Distributed estimation and control applications using linear consensus algorithms

Federica Garin and Luca Schenato

Abstract In this chapter we present a popular class of distributed algorithms, known as linear consensus algorithms, which have the ability to compute the global average of local quantities. These algorithms are particularly suitable in the context of multi-agent systems and networked control systems, i.e. control systems that are physically distributed and cooperate by exchanging information through a communication network. We present the main results available in the literature about the analysis and design of linear consensus algorithms,for both synchronous and asynchronous implementations. We then show that many control, optimization and estimation problems such as least squares, sensor calibration, vehicle coordination and Kalman filtering can be cast as the computation of some sort of averages, therefore being suitable for consensus algorithms. We finally conclude by presenting very recent studies about the performance of many of these control and estimation problems, which give rise to novel metrics for the consensus algorithms. These indexes of performance are rather different from more traditional metrics like the rate of convergence and have fundamental consequences on the design of consensus algorithms.

1 Introduction

In the past decades we have being witnessing the growth of engineering systems composed by a large number of devices that can communicate and cooperate to achieve a common goal. Although complex large-scale monitoring and control sys-

Federica Garin
Department of Information Engineering, University of Padova, Italy e-mail: federica.garin@dei.unipd.it

Luca Schenato
Department of Information Engineering, University of Padova, Italy e-mail: schenato@dei.unipd.it

tems are not new, as for example nuclear plants and air traffic control, a new architectural paradigm is emerging, mainly due to the adoption of smart agents, i.e., devices that have the ability to cooperate and to take autonomous decisions without any supervisory system. In fact, traditional large-scale systems have a centralized or at best a hierarchical architecture, which has the advantage to be relatively easy to be designed and has safety guarantees. However, these systems require very reliable sensors and actuators, are generally very expensive, and do not scale well due to communication and computation limitations. The recent trend to avoid these problems is to substitute costly sensors, actuators and communication systems with a larger number of devices that can autonomously compensate potential failures and computation limitations through communication and cooperation. Although very promising, this new paradigm brings new problems into the picture, mainly due to the lack of analysis and design tools for such systems. In particular, there are only few tools for predicting the global behavior of the system as a whole starting from the local sensing and control rules adopted by the smart sensors and actuators. As a consequence, there has been a strong effort in past years by many engineering areas to develop such tools.

One of the most promising tools are the linear consensus algorithms, which are simple distributed algorithms which require only minimal computation, communication and synchronization to compute averages of local quantities that reside in each device. These algorithms have their roots in the analysis of Markov chains [53] and have been deeply studied within the computer science community for load balancing [61, 42] and within the linear algebra community for the asynchronous solution of linear systems [30, 56]. More recently they have been rediscovered and applied by the control and robotics communities for cooperative coordination of multi-agent systems, as surveyed in [52, 51] and in the recent book [12].

The spirit of this chapter is mostly tutorial. We start in Section 2 by presenting a coherent description of the linear consensus algorithms and by surveying the most important results. No prior knowledge is required except for standard linear algebra and control systems theory. A special attention has been placed on the design of such algorithms, which, in our opinion, is one of the most relevant aspects for a control engineer. In Section 3 we illustrate through some examples how these algorithms can be applied to relevant estimation and control problems such as least squares, sensor calibration, and vehicle coordination, just to name a few. Section 4 presents some more recent research directions. More precisely, starting from the analysis of control applications of consensus algorithms, such as those described in Section 3, we show that the performance indexes to be considered are different from the traditional index given by rate of convergence, i.e. the essential spectral radius of the consensus matrix, and in general this index depends on all the eigenvalues of the consensus matrix. This observation has relevant consequences in terms of analysis and design of consensus algorithms, which goes beyond the current results and opens up new research directions, which we believe are particularly relevant for the control community.

2 Linear consensus algorithms: definitions and main results

In this section, we review some of the main results on the analysis and design of consensus algorithms and we also provide references for more recent developments under different scenarios and assumptions. In particular, we will concentrate on linear discrete-time consensus algorithms. However we will give some references to continuous time and nonlinear consensus. We start by introducing some mathematical preliminaries. Let us consider the following linear update equation:

$$x(t+1) = Q(t)x(t) \tag{1}$$

where $x(t) = [x_1(t)\ x_2(t)\ \cdots\ x_N(t)]^T \in \mathbb{R}^N$ and, for all t, $Q(t) \in \mathbb{R}^{N\times N}$ is a *stochastic matrix*, i.e. $[Q(t)]_{ij} = q_{ij}(t) \geq 0$ and $\sum_{j=1}^N q_{ij} = 1, \forall i$, i.e. each row sums to unity. Equation (1) can be written as

$$x_i(t+1) = \sum_{j=1}^N q_{ij}(t)x_i(t), \quad i = 1,\ldots,N \tag{2}$$

$$= x_i(t) + \sum_{j\neq i} q_{ij}(t)(x_j(t) - x_i(t)) \tag{3}$$

where the local updates of each component of the vector x is written explicitly.

A stochastic matrix Q is said *doubly-stochastic* if also $\sum_{i=1}^N q_{ij} = 1, \forall j$, i.e. each column sums to unity. Clearly if a stochastic matrix is symmetric, i.e. $Q = Q^T$, then it is also doubly-stochastic. An important class of doubly-stochastic matrices is given by the class of stochastic matrices which are also circulant. A matrix $Q = \mathrm{circ}(c_1, c_2, \ldots, c_N)$ is a *circulant matrix* if

$$Q = \begin{bmatrix} c_1 & c_2 & c_3 & \cdots & c_N \\ c_N & c_1 & c_2 & \cdots & c_{N-1} \\ \vdots & & \ddots & & \vdots \\ c_2 & c_3 & c_4 & \cdots & c_1 \end{bmatrix} \tag{4}$$

All eigenvalues λ_i of a stochastic matrix Q are included in the unit circle, i.e. $|\lambda_i| \leq 1$, and the vector $\mathbf{1} = [1\,1\cdots 1]^T \in \mathbb{R}^N$ is an eigenvector for Q and its eigenvalue is equal to one, i.e $Q\mathbf{1} = \mathbf{1}$. The *essential spectral radius* $\mathtt{esr}(Q)$ of a stochastic matrix Q is defined as the second largest eigenvalue in modulus of the matrix Q, i.e. if we consider the ordered eigenvalues in modulus $1 = |\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_N|$, then $\mathtt{esr}(Q) = |\lambda_2|$.

Many important results about convergence of consensus algorithms can be reframed as graph properties. Therefore we provide some useful preliminary definitions. We define the (directed) graph associated with a stochastic matrix Q as $\mathscr{G}_Q = (\mathscr{N}, \mathscr{E}_Q)$, where the nodes are $\mathscr{N} = \{1,2,\ldots,N\}$ and the edges are $\mathscr{E}_Q = \{(j,i)\,|\,q_{ij} > 0\}$, i.e. $(j,i) \in \mathscr{E}$ implies that node i can receive information from node j. A graph is *undirected* if $(i,j) \in \mathscr{E}$ implies that also $(j,i) \in \mathscr{E}$.

We also say that a matrix Q is *compatible* with the graph $\mathscr{G} = (\mathscr{N}, \mathscr{E})$ if its associated graph $\mathscr{G}_Q = (\mathscr{N}, \mathscr{E}_Q)$ is such that $\mathscr{G}_Q \subseteq \mathscr{G}$, i.e., is a subgraph of $\mathscr{G}$. We denote with $\mathbb{G}_{\text{sl}}$ the set of graphs which include all self-loops, i.e. $\mathscr{G} \in \mathbb{G}_{\text{sl}}$ if and only if $(i,i) \in \mathscr{E}, \forall i \in \mathscr{N}$. The *in-degree* of a node i is defined as $d_{in}(i) = |\mathscr{V}_{in}(i)|$, where $\mathscr{V}_{in}(i) = \{j \,|\, (j,i) \in \mathscr{E}, i \neq j\}$ is the set of neighbors that can send information to i and $|\cdot|$ indicates the cardinality of a set. Similarly, the *out-degree* of a node i is defined as $d_{out}(i) = |\mathscr{V}_{out}(i)|$ and $\mathscr{V}_{out}(i) = \{j \,|\, (i,j) \in \mathscr{E}, i \neq j\}$. For an undirected graph, in-neighbors and out-neighbors of a node i coincide and they are simply denoted by the set $\mathscr{V}(i)$ whose degree is $d(i) = |\mathscr{V}(i)|$.

The *adjacency matrix* $A \in \{0,1\}^{N \times N}$ of a graph $\mathscr{G} = (\mathscr{N}, \mathscr{E})$ is defined as $[A]_{ij} = 1$ if $(i,j) \in \mathscr{E}$ and $i \neq j$, and $[A]_{ij} = 0$ otherwise. The *Laplacian matrix* L of a undirected graph is defined as $L = D - A$, where $D = \text{diag}\{d(1), d(2), \ldots, d(N)\}$ is diagonal and $d(i)$ is the degree of node i. The Laplacian L is positive semidefinite and $L\mathbf{1} = 0$.

A graph is *rooted* if there exists a node $k \in \mathscr{N}$ such that for any other node $j \in \mathscr{N}$ there is a unique path from k to j. A graph is *strongly connected* if there is a path from any node to any other node in the graph. Clearly a strongly connected graph implies that it is also rooted for any node. The *diameter* of a graph is defined as the length of the longest among all shortest paths connecting any two nodes in a strongly connected graph. A graph is *complete* if $(i,j) \in \mathscr{E}, \forall i,j \in \mathscr{N}$. The *union* of two graphs $\mathscr{G}_1 = (\mathscr{N}, \mathscr{E}_1)$ and $\mathscr{G}_2 = (\mathscr{N}, \mathscr{E}_2)$ is defined as the graph $\mathscr{G} = (\mathscr{N}, \mathscr{E}) = \mathscr{G}_2 \cup \mathscr{G}_1$ where $\mathscr{E} = \mathscr{E}_1 \cup \mathscr{E}_1$.

2.1 Analysis

In this section we describe three main frameworks for modeling consensus algorithms. The first is related to static synchronous implementation, where updates at each node are performed simultaneously, thus being well-represented by constant matrices. The second and the third are both more suitable for modeling asynchronous implementations, where information exchanges and local variable updates are not necessarily coordinated, thus being well-represented by time-varying matrices. The second framework addresses the problem of finding the weakest sufficient conditions that guarantee convergence to consensus from a worst-case point of view, thus being able to characterize a wide class of consensus implementations. The drawback of this approach is that it is very hard to estimate performance indexes such as the rate of convergence and, when possible, the predictions are often over-pessimistic. The third framework considers randomized asynchronous implementations which has three main advantages as compared to the second approach. The first advantage is that randomized communication and updates require almost no coordination among nodes and are easy to implement in practice. The second advantage is that this approach naturally models stochastic nature of the environment, such as communication losses, communication noise and quantization. The third advantage is that the estimation of performance such as rate of convergence is closer

to the experimental performance observed through simulations and experiments.

Let us consider the following definitions:

Definition 1. Let us consider Eqn. (1). We say that $Q(t)$ solves the **consensus problem** if $\lim_{t\to\infty} x_i(t) = \alpha$, $\forall i = 1,\dots,N$, where $x_i(t)$ is the i-th component of the vector $x(t)$. We say that $Q(t)$ solves the **average consensus problem** if in addition to the previous condition we have $\alpha = \frac{1}{N}\sum_{i=1}^{N} x_i(0)$. If $Q(t)$ is a random variable, then we say that Q solves the **probabilistic (average) consensus problem** if the limit above exists almost surely.

These definitions include a wide class of consensus strategies: strategies with a time–invariant matrix $Q(t) = Q$, deterministic time-varying strategies $Q(t)$, and randomized strategies where $Q(t)$ is drawn from a set of stochastic matrices $\mathscr{Q}$ according to a probability distribution. The next theorem describes some sufficient conditions which guarantee deterministic and probabilistic (average) consensus.

Theorem 1. *Let us consider the sequence of constant matrices $Q(t) = Q$. If the graph $\mathscr{G}_Q \in \mathbb{G}_{\text{sl}}$ and is rooted, then Q solves the consensus problem, and*

$$\lim_{t\to\infty} Q^t = \mathbf{1}\eta^T$$

where $\eta \in \mathbb{R}^N$ is the left eigenvector of Q for the eigenvalue one and has the properties $\eta_i \geq 0$ and $\mathbf{1}^T\eta = 1$. If $\mathscr{G}_Q$ is strongly connected, then $\eta_i > 0, \forall i$. If in addition Q is doubly-stochastic, then $\mathscr{G}_Q$ is strongly connected and Q solves the average consensus problem, i.e. $\eta = \frac{1}{N}\mathbf{1}$. Moreover, in all cases the convergence is exponential and its rate is given by the essential spectral radius $\mathtt{esr}(Q)$.

