



Adaptive decentralized PID controllers design using JITL modeling methodology

Xin Yang^a, Li Jia^{b,*}, Min-Sen Chiu^{a,*}

^a National University of Singapore, Department of Chemical and Biomolecular Engineering, Singapore 117576, Singapore

^b Shanghai Key Laboratory of Power Station Automation Technology, Department of Automation, College of Mechatronics Engineering and Automation, Shanghai University, Shanghai 200072, China

ARTICLE INFO

Article history:

Received 24 May 2011

Received in revised form 12 April 2012

Accepted 31 May 2012

Available online 20 July 2012

Keywords:

Decentralized PID controller

Lyapunov method

Multivariable systems

Just-in-Time Learning

ABSTRACT

In this paper, an adaptive decentralized PID design is developed for multivariable systems. In the proposed design, the controller parameters are adjusted by an updating algorithm derived based on the Lyapunov method such that the predicted tracking error converges asymptotically. Toward this end, the Just-in-Time Learning method is incorporated into the proposed design to provide the information required for the updating algorithm. Simulation results illustrate that the proposed design achieves better control performance than its corresponding benchmark designs reported in the literature.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

As the most widely used controller in the process industries, the conventional Proportional-Integral-Derivative (PID) controller design techniques are mostly based on linear process models. However, most chemical and biochemical processes exhibit nonlinear and multivariable behavior. As a result, the conventional PID controller may not perform well when applied to nonlinear systems that operate over a wide range of conditions. To improve the control performance for the nonlinear system, various adaptive PID controller designs have been developed using adaptation algorithms based on different methods [1–17].

In previous studies for adaptive PID controller designs, empirical models like neural network (NN) models have received much attention. This is because NN models have the capability to approximate any nonlinear function to arbitrary degree of accuracy. However, one fundamental limitation of these empirical models is that it is difficult for them to be updated on-line when the process dynamics move away from the nominal operating space. To alleviate this problem, Just-in-Time Learning (JITL) technique is an attractive alternative for modeling the nonlinear processes [18–24]. In this modeling framework, local model such as lower-order ARX models at each sampling instant is built based on the query data and relevant data selected from reference database based on similarity measures. In the early studies on the JITL technique, distance metric was predominantly used to evaluate similarities between the two

data samples [18–22]. To improve JITL's predictive performance, an enhanced JITL algorithm using a similarity measure by combining the conventional distance metric with an additional angle metric was proposed [23]. Recently, a correlation-based JITL (Co-JITL) method was proposed for soft sensor design [24]. In the Co-JITL method, the relevant data used for local modeling are selected based on the correlation measure by the Q and T^2 statistics. By using the enhanced JITL method, several adaptive PID controller design methods were developed [25–27]. In [27], an adaptive PID controller design was developed based on an updating algorithm derived by the Lyapunov method to guarantee the convergence of tracking error. However, this result is restricted to the SISO nonlinear systems. Therefore, the aim of this paper is to extend this previous result to the design of an adaptive decentralized PID controller for multivariable nonlinear systems.

In this paper, the Co-JITL method is incorporated into the proposed adaptive decentralized PID controller design for multivariable systems. In the proposed method, the local model obtained on-line by the Co-JITL at each sampling instant is used to adjust the controller parameters by an updating algorithm derived by the Lyapunov method to guarantee the convergence of predicted tracking error. Two examples are presented to illustrate the proposed control strategy and a comparison with its corresponding benchmark designs reported in the literature is made.

2. Adaptive decentralized PID controller design

As the proposed control strategy utilizes the JITL method to update the decentralized PID controller, a brief review of the JITL

* Corresponding authors.

E-mail addresses: jjiali@staff.shu.edu.cn (L. Jia), checms@nus.edu.sg (M.-S. Chiu).

modeling method is given below for ease of reference. More details can be found in [23,24].

2.1. JITL modeling technique

There are three main steps in the JITL methods to predict the model output corresponding to the query data: (1) relevant data samples in the reference 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) model output is calculated based on this local model and 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 according to the aforementioned procedure.

As a low-order model is usually employed by the JITL methods, without the loss of generality, consider the following second-order ARX (Auto-Regression with eXogeneous inputs) model:

$$\hat{y}(k) = \alpha_1^k y(k-1) + \alpha_2^k y(k-2) + \beta^k u(k-1) \quad (1)$$

where $\hat{y}(k)$ is the predicted output by the JITL at the k th sampling instant, $y(k-1)$ and $u(k-1)$ are the output and manipulated variables at the $(k-1)$ th sampling instant, α_1^k, α_2^k and β^k are the model parameters at the k th sampling instant. The regression vector for the ARX model given in Eq. (1) is given by:

$$\mathbf{x}(k) = [y(k-1) \quad y(k-2) \quad u(k-1)]^T \quad (2)$$

Suppose that reference database \mathbf{Z} used for the Co-JITL method consists of N data, $\mathbf{Z} = [\mathbf{z}_1 \quad \dots \quad \mathbf{z}_N]^T$ where $\mathbf{z}_l = [\mathbf{x}(l)^T \quad y(l)]^T$ for $l=1-N$, the objective of Co-JITL is to obtain a local ARX model to approximate the actual nonlinear systems at each sampling instant by focusing on the relevant region around the current operating condition represented by the query data \mathbf{z}^q that has identical structure of \mathbf{z}_l . The first step is to select the relevant dataset from reference database that resembles the query data. To do so, reference database is divided into smaller datasets \mathbf{Z}_l where each dataset consists of W data denoted by $\mathbf{Z}_l = [\mathbf{z}_l \quad \dots \quad \mathbf{z}_{l+W-1}]^T$. The similarity between the query data and each dataset \mathbf{Z}_l is evaluated by the weighted sum of Q and T^2 statistics as follows [24,28]:

$$J_l = \kappa Q + (1 - \kappa) T^2 \quad (3)$$

where κ is bounded between 0 and 1.

