Binary Consensus with Soft Information Processing in Cooperative Networks

Yongxiang Ruan and Yasamin Mostofi
Department of Electrical and Computer Engineering
University of New Mexico Albuquerque, New Mexico 87113, USA
Email: {edwinruan,ymostofi}@ece.unm.edu

***Abstract*— In this paper we consider reaching binary consensus over a network with AWGN channels. We consider the case where knowledge of the corresponding link qualities is available at every receiving node. We propose novel soft information processing approaches to improve the performance in the presence of noisy links. We characterize the performance and derive an expression for the second largest eigenvalue. We show that soft information processing can improve the performance drastically. We furthermore show that, by statistically learning the voting patterns, we can solve the undesirable asymptotic behavior of binary consensus.**

I. Introduction

Cooperative decision-making and control has received considerable attention in recent years. Such problems arise in many different areas such as environmental monitoring, surveillance and security, smart homes and factories, target tracking and military systems. Consider a scenario where a network of agents wants to perform a task jointly. Each agent has limited sensing capabilities and has to rely on the group for improving its estimation/detection quality. Consensus problems arise when the agents need to reach an agreement on the value of a parameter and can be categorized into two groups: *Estimation Consensus* and *Detection Consensus*.

Estimation consensus refers to the problems where the parameter of interest can take values over an infinite set or an unknown finite set. These problems received considerable attention over the past few years [1]-[9]. *Detection Consensus*, on the other hand, refers to the problems in which the parameter of interest takes values from a finite known set. Then the update protocol that each agent will utilize becomes non-linear. We referred to a subset of detection consensus problems where the network is trying to reach an agreement over a parameter that can only have two values as *binary consensus* [10]. For instance, networked detection of fire falls into this category. While estimation consensus problems have received considerable attention, detection consensus problems have mainly remained unexplored. [11] considered and characterized phase transition of a binary consensus problem in the presence of a uniformly distributed communication noise. Since the probability density function of this noise is bounded, there exists a transition point beyond which consensus will be reached in this case [11]. In most consensus applications, the agents will communicate their status wirelessly. Therefore, the received data will be corrupted by the receiver noise, which is best modeled as an additive Gaussian noise [12]. It then becomes important to analyze the performance of consensus problems in the presence of Gaussian communication noise. In [10], we considered a networked binary consensus problem with Gaussian communication noise. Since the noise is not bounded, there is no transition point beyond which consensus is guaranteed. Instead, we utilized a probabilistic approach to characterize and understand the behavior of the network. We showed that the steady-state behavior of such systems is undesirable, independent of the amount of communication noise variance, as the network loses the memory of the initial state. We then derived an expression for the second largest eigenvalue of the system to characterize the dynamics of the network. We showed that as communication noise increases, the performance can degrade drastically. It should be noted that the binary consensus problem considered in this paper (as well as that of [10] and [11]) is fundamentally different from the problem of belief propagation [13] since we are only interested in sending one bit of information. This results in a fundamentally different problem formulation and poses more challenges. However, it is an important problem since sending as few bits as possible is crucial in several low-power sensor network applications.

To improve the performance and robustness of binary consensus, in this paper we propose *soft information processing* approaches for reaching binary consensus over AWGN channels. Techniques based on soft information processing have been utilized in coding theory and equalization in order to prevent error propagation in the presence of channel noise. In this paper we show how to build decision making functions that utilize soft approaches. We show that our proposed approach can improve consensus performance drastically. We characterize this mathematically by deriving a close approximation for the second largest eigenvalue of the underlying transition probability matrix. We furthermore show the impact of statistical learning on preventing the undesirable asymptotic behavior.

II. System Model

Consider M agents that want to reach consensus on the occurrence of an event. Each agent makes a decision on the occurrence of the event based on its one-time local sensor

This work was supported in part by the Defense Threat Reduction Agency through grant HDTRA1-07-1-0036.

measurement. Let $b_j(0) \in \{0, 1\}$ denote the initial decision of the j^{th} node, based on its local measurement, where $b_j = 1$ indicates that the j^{th} agent votes that the event occurred whereas $b_j = 0$ denotes otherwise. Then the goal is that all the agents reach a decision that is equal to the majority of the initial decisions. For instance, in a cooperative fire detection scenario, each node has an initial opinion as to if there is a fire or not. However, as a network they may act only based on the majority vote. Therefore, it is desirable that every node reaches the majority of the initial votes without a group leader. As it may happen in realistic scenarios, the nodes may not have any information on the sensing quality of themselves or others. Therefore, in this paper, the main goal is that each node reaches the majority of the initial votes. Considering sensing quality of the nodes is among possible extensions of this work.

In order to reach consensus, the agents will communicate their decisions over AWGN (Additive White Gaussian Noise) channels. Each agent sends its binary vote (only one bit of information) to the rest of the group and revises its vote based on the received information. This process will go on for a while. We say that *accurate* consensus is achieved if each agent reaches the majority of the votes, as was defined in [10]. For instance, if 4 out of 7 nodes are voting 1 initially, it is desirable that through communication they convince the rest of the group to vote one. If all the nodes vote zero in this case, accurate consensus is not achieved.

Each transmission is corrupted by receiver thermal noise as it passes through an AWGN channel. When a node receives the decisions of other nodes, the receptions can happen in different frequencies or time slots [14]. Then each reception will experience a different (uncorrelated) sample of the receiver thermal noise.[1] Let $b_{j,i}(k)$ represent the reception of the i^{th} node from the transmission of the j^{th} one at k^{th} time step. We will have $b_{j,i}(k) = b_j(k) + n_{j,i}(k)$ for $1 \leq i, j \leq M$, where $n_{j,i}(k)$ is the noise at k^{th} time step in the transmission of the information from the j^{th} node to the i^{th} one. This noise is zero-mean Gaussian [14] with the variance of σ_n^2. In this paper we assume that the graph is time-invariant and fully connected to focus on designing intelligent cooperative information processing techniques. Furthermore, we take all the noises of the links to be identically distributed. We consider the impact of relaxing these assumptions in [15]. It should, however, be noted that due to the presence of noisy channels, no node knows all the information even for a fully connected graph. This poses interesting challenges in reaching consensus, which this paper addresses.

