# Local Average Consensus in Distributed Measurement of Spatial-Temporal Varying Parameters: 1D Case


Kai Cai [a], Brian D.O. Anderson [b], Changbin Yu [c], Guoqiang Mao [d]

[a] *Urban Research Plaza, Osaka City University, Osaka, Japan*

[b] *Australian National University and NICTA Ltd, Canberra, Australia*

[c] *Australian National University and NICTA Ltd, Canberra, Australia, and Shandong Computer Science Center, Jinan, China*

[d] *School of Computing and Communications, University of Technology Sydney and NICTA Ltd, Australia*



**Abstract**

We study a new variant of consensus problems, termed 'local average consensus', in networks of agents. We consider the task of using sensor networks to perform distributed measurement of a parameter which has both spatial (in this paper 1D) and temporal variations. Our idea is to maintain potentially useful local information regarding spatial variation, as contrasted with reaching a single, global consensus, as well as to mitigate the effect of measurement errors. We employ two schemes for computation of local average consensus: exponential weighting and uniform finite window. In both schemes, we design local average consensus algorithms to address first the case where the measured parameter has spatial variation but is constant in time, and then the case where the measured parameter has both spatial and temporal variations. Our designed algorithms are distributed, in that information is exchanged only among neighbors. Moreover, we analyze both spatial and temporal frequency responses and noise propagation associated with the algorithms. The tradeoffs of using local consensus, as compared to standard global consensus, include higher memory requirement and degraded noise performance. Arbitrary updating weights and random spacing between sensors are also analyzed in the proposed algorithms.



*Key words:* Local average consensus, sensor networks, spatial/temporal frequency response, noise propagation, bandwidth


## 1 Introduction

Consensus of multi-agent systems comes in many varieties (e.g. [12, 19, 21]), and in this paper, we focus on a particular variety, namely average consensus (e.g. [5, 22, 24]). This refers to an arrangement where each of a network of agents is associated with a value of a certain variable, and a process occurs which ends up with all agents learning the average value of the variable. Finding an average of a set of values is apparently conceptually trivial; what makes average consensus nontrivial is the fact that an imposed graphical structure limits the nature of the steps that can be part of the averaging algorithm, each agent only being allowed to exchange information with its neighbors, as defined by an overlaid graphical structure. Issues also arise of noise performance, effect of time delay, agent/link loss, etc ( [13, 15, 17]).

Finding an average also throws away much information. In many situations, one might well envisage that a *local average* might be useful, retaining the characteristics of local information meanwhile mitigating the effect of measurement error. For instance, one thousand weather stations across a city, instead of giving a single air pollution reading, might validly be used to identify hotspots of pollution, i.e. localities with high pollution; thus, instead of a global average, a form of local averaging, still mitigating the effects of some noise, might be useful. We term this variant 'local (average) consensus', and distinguish it from the normal sort of consensus, termed here by way of contrast 'global (average) consensus'.

In local average consensus, each agent $i$ aims to compute the linear combination $a_{i,i}x_i + (a_{i,i-1}x_{i-1} + a_{i,i-2}x_{i-2} + \cdots) + (a_{i,i+1}x_{i+1} + a_{i,i+2}x_{i+2} + \cdots)$, where $x_j$ $(j = i, i\pm1, i\pm2, ...)$ are measurements and $a_{i,j}$ are weights that reflect local information around $i$ in some reasonable sense. Accordingly, defining a local average amounts to choosing appropriate


⋆ K. Cai's work was supported by Program to Disseminate Tenure Tracking System and Grants-in-Aid for Scientific Research No. 26870169, MEXT, Japan. B.D.O. Anderson's work was supported by the Australian Research Council through DP-130103610 and DP-110100538 and by National ICT Australia. C. Yu's work was supported by the Australian Research Council through DP-130103610 and a Queen Elizabeth II Fellowship under DP-110100538, the National Natural Science Foundation of China (61375072), and the Overseas Expert Program of Shandong Province.



*Email addresses:* `kai.cai@info.eng.osaka-cu.ac.jp` (Kai Cai), `brian.anderson@anu.edu.au` (Brian D.O. Anderson), `brad.yu@anu.edu.au` (Changbin Yu), `g.mao@ieee.org` (Guoqiang Mao).

weights $a_{i,j}$ and is by no means unique. We consider two schemes for defining a local average, both assigning large weights to locally measured information (precise definitions are given in Section 2 below). One involves the use of exponential weights, to reflect 'closeness' of the variable measured in both topological and geographical distance (viz. the further a neighbor is, the lesser its value will affect the agent's computation of its 'local average'). The other scheme applies uniform weights to all measurements within a finite window centered at each agent, to reflect locality and to reduce computation burden. In addition, these two schemes have an appealing feature that the weights involved may be parameterized using a *single* parameter; this renders our algorithm amenable for technical analysis as well as convenient for practical use.

In both schemes, we design local consensus algorithms to first address the case where the measured variable has spatial variation but is constant in time, and then the case where the measured variable has both spatial and temporal variations. In this paper we consider spatial variation in 1D for simplicity. Our designed algorithms have the following properties:

• The algorithms are distributed: i.e. information exchange is allowed only among neighboring agents.

• The algorithms will respond to time-variation of measured variables without delay.

• The algorithms contain a single parameter (corresponding to the one for respective weights in each scheme) which controls the distance over which effective local averaging occurs.

• The algorithms are robust against errors in spacing, as well as against errors in measured variables.

As we will see, these algorithms have higher memory requirement than that of a global consensus algorithm (the latter can be made memoryless). Moreover, we study two generalizations of the local consensus algorithms, one with arbitrary weights and the other with random spacing between sensors.

We also analyze the spatial and temporal frequency responses of the designed local consensus algorithms, and noise propagation associated with these algorithms. To obtain a fully analytical result we limit our study to a 1D sensor network, which can find its application in power line monitoring, canal/river monitoring, detection of border intrusions, structural monitoring of railways/bridges/pipelines, and road traffic control [1, 7, 9, 11, 25]. One example of road traffic control is to monitor vehicular density along a long stretch of highway, a parameter naturally spatial- and temporal-varying. Suppose there are a number of sensors spread along the highway; the vehicular density at each measurement point is correlated and the correlation reduces with the distance from the measurement point. Local average consensus may help identify congested sections of the highway and their time shifts, whereas global consensus is much less useful in this context.

We note that [20] proposed a "consensus filter" which allows the nodes of sensor networks to track the average of their time-varying noisy measurements. This problem is called "dynamic average consensus", which is later further studied in e.g. [3, 8], and also in [2, 6, 10] under a different name "coordinated average tracking". These works, however, deal still with global average consensus, because all nodes are required to track *the same* time-varying average value. By contrast, our goal of local average consensus is to have each node track the time-varying average value only within its spatial neighborhood, thereby retaining characteristics of locally measured information.

Also related is the work on distributed estimation of (time-varying) multi-dimensional parameter; different approaches have been proposed, notably consensus plus Kalman filtering [18], consensus plus least-mean-square adaptation [16], and consensus plus innovation [14]. The first main difference between our work and the above is the approach to reducing noise effect: In [14,16,18] the filtering/adaptation/innovation part serves to reduce noise; by contrast, our approach uses (local) averaging itself to reduce noise. As a consequence, the algorithms designed in [14, 16, 18] require constantly making new measurements no matter the parameter is time-varying or not. By contrast, our algorithms needs only the initial measurements in the case of time-invariant parameter, and for time-varying parameter, new measurements are made solely for the purpose of tracking temporal variation of the parameter, not for reducing noise. In addition, our algorithms contain a single parameter which can be easily tuned for noise and tracking performance; this feature renders our algorithms more convenient to use as compard those in [14, 16, 18] having more parameters.

In the following, Section 2 presents local average consensus algorithms for the case where the measured variable has spatial variation but is constant in time. Section 3 and Section 4 investigate spatial frequency response and noise propagation of the designed algorithms. Section 5 studies arbitrary weights and random spacing in the proposed local averaging algorithms. Section 6 presents local consensus algorithms for the case where the measured variable has both spatial and temporal variations. This allows the treatment of Section 7 of the frequency response associated with time variations. Finally, Section 8 states our conclusions. The conference predecessor of this paper is [4]. This paper differs from [4] through inclusion of proofs of results, development of material on the frequency response to time-variation in measured variables, and analysis of random spacing and arbitrary weights in the proposed algorithms.

## 2 Distributed Local Consensus Algorithms

Consider a variable whose values vary in 1D space, and/or in addition vary in time. Suppose we have a (possibly infinite) chain of sensors to be placed (uniformly) along the 1D space. Each sensor $i$ has two variables: a measurement variable $x_i$ and a consensus variable $y_i$. At each time $k = 0, 1, 2, ...$ each sensor $i$ takes a measurement $x_i(k)$ (potentially noisy) of the variable. Our goal is to design distributed algorithms which update each sensor $i$'s consensus variable $y_i(k)$, based on $x_i(k)$ and information only from the two immediate neighbors $i-1$ and $i+1$, such that $y_i(k)$ converges to a value which reflects spatial-temporal variations of the variable.

In this section, we focus on the case where all local measurements are time-invariant, i.e. $x_i(k) = x_i$ (a constant) for all $i, k$. The time-varying case will be addressed in Section 6, below. We consider two types of weighting schemes: exponential weighting and uniform finite window.

