Korean J. Chem. Eng., 40(5), 1023-1036 (2023)
DOI: 10.1007/s11814-022-1335-5

pISSN: 0256-1115
eISSN: 1975-7220

Soft sensor development based on just-in-time learning and dynamic time warping for multi-grade processes

Min Jun Song, Sung Hyun Ju, and Jong Min Lee†

School of Chemical and Biological Engineering, Institute of Chemical Processes, Seoul National University, Gwanak-ro 1, Gwanak-gu, Seoul 08826, Korea
(*Received 28 July 2022 • Revised 31 October 2022 • Accepted 6 November 2022*)

Abstract−This study presents the development of soft sensors based on just-in-time learning (JITL) and dynamic time warping (DTW) for online quality prediction in multi-grade processes. Most industrial chemical processes are multi-grade processes that produce multiple products with distinct properties. Multi-grade processes, however, are difficult to monitor and control due to frequent process transitions and abrupt changes in operating conditions. The DTW-based JITL soft sensor modeling approach is proposed as a solution to the complexity of multi-grade process modeling. In the JITL modeling approach, a local model is trained online using historical samples that are similar to the query sample, allowing the model to account for multi-grade characteristics and process drifts. To account for process dynamics and temporal correlations, the suggested approach utilizes a data sequence as an input rather than a single data point. DTW calculates the similarity of data sequences by stretching the sequences to determine an optimal warping path. Additionally, sensitivity analyses of model hyperparameters are performed and a cross-correlation-based hyperparameter optimization approach is proposed. The advantages of the proposed approach are verified via multi-grade simulation studies. As a result, the proposed model outperforms a conventional JITL model based on the Euclidean distance.

Keywords: Just-in-Time Learning (JITL), Dynamic Time Warping (DTW), Soft Sensor, Multi-Grade Process, Machine Learning

INTRODUCTION

There is an increasing market need in the chemical sector for the manufacturing of a variety of chemical products with specific applications. To satisfy fluctuating market demand, many chemical processes produce multiple grades of products with distinct qualities within the same facility. For example, polymer products are manufactured in a variety of grades with varying chemical and mechanical properties, such as melt index, tensile strength, and transparency [1-7]. Due to the frequent grade transitions in multi-grade operations, fast changes in operating conditions are unavoidable. As a result, large settling times and overshoots in product quality occur during process transitions, resulting in off-specification products [1-5]. Additionally, online measurements of the quality of chemical products are not available in most industrial processes [1-6]. Therefore, a soft sensor model with high prediction accuracy is necessary for quality monitoring and control of multi-grade processes.

There are two main types of soft sensor modeling approaches: mechanistic modeling and data-driven modeling. The mechanistic models require process knowledge, including reaction mechanisms and thermodynamic properties. However, development of an accurate mechanistic model for a multi-grade process is extremely difficult due to the process's strong nonlinearity and time-varying dynamics.

Data-driven approaches, as opposed to mechanistic modeling approaches, simply utilize process data and quality measurements to construct a soft sensor. Due to the rapid growth of data technology, it has become possible to collect massive quantities of high-dimensional process data from industrial processes. To construct soft sensor models for the quality prediction of chemical processes, multiple data-driven modeling approaches have been implemented. Latent variable methods such as principal component analysis (PCA) [8-10] and partial least squares (PLS) [11-16], which transform process variables into linearly independent latent variables, are among the most common data-driven modeling approaches. Additionally, machine learning modeling approaches based on support vector machines (SVM) [15-17], Gaussian process regression (GPR) [18-20], artificial neural network (ANN) [15,21,22], and long short-term memory (LSTM) network [23,24] have been used to predict the quality of chemical and biochemical processes.

However, multi-grade process modeling presents various issues that are difficult for a single global model to address [2-7]. First, one grade's operating conditions are distinct from those of other grades. As the number of product grades increases, it becomes more challenging for a single soft sensor to precisely mimic all process variables for each grade. When estimating the quality of a new product grade that was not included in the training dataset, for instance, a model's prediction performance could be drastically reduced [6]. Furthermore, industrial processes exhibit time-varying dynamics due to the drift of process characteristics, such as catalyst deactivation, instrument degradation, and fouling. These process drifts result in the gradual degradation of the prediction performance of a soft sensor model [7]. The wide disparity between the amount of samples of each grade poses an additional difficulty for training a global

†To whom correspondence should be addressed.
E-mail: jongmin@snu.ac.kr

soft sensor model. To build an accurate model of a soft sensor, sufficient samples of each grade are necessary. However, the measurement data from a specific grade may be very limited because the operating mode changes based on market demands. Due to the short duration of a grade changeover, only a small number of measurement samples are available for simulating the transient dynamics.

Therefore, numerous investigations have been conducted to build soft sensors that can account for multi-grade and time-varying characteristics of chemical processes. One approach is to iteratively update a soft sensor model with new measurement data. Examples include recursive PCA [25-27] and recursive PLS [28-30], which are based on latent variables method. While recursive models have demonstrated enhanced prediction performance for systems with slow time-varying dynamics, they are not suitable for multi-grade chemical processes where abrupt and rapid changes in operating conditions occur during grade changeover [2,7,31-36].

In recent years, just-in-time learning (JITL) soft sensors have been used in a variety of chemical processes and have exhibited good prediction performance with various data-driven modeling methods, including latent variable methods [35,37-39], SVR [2,3, 32], GPR [33,40,41], relevance vector machines [42], LSTM [43], and autoencoder [36,44-46]. In the JITL framework, a local model is trained online using only the most similar historical data samples to the query sample. Thus, the similarity measurement employed in JITL modeling has a substantial effect on the performance of a model. One of the most widely used metrics for calculating the similarity between two samples is the Euclidean distance. Additionally, the information regarding the angle between two samples was combined with the Euclidean distance to determine similarity [34]. JITL soft sensors have embraced additional similarity metrics, including the Mahalanobis distance [35,43] and the Kullback-Leibler divergence [36,45,46]. However, temporal correlations inherent in process data are not considered in the computations of similarity described above. The process data from industrial chemical processes exhibit highly nonlinear, complicated, and temporally correlated dynamics. To build an accurate JITL soft sensor for multi-grade chemical processes requires a similarity measurement that takes temporal correlations of process variables into account.

In this paper, we present a JITL soft sensor modeling framework where the similarity between two multivariate time series data is calculated using dynamic time warping (DTW). DTW is a similarity measurement method for two different data sequences [47,48]. The similarity between two time series of different lengths can be calculated using DTW by stretching or compressing a times series to match another. In recent years, DTW has been utilized for a variety of dynamic time series problems, including handwriting recognition [49], gesture recognition [50], traffic speed prediction [51], state of health estimation [52], fault detection [53], and batch trajectory synchronization [54].

Three main contributions are accomplished in the proposed DTW-based JITL modeling approach. First, the process dynamics and temporal correlations in the process data are considered by training a machine learning model with time series data as opposed to data points. Previous studies have shown that the prediction performance of latent variable models and machine learning models can be improved by augmenting input data with time-lagged data [55,56]. Second, the ability of selecting the most relevant time series from the historical database is improved by utilizing DTW as a similarity measurement. Third, the implications of the DTW path constraint and input-output cross-correlation on the modeling performance are discussed and a DTW-based JITL soft sensor modeling approach with the hyperparameter optimization is proposed. The prediction performance of the proposed DTW-based JITL soft sensor is evaulated with three multi-grade dynamic simulation studies and compared to that of the conventional JITL method based on the Euclidean distance.

The remaining sections of this work are structured as follows. Section 2 introduces the approaches applied in the proposed modeling approach, including JITL modeling and DTW. In Section 3, a DTW-based JITL soft sensor modeling method for multi-grade processes is proposed. Section 4 provides the results of sensitivity analysis of the hyperparameters of the proposed modeling approach. In Section 5, a DTW-based JITL modeling approach with the optimized model hyperparameters is proposed and the modeling results is discussed. Finally, Section 6 contains the concluding remarks.

METHODOLOGY

1. Just-in-time Learning Modeling

Just-in-time learning (JITL) modeling approach is one of the

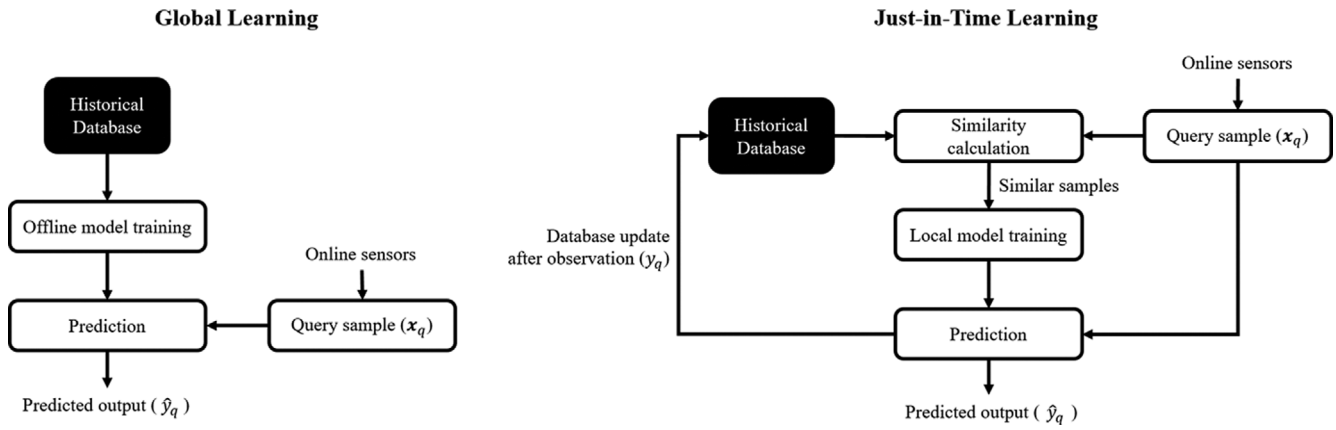

Fig. 1. Basic algorithms of classic global learning and just-in-time learning approaches.

modeling methods developed to address process drift and multi-grade characteristics. In JITL modeling, rather than training a single global model with all historical data, a local model is trained in an online manner for each query sample using the most similar samples. The trained local model is then discarded after the prediction for the query sample has been made using the local model.

