

# ESTIMATION AND CONTROL OF INDUSTRIAL PROCESSES WITH PARTICLE FILTERS

Rubén Morales-Menéndez\*

Nando de Freitas†

David Poole‡

We present a probabilistic approach to state estimation and control of industrial processes. In particular, we adopt a jump Markov linear Gaussian (JMLG) model to describe an industrial heat exchanger. The parameters of this model are identified with the expectation maximisation (EM) algorithm. After identification, particle filtering algorithms are adopted to diagnose, in real-time, the state of operation of the heat exchanger. The particle filtering estimates are then used to drive an automatic control system.

**Keyword:** State Estimation, Control, Particle Filtering, Jump Markov Linear Gaussian Systems

## 1. Introduction

State estimation plays a critical role in modern diagnosis and control systems. Early detection of changes in the states of industrial process can be used to plan maintenance or to choose a suitable control policy. These changes are typically very subtle. They depend on operating conditions and on complex interactions of many discrete and continuous variables. It is often difficult for a human operator to evaluate or diagnose the process continuously [1, 2].

Here, we propose and real-time, automatic strategy for estimating the states of industrial processes from noisy measurements of continuous variables. This approach enables us to reduce the cognitive load experienced by human operators. It also serves to minimise the number of instruments and to open up room for sophisticated control strategies.

In particular, we adopt a jump Markov linear Gaussian (JMLG) model to describe an industrial heat exchanger with different linear regimes of operation. A discrete state variable controls the switching between the various linear regimes. The parameters of each regime are identified off-line with the EM algorithm [3]. Once the stationary parameters have been identified, real-time Rao-Blackwellised particle filtering (RBPF) algorithms are used to estimate the continuous and discrete states of the system on-line [4, 5]. These estimates are used to determine to control policy of a PID controller.

The paper is organised as follows. Section 2. presents and overview of our approach. Section 3. describes various particle filtering algorithms. Section 4. discusses the results and Section 5. concludes the paper.

## 2. Overview

We represent a complex nonlinear process (a heat exchanger) with a dynamic mixture of linear processes. In addition to the continuous state variables corresponding to each linear process, we have a discrete state variable that determines the linear regime of operation. We acquire data for each regime separately. This data enables us to do off-line identification with the EM algorithm. Subsequently, a particle filter uses these parameters and new measurements to estimate the discrete state of operation on-line. Knowledge of this state enables us to choose appropriate control strategies.

---

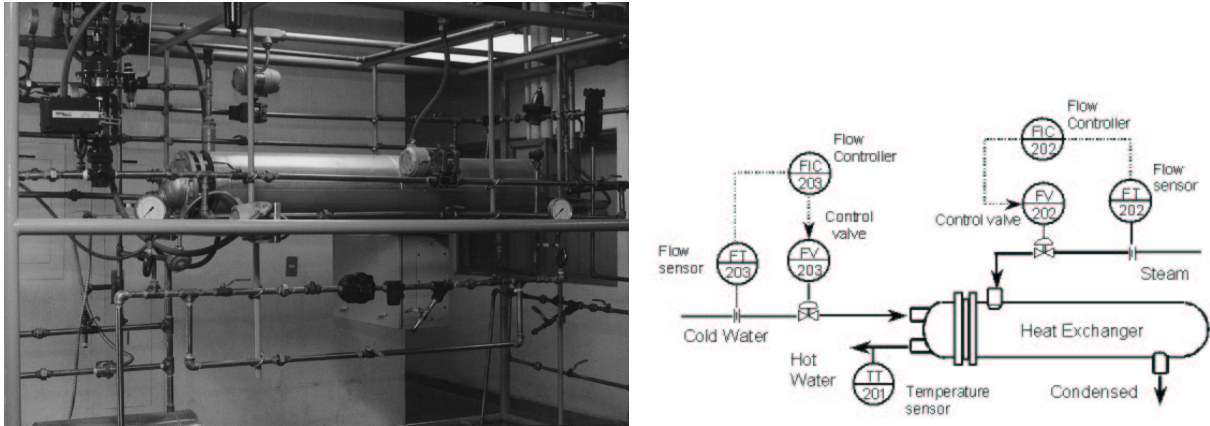
\* Mechatronics and Automation Dept, ITESM Mty, México , rmm@itesm.mx

