Estimation and control of industrial processes with particle filters

Rubén Morales-Menéndez
Mechatronics and Automation Dept.
ITESM Campus Monterrey
Monterrey, NL, México 64,849
rmm@itesm.mx

Nando de Freitas and David Poole
Computer Science Dept.
University of British Columbia
Vancouver, BC, Canada V6T 1Z4
{nando,poole}@cs.ubc.ca

Abstract

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.

Index terms: 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 processes can be used to plan maintenance, to choose a suitable control policy, to reduce reprocessing or to improve performance. 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].

We base our work on a real-time, automatic strategy for estimating the states of industrial processes from noisy measurements of continuous variables using particle filters [2, 3, 4]. This approach enables us to reduce the cognitive load experienced by human operators. It also serves to minimize 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. This model allows us to capture the nonlinear process behaviour and disturbance transient responses in a very simple way.

The parameters of each regime are identified off-line with the EM algorithm [5]. 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 [2, 3, 4, 6] and identify the most probable operating condition. A standard PID controller is adjusted with this estimation on-line. Preliminary results show better performance (less overshoot and shorter settling time) and stable behaviour when the process moves away from its normal operating conditions. This paper shows how the results found in [4] can be used for control.

The paper is organized as follows. Section 2 presents an 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.

2.1 Process monitored

We monitored an industrial heat exchanger, Figure 1. This exchanger heats 10 gpm of water from 25^oC to 70^oC using steam at 5 kg/cm^2. This process is fully instrumented and is operated by the Honeywell TDC 3000 (LCN version) industrial distributed control system [7].

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

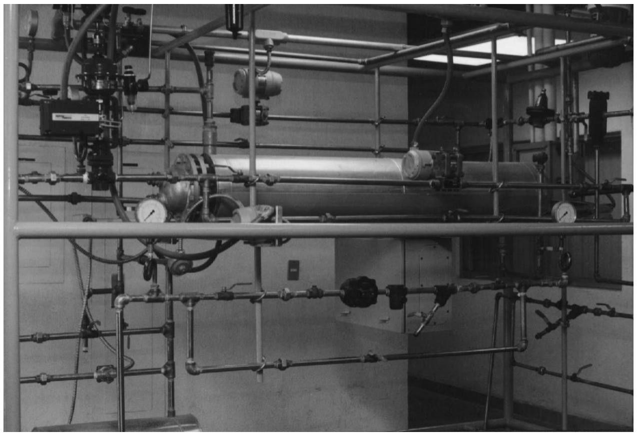

Figure 1: Industrial heat exchanger.

five linear operating ranges shown in Table 1 (steam valve was 32 % open).

Table 1: Linear operating conditions.

z_t	Name	Input water	Water temperature
1	Very high flow	59-71 %	39.69 ^oC
2	High flow	55-59 %	41.68 ^oC
3	Normal flow	44-55 %	44.05 ^oC
4	Low flow	35-44 %	47.26 ^oC
5	Very low flow	29-35 %	51.40 ^oC

2.2 Mathematical model

We adopted the following JMLG model [2]

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

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), $z_t \in \{1, \ldots, n_z\}$ denotes the unknown discrete states. 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 $D(z_t)D(z_t)^{\mathsf{T}} > 0$ 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 45 minutes and collected 2,000 time samples.

2.4 Parameter identification

The identification consisted of two stages. In the first stage, we adopted an open-loop step response technique 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 applied a maximum likelihood (EM) algorithm [5] to refine the estimates of $(A(z_t), C(z_t), F(z_t))$ 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 [5]. 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, Figure 2.4. The upper graph shows the discrete state of operation. The lower graph shows the real data (output water temperature, y_t) and the synthetic data generated by the JMLG model.

