An Efficient Quantization Algorithm for Solving Average-Consensus Problems

Giancarlo Baldan and Sandro Zampieri

Department of information Engineering, university of Padova, Padova, Italy. (emails: {gbaldani,zampi}@dei.unipd.it)

***Abstract*— In this work we present a new algorithm to solve the average–consensus problem. The main goal of this algorithm is to obtain exact convergence despite the existence of quantized communication channels between the agents. Starting from the Zoom–in Zoom–out strategy already presented in [5], we introduce the equations describing the behaviour of the algorithm and we formally prove the asymptotic agreement. We will also show that, under a reasonable hypothesis, the algorithm parameters ensuring convergence, can be chosen regardless the number of agents.**

I. INTRODUCTION

In recent years, motivated by the possible great diffusion of wireless networks, researchers have addressed their efforts in finding algorithms able to solve specific problems in a distributed way. Decentralized control of autonomous vehicles and distributed kalman filtering are two examples of what is potentially achievable exploiting, with a suitable algorithm, all the capabilities of a wireless network.

A common feature of these algorithms is that they have to take into account many constraints on the information flow, since the agents can exchange information through some communication network. Such a network will be hereafter modelled as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of agents $\mathcal{V} = \{1, \ldots, n\}$ while $\mathcal{E}$, the set of edges, is a subset of $\mathcal{V} \times \mathcal{V}$ such that $(i, j) \in \mathcal{E}$ iff the agent j can send information to the agent i.

One of the simplest problem for which a distributed algorithm have been found is the so called average–consensus problem, where a network of interconnected agents is required to compute the mean of some numbers. The first approach appeared in the literature (see, i.e. [1]) modelled every agent's state as a real value $x_i(t)$, and set the evolution in time accordingly to these difference equations

$$x_i(t+1) = \sum_{i=1}^{n} p_{ij} x_j(t), \tag{1}$$

where p_{ij} are coefficients complying with the communication constraints between agents, thus $p_{ij} \neq 0$ only if the edge (i, j) belongs to $\mathcal{E}$. Equation (1) imply that every agent, during algorithm evolution, keeps update his state with a proper average among his state and those of his neighbours.

More compactly we can write

$$x(t+1) = Px(t) = (I + K)x(t), \tag{2}$$

where $P = (p_{ij}) \in \mathbb{R}^{n \times n}$, $K = (k_{ij}) = P–I$ and $x(t) \in \mathbb{R}^n$ groups all agents states in a single state vector.

In [1] it is shown that, if the graph $\mathcal{G}$ is strongly connected, every irreducible doubly stochastic[1] matrix P drives the system dynamic to the consensus, namely

$$\lim_{t \to \infty} x(t) = \frac{\mathbb{1}^T x(0)}{n} \mathbb{1}, \tag{3}$$

where $\mathbb{1} \in \mathbb{R}^n$ is the vector of all ones.

This simply algorithm implicitly assumes also that the communication network is ideal, so that real numbers are exchanged between agents without any loss or degradation of information. Of course this assumption, in real applications, is not realistic due to energy and bandwidth limitations and, for this reason, a lot of literature has been devoted to investigate the effects of noise or packet drop on the algorithm performances (see i.e. [2] and [4]).

Another limitation of the algorithm (2) is that it requires the agents to communicate each others their actual states. As already pointed out, these measures belong to $\mathbb{R}$ or, more generally, to a non countable set and, thus, cannot be sent through a digital channel in a finite time. A naive solution to this problem is to send information with a loss of precision by coding data with a fixed number of bit. Usually this solution is implemented using the same coding used by agents CPU's to encode real numbers which is generally the standard floating-point double precision (64 bits) encoding provided by IEEE.

This solution has two great disadvantages. The first one is related to exploitation of the communication channel which is generally unoptimized. One need only reflect on the fact that, when the consensus is almost reached, the agents keep sending whole words of 64 bits length while the information is, at that point, contained only in a few less significant bits. The second disadvantage is that the precision loss during communication, together with round–off errors affecting agents computations, cause the algorithm to converge to a value that could be slightly different from the initial mean of the states.

⋆The research leading to these results has received funding from the European Community's Seventh Framework Programme under agreement n. FP7-ICT-223866-FeedNetBack.

[1]Under the assumption made on the graph, it is always possible to find such a matrix complying with communication constraints.

ISBN 978-963-311-369-1

In [3] and [5] some analysis on quantization effects in standard consensus algorithm are presented as well as an improved algorithm able to achieve exact average–consensus using a finite number of bits every algorithm step. Unfortunately the estimation on the required number of bits provided in [5] increases as the number of agents grows despite simulations show a different trend. Motivated by such numerical results, in this paper we present a novel algorithm, to which we will refer as the Zoom–in Zoom–out (ZIZO) algorithm, that can achieve exact convergence as well and whose parameters can be chosen, with some reasonable assumptions, regardless to the number of agents. Even if the theoretical finiteness of the number of bits required by this algorithm it is not yet proven, numerical results suggest that this number is finite, it depends only on the algorithm parameters and that it is independent of the number of agents.

II. ALGORITHM IMPLEMENTATION

The idea behind ZIZO–algorithm is to slightly modify equation (2) in order to obtain a state evolution of the form

$$x(t+1) = x(t) + K\hat{x}(t), \tag{4}$$

where $\hat{x}(t)$ is a suitable estimation of the real state at time t.