### *2.1 Exponential Weighting*

For computing a local average at sensor $i$, it is natural to assign larger weights to information that is spatially closer to $i$. One way of doing so is to assign an exponential weight $\rho^j$, $\rho \in (0,1)$ and $j$ a nonnegative integer, to a measurement taken at distance $j$ from $i$. For this scheme, we formulate the following problem, adopting the reasonable assumption that there is a bound $M < \infty$ such that measurement variables $|x_i| < M$ for all $i$.

*Problem 1.* Let $\rho \in (0,1)$. Design a distributed algorithm to update each sensor $i$'s consensus variable $y_i(k)$ such that

$$\lim_{k\to\infty} y_i(k) = \frac{1-\rho}{1+\rho}\left(x_i + \sum_{j=1}^{\infty} \rho^j (x_{i-j} + x_{i+j})\right). \tag{1}$$

Thus, exponentially decaying weights, at the rate $\rho$, are assigned to the information from both forward and backward directions. Note that the limit of $y_i(k)$ exists because all $x_i$ are assumed bounded. The scaling constant $\frac{1-\rho}{1+\rho}$ ensures that, if all $x_i$ are the same, $y_i(k)$ is in the limit equal to $x_i$.

We propose the following distributed algorithm to solve Problem 1. For every $i$,

$$y_i(0) = \frac{1-\rho}{1+\rho} x_i \tag{2a}$$
$$y_i(1) = y_i(0) + \rho(y_{i-1}(0) + y_{i+1}(0)) \tag{2b}$$
$$\begin{aligned} y_i(2) = y_i(1) + \rho(y_{i-1}(1) - y_{i-1}(0)) + \\ \rho(y_{i+1}(1) - y_{i+1}(0)) - \rho^2 2y_i(0) \end{aligned} \tag{2c}$$
$$\begin{aligned} y_i(k+1) = y_i(k) + \rho(y_{i-1}(k) - y_{i-1}(k-1)) + \\ \rho(y_{i+1}(k) - y_{i+1}(k-1)) - \rho^2(y_i(k-1) - y_i(k-2)),\ k \geq 2. \end{aligned} \tag{2d}$$

Each sensor $i$ needs information only from its two immediate neighbors: $y_{i-1}(k)$ and $y_{i+1}(k)$, $k = 0, 1, ....$ At each iteration $k$ ($\geq 2$), the quantities used to update $y_i(k)$ are $y_{i-1}(k) - y_{i-1}(k-1)$, $y_{i+1}(k) - y_{i+1}(k-1)$, and $y_i(k-1) - y_i(k-2)$. Thus more memories are required in this local consensus algorithm than in a global consensus algorithm, though the increase is obviously modest.

**Theorem 1** *Algorithm (2) solves Problem 1.*

*Proof.* We will show by induction on $k \geq 1$ that

$$y_i(k) = y_i(k-1) + \rho^k (y_{i-k}(0) + y_{i+k}(0)), \quad \forall i. \tag{3}$$

This leads to

$$\begin{aligned} y_i(k) &= y_i(0) + \sum_{j=1}^{k} \rho^j (y_{i-j}(0) + y_{i+j}(0)) \\ &= \frac{1-\rho}{1+\rho}\left(x_i + \sum_{j=1}^{k} \rho^j (x_{i-j} + x_{i+j})\right), \quad \forall i. \end{aligned}$$

The second equality above is due to (2a). Then taking the limit as $k \to \infty$ yields (1). That the limit exists follows from the fact that $|x_i| < M < \infty$ and $\rho \in (0,1)$.

First, it is easily verified from (2b), (2c) that (3) holds when $k = 1, 2$. Now let $k \geq 2$ and suppose (3) holds for all $k' \in [1, k]$. According to (2d) we derive

$$\begin{aligned} y_i(k+1) &= y_i(k) + \rho(\rho^k (y_{i-k-1}(0) + y_{i+k-1}(0))) + \\ &\quad \rho(\rho^k (y_{i-k+1}(0) + y_{i+k+1}(0))) - \\ &\quad \rho^2(\rho^{k-1}(y_{i-k+1}(0) + y_{i+k-1}(0))) \\ &= y_i(k) + \rho^{k+1}(y_{i-k-1}(0) + y_{i+k+1}(0)). \end{aligned} \tag{4}$$

Therefore, (3) holds for all $k \geq 1$. ■

Note from the derivation in (4) that in the scheme (2d), $y_{i-1}(k) - y_{i-1}(k-1)$ produces new information $y_{i-k-1}(0) + y_{i+k-1}(0)$ (resp. $y_{i+1}(k) - y_{i+1}(k-1)$ produces $y_{i-k+1}(0) + y_{i+k+1}(0)$), and $y_i(k-1) - y_i(k-2)$ is a correction term which cancels the redundant information $y_{i-k+1}(0) + y_{i+k-1}(0)$.

**Remark 2** *An extension of Algorithm (2) is immediate. Each sensor $i$ weights information from the backward direction differently from the forward direction, using exponential weights $\rho_b$ and $\rho_f \in (0,1)$, respectively. Here we assume that each sensor may distinguish backward direction from forward one, by means of e.g. using a one-bit compass for a line graph. Then revise Algorithm (2) as follows (omitting the similar initialization steps):*

$$\begin{aligned} y_i(k+1) = y_i(k) + \rho_b(y_{i-1}(k) - y_{i-1}(k-1)) + \\ \rho_f(y_{i+1}(k) - y_{i+1}(k-1)) - \rho_b\rho_f(y_i(k-1) - y_i(k-2)), \\ k \geq 2. \end{aligned} \tag{5}$$

*This revised algorithm yields*

$$\lim_{k\to\infty} y_i(k) = \frac{(1-\rho_b)(1-\rho_f)}{1-\rho_b\rho_f}\left(x_i + \sum_{j=1}^{\infty}(\rho_b^j x_{i-j} + \rho_f^j x_{i+j})\right).$$

*The proof of this claim is similar to that of Theorem 1.*

### *2.2 Uniform Finite Window*

An alternative to exponential weighting is to have a finite window for each sensor such that every agent's information within the window is weighted uniformly, and the information outside the window discarded. For time-invariant measurements, this is to compute the average of measurements within the window. We formulate the problem.

*Problem 2.* Let $L \geq 1$ be an integer, and $2L+1$ the length of the finite window of sensor $i$; i.e. sensor $i$ uses measurement information from $L$ neighbors in each direction. Suppose $i$ knows $L$. Design a distributed algorithm to update each $i$'s consensus variable $y_i(k)$ such that

$$y_i(L) = \frac{1}{2L+1}\left(x_i + \sum_{j=1}^{L}(x_{i-j} + x_{i+j})\right). \tag{6}$$

Thus it is required that the average of $2L+1$ measurements be computed in $L$ steps.

A variation of Algorithm (2) will solve Problem 2:

$$y_i(0) = \frac{1}{2L+1} x_i \tag{7a}$$
$$y_i(1) = y_i(0) + (y_{i-1}(0) + y_{i+1}(0)) \tag{7b}$$
$$\begin{aligned} y_i(2) = y_i(1) + (y_{i-1}(1) - y_{i-1}(0)) + \\ (y_{i+1}(1) - y_{i+1}(0)) - 2y_i(0) \end{aligned} \tag{7c}$$
$$\begin{aligned} y_i(k+1) = y_i(k) + (y_{i-1}(k) - y_{i-1}(k-1)) + \\ (y_{i+1}(k) - y_{i+1}(k-1)) - (y_i(k-1) - y_i(k-2)), \\ k \in [2, L-1]. \end{aligned} \tag{7d}$$

The memory requirement of this algorithm is the same as Algorithm (2): i.e. $y_{i-1}(k)-y_{i-1}(k-1)$, $y_{i+1}(k)-y_{i+1}(k-1)$, and $y_i(k-1)-y_i(k-2)$ are needed to update $y_i(k)$ for $k \in [2, L-1]$. Note, however, that the present algorithm terminates after $L$ steps because of finite window as well as static measurements. When measurements are time-varying (see Section 6.2 below), by contrast, the corresponding algorithm will need to keep track of temporal variations.

