An Enhanced Just-in-Time Learning Methodology for Process Modeling

Cheng Cheng*, Yoshihiro Hashimoto†, Min-Sen Chiu*

*Department of Chemical and Environmental Engineering
National University of Singapore, Singapore
e-mail: checms@nus.edu.sg

†Department of Systems Management and Engineering
Nagoya Institute of Technology, Japan
e-mail: hasimoto@manage.nitech.ac.jp

Abstract

A new just-in-time learning methodology for nonlinear process modeling is developed in this paper. In the proposed method, both distance measure and angle measure are used to evaluate the similarity between data, which is not exploited in the conventional methods. In addition, parametric stability constraints are incorporated into the proposed method to address the stability of local models. Furthermore, a new procedure of selecting the relevant data set is proposed. The proposed methodology is illustrated by a case study of modeling a polymerization reactor. The adaptive ability of the just-in-time learning is also evaluated.

1 Introduction

Mathematical models are often required for purposes of process modeling, control, and fault detection and isolation. However, most chemical processes are multivariable and nonlinear in nature, and their dynamics can be time-varying. Thus first-principle models are often unavailable due to the lack of complete physicochemical knowledge of chemical processes. An alternative approach is to develop data-driven methods to build model from process data measured in industrial processes.

Traditional treatments of the data-driven modeling methods focus on global approaches, such as neural networks, fuzzy set and other kinds of non-linear parametric models [1]. However, when dealing with large sets of data, this approach becomes less attractive to deal with because of the difficulties in specifying model structure and the complexity of the associated optimization problem, which is usually highly non-convex. Another fundamental limitation of these models is their inability to extrapolate accurately once the information is outside the range of the data used to generate or train the model. On the other hand, the idea of local modeling is to approximate a nonlinear system with a set of relatively simple local models valid in a certain operating regimes. The T-S fuzzy model [2] and neuro-fuzzy network [1], [3] are well-known examples of local modeling approach. However, the local modeling approaches suffer the drawbacks of requiring a priori knowledge to determine the partition of operating space and complicated training strategy to determine the optimal parameters of the models.

To alleviate the aforementioned difficult problems, Just-in-Time Learning (JITL) [4] was recently developed as an attractive alternative for modelling the nonlinear systems. It is also know as instance-based learning [5], locally weighted model [6], lazy learning [7], or model-on-demand [8] in the literature. However, distance measures are overwhelmingly used in the previous work [5-8]. Complementary information available from angular relationship has not been exploited. In addition, the stability of local model is not addressed in the previous work, resulting in unstable local models even when the process is stable. In this paper, by incorporating the stability constraints, an enhanced JITL methodology based on both angle measure and distance measure is proposed. In addition, a new procedure of selecting the relevant data set is proposed. One case study of modeling an isothermal free-radical polymerization reactor is presented to evaluate the efficiency of the proposed method. Online adaptive ability of JITL is also studied.

2 Just-in-Time Learning

Comparing with the traditional modeling method, JITL has no standard learning phase. It merely gathers the data and stores them in the database and the computation is not performed until a query data arrives. In the inquiry stage, there are three main steps: (1) the relevant data samples in the database are searched to match the query data by some nearest neighborhood criterion; (2) a local
model is built based on the relevant data; (3) the local model predicts the model output based on the current query data. The local model is then discarded right after the answer is obtained. When the next query data comes, a new local model will be built based on the aforementioned procedure. It should be noted that JITL model is only locally valid for the operating condition characterized by the current query data. In this sense, JITL constructs local approximation of the dynamic systems. Therefore, a simple model structure can be chosen, e.g. an ARX model. This feature allows JITL to be conveniently incorporated into the model based controller designs or process monitoring methods.

Another advantage of JITL is its inherently adaptive nature, which is achieved by storing the current measured data into the database [1]. In contrast, neural network and neuro-fuzzy models require model update from scratch, namely both network structure (e.g. the number of hidden neurons in the former case and the number of the fuzzy rules in the latter) and model parameters may need to be changed simultaneously. Evidently, this procedure is not only too expensive from computational point of view, but also it will interrupt the plant operation, if these models are used for other purposes like model based controller design.

