

Accelerating Consensus by Spectral Clustering and Polynomial Filters

Simon Apers ^{*} and Alain Sarlette [†]

March 5, 2015

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. 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. A possible loss of robustness of the polynomial filter is also highlighted.

1 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 and sensor fusion has been a major focus. For synchronized fixed networks, one can optimize the weights on the links [2], add local memory [4], or introduce time-varying filters [5, 6]. The present paper establishes the benefit of combining polynomial filtering with optimization of link weights.

Consider 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 $x = (x_1, x_2, \dots, x_N) \in \mathbb{R}^N$. 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 $k \in \mathbb{N}$, $w \in \mathbf{R}$ 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

$$x_1 = x_2 = \dots = x_N =: c \text{ or equivalently } P_w x = x$$

(if the graph is connected); and *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 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’’ $1 - \mu(P_w)$. For a given graph, the fastest convergence is obtained by choosing the edge weights to maximize the spectral gap. This is a convex problem [2], whose solution we call the Fastest Single-Step Convergence network (FSSC)¹. In case of a fixed L , with knowledge of bounds $\underline{\lambda} < \bar{\lambda}$ on its eigenvalues, the optimal μ is obtained by choosing w such that $\mu = (1-w\underline{\lambda}) = -(1-w\bar{\lambda})$.

Accelerated consensus denotes the expansion of node and communication features, under the same graph constraint, to improve convergence speed.

In [4] and later papers, *memory slots* are added at each node to get

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

with some fixed $w_1, w_2 \in \mathbb{R}$. Extending the memory registers does not improve convergence speed if only bounds $\underline{\lambda} < \bar{\lambda}$ on the eigenvalues of a fixed L are known [7].

Another approach is *polynomial filtering*, where a time-varying choice of P_w accelerates convergence [5, 6]. The idea is that if L has eigenvectors \tilde{x}_i with eigenvalues λ_i , then after t steps of $P_w(k) = I - w(k)L$, each eigenvector has been multiplied by $p_t(\lambda_i) = \prod_{k=1}^t (1 - w(k)\lambda_i)$. Through choosing a set of $w(k)$, p_t can be any polynomial of order t satisfying $p_t(0) = 1$; whereas constant w restricts to $p_t(\lambda_i) = (1 - w\lambda_i)^t$. Choosing $w(k) = 1/\lambda_k$ would imply finite-time convergence. The latter requires not only to implement a high-order polynomial filter (see Section 4.1), but also to know the eigenvalues exactly. In the opposite case where we only know that the eigenvalues lie in a bounded interval, we must apply the following result.

Proposition 1. *If the eigenvalues of L span the whole interval $\lambda_i \in [\underline{\lambda}, \bar{\lambda}]$, then the optimal memory slot dynamics (2) is at least as fast as (1) with any time-varying w (i.e. any order of polynomial filtering).*

Proof. [5] gives the optimal polynomial filter and its worst-case long-term convergence speed: for w such that $\mu = (1-w\underline{\lambda}) = -(1-w\bar{\lambda})$ we have $\max_{|1-w\lambda| < \mu} [p_t(\lambda)] =$

¹Boyd et al. study the optimization for Markov chains, with the constraint $[P_w]_{i,j} > 0$ for all i, j . For consensus that constraint also favors robustness to network changes. In the present paper we drop it, for a fairer comparison with other algorithms; the optimization remains a convex problem.

$1/|T_t(1/\mu)|$ with p_t rescaled from a t -order Chebyshev polynomial T_t . A function plot readily shows that

$$\frac{1}{|T_t(1/\mu)|^{1/t}} \geq \frac{1}{\mu} - \sqrt{\frac{1}{\mu^2} - 1} \quad (3)$$

for all $\mu \in (0, 1)$, with equality as $t \rightarrow +\infty$. The right side of (3) is precisely the worst-case convergence speed with optimal (2), see [4, 7]. \square

Whether a synchronous variation of w or a local memory slot is a more demanding resource, is application dependent. One local memory allows to evaluate T_t with a stable recurrence [4]. Our new observation with Proposition 1 is that with respect to this implementation, (2) is superior.

