52nd IEEE Conference on Decision and Control
December 10-13, 2013. Florence, Italy

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

Kai Cai, Brian D.O. Anderson and Changbin Yu

***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 spatial frequency response and noise propagation associated to the algorithms. The tradeoffs of using local consensus, as compared to standard global consensus, include higher memory requirement and degraded noise performance.**

I. Introduction

Consensus of multi-agent systems comes in many varieties (e.g. [1]–[3]), and in this paper, we focus on a particular variety, namely average consensus (e.g. [4]–[6]). 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, transient performance, effect of time delay, agent loss, etc.

Finding an average also throws away much information. In many situations, one might well envisage that a local average might be useful, mitigating for instance the effect of measurement error. 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; generally, 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'.

We consider two schemes for computation of local average consensus. One involves the use of exponential weights to reflect 'closeness' of the agents 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 employs a finite window to reduce computation burden; the bounds of the finite window will be case-dependent in applications. In both schemes, we design local consensus algorithms 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 this paper we consider spatial variation in 1D for simplicity. The designed local consensus algorithms are distributed, as their global consensus counterparts, in that information exchange is allowed only among neighbors. As we will see, these algorithms have higher memory requirement than that of the global consensus (the latter can be made memoryless).

We also seek to understand the properties of the designed local consensus algorithms. In particular, we analyze spatial frequency response and noise propagation associated to the algorithms. To obtain a fully analytical result we limit our study to 1D sensor network [7], which can find its application in power line monitoring, canal/river monitoring, structural monitoring of railway and/or bridges.

The rest of the paper is organized as follows. Section II presents local average consensus algorithms for case where the measured variable has spatial variation but is constant in time. Section III and Section IV investigate spatial frequency response and noise propagation of the designed algorithms. Section V presents local consensus algorithms for the case where the measured variable has both spatial and temporal variations. Finally, Section VI states our conclusions.

II. 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 $k =$

K. Cai is with the University of Toronto, Toronto, Ontario, Canada. His work was supported by the Australian National University and NICTA Ltd, Canberra, ACT Australia. Email: kai.cai@scg.utoronto.ca. B.D.O. Anderson is with the Australian National University and NICTA Ltd, Canberra, A.C.T., Australia. His work was supported by the Australian Research Council through DP-130103610 and DP-110100538 and by National ICT Australia. Email: brian.anderson@anu.edu.au. C. Yu is with the Australian National University and NICTA Ltd, Canberra, A.C.T., Australia, and also with Shandong Computer Science Center, Jinan, China. The work of C. Yu was supported by the Australian Research Council through DP-130103610 and a Queen Elizabeth II Fellowship under DP- 110100538, and the Overseas Expert Program of Shandong Province. Email: brad.yu @anu.edu.au.

978-1-4673-5716-6/13/$31.00 ©2013 IEEE

$0, 1, 2, \ldots$ 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 (as we define below).

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 V, below. We consider two types of weighting schemes: exponential weighting and uniform finite window.

A. 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 ρ^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

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

Thus, exponentially decaying weights, at the rate ρ, 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 $(1-\rho)/(1+\rho)$ 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 all i,

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

Each sensor i needs information only from its two immediate neighbors: $y_{i-1}(k)$ and $y_{i+1}(k)$, $k = 0, 1, \ldots$. At each iteration k (≥ 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 memory is 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

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

This leads to

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

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

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

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 1: An extension of Algorithm (2) is immediate. Each sensor i weights information from the backward direction differently from the forward direction, using exponential weights ρ_b and $\rho_f \in (0, 1)$, respectively. Then revise Algorithm (2) as follows:

$$y_i(0) = \frac{(1-\rho_b)(1-\rho_f)}{1-\rho_b\rho_f} x_i \quad (5a)$$
$$y_i(1) = y_i(0) + \rho_b y_{i-1}(0) + \rho_f y_{i+1}(0) \quad (5b)$$
$$y_i(2) = y_i(1) + \rho_b(y_{i-1}(1) - y_{i-1}(0)) + \rho_f(y_{i+1}(1) - y_{i+1}(0)) - \rho_b\rho_f 2 y_i(0) \quad (5c)$$
$$y_i(k+1) = y_i(k) + \rho_b(y_{i-1}(k) - y_{i-1}(k-1)) + \rho_f(y_{i+1}(k) - y_{i+1}(k-1)) - \rho_b\rho_f(y_i(k-1) - y_i(k-2)),\ k \geq 2. \quad (5d)$$

