A comprehensive hybrid first principles/machine learning modeling framework for complex industrial processes

Bei Sun[a,b], Chunhua Yang[a,*], Yalin Wang[a], Weihua Gui[a], Ian Craig[c], Laurentz Olivier[c]

[a]*School of Automation, Central South University, Changsha, China 410083*
[b]*the Peng Cheng Laboratory, Shenzhen, China 518000*
[c]*Department of Electrical, Electronic, and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa*

Abstract

The selection of an appropriate descriptive system and modeling framework to capture system dynamics and support process control applications is a fundamental problem in the operation of industrial processes. In this study, to account for the the highly complex dynamics of industrial process and additional requirements imposed by smart and optimal manufacturing systems, an extended state space descriptive system, named comprehensive state space, is first designed. Then, based on the descriptive system, a hybrid first principles/machine learning modeling framework is proposed. The hybrid model is formulated as a combination of a nominal term and a deviation term. The nominal term covers the underlying physicochemical principles. The deviation term handles the effects of high-dimensional influence factors using regression of low-dimensional deep process features. To handle the multimodal and time-varying properties of process dynamics, the comprehensive state space is divided into subspaces indicating different operating conditions. The model parameters are identified and trained for each operating condition to form the sub-models. Then the system dynamics are formulated as a weighted sum of sub-models, with the weights being the probabilities that the current operating point belongs to different operating conditions. The weights update with the movement of the operating point in the comprehensive state space. Moreover, the descriptive system provides a platform for visualization, and can act as a digital twin of the physical process. A case study illustrates the feasibility and performance of the proposed descriptive system.

Keywords: comprehensive state space; descriptive system; modeling; machine learning

1. Introduction

Process industries, which mainly include iron and steel, nonferrous metals, petrochemical and architectural materials, are the cornerstones of the economies of many countries (Craig et al. (2011)). By using continuous and complex physicochemical processes, they transform elementary raw materials into products that support economic development and major engineering projects. The utilization rate of raw materials, energy consumption, production costs and environmental effects are the main concerns in the daily operation of process industries. With the aim of high-efficiency and green production, modeling, optimization and control of industrial processes have long been recognized as important and challenging problems.

Descriptive systems and modeling approaches play a fundamental role in capturing process dynamics and support process control applications (Seborg et al. (2016)). Traditionally, a description of system dynamics involves the use of a set of mathematical models to represent the physical and chemical phenomena, static and dynamic behaviors, as well as the causal relationships among observed quantities, including manipulated inputs and technical indexes. These mathematical models are widely used in model predictive control (MPC), state estimation, soft sensing, process monitoring, fault detection and diagnosis (FDD), operational optimization, plant design, process simulation, control performance evaluation, etc. (Olivier & Craig (2013), Pantelides & Renfro (2013)). The types of system models are many, caused by differences in the modeling approach and the purpose of the model. According to Ljung (2010), system models can be categorized using a whole palette of grey shades from white to black, including white models (first principle models described by e.g. differential/algebraic equations (DAEs)), off-white models (white models with unknown or uncertain parameters described by e.g. state-space models), smoke-grey models (e.g., semi-physical modeling), steel-grey models (models linearized around an operating point), slate-grey models (e.g., hybrid models, block-oriented models) and black models (e.g., neural network, support vector machines). All these models are essentially aimed at extracting facts about the process dynamics from the measured data, information and knowledge.

Therefore, determination of the model structure, utilization of measured data, information and knowledge of reaction mechanisms are vital in process modeling. Among these models, hybrid models provide more flexibility in defining the model structure and utilizing available information (von Stosch et al. (2017)). Akkisetty et al. (2010) proposed a semi-empirical hybrid model which integrates a population balance model

[*] Corresponding author
Email address: ychh@csu.edu.cn (Chunhua Yang)

with a neural network model to predict the milled particle size distribution. In Alavi et al. (2014), a neural network model is incorporated within the mass balance model of an adiabatic fixed-bed reactor to reduce the computation load. Keskitalo & Leiviskä (2015) combined several artificial neural networks into ensembles to account for the difference between the mechanistic model outputs and the real process values. Chaffart & Ricardez-Sandoval (2018) proposed a hybrid model for the simulation of thin film deposition. The multiscale mechanistic model combined continuum differential equations describing the transport of the precursor gas phase and a stochastic partial differential equation (SPDE) predicting the evolution of the thin film surface. In order to accurately predict the thin film growth over wide operating ranges, an artificial neural network was trained to predict the coefficients of the mechanistic model. These hybrid modeling approaches mainly combine first-principle models with neural network models that are used to predict the modeling error or the coefficients of the first principle model. However, the multimodal and time-varying properties of the process dynamics are not considered.

With the shrinking of high grade mineral resources, mixed mineral supply, and fluctuating feed conditions, industrial processes in the current age are highly complex. The process dynamics exhibit different modes and vary with time. It is often not possible for traditional process models to include all the required information of such systems. Varying degrees of model uncertainty therefore exist, depending on the modeling effort. A descriptive model structure that comprehensively describes such a system and that reduces model uncertainty to a minimum, should be able to:

(i) Support the comprehensive and precise description of an industrial process.

(ii) Be *”flexible enough to cover many relevant nonlinear phenomena, at the same time as they allow inclusion of physical insight in order not to be too flexible”* (Ljung (2010)).

(iii) *”Describe static and dynamic process characteristics in the whole operational range”* (Hodouin et al. (2001)).

Correspondingly, a descriptive system supporting such a model type should comprehensively cover the essential factors influencing process dynamics. In addition, in order to realize smart and optimal manufacturing, such descriptive system should also be able to:

(i) Serve as an interface for digitalization and visualization.

(ii) Be compatible with both data analytics and control design.

(iii) Serve as a container for different sources of information collected along the production life-cycle.

(iv) Systematically support process control applications along the production life-cycle, e.g., modeling, optimization, FDD and control.

In this study, an extended state space descriptive system, namely Comprehensive State Space (CSS), is first designed to cope with the high complexity of industrial processes. The CSS descriptive system is a vector space which covers the essential factors that influence the process dynamics including inlet conditions, reaction conditions and output states. Then, under the CSS descriptive system, a hybrid first principles/machine learning modeling framework, is proposed. In the hybrid modeling framework, a practical industrial process is viewed as an interconnected system whose dynamics are formulated as a combination of a nominal term and a deviation term. The nominal term is a first-principle model (FPM) of the process. The deviation term accounts for the affects of the high-dimensional influence factors uncovered by the FPM. It is expressed in a regression form with its inputs being the low-dimensional deep process features extracted from production data via stacked auto-encoders (SAE) (Vincent et al. (2010)). This model structure contains the underlying physicochemical principles while incorporating high-dimensional influence factors in a low-dimensional manner. To handle the multimodal and time-varying properties of process dynamics, the deep process features are first used to divide the CSS into subspaces. A subspace is a partition of CSS indicating certain operating condition. By identifying model parameters under different operating conditions, sub-models of the process are obtained. Then, the process model is expressed as a weighted sum of the sub-models. The weights are the probabilities that the current operating point belongs to different operating conditions and changes with the movement of the operating point in the CSS. Moreover, the CSS description enables the visualization of the evolution of the operating point and the digitalization of the process.

The rest of this paper is organized as follows. The complexity of industrial processes is first analyzed in Section 2. Then, the definition of the comprehensive state space descriptive system is given in Section 3. Based on the CSS descriptive system, a hybrid first principles/machine learning modeling framework is proposed in Section 4. For demonstration purpose, the feasibility and capability of the hybrid modeling framework and CSS descriptive system is illustrated via a case study in Section 5. Conclusions are drawn in Section 6.

2. Highly complex industrial processes

A plant is designed as a combination of multiple unit processes with specific functions. An entire plant could be viewed as a material flow and processing network in which a unit process or a single piece of equipment acts as a node. Each unit process is interconnected with other unit processes through mass/heat transfers, recycles or reentrances (Fig. 1) (Lee & Lee (2014)). As a consequence, the inlet conditions of a unit process is prone to variations caused by:

(i) The fluctuations of the physicochemical properties and composition of the raw materials caused by the change of suppliers, price volatility of raw material and reagents, etc.

(ii) Fluctuations in the feed flow rate.

(iii) Disturbances in upstream unit processes.

(iv) Plant-wide adjustment caused by malfunctions, maintenance, etc.

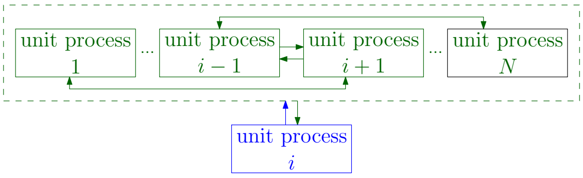

Figure 1: A plant composed of N unit processes

In a unit process or a single piece of equipment, there can exist multiple main reactions, each reaction corresponding to a reaction step, especially for processes with slow dynamics or with complex production technology. In addition, the heterogeneous nature of the raw feed materials with its associated elements/compounds causes side reactions to occur besides the main reactions for which the process was designed. Whereas the kinetic model of the main reactions can be derived by applying chemical and physical principles, the practical dynamics of the main reactions are more complex (Fig. 2):

(i) The side reactions can interact with the main reactions by competition or promotion.
(ii) The dynamics of the side reactions and their interaction mechanisms with the main reactions are only partially known in most cases.
(iii) The variation of the physicochemical properties and composition of the feed material can change the type of the side reactions, and the proportions of the side and main reactions.

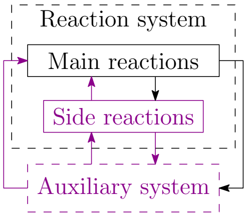

Figure 2: The interactions between the main reactions, side reactions and auxiliary system in a unit process

Take the zinc hydrometallurgy process as an example. In its cobalt removal process, the main reaction that occurs is the electrode reaction between cobalt and zinc catalyzed by arsenic trioxide (Sun et al. (2014)). The copper ions from the previous copper removal processes are involved in the following two reactions:

$$\mathrm{Cu^{2+} + Zn = Cu + Zn^{2+}} \tag{1}$$

$$\mathrm{HAsO_2 + 3Cu^{2+} + 3H^{+} + 4.5Zn = Cu_3As + 4.5Zn^{2+} + 2H_2O} \tag{2}$$

Reaction (1) consumes zinc dust. Reaction (2) produces a Cu-As alloy on the surface of the zinc dust which then serves as a substrate for the cementation of cobalt ions.

$$\mathrm{HAsO_2 + Co^{2+} + 3H^{+} + 2.5Zn = CoAs + 2.5Zn^{2+} + 2H_2O} \tag{3}$$

Thus, an appropriate amount of copper ions would promote cobalt removal. An excessive amount of copper ions will however compete with the cobalt ions by consuming zinc dust, and hence impede the cobalt removal process.

