Fault Diagnosis Using Consensus of Markov Chains

Dejan P. Jovanović
Department of Mathematics
The University of Queensland
Qld 4072 AUSTRALIA
Email: dejan.jovanovic@uqconnect.edu.au

Philip K. Pollett
Department of Mathematics
The University of Queensland
Qld 4072 AUSTRALIA
Email: pkp@maths.uq.edu.au

***Abstract*—A fault diagnosis procedure is proposed based on consensus in a group of local agents/experts. Local models are represented by Markov chains and modelling consensus as a mixture of these allows estimation of optimal ratings using an EM framework. To deal with the unobservable case the procedure is extended to accommodate Hidden Markov models (HMMs).**

***Index Terms*—Fault diagnosis, consensus algorithm, mixtures of Markov chains, the EM algorithm, Hidden Markov Models (HMM), multi-agent systems.**

I. Introduction

Fault diagnosis is a procedure for recognising unwanted behaviour in monitored systems. It comprises three sequential steps (known as the fundamental tasks of fault diagnosis [1]). The first, *fault detection*, is deciding whether the characteristics of the system in question are outside permissible limits. The second, *fault identification*, is determining which subsystems contain a fault of a particular type and the time when it occurred. Finally, *fault analysis* provides insight into the time-varying characteristics of the fault and the scale of disturbance that occurred.

To detect changes in relevant system characteristics it is necessary to have an adequate reference model. In the existing literature on fault diagnosis ([1], [2], [3], [4], [5]), there are three well-known approaches to system modelling. In the *model-based* approach, the relationship between system inputs and outputs is described by a parameterized mathematical model. In the *knowledge-based* approach, causal relationships between system events, observed symptoms and faults are modelled. And, in the *data-based* approach, probabilistic signal models are used. The choice depends on the problem at hand, and the task is usually not simple, particularly for distributed systems, where there are many interrelated and interconnected parts.

The theory of automatic control suggest simplifying the model by decomposing the system into a number of subsystems. Whether the decomposition is deterministic [6] or probabilistic [7], it is assumed that each subsystem is equipped with its own local observations and a set of local models that describe different working conditions. Regardless of which technique used, the assumptions that local measurements in each subsystem are taken independently and the subsystems evolve independently, give rise to the following two problems that need addressed: cooperation between subsystems, in particular, modelling interaction between neighbouring subsystems, and making a decision in a group of neighbouring subsystems.

We consider here only probabilistic models and assume that the operating modes, either normal or faulty, of any subsystem can be described by a Markov chain [8]. We propose a consensus-based approach to address problems of cooperation and group decision. For consensus, we determine a common distribution in a group of agents (or experts), an idea that goes back to the early sixties, when Stone [9] introduced pooling of opinions to determine a group decision. Stone assumed that each opinion was modelled by a continuous probability distribution, and the opinion pool as a mixture of distributions. However, he considered only equal weights. DeGroot [10] extended this idea in two ways: first, by introducing an iterative approach whereby each agent revises its own subjective probability distribution based on changes in other group members opinions, and second, by allowing the possibility of distinct weights and specifying convergence conditions for the iteration. The convergence conditions were generalized further by Chatterjee and Seneta [11]. We follow DeGroot's approach by applying an iterative algorithm to determine a common distribution, but we extended this using Markov chains instead of continuous distributions. Additionally, we propose a procedure for optimally selecting weights using an EM framework.

The proposed fault diagnosis algorithm consist of the following set of tasks. The subsystems, obtained by decomposition, comprise the group of agents. All share information, including observations and the most likely parameters of the consensus distribution, each member estimating this distribution. Assuming that the conditions for convergence of the consensus algorithm are satisfied, members of the group come to consensus (an identical consensus distribution). Once consensus is achieved, implementation of the fundamental tasks of fault diagnosis can commence. To decide whether there is a fault, each group member compares its local distribution (model) with a consensus distribution. More precisely, the "distance" between two distributions is measured; if it is greater than 0, a fault is recorded. Thus, we assume that normal operating conditions are described by a single model and that there are a number of models indicating a

978-1-4244-8970-1/11/$26.00 ©2011 IEEE

fault. An essential assumption is that each subsystem can detect its own fault, based on changes in a selected local model.

Once it is established that there are one or more faulty subsystems in the group, the next step is to determine which subsystems are faulty and what type of fault is present. There are two scenarios. In the first a subsystem identifies its own fault, and checks if there are group members affected by the same fault (the agent compares its own local model with other local models in the group by measuring the distance between corresponding distributions). In the second, if the subsystem has no fault then the faulty subsystems can be detected in the same manner by measuring the distance between distributions. The existence of different distance values indicates multiple faults.