Principal component analysis (PCA), which compresses datasets to lower dimensions in order to find linear combinations of variables that can best describe the important trends or patterns in the dataset, is used to compute Q statistic, which is the distance between the query data and the subspace spanned by the principal components of \mathbf{Z}_l . In PCA, for a data matrix \mathbf{Z}_l , the loading matrix \mathbf{V}_p , where the column space of \mathbf{V}_p is the subspace spanned by principal components and the score matrix \mathbf{T}_p is a projection of \mathbf{Z}_l onto the subspace spanned by principal components, are obtained by:

$$\mathbf{T}_p = \mathbf{Z}_l \mathbf{V}_p \quad (4)$$

Using \mathbf{V}_p , the data matrix \mathbf{Z}_l is reconstructed as follows:

$$\hat{\mathbf{Z}}_l = \mathbf{T}_p \mathbf{V}_p^T = \mathbf{Z}_l \mathbf{V}_p \mathbf{V}_p^T \quad (5)$$

The Q statistic is then defined by the errors calculated by,

$$Q = \sum_{i=1}^W (q_i - \hat{q}_i)^2 \quad (6)$$

where q_i and \hat{q}_i denote the i th row of \mathbf{Z}_l^T and $\hat{\mathbf{Z}}_l^T$, respectively. Hence, a smaller Q indicates a smaller error and a higher correlation.

The other important statistic used in the Co-JITL is the T^2 statistic, which is also known as the Hotelling's statistic. The T^2 statistic

defined by Eq. (7) is used as the measure of the normalized distance from the origin in the subspace spanned by the principal components. A smaller value of T^2 static indicates that the query data lies closer to the mean of \mathbf{Z}_l .

$$T^2 = \sum_{r=1}^p \frac{t_r^2}{\sigma_{t_r}^2} \quad (7)$$

where σ_{t_r} is the standard deviation of the r th score, t_r , and p stands for the number of principle components.

According to Eq. (3), the relevant dataset is chosen to be the one resulting in the smallest J_l . As the initial reference database constructed using process input and output data obtained around the nominal operating condition may not provide satisfactory predictive performance for new operating region where the process data may not be available to construct the initial reference database, Co-JITL's modeling accuracy can be improved by adding new process data into the reference database at each sampling instant. Furthermore, to lessen the computational burden for the Co-JITL modeling, a threshold \bar{J} is introduced such that if $J_l^{\#}$ calculated based on the query data and previous relevant dataset is no larger than \bar{J} , the previous relevant dataset remains to be the relevant dataset for the current sampling instant and the previous ARX model is used for the Co-JITL prediction. Otherwise, the relevant dataset is updated by the dataset corresponding to the minimum index J_l and the corresponding ARX model is the current local model used for Co-JITL's prediction [24].

2.2. Proposed controller design

The proposed adaptive decentralized PID controller design is depicted in Fig. 1, where the JITL technique is mainly used to identify the current process dynamics at each sampling instant as discussed previously, the following second-order ARX model is employed in the Co-JITL method:

$$\hat{y}_1(k) = \alpha_{1,1}^k y_1(k-1) + \alpha_{2,1}^k y_1(k-2) + \beta_1^k u_1(k-1)$$

$$\hat{y}_2(k) = \alpha_{1,2}^k y_2(k-1) + \alpha_{2,2}^k y_2(k-2) + \beta_2^k u_2(k-1)$$

\vdots

$$\hat{y}_m(k) = \alpha_{1,m}^k y_m(k-1) + \alpha_{2,m}^k y_m(k-2) + \beta_m^k u_m(k-1)$$

or

$$\hat{\mathbf{y}}(k) = \text{diag}[\alpha_{1,i}^k]_{i=1-m} \mathbf{y}(k-1) + \text{diag}[\alpha_{2,i}^k]_{i=1-m} \mathbf{y}(k-2) + \text{diag}[\beta_i^k]_{i=1-m} \mathbf{u}(k-1) \quad (8)$$

where $\hat{\mathbf{y}}(k) = [\hat{y}_1(k) \quad \hat{y}_2(k) \quad \dots \quad \hat{y}_m(k)]^T \in R^m$ is the predicted output by the Co-JITL at the k th sampling instant, m is the number of input/output variables, $\mathbf{y}(k-1) \in R^m$ and $\mathbf{u}(k-1) \in R^m$ are the process output and input variables at the $(k-1)$ th sampling instant,

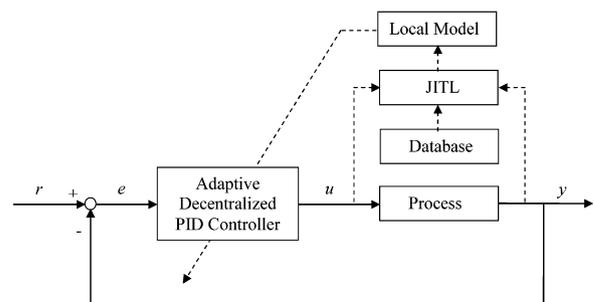


Fig. 1. JITL-based adaptive decentralized PID control scheme.

متن کامل مقاله

دریافت فوری ←

ISIArticles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات