To appear, 2005 IFAC World Congress
http://www.cds.caltech.edu/~murray/papers/2005g_sm05-ifac.html

DISTRIBUTED SENSOR FUSION USING DYNAMIC CONSENSUS

Demetri P. Spanos * **Richard M. Murray** *

* *California Institute of Technology*

Abstract: This work is an extension to a companion paper describing consensus-tracking for networked agents, and shows how those results can be applied to obtain least-squares fused estimates based on spatially distributed measurements. This mechanism is very robust to changes in the underlying network topology and performance, making it an interesting candidate for sensor fusion on autonomous mobile networks. We conclude with an example of a preliminary application to distributed Kalman Filtering using the proposed technique, illustrating the dependence of the performance on the structure of the underlying network.

Keywords: Agents, Decentralized control, Decentralized systems, Graph theoretic models, Sensor Fusion

1. INTRODUCTION

Sensor networks are a prominent example of cooperative automated and computing technology. Their defining characteristic is the utilization of connections among several low-cost devices in order to obtain global performance vastly in excess of the capabilities of any individual device. Thus, they provide a natural venue for the development of cooperative distributed algorithms. For a general introduction to this area, see (Akyildiz *et al.* 2002).

This work presents a novel approach to distributed least-squares sensor fusion, and a preliminary application to distributed Kalman Filtering. While many results exist broadly within this area, the word *distributed* is somewhat overused, and for the purposes of this paper, it will mean the following:

- The algorithm cannot rely on any particular communication topology, and must be robust to changes in the topology.
- The algorithm must provide *every* member of the network with the optimal estimate, not merely one distinguished coordinator or base station.
- The algorithm must only utilize peer-to-peer interactions, and should not involve global dissemination or "flooding" of local measurements or estimates.

The last two points on this list are significant, as most work in sensor fusion relies on the availability of a fast network for global sharing of local variables. While such an assumption is justified in some cases, it is certainly not justified for situations involving large numbers of sensors, high-dimensional measurements, and an unpredictable low-bandwidth ad-hoc network.

The proposed approach is unusual, in that the fusion process is *dynamical*. The estimates of individual members of the network evolve according to a linear dynamical system, the *Laplacian* consensus dynamics, which represents a discrete version of a diffusion process. This mechanism is extremely robust to the various uncertainties arising in mobile sensor networks, including time-varying link availability, splitting and merging of sub-networks, and unpredictable asynchronous operation. This dynamical approach also allows one to clearly understand the role of network performance in the performance of the sensor fusion algorithm.

The sensor-fusion algorithm presented hinges on the ability to dynamically provide the following two pieces of information to each member of the network:

(1) The least-squares estimate of a vector, based on the measurements of all the agents.
(2) The covariance of this fused least-squares estimate.

Assuming that the measurement errors are independent, and that all members of the network have access to the process model to be used in the Kalman filter, this information is sufficient for each agent to reconstruct the optimal Kalman filter estimate.

In order to provide the above information, two multi-variable modifications of the Laplacian dynamics will be examined, which will allow tracking of arbitrary multi-linear combinations of individual measurement signals, i.e. quantities of the form:

$$\begin{aligned}\bar{\mathbf{z}} &\doteq \sum_i M_i(t)\mathbf{z}_i(t)\\ \mathbf{z}_i(t) &\in \mathbf{R}^m\\ M_i(t) &\in \mathbf{R}^{m\times m}.\end{aligned}$$

One approach will be based on a "naíve" LTI implementation, while the other will be an LPV approach. The latter, though more difficult to analyze, results in significant communication savings.

2. BACKGROUND

This work is primarily based on work arising in distributed coordination problems, but our target application is in sensor networks and decentralized estimation. We refer the reader to the works of (Estrin *et al.* 1999), (Ogren *et al.* 2004), (Heinzelman *et al.* 1999), and references therein for various aspects of this problem. The reader is also directed to related work in (Xiao *et al.* 2005).

Within the area of distributed coordination we focus on consensus problems, i.e. situations in which all members of some network are required to achieve some common output value using only local interactions and without access to a global coordinator. There is a large literature on general consensus problems in the Computer Science community, e.g. (Lynch 1997). Recently, consensus problems have also attracted interest in the Control community; the works of (Jadbabaie *et al.* 2002) and (Moreau 2003) are representative of general coordination problems, and the work of (Fax and Murray 2004) analyzes a specific coordination problem involving vehicle formations. Our work closely mirrors the philosophy and analysis techniques of the latter.