Upon receiving the decisions of other nodes, the i^{th} node updates its vote. We will have,

$$b_i(k+1) = F\left(b_{1,i}(k), b_{2,i}(k), \ldots, b_{M,i}(k)\right), \tag{1}$$

where $b_{i,i} = b_i$ and $F(.)$ represents a decision-making function.

[1] If the transmissions are using the same frequency and time slots, they should be differentiated in codes. In this case, a node will experience different interference terms in the reception of the decisions of others.

In [10], binary consensus over noisy links was considered when no knowledge of link qualities was available. Then the following decision-making function was utilized:

$$b_i(k+1) = Dec\left(\frac{1}{M}\sum_{j=1}^{M} b_{j,i}(k)\right), \tag{2}$$

where $Dec(x) = \begin{cases} 1 & x \geq 0.5 \\ 0 & x < 0.5 \end{cases}$. This function finds the majority of the votes and would have worked perfectly if links were ideal. However, as link qualities degrade, its performance gets worse. Fig. 1 shows the performance of this decision-making function, in terms of probability of accurate consensus, for different link qualities. It can be seen that as noise level increases, the network performance degrades considerably. Furthermore, the probability of reaching accurate consensus increases only up to a certain point. To explain this, the asymptotic and transient behavior of the network was then characterized mathematically and an expression for the second largest eigenvalue of its transition matrix was derived [10].

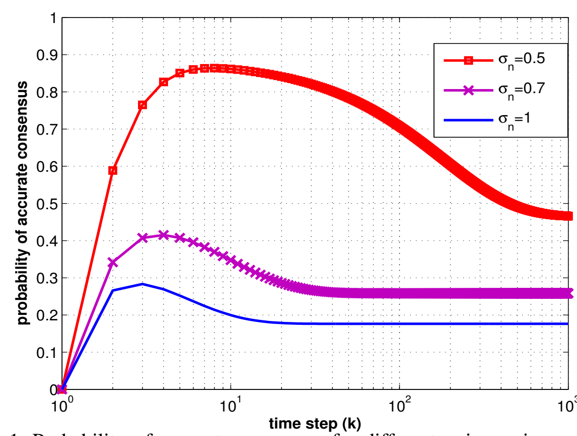

Fig. 1. Probability of accurate consensus for different noise variance σ_n^2 – performance of binary consensus when no knowledge of link qualities is utilized [10]

The main deficiency of the decision-making function of Eq. 2 is that it does not utilize potential knowledge of link qualities. In the receiver, noise variance (or equivalently Signal to Noise Ratio) can often be measured and can therefore be utilized in the consensus process [16]. It is the goal of this paper to design intelligent information processing techniques, based on using the knowledge of noise variance, to improve the performance of networked consensus.

III. Soft Information Processing

As can be seen in Eq. 2, noisy receptions are added inside the $Dec(.)$ function. If these receptions could be "cleaned" beforehand, the performance could improve. A natural choice for this is for the i^{th} node to pass each $b_{j,i}$ through a threshold function to determine if zero or one was sent as is done in the communication literature:

$$b_{j,iD}(k) = \Upsilon_{\text{hard}}\left(b_{j,i}(k)\right), \tag{3}$$

where $\Upsilon_{\text{hard}}(x) = Dec(x)$. The problem with this approach, however, is error propagation. This means that for high level of noise, the decision function, $\Upsilon_{\text{hard}}(.)$, can increase the level of error by translating the received vote to 0 or 1 erroneously. Since the error propagates in the network through repeated communication, this method does not improve the performance that much. Therefore, instead of using the "hard" decision-making function of Eq. 3, we propose to use soft information processing. Soft information processing has been used in the context of channel equalization and coding in the communication literature [17]. The main idea is that by having a softer translation of the received information, the performance can improve drastically. In this paper we apply the same concept to network consensus. The soft decision for $b_{j,i}(k)$ can be defined as follows:

$$\begin{aligned}\Upsilon_{\text{soft}}(b_{j,i}(k)) =& E[b_j(k)|b_{j,i}(k)] \qquad (4)\\ =& \text{Prob}[b_j(k)=1|b_{j,i}(k)]\\ =& \frac{\text{Prob}[b_{j,i}(k)|b_j(k)=1]\times\text{Prob}[b_j(k)=1]}{\text{Prob}[b_{j,i}(k)]},\end{aligned}$$

where $\text{Prob}[b_{j,i}(k)|b_j(k)=1] = \varphi(b_{j,i}(k)-1)$ and $\varphi(x) = \frac{1}{\sqrt{2\pi}\sigma_n}e^{-\frac{x^2}{2\sigma_n^2}}$. By substituting $\text{Prob}[b_{j,i}(k)|b_j(k)=1]$ in Eq. 4 and rearranging the terms, we will have

$$E[b_j(k)|b_{j,i}(k)] = \frac{\text{Prob}[b_j(k)=1]}{\text{Prob}[b_j(k)=1]+\text{Prob}[b_j(k)=0]e^{\frac{-2b_{j,i}(k)+1}{2\sigma_n^2}}}. \qquad (5)$$

Therefore, agent i will use the following function to update its decision:

$$b_i(k+1) = Dec\left(\frac{1}{M}\left[b_i(k)+\sum_{j\neq i}^{M}E[b_j(k)|b_{j,i}(k)]\right]\right), \qquad (6)$$

where $Dec(\cdot)$ is defined in Eq. 2. Alternatively, Eq. 6 can be thought of as a decision-making function based on the best estimate of the i^{th} node for $\sum_{j\neq i}b_j(k)$.