However, in the following, we show how much more can be gained by the polynomial filter when (somewhat) more is known about L . We restrict our investigation to a two-step alternating scheme:

$$x(k+2) = (I-w_2L)(I-w_1L)x(k), \quad (4)$$

performing second-order polynomial filtering on the L -spectrum (or equivalently on the P -spectrum).

Note that acceleration schemes have also been proposed for Markov chains (see e.g. [3] and related work). However, the most prominent schemes explicitly build on full knowledge of a given \mathcal{G} , which differs from our setting.

The remainder of the paper is organized as follows. In Section 2 we compute the optimal second-order polynomial and the associated gain in convergence speed. We also mention some special graphs that allow 2-step consensus with possible symmetry-breaking. In Section 3 we consider the optimization of graph weights towards polynomial filtering. Section 4 discusses the implementation of the acceleration scheme and its robustness properties.

2 Optimizing the polynomial filter

We now investigate the form of the optimal second-order polynomial filter (4) for a given graph \mathcal{G} , about which we possibly know more than just the spectral gap. Before the rescaling of L in (4), we can assume without loss of generality that $P := I - L$ is centered, i.e. $-\mu(P)x^T x \leq x^T P x \leq \mu(P)x^T x$ for all x with $\sum_{i=1}^N x_i = 0$, with each equality achieved for some x , and where $\mu(P)$ is the SLEM of P . In this situation, the optimal polynomial filter only depends on P 's SLEM and *Smallest Eigenvalue Modulus* $\sigma(P)$ (SEM).

Explicitly, rewrite (4) as

$$x(k+2) = \frac{(P - z_1)(P - z_2)}{(1 - z_1)(1 - z_2)} x(k) =: p_2(P) x(k) \quad (5)$$

and denote by $\{\lambda_i : i = 1, 2, \dots, N\}$ the eigenvalues of P , with $\lambda_1 = 1$ corresponding to the consensus eigenspace. Note that the filter p_2 can be any second-order polynomial restricted to $p_2(1) = 1$. 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 (5) is given by*

$$\mu_2(P) = \frac{\mu(P)^2 - \sigma(P)^2}{2 - \mu(P)^2 - \sigma(P)^2} \quad (6)$$

and obtained with the unique polynomial

$$p_2(P) = \frac{P^2 - \mu(P)^2/2 - \sigma(P)^2/2}{1 - \mu(P)^2/2 - \sigma(P)^2/2}. \quad (7)$$

Proof. For any $[|a|, |b|] \subseteq [0, 1]$, the second-order polynomial $p_2(x)$ for which $p_2(1) = 1$ and which minimizes $\max_{x \in \{a, b, -b\}} |p_2(x)|$ is determined by $p_2(b) = p_2(-b) = -p_2(a) = -p_2(-a)$. Replacing $|a|, |b|$ by particular eigenvalues $\sigma(P), \mu(P)$, this polynomial bounds the best possible performance. Moreover, for this same polynomial we have $|p_2(x)| \leq |p_2(b)|$ for any $|x| \in [a, b]$, so it actually gives the best performance, independent of the other eigenvalues of P . \square

The convergence rate $\mu_2(P)$ must be compared with $\mu(P)^2$, the convergence over *two steps* of the standard consensus algorithm (1). The resulting improvement by p_2 -acceleration is illustrated on Figure 1.

2.1 Favorable graphs and symmetry breaking

In light of the previous result, a given weighted graph \mathcal{G} will allow substantial acceleration through second-order filtering if its (e.g. Laplacian) eigenvalues are clustered in two sets whose width is small compared to the distance between the sets. This situation easily generalizes to the case of n clusters for n -order polynomial filters, at least conceptually; explicit forms generalizing (6),(7) may be more difficult to obtain.

This observation motivates a different way of optimizing network links, if some design freedom is available. Before turning to an investigation of such optimization, we briefly consider the special cases where *finite-time convergence to consensus is achieved* in two steps.