The fundamental algorithm of JITL soft sensor modeling consists of three steps as shown in Fig. 1. When a new query is received, the similarities between the query sample and the stored samples are first calculated. The second step involves training a local prediction model using only the most relevant samples. Finally, the quality of a sample is predicted using the local model trained in the second step. The above three steps of modeling are repeated every time a new data sample is collected.

Since a local model is trained with a similar set of data samples only, the modeling performance of a JITL model highly depends on how similar samples are selected from the database. The Euclidean distance is one of the most often employed similarity metrics for two data samples. The Euclidean distance, ED, between a query sample $\mathbf{x}_q$ and a sample $\mathbf{x}_n$ from the database is calculated as in (1).

$$\mathrm{ED}(\mathbf{x}_q, \mathbf{x}_n) = \sqrt{\sum_{i=1}^{n_{var}} (x_{q,i} - x_{n,i})^2} \tag{1}$$

where n_{var} is the number of variables of the vector $\mathbf{x}$. $x_{q,i}$ and $x_{n,i}$ are the i-th variable of $\mathbf{x}_q$ and $\mathbf{x}_n$, respectively.

Another method used to measure similarity in JITL modeling is to consider both the distance and angle between two samples. The cosine of the angle between two samples, $\mathbf{x}_q$ and $\mathbf{x}_n$, is calculated as in (2).

$$\cos(\theta_{qn}) = \frac{\langle \mathbf{x}_q, \mathbf{x}_n \rangle}{\|\mathbf{x}_q\|_2 \|\mathbf{x}_n\|_2} \tag{2}$$

where θ_{qn} is the angle between $\mathbf{x}_q$ and $\mathbf{x}_n$. Two data samples are considered far apart when the cosine value of the angle between them is negative. In training local models, only data samples with positive cosine values are employed.

2. Dynamic Time Warping

Dynamic time warping (DTW) is a measure of distance between two data sequences. One of the main advantages of DTW is that the similarity between two time series of different lengths can be measured using DTW [47-54]. For example, Fig. 2 illustrates the alignment of two time series of sine functions of different frequencies and delays using DTW.

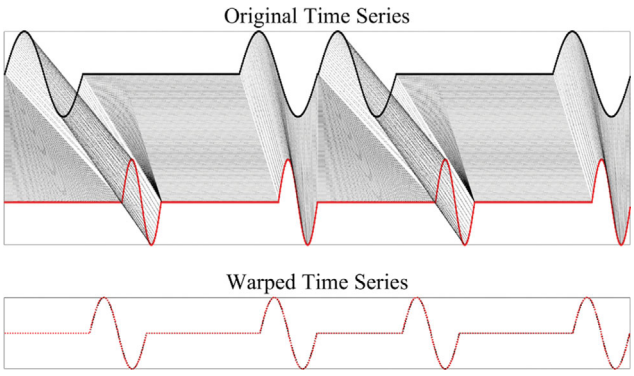

Fig. 2. Example of warping two time series using DTW.

For two time series, $X=\{x_1, x_2, \ldots, x_m\}$ of length m and $Y=\{y_1, y_2, \ldots, y_n\}$ of length n, the warping path $W=\{w_1, w_2, \ldots, w_k\}$ of length k must satisfy the following constraints. First, the boundary conditions at the initial and final points must be satisfied.

$$w_1 = (1, 1) \tag{3}$$

$$w_k = (m, n) \tag{4}$$

where $w_i=(a, b)$ indicates that $X(a)=x_a$ corresponds to $Y(b)=y_b$ in the path W. Second, the warping path must satisfy the continuity condition. If $w_i=(a, b)$ and $w_{i+1}=(a', b')$, then $a'-a\leq 1$ and $b'-b\leq 1$. Finally, the warping path must be monotonic, which limits the warping path from moving backwards in time. If $w_i=(a, b)$ and $w_{i+1}=(a', b')$, then $a'-a\geq 0$ and $b'-b\geq 0$. The continuity and monotonicity conditions can be combined and expressed as follows:

$$0\leq a'-a\leq 1,\ a, a'\in\mathbb{N} \tag{5}$$

$$0\leq b'-b\leq 1,\ b, b'\in\mathbb{N} \tag{6}$$

As a result, $w_{i+1}=(a+1, b)$, $(a, b+1)$, or $(a+1, b+1)$. However, there are a number of warping paths satisfying the above constraints. DTW finds the optimal warping path that has minimum cumulative distance.

$$\mathrm{DTW}(X, Y) = \min_{W} \sum_{i=1}^{k} d_i \tag{7}$$

where d_i is distance between $X(a)=x_a$ and $Y(b)=y_b$ when $w_i=(a, b)$.

Another constraint on the warping path may be introduced in order to reduce computational requirement of DTW [47,48]. By limiting the warping window width, δ, the number of possible warping paths is reduced. Additionally, the window width constraint makes the resulting warping path closer to diagonal connecting $W_1=(1, 1)$ and $W_k=(m, n)$, preventing excessive stretching and compression of sequences when outliers are present.

DTW-BASED JITL MODELING APPROACH

In this section, a JITL soft sensor modeling method for multi-grade processes is developed by applying DTW to calculating similarities between data sequences. There are two main advantages of the proposed modeling approach. First, the data sequence is used as a model input in order to consider process dynamics and temporal correlations in the process data. By augmenting the input variables with time-lagged data, the modeling performance of a data-driven model for a dynamic system may be increased. Second, the similarity between two data sequences is measured using DTW.

The similarity measurements previously studied in JITL models, including DTW, are summarized in Table 1. The most common way is to use the Euclidean distance and angle between two data samples because they are simple to calculate. However, the temporal correlations in the data are disregarded when the Euclidean distance is used to compare similarities. Recently, there have been studies to build a JITL soft sensor model with similarity calculation methods based on the statistical information of data points. For instance, the Mahalanobis distance has been adopted to JITL

Table 1. Comparison of distance measure used in JITL modeling

Method	Data type	Definition
Euclidean distance	Data point	$\mathrm{ED}(\mathbf{x}_q, \mathbf{x}_n)=\sqrt{\sum_{i=1}^{n_{var}}(\mathrm{x}_{q,i}-\mathrm{x}_{n,i})^2}$
Euclidean distance with angle	Data point	$\cos(\theta_{qn})=\frac{\langle \mathbf{x}_q, \mathbf{x}_n\rangle}{\|\mathbf{x}_q\|_2\|\mathbf{x}_n\|_2}$
Mahalanobis distance	Data point and distribution	$\mathrm{MD}(\mathbf{x}, \mathbf{y}; \mathbf{P})=\sqrt{(\mathbf{x}-\mathbf{y})^T\mathrm{S}^{-1}(\mathbf{x}-\mathbf{y})}$
Kullback-Leibler divergence	Distribution	$\mathrm{KLD}(\mathrm{P}\|\|\mathrm{Q})=\int_{-\infty}^{\infty}\mathrm{p}(\mathrm{x})\frac{\log(\mathrm{p}(\mathrm{x}))}{\log(\mathrm{q}(\mathrm{x}))}$
Dynamic time warping	Data sequence	$\mathrm{DTW}(\mathrm{X}, \mathrm{Y})=\min_{W}\sum_{i=1}^{k}\mathrm{d}_i$

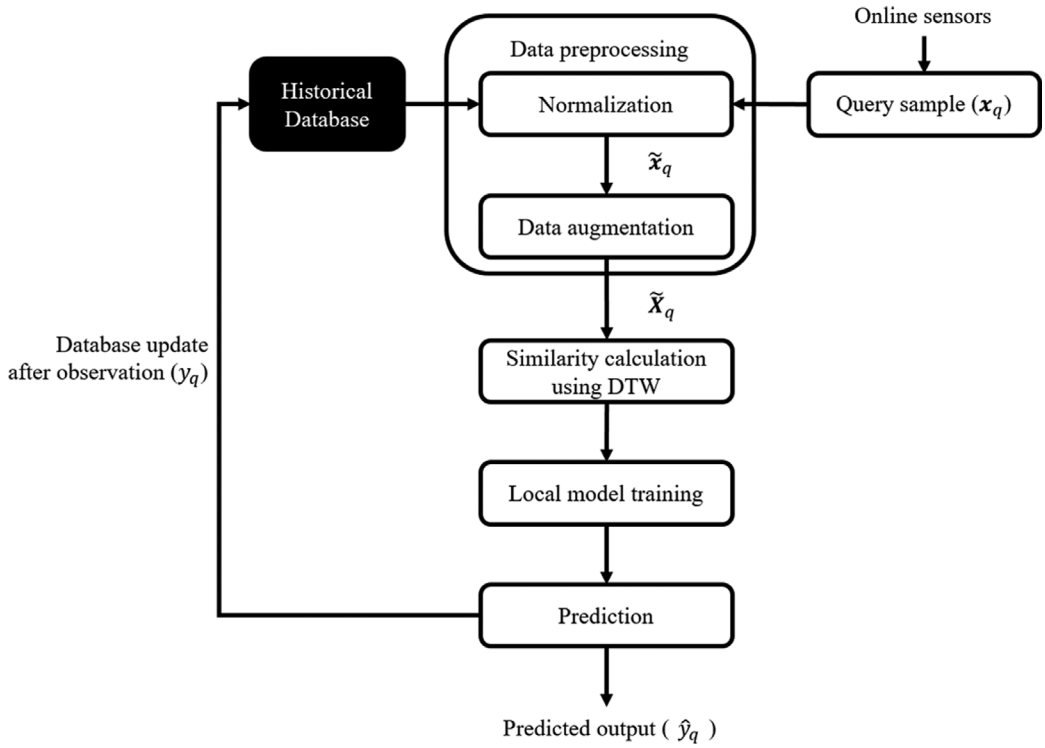

Fig. 3. Algorithm of the DTW-based JITL model.

