processes

Article

Batch Process Modeling with Few-Shot Learning

Shaowu Gu [1], Junghui Chen [2,*] and Lei Xie [1,*]

[1] State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China; 11632043@zju.edu.cn
[2] Department of Chemical Engineering, Chung-Yuan Christian University, Chung-Li District, Taoyuan 320314, China
* Correspondence: jason@wavenet.cycu.edu.tw (J.C.); leix@iipc.zju.edu.cn (L.X.)

Abstract: Batch processes in the biopharmaceutical and chemical manufacturing industries often develop new products to meet changing market demands. When the dynamic models of these new products are trained, dynamic modeling with limited data for each product can lead to inaccurate results. One solution is to extract useful knowledge from past historical production data that can be applied to the product of a new grade. In this way, the model can be built quickly without having to wait for additional modeling data. In this study, a subspace identification combined common feature learning scheme is proposed to quickly learn a model of a new grade. The proposed modified state-space model contains common and special parameter matrices. Past batch data can be used to train common parameter matrices. Then, the parameters can be directly transferred into a new SID model for a new grade of the product. The new SID model can be quickly well trained even though there is a limited batch of data. The effectiveness of the proposed algorithm is demonstrated in a numerical example and a case of an industrial penicillin process. In these cases, the proposed common feature extraction for the SID learning framework can achieve higher performance in the multi-input and multi-output batch process regression problem.

Keywords: batch process; few-shot learning; common feature space; subspace identification

check for updates

Citation: Gu, S.; Chen, J.; Xie, L. Batch Process Modeling with Few-Shot Learning. *Processes* **2023**, *11*, 1481. https://doi.org/10.3390/pr11051481

Academic Editors: Nazih Moubayed, Mohand Djeziri, Hiba Al-Sheikh and Andrew Hoadley

Received: 17 March 2023
Revised: 29 April 2023
Accepted: 8 May 2023
Published: 12 May 2023

1. Introduction

The batch process is more flexible than the large-scale continuous production process as the former can better respond to market changes and customer needs. Contrary to the era of big data advocated for in recent years, in many fields, such as chemical process control [1,2] and military electronic product production [3], data are collected with diverse small batch production models employed to cope with rapid product changes [4]. Moreover, the difficulty in collecting data involves many key variables [5], resulting in the problem of small samples in the batch process. Such small samples are called the “low-N problem” [6]. It is difficult to obtain sufficient batches of production data in a certain manufacturing period because of the long operation time, complex operation steps, expensive costs for data collection, and insufficient manpower and material resources. It is not uncommon for the new chemical compounds to run only once or twice in the production equipment, resulting in a low-N scenario in the production of new products [7].

The problem of small samples has been studied in many fields. Some solutions have been used to enhance the robustness of the results obtained from the follow-up training by adding more information to the data or to assist the training of the target by transferring knowledge or data from other aspects [8]. Tulsyan et al. [7] considered using the data sampled repeatedly by hardware to train a GP-based generator and generate a large number of data; then, the authors used the generated data for further process modeling. This method of requiring hardware to repeat sampling may not be implemented in every process; in particular, the measurement data of some key variables are difficult to obtain many times. In addition, the quality of the samples generated by the generator providing auxiliary information is not as good as expected. Zhang et al. [9] combined the physical model and data-driven model, and then a hybrid model framework was proposed to model the process with fewer data in some operating conditions and applied to the dynamic model prediction of algae lutein synthesis. This method is not for practical applications as it is often difficult to obtain the mathematical model in most of the complicated batch processes.

In the industrial batch process, there may be a number of data for other similar products produced in the past. As the current producing product and the past similar product data are measured in the same process, they must share some similar or even the same features. It is reasonable to extract the information from data in these different grades of products that are highly relevant to the current (or target) produced product. Then, the target process model with the transferred data information can be enhanced. Jaeckle et al. [10] proposed extended principal component regression (EPCR), which combines the output data of similar old processes with the output data of the new process for calculation. However, EPCR does not consider the input information of the process, which is unfavorable for a regression problem. Muñoz et al. [11] extended EPCR, considered the input information, and proposed Joint-Y PLS (JY-PLS). In JY-PLS, the variables of the output data are supposed to have the same statistical distributions. Recently, the method based on JY-PLS has been applied to process quality prediction and soft sensing [12,13]. Chu et al. [14] considered the shortage of new data in the initial operation stage of the batch and used the latent variable process migration model (LV-PTM) to transfer similar process data to the new batch of the initial stage. These transfer-based methods require sufficient source data from the same source, which may not be applicable to all kinds of small data scenarios. The above methods based on transfer learning allow the target to transfer knowledge from the data of a single source. However, the data from a single source often does not have enough information as a reliable source of knowledge. The data collected are often diverse, and the number of individual data is small. Therefore, it is necessary to maximize and efficiently use all source data.

Yamaguchi et al. [15] used multi-task learning (MTL) to address the data scarcity of each product in the multi-level batch process. By sharing useful information among multiple related grade products, MTL can improve the accuracy of the model in the case of data shortage. Although MTL makes use of the data of all small sample products for mutual assistance and complementarity, it does not focus on the modeling and analysis of new products. It also does not provide the common information of all collected data. To obtain knowledge transferred from the multiple tasks for the learning of new tasks, Tripuraneni et al. [16] proposed a method to learn common feature representation from multiple tasks, so the data of the new task can be projected into this representation, which can reduce the number of samples required to find the best regressor on the new task. Combining data from multiple products with small samples can help compensate for the limitations of using the data from a single source. However, in the research conducted by Tripuraneni et al. [16], only the common feature space of the input was extracted without considering the dynamic variables. It is impossible to extract shared knowledge of industrial processes in general dynamic MIMO systems.

In this study, a method for batch process modeling with the low-N problem is proposed. The method extracts common features from multi-grade batch process data and incorporates these features into the subspace identification for the new batch process modeling. The problem and novelties considered in this study are summarized as follows:

- The proposed method considers the characteristics of batch processes, including their multi-input and multi-output structures, their dynamic behavior, and the issue of the uneven length of collected data. The dynamic behavior of the batch process is described using a linear time-invariant state-space (LTI-SS) model, and the original data are used for modeling without the need for data warping.
- To extract common parts from the historical data, we propose a modified version of the state-space model, which further divides the original model parameters into common and individual parts.

- When using the proposed model, the input-output equation derived from the proposed LTI-SS model has two coupled common features. We introduce the technique of oblique projection to separate the two parameters so that it can avoid computationally expensive iterative solutions and problems that may not converge.
- Based on the two separate sets of equations, we derive and organize their final solutions, which can be obtained by calculating the eigenvalue decomposition.
- Based on the modified state-space model, we combine it with the subspace identification method, and through the substitution of common knowledge, we can effectively improve the modeling performance in the case of a few samples.

The rest of this article is organized as follows. In Section 2, the input-output equation and the problem to be solved are specifically defined. Section 3 introduces the oblique projection to separate the input-output equation into z-space and u-space, respectively. In Section 4, the common feature parameter matrices in the model are estimated by solving the objective function defined in Section 3. In Section 5, the calculated common feature parameter matrices are applied to model the batch data of a new task. Then, the model of the new task can be quickly identified even though there is a small number of batch data in the new task. In Section 6, two examples, including a numerical case and a case of industrial penicillin production, show the performance of the proposed method. In Section 7, the conclusion is given. Finally, to enable readers to quickly and clearly understand the meaning of each symbol, a table is included in Nomenclature; it allows readers to easily compare and refer to it while reading.

2. Problem Formulation and Description

The chemical batch process often exhibits nonlinear and dynamic behaviors. Measurements of process variables are expected to be strongly serially correlated. There are additional characteristics of the long duration and operation in different operating conditions at different phases. To ensure the process is stirred evenly and the reaction is complete, the process is deliberately operated in a fixed operating condition for a while at each operation phase. Thus, the local behavior at each operation phase tends to be linear. To produce different grades of products, the batch process is operated in different operating conditions at different phases, but there are similar behaviors in the physical or chemical properties in each production process [17]. Thus, very limited batch data are often available for each grade of the operating batch, particularly in the new grade produced at the moment of the initial operation period. Only using the limited data of the new batch is certainly insufficient for establishing reliable models, while the model trained by directly extracting knowledge from all grades of batch data has a bias in favor of the graded products with more batch data.

To obtain common knowledge from different grade sources, consider G different types of batch production processes, which are diverse but similar. The dataset $\mathcal{D}_g$ of the task (the production grade) g is:

$$\mathcal{D}_g = \begin{bmatrix} (\mathbf{u}_{g,1}^{1}, \mathbf{y}_{g,1}^{1}) & \cdots & (\mathbf{u}_{g,K_g^{I_g}}^{I_g}, \mathbf{y}_{g,K_g^{I_g}}^{I_g}) \end{bmatrix} \tag{1}$$

where $I_g^s; g = 1, \cdots, G$ is the number of batches in the production grade g, $\mathbf{u}_{g,k}^{i} \in \mathbb{R}^{L\times 1}$ and $\mathbf{y}_{g,k}^{i} \in \mathbb{R}^{M\times 1}$ are the input data and the output data at the kth sampling time of the ith batch data in the production grade g, respectively, $i = 1, \cdots, I_g$; $k = k_0, k_0 + 1, \cdots, K_g^i$, $g = 1, \cdots, G$, and K_g^i is the operation time of the ith batch in the production grade g.

Similarly, for the modeling of new tasks, new batches in the production grade $G + 1$ are

$$\mathcal{D}_{G+1} = \left[\left(\mathbf{u}_{G+1,1}^{1}, \mathbf{y}_{G+1,1}^{1}\right), \cdots, \left(\mathbf{u}_{G+1,K_{G+1}^{I_{G+1}}}^{I_{G+1}}, \mathbf{y}_{G+1,K_{G+1}^{I_{G+1}}}^{I_{G+1}}\right) \right] \tag{2}$$

To properly describe the dynamic characteristics of the operating batch process, the state-space model is appropriate for multi-input and multi-output dynamic processes. Thus, for the production grade g of the operating batch process, a linear time-invariant state-space model for each phase can be written by:

$$\begin{aligned} \mathbf{x}^i_{g,k+1} &= \mathbf{A}_g \mathbf{x}^i_k + \mathbf{B}_g \mathbf{u}^i_{g,k} \\ \mathbf{y}^i_{g,k} &= \mathbf{C}_g \mathbf{x}^i_{g,k} + \boldsymbol{\varepsilon}^i_{g,k} \end{aligned} \tag{3}$$

where $\mathbf{y}^i_{g,k} \in \mathbb{R}^{M \times 1}$ and $\mathbf{x}^i_{g,k} \in \mathbb{R}^{N_x \times 1}$ are M-dimensional system output and N_x-dimensional system states of the ith batch in the production grade g at the sampling time k, respectively. $\mathbf{u}^i_{g,k} \in \mathbb{R}^{L \times 1}$ is the L-dimensional system input of batch i in the production g at the time point k. It is assumed to have Gaussian distribution. $\boldsymbol{\varepsilon}^i_{g,k} \in \mathbb{R}^{M \times 1}$ is an M-dimensional additional noise of batch i in the production grade g at the time point k. It is independent of $\mathbf{u}^i_{g,k}$ and follows the Gaussian distribution. $\mathbf{A}_g \in \mathbb{R}^{N_x \times N_x}$, $\mathbf{C}_g \in \mathbb{R}^{M \times N_x}$, and $\mathbf{B}_g \in \mathbb{R}^{N_x \times L}$ are the parameters of batch production grade g. Some common features ($\mathbf{C}_c$ and $\mathbf{B}_c$) in the parameters $\mathbf{C}_g$ and $\mathbf{B}_g$ are shared in all different grades, expressed as:

$$\begin{aligned} \mathbf{C}_g &= \mathbf{C}_c \mathbf{Q}_g \\ \mathbf{B}_g &= \mathbf{R}_g \mathbf{B}_c \end{aligned} \tag{4}$$

where $\mathbf{C}_c \in \mathbb{R}^{M \times N_q}$ and $\mathbf{B}_c \in \mathbb{R}^{N_r \times L}$ have N_q orthogonal column vectors and N_r orthogonal row vectors, respectively, and $\mathbf{Q}_g \in \mathbb{R}^{N_q \times N_x}$ and $\mathbf{R}_g \in \mathbb{R}^{N_r \times L}$ are the remaining parts for each grade. With the common features ($\mathbf{C}_c$ and $\mathbf{B}_c$), Equation (3) can be rewritten as

$$\begin{aligned} \mathbf{x}^i_{g,k+1} &= \mathbf{A}_g \mathbf{x}^i_{g,k} + \mathbf{R}_g \mathbf{B}_c \mathbf{u}^i_{g,k} \\ \mathbf{y}^i_{g,k} &= \mathbf{C}_c \mathbf{Q}_g \mathbf{x}^i_{g,k} + \boldsymbol{\varepsilon}^i_{g,k} \end{aligned} \tag{5}$$