Moreover, kinetic models are often not a reflection of current practice as production conditions can differ from the experimental conditions used to derive these models. Auxiliary systems often form part of unit process to supply necessary reaction conditions, e.g., temperature and pH (Fig. 2). So,

(i) Changing the configuration of the auxiliary system can also affect the dynamics of the controlled reactions.
(ii) In addition, change of the reaction conditions can hinder part of the reactions and promote others, resulting in the variation of the process dynamics.

Therefore, for each unit process or a single piece of equipment, the main reactions are exposed to the inlet and reaction conditions. The inlet conditions are affected by the upstream and downstream unit processes, while the reaction conditions are determined by the configuration of the auxiliary system and type and portion of the side reactions. In this sense, the practical dynamics of the main reactions is an outcome of the intricate interactions among the main reactions, internal environment (side reactions, auxiliary system) and external environment (interconnected unit processes) (Fig. 3).

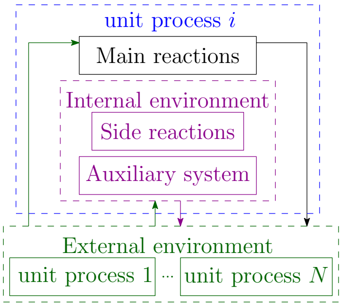

Figure 3: Interaction between the main reactions, internal environment and external environment

To sum up, an industrial process forms part of a multi-scale system ranging from molecular reactions, particle collisions, equipment interactions (basic production unit), plant-wide interactions (whole production life-cycle) to the global market (the global value-chain that affects plant production and profit). In addition, due to the inherent complexity and the aforementioned intricate interactions, the dynamics of the main reactions are nonlinear, containing uncertainties and are not consistent under different inlet and reaction conditions. Therefore, building a comprehensive and precise model of an industrial process, which can cover the dynamics of an industrial process under various operating conditions, requires knowledge from multiple disciplines. On the other hand, various restrictions in plant testing result in insufficient information about many aspects of the system dynamics, that could lead to significant variance errors in the resulting model (Lee & Lee (2005)). In order to more accurately describe highly complex industrial processes, additional descriptive methods, beyond those described in Ljung (2010), are required to increase the utilization of information.

3. Definition of comprehensive state space

As discussed in Section 2, the system dynamics and the intricate interactions that exist in a unit process can be illustrated by Fig. 4. If the output states are denoted as $\mathbf{x}_\mathrm{O}$, the inlet conditions as $\mathbf{x}_\mathrm{I}$, the reaction conditions as $\mathbf{x}_\mathrm{R}$, then the dynamics of the output states can be described by

$$\dot{\mathbf{x}}_\mathrm{O} = \mathbf{g}(\mathbf{x}_\mathrm{O}, \mathbf{x}_\mathrm{I}, \mathbf{x}_\mathrm{R}) \tag{4}$$

where $\mathbf{g}(\cdot)$ is a function representing the relation between $\dot{\mathbf{x}}_\mathrm{O}$ and $\mathbf{x}_\mathrm{O}$, $\mathbf{x}_\mathrm{I}$, $\mathbf{x}_\mathrm{R}$. In this sense, the traditional state-space, which has its states being the output states $\mathbf{x}_\mathrm{O}$, can be extended, by incorporating the inlet conditions $\mathbf{x}_\mathrm{I}$ and reaction conditions $\mathbf{x}_\mathrm{R}$, to a 'Comprehensive state space' (Fig. 5).

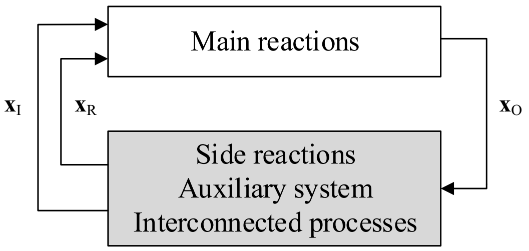

Figure 4: Unit process interactions

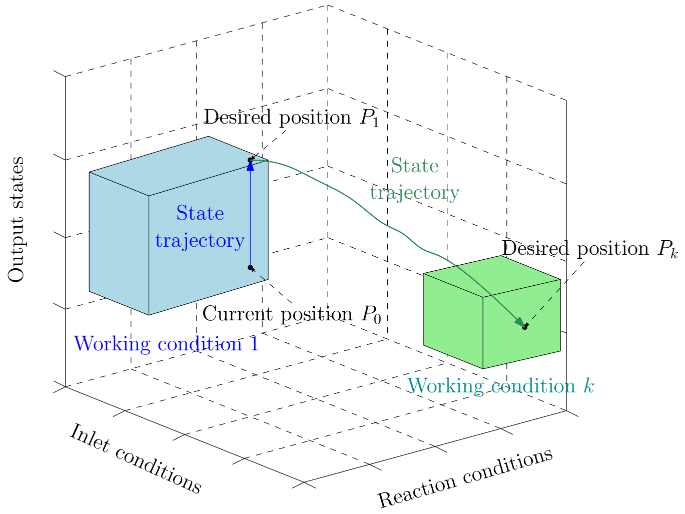

Figure 5: Illustration of comprehensive state space (P_0, P_1, P_k are different points in comprehensive state space)

Comprehensive state space : Comprehensive state space is a three dimensional vector space with each dimension being the codes of:

- Output states ($\mathbf{x}_\mathrm{O}$): the output states that describe the dynamic behavior of the main reactions, which are mainly the controlled technical indexes, e.g., outlet ion concentration of a continuous stirred tank reactor.
- Inlet conditions ($\mathbf{x}_\mathrm{I}$): the quantity, physicochemical properties, composition of the feed material, e.g., inlet flow rate, species concentration.
- Reaction conditions ($\mathbf{x}_\mathrm{R}$): the conditions under which the main and side reactions take place. e.g. temperature, pH, stirring rate.

As $\mathbf{x}_\mathrm{O}$, $\mathbf{x}_\mathrm{I}$ and $\mathbf{x}_\mathrm{R}$ are high-dimensional, they are compressed into one-dimensional variables according to certain coding rules (Bierbrauer (2016)). As shown in Fig. 6, $\mathbf{x}_\mathrm{O}$, $\mathbf{x}_\mathrm{I}$ and $\mathbf{x}_\mathrm{R}$ are coded as hexadecimal numbers. Different digits of the hexadecimal code represent different variables, e.g., three variables in the reaction conditions $\mathbf{x}_{\mathrm{R}1}$, $\mathbf{x}_{\mathrm{R}2}$, $\mathbf{x}_{\mathrm{R}3}$ take the value of '06', '2A' and '05' respectively.

The CSS description system covers more influence factors than the traditional state space composed solely of output states, which reduces the uncertainty in modeling. As shown in Figures. 5 and 6, each point in the comprehensive state space represents different operating points with corresponding attributes, including model structure, value of model parameters, type of operating condition, etc. The coordinate of a operating point depends on the value of output states, inlet conditions and reaction conditions (Fig. 6). The 'coordinate-attributes' information vectors of each point in the CSS form a comprehensive description of the process. On the other hand, the CSS serves as a container for the 'coordinate-attributes' information pairs, which can then act as a digital twin of the physical process.

In the following two sections, a study is performed to show how the CSS can support operating condition recognition and process modeling, or in other words, how to obtain the attributes of the operating condition and process model under the CSS framework.

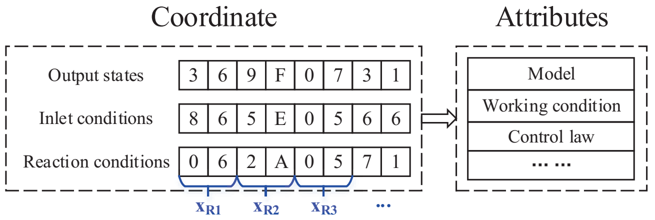

Figure 6: Coordinate and attributes of a point in comprehensive state space

Remark 1: Consider the output states $\mathbf{x}_\mathrm{O}$ and a fixed-length time interval $[t_0, t_f]$. If the initial values of the output states $\mathbf{x}_\mathrm{O}(t_0)$ are the same, and the inlet conditions and reaction conditions are the same during $[t_0, t_f]$, then starting from any t_0, the final value of $\mathbf{x}_\mathrm{O}(t_f)$ is the same.

Remark 2: In most cases, the manipulated variables affect the system dynamics by changing the reaction conditions $\mathbf{x}_\mathrm{R}$, e.g., in the cobalt removal process, the flow rate of spent acid affects the pH, the dosage of zinc dust affects the overall oxidation-reduction atmosphere which is represented by the oxidation-reduction potential (Sun et al. (2013)). However, if under certain working conditions the inlet conditions have to be changed, some variables that determine the inlet conditions then become manipulated variables.

Remark 3: Inside a sufficiently small subspace in the entire 'comprehensive state space', the system dynamics, i.e. the model structure and parameters can be regarded as constant. As in different subspaces, the inlet conditions, reaction conditions and output states are different. Therefore, different subspaces indicate different operating conditions.

Remark 4: If the process behavior inside each subspace can be obtained, then the comprehensive state space can serve as a digital twin of the physical process. The operators can then access the location of the current operating condition in the comprehensive state space to follow and monitor the evolution of the process.

4. A hybrid first principles/machine learning modeling framework

In this section, a CSS-based hybrid first principles/machine learning modeling framework is presented. The process is modeled as a combination of a nominal first-principle model and a machine-learning based input-output model accounting for the deviation between the nominal model and actual process dynamics. The low-dimensional deep process features are first extracted to divide the CSS into different partitions indicating different operating conditions. Then, the sub-models under each operating condition are obtained by identifying the model parameters. The dynamic process model is formulated as a weighted sum of the sub-models. The weights are the probabilities that the current operating point belongs to different operating conditions and update continuously with the movement of the working point in the CSS.