From (4) it turns out that each agent should keep trace of his neighbours' state estimations and has to perform additional operations in order to keep them updated.

In detail, generic ith agent has to manage the following variables:

- $x_i(t)$, the state of the agent.
- $\hat{x}_{ij}(t)\,\forall j : (i,j) \in \mathcal{E}$, the estimations, made by the agent i, of his neighbour j state as well as the estimation of his own measure even if this could seems paradoxical.
- $z_{ij}(t)$, the zoom factors associated to the estimation $\hat{x}_{ij}(t)$, whose aim will be clearer later.

The algorithm has three positive parameters k_1, k_2 and q as well as the matrix of coefficients K which is supposed to be given and designed to obtain, in the linear case, a suitable convergence rate over a given graph. At every time t the generic ith agent performs these steps:

1) It computes the quantization level l_i related to his state x_i, namely
$$l_i(t) = \left\lfloor \frac{x_i(t) - \hat{x}_{ii}(t)}{q|z_{ii}(t)|} \right\rfloor + \frac{1}{2} \tag{5}$$
if $|z_{ii}(t)| \neq 0$. Of course $l_i(t)$ takes values only in a countable set.
2) It sends the level $l_i(t)$ to all his neighbours. This is the step which involves communication between agents.
3) It computes the quantities
$$f_{ij}(t) = l_j(t)q|z_{ij}(t)| \tag{6}$$
using the levels l_j received from his neighbours.
4) It sets the variables for the next time step
$$\begin{cases} \hat{x}_{ij}(t+1) = \hat{x}_{ij}(t) + f_{ij}(t) \\ x_i(t+1) = x_i(t) + \displaystyle\sum_{j:(i,j)\in\mathcal{E}} k_{ij}\hat{x}_{ij}(t+1) \\ z_{ij}(t+1) = k_1|z_{ij}(t)|\text{sgn}(f_{ij}(t)) + k_2 f_{ij}(t) \end{cases} \tag{7}$$

where $\text{sgn}(x) = \begin{cases} 1 & \text{if} \quad x \geq 0 \\ -1 & \text{if} \quad x < 0 \end{cases}$.

It is clear from (5) and (6) that the magnitude of zoom factors afflict the state estimations accuracy. The bigger the zoom factors are, the roughly the estimations will be; this zoom–reliant estimation behaviour justify the name given to the algorithm. Since zoom factors play such a fundamental role, particular care was given to the design of their dynamics. More precisely in the third of (7), the term $k_2 f_{ij}(t)$ should guarantee the zoom factors to follow the difference between real states and estimations, thus improving the estimation quality as well as the estimations come close to the real states. The term $k_1|z_{ij}(t)|\text{sgn}(f_{ij}(t))$, on the other hand, prevent zoom factors from becoming too small in a single algorithm step as a consequence of a fortuitous coincidence between states and estimations

Remark 1: Note that, from the algorithm equations, the following holds

$$|z_{ij}(t)| = k_1|z_{ij}(t-1)| + k_2|f_{ij}(t)| \geq k_1|z_{ij}(t-1)|,$$

thus, if $z_{ij}(0) \neq 0$, we have $z_{ij}(t) \neq 0\ \forall t$ and equation (5) is always well–posed.

In the following part of this section we will look for a suitable nonlinear state–space system able to describe how agents' states and estimation variables evolve. For this purpose a useful simplification is given in the following proposition.

Proposition 2: If the ZIZO–Algorithm is initially synchronized, that is $\hat{x}_{ii}(0) = \hat{x}_{ji}(0)\ \ z_{ii}(k) = z_{ji}(0)\,\forall (i,j) \in \mathcal{E}$, then it stays synchronized.

Proof: The proof is done by induction. Since for $t = 0$ the thesis coincide with the hypothesis we suppose that the ZIZO–Algorithm is synchronized for $t = 1, 2, \ldots, k$, then we have

$$\hat{x}_{ii}(k) = \hat{x}_{ji}(k) \qquad z_{ii}(k) = z_{ji}(k)$$

. Using these relations, the updating equations yield

$$\begin{aligned} \hat{x}_{ij}(k+1) &= \hat{x}_{ij}(k) + f_{ij}(k) = \hat{x}_{jj}(k) + l_j(k)q|z_{ij}(k)| = \\ &= \hat{x}_{jj}(k) + f_{jj}(k) = \hat{x}_{jj}(k+1). \end{aligned}$$

$$\begin{aligned} \hat{z}_{ij}(k+1) &= k_1|z_{ij}(k)|\text{sgn}(f_{ij}(k)) + k_2 f_{ij}(k) = \\ &= k_1|z_{jj}(k)|\text{sgn}(f_{jj}(k)) + k_2 f_{jj}(k) = z_{jj}(k+1). \end{aligned}$$

which prove that the algorithm is synchronized $\forall t \geq 0$ ■

Under the hypothesis of proposition 2, the whole network state is described by $3n$ variables, namely x_i, $\hat{x}_i = \hat{x}_{ii} = \hat{x}_{ji}$ and $z_i = z_{ii} = z_{ji}$ and equations (7) can be easily rearranged as follow

