

---

# Real-time monitoring of complex industrial processes with particle filters

---

**Rubén Morales-Menéndez\***

Dept. of Mechatronics and Automation  
ITESM campus Monterrey  
Monterrey, NL México  
rmm@itesm.mx

**Nando de Freitas and David Poole**

Dept. of Computer Science  
University of British Columbia  
Vancouver, BC, V6T 1Z4, Canada  
{nando,poole}@cs.ubc.ca

## Abstract

This paper discusses the application of particle filtering algorithms to fault diagnosis in complex industrial processes. We consider three ubiquitous processes: an industrial dryer, a level tank, and a continuous stirred tank reactor. For these applications, we compared three particle filtering variants: standard particle filtering, Rao-Blackwellised particle filtering and a version of Rao-Blackwellised particle filtering that does one-step look-ahead to select good sampling regions. We show that the overhead of the extra processing per particle of the more sophisticated methods is more than compensated by the decrease in error and variance.

## 1 Introduction

Real-time monitoring is important in many areas such as robot navigation or diagnosis of complex systems [1, 2]. This paper considers online monitoring of complex industrial processes. The processes have a number of discrete states, corresponding to different combinations of faults. The dynamics can be very different based on the discrete states. Even if there are very few discrete states, exact monitoring is computationally unfeasible as the state of the system depends on the history of the discrete states. However there is a need to monitor these systems in real time to determine what faults could have occurred.

This paper investigates the feasibility of using particle filtering (PF) for online monitoring. It also proposes some powerful variants of PF. These variants involve doing more computation per particle for each time step. We wanted to investigate whether we could do real-time monitoring and whether the extra cost of the more sophisticated methods was worthwhile in these real-world domains.

## 2 Processes monitored

We analyzed three industrial processes: an *industrial dryer*, a *level-tank* and a *stirred tank reactor*<sup>1</sup>. In each of these, we physically inserted a sequence of faults into the system

---

\* Visiting Scholar (2000-2003) at The University of British Columbia.

<sup>1</sup>For the stirred tank reactor, we used a sophisticated nonlinear model of the system rather than the physical system.

and made appropriate measurements. The nonlinear models that we used in the stochastic simulation were obtained through open-loop step responses for each discrete state [3]. The parametric identification procedure was guided by the minimum squares error algorithm [4] and validated with the “Control Station” software [5]. The discrete-time state space representation was obtained by a standard procedure in control engineering [3].

## 2.1 Industrial dryer

An industrial dryer is a thermal process that converts electricity to heat. As shown in Figure 1, we measure the temperature of the output air-flow. Normal operation corresponds

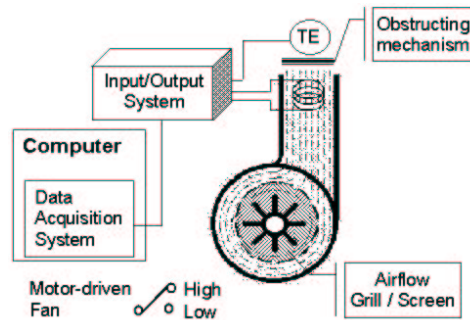


Figure 1: Industrial dryer.

to low fan speed, open air-flow grill and clean temperature sensor (we denote this discrete state  $z_t = 1$ ). We induced 3 types of fault:  $z_t = 2$  faulty fan,  $z_t = 3$  faulty grill (the grill is closed), and  $z_t = 4$  faulty fan and grill.

## 2.2 Level tank

Many continuous industrial processes need to control the amount of accumulated material using level measurement, such as evaporators, distillation columns or boilers. We worked with a level-tank system that exhibits the dynamic behaviour of these important processes, see Figure 2. For simplicity, we are not showing all the instruments and control devices in the diagram. A by-pass pipe and two manual valves ( $V_1$  and  $V_2$ ) were used to induce typical faulty states.

