



**HAL**  
open science

# Adding a single memory per agent gives the fastest average consensus

Alain Sarlette

► **To cite this version:**

Alain Sarlette. Adding a single memory per agent gives the fastest average consensus. 2014. hal-01093907v1

**HAL Id: hal-01093907**

**<https://inria.hal.science/hal-01093907v1>**

Preprint submitted on 5 Jan 2015 (v1), last revised 28 Dec 2015 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Adding a single memory per agent gives the fastest average consensus

A. Sarlette — alain.sarlette@inria.fr  
INRIA team QUANTIC, Rocquencourt, France  
& SYSTeMS, Ghent University, Belgium.

November 2014

## Abstract

Previous papers have proposed to add memory registers to the individual dynamics of discrete-time linear agents to move faster towards average consensus under interactions dictated by a given but unknown graph. They have proved that adding one memory slot per agent allows faster convergence. We here prove that this situation *cannot* be improved by adding more memory slots. We conclude by discussing a more general framework for our result in an algorithmic context.

## 1 Problem definition

The basic consensus algorithm in discrete time [1] writes

$$x(t+1) = x(t) - \alpha L x(t), \quad x = [x_1; x_2; \dots; x_N] \in \mathbb{R}^N, \quad (1)$$

where  $x_i \in \mathbb{R}$  is the state of agent  $i$ ,  $\alpha > 0$  is a gain, and  $L$  is the Laplacian matrix characterizing interactions among agents: component  $j$  of  $Lx$  equals  $\sum_{i=1}^N w_{j,i}(x_j - x_i)$  with weights  $w_{j,i} \geq 0$ . Usually  $L$  contains many zeros, as each agent interacts with a limited number of fellows. For the algorithm to compute the average of the initial values  $x_i(0)$ , we assume  $w_{j,i} = w_{i,j}$  for all  $i, j$ . The Laplacian is then symmetric nonnegative definite, and if the interactions form a connected graph it has a single eigenvalue  $\lambda_1 = 0$  with eigenvector  $v_1 = [1; 1; \dots; 1]$ , i.e. consensus. Correspondingly, the average  $\frac{1}{N} \sum_{i=1}^N x_i(t)$  is invariant under (1). When  $(I_N - \alpha L)$  has nonnegative entries, with  $I_N$  the  $N \times N$  identity matrix, it can equivalently be viewed as the transition matrix of a Markov chain; symmetric  $L$  implies that the limiting distribution is uniform.

For time-invariant  $L$ , the convergence speed of (1) is dictated by the largest eigenvalue, in modulus, of  $(I_N - \alpha L)$  excluding the trivial  $\lambda_1 = 0$ . In the orthonormal basis corresponding to the eigenvectors of  $L$  (so-called “modes”), the system decouples into

$$\tilde{x}_i(t+1) = (1 - \alpha \lambda_i) \tilde{x}_i(t), \quad i = 1, 2, \dots, N, \quad (2)$$

with  $\tilde{x}_i$  the coefficient of mode  $i$ . If we only know that the eigenvalues of  $L$  belong to an interval  $\lambda_i \in [\underline{\lambda}, \bar{\lambda}] \subset \mathbb{R}_{>0}$  for  $i = 2, 3, \dots, N$ , then the convergence speed — in terms of eigenvalues of (2) guaranteed over all  $\lambda_i \in [\underline{\lambda}, \bar{\lambda}]$  — is optimal when  $\alpha$  is selected to satisfy  $(1 - \alpha \underline{\lambda}) = -(1 - \alpha \bar{\lambda})$ . This gives  $\alpha = \frac{2}{\underline{\lambda} + \bar{\lambda}}$  and worst eigenvalue  $\mu = \frac{\bar{\lambda} - \underline{\lambda}}{\bar{\lambda} + \underline{\lambda}}$ .