$$\begin{cases} \hat{x}_i(t+1) = \hat{x}_i(t) + f_q(x_i(t) - \hat{x}_i(t), z_i(t)) \\ z_i(t+1) = g_{k_1,k_2,q}(x_i(t) - \hat{x}_i(t), z_i(t)) \\ x_i(t+1) = x_i(t) + \displaystyle\sum_{j:(i,j)\in\mathcal{E}} k_{ij}\hat{x}_j(t+1) \end{cases},$$

where f_q and $g_{k_1,k_2,q}$ are two nonlinear functions defined by

$$\begin{aligned} f_q(x - \hat{x}, z) &= q|z| \left(\frac{1}{2} + \left\lfloor \frac{x - \hat{x}}{q|z|} \right\rfloor \right), \\ g_{k_1,k_2,q}(x - \hat{x}, z) &= k_1|z|\text{sgn}\left(f_q(x - \hat{x}, z)\right) + \\ &\quad + k_2 f_q(x - \hat{x}, z). \end{aligned}$$

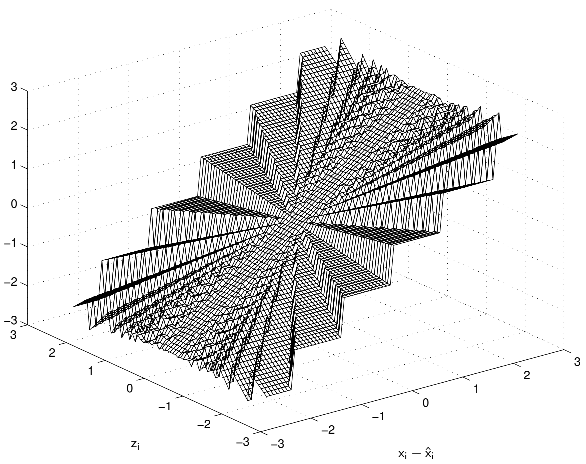

Fig. 1. A 3D visualization of the function $f_{0.4}(x_i - \hat{x}_i, z_i)$

From the visualization of function $f_q(x_i-\hat{x}_i, z_i)$ presented in Fig. 1, it is once again clear that the smaller the zoom factor z_i is, the more precise the difference between x_i and $\hat{x}_i$ sent from transmitter to receiver would be, thus improving the estimation accuracy.

More compactly the equations of the system can be written as

$$\begin{cases} x(t+1) &= x(t) + K\left[\hat{x}(t) + \Phi_q\left(x(t) - \hat{x}(t), z(t)\right)\right] \\ \hat{x}(t+1) &= \hat{x}(t) + \Phi_q\left(x(t) - \hat{x}(t), z(t)\right) \\ z(t+1) &= \Gamma_{k_1,k_2,q}\left(x(t) - \hat{x}(t), z(t)\right) \end{cases},$$

where we have grouped again the state variables into state vectors and we have collected the scalar nonlinear functions into two multidimensional functions

$$\Phi_q(x - \hat{x}, z) = \begin{bmatrix} f_q(x_1 - \hat{x}_1, z_1) \\ \vdots \\ f_q(x_n - \hat{x}_n, z_n) \end{bmatrix},$$

$$\Gamma_{k_1,k_2,q}(x - \hat{x}, z) = \begin{bmatrix} g_{k_1,k_2,q}(x_1 - \hat{x}_1, z_1) \\ \vdots \\ g_{k_1,k_2,q}(x_n - \hat{x}_n, z_n) \end{bmatrix}.$$

Finally, if we introduce the functions

$$\begin{cases} \overline{\Phi}_q(x - \hat{x}, z) = \Phi_q(x - \hat{x}, z) - (x - \hat{x}) \\ \overline{\Gamma}_{k_1,k_2,q}(x - \hat{x}, z) = \Gamma_{k_1,k_2,q}(\delta, z) - k_2(x - \hat{x}). \end{cases}, \quad (8)$$

the system becomes

$$\begin{cases} x(t+1) = Px(t) + K\overline{\Phi}_q\left(x(t) - \hat{x}(t), z(t)\right) \\ \hat{x}(t+1) = x(t) + \overline{\Phi}_q\left(x(t) - \hat{x}(t), z(t)\right) \\ z(t+1) = k_2(x(t)-\hat{x}(t)) + \overline{\Gamma}_{k_1,k_2,q}\left(x(t)-\hat{x}(t), z(t)\right) \end{cases}. \quad (9)$$

III. CONVERGENCE ANALYSIS

In this section we will prove that, if the agents are synchronized and the matrix P is such that the linear algorithm (2) can achieve consensus, the system (9) is able to drive the agents' states to the consensus as well, more precisely

$$\lim_{t\to\infty} \begin{bmatrix} x(t) \\ \hat{x}(t) \\ z(t) \end{bmatrix} = \frac{\mathbb{1}^T x_0}{n} \begin{bmatrix} \mathbb{1} \\ \mathbb{1} \\ 0 \end{bmatrix} \quad \forall \begin{bmatrix} x_0 \\ \hat{x}_0 \\ z_0 \end{bmatrix} \in \mathbb{R}^{3n}. \quad (10)$$

Since no analytical tools have been developed to study consensus–like convergence, this section is divided into two different part. In the first one we will transform the consensus problem into a stability one, to which a lot of literature has been devoted. In the second part we will then apply the well–known small–gain theorem to prove the convergence of the algorithm.

