Particle Filters for Real-Time Fault Diagnosis in Hybrid Systems

M. H. REFAN [1], MAHDI BASHOOKI [2], SAREH BAHMANPOUR [2]
[1] S. Rajaee University, Lavizan, Tehran 16788, IRAN
[2] MAPNA Electrical and Control Company (MECO), Karaj, IRAN
refan@mapnaec.com, mehdibashooki@yahoo.com , sarehbahmanpour@hotmail.com

Abstract: - Embedded systems are composed of a large number of components that interact with the physical world via a set of sensors and actuators, have their own computational capabilities, and communicate with each other via a wired or wireless network. Diagnostic systems for such applications must address new challenges caused by the distribution of resources, the networking environment, and the tight coupling between the computational and physical worlds. Our approach is to move from centralized, discrete or continuous techniques toward a distributed, hybrid diagnosis architecture. Monitoring and diagnosis of any dynamical system depend crucially on the ability to estimate the system state given the observations. Estimation for hybrid systems is particularly challenging, because it requires keeping track of multiple models and the transitions between them. This paper presents a particle filtering based on estimation algorithm that addresses the challenge of the interaction between continuous and discrete dynamics in hybrid systems.

Keywords: - State estimation, Fault diagnosis, Hybrid systems, Particle filtering, JMLG model.

1 Introduction

In Embedded systems, the physical plant is composed of a large number of distributed nodes, each of which performs a moderate amount of computation, collaborates with other nodes via a wired or wireless network, and is embedded in the physical word via a set of sensors and actuators. Such systems can be best represented by hybrid models and present a number of interesting new challenges for diagnostic systems.

The diagnosis problem is to determine the current state of a system given a stream of observations of that system. In traditional model-based diagnosis systems such as Livingstione [1], diagnosis performes by maintaining a set of candidate hypotheses about the current state of the system, and using the model to predict the expected future state of the system given each candidate. The predicted states are then compared with the observations of what actually occurred. If the observations are consistent with a particular state that is predicted, that state is kept as a candidate hypothesis. If they are inconsistent, the candidate is discarded.

In the hybrid model, the task is to determine the best action to perform the given current estimate of actual state of the system. This estimate, referred to as the belief state, is exactly what we would like to determine in the diagnosis problem, and the problem of keeping the belief state update is well understood in the decision theory literature.

Unfortunately, maintaining an exact belief state is computationally intractable for the type of problem we are interested in. Since our model contains both discrete and continuous variables, the belief state is a set of multidimensional probability distributions over the continuous state variables, with one such distribution for each mode of the system. These distributions may not even be unimodal, so just representing the belief state is a complex problem, but updating it when new observations are made is intractable for hybrid models in all but the simplest model of models. Therefore, an approximation needs to be made.

A particle filter represents a probability distribution using a set of discrete samples, referred to as particles, each of which has an associated weight. The set of weighted particles constitutes an approximation to the belief state, and has the advantage over other approaches such as Kalman filters that represent arbitrary distributions. To update the distribution when a new observation is made, we treat each particle as a hypothesis about the state of the system, apply the model to it to move it to a new state, and multiply the weight of the particle by the likelihood of making the observation in that new state. To prevent a small number of particles from dominating the probability distribution, the particles are then resample, with new set of particles, each of weight one, being constructed by selecting samples randomly based on their weight from old set [2].

Particle filters have already proven very successful for a number of tasks, including visual tracking and robot navigation. An important thing to note is that standard particle filters treat the model essentially as a black box, using it only to predict future states of the system [3], [4].

In the next section, we will discuss the hybrid model, which we used to test Particle Filtering algorithm. In section 3, we will describe particle filtering and demonstrate its weaknesses when applied to diagnosis problems. Modifications to the standard particle filter in detail will be discussed in section 4. Finally, in section 5 and 6 we will present some preliminary results on experimental data, using a simple version of our proposed approach.

2 Hybrid Models

Many probabilistic time series models come from either Hidden Markov Models (HMMs) or stochastic linear dynamical systems commonly known as State-Space Models (SSMs). Using a single discrete random variable- the hidden state-hidden Markov models can represent the past information of a sequence. The prior probability distribution of this state can be calculated from the previous hidden state and stochastic transition matrix. If we know the state at any time, the past, present, and future observations become statistically independent, the Markov independence property.

