Estimation of States of Nonlinear Systems using a Particle Filter

Syed A. Imtiaz, Kallol Roy[1], Biao Huang, Sirish L. Shah, Phanindra Jampana

Department of Chemical and Materials Engineering
University of Alberta
Edmonton, AB, T6G 2G6, Canada
sirish.shah@ualberta.ca
http:\\www.ualberta.ca\ ~ slshah

***Abstract*— Partilce Filters can estimate the states of nonlinear and non-Gaussian systems without any approximation when the number of particles tends to infinity. However, the method is not popular in industry because the implementation details are missing in the literature. In this paper we discuss several implementation issues and propose novel techniques for tuning the Particle filter and dealing with multi-rate data. The performance of the proposed methodologies are demonstrated using a simulated non-linear CSTR and an Experimental Four Tank system.**

I. Introduction

The estimation of unmeasured states is an important problem in process industries, primarily because knowledge of such states lead to better control. Processes (e.g., power generation, nuclear, chemical etc.) are generally nonlinear. However, often within the region of operation they can be approximated by linear models. Different linear state estimation methods such as, Data Reconciliation and Kalman Filters are used to estimate the states. Several methods have also been developed to deal with the nonlinearities of the process,e.g., Extended Kalman Filter (EKF), Unscented Kalman Filter(UKF), Moving Horizon Estimation (MHE) etc. Some of these methods are not optimal for dealing with the nonlinearities, and moreover all of these methods fail if the posterior states or the noise characteristics are non-Gaussian. In this context Sequential Importance Re-sampling (SIR) or Particle Filtering is very relevant, as this can deal with the nonlinearities, as well as, non-Gaussianity of the posterior density of the states and measurements. Sequential Importance Re-sampling Filters or Particle Filters belong to the class of Bayesian Filters. The improved performance of the filters have been demonstrated by [13]. However, the application of Particle Filter in state estimation is not yet popular as the implementation details are missing in the literature. Many of the application challenges arise from the fact that obtaining non-linear model is time consuming and not easy as it requires fairly accurate physio-chemical or mechanistic description of the process. Therefore it is not uncommon to see significant mismatch between the process and the model, and unless due attention is given to tuning the filter, the performance of the filter may be very poor. In addition, there also arises challenges on the measurement side. In process industries all measurements may not be always available at every sampling event. Often measurements may be missing due to lack of sensors, multi-rate sampling strategy, process upset etc. So there is a need to adapt the estimation filter for missing data. In this paper we investigate these implementation issues and propose novel techniques for tuning the Particle filter and dealing with multi-rate data.

II. Problem Formulation and Notations

The following non-linear form of the state space model is used for the present filtering problem.

$$\begin{aligned} x_k &= f\left(x_{k-1}, u_{k-1}\right) + \omega_k \\ y_k &= g\left(x_k\right) + \nu_k \end{aligned} \tag{1}$$

where x_k is the state vector, f is the non-linear system equation , g is measurement equation and y_k is the measurement vector. System noise ω_k represents disturbances, all unmodelled dynamics, and any mismatch between the process and the model. Measurement noise, ν_k captures the inaccuracy in measuring system. Unlike most other state estimation methods, process or measurement noise do not necessarily have to be assumed to be Gaussian; rather any standard or non-standard distribution can be assumed depending on the behavior of the process noise or the measurement noise.

III. Sequential Importance Re-sampling

The state estimation objective is to evaluate the following equation:

$$E_{p(x_{1:n}|y_{1:n})}[x_n] = \int_{x_n} x_n p(x_{1:n}|y_{1:n}) dx_n \tag{2}$$

SIR filter evaluates the posterior density function $p(x_{1:n}|y_{1:n})$. There are two issues associated with the evaluation of the above equation: (i) it is not possible to directly sample from $p(x_{1:n}|y_{1:n})$ (ii) it has to be evaluated in an on-line recursive manner. With the arrival of each new measurement the conditional density function needs to be updated without evaluating the previous states. Importance Sampling (IS) is used to sample from density functions which are difficult to sample and an importance function of the form $q(x_{1:n}|y_{1:n})$ allows recursive evaluation of the importance weight in time.

