Robustness of the Adaptive Bellman-Ford Algorithm: Global Stability and Ultimate Bounds

Yuanqiu Mo, Student Member, IEEE, Soura Dasgupta, Fellow, IEEE, and Jacob Beal, Senior Member, IEEE

Abstract—Self-stabilizing distance estimation algorithms are an important building block of many distributed systems, such as seen in the emerging field of Aggregate Programming. Their safe use in feedback systems or under persistent perturbations has not previously been formally analyzed. Self-stabilization only involves eventual convergence, and is not endowed with robustness properties associated with global uniform asymptotic stability and thus does not guarantee stability under perturbations or feedback. We formulate a Lyapunov function to analyze the Adaptive Bellman-Ford distance estimation algorithm and use it to prove global uniform asymptotic stability, a property which the classical Bellman-Ford algorithm lacks. Global uniform asymptotic stability assures a measure of robustness to structural perturbations, empirically observed by us in a previous work. We also show that the algorithm is ultimately bounded under bounded measurement error and device mobility and provide a tight bound on the ultimate bound and the time to attain it.

Index Terms—Ultimate Boundedness, Lyapunov Function, Stability, Robustness, Aggregating Computing, Internet of Things.

I. INTRODUCTION

As our world becomes increasingly interdependent and interconnected, recent decades have seen a proliferation of complex networked and distributed systems, typically composed of multiple subsystems (physical and logical), which may themselves be distributed. The resilience, safety, and dynamical performance of such systems are of critical importance, but in general it is still extremely difficult to analyze compositions of distributed algorithms.

Robustness and stability have been addressed for limited classes of large-scale distributed systems in the controls literature for decades, using a mature set of tools from stability theory [1]. The ongoing dispersion of services into local devices, as occurs in the domains of smart cities, tactical information sharing, personal and home area networks, and the Internet of Things (IoT) [11], however, poses new and challenging problems for analysis and design. In particular, realizing the potential of these domains requires devices to

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. Approved for public release, distribution unlimited (DARPA DISTAR case 28473, 9/18/2017).

Y. Mo and S. Dasgupta are with the University of Iowa, Iowa City, Iowa 52242 USA (e-mail: yuanqiu-mo@uiowa.edu, soura-dasgupta@uiowa.edu). S. Dasgupta is also a Visiting Professor at Shandong Computer Science Center, Shandong Provincial Key Laboratory of Computer Networks, China.

J. Beal is with Raytheon BBN Technologies, Cambridge, MA, USA 02138 USA (e-mail: jakebeal@ieee.org)

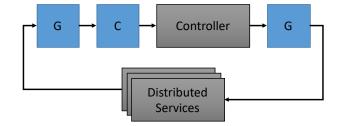


Fig. 1. Distributed systems are often composed of multiple interacting distributed subsystems: for example, in this notional example system, a set of distributed services are managed by a controller device, which accepts load information as input and provides a resource allocation plan as its output. The blue subsystems are aggregate computing building blocks. The two to the left of the controller are composed to implement information collection. The resource allocation plan is disseminated by the block to the right.

interact safely and seamlessly with other devices in their vicinity through low latency peer to peer communications, often in feedback loops, with individual blocks independently subjected to perturbations due to mobility, uncertainty and noise. At the same time, however, these systems are open, in the sense that they are expected to support an unbounded and rapidly evolving collection of distributed services, represented by algorithms. To address these challenges, we need a framework for analyzing the *composition* of distributed services, both to guide service engineering and to support runtime monitoring and management of complex compositions of dispersed services.

Aggregate computing offers a potential approach to this challenge by viewing the basic computing unit as a physical region comprising a collection of interacting computing devices, rather than an individual physical device [11]. In particular, [11] introduces a separation of concerns into different abstraction layers, much like the OSI model [12] does for communication between individual devices, factoring the overall task of distributed system design into sub-tasks of device-level communication and discovery, coherence between collective and local operations, resilience, and programmability. Within this framework, we focus specifically on the "basis set" approach to resilient design introduced in [13] and [14], which show that a broad class of dispersed services can be described by composition of 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. Figure 1 illustrates a typical example of such systems, in this case making use of G and C blocks. A key function of these

blocks is to coordinate interacting devices using packet-based message passing in a highly distributed environment.

While the dynamics of these basis set systems appear pragmatically amenable to effective composition [27]–[30], no framework for analyzing the stability of such compositions as yet exists. Our long term aim is thus to delineate the circumstances under which compositions of these blocks do or do not ensure stable behavior. As explained in Section II, this in turn is facilitated by developing a Lyapunov framework that can be used to establish (i) global uniform asymptotic stability of these individual blocks under ideal conditions and (ii) ultimate bounds under persistent perturbations. Roughly speaking, global uniform asymptotic stability implies that convergence characteristics are independent of the initial time. As noted through an example in Section VII-A, it is also important to characterize the time it takes to converge and/or to attain ultimate bounds.

As a step toward this goal, we analyze an archetypal and commonly used version of the aggregate computing *G*-block, the Adaptive Bellman-Ford (ABF) algorithm, a globally asymptotically stable variant of the classical Bellman-Ford algorithm [20], [21]. As explained in Section II, the classical Bellman-Ford algorithm [20], [21] is not globally stable. This work builds on a preliminary conference version [17], which lacks proofs, characterization of convergence time, and perturbation analysis. We also report several structural insights, the most significant being that distributed algorithms such as ABF require unusual Lyaponov functions, as the state update of a node does not rely on its current value but rather on the value of some distinguished neighbors. The work in this paper thus completes the first critical step toward a general theory of understanding the dynamics of aggregate computing.

Following a review of related work and background in Section II, we formalize the problem in Section III and the ABF algorithm in Section IV. In Section V, we formulate a Lyapunov function based on greatest overestimation error and least underestimation error, prove that this function is non-increasing, and use it to prove global uniform asymptotic stability of ABF in Section VI. We next provide a tight bound on convergence time for ABF, in terms of the structural characteristics of the underlying graph, identifying an intrinsic asymmetry in the behavior of overestimates and underestimates. In Section VII, we apply the Lyapunov analysis to behavior under persistent perturbations with two physical interpretations: bounded motion of nodes and noisy distance measurements, then show that under such perturbations ABF is ultimately bounded (per the definition in [1]), furnish a tight ultimate bound, and tightly upper bound the time taken to attain it. Section VIII gives simulation and Section IX summarizes and discusses implications and future work.

II. RELATED WORK AND APPROACH

Robustness and stability studies for some large-scale distributed systems spawns decades with [2] serving as a seminal

 1 The stationary point x^* of a system x(t+1)=f(x(t),t) is globally uniformly asymptotically stable, if for all $x(t_0)$ and $\epsilon>0$, there exists a T independent of the initial time t_0 , such that $\|x(t)-x^*\| \leq \epsilon$, for all $t\geq T$. The system itself is called globally uniformly asymptotically stable.

(though not the first) example. In recent years such research has been dominated by the control of multiagent systems, exemplified by consensus theory (see [3] and references therein) and formation control, (see [4]- [8] and references therein). Their analysis has leveraged classical stability theory tools like Lyapunov theory, passivity theory [9], [10], center manifold theory [5], [8], and the Perron-Frobenius Theorem [3].

As a first step in understanding the robust stability of arbitrary compositions of blocks in Aggregate Computing, it is important to understand how these individual systems behave under perturbations. More precisely are there stability properties robust to these perturbations? Does stability in the ideal unperturbed setting translate to acceptable behavior in the face of perturbations? While these blocks are known to be self-stabilizing, [15], robust behavior cannot be deduced by the mere demonstration of self-stabilization or even asymptotic stability. Rather, as is now well understood in the adaptive systems literature, [18], one should instead show global uniform asymptotic stability of the unperturbed system, as it guarantees total stability [19], an ability to withstand modest departures from idealizing assumptions. In particular, in a totally stable system the state remains bounded in face of sufficiently small perturbations in the system equations. Thus, as in the literature of multiagent systems, we begin to develop a framework for analyzing the stability, safety, and dynamical behavior of arbitrary compositions of a basis set of distributed algorithms by leveraging Lyapunov based tools to prove the global uniform asymptotic stability of one distinguished block, and to explicitly analyze its behavior under perturbations.

Specifically, we focus on a commonly used, archetypal G-block, many of whose behaviors are inherited by other types of G-blocks. The G-block in question is the ABF algorithm, an adaptive version of the classical Bellman-Ford algorithm, [20], [21], which estimates in a distributed fashion distances of nodes in an undirected graph from the nearest in a designated subset of source nodes. This variant is needed as the classical Bellman-Ford algorithm is not globally uniformly asymptotically stable mandating as it does that all initial distance estimates be larger than the actual distances. Under persistent topological perturbations (e.g. from interaction with other components in a feedback system) these stringent initial condition requirements cannot be met.

Our first contribution is to formulate a Lyapunov function, prove that it is always non-increasing, and then use it to demonstrate the global uniform asymptotic stability of ABF. The Lyapunov function itself is the sum of two terms: The largest positive distance estimation error, i.e. the greatest overestimation error, Δ^+ and the magnitude of the most negative estimation error, i.e. the greatest underestimation error, Δ^- . We show that this Lyapunov function is nonincreasing and use it to prove global uniform asymptotic stability. This formally validates the observed robustness to structural perturbations of the underlying graph, including perturbations caused by certain types of feedback relations empirically analyzed in [30].

The second contribution is to provide a *tight bound* on the time to converge, in terms of the structural characteristics of the underlying graph. This is of more than just academic interest, as it reveals some important dependencies

that influence distance estimation algorithms. Our preliminary investigation in [26] suggests similar interdependencies also characterize the behavior of algorithms representing C blocks. The graph driven dependencies that we identify cause an intrinsic asymmetry in the respective behavior of Δ^+ and Δ^- . Specifically, we show that Δ^+ converges within what we call the *effective diameter* of the graph. On the other hand, while the convergence of Δ^- is still bounded proportional to effective diameter time, it could be much slower than for Δ^+ , and is in fact limited by short links in the graph.

We apply the Lyapunov analysis to behavior under persistent perturbations with two physical interpretations: First, that nonsource nodes experience persistent bounded motion around a nominal point in space, and second, that the measurement of distance between nodes is noisy. We show that under such persistent, de facto or actual, perturbations the algorithm is ultimately bounded, furnish a tight ultimate bound, and tightly upper bound the time taken to attain it. We provide a concrete aggregate computing application involving a cascade combination of a G and a C block, where the ultimate bound as well as the time taken to attain it are both of critical importance. Beyond that, the demonstration of the ultimate bound goes straight to the core objective that animates this line of research: developing a framework for stability analysis under potential feedback interconnections. Ultimate boundedness unveils the prospect of developing variants of the small gain theorem [1] that can be employed in such analysis. While ultimate boundedness by itself is not enough to invoke the classical small gain theorem, there are more sophisticated variants of this theorem that rely on ultimate bounds [22], and have been used to demonstrate closed loop stability.