Similarly, using a real-valued hidden state vector, state-space models can represent past information. Again, conditioned on this state vector, the past, present, and future observations are statistically independent. The dependency between the present state vector and the previous state vector is specified through the dynamic equations of the system and the noise model. A common case occurs when these equations are linear and the noise model is Gaussian. This model is also known as a linear dynamical system or Kalman filter model. HMMs and SSMs are well-known models; however, most real and interesting processes cannot be characterized by either purely discrete or purely linear-Gaussian dynamics.

Typical industrial processes may have multiples discrete modes of behavior, each of which has approximately linear dynamics. We are interested in dynamical systems which are characterized by a combination of discrete and continuous dynamics. Switching state-space models, or Jump Markov Linear Gaussian (JMLG) systems, are a natural generalization of hidden Markov models and state space models in which the dynamics can change in a discrete manner from one linear operating regime to another [5].

2.1 State-Space Model (SSM)

A state-space model defines a probability density over a time series of real-valued observation vectors by assuming that the observations were generated from a sequence of hidden state vectors. The hidden state vectors obey the Markov independence property. The joint probability for the sequences of states and observations can be represented as:

$$p(x_{1:T}, y_{1:T}) = p(x_1)p(y_1 \mid x_1)\prod_{t=2}^{T} p(x_t \mid x_{t-1})p(y_t \mid x_t) \qquad (1)$$

Figure 1, which is a Directed Acyclic Graph (DAG), shows the conditional independencies specified by equation (1). Each node is conditionally independent of its non-descendents given its parents. Shaded nodes represent observable variables and unshaded nodes represent hidden variables.

The state transition function is

$$x_{t+1} = Ax_t + Bw_{t+1} + Fu_{t+1} \qquad (2)$$

Where A is a state transition matrix, B is the noise state matrix, F is the input matrix, w_t is Gaussian, such as, $w_t \sim N(0, I)$ with covariance Q, and u_t is the input observation. The initial state is $x_0 \sim N(\mu_0, \Sigma_0)$. Equation (2) ensures that if $p(x_t)$ is Gaussian, then $p(x_{t+1})$ is Gaussian too. The output function is

$$y_t = Cx_t + D\nu_t + Gu_t \qquad (3)$$

Where C is the output matrix, D is the output noise matrix, ν_t is Gaussian, such as $\nu_t \sim N(0, I)$ with covariance R, and G is usually a null matrix for most applications.

$p(y_t \mid x_t)$ is also Gaussian, given by equation (4)

$$p(y_t \mid x_t) = (2\pi)^{-\frac{n_y}{2}} \mid R \mid^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(y_t - Cx_t - Gu_t)' R^{-1}(y_t - Cx_t - Gu_t)\right] \quad (4)$$

The problem of state estimation or inference for state space models consists of estimating the posterior probabilities of the hidden variables given a sequence of the observed variables. Assuming the local likelihood functions for the observations are Gaussian and the priors for the hidden states are Gaussian, the resulting posterior is also Gaussian [6].

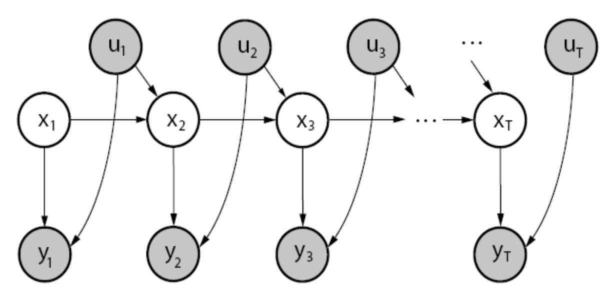

Figure1. Full state-space model.

2.2 Hidden Markov Model (HMM)

Like the state space model, the hidden Markov model defines probability distributions over sequences of observations, $y_{1:t}$. The distribution over sequences is obtained by specifying a distribution over observations at each time step t given a discrete hidden state z_t (as opposed to the continuous state in an SSM), together with the probability of transitioning from one hidden state to another. The joint probability for the sequences of states z_t and observations y_t can be factored as in equation (5)

$$p(z_{1:T}, y_{1:T}) = p(z_1)p(y_1 \mid z_1)\prod_{t=2}^{T} p(z_t \mid z_{t-1})p(y_t \mid z_t) \quad (5)$$

This equation obeys the Markov independence property. Figure 2 shows the conditional independencies specified by equation (5), where $z_0 \sim p(z_0)$. In the HMM framework, the state is represented by a single multinomial variable; this variable can take one of n_z discrete values, $z_t \in \{1,\dots,n_z\}$. The state transition probabilities are defined by $p(z_t \mid z_{t-1})$. If the observables are discrete symbols taking one of n_y values, the observation probabilities will be represented by $p(y_t \mid z_t)$. However, for a continuous observation vector, $p(y_t \mid z_t)$ can be modeled in many different forms, such as Gaussian, a mixture of Gaussian, etc.

