Robust Stability of Spreading Blocks in Aggregate Programming

Yuanqiu Mo, Soura Dasgupta and Jacob Beal

***Abstract*—Self-stabilizing (asymptotically stable) information spreading algorithms are an important building block of many distributed systems featuring in Aggregate Computing, but the dynamics of their convergence has largely remained elusive, except in a special case of a distance finding variant known as the Adaptive Bellma-Ford (ABF) Algorithm. As a how they behave in interconnections involving other building blocks, it is important to develop a framework to demonstrate their robust stability. This paper addresses this shortcoming by analyzing a very general block of which ABF is a special case. It provides a proof of global uniform asymptotic stability and ultimate bounds on the state error in face of persistent perturbations.**

I. Introduction

Recent years have witnessed the emergence of complex networked distributed systems involving compositions of numerous physical and logical systems that may themselves be distributed. Understanding their dynamics, stability and reliability are of paramount importance. Accordingly the controls literature has witnessed significant research on the stability of networked control systems, e.g. [2] - [6].

This paper on its part is concerned with a complementary set of new challenges posed by the analysis and design of systems that enable the dispersion of services to local devices as in smart cities, tactical information sharing, personal and home area networks, and the Internet of Things (IoT) [9]. Realizing the potential of these domains requires devices to *safely and seamlessly* collaborate with neighboring devices through low latency peer to peer communications, often in feedback loops, with individual blocks independently subjected to perturbations due to mobility, uncertainty and noise. These systems are open, i.e. are expected to support an unbounded and rapidly evolving collection of distributed services. Their analysis requires a framework for analyzing the *composition* of distributed services, to guide service engineering and support run time monitoring and management of complex compositions of dispersed services.

The emerging field of *aggregate computing* is conceived to address the ensuing challenges. Aggregate computing views the basic computing unit as a physical region comprising a groups of interacting computing devices, rather than an individual physical device [9]. Much like the OSI model for communication networks, [9] abstracts the system into different layers by factoring distributed system design into separate tasks of device-level communication and discovery, coherence between collective and local operations, resilience, and programmability. In particular, [7] and [8], show that a broad class of dispersed services are captured by the interaction between three types of *building block* distributed algorithms: (i) G-blocks that spread information through a network of devices, (ii) C-blocks that summarize salient information about the network to be used by interacting units, and (iii) T-blocks that maintain temporary state. While the dynamics of these basis set systems appear amenable to effective composition [10]–[12], the formal analysis of each of these blocks has been limited to demonstrating self-stabilization, which lacks the explicit notion of robustness inherent to such concepts in the stability theory literature as uniform asymptotic stability, [1].

As a first step towards understanding the stability of possibly feedback interconnection of aggregate computing blocks one must study them individually to see whether their behavior is robust to perturbations. At the minimum this in turn requires their uniform asymptotic stability without perturbations. This is so as the uniform asymptotic stability of unperturbed systems permits them to be well behaved under modest perturbations, [15]- [18]. Beyond that, finding ultimate bounds in the face of perturbations, would permit the use of sophisticated refinements, [22], of the small gain theorem, [1], to help establish closed loop stability.

Indeed in [24] and [25] we have conducted such an analysis of a very special G-block: The Adaptive Bellman-Ford (ABF) algorithm, which is an adaptive version of the classical Bellman-Ford algorithm, [19], [20]. The ABF adaptively finds the distance of nodes in a network from a set of sources. In this paper we look at a much more general variant of the G-block, [23]. This general G-block accommodates non-Euclidean distance metrics, as for example those that discourage certain routes, permits implementing *broadcast* where each source broadcasts a stored value to nodes nearest to it, and other features, some of which are described in Section II.

After introducing the algorithm in Section II we provide a stability theoretic proof of its global uniform asymptotic convergence in Section III. This proof is much more nontrivial as compared to the analysis for ABF, in [24] and [25]. For starters ABF assumes that all source states, representing the distance from the set of sources, are held at zero. There is a comparable set of nodes in the general algorithm, but the states of these nodes may not converge to these values. In fact unlike ABF one cannot even *a priori* assume the existence of stationary points: Rather it emerges as a byproduct of a proof. While for ABF we were able to find a simple non-

Mo and Dasgupta are with the Dept. of ECE, University of Iowa, Iowa City, IA 52242, USA. Dasgupta has a visiting position in the Shandong Computer Science Center (National Supercomputer Center in Jinan), Shandong Provincial Key Laboratory of Computer Networks. [yuanqiu-mo,soura-dasgupta]@uiowa.edu. Beal is with the Raytheon BBN Technologies Cambridge, MA, USA 02138. jakebeal@ieee.org. This work has been supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001117C0049. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. This document does not contain technology or technical data controlled under either U.S. International Traffic in Arms Regulation or U.S. Export Administration Regulations.

increasing Lyapunov function, none such is readily available for this analysis. There is a simple Lyapunov function that is nonincreasing but *only after a certain time.* Despite this fact, with an additional Lipschitz condition on the update kernel we are able to establish ultimate bounds on the state error under persistent perturbations, in Section IV. Section V gives several simulations. Section VI concludes.

II. A GENERAL SPREADING BLOCK

We consider an undirected graph, $\mathcal{G}$ with node set $V = \{1, 2, \cdots, N\}$. Nodes i and k are *neighbors* if they share an edge. The set of neighbors of i, will be denoted by $\mathcal{N}(i)$. In this section we present one of the two versions of the general spreading or G-block enunciated in [23]. As the formulation in [23] uses language from field calculus which most control theorists are unfamiliar with, we translate the formulation in [23] to notation and framework that are more standard for the control setup.

Suppose the desired state at the i-th node is x_i. Then in the t-th iteration its *estimate* $\hat{x}_i(t)$ obeys

$$\hat{x}_i(t+1) = \min\left\{\min_{k\in\mathcal{N}(i)}\{f(\hat{x}_k(t), e_{ik})\}, s_i\right\}, \forall t \geq t_0. \quad (1)$$

The e_{ik} define the structural aspects of $\mathcal{G}$; e.g. they may be the edge lengths between neighbors; s_i is the *maximum value that* $\hat{x}_i(t)$ *can acquire*. For some they are infinity. Define

$$\mathcal{S}^* = \{i \in V | s_i < \infty\}. \quad (2)$$

The following assumption holds.

Assumption 1. *The underlying graph is connected. All* $s_i \geq s_{\min} \geq 0$, $e_{ik} = e_{ki} \geq \epsilon > 0$, $\mathcal{S}^*$ *defined in (2) is nonempty and* $\mathcal{S}^* \neq V$. *Further* $f(a, b)$ *obeys for some* $\delta > 0$,

$$f(a, b) > a + \delta, \quad (3)$$

$$f(a_1, b) \geq f(a_2, b), \text{ if } a_1 \geq a_2. \quad (4)$$

and is finite for all finite a, b.