4.1. Deep feature extraction

Deep feature extraction is used to transform high-dimensional raw data into a low-dimensional deep process feature set which can be used to detect the patterns in the inputs via multiple levels of representation (LeCun et al. (2015), Chen et al. (2016), Shang et al. (2014)). Deep learning methods with higher layers of representation can learn very complex nonlinear functions. SAE is a type of unsupervised deep network, which is composed of multiple-level and stacked auto-encoders. As shown in Fig. 7, an auto-encoder has an input layer, output layer and one hidden layer. It aims to reproduce the input signal in the output layer via encoding and decoding, and extract the latent representation of inputs by limiting the number of hidden units. An auto-encoder is configured by minimizing the 'input-output' approximation error, while the extracted latent representation serves as the inputs of its successive auto-encoder.

Consider the *i*th auto-encoder that first extracts a set of latent variables from the inputs,

$$\mathbf{Y}_i = \mathbf{H}(\mathbf{W}_i\mathbf{X}_i + \mathbf{b}_i) \tag{5}$$

then the decoder reconstructs the outputs from the latent variables

$$\hat{\mathbf{Y}}_i = \mathbf{H}^{'}(\mathbf{W}_i^{'}\mathbf{Y}_i + \mathbf{b}_i^{'}) \tag{6}$$

where $\mathbf{X}_i \in \mathbb{R}^{n_i}$ represents the input, $\mathbf{Y}_i \in \mathbb{R}^{m_i}$ represents the hidden variables, $\hat{\mathbf{Y}}_i \in \mathbb{R}^{n_i}$ represents the output, $\mathbf{H}(\cdot)$ and $\mathbf{H}^{'}(\cdot)$ are the element-wise activation functions, e.g., sigmoid function, $\mathbf{W}_i \in \mathbb{R}^{m_i \times n_i}$, $\mathbf{W}_i^{'} \in \mathbb{R}^{n_i \times m_i}$ and $\mathbf{b}_i \in \mathbb{R}^{m_i}$, $\mathbf{b}_i^{'} \in \mathbb{R}^{n_i}$ are the weighting matrices and bias vectors respectively, that can be identified by minimizing the reconstruction error on a training set with M samples:

$$\ell_{\mathrm{SAE}}(\mathbf{W}_i, \mathbf{W}_i^{'}, \mathbf{b}_i, \mathbf{b}_i^{'}) = \sum_{j=1}^{M} \|\mathbf{X}_i^{(j)} - \hat{\mathbf{Y}}_i^{(j)}\|^2 \tag{7}$$

where $\mathbf{X}_i^{(j)}$ and $\hat{\mathbf{Y}}_i^{(j)}$ are the inputs and the reconstruction of the inputs of the jth training sample.

The pre-training of the SAE is conducted in a layer-wise manner. After all the auto-encoders are pre-trained, the hidden variables of the last auto-encoder are taken as the deep process feature set $\mathbf{z} = \mathbf{Y}_{\mathrm{L}}$. The initial regression function parameters of the deep process features α are identified by minimizing the regression error:

$$\ell_{\mathrm{Regression}}(\alpha_1, \cdots, \alpha_N) = \sum_{j=1}^{M} \|\mathbf{x}_{\mathrm{O}}^{\mathrm{deviation}^{(j)}} - \mathbf{E}(\mathbf{z}^{(j)})\|^2 \tag{8}$$

where $\mathbf{x}_{\mathrm{O}}^{\mathrm{deviation}^{(j)}}$ and $z_i^{(j)}$ are the deviation terms and the deep process feature variables of the jth training sample. $\mathbf{E}(\cdot)$ is a regression function set of the deep process feature $\mathbf{z}$. After pre-training, $[\alpha_1, \cdots, \alpha_N]$ and $[\mathbf{W}_i, \mathbf{W}_i^{'}, \mathbf{b}_i, \mathbf{b}_i^{'}]$ are used as initial solutions to fine-tune the weights and minimize the overall regression modeling error via back-propagation.

4.2. Deep feature space partitioning

After the extraction, the deep process features are fed to the operating condition classifier to divide the deep process feature space, e.g. Support Vector Machine (SVM), Logistic Regression (LR), *k*-Nearest Neighbors (*k*-NN), Decision Tree (DT), and Random Forest (RF). In this study, a sequential-manner feature space dividing approach is proposed. At first a rough division of the feature space is done using a *k*-Dimensional Tree (KD-Tree) (Bentley (1975), Zhang et al. (2018)). Then a LR classifier is adopted for finer divisions (Harrell (2015)). After the feature space is divided, the partitions of the deep process feature space are 'decoded' to the subspaces of the comprehensive state space. The steps for subdividing the process state are briefly introduced in Section 4.2.1 and 4.2.2.

4.2.1. Rough division using a KD-Tree

A KD-Tree is a space partitioning algorithm for a *k*-dimensional space. It is a variant of a binary tree with each node being *k*-dimensional. It splits the space into two half-spaces at each level iteratively until a desired number of partitions is achieved. The detailed steps are as follows (Fig. 8):

Step 1: Determine the desired numbers of partitions N_P.

Step 2: Calculate the required number of features and the

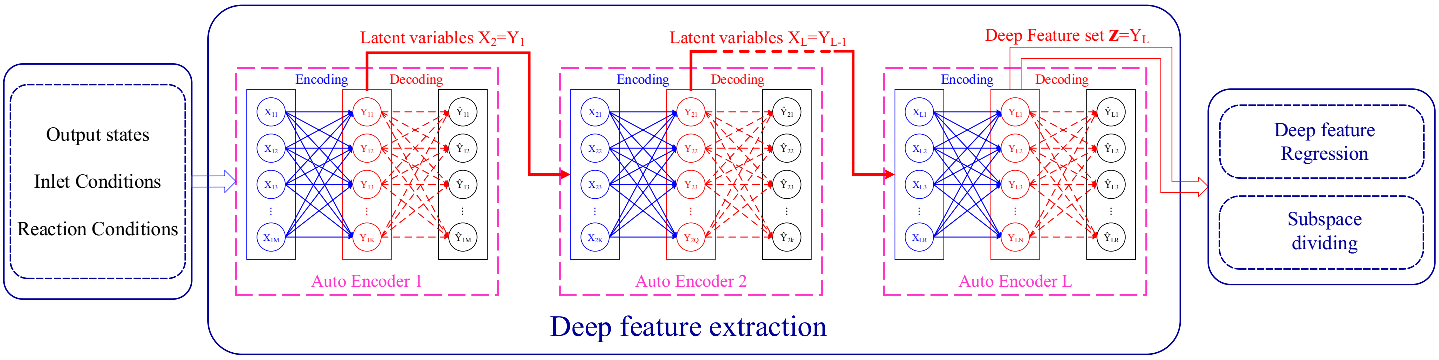

Figure 7: Deep feature extraction, regression, and subspace dividing

number of splits. If $2^{N-1} < N_P \leqslant 2^N$, $N \geqslant 1$, then N features and N_S number of splits are required.

$$N_S = (2^{N-1} - 1) + (N_P - 2^{N-1}) = N_P - 1 \tag{9}$$

Step 3: Normalize the feature data. Calculate the variances of the features, and rank the features according to the magnitude of their variances. Let $j = 1$.
Step 4: For the jth feature, find its median value, for the partition $\mathrm{S}_{(j-1)q}$, $q = 1$ to $2^{(j-1)}$, generate a splitting hyperplane which is perpendicular to the corresponding axis and taking the median value on the axis of the jth feature in the feature space. Rank the resulting new half-spaces $\mathrm{S}_{j(2q-1)}, \mathrm{S}_{j(2q)}$ according to the magnitude of the variance of the $(j+1)$th feature.
Step 5: If $j < N - 1$, then let $j = j + 1$, and repeat ***Step*** 4 until $j = N - 1$.
Step 6: If $2^j < N_P$, then from $q = 1$ to $q = 2^j$, divide the partition S_{jq} using the Nth feature until $2q + (2^j - q) = N_P$. The original feature space S_0 is partitioned into $\{\mathrm{S}_{\mathrm{N}1}, \mathrm{S}_{\mathrm{N}2}, \cdots, \mathrm{S}_{\mathrm{N}(2q)}, \mathrm{S}_{\mathrm{j}(q+1)}, \cdots, \mathrm{S}_{\mathrm{j}(2^j)}\}$ (denoted as $\{\mathrm{S}^{(1)}, \mathrm{S}^{(2)}, \cdots, \mathrm{S}^{(\mathrm{N_P})}\}$ hereafter).

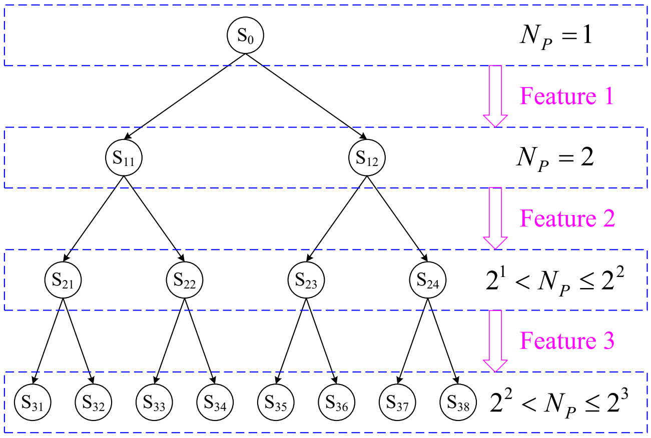

Figure 8: Rough division using a KD-Tree

Remark 5: The feature variance is used as criterion when doing rough division. A large feature variance indicates a wide distribution of the data points on that feature dimension, which has a higher dividing efficiency compared with other features with smaller variances.

Remark 6: The variance and covariance of $\mathbf{z}$ determines the first and second order properties of the deep process features (Ljung (1999)), so that the statistical features of $\mathbf{z}$ can also be used for process monitoring and fault detection.