Fig. 2 summarizes the steps involved in soft information processing, where $\hat{p}_{j,i}(k)$ is the estimate of the i^{th} node of $\text{Prob}[b_j(k)=1]$. In this paper, we consider both cases where the receiver estimates $\text{Prob}[b_j(k)=1]$ as well the scenario where it assumes that $\text{Prob}[b_j(k)=1]$ is 0.5 (the latter is the natural assumption when this knowledge is not available in the receiver). We first start by considering the case where the receiver assumes that $\text{Prob}[b_j(k)=1]$ is 0.5. In this paper, we refer to this case as *basic soft*. Then *learning soft* will refer to the case where the i^{th} node learns $\text{Prob}[b_j(k)=1]$ statistically and incorporates it in the decision-making process.

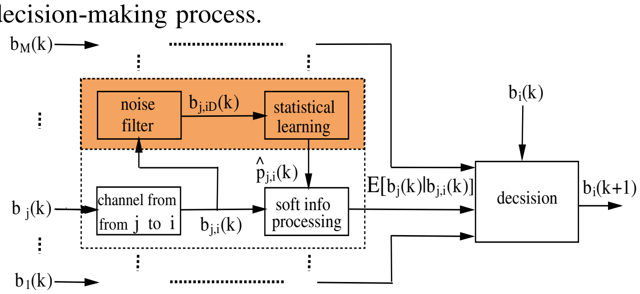

Fig. 2. An illustration of consensus with soft information processing – the shaded area is used when statistical learning is deployed (Section V) and the rest demonstrates basic soft (Section IV)

IV. Basic Soft Information Processing

In this section, we are going to analyze the properties and performance of the basic soft information processing case. As can be seen from Eq. 5 and 6, the argument inside the $Dec(.)$ function is a nonlinear function of $b_{j,i}(k)$s. Since in this paper we are interested in characterizing the dynamics of the network, we will first linearize the argument inside the $Dec(.)$ function. Our results will show that this linearization has a negligible impact on the performance.

A. Linearization

In the basic soft information processing case, we assumed that the i^{th} node does not have an estimate of $\text{Prob}[b_j(k)=1]$ for $j\neq i$. Therefore, it will assume that $\text{Prob}[b_j(k)=1] = \text{Prob}[b_j(k)=0] = 0.5$ for $j\neq i$. Then we will have the following for the basic soft case

$$b_i(k+1) = Dec\left(\frac{1}{M}\left[b_i(k)+\sum_{j=1,j\neq i}^{M}\frac{1}{1+e^{\frac{-2b_{j,i}(k)+1}{2\sigma_n^2}}}\right]\right). \qquad (7)$$

Next we will linearize $\frac{1}{1+e^{\frac{-2b_{j,i}(k)+1}{2\sigma_n^2}}}$, considering the probability distribution of $b_{j,i}(k)$.

Lemma 1: $\alpha = \frac{1}{1+4\sigma_n^2}$ and $\beta = \frac{2\sigma_n^2}{1+4\sigma_n^2}$ minimize the following cost:

$$Cost_l = \int_{-\infty}^{+\infty}\left[\frac{1}{1+e^{\frac{-2x+1}{2\sigma_n^2}}}-(\alpha x+\beta)\right]^2\varsigma(x)dx, \qquad (8)$$

where $\varsigma(x) = \frac{1}{2\sqrt{2\pi}\sigma_n}\left(e^{-\frac{x^2}{2\sigma_n^2}}+e^{-\frac{(x-1)^2}{2\sigma_n^2}}\right)$ is the pdf of random variable x.

Proof: By equating the partial derivatives to zero, we will have $E\left[\frac{x}{1+e^{\frac{-2x+1}{2\sigma_n^2}}}\right] = E[x^2]\alpha + E[x]\beta$, $E\left[\frac{1}{1+e^{\frac{-2x+1}{2\sigma_n^2}}}\right] = E[x]\alpha+\beta$. After a long derivation, it can be shown that $\alpha = \frac{1}{1+4\sigma_n^2}$ and $\beta = \frac{2\sigma_n^2}{1+4\sigma_n^2}$ are the solutions. ■

It can be easily confirmed that $b_{j,i}(k)$ has a pdf of $\varsigma(b_{j,i}(k))$, where $\varsigma(.)$ is as denoted in Lemma 1 ($n_{j,i}(k)$ is a zero mean Gaussian random variable with variance of σ_n^2). Therefore we will have the following decision-making function after linearization using Lemma 1:

$$b_i(k+1) = Dec\left(\frac{1}{M}\left[b_i(k)+\alpha\sum_{j=1,j\neq i}^{M}b_j(k)\right]+\gamma+\alpha w_i(k)\right), \qquad (9)$$

where $\alpha w_i(k)$ is a zero mean Gaussian noise with the variance of σ_s^2,

$$\alpha = \frac{1}{1+4\sigma_n^2},\ \sigma_s^2 = \frac{(M-1)\alpha^2\sigma_n^2}{M^2},\ \text{and}\quad \gamma = \frac{M-1}{2M}(1-\alpha). \qquad (10)$$

Next we will characterize the dynamics of this decision-making function.