† Computer Science Dept, University of British Columbia, Canada, nando@cs.ubc.ca

‡ Computer Science Dept, University of British Columbia, Canada, poole@cs.ubc.ca

## 2.1. Process monitored

We monitored an industrial heat exchanger, shown on the left of Figure 1. This exchanger heats 10 gpm of water from  $25^{\circ}\text{C}$  to  $70^{\circ}\text{C}$  using steam at  $5\text{ kg/cm}^2$ . It is in a horizontal arrangement with 11 Cu-Ni(90-10) tubes of  $3/4$  inches external diameter, which are in a BWG-20 triangular distribution. The container is 6 inches in external diameter and 70 inches in length. This process is fully instrumented and is operated by the Honeywell TDC 3000 industrial distributed control system [6]. The right graph in Figure 1 shows a conceptual diagram of the main instrumentation. Table 1 describes each tag-name.



**Figure 1.** Heat exchanger and instrumentation diagram.

**Table 1.** Heat exchanger instrumentation.

Tag-name	Name	Description	units
FT203	Flow transmitter	Input water flow	%
FV203	Control valve	Input water valve	%
FIC203	Flow controller	Input water flow PID	
FT202	Flow transmitter	Steam flow	%
FV202	Control valve	Steam valve	%
FIC202	Flow controller	Steam flow PID	
TT201	Temperature transmitter	Output water temperature	$^{\circ}\text{C}$

The key variable in this thermal process is the output water temperature. We can control this temperature by manipulating the steam flow. The dynamic characteristics (transient response) of the heat exchange are strongly influenced by the input water flow. The relationship among input water flow and output water temperature is nonlinear. We found five linear operating ranges shown in Table 2.

**Table 2.** Linear operating conditions.

Model	Name	FT203 Range	FV202	TT201
1	Very high flow	59-71 %	32 %	$39.69^{\circ}\text{C}$
2	High flow	55-59 %	32 %	$41.68^{\circ}\text{C}$
3	Normal flow	44-55 %	32 %	$44.05^{\circ}\text{C}$
4	Low flow	35-44 %	32 %	$47.26^{\circ}\text{C}$
5	Very low flow	29-35 %	32 %	$51.40^{\circ}\text{C}$

## 2.2. Mathematical model

We adopted the following JMLG 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 (output water temperature),  $x_t \in \mathbb{R}^{n_x}$  denotes the unknown continuous states,  $u_t \in \mathcal{U}$  is a known control signal (steam flow),  $z_t \in \{1, \dots, n_z\}$  denotes the unknown discrete states described in Table 2. We assume the noise processes are *i.i.d.* Gaussian:  $w_t \sim \mathcal{N}(0, I)$  and  $v_t \sim \mathcal{N}(0, I)$ . Note that the parameters  $(A, B, C, D, F, G)$  depend on the discrete state of operation. For each discrete state, we have a single linear-Gaussian model. We ensure that  $D(z_t)D(z_t)^T > 0$  for any  $z_t$ . The initial states are  $x_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$  and  $z_0 \sim P(z_0)$ .

## 2.3. Data acquisition

We engineered a transition matrix  $P(z_t|z_{t-1})$  to physically change the operating conditions of the heat exchanger. For each regime of operation, we monitored the water temperature,  $y_t$ , for more than 30 minutes and collected 1000 time samples.

## 2.4. Parameter identification

The identification consisted of two stages. In the first stage, we adopted an open-loop step response technique [7] to obtain the dynamic model for each discrete state. The parametric identification was guided by the minimum squares error algorithm [8]. The discrete-time state space representation, consisting of matrices  $(A(z_t), C(z_t), F(z_t), G(z_t))$ , was generated by a standard procedure in control engineering [7]. The matrix  $G(z_t)$  was null in our application.

