Accelerating consensus by spectral clustering and polynomial filters

Simon Apers, Alain Sarlette

Accelerating Consensus by Spectral Clustering and Polynomial Filters

Simon Apers and Alain Sarlette

Abstract—It is known that polynomial filtering can accelerate the convergence towards average consensus on an undirected network. In this paper the gain of a second-order filtering is investigated in more detail. A set of graphs is determined for which consensus can be attained in finite time, and a preconditioner is proposed to adapt the undirected weights of any given graph to achieve fastest convergence with the polynomial filter. The corresponding cost function differs from the traditional spectral gap, as it favors grouping the eigenvalues in two clusters and can favor symmetry breaking. A possible loss of robustness of the polynomial filter is also highlighted.

I. CONSENSUS ACCELERATION

SINCE their introduction in [1], (discrete-time) consensus algorithms have attracted almost as much attention as their dual, fast mixing Markov chains [2], [3]. Improving the convergence speed of this basic building block for e.g., distributed computation [4], [5], Kalman filtering [6], [7] or control of distributed systems [8]–[10] has been a major focus, whose results cannot be comprehensively reviewed here. A few existing approaches are mentioned below, after introducing some basic definitions to facilitate an explicit discussion. For synchronized fixed networks, some particular acceleration methods include: optimizing the weights on the links [2], [11], adding local memory [12], or introducing time-varying filters [13], [14]. The purpose of the present paper is to study where and how the polynomial filter [13] can be helpful, in particular in the novel context of combining it with optimization of link weights. The analysis focuses on spectral properties, to presumably facilitate integration of the insights into more general linear dynamical networks, and show explicit connections to spectral graph theory as treated in e.g., [15]. Before detailing the related state of knowledge as well as our contributions, let us introduce the basic setting.

The consensus setting considers an undirected and connected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with N nodes $\in \mathcal{V}$ and m edges $\in \mathcal{E}$. Denote the node states as $x = (x_1, x_2, ..., x_N) \in \mathbb{R}^N$. The basic linear consensus dynamics on $\mathcal{G}$ is

$$x_i(k+1) = x_i(k) + w \sum_{(i,j) \in \mathcal{E}} L_{ij}(x_j(k) - x_i(k)) \quad (1)$$

with $k \in \mathbb{N}$, $w \in \mathbb{R}$ some gain and L a symmetric matrix of edge weights called the Laplacian. We can rewrite (1) as

$$x(k+1) = (I - wL) x(k) =: P_w x(k),$$

S. Apers is with the SYSTeMS group, Ghent University, Belgium; e-mail: simon.apers@ugent.be

A. Sarlette is with the QUANTIC project team, INRIA Rocquencourt, France and the SYSTeMS group, Ghent University, Belgium; e-mail: alain.sarlette@inria.fr

with I the identity matrix. *Consensus* is the stationary state satisfying

$$x_1 = x_2 = ... = x_N =: c \quad \text{or equivalently} \quad L x = 0$$

(if the graph is connected). *Average consensus* requires the consensus value c to satisfy $c = \frac{1}{N} \sum_{i=1}^{N} x_i(0)$. The convergence speed of (1) towards consensus is governed by P_w's second[1] largest eigenvalue modulus (SLEM)

$$\mu(P_w) = \max\{|x^T P_w x| : x^T x \leq 1, \sum_{k=1}^{N} x_k = 0\}$$

or the "spectral gap" $1 - \mu(P_w)$. Note that if the graph is disconnected, $\mu(P_w) = 1$ and consensus cannot be reached. For a given connected graph, the fastest convergence is obtained by choosing the edge weights to maximize the spectral gap. This is a convex problem [2], [11], whose solution we call the Fastest Single-Step Convergence network (FSSC)[2]. In case of a fixed L, with knowledge of bounds on its eigenvalues $\underline{\lambda} \leq \{x^T L x : x^T x = 1, \sum_{k=1}^{N} x_k = 0\} \leq \bar{\lambda}$, the optimal $\mu(P_w)$ is obtained by choosing $w = w_c$, with w_c selected such that $(1 - w_c \underline{\lambda}) = -(1 - w_c \bar{\lambda})$ [and thus $= \mu(P_{w_c})$]. We call the corresponding P_{w_c} *centered*. In all the following algorithms, L is always multiplied by a scalar design parameter. We can thus without loss of generality, by possibly rescaling this parameter, assume that $\underline{\lambda} + \bar{\lambda} = 2$. This implies $w_c = 1$, i.e., $P := I - L$ is *centered* such that $(1 - \underline{\lambda}) = -(1 - \bar{\lambda}) = \mu(P)$. This assumption allows us to simplify further discussion.

Accelerated consensus denotes the expansion of node and/or communication features, under the same graph constraint, to improve convergence speed. These algorithm modifications go further than the weight optimization methods proposed by [2], [11]. We next highlight some proposed acceleration methods in order to position our work. For a more comprehensive overview see e.g., the introduction of [16].

A first acceleration approach consists of adding "momentum" to the iteration steps. *Memory slots* are added at each node to get

$$x(k+1) = (I - w_1 L)x(k) + w_2(x(k-1) - x(k)) \quad (2)$$

---

[1]This holds if w is not chosen too large, such that the largest eigenvalue modulus of P_w is 1, with the consensus state as eigenvector. It is standard to assume that this condition is satisfied, and the constructions made throughout the paper ensure that we follow this convention. Else, P_w would have an eigenvalue modulus larger than 1 and be unstable.

[2]In [2], Boyd et al. demand the weights to be positive, $[P_w]_{i,j} > 0$ for all i, j. This constraint favors robustness to network changes. In the present paper we drop it, as Boyd et al. do in [11], for a fairer comparison with other algorithms whose robustness is not established.

with some fixed $w_1, w_2 \in \mathbb{R}$. A similar approach is presented in a.o. [12], [17], optimizing the scheme with limited knowledge of the graph (spectrum). Extending the memory registers to $x(k - M)$ with $M > 1$ has been shown not to improve convergence speed, if the knowledge about the interaction graph reduces to bounds on its Laplacian's extreme eigenvalues, $\underline{\lambda} \leq \{x^T L x \; : \; x^T x = 1, \sum_{k=1}^N x_k = 0\} \leq \bar{\lambda}$ [18]. Based on different knowledge of the graph, namely an upper bound on the number of nodes in the graph, [16] proposes a consensus scheme similar to (2) and whose convergence time is linear in this upper bound. Also related to this approach are *lifted Markov chains*, as proposed by [19] and [3]. These however build on more detailed knowledge of the graph than bounds on its Laplacian's extreme eigenvalues.

Another approach is *polynomial filtering*, where a time-varying choice of w in (1) accelerates convergence. The idea is that if L has eigenvectors $\tilde{x}_i$ with eigenvalues λ_i, then after r steps of (1) with $w(k)$ time-dependent, each $\tilde{x}_i$ has been multiplied by $p_r(\lambda_i) = \Pi_{k=1}^r (1 - w(k)\lambda_i)$. This p_r can be made equal to any polynomial of order r satisfying $p_r(0) = 1$, by selecting appropriate $w(k)$; whereas constant w restricts to $p_r(\lambda_i) = (1 - w\lambda_i)^r$. In particular, choosing $w(k) = 1/\lambda_k$ would imply $p_r(\lambda_i) = 0$ for all $i < r$, hence finite-time convergence in at most $N - 1$ steps. A similar approach, and its extension allowing the individual entries of the L matrix to change independently with time, is presented in e.g., [14], [20]–[24], and is related to the field of *finite-time consensus*. We note that towards optimal tuning, varying individual entries in L is mostly deemed intractable, such that most examples in the papers actually involve polynomial filtering. A finite-time convergence result similar to polynomial filtering can be obtained by combining N local memory slots with a time-varying protocol, see e.g. [25], [26]. This however requires not only to implement a high-order polynomial filter or large node memory, but also to know the eigenvalues of L exactly. The (optimal polynomial related to the) latter can be evaluated online, assuming larger node and computation capabilities at the nodes [25], [26]; this goes into the field of network identification and involves entirely different resources. Therefore, practical considerations such as robustness and memory requirements have led to two related, asymptotically converging methods. In the case where the order of the polynomial is limited ($r < N$) while the full spectrum is known, [27] constructs a semi-definite program to construct the optimal polynomial. If knowledge on the spectrum is limited to bounds $\underline{\lambda}, \bar{\lambda}$ such that $\underline{\lambda} \leq \{x^T L x \; : \; x^T x = 1, \sum_{k=1}^N x_k = 0\} \leq \bar{\lambda}$, then [13] proves the closed-form expression for the optimal polynomial p_r of given order r. Furthermore, they provide a scheme which at any time t efficiently applies the optimal polynomial filter p_t. In the present paper, we consider the algorithm that recursively applies a polynomial p_r of fixed (small) order r, and we investigate how a little more knowledge than $\underline{\lambda}, \bar{\lambda}$ about the eigenvalues of L allows to accelerate convergence.