We also note that ABF is only one of a large family of algorithms for calculating shortest paths through graphs, all of which are potentially suited to serve as G-blocks, e.g., [28], [29], and [31]. Numerical comparisons between these and ABF given in [28], [29], [31], and [27] show that these alternative algorithms can provide better dynamical behavior for certain applications. We focus on ABF rather than these alternatives, however, because it is the simplest and most tractable for analysis, yet closely enough related to the others that the analytical approach here is likely to generalize to their analysis as well. More distantly related algorithms include various search and path planning algorithms (e.g., [32]- [37]), which have a fundamentally different model focused on computation (e.g., graph processing on a single or parallel processor machine) rather than coordination of devices via messagepassing in a highly-distributed environment. Their focus is on minimizing the number of operations (communications) that it takes to converge in the smallest portion of the graph relevant to navigation between a source and a destination. While they differ in their ability to accommodate various structural graph changes like new nodes, edge length changes and mobility, their analysis assumes that the changes are sufficiently slow to permit convergence between successive instances of structural changes. They are designed to acheive as fast a convergence as is possible, so that faster changes can be tolerated. In contrast the analysis here accommodates persistent perturbations in every iteration of the algorithm, that

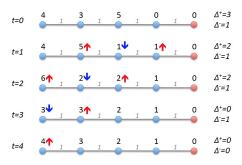


Fig. 2. Illustration of ABF (adapted from [30]). Individual distance estimates may go up and down, but the greatest overestimate (Δ^+) and least underestimate (Δ^-) are monotonic. This example shows a line network of five nodes (circles, source red, others blue) with unit edges (grey links); distance estimates evolve from initial t=0 to converge to their correct values at t=4. The numbers on the edges are the edge lengths. The numbers on the nodes are their current distance estimates.

never settle down to permit such convergence. Further, rather than focusing just on the subgraph important to navigation from a source to a destination, a G-block like ABF must spread distance information throughout the network. Continuous time shortest path algorithms, such as the continuous time Bellman-Ford [38] are also related but do not fit well with a message-passing paradigm: a packet-based message passing environment is inherently discrete and benefits strongly from finite time convergence, which allows communication optimization based on non-changing values.

III. PRELIMINARIES AND THE PROBLEM STATEMENT

In this section we set up the rest of the paper by outlining the general framework and specifying a distance estimation problem that serves as a typical example of a *G*-block for information spreading over a network. Section III-A provides a graphical framework. Section III-B defines the problem.

A. The Graphical Framework

We consider undirected graphs G=(V,E), with V the set of vertices and E the set of undirected edges: Each node is a device. Edges have a dual meaning. They indicate the existence of a communication link between devices. They also define paths between nodes, in the sense that a path exists between two nodes if they are connected through a set of edges. We call node i a neighbor of node j if there is an edge between i and j. We define $\mathcal{N}(i)$ as the set of all neighbors of node i. Further, no node is deemed to be its own neighbor: $i \notin \mathcal{N}(i)$.

We define the edge length between node i and node j as e_{ij} , and assume that there exists an e_{\min} such that:

$$e_{ij} > e_{\min} > 0, \ \forall i \in V \text{ and } j \in \mathcal{N}(i),$$
 (1)

i.e., edge lengths between neighbors are all positive. Thus in the graph in Figure 2 each edge length is 1. The distance d_{ij} between two nodes is the shortest walk from i to j (the distance to itself, of course, being zero). Thus, the distance between the third and the last nodes in Figure 2 is 2. The principle of optimality specifies the recursion:

$$d_{ij} = \min_{k \in \mathcal{N}(i)} \{e_{ik} + d_{jk}\}. \tag{2}$$

This is also in effect a statement of the triangle inequality.

A subset S of the nodes in the graph will form a source set. Our goal is to find the shortest distance between each node and the source set S. More precisely we must find:

$$d_i = \min_{k \in S} \{d_{ik}\}. \tag{3}$$

In view of (2), d_i obeys the recursion:

$$d_i = \min_{k \in \mathcal{N}(i)} \{e_{ik} + d_k\} \tag{4}$$

and

$$d_i = 0, \ \forall i \in S. \tag{5}$$

To avoid trivialities we make the following assumption:

Assumption 1. The graph G = (V, E) is connected, undirected, $S \neq V$, with edge lengths e_{ij} obeying (1) and the distance d_i of node i from the source set S obeying (4).

We now make an additional definition:

Definition 1. A k that minimizes the right hand side of (4) is a true constraining node of $i \in V \setminus S$. As there may be two nodes k and l such that $d_l + e_{il} = d_k + e_{ik}$, a node may have multiple true constraining nodes. The set of true constraining nodes of a node $i \in V \setminus S$ is denoted as C(i).

In view of (4) the following holds:

$$d_k < d_i, \ \forall k \in \mathcal{C}(i).$$
 (6)

The sets C(i) represents a structural characteristic of G = (V, E) and S. The following related definition is crucial:

Definition 2. For a connected graph G, consider any sequence of nodes such that the predecessor of each node is one of its true constraining nodes. Define $\mathcal{D}(G)$, the effective diameter of G, as the longest length such a sequence can have in G (i.e., the diameter of the shortest path forest rooted at S).

The effective diameter is always bounded, per the following:

Lemma 1. Under Assumption 1, $\mathcal{D}(G)$ defined above is finite.

Proof. As defined in Definition 2, consider a sequence of nodes k_i in G such that, for all k_{i-1} is a true constraining node of k_i . Since there are only a finite number of nodes in the graph, the only way that $\mathcal{D}(G)$ can be infinite is if for some i > j, $k_i = k_j$. From (6) this leads to the contradiction:

$$d_{k_i} > d_{k_i} = d_{k_i}. (7)$$

Every full sequence of the form described in Definition 2 commences at a source and ends at an extreme node.

This and other concepts thus far presented are exemplified through the two graphs labeled G and G^- in Figure 3. In G the true constraining node of D is E, as:

$$d_C + e_{CD} = 2.1 + 0.7 > d_E + e_{ED} = 1.3 + 1.3.$$

In contrast in G^- , the true constraining node of D is C as:

$$d_C + e_{CD} = 0.3 + 0.1 < d_E + e_{ED} = 0.7 + 0.7.$$

The longest sequence where each node is a true constraining node of its predecessor, is $\{S, A, B, C\}$ in G while it is $\{S, A, B, C, D\}$ in G^- . Thus $\mathcal{D}(G) = 4$ and $\mathcal{D}(G^-) = 5$.

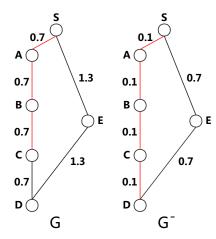


Fig. 3. Examples of effective diameter, showing edge length as labels on edges. In the left graph G, $\mathcal{D}(G)=4$ and comes from the sequence S, A, B and C, where each node is the true constraining node of its successor. On the other hand in the right graph G^- , $\mathcal{D}(G^-)=5$ and comes from the sequence S, A, B, C and D.

B. Problem Statement

The distance estimation we desire must be recursive and *distributed* in the sense that in executing the algorithm at time t, the i-th node knows only:

- (A) Its edge length from each of its neighbors.
- (B) The current estimated distance of its neighbors from the source set, i.e. $\hat{d}_i(t)$ for all $j \in \mathcal{N}(i)$.

These requirements are practical, as the existence of edges between neighbors means they can communicate their distance estimates to each other. The classical Bellman-Ford algorithm [20], [21], is a well known solution to this problem but is not globally stable. Instead in Section IV we describe an adaptive version of this algorithm (see [28]) which we have empirically studied in [30] in isolation and in feedback loops.

IV. ADAPTIVE BELLMAN-FORD ALGORITHM

In the classical Bellman-Ford algorithm [20], [21] distance from every node in an arbitrary graph to a designated source node is estimated by the relaxation of a triangle inequality constraint across weighted graph edges. However, the classical algorithm only works if the initial distance estimates are all overestimates, i.e. with t_0 the initial time, for all i

$$\hat{d}_i(t_0) > d_i. \tag{8}$$

Thus by definition the classical Bellman-Ford algorithm is not globally uniformly asymptotically stable. In an interconnected environment, the input to the algorithm may be graph topology or the source set, which may change over time: At a given instant the current estimate may well fall below the true current distance. Classical Bellman-Ford cannot survive such perturbations, prompting the adaptive variant.

This algorithm is based closely on the classical Bellman-Ford algorithm, but unlike that algorithm, computes distances to the nearest member of a set of source nodes rather than just a single node. Moreover, we wish to support the case where the set of sources and/or the graph may change. Thus ABF is an *adaptive algorithm* that a) sets the distance estimate of every source node to zero, and b) for all other nodes, rather than starting at infinity and always decreasing, recomputes distance estimates periodically, ignoring the current estimate at a node and using only the minimum of the triangle inequality constraints of its neighbors.

In particular, suppose $\hat{d}_i(t)$ is the current estimated distance of i from the source set. Then the algorithm is:

$$\hat{d}_i(t+1) = \begin{cases} \min_{j \in \mathcal{N}(i)} \left\{ \hat{d}_j(t) + e_{ij} \right\} & i \notin S \\ 0 & i \in S \end{cases}, \ \forall t \ge t_0.$$
(9)

Observe that (9) respects the information structure imposed in (A) and (B) of Section III-B, and seeks to emulate (4) treating the available distance estimates as their true values.

The behavior of this algorithm reduces to something very close to classical Bellman-Ford in the case where there is precisely one source node and neither the graph nor the source ever change. We have previously presented an empirical analysis and experimental results on the dynamics of this algorithm in [30]; in this paper we give a formal analysis.

Analogous to true constraining nodes defined in Definition 1, we have the following definition:

Definition 3. A minimizing j in the first equation of (9) will be called a current constraining node of i at time t.

A current constraining node of i constrains the distance estimate of i. Note while true constraining nodes are fixed by the graph topology, current constraining nodes may change from iteration to iteration. Figure 2 illustrates the execution of the algorithm. The numbers on the edges are edge lengths; those on the nodes their estimated distances from the source labeled red. Observe the true constraining node of the second node from the left is the third node, though its current constraining node at t=1 is the first node from the left.

We will later show that $\hat{d}_i = d_i$ for all $i \in V$ constitutes the only stationary point of (9), thus at least partially validating ABF as a distance finding tool.

