A biologically inspired approach to real-valued average consensus over quantized channels with arbitrary deterministic accuracy

Andrea Censi, Richard M. Murray

Abstract—This paper concerns the average consensus problem with the constraint of quantized communication between nodes. A broad class of algorithms is analyzed, in which the transmission strategy, which decides what value to communicate to the neighbours, can include various kinds of rounding, probabilistic quantization, and bounded noise. The arbitrariness of the transmission strategy is compensated by a feedback mechanism using a self-inhibitory action; this is inspired by various neuronal models. Using these algorithms, the average of the nodes state is not conserved across iterations, and the nodes do not converge to an equilibrium; however, we show that both kinds of errors can be made as small as desired. Bounds on these quantities involve the spectral properties of the graph and can be proved by employing elementary techniques of LTI systems analysis.

I. INTRODUCTION

Consider the following variation of the average consensus problem: each node in a graph knows a number \( x_i(0) \in \mathbb{R} \), and the goal is to drive each node’s belief to the initial average \( \alpha \), with the limitation that nodes can communicate only with their neighbours on the graph, and that the channel is quantized.

At first sight, this problem looks like a false problem, for if one can send even only one bit over a channel, then one can send anything, by creating an adequate coding. For example, if a resolution of \( 2^{-8} \) is needed, then one could consider 8 consecutive bits as a code word, where the \( i \)-th bit would represent the \( i \)-th bit in the binary expansion of the number being submitted. With such coding, one can apply the standard consensus algorithms which will achieve a precision of \( 2^{-8} \). If more precision is required, one can use a 16 bit word, and so on.

This is true, but the problem can be framed in another way: given a certain quantization, how precise can the consensus be? Previous work always assumed that the achievable precision would have been in the order of the quantization step; instead, we show that consensus can be reached with a precision which is arbitrarily small, at the expense of slower convergence, and without “cheating” by messing with the channel coding. For example it is possible to attain a precision of \( 2^{-10} \) even using 8 bit words.

Also note that there are situations in which the channel carries only one bit, and there is no complexity available to do network coding. Consider the cartoonish representation of a biological neural network in Fig. 1. Assume some of the neurons are sensible to the same external stimulus (temperature, light intensity, sweetness, etc.) and that we wish to obtain an average of these measures (which we think as real-valued excitation levels). This can be cast as an average consensus problem with \( 0 - 1 \) communication links, where \{0, 1\} can be mapped to \{no spike, spike\}. How would you design a biologically-plausible algorithm such that neurons can synchronize their internal state, using only spikes?

It is well known that neurons communicate utilizing trains of spikes. Spikes are a all-or-none phenomenon: either the neuron fires, or it does not. Increased neuronal activity results in more frequent spikes. Even without delving into the biologically complexity of the neuron, it is known that there is a biological mechanism that converts an internal “state” to a spikes representation by modulating the frequency. There exist arbitrarily complicated models for the firing behavior of the neuron; these range from the first order linear threshold-and-fire model to four-dimensional nonlinear models [9], [7]. These models all have some features in common: the spiking is a highly nonlinear, and partly non-predictable, phenomenon; a spike has a self-inhibitory effect on the neuron, end either a excitatory or inhibitory effect on other connected neurons.

The transmission strategy we propose is based on an arbitrary “spiking function” \( \psi \) (any function or random variable such that \(|\psi(z) - z|\) is bounded). A spike has a self-inhibitory effect on the node’s state; and an excitatory effect on the connected nodes. These two effects are linear, and together mitigate the arbitrariness of the nonlinear/stochastic \( \psi(z) \). Albeit we provide a dry mathematical representation using the quantization operator, it is simple enough it should be reproducible on wetware. For the neuron network in Fig. 1, our algorithm is such that eventually each axon fires with a frequency proportional to the average of the external stimuli.

