International Journal of Control
Vol. 82, No. 11, July 2009, 2095–2114

ISSN: 0020-7179 print/ISSN 1366-5820 online

DOI: 10.1080/00207170902912056
http://www.informaworld.com

International Journal of Control
Vol. 82, No. 11, July 2009, 2095–2114

RESEARCH ARTICLE

A Framework for Decentralised Feedback Connectivity Control with Application to Sensor Networks

F. Knorn[a*], R. Stanojevic[a], M. Corless[b], and R. Shorten[a]

[a]*Hamilton Institute, National University of Ireland Maynooth, Maynooth, Co. Kildare, Ireland*;
[b]*School of Aeronautics & Astronautics, Purdue University, West Lafayette, IN 47907, USA*

(*Received 21 October 2008; final version received 20 March 2009*)

In this paper we propose a decentralised algorithm for connectivity maintenance in a distributed sensor network. Our algorithm uses the dynamics of a consensus algorithm to estimate the connectivity of a network topology in a decentralised manner. These estimates are then used to inform a decentralised control algorithm that regulates the network connectivity to some desired level. Under certain realistic assumptions we show the closed loop dynamics can be described as a consensus algorithm with an input, and eventually reduces to a scalar system. Bounds are given to ensure the stability of the algorithm and examples are given to illustrate the efficacy of the proposed algorithm.

Keywords: Topology control; distributed averaging; consensus algorithms with inputs; rate of convergence

1 Introduction

Recent years have witnessed a growing interest in the control community in problems that arise when dynamic systems evolve over graphs. While the most high profile of these applications are in consensus applications such as formation flying and synchronisation problems Jadbabaie et al. (2003), Olfati-Saber and Murray (2004), Zavlanos and Pappas (2005), Blondel et al. (2005), Moreau (2005), and in wireless and sensor networks Wattenhofer et al. (2001), Cabrera et al. (2007), many other applications have arisen where the manner in which the network topologies change affect the performance of algorithms over which they evolve. In such applications, an essential requirement is that the topology of the graph be such that some basic properties required to support communication and control are satisfied, the most basic of these being that the network be connected. Considerations of this kind have given rise to the emerging field of network topology control.

Roughly speaking, a graph is (strongly) connected if there exists at least one communication path from any one network node to another. Clearly, graph connectivity is an essential component in situations where a group of network nodes must work together, in a decentralised manner, to achieve some global task. This issue of graph connectivity is therefore very important and has achieved much attention in various contexts recently. It appears that this work has followed three lines of enquiry. In the graph theory literature, attempts have been made to identify and grow graphs with pre-specified connectivity properties; see Fallat and Kirkland (1998), Ghosh and Boyd (2006), Boyd et al. (2004), and the references therein for an overview of this work. In the computer science and networking communities several attempts have been also made to identify local (node based) constraints to guarantee certain forms of graph connectivity. For example, the sector rule proposed by Wattenhofer *et al.*, is one such rule that gives rise to certain

*Corresponding author. Email: florian@knorn.org

ISSN: 0020-7179 print/ISSN 1366-5820 online

DOI: 10.1080/00207170902912056
http://www.informaworld.com

types of connected graphs Wattenhofer et al. (2001). Recently, work in this direction has been extended to reflect not only topological considerations, but also the effect of physical constraints such as power and interference, in achieving these objectives. Finally, a third strand of work has recently emerged in the control and robotics community. Roughly speaking, this work involved using feedback principles achieving graphs with a desired topology. Examples of this work can be found in Ramanathan and Rosales-Hain (2000), Cabrera et al. (2007), Gennaro and Jadbabaie (2006), Dimarogonas and Kyriakopoulos (2008), Ji and Egerstedt (2005) and the references therein. In particular, Gennaro and Jadbabaie have proposed a different approach to distributed control of the second smallest eigenvalue of the graph's Laplacian, Gennaro and Jadbabaie (2006). In their work however, nodes have a fixed communication radius, but change positions relative to each other to achieve a desired connectivity.

The work described in this document belongs to this latter category. Regulation of the connectivity of a given graph is difficult because graph connectivity is a global property, whereas typically, nodes (or agents) can only act locally. Thus, any algorithm for maintaining graph connectivity must be decentralised if it is to be of any practical value. Our objective here is to propose one such algorithm; namely, a decentralised algorithm that is simple to implement yet regulates the connectivity of a given graph to some specified level. Our proposal, which makes use of the dynamics of a consensus algorithm consists of two parts. First we propose a decentralised estimation scheme whereby each node can estimate the level of graph connectivity based on only locally available measurements. We then propose a control strategy to regulate graph connectivity about a specified set-point. Importantly, under mild assumptions, our strategy is provably stable, converges to the set-point and is robust to sensor or node failure. Furthermore, as we shall see, our control framework is sufficiently general to allow other constraints such as local power, interference, or node density to be part of a connectivity/interference tradeoff.

The work described in this document differs from that in the literature in a number of aspects. Firstly, some of the previous results are of a probabilistic nature, i.e. they draw statistical conclusions of the type "in average, roughly every third graph of this kind should be connected". However, the application scenario that we have in mind consists of a concrete situation where a number of sensors are placed randomly in space (for example, a set of nodes dropped over a lake, each node communicating only with a subset of its neighbours). In this case, drawing probabilistic conclusions is of little help, as we would like to find results for this particular instance of the problem. We are also interested in situations where information mixes quickly across the graph. Thus, we must specifically account for the speed at which information passing takes place and not just that the graph is connected. Finally, we wish to develop algorithms that can be used irrespective of graph type; in other words we wish to break free of the assumption that the underlying graph structure is symmetric. This delineates our work from much of the recent work in the area (i. e. de Cabrera, Gennaro, Boyd) where much of the underlying mathematical machinery is based upon contructs that are unique to symmetric matrices such as the Raleigh quotient. Finally, we argue that our algorithms are very simple to implement and require minimal computational requirements, and give rise to graph growing techniques with truly scale-free properties.

2 Basic idea

Our basic idea for connectivity estimation is based on the observation that dynamic systems evolving on graphs often reveal topological properties about the graph itself. One such algorithm is a consensus algorithm. Consensus, or distributed averaging algorithms, have been the subject of a inordinate amount of attention in the past decade, arising in applications such as distributed sensing, clock synchronisation, flocking, and in fusion of Kalman filter data; see for instance Reynolds (1987), Vicsek et al. (1995), Estrin et al. (2001), Jadbabaie et al. (2003). Distributed averaging is strongly related to the theory of Markov chains, and to (non)homogeneous matrix products. While the primary focus of the present paper is not on the dynamics of consensus algorithms, it is important to note here that the second eigenvalue in magnitude of the averaging matrix (see notation section below) determines the rate at which the nodes in the network achieve consensus. As a graph becomes less connected this second eigenvalue becomes closer to unity (when rate of convergence is used as a measure of connectivity). Further, as we shall see, a simple algorithm can be used, together with elementary techniques from system identification, to locally estimate this eigenvalue in a decentralised manner.

Let us briefly illustrate these basic points in Figure 1. Here, we show the average value of

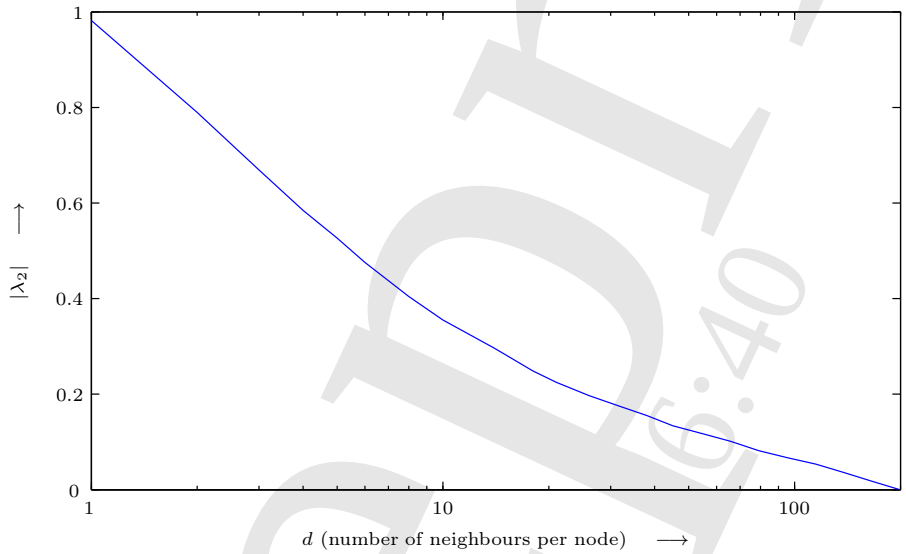

Figure 1. Average second largest eigenvalue of the unweighted averaging matrix of d-regular random graphs on 200 nodes.

the second largest eigenvalue in magnitude of the averaging matrix of random (regular) graphs.[1] The averaging matrix was constructed directly from a stochastic normalisation of the adjacency matrix of the underlying graph. In the plot, the value of the second largest eigenvalue drops monotonically with increasing regularity (number of neighbours per node). Although this is a very special type of graph, it shows that a single value can give an indication of the connectivity situation of a graph.

Comment: Classically, the second smallest eigenvalue of the *Laplacian* (or *transition Laplacian*) matrix of a graph has been used as an algebraic measure for connectivity, Fiedler (1973), Chung (1997). However, usually Laplacians are defined for symmetric graphs and this is an unnatural restriction that we would like to eliminate. In contrast, the second largest eigenvalue (in magnitude) of an averaging matrix is also an excellent candidate to indicate the degree of

[1] A d-regular graph is a graph where each node has exactly d neighbours (here chosen at random).

1: Deploy pre-configured nodes and initialise network by choosing random initial communication radii such that network is connected.
2: By running a consensus algorithm on the network, each node estimates the second largest eigenvalue of the averaging matrix based on the convergence of its own state.
3: For each node, if the estimated eigenvalue is smaller than some desired value, decrease the broadcast radius; if the estimate is larger, increase the radius.
4: Go to 2.

Figure 2. Pseudocode for connection of estimation and control scheme.

connectivity of an entire graph (whether the underlying graph is directed or not) with the added benefit of being available locally to each node using simple estimation techniques.

Knowledge of global information such as level of connectivity, based on purely *local* information, offers a wide range of local sensor actions with the objective of connectivity maintenance, one of which is presented here in this paper. For example, in wireless networks, a possible action is for nodes to adjust the power of their radio transmissions, based on the local estimate of connectivity. Concretely, this could mean to reduce the communication radius if the connectivity is estimated to be larger than required (as decreasing the radius will lead to reducing the number of neighbours, hence reducing connectivity). Pseudocode for such a strategy is given in Figure 2.

That such a strategy is well posed is evident and follows from the basic observation that if all nodes increase their communication radii sufficiently, then the graph will eventually become more densely connected. The issues that make the realisation of such strategies challenging in a practical environment concern decentralised estimation of the second largest eigenvalue of the averaging matrix, and proving that the resulting closed loop strategy is robustly stable. Resolving these issues are the main concern of this paper.

In the next section we introduce basic notation. We will then present our decentralised estimation scheme for the second largest eigenvalue. We discuss in Section 4 how this value could be used to control the networks connectivity by proposing a simple controller based on these estimates, and determine the conditions for the stability of the decentralised closed loop system. Results from simulations are then presented in Section 4.2. Finally conclusions and future directions are given in the last section.

3 Decentralised estimation of the second eigenvalue

3.1 *Notation*

Let $\mathscr{G} = (\mathcal{V}, \mathcal{A})$ be a static graph (directed or undirected) on n nodes, with vertex set $\mathcal{V} = \{1, \dots, n\}$ and edge set $\mathcal{A} \in \mathcal{V} \times \mathcal{V}$. We assume that $\mathscr{G}$ describes a set of sensors and the manner in which they communicate with each other; namely, a directed edge exists between node i and j if node (sensor) i can communicate directly to node j. For each node i, we let the set $\mathcal{N}_i$ consist of node i and the nodes which can communicate directly with i.

Further, we assume that a consensus (averaging) algorithm evolves on the graph $\mathscr{G}$. Formally, associate to each node $i = 1, \dots, n$ in the network a state $x_i \in \mathbb{R}$. The state of node i at time k is denoted $x_i(k)$, and the network's state (i. e. the states of all the nodes combined) is the column vector $\boldsymbol{x}(k) = \begin{pmatrix} x_1(k) & \dots & x_n(k) \end{pmatrix}^\mathsf{T}$. For each node $i = 1, \dots, n$, the iterative distributed averaging mentioned earlier can then be written as

$$x_i(k+1) = \sum_{j=1}^{n} \alpha_{ij} x_j(k) \qquad \text{where} \qquad \sum_{j=1}^{n} \alpha_{ij} = 1 \qquad \text{and} \qquad \begin{cases} \alpha_{ij} > 0 & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

for $k = 0, 1, 2, \dots$ with some initial condition $x_i(0) = x_{i0}$. This relation can be written for the overall network as

$$\boldsymbol{x}(k+1) = \boldsymbol{P}\boldsymbol{x}(k) \qquad \text{where} \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{2}$$

and where the stochastic, non-negative $\boldsymbol{P} = (\alpha_{ij})$ is called the *averaging matrix*.

