2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Multiscale Consensus for Decentralized Estimation and Its Application to Building Systems

Jong-Han Kim, Matthew West, Eelco Scholte, and Satish Narayanan

Abstract— **Multiscale approaches to accelerate the convergence of decentralized consensus problems are introduced. Consecutive consensus iterations are executed on several scales to achieve fast convergence for networks with poor connectivity. As an example the proposed algorithm is applied to the decentralized Kalman filtering problem for estimation of contaminants in building systems. Two conventional observers are designed and convergence is compared with respect to the number of communications necessary, which is an effective measure of system complexity. It is demonstrated that the proposed multiscale scheme substantially accelerates the decentralized consensus. Future extensions and directions are briefly summarized.**

I. INTRODUCTION

Decentralized estimation and control techniques have been extensively studied as a robust and scalable solution for large scale systems with uncertainty. The classical work was initiated in the 1970s [1] and there has been much recent interest due to the development of the cooperative concept for networked robotic agent systems [2]–[3].

The decentralized Kalman filtering problem is among the results of those efforts [3]. It can be interpreted as a combination of individual Kalman filters with a *consensus* scheme: an iterative scheme to have every node compute the average value of specific quantities. During the consensus iteration, every node communicates specific data with connected nodes to update their data. As this consensus limits to the exact average, the decentralized Kalman filter of every node converges to the centralized Kalman filter. Such consensus problems also occur in the area of multi-agent coordination. e.g., formation, alignment, decision making, synchronization, data fusion, and so on [4].

However, existing consensus schemes require a great deal of communication or unrealistically dense network topologies to ensure acceptable convergence in practice when applied to large scale networks. Ironically, this prevents the practical implementation of decentralized estimation techniques for large scale real world problems with limited bandwidth, even though they were originally aimed at exactly such large systems. There have been a variety of efforts to produce fast convergence of decentralized consensus [5]–[7].

This work was conducted during the 2007 summer internship for J.-H. Kim and funded by United Technologies Research Center. The authors are grateful to Dr. Andrzej Banaszuk for insightful discussions and guidance throughout this effort.

J.-H. Kim is with the Department of Aeronautics and Astronautics, Stanford University. jonghank@stanford.edu

M. West is with the Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign. mwest@uiuc.edu

E. Scholte and S. Narayanan are with the United Technologies Research Center (UTRC). {scholte, narayas}@utrc.utc.com

On the other hand, the *multigrid* computational method, which was originally developed for efficiently solving elliptic boundary value problems, is an example of a scalable linear iterative solvers and is a well-established technique for solving large-scale problems. The method builds a hierarchy in the state domain and separates solving the various wave-number components on different layers, thus quickly decaying various scales of different wave-number components of the residual. This results in the accelerated convergence when applied with conventional iterative relaxation schemes (e.g., Jacobi or Gauss-Seidel) [8]–[9].

This research reported here was initiated by observing that the slow convergence of decentralized consensus methods is similar to what is encountered in conventional relaxation schemes; high wave-number components diminish quickly in several iterations, but after that the nodes do not update very much since they have local information only, causing a deceleration. Therefore a multiscale scheme can accelerate the slow convergence of decentralized consensus. The basic idea of this is to construct a virtual multilevel hierarchy, across which the local information is passed to distant nodes.

We demonstrate this simple concept in a basic consensus problem with poor network connectivity, and it is shown that the proposed scheme substantially accelerates the convergence of the decentralized estimation of contaminants in building systems. Some advantages and disadvantages are discussed and future research directions are described.

II. BUILDING SYSTEM PROBLEM

Real-time knowledge of dynamic indoor environment parameters, such as occupant distribution as well as thermal and airflow state, is critical to the management and optimization of building energy, occupant comfort and safety (such as from fire related threats or malicious attacks).