A. Censi and R. M. Murray are with the Control & Dynamical Systems department, California Institute of Technology, MC 107-81, 1200 E. California Blvd., 91125, Pasadena, CA. {andrea,murray}@cds.caltech.edu
II. RELATED WORK

A consensus algorithm over quantized channels is a two-part strategy:

1) Communication strategy: how to decide the value $y_j(k)$ to send to the neighbors.

2) Update strategy: how to update one node’s value $x_i(k)$ based on the $y_j(k)$ received from its neighbors.

We will be consistent through the paper to use the index $j$ to refer to the communication strategy, and index $i$ to refer to the update strategy.

Many works dealt with reaching a quantized consensus, in the sense that $x_j(k) \in \mathbb{Z}$, and in the limit the states differ at most by 1; for the analysis of this problem see [6] and references therein.

In [2], [3], [4], [5] the authors consider the problem of reaching a consensus in $\mathbb{R}$ using quantized channels. They propose the following communication strategy:

$$y_j(k) = q(x_j(k))$$

where $q(x)$ rounds $x$ to the nearest integer, and the following updating strategy:

$$x_i(k + 1) = x_i(k) - y_i(k) + \sum_j P_{i,j} y_j(k)$$

where $P$ is any doubly stochastic matrix with positive diagonal and with the corresponding graph strongly connected.

This algorithm is such that:

- The average is not conserved.
- Nodes converge to a consensus $\tau \in \mathbb{Z}$.
- The expected value of $\tau$ is $\alpha$.

III. PROPOSED STRATEGY

As for the updating strategy, we use the same update rule discussed as the base case in the tutorial paper [8]:

$$x_i(k + 1) = x_i(k) + \epsilon \sum_j a_{ij} (y_j(k) - x_i(k))$$

with $\epsilon = \frac{\eta}{\Delta}$, $\eta \in (0, 1)$ (3)

The difference is that we use $y_j(k)$, the value transmitted by node $j$, instead of using the real state $x_j(k)$.

Our communication strategy relies on the definition of an auxiliary state variable $c_j(k) \in \mathbb{R}$, for which initially $c_j(0) = 0$, and a certain function $\psi : \mathbb{R} \rightarrow \mathbb{Z}$, or random variable as well, such that

$$|\psi(y) - y| \leq \beta$$

for some $\beta$. Then our proposed communication strategy is:

$$\begin{align*}
y_j(k) &= \psi(x_j(k) - c_j(k)) \\
c_j(k + 1) &= c_j(k) + (y_j(k) - x_j(k))
\end{align*}$$

(5)
Note that the auxiliary variable \(c_j(k)\) integrates the error in approximating \(x_j\) with \(y_j\). This error is then used as a negative feedback for \(\psi\). Each time \(\psi\) produces a “pulse”, this has a self-inhibitory effect on the generation of the next pulse, which imitates the effect of a inhibitory post-synaptic potential in the neuron.

Examples of this kind of functions include:

- **Deterministic rounding function**: Define the function \(q : \mathbb{R} \rightarrow \mathbb{Z}\) such that \(q(x)\) is the integer closest to \(x\). Then one can choose
  \[
  \psi(x) = q(x), \quad \beta = 0.5
  \]
  For this particular choice of \(\psi\), the system (5) is an “encoder” for \(x\), in the sense that if \(x\) is fixed the time average of \(y\) is \(x\). Unfortunately, the proofs for the main results in the next section are somehow technical and do not give an intuition of what is really going on here. For this reason, we added Section V, which is intended to be a light read to give the right intuition about the properties of the subsystem (5).

