Distributed and Collaborative Primary Signal Feature Estimation for Cognitive Radios under Communication Constraints

Zhenlei Shen, Yan Li, Shalinee Kishore
Electrical and Computer Engineering
Lehigh University
Bethlehem, PA 18015
{zhs2, yal2, skishore}@lehigh.edu

Aylin Yener
Electrical Engineering
Pennsylvania State University
University Park, PA 16802
yener@ee.psu.edu

***Abstract*— Collaborative algorithms are needed to improve the reliability of spectrum sensing in a network of cognitive radios (CRs). This work studies a consensus based approach to sharing spectral measurements between a multihop network of CRs. Specifically, the impact of link errors are incorporated in determining the convergence behavior of consensus based spectrum sensing. Results show that as the number of message exchanges increases, the convergence time and the deviation of the convergence value increase. Hierarchical consensus, a modification to the original consensus algorithm, is proposed to reduce the number of message exchanges while still obtaining the collaborative gains of shared spectrum sensing.**

I. Introduction

Cognitive radios [1], [2] utilize bandwidth resources based on the spectrum utilization of authorized or *primary* users of spectral bands. These agile radios can autonomously detect the radio environment and exploit instances of primary user inactivity to dynamically communicate with each other. The result is improved spectrum utilization, a promise that has fueled much recent research in this area.

A primary task of CRs is to therefore scan a wide range of frequencies and to identify those spectral bands currently unused by primary users. A great challenge in this task lies in the uncertainty of measurements made over the wireless channel. For example, the link between a CR and a primary user may be in a deep fade and can result in an erroneous (and potentially detrimental) decision that the primary user is absent. To overcome this unreliability, local CRs can collaborate in making spectrum decisions [3] [4].

The basic premise of cooperative spectrum sensing is that local CRs share their measurements (e.g., received power) with each other and use the improved reliability of their collective data to detect and estimate signal features of primary users. When a centralized entity (e.g., a fusion center) is present, it can collect data from all local CRs over an appropriate side channel and make spectrum decisions/estimates based on global spectral information; it can then broadcast these conclusions to all CRs. More practically, since CRs are self-organized, *distributed* cooperative spectrum sensing algorithms are needed to improve the reliability of the sensing process. One such cooperative scheme is based on *consensus* [5] [6], where CRs exchange measured data with their neighbors and then update their decisions according to a consensus rule till a prescribed convergence criterion is met.

Although cooperation is desirable to overcome unreliability of the wireless channel, cooperation also requires CRs to exchange spectral measurements over this same inherently unreliable media. That is, when any message (e.g., measured spectral data) is exchanged over a wireless channel, the communication link, and therefore the received message, may be in error. In case of the consensus based spectrum sensing scheme, such link errors add disturbance to the messages transmitted between CRs during data exchange. Since measured spectral data may be exchanged in error during each iteration of the consensus algorithm, convergence will be harder to achieve and convergence time may increase dramatically. A contribution of this paper is to quantify how the convergence time of the consensus based spectrum sensing scheme is impacted by such link errors over wireless channels.

Since the link errors are directly related to the number of message exchanges, the number of collaborative nodes must be limited to reduce the number of link errors. Therefore, we propose a hierarchical consensus algorithm to alleviate the effect of link errors. In this scheme, CRs are first grouped into clusters and achieve in-cluster consensus. Then cluster heads exchange data and run the consensus algorithm to reach final agreement on spectrum usage. In this way, the number of CRs involved in the consensus process is reduced and the convergence time can be reduced dramatically.

The paper is organized as follows: Consensus algorithm is presented and reviewed in Section II. The effect of link errors on the performance of the consensus algorithm is analyzed in Section III. Numerical results are presented in Section IV, where we also introduce and present results for the hierarchical consensus based scheme. Conclusions are made in Section V.

II. Collaborative Consensus Algorithm

There are N cognitive radios distributed in a geographic region $\mathcal{R}$ which contains one local spectrum band of interest. We assume the N CRs (denoted as set S) have access to a common control channel over which they can communicate limited

978-1-4244-2110-7/08/$25.00 ©2007 IEEE

information to each other. Further, all CRs are connected either directly or indirectly via multi-hops. In the following, we simplify our discussion to examine consensus based spectrum sensing for detecting the presence of a primary user in a given spectrum band. Extensions to more complex signal feature estimations of primary users over large spectral bands (i.e., multiband sensing) are straightforward. Furthermore, we assume throughout this work that the primary user status is static for the time period under consideration.