$$E_{p(x_{1:n}|y_{1:n})}[x_n] = \int_{x_n} x_n \frac{p(x_{1:n}|y_{1:n})}{q(x_{1:n}|y_{1:n})} q(x_{1:n}|y_{1:n}) dx_n \tag{3}$$

where $W_n = \frac{p(x_{1:n}|y_{1:n})}{q(x_{1:n}|y_{1:n})}$ is the recursive weight. The proposal distribution $q(x_{1:n}|y_{1:n})$ can be factorized into the following form:

$$q(x_{1:n}|y_{1:n}) = q(x_n|x_{1:n-1}, y_{1:n}) q(x_{1:n-1}|y_{1:n-1}) \tag{4}$$

Since the states are Markovian $q(x_n|x_{1:n-1}) = q(x_n|x_{n-1})$ Equation 4 can be further simplified and written as

$$q(x_{1:n}|y_{1:n}) = q(x_n|x_{n-1}, y_{1:n}) q(x_{1:n-1}|y_{1:n-1}) \tag{5}$$

[1] Visiting from Bhabha Atomic Research Centre, Bombay, India

The target distribution $p(x_{1:n}|y_{1:n})$ can be factorized to the following form:

$$p(x_{1:n}|y_{1:n}) = p(x_n|x_{1:n-1}, y_{1:n})p(x_{1:n-1}|y_{1:n}) \quad (6)$$
$$= p(x_n|y_{1:n})p(x_{1:n-1}|y_{1:n}) \quad (7)$$
$$= \frac{p(y_n|x_n)p(x_n|y_{1:n-1})}{p(y_n|y_{1:n-1})}p(x_{1:n-1}|y_{1:n-1}) \quad (8)$$

Therefore using the factored form of Equation 5 and 8 the importance weight W_n can be updated recursively,

$$W_n = \frac{p(x_{1:n}|y_{1:n})}{q(x_{1:n}|y_{1:n})} \quad (9)$$
$$\propto \frac{p(y_n|x_n)p(x_n|x_{n-1})p(x_{1:n-1}|y_{1:n-1})}{q(x_n|x_{1:n-1}, y_{1:n})q(x_{1:n-1}|y_{1:n-1})} \quad (10)$$
$$\propto \frac{p(y_n|x_n)p(x_n|x_{n-1})}{q(x_n|x_{1:n-1}, y_{1:n})}W_{n-1} \quad (11)$$

With transition density as the importance function weight is given by,

$$W_n \propto \frac{p(y_n|x_n)p(x_n|x_{n-1})}{q(x_n|x_{1:n-1}, y_{1:n})}W_{n-1} \quad (12)$$
$$\propto p(y_n|x_n)W_{n-1} \quad (13)$$

The implementation of SIR algorithm is shown schematically in Figure 1. SIR filters have essentially two steps, (i)Prediction and (ii) Update. With Transition Density as the importance function these two steps reduce the implementation to the following actions:

Prediction Each sample is passed through the system model to obtain samples from the prior at time step k: $x_k(i)^* = f(x_{k-1}(i)^*) + w_{k-1}$ where $w_{k-1} \sim p(w_{k-1})$.

Update The update step is essentially an implementation of the Bayes rule $p(x_k|x_{k-1}, y_k) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}$. However instead of performing the calculations on density functions, the operation is carried out on the discrete samples. On receipt of the measurement y_k the likelihood is used to calculated the weights. The generated weights shape the sampled particles from the prior, $p(x_k|x_{k-1})$ to the samples from posterior, $p(x_k|x_{k-1}, y_k)$. The update step is executed using a re-sampling strategy. After re-sampling the weights of the particles are reset to $\frac{1}{N_p}$. Equation 11 is modified in the following form:

$$W_n \propto p(y_n|x_n)W_{n-1} \quad (14)$$
$$\propto p(y_n|x_n)\frac{1}{N_p} \quad (15)$$

At any given time step, the weights are given by the likelihood function $p(y_k|x_k)$. Re-sampling transfers the weight information to the samples. The re-sampled samples progress to the next time step through the state transition equation. Therefore the measurement information also gets transmitted from one time step to the next time step through re-sampling.