B. State Transition Matrix

Let $S(k) = \sum_{i=1}^{M} b_i(k)$ represent the number of nodes voting 1 at time step k. $S(k)$ can be divided into two parts: $S(k) = V(k) + U(k)$, where $V(k)$ is the number of nodes that are voting one in state k (time step k) and voted one in state $k-1$. Similarly, $U(k)$ represents the number of nodes that are voting one in state k but voted zero in state $k-1$. Given $S(k) = n$, both $V(k+1)$ and $U(k+1)$ become binomially distributed:

$$\text{Prob}\left[V(k+1) = m|S(k) = n\right] = \binom{n}{m} \kappa_{n,1}^{m}(1-\kappa_{n,1})^{n-m},$$
$$\text{Prob}\left[U(k+1) = m|S(k) = n\right] = \binom{M-n}{m} \kappa_{n,0}^{m}(1-\kappa_{n,0})^{M-n-m}, \tag{11}$$

where

$$\begin{aligned}\kappa_{n,1} =& \text{Prob}[b_i(k+1) = 1|b_i(k) = 1 \ \& \ S(k) = n] \\ =& Q\left(\frac{0.5 - \gamma - \frac{1+(n-1)\alpha}{M}}{\sigma_s}\right), \ \forall i, \\ \kappa_{n,0} =& \text{Prob}[b_i(k+1) = 1|b_i(k) = 0 \ \& \ S(k) = n] \\ =& Q\left(\frac{0.5 - \gamma - \frac{n\alpha}{M}}{\sigma_s}\right), \ \forall i.\end{aligned} \tag{12}$$

Then, given current state $S(k) = n$, the pdf of $S(k+1)$ is a convolution of two independent binomial distributions corresponding to $V(k+1)$ and $U(k+1)$. Let $P_{n,m}$ represent the probability of going from state n to state m. We will have,

$$\begin{aligned}P_{n,m} =& \text{Prob}[S(k+1) = m|S(k) = n] \\ =& \sum_{x=\psi_{n,m}}^{\psi'_{n,m}} f(x, n, \kappa_{n,1}) f(m-x, M-n, \kappa_{n,0}) \\ =& \sum_{x=\psi_{n,m}}^{\psi'_{n,m}} \binom{n}{x} \kappa_{n,1}^{x}(1-\kappa_{n,1})^{n-x} \\ & \times \binom{M-n}{m-x} \kappa_{n,0}^{m-x}(1-\kappa_{n,0})^{M-n-m+x},\end{aligned} \tag{13}$$

where $\psi_{n,m} = \max(0, m+n-M)$, $\psi'_{n,m} = \min(n,m)$ and $f(x,n,q) = \binom{n}{x} q^x (1-q)^{n-x}$ is the pdf of a binomial distribution with q as the success probability. α, γ and σ_s are as defined in Eq. 10. Then $S(k)$, i.e. the number of nodes voting 1 at time step k, is the sufficient information to represent the state of the network. Fig. 3 shows the transition from state n to state m. Furthermore,

$$E[S(k+1)|S(k) = n] = n\kappa_{n,1} + (M-n)\kappa_{n,0}. \tag{14}$$

Let $\Pi_i(k)$ represent the probability that $i \in \Omega_M$ of the agents are voting 1 at k^{th} time step, where $\Omega_M = \{0, 1, 2, \ldots, M\}$: $\Pi_i(k) = \text{Prob}\left[S(k) = i\right], \quad i \in \Omega_M$. Then

$$\Pi(k+1) = P^T \Pi(k), \tag{15}$$

where $\Pi(k) = \left[\begin{array}{cccc} \Pi_0(k) & \Pi_1(k) & \ldots & \Pi_M(k) \end{array}\right]^T$ and $P = [P_{n,m}]$.

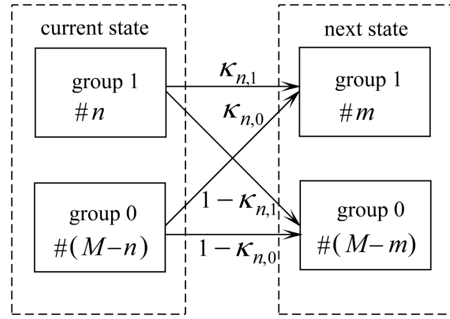

Fig. 3. The state transition from state n to state m – group 1 refers to the set of nodes that are voting one

1) Properties of the transition matrix: Let $\lambda_0, \lambda_1, \cdots, \lambda_M$ represent the eigenvalues of matrix P in a decreasing order: $|\lambda_0| \geq |\lambda_1| \geq \cdots \geq |\lambda_M|$. Matrix P will have the following properties for $\sigma_n \neq 0$:

a) P is a row stochastic matrix, i.e. $\forall n$, $\sum_{m=0}^{M} P_{n,m} = 1$.
b) From the Gersgorin disk theorem [18], the eigenvalues of P are located in the following area: $\bigcup_{i=0}^{M}\{z \in C : |z - P_{i,i}| \leq 1 - P_{i,i}\}$. This means that $|\lambda_i| \leq 1, \ \forall i$.
c) From Perron's theorem [19], $\lambda_0 = 1$ is a simple eigenvalue of P and $[P^T]^k \to L$ as $k \to \infty$, where $L = \mu\nu^T$, $\mu = P^T\mu$, $\nu = P\nu$, and $\mu^T\nu = 1$, $\nu = [11\cdots1]^T$.

Lemma 2: $\kappa_{n,1} = 1 - \kappa_{M-n,0}$.

Proof: We have $\kappa_{n,1} = Q\left(\frac{0.5-\gamma-\frac{1+(n-1)\alpha}{M}}{\sigma_s}\right)$ and $1 - \kappa_{M-n,0} = 1 - Q\left(\frac{0.5-\gamma-\frac{(M-n)\alpha}{M}}{\sigma_s}\right) = Q\left(\frac{-0.5+\gamma+\frac{(M-n)\alpha}{M}}{\sigma_s}\right) = \kappa_{n,1}$, where the last equality is written using the relationship between γ and α of Eq. 10. ■

Lemma 3: The P matrix is a centrosymmetric matrix. i.e. the $(M-n)^{\text{th}}$ row is a reverse repeated version of the n^{th} row: $P_{M-n,M-m} = P_{n,m}$ for $0 \leq n, m \leq M$.