In the existing literature on fault diagnosis in distributed systems, the consensus algorithm was applied in the work of Franco *et al.* [12] and Ferrari *et al.* [13]. Although they used a consensus-based approach, it differs from ours in that first and second order moments only were used, rather than the entire distribution as we do here. Petri Nets are used extensively in fault diagnosis in distributed systems [14]. Particularly interesting is the work of Benveniste *et al.* [15], who introduced a probabilistic extension of Petri Nets for distributed and concurrent systems (components that evolve independently) and applied this to fault diagnosis. However, they did not consider the problem of fault diagnosis within a consensus framework. Our work is motivated by the need to diagnose faults in electrical power systems. Accordingly, we mention work of Kato *et al.* [16] and Garza *et al.* [17]. In [16] a multi-agent approach was suggested to locate and isolate fault zones. In [17] this problem was considered within a probabilistic framework using dynamic Bayesian networks. Our approach is different in that we use a consensus framework for fault diagnostics.

The rest of this paper is organized as follows. The problem is formulated in Section II and this is followed by Section III, where the likelihood function for our mixture of Markov chains and the EM procedure for selecting optimal weights are derived. In Section IV an extension to the case of unobserved Markov chains is proposed. Finally, simulation results are presented in Section V.

II. Problem Formulation

In the proposed consensus-based fault diagnosis algorithm for subsystems' modelling, Markov chains are employed. The terms *subsystem* and *agent* will be used interchangeably. It is assumed that each agent has local observations and a local model is represented by a transition probability matrix. Furthermore, agents can exchange information over a computer network; specifically, each agent can know other agents' observations and transition matrices. The idea is to modify the probability distributions within a group of agents in such a way that, under certain conditions, those of all agents in the group converge [11] to a common distribution. Since the proposed algorithm is designed for fault diagnosis in decentralized systems, this alone does not imply that information about the group is collected and processed by a single preselected group member. A common distribution is obtained by exchanging information (messages) and negotiation between group members; this negotiation is effected by an iterative procedure [10] for updating the transition probabilities.

Starting from iteration $\tau = 1$, agent i updates its transition matrix by taking a weighted sum of the other agents' matrices and its own. Explicitly, the updated transition matrix for agent i at iteration $\tau \geq 1$ is given by

$$\mathbf{P}_i^{(\tau)} = \sum_{j=1}^{K} \psi_{ij} \mathbf{P}_j^{(\tau-1)}, \qquad i = 1, \ldots, K, \tag{1}$$

where ψ_{ij} is a coefficient assigned by agent i to agent j to rate influence of agent j on agent i. $\mathbf{P}_i^{(\tau)}$ is the updated transition matrix of agent i at iteration τ and $\mathbf{P}_j^{(\tau-1)}$ is a transition matrix of agent j from the previous iteration participating in the revision of transition matrix of agent i. The transition matrices $\mathbf{P}_j^{(0)}$ at iteration $\tau = 0$ denote initially pooled transition matrices. An essential postulate is that all Markov chains have the same state space $S = \{1, 2, ..., N\}$.

It is assumed that $\Psi = [\psi_{ij}]$ $(K \times K)$ is an ergodic stochastic matrix with strictly positive entries. The updating procedure at iteration τ is represented for all agents in a group by

$$\begin{bmatrix} \mathbf{P}_1^{(\tau)} \\ \mathbf{P}_2^{(\tau)} \\ \vdots \\ \mathbf{P}_K^{(\tau)} \end{bmatrix} = \begin{bmatrix} \psi_{11} I_N & \psi_{12} I_N & \ldots \psi_{1K} I_N \\ \psi_{21} I_N & \psi_{22} I_N & \ldots \psi_{2K} I_N \\ \vdots & \vdots & \vdots \\ \psi_{K1} I_N & \psi_{K2} I_N & \ldots \psi_{KK} I_N \end{bmatrix} \begin{bmatrix} \mathbf{P}_1^{(\tau-1)} \\ \mathbf{P}_2^{(\tau-1)} \\ \vdots \\ \mathbf{P}_K^{(\tau-1)} \end{bmatrix} \tag{2}$$

where I_N is $N \times N$ identity matrix. Define the group transition matrix at iteration τ to be the block matrix $\mathcal{P}^{(\tau)} = [\mathbf{P}_1^{(\tau)} | \mathbf{P}_2^{(\tau)} | ... | \mathbf{P}_K^{(\tau)}]^T$, where T denotes transpose. Then, equation (2) can be expressed compactly as $\mathcal{P}^{(\tau)} = [\Psi \otimes I_N] \mathcal{P}^{(\tau-1)}$, or equivalently

$$\mathcal{P}^{(\tau)} = [\Psi \otimes I_N]^{(\tau)} \mathcal{P}^{(0)}, \tag{3}$$

where $\otimes$ is the Kronecker product [18] and $\mathcal{P}^{(0)} = [\mathbf{P}_1^{(0)} | \mathbf{P}_2^{(0)} | ... | \mathbf{P}_K^{(0)}]^T$ is the block matrix consisting of the initial transition matrices participating in the algorithm. Convergence of (2) is assured under the condition that $\Psi \otimes I_N$ is a contraction, that is, $\|\Psi \otimes I_N\| \leq 1$. We exploit the following properties of the Kronecker product [18], [19]:

Lemma 1: If A is an $m_A \times n_A$ matrix and B is an $m_B \times n_B$ matrix, then, for all $\|\cdot\|_p$-norms,

$$\|A \otimes B\| = \|A\| \|B\|.$$

Lemma 2: If A and B are square matrices, then

$$(A \otimes B)^n = A^n \otimes B^n.$$

Since $\|\Psi\|_\infty = 1$ and $\|I_N\|_\infty = 1$, applying Lemma 1 to $\Psi \otimes I_N$ shows that $\|\Psi \otimes I_N\| \leq 1$. Furthermore, applying Lemma 2 to the group transition matrix $\mathcal{P}^{(\tau)}$, given by (3), we obtain

$$\mathcal{P}^{(\tau)} = [\Psi^{(\tau)} \otimes I_N^{(\tau)}]\mathcal{P}^{(0)}. \tag{6}$$

As τ goes to infinity $\mathcal{P}^{(\tau)}$ approaches to the group consensus matrix $\mathcal{P}_c$ defined by

$$\mathcal{P}_c = \begin{bmatrix} \mathbf{P}_c \\ \mathbf{P}_c \\ \vdots \\ \mathbf{P}_c \end{bmatrix} = \begin{bmatrix} \pi_{\Psi_1} I_N & \pi_{\Psi_2} I_N \dots \pi_{\Psi_K} I_N \\ \pi_{\Psi_1} I_N & \pi_{\Psi_2} I_N \dots \pi_{\Psi_K} I_N \\ \vdots & \vdots \qquad \vdots \\ \pi_{\Psi_1} I_N & \pi_{\Psi_2} I_N \dots \pi_{\Psi_K} I_N \end{bmatrix} \begin{bmatrix} \mathbf{P}_1^{(0)} \\ \mathbf{P}_2^{(0)} \\ \vdots \\ \mathbf{P}_K^{(0)} \end{bmatrix} \tag{7}$$

where $\pi_\Psi = [\pi_{\Psi_1} \pi_{\Psi_2} \dots \pi_{\Psi_K}]$ is the the limiting distribution of the stochastic matrix Ψ.

It follows that, for the iterative procedure (6) to converge, the weights must be chosen so that Ψ is ergodic [11]. It remains to specify how to estimate entries of our stochastic matrix in the first iteration of the algorithm. Since the right-hand side of (1) is a mixture of transition matrices [20], [21], the weights ψ_{ij} of agent i can be interpreted as the distribution of a latent variable.

In the next section we derive the likelihood function for our Markov chain mixture, which is used in the subsequent EM framework to estimate our weights (consensus ratings). Once the optimal ratings are estimated and the corresponding Ψ is formed, its stationary distribution (denoted by π_Ψ) can be evaluated. From that and the initially pooled transition matrices $\mathcal{P}^{(0)}$, a consensus distribution $\mathbf{P}_c$ and corresponding stationary distribution π_c can be determined.

III. Estimation of Optimal Ratings in the Consensus Problem

To estimate weights in our algorithm we must determine a likelihood function for the linear combination of transition matrices involved in the consensus scheme

$$\mathbf{P}_i = \psi_{i1}\mathbf{P}_1 + \psi_{i2}\mathbf{P}_2 + \dots + \psi_{iK}\mathbf{P}_K = \sum_{j=1}^{K} \psi_{ij}\mathbf{P}_j \tag{8}$$

where $\mathbf{P}_i$, $\mathbf{P}_j$ and ψ_{ij} refer to the i^{th} agent in (2) for $\tau = 1$ (the iteration indices are omitted). In a group of agents it is assumed that each has observed its own state sequence and corresponding transition matrix. When a particular agent i revises its own transition matrix, it invites the other agents to transmit theirs. Agent i will then adapt its own transition matrix based on the information received. We will explain how these weights depend on state sequences and corresponding transition probabilities of each of the agents in the group. In a slight abuse of notation, ψ_{ij} will be replaced by ψ_j, meaning that agent i changes its distribution under the influence of information coming from agent j (the index i is omitted). We follow an approach of Anderson and Goodman [22], but extended this to Markov chain mixtures.