**Theorem 3** *Algorithm (7) solves Problem 2.*

*Proof.* Similar to the proof of Theorem 1, we derive for $k \in [1, L]$ that

$$y_i(k) = y_i(k-1) + (y_{i-k}(0) + y_{i+k}(0)), \quad \forall i. \tag{8}$$

This leads to

$$\begin{aligned} y_i(L) &= y_i(0) + \sum_{j=1}^{L}(y_{i-j}(0) + y_{i+j}(0)) \\ &= \frac{1}{2L+1}\left(x_i + \sum_{j=1}^{L}(x_{i-j} + x_{i+j})\right), \quad \forall i. \end{aligned}$$

The second equality above is due to (7a). ∎

**Remark 4** *Individual sensors may have different window lengths, $L_i \geq 1$. In this case, we impose the condition that the neighboring lengths may differ no more than one, i.e.*

$$|L_i - L_{i+1}| \leq 1, \quad |L_i - L_{i-1}| \leq 1, \quad \forall i \tag{9}$$

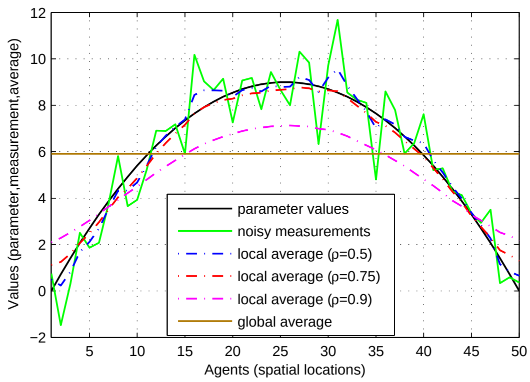
*and replace $L$ by $L_i$ throughout Algorithm (7). Then from (7d) and when $k = L_i - 1$ (the final update), we have*

$$y_i(L_i) = y_i(L_i-1) + (y_{i-1}(L_i-1) - y_{i-1}(L_i-2)) + (y_{i+1}(L_i-1) - y_{i+1}(L_i-2)) - (y_i(L_i-2) - y_i(L_i-3)).$$

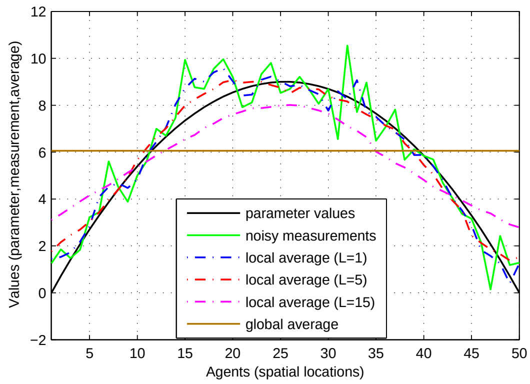
*Condition (9) ensures that both $y_{i-1}(L_i-1)$ and $y_{i+1}(L_i-1)$ exist. Hence the same argument as that validating Algorithm (7) proves that the revised algorithm with $L_i$ computes*

$$y_i(L_i) = \frac{1}{2L_i+1}x_i + \sum_{j=1}^{L_i}\left(\frac{1}{2L_{i-j}+1}x_{i-j} + \frac{1}{2L_{i+j}+1}x_{i+j}\right).$$

We have designed local consensus algorithms using two different schemes: exponential weighting and uniform finite window weighting. A simulation is displayed in Figure 1 to illustrate the performance of the algorithms (2) and (7) for different values of the respective parameters, $\rho$ or $L$. In exponential weighting, if $\rho$ is too small (e.g. $\rho = 0.5$), the algorithm has poor noise performance; while large $\rho$ (e.g. $\rho = 0.9$) substantially smooths out noise, it lowers the algorithm's performance of tracking local information. Small $L$ (e.g. $L = 1$) and large $L$ (e.g. $L = 15$) have similar effects on the performance of the finite window algorithm. Moreover, it is plausible to establish a certain relation between $\rho$ and $L$ under which the two algorithms have (roughly) the same tracking and noise performance. We will study these performance issues in the next two sections, by analyzing the algorithms' spatial frequency response and noise propagation.




(a) Exponential weighting




(b) Uniform finite window

Fig. 1. Simulation example: performance of algorithms (2) and (7) for difference values of $\rho$ or $L$, respectively. 50 agents are aligned to measure a physical parameter having a parabola spatial distribution (black curve). Measurements (green curve) are corrupted by (independent) noise of mean zero and variance one. Plots for 3 different values of $\rho$ and $L$ are displayed, showing different tracking and noise performance of the respective algorithms. Global average smooths out noise but throws away local information.

## 3 Spatial Frequency Response

The whole concept of local consensus is based on the precept that global consensus may suppress too much information that might be of interest. In effect, global (average) consensus applies a filter to spatial information which leaves the DC component intact, and completely suppresses all other frequencies. Our task in this section is to study the extent to which local consensus in contrast does not destroy all information regarding spatial variation, and the tool we use to do this is to look at a spatial frequency response. Further, there is a trade-off in using local consensus, apart from additional computational complexity as noted in Section 2: there is less mitigation–obviously–of the effect of noise. We also consider this point in the next section.

We associate with the measured variable and consensus variable sequences $\{x_i, -\infty < i < \infty\}$ and $\{y_i, -\infty < i < \infty\}$

their spatial $Z$-transforms $\mathcal{X}(Z), \mathcal{Y}(Z)$ defined by

$$\mathcal{X}(Z) = \sum_{-\infty}^{\infty} x_i Z^{-i} \quad \mathcal{Y}(Z) = \sum_{-\infty}^{\infty} y_i Z^{-i} \tag{10}$$

Spatial $Z$-transforms capture spatial frequency content, and are a potentially useful tool for analysing the relationship between measured variables and consensus variables.

Our aim is to understand how, when the measured variable sequence has spatially sinusoidal variation at frequency $\omega$, the steady state values of the consensus variables $y_i$ depend on $\rho$ and $\omega$. In a practical situation, spatial variation may not necessarily be sinusoidal. The benefit of the sinusoidal analysis is that it leads to a transfer function and hence to a concept of bandwidth for the average consensus algorithm, i.e. a notion of a spatial frequency below which variations can be reasonably tracked even when the algorithm is operating, while spatially faster variations will be suppressed or filtered out in deriving the local average consensus. We shall first consider local consensus with exponential weighting, and then local consensus with a uniform finite window.

### *3.1 Exponential Weighting*

The calculation using $Z$-transforms proceeds as follows. Starting with the steady state equation (cf. (1))

$$y_i = \frac{1-\rho}{1+\rho}(x_i + \rho x_{i-1} + \rho^2 x_{i-2} + \cdots + \rho x_{i+1} + \rho^2 x_{i+2} + \cdots) \tag{11}$$

one has

$$Z^{-i} y_i = \frac{1-\rho}{1+\rho}[x_i Z^{-i} + Z^{-1}\rho x_{i-1} Z^{-(i-1)} + Z^{-2}\rho^2 x_{i-2} Z^{-(i-2)} + \cdots + Z\rho x_{i+1} Z^{-(i+1)} + Z^2 \rho^2 x_{i+2} Z^{-(i+2)} + \cdots] \tag{12}$$

Summing from $i = -\infty$ to $\infty$ yields

$$\mathcal{Y}(Z) = \frac{1-\rho}{1+\rho}[1 + Z^{-1}\rho + Z^{-2}\rho^2 + \cdots + Z\rho + Z^2\rho^2 + \cdots]\mathcal{X}(Z) = \frac{1-\rho}{1+\rho}[1 + \frac{\rho Z^{-1}}{1-\rho Z^{-1}} + \frac{\rho Z}{1-\rho Z}]\mathcal{X}(Z)$$

or

$$\mathcal{Y}(Z) = \frac{(1-\rho)^2}{(1-\rho Z^{-1})(1-\rho Z)}\mathcal{X}(Z) \tag{13}$$

For future reference, define the transfer function

$$\mathcal{H}(Z) = \frac{(1-\rho)^2}{(1-\rho Z^{-1})(1-\rho Z)} \tag{14}$$

For $Z = \exp(j\omega)$, $\mathcal{H}(Z)$ is real and positive. However, for arbitrary $Z$ in general its value is complex. It has two poles which are mirror images through the unit circle of each other.

Now suppose that the measured variable sequence $x_i$ is sinusoidal, thus $x_i = \exp(ji\omega_0)$, where $j = \sqrt{-1}$. The associated

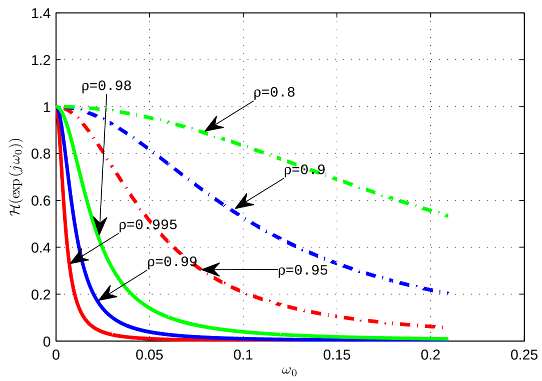



Fig. 2. Plot of $\mathcal{H}(\exp(j\omega_0))$ in (17) near origin for different values of $\rho$

$Z$-transform $\mathcal{X}(Z)$ is formally given by $\sum_{i=-\infty}^{\infty} x_i Z^{-i}$. When $Z = \exp(j\omega)$, there holds $\mathcal{X}(\exp(j\omega)) = \sum_{i=-\infty}^{\infty} \exp(ji(\omega - \omega_0)) = 2\pi\delta(\omega - \omega_0)$, where we are appealing to the fact that the delta function $\delta(x)$ is the limit of a multiple of the Dirichlet kernel

$$D_N(x) = \sum_{i=-N}^{N} \exp(jix) = \frac{\sin((N+\frac{1}{2})x)}{\sin(x/2)} \tag{15}$$

i.e. $\delta(x) = \frac{1}{2\pi}\lim_{N\to\infty} D_N(x) = \frac{1}{2\pi}\sum_{i=-\infty}^{\infty} \exp(jix)$. In formal terms, it follows from (13) and (14) that the associated $Z$-transform of the consensus variable, i.e. $\mathcal{Y}(Z)$, is given by

$$\mathcal{Y}(\exp(j\omega)) = \mathcal{H}(\exp(j\omega))2\pi\delta(\omega - \omega_0) \tag{16}$$

Equivalently, the consensus variable is also sinusoidal at frequency $\omega_0$ and with phase shift and amplitude defined by $\mathcal{H}(\exp(j\omega_0))$. The phase shift is easily checked to be zero for all $\omega_0$, and the amplitude is in fact the value of $\mathcal{H}$ itself, viz.

$$\mathcal{H}(\exp(j\omega_0)) = \frac{(1-\rho)^2}{1+\rho^2 - 2\rho\cos\omega_0} \tag{17}$$

Observe that if $\omega_0 = 0$, i.e. the measured variable is a constant or spatially invariant, then $\mathcal{H}(1) = 1$ irrespective of $\rho$, i.e. the consensus variable is the same constant – as we would expect. Observe further that for fixed $\omega_0 \neq 0$, as $\rho \to 1$, $\mathcal{H}(\exp(j\omega_0)) \to 0$, which is consistent with the fact that with $\rho = 1$, the average value of the measured variable, viz. 0, will propagate through to be the value everywhere of the consensus variable.