This revised algorithm yields

$$\lim_{k\to\infty} y_i(k) = \frac{(1-\rho_b)(1-\rho_f)}{1-\rho_b\rho_f}\left(x_i + \sum_{j=1}^{\infty} (\rho_b^j x_{i-j} + \rho_f^j x_{i+j})\right). \quad (6)$$

The proof of this claim is almost the same as that validating Algorithm (2).

B. 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

$$y_i(L) = \frac{1}{2L+1}\left(x_i + \sum_{j=1}^{L}(x_{i-j} + x_{i+j})\right). \tag{7}$$

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.

$$y_i(0) = \frac{1}{2L+1}x_i \tag{8a}$$
$$y_i(1) = y_i(0) + (y_{i-1}(0) + y_{i+1}(0)) \tag{8b}$$
$$y_i(2) = y_i(1) + (y_{i-1}(1) - y_{i-1}(0)) + (y_{i+1}(1) - y_{i+1}(0)) - 2y_i(0) \tag{8c}$$
$$y_i(k+1) = y_i(k) + (y_{i-1}(k) - y_{i-1}(k-1)) + (y_{i+1}(k) - y_{i+1}(k-1)) - (y_i(k-1) - y_i(k-2)), \quad k \in [2, L-1]. \tag{8d}$$

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 V-B below), by contrast, the corresponding algorithm will need to keep track of temporal variations. Indeed, a significant variant on Algorithm (8) is needed, while the variation required for Algorithm 2 in comparison is minor.

Theorem 2: Algorithm (8) solves Problem 2.

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

$$y_i(k) = y_i(k-1) + (y_{i-k}(0) + y_{i+k}(0)), \quad \forall i. \tag{9}$$

This leads to

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

The second equality above is due to (8a). ■

Remark 2: 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.

$$|L_i - L_{i+1}| \leq 1, \quad |L_i - L_{i-1}| \leq 1, \quad \forall i \tag{10}$$

and replace L by L_i throughout Algorithm (8). Then from (8d) 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 (10) 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 (8) proves that the revised algorithm with L_i computes

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

III. 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 II: 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

$$\mathcal{X}(Z) = \sum_{-\infty}^{\infty} x_i Z^{-i} \qquad \mathcal{Y}(Z) = \sum_{-\infty}^{\infty} y_i Z^{-i} \tag{11}$$

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 ω, the steady state values of the consensus variables y_i depend on ρ and ω. Of course, 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.

A. Exponential Weighting

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

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

one has

$$Z^{-i} y_i = \frac{1-\rho}{1+\rho}[x_i Z^{-i} + Z^{-1}\rho x_{i-1} Z^{-(i-1)} + Z^{-2}\rho^2 x_{i-2} Z^{-(i-2)} + \cdots + Z\rho x_{i+1} Z^{-(i+1)} + Z^2\rho^2 x_{i+2} Z^{-(i+2)} + \cdots] \tag{13}$$

Summing from $i = -\infty$ to ∞ 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

$$\mathcal{Y}(Z) = \frac{(1-\rho)^2}{(1-\rho Z^{-1})(1-\rho Z)}\mathcal{X}(Z) \quad (14)$$

For future reference, define the transfer function

$$\mathcal{H}(Z) = \frac{(1-\rho)^2}{(1-\rho Z^{-1})(1-\rho Z)} \quad (15)$$

For $Z = \exp(j\omega)$, the transfer function 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 Z-transform $\mathcal{X}(Z)$ is formally given by $\sum_{i=-\infty}^{\infty} x_i Z^{-i}$. When $Z = \exp(j\omega)$, there holds

$$\mathcal{X}(\exp(j\omega)) = \sum_{i=-\infty}^{\infty} \exp(ji(\omega - \omega_0)) = 2\pi\delta(\omega - \omega_0) \quad (16)$$

where we are appealing to the fact that the delta function $\delta(x)$ is the limit of a multiple of the Dirichlet kernel

$$D_N(x) = \sum_{i=-N}^{N} \exp(jix) = \frac{\sin((N+\frac{1}{2})x)}{\sin(x/2)} \quad (17)$$

i.e.