CRs make decisions regarding the presence/absence of the primary user based on some statistics y (e.g., signal strength). Since individual sensing is unreliable due to the uncertainty in the wireless channel, collaboration among CRs is desired to improve the sensing process. The consensus algorithm [5] [6] can be used to achieve agreement among multiple agents in a distributed manner; the result is an agreement that reduces uncertainties of individual observations. We import the consensus approach here to help CRs cooperate with each other and achieve more reliable spectrum sensing. In the following, we give a brief review of the general consensus approach and its known convergence properties (in the absence of link errors).

The consensus algorithm has the following steps:

1) Initialization: CRs collect signal strength $\mathbf{y}(0)$;

2) Iteration: CRs exchange message with their neighbors. The data at each individual CR is then updated according to the following rule

$$\mathbf{y}(t) = \mathbf{A}\mathbf{y}(t-1) = \mathbf{A}^t\mathbf{y}(0) \tag{1}$$

where $\mathbf{y}(t)$ is the message vector at iteration t; $\mathbf{A}$ is the updating matrix which can be derived based on a desired convergence property. For example, $\mathbf{A}$ must be a doubly stochastic matrix to achieve *average consensus* [5]. Average consensus ensures that upon convergence each node's agreed message is the average of the initial measurements made by each individual node. In the following, we study consensus assuming this average consensus rule is applied at each CR.

3) Convergence: Step 2 is repeated until $\mathbf{y}(t)$ meets a prescribed convergence criterion. In the following description, we detail an example of a practical convergence requirement.

To review the convergence properties of the algorithm given above (and assuming no link errors), we first decompose the matrix $\mathbf{A}$ using its spectral form [7] as

$$\mathbf{A} = \sum_{i=1}^{N} \lambda_i \mathbf{u}_i \mathbf{u}_i' \tag{2}$$

where $\mathbf{u}_i$ is unit length column eigenvector of matrix $\mathbf{A}$ corresponding to eigenvalue λ_i, which satisfies

$$\mathbf{A}\mathbf{u}_i = \lambda_i \mathbf{u}_i, \quad ||\mathbf{u}_i||^2 = \mathbf{u}_i'\mathbf{u}_i = 1. \tag{3}$$

Since the adjacency matrix $\mathbf{A}$ is doubly stochastic (i.e., assuming the average consensus rule is applied), it has a unique largest eigenvalue of value 1 and the corresponding eigenvector is $\mathbf{u}_1 = \mathbf{1}/\sqrt{N}$ [5]. The rest of the eigenvalues $\lambda_i, i = 2, \ldots, N$ are strictly less than 1. Therefore, the consensus algorithm converges to

$$y^* = \frac{1}{N}\sum_{i=1}^{N} y_i(0), \tag{4}$$

that is, average consensus is achieved among the initial observations and the variance of this agreed observation is

$$\mathrm{Var}[y^*] = \frac{1}{N}\mathrm{Var}[y]. \tag{5}$$

Above equation shows that the uncertainty of the final decision is reduced by a factor of N compared to the individual measurement, i.e., the sensing reliability is improved. This reduction by the factor of N is the collaboration gain from the consensus algorithm.

For practical implementation, the consensus algorithm stops when the disagreement $||\mathbf{e}(t)||$ between the message vector and the convergence value is below some level ϵ. The convergence time T_c can be derived in the following way,

$$\begin{aligned} \mathbf{e}(t) &= \mathbf{y}(t) - y^*\mathbf{1} \\ &= \sum_{i=2}^{N} \lambda_i^t \mathbf{u}_i \mathbf{u}_i' \mathbf{e}(0) \\ &\approx \lambda_2^t \mathbf{u}_2 \mathbf{u}_2' \mathbf{n} \end{aligned}$$

where the approximation step comes from $\lambda_2^t \gg \lambda_j^t, j = 3, ..., N$ and $\mathbf{n}$ is the initial disturbance vector to the measurements made over the wireless channel, i.e., the effect of unreliable spectral measurements.

Assuming that item u_{2i} are i.i.d., we obtain $\mathbb{E}[u_{2i}^2] = 1/N$ from equation (3). Then, the expectation of $||\mathbf{e}(t)||^2$ can be computed as

$$\begin{aligned} \mathbb{E}[||\mathbf{e}(t)||^2] &= \lambda_2^{2t} \sum_{i=1}^{N}\sum_{j=1}^{N} \mathbb{E}[u_{2i}^2]\mathbb{E}[u_{2j}^2]\mathbb{E}[n^2] \\ &\approx \lambda_2^{2t}\sigma^2 \end{aligned} \tag{6}$$

where σ^2 is the variance of the initial measurement.