- **Ceiling/floor functions**:
  \[
  \psi(x) = \lfloor x \rfloor, \quad \beta = 1
  \]
  \[
  \psi(x) = \lceil x \rceil, \quad \beta = 1
  \]

- **Threshold-and-fire**: Fire to the next integer over an arbitrary threshold \(\overline{\alpha}\).
  \[
  \psi(x) = \psi(m + p) = \begin{cases} m + 1 & \text{if } p > \overline{\alpha} \\ m & \text{otherwise} \end{cases}
  \]
  \[
  \beta = \max\{\overline{\alpha}, 1 - \overline{\alpha}\}
  \]

- **Probabilistic rounding functions**: One can use the probabilistic quantization defined in (1), setting
  \[
  q = q^p, \quad \beta = 1
  \]

- **Completely random rounding function**: Choose randomly between the previous and the next integer:
  \[
  \psi(x) = \begin{cases} \lfloor x \rfloor & \text{with probability } p \\ \lceil x \rceil & \text{with probability } 1 - p \end{cases}
  \]
  \[
  \beta = 1
  \]

In the case without quantization \((y_j(k) = x_j(k))\), it is possible to obtain, with very mild assumptions, two nice properties of the update strategy (2): the mean is conserved across iterations, and the disagreement tends to zero. This is not true in our case: the mean is not conserved, and the states do not converge to a consensus. However, we can provide bounds on the errors which depend linearly on \(\epsilon\) and so can be made as small as desired, and, in particular, much smaller than the quantization.

\footnote{The fact that \(q(0.5)\) be 0 or 1 is immaterial but it is important to decide; in this paper, we see the glass half full and round up to the next integer: \(q(0.5) = 1\) and \(q(-0.5) = 0\). Note that the common software implementations (C standard library, Matlab) round away from zero: \texttt{round}(0.5) = 1 and \texttt{round}(-0.5) = -1.}

### Table II

**SYMBOLS USED IN THIS PAPER**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n)</td>
<td>number of nodes</td>
</tr>
<tr>
<td>(A)</td>
<td>adjacency matrix</td>
</tr>
<tr>
<td>(\Delta)</td>
<td>graph degree</td>
</tr>
<tr>
<td>(P)</td>
<td>Perron matrix; (P = I - \epsilon L) with (\epsilon &lt; 1/\Delta)</td>
</tr>
<tr>
<td>(L)</td>
<td>Laplacian matrix; (L = D - A)</td>
</tr>
<tr>
<td>(q_j(k))</td>
<td>node state</td>
</tr>
<tr>
<td>(\alpha)</td>
<td>target value: (\alpha = \sum x_i(0)/n)</td>
</tr>
<tr>
<td>(y_j(k))</td>
<td>value transmitted by node (j) to neighbors at time (k)</td>
</tr>
<tr>
<td>(d(k))</td>
<td>drift</td>
</tr>
<tr>
<td>(\varphi(k))</td>
<td>average disagreement</td>
</tr>
<tr>
<td>(q(x))</td>
<td>nearest integer to (x)</td>
</tr>
<tr>
<td>(q^p(x))</td>
<td>Probabilistic quantization for (x)</td>
</tr>
<tr>
<td>(\psi(x))</td>
<td>auxiliary state variable</td>
</tr>
<tr>
<td>(I)</td>
<td>(n \times 1) vector of 1s</td>
</tr>
</tbody>
</table>

To measure the performance of the algorithm, we define two error measures. The first is the drift from the mean:

\[
\frac{d(k)}{\left|\frac{1}{n} \sum \alpha_i(k) - \alpha\right|}
\]

The other is the disagreement among the nodes:

\[
\sum_{i,j} a_{ij} (x_i(k) - x_j(k))^2 = x(k)^T L x(k)
\]

where \(L\) is the Laplacian of the graph \((L = D - A, \text{ where } D\) is the degree matrix and \(A\) is the adjacency matrix). Because we are interested in the performance as the number of nodes grows, we look at the average disagreement \(\varphi(k)\), which we define as

\[
\varphi(k) = \left[ \frac{1}{n\Delta} \sum_{i,j} a_{ij} (x_i(k) - x_j(k))^2 \right]^{1/2}
\]

Notice we use \(n\Delta\) as an approximation of the number of edges; the square root is to obtain a linear measure to comparable with \(d(k)\).

