GLOBALLY OPTIMAL DECENTRALIZED SPATIAL SMOOTHING FOR WIRELESS SENSOR NETWORKS WITH LOCAL INTERACTIONS

S. Barbarossa, T. Battisti

INFOCOM Dept., Univ. of Rome "La Sapienza "
Via Eudossiana 18, 00184 Rome, Italy

A. Swami

ARL
Adelphi, MD 20783, USA

ABSTRACT

In most sensor network applications, the vector containing the observations gathered by the sensors lies in a space of dimension equal to the number of nodes, typically because of observation noise, even though the useful signal belongs to a subspace of much smaller dimension. This motivates smoothing or rank reduction. We formulate a convex optimization problem, where we incorporate a fidelity constraint that prevents the final smoothed estimate from diverging too far from the observations. This leads to a distributed algorithm in which nodes exchange updates only with neighboring nodes. We show that the widely studied consensus algorithm is indeed only a very specific case of our more general formulation. Finally, we study the convergence rate and propose some approaches to maximize it.

Index Terms— Distributed smoothing, consensus algorithm, sensor networks.

1. INTRODUCTION

In this paper we consider a wireless sensor network that monitors a field of values, e.g., a field of temperatures or the concentration of a given contaminant. In most cases, the useful signal is a smoothed function, as a result of a diffusion process. However, typically the set of measurements is not at all smoothed because of the observation noise. This motivates the design of sensor networks that are able to apply some kind of filtering or smoothing to the data, in order to limit the influence of noise on the final decision. If possible, it would be important to implement these filtering operations in a distributed way, i.e. without the presence of a sink node that has to collect all the data to apply the required filtering operation. This problem has been studied extensively in the case where the useful signal is homogeneous, that is spatially constant. In such a case, the so called consensus algorithms are able to provide the globally optimal estimate with a network of only locally interacting sensors, see e.g., [1], [2]. Another problem that has received significant attention has been the so called distributed Kalman filtering algorithm, see, e.g. [3], [4], where the goal is to track a non-stationary process, typically modeled as a Gauss-Markov field.

Conversely, the goal of this work is to provide a distributed mechanism to perform a spatial smoothing of a stationary, inhomogeneous field. Furthermore, we incorporate a fidelity constraint in the optimization process to prevent the the final estimate from diverging too far from the observations. In particular, we show that if the useful signal is a continuous function, i.e., it can be locally approximated by a low order polynomial expansion, then the smoothing operation can be implemented with a totally distributed network where each node is linked only to a set of neighbors, with the degree of each node depending only on the degree of the approximating polynomial. Then we generalize the algorithm to handle the case of discontinuous functions. As expected, we will show that the well known consensus algorithm can be seen as a particular case of our more general formulation.

The paper is organized as follows. In Section 2, we formulate the problem and motivate its usefulness. In Section 3, we propose a spatial smoothing mechanism with no fidelity constraint. In Section 4, we add the fidelity constraint. Finally, in Section 5, we show how to maximize the convergence rate.

This work has been partially funded by the WINSOC project, Contract Number 0033914, co-funded by the INFSO DG of the European Commission and by ARL/ERO, Contract N62558-05-P-0458.

2. MOTIVATIONS AND PROBLEM STATEMENT

We consider a sensor network composed of N nodes, each equipped with three basic components:i) a transducer that senses the physical parameter of interest; ii) a discrete dynamical system whose state is initialized with the local measurements; iii) a radio interface that transmits the state of the dynamical system and receives the state transmitted by the other nodes, thus ensuring interaction among the nodes. The observation of the sensor located at (x_i, y_i) is