Equation (5) is then expressed in the form of the Hankel matrices.

$$\mathbf{Y}^i_{g,f} = \boldsymbol{\Gamma}_{g,x} \mathbf{X}^i_{g,f} + \mathbf{L}_{g,u} \mathbf{U}^i_{g,f} + \mathbf{E}^i_{g,f} \tag{6}$$

where all of the Hankel matrices have a dynamic window size of N, and the number of moving windows is $J^i_g = K^i_g - N$:

$$\mathbf{Y}^i_{g,f} = \begin{bmatrix} \mathbf{y}^i_{g,N+1} & \mathbf{y}^i_{g,N+2} & \cdots & \mathbf{y}^i_{g,N+J^i_g} \\ \mathbf{y}^i_{g,N+2} & \mathbf{y}^i_{g,N+3} & \cdots & \mathbf{y}^i_{g,N+J^i_g+1} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{y}^i_{g,2N} & \mathbf{y}^i_{g,2N+1} & \cdots & \mathbf{y}^i_{g,2N+J^i_g-1} \end{bmatrix} \in \mathbb{R}^{NM \times J^i_g} \tag{7}$$

$$\mathbf{U}^i_{g,f} = \begin{bmatrix} \mathbf{u}^i_{g,N+1} & \mathbf{u}^i_{g,N+2} & \cdots & \mathbf{u}^i_{g,N+J^i_g} \\ \mathbf{u}^i_{g,N+2} & \mathbf{u}^i_{g,N+3} & \cdots & \mathbf{u}^i_{g,N+J^i_g+1} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{u}^i_{g,2N} & \mathbf{u}^i_{g,2N+1} & \cdots & \mathbf{u}^i_{g,2N+J^i_g-1} \end{bmatrix} \in \mathbb{R}^{NL \times J^i_g} \tag{8}$$

$$\mathbf{X}^i_{g,f} = \begin{bmatrix} \mathbf{x}^i_{g,N+1} & \mathbf{x}^i_{g,N+1} & \cdots & \mathbf{x}^i_{g,N+J^i_g-1} \end{bmatrix} \in \mathbb{R}^{N_x \times J^i_g} \tag{9}$$

$$\mathbf{E}^i_{g,f} = \begin{bmatrix} \boldsymbol{\varepsilon}^i_{g,N+1} & \boldsymbol{\varepsilon}^i_{g,N+2} & \cdots & \boldsymbol{\varepsilon}^i_{g,N+J^i_g} \\ \boldsymbol{\varepsilon}^i_{g,N+2} & \boldsymbol{\varepsilon}^i_{g,N+3} & \cdots & \boldsymbol{\varepsilon}^i_{g,N+J^i_g+1} \\ \vdots & \vdots & \cdots & \vdots \\ \boldsymbol{\varepsilon}^i_{g,2N} & \boldsymbol{\varepsilon}^i_{g,2N+1} & \cdots & \boldsymbol{\varepsilon}^i_{g,2N+J^i_g-1} \end{bmatrix} \in \mathbb{R}^{NM \times J^i_g} \tag{10}$$

$$\boldsymbol{\Gamma}_{g,x} = \begin{bmatrix} \mathbf{C}_c\mathbf{Q}_g \\ \mathbf{C}_c\mathbf{Q}_g\mathbf{A}_g \\ \vdots \\ \mathbf{C}_c\mathbf{Q}_g\mathbf{A}_g^{N-1} \end{bmatrix} \in \mathbb{R}^{NM \times N_x} \tag{11}$$

$$\mathbf{L}_{g,u} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}_c\mathbf{Q}_g\mathbf{R}_g\mathbf{B}_c & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{C}_c\mathbf{Q}_g\mathbf{A}_g^{N-2}\mathbf{R}_g\mathbf{B}_c & \cdots & \mathbf{C}_c\mathbf{Q}_g\mathbf{R}_g\mathbf{B}_c & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{NM \times NL} \tag{12}$$

As the state term $\mathbf{X}^i_{g,f}$ of the system is unknown, the past input and output data are used to approximate it:

$$\mathbf{X}^i_{g,f} \approx \boldsymbol{\Phi}_{g,z}\mathbf{Z}^i_{g,p} \tag{13}$$

where

$$\boldsymbol{\Phi}_{g,z} = \begin{bmatrix} \mathbf{A}_g^N\boldsymbol{\Gamma}^\dagger_{g,x} & \boldsymbol{\Delta}_g - \mathbf{A}_g^N\boldsymbol{\Gamma}^\dagger_{g,x}\mathbf{L}_{g,u} \end{bmatrix} \in \mathbb{R}^{N_x \times N(M+L)}$$

$$\boldsymbol{\Delta}_g = \begin{bmatrix} \mathbf{A}_g^{N-1}\mathbf{B}_g & \mathbf{A}_g^{N-2}\mathbf{B}_g & \cdots & \mathbf{B}_g \end{bmatrix}; \mathbf{Z}^i_{g,p} = \begin{bmatrix} \mathbf{Y}^i_{g,p} \\ \mathbf{U}^i_{g,p} \end{bmatrix} \in \mathbb{R}^{N(M+L) \times J^i_g}$$

$$\mathbf{Y}^i_{g,p} = \begin{bmatrix} \mathbf{y}^i_{g,1} & \mathbf{y}^i_{g,2} & \cdots & \mathbf{y}^i_{g,J^i_g} \\ \mathbf{y}^i_{g,2} & \mathbf{y}^i_{g,3} & \cdots & \mathbf{y}^i_{g,J^i_g+1} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{y}^i_{g,N} & \mathbf{y}^i_{g,N+1} & \cdots & \mathbf{y}^i_{g,N+J^i_g-1} \end{bmatrix}; \mathbf{U}^i_{g,p} = \begin{bmatrix} \mathbf{u}^i_{g,1} & \mathbf{u}^i_{g,2} & \cdots & \mathbf{u}^i_{g,J^i_g} \\ \mathbf{u}^i_{g,2} & \mathbf{u}^i_{g,3} & \cdots & \mathbf{u}^i_{g,J^i_g+1} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{u}^i_{g,N} & \mathbf{u}^i_{g,N+1} & \cdots & \mathbf{u}^i_{g,N+J^i_g-1} \end{bmatrix}$$

Equation (6) can be further rewritten as:

$$\mathbf{Y}^i_{g,f} = \boldsymbol{\Gamma}_{g,x}\boldsymbol{\Phi}_{g,z}\mathbf{Z}^i_{g,p} + \mathbf{L}_{g,u}\mathbf{U}^i_{g,f} + \mathbf{E}^i_{g,f} \tag{14}$$

Now $\boldsymbol{\Gamma}_{g,x}$ (Equation (11)) and $\mathbf{L}_{g,u}$ (Equation (12)) are substituted into Equation (14), expressed as:

$$\begin{aligned} \mathbf{Y}^i_{g,f} = (\mathbf{I} \otimes \mathbf{C}_c) & \begin{bmatrix} \mathbf{Q}_g \\ \mathbf{Q}_g\mathbf{A}_g \\ \vdots \\ \mathbf{Q}_g\mathbf{A}_g^{N-1} \end{bmatrix} \boldsymbol{\Phi}_{g,z}\mathbf{Z}^i_{g,p} \\ +(\mathbf{I} \otimes \mathbf{C}_c) & \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{Q}_g\mathbf{R}_g & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{Q}_g\mathbf{A}_g^{N-2}\mathbf{R}_g & \cdots & \mathbf{Q}_g\mathbf{R}_g & \mathbf{0} \end{bmatrix} (\mathbf{I} \otimes \mathbf{B}_c)\mathbf{U}^i_{g,f} + \mathbf{E}^i_{g,f} \end{aligned} \tag{15}$$

To obtain a more compact expression, some notations shown as follows are defined.

$$\boldsymbol{\Omega} = (\mathbf{I}_N \otimes \mathbf{C}_c) \in \mathbb{R}^{NM \times NN_q}\boldsymbol{\Theta} = (\mathbf{I}_N \otimes \mathbf{B}_c) \in \mathbb{R}^{NN_r \times NL} \tag{16}$$

$$\mathbf{O}_g = \begin{bmatrix} \mathbf{Q}_g \\ \mathbf{Q}_g\mathbf{A}_g \\ \vdots \\ \mathbf{Q}_g\mathbf{A}_g^{N-1} \end{bmatrix} \boldsymbol{\Phi}_{g,z} \in \mathbb{R}^{NN_q \times N(M+L)} \boldsymbol{\Xi}_g = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{Q}_g\mathbf{R}_g & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{Q}_g\mathbf{A}_g^{N-2}\mathbf{R}_g & \cdots & \mathbf{Q}_g\mathbf{R}_g & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{NN_q \times NN_r} \tag{17}$$

Thus, Equation (15) becomes:

$$\mathbf{Y}^i_{g,f} = \boldsymbol{\Omega}\mathbf{O}_g\mathbf{Z}^i_{g,p} + \boldsymbol{\Omega}\boldsymbol{\Xi}_g\boldsymbol{\Theta}\mathbf{U}^i_{g,f} + \mathbf{E}^i_{g,f} \tag{18}$$

Then, each column in Equation (18) can be written as:

$$\mathbf{y}^i_{g,f,j} = \boldsymbol{\Omega}\mathbf{O}_g\mathbf{z}^i_{g,j} + \boldsymbol{\Omega}\boldsymbol{\Xi}_g\boldsymbol{\Theta}\mathbf{u}^i_{g,f,j} + \boldsymbol{\varepsilon}^i_{g,f,j} \tag{19}$$

where $\boldsymbol{\Omega}$ and $\boldsymbol{\Theta}$ are unknown parameter matrices to be estimated. In this work, an oblique projection, which is used to solve $\boldsymbol{\Omega}$ and $\boldsymbol{\Theta}$, is discussed in the next subsection.