Observe that if $\rho$ is close to 1, i.e. $1-\rho$ is small, a straightforward calculation shows that with $\omega_0 = 1-\rho$, the value of $\mathcal{H}$ is approximately 1/2. Thus crudely, $\rho$ (for values close to 1) determines the bandwidth as $O(1-\rho)$. More generally, we observe from the Figures 2 and 3 (which show behavior near the origin and over $[0, \pi]$, respectively), that

(1) For any $\rho$, $\mathcal{H}(\exp(j\omega_0))$ is monotonic decreasing in $\omega_0$, from a value of 1 at $\omega_0 = 0$ to a value of $\frac{(1-\rho)^2}{(1+\rho)^2}$ at $\omega_0 = \pi$.

(2) For values of $1-\rho$ between zero and at least 0.2, $\mathcal{H}(\exp(j\omega_0))$ takes a value of about $\frac{1}{2}$ when $\omega_0 = 1-\rho$.

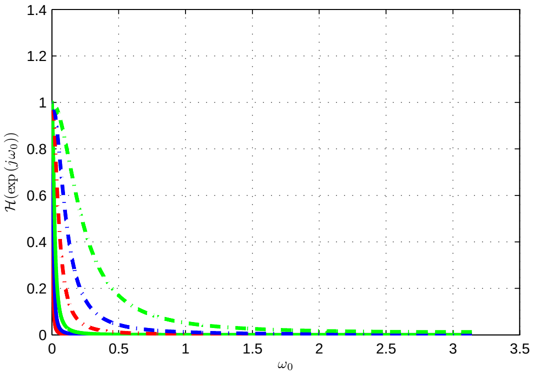



Fig. 3. Plot of $\mathcal{H}(\exp(j\omega_0))$ in (17) over $[0, \pi]$ for different values of $\rho$. The colour coding is as for Figure 2.

The above calculations assume that there are an infinite number of measuring agents. When the number is finite, it is clear that the results will undergo some variation. When the hop distance to the array boundary, call it $d$, from a particular agent, is such that $\rho^d$ is very small, the error will obviously be minor. In the vicinity of the boundary, the errors will be greater, and a kind of end effect will be observed. The results for an infinite number of agents are accordingly indicative of the results for a finite number.

### *3.2 Uniform Finite Window*

From (6), the steady-state equation in this case is

$$y_i = \frac{1}{2L+1}\sum_{k=-L}^{L} x_{i+k} \tag{18}$$

and it is straightforward to establish that

$$\mathcal{Y}(Z) = \frac{1}{2L+1}\sum_{k=-L}^{L} Z^k \mathcal{X}(Z) \tag{19}$$

The transfer function $\mathcal{H}(Z)$ is simply $\frac{1}{2L+1}\sum_{k=-L}^{L} Z^k$ so that

$$\mathcal{H}(\exp(j\omega)) = \frac{1}{2L+1}\frac{\sin((L+\frac{1}{2})\omega)}{\sin(\omega/2)} \tag{20}$$

The shape of the Dirichlet kernel is well known; $\mathcal{H}$ assumes its maximum value of 1 at $\omega = 0$, and the bandwidth is roughly $\frac{1.7}{L+1/2}$, adjustable by $L$. Evidently, the bandwidths in the exponential weighted case and the uniform finite window case are of the same order when

$$1 - \rho = \frac{1.7}{L+1/2}. \tag{21}$$

Put another way, and roughly speaking, a window length of $2L+1$ allows spatial variation of a bandwidth $\Omega$ to pass through the averaging process when $L\Omega$ is about 1.7.

## 4 Noise Propagation

As mentioned already, the noise performance when local consensus is used will be worse than that when global consensus is used. To fix ideas, suppose that for each $i$, measurement agent $i$ has its measurement contaminated by additive noise $\epsilon_i$ of zero mean and variance $\sigma^2$, with the noise at any two agents being independent. Note that bias is zero.

Then if there are $N$ agents, the error in the average will be $(1/N)\sum_{i=1}^{N}\epsilon_i$, which has variance $\frac{\sigma^2}{N}$. Obviously this goes to zero as $N \to \infty$. When the uniform finite window of length $2L+1$ is used, this same thinking shows that the error variance is $\frac{\sigma^2}{2L+1}$. Thus the signal-to-noise ratio (SNR) at each $i$ is $\frac{y_i^2(2L+1)}{\sigma^2}$, with the signal $y_i$ in (18).

Now suppose that exponential weighting is used. In local average consensus the error will be

$$\frac{1-\rho}{1+\rho}[\epsilon_i + \rho\epsilon_{i-1} + \rho^2\epsilon_{i-2} + \dots + \rho\epsilon_{i+1} + \rho^2\epsilon_{i+2} + \dots] \tag{22}$$

and the variance is given by

$$\begin{aligned} &(\frac{1-\rho}{1+\rho})^2[1 + 2\rho^2 + 2\rho^4 + \dots]\sigma^2 \qquad (23)\\ &= (\frac{1-\rho}{1+\rho})^2[\frac{2}{1-\rho^2} - 1]\sigma^2 = (1-\rho)\frac{1+\rho^2}{(1+\rho)^3}\sigma^2 \end{aligned}$$

This lies in the interval $(\frac{1}{4}(1-\rho)\sigma^2, (1-\rho)\sigma^2)$, and for $\rho$ close to 1, the error is approximately equal to the lower bound. Indeed, the closer $\rho$ is to 1, the less is the error variance. Note that in this case the SNR at each $i$ is $\frac{y_i^2}{\sigma^2(1-\rho)}\frac{(1+\rho)^3}{1+\rho^2}$, with the signal $y_i$ in (11). It is not hard to verify that a uniform finite window of length $2L+1$ and an exponential weighting of $\rho = \frac{2L-3}{2L+1}$ yield the same variance. Equivalently, this condition is $1-\rho = \frac{2}{L+1/2}$, which means that exponential weighting and uniform finite window weighting, if they achieve the same bandwidth (cf. (21)), also have approximately the same noise performance. The same condition incidentally says that $\rho^L \approx e^{-1}$, implying that the finite window width with uniform weighting has width determined by the number of steps over which the exponential weighting dies off by a factor of $e$. These observations also mean, unsurprisingly, that when $L$ or $\rho$ are adjusted, noise variance is proportional to bandwidth.

## 5 Generalizations

### *5.1 Arbitrary Weighting*

To this point, we have considered two special types of weights. It is at least of academic interest to consider what might happen with essentially *arbitrary* weights. These might for example reflect known and nonuniform spacings between agents. We adopt the following assumption.

**Assumption 5** *Let $a_{ij} \neq 0$ for all $i, j$. For every $i$, the sum $a_{ii}x_i + \sum_{j=1}^{\infty}(a_{i,i-j}x_{i-j} + a_{i,i+j}x_{i+j})$ is finite, and $K := a_{ii} + \sum_{j=1}^{\infty}(a_{i,i-j} + a_{i,i+j})$.*

*Problem 3.* Design a distributed algorithm to update each sensor $i$'s consensus variable $y_i(k)$ such that

$$\lim_{k\to\infty} y_i(k) = \frac{1}{K}\left(a_{ii}x_i + \sum_{j=1}^{\infty}(a_{i,i-j}x_{i-j} + a_{i,i+j}x_{i+j})\right). \tag{24}$$

The constant $1/K$ ensures again that, if all $x_i$ are the same, $y_i(k)$ is in the limit equal to $x_i$.

To solve Problem 3, we consider a modified approach: Let each sensor $i$ have two additional consensus variables, $y_i^F(k)$ and $y_i^B(k)$; $y_i^F(k)$ (resp. $y_i^B(k)$) is updated based on $x_i$ and information from the forward neighbor $i+1$ (resp. the backward neighbor $i-1$). This approach separates the updates of consensus variables between the forward and the backward directions. As we will see, the separation effectively avoids term cancelations needed in the algorithms in Section 2, which we find difficult in the case of arbitrary weights.