modeling [35,43]. In contrast to the Euclidean distance, which only considers two data points, the Mahalanobis distance measures the distance between a point and a distribution by considering sample mean and covariance. The Mahalanobis distance calculates the inverse-covariance weighted distance between a point and sample mean. The following definition describes the Mahalanobis distance between a point **x** and a distribution **P**.

$$\mathrm{MD}(\mathbf{x}, \mathbf{P})=\sqrt{(\mathbf{x}-\mu)^T\mathrm{S}^{-1}(\mathbf{x}-\mu)} \tag{8}$$

where μ and S are the sample mean and covariance matrix of the distribution **P**, respectively. Additionally, the Mahalanobis distance between two data points, **x** and **y**, with respect to a distribution **P** is calculated as in (9).

$$\mathrm{MD}(\mathbf{x}, \mathbf{y}; \mathbf{P})=\sqrt{(\mathbf{x}-\mathbf{y})^T\mathrm{S}^{-1}(\mathbf{x}-\mathbf{y})} \tag{9}$$

Kullback-Leibler divergence, commonly known as relative entropy, is an additional similarity measure based on probability statistics. It is utilized to determine how dissimilar one distribution is to another distribution and is defined as the expectation of the logarithmic difference between two distributions. For two continuous probability distributions, P and Q, the Kullback-Leibler divergence is calculated as in (10) by

$$\mathrm{KLD}(\mathrm{P}\|\|\mathrm{Q})=\int_{-\infty}^{\infty}\mathrm{p}(\mathrm{x})\frac{\log(\mathrm{p}(\mathrm{x}))}{\log(\mathrm{q}(\mathrm{x}))} \tag{10}$$

where p and q are the probability densities of P and Q, respectively. Thus, the probability distributions of variables are required for calculation of Kullback-Leibler divergence. Therefore, there have been studies of JITL soft sensors where the process variables are first transformed into latent variables that can be expressed by Gaussian distribution [36,45,46].

On the other hand, DTW measures the distance between two

data sequences without any probabilistic information or transformation of variables. Additionally, DTW can calculate similarity even when a data sequence is distorted by frequency change or process drift. Therefore, in the suggested JITL modeling method, DTW is utilized to calculate the similarity of data sequences.

Fig. 3 illustrates the fundamental algorithm of the DTW-based JITL modeling technique. First, the query sample, $\mathbf{x}_q$, is collected from online sensors. Then the historical and query data are normalized so that the variables have zero means and unit variances, as described below.

$$\tilde{x}_{q,i}=\frac{x_{q,i}-\mu_i}{\sigma_i} \tag{11}$$

where $x_{q,i}$ is the i-th variable of x_q, μ_i and σ_i are the mean and standard deviation of the i-th variable, respectively. The next step is to augment a single data sample $\tilde{\mathbf{x}}_q$ into a data sequence $\tilde{\mathbf{X}}_q$ with time-lagged data as in (12).

$$\tilde{\mathbf{X}}_q=[\tilde{\mathbf{x}}_{q-n+1}, \tilde{\mathbf{x}}_{q-n+2}, \ldots, \tilde{\mathbf{x}}_q]$$
$$=\begin{bmatrix} \tilde{\mathbf{x}}_{q-n+1,1} & \tilde{\mathbf{x}}_{q-n+2,1} & \cdots & \tilde{\mathbf{x}}_{q,1} \\ \tilde{\mathbf{x}}_{q-n+1,2} & \tilde{\mathbf{x}}_{q-n+2,2} & \cdots & \tilde{\mathbf{x}}_{q,2} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{x}}_{q-n+1,n_{var}} & \tilde{\mathbf{x}}_{q-n+2,n_{var}} & \cdots & \tilde{\mathbf{x}}_{q,n_{var}} \end{bmatrix} \tag{12}$$

where n is the fixed window length for data augmentation. As a result of data preprocessing, data samples are transformed into data sequences of length n. In the second step of JITL modeling, the similarities between the normalized data sequence $\tilde{\mathbf{X}}_q$ and the sequences of the database are calculated using DTW. Then a local machine learning soft sensor model is trained with the historical data sequences which are most similar to $\tilde{\mathbf{X}}_q$. Two data-driven modeling methods, SVM and GPR, are used to construct a local model in this study. Finally, the prediction $\hat{y}_q$ for the quality of the query sample x_q is made with the trained local model. The local model is discarded and the above modeling steps are repeated when a new query sample is acquired.

SENSITIVITY ANALYSIS OF MODEL HYPERPARAMETERS

This section evaluates the effects of model hyperparameters on the performance of the proposed DTW-based JITL model by simulating two multi-grade systems. The prediction accuracy and computational requirement for the proposed DTW-based JITL modeling approach are highly dependent on three hyperparameters: the input length, the width of the warping window, and the number of similar samples chosen from the database. Two multi-grade simulation case studies are considered in order to undertake sensitivity analysis on these hyperparameters.

The first simulation case study is a simple nonlinear dynamic system where the output is determined by the past ten samples of the inputs. Additionally, the inputs change periodically depending on the grade to simulate multi-grade characteristics. The system is defined as follows:

$$y_1=A_1\cdot X_1+B_1\cdot X_2+C_1\cdot X_1\cdot X_2 \tag{13}$$

$$y_2=A_2\cdot X_1+B_2\cdot X_2+C_2\cdot X_1\cdot X_2 \tag{14}$$

$$y=\log(y_1\cdot y_2) \tag{15}$$

where $X_1=[x_{1,t-9}, x_{1,t-8}, \ldots, x_{1,t}]$ and $x_{i,k}$ is the i-th input variable at time k. A, B, and C are the coefficient vectors of length 10. Gaussian random noise is added to the inputs and output. The values of coefficients, including A, B, and C, are present in the Appendix section. A total of 520 data samples were obtained as a simulation results, which is shown in Fig. 4. The first 50 percent of the samples were used as the training dataset, while the remaining 50 percent of the samples were used as the testing dataset.

In the second case study, a sequence of three continuous stirred tank reactors (CSTRs) was used to simulate a multi-grade chemical system with time delay and multi-grade products. The simple schematic of the CSTR system is illustrated in Fig. 5 and the nomenclature and nominal operating conditions are summarized in Table 2.

The process model of each reactor consists of two nonlinear ordinary differential equations [57]. Two chemical species, A and B, exist in the reactors and undergo irreversible and exothermic chemical reaction, A→B.

$$\dot{C}_{Ai}=\frac{q_i}{V_i}(C_{A(i-1)}-C_{Ai})-k_0C_{Ai}\exp\left(-\frac{E}{RT}\right) \tag{16}$$

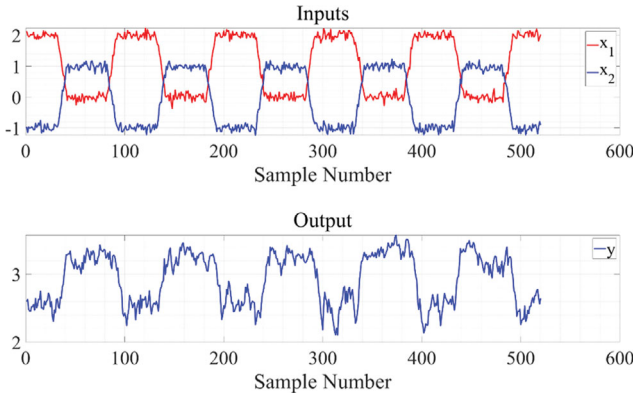

Fig. 4. Simulation data obtained from the nonlinear numerical example.

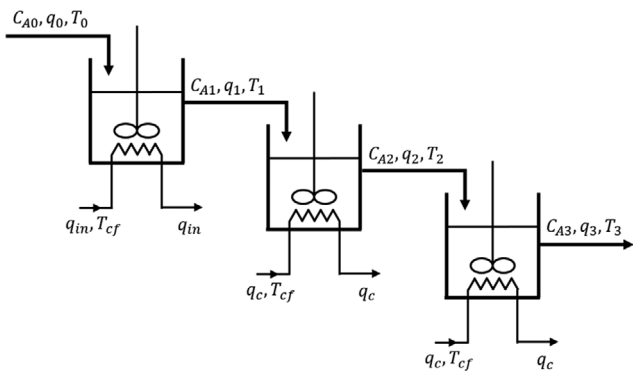

Fig. 5. Schematic of the chemical system consisting of three CSTRs in series.