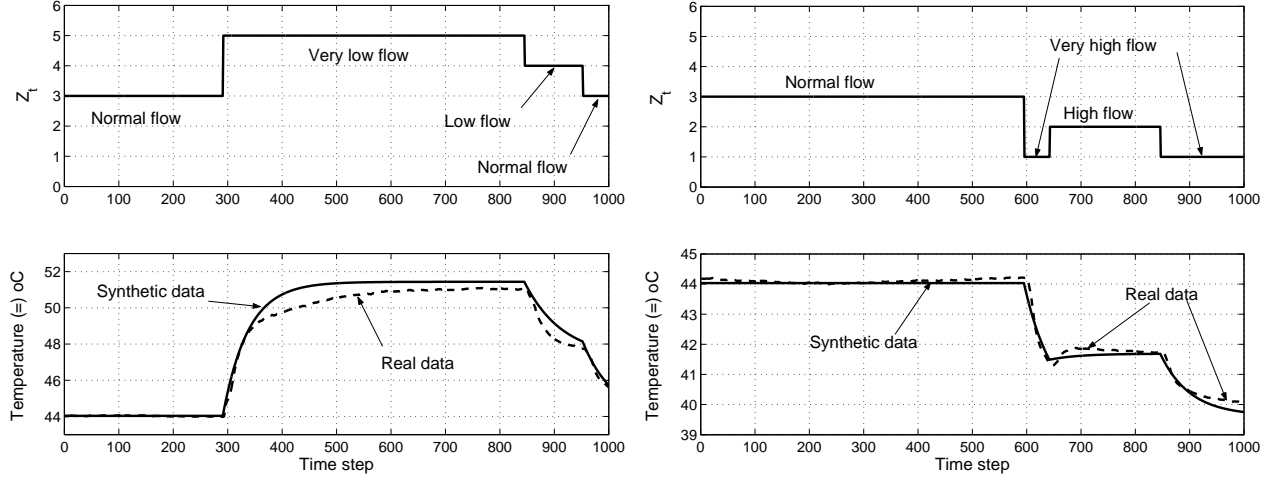
In the second stage, we apply a maximum likelihood (EM) algorithm [3] to refine the estimates of  $(A(z_t), C(z_t), F(z_t), G(z_t))$  and to compute the noise matrices  $(B(z_t), D(z_t))$ . This algorithm consisted of two steps. In the E step, a Rauch-Tung-Striebel Kalman smoother was used to compute the sufficient statistics of the Gaussian states. In the M step, we updated the matrices of parameters using analytically derived equations [3]. We repeated this procedure for each discrete state. Note that the first identification stage contributes significantly toward avoiding convergence to shallow local maxima of the likelihood function.

To show that the JMLG model can successfully describe the heat exchanger with a dynamic mixture of linear models, we compared the measured data to data generated by the model. The graphs in Figure 2 show this data. The upper graphs show the discrete state of operation. The lower graphs show the real data (output water temperature,  $y_t$ ) and the synthetic data generated by the JMLG model. The JMLG model seems to be rich enough to model the nonlinear heat exchanger.

## 2.5. On-line Bayesian monitoring

Our main goal is to determine the discrete state of operation. That is, we want to compute the marginal posterior distribution<sup>1</sup> of the discrete states  $P(z_{0:t}|y_{1:t})$ . (In practice, we avoid storing past trajectories and, hence, focus on the filtering distribution  $P(z_t|y_{1:t})$  as  $t$  increases.) The marginal posterior can

<sup>1</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  $P(d\theta_t) = p(\theta_t) d\theta_t$ .



**Figure 2.** Real data and generated data for two random sequences.

be derived from the posterior distribution  $P(dx_{0:t}, z_{0:t}|y_{1:t})$  using 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. Here, we adopt particle filtering techniques.

### 3. 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 [5] 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 [9].

