

# Consensus with Ternary Messages

Alex Olshevsky

**Abstract**—We provide a protocol for real-valued average consensus by networks of agents which exchange only a single message from the ternary alphabet  $\{-1, 0, 1\}$  between neighbors at each step. Our protocol works on time-varying undirected graphs subject to a connectivity condition, has a worst-case convergence time which is polynomial in the number of agents and the initial values, and requires no global knowledge about the graph topologies on the part of each node to implement except for knowing an upper bound on the degrees of its neighbors.

## I. INTRODUCTION

The average consensus problem asks for a protocol by means of which  $n$  agents with initial values  $x_1(0), \dots, x_n(0)$  can compute the average  $\frac{1}{n} \sum_{i=1}^n x_i(0)$  subject to unpredictably time-varying restrictions on inter-agent communication. Recent years have seen a surge of interest in consensus protocols due to their widespread use as building blocks for distributed control laws in multi-agent systems; for example, consensus protocols have been used for formation maintenance [18], [12], [9], coverage control [24], [25], [8], network clock synchronization [23], [6], [33], distributed task assignment and partitioning [10], statistical inference in sensor networks [1], [35], [11], and many other contexts in which centralized control is absent and agent motion and time-varying interference can lead to repeated failures of communication or sensing.

Consensus protocols typically involve each node  $i$  maintaining a variable  $x_i(t)$  which is updated from time  $t-1$  to time  $t$  by setting  $x_i(t)$  to be a convex combination of those  $x_j(t-1)$  for which  $j$  and  $i$  are neighbors in some undirected graph  $G(t) = (\{1, \dots, n\}, E(t))$ . The graph sequence  $G(t)$  is meant to capture the constraints which dictate which pairs of agents can communicate or sense each other at each time. It is assumed that each  $G(t)$  has a self-loop at every node so that  $x_i(t)$  always depends on its previous value  $x_i(t-1)$ . A typical update is the Metropolis iteration (first introduced within the context of consensus in [35]), defined as

$$x_i(t) = x_i(t-1) + \sum_{j \in N_i(t)} \frac{x_j(t-1) - x_i(t-1)}{D(i, j, t)} \quad (1)$$

where  $N_i(t)$  is the set of neighbors of node  $i$  in  $G(t)$ ,  $d_i(t)$  is the degree of node  $i$  in  $G(t)$ , and  $D(i, j, t)$  is a collection of numbers, defined for all pairs  $i, j$  such that  $(i, j) \in G(t)$ , satisfying the symmetry conditions  $D(i, j, t) = D(j, i, t)$  and the lower bounds  $D(i, j, t) \geq \max(d(i), d_j(t))$ . Note that choosing  $D(i, j, t) = \max(d_i(t), d_j(t))$  works, and choosing

$D(i, j, t) = n$  or any common upper bound on  $n$  works as well. We will assume that the numbers  $D(i, j, t)$  are uniformly bounded from above. We note that the nodes do not need to know any global information about the sequence  $G(t)$  to implement this iteration, and we may in fact assume that  $G(t)$  is exogenously or adversarially given subject to the connectivity constraint we will next describe. Namely, we will assume that the graph sequence  $G(t)$  satisfies the long-term connectivity condition

For each  $k$ , the graph  $(\{1, \dots, n\}, \cup_{t=k}^{\infty} E(t))$  is connected, (2)

meaning that the communication restrictions faced by the nodes do not disconnect the network into noncommunicating groups after some finite time  $k$ . Under this assumption, the Metropolis iteration of Eq. (1) has the property that

$$\lim_{t \rightarrow \infty} x_i(t) = \frac{1}{n} \sum_{i=1}^n x_i(t) \quad \text{for each } i = 1, \dots, n$$

meaning that all agents succeed in converging to the average. We refer the reader to the papers [2], [16], [26], [28], [34], [35] for proofs of this and similar assertions.

Consensus protocols have proven to be useful for multi-agent control due to their attractive robustness properties (namely, that the communication sequence  $G(t)$  can vary arbitrarily and unpredictably subject only to the relatively weak long-term connectivity constraint of Eq. (2)) and most importantly due to their local, distributed nature. Indeed, all the information that node  $i$  needs to implement the Metropolis iteration, from its neighbor's values to upper bounds on its neighbors' degrees, is locally available. In the event that information exchange is wireless, nodes may broadcast their values and degrees to their neighbors; on the other hand, if  $x_i(t)$  are quantities like positions or velocities which are physically measured by neighbors, implementation of the Metropolis iteration usually requires nodes to know upper bounds  $D(i, j, t)$  on the degrees of their neighbors. These bounds can be based on the inherent sensing constraints (e.g., no node can be sensing too many other nodes simultaneously) or on an upper bound for the total number of nodes in the system.