Note (3) makes $f(a, b)$ *progressive* in a, while (4), makes it *monotonic* in a. The definition of progressive is stronger than that in [23], which only requires that $f(a, b) > a$. This is so as [23] is only concerned with self-stabilization, which unlike asymptotic stability does not have the notion of robustness attached to it. By contrast we are concerned with robust behavior and thus require this strengthened condition. While no such conditions are imposed on with respect to b for the convergence analysis in Section III they are needed in Section IV, where ultimate bounds under parturbations are provided. The stationary point $x = [x_1, \cdots, x_N]^T$ of (1) obeys:

$$x_i = \min\left\{\min_{k\in\mathcal{N}(i)}\{f(x_k, e_{ik})\}, s_i\right\}, \ \forall\ i \in V. \quad (5)$$

The existence of this stationary point will emerge as a byproduct of our convergence proof. We make a few definitions one of which assumes the existence of the stationary points.

Definition 1. *If in (5),* $x_i = s_i$ *then we say that* i *is its own true constraining node. Otherwise, the minimizing* k *in (5) is a true constraining node of* i. *As* i *may have more than one true constraining node, its set of true constraining nodes is designated as* $\mathcal{C}(i)$. *Similarly, if in (1),* $\hat{x}_i(t) = s_i$, *then we say that* i *is its constraining node at time* t. *Otherwise, the minimizing* k *in (1) is a constraining node of* i *at time* t. *As* i *may have more than one constraining node, its set of true constraining nodes at time* t *is designated as* $\hat{\mathcal{C}}(i, t)$.

We now provide some examples of (1). The first and simplest is the *Adaptive Bellman-Ford* (ABF) algothm a variant of the classical Bellman-Ford algorithm, [19], [20], for finding the *Euclidean* distance of nodes from a set of source nodes S. In particular with $\hat{d}_k(t)$ the estimated distance of the k-th node from a set of sources S, ABF proceeds as

$$\hat{d}_i(t+1) = \begin{cases} \min_{j\in\mathcal{N}(i)}\left\{\hat{d}_j(t) + e_{ij}\right\} & i \notin S \\ 0 & i \in S \end{cases}, \quad (6)$$

where $e_{ij} > \epsilon > 0$ is the edge length between nodes i and j. Observe (6) is a special case of (1): $f(a, b) = a + b > a$, $\hat{d}_i = \hat{x}_i,$, $s_i = 0$ if $i \in S$ and $s_i = \infty$ for all $i \notin S$. Unlike the classical Bellman-Ford algorithm, ABF does not require that $\hat{d}_i(t_0)$ at the initial time be no smaller than its true value d_i. It also accommodates multiple sources that the classical algorithm does not. We oberve that we have analyzed ABF in a Lyapunov framework in [24] and [25]. The latter reference also provides ultimate error bounds under perturbations. The generalizations provided here are nontrivially different as unlike ABF, in (1), f is permitted to be nonlinear.

Observe (1) generalizes (6) in several ways. First, it permits non-Euclidean distance measures. For example, the length of a path might be quantified not by distance but by time to traverse, modulated by traffic, speed limits, intersections, etc. Other measures move further away from typical notions of distance: for example, metrics for the cumulative exposure to a space-filling hazard such as radiation or hostile action.

Second, in ABF $s_i = 0$, $\forall\ i \in \mathcal{S}^*$. In contrast, in (1) the finite s_i may be nonzero, and in the stationary state values of these nodes need not equal s_i and may in fact be smaller. For example, the set of finite s_i might be exfiltration points from a tactical network, with values equal to the time needed to transmit information back to a supporting cloud environment. In this scenario, a fast link through an airborne asset would have a low s_i value, while a slow link through a satellite would have a high s_i value, causing exfiltration to route through the airborne asset when it is nearby, but shift to the backup satellite link when that high-speed link is not available.

Finally, (1) accommodates computations besides distance-like fields, such as *broadcast*, where each device takes on the most recent value held by the source nearest to it. This is accomplished by using a pair rather than a single value for the state x_i, where the first element of the pair a distance metric as before, but where the second element is an arbitrary state. Using a lexicographic ordering over pairs, each device then takes its second pair value as an arbitrary computation on the second pair value held by its current constraining node. Broadcast is the simplest such function, which uses the identity function for the second pair value, that is carried outward from each source along shortest paths.

III. Convergence without perturbations

We now analyze the convergence of (1) without perturbations, i.e. when the e_{ik}, the edge set, and s_i do not change, and when Assumption 1 holds. A key difference between this analysis and that for ABF in [24] and [25] is that in these references, the linearity of $f(\cdot,\cdot)$ permits a formulation of a Lyapunov function. The very general nature of $f(\cdot,\cdot)$ here makes finding this Lyapunov function harder. This is particularly exacerbated by the fact that unlike ABF one cannot even *a priori* assume the existence of stationary points let alone their uniqueness. This is so even when like ABF $f(a,b) = a+b$, but members of $\mathcal{S}^*$ defined in (2) may have nonzero s_i values. Instead we first prove convergence exploiting purely the structure of (1), thus implicitly proving the existence of a stationary point.

Define two time varying sets. The first at time t comprises nodes in $\mathcal{S}^*$ that acquire their maximum values s_i, i.e.

$$\mathcal{S}(t) = \{i \in \mathcal{S}^* | \hat{x}_i(t) = s_i\}. \tag{7}$$

The second set $\mathcal{U}(t)$ requires a recursive definition below.

Definition 2.

$$\mathcal{U}(t_0 - 1) = \phi. \tag{8}$$

Further, $\mathcal{U}(t+1) = S(t+1) \bigcup R(t+1)$ where $R(t+1)$ comprises nodes constrained at t by a member of $\mathcal{U}(t)$.

For our first lemma we define a function that we will prove is strictly increasing while the set $\mathcal{U}(t) \neq V$:

$$\hat{x}_{\min}(t) = \min_{j \notin \mathcal{U}(t)} \{\hat{x}_j(t)\}. \tag{9}$$

Lemma 1. *Consider (1), with $\mathcal{U}(t)$ and $\hat{x}_{\min}(t)$ in Definition 2 and (9), respectively, and Assumption 1 in force. Then for all $\hat{x}(t_0) \geq 0$ the following holds while the set $\mathcal{U}(t) \neq V$:*

$$\hat{x}_{\min}(t+1) > \hat{x}_{\min}(t). \tag{10}$$

Proof. Suppose for some $k \notin \mathcal{U}(t+1)$, $\hat{x}_{\min}(t+1) = \hat{x}_k(t+1)$. Consider any $j \in \hat{\mathcal{C}}(k,t)$. To establish a contradiction, suppose $j = k$. Then from Definition 1, $\hat{x}_k(t+1) = s_k$ i.e. from (7), $k \in \mathcal{S}(t+1) \subset \mathcal{U}(t+1)$ leading to a contradiction. Thus $j \neq k$. As $k \notin \mathcal{U}(t+1)$, by Defintion 2, $j \notin \mathcal{U}(t)$.

