

# A Monitoring and Diagnosis System for Electric Machines Based on Bayesian Networks

José Leonardo Flores Quintanilla  
ITESM San Luis campus, México  
Mechatronics Undergraduate Program  
Av. Eugenio Garza Sada 300  
C.P. 78211 San Luis Potos, SLP., México  
joselfq@itesm.mx

Rubén Morales-Menéndez  
Center for Industrial Automation  
ITESM Monterrey campus, México  
Av. Eugenio Garza Sada 2501  
C.P. 64849 Monterrey, N.L.  
rmm@itesm.mx

Luis Eduardo Garza Castañón  
Mechatronics Department  
ITESM Monterrey campus, México  
Av. Eugenio Garza Sada 2501  
C.P. 64849 Monterrey, N.L.  
legarza@itesm.mx

Juan Pablo Nieto González  
ITESM Saltillo campus, México  
Mechatronics Department  
Prol. Juan de la Barrera Ote. 1241  
C.P. 25270 Saltillo, Coahuila  
juan.pablo.nieto@itesm.mx

Ricardo Ramírez  
Mechatronics Undergraduate Program  
ITESM Monterrey campus, México  
Av. Eugenio Garza Sada 2501  
C.P. 64849 Monterrey, N.L.  
ricardo.ramirez@itesm.mx

## Abstract

This paper shows the integration of Bayesian networks (BN) and Particle Filtering (PF) to design a system of fault detection and diagnosis. In a production line there are several interconnected machines which have several components. If a failure occurs in this production line, it is very difficult to find the faulty element. BN are used to predict a set of machines which have a high probability to fail. Then, these machines are monitored continuously to quickly detect when a component has failed. We tested our ideas in the simulation of a productive process which has two machines. The objective is to find the faulty machine and its damaged element.

**Keywords:** FDI, Bayesian Networks, Particle Filtering, look ahead Rao-Blackwellized Particle Filter.

ers and microprocessors began. As soon as the use of the computers and microprocessors were expanded to automation and monitoring of several processes, a method of supervising hundreds of control loops was required. However, these systems require a diagnosis to establish some decisions. A kind of supervision to tell human beings the status of certain physical variables is necessary. So that we determine whether the system is working properly or not.

When a machine fails in a real process, the maintenance people look for the faulty elements, and correct them, which take minutes, hours or even days to finish depending on the complexity of the failure. Therefore, if the maintenance personnel knows in advance the damaged elements the production line will stop for a short time.

## 1.1 Fault Diagnosis Methods

The task of fault diagnosis is difficult due to complex fault scenarios. For example, mechanical, electrical and electronic devices are related, work together. Uncertainty in information, and lack of do-

## 1 Introduction

As a result of experimental research and the need for treatment of a lot of data, the use of comput-

main knowledge.

According to [4] fault detection methods are classified in model-free methods: physical redundancy, special sensors, limit checking, etc., and model-based methods: Kalman filter, diagnostic observers, parity relations, etc.

Model-based methods utilize an explicit mathematical model of the monitored plant; their natural mathematical description is in the form of differential equations, or equivalent transformed representations. Most of the model based fault detection and diagnosis methods rely on analytical redundancy.

## 1.2 Bayesian networks fault detection and diagnosis

The previously mentioned methods are powerful if the designer considers all the variables of the modeled-process being represented. However, it is very difficult to establish all the model equations as the system grows in complexity. Furthermore, all these methods do not take into account uncertainty, which is important because the presence of uncertainty changes radically in the way decisions are made.

BNs are frameworks which do not need a complete mathematical model of the system, and they work well in the presence of uncertainty [4]. That is why BNs are an important alternative that solve fault diagnostic problems.

We will show the integration of BNs to solve a fault diagnosis problem.

## 1.3 Problem description

A production line in a factory can be thought of as a set of interconnected machines. These machines are made up of several components which can be seen as series RL circuits. As the ratio of possible failures in this system is high, because of the number of total components, a monitoring system for faulty component is required. A monitoring system which consists of two phases, can be implemented in the following way:

Phase I. A BN based on data, models the system behavior in probabilistic terms. It considers that we have  $n$  interconnected machines. The failure probabilities of each machine, based on evidences, are updated periodically. The BN is used to select the machines with a high probability of failure. These machines are continuously monitored to find the damaged components.