We remark that a number of recent advances in multi-agent control have proceeded by reduction to an appropriately defined consensus problem [8], [12], [9], [23], [33]. Moreover, under slightly stronger assumptions on the sequence  $G(t)$ , it is possible to design average consensus algorithms with convergence time bounds which scale quadratically with the number of nodes  $n$  [27].

Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, 61801, USA (aolshev2@illinois.edu).

However, a limitation of consensus protocols lies in the assumption that agent  $i$  can update  $x_i(t)$  as a function of the states  $x_j(t-1)$  of its neighbors in the graph  $G(t)$ . While this is very natural in some settings (such as formation or coverage control, where  $x_j(t)$  represents the position of agent  $j$ , which neighboring agent  $i$  can be reasonably assumed to sense) in other contexts (such as clock synchronization or statistical inference in sensor networks) this implies that the agents can exchange real numbers at every step, a clearly unrealistic assumption.

Indeed, sensor networks which run clock synchronization or distributed estimation protocols cannot transmit real-valued messages to neighbors; the physics of information transmission over the wireless medium instead limit the nodes to messages which come from a finite alphabet. The goal of this paper is to provide a consensus protocol which works in this setting. Specifically, we provide a deterministic protocol for the nodes to exchange messages with neighbors and update  $x_i(t)$  which has the following properties:

- 1) For every time-varying sequence  $G(t)$  subject to a certain connectivity assumption, each  $x_i(t)$  converges to the average of the initial values  $\frac{1}{n} \sum_{i=1}^n x_i(0)$ .
- 2) The number of bits that is transmitted from agent to agent at each time  $t$  is bounded above independently of  $t, n$  and all other problem parameters (in fact in our protocol the agents transmit a single element from the ternary alphabet  $\{-1, 0, 1\}$  to each neighbor at every  $t$ ).
- 3) The protocol does not require any global knowledge of the graph sequence  $G(t)$  on the part of the nodes.
- 4) The protocol has a worst-case convergence time which is provably scales polynomially in the number of nodes  $n$  and the initial values  $x_i(0)$ .

We remark that features (1) and (3) are particularly crucial since much of the attractiveness of consensus protocols arises from their abilities to cope with link failures and their local, distributed nature. Feature (4) is clearly useful: the appeal of any protocol is increased by the availability of polynomial worst-case convergence bounds. Finally, feature (2) ensures that we do not implicitly assume that an arbitrarily large amount of data can be transmitted on each link before the graph changes.

Much like the Metropolis protocol, our protocol assumes every node has access to upper bounds  $D(i, j, t)$  on the degrees of its neighbors. Moreover, we remark that our protocol may be considered to have binary (rather than ternary) messages if we identify no transmission on a link with a “0” message.

A number of consensus protocols in which agents exchange finitely many bits at every step have recently been proposed. We mention the paper [17] which initiated the literature, the follow-on works [27], [13], [20], [32], [22], as well as the related papers [3], [4]. In [17] and follow-up papers nodes exchange quantized values in order to eventually approach a neighborhood of the initial average. Variations on this theme have been studied in the past several

years, and we mention that the recent preprint [32] has the best known convergence time bounds at the time of writing for these updates in discrete time. Finally, a similar consensus process has been studied recently over finite fields in [30].

More directly related to the current work are the papers [5], [7], [14], [19], [21] which considered the problem using quantized communications to converge to the real-valued average of the initial values *exactly*, as we do here. The present paper is directly motivated by this literature, in particular by the observation that each of the protocols in these papers lacks at least two of the features (1)–(4) above. Indeed, many of these protocols only work on fixed graphs, and furthermore utilize updates at each node which depend on global information about the network (for example by having each node’s update depend on the eigenvalues of a matrix built from the network). By contrast, the protocol we propose in this paper requires no knowledge about the time-varying graph sequence  $G(t)$  to implement with the possible exception of each node having upper bounds on the degrees of its neighbors.