V. A LYAPUNOV FUNCTION

The goal of this section is to postulate a discrete time Lyapunov (or energy) function and demonstrate that it is nondecreasing. On the face of it, distance estimation errors,

$$\Delta_i(t) = \hat{d}_i(t) - d_i,\tag{10}$$

appear to form a natural measure of the algorithm's performance. However, as seen in Figure 2, $\Delta_i(t)$ may well increase in magnitude for individual nodes. This stems from the nature of ABF given in (9): $\hat{d}_i(t+1)$ does not explicitly depend on $\hat{d}_i(t)$. Instead, as will be evident in the sequel, depending on its sign, $\Delta_i(t+1)$ bears a natural comparison with $\Delta_j(t)$ where j is among one of two distinguished neighbors of i: either a true constraining node of i or a current constraining node at time t. This subtlety constitutes a key distinction between the analysis here and typical discrete time Lyapunov analyses.

One requires a more global point of comparison from one iteration to the next. As empirically studied in [30], the greatest overestimate of the error $\Delta^+(t)$ and the least underestimate of the error $\Delta^-(t)$ below collectively provide such a comparison:

$$\Delta^{+}(t) = \max \left[0, \max_{i} \Delta_{i}(t) \right]$$
 (11)

$$\Delta^{-}(t) = \max \left[0, -\min_{i} \Delta_{i}(t) \right]. \tag{12}$$

Should, as empirically suggested by [30], each of these be non-increasing then their sum forms a natural Lyapunov function:

$$L(t) = \Delta^{+}(t) + \Delta^{-}(t). \tag{13}$$

Indeed in Figure 2, while individual Δ_i may increase in magnitude, Δ^+ and Δ^- never do. The rest of this section verifies the validity of (13) as a Lyapunov function.

We begin by noting that this function clearly meets the nonnegativity requirement for a Lyapunov function as

$$L(t) \ge 0,\tag{14}$$

with equality holding iff for all i, $\Delta_i(t) = 0$. As a matter of fact, it can readily be verified that L(t) acts as a valid norm for a vector of the distance estimation errors.

As a preface to proving that L(t) is also non-increasing, we define $\mathcal{K}_+(t)$ as a set comprising all nodes whose error equals $\Delta^+(t)$. More precisely:

$$\mathcal{K}_{+}(t) = \left\{ i \in V | \Delta_{i}(t) = \Delta^{+}(t) \right\}. \tag{15}$$

Similarly,

$$\mathcal{K}_{-}(t) = \{ i \in V | \Delta_i(t) = -\Delta^{-}(t) \}.$$
 (16)

If $\Delta^+(t) \neq 0$ then each member of $\mathcal{K}_+(t)$ has the largest estimation error. This is however, not necessarily true if $\Delta^+(t) = 0$, as then $\Delta_i(t) \leq 0$, for all $i \in V$. If $\Delta^-(t) \neq 0$ then its members $\mathcal{K}_-(t)$ have the most negative estimation error. We now prove the non-increasing property of $\Delta^+(t)$.

Lemma 2. Consider (9) under Assumption 1. Then with Δ^+ defined in (11), for all t,

$$\Delta^{+}(t+1) \le \Delta^{+}(t). \tag{17}$$

Further, consider $K_+(t)$ in (15), and suppose $\Delta_+(t) > 0$. Then equality in (17) holds iff there exists $j \in K_+(t)$ that is both a current and a true constraining node (see definitions 1 and 3) of a member of $K_+(t+1)$.

Proof. As $\Delta^+(\cdot) \geq 0$, (17) holds if $\Delta^+(t+1) = 0$. Assume $\Delta^+(t+1) > 0$ throughout the proof. Consider $l \in \mathcal{K}_+(t+1)$ and any neighbor $j \in \mathcal{N}(l)$ that is a true constraining node of l, i.e. from (4) and Definition 1,

$$d_l = d_j + e_{lj} (18)$$

Then from (15) we find that (17) is proved through:

$$\Delta^{+}(t+1) = \Delta_{l}(t+1)
= \hat{d}_{l}(t+1) - d_{l}
\leq \hat{d}_{j}(t) + e_{lj} - d_{l}
= \hat{d}_{j}(t) + e_{lj} - e_{lj} - d_{j}
= \Delta_{j}(t)
\leq \Delta^{+}(t),$$
(19)

where (19) comes from (9), and (20) from (11).

Suppose there is a $j \in \mathcal{K}_+(t)$ that is both a constraining and a current constraining node of $l \in \mathcal{K}_+(t+1)$. From Definition 1, (18) holds; (19) is an equality as $j \in \mathcal{C}(l)$; and (20) is an equality as $j \in \mathcal{K}_+(t)$. Thus equality in (17) holds.

Now suppose equality in (17) holds. Then in the sequence of inequalities above one can choose $l \in \mathcal{K}_+(t+1)$ and $j \in \mathcal{C}(l)$, i.e. a j that obeys (18), for which both (19) and (20) are equalities. From (15), (20) implies that $j \in \mathcal{K}_+(t)$. As (18) implies that $j \in \mathcal{C}(l)$ and $l \in \mathcal{K}_+(t+1)$ this must mean a node in $\mathcal{K}_+(t)$ is a true constraining node of $l \in \mathcal{K}_+(t+1)$. As equality also holds in (19), this j is also a current constraining node of l. The result follows.

The biggest takeaways from this lemma are that $\Delta^+(t)$ cannot increase, and that a strict decrease eventuates from iteration t to t+1 unless a node with the largest overestimate at time t is both a current and a true constraining node of a node that inherits the largest overestimate at time t+1. Thus the condition for lack of strict decrease for Δ^+ is very stringent. We next address Δ^- .

There are more subtle properties of (9) exposed by the proof. Referring back to the italicized statement at the beginning of this section, the correct comparison point of an overestimate $\Delta_l(t+1)$ is not $\Delta_l(t)$ but in fact the overestimate at t of one of its true constraining nodes j. In particular with $j \in C(l)$,

$$\Delta_l(t+1) \le \Delta_j(t),\tag{21}$$

i.e, this new overestimate cannot exceed the overestimates of the true constraining nodes of l.

Lemma 3. Consider (9) under Assumption 1. Then with Δ^- defined in (12), for all t,

$$\Delta^{-}(t+1) < \Delta^{-}(t). \tag{22}$$

With $K_{-}(t)$ as in (16), unless $\Delta^{-}(t) = 0$, equality in (22) holds iff there exists $j \in K_{-}(t)$ that is both a true and current constraining node of a member of $K_{-}(t+1)$.

Proof. As $\Delta^-(t+1)$ is nonnegative (22) holds if $\Delta^-(t+1) = 0$. Thus assume $\Delta^-(t+1) > 0$. Consider any $l \in \mathcal{K}_-(t+1)$. Because of (9) there is a $j \in \mathcal{N}(l)$, such that

$$\hat{d}_l(t+1) = \hat{d}_j(t) + e_{lj}$$
 (23)

Further, $\Delta^-(t)$ cannot increase as

$$\Delta^{-}(t+1) = -\Delta_{l}(t+1)
= d_{l} - \hat{d}_{l}(t+1)
= d_{l} - \hat{d}_{j}(t) - e_{lj}
\leq e_{lj} + d_{j} - \hat{d}_{j}(t) - e_{lj}$$
(24)
$$= -\Delta_{j}(t)$$
(25)

$$\leq \Delta^{-}(t)$$
 (26)

where (24) comes from (4) and (26) follows from (12).

Suppose equality in (22) holds. Then for some $l \in \mathcal{K}_{-}(t+1)$ and a j satisfying (23), both (24) and (26) are equalities. From Definition 3, j is a current constraining node of l. From Definition 1 equality in (24) implies that j is also a true constraining node of l. From (16), equality in (26) implies

that $j \in \mathcal{K}_{-}(t)$. Thus, as $l \in \mathcal{K}_{-}(t+1)$, $\Delta^{-}(t+1) = \Delta^{-}(t)$ only if there exists $j \in \mathcal{K}_{-}(t)$ that is a true constraining node of an $l \in \mathcal{K}_{-}(t+1)$.

On the other hand suppose for some $l \in \mathcal{K}_{-}(t+1)$, there is a $j \in \mathcal{K}_{-}(t)$ that is both a current and true constraining node of l. Then from Definition 3, (23) holds. Further $j \in \mathcal{K}_{-}(t)$ implies equality holds in (26). As j is a true constraining node of l equality also holds in (24), proving equality in (22).

Again, the biggest takeaways from this lemma are that $\Delta^-(t)$ cannot increase and that a strict decrease eventuates from iteration t to t+1 unless a node at t with the most negative error is both a true and current constraining node of a node that inherits the largest underestimate at time t+1.

Further the proof reveals some subtle properties of (9) that stand in contrast to the case of overestimates addressed by Lemma 2. Referring back to the italicized statement at the beginning of this section, the correct comparison point of an underestimate $\Delta_l(t+1)$ is not $\Delta_l(t)$ but the underestimate at time t of the current constraining nodes in (9). Specifically unlike over estimates, for an underestimate, (21) holds but with j a constraining rather than a true constraining node. Mathematically, that is because the operational triangular inequality used in the lemma proofs is represented by (9) for underestimates and by (4) for overestimates. As will be evident in the next section this has significant consequences to the relative convergence rates of overestimates and underestimates. In particular, while $\Delta^+(\cdot)$, declines rapidly, $\Delta^-(\cdot)$ need not.

Lemma 2 and Lemma 3 together show that for all t,

$$L(t+1) \le L(t),\tag{27}$$

validating the fact that L(t) is indeed a Lyapunov function. Moreover, equality in (27) holds under stringent conditions. In fact as shown below in Theorem 1, a strict decline in L(t) must occur every $\mathcal{D}(G)$ iterations, where $\mathcal{D}(G)$ is the effective diameter in Definition 2. Theorem 1 also provides the aesthetically appealing result that $\hat{d}_i = d_i$, for all $i \in V$ is the only stationary point of ABF.

Theorem 1. Under the conditions of Lemma 2 and Lemma 3, with $\mathcal{D}(G)$ as in Definition 2, L(t) as in (13), unless L(t) = 0,

$$L(t + \mathcal{D}(G) - 1) < L(t), \ \forall t \ge t_0.$$
 (28)

Further $\hat{d}_i = d_i$, $\forall i \in V$ is the only stationary point of (9).

