteration corresponding to $\mathcal{F}$, starting with $\mathcal{X}_{\mathcal{O}}$ and defined by $\mathcal{I}$ and $\mathcal{S}$ can be denoted by $(\mathcal{F}, \mathcal{X}_{\mathcal{O}}, \mathcal{I}, \mathcal{S})$. A *fixed point* ξ of a multiple point operator $F : \mathbb{R}^{nm} \to \mathbb{R}^n$ is a vector $\xi \in \mathbb{R}^n$ which satisfies $F(\xi, \dots, \xi) = \xi$, and a *common fixed point* of a pool is a fixed point of all its operators in this sense.

In the remainder of this section, we introduce some conditions for the elements of an asynchronous iteration to study its convergence property.

B. Paracontracting Operators

We first introduce criteria of contraction for the pool $\mathcal{F}$, where a common fixed point is to be found.

Definition 2: Let $\mathcal{F}$ be a pool of operators as in Definition 1.

(i) If for all $i \in \mathbb{I}$, $X, Y \in D^{m_i}$ and a norm $\|\cdot\|$

$$\left\|F^i(X) - F^i(Y)\right\| < \max_j \left\|x^j - y^j\right\| \quad \text{or}$$
$$F^i(X) - F^i(Y) = x^j - y^j,\ \forall j \in \{1, \dots, m_i\},$$

then $\mathcal{F}$ is called *strictly nonexpansive* on D.

(ii) If for all $i \in \mathbb{I}$, $X \in D^{m_i}$ and a norm $\|\cdot\|$, F^i is continuous on D^{m_i}, then $\mathcal{F}$ is *paracontracting* on D, if for any fixed point $\xi \in \mathbb{R}^n$ of F^i,

$$\left\|F^i(X) - \xi\right\| < \max_j \left\|x^j - \xi\right\| \quad \text{or}$$
$$X = (x, \dots, x) \ \text{and} \ x \ \text{is a fixed point of} \ F^i.$$

It is easy to see that every strictly non-expansive pool is paracontracting. For concrete paracontracting operators examples, the authors refer readers to [20], [23].

Next we investigate $\mathcal{I}$ and $\mathcal{S}$.

C. Regularity Assumptions

Definition 3: Let $(\mathcal{F}, \mathcal{X}_{\mathcal{O}}, \mathcal{I}, \mathcal{S})$ be an asynchronous iteration. Then

(i) $\mathcal{S}$ is called *regulated*, if

$$s := \max_{k,l} k - s^l(k) \quad (4)$$

exists.

(ii) $\mathcal{I}$ is an *index-regulated sequence*, if for all $i \in \mathbb{I}$ there is a number $c_i \in \mathbb{N}_0$ such that for all $k \in \mathbb{N}_0$

$$i \in \{I(k)\} \cup \{I(k+1)\} \cup \dots \cup \{I(k+c_i)\}. \quad (5)$$

(iii) $\mathcal{I}$ is called *regulated*, if there is a number $c \in \mathbb{N}_0$, such that for all $k \in \mathbb{N}_0$

$$\{I(k)\} \cup \{I(k+1)\} \cup \dots \cup \{I(k+c)\} = \mathbb{I}. \quad (6)$$

D. Iteration Graph and Confluence

The communication assumptions define the coupling among agents or, more generally, the coupling of an iteration process. The existing assumptions often rely on interaction graphs to describe the "spatial" coupling among agents. However, ambiguity arises when asynchronism (e.g., delays) is allowed since the "temporal" coupling cannot be described directly by interaction graphs. In the asynchronous setting, there is a need to differentiate the states of the same agent at different time instants.

To this end, we associate an *iteration graph* with the asynchronous iteration $(\mathcal{F}, \mathcal{X}_{\mathcal{O}}, \mathcal{I}, \mathcal{S})$. Every iteration, including initial vectors, gets a vertex, so the set of vertices is $V = \mathbb{N}_0 \cup \{-1, \dots, -M\}$. A pair (k_1, k_2) is an element of the set of edges E in the iteration graph (V, E), if and only if the k_1-th iteration vector is used for the computation of the k_2-th iteration vector.