From the stopping criterion $\mathbb{E}[||\mathbf{e}(t)||] \leq \epsilon$, the convergence time T_c of the consensus algorithm can be calculated as

$$T_c = \frac{\log(\sigma/\epsilon)}{\log(1/\lambda_2)}. \tag{7}$$

From equation (6) and (7), we see that the nature of the consensus algorithm is to drive the disagreement among CRs to zero through consensus updates.

III. Consensus Algorithm with Link Errors

In the discussion above, we assume the consensus algorithm operates on data exchanges over error free communication channels. However, in practice, the wireless channel experiences fading and introduces errors to exchanged messages. We investigate here the impact of such link errors on the convergence rate of the average consensus approach.

Link Error Model: As the reliability of the wireless channel is unpredictable, not every link introduces error. For this reason, we assume link errors occur with probability p (i.e., messages are exchanged error free with probability $1-p$); and when there are errors, we assume they are Gaussian in nature. The Gaussian error model can be justified by the different protection levels applied to message, e.g., the most significant bits of a message have much lower error probability than the least significant bits. In such a way, when an error occurs, a message is more likely to shift to its closer neighboring messages than those faraway messages (in terms of disturbance). To ease analysis, we model such disturbance as Gaussian random variable.

Mathematically, we assume the link error, denoted as Z, has the following distribution

$$Z = \begin{cases} 0, & \text{with prob } 1-p \\ \mathcal{N}(0,\sigma_l^2), & \text{with prob } p \end{cases}, \tag{8}$$

where σ_l^2 is the variance of link errors. Therefore, at time t, the message after exchange can be written as

$$\mathbf{y}'(t) = \mathbf{y}(t-1) + \mathbf{z}(t) \tag{9}$$

where $\mathbf{y}(t-1)$ is the message sent out by CR i at time t (i.e., message after update at time $t-1$); $\mathbf{y}'(t)$ is the message[1] received by its neighboring CRs at time t; $\mathbf{z}(t)$ is the error vector such that each element is independent and distributed as Z.

With link errors, the message vector at time t after consensus update can be obtained by substituting equation (9) into equation (1). The updating rule for consensus algorithm with link errors can be written as

$$\begin{aligned} \mathbf{y}(t) &= \mathbf{A}^t\mathbf{y}(0) + \sum_{k=1}^{t} \mathbf{A}^{t+1-k}\mathbf{z}(k) \\ &= \mathbf{r}(t) + \mathbf{w}(t) \end{aligned} \tag{10}$$

where $\mathbf{r}(t)$ corresponds to the conventional consensus iteration without link errors; and $\mathbf{w}(t)$ corresponds to the accumulated link errors up to time t.

For $\mathbf{y}(t)$ to converge, both $\mathbf{r}(t)$ and $\mathbf{w}(t)$ must converge. The first item is the conventional consensus algorithm and it is guaranteed to converge according to the analysis of previous section. Thus, the convergence property is largely determined by the accumulated link errors $\mathbf{w}(t)$. The following questions must be answered:

- Will $\mathbf{y}(t)$ converge?
- How do we define a practical convergence criteria when link errors may be present?
- What is the convergence rate?

Unlike the error-free consensus algorithm in which the disagreement between nodes diminishes to zero as the iteration process evolves, the link errors (if and when they occur) will constantly add disturbance during message exchange and as a result the overall disagreement will not diminish. That is, $\mathbf{y}(t)$ does not converge from equation (10) as t goes to ∞. For this reason, we must define practical convergence carefully.

[1]Though in reality, it is likely that nodes experience different error, we assume that the receiving nodes experience the same error. This simple model will greatly ease subsequent analysis and provides valuable insight on how link errors affect the consensus algorithm.

First, we define a random vector $\mathbf{v}(t)$ which represents the disagreement between message $\mathbf{y}(t)$ and its average, e.g.,

$$\mathbf{v}(t) = \mathbf{y}(t) - y^*\mathbf{1} \tag{11}$$

where y^* is the average value of message vector at time t

$$y^* = r^* + w^* = \frac{1}{N}\mathbf{1}'\mathbf{y}(0) + \sum_{k=1}^{t}\left(\frac{1}{N}\mathbf{1}'\mathbf{z}(k)\right)$$

where r^* and w^* represent the average of the original observations and the accumulated link errors up to time t.