$$g(x_i, y_i) = f(x_i, y_i) + w(x_i, y_i), \quad i = 1, ..., N \tag{1}$$

where $f(x_i, y_i)$ is the useful field and $w(x_i, y_i)$ is noise. In vector notation, we may write $\boldsymbol{g} = \boldsymbol{f} + \boldsymbol{w}$, with the three vectors having dimension N. For simplicity, we assume that the nodes are uniformly spaced over a 2-dimensional grid and that the observed field does not vary with time. However, the extension to the more general case can be done in a straightforward manner. Our goal is to provide a smoothed version $\hat{f}(x, y)$ of the useful field $f(x, y)$, in order to attenuate as much as possible the effect of noise, using a totally distributed algorithm. If the useful signal is constant, its value can be obtained, for example, with the well known average consensus (AC) algorithm, which is totally distributed. The AC algorithm proceeds as follows. Every node initializes its own state value $z_i(0) = g(x_i, y_i)$ and then it updates this value through linear interactions with other nodes according to the following equation

$$z_i(k+1) = z_i(k) + \varepsilon \sum_{j \in \mathcal{N}_i} a_{ij}(z_j(k) - z_i(k)), \tag{2}$$

where $\mathcal{N}_i$ is the set of neighbors of node i, the coefficients a_{ij} are real and non-negative ($a_{ii} = 0$), and k is the time or iteration index. Although synchronous updates are assumed here, these results extend to the asynchronous case, and to stationary node and link failure models easily. Equation (2) can be recast in matrix form as [1]

$$\boldsymbol{z}(k+1) = (\boldsymbol{I} - \varepsilon \boldsymbol{L})\boldsymbol{z}(k), \tag{3}$$

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2008		**00-00-2008 to 00-00-2008**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Globally Optimal Decentralized Spatial Smoothing for Wireless Sensor Networks With Local Interactions	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **INFOCOM Dept.,,Univ. of Rome ?La Sapienza,Via Eudossiana 18, 00184 Rome, Italy, ,**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002091. Presented at the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2008), Held in Las Vegas, Nevada on March 30-April 4, 2008. Government or Federal Purpose Rights License

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**4**	

where $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ is the graph Laplacian, $\boldsymbol{A}$ is the adjacency matrix, with entries the coefficients a_{ij}, and $\boldsymbol{D}$ is the degree matrix, i.e. a diagonal matrix whose i-th entry is $d_i = \sum_{j \in \mathcal{N}_i} a_{ij}$. For the purpose of this work, the matrix $\boldsymbol{A}$ (and then $\boldsymbol{L}$) may be assumed to be a symmetric matrix. Let $\lambda_N(\boldsymbol{L})$ denote the largest eigenvalue of $\boldsymbol{L}$. It is straightforward to show (see, e.g. [1]) that, if $0 < \varepsilon < 2/\lambda_N(\boldsymbol{L})$ and the graph describing the network is connected and balanced, then the vector $\boldsymbol{z}(k)$ converges to the average consensus, i.e.,

$$\lim_{k\to\infty} \mathbf{z}(k) = \frac{1}{N}\mathbf{1}\mathbf{1}^T \boldsymbol{g}, \tag{4}$$

where $\mathbf{1}$ is the vector with all ones.

This algorithm is totally decentralized, but it represents an extreme form of smoothing, as it destroys any potential spatial variation in the field of interest, which in most cases is spatially inhomogeneous. The goal of this paper is to propose an algorithm to smooth the observation, using a totally distributed approach, but without forcing the final result to converge to a constant value. We formulate our problem as an optimization problem with a fidelity constraint. Formally, we want to find the estimated field $\hat{\mathbf{f}} := (\hat{f}(x_1, y_1), \ldots, \hat{f}(x_N, y_N))$ as the minimizer of the functional

$$J(\hat{\mathbf{f}}) = \mu\, V(\hat{\mathbf{f}}) + (1-\mu)\|\hat{\mathbf{f}} - \boldsymbol{g}\|^2, \tag{5}$$

where $V(\hat{\mathbf{f}})$ is a convex function measuring the smoothness of $\hat{\mathbf{f}}$, the term $\|\hat{\mathbf{f}} - \boldsymbol{g}\|^2$ measures the distance between the final estimate and the initial observation $\boldsymbol{g}$, and μ is a real parameter lying between 0 and 1. By varying μ, we vary the relative importance of smoothing versus that of fidelity to the initial measurement. Since $J(\hat{\mathbf{f}})$ is a convex function of $\hat{\mathbf{f}}$, the minimization of $J(\hat{\mathbf{f}})$ admits a unique solution. What is critical in our approach is that the minimization of (5) has to be achieved using a network with no fusion center, whose nodes exchange information only with its neighbors. This is useful to simplify the operations at each node and, most important, to limit the waste of energy related to the exchange of data among sensors. We will consider first the case of pure smoothing (no fidelity constraint). Then, we will add the fidelity constraint.