Proof. Suppose L(t)>0. From Lemma 2 and Lemma 3, (27) holds. Suppose now for some t and T and all $s\in\{1,\cdots,T-1\}$, L(t+s)=L(t). Then from Lemma 2 and Lemma 3 there exists a sequence of nodes n_1,\cdots,n_T , such that n_i is a true constraining node of n_{i+1} . From Lemma 1 this means $T\leq \mathcal{D}(G)$. In fact $T\leq \mathcal{D}(G)-1$. To establish a contradiction, suppose $T=\mathcal{D}(G)$. Then in the proofs of Lemma 2 and Lemma 3, $j=n_1\in S$. Thus from (20) and (26), $\Delta^+(t)=\Delta^-(t)=L(t)=0$. Thus (28) holds.

Suppose for all $i \in V$, $\hat{d}_i = d_i$. For $i \in S$, and all t, $\hat{d}_i(t) = 0 = d_i$, $\hat{d}_i(t+1) = 0 = d_i$, also holds. Now consider any $i \in V \setminus S$. Then from (4), there holds:

$$\hat{d}_i(t+1) = \min_{j \in \mathcal{N}(i)} \left\{ \hat{d}_j(t) + e_{ij} \right\}$$
$$= \min_{j \in \mathcal{N}(i)} \left\{ d_j + e_{ij} \right\}$$
$$= d_i.$$

Thus indeed $\hat{d}_i = d_i$, $\forall i \in V$ is a stationary point of (9). Now consider any other candidate stationary point $\hat{d}_i = d_i^*$, with

$$d^* = \left[d_1^*, \cdots, d_{|V|}^* \right] \neq \left[d_1, \cdots, d_{|V|} \right].$$
 (29)

Suppose also at some t, and all $i \in V$, $\hat{d}_i = d_i^*$. Then L(t) > 0. From the first part of this theorem, (28) holds and

$$\left[\hat{d}_1(t+\mathcal{D}(G)),\cdots,\hat{d}_{|V|}(t+\mathcal{D}(G))\right]\neq d^*.$$

Thus d^* cannot be a stationary point.

Of course without establishing a uniform bound from below on the extent of decline in (28), we cannot establish global uniform asymptotic stability. The next section does just that.

VI. GLOBAL UNIFORM ASYMPTOTIC STABILITY

We now establish the global uniform asymptotic stability of ABF and *tightly bound its convergence time*. Recall that (13) has two components (the largest overestimate $\Delta^+(t)$ and the largest underestimate $\Delta^-(t)$), that the classical Bellman-Ford algorithm only copes with overestimates as it initializes to ensure (8), and that the motivation behind (9) is to permit underestimates.

It turns out that there is a fundamental disparity between the behaviors of under and overestimates in (9): Overestimates converge rapidly. Underestimates do not. Why this disparity? The key lies in (21). When $\Delta_l(t+1) > 0$ the j in (21) is a true constraining node of l, while if $\Delta_l(t+1) < 0$, it is a current constraining node of l for the algorithm at time t. While true constraining nodes are fixed by the graph, current constraining nodes may change. Moreover, a pair of nodes may constrain each other at alternate instants, and should they share a short edge, their distance estimates rise slowly in tandem by small amounts. Dubbed in [28] as the rising value problem, this can lead to slow convergence.

By contrast, the following theorem shows that the overestimates all vanish to zero in at most $\mathcal{D}(G) - 1$ steps, where $\mathcal{D}(G) - 1$ is the effective diameter defined in Definition 2.

Theorem 2. Under Assumption 1, $\Delta^+(t)$ defined in (11) obeys

$$\Delta^{+}(t) = 0, \ \forall \ t \ge t_0 + \mathcal{D}(G) - 1.$$
 (30)

Further, for every n = |V| > 1 there is a G = (V, E), obeying Assumption 1 and a set of initial conditions such that $\Delta^+(t) > 0$, for all $t < t_0 + \mathcal{D}(G) - 1$,

Proof. As G is connected, every node belongs to a sequence of nodes n_1, n_2, \dots, n_T , such that n_i is the true constraining

node of n_{i+1} and $n_1 \in S$. From Lemma 1, $T \leq \mathcal{D}(G)$. We now assert and prove by induction that,

$$\Delta_{n_i}(t) < 0, \ \forall \ t > i - 1 + t_0, \ \text{and} \ i < T.$$
 (31)

Then the result is proved from (11). As $n_1 \in S$, (31) holds from (9). Now suppose it holds for some $i \in \{0, \dots, T-1\}$. As $n_i \in \mathcal{C}(n_{i+1}) \subset \mathcal{N}(n_{i+1})$, from (9), (4) and the induction hypothesis, for all $t \geq i + 1 + t_0$,

$$\begin{array}{rcl} \hat{d}_{n_{i+1}}(t) & \leq & \hat{d}_{n_i}(t-1) + e_{n_{i+1}n_i} \\ & \leq & d_{n_i} + e_{n_{i+1}n_i} \\ & = & d_{n_{i+1}}. \end{array}$$

Thus (31) and hence (30) is true.

For the second part of the theorem, we first observe that if j is both the true and current constraining node of i. Then

$$\Delta_{i}(t+1) = \hat{d}_{i}(t+1) - d_{i}
= \hat{d}_{i}(t+1) - d_{j} - e_{ij}
= \hat{d}_{j}(t) + e_{ij} - d_{j} - e_{ij}
= \Delta_{j}(t).$$
(32)

In the subgraph, \hat{G} comprising the nodes S and $1, \cdots, n$ in Figure 4, $\mathcal{D}(\hat{G}) = n+1$. Denote S=0. Suppose for all $i \in \{1, \cdots, n\}, \ 0 < \Delta_i(t_0) < e$. The result is proved if we show that for all $0 \le j \le t < i \in \{1, \cdots, n\}$

$$0 < \Delta_i(t + t_0) < e, \ \Delta_j(t + t_0) = 0, \tag{33}$$

and i-1 is the current constraining node of i. Observe for all $k \in \{1, \cdots, n\}$, $d_k = ke, k-1$ is the true constraining node of k, and for all $k \in \{1, \cdots, n-1\}$ (33) implies that

$$\begin{array}{rcl} \hat{d}_{k+1}(t+t_0) + e_{k+1,k} & = & (k+1)e + \Delta_{k+1}(t+t_0) + e \\ & \geq & (k+2)e \\ & = & d_{k-1} + 3e \\ & > & d_{k-1} + \Delta_{k-1}(t+t_0) + 2e \\ & > & \hat{d}_{k-1}(t+t_0) + e_{k-1,k}, \end{array}$$

i.e. k-1 is the current constraining node of k. As n-1 is the only neighbor of n, this is also true for k=n. Use induction to prove (33), which holds for t=0. Suppose it holds for $1 \leq t < m < n$. As $0 < \Delta_i(m+t_0) < e$ for all $i \in \{m+1, \cdots, n\}$, from (32), the inequality in (33) holds. As $\Delta_{m-1}(t+m-1)=0$, and m-1 is the current constraining node of m, so does the equality.

The proof of (30) implicitly expands on (21). Specifically, one can view n_i in the proof as being a node that is effectively n_i-1 hops away from the source set. At the i-th instant suppose l in (21) is a node that is (i+1) effective hops away. Then for this node the actual error is forced to be nonpositive, as j its true constraining node, one effective hop closer to the source set, acquires a nonpositive estimation error an iteration earlier. As we show below underestimates lack this property.

Depending on the initial distance estimates and the graph topology, $\Delta^+(t)$ may converge to zero in fewer than $\mathcal{D}(G)-1$ rounds. This may happen for example, if n_i acquires a negative

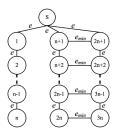


Fig. 4. Illustration of the tightness of convergence time. The subgraph comprising the nodes S and $\{1,\cdots,n\}$ is used in the proof of Theorem 2. The entire graph is used in the proof of Theorem 5.

error after i-2 iterations, then convergence of $\Delta^+(t)$ to zero occurs an iteration sooner. Nonetheless the upper bound on the time for Δ^+ to converge is tight in the sense that for all $|V| \geq 2$, there is a graph and an initialization where convergence cannot occur before $\mathcal{D}(G)-1$ iterations. The proposition below shows more: If no initial estimate is an underestimate, then convergence occurs in at most $\mathcal{D}(G)-1$ iterations. The proof follows from Lemma 3 and Theorem 2.

Proposition 1. Under Assumption 1, suppose $\Delta^-(0) = 0$. Then the Lyapunov function L(t) defined in (13)

$$L(t) = 0, \ \forall t \ge \mathcal{D}(G) - 1,$$

where $\mathcal{D}(G)$ is defined in Definition 2.

This underscores why the classical Bellman-Ford algorithm requires that $\Delta^-(0)=0$. The next theorem quantifies the rising value problem that causes a slower decline in $\Delta^-(t)$. Indeed, the classical Bellman-Ford algorithm, which assumes (8) will also yield the same convergence time. But of course it cannot cope with initial underestimates.

Theorem 3. Under the conditions of Lemma 3, consider a pair of nodes i and j such that $\Delta_i(t_0) < 0$ and

$$j = \arg\min_{j \in \mathcal{N}(i)} e_{ij}.$$

Then $\Delta_i(T) \geq 0$ implies

$$T \ge t_0 - \frac{\Delta_i(t_0)}{e_{ii}}.$$

Proof. From (9) for any t

$$\hat{d}_j(t+1) \leq \hat{d}_i(t) + e_{ij}. \tag{34}$$

Likewise, as $j \in \mathcal{N}(i)$, using (34) the result follows as,

$$\Delta_{i}(t+2) = \hat{d}_{i}(t+2) - d_{i}
\leq \hat{d}_{j}(t+1) + e_{ij} - d_{i}
\leq \hat{d}_{i}(t) + 2e_{ij} - d_{i}
= \Delta_{i}(t) + 2e_{ij}.$$

Thus the rising value problem may occur even if a pair of nodes with underestimated distance estimates do not constrain each other. Rather the rise is limited by the smallest edge length impinging on a node with the largest underestimate.

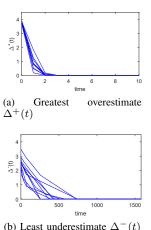


Fig. 5. Trace of (a) greatest overestimate $\Delta^+(t)$ and (b) least underestimate $\Delta^-(t)$ for 10 runs of 500 nodes randomly distributed in a 4x1 sq. km area, showing that overestimates correct much faster than underestimates.