Table 2. Nomenclature and nominal operating conditions of the CSTR system

Symbol	Description	Operating condition	Unit
C_{A0}	Concentration of A in the feed stream into the first reactor	1.00	mol l^{-1}
C_{Ai}	Concentration of A in the *i*-th reactor (i=1, 2, 3)	0.0836	mol l^{-1}
C_p	Heat capacity of the reaction mixture	1.00	cal g^{-1} K^{-1}
C_{pc}	Heat capacity of the coolant	1.00	cal g^{-1} K^{-1}
E/R	Fraction of the activation energy divided by the gas constant	9.95×10^{3}	K
hA	Product of the heat transfer coefficient and heat transfer area	7.00×10^{5}	cal min^{-1} K^{-1}
k_0	Pre-exponential factor	7.20×10^{5}	min^{-1}
q_c	Flowrate of coolant for the second and third reactor	103.41	l min^{-1}
q_i	Flowrate of feed stream into the *i*-th reactor (i=1, 2, 3)	100	l min^{-1}
q_{in}	Flowrate of coolant for the first reactor	103.41	l min^{-1}
T_0	Temperature of the feed stream into the first reactor	350	K
T_c	Temperature of coolant	350	K
T_i	Temperature of the *i*-th reactor (i=1, 2, 3)	440.2	K
V_i	Volume of the *i*-th reactor (i=1, 2, 3)	100	l
ΔH	Heat of reaction	-2.00×10^{5}	cal mol^{-1}
ρ	Density of the reaction mixture	1,000	g l^{-1}
ρ_c	Density of coolant	1,000	g l^{-1}

$$\dot{T}_i=\frac{q_i}{V_i}(T_{i-1}-T_i)+\frac{(-\Delta H)k_0C_{Ai}}{\rho C_p}\exp\left(-\frac{E}{RT}\right)+\frac{\rho_c C_{pc}}{\rho C_p V}q_c\left[1-\exp\left(-\frac{hA}{q_c\rho_c C_{pc}}\right)\right](T_{cf}-T_i) \quad (17)$$

Since the flowrates of all streams are set equal, the reactor volumes are equal and remain unchanged during simulation. Additionally, transport delays of three minutes are added between the reactors. The manipulated variable of the system is the flowrate of the coolant for the first reactor, q_{in}. It is assumed that only the manipulated variable, q_{in}, and the temperature of the first reactor, T_1, are measured. The output of the process is the effluent concentration of A from the third reactor, C_{A3}. Fig. 6 illustrates the simulation result of the CSTR system. A total of four grades and their grade changeovers were simulated by manipulating the coolant flowrate or the first reactor. Additionally, the Gaussian random noise was added to the input and output variables. The simulation time was 500 minutes and a total of 1001 data samples were obtained as a result. The first 501 samples were used as the original training dataset and the remaining 500 samples were used as the testing dataset.

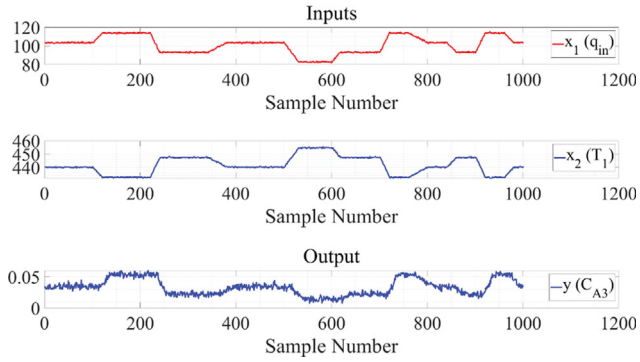

Fig. 6. Simulation data obtained from the CSTR system.

SVM and GPR models were trained for the sensitivity analysis of the hyperparameters for both case studies. The prediction accuracy of the soft sensors was evaluated using the root mean squared error (RMSE) defined as in (18):

$$\mathrm{RMSE}=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2} \quad (18)$$

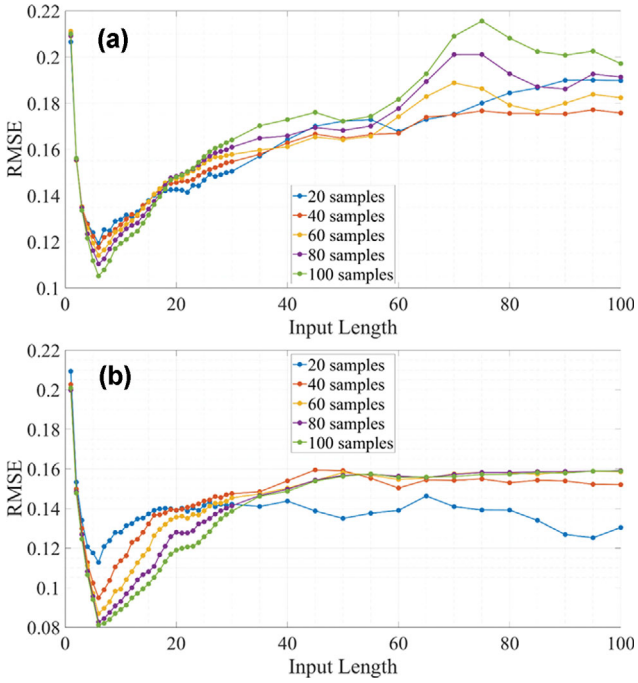

Fig. 7. Prediction accuracy of models with different input lengths and number of similar samples for the numerical example (a) SVM models, (b) GPR models.

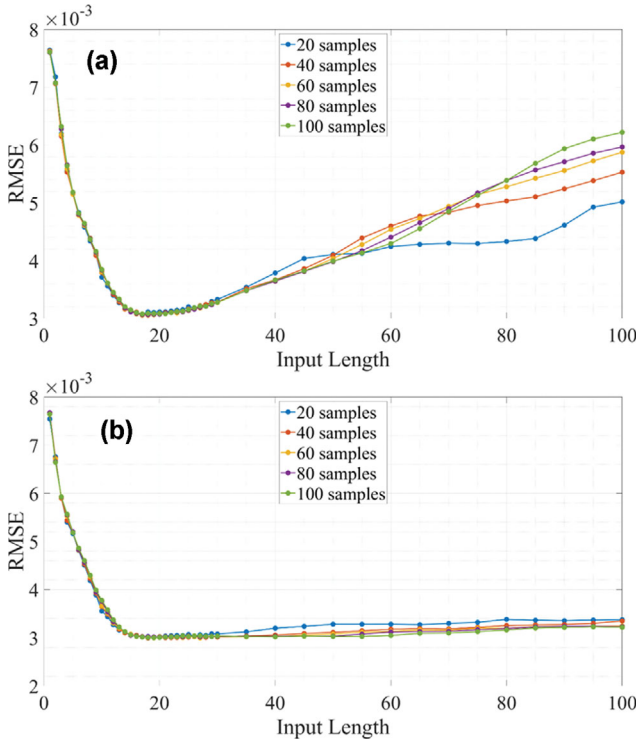

Fig. 8. Prediction accuracy of models with different input lengths and number of similar samples for the CSTR example (a) SVM models, (b) GPR models.

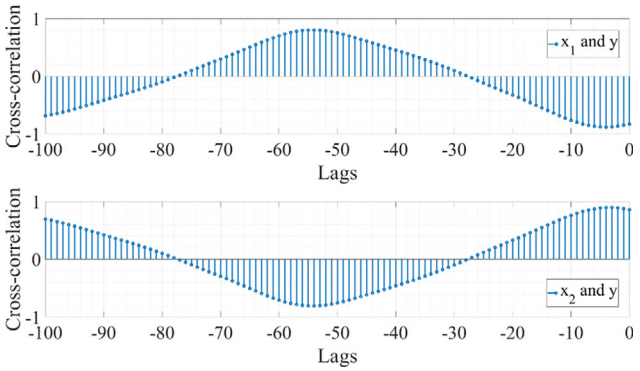

Fig. 9. Cross-correlation between input and output variables of the numerical example.

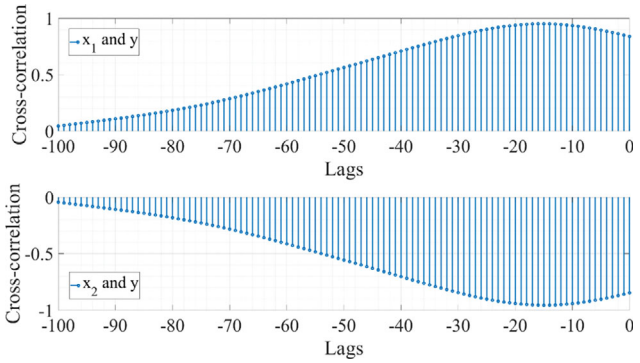

Fig. 10. Cross-correlation between input and output variables of the CSTR example.

where N is the number of samples, y_i and $\hat{y}_i$ are the measured and predicted outputs, respectively. Additionally, the CPU time was measured on an Intel Core i7-8700 CPU @3.20 GHz to evaluate the computational requirement.

1. Input Length and Number of Similar Samples

The length of the input sequence and the number of similar samples used in local modeling affect not only the required calculation time but also the prediction accuracy of the proposed model. Therefore, the effects of the hyperparameters on the prediction performance of the DTW-based JITL models are analyzed for both case studies in this subsection. Figs. 7 and 8 depict the prediction accuracy of the DTW-based JITL models with varying input lengths and the number of similar samples on the testing datasets of the numerical and CSTR examples, respectively. Both the SVM and GPR models for the numerical example reach the lowest RMSE values at the input length of 6, regardless of the number of samples used for local modeling. On the other hand, the DTW-based JITL models for the CSTR system achieve the best prediction accuracy at the input length close to 18.