3. SPATIAL SMOOTHING WITH NO FIDELITY CONSTRAINT

If the useful field $f(x, y)$ is a continuous function of the spatial coordinates, according to the Weierstrass' theorem it can be approximated by a two-dimensional polynomial of finite order in the variables x and y, with an arbitrarily small error. Let us denote with K the order of the polynomial in both variables x and y. In such a case, we want to minimize the following function

$$V(\boldsymbol{g}) = \frac{1}{2} \sum_{i\in\mathcal{N}} \sum_{m=0}^{K} [\nabla_x^{(K-m)} \nabla_y^{(m)} g(x_i, y_i)]^2, \tag{6}$$

where $\nabla_x^{(m)}$ and $\nabla_y^{(m)}$ denote the $m-$th order difference operator with respect to the variables x and y, respectively. More specifically, the operator is defined through the following properties:

$$\begin{array}{l} \nabla_x^{(0)} g(x_i, y_i) = g(x_i, y_i);\ \nabla_y^{(0)} g(x_i, y_i) = g(x_i, y_i); \\ \nabla_x^{(1)} g(x_i, y_i) = g(x_i, y_i) - g(x_{i-1}, y_i); \\ \nabla_y^{(1)} g(x_i, y_i) = g(x_i, y_i) - g(x_i, y_{i-1}) \\ \nabla_x^{(n)} g(x_i, y_i) = \nabla_x^{(1)} \left[\nabla_x^{(n-1)} g(x_i, y_i)\right]; ... \end{array} \tag{7}$$

To take into account border effects, $\mathcal{N}$ is the set of indices for which the above differences can be properly computed; also recall that we have assumed a uniform 2D grid for simplicity of exposition. Since (6) is a quadratic form on $\boldsymbol{g}$, its minimum can be reached using the steepest descent method

$$\boldsymbol{z}(k+1) = \boldsymbol{z}(k) - \varepsilon \boldsymbol{L}\boldsymbol{z}(k) \triangleq \boldsymbol{W}\boldsymbol{z}(k), \tag{8}$$

with initialization $\boldsymbol{z}(0) = \boldsymbol{g}$, having set $\boldsymbol{W} \triangleq \boldsymbol{I} - \varepsilon \boldsymbol{L}$. It is useful to remark that, by construction $\boldsymbol{L}$ is a *positive semidefinite, symmetric, band* matrix. Hence, there exists a unitary matrix $\mathbf{U}$ and a diagonal matrix Λ with real elements such that:

$$\mathbf{L} = \begin{pmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{pmatrix} \begin{pmatrix} \Lambda_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{U}_1^H \\ \mathbf{U}_2^H \end{pmatrix} \tag{9}$$

where $\mathbf{U}_2$ has dimension $N \times L$, with L denoting the dimension of the kernel of $\boldsymbol{L}$. The columns of $\mathbf{U}_2$ are the vectors spanning the kernel of $\boldsymbol{L}$. Let $\lambda_i(\mathbf{L})$ and $\lambda_i(\mathbf{W})$ denote the eigenvalues of $\boldsymbol{L}$ and $\boldsymbol{W}$; we assume that these eigenvalues are ordered in non-decreasing order. We can always choose ε, so that the eigenvalues of $\mathbf{W}$ satisfy $0 < |\lambda_i(\mathbf{W})| < 1, \forall 1$. This property is clearly achieved by setting

$$0 < \varepsilon < \frac{2}{\lambda_N(\mathbf{L})}. \tag{10}$$

With this choice, it is straightforward to verify that

$$\lim_{k\to\infty} \mathbf{z}(k) = \lim_{k\to\infty} \sum_{i=1}^{N} \lambda_i^k(\mathbf{W}) \mathbf{u}_i \mathbf{u}_i^* \mathbf{z}(0) = \mathbf{U}_2 \mathbf{U}_2^H \mathbf{z}(0), \tag{11}$$