Below we illustrate the concept of iteration graph through an example. The interaction topologies of a three-agent system at different time instants are shown in Fig. 1(a). The edge $(v_2,\ v_3)$ at time $t = 0$ describes that agent 2 communicates its state information to agent 3. From the interaction topology, it is not clear whether the latest state information of agent 2 or the delayed version was transmitted to agent 3. This ambiguity entails the introduction of iteration graph.

As shown in Fig. 1(b), the iteration graph uses event-based index k rather than t. Vertices -3, -2, -1 denote the initial conditions of agent v_3, v_2, v_1, respectively. At time $k = 0$, v_3 updates its state using v_2's state (but not its own state), that is $x_3(1) = F^{I(0)}(x_2(s^2(0))) = F^{I(0)}(x_2(-2))$, $F^{I(0)} \in \mathcal{F}$. By construction, we add to the iteration graph a vertex 0 and an edge from

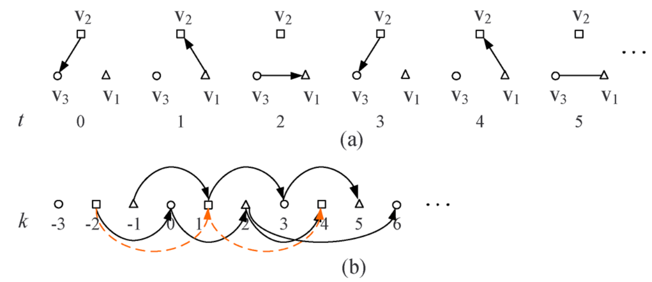

Fig. 1. Interaction topologies of an asynchronous system and its associated iteration graph. Nodes -3, -2, -1 denote initial conditions of agents v_3, v_2, v_1, respectively.

vertex -2 to vertex 0. Also assume that only v_2 can access its own state and v_3 and v_1 cannot. Therefore, we did not see an edge from vertex -3 to vertex 0 in the iteration graph. At time instant $k = 1$, v_2 iterates using its own state and v_1's state, corresponding to two edges, $(-1, 1)$ and $(-2, 1)$, in the iteration graph. The updating equation is $x_2(2) = F^{I(1)}(x_1(s^1(1), x_2(s^2(1)))) = F^{I(1)}(x_1(-1), x_2(-2))$. The bidirectional link at time $t = 5$ can be treated as two events which lead to two nodes (5 and 6) in the iteration graph. Remind that the above updating equations have the form of (2) but can be cast to the more general form of (3) via variable transformations, to be shown in the next section.

Remark 1: The *delay graph* in [19] is close in spirit to the iteration graph. The procedure to construct iteration graphs also reminds us the net unfolding method for Petri nets [24].

The confluent condition imposed upon an iteration graph means that every operator is sufficiently involved in the iteration process.

Definition 4: Let $(\mathcal{F}, \mathcal{X}_\mathcal{O}, \mathcal{I}, \mathcal{S})$ be an synchronous iteration. The iteration graph of $(\mathcal{F}, \mathcal{X}_\mathcal{O}, \mathcal{I}, \mathcal{S})$ is the digraph (V, E), whose vertices V are $\mathbb{N}_0 \cup \{-1, \ldots, -M\}$, and whose edges E are given by

$$(k, k_0) \in E, \text{ iff there is an } 1 \le l \le m_{I(k_0-1)} \\ \text{such that } s^l(k_0 - 1) = k.$$

$(\mathcal{F}, \mathcal{X}_\mathcal{O}, \mathcal{I}, \mathcal{S})$ is called *confluent*, if there are numbers $n_0 \in \mathbb{N}$, $b \in \mathbb{N}$ and a sequence b_k $(k = n_0, n_0 + 1, \ldots)$ in $\mathbb{N}$, such that for all $k \ge n_0$ the following is true:

i) For every vertex $k_0 \ge k$ there is a directed path from b_k to k_0 in (V, E);
ii) $k - b_k \le B$;
iii) $\mathcal{S}$ is regulated;
iv) for every $i \in \mathbb{I}$ there is a $c_i \in \mathbb{N}$ so that for all $k \ge n_0$ there is a vertex w_k^i in V, which is a successor of b_k and a predecessor of b_{k+c_i}, and for which is $I(w_k^i - 1) = i$.