3. Model Decomposition into the z-Space and the u-Space Using Oblique Projection

Since $\boldsymbol{\Omega}$ and $\boldsymbol{\Theta}$ in Equation (18) are coupled together, to avoid solving these two parameters with the iterative way, the technique of oblique projection [18] is applied here to separate Equation (18) into two subspaces which lie in $\mathbf{z}^i_{g,j}$ and $\mathbf{u}^i_{g,f,j}$, respectively. The oblique projection $/_{\mathbf{u}^i_{g,f,j}}\mathbf{z}^i_{g,j}$ represents the projection onto $\mathbf{z}^i_{g,j}$ along the direction parallel to the input vector $\mathbf{u}^i_{g,f,j}$.

$$/_{\mathbf{u}^i_{g,f,j}}\mathbf{z}^i_{g,j} = \left(\mathbf{z}^i_{g,j}\right)^T\left(\mathbf{z}^i_{g,j}\left(\mathbf{P}^i_{u,g,j}\right)^{\perp}\left(\mathbf{z}^i_{g,j}\right)^T\right)^{-1}\mathbf{z}^i_{g,j}\left(\mathbf{P}^i_{u,g,j}\right)^{\perp} \tag{20}$$

where $\left(\mathbf{P}^i_{u,g,j}\right)^{\perp} = \mathbf{I} - \left(\mathbf{u}^i_{g,f,j}\right)^T\left(\mathbf{u}^i_{g,f,j}\left(\mathbf{u}^i_{g,f,j}\right)^T\right)\mathbf{u}^i_{g,f,j}$ is the projection operator for computing orthogonal complementary spaces of $\mathbf{u}^i_{g,f,j}$. With Equations (19) and (20), it is projected along the input matrix $\mathbf{u}^i_{g,f,j}$ onto $\mathbf{z}^i_{g,j}$, expressed as:

$$\mathbf{y}^i_{g,f,j}/_{\mathbf{u}^i_{g,f,j}}\mathbf{z}^i_{g,j} = \boldsymbol{\Omega}\mathbf{O}_g\mathbf{z}^i_{g,j}/_{\mathbf{u}^i_{g,f,j}}\mathbf{z}^i_{g,j} + \boldsymbol{\Omega}\boldsymbol{\Xi}_g\boldsymbol{\Theta}\mathbf{u}^i_{g,f,j}/_{\mathbf{u}^i_{g,f,j}}\mathbf{z}^i_{g,j} + \boldsymbol{\varepsilon}^i_{g,f,j}/_{\mathbf{u}^i_{g,f,j}}\mathbf{z}^i_{g,j} \tag{21}$$

According to the assumption of data $\mathbf{u}_{g,f,j}/_{\mathbf{u}^i_{g,f,j}}\mathbf{z}^i_{g,j} = 0$ and $\boldsymbol{\varepsilon}^i_{g,f,j}/_{\mathbf{u}^i_{g,f,j}}\mathbf{z}^i_{g,j} = 0$. Define the notation $\boldsymbol{\varphi}^i_{g,j} = \mathbf{y}^i_{g,f,j}/_{\mathbf{u}^i_{g,f,j}}\mathbf{z}^i_{g,j}$; then, the above equation can be further expressed as:

$$\boldsymbol{\varphi}^i_{g,j} = \boldsymbol{\Omega}\mathbf{O}_g\mathbf{z}^i_{g,j} \tag{22}$$

By combining all of the batch data $g = 1, \cdots, G$; $i = 1, \cdots, I_g$; $j = 1, \cdots, J_g^{(i)}$, one wants to search for the parameter matrices, $\mathbf{O}_g$ and $\boldsymbol{\Omega}$. Assume that $\boldsymbol{\Omega}$ has the orthogonal structure, and then minimize the sum of squared error defined by:

$$\begin{gathered} \min_{\boldsymbol{\Omega},\mathbf{O}_g}\left\|\boldsymbol{\Psi}^B_g - \boldsymbol{\Omega}\mathbf{O}_g\mathbf{Z}^B_{g,p}\right\|_F^2 \\ \text{s.t. } \boldsymbol{\Omega}^T\boldsymbol{\Omega} = \mathbf{I} \end{gathered} \tag{23}$$

where $\|\cdot\|_F$ is the Frobenius norm, and $g = 1, \cdots, G$.

$$\boldsymbol{\Psi}^i_g = \begin{bmatrix}\boldsymbol{\varphi}^i_{g,1} & \cdots & \boldsymbol{\varphi}^i_{g,J^i_g}\end{bmatrix} \quad \mathbf{Z}^i_{g,p} = \begin{bmatrix}\mathbf{z}^i_{g,1} & \cdots & \mathbf{z}^i_{g,J^i_g}\end{bmatrix} \tag{24}$$

and

$$\mathbf{\Psi}_g^B = [\mathbf{\Psi}_g^1 \quad \cdots \quad \mathbf{\Psi}_g^{I_g}] \quad \mathbf{Z}_{g,p}^B = [\mathbf{Z}_{g,p}^1 \quad \cdots \quad \mathbf{Z}_{g,p}^{I_g}] \tag{25}$$

Equation (23) is called the z-space model.

Similarly, using the oblique projection method, Equation (19) is projected onto $\mathbf{u}_{g,f,j}^i$ along with the matrix of the past inputs and the outputs $\mathbf{z}_{g,j}^i$.

$$/_{\mathbf{z}_{g,j}^i}\mathbf{u}_{g,f,j}^i = \left(\mathbf{u}_{g,f,j}^i\right)^T \left(\mathbf{u}_{g,f,j}^i\left(\mathbf{P}_{z,g,j}^i\right)^{\perp}\left(\mathbf{u}_{g,f,j}^i\right)^T\right)^{-1}\mathbf{u}_{g,f,j}^i\left(\mathbf{P}_{z,g,j}^i\right)^{\perp} \tag{26}$$

where $\left(\mathbf{P}_{z,g,j}^i\right)^{\perp} = \mathbf{I} - \left(\mathbf{z}_{g,j}^i\right)^T\left(\mathbf{z}_{g,j}^i\left(\mathbf{z}_{g,j}^i\right)^T\right)\mathbf{z}_{g,j}^i$ is the projection operator for computing orthogonal complementary spaces of $\mathbf{z}_{g,j}^i$. With Equations (19) and (26), it is projected along $\mathbf{z}_{g,j}^i$ onto the input matrix $\mathbf{u}_{g,f,j}^i$, and both sides multiplied by $\mathbf{\Omega}^T$ can expressed as:

$$\mathbf{\Omega}^T\mathbf{y}_{g,f,j}^i/_{\mathbf{z}_{g,j}^i}\mathbf{u}_{g,f,j}^i = \mathbf{O}_g\mathbf{z}_{g,j}^i/_{\mathbf{z}_{g,j}^i}\mathbf{u}_{g,f,j}^i + \Xi_g\mathbf{\Theta}\mathbf{u}_{g,f,j}^i/_{\mathbf{z}_{g,j}^i}\mathbf{u}_{g,f,j}^i + \mathbf{\Omega}^T\boldsymbol{\varepsilon}_{g,f,j}^i/_{\mathbf{z}_{g,j}^i}\mathbf{u}_{g,f,j}^i \tag{27}$$

where $\mathbf{z}_{g,j}^i/_{\mathbf{z}_{g,j}^i}\mathbf{u}_{g,f,j}^i = 0$, $\boldsymbol{\varepsilon}_{g,f,j}^i/_{\mathbf{z}_{g,j}^i}\mathbf{u}_{g,f,j}^i = 0$ and $\mathbf{u}_{g,f,j}^i/_{\mathbf{z}_{g,j}^i}\mathbf{u}_{g,f,j}^i = \mathbf{u}_{g,f,j}^i$. Define notation $\boldsymbol{\lambda}_{g,j}^i = \mathbf{\Omega}^T\mathbf{y}_{g,f,j}^i/_{\mathbf{z}_{g,j}^i}\mathbf{u}_{g,f,j}^i$; then, the above equation can be further expressed as:

$$\boldsymbol{\lambda}_{g,j}^i = \Xi_g\mathbf{\Theta}\mathbf{u}_{g,f,j}^i \tag{28}$$

Similarly, all batch data $g = 1,\cdots,G$; $i = 1,\cdots,I_g$; $j = 1,\cdots,J_g^{(i)}$ are combined. One wants to search for the parameter matrices, Ξ_g and $\mathbf{\Theta}$. First, assume that $\mathbf{\Theta}$ has the orthogonal structure; then, minimize the sum of squared errors defined by:

$$\begin{gathered}\min_{\mathbf{\Theta},\,\Xi_g}\left\|\mathbf{\Lambda}_g^B - \Xi_g\mathbf{\Theta}\mathbf{U}_{g,f}^B\right\|_F^2 \\ \text{s.t. } \mathbf{\Theta}\mathbf{\Theta}^T = \mathbf{I}\end{gathered} \tag{29}$$

where

$$\mathbf{\Lambda}_g^i = [\boldsymbol{\lambda}_{g,1}^i \quad \cdots \quad \boldsymbol{\lambda}_{J_g^i}^i] \quad \mathbf{U}_{g,f}^i = [\mathbf{u}_{g,f,1}^i \quad \cdots \quad \mathbf{u}_{g,f,J_g^i}^i] \tag{30}$$

and

$$\mathbf{\Lambda}_g^B = [\mathbf{\Lambda}_g^1 \quad \cdots \quad \mathbf{\Lambda}_g^{I_g}] \quad \mathbf{U}_{g,f}^B = [\mathbf{U}_{g,f}^1 \quad \cdots \quad \mathbf{U}_{g,f}^{I_g}] \tag{31}$$

Equation (29) is called the u-space model. Thus, the common parameter matrices ($\mathbf{\Omega}$ and $\mathbf{\Theta}$) and the individual parameter matrices ($\mathbf{O}_g$ and Ξ_g) of the model defined in Equation (19) can be solved by separately minimizing the two objective functions.

4. Common Feature Parameter Matrices (Ω and Θ) Estimation

4.1. $\mathbf{\Omega}$ *and* $\mathbf{O}_g$ *Estimation in the z-Space*

To solve Equation (23), first consider the loss function for each sub-task parameter, expressed as:

$$J_1^g = \left\|\mathbf{\Psi}_g^B - \mathbf{\Omega}\mathbf{O}_g\mathbf{Z}_{g,p}^B\right\|_F^2 \tag{32}$$

Taking Equation (32) derivative with respect to (w.r.t.) each $\mathbf{O}_g$; $g = 1,\cdots,G$ is:

$$\frac{\partial J_1^g}{\partial \mathbf{O}_g} = 0 = -2\mathbf{\Omega}^T\mathbf{\Psi}_g^B\left(\mathbf{Z}_{g,p}^B\right)^T - 2\mathbf{O}_g\mathbf{Z}_{g,p}^B\left(\mathbf{Z}_{g,p}^B\right)^T \tag{33}$$

Then, the solution to $\mathbf{O}_g; g = 1, \cdots, G$ is directly given.

$$\mathbf{O}_g = \mathbf{\Omega}^T \mathbf{\Psi}_g^B \left(\mathbf{Z}_{g,p}^B\right)^T \left(\mathbf{Z}_{g,p}^B \left(\mathbf{Z}_{g,p}^B\right)^T\right)^{-1} \tag{34}$$

With the expression of the optimal model parameter $\mathbf{O}_g$, Equation (34) is substituted into the loss function (Equation (32)); then, the optimization problem for the common feature parameter $\mathbf{\Omega}$ becomes:

$$\begin{gathered}\min_{\mathbf{\Omega}} \left\| \mathbf{\Psi}_g^B - \mathbf{\Omega}\mathbf{\Omega}^T \mathbf{\Psi}_g^B \left(\mathbf{Z}_{g,p}^B\right)^T \left(\mathbf{Z}_{g,p}^B \left(\mathbf{Z}_{g,p}^B\right)^T\right)^{-1} \mathbf{Z}_{g,p}^B \right\|_F^2 \\ \text{s.t. } \mathbf{\Omega}^T\mathbf{\Omega} = \mathbf{I}\end{gathered} \tag{35}$$

According to the definition of the Frobenius norm, Equation (35) can be expanded as:

$$\begin{aligned}&\sum_{g=1}^{G} \left\| \mathbf{\Psi}_g^B - \mathbf{\Omega}\mathbf{\Omega}^T \mathbf{\Psi}_g^B \left(\mathbf{Z}_{g,p}^B\right)^T \left(\mathbf{Z}_{g,p}^B \left(\mathbf{Z}_{g,p}^B\right)^T\right)^{-1} \mathbf{Z}_{g,p}^B \right\|_F^2 \\ &= \sum_{g=1}^{G} \left(tr\left(\left(\mathbf{\Psi}_g^B\right)^T \mathbf{\Psi}_g^B\right) - tr\left(\mathbf{\Omega}^T \mathbf{\Psi}_g^B \left(\mathbf{Z}_{g,p}^B\right)^T \left(\mathbf{Z}_{g,p}^B \left(\mathbf{Z}_{g,p}^B\right)^T\right)^{-1} \mathbf{Z}_{g,p}^B \left(\mathbf{\Psi}_g^B\right)^T \mathbf{\Omega}\right)\right)\end{aligned} \tag{36}$$