For the proposed strategy, we can prove the following bounds, that are valid for any balanced graph:

\[
\frac{\lim_{k \to \infty} \varphi(k)}{\sqrt{6} \cdot \eta \beta \lambda_n(L)} \leq \frac{\lambda_2(L)}{\lambda_2(L)}
\]

In this paper, with abuse of notation, we write \(\lim_{k \to \infty} \varphi(k) \leq c\) in the sense that \(\varphi(k) \leq c + f(k)\) with \(\lim_{k \to \infty} f(k) = 0\). These bounds show that the errors are linear in \(\epsilon\).

Note in (7) the three factors that impact the accuracy: \(\eta\) is in arbitrary number in \((0, 1)\) that appears in the update strategy, \(\beta\) depends on the quantization error and quantization strategy, and of course \(\lambda_n(L) / \lambda_2(L)\) depends on the graph.

By choosing a \(\eta\), we can make these two errors as small as desired; the trade-off is that a small \(\eta\) leads to small convergence.
IV. MAIN RESULTS

We briefly recalls the main properties of graphs that we use in the following:

- For undirected graphs, \( L \) is a symmetric positive semidefinite matrix. The eigenvalues are bounded by the degree of the graph: \( \lambda_1 \{ L \} \leq 2\Delta \). The smallest eigenvalue \( \lambda_1 \{ L \} \) is 0. The largest eigenvalue \( \lambda_n \{ L \} \) is at least \( \Delta \).
- The second smallest \( \lambda_2 \{ L \} \) is different than zero if the graph is connected.
- If the graph is balanced, then 1 is in the kernel of \( L \): \( 1^T L = 0 \).
- If \( \epsilon < 1/\Delta \), then \( P = I - \epsilon L \) is a doubly stochastic matrix. 1 is an eigenvector for the simple eigenvalue 1 of \( P \): \( 1^T P = 1 \). The other eigenvalues of \( P \) are strictly less than 1.
- \( P \) and \( L \) have the same eigenvectors. If \( \lambda_j \) is the \( j \)-th eigenvalue of \( L \), then the \( j \)-th eigenvalue for \( P \) is \( \mu_j = 1 - \epsilon \lambda_j \).
- \( \lim_{k \to \infty} P^k L = 0 \)

Firstly we prove a simple lemma about the only invariant in the system:

**Lemma 1:** Let \( d_j \) be the degree of node \( j \). Then the following quantity \( V(k) \) is invariant:

\[
V(k) = \sum_i (x_i(k) - \epsilon d_j c_j(k)) = 1^T (x(k) - \epsilon Dc(k))
\]

**Proof:** Notice that \( y(k) \) can be written as:

\[
y(k) = x(k) + [c(k+1) - c(k)]
\]

Hence the dynamics can be rewritten as:

\[
x(k+1) = P x(k) + \epsilon A(c(k+1) - c(k))
\]

Now a straight computation gives:

\[
V(k+1) = 1^T (x(k+1) - \epsilon Dc(k+1)) = 1^T (P x(k) + \epsilon A(c(k+1) - c(k)) - Dc(k+1)) = 1^T (P x(k) - Dc(k) + cL(c(k+1))) = 1^T (x(k) - Dc(k)) = V(k)
\]

**Lemma 2:** \( |c_j(k)| \leq \beta \)

**Proof:**

\[
|c_j(k+1)| = |c_j(k) + (y_j(k) - x_j(k))| = |\psi(x_j(k) - c_j(k)) - (x_j(k) - c_j(k))| = |\psi(z) - z| \leq \beta
\]

Given the previous lemma, the bound on the drift is an easy consequence:

**Proposition 1:**

\[
d(k) \leq \eta \beta
\]

**Proof:** Notice that

\[
\alpha = \frac{1}{n} V(0) = \frac{1}{n} V(k) = \frac{1}{n} 1^T (x(k) - \epsilon Dc(k))
\]

hence

\[
\left| \frac{1}{n} 1^T x(k) - \alpha \right| = \left| \frac{1}{n} 1^T Dc(k) \right| \leq \frac{1}{n} Tr(D) \beta \leq \frac{1}{n} (n\Delta) \beta = \epsilon \Delta \beta = \eta \beta
\]