The cross-correlations between the input and output variables of the numerical example and CSTR system are shown in Fig. 9 and Fig. 10, respectively. The absolute values of cross-correlations reach their maxima at time lags of 4 and 54 for the numerical example. Furthermore, the cross-correlations exhibit periodical changes because the grade and input variables change periodically with a period of 50 time steps. On the other hand, the output variable of the CSTR system is most correlated with the first and second input variables at time lags of 16 and 15, respectively.

The results from varying input lengths and cross-correlations suggest that the best modeling performance is achieved when a model takes time series of length slightly longer than the time lags of maximum cross-correlation between input and output. Additionally, the modeling performance of the JITL models decreases as the input length becomes longer or shorter than the optimal time lags as shown in Fig. 7 and Fig. 8, which represents the importance of finding the optimal length for input data sequences. If the input data sequence is longer than the cross-correlation between the input and output, the sequences are more likely to lose their original characteristics and dynamics through warping. On the other hand, if the input time series is too short, the temporal correlations of the variables are not sufficiently considered in modeling, resulting in poor prediction performance. Therefore, it is necessary to find the optimal length for input augmentation by calculating the cross-correlation between the input and output variables.

In addition, Figs. 7 and 8 illustrate the effect of the number of similar samples selected from the database on the prediction accuracy of the proposed model. The prediction accuracy of the proposed model increases as more samples are used in the numerical example for the optimal input length, 6. Since the nonlinear system changes periodically between two distinct operation modes, many samples exist for each mode. Therefore, the local models trained with more samples predict the output more accurately. Additionally, the numerical example's system is much simpler than the

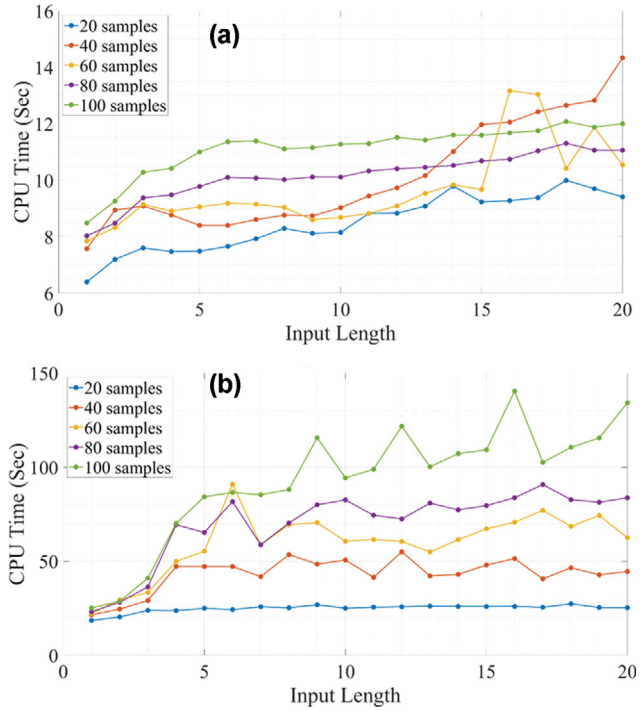

Fig. 11. CPU time spent calculating the proposed modeling approach based on GPR (a) numerical example (b) CSTR example.

reactor systems, making it easier for regression models to learn when provided more data samples.

On the other hand, the DTW-based JITL models for the CSTR example achieved the best prediction performance when 40 similar samples with an input length of 18 were utilized. When too many samples are used in model training, it is more likely that the training dataset will contain samples that are not similar or relevant to query data, particularly for grade changeovers whose duration is short compared to steady-state operations, thereby decreasing the performance of the model. In addition, Fig. 11 illustrates the CPU required to compute the whole testing procedure for the proposed GPR-based model. As the number of training samples increases, the CPU time required by the proposed model increases. To achieve the optimal prediction accuracy in an acceptable amount of time, it is crucial to examine the tradeoff between prediction accuracy and computation time by selecting an adequate number of similar samples for each target system.

2. Warping Path Constraint

In this subsection, the effect of the temporal constraint on the warping window size of DTW on the modeling performance in the DTW-based JITL modeling approach is analyzed. Without the warping path constraint, DTW stretches and compresses data sequences as much as possible to make the warped sequences as similar as possible. However, unconstrained warping is susceptible to outliers in the time series. Additionally, unconstrained warping may distort data sequences excessively so that the temporal correlations and dynamics of the raw sequences rarely remain in the warped sequences. Therefore, the accuracy of distance measure and computational efficiency can be improved by constraining the warping window width, δ, in a fixed range [48].

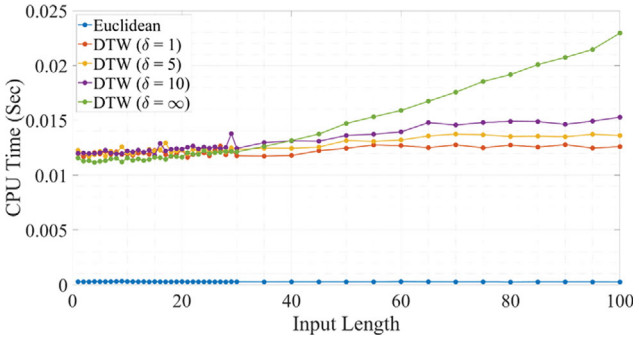

Fig. 12. CPU time spent calculating similarities between samples of the numerical example.

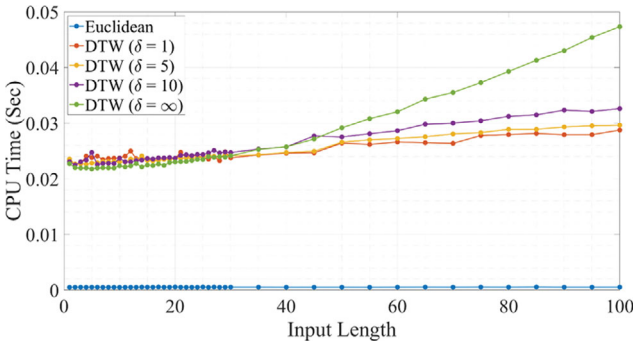

Fig. 13. CPU time spent calculating similarities between samples of the CSTR example.

Figs. 12 and 13 show the average CPU time required to calculate the similarities between the query and stored data sequences using the Euclidean distance, unconstrained DTW, and constrained DTW for the numerical and CSTR examples, respectively. For both case studies, the computation times for the Euclidean distance were less than 0.001 seconds, which is significantly less than those of DTW. On the other hand, unconstrained DTW calculations for data sequences of length 10 required 0.011571 and 0.022339 seconds in the numerical and CSTR examples, respectively. As illustrated in Figs. 12 and 13, the constrained DTW with a narrower warping window width takes less computation time than the unconstrained DTW because only a small number of warping pathways must be determined. Moreover, longer data sequences necessitate additional calculation time, which is especially evident for the unconstrained DTW. Nevertheless, unconstrained DTW computations with an input length of 100 required 0.022965 and 0.047334 seconds for the numerical and CSTR examples, respectively, demonstrating that the proposed modeling approach can be utilized for online applications with short sampling periods.

Fig. 14 and 15 illustrate, for numerical and CSTR examples, respectively, a comparison of the prediction accuracy of the proposed models with different similarity measures and constraints as measured by root mean square error (RMSE). The models were trained using 40 similar samples and the input data sequences with a length of 6 and 18 for the numerical and CSTR examples, respectively. For both case studies, the prediction accuracy of the models improved as the width of the warping window increased. On the other

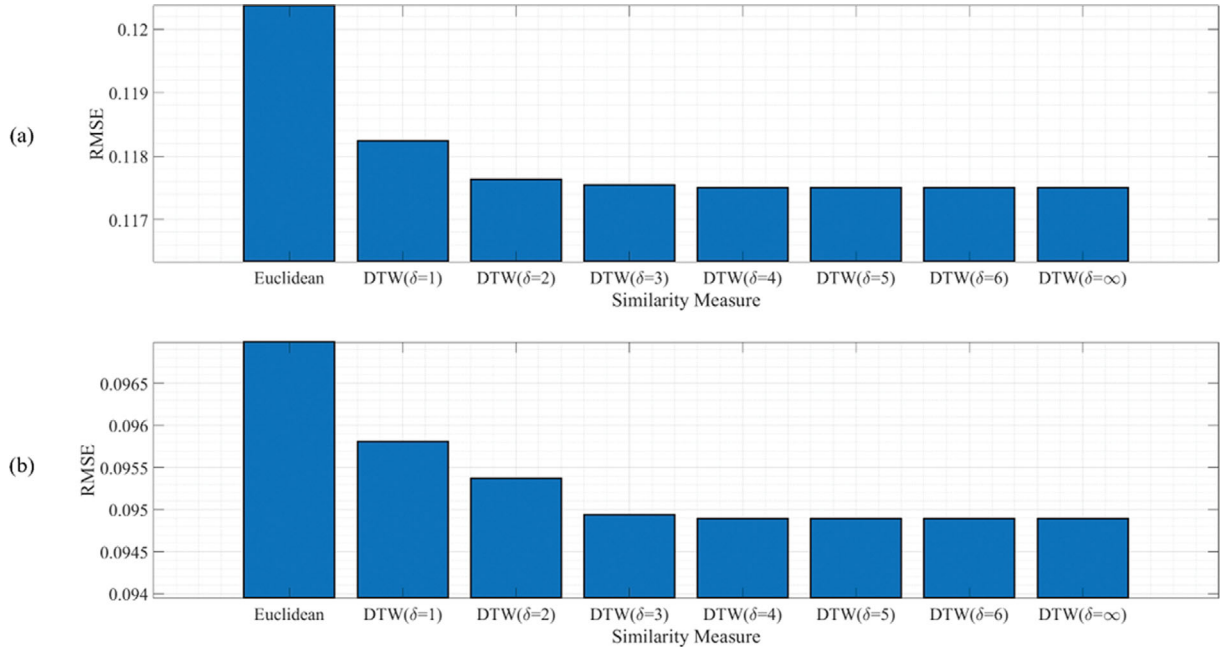

Fig. 14. Prediction accuracy of models using the Euclidean distance and DTW with different warping window width constraint for the numerical example (a) SVM models, (b) GPR models.