where the columns of $\boldsymbol{U}_2$ are exactly the vectors spanning the kernel of $\boldsymbol{L}$. Hence, expression (11) states that the *final value coincides with the projection of the observation vector onto the nullspace of* $\boldsymbol{L}$. This is an important property that significantly enlarges the possibility of smoothing or, more generally, subspace reduction of the observation, by a proper construction of the matrix $\boldsymbol{L}$. Some examples may be helpful to grasp the structural properties of $\boldsymbol{L}$ and, as a consequence, the final value.

Let us consider the one-dimensional case, for simplicity of notation. In such a case, we have, for example:

a) $K = 1$,

$$\mathbf{L} = \begin{pmatrix} 1 & -1 & 0 & 0 & \ldots & 0 \\ -1 & 2 & -1 & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -1 & 2 & -1 \\ 0 & \ldots & \ldots & 0 & -1 & 1 \end{pmatrix}$$

In this case, the role of $\mathbf{L}$ is the same as the Laplacian matrix in (3), where $a_{ij} = 1$, if $|i - j| = 1$, and 0 otherwise. This happens when each node has only two neighbors (except the border nodes having only one neighbor). The matrix $\boldsymbol{L}$ has, in this case, a null eigenvalue of multiplicity one. Since each row of $\boldsymbol{L}$ has zero row sum, the eigenvector associated with the null eigenvalue of $\boldsymbol{L}$ is the vector $\mathbf{1}$ composed of all ones. Hence, the final result is as in (4), as in the conventional average consensus algorithm.

b) $K = 2$,

$$\mathbf{L} = \begin{pmatrix} 1 & -2 & 1 & 0 & 0 & 0 & \dots & 0 \\ -2 & 5 & -4 & \ddots & \ddots & \ddots & \ddots & 0 \\ 1 & -4 & 6 & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 6 & -4 & 1 \\ \vdots & \ddots & \ddots & \ddots & \ddots & -4 & 5 & -2 \\ 0 & \dots & \dots & \dots & 0 & 1 & -2 & 1 \end{pmatrix}$$

In this case, the nullspace of $\boldsymbol{L}$ has dimensionality two and it is spanned by a linear combinations of polynomials of degree zero and one. An orthonormal set is given, in this case, by the Legendre polynomials of degree zero and one. Hence the final vector is a straight line. In general, it is easy to check that, given the structure of the cost function $V(\boldsymbol{g})$ in (6), the nullspace of $\boldsymbol{L}$ has dimensionality K and it is spanned by the polynomials of orders from 0 to $K-1$. Since any continuous function can be approximated with an arbitrarily small error, by a polynomial, the above method provides then a distributed tool to approximate any continuous field of values. The network degree corresponding to a given value of K is simply $2K$.

An example of the application of the previous method to a 2D field is reported in Fig. 1. In this case, the useful field is a paraboloid and the measurement is corrupted by zero mean white Gaussian noise. The observations are represented, in Fig. 1, by the circles. The final state vector is represented by the paraboloid shown in Fig. 1, which is almost perfectly superimposed on the useful field.

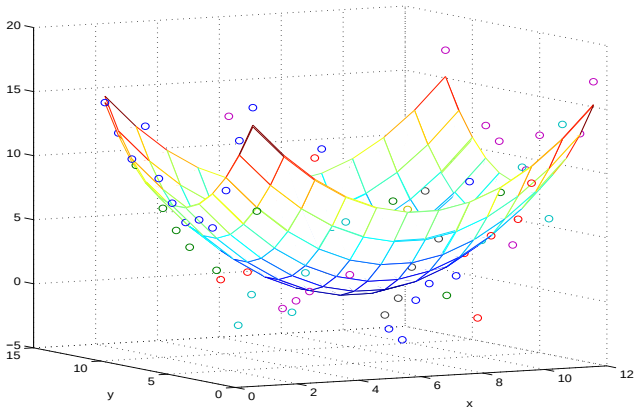

Fig. 1. Reconstruction of a 2D field.

