



Kempton, L., Herrmann, G., & Di Bernardo, M. (2016). Distributed Adaptive Optimization and Control of Network Structures. In 2016 IEEE 55th Conference on Decision and Control (CDC 2016). (pp. 5839-5844). Institute of Electrical and Electronics Engineers (IEEE). DOI: 10.1109/CDC.2016.7799167

Peer reviewed version

Link to published version (if available):  
[10.1109/CDC.2016.7799167](https://doi.org/10.1109/CDC.2016.7799167)

[Link to publication record in Explore Bristol Research](#)  
PDF-document

This is the accepted author manuscript (AAM). The final published version (version of record) is available online via IEEE at <https://doi.org/10.1109/CDC.2016.7799167>. Please refer to any applicable terms of use of the publisher.

## University of Bristol - Explore Bristol Research

### General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:  
<http://www.bristol.ac.uk/pure/about/ebr-terms.html>

# Distributed Adaptive Optimization and Control of Network Structures

Louis Kempton, Guido Herrmann and Mario di Bernardo

**Abstract**—In this paper we present a generic distributed weight adaptation framework to optimize some network observables of interest. We focus on the algebraic connectivity  $\lambda_2$ , the spectral radius  $\lambda_n$ , the synchronizability  $\lambda_n/\lambda_2$ , or the total effective graph resistance  $\Omega$  of undirected weighted networks, and describe distributed systems for the estimation of these functions and their derivatives for on-line adaptation of the edge weights.

## I. INTRODUCTION

When controlling a multi-agent network, the structure of the network plays a vitally important role in determining the performance of the system. Functions of the eigenvalues of the Graph Laplacian matrix  $\mathbf{L}$ ,  $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ , have been shown to be instrumental in determining the properties and performance of a wide range of multi-agent network systems. For example, the second smallest eigenvalue of  $\mathbf{L}$ , the algebraic connectivity  $\lambda_2$ , governs the convergence rate in many consensus algorithms [1], [2]. In nonlinear synchronization applications, the local transversal stability of the synchronous solution is dependent on  $\lambda_2$  or the eigenratio  $\lambda_n/\lambda_2$  [3], depending on the specific shape of the Master Stability Function [4], [5]. For continuous time linear consensus with time delays, the largest eigenvalue  $\lambda_n$  becomes important for guaranteeing convergence [1]. The robustness of a linear consensus system to additive white noise can be quantified using the Total Effective Graph Resistance [6], [7]:

$$\Omega = n \sum_{i=2}^n \frac{1}{\lambda_i} \quad (1)$$

As such the control and optimization of these spectral functions is important for designing network systems with desirable properties, and much research has been undertaken in this field. In [8] and references therein, a number of functions of graph Laplacian eigenvalues, including the algebraic connectivity and effective graph resistance, are optimized centrally over the edge weights, using semi-definite programming (SDP). Maximising the algebraic connectivity of a state-dependent Laplacian is achieved in [9] and a distributed solution to this problem is presented in [10], achieved through repeated solution of local SDPs. A number of techniques have also been proposed to maintain connectivity of mobile robot networks [11], [12], [13], [14], [15],

This research is funded by the EPSRC.

L. Kempton is with the Bristol Centre for Complexity Sciences, University of Bristol, UK [l.kempton@bristol.ac.uk](mailto:l.kempton@bristol.ac.uk)

G. Herrmann is with the Department of Mechanical Engineering, University of Bristol, UK [g.herrmann@bristol.ac.uk](mailto:g.herrmann@bristol.ac.uk)

M. di Bernardo is with the Department of Engineering Mathematics, University of Bristol, UK. Also with the Department of Electrical Engineering and Information Technology, University of Naples Federico II, Italy [m.dibernardo@bristol.ac.uk](mailto:m.dibernardo@bristol.ac.uk)

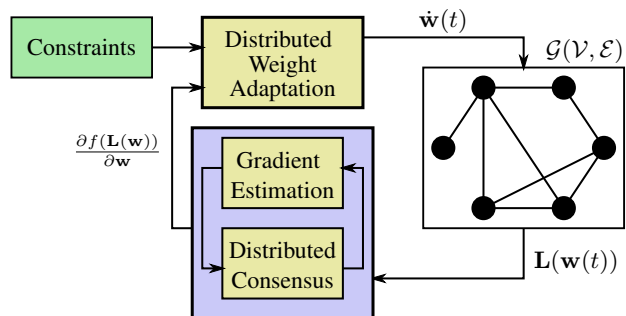


Fig. 1. General overview of the distributed estimator-optimizer system. Distributed estimation of the gradient of the objective function is accomplished in the blue box. Typically this system will require agents to reach consensus on some global variables, and distributed Proportional-Integral consensus is employed to achieve this [21]. The estimated gradient is fed into the optimizer system which formulates a control law for the weight update, using gradient descent whilst enforcing the constraints. The network adapts according to this control law, with the current network structure influencing the gradient estimation.

with [15] proposing an adaptive method for estimating the algebraic connectivity in a completely decentralized manner. This adaptive method was applied in [16] to solve a constrained optimization problem using a weight adaptation law utilizing adaptive logarithmic barrier functions. Minimization of the eigenratio is explored in [17], [18] using simulated annealing, and discrete optimization is employed to increase the synchronisability of the network in [19]. In [20], power iteration was employed for distributed optimization of  $\lambda_2$  and  $\lambda_n$ , but with the restriction that the method breaks down when these eigenvalues are not sufficiently separated.