Let $\lambda_1, \dots, \lambda_n$ be the eigenvalues of $\boldsymbol{P}$ and assume that they are ordered so that $|\lambda_i| \geq |\lambda_j|$ when $i \leq j$. To make matters more tractable we shall assume in the remainder of this paper that $\boldsymbol{P}$ is diagonalisable.[1] Further, by making this assumption we have that $\boldsymbol{P}$ has n linearly independent eigenvectors, $\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_n$ corresponding to the eigenvalues $\lambda_1, \dots, \lambda_n$. Thus these eigenvectors form a basis for $\mathbb{R}^n$ and every initial state $\boldsymbol{x}_0$ can be uniquely expressed as

$$\boldsymbol{x}_0 = c_1\boldsymbol{\nu}_1 + c_2\boldsymbol{\nu}_2 + \cdots + c_n\boldsymbol{\nu}_n$$

for some scalars $c_1, \dots, c_n$. Hence,

$$\begin{aligned} \boldsymbol{x}(k) &= \boldsymbol{P}^k\boldsymbol{x}_0 = \boldsymbol{P}^k(c_1\boldsymbol{\nu}_1 + c_2\boldsymbol{\nu}_2 + \cdots + c_n\boldsymbol{\nu}_n) \\ &= c_1\lambda_1^k\boldsymbol{\nu}_1 + c_2\lambda_2^k\boldsymbol{\nu}_2 + \cdots + c_n\lambda_n^k\boldsymbol{\nu}_n \end{aligned}$$

If the underlying graph is strongly connected, and since $\boldsymbol{P}$ has positive entries along the main diagonal, it follows that $\boldsymbol{P}$ is primitive, Horn and Johnson (1985). Thus, the Perron eigenvalue $\lambda_1 = 1$ is simple and all other eigenvalues are smaller in magnitude. Also, $\boldsymbol{\nu}_1 = \mathbf{1} := \begin{pmatrix} 1 & \dots & 1 \end{pmatrix}^\mathsf{T}$; hence

$$\boldsymbol{x}(k) = c_1\mathbf{1} + \lambda_2^k \left[\sum_{j=2}^{n} c_j \left(\frac{\lambda_j}{\lambda_2} \right)^k \boldsymbol{\nu}_j \right] \tag{3}$$

and $\|\boldsymbol{x}(k) - c_1\mathbf{1}\| \leq |\lambda_2|^k \beta(\boldsymbol{x}_0)$ where $\beta(\boldsymbol{x}_0) = \sum_{j=2}^{n} |c_j| \, \|\boldsymbol{\nu}_j\|$.

In this case, $\boldsymbol{x}(k)$ converges exponentially to $c_1\mathbf{1}$ and the rate of convergence is bounded by $|\lambda_2|$. In other words, the rate of convergence of the distributed averaging can be measured by the

[1] Since the set of diagonalisable matrices is dense in the set of stochastic matrices, this assumption is an entirely reasonable one to make.

1: $z_i(k) = x_i(k) - x_i(k-1)$
2: $A = \textbf{Estimate\_real}(z_i())$
3: $B = \textbf{RLS\_real}(z_i())$
4: $C = \textbf{Estimate\_complex}(z_i())$
5: **if** $|A - B| < \epsilon$
6: **return** A
7: **else**
8: **return** C
9: **end if**

Figure 3. Pseudocode for the overall estimation method of $|\lambda_2|$.

magnitude of λ_2. Together with the intuition that the more the graph is connected, the faster the averaging should converge, we can now see that $|\lambda_2|$ may very well be used as a proxy for the level of connectivity of the graph.

3.2 *Estimation*

We now provide a simple method by which all nodes in the network may estimate λ_2 based only on local measurements.

Our basic idea is as follows. Once we know whether λ_2 is real or complex, different methods can be used to accurately estimate its magnitude based only on local measurements. For example, when λ_2 is real then the direct estimation method described by Proposition 3.1 will yield a correct estimate of $|\lambda_2|$. Also, the dynamic system that governs the evolution of $z_i(k) := x_i(k) - x_i(k-1)$ can be modelled asymptotically as a first order linear system (with a noise term that decays to zero) if λ_2 is real valued. This linear system can be identified through an estimation method such as recursive least squares (RLS) providing another estimate of the absolute value of λ_2. When λ_2 is complex, another estimation method, based on Proposition 3.2 below can be applied. Thus with appropriate numerical conditioning of the values of $z_i(k)$, estimation of λ_2 can be carried out in a straightforward manner.

Usually (the exception being symmetric graphs) it is not clear *a priori* whether the averaging matrix $\boldsymbol{P}$ has a real or complex second eigenvalue. Thus we must develop a method for determining whether or not this eigenvalue is real or complex. We use three estimators, running in parallel, to achieve this. Specifically, we first obtain estimates for λ_2 from the estimator based on Proposition 3.1 as well as the recursive least squares scheme, both of which are guaranteed to work only when λ_2 is real. If both estimates of λ_2 match to a certain degree (that is, the absolute difference between the two values is less than some threshold ϵ), we assume that λ_2 is real and use these estimates. However, if the estimates do not match sufficiently, we consider λ_2 to be complex and use the estimate obtained based on the Proposition 3.2 (which is guaranteed to converge to the correct value in that case). The pseudocode for this strategy is given in Figure 3.

In the rest of the section we provide the details explaining what each of the functions **Estimate_real()**, **RLS_real()** and **Estimate_complex()** do. All three functions require the distributed averaging algorithm to be run on the network. Furthermore, each node is assumed to be able to store a small number of its own past states (but at least four).

3.2.1 Estimate_real()

The following proposition provides a method of estimating the value of the second largest eigenvalue of the averaging matrix when it is real valued.

Proposition 3.1 Decentralised estimation of real valued λ_2 **:**

Let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ be a static, strongly connected network with averaging matrix $\boldsymbol{P}$ such that its second largest eigenvalue in magnitude λ_2 is real and satisfies $|\lambda_2| > |\lambda_j|$ for all $j > 2$.

Consider any node i and let $z_i(k) := x_i(k) - x_i(k-1)$ where $\boldsymbol{x}(k)$ is determined by the distributed averaging algorithm (2) running on the network with a sufficiently general initial condition. Consider any positive integer m and for $k \geq m+1$, let

$$\tilde{\lambda}_{2i}(k) = \left| \frac{z_i(k)}{z_i(k-m)} \right|^{1/m} \tag{4}$$

be an estimate of $|\lambda_2|$. Then $\lim_{k\to\infty} \tilde{\lambda}_{2i}(k) = |\lambda_2|$.

Proof: See Appendix A.

Hence, if the prerequisites are met, for $k \geq m+1$, each node can calculate an estimate of $|\lambda_2|$ through (4) that converges to the true value as k grows.

Comment: It also follows from the proof that larger the gap between $|\lambda_2|$ and $|\lambda_3|$ the faster the estimates $\tilde{\lambda}_{2i}(k)$ will converge to the true value of $|\lambda_2|$.

3.2.2 RLS_real()

When λ_2 is real we can also use a recursive least squares algorithm for estimating λ_2. It can be seen from (A1) that by letting $m = 1$ we have for $k = 1, 2, \ldots$ the following relationship (asymptotically)

$$|z_i(k+1)| \simeq |\lambda_2| \cdot |z_i(k)| \tag{5}$$

Applying a suitably parametrised recursive least squares algorithm, see for instance Åström and Wittenmark (1997), should then also yield good estimates for $|\lambda_2|$.

3.2.3 Estimate_complex()

The next proposition provides a method for estimating the magnitude of a complex valued λ_2. When λ_2 is complex, its complex conjugate $\bar{\lambda}_2$ is also an eigenvalue of $\boldsymbol{P}$ with the same magnitude. If we assume that $|\lambda_2| > |\lambda_j|$ for all $j = 4, \ldots, n$ then, recalling (3), it is straightforward to show that, for each node i, the variable $z_i(k) = x_i(k) - x_i(k-1)$ can be written as

$$z_i(k) = c_i \lambda_2^k + \bar{c}_i \bar{\lambda}_2^k + |\lambda_2|^k O_i(k) \tag{6}$$

where $O_i(k) \to 0$ as $k \to \infty$ and $c_i, \bar{c}_i \neq 0$ for a sufficiently general initial condition of the averaging algorithm.

Proposition 3.2 Decentralised estimation of the magnitude of a complex λ_2 **:**

Let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ be a static, strongly connected network with averaging matrix $\boldsymbol{P}$ such that its second largest eigenvalue in magnitude λ_2 is complex with non-zero imaginary part and $|\lambda_2| = |\bar{\lambda}_2| > |\lambda_j|$ for $j > 2$.

Consider any node i and let

$$\zeta_i(k) := z_i(k) z_i(k-2) - z_i(k-1)^2 \tag{7}$$

where $z_i(k) = x_i(k) - x_i(k-1)$ and $\boldsymbol{x}(k)$ is determined by the distributed averaging algorithm (2) running on the network with a sufficiently general initial condition. Consider any positive integer m and for $k \geq m+3$, let

$$\tilde{\lambda}_{2i}(k) = \left| \frac{\zeta_i(k)}{\zeta_i(k-m)} \right|^{\frac{1}{2m}} \tag{8}$$

be an estimate of $|\lambda_2|$. *Then* $\lim_{k\to\infty} \tilde{\lambda}_{2i}(k) = |\lambda_2|$.

Proof: See Appendix B.

Based on this proposition, if λ_2 is complex and each node calculates an estimate of $|\lambda_2|$ through (8) then, the estimate will converge to the true value as k grows.

Comment: By its very nature, when running the consensus algorithm over a connected network, the states of all nodes will converge to a common value. In that case, the difference in states $z_i(k)$ will tend to zero. On the one hand, the calculation of the $z_i(k)$ will be less and less exact as the $z_i(k)$ approach zero, and on the other, when using the algorithms based on Propositions 3.1 and 3.2, the division of $z_i(k)$ by $z_i(k-m)$ resp. $\zeta_i(k)$ by $\zeta_i(k-m)$ will also become more and more problematic. It is, however, not too difficult to solve these problems. Simply whenever some node's state $x_i(k)$ agrees with all of its neighbours on the top s digits, it stops broadcasting those top s digits and keeps exchanging only the lower weight digits.

We must assume that in an actual implementation sufficiently exact numerical computations can be provided as the current approach does not take into account the inherently limited accuracy of numerical calculations in digital processors.

Comment: In this section we have assumed that there is a spectral gap between λ_2 (and its conjugates) and the remaining eigenvalues of the matrix $\boldsymbol{P}$. Since the set of matrices satisfying this property is dense in the set of stochastic matrices, this assumption is also entirely reasonable. However, the case where $|\lambda_2| = |\lambda_3| \geq |\lambda_4| \geq \ldots$ can also be accommodated in our framework by including more estimators, similar to the ones presented above, and by modifying the logic described in Fig. 3 accordingly. This is omitted here for ease of exposition, and because the aforementioned case is a low probability event.

3.3 *Results*

In the following two examples, we generated a two-dimensional random geometric graph with random connection radii for each node. These type of graphs are often used when modelling wireless networks, in particular wireless sensor networks, Penrose (2003), Santi (2005). A random *geometric graph* or *disc graph* is created as follows: Place n nodes uniformly distributed in the unit square, then interconnect the nodes based on the so-called *distance parameters* or *connection radii* of the nodes. That is, each node i has a parameter r_i based on which it connects (or "sends information") to other nodes that are closer than r_i from it: if some node j is at (Euclidian) distance d_{ij} from node i then there is an edge from node i to node j (i. e. node j is in reach) if and only if $d_{ij} \leq r_i$.

Figures 4 and 5 show the outputs of our three estimation schemes as well as their combination for two different situations: one where λ_2 is real, and one where λ_2 is complex. For each case we have plotted each nodes' estimates of $|\lambda_2|$ as a function of time (iterations of the estimation schemes), provided by the different algorithms, as well as the combination of them as proposed in Figure 3. From top to bottom, the subplots show the evolution of the estimates based on A) Proposition 1, B) recursive least squares and C) Proposition 2; as well as their combination in the last subplot. The true value of $|\lambda_2|$ is indicated by the thick horizontal line.

Comment: The following parameters were used. The random disc graphs on $n = 20$ nodes were created using connection radii r_i uniformly distributed in the interval $[\,0.1\,, 0.6\,]$. We used $m = 5$ in the algorithm based on Proposition 3.1, and $m = 1$ in that based on Proposition 3.2. The initial estimates of the recursive least squares algorithm was set to 0.5. Finally, the combination of the estimates was done using the threshold $\epsilon = 0.005$.