**Proposition 2:** Eventually, the disagreement is bounded by \( \epsilon \):

\[
\lim_{k \to \infty} \varphi(k) \leq \sqrt{6} \cdot \eta \beta \frac{\lambda_n \{ L \}}{\lambda_2 \{ L \}}
\]

**Proof:** Recall that the dynamics can be written as:

\[
x(k+1) = P x(k) + \epsilon A(c(k+1) - c(k))
\]

Note that \( \epsilon A(c(k)) \) is added to both \( x(k) \) (with a plus sign) and \( x(k+1) \) (with a minus sign). The contribution to \( x(k+1) \) is globally \( +P \epsilon A x(k) - \epsilon A(c(k)) = \epsilon (P-I) A c(k) = -\epsilon^2 L A c(k) \). Consequently, the state has the following closed form expression:

\[
x(k) = P^k x(0) + \epsilon A c(k) - \epsilon^2 \sum_{\tau=1}^{k-1} P^{k-\tau} L A c(\tau) - \epsilon P^k A c(0)
\]

We want to compute the limit of the disagreement function \( x(k)^T L x(k) \) as \( k \to \infty \). Note that it is composed by 6 terms:

\[
x(k)^T L x(k) =
\]

\[
x(0)^T P^k L P^k x(0) +
\]

\[
+ \epsilon^2 c(k)^T A L A c(k) +
\]

\[
+ \epsilon^4 \sum_{m=1}^{k-1} \sum_{\tau=1}^{k-1} c(\tau)^T A L P^{k-\tau} L P^{k-\tau} A c(\tau) +
\]

\[
+ \epsilon^4 (0)^T P^k L A c(k) +
\]

\[
- \epsilon^2 x(0)^T P^k L \sum_{\tau=1}^{k-1} P^{k-\tau} L A c(\tau) +
\]

\[
- \epsilon^2 c(k)^T A \sum_{\tau=1}^{k-1} L P^{k-\tau} L A c(\tau)
\]

Because \( P^k L \to 0 \), the terms (10), (13), (14) disappear. The term (11) can be bounded as follows:

\[
|\epsilon^2 c(k)^T A L A c(k)| \leq
\]

\[
\leq \beta^2 \epsilon^2 \max_{||u||_\infty \leq 1} |u^T A L A u|
\]

\[
\leq \beta^2 \epsilon^2 \Delta^2 \max_{||u||_\infty \leq 1} |u^T L u|
\]