We are specifically interested in weighted-average consensus, in which each member of the network must converge to a specified weighted average of the input values. The static scalar case was examined comprehensively in (Olfati-Saber and Murray 2004), and was accomplished using a distributed dynamics utilizing the *Laplacian* matrix of the communication network. The dynamics of each agent takes the form

$$\dot{x}_i = \frac{1}{w_i}\sum_{j\in N_i}(x_j - x_i)\,, \tag{1}$$

where w_i is a positive weight and N_i is the *neighborhood* of agent i, i.e. the set of agents with which it can communicate. This is a stable linear dynamical system and so converges exponentially (see (Olfati-Saber 2005) for an extensive treatment of convergence rate). It was shown that under a fairly general model of switching topology, this dynamics drives all agents' states to the following weighted average of the initial states:

$$x_i(t) \to \frac{\sum_i w_i x_i(0)}{\sum_i w_i} \quad \text{for all } i \in V. \tag{2}$$

Note that the above weighted average has the structure of a least-squares estimator for the mean of a Gaussian random variable based on statistically independent observations, assuming the weights w_i are interpreted as reciprocals of variances.

In this work we will also make use of a dynamic extension presented in a companion paper (Spanos *et al.* 2005), whose simplest form is just

$$\begin{aligned}\dot{x}_i &= \frac{1}{w_i}\sum_{j\in N_i}(x_j - x_i) + \dot{z}_i\\ x_i(0) &= z_i(0).\end{aligned}$$

It is shown in the companion paper that this algorithm (and other similar designs) makes the x_i variables track the weighted average

$$\frac{\sum_i w_i z_i(t)}{\sum_i w_i}. \tag{3}$$

It was also shown that this mechanism can be modified (preserving the distributed structure) to handle two additional phenomena common on mobile wireless networks: arbitrarily large non-uniform time-delays, and splitting and merging of sub-networks. All of the results in this paper exploit the tracking result mentioned above.

Finally, we refer the reader to (Mehyar *et al.* 2005), which discusses how to implement the Laplacian consensus dynamics in a truly asynchronous peer-to-peer environment. Specifically, the results of this work show that exponential convergence can be guaranteed even in the asynchronous case, under a very general assumption about the asynchronous timing.

3. NOTATION

Consider a set V of N interconnected agents, labeled by an index $i = 1, 2, \dots, N$. We model their communication as a connected graph $G = (V, E)$, where an edge (i, j) is in E if and only if agents i and j can communicate. We will only consider bidirectional communication patterns, i.e. $(i, j) \in E \Leftrightarrow (j, i) \in E$. The notation N_i will denote the *neighborhood* of node I, i.e. the set of nodes to which i is connected in the graph G. We will assume that the number of nodes on the network, N, is known to all agents; this can easily be accomplished using distributed methods (including the methods from this paper), but we do not discuss them in detail.

Each of the agents has a local measurement input $z_i(t) \in \mathbf{R}^m$. The agents also each have an associated positive-definite "weight" matrix, $W_i(t) \in \mathbf{R}^{m\times m}$. We wish the agents to track the quantity

$$\bar{\mathbf{z}} \doteq \left(\sum_i W_i(t)\right)^{-1}\left(\sum_i W_i(t)\mathbf{z}_i(t)\right) \qquad (4)$$

If the $\mathbf{z}_i$ terms are interpreted as independent unbiased noisy observations of a physical variable $\mathbf{p}(t) \in \mathbf{R}^m$ (each observation with multivariate Gaussian distribution), and the W_i terms are interpreted as inverse covariance matrices, then this quantity is just the instantaneous least-squares estimate of $\mathbf{p}(t)$.

In order to accomplish the tracking, each agent will maintain a local estimate variable $\mathbf{x}_i$, and the dynamics of this variable will be linked to the values of the $\mathbf{x}_j$ within the neighborhood N_i. We will discuss two main approaches to this problem, one which is two parallel LTI systems (a useful approach for performance analysis), and one which is a single LPV system (which requires significantly less communication).

4. STATIC MULTI-VARIABLE CONSENSUS

First we address the static multi-variable weighted-average problem, in which the weights W_i are constant in time. If the weights are diagonal, then this is just m decoupled scalar dynamic consensus problems, and the algorithm and proof found in (Olfati-Saber and Murray 2004) apply. For non-diagonal weights, i.e. situations in which consensus in one dimension is coupled with consensus in another dimension, we must make a small modification. The following two results are extremely simple, but they form the basis for the remainder of the analysis.