The state-of-the-art in modeling indoor thermal, airflow and other gaseous flow phenomena is the use of computational fluid dynamics simulations which are not suitable for real-time applications. Methods to estimate and control the multiscale, spatially distributed, dynamic indoor airflow, smoke flow and contaminant transport phenomena in real-time are lacking. Reduced-order representations of such multiscale dynamic phenomena (such as with Galerkin models extracted from lower dimensional projections of the governing equations) can be used for this but suffer from a lack of accuracy. On the other hand, the pervasive use of sensors enables access to information which can be used for real-time monitoring purposes in concert with the reduced-order models. Furthermore, advent of wireless sensor tech-

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

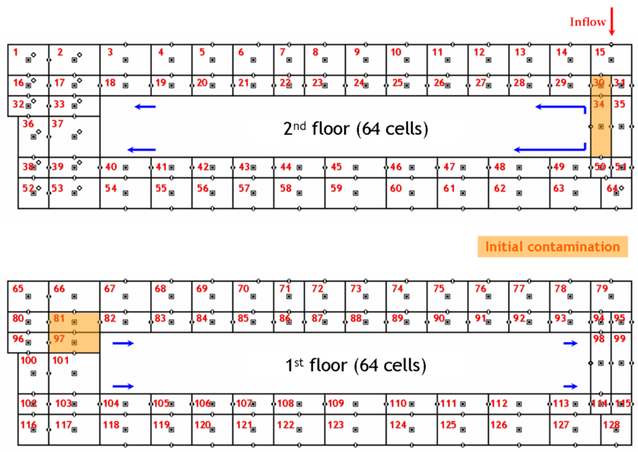

Fig. 1. Building geometry model with 128 nodes.

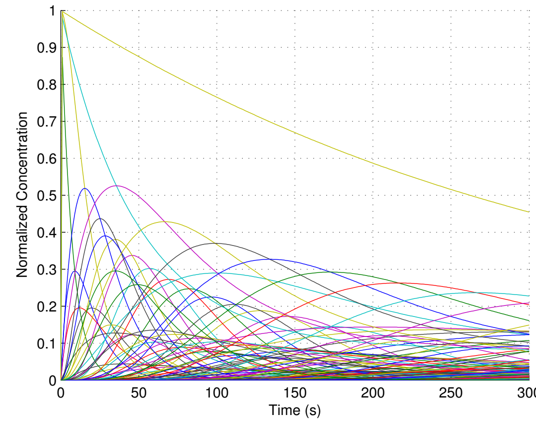

Fig. 2. Impulse response of the building system. Different lines show the contaminant concentrations in the various nodes of the building.

nologies enables distributed computations to support real-time building environment monitoring applications.

Techniques for computationally efficient real-time monitoring and distributed estimation are needed, and are the focus of this paper.

A. *Dynamic model*

The proposed algorithm is applied to a decentralized contaminant estimation problem in a building system. The building of interest is modeled by 128 nodes representing lumped elements, in which the flow and contamination properties are uniform. A constant inward flow of air is introduced at a corner on second floor, and the outflow openings exist wherever the windows are open to the outside. With these boundary conditions and the trace contaminant assumption (the contaminant concentration does not affect the flow field), the contaminant transport model reduces to a simple linear time-invariant system. For more detailed description on the dynamic model, see [10]. The model is

$$\rho_i V_i \frac{dC_i}{dt} = \sum_j F_{ji} C_j - \sum_j F_{ij} C_i + G_i - R_i C_i$$

$$\begin{aligned} \rho &: \text{Density} \\ V &: \text{Volume} \\ C &: \text{Contaminant concentration} \\ F_{ji} &: \text{Mass flow rate from node } j \text{ to } i \\ G &: \text{Contaminant generation rate} \\ R &: \text{Mass removal rate} \end{aligned}$$

where F_{ji} is nonzero only when the node i and j are adjoining nodes.

Among the 128 nodes, four nodes are assumed to have high initial contaminant concentration $C_{\max}$. A typical impulse response to this initial condition is plotted in Fig. 2.

For the estimation problem, we assume that the measurement units (sensors with a small CPU and communication unit inside) are located at the odd numbered nodes, with the sensor communication link established along a single line (i.e., $1 \leftrightarrow 3 \leftrightarrow 5 \leftrightarrow \cdots\cdots \leftrightarrow 125 \leftrightarrow 127$).