4.2.2. Fine division based on LR

The KD-Tree based rough division process provides an initial solution for the LR-based fine division. LR classifier estimates and discriminates among the probabilities that an observation belongs to different classes:

$$P(y|\mathbf{z}) = \frac{1}{1 + \mathrm{e}^{-\sigma(\mathbf{z})}} \tag{10}$$

$$\sigma(\mathbf{z}) = \beta_0 + \sum_{i=1}^{N} \beta_i z_i \tag{11}$$

where $P(y|\mathbf{z})$ is the probability that input $\mathbf{z}$ belongs to the class with label y, e.g. if $P(y|\mathbf{z}) > 0.5$, i.e. $\sigma(\mathbf{z}) > 0$, then $\mathbf{z}$ belongs to class y. $\sigma(\mathbf{z}) = 0$ is the decision boundary. $\boldsymbol{\beta} = [\beta_0, \beta_1, \cdots, \beta_N]$ are the coefficients to be learned for each class. The detailed steps are as follows:
Step 1: Let $r = 1$.
Step 2: For partition $S^{(r)}$, train the coefficient set $\boldsymbol{\beta}^{(r)}$ by minimizing the regression error:

$$\begin{aligned} \ell_{\mathrm{LR}}^{(r)}(\boldsymbol{\beta}^{(r)}) &= \sum_{j=1}^{M} [\frac{1}{1 + \mathrm{e}^{-\sigma^{(r)}(\mathbf{z}_j)}} - y^{(j)}]^2 \\ &= \sum_{j=1}^{M} [\frac{1}{1 + \mathrm{e}^{-[\beta_0^{(r)} + \sum_{i=1}^{N} \beta_i^{(r)} z_i^{(j)}]}} - y^{(j)}]^2 \end{aligned} \tag{12}$$

where $y^{(j)}$ is the class label of the jth sample obtained by rough division.
Step 3: If $r < N_P$, $r = r + 1$, repeat ***Step*** 2.
Step 4: The partition with boundary $\sigma^{(r)}(\mathbf{z}) = 0$ is the rth subspace of the entire feature space.

Remark 7: The partitioning of the deep process feature space can be transformed to the partitioning of the original CSS.

The borders of the subspaces are not time-invariant. When a sufficient amount of new data samples appear, the process of roughly and finely dividing the feature space, should be repeated to update the partition.

4.3. A hybrid modeling framework

This section presents the hybrid modeling framework in the context of CSS. To start with, two typical methods to describe process dynamics, i.e. state-space based first principle modeling (SS-FPM) and machine learning based input/output modeling (ML-IOM), are revisited.

4.3.1. First principle based state-space modeling

First principle models that are based on the inherent physical or chemical laws of a process, e.g. mass and energy balances, reaction kinetics, hydrodynamics, and thermodynamics, can often be described in state-space form. State-space models consist of a set of input, output and state variables that are related by first-order differential or difference equations. The state-space approach is powerful as it provides a unifying framework for representing many classes of system equations including stochastic, nonlinear and time-varying multivariable systems (Brogan (1991)).

The model parameters in such state-space models are typically physical coefficients or constants (Ljung (1999)). Examples of such models include a continuously-stirred tank heater model (Thornhill et al. (2008)), a flotation cell model (Bascur (1982)), and a grinding circuit model (Le Roux et al. (2013)). However, this modeling approach requires significant process knowledge, which is not always available (Kadlec et al. (2009)). As a result not all the factors that influence the process are captured in the model.

4.3.2. Machine learning based input/output modeling

Machine learning based input/output models represent empirical mathematical or statistical correlations between input and output variables derived from plant data, especially for complex systems whose rigorous theoretical model involves a large number of DAEs and unknown parameters (Seborg et al. (2016), Ge (2017)).

ML-IOM involves the use of machine learning approaches (unsupervised learning, supervised learning, semi-supervised learning, reinforcement learning) which project the real process to a data space or latent variable space and construct statistical models for correlation analysis, prediction, soft sensing, process monitoring, pattern recognition, and FDD (Ge et al. (2017)). The main actions involved in ML-IOM approaches include regression, classification, clustering, coordinate transformation, and statistical properties analysis. These methods make use of a large volume of production data that contain useful information and knowledge about the process dynamics and running status. In addition, by projection, the evolution of operating conditions can be visualized in the data space. However, the performance of data driven modeling relies on the quality and operation ranges covered by the training data.

4.3.3. Differences and connections

The two modeling methods described in Sections 4.3.1 and 4.3.2 use different sources of data, information and knowledge (Fig. 9), and have their pros and cons:

(i) The SS-FPM utilizes inherent physical or chemical laws such as the underlying physicochemical laws of a process. However, not all the factors that influence the process are captured in the model.

(ii) The ML-IOM utilizes routinely collected production data and machine learning algorithms. It can approximate the relationship between process inputs and outputs with very high accuracy. However, it does not capture the internal dynamics of a system.

However, both methods are informed by probability theory and statistics, e.g., conservation laws are derived from experimental data, and first principle models often contain empirical relationships. The combination of these two methods could increase the utilization rate of data, information and knowledge, thus leading to the discovery of more facets of the process dynamics.

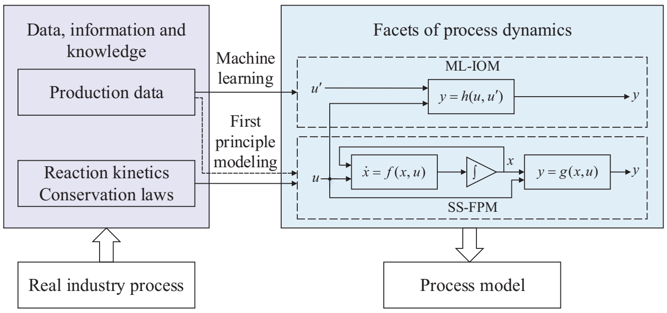

Figure 9: Differences and connections between state-space and data-driven modeling

4.3.4. CSS-based hybrid modeling

The system dynamics described by (4) can be decomposed as a combination of a nominal term and a deviation term

$$\dot{\mathbf{x}}_{\mathrm{O}} = \mathbf{f}(\mathbf{x}_{\mathrm{O}}, \Theta, \mathbf{u}) + \boldsymbol{\epsilon}(\mathbf{x}_{\mathrm{O}}, \mathbf{x}_{\mathrm{I}}, \mathbf{x}_{\mathrm{R}}) \tag{13}$$

where $\mathbf{f}(\cdot)$ is the nominal term or the first principle model containing the main reactions (white model), Θ represents the model parameters which take different values under different configurations of the inlet and reaction conditions, which can be identified using historical operation data. $\boldsymbol{\epsilon}(\cdot) = \mathbf{g}(\cdot) - \mathbf{f}(\cdot)$ is the deviation between the nominal model and the real dynamics.

This formulation keeps the nominal dynamics of the main reactions in a state-space form, and can account for the intricate interactions by introducing $\epsilon(\cdot)$ which then needs to be determined using data analytics (grey or black model), as shown in Fig. 10.

Considering each operating condition, the process dynamics can be realized as following:

$$\mathbf{x}_{\mathrm{O}}(t_f) = \mathbf{x}_{\mathrm{O}}^{\mathrm{nominal}}(t_f) + \mathbf{x}_{\mathrm{O}}^{\mathrm{deviation}}(t_f) \tag{14}$$

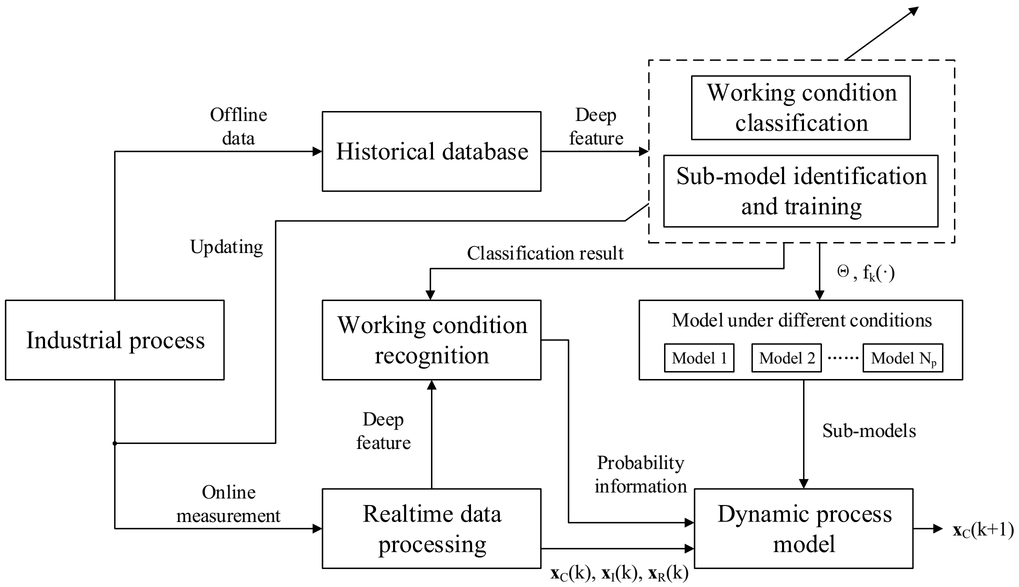

Figure 11: Hybrid modeling framework in the context of CSS

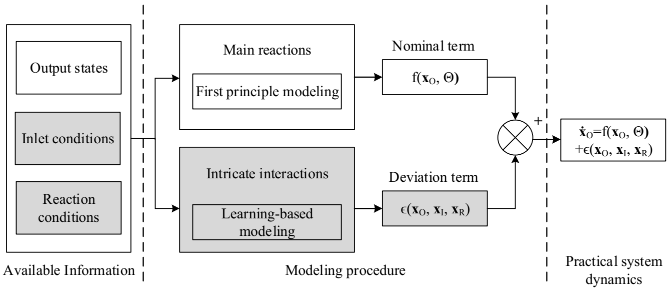

Figure 10: Dynamic modeling framework in the context of CSS

where

$$\mathbf{x}_{\mathrm{O}}^{\mathrm{nominal}}(t_f) = \mathbf{x}_{\mathrm{O}}^{\mathrm{nominal}}(t_0) + \int_{t_0}^{t_f} \mathbf{f}(\mathbf{x}_{\mathrm{O}}, \Theta)dt \tag{15}$$

$$\mathbf{x}_{\mathrm{O}}^{\mathrm{deviation}}(t_f) = \mathbf{x}_{\mathrm{O}}^{\mathrm{deviation}}(t_0) + \int_{t_0}^{t_f} \boldsymbol{\epsilon}(\mathbf{x}_{\mathrm{O}}, \mathbf{x}_{\mathrm{I}}, \mathbf{x}_{\mathrm{R}})dt \tag{16}$$