Now using the two consensus variables $y_i^F(k)$ and $y_i^B(k)$, we present the following distributed algorithm. For all $i$,

$$y_i^F(0) = y_i^B(0) = \frac{1}{K}a_{ii}x_i \tag{25a}$$

$$y_i^F(1) = y_i^F(0) + \frac{a_{i,i+1}}{a_{i+1,i+1}}y_{i+1}^F(0) \tag{25b}$$

$$y_i^B(1) = y_i^B(0) + \frac{a_{i,i-1}}{a_{i-1,i-1}}y_{i-1}^B(0)$$

$$y_i^F(2) = y_i^F(1) + \frac{a_{i,i+2}}{a_{i+1,i+2}}(y_{i+1}^F(1) - y_{i+1}^F(0)) \tag{25c}$$

$$y_i^B(2) = y_i^B(1) + \frac{a_{i,i-2}}{a_{i-1,i-2}}(y_{i-1}^B(1) - y_{i-1}^B(0))$$

$$y_i^F(k+1) = y_i^F(k) + \frac{a_{i,i+k+1}}{a_{i+1,i+k+1}}(y_{i+1}^F(k) - y_{i+1}^F(k-1)) \tag{25d}$$

$$y_i^B(k+1) = y_i^B(k) + \frac{a_{i,i-k-1}}{a_{i-1,i-k-1}}(y_{i-1}^B(k) - y_{i-1}^B(k-1)),\ k \geq 2.$$

In the above algorithm, each sensor $i$ requires two consensus variables and needs to know the weights used by its two neighbors, in addition to the memory requirement of the algorithms in Section 2. Finally, values of $y_i^F(k)$ and $y_i^B(k)$ are glued together to produce $y_i(k)$ as follows:

$$y_i(k) = y_i^F(k) + y_i^B(k) - \frac{1}{K}a_{ii}x_i,\ \forall k \geq 0. \tag{26}$$

The last term above serves to correct that the initial $(1/K)a_{ii}x_i$ value in (25a) is added twice.

**Theorem 6** *Let Assumption 5 hold. Then Algorithm (25)-(26) solves Problem 3.*

*Proof.* First, we show by induction on $k \geq 1$ that for all $i$,

$$y_i^F(k) = y_i^F(k-1) + \frac{1}{K}a_{i,i+k}x_{i+k}. \tag{27}$$

It is easily verified from (25b), (25c) that (27) holds when $k = 1, 2$. Now let $k \geq 2$ and suppose (27) holds for $k$. According to (25d) we derive $y_i^F(k+1) = y_i^F(k) + \frac{1}{K}a_{i,i+k+1}x_{i+k+1}$. Therefore, (27) holds for all $k \geq 1$, and leads to

$$y_i^F(k) = \frac{1}{K}\left(a_{ii}x_i + \sum_{j=1}^{k} a_{i,i+j}x_{i+j}\right),\quad \forall i.$$

Similarly, for $y_i^B(k)$, we derive

$$y_i^B(k) = \frac{1}{K}\left(a_{ii}x_i + \sum_{j=1}^{k} a_{i,i-j}x_{i-j}\right),\quad \forall i.$$

Now by (26),

$$y_i(k) = \frac{1}{K}\left(a_{ii}x_i + \sum_{j=1}^{k}(a_{i,i-j}x_{i-j} + a_{i,i+j}x_{i+j})\right),\quad \forall i.$$

Then taking the limit as $k \to \infty$ yields (24). That the limit exists follows from Assumption 5. ■

### *5.2 Random Spacing*

If the arbitrary weights studied in the previous subsection reflect nonuniform distances between successive sensors, we may assume that these distances are random, in accordance with some probability law. Two different possibilities are that (a) they are Poisson distributed, say with intensity 1 for convenience, or (b) the inter sensor distances are uniformly distributed in an interval $[1-\eta, 1+\eta]$ where $\eta$ is known. Different physical mechanisms could typically lead to these two situations. In the first case, sensor distances are independent. In the second case, we make the explicit assumption that inter sensor distances are independent random variables.

Based on the treatment already derived for the case corresponding to uniform spacing in Section 2.1, where a weighting of $\rho^d$ applies at a given sensor to the measurement passed to it and made at a sensor $d$ units away, we suggest that the relevant weighting to apply to the measurement collected at sensor $j$ and used at sensor $i < j$ is $\rho^{d_{ij}} := \rho^{d_{i,i+1}+d_{i+1,i+2}+\cdots+d_{j-1,j}}$, with $d_{i,i+j}$ denoting the distance between sensors $i$ and $i+j$.

The full expression for the average consensus variable at node $i$ is then

$$y_i = K[x_i + \sum_{j=1}^{\infty}\rho^{d_{i,i+j}}x_{i+j} + \sum_{j=1}^{\infty}\rho^{d_{i,i-j}}x_{i-j}] \tag{28}$$

Here $K$ is a normalization constant. Next, we determine $K$.

In the deterministic case (Section 2.1), the normalization constant $(\frac{1-\rho}{1+\rho})$ was chosen to ensure that if all measured variables had the same value, $a$ say, then the average consensus variable also took the value $a$. In the random case, we can seek this requirement. But it turns out that we can only assure that $E[y_i] = a$. It would then be relevant to consider the question of the variance in $y_i$. This is also covered below.

Let us now assume $a = 1$ for convenience. Then

$$y_i = K[1 + \sum_{j=1}^{\infty}\rho^{d_{i,i+j}} + \sum_{j=1}^{\infty}\rho^{d_{i,i-j}}] \tag{29}$$

Define two random variables

$$u = \sum_{j=0}^{\infty}\rho^{d_{i,i+j}},\quad v = \sum_{j=0}^{\infty}\rho^{d_{i,i-j}} \tag{30}$$

(Take $d_{i,i} = 0$, so that the first summand in each case is 1.) Then $u, v$ have the same distribution and are independent. It is obvious that

$$y_i = K[u + v - 1] \tag{31}$$

This equation makes clear that $y_i$ is indeed a random variable, so that $K$ can only be chosen to ensure that $E[y_i] = 1$. Now observe further that

$$u = 1 + \rho^{d_{i,i+1}} \sum_{j=1}^{\infty} \rho^{d_{i+1,i+j}} = 1 + \rho^{d_{i,i+1}} w \qquad (32)$$

where, crucially, $w$ evidently has the same distribution as $u$, but is independent of the random variable $\rho^{d_{i,i+1}}$. Hence there holds $E[u] = 1 + E[\rho^{d_{i,i+1}}]E[u]$, whence $E[u] = (1 - E[\rho^{d_{i,i+1}}])^{-1}$ and then to assure $E[y_i] = 1$, equation (31) implies that we need

$$K = \frac{1 - E[\rho^{d_{i,i+1}}]}{1 + E[\rho^{d_{i,i+1}}]} \qquad (33)$$

Now suppose the distribution of $d_{i,i+1}$ is Poisson with intensity 1, for which the probability density is $e^{-d}$. The expected value of $\rho^{d_{i,i+1}}$ is then $[1 - \log \rho]^{-1}$, so that

$$K = \frac{-\log \rho}{2 - \log \rho} \qquad (34)$$

We remark that when $1 - \rho$ is small, both $K$ and the expression applicable in the deterministic case, viz. $\frac{1-\rho}{1+\rho}$, are approximately $\frac{1}{2}(1 - \rho)$.

If the distribution of $d_{i,i+1}$ is uniform in $[1 - \eta, 1 + \eta]$, then the expected value of $\rho^{d_{i,i+1}}$ is $\frac{1}{2\eta \log \rho}[\rho^{1+\eta} - \rho^{1-\eta}]$, (the limit of which is $\rho$ when $\eta \to 0$, as expected). The value of $K$ is:

$$K = \frac{2\eta \log \rho - (\rho^{1+\eta} - \rho^{1-\eta})}{2\eta \log \rho + \rho^{1+\eta} - \rho^{1-\eta}}. \qquad (35)$$

Once again, one can verify that when $1 - \rho$ is small, the expression is approximately $\frac{1}{2}(1 - \rho)$.

Now since we can only assure in the event all $x_i$ assume the value that $E[y_i]$ takes that value, rather than $y_i$ itself, it is of interest to consider what the error might be. Guidance as to the error follows from the variance $E(y_i - E[y_i])^2$. We can work out the variance also, in the following way. From (31) and the fact that $u, v$ are independent but with the same distribution, there follows, in obvious notation $\sigma_y^2 = 2K^2\sigma_u^2$.

Now if $x, y$ are two independent random variables with $z = xy$, there holds $\sigma_z^2 = \sigma_x^2\sigma_y^2 + \sigma_x^2 E[y]^2 + E[x]^2\sigma_y^2$, and using this it follows from (32) and the fact that $\xi := \rho^{d_{i,i+1}}$ and $w$ are independent, $w$ having the same distribution as $u$, that $\sigma_u^2 = \sigma_\xi^2\sigma_u^2 + \sigma_\xi^2 E[u]^2 + E[\xi]^2\sigma_u^2$, or

$$\sigma_u^2 = \frac{\sigma_\xi^2 E[u]^2}{1 - \sigma_\xi^2 - E[\xi]^2} = \frac{\sigma_\xi^2 E[u]^2}{1 - E[\xi^2]} \qquad (36)$$

It is straightforward to check that

$$E[\xi^2] = \frac{1}{1 - 2\log \rho}, \quad \sigma_\xi^2 = \frac{1}{1 - 2\log \rho} - \frac{1}{(1 - \log \rho)^2} \qquad (37)$$
$$\sigma_u^2 = -\frac{1}{2\log \rho}, \quad \sigma_y^2 = 2K^2\sigma_u^2 = -\frac{\log \rho}{(2 - \log \rho)^2}$$

Thus $\sigma_y^2$ is of the order of $-\log \rho$. When $x := 1 - \rho$, this is approximately $x$. Comparing this variance with the error variance arising in $y_i$ with deterministic spacing but error variance $\sigma^2 = 1$ of additive noise perturbing each measured variable, we see that the error is of a similar magnitude.

# 6 Local Consensus with Time-Varying Measurements

We have so far considered time-invariant local measurements. In practice, however, most measured variables are time-varying: e.g. temperature, pollution, and current in power lines. In this section, we consider that each measurement variable $x_i(k)$ is time-varying, i.e. a function of time $k$, and design distributed algorithms to track temporal variations of measurements, in addition to spatial variations.

Note that in typical studies of global average consensus, it is not common to postulate that local variables change over time. Nevertheless, convergence rates are often considered, being identified as exponential, and there are numerous results that seek to identify such rates (see e.g. [19, 23]). The rates themselves are indicative of the bandwidth of variation of measured variables whose average can be tracked by the global consensus algorithms.

In the sequel, we will again consider the two schemes: first exponential weighting, and then uniform finite window.

## 6.1 *Exponential Weighting*

Henceforth, we shall assume that there is a bound $M < \infty$ such that measured variables $|x_i(k)| < M$ for all $i, k$.

*Problem 3.* Let $\rho \in (0, 1)$. Design a distributed algorithm to update each sensor $i$'s consensus variable $y_i(k)$ such that

$$y_i(k) = \frac{1-\rho}{1+\rho}\Big(x_i(k) + \sum_{j=1}^{k} \rho^j (x_{i-j}(k-j) + x_{i+j}(k-j))\Big). \qquad (38)$$

By the assumption made above, $|y_i(k)|$ is finite for all $i, k$. In (38), an exponential weight $\rho^j$ is applied to measurements from $j$ steps away sensors in both directions with $j$ time delay. In this way temporal changes of $x_i$ are taken into account. Note that $y_i(k)$ in (38) is identical to (1) in the limit if the measurements are actually constant.

Extending Algorithm (2), we propose the following distributed algorithm, which differs from (2) by inclusion of additional terms reflecting temporal changes in local measurement values.

$$y_i(0) = \lambda x_i(0), \quad \lambda := \frac{1-\rho}{1+\rho} \qquad (39a)$$
$$y_i(1) = y_i(0) + \rho(y_{i-1}(0) + y_{i+1}(0)) + \lambda(x_i(1) - x_i(0)) \qquad (39b)$$
$$y_i(2) = y_i(1) + \rho(y_{i-1}(1) - y_{i-1}(0)) + \rho(y_{i+1}(1) - y_{i+1}(0)) - \rho^2 2y_i(0) + \lambda(x_i(2) - x_i(1)) \qquad (39c)$$
$$y_i(k+1) = y_i(k) + \rho(y_{i-1}(k) - y_{i-1}(k-1)) + \rho(y_{i+1}(k) - y_{i+1}(k-1)) - \rho^2(y_i(k-1) - y_i(k-2)) + \lambda(x_i(k+1) - x_i(k)) - \rho^2\lambda(x_i(k-1) - x_i(k-2)), \ k \geq 2. \qquad (39d)$$

This algorithm reduces to Algorithm (2) for time-invariant measurements. Note that each sensor $i$ needs information only from its two immediate neighbors: $y_{i-1}(k)$ and $y_{i+1}(k)$, $k = 0, 1, \ldots$. Note that sensor $i$ does not need its neighbors' measurement variables $x_{i-1}(k)$ and $x_{i+1}(k)$. Compared to Algorithm (2), two additional quantities (requiring further modest increase in local memory) are used to update $y_i(k)$: $x_i(k+1) - x_i(k)$ and $x_i(k-1) - x_i(k-2)$; both represent changes in local measurements at different times. As we will see below, $x_i(k+1) - x_i(k)$ provides new information, while $x_i(k-1) - x_i(k-2)$ is used as a correction term.

**Theorem 7** *Algorithm (39) solves Problem 3.*

*Proof.* It is easily verified from (39b) that $y_i(1) = \lambda(x_i(1) + \rho(x_{i-1}(0) + x_{i+1}(0)))$ and from (39c) that

$$\begin{aligned} y_i(2) &= y_i(1) + \rho^2(y_{i-2}(0) + y_{i+2}(0)) + \lambda\Big[(x_i(2) - x_i(1)) \\ &\quad + \rho\big((x_{i-1}(1) - x_{i-1}(0)) + (x_{i+1}(1) - x_{i+1}(0))\big)\Big] \quad (40a) \\ &= \lambda\Big(x_i(2) + \rho(x_{i-1}(1) + x_{i+1}(1)) \\ &\quad + \rho^2(x_{i-2}(0) + x_{i+2}(0))\Big) \quad (40b) \end{aligned}$$

By (40a) we obtain the expressions of $y_{i-1}(2) - y_{i-1}(1)$ and $y_{i+1}(2) - y_{i+1}(1)$; also by (39b) we have $y_i(1) - y_i(0)$. Substituting these three terms into (39d) yields

$$\begin{aligned} y_i(3) &= y_i(2) + \rho^3(y_{i-3}(0) + y_{i+3}(0)) + \lambda\Big[(x_i(3) - x_i(2)) \\ &\quad + \rho\big((x_{i-1}(2) - x_{i-1}(1)) + (x_{i+1}(2) - x_{i+1}(1))\big) + \\ &\quad \rho^2\big((x_{i-2}(1) - x_{i-2}(0)) + (x_{i+2}(1) - x_{i+2}(0))\big)\Big]. \quad (41) \end{aligned}$$

In deriving the second equality above, the terms $\rho^3((y_{i-1}(0) + y_{i+1}(0)))$ and $2\rho^2\lambda(x_i(1) - x_i(0))$ are canceled. Now substituting the expression (40b) of $y_i(2)$ into (41), and canceling the terms $\lambda x_i(2)$, $\rho\lambda(x_{i-1}(1) + x_{i+1}(1))$, and $\rho^2\lambda(x_{i-2}(0) + x_{i+2}(0))$, we derive

$$\begin{aligned} y_i(3) &= \lambda\Big(x_i(3) + \rho(x_{i-1}(2) + x_{i+1}(2)) + \\ &\quad \rho^2(x_{i-2}(1) + x_{i+2}(1)) + \rho^3(x_{i-3}(0) + x_{i+3}(0))\Big). \end{aligned}$$

By the same procedure, inductively we can derive $y_i(k)$ for $k = 4, 5, ...$, and conclude that (38) holds for all $k$. ■

### 6.2 Uniform Finite Window

The finite window case with time-varying measurements is challenging, because all information outside the window has to be discarded, and temporal variations of information within the window have to be tracked. We state the problem:

*Problem 4.* Let $L \geq 1$ be an integer, and $2L+1$ the length of the finite window of sensor $i$; i.e. sensor $i$ uses measurement information from $L$ neighbors in each direction. Suppose $i$ knows $L$. Design a distributed algorithm to update each $i$'s consensus variable $y_i(k)$ such that

$$\begin{aligned} y_i(k) &= \frac{1}{2L+1}\left(x_i(k) + \sum_{j=1}^{k}(x_{i-j}(k-j) + x_{i+j}(k-j))\right) \\ &\qquad\qquad \text{if } k \leq L; \\ y_i(k) &= \frac{1}{2L+1}\left(x_i(k) + \sum_{j=1}^{L}(x_{i-j}(k-j) + x_{i+j}(k-j))\right) \\ &\qquad\qquad \text{if } k > L. \end{aligned} \quad (42)$$

The explanation for the time arguments associated with $x_{i-j}$ and $x_{i+j}$ on the right of (42) is as follows. At each time step, values can be 'passed' by exactly one hop. Hence, it takes $j$ time instances for a measured variable at sensor $i-j$ to be perceived at sensor $j$. Therefore the consensus variable $y_i(k)$ can depend on $x_{i-j}(k-j)$ (resp. $x_{i+j}(k-j)$ but no later value of $x_{i-j}(k-j)$ (resp. $x_{i+j}(k-j)$). Note that if the measurements are actually constant, then $y_i(k)$ in (42) is identical with (6) for all $k \geq L$.

The distributed algorithm we design to solve Problem 4 has several features. First, it needs an additional vector of variables $z_i = [z_{i0}\ z_{i1}\ \cdots\ z_{i(L)}]^T$ of $L+1$ components for each sensor $i$, and $z_i$ needs to be updated along with consensus variable $y_i$ and communicated to the two immediate neighbors $i-1$ and $i+1$. Second, the scheme for each component of $z_i$ is similar to Algorithm (7). Finally, we will see that the $j$th component $z_{ij}$, $j \in [0, L]$, contributes to tracking all local measurements $x_l(k)$, $l \in [i-L, i+L]$, in the finite window for time $k = j \pmod{L+1}$.

We first present the update scheme for vector $z_i$ (c.f. Algorithm (7)). For every $j \in [0, L]$, if $k < j$,

$$z_{ij}(k) = 0; \quad (43)$$

if $k \geq j$ and $k = j \pmod{L+1}$,

$$\begin{aligned} z_{ij}(k) &= \frac{1}{2L+1}x_i(k), && (44a) \\ z_{ij}(k+1) &= z_{ij}(k) + (z_{(i-1)j}(k) + z_{(i+1)j}(k)) && (44b) \\ z_{ij}(k+2) &= z_{ij}(k+1) + (z_{(i-1)j}(k+1) - z_{(i-1)j}(k)) && (44c) \\ &\quad + (z_{(i+1)j}(k+1) - z_{(i+1)j}(k)) - 2z_{ij}(k) \\ z_{ij}(k+3) &= z_{ij}(k+2) + (z_{(i-1)j}(k+2) - z_{(i-1)j}(k+1)) && (44d) \\ &\quad + (z_{(i+1)j}(k+2) - z_{(i+1)j}(k+1)) - (z_{ij}(k+1) - z_{ij}(k)) \\ &\ \vdots && (44e) \\ z_{ij}(k+L) &= z_{ij}(k+L-1) + && (44f) \\ &\quad (z_{(i-1)j}(k+L-1) - z_{(i-1)j}(k+L-2)) + \\ &\quad (z_{(i+1)j}(k+L-1) - z_{(i+1)j}(k+L-2)) - \\ &\quad (z_{ij}(k+L-2) - z_{ij}(k+L-3)) \end{aligned}$$

The update of each component $z_{ij}$, $j \in [0, L]$, is *periodic* with period $L+1$ for $k \geq j$. The following is the update scheme for consensus variable $y_i$.

$$\begin{aligned} y_i(k) &= z_{ij}(k) + \sum_{l=0, l\neq j}^{L}(z_{il}(k) - z_{il}(k-1)), \\ &\qquad\qquad j = k \pmod{L+1}. \end{aligned} \quad (45)$$

We now state the main result of this subsection.

**Theorem 8** *Algorithm (43)-(45) solves Problem 4.*

*Proof.* First, at $k = 0$, we have from (43), (44a) that $z_{i0}(0) = (1/(2L+1))x_i(0)$ and $z_{ij}(0) = 0$, $j = 1, ..., L$. So by (45) $y_i(0) = z_{i0}(0) = (1/(2L+1))x_i(0)$.

Let $k \geq 1$ and fix $j = k \pmod{L+1}$. Similar to the proof of Theorem 3, in particular Equation (8), we derive

$$z_{ij}(k) = \frac{1}{2L+1} x_i(k) \qquad \text{(again by (44a))}$$

$$z_{i(j-1)}(k) = z_{i(j-1)}(k-1) + \frac{1}{2L+1}(x_{i-1}(k-1) + x_{i+1}(k-1))$$

$$z_{i(j-2)}(k) = z_{i(j-2)}(k-1) + \frac{1}{2L+1}(x_{i-2}(k-2) + x_{i+2}(k-2))$$

$$\vdots$$

$$z_{i0}(k) = z_{i0}(k-1) + \frac{1}{2L+1}(x_{i-j}(k-j) + x_{i+j}(k-j)).$$

Now if $k \leq L$ (thus $j = k$), then by (43) $z_{i(j+1)}(k) = \cdots = z_{i(L)}(k) = 0$. Therefore by (45),

$$\begin{aligned} y_i(k) &= z_{ij}(k) + \sum_{l=0}^{j-1}(z_{il}(k) - z_{il}(k-1)) \\ &= \frac{1}{2L+1}\left(x_i(k) + \sum_{j=1}^{k}(x_{i-j}(k-j) + x_{i+j}(k-j))\right). \end{aligned}$$

This is the first part of (42).

If $k > L$, then again similar to Equation (8) we derive

$$\begin{aligned} z_{i(L)}(k) = z_{i(L)}(k-1) + \frac{1}{2L+1}(x_{i-j-1}(k-j-1) + \\ x_{i+j+1}(k-j-1)) \end{aligned}$$

$$\vdots$$

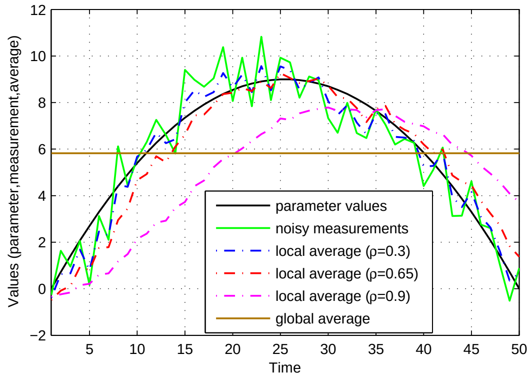
$$\begin{aligned} z_{i(j+1)}(k) = z_{i(j+1)}(k-1) + \frac{1}{2L+1}(x_{i-L}(k-L) + \\ x_{i+L}(k-L)). \end{aligned}$$

Therefore by (45),

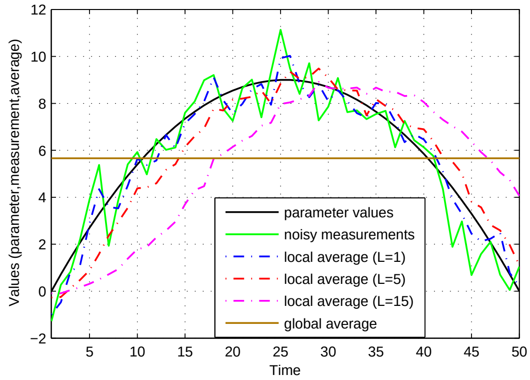
$$\begin{aligned} y_i(k) &= z_{ij}(k) + \sum_{l=0, l\neq j}^{L}(z_{il}(k) - z_{il}(k-1)) \\ &= \frac{1}{2L+1}\left(x_i(k) + \sum_{j=1}^{L}(x_{i-j}(k-j) + x_{i+j}(k-j))\right). \end{aligned}$$

This is the second part of (42), and the proof is complete. ∎

We have designed exponential weighting and uniform finite window local consensus algorithms for time-varying measured variables. A simulation is displayed in Figure 4 to illustrate the performance of the algorithms (39) and (43)-(45) for different values of the respective parameters, $\rho$ or $L$. In exponential weighting, if $\rho$ is too small (e.g. $\rho = 0.3$), the algorithm has poor noise performance; while large $\rho$ (e.g. $\rho = 0.9$) substantially smooths out noise, it causes time lag for the algorithm to track local information. Small $L$ (e.g. $L = 1$) and large $L$ (e.g. $L = 15$) have similar effects on the performance of the finite window algorithm. In the next section, we study these performance issues by analyzing the frequency response for these two local consensus algorithms, with respect to both spatial and temporal variations.




(a) Exponential weighting




(b) Uniform finite window

Fig. 4. Simulation example: performance of algorithms (39) and (43)-(45) for difference values of $\rho$ or $L$, respectively. A physical parameter to be measured has a parabola temporal variation (black curve), assuming no spatial variation. Time-varying measurements (green curve) are corrupted by (independent) noise of mean zero and variance one. Plots for 3 different values of $\rho$ and $L$ are displayed, showing different tracking and noise performance of the respective algorithms. Global average smooths out noise but throws away local information.

## 7 Temporal Frequency Response

In this section, we consider the question of how changes in the measured variables propagate to become changes in the consensus variables. Specifically, we consider how sinusoidal variations in measured variables reflects through, as a function of frequency, to time-variation of the local consensus variables. As with the case of spatial variation, we are interested in understanding what speed of variations might be trackable by the local consensus algorithm, through the identification of a transfer function and associated bandwidth. This question is rather understudied for global consensus.

We shall first consider a special situation, viz. one where there is no spatial variation, but merely sinusoidal time-variation, i.e. for all $i$, there holds $x_i(k) = e^{j\omega_0 k}$. Recall that in studying spatial variation, we considered the special case where there was no time-variation. Studying these special situations allow clearer examination of the separate effects of time-variation and spatial variation.

Now when values are independent of the spatial index $i$, equation (39d) yields

$$y_i(k+1) = (1+2\rho)y_i(k) - (2\rho+\rho^2)y_i(k-1) + \rho^2 y_i(k-2) + \frac{1-\rho}{1+\rho}[x_i(k+1) - x_i(k) + -\rho^2(x_i(k-1) - x_i(k-2))]$$

The transfer function linking the measured to consensus variables is then

$$\mathcal{K}(e^{j\omega}) = \frac{\frac{1-\rho}{1+\rho}[1 - e^{-j\omega} - \rho^2(e^{2j\omega} - e^{3j\omega})]}{1 - (1+2\rho)e^{-j\omega} + (2\rho+\rho^2)e^{-2j\omega} - \rho^2 e^{-3j\omega}} = \frac{\frac{1-\rho}{1+\rho}[1 - \rho^2 e^{-2j\omega}]}{(1-\rho e^{-j\omega})^2} \quad (46)$$

Evidently, the transfer functions $\mathcal{K}(e^{j\omega})$ and $\mathcal{H}(e^{j\omega})$ in (17) are not that different in terms of the way their magnitude depends on $\omega$ and $\rho$. Indeed, once again one can verify that if $1-\rho$ is small and $\omega = 1-\rho$, then $\mathcal{K}$ is approximately $1/2$. So the spatial and temporal bandwidths are about the same. This appears consistent with the assumption that a spatial progression of one hop occurs in each time update, i.e. values propagate with effectively unit velocity. Of course, the poles and zeros for the spatial transfer function lie symmetrically inside and outside the unit circle, in contrast to the time-based frequency response.

The treatment of time variation when the uniform finite window approach is being used is also simple. Analogously to (46), we can obtain for

$$\mathcal{K}'(e^{j\omega}) = \frac{1}{2L+1}[1 + 2(e^{-j\omega} + \cdots + e^{-Lj\omega})] \quad (47)$$

When $\frac{2}{L+1/2}$ is small (this corresponds to the condition $1-\rho$ is small for the exponential weighting case), we derivethat the frequency at which $|\mathcal{K}'(e^{j\omega})|$ assumes the value $1/2$ is approximately $\frac{4}{L+1/2}$.

We remark that the considerations applicable to spatial variation without temporal variation or to temporal variation without spatial variation will apply (because of the linearity of the whole system) to a situation where both types of variation are present in the measured variables. Thus if the measured variable variation places them in the spatial bandwidth and outside temporal bandwidth, or the reverse, the averaging process will attenuate or suppress the variation.