## 2.3 Continuous stirred tank reactor

A Continuous Stirred Tank Reactor (CSTR) is a complex nonlinear, multivariable system; see Figure 2. It involves a second-order exothermic reaction ( $2A \rightarrow B$ ), where 2 components of  $A$  react irreversibly and at specific reaction rate  $k$  to form product  $B$  [6]. The reaction rate constant  $k$  follows the Arrhenius equation, which shows that the effect of temperature is usually exponential. The overall reaction rate is  $\mathcal{R} = kC_A^2$ , where  $C_A$  denotes the concentration of  $A$ . We induced typical faults (dirty surfaces and sedimentation) by modifying the heat transfer coefficient in the differential equation model; see [1] for a similar scenario.

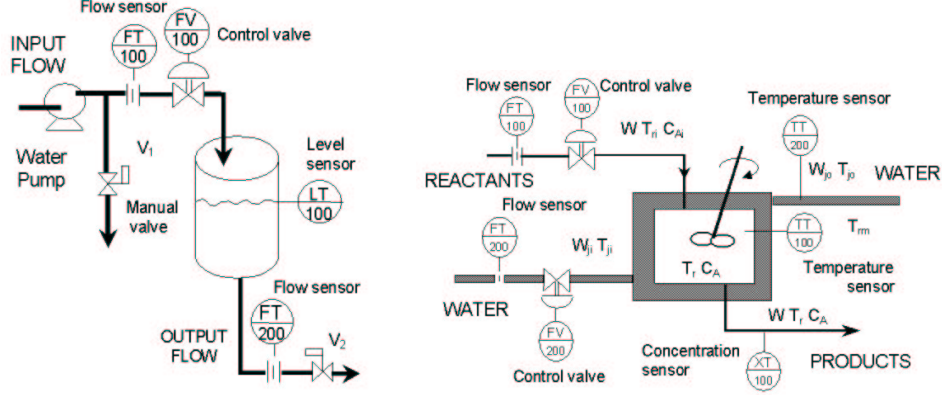


Figure 2: Level-tank and Nonlinear continuous stirred tank reactor (CSTR).

### 3 Mathematical model

We adopted the following jump Markov linear Gaussian model:

$$\begin{aligned}
 z_t &\sim P(z_t|z_{t-1}) \\
 x_t &= A(z_t)x_{t-1} + B(z_t)w_t + F(z_t)u_t \\
 y_t &= C(z_t)x_t + D(z_t)v_t + G(z_t)u_t,
 \end{aligned}$$

where  $y_t \in \mathbb{R}^{n_y}$  denotes the measurements,  $x_t \in \mathbb{R}^{n_x}$  denotes the unknown continuous states,  $u_t \in \mathcal{U}$  is a known control signal,  $z_t \in \{1, \dots, n_z\}$  denotes the unknown discrete states (normal operation and faulty conditions). The noise processes are *i.i.d* Gaussian:  $w_t \sim \mathcal{N}(0, I)$  and  $v_t \sim \mathcal{N}(0, I)$ . The parameters  $(A, B, C, D, F, G, P(z_t|z_{t-1}))$  are identified matrices with  $D(z_t)D(z_t)^T > 0$  for any  $z_t$ . The initial states are  $x_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$  and  $z_0 \sim P(z_0)$ . The important thing to notice is that for each realization of  $z_t$ , we have a single linear-Gaussian model. If we knew  $z_t$ , we could solve for  $x_t$  exactly using the Kalman filter algorithm.

The aim of the analysis is to compute the marginal posterior distribution<sup>2</sup> of the discrete states  $P(z_{0:t}|y_{1:t})$ . This distribution can be derived from the posterior distribution  $P(dx_{0:t}, z_{0:t}|y_{1:t})$  by standard marginalisation. The posterior density satisfies the following recursion

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

This recursion involves intractable integrals. One, therefore, has to resort to some form of numerical approximation scheme.