In this paper, we present a generic multi-level framework for the optimization of a number of spectral functions of the graph Laplacian, as presented in Figure 1. Distributed estimation of the gradient of the objective function forms the foundation of the system, with the gradient estimator frequently requiring the use of a number of average consensus sub-systems. These consensus subsystems allow individuals in the network to cooperatively infer norms of vectors. Finally, a slower weight adaptation layer uses the estimated gradients to evolve the network to an optimal structure.

Our contribution in this paper is threefold. Firstly, we describe a dynamical system for the adaptation of the edge weights that performs gradient descent using the estimated gradients and enforces the boundary conditions, so that an optimal configuration may be reached adaptively and in a totally distributed manner. Secondly, we describe a distributed adaptive system of stochastic differential equations that can be used to estimate the total effective graph resistance and its partial derivatives in an entirely distributed manner. Thirdly, we collect together a number of related network optimiza-

tion problems and discuss their common structures. We demonstrate our approach with several numerical examples, highlighting some of the most interesting features.

## II. WEIGHT ADAPTATION

We begin by proposing a weight adaptation method that finds a local optimum of the constrained continuous optimization problem with objective function  $f(\mathbf{w})$  and  $p$  inequality constraints  $g_k(\mathbf{w})$ , where  $\mathbf{w}$  is the vector of edge weights in the network of interest:

$$\underset{\mathbf{w}}{\text{minimize}} \quad f(\mathbf{w}) \quad (2,a)$$

$$\text{subject to} \quad g_k(\mathbf{w}) \leq 0 \quad \forall k \in 1, \dots, p \quad (2,b)$$

The weight adaptation method we use is the set of ordinary differential equations (ODEs),

$$\dot{w}_{\{i,j\}} = -\frac{\partial f(\mathbf{w})}{\partial w_{\{i,j\}}} - \sum_k \frac{\partial g_k(\mathbf{w})}{\partial w_{\{i,j\}}} \nu_k \quad (3,a)$$

$$\dot{\nu}_k = g_k(\mathbf{w}) \nu_k \quad (3,b)$$

which is based on the first order necessary Karush-Kuhn-Tucker (KKT) conditions [22]. In this equation, the variables  $\nu_k$  behave as KKT multipliers, the dual variables, where the  $w_{\{i,j\}}$  are the primal variables. The first order necessary KKT conditions for a local minimum  $\mathbf{w}^*$  are simply:

- Stationarity:  $\nabla f(\mathbf{w}^*) + \sum_k \nabla g_k(\mathbf{w}^*) \nu_k^* = \mathbf{0}$  which is equivalent to the stationarity of the primal variables  $\dot{\mathbf{w}} = \mathbf{0}$  in Equation (3,a).
- Dual Feasibility:  $\nu_k^* \geq 0$  for all  $k$ . This is guaranteed if the initial conditions  $\nu_k(0)$  are chosen to be positive.
- Primal Feasibility:  $g_k(\mathbf{w}^*) \leq 0$  for all  $k$ . If we assume that the primal variables are infeasible, i.e. there is some  $g_k(\mathbf{w}) > 0$ , then  $\nu_k$  will not be stationary so long as  $\nu_k \neq 0$ . (In the case that  $\nu_k^* = 0$ ,  $g_k(\mathbf{w}^*) > 0$  this stationary point is unstable.)
- Complementary slackness:  $g_k(\mathbf{w}^*) \nu_k^* = 0$  for all  $k$ . This is equivalent to the stationarity of the dual variables  $\dot{\nu}_k = 0$  for all  $k$  in Equation (3,b).

Thus, the stationary point of the system defined by (3) satisfies all of the first order necessary KKT conditions. This is sufficient for optimality if  $f(\mathbf{w})$  is convex and all inequality constraints  $g_k(\mathbf{w})$  are convex and continuously differentiable.

This weight adaptation law is not distributed as the partial derivatives  $\frac{\partial f(\mathbf{w})}{\partial w_{\{i,j\}}}$  and  $\frac{\partial g_k(\mathbf{w})}{\partial w_{\{i,j\}}}$ , and the constraints  $g_k(\mathbf{w})$  may be functions of variables not available to every node. To distribute this weight adaptation law, estimates of the partial derivatives and boundary conditions must be available to each edge.

## III. DISTRIBUTED ESTIMATION OF GRADIENTS

Next, we review distributed, adaptive strategies for estimation of the gradients of some global observables.

### A. Algebraic Connectivity $\lambda_2$

It was shown in [15] that an estimate  $\mathbf{a}$  of the eigenvector  $\mathbf{v}_2$  corresponding to the algebraic connectivity  $\lambda_2$  can be

obtained through the adaptive system:

$$\dot{\mathbf{a}} = -k_1 \frac{\mathbf{1}\mathbf{1}^\top \mathbf{a}}{n} - k_2 \mathbf{L}(\mathbf{w}) \mathbf{a} - k_3 \left( \frac{\mathbf{a}^\top \mathbf{a}}{n} - 1 \right) \mathbf{a} \quad (4)$$

where  $k_1$ ,  $k_2$ , and  $k_3$  are control parameters to be chosen, with  $k_1 > k_2 \lambda_2$ ,  $k_3 > k_2 \lambda_2$ . The algebraic connectivity  $\lambda_2$  can then be computed using the norm of the equilibrium position  $\mathbf{a}^*$  of the system [15] as:

$$\lambda_2 = \frac{k_3}{k_2} \left( 1 - \frac{\mathbf{a}^{*\top} \mathbf{a}^*}{n} \right) \quad (5)$$