We next state our main result in Section II, where we provide an informal description as well as a formal statement of our protocol. The proof of the main convergence theorem is not given in this document, and may be found in a technical report of the same title as the current paper posted on the arxiv [29].

## II. OUR RESULTS

We first reprise the notation we have introduced:  $G(t)$  is a sequence of undirected graphs with a self-loop at every node,  $N_i(t)$  is the set of neighbors of node  $i$  in  $G(t)$ , and  $d_i(t)$  is the degree of node  $i$  in  $G(t)$ .

### A. Intuitive description of the protocol

We begin by informally describing the idea behind our protocol. We would like to run the Metropolis update of Eq. (1), but without the ability to transmit real numbers, node  $i$  will not know the values of its neighbors  $x_j(t-1)$  exactly. Consequently, node  $i$  will maintain estimates of  $x_j(t)$ ; we will use the notation  $\hat{x}_{i,j,\text{in}}(t)$  for the estimate that node  $i$  has at time  $t$  for  $x_j(t)$ .

At each time  $t$ , node  $i$  will receive a message from each of its neighbors  $j \in N_i(t)$  from the alphabet  $\{-1, 0, 1\}$ . If it receives a  $+1$ , it will add  $1/t^\alpha$  to  $\hat{x}_{i,j,\text{in}}$ ; if it receives a  $-1$ , it adds a  $-1/t^\alpha$  to the same; and if it receives a zero, it leaves  $\hat{x}_{i,j,\text{in}}$  unchanged. Naturally, node  $j$  decides to send a  $1, 0, -1$  depending on whether  $\hat{x}_{i,j,\text{in}}(t-1)$  is too low by at least  $1/t^\alpha$ , within  $1/t^\alpha$  of the true value, or too high by a factor of at least  $1/t^\alpha$ , respectively. A key point is that even though it is node  $i$  which maintains the variable  $\hat{x}_{i,j,\text{in}}$ , node  $j$  knows what it is because it is built from previous messages sent by node  $j$  to node  $i$ .

Note that this update rules may require nodes to send different messages to different neighbors.

The number  $\alpha$  will be in  $(0, 1)$  so that

$$\sum_{t=1}^n \frac{1}{t^\alpha} = +\infty$$

which will be key in ensuring that the estimates  $\hat{x}_{i,j,\text{in}}$  increasingly become accurate for links that appear regularly.

Meanwhile, the nodes will implement the Metropolis update, however with each node  $i$  using its latest estimate  $\hat{x}_{i,j,\text{in}}$  instead of the true value  $x_j(t-1)$ . However, node  $i$  will not include all its neighbors in the Metropolis updates; rather, it will include only those nodes  $j$  which sent it a zero at time  $t$  (meaning that node  $i$  estimate of  $j$ 's value is not too far from the truth) and whose associated estimates  $\hat{x}_{i,j,\text{in}}(t)$  are sufficiently far from  $x_i(t)$  (so that the inevitable perturbation error inherent in using imprecise estimates does not affect the convergence analysis too much).

Furthermore, we introduce a stepsize of  $1/t^\beta$  into the Metropolis algorithm where  $\beta > \alpha$  and  $\beta \in (0,1)$ . By choosing  $\beta > \alpha$ , we ensure that agents change their values  $x_i$  slower than estimates  $\hat{x}_{i,j,\text{in}}$  change. Intuitively, while the estimates  $\hat{x}_{i,j,\text{in}}$  will get accurate by an additive factor of  $1/t^\alpha$  whenever they are very inaccurate, the values  $x_i$  will have their movement attenuated by a factor of  $1/t^\beta$ . The introduction of the  $1/t^\beta$  stepsize ensures that not only do the estimates eventually "catch up" to the true values but also that we can give simple bounds on how long it takes until estimates become accurate.

### B. Formal description of the protocol

The nodes begin with initial values  $x_i(0)$ . Every node  $i$  will maintain variables  $\hat{x}_{i,j,\text{in}}(t), \hat{x}_{i,j,\text{out}}(t)$  for every node which has been its neighbor in some past  $G(t)$ . Node  $i$  initializes

$$\hat{x}_{i,j,\text{in}}(t-1) = \hat{x}_{i,j,\text{out}}(t-1) = 0$$

at the first time  $t$  when the edge  $(i,j)$  belongs to  $E(t)$ .