$\mathbf{x}_{\mathrm{O}}^{\mathrm{nominal}}(t_f)$ is the nominal term derived from the FPM of the main reactions (Buzzi-Ferraris & Manenti (2009), Pantelides & Renfro (2013)). $\mathbf{x}_{\mathrm{O}}^{\mathrm{deviation}}(t_f)$ is the deviation term with the output states $\mathbf{x}_{\mathrm{O}}$, inlet conditions $\mathbf{x}_{\mathrm{I}}$ and reaction conditions $\mathbf{x}_{\mathrm{R}}$ as its inputs. Thus, the value of $\mathbf{x}_{\mathrm{O}}$ can be expressed as

$$\mathbf{x}_{\mathrm{O}}(t_f) = \mathbf{x}_{\mathrm{O}}(t_0) + \int_{t_0}^{t_f} [\mathbf{f}(\mathbf{x}_{\mathrm{O}}, \Theta) + \boldsymbol{\epsilon}(\mathbf{x}_{\mathrm{O}}, \mathbf{x}_{\mathrm{I}}, \mathbf{x}_{\mathrm{R}})]dt \tag{17}$$

or, more practically, in a nonlinear discrete form

$$\begin{aligned} \mathbf{x}_{\mathrm{O}}(k+1) = \mathbf{x}_{\mathrm{O}}(k) &+ \mathbf{f}_k(\mathbf{x}_{\mathrm{O}}(k), \Theta(k)) \\ &+ \boldsymbol{\epsilon}_k(\mathbf{x}_{\mathrm{O}}(k), \mathbf{x}_{\mathrm{I}}(k), \mathbf{x}_{\mathrm{R}}(k)) \end{aligned} \tag{18}$$

where k indicates the kth sampling instant, $\mathbf{f}_k(\mathbf{x}_{\mathrm{O}}(k), \Theta(k))$ and $\boldsymbol{\epsilon}_k(\mathbf{x}_{\mathrm{O}}(k), \mathbf{x}_{\mathrm{I}}(k), \mathbf{x}_{\mathrm{R}}(k))$ are discretized counterparts of $\int_{t_k}^{t_{k+1}} \mathbf{f}(\mathbf{x}_{\mathrm{O}}, \Theta)dt$ and $\int_{t_k}^{t_{k+1}} \boldsymbol{\epsilon}(\mathbf{x}_{\mathrm{O}}, \mathbf{x}_{\mathrm{I}}, \mathbf{x}_{\mathrm{R}})dt$, respectively.

To make the model usable, the high-dimensional matrix $\boldsymbol{\epsilon}_k(\mathbf{x}_{\mathrm{O}}(k), \mathbf{x}_{\mathrm{I}}(k), \mathbf{x}_{\mathrm{R}}(k))$ is approximated in this study by the nonlinear regression form

$$\boldsymbol{\epsilon}_k = \mathbf{U}(\mathbf{z}(k)) = [\mathrm{U}_1(\mathbf{z}(k)) \quad \mathrm{U}_2(\mathbf{z}(k)) \cdots \mathrm{U}_{\mathrm{N}}(\mathbf{z}(k))]^{\mathrm{T}} \tag{19}$$

where $\mathbf{z}$ is the vector of low-dimensional deep process features extracted from the high-dimensional contributors, and $\mathbf{U}(\cdot)$ is the regression function set of the deep process feature variables.

As the process evolves in the CSS, the model parameters are subject to the inlet conditions, reaction conditions and output states. However, these parameters are identified for each operating condition, not each point in the CSS. Consider the time-varying characteristic of the production environment, the probability information provided by the LR can be incorporated into the dynamic process model (Fig. 11).

$$\mathbf{x}_{\mathrm{O}}(k+1) = \mathbf{x}_{\mathrm{O}}(k) + \sum_{i=1}^{N_p} P(i|\mathbf{z})[\mathbf{f}_k^{(i)}(\mathbf{x}_{\mathrm{O}}(k), \Theta(k)) + \mathbf{U}^{(i)}(\mathbf{z}(k))] \tag{20}$$

where $\mathbf{f}_k^{(i)}(\mathbf{x}_{\mathrm{O}}(k), \Theta(k))$ and $\mathbf{U}^{(i)}(\mathbf{z}(k))$ indicate the corresponding terms under operating condition i, i.e. sub-models. Compared with some typical hybrid modeling approaches (Psichogios & Ungar (1992), Abonyi et al. (2002)), the proposed CSS-based modeling approach considers the multimodality and updates the weights of each sub-model to enable the dynamic modeling of the process.

5. Case study

In this section the CSS descriptive framework is used to develop a process model for a cobalt removal process.

5.1. *Process description*

Cobalt removal is performed in a unit process in zinc hydrometallurgy. Its main function is to decrease the cobalt ion concentration in the zinc sulfate solution such that the cobalt ion concentration after removal is lower than a safety limit. The cobalt removal process usually consists of four continuously stirred tank reactors and a thickener (Fig. 12). Zinc dust is added into each reactor to replace the cobalt ions. Steam and spent acid are used to maintain the required reaction conditions (e.g., temperature 65 ∼ 75°C and pH 4.5 ∼ 5.5). Catalyst is provided to promote and accelerate cobalt removal. After retention in the consecutive reactors, the solution enters the thickener where solid-liquid separation takes place. The resultant solids, which can be used as crystal nuclei for cementation, is deposited and recycled to promote impurity removal. The overflow is further filtered and delivered to a subsequent process.

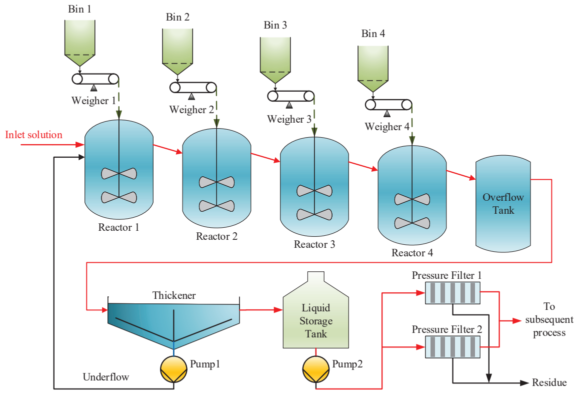

Figure 12: A cobalt removal process

The principle of cobalt removal is to use zinc dust to replace the cobalt ion impurities. However, the replacement is not trivial. As shown in Fig. 13, cobalt removal is catalysed by electro-positive metal salts, e.g., arsenic trioxide, which reacts with residual copper ions and zinc dust (Reaction 1), and provides reaction surfaces for cobalt removal (Reaction 2). Besides the main reactions, the residual copper ions and hydrogen ions in the solution also react with zinc dust (Reactions 3 and 4). Reactions 3 and 4 affect cobalt removal by competing with cobalt ions for zinc dust and changing the oxidation-reduction atmosphere. If zinc dust is overdosed, Reactions 5 and 6 will take place. These two reactions are undesirable, and produce elemental arsenic and highly toxic arsine gas. Besides, other elements and compounds in the inlet solution can result in other side reactions. The flow rates of the inlet solution, spent acid, arsenic trioxide, underflow, steam and zinc dust can affect the process dynamics. In addition, the interactions with its preceding and subsequent unit processes can also affect cobalt removal (Sun et al. (2013)).

5.2. *Experiment setup*

In the case study, 1050 data samples were collected over a 4 months period from an industrial zinc smelting plant (The data is scaled and desensitized for confidentiality reasons). The data were divided into two parts: 1000 samples for training and 50 samples for verification. Each data sample contains the values of the output states, inlet conditions and reaction conditions. The time span of the data samples ensure that a majority of the operating conditions are covered. In addition, with this amount of data samples, deep process feature extraction approaches can be applied. Knowledge of the reaction kinetics and the production process is obtained from studying the process mechanisms, data analysis, as well as discussions with experienced operators and technicians.

The aim of this case study is twofold, i.e. to test if the working conditions can be recognized and how well the model performs under the CSS framework. The steps of the case study are listed as follows:

(i) Extract the low dimensional deep process features from the high dimensional process variables.
(ii) Divide the deep process feature space into subspaces with each of them indicating different operating conditions.
(iii) Test if the working conditions can be recognized and how well the model performs under the CSS framework.
(iv) Build the nominal kinetic model, and formulate the process dynamics using the CSS-based modeling framework.
(v) Compare the performance of the nominal kinetic model and the CSS-based modeling approach.

5.3. *Results*

According to the reaction mechanism, the output states were selected as the cobalt ion outlet concentrations of the four reactors. The variables indicating reaction conditions were chosen as the oxidation reduction potential (ORP) of each reactor, flow rate of the arsenic trioxide and spent acid. ORP presents the overall oxidation-reduction atmosphere inside the reactor, while the flow rate of the arsenic trioxide and spent acid determine the catalytic condition and the pH of the solution respectively. The temperature affects the reaction rate. However, the solution temperature in the reactor rarely change, and was therefore omitted from the simulation. The flow rate, pH, and concentrations of the metallic ions (e.g., Co^{2+}, Cu^{2+}, Zn^{2+}, etc.) in the feed solution demonstrate the inlet conditions (Table 1). To sum up, there are 4 output states, 12 variables indicating inlet conditions and 6 for the reaction conditions. Therefore, the dimension of output states, inlet conditions and reaction conditions is 22 in total. The correlations among the condition variables and output states are obtained by Pearson correlation analysis, as shown in Table 2.