E. A Convergence Theorem

A simplified version of the main result in [20] is now given.

Theorem 1: Let $\mathcal{F}$ be a paracontracting pool on $D \subset \mathbb{R}^n$, and assume that $\mathcal{F}$ has a common fixed point $\xi \in D$. Then any confluent asynchronous iterations $(\mathcal{F}, \mathcal{X}_\mathcal{O}, \mathcal{I}, \mathcal{S})$ converges to a common fixed point of $\mathcal{F}$.

IV. Nonlinear Asynchronous Consensus Protocols

The application of Theorem 1 involves three steps:

1 Formulate the original problem as an asynchronous iteration problem;
2 Verify the paracontracting property of the pool of operators;
3 Verify that the coupling among operators is confluent;

In the following, we show step by step how to cast asynchronous consensus problems into asynchronous iterations of form (3) so that Theorem 1 can be applied to obtain convergence results.

A. An Equivalent Formulation for Asynchronous Consensus Problems

In (3), the whole vector $x \in D \subset \mathbb{R}^n$ is updated at every iteration step. All components of x have the same delay. The asynchronous consensus updating (2) does not have the above characteristics. In order to convert (2) into a form of (3), we introduce an auxiliary system with new states $y(k)$.

Lemma 1: The asynchronous consensus problems defined in (2) can be formulated as asynchronous iterations.

Proof: Assume without loss of generality that the numbering of $s^l(k)$, $k = 0, 1, \ldots$, is chosen in such a manner that all components $x_l(s^l(k))$ in (2) themselves are updated at time $s^l(k)$, i.e.,

$$I_{s^i(k)-1} = i, \forall k \in \mathbb{N},\ i \in \{1, \ldots, m\} \text{ with } s^i(k) \ge 1, \tag{7}$$

where I_k is a shorthand notation for $I(k)$. Also assume all initial vectors are multiples of $\mathbf{1}$ (without loss of generality), that is

$$x(-k) := x_k(0)\mathbf{1},\ \forall k = 1, \ldots, m,$$

and renumber in this way the elements of the sequences of $s^l(k)$, $k = 0, 1, \ldots, l = 1, \ldots, m$, for which $s^l(k) = 0$.

Consider the asynchronous iteration $(\mathcal{F}, \mathcal{Y}_\mathcal{O}, \mathcal{I}, \mathcal{S})$

$$y(k+1) := F^{I_k}\left(y(\tilde{s}^1(k)), \ldots, y(\tilde{s}^{m_{I_k}}(k))\right),\ k = 0, 1, \ldots \tag{8}$$

where $\mathcal{F} = \{F^{I_k} | k = 0, 1, \ldots\}$ as in (2), $\mathcal{I} = I_k$, $k = 0, 1, \ldots$, $\mathcal{S} = \{\tilde{s}^i(k) | k = 0, 1, \ldots; i = 1, \ldots, m_{I_k}\}$ is given by

$$\tilde{s}^i(k) := s^{m_{I_k}(i)}(k),\ \forall k \in \mathbb{N}_0,\ i = 1, \ldots, m_{I_k}, \tag{9}$$

and $\mathcal{Y}_O$ by $y(-l) := x_1(-l), l = 1, \ldots, m$.

Using induction on k, the asynchronous iteration (8) generates

$$y(k+1) = x_{I_k}(k+1),\ \forall k \in \mathbb{N}_0. \tag{10}$$

$\square$

Remark 2: The asynchronous iteration (8)–(9) is equivalent to the asynchronous consensus formulation (2) in the sense of (10). However, their convergence speeds could be different if two or more agents update their states exactly at the same time t, therefore resulting in a series of events at different event-index k.

B. Verification of the Paracontracting Property

The second step in the convergence analysis of asynchronous consensus (2) is to verify the paracontracting property of the pool of operators F^i, $i \in \mathbb{I}$. To test such a property against an arbitrary pool of operators is often not straight-forward because it is non-trivial to choose an appropriate norm in Definition 2 with respect to which F^i's are paracontracting.

Here, we show that the widely used convex combination operator (e.g., in [21]) is strictly nonexpansive (therefore also paracontracting) with respect to the infinity norm.