The study of finite-time consensus has its own line of work, to which the present results connect through [6, 8] and the like. 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 (6),(7) imply perfect consensus after one application of (4). Graphs \mathcal{G} for which P_1, P_2 can be selected such that

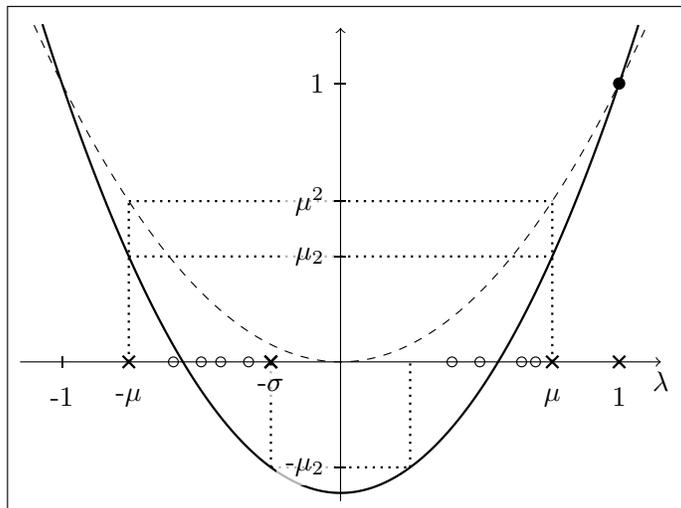


Figure 1: Comparison of SLEM after 2 steps of standard consensus (μ^2) and convergence rate under optimal p_2 -acceleration (μ_2), for some arbitrary P . The critical eigenvalues, determining the convergence rate, are distinguished (\times) from the other ones (\circ). The polynomials $y = \lambda^2$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

$P_1 P_2$ features deadbeat convergence are characterized in [6]. The polynomial filter is restricted to $P_2 = cI + dP_1$, with $c, d \in \mathbb{R}$. It is presently unclear how much more restrictive this is, but the following results nevertheless carry over from [6].

Proposition 2. • *If 2-step convergence can be reached on a graph \mathcal{G} then its diameter $d(\mathcal{G}) \leq 2$.*

- *$d(\mathcal{G}) \leq 2$ is not sufficient for 2-step convergence.*
- *If in addition \mathcal{G} is distance-regular, then 2-step convergence can be attained.*

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) converge in two steps with p_2 -acceleration; while for (1) only the complete graph converges in finite time. Lists of graphs with only two different nonzero eigenvalues can be found in the literature, see e.g. [9, Table 14.2, Table 14.4, Chapter 15.2] assuming uniform weights.

Examining the complete bipartite graphs $K(\ell, m)$ between sets of ℓ and m nodes allows to illustrate interesting finite-time convergence properties. Both $\mathcal{K}(m, m)$ and the star graph $\mathcal{K}(1, m)$ allow finite-time convergence using p_2 with uniform link weights, 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. Regarding μ^2 (FSSC) this choice is optimal, see [2, 10]. Regarding μ_2 , 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 asym-

metric weights $p \neq q$ (see figure 2), the nonzero eigenvalues are proportional to:

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

By choosing $q = \frac{1}{2}(m \pm \sqrt{m^2 - 4})p$, this set reduces to two distinct values and $\mu_2(P) = 0$. This possible benefit of symmetry breaking contrasts with standard consensus (1), for which [10] shows that keeping the edge-transitivity symmetry in the weights does lead to the FSSC.

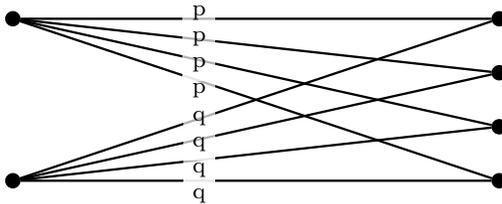


Figure 2: The complete bipartite graph $\mathcal{K}(2, 4)$ with weights p and q .

Optimization of the edge weights towards p_2 -acceleration, for a given graph structure, is further considered in the following section.