To facilitate the ensuing developments, the algorithm of JITL is described next. Suppose that a database consisting of process data \((y_i, x_i)_{i=1-N}\), \(y_i \in R\), \(x_i \in R^m\), is collected. It is worthwhile pointing out here that the vector \(x_i\) is formed by the past values of both process input and process output as typically required in building an ARX model. Given a specific query data \(x_q\), whose elements are identical to those defined for \(x_i\), the objective of JITL is to predict the model output \(\hat{y}_q\) according to the known database. In the literature, distance measure \(d(x_q, x_i)\), e.g. Euclidean norm \(d(x_q, x_i) = \|x_q - x_i\|_2\), is overwhelmingly used to select the relevant data set from the database by evaluating the relevance (or similarity) between the query data \(x_q\) and \(x_i\) in the entire database, i.e. smaller value of distance measure indicates greater similarity between \(x_q\) and \(x_i\). To do so, a weight \(w_i\) is assigned to each data \(x_i\) and it is calculated by the kernel functions, \(w_i = \sqrt{K(d(x_q, x_i)/h)}\), where \(h\) is the bandwidth of the kernel function \(K\), which is usually computed using a Gaussian function, \(K(d) = e^{-d^2}\). If a linear model is employed to predict the model output \(\hat{y}_q\), the query answer is [6]:

\[
\hat{y}_q = x_q^T (P^TP)^{-1} P^Tv
\]  

where \(P = W\Phi\), \(v = Wy\), \(W \in R^{N \times N}\) is a diagonal matrix with diagonal elements \(w_i\), \(\Phi \in R^{N \times n}\) is the matrix with every row corresponding to \(x_i^T\), and \(y = [y_1, y_2, \ldots, y_N]^T\).

In JITL, PRESS statistic [9] is used to perform leave-one-out cross validation to assess the generalization capability of the local linear model [6]. In doing so, the optimal value of \(h\), \(h_{opt}\), and the model output \(\hat{y}_q\) can be determined as follows: for a given \(h\), eq. (1) is used to compute the predicted output and the validation error is calculated by the leave-one-out cross validation test. This procedure repeats for a number of \(h\) and \(h_{opt}\) is chosen as the one with the smallest validation error. With \(h_{opt}\) known, the optimal model prediction is then computed by eq. (1) for the current query data \(x_q\).

As mentioned above, each local model obtained by JITL is only locally valid around the query data, therefore simple model structure can be chosen as local model at each query point. For a dynamic system, ARX model can be chosen as the local model for JITL. The ARX model is given as follows:

\[
\hat{y}(k) = z^T(k-1) \Psi
\]  

where \(\hat{y}(k)\) is the model output at the \(k\)-th sampling instant, \(z(k-1)\) is the regression vector, and \(\Psi\) is the model parameter vector as given by:

\[
z(k-1) = [y(k-1), u(k-n_y), u(k-n_d-1), \ldots, u(k-n_d-n_y)]^T
\]  

\[
\Psi = [\psi_1, \ldots, \psi_{n_y}, \psi_{n_y+1}, \ldots, \psi_{n_y+n_u}]^T
\]  

where \(n_y\) and \(n_u\) are integers related to the model’s order, and \(n_d\) is the process time delay. By comparing Eqs. (1) and (2), it is evident that the local model parameters obtained by JITL method is computed as \((P^TP)^{-1} P^Tv\). Furthermore, the vector \(x_i\) in the database and query data \(x_q\) have the same input and output variables as those defined for \(z(k-1)\). For example, for a first-order model with \(n_y = n_u = 1\) and \(n_d = 0\), the database \((y_i, x_i)_{i=1-N}\) is given by \([y(k), y(k-1), u(k-1)]_{k=1-N}\), where \(y(k)\) and \(u(k)\) are the process output and input data collected at the \(k\)-th sampling instant in the identification test. Similarly, in the prediction phase, the query data \(x_q\) at the \((k-1)\)-th sampling instant is arranged in the form of \([y(k-1), u(k-1)]^T\) as the input to the JITL algorithm, from which the predicted output at the next sampling instant \(\hat{y}(k)\) can be computed.
### 3 New JITL Methodology

In the preceding section, it is evident that conventional JITL methods only use distance measure to evaluate the similarity between two data samples. However, considering data observations as points in space leads to two types of measures: distance between two points, and angles between two vectors. Therefore, by incorporating angular relationship in the formulation of JITL, it is expected that the resulting method should give more accurate prediction over the existing methods. Therefore, by using both distance measure and angle measure the following similarity criterion is proposed:

\[
s_j = \gamma \sqrt{e^{-d^2(x_i, x_j)}} + (1 - \gamma) \cdot \cos(\theta_j)
\]

where \(\gamma\) is a weight parameter and is constrained between 0 and 1, \(\theta_j\) is the angle between \(\Delta x_q\) and \(\Delta x_i\), \(\Delta x_i = x_i - x_{i-1}\), \(\Delta x_q = x_q - x_{q-1}\), and \(s_j\) is the similarity number bounded between 0 and 1.