When Phase I is completed, the diagnosis system will make different decisions depending on the

results in Phase I according to the following.

1. If the probability of failure of the machines does not change, then PF will continue monitoring the same machines to find damaged components.
2. If the probability of failure, in a non monitored machine, increases to a warning level, then a PF must be started to monitor this machine.
3. If the probability of failure, in a monitored machine, increases, then, PF must increase its reliability, taken a greater number of particles.
4. If the probability of failure, in a monitored machine, decreases, then PF must reduce the number of particles in a first stage. If the tendency remains, the PF must stop in that machine.

Phase II. In this phase, a Particle Filter algorithm is continuously running in the selected machines in Phase I, to detect a failure in the moment that it appears. PF increases, decreases its number of particle depending on the results of Phase I.

The initial BN will be updated frequently to avoid remaining static. Time and effort are required to update the BN. If failures frequently appear in Phase I, possibly Phase II will not detect the problem unless the problem comes from a machine where PF is already running.

# 2 Methodology

## 2.1 Considerations

We simulate a part of a productive process where there are two machines controlled and supervised by the same control panel, Figure 1.

Each machine is composed of two or three identical RL series circuits. Besides, each machine as well as its individual circuits has its breakers which protect the rest of the system against overcurrent. Also, these breakers are used when machines or individual circuits need maintenance.

The interaction between continuous and discrete variables is shown in this application. Discrete variables are the machines and circuit breakers, and the continuous variables are given by the main current in the whole system as well as the circuit parameters.

The main tasks are to determine when there is a fault, which of the two machines is faulty mode and which circuit has the problems. In addition, we need to show which of the two components of the individual faulty circuit are damaged and what kind of

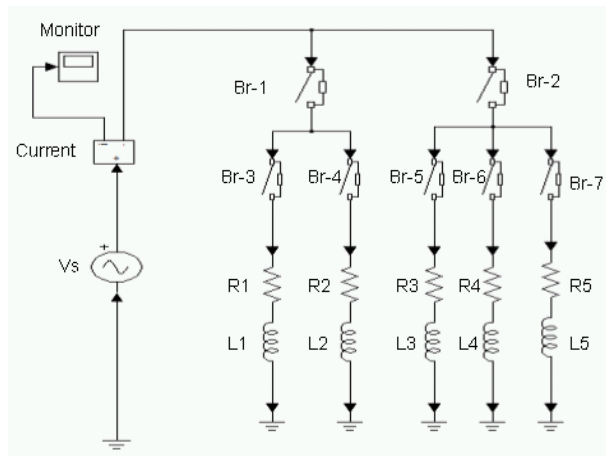


Figure 1: Schematic diagram of the two machines with their serie resistance-inductance circuits.

fault it has, according to the limits permitted. That is, say whether the faulty component passed the low or high limit.

A normal condition is when none of the circuit components have passed its low or high limit values. These limits are calculated using a change of 20% of the original values of the components. Therefore, if there is only one circuit connected and there is a change higher than 5% in the main current, a fault is probably occurring. Similarly, if there are two circuits connected the change in main flow current indicates a probable fault is 2.5 %; three connected circuits is 1.70 %; if there are four connected circuits the respective main current indicates a probable fault is 1.25 % and for five it is 1.10 %.

## 2.2 Design of the fault detection and diagnosis system

It is important to say that a requirement to design intelligent systems is modularity, which makes their analysis more feasible and prepares them to grow easily. This action will help the designer too, because it is easy to generate specialized agents to do certain tasks and later join those agents to generate a complex system. Figure 2 shows the general proposed architecture, [3].

In Figure 2 the problem is divided in two important phases, each having its requirements.

We will describe each block:

1. Facts, assumptions, knowledge base. This block contains all the known facts about the process under diagnosis, together with a set of assumptions regarding the behavior of the process under certain conditions.