4. SPATIAL SMOOTHING WITH FIDELITY CONSTRAINT

In principle, we can improve the approximation in the above method by increasing the value of K. However, the degree of the network, defined as the maximum number of neighbors of each node, increases with K. This induces a greater waste of energy. Hence, it is necessary to find the right trade-off between energy consumption and approximation error. Furthermore, if the useful field presents discontinuities, we could be forced to use very high values of K and still have a nonnegligible error. It is then useful to devise some variants of the previous method that still allow us to have a limited value of the network degree. To this end, we use now as cost function the expression (5), with $V(\boldsymbol{g})$ given in (6), with $\mu \in (0,1)$. In this case, the steepest descent method leads to

$$\mathbf{z}(k+1) = [(1-\varepsilon(1-\mu))\boldsymbol{I} - \varepsilon\mu\,\mathbf{L}\,]\boldsymbol{z}(k) + \varepsilon(1-\lambda)\,\mathbf{z}(0), \quad (12)$$

with $\boldsymbol{z}(0) = \boldsymbol{g}$. Introducing the matrix $\boldsymbol{W} \triangleq (1-\varepsilon(1-\mu))\boldsymbol{I} - \varepsilon\mu\mathbf{L}$, we can guarantee that the eigenvalues $\lambda_i(\boldsymbol{W})$ of $\boldsymbol{W}$ are strictly between -1 and 1, by setting

$$0 < \varepsilon < \min_i \left\{ \frac{2}{\lambda_i(\boldsymbol{L}) + 1 - \mu} \right\} = \frac{2}{\lambda_N(\boldsymbol{L}) + 1 - \mu}. \quad (13)$$

With a few simple algebraic manipulations, we can rewrite $\boldsymbol{z}(k)$ as

$$\begin{aligned} \boldsymbol{z}(k) &= \boldsymbol{W}^k\mathbf{z}(0) + \varepsilon(1-\mu)\textstyle\sum_{n=0}^{k-1}\mathbf{W}^k\boldsymbol{z}(0) \\ &= \boldsymbol{W}^k\mathbf{z}(0) + \varepsilon(1-\mu)(\mathbf{I}-\mathbf{W})^{-1}(\mathbf{I}-\mathbf{W}^k)\boldsymbol{z}(k). \end{aligned} \quad (14)$$

Choosing ε according to (13), the state vector converges to

$$\lim_{k\to\infty} \mathbf{z}(k) = \left(\boldsymbol{I} + \frac{\mu}{1-\mu}\,\boldsymbol{L}\right)^{-1}\boldsymbol{g}. \quad (15)$$

Depending on the value of μ, we may give different relative importance to smoothing or fidelity to the original observation. In the extreme case of $\mu = 0$, the network does not apply any smoothing, i.e. $\lim_{k\to\infty}\mathbf{z}(k) = \boldsymbol{g}$, whereas, at the other extreme, when $\mu = 1$, the final value coincides with the projection of the observation onto the nullspace of the Laplacian matrix, as proved in the previous section.

A numerical example is reported in Fig.2, relative to a one-dimensional network located over a straight line. The observed signal in this case is a sinusoid (dashed line) and the observation (dots) is corrupted by white Gaussian noise. The SNR is 5 dB. Smoothing has been performed using the simple algorithm (8), with $K = 3$. In this case, with $\mu = 1$, the method projects the observed vector onto the space spanned by second order polynomials. Since the observation is a noisy sinusoid, the final result (dash-dotted line) is not very good. However, as soon as μ is slightly less than one, the method is forced to take into account the fidelity to the observation, and the final result is much better than in the previous case. Using $\mu = 0.9999$, for example, the result of the smoothing operation is represented by the solid line and we can see that the approximation is now pretty good.