At each iteration  $t = 1, 2, 3, \dots$ , node  $i$  will send a value from the set  $\{-1, 0, 1\}$  to each of its neighbors  $j$  in  $G(t)$ . The value  $i$  sends to  $j$  is

$$q_{i \rightarrow j}(t) = R[t^\alpha(x_i(t-1) - \hat{x}_{i,j,\text{out}}(t-1))]$$

where

$$R[x] = \begin{cases} 1 & \text{if } x > 1 \\ 0 & \text{if } -1 \leq x \leq 1 \\ -1 & \text{if } x < -1 \end{cases}$$

Note that node  $i$  may send different messages to different neighbors. After sending  $q_{i \rightarrow j}(t)$  to each neighbor  $j \in N_i(t)$  and receiving  $q_{j \rightarrow i}(t)$  from the same, node  $i$  updates the values  $\hat{x}_{i,j,\text{in}}(t), \hat{x}_{i,j,\text{out}}(t)$  as

$$\begin{aligned} \hat{x}_{i,j,\text{out}}(t) &= \hat{x}_{i,j,\text{out}}(t-1) + \frac{q_{i \rightarrow j}(t)}{t^\alpha} \\ \hat{x}_{i,j,\text{in}}(t) &= \hat{x}_{i,j,\text{in}}(t-1) + \frac{q_{j \rightarrow i}(t)}{t^\alpha} \end{aligned}$$

and then updates its value  $x_i(t)$  as

$$x_i(t) = x_i(t-1) + \frac{1}{t^\beta} \sum_{j \in S(i,t)} \frac{\hat{x}_{i,j,\text{in}}(t) - \hat{x}_{i,j,\text{out}}(t)}{4D(i,j,t)} \quad (3)$$

Here  $D(i,j,t)$  is, as in the Metropolis protocol, a collection of numbers satisfying the symmetry conditions

$D(i,j,t) = D(j,i,t)$  and the lower bounds  $D(i,j,t) \geq \max(d_i(t), d_j(t))$ ; and  $S(i,t)$  is the set of neighbors  $j$  of node  $i$  in  $G(t)$  which satisfy

$$|\hat{x}_{i,j,\text{in}}(t) - \hat{x}_{i,j,\text{out}}(t)| > \frac{4}{t^\alpha}$$

and

$$q_{j \rightarrow i}(t) = 0, \quad q_{i \rightarrow j}(t) = 0.$$

### C. Main result

We now provide the main convergence theorem for our protocol. We begin with a few definitions needed to specify the class of graph sequences on which our protocol is guaranteed to work.

**Definition 1.** We will call the graph sequence  $G(t)$  *B-core-connected* if there exists a set of edges  $E_\infty \subset \cup_t E(t)$  such that the graph  $(\{1, \dots, n\}, E_\infty)$  is connected; and

$$E_\infty \subset \cup_{t=kB+1}^{(k+1)B} E(t)$$

for every nonnegative integer  $k$ .

That is, a sequence is *B-core-connected* if there is a set of edges, forming a connected graph, each of which appears in every interval  $[kB+1, (k+1)B]$ . We will say that the edges in  $E_\infty$  are *core edges*. We will discuss the relationship between this assumption and the more usual *B-connectivity* assumption typically made in the analysis of consensus algorithms shortly.

Our main result provides upper bounds on the time until our consensus protocol reduces a certain measure of disagreement to a small value forever. We now define this measure (this is  $V_2(t)$  in the next definition) as well as several related concepts.

### Definition 2.

$$\begin{aligned} M(x) &= \max_{i=1, \dots, n} x_i \\ m(x) &= \min_{i=1, \dots, n} x_i \\ W(x) &= M(x) - m(x) \\ V_2(x) &= \sqrt{\sum_{i=1}^n \left( x_i - \frac{1}{n} \sum_{j=1}^n x_j \right)^2} \end{aligned}$$

Note that will use the natural shorthands  $M(t), m(t), W(t), V_2(t)$  for  $M(x(t)), m(x(t)), W(x(t)), V_2(x(t))$ . Finally, we will use the notation  $D = \sup_{i,j,t} D(i,j,t)$  for the supremum of the all the quantities  $D(i,j,t)$  upper bounding the degrees. Note that if  $D(i,j,t) = \max_i(d_i(t), d_j(t))$  then we can trivially bound  $D \leq n$ .