3 Preconditioning graphs: spectral clustering

Theorem 1 describes the convergence rate obtained with an optimal polynomial filter for a given weighted graph (spectrum). This section treats the optimization of the edge weights in compliance with the graph connectivity, which we call *preconditioning* the graph towards p_2 -acceleration. This is in the same spirit as the FSSC. However, optimal *polynomial* acceleration favors an eigenvalue spectrum in distinct *clusters*. A positive point is that Theorem 1 gives an explicit expression for the optimal polynomial, so we can efficiently concentrate on optimizing P . For a given $\mathcal{G}(\mathcal{V}, \mathcal{E})$, we denote by P^* the FSSC i.e. the optimal choice of weights for (1), by $P_{p_2}^*$ the optimal weights for p_2 acceleration.

3.1 Bound on fastest convergence rate

We first get a bound on the possible acceleration. Given \mathcal{G} , denote \mathcal{G}^2 its *square graph*, in which two nodes are linked by an edge iff in \mathcal{G} they are linked by a path of ≤ 2 edges. We denote an arbitrary weight matrix on this power graph by $P_{\mathcal{G}^2}$.

Proposition 3. *The fastest convergence rate attainable on \mathcal{G} using p_2 -acceleration is bounded by the SLEM of the Fastest Single-Step Consensus on \mathcal{G}^2 , i.e.*

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

Proof. For $p_2(P)$ we effectively tune the same edge weights as for $P_{\mathcal{G}^2}$, except that in the latter the individual weights are unconstrained while in the former they are interdependent through the polynomial structure. \square

This bound is not always tight: 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 2 however states that this does not imply $\mu_2(P_{p_2}^*) = 0$.

3.2 Numerical optimization

The FSSC choice of P^* does not necessarily minimize μ_2 , see e.g. the $\mathcal{K}(2, m)$ -graph (section 2.1); 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 3, we try to reformulate the problem 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 improve convexity as the set of all $p_2(P)$ is a non-convex subset of the set of all $P_{\mathcal{G}^2}$ (see Proposition 6 in appendix). So to have a convex problem we indeed need to relax the subspace to all $P_{\mathcal{G}^2}$. 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 reformulation is found. In the meantime, we have performed a numerical optimization by *gradient descent*, minimizing the explicit function μ_2 as a function of the edge weights. 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 one to evaluate the potential of preconditioning for p_2 -acceleration. As the used formulation is non-convex, the method converges to a local minimum, not guaranteeing the global optimum. The bound $\mu(\mathcal{G}^2)$ from Proposition 3 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 for acceleration preconditioning ($P_{p_2}^*$, hopefully). The difference is graphically striking as the preconditioning brings $\mu_2(P^*) = 0.0626$ down to $\mu_2(P_{p_2}^*) = 0.0088$. 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 of star graphs (section 2.1). We observe the following behavior in simulations.

- The behavior of the preconditioner and acceleration was examined on a large set of Erdős-Rényi graphs. A clear trend appears when examining

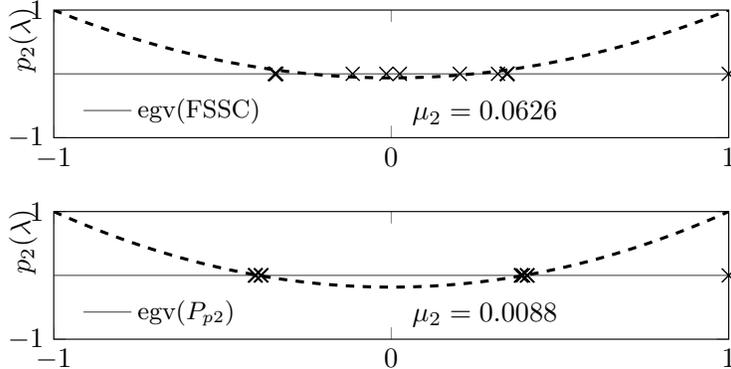


Figure 3: Spectral clustering performed on an Erdős-Rényi graph $\mathcal{G}(20, 9)$. The plot shows the spectrum of both the FSSC P^* and the optimized P_{p_2} , as well as their optimal p_2 -polynomials and corresponding μ_2 values.