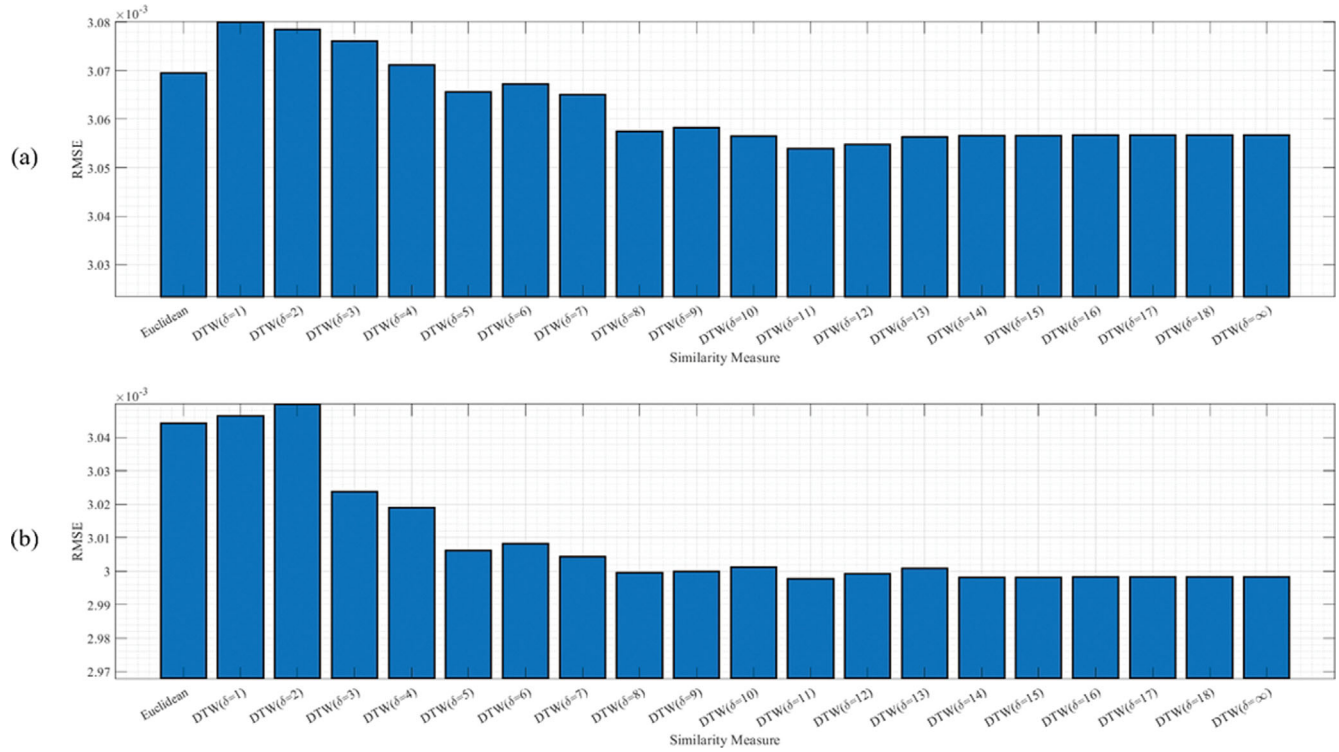

Fig. 15. Prediction accuracy of models using the Euclidean distance and DTW with different warping window width constraint for the CSTR example (a) SVM models, (b) GPR models.

hand, a constraint on the warping window width that is overly stringent may reduce prediction performance because only warping paths close to the raw time series are evaluated. As shown in Fig. 14 and 15, RMSE decreases and reaches plateaus when the warping window constraint is approximately half the input length. Therefore, by limiting the warping window width to half the length of the input sequence, it is possible to attain optimal prediction accuracy with a reduced computing demand.

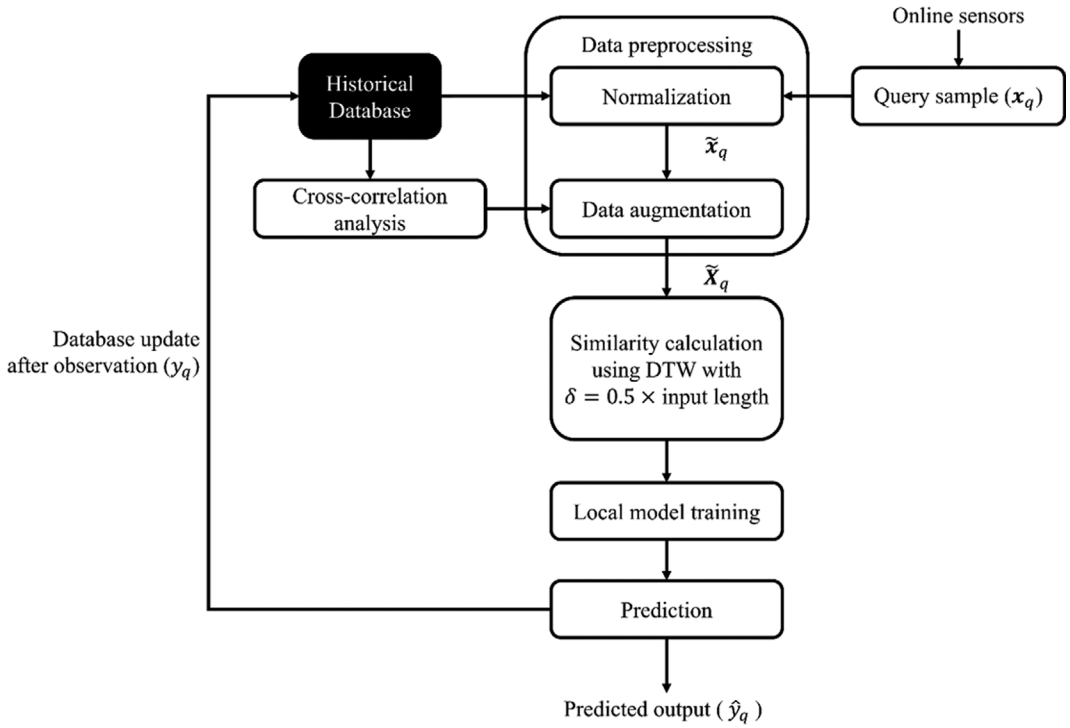

Fig. 16. Algorithm of the proposed DTW-based JITL model with hyperparameter optimization.

OPTIMIZED DTW-BASED JITL MODELING APPROACH

1. Proposed Modeling Approach

In this section, we propose a DTW-based JITL soft sensor modeling approach with the optimized model hyperparameters. The results from Section 4 reveal that the prediction accuracy and computational requirement of the DTW-based JITL model depend on three hyperparameters: input length, number of similar samples, and width constraint of the warping window. Fig. 16 illustrates the algorithm of the proposed modeling approach. The proposed modeling algorithm differs from the previous DTW-based JITL modeling algorithm introduced in Section 3 in that the input length and DTW constraint are optimized by analyzing cross-correlation between the input and output variables. First, the cross-correlation coefficients between the input and output variables are calculated using the stored data samples in the historical database. Then, the data samples are augmented to data sequences of length slightly longer than the time lags of maximum cross-correlation. In this study, the length of the augmented input is determined by adding two to the time lag of maximum cross-correlation, which is the optimal value from the results of Subsection 4.1. Then, the similarities between the augmented data sequences are calculated using DTW under the constraint of the warping window width. The maximum warping width, δ, is determined to be half the length of the input sequence. The subsequent steps are identical to the original DTW-based JITL modeling procedure.

2. Results and Discussion

We simulated multi-grade case studies to evaluate the prediction performance of the proposed DTW-based JITL model to verify the effectiveness of the proposed modeling approach. Additionally, conventional JITL soft sensors based on the Euclidean distance were developed for performance comparison. The prediction accuracy of the soft sensors was evaluated using four statistical indices: root mean squared error (RMSE), mean absolute percentage error (MAPE), Theil's inequality coefficient (TIC), and coefficient of determination (R^2).

$$\mathrm{MAPE} = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{19}$$

$$\mathrm{TIC} = \frac{\sqrt{\Sigma_{i=1}^{N}(y_i - \hat{y}_i)^2}}{\sqrt{\Sigma_{i=1}^{N} y_i^2}\sqrt{\Sigma_{i=1}^{N}\hat{y}_i^2}} \tag{20}$$

$$R^2 = 1 - \frac{\Sigma_{i=1}^{N}(y_i - \hat{y}_i)^2}{\Sigma_{i=1}^{N}(y_i - \bar{y}_i)^2} \tag{21}$$

where N is the number of samples, $\bar{y}$ is the mean value of y, y_i and $\hat{y}_i$ are the measured and predicted output variable, respectively. For both case studies, two machine learning regression models, SVM and GPR, were utilized as local predictive models and the number of similar samples drawn from the database for local model training was 40. The augmented input lengths were 6 and 18 for the numerical and CSTR examples, respectively. Thus, the respective maximum DTW window width are 3 and 9 for the numerical and CSTR examples.