This theorem is well known and can be found in many textbooks on Markov chains such as in [53]. The assumption that $\mathscr{G}_Q \in \mathbb{G}_{\text{sl}}$ is not necessary to achieve consensus; for example consider $Q = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$, for which $x(t) = x_1(0)\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ for each $t \geq 1$ and $x(0) = [x_1(0)\ x_2(0)]^T$. However, some additional assumption besides $\mathscr{G}_Q$ being rooted is actually needed in order to guarantee consensus: for example $Q = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ is such that $\mathscr{G}_Q$ is rooted, but it gives $x(2t) = \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix}$ and $x(2t+1) = \begin{bmatrix} x_2(0) \\ x_1(0) \end{bmatrix}$ for all t. In this chapter, for the sake of simplicity, we will use the assumption that $\mathscr{G}_Q \in \mathbb{G}_{\text{sl}}$, also noting that this is a very mild requirement since it means that any agent can communicate to itself; however in some cases, such as in the de Bruijn graphs [24], it is useful to consider also graphs not in $\mathbb{G}_{\text{sl}}$.

Besides the results on constant matrices Q, much research has been devoted to the analysis of time-varying linear consensus which is addressed by the next theorem.

Theorem 2. *Consider the deterministic sequence of stochastic matrices $\{Q(t)\}_{t=0}^{+\infty}$ and the corresponding associated graphs $\mathscr{G}(t) = \mathscr{G}_{Q(t)}$. Suppose $\mathscr{G}(t) \in \mathbb{G}_{\text{sl}}, \forall t$.*

Then the sequence $Q(t)$ solves the consensus problem if and only if there exists a finite positive integer number T such that the graphs $\overline{\mathscr{G}}(\tau)$ obtained from the union of the graphs $\mathscr{G}(\tau)$ in the following way: $\overline{\mathscr{G}}(\tau) = \mathscr{G}(\tau) \cup \mathscr{G}(\tau+1) \cup \ldots \cup \mathscr{G}(\tau+T-1)$ with $\tau = 0,1,\ldots$ are all rooted. If in addition the matrices $Q(t)$ are all doubly-stochastic, then they solve the average consensus problem.

A simple proof of the previous theorem can be found in [41], but its roots can be tracked back at least to [61], and it has been rediscovered several times in the past years [33, 50, 8, 13]. The previous theorem states that it is not necessary for graphs associated to the matrices $Q(t)$ to be connected at all time, but only over a time window. This assumption basically guarantees that information travels, possibly with some delay, from at least one node to all other nodes infinitely many times. What is particularly remarkable in this theorem and also in Theorem 1, is that convergence is completely characterized by connectivity properties of the graphs $\mathscr{G}_{Q(t)}$, regardless of the specific values of the entries of the matrices $Q(t)$. On the other hand, the negative side is that the rate of convergence is hard to estimate since it is based on worst-case analysis. Therefore in general it is over-pessimistic and of little practical use. Recent work has tried to address this problem by finding tighter bounds on the rate of convergence while adding only general constraints on the topological properties of the graphs $\mathscr{G}_{Q(t)}$ and on the numerical values for the entries of $Q(t)$ [2].

A more recent approach to consensus is to model time-varying consensus in term of randomized strategies. The advantage of a randomized approach is to preserve simple convergence conditions based on graph properties while obtaining good estimates for the rate of convergence of typical realizations. The next theorem provides convergence conditions in a randomized context.

Theorem 3. *Consider a random i.i.d. sequence of stochastic matrices $\{Q(t)\}_{t=0}^{+\infty}$ drawn according to some probability distribution from the set $\mathscr{Q}$, and the stochastic matrix $\overline{Q} = \mathbb{E}[Q(t)]$. If the graphs $\mathscr{G}(t) = \mathscr{G}_{Q(t)} \in \mathbb{G}_{\mathrm{sl}}, \forall t$ and if $\mathscr{G}_{\overline{Q}}$ is rooted, then the sequence $Q(t)$ solves the probabilistic consensus problem. The rate of convergence in mean square sense defined as $\rho = \sup_{x(0)} \limsup_{t\to\infty} (\mathbb{E}[||x(t) - x(\infty)||^2])^{1/t}$ is bounded by*

$$(\mathtt{esr}(\overline{Q}))^2 \leq \rho \leq \mathtt{sr}(\mathbb{E}[Q^T(t)\Omega Q(t)])$$

where $\Omega := I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$ and $\mathtt{sr}(P)$ indicates the spectral radius of the matrix P, i.e. its largest eigenvalue in absolute value. If in addition $Q(t)$ are all doubly-stochastic, then they solve the probabilistic average consensus problem.

The proof of this theorem can be found in [26]. Similarly to the previous two theorems, even in a randomized scenario the convergence conditions are characterized in terms of graphs connectivity properties. In particular, it states that convergence is guaranteed if the graph is connected on average. However, differently from Theorem 2, the randomized framework provides tighter bounds on the rate of convergence. Another advantage of considering a randomized framework is the ability to model scenarios subject to random communication links or nodes failure.

There is a rich literature on randomized consensus that extends the results of the previous theorem. One direction is to find weaker convergence conditions, more

specifically by relaxing the hypothesis of i.i.d. sequences to ergodicity only [58]. Another direction is to add additional hypotheses on the matrices $Q(t)$ or on the set $\mathscr{Q}$ in order to improve the convergence bounds. For example, in [11] it was shown that if $Q(t)$ are symmetric and idempotent. i.e. $Q(t) = Q^T(t)$ and $Q^2(t) = Q(t)$, then the upper bound is given by $\mathtt{sr}(\mathbb{E}[Q^T(t)\Omega Q(t)]) = \mathtt{esr}(\overline{Q})$.

There is also a rich literature on the analysis of consensus under different scenarios. For example, there is an equivalent version of the consensus problem in continuous time given by

$$\dot{x} = A(t)x \tag{5}$$

where A is a *Metzler matrix*, i.e. a matrix whose off-diagonal elements are nonnegative and the row-sum is null, i.e. $A\mathbf{1} = 0$. This types of systems have been well characterized by Moreau [40]. For example, the opposite of a Laplacian matrix is a Metzler matrix, which implies that $\dot{x} = -Lx$ achieves consensus under general connectivity properties of the associated graph. The continuous time framework is particularly suitable for modeling flocking and vehicle dynamics [28, 52, 59].

Another research direction is concerned with convergence conditions for consensus with delayed information, i.e. for consensus whose dynamics can be written as

$$x_i(t+1) = \sum_{j=1}^{N} q_{ij} x_j(t - \tau_i(t)), \;\; i = 1, \ldots, N$$

where the delay $\tau_i(t)$ can be unknown and time-varying [46, 8, 7, 60, 54, 62]. The main finding is that consensus is very robust to delay, which is particularly important in networked systems where delay is unavoidable. This comes from the observation that the convex hull of the points $x_i(t)$ can only shrink or remain constant, and delay only marginally affects this property [41, 8].

Also much interest has been generated from consensus subject to quantization and in particular to quantized communication. In this context the dynamics can be written as

$$x_i(t+1) = \sum_{j=1}^{N} q_{ij} \mathtt{q}_d(x_j(t)), \;\; i = 1, \ldots, N$$

where $\mathtt{q}_d(\cdot) : \mathbb{R} \to \mathtt{Q}_d$ and $\mathtt{Q}_d$ is a finite or countable set. A typical example is $\mathtt{q}_d(x) = \lfloor x \rfloor$, where $\lfloor x \rfloor$ indicates the largest integer smaller than x. This problem is particularly challenging due to the fact that quantization acts similarly to noise, thus being particularly harmful since the consensus matrices $Q(t)$ are not strictly stable but always have an eigenvalue in one and convergence might not be guaranteed. Therefore, much effort has been given in finding quantization strategies and quantization functions that still guarantee consensus [37, 18, 29, 38, 36, 43].

Another interesting aspect is related to consensus subject to lossy communication, i.e. a scenario where communication scheduled between two nodes fails due to random interference or noise. This scenario naturally fits the randomized framework of Theorem 3, however it also requires the design of a compensation mechanism when a packet is lost. Different strategies have been proposed and studied [35, 27, 47]. For example a natural scheme is to compensate for the lost packets by

replacing the the lost value x_j from the transmitting node j with the self value x_i of the receiving node i, more formally:

$$x_i(t+1) = \Big(q_{ii} + \sum_{j=1, i\neq j}^{N} \big(1-\gamma_{ij}(t)\big)q_{ij}\Big)x_i + \sum_{j=1, i\neq j}^{N} \gamma_{ij}(t)q_{ij}x_j(t), \;\; i=1,\dots,N$$

where $\gamma_{ij}(t)$ is a random variable such that $\gamma_{ij}(t)=1$ if transmission at time t from node j to node i was successful, and $\gamma_{ij}(t)=0$ otherwise [27]. These works show that packet loss in general does not affect convergence to consensus, but it can reduce convergence rate and change the final consensus value as compared to ideal scenario with perfect communication, i.e. $\gamma_{ij}(t)=1, \forall i,j,t$.

A different setting is studied in [64], where additive noise is included in the consensus dynamics, i.e.

$$x(t+1) = Qx(t) + v(t)\,.$$

Note that, in all cases described above, noise affects the speed of convergence and the final value obtained (which is not the desired average), but does not prevent convergence. Differently, in the case when there is noise in the transmissions among nodes (without feedback), so that the messages sent by an agent are received by its neighbors corrupted by noises which might be different, and which are unknown to the sender, then convergence itself is an issue. The difficulty is in the design of a modified consensus algorithm capable of avoiding noise accumulation. Algorithms dealing with variations on this setting have been designed and analyzed by various authors, e.g. [49, 32, 34] (using time-varying weights in the consensus algorithm, to decrease the effect of neighbors' noise) and [16] (using error-correcting codes of increasing length to decrease the communication noise).

2.2 Design

Up to now, we provided a short overview of the properties of consensus algorithms under different scenarios and assumptions. However, in many engineering applications it is also very important to be able to design such algorithms. From a consensus design perspective, the design space is given by the communication graph $\mathscr{G} = \{\mathscr{N}, \mathscr{E}\}$ of a network of $N = |\mathscr{N}|$ agents, and the design problem consists in finding suitable $Q(t)$ compatible with $\mathscr{G}$ that achieve consensus or average consensus. We assume that the graph $\mathscr{G}$ includes self-loops, i.e. $\mathscr{G} \in \mathbb{G}_{\text{sl}}$, and that it is at least rooted.

There are two main approaches to design. The first focuses on local design methods which require only local information, i.e. each node can design its communication and consensus updates weights almost independently of the other nodes. Obviously, with this approach optimality with respect to some performance index is not guaranteed. The second approach focuses on methods which try to optimize some global performance index. As a consequence, this often leads to a centralized

optimization problem that strongly depends on the topology and might be suitable if the network static and has small size. We start by presenting these two approaches first within the context of static consensus, i.e. $Q(t) = Q$ and then in the context of time-varying consensus strategies.

2.2.1 Matrix design – static consensus: Q

If only consensus is required then a simple local strategy to design the matrix Q is given by:

$$q_{ij} = \frac{1}{d_{in}(i)+1}, \ \ (j,i) \in \mathscr{E}$$

Clearly $\mathscr{G}_Q = \mathscr{G}$, and Q is stochastic, thus satisfying hypotheses of Theorem 1.

Differently, if average consensus is required, various solutions are possible. If the graph is undirected a possible solution is to choose:

$$q_{ij} = \begin{cases} \varepsilon & \text{if } (j,i) \in \mathscr{E} \text{ and } i \neq j \\ 1 - \varepsilon d(i) & \text{if } i = j \end{cases} \tag{6}$$

where $\varepsilon < \frac{1}{\max_i d(i)}$. This matrix is clearly symmetric since the non-zero off-diagonal terms are all equal and positive $q_{ij} = q_{ji} = \varepsilon, \forall i, j$. The condition on ε is necessary to guarantee that all diagonal terms are positive. As a consequence, Q is a stochastic symmetric matrix, therefore it is also doubly-stochastic. Moreover $\mathscr{G}_Q = \mathscr{G}$ and by hypothesis $\mathscr{G}$ is rooted[1], thus satisfying hypotheses of Theorem 1. Note that this matrix is strongly related to the Laplacian matrix L of the graph $\mathscr{G}$. In fact, consider the discretized dynamics of Eqn. (5) where $A = -L$ with time step ε, i.e. $x(t+1) = e^{-\varepsilon L}x(t) = Q_d x(t)$, then the first order expansion of Q_d, i.e. $Q_d = I - \varepsilon L + O(\varepsilon)$, has the same structure of the Q given by Eqn. (6).