A. System reduction

We start pointing out that, since P is doubly stochastic, the system (9) is invariant in respect to any change of variables given by

$$y = x - \gamma\mathbb{1} \qquad \hat{y} = \hat{x} - \gamma\mathbb{1},$$

and, therefore, it suffices to prove (10) only for inital state with zero mean, that is

$$\lim_{t\to\infty} \begin{bmatrix} x(t) \\ \hat{x}(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \forall \begin{bmatrix} x_0 \\ \hat{x}_0 \\ z_0 \end{bmatrix} \in \mathbb{R}^{3n} : \mathbb{1}^T x_0 = 0. \quad (11)$$

Moreover, from the first of (9), we obtain[2]

$$\mathbb{1}^T x(t+1) = \mathbb{1}^T x(t),$$

which easily proofs that the mean of agents' states is preserved during algorithm execution or, in other words, that the system's trajectory will always lie in the subspace $\mathcal{U} = \left(\left[\mathbb{1}^T \; 0^T \; 0^T\right]^T\right)^\perp$ whenever the initial state has zero mean.

In order to obtain the equation of the system restricted to the subspace $\mathcal{U}$, we introduce the following linear transformation

$$\tilde{x} = Tx = \begin{bmatrix} I_{n-1} & 0 \\ \mathbb{1}^T & 1 \end{bmatrix} x = \begin{bmatrix} x_1 \\ \vdots \\ x_{n-1} \\ \mathbb{1}^T x \end{bmatrix} = \begin{bmatrix} \eta \\ \sigma \end{bmatrix}, \quad (12)$$

which is invertible since we have[3]

$$T^{-1} = \begin{bmatrix} I_{n-1} & 0 \\ -\mathbb{1}^T & 1 \end{bmatrix} = \left[\begin{array}{c|c} H & v \end{array}\right].$$

Using the relation $x = T^{-1}\tilde{x}$, the equations of the system in the new state space become

$$\begin{cases} \tilde{x}(t+1) = TPT^{-1}\tilde{x}(t) + TK\overline{\Phi}_q\left(T^{-1}\tilde{x}(t) - \hat{x}(t), z(t)\right) \\ \hat{x}(t+1) = T^{-1}\tilde{x}(t) + \overline{\Phi}_q\left(T^{-1}\tilde{x}(t) - \hat{x}(t), z(t)\right) \\ z(t+1) = k_2 T^{-1}\tilde{x}(t) - k_2\hat{x}(t) + \\ \qquad\qquad + \overline{\Gamma}_{q,k_1,k_2}\left(T^{-1}\tilde{x}(t) - \hat{x}(t), z(t)\right) \end{cases},$$

where the matrices TPT^{-1} and TK have many notable symmetries that can be appreciated introducing the following congruent partition on the matrices P and K

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}, \quad P_{11} \in \mathbb{R}^{n-1\times n-1},$$

[2]We remark that $\mathbb{1}^T P = \mathbb{1}^T$ and $\mathbb{1}^T K = 0^T$.

[3]It is easy to see that $TT^{-1} = T^{-1}T = I_n$.

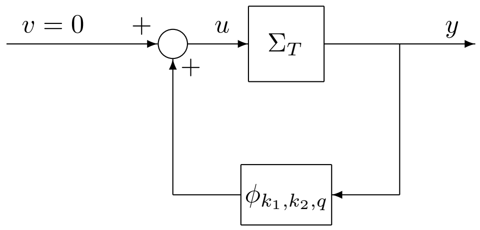

Fig. 2. Positive feedback connection between Σ_T and $\phi_{k_1,k_2,q}$

$$K=\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22}\end{bmatrix}, \quad K_{11}\in\mathbb{R}^{n-1\times n-1}.$$

With this partition we obtain

$$\begin{aligned} TPT^{-1} &= \begin{bmatrix} I_{n-1} & 0 \\ \mathbb{1}^T & 1\end{bmatrix}\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22}\end{bmatrix}\begin{bmatrix} I_{n-1} & 0 \\ -\mathbb{1}^T & 1\end{bmatrix} \\ &= \begin{bmatrix} P_{11}-P_{12}\mathbb{1}^T & P_{12} \\ 0^T & 1\end{bmatrix} = \begin{bmatrix} F & P_{12} \\ 0^T & 1\end{bmatrix}, \\ TK &= \begin{bmatrix} I_{n-1} & 0 \\ \mathbb{1}^T & 1\end{bmatrix}\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22}\end{bmatrix} \\ &= \begin{bmatrix} K_{11} & K_{12} \\ 0^T & 0\end{bmatrix} = \begin{bmatrix} G \\ 0^T \end{bmatrix}, \end{aligned} \tag{13}$$

and rewriting system's equations with the partitioned state inducted by (12), reminding that if the initial state has zero mean then $\sigma(t)=\mathbb{1}^T x(t)=0 \quad \forall t\geq 0$, the evolution of remaining variables is given by

$$\begin{cases} \eta(t+1) &= F\eta(t)+G\overline{\Phi}_q\left(H\eta(t)-\hat{x}(t),z(t)\right) \\ \hat{x}(t+1) &= H\eta(t)+\overline{\Phi}_q\left(H\eta(t)-\hat{x}(t),z(t)\right) \\ z(t+1) &= k_2H\eta x(t)-k_2\hat{x}(t)+ \\ & \quad +\overline{\Gamma}_{q,k_1,k_2}\left(H\eta(t)-\hat{x}(t),z(t)\right) \end{cases}, \tag{14}$$

where the state space has a smaller dimension of $3n-1$.