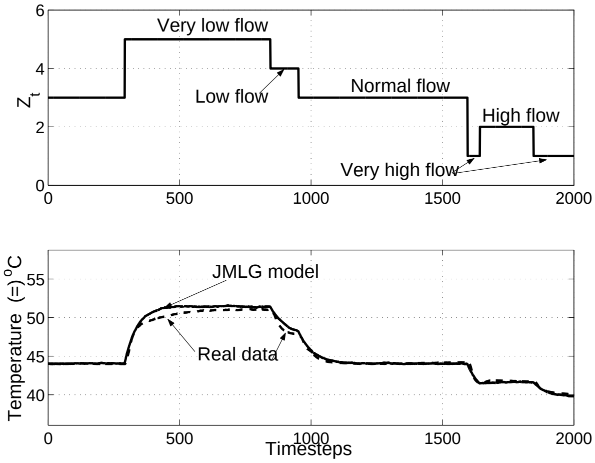

Figure 2: Real data and generated data

2.5 On-line Bayesian monitoring

In order to control the system, we determine the discrete state of operation. That is, we want to compute the marginal posterior distribution[1] of the discrete states $P\left(z_{0:t}|y_{1:t}\right)$. (In

[1] NOTATION: For a generic vector θ, we adopt the notation $\theta_{1:t} \triangleq (\theta_1, \theta_2, \ldots, \theta_t)'$ to denote all the entries of this vector at time t. For simplicity, we use θ_t to denote both the random variable and its realisation. Consequently, we express continuous probability distributions using

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 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:

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

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

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

where $\delta_{x_{0:t}^{(i)}, z_{0:t}^{(i)}}(dx_{0:t}, z_{0:t})$ 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 $P(dx_{0:t-1}^{(i)}, z_{0:t-1}^{(i)}|y_{1:t-1})$, *PF* enables us to compute N particles $\{x_{0:t}^{(i)}, z_{0:t}^{(i)}\}_{i=1}^N$ approximately distributed according to $P(dx_{0:t}^{(i)}, z_{0:t}^{(i)}|y_{1:t})$, 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 [6] 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 factorization $p(x_{0:t}, z_{0:t}|\, y_{1:t}) = p(x_{0:t}|\, y_{1:t}, z_{0:t})\, p(z_{0:t}|\, y_{1:t})$, [10, 11, 3], it is possible to design more efficient *PF* algorithms, and get results with less variance. *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

$$\begin{aligned} p(z_{0:t}|y_{1:t}) &= p(z_{0:t-1}|y_{1:t-1}) \times \\ &\quad \frac{p(y_t|y_{1:t-1}, z_{0:t})\, p(z_t|z_{t-1})}{p(y_t|y_{1:t-1})} \end{aligned} \tag{2}$$

Sequential importance sampling step

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

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

  and set $\left(\widehat{x}_{0:t}^{(i)}, \widehat{z}_{0:t}^{(i)}\right) \triangleq \left(\widehat{x}_t^{(i)}, \widehat{z}_t^{(i)}, x_{0:t-1}^{(i)}, z_{0:t-1}^{(i)}\right)$.
- For $i = 1, ..., N$, evaluate and normalize the importance weights

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

Selection step

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

Figure 3: *PF* algorithm at time t.

This recursion, as equation (1), involves intractable integrals, 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

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

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. This is the basis of the *RBPF* algorithm that was adopted in [2, 11]. This idea can be improved [3, 4]. Let us expand the expression for the importance weights:

$$\begin{aligned} w_t &= \frac{p(z_{0:t}|y_{1:t})}{q(z_{0:t}|y_{1:t})} \\ &= \frac{p(z_{0:t-1}|y_{1:t})}{p(z_{0:t-1}|y_{1:t-1})} \frac{p(z_t|z_{0:t-1}, y_{1:t})}{q(z_t|z_{0:t-1}, y_{1:t})} \qquad (3) \\ &\propto \frac{p(y_t|y_{1:t-1}, z_{0:t})\, p(z_t|z_{0:t-1}, y_{1:t-1})}{q(z_t|z_{0:t-1}, y_{1:t})}. \qquad (4) \end{aligned}$$

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

$P(d\theta_t)$ 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 μ (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 $P(d\theta_t) = p(\theta_t)\, d\theta_t$.

not sampling past trajectories. Sampling past trajectories requires solving an intractable integral [12, 3].