\[
\leq \beta^2 \epsilon^2 \Delta^2 n \lambda_n \{ L \}
\]
The term (15) can be bounded as
\[
|c^3 e(k)^T A \sum_{\tau=1}^{k-1} L P^{k-\tau} L A c(\tau)| \leq \\
\beta^2 \epsilon^2 \Delta^2 n \sum_{\tau=1}^{k-1} \lambda_{\max} \left\{ L P^{k-\tau} L \right\}
\]
Recall that \(P\) and \(L\) have the same eigenvectors; hence the typical eigenvalue for the matrix \(L P^{k-\tau} L\) has the value
\[
\lambda_i \left\{ L P^{k-\tau} L \right\} = \lambda_i \{L\}^2 (1 - \epsilon \lambda_i \{L\})^{k-\tau}
\]
This expression can be bounded by choosing the largest eigenvalue \(\lambda_n\) for the first factor, and the smallest non-zero eigenvalue \(\lambda_2\) for the second factor.
\[
\lambda_{\max} \left\{ L P^{k-\tau} L \right\} \leq \lambda_n \{L\}^2 (1 - \epsilon \lambda_2 \{L\})^{k-\tau}
\]
The sum of the series can be computed as:
\[
\sum_{\tau=1}^{k-1} (1-q)^{k-\tau} = \frac{(1-q)}{q} \left[ 1 - (1-q)^{k-1} \right]
\]
Hence for the sixth term:
\[
|c^3 e(k)^T A \sum_{\tau=1}^{k-1} L P^{k-\tau} L A c(\tau)| \leq \\
\beta^2 \epsilon^2 \Delta^2 n \sum_{\tau=1}^{k-1} \lambda_{\max} \left\{ L P^{k-\tau} L \right\}
\]
We find again that
\[
\lambda_{\max} \left\{ L P^{k-\tau} L \right\} \leq \\
\lambda_n^3 \{L\} (1 - \epsilon \lambda_2 \{L\})^{(k-m)+(k-\tau)}
\]
and for the series:
\[
\sum_{\tau=1}^{k-1-k-1} (1 - \epsilon \lambda_2 \{L\})^{(k-m)+(k-\tau)} = \\
\frac{(1-\epsilon \lambda_2 \{L\})^2}{\epsilon^2 \lambda_2 \{L\}^2} \left[ 1 - (1 - \epsilon \lambda_2 \{L\})^{k-1} \right]^2
\]
And hence the bound for the sixth term is
\[
|c^3 \sum_{m=1}^{k-1} \sum_{\tau=1}^{k-1} e(\tau)^T A L P^{k-m} L P^{k-\tau} L A c(\tau)| \leq \\
\beta^2 \epsilon^2 \Delta^2 n \sum_{\tau=1}^{k-1-k-1} \lambda_{\max} \left\{ L P^{k-\tau} L \right\}
\]
Finally, the limit for the disagreement function is
\[
\lim_{k \to \infty} e(k)^T L x(k) \leq \beta^2 \epsilon^2 n \Delta \lambda_n \{L\} \left( 1 + \frac{\lambda_n \{L\}}{\lambda_2 \{L\} \lambda_2 \{L\}} \right)
\]
Note that \(\lambda_n / \lambda_2 \geq 1\); a good approximation is
\[
\left( 1 + \frac{\lambda_n \{L\}}{\lambda_2 \{L\} \lambda_2 \{L\}} \right) \leq \frac{\lambda_2 \{L\}^2}{\lambda_2 \{L\} \lambda_2 \{L\}}
\]
At this point, use the fact that \(\lambda_n \{L\} \leq 2\Delta:
\[
\lim_{k \to \infty} e(k)^T L x(k) \leq 6 \beta^2 \epsilon^2 n \Delta \lambda_n \{L\} \lambda_2 \{L\}
\]
Hence for the average disagreement:
\[
\lim_{k \to \infty} \frac{1}{n} e(k)^T L x(k) \leq \sqrt{6} \cdot \epsilon \beta \lambda_n \{L\}
\]

V. WHY ALL OF THIS ACTUALLY WORKS

Hopefully last section’s tedious proof convinced your mind that our algorithm works. In this section, we shall try to convince also your heart by (re)stating some quite trivial properties of the transmission part of the strategy, in the case one chooses \(\psi\) as the rounding function \(q\).

Recall \(q(x)\) is the integer closest to \(x\), and define \(r(x) = q(x) - x\). Then rewrite (5) by considering a fixed state \(x(k) = v\):
\[
\begin{cases}
q(k) = q(v - c(k)) \\
c(k+1) = r(v - c(k))
\end{cases}
\]
(16)
With these choices, this system is an “encoder” for \(v\), in the sense that it produces a train of spikes whose average in the long run is \(v\). More precisely:

**Proposition 3:** Consider the dynamical system (16), with the initial state \(c(0) = \gamma \in (-0.5, 0.5]\) and the encoded value \(v \in \mathbb{R}\) considered fixed. Let \(S(k)\) be the sum of the first \(k\) elements in the series:
\[
S(k) = \sum_{i=0}^{k-1} y(i)
\]
Then:
1) The time average of \(y\) converges to \(v\):
\[
\lim_{k \to \infty} \frac{1}{k} S(k) = v
\]
(18)
2) The convergence rate is linear. If starting from \(\gamma = 0\), then the following bound holds:
\[
\left| \frac{1}{k} S(k) - v \right| \leq 0.5 \frac{1}{k}
\]
(19)
3) For arbitrary \(\gamma \in (-0.5, 0.5]\):
\[
\left| \frac{1}{k} S(k) - v \right| \leq \frac{1}{k}
\]
(20)
(this is valid when \(S(k)\) is the sum of any \(k\) consecutive numbers in the sequence).
4) If \( v = p/q \in \mathbb{Q} \), then \( c(k) \) is periodic with period \( q \). Otherwise it is aperiodic.