Another possible strategy for undirected graphs is based on the Metropolis-Hastings weights:

$$q_{ij} = \begin{cases} \frac{1}{\max(d(i),d(j))+1} & \text{if } (j,i) \in \mathscr{E} \text{ and } i \neq j \\ 1 - \sum_{j=1, i\neq j}^{N} q_{ij} & \text{if } i = j \end{cases} \tag{7}$$

Clearly the matrix Q is symmetric and the diagonal elements are strictly positive since $q_{ii} = 1 - \sum_{j=1, i\neq j}^{N} q_{ij} \geq 1 - \sum_{j=1, i\neq j, (i,j)\in\mathscr{E}}^{N} \frac{1}{d(i)+1} = 1 - \frac{d(i)}{d(i)+1} = \frac{1}{d(i)+1} > 0$, therefore Q is doubly-stochastic and $\mathscr{G}_Q = \mathscr{G}$ which are sufficient conditions to guarantee average consensus. As compared to the strategy based on the Laplacian of Eqn. (6), the strategy based on the Metropolis weights of Eqn. (7) is local, i.e. each node requires only the knowledge of local information, namely the degrees of its neighbors, while the former requires the knowledge of an upper bound on the degree of all nodes of the network. Moreover, the Metropolis-based consensus matrix has in general faster convergence rate than the Laplacian-based consensus matrix.

[1] If an undirected graph is rooted, then it is also strongly connected.

If the communication graph $\mathscr{G}$ is directed, then the design of a consistent doubly-stochastic matrix is not trivial. A possible strategy is based on the design of a doubly-stochastic matrix based on a convex combination of permutation matrices, where a permutation matrix P is defined as $P \in \{0,1\}^{N\times N}, \mathbf{1}^T P = \mathbf{1}^T, P\mathbf{1} = \mathbf{1}$. Note that a permutation matrix is doubly-stochastic. This procedure is basically an application of the Birkhoff's Theorem [39]. We start from the assumption that the graph is strongly connected. This implies that for each edge $e = (j,i) \in \mathscr{E}$ there exists a path connecting node i to node j, which in turns implies there exists at least one simple cycle $\mathscr{C}$ in the graph including the edge e, i.e. there exists a sequence of non repeated vertices $\ell_1, \ell_2, \ldots, \ell_L \in \mathscr{N}$ such that $\ell_1 = i, \ell_L = j$, $(\ell_i, \ell_{i+1}) \in \mathscr{E}$ for $i = 1, \ldots, L-1$ and $(\ell_L, \ell_1) \in \mathscr{E}$. Associated to this cycle it is possible to define a permutation matrix P_e as follows:

$$\begin{array}{ll} [P_e]_{\ell_r \ell_{r+1}} = 1 & \text{for } r = 1, \ldots, L-1 \\ [P_e]_{\ell_L \ell_1} = 1 & \\ [P_e]_{kk} = 1 & \text{for } k \neq \ell_r,\ r = 1, \ldots, L \\ [P_e]_{hk} = 0 & \text{otherwise} \end{array}$$

Clearly $\mathscr{G}_{P_e} \subseteq \mathscr{G}$. According to this procedure it is always possible to find M cycles in the graph $\mathscr{G}$ and permutation matrices $P_i, i = 1, \ldots, M$ constructed as above, that includes all edges of the graphs. Let us consider now the matrix $Q = a_0 I + \sum_{i=1}^{M} a_i P_i$ where $a_i > 0, \forall i = 0, \ldots, M$ and $\sum_{i=0}^{M} a_i = 1$, then Q is still doubly-stochastic since it is a convex combination of doubly-stochastic matrices. Also since all edges of $\mathscr{G}$ are included in Q, then $\mathscr{G}_Q = \mathscr{G}$. These two facts guarantee that Q achieves average consensus.

However, this procedure is rather tedious and requires global knowledge of the graph topology. There is an elegant alternative solution to achieve average consensus [1], which requires only local knowledge of the graph topology. Let us consider the matrix Q designed as follows:

$$q_{ij} = \frac{1}{d_{out}(j) + 1}, \quad (j,i) \in \mathscr{E}$$

This matrix is column-stochastic, i.e. its transpose is stochastic ($Q^T\mathbf{1} = \mathbf{1}$), and $\mathscr{G}_Q = \mathscr{G}$ is strongly connected. This implies by Theorem 1 that $\lim_{t\to\infty} Q^t = \lim_{t\to\infty}((Q^T)^t)^T = (\mathbf{1}\rho^T)^T = \rho\mathbf{1}^T$ where $\rho_i > 0, \forall i$. Now let us consider $z(t+1) = Qz(t)$ and $w(t+1) = Qw(t)$ where the initial condition are $z(0) = x(0)$ and $w(0) = \mathbf{1}$, and the $x(t)$ such that $x_i(t) = \frac{z_i(t)}{w_i(t)}$. From $\lim_{t\to\infty} Q^t = \rho\mathbf{1}^T$, it follows that $\lim_{t\to\infty} z(t) = \big(\sum_{i=1}^{N} z_i(0)\big)\rho = \big(\sum_{i=1}^{N} x_i(0)\big)\rho$ and $\lim_{t\to\infty} w(t) = \big(\sum_{i=1}^{N} w_i(0)\big)\rho = N\rho$, therefore $\lim_{t\to\infty} x_i(t) = \frac{\rho_i\left(\sum_{i=1}^{N} x_i(0)\right)}{\rho_i N} = \frac{1}{N}\sum_{i=1}^{N} x_i(0)$ as desired. Note that average consensus is achieved through a nonlinear algorithm that uses two parallel linear iterative updates very similar to standard consensus. The weak point of this approach is that perfect communication is required since the algorithm can become unstable if lossy links are considered.

So far, we just considered design strategies to achieve consensus or average consensus, but we did not discuss about their rate of convergence. Design of consensus

algorithms with fast rate of convergence is not a trivial task. If simple consensus is required, there is a simple strategy that achieves in a finite number of steps. Given a rooted graph, it is always possible to find a tree that connects one node, namely the root, to all other nodes. Without loss of generality, assume that the root is node $i = 1$, and let us consider only the set of directed edges associated with this tree, i.e. $\mathscr{E}_{\text{tree}} \subseteq \mathscr{E}$. Note that $\mathscr{E}_{\text{tree}}$ does not contain self-loops. Let us consider the matrix Q designed as follows:

$$q_{11} = 1, \;\; q_{ij} = 1 \;\; (j,i) \in \mathscr{E}_{\text{tree}}, j \neq 1$$

Clearly the matrix is stochastic and it is not difficult to see that $Q^t = \mathbf{1}[1\ 0\ \cdots\ 0]$ for $t \geq \ell$, i.e. $x_i(t) = x_1(0)$ for $t \geq \ell$, where ℓ is the maximum distance of all nodes from the root. This implies that $\texttt{esr}(Q) = 0$. In other words, each node sets the value of its variable $x_i(t)$ to the value received from its parents, therefore after a finite number of steps all nodes will have a copy of the initial condition of the root. This gives very fast convergence rate even for very large networks, as long as the diameter, i.e. the largest path distance within any two nodes, is small.

If average consensus is required, then the previous strategy is obviously not suitable. Optimal design of Q in terms of fast rate of convergence is not trivial in directed graph. If the graph is undirected, then it has been shown by Xiao et al. [63] that finding a symmetric stochastic matrix consistent with the graph with smallest `esr` is a convex problem. i.e.

$$\begin{aligned} \min_{Q} \;\; & \texttt{esr}(Q) \\ \text{s.t.} \;\; & Q = Q^T, Q\mathbf{1} = \mathbf{1}, [Q]_{ij} \geq 0, \mathscr{G}_Q = \mathscr{G} \end{aligned}$$

Actually the non-negativeness constraint on the elements of Q is not necessary to have a convex problem, and therefore can be removed, thus providing a matrix Q with possible negative entries which can lead to an even smaller `esr`. On the other hand, this is a centralized optimization problem, and the whole topology of the network is needed to find the optimal solution. Local optimization strategies to minimize the `esr` are still an open area of research.

2.2.2 Matrix design – dynamic consensus: $Q(t)$

Now, we address the problem of designing dynamic consensus strategies where the consensus matrix is not constant but can change over time. The major drawback of static consensus is that it requires some sort of synchronization among all nodes of the network. In fact, between one iteration and the subsequent iteration, nodes need to exchange information and then update their local variables simultaneously. This can be difficult to enforce or simply too costly. Therefore, there is much interest in designing consensus strategies that require little coordination and synchronization among nodes. These algorithms are also referred as *asynchronous algorithms*. Some of the most popular asynchronous strategies are motivated by practical consideration

based on the communication schemes that can be implemented on networks. These include broadcast [3], asymmetric gossip [25] and symmetric gossip [11].

In the *broadcast* scheme, one node i transmits its information to all its neighbors $\mathcal{V}_{out}(i)$, and each receiving node updates its local variable using consensus. More formally, given a possibly directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, then $Q(t) \in \mathbb{Q}_B = \{Q_1, Q_2, \ldots, Q_N\}$, where $N = |\mathcal{N}|$ and

$$Q_i = I - w \sum_{j \in \mathcal{V}_{out}(i)} e_j (e_j - e_i)^T$$

where $w \in (0,1)$, I is the identity matrix of dimension N, and $e_i \in \mathbb{R}^N$ is a vector of all zeros except for the i-th entry which is set to one. Clearly all Q_i are stochastic, have self-loops, and $\mathcal{G}_{Q_i} \subseteq \mathcal{G}$.

Differently, in the *asymmetric gossip* one node i selects only one of its possible neighbors $\mathcal{V}_{out}(i)$, which after receiving the message updates its local variable. More formally, given a possibly directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, then $Q(t) \in \mathbb{Q}_{AG} = \{Q^{ij} \mid (i,j) \in \mathcal{E}, i \neq j\}$, where

$$Q^{ij} = I - w e_j (e_j - e_i)^T$$

where $w \in (0,1)$ and e_i are defined as above. Clearly all Q^{ij} are stochastic, have self-loops, and $\mathcal{G}_{Q^{ij}} \subseteq \mathcal{G}$. Note that even if the graph $\mathcal{G}$ is undirected, than the matrices Q^{ij} are only stochastic and do not guarantee average consensus. The same consideration applies to the broadcast matrices Q_i defined above.

The *symmetric gossip* is applicable only to undirected graphs. In this scheme, one node i transmits its information to only one of its neighbors j, which in turn transmits back to the node i another message with its local value. Only after the completion of this procedure the two nodes update their local values using a consensus scheme based on the same weight w. More formally, given the undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, then $Q(t) \in \mathbb{Q}_{SG} = \{Q^{ij} \mid (i,j) \in \mathcal{E}, i \neq j\}$, where

$$Q^{ij} = I - w(e_j - e_i)(e_j - e_i)^T$$

Clearly all Q^{ij} are doubly-stochastic, are idempotent (i.e., $(Q^{ij})^2 = Q^{ij}$), have self-loops, and $\mathcal{G}_{Q^{ij}} \subseteq \mathcal{G}$. Although symmetric gossip is somewhat more complex from a communication point of view, differently from broadcast and asymmetric gossip, it has the advantage to preserve the average at any time instant, therefore convergence to consensus automatically guarantees convergence to average consensus.

At this point, the design problem is how to select a sequence of $Q(t)$ from the sets defined above for the broadcast, asymmetric gossip and symmetric gossip, and how to choose the consensus weight w. In general the consensus weight is set to $w = 1/2$ and more attention is paid on the drawing of matrices $Q(t)$. One approach is to deterministically select these matrices according to some sequence, however this still requires some sort of coordination and synchronization. A more natural approach is to select these matrices randomly, possibly according to some i.i.d. distribution on the sets $\mathbb{Q}$. This distribution can be represented by a vector $p \in \mathbb{R}^N$, such that

$p \geq 0$ and $\mathbf{1}^T p = 1$ for the broadcast model, where $p_i = \mathbb{P}[Q(t) = Q_i]$. Similarly, the probability distribution in the symmetric and asymmetric gossip can be represented by a matrix $P \in \mathbb{R}^{N \times N}$ which is nonnegative, i.e. $[P]_{ij} \geq 0$, is consistent with the graph, i.e. $\mathscr{G}_P \subseteq \mathscr{G}$, and sum to unity, i.e. $\mathbf{1}^T P \mathbf{1} = 1$, where $[P]_{ij} = \mathbb{P}[Q(t) = Q^{ij}]$. In this case, the design space corresponds to the probability distribution of these sets, i.e. the vector p or the matrix P. The proper framework to analyze these strategies is given by Theorem 3. Many results about exact rate of convergence and its optimal design are available for communications graphs $\mathscr{G}$ that present special symmetries like complete graphs, circulant graphs, hypercubes, and tori [17, 26]. Differently, for general undirected graphs, Boyd et al. [11] showed that under the randomized symmetric gossip schemes with weight $w = 1/2$, the rate of convergence can be bound by $\rho \leq \mathtt{esr}(\overline{Q})$ thus suggesting the following optimization criteria for maximizing the rate of convergence:

$$\begin{aligned} &\min_{P} \ \mathtt{esr}(\overline{Q}) \\ &\text{s.t. } \ \overline{Q} = \sum_{i=1}^{N} \sum_{j=1}^{N} [P]_{ij} Q^{ij}, \ [P]_{ij} \geq 0, \mathbf{1}^T P \mathbf{1} = 1, \ \mathscr{G}_P \subseteq \mathscr{G} \end{aligned}$$

which turns out to be a convex problem. This optimization problem is a centralized problem, however the authors in [11] suggested also suboptimal decentralized optimization schemes. Fagnani et al. [25] studied the asymmetric gossip for general undirected graphs and showed that rate of convergence can be bound by $\rho \leq \mathtt{sr}([Q^T(0)\Omega Q(0)]) = 1 - 2w\big((1-w) - wN^{-1}\big)\mu$, where μ is the smallest positive eigenvalue of the positive semidefinite matrix $S = \mathrm{diag}(P\mathbf{1}) - (P + P^T)/2$, where $\mathrm{diag}(x) : \mathbb{R}^n \to \mathbb{R}^{n \times n}$ indicates a diagonal matrix whole diagonal entries are the entries of the vector x. Therefore in this scenario a possible optimization criterium for minimizing the rate of convergence is to minimize ρ which is minimized by setting $w = \frac{1}{2}\frac{N+1}{N} \approx \frac{1}{2}$ and by maximizing μ. If we restrict to symmetric probability matrices $P = P^T$, maximizing μ is equivalent to the following convex optimization problem:

$$\begin{aligned} &\max_{P,\varepsilon} \ \varepsilon \\ &\text{s.t. } \ \mathrm{diag}(P\mathbf{1}) - P \geq \varepsilon I, \ P = P^T, \ [P]_{ij} \geq 0, \mathbf{1}^T P \mathbf{1} = 1, \ \mathscr{G}_P \subseteq \mathscr{G} \end{aligned}$$

Similarly to [11] also this optimization problem is centralized and therefore might not be suitable for fully distributed optimization.