It is assumed that the state sequence $\{X_t,\, 0 \leq t \leq T\}$ of agent i is driven by (8). Transitions in this sequence can come from one of K agents in the group. Since each agent k is weighted by some value ψ_j, the probability of a particular transition at time t, from state x_{i-1} to state x_i, can be modelled as the product of two probabilities: the probability of a transition from one state to another and the probability that the transition itself is caused by agent k. Consequently, the probability of the state sequence $x_0, \dots, x_T$ is

$$\psi_{(x_0 x_1)_k} p_{(x_0 x_1)_k} \psi_{(x_1 x_2)_k} p_{(x_1 x_2)_k} \dots \psi_{(x_{T-1} x_T)_k} p_{(x_{T-1} x_T)_k} \tag{9}$$

Expression (9) can be further extended by introducing a random process (Z_t) to model random selection of the source k of a particular transition from x_{i-1} to x_i. Since this transition at time t can come from only one source, an indicator $I_{\{Z_t=k\}}$ of this source is introduced. In that case, the weight $\psi_{(x_{i-1}x_i)_k}$ can be interpreted as the probability that a particular transition probability $p_{(x_{i-1}x_i)_k}$ comes from agent k, denoted as $P(Z_t = k)$. Thus, for each transition from x_{i-1} to x_i, expression (9) is modified to obtain

$$\prod_{k=1}^{K} \{\psi_{(x_0 x_1)_k} p_{(x_0 x_1)_k}\}^{I_{\{Z_1=k\}}} \prod_{k=1}^{K} \{\psi_{(x_1 x_2)_k} p_{(x_1 x_2)_k}\}^{I_{\{Z_2=k\}}} \\ \cdots \prod_{k=1}^{K} \{\psi_{(x_{T-1} x_T)_k} p_{(x_{T-1} x_T)_k}\}^{I_{\{Z_T=k\}}}_{.} \tag{10}$$

The next step towards calculating (9) requires counting the number of transitions, from x_{i-1} to x_i for agent k until time t on the entire sequence, as follows:

$$\prod_{x_0 x_1} \prod_{k=1}^{K} \{\psi_{(x_0 x_1)_k} p_{(x_0 x_1)_k}\}^{I_{\{Z_1=k\}} N_1(x_0 x_1)_k} \\ \prod_{x_1 x_2} \prod_{k=1}^{K} \{\psi_{(x_1 x_2)_k} p_{(x_1 x_2)_k}\}^{I_{\{Z_2=k\}} N_2(x_1 x_2)_k} \dots \\ \prod_{x_{T-1} x_T} \prod_{k=1}^{K} \{\psi_{(x_{T-1} x_T)_k} p_{(x_{T-1} x_T)_k}\}^{I_{\{Z_T=k\}} N_T(x_{T-1} x_T)_k}_{,} \tag{11}$$

or, more compactly,

$$\prod_{t=1}^{T} \prod_{i,j=1}^{N} \prod_{k=1}^{K} \{P(Z_t{=}k) \times \\ \times P(X_t{=}j | X_{t-1}{=}i, Z_t{=}k)\}^{I_{\{Z_t=k\}} N_t(X_{t-1}=i, X_t=j)_k} \tag{12}$$

To simplify notation, $P(Z_t = k)$ will be denoted as ψ_k, the transition probability $P(X_t = j | X_{t-1} = i, Z_t = k)$ will be given in the shortened form $P_t(i,j)_k$ and the number of transitions by time t for a particular agent k by $N_t(i,j)_k$:

$$\prod_{t=1}^{T} \prod_{i,j=1}^{N} \prod_{k=1}^{K} \{\psi_k P_t(i,j)_k\}^{I_{\{Z_t=k\}} N_t(i,j)_k} \tag{13}$$

It is apparent from (13) that the random variable Z_t is not directly observable. However, this incomplete-data problem can be converted to a complete-data problem; if the problem is extended to find the likelihood of the sequence $\{(X_t, Z_t), 0 \leq t \leq T\}$ instead, it opens up the possibility of using the EM framework [23].

As previously discussed, the expression (13) is a likelihood function of the complete-data vector whose logarithm is given by

$$\log P(\mathbf{X}_0^T, \mathbf{Z}_1^T) = \sum_{t=1}^{T}\sum_{i,j=1}^{N}\sum_{k=1}^{K} I_{\{Z_t=k\}} N_t(i,j)_k \times \\ \times \{\log P_t(i,j)_k + \log \psi_k\} \tag{14}$$

The EM algorithm is a two-step iterative procedure. In the first step, called the E-step, the Q function is calculated, which is the mathematical expectation of (14) given observations $\{X_t, 0 \leq t \leq T\}$: $Q = E\{\log P(\mathbf{X}_0^T, \mathbf{Z}_1^T)|\mathbf{X}_0^T\}$. Its evaluation is reduced to computing the mathematical expectations of indicator functions, because the transition probabilities $P_t(i,j)_k$, counts $N_t(i,j)_k$ and initial mixing proportions ψ_k are known in advance. By Bayes' theorem

$$P(A|B \cap C) = \frac{P(B|A \cap C)P(A|C)}{P(B|C)} \tag{15}$$

Furthermore, assuming that X_t depends only on X_{t-1} and $I_{\{Z_t=k\}}$, as well as presuming that $I_{\{Z_t=k\}}$ and X_{t-1} are independent, the mathematical expectation of the indicator function is given as follows:

$$E\{I_{\{Z_t=k\}}|\mathbf{X}_0^t\} \\ = \frac{P(X_t=j|X_{t-1}=i, I_{\{Z_t=k\}}=1)P(I_{\{Z_t=k\}}=1)}{P(X_t=j|X_{t-1}=j)} \\ = \frac{P(X_t=j|X_{t-1}=i, I_{\{Z_t=k\}}=1)P(I_{\{Z_t=k\}}=1)}{\sum_{h=1}^{K} P(X_t=j|X_{t-1}=i, I_{\{Z_t=h\}}=1)P(I_{\{Z_t=h\}}=1)} \\ = \frac{\psi_k P_t(i,j)_k}{\sum_{h=1}^{K} \psi_h P_t(i,j)_h} = \varphi_t(i,j)_k \tag{16}$$

Finally an expression for Q is given by

$$Q = \sum_{t=1}^{T}\sum_{i,j=1}^{N}\sum_{k=1}^{K} \varphi_t(i,j)_k N_t(i,j)_k \{\log P_t(i,j)_k + \log \psi_k\} \tag{17}$$

In the second step of the EM algorithm, called the M-step, previously assumed parameters are optimized based on the expectation of the log likelihood. Applying Lagrange's method to (17), with the constraint $\sum_{k=1}^{K} \psi_k = 1$, the following expression is obtained:

$$Q = \sum_{t=1}^{T}\sum_{i,j=1}^{N}\sum_{k=1}^{K} \varphi_t(i,j)_k N_t(i,j)_k \times \\ \times \{\log P_t(i,j)_k + \log \psi_k\} - \lambda \left(\sum_{k=1}^{K} \psi_k - 1\right). \tag{18}$$

After taking the derivative of Q with respect to ψ_k we get

$$\frac{\partial Q}{\partial \psi_k} = \frac{\sum_{t=1}^{T}\sum_{i,j=1}^{N} \varphi_t(i,j)_k N_t(i,j)_k}{\psi_k} - \lambda = 0 \tag{19}$$

Rearranging (19) and using the constraint we obtain

$$\lambda = -\sum_{h=1}^{K}\sum_{t=1}^{T}\sum_{i,j=1}^{N} \varphi_t(i,j)_h N_t(i,j)_h.$$

Finally, the updated equation for ψ_k is given by

$$\psi_k = \frac{\sum_{t=1}^{T}\sum_{i,j=1}^{N} \varphi_t(i,j)_k N_t(i,j)_k}{\sum_{h=1}^{K}\sum_{t=1}^{T}\sum_{i,j=1}^{N} \varphi_t(i,j)_h N_t(i,j)_h}. \tag{20}$$

The previously described procedure is the first step of the EM algorithm: $\iota = 0$. By altering the E-step and M-step, in each iteration ι, the function $Q_{(\iota)}$ is calculated and the parameters are optimized. From an implementation point of view there are two ways to halt the procedure. The first is when the difference in the value of Q is below some threshold Θ assumed in advance, that is, $Q_{(\iota)} - Q_{(\iota-1)} \leq \Theta$. The second is to specify in advance the total number of iterations Υ and stop when $\iota \geq \Upsilon$.

IV. Extension to Unobservable Markov Chains

In the previous section, observable Markov chains were considered, where each state of the system corresponds to an observable discrete event. This is not realistic, as states are not usually directly observable, if for no other reason than because of measurement noise. So, we must extend our algorithm to the case where the states are hidden: Hidden Markov Models (HMM).

In order to apply HMM [24], [25] methods here, it will be necessary to establish a link with the solutions of the three essential problems of HMMs described by Rabiner [24]. The basic elements of a HMM are the number of states N, the number of distinct observation symbols (discrete observations) $O=\{o_1, o_2, \ldots, o_M\}$, or the number of mixture components (continuous observations) M, and, the state transition probabilities $\boldsymbol{A}=[a_{ij}]_{N\times N}$, where

$$a_{ij} \equiv P(X_t = j|X_{t-1} = i), \ 1 \leq i, j \leq N. \tag{21}$$

The next element is an observation probability defined as a matrix $\boldsymbol{B}=[b_j(y_t)]_{N\times 1}$, where $b_j(y_t)$ is a probability of a particular observation. For discrete observations,

$$b_j(y_t) \equiv P(Y_t = o_m|X_t = j), \ 1 \leq j \leq N, \ 1 \leq m \leq M. \tag{22}$$

A general representation of a continuous observation process is via a finite mixture of the form

$$b_j(y_t) \equiv \sum_{k=1}^{M} \alpha_{jk} f(y_t; \Theta_{jk}), \ 1 \leq j \leq N, \ 1 \leq m \leq M. \tag{23}$$

The final element is the initial state distribution $\mathbf{\Pi} = [\pi_k]_{N\times 1}$. We use the following compact notation:

$$\lambda = (\boldsymbol{A}, \boldsymbol{B}, \mathbf{\Pi}), \tag{24}$$

where N and M are defined implicitly.