#### 3.1. 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 PF 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

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

Sequential importance sampling step

- For  $i = 1, \dots, N$ , sample from the transition priors

$$\hat{z}_t^{(i)} \sim P(z_t | z_{t-1}^{(i)}) \quad \text{and} \quad \hat{x}_t^{(i)} \sim P(dx_t | x_{t-1}^{(i)}, z_t^{(i)})$$

and set  $(\hat{x}_{0:t}^{(i)}, \hat{z}_{0:t}^{(i)}) \triangleq (\hat{x}_t^{(i)}, \hat{z}_t^{(i)}, x_{0:t-1}^{(i)}, z_{0:t-1}^{(i)})$ .

- For  $i = 1, \dots, N$ , evaluate and normalize the importance weights

$$w_t^{(i)} \propto p(y_t | \hat{x}_t^{(i)}, \hat{z}_t^{(i)})$$

Selection step

- Multiply/Discard particles  $\{\hat{x}_{0:t}^{(i)}, \hat{z}_{0:t}^{(i)}\}_{i=1}^N$  with respect to high/low importance weights  $w_t^{(i)}$  to obtain  $N$  particles  $\{x_{0:t}^{(i)}, z_{0:t}^{(i)}\}_{i=1}^N$ .

**Figure 3.** PF algorithm at time  $t$ .

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

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

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

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

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:

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

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 [4]. Here, we introduce extra improvements. Let us expand the expression for the importance weights:

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

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

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. In addition to Rao-Blackwellisation, this leads to substantial improvements 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 [5, 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.

## 4. Results and Discussion

### 4.1. State estimation

We tested the three inference algorithms on 10 real datasets. A representative set of results is depicted in Figure 5. The left graph shows the diagnosis error versus computing time per time-step (the signal sampling time was 2 seconds), while the right graph shows the diagnosis error versus the number of particles. The diagnosis error represents how many discrete states were not identified properly. It was calculated for 25 independent runs (1000 time steps each). The graphs shows that RBPF2 always works significantly better (low error rate and very low variance).

The graph in Figure 5 also shows that even for 1 particle, RBPF2 is able to track the discrete state in real time. This is possible thanks to the high accuracy of the sensors (variance = 0.005). That is, 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.

Figure 6 shows the tracking performance of the three algorithms when a step change occurs. By this stage, PF has lost track entirely. RBPF fails to recover when the step occurs. RBPF2, on the other hand, recovers reasonably quickly. Note that the step change leads to an increase in uncertainty. Hence, the RBPF2 estimate of the variance of the continuous states increases accordingly.

#### Kalman prediction step

- For  $i=1, \dots, N$ , and for  $z_t = 1, \dots, n_z$  compute

$$\hat{\mu}_{t|t-1}^{(i)}(z_t), \hat{\Sigma}_{t|t-1}^{(i)}(z_t), \hat{y}_{t|t-1}^{(i)}(z_t), \hat{S}_t^{(i)}(z_t)$$

- For  $i=1, \dots, N$ , evaluate and normalize the importance weights

$$w_t^{(i)} = p(y_t | y_{1:t-1}, z_{0:t-1}^{(i)}) = \sum_{z_t=1}^{n_z} \mathcal{N}(\hat{y}_{t|t-1}^{(i)}(z_t), \hat{S}_t^{(i)}(z_t)) p(z_t | z_{t-1}^{(i)})$$

#### Selection step

- Multiply/Discard particles  $\left\{ \hat{\mu}_{t-1}^{(i)}, \hat{\Sigma}_{t-1}^{(i)}, \hat{z}_{0:t-1}^{(i)} \right\}_{i=1}^N$  with respect to high/low importance weights  $w_t^{(i)}$  to obtain  $N$  particles  $\left\{ \mu_{t-1}^{(i)}, \Sigma_{t-1}^{(i)}, z_{0:t-1}^{(i)} \right\}_{i=1}^N$ .

