Iterative Learning State Estimation for Batch Process*

Zhonggai Zhao¹, Pengcheng Qi² and Fei Liu²

Abstract—Unlike continuous processes, a batch process contains many batch runs. Considering the repetitive nature of batch processes, an iterative learning strategy is proposed to estimate the state in batch processes, where the state prediction is updated twice rather than once in conventional state estimation methods: within a batch run, the measurements are employed to update the state prediction to obtain the state estimate; along the batch dimension, the estimation performance of previous batch runs is used as a learning reference for the state estimate according to the repetitive nature. As a result, the current batch run is related with previous batch runs during the state estimation, and the information of the whole batch process is incorporated. Considering that the batch process is characterized by nonlinearity and non-Gaussianity, the particle filtering method is employed as the key algorithm for the state estimation. The effectiveness and practicability of the proposed method is indicated by its application in a beer fermentation process.

I. INTRODUCTION

Kalman filter (KF) is the most popular state estimation method for the linear process with Gaussian noise [1], [2]. Through converting the nonlinear model into a linear one by Taylor series expansion, the extended KF (EKF) method introduces KF for the state estimation in the nonlinear processes. However, the use of EKF is based on the assumption of small difference between the theoretical solution and the actual solution of the nonlinear model, so it is only suitable for the slightly nonlinear process [3]. Through unscented transformation, the unscented Kalman filter (UKF) method can provide an accurate solution for severely nonlinear processes [4]. Wang estimated biological variables by employing UKF in fed-batch fermentation processes [5]. However, the KF method and its expansions are limited to the processes with Gaussian noise. To relax the limitation, the particle filtering (PF) method is proposed, where the state space is divided into parts formed by particles according to the posterior probability density function (PDF). In PF, particles are randomly drawn from the state space to represent the PDF [6]. Due to little restriction, the PF method has become a research focus [7]. Chen introduced the PF method in the batch process, and proved its superiority over the EKF method in a benchmark batch process [8].

However, the conventional state estimation methods for the batch process, as mentioned above, are the simply duplications of those for the continuous process. Unlike the continuous process, a whole batch process may involve a number of batch runs, and the batch process is characterized by one more dimension, i.e. the batch dimension, than the continuous process. Along the time dimension, similar to the continuous process, the batch process evolves over time, but along the batch dimension, the repetitive nature relates all the batch runs. Therefore, in addition to the state space model, the repetitive nature should be taken as useful information in the state estimation in batch processes. Considering that all batch runs shared a slowly varying initial state, Zhao incorporated previous batch runs in the current state estimation by the relation of initial states [9]. However, other states except the initial state are not considered to be related with other batch runs. To be general, given a two-dimensional state space model, Zhao investigated the state estimation by characterizing the batch process with two-dimensional dynamics both within a batch run and across batch runs [10], and the previous batches and the current batch were combined for state estimation. However, in practice, it is difficult to develop an accurate two-dimensional dynamic model for the actual batch process, besides, this method suffers from large calculation.

Actually, in the control field, the repetitive nature of batch runs has gained popular attention to achieve the control performance improvement. The iterative learning control method is a widely used strategy for the control of repetitive processes. To minimize the tracking error between the reference and the plant output, the ILC adjusts the input trajectory according to the control performance of previous batch runs. Through iterative adjustment of batch runs, the input trajectory tends to converge, and error between the reference and the plant output approaches to zero. A detailed review of ILC may refer to [11], [12]. Based on the iterative learning idea, the parameter of batch processes is iteratively estimated, and the parameter estimate is updated according to the error of previous batch runs between the plant output and estimated output [13]. However, to our most, the repetitive nature is not considered in the state estimation. Motivated by the iterative learning idea, this paper proposes an iterative learning state estimation method. Setting the same initial state guess as the previous batch run, according to the repetitive nature, the state estimate obtained along the time dimension is corrected by estimation performance of the previous batch run. Therefore, the proposed iterative learning state estimation method can incorporate both the dynamic information over time and the repetitive information along

---

¹This work is supported by the National Natural Science Foundation of China (NSFC 61573169) and the Fundamental Research Funds for the Central Universities (JUSRP51407B).
²Zhonggai Zhao is with Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University, Wuxi, China (e-mail: gaizhao@jiangnan.edu.cn).
³Pengcheng Qi (email: weinchengpeng@163.com) and Fei Liu (email: fliu@jiangnan.edu.cn) are both with Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University, Wuxi, China.
the batch dimension in the state estimation of the current batch run.