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 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 first-order consensus; 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 3 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, 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 3 was reached on several graphs of diameter 3,4 and 5. This seems to indicate that the graph diameter is not a limiting factor for the preconditioner. Surprisingly, it appears that the role of the graph diameter in possible convergence speed is an open question in the literature also for the FSSC/FMMC problem.
- 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

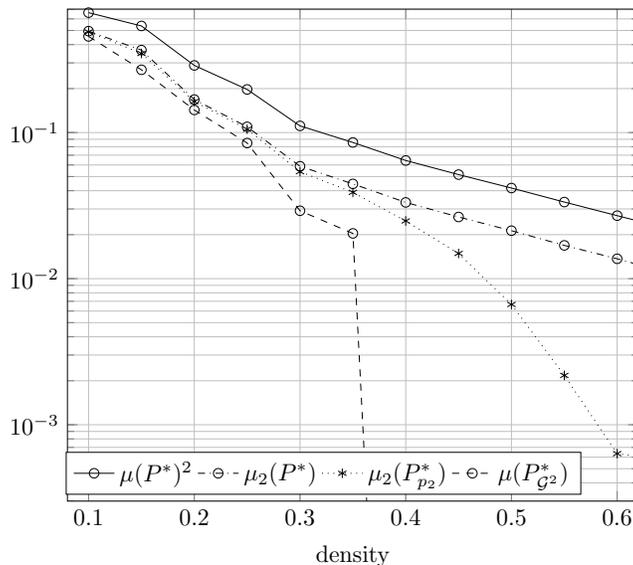


Figure 4: Performance of clustering on an Erdős-Rényi graph with increasing density. Depicted are $\mu(P^*)^2$ using the FSSC during two steps, its p_2 -acceleration $\mu_2(P^*)$, the p_2 -acceleration with optimized graph $\mu_2(P_{p_2}^*)$ (hopefully), and the lower bound $\mu(P_{G_2}^*)$.

highly degenerate in $\pm\mu$ and $\pm\sigma$. This is similar to the degeneracy found at $\pm\mu$ in the spectrum of the FSSC.

4 Robustness

4.1 Implementation in alternating steps

Implementation of the acceleration scheme (7) requires to sequentially apply two matrices P_- and P_+ , such that

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

A linear implementation (4) compatible with one-step communication links for fixed P , necessarily takes the form

$$P_- = a \frac{P - z}{1 - z}, \quad P_+ = \frac{1}{a} \frac{P + z}{1 + z}, \quad a \in \mathbb{R}. \quad (8)$$

The effect of P_- and P_+ on a given eigenvector is obtained just by replacing P in (8) by the corresponding eigenvalue. We immediately see that the eigenvalue 1 of P , 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 . 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 1 of P .

Proposition 4. *For given z : • The matrices P_+ and P_- with $a = 1$ are both stable for consensus if and only if $\mu(P) < 1 - 2z$.*

• *The iteration towards consensus involves stable steps for a proper choice of $a \neq 1$ if and only if $\mu(P) < 1 - z$.*

For given $\mu(P)$ it is always possible to restrict z such that these stability conditions are satisfied, possibly by taking it smaller than the optimal acceleration value $z^2 = (\mu(P)^2 + \sigma(P)^2)/2$.

Proof. Choosing $a = 1$ we get, by linearity of both P_+ and P_- in P and hence in its eigenvalues: $\max\{|\mu(P_+)|, |\mu(P_-)|\} = (\mu(P) + z)/(1 - z)$. The latter is smaller than 1 if and only if $\mu(P) < 1 - 2z$; while the eigenvalue 1 of P remains unchanged in P_+ and P_- when $a = 1$. Choosing $a = \sqrt{1 - z}/\sqrt{1 + z}$ minimizes $\max\{|\mu(P_+)|, |\mu(P_-)|\}$, at the value $(\mu(P) + z)/\sqrt{1 - z^2}$. The latter is smaller than 1 if and only if $\mu(P) < 1 - z$. \square

Figure 5 illustrates the second case of Proposition 4. 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. But 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 P_-^{st} , which takes $a \neq 1$. In this case, more frequent applications of 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.