#### Sequential importance sampling step

- *Kalman prediction.* For  $i=1, \dots, N$ , and for  $z_t = 1, \dots, n_z$  using the resampled information, re-compute

$$\hat{\mu}_{t|t-1}^{(i)}(z_t), \hat{\Sigma}_{t|t-1}^{(i)}(z_t), \hat{y}_{t|t-1}^{(i)}(z_t), \hat{S}_t^{(i)}(z_t)$$

- For  $z_t = 1, \dots, n_z$  compute

$$p(z_t | z_{0:t-1}^{(i)}, y_{1:t}) \propto \mathcal{N}(\hat{y}_{t|t-1}^{(i)}(z_t), \hat{S}_t^{(i)}(z_t)) p(z_t | z_{t-1}^{(i)})$$

- *Sampling step*

$$z_t^{(i)} \sim p(z_t | z_{0:t-1}^{(i)}, y_{1:t})$$

#### Updating step

- For  $i=1, \dots, N$ , use one step of the Kalman recursion to compute the sufficient statistics  $\left\{ \mu_t^{(i)}, \Sigma_t^{(i)} \right\}$  given  $\left\{ \hat{\mu}_{t|t-1}^{(i)}(z_t^{(i)}), \hat{\Sigma}_{t|t-1}^{(i)}(z_t^{(i)}) \right\}$ .

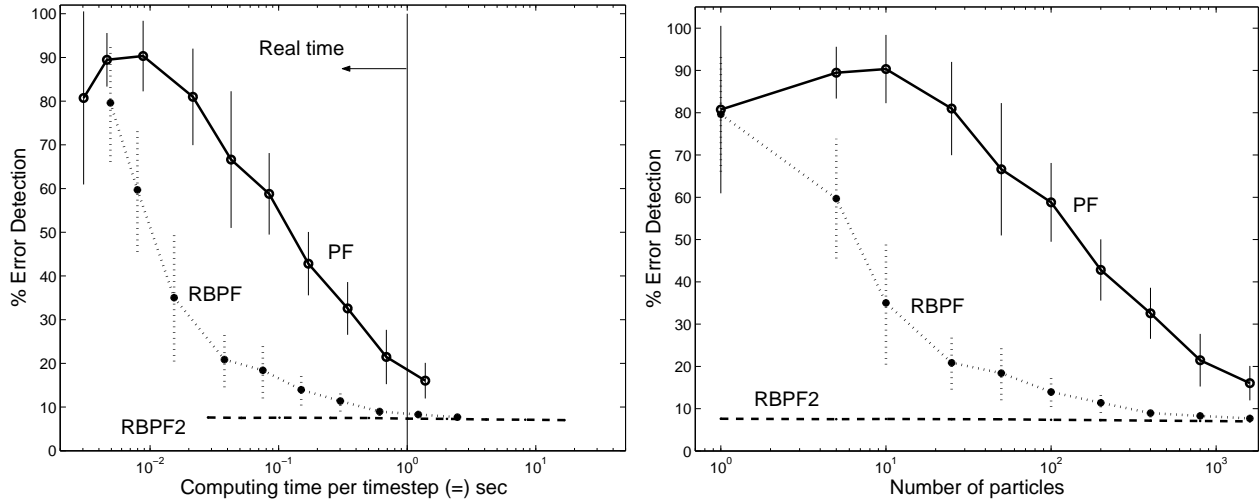
**Figure 4.** RBPF2 algorithm at time  $t$ . The algorithm uses an optimal proposal distribution. It also selects particles from time  $t - 1$  using the information at time  $t$ .