It suffices to observe that the state variable in (14) are now unconstrained and that the property (11) requires x, and thus $\tilde{x}$ and η, to converge toward zero, to conclude that the convergence of the algorithm is ensured whenever the system (14) is proven to be globally asymptotically stable.

Remark 3: Note that, since P has an eigenvalue in 1 while the others lie inside the unit circle, from (13) the matrix F turns out to be asymptotically stable.

B. Main result on convergence

To prove global asymptotic stability of system (14), we will use the small–gain theorem whose discrete–time version is well–presented in [9] (see also [11] for a dissertation on the bounded real lemma).

We start pointing out that the system (14) can easily be seen as a positive feedback interconnection between a linear system $\Sigma_T=(A_{k_2},B,C)$ and a static nonlinear function $\phi_{k_1,k_2,q}$ as shown in Fig. 2.

The linear system's matrices are given by

$$A_{k_2}=\begin{bmatrix} F & 0 & 0 \\ H & 0 & 0 \\ k_2H & -k_2I_n & 0\end{bmatrix}\in\mathbb{R}^{3n-1\times 3n-1}$$

$$B=\begin{bmatrix} G & 0 \\ I_n & 0 \\ 0 & I_n\end{bmatrix}\in\mathbb{R}^{3n-1\times 2n}$$

$$C=\begin{bmatrix} H & -I_n & 0 \\ 0 & 0 & I_n\end{bmatrix}\in\mathbb{R}^{2n\times 3n-1},$$

while the nonlinear function is defined as

$$\begin{array}{cccc} \phi_{k_1,k_2,q}: & \mathbb{R}^{2n} & \to & \mathbb{R}^{2n} \\ & y=\begin{bmatrix} y_1 \\ y_2\end{bmatrix} & \mapsto & \begin{bmatrix} \overline{\Phi}_q\left(y_1,y_2\right) \\ \overline{\Gamma}_{k_1,k_2,q}\left(y_1,y_2\right)\end{bmatrix}. \end{array}$$

In order to apply the small–gain theorem, we need to compute both the maximum $\mathcal{L}_2$ gain of $\phi_{k_1,k_2,q}$, γ_ϕ, and the $\mathcal{L}_2$ gain of the linear system Σ_T, γ_T.

The estimation of γ_ϕ is quite easy since from (8) we obtain the inequalities

$$\left|\overline{\Phi}_q^{(i)}\right|\leq\frac{q}{2}\left|y_2^{(i)}\right|,$$

$$\left|\overline{\Gamma}_{k_1,k_2,q}^{(i)}\right|\leq k_1\left|y_2^{(i)}\right|+k_2\frac{q}{2}\left|y_2^{(i)}\right|=\left(k_1+k_2\frac{q}{2}\right)\left|y_2^{(i)}\right|,$$

which lead us to

$$\begin{aligned} \|\phi_{k_1,k_2,q}(y)\|_2 &= \sqrt{\sum_{i=1}^n\left(\overline{\Phi}_q^{(i)}\right)^2+\sum_{i=1}^n\left(\overline{\Gamma}_{k_1,k_2,q}^{(i)}\right)^2} \\ &\leq \sqrt{\left(\frac{q}{2}\right)^2+\left(k_1+k_2\frac{q}{2}\right)^2}\sqrt{\sum_{i=1}^n\left(y_2^{(i)}\right)^2} \\ &\leq \sqrt{\left(\frac{q}{2}\right)^2+\left(k_1+k_2\frac{q}{2}\right)^2}\|y\|_2 \end{aligned}$$

Therefore for γ_ϕ we obtain the following bound

$$\gamma_\phi=\sup_{y\in\mathbb{R}^{2n}}\frac{\|\phi_{k_1,k_2,q}(y)\|_2}{\|y\|_2}\leq\sqrt{\left(\frac{q}{2}\right)^2+\left(k_1+k_2\frac{q}{2}\right)^2}. \tag{15}$$

The computation of γ_T is a more challenging task. We start recalling that the $\mathcal{L}_2$ gain of a linear system Σ coincide with the $\mathcal{L}_\infty$ norm of his transfer matrix $W(z)$ defined as

$$\begin{aligned} \gamma_T=\|W\|_\infty &= \max_{\vartheta\in[0,2\pi]}\sigma_{\max}\left[W(e^{j\vartheta})\right] \\ &= \sqrt{\max_{\vartheta\in[0,2\pi]}\lambda_{\max}\left[W^T(e^{-j\vartheta})W(e^{j\vartheta})\right]}, \end{aligned}$$

if W is BIBO-stable while in the case that $W(z)$ has some pole in $|z|>=1$, the norm is ∞.