These bounds are both reached for \( k = 0 \). Note also that 0.5/\( k \) is the minimum error you can have if you were to choose the \( k \) numbers \( y(0), \ldots, y(k-1) \) according to any other algorithm in this sense the encoding strategy is optimal.

Proof: The proof is nothing fancy. Some properties that we need are that for \( i \in \mathbb{Z} \):

\[
q(x + i) = q(x) + i \\
r(x + i) = r(x)
\]

\(|q(x - \gamma) - x| \leq 1 \) if \( \gamma \in (-0.5, 0.5] \)

Firstly, we shall show by induction that

\[ c(k) = r(kv - \gamma) \quad (21) \]

For \( k = 0 \), it is verified: \( c(0) = r(0 - \gamma) = \gamma \). Assume it is valid for \( c(k) \). Then for \( c(k+1) \):

\[
c(k+1) = r(v - c(k)) = r(v - r(kv - \gamma)) = r(v - (q(kv - \gamma) - (kv - \gamma))) = r((k+1)v - \gamma + q(kv - \gamma)) = r((k+1)v - \gamma)
\]

Given (21), we find this expression for \( y(k) \):

\[
y(k) = q(v - c(k)) = q(v - q(kv - \gamma) + (kv - \gamma)) = q(v + (kv - \gamma)) - q(kv - \gamma) = q((k+1)v - \gamma) - q(kv - \gamma)
\]

Now note that the terms of \( y(k) \) cancel each other when summed together:

\[
S(k) = q(kv - \gamma) - q(-\gamma) = q(kv - \gamma)
\]

Hence finally we can prove (18)

\[
\lim_{k \to \infty} \frac{1}{k} S(k) = \lim_{k \to \infty} \frac{1}{k} q(kv - \alpha) = v
\]

For proving (19) and (20), note that

\[
\left| \frac{1}{k} S(k) - v \right| = \frac{|q(kv - \gamma) - kv|}{k}
\]

For \( \gamma = 0 \), note that \( |q(kv - \gamma) - kv| = |q(kv) - v| \leq 0.5 \), and for \( \gamma \in (0.5, 0.5] \), note that \( |q(kv - \gamma) - kv| \leq 1 \).

To prove the last point, note that \( c(k) = c(0) \) implies

\[
q(kv - \gamma) - (kv - \gamma) = \gamma \\
\therefore v = \frac{q(kv - \gamma)}{k} \in \mathbb{Q}
\]

Conversely, assume \( v = p/q \in \mathbb{Q} \). Then

\[
c(q) = q(p - \gamma) - (p - \gamma) = q(p) - p + \gamma = \gamma = c(0)
\]

In Fig. 3 you can see an example of encoding \( \pi \) using 100 samples, using \( \gamma = 0 \) and \( \gamma = -0.49 \).

VI. SIMULATIONS

The Matlab source code for the simulations is available at:

http://pure.org/censi/2008/consensus

We consider here \( \psi = q \) because it has the lowest \( \beta = 0.5 \). Fig. 2 shows an example run of our algorithm on a circular graph with \( n = 10 \) nodes, for \( \eta = 0.01 \) and \( \eta = 0.05 \). Note that the states eventually converge to a periodic function. The results are qualitatively and quantitatively similar for other choices of \( \psi \); in general, deterministic \( \psi \)'s make the system converge to a periodic function, while of course that is not true for probabilistic quantizations.