When the network has a real valued λ_2, it can be seen that the each node's estimates using the first two estimators converge quickly to the correct value. The estimates of the third estimator

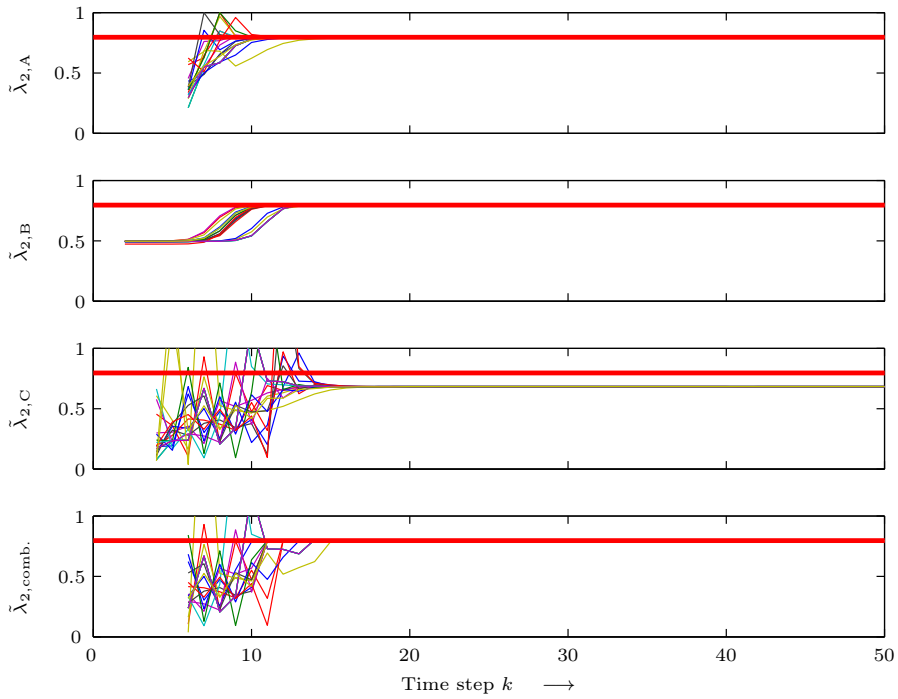

Figure 4. Comparison of the estimation schemes for real valued $\lambda_2 \simeq 0.80$.

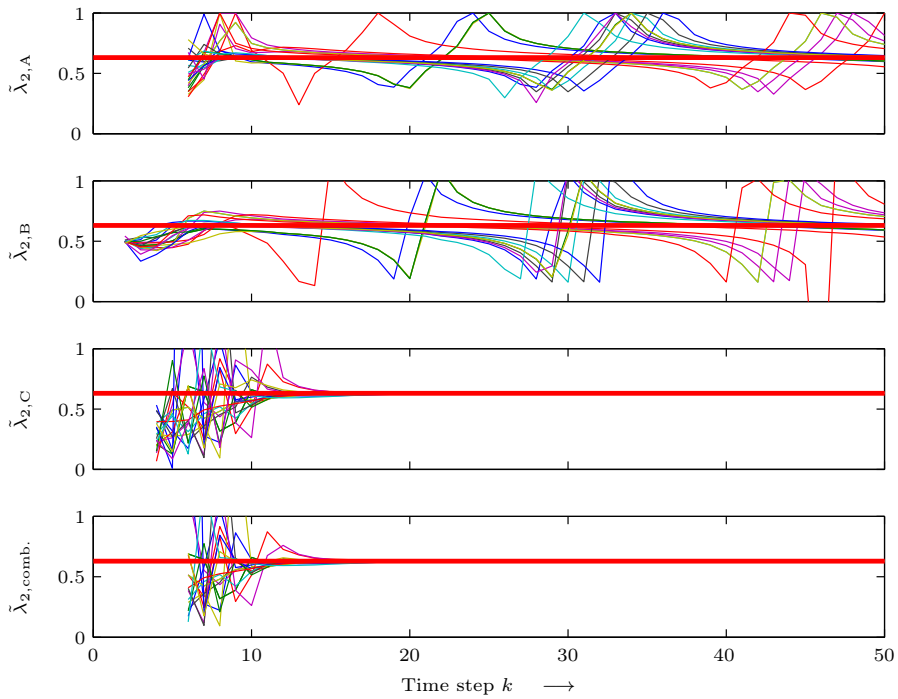

Figure 5. Comparison of the estimation schemes for complex $\lambda_2 \simeq 0.63 + 0.05\mathrm{i}$.

also converge, but to the wrong value. Since the two estimators targeted at a real valued λ_2 both converge to the same value, the error between them quickly both drops below the preset threshold, and the combination scheme correctly switches to returning the value of the first estimator.

In the complex case, Figure 5, the situation is different. Both the estimates of the estimators aimed at real valued λ_2 do not converge to the correct value of $|\lambda_2|$, but rather oscillate around it. The error between them is sufficiently large so that the combination scheme returns the value of the third estimator, which in turn now provides correct estimates.

4 Decentralised connectivity control

We now present our algorithm for decentralised connectivity control. Please note that, by an abuse of notation, we shall simply use λ in the remainder of the paper to refer to $|\lambda_2|$. As mentioned already, we wish to adjust the communication radius of each sensor in the network, $\{r_1, \ldots, r_n\}$ based on a local estimation of λ, with the ultimate objective of regulating λ to some neighbourhood of a target value; namely so that $\left|\lambda - \lambda^*\right| < \varepsilon$ for some $\lambda^* \in (0, 1)$ and $\varepsilon > 0$. Since we are trying to address situations in which individual sensors may fail resulting in a change in network connectivity, we are inherently dealing with situations where the network topology is slowly (but not constantly) changing. In what follows we therefore make the assumption of quasi-stationarity; specifically, we assume that the local estimators operate over very fast time scales when compared with the local control actions (local radius updates). This assumption greatly facilitates analytical tractability and makes our convergence proofs somewhat easier to develop. Finally, since there may be many sets of communication radii $\{r_1, \ldots, r_n\}$ that guarantee $\left|\lambda - \lambda^*\right| < \varepsilon$, we shall make additional assumptions to guarantee that the closed loop algorithm converges to a *unique* set of radii; namely, we seek a control action that guarantees convergence of all radii to the same value. We emphasise again that this assumption is made to facilitate analytical tractability, but it can also be motivated from a practical standpoint, where having all nodes use the same broadcast radius should contribute to similar battery lifetimes of the nodes. However, our framework is sufficiently general to allow other quantities of interest to be included (for instance, equal numbers of neighbours, maximum numbers of neighbours); although, the convergence proofs will change accordingly.

Our control algorithm is motivated by the following easily established results.

Theorem 4.1: *Let $\boldsymbol{P}(k) \in \mathbb{R}^{n\times n}$ be a sequence matrices taken from a finite set of primitive, row-stochastic matrices with strictly positive main diagonal entries, and $\theta\big(\boldsymbol{x}(k), k\big)$ a sequence of real numbers.*

If $\boldsymbol{x}(k) = \big(x_1(k) \ \ldots \ x_n(k)\big)^{\mathsf{T}}$ evolves for some $\boldsymbol{x}(0) = \boldsymbol{x}_0 \in \mathbb{R}^n$ according to

$$\boldsymbol{x}(k+1) = \boldsymbol{P}(k)\boldsymbol{x}(k) + \theta\big(\boldsymbol{x}(k), k\big)\mathbf{1} \tag{9}$$

where $\mathbf{1} = \big(1 \ \ldots \ 1\big)^{\mathsf{T}}$ then, the elements of $\boldsymbol{x}(k)$ will approach each other over time, that is

$$\lim_{k\to\infty} x_i(k) - x_j(k) = 0 \qquad \textit{for} \qquad i, j \in \{1, \ldots, n\}$$

Proof: See Appendix C.

In fact, this result can easily be extended to classes of nonlinear consensus operators using the recent results of Moreau (2005).

Theorem 4.2: *Let $\mathscr{G}(k) = \big(\mathcal{V}, \mathcal{A}(k)\big)$ be a sequence of strongly connected graphs, $\theta\big(\boldsymbol{x}(k), k\big)$ be a sequence of finite real numbers and f be a map on $\mathscr{G}(k)$ satisfying the following conditions. Associated to each directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ with node set $\mathcal{V} = \{1, \ldots, n\}$, each node $i \in \mathcal{V}$ and each state $\boldsymbol{x} \in \mathcal{X}^n$, there is a compact set $\mathcal{E}_i(\mathcal{A})(\boldsymbol{x}) \subset \mathcal{X}$ satisfying:*

(1) $f_i(\boldsymbol{x}, k) \in \mathcal{E}_i\big(\mathcal{A}(k)\big)(\boldsymbol{x}) \qquad \forall k \in \mathbb{N} \qquad \forall \boldsymbol{x} \in \mathcal{X}^n$,
(2) $\mathcal{E}_i(\mathcal{A})(\boldsymbol{x}) = \{x_i\}$ whenever the states of node i and its neighbouring nodes j are all equal,
(3) $\mathcal{E}_i(\mathcal{A})(\boldsymbol{x})$ is contained in the relative interior of the convex hull of the states of node i and its neighbouring nodes j whenever the states of node i and its neighbouring nodes j are not all equal,

(4) $\mathcal{E}_i(\mathcal{A})(\boldsymbol{x})$ *depends continuously on* $\boldsymbol{x}$*, that is, the set-valued function* $\mathcal{E}_i(\mathcal{A}) : \mathcal{X}^n \rightrightarrows \mathcal{X}$ *is continuous.*[1]

Then, if $\boldsymbol{x}(k) = \big(x_1(k) \ \dots \ x_n(k)\big)^\mathsf{T}$ *evolves for some* $\boldsymbol{x}(0) = \boldsymbol{x}_0$ *according to*

$$\boldsymbol{x}(k+1) = f\big(\boldsymbol{x}(k), k\big) + \theta\big(\boldsymbol{x}(k), k\big)\mathbf{1} \tag{10}$$

the elements of $\boldsymbol{x}(k)$ *will approach each other over time, that is*

$$\lim_{k\to\infty} x_i(k) - x_j(k) = 0 \qquad \textit{for} \qquad i, j \in \{1, \dots, n\}$$

Proof: See Appendix D.

In our context, both Theorems 4.1 and 4.2 are very useful. They indicate that consensus algorithms with an input term, that can depend on the consensus states, eventually become *scalar*. That is, their stability and convergence properties are eventually governed by the scalar equation

$$x(k+1) = x(k) + \theta\big(x(k), k\big)$$

Since the properties of such systems are well understood, the above theorems offer interesting possibilities for the design of control laws.

With this in mind we propose updating individual radii using a convex combination of their neighbours' values, plus an input term that depends on the estimated second largest eigenvalue. Specifically, we propose the following decentralised control law

$$\boldsymbol{r}(k+1) = \boldsymbol{P}_c(k)\boldsymbol{r}(k) + \eta\Big[\lambda\big(\boldsymbol{r}(k)\big) - \lambda^*\Big]\mathbf{1} \tag{11}$$

for some $\boldsymbol{r}(0) = \boldsymbol{r}_0$. Here $\boldsymbol{P}_c(k)$ is a sequence of primitive, row-stochastic averaging matrices on the graphs induced by $\boldsymbol{r}(k)$, $\lambda\big(\boldsymbol{r}\big)$ is the magnitude of the second largest eigenvalue of the averaging matrix $\boldsymbol{P}$ as in (2) for the graph induced by $\boldsymbol{r}$, and $\eta > 0$ is a suitable control gain. We are then guaranteed by Theorem 4.1 that the radii will converge to a common value.

Comment: Any other consensus scheme (to which Theorem 4.2 can be applied) may be used here. The proposed controller is decentralised in that each node only requires the radius information of its neighbours, information that can easily be broadcast along the communication that is necessary to run the consensus algorithm needed to estimate $\lambda(k)$ in the first place.

It remains to determine conditions on the control gain η so that $\lambda\big(\boldsymbol{r}(k)\big)$ converges to a desired neighbourhood of λ^*.

4.1 *Conditions for convergence of decentralised control*

As we have shown, it follows from the closed loop dynamics that we can assume at some stage that all radii have converged to a common value. In that case, (11) will be reduced to a scalar equation for the whole network:

$$r(k+1) = r(k) + \eta\Big[\lambda\big(r(k)\big) - \lambda^*\Big] \tag{12}$$

[1]Put simply, these four conditions require that the updated state of each node must be a strict convex combination of its own and its neighbours' states, and that the update function must be continuous.