<sup>2</sup>NOTATION: For a generic vector  $\theta$ , we adopt the notation  $\theta_{1:t} \triangleq (\theta_1, \theta_2, \dots, \theta_t)'$  to denote all the entries of this vector at time  $t$ . For simplicity, we use  $\theta_t$  to denote both the random variable and its realisation. Consequently, we express continuous probability distributions using  $P(d\theta_t)$  instead of  $\Pr(\theta_t \in d\theta_t)$  and discrete distributions using  $P(\theta_t)$  instead of  $\Pr(\theta_t = \theta_t)$ . If these distributions admit densities with respect to an underlying measure  $\mu$  (counting or Lebesgue), we denote these densities by  $p(\theta_t)$ . For example, when considering the space  $\mathbb{R}^n$ , we will use the Lebesgue measure,  $\mu = d\theta_t$ , so that  $P(d\theta_t) = p(\theta_t) d\theta_t$ .

## 4 Particle filtering

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

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

where  $\delta_{x_{0:t}, z_{0:t}}^{(i)}(dx_{0:t}, z_{0:t})$  denotes the Dirac-delta function. Given  $N$  particles  $\{x_{0:t-1}^{(i)}, z_{0:t-1}^{(i)}\}_{i=1}^N$  at time  $t-1$ , approximately distributed according to  $P(dx_{0:t-1}, z_{0:t-1}|y_{1:t-1})$ , PF enables us to compute  $N$  particles  $\{x_{0:t}^{(i)}, z_{0:t}^{(i)}\}_{i=1}^N$  approximately distributed according to  $P(dx_{0:t}, z_{0:t}|y_{1:t})$ , 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. The basic algorithm, Figure (3), consists of two steps: sequential importance sampling and selection (see [7] for a detailed derivation). This algorithm uses the transition priors as proposal distributions;  $Q(x_{0:t}, z_{0:t}|y_{1:t}) = P(x_t|x_{t-1}, z_t)P(z_t|z_{t-1})$ . For the selection step, we used a state-of-the-art minimum variance resampling algorithm [8].

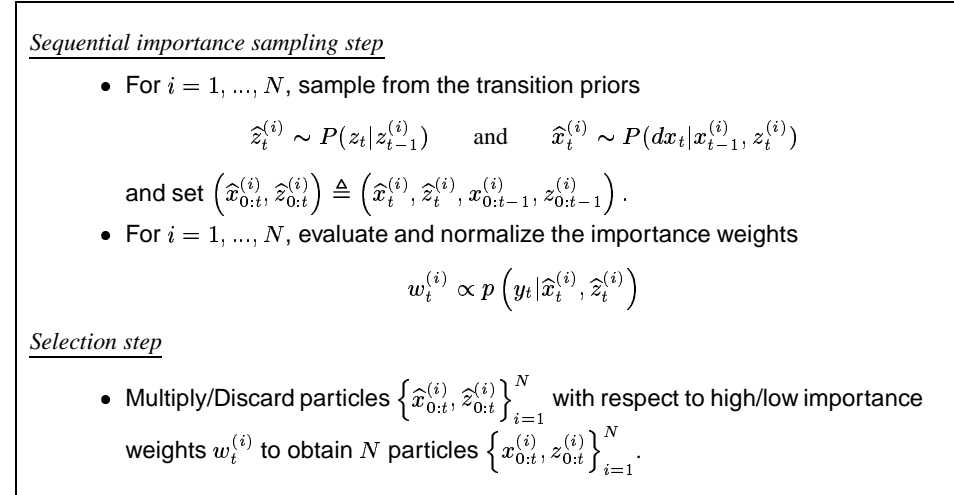


Figure 3: PF algorithm at time  $t$ .

## 5 Improved Rao-Blackwellised particle filtering

By considering the factorisation  $p(x_{0:t}, z_{0:t}|y_{1:t}) = p(x_{0:t}|y_{1:t}, z_{0:t})p(z_{0:t}|y_{1:t})$ , it is possible to design more efficient algorithms. *The density  $p(x_{0:t}|y_{1:t}, z_{0:t})$  is Gaussian and can be computed analytically if we know the marginal posterior density  $p(z_{0:t}|y_{1:t})$ .* This density satisfies the alternative recursion

$$p(z_{0:t}|y_{1:t}) = p(z_{0:t-1}|y_{1:t-1}) \frac{p(y_t|y_{1:t-1}, z_{0:t})p(z_t|z_{t-1})}{p(y_t|y_{1:t-1})} \quad (2)$$