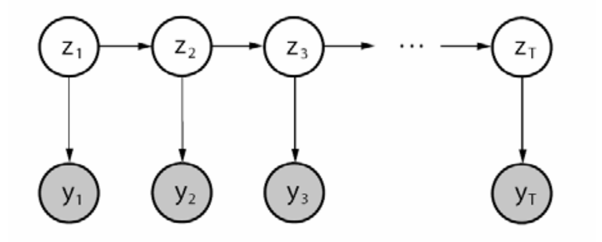

Figure 2. Hidden Markov model graph.

2.3 Hybrid Models

A natural way to improve both models is to combine them. Such combinations are known as hybrid models, state-space models with switching, or jump-linear systems. Basically, hybrid models combine the discrete transition structure of Hidden Markov models with the linear dynamics of state space models. A lot of work has been done using this idea in different domains.

We will work with the following hybrid model, the JMLG model. The dynamic behavior for the simplest case of this model was described by equations (2-3). We will generalize it here:

$$z_t \sim p(z_t \mid z_{t-1}) \quad (6)$$

$$x_t = A(z_t)x_{t-1} + B(z_t)w_t + F(z_t)u_t \quad (7)$$

$$y_t = C(z_t)x_t + D(z_t)v_t + G(z_t)u_t \quad (8)$$

Where $y_t \in R^{n_y}$ denotes the measurements, $x_t \in R^{n_x}$ denotes the unknown continuous states, $u_t \in U$ is a known input, and $z_t \in \{1,\dots,n_z\}$ denotes the unknown discrete modes. The noises are Gaussian: $w_t \sim N(0,I)$ and $v_t \sim N(0,I)$. Note that the parameters $(A(i),B(i),C(i),D(i),F(i),G(i))_{i=1}^{n_z}$ depend on the discrete mode. For each discrete mode, we have a single linear-Gaussian model and initial states are $x_0 \sim N(\mu_0,\Sigma_0)$ and $z_0 \sim p(z_0)$. We should ensure that for any quantity of i, $D(i)D(i)^T > 0$.

Assume that we have a JMLG model, a sequence of observations $y_{1:t}$ and control inputs $u_{1:t}$, and we want to have a real time estimation of the most likely hybrid state $\{z_t, x_t\}$ at each time t. Essentially, given the observations in the course of time, we are looking to find the discrete modes.

The inference task for any property of the discrete modes and continuous states relies on the joint probability distribution $p(x_{0:t}, z_{0:t} \mid y_{1:t}, u_{1:t})$. The goal of the analysis is to compute the marginal posterior distribution of the discrete modes $p(z_{0:t} \mid y_{1:t})$. This distribution can be derived from the posterior distribution, $p(x_{0:t}, z_{0:t} \mid y_{1:t})$ by standard marginalization. The posterior density satisfies the following recursion:

$$p(x_{0:t}, z_{0:t} \mid y_{1:t}) = p(x_{0:t-1}, z_{0:t-1} \mid y_{1:t-1}) \times \frac{p(y_t \mid x_t, z_t) p(x_t, z_t \mid x_{t-1}, z_{t-1})}{p(y_t \mid y_{1:t-1})} \quad (9)$$

This recursion involves intractable integrals in the denominator. Using numerical methods such as Particle Filtering technique can help us to approximate this integral.