The elements of a HMM are connected through the three basic problems identified by Rabiner [24].

Problem 1 : The efficient computation of the probability $P(Y|\lambda)$ of the observed sequence $Y = Y_1, Y_2, ..., Y_T$, given the model $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Pi})$

Problem 2 : Estimation of the hidden state sequence $X = X_1, X_2, ..., X_T$, given the observation sequence $Y = Y_1, Y_2, \dots, Y_T$ and a model $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Pi})$

Problem 3 : Estimation of the parameters of the model $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Pi})$, to maximize $P(Y|\lambda)$.

To solve Problem 1 we used the forward-backward procedure, which is a recursive method for efficient calculation of the probability of the observed sequence. To maximize the probability of a hidden state sequence for given observations, the Viterbi algorithm is used for Problem 2. Problem 3 is addressed using the Baum-Welch algorithm.

The algorithm described in Sections II and III is associated with the HMM in the following way. Suppose that each agent i in the group has an identical set of HMM models $\Lambda^i = \{\lambda_i(\Theta_1)\dots\lambda_i(\Theta_R)\}$, where $\lambda_i(\Theta_k)$ is model (24) and Θ_k is the vector of the parameters of a model k. It is assumed that the R models in Λ^i fully describe local behaviour in that normal and faulty regimes are modelled by this set. Also, as previously introduced, each agent i has its own set of observations Y_t^i=$\{y_1^i, y_1^i, \dots, y_T^i\}$.

The Baum-Welch algorithm is applied to get the models for normal and faulty working regimes in Λ^i. The forward-backward procedure is used to select the most likely model for given observations. Finally, the Viterbi algorithm is used to estimate the hidden sequence X_t^i=$\{x_1^i, x_2^i, ..., x_T^i\}$ corresponding to the observations Y_t^i=$\{y_1^i, y_2^i, ..., y_T^i\}$.

To summarize, for each agent i in the group, once the most likely model $\lambda_i(\Theta_k)$ for given observations Y_t^i is determined and the hidden sequence X_t^i is estimated, the methodology introduced in Sections II and III can be applied directly.

V. Simulation Results

We illustrate the method using a hypothetical system that can be decomposed into three subsystems, the corresponding agents labelled 1, 2 and 3. In addition it is supposed that there are three possible agent working regimes, each modelled by two-state Markov chains. Two Markov chains (transition matrices $\mathbf{A}_1$ and $\mathbf{A}_2$) represent faulty modes, and the third ($\mathbf{A}_3$) models normal operating mode; they are specified in Table I. The corresponding stationary distributions [26] are $\pi_{\mathbf{A}_1}$=[0.2052 0.7948], $\pi_{\mathbf{A}_2}$=[0.7717 0.2283] and $\pi_{\mathbf{A}_3}$=[0.4953 0.5047], respectively. As indicated earlier, part of our fault diagnosis scheme is based on measuring the distance between distributions. To measure the distance between the stationary distributions, π_1 and π_2, we use the L_2-norm:

$$\delta(\pi_1, \pi_2)=\sum_{i=1}^{N}\left(\sum_{j=1}^{i}\pi_1(j) - \sum_{j=1}^{i}\pi_2(j)\right)^2, \qquad (25)$$

where N is a number of discrete states. Using (25) a fault table (dictionary) can be created. This is available to all agents; once an agent determines its own operating mode, it is used to uniquely identify the mode of the other agents in the group relative to its own. Table II shows the fault table for our example. Since it is symmetric, the total number of different values, as a function of the number of operation modes n, is $2^{-1}(n^2 - n)$.

TABLE I
Transition probabilities of operating modes

$\mathbf{A}_1$		$\mathbf{A}_2$		$\mathbf{A}_3$	
0.4745	0.5255	0.8343	0.1657	0.5009	0.4991
0.1357	0.8643	0.5601	0.4399	0.4899	0.5101

TABLE II
The Fault Table (Dictionary)

$\delta(\pi_1, \pi_2)$	$\pi_{\mathbf{A}_1}$	$\pi_{\mathbf{A}_2}$	$\pi_{\mathbf{A}_3}$
$\pi_{\mathbf{A}_1}$	0	0.3209	0.0842
$\pi_{\mathbf{A}_2}$	0.3209	0	0.0764
$\pi_{\mathbf{A}_3}$	0.0842	0.0764	0

Suppose further that for each Markov chain the state sequences are given on interval $t \in [1, T]$ with $T = 100$ (see Fig. 1), denoted as $X_1(t)$, $X_2(t)$ and $X_3(t)$, respectively. The total number of transitions from states i to j is counted in order to determine matrices $N_t(i, j)_k$, for $k = 1, \dots, K$ and $t = 1, \dots, T$, as given in Table III. To start