II. STATE ESTIMATION WITH PARTICLE FILTER

In this section, the following state space model is concerned:

\[
\begin{align*}
  x_k &= f(x_{k-1}) + \omega_{k-1} \\
  y_k &= h(x_k) + \nu_k
\end{align*}
\]  

where the state \( x_k \) at the \( k \)th sampling instant transits according to the mapping function \( f(\cdot) \) plus the process noise \( \omega_{k-1} \), and the measurement \( y_k \) is obtained by the measuring function \( h(\cdot) \) and the measurement noise \( \nu_k \). Assuming that all available measurements up to the \( k \)th sampling instant are denoted as \( Y_k = \{y_1,y_2,\cdots,y_k\} \), according to the Bayesian filtering algorithm, the state PDF is estimated through two steps including a prediction step and an update step [14]. To make the Bayesian filtering algorithm tractable, two assumptions are made as follows:

**Assumption 1** The state follows the first-order Markov process, i.e.:

\[
p(x_k|X_{k-1},Y_{k-1}) = p(x_k|X_{k-1})
\]  

**Assumption 2** The measurements are conditionally independent given the state:

\[
p(y_k|X_{k},Y_{k-1}) = p(y_k|X_k)
\]  

where \( X_k = \{x_1,x_2,\cdots,x_k\} \).

The two steps are implemented as follows:

1. **Prediction step:**

\[
p(x_k|Y_{k-1}) = \int p(x_k|x_{k-1}|Y_{k-1})dx_{k-1} = p(x_k|x_{k-1})p(x_{k-1}|Y_{k-1})dx_{k-1}
\]  

where \( p(x_k|x_{k-1}) \) is determined by the state transition model and \( p(x_{k-1}|Y_{k-1}) \) is the posterior PDF of the previous state.

2. **Update step:**

\[
p(x_k|Y_k) = \frac{p(y_k|x_k)p(x_k|Y_{k-1})}{p(y_k|Y_{k-1})}
\]  

By incorporating the measurement \( y_k \), \( p(x_k|Y_{k-1}) \) is updated to obtain the posterior PDF \( p(x_k|Y_k) \). Iteratively computing (4) and (5) constitutes the recursive Bayesian estimation algorithm, which may obtain the state posterior PDF.

By using Monte Carlo method, PF can realize recursive Bayesian filtering on the nonlinear and non-Gaussian model. To solve the intractable integration, the PDF is empirically expressed by a weighted sum of \( N \) particles drawn from the PDF \( p(x_k|Y_k) \) as [15]:

\[
p(x_k|Y_k) = \frac{1}{N} \sum_{m=1}^{N} \delta(x_k - x^m_k)
\]  

Due to the difficulty of directly drawing particles from the unknown true posterior PDF, an easy-to-implement PDF, the so-called importance distribution, is proposed instead of the true posterior PDF to generate particles. \( p(x_k|x_{k-1}) \) is usually selected as the importance distribution, so the posterior PDF can be represented as [16]:

\[
p(x_k|Y_k) = \sum_{m=1}^{N} w^m_k \delta(x_k - x^m_k)
\]  

where \( w^m_k \) is the associated weight of the particle, and it can be derived as [17]:

\[
w^m_k \propto p(y_k|x^m_k)
\]  

It should be noted that the weights will be gradually distortedly distributed with the increasing number of samples used for state estimation, eventually most particles are obtained with zero weights, which may cause an undesirable performance. To pay more emphasis on the important particles, the particles are resampled, by which those with high weights are multiplied and those with low weights are discarded. After resampling, new particles are reproduced with the same weight \( 1/N \) [18].

In summary, a generic PF algorithm is illustrated as follows:

1. **Step (a) Initialization:** the initial particles are generated according to the prior information, and set \( k = 1 \).

2. **Step (b) Prediction:** based on the state space model, the prior particles are generated from \( p(x_k|x_{k-1}) \).

3. **Step (c) Weights calculation:** the weights of the particles are calculated and then normalized using the current measurements.

4. **Step (d) Resampling:** draw the posterior particles from the prior particles, and reset all the weights to \( 1/N \).

5. **Step (e) Estimate state through:** \( \hat{x}_k = \frac{1}{k} \sum_{m=1}^{N} x^m_k \), and set \( k = k + 1 \). Go back to Step (b).