A. Implementation Issues in SIR Filter

Several implementation issues have been pointed out by [5] in their original paper. The problem arises from the fact that a limited sample size is used for approximating the probability density functions. Therefore a good overlap between the prior and the likelihood is important. The prior should be proposed in a way such that, there will exist many samples from the prior $p(x_k|x_{k-1}, y_{1:k-1})$ in the region where the likelihood $p(y_k|x_k)$ takes significant values. This will ensure that many samples from the prior will receive large weights. If the prior space and the likelihood space is very different then most of the samples will receive very small weights and thereby get wasted. [5] suggested use of larger process noise than the actual process noise to expand the proposal distribution. This was called 'roughening' of the prior. Results have been shown that the tracking performance was improved by roughening the priors. However, if the SIR filter is used for estimating the unmeasured states this may not be adequate. Increasing the process noise arbitrarily is also equivalent to suggesting that the mismatch between the process and the model is large. Under this scenario the model is considered less reliable than the measurements. One consequence of this is that the estimated states may get very oscillatory and in the extreme cases diverge from the true states. Also there is no direct way of tuning the parameters if the filter is being used for estimation of unmeasured states. In this paper we investigate this further from a state estimation perspective and propose the following practical ways of tuning the filter.

1) Weights vs. a-Prioi State Plot: In SIR filter the transition density is used as the importance function and therefore the structure of the prior and the likelihood is fixed. Process and measurement noise are the only two parameters that can be used for tuning the filter. The process noise essentially governs the spread of the prior samples, and the ratio of measurement noise to process noise controls the variance of the weights. An increase in the ratio makes the weights more uniform. The relative magnitude of the process noise and the measurement noise also indicates where we have more confidence, in the model or the measurement noise. Therefore it is important to maintain a balance between these two quantities. However, measurements of all states are not always available to tune the filter parameters directly. A qualitative way of finding the range of the tuning parameters may be by visualizing the 'weights vs. a-priori states' plot. Both process noise and the measurement noise appear in the likelihood expression and the ratio between the process noise and measurement noise will determine the variance of the weights. A schematic diagram of the 'weights vs. a-priori states' is shown in Figure 2. The top curve with large variance in weights may lead to excessive re-sampling and ultimately select very few samples from the priors. On the other hand, a flat distribution of weights signify that the update step is not taking a part in shaping the prior. Although according to the discussion of optimal prior we would like to have variance in the weights close to zero, but from a practical point of view this is not feasible. Primarily because the importance function is not optimal and in this case we would like the re-sampling to shape the priors. Since there is mismatch between the process and the model, we would like the measurements to play a role in updating the states. Therefore one is justified to use a distribution of weights which is slightly curved. In Figure 2 the region within the dotted lines is the desired shape for the weights. The exact shape will be dependent on the specific system. The strategy would be to first select a process noise large enough to show some jitters in the estimated states but not to make it too oscillatory. Then the measurement noise can be increased until we converge to the weights which look similar to that shown in Figure 2.