2.2.3 Graph design

In the previous sections, we focused on the issue of how to design the coefficients of the matrix Q for a given communication graph $\mathscr{G}$. However, there are scenarios for which also the communication graph can be designed, therefore it is useful to understand the effect of the graph topology on the performance and how it scales as

the number of nodes increases. Also, it is important to note that, in many cases, the effect of the graph topology on performance is much more relevant than the actual choice of the weights, i.e. of the non-zero entries of Q. In fact, for example, Xiao et al. [64] studied consensus over random geometric graphs [48] and compared optimal design with suboptimal decentralized strategies like the consensus based on the Metropolis matrix, showing that performance difference was not so drammatically different and seemed to scale similarly with the graph size.

In this context, let us consider the static consensus $x(t+1) = Qx(t)$. Asking what graph allows for the fastest convergence, without any further constraint, is trivially answered (the complete graph, i.e. every pair of nodes is connected by an edge) and is not very meaningful: the complete graph corresponds to centralized computation. A more interesting question is asked by Delvenne et al. [23, 24]: what is the best graph, under the constraint that each agent receives at most ν messages at each iteration (i.e., $\mathscr{G}_Q$ has bounded in-degree)? The answer is given by a family of graphs known as *de Bruijn graphs*, well-known in the computer science literature for their expansion properties, and capable of giving the exact average in finite time (not only $\lim_{t\to\infty} x(t) = \frac{1}{N}\mathbf{1}^T x(0)$, but also $x(\bar{t}) = \frac{1}{N}\mathbf{1}^T x(0)$ for some $\bar{t}$), and moreover the time $\bar{t}$ is the smallest possible with the constraint on the in-degree.

The very good performance of de Bruijn graphs is surprising if compared with a family of graphs, *Abelian Cayley graphs* [17], which are grids on d-dimensional tori (a circle for $d = 1$), and whose algebraic structure (a generalization of circulant matrices) allows to compute the eigenvalues and to prove that $\mathtt{esr}(Q) \geq 1 - cN^{\frac{1}{\nu+1}}$, where ν is the degree of the nodes and c is a positive scalar independent of the graph. This proves that, when $N \to \infty$, $\mathtt{esr}(Q) \to 1$, i.e., convergence is considerably slowed down by the size of the network. However, this is not always the case: in addition to de Bruijn graphs, there are other significant classes of graphs, known as *expander graphs*, such that $\mathtt{esr}(Q)$ is bounded away from 1 when $N \to \infty$ (see [45] for the study of such graphs in the context of consensus algorithms). A particular family of graphs which allow fast information transfer (having a small diameter despite the small degree of each node) are the so-called *small-world* graph, which are considered as a reasonable model for many social interactions (e.g., the collaboration graph for scientific authors, or the spread of some diseases) and for the world-wide web; they have been studied in the consensus literature by Olfati-Saber [44] and Tahbaz-Salehi et al. [57].

All such graphs have good properties in terms of fast convergence, despite the small (average) number of neighbors of each node, and as opposed to Abelian Cayley graphs (roughly speaking: grids) where convergence is very slow for large networks. The key fact that makes this difference is that in grids not only the number of neighbors is little, but also their position is forced to be local, in a somehow geometrical sense. In many practical deployments of sensor networks, geometrical constraints are indeed present, and thus the very structured and symmetrical Abelian Cayley graphs can be thought as an idealized version of realistic settings, and are important in that they underline the strong limitations that such locality constraint has on performance and gives guidelines for the design of the number of nodes in the network, in the case when the topology is bound to have such a given struc-

ture and the size only is the objective of design. A step towards a more realistic, less structured family of graphs where geometrical bounds are enforced is the study of *random geometric graphs* [48]. Random geometric graphs are undirected graphs which are widely used to model wireless sensor networks, and they are obtained by randomly generating points in the Euclidean space (usually, in the plane) according to a Poisson point process (the number of points in any bounded region is a Poisson random variable with average proportional to the area, and the position of points is uniformly distributed in the region) and then drawing an edge between two nodes if and only if their relative distance is smaller than a predefined communication radius r.

The analysis of the effect of the graph topology on performance has been considered also for time-varying consensus algorithms, and particularly for randomized algorithms (as opposed to the previously-mentioned results, where families of random graphs were considered in the sense that the one time-invariant graph is randomly selected before starting the algorithm). An early work by Hatano et al. [31] studies the case where, at each time step, the graph is chosen randomly according to the Erdős-Rényi model, i.e., the presence or absence of edges between any pair of nodes are given by i.i.d. Bernoulli random variables. A more recent research line has studied convergence of various randomized gossip algorithms, when the random activation of a node or of an edge is restricted to an underlying graph smaller than the complete graph. In this context, a relevant result by Fagnani et al. [26] concerns the rate of convergence of various algorithms (including symmetric, asymmetric and broadcast gossip) when the underlying graph is an Abelian Cayley graph. Another very interesting result can be found in [11], where the rate of convergence of symmetric gossip is found for random geometric graphs and compared to the faster convergence in the preferential connectivity model (a popular model for the graph of the world wide web, and an example of small-world graph).

3 Estimation and control problems as average consensus

In this section we illustrate with few examples that some estimation and control problems can be reframed as the computation of the average of some quantities, which therefore can be efficiently computed in a distributed fashion using average consensus algorithms.

3.1 Parameter estimation with heterogeneous sensors

Let us consider N sensors that measure a noisy version of the true parameter $\theta \in \mathbb{R}$ as follows:

$$y_i = \theta + v_i, \;\; v_i \sim \mathcal{N}(0, \sigma_i^2), \;\; i = 1, \ldots, N$$

where v_i are independent zero-mean random variable with covariance σ_i^2, i.e. sensors have different accuracy. The minimum-variance estimate of the parameter θ, given all the measurements, is given by:

$$\hat{\theta}_{\mathrm{MV}} = \sum_{i=1}^{N} \alpha_i y_i, \qquad \alpha_i = \frac{\frac{1}{\sigma_i^2}}{\sum_{j=1}^{N} \frac{1}{\sigma_j^2}}$$

i.e. it is a convex combination of the measurements. It is easy to see that the previous estimator can be written as:

$$\hat{\theta}_{\mathrm{MV}} = \frac{\frac{1}{N}\sum_{i=1}^{N} \frac{1}{\sigma_i^2} y_i}{\frac{1}{N}\sum_{j=1}^{N} \frac{1}{\sigma_j^2}}$$

i.e. it is the ratio of two averages. Therefore, it can be asymptotically computed in a distributed fashion using two average consensus algorithms in parallel whose initial condition are set to $x_i^y(0) = \frac{1}{\sigma_i^2} y_i$ and $x_i^\sigma(0) = \frac{1}{\sigma_i^2}$, so that

$$\lim_{t\to+\infty} \hat{\theta}_i(t) := \frac{x_i^y(t)}{x_i^\sigma(t)} = \hat{\theta}_{\mathrm{MV}}, \ \ \forall i.$$

3.2 Node counting in a network

In many applications it is important to know how many nodes there are in a network. This can be easily computed via an average consensus algorithm, by setting all the initial conditions to zero except for one node, i.e. $x_1(0) = 1$ and $x_i(0) = 0, i = 2, \ldots, N$. Since average consensus guarantees converge to the average of initial conditions, an asymptotically correct estimator of the total number of node N is given by:

$$\hat{N}_i(t) := \frac{1}{x_i(t)},$$

because

$$\lim_{t\to+\infty} \hat{N}_i(t) = \lim_{t\to\infty} \frac{1}{x_i(t)} = \frac{1}{\frac{1}{N}\sum_{j=1}^{N} x_i(0)} = N, \ \ \forall i.$$

3.3 Generalized averages

Besides the common arithmetic average it is also possible to compute other types of averages such as

$$z_\alpha = \sqrt[\alpha]{\frac{1}{N}\sum_{i=1}^{N} y_i^\alpha}$$

where $\alpha = 1$ gives rise to the usual arithmetic average, $\alpha = 2$ the mean square, $\alpha = -1$ the harmonic mean. Also note that $z_\infty := \lim_{\alpha\to+\infty} z_\alpha = \max_i y_i$ [6, 21]. These generalized averages can be computed using average consensus by setting the initial condition $x_i(0) = y_i^\alpha$ and computing an estimate of the desired average as follows:

$$\lim_{t\to+\infty} \hat{z}_i(t) := \sqrt[\alpha]{x_i(t)} = z_\alpha, \ \ \forall i$$

Another important average is the geometric mean defined as:

$$z_g = \sqrt[N]{\prod_{i=1}^{N} y_i}$$

The geometric mean can be written as $z_g = \exp(\log z_g) = \exp\left(\sum_{i=1}^{N} \log y_i\right)$, therefore it can be computed using average consensus by setting the initial conditions to $x_i(0) = \log y_i$ and the following estimator:

$$\lim_{t\to+\infty} \hat{z}_i(t) := \exp(N x_i(t)) = z_g, \ \ \forall i$$

Note, however, that in this case the number of nodes N needs to be known in advance.

3.4 Vehicle rendezvous

An important example of vehicle formation control is the rendezvous problem (see e.g. [12]), where all vehicles are required to meet at a common location using only relative position information for all initial conditions. In its simplest formulation, the vehicle dynamics is given by

$$x_i(t+1) = x_i(t) + u_i(t)$$

and the goal is to find a linear control strategy which uses only relative distance information, i.e.

$$u_i(t) = \sum_{j=1}^{N} q_{ij}(t)(x_j(t) - x_i(t))$$

such that $\lim_{t\to+\infty} x_i(t) = \bar{x}$ for some $\bar{x}$. This is indeed a consensus problem that can be solved by choosing the weights $q_{ij}(t)$ that guarantees convergence[2]. Besides

[2] In realistic scenarios the gains q_{ij} are a function of vehicle location, i.e. $q_{ij} = q_{ij}(x)$. A typical model is to consider limited communication range $r > 0$, i.e. $q_{ij} = 0$ if $|x_i - x_j| > r$. This gives rise to nonlinear dynamics which is not captured by the model presented in Section 2. The analysis

convergence, it is also relevant to compute performance of the rendezvous strategy. A natural approach is to consider a linear quadratic (LQ) measure given by:

$$J_{\mathrm{LQ}} = J_x + \varepsilon J_u = \sum_{t=0}^{\infty} ||x(t) - x(\infty)||^2 + \varepsilon \sum_{t=0}^{\infty} ||u(t)||^2$$

where $x = [x_1\ x_2\ \cdots\ x_N]^T$, $u = [u_1\ u_2\ \cdots\ u_N]^T$, and ε is a positive scalar that trades off the integral square error of all vehicles from the rendezvous location $x(\infty) = \bar{x}\mathbf{1}$, namely J_x, versus the integral energy of all vehicles required to achieve consensus, namely J_u.