III. PROBLEM DESCRIPTION

Within a batch run, a batch process is similar as the continuous process which evolves over time. However, a typical batch process contains many batch runs. Therefore, for the whole batch process, besides the dynamics within a batch run, the batch process evolves over batch with repetitive trajectory, and the repetitive nature along the batch dimension is of the same importance as the dynamics over time. In the batch process, the conventional state estimation methods are the replication of those methods in continuous processes, and the repetitive nature is discarded and the state update is confined within a single batch run. As a result, only the information along the time dimension rather than that along the batch dimension is considered in the state estimation. However, due to the repetitive nature, the previous batch run can provide useful information for the next one. The dynamics of batch processes is illustrated in Figure 1. Obviously, in each batch run, each state will vary along the time dimension, and each state has a trajectory. Comparing with different batch runs, the same initial state will result in a similar state trajectory, which means that the state exhibits repetitive nature along the batch dimension.
IV. ITERATIVE LEARNING ESTIMATION IN BATCH PROCESSES

To illustrate the application of repetitive nature in the batch process, this section reviews the iterative learning control method and the iterative learning parameter estimation algorithm. The principle of iterative learning strategy is to make adjustment on the current batch run by learning the previous batch runs.

A. Iterative learning control in batch processes

The iterative learning control (ILC) method employs the iterative learning concept to design the controller in batch processes [11]. By employing a control trajectory, a plant output is obtained in a batch run. Based on the error between the output reference and the plant output in the previous batch run, ILC adjusts the control trajectory, and then uses it for the current batch to minimize the tracking error of the output reference. It has been proved that the plant output may converge to the reference trajectory with the growing number of batch runs [12]. The reference output trajectory is written as:

$$Y_{\text{ref}} = [y^1_{\text{ref}}, y^2_{\text{ref}}, \ldots, y^N_{\text{ref}}]$$

where $N$ is the duration of batch run. The error $e_k$ between the output reference and the plant output in the $k$th batch run is represented as

$$e_k = Y_{\text{ref}} - Y_k$$

where $Y_k$ is the plant output of the $k$th batch run. The objective of ILC is to minimize the error $e_k$ such that

$$\lim ||e_k|| = 0$$

According to the ILC method, the control trajectory of the next batch run is updated according to the proportional error as

$$U_{k+1} = U_k + He_k$$

where $H$ is the proportional matrix.

B. Iterative learning parameter estimation in batch processes

The iterative learning parameter estimation is similar to the ILC method, and the parameter estimate rather than the control trajectory is updated along the batch dimension [13]. Consider a system as

$$\begin{align*}
\dot{x} &= f(x,u,\theta) \\
y &= g(x,u,\theta)
\end{align*}$$

The objective of parameter estimation is to derive the parameter $\theta$ through the measurement $y$ and input $u$. The iterative learning parameter estimation method achieves minimization of the error between the estimated output and the plant output through the adjustment of parameter estimation. The input trajectory of each batch is computed according to the model whose parameter is estimated previously, and within a batch run the input trajectory is kept unchanged. Therefore, in this method, the input trajectory $U_k$ is derived based on the parameter estimate $\theta_{k-1}$, and then the output $\hat{Y}_k$ is estimated by $\theta_{k-1}$ and $U_k$. Using the estimation error, the parameter estimate $\hat{\theta}_k$ is obtained. The estimated output of batch $k$ is represented as

$$\hat{Y}_k = [y^1_k, y^2_k, \ldots, y^N_k]$$

and the estimation error is

$$E_k = Y_k - \hat{Y}_k$$

Then, the parameter estimate is adjusted to minimize $||E_k||$ as

$$\theta_{k+1} = \theta_k + He_k$$

So the estimation procedure can be summarized below:

Step 1 Give the guess initial parameter $\theta_0$, and set $k = 1$;  
Step 2 Based on $\theta_{k-1}$ and the model, the input trajectory $U_k$ is updated;  
Step 3 Predict output of batch $k$, $\hat{Y}_k$;  
Step 4 Update the parameter estimate $\theta_k$ according to (16) at the end of batch $k$;  
Step 5 Return to Step 2.

V. ITERATIVE LEARNING STATE ESTIMATION

An iterative learning state estimation method is proposed in this section. Consider a batch process as follows:

$$\begin{align*}
x_{i,k} &= f(x_{i,k-1}) + \omega_{i,k-1} \\
y_{i,k} &= h(x_{i,k}) + v_{i,k}
\end{align*}$$

where the measurement error $v_{i,k}$ satisfies Gaussian distribution $N(0, \sigma^2 I)$. Next we will make an assumption on the batch process.