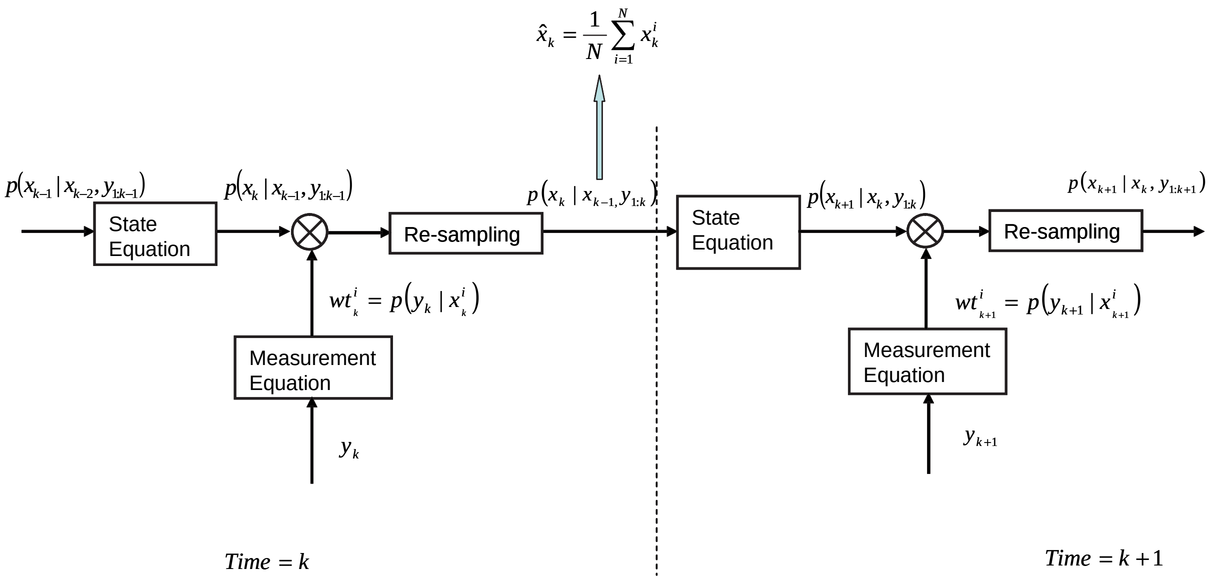

Fig. 1. *Schematic diagram explaining the implementation steps of the SIR algorithm*

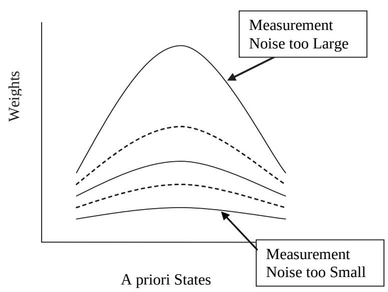

Fig. 2. *A schematic diagram of weight vs. a priori state plot depicting the optimum region for tuning purpose*

Finally more tuning can be done to match the measurement with the predicted values.

2) Annealing of the Weights: An increase in the process noise may increases the variance of the weights and produce lots of jitters in the estimated states. Since optimal weighting suggests decreasing the variance of the weights, a heuristical way is to use an annealing term on the likelihood to reduce the variance of the weights,

$$wt_k{}^i = p(y_k|x_k{}^i)^\alpha where \quad \alpha \sim \; [0 \quad 1] \tag{16}$$

Normalized weights,

$$\bar{wt_k}^i = \frac{wt_k{}^i}{\sum_{i=1}{}^N wt_k{}^i} \tag{17}$$

where α is the annealing parameter. Decrease of variance of the weights has a direct effect on the re-sampling step. Since the variance of weights will be smaller, the re-sampled values will be a progeny of many values from the original samples. Therefore, the estimated state will be an average of a set of diverse samples and any unusual spike will die out. The implementation of these two strategies will be further explained with examples in Section IV.

B. Multi-rate Data Handling in SIR

In the process industry often quality variables (e.g., concentration) are less frequently measured compared to the condition variables (e.g., Temperature) which gives rise to multi-rate data. Since the filters are developed for regularly sampled data, they need to be adapted to multi-rate data so that all available information is used. Let us consider the system described in Section II. In a multi-rate sampling setting, at any time instant *k*, the measurements can be divided into two distinct parts $y_k = [y_k{}^{fast} y_k{}^{slow}]$. $y_k{}^{fast}$ is from sensors with faster sampling rate and $y_k{}^{slow}$ is from sensors with a slower sampling rate and only available at some intermittent sampling points, we denote these sampling points as ‘Major Sampling Event’ while the instants where only fast sampled measurements are available are denoted as ‘Minor Sampling Event’. The prediction step remains unchanged, only the update step needs to be adapted for multi-rate data. The following two modifications of the update step are proposed for dealing with multi-rate data in the SIR filter. ***Strategy I*** The weights at the update step of a SIR filter are given by the likelihood equation:

$$wt^i = p(y_k|x_k{}^i)$$

A two step update procedure can be used to calculate the weights of the SIR filter. *Minor Sampling Event:* At the minor sampling event $y_k = y_k{}^{fast}$ and the weights are given by,

$$wt^i = p(y_k{}^{fast}|x_k{}^{i,fast}) \tag{18}$$

$$R = [R^{fast}] \tag{19}$$

Major Sampling Event: The slow sampled measurement are available at the major sampling event and the measurements become $y_k = [y_k{}^{fast} y_k{}^{slow}]$. Weights can be calculated by the following conditional density function,

$$wt_k^i = p([y_k{}^{fast} y_k{}^{slow}]|[x_k{}^{i,fast} x_k{}^{i,slow}]) \tag{20}$$