Even if this is only a simple example, Fig. 2 suggests that the choice of μ can have a strong impact on the smoothing operation. To quantify the final distortion, we can compute the mean square error, averaged over the noise realizations. Introducing the matrix $\boldsymbol{P}(\mu) = \left(\boldsymbol{I} + \frac{\mu}{1-\mu}\,\boldsymbol{L}\right)^{-1}$, the final MSE is

$$MSE(\mu) = \|(\boldsymbol{P}(\mu) - \boldsymbol{I})\boldsymbol{f}\|^2 + \sigma_n^2 \mathrm{tr}(\boldsymbol{P}(\mu)\boldsymbol{P}(\mu)^T). \quad (16)$$

In the case of a sinusoidal function, this function, normalized to $\|\boldsymbol{f}\|^2$, is reported in Fig. 3, for different values of μ and σ_n^2. As expected, there is an optimal value of μ that depends on the noise level: When there is no noise, it is better to apply no smoothing at all, and thus the best value of μ is zero; conversely, as the noise increases, it is better to use values of μ closer and closer to one.

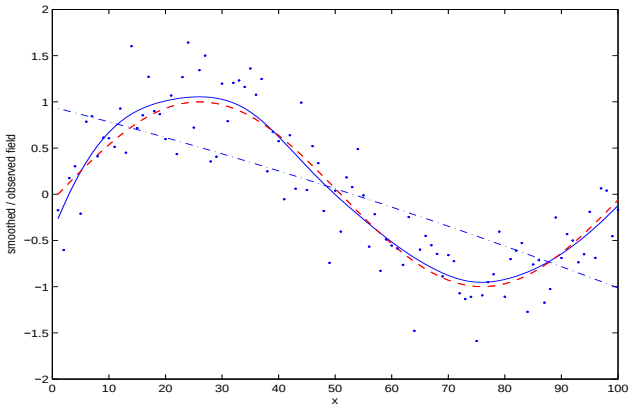

Fig. 2. Reconstruction of a noisy sinusoid.

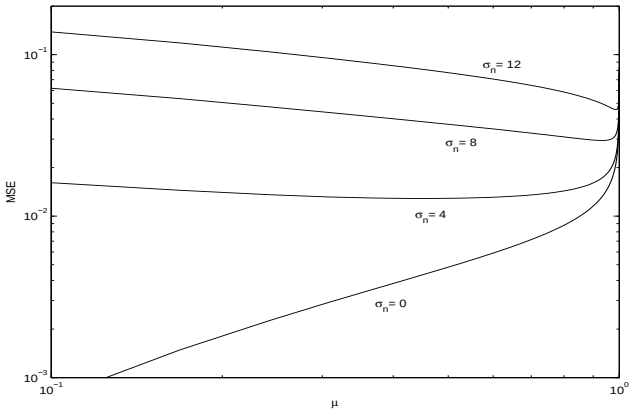

Fig. 3. MSE as a function of μ and σ_n.

5. CONVERGENCE RATE

A critical issue in most distributed processing algorithms to be implemented in decentralized sensor networks is the convergence rate. It is clear in fact that a slow convergence rate implies a waste of time and energy. And energy is an especially critical and scarce resource.

The distributed projection method illustrated in Section 3 requires only that the nullspace of the matrix $\boldsymbol{L}$ to be used for the state update in (8) be composed of the right eigenvectors (polynomials), but it does not impose any condition on the eigenvalues of ($\boldsymbol{L}$). Hence, we can optimize the choice of the eigenvalues in order to maximize the convergence rate.

From (11), the convergence is exponential and the convergence rate is given by the slowest mode, i.e. the highest eigenvalue of $\boldsymbol{W}$ strictly less than 1. This value is $\lambda_{N-1}(\boldsymbol{W}) = 1 - \varepsilon\lambda_2(\boldsymbol{L})$. Requiring that the corresponding mode decays from 1 to α, with $\alpha < 1$, we get a convergence time

$$T_c = \frac{\log \alpha}{\log(1 - \varepsilon\lambda_2(\boldsymbol{L}))}. \qquad (17)$$