Further to this, it was shown that the system can be completely distributed through the use of distributed estimates of the arithmetic mean of  $\mathbf{a}$ ,  $\frac{\mathbf{1}\mathbf{1}^\top \mathbf{a}}{n}$ , and the mean of the squared components of  $\mathbf{a}$ ,  $\frac{\mathbf{a}^\top \mathbf{a}}{n}$ . These distributed estimates were made using two separate Proportional-Integral (PI) average consensus estimators [21]. In the first PI average consensus estimator each agent maintains an estimate for the mean of the components of  $\mathbf{a}$ , so that a vector,  $\pi_{\mathbf{a}}$ , tracks the vector  $\frac{\mathbf{1}\mathbf{1}^\top \mathbf{a}}{n}$ . Another vector  $\mathbf{z}_{\mathbf{a}}$  is the collection of integrator variables, and is not used outside of the consensus estimator [21]. Specifically, we have:

$$\begin{aligned} \dot{\pi}_{\mathbf{a}} &= k_g (\mathbf{a} - \pi_{\mathbf{a}}) - k_P \mathbf{L}(\mathbf{w}) \pi_{\mathbf{a}} + k_I \mathbf{L}(\mathbf{w}) \mathbf{z}_{\mathbf{a}} \\ \dot{\mathbf{z}}_{\mathbf{a}} &= -k_I \mathbf{L}(\mathbf{w}) \pi_{\mathbf{a}} \end{aligned} \quad (6)$$

In the second PI average consensus estimator, the vector of proportional variables  $\pi_{\mathbf{a}^2}$  tracks the vector  $\frac{\mathbf{1}\mathbf{a}^\top \mathbf{a}}{n}$

$$\begin{aligned} \dot{\pi}_{\mathbf{a}^2} &= k_g (\mathbf{a}^2 - \pi_{\mathbf{a}^2}) - k_P \mathbf{L}(\mathbf{w}) \pi_{\mathbf{a}^2} + k_I \mathbf{L}(\mathbf{w}) \mathbf{z}_{\mathbf{a}^2} \\ \dot{\mathbf{z}}_{\mathbf{a}^2} &= -k_I \mathbf{L}(\mathbf{w}) \pi_{\mathbf{a}^2} \end{aligned} \quad (7)$$

where the notation  $\mathbf{a}^2$  implies component-wise squaring, defining the vector with components  $a_i^2$ . The control gains  $k_g$ ,  $k_P$ , and  $k_I$  are required to be suitably larger than the control gains of the eigenvector estimator, so that the PI consensus subsystems converge much more rapidly.

Combining equations (4), (6), and (7), a distributed estimate of  $\lambda_2$  can be obtained by solving the equations:

$$\begin{aligned} \dot{a}_i &= -k_1 \pi_{\mathbf{a},i} + k_2 \sum_{j \in \mathcal{N}_i} w_{\{i,j\}} (a_j - a_i) - k_3 (\pi_{\mathbf{a}^2,i} - 1) a_i \\ \dot{\pi}_{\mathbf{a},i} &= k_g (a_i - \pi_{\mathbf{a},i}) + k_P \sum_{j \in \mathcal{N}_i} w_{\{i,j\}} (\pi_{\mathbf{a},j} - \pi_{\mathbf{a},i}) \\ &\quad - k_I \sum_{j \in \mathcal{N}_i} w_{\{i,j\}} (z_{\mathbf{a},j} - z_{\mathbf{a},i}) \\ \dot{z}_{\mathbf{a},i} &= k_I \sum_{j \in \mathcal{N}_i} w_{\{i,j\}} (\pi_{\mathbf{a},j} - \pi_{\mathbf{a},i}) \\ \dot{\pi}_{\mathbf{a}^2,i} &= k_g (a_i^2 - \pi_{\mathbf{a}^2,i}) + k_P \sum_{j \in \mathcal{N}_i} w_{\{i,j\}} (\pi_{\mathbf{a}^2,j} - \pi_{\mathbf{a}^2,i}) \\ &\quad - k_I \sum_{j \in \mathcal{N}_i} w_{\{i,j\}} (z_{\mathbf{a}^2,j} - z_{\mathbf{a}^2,i}) \\ \dot{z}_{\mathbf{a}^2,i} &= k_I \sum_{j \in \mathcal{N}_i} w_{\{i,j\}} (\pi_{\mathbf{a}^2,j} - \pi_{\mathbf{a}^2,i}) \end{aligned} \quad (8)$$

where  $\mathcal{N}_i$  is defined as the set of neighbors of node  $i$ . Thus, the computational and memory load of the estimator grows linearly with the number of neighbors, not the size of the network. Each node may then make a local estimate of the

algebraic connectivity,  $\widehat{\lambda}_2^{(i)}$ , as in (5):

$$\widehat{\lambda}_2^{(i)} = \frac{k_3}{k_2} (1 - \pi_{a^2, i}) \quad (9)$$

Finally, each node can estimate the partial derivative of the algebraic connectivity with respect to the weights of the edges that connect it to each of its neighbors [15] as:

$$\frac{\partial \widehat{\lambda}_2^{(i)}}{\partial w_{\{i, j\}}} = \frac{(a_i - a_j)^2}{n \pi_{a^2, i}} \quad (10)$$

*Example 1:* We apply the techniques to controlling a network of autonomous mobile robots so that a formation with high algebraic connectivity emerges [10]. Each robot  $i$  in the network is located at a position  $\mathbf{x}_i$ , with the collection of all robots positions being denoted  $\mathbf{x}$ . The strength of communication between two neighboring robots  $i$  and  $j$  is represented by the edge weight  $w_{\{i, j\}}$ , which is a non-increasing function of the distance between robots.