Thus, $\mathbf{v}(t)$ can be written as

$$\begin{aligned} \mathbf{v}(t) &= \mathbf{r}(t) - r^*\mathbf{1} + \mathbf{w}(t) - w^*\mathbf{1} \\ &\approx \mathbf{w}(t) - w^*\mathbf{1} \end{aligned} \tag{12}$$

where the approximation comes from the fact that the convergence time with link errors is longer than the conventional error-free consensus algorithm. For this reason, we assume the disagreement $\mathbf{r}(t) - r^*\mathbf{1}$ is negligible compared to the item $\mathbf{w}(t) - w^*\mathbf{1}$.

Therefore, the disagreement vector $\mathbf{v}(t)$ can be written as

$$\begin{aligned} \mathbf{v}(t) &= \sum_{k=1}^{t}\left(\sum_{i=2}^{N}\lambda_i^{t+1-k}\mathbf{u}_i\mathbf{u}_i'\mathbf{z}(k)\right) \\ &\approx \sum_{k=1}^{t}\lambda_2^{t+1-k}\mathbf{u_2}\mathbf{u}_2'\mathbf{z}(k). \end{aligned} \tag{13}$$

Now, we can define the convergence criterion as the *first* occurrence when $||\mathbf{v}(t)||$ is below level ϵ, i.e.,

$$T = \min\{t : ||\mathbf{v}(t)|| \le \epsilon, ||\mathbf{v}(l)|| > \epsilon, l = 1,..,t-1\}. \tag{14}$$

From the above definition, we see that T is determined by the distribution of $\mathbf{v}(t)$, which is in turn determined by the link errors.

We compute $||\mathbf{v}(t)||^2$ in the following way, and obtain

$$\begin{aligned} ||\mathbf{v}(t)||^2 &= \sum_{k=1}^{t}\lambda_2^{2(t+1-k)}\left(\sum_{i=1}^{N}u_{2i}^2\sum_{j=1}^{N}u_{2j}^2 z_{kj}^2\right) \\ &\approx \sum_{k=1}^{t}\lambda_2^{2(t+1-k)}\left(\frac{1}{N}\sum_{j=1}^{N}z_{kj}^2\right) \end{aligned} \tag{15}$$

where we approximate $u_{2i}^2 = 1/N$; and z_{kj} represents the link error j at time k, i.e., it is the j-th element of the vector $\mathbf{z}(k)$.

Let the random variable X represent the effect of link errors during any one iteration, i.e.,

$$X = \frac{1}{N}\sum_{j=1}^{N} z_j^2. \tag{16}$$

Since Z is Gaussian distributed with probability p, the characteristic function of Z^2 can be computed as

$$\varphi_{Z^2}(v) = 1 - p + \frac{p}{\sqrt{1 - 2iv\sigma_l^2}}.$$

Then, the characteristic function of X can be computed as

$$\varphi_X(v) = \left(1 - p + \frac{p}{\sqrt{1 - 2iv\sigma_l^2/N}}\right)^N. \quad (17)$$

From the characteristic function, we can obtain the distribution $f_X(x)$ of X with the following computation

$$f_X(v) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-ivx}\varphi_X(v)dv. \quad (18)$$

Now, define random variable $W_t = ||\mathbf{v}(t)||^2$ as the accumulated link error up to time t, which can be written as

$$W_t = \sum_{k=1}^{t} \lambda_2^{2(t+1-k)} X_k. \quad (19)$$

We see that the accumulated link error is an exponential sum of link errors during each iteration. It puts more weight on recent errors and gradually forgets old errors. The distribution of W_t can be derived from the distribution of X through the characteristic function method, i.e.,

$$\varphi_{W_t}(v) = \prod_{k=1}^{t} \varphi_X(\lambda_2^{2(t+1-k)} v). \quad (20)$$

Therefore, we can obtain the distribution $f_{W_t}(w)$ as

$$f_{W_t}(w) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-ivw}\varphi_{W_t}(v)dv. \quad (21)$$