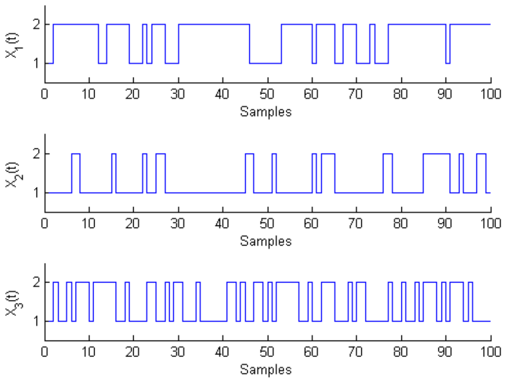

Fig. 1. State sequences $X_1(t)$,$X_2(t)$ and $X_3(t)$

TABLE III
Number of transitions in state sequence

$\mathbf{N}_1$		$\mathbf{N}_2$		$\mathbf{N}_3$	
16	11	63	12	28	25
10	62	12	12	25	21

the EM algorithm, entries in Ψ_0^{EM} were randomly chosen. After the EM algorithm is halted, Ψ is formed based on estimated values of mixture proportions. For the resulting Ψ, the stationary distribution π_{Ψ} is computed. Using this, a consensus $\mathbf{P}_c$ is determined, and, once estimated, its stationary distribution π_c can be calculated. Finally, by applying (25), any discrepancy between the consensus stationary distribution and stationary distribution of the present operation mode can be detected, as discussed in Section I and summarized in **Algorithm 1** below.

Algorithm 1 Fault Diagnosis - agent i

```
if $\delta(\pi_c, \pi_{P_i^{(0)}}) > 0$ then
    for all $j \in K \backslash \{i\}$ do
        if $\delta(\pi_{P_i^{(0)}}, \pi_{P_j^{(0)}}) > 0$ then
            Search the fault table to identify the model $P_j^{(0)}$
            associated with agent $j$
        end if
    end for
end if
```

A. One faulty agent in the group

Suppose that one of the agents, for example agent 1, has a fault, modelled by $\mathbf{A}_1$. In that case $\mathbf{P}_1^{(0)}$=$\mathbf{A}_1$, $\mathbf{P}_2^{(0)}$=$\mathbf{A}_3$ and $\mathbf{P}_3^{(0)}$=$\mathbf{A}_3$ are the initially pooled transition matrices. The results of the estimation procedure are summarized in Tables IV and V. Once the stationary distribution π_c of the consensus matrix is determined, **Algorithm 1** is applied. Since $\delta(\pi_c, \pi_{P_1^{(0)}})$=0.0285 (> 0) and $\delta(\pi_c, \pi_{P_2^{(0)}})$=$\delta(\pi_c, \pi_{P_3^{(0)}})$=0.0147 (> 0), each agent will detect that a fault has occurred in the system. In the second step, each agent compares the distance between the stationary distribution of its operating mode with stationary distributions of the other agents in the group. Since each agent has a same fault table, a simple search allows each to identify the working conditions of others. Since the estimated distances are $\delta(\pi_{P_1^{(0)}}, \pi_{P_2^{(0)}})$=$\delta(\pi_{P_1^{(0)}}, \pi_{P_3^{(0)}})$=0.0842 and $\delta(\pi_{P_2^{(0)}}, \pi_{P_3^{(0)}})$=0, this implies an operating mode of $\mathbf{A}_1$ for agent 1 and $\mathbf{A}_3$ for agents 2 and 3.

TABLE IV
Optimal ratings in a group of Markov chains

Ψ_0^{EM}			Ψ			π_{Ψ}^T
0.3084	0.3833	0.3083	0.5182	0.2670	0.2148	0.5182
0.4198	0.1264	0.4538	0.5182	0.1050	0.3768	0.2484
0.6548	0.2576	0.0876	0.5182	0.3596	0.1222	0.2334

TABLE V
Consensus matrix and its stationary distribution

P_c		π_c^T
0.4872	0.5128	0.3740
0.3064	0.6936	0.6260

B. Identical agents in the group

When the transition matrices are identical, (16) depends only on the initial weights. That is, $\varphi_t(i,j)_k = \psi_k(0)$ for all i and j. Since all state sequences have a same length $\sum_{t=1}^{T} \sum_{i,j=1}^{N} N_t(i,j)_k = T-1$, the updating equation (20) is $\psi_k(1) = \psi_k(0)$ for all $k = 1, ..., K$. Since ψ_k stays unchanged after the first iteration, the EM algorithm stops. Similarity is detected and computation time saved. This occurs when all agents are non faulty or when all are effected by a same fault type.