Besides these two approaches, more extensive consensus acceleration protocols have been proposed and can provide faster convergence than polynomial filtering in a similar setting, for example by estimating spectral properties online as in [25], [26]. These however generally require more complicated local online computations, and more importantly, the communication of multiple values per edge at each time step. This contrasts with polynomial filtering and local memory-based acceleration, which only require a linear local computation and communication of a single scalar value per time step along each edge. The more suitable algorithm type depends largely on the considered application framework. Our interest here in polynomial filtering and local memory-based acceleration can be motivated by anticipating the possible easy integration of such consensus routines into linear networked systems with more complex local dynamics.

Our contribution in the present paper concerns a more in-depth analysis of polynomial filtering, in particular proposing an associated optimal tuning of Laplacian weights which appears to deliver nontrivial tuning results.

Comparing polynomial filtering to the "momentum" strategy yields the following observation, which to our knowledge is new. Let $m_k(\lambda, w_1, w_2)$, $k = 1, 2$ denote the two poles of the linear system obtained by replacing L with $\lambda \in \mathbb{R}$ in (2); and let $f(\lambda, r, \{w(k)\}) = |p_r(\lambda)|^{1/r}$. Thus $m(\lambda, w_1, w_2) := \max\{|m_1(\lambda, w_1, w_2)|, |m_2(\lambda, w_1, w_2)|\}$ and $f(\lambda, r, \{w(k)\})$ characterize the asymptotic convergence speed per time step of respectively the "local memory momentum" algorithm, and the algorithm that repeatedly iterates the order r polynomial filter, for a network mode corresponding to eigenvalue λ of L.

Proposition 1. *If only bounds on the extreme nonzero eigenvalues of L are known, then the optimal (2) has a faster asymptotic convergence rate than repeating the optimal polynomial filter of arbitrary order $r \in \mathbb{N}$, i.e.:*

$$\min_{w_1, w_2} \left(\max_{\lambda \in [\underline{\lambda}, \bar{\lambda}]} (m(\lambda, w_1, w_2)) \right)$$
$$\leq \min_{\{w(k): k=1,2,\ldots,r\}} \left(\max_{\lambda \in [\underline{\lambda}, \bar{\lambda}]} (f(\lambda, r, \{w(k)\})) \right) .$$

Proof. We recall that to simplify the discussion we assume L scaled such that $\underline{\lambda} + \bar{\lambda} = 2$; as Laplacian eigenvalues are nonnegative this readily implies $\underline{\lambda} \in (0, 1)$. The paper [13] explicitly computes the optimal order-r polynomial filter and its worst-case convergence rate over $\lambda \in [\underline{\lambda}, \bar{\lambda}]$. Those are obtained with $w(k) = \overline{w}(k)$ selected such that $p_r(\lambda)$ is proportional to $T_r(1 - \lambda)$, with T_r the r-order Chebyshev polynomial, and yields

$$\max_{\lambda \in [\underline{\lambda}, \bar{\lambda}]} (f(\lambda, r, \{\overline{w}(k)\})) = 1/\left| T_r(\tfrac{1}{1-\underline{\lambda}}) \right|^{1/r} .$$

The optimal values $(\overline{w}_1, \overline{w}_2)$ of (w_1, w_2) in (2) are given in e.g., [12], [18], with associated asymptotic convergence rate

$$\max_{\lambda \in [\underline{\lambda}, \bar{\lambda}]} (m(\lambda, \overline{w}_1, \overline{w}_2)) = \frac{1}{1 - \underline{\lambda}} - \sqrt{\frac{1}{(1-\underline{\lambda})^2} - 1} . \quad (3)$$

A function plot readily shows that

$$1/\left| T_r(\tfrac{1}{1-\underline{\lambda}}) \right|^{1/r} \geq \frac{1}{1 - \underline{\lambda}} - \sqrt{\frac{1}{(1-\underline{\lambda})^2} - 1} \quad (4)$$

for all positive integers r and for all $\underline{\lambda} \in [0, 1]$. $\square$

Whether synchronous variation of $w(k)$ or a local memory slot at each node is a more demanding resource, is application dependent. A single scalar memory at each node should mostly not be too demanding, and in this case Proposition 1 shows that (2) is superior to the fixed-order polynomial filter, at least when L can have eigenvalues anywhere in $[\underline{\lambda}, \bar{\lambda}]$, as analyzed in [13]. At best, when r grows with t as proposed in [13], the inequality in (4) tends to an equality[3] However, when more is known about the spectrum of L, the polynomial filter can start beating the "momentum" memory strategy (2) significantly. This is exemplified above by the extreme case where $w(k) = 1/\lambda_k$, implying finite-time convergence, whereas (2) can only converge asymptotically as soon as $\underline{\lambda} \neq \bar{\lambda}$.

Therefore, the purpose of the present paper is to extend [13], [27] by analyzing how much more can be gained by the polynomial filter when (somewhat) more is known about L than the eigenvalue bounds $[\underline{\lambda}, \bar{\lambda}]$. We restrict our investigation to a two-step alternating scheme:

$$x(k+2) = (I\text{-}w_2 L)(I\text{-}w_1 L)\, x(k) \,, \tag{5}$$

performing second-order polynomial filtering on the spectrum of L. Despite this restriction, we are able to highlight several interesting acceleration features. First, we compute how additional knowledge on L influences the optimal polynomial, i.e., optimal values for w_1 and w_2 in (5). In particular, we show that if somewhat more than the bounds $[\underline{\lambda}, \bar{\lambda}]$ are known, polynomial filtering can in fact beat the memory slot algorithm (2), unlike in the situation of Proposition 1. Secondly, we examine how polynomial acceleration can benefit from optimizing the weights of the graph edges, as was done for standard consensus in [2], [11]. This merger of two acceleration techniques leads to a non-convex optimization problem and can favor symmetry-breaking. We also prove (see Appendix A) that polynomial filtering is an actual restriction with respect to allowing the entries of L to vary independently of each other.

The remainder of the paper is organized as follows. In Section II we compute the optimal second-order polynomial and the associated stability and gain in convergence speed. In Section III we consider the optimization of graph weights towards polynomial filtering. We also mention some special graphs that allow 2-step consensus with possible symmetry-breaking. Section IV discusses the practical implementation of the acceleration scheme and its robustness properties.

II. Optimizing the Polynomial Filter

We now investigate the optimal polynomial filter (5) for a given graph Laplacian L, about which we possibly know more than just the spectral gap. We recall that we assume, without loss of generality, that L is scaled such that $P = I - L$ is centered. We first show that under this condition, the optimal polynomial is easily formulated and only depends on bounds on P's SLEM $\mu(P)$ and on its *Smallest Eigenvalue Modulus*.

[3]Then since the system (2) has a double eigenvalue, the scheme of [13] converges in fact faster, but only by a non-exponential factor.

Definition 1. *The* Smallest Eigenvalue Modulus $\sigma(P)$ *(SEM) of a symmetric matrix P is defined as*

$$\sigma(P) := \min\{|x^T P x| \,:\, x^T x \geq 1\}.$$

In words, the eigenvalues of P are guaranteed to belong to $[-\mu(P), -\sigma(P)] \cup [\sigma(P), \mu(P)] \cup \{1\}$. For future reference, we denote by $\{\lambda_i \,:\, i = 1, 2, ..., N\}$ the eigenvalues of P, with $\lambda_1 = 1$ corresponding to the consensus eigenspace. Then $\{\lambda_2, \lambda_3, ..., \lambda_N\} \in [-\mu(P), -\sigma(P)] \cup [\sigma(P), \mu(P)]$, provided the graph is connected.