## 4.2. Control system application

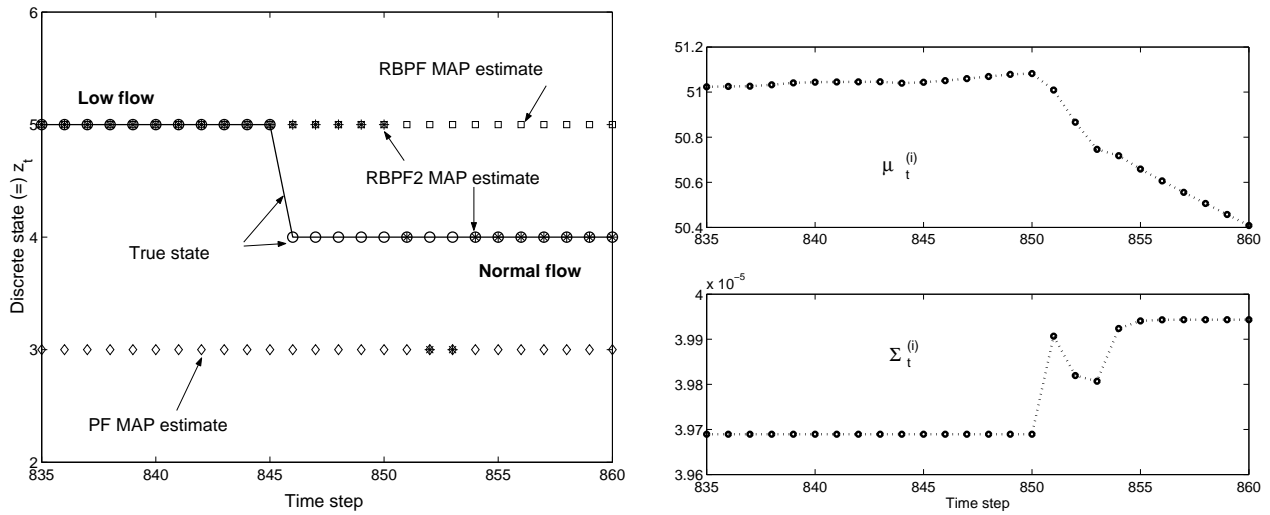
Figure 7 shows a feedback control system, where TIC201 represents a PID controller. (We used a very conventional PID tuning technique [12] and a classical PID equation for this comparative simulation.) This controller regulates the output water temperature around a set-point ( $44^\circ C$ ) by manipulating the steam flow. The basic PID control system has only one set of tuning parameters based on the *Normal flow* model, which is the most representative state. The improved RBPF2-PID control system, on the other hand, has a different set of tuning parameters for each operating condition. In particular, we use RBPF2 to determine the most likely state. Consequently, the PID controller uses the best set of parameters for each discrete state.

We simulated both control systems. Figure 8 shows the changes in operating conditions and the corresponding performance for each control system. The RBPF2-PID exhibited a better transient response (less overshoot and settling time). As shown in the left plot, the simple PID controller can become unstable. This is a consequence of the PID controller maintaining the same tuning parameters despite changes in dynamic behaviour. The RBPF2-PID system, on the other hand, showed stable behaviour.

It is important to note that we could improve this control system without state estimation by designing



**Figure 5.** Diagnosis error versus computing time and number of particles.



**Figure 6.** State transition estimation using 1 particle. Only RBPf2 manages to recover quickly enough after the step change. The right plots show the mean and variance of the continuous states estimated by RBPf2.

a standard feedforward/feedback strategy, but this would demand an additional sensor (e.g. FT203 input water flow) and a nontrivial dynamic lead/lag function [13].

## 5. Conclusions

We presented a probabilistic method for state estimation and control of a complex industrial system. Our experiments demonstrated that our approach, combining EM for parameter estimation and RBPf2 for on-line real-time estimation, works well when controlling an industrial heat exchanger with a conventional PID controller.



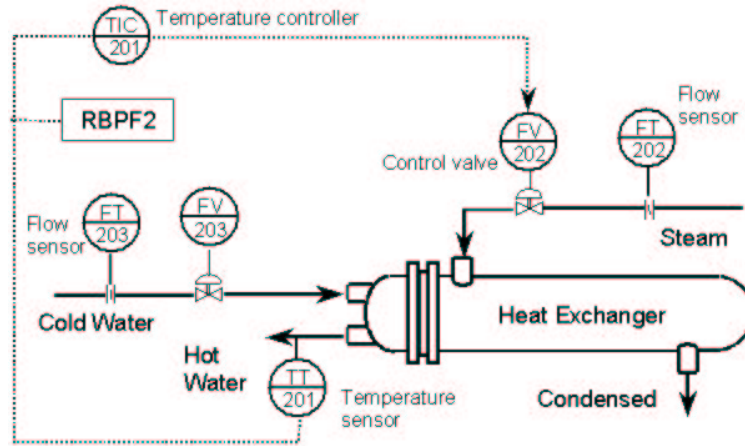


Figure 7. TIC201 : Temperature control system.