Another shortcoming of the existing JITL methods is that the weights for all the data in the database are determined by the proposed similarity criterion eq. (2), as determined by the proposed similarity criterion eq. (5), are used in the regression. Specifically, two parameters \(k_{\text{min}}\) and \(k_{\text{max}}\) are chosen such that only the relevant data sets formed by the \(k_{\text{min}}\)-th relevant data to the \(k_{\text{max}}\)-th relevant data are used in the regression. Usually \(k_{\text{min}}\) and \(k_{\text{max}}\) are much smaller than the number of data in the entire database, therefore the computational burden is significantly reduced compared to conventional JITL methods.

Lastly, it is noted that stability of local model is not taken into account in the conventional JITL methods. As a result, some local models generated by JITL may be unstable even when the database employed and the query points are from the stable process. This feature is not desirable, especially when these models are to be employed in the controller design. Thus, for a stable system, the parameters of each ARX model obtained by JITL need to be verified whether they satisfy the stability constraint or not. In case the parameters fail to satisfy the stability constraint, a constrained optimization problem can be incorporated into JITL to obtain the stable model. Similar procedure can be devised to obtain an unstable local model when the process of interest is unstable. The parametric stability constraints imposed on \(\Psi\) will be discussed in the ensuing development.

The detailed algorithm of the proposed JITL methodology is described as follows. Given a database \((y_j, x_j)_{j=1}^N\), the parameters \(k_{\text{min}}\) \(k_{\text{max}}\), and weight parameter \(\gamma\), and a query data \(x_q\):

**Step 1:** Compute the distance and angle between \(x_q\) and each data \((y_j, x_j)\):

\[
d_l = \|x_q - x_l\|_2
\]

\[
\cos(\theta_j) = \frac{\Delta x_q^T \Delta x_l}{\|\Delta x_q\| \|\Delta x_l\|_2}
\]

If \(\cos(\theta_j) \geq 0\), compute the similarity number \(s_j\):

\[
s_j = \gamma \sqrt{e^{-d^2}} + (1 - \gamma) \cdot \cos(\theta_j)
\]

If \(\cos(\theta_j) < 0\), the data \((y_j, x_j)\) is discarded.

**Step 2:** Arrange all \(s_j\) in the descending order. For \(l = k_{\text{min}}\) to \(k_{\text{max}}\), the relevant data set \((y_j, x_j)\) are constructed by selecting \(l\) most relevant data \((y_j, x_j)\) corresponding to the largest \(s_j\) to the \(l\)-th largest \(s_j\). Denote \(W_l \in R^{lxl}\) a diagonal weight matrix with diagonal elements being the first \(l\) largest values of \(s_j\), and calculate:

\[
P_l = W_l \Phi_l
\]

\[
v_l = W_l y_l
\]

The local model parameters are then computed by:

\[
\Psi_l = (P_l^T P_l)^{-1} P_l^T v_l
\]

where \((P_l^T P_l)^{-1}\) is calculated by SVD method. Next, the leave-one-out cross validation test is conducted and the validation error is calculated by (Myers, 1990):

\[
e_l = \frac{1}{N} \sum_{j=1}^N \frac{(y_j - \phi_j^T (P_l^T P_l)^{-1} P_l^T v_l)^2}{1 - \phi_j^T (P_l^T P_l)^{-1} p_j}
\]

where \(y_j\) is the \(j\)-th element of \(y_l\), \(\phi_j^T\) and \(p_j\) are the \(j\)-th row vector of \(\Phi_l\) and \(P_l\), respectively.

**Step 3:** According to the validation errors, the optimal \(l\) is determined by:

\[
l_{\text{opt}} = \arg \min_{l} e_l
\]

**Step 4:** Verify the stability of local model built by the optimal model parameters \(\Psi_{l_{\text{opt}}}\). Because JITL
constructs the local approximation of the dynamic systems, only the stability constraints of first-order and second-order models are given as follows:

First-order model:

\[ -1 < \psi_1 < 1 \]  
(14)

Second-order model:

\[
\begin{bmatrix}
1 & 1 \\
-1 & 1
\end{bmatrix}
\begin{bmatrix}
\psi_1 \\
\psi_2
\end{bmatrix} < \begin{bmatrix} 1 \\ 1 \end{bmatrix} \]  
(15)

\[-1 < \psi_2 < 1 \]  
(16)