**Assumption 3** The batch process is characterized by repetitive nature, and the repetitive nature refers to that the same initial state may result in the same state trajectory over time for each batch run, and the same initial guess of state may lead to the similar estimation trajectory. According to the repetitive nature, the states for the current batch run follow an almost same unknown trajectory as for the previous batch run. Therefore, the estimation error of measurements may be employed for the state update in the next batch run.
A. Iterative learning state estimation strategy

In the $i$th batch run, given a guessed initial state $\hat{x}_{i,0}$, the state estimate trajectory can be written as

$$\hat{X}_i = [\hat{x}_{i,0}, \hat{x}_{i,1}, \hat{x}_{i,2}, \ldots, \hat{x}_{i,N}]$$

(18)

The estimated measurements of this batch run are represented as

$$\hat{Y}_i = [\hat{y}_{i,1}, \hat{y}_{i,2}, \ldots, \hat{y}_{i,N}]$$

(19)

where the estimate $\hat{y}_{i,k}$ is derived as:

$$\hat{y}_{i,k} = h(\hat{x}_{i,k})$$

(20)

So, the estimation errors of measurements are computed as

$$E_i = Y_i - \hat{Y}_i = [e_{i,1}, e_{i,2}, \ldots, e_{i,N}]$$

(21)

According to the state space model of batch processes, the optimal estimates of states should satisfy

$$\hat{X}_i = \min_{X_i} E_i$$

(22)

Due to the poor initial state guess, the optimal estimates cannot be achieved. The estimation error of measurements can be employed to adjust the state estimates in the next batch run as follows:

$$\hat{X}_{i+1}^{Cor} = \hat{X}_{i+1} + PE_i$$

(23)

where $P$ is the proportional matrix, and $\hat{X}_{i+1}$ is the state estimate only using the measurements of the $(i+1)$th batch run. Figure 2 shows the iterative learning state estimation method. Observe that the generic particle filter is a closed-loop method for the state estimation along the time dimension, however, it is open-loop along the batch dimension. Based on the adjustment using the measurements in the generic particle filter, the proposed method also incorporates the repetitive nature in the state estimation, and it can make the state estimation closed-loop both along the batch dimension and along the time dimension. Through two-dimensional adjustment, a better estimation performance will be achieved in batch processes than the generic particle filter which adjusts the estimation along only one dimension.

B. Determination of initial state guess

Obviously, different batch runs sharing the same initial state guess is a premise for the proposed iterative learning state estimation method. It is difficult to update the state estimation according to the estimation error of the batch runs resulted by different initial state guess, so the initial state estimation should be guessed as same as the previous batch run to correct the state estimation. However, the initial state guess of previous batch run should also be updated. Next we will take the first batch run as an example to illustrate the determination of initial state guess.

In the first batch run, the initial state is poorly guessed as $\hat{x}_{1,0}$, after this batch run is complete, the state estimation trajectory is obtained as follows

$$\hat{X}_1 = [\hat{x}_{1,0}, \hat{x}_{1,1}, \hat{x}_{1,2}, \ldots, \hat{x}_{1,N}]$$

(24)

So the estimation error of measurements is derived by the state estimate as

$$E_1 = [e_{1,1}, e_{1,2}, \ldots, e_{1,N}]$$

(25)

For the second batch run, if the initial state guess is also set to $\hat{x}_{1,0}$, the state estimate trajectory $\hat{X}_1$ is obtained, and the estimation error is similar as the first batch run, so the estimation error of the first batch run is used to update the state estimates. However, after the state estimation is updated along the batch dimension, the state estimate trajectory $\hat{X}_1^{Cor}$ is not caused by the initial state guess $\hat{x}_{1,0}$. In other words, the initial state guess $\hat{x}_{1,0}$ should also be updated along the batch dimension. Due to no measurements for the initial state, in the second batch run, the initial state guess which results in $\hat{X}_1^{Cor}$ can be derived by the forward-backward smoothing method. The state smoothing is achieved by marginalizing the joint distribution $p(x_{i,j}, x_{i,j+1}|Y_i)$ as follows:

$$p(x_{i,j}|Y_i) = \int p(x_{i,j}, x_{i,j+1}|Y_i) dx_{i,j+1}$$

$$= \int p(x_{i,j}|x_{i,j+1}, Y_i) p(x_{i,j+1}|Y_i) dx_{i,j+1}$$

$$= \int p(x_{i,j}|Y_{i-1}) \int p(x_{i,j+1}|Y_i)p(x_{i,j+1}|x_{i,j}) dx_{i,j+1}$$

(26)