3 Standard Particle Filtering

In the PF setting, we use a weighted set of samples (particles) $\left\{\left(x_{0:t}^{(i)}, z_{0:t}^{(i)}\right), w_t^{(i)}\right\}_{i=1}^{N}$ to approximate the posterior with the following point-mass distribution

$$\hat{P}_N(dx_{0:t}, z_{0:t} \mid y_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{x_{0:t}^{(i)}, z_{0:t}^{(i)}}(dx_{0:t}, z_{0:t}) \quad (10)$$

Where $\delta_{x_{0:t}^{(i)}, z_{0:t}^{(i)}}(dx_{0:t}, z_{0:t})$ denotes the Dirac-Delta function. Given N Particles $\left\{x_{0:t-1}^{(i)}, z_{0:t-1}^{(i)}\right\}_{i=1}^{N}$ at time t-1, approximately distributed according to $P(dx_{0:t-1}^{(i)}, z_{0:t-1}^{(i)} \mid y_{1:t-1})$. PF enables us to compute N particles $\left\{x_{0:t}^{(i)}, z_{0:t}^{(i)}\right\}_{i=1}^{N}$ approximately distributed according to $P(dx_{0:t}^{(i)}, z_{0:t}^{(i)} \mid y_{1:t})$, at time t. Since we cannot sample from the posterior directly, the PF update is accomplished by introducing an appropriate importance proposal distribution $Q(dx_{0:t}, z_{0:t})$ from which we can obtain samples [8].

Unfortunately, there are a number of difficulties in applying particle filters to diagnosis problems. In particular, the filter must have a particle in a particular state before the probability of that state can be evaluated. If a state has no particles in it, its probability of being the true state of the system is zero. This is a particular problem in diagnosis problems because the transition probabilities to fault states are typically very low, so particles are unlikely to end up in fault states during the Monte Carlo prediction step.

The simplest solution to this sampling problem is to increase the number of particles being used. Given the constraints imposed on on-board systems, this approach is probably unrealistic. An important point to note is that standard particle filters treat the model essentially as a black box, using it only to predict future states of the system. We have described one approach which exploits some of the analytical structure of the JMLG model. Basically, if we know the values of the discrete modes z_t, it is possible to compute the distribution of the continuous states x_t exactly. We can therefore combine a particle filter to compute the distribution of the discrete modes with a bank of Kalman filters to compute the distribution of the continuous states. That is, we approximate the posterior distribution with a recursive, stochastic mixture of Gaussian. This strategy is known as Rao-Blackwellization because it is related to the Rao-Blackwell formula [9].

4 Rao-Blackwellised Particle Filtering

By considering the factorization $p(x_{0:t}, z_{0:t} \mid y_{1:t}) = p(x_{0:t} \mid y_{1:t}, z_{0:t}) p(z_{0:t} \mid y_{1:t})$, it is possible to design more efficient PF algorithms.

The density $p(x_{0:t} \mid y_{1:t}, z_{0:t})$ is Gaussian and can be computed analytically if we know the marginal posterior density $p(z_{0:t} \mid y_{1:t})$. This density satisfies the alternative recursion:

$$p(z_{0:t} \mid y_{1:t}) = p(z_{0:t-1} \mid y_{1:t-1}) \frac{p(y_t \mid y_{1:t-1}, z_{0:t}) p(z_t \mid z_{t-1})}{p(y_t \mid y_{1:t-1})} \quad (11)$$

If equation (9) does not admit a closed-form expression, then equation (11) does not admit one either and sampling-based methods are still required. Also note that the term $p(y_t \mid y_{1:t-1}, z_{0:t})$ in equation (11) does not simplify to $p(y_t \mid z_t)$ because there is a dependency on past values through $x_{0:t}$. Now, assume that we can use a weighted set of samples $\left\{z_{0:t}^{(i)}, w_t^{(i)}\right\}_{i=1}^{N}$ to represent the marginal posterior distribution

$$\hat{P}_N(z_{0:t} \mid y_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{x_{0:t}^{(i)}}(z_{0:t}) \quad (12)$$

The marginal density of $x_{0:t}$ is a Gaussian mixture

$$\hat{P}_N(x_{0:t} \mid y_{1:t}) = \int p(x_{0:t} \mid z_{0:t}, y_{1:t}) dP(z_{0:t} \mid y_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} p(x_{0:t} \mid y_{1:t}, z_{0:t}^{(i)}) \quad (13)$$

That can be computed efficiently with a stochastic bank of Kalman filters. That is, we use PF to estimate the distribution of z_t and exact computations to estimate the mean and variance of z_t. This is a basis of the RBPF algorithm that was adopted in [4].

5 Results

We tested two inference algorithms for JMLG model, which described in equations (6) to (8) using experimental data. These simulations, were done for three and ten discrete modes using N=100/500, and T=50. The results are shown in the following Figures.

Figures 3 and 4 plot the tracking errors for N=100, which used three and ten discrete modes respectively. As it can be seen, when the number of discrete modes grows up, the RBPF algorithm shows better result to estimate the state of the experimental system.

Figures 5 and 6 are the simulation result for N=500 with $n_z = 3$ and $n_z = 10$ respectively.

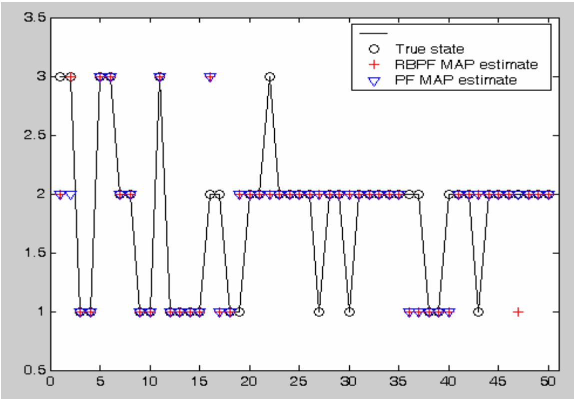

Figure 3. Experimental simulation (N=100; $n_z = 3$)

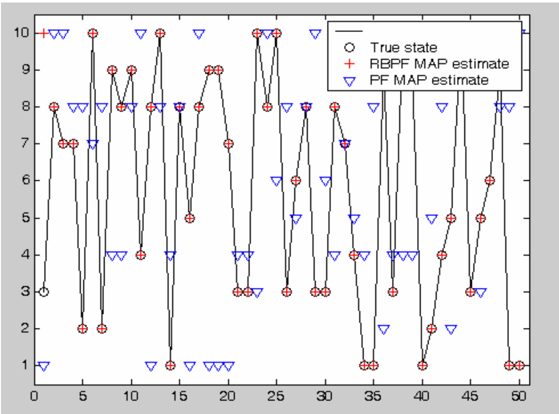

Figure 4. Experimental simulation (N=100; $n_z = 10$)

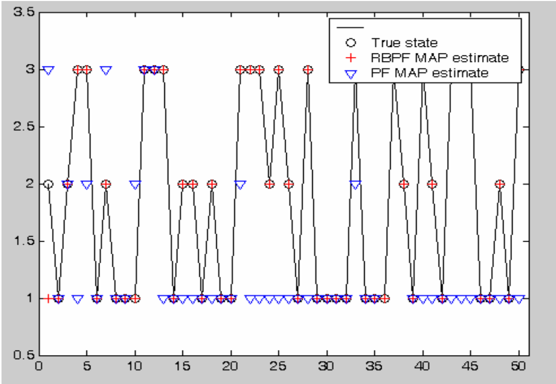

Figure 5. Experimental simulation (N=500; $n_z = 3$)

Graphs illustrate that increasing in the number of particles will rich the result. But it increase computational time, and in some real-time purposes it could cost for a rapid processor.

The probability distribution of RBPF algorithm for N=100 and N=400 are shown in Figure 7 and 8 respectively.

6 Conclusions

Results show that the RBPF algorithm gives a very low diagnosis error per number of particles. It works significantly better than standard PF. RBPF also gives a very low diagnosis error per unit of computing time, despite its greater computational expense per particle compared with standard PF. Faulty conditions usually have very low probabilities. Standard numerical approximations have trouble with this situation because a very small number of particles are assigned to a faulty discrete mode, despite the observations. However, RBPF samples the possible discrete modes from their true posterior distribution, capturing evidence of faulty conditions and allowing them to be identified. RBPF also gives lower variance than standard PF per number of particles. This advantage, based on the Rao-Blackwell formula, grows as the number of particle is increased.

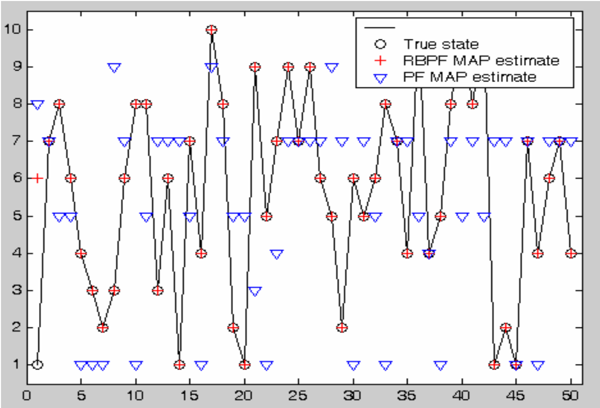

Figure 6. . Experimental simulation (N=500; $n_z = 10$)

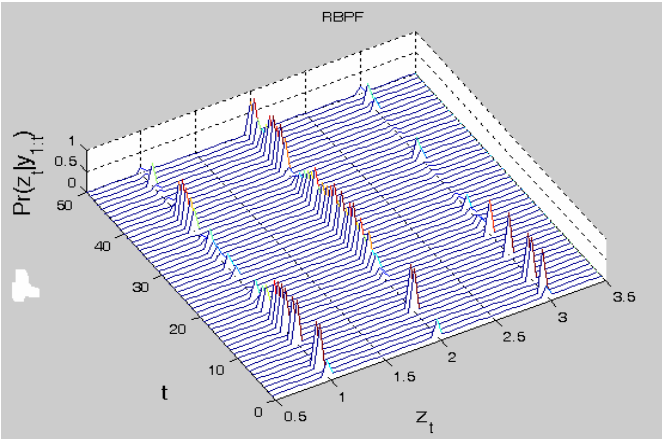

Figure 7. Probability distribution For RBPF algorithm N=100

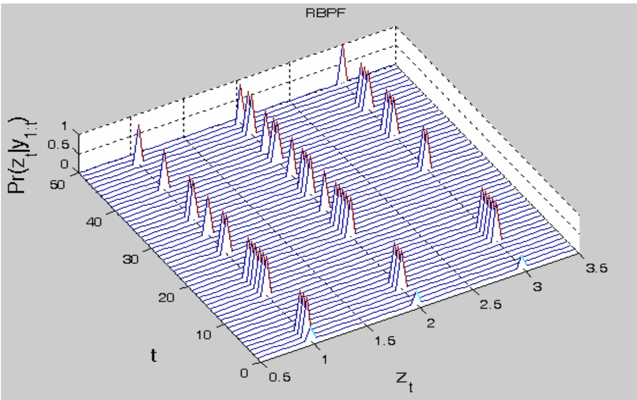

Figure 8. Probability distribution For RBPF algorithm N=500

References

[1] Brian C. Williams and P. padndurang Nayak, "A model-based approach to reactive self-configuring systems", inproceedings of the thirteenth National Conference on Artificial Intelligence and Eight Innovative Applications of Artificial Intelligence conference, pp. 971-978, port-land, oregan, 1996. AAAI Press/ The MIT Press.

[2] M. S. Grewal and A. P. Andrews, kalman filtering: Theory and Practice, Prentice Hall, 1993.

[3] S.Bahmanpour, M. Bashooki, M. H. Refan "State Estimation and Fault Diagnosis of Industrial Process by Using of Particle Filters"6th WSEAS International Conference on SIGNAL PROCESSING, ROBOTICS and AUTOMATION (ISPRA '07) 16-19 Feb 2007

[4] S. Bahmanpour, M. Bashooki, M. H. Refan "Real-Time Monitoring and Diagnosis in Dynamic Systems Using Particle Filtering Methods" WSEAS Transaction on signal processing , Issue 2, Vol. 3, Feb 2007. ISSN:1790-5022.

[5] A Doucet, N Freitas, and N J Gordon, editors. Sequential Monte Carlo Methods in Practice. Springer-Verlag, 2001.

[6] Z Ghahramani and G. E. Hinton. "Variational learning for switching state space model", Neural Computation, (12) 4: 963-996, 1998

[7] R. J. Elliot , L. Aggoun, and J. B. Moore. "Hidden Markov Models estimation and control". Springer-Verlag, New York, 1995.

[8] G Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. Journal of Computational and Graphical Statistics, 5: 1-25, 1996.

[9] R M Menendez. Real-time monitoring and diagnosis in dynamic systems using particle filtering methods. Chapters 3, 4 and 5. PHD Thesis in Intelligent systems, Monterry Institute, Mexico, 2003.