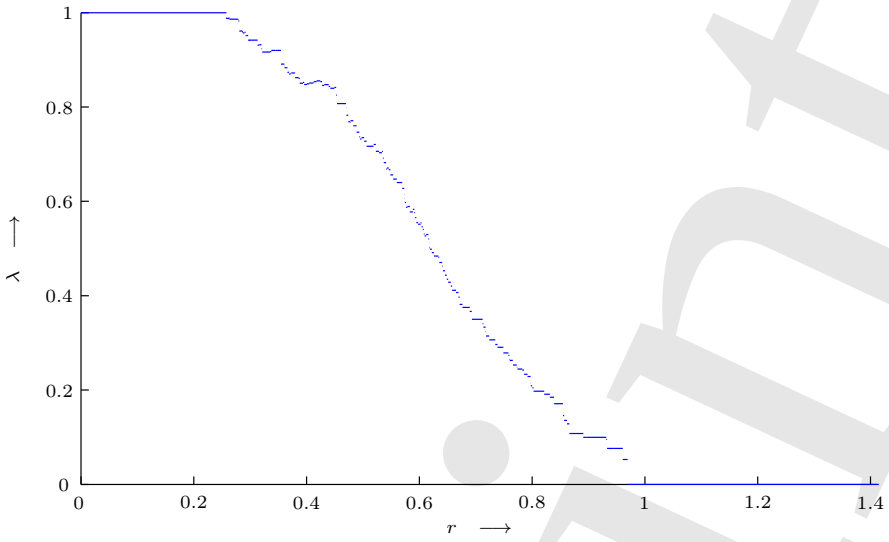

Figure 6. Plot of $\lambda(r)$, the magnitude of the second largest eigenvalue of the averaging matrix of a random (undirected) disc graph on 20 nodes as a function of the (common) communication radius r.

for some $r(0) = r_0$. Note that we write $\lambda\big(r(k)\big)$ since the second largest eigenvalue of the averaging matrix of the network depends on the communication radius used by the nodes. Ideally we would like to ensure that $\lambda\big(r(k)\big)$ asymptotically approaches λ^* under the assumption that the estimation part of the algorithm can be completely decoupled from the closed loop control. As we shall see, even under this considerable simplification, proving stability is nontrivial. In particular, two practical issues arise.

1. Quantisation. The first complication arises from the following observation. Normally, with problems of this type, one makes use of the fact that the eigenvalues of the consensus matrix vary as a continuous function of the matrix entries. In what we are proposing, the entries of $\boldsymbol{P}$ are either zero, or jump to some non-zero value as we adjust the communication radius of each node. In other words, the matrix entries vary abruptly as a result of the control action; consequently, the result of this is that $\lambda(r)$ also changes abruptly. Thus, it is clear that not every arbitrary second largest eigenvalue value in the $(\,0\,,1\,)$ interval can be achieved through feedback of the proposed type. Rather, the network can only produce a finite set of values, corresponding to the (limited number of) different possible topologies of the network with a fixed number of nodes in fixed locations. This fact is depicted in Figure 6. The plot shows how the magnitude of the second largest eigenvalue changes with the (common) communication radius for a given random disc graph on 20 nodes. Note that the curve is not continuous, but broken up into segments. A given magnitude of the second largest eigenvalue never corresponds to just a single radius, but a range of radii. Thus the best we can hope for is to converge to some neighbourhood of λ^*. Of course, for our application, this is entirely satisfactory as both connectivity and bounds on rates of information transmission in the network are controlled using this strategy.

2. Monotonicity. A second complication arises due to the fact that we do not precisely know the relationship between $\lambda(r)$ and r. In fact, the previous example shows that this relationship need not even be monotonic. However, it is reasonable to assume that the aforementioned relationship is approximately monotonic. This follows from the following argument. Our strategy is motivated by the intuition that as the radii of the individual nodes increase (decrease), roughly speaking, the second largest eigenvalue of $\boldsymbol{P}$ also will decrease (increase). Referring to Hartfiel (1998), we know that the coefficient of ergodicity of a stochastic matrix is an upper bound on the magnitude of the second largest eigenvalue, so $|\lambda| \leq \tau(\boldsymbol{P})$. Recall that for a stochastic matrix $\boldsymbol{A}$,

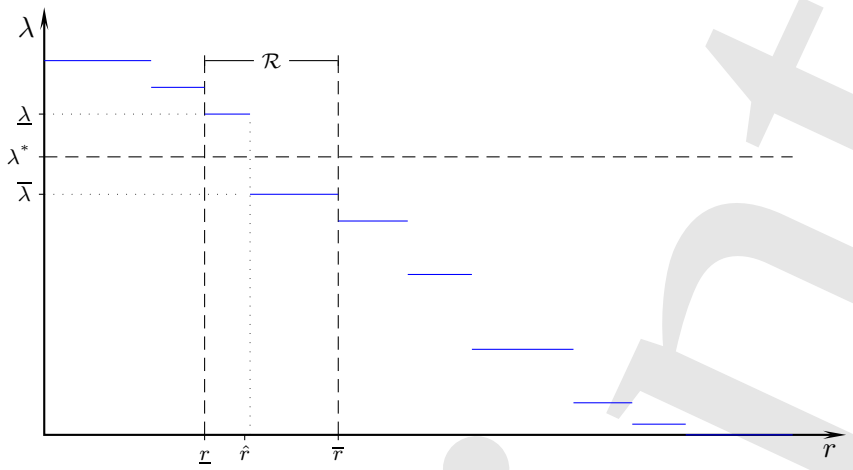

Figure 7. Illustration of a monotonic $\lambda(r)$ curve with some relevant points highlighted relative to λ^* highlighted.

using the L_1 norm, $\tau(\boldsymbol{A})$ is defined as

$$\tau(\boldsymbol{A}) = \frac{1}{2} \max_{i \neq j} \|\boldsymbol{a}_i - \boldsymbol{a}_j\|_1$$

where $\boldsymbol{a}_i$ is the ith row of $\boldsymbol{A}$. Thus, when the rows of $\boldsymbol{P}$ become ever closer to each other as measured by the L_1 norm, $\tau(\boldsymbol{P})$ decreases, and thus the magnitude of the second eigenvalue will also eventually decrease. So even though we are not assured of a locally monotonic relationship, in principle it should still be possible to regulate the magnitude of this second eigenvalue to a neighbourhood around some target value, if we have some knowledge of the approximate manner in which $\lambda(r)$ varies with r.

Before we present our convergence results, some further notation is helpful. Once again, to ease exposition please refer to Figure 7 as we give the following definitions. Let

$$\underline{\lambda} := \inf\left\{\lambda(r) : \lambda(r) \geq \lambda^*\right\} \qquad \text{and} \qquad \overline{\lambda} := \sup\left\{\lambda(r) : \lambda(r) \leq \lambda^*\right\}$$

Then $\overline{\lambda} \leq \lambda^* \leq \underline{\lambda}$. Put simply, for any λ^* there is a feasible λ "just above" and "just below", called $\underline{\lambda}$ and $\overline{\lambda}$ respectively. Now define the following radii

$$\underline{r} := \inf\left\{r : \lambda(r) \leq \underline{\lambda}\right\} \qquad \text{and} \qquad \overline{r} := \sup\left\{r : \lambda(r) \geq \overline{\lambda}\right\}$$

Then $\lambda(r) > \underline{\lambda}$ for $r < \underline{r}$ and $\lambda(r) < \overline{\lambda}$ for $r > \overline{r}$. The radii $\underline{r}$ resp. $\overline{r}$ then are the smallest resp. largest radius so that $\lambda(\underline{r}) \leq \underline{\lambda}$ resp. $\lambda(\overline{r}) \geq \overline{\lambda}$. Finally, we also define the closed interval $\mathcal{R} = [\,\underline{r}\,, \overline{r}\,]$.

With the above definitions, the following two theorems provide simple conditions on the controller gain η so that the system (12) converges to within the interval $\mathcal{R}$ (attractivity), and stays in that interval once it has entered it (invariance). Note that these bounds can be explicitly calculated *a priori* for graphs due to different statistical sensor distributions (or they could be estimated in real time by each node in a decentralised fashion). The important point to note is that the convergence of the controlled system is guaranteed provided that the controller gain is small enough.

Theorem 4.3 contains a condition on η which guarantees that if the system starts in $\mathcal{R}$ it will remain in $\mathcal{R}$. Application of the theorem requires that the graph of λ satisfies the following condition when r is in $\mathcal{R}$: There exists $\kappa_0 > 0$ such that

$$-\kappa_0(r - \underline{r}) \leq \lambda(r) - \lambda^* \leq -\kappa_0(r - \overline{r}) \qquad \text{for} \qquad \underline{r} \leq r \leq \bar{r} \tag{13}$$

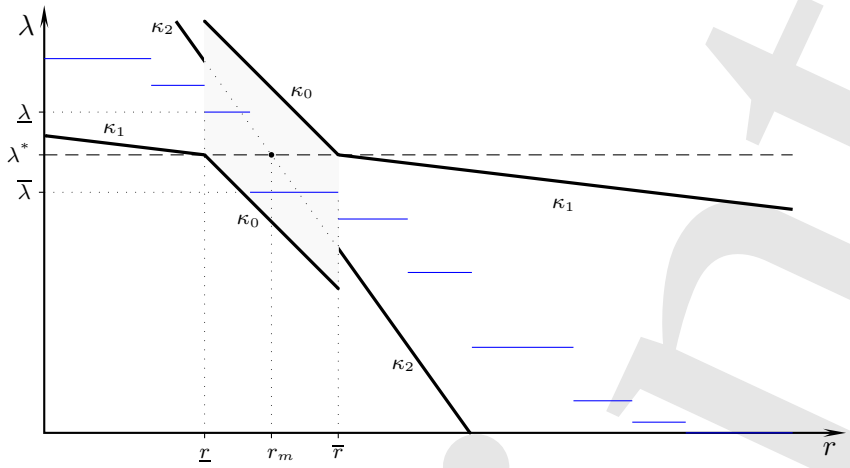

Figure 8. Illustration of the bounds on $\lambda(r)$ as required by Theorems 4.3 and 4.4. See also Figure 11 for a real example of this sketch.

Please refer to Figure 8 for an illustration of these bounds.

Theorem 4.3 Invariance of $\mathcal{R}$**:** *Consider a scalar system described by (12) and let κ_0 and the interval $\mathcal{R}$ be as defined above. Suppose that the control gain $\eta \geq 0$ is chosen such that*

$$\eta\kappa_0 \leq 1 \tag{14}$$

Then whenever $r(0) \in \mathcal{R}$, the resulting sequence $r(k)$ will stay in $\mathcal{R}$ for all $k \geq 0$.

Proof: See Appendix E.

To discuss convergence of the solutions of system (12) to $\mathcal{R}$ we let

$$d(k) := \begin{cases} \underline{r} - r(k) & \text{if} \quad r(k) < \underline{r} \\ 0 & \text{if} \quad \underline{r} \leq r(k) \leq \overline{r} \\ r(k) - \overline{r} & \text{if} \quad r(k) \geq \overline{r} \end{cases} \tag{15}$$

be the distance of $r(k)$ from $\mathcal{R}$. Then we say that $r(k)$ converges to $\mathcal{R}$ if $\lim_{k\to\infty} d(k) = 0$.

The next theorem contains a condition on η which guarantees that all solutions of the system converge to $\mathcal{R}$. Use of this theorem requires that λ satisfy the following sector conditions: There exist constants $\kappa_2 \geq \kappa_1 > 0$ such that

$$\begin{aligned} -\kappa_1(r - \underline{r}) \leq \lambda(r) - \lambda^* \leq -\kappa_2(r - r_m) \quad &\text{for} \quad 0 < r < \underline{r} \\ -\kappa_2(r - r_m) \leq \lambda(r) - \lambda^* \leq -\kappa_1(r - \bar{r}) \quad &\text{for} \quad \bar{r} < r \leq \sqrt{2} \end{aligned}$$

where $r_m := (\underline{r} + \overline{r})/2$. Please refer to Figure 8 for an illustration of those sector bounds.

Theorem 4.4 Attractivity of $\mathcal{R}$**:** *Consider a scalar system described by (12) and let $\kappa_0, \kappa_1, \kappa_2$ and the interval $\mathcal{R}$ be as defined above. Suppose that the control gain $\eta > 0$ is chosen such that $\eta\kappa_0 \leq 1$ and*

$$\eta\kappa_2 < 2 \tag{16}$$

Then every solution of (12) converges to $\mathcal{R}$.

Proof: See Appendix F.

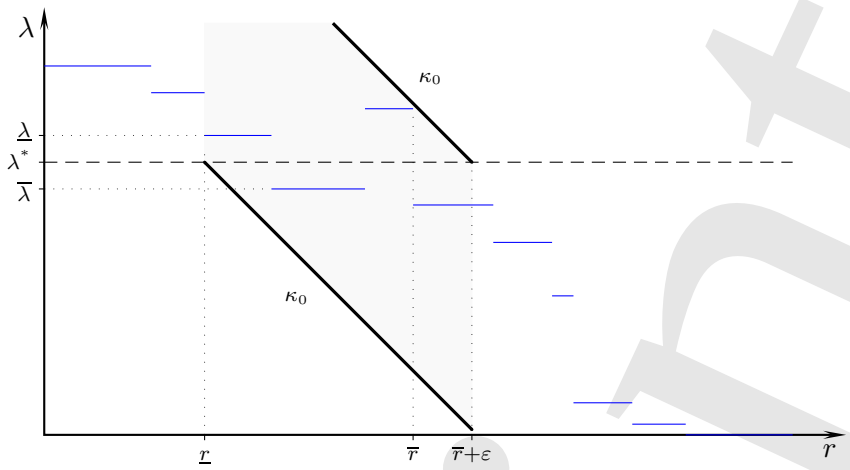

Figure 9. Illustration of $\lambda(r)$ curve that is not monotonic.