In practice, the time step of discrete-time consensus is mostly limited by communication speed, much more rarely by local computation power. Hence [2] and later [3, 4, 5] propose to improve convergence speed by adding local dynamics to each agent, namely a memory slot:

$$x_i(t+1) = x_i(t) - \alpha \sum_{j=1}^N w_{i,j} (x_i(t) - x_j(t)) - \beta_1 (x_i(t) - x_i(t-1)). \quad (3)$$

The papers analyze in detail only this case of one memory slot and prove that with  $\beta_1 < 0$  it allows faster convergence. This spurs the natural question:

**How much can be gained by adding more memory to the system?**

The answer is **strictly nothing**, as we prove in the present paper:

**Theorem 1:** *Consider a consensus dynamics with memory slots, of the type:*

$$x_i(t+1) = x_i(t) - \alpha \sum_{j=1}^N w_{i,j} (x_i(t) - x_j(t)) - \sum_{k=1}^{M-1} \beta_k (x_i(t) - x_i(t-k)), \quad (4)$$

where  $\alpha, \beta_1, \beta_2, \dots, \beta_{M-1}$  can be freely chosen in  $\mathbb{R}$  while the  $w_{i,j} = w_{j,i}$  are fixed and define a symmetric Laplacian  $L$ . If we only know about  $L$  that its nonzero eigenvalues  $\lambda_i$  belong to  $[\underline{\lambda}, \bar{\lambda}] \subset \mathbb{R}_{>0}$ , then the fastest convergence guarantee<sup>1</sup> is obtained by taking  $\beta_k = 0$  for all  $k > 1$  and optimizing  $\alpha, \beta_1$  for a one-memory-slot consensus as in [2].

**Remark 1:** In [2] and related work, it is mentioned that  $\beta_1 < 0$  is necessary to get any acceleration over the memoryless case. As an auxiliary result, it is not difficult to show for (4) that if we are restricted to  $\beta_k \geq 0$  for all  $k$  then no acceleration with respect to the memoryless case can be achieved.

The fact that the eigenvalues of  $L$  can take *any* values in  $[\underline{\lambda}, \bar{\lambda}]$  is important for the property to hold. If more is known about the  $\lambda_i$ , then there are faster solutions (see Remark 2 at the end of Section 2). The following proof for Theorem 1 is a polynomial analysis. We currently have no satisfactory intuitive explanation. It should be linked to gain-robust optimal (“ensemble optimal”) control, although in discrete-time we are not aware of the corresponding result. Equivalently, it is an optimal control problem for a so-called (non-symmetric) interval matrix or linear parametric uncertainty system, along the lines of Kharitonov’s theorem. Unfortunately the latter does not generalize easily to discrete-time systems [6, 7], it is hence unclear how this approach might yield an analytical fastest convergence result. Theorem 1 probably also links to information theory, considering an agent symmetrically exchanging data with a poorly characterized network.

## 2 General reformulation

As all the agents have the same local dynamics, (4) is still decoupled into the modes of  $L$ :

$$\tilde{x}_i(t+1) = (1 - \alpha\lambda_i)\tilde{x}_i(t) - \sum_{k=1}^{M-1} \beta_k (\tilde{x}_i(t) - \tilde{x}_i(t-k)).$$

---

<sup>1</sup>worst eigenvalue of (4) over all  $\lambda_i \in [\underline{\lambda}, \bar{\lambda}]$

The convergence speed for each mode is governed by the roots of

$$P(z) = z^M - (1 - \alpha \lambda_i) z^{M-1} + \sum_{k=1}^{M-2} \beta_{M-k-1} (z^{M-1} - z^k). \quad (5)$$

This set can also be seen as an ensemble of closed-loop systems resulting from proportional feedback with different gains  $\lambda_i$ . We then want to design the (unique) plant  $\alpha, \beta_1, \dots, \beta_{M-1}$  to get the fastest possible worst-case performance over the ensemble of feedback gains  $\lambda_i \in [\underline{\lambda}, \bar{\lambda}]$ .

## 2.1 Optimal solution with one memory slot

The optimal tuning [2] for  $M = 2$  can be obtained as follows. Note that  $\beta_1 \geq 1$  always leads to an unstable eigenvalue. Then explicitly write the roots  $z_{\pm}$  of  $P(z)$  as a function of  $\lambda'_i = (1 - \alpha \lambda_i - \beta_1)/(1 - \beta_1)$ .

- We observe that for fixed  $\beta_1$ , the worst root of  $P(z)$  is a monotonically increasing function of  $|\lambda'_i|$ . Thus for fixed  $\beta_1$ , we choose  $\alpha$  such that  $(1 - \alpha \underline{\lambda} - \beta_1) = -(1 - \alpha \bar{\lambda} - \beta_1) =: \mu(1 - \beta_1)$  and get  $\mu = \frac{\bar{\lambda} - \underline{\lambda}}{\lambda + \bar{\lambda}}$ . Indeed this minimizes  $\max_i(|\lambda'_i|)$ , hence the worst root for given  $\beta_1$ .