III. Decentralized Kalman Filter

The decentralized Kalman filter can be interpreted as three consecutive steps: *prediction*, *consensus*, and *measurement update*. Two of them (prediction and measurement update) are identical to those of the conventional centralized Kalman filter. During the consensus step the system dynamics is assumed to be in steady state. A variety of distributed Kalman filters using consensus algorithms can be found in [3], though the simplest form is applied in this paper.

The inverse covariance form of the Kalman filter is frequently used for decentralized applications, since the measurement information is easily decoupled and distributed. Additionally it is numerically more stable and does not require an accurate estimate of the unknown initial state [11].

A. *Prediction*

The information matrices and vectors are propagated individually by every node. For node i,

$$\begin{aligned} Y_i(t|t-1) &= \left(F Y_i^{-1}(t-1|t-1) F^T + Q\right)^{-1} \\ y_i(t|t-1) &= Y_i(t|t-1) F Y_i^{-1}(t-1|t-1) y_i(t-1|t-1) \end{aligned}$$

where $Y_i = P_i^{-1}$ and $y_i = Y_i \hat{x}_i$ are called the information matrix and the information state of node i, respectively. P_i and $\hat{x}_i$ represent the covariance and the state estimate of node i. F denotes the system transition matrix and Q is the process covariance matrix. The state estimate of node i is given by $\hat{x}_i = Y_i^{-1} y_i$.

B. *Consensus*

Consensus is an iterative process to let every node in a networked group of n nodes on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ asymptotically compute the average value of specific information using only local communication. Whenever a node takes a new measurement $z_i(t)$, where $i \in \mathcal{V}$ denotes the node number, it initializes the information contribution $S_i(t, 0)$

and $s_i(t,0)$, then updates them based on the available local knowledge. A variation of the space-time diffusion scheme [12] is employed here for the consensus algorithm. Note that we have two different time indices; t is for time, and k is for the consensus sweeps.

Initialize:

$$S_i(t,0) = H_i^T R_i^{-1} H_i$$
$$s_i(t,0) = H_i^T R_i^{-1} z_i(t)$$

Consensus:

$$S_i(t,k+1) = S_i(t,k) + \sum\nolimits_{(i,j)\in\mathcal{E}(k)} d_j(t,k) w_{ij}(t,k)\left(S_j(t,k) - S_i(t,k)\right)$$
$$s_i(t,k+1) = s_i(t,k) + \sum\nolimits_{(i,j)\in\mathcal{E}(k)} d_j(t,k) w_{ij}(t,k)\left(s_j(t,k) - s_i(t,k)\right)$$

where H_i denotes the measurement matrix and R_i is the noise covariance matrix.

The weights w_{ij} are generally chosen to yield fast consensus. Although numerical techniques to compute the optimal weights (for the fastest mixing) were suggested recently [5], a simple heuristic choice, the Metropolis weight, is still a strong candidate for decentralized consensus in that it requires only knowledge of the local topology.

We call $d_i(t,k)$ the sweep degree, which represents the total number of consensus sweeps that node i has taken to time t and k sweeps. By introduction of the sweep degree, the information from the nodes with different sweep degrees is weighted unequally. i.e., we believe that the information from a "more experienced" node is more reliable. In the conventional synchronous consensus, $d_i(t,k)$ is same for all i since $d_i(t,k)$ increases equally for every node, therefore the composite weights $d_i(t,k)w_{ij}(t,k)$ are time invariant.