Lemma 2 (Convex Combination Operators): Given an asynchronous iteration (8), the operator F^i is defined such that $y(k+1)$ is in the relative interior of the convex hull of $y(\tilde{s}^1(k)), \ldots, y(\tilde{s}^{m_{I_k}}(k))$, $i = I_k$, $k = 0, 1, \ldots$.

Then the pool of operators F^i are strictly nonexpansive with respect to infinity norm.

Proof: Since $y(k+1)$ is in the convex hull, it can be represented as a convex combination of $y(\tilde{s}^1(k)), \ldots, y(\tilde{s}^{m_i}(k))$, $i = I_k$, i.e.,

$y(k+1) = \sum_{j=1}^{m_i} f_{ij}(k) y(\tilde{s}^j(k))$ where f_{ij} are nonnegative numbers such that $\sum_{j=1}^{m_i} f_{ij}(k) = 1$.

Let Q denote the cardinality of $\mathcal{F}$, which is finite for a multiagent systems with a finite number of agents. Then the pool $\mathcal{F} = \{F^i | i = 1, \ldots, Q\}$ defined by $F^i : \mathbb{R}^{nm_i} \to \mathbb{R}^n$,

$$F^i(y^1, \ldots, y^{m_i}) := \sum_{j=1}^{m_i} f_{ij} y^j, \; i = 1, \ldots, Q \tag{11}$$

is strictly non-expansive on all closed sets $D \subset \mathbb{R}^n$. This follows from the fact, for all $i \in \{1, \ldots, Q\}$, $x, y \in \mathbb{R}^{nm_i}$,

$$\left\| F^i(x) - F^i(y) \right\| \leq \sum_{l=1}^{m_i} f_{il} \left\| x^l - y^l \right\| \leq \max_{1 \leq l \leq m_i} \left\| x^l - y^l \right\| \tag{12}$$

and equality holds if and only if $x^l - y^l = z = F^i(x-y)$, $\forall l = 1, \ldots, m_i$, for some $z \in \mathbb{R}^n$.

C. Verification of Confluence

Confluent conditions in Definition 4 describe coupling of an iteration process, needed to avoid divergence. Before specifying sufficient conditions for iteration graphs to be confluent, we shall first introduce the notion of "graph composition" used in [19].

Let $\mathcal{G}$ be the set of all directed graphs with vertex set $\mathcal{N} = \{1, 2, \ldots, m\}$. The *composition* $G_2 \circ G_1$ of graphs $G_1 \in \mathcal{G}$ and $G_2 \in \mathcal{G}$ is the directed graph with vertex set $\mathcal{N}$ and edge set defined in such a way so that (u, v) is an edge of the composition just in case there is a vertex w such that (u, w) is an edge of G_1 and (w, v) is an edge of G_2.

A vertex i is a *root* of a directed graph G if for each other vertex j of G, there is a directed path from i to j. By a *rooted graph* $G \in \mathcal{G}$ is meant a graph which possesses at least one root. We say that a finite of sequence of directed graphs $G_{p_1}, G_{p_2}, \ldots, G_{p_k}$ from $\mathcal{G}$ is *jointly rooted* if the composition $G_{p_k} \circ G_{p_{k-1}} \circ \ldots \circ G_{p_1}$ is a rooted graph. An infinite sequence of graphs $G_{p_1}, G_{p_2}, \ldots$, in $\mathcal{G}$ is *repeatedly jointly rooted* if there is a positive integer r for which each finite sequence $G_{p_{r(k-1)+1}}, \ldots, G_{p_{rk}}$, $k \geq 1$ is jointly rooted.

Lemma 3: For any trajectory of the system determined by (8) along which the sequence of interaction graphs $G(0), G(1), \ldots$ is repeatedly jointly rooted. In addition, we assume

Assumption 1:

(a) Agent i_0 always uses its own latest state to update its current state. That is, $s^{i_0}(k) = \max\{k_0 \leq k | I_{k_0 - 1} = i_0\}$ for all $k > \min\{k_0 \in \mathbb{N}_0 | I_{k_0} = i_0\}$ with $I_k = i_0$.

(b) $\mathcal{I} = I_k$, $k = 0, 1, \ldots$, is regulated.

(c) $k - s^l(k) \leq s$, $\forall k \in \mathbb{N}_0, l = 1, \ldots, m$, for an $s \in \mathbb{N}_0$.