Proof:

$$P_{M-n,M-m} = \sum_{x=\psi_{M-n,M-m}}^{\psi'_{M-n,M-m}} f\left(x, M-n, \kappa_{M-n,1}\right) f\left(M-m-x, n, \kappa_{M-n,0}\right). \tag{16}$$

From Lemma 2, $\kappa_{n,1} = 1 - \kappa_{M-n,0}$. Then we have,

$$P_{M-n,M-m} = \sum_{x=\psi_{M-n,M-m}}^{\psi'_{M-n,M-m}} f\left(x, M-n, 1-\kappa_{n,0}\right) f\left(M-m-x, n, 1-\kappa_{n,1}\right). \tag{17}$$

Let $r = x - M + n + m$. Then we have

$$\begin{aligned}& P_{M-n,M-m} \\ =& \sum_{r=\psi_{n,m}}^{\psi'_{n,m}} f\left(m-r, M-n, 1-\kappa_{n,0}\right) f\left(r, n, 1-\kappa_{n,1}\right) \\ =& P_{n,m}.\end{aligned} \tag{18}$$

■

2) Case of $\sigma_n = 0$: For ideal links, we have $b_j(k) = b_{j,i}(k)$ and $b_i(k+1) = Dec(\frac{1}{M}\sum_{j=1}^{M} b_j(k))$, which means that the network will reach accurate consensus in one step. In this case, $\lambda_0 = \lambda_1 = 1$ and $\lambda_i = 0$ for $2 \leq i \leq M$ (see [10] for more details).

3) Case of $\sigma_n \to \infty$: In this case, we have $\lim_{\sigma_n \to \infty} \alpha = 0$ and $\lim_{\sigma_n \to \infty} \gamma = \frac{M-1}{2M}$. Therefore, it can be confirmed from Eq. 9 that each node will keep its own vote and does not take any communication into account, as should be the case.

C. Second Largest Eigenvalue

It can be seen from the properties of the transition matrix that the asymptotic value of $\Pi(k)$ is independent of the initial state: $\lim_{k\to\infty} \Pi(k) = \frac{\mu}{\text{sum}(\mu)}$, where μ is the right eigenvector of P^T. This is what is referred to as "memoryless consensus" in [10], i.e. the network loses the memory of the initial state asymptotically. While this is not desirable, the network can still be in consensus with high probability for a long period of time. To characterize this, the second largest eigenvalue of P plays a key role as discussed in [10]. In this part, we derive an expression for λ_1, the second largest eigenvalue of the transition matrix. As we saw in the previous section, $|\lambda_1| \leq 1$ with $\lambda_1 = 1$ only if $\sigma_n = 0$. Therefore, intuitively, the larger the second eigenvalue is, the better the performance should be. We will show that our proposed soft decision-making approach will result in the network staying in accurate consensus with higher probability for a longer period of time. We then show, in the next section, that our proposed learning soft strategy will solve the "memoryless consensus" issue.

Lemma 4: Let $\xi = \lceil \frac{M}{2} \rceil$ and $\eta_n = E[S(k+1)|S(k) = n]/M$, where $E[S(k+1)|S(k) = n]$ is defined in Eq. 14. Let P represent the transition probability matrix generated using $\kappa_{n,1}$ and $\kappa_{n,0}$ of Eq. 12. We will have

$$\sum_{m=0}^{\xi-1} (\frac{M}{2} - m)(P_{n,m} - P_{M-n,m}) = \frac{M}{2}(1 - 2\eta_n). \quad (19)$$

Proof: We have

$$\begin{aligned}
&\sum_{m=0}^{\xi-1} (\frac{M}{2} - m)(P_{n,m} - P_{M-n,m}) = \\
&\sum_{m=0}^{\xi-1} (\frac{M}{2} - m) \sum_{x=\psi_{n,m}}^{\psi'_{n,m}} f(x, n, \kappa_{n,1}) f(m - x, M - n, \kappa_{n,0}) \\
&- \sum_{m=0}^{\xi-1} (\frac{M}{2} - m) \sum_{y=\psi_{M-n,m}}^{\psi'_{M-n,m}} f(y, M - n, \kappa_{M-n,1}) \\
&\times f(m - y, n, \kappa_{M-n,0}),
\end{aligned} \quad (20)$$

where $\psi_{n,m}$, $\psi'_{n,m}$ are defined in Eq. 13. By using the fact that $\kappa_{M-n,0} = 1 - \kappa_{n,1}$ from Lemma 2, we will have

$$\begin{aligned}
&- \sum_{m=0}^{\xi-1} (\frac{M}{2} - m) \\
&\times \sum_{y=\psi_{M-n,m}}^{\psi'_{M-n,m}} f(y, M - n, \kappa_{M-n,1}) f(m - y, n, \kappa_{M-n,0}) \\
&= \sum_{m=\xi}^{M} (\frac{M}{2} - m) \sum_{x=\psi_{n,m}}^{\psi'_{n,m}} f(x, n, \kappa_{n,1}) f(m - x, M - n, \kappa_{n,0}).
\end{aligned} \quad (21)$$

Therefore,

$$\begin{aligned}
&\sum_{m=0}^{\xi-1} (\frac{M}{2} - m)(P_{n,m} - P_{M-n,m}) = \sum_{m=0}^{M} (\frac{M}{2} - m) P_{n,m} \\
&= \sum_{m=0}^{M} (\frac{M}{2} - m) \text{Prob}[S(k+1) = m|S(k) = n] \\
&= \frac{M}{2} - E[S(k+1)|S(k) = n] = \frac{M}{2}(1 - 2\eta_n).
\end{aligned} \quad (22)$$

■

Due to the complicated structure of the transition matrix of Eq. 13, finding an exact expression for the second largest eigenvalue is not feasible. Instead, we derive a tight approximation for the second largest eigenvalue by linearizing the Q function.