Proposition 1. Suppose the graph G is connected, W_i are positive definite elements of $\mathbf{R}^{m\times m}$, and consider the dynamics

$$\dot{\mathbf{x}}_i = W_i^{-1}\sum_{j\in N_i}(\mathbf{x}_j - \mathbf{x}_i)$$
$$\mathbf{x}_i(0) = \mathbf{z}_i.$$

Then, we have for all $i \in V$:

$$\lim_{t\to\infty}\mathbf{x}_i(t) = \left(\sum_{i\in V} W_i\right)^{-1}\left(\sum_{i\in V} W_i\mathbf{z}_i\right) \qquad (5)$$

PROOF. [Sketch] It is straightforward to derive the following conservation property:

$$\left(\sum_{i\in V} W_i\mathbf{x}_i(t)\right) = \left(\sum_{i\in V} W_i\mathbf{z}_i\right) \text{ for all } t.$$

Exploiting the connectedness of G, it can also be shown that all equilibria of the dynamics satisfy

$$\mathbf{x}_i = \mathbf{x}_j \text{ for all } i, j \in V.$$

Finally, by using the positive-definiteness of the weights W_i, one can show that this dynamics is stable, and so must converge to an equilibrium. This, combined with the above two equations implies the desired result.

□

This result can be applied to obtain a least-squares estimate of a single set of spatially distributed measurements. In order to obtain the covariance of this least-squares estimate, we simply need to carry out a consensus on the inverses of the covariance matrices.

Corollary 2. Suppose the graph G is connected, and consider the following dynamics, with states $V_i \in \mathbf{R}^{m\times m}$ and nonlinear outputs $Y_i \in \mathbf{R}^{m\times m}$:

$$\dot{V}_i = \sum_{j\in N_i}(V_j - V_i)$$
$$Y_i = \frac{1}{N}V_i^{-1}$$
$$V_i(0) = W_i^{-1}$$

Then, we have for all $i \in V$

$$\lim_{t\to\infty} Y_i(t) = \left(\sum_i W_i^{-1}\right)^{-1}.$$

Thus the outputs Y_i all converge to the harmonic mean of the input matrices W_i, divided by N. Again, if we interpret the W_i matrices as covariance matrices, then this limiting value is just the covariance of the least-squares estimate (see any reference on Kalman filtering for details).

With the simple results of this section, we can obtain the least-squares estimate based on a single set of measurements, and also obtain the covariance of this estimate in a distributed way. The following two sections show two methods to extend this to the dynamic case, although the treatment is necessarily cursory due to length limitations.

5. DYNAMICALLY WEIGHTED MULTI-VARIABLE CONSENSUS: LTI

We now allow both the inputs $\mathbf{z}_i(t)$ and the weighting matrices $W_i(t)$ to vary in time. Because the "input" terms $W_i\mathbf{z}_i$ enter as a product, it is difficult to cast this as a single LTI multi-variable consensus problem. Instead, we will decompose the problem into two sub-problems analyzable using the methods from the companion paper: average consensus on the *weighted outputs*, and average consensus on the *weighting matrices*.

We will refer to the Laplace transforms of the signals $\mathbf{z}_i$, denoted $\mathbf{Z}_i(s)$, as well as of the weighting matrices $W_i(t)$, denoted $\mathbf{W}_i(s)$. We will also append to each agent an additional matrix-valued state, $M_i \in \mathbf{R}^{m\times m}$.

Proposition 3. Suppose the graph G is connected, and that the signals $\mathbf{Z}_i(s)$ and $\mathbf{W}_i(s)$ have all their poles in the left-half plane, with at most one pole at $s = 0$. Denote the steady-state values of these signals $\mathbf{z}_i(\infty)$ and $W_i(\infty)$. Consider the following dynamics (with nonlinear outputs $\mathbf{y}_i \in \mathbf{R}^m$):

$$\begin{aligned}
\dot{\mathbf{x}}_i &= \sum_{j\in N_i} (\mathbf{x}_j - \mathbf{x}_i) + W_i\dot{\mathbf{z}}_i + \dot{W}_i\mathbf{z}_i,\\
\dot{M}_i &= \sum_{j\in N_i} (M_j - M_i) + \dot{W}_i,\\
\mathbf{y}_i &= M_i^{-1}\mathbf{x}_i,\\
\mathbf{x}_i(0) &= W_i(0)\mathbf{z}_i(0),\\
M_i(0) &= W_i(0).
\end{aligned}$$

Then we have, for all $i \in V$

$$\lim_{t\to\infty} \mathbf{y}_i(t) = \left(\sum_{i\in V} W_i(\infty)\right)^{-1} \left(W_i(\infty)\mathbf{z}_i(\infty)\right),$$

i.e. the outputs all track the weighted-average consensus with zero steady-state error.