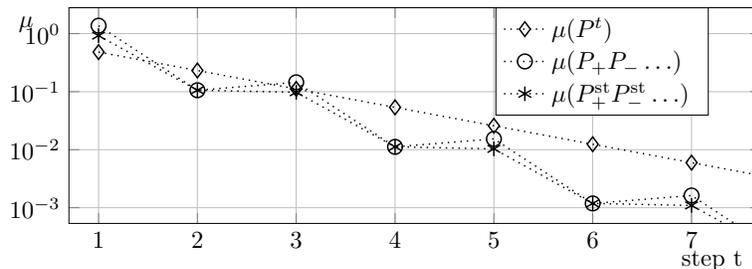


Figure 5: Convergence of the individual steps when using standard consensus (P), optimal p_2 acceleration with $a = 1$ (periodic repetition of $P_+P_- \dots$) and with $a = \sqrt{1 - z}/\sqrt{1 + z}$ (periodic repetition of $P_+^{st}P_-^{st} \dots$), for the graph with optimized weights shown in figure 6.

4.2 Robustness to link failure

Consider a consensus scheme tailored to an initial 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 standard consensus dynamics will remain stable under this failure, as P' remains doubly-stochastic, see e.g. [9]. Link failure affects the p_2 -accelerated dynamics as follows.

Proposition 5. • *For certain graphs, the optimal p_2 filter for a weight matrix P associated to positive edge weights can become unstable with weights P' in which a link has failed (permanent link failure), and also if the link fails one step out of two (resonant link failure).*

- *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.*
- *Any P matrix which features some negative edge weights can become unstable under specific link failures, both for standard consensus and with p_2 acceleration.*
- *Robustness to link failure is independent of the choice of a in (8).*

Proof. The last point 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 (8), possibly with different P matrices but still with the a factors canceling.

The third point 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 polynomial filter is applied to a different set of eigenvalues $\in [-1, 1]$, all closer to 1 than the original ones [9]. If $|p_2(\lambda)| \leq 1$ for all $\lambda \in [-1, 1]$ 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, see Figure 6. This leads to the condition $z \leq 1/\sqrt{2}$.

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 expresses $\left| \frac{-\mu(P)-z}{1-z} \right| < 1$. \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 clustered spectrum has an unstable region in the center, i.e. $|p_2| > 1$ on some interval inside $[-1, 1]$. Indeed, when any of the dashed edges fails, one of the eigenvalues hops into this region. For both permanent and resonant failure the scheme will turn unstable. Constraining the polynomial according to Proposition 5 restores robustness of the scheme but lowers its acceleration.

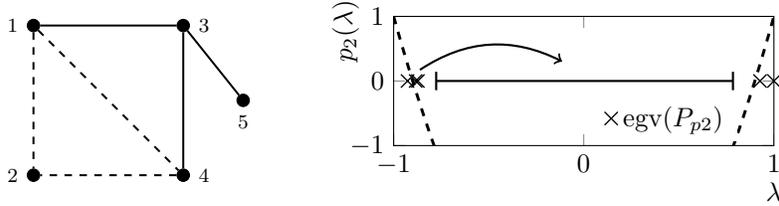


Figure 6: The graph on 5 nodes defined by the union of full and dashed edges on the left, is preconditioned for p_2 acceleration with weights $(1, 2), (2, 4) = 0.628$, $(1, 4) = 0.605$, $(1, 3), (3, 4) = 0.045$, $(3, 5) = 0.926$. Its (highly clustered) spectrum and optimal p_2 are shown on the right. 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.

5 Discussion

5.1 Higher Order Polynomial Filtering

While this paper focuses on second-order acceleration, several results can be extended to arbitrary order polynomials p_M while others 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 $\sigma(P)$.
- Graphs allowing finite-time convergence can be investigated, with obvious improvement as M tends to N , the graph order. Note that still for p_2 , we do not have an exact answer to the question, among others due to the necessity to consider beneficial symmetry-breakings.
- The bound of Proposition 3 holds verbatim in the form $\mu_M(P_{p_M}^*) \geq \mu(P_{\mathcal{G}_M}^*)$.
- The non-convexity of Proposition 6 holds, replacing each edge in the example of the proof by a path of appropriate length.
- 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. [8]).
- The robustness discussion remains qualitatively the same.