If \( \Psi_{\text{opt}} \) satisfies the stability constraint, the predicted output for query data is computed as

\[
(\hat{y}_q)_{\text{opt}} = x_q^T \Psi_{\text{opt}} \]  
(17)

Otherwise, \( \Psi_{\text{opt}} \) is used as the initial value in the following optimization problem subject to appropriate stability constraint, Eq. (14) or Eqs. (15) and (16).

\[
\min_{\Psi} \| P_{\text{opt}} \Psi - v_{\text{opt}} \|_2 \]  
(18)

With the optimal solution \( \Psi^*_{\text{opt}} \) obtained from Eq. (18), the predicted output for query data is then calculated as \( x_q^T \Psi^*_{\text{opt}} \).

Step 5: When the next query data comes, go to step 1.

### Example

Considering a polymerization reactor, in which an isothermal free-radical polymerization of methyl methacrylate is carried out using azo-bis-isobutyronitrile as initiator and toluene as solvent. The output variable is the number average molecular weight (NAMW), \( y \), and the input variable is the inlet initiator flow rate \( F_i \). This reactor can be described by the following equations [10]:

\[
\frac{dC_m}{dt} = -(k_p + k_{in}) C_m P_0 + \frac{F(C_{in} - C_m)}{V} \]  
(19)

\[
\frac{dC_i}{dt} = -k_i C_i + \frac{F(C_{in} - FC_i)}{V} \]  
(20)

\[
\frac{dD_0}{dt} = (0.5k_T + k_f) P_0^2 + k_{in} C_m P_0 - \frac{FD_0}{V} \]  
(21)

\[
\frac{dD}{dt} = M_m (k_p + k_{in}) C_m P_0 - \frac{FD}{V} \]  
(22)

where \( y = \frac{D_l}{D_0} \), \( P_0 = \left[ \frac{2f^* k_i C_i}{k_T + k_f} \right]^{0.5} \). The model parameters and steady-state operating condition can be found in [10].

The database is generated by adding a periodic pseudo-random sequence signal with a switching probability of 0.25 to the process input \( F_i \). Using a sampling time of 0.03h, 2500 input/output data as shown in Fig. 1 are simulated to build the database. The data are scaled by their respective nominal values, i.e. \( \tilde{u} = \frac{u - u_0}{0.01} \) and \( \tilde{y} = \frac{y - y_0}{10000} \). The scaled output range is \( \tilde{y} \in [-1.26 \ 1.72] \).

To proceed with the proposed JITL algorithm, \( x \) is chosen to be the second order ARX model and set \( k_{\text{min}} = 8 \) and \( k_{\text{max}} = 60 \). To determine the optimal \( \gamma \), Table 1 lists the mean-squared-error (MSE) of the validation test for three values of \( \gamma \). As can be seen, the error decreases initially as \( \gamma \) decreases from 0.95 to 0.90, after which the error starts to increase. Therefore, the optimal \( \gamma \) is chosen to be 0.90. Based on the identical database, Conventional JITL is also considered for comparison purpose. Table 1 and Figures 2 and 3 compare the predictive performance of these two methods. It is evident that the proposed JITL methodology outperforms the conventional JITL, which is based on the distance measure alone.

**Table 1. Performance of JITL for various values of \( \gamma \)**

<table>
<thead>
<tr>
<th>Distance measure</th>
<th>( \gamma = 0.95 )</th>
<th>( \gamma = 0.90 )</th>
<th>( \gamma = 0.85 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 5.597 \times 10^{-4} )</td>
<td>( 2.461 \times 10^{-4} )</td>
<td>( 1.253 \times 10^{-4} )</td>
<td>( 1.320 \times 10^{-4} )</td>
</tr>
</tbody>
</table>
To illustrate that the proposed method can be made adaptive by simply adding the process data online to the database, step changes in $F_i$ are introduced such that the output variable is away from the normal operating space of current database. Two scenarios are studied: non-adaptive and adaptive JITL. In the former, the database is fixed, whereas in the latter the database is constantly updated by adding the new available input-output data to the database at each sampling time. Simulation results in Fig. 4 show that significantly smaller modeling error is achieved by adaptive JITL as compared with non-adaptive JITL.

5 Conclusion

In this paper, a new JITL methodology for nonlinear process modeling is proposed. Simulation study of a polymerization reactor is used to evaluate the efficiency of the proposed method. It is also demonstrated that JITL can be made adaptive online by simply adding the new data to the database.

Reference