PROOF. [Sketch] The work in the companion paper (Spanos *et al.* 2005), combined with Proposition 1 (for the extension to the multi-variable case) shows that, for all i, the $\mathbf{x}_i$ and M_i states satisfy

$$\begin{aligned}
\lim_{t\to\infty} \mathbf{x}_i &= \frac{1}{N}\sum_{i\in V} W_i(\infty)\mathbf{z}_i(\infty)\\
\lim_{t\to\infty} M_i &= \frac{1}{N}\left(\sum_{i\in V} W_i(\infty)\right).
\end{aligned}$$

This, combined with the output equation, implies the desired result.

□

The "communication complexity" of this dynamics is $O(m^2)$, in the sense that both the vectors $\mathbf{x}_i$ and the matrices M_i must be shared across links. Even if one exploits the symmetries of covariance matrices, the agents must still exchange $\frac{1}{2}\left(m^2 + m\right)$ scalars across each link. If one also wishes to obtain an estimate of the covariance using the mechanism in the previous section, one must share an additional $\frac{1}{2}\left(m^2 - m\right)$ variables, for a total of m^2.

6. DYNAMICALLY WEIGHTED MULTI-VARIABLE CONSENSUS: LPV

We again consider the case of time-varying measurement signals $\mathbf{z}_i(t)$ and weighting matrices $W_i(t)$, but will now present an LPV system that does not require the appended matrix states M_i, or the nonlinear outputs $\mathbf{y}_i$. The price to be paid is that the transient performance of the following LPV system is difficult to analyze, and depends on the properties of the evolution of the weighting matrices $W_i(t)$. We present the following proposition without proof, as it is a straightforward application of ideas already discussed.

Proposition 4. Suppose the graph G is connected, and that the signals $\mathbf{Z}_i(s)$ and $\mathbf{W}_i(s)$ have all their poles in the left-half plane, with at most one pole at $s = 0$. Denote the steady-state values of these signals $\mathbf{z}_i(\infty)$ and $W_i(\infty)$. Consider the following dynamics:

$$\begin{aligned}
\dot{\mathbf{x}}_i &= W_i^{-1}\left(\mathbf{x}_j - \mathbf{x}_i\right) + \dot{\mathbf{z}}_i + W_i^{-1}\dot{W}_i\left(\mathbf{z}_i - \mathbf{x}_i\right)\\
\mathbf{x}_i(0) &= \mathbf{z}_i(0).
\end{aligned}$$

Then we have, for all $i \in V$

$$\lim_{t\to\infty} \mathbf{x}_i(t) = \left(\sum_{i\in V} W_i(\infty)\right)^{-1} (W_i(\infty)\mathbf{z}_i(\infty)) .$$

This LPV mechanism only requires that m variables be shared across each link, and so represents a significant reduction in the amount of communication required to obtain asymptotic tracking. If one also wishes to obtain the covariance, then the methods from Section 4 must be applied, and so the resulting design requires $\frac{1}{2}(m^2+m)$ variables to be exchanged across each link.

7. PRELIMINARY APPLICATION: A DISTRIBUTED KALMAN FILTER

We now outline how to apply the least-squares tracking algorithm to obtain a distributed Kalman filter. We show only the synchronous discrete-time LTI case for simplicity.

Suppose then, that a process $\mathbf{p}(t) \in \mathbf{R}^m, t \in \mathbf{N}$ evolves according to the discrete-time model

$$\mathbf{p}(t+1) = A\mathbf{p}(t) + Q_t$$

where Q is zero-mean Gaussian white noise of known covariance W_Q. The initial condition $\mathbf{p}(0)$ is also assumed Gaussian with expectation $\bar{\mathbf{p}}$ and covariance W_p.