From (17), it is clear that the minimization of the convergence time corresponds to minimizing the second smallest eigenvalue of $\boldsymbol{L}$. To make a fair comparison with the method presented in Section 3, with the matrix $\boldsymbol{L}$ resulting from the computation of the gradient of $V(\boldsymbol{g})$ in (6), we impose a constraint on the trace of $\boldsymbol{L}$ and look for the optimal distribution of eigenvalues that provides the largest $\lambda_2(\boldsymbol{L})$. The result is very simple: the optimal set of eigenvalues corresponds to having all the eigenvalues equal to each other and equal to $\mathrm{tr}(\boldsymbol{L})/\mathrm{rank}(\boldsymbol{L})$. We denote by $\boldsymbol{L}'$ the matrix having the same eigenvectors as $\boldsymbol{L}$, same nullspace, but all the nonnull eigenvalues equal to each other. The price to be paid for the use of this matrix

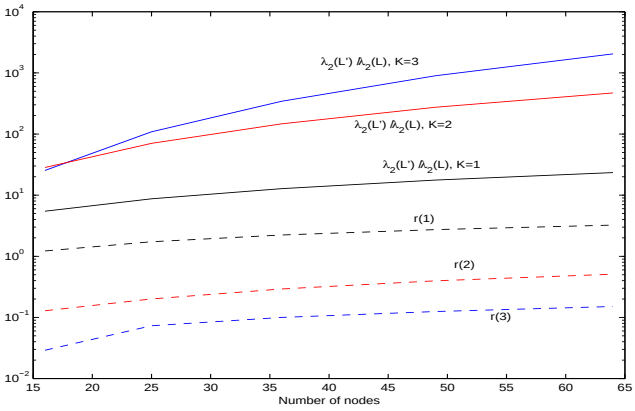

Fig. 4. MSE as a function of μ and σ_n.

$\boldsymbol{L}'$ is a sort of spillover effect: Whereas the matrix $\boldsymbol{L}$ resulting from taking the gradient of $V(\boldsymbol{g})$ in (6) is a band matrix, the matrix $\boldsymbol{L}'$ is not necessarily a band matrix. This means that every node in the network has, in general, a higher degree and this implies a higher transmit power for each node. Setting $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, as in the conventional Laplacian, we studied the spillover ratio $r(K) = \sum_{\mathcal{I}} a_{ij}^2 / \sum_{\mathcal{B}} a_{ij}^2$, where $\mathcal{B}$ is the set of indices falling in the same band as the initial matrix $\boldsymbol{L}$, whereas $\mathcal{I}$ indicates the set of indices falling outside the initial band. In Fig. 4, we report the ratio of the smallest eigenvalues $\lambda_2(\boldsymbol{L}')/\lambda_2(\boldsymbol{L})$ (solid lines), denoting the increase of convergence rate resulting from forcing all the eigenvalues to be equal to each other, for $K = 1, 2,$ and 3. In the same figure, we plot the spill-over ratio $r(K)$, also for $K = 1, 2,$ and 3, as a function of the number of nodes (dashed lines). We can notice how a proper allocation of eigenvalues produces a substantial increase of the convergence rate, more and more evident as the number of nodes increases. At the same time, this determines a higher degree for each node, as evidenced by the dashed line curves, which implies a higher required transmit power at each node. An interesting, still open question, is then how to find out the optimal eigenvalue distribution that minimizes the overall energy required for achieving the final smoothed signal, taking into account both convergence rate and local transmit power.

6. REFERENCES

[1] R. Olfati-Saber, J. A. Fax, R. M. Murray, "Consensus and Cooperation in Networked Multi-agent Systems," in *Proc. of the IEEE*, vol. 95, no. 1, pp. 215-233, Jan. 2007.

[2] S. Barbarossa and G. Scutari, "Bio-inspired Sensor Network Design: Distributed Decision Through Self-synchronization," *IEEE Signal Processing Magazine*, vol. 24, no. 3, May 2007, pp. 26–35.

[3] R. Olfati-Saber, "Distributed Kalman Filtering with Embedded Consensus Filters," in *Proc. of IEEE CDC 2005*, Seville, Spain, 2005, pp. 8179–8184.

[4] I. Schizas, G. B. Giannakis, S. I. Roumeliotis, A. Ribeiro, "Anytime Optimal Distributed Kalman Filtering and Smoothing", in *Proc. of IEEE SSP '07*, pp. 368–372, Aug. 2007.