This behavior is shown in Figure 5, where 500 nodes, including a solitary source, are uniformly distributed in a 4x1 kM field. Each has a communication range of 0.25 kilometers, i.e, the average size of $\mathcal{N}(i)$ is 20. The initial distance estimates of non-source nodes are chosen randomly in $\mathcal{U}(0, 4.12)$ kilometers. Figure 5 shows the results of 10 simulations, each run synchronously for 2000 seconds with 1 second per round. The results are consistent with our analysis: $\Delta^+(t)$ decreases rapidly to zero within at most 4 rounds, even though $\mathcal{D}(G)=14$. On the other hand, $\Delta^-(t)$, limited by close pairs of nodes, has much slower convergence. Observe that the classical Bellman-Ford algorithm would have yielded the same type of trace as in Figure 5(a), but cannot deal with the violation of (8).

The next lemma helps lower bound the decline in $\Delta^-(t)$.

Lemma 4. Under the conditions of lemmas 2 and 3 define,

$$S^{-}(t) = \{ i \in V | \Delta_i(t) < 0 \}, \tag{35}$$

and

$$\hat{d}_{\min}(t) = \min_{i \in \mathcal{S} (t)} \{\hat{d}_i(t)\}. \tag{36}$$

Then with e_{\min} defined in (1), the following holds unless $S^-(t+1)$ is empty:

$$\hat{d}_{\min}(t+1) \ge \hat{d}_{\min}(t) + e_{\min}, \ \forall t \ge t_0.$$
 (37)

Proof. Suppose $\mathcal{S}^-(t+1)$ is not empty. Then from Lemma 3, and (16), $\mathcal{S}^-(t)$ cannot be empty. Consider any $i \in \mathcal{S}^-(t+1)$ and suppose j is its current constraining node at t. Then we assert that $j \in \mathcal{S}^-(t)$. Indeed assume $j \notin \mathcal{S}^-(t)$. Thus $\hat{d}_j(t) \geq d_j$. As $j \in \mathcal{N}(i)$, from Definition 3 and (4),

$$\hat{d}_i(t+1) = \hat{d}_j(t) + e_{ij}
\geq d_j + e_{ij}
> d_i.$$

Thus $i \notin S^-(t+1)$, establishing a contradiction. Hence $j \in S^-(t)$. Then from (35) and (36), (37) holds as $\forall i \in S^-(t+1)$,

$$\hat{d}_i(t+1) = \hat{d}_j(t) + e_{ij}
\geq \hat{d}_{\min}(t) + e_{min}.$$

We now prove that the decline in (28) is uniformly lower bounded, proving global uniform asymptotic stability.

Theorem 4. Under conditions of Theorem 1, there exists $\alpha > 0$, dependent on the initial conditions but not t_0 , such that

$$0 \le L(t + \mathcal{D}(G) - 1) \le \max[L(t) - \alpha, 0], \ \forall \ t \ge t_0.$$
 (38)

Further the unique stationary point of (9), $\Delta_i = 0$ for all $i \in V$, is globally uniformly asymptotically stable.

Proof. As the postulated α in (38) is fixed by the initial conditions and is independent of the initial time t_0 , (38) proves global uniform asymptotic stability. From Theorem 2, the set of $\Delta^+(t)$ is a finite countable set that includes zero. Similarly, because of Lemma 3 and (37) of Lemma 4, so is the set of $\Delta^-(t)$, as \hat{d}_{\min} in (36), must equal the true distance it estimates in a finite time. Thus \mathcal{L} the set of L(t), for all $t \geq t_0$ is a finite countable set. Then α , the smallest difference between the elements of this set, is positive unless $L(t_0) = 0$, and while dependent on the initial conditions, is independent of t_0 . From Theorem 1 a decrease in L(t) occurs every $\mathcal{D}(G)$ iterations. As $L(t) \in \mathcal{L}$, any change must be by at least α .

We now tightly bound the time to convergence.

Theorem 5. Consider (9) under the conditions of Theorem 4, $\mathcal{D}(G)$ as in Definition 2 and e_{\min} in (1). Define

$$d_{\max}(G) = \max_{i \in V} \{d_i\},\tag{39}$$

for G = (V, E). Then L(t) = 0, $\forall t \ge t_0 + T$, where,

$$T = \max \left\{ \mathcal{D}(G) - 1, \left\lceil \frac{d_{\max} - \hat{d}_{\min}(t_0)}{e_{\min}} \right\rceil \right\}. \quad (40)$$

Further for every n = |V| > 3, there exists a G satisfying Assumption 1 for which L(t) > 0 for all t < T.

Proof. From Theorem 2, $\Delta^+(t_0 + \mathcal{D}(G) - 1) = 0$, accounting for $\mathcal{D}(G)$ in (40). From Lemma 4, and any $i \in \mathcal{S}^-(t)$ in (35), one obtains for any $t \geq t_0$,

$$\begin{array}{lcl} \Delta^-(t) & \leq & d_i - \hat{d}_i(t) \\ & \leq & d_{\max} - \hat{d}_{\min}(t) \\ & \leq & d_{\max} - \hat{d}_{\min}(t_0) - (t - t_0) e_{\min}. \end{array}$$

Thus $\Delta^-(t) = 0$, whenever

$$t - t_0 \ge T_- = \left[\frac{d_{\max}(G) - \hat{d}_{\min}(t_0)}{e_{\min}} \right].$$
 (41)

To prove that convergence time can be as much as T, consider the graph in Figure 4 with $e_{\min} < e$. Assume for all $i \in \{1, \cdots, n\}$, $\hat{d}_i(t_0) > d_i$, and $\hat{d}_{n+i}(t_0) = \hat{d}_{2n+i}(t_0) = 0$. Then for all $i \in \{1, \cdots, n\}$, and all t, $\Delta_i(t) \geq 0$, and $\Delta_{n+i}(t) = \Delta_{2n+i}(t) \leq 0$. From the proof of Theorem 2, it takes exactly $\mathcal{D}(G) - 1$ iterations for Δ^+ to converge, and as shown in the appendix, exactly T_- iterations for Δ^- to converge.

Thus the bound on convergence time is tight, though the worst case nature of the analysis also makes it conservative.

VII. ROBUSTNESS UNDER PERTURBATIONS

We now turn to the robustness of ABF to possibly *persistent* perturbations in non-source nodes with the goal of demonstrating ultimate boundedness (per the definition in [1]) of distance estimates around nominal distance values. We consider the behavior of ABF in a framework with two physical interpretations. 1) Nodes experience bounded, potentially perpetual motion around nominal locations. In aggregate computing this captures an incremental version of the common scenario of mobile computing devices or of imprecise localization. 2) A node receives noisy distance estimates of its neighbors.

Consider first the case where each *non-source* node moves around a nominal position, i.e., edge lengths change from their nominal values e_{ij} as

$$\bar{e}_{ij}(t) = e_{ij} + \epsilon_{ij}(t). \tag{42}$$

Mobility is assumed to be both bounded and small, i.e., there exists an ϵ such that with e_{\min} defined in (1),

$$|\epsilon_{ij}(t)| < \epsilon < e_{\min}.$$
 (43)

This ensures that no edge length is ever negative. Based on this assumption of bounded mobility, we also assume that the set of neighbors $\mathcal{N}(i)$ of each node i does not change.

This also accommodates the setting where noisy estimates of $\hat{d}_i(t)$ available to its neighbors, with $\epsilon_{ij}(t)$ modeling the noise. Unlike the setting of mobility in this case we cannot assume that the noise is symmetric, and permit

$$\bar{e}_{ij}(t) \neq \bar{e}_{ji}(t). \tag{44}$$

In particular $\bar{e}_{ij}(t)$ is the noisy edge length seen by node i as opposed to node j. Thus (9) must be interpreted as:

$$\hat{d}_i(t+1) = \begin{cases} \min_{j \in \mathcal{N}(i)} \left\{ \hat{d}_j(t) + \bar{e}_{ij}(t) \right\} & i \notin S \\ 0 & i \in S \end{cases}$$
 (45)

As

$$\hat{d}_{j}(t) + \bar{e}_{ij}(t) = \hat{d}_{j}(t) + e_{ij} + \epsilon_{ij}(t)$$

= $(\hat{d}_{i}(t) + \epsilon_{ij}(t)) + e_{ij}$,

this captures the execution of ABF with noisy measurements of $\hat{d}_j(t)$. The problem formulation is otherwise unchanged. The nominal graph is still G=(V,E) of Section III-A, undirected in that $i\in\mathcal{N}(i)$ implies $j\in\mathcal{N}(j)$ and $e_{ij}=e_{ji}$, and the goal is to study the perturbations of $\hat{d}_i(t)$ from the nominal distances d_i . We now define a shrunken version of G, which represents the graph with the shortest links permitted by our perturbation model. We will show that the distance estimates provided by ABF when applied to this graph lower bounds all $\hat{d}_i(t)$. This will help provide the lower ultimate bounds and the time to attain them.

Definition 4. Given G = (V, E), the undirected graph $G^- = (V, E^-)$, has the property that edge $(i, j) \in E^-$ iff $(i, j) \in E$. The edge length e^-_{ij} between nodes i and j obeys

$$e_{ij}^{-} = e_{ij} - \epsilon, \tag{46}$$

with ϵ defined in (43). Further G^- has the same source set S as G, each i has the same set of neighbors as in G, and

 D_i is the shortest distance between i and the nearest source node, i.e. plays the role of d_i in G.

Our goal is to prove the ultimate boundedness of the $\hat{d}_i(t)$ provided by (45) by examining their difference with the nominal distances d_i , through L(t), which retains its definition in (13). Note $\Delta_i(t) = \hat{d}_i(t) - d_i$ where d_i are the distances in G. We summarize the underlying assumptions.

Assumption 2. Both the graphs G and G^- defined in Definition 4 obey Assumption 1. The set of neighbors of node i, is time invariant for all $i \in V$ as is the source set. None of the source nodes move. The edge length of each pair of nodes i and j is given by (42) under (43) and (1). Although (44) applies, $e_{ij} = e_{ji}$ still holds. Also assume that $t_0 = 0$.

Note $t_0=0$ is without loss of generality because of the fact that its uniform asymptotic stability, guarantees that the behavior of (9) is independent of t_0 . Despite the topological similarity of the graphs G^- and G, their respective effective diameters $\mathcal{D}(G^-)$ and $\mathcal{D}(G)$, defined in Definition 2, may differ. This is so as the true constraining nodes in the graphs may be different, as is illustrated by the two graphs in Figure 3. In G the true constraining node of D is E while in G^- , it is C. Further $\mathcal{D}(G)=4<\mathcal{D}(G^-)=5$. Lemma 7 shows that

$$\mathcal{D}(G^{-}) \ge \mathcal{D}(G). \tag{47}$$

The next lemma proves the ultimate boundedness of $\Delta^+(t)$.