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

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

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:

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

and, hence, the importance weights simplify to

$$\begin{aligned} w_t &\propto p\left(y_t|y_{1:t-1}, z_{0:t-1}\right) \\ &\propto \sum_{z_t=1}^{n_z} p\left(y_t|y_{1:t-1}, z_{0:t-1}, z_t\right) p\left(z_t|z_{t-1}\right) \end{aligned} \tag{7}$$

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 [6, 13]. 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, *look-ahead RBPF (la-RBPF)* [3, 4] whose pseudocode is shown in Figure 4, where $\mu_{t|t-1} \triangleq \mathbb{E}\left(x_t|\, y_{1:t-1}\right)$, $\mu_t \triangleq \mathbb{E}\left(x_t|\, y_{1:t}\right)$, $y_{\,t|t-1} \triangleq \mathbb{E}\left(y_t|\, y_{1:t-1}\right)$, $\Sigma_{t|t-1} \triangleq cov\left(x_t|\, y_{1:t-1}\right)$, $\Sigma_t \triangleq cov\left(x_t|\, y_{1:t}\right)$ and $S_t \triangleq cov\left(y_t|\, y_{1:t-1}\right)$.

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 Figures (5-6).

```
Kalman prediction step
  • For i=1, ..., N, and for $z_t = 1,\dots,n_z$ compute
    $\widehat{\mu}^{(i)}_{t|t-1}(z_t)$, $\widehat{\Sigma}^{(i)}_{t|t-1}(z_t)$, $\widehat{y}^{(i)}_{t|t-1}(z_t)$, $\widehat{S}^{(i)}_t(z_t)$
  • For i=1 , . . . , N , evaluate and normalize the importance weights
      $w_t^{(i)} = p(y_t|y_{1:t-1}, z^{(i)}_{0:t-1})$
      $\quad = \sum_{z_t=1}^{n_z} \mathcal{N}(\widehat{y}^{(i)}_{t|t-1}(z_t), \widehat{S}^{(i)}_t(z_t)) p(z_t|z^{(i)}_{t-1})$
Selection step
  • Multiply/Discard particles $\left\{\widehat{\mu}^{(i)}_{t-1}, \widehat{\Sigma}^{(i)}_{t-1}, \widehat{z}^{(i)}_{0:t-1}\right\}_{i=1}^N$
    with respect to high/low importance weights $w_t^{(i)}$ to obtain $N$ particles $\left\{\mu^{(i)}_{t-1}, \Sigma^{(i)}_{t-1}, z^{(i)}_{0:t-1}\right\}_{i=1}^N$.
Sequential importance sampling step
  • Kalman prediction. For i=1, ..., N, and for $z_t = 1,\dots,n_z$ using the resampled information, re-compute
    $\widehat{\mu}^{(i)}_{t|t-1}(z_t)$, $\widehat{\Sigma}^{(i)}_{t|t-1}(z_t)$, $\widehat{y}^{(i)}_{t|t-1}(z_t)$, $\widehat{S}^{(i)}_t(z_t)$
  • For $z_t = 1,\dots,n_z$ compute
      $p(z_t|z^{(i)}_{0:t-1}, y_{1:t}) \propto \mathcal{N}(\widehat{y}^{(i)}_{t|t-1}(z_t), \widehat{S}^{(i)}_t(z_t)) p(z_t|z^{(i)}_{t-1})$
  • Sampling step $z_t^{(i)} \sim p(z_t|z^{(i)}_{0:t-1}, y_{1:t})$
Updating step
  • For i=1 , . . . , N, use one step of the Kalman recursion to compute the sufficient statistics
    $\left\{\mu_t^{(i)}, \Sigma_t^{(i)}\right\}$ given $\left\{\widehat{\mu}_{t|t-1}(z_t^{(i)}), \widehat{\Sigma}_{t|t-1}(z_t^{(i)})\right\}$.
```

Figure 4: *la-RBPF* algorithm at time t.

Figure 5 shows the diagnosis error versus the number of particles, while Figure 6 shows the diagnosis error versus computing time per time-step (the signal sampling time was 2 seconds).

The diagnosis error represents how many discrete states were not identified properly. It was calculated for 25 independent runs (2,000 time steps each). The graphs show that *la-RBPF* always works significantly better (low error rate and very low variance).

These graphs also show that even for 1 particle, *la-RBPF* is able to track the discrete state in real time. This is possible thanks to the high accuracy of the sensors (variance = 0.005). Note that this also works better for less accurate sensors [4]. That is, the distributions are very peaked and we are simply tracking the mode. Note that *la-RBPF* 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 7 shows a representative example of the tracking performance of the three algorithms when a step change occurs.

