Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

Incorporating Process Knowledge into Latent Variable Models: An Application to Root Cause Analysis in Bioprocesses

Tobias Overgaard[ab], Maria-Ona Bertran[c], and John B. Jørgensen[a], and Bo F. Nielsen[a]*

[a]Technical University of Denmark, Department of Applied Mathematics and Computer Science, Kgs. Lyngby, Denmark
[b]Novo Nordisk A/S, PS API Manufacturing, Science & Technology, Bagsværd, Denmark
[c]Novo Nordisk A/S, PS API Expansions, Kalundborg, Denmark
*Corresponding Author: bfni@dtu.dk

ABSTRACT

Incorporating process knowledge from various sources often presents challenges in process development, optimization, and control. To utilize available knowledge, linking existing process models is a viable approach. This work introduces a methodology using latent variable models, specifically sequential and orthogonalized partial least squares (SO-PLS), to capture and quantify the contribution of first-principles knowledge in process models. Applied to a continuously stirred tank reactor (CSTR) case study, the methodology demonstrates how available knowledge can be quantified and how structural and parametric errors in first-principles are addressed using measured data. The methodology is discussed in relation to root cause analysis in bioprocesses.

Keywords: Process models, Root cause analysis, Latent variable models, Multiblock partial least squares

INTRODUCTION

Process development, optimization, and control based on process modeling are significantly advancing pharmaceutical manufacturing. These efforts are reducing development times and manufacturing costs, improving productivity and quality control, and enhancing overall process understanding [1]. Using terminology from knowledge management, process models can be seen as representations of accumulated expert knowledge generated over the years. These models enable the simulation of system behavior under various perturbations [2]. In the pharmaceutical industry, process models are typically developed for individual unit operations to capture specific product and technology phenomena. Recently, *integrated process models* have emerged, representing processes with multiple steps or unit operations. These models transform the output of one unit into the input of the next, simulating material flow across operations and capturing interactions between upstream and downstream equipment. Applications include control strategy design, risk analysis, experimental design, and process optimization [3-5], and they range from being first-principles-based [5] to fully data-driven [4].

In first-principles modeling, we rely on a physical understanding of the system, where the model's equations represent the known mechanisms driving the process. Parameters are typically tuned using data from well-designed laboratory experiments, allowing for some extrapolation beyond the calibration range. However, when comparing first-principles models to historical data from a full-scale commercial plant, outputs may not match accurately. This discrepancy can be due to limited knowledge of the process, oversimplification of physical phenomena, or inappropriate parameter values [6].

Due to their simplicity, ease of development, and increasing computational efficiency, data-driven models are an attractive alternative. These models involve selecting a structure and identifying parameters using available data [6], which can be readily sourced from full-scale plants. However, since many observations remain constant during plant operation and rarely change, extrapolation can be challenging for data-driven models.

Therefore, it is of interest to develop schemes that combine both modeling techniques to create more effective and efficient process models. For integrated process models specifically, this involves incorporating connectivity information from the process flowsheet along with first-principles knowledge from each unit operation.

Incorporating connectivity information into process models often relies on data from full-scale commercial manufacturing, where consecutive runs of an integrated

process are logged. Given the vast amount of data and the presence of complex, unknown physical phenomena in a full-scale plant, a data-driven foundation is often chosen for integrated process models. Among various methods, sequential-orthogonalized partial least squares (SO-PLS) has recently emerged as a promising approach [7-9]. SO-PLS sequentially applies regular PLS to different data blocks, each representing a specific unit operation, and uses orthogonalization between steps to eliminate overlapping information. By ordering the blocks with respect to the flowsheet topology and removing redundant information explained by upstream units through orthogonalization, this model naturally incorporates connectivity information. Additionally, being based on PLS, it effectively handles the high-dimensional data matrices common in full-scale plants.