We aim to maximize the algebraic connectivity of the state-dependent weighted graph Laplacian, whilst ensuring that the network remains reasonably spaced and avoids collisions, by enforcing an upper bound on the weighted degree of each node, which is simply the diagonal elements of the graph Laplacian  $l_{i, i}$ . Specifically we tackle the non-convex optimization problem:

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && -\lambda_2(\mathbf{L}(\mathbf{w}(\mathbf{x}))) \\ & \text{subject to} && l_{i, i}(\mathbf{x}) \leq 1 \quad \forall i \in 1, \dots, n \end{aligned} \quad (11)$$

Unlike the convex problem of maximising  $\lambda_2$  over  $\mathbf{w}$  [8], this formulation is non-convex due to the fact that edge weights are now functions of the positions  $\mathbf{x}_i$  of the robots. In this example we choose to model this relation by an inverse-square law reflecting the intensity of signal strength of a radio transmitter, assuming that the antenna is radiating in all directions equally in three dimensions,

$$w_{\{i, j\}} = \sigma(\mathbf{x}) = \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2} \quad (12)$$

Other monotonic non-increasing weight functions may be used, for example sigmoids, ramps, and hyperbolae [12], [10]. One advantage of using an inverse-square law is that the edge weight grows unboundedly as the distance between neighboring robots tends to zero. As the maximum weighted degree of each robot is bounded, this provides an effective action for preventing collisions.

The distributed weight adaptation law is formulated according to Equations (3,a) and (3,b), using local estimates of global functions, and using the chain rule as the objective function is now a function  $f(\mathbf{w}(\mathbf{x}))$ :

$$\dot{\mathbf{x}}_i = k_w \sum_{j \in \mathcal{N}_i} \left( \frac{\partial \widehat{\lambda}_2^{(i)}}{\partial w_{\{i, j\}}} - \nu_i - \nu_j \right) \frac{\partial w_{\{i, j\}}}{\partial \mathbf{x}_i} \quad (13)$$

$$\dot{\nu}_i = (l_{i, i} - 1)\nu_i$$

On top of the 5 ODEs from the algebraic connectivity estimator, each robot now follows a further ODE for the maximum weighted degree boundary condition, and a further  $d$  ODEs for controlling position, where  $d$  is the dimension

of the space in which the robots are controlled. Thus, in our example where robots move in the plane, each robot follows a system of 8 coupled ODEs. Crucially, this remains fixed even for arbitrarily large networks, and the number of variables in each ODE scales linearly with the number of neighbors of each node. Therefore, the memory and computational requirements of each robot in the network scales well even for large networks. The initial network shown in Figure 3 evolves according to this networked system of ODEs and results in a locally optimal formation. As the robots' positions change over time, the strength of the edge weights change, resulting in a dynamically weighted graph Laplacian  $\mathbf{L}(t)$ . The eigenvalues of this dynamic network are shown in Figure 2, clearly showing an increase in the algebraic connectivity over time.

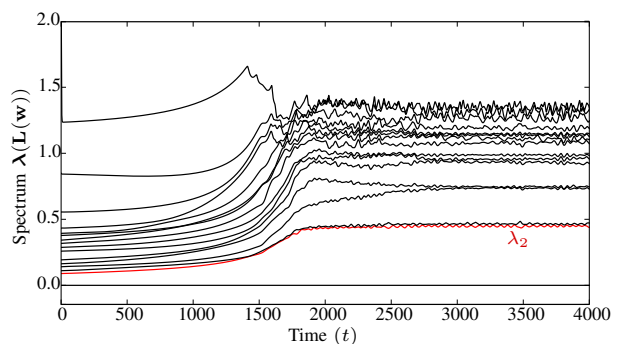


Fig. 2. The spectrum of the weighted graph Laplacian as the network evolves in time. It can be seen that the algebraic connectivity (highlighted in red) is increased over time, however, in the limit, the network system falls into a persistent oscillation as  $\lambda_2 = \lambda_3$  at the locally optimal positions.

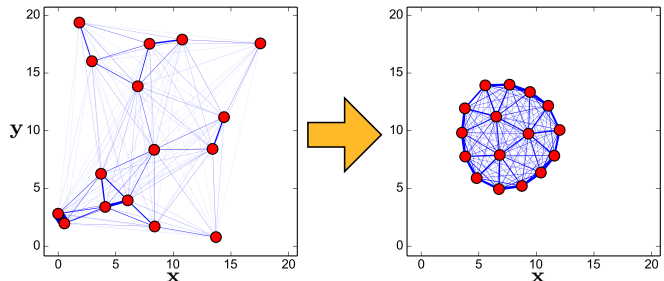


Fig. 3. The network adapts from the initial positions  $(x_i, y_i)$  of 16 agents, chosen uniformly at random in the interval  $([0, 20], [0, 20])$  (left). A complete communication graph is chosen, with edge weights determined by the Euclidean distance between agents, and illustrated in the figure using the thickness of the edges. After a simulated 4000 seconds, the robots have arranged themselves into a locally optimal formation, maximising the value of  $\lambda_2(\mathbf{w}(\mathbf{x}, \mathbf{y}))$ . For this specific problem, the robots have arranged themselves into a ring of 13, with a second ring of 3 located inside.