Towards the optimal polynomial, we start by defining $z_i = 1 - 1/w_i$, for $i = 1, 2$ such that (5) can be rewritten

$$x(k+2) = \left(I - \frac{1}{1-z_1}L\right)\left(I - \frac{1}{1-z_2}L\right) \tag{6}$$

$$= \frac{(P - z_1 I)(P - z_2 I)}{(1-z_1)(1-z_2)}\, x(k) =: p_2(P)\, x(k). \tag{7}$$

We can thus view the polynomial filter as modifying the convergence rate not as a function of L but as a function of a given P; we recall that we assume P to be centered, i.e. $\lambda_2 = -\lambda_N = \mu(P)$. The filter p_2 parameterized by z_1, z_2 can be any second-order polynomial restricted to $p_2(1) = 1$. Through (6), the eigenvector of P associated to λ_i will be multiplied by $p_2(\lambda_i)$ over two time steps. Hence we define the optimal exponential convergence rate as

$$\mu_2(P) \triangleq \min_{z_1, z_2}\left[\max_{i>1} |p_2(\lambda_i)|\right].$$

Theorem 1. *Consider a connected, undirected graph $\mathcal{G}$ with given centered weight matrix P. The optimal convergence rate attainable through p_2-acceleration (6) is given by*

$$\mu_2(P) = \frac{\mu(P)^2 - \sigma(P)^2}{2 - \mu(P)^2 - \sigma(P)^2} \tag{8}$$

and obtained with the unique polynomial

$$p_2^*(P) \triangleq \frac{P^2 - \mu(P)^2/2 - \sigma(P)^2/2}{1 - \mu(P)^2/2 - \sigma(P)^2/2}. \tag{9}$$

Proof. For any given $[|a|, |b|] \subseteq [0, 1]$, define the second-order polynomial $p_2^*(x)$ satisfying $p_2^*(1) = 1$ and minimizing $\max_{x \in \{-b, -a, a, b\}} |p_2^*(x)|$. A rapid function analysis shows that it is determined by $p_2^*(b) = p_2^*(-b) = -p_2^*(a) = -p_2^*(-a)$ (see Fig. 1). Replacing $|a|, |b|$ by the particular eigenvalues $\sigma(P), \mu(P)$ of P, we get

$$\max_{\lambda \in \{-\mu, -\sigma, \sigma, \mu\}} |p_2(\lambda)| \geq \max_{\lambda \in \{-\mu, -\sigma, \sigma, \mu\}} |p_2^*(\lambda)|.$$

Noting furthermore that $|p_2^*(x)| \leq |p_2^*(b)|$ for any $|x| \in [|a|, |b|]$, we actually get

$$\max_{i>1} |p_2(\lambda_i)| \geq \max_{i>1} |p_2^*(\lambda_i)| = |p_2^*(\mu(P))|,$$

independently of the other eigenvalues of P. $\square$

The convergence rate of the polynomial filter over $2t$ time steps is $\mu_2(P)^t$. The corresponding unaccelerated consensus algorithm would converge as $\mu(P)^{2t}$. The resulting improvement by optimal p_2-acceleration is illustrated on Fig.1.

Corollary 1. *The 2nd-order polynomial acceleration scheme (5) with optimal w_1, w_2 always outperforms standard consensus (1) with optimal w, i.e., $\mu_2(P) < \mu(P)^2$, even if no bound is known on the SEM of P (i.e., assuming $\sigma(P) = 0$). Furthermore, for any $\mu \in (0,1)$, the optimal polynomial (6) is faster than the optimal memory slot scheme (2) if*

$$\sigma \in (f(\mu), \mu] \quad \text{with} \quad f(\mu) = \sqrt{1 - \sqrt{1 - \mu^2}}\,.$$

Proof. The first part follows immediately from the fact that

$$\mu_2 = \frac{\mu^2 - \sigma^2}{2 - \mu^2 - \sigma^2} \leq \frac{\mu^2}{2 - \mu^2} < \mu^2$$

since $\mu^2 < 1$ and $2 - \mu^2 > 1$ for $\mu < 1$.

The second part comes from the fact that for given μ, the additional knowledge of σ would not allow to improve the optimal memory slot dynamics, *if the scheme is restricted to a single memory*. This result is a straightforward adaptation of Proposition 5 in [18] to the setting of knowing σ, and is mentioned in its Section 4. The optimal memory scheme centers P (i.e. $\mu = 1 - \underline{\lambda}$, see [12], [18]) and has convergence rate as presented in equation (3). Comparing this result to Theorem 1, the statement of the Corollary would be:

$$0 \leq \sqrt{\frac{\mu^2 - \sigma^2}{2 - \mu^2 - \sigma^2}} \leq \frac{1}{\mu} - \sqrt{\frac{1}{\mu^2} - 1}, \ \ \text{for } \sigma \in [f(\mu), \mu].$$

The left hand side tends to zero when σ tends to μ (unless $\mu = 1$) whereas the right hand side does not (unless $\mu = 0$). Thus for each $\mu \in (0,1)$ there must exist some $f(\mu) < \mu$ such that the inequality holds. A few algebraic computations lead to the announced expression of $f(\mu)$. $\qquad\square$
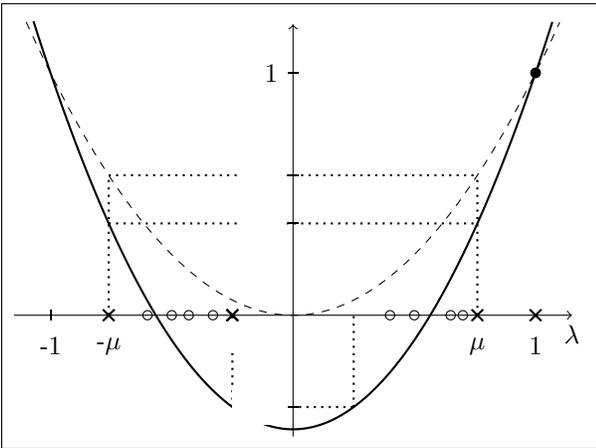

Figure 1: Comparison of SLEM of P^2 (applying 2 steps of standard consensus: μ^2) and SLEM of $p_2^*(P)$ (applying the 2-step procedure (6) one time: μ_2), for some arbitrary P. The critical eigenvalues, determining the convergence rate, are distinguished ($\times$) from the other ones ($\circ$). The polynomials $y = \lambda^2$ (dotted lines, standard consensus) and $y = p_2^*(\lambda)$ (plain line, polynomial filter) illustrate the acceleration mechanism.

A. Robust stability with respect to estimates $\widetilde{\mu}$ and $\widetilde{\sigma}$

If μ and σ are not known exactly, a safe strategy is to tune the polynomial filter on the basis of an upper bound on μ and a lower bound on σ. This might however be overly conservative,

and if a small probability of deceleration is tolerated one might want to tune the polynomial filter with estimates $\widetilde{\mu}$ and $\widetilde{\sigma}$ which are not guaranteed bounds on μ and σ. In the following we consider our scheme in the case of such nonstrict estimates. This is in some sense the converse of Theorem 1 (given a polynomial, which P are admissible?).

Proposition 2. *Consider the dynamics (6) with*

$$p_2(P) = \frac{P^2 - \widetilde{\mu}^2/2 - \widetilde{\sigma}^2/2}{1 - \widetilde{\mu}^2/2 - \widetilde{\sigma}^2/2}$$

for some $\widetilde{\sigma} \leq \widetilde{\mu} < 1$.
(a) The system is stable if and only if all the eigenvalues of P belong to $[-1, -f] \cup [f, 1]$, i.e., $\mu(P) \leq 1$ and $\sigma(P) \geq f$, where $f = \sqrt{\widetilde{\mu}^2 + \widetilde{\sigma}^2 - 1}$. In particular if $\widetilde{\mu}^2 + \widetilde{\sigma}^2 < 1$ then the system is stable if and only if $\mu(P) \leq 1$.
(b) The system converges faster towards consensus asymptotically than the system $x(k+1) = P\, x(k)$, provided $\mu(P) \leq 1$ and $\sigma(P) \geq \widetilde{f} = \sqrt{\widetilde{\mu}^2 + \widetilde{\sigma}^2 - 1 + (1 - \mu^2)\left(1 - \frac{\widetilde{\mu}^2}{2} - \frac{\widetilde{\sigma}^2}{2}\right)}$.

Proof. Writing the conditions for having $|p_2(\lambda_i)| \leq 1$ (resp. $|p_2(\lambda_i)| \leq \mu^2$) directly yields (a) (resp. (b)). $\qquad\square$

From Corollary 1 and Proposition 2, it follows that tuning (6) with $\widetilde{\mu}^2 + \widetilde{\sigma}^2 < 1$ allows to combine (suboptimal) polynomial speedup while assuring stability even if the central eigenvalue gap vanishes ($\sigma(P) = 0$). The latter case is the one analyzed in [13]. This property will come back when analyzing robustness to link failure (see Section IV-B).

III. SPECTRAL CLUSTERING

In light of the above results, a given weighted graph $\mathcal{G}$ will allow substantial acceleration through second-order filtering if its Laplacian eigenvalues are clustered in two sets whose width is small compared to the distance between the sets.