Inspired by the use of SO-PLS to incorporate flowsheet connectivity information, this paper presents a systematic methodology to further augment the model with first-principles knowledge, utilizing elements from hybrid modeling and path modeling [10]. While the concept is initially demonstrated for a single unit operation, it can be readily extended to encompass a multi-unit process. The key questions we aim to address are:

- *How can we incorporate flowsheet connectivity information and unit-specific first-principles knowledge into a unified model?*
- *How can we quantify the variance explained by first-principles knowledge versus measured data?*
- *How can the approach be used to identify root causes of deviations from quality or productivity targets in an integrated bioprocess?*

This paper addresses the first two questions, while the last question will be explored in future work in combination with the framework presented in [9]. The paper is organized as follows: First, we provide background on SO-PLS and explain how this model fits within parallel hybrid architectures. Next, we describe the methodology for incorporating first-principles knowledge into the SO-PLS model. This concept is demonstrated using a toy example based on the van de Vusse reaction scheme [11]. Finally, we discuss this work in the context of integrated process modeling and present future work and challenges.

MODELING BACKGROUND

This work is based on partial least squares (PLS), also known as projection to latent structures. Consider an input data matrix $\boldsymbol{X} \in \mathbb{R}^{N\times J}$ and an output data matrix $\boldsymbol{Y} \in \mathbb{R}^{N\times K}$. The core principle of PLS is to build a model by relating the latent variables (LVs) associated with $\boldsymbol{X}$ and $\boldsymbol{Y}$. PLS achieves this by linearly transforming both $\boldsymbol{X}$ and $\boldsymbol{Y}$ into their respective LVs and then relating these transformed variables by maximizing their covariance. For more details, see [12]. In process modeling, PLS is particularly useful for batch processes, with specialized algorithms for handling batches of varying lengths. While process knowledge is often incorporated into PLS models through additional calculated variables, the direct integration of first-principles models with PLS has been explored to a limited extent [13].

When combining first-principles and data-driven models, often referred to as hybrid or grey-box models, two main architectures are commonly discussed. The first is the series approach, where a first-principles model is developed initially, followed by a data-driven model, such as neural networks, to determine the parameters. The second is the parallel approach, where a data-driven model is used to capture the error (residuals) between the actual process outputs and the estimated outputs from the first-principles model. In this approach, a neural network can serve as the data-driven model, but any data-driven model can be used [14].

As will be shown, the presented methodology bears resemblance to the parallel hybrid architecture. For reference, this architecture is illustrated in Fig. 1.

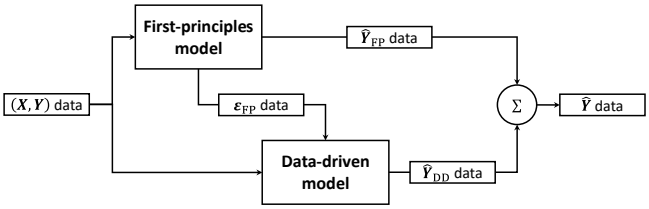

Figure 1. The parallel hybrid architecture. $\boldsymbol{\varepsilon}$ refers to the residuals of the estimates made by the first-principles model. $\widehat{\boldsymbol{Y}}$ refers to the estimates made by a model.

Sequential-orthogonalized PLS

To illustrate the SO-PLS method, we will focus on the case with two (autoscaled) input blocks, named $\boldsymbol{X} \in \mathbb{R}^{N\times J}$ and $\boldsymbol{Z} \in \mathbb{R}^{N\times L}$, and an (autoscaled) output block, or response matrix, referred to as $\boldsymbol{Y} \in \mathbb{R}^{N\times K}$. The multiblock linear regression model estimated by SO-PLS can be represented by the following equation

$$\boldsymbol{Y} = \boldsymbol{XB} + \boldsymbol{ZC} + \boldsymbol{F}, \tag{1}$$

where $\boldsymbol{B} \in \mathbb{R}^{J\times K}$ and $\boldsymbol{C} \in \mathbb{R}^{L\times K}$ are the regression coefficients, and $\boldsymbol{F} \in \mathbb{R}^{N\times K}$ is the residual matrix. As explained later in the methodology section, $\boldsymbol{X}$ will serve as the *mechanistic block*, containing output data generated by the first-principles model. In contrast, $\boldsymbol{Z}$ will be the *measured block*, containing various process variables, including some measured process outputs.