Lemma 5. Consider (45), under Assumption 2. Then $\Delta^+(t) \leq (\mathcal{D}(G) - 1)\epsilon$ for all $t \geq \mathcal{D}(G) - 1$, where $\mathcal{D}(G)$ is as in Definition 2 and $\Delta^+(t)$ is as in (11).

Proof. Consider the sequence $n_1, n_2, ..., n_T$ in the proof of Theorem 2 where the true constraining nodes are for the graph G. As $T < \mathcal{D}(G)$, the result holds if

$$\hat{d}_{n_i}(t) \le d_{n_i} + (i-1)\epsilon, \ \forall \ i \in \{1, \dots, T\} \ \text{and} \ t \ge i-1.$$
 (48)

We prove (48) by induction. It is true for i=1 from (45) as $n_1 \in S$. Thus suppose it holds for some $i \in \{1, \dots, T-1\}$. Then from (45), (42), (43), and the definition of a true constraining node we have for all $t \geq i-1$

$$\begin{split} \hat{d}_{n_{i+1}}(t+1) & \leq & \hat{d}_{n_i}(t) + \bar{e}_{n_{i+1}n_i}(t) \\ & \leq & d_{n_i} + (i-1)\epsilon + e_{n_{i+1}n_i} + \epsilon_{n_{i+1}n_i}(t) \\ & = & d_{n_{i+1}} + (i-1)\epsilon + \epsilon_{n_{i+1}n_i}(t) \\ & \leq & d_{n_{i+1}} + (i-1)\epsilon + \epsilon \\ & = & d_{n_{i+1}} + i\epsilon. \end{split}$$

Thus (48) and hence the result follows.

To address $\Delta^-(t)$ we take an approach like the *comparison* principle [1], by providing a lemma, proved in the appendix, that establishes a connection between distance estimates in G and those in its shrunken version, G^- , defined in Definition 4.

Lemma 6. Suppose Assumption 2 holds. Consider

$$\hat{D}_{i}(t+1) = \begin{cases} \min_{j \in \mathcal{N}(i)} \left\{ \hat{D}_{j}(t) + e_{ij} - \epsilon \right\} & i \notin S \\ 0 & i \in S \end{cases}, (49)$$

and (45). Suppose for all $i \in V$, $\hat{D}_i(0) = \hat{d}_i(0)$. Then $\hat{d}_i(t) \ge \hat{D}_i(t)$, $\forall t > 0$ and for all $i \in V$.

Thus the estimates offered by (45) are uniformly lower bounded by the estimates, $\hat{D}_i(t)$, from the identically initialized Adaptive Bellman-Ford algorithm applied to the graph G^- . As G^- satisfies the same assumptions as G, and is perturbation free, the greatest underestimation error it offers converges to zero. To establish an ultimate bound on Δ^- it thus suffices to relate distances D_i in G^- , to distances d_i in G. The next lemma, proved in the appendix, does just that.

Lemma 7. Suppose Assumption 2 holds. Then for all $i \in V$,

$$d_i \le D_i + (\mathcal{D}(G^-) - 1)\epsilon,\tag{50}$$

where G^- is in Definition 4 and $\mathcal{D}(.)$ is in Definition 2. Further, (47) holds.

We now prove the ultimate boundedness of Δ^- .

Lemma 8. Consider (45) under Assumption 2. Then with $\mathcal{D}(G^-)$, e_{\min} , $d_{\max}(G^-)$ and \hat{d}_{\min} defined in Definition 2, (1), Theorem 5 and Lemma 4 respectively, for all

$$t \ge T = \frac{d_{\max}(G^-) - \hat{d}_{\min}(0)}{e_{\min} - \epsilon},\tag{51}$$

$$\Delta^{-}(t) \le (\mathcal{D}(G^{-}) - 1)\epsilon. \tag{52}$$

Proof. Consider the algorithm in (49) with the initialization in Lemma 6. As G^- satisfies Assumption 1, $\hat{D}_i(t)$ converges to D_i (see Definition 4) in a finite time T_1 . Thus from Lemma 6, for all $t \geq T_1$ and $i \in V$, $\hat{d}_i(t) \geq D_i$. Thus the upper bound in (52) follows from Lemma 7 and

$$-\Delta_{i}(t) \leq d_{i} - D_{i}$$

$$\leq D_{i} + (\mathcal{D}(G^{-}) - 1)\epsilon - D_{i}$$

$$= (\mathcal{D}(G^{-}) - 1)\epsilon,$$

Time to attain (52) is that for the greatest underestimate in (49) to go to zero. This is at most T in (51) from the following: (i) Theorem 5. (ii) The minimum initial estimate in (49) is $\hat{d}_{\min}(0)$. (iii) The shortest link in G^- is at least $e_{\min} - \epsilon$.

Note $d_{max}(G^-)$ can range from $d_{max}(G) - \epsilon$, e.g. when the node with the largest distance has a source as its true constraining node, to $d_{max}(G) - \mathcal{D}(G^-)\epsilon$.

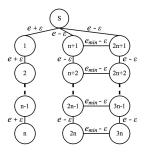


Fig. 6. Illustration of the tightness of convergence time with perturbations.

Theorem 6. Consider (45), the conditions of Lemma 8 and Lemma 5. With T as in (54), L(t) in (13) obeys

$$L(t) \le (\mathcal{D}(G) + \mathcal{D}(G^{-}) - 2) \epsilon, \ \forall t \ge T.$$
 (53)

$$T = \max \left\{ \mathcal{D}(G) - 1, \frac{d_{\max}(G^{-}) - \hat{d}_{\min}(0)}{e_{\min} - \epsilon} \right\}. \tag{54}$$

Further, for every n = |V| > 3, there exists a graph and perturbations conforming to Assumption 2 for which the bound in (53) is attained in precisely T iterations.

Proof. Ultimate boundedness follows from Lemma 8 and Lemma 5. Consider a perturbed version of the nominal graph G in Figure 4 given in Figure 6. The perturbation obeys Assumption 2 and the perturbed graph itself obeys Assumption 1. Since the perturbed graph has fixed edges its distance estimates will converge to the correct values. In particular \hat{d}_n will converge to $d_n + n\epsilon$ and \hat{d}_{2n} to $d_{2n} - n\epsilon$. It is readily checked that these are in fact the greatest over and underestimates respectively. Thus, as $\mathcal{D}(G) = \mathcal{D}(G^-) = n+1$, (53) is precisely met. Suppose now $\hat{d}_i(0) > i(e+\epsilon)$ for all $i \in \{1, \dots, n\}$, and $\hat{d}_i(0) = 0$ for all $i \in \{n+1, \dots, 3n\}$, then the arguments given in the proof of Theorem 5 establish that convergence of \hat{d}_n and \hat{d}_{2n} occur precisely by the first and second terms on the right side of (54).

Though *tight*, these bounds are conservative.

A. Interpretation and Consolidation

If one views the underlying nonlinear system as one from the inputs $\epsilon_{ij}(t)$ in (42) to the errors $\Delta_i(t)$, then $\mathcal{D}(G)+\mathcal{D}(G^-)-2$ in (53) acts as a gain that relates the ℓ_∞ bound, $(|V|-|S|)\epsilon$ on the vector of inputs to the *ultimate bound* on L(t), which in turn serves as a norm of the vector of $\Delta_i(t)$. There is a subtlety: $\mathcal{D}(G^-)$ does depend on ϵ . However, with modest conservatism, in view of (47), it can be replaced by e_{\min} , leading to a classical ultimate bound that is linear in ϵ .

This in and of itself goes to the heart of issues motivating this line of research: Namely to assess the stability of interconnections of aggregate computing building blocks, possibly with feedback interactions. A classical and effective device for such an assessment is provided by the celebrated *small gain theorem*. The ultimate boundedness established here holds forth the prospect of formulating variations of the classical small gain theorem in the tradition of [22], that have been shown to be useful in demonstrating closed loop stability.

Ultimately, Theorem 6 establishes the aesthetically pleasing result that despite persistent perturbations, estimation errors in ABF settle into a range of values proportional to ϵ . The time it takes to settle down, given in (54) is also of importance in aggregate computing. For example, consider the combination, shown in Figure 1, of a G block represented by (9) and a summarizing C block, with the goal of determining the total resource load of distributed services on computing devices in the network. Significantly over-counting or under-counting this number may lead to unacceptable consequences. This value can be computed with a C algorithm, which determines a

spanning tree and then iteratively sums the load at descendants of a source node in the spanning tree. The spanning tree, in turn, is determined adaptively and summations over descendants iteratively computed using the current distance estimates offered by a G block. As shown in [26], the transient phase of ABF can result in over or under counting and there is virtue in waiting out the transient phase. Its characterization thus has a critical role in safe aggregate computing. Unsurprisingly, the transient phase is dominated by the behavior of underestimates, and restricted by e_{\min} .

VIII. SIMULATIONS AND DISCUSSION

We empirically confirm the results presented in the prior sections through simulations under three classes of persistent perturbations: device movement, error in distance measurements, and *periodic changes of source location to induce large perturbations*. Unless otherwise noted, all use 500 nodes, one of which is a source, distributed randomly in a 4x1 km field, communicating over a 0.25 km radius, and run synchronously for 2000 simulated 1-second rounds, with $\hat{d}_i(0) \in \mathcal{U}(0,4.12)$ km. As under perturbations, (8) will not be sustained, the classical Bellman-Ford algorithm cannot cope with them.

A. Device Movement

At each t a node is perturbed from its nominal location by $[r\cos\theta,r\sin\theta]^T$ with $r\sim\mathcal{U}(0,0.5e_{min}),\,e_{min}=0.0048,$ and $\theta\sim\mathcal{U}(0,2\pi).$ Thus $\epsilon=e_{\min}$ in (42). Note (42) and (44) are satisfied with probability 1. The results are in Figure 7.

Lemma 5 and Lemma 8, predict ultimate bounds for Δ^+ and Δ^- of $(\mathcal{D}(G)-1)\epsilon$ and $(\mathcal{D}(G^-)-1)\epsilon$, respectively. In Figure 7(a) $\Delta^+(t)$ goes lower than $(\mathcal{D}(G)-1)\epsilon$ after 3 rounds, $\mathcal{D}(G)=17$. On the other hand, Figure 7(b) shows that $\Delta^-(t)$ is still constrained by the "rising value problem" and needs a much longer time than $\Delta^+(t)$ to drop below $(\mathcal{D}(G^-)-1)\epsilon$.

Figure 7(c) and (d) depict snapshots well beyond the time after the ultimate bounds are attained. Unsurprisingly, due to the worst case nature of the Lyapunov based analysis, which is inherently conservative, there is a significant gap between $\Delta^+(t)$, $\Delta^-(t)$ and their corresponding ultimate upper bounds.