Table 1: Variables selected for the CSS

State type	Variables
Output states	c_1, c_2, c_3, c_4
Reaction conditions	v_1, v_2, v_3, v_4, f_{As}, f_{Acid}
Inlet conditions	f_0, m_{pH}, c_0, c_{Cu}, c_{Zn}, c_{Cd}, c_{Ni}, c_{As}, c_{Sb}, c_{Ge}, c_{Fe}, c_{Ca}

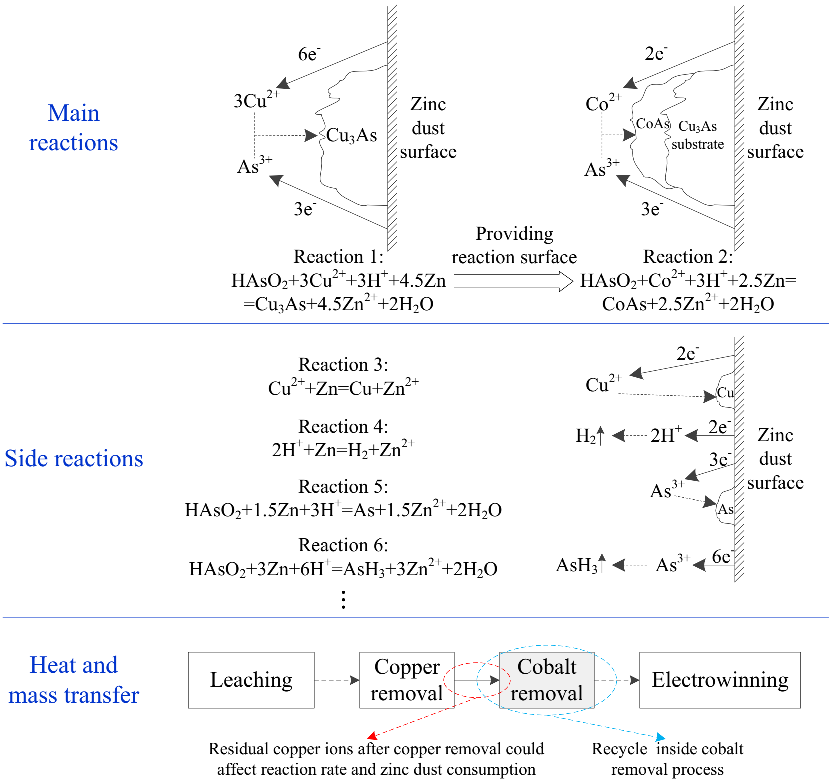

Figure 13: Main and side reactions in a cobalt removal process

Table 2: The correlations between the condition variables and output states

Variables	c_1	c_2	c_3	c_4
v_1	0.2143	0.1476	0.1572	0.3085
v_2	0.0580	0.0419	0.0206	-0.0863
v_3	-0.1221	-0.1567	-0.2421	-0.1648
v_4	0.1511	0.0595	0.0494	0.0129
f_{As}	0.3712	0.3427	0.3759	0.3942
f_{Acid}	0.0128	-0.0555	0.0420	0.1625
f_0	0.3467	0.2015	0.2529	0.2664
m_{pH}	0.1390	0.0683	0.0469	0.0354
c_0	0.5910	0.3890	0.3085	0.4931
c_{Cu}	0.1819	0.2220	0.2317	0.3073
c_{Zn}	0.0481	0.0938	0.0696	0.0772
c_{Cd}	-0.2684	-0.1675	-0.1493	-0.1023
c_{Ni}	0.0182	-0.0515	-0.0377	-0.0005
c_{As}	0.1889	0.1144	0.1432	0.2587
c_{Sb}	-0.0666	-0.0958	-0.0936	-0.0247
c_{Ge}	-0.1512	-0.2045	-0.1987	-0.1254
c_{Fe}	0.1264	0.0946	0.0443	-0.0388
c_{Ca}	0.1299	0.0040	-0.0385	-0.0737

Based on a process analysis, the number of operating conditions was determined as $N_P = 7$. Three deep process features are therefore required to divide the deep process feature space into 7 subspaces. To extract the 3 deep process features, two auto encoders were used to form an SAE. The number of latent variables in the two auto encoders are 18 and 3 respectively. The activation function adopted is the sigmoid function:

$$f_{\mathrm{sig}}(x) = \frac{1}{1 + \mathrm{e}^{-x}} \tag{21}$$

Then, rough division was conducted based on the variance of the 3 deep process features, and the resulting subspaces were denoted as S_{31} (Working condition type 1), S_{32} (Type 2), S_{33} (Type 3), S_{34} (Type 4), S_{35} (Type 5), S_{36} (Type 6), S_{24} (Type 7) (Figs. 8, 14). Then, fine division based on LR followed. The LR provides probability class information, which is useful especially for the junctions of two or more subspaces (Fig. 14). Fig. 14 shows the location and the probability that the operating point belongs to different operating conditions of the 50 test samples in the deep process feature space. The 30th sample is on the border of operating conditions S35 and S36. The probability that it belongs to these two operating conditions are 85% and 15%, respectively. The corresponding partitioning of the CSS is shown in Fig. 15.

The nominal part of the model is derived using mass balances and first order reaction kinetics. Assuming ideal mixing conditions, i.e. the reaction rate and solution temperature are

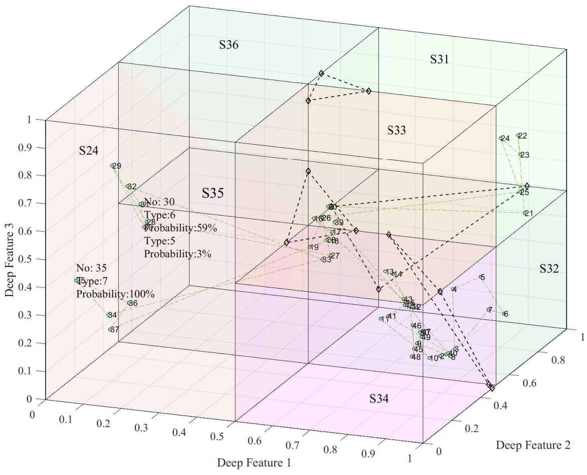

Figure 14: Partitioning of the deep process feature space

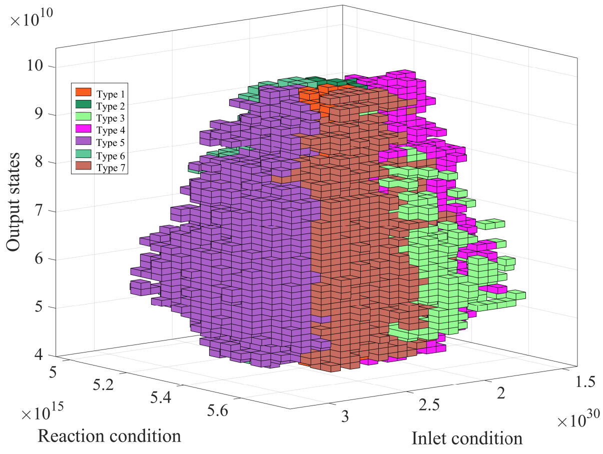

Figure 15: Partitioning of the CSS

uniform throughout the reactor, then:

$$\frac{dc_i}{dt} = \frac{f_{i-1}}{V}c_{i-1} - \frac{f_i}{V}c_i - r_i c_i \tag{22}$$

where c_i, r_i and f_i are the outlet cobalt ion concentration, reaction rate, and outlet flow rate of the ith reactor respectively, with $i = 1, 2, \cdots, N$. V is the volume of the reactor. c_0 and f_0 are the cobalt ion concentration and the flow rate of the inlet solution of the first reactor, respectively.

The reaction rate term r can be derived using the Arrhenius equation and electrode kinetics (Antropov (1977), Zha (2002)):

$$r = A_0 \beta g_s e^{-\frac{E_e + 2\gamma F(v - v_{eq})}{RT}} \tag{23}$$

where v is the Oxidation Reduction Potential (ORP). It represents the oxidation-reduction ability of the solution, and is controlled by the addition of zinc dust, u. The physical meanings of the parameters in the process model are listed in Table 3. To increase the identification accuracy, model parameters A_0, β and g_s were combined as a new parameter A_β in the identification of model parameters (Buzzi-Ferraris & Manenti (2009)). The value of model parameters under different operating conditions are shown in Table 4. The performance of the nominal kinetic model is shown in Fig. 16 and Table 5. For comparison purpose, the performance of pure data-driven model (PDDM) is also included. The PDDM is a combination of SAE and radius basis neural network (RBFNN). The input of the RBFNN is the deep process features extracted using SAE. The output of the RBFNN is the estimated value of cobalt ion concentrations of each reactor. As indicated by the results, compared with kinetic model, PDDM is less sensitive to the change of operating conditions.

Table 3: Physical meaning of the parameters in the process model

Parameter (unit)	Physical meaning
$A_0(s^{-1})$	frequency factor of the reaction
β	reaction surface area available on a unit area of the crystal nucleus
g_s	weight of crystal nucleus per unit volume of the reactor
$E_e(\mathrm{J \cdot mol^{-1}})$	standard activation energy of the reaction
γ	variation factor between the electrode potential and the cathode activation energy
$v_{eq}(\mathrm{V})$	equilibrium potential of the cathode reaction
$v(\mathrm{V})$	oxidation reduction potential of the solution in the reactor
$T(\mathrm{K})$	reaction temperature
$\mathrm{F(C \cdot mol^{-1})}$	Faraday constant, F = 96485
$\mathrm{R(J \cdot mol^{-1}K^{-1})}$	ideal gas constant, R = 8.314

To account for the unknown dynamics, a deviation term $c_i^{\text{deviation}}$ is added to the nominal model, then under certain operating condition, for each reactor:

$$c_i(t) = c_i(0) + \int_0^t \left(\frac{f_{i-1}}{V}c_{i-1} - \frac{f_i}{V}c_i - r_i c_i \right) d\tau + c_i^{\text{deviation}} \tag{24}$$

The deviation term is a neural network model with its inputs being the deep process features. Its output is the estimated