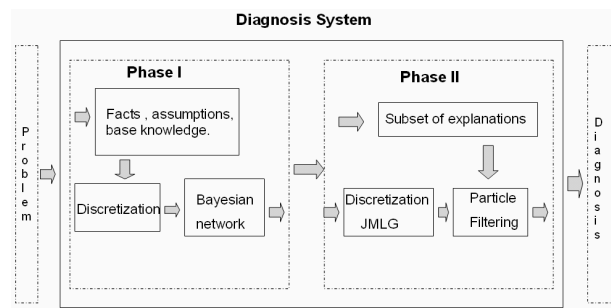


Figure 2: General architecture for the diagnosis system.

2. Data discretization for first phase. Initially continuous variables are discretized. In our application the only continuous variable that we discretized was the main current.
3. Bayesian network. This block contains a specialized software "Power Constructor"<sup>1</sup> developed to create BN; It uses data bases in the form of discrete variables. We treat the BN using evidence in a set of variables and look at the rest of variables to predict the possible faults that could be presented. We use Hugin<sup>2</sup> software in this step.

A BN is a data structure used to represent dependencies among variables; this network has the full joint probability distribution [10]. The nodes of the BN are random variables that may be continuous or discrete; each node has a conditional probability distribution which gives us information about the effect of the parents on the node. The node X is said to be a parent of the node Y if there is an arrow from node X to node Y. Furthermore, if the network has time varying relations between its variables, the network is called dynamic Bayesian network (DBN).

4. Subset of explanations. Once that data has been analyzed a subset of explanations is given where a possible fault is located in the system. This is just a subset of the results, taking into account just the ones with major values in probability, as indicated.
5. Data discretization for second phase. Discretization is needed for the continuous dynamic behavior of the system.
6. Particle Filtering. This is a block of the final part of the diagnosis, where we use PF techniques.

<sup>1</sup><http://www.cs.ualberta.ca/~jcheng/bnpc.htm>

<sup>2</sup><http://www.hugin.com>

7. Final Diagnosis. A final diagnosis is given according to the integration of the two techniques named above and shown in following subsections.

## 2.3 Implementation of Phase I

We will briefly describe the basic steps for the first phase:

1. The first step was the creation of a table, which describes the behavior of all the breakers which constitute the whole system as well as the behavior of the main current. This means we made a table with all of combinations of breaker states; as a result, the data is generated for the BN of the whole system.
2. The second step was the discretization of the variable called current, which is a continuous variable. The result are ten different fault states.
3. The third step was the construction of the Bayesian networks using the software "Power constructor". This is done by importing the data base of all discretized system variables from Excel.
4. The fourth step is to use Hugin, a specialized software, to make inferences and analysis in Bayesian networks.

## 2.4 Implementation of Phase II

Given the most recent evidence, we know which machines have the highest probability of failing having completed the first phase. Now, in Phase II, we will monitor those machines to detect which component is failing.

### 2.4.1 Particle Filter algorithm

A Particle Filter (PF), a Markov chain Monte Carlo algorithm, approximates the belief state using a set of samples called particles. The distribution of the particles is updated taking into account the last available samples as time increases. The standard PF algorithm consists of three sequential steps:

1. Monte Carlo step. This step takes into account the evolution of the system as time increases. Mathematically, it has the following representation, [2] :

$$z_t \sim p(z_t | z_{t-1}) \quad (1)$$

$$x_{t+1} = A(z_t)x_t + B(z_t)\gamma_t + F(z_t)u_t \quad (2)$$

$$y_t = C(z_t)x_t + D(z_t)v_t \quad (3)$$

The previous stochastic model of the system is used to generate the predicted future state for each particle. We sampled a discrete mode equation (4) then the continuous state given the new discrete mode equation (5) .

$$\hat{z}_t^{(i)} \sim p(z_t | z_{t-1}^{(i)}) \quad (4)$$

$$\hat{x}_t^{(i)} \sim p(x_t | \hat{z}_t^{(i)}, x_{t-1}^{(i)}) \quad (5)$$

2. Sequential Importance Sampling step. With conditioning on the new information and the Bayes rule, each particle is weighted by the likelihood of the observations in the updated state represented by that particle equation (6).

$$\hat{w}_t^{(i)} \leftarrow p(y_t | \hat{z}_t^{(i)}, \hat{x}_t^{(i)}) \quad (6)$$

3. Selection step. High-weight particles are replaced by several particles while low-weight particles tend to disappear.

PF algorithms approximate the true posterior belief state given observations  $y_{1:t}$  by a set of particles.

$$p(z_t, x_t | y_{1:t}) = \frac{1}{N} \sum_{i=1}^N w_t^{(i)} \delta_{(z_t, x_t)}(z_t^{(i)}, x_t^{(i)}) \quad (7)$$

Where  $w_t^{(i)}$  is the weight of a particle,  $z_t^{(i)}$  are the discrete modes,  $x_t^{(i)}$  are the continuous parameters and  $\delta_{(\cdot)}(\cdot)$  is Dirac delta function.

Rao-Blackwellized Particle Filtering is a PF variant which uses some of the analytical structure of the model.

If we know the values of the discrete modes  $z_t$ , it is possible to calculate the distribution of the continuous states  $x_t$ . According to the Rao-Blackwell theorem, this leads to estimators with less variance than those obtained using only pure Monte Carlo sampling. Thus, if we can generate particles of  $z$  and analytically evaluate the expectation of  $x$  given  $z$ , we will need less particles for a given accuracy. We can therefore combine a PF , which compute the distribution of the discrete modes, with a bank of Kalman filters, which compute the distribution of the continuous states.

[6] proposed a further improvement look-ahead RBPF. While evaluating the importance of weights for particles at time  $t$ , la-RBPF looks ahead one step to see the behavior of the measurements at

time  $t + 1$ . It then uses this information to compute better weights at time  $t$ . The basic sequential steps are Kalman prediction, Selection, Sequential Importance Sampling, and Kalman updating.

## 2.5 Experimental Tests

### 2.5.1 Phase I

An expert generates data for the machines using maintenance records, data sheets from suppliers, intuition etc. Once the data base has been created, this file is opened with *Power Constructor* to create the BN and then, this information is exported to *Hugin*. The join distribution for the whole network appears in this software. Figure 3 shows the Bayesian network.

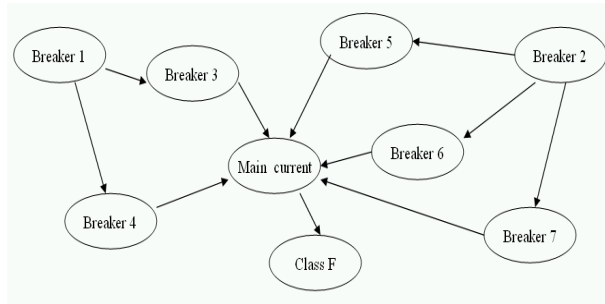


Figure 3: System's Bayesian network

When the BN is in *Hugin*, there is the possibility of making some inferences on certain variables which predict the behavior of the system as shown in Figure 4.

Figure 4 shows an evidence of 7.35 amperes. The subset of explanations is the following:

- Breaker 1 has a probability of 50.82% of being opened; therefore, machine 1 has a probability of 50.82% of being turned off. This breaker connects the power supply to the machine.
- Breaker 2 is closed; thus, machine 2 has a probability of 93.74% of being turned on.
- Breakers 3 and 4 are opened; therefore, machine 1 is turned off.
- Breaker 7 is opened.
- Breakers 6 and 5 are closed.
- Class F, the final node, indicates that circuits 2, 3 and 4 have a probability of being faulty. They have the maximum values of probability.

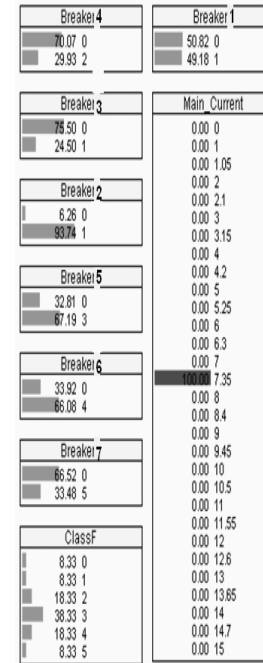


Figure 4: Evidence of 7.35 amperes of current

### 2.5.2 Phase II

For this phase, we tested the system with several runs and different number of particles. We have executed 20 runs without the presence of noise in the system, as well as 10 runs taking into account different values of noise in process and measurements. Finally, we obtained the mean ( $\mu$ ) and standard deviation ( $\sigma$ ) of the diagnosis error in all runs. The diagnosis error is the result of the correct estimated states divided by the total states.