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

5.2 Conclusions

In this paper we have characterized the possibilities to accelerate linear consensus by second-order polynomial filtering as proposed in [5]. 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. However when more is known about the graph spectrum, performance can be improved significantly. 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.

A few academic questions remain open. A particular one is whether the possibility of achieving consensus in k steps with a time-varying weight matrix on a fixed graph [6] implies that a k -order polynomial filter can also achieve finite-time consensus. A more general question, for which we have been surprised to find no answer in the literature even regarding standard consensus algorithms, is how 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 [11]. 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 [12]. We are currently working on formalizing this link in the emerging field of quantum systems engineering.

Proposition 6. *The set of all $p_2(P)$ is a non-convex subset of the set of all $P_{\mathcal{G}^2}$.*

Proof. Consider $\mathcal{K}(1,4)$, the star graph on 5 nodes (see fig. 7). Let a given P' have equal positive weights only on the edges (1,2) and (1,3), and 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 $p_2''(P'')$ correspond to positive weights on respectively (1,2),(1,3),(2,3) and (1,4),(1,5),(4,5). Their convex combination $1/2 p_2''(P'') + 1/2 p_2'(P')$ cannot be generated by any $p_2(P)$, because (strong) weights on (2,3), (4,5) require positive weights on all 4 edges of the star, which in turn unavoidably imply (non-negligible) positive weights on (2,4),(2,5),(3,4),(3,5) in any second-order $p_2(P)$. \square

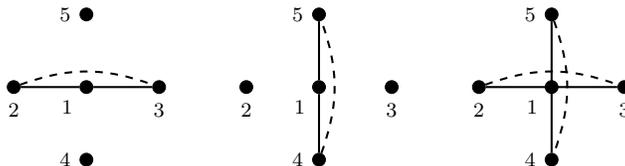


Figure 7: *Left:* P' , dashed line added for $p_2'(P')$. *Center:* idem for P'' and $p_2''(P'')$. *Right:* Convex combination.

Acknowledgment

The authors thank F.Ticozzi for discussions that have drawn their interest to this topic and B.Gerencsér for fruitful discussions mainly about Section 2.1 and Proposition 6. The authors are partially supported by the Interuniversity Attraction Poles program DYSCO, funded by the Belgian Science Policy Office.

References

- [1] J. N. Tsitsiklis, Problems in decentralized decision making and computation., Ph.D. thesis (1984).
- [2] S. Boyd, P. Diaconis, L. Xiao, Fastest mixing markov chain on a graph, *SIAM review* 46 (4) (2004) 667–689.
- [3] F. Chen, L. Lovász, 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] S. Muthukrishnan, B. Ghosh, M. H. Schultz, First- and second-order diffusive methods for rapid, coarse, distributed load balancing, *Theory of Computing Systems* 31 (4) (1998) 331–354. doi:10.1007/s002240000092.
- [5] E. Montijano, J. I. Montijano, C. Sagues, Chebyshev polynomials in distributed consensus applications, *Signal Processing, IEEE Transactions on* 61 (3) (2013) 693–706.
- [6] J. M. Hendrickx, R. M. Jungers, A. Olshevsky, G. Vankeerberghen, Graph diameter, eigenvalues, and minimum-time consensus, *Automatica* 50 (2) (2014) 635–640.
- [7] A. Sarlette, Adding a single memory per agent gives the fastest average consensus, Submitted to *IEEE Trans. Automatic Control*.
- [8] A. Sandryhaila, S. Kar, 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.
- [9] A. E. Brouwer, W. H. Haemers, *Spectra of graphs*, Springer Science & Business Media, 2011.
- [10] S. Boyd, P. Diaconis, P. Parrilo, L. Xiao, Fastest mixing markov chain on graphs with symmetries, *SIAM Journal on Optimization* 20 (2) (2009) 792–819.
- [11] D. Aeyels, J. L. Willems, Pole assignment for linear time-invariant systems by periodic memoryless output feedback, *Automatica* 28 (6) (1992) 1159–1168.

- [12] Y. Aharonov, L. Davidovich, N. Zagury, Quantum random walks, *Physical Review A* 48 (2) (1993) 1687.