3.5 Least Squares Data Regression

Least squares are one of the most popular estimation techniques in data regression, where the objective is to estimate a function $y = f(x)$, from a noisy data set $\mathscr{D} = \{(x_i, y_i)\}_{i=1}^N$. A standard approach is to propose a parametrized function $f_\theta(x) := \sum_{j=1}^M \theta_i g_i(x)$, where $g_i(x)$ are known functions, often called basis functions, and $\theta_i, i = 1\ldots, M$ are unknown parameters to be determined based on the data set $\mathscr{D}$. The least squares estimate of the parameter vector $\theta = [\theta_1\ \theta_2\ \cdots\ \theta_M]^T$ is defined as

$$\hat{\theta}_{\mathrm{LS}} = \arg\min_{\theta} \sum_{i=1}^{N} (y_i - f_\theta(x_i))^2$$

If we define the vectors $g_i = [g_1(x_i)\ g_2(x_i) \cdots\ g_M(x_i)]^T \in \mathbb{R}^M, i = 1, \ldots, N$, $y = [y_1\ y_2\ \cdots\ y_M]^T \in \mathbb{R}^N$, and the matrix $G = [g_1\ g_2\ \cdots\ g_M]^T \in \mathbb{R}^{N\times M}$, then we have

$$\begin{aligned}\hat{\theta}_{\mathrm{LS}} &= \arg\min_{\theta} \|y - G\theta\|^2 = (G^T G)^{-1} G^T y = \left(\sum_{i=1}^{N} g_i g_i^T\right)^{-1} \left(\sum_{i=1}^{N} g_i y_i\right) \\ &= \left(\frac{1}{N}\sum_{i=1}^{N} g_i g_i^T\right)^{-1} \left(\frac{1}{N}\sum_{i=1}^{N} g_i y_i\right)\end{aligned}$$

under the implicit assumption that $(G^T G)^{-1}$ exists. From last equation it is clear that the estimate can be computed as a nonlinear combination of two averages, therefore a consensus based strategy is to run two average consensus algorithms with initial conditions $x_i^{gg}(0) = g_i g_i^T \in \mathbb{R}^{M\times M}$ and $x_i^{gy}(0) = g_i y_i \in \mathbb{R}^M$, and then asymptotically computing the least square estimate as:

$$\lim_{t\to+\infty} \hat{\theta}_i(t) := \left(x_i^{gg}(t)\right)^{-1} x_i^{gy}(t) = \hat{\theta}_{\mathrm{LS}}, \quad \forall i$$

---

of these systems is beyond the scope of this work and we refer the interested reader to [22] an references therein.

Note that in this scenario x_i^{gg} are matrices and x_i^{gy} are vectors, therefore they are not scalar as usually considered in Eqn. (1), however all results of Section 2 still apply by considering the local updates rules of Eqn. (2) or Eqn. (3) [65, 9].

3.6 Sensor calibration

Often inexpensive sensors might be affected by unknown offsets due to fabrication imperfections or aging. A common example is given by the sensor that measures the signal strength, the RSSI, in the radio chip of commercial wireless sensor nodes [9]. The RSSI is often used to estimate the relative distance between two of these wireless nodes for localization and tracking applications. More precisely the signal strength y_{ij} measured by node i from node j can be modeled as:

$$y_{ij} = f(\xi_i, \xi_j) + o_i$$

where ξ_i and ξ_j are the locations of the receiving node i and the transmitter node j, respectively, and o_i is the offset of the receiving node. Typically, $f(\xi_i, \xi_j)$ is a function of the distance $\|\xi_i - \xi_j\|$ only, but in indoor environments this cannot be the case. However, it still holds that

$$f(\xi_i, \xi_j) = f(\xi_j, \xi_i),$$

i.e. the function f is symmetric in terms of nodes locations. The objective of calibration is to estimate the offset o_i for each node in order to remove it from the measurements. This is clearly impossible, unless at least one node is calibrated or if the function f and the node locations ξ are known. A less demanding requirement is to find offset estimates $\hat{o}_i$ such that $o_i - \hat{o}_i = \bar{o}$ for all i, i.e. to be able to have all nodes with the same offset $\bar{o}$. This can be interpreted as a consensus problem on the variable $x_i(t) = o_i - \hat{o}_i(t)$. However, this is still an undetermined problem since $\bar{o}$ is arbitrary. One solution to remove this ambiguity is to choose one node as a reference, for example node $i = 1$, i.e. $\bar{o} = o_1$. A less arbitrary choice is to find $\bar{o}$ such that

$$\arg\min_{\bar{o}} \sum_{i=1}^{N} \hat{o}_i^2 = \arg\min_{\bar{o}} \sum_{i=1}^{N} (o_i - \bar{o})^2 = \frac{1}{N} \sum_{i=1}^{N} o_i = \frac{1}{N} \sum_{i=1}^{N} x_i(0)$$

where the last equality is obtained by setting $\hat{o}_i(0) = 0$. This strategy, which aims at minimizing the magnitude of offset compensation terms $\hat{o}_i$, implies that average consensus is to be sought. By substituting $x_i(t) = o_i - \hat{o}_i(t)$ into Eqn. (3) we get:

$$
\begin{aligned}
o_i - \hat{o}_i(t+1) &= o_i - \hat{o}_i(t) + \sum_{j=1}^{N} q_{ij}(t)\big(o_j - \hat{o}_j(t) - (o_i - \hat{o}_i(t))\big) \\
\hat{o}_i(t+1) &= \hat{o}_i(t) - \sum_{j=1}^{N} q_{ij}(t)\big(f_{ji} + o_j - \hat{o}_j(t) - (f_{ij} + o_i - \hat{o}_i(t))\big) \\
&= \hat{o}_i(t) + \sum_{j=1}^{N} q_{ij}(t)\big(\hat{o}_j(t) - \hat{o}_i(t) + y_{ij} - y_{ji}\big)
\end{aligned}
$$

where we used the notation $f(\xi_i,\xi_j) = f_{ij}$ and the assumption that $f_{ij} = f_{ji}$. From average consensus we have that:

$$
\lim_{t\to+\infty} \hat{o}_i(t) = o_i - \frac{1}{N}\sum_{j=1}^{N} o_i
$$

From this expression, it is clear that if the offset are normally distributed, i.e. $o_i \sim \mathcal{N}(0,\sigma^2)$, then $\lim_{N\to+\infty} |\hat{o}_i(\infty) - o_i| = 0$ almost surely, i.e. if the number of nodes is large, then the offset estimate is very close to the true offset.

3.7 Kalman Filtering

Estimation of dynamical systems is another important area. Let us consider the following dynamical systems observed by N sensors:

$$
\begin{aligned}
\xi(t+1) &= A\xi(t) + w(t) \\
y_i(t) &= C_i\xi(t) + v_i(t), \quad i = 1,\dots,N
\end{aligned}
$$

where $w(t) \sim \mathcal{N}(0,Q)$ and $v_i(t) \sim \mathcal{N}(0,R_i)$ are uncorrelated white Gaussian noises. If we define the new vectors $y(t) = [y_1(t)\ y_2(t)\ \cdots\ y_N(t)]^T$ and $v(t) = [v_1(t)\ v_2(t)\ \cdots\ v_N(t)]^T$. The minimum error covariance estimate is given by $\hat{\xi}(h|t) := \mathbb{E}[\xi(h)\,|\,y(t),y(t-1)\dots y(1)]$ and its error variance is $P(h|t) := \mathrm{Var}(\xi(h) - \hat{\xi}(h|t))$. The optimal estimator is known as the Kalman Filter, whose equations are given by:

$$
\begin{aligned}
\hat{\xi}(t|t-1) &= A\hat{\xi}(t-1|t-1) \\
P(t|t-1) &= AP(t-1|t-1)A^T + Q \\
\hat{\xi}(t|t) &= \hat{\xi}(t|t-1) + P(t|t-1)C^T(CP(t|t-1)C^T + R)^{-1}(y(t) - C\hat{\xi}(t|t-1)) \\
P(t|t) &= P(t|t-1) - P(t|t-1)C^T(CP(t|t-1)C^T + R)^{-1}CP(t|t-1)
\end{aligned}
$$

The first two equations are known as the prediction step, while the last two equations are known as the correction step. Using the matrix inversion lemma, the correction step can be written as

$$\begin{aligned}
\hat{\xi}(t|t) &= P(t|t)(P(t|t-1)\hat{\xi}(t|t-1)+C^TR^{-1}y(t)) \\
&= P(t|t)(P(t|t-1)\hat{\xi}(t|t-1)+\sum_{i=1}^{N}C_i^TR_i^{-1}y_i(t)) \\
&= P(t|t)(P(t|t-1)\hat{\xi}(t|t-1)+z(t)) \\
P(t|t) &= (P(t|t-1)+C^TR^{-1}C)^{-1} = (P(t|t-1)+\sum_{i=1}^{N}C_i^TR_i^{-1}C_i)^{-1} \\
&= (P(t|t-1)+Z)^{-1}
\end{aligned}$$

which are also known as the inverse covariance filter. From these equations it is evident that the sufficient statistics necessary to recover the centralized Kalman filter are the quantities $z(t) = N(\frac{1}{N}\sum_{i=1}^{N}C_i^TR_i^{-1}y_i(t))$ and $Z = N(\frac{1}{N}\sum_{i=1}^{N}C_i^TR_i^{-1}C_i)$ which are averages of local quantities. Therefore, a possible strategy to run a local filter on each local node, which, between two measurements $y(t-1)$ and $y(t)$, runs m iterations of the average consensus algorithm to recover $z(t)$ and Z, and then updates its estimate using the centralized Kalman gain. If m is sufficiently large and if the total number of nodes N is known to each sensor, then each local filter coincides with the centralized Kalman filter [55]. If m is not sufficiently large to guarantee that the consensus has converged, then performance of the local filters needs to evaluated and also the consensus algorithms design should be designed accordingly to improve it. In this context [14], if scalar dynamics is considered, i.e. $\xi \in \mathbb{R}$ where $A = C_i = 1, \forall i$, $Q = q$, and $R = r$, then the equations for the consensus-based Kalman filter can be written as

$$\begin{cases}
\hat{x}(t|t-1) = Q^m\hat{x}(t-1|t-1) \\
\hat{x}(t|t) = (1-\ell)\hat{x}(t|t-1)+\ell y(t)
\end{cases} \tag{8}$$

where $\hat{x} = [\hat{x}_1(t)\ \hat{x}_2(t)\ \cdots \hat{x}_N(t)]^T \in \mathbb{R}^N$ is the vector of the local estimators of the true state ξ at each node and $\ell \in (0,1)$ is the Kalman gain.

4 Control-based performance metrics for consensus algorithms

The performance analysis of consensus algorithms presented in Sect. 2, which exploits results from Markov chains literature, is focused on predicting the speed of convergence to the average. This is very useful, but however it is not the whole story. In fact, when convergence to the average is not an objective per se, but is used to solve an estimation or control problem, it is important to consider different performance measures, more tightly related to the actual objective pursued. Also, the introduction of other performance indices allows a better understanding of large-scale networks, because for some very relevant families of communication graphs, e.g., for grids (lattices), the essential spectral radius goes to one when the number of agents N grows, so that it is not clear whether $\mathtt{esr}(Q)^t$ will go to zero or not,

if both N and t tend to infinity. In this section, we will present examples of some alternative performance indices, and references to the relevant literature; however, this research topic is very recent and presently active, so that very likely new papers will appear in the next years.

For the sake of simplicity, we restrict our attention to constant Q, instead of studying all the (randomly)-time-varying schemes introduced in the previous sections. Moreover, we will always assume that $\mathscr{G}_Q$ is rooted and has all self-loops, so that Thm. 1 holds true. Additional assumptions that we will often use are that Q is doubly-stochastic, so that $\eta = \frac{1}{N}\mathbf{1}$, and that Q is normal, i.e., $Q^TQ = QQ^T$; under these assumptions, all the costs we consider can be re-written as simple functions of the eigenvalues of Q.

4.1 Performance indices

In this sections we give some examples of performance metrics arising in the use of consensus algorithm for estimation or control tasks. This is not a comprehensive list of all indices presented in the recent literature on distributed estimation and networked control; for example, we do not present here the interesting results related to estimation from relative measurements [5], to the costs arising from vehicle formation control [4], and clock synchronization [15].

4.1.1 LQ cost

As discussed in Sect. 3.4, an interesting performance metric is the LQ cost $J_{\mathrm{LQ}} = J_x + \varepsilon J_u$, where $J_x = \frac{1}{N}\sum_{t\geq 0}\mathbb{E}\left(\|x(t)-x(\infty)\|^2\right)$ is related to the speed of convergence, while a second term $J_u = \frac{1}{N}\sum_{t\geq 0}\mathbb{E}\left(\|x(t+1)-x(t)\|^2\right)$ takes into account the energy of the input sequence.

Let us see how to obtain easier expressions for J_x and J_u [23, 20]. Let us focus on the case when Q is doubly-stochastic, so that $x(\infty) = \frac{1}{N}\mathbf{1}^Tx(0)$. Under this assumption, the following equalities hold true[3]:

$$J_x = \frac{1}{N}\sum_{t\geq 0}\left\|Q^t - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T\right\|_{\mathrm{F}}^2 \quad \text{and} \quad J_u = \frac{1}{N}\sum_{t\geq 0}\left\|Q^{t+1} - Q^t\right\|_{\mathrm{F}}^2, \tag{9}$$

[3] J_x and J_u might be infinite for some choices of Q. A sufficient condition for convergence of both costs is that Q is doubly-stochastic, $\mathscr{G}_Q$ is rooted and $Q \in \mathbb{G}_{\mathrm{sl}}$. This is easily proved from Eqn. (9) using the following property of Frobenius norm: $\|AB\|_{\mathrm{F}} \leq \|A\|_{\mathrm{F}}\|B\|_{\mathrm{F}}$. Thus, $J_x \leq \frac{1}{N}\sum_{t=0}^{\infty}\left\|\left(Q-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\right\|_{\mathrm{F}}^{2t} = \frac{1}{N}\operatorname{tr}\sum_{t=0}^{\infty}\left(Q^TQ-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right)^t$, where the convergence of the last series is ensured by the fact that Q^TQ is stochastic (Q being doubly-stochastic) and $\mathscr{G}_{Q^TQ}$ is a subgraph of $\mathscr{G}_Q$ (thanks to the self-loops in $\mathscr{G}_Q$) and thus inherits its properties. A similar proof can be given also for J_u, after noting that $J_u = \sum_{t\geq 0}\|(Q-\frac{1}{N}\mathbf{1}\mathbf{1}^T)^t(Q-I)\|_{\mathrm{F}}^2$.

where $\|\cdot\|_{\mathrm{F}}$ the Frobenius norm of a square matrix, i.e., $\|A\|_{\mathrm{F}} = \sqrt{\mathrm{tr} A^T A}$.

If in addition Q is normal, then the expression furtherly simplifies to:

$$J_x = \tfrac{1}{N} \sum_{\substack{\lambda \in \Lambda(Q) \\ \lambda \neq 1}} \frac{1}{1-|\lambda|^2} \quad \text{and} \quad J_u = \tfrac{1}{N} \sum_{\substack{\lambda \in \Lambda(Q) \\ \lambda \neq 1}} \frac{|1-\lambda|^2}{1-|\lambda|^2} \tag{10}$$

where $\Lambda(Q)$ denotes the set of all eigenvalues of Q (with their multiplicity).

The proof —as all proofs in this section— repeatedly uses linearity of expectation and of trace, plus the observation that for any scalar $a \in \mathbb{R}$ we have $a = \mathrm{tr}\, a$, and the property $\mathrm{tr}(ABC) = \mathrm{tr}(CAB)$ where A, B, C are matrices of suitable size.

The first expression in Eqn. (9) is obtained as follows:

$$\begin{aligned} J_x &= \frac{1}{N} \sum_{t \geq 0} \mathbb{E}\left[x(0)^T (Q^t - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T)^T (Q^t - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T) x(0))\right] \\ &= \frac{1}{N} \sum_{t \geq 0} \mathbb{E}\left[\mathrm{tr}\left(x(0)^T (Q^t - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T)^T (Q^t - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T) x(0)\right)\right] \\ &= \frac{1}{N} \sum_{t \geq 0} \mathrm{tr}\left((Q^t - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T)^T (Q^t - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T) \mathbb{E}\left[x(0)x(0)^T\right]\right) . \end{aligned}$$

where we assume uniform distribution of initial conditions, i.e. $\mathbb{E}[x(0)x(0)^T] = I$. The second expression is easily obtained by the same techniques.

In order to prove Eqn. (10), we recall that normality of Q implies that all powers of Q, as well as Q^T and $Q^T Q$ are diagonalized with the same change of basis. Moreover, by stochasticity and primitivity of Q, also $Q - \frac{1}{N}\mathbf{1}\mathbf{1}^T$ (and all its powers, and its transpose) are diagonalized in that same basis and, denoting the eigenvalues of Q by $\lambda_1 = 1, \lambda_2, \ldots, \lambda_N$, we have that the eigenvalues of $Q^t - \frac{1}{N}\mathbf{1}\mathbf{1}^t$ are $\lambda_1 - 1 = 0, \lambda_2 - 0 = \lambda_2, \ldots, \lambda_N - 0 = \lambda_N$, so that $\|Q^t - \frac{1}{N}\mathbf{1}\mathbf{1}^t\|_{\mathrm{F}}^2 = \sum_{h=2}^N \|\lambda_h\|^{2t}$, and finally $J_x = \frac{1}{N}\sum_{h=2}^N \sum_{t \geq 0} (\|\lambda_h\|^2)^t = \frac{1}{N}\sum_{h=2}^N \frac{1}{1-\|\lambda_h\|^2}$.

For the second part of Eqn. (10), note $Q^{t+1} - Q^t$ is normal and has eigenvalues $\lambda_h^t(\lambda_h - 1)$ for $h = 1, \ldots, N$, and then conclude with the same technique as above.

4.1.2 Steady-state performance for noisy or quantized consensus

For the consensus algorithm of Eqn. (1), Thm. 1 tells everything about steady-state performance: when $t \to \infty$, $x(t) \to x(\infty) := \eta^T x(0)\mathbf{1}$, and if Q is doubly-stochastic, then $\eta = \frac{1}{N}\mathbf{1}$. However this is no longer true if there is noise in the consensus process, or quantization in the exchanged messages.

In the presence of noise within the successive iterations of the consensus algorithm, the steady state can be different from the average of the initial values, despite Q being doubly-stochastic. Here we present a case analyzed in [64], where the noise is additive.

Consider the following consensus algorithm affected by noise:

$$x(t+1) = Qx(t) + v(t)\,,$$

where $\{v_i(t)\}$ are noises uncorrelated w.r.t. both i and t, with zero mean and unit variance. Consider the case when Q is doubly-stochastic, so that, for any initial condition $x(0)$, $\mathbf{1}^T\mathbb{E}[x(t)] = \mathbf{1}^T x(0)$ for all t, and $\mathbb{E}[x(t)] \to \frac{1}{N}\mathbf{1}^T x(0)$. However, it is clear that the average-preserving property, and the convergence to $\frac{1}{N}\mathbf{1}^T x(0)$ are true only in expectation, and not for all realizations of the noise process. Thus, it is more reasonable to define the error as the distance from current average $\delta(t) = x(t) - \frac{1}{N}\mathbf{1}\mathbf{1}^T x(t)$) rather than distance from average consensus, which might not even exist. Hence, the relevant average quadratic cost is here defined as

$$J_{\text{noisy}} := \lim_{t\to\infty} \frac{1}{N}\mathbb{E}\left[\|x(t) - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T x(t)\|^2\right]$$

Notice that J_{noisy} turns out to be the same as the cost J_x introduced when studying the LQ-cost. In fact, note that

$$x(t) = Q^t x(0) + \sum_{s=0}^{t-1} Q^s v(t-1-s)\,,$$

so that $\delta(t) = (Q^t - \frac{1}{N}\mathbf{1}\mathbf{1}^T)x(0) + \sum_{s=0}^{t-1}(Q^s - \frac{1}{N}\mathbf{1}\mathbf{1}^T)v(t-1-s)$. Thus, by the statistical assumptions on the noises (zero-mean, uncorrelated, unit variance):

$$\begin{aligned}
\mathbb{E}\left[\|\delta(t)\|^2\right] &= \|(Q^t - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T)x(0)\|^2 \\
&\quad + 2\sum_{s=0}^{t-1} x(0)^T (Q^t - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T)^T (Q^s - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T)\mathbb{E}[v(t-1-s)] \\
&\quad + \sum_{r,s=0}^{t-1} \operatorname{tr}\left\{(Q^r - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T)^T (Q^s - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T)\mathbb{E}[v(t-1-r)v(t-1-s)]\right\} \\
&= \|(Q^t - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T)x(0)\|^2 + \sum_{s=0}^{t-1} \operatorname{tr}(Q^s - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T)^T (Q^s - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T)
\end{aligned}$$

When $t \to \infty$, the first term goes to zero, while the sum becomes an infinite sum, thus ending the proof.

A similar cost has been considered in [29], where however the noise was used as a model for **quantization** error, and thus noise appears in the equation in a different way, as follows:

$$x(t+1) = Q[(x(t) + v(t)] - v(t)$$

The fact that noise is multiplied by Q takes into account that the quantization error is within all messages passed to neighbors, while the substraction $-v(t)$ is possible,

as every agent knows its own quantization error, and is useful for avoiding accumulation of errors over time: in this way, the average $\frac{1}{N}\mathbf{1}^T x(t)$ is kept constant.

As in the previous case, the assumption is that $v_i(t)$'s are uncorrelated with respect to both i and t, and have zero-mean and unit variance, and Q is doubly-stochastic, so that $\mathbb{E}x(t) \to \frac{1}{N}\mathbf{1}^T x(0)$. Again the relevant cost is the variance of the distance from consensus $\delta(t) = x(t) - \frac{1}{N}\mathbf{1}\mathbf{1}^T x(t)$, in the limit of infinite number of iterations:

$$J_{\text{quantiz}} := \lim_{t\to\infty} \frac{1}{N}\mathbb{E}\left(\|x(t) - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T x(t)\|^2\right)$$

Clearly, due to the different update equation for $x(t)$, this will result in an expression for J_{quantiz} different from the one for J_{noisy}; it turns out that J_{quantiz} is equal to the cost J_u defined when dealing with the LQ-cost.

To prove this, notice that

$$x(t+1) = Q^t x(0) + \sum_{s=0}^{t-1} Q^s (Q-I) v(t-1-s)$$

so that $\delta(t+1) = (Q^t - \frac{1}{N}\mathbf{1}\mathbf{1}^T)x(0) + \sum_{s=0}^{t-1}(Q^{s+1} - Q^s)v(t-1-s)$.

By exploiting linearity of expectation and of trace, and the fact that arguments of the trace can be cyclically permuted, together with the assumptions on the noise, we get

$$\mathbb{E}\left(\|\delta(t)\|^2\right) = \|Q^t - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T)x(0)\|^2 + \sum_{s=0}^{t-1} \operatorname{tr}\left\{(Q^{s+1} - Q^s)^T (Q^{s+1} - Q^s)\right\}$$

By taking the limit for $t \to \infty$, the first term goes to zero, while the summation becomes an infinite sum, giving $J_{\text{quantiz}} = \frac{1}{N}\sum_{t\geq 0} \|Q^{t+1} - Q^t\|_{\text{F}}$ and thus ending the proof.

4.1.3 Performance of static estimation algorithm

Consider the static estimation problem described in Sect. 3.1, but in the simplest case, when all sensors have the same variance $\sigma^2 = 1$. In this case, the best estimate is the average, $\hat{\theta}_{\text{MV}} = \frac{1}{N}\mathbf{1}^T x(0)$ and the sensors can compute it in a distributed way by simply using a consensus algorithm $x(t+1) = Qx(t)$, for some stochastic matrix Q. What is peculiar to this setting, is that the focus is not on how precisely the average is computed, but on how good the estimate of θ is. In fact, knowing that $x(t)$ converges to $x(\infty) = \eta^T x(0)\mathbf{1}$ does not answer the questions on how well is θ estimated by $x(\infty)$ if the matrix Q is not doubly-stochastic and on how well is θ estimated after t iterations of the algorithm.

To address these questions, we consider the estimation error $e(t) := x(t) - \theta\mathbf{1}$. To answer the first question, let us first notice that, if Q is doubly-stochastic, then $x(\infty)$

is the average of the measurements, i.e., $x(\infty) = \hat{\theta}_{\mathrm{MV}}\mathbf{1}$, and $\hat{\theta}_{\mathrm{MV}}$ has zero-mean and variance $\frac{1}{N}$. If Q is not doubly-stochastic, then it is interesting to study the error; it is easy to see that $e(\infty) = \mathbf{1}\eta^T v$, and so $\mathbb{E}[e(\infty)] = 0$, while its covariance matrix is

$$\mathbb{E}\left[e(\infty)e(\infty)^T\right] = \mathbf{1}\eta^T\mathbb{E}\left[vv^T\right]\eta\mathbf{1}^T = \mathbf{1}\|\eta\|^2\mathbf{1}^T = \|\eta\|^2\mathbf{1}\mathbf{1}^T,$$

i.e., each sensor's final estimate has variance $\|\eta\|^2$. Notice that $1/N \leq \|\eta\|^2 \leq 1$, since $\|\eta\|_1 = 1$.

Now let us turn our attention to the more interesting problem of understanding how well θ is estimated after a finite number of iterations, t, studying $e(t)$. More precisely, the relevant performance measure is the average quadratic error, defined as

$$J_{\mathrm{estim}}(t) := \tfrac{1}{N}\mathbb{E}\left[\|x(t) - \theta\mathbf{1}\|^2\right]$$

This cost can be re-written as:

$$J_{\mathrm{estim}}(t) = \tfrac{1}{N}\,\mathrm{tr}[(Q^T)^t Q^t]$$

and, if Q is normal, the expression simplifies as follows:

$$J_{\mathrm{estim}}(t) = \tfrac{1}{N}\sum_{\lambda\in\Lambda(Q)}|\lambda|^{2t}. \tag{11}$$

To prove the first claim, note that

$$J_{\mathrm{estim}}(t) = \tfrac{1}{N}\mathbb{E}\left[\|Q^t x(0) - \theta\mathbf{1}\|^2\right] = \tfrac{1}{N}\mathbb{E}\left[\|(Q^t - I)\theta\mathbf{1} + Q^t v\|^2\right] = \tfrac{1}{N}\mathbb{E}\left[v^T(Q^t)^T Q^t v\right]$$

from which the claim follows by taking the trace and cyclically permuting its arguments, recalling that $\mathbb{E}[vv^T] = I$. Then the simplified expression for normal Q is immediate.

4.1.4 Distributed Kalman filter

Consider the distributed Kalman filter presented in Sect. 3.7, and in particular its scalar version described in Eqn. (8). There are different ways of analyzing performance of such algorithm. One interesting performance index is the asymptotic quadratic estimation error, defined as:

$$J_{\mathrm{K,est}} := \tfrac{1}{N}\lim_{t\to\infty}\mathbb{E}\left[\|\hat{x}(t|t) - x(t)\mathbf{1}\|^2\right]$$

This cost can be re-formulated as follows:

$$J_{\mathrm{K,est}} = \frac{q(1-\ell)^2}{1-(1-\ell)^2} + \tfrac{1}{N}\sum_{s=0}^{\infty}\|(1-\ell)^s Q^{sm}\|_{\mathrm{F}}^2$$

and, in the case when Q is normal, the following easier characterization holds true:

$$J_{\mathrm{K,est}} = \frac{q(1-\ell)^2}{1-(1-\ell)^2} + \frac{r\ell^2}{N} \sum_{\lambda \in \Lambda(Q)} \frac{1}{1-(1-\ell)^2|\lambda|^{2m}}\,.$$

Another relevant performance metric is the asymptotic quadratic prediction error

$$J_{\mathrm{K,pred}} := \tfrac{1}{N} \lim_{t\to\infty} \mathbb{E}\left[\|\hat{x}(t|t-1) - x(t)\mathbf{1}\|^2\right],$$

which can be re-written as

$$J_{\mathrm{K,pred}} = \frac{q}{1-(1-\ell)^2} + \frac{r\ell^2}{N} \sum_{s=0}^{\infty} \left\| (1-\ell)^s Q^{(s+1)m} \right\|_{\mathrm{F}}^2$$

and, for normal Q, is also equal to

$$J_{\mathrm{K,pred}} = \frac{q}{1-(1-\ell)^2} + \frac{r\ell^2}{N} \sum_{\lambda \in \Lambda(Q)} \frac{|\lambda|^{2m}}{1-(1-\ell)^2|\lambda|^{2m}}\,.$$

The techniques used for obtaining the simplified expressions are similar to those shown for the costs previously presented and details can be found in [14].

4.2 Evaluation and optimization of performance indices

Clearly any performance index can be numerically computed for a given matrix Q, and gives a way of comparing the quality of different choices for Q. However, there are two research lines which lead to interesting results using some performance index. A first line concerns optimization of a chosen cost among all matrices Q consistent with a given communication graph. A second interesting direction is the study of the different costs for some relevant families of graphs and matrices, in particular for large-scale graphs. The more classical results in this two directions when the performance index is the essential spectral radius are discussed in Section 2.2.

Providing a comprehensive summary of the results is beyond the scope of this chapter: we give here some examples, so as to illustrate some curious or unexpected results and motivate the need for different performance metrics, and then we give pointers to some relevant literature, with the disclaimer that —this being a very recent and still active research area— our reference list will surely turn out to be incomplete.

An interesting work on design of the entries of Q for a given graph by optimization of a cost different from $\mathtt{esr}(Q)$ is Xiao et al. [64]. Here noisy consensus is analyzed, so that the relevant metric is $J_{\mathrm{noisy}} = J_x$. The authors show that the problem of finding, for a given graph and among all symmetric choices of weights, the weights minimizing J_{noisy}, is a convex optimization problem, and they provide efficient (although centralized) algorithms for its solution. They also compare numerically, for various graphs topologies, the three costs J_{noisy} obtained with the optimal Q, with

the Q minimizing $\mathtt{esr}Q$ and the simple Metropolis rule; for some topologies the difference is significant, while for other graphs the three results are very similar.

Another example is Carli et al. [14], where the problem of optimizing $J_{\mathrm{K,pred}}$ for a given graph among normal matrices is examined. The first interesting result is that symmetric matrices are indeed optimal, and then, the authors prove that, for fixed ℓ, the optimization problem among symmetric matrices is convex in Q; however, despite the problem being also convex in ℓ, it is not jointly convex in Q and ℓ. Then simplified problems (under the limit for infinite communication or for small measuerement noise) are studied more in detail.

The optimality of de Bruijn graphs with respect to convergence speed, among all graphs with bounded in-degree, is confirmed, at least asymptotically in N and for small ε, also when the LQ cost is considered [23].

Another approach which is receiving much attention is the study of asymptotic performance in large-scale graphs. The idea is to consider families of graphs of increasing size, sharing the same common properties (in some sense that will be specified in the examples, having the same shape), and to analyze how the cost scales with the number of nodes. This is more an analysis than a design problem, but it gives useful hints on the number of nodes. Here we present a simple example.

Example 1 (Circle). Consider a graph $\mathscr{G}_N$ consisting of a circle o N nodes, where each node has a self-loop and an outgoing edge towards its neighbor on the right. Consider a coefficient $1/2$ on each edge, so that $Q_N = \mathrm{circ}(1/2, 1/2, 0, \dots, 0)$ is a circulant matrix. Because $\mathscr{G}_N$ is circulant, we know that it is normal, and we can easily compute its eigenvalues: $\Lambda(Q_N) = \{\frac{1}{2} + \frac{1}{2}e^{i\frac{2\pi}{N}h}, h = 0, \dots, N-1\}$ [17]. Thus, the essential spectral radius is

$$\mathtt{esr}(Q_N) = |\lambda_1^2| = \sqrt{\tfrac{1}{2}(1+\cos(\tfrac{2\pi}{N}))} = 1 - \tfrac{\pi^2}{N^2} + O(\tfrac{1}{N^4}) \quad \text{for } N \to \infty.$$

Now we can plug the expression for the eigenvalues in Equations (10) and (11). Then, an explicit computation (see e.g. [19]) gives that

$$J_u = 1 - \frac{1}{N}$$

while some careful upper and lower bounds (see e.g. [20]) show that

$$c_1 N \le J_x(Q_N) \le c_2 N \quad \text{and} \quad c_3 \max\left\{\tfrac{1}{N}, \tfrac{1}{\sqrt{t}}\right\} \le J_{\mathrm{estim}}(Q_N, t) \le c_4 \max\left\{\tfrac{1}{N}, \tfrac{1}{\sqrt{t}}\right\},$$

where c_1, c_2, c_3, c_4 are positive numbers independent of N.

It is interesting to compare the performance of the circle with that of a complete graph, i.e., with the case $Q'_N = \frac{1}{N}\mathbf{1}\mathbf{1}^T$, where in one step the exact average is computed. It is easy to see that the eigenvalues of Q' are 1 with multiplicity 1 and 0 with multiplicity $N-1$, so that $\mathtt{esr}(Q'_N) = 0$, $J_x(Q'_N) = 1 - \frac{1}{N}$, $J_{\mathrm{estim}}(Q'_N, t) = \frac{1}{N}$ for all $t \ge 1$). Intuitively, performance of the circle is much worse, because of the slow flow of information, as opposed to the complete exchange of messages in one single iteration for the complete graph. This intuition is confirmed for most performance

indices; however, it is interesting to note that $J_x(Q_N) = J_x(Q'_N)$ is actually the same as for the circle and for the complete graph, thus showing that a different choice of performance metric can lead to significantly different results.

The key point that allows to study the example of the circle is the fact that an expression for the eigenvalues is easily found, thanks to the algebraic structure of Q, which is circulant. The same can be done more in general, for the case of circles with more edges towards neighbors (giving rise to different circulant matrices) and for higher dimension, where the underlying algebraic structure is that of Cayley graphs, Cayley matrices and discrete Fourier transform over Abelian groups (see e.g. [17]). The result presented in [20] concerns grids on d-dimensional torus, or grids on d-dimensional cubes with some assumptions of symmetry of the coefficients and suitable border conditions, and in both cases with local neighborhoods (bounded difference among labels of nodes connected by an edge). It states that

$$c_1 f_d(N) \leq J_x \leq c_2 f_d(N) \quad \text{and} \quad c_3 \max\left\{\tfrac{1}{N}, \tfrac{1}{(\sqrt{t})^d}\right\} \leq J_{\text{estim}}(t) \leq c_4 \max\left\{\tfrac{1}{N}, \tfrac{1}{(\sqrt{t})^d}\right\},$$

where $f_1(N) = N$, $f_2(N) = \log N$ and $f_d(N) = 1$ for all $d \geq 3$, and where c_1, c_2, c_3, c_4 are positive numbers independent of N.

The study of Cayley graphs, although motivated by the algebraic structure that allows to tackle the analysis, is interesting, because they are a simplified and idealized version of communication scenarios of practical interest. In particular, they capture the effects on performance of the strong constraint that communication is local, not only in the sense of a little number of neighbors, but also with a bound on the distance among connected agents. The study of more irregular and realistic scenarios of communication with geometric constraints is the subject of on-going research, where two main directions are being explored. On the one side, there is an interest in the random geometric graph model (points thrown uniformly at random in a portion of space and edges among all pairs of vertices within a given distance r), for which simulations show a behavior very similar to that of a grid (see e.g. [20]), but a rigorous theory is still missing: most of known results concern only the essential spectral radius and not all the spectrum. On the other side, there is the idea to study perturbations of known graphs; this is completely different from traditional theory of perturbation of matrices, because here perturbations are not continuous, and are little in the sense that only few edges (with respect to the graph size) are removed or added or receive different weight. In this direction, a useful tool (because of its monotonicity properties with respect to edge insertion) is the analogy between reversible Markov chains and resistive electrical networks, exploited e.g. in [5].

We conclude this section by presenting in detail an example that clarifies how comparing two families of graphs by two different performance measures can indeed significantly change the result, leading to a different definition of the 'best' graph. This is a toy example, not very sensible in practice, but easily highlighting which issues can arise.

Example 2. Let N be an even number, and consider $\mathscr{G}_N$ a graph consisting of two disconnected complete graphs, each on $N/2$ nodes; Fig. 1 depicts $\mathscr{G}_{10}$ as an example.

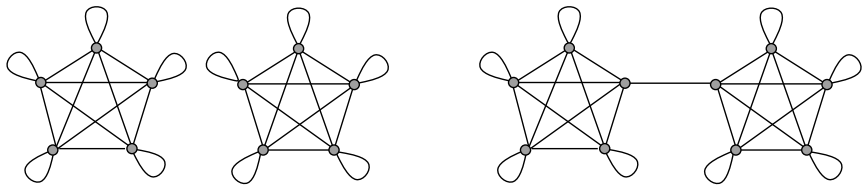
Associate to each edge a coefficient $2/N$, so that Q_N has the following form:

$$Q_N = \left[\begin{array}{c|c} \frac{2}{N}\mathbf{1}\mathbf{1}^T & 0 \\ \hline 0 & \frac{2}{N}\mathbf{1}\mathbf{1}^T \end{array}\right].$$

We would like to compare performance of this Q_N with the circle presented as Example 1, by looking at the essential spectral radius, and then by looking at the estimation error J_{estim}. The eigenvalues of Q_N are simply 1 with multiplicity 2 and the eigenvalue 0 with multiplicity $N-2$, so that $\texttt{esr}(Q_N) = 1$, which is worse than the circle. However, for all $t \geq 1$, $J_{\text{estim}}(Q_N) = \frac{2}{N}$, which is almost as good as the best possible error (the error variance in the case of centralized estimation, $\frac{1}{N}$), as opposed to the circle which, for large N, has a very slow convergence.

Behind computation of the eigenvalues, there is an intuitive explanation of what happens. In the graphs $\mathcal{G}_N$, the essential spectral radius 1 describes the fact that the graph is disconnected, and thus no convergence is possible to the average of all initial values: simply no information can transit from one group to another; nevertheless, the estimation error is very good for large N, because it is the average of $N/2$ measurements, and it is computed very fast, in one iteration, thanks to the complete graph which gives centralized computation within the group of $N/2$ agents. Conversely, in the circle average consensus can be reached asymptotically, as described by the essential spectral radius smaller than one, but convergence is very slow for large N ($\texttt{esr} = 1 - \frac{\pi^2}{N^2} + O(\frac{1}{N^2})$), and a reasonably good estimation error is achieved only after a long time.