This observation motivates a new way of optimizing the weights on network links of an initially unweighted graph $\mathcal{G}$. Therefore in the following, we assume the sets $\mathcal{V}, \mathcal{E}$ of nodes and edges of $\mathcal{G}$ are given, but we are free to tune the weight assigned to each edge of $\mathcal{E}$. Explicitly, the system designer would solve the following optimization problem (O2):

given $\quad\quad \mathcal{G}(\mathcal{V}, \mathcal{E})$

minimize$_{P, z_1, z_2}$ $\quad \max_{i>1} |p_2(\lambda_i)|$

subject to $\quad 1 = \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N$ eigenvalues of P,
$\quad\quad\quad\quad\quad P \in \mathbb{R}^{N \times N}, P = P^T, \sum_{i=1}^{N} P(i,j) = 1,$
$\quad\quad\quad\quad\quad P(i,j) = 0$ if $(i,j) \notin \mathcal{E}$.

We call *compatible* with $\mathcal{G}(\mathcal{V}, \mathcal{E})$ a matrix P satisfying the conditions of O2. This setting and related conclusions easily generalize at least conceptually to r-order polynomial filters with $r > 2$: grouping the eigenvalues of P into r tight clusters would favor fast convergence. Explicit forms generalizing (8),(9) to $r > 2$ are however more difficult to obtain.

Before turning to an investigation of O2 in general, we briefly consider special cases where *finite-time convergence* to consensus is achieved in two steps. Although this only covers a few graphs, the gained insights connect with the existing literature on finite-time consensus, see e.g., [14], [20]–[24].

A. Finite-time consensus and symmetry breaking

Obviously, if a P matrix in our framework has only two distinct eigenvalues $\lambda_2 = -\lambda_3$ besides the invariant space with $\lambda_1 = 1$, then $\sigma(P) = \mu(P)$ and (8),(9) imply perfect consensus after one application of (5). Conversely, if P has more than two distinct eigenvalues besides $\lambda = 1$ then (6) can only induce asymptotic convergence.

In e.g., [14], some graphs $\mathcal{G}$ are characterized for which two compatible update matrices P_1, P_2 can be selected such that already after two steps $x(2) = P_2 P_1 x(0)$ is a consensus state. Pursuing such two-step convergence in a polynomial filter, i.e., through O2, essentially solves the same optimization problem with the additional restriction that there must exist a $c \in \mathbb{R}$ such that $P_2 = cI + (1-c)P_1$. The following properties illustrate the similarities and differences regarding finite-time consensus with this additional restriction.

Proposition 3. *Denote by $d(\mathcal{G})$ the diameter of a graph $\mathcal{G}$. (a) Having $d(\mathcal{G}) \leq 2$ is necessary but not sufficient to allow two-step convergence, i.e., reaching a minimum 0 as the solution of O2. If $\mathcal{G}$ is distance-regular, then $d(\mathcal{G}) \leq 2$ is sufficient for two-step convergence.*
(b) There are graphs $\mathcal{G}$ for which there exist compatible P_2, P_1 such that $x(2) = P_2 P_1 x(0)$ is a consensus state, but for which the solution of O2 is strictly larger than 0 i.e., the additional restriction $P_2 = cI + dP_1$ forbids finite-time consensus.

Proof. Property (a) carries over from [14], the restriction $P_2 = cI + dP_1$ does not affect the associated proofs. Property (b) is proved by a counterexample in Appendix A. $\quad\square$

We have investigated small graphs exhaustively. The condition $d(\mathcal{G}) \leq 2$ turns out to be sufficient for $N \leq 5$ at least, showing that 5 out of the 6 four-node graphs (resp. 15 out of 21 five-node graphs) can converge in two steps with p_2-acceleration. The counterexample proving Proposition 3 is a 6-node graph. Lists of graphs whose Laplacians have only two different nonzero eigenvalues can be found in the literature, see e.g., [28, Table 14.2, Table 14.4, Chapter 15.2], assuming uniform weights, and [15]. All these graphs allow finite-time convergence with the polynomial filter; while for (1) only the complete graph converges in finite time.

The complete bipartite graphs $K(\ell, m)$ between sets of ℓ and m nodes allow to illustrate interesting finite-time convergence properties. Both $\mathcal{K}(m, m)$ and the star graph $\mathcal{K}(1, m)$ with uniform link weights allow two-step convergence using p_2^*, although only $\mathcal{K}(m, m)$ is distance-regular. For the other $\mathcal{K}(\ell, m)$ cases, uniform weights lead to 3 distinct nonzero eigenvalues in the Laplacian. Regarding μ, i.e., convergence rate with (1), the choice of uniform weights is optimal, as a consequence of symmetry preservation, see [2], [29]. Regarding μ_2 however, i.e., convergence with (5), a nonuniform weight selection might further accelerate the convergence. In particular, a *symmetry breaking* on $\mathcal{K}(2, m)$ does allow finite-time convergence. Indeed, with asymmetric weights $p \neq q$ (see Figure 2), the nonzero eigenvalues of L all belong to the set:

$$\left\{ p+q, \quad \frac{(m+1)(p+q)}{2} \pm \frac{\sqrt{(m+1)^2(p+q)^2 - 4pqm(m+2)}}{2} \right\}.$$

By choosing $q = \frac{1}{2}(m \pm \sqrt{m^2 - 4})p$, this set reduces to two distinct values, such that the optimal polynomial filter will yield $\mu_2(P) = 0$ i.e., finite-time convergence. *This possible benefit of symmetry breaking contrasts with standard consensus* (1)*, for which [29] shows that keeping the edge-transitivity symmetry in the weights does lead to the fastest convergence (FSSC).*
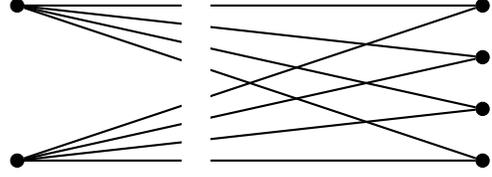

Figure 2: Complete bipartite graph $\mathcal{K}(2, 4)$ with weights p and q.

B. Preconditioning general graphs

This section treats the optimization of the edge weights in compliance with a given graph connectivity, which we call *preconditioning* the graph towards p_2-acceleration. Indeed, Theorem 1 gives an explicit expression for the optimal z_1, z_2 and the associated expression of $\mu_2(P) = \max_{i>1} |p_2(\lambda_i)|$ as a function of $\mu(P)$ and $\sigma(P)$ only. Therefore towards solving O2, we can concentrate on optimizing P. This is in the same spirit as the FSSC, but optimal *polynomial* acceleration favors an eigenvalue spectrum in distinct *clusters*. For a given $\mathcal{G}(\mathcal{V}, \mathcal{E})$, we denote by P^* the FSSC i.e., the optimal choice of weights for (1), and by P_{p2}^* the optimal weights for p_2 acceleration, i.e., those leading to the optimal solution of O2.

Despite the explicit expression for optimal z_1, z_2 as a function of P, it turns out that finding the P_{p2}^* solving O2 is not an easy problem. We hence characterize some of its properties and illustrate its potential by numerical optimization.

We first get a bound on the possible acceleration.

Definition 2. *Given a graph $\mathcal{G}$, denote by $\mathcal{G}^2$ its square graph, in which two nodes are linked by an edge if and only if in $\mathcal{G}$ they are linked by a path of ≤ 2 edges. We denote an arbitrary weight matrix compatible with this power graph as $P_{\mathcal{G}^2}$.*

Proposition 4. *The fastest convergence rate attainable on $\mathcal{G}$ using p_2-acceleration is bounded by the convergence rate of the FSSC on the square graph $\mathcal{G}^2$, i.e.,*

$$\mu_2(P_{p_2}^*) \geq \mu(P_{\mathcal{G}^2}^*) .$$

There exist graphs for which this inequality is strict.

Proof. The FSSC on a graph $\mathcal{G}'(\mathcal{V}, \mathcal{E}')$ solves

$$\begin{aligned} \underset{P}{\text{minimize}} \quad & \max_{i>1} |\lambda_i| \\ \text{subject to} \quad & 1 = \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N \text{ eigenvalues of } P, \\ & P \text{ compatible with } \mathcal{G}'(\mathcal{V}, \mathcal{E}'). \end{aligned}$$

For any choice of z_1, z_2, the matrix $p_2(P)$ is compatible with $\mathcal{G}^2$. Hence the solution of O2 on $\mathcal{G}$ is included in the possible solutions of the FSSC on $\mathcal{G}^2$. If the diameter $d(\mathcal{G}) \leq 2$ then $\mathcal{G}^2$ is completely connected and $\mu(P_{\mathcal{G}^2}^*) = 0$. The second item of Proposition 3 however states that there exist graphs for which $d(\mathcal{G}) \leq 2$ does not imply $\mu_2(P_{p_2}^*) = 0$. $\quad\square$

C. Numerical optimization

The FSSC weights $P = P^*$ do not necessarily minimize μ_2, see e.g., the graph $\mathcal{K}(2, m)$ in Section III-A; thus an alternative optimization is required.