Once we have obtained the distribution $f_{W_t}(w)$, we can compute the distribution of convergence time T in the following way. First, notice that W_t can be written as

$$W_t = \lambda_2^2 W_{t-1} + \lambda_2^2 X_t, \quad (22)$$

thus, W_t is a Markov process and has the conditional independence property, i.e.,

$$\Pr[W_t | W_1, \ldots, W_{t-1}] = \Pr[W_t | W_{t-1}]. \quad (23)$$

Based on the distribution of $W_1, W_2, \ldots$, we can write the distribution of convergence time T as

$$\begin{aligned} \Pr[T = t] &= \Pr[W_t \le \epsilon^2, W_k > \epsilon^2, k = 1, \ldots, t-1] \\ &= P_t \cdot \prod_{k=2}^{t-1}(1 - P_k) \cdot \Pr[W_1 > \epsilon^2] \quad (24) \end{aligned}$$

where P_t is the probability of convergence at time t given failure of convergence at time $t-1$, which is computed as

$$\begin{aligned} P_t &= \Pr[W_t \le \epsilon^2 | W_{t-1} > \epsilon^2] \quad (25) \\ &= \int_{w_{t-1}=\epsilon^2}^{\frac{\epsilon^2}{\lambda_2^2}} F_X\left(\frac{\epsilon^2}{\lambda_2^2} - w_{t-1}\right) f_{W_{t-1}}(w_{t-1}) dw_{t-1} \end{aligned}$$

where $F_X(x)$ is the CDF defined in equation (18).

From equation (24), we can rewrite $\Pr[T = t]$ in an iterative manner, which is

$$\Pr[T = t] = \frac{P_t}{P_{t-1}}(1 - P_{t-1})\Pr[T = t-1]. \quad (26)$$

Finally, the distribution of $\Pr[T = t]$ can be calculated iteratively with the initial distribution $\Pr[T = 1]$ as

$$\Pr[T = 1] = \Pr[W_1 \le \epsilon^2] = F_X\left(\frac{\epsilon^2}{\lambda_2^2}\right). \quad (27)$$

Though we cannot obtain a closed-form formula for the distribution of T, we have the following observations:

- The higher the link error probability p is, the more errors are introduced, thus the longer the convergence time is;
- The larger the number of cooperating CRs N is, the more link errors occur during message exchange, thus the longer the convergence time is; and
- The poorer the CRs' connectivity is, the larger λ_2 is, and the larger the variable W_t is in equation (19), thus the longer the convergence time is.

IV. Simulation Results

In this section, we give simulation results for the consensus algorithm with communication link errors. CRs are uniformly distributed in a square region of side length of 1km. The transmit powers of the CRs are set so that the average number of neighboring nodes (per CR) is half the total number of CRs in the network. We assume the primary user is located faraway from the CRs and is thus roughly the same distance from each of the CRs. We therefore assume the CRs obtain Gaussian distributed power measurements (due to slow fading) from this primary user. The deviation of signal strength measurements is $\sigma = 1$ and the deviation of link errors is $\sigma_l = 0.5$. The convergence criterion is defined in equation (14) and $\epsilon = 0.05$.

Fig. 1 plots the average convergence time for consensus algorithm with link errors as a function of link error probability. The number of cooperating CRs are 20 and 40 respectively. For relatively reliable links, i.e., for error probability below 0.1, the increase in convergence time is moderate. However as p increases, convergence is more difficult to obtain since high probability of link errors disturb the message exchange process. As a result, the convergence time increases much faster when the error probability increases.

The practical convergence criterion used here requires a relatively small number of link errors for fast convergence. As the number of cooperating nodes increase, likelihood of errors increases during each iteration. The convergence time subsequently increases as well. For example, with an error probability of 0.2, it takes almost 10 times as many iterations for a network of 40 nodes to converge as compared to a network of 20 nodes. This sharp increase in convergence time motivates us to search for modifications to the consensus scheme so that even large networks can expect reasonable convergence time.

In Fig. 2, we show how closely the consensus algorithm achieves average consensus under link errors. Specifically, we plot the standard deviation (to the average consensus value) as a function of the link error probability p. Not surprisingly, as the link error probability increases, the deviation increases. As the error probability increases, more errors disturb the data exchange process; furthermore, the convergence time increases which in turn introduces additional rounds of such disturbances. The overall effect is a larger deviation to the true average value. However, we note here that when the error probability is zero or small, the convergent value is closer to the true average when there are more nodes in the network, i.e., when there are 40 nodes versus when there are 20 nodes. This is due to the collaboration gain offered by a larger number of nodes to the initial disturbance, i.e., the unreliability in the initial data measurements. However, as the error probability increases, the number of message exchanges (and thus, the number of disturbances) increases more dramatically in a larger network and results in a larger deviation.

Based on our above observations, we propose a hierarchical consensus algorithm. The motivation is to reduce the number of nodes involved in the consensus process (thereby reducing the number of message exchanges) while still including data observed by each CR in the network. In this scheme, CRs are grouped into clusters and each cluster is assigned a cluster head. Each CR is assumed to be one hop from its cluster head and therefore directly communicates its observed data to its associated cluster head. The cluster heads then exchange data with each other using the standard consensus approach.