$$R = \begin{bmatrix} R^{fast} & 0 \\ 0 & R^{slow} \end{bmatrix}$$

Also R^{slow} is set to a value much smaller than R^{fast} to give more weight to slow sampled measurements at the major sampling events. The main disadvantage of the method is at the minor sampling events it relies heavily on the prediction equation for state estimation. This is disadvantageous because in most of the cases there would be a mismatch between the process and the model. Therefore the states estimated by the prediction equation should not be trusted absolutely. ***Strategy II*** The major sampling events contain information about the mismatch between the model predicted values and the slow sampled measurements. This information can be used to correct the model predicted value. However, in order to make the estimation robust, a Multiple Imputation [6] strategy can be adopted in the update stage. The main idea of Multiple Imputation is to fill the missing measurements with all possible estimates of the measurements. This will result in several complete data sets. Parameters are estimated for each of the imputed data set and an average can be used as the estimate. The method provides a way of evaluating the sensitivity of the estimated parameters to the imputed missing values. The details of the method can be found in [6]. In an MI framework the update step of SIR filter can be implemented as follows: *Major Sampling Event:* At the major sampling event both fast and slow measurements are available $y_k = [{y_k}^{fast}{y_k}^{slow}]$ and the weights are calculated using Equation 20. In addition to that, the residual between the slow sampled measurement and the model predicted state, $r_k = y_k - x_k$ is calculated and stored for predicting the unmeasured measurements at minor sampling events. *Minor Sampling Event:* All possible values of the missing measurements, ${y_k}^{slow,(d)}$ are calculated by adding the residuals calculated at the Major Sampling Events,

$$\hat{y}_k^{slow,(d)} = \bar{x}_k + r_k^{(d)}$$

The estimated values are imputed in the data set and the complete data set is given by, ${y_k}^(d) = [{y_k}^{fast}{y_k}^{slow,(d)}]$ where $d = [1 \cdots D]$. Weights are calculated for each data set,

$${wt_k}^{i,d} = p([{y_k}^{fast}{y_k}^{slow,d}]|[{x_k}^{i,fast}{x_k}^{i,slow}])$$

At any time step *k* there would be DN weights, where *N* is the number of samples and *D* is the number of imputations. The calculated weights are used to re-sample the a priori states. Finally, the expected value and the variance of the state is given by,

$$E_{p(x_k|y_k)}(x_k) \approx \frac{1}{N}\sum_{i=1}^{N}\frac{1}{D}\sum_{d=1}^{D} x_k^{i,d} = \bar{x}_k \quad (21)$$

$$var(x_k) = \frac{1}{(N-1)(D-1)}\sum_{i=1}^{N}\sum_{i=1}^{D}({x_k}^{i,d} - \bar{x}_k)^2 \quad (22)$$

After re-sampling there would be ND sets of state vectors. For calculating the priors of the next time step, *N* samples can be selected randomly and translated through the state transition equation. The method is computationally intensive however, it has a similar effect to that of boosting the prior and therefore minimizes sample impoverishment.

IV. Results and Discussions

In this section we illustrate the proposed methodology to overcome different implementation difficulties of a Particle Filter using a simulated non-linear adiabatic CSTR [7] and a laboratory scale Four Tank setup.

A. Non-linear CSTR

The non-linear CSTR and its operating conditions have been taken from [7]. This system was also studied by [13] to demonstrate the different aspects of SIR Filters. The governing equations of the system are as follows:

$$\frac{dC}{dt} = \frac{q}{V}(C_0 - C) - kCe^{-E_A/T}$$
$$\frac{dT}{dt} = \frac{q}{V}(T_0 - T) - \frac{\Delta H}{\rho C_p}kCe^{-E_A/T} - \frac{UA}{\rho C_p V}(T - T_c) \quad (23)$$

where *C* is the concentration, *T* is the temperature, *q* is the flowrate, *V* is the volume of the reactor, C_0 and T_0 are inflow concentration and temperature, $kCe^{-E_A/T}$ is the reaction rate, ΔH is the heat of reaction, ρ is the density, C_p is the specific heat, *U* and *A* are the effective overall heat-transfer coefficient and area of heat transfer, T_c is the temperature of the cooling fluid. We consider two measurement strategies for the system. In a single rate sampling setup only the temperature sensor is available and measured at 1 sec sampling interval. The measurement equation has the following form,

$$y_k = 0.5T_k + \nu_k^I \quad (24)$$

A multi-rate sampling strategy was also considered to demonstrate the performance of the proposed multi-rate SIR filter. In a multi-rate sampling setup the temperature measurements are available at every second, while concentration is measured at every 1 minute interval. The measurement equation at the major sampling instant can be written as,

$$y_k^{major} = 0.5\begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}\begin{bmatrix} T_k \\ C_k \end{bmatrix} + \nu_k^{major} \quad (25)$$

One of the main objectives of this exercise is to study the effect of system noise and process noise on the estimation performance of the SIR filter. Therefore, the system was simulated for a wide range of process noise and measurement noise. We also considered poor guesses as initial conditions for the system , common in many practical situations, to demonstrate the convergence of the algorithm.