## B. Largest Eigenvalue $\lambda_n$ and the Eigenratio $\lambda_n/\lambda_2$

The largest eigenvalue of  $\mathbf{L}(\mathbf{w})$  is equivalent to the spectral radius of  $\mathbf{L}(\mathbf{w})$ , and is important in the average consensus problem with equal time-delays of  $\tau$  [1], where it is required that  $\lambda_n < \frac{\pi}{2\tau}$ ,  $\lambda_2 > 0$  for the system,

$$\dot{\mathbf{x}}(t) = -\mathbf{L}(\mathbf{w})\mathbf{x}(t - \tau) \quad (14)$$

to globally asymptotically converge. This eigenvalue is also important in the discrete time linear consensus system,

$$\mathbf{x}(k + 1) = (\mathbf{I} - h\mathbf{L}(\mathbf{w}))\mathbf{x}(k) \quad (15)$$

where convergence on the average value is determined by the condition  $\lambda_n < 2/h$ ,  $\lambda_2 > 0$ , so minimizing  $\lambda_n$  means that consensus will be reached for larger step size  $h$ . Moreover, the speed of convergence is determined by the function  $\max\{|1 - h\lambda_n|, |1 - h\lambda_2|\}$ , with smaller values resulting in faster convergence.

The eigenratio  $\lambda_n/\lambda_2$  is often referred to in the literature as the synchronizability [3], [23] of a network as it determines whether or not the synchronous solution in a class of network coupled oscillators may be stabilized for any coupling strength [4], [5]. A lower eigenratio then results in a locally, transversally stable synchronous solution for a wider range of coupling strengths [17].

Using a similar approach to the algebraic connectivity estimator, the largest eigenvalue  $\lambda_n$  can be estimated using an entirely distributed approach [24]. Specifically an estimate  $\mathbf{b}$  of the eigenvector associated with the largest eigenvalue can be obtained by solving the equation,

$$\dot{\mathbf{b}} = -k_1 \frac{\mathbf{1}\mathbf{1}^\top \mathbf{b}}{n} + k_2 \mathbf{L}\mathbf{b} - k_3 \left(1 - \frac{\mathbf{b}^\top \mathbf{b}}{n}\right) \mathbf{b} \quad (16)$$

Instead of converging under the action of  $\mathbf{L}(\mathbf{w})$  so that the slowest mode dominates, as happens in (4), the second term in the right hand side of (16) provides a diverging force so that the fastest mode dominates. Again, renormalization is achieved through the terms in (16) scaled by  $k_3$ , so that the estimator vector  $\mathbf{b}$  remains bounded. In this system, the vector  $\mathbf{b}$  converges onto  $\mathbf{b}^*$ , which has direction of the eigenvector  $\mathbf{v}_n$  associated with the eigenvalue  $\lambda_n$ , with magnitude such that:

$$\lambda_n = \frac{k_3}{k_2} \left( \frac{\mathbf{b}^{*\top} \mathbf{b}^*}{n} - 1 \right) \quad (17)$$

Again, two PI average consensus estimators are employed to completely distribute the system, so that for  $\lambda_n$  estimation each agent  $i$  follows the 5 ordinary differential equations:

$$\begin{aligned} \dot{b}_i &= -k_1 \pi_{\mathbf{b},i} - k_2 \sum_{j \in \mathcal{N}_i} w_{\{i,j\}} (b_j - b_i) - k_3 (\pi_{\mathbf{b}^2,i} - 1) b_i \\ \dot{\pi}_{\mathbf{b},i} &= k_g (b_i - \pi_{\mathbf{b},i}) + k_P \sum_{j \in \mathcal{N}_i} w_{\{i,j\}} (\pi_{\mathbf{b},j} - \pi_{\mathbf{b},i}) \\ &\quad - k_I \sum_{j \in \mathcal{N}_i} w_{\{i,j\}} (z_{\mathbf{b},j} - z_{\mathbf{b},i}) \\ \dot{z}_{\mathbf{b},i} &= k_I \sum_{j \in \mathcal{N}_i} w_{\{i,j\}} (\pi_{\mathbf{b},j} - \pi_{\mathbf{b},i}) \\ \dot{\pi}_{\mathbf{b}^2,i} &= k_g (b_i^2 - \pi_{\mathbf{b}^2,i}) + k_P \sum_{j \in \mathcal{N}_i} w_{\{i,j\}} (\pi_{\mathbf{b}^2,j} - \pi_{\mathbf{b}^2,i}) \\ &\quad - k_I \sum_{j \in \mathcal{N}_i} w_{\{i,j\}} (z_{\mathbf{b}^2,j} - z_{\mathbf{b}^2,i}) \\ \dot{z}_{\mathbf{b}^2,i} &= k_I \sum_{j \in \mathcal{N}_i} w_{\{i,j\}} (\pi_{\mathbf{b}^2,j} - \pi_{\mathbf{b}^2,i}) \end{aligned} \quad (18)$$

Agent  $i$  can then make a local estimate  $\widehat{\lambda}_n^{(i)}$  using the fact that the PI consensus estimator for  $\frac{\mathbf{b}^\top \mathbf{b}}{n}$  converges on  $\frac{\mathbf{b}^{*\top} \mathbf{b}^*}{n}$  as  $\mathbf{b} \rightarrow \mathbf{b}^*$ :

$$\widehat{\lambda}_n^{(i)} = \frac{k_3}{k_2} (\pi_{\mathbf{b}^2,i} - 1) \quad (19)$$

Agent  $i$ 's local estimate for the partial derivative of  $\lambda_n$  with respect to the edge weights of edges that connect  $i$  to each of its neighbors  $j \in \mathcal{N}_i$  can be made [24]:

$$\frac{\widehat{\partial \lambda_n}^{(i)}}{\partial w_{\{i,j\}}} = \frac{(b_i - b_j)^2}{n \pi_{\mathbf{b}^2,i}} \quad (20)$$

which is available to both parent nodes  $i$  and  $j$  of the edge. If both the  $\lambda_2$  and  $\lambda_n$  estimators are run concurrently so that each agent follows a dynamical system of 10 ordinary differential equations, it is simple to infer a local estimate of the eigenratio  $\lambda_n/\lambda_2$ , and its gradient with respect to each neighboring edge using the quotient rule:

$$\frac{\widehat{\partial \lambda_n / \lambda_2}^{(i)}}{\partial w_{\{i,j\}}} = \frac{\widehat{\lambda}_2^{(i)} \frac{\widehat{\partial \lambda_n}^{(i)}}{\partial w_{\{i,j\}}} - \widehat{\lambda}_n^{(i)} \frac{\widehat{\partial \lambda_2}^{(i)}}{\partial w_{\{i,j\}}}}{\left(\widehat{\lambda}_n^{(i)}\right)^2} \quad (21)$$