Tables 3 and 4 summarize modeling results of the proposed optimized DTW-based JITL models and Euclidean distance-based JITL models for the numerical and CSTR examples, respectively. Since the simulated output was determined by both current and past

Table 3. Prediction results of all trained models on the testing dataset of the numerical example

Data-driven model	Input length	Similarity measure	RMSE	MAPE (%)	TIC	R^2
SVM	1	Euclidean distance	0.2067	5.425	0.03497	0.7312
SVM	6	Euclidean distance	0.1204	3.421	0.02024	0.9089
SVM	6	DTW (δ=3)	0.1175	3.320	0.01977	0.9131
SVM	6	DTW (δ=∞)	0.1175	3.318	0.01976	0.9131
GPR	1	Euclidean distance	0.2056	5.617	0.03473	0.7341
GPR	6	Euclidean distance	0.09699	2.737	0.01631	0.9408
GPR	6	DTW (δ=3)	0.09494	2.676	0.01598	0.9433
GPR	6	DTW (δ=∞)	0.09489	2.675	0.01597	0.9433

Table 4. Prediction results of all trained models on the testing dataset of the CSTR example

Data-driven model	Input length	Similarity measure	RMSE ($\times 10^3$)	MAPE (%)	TIC	R^2
SVM	1	Euclidean distance	7.634	21.93	0.1127	0.6934
SVM	18	Euclidean distance	3.069	10.81	0.04530	0.9504
SVM	18	DTW (δ=9)	3.058	10.71	0.04512	0.9508
SVM	18	DTW (δ=∞)	3.057	10.71	0.04510	0.9509
GPR	1	Euclidean distance	7.529	22.13	0.1118	0.7018
GPR	18	Euclidean distance	3.044	10.65	0.04490	0.9513
GPR	18	DTW (δ=9)	3.000	10.41	0.04424	0.9527
GPR	18	DTW (δ=∞)	2.998	10.41	0.04421	0.9527

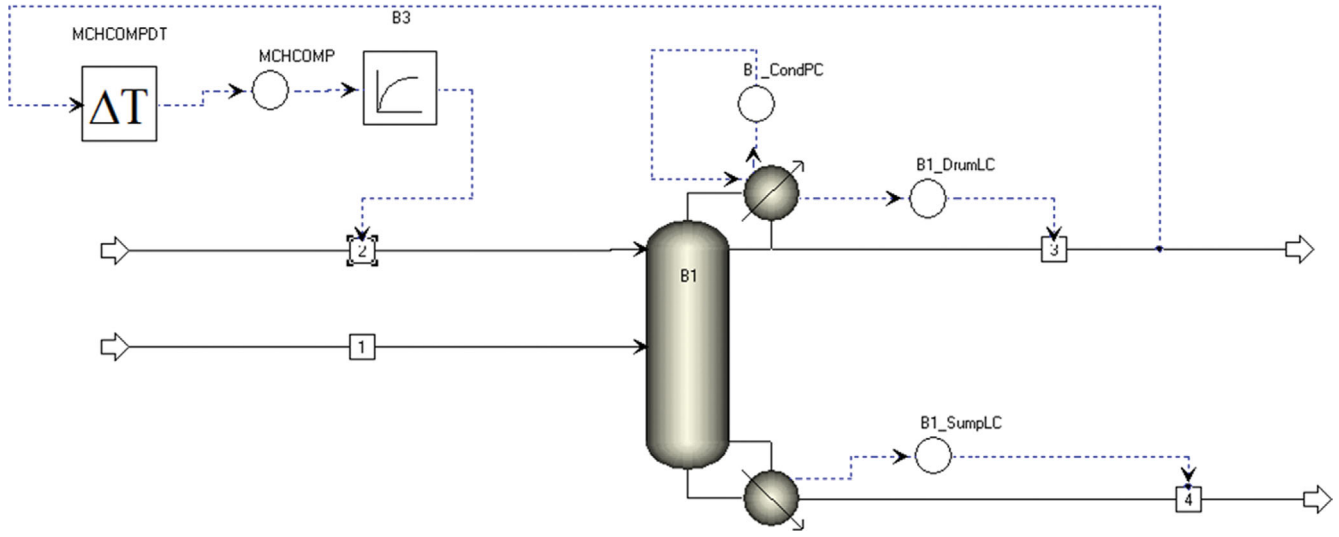

Fig. 17. Schematic of the MCH distillation process.

inputs, the models taking time series as input outperform the models taking data points as input. Additionally, the DTW-based JITL models demonstrate better prediction accuracy than the Euclidean distance-based JITL models. For instance, the RMSE, MAPE, TIC, and R^2 of the constrained DTW-based GPR model for the CSTR system are 3.000, 10.41, 0.04424, and 0.9527, respectively, which represents better prediction accuracy than the Euclidean distance-based GPR model whose RMSE, MAPE, TIC, and R^2 are 3.044, 10.65, 0.04490, and 0.9513, respectively. Additionally, the prediction performance of a constrained DTW-based JITL model is comparable to that of an unconstrained DTW-based JITL model.

To further demonstrate the efficiency of the proposed model, a distillation process where methylcyclohexane (MCH) is recovered from a mixture was simulated. Fig. 17 illustrates the schematic of the process in which the purity of MCH, which is the output variable, is controlled by manipulating the flowrate of the phenol inlet

Table 5. Prediction results of all trained models on the testing dataset of the distillation example

Data-driven model	Input length	Similarity measure	RMSE ($\times 10^3$)	MAPE (%)	TIC ($\times 10^3$)	R^2
SVM	1	Euclidean distance	2.521	0.1321	1.2792	0.6179
SVM	30	Euclidean distance	0.5751	0.02568	0.2919	0.9801
SVM	30	DTW (δ=15)	0.4556	0.02020	0.2312	0.9875
SVM	30	DTW (δ=∞)	0.4635	0.02043	0.2353	0.9871
GPR	1	Euclidean distance	2.348	0.1348	1.192	0.6686
GPR	30	Euclidean distance	0.3407	0.01478	0.1729	0.9930
GPR	30	DTW (δ=15)	0.3124	0.01348	0.1585	0.9941
GPR	30	DTW (δ=∞)	0.3072	0.01339	0.1559	0.9943

stream, which is the input variable. The sampling time and total simulation time was 0.01 h and 30 h, respectively. Thus, a total of 3,000 samples were obtained, of which 50% were used as the testing dataset. Table 5 summarizes prediction performance of the proposed models and conventional Euclidean distance-based JITL models on the testing dataset of the distillation example. The window length calculated from the cross-correlation analysis and warping window width for the proposed model were 30 and 15, respectively. For both data-driven regression models, the proposed constrained DTW-based JITL models demonstrate better prediction accuracy compared to the Euclidean distance-based models. In summary, the modeling results indicate that improved prediction accuracy with reduced computation can be achieved by the proposed DTW-based JITL modeling approach with hyperparameter optimization.

CONCLUSIONS

We developed a DTW-based JITL soft sensor modeling approach for multi-grade processes and compared its prediction performance with the classic JITL modeling method based on the Euclidean distance. Prediction accuracy was improved by considering the temporal correlations and process dynamics in modeling. First, the proposed modeling approach takes time series rather than data points as its input in order to account for dynamic changes in input variables. Raw process variables are augmented with time-lagged data to form fixed-length data sequences. Second, the similarity between the query and the historical data sequences is measured using DTW. DTW finds the optimal warping path by stretching and compressing the sequences, resulting in more accurate distance measure between data sequences compared to the Euclidean distance.

Additionally, the effects of the model hyperparameters, including the length of input data sequences, number of similar samples, and warping path constraint, on the prediction accuracy were analyzed using SVM and GPR models. First, the optimal window size of a data sequence depends on the cross-correlation between the input and output variables. Second, constraining DTW path within a fixed width reduces computational requirements while maintaining prediction performance. Finally, the DTW-based JITL modeling algorithm with hyperparameter optimization was proposed and its modeling performance was evaluated using three multi-grade simulation case studies. As a result, the proposed DTW-based JITL models outperformed classic JITL models based on the Euclidean distance.