- Assuming this optimal  $\alpha$  for each  $\beta_1$ , and considering the worst Laplacian eigenvalue  $|\lambda'_k| = \mu$ , we observe that the associated  $\max\{|z_+|, |z_-|\}$  is minimized when  $\beta_1$  is chosen to satisfy  $z_+ = z_-$ . Furthermore, we observe that for  $|\lambda'_j| < \mu$  the roots will then be complex and have the same modulus as for  $|\lambda'_k| = \mu$ . This determines the optimal values

$$\begin{aligned} \beta_1 &= -\left(\frac{2}{\mu^2} - 1\right) + \frac{2}{\mu^2} \sqrt{1 - \mu^2} < 0, \\ |z_{\pm}| &= \nu := \frac{1}{\mu} - \sqrt{\frac{1}{\mu^2} - 1} \quad \forall \lambda_j \in [\underline{\lambda}, \bar{\lambda}]. \end{aligned}$$

This  $\nu$  increases with  $\mu$ . For  $\mu = 1 - \varepsilon$ , we have  $\nu < 1 - \sqrt{\varepsilon}$  and the bound gets tight as  $\varepsilon \rightarrow 1$ .

## 2.2 General comparison form

As our goal is to show that one memory slot is optimal, we reformulate (5) using the just computed result. We first reformulate the effect of  $\alpha$ . Starting from

$$P(z) = (z - 1)z^{M-1} + \alpha \lambda_i z^{M-1} + (z - 1)P_{M-2}(z)$$

with completely free polynomial  $P_{M-2}(z)$  of order  $M - 2$ , we take  $\alpha_*$  the optimal value from Section 2.1 and multiply by  $\alpha_*/\alpha$  to obtain

$$P_*(z) = \alpha_* \lambda_i z^{M-1} + (z - 1) \left( \frac{\alpha_*}{\alpha} z^{M-1} + P'_{M-2}(z) \right)$$

for some other polynomial  $P'_{M-2}$ . This is well-defined since  $\alpha = 0$  is not an option: it would mean zero use of network information. The last bracket now contains a free polynomial of order  $M - 1$ . Separating out the choice that would retrieve the case of Section 2.1, we write:

$$P_*(z) = (z - \nu e^{i\theta_k})(z - \nu e^{-i\theta_k})z^{M-2} + (z - 1)P_{M-1}(z).$$

The first term represents the optimal polynomial when using only one memory slot. As explained above, the roots in that case have a constant modulus  $\nu$  for all  $\lambda'_k$  and their phases span the values from  $\theta_k = 0$  for  $\lambda'_k = \mu$  through to  $\theta_k = \pi$  for  $\lambda'_k = -\mu$ . The second term,

with a free polynomial  $P_{M-1}$  of degree at most  $M-1$ , represents all changes that are possible by modifying  $\alpha, \beta_1$  and by adding further memory slots with weights  $\beta_2, \beta_3, \dots, \beta_{M-1}$ .

To prove Theorem 1, we must show that for any  $P_{M-1}(z)$ , there will always be a  $\theta_k$  for which  $P_*(z)$  has a root of modulus  $\geq \nu$ . Defining  $y = z/\nu$ , we must equivalently prove that for any  $\nu$  and  $P'_{M-1}$ , there exists a  $\theta_k$  for which

$$P'(y) = (y - e^{i\theta_k})(y - e^{-i\theta_k})y^{M-2} + (y - \frac{1}{\nu})P'_{M-1}(y) \quad (6)$$

has a root of modulus  $\geq 1$ . This looks like a discrete-time robust (in)stability question. Surprisingly, we know no standard result that would straightforwardly answer it. Hence we provide an answer by explicitly analyzing  $P'$  in the complex plane.