## 3 Results

### 3.1 Discussion

Table 1 shows a summary of the system behavior for the la-RBPF algorithm using different number of particles and different values of noise.

Note that the mean and standard deviation of error diagnosis are smaller as the number of particle grows up.

Figure 5 shows the behavior for 50 particles with low level noise. Top graphs show the actual states and the measured variable of the process. Bottom plot shows the behavior of the la-RBPF in one run out of the ten runs with noise.

Noise		10 Particles		50 Particles	
p	m	$\mu$ %	$\sigma$ %	$\mu$ %	$\sigma$ %
0.0001	0.0001	4.08	8.69	0.45	1.00
0.01	0.01	16.66	4.41	12.53	2.67
0.01	0.1	23.65	3.47	20.49	2.38
0.1	0.1	30.49	0.98	30.75	0.54

Table 1: la-RBPF performance for different number of particles and noises; p is for process noise and m is for measurement noise

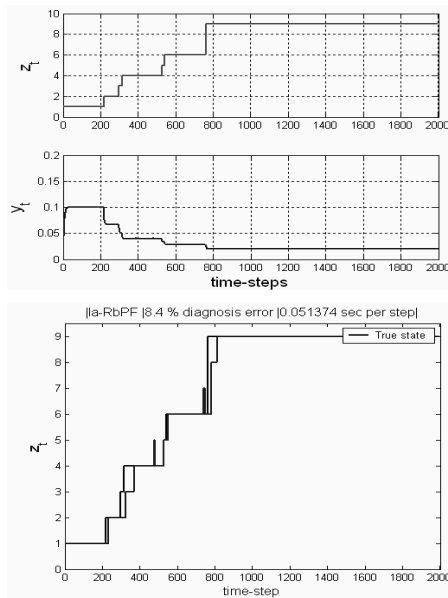


Figure 5: State estimation with 50 Particles and small level of noise

As we can see in table 1 in presence of noise the algorithm has low performance. We can cope with high noise level increasing the number of particle; however, high level noise kills la-RBPF features.

### 3.2 Conclusions

In this paper the application was focused on a simple model of a production line which have two machines. However, a real production line will have many circuits similar to the analyzed model. In the real model, which has many components that might fail, it is required to choose only those machines with greater probability to fail. This is done using BN. Then the PF technique is used to continuously monitor those machines to detect if a component has changed its parameters values. The BN is periodically updated based on evidence to have a reliable system.

### Acknowledgements

Authors would like to thank the support given by their campus to continue their studies as well as the dedication of their teachers to reach the goal.

### References

- [1] N de Freitas, R Dearden, F Hutter, R Morales-Menendez, J Mutch, and D Poole. *Diagnosis by a waiter and a Mars explorer*. IEEE, Special issue on Sequential State Estimation, 92(3), pp 455-468, March 2003.
- [2] Nando de Freitas. *Rao-Blackwellised Particle Filtering for Fault Diagnosis*. IEEE Aerospace conference 2001.
- [3] L Garza. *Hybrid Systems Fault Diagnosis with a Probabilistic Logic Reasoning Framework*. PhD. Thesis, 2001.
- [4] Janos J Gertler. *Fault detection and diagnosis in engineering systems*. Marcel Dekker, Inc., USA 1998.
- [5] C Andrieu and A Doucet. *Particle Filtering for partially observed gaussian state space models*, 2002.
- [6] R.Morales-Menéndez, N de Freitas, D. Poole *Real-time monitoring of processes using particle filters*, NIPS 15, pp 1457-1464, 2002 .
- [7] A Dubi. *Monte Carlo Applications in Systems Engineering*. Wiley, 1999.
- [8] S Russell and P Norvig. *Artificial Intelligence A Modern Approach*. Prentice Hall, 2<sup>nd</sup> edition, 2003
- [9] V Verma, G Gordon, R Simmons and S Thrun. *Particle Filters for Rover Fault Diagnosis*. To appear in Robotics & Automation Magazine, special issue on Human Centered Robotics and Dependability June 2004.
- [10] Judea Pearl *Probabilistic Reasoning in Intelligent Systems*. Networks of Plausible Inference, Morgan Kaufmann, 1988.