Expressed as a function of the individual edge weights, $\mu_2(P)$ as defined in Theorem 1 is not a convex function, unlike $\mu(P)$ for the FSSC. Given the convexity of $\mu(\cdot)$ as a function of its argument and inspired by Proposition 4, we can try to reformulate ○2 as minimizing $\mu(P_{\mathcal{G}^2})$ over all $P_{\mathcal{G}^2}$ which can be represented as $p_2(P)$. The resulting relaxation however is not expected to always improve convexity, as for interesting graphs $\mathcal{G}$ the set of all $p_2(P)$ can be a non-convex subset of the set of all $P_{\mathcal{G}^2}$ (see Proposition 7 in Appendix B). So to have a convex problem we would indeed need to relax the search space to some convex subset of all $P_{\mathcal{G}^2}$ compatible with $\mathcal{G}^2$. The solution of the relaxed problem would then have to be reprojected into the set of all $p_2(P)$. And for this, it is well-known that variations on matrix elements give little insight on the induced variations in eigenvalues.

Thus optimal preconditioning towards p_2-acceleration seems significantly more difficult numerically than FSSC, unless a better formulation is found. In the meantime, we have performed a numerical optimization by *gradient descent* on $\mu_2(P)$ as a function of the edge weights in P. We claim by no means that this is the best strategy in terms of complexity or results, at this point it is just a feasible method to evaluate the potential of preconditioning for p_2-acceleration. Since this formulation is non-convex, the method converges to a local minimum, not guaranteeing the global optimum. The bound $\mu(\mathcal{G}^2)$ from Proposition 4 can give an indication on the quality of the obtained P matrix.

Figure 3 shows the spectrum of an Erdős-Rényi graph with 20 nodes, 96 edges and diameter 2. Its weights have been optimized respectively for FSSC (P^*) and towards fastest polynomial filtering (i.e., P_{p2}^* solving ○2, hopefully). The difference in convergence rate when these different P matrices are used with an optimal second-order polynomial filter (9) is graphically striking, as the preconditioning brings $\mu_2(P^*) = 0.0626$ down to $\mu_2(P_{p2}^*) = 0.0088$. This is further to be compared to $\mu^2(P^*) = 0.1181$ for the FSSC without p_2 acceleration and to the lower bound $\mu(P_{\mathcal{G}^2}^*) = 0$.

Beneficial clustering of the eigenvalues around two polynomial zeros $\pm z$ as on Fig. 3 is not always possible. It typically deteriorates with decreasing number of edges per node, thus as the amount of degrees of freedom available for optimizing a constant number of eigenvalues decreases. That this is not a general rule follows from the perfect clustering and finite-time convergence of star graphs (section III-A). We observe the following behavior in simulations.

- The behavior of the preconditioner and related acceleration was examined on a large set of Erdős-Rényi graphs and of random geometric graphs (generated using random dots on a $[0, 1]^2$-square, with different neighborhood radii). For both types of graphs, a clear trend appears when examining the average results as a function of the density, i.e. the number of edges compared to the complete graph, as shown in figure 4. When less than
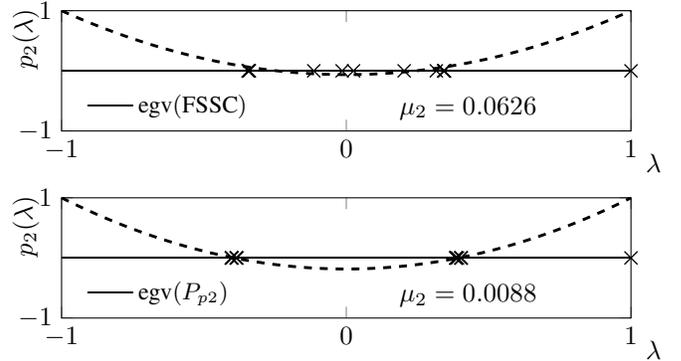

Figure 3: Spectral clustering performed on an Erdős-Rényi graph $\mathcal{G}(20, 96)$. The plot shows the spectrum of both the FSSC P^* and the optimized P_{p2}, as well as their optimal p_2-polynomials and corresponding μ_2 values.

about 30% of the edges are present w.r.t. the complete graph, the most significant acceleration is obtained by just taking the graph optimized for the FSSC, which is easily computable, and applying the optimal polynomial filter to it instead of a standard first-order consensus algorithm; trials to further adjust the weights towards faster convergence do not really pay off (in average). For densities higher than about 30 %, the preconditioner starts to significantly pay off, with gains up to orders of magnitude.

- This behavior and the previously mentioned lower bound of Proposition 4 might suggest that a sort of phase transition should appear as a function of the *diameter* of the graph. To investigate this we have partitioned the set of Erdős-Rényi graphs of a given density, into subsets as a function of their diameter. Except for the graphs with finite-time convergence at diameter 2, the acceleration ratio achieved by the preconditioner with respect to the FSSC appears to have *the same* distribution on these subsets. Moreover, the lower bound of Proposition 4 was reached on several graphs of diameter 3, 4 and 5. This seems to indicate that the graph diameter is not a directly limiting factor for the preconditioner. Surprisingly, it appears that the role of the graph diameter in possible convergence speed is still an open question in the literature also for the FSSC/FMMC problem.

- We have investigated by simulation the effect of graph size, i.e., from 20 up to 100 nodes, on Erdős-Rényi and random geometric graphs of fixed density. The preconditioning gain, defined as $(\mu_2(P_{p_2}^*) - \mu(P_{\mathcal{G}^2}^*))/(\mu_2(P^*) - \mu(P_{\mathcal{G}^2}^*))$, appears to stay constant on average.

- The optimal P matrix sometimes contains negative elements, also on off-diagonal entries. This indicates that *repulsion* between certain nodes can accelerate consensus with p_2. Such negative entries however can be undesirable for robustness, in which case one can easily exclude them in the optimization process.

- Unsurprisingly, a spectrum well clustered with the preconditioner is often highly degenerate in $\pm\mu$ and $\pm\sigma$. This is similar to the degeneracy found at $\pm\mu$ in the spectrum of the FSSC.
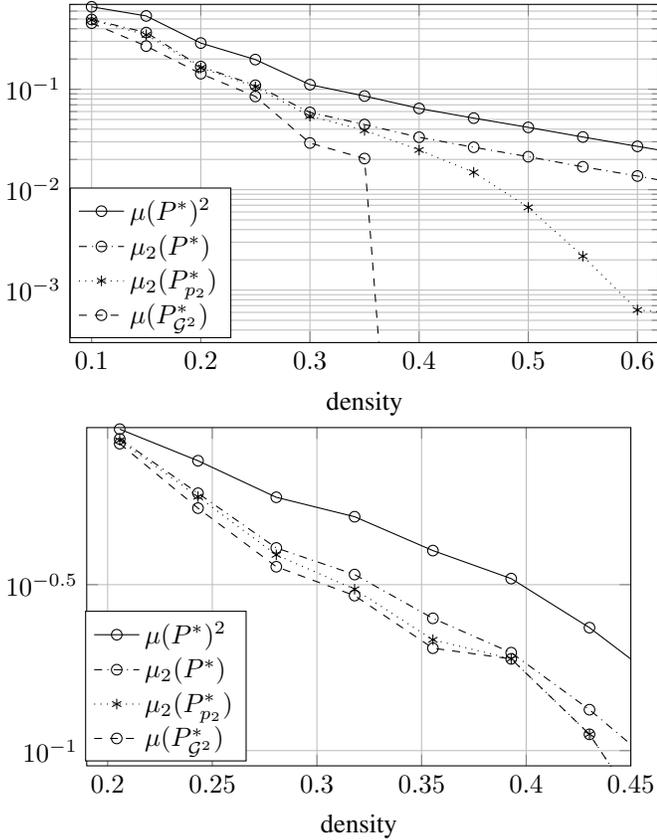
Figure 4: Performance of spectral clustering on an Erdős-Rényi graph with 50 nodes (top) and a random geometric graph with 30 nodes (bottom) as a function of edge density. Depicted are $\mu(P^*)^2$ which corresponds to using the FSSC and (1) over two steps; $\mu_2(P^*)$ the convergence rate with optimal second-order polynomial filter, applied to the given matrix P^* computed for the FSSC; and (hopefully) $\mu_2(P^*_{p2})$, the convergence rate obtained by solving O2 towards optimal tuning of the polynomial and of P.