Space-sweep degree:

$$d_i^{SS}(t,k) = d_i(t,k) + \sum_{(i,j)\in\mathcal{E}(k)} d_j(t,k)$$

Metropolis weights:

$$w_{ij} = \frac{1}{\max\left\{d_i^{SS}(t,k), d_j^{SS}(t,k)\right\}} \quad \text{if } (i,j)\in\mathcal{E}(k)$$

As the consensus reaches the average, the information contribution of each node converges to the following, where n represents the total number of sensor nodes.

$$\lim_{k\to\infty} S_i(t,k) = \frac{1}{n}\sum_{i=1}^{n} H_i^T R_i^{-1} H_i = \frac{1}{n} H^T R^{-1} H$$
$$\lim_{k\to\infty} s_i(t,k) = \frac{1}{n}\sum_{i=1}^{n} H_i^T R_i^{-1} z_i(t) = \frac{1}{n} H^T R^{-1} z(t)$$

The right-hand-side terms are the information contributions of all the measurements, which may be computed as for the centralized Kalman filter. Therefore, every node is now able to reconstruct the information contribution for the centralized Kalman filter. i.e., as the consensus converges asymptotically, all individual filters converge to the optimal centralized estimator.

C. Measurement Update

Each node reconstructs the information contribution from its own averaged value. Thus the measurement update is simply

$$Y_i(t|t) = Y_i(t|t-1) + H^T R^{-1} H = Y_i(t|t-1) + nS_i(t,k)$$
$$y_i(t|t) = y_i(t|t-1) + H^T R^{-1} z(t) = y_i(t|t-1) + ns_i(t,k)$$

All the nodes are guaranteed to have the same state estimates as that of the centralized Kalman filter if ideal consensus is provided. Hence, the decentralized Kalman filtering problem is reduced to the consensus problem in the quasi steady-state sense with this formulation.

IV. Multiscale Consensus

A. Introductory Example

In this introductory problem, we have 64 sensors connected in a single line (tridiagonal graph Laplacian). The sensors have randomly distributed initial measurements, and we will test the consensus algorithm to track how the estimates converge to the average of initial measurements.

The consensus history by a conventional scheme is shown in Fig. 3. It is observed that the high wave-number components diminish rapidly in several consensus sweeps, resulting in a smooth profile and slow convergence thereafter. This premature stagnation occurs because the low wave-number components correspond to the slowly decaying modes (eigenvalues close to 1) of the weight matrix, whereas the high wave-number components are associated with rapidly decaying small eigenvalues. Unfortunately, this is inevitable for conventional consensus schemes which are based on local diffusion mechanisms.

B. Multiscale consensus

In this simple example we observed that the low wave-number components diminish slowly since the nodes have local information only. Once the spatial profile smooths, the nodes start to reduce the update amount. This is because each node, to its (local) knowledge, believes it has achieved a satisfactory approximation of the true average, even though it still has a large deviation from a global-scale view. This sort of problem is frequently encountered in iterative methods for solving systems of linear equations.

In order to resolve this problem, we propose a multiscale consensus scheme which transfers the information between distant nodes, so that the nodes can obtain the global information on multiple scales. In principle, the basic concept of this approach is analogous to the fundamental multigrid computation idea.

A virtual hierarchy of nodes is constructed, in each level of which the consensus scheme is executed on a different scale. We call a series of consensus sweeps along the different levels a *cycle*, a term from the multigrid computation field.

Determining the structure of the hierarchy and the cycle (the number of layers, the number of consensus sweeps in each layer, and the sequence of levels in which the consecutive consensus sweeps occur) is not trivial and could

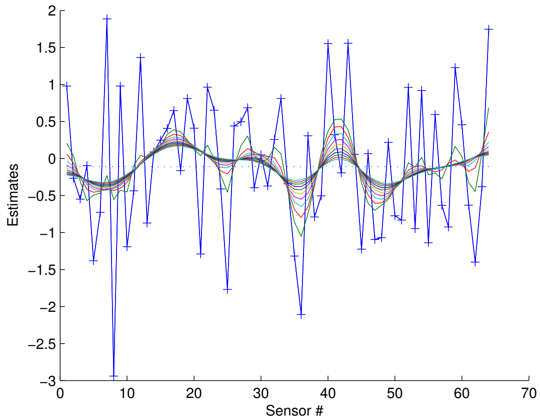

Fig. 3. Consensus history by a conventional scheme, 2 consensus sweeps (2 WU) between each curve.