Then

$$\begin{aligned} \hat{x}_{\min}(t+1) &= \hat{x}_k(t+1) \\ &= f(\hat{x}_j(t), e_{kj}) \\ &> \hat{x}_j(t) + \delta \\ &> \hat{x}_{\min}(t), \end{aligned}$$

where the last inequality follows from (9) and the fact that $j \notin \mathcal{U}(t)$. □

The next lemma proves that at least one member of $\mathcal{S}^*$ must converge to its maximum value. To this end define

$$s_{\min} = \min_{j \in \mathcal{S}^*} \{s_j\}. \tag{11}$$

Lemma 2. *Under the conditions of Lemma 1, suppose for some $i \in \mathcal{S}^*$, $s_{\min}$ in (11) equals s_i. Define*

$$\bar{x}(t) = \min_{j \in V} \{\hat{x}_j(t)\}. \tag{12}$$

Then either

$$\bar{x}(t) = \hat{x}_i(t) = s_i \tag{13}$$

or

$$\bar{x}(t+1) \geq \bar{x}(t) + \delta. \tag{14}$$

Further there exist a $T \geq t_0$, such that for all $t \geq T$, $\hat{x}_i(t) = s_i$.

Proof. Proving that either (13) or (14) holds will prove the lemma. Suppose (14) is violated. Then from the progressive property in (3), for some $j \in \mathcal{S}^*$

$$\bar{x}(t+1) = s_j = \hat{x}_j(t+1) = s_j = \bar{x}(t) \geq s_i \geq \hat{x}_i(t).$$

Thus unless (13) holds (14) must hold. As s_i is the maximum value of $\hat{x}_i(t)$, the result follows. □

As will be evident in the sequel this will be important in the characterization of stationary points. Observe also that this lemma states that *at least one* member of $\mathcal{S}^*$ converges to its *maximum value* in a finite time, and does not preclude others from doing so. Further this also proves that all states are bounded.

Lemma 3. *Under the conditions of Lemma 1 there exists an M, dependent on the initial states $\hat{x}_i(t_0) \geq 0$ such that for all $t \geq t_0$, and all $i \in V$, $|\hat{x}_i(t)| < M$.*

Proof. From Lemma 2 there exist T and $i \in \mathcal{S}^*$ such that for all $t \geq T$, $\hat{x}_i(t) = s_i$. Since $f(\cdot,\cdot)$ is finite for finite arguments, (1) cannot have finite escape time i.e. all states are finite at $t = T$. As the graph is connected there is a path from i to every node in the graph. Consider any $k \in \mathcal{N}(i)$. Then from (1) for all $t \geq T$

$$\hat{x}_k(t+1) \leq f(s_i, e_{ik}). \tag{15}$$

Continuing in this vein all nodes are bounded. □

In view of Lemma 1 this immediately yields the following result:

Lemma 4. *Under the conditions of Lemma 1 there is a T_1 such that for all $t \geq T_1$, $U(t) = V$.*

We will now prove that for all $t > T_1$ defined in (4), all states are nonincreasing. This in particular proves that should x_i that satisfy (5) exist then

$$\hat{x}_i(t) \geq x_i, \ \forall i \in V, \ t \geq T_1. \tag{16}$$

Theorem 1. *With T_1 as in Lemma 4, the following holds.*

$$\hat{x}_i(t+1) \leq \hat{x}_i(t), \ \forall i \in V, \ t \geq T_1.$$

Proof. There exists an $n \geq 0$ such that $U(T_1 - n - 1) = \phi$, but $U(t)$ is nonempty for all $t \in \{T_1 - n, T_1 - n + 1, \cdots, T_1\}$. From Definition 2,

$$U(T_1 - n) = S(T_1 - n) \neq \phi. \tag{17}$$

Now consider any $i_0 \in U(T_1)$. Then by Definition 2, there is a sequence of nodes $i_k \in \mathcal{U}(T_1 - k)$, $k \in \{1, \cdots, n\}$ such that $i_k \in \hat{\mathcal{C}}(i_{k-1}, T_1 - k)$. We now assert that for all $k \in \{1, \cdots, n\}$

$$\hat{x}_{i_k}(t) \leq \hat{x}_{i_k}(T_1 - k), \ \forall t \geq T_1 - k. \tag{18}$$

We will prove (18) by induction on k. From (17), $i_n \in S(T_1 - n)$. By definition of $S(t)$ in (7), $\hat{x}_{i_n}(T_1 - n)$ has its maximum value s_{i_n}. Thus the result holds for $k = n$. Now suppose it holds for all $l \in \{k, \cdots, n\}$, $k > 0$. As $i_k \in \hat{\mathcal{C}}(i_{k-1}, T_1 - k)$, it follows that $i_k \in \mathcal{N}(i_{k-1})$. Then from (3, (4) and (1), for any $t \geq T_1 - k + 1$, there holds

$$\begin{aligned}
\hat{x}_{i_{k-1}}(t) &= \min\left\{\min_{j \in \mathcal{N}(i_{k-1})}\left\{f\left(x_j(t-1), e_{i_{k-1},j}\right)\right\}, s_{i_{k-1}}\right\} \\
&\leq \min\left\{f\left(x_{i_k}(t-1), e_{i_{k-1},i_k}\right), s_{i_{k-1}}\right\} \\
&\leq \min\left\{f\left(x_{i_k}(T_1-k), e_{i_{k-1},i_k}\right), s_{i_{k-1}}\right\} \\
&= \hat{x}_{i_k}(T_1 - k + 1)
\end{aligned}$$

where the last step uses the fact that $i_k \in \hat{\mathcal{C}}(i_{k-1}, T_1 - k)$. Thus for every $i_0 \in \mathcal{U}(t)$, $t \geq T_1$, $\hat{x}_{i_0}(t+1) \leq \hat{x}_{i_0}(t)$. As $\mathcal{U}(t) = V$ for all $t \geq T_1$, the result follows. □

We now have the proof of convergence to a stationary point.

Theorem 2. *Under the conditions of Theorem 1, there exist x_i such that for all $i \in V$*

$$\lim_{t \to \infty} \hat{x}_i(t) = x_i.$$

Further the convergence is uniform in t_0.

Proof. From Lemma 2, for all $t \geq T$, defined in the lemma, and all $j \in V$, $\hat{x}_j(t) \geq s_{\min}$. From Theorem 1 all $\hat{x}_j(t)$ are nonincreasing for all $t \geq T_1$. From Lemma 3 all states are bounded. Thus for for all $t \geq T_1$ each $\hat{x}_j(t)$ is a nonincreasing sequence on a compact set, and hence must converge. □

We now turn to proving the uniqueness of this stationary point. To this end we make a definition.