The system Σ_T is certainly BIBO-stable because, as we have already pointed out, the matrix F is asymptotically stable and, thus, A_{k_2} is stable as well. After some simple calculations, the transfer matrix of Σ_T turns out to be given by

$$W_T(z)=\begin{bmatrix} W_d(z) & 0 \\ k_2z^{-1}W_d(z) & z^{-1}I_n\end{bmatrix},$$

where

$$W_d(z)=(1-z^{-1})H(zI_{n-1}-F)^{-1}G-z^{-1}I_n.$$

Then the matrix $W_T^T(e^{-j\vartheta})W_T(e^{j\vartheta})$ turns out to be:

$$\begin{bmatrix} (1+k_2^2)V_\vartheta & k_2e^{j\vartheta}W_d^T\left(e^{-j\vartheta}\right) \\ k_2e^{-j\vartheta}W_d\left(e^{j\vartheta}\right) & I_n\end{bmatrix},$$

where $V_\vartheta = W_d^T\left(e^{-j\vartheta}\right)W_d\left(e^{j\vartheta}\right)$.

The eigenvalues of $W_T^T(e^{-j\vartheta})W_T(e^{j\vartheta})$ can then be computed by solving the following equation in z

$$\det\begin{bmatrix} zI_n-(1+k_2^2)V_\vartheta & -k_2e^{j\vartheta}W_d^T\left(e^{-j\vartheta}\right) \\ -k_2e^{-j\vartheta}W_d\left(e^{j\vartheta}\right) & (z-1)I_n \end{bmatrix}=0. \quad (16)$$

Since the matrix in the last equation is a block–like matrix with commuting blocks, applying the result in [10], equation (16) is equivalent to

$$\begin{aligned} 0 &= \det\left[(z-1)\left(zI_n-(1+k_2^2)V_\vartheta\right)-k_2^2V_\vartheta\right] \\ &= \det\left[z^2I_n - z\left[I_n+(1+k_2^2)V_\vartheta\right]+V_\vartheta\right], \end{aligned}$$

from which we deduce, after a suitable diagonalization of the hermitian matrix V_ϑ, that the eigenvalues of $W_T^T(e^{-j\vartheta})W_T(e^{j\vartheta})$ are given by

$$\mu_\vartheta^{(i)}=\frac{1+(1+k_2^2)\lambda_\vartheta^{(i)}\pm\sqrt{(k_2^2+1)^2\left(\lambda_\vartheta^{(i)}\right)^2+2(k_2^2-1)\lambda_\vartheta^{(i)}+1}}{2},$$

where $\lambda_\vartheta^{(i)}$ are the eigenvalues of V_ϑ.

Since the scalar function

$$h(x)=\frac{1+(1+k_2^2)x+\sqrt{(k_2^2+1)^2x^2+2(k_2^2-1)x+1}}{2}$$

is monotone non decreasing in $[0,\infty)$, the gain γ_T turns out to be

$$\begin{aligned}\gamma_T &= \sqrt{\max_{\substack{\vartheta\in[0,2\pi]\\ i=1,\dots,n}}\mu_\vartheta^{(i)}} \\ &= \sqrt{\frac{1+(1+k_2^2)\rho+\sqrt{(k_2^2+1)^2\rho^2+2(k_2^2-1)\rho+1}}{2}}, \end{aligned} \quad (17)$$

where[4]

$$\rho=\max_{\substack{\vartheta\in[0,2\pi]\\ i=1,\dots,n}}\lambda_\vartheta^{(i)}=\max_{\substack{\vartheta\in[0,\pi]\\ i=1,\dots,n}}\lambda_\vartheta^{(i)}=\|W_d(z)\|_\infty^2.$$

In order to relate ρ to the given matrix P, we first need to do some manipulation on the expression for W_d. Let us start observing that

$$T\left(e^{j\vartheta}I_n-P\right)T^{-1}=\begin{bmatrix} e^{j\vartheta}I_{n-1}-F & -P_{12} \\ 0^T & e^{j\vartheta}-1\end{bmatrix},$$

which implies:

$$e^{j\vartheta}I_n-P=T^{-1}\begin{bmatrix} e^{j\vartheta}I_{n-1}-F & -P_{12} \\ 0^T & e^{j\vartheta}-1\end{bmatrix}T. \quad (18)$$

Since both member of (18) are non singular in $(0,\pi]$, we can obtain

$$\begin{aligned} &(e^{j\vartheta}I_n-P)^{-1}K = \\ &= T^{-1}\begin{bmatrix} \left(e^{j\vartheta}I_{n-1}-F\right)^{-1} & \star \\ 0^T & \left(e^{j\vartheta}-1\right)^{-1}\end{bmatrix}TK \\ &= \begin{bmatrix} H \mid v \end{bmatrix}\begin{bmatrix} \left(e^{j\vartheta}I_{n-1}-F\right)^{-1}G \\ 0^T \end{bmatrix} \\ &= H(e^{j\vartheta}I_{n-1}-F)^{-1}G. \end{aligned}$$

[4]We remark that, since V_ϑ is Hermitian, his eigenvalues are symmetric with respect to π.