We can now state the main result of this paper. The proof may be found in the technical report [29].

**Theorem 3.** If  $0 < \alpha < \beta < 1$  then for all nodes  $i$ , initial values  $x(0)$ , and  $B$ -core-connected sequences  $G(t)$ , it is true that

$$\lim_{t \rightarrow \infty} x_i(t) = \frac{1}{n} \sum_{j=1}^n x_j(0).$$

Moreover, if<sup>1</sup> Eq. (4) holds then

$$V_2(x(t)) \leq \epsilon$$

for all  $i, j$ .

Our result states that the consensus protocol of the previous section succeeds in computing the average on  $B$ -core-connected sequence and provides a bound for its convergence time.

**Remark 4.** While the above convergence time expression is somewhat unwieldy, we emphasize that it is polynomial in  $n, \epsilon, B, W(0), V(0), \|x(0)\|_\infty$  for any choice of  $\alpha$  and  $\beta$  satisfying the assumption  $0 < \alpha < \beta < 1$ .

**Remark 5.** Observe that there is a tradeoff between the steady-state and transient terms in the convergence time. In particular, as  $\epsilon \rightarrow 0$ , the term  $(8n^{1.5}/\epsilon)^{1/\alpha}$  is going to dominate all the other terms in the convergence time expression. Consequently, to obtain the best asymptotic decay rate we should choose  $\alpha$  close to 1 which means we must also choose  $\beta$  close to 1. In that case, error decay at time  $t$  decays nearly as well as  $O(n^{1.5}/t)$  as  $t$  approaches infinity. However, choosing  $\alpha_n \rightarrow 1$  and  $\beta_n \rightarrow 1$  causes the transient term to blow up, so that it takes longer and longer until the asymptotically dominant term dominates the other terms.

In general, there is no single best choice of  $\alpha, \beta$  which optimizes the convergence time expression; rather every choice gives us a tradeoff between transient bounds and steady-state decay. For example, choosing  $\alpha = 3/4, \beta = 7/8$  gives us that the time until  $V_2(x(t)) \leq \epsilon$  is

$$O\left((B + BW(0))^8\right) + O(B\|x(0)\|_\infty)^8 + O(B) + O(n^{24}D^8B^8) \\ + \max\left(O(n^{24}D^8B^8(\log \frac{V_2(0)}{\epsilon})^8), O\left(\frac{n^2}{\epsilon^{4/3}}\right)\right)$$

Note that every term which does not have an  $\epsilon$  in it will become negligible as  $\epsilon \rightarrow 0$ . In this limit, the dominant term will be  $O(\frac{8n^2}{\epsilon^{4/3}})$  which will grow faster as  $\epsilon \rightarrow 0$  than the logarithmic term  $O(n^{24}D^8B^8 \log \frac{V_2(0)}{\epsilon})$ .

On the other hand, choosing  $\alpha = 1/4, \beta = 1/2$  gives us that the time until  $V_2(x(t)) \leq \epsilon$  is

$$O\left((B + BW(0))^4\right) + O(B\|x(0)\|_\infty)^{8/3} + O(B) + O(n^6D^2B^2) \\ + \max\left(O(n^6D^2B^2(\log \frac{V_2(0)}{\epsilon})^2), O\left(\frac{n^6}{\epsilon^4}\right)\right)$$

<sup>1</sup>The notation  $\lceil x \rceil$  refers to the smallest integer which is at least  $x$ .

which has a worse asymptotic term  $O(\frac{n^6}{\epsilon^4})$  but the rest of the terms are smaller if  $B, W(0), \|x(0)\|_\infty, V_2(0)$  are large.

**Remark 6.** The analysis of consensus algorithms usually relies on a weaker notion of connectivity, namely the so-called  $B$ -connectivity (sometimes called uniform connectivity) condition: a sequence  $G(t)$  of undirected graphs is called  $B$ -connected if for each  $k$ , the graph

$$(\{1, \dots, n\}, \cup_{t=kB+1}^{(k+1)B} E(t))$$

is connected. This encompasses a wider class of sequences compared to notion of  $B$ -core-connectivity which we use, which requires not only the above graph to be connected, but also that it has the same connected subgraph for every  $k$ .