Considering the part related to $\mathbf{\Omega}$, the objective function (36) can be transformed to:

$$\begin{gathered}\max_{\mathbf{\Omega}} \sum_{g=1}^{G} \left(tr\left(\mathbf{\Omega}^T \mathbf{\Psi}_g^B \left(\mathbf{Z}_{g,p}^B\right)^T \left(\mathbf{Z}_{g,p}^B \left(\mathbf{Z}_{g,p}^B\right)^T\right)^{-1} \mathbf{Z}_{g,p}^B \left(\mathbf{\Psi}_g^B\right)^T \mathbf{\Omega}\right)\right) \\ \text{s.t. } \mathbf{\Omega}^T\mathbf{\Omega} = \mathbf{I}\end{gathered} \tag{37}$$

The gradient of the objective function at the optimum must be zero. Then, Equation (36) can be written as:

$$\mathbf{\Psi}_{\text{MT-L}} \mathbf{\Omega} = \alpha \mathbf{\Omega} \tag{38}$$

where α is Lagrange multiplier, and

$$\mathbf{\Psi}_{\text{MT-L}} = \sum_{g=1}^{G} \left(\mathbf{\Psi}_g^B \left(\mathbf{Z}_{g,p}^B\right)^T \left(\mathbf{Z}_{g,p}^B \left(\mathbf{Z}_{g,p}^B\right)^T\right)^{-1} \mathbf{Z}_{g,p}^B \left(\mathbf{\Psi}_g^B\right)^T \right) \tag{39}$$

Compute the eigen-decomposition of $\mathbf{\Psi}_{\text{MT-L}}$, and let $\mathbf{\Omega}$ consist of the N_q eigenvectors of $\mathbf{\Psi}_{\text{MT-L}}$ that have the largest N_q eigenvalues. Once $\mathbf{\Omega}$ is obtained, $\mathbf{O}_g$ can be directly calculated in Equation (34).

4.2. $\mathbf{\Theta}$ *and* Ξ_g *Estimation in the u-Space*

Similar to the procedure for solving Equation (29), consider the loss function for each sub-task parameter:

$$J_2^g = \left\| \mathbf{\Lambda}_g^B - \Xi_g \mathbf{\Theta} \mathbf{U}_{g,f}^B \right\|_F^2 \tag{40}$$

where $g = 1, \cdots, G$. By taking Equation (40) derivative w.r.t. each Ξ_g and setting it equal to zero, one can obtain:

$$\frac{\partial J_2^g}{\partial \Xi_g} = 0 = -2\mathbf{\Lambda}_g^B \left(\mathbf{U}_{g,f}^B\right)^T \mathbf{\Theta}^T - 2\Xi_g \mathbf{\Theta} \mathbf{\Lambda}_g^B \left(\mathbf{\Lambda}_g^B\right)^T \mathbf{\Theta}^T \tag{41}$$

Then, the solution to $\Xi_g; g = 1, \cdots, G$ is directly given.

$$\Xi_g = \mathbf{\Lambda}_g^B \left(\mathbf{U}_{g,f}^B\right)^T \mathbf{\Theta}^T \left(\mathbf{\Theta} \mathbf{U}_{g,f}^B \left(\mathbf{U}_{g,f}^B\right)^T \mathbf{\Theta}^T\right)^{-1} \tag{42}$$

With the expression of the optimal model parameter matrix Ξ_g, Equation (42) is substituted into the objective function (Equation (29)); then, the optimization problem for the common feature parameter $\mathbf{\Theta}$ becomes:

$$\min_{\mathbf{\Theta}} \sum_{g=1}^{G} \left\| \mathbf{\Lambda}_g^B - \mathbf{\Lambda}_g^B \left(\mathbf{U}_{g,f}^B\right)^T \mathbf{\Theta}^T \left(\mathbf{\Theta}\mathbf{U}_{g,f}^B \left(\mathbf{U}_{g,f}^B\right)^T \mathbf{\Theta}^T\right)^{-1} \mathbf{\Theta}\mathbf{U}_{g,f}^B \right\|_F^2 \\ \text{s.t. } \mathbf{\Theta}\mathbf{\Theta}^T = \mathbf{I} \tag{43}$$

With the definition of the Frobenius norm, the loss function in Equation (43) can be expanded as:

$$\begin{aligned} &\sum_{g=1}^{G} \left\| \mathbf{\Lambda}_g^q - \mathbf{\Lambda}_g^B \left(\mathbf{U}_{g,f}^B\right)^T \mathbf{\Theta}^T \left(\mathbf{\Theta}\mathbf{U}_{g,f}^B \left(\mathbf{U}_{g,f}^B\right)^T \mathbf{\Theta}^T\right)^{-1} \mathbf{\Theta}\mathbf{U}_{g,f}^B \right\|_F^2 \\ &= \sum_{g=1}^{G} \left(tr\left(\left(\mathbf{\Lambda}_g^B\right)^T \mathbf{\Lambda}_g^B \right) - tr\left(\left(\mathbf{\Lambda}_g^B\right)^T \mathbf{\Lambda}_g^B \left(\mathbf{U}_{g,f}^B\right)^T \mathbf{\Theta}^T \left(\mathbf{\Theta}\mathbf{U}_{g,f}^B \left(\mathbf{U}_{g,f}^B\right)^T \mathbf{\Theta}^T\right)^{-1} \mathbf{\Theta}\mathbf{U}_{g,f}^B \right)\right) \end{aligned} \tag{44}$$

Considering the part related to $\mathbf{\Theta}$, the objective function (43) can be transformed to:

$$\max_{\mathbf{\Theta}} \sum_{g=1}^{G} \left(tr\left(\frac{\mathbf{\Theta}\mathbf{U}_{g,f}^B \left(\mathbf{\Lambda}_g^B\right)^T \mathbf{\Lambda}_g^B \left(\mathbf{U}_{g,f}^B\right)^T \mathbf{\Theta}^T}{\mathbf{\Theta}\mathbf{U}_{g,f}^B \left(\mathbf{U}_{g,f}^B\right)^T \mathbf{\Theta}^T} \right)\right) \\ \text{s.t. } \mathbf{\Theta}\mathbf{\Theta}^T = \mathbf{I} \tag{45}$$

Solving Equation (44) is a nonlinear optimization problem. To reduce the complexity of nonlinear calculation, $\mathbf{\Theta} = \begin{bmatrix} \theta_1 & \cdots & \theta_{NN_r} \end{bmatrix}^T$ is calculated row by row. First, take the first-row vector θ_1 of $\mathbf{\Theta}$ in Equation (45) and make it a derivative w.r.t. θ_1, which is expressed by:

$$2\sum_{g=1}^{G} \left(\frac{\mathcal{B}_g \theta_1}{\theta_1^T \mathcal{W}_g \theta_1} - \frac{\theta_1^T \mathcal{B}_g \theta_1 \mathcal{W}_g \theta_1}{\left(\theta_1^T \mathcal{W}_g \theta_1\right)^2} \right) = 2\beta_1 \theta_1 \tag{46}$$

where $\mathcal{B}_g = \mathbf{U}_{g,f}^B \left(\mathbf{\Lambda}_g^B\right)^T \mathbf{\Lambda}_g^B \left(\mathbf{U}_{g,f}^B\right)^T$, $\mathcal{W}_g = \mathbf{U}_{g,f}^B \left(\mathbf{U}_{g,f}^B\right)^T$; $g = 1, \cdots, G$. Let $\phi_{\mathcal{W}_g,1} = \theta_1^T \mathcal{W}_g \theta_1$, and $\phi_{\mathcal{B}_g,1} = \theta_1^T \mathcal{B}_g \theta_1$. Equation (46) can be compactly expressed as

$$\mathbf{E}(\theta_1)\theta_1 = \beta_1 \theta_1 \tag{47}$$

where

$$\mathbf{E}(\theta_1) = \left(\sum_{g=1}^{G} \left(\frac{\mathcal{B}_g}{\phi_{\mathcal{W}_g,1}} - \frac{\phi_{\mathcal{B}_g,1} \mathcal{W}_g}{\phi_{\mathcal{W}_g,1}^2} \right)\right) \tag{48}$$

Equation (47) is a nonlinear eigenvalue problem associated with the extracted eigenvector. For this problem, an iterative algorithm can be used [19]. In the first iteration, with the guessed $\theta_1^{(0)}$, the left-hand side of Equation (47) can be computed and an approximation to the eigenvector $\theta_1^{(1)}$ associated with the smallest eigenvalue of $\mathbf{E}\left(\theta_1^{(0)}\right)$ can be estimated. Then, the left-hand side of Equation (47) with the new computed $\theta_1^{(1)}$ is updated, and the updated eigenvector $\theta_1^{(2)}$ associated with the smallest eigenvalue of $\mathbf{E}\left(\theta_1^{(1)}\right)$ can be estimated. The above same procedure is repeated until $\theta_1^{(1)}$ has no significant change. Once the optimal result $\hat{\theta}_1$ is found, the deflation technique is used to remove the support and the query sets related to the direction $\hat{\theta}_1$:

$$\left(\mathbf{U}_{g,f}\right)^{(2)} = \left(\mathbf{U}_{g,f}\right)^{(1)} - \hat{\theta}_1 \left(\hat{\theta}_1^T \hat{\theta}_1\right)^{-1} \hat{\theta}_1^T \left(\mathbf{U}_{g,f}\right)^{(1)} \tag{49}$$

Then, the second parameter vector of $\boldsymbol{\theta}_1$ is solved based on Equation (47). The above procedure is repeated until the desired number of vectors is obtained. The whole iterative procedure originated is applied to solving Equation (47) as shown in Algorithm 1.

Algorithm 1: Iterative procedure for the of the parameter $\boldsymbol{\Theta}$ estimation.

```
Input:
$\mathcal{B}_g^{(0)}, \mathcal{W}_g^{(0)}$ for $g = 1, 2, \cdots, G$ and $\boldsymbol{\theta}_i^{(0)}$
Process:
1.  For $i = 1, \cdots, NN_r$
    (1)  For $t = 1, \cdots$, until convergence do Compute the eigenvalue decomposition of
         $\mathbf{E}^{(t)}\left(\boldsymbol{\theta}_i^{(t-1)}\right)$ and select the eigenvector to the largest eigenvalues $\boldsymbol{\theta}_i^{(t)}$ End for
    (2)  $\left(\mathbf{U}_{g,f}\right)^{(i)} = \left(\mathbf{U}_{g,f}\right)^{(i-1)} - \hat{\boldsymbol{\theta}}_i\left(\hat{\boldsymbol{\theta}}_i^T\hat{\boldsymbol{\theta}}_i\right)^{-1}\hat{\boldsymbol{\theta}}_i^T\left(\mathbf{U}_{g,f}\right)^{(i-1)}$
    (3)  Update $\mathcal{B}_g^{(i)}$ and $\mathcal{W}_g^{(i)}$ by using $\left(\mathbf{U}_{g,f}\right)^{(i)}$ End for
Output:
$\hat{\boldsymbol{\Theta}} = \begin{bmatrix} \hat{\boldsymbol{\theta}}_1^T \\ \vdots \\ \hat{\boldsymbol{\theta}}_{NN_r}^T \end{bmatrix}$
```

For clarity, the entire procedure for estimating the parameter matrices $(\boldsymbol{\Omega}, \boldsymbol{\Theta})$ of the model is summarized as shown in Algorithm 2.