Definition 3. *In a stationary point satisfying (5) the source set $\mathcal{S}_\infty = \{i | x_i = s_i\}$.*

We now argue that $\mathcal{S}_\infty$ is unique through a process of elimination. Without loss of generality assume that $s_1 = s_{\min}$ defined in Lemma 2. From Lemma 2

$$i_1 \in \mathcal{S}_\infty. \tag{19}$$

Consider now two nodes $i \in \mathcal{S}^*$ and $j \in \mathcal{S}^*$. As the graph is connected there is at least one path between $i = k_0 \to k_1 \to k_2 \to \cdots, \to k_l = j$. Define the sequence $c_{k_0} = s_i$ and

$$c_{k_{i+1}} = \min\left\{f(c_{k_i}, e_{k_i,k_{i+1}}), s_{k_{i+1}}\right\}. \tag{20}$$

We will call this the shortest path from i to j if c_j is the minimum over all such paths, and the minimum c_j the distance between i and j. From (19) any j cannot be a source if the distance between 1 and j is smaller than s_j. Eliminate all such j. In this candidate source set obtained after the removal of these nodes, there is any node whose distance from another is smaller than its maximum value, then all such nodes must also be removed from the source set. This way we can arrive at a final source set which because of (19) is not empty. Further one can show that it is unique and determined entirely by $f(\cdot,\cdot)$, e_{ik} and the s_i. One can also then show that the the remaining x_i values are unique.

We conclude this section by upper bounding the time to converge. To this end we need a lemma.

Lemma 5. *Consider true constraining nodes defined in 1 and a sequence of nodes $k_1, \cdots, k_l$ such that k_1 is a source, k_i is the true constraining node of k_{i+1}, and $k_i \neq k_{i+1}$. Then there exists a number $\mathcal{D}(\mathcal{G})$, called the effective dimameter of the graph $\mathcal{G}$ such that $l \leq \mathcal{D}(\mathcal{G})$.*

Proof. Since $k_i \neq k_{i+1}$, k_i is not a source for $i > 1$. Further the progressive property ensures that $x_{k_i} < x_{k_{i+1}}$. Thus, this sequence has no cycles. Thus as the number of nodes in the graph is finite, the result follows. □

A byproduct of Theorem 1 is that there exists a $T_2 \leq T_1$ such that

$$\hat{x}_i(T_2) \geq x_i, \ \forall i \in V. \tag{21}$$

We first prove the following Lemma.

Lemma 6. *Under the conditions of Theorem 1 suppose there exists a T_2 such that (21) holds. Then*

$$\hat{x}_i(t) \geq x_i, \ \forall i \in V, \ t \geq T_2. \tag{22}$$

Proof. Use induction. Observe that (22) holds for $t = T_2$. Suppose it holds for some $t \geq T_2$. Suppose $l \in \hat{\mathcal{C}}(i,t)$ and $k \in \mathcal{C}(i)$. Then from (1), (5) and (4), the result follows as:

$$\begin{aligned}
\hat{x}_i(t+1) &= \min\left\{f(\hat{x}_l(t), e_{il}), s_i\right\} \\
&\geq \min\left\{f(x_l(t), e_{il}), s_i\right\} \\
&\geq \min\left\{f(x_k(t), e_{ik}), s_i\right\} \\
&= x_k.
\end{aligned}$$

□

We now provide an upper bound on the time to converge in terms of T_2.

Theorem 3. *Under the conditions of Lemma 6 with T_2 defined in that lemma, (1) converges in at most $T_2 + \mathcal{D}(\mathcal{G})$ steps.*

Proof. Because of Theorem 1, (16) holds. Now consider a sequence of true constraining nodes $k_1, \cdots, k_l$ defined in Lemma 5. Then the result will follow from Lemma 5 if we show that for all $i \in \{1, \cdots, l\}$

$$\hat{x}_{k_i}(T_2 + i) = x_{k_i}. \tag{23}$$

We now show this by induction. As k_1 is a source and is thus its constraining node, and (16) holds, from (1) we obtain:

$$\begin{aligned}
x_{k_1} &= s_{k_1} \\
&\leq \hat{x}_{k_1}(T_2 + 1) \\
&\leq s_{k_1} \\
&= x_{k_1}.
\end{aligned}$$

Thus indeed (23) holds for $i = 1$. Now suppose it holds for all $i \in \{1, \cdots, m-1\}$. Then as $k_{m-1} \in \mathcal{C}(k_m)$ and is thus a neighbor of k_m, we have that

$$\begin{aligned}
x_{k_m} &\leq \hat{x}_{k_m}(T_2 + m) \\
&\leq f(\hat{x}_{k_{m-1}}(T_2 + m - 1), e_{k_m,k_{m-1}}) \\
&= f(x_{k_{m-1}}(T_2 + m - 1), e_{k_m,k_{m-1}}) \\
&= x_{k_m}.
\end{aligned}$$

This proves the result.

□

What about a bound on T_2 the first time at which the smallest unconverged state exceeds or equals

$$x_{\max} = \max_{i \in V}\{x_i\}. \tag{24}$$

Arguing similarly to the proof Lemma 2, the smallest unconverged if it increases does so by at least δ. Thus with $\bar{x}(t_0)$ defined in the proof of Lemma 2.

$$T_2 \leq \left\lceil \frac{x_{\max} - \hat{x}_{\min}(t_0)}{\delta} \right\rceil . \tag{25}$$

Thus the time to converge is no greater than

$$\mathcal{D}(G) + \left\lceil \frac{x_{\max} - \hat{x}_{\min}(t_0)}{\delta} \right\rceil .$$

IV. Ultimate Bounds

In this section, we present the robustness of the general G-block under perturbations. In [25], robustness of ABF is demonstrated by showing ultimate boundness of distance estimates around nominal distance values while edge length change from its nominal value. For general G-block, we again present its robustness by showing its ultimate robustness under perturbations.

First, we introduce the following assumption.

Assumption 2. *Now suppose $f(\cdot,\cdot)$ is monotonic with respect to its second argument, $f(a,b)$ obeys*

$$f(a,b_1) \geq f(a,b_2), \text{ if } b_1 \geq b_2. \tag{26}$$

Further, $f(\cdot,\cdot)$ is Lipschitz continuous on its second argument, such that

$$|f(\hat{x}_k(t), e_{ik}(t)) - f(\hat{x}_k(t), e_{ik})| \leq L_1 |e_{ik}(t) - e_{ik}| \tag{27}$$

with $L_1 > 0$. Meanwhile, we also assume $f(\cdot,\cdot)$ is Lipschitz continuous on its first argument, such that

$$|f(a_1,b) - f(a_2,b)| \leq L_2 |a_1 - a_2| \tag{28}$$

where $L_2 > 0$.

Now we consider the case e_{ij} in (1) is not a constant but a variable $e_{ij}(t)$ satisfying

$$e_{ij}(t) = e_{ij} + \epsilon_{ij}(t) \tag{29}$$

with $|\epsilon_{ij}(t)| \leq \epsilon$. Then (27) can be further interpreted as

$$|f(\hat{x}_k(t), e_{ik}(t)) - f(\hat{x}_k(t), e_{ik})| \leq L_1 \epsilon \tag{30}$$

Now $\forall i \in V$ and $\forall k \in \mathcal{N}(i)$, (1) can be interpreted as:

$$\hat{x}_i(t+1) = \min\left\{ \min_{k \in \mathcal{N}(i)} \left\{ f\left(\hat{x}_k(t), e_{ik}(t)\right) \right\}, s_i \right\}. \tag{31}$$

The next lemma provides the upper bound for $\hat{x}_i(t) - x_i,\ i \in V$.

Lemma 7. *Consider (31), then $\forall i \in V$, $\hat{x}_i(t) - x_i \leq \sum_{n=0}^{\mathcal{D}(\mathcal{G})-2} L_2^n L_1 \epsilon$ with $\mathcal{D}(\mathcal{G})$ defined in Lemma 5.*

Proof. Considering the sequance $n_1, n_2, ..., n_T$ where n_{k+1} is the true constraining node of n_k for $k \in \{1, 2, .., T-1\}$. As $T \leq \mathcal{D}(\mathcal{G})$, the result holds if for $t \geq i-1$

$$\hat{x}_{n_i}(t) - x_{n_i} \leq \begin{cases} \sum_{n=0}^{i-2} L_2^n L_1 \epsilon & i \in \{2, \cdots, T\} \\ 0 & i = 1 \end{cases} \tag{32}$$

We prove (32) by introduction. It is true for $i = 1$ as $x_{n_1} = s_{n_1}$ and $\hat{x}_{n_1}(t) \leq s_{n_1}$ for all t. (32) also holds for $i = 2$ as for $t \geq 1$

$$\begin{aligned} \hat{x}_{n_2}(t) &\leq f(\hat{x}_{n_1}(t), e_{n_1 n_2}(t)) \\ &\leq f(\hat{x}_{n_1}(t), e_{n_1 n_2} + \epsilon) \\ &\leq f(s_{n_1}, e_{n_1 n_2} + \epsilon) \\ &\leq x_{n_2} + L_1 \epsilon \end{aligned}$$

Suppose (32) holds for some $i \in \{1, 2, ..., T-1\}$, then for $t \geq i-1$ and n_{i+1}

$$\begin{aligned} \hat{x}_{n_{i+1}}(t+1) &\leq f(\hat{x}_{n_i}(t), e_{n_i n_{i+1}}(t)) \\ &\leq f(\hat{x}_{n_i}(t), e_{n_i n_{i+1}}) + L_1 \epsilon \\ &\leq f(x_{n_i} + \sum_{n=0}^{i-2} L_2^n L_1 \epsilon, e_{n_i n_{i+1}}) + L_1 \epsilon \\ &\leq f(x_{n_i}, e_{n_i n_{i+1}}) + L_2 \sum_{n=0}^{i-2} L_2^n L_1 \epsilon + L_1 \epsilon \\ &= x_{n_{i+1}} + \sum_{n=0}^{i-1} L_2^n L_1 \epsilon \end{aligned}$$

Thus (32) and our result follows. □

To address the lower bound of $\hat{x}_i - \hat{x}_i(t)$, we use an approach like *comparison principle* [1]. First, we introduce the following definition.

Definition 4. *Given a graph $\mathcal{G}$, we define a $\mathcal{G}^-$, and $\mathcal{G}^-$ is a shrunken version of $\mathcal{G}$ such that, $\forall i \in V$ and $j \in \mathcal{N}(i)$ in $\mathcal{G}$, e_{ij} becomes e_{ij}^- in $\mathcal{G}^-$, and e_{ij}^- obeys*

$$e_{ij}^- = e_{ij} - \epsilon \tag{33}$$

Lemma 8. *Given a graph $\mathcal{G}$ and its shrunken version $\mathcal{G}^-$ defined in Definition 4, consider*

$$\hat{X}_i(t+1) = \min\left\{ \min_{k \in \mathcal{N}(i)} \left\{ f\left(\hat{X}_k(t), e_{ik} - \epsilon\right) \right\}, s_i \right\}. \tag{34}$$

for $\mathcal{G}^-$ and (31), suppose for all $i \in V$, $\hat{X}_i(0) = \hat{x}_i(0)$. Then $\hat{x}_i(t) \geq \hat{X}_i(t), \forall t \geq 0$ and $i \in V$.

Proof. We prove by induction, the result holds for $t = 0$, suppose for some $t \geq 0$, $\hat{x}_i(t) \geq \hat{X}_i(t), \forall i \in V$. Suppose $j \in \mathcal{N}(i)$ is a current constraining node of i at time t in (31) while $k \in \mathcal{N}(i)$ is a current constraining node of i at time t in (34). Then we have

$$\begin{aligned} \hat{X}_i(t+1) &= f(\hat{X}_k(t), e_{ik} - \epsilon) \\ &\leq f(\hat{X}_j(t), e_{ij} - \epsilon) \\ &\leq f(\hat{X}_j(t), e_{ij}(t)) \\ &\leq f(\hat{x}_j(t), e_{ij}(t)) \\ &= \hat{x}_i(t+1) \end{aligned} \tag{35}$$

□

Thus the estimates offered by (31) are uniformly lower bounded by the estimates, $\hat{X}_i(t)$. Now we introduce the following lemma.

Lemma 9. *$\forall i \in V$, $x_i \leq X_i + \sum_{n=0}^{D(\mathcal{G}^-)-2} L_2^n L_1 \epsilon$ where $\mathcal{G}^-$ is in Definition 4 and $D(\cdot)$ is in Lemma 5.*

Proof. Consider nodes $n_1, n_2, ..., n_{T_1}$ such that $X_{n_1} = s_{n_1}$, and for all $i \in \{1, ..., T_1 - 1\}$, n_i is a true constraining node of n_{i+1} in $\mathcal{G}^-$. Each node is in one such sequence. We assert that

$$x_{n_i} - X_{n_i} \leq \begin{cases} \sum_{n=0}^{i-2} L_2^n L_1 \epsilon & i \in \{2, \cdots, T_1\} \\ 0 & i = 1 \end{cases} \tag{36}$$

As $x_{n_i} \leq s_{n_i}$, (36) holds for $i = 1$. As n_i and n_{i+1} are neighbors in both $\mathcal{G}$ and $\mathcal{G}^-$, n_i is the true constraining node of n_{i+1} in $\mathcal{G}^-$, for $i = 2$, it follows that

$$\begin{aligned} x_{n_2} &\leq f(x_{n_1}, e_{n_1 n_2}) \\ &\leq f(X_{n_1}, e_{n_1 n_2}) \\ &\leq f(X_{n_1}, e_{n_1 n_2} - \epsilon) + L_1 \epsilon \\ &\leq X_{n_2} + L_1 \epsilon \end{aligned}$$

Suppose it holds for some $i \in \{1, ..., T_1 - 1\}$. As $T_1 \leq \mathcal{D}(\mathcal{G}^-)$, (36) holds as

$$\begin{aligned} x_{n_{i+1}} &\leq f(x_{n_i}, e_{n_i n_{i+1}}) \\ &\leq f(X_{n_i} + \sum_{n=0}^{i-2} L_2^n L_1 \epsilon, e_{n_i n_{i+1}}) \\ &\leq f(X_{n_i}, e_{n_i n_{i+1}}) + L_2 \sum_{n=0}^{i-2} L_2^n L_1 \epsilon \\ &\leq f(X_{n_i}, e_{n_i n_{i+1}} - \epsilon) + L_1 \epsilon + L_2 \sum_{n=0}^{i-2} L_2^n L_1 \epsilon \\ &= X_{n_{i+1}} + \sum_{n=0}^{i-1} L_2^n L_1 \epsilon \end{aligned}$$

Then our result follows. □

As $\mathcal{G}^-$ a perturbation free graph, based on our theorem in section III, after a finite time T_1, $\forall i \in V$ in $\mathcal{G}^-$, $\hat{X}_i(t)$ will converge to X_i. Then we have the following lemma.

Lemma 10. *Consider (31), and suppose $\forall i \in V$, $\hat{X}_i(t) = X_i$ for $t \geq T_1$, then for $t \geq T_1$*

$$x_i - \hat{x}_i(t) \leq \sum_{n=0}^{D(\mathcal{G}^-)-2} L_2^n L_1 \epsilon \tag{37}$$

Proof.

$$\begin{aligned} x_i - \hat{x}_i(t) &\leq X_i + \sum_{n=0}^{D(\mathcal{G}^-)-2} L_2^n L_1 \epsilon - \hat{x}_i(t) & (38) \\ &\leq X_i + \sum_{n=0}^{D(\mathcal{G}^-)-2} L_2^n L_1 \epsilon - X_i & (39) \\ &= \sum_{n=0}^{D(\mathcal{G}^-)-2} L_2^n L_1 \epsilon \end{aligned}$$

Here (38) comes from Lemma 9. As $\hat{X}_i(t)$ converge to X_i for $t \geq T_1$ in graph $\mathcal{G}^-$, from Lemma 8, $\hat{x}_i(t) \geq X_i$ for $t \geq T_1$, then (39) follows. □

V. Simulations

In this section, we empirically confirm the results presented in the prior sections through simulations, and four simulations we presented have different settings.

In our first simulation, 200 nodes, two of which are sources, are randomly distributed in a 4×1km field, communicating over a 0.6 km radius, run synchronously. Two source nodes, defined as 1 and 2 are located at [0.25 0.5] and [3.75 0.5], respectively. The algorithm we use here follows (1) with

$$\hat{x}_i(t+1) = \min \left\{ \min_{k \in \mathcal{N}(i)} \{\hat{x}_k(t) + e_{ik}\}, s_i \right\} \tag{40}$$

and

$$s_i = \begin{cases} 5 & i = 1 \\ 0.1 & i = 2 \\ \infty & \text{else} \end{cases}$$

Here e_{ik} refers to the distance between node i and k, this is the same as ABF except that s_i is redefined. Source with $s_1 = 5$ can be seen as a low-speed link while source with $s_2 = 0.1$ is a high-speed one, non-source nodes have routed through sources will be labeled with the color of the source.

In Figure 1, initially, non-source nodes choose to route to the source closer to them by distance, while some nodes far from sources remain not routed. Shortly, all non-source nodes will route through one of the two sources. Finally, all nodes, including the low-speed link will route to the high-speed link.

After all nodes routed to the high-speed link, we turn it off so that there is only one source in the network. As we can see from Figure 2, all the nodes will gradually route to the sole source, which is a low-speed link.

In Figure 3, 500 nodes are randomly distributed in a 4×4km area, communicating over a 0.2 km radius, run synchronously. Five source nodes marked as solid circles with different colors will broadcast their IDs, and a non-source node will be in the same color as the source if it receives the source ID. The algorithm used here follows (40) with a slightly different s_i defined as follows

$$s_i = \begin{cases} 0 & i \in S \\ \infty & i \notin S \end{cases}$$

where S represents the set of sources. As Figure 3 shows, each non-source node will finally receives the ID of its nearest source, and the network will be partitioned into 5 parts.

We also carry out simulations by using (1) under non-Euclidean metric. In Figure 4, 500 nodes are randomly distributed in a 4×4km field, communicating over a 0.6 km radius, run synchronously. The sole source at (0.3, 0.3) keeps broadcasting its ID. In the middle of the area, there exists a 2.5×2.5km radiation zone, and nodes outside this region will never get into this region by choosing not to communicate with nodes inside it. For nodes inside the radiation zone, it will leave the radiation zone if one of its neighbors has received the source ID. The whole area is radiating contaminated materials, suppose node i is communicating with node j in time t, then