The agents each observe the process at every time-step according to

$$\mathbf{z}_i(t) = \mathbf{p}(t) + \mathbf{n}_i(t)$$

where the $\mathbf{n}_i$ are independent zero-mean Gaussian random variables with covariance $W_i(t)$. The agents will each maintain four variables to carry out the estimation process: the vectors $\hat{\mathbf{p}}_i(t)$ and $\mathbf{x}_i(t)$ in $\mathbf{R}^m$, and the matrices $\hat{P}_i(t)$ and $M_i(t)$. The two vectors will be the estimate states of the Kalman filter and the consensus loop, respectively, while the two matrices represent (approximations to) the covariance of the Kalman filter estimate and the covariance matrix of the globally fused measurements.

The algorithm presented is parametrized by an integer $n \geq 0$, which indicates how many consensus updates are allowed per Kalman filter update. Theis number represents some separation of time-scales; the network should be faster than the physical dynamics if one expects to be able to use all the information available in producing estimates.

The algorithm consists of two loops: an outer loop for the Kalman filter, and an inner loop for the consensus updating. Only the Kalman filter loop is indexed, for notational clarity.

(1) Initialize $\hat{\mathbf{p}}_i(0) = \bar{\mathbf{p}}$ and $\hat{P}_i(0) = W_p$.
(2) Observe the measurements at $t = 0$, and initialize $\mathbf{x}_i(0) = W_i\mathbf{z}_i(0)$, $M_i(0) = W_i(0)$, and $V_i(0) = W_i^{-1}$.
(3) Run n iterations of the weighted dynamic consensus algorithm to obtain $\tilde{\mathbf{x}}_i(0)$, $\tilde{W}_i(0)$.
(4) Use the output equations from Sections 4 and 5 to obtain $\mathbf{y}_i$ and Y_i.
(5) Treating the $\tilde{\mathbf{y}}_i$ and $\tilde{Y}_i$ as observation and covariance, fuse with the prior estimate $\hat{\mathbf{p}}_i(0)$.
(6) Using the process model, propagate the least-squares estimate and covariance forward in time.
(7) Observe the new measurements $\mathbf{z}_i(1)$ and the error covariance $W_i(1)$, and apply the corresponding inputs to the consensus dynamics.
(8) Repeat from Step 2.

Some comments are in order at this point. First, for $n = 0$, this is just a local Kalman filter for each agent, using only local measurements. Second, as $n \to \infty$, the consensus loop drives the local $\mathbf{x}_i$ and M_i variables to the *global* least-squares estimate and covariance. For intermediate values of n, the consensus estimate varies "smoothly" as a function of n. Specifically, the convergence is exponential in the product of n and the algebraic connectivity of the network λ_2 (See (Olfati-Saber and Murray 2004) for details). Thus, one can understand the performance of this approximation algorithm both as a function of connection *speed* (directly related to n), and connection *density* (directly related to λ_2). This is presumably useful in designing distributed feedback loops around the distributed estimator, where a designer will want some quantifiable measure of performance.

An example of this algorithm in action is depicted in Figure 2, where nine agents attempt to track an object moving a circular trajectory. The process model is a discretization of $\ddot{x} = w$, and the covariance of the measurements depends quadratically on the distance between the sensor and the object. The performance of this algorithm as a function of the number of messages exchanged is shown in Figure 3.

8. SUMMARY

We have shown how to utilize weighted-average consensus as a tracking mechanism for distributed least-squares estimation, and also as a tool for distributed calculation of the associated covariance. This performance is analyzable in terms of the algebraic connectivity of the communication network, and the speed of the consensus loop (network dynamics) relative to the Kalman filter loop (physical process dynamics). Our approach also inherits various robustness properties of the

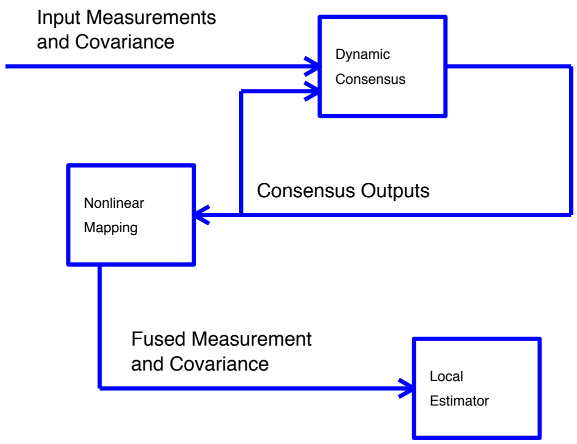

Fig. 1. The structure of the distributed Kalman filter; a dynamical algorithm is used to track the optimal fusion of the spatially distributed inputs, and this is then passed to local estimators.