Algorithm 2: Algorithm for the estimation of the common features $\boldsymbol{\Theta}$ and $\boldsymbol{\Omega}$.

```
Input:
The normalized set data, $\mathbf{u}_g^i \in \mathbb{R}^{L\times 1}$, $\mathbf{y}_g^i \in \mathbb{R}^{M\times 1}$; $i = 1, \cdots, I_g$; $g = 1, \cdots, G$
The dynamic step size $N$
The common feature parameter sizes $N_q$ and $N_r$
Process:
1.  Arrange the input and output data of each batch of all the products into the Hankel matrix
    according to equation (18)
2.  Using oblique projections equation (20) and (26), get $\boldsymbol{\varphi}_{g,j}^i$ and $\boldsymbol{\lambda}_{g,j}^i$
3.  According to equation (38), calculate the SVD decomposition of $\boldsymbol{\Psi}_{\text{MT-L}}$ and take the first
    $NN_q$ eigenvectors to obtain the common parameter matrix $\boldsymbol{\Omega}$
4.  Substitute the estimated $\boldsymbol{\Omega}$ from into equation (28), and then calculate $\boldsymbol{\Theta}$ according to
    Algorithm1.
Output:
The common feature parameters $\boldsymbol{\Omega}$ and $\boldsymbol{\Theta}$
```

5. Refining Parameter Matrices Using Testing Sets

Subspace Identification with the Common Feature Parameter Matrices

Once the common feature parameter matrices $\hat{\boldsymbol{\Omega}}$ and $\hat{\boldsymbol{\Theta}}$ are estimated using the meta-training sets, the pre-trained model structure and the corresponding parameter matrices will be transferred to the new batch $G + 1$. Then, the new batch with limited batch data can be quickly learned. Substituting $\hat{\boldsymbol{\Omega}}$ and $\hat{\boldsymbol{\Theta}}$ into Equation (18), the input-output matrix equation of the ith batch process is:

$$\mathbf{Y}_f^i = \boldsymbol{\Omega}\widetilde{\mathbf{O}}\boldsymbol{\Omega}^T\mathbf{Y}_p^i + \boldsymbol{\Omega}\widetilde{\boldsymbol{\Xi}}\boldsymbol{\Theta}\mathbf{U}_p^i + \boldsymbol{\Omega}\boldsymbol{\Xi}\boldsymbol{\Theta}\mathbf{U}_f^i + \mathbf{E}_f^i \quad (50)$$

where the batch grade index is omitted as the new batch $G + 1$ has only one grade.

$$\widetilde{\mathbf{O}} = \left[\begin{bmatrix} \mathbf{Q} \\ \mathbf{QA} \\ \vdots \\ \mathbf{QA}^{N-1} \end{bmatrix} \mathbf{A}^N \begin{bmatrix} \mathbf{Q} \\ \mathbf{QA} \\ \vdots \\ \mathbf{QA}^{N-1} \end{bmatrix}^{\dagger} \quad \begin{bmatrix} \mathbf{Q} \\ \mathbf{QA} \\ \vdots \\ \mathbf{QA}^{N-1} \end{bmatrix} \begin{bmatrix} \mathbf{A}^{N-1}\mathbf{R} & \mathbf{A}^{N-2}\mathbf{R} & \cdots & \mathbf{R} \end{bmatrix} - \mathbf{A}^N \begin{bmatrix} \mathbf{Q} \\ \mathbf{QA} \\ \vdots \\ \mathbf{QA}^{N-1} \end{bmatrix}^{\dagger} \boldsymbol{\Xi} \right] \tag{51}$$

$$\widetilde{\mathbf{Z}}_p^i = \begin{bmatrix} \boldsymbol{\Omega}^T \mathbf{Y}_p^i \\ \boldsymbol{\Theta} \mathbf{U}_p^i \end{bmatrix} \tag{52}$$

The first three terms of the right-hand side of Equation (50) are combined

$$\widetilde{\mathbf{Y}}_f^i = \mathbf{P}\widetilde{\mathbf{W}}_{ZU}^i + \widetilde{\mathbf{E}}_f^i \tag{53}$$

where $\widetilde{\mathbf{Y}}_f^i = \hat{\boldsymbol{\Omega}}^T \mathbf{Y}_f^i$, $\mathbf{P} = \begin{bmatrix} \mathbf{O} & \boldsymbol{\Xi} \end{bmatrix}$, $\widetilde{\mathbf{E}}_f^i = \hat{\boldsymbol{\Omega}}^T \mathbf{E}_f^i$, $\widetilde{\mathbf{U}}_f^i = \hat{\boldsymbol{\Theta}} \mathbf{U}_f^i$, and $\widetilde{\mathbf{W}}_{ZU}^i = \begin{bmatrix} \mathbf{Z}_p^i \\ \widetilde{\mathbf{U}}_f^i \end{bmatrix}$.

For the solution of Equation (53), the sub-blocks ($\widetilde{\mathbf{Y}}_f^i$ and $\widetilde{\mathbf{W}}_{ZU}^i$) of all I batches horizontally concatenate as follows:

$$\widetilde{\mathbf{Y}}_B = \begin{bmatrix} \widetilde{\mathbf{Y}}_f^1 & \cdots & \widetilde{\mathbf{Y}}_f^I \end{bmatrix} \quad \widetilde{\mathbf{W}}_{ZU,B} = \begin{bmatrix} \widetilde{\mathbf{W}}_{ZU}^1 & \cdots & \widetilde{\mathbf{W}}_{ZU}^I \end{bmatrix} \tag{54}$$

Then, the objective function is written as:

$$\min_{\mathbf{P}} \left\| \widetilde{\mathbf{Y}}_B - \mathbf{P}\widetilde{\mathbf{W}}_{ZU,B} \right\|_2^2 \tag{55}$$

The solution to Equation (55) can be obtained using the least-squares method, and $\hat{\mathbf{P}}$ can be solved uniquely.

$$\hat{\mathbf{P}} = \widetilde{\mathbf{Y}}_B \widetilde{\mathbf{W}}_{ZU,B}^T \left(\widetilde{\mathbf{W}}_{ZU,B} \widetilde{\mathbf{W}}_{ZU,B}^T \right)^{-1} \tag{56}$$

To further obtain the state-space model parameters, the estimated parameter $\hat{\mathbf{P}}$ is divided into the corresponding matrix:

$$\hat{\mathbf{P}} = \begin{bmatrix} \hat{\mathbf{O}} & \hat{\boldsymbol{\Xi}} \end{bmatrix}$$

According to Equations (6) and (18), $\boldsymbol{\Omega}^T \boldsymbol{\Gamma}_x \mathbf{X}_f = \mathbf{O}\mathbf{Z}_p$. Decompose $\hat{\mathbf{O}}\mathbf{Z}_p$ with SVD:

$$\hat{\mathbf{O}}\mathbf{Z}_p = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \sum_1 & 0 \\ 0 & \sum_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix} \tag{57}$$

where $\sum_1$ is the dominant singular value. The observability matrix and the state sequence of the system can be directly determined from the above SVD:

$$\boldsymbol{\Gamma}_x^{\boldsymbol{\Omega}} = \hat{\boldsymbol{\Omega}}^T \boldsymbol{\Gamma}_x = \mathbf{U}_1 \sum\nolimits_1^{1/2} = \begin{bmatrix} \hat{\mathbf{Q}} \\ \hat{\mathbf{Q}}\hat{\mathbf{A}} \\ \vdots \\ \hat{\mathbf{Q}}\hat{\mathbf{A}}^{N-1} \end{bmatrix} \tag{58}$$

$$\hat{\mathbf{X}}_f = \sum\nolimits_1^{1/2} \mathbf{V}_1^T = \begin{bmatrix} \hat{\mathbf{x}}_{N+1} & \cdots & \hat{\mathbf{x}}_J \end{bmatrix} \tag{59}$$

According to the definition of $\boldsymbol{\Gamma}_x^{\boldsymbol{\Omega}}$, the parameter matrix ($\hat{\mathbf{A}}$) in the new batch production is:

$$\hat{\mathbf{A}} = \left(\boldsymbol{\Gamma}_x^{\boldsymbol{\Omega}}(N_x + 1 : end, :)\right)^{\dagger} \boldsymbol{\Gamma}_x^{\boldsymbol{\Omega}}(1 : end - N_x, :) \tag{60}$$

According to the model in Equation (3), the estimated results $\hat{\mathbf{X}}_f$ and $\hat{\mathbf{A}}$ are substituted to obtain:

$$\begin{bmatrix} \hat{\mathbf{x}}_{N+2} & \cdots & \hat{\mathbf{x}}_J \end{bmatrix} - \hat{\mathbf{A}}\begin{bmatrix} \hat{\mathbf{x}}_{N+1} & \cdots & \hat{\mathbf{x}}_{J-1} \end{bmatrix} = \mathbf{B}\begin{bmatrix} \mathbf{u}_{N+1} & \cdots & \mathbf{u}_{J-1} \end{bmatrix} \tag{61}$$

$$\begin{bmatrix} \mathbf{y}_{N+1} & \cdots & \mathbf{y}_J \end{bmatrix} = \mathbf{C}\begin{bmatrix} \hat{\mathbf{x}}_{N+1} & \cdots & \hat{\mathbf{x}}_J \end{bmatrix} + \begin{bmatrix} \boldsymbol{\varepsilon}_{N+1} & \cdots & \boldsymbol{\varepsilon}_J \end{bmatrix} \tag{62}$$

In addition, the parameter matrices **B** and **C** in Equations (61) and (62) can be obtained using the least-squares method.

6. Case Study

Two cases were used to verify the proposed MT-L SID: a numerical example and an industrial penicillin batch production process.

6.1. Numerical Example

Consider a batch system whose model generates input and output data according to Equation (5). Each stage of batch production is regarded as a task (g). The input and output numbers are $L = 20$ and $M = 20$, respectively. The order of the system state is $N_x = 4$. The input of the system ($\mathbf{u}_{g,k}^i$) is random samples generated from the Gaussian distribution $\mathcal{N}\left(0, \sqrt{2}\mathbf{I}_L\right)$, and the noise of the system is $\boldsymbol{\varepsilon}_{g,k}^i \sim \mathcal{N}(0, 0.1\mathbf{I}_M)$. The parameter matrices $\mathbf{C}_c \in \mathbb{R}^{M \times N_q}$ and $\mathbf{B}_c \in \mathbb{R}^{N_r \times L}$ with $N_q = 3$ and $N_r = 3$ shared in all batches are randomly generated through the Gaussian distribution and orthogonalized. The parameters, $\mathbf{A}_g$, $\mathbf{R}_g$, and $\mathbf{Q}_g$ vary with different tasks (batch grades). They are randomly generated with the Gaussian distributions $\mathcal{N}\left(0, \sqrt{0.2}\right)$, $\mathcal{N}\left(0, \sqrt{1.9}\right)$, and $\mathcal{N}\left(0, \sqrt{2.8}\right)$, respectively. The initial state of each grade of the batch g is $\mathbf{x}_{g,k}^i = \mathbf{0}_{N_x}$. Assume that there are G different batch grades in the training phase. The data generation of the new batch grade $G + 1$ is still based on the same above assumptions of model parameter matrices.

To verify the proposed common feature extraction scheme for identifying the subspace model, four cases are demonstrated, including (1) a varying number of grades, (2) a varying number of data in each batch grade, (3) identifying the subspace model for a new grade, and (4) modeling performance with the different common feature parameters N_q and N_r. The LF-MoM method in [16] was adapted to calculate Equation (28), and the results were used to compare the proposed methods. The principal angle was used to evaluate the difference between the estimated common feature parameter and the actual one. The concept of principal angles between subspaces was first introduced by [20]. Then, Hotelling and Harold [21] defined the form of the canonical relationship of principal angles in the statistical theory. In this paper, the calculation method of the principal angle is based on singular value decomposition (SVD) [22,23].

When using the proposed method to extract common knowledge, the adjustable input parameters are listed in Algorithm 2, including the dynamic window size, N, and the dimensions N_q and N_r of the two common parameters.

The selection of parameter N is used to extract dynamic features of the process, which theoretically must be greater than or equal to the potential state order N_x of the model. In the case of $N \geq N_x$, the performance of the identification will not significantly improve and may even decrease. The larger N is, the more parameters must be calculated, and a higher data volume is required. In the case of fewer samples, the modeling performance will decrease. On the other hand, in the case of $N < N_x$, it cannot adequately reflect the dynamic characteristics of the process, and the resulting model cannot effectively describe the process behavior. Therefore, it is necessary to adjust N appropriately when identifying batch processes with a few samples:

(1) Varying numbers of grades

Assume that each grade has only two batch data for the common feature extraction stage. Take the number of samples $K_g = 62$ in each training batch. The batch indicator is omitted in this case. The input and output Hankel matrices $\mathbf{Y}_f$, $\mathbf{U}_f$, and $\mathbf{Z}_p$ are arranged by taking $N = 6$ and $J_g = 50$ ($J_g = K_g - N$) from the training data. Then, $\mathbf{\Psi}_g$ and $\mathbf{\Lambda}_g$ are obtained by the oblique projection.