B. Tuning of SIR filter

The graphical method described in Section III-A.1 was used to tune the SIR filter for the CSTR system. The process noise and the measurement noise for the system on a normalized scale are,

$$p(\omega_k) \sim N\left(\begin{matrix} 0 \\ 0 \end{matrix}, \begin{matrix} 0.0001 & 0 \\ 0 & 0.0004 \end{matrix}\right) \quad (26)$$
$$p(\nu_k) \sim N(0, 0.2) \quad (27)$$

In the filter we used a process noise one order higher than the actual process noise. Setting the process noise to a higher value makes the prior wide and helps the prior to cover the actual state of the system. However the variance should not be increased more than one order of magnitude as it gives rise

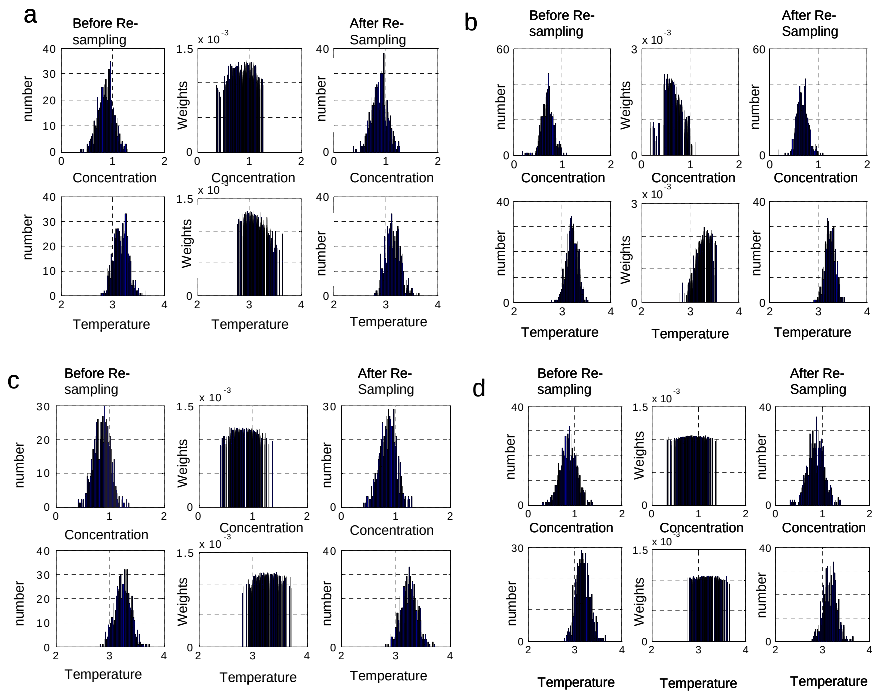

Fig. 3. *Weights vs. a-priori state plot depicting the tuning methodology of the non-linear CSTR*

to large oscillations in the estimated states of the unmeasured variables. Keeping the process noise to the above value we changed the measurement noise of the process and at the same time observed the shape of the 'weight vs. a-priori state' plots. The shape of the weights for four different cases are shown in Figures 3(a) to (d). In each of these figures we also plotted the distribution of the states before and after re-sampling. Clearly if the variance of the weights is very large, the distribution of the state changes significantly before and after the re-sampling. The sum squared error between the estimated concentration and actual concentration for these four cases are plotted in Figure 4 in the same order. The smallest sum squared error was obtained for Case (c). From the 'weight vs. a-priori state plot' it is also evident that the variance of the weights is moderate for Case (c), and due to re-sampling the distribution of the states did not change significantly. Therefore under this condition both the model and the measurements played equal roles in estimating the states. The 'Weights vs. a-priori plot' gives an qualitative idea about the ratio of the process noise and the measurement noise where the filter may work best. The method works well if an order of magnitude information on the variance of the process noise is available. However, in addition to the 'Weights vs. a-priori plot' we suggest visual inspection of the trend plots of the measured states and the estimated states. Annealing of the weights is another way of tuning the particle filter. The effect of annealing of the weights is demonstrated in Figures 5 and 6. In both cases we used the same process noise and measurement noise as tuning parameter. However Figure 6 shows the estimates of the states when an annealing parameter $\alpha = 0.2$ was used. Clearly the jitters have disappeared and the trend of the estimated states are very similar to the expected behavior of the states.