VI. Discussion

A fault diagnosis scheme is proposed for distributed systems, in which subsystems are represented by agents. Operating modes of agents are modelled by Markov chains. The agents form a group, whose common (consensus) transition matrix is estimated. Change in consensus within the group is monitored and, once a change is detected, the distances between the stationary distributions of operating modes are estimated in order to identify the new condition of the system. Future work includes the practical implementation of our algorithm to fault diagnosis in power systems. To be fully applicable it will be necessary to extend the HMM to continuous observation schemes in order to deal with a broad class of nonlinear time series.

VII. Acknowledgment

This work was supported by the Australian Research Council Centre of Excellence for Mathematics and Statistics of Complex Systems (MASCOS). We thank the anonymous reviewers for their valuable comments and suggestions.

References

[1] R. Patton, P. Frank, and R. Clark, Eds., *Issues of fault diagnosis for dynamic systems.* London : New York: Springer, 2000.

[2] J. Gertler, *Fault detection and diagnosis in engineering systems.* Marcel Dekker, Inc., 1998.

[3] S. X. Ding, *Model-based fault diagnosis techniques: design schemes, algorithms, and tools.* Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2008.

[4] L. H. C. E. Russell and R. D. Braatz, *Data-driven methods for fault detection and diagnosis in chemical processes.* New York : Springer, 2000.

[5] S. Simani, C. Fantuzzi, and R. J. Patton, *Model-based fault diagnosis in dynamic systems using identification techniques.* London ; New York : Springer, 2003.

[6] D. Siljak, *Decentralized control of complex systems.* Academic Press, 1991.

[7] E. Fabre and V. Pigourier, "Monitoring distributed systems with distributed algorithms," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 1, 2002, pp. 411 – 416 vol.1.

[8] J. R. Norris, *Markov chains.* Cambridge ; New York : Cambridge University Press, 1997.

[9] M. Stone, "The opinion pool," *The Annals of Mathematical Statistics*, vol. 32, no. 4, pp. 1339–1342, 1961.

[10] M. H. DeGroot, "Reaching a consensus," *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118–121, 1974.

[11] S. Chatterjee and E. Seneta, "Towards consensus: Some convergence theorems on repeated averaging," *Journal of Applied Probability*, vol. 14, no. 1, pp. pp. 89–97, 1977.

[12] E. Franco, R. Olfati-Saber, T. Parisini, and M. Polycarpou, “Distributed fault diagnosis using sensor networks and consensus-based filters,” 2006, pp. 386–391.

[13] R. Ferrari, T. Parisini, and M. Polycarpou, “Distributed fault diagnosis with overlapping decompositions and consensus filters,” in *American Control Conference, 2007. ACC ’07*, 2007, pp. 693 –698.

[14] A. Aghasaryan, E. Fabre, A. Benveniste, R. Boubour, and C. Jard, “Fault detection and diagnosis in distributed systems: An approach by partially stochastic petri nets,” *Discrete Event Dynamic Systems*, vol. 8, pp. 203–231, 1998.

[15] A. Benveniste, E. Fabre, and S. Haar, “Markov nets: probabilistic models for distributed and concurrent systems,” *Automatic Control, IEEE Transactions on*, vol. 48, no. 11, pp. 1936 – 1950, 2003.

[16] T. Kato, H. Kanamori, Y. Suzuoki, and T. Funabashi, “Multiagent based control and protection of power distributed system - protection scheme with simplified information utilization - ,” in *Intelligent Systems Application to Power Systems, 2005. Proceedings of the 13th International Conference on*, 2005, pp. 49 –54.

[17] L. Garza, F. Cantu, and S. Acevedo, “Integration of fault detection and diagnosis in a probabilistic logic framework,” in *Advances in Artificial Intelligence, IBERAMIA 2002*, F. Garijo, J. Riquelme, and M. Toro, Eds. Springer Berlin Heidelberg, 2002, vol. 2527, pp. 265–274.

[18] R. A. Horn and C. R. Johnson, *Topics in matrix analysis.* Cambridge ; Melbourne : Cambridge University Press, 1991.

[19] A. N. Langville and W. J. Stewart, “The kronecker product and stochastic automata networks,” *Journal of Computational and Applied Mathematics*, vol. 167, no. 2, pp. 429 – 447, 2004.

[20] T. Batu, S. Guha, and S. Kannan, “Inferring mixtures of markov chains,” National Institute of Standards and Technology, Tech. Rep., 2002.

[21] H. Frydman, “Estimation in the Mixture of Markov chains,” *SSRN eLibrary*, 2003.

[22] T. W. Anderson and L. A. Goodman, “Statistical inference about markov chains,” *The Annals of Mathematical Statistics*, vol. 28, no. 1, pp. 89–110, 1957.

[23] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[24] L. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 2, pp. 257–286, 1989.

[25] O. Cappé, E. Moulines, and T. Ryden, *Inference in Hidden Markov Models.* Springer, 2005.

[26] W. J. Stewart, *Introduction to the numerical solution of Markov chains.* Princeton, N.J. : Princeton University Press, 1994.