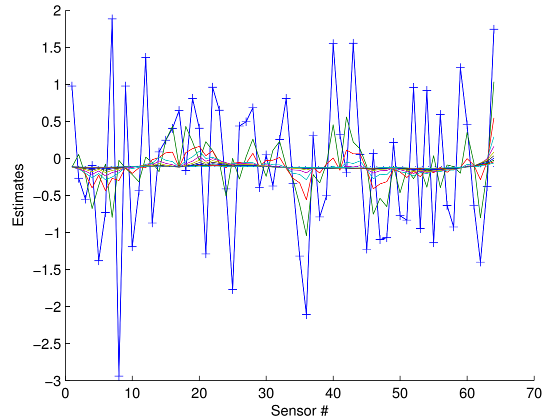

Fig. 4. Consensus history by a multiscale scheme, 1 cycle of 5 layers with $\nu = [\nu_f \; \nu_c] = [1 \; 2]$ (about 2 WU) between each curve.

be formulated as an optimization problem. However, here we present one of the simplest choices for such a scheme.

Basic multiscale consensus scheme:

1) *For the finest level (L_1),*
   a) *Execute ν_f consensus sweeps for all nodes (Increase the sweep degree of the nodes)*
2) *For the next coarsest level (L_2)*
   a) *Pick the nodes numbered $2k+1$ (The rest sleep and just pass the information)*
   b) *Execute ν_f consensus sweeps for all nodes (Increase the sweep degree of the selected nodes)*

   ⋮
3) *For the l-th level (L_l)*
   a) *Pick the nodes numbered $2^{l-1}k+1$ (The rest sleep and just pass the information)*
   b) *$\nu = \nu_f$ if $l \leq \frac{1}{2}\log_2 n$, otherwise $\nu = \nu_c$*
   c) *Execute ν consensus sweeps for all nodes (Increase the sweep degree of the selected nodes)*

   ⋮
4) *Finishing the coarsest level, go back to 1)*

Note that most of nodes undergo periodical wake-sleep transitions, which is governed by the cycle structure. When a node is sleeping, it does not use the incoming message but passes it to the other side; this results in the accelerated convergence. We assume that the cost of message passing by sleeping nodes is negligible, so instantaneous message passing occurs between distant nodes through sleeping nodes. i.e., we presume that the cycle is equivalent to a periodic change of graph topology. This may not be a reasonable assumption for some applications, a point that will be addressed in future work.

The consensus history of the proposed multiscale scheme is plotted in Fig. 4. It is obvious that the proposed scheme substantially accelerates the convergence rate, eliminating low wave-number components efficiently.

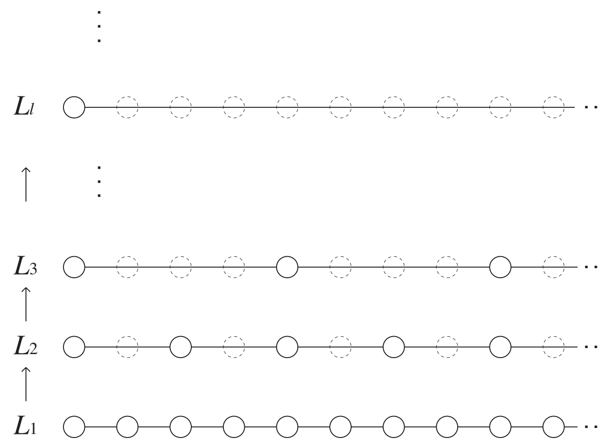

Fig. 5. Basic concept of multiscale consensus (Note that the graph reduces by half at each higher level).

Fig. 6 and Fig. 7 display the convergence history of several multiscale schemes and the conventional scheme. Various schemes with different sweep numbers are shown in Fig. 6, while the number of layers varies in Fig. 7. Different cycles lead to slight changes in performance, though any selection obviously accelerates the convergence greatly. Convergence rates are compared with respect to the total number of communications, a practically reasonable measure of the system complexity and the power consumption. The appropriate metric is defined in the following section.