IV. IMPLEMENTATION AND ROBUSTNESS

A. Implementation in alternating steps

Practical implementation of the optimal acceleration scheme (9) requires to sequentially apply two state update matrices P_- and P_+, such that

$$P_- P_+ = \frac{P^2 - z^2 I}{1 - z^2}, \qquad \text{with } z^2 = \frac{\mu(P)^2 + \sigma(P)^2}{2}.$$

A linear implementation (5) compatible with one-step communication links for fixed P, necessarily takes the form

$$P_- = a \frac{P - zI}{1 - z}, \qquad P_+ = \frac{1}{a} \frac{P + zI}{1 + z}, \qquad a \in \mathbb{R}. \quad (10)$$

The effect of P_- and P_+ on a given eigenvector of P is obtained just by replacing P in (10) by the corresponding eigenvalue λ_i. We immediately see that the eigenvalue $\lambda_1 = 1$, corresponding to the consensus eigenvector, will be multiplied alternatively by a and by $1/a$. In this sense $a > 1$ (or $1/a > 1$) thus implies an "unstable" step with P_- (or with P_+) for the consensus eigenvector. The consensus value is kept *at each step* only if $a = 1$, else it is recovered every second step. However, taking $a \neq 1$ can be interesting regarding the remaining eigenvalues of P.

Definition 3. *We say that the* iteration towards consensus involves stable steps *if both $\mu(P_+) < 1$ and $\mu(P_-) < 1$, i.e., they are stable on the eigenspace orthogonal to the consensus eigenvector associated to the trivial eigenvalue $\lambda_1 = 1$ of P.*

It can also be beneficial towards robustness to depart from the fastest convergence rate and take $z^2 \neq \frac{\mu(P)^2 + \sigma(P)^2}{2}$. Note that the transformation $(z, a, P_\pm) \to (-z, 1/a, P_\mp)$ leaves (10) invariant, so we can assume $z > 0$ by convention. The related properties are characterized as follows.

Proposition 5. *For a given choice of $z \in [0, 1]$:*
(a) With $a = 1$, the matrices P_+ and P_- are both stable if and only if $\mu(P) \leq 1 - 2z$.
(b) There exists $a \neq 1$ such that the iteration towards consensus involves stable steps, if and only if $\mu(P) \leq \sqrt{1 - z^2} - z$. For $z > 1$ there exists no $a \in \mathbb{R}$ for which the iteration towards consensus involves stable steps.

Proof. The iteration towards consensus involves stable steps if and only if $\zeta := \max\{|\mu(P_+)|, |\mu(P_-)|\} < 1$. By linearity of both P_+ and P_- in P and hence in its eigenvalues, we know that

$$\zeta \in \left\{ \left| a \frac{\pm\mu - z}{1 - z} \right|, \; \left| \frac{1}{a} \frac{\pm\mu + z}{1 + z} \right| \right\}.$$

Thanks to this, a quick analysis shows that the choice $a = \sqrt{1 - z}/\sqrt{1 + z}$ minimizes ζ, with associated value $\zeta = (\mu(P) + z)/\sqrt{1 - z^2}$, proving (b). Towards (a), the condition results from $\zeta < 1$ with $a = 1$. Moreover with $a = 1$ the eigenvalue $\lambda_1 = 1$ of P yields eigenvalues 1 for P_+ and P_-, so in fact the matrices are stable. □

For given $\mu(P)$ it is always possible to restrict z such that the stability conditions are satisfied, possibly by taking z smaller than the optimal acceleration value $z^2 = \frac{\mu(P)^2 + \sigma(P)^2}{2}$. The following corollary is easy to check.

Corollary 2. *When taking the optimal acceleration value $z = \sqrt{(\mu(P)^2 + \sigma(P)^2)/2}$:*
(a) The stability condition of Prop.5(a) is satisfied if and only if $\sigma \leq \mu \leq \sqrt{2 - 2\sigma^2} - 1$; this is possible only if $\sigma \leq 1/3$.
(b) The stability condition of Prop.5(b) is satisfied if and only if $\sigma \leq \mu \leq \sqrt{1 - \frac{\sigma^2}{2} - \sqrt{\frac{1}{2} - \frac{\sigma^4}{4}}}$; this is possible only if $\sigma \leq 1/\sqrt{5} \simeq 1.34/3$.

In particular, for finite-time consensus with stable steps, this yields the following conditions: the Laplacian L has three distinct eigenvalues $\lambda_1 = 0 < \lambda_2 < \lambda_3$ and independently of its rescaling, to satisfy the condition Cor.2(a) we need $\lambda_3 - \lambda_2 < \lambda_2$; to satisfy Cor.2(b) we need $\lambda_3 - \lambda_2 < 1.618 \lambda_2$.

The conditions for Cor.2(a) are stronger than for Cor.2(b), showing that $a \neq 1$ can be beneficial towards robustness. Figure 5 illustrates such a situation, where z has been fixed to the optimal value solving O2.

The stability of individual steps can be relevant if we cannot ensure that all intended steps will be applied, e.g., due to synchronization issues. For standard consensus with P this implies no problem, just skipped steps. In accelerated consensus with $a = 1$, if by chance P_- is applied more

frequently than P_+ in the situation of Fig. 5, then one mode increases in an unstable way. We can prevent this risk of instability with $\widetilde{P}_\pm$, which are identical to $P_\pm$ except they take $a \neq 1$. In this case, more frequent applications of $\widetilde{P}_-$ will just change the consensus *value* to something (unstably) different from the average of initial values, but it will not prevent the agents from converging to consensus. The preferable tradeoff depends on the practical situation.
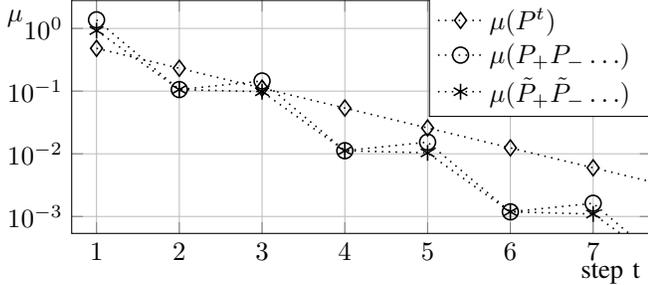

Figure 5: Convergence of the individual steps when using standard consensus (P), optimal p_2 acceleration with $a = 1$ (periodic repetition of $P_+ P_- \ldots$) and with $a = \sqrt{1-z}/\sqrt{1+z}$ (periodic repetition of $\widetilde{P}_+ \widetilde{P}_- \ldots$). The values in this example are obtained by solving O2, i.e., with optimized edge weights and polynomial (z), for the graph shown in figure 6.

B. Robustness to link failure

Consider a consensus scheme tailored to an undirected graph $\mathcal{G}$ with weight matrix P, featuring positive weight on each link. A sudden edge failure leads to a modified graph $\mathcal{G}'$ with weights P'. The latter is obtained, in the standard model of edge failure, by dropping the associated term $(x_j - x_i)$ in (1) and redefining L from there. The standard consensus dynamics will always remain stable under such failure since P' remains doubly-stochastic (see e.g., [28]). I.e. although the node values will not necessarily converge to consensus anymore when link failures lead to a disconnected graph, they will never asymptotically drift away from each other. The p_2-accelerated dynamics are affected as follows by link failure.

Proposition 6. *(a) There exist graphs $\mathcal{G}$ and P associated to positive edge weights, for which the optimal p_2 filter can become unstable with weights P' resulting from a link failure (permanent link failure), and also if the link fails one step out of two (resonant link failure).*
(b) Restricting $z \leq 1/\sqrt{2}$ (resp. $z \leq (1-\mu(P))/2$) in the polynomial filter ensures stability under permanent (resp. resonant) link failures from a P associated to positive edge weights, as for standard consensus.
(c) Any P matrix which features some negative edge weights can become unstable under link failures, both for standard consensus and with p_2 acceleration.
(d) Robustness to link failure is independent of the choice of a in (10).