Thus, the theorem gives a condition on the control gain so that the closed loop system (12) converges to the interval $\mathcal{R}$.

Comment: If $\lambda(r)$ is not monotonic with r then it is possible that $\lim_{r\to\bar{r}_-}\lambda(r) > \lambda^*$ where the notation means that the limit is taken from the left; see Figure 9. If this occurs, one cannot satisfy (13) with any $\kappa_0 > 0$. In this case (13) can be satisfied by replacing $\bar{r}$ with $\bar{r}_\varepsilon$ where $\bar{r}_\varepsilon = \bar{r} + \varepsilon$ and $\varepsilon > 0$; of course κ_0 will depend on ε; see Figure 9. A similar remark holds if $\lim_{r\to\underline{r}_+}\lambda(r) < \lambda^*$.

Comment: It is possible that with the above control law the network may accidentally become disconnected. The closer λ^* is to one, the more likely this may happen: For instance, assume at time step k the estimated $\lambda(k)$ is smaller than λ^*. In that case, all the nodes will reduce their radius by a certain amount (that is, by $\eta\big[\lambda(k) - \lambda^*\big]$). Now, if updated radii are so small that a particularly "outlying" node becomes "out of reach", the graph will disconnect.

However, in general, the disconnection of the graph can easily prevented by setting a certain minimum radius that the nodes are allowed to use: this would be the smallest common radius (plus, maybe, a safety margin) that would still guarantee connectedness of the network, i.e. it would correspond to the largest internode-distance. This information can either be preprogrammed into the nodes at the time of deployment (if a the corresponding maximum internode-distance can be guaranteed), or after deployment. In any case, this only needs to be done once, as we assume that the nodes do note change their position after deployment.

4.2 *Simulation results*

To conclude this section, we now present some simulation results. Most of the plots shown in this section are based on random disc graphs of 50 nodes, with initial radii uniformly distributed in $[0.1, 0.6]$, and $\lambda^* = 0.8$.

First we show a series of plots to illustrate the pertinent features of our stability proofs, then we show the general performance of our proposed controller, and finally examples of modified control objectives.

4.2.1 Example 1: Controller stability bounds

Figures 10 and 11 show an experimentally obtained $\lambda(r)$ curve, the second figure being a close-up view of the first. Picking $\lambda^* = 0.8$ we indicate the values of $\underline{\lambda}$ and $\bar{\lambda}$, as well as $\underline{r}$, $\overline{r}$ and r_m with dotted lines.

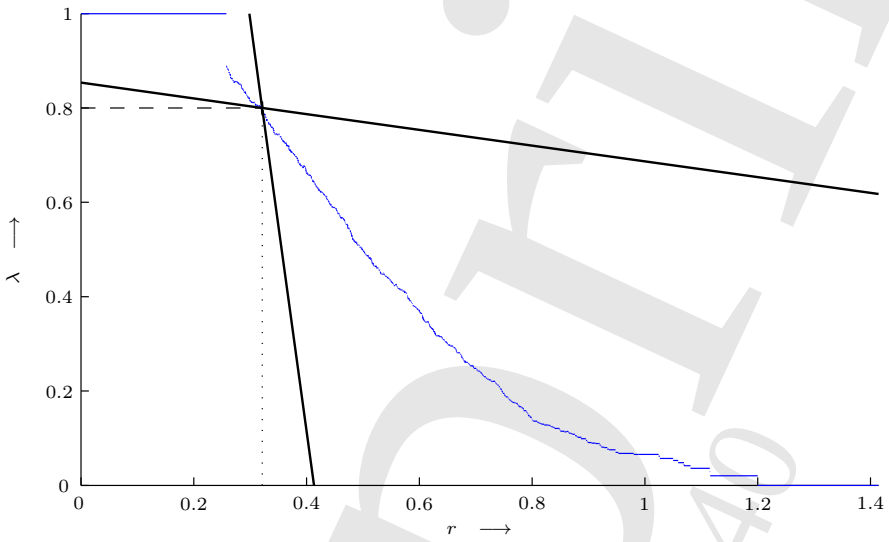

Figure 10. Actual $\lambda(r)$ of a random disc graph on 50 nodes, with an example of the bounds as required for by Theorems 4.3 and 4.4 drawn for $\lambda^* = 0.8$.

We then determined the bounds κ_0, κ_1 and κ_2 on the curve, which are indicated by the thicker lines, similar to Figure 8. The actual values of those bounds are $\kappa_0 \simeq 14.3$, $\kappa_1 \simeq 0.17$ and

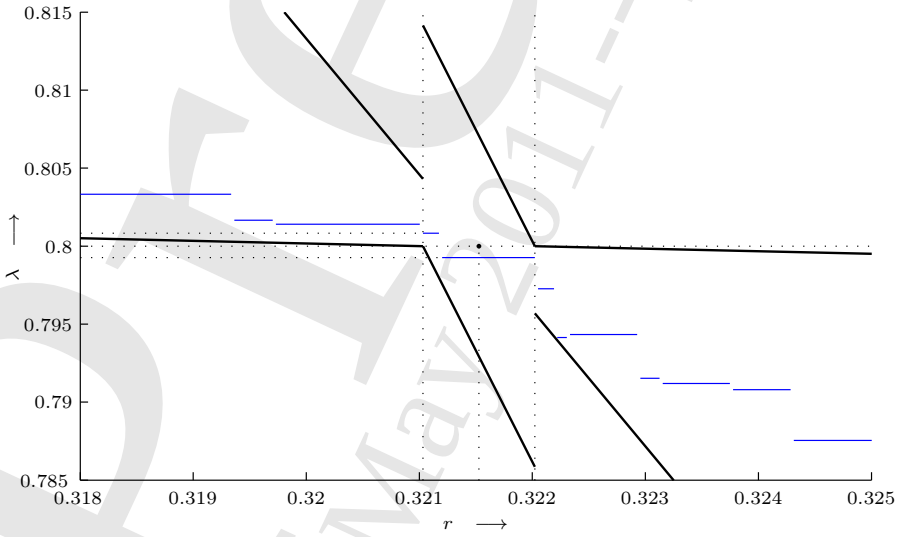

Figure 11. Magnified view of the region around (λ^*, r_m) from the previous plot.

$\kappa_2 \simeq 8.72$.[1] When controlling the nodes' radii with (11), Theorem 4.3 requires that η has to be less than $1/\kappa_0 \simeq 0.067$ to guarantee invariance of the corresponding interval $\mathcal{R} \simeq [0.321\,, 0.322]$. Attractivity of $\mathcal{R}$ according to Theorem 4.4 in turn requires η to be less than $2/\kappa_2 \simeq 0.23$.

Thus setting $\eta = 0.05$, we re-initialised the network with randomly distributed radii in the $[0.1\,, 0.6]$ and ran the controller on the network. As we can see in Figure 12 the convergence of both the radii and λ is smooth and fast.

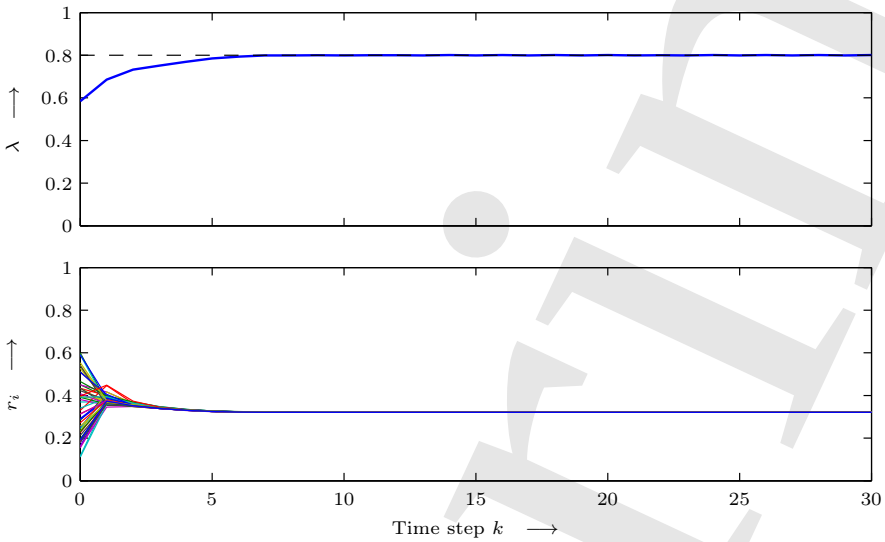

Figure 12. Evolution of $\lambda(k)$ and the individual nodes' radii $r_i(k)$ in the 50 node network analysed in Figure 10, for $\lambda^* = 0.8$, with $\eta = 0.05$.

[1]Note that tighter bounds can be found.

4.2.2 Example 2: Combining Control and Estimation

In the previous example we displayed the converged values of the estimation scheme. To show in more detail how estimation and control scheme work together, we present Figure 13. Plotted is again the evolution of the nodes' radii under control action (11) as well as the estimates of λ, shown in the upper subplot. These estimates where calculated as described in Section 3. We allowed 100 timesteps for the estimation scheme to converge, before taking a control action based on the new estimates.

It can be seen that after every topology change all nodes' estimates converge to a common value and that the control scheme successfully regulates the second largest eigenvalue of the network to $\lambda^* = 0.8$.

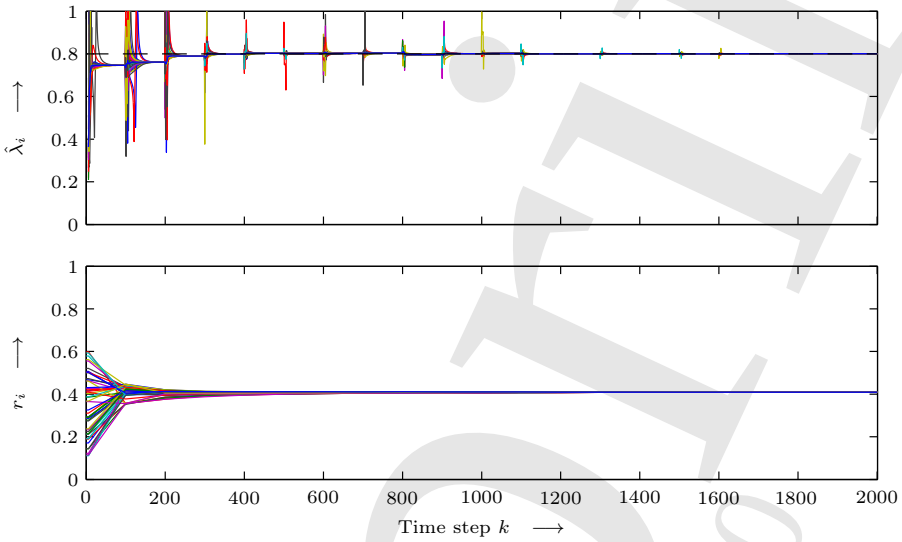

Figure 13. Evolution of the estimates of $\lambda(k)$ and the individual nodes' radii $r_i(k)$, as the controller updates the radii every 100 iterations of the estimation scheme, for $\lambda^* = 0.8$, with $\eta = 0.05$

4.2.3 Further Examples of control

Next, we present another example that depicts how the second largest eigenvalue in magnitude and the nodes' radii change over time, as the nodes control their radii using (11).

Figure 14 shows a situation where $\lambda^* = 0.5$ was required. As this represents a very densly connected network, all nodes had to increase their radius. In turn, in Figure 15 we start off with an extremely highly connected network (it was almost fully connected), and all nodes have to siginificantly decrease their radii to achieve the desire $\lambda^* = 0.8$.

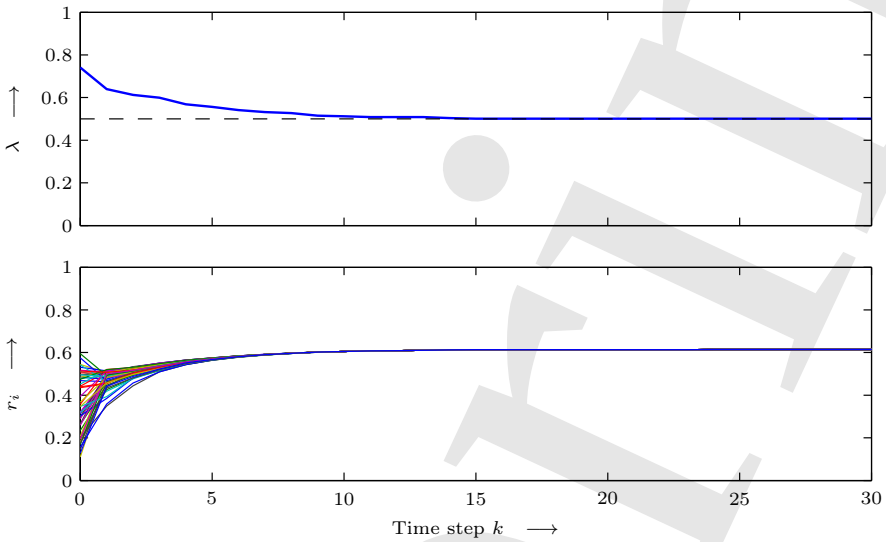

Figure 14. Evolution of $\lambda(k)$ and the individual nodes' radii $r_i(k)$ in a network of 50 nodes for $\lambda^* = 0.5$, with $\eta = 0.05$.