Therefore, $p(x_{i,j}|Y_i)$ is approximated as

$$p(x_{i,j}|Y_i) = \sum_{n=1}^{N} w_{i,j}^n \delta(x_{i,j} - x_{i,j}^n)$$

(27)

where the smoothing weight $w_{i,j}^n$ is derived as:

$$w_{i,j}^n = w_{i,j}^n \sum_{m=1}^{N} w_{i,j+1}^m \frac{p(x_{i,j+1}^m|\theta)}{\sum_{k=1}^{N} w_{i,j+1}^k p(x_{i,j+1}^k|\theta)}$$

(28)

Employing the particles generated by the forward filtering method and starting from $w_{i,j}^n$, the forward-backward smoothing algorithm recursively derives the smoothing weights backward, and then the smoothed value is calculated. It should be noted that all the particles should be generated according to $\hat{X}_1^{Cor}$. 

Fig. 2. Iterative learning state estimation for batch process
Therefore, the proposed iterative learning state estimation method can be summarized as Figure 3. In the first batch run, the state estimation and the estimation error of measurements are performed according to the generic particle filter. For the other batch run, the initial state guess is set as same as that of the previous batch run, then the state and the measurement error are estimated by employing the generic particle filter and the iterative learning method. After this batch run is complete, its initial state guess is re-derived by the forward-backward smoothing algorithm, and this smoothed initial state guess may be treated as the initial state guess of the next batch run.

VI. SIMULATION

In this section, to evaluate the performance of the proposed approach, the root mean square estimation error (RMSE) [19] of each batch is employed as performance index:

$$RMSE = \sqrt{\frac{1}{K} \sum_{k=1}^{K} (x_{i,k} - \hat{x}_{i,k})^2}$$

(29)

where the estimation results with smaller RMSE indicates the better estimation performance.

The state space model for a beer fermentation process is represented as [20]:

$$S_{i,k} = \frac{-\mu T_c S_{i,k-1} + \mu}{K_x + S_{i,k-1}} + \omega_{i,k-1}$$

$$X_{i,k} = \frac{\mu T_c X_{i,k-1} + X_{i,k-1}}{K_x + S_{i,k-1}} - b T_c X_{i,k-1} + X_{i,k-1} + \omega_{i,k-1}$$

(30)

where $S_{i,k}$, $X_{i,k}$ are the states representing the substrate concentration, biomass concentration, respectively, $y_{i,k,S}$ and $y_{i,k,X}$ are the measurements, and $T_c$ is the sampling interval. The parameters $\mu$, $b$, $K_s$, $K_x$ are equal to 0.78, 0.058, 0.0251, 0.0252 and 0.7464, respectively. The initial state is set to $x_{1,1} = [60, 2]^T$, and the sampling interval is set to 0.01h. The initial state transits according to Gaussian distribution $N(0, 0.00001)$, and measurement noise follows $N(0, 0.01)$. Considering the initial state guess $\hat{x}_{1,1} = [63, 0.8]^T$ and 30h for each batch run, and 60 batch runs are simulated. PF is employed to estimate states by 50 particles, the state transition error are estimated by employing the generic particle filter. For each batch run, the initial state guess is set as same as that of the previous batch run, and 60 batch runs are simulated. PF is employed to estimate states by 50 particles, the state transition noise and measurement noise are guessed to follow Gaussian distribution $N(0, 0.00001)$ and $N(0, 0.01)$, respectively.

Setting the proportional matrix $[0.005 \ 0; 0 \ 0.001]$, the proposed iterative learning state estimation method is applied in the beer fermentation process. The estimation results are shown in Figure 4 and Figure 5.

In the first batch, the state estimation can only use the information along the time dimension, so the estimation performance is undesirable. Obviously, the estimation error tends to converge with increasing batch runs used for the state estimation, which is caused by incorporating more and more batch-to-batch information in the state estimation. As a result, the state estimate trajectory almost coincides with the true state trajectory in the 60th batch run. Therefore, through iterative learning along the batch dimension, the state estimates may approach the true values.

VII. CONCLUSION

The conventional methods estimate the state within a single batch run for batch processes. This paper proposes an iterative learning state estimation algorithm for the batch process, and the information from the previous batches can be used to update the state estimate in the current batch. In the proposed method, the estimation error between the estimated measurement and the practical measurement is treated as a probe, and the state estimate of the next batch run can learn from the information of the probe and then be updated along the batch dimension. Considering the batch process feature, the Bayesian recursive estimation algorithm along with particle filter are employed as the key algorithms for the filtering. The simulation proves that the estimation results get better as the number of batch increases.

REFERENCES


Fig. 4. State estimation results in beer fermentation process

Fig. 5. State estimation error in beer fermentation process