Proof. Point (d) results from the fact that even if links fail at some times, in absence of other casualties, the scheme keeps alternating the two steps of (10), possibly with different P matrices but still with the a factors canceling:
$P'_- P''_+ = a \frac{P'-zI}{1-z} \cdot \frac{1}{a} \frac{P''+zI}{1+z} = \frac{P'-z}{1-z} \frac{P''+z}{1+z}$. Point (c) is trivial if

we consider the case where all links with positive edge weights (attraction between agents) fail, while all links with negative edge weights (repulsion between agents) remain. Indeed, this leaves only repulsive dynamics and we can write $P' = I + L'$ where L' is a Laplacian with non-negative eigenvalues – i.e. some eigenvalues of P' will necessarily be larger than 1. Let us now turn to the case of P restricted to positive edge weights. Under permanent link failure, the issue is the same as discussed in section II-A. The polynomial filter is applied to a different set of eigenvalues $\in [-1, 1]$, all closer to 1 than the original ones [28]. If $|p_2(\lambda)| \leq 1$ for all $\lambda \in [-1, 1]$ then there is no risk of instability. However, if $p_2(0) < -1$, an eigenvalue of P might become close to zero after link failure and lead to instability; Figure 6 illustrates this negative effect on a specific graph, proving (a). Excluding $p_2(0) < -1$ leads to the condition $z \leq 1/\sqrt{2}$ for (b), which is strictly equivalent to $\widetilde{\mu}^2 + \widetilde{\sigma}^2 < 1$ in Prop.2(a).
Under resonant link failure, when P'_+ corresponds to all links failing and P'_- to no link failure, the eigenvector of $\mu(P)$ is multiplied by $\frac{-\mu(P)-z}{1-z}$ over two time steps. This is the worst case: just consider the worst graph achievable with link failures separately for each step. The condition in (b) accordingly expresses $\left| \frac{-\mu(P)-z}{1-z} \right| < 1$. The graph of Fig. 6 also features instability under resonant link failure, hence proving (a). $\square$

The instability under link failure is caused by a potentially unstable region on the polynomial, characteristic of highly clustered spectra. Figure 6 shows a 5-node graph whose spectrum is so clustered that the optimal polynomial has an unstable region in the center, i.e., $|p_2| > 1$ on some interval inside $[-1, 1]$. And indeed when any of the dashed edges fails, one of the eigenvalues hops into this region. One easily checks numerically that for both permanent and resonant failure the scheme can turn unstable. Constraining the polynomial according to Prop.6(b) restores robustness of the scheme to link failure, but lowers its acceleration.
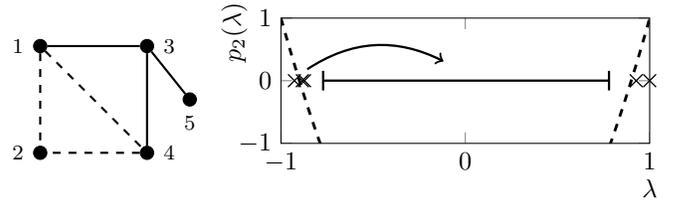

Figure 6: The graph on 5 nodes defined by the union of full and dashed edges on the left, is preconditioned for p_2 acceleration with weights 0.628 on $(1,2), (2,4)$, 0.605 on $(1,4)$, 0.045 on $(1,3), (3,4)$, 0.926 on $(3,5)$. Its highly clustered spectrum ($\times$) and optimal p_2 are shown on the right. Failure of any of the dashed edges leads to instability as an eigenvalue moves into the unstable region of p_2, as indicated by the arrow on the spectrum.

V. FINAL DISCUSSION

A. Higher Order Polynomial Filtering

While this paper focuses on second-order acceleration, the following results can be extended to arbitrary order polynomials p_r:

- The bound of Proposition 4 holds verbatim in the form $\mu_r(P^*_{p_M}) \geq \mu(P^*_{\mathcal{G}^r})$ with $\mathcal{G}^r$ the rth power graph.
- The non-convexity proof of Appendix B holds, replacing each edge in the example by a path of appropriate length.
- The robustness discussion remains qualitatively the same.

Other questions remain open:

- The closed form for the optimal polynomial — if there actually is a unique one — still needs to be investigated. It will certainly exploit more graph information than $\mu(P)$ and $\sigma(P)$. Although not a closed form, the approach of [27] might allow solving for the optimal polynomial.
- Graphs allowing finite-time convergence can be investigated, with obvious improvement as r tends to N. Note that still for p_2 we do not have a complete answer, among others due to the necessity to consider beneficial symmetry-breaking.
- For the gradient descent and associated investigation, a closed form for the optimal polynomial would be welcome, else the polynomial parameters can be part of the optimization variables (see e.g. [24]).

However, relevant situations for *practical implementation* of consensus with high-order polynomial filters would probably be the first question to consider.

B. Conclusions

In this paper we have characterized the possibilities to accelerate linear consensus by second-order polynomial filtering as proposed in [13]. We have observed that this strategy is beaten by an acceleration based on local memory slots if only an upper and a lower bound are known on the graph spectrum (Proposition 1). However when more is known about the graph spectrum, performance can be improved significantly (Corollary 1). For a graph with fixed weights the optimal filter and its convergence rate were derived exactly. A preconditioner is proposed which optimizes the edge weights of a given graph, clustering its eigenvalues towards better polynomial acceleration. Unlike for standard consensus this optimization appears to be non-convex. Significant payoffs are obtained especially for graphs with high edge density. It is not clear at this point which other easily distinguishable factors are beneficial towards polynomial acceleration. We have been surprised to find no answer in the literature, even regarding standard consensus algorithms, about how precisely the diameter of a graph might bound the best achievable convergence rate with optimized edge weights.

In a broader scope, we notice that an approach similar to polynomial filtering has been proposed a few decades ago to control LTI systems using periodic memoryless output feedback [30]. They show that introducing periodically varying feedback can widen the eigenvalue assignment possibilities. We anticipate that those accelerations based on additional memory or time-dependent actions could also be linked to the memory effects and parallel actions present in quantum random walks [31]. We are currently working on formalizing this link in the emerging field of quantum systems engineering.

APPENDIX A

Figure 7 shows the 6-node, diameter-2 graph called coantenna. The graph allows to reach 2-step consensus by applying $x(2) = P_2 P_1 x(0)$, using the compatible matrices P_1 and P_2 shown in equation (12) below. We now prove that two-step consensus is however impossible if P_1 and P_2 are restricted to be symmetric, as well as for non-symmetric P_i but under the restriction of polynomial filtering $P_2 = cI + (1-c)P_1$, $c \in \mathbb{R}$. Note that in Prop.3(b) we have both restrictions.

Let $\vec{1} \in \mathbb{R}^N$ the column vector with all components equal to 1. Two-step consensus would imply the existence of $P_2 = A$ and $P_1 = B$, both compatible with $\mathcal{G}$, and such that $AB = \vec{1}\vec{1}^T/6$. Denoting $C = AB$, this implies:

- $C_{2,5} = A_{2,1}B_{1,5} = 1/6$, $C_{2,6} = A_{2,1}B_{1,6} = 1/6$ $\Rightarrow B_{1,5} = B_{1,6}$
- $C_{3,6} = A_{3,1}B_{1,6} = 1/6$ $\Rightarrow A_{2,1} = A_{3,1}$
- $C_{3,5} = A_{3,4}B_{4,5} + A_{3,1}B_{1,5} = A_{3,4}B_{4,5} + 1/6 = 1/6$ $\Rightarrow A_{3,4}B_{4,5} = 0$
- same reasoning for inverse paths $\Rightarrow A_{5,1} = A_{6,1}$ and $B_{1,2} = B_{1,3}$ and $A_{5,4}B_{4,3} = 0$.

Using two symmetric matrices A and B: Then $A_{3,4}B_{4,5} = 0$ implies either $A_{3,4} = A_{4,3} = 0$ or $B_{4,5} = B_{5,4} = 0$. Yet neither of the latter is possible since $C_{4,2} = A_{4,3}B_{3,2} = 1/6$ and $C_{6,4} = A_{6,5}B_{5,4} = 1/6$.