Then, $(\mathcal{F}, \mathcal{Y}_{\mathcal{O}}, \mathcal{I}, \mathcal{S})$ of (8) is confluent.

Proof: Due to Assumption 1(a), in the iteration graphs of $(\mathcal{F}, \mathcal{Y}_{\mathcal{O}}, \mathcal{I}, \mathcal{S})$ all vertices $k \geq 1$ with $I_{k-1} = i_0$ are connected by a directed path. Since the sequence of interaction graphs is repeatedly jointly rooted, there is an $n_1 \in \mathbb{N}$, such that all vertices $k \geq n_1$ are successors of vertices a_k, for which $I_{a_k - 1} = i_0$. Due to Assumptions 1(b)&(c), we can see that $k - a_k$, $k = n_1, n_1 + 1, \ldots$ is bounded by some $a \in \mathbb{N}$. Hence, there is an $n_0 \in \mathbb{N}$, such that for all $k \geq n_0$

$$b_k := \max\left\{k' \in \mathbb{N} | (k' \leq k - a) \wedge I_{k'-1} = i_0\right\} \tag{13}$$

exists. Then, for all $k_0 \geq k$, k_0 is a successor of b_k and the sequence $k - b_k$, $k = n_0, n_0 + 1, \ldots$ is bounded by some $b \in \mathbb{N}$.

Following the same argument, by the repeatedly jointly rooted condition and the regularity of $\mathcal{I}, \mathcal{S}$, there is a $c \in \mathbb{N}$, independent of i, k so that for all $k \geq n_0$, $i \in \{1, \ldots, Q\}$, there is a path from b_k to b_{k+c} containing a vertex w_k^i with $I_{w_k^i} - 1 = i$. This concludes the proof. □

D. A Nonlinear Consensus Protocol

We are now ready to claim a new consensus result where F^i is allowed to be nonlinear.

Theorem 2: Consider the asynchronous consensus problem of the form

$$x_i(k+1) = F^{I(k)}\left(x_1(s^1(k), x_2(s^2(k)), \ldots, x_m(s^m(k))\right). \tag{14}$$

The pool of operators $\mathcal{F} = \left\{F^i : \mathbb{R}^n \to \mathbb{R} | i \in \{1, \ldots, Q\}\right\}$ is of the convex combination type and has a common fixed point at $\mathbf{1} \in \mathbb{R}^n$.

Then, under Assumptions 1(a), (b), (c) (in Lemma 3) and the repeatedly jointly rooted condition, the nonlinear protocol (14) or, equivalently, (8) guarantees asymptotic consensus.

Proof: Follows from Theorem 1 and Lemmas 1, 2 and 3. □

Remark 3: From Fig. 1, we can intuitively understand why the existence of of agent i_0 is necessary. Suppose that no agents use their past values during the process (thus no dashed edges). After removing the dashed edges from Fig. 1(b), the iteration graph is no longer confluent since there is no directed path from an odd-numbered vertex to an even-numbered vertex, and vice versa. This shows the necessity of existence of i_0 in Theorem 2.

The convex combination type of operators can be relaxed to the less restrictive type of paracontracting as stated in Theorem 1. Therefore, Theorem 2 can be used to study multiagent systems with nonlinear couplings, applicable to a wider range of applications than those of its linear counterparts in [4], [19]. Potential applications include distributed time synchronization, rendezvous and formation tracking of multirobot systems.

V. An Application: Robot Rendezvous via Center-of-Gravity Algorithms

One basic coordination task in multirobot systems is *rendezvous* [25], [26]. The *rendezvous* task requires the robots to converge to a single point. A common approach to this task relies on the robots calculating the *center of gravity* of the group and moving towards it. In this section, we prove the convergence of asynchronous center-of-gravity algorithm in [27] by using Theorem 2.

A. The Sudden-Stop Model

Each robot i in a group of m robots operates individually, repeatedly going through simple cycles consisting of three steps:

1 *Look*: Identify the locations of all robots and obtain (instantaneously) a multiset of points $P = \{pos_1, \ldots, pos_m\}$ defining the current configuration. The robots are indistinguishable, so i knows its own location pos_i but does not know the identity of the robots at each of the other points. When two or more robots reside at the same point, all robots will detect this fact.