$$\delta(x) = \frac{1}{2\pi}\lim_{N\to\infty} D_N(x) = \frac{1}{2\pi}\sum_{i=-\infty}^{\infty} \exp(jix) \quad (18)$$

In formal terms, it follows from (14) and (15) that the associated Z-transform of the consensus variable, i.e. $\mathcal{Y}(Z)$, is given by

$$\mathcal{Y}(\exp(j\omega)) = \mathcal{H}(\exp(j\omega))2\pi\delta(\omega - \omega_0) \quad (19)$$

Equivalently, the consensus variable is also sinusoidal at frequency ω_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 ω_0, and the amplitude is in fact the value of $\mathcal{H}$ itself, viz.

$$\mathcal{H}(\exp(j\omega_0)) = \frac{(1-\rho)^2}{1+\rho^2 - 2\rho\cos\omega_0} \quad (20)$$

Observe that if $\omega_0 = 0$, i.e. the measured variable is a constant or spatially invariant, then $\mathcal{H}(1) = 1$ irrespective of ρ, 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 ρ 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, ρ (for values close to 1) determines the bandwidth as $O(1-\rho)$. More generally, we observe from the Figures 1 and 2 (which show behaviour near the origin and over $[0, \pi]$), that

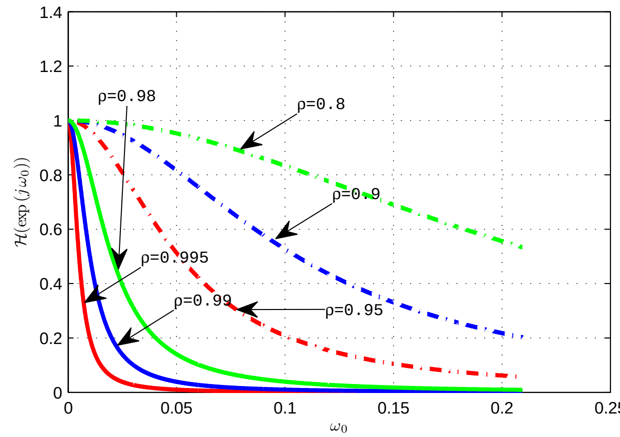

Fig. 1. Plot of $\mathcal{H}(\exp(j\omega_0))$ in (20) near origin for different values of ρ

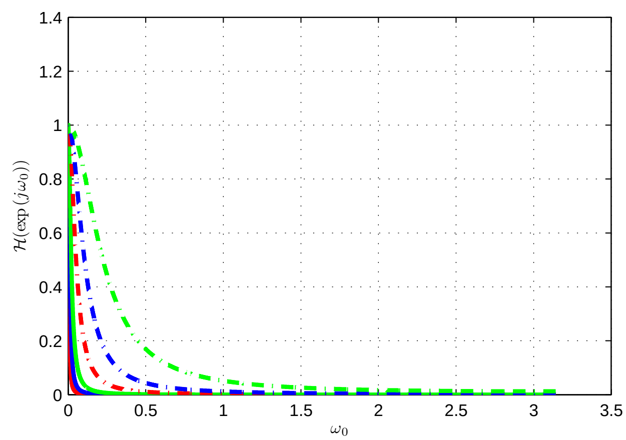

Fig. 2. Plot of $\mathcal{H}(\exp(j\omega_0))$ in (20) over $[0, \pi]$ for different values of ρ. The colour coding is as for Figure 1.

1) For any ρ, $\mathcal{H}(\exp(j\omega_0))$ is monotonic decreasing in ω_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 $\omega_0 = 1-\rho$.

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 ρ^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.