Assumption 1: For small enough x, we can linearize the $Q(.)$ function around the origin via the first order approximation of the Taylor series: $Q(x) \approx Q(0) - \frac{x}{\sqrt{2\pi}}$. By applying this approximation to $\kappa_{n,1}$ and $\kappa_{n,0}$ and using the definition of η_n, we have

$$\begin{aligned}
\eta_n &\approx 0.5 - (1 - \frac{2n}{M}) \frac{(M-1)\alpha + 1}{2\sqrt{2\pi} M \sigma_s}, \\
\eta_0 &\approx 0.5 - \frac{(M-1)\alpha + 1}{2\sqrt{2\pi} M \sigma_s}.
\end{aligned} \quad (23)$$

Therefore, for the purpose of finding the second largest eigenvalue, we can approximate η_n as follows,

$$\eta_{n,\text{approx}} = \frac{n}{M} + (1 - \frac{2n}{M})\eta_0. \quad (24)$$

It can be confirmed from Eq. 12 that this approximation becomes more accurate for either large communication noise variances or small ones. Next, we use the properties of the centrosymmetric matrices [20], [21] to derive a tight approximation for the second largest eigenvalue.

Lemma 5: (a) Let G represent an $n \times n$ centrosymmetric matrix, where $n = 2n'$. G can be partitioned as $\begin{bmatrix} A & BJ \\ JB & JAJ \end{bmatrix}$, where A, B are $n' \times n'$ matrices and J represents an $n' \times n'$ backward identity matrix: $J_{i,j} = \delta_{i,n'-j}$ for $0 \leq i, j \leq n' - 1$ ($\delta_{i,j}$ is the Kronecker delta). Furthermore, G is similar to

$$K_e^{-1} G K_e = \begin{bmatrix} A + B & 0 \\ 0 & A - B \end{bmatrix}, \quad (25)$$

where $K_e = \begin{bmatrix} I & -I \\ J & J \end{bmatrix}$ is an $n \times n$ matrix and I is an $n' \times n'$ identity matrix.

(b) Let G represent an $n \times n$ centrosymmetric matrix, where $n = 2n' + 1$. G can be partitioned as $\begin{bmatrix} A & a & BJ \\ b & c & bJ \\ JB & Ja & JAJ \end{bmatrix}$, where A, B are $n' \times n'$ matrices, a is an $n' \times 1$ matrix, b is a $1 \times n'$ matrix and c is a scalar. Furthermore, G is similar to

$$K_o^{-1} G K_o = \begin{bmatrix} A + B & a & 0 \\ 2b & c & 0 \\ 0 & 0 & A - B \end{bmatrix}, \quad (26)$$

where $K_o = \begin{bmatrix} I & 0 & -I \\ 0 & 1 & 0 \\ J & 0 & J \end{bmatrix}$.

Proof: See [21]. ■

Since P is an $(M+1)\times(M+1)$ centrosymmetric matrix, we can apply Lemma 5. If $M+1 = 2\xi$, it can be easily seen that $A_{n,m} = P_{n,m}$ and $B_{n,m} = P_{M-n,m}$ for $0 \le n, m \le \xi - 1$ (note that we are indexing matrix elements from zero). If $M+1 = 2\xi+1$, we will have $A_{n,m} = P_{n,m}$, $B_{n,m} = P_{M-n,m}$, $a_n = P_{n,\frac{M}{2}}$, $b_m = P_{\frac{M}{2},m}$ and $c = P_{\frac{M}{2},\frac{M}{2}}$ for $0 \le n, m \le \xi - 1$.

Lemma 6: Let P_{approx} represent the transition probability matrix generated under Assumption 1. Then, $1 - 2\eta_0$ is an eigenvalue of $(A-B)_{\text{approx}}$, with the corresponding eigenvector as $\begin{bmatrix} \frac{M}{2} & \frac{M}{2}-1 & \cdots & \frac{M}{2}-(\xi-1) \end{bmatrix}^T$, where $\xi = \lceil \frac{M}{2} \rceil$.

Proof: We have $A - B = \begin{bmatrix} P_{n,m} - P_{M-n,m} \end{bmatrix}$ for $0 \le n, m \le \xi - 1$. Let $\chi = (A - B)_{\text{approx}} \begin{bmatrix} \frac{M}{2} & \frac{M}{2}-1 & \cdots & \frac{M}{2}-(\xi-1) \end{bmatrix}^T$. Then the n^{th} element of vector χ will be as follows for $0 \le n \le \xi-1$: $\chi_n = \sum_{m=0}^{\xi-1}(\frac{M}{2} - m)(P_{n,m,\text{approx}} - P_{M-n,m,\text{approx}})$. By using Lemma 4 and Eq. 24, we will have: $\chi_n = \frac{M}{2}(1 - 2\eta_{n,\text{approx}}) = \frac{M}{2}(1-\frac{2n}{M})(1-2\eta_0) = (\frac{M}{2}-n)(1-2\eta_0)$, $0 \le n \le \xi - 1$. Then, $\begin{bmatrix} \frac{M}{2} & \frac{M}{2}-1 & \cdots & \frac{M}{2}-(\xi-1) \end{bmatrix}^T$ is a right eigenvector of $(A-B)_{\text{approx}}$ and $1 - 2\eta_0$ is the corresponding eigenvalue. ■

Theorem 1: Let P_{approx} represent the transition probability matrix generated under Assumption 1. Let $\lambda_{1,\text{approx}}$ represent the second largest eigenvalue of P_{approx}. Then we have, $\lambda_{1,\text{approx}} = 1 - 2\eta_0 = 1 - 2Q(\frac{4\sigma_n^2+M}{2\sigma_n\sqrt{M-1}})$.