2 *Compute*: Execute the algorithm Go_to_COG, resulting in a goal point pos_G.

3 *Move*: Move on a straight line towards the point pos_G. The robot might stop before reaching its goal point pos_G but is guaranteed to traverse a distance of at least L (unless it has reached the goal). The value of L is not assumed to be known to the robots, and they cannot use it in their calculations.

This model, which allows the robot to suddenly stop short of reaching its goal point, is henceforth referred to as the *sudden-stop* model [27].

Algorithm Go_to_COG and Its Convergent Property

Denote by $\bar{r}_i(t)$ the location of robot i at time t.

Algorithm Go_to_COG

1 Calculate the center of gravity, $\bar{c}_i(t) = (1/m)\sum_j \bar{r}_j(t)$.
2 Move to the point $\bar{c}_i(t)$.

As mentioned earlier, the move may terminate before the robot i actually reaches the point $\bar{c}_i(t)$. The point at which the robot does stop its movement is thus referred to as its *destination point*. More formally, define the destination point $\bar{\gamma}_i(t)$ of robot i to be the final point of the movement made by i following the last Look performed by i before or at time t.

In this subsection, we prove that the Algorithm Go_to_COG guarantees the convergence of m robots for any $m \geq 2$ in the case of the asynchronous model. In order to use Theorem 2, we need the following fact from [27].

Lemma 4: If for some time t_0, $\bar{r}(t_0)$ and $\bar{\gamma}(t_0)$ for all i reside in the interior of a closed convex curve, $\mathcal{C}$, then for every time $t > t_0$, $\bar{r}(t)$ and $\bar{\gamma}(t)$ also reside in the interior of $\mathcal{C}$ for every $1 \leq i \leq m$.

Theorem 3: In the full (sudden-stop) asynchronous model for any $m \geq 2$, in d dimensional Euclidean space, m robots performing Algorithm Go_to_COG will converge provided that each robot is activated infinitely often in an infinite execution (the "fairness" assumption).

Proof: By introducing the event time index k, we can regard $\bar{r}$ as a delayed version of $\bar{\gamma}$. Then the updating equation for $\bar{\gamma}$ has a form of

$$\bar{\gamma}_i(k+1) = F^{I(k)}(\bar{\gamma}_1(s^1(k)), \ldots, \bar{\gamma}_m(s^m(k))) \tag{15}$$

where the operator $F^{I(k)}$ is of convex combination type due to Lemma 4.

Due to the physical nature of updating process, a simple cycle of look, compute, and move takes a finite amount of time bounded from below. This observation plus the fairness assumption lead to Assumption 1(a), (b), (c) (in Lemma 3) fulfilled.

From Algorithm Go_to_COG, we can see that every robot always uses all $\bar{\gamma}_i$'s ($1 \leq i \leq m$) in the updating equation. The confluence condition is henceforth satisfied from the fact that the composition of interaction graphs is a complete graph across a finite time interval.

Theorem 3 follows from Theorem 2. □

VI. Conclusion

In this note, a novel asynchronous consensus result was introduced and shown using nonlinear paracontracitons and confluence. This result is more general than existing ones and provides a powerful tool to study a wider range of applications. In particular, we applied our results to the robot rendezvous problem. For future research, it will be interesting to get an estimate of the convergence speed in the nonlinear case; cf. [19], [28].

References

[1] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Automat. Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003.

[2] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Automat. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[3] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Automat. Control*, vol. 50, no. 5, pp. 655–661, May 2005.

[4] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. 44th IEEE CDC and ECC'05*, Seville, Spain, 2005, pp. 2996–3000.

[5] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multiagent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[6] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, 2004.

[7] R. Olfati-Saber, "Ultrafast consensus in small-world networks," in *Proc. ACC'05*, Portland, OR, Jun. 2005, vol. 4, pp. 2371–2378.

[8] Z. Jin and R. Murray, "Multihop relay protocols for fast consensus seeking," in *Proc. 45th IEEE CDC*, San Diego, CA, Dec. 2006, pp. 1001–1006.

[9] H. G. Tanner and D. K. Christodoulakis, "State synchronization in local interaction networks is robust with respect to time delays," in *Proc. 44th IEEE CDC and ECC'05*, Seville, Spain, 2005, pp. 4945–4950.