With different grade numbers, G, the common feature parameter matrices ($\hat{\mathbf{\Omega}}$ and $\hat{\mathbf{\Theta}}$) were estimated. The sine values of the principal angles between the estimated common feature parameter matrices and the actual common feature parameter matrices are shown in Figure 1. The x-axis in Figure 1 is the number of tasks used in the common feature extraction stage, and the y-axis is the sine value of the principal angle $\sin\theta$ between the actual and the estimated common feature parameter matrices. The smaller $\sin\theta$ is, the closer the estimated and the actual parameter matrices are. The common characteristic parameters of output variables were not discussed in [17], so only the MT-L SID test results are available. The results show that the accuracy of the parameter matrix $\hat{\mathbf{\Omega}}$ can be achieved with fewer data. In the estimation of the parameter matrix $\hat{\mathbf{\Theta}}$, both the results of MT-L SID and LF-MoM are to be improved and need more training grades than the estimated parameter matrix $\hat{\mathbf{\Omega}}$, as the former is for the transmitter model, while the latter is for the emission model. More training grades are to be included for estimating a good parameter matrix $\hat{\mathbf{\Theta}}$. Furthermore, the results show that MT-L SID can achieve a slightly better performance than LF-MoM.

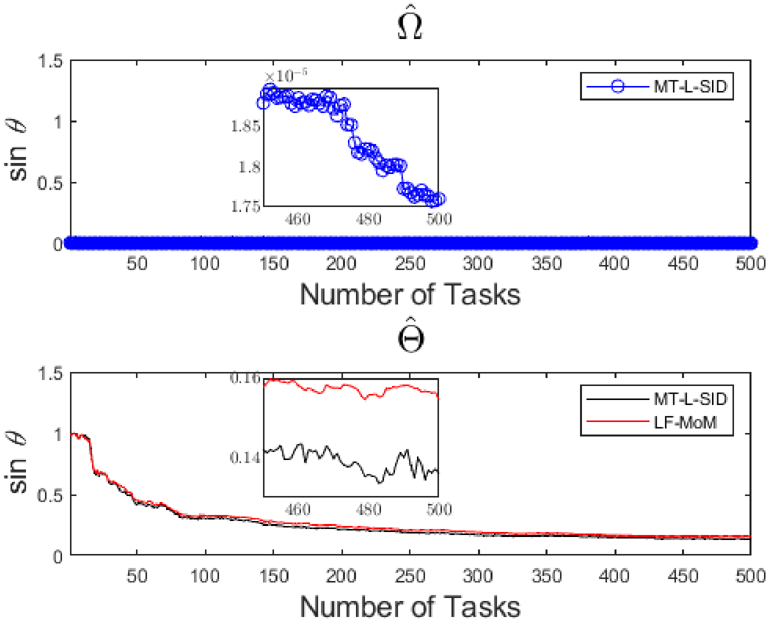


Figure 1. Sine values of principal angles between estimated and actual $\hat{\mathbf{\Omega}}$, and between estimated and actual $\hat{\mathbf{\Theta}}$, respectively, with a varying number of tasks.

(2) Varying numbers of data in each batch grade

In the second case, the number of grades was fixed at $G = 30$, but different numbers of training samples were applied. With the different samples in each batch, the corresponding dynamic windows J_g of the Hankel matrix were changed from 31 to 500, but the window size remained at $N = 6$. The principal angles between the estimated common feature parameter matrices and the actual common feature parameter matrices are shown in Figure 2. The x-axis of Figure 2 represents the number of dynamic windows contained in each training batch, and the y-axis is the sine value of the principal angle $\sin\theta$ between actual and estimated common feature parameter matrices. Similar to Case 1, the parameter matrix $\hat{\boldsymbol{\Omega}}$ with fewer data samples still can provide a good model performance. At the same grade number, increasing the number of dynamic windows can improve the accuracy of the parameter matrix $\hat{\boldsymbol{\Theta}}$ to a certain extent. In addition, $\hat{\boldsymbol{\Theta}}$ estimated from MT-L SID showed better accuracy than LF-MoM.

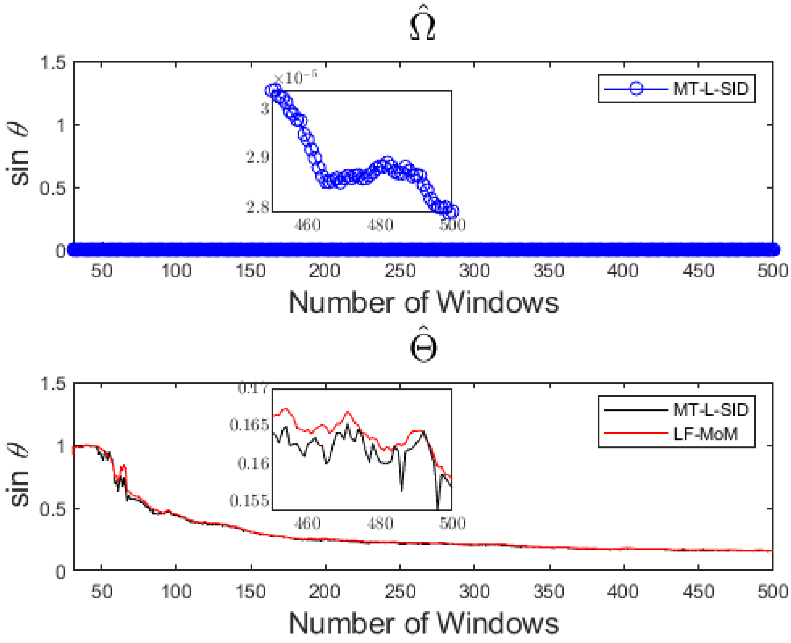


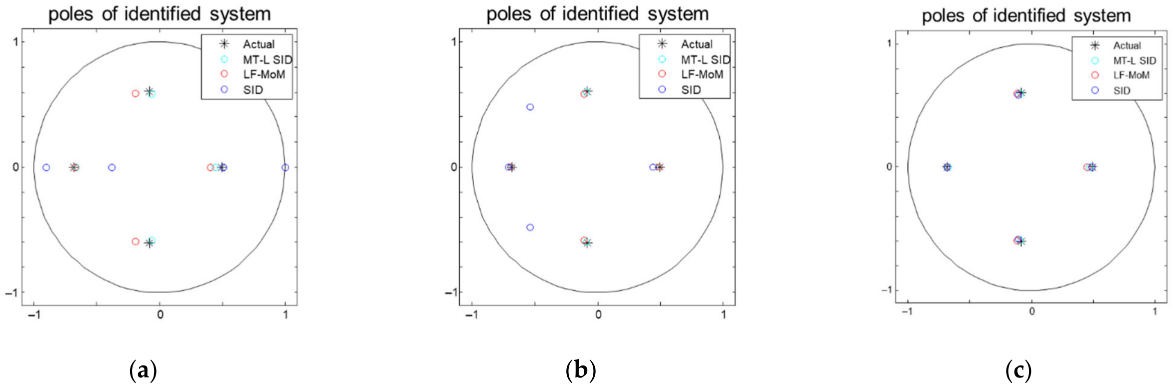
Figure 2. Sine value of the principal angle between estimated and actual $\hat{\boldsymbol{\Omega}}$, and between the estimated and actual $\hat{\boldsymbol{\Theta}}$, respectively, with varying numbers of dynamic windows.

(3) System identification of a new grade

A new grade of the product was produced through a single batch. The system model of the new grade was then identified. The new grade in the testing stage had one batch for training and one for testing. The data of one new batch were arranged into the Hankel matrix with $N = 6$. With the common feature parameter matrices $\hat{\boldsymbol{\Omega}}$ and $\hat{\boldsymbol{\Theta}}$ obtained with 500 tasks and $J = 50$, the model parameter matrices $\hat{\mathbf{O}}$ and $\hat{\boldsymbol{\Xi}}$ were identified by Equation (56); the corresponding parameter matrices ($\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, and $\hat{\mathbf{C}}$) of the state-space model can be calculated using Equations (57)–(60).

Three models, including the conventional subspace identification (SID), the SID combining the common feature parameters estimated from MT-L SID, and the SID from LF-MoM were, respectively, used for comparisons. Figure 3 shows the poles of the estimated batch parameter matrix $\hat{\mathbf{A}}$ with different numbers of samples for the three models. The black asterisk and the blue, cyan, and red circles represent the actual model, SID, MT-L SID, and SID with common feature parameters estimated from LF-MoM, respectively. In Figure 3a, when the data are insufficient ($J_{G+1} = 25$), the proposed MT-L SID outperformed other models. With the increasing number of training samples, all models were very close to the actual ones in Figure 3b with $J_{G+1} = 40$ and in Figure 3c with $J_{G+1} = 100$. As for all the case studies, the proposed learning scheme can significantly reduce the training sample requirements in each grade compared to the other modeling schemes.

In the testing stage, the demand for training data can be reduced by including common feature parameter matrices when the target batch data are insufficient. If the target batch data collected in the test phase are already sufficient, then the model directly using only the target batch data without common feature parameters can achieve accurate estimated parameters or better. In this case, adding the common feature parameters for estimation becomes unnecessary. However, at the stage of modeling a new batch process, the proposed learning scheme plays an important role.



(**a**) (**b**) (**c**)

Figure 3. Comparison of poles calculated by two different models and the actual model in different training samples: (**a**) $J_{G+1} = 25$, (**b**) $J_{G+1} = 40$, and (**c**) $J_{G+1} = 100$.

(4) Modeling performance with the different common feature parameters N_q and N_r

In the fourth case, quantity and batch grade data were considered to compare the support of selecting different common parameter structures in modeling new grades. Assume there are 100 grades of data, each with two batches of data, and each batch has an operating time of $K_g = 62$. In terms of parameter settings, take $N = 6$; $J_g = 50$. Consider the case where N_q and N_r have different dimensions. Substitute the calculated common feature parameters into the modeling of new product grade data, and further calculate the MSE from the trained model parameters on the test data as an indicator to measure the transfer performance. The calculated results are shown in Table 1.

Table 1. MSE calculation with different structures of common feature parameters.

	N_q = 1; N_r = 1	N_q = 3; N_r = 3	N_q = 5; N_r = 5	N_q = 10; N_r = 10	N_q = 15; N_r = 15
MT-L SID	7.8861×10^8	11.4305	32.2349	41.4559	3.605×10^6

From Table 1, it can be seen that the best modeling performance can be achieved when the dimensions of the common parameters are the same as the real dimensions. The dimensions N_q and N_r of the common parameters determine the amount of information transferred from historical data to new batch product modeling. If the selected dimension is greater than the potential actual dimension, then there is still information that can be transferred but not used. If it is smaller than the potential actual dimension, then it will compress the size of the structure describing individual parameters, so the description of the process in system identification is probably under-fitting. Therefore, similarly, the dimensions of common parameters also require several attempts to choose a more appropriate size.

6.2. Industrial Process for Penicillin Fermentation

A simulation of industrial-scale penicillin (IndPen) production performed by the authors of [24] was used here to validate the proposed method. The simulation was developed using the historical batch records of a 100,000 L penicillin fermentation using a high-yielding industrial strain of *Penicillium chrysogenum*, and all the available process inputs and outputs were accurately simulated. At the same time, IndPen also integrated real Raman spectroscopy equipment. In addition to modeling all required online and offline variables, IndPen also considered the growth, morphology, metabolites, and degradation of *Penicillium chrysogenum* fermentation on a large scale. The process flow sheet from [24] is shown in Figure 4. Further details of the model can be found in [24,25].