## 8 Conclusions

We have studied local average consensus in distributed measurement of a variable using 1D sensor networks. Distributed local consensus algorithms have been designed to address first the case where the measured variable has spatial variation but is constant in time, and then the case where the measured variable has both spatial and temporal variations. In Table 1 we summarize the memory requirements of designed algorithms. Two schemes for local average computation have been employed: exponential weighting and uniform finite window. Further, we have analyzed temporal-spatial frequency response and noise propagation associated to the algorithms. Arbitrary updating weights and random spacing between sensors have been analyzed in the algorithms.

In ongoing work we have studied two dimensional arrays. With a uniform grid, results rather like those with fixed $\rho$ and $L$ can be obtained, but for a general two dimensional array, a theory appears needed and is currently under development.

## References


[1] M. Arik and O. B. Akan. Collaborative mobile target imaging in UWB wireless radar sensor networks. *IEEE J. Selected Areas in Communications*, 28(6):950–961, 2010.

[2] H. Bai, M. Arcak, and J. T. Wen. Adaptive motion coordination: using relative velocity feedback to track a reference velocity. *Automatica*, 45(4):1020–1025, 2009.

[3] H. Bai, R. A. Freeman, and K. M. Lynch. Robust dynamic average consensus of time-varying inputs. In *Proc. 49th IEEE Conf. Decision and Control*, pages 3104–3109, Atlanta, GA, 2010.

[4] K. Cai, B. D.O. Anderson, C. Yu, and G. Mao. Local average consensus in distributed measurement of spatial-temporal varying parameters: 1d case. In *Proc. 52nd IEEE Conf. on Decision and Control*, pages 2139–2144, Florence, Italy, 2013.

[5] K. Cai and H. Ishii. Average consensus on general strongly connected digraphs. *Automatica*, 48(11):2750–2761, 2012.

[6] Y. Cao, W. Ren, and Y. Li. Distributed discrete-time coordinated tracking with a time-varying reference state and limited communication. *Automatica*, 45(5):1299–1305, 2009.

[7] Y. Chen and J. Hwang. A power-line-based sensor network for proactive electrical fire precaution and early discovery. *IEEE Trans. Power Delivery*, 23(2):633–639, 2008.

[8] R. A. Freeman, P. Yang, and K. M. Lynch. Stability and convergence properties of dynamic average consensus estimators. In *Proc. 45th IEEE Conf. Decision and Control*, pages 338–343, San Diego, CA, 2006.

[9] H. Gharavi and S.P. Kumar (Guest Editors). Special issue on sensor networks and applications. *Proc. IEEE*, 91(8), 2003.

[10] Y. Hong, J. Hu, and L. Gao. Tracking control for multi-agent consensus with an active leader and variable topology. *Automatica*, 42(7):1177–1182, 2006.

[11] B. Huang, C. Yu, and B.D.O. Anderson. Analyzing localization errors in one-dimensional sensor networks. *Sig. Processing*, 92(2):427438, 2012.

[12] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents sing nearest neighbor rules. *IEEE Trans. Autom. Control*, 48(6):988–1001, 2003.

[13] S. Kar and J. Moura. Distributed consensus algorithms in sensor networks with imperfect communication: link failures and channel noise. *IEEE Trans. Sig. Processing*, 57(1):355–369, 2009.

[14] S. Kar and J. M.F. Moura. Convergence rate analysis of distributed gossip (linear parameter) estimation: fundamental limits and tradeoffs. *IEEE J. Selected Topics in Sig. Processing*, 5(4):674–690, 2011.

[15] X. Liu, W. Lu, and T. Chen. Consensus of multi-agent systems with unbounded time-varying delays. *IEEE Trans. Autom. Control*, 55(10):2396–2401, 2010.

[16] C. G. Lopes and A. H. Sayed. Diffusion least-mean squares over adaptive networks: formulation and performance analysis. *IEEE Trans. Sig. Processing*, 56(7):3122–3136, 2008.

[17] S. Lovisari and S. Zampieri. Performance metrics in the consensus problem: a survey. In *Proc. 4th IFAC Symp. on System, Structure and Control*, pages 324–335, 2010.

Table 1
Memory requirements of local consensus algorithms

| Time-invariant measurements | | | Time-varying measurements | |
|---|---|---|---|---|
| Exponential weighting | Uniform finite window | Arbitrary weights | Exponential weighting | Uniform finite window |
| 4 | 4 | 5 | 6 | 4(L+1)+L |


[18] R. Olfati-Saber. Distributed kalman filtering for sensor networks. In *Proc. 46th IEEE Conf. on Decision and Control*, pages 5492–5498, New Orleans, LA, 2007.

[19] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proc. IEEE*, 95(1):215–233, 2007.

[20] R. Olfati-Saber and J. S. Shamma. Consensus filters for sensor networks and distributed sensor fusion. In *Proc. 44th IEEE Conf. on Decision and Control and Eur. Control Conf.*, pages 6698–6703, Seville, Spain, 2005.

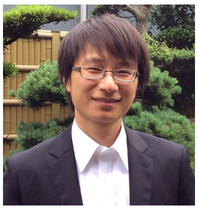
[21] W. Ren, R. W. Beard, and E. M. Atkins. Information consensus in multivehicle cooperative control. *IEEE Control Systems Magazine*, 27(2):71–82, 2007.

[22] K. Topley and V. Krishnamurthy. Average-consensus in a deterministic framework–part I: strong connectivity. *IEEE Trans. Sig. Processing*, 60(12):6590–6603, 2012.

[23] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.

[24] L. Xiao, S. Boyd, and S.-J. Kim. Distributed average consensus with least-mean-square deviation. *J. of Parallel and Distributed Computing*, 67(1):33–46, 2007.

[25] S. Yoon, W. Ye, J. Heidemann, B. Littlefield, and C. Shahabi. Swats: wireless sensor networks for steamflood and waterflood pipeline monitoring. *IEEE Trans. Sig. Processing*, 55(2):684–696, 2007.

**Kai Cai** received the B. Eng. degree in Electrical Engineering from Zhejiang University (China) in 2006, the M.A.Sc. degree in Electrical and Computer Engineering from the University of Toronto (Canada) in 2008, and the Ph.D. degree in Systems Science from Tokyo Institute of Technology (Japan) in 2011. He is currently an associate professor in Osaka City University; preceding this position he was an assistant professor in the University of Tokyo 2013-2014, and a postdoctoral fellow in the University of Toronto 2011-2013. Dr. Cai's research interests include distributed control of multi-agent systems, distributed control of discrete-event systems, and control architecture of complex networked systems. In 2013 he received the Best Paper Award of SICE; in 2010 the Best Student Paper Award of IEEE Multi-Conference on Systems and Control, and Young Authors Award of SICE.

**Brian Anderson** was born in Sydney, Australia, and educated at Sydney University in mathematics and electrical engineering, with PhD in electrical engineering from Stanford University in 1966. He is a Distinguished Professor at the Australian National University and Distinguished Researcher in National ICT Australia (NICTA). His awards include the IFAC Quazza Medal in 1999, the IEEE Control Systems Award of 1997, the 2001 IEEE James H Mulligan, Jr Education Medal, and the Bode Prize of the IEEE Control System Society in 1992, and a number of other medals and best paper prizes. He is a Fellow of the Australian Academy of Science, the Australian Academy of Technological Sciences and Engineering, the Royal Society, Honorary Fellow of the Institution of Engineers, Australia, and a Foreign Associate of the US National Academy of Engineering. He holds honorary doctorates from a number of universities, including Université Catholique de Louvain, Belgium, and ETH, Zürich. He is a past president of the International Federation of Automatic Control and the Australian Academy of Science. He served as the first President of NICTA, and was a member of company boards, including Cochlear Ltd, the world's major supplier of bionic ears, and a member of the Prime Minister's Science Council under three prime ministers. His current research interests are in distributed control, sensor networks and econometric modelling.

**Changbin Yu** received the B.Eng(Hon 1) degree from Nanyang Technological University, Singapore in 2004 and the Ph.D. degree from the Australian National University, Australia, in 2008. Since then he has been a faculty member at the Australian National University and subsequently holding various positions with Hangzhou Dianzi University, National ICT Australia and Shandong Computer Science Center. He had won a competitive Australian Postdoctoral Fellowship (APD) in 2007 and a prestigious ARC Queen Elizabeth II Fellowship (QEII) in 2010. He was also a recipient of Australian Government Endeavour Asia Award (2005), Chinese Government Outstanding Overseas Students Award (2006), Asian Journal of Control Best Paper Award (20062009), etc. His current research interests include control of autonomous aerial vehicles, multi-agent systems and human-robot interactions. He is a Senior Member of IEEE and a member of IFAC Technical Committee on Networked Systems. He serves as a subject editor for International Journal of Robust and Nonlinear Control and an associate editor for System & Control Letters and IET Control Theory & Applications.

**Guoqiang Mao** (S'98-M'02-SM'08) received PhD in telecommunications engineering in 2002 from Edith Cowan University. He currently holds the position of Professor of Wireless Networking, Director of Center for Real-time Information Networks at the University of Technology, Sydney. He has published more than 100 papers in international conferences and journals, which have been cited more than 2500 times. His research interest includes intelligent transport systems, applied graph theory and its applications in networking, wireless multihop networks, wireless localization techniques and network performance analysis.