Nevertheless, since these geometric considerations are scale-free, the size of the distance estimate errors should still be linear in the perturbation size. To validate this hypothesis, we set the initial distance estimates of nodes to be the same as their true values, and run the simulation for the same graph with different perturbation amplitudes. Figure 8 shows that, as expected, the mean values of $\Delta^+(t)$ and $\Delta^-(t)$ are roughly proportional to the amplitudes of the perturbations.

B. Measurement Error

Next consider static nodes with asymmetric noise in the estimated e_{ij} . Figure 9 shows the results of simulation in which measurement errors are sampled from a uniform distribution of $\mathcal{U}(-0.5e_{\min}, 0.5e_{\min})$ in each round. In this case $\epsilon = 0.5e_{\min} = 0.5 \times .0049$. The behavior is similar to the case of device motion. When the amplitude of the measurement error is varied, as seen in Figure 10, as expected the mean values of $\Delta^+(t)$ and $\Delta^-(t)$ are roughly proportional.

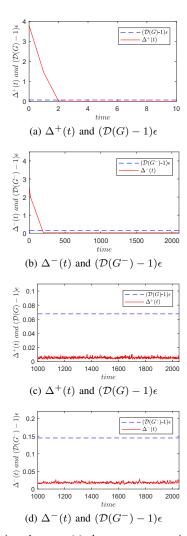


Fig. 7. Comparison between (a) the greatest overestimate $\Delta^+(t)$ and $(\mathcal{D}(G)-1)\epsilon$, and (b) comparison between the least underestimate $\Delta^-(t)$ and $(\mathcal{D}(G^-)-1)\epsilon$ (b), as well as their partial enlarged views (c) and (d). In this example, each node is moving within a circle with a radius of $0.5e_{\min}$.

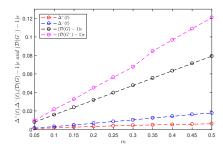


Fig. 8. Mean values of $\Delta^+(t)$ and $\Delta^-(t)$ and their upper bounds with different amplitudes of perturbation. In this example, initial distance estimates are set the same as true values, all nodes are moving within a circle with a radius uniformly distributed in $(0,m\epsilon)$, and m is ranging from 0.05 to 0.5 with an interval 0.05. Here $e_{min}=0.0040$ and $\epsilon=2me_{\min}$

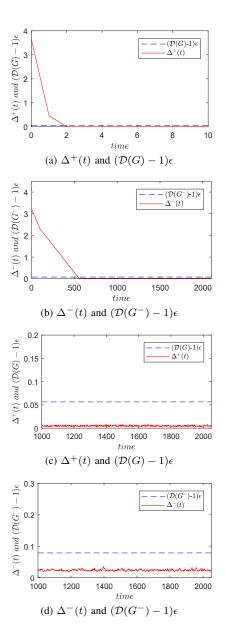


Fig. 9. Comparison between (a) the greatest overestimate $\Delta^+(t)$ and $(\mathcal{D}(G)-1)\epsilon$, and (b) comparison between the least underestimate $\Delta^-(t)$ and $(\mathcal{D}(G^-)-1)\epsilon$ (b), as well as their partial enlarged views (c) and (d). In this example, edge lengths are perturbed by measurement errors in $\mathcal{U}(-0.5e_{\min}, 0.5e_{\min})$.

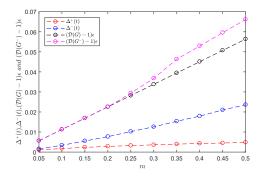


Fig. 10. Mean values of $\Delta^+(t)$ and $\Delta^-(t)$ and their upper bounds with different amplitudes of measurement errors. In this example, initial distance estimates are set the same as true values, and measurement errors obey a uniform distribution $\mathcal{U}(-me_{\min},me_{\min})$ with m ranging from 0.05 to 0.5 with an interval 0.05; $\epsilon=me_{\min},\,e_{min}=0.0049$.

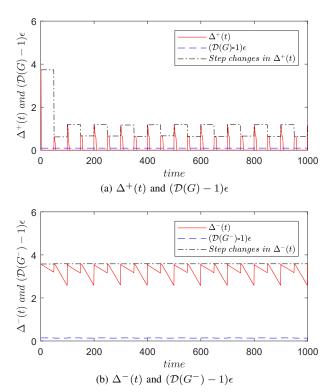


Fig. 11. Comparison between (a) the greatest overestimate $\Delta^+(t)$, $(\mathcal{D}(G)-1)\epsilon$ and the theoretical bound of $\Delta^+(t)$ for unperturbed system, and (b) comparison between the least underestimate $\Delta^-(t)$, $(\mathcal{D}(G^-)-1)\epsilon$ and the theoretical bound of $\Delta^-(t)$ for unperturbed system. In this example, there are two source nodes in the graph, one is at [3.75, 0.5] and the other is at [0.25, 0.5], switching between source nodes every 50 simulated seconds while all nodes are moving within a circle with a radius of $0.5\epsilon_{\min}$.

C. Source Change

The previous two settings involve small perturbations. We now simulate large errors by periodic change of the source set in addition to device movements. The graph periodically alternates between two sources nodes at [3.75, 0.5] and [0.25, 0.5]. The remaining 498 nodes are randomly distributed in a 4x1 km field. At each transition the old source inherits a distance estimate of zero and other nodes acquire a large estimation error. The non-source nodes still move around their nominal values, in a disk with radius in $\mathcal{U}(0, 0.5e_{\min})$, $e_{min} = 0.0044$ and $\epsilon = e_{\min}$. To mimic frequent changes, we alternate the sources every 50 seconds while the simulation runs for 2000 iterations. The results in Figure 11 show that due to the fast convergence rate, $\Delta^+(t)$ will drop below its ultimate upper bound, then bounce back up again when the source changes. The two different "spike" patterns of Δ^+ result from the alternation of the two source nodes. On the other hand, due to its slow convergence rate, $\Delta^-(t)$ in this case never attains its ultimate bound. Yet, even if errors are not small due to the frequent change of source nodes, the algorithm reduces them from the large values they acquire at source transitions. Under less frequent changes $\Delta^-(t)$ converges as well, e.g. in Figure 12, where the sources switch every 1000 seconds.

IX. CONTRIBUTIONS AND FUTURE DIRECTIONS

We have analyzed ABF, by formulating and using a Lyapunov function based on maximal error, to demonstrate global

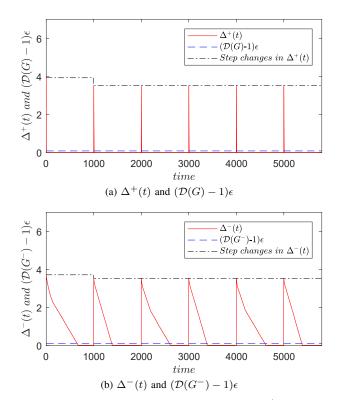


Fig. 12. Comparison between (a) the greatest overestimate $\Delta^+(t)$, $(\mathcal{D}(G)-1)\epsilon$ and the theoretical bound of $\Delta^+(t)$ for unperturbed system, and (b) comparison between the least underestimate $\Delta^-(t)$, $(\mathcal{D}(G^-)-1)\epsilon$ and the theoretical bound of $\Delta^-(t)$ for unperturbed system. In this example, there are two source nodes in the graph, one is at [3.75, 0.5] and the other is at [0.25, 0.5], switching between source nodes every 1000 simulated seconds while all nodes are moving within a circle with a radius of $0.5\epsilon_{\min}$.

uniform asymptotic stability. We have further shown that the algorithm is ultimately bounded in face of persistent mobility of devices or communication noise, have characterized the time to attain the ultimate bound and have validated these results empirically in simulation. The ultimate bounds and the time to attain them are tight.

The results presented in this paper are of immediate and practical use in the design and analysis of any distributed system that uses distance estimates. Our analysis also introduces a novel approach to the construction of Lyapunov functions that exploit dependencies between devices to effectively analyze distributed algorithms with complex interaction patterns. Building on these results, in future work we aim to take a similar approach to expand from Adaptive Bellman-Ford to alternative distance estimation algorithms, such as in [28], [29], and [31], for which complimentary traits have been empirically observed. More broadly, we aim to obtain similar results for other uses of G-block, as well as other aggregate computing building blocks. The ultimate boundedness in particular opens up the prospect of establishing small gain type stability, [22], of G-blocks appearing in feedback loops and possibly through equivalent theorems, [23], like the passivity theorem and its offshoots, [24], [25]. Such results would allow the principled design and analysis of stability and convergence time for a large class of distributed systems, thus offering improvements in safety and efficiency across many areas of application.

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] L. Krick, M.E. Broucke, and B.A. Francis. Stabilization of Infinitesimally Rigid Formations of Multi-Robot Networks. *International Journal of Control*, 82(3):423–439, 2009.
- [6] B. Fidan, S. Dasgupta and B. D. O. Anderson, "Adaptive range-measurement-based target pursuit", *International Journal of Adaptive Control and Signal Processing*, pp. 66-81, 2013.
- [7] M. Cao, C. Yu, A. S. Morse, B. D. O. Anderson and S. Dasgupta. "Generalized controller for directed triangle formations", in *Proceedings of the IFAC World Congress*, 2008.
- [8] 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.
- [9] M. Arcak, "Passivity as a Design Tool for Group Coordination", *IEEE Transactions on Automatic Control*, pp. 1380-1390, 2007.
- [10] T. Hatanaka, N. Chopra, M. Fujita, and M. Spong, Passivity-Based Control and Estimation in Networked Robotics. Communications and Control Engineering. Springer International Publishing, 2015
- [11] J. Beal, D. Pianini, and M. Viroli, "Aggregate programming for the internet of things," *IEEE Computer*, vol. 48, no. 9, pp. 22–30, 2015.
- [12] H. Zimmermann, "OSI reference model—The ISO model of architecture for open systems interconnection." *IEEE Transactions on Communica*tions vol. 28, no. 4, pp 425–432, 1980.
- [13] J. Beal and M. Viroli, "Building blocks for aggregate programming of self-organising applications," in Workshop on Foundations of Complex Adaptive Systems (FOCAS), 2014.
- [14] M. Viroli and F. Damiani, "A calculus of self-stabilising computational fields," in *Coordination* 2014, 2014, pp. 163–178.
- [15] S. Dolev, Self-Stabilization, MIT Press, 2000.
- [16] M Schneider, "Self-stabilization," ACM Computing Surveys, vol. 25, pp. 45–67, 1993.
- [17] 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.
- [18] 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.
- [19] W. Hahn, Stability of Motion, Prentice Hall, 1967.
- [20] R. E. Bellman, "On a routing problem," Quarterly of Applied Mathematics, vol. 16, pp. 87-90, 1958.
- [21] L. R. Ford Jr., "Network flow theory," Tech. Rep. Paper P-923, RAND Corporation, 1956.
- [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] B. D. O. Anderson, "The small-gain theorem, the passivity theorem and their equivalence", *Journal of the Franklin Institute*, pp. 105-115, 1972.
- [24] M. Fu, S. Dasgupta and Y. C. Soh, "Integral quadratic constraint approach vs. multiplier approach", *Automatica*, pp. 281-287, 2005.
- [25] 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.
- [26] Y. Mo, J. Beal, and S. Dasgupta, "Error in Self-Stabilizing Spanning-Tree Estimation of Collective State" in *IEEE International Conference* on Self-Adaptive and Self-Organizing Systems Workshops (SASOS), IEEE, September 2017.
- [27] 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.
- [28] J. Beal, J. Bachrach, D. Vickery, and M. Tobenkin, "Fast self-healing gradients," in ACM Symp. on Applied Computing, 2008.
- [29] J. Beal, "Flexible self-healing gradients." in ACM Symp. on Applied Computing, 2009.