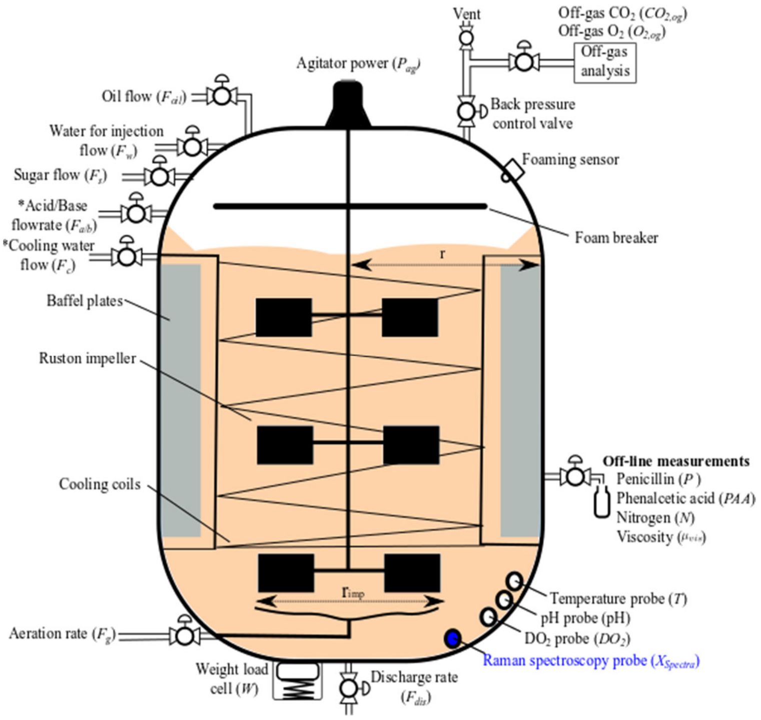


Figure 4. Flowchart of IndPen fermentation process. The * in the figure represents the variable controlled by the PID controller and the highlighted variable in blue indicates that the variable can be freely selected to be recorded and used to control PAA.

The input variables and online collected output variables of the fed-batch penicillin process are listed in Table 2. In the penicillin production process, the sampling time of the variable was 24 min. The variables were divided into two parts for control, namely automatic control variables and manual control variables. The temperature and pH were controlled by automatic control variables and regulated by a proportional integral differential (PID) feedback loop. In the manual control variables, the substrate flow rate and the phenylacetic acid flow rate were controlled by the recipe-driven method following the fixed curve of the whole batch or by the operator operating the fixed curve of the whole batch (depending on the operator). This control mode copied the control actions observed by the operator manually adjusting the whole batch of the substrate flow rate and the phenylacetic acid flow rate.

Table 2. Input and output variables of the IndPen fermentation process.

Input Variables	Output Variables
$F_{a/b}$: Acid/base flow rate	DO_2: Dissolved oxygen conc.
F_h: Heating flow rate	pH: pH
F_c: Cooling water flow rate	$T_{\tan k}$: Temperature of the tank
F_{PAA}: Phenylacetic acid flow rate	$CO_{2,og}$: Off-gas carbon dioxide
F_g: Aeration rate	W: Vessel weight
F_w: Water for injection flow rate	$O_{2,og}$: Off-gas oxygen
F_s: Substrate flow rate	
F_{oil}: Oil flow rate	

In IndPen, the operation mode was adjusted as follows.

- Changing environment temperature.
- Control strategy (recipe-driven (i.e., SBC)).

- Option to include inhibitory effects on the growth rates during DO_2, N, and PAA limitation, as well as during excessive PAA and CO_2 concentrations and sub-optimal T and pH operation.
- Whether to use Raman spectroscopy to control PAA.

According to the selection of the above modes, 14 different modes were collected. The first 10 modes were operated in ambient temperature condition 1: substrate feed temperature 288 K, substrate feed cold water 288 K, air temperature 290 K, and inlet coolant temperature 285 K. The operation mode setting under these temperature conditions is shown in Table 3. The operation mode setting under temperature condition 2—substrate feed temperature 293 K, substrate supply cold water 293 K, air temperature 298 K, and inlet coolant temperature 288 K—is shown in Table 3.

Table 3. Batch runs of product of each operation mode at ambient temperature condition 1.

No.	Control Strategy	Raman Spectroscopy	Inhibition
1	Sequential batch control	Only record the Raman data	DO_2, T, pH, CO_2, PAA, N
2	Operator controller batch	Only record the Raman data	DO_2, T, pH, CO_2, PAA, N
3	Sequential batch control	Use Raman data to control PAA	DO_2, T, pH, CO_2, PAA, N
4	Operator controller batch	Use Raman data to control PAA	DO_2, T, pH, CO_2, PAA, N
5	Sequential batch control	Use Raman data to control PAA	No inhibition
6	Sequential batch control	Use Raman data to control PAA	DO_2, T, pH
7	Operator controller batch	Use Raman data to control PAA	DO_2, T, pH
8	Operator controller batch	Use Raman data to control PAA	No inhibition
9	Operator controller batch	Only record the Raman data	No inhibition
10	Sequential batch control	Only record the Raman data	No inhibition
11	Sequential batch control	Only record the Raman data	DO_2, T, pH, CO_2, PAA, N
12	Operator controller batch	Only record the Raman data	DO_2, T, pH, CO_2, PAA, N
13	Sequential batch control	Use Raman data to control PAA	DO_2, T, pH, CO_2, PAA, N
14	Operator controller batch	Use Raman data to control PAA	DO_2, T, pH, CO_2, PAA, N

According to the variables in Table 2, the input and output profiles of mode 1 are shown in Figures 5 and 6, respectively.

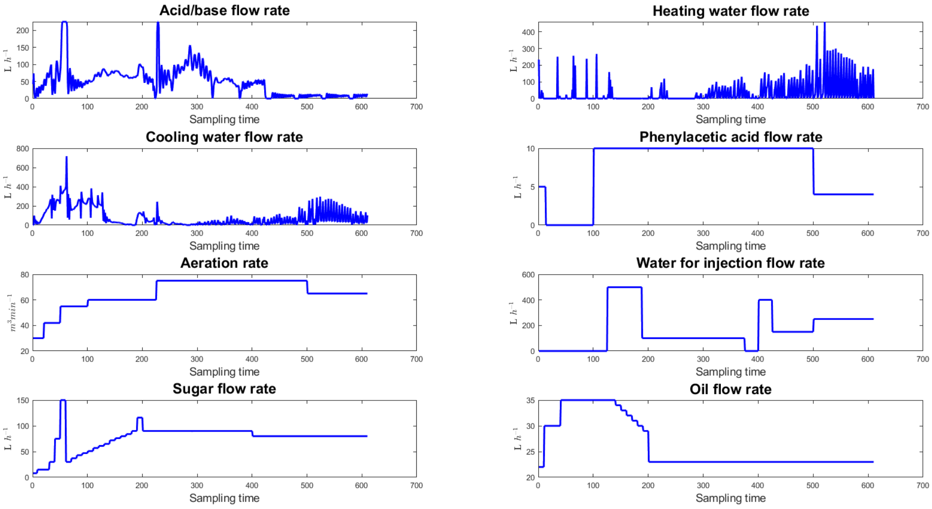


Figure 5. The profile of input variables.

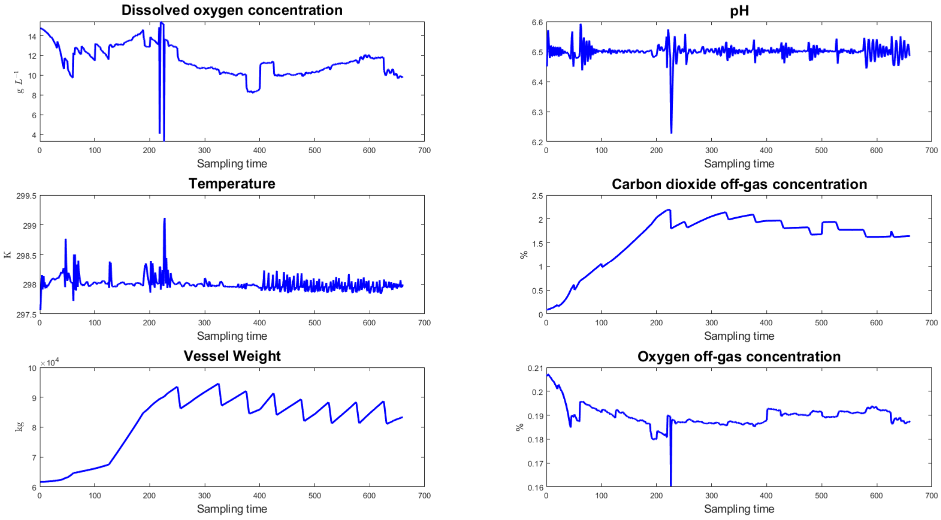


Figure 6. The profile of output variables.

In the case of a small sample of batch data, operation mode 10 was selected for validation. The common knowledge was extracted from the data of other modes excluding the test. The proposed MT-L SID was compared with the combination of LF-MoM and SID and the conventional SID.

- Few-shot learning in mode 10 data

Considering the modeling of fed-batch penicillin process in operation mode 10, the common feature extraction involved using the batch data in other modes, and each of them was five batches. The dimensions of the common feature parameter matrices with $N = 36$ were $N_r = 5$ and $N_q = 5$. In the testing stage, nine batch data from mode 10 were used for training, and the rest of the batch was used for verification. The results are shown in Figure 7.

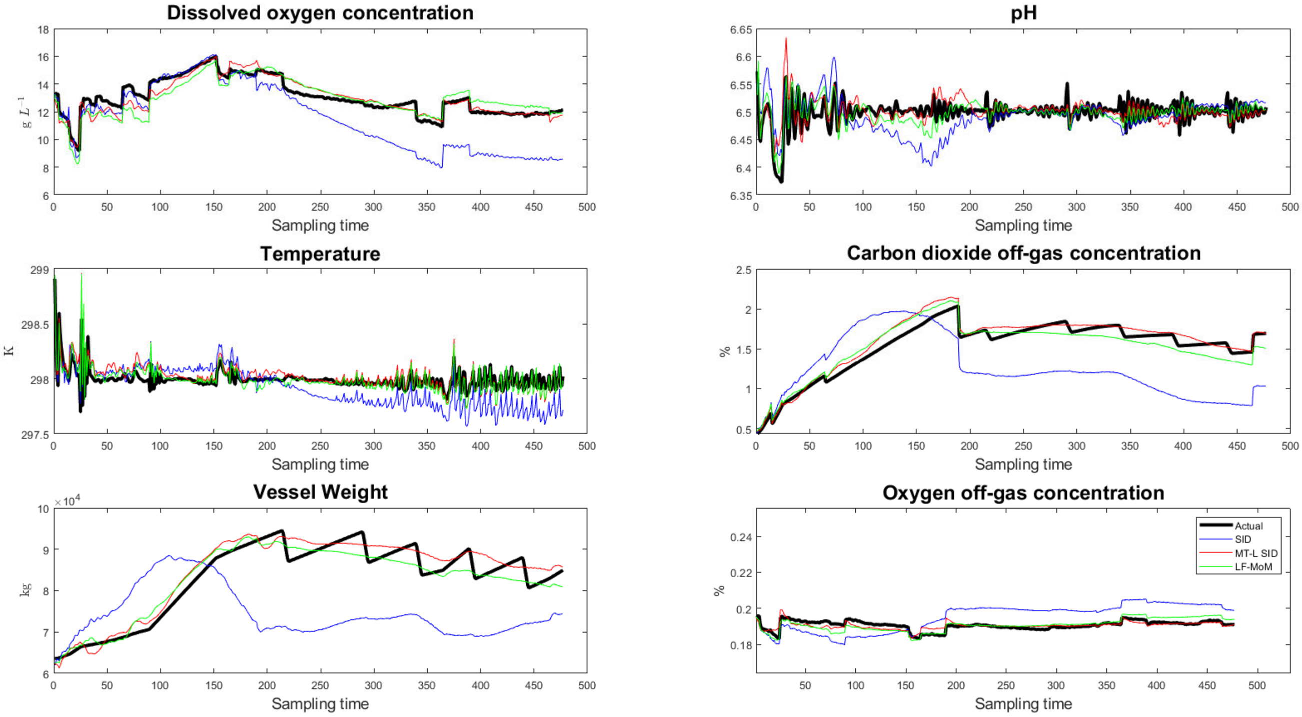


Figure 7. Predicted outputs in operation mode 10.