We compare our algorithm to the one proposed by Carli et al. on a set of canonical graphs. The results of the simulations are shown in Table V, along with the relevant properties (eigenvalues) for the graphs. The proposed method does indeed deliver on its claims of accuracy. However, the bounds we found are very loose.

VII. FUTURE WORK

The bounds derived are quite loose: the reason is that they are valid for any function \( \psi \), no matter how random an ill-behaved, and for a generic graph. Hence there are two possible improvements: either consider a particular function (and the most interesting case is \( \psi = q \)), or improve the analysis for a particular graph. The analysis is particularly pessimistic for a circular graph, because \( \lambda_0/\lambda_2 \sim n^2 \), while in practice we observed that the disagreement is independent of the number of nodes.

The speed of convergence must be investigated further. Far from convergence, before the quantization error becomes relevant, the convergence appears to be exponential, and this should be easily proved. When the quantization error is dominant, the analysis appears complicated because of the nonlinearity of \( \psi \).

Moreover, there is a trade-off that should be investigated. By choosing a small \( \epsilon \), one can make the consensus as precise as desired. The trade-off is that a small \( \epsilon \) leads to slow convergence. The solution would be to start with a large \( \epsilon \) and then decrease; this would be a problem of gain scheduling in \( \epsilon(k) \).

Acknowledgements. Thanks to Li Na for disproving a conjecture of ours.

REFERENCES


### Table III

**EXPERIMENTS**

<table>
<thead>
<tr>
<th>graph</th>
<th>nodes</th>
<th>$\Delta$</th>
<th>$\lambda_1 L$</th>
<th>$\lambda_2 L$</th>
<th>$\lambda_1 L / \lambda_2 L$</th>
<th>Carli et al. $\phi(k)$</th>
<th>Proposed $\phi(k)$</th>
<th>bound (8)</th>
<th>bound (9)</th>
</tr>
</thead>
<tbody>
<tr>
<td>star</td>
<td>10</td>
<td>$n - 1$</td>
<td>$n$</td>
<td>1</td>
<td>$n$</td>
<td>0.01263707</td>
<td>0.00341001</td>
<td>0.00003262</td>
<td>1.22474487</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00016665</td>
<td>0.00000153</td>
<td>3.67423461</td>
<td></td>
</tr>
<tr>
<td>complete</td>
<td>10</td>
<td>$n - 1$</td>
<td>$n$</td>
<td>$n$</td>
<td>1</td>
<td>0.05904423</td>
<td>0.00756524</td>
<td>0.00001391</td>
<td>0.12247449</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00320243</td>
<td>0.000000117</td>
<td>0.12247449</td>
<td></td>
</tr>
<tr>
<td>ring</td>
<td>10</td>
<td>2</td>
<td>4</td>
<td>$2 - 2 \cos(2\pi/n)$</td>
<td>$n^2 + 2\pi^2$</td>
<td>0.04214771</td>
<td>0.01512196</td>
<td>0.00137799</td>
<td>1.28256948</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.02467507</td>
<td>0.00740776</td>
<td>0.00076440</td>
<td>11.20924853</td>
</tr>
<tr>
<td>chain</td>
<td>10</td>
<td>2</td>
<td>4</td>
<td>$2 - 2 \cos(2\pi/n)$</td>
<td>$n^2 + 2\pi^2$</td>
<td>0.03594107</td>
<td>0.01767899</td>
<td>0.00061120</td>
<td>4.88225660</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.00911454</td>
<td>0.01767899</td>
<td>0.00136018</td>
<td>44.59170875</td>
</tr>
</tbody>
</table>


Figure 2. Consensus on a circular graph with $n = 10$ nodes.

Figure 3. Encoding $\pi$ with the encoder described by (16), with 100 samples, using $\gamma = 0$ and $\gamma = -0.49$. In the first case there are 14 spikes, giving $S(k) = 3.14$, and in the second, there are 15, giving the approximation 3.15. In both cases, the error is compatible with the bounds (19)-(20).