Laplacian consensus dynamics, including the ability to handle variable topologies, splitting and merging, large delays, and asynchronous operation.

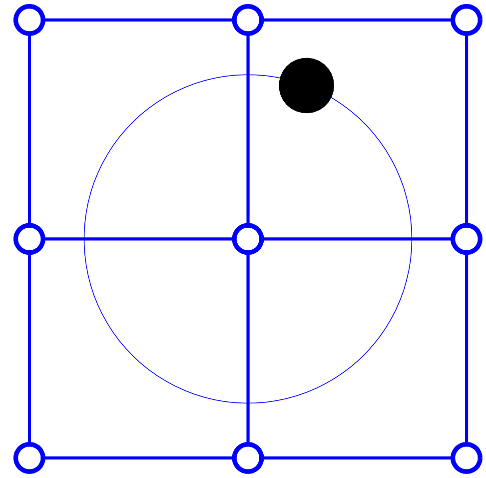

Fig. 2. The object to be tracked moves in a circular trajectory, and the network of agents attempts to maintain an estimate of its position using the proposed distributed Kalman filter.

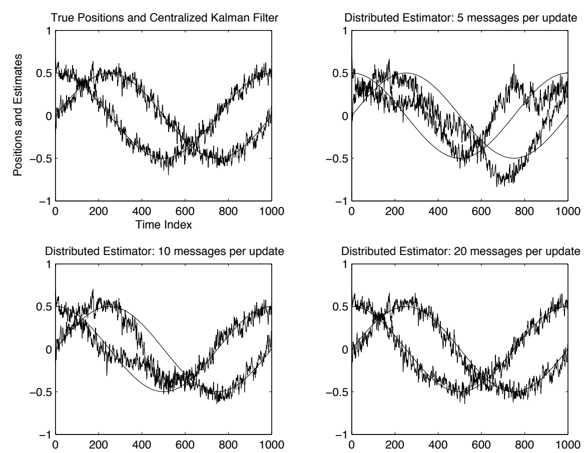

Fig. 3. The performance of the distributed estimator as the network speed is varied; the estimate trajectories shown are the worst-case in mean-square error across the network. As the number of message exchanges increases, the estimate approaches the centralized Kalman filter.

REFERENCES

Akyildiz, I., W. Su, Y. Sankarasubramniam and E. Cayirci (2002). A survey on sensor networks. *IEEE Communications Magazine* pp. 102–114.

Estrin, D., R. Govindan, J. Heidemann and S. Kumar (1999). Next century challenges: scalable coordination in sensor networks. *Proc. of Mob. Comp. and Net.*

Fax, J.A. and R. M. Murray (2004). Information flow and cooperative control of vehicle formations. *IEEE Transaction on Automatic Control.*

Heinzelman, W. R., J. Kulik and H. Balakrishnan (1999). Adaptive protocols for information dissemination in wireless sensor networks. *Proc. of Mob. Comp. and Net.*

Jadbabaie, A., J. Lin and A. S. Morse (2002). Coordination of groups of mobile autonomous agents using nearest neighbor rules.. *Proceedings of the CDC.*

Lynch, N. (1997). *Distributed Algorithms.* Morgan Kauffman Publishers.

Mehyar, M., D.P. Spanos and S.H. Low (2005). Distributed averaging on a peer-to-peer network. *Preprint (Submitted to IEEE Infocom).*

Moreau, L. (2003). Leaderless coordination via bidirectional and unidirectional time-dependent communication. *Proceedings of the CDC.*

Ogren, P., E. Fiorelli and N. E. Leonard (2004). Cooperative control of mobile sensor networks: adaptive gradient climbing in a distributed environment. *IEEE Trans. on Automatic Control.*

Olfati-Saber, R. (2005). Ultrafast consensus on small-world networks. *Proceedings of ACC.*

Olfati-Saber, R. and R. M. Murray (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control.*

Spanos, D.P., R. Olfati-Saber and R.M. Murray (2005). Dynamic consensus on mobile networks. *Preprint (Submitted to IFAC).*

Xiao, L., S. Lall and S. Boyd (2005). A scheme for asynchronous sensor fusion based on average consensus. *Proceedings of IPSN.*