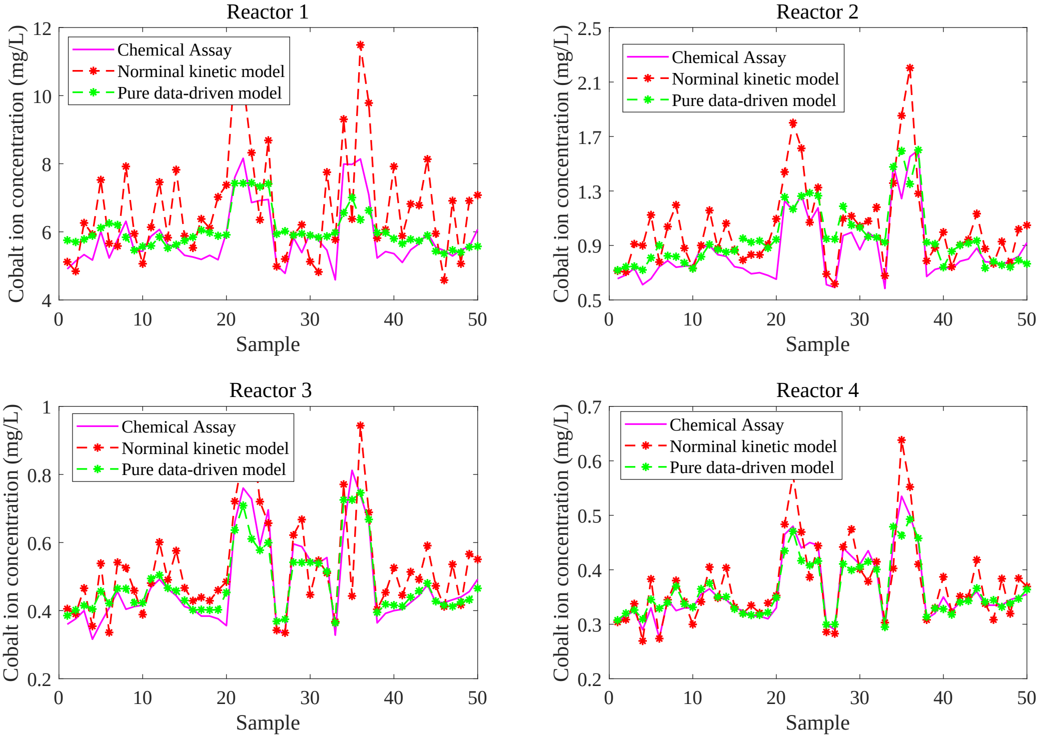

Figure 16: Performance of the nominal kinetic model and pure data-driven model

Table 4: Values of model parameters under different operating conditions

Working condition	Reactor	A_β	E_e	γ	v_{eq}
S31	Reactor 1	6981106	72932	0.618	-0.300
	Reactor 2	7286393	79847	0.625	-0.332
	Reactor 3	8073669	79998	0.706	-0.373
	Reactor 4	8983144	71945	0.664	-0.436
S32	Reactor 1	7362697	72747	0.600	-0.300
	Reactor 2	7832393	79968	0.689	-0.367
	Reactor 3	8668537	77202	0.671	-0.392
	Reactor 4	6562077	72362	0.603	-0.404
S33	Reactor 1	5984033	72400	0.600	-0.300
	Reactor 2	8365405	78822	0.601	-0.348
	Reactor 3	5308410	79043	0.608	-0.348
	Reactor 4	8186106	72876	0.750	-0.463
S34	Reactor 1	7638676	73391	0.600	-0.300
	Reactor 2	8858660	77157	0.797	-0.427
	Reactor 3	6213033	77216	0.791	-0.429
	Reactor 4	5874675	78829	0.621	-0.368
S35	Reactor 1	5957729	72265	0.600	-0.300
	Reactor 2	7970208	79557	0.630	-0.352
	Reactor 3	8510620	77102	0.734	-0.417
	Reactor 4	8817141	73734	0.695	-0.440
S36	Reactor 1	5349874	72240	0.600	-0.300
	Reactor 2	6516429	74700	0.616	-0.385
	Reactor 3	5255406	78698	0.772	-0.414
	Reactor 4	7004144	71521	0.704	-0.457
S31	Reactor 1	5311958	72624	0.600	-0.300
	Reactor 2	5438328	77854	0.671	-0.382
	Reactor 3	8817776	70562	0.617	-0.440
	Reactor 4	8756903	79996	0.640	-0.379

deviation between the nominal model and practical system dynamics. Eq.(24) is a basic model of process dynamics. Its model parameters take different values under different working conditions. The formulation of the overall system dynamics was the same as Eq.(20). The performance of the integrated model is shown in Fig. 17. The performance comparison between the integrated model and the nominal model is given in Table 5, and Fig. 18 using a four plot figure (Yuan et al. (2018)), respectively. The proposed modeling approach has higher accuracy on average, and its performance is more stable. This indicates by combining reaction kinetics, production data and information about the current operating conditions, the system dynamics can be described more comprehensively. This is due to the higher information utilization rate under the CSS framework.

5.4. *Discussion*

It can be observed from the above results that the CSS framework provides:

(i) Improved modeling accuracy: By using the CSS modeling approach, the ARE of the four reactors are lower than 10%, which is satisfactory for an industrial application. The norm of the Fisher information matrix of the kinetic model and the integrated model have orders of magnitude of 10^4 and 10^{15}, respectively. This indicates that more information is utilized in the

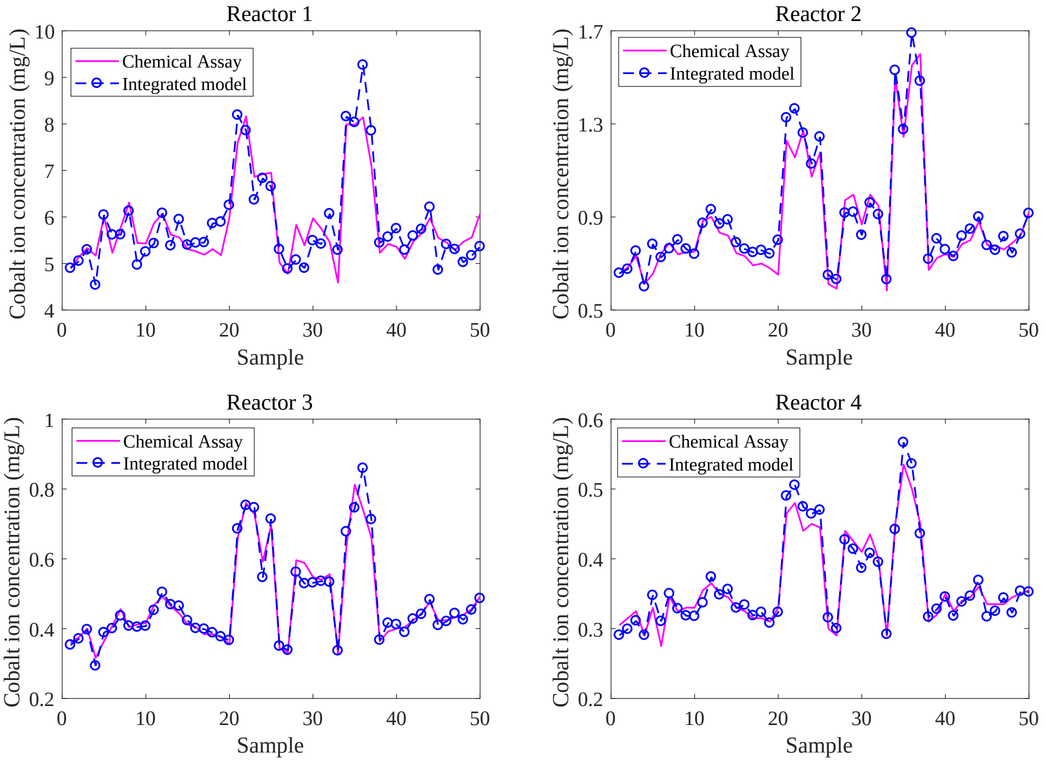

Figure 17: Performance of the integrated model under the CSS modeling framework

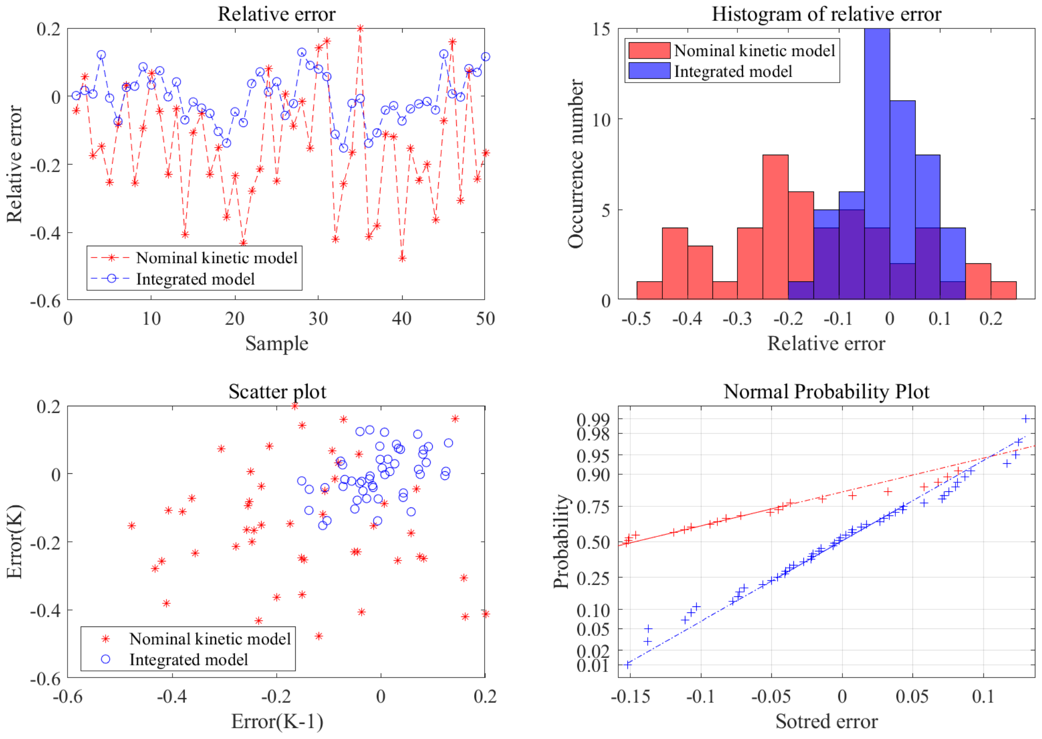

Figure 18: Four plot performance comparison between the integrated model and the nominal kinetic model

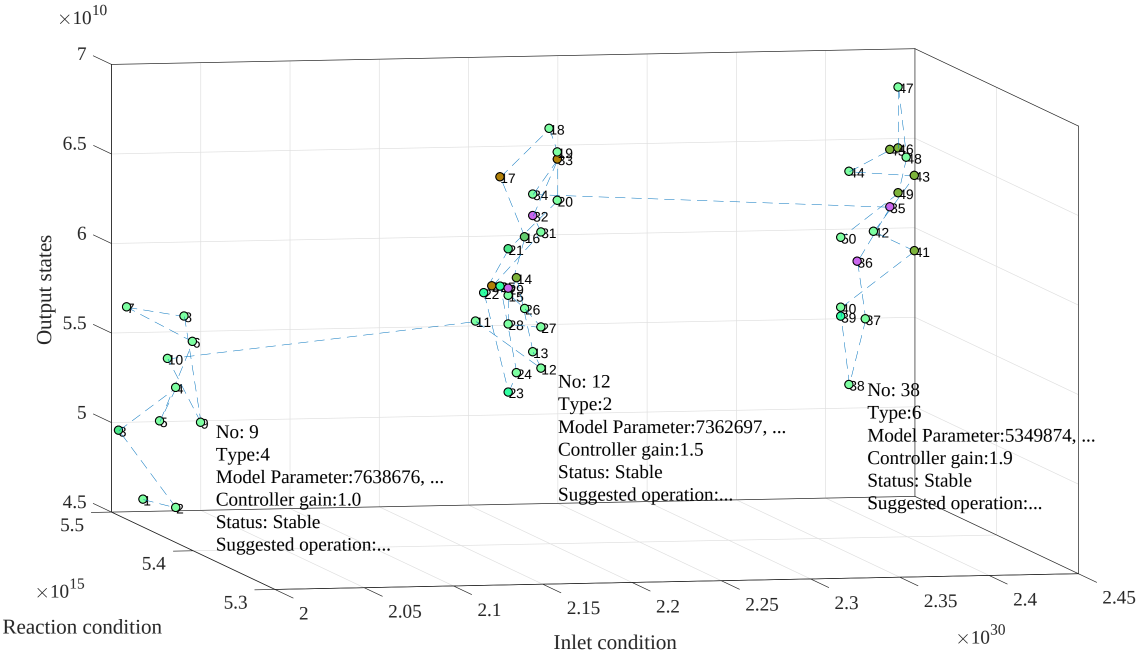

Figure 19: Trajectory of the test samples in the CSS

Table 5: Performance comparison between nominal kinetic model, pure data-driven model and CSS modeling framework using average relative error (ARE) and root mean square error (RMSE)