**Remark 2:** *It is important to insist that we examine the worst possible roots over all  $\lambda_k$  in the interval  $[\underline{\lambda}, \bar{\lambda}]$ , hence over all  $\theta_k \in [0, \pi]$ . Indeed, if more is known about the  $\lambda_k$ , then it is possible to better speed up the system e.g. with so-called polynomial-based algorithms [8, 9, 10]. In the extreme case where  $L$  has  $M$  different eigenvalues whose values are exactly known, it is possible to construct a polynomial of order  $M$  that achieves deadbeat convergence on each eigenvalue of  $L$  at a time, with simple periodic algorithms. These can be translated to invariant memory-slot-based dynamics at least for specific initializations.*

### 3 Proving: any $P'(y)$ has a $|\text{root}| \geq 1$ for some $\theta_k$

The roots of  $P'(y)$  are the points in the complex plane where

$$\begin{aligned} P_1 &:= (y - e^{i\theta_k})(y - e^{-i\theta_k})y^{M-2} \quad \text{and} \\ P_2 &:= (\frac{1}{\nu} - y)P'_{M-1}(y) \end{aligned}$$

have both the same phase modulo  $2\pi$  and the same magnitude. We examine such points as a function of the roots of  $P_1$  and  $P_2$ .  $P_1$  has  $M-2$  roots at zero and two complex conjugate roots on the unit circle, which we should place for the worst for each given  $P_2$ . The latter has degree  $M-g$  for some  $g \geq 0$ , it has at least one root  $1/\nu > 1$  on the positive real axis, and all other roots basically free. Note that the magnitude of polynomials is continuous everywhere, while the phase modulo  $2\pi$  is continuous everywhere except at roots.

Let us first exclude a special case. If  $P_2$  has a root  $y_*$  on the unit circle, then taking  $e^{i\theta_k} = y_*$  readily assigns this same root to  $P'$ , hence it trivially has a root of modulus  $\geq 1$ ; thus it cannot improve the single-memory-slot case. The remaining  $P_2$  candidates have the following property.

**Lemma 3.0:** *For any given  $P_2$  with no roots on the unit circle, there exists a  $\delta > 0$  such that for  $|y| \geq 1$  we have:  $|P_1(y)| > |P_2(y)|$  whenever  $y$  is in a  $\delta$ -neighborhood of a zero of  $P_2$  and conversely  $|P_2(y)| > |P_1(y)|$  whenever  $y$  is in a  $\delta$ -neighborhood of a zero of  $P_1$ .*

Proof: Since the zeros of  $P_1$  are disjoint from those of  $P_2$  for  $|y| \geq 1$ , we have whenever  $y_1$  is a zero of  $P_2$ :  $\lim_{y \rightarrow y_1} (P_2(y)/P_1(y)) = 0$ . We can then build a circular neighborhood around each such zero where  $P_2(y)/P_1(y) < 1$ . After doing the same around zeros of  $P_1$ , we take  $\delta$  to be the minimum radius over all the neighborhoods.  $\square$

### 3.1 A link with root locus analysis

We will consider the locus in the complex plane where  $P_1$  and  $P_2$  have the same phase modulo  $2\pi$ . This is equivalent to a root locus analysis [11],[12, Chap.6.2]: *In fact the equal-phases curves are the curves which the roots of*

$$1 + \kappa \frac{-P_2(s)}{P_1(s)} \quad , \quad s \in \mathbb{C} \quad , \quad (7)$$

would follow as the gain  $\kappa$  varies between 0 and  $+\infty$ . The root locus formulation tells us that the locus where  $P_1$  and  $P_2$  have the same phase is a set of  $M$  possibly intersecting curves<sup>2</sup>. Other properties follow:

**Lemma 3.1:** *The root locus curves have the following properties:*

(a) *If  $P_1$  and  $P_2$  have the same degree i.e.  $g = 0$ , then each root locus curve connects a zero of  $P_1$  to a zero of  $P_2$ .*

(b) *If  $g > 0$ , then  $g$  remaining curves starting at a zero of  $P_1$  head to infinity.*

(c) *A root locus curve does never self-intersect.*

*When root locus curves intersect, we have some freedom about which outgoing branch we associate with which incoming branch.*

(d) *For every branch of a root locus curve with negative imaginary part there is a symmetric branch with positive imaginary part, and one canonically selects a root locus curve to always stay in the closed e.g. upper half plane.*

(e) *For whatever branching choice, when we move from a zero of  $P_2$  to the corresponding zero of  $P_1$  along a root locus curve, at our left the phase of  $P_2$  is slightly larger than the phase of  $P_1$  while at our right the phase of  $P_2$  is slightly smaller.*

All these properties are standard, except maybe the last one which is a direct consequence of local phase analysis. The special feature of our problem is that for any chosen  $P_2$ , we should examine the root loci for the worst possible  $e^{i\theta_k}$ .