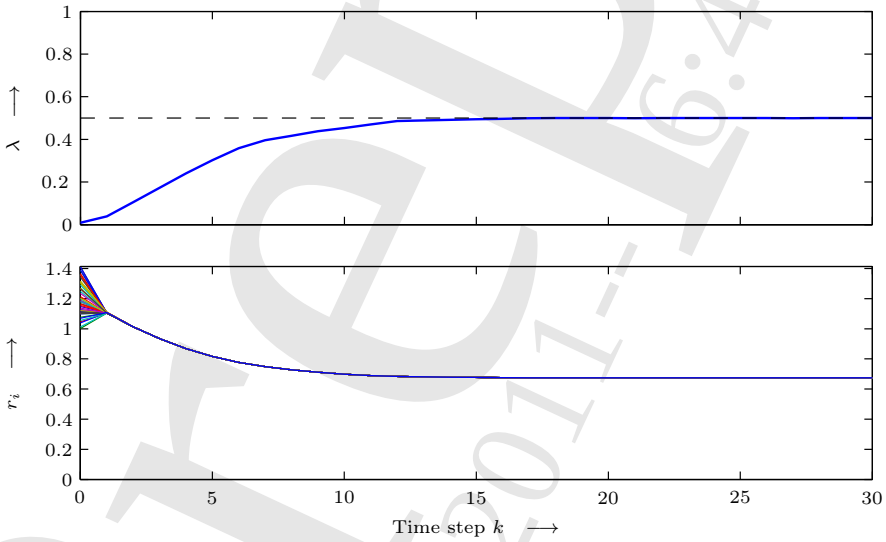

Figure 15. Evolution of $\lambda(k)$ and the individual nodes' radii $r_i(k)$ in a network of 50 nodes with very large initial radii, for $\lambda^* = 0.8$, with $\eta = 0.05$.

The plots in Figure 16 show a sceanario where the network had to react to a change in topology: at $k = 30$ we randomly "killed" half of the nodes, thus reducing the network to 25 nodes. The resulting network's second largest eigenvalue in magnitude is larger than desired (i. e. it is less connected), and thus the controller compensates this by increasing the remaining nodes' radii until $\lambda^* = 0.8$ is achieved again.

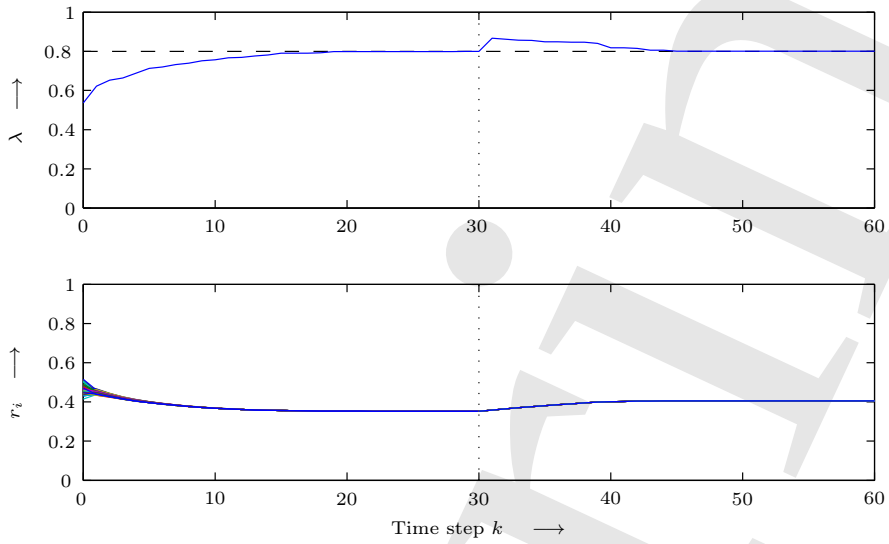

Figure 16. Evolution of $\lambda(k)$ and the individual nodes' radii $r_i(k)$ in a network of 50 nodes, where 25 nodes are removed at $k = 30$ (for $\lambda^* = 0.8$, with $\eta = 0.05$).

4.2.4 Validation of control results

In Figure 17 we compare the converged radii of our controller for several different λ^* (circles) with the second largest eigenvalue in magnitude of the averaging matrix of random disc graphs created with different initial radii (crosses). Until now we have only shown individual results from single instances of graphs. This plot is to demonstrate that our estimation and control scheme works over a whole range of set points, for any number of trials.

The data points marked by crosses were obtained as follows. Picking 17 different values of r we generated 1000 random geometric graphs (on 50 nodes) for each radius. Next, we calculated the second largest eigenvalue of the resulting averaging matrix of each graph $\lambda(r)$, and finally plotted the average value against the initial r value used. In turn, the data points marked by circles were generated by choosing 14 different values for λ^*, generating 1000 graphs and running the control algorithm on the network. The resulting converged (common) radii $r_{\text{conv}}(\lambda^*)$ were then averaged and the value plotted against the particular λ^* chosen.

As all points appear to lay on the same curve, the plot indicates that nodes radii set by the controller indeed converge to the corrected value over the entire range of sensible λ^* values.

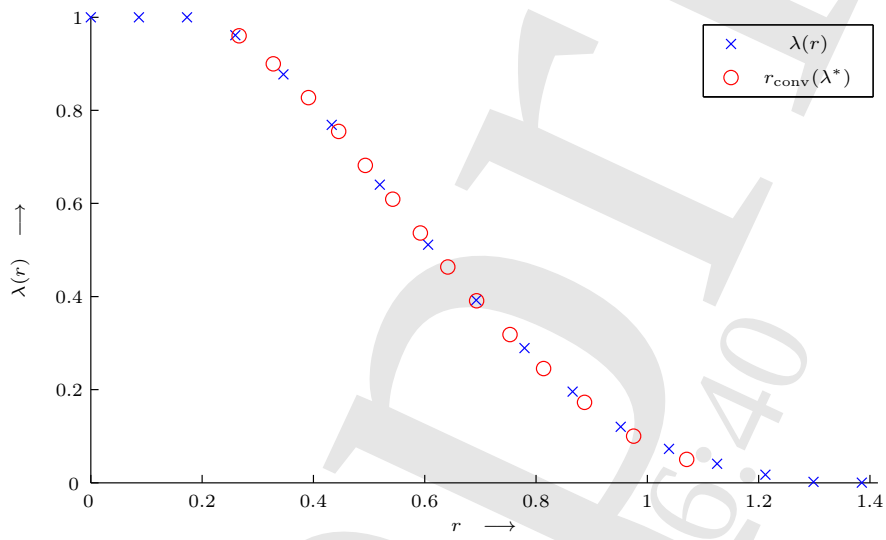

Figure 17. Crosses: average $\lambda(r)$ of 1000 geometric graphs on 50 nodes created with common radius r. Circles: Average converged radii r_{conv} after control targeted at different λ^* values, for 1000 trials each (where the initial radii where randomly distributed).

4.2.5 Examples of other control objectives

As we mentioned in Section 2, our control scheme is general enough to allow objectives other than a common radius while achieving a desired λ^*.

Imagine a situation in which some nodes are equipped with a longer-lasting power supply and we can allow those nodes to have a larger radius than most of the other nodes in the network. This would correspond to weighting the nodes' radii in the averaging scheme. It is possible to include such weighting in our framework, and all the proofs directly hold with but a small modification, Knorn et al. (2009). An example of this is given in Figure 18, where by design we wish one node to have twice the radius as the others, and one node half the radius. As can be seen, the eigenvalue of the network converges quickly to its desired value of $\lambda^* = 0.8$, and the nodes radii converge to a common value with the exception of the two nodes of different weighting.

Comment: Note that such a weighting results, contrary to the other cases, in a *directed* network (that is, a non-symmetric averaging matrix), even in steady state. As we mentioned earlier it is an important feature of our algorithms that they work in both undirected and directed networks.

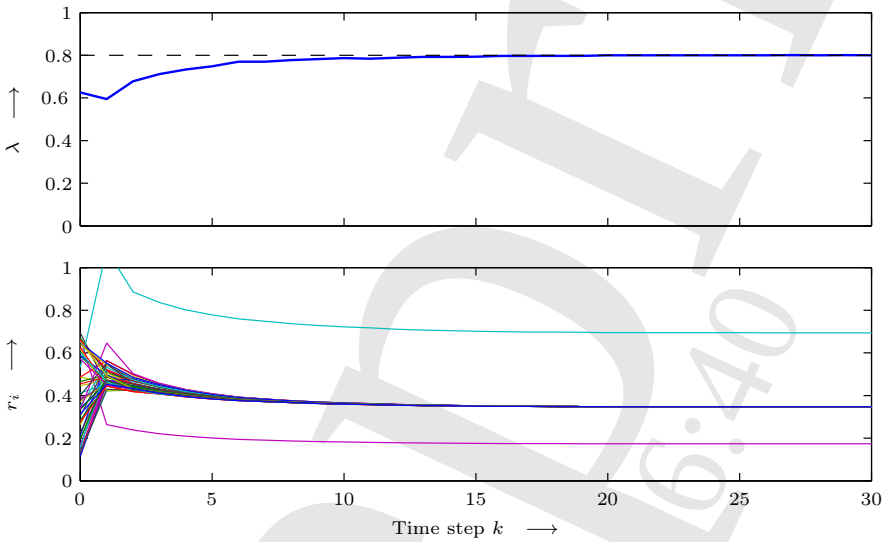

Figure 18. Evolution of $\lambda(k)$ and the individual nodes' radii $r_i(k)$ in a network of 50 where two nodes where to have twice resp. half the radius as their peers. Again, $\lambda^* = 0.8$ and $\eta = 0.05$.

Finally we now present an example where a completely different control objective is desired. Regulating the second largest eigenvalue in magnitude, here we do not care about the radii but rather about the number of neighbours of each node. In Figure 19 we required the nodes to achieve consensus on the number of neighbours, rather than the radii. Although one needs to redo the proof of stability, we can see that the network converges to a stable solution.

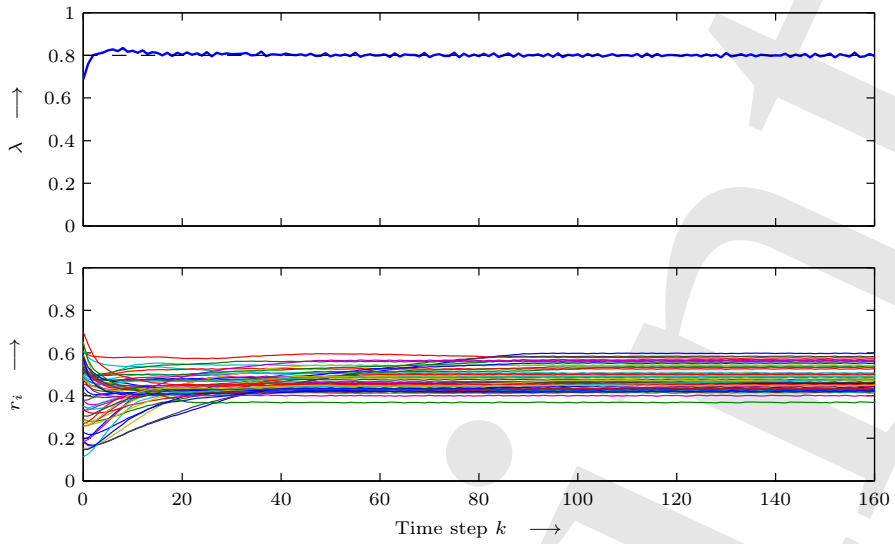

Figure 19. Evolution of $\lambda(k)$ and the individual nodes' radii $r_i(k)$, 50 nodes, consensus on number of neighbours.

5 Conclusion

In this paper we have presented a general framework for controlling the topological properties of a network of distributed sensors. Our framework breaks free of many of the assumptions of previous work such as graph symmetry, and utilises only simple ideas from control and estimation to regulate important graph properties. Conditions for the stability of our algorithms are presented. Roughly speaking, these results state that if the nodes are not too aggressive in the manner in which they expand or contract their neighbourhood set, stability is assured.