Thanks to this last relation, we can rewrite the expression for W_d as follows

$$\begin{aligned} &W_d(e^{j\vartheta}) = \\ &= (1-e^{-j\vartheta})H\left(e^{j\vartheta}I_{n-1}-F\right)^{-1}G-e^{-j\vartheta}I_n \\ &= (1-e^{-j\vartheta})\left(e^{j\vartheta}I_n-P\right)^{-1}(P-I_n)-e^{-j\vartheta}I_n \\ &= (1-e^{-j\vartheta})\left(e^{j\vartheta}I_n-P\right)^{-1}(P-I_n)+(1-e^{-j\vartheta})I_n-I_n \\ &= (1-e^{-j\vartheta})\left(e^{j\vartheta}I_n-P\right)^{-1}\left[(P-I_n)+(e^{j\vartheta}I_n-P)\right]-I_n \\ &= -\left|1-e^{-j\vartheta}\right|^2\left(e^{j\vartheta}I_n-P\right)^{-1}-I_n \\ &= -2(1-\cos\vartheta)\left(e^{j\vartheta}I_n-P\right)^{-1}-I_n \quad \forall\vartheta\in(0,\pi]. \end{aligned}$$

If we finally suppose the matrix P to be symmetric, the expression for V_ϑ yields

$$\begin{aligned} V_\vartheta = &\left[2(1-\cos\vartheta)\left(e^{-j\vartheta}I_n-P\right)^{-1}+I_n\right]\times \\ &\times\left[2(1-\cos\vartheta)\left(e^{j\vartheta}I_n-P\right)^{-1}+I_n\right]. \end{aligned}$$

Let now v be an eigenvector of P related to the eigenvalue λ. Reminding that, since the matrix $e^{\pm j\vartheta}I_n-P$ is invertible then v is also an eigenvector of the matrix $\left(e^{\pm j\vartheta}I_n-P\right)^{-1}$ related to the eigenvalue $\frac{1}{e^{\pm j\vartheta}-\lambda}$, we get:

$$\begin{aligned} V_\vartheta v &= \left[-\frac{2(1-\cos\vartheta)}{e^{j\vartheta}-\lambda}-1\right]W_d(e^{-j\vartheta})v \\ &= \left[\frac{2(1-\cos\vartheta)}{e^{-j\vartheta}-\lambda}+1\right]\left[\frac{2(1-\cos\vartheta)}{e^{j\vartheta}-\lambda}+1\right]v \\ &= \frac{2(\lambda-2)\cos\vartheta+\lambda^2-4\lambda+5}{1-2\lambda\cos\vartheta+\lambda^2}v. \end{aligned} \quad (19)$$

The (19) proves that every eigenvalue of V_ϑ can be obtained as

$$\mu_\lambda(\vartheta)=\frac{2(\lambda-2)\cos\vartheta+\lambda^2-4\lambda+5}{1-2\lambda\cos\vartheta+\lambda^2},$$

where λ is an eigenvalue of P.

An easy calculus shows that the derivative of $\mu_\lambda(\vartheta)$ with respect to ϑ is given by

$$\frac{d\mu_\lambda(\vartheta)}{d\vartheta}=\frac{4(1-\lambda)^3\sin\vartheta}{(1-2\lambda\cos\vartheta+\lambda^2)^2},$$

which, in turn, proves[5] that the eigenvalues of V_ϑ are all monotone nondecreasing in $(0,\pi]$.

By virtue of this result, the expression for ρ become

$$\begin{aligned} \rho &= \max_{\substack{\vartheta\in[0,\pi]\\ \lambda\in\Lambda(P)}}\mu_\lambda(\vartheta)=\max_{\lambda\in\Lambda(P)}\mu_\lambda(\pi) \\ &= \max_{\lambda\in\Lambda(P)}\left(\frac{3-\lambda}{1+\lambda}\right)^2=\left(\frac{3-\lambda_{\min}}{1+\lambda_{\min}}\right)^2, \end{aligned} \quad (20)$$

where $\lambda_{\min}$ is the minimum eigenvalue of P and the last equality holds since $\left(\frac{3-\lambda}{1+\lambda}\right)^2$ is monotone decreasing in $(-1,1]$.

All the results obtained in this section are summarized in the following theorem.

[5]We remark that, since P is stochastic, then $|\lambda|\le 1$.

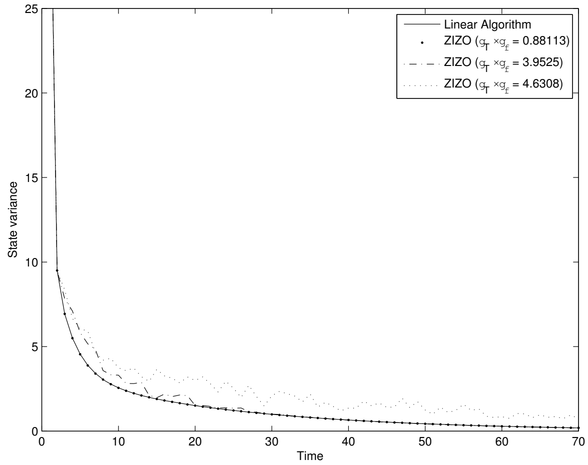

Fig. 3. State variance evolution with different parameters' choices

Theorem 4 (Convergence Theorem): Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph and let P be an irreducible doubly stochastic symmetric matrix complying with the communication constraints given by $\mathcal{G}$. Let ρ defined in (20). Then the ZIZO–Algorithm converge to the average–consensus regardless[6] of the initial choice of vectors x_0, $\hat{x}_0$ and z_0 if the three positive parameters of the algorithm q, k_1 and k_2 satisfy the following inequality:

$$\left[\left(\frac{q}{2}\right)^2 + \left(k_1 + k_2\frac{q}{2}\right)^2\right] \times$$
$$\times \frac{1+(1+k_2^2)\rho + \sqrt{(k_2^2+1)^2\rho^2+2(k_2^2-1)\rho+1}}{2} < 1.$$