### 3.2 How displacing $e^{i\theta_k}$ affects the root locus

We are able to characterize as follows how displacing  $e^{i\theta_k}$  affects the phase of  $P_1(y)$ . From there we get a key result about the dependence of the root locus curves on  $\theta_k$ .

**Lemma 3.2:** (a) *When  $y = e^{i\phi}$ ,  $\phi \in [0, \pi]$  evolves along the exterior upper half of the unit circle, the phase of  $P_1(y)$  equals  $(M - 1)\phi$  for all  $\phi < \theta_k$  and  $(M - 1)\phi + \pi$  for all  $\phi > \theta_k$ .*

(b) *When  $y = r e^{i\phi}$ ,  $\phi \in [0, \pi]$ ,  $r > 1$  evolves in the upper half plane strictly outside of the unit circle, the phase of  $P_1(y)$  is bounded strictly inside  $M\phi + (-\pi, \pi)$  for any value of  $\theta_k$ . For fixed  $y$ , it monotonically decreases as  $\theta_k$  increases.*

**Proof:** Denote by  $\phi_A$  and  $\phi_B$  the phases of  $(y - e^{i\theta_k})$  and of  $(y - e^{-i\theta_k})$  respectively. The phase of  $P_1(y = r e^{i\phi})$  equals  $(M - 2)\phi + \phi_A + \phi_B$ . Consider  $y = e^{i\phi}$  with  $\phi < \theta_k$ . By the inscribed angle theorem, the angle made by  $(e^{i\theta_k}, e^{i\phi}, e^{-i\theta_k})$  equals  $\pi - \theta_k$ . Then considering the equal angles in the isosceles triangle  $0, e^{i\phi}, e^{-i\theta_k}$  we get  $\pi/2 - \theta_k + \pi/2 - \phi_B = \pi - \theta_k - \phi + \phi_A$ , which leads to the first equality of (a). For  $\phi > \theta_k$  the proof is similar.

For  $y = r e^{i\phi}$ , the angle  $|\phi_A - \phi|$  made by  $(0, r e^{i\phi}, e^{i\theta_k})$  lies outside the circle and spans

---

<sup>2</sup>Unless  $P_2 \equiv \eta P_1$  for some  $\eta \in \mathbb{R}$ , which we can exclude as  $P_2$  has a zero outside the unit circle and  $P_1$  not.

half a circle, hence it is strictly lower than  $\pi/2$ . The same holds for  $\phi_B$ , hence  $2\phi$  is a  $\pi$ -accurate approximation of  $\phi_A + \phi_B$ . The monotonicity is obtained by analyzing the phase of  $f(t) = y^2 + 1 - 2ty$  as a function of  $t = \cos(\theta) \in [-1, 1]$ ; this  $f(t)$  describes a line in  $\mathbb{C}$ .  $\square$

**Corollary 3.3:** *Consider  $\mathcal{S}_{\theta_k} \subseteq \{y \in \mathbb{C} : \text{Im}(y) \geq 0, |y| \geq 1\}$  a singly connected compact set, whose boundary is composed*

- *in part by a canonical root locus curve  $\mathcal{C}_{\theta_k}$  for some fixed  $\theta_k$ , and such that  $\mathcal{S}_{\theta_k}$  lies to the left when following the curve from a zero of  $P_2$  towards a zero of  $P_1$ ;*
- *for the other part by any fixed curve.*

*Then  $\mathcal{C}_{\theta_k}$  moves (non-strictly) towards the interior of  $\mathcal{S}_{\theta_k}$  when  $\theta_k$  decreases.*

Proof: At all points where  $\mathcal{C}_{\theta_k}$  does not pass through a zero of  $P_2$ , i.e. the phases are continuous, this is a direct consequence of Lemma 3.1(e) and Lemma 3.2(b) — irrespective of the fact that the topology of the different root loci might be modified as  $\theta_k$  changes. Whereas just (7) itself excludes that the root locus might pass in a neighborhood of a zero of  $P_2$  which is not also a zero of  $P_1$  (the latter exception was treated at the beginning of the Section) for any finite  $\kappa > 0$ .  $\square$

### 3.3 Constructing a counterexample for any $P_2$

We conclude with the following construction. Consider any fixed  $P_2$ , select  $\delta$  according to Lemma 3.0 and let  $0 < \varepsilon < \delta \ll 1$ .