Performance measure	Reactor 1	Reactor 2	Reactor 3	Reactor 4
ARE of FPM	15.19%	15.55%	16.14%	17.21%
ARE of PDDM	8.38%	10.70%	6.67%	8.32%
ARE of CSS model	5.7%	4.02%	3.33%	7.90%
RMSE of FPM	1.3967	0.6132	0.2506	0.0706
RMSE of PDDM	0.6296	0.3877	0.1069	0.0397
RMSE of CSS model	0.4166	0.1568	0.0679	0.0314

integrated modeling approach. Therefore, higher model accuracy is obtained as an outcome.

(ii) Intuitive process monitoring: CSS provides a visual abstraction of the physical process. The location and the evolution of the process in the CSS can be monitored intuitively. Fig. 19 shows the evolution trajectory of the 50 test samples. Different marker colors indicate different operating conditions. The variation of model parameters (see Table 4) and operating conditions along the evolution trajectory can be observed from Fig. 19.

(iii) Comprehensive descriptive capability: The CSS descriptive framework covers the essential influence factors of the process. These factors are organized to form a digitized container of the physical process which can accommodate various attributes of an operating point under many different operating conditions, including e.g., model parameters, controller gain, status, and suggested operation.

6. Conclusions

In this study, a comprehensive state space descriptive system and a corresponding hybrid first principles/machine learning modeling framework were proposed. As a wide variety of process control applications is model-based, the CSS descriptive system can better support process control in terms of better understanding of process dynamics. In addition, its intuitive nature provides a platform for process monitoring. It provides a comprehensive yet 'not overly complex' way to describe the dynamics of a process that enables the digitalization and visualization of a physical process. The CSS descriptive system is an open framework, which can serve as a container for the fusion of data, information and knowledge from various sources. Due to the above characteristics of CSS, it can also serve as a 'digital twin' of a physical process. The physical process and its CSS counterpart can therefore act as a 'cyber-physical component' in a smart factory. The results presented in this paper are preliminary and mainly considers process modeling in the context of CSS. For future extension, the following should be considered:

- The dividing of the deep process feature space is based on the variance of deep process features. This dividing approach is easy to implement. However, the physical meaning of the subspaces are not clear. Therefore, generating subspaces with concrete physical meanings can provide more useful information to the operators.
- Transforming the system dynamics into a linear time varying (LTV) format by using a U-model (Zhu et al. (2019)) approach to describe the nominal term and deviation term respectively. This would resulted in an

LTV model of the physical process, which can bridge the gap between the nonlinear system description and linear controller design approaches with well established properties.

- Integrating modeling, control, and estimation in an interactive and systematic framework to gradually increase the understanding of the system dynamics and achieve intelligent autonomous control.

Solutions to the above problems and the methods proposed in this study can enrich the theoretical foundations for the smart and optimal manufacturing in the process industries.

References

Abonyi, J., Madar, J., & Szeifert, F. (2002). Combining first principles models and neural networks for generic model control. In R. Roy, M. Köppen, S. Ovaska, T. Furuhashi, & F. Hoffmann (Eds.), *Soft Computing and Industry* (pp. 111–122). London: Springer.

Akkisetty, P. K., Lee, U., Reklaitis, G. V., & Venkatasubramanian, V. (2010). Population balance model-based hybrid neural network for a pharmaceutical milling process. *Journal of Pharmaceutical Innovation*, *5*, 161–168.

Alavi, M., Jazayeri-Rad, H., & Behbahani, R. M. (2014). Optimizing the feed conditions in a dimethyl ether production process to maximize methanol conversion using a hybrid first principle neural network approach. *Chemical Engineering Communications*, *201*, 650–673.

Antropov, L. I. (1977). *Theoretical Electrochemistry*. Moscow: Mir Publishers.

Bascur, O. A. (1982). *Modelling and computer control of a flotation cell*. Ph.D. thesis , University of Utah, Salt Lake City, Utah, US.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, *18*, 509–517.

Bierbrauer, J. (2016). *Introduction to Coding Theory*. New York: Chapman and Hall/CRC.

Brogan, W. L. (1991). *Modern Control Theory*. (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

Buzzi-Ferraris, G., & Manenti, F. (2009). Kinetic models analysis. *Chemical Engineering Science*, *64*, 1061–1074.

Chaffart, D., & Ricardez-Sandoval, L. A. (2018). Optimization and control of a thin film growth process: A hybrid first principles/artificial neural network based multiscale modelling approach. *Computers & Chemical Engineering*, *119*, 465–479.

Chen, Y., Jiang, H., Li, C., Jia, X., & Ghamisi, P. (2016). Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, *54*, 6232–6251.

Craig, I., Aldrich, C., Braatz, R., Cuzzola, F., Domlan, E., Engell, S., Hahn, J., Havlena, V., Horch, A., Huang, B. et al. (2011). Control in the process industries. In T. Samad, & A. M. Annaswamy (Eds.), *The impact of control technology*. IEEE Control Systems Society.

Ge, Z. (2017). Review on data-driven modeling and monitoring for plant-wide industrial processes. *Chemometrics and Intelligent Laboratory Systems*, *171*, 16–25.

Ge, Z., Song, Z., Ding, S. X., & Huang, B. (2017). Data mining and analytics in the process industry: the role of machine learning. *IEEE Access*, *5*, 20590–20616.

Harrell, F. E. (2015). *Regression Modeling Strategies*. New York: Springer.

Hodouin, D., Jämsä-Jounela, S.-L., Carvalho, M., & Bergh, L. (2001). State of the art and challenges in mineral processing control. *Control Engineering Practice*, *9*, 995–1005.

Kadlec, P., Gabrys, B., & Strandt, S. (2009). Data-driven soft sensors in the process industry. *Computers & Chemical Engineering*, *33*, 795–814.

Keskitalo, J., & Leiviskä, K. (2015). Artificial neural network ensembles in hybrid modelling of activated sludge plant. In P. Angelov, K. T. Atanassov, L. Doukovska, M. Hadjiski, V. Jotsov, J. Kacprzyk, N. Kasabov, S. Sotirov, E. Szmidt, & S. Zadrożny (Eds.), *Intelligent Systems' 2014* (pp. 683–694). Cham: Springer.

Le Roux, J. D., Craig, I. K., Hulbert, D., & Hinde, A. (2013). Analysis and validation of a run-of-mine ore grinding mill circuit model for process control. *Minerals Engineering*, *43*, 121–134.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444.

Lee, J. H., & Lee, J. M. (2014). Progress and challenges in control of chemical processes. *Annual Review of Chemical and Biomolecular Engineering*, *5*, 383–404.

Lee, J. M., & Lee, J. H. (2005). Approximate dynamic programming-based approaches for input–output data-driven control of nonlinear processes. *Automatica*, *41*, 1281–1288.

Ljung, L. (1999). *System Identification: Theory for The User*. (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Ljung, L. (2010). Perspectives on system identification. *Annual Reviews in Control*, *34*, 1–12.

Olivier, L. E., & Craig, I. K. (2013). Model-plant mismatch detection and model update for a run-of-mine ore milling circuit under model predictive control. *Journal of Process Control*, *23*, 100–107.

Pantelides, C. C., & Renfro, J. (2013). The online use of first-principles models in process operations: Review, current status and future needs. *Computers & Chemical Engineering*, *51*, 136–148.

Psichogios, D. C., & Ungar, L. H. (1992). A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, *38*, 1499–1511.

Seborg, D. E., Mellichamp, D. A., Edgar, T. F., & Doyle III, F. J. (2016). *Process Dynamics and Control*. (4th ed.). Hoboken, NJ: John Wiley & Sons.

Shang, C., Yang, F., Huang, D., & Lyu, W. (2014). Data-driven soft sensor development based on deep learning technique. *Journal of Process Control*, *24*, 223–233.

von Stosch, M., Zhang, J., & Willis, M. (2017). Hybrid neural network modelling for process monitoring and control. In A. Basile, M. Alavi, & S. Curcio (Eds.), *Artificial Neural Networks in Chemical Engineering* (pp. 205–228). New York: Nova Science Publishers, Inc.

Sun, B., Gui, W., Wang, Y., & Yang, C. (2014). Intelligent optimal setting control of a cobalt removal process. *Journal of Process Control*, *24*, 586–599.

Sun, B., Gui, W., Wu, T., Wang, Y., & Yang, C. (2013). An integrated prediction model of cobalt ion concentration based on oxidation–reduction potential. *Hydrometallurgy*, *140*, 102–110.

Thornhill, N. F., Patwardhan, S. C., & Shah, S. L. (2008). A continuous stirred tank heater simulation model with applications. *Journal of Process Control*, *18*, 347–360.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, *11*, 3371–3408.

Yuan, X., Huang, B., Wang, Y., Yang, C., & Gui, W. (2018). Deep learning based feature representation and its application for soft sensor modeling with variable-wise weighted SAE. *IEEE Transactions on Industrial Informatics*, *14*, 3235–3243.

Zha, Q. (2002). *Introduction to the Kinetics of Electrode Processes*. Beijing: Science Press.

Zhang, J., Guo, H., Hong, F., Yuan, X., & Peterka, T. (2018). Dynamic load balancing based on constrained K-D tree decomposition for parallel particle tracing. *IEEE Transactions on Visualization and Computer Graphics*, *24*, 954–963.

Zhu, Q., Zhang, W., Zhang, J., & Sun, B. (2019). U-neural network-enhanced control of nonlinear dynamic systems. *Neurocomputing*, *352*, 12–21.