If equation (1) does not admit a closed-form expression, then equation (2) does not admit one either and sampling-based methods are still required. (Also note that the term

$p(y_t|y_{1:t-1}, z_{0:t})$  in equation (2) does not simplify to  $p(y_t|z_t)$  because there is a dependency on past values through  $x_{0:t}$ . Now assuming that we can use a weighted set of samples  $\{z_{0:t}^{(i)}, w_t^{(i)}\}_{i=1}^N$  to represent the marginal posterior distribution

$$\widehat{P}_N(z_{0:t}|y_{1:t}) = \sum_{i=1}^N w_t^{(i)} \delta_{z_{0:t}^{(i)}}(z_{0:t}),$$

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

$$\widehat{p}_N(x_{0:t}|y_{1:t}) = \int p(x_{0:t}|z_{0:t}, y_{1:t}) dP(z_{0:t}|y_{1:t}) = \sum_{i=1}^N w_t^{(i)} p(x_{0:t}|y_{1:t}, z_{0:t}^{(i)})$$

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 (Kalman filter) to estimate the mean and variance of  $z_t$ . In particular, we sample  $z_t^{(i)}$  and then propagate the mean  $\mu_t^{(i)}$  and covariance  $\Sigma_t^{(i)}$  of  $x_t$  with a Kalman filter:

$$\begin{aligned} \mu_{t|t-1}^{(i)} &= A(z_t^{(i)})\mu_{t-1}^{(i)} + F(z_t^{(i)})u_t \\ \Sigma_{t|t-1}^{(i)} &= A(z_t^{(i)})\Sigma_{t-1}^{(i)}A(z_t^{(i)})^\top + B(z_t^{(i)})B(z_t^{(i)})^\top \\ S_t^{(i)} &= C(z_t^{(i)})\Sigma_{t|t-1}^{(i)}C(z_t^{(i)})^\top + D(z_t^{(i)})D(z_t^{(i)})^\top \\ y_{t|t-1}^{(i)} &= C(z_t^{(i)})\mu_{t|t-1}^{(i)} + G(z_t^{(i)})u_t \\ \mu_t^{(i)} &= \mu_{t|t-1}^{(i)} + \Sigma_{t|t-1}^{(i)}C(z_t^{(i)})^\top S_t^{-1(i)}(y_t - y_{t|t-1}^{(i)}) \\ \Sigma_t^{(i)} &= \Sigma_{t|t-1}^{(i)} - \Sigma_{t|t-1}^{(i)}C(z_t^{(i)})^\top S_t^{-1(i)}C(z_t^{(i)})\Sigma_{t|t-1}^{(i)}, \end{aligned}$$

where  $\mu_{t|t-1} \triangleq \mathbb{E}(x_t|y_{1:t-1})$ ,  $\mu_t \triangleq \mathbb{E}(x_t|y_{1:t})$ ,  $y_{t|t-1} \triangleq \mathbb{E}(y_t|y_{1:t-1})$ ,  $\Sigma_{t|t-1} \triangleq \text{cov}(x_t|y_{1:t-1})$ ,  $\Sigma_t \triangleq \text{cov}(x_t|y_{1:t})$  and  $S_t \triangleq \text{cov}(y_t|y_{1:t-1})$ .

This is the basis of the RBPF algorithm that was adopted in [9]. Here, we introduce an extra improvement. Let us expand the expression for the importance weights:

$$w_t = \frac{p(z_{0:t}|y_{1:t})}{q(z_{0:t}|y_{1:t})} = \frac{p(z_{0:t-1}|y_{1:t})}{p(z_{0:t-1}|y_{1:t-1})} \frac{p(z_t|z_{0:t-1}, y_{1:t})}{q(z_t|z_{0:t-1}, y_{1:t})} \quad (3)$$

$$\propto \frac{p(y_t|y_{1:t-1}, z_{0:t})}{q(z_t|z_{0:t-1}, y_{1:t})}. \quad (4)$$

The proposal choice,  $q(z_{0:t}|y_{1:t}) = q(z_t|z_{0:t-1}, y_{1:t})p(z_{0:t-1}|y_{1:t-1})$ , states that we are not sampling past trajectories. Sampling past trajectories requires solving an intractable integral [10].