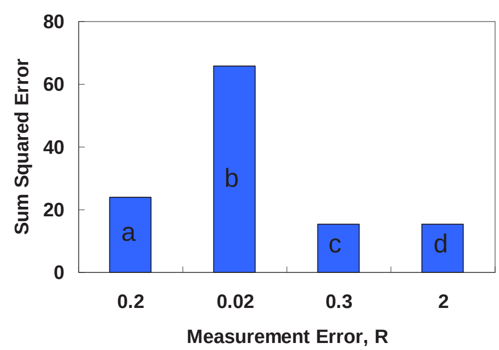

Fig. 4. *Effect of tuning measurement noise on the sum squared error between the concentration and estimated concentration*

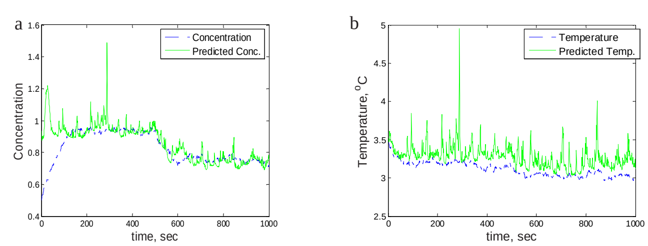

Fig. 5. *Trend plots of the actual and predicted concentration and temperature showing jitters in the estimates due to poor tuning*

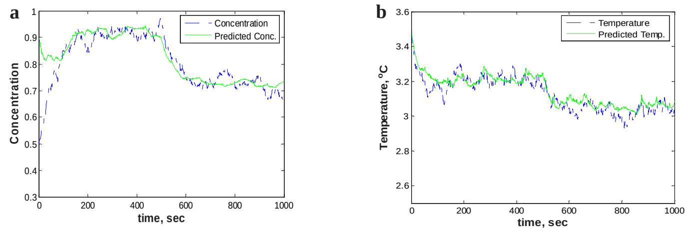

Fig. 6. *Trend plots of the actual and predicted concentration and temperature showing smooth behavior of the predicted states due to annealing of weights*

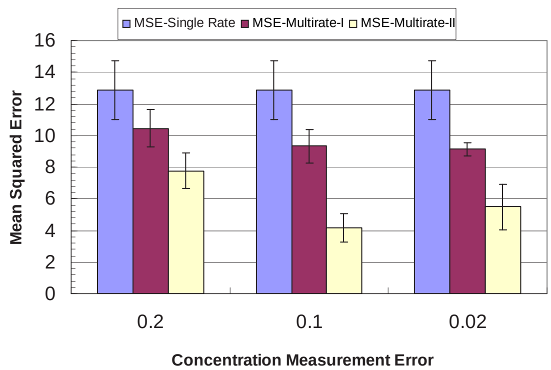

Fig. 7. *Mean Squared Error comparing the estimation performance of the multi-rate strategy with the single rate strategy*

C. Application to Multi-rate Data

The Mean Squared Error between the actual concentrations and the estimated concentrations are plotted in Figure 7. The MSE were calculated from 20 Monte Carlo simulations. The additional information made the system more observable and improved the estimates of the states. The Multiple Imputation Strategy gave the lowest MSE, however computationally by far it was the most expensive method.

D. Experimental Four Tank System

Experiment on the Four Tank system was carried out at the process control laboratory of University of Alberta. The data and the dimensions of the system can be obtained by contacting the corresponding author. A mechanistic model of the system was developed using the measured dimension of the tank. Details of the model can be found in [9]. The heights of the tanks are the states of the system. Though the measurements of all four heights were available, in applying the filter we assumed that only the heights of tank 1 and tank 2 are measured and heights of tank 3 and tank 4 are unmeasured states. The filter was tuned using the tuning rule discussed in the previous sections. The trend plots in Figure 8 shows good agreement between the true states and the estimated states by the particle filter.