B. Uniform Finite Window

From (7), the steady-state equation in this case is

$$y_i = \frac{1}{2L+1} \sum_{k=-L}^{L} x_{i+k} \tag{21}$$

and it is straightforward to establish that

$$\mathcal{Y}(Z) = \frac{1}{2L+1} \sum_{k=-L}^{L} Z^k \mathcal{X}(Z) \tag{22}$$

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

$$\mathcal{H}(\exp(j\omega)) = \frac{1}{2L+1} \frac{\sin((L+\frac{1}{2})\omega)}{\sin(\omega/2)} \tag{23}$$

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 $\frac{1.7}{L+1/2}$. Evidently, the bandwidths in the exponential weighted case and the uniform finite window case are of the same order when

$$1 - \rho = \frac{1.7}{L+1/2}. \tag{24}$$

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

IV. 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 ϵ_i of zero mean and variance σ^2, with the noise at any two agents being independent.

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 $2L_i+1$ is used, this same thinking shows that the error variance is $\frac{\sigma^2}{2L_i+1}$.

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

$$\frac{1-\rho}{1+\rho}[\epsilon_i + \rho\epsilon_{i-1} + \rho^2\epsilon_{i-2} + \cdots + \rho\epsilon_{i+1} + \rho^2\epsilon_{i+2} + \ldots] \tag{25}$$

and the variance is given by

$$\begin{aligned} &\left(\tfrac{1-\rho}{1+\rho}\right)^2[1 + 2\rho^2 + 2\rho^4 + \ldots]\sigma^2 \\ &= \left(\tfrac{1-\rho}{1+\rho}\right)^2[\tfrac{2}{1-\rho^2} - 1]\sigma^2 \\ &= (1-\rho)\tfrac{1+\rho^2}{(1+\rho)^3}\sigma^2 \end{aligned} \tag{26}$$

This lies in the interval $(\frac{1}{4}(1-\rho), 1-\rho))$, and for ρ close to 1, the error is approximately equal to the lower bound. Indeed, the closer ρ is to 1, the less is the error variance. It is not hard to verify that a uniform finite window of length $2L_i+1$ and an exponential weighting of $\rho = \frac{2L_i-3}{2L_i+1}$ yield the same variance. Equivalently, this condition is $1-\rho = \frac{2}{L_i+1/2}$, which means that exponential weighting and uniform finite window weighting, if they achieve the same bandwidth (cf. (24)), also have approximately the same noise performance. The same condition incidentally says that $\rho^{L_i} \approx e^{-1}$, 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_i or ρ are adjusted, noise variance is proportional to bandwidth.

V. 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/voltage 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.

In the sequel, we will again consider the two schemes: first exponential weighting, and then uniform finite window. The proofs of the results can be found in [8].

A. Exponential Weighting

Henceforth, we shall assume as is reasonable 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

$$y_i(k) = \frac{1-\rho}{1+\rho}\Big(x_i(k) + \sum_{j=1}^{k}\rho^j(x_{i-j}(k-j) + x_{i+j}(k-j))\Big). \tag{27}$$

An exponential weight ρ^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.

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.

$$y_i(0) = \lambda x_i(0), \quad \lambda := \frac{1-\rho}{1+\rho} \tag{28a}$$

$$y_i(1) = y_i(0) + \rho(y_{i-1}(0) + y_{i+1}(0)) + \lambda(x_i(1) - x_i(0)) \tag{28b}$$

$$\begin{aligned} y_i(2) = y_i(1) + \rho(y_{i-1}(1) - y_{i-1}(0)) + \\ \rho(y_{i+1}(1) - y_{i+1}(0)) - \rho^2 2y_i(0) + \lambda(x_i(2) - x_i(1)) \end{aligned} \tag{28c}$$

$$\begin{aligned} &y_i(k+1) = y_i(k) + \rho(y_{i-1}(k) - y_{i-1}(k-1)) + \\ &\rho(y_{i+1}(k) - y_{i+1}(k-1)) - \rho^2(y_i(k-1) - y_i(k-2)) + \\ &\lambda(x_i(k+1) - x_i(k)) - \rho^2\lambda(x_i(k-1) - x_i(k-2)),\ k \geq 2. \end{aligned} \tag{28d}$$

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. Indeed, $x_i(k+1) - x_i(k)$

provides new information, while $x_i(k-1) - x_i(k-2)$ is used as a correction term.

Theorem 3: Algorithm (28) solves Problem 3.