We could use the transition prior as proposal distribution:  $q(z_t|z_{0:t-1}, y_{1:t}) = p(z_t|z_{0:t-1}, y_{1:t-1}) = p(z_t|z_{t-1})$ . Then, according to equation (4), the importance weights simplify to the predictive density

$$w_t \propto p(y_t|y_{1:t-1}, z_{0:t}) = \mathcal{N}(y_t; y_{t|t-1}, S_t). \quad (5)$$

However, we can do better by noticing that according to equation (3), the optimal proposal distribution corresponds to the choice  $q(z_t|z_{0:t-1}, y_{1:t}) = p(z_t|z_{0:t-1}, y_{1:t})$ . This distribution satisfies Bayes rule:

$$p(z_t|z_{0:t-1}, y_{1:t}) = \frac{p(y_t|y_{1:t-1}, z_{0:t})p(z_t|z_{0:t-1}, y_{1:t-1})}{p(y_t|y_{1:t-1}, z_{0:t-1})} \quad (6)$$

and, hence, the importance weights simplify to

$$w_t \propto p(y_t|y_{1:t-1}, z_{0:t-1}) = \sum_{z_t=1}^{n_z} p(y_t|y_{1:t-1}, z_{0:t-1}, z_t) p(z_t|z_{t-1}) \quad (7)$$

When the number of discrete states is small, say 10 or 100, we can compute the distributions in equations (6) and (7) analytically. This is the second major improvement over standard particle filters. Yet, a further improvement can still be attained.

Even when using the optimal importance distribution, there is a discrepancy arising from the ratio  $p(z_{0:t-1}|y_{1:t})/p(z_{0:t-1}|y_{1:t-1})$  in equation (3). This discrepancy is what causes the well known problem of sample impoverishment in all particle filters [7, 11]. To circumvent it to a significant extent, we note that the importance weights do not depend on  $z_t$  (we are marginalising over this variable). *It is therefore possible to select particles before the sampling step. That is, one chooses the fittest particles at time  $t - 1$  using the information at time  $t$ .* This observation leads to an efficient algorithm (RBPF2), whose pseudocode is shown in Figure 4. Note that for standard PF, Figure 3, the importance weights depend on the sample  $z_t^{(i)}$ , thus not permitting selection before sampling. Selecting particles before sampling results in a richer sample set at the end of each time step.