1. Take  $\theta_k = \pi - \varepsilon$  and draw a root locus curve  $\mathcal{C}'$  that starts at the root  $1/\nu$  of  $P_2$  (Lemma 3.1(a),(b)) and that always stays in the upper half complex plane (Lemma 3.1(d)). As it has to connect to a zero of  $P_1$  (Lemma 3.1(a),(b)),  $\mathcal{C}'$  must intersect the unit circle. Denote by  $y'$  the first point where this happens as we follow  $\mathcal{C}'$  starting at  $1/\nu$ .
- 2.A If  $y'$  is  $\delta$ -close to  $e^{i\theta_k} \simeq -1$  we are done. Indeed by Lemma 3.0, at points  $\delta$ -close to  $1/\nu$  the magnitude of  $P_1$  is dominating and at points  $\delta$ -close to  $e^{i\theta_k}$  the magnitude of  $P_2$  is dominating. If  $y'$  belongs to the latter, then somewhere between  $1/\nu$  and  $y'$  along  $\mathcal{C}'$  the magnitudes of  $P_1$  and  $P_2$  must be equal, defining a root of  $P'$  outside the unit disc.
- 2.B If  $y' = e^{i\phi}$  is not  $\delta$ -close to  $e^{i\theta_k}$ , then necessarily  $\phi < \theta_k$ , since  $\theta_k + \delta > \theta_k + \varepsilon + \pi$  is not in the upper half plane. We shall hence continuously move  $e^{i\theta_k}$  anti-clockwise.
- 3.B As  $\theta_k$  decreases,  $\mathcal{C}'$  deviates like stated in Corollary 3.3. More explicitly, we construct  $\mathcal{S}_{\theta_k}$  with as fixed boundary the positive real axis between  $y = 1$  and  $y = 1/\nu$  and the upper half unit circle. Initially the intersection of this fixed part of the boundary with  $\mathcal{C}'$ , the other part of the boundary, occurs at  $y' = e^{i\phi}$  with  $\phi < \theta_k$ . Note that by Lemma 3.1(d) the root locus curve remains confined to  $\text{Im}(y) \geq 0$  and by Lemma 3.1(a),(b) it must keep intersecting the unit circle. Hence Corollary 3.3 implies that when  $\theta_k$  decreases,  $\phi$  decreases (maybe non-strictly, maybe in jumps but never letting  $y'$  leave the upper half unit circle).
- 4.B With this behavior, it is inevitable that for some value  $e^{i\theta_k}$  catches up with  $y'$ , at latest when  $\theta_k = \delta$ . Then we are back in the situation of item 2.A above. So due to this possible value of  $\theta_k$ , the given  $P_2$  can provably not improve the situation with respect to Section 2.1. This concludes the proof.

## 4 Perspective

We have proved that the maximal acceleration of average consensus achievable by adding  $M - 1$  memory slots to each agent, is already achieved with  $M - 1 = 1$ . More precisely, this holds true when examining the worst possible performance for a constant network about which we only know bounds  $[\underline{\lambda}, \bar{\lambda}]$  on the eigenvalues of its Laplacian matrix.

The consensus context just incidentally provides a motivation for the following more general setting. Consider a general linear algorithm

$$x(t+1) = s + G(x(t) - s) \quad (8)$$

where  $s \in \mathbb{R}^N$  is the unknown solution that we search with  $x \in \mathbb{R}^N$ , and  $G$  is some constant  $N \times N$  matrix with *real* eigenvalues  $\gamma_k$ . Just knowing that  $\gamma_k \in [\underline{\gamma}, \bar{\gamma}]$  for all  $k$ , we can accelerate the convergence per iteration by moving by a weighted step  $\alpha$ ,

$$x(t+1) = x(t) + \alpha(s + G(x(t) - s) - x(t)). \quad (9)$$

For each mode  $k$ , we get  $\tilde{x}_k(t+1) = \tilde{\gamma}_k \tilde{x}_k(t) + (1 - \tilde{\gamma}_k) \tilde{s}_k$  where  $\tilde{\gamma}_k = 1 - \alpha + \alpha\gamma_k$ , and we should choose  $\alpha$  to have  $\min_k[\tilde{\gamma}_k] = -\max_k[\tilde{\gamma}_k]$  according to the known bounds  $[\underline{\gamma}, \bar{\gamma}]$ . From there, we can further accelerate convergence by using a simple memory slot,

$$x(t+1) = x(t) + \alpha(s + G(x(t) - s) - x(t)) + \beta(x(t-1) - x(t)). \quad (10)$$

This is not unlike techniques used in the optimization algorithms literature. The result of Section 2.1, with adapted notations, gives the optimal parameter choice. Our Theorem 1 implies that no further acceleration is possible by adding more memory slots.