V. Conclusion

The proposed tuning of Particle Filter using 'Weights vs. a priori state' plot was tested on other non-linear processes with a wide range of process noise and measurement noise. In this paper we reported partial results from one simulated and one experimental case study. This is an easy to use effective visualization tool which can help the practicing engineers to tune the Particle Filter. The Particle Filter is also extended to the Multiple Imputation framework for dealing with multi-rate data. In the presence of large process and measurement noise, the additional information from slow sampled measurements improve the estimates of the states.

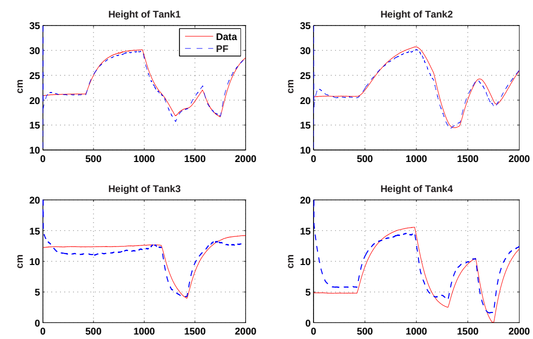

Fig. 8. *Trend plots of the heights of the Experimental Four tank system and the predicted values by Particle Filter*

References

[1] Arulampalam, M. S. and Maskell, S. and Gordon, N. and Clapp, T.," "A Tutorial on Particle Filter for Online Nonlinear/Non-Gaussian Bayesian Tracking,"*IEEE Transactions on Signal Processing*, vol. 50(2), pp.197-188",2002".

[2] Doucet, A. and Godsill, S. and Andrieu, C., "Berzuini, C. and Best, N. and Gilks, W. and Larizza, C."Dynamic conditional independence models and markov chain Monte Carlo methods,"*Journal of American Statistical Association*, vol. 92, pp.1403-1412",1997".

[3] Doucet, A., "On sequential simulation-based methods forBayesian filtering," *echnical Report. University of Cambridge, Dept. of Engineering, CUED-F-ENG-TR310. Available on the MCMC preprint service at http://www.stats.bris.ac.uk/MCMC*,1998".

[4] Doucet, A. and Godsill, S. and Andrieu, C., "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 12, pp.197-208",2000".

[5] Gordon, N. J. and Salmond, D. J. and Smith, A. F. M., "Novel approach to nonlinear or non-Gaussian Bayesian state estimation," *IEEE Proceedings on Radar and Signal Processing*, vol. 140, pp.107-113,1993".

[6] Rubin, D. B. *Multiple Imputations for nonresponse in surveys*, John Wiley and Sons Inc, NY , 1987.

[7] Henson, M. A. and Seborg, D. E., *Nonlinear Process Control*, Prentice Hall PTR, Upper Saddle River, NJ , 1997.

[8] Handschin, J. E. and Mayne, D. Q., "Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering," *International Journal of Control*, vol. 9, pp.547-559, 1969".

[9] Johansson, K. H. , "The quadruple-tank process: A multivariable laboratory process with an adjustable zero," *IEEE Transactions on control systems technology*, vol. 8 (3), pp.456-465, 2000".

[10] Liu, S. J. and Chen, R., "Sequential Monte Carlo methods for Dynamic Systems," *Journal of American Statistical Association*, vol. 93(443), pp.1032-1044, 1998".

[11] Murray, I., "Note on rejection sampling and exact sampling with the Metropolised Independence Sampler," *http://www.gatsby.ucl.ac.uk* ,2004".

[12] Smith, A. F. M. and Gelfand, A. E., "Bayesian Estimation via Bayesian Statistics without Tears: A Sampling-Resampling Perspective," *The American Statistician*, vol. 46(2), pp.84-88, 1992".

[13] Chen, W. S. and Bakshi, B. R. and Goel, P. K. and Ungarala, S., "Bayesian Estimation via Sequential Monte Carlo Sampling: Unconstained Nonlinear Dynamic Systems," *Industrial Engineering and Chemistry Research*, vol. 43, pp.4012-4025, 2004.

[14] Zaritskii, V. S. and Svetnik, V. B. and Shimelevich, L. I., "Monte Carlo technique in problems of optimal data processing," *Automation and Remote Control*, vol. 12, pp. 95-103, 1975.