Proof: From Lemma 6, we know that $1 - 2\eta_0$ is an eigenvalue of $(A-B)_{\text{approx}}$. From Lemma 5, we know that eigenvalues of $(A-B)_{\text{approx}}$ are eigenvalues of P_{approx}. Therefore, $1-2\eta_0$ is an eigenvalue of P_{approx}. Furthermore, it can be easily confirmed that as $\sigma_n \to 0$, $1-2\eta_0$ goes to one, which only happens to the second largest eigenvalue (See Section IV-B. It can be similarly confirmed for P_{approx}). Therefore $\lambda_{1,\text{approx}} = 1 - 2\eta_0 = 1 - 2Q(\frac{4\sigma_n^2+M}{2\sigma_n\sqrt{M-1}})$. ■

To see how well this approximation works, Fig. 4 shows the second largest eigenvalue and its approximation as a function of σ_n and for $M = 4$, $M = 10$ and $M = 20$. As can be seen, the approximation works well especially for smaller M, larger σ_n, or σ_n close to zero. For comparison, the second largest eigenvalue is $\lambda_{1,\text{original}} = 1 - 2Q(\frac{M}{2\sigma_n\sqrt{M-1}})$ for the case where no knowledge of channel qualities was used, as indicated by Eq. 2 [10]. It can be confirmed that $\lambda_{1,\text{original}}$ decreases drastically as σ_n increases, getting values much smaller than one. In our case, however, $\lambda_{1,\text{approx}}$ of Theorem 1 behaves differently. At lower σ_n, as σ_n increases, it will decrease. However, at higher σ_n, $\lambda_{1,\text{approx}}$ starts increasing. This is due to the fact that soft information processing weighs the received information based on link qualities. Therefore, at considerably high σ_n, the received information is almost ignored, which results in $\lambda_{1,\text{approx}}$ approaching one.

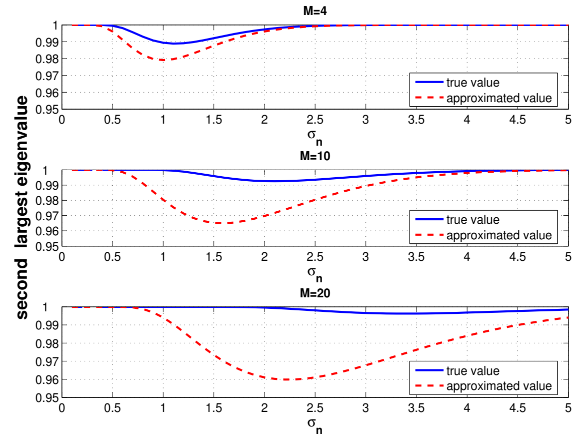

Fig. 4. Approximation of the second largest eigenvalue for basic soft case

V. Soft Information Processing with Statistical Learning

As can be seen from the true structure of the soft information processing function of Eq. 5 and 6, the i^{th} node needs to have the knowledge of $\text{Prob}[b_j(k) = 1]$ for $j \ne i$. Such information is not readily available in the receiver. However, it can be statistically estimated, as is shown in the highlighted part of Fig. 2. Node i will pass $b_{j,i}(k)$ through the hard decision function, as indicated by Eq. 3, to generate $b_{j,iD}(k)$. At the k^{th} time step, node i counts the number of times that $b_{j,iD}$ became one in a given time interval to estimate $\text{Prob}[b_j(k) = 1]$. Let $\hat{p}_{j,i}(k)$ represent i^{th} node's estimate of $\text{Prob}[b_j(k) = 1]$. Then we will have the following form of decision-making:

$$b_i(k+1) = Dec\left(\frac{1}{M} \times \left[b_i(k) + \sum_{j\ne i}^{M} \frac{\hat{p}_{j,i}(k)}{\hat{p}_{j,i}(k) + (1-\hat{p}_{j,i}(k))\, e^{\frac{-2b_{j,i}(k)+1}{2\sigma_n^2}}} \right]\right). \quad (27)$$

Similar to Section IV-A, we will get the following after linearization:

$$b_i(k+1) = Dec\left(\frac{1}{M}\left[b_i(k) + \sum_{j\ne i}^{M} \alpha_{j,i}(k) b_j(k)\right] + \gamma_i(k) + \frac{\sum_{j\ne i}\alpha_{j,i}(k) n_{j,i}(k)}{M}\right), \quad (28)$$

where $\alpha_{j,i}(k)$ and $\gamma_i(k)$ can be found similarly.

Since $\alpha_{j,i}$ are not the same for different js, as compared to the basic soft case of Eq. 9, $S(k)$ is no longer sufficient information to represent the state of the network (it can be proved mathematically). We instead have to use vector $[b_1(k)\ b_2(k) \cdots b_M(k)]$ as the state of the network, which will result in a state transition matrix of size $2^M \times 2^M$ with

$$\text{Prob}[b_i(k+1) = 1 | b_1(k), b_2(k), \cdots, b_M(k)] = Q\left(\frac{0.5 - \gamma_i(k) - \frac{1}{M}[b_i(k) + \sum_{j\ne i}^{M}\alpha_{j,i}(k) b_j(k)]}{\sigma_s(k)}\right), \quad (29)$$

where $\sigma_s^2(k) = \frac{\sum_{j \neq i} \alpha_{j,i}^2(k)\sigma_n^2}{M^2}$. Finding a closed-form expression for the 2nd largest eigenvalue of this case is considerably challenging. Instead, we characterize the performance through simulation in the next part.

VI. Performance of the Proposed Approaches

Fig. 5 shows the performance of the proposed soft information processing approaches. For comparison, the performance of the original algorithm of Eq. 2, where no knowledge of link qualities is used, is also plotted. It can be seen that the proposed strategies can improve the performance drastically. Furthermore, by comparing basic soft and basic soft linear cases, it can be seen that the linearization of Eq. 7 did not impact the performance (we found this to be the case for other σ_ns as well). As can be seen, the basic soft information-processing approach of Section IV can improve the performance drastically compared to the original case. However, its asymptotic behavior is still undesirable as the probability of accurate consensus starts to decrease after a while. It can be seen that by statistically learning the probability distribution of the votes of other nodes, learning soft can improve the performance considerably and have a desirable asymptotic behavior. It should be noted that the discontinuity of the learning soft curve (at around $k = 10$) is due to the fact that each node starts the statistical learning process after 10 iterations in order to accumulate enough data.