The readers concerned with the fact that $\mathcal{G}_N$ is disconnected (and thus violates the assumptions made throughout this chapter) may consider a slightly modified graph $\tilde{\mathcal{G}}_N$, as shown in Figure 1(b), still associating a coefficient $2/N$ with each edge; Let us denote by $\tilde{Q}_N$ the matrix so modified. This graph is studied in [10] under the name $K_{N/2} - K_{N/2}$ and [10, Prop. 5.1] gives the exact computation of all eigenvalues of $\tilde{Q}_N$: $\Lambda(\tilde{Q}_N)$ has 1 with multiplicity 1, 0 with multiplicity $N-3$ and then $\frac{1}{2} - \frac{2}{N} \pm \frac{1}{2}\sqrt{1 + \frac{8}{N} - \frac{16}{N^2}}$ with multiplicity 1 each. Here the single edge connecting the two subgroups of agents allows only a quite slow convergence ($\texttt{esr}(\tilde{Q}_N) = 1 - \frac{8}{N^2} + O(\frac{1}{N^2})$, very similar to that of the circle), while the estimation error becomes very good after few iterations ($J_{\text{estim}}(\tilde{Q}_N) \leq \frac{3}{N}$ for all $t \geq 1$).



(a) Graph $\mathcal{G}_{10}$ in Example 2 (b) Graph $\tilde{\mathcal{G}}_{10}$ in Example 2

Fig. 1 Communication graphs considered in Example 2.

5 Conclusion

In this chapter we have tried to present a comprehensive view of the linear consensus algorithms from a control and estimation perspective, by reviewing the most important results available in the literature, by showing some of the possible applications in control and estimation, and by presenting which are suitable control-based indices of performance for the consensus algorithm design.

We believe that much has still to be done in this area, in particular in two directions. The first direction points to finding which traditional control and estimation problems can be cast as consensus problems. In fact, although not all problems can be cast as averages of local quantities, if they can be approximated as so, we could exploit the effectiveness and strong robustness of consensus algorithms. The second direction addresses the implications of the new control-based performance metrics for the design of the consensus algorithms. In fact, as we illustrated with few toy examples, they give rise to design criteria that can be quite different from the traditional ones.

Acknowledgements We would like to thank Sandro Zampieri for many useful discussions on the topics of this chapter and for reviewing the original manuscript, and Paolo Frasca for his suggestions. We acknowledge funding from the European Community's Seventh Framework Programme under agreement n. FP7-ICT-223866-FeedNetBack and from the Italian CaRiPaRo Foundation under project WISE-WAI.

References

1. M. Alighanbari and J.P. How. An unbiased Kalman consensus algorithm. In *Proceedings of IEEE American Control Conference (ACC'06)*, 2006.
2. D. Angeli and P.-A. Bliman. Tight estimates for non-stationary consensus with fixed underlying spanning tree. In *Proceedings of 17th IFAC World Congress (IFAC'08)*, 2008.
3. T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione. Broadcast gossip algorithms for consensus. *IEEE Transactions on Signal Processing*, 57(7):2748–2761, 2009.
4. B. Bamieh, M. Jovanovic, P. Mitra, and S. Patterson. Coherence in large-scale networks: Dimension dependent limitations of local feedback. *Submitted to IEEE Trans. Aut. Cont.*, 2009.
5. Prabir Barooah. *Estimation and Control with Relative Measurements: Algorithms and Scaling Laws*. PhD thesis, University of California, Santa Barbara, 2007.
6. D. Bauso, L. Giarré, and R. Pesenti. Nonlinear protocols for optimal distributed consensus in networks of dynamic agents. *Systems and Control Letters*, 55:918–928, 2006.
7. P.A. Bliman and G. Ferrari-Trecate. Average consensus problems in networks of agents with delayed communications. *Proceedings of 44th IEEE Conference on Decision and Control (CDC'05)*, 44(8):1985–1995, 2008.
8. V.D. Blondel, J.M. Hendrickx, A. Olshevsky, and J.N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Proceedings of 44th IEEE Conference on Decision and Control (CDC'05) and European Control Conference (ECC'05)*, pages 2996– 3000, Dec. 2005.
9. S. Bolognani, S. Del Favero, L. Schenato, and D. Varagnolo. Consensus-based distributed sensor calibration and least-square parameter identification in wsns. *International Journal of Robust and Nonlinear Control*, 20(2):176–193, 2010.

10. S. Boyd, P. Diaconis, P. Parrilo, and L. Xiao. Symmetry analysis of reversible Markov chains. *Internet Mathematics*, 2:31–71, 2005.
11. S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory/ACM Transactions on Networking*, 52(6):2508–2530, June 2006.
12. F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks*. Applied Mathematics Series. Princeton University Press, 2009.
13. M. Cao, A.S. Morse, and B.D.O. Anderson. Reaching a consensus in a dynamically changing environment a graphical approach. *SIAM Journal on Control and Optimization*, 47(2):575–600, 2008.
14. R. Carli, A. Chiuso, L. Schenato, and S. Zampieri. Distributed Kalman filtering based on consensus strategies. *IEEE Journal on Selected Areas in Communications*, 26(4):622–633, 2008.
15. R. Carli, A. Chiuso, S. Zampieri, and L. Schenato. A PI consensus controller for networked clocks synchronization. In *IFAC World Congress on Automatic Control (IFAC 08)*, 2008.
16. R. Carli, G. Como, P. Frasca, and F. Garin. Average consensus on digital noisy networks. In *Proc. of 1st IFAC Workshop on Estimation and Control of Networked Systems (NecSys'09)*, pages 36–41, 24-26 Sept. 2009.
17. R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri. Communication constraints in the average consensus problem. *Automatica*, 44(3):671–684, 2008.
18. R. Carli, P. Frasca, F. Fagnani, and S. Zampieri. Gossip consensus algorithms via quantized communication. *Automatica*, 46:70–80, 2010.
19. Ruggero Carli. *Some issues on the Average Consensus Problem*. PhD thesis, Università di Padova, Italy, 2008.
20. Ruggero Carli, Federica Garin, and Sandro Zampieri. Quadratic indices for the analysis of consensus algorithms. In *Proc. of the 4th Information Theory and Applications Workshop*, pages 96–104, La Jolla, CA, USA, Feb. 2009.
21. J. Cortés. Distributed algorithms for reaching consensus on general functions. *Automatica*, 44:726–737, 2008.
22. J. Cortes, S. Martinez, and F. Bullo. Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control*, 51(8):1289– 1298, Aug. 2006.
23. Jean-Charles Delvenne, Ruggero Carli, and Sandro Zampieri. Optimal strategies in the average consensus problem. In *Proc. of the IEEE Conference on Decision and Control 2007*, pages 2498–2503, 2007.
24. Jean-Charles Delvenne, Ruggero Carli, and Sandro Zampieri. Optimal strategies in the average consensus problem. *Systems and Control Letters*, 58:759–765, 2009.
25. F. Fagnani and S. Zampieri. Asymmetric randomized gossip algorithms for consensus. In *Proceedings of 17th IFAC World Congress (IFAC'08)*, 2008.
26. F. Fagnani and S. Zampieri. Randomized consensus algorithms over large scale networks. *IEEE Journal on Selected Areas in Communications*, 26(4):634–649, May 2008.
27. F. Fagnani and S. Zampieri. Average consensus with packet drop communication. *SIAM Journal on Control and Optimization*, 48:102–133, 2009.
28. J.F. Fax and R.M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, September 2004.
29. Paolo Frasca, Ruggero Carli, Fabio Fagnani, and Sandro Zampieri. Average consensus on networks with quantized communication. *International Journal of Robust and Non-Linear Control*, 19(16):1787–1816, 2009.
30. Andreas Frommer and Daniel B. Szyld. On asynchronous iterations. *Journal of Computation and Applied Mathematics*, 123:201–216, 2000.
31. Y. Hatano and M. Mesbahi. Agreement over random networks. *IEEE Transactions on Automatic Control*, 50(11):1867– 1872, Nov. 2005.
32. M. Huang and J. H. Manton. Coordination and consensus of networked agents with noisy measurements: Stochastic algorithms and asymptotic behavior. *SIAM Journal on Control and Optimization*, 48(1):134–161, 2009.

33. A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, June 2003.
34. S. Kar, J. M. F. Moura, and K. Ramanan. Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication. *Submitted*, 2008.
35. S. Kar and J.M.F. Moura. Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise. *IEEE Transactions on Signal Processing*, 57(5):355–369, 2009.
36. S. Kar and J.M.F. Moura. Distributed consensus algorithms in sensor networks: Quantized data and random link failures. *IEEE Transactions on Signal Processing*, 58(3):1383–1400, 2010.
37. A. Kashyap, T. Basar, and R. Srikant. Quantized consensus. *Automatica*, 43(7):1192–1203, 2007.
38. Javad Lavaei and Richard M. Murray. On quantized consensus by means of gossip algorithm–part I: Convergence proof. In *Proceedings of the American Control Conference (ACC'09)*, 2009.
39. H. Minc. *Nonnegative matrices*. John Wiley & Sons, 1988.
40. L. Moreau. Stability of continuous-time distributed consensus algorithms. In *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC'04)*, volume 4, pages 3998– 4003, December 2004.
41. L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169– 182, Feb. 2005.
42. S. Muthukrihnan, B. Ghosh, and M. H. Schultz. First and second order diffusive methods for rapid, coarse, distributed load balancing. *Theory of Computing Systems*, 31:331–354, 1998.
43. A. Nedic, A. Olshevsky, A. Ozdaglar, and J.N. Tsitsiklis. On distributed averaging algorithms and quantization effects. *IEEE Transaction on Automatic Control*, 54(11):2506–2517, 2009.
44. R. Olfati-Saber. Ultrafast consensus in small-world networks. In *Proceedings of the 2005 American Control Conference (ACC'05)*, volume 4, pages 2371– 2378, 2005.
45. R. Olfati-Saber. Algebraic connectivity ratio of ramanujan graphs. In *Proceedings of the 2007 American Control Conference*, July 2007.
46. R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.
47. S. Patterson, B. Bamieh, and A. El-Abbadi. Convergence rates of distributed average consensus with stochastic link failures. *IEEE Transactions on Automatic Control*, 2009.
48. M. Penrose. *Random Geometric Graphs*. Oxford University Press, 2003.
49. R. Rajagopal and M. J. Wainwright. Network-based consensus averaging with general noisy channels. Technical Report 751, Dept. of Statistics, UC Berkeley, 2008.
50. W. Ren and R.W. Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50(5):655– 661, May. 2005.
51. W. Ren, R.W. Beard, and E.M. Atkins. Information consensus in multivehicle cooperative control. *IEEE Control Systems Magazine*, 27(2):71–82, 2007.
52. R. Olfati Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in multi-agent networked systems. *Proceedings of IEEE*, 95(1):215–233, January 2007.
53. E. Seneta. *Non-negative Matrices and Markov Chains*. John Wiley & Sons, Inc., Springer, 2006.
54. A. Seuret, D.V. Dimarogonas, and K.H. Johansson. Consensus under communication delays. In *Proceedings of the 47nd IEEE Conference on Decision and Control (CDC'08)*, pages 4922–4927, December 2008.
55. D. P. Spanos, R. Olfati-Saber, and R. M. Murray. Distributed Kalman filtering in sensor networks with quantifiable performance. In *Proccedings of the Information Processing for Sensor Networks (IPSN'05)*, pages 133– 139, 2005.
56. John C. Strikwerda. A probabilistic analysis of asynchronous iteration. *Journal of Linear Algebra and its Applications*, 349:125–154, 2002.

57. A. Tahbaz-Salehi and A. Jadbabaie. Small world phenomenon, rapidly mixing Markov chains, and average consensus algorithms. In *Proceedings of IEEE Conference on Decision and Control (CDC'07)*, pages 276–281, 2007.
58. Alireza Tahbaz-Salehi and Ali Jadbabaie. Consensus over ergodic stationary graph processes. *IEEE Transaction on Automatic Control*, 54(12), 2009.
59. H. Tanner, A. Jadbabaie, and G.J. Pappas. Flocking in fixed and switching networks. *IEEE Transaction on Automatic Control*, 52(5):863–868, 2007.
60. H.G. Tanner and D.K. Christodoulakis. The stability of synchronization in local-interaction networks is robust with respect to time delays. In *Proceedings of the 44nd IEEE Conference on Decision and Control (CDC'05)*, pages 4945–4950, December 2005.
61. J.N. Tsitsiklis and M. Athans. Convergence adn asymptotic agreement in distributed decision problems. *IEEE Transactions on Automatic Control*, 29(1):42–50, 1984.
62. Feng Xiao and Long Wang. Consensus protocols for discrete-time multi-agent systems with time-varying delays. *Automatica*, 44(10):2577–2582, 1998.
63. L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems and Control Letters*, 53(1):65–78, September 2004.
64. L. Xiao, S. Boyd, and S.-J. Kim. Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 67(1):33–46, September 2007.
65. L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proccedings of the Information Processing for Sensor Networks (IPSN'05)*, pages 63– 70, 2005.