V. Decentralized Kalman Filter with Multiscale Consensus

The advantage by the multiscale scheme was demonstrated in the previous example. Now the proposed algorithm is applied to the decentralized Kalman filtering problem.

A. Centralized Estimation

As a reference, the estimation by the centralized Kalman filter is shown in Fig. 8. The process noise covariance is given by $Q = (0.002C_{\max})^2 I_{128}$ and the measurement

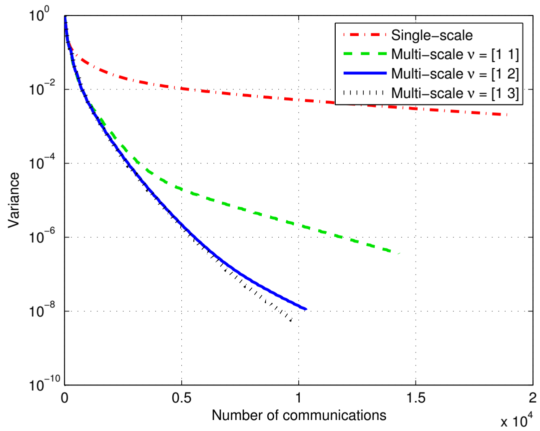

Fig. 6. Convergence with various sweep numbers (number of layer $N = 5$ fixed).

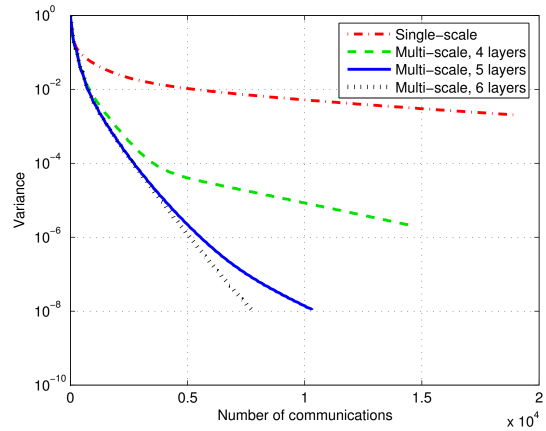

Fig. 7. Convergence with various numbers of layers (sweep number $\nu = [\nu_f \ \nu_c] = [1 \ 2]$ fixed).

noise covariance is $R = (0.1C_{\max})^2 I_{64}$. The initial guess starts from zero with variance $P_0 = (0.5C_{\max})^2 I_{128}$. The same noise and initial properties are used for the following decentralized estimation problems. For easy visual interpretation, only the estimate of the 82nd variable (contaminant concentration in node 82) is plotted.

B. Decentralized Estimation

The estimation by the decentralized Kalman filter with conventional consensus is shown in Fig. 9 (20 times consensus rate) and Fig. 10 (400 times consensus rate). The estimates of all 64 sensor units are plotted together; i.e., if the consensus reached the correct average and the decentralized estimation converged, all 64 plots should overlap the centralized estimation curve (Fig. 8). The phrase "400 times consensus rate" implies that every sensor unit communicates with neighboring units 400 times between the measurements within a single iteration of the Kalman filter. Notice the unsatisfactory convergence despite the high consensus rate and the resulting high communication requirements.

C. Decentralized Estimation with Multiscale Consensus

Before applying the proposed algorithm, we define a measure of computation and communication complexity, the WU (work unit). A WU is defined as the total number of communications required for one conventional consensus sweep. For the sensor arrangement of this particular problem, 1 WU corresponds to 126 communications.

The proposed algorithm is applied to the decentralized Kalman filter and the result is shown in Fig. 11 (10 cycles). Among the presented cycles, $N = 5$ with $\nu = [\nu_f \ \nu_c] = [1 \ 2]$ is used for this application. The communication complexity of this cycle corresponds to about 2 WU (i.e., 258 communications).

The simulation results show that the convergence is remarkably accelerated by using the multiscale consensus scheme. The estimation consensus with 10 cycles (about 20 WU) is comparable to or better than that with 400 times conventional consensus sweeps (400 WU), which requires 20 times as many communications.