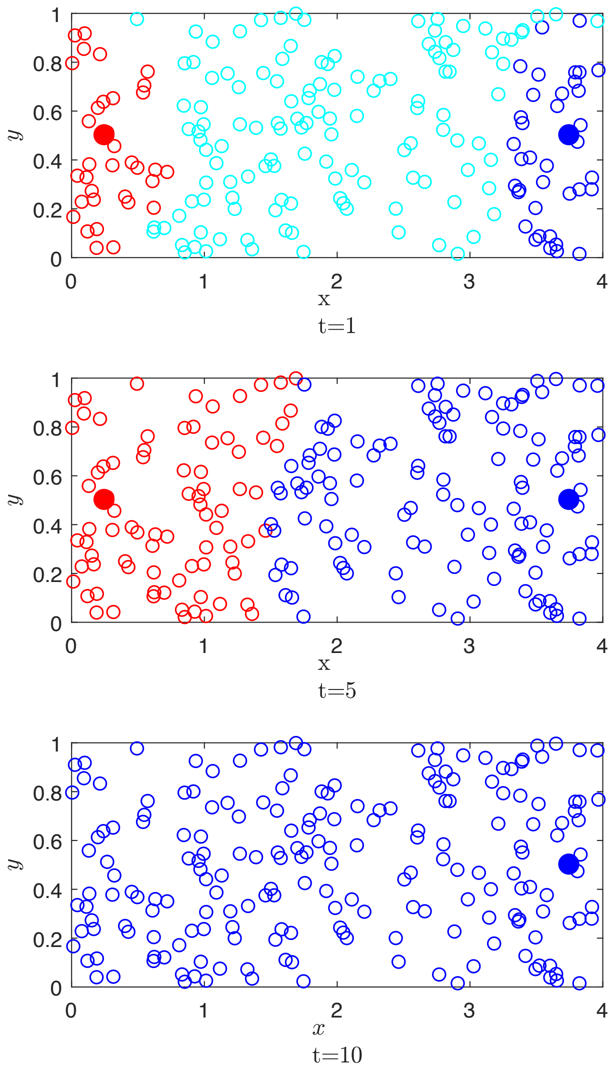

Fig. 1. In this example, 200 nodes are randomly distributed in a 4×1 km area, and two source nodes marked as solid red circle and solid blue circle are low-speed link and high-speed link, respectively. Here circles in blue represent nodes routing through high-speed link while circles in red represent nodes routing through low-speed link, circles in light blue are nodes not routing through any source. It shows that finally all nodes will route to the high-speed link.

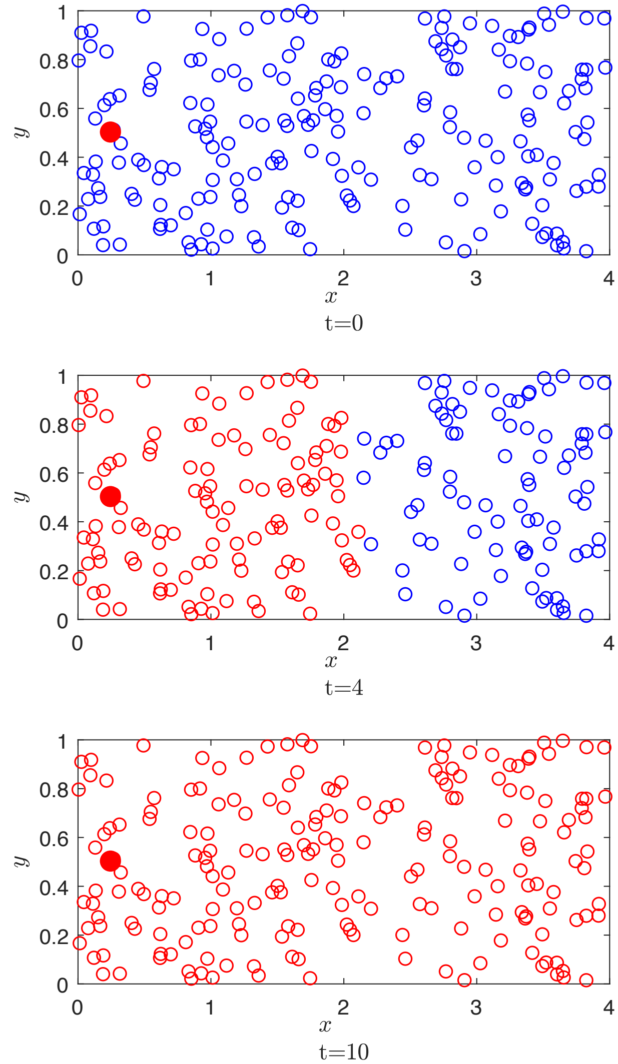

Fig. 2. In this example, after turning the high-speed link off, all nodes will gradually route to the sole source, the low-speed one marked as solid red circle.

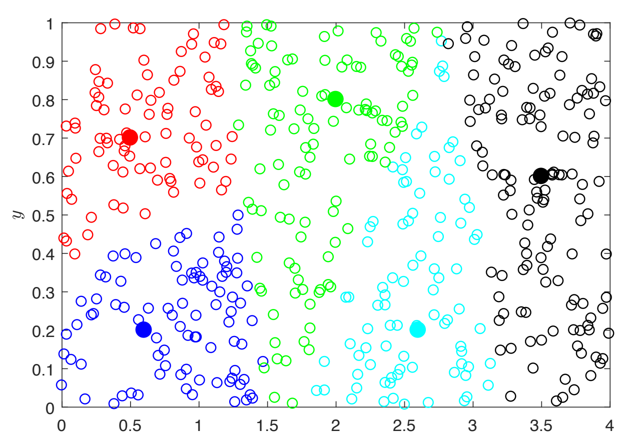

Fig. 3. In this example, 500 nodes are randomly distributed in a 4×4 km area, and among them there are 5 source nodes, represented by blue, red, black, green and lihgt blue solid balls. Their locations are (0.6,0.2), (0.5,0.7), (3.5,0.6), (2,0.8), (2.6,0.2). Each source will broadcast its ID, and nodes will have the same color as the source if they receive the source ID. It shows that the field will be partitioned by five nodes.

it will receive a number of $1000 * e_{ij}$ units of contaminated materials if j is in the radiation zone, while i will only get $1 * e_{ij}$ units if j is not in the radiation zone, besides, the calculation of contaminate materials is cumulative.

As Figure 4 shows, nodes outside the radiation zone will detour around the radiation zone. For nodes inside the radiation zone, those closer to the source will have fewer contaminated materials as they spend less time getting rid of the radiation zone.

In Figure 5, 120 nodes, including the sole source, are randomly distributed in a 4×1km field, communicating over a 1 km radius, run synchronously. The algorithm used is as follows,

$$\hat{x}_i(t+1) = \min\left\{\min_{k\in\mathcal{N}(i)}\left\{\hat{x}_k(t) + 0.5sin(\hat{x}_k(t)) + 1\right\}, s_i\right\}$$

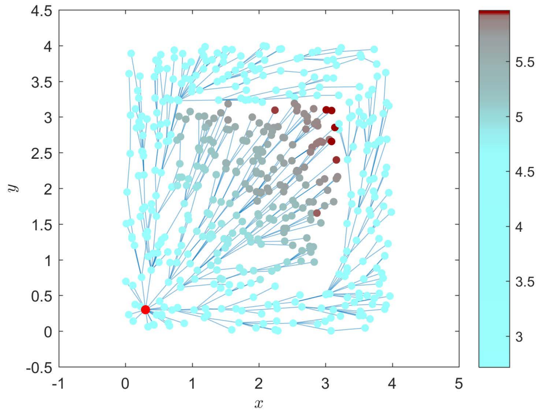

Fig. 4. In this example, 500 nodes are randomly distributed in a 4×4 km area, and there is a 2.5×2.5km radiation zone in the middle of the field. Those communicating with nodes inside the radiaton zone will receive a 1000 unit/km dose of contaminated materials, while those communicating with nodes outside the zone will receive a 1 unit/km dose. The colorbar represents the amount of contaminated materials, and uses a log-10 scale. It shows that nodes outside the radiation zone will detour around the zone, and for nodes inside the zone, those closer to the source appear to be less contaminated.