As commented in Remark 1 for Algorithm (2), we may similarly extend Algorithm (28) to the case where sensors assign different exponential weights to information from the backward and the forward directions, using $\rho_b, \rho_f \in (0,1)$.

B. 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 formally.

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

$$\begin{aligned} y_i(k) &= \frac{1}{2L+1}\left(x_i(k) + \sum_{j=1}^{k}(x_{i-j}(k-j) + x_{i+j}(k-j))\right) \quad \text{if } k \leq L; \\ y_i(k) &= \frac{1}{2L+1}\left(x_i(k) + \sum_{j=1}^{L}(x_{i-j}(k-j) + x_{i+j}(k-j))\right) \quad \text{if } k > L. \end{aligned} \tag{29}$$

The explanation for the time arguments associated with x_{i-j} and x_{i+j} on the right of (29) 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. $x_{i+j}(k-j)$).

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 (8). Finally, the jth 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 (8)). For every $j \in [0, L]$, if $k < j$,

$$z_{ij}(k) = 0; \tag{30}$$

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

$$z_{ij}(k) = \frac{1}{2L+1} x_i(k) \tag{31a}$$

$$z_{ij}(k+1) = z_{ij}(k) + (z_{(i-1)j}(k) + z_{(i+1)j}(k)) \tag{31b}$$

$$\begin{aligned} z_{ij}(k+2) = z_{ij}(k+1) &+ (z_{(i-1)j}(k+1) - z_{(i-1)j}(k)) \\ &+ (z_{(i+1)j}(k+1) - z_{(i+1)j}(k)) - 2z_{ij}(k) \end{aligned} \tag{31c}$$

$$\begin{aligned} z_{ij}(k+3) = z_{ij}(k+2) &+ (z_{(i-1)j}(k+2) - z_{(i-1)j}(k+1)) \\ &+ (z_{(i+1)j}(k+2) - z_{(i+1)j}(k+1)) - (z_{ij}(k+1) - z_{ij}(k)) \end{aligned} \tag{31d}$$

$$\vdots$$

$$\begin{aligned} z_{ij}(k+L) = z_{ij}(k+L-1) &+ \\ (z_{(i-1)j}(k+L-1) &- z_{(i-1)j}(k+L-2)) + \\ (z_{(i+1)j}(k+L-1) &- z_{(i+1)j}(k+L-2)) - \\ (z_{ij}(k+L-2) &- z_{ij}(k+L-3)) \end{aligned} \tag{31e}$$

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.

$$y_i(k) = z_{ij}(k) + \sum_{l=0, l\neq j}^{L}(z_{il}(k) - z_{il}(k-1)), \quad j = k \pmod{L+1}. \tag{32}$$

Theorem 4: Algorithm (30)-(32) solves Problem 4.

We refer to [8] for frequency response analyses of the two local consensus algorithms designed in this section, with respect to both spatial and temporal variations.

VI. 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. Two schemes for local average computation have been employed: exponential weighting and uniform finite window. Further, we have analyzed spatial frequency response and noise propagation associated to the algorithms. In future work, we aim to extend the algorithms to higher dimensions.

References

[1] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents sing nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[2] W. Ren, R. Beard, and E. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proc. of American Control Conf.*, Baltimore, MD, 2005, pp. 1859–1864.

[3] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[4] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *J. of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33–46, 2007.

[5] K. Cai and H. Ishii, "Average consensus on general strongly connected digraphs," *Automatica*, vol. 48, no. 11, pp. 2750–2761, 2012.

[6] K. Topley and V. Krishnamurthy, "Average-consensus in a deterministic framework–part I: strong connectivity," *IEEE Trans. Sig. Processing*, vol. 60, no. 12, pp. 6590–6603, 2012.

[7] B. Huang, C. Yu, and B. Anderson, "Analyzing localization errors in one-dimensional sensor networks," *Sig. Processing*, vol. 92, no. 2, p. 427438, 2012.

[8] K. Cai, B. Anderson, C. Yu, and G. Mao, "Local average consensus in distributed measurement of spatial-temporal varying parameters: 1d case," technical report, 2013. Online at http://arxiv.org/abs/1308.6641.