*Example 2:* We take the problem of minimising the spectral radius of the graph Laplacian  $\lambda_n$  so that simple consensus with uniform time delays will converge for the largest time delay [1]. Connectivity of the network is guaranteed by enforcing a positive lower bound on the algebraic connectivity. In this example we arbitrarily choose a lower bound  $\lambda_2 \geq 2$  and solve the optimization problem:

$$\begin{aligned} &\underset{\mathbf{w}}{\text{minimize}} && \lambda_n(\mathbf{L}(\mathbf{w})) \\ &\text{subject to} && \lambda_2(\mathbf{L}(\mathbf{w})) \geq 2 \end{aligned} \quad (22)$$

It can be seen that the objective function is convex, as is the inequality constraint, so that the first order necessary KKT conditions are also sufficient for the local optimum to be the global optimum.

Each node follows both the  $\lambda_2$  and  $\lambda_n$  estimators (Equations (8) and (18)) so that each node can infer the partial derivatives of these functions with respect to the weights of each neighboring edge. Then weight adaptation is accomplished according to (3) as:

$$\begin{aligned} \dot{w}_{\{i,j\}} &= \frac{k_w}{2} \sum_{k \in \{i,j\}} \left( \nu_k \frac{\widehat{\partial \lambda_2}^{(k)}}{\partial w_{\{i,j\}}} - \frac{\widehat{\partial \lambda_n}^{(k)}}{\partial w_{\{i,j\}}} \right) \\ \dot{\nu}_i &= (2 - \widehat{\lambda}_2^{(i)}) \nu_i \end{aligned} \quad (23)$$

Each node maintains an estimate of the dual variable for the optimization problem  $\nu_i$ , and each edge  $\{i, j\}$  adapts its weight according to the average of its parent nodes  $i$  and  $j$  estimated gradients and dual variables.

Starting from the randomly chosen network, shown inset in Figure 4, with  $n = 16$  nodes and  $m = 47$  edges each of initial weight  $w_{\{i,j\}}(0) = 1$ , each node follows the set of 11 ODEs defined by Equations (8), (18) and (23) while communicating with its set of neighbors, so that each edge weight follows the one ODE defined in Equation (23). The results of the distributed optimization are shown in Figure 4, displaying the trajectories of the Laplacian eigenvalues.

### C. Total Effective Graph Resistance

We focus next on optimising a spectral function which depends on all  $n - 1$  non-trivial eigenvalues of the Laplacian

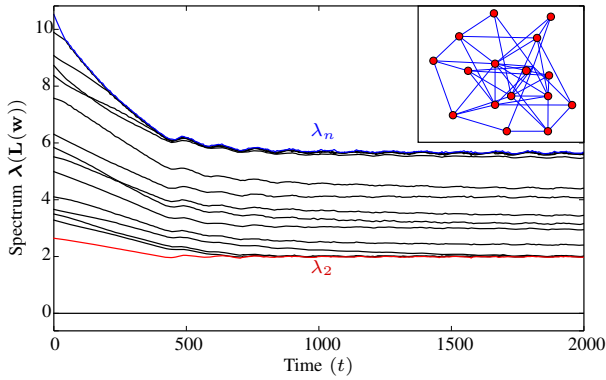


Fig. 4. Edge weights are adapted using a totally distributed method to minimize  $\lambda_n$  (highlighted in blue) whilst maintaining the constraint  $\lambda_2 \geq 2$ . The initial network (inset) has  $\lambda_n \approx 10.52$  and  $\lambda_2 \approx 2.64$  with unit edge weights. At  $t = 2000$  the network has decreased its spectral radius to  $\lambda_n \approx 5.66$ , whilst maintaining an algebraic connectivity of  $\lambda_2 \approx 1.99$ . Note that this is slightly in the infeasible region due to oscillation about the solution.

matrix, the total effective graph resistance  $\Omega$ , given by:

$$\Omega(\mathbf{w}) = n \sum_{i=2}^n \frac{1}{\lambda_i(\mathbf{w})} \quad (24)$$

We could estimate all  $n - 1$  eigenvalues by successive deflation. For example, we could estimate  $\lambda_3$  in a similar manner to the algebraic connectivity estimator, but deflating on the consensus mode  $\mathbf{v}_1$  and also on  $\mathbf{v}_2$ , using the estimate for  $\mathbf{v}_2$  from Equation (8). This process could be continued for all  $n - 1$  eigenvalues, but would rapidly become unwieldy for even modest sized networks.

Thus, if we want to provide a distributed estimator with a relatively few number of local variables at each node (that does not grow as the network becomes arbitrarily large) another method must be employed. A solution is to excite all modes with equal energy using white noise. We assume that each agent has access to a source of white noise. This is simple in the discrete time implementation where we approximate the Ito integral of the Stochastic Differential Equation (SDE) using the Milstein method [25]. Then we need only assume that each agent has access to their own normally distributed pseudo-random number generator, available in the vast majority of programming languages.

The SDE for making a distributed estimate of the total effective graph resistance can be formulated using the result that the linear consensus system with additive white noise of intensity  $\sigma$  has expected variance at long time, proportional to the total effective graph resistance [6]. As such we introduce the SDE [26]:

$$\begin{aligned} d\mathbf{a} &= (-k_1\boldsymbol{\pi}_a - k_2\mathbf{L}\mathbf{a})dt + d\mathbf{W} \\ d\boldsymbol{\pi}_a &= (k_\gamma(\mathbf{a} - \boldsymbol{\pi}_a) - k_P\mathbf{L}\boldsymbol{\pi}_a + k_I\mathbf{L}\mathbf{z}_a)dt \\ dz_a &= -k_I\mathbf{L}\boldsymbol{\pi}_a dt \\ d\boldsymbol{\pi}_{a^2} &= (k_\gamma(\mathbf{a}^2 - \boldsymbol{\pi}_{a^2}) - k_P\mathbf{L}\boldsymbol{\pi}_{a^2} + k_I\mathbf{L}\mathbf{z}_{a^2})dt \\ dz_{a^2} &= -k_I\mathbf{L}\boldsymbol{\pi}_{a^2} dt \\ dy &= k_L(\boldsymbol{\pi}_{a^2} - \mathbf{y})dt \end{aligned} \quad (25)$$

This has a very similar form to the systems used for the

distributed estimation of  $\lambda_2$  and  $\lambda_n$ , but now all modes are excited with equal energy using additive white noise ( $d\mathbf{W}$  is a vector of length  $n$  of independent Wiener processes of intensity 1).

The second PI consensus estimator  $\boldsymbol{\pi}_{a^2}, \mathbf{z}_{a^2}$  is used to estimate the variance of the vector  $\mathbf{a}$  (it has expected mean of 0 due to deflation on the consensus mode), and this is fed into a simple first order low pass filter with time constant  $\frac{1}{k_L}$  to remove some of the noise. Each node can then make a distributed estimate of the total effective graph resistance:

$$\widehat{\Omega}^{(i)} = 2k_2n \left( ny_i - \frac{1}{2k_1} \right) \quad (26)$$

To estimate the partial derivatives of the total effective graph resistance, an extension to the system (25) is required. Specifically, the following further equations need to be added to (25). For further details on this extended system see [26].

$$\begin{aligned} d\mathbf{b} &= (-k_1\boldsymbol{\pi}_b - k_2\mathbf{L}\mathbf{c})dt + d\mathbf{W} \\ d\mathbf{c} &= k_\gamma(\mathbf{L}\mathbf{b} - \mathbf{c})dt \\ d\boldsymbol{\pi}_b &= (k_\gamma(\mathbf{b} - \boldsymbol{\pi}_b) - k_P\mathbf{L}\boldsymbol{\pi}_b + k_I\mathbf{L}\mathbf{z}_b)dt \\ dz_b &= -k_I\mathbf{L}\boldsymbol{\pi}_b dt \end{aligned} \quad (27)$$

Each node is now able to make a low-pass filtered estimate of the partial derivative of the total effective graph resistance by means of the equations:

$$\begin{aligned} \dot{q}_{\{i,j\}} &= k_L((b_i - b_j)^2 - q_{\{i,j\}}) \\ \frac{\widehat{\partial\Omega}^{(i)}}{\partial w_{\{i,j\}}} &= -2k_2q_{\{i,j\}} \end{aligned} \quad (28)$$

*Example 3:* We now use these two stochastic systems (25) and (27) with the filtered gradient estimation (28) to illustrate a set point control problem. Given an arbitrary connected undirected weighted network, we wish to devise a distributed weight control law that will drive the total effective resistance of the network  $\Omega$  to a desired value. In this example, we choose to drive  $\Omega$  to 10, formulating the unconstrained minimization problem:

$$\underset{\mathbf{w}}{\text{minimize}} \quad (\Omega(\mathbf{L}(\mathbf{w})) - 10)^2 \quad (29)$$

which will most likely have multiple solutions. Nevertheless we can formulate the distributed weight control law:

$$\dot{w}_{\{i,j\}} = - \sum_{k \in \{i,j\}} \left( \widehat{\Omega}^{(k)} - 10 \right) \frac{\widehat{\partial\Omega}^{(k)}}{\partial w_{\{i,j\}}} \quad (30)$$

Figure 5 shows the edge weights in a randomly chosen network of  $n = 8$  nodes and  $m = 14$  edges, adapting under the control law to drive the total effective resistance of the network to the desired value of 10.

An important feature of this system to note is that edge weights never settle to fixed values. This is due to the inherently stochastic nature of the control law, and the persistent excitation of edges. Moreover, the optimization problem described in (29) will have multiple solutions, and so edge weights may converge to a neutrally stable manifold. Nevertheless, the weight control law results in a network whose total graph resistance is driven towards the desired



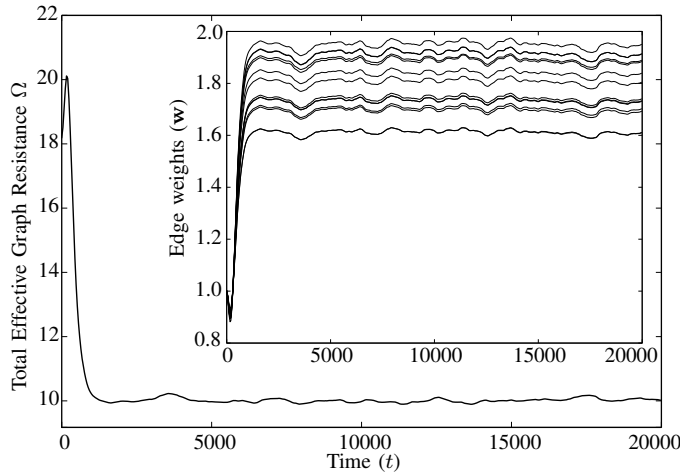


Fig. 5. The resultant total effective graph resistance is controlled towards a desired value of 10. Some stochasticity remains in the system, but the expected value of  $\Omega$  tends towards the desired value as  $t \rightarrow \infty$ .

value, and stays close to it for all time.

#### IV. CONCLUDING REMARKS

We have formulated a generic weight adaptation law for constrained, continuous optimization, which, when combined with the distributed estimators, can be utilized to find optima of several common network optimization problems in a wholly distributed manner. These we demonstrated through the use of three example problems: formation control to maximize algebraic connectivity, maximising the allowable delay in linear consensus whilst maintaining a minimum connectivity, and control of a network to a desired total effective graph resistance.

A common theme in all of these estimators is the separation of time scales. It is required that the weight adaptation happens on a slower time scale than the estimation system so that the estimators can be assumed to have converged. Likewise, the estimators rely on PI average consensus to infer quantities proportional to the mean or 2-norm, and these need to converge faster than the estimators. When eigenvalues are not distinct, a small change in the edge weights may result in a rapid change in the gradient of the spectral function as eigenvalues change order. The estimator systems will need some time to reconverge on the new gradient, and in this time the weight adaptation will overshoot. This sets up a limit cycle around the optimum if at the optimum the eigenvalues are not distinct. The size and frequency of this limit cycle can be controlled through the size of the separation in time-scales of the subsystems. The theoretical analysis of convergence and stability of the method presented in this paper can be obtained by using singular perturbation techniques and will be presented elsewhere.

#### REFERENCES

- [1] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, 2004.
- [2] R. Olfati-Saber, A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, 2007.
- [3] F. Comellas and S. Gago, "Synchronizability of complex networks," *J. Phys. A*, 2007.
- [4] L. Huang, Q. Chen, Y.-C. Lai, and L. M. Pecora, "Generic behavior of master-stability functions in coupled nonlinear dynamical systems," *Phys. Rev. E*, 2009.
- [5] L. M. Pecora and T. L. Carroll, "Master stability functions for synchronized coupled systems," *Phys. Rev. Lett.*, 1998.
- [6] G. F. Young, L. Scardovi, and N. E. Leonard, "Robustness of noisy consensus dynamics with directed communication," in *Proc. Am. Control Conf.*, IEEE, 2010.
- [7] A. Ghosh, S. Boyd, and A. Saberi, "Minimizing effective resistance of a graph," *SIAM Rev.*, 2008.
- [8] S. Boyd, "Convex optimization of graph Laplacian eigenvalues," in *Proc. Intern. Congr. Math.*, 2006.
- [9] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian," *IEEE Trans. Autom. Control*, 2006.
- [10] A. Simonetto, T. Keviczky, and R. Babuška, "Constrained distributed algebraic connectivity maximization in robotic networks," *Automatica*, 2013.
- [11] M. C. De Gennaro and A. Jadbabaie, "Decentralized control of connectivity for multi-agent systems," in *Proc. 45th IEEE Conf. Decis. Control*, IEEE, 2006.
- [12] B. M. M. Zavlanos, M. B. Egerstedt, and G. J. Pappas, "Graph-theoretic connectivity control of mobile robot networks," *Proc. IEEE*, 2011.
- [13] M. M. Zavlanos and G. J. Pappas, "Distributed connectivity control of mobile networks," *IEEE Trans. Robot.*, 2008.
- [14] M. Schuresko and J. Cortés, "Distributed motion constraints for algebraic connectivity of robotic networks," *J. Intell. Robot Syst.*, 2009.
- [15] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity for mobile sensor networks," *Automatica*, 2010.
- [16] L. Kempton, G. Herrmann, and M. di Bernardo, "Adaptive weight selection for optimal consensus performance," in *Proc. 53rd IEEE Conf. Decis. Control*, IEEE, 2014.
- [17] L. Donetti, F. Neri, and M. A. Munoz, "Optimal network topologies: expanders, cages, ramanujan graphs, entangled networks and all that," *J. Stat. Mech. Theor. Exp.*, 2006.
- [18] T. E. Gorochoowski, M. di Bernardo, and C. S. Grierson, "Evolving enhanced topologies for the synchronization of dynamical complex networks," *Phys. Rev. E*, 2010.
- [19] T. Liu, D. J. Hill, and J. Zhao, "Synchronization of dynamical networks by network control," *IEEE Trans. Autom. Control*, 2012.
- [20] A. Bertrand and M. Moonen, "Topology-aware distributed adaptation of laplacian weights for in-network averaging," in *21st Eur. Signal Process. Conf.*
- [21] R. A. Freeman, P. Yang, and K. M. Lynch, "Stability and convergence properties of dynamic average consensus estimators," in *Proc. 45th IEEE Conf. Decis. Control*, IEEE, 2006.
- [22] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [23] G. Chen and Z. Duan, "Network synchronizability analysis: a graph-theoretic approach," *Chaos*, 2008.
- [24] L. Kempton, G. Herrmann, and M. di Bernardo, "Self-organization of weighted networks for optimal synchronizability," *arXiv preprint arXiv:1505.07279*, 2015.
- [25] G. Mil'shtein, "Approximate integration of stochastic differential equations," *Theory Probab. Appl.*, 1975.
- [26] L. Kempton, G. Herrmann, and M. di Bernardo, "Distributed estimation and control of the total effective graph resistance," 2016. In preparation.