As opposed to the flat consensus algorithm described earlier, only a small fraction of CRs (cluster heads) participate in the iterative consensus procedure. However, the data exchanged by these cluster heads incorporates all data within its cluster, i.e., all nodes contribute to all over agreement result. Since the number of links involved in the consensus algorithm is reduced, fewer disturbances are introduced. Thus, link errors have less effect on convergence and the convergence time is reduced.

To demonstrate the gains offered by hierarchical consensus, we offer the following sample results: In a network of 40 nodes, we assume that every two nodes form a cluster. That is, there are 20 clusters in the network. In this case, for a link error of 0.2, the number of iterations needed for convergence is observed to be 95. This is roughly the convergence time we observed for flat consensus in a network of 20 CRs. Furthermore, the deviation to the true average value is reduced by a factor of 2. We also note that if more CRs are allowed in each cluster, further reduction in convergence time can be obtained. Thus, we conclude that hierarchical consensus has the potential to reduce convergence time and perhaps improve deviation dramatically. For this reason, we aim to study properties of hierarchical consensus in future work.

V. Conclusion

In this work, we apply the consensus algorithm to a network of CRs to improve the spectrum sensing process. Specifically, we consider the impact of link errors on the convergence behavior of the consensus algorithm. The convergence time and error in convergence (to the true average value) increases as the number of message exchanges increases. As a result, networks with larger number of nodes or higher link error probabilities have larger convergence time and more deviation. We propose hierarchical consensus as an approach to alleviate these effects of link errors.

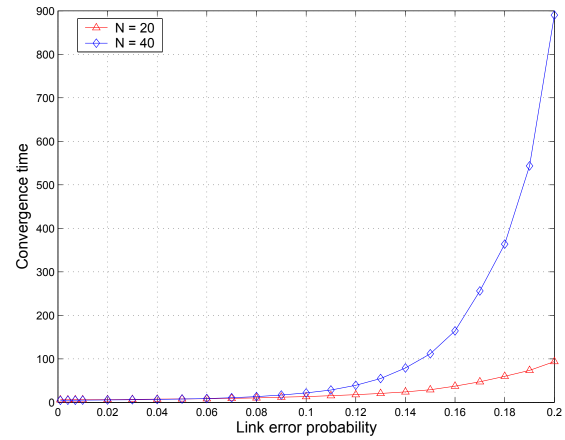

Fig. 1. Average convergence time of consensus algorithm with link errors as a function of probability of link errors.

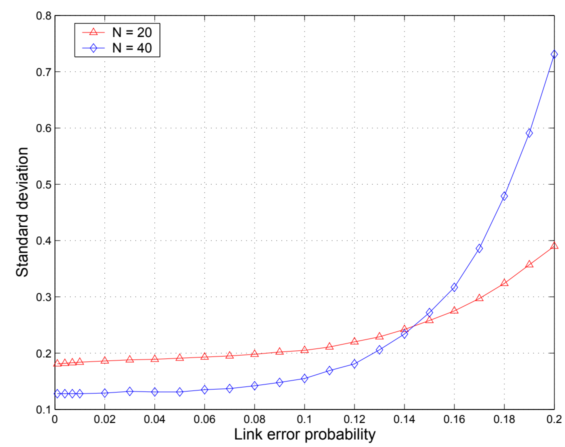

Fig. 2. Standard deviation of convergence value of consensus algorithm with link errors as a function of probability of link errors.

References

[1] J. Mitola III and G. Q. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Personal Commun. Mag.*, pp. 13–18, Aug. 1999.

[2] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Select. Areas Commun.*, pp. 201–220, Feb. 2005.

[3] G. Ganesan and Y. G. Li, "Cooperative spectrum sensing in cognitive radio networks," in *Proc. Dynamic Spectrum Access Networks, DySPAN 2005*, Baltimore, MD, Nov. 2005.

[4] A. Sahai, R. Tandra, S. M. Mishra, and N. Hoven, "Fundamental design tradeoffs in cognitive radio systems," in *Proc. of ACM TAPAS*, Aug. 2006.

[5] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. 4th Intl. Conf. on Information Processing in Sensor Networks (IPSN)*, Los Angeles, CA, Apr. 2005.

[6] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," in *Proc. IEEE Conf. on Decision and Control*, San Diego, CA, Dec. 2006.

[7] C. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics, 2001.