Using a polynomial filter: Let A and B be non-symmetric, but restricted to $B = cI + (1-c)A$, $c \in \mathbb{R}$. We immediately see that this implies

$$A_{i,j} = c B_{i,j} \text{ for all } i \neq j . \qquad (11)$$

Then $A_{3,4}B_{4,5} = 0$ requires either $A_{3,4} = 0$ which implies $B_{3,4} = 0$, or $B_{4,5} = 0$ which implies $A_{4,5} = 0$. The choice $A_{3,4} = B_{3,4} = 0$ can be implemented on a graph $\mathcal{G}'$ which is identical to the co-antenna except that the edge $(3,4)$ is absent; yet this $\mathcal{G}'$ has a diameter 3 and thus by Prop.3, using A, B compatible with $\mathcal{G}'$ cannot allow two-step convergence. The choice $A_{4,5} = B_{4,5} = 0$ is similar.
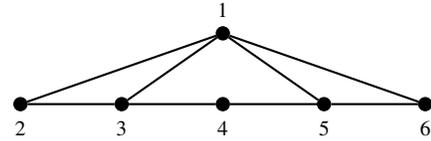

Figure 7: Coantenna graph.

$$P_2 P_1 = \frac{2}{3} \begin{bmatrix} 5/8 & 5/4 & 5/4 & 0 & 5/4 & 5/4 \\ 15/16 & 5/4 & 5/4 & 0 & 0 & 0 \\ -1/2 & -1 & -1 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 \\ -1/2 & 0 & 0 & 1/2 & -1 & -1 \\ 15/16 & 0 & 0 & 0 & 5/4 & 5/4 \end{bmatrix}$$
$$\times \frac{1}{4} \begin{bmatrix} 4/5 & 4/5 & 1 & 0 & 1 & 4/5 \\ 4/5 & 8/5 & 2 & 0 & 0 & 0 \\ 4/5 & 8/5 & 2 & 0 & 0 & 0 \\ 0 & 0 & -1 & 4 & -1 & 0 \\ 4/5 & 0 & 0 & 0 & 2 & 8/5 \\ 4/5 & 0 & 0 & 0 & 2 & 6/5 \end{bmatrix} = \frac{1}{6} \vec{1}\vec{1}^T . \qquad (12)$$

APPENDIX B

Proposition 7. *There exist graphs $\mathcal{G}$ such that the set of all $p_2(P)$, P compatible with $\mathcal{G}$, is a non-convex subset of the set of all $P_{\mathcal{G}^2}$, $P_{\mathcal{G}^2}$ compatible with $\mathcal{G}^2$.*

Proof. Consider $\mathcal{G} = \mathcal{K}(1,4)$, the star graph on 5 nodes (see fig. 8). Let P' compatible with $\mathcal{G}$ have equal positive weights only on the edges (1,2) and (1,3), and another compatible P'' have the same positive weights only on (1,4) and (1,5). Then for p_2', p_2'' some second-order polynomials, $p_2'(P')$ and $p_2''(P'')$ have positive weights on respectively (1,2),(1,3),(2,3) and (1,4),(1,5),(4,5). Their convex combination $P_{\mathcal{G}^2} = 1/2\, p_2''(P'') + 1/2\, p_2'(P')$, which is compatible with $\mathcal{G}^2$, cannot be generated by any $p_2(P)$. Indeed, $p_2(P)$ having (strong) weights on (2,3), (4,5) would require P to have positive weights on all its 4 edges, which in turn unavoidably implies that $p_2(P)$ would have (non-negligible) positive weights on (2,4),(2,5),(3,4),(3,5). $\qquad\square$
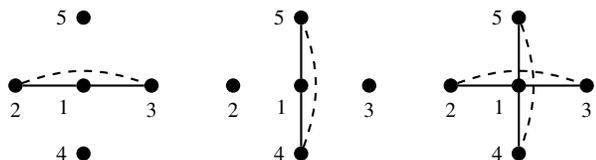

Figure 8: *Left: P'*, dashed line added for $p_2'(P')$. *Center:* idem for P'' and $p_2''(P'')$. *Right:* Convex combination.

REFERENCES

[1] J. N. Tsitsiklis, "Problems in decentralized decision making and computation." Ph.D. dissertation, 1984.

[2] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing markov chain on a graph," *SIAM review*, vol. 46, no. 4, pp. 667–689, 2004.

[3] F. Chen, L. Lovász, and I. Pak, "Lifting markov chains to speed up mixing," in *Proceedings of the thirty-first annual ACM symposium on Theory of computing.* ACM, 1999, pp. 275–281.

[4] J. N. Tsitsiklis, D. P. Bertsekas, M. Athans *et al.*, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE transactions on automatic control*, vol. 31, no. 9, pp. 803–812, 1986.

[5] A. Ribeiro, I. D. Schizas, S. Roumeliotis, G. B. Giannakis *et al.*, "Kalman filtering in wireless sensor networks," *Control Systems, IEEE*, vol. 30, no. 2, pp. 66–86, 2010.

[6] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Approximate distributed kalman filtering in sensor networks with quantifiable performance," in *Proceedings of the 4th international symposium on Information processing in sensor networks.* IEEE Press, 2005, p. 18.

[7] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Distributed kalman filtering based on consensus strategies," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 4, pp. 622–633, 2008.

[8] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *American Control Conference, 2005. Proceedings of the 2005.* IEEE, 2005, pp. 1859–1864.

[9] V. Blondel, J. M. Hendrickx, A. Olshevsky, J. Tsitsiklis *et al.*, "Convergence in multiagent coordination, consensus, and flocking," in *IEEE Conference on Decision and Control*, vol. 44, no. 3. IEEE; 1998, 2005, p. 2996.

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

[11] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[12] S. Muthukrishnan, B. Ghosh, and M. H. Schultz, "First- and second-order diffusive methods for rapid, coarse, distributed load balancing," *Theory of Computing Systems*, vol. 31, no. 4, pp. 331–354, 1998.

[13] E. Montijano, J. I. Montijano, and C. Sagues, "Chebyshev polynomials in distributed consensus applications," *Signal Processing, IEEE Transactions on*, vol. 61, no. 3, pp. 693–706, 2013.

[14] J. M. Hendrickx, R. M. Jungers, A. Olshevsky, and G. Vankeerberghen, "Graph diameter, eigenvalues, and minimum-time consensus," *Automatica*, vol. 50, no. 2, pp. 635–640, 2014.

[15] L. Hogben, "Spectral graph theory and the inverse eigenvalue problem of a graph," *Electronic Journal of Linear Algebra*, vol. 14, no. 1, p. 3, 2005.

[16] A. Olshevsky, "Linear time average consensus on fixed graphs and implications for decentralized optimization and multi-agent control," *ArXiv e-prints*, 2014.

[17] M. Cao, D. A. Spielman, and E. M. Yeh, "Accelerated gossip algorithms for distributed computation," in *Proc. of the 44th Annual Allerton Conference on Communication, Control, and Computation.* Citeseer, 2006, pp. 952–959.

[18] A. Sarlette, "Adding a single memory per agent gives the fastest average consensus," *arXiv preprint arXiv:1412.0402. conditionally accepted by IEEE Trans. Automatic Control*, 2014.

[19] P. Diaconis, S. Holmes, and R. M. Neal, "Analysis of a nonreversible markov chain sampler," *Annals of Applied Probability*, pp. 726–752, 2000.

[20] J. M. Hendrickx, G. Shi, and K. H. Johansson, "Finite-time consensus using stochastic matrices with positive diagonals," *Automatic Control, IEEE Transactions on*, vol. 60, no. 4, pp. 1070–1073, 2015.

[21] A. Y. Kibangou, "Finite-time average consensus based protocol for distributed estimation over awgn channels," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on.* IEEE, 2011, pp. 5595–5600.

[22] S. Safavi and U. Khan, "Revisiting finite-time distributed algorithms via successive nulling of eigenvalues," *Signal Processing Letters, IEEE*, vol. 22, no. 1, pp. 54–57, 2015.

[23] C.-K. Ko and X. Gao, "On matrix factorization and finite-time average-consensus," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on.* IEEE, 2009, pp. 5798–5803.

[24] A. Sandryhaila, S. Kar, and J. M. Moura, "Finite-time distributed consensus through graph filters," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on.* IEEE, 2014, pp. 1080–1084.

[25] S. Sundaram and C. N. Hadjicostis, "Finite-time distributed consensus in graphs with time-invariant topologies," in *American Control Conference, 2007. ACC'07.* IEEE, 2007, pp. 711–716.

[26] E. Kokiopoulou and P. Frossard, "Accelerating distributed consensus using extrapolation," *Signal Processing Letters, IEEE*, vol. 14, no. 10, pp. 665–668, 2007.

[27] ——, "Polynomial filtering for fast convergence in distributed consensus," *Signal Processing, IEEE Transactions on*, vol. 57, no. 1, pp. 342–354, 2009.

[28] A. E. Brouwer and W. H. Haemers, *Spectra of graphs.* Springer Science & Business Media, 2011.

[29] S. Boyd, P. Diaconis, P. Parrilo, and L. Xiao, "Fastest mixing markov chain on graphs with symmetries," *SIAM Journal on Optimization*, vol. 20, no. 2, pp. 792–819, 2009.

[30] D. Aeyels and J. L. Willems, "Pole assignment for linear time-invariant systems by periodic memoryless output feedback," *Automatica*, vol. 28, no. 6, pp. 1159–1168, 1992.

[31] Y. Aharonov, L. Davidovich, and N. Zagury, "Quantum random walks," *Physical Review A*, vol. 48, no. 2, p. 1687, 1993.