While examples are presented to illustrate the efficacy, and promise, of the approach, many open questions remain to be resolved. The most important of these concerns the fact that the relationship between the network states and the eigenvalue locus is not known exactly *a priori*. This however may not be an important issue as it can certainly be estimated off-line for typical graphs, and can be estimated by each of the sensors in an adaptive fashion as the consensus algorithm evolves. This open question will be the subject of our further research, as well as applying the technique in real world sensor networks. Further work will also include extending the framework to include power and interference issues, as well as developing stochastic versions of the algorithm.

Acknowledgements

This work was supported by *Science Foundation Ireland* PI Award 07/IN.1/1901.

References

Blondel, V.D., Hendrickx, J.M., Olshevsky, A., and Tsitsiklis, J.N. (2005), "Convergence in Multiagent Coordination, Consensus, and Flocking," in *CDC-ECC '05: Proceedings of the Joint 44th IEEE Conference on Decision and Control, and the European Control Conference*, Seville, Spain, December, pp. 2996–3000.

Boyd, S., Diaconis, P., and Xiao, L. (2004), "Fastest Mixing Markov Chain on a Graph," *SIAM Review*, 46, 667–689.

Cabrera, J.B.D., Ramanatha, R., Gutiérrez, C., and Mehra, R.K. (2007), "Stable Topology for Mobile Ad-Hoc Networks," *IEEE Communications Letters*, 11, 574–576.

Chung, F.R.K., *Spectral Graph Theory*, no. 92 in: CBMS Regional Conference Series in Mathematics, Province, RI, USA: American Mathematical Society (1997).

Dimarogonas, D.V., and Kyriakopoulos, K.J. (2008), "Connectedness Preserving Distributed Swarm Aggregation for Multiple Kinematic Robots," *IEEE Transactions on Robotics*, 24, 1213–1223.

Estrin, D., Girod, L.D., Pottie, G.J., and Srivastava, M. (2001), "Instrumenting the world with wireless sensor networks," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, Salt Lake City, UT, USA, May, Vol. 4, pp. 2033–2036.

Fallat, S.M., and Kirkland, S. (1998), "Extremizing Algebraic Connectivity Subject to Graph Theoretic Constraints," *Electronic Journal of Linear Algebra*, 3, 48–74.

Fiedler, M. (1973), "Algebraic Connectivity of Graphs," *Czechoslovak Mathematical Journal*, 23, 298–305.

Gennaro, M.C.D., and Jadbabaie, A. (2006), "Decentralized Control of Connectivity for Multi–Agent Systems," in *CDC'06: Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, USA, December, pp. 3628–3633.

Ghosh, A., and Boyd, S.P. (2006), "Growing Well-connected Graphs," in *Proceedings 45th IEEE Conference on Decision and Control*, San Diego, CA, USA, December, San Diego, CA, USA, pp. 6605–6611.

Hartfiel, D.J., *Markov Set-Chains*, Vol. 1695 of *Lecture Notes in Mathematics*, Springer Berlin / Heidelberg, Germany (1998).

Horn, R.A., and Johnson, C.R., *Matrix Analysis*, Cambridge, UK: Cambridge University Press (1985).

Jadbabaie, A., Lin, J., and Morse, A.S. (2003), "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, 48, 988–1001.

Ji, M., and Egerstedt, M. (2005), "Connectedness preserving distributed coordination control over dynamic graphs," in *ACC'05: Proceedings of the 2005 American Control Conference*, Portland, OR, USA, June, pp. 93–98.

Knorn, F., Stanojevic, R., Corless, M., and Shorten, R., "A Problem in Positive Systems Stability Arising in Topology Control." To appear in: *POSTA '09: Proceedings of the Third Multidisciplinary International Symposium on Positive Systems*, Valencia, Spain (2009).

Moreau, L. (2005), "Stability of Multiagent Systems with Time-Dependent Communication Links," *IEEE Transactions on Automatic Control*, 50, 169–182.

Olfati-Saber, R., and Murray, R.M. (2004), "Consensus Problems in Networks of Agents with Switching Topology and Time-Delays," *IEEE Transactions on Automatic Control*, 49, 1520–1533.

Penrose, M., *Random Geometric Graphs*, Vol. 5 of *Oxford Studies in Probability*, New York, NY, USA: Oxford University Press (2003).

Ramanathan, R., and Rosales-Hain, R. (2000), "Topology Control of Multihop Wireless Networks using Transmit Power Adjustment," in *INFOCOM'00: Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, Tel Aviv, Israel, March,

Vol. 2, pp. 404–413.

Reynolds, C.W. (1987), "Flocks, Herds, and Schools: A Distributed Behavioral Model," in *SIGGRAPH'87: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, Anaheim, CA, USA, July, pp. 25–34.

Santi, P., *Topology Control in Wireless Ad Hoc and Sensor Networks*, Chichester, UK: John Wiley & Sons, Inc. (2005).

Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., and Shochet, O. (1995), "Novel Type of Phase Transition in a System of Self-Driven Particles," *Physical Review Letters*, 75, 1226–1229.

Wattenhofer, R., Li, L., Bahl, P., and Wang, Y.M. (2001), "Distributed Topology Control for Wireless Multihop Ad-hoc Networks," in *INFOCOM'01: Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, Anchorage, AK, USA, April, Vol. 3, pp. 1388–1397.

Zavlanos, M.M., and Pappas, G.J. (2005), "Controlling Connectivity of Dynamic Graphs," in *CDC-ECC '05: Proceedings of the Joint 44th IEEE Conference on Decision and Control, and the European Control Conference*, Seville, Spain, December, pp. 6388–6393.

Åström, K.J., and Wittenmark, B., *Computer-controlled systems: Theory and Design*, 3rd ed., Upper Saddle River, NJ, USA: Prentice-Hall, Inc. (1997).

Appendix A: Proof of Proposition 3.1

Recall from (3) that for node i:

$$x_i(k) = c_1 + \lambda_2^k \underbrace{\left[\sum_{j=2}^{n} c_j \left(\frac{\lambda_j}{\lambda_2}\right)^k \nu_{ji}\right]}_{=:\psi_i(k)}$$

where ν_{ji} denotes the ith element of the jth eigenvector of $\boldsymbol{P}$. We then have for $k > m+1$

$$\begin{aligned}\frac{z_i(k)}{z_i(k-m)} &= \frac{x_i(k) - x_i(k-1)}{x_i(k-m) - x_i(k-m-1)} = \frac{\lambda_2^k\,\psi_i(k) \;-\; \lambda_2^{k-1}\,\psi_i(k-1)}{\lambda_2^{k-m}\,\psi_i(k-m) \;-\; \lambda_2^{k-m-1}\,\psi_i(k-m-1)} \\ &= \lambda_2^m \underbrace{\frac{\psi_i(k) \;-\; \lambda_2^{-1}\,\psi_i(k-1)}{\psi_i(k-m) \;-\; \lambda_2^{-1}\,\psi_i(k-m-1)}}_{=:w_i(k,m)} \end{aligned} \tag{A1}$$

and taking the mth root of the absolute values of both sides

$$\underbrace{\left|\frac{z_i(k)}{z_i(k-m)}\right|^{1/m}}_{\tilde{\lambda}_{2i}(k)} = |\lambda_2| \cdot \left|w_i(k,m)\right|^{1/m} \tag{A2}$$

From the last equation we can see that the estimate $\tilde{\lambda}_{2i}(k)$ approaches the true absolute value of the second largest eigenvalue if and only if the $|w_i(k,m)| \to 1$, as k grows. Since

$$\psi_i(k) = c_2\nu_{2i} + \sum_{j=3}^{n} c_j \left(\frac{\lambda_j}{\lambda_2}\right)^k \nu_{ji}$$

it will converge to $c_2\nu_{2i}$ as k grows, as by assumption $\left|\frac{\lambda_j}{\lambda_2}\right|^k < 1$ for $j = 3, \ldots, n$. For a general initial condition $c_2\nu_{2i}$ is non-zero and, using (A1), we now have that $|w_i(k,m)| \to 1$ and thus $\tilde{\lambda}_{2i}(k) \to |\lambda_2|$ as $k \to \infty$. □

Appendix B: Proof of Proposition 3.2

For any node i, substituting expression (6) into (7), and dropping the subscripts "$_i$" and "$_2$" to increase legibility, yields

$$\begin{aligned}\zeta(k) &= \Big[c\lambda^k + \bar{c}\bar{\lambda}^k + |\lambda|^k O(k)\Big]\Big[c\lambda^{(k-2)} + \bar{c}\bar{\lambda}^{(k-2)} + |\lambda|^{(k-2)} O(k-2)\Big] \\ &\qquad - \Big[c\lambda^{(k-1)} + \bar{c}\bar{\lambda}^{(k-1)} + |\lambda|^{(k-1)} O(k-1)\Big]^2 \\ &= c\bar{c}\Big[\lambda^k\bar{\lambda}^{(k-2)} + \bar{\lambda}^k\lambda^{(k-2)} - 2\lambda^{(k-1)}\bar{\lambda}^{(k-1)}\Big] + |\lambda|^{(2k-4)}\tilde{O}(k) \\ &= |c|^2|\lambda|^{(2k-4)}\Big[\bar{\lambda}^2 + \lambda^2 - 2\lambda\bar{\lambda}\Big] + |\lambda|^{(2k-4)}\tilde{O}(k) \\ &= |\lambda|^{(2k-4)}\Big[|c|^2(\lambda - \bar{\lambda})^2 + \tilde{O}(k)\Big] \end{aligned} \tag{B1}$$

where

$$\tilde{O}(k) = |\lambda|^2 \Big\{ O(k-2) \left[c \left(\frac{\lambda}{|\lambda|} \right)^k + \bar{c} \left(\frac{\bar{\lambda}}{|\lambda|} \right)^k \right] + O(k) \left[c \left(\frac{\lambda}{|\lambda|} \right)^{(k-2)} + \bar{c} \left(\frac{\bar{\lambda}}{|\lambda|} \right)^{(k-2)} \right] + O(k)O(k-2) - 2\,O(k-1) \left[c \left(\frac{\lambda}{|\lambda|} \right)^{(k-1)} + \bar{c} \left(\frac{\bar{\lambda}}{|\lambda|} \right)^{(k-1)} \right] - O(k-1)^2 \Big\}$$

We note that since $O(k) \to 0$ as $k \to 0$, we also have

$$\lim_{k \to \infty} \tilde{O}(k) = 0 \tag{B2}$$

Furthermore, since $c, \bar{c} \neq 0$ and λ has nonzero imaginary part, $|c|^2(\lambda - \bar{\lambda})^2$ is nonzero, and thus $\zeta(k)$ in (B1) is also non-zero for k sufficiently large. Finally,

$$\left| \frac{\zeta(k)}{\zeta(k-m)} \right| = |\lambda|^{2m} \left[\frac{|c|^2(\lambda - \bar{\lambda})^2 + \tilde{O}(k)}{|c|^2(\lambda - \bar{\lambda})^2 + \tilde{O}(k-m)} \right] \tag{B3}$$

From this last expression and (B2) we obtain that

$$\lim_{k \to \infty} \left| \frac{\zeta(k)}{\zeta(k-m)} \right|^{\frac{1}{2m}} = |\lambda| \tag{B4}$$

which completes the proof. □

Appendix C: Proof of Theorem 4.1

For $k \geq 1$, define

$$\tilde{\boldsymbol{x}}(k) := \sigma(k)\mathbf{1} \qquad \text{where} \qquad \sigma(k) := \sum_{i=0}^{k-1} \theta\big(x(i), i\big) \tag{C1}$$

Since $\boldsymbol{P}(k)$ is row-stochastic,

$$\boldsymbol{P}(k)\tilde{\boldsymbol{x}}(k) = \boldsymbol{P}(k)\big[\sigma(k)\mathbf{1}\big] = \sigma(k)\boldsymbol{P}(k)\mathbf{1} = \sigma(k)\mathbf{1} = \tilde{\boldsymbol{x}}(k)$$

Hence

$$\begin{aligned} \tilde{\boldsymbol{x}}(k+1) &= \tilde{\boldsymbol{x}}(k) + \theta\big(x(k), k\big)\mathbf{1} \\ &= \boldsymbol{P}(k)\tilde{\boldsymbol{x}}(k) + \theta\big(x(k), k\big)\mathbf{1} \end{aligned} \tag{C2}$$

Letting $\boldsymbol{y}(k) = \boldsymbol{x}(k) - \tilde{\boldsymbol{x}}(k)$, it follows from (9) and (C2) that $\boldsymbol{y}(k+1) = \boldsymbol{P}(k)\boldsymbol{y}(k)$. Since all the $\boldsymbol{P}(k)$ are taken from a finite set of primitive and row-stochastic matrices, there exists a constant scalar $\bar{\theta}$ such that

$$\lim_{k \to \infty} \boldsymbol{y}(k) = \bar{\theta}\mathbf{1} \tag{C3}$$

see for instance Hartfiel (1998). This means that as $k \to \infty$, the elements in $\boldsymbol{y}(k)$ approach a common value, $\bar{\theta}$. Since $\boldsymbol{x}(k) = \boldsymbol{y}(k) + \sigma(k)\mathbf{1}$ the desired result follows. □

Appendix D: Proof of Theorem 4.2

Start by defining

$$\boldsymbol{y}(k) := \boldsymbol{x}(k) - \sigma(k)\mathbf{1} \qquad \text{where} \qquad \sigma(k) := \sum_{i=0}^{k-1} \theta\big(\boldsymbol{x}(i), i\big) \tag{D1}$$

Then $\sigma(k+1) = \sigma(k) + \theta\big(\boldsymbol{x}(k), k\big)$ and

$$\begin{aligned}
\boldsymbol{y}(k+1) &= \boldsymbol{x}(k+1) - \sigma(k+1)\mathbf{1} \\
&\overset{(10)}{=} f\big(\boldsymbol{x}(k), k\big) + \theta\big(\boldsymbol{x}(k), k\big)\mathbf{1} - \Big[\sigma(k) + \theta\big(\boldsymbol{x}(k), k\big)\Big]\mathbf{1} \\
&\overset{(D1)}{=} \underbrace{f\big(\boldsymbol{y}(k) + \sigma(k)\mathbf{1}\ ,\ k\big) - \sigma(k)\mathbf{1}}_{:= g(\boldsymbol{y}(k), k)}
\end{aligned}$$

Now, if g satisfies all of the assumptions (1)–(4) of the theorem, the results from Moreau (2005) guarantee that all entries in $\boldsymbol{y}(k)$ will converge to a common value, and hence, through (D1), the values in $\boldsymbol{x}(k)$ have to approach each other. So let us test g for each of the four assumptions.