A direct extension of our result would let each subsystem follow general linear dynamics, not restricted to memory slots. It is in fact not clear if anything can be gained in that way.

The basic consensus algorithm (1) has a proven robustness to network incidents<sup>3</sup>. For the case with memory slot, one can actually construct examples where packet drops lead to instability. Thus the benefit of more than two memory slots might be reevaluated towards robustness.

The optimal acceleration of the dual Markov chains through “liftings” has also been extensively characterized in the literature, see e.g. [13, 14]. However, those approaches (i) exploit knowledge of the particular network to build the accelerating scheme, while (ii) the allowed dynamics is restricted to positive systems (see Remark 1). Hence a possible link with our result remains to be investigated.

## References

- [1] J. N. Tsitsiklis, “Problems in decentralized decision making and computation,” *PhD Thesis, MIT*, 1984.

---

<sup>3</sup>Stability under switching networks holds thanks to positivity for any directed graphs with positive  $I - \alpha L$  [1], and thanks to the common Lyapunov function  $\sum_i (x_i)^2$  for any undirected graphs as long as  $I - \alpha L$  has eigenvalues inside the unit circle. The additional conditions always hold if we start from a stable network and “packet drops” occur.



- [2] S. Muthukrishnan, B. Ghosh, and M. H. Schultz, “First- and second-order diffusive methods for rapid, coarse, distributed load balancing,” *Theory Comput. Systems*, vol. 31, pp. 331–354, 1998.
- [3] B. Johansson and M. Johansson, “Faster linear iterations for distributed averaging,” *Proceedings of the 17th IFAC World Congress*, pp. 2861–2866, 2008.
- [4] E. Ghadimi, I. Shames, and M. Johansson, “Accelerated gradient methods for networked optimization,” *arxiv*, vol. 1211.2132, 2012.
- [5] J.Liu, B. Anderson, M.Cao, and S. Morse, “Analysis of accelerated gossip algorithms,” *Automatica*, vol. 49, pp. 873–883, 2013.
- [6] C. V. Hollot and A. C. Bartlett, “Some discrete-time counterparts to Kharitonov’s stability criterion for uncertain systems,” *IEEE Trans. Automatic Control*, vol. 31, no. 4, pp. 355–356, 1986.
- [7] P. Vaidyanathan, “Derivation of new and existing discrete-time Kharitonov theorems based on discrete-time reactances,” *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 38, no. 2, pp. 277–285, 1990.
- [8] E. Montijano, J. Montijano, and C. Saúés, “Chebyshev polynomials in distributed consensus applications,” *IEEE Trans. Signal Proc.*, vol. 61, no. 3, pp. 693–706, 2013.
- [9] E.Kokiopoulou and P. Frossard, “Polynomial filtering for fast convergence in distributed consensus,” *arxiv*, vol. 0802.3992, 2008.
- [10] S. Sundaram and C. Hadjicostis, “Finite-time distributed consensus in graphs with time-invariant topologies,” *Proc. American Control Conference*, pp. 711–716, 2007.
- [11] W. R. Evans, “Control system synthesis by root locus method,” *American Institute of Electrical Engineers, Transactions of the*, vol. 69, no. 1, pp. 66–69, 1950.
- [12] M. Srivastava, M. Srivastava, and S. Bhatnagar, *Control Systems*. Tata MacGraw-Hill, 2009.
- [13] F. Chen, L. Lovasz, and I. Pak, “Lifting Markov chains to speed up mixing,” *Proc.STOC’99*, 1999.
- [14] K. Jung, D.Shah, and J.Shin, “Fast and slim lifted markov chains,” *Proc. Allerton Conference on Communication, Control and Computing*, 2007.

## Acknowledgements

This research has been carried out within the Interuniversity Attraction Poles network DYSCO. The author wants to thank F.Ticozzi and S.Zampieri for stimulating discussions.