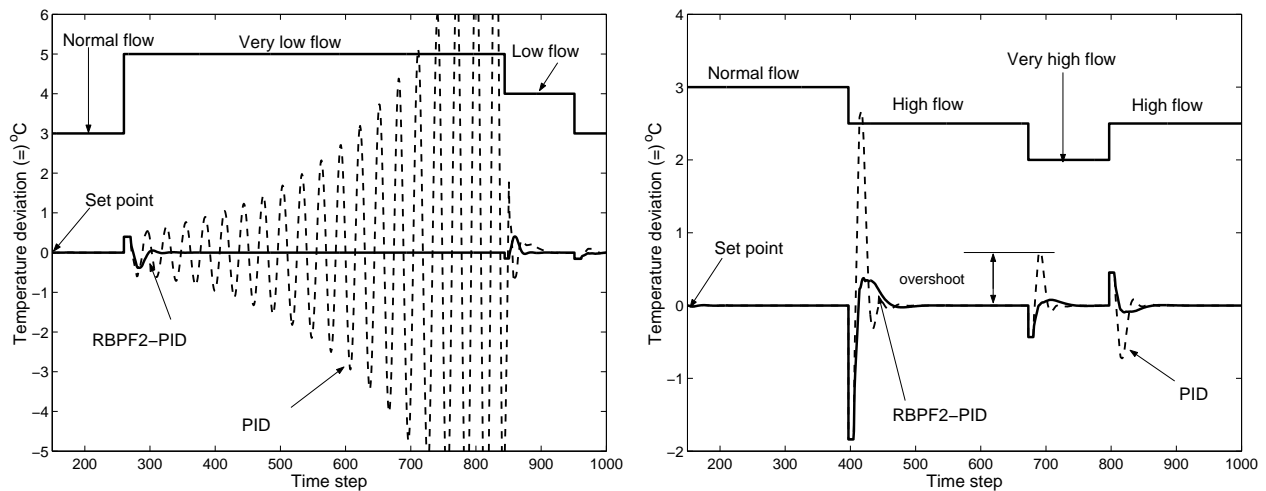


Figure 8. Control system simulation.

## Acknowledgments

We thank Zoubin Ghahramani for providing the EM algorithm code for linear dynamical systems. Ruben Morales-Menéndez was partly supported by the Government of Canada (ICCS), and UBC CS department. David Poole and Nando de Freitas are supported by NSERC.

## References

- [1] U Enste and F Uecker. Use of supervisory information in process control. *Control Engineering Journal*, pages 234–241, October 2000.
- [2] D Dvorak and B Kuipers. Model-based monitoring of dynamic systems. In *11<sup>th</sup> IJCAI*, pages 1238–1243, Detroit, USA, August 1989.
- [3] Z Ghahramani and G E Hinton. Parameter estimation for linear dynamical system. Technical Report CRG-TR-96-2, Department of Computer Science University of Toronto, Toronto, 1996.

- [4] N de Freitas. Rao-Blackwellised particle filtering for fault diagnosis. In *IEEE Aerospace Conference*, 2001.
- [5] A Doucet, N de Freitas, and N J Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.
- [6] R Morales Menendez. Process identification and digital controller design. Master's thesis, ITESM Monterrey, Mexico, 1992.
- [7] K Ogata. *Discrete-Time Control Systems*. Prentice Hall, second edition, 1995.
- [8] L Ljung. *System Identification: Theory for the User*. Prentice-Hall, 1987.
- [9] G Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5:1–25, 1996.
- [10] C Andrieu, A Doucet, and E Punskeya. 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.
- [12] C A Smith and A B Corripio. *Principles and Practice of Automatic Process Control*. John Wiley & Sons, second edition, 1997.
- [13] R Morales-Menendez and C Narvaez. Control strategies design. Technical report, Control Engineering Dept ITESM campus Monterrey, Mexico, 1997.