VI. Conclusion and Further Research

As an example of multiscale estimation techniques, a multiscale consensus scheme is presented. The proposed algorithm is applied to example decentralized estimation problems and is demonstrated to accelerate the convergence of the estimation consensus significantly. The proposed scheme leads to an order-of-magnitude reduction of the communication cost and a more tractable design of the observer system, while still maintaining the merits of the decentralized scheme.

The proposed algorithm is effective for problems with a large number of sensor measurements with poor network connectivity. In addition, if there are far fewer state variables than measurements, it is even more effective (e.g., a set of sonobuoys deployed for tracking a submarine).

However, for problems in which the state dimension is comparable to the number of measurements or increases with network size (e.g., this building problem), additional improvements can be made by decomposing the state transition matrix as well.

The principal problem is that the dimensions of the information matrix and vector to be communicated between nodes are still large (i.e., equal to the state dimension). Moreover, the nodes must store and invert large matrices. We claim that this can be improved by another multiscale approach: *multiscale representation of the state*. In this approach, every node maintains and estimates the fine-scale knowledge about the adjacent nodes, and coarse-scale information about the distant nodes. For instance, a node on the first floor of the building has detailed knowledge of what is evolving in the first floor, while maintaining only simplified knowledge of the 4th floor and even coarser knowledge of the 20th floor.

Another acceleration is possible using *multiple time-scale separation*. Different update rates for neighboring nodes and remote nodes will help to reduce not only the computational

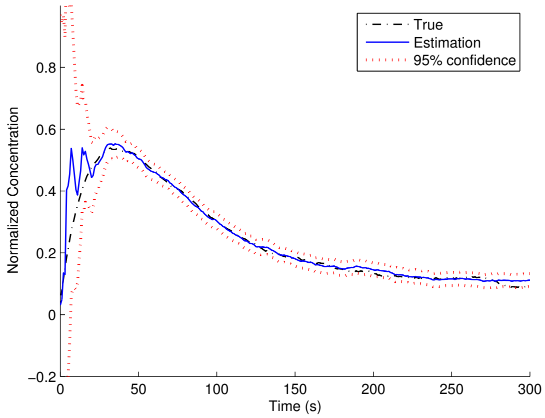

Fig. 8. Centralized estimation with the standard Kalman filter (82nd node).

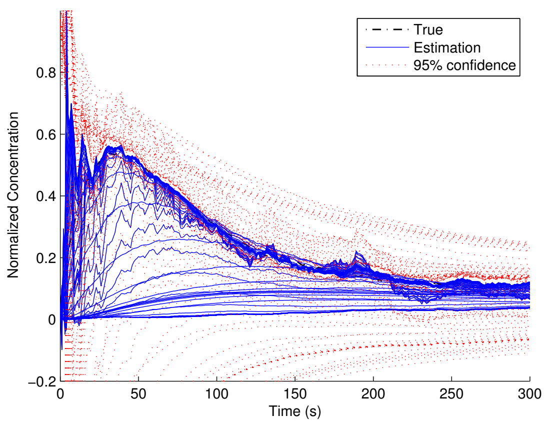

Fig. 9. Decentralized estimation with conventional consensus at 20 times consensus rate (20 WU). The system cannot maintain a coherent estimate at this rate.

load but also the communication cost. That is, a node on the first floor may not require frequent updates for what is happening on the 20th floor. Additionally, one may compute the different time-scale modes in the system. For example, eigenmodes with different time-scales that can be estimated on different time-scales are observed in Fig. 2.

Other topics of interest include: 1) eliminating the hierarchical structures and the instantaneous message passing assumption of the proposed scheme [13], and 2) applying the proposed scheme to more sophisticated distributed estimation algorithms, e.g., *Kalman-Consensus filter* [3].