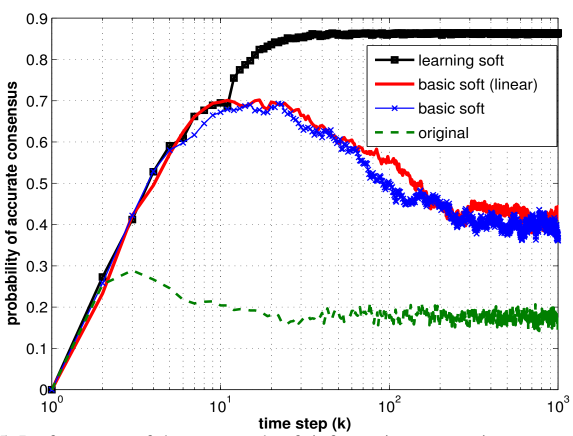

Fig. 5. Performance of the proposed soft information processing approaches for $\sigma_n = 1$

VII. Conclusion and Further Extensions

In this paper we considered binary consensus over noisy communication links. We proposed novel soft information processing approaches in order to have more robust consensus behavior in the presence of noisy links. We characterized the behavior of a network that deploys the proposed strategies and found an expression for the second largest eigenvalue of the transition matrix. We furthermore showed that by statistically learning the voting patterns of the nodes, we can have a desirable asymptotic behavior. Overall, our results show the drastic improvement gained through using soft information processing. We are currently working on extending the proposed framework to more general cases including not fully connected, time-varying graphs using some of the results that we have derived in [15] and [22].

VIII. Acknowledgment

The authors would like to thank Mr. Mehrzad Malmirchegini for useful discussions.

References

[1] W. Ren and R. W. Beard, “Consensus seeking in multi-agent systems under dynamically changing interaction topologies,” IEEE Transactions on Automatic Control, vol. 50, no. 5, pp. 655-661, 2005.

[2] R. Olfati-Saber and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” IEEE Transactions on Automatic Con- trol, vol. 49, no. 9, pp. 1520-33, 2004.

[3] Y. Hatano and M. Mesbahi, “Agreement over random networks,” IEEE Trans. on Automatic Control, vol. 50, no. 11, pp. 1867-1872, 2005.

[4] M. Mesbahi, “State-dependent graphs,” Proceedings of 42nd IEEE conference on Decision and Control, 2003.

[5] J. Lin, A. S. Morse, and B. D. O. Anderson, “The multi-agent rendezvous problem - part 1: the synchronous case,” SIAM Journal on Control and Optimization, 2006.

[6] A. Jadbabaie, J. Lin, and A. S. Morse, “Coordination of groups of mobile autonomous agents using nearest neighbor rules,” IEEE Transactions on Automatic Control, vol. 48, no. 6, 2003.

[7] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, “Approximate distributed Kalman filtering in sensor networks with quantifiable performance,” in the proceedings of 4th international conference on Information Processing in Sensor Networks (IPSN), pp. 133-139, 2005.

[8] W. Ren, R. Beard, D. Kingston, “Multi-agent Kalman consensus with relative uncertainty,” Proc. of 2005 American Control Conference, 2005.

[9] W. Ren, R. Beard and E. Atkins, “A Survey of Consensus Problems in Multi-agent Coordination,” Proc. of 2005 American Control Conference, 2005.

[10] Y. Mostofi, “Binary Consensus with Gaussian Communication Noise: A Probabilistic Approach,” Proceedings of the 46th IEEE Conference on Decision and Control (CDC), Dec. 2007

[11] J. Liu, V. Yadav, H. Sehgal, J. Olson, H. Liu, and N. Elia, “Phase Transitions on Fixed Connected Graphs and Random Graphs in the Presence of Noise,” in the proceedings of 44th IEEE Conference on Decision and Control, pp. 2996-3000, 2005.

[12] John G. Proakis, *Digital Communication*. McGraw Hill Series in Electrical and Computer Engineering, 3rd edition, 1995.

[13] R. Olfati-Saber, E. Franco, E. Frazzoli, and J. S. Shamma, “Belief Consensus and Distributed Hypothesis Testing in Sensor Networks,” Network Embedded Sensing and Control, Vol. 331, pp. 169-182, 2006.

[14] W. Jakes, Microwave Mobile Communications. IEEE Press, 1974.

[15] Y. Yuan, Y. Mostofi, “Impact of Link Qualities and Network Topology on Binary Consensus”, submitted to the American Control Conference, 2009.

[16] Theodore S. Rappaport, *Wireless Communications: Principles and Practice, Second Edition*.

[17] J. Wen and J. D. Villasenor, “Utilizing soft information in decoding of variable length codes,” Proc. of IEEE Data Compression Conference, 1999.

[18] R. Horn and C. Johnson, *Matrix analysis*. Cambridge University Press 1999.

[19] Abraham Berman and Robert J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*. Academic Press 1979.

[20] James R. Weaver, “Centrosymmetric (Cross-Symmetric) Matrices, Their Basic Properties, Eigenvalues, and Eigenvectors,” The American Mathematical Monthly, Vol. 92, No. 10. (Dec., 1985), pp. 711-717.

[21] A.R.Collar, “On Centrosysmmetric and Centroskew Matrices,” The Quarterly Journal of Mechanics and Applied Mathematics, Vol.15, No3, pp. 265-281 1962.

[22] M. Malmirchegini, Y. Ruan, Y. Mostofi, “Binary Consensus Over Fading Channels: A Best Affine Estimation Approach,” Proc. IEEE Globecom, November 2008.