[10] W. Wang and J.-J. Slotine, "Contraction analysis of time-delayed communications and group cooperation," *IEEE Trans. Automat. Control*, vol. 51, no. 4, pp. 712–717, Apr. 2006.

[11] D. Angeli and P.-A. Bliman, "Extension of a result by Moreau on stability of leaderless multiagent systems," in *Proc. 44th IEEE CDC and ECC'05*, Seville, Spain, 2005, pp. 759–764.

[12] F. Xiao and L. Wang, "State consensus for multiagent systems with switching topologies and time-varying delays," *Int. J. Control*, vol. 79, no. 10, pp. 1277–1284, 2006.

[13] Y. Liu and K. M. Passino, "Cohesive behaviors of multiagent systems with information flow constraints," *IEEE Trans. Automat. Control*, vol. 51, no. 11, pp. 1734–1748, Nov. 2006.

[14] Y. Hatano, A. K. Das, and M. Mesbahi, "Agreement in presense of noise: Pseudogradients on random geometric networks," in *Proc. 44th IEEE CDC and ECC'05*, Seville, Spain, 2005, pp. 6382–6387.

[15] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Automat. Control*, vol. AC-31, no. 9, pp. 803–812, Sep. 1986.

[16] L. Fang, P. J. Antsaklis, and A. Tzimas, "Asynchronous consensus protocols: Preliminary results, simulations and open questions," in *Proc. 44th IEEE CDC and ECC'05*, Seville, Spain, 2005, pp. 2194–2199.

[17] L. Fang and P. Antsaklis, "On communication requirements for multi-agent consensus seeking," in *Networked Embedded Sensing and Control*. New York: Springer, 2006, vol. 331, pp. 53–67.

[18] M. Cao, A. S. Morse, and B. D. O. Anderson, "Coordination of an asynchronous multiagent system via averaging," in *Proc. 16th IFAC World Congress*, Jul. 4–8, 2005 [Online]. Available: http://ww.eng.yale.edu/controls/2005/IFAC_Final.pdf

[19] M. Cao, A. S. Morse, and B. D. O. Anderson, "Reaching a consensus in a dynamically changing environment – Convergence rates, measurement delays and asynchronous events," *SIAM J. Control Optim.*, vol. 47, no. 2, pp. 601–623, Feb. 2008.

[20] M. Pott, "On the convergence of asynchronous iteration methods for nonlinear paracontractions and consistent linear systems," *Linear Algebra Applicat.*, vol. 283, pp. 1–33, 1998.

[21] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Trans. Automat. Control*, vol. 50, no. 2, pp. 169–182, Feb. 2005.

[22] C. W. Wu, "Synchronization and convergence of linear dynamics in random directed networks," *IEEE Trans. Automat. Control*, vol. 51, no. 7, pp. 1207–1210, Jul. 2006.

[23] L. Elsner, I. Koltracht, and M. Neumann, "Convergence of sequential and asynchronous nonlinear paracontractions," *Numer. Math.*, vol. 62, pp. 305–319, 1992.

[24] A. Benveniste, E. Fabre, S. Haar, and C. Jard, "Diagnosis of asynchronous discrete-event systems: A net unfolding approach," *IEEE Trans. Automat. Control*, vol. 48, no. 5, pp. 714–727, May 2003.

[25] J. Lin, A. S. Morse, and B. D. O. Anderson, "The multiagent rendezvous problem – The asynchronous case," in *Proc. 43rd IEEE CDC*, Atlantis, Paradise Island, Bahamas, 2004, pp. 1926–1931.

[26] J. Cortes, S. Martinez, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions," *IEEE Trans. Automat. Control*, vol. 51, no. 8, pp. 1289–1298, Aug. 2006.

[27] R. Cohen and D. Peleg, "Convergence properties of the gravitational algorithm in asynchronous robot systems," *SIAM J. Comput.*, vol. 34, no. 6, pp. 1516–1528, 2005.

[28] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *Math. Optim. Control* pp. 1–23, Dec. 2006 [Online]. Available: http://web.mit.edu/jnt/www/Papers/P-06-flock-rate-sub.pdf