However, our convergence proof works on  $B$ -connected sequences, so that  $\lim_{t \rightarrow \infty} x_i(t) = \frac{1}{n} \sum_{j=1}^n x_j(0)$  holds in fact on all  $B$ -connected undirected graph sequences  $G(t)$ . We will justify this claim in a remark following the main proof. However, the convergence-time result is only known to hold for  $B$ -core-connected sequences, and it is an open problem whether one can obtain a similar convergence time on  $B$ -connected sequences.

**Remark 7.** Our protocol requires node  $i$  to keep track of the numbers  $\hat{x}_{i,j,\text{in}}, \hat{x}_{i,j,\text{out}}$  for every other node  $j$  it interacts with. By contrast, the standard consensus algorithm keeps track of only a single real number, namely  $x_i(t)$  at node  $i$ . Unfortunately, it seems that a “storage blowup” phenomenon of this sort is unavoidable, though it remains an open question to prove this in a formal sense.

As a consequence, our consensus protocol is most attractive on graphs sequences  $G(t)$  in which every node interacts with a relatively small number of neighbors. One such example is the geometric random graph model of wireless networks, in which sensors have locations in  $[0, 1]^2$  and every node is connected to an expected  $O(\log n)$  neighbors [31]. Our consensus protocol will work in such networks, even if unpredictable interference or malicious jamming makes links unreliable, with every node needing to store only  $O(\log n)$  additional estimates.

**Remark 8.** We have not assumed that the nodes know the constant  $B$ . However, if the nodes do happen to know  $B$  or an upper bound on it, they may use it to somewhat reduce their storage requirements. Rather than keep track of the estimates  $\hat{x}_{i,j,\text{in}}$  for every node  $j$  that it has interacted with in the past, node  $i$  can instead just keep track of estimates for just those nodes it has interacted with in the past  $B$  steps.

Our main result, Theorem 3, will then hold verbatim, since our proof of this theorem (in the next section) proceeds by analyzing reductions in  $V_2(t)$  obtained when neighbors connected by core edges include each other in their Metropolis updates, which are unaffected by this modification.

$$t \geq 2^{\frac{1}{1-\beta}} \left( 2^{\frac{2}{1-\alpha}} [32B + 8BW(0)]^{\frac{1}{\beta-\alpha}} + (32B\|x(0)\|_{\infty})^{\frac{2}{1-\alpha}} + 11B + (300n^3DB)^{\frac{1}{1-\beta}} \right) + \max \left( \left( 150n^3DB \log \frac{\epsilon}{V_2(0)} \right)^{\frac{1}{1-\beta}}, \left( \frac{8n^{1.5}}{\epsilon} \right)^{\frac{1}{\alpha}} \right) \quad (4)$$

**Remark 9.** The introduction of the stepsize  $1/t^{\beta}$  with  $\beta > \alpha$  is crucial for our proof here. A main thrust of our argument is that after some transient period, we will have that  $1/t^{\beta}$  is negligible compared to  $1/t^{\alpha}$ , so that estimates across each core link get accurate due to messages between nodes much faster than they drift apart due to node updates. This argument is the source of the exponent  $\frac{1}{\beta-\alpha}$  in the first term of our final convergence time bound.