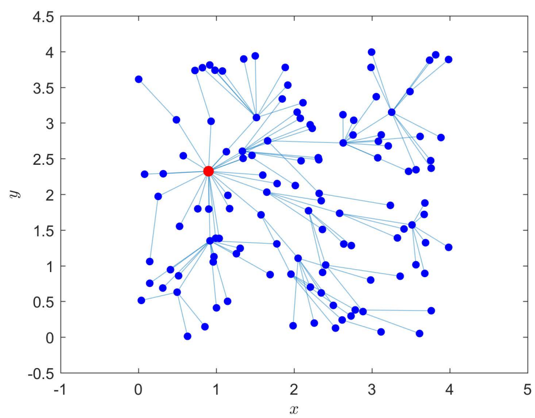

Fig. 5. In this example, 120 nodes are randomly distributed in a 4×1 km area, the solid red circle represents the source, and solid blue circles represent non-source nodes. We apply (1) to this network with $f(a,b) = a + 0.5sin(a) + 1$. It shows that each node will finally find a shortest path towards the source.

such that $f(\cdot,\cdot)$ is a non-linear function, and s_i follows

$$s_i = \begin{cases} 0 & i \in S \\ \infty & i \notin S \end{cases}$$

where S represents the set of sources. From Figure 5, we can see that all nodes will finally find a shortest path towards the source, which verify our theories in section III.

VI. Conclusion

We have provided a global uniform asymptotic stability of a general building block of Aggregate Computing that spreads information through a network. We have also established ultimate bounds in face of persistent perturbations, with an additional Lipschitz condition on the upadate kernel. We view this as an early step towards our long term goal of provong stability of feedback interconnections featuring these blocks.

References

[1] H. K. Khalil, *Nonlinear Systems*, Prentice Hall, 2002.

[2] M. Vidyasagar, "Decomposition techniques for large-scale systems with nonadditive interactions: Stability and stabilizability," *IEEE Transactions on Automatic Control*, vol. 25, pp. 773 – 779, 1980.

[3] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, pp. 215 – 233, 2007.

[4] J. Baillieul and A. Suri, "Information patterns and hedging brockett's theorem in controlling vehicle formations," in *Proceedings of 42nd IEEE Conference on Decision and Control*, 2003.

[5] S. Dasgupta, and B. D. O. Anderson, "Controlling rectangular formations", in *Proceedings of AUCC*, 2011, Melbourne, Australia.

[6] T. H. Summers, C. Yu, S. Dasgupta, and B. D. O. Anderson, "Control of minimally persistent leader-remote-follower and coleader formations in the plane," *IEEE Transactions on Automatic Control*, vol. 56, pp. 2778 – 2792, 2011.

[7] J. Beal and M. Viroli, "Building blocks for aggregate programming of self-organising applications," in *Workshop on Foundations of Complex Adaptive Systems (FOCAS)*, 2014.

[8] M. Viroli and D. Ferruccio, "A calculus of self-stabilising computational fields," in *Coordination 2014*, 2014, pp. 163–178.

[9] J. Beal, D. Pianini, and M. Viroli, "Aggregate programming for the internet of things," *IEEE Computer*, vol. 48, no. 9, pp. 22–30, September 2015.

[10] A. Kumar, J. Beal, S. Dasgupta, and R. Mudumbai, "Toward predicting distributed systems dynamics," in *Spatial and COllective PErvasive Computing Systems (SCOPES)*. IEEE, September 2015, pp. 68–73.

[11] M. Viroli, J. Beal, F. Damiani, and D. Pianini, "Efficient engineering of complex self-organizing systems by self-stabilising fields," in *IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*. IEEE, September 2015, pp. 81–90.

[12] J. Beal, J. Bachrach, D. Vickery, and M. Tobenkin, "Fast self-healing gradients," in *ACM Symp. on Applied Computing*, 2008.

[13] J. Beal, "Flexible self-healing gradients," in *Proceedings of the 2009 ACM Symposium on Applied Computing*. 2009, pp. 1197–1201, ACM.

[14] M. Viroli, G. Audrito, J. Beal, F. Damiani, and D. Pianini, "Engineering Resilient Collective Adaptive Systems by Self-Stabilisation," in *ACM Transactions on Modeling and Computer Simulation* to appear. Also available from:https://arxiv.org/abs/1711.08297.

[15] B. D. O. Anderson, R. R. Bitmead, C. R. Johnson, P. V. Kokotovic, R. L. Kosut, I. M. Y. Mareels, L. Praly, and B. D. Riedle, *Stability of Adaptive Systems: Passivity and Averaging Analysis*, MIT Press, 1986.

[16] Dasgupta, S., Anderson, B.D.O and Kaye, R.J., "Identification of physical parameters in structured systems," *Automatica*, March 1988, vol. 24, pp 217-225.

[17] S. Dasgupta, C. R. Johnson Jr and A.M. Baksho, "Signsign LMS convergence with independent stochastic inputs", *IEEE Transactions on Information Theory*, pp. 197-201, 1990.

[18] W. Hahn, *Stability of Motion*, Prentice Hall, 1967.

[19] R. E. Bellman, "On a routing problem," *Quarterly of Applied Mathematics*, vol. 16, pp. 87–90, 1958.

[20] L. R. Ford Jr., "Network flow theory," Tech. Rep. Paper P-923, RAND Corporation, 1956.

[21] M. Fu and S. Dasgupta, "Parametric lyapunov functions for uncertain systems: The multiplier approach," *Advances in linear matrix inequality methods in control*, pp. 95–108, 2000.

[22] Z. Jiang, I. M. Y. Mareels, and Y. Wang, "A lyapunov formulation of the nonlinear small-gain theorem for interconnected ISS systems," *Automatica*, vol. 32, pp. 1211 – 1215, 1996.

[23] M. Viroli, G. Audrito, J. Beal, F. Damiani, and D. Pianini, "Engineering Resilient Collective Adaptive Systems by Self-Stabilisation," in *ACM Transactions on Modeling and Computer Simulation* to appear. Also available from:https://arxiv.org/abs/1711.08297.

[24] S. Dasgupta and J. Beal, "A Lyapunov Analysis for the Robust Stability of an Adaptive Bellman-Ford Algorithm," in *Proceedings of 55th IEEE Conference on Decision and Control*, 2016.

[25] Y. Mo, S. Dasgupta and J. Beal, "Robustness of the Adaptive Bellman-Ford Algorithm: Global Stability and Ultimate Bounds", submitted to *IEEE Transactions on Automatic Control.*