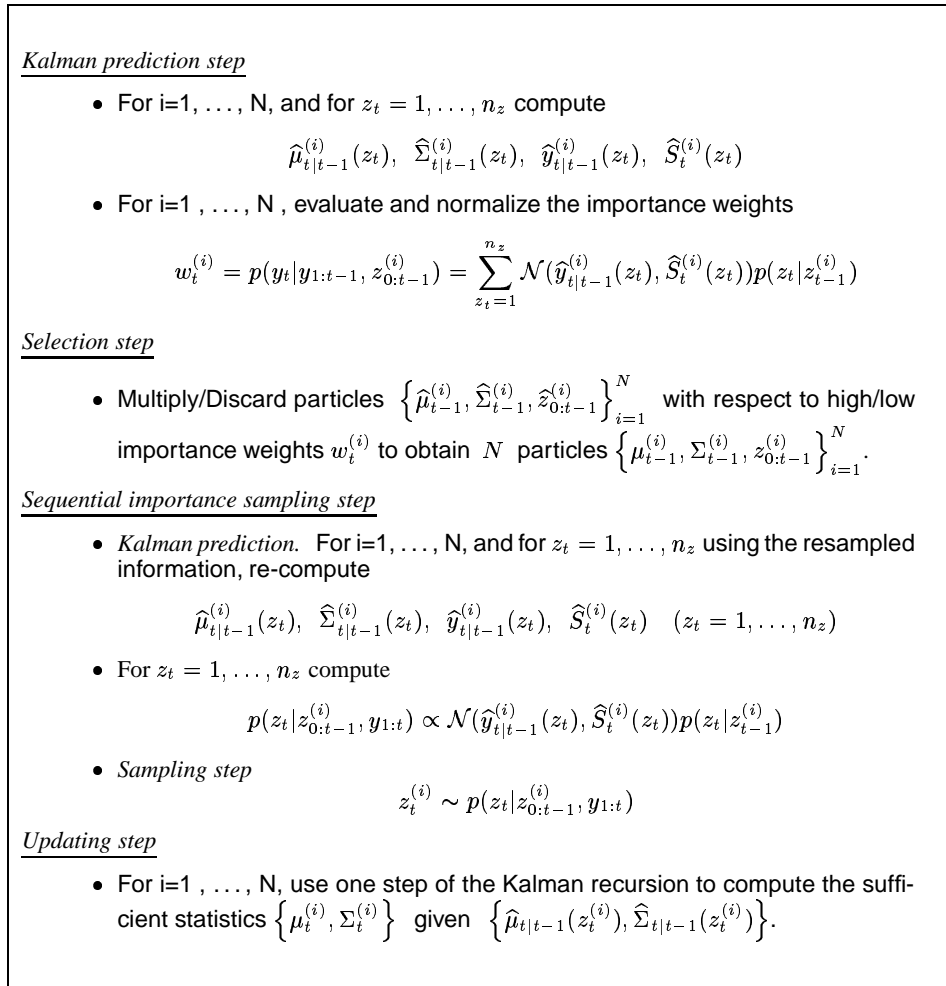


Figure 4: RBPF2 algorithm at time  $t$ . The algorithm uses an optimal proposal distribution. It also selects particles from time  $t - 1$  using the information at time  $t$ .

## 6 Results

The results are shown in Figures 5 to 7. The left graphs show error detection versus computing time per time-step (the signal sampling time was 1 second). The right graphs show the error detection versus number of particles. The error detection represents how many discrete states were not identified properly, and was calculated for 25 independent runs (1,000 time steps each). The graphs show that RBPF2 works significantly better (low error rate and very low variance). This is essential for real-time operation with low error rates.

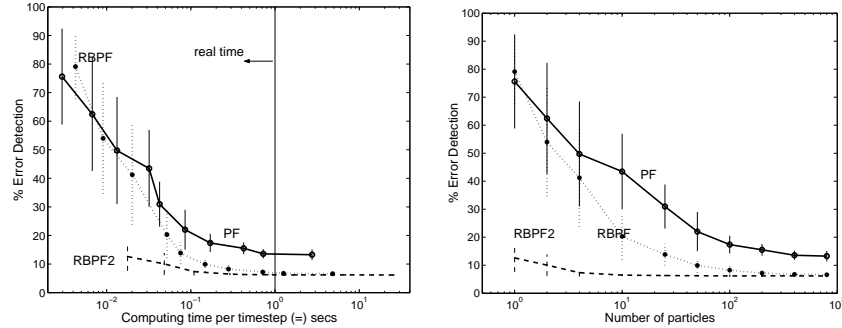


Figure 5: Industrial dryer: error detection vs computing time and number of particles.

The graphs also show that even for 1 particle, RBPF2 is able to track the discrete state. The reason for this is that the sensors are very accurate (variance=0.01). Consequently, the distributions are very peaked and we are simply tracking the mode. Note that RBPF2 is the only filter that uses the most recent information in the proposal distribution. Since the measurements are very accurate, it finds the mode easily. We repeated the level-tank experiments with noisier sensors (variance=0.08) and obtained the results shown in Figure 8. Noisier sensors, as expected, reduce the accuracy of RBPF2 with a small number of particles. However, it is still possible to achieve low error rates in real-time. Since modern industrial and robotic sensors tend to be very accurate, we conclude that RBPF2 has great potential.