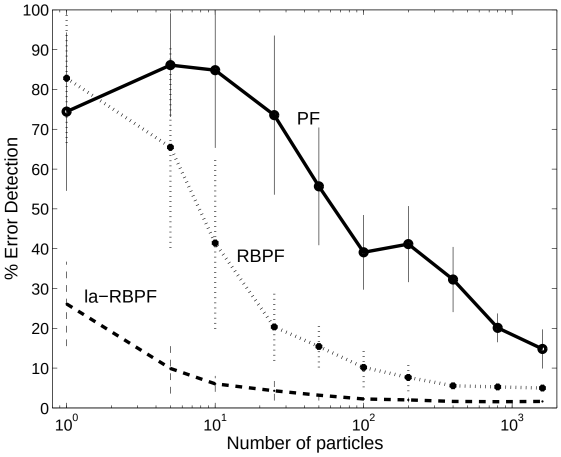

Figure 5: Diagnosis error versus number of particles.

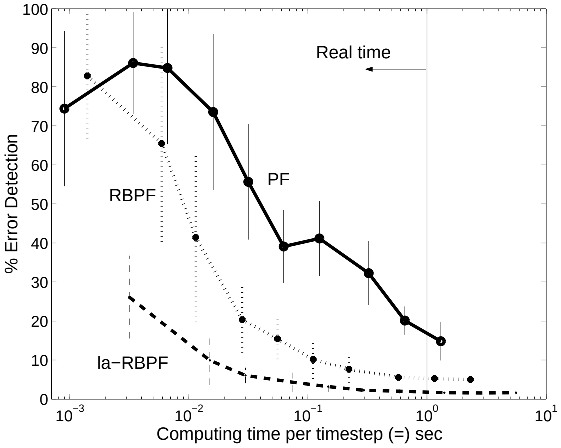

Figure 6: Diagnosis error versus computing time.

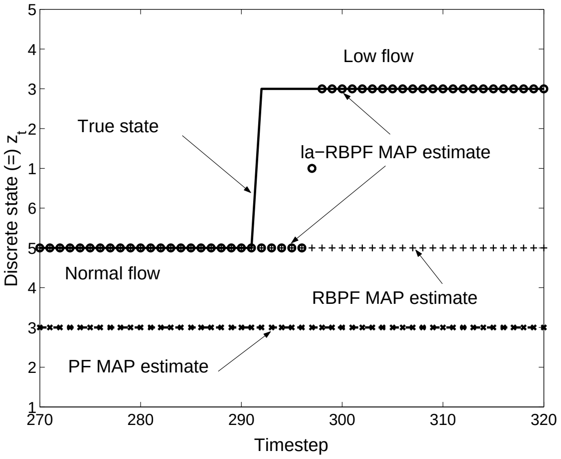

Figure 7: How the different algorithms track a typical state change.

By this stage, PF has lost track entirely. RBPF fails to recover when the step occurs. *la-RBPF*, on the other hand, recovers reasonably quickly. Note that the step change leads to an increase in uncertainty, and the *la-RBPF* estimate of the variance of the continuous states increases.

4.2 Control system application

Figure 8 presents a heat exchanger conceptual diagram and its main instrumentation; standard ISA nomenclature was used (see Table 2 for a complete description). A feedback control system is shown where TIC201 represents a PID controller. (We used a very conventional PID tuning technique [14] and a classical PID equation for this comparative simulation.) This controller regulates the output water temperature around a set-point (44 ^oC) by manipulating the steam flow. The basic PID control system has only one set of tuning parameters based on the *Normal flow* state, which is the most representative state. The improved *la-RBPF*-PID control system, on the other hand, has a different set of tuning parameters for each operating condition. In particular, we use *la-RBPF* 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, in order to show the changes in operating conditions and the corresponding performance for each one. In Figure 9 the *la-RBPF*-PID exhibited a better transient response (less overshoot and settling time). The PID is a robust controller which can work under small changes in operating conditions (timesteps 150-650); however, the simple PID controller can become unstable for the very low flow conditions (timesteps 650-950). This is a consequence of the PID