- [30] 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.
- [31] G. Audrito, F. Damiani, and M. Viroli, "Optimally-Self-Healing Distributed Gradient Structures Through Bounded Information Speed." In *International Conference on Coordination Languages and Models*, Springer, pp. 59–77, 2017.
- [32] A. Stentz, "The focussed D* algorithm for real-time replanning", in Proc. International Joint Conference on Artificial Intelligence, 1995.
- [33] S.Koenig, M. Likhachev, and D. Furcy, "D* lite", in *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, 2002.
- [34] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong planning A*.", Artificial Intelligence Journal, pp. 93–146, 2004.
- [35] S. Karaman, and E. Frazzoli, "Sampling-based algorithms for optimal motion planning", Int. Journal of Robotics Research, pp. 846–894, 2011.
- [36] O. Arslan, and P. Tsiotras, "Use of relaxation methods in sampling-based algorithms for optimal motion planning", in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2013.
- [37] M. Otte and E. Frazzoli, "RRTX: Asymptotically optimal single-query sampling-based motion planning with quick replanning", *The Interna*tional Journal of Robotics Research, pp. 797-822, 2016.
- [38] Y. Shang and S. Li, "Distributed Biased Min-Consensus With Applications to Shortest Path Planning", *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 5429-5436, 2017.

APPENDIX

Proof of the tightness of the bound on the time for Δ^- to converge: For simplicity assume $t_0 = 0$. Suppose

$$\left[\frac{e}{e_{min}}\right] = m. \tag{55}$$

Then we assert that for all $i \in \{1, \dots, n\}$,

$$\hat{d}_{n+i}(t) = \hat{d}_{2n+i}(t) = \begin{cases} te_{\min} & i \in \left\{ \left\lfloor \frac{t}{m} \right\rfloor, \dots, n \right\} \\ d_{n+i} & \text{otherwise} \end{cases} . (56)$$

Call each pair 2n+i and n+i partners. When $t\in\{1,\cdots,m-1\}$, as $e>e_{\min}$, and $\hat{d}_{2n+i}(0)=\hat{d}_{n+i}(0)=0$, these nodes constrain each other and their respective distance estimates rise by increments of m. Thus all $\hat{d}_{2n+i}(m)=\hat{d}_{n+i}(m)=e$. Thus $\hat{d}_{2n+1}(m)=\hat{d}_{n+1}(m)=d_{2n+1}(m)=d_{n+1}(m)=e$, and at t=m the nodes n+1 and n+2 are constrained by S, while the remaining are constrained by their partners. Since distance estimates cannot fall in value, for all $t\geq m$ n+1 and n+2 are constrained by S, and $\hat{d}_{2n+1}(t)=\hat{d}_{n+1}(t)=d_{2n+1}(t)=d_{n+1}(t)=e$, for all $t\geq m$. Continuing this argument (56) is readily proved. Thus,

$$\hat{d}_{3n-1}(t) = \hat{d}_{2n-1}(t) < ne = d_{2n-1}(t) = d_{3n-1}(t), \ \forall t < mn.$$
(57)

Result follows from (41) as $\hat{d}_{\min}(0) = 0$, $d_{\max}(G) = ne$ and

$$mn = n \left[\frac{e}{e_{min}} \right] = \left[\frac{d_{\max}(G) - \hat{d}_{\min}(0)}{e_{min}} \right].$$

Proof of Lemma 6: Use induction. As the result holds for t=0, suppose for some $t\geq 0$, $\hat{d}_i(t)\geq \hat{D}_i(t)$, $\forall i\in V$. Suppose $j\in\mathcal{N}(i)$ is a current constraining node of i at time t in (45) while $k\in\mathcal{N}(i)$ is a current constraining node of i at time t in (49). Then from (42) and (43), there obtains:

$$\hat{D}_{i}(t+1) = \hat{D}_{k}(t) + e_{ik} - \epsilon
\leq \hat{D}_{j}(t) + e_{ij} - \epsilon
\leq \hat{d}_{j}(t) + e_{ij} - \epsilon
\leq \hat{d}_{j}(t) + e_{ij} + \epsilon_{ij}(t)
= \hat{d}_{i}(t+1).$$
(58)

Proof of Lemma 7: Consider nodes n_1, \dots, n_{T_1} , such that $n_1 \in S$, and for all $i \in \{1, \dots, T_1 - 1\}$ n_i is a true constraining node of n_{i+1} in G^- . Every node is in one such sequence. We assert that for all $i \in \{1, \dots, T_1\}$,

$$d_{n_i} \le D_{n_i} + (i-1)\epsilon. \tag{59}$$

As $n_1 \in S$ the result holds for i=1. Now suppose it holds for some $i \in \{1, \dots, T_1-1\}$. As n_i and n_{i+1} are neighbors in both G and G^- , n_i is the true constraining node of n_{i+1} in G^- , and (46) holds, (59) holds as

$$\begin{array}{rcl} d_{n_{i+1}} & \leq & d_{n_i} + e_{n_i n_{i+1}} \\ & \leq & D_{n_i} + (i-1)\epsilon + e_{n_i n_{i+1}} \\ & = & D_{n_i} + (i-1)\epsilon + e_{n_i n_{i+1}}^- + \epsilon \\ & = & D_{n_{i+1}} + i\epsilon. \end{array}$$

Then (50) follows as from Lemma 1, $T_1 \leq \mathcal{D}(G^-)$.

To prove (47) suppose $\mathcal{D}(G) > \mathcal{D}(G^-)$. Then there is a sequence of nodes in G, without loss of generality $\{1, \cdots, n\}$, such that $1 \in S$ and for all $i \in \{1, \cdots, n\}$, i is a true constraining node of i+1. Further

$$d_n = \sum_{i=1}^{n-1} e_{i,i+1}. (60)$$

As $\mathcal{D}(G) > \mathcal{D}(G^-)$, there is a sequence of nodes $\{p_1, \cdots, p_l, n\}$ such that in G^- , $p_1 \in S$ and p_i is a true constraining node of p_{i+1} , p_l is a true constraining node of n and l+1 < n. Further

$$D_n = \sum_{i=1}^{l-1} (e_{p_i, p_{i+1}} - \epsilon) + e_{p_l, n} - \epsilon.$$
 (61)

By definition the distance from S along the path comprising $\{1, \dots, n\}$ in G^- is less than D_n , i.e.

$$\sum_{i=1}^{l-1} (e_{p_i, p_{i+1}} - \epsilon) + e_{p_l, n} - \epsilon < \sum_{i=1}^{n-1} (e_{i, i+1} - \epsilon)$$

$$\Rightarrow \sum_{i=1}^{l-1} e_{p_i, p_{i+1}} + e_{p_l, n} < \sum_{i=1}^{n-1} e_{i, i+1},$$

where the last inequality follows from the fact that l + 1 < n, and violates (60). The contradiction proves (47).



Yuanqiu Mo was born in Yangzhou, China, in 1991. He received the B.E. degree in electrical engineering from the Nantong University, Nantong, China, in 2013, and the M.E. degree in electrical engineering from the Harbin Institute of Technology, Harbin, China, in 2015. He is currently working towards the Ph.D. degree in electrical engineering at the University of Iowa. His research interests include control systems and compressed sensing for MRI.



Soura Dasgupta, (M'87, SM'93, F'98) was born in 1959 in Calcutta, India. He received the B.E. degree (Hons. I) in Electrical Engineering from the University of Queensland (Australia) in 1980, and the Ph.D. in Systems Engineering from the Australian National University, in 1985. He is currently Professor of Electrical and Computer Engineering at the University of Iowa, U.S.A and holds an appointment with Shandong Computer Science Center (National Supercomputer Center in Jinan), Shandong Provincial Key Laboratory of Computer Networks.

In 1981, he was a Junior Research Fellow in the Electronics and Communications Sciences Unit at the Indian Statistical Institute, Calcutta. He has held visiting appointments at the University of Notre Dame, University of Iowa, Universite Catholique de Louvain-La-Neuve, Belgium, Tata Consulting Services, Hyderabad, and the Australian National University

From 1988 to 1991, 1998 to 2009 and 2004 to 2007 he respectively served as an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IEEE Control Systems Society Conference Editorial Board, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS- II. He is a co-recipient of the Gullimen Cauer Award for the best paper published in the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS in the calendar years of 1990 and 1991, a past Presidential Faculty Fellow, a past subject editor for the International Journal of Adaptive Control and Signal Processing, and a member of the editorial board of the EURASIP Journal of Wireless Communications. In 2012 he was awarded the University Iowa Collegiate Teaching award. In the same year he was selected by the graduating class for an award on excellence in teaching and commitment to student success. Since 2015 he has been a 1000 Talents Scholar in the People's Republic of China.

His research interests are in Controls, Signal Processing and Communications. He was elected a Fellow of the IEEE in 1998.



Jacob Beal is a scientist at Raytheon BBN Technologies in Cambridge, Massachusetts. His research focuses on the engineering of robust adaptive systems, particularly on the problems of aggregate-level modeling and control for spatially distributed systems like pervasive wireless networks, robotic swarms, and natural or engineered biological cells. Dr. Beal received a PhD in electrical engineering and computer science from MIT. He is an associate editor of ACM Transactions on Autonomous and Adaptive Systems, is on the steering committee of

the IEEE International Conference on Self-Adapting and Self-Organizing Systems (SASO), and is a founder of the Spatial Computing Workshop series. He is a Senior Member of IEEE. Contact him at jakebeal@ieee.org