Proof: The convergence is ensured by the small–gain theorem, provided that the product of γ_T and γ_ϕ is less than 1. The thesis then easily follows from the expression for γ_ϕ and γ_T, respectively found in (15) and (17). ■

As a consequence of theorem 4, ZIZO–algorithm's parameters depend only on the minimum eigenvalue of the matrix P and the convergence is assured whenever the matrix P guarantee convergence using the linear algorithm[7]. Since the minimum eigenvalue is usually independed of the number of agents[8] or can be forced to be greater to any constant β without any further constraint on $\mathcal{G}$ providing that the distributed constraints $p_{ii} \geq \frac{\beta+1}{2}$ hold, it turns out that the algorithm's parameters can be chosen regardless to the dimension of $\mathcal{G}$.

In Fig. 3 we show some simulation results obtained with a random geometric graph with 20 agents. The weights' matrix P was generated using Metropolis–algorithm, thus obtaining $\rho \simeq 12.7$, and the algorithm's parameters are chosen in order to obtain different values for the closed loop gain $\gamma_T\gamma_\phi$. From these numerical results it seems that the result stated in theorem 4 is still improvable since the algorithm seems to converge even for closed loop gain greater than 1; moreover the ZIZO–algorithm's performances seems very close to those of the linear algorithm (2) whenever the parameters fulfill the requirements of theorem 4 despite any bound on the convergence rate derivable from the small gain theorem seems very poor. This discrepancy between theoretical and sperimental results is not surprising and is due to the conservativeness of the small-gain theorem.

[6] As already pointed out, we need to force $z_0^i \neq 0$ to ensure the implementability of the algorithm. What really matter is the arbitrariness on the choice of x_0.

[7] This implies that $\mathcal{G}$ must be connected

[8] So it is in the wide family of random geometric graphs.

IV. CONCLUSIONS AND FUTURE WORK

In this work we presented a novel algorithm able to achieve average consensus in a network of agents with quantized transmissions. We showed that the algorithm's parameters can be chosen in a distributed fashion without knowing the number of agents composing the network, thus making it suitable in distributed applications such as wireless sensor networks. We also provided simulations results showing that the performances of the algorithm can be very close to those of the ideal algorithm. An investigation on the number of bits required by the algorithm is part of our plan for the future. From simulations the algorithm seems to use a number of bits which depends on the parameters choice but never exceeding 14 bits in over a million simulations. Moreover the algorithm does not seems to be very robust to noise effects, due to unsynchronization between agents as a consequence of transmission errors. Different algorithms able to keep the synchronization are also being studied.

REFERENCES

[1] R. Olfati–Saber, R. M. Murray. Consensus problems in networks of agents with switching topology and time–delays. *IEEE Transactions on automatic control,* 49(9):1520–1533, 2004

[2] S. Boyd, A. Ghosh , B. Prabhakar, D. Shah. Randomized gossip algorithms. *IEEE Transactions on information theory,* 52(6):2508–2530, 2006

[3] R. Carli, F. Fagnani, P. Frasca, T. Taylor, S. Zampieri. Average consensus on networks with transmission noise or quantization. *Proc. of European Control Conference,* 2007

[4] F. Fagnani, S. Zampieri. Average consensus with packet drop communication. *45th IEEE Conference on Decision and Control,* 2006

[5] R. Carli, F. Fagnani, P. Frasca, S. Zampieri. Efficient quantized techniques for consensus algorithms. *Proc. of the 3rd international workshop on Networked Control Systems: Tollerant to faults*, Nancy, 2007

[6] R. W. Brockett, D. Liberzon. Quantized feedback stabilization of linear systems. *IEEE Transactions on automatic control,* 45(7): 1279–1289, 2000

[7] A. Nedić, A. Olshevsky, A. Ozdaglar, J. Tsitsiklis. On distributed average consensus algorithms and quantization effecs. *Submitted,* 2007.

[8] A. Kashyap, T. Başar, R. Srikant. Quantized consensus. *Automatica,* 43: 1192–1203, 2007

[9] W. M. Haddad, D. S. Bernstein. Explicit construction of quadratic Lyapunov functions for the small gain, positivity, circle, and popov theorems and their application to robust stability. Part II: discrete–time theory. *International journal of robust and nonlinear control,* 4:249–265, 1994

[10] J. R. Silvester Determinants of block matrices. *The Mathematical Gazette* 84(501):460–467, 2000

[11] C. E. de Souza, L. Xie. On the discrete–time Bounded Real Lemma with application in the characterization of static state feedback H_∞ controllers. *Systems & Control Letters* 18:61–71, 1992