REFERENCES

[1] J. L. Speyer, “Computation and transmission requirements for a decentralized linear-quadratic-gaussian control problem,” *IEEE Trans. on Automatic Control,* vol. AC-24, no. 2, pp. 266-269, Apr. 1979.
[2] B. S. Rao and H. F. Durrant-Whyte, “Fully decentralized algorithm for multisensor Kalman filtering,” in *IEE Proceedings-D,* vol. 138, no. 5, pp. 413-420, Sep. 1991.
[3] R. Olfati-Saber, “Distributed Kalman filtering for sensor networks,” in *Proc. 46th IEEE Conference on Decision and Control,* pp. 5492-5498, Dec. 2007.

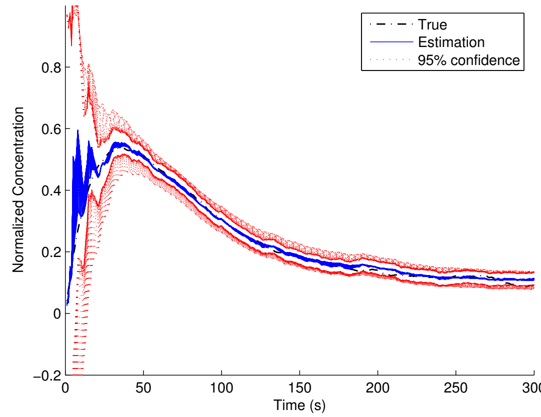

Fig. 10. Decentralized estimation with conventional consensus at 400 times consensus rate (400 WU), allowing a coherent estimate.

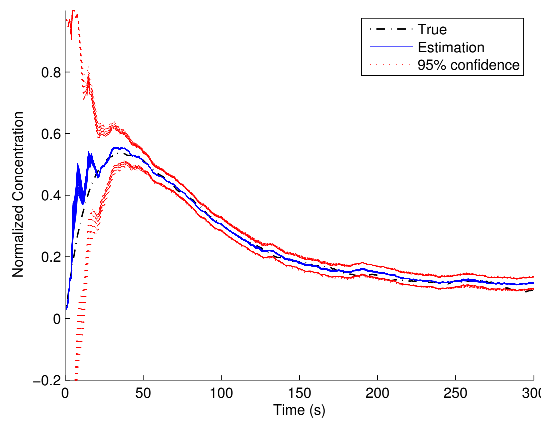

Fig. 11. Decentralized estimation with multiscale consensus using 10 cycles (about 20 WU). A coherent estimate is maintained, in contrast to the conventional method at 20 WU.

[4] W. Ren, R. W. Beard, and Ella M. Atkins, “A survey of consensus problems in multi-agent coordination,” in *Proc. 2005 American Control Conference,* pp. 1859-1864, June 2005.
[5] S. Boyd, P. Diaconis, and L. Xiao, “Fastest mixing Markov chain on a graph,” *SIAM Review,* vol. 46, no. 4., pp. 667-689, Oct. 2004.
[6] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Systems and Control Letters,* vol. 53, pp. 65-78, 2004.
[7] R. Olfati-Saber, “Ultrafast consensus in small-world networks,” in *Proc. 2005 American Control Conference,* pp. 2371-2378, June 2005.
[8] I. Yavneh, “Why multigrid methods are so efficient,” *Computing in Science & Engineering,* vol. 8, no. 6, pp. 12-22, Nov. 2006.
[9] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial.* SIAM, 2000.
[10] *CONTAM 2.4 User Guide and Program Documentation,* NIST, Oct. 2006. Available: NIST Multizone modeling website http://www.bfrl.nist.gov/IAQanalysis/index.htm
[11] P. S. Maybeck, *Stochastic Models, Estimation, and Control, Vol.1,* Academic Press, 1979.
[12] L. Xiao, S. Boyd, and S. Lall, “A space-time diffusion scheme for peer-to-peer least-squares estimation,” in *Proc. 5th International Symposium on Information Processing in Sensor Networks,* pp. 168-176, Apr. 2006.
[13] J.-H. Kim, M. West, S. Lall, E. Scholte, and A. Banaszuk, “Stochastic multiscale approaches to consensus problems,” *To appear, Conference on Decision and Control,* Dec. 2008.