#### REFERENCES

- [1] S. Boyd, A. Ghosh, B. Prabhakar, D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508-2530, 2006.
- [2] V.D. Blondel, J.M. Hendrickx, A. Olshevsky, J.N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," *Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, Spain, 2005.
- [3] K. Cai, H. Ishii, "Quantized consensus and averaging on gossip digraphs," *IEEE Transactions on Automatic Control*, vol. 56, no. 9, p. 2087-2100, 2011.
- [4] K. Cai, H. Ishii, "Convergence time analysis of quantized gossip consensus on digraphs," *Automatica*, vol. 48, no. 9, p. 2344-2351, 2012.
- [5] R. Carli, F. Bullo and S. Zampieri, "Quantized average consensus via dynamic coding/decoding schemes," *International Journal of Nonlinear and Robust Control*, vol. 20, no. 2, pp. 156-175, 2010.
- [6] R. Carli, A. Chiuso, L. Schenato, S. Zampieri, "Optimal synchronization for networks of noisy double integrators," *IEEE Transactions on Automatic Control*, vol. 56, no. 5, pp. 1146-1152, 2011.
- [7] R. Carli, F. Fagnani, P. Frasca and S. Zampieri, "Gossip consensus algorithms via quantized communication," *Automatica*, vol. 46, no. 1, pp. 70-80, 2010.
- [8] C. Gao, J. Cortes, F. Bullo, "Notes on averaging over acyclic graphs and discrete coverage control," *Automatica*, vol. 44, no. 8, pp. 2120-2127, 2008.
- [9] J.S. Caughman, G. Lafferriere, J.J.P. Veerman, and A. Williams, "Decentralized control of vehicle formations," *Systems and Control Letters*, vol. 54, no. 9, pp. 899-910, 2005.
- [10] H.L. Choi, L. Brunet, J.P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912-926, 2009.
- [11] A. Dimakis, A.D. Sarwate, M. J. Wainwright, "Geographic gossip: efficient averaging for sensor networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 3, pp. 1205-1216, 2008.
- [12] J.A. Fax and R.M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465-1476, 2004.
- [13] P. Frasca, "Continuous-time quantized consensus: convergence of Krasovskii solutions," *Systems and Control Letters*, vol. 61, no. 2, pp. 273-278, 2012.
- [14] P. Frasca, R. Carli, F. Fagnani and S. Zampieri, "Average consensus on networks with quantized communication," *International Journal of Nonlinear and Robust Control*, vol. 19, no. 16, pp. 1787-1816, 2009.
- [15] M. Guo, D. V. Dimarogonas, "Consensus with quantized relative state measurements," *Automatica*, vol. 49, no. 8, pp. 2531-2537, 2013.
- [16] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 3, pp. 988-1001, 2003.
- [17] A. Kashyap, T. Basar and R. Srikant, "Quantized consensus," *Automatica*, vol. 43, no. 7, pp. 1192-1203, 2007.
- [18] J.R. Lawton, R.W. Beard, and B. Young, "A decentralized approach to formation maneuvers," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 6, pp. 933-941, 2003.
- [19] T. Li, M. Fu, L. Xie and J. F. Zhang, "Distributed consensus with limited communication data rate," *IEEE Transactions on Automatic Control*, vol. 56, no. 2, pp. 279-292, 2011.
- [20] S. Liu, T. Li, L. Xie, M. Fu, J. F. Zhang, "Continuous-time and sampled-data based average consensus with logarithmic quantizers," preprint.
- [21] S. Liu, T. Li, L. Xie, "Distributed consensus for multi-agent systems with communication delays and limited data rate," *SIAM Journal on Control and Optimization*, vol. 49, no. 6, pp. 2239-2262, 2011.
- [22] J. Lavaei, R.M. Murray, "Quantized consensus by means of gossip algorithm," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 19-32, 2012.
- [23] Q. Li, D. Rus, "Global clock synchronization in sensor networks," *IEEE Transactions on Computers*, vol. 55, no. 2, pp. 214-226, 2006.
- [24] S. Martinez, T. Karatas, F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243-255, 2004.
- [25] S. Martinez, J. Cortes, F. Bullo, "Motion coordination with distributed information," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 4, pp. 75-88, 2007.
- [26] L. Moreau, "Stability of multi-agent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, pp. 169-182, 2005.
- [27] A. Nedic, A. Olshevsky, A. Ozdaglar and J. N. Tsitsiklis, "On distributed averaging algorithms and quantization effects," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2506-2517, 2009.
- [28] R. Olfati-Saber, R.M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 3, pp. 1250-1533, 2004.
- [29] A. Olshevsky, "Consensus with Ternary Messages," preprint, <http://arxiv.org/abs/1212.5768>.
- [30] F. Pasqualetti, D. Borra, F. Bullo, "Consensus networks over finite fields," preprint, available at <http://arxiv.org/abs/1301.4587>.
- [31] M. Penrose, *Random Geometric Graphs*, Oxford University Press, 2003.
- [32] S. Shang, P. W. Cuff, P. Hui, S. R. Kulkarni, "An upper bound on the convergence time for quantized consensus," preprint, available at <http://arxiv.org/abs/1208.0788>.
- [33] L. Schenato and F. Fiorentin, "Average timesynch: a consensus-based protocol for clock synchronization in wireless sensor networks," *Automatica*, vol. 47, no. 9, pp. 1878-1886, 2011.
- [34] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, 1986, pp. 803-812.
- [35] L. Xiao, S. Boyd, S.J. Kim, "Distributed average consensus with least-mean square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33-46, 2007.