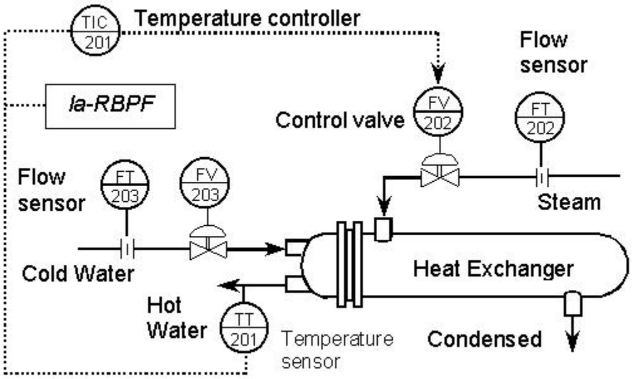

Figure 8: Temperature feedback control system.

Table 2: Heat exchanger instrumentation.

Tag name	Instrument/Description
FT203	Flow transmitter/Input water flow, %
FV203	Control valve/Input water valve, %
FT202	Flow transmitter/Steam flow, %
FV202	Control valve/Steam valve, %
TT201	Temp transmitter/Output water, ^oC
TIC201	Controller/Output water temperature PID

controller maintaining the same tuning parameters despite changes in dynamic behaviour. For the very low input water flow the gain and dead time grow considerably and the control system becomes unstable. The *la-RBPF*-PID system, on the other hand, showed stable behaviour because the PID is adapting its parameters as the dynamic process changes.

We could tune the PID controller for the worst condition (very low flow) and always get a stable control system; however, the control system would perform poorly under normal conditions (the most common state.)

It is important to note that we could improve this control system without state estimation by designing 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 [15].

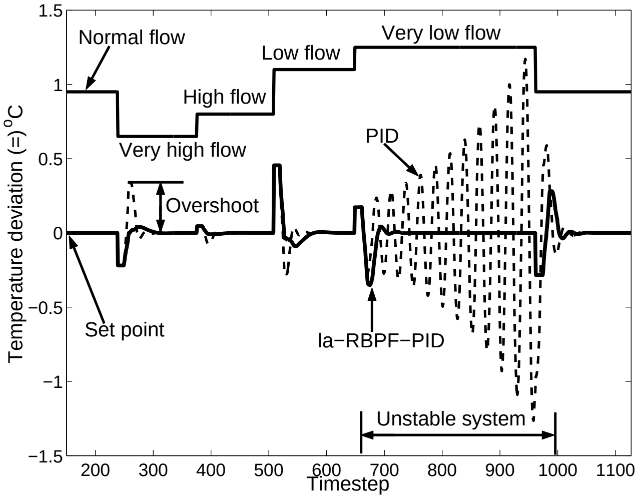

Figure 9: Control system simulation.

5 Conclusions

We used 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 *la-RBPF* for on-line real-time estimation, works well when controlling an industrial heat exchanger with a conventional PID controller. One can extend this idea and design flexible control strategies for complex processes or processes under many disturbances.

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

References

[1] D L Dvorak. *Monitoring and Diagnosis of Continuous Dynamic Systems usign Semiquantitative simulation*. PhD thesis, University of Texas at Austin, Austin, Tx, USA, 1992.

[2] N de Freitas. Rao-Blackwellised particle filtering for fault diagnosis. In *IEEE aerospace conference*, 2001.

[3] A Doucet, N J Gordon, and V Krishnamurthy. Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–624, 2001.

[4] R Morales-Menendez, N de Freitas, and D Poole. Real-time monitoring of complex industrial processes with particle filters. In *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2002. MIT Press.

[5] 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.

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

[7] R Morales-Menendez. Process identification and digital controller design. Master's thesis, ITESM campus Monterrey, México, July 1992.

[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] G Casella and C P Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

[11] A Doucet, N de Freitas, K Murphy, and S Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In C Boutilier and M Godszmidt, editors, *Uncertainty in artificial intelligence*, pages 176–183. Morgan Kaufmann Publishers, 2000.

[12] C Andrieu, A Doucet, and E Punskaya. 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.

[13] M Pitt and N Shephard. Filtering via simulation: auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599, 1999.

[14] C A Smith and A B Corripio. *Principles and Practice of Automatic Process Control*. John Wiley & Sons, second edition, 1997.

[15] R Morales-Menendez and C Narvaez. Control strategies design. Technical report, Control Engineering Dept ITESM campus Monterrey, Mexico, 1997.