In Figure 7, the black lines are the actual outputs, the blue lines are the predicted outputs without using common feature parameter matrices for modeling, the red lines are the predicted outputs of the MT-L SID, and the green lines are the predicted outputs of the model with LF-MoM. Although the outputs of the red lines do not completely follow the outputs well, the proposed learning model can better describe the process behaviors than the model established without using common feature parameter matrices and SID with LF-MoM. The MSE of these methods with five operation conditions (modes) are shown in Table 4. In Table 4, the process variables are scaled to the same size. The results show that in few-shot batch modeling, the performance of the proposed MT-L SID was helpful and better than LF-MoM and conventional SID.

Table 4. MSEs of SID, MT-L SID, and LF-MoM models in the IndPen fermentation process case.

	SID	LF-MoM	MT-L SID
Mode 1	282.2638	2.8839	2.6048
Mode 2	29.2447	15.1013	11.7451
Mode 3	86.4948	101.8927	14.5548
Mode 4	12.4861	4.5348	3.9649
Mode 10	12.2479	2.4549	2.3101

7. Conclusions

The data collected from the industrial batch process sometimes encounter the low-N problem. Such a small number of batch-run data will greatly degrade the performance of data-driven models. In this study, multitask-learning SID was proposed for modeling batch processes, which can transfer knowledge from multiple batch process data. The proposed method uses historical batch data with similar multiple batch production to train the state-space model and extract common feature parameter matrices. Then, during modeling, a batch with a new grade, as well as the input and output variables of the historical batch data, is projected into the common feature space to reduce the sample requirements for the training of new model parameters. Thus, the model of a new batch with a few batch data can be effectively and quickly constructed. In addition, the purpose of using oblique projection in our method is to avoid solving the common parameters of two objective functions simultaneously during the solution process. This technique can decouple the two parameters so that the solutions of the two parameters do not affect each other. According to the objective functions we have defined, the derivation result of solving the common parameters corresponds to solving the eigenvalue problem. The common parameters correspond to the maximum eigenvalue of the data matrix after the oblique projection transformation. The contributions of this study are summarized as follows:

- A modified linear time-invariant state-space model was proposed, which separates the common features and individual features corresponding to the input and input parameters so that it can be used to express the extraction of common features in the multi-input, multi-output dynamic system.
- The multi-task learning-based N4SID, called MT-L SID, was developed. The proposed SID model of the common feature parameters was learned from historical batch data. It can be effectively applied to the batch process with new grades. Then, the data requirement of a new batch process modeling can be reduced.
- With the oblique projection, MT-L SID can separate input-output equations to calculate the common feature parameter matrices. The calculations are straightforward without heavy iterations.

Finally, the proposed method was validated based on a numerical example and an industrial penicillin production process. In the generated numerical examples, increasing the number of batches (tasks) or the amount of data in individual batches improved the accuracy of common feature parameters. In the production of penicillin, the proposed method showed better prediction performance in the case of a few samples.

Author Contributions: Conceptualization, S.G., J.C. and L.X.; formal analysis, S.G. and J.C.; investigation, S.G., J.C. and L.X.; methodology, S.G. and J.C.; resources, S.G. and J.C.; software, S.G.; supervision, S.G., J.C. and L.X.; visualization, S.G.; writing—original draft, S.G. and J.C.; writing—review and editing, S.G. and J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Technology, Taiwan (MOST 109-2221-E-033-060-MY3), and National Natural Science Foundation of P.R. China (62073286).

Data Availability Statement: Data are available upon request due to restrictions privacy. The data presented in this study are available upon request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

$\mathcal{D}_g$	The data set of batch grade g
I_g	Number of batches in grade g
N	Dynamic window size
$\mathbf{x}^i_{g,k}$	System state of batch i in grade g at time k
$\mathbf{u}^i_{g,k}$	System input of batch i in grade g at time k
$\mathbf{y}^i_{g,k}$	System output of batch i in grade g at time k
$\boldsymbol{\varepsilon}^i_{g,k}$	Additional noise of batch i in grade g at time k
$\mathbf{B}_g$	Parameter of grade g
$\mathbf{R}_g$	Individual part of parameter $\mathbf{B}_g$
$\mathbf{C}_c$	Common part of parameter $\mathbf{C}_g$
$\mathbf{B}_c$	Common part of parameter $\mathbf{B}_g$
$\boldsymbol{\Omega}$	Extended of parameter $\mathbf{C}_c$
$\boldsymbol{\lambda}^i_{g,j}$	$\mathbf{y}^i_{g,f,j}$ projection onto $\mathbf{u}^i_{g,f,j}$ along the direction parallel to $\mathbf{z}^i_{g,j}$
$\boldsymbol{\Psi}^B_g$	Total batch data of $\boldsymbol{\varphi}^i_{g,j}$
$\boldsymbol{\Lambda}^B_g$	Total batch data of $\boldsymbol{\lambda}^i_{g,j}$
G	Number of grades
K^i_g	Operation time of batch i in grade g
J^i_g	Number of moving windows
N_x	Number of system states
L	Number of system inputs
M	Number of system outputs
$\mathbf{A}_g$	Parameter of grade g
$\mathbf{C}_g$	Parameter of grade g
$\mathbf{Q}_g$	Individual part of parameter $\mathbf{C}_g$
N_q	Dimention of parameter $\mathbf{C}_c$
N_r	Dimention of parameter $\mathbf{B}_c$
$\boldsymbol{\Theta}$	Extended of parameter $\mathbf{B}_c$
$\boldsymbol{\varphi}^i_{g,j}$	$\mathbf{y}^i_{g,f,j}$ projection onto $\mathbf{z}^i_{g,j}$ along the direction parallel to $\mathbf{u}^i_{g,f,j}$
$\mathbf{Z}^B_{g,p}$	Total batch of $\mathbf{z}^i_{g,j}$
$\mathbf{U}^B_{g,f}$	Total batch of $\mathbf{u}^i_{g,f,j}$

References

1. Song, H.; Xu, R.; Wang, C. Research on statistical process control method for multi-variety and small batch production mode. In Proceedings of the 2020 Chinese Control And Decision Conference (CCDC), Hefei, China, 22–24 August 2020; pp. 2377–2381.
2. Wang, J.; Zhang, T.; Wang, C.; Shi, X. Optimizing the Uncertainty of PPM on Small Batch of Quality Data. In Proceedings of the 2021 IEEE 6th International Conference on Smart Cloud (SmartCloud), Newark, NJ, USA, 6–8 November 2021; pp. 107–110.
3. Cao, Y.; Feng, Z.; Jiang, Q. Automatic Data Acquisition Technology for SMT Manufacturing Based on Multi-Variety and Small-Batch. In *Proceedings of the Seventh Asia International Symposium on Mechatronics*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 528–537. [CrossRef]
4. Li, Q.; Wei, F.; Zhou, S. Early warning systems for multi-variety and small batch manufacturing based on active learning. *J. Intell. Fuzzy Syst.* **2017**, *33*, 2945–2952. [CrossRef]
5. Cui, L.; Peng, Y.; Ding, L.; Lu, D. An improved batch fluidized drying experimental design based on digital sensors and a minicomputer. *Eng. Rep.* **2021**, *3*, e12366. [CrossRef]

6. Tulsyan, A.; Garvin, C.; Ündey, C. Advances in industrial biopharmaceutical batch process monitoring: Machine-learning methods for small data problems. *Biotechnol. Bioeng.* **2018**, *115*, 1915–1924. [CrossRef] [PubMed]
7. Tulsyan, A.; Garvin, C.; Undey, C. Industrial batch process monitoring with limited data. *J. Process Control* **2019**, *77*, 114–133. [CrossRef]
8. Weiss, K.; Khoshgoftaar, T.M.; Wang, D. A survey of transfer learning. *J. Big Data* **2016**, *3*, 9. [CrossRef]
9. Zhang, D.; Del Rio-Chanona, E.A.; Petsagkourakis, P.; Wagner, J. Hybrid physics-based and data-driven modeling for bioprocess online simulation and optimization. *Biotechnol. Bioeng.* **2019**, *116*, 2919–2930. [CrossRef] [PubMed]
10. Jaeckle, C.M.; MacGregor, J.F. Product transfer between plants using historical process data. *AIChE J.* **2000**, *46*, 1989–1997. [CrossRef]
11. Muñoz, S.G.; MacGregor, J.F.; Kourti, T. Product transfer between sites using Joint-Y PLS. *Chemom. Intell. Lab. Syst.* **2005**, *79*, 101–114. [CrossRef]
12. Rudnitskaya, A.; Costa, A.M.S.; Delgadillo, I. Calibration update strategies for an array of potentiometric chemical sensors. *Sens. Actuators B Chem.* **2017**, *238*, 1181–1189. [CrossRef]
13. Jia, R.; Zhang, S.; You, F. Transfer learning for end-product quality prediction of batch processes using domain-adaption joint-Y PLS. *Comput. Chem. Eng.* **2020**, *140*, 106943. [CrossRef]
14. Chu, F.; Zhao, X.; Yao, Y.; Chen, T.; Wang, F. Transfer learning for batch process optimal control using LV-PTM and adaptive control strategy. *J. Process Control* **2019**, *81*, 197–208. [CrossRef]
15. Yamaguchi, T.; Yamashita, Y. Quality prediction for multi-grade batch process using sparse flexible clustered multi-task learning. *Comput. Chem. Eng.* **2021**, *150*, 107320. [CrossRef]
16. Tripuraneni, N.; Jin, C.; Jordan, M. Provable meta-learning of linear representations. In Proceedings of the International Conference on Machine Learning, Vienna, Austria, 18–24 July 2021; pp. 10434–10443.
17. Lu, J.; Yao, K.; Gao, F. Process similarity and developing new process models through migration. *AIChE J.* **2009**, *55*, 2318–2328. [CrossRef]
18. Behrens, R.T.; Scharf, L.L. Signal processing applications of oblique projection operators. *IEEE Trans. Signal Process.* **1994**, *42*, 1413–1424. [CrossRef]
19. Li, Z.; Nie, F.; Chang, X.; Yang, Y. Beyond trace ratio: Weighted harmonic mean of trace ratios for multiclass discriminant analysis. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2100–2110. [CrossRef]
20. Jordan, C. Essai sur la géométrie à n dimensions. *Bull. Soc. Math. Fr.* **1875**, *3*, 103–174. [CrossRef]
21. Hotelling, H. Relations between two sets of variates. In *Breakthroughs in Statistics*; Springer: Berlin/Heidelberg, Germany, 1992; pp. 162–190. [CrossRef]
22. Björck, Å.; Golub, G.H. Numerical methods for computing angles between linear subspaces. *Math. Comput.* **1973**, *27*, 579–594. [CrossRef]
23. Golub, G.H.; Zha, H. The canonical correlations of matrix pairs and their numerical computation. In *Linear Algebra for Signal Processing*; Springer: Berlin/Heidelberg, Germany, 1995; pp. 27–49. [CrossRef]
24. Goldrick, S.; Ştefan, A.; Lovett, D.; Montague, G.; Lennox, B. The development of an industrial-scale fed-batch fermentation simulation. *J. Biotechnol.* **2015**, *193*, 70–82. [CrossRef] [PubMed]
25. Goldrick, S.; Duran-Villalobos, C.A.; Jankauskas, K.; Lovett, D.; Farid, S.S.; Lennox, B. Modern day monitoring and control challenges outlined on an industrial-scale benchmark fermentation process. *Comput. Chem. Eng.* **2019**, *130*, 106471. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.