## References

- [1] J Chen and J Howell. A self-validating control system based approach to plant fault detection and diagnosis. *Computers and Chemical Engineering*, 25:337–358, 2001.
- [2] S Thrun, J Langford, and V Verma. Risk sensitive particle filters. In S Becker T. G Dietterich and Z Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.
- [3] K Ogata. *Discrete-Time Control Systems*. Prentice Hall, second edition, 1995.
- [4] L Ljung. *System Identification: Theory for the User*. Prentice-Hall, 1987.
- [5] D Cooper. *Control Station*. University of Connecticut, third edition, 2001.
- [6] W L Luyben. *Process Modeling, Simulation and Control for Chemical Engineers*. Mc Graw-Hill Publishing Company, second edition, 1989.
- [7] A Doucet, N de Freitas, and N J Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.
- [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.

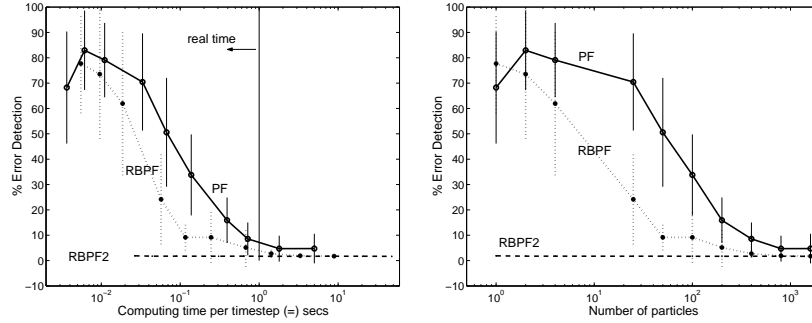


Figure 6: Level-tank (accurate sensors): error detection vs computing time and number of particles.

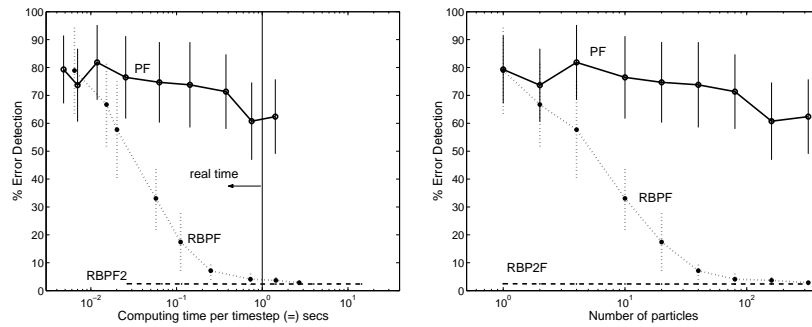


Figure 7: CSTR: error detection vs computing time and number of particles.

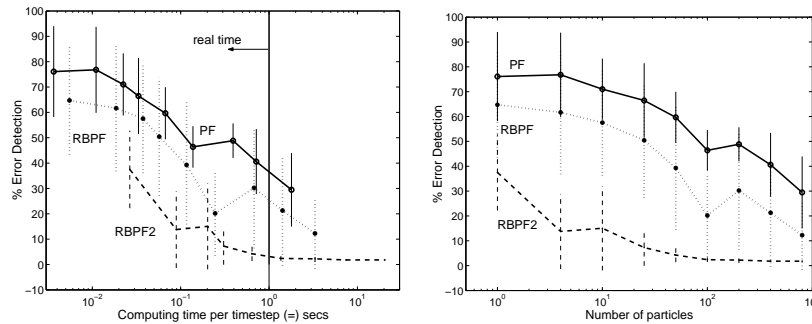


Figure 8: Level-tank (noisy sensors): error detection vs computing time and number of particles.

[9] N de Freitas. Rao-Blackwellised particle filtering for fault diagnosis. In *IEEE Aerospace Conference*, 2001.

[10] C Andrieu, A Doucet, and E Punsakaya. Sequential Monte Carlo methods for optimal filtering. In A Doucet, N de Freitas, and N J Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[11] M K Pitt and N Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999.