NOMENCLATURE

C_{A0} : concentration of A in the feed stream into the first reactor [mol l^{-1}]
C_{Ai} : concentration of A in the *i*-th reactor (i=1, 2, 3) [mol l^{-1}]
C_p : heat capacity of the reaction mixture [cal g^{-1} K^{-1}]
C_{pc} : heat capacity of the coolant [cal g^{-1} K^{-1}]
DTW : distance calculated by dynamic time warping
ED : euclidean distance
E/R : fraction of the activation energy divided by the gas constant [K]
hA : product of the heat transfer coefficient and heat transfer area [cal min^{-1} K^{-1}]
k_0 : pre-exponential factor [min^{-1}]
KLD : Kullback-Leibler divergence
MD : mahalanobis distance
q_c : flowrate of coolant for the second and third reactor [l min^{-1}]
q_i : flowrate of feed stream into the *i*-th reactor (i=1, 2, 3) [l min^{-1}]
q_{in} : flowrate of coolant for the first reactor [l min^{-1}]
T_0 : temperature of the feed stream into the first reactor [K]
T_c : temperature of coolant [K]
T_i : temperature of the *i*-th reactor (i=1, 2, 3) [K]
V_i : volume of the *i*-th reactor (i=1, 2, 3) [l]
ΔH : heat of reaction [cal mol^{-1}]
ρ : density of the reaction mixture [g l^{-1}]
ρ_c : density of coolant [g l^{-1}]

ABBREVIATIONS

ANN : artificial neural network
CSTR : continuous stirred tank reactor
DTW : dynamic time warping
GPR : Gaussian process regression
JITL : just-in-time learning
LSTM : long short-term memory network
MAPE : mean absolute percentage error
PCA : principal component analysis
PLS : partial least squares

RMSE : root mean squared error
SVM : support vector machine
TIC : Theil's inequality coefficient

REFERENCES

1. M. Ohshima and M. Tanigaki, *J. Process Control*, **10**, 135 (2000).
2. Y. Liu and J. Chen, *J. Process Control*, **23**, 793 (2013).
3. Y. Liu, Z. Gao and J. Chen, *Chem. Eng. Sci.*, **102**, 602 (2013).
4. M. Kim, Y. H. Han, I. S. Han and C. Han, *Ind. Eng. Chem. Res.*, **44**, 334 (2005).
5. R. Sharmin, U. Sundararaj, S. Shah, L. V. Griend and Y. J. Sun, *Chem. Eng. Sci.*, **61**, 6372 (2006).
6. J. Liu, *Control Eng. Pract.*, **15**, 769 (2007).
7. P. Kadlec, R. Grbic and B. Gabrys, *Comput. Chem. Eng.*, **35**, 1 (2011).
8. S. J. Qin, H. Yue and R. Dunia, *Ind. Eng. Chem. Res.*, **36**, 1675 (1997).
9. Z. Ge, *Control Eng. Pract.*, **31**, 9 (2014).
10. M. K. Hartnett, G. Lightbody and G. W. Irwin, *Chemom. Intell. Lab. Syst.*, **40**, 215 (1998).
11. J. Yu, *Ind. Eng. Chem. Res.*, **51**, 13227 (2012).
12. W. Shao and X. Tian, *Chem. Eng. Res. Des.*, **95**, 113 (2015).
13. Y. Matsuyama, S. Kim and S. Hasebe, *Comput. Chem. Eng.*, **146**, 107224 (2021).
14. S. Park and C. Han, *Comput. Chem. Eng.*, **24**, 871 (2000).
15. I. S. Han, C. Han and C. B. Chung, *J. Appl. Polym. Sci.*, **95**, 967 (2005).
16. T. C. Park, T. Y. Kim and Y. K. Yeo, *Korean J. Chem. Eng.*, **27**, 1662 (2010).
17. H. Jin, X. Chen, J. Yang, H. Zhang, L. Wang and L. Wu, *Chem. Eng. Sci.*, **131**, 282 (2015).
18. J. Yu, *Chem. Eng. Sci.*, **82**, 22 (2012).
19. R. Grbić, D. Slišković and P. Kadlec, *Comput. Chem. Eng.*, **58**, 84 (2013).
20. Y. Liu, T. Chen and J. Chen, *Ind. Eng. Chem. Res.*, **54**, 5037 (2015).
21. J. C. B. Gonzaga, L. A. C. Meleiro, C. Kiang and R. Maciel Filho, *Comput. Chem. Eng.*, **33**, 43 (2009).
22. A. J. De Assis and R. Maciel Filho, *Comput. Chem. Eng.*, **24**, 1099 (2000).
23. X. Yuan and Y. Wang, *IEEE Trans. Industr. Inform.*, **16**, 3168 (2019).
24. X. Yuan, L. Li, Y. A. W. Shardt, Y. Wang and C. Yang, *IEEE Trans. Ind. Electron.*, **68**, 4404 (2021).
25. W. Li, H. H. Yue, S. Valle-Cervantes and S. J. Qin, *J. Process Control*, **10**, 471 (2000).
26. H. D. Jin, Y. H. Lee, G. Lee and C. Han, *Ind. Eng. Chem. Res.*, **45**, 696 (2006).
27. X. Wang, U. Kruger and G. W. Irwin, *Ind. Eng. Chem. Res.*, **44**, 5691 (2005).
28. S. J. Qin, *Comput. Chem. Eng.*, **22**, 503 (1998).
29. B. S. Dayal and J. F. MacGregor, *J. Process Control*, **7**, 169 (1997).
30. F. Ahmed, S. Nazir and Y. K. Yeo, *Korean J. Chem. Eng.*, **26**, 14 (2009).
31. L. Xie, J. Zeng and C. Gao, *IEEE Trans. Control Syst. Technol.*, **22**, 360 (2014).
32. Y. Liu, Z. Gao, P. Li and H. Wang, *Ind. Eng. Chem. Res.*, **51**, 4313 (2012).
33. K. Yang, H. Jin, X. Chen, J. Dai, L. Wang and D. Zhang, *Chemom. Intell. Lab. Syst.*, **155**, 170 (2016).
34. X. Yuan, J. Zhou, Y. Wang and C. Yang, *J. Chemom.*, **32**, e3040 (2018).
35. X. Yuan, Z. Ge, B. Huang, Z. Song and Y. Wang, *IEEE Trans. Industr. Inform.*, **13**, 532 (2017).
36. F. Guo, W. Bai and B. Huang, *J. Process Control*, **92**, 90 (2020).
37. H. Kaneko and K. Funatsu, *AIChE J.*, **62**, 717 (2016).
38. J. Liu, T. Liu and J. Chen, *Chem. Eng. Sci.*, **191**, 31 (2018).
39. J. Liu, J. Hou and J. Chem, *Comput. Chem. Eng.*, **154**, 107469 (2021).
40. Y. Liu and Z. Gao, *J. Appl. Polym. Sci.*, **132**, 41958 (2015).
41. Y. Liu, Y. Liang and Z. Gao, *J. Appl. Polym. Sci.*, **134**, 45094 (2017).
42. J. Wang, K. Qiu, R. Wang, X. Zhou and Y. Guo, *IEEE Trans. Instrum. Meas.*, **70**, 1 (2021).
43. J. Zheng, F. Shen and L. Ye, *IEEE Access*, **9**, 72172 (2021).
44. Y. Wu, D. Liu, X. Yuan and Y. Wang, *IEEE Sens.*, **21**, 3497 (2021).
45. F. Guo and B. Huang, *Chemom. Intell. Lab. Syst.*, **204**, 104118 (2020).
46. F. Guo, B. Wei and B. Huang, *Comput. Chem. Eng.*, **146**, 107230 (2021).
47. H. Sakoe and S. Chiba, *IEEE Trans. Acoust. Speech Signal Process.*, **26**, 43 (1978).
48. H. Ding, G. Trajcevski, P. Scheuermann, X. Wang and E. J. Keogh, *Proc. VLDB Endow.*, **1**, 1542 (2008).
49. A. Kholmatov and B. Yanikoglu, *Pattern Recognit. Lett.*, **26**, 2400 (2005).
50. N. Gillian, R. B. Knapp and S. O'Modhrain, in *Proc. of the 11th International Conference on New Interfaces for Musical Expression*, 337 (2011).
51. X. Meng, H. Fu, L. Peng, G. Liu, Y. Yu, Z. Wang and E. Chen, *IEEE Trans. Intell. Trasp. Syst.*, **23**, 2021 (2022).
52. K. Q. Zhou, Y. Qin, B. P. L. Lau, C. Yuen and S. Adams, in *IECON 2021-47th Annual Conference of the IEEE Industrial Electronics Society* (2021).
53. Y. Si, Z. Chen, J. Sun, D. Zhang and P. Qian, *IEEE Access*, **8**, 108359 (2020).
54. A. Kassidas, J. F. MacGregor and P. A. Taylor, *AIChE J.*, **44**, 864 (1998).
55. W. Ku, R. H. Storer and C. Georgakis, *Chemom. Intell. Lab. Syst.*, **30**, 179 (1995).
56. S. Heo and J. H. Lee, *IFAC-PapersOnLine*, **51**, 470 (2018).
57. E. P. Nahas, M. A. Henson and D. E. Seborg, *Comput. Chem. Eng.*, **16**, 1039 (1992).

APPENDIX

Table A1. Values of coefficient vectors of numerical example

Symbol	Value
A_1	[0.0403, 0.0543, 0.1291, 0.1995, 0.3456, 0.4454, 0.6008, 0.8935, 0.9363, 0.9462]
A_2	[0.1578, 0.1715, 0.3566, 0.4147, 0.4849, 0.5166, 0.6569, 0.6865, 0.9870, 0.9916]
B_1	[0.0579, 0.1318, 0.1556, 0.2551, 0.6964, 0.7609, 0.8176, 0.8575, 0.8957, 0.9820]
B_2	[0.0184, 0.1433, 0.2459, 0.3627, 0.3914, 0.4347, 0.7435, 0.7720, 0.8422, 0.8483]
C_1	[0.0112, 0.0244, 0.0272, 0.0450, 0.0581, 0.0684, 0.0711, 0.0718, 0.0736, 0.0933]
C_2	[0.0032, 0.0139, 0.0192, 0.0305, 0.0475, 0.0568, 0.0650, 0.0842, 0.0914, 0.0940]