(1) For all nodes $i \in \mathcal{V}$,

$$g_i\big(\boldsymbol{y}(k), k\big) = f\big(\boldsymbol{y}(k) + \sigma(k)\mathbf{1}\ ,\ k\big) - \sigma(k) \in \underbrace{\mathcal{E}_i\big(\mathcal{A}(k)\big)\big(\boldsymbol{y}(k) + \sigma(k)\mathbf{1}\big) - \sigma(k)}_{=: \hat{\mathcal{E}}_i(\mathcal{A}(k))(\boldsymbol{y}(k))}$$

Clearly, if $f\big(\boldsymbol{x}(k), k\big) \in \mathcal{E}_i\big(\mathcal{A}(k)\big)\big(\boldsymbol{x}(k)\big)$ for all $i \in \mathcal{V}$, $k \in \mathbb{N}$ and $\boldsymbol{x} \in \mathcal{X}^n$, and if $\mathcal{E}_i\big(\mathcal{A}(k)\big)\big(\boldsymbol{x}(k)\big)$ is compact, then $\hat{\mathcal{E}}_i\big(\mathcal{A}(k)\big)\big(\boldsymbol{y}(k)\big)$ is also compact given σ is bounded.

(2) Whenever the states of node i and its neighbours are all equal, that is $y_i(k) = y_j(k)$ for all $j \in \mathcal{N}_i$,

$$\hat{\mathcal{E}}_i\big(\mathcal{A}(k)\big)\big(\boldsymbol{y}(k)\big) = \mathcal{E}_i\big(\mathcal{A}(k)\big)\big(\boldsymbol{y}(k) + \sigma(k)\mathbf{1}\big) - \sigma(k) = \big\{y_i(k) + \sigma(k)\big\} - \sigma(k) = \{y_i(k)\}$$

(3) Assume the states of node i and its neighbours $j \in \mathcal{N}_i$ are not all equal. If $\mathcal{E}_i\big(\mathcal{A}(k)\big)\big(\boldsymbol{x}(k)\big)$ is contained in the relative interior of the convex hull ($\operatorname{conv}\{\cdot\}$) of the states of node i and its neighbours, we have

$$\begin{aligned}
\mathcal{E}_i\big(\mathcal{A}(k)\big)\big(\boldsymbol{x}(k)\big) &\subset \operatorname*{convh}_{j \in \mathcal{N}_i} \big\{x_j(k)\big\} \\
\mathcal{E}_i\big(\mathcal{A}(k)\big)\big(\boldsymbol{x}(k) - \sigma(k)\mathbf{1} + \sigma(k)\mathbf{1}\big) &\subset \operatorname*{convh}_{j \in \mathcal{N}_i} \big\{x_j(k) + \sigma(k) - \sigma(k)\big\} \\
\mathcal{E}_i\big(\mathcal{A}(k)\big)\big(\boldsymbol{y}(k) + \sigma(k)\mathbf{1}\big) - \sigma(k) &\subset \operatorname*{convh}_{j \in \mathcal{N}_i} \big\{y_j(k) + \sigma(k)\big\} - \sigma(k)
\end{aligned}$$

and with convh$\{\cdot\}$ being a linear operator

$$\mathcal{E}_i\big(\mathcal{A}(k)\big)\big(\boldsymbol{y}(k)+\sigma(k)\mathbf{1}\big)-\sigma(k)\subset\operatorname*{convh}_{j\in\mathcal{N}_i}\big\{y_j(k)+\sigma(k)-\sigma(k)\big\}$$
$$\hat{\mathcal{E}}_i\big(\mathcal{A}(k)\big)\big(\boldsymbol{y}(k)\big)\subset\operatorname*{convh}_{j\in\mathcal{N}_i}\big\{y_j(k)\big\}$$

(4) If $\mathcal{E}_i\big(\mathcal{A}(k)\big)\big(\boldsymbol{x}(k)\big)$ depends continuously on $\boldsymbol{x}(k)$, so will $\mathcal{E}_i\big(\mathcal{A}(k)\big)\big(\boldsymbol{x}(k)+\sigma(k)\mathbf{1}\big)-\sigma(k)\mathbf{1}=\hat{\mathcal{E}}_i\big(\mathcal{A}(k)\big)\big(\boldsymbol{y}(k)\big)$.

We have thus established, that the update map g satisfies Assumption 1. Assuming that the graphs never disconnect, we can now apply Theorem 1 from Moreau (2005). It guarantees that the entries in $\boldsymbol{y}(k)$ will converge to a common value, and thus, through (D1) the states $\boldsymbol{x}(k)$ have to approach each other so that $x_i(k)-x_j(k)\to 0$ as $k\to\infty$. □

Appendix E: Proof of Theorem 4.3

Suppose that $r(k)\in\mathcal{R}$. We need to show that $r(k+1)\in\mathcal{R}$. Then we will have demonstrated invariance of $\mathcal{R}$. We first show that $r(k+1)\le\overline{r}$. Since $\eta\ge 0$ and $\eta\kappa_0\le 1$, it follows from condition (13) and $r(k)\in\mathcal{R}$ that

$$\begin{aligned}\eta\big[\lambda\big(r(k)\big)-\lambda^*\big]&\le-\eta\kappa_0\big[r(k)-\overline{r}\big]\\&\le\eta\kappa_0\big[\overline{r}-r(k)\big]\\&\le\bar{r}-r(k)\,;\end{aligned}$$

hence

$$\begin{aligned}r(k+1)&=r(k)+\eta\big[\lambda\big(r(k)\big)-\lambda^*\big]\\&\le r(k)+\overline{r}-r(k)\\&\le\overline{r}\end{aligned}$$

Next, we show that $r(k+1)\ge\underline{r}$. Since $\eta\ge 0$ and $\eta\kappa_0\le 1$, it follows from condition (13) and $r(k)\in\mathcal{R}$ that

$$\begin{aligned}\eta\big[\lambda\big(r(k)\big)-\lambda^*\big]&\ge-\eta\kappa_0\big[r(k)-\underline{r}\big]\\&\ge-\big[r(k)-\underline{r}\big]\\&\ge\underline{r}-r(k)\,;\end{aligned}$$

hence

$$\begin{aligned}r(k+1)&=r(k)+\eta\big[\lambda\big(r(k)\big)-\lambda^*\big]\\&\ge r(k)+\underline{r}-r(k)\\&\ge\underline{r}\end{aligned}$$

□

Appendix F: Proof of Theorem 4.4

Letting $\alpha := \max\{\, 1-\eta\kappa_1 \,, \eta\kappa_2 - 1 \,\}$ we will show that

$$d(k+1) \leq \alpha d(k) \tag{F1}$$

and hence $d(k) \leq \alpha^k d(0)$. Since by assumption $|\alpha| < 1$, we then obtain that $\lim_{k\to\infty} d(k) = 0$. Since $\mathcal{R}$ is invariant, we need only discuss the situations for which $r(k) \notin \mathcal{R}$ as well as $r(k+1) \notin \mathcal{R}$. There are four cases to consider.

(1) $r(k) < \underline{r}$ and $r(k+1) < \overline{r}$. In this case $d(k) = \underline{r} - r(k)$ and

$$\begin{aligned} d(k+1) &= \underline{r} - r(k+1) \\ &= \underline{r} - r(k) - \eta\Big[\lambda\big(r(k)\big) - \lambda^*\Big] \\ &\leq \underline{r} - r(k) - \eta\kappa_1\big[\underline{r} - r(k)\big] \\ &\leq (1-\eta\kappa_1)\big[\underline{r} - r(k)\big] \\ &\leq (1-\eta\kappa_1) d(k) \end{aligned}$$

that is $d(k+1) \leq (1-\eta\kappa_1)d(k)$, and thus (F1) holds.

(2) $r(k) < \underline{r}$ and $r(k+1) > \overline{r}$. In this case $d(k) = \underline{r} - r(k)$ and

$$\begin{aligned} d(k+1) &= r(k+1) - \bar{r} \\ &= r(k) + \eta\Big[\lambda\big(r(k)\big) - \lambda^*\Big] - \bar{r} \\ &\leq r(k) + \eta\kappa_2\big[r_m - r(k)\big] - \bar{r} \\ &\leq (1-\eta\kappa_2) r(k) + \eta\kappa_2 r_m - \bar{r} \end{aligned}$$

Recalling that $\eta\kappa_2 < 2$ and $r_m = (\underline{r} + \bar{r})/2$, we can see that

$$\begin{aligned} \eta\kappa_2 r_m - \bar{r} &= -\left(1 - \frac{\eta\kappa_2}{2}\right)\overline{r} + \frac{\eta\kappa_2}{2}\underline{r} \\ &\leq -\left(1 - \frac{\eta\kappa_2}{2}\right)\underline{r} + \frac{\eta\kappa_2}{2}\underline{r} \\ &\leq -(1-\eta\kappa_2)\underline{r} \end{aligned}$$

Hence

$$\begin{aligned} d(k+1) &\leq (1-\eta\kappa_2) r(k) - (1-\eta\kappa_2)\underline{r} \\ &\leq (\eta\kappa_2 - 1) d(k) \end{aligned}$$

and thus (F1) holds.

(3) $r(k) > \overline{r}$ and $r(k+1) < \overline{r}$. In this case $d(k) = r(k) - \bar{r}$ and

$$\begin{aligned} d(k+1) &= \underline{r} - r(k+1) \\ &= \underline{r} - r(k) - \eta\Big[\lambda\big(r(k)\big) - \lambda^*\Big] \\ &\leq \underline{r} - r(k) - \eta\kappa_2\big[r_m - r(k)\big] \\ &\leq -(1-\eta\kappa_2)r(k) + \underline{r} - \eta\kappa_2 r_m \end{aligned}$$

Again, we can see that since $\eta\kappa_2 < 2$

$$\begin{aligned} \underline{r} - \eta\kappa_2 r_m &= \Big(1 - \frac{\eta\kappa_2}{2}\Big)\underline{r} - \frac{\eta\kappa_2}{2}\bar{r} \\ &\leq \Big(1 - \frac{\eta\kappa_2}{2}\Big)\bar{r} - \frac{\eta\kappa_2}{2}\bar{r} \\ &\leq (1-\eta\kappa_2)\bar{r} \end{aligned}$$

Hence

$$\begin{aligned} d(k+1) &\leq -(1-\eta\kappa_2)r(k) + (1-\eta\kappa_2)\bar{r} \\ &\leq (\eta\kappa_2 - 1)d(k) \end{aligned}$$

and thus (F1) holds.

(4) $r(k) > \overline{r}$ and $r(k+1) > \overline{r}$. In this case $d(k) = \underline{r}(k) - \bar{r}$ and

$$\begin{aligned} d(k+1) &= r(k+1) - \bar{r} \\ &= r(k) + \eta\Big[\lambda\big(r(k)\big) - \lambda^*\Big] - \bar{r} \\ &\leq r(k) - \bar{r} + \eta\kappa_1\big[\bar{r} - r(k)\big] \\ &\leq (1-\eta\kappa_1)\big[r(k) - \bar{r}\big] \\ &\leq (1-\eta\kappa_1)d(k) \end{aligned}$$

that is $d(k+1) \leq (1-\eta\kappa_1)d(k)$, and thus (F1) holds. □