The SO-PLS algorithm is given in Tbl. 1 and relies solely on regular PLS [9]. Recall that regular PLS for input block $\boldsymbol{X}$ and output block $\boldsymbol{Y}$ produces a set of $\boldsymbol{X}$-scores ($\boldsymbol{T}$) summarizing the relationships between samples, a set of $\boldsymbol{X}$-loadings ($\boldsymbol{P}$) summarizing the relationships between

variables, and an optimal weight matrix ($\boldsymbol{R}$), transforming $\boldsymbol{X}$ into $\boldsymbol{T}$ through $\boldsymbol{XR} = \boldsymbol{T}$. The five steps of the algorithm are briefly described as follows: *(i)* First, a regular PLS regression is performed between the response matrix $\boldsymbol{Y}$ and mechanistic block $\boldsymbol{X}$, resulting in $\boldsymbol{X}$-scores $\boldsymbol{T}^{(1)}$ and $\boldsymbol{X}$-loadings $\boldsymbol{P}^{(1)}$.*(ii)* The measured block $\boldsymbol{Z}$ is orthogonalized with respect to $\boldsymbol{T}^{(1)}$ using the projection matrix

$$\boldsymbol{H} = \boldsymbol{T}^{(1)} \left(\left(\boldsymbol{T}^{(1)}\right)^{\intercal} \boldsymbol{T}^{(1)} \right)^{-1} \left(\boldsymbol{T}^{(1)}\right)^{\intercal}, \tag{2}$$

which is also commonly referred to as the *hat matrix* in statistics. The orthogonalization involves removing the shared information between $\boldsymbol{X}$ and $\boldsymbol{Z}$ embedded in the projection, resulting in $\boldsymbol{Z}^{\mathrm{ort}} = (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{Z}$. *(iii)* Next, a similar orthogonalization is performed on the response matrix, yielding the residuals $\boldsymbol{Y}^{\mathrm{ort}} = (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{Y}$. *(iv)* After orthogonalization, another regular PLS regression is performed between $\boldsymbol{Y}^{\mathrm{ort}}$ and $\boldsymbol{Z}^{\mathrm{ort}}$, resulting in $\boldsymbol{Z}^{\mathrm{ort}}$-scores $\boldsymbol{T}^{(2)}$ and $\boldsymbol{Z}^{\mathrm{ort}}$-loadings $\boldsymbol{P}^{(2)}$. *(v)* Finally, $\boldsymbol{T}^{(1)}$ and $\boldsymbol{T}^{(2)}$ are used as independent variables in a least squares regression with $\boldsymbol{Y}$. The estimate $\widehat{\boldsymbol{Y}}$ (with coefficients $\boldsymbol{Q}^{(1)}, \boldsymbol{Q}^{(2)}$) is given by

$$\widehat{\boldsymbol{Y}} = \boldsymbol{T}^{(1)}\boldsymbol{Q}^{(1)} + \boldsymbol{T}^{(2)}\boldsymbol{Q}^{(2)}. \tag{3}$$

Table 1: Pseudo-code for (two-block) SO-PLS model

Mechanistic block		
(i)	$\mathrm{PLS}(\boldsymbol{X}, \boldsymbol{Y}) \implies \boldsymbol{T}^{(1)}, \boldsymbol{P}^{(1)}$	PLS regression for $\boldsymbol{X}$ and $\boldsymbol{Y}$
(ii)	$\boldsymbol{Z}^{\mathrm{ort}} = (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{Z}$	Orthogonalization of $\boldsymbol{Z}$ on $\boldsymbol{T}^{(1)}$
(iii)	$\boldsymbol{Y}^{\mathrm{ort}} = (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{Y}$	Orthogonalization of $\boldsymbol{Y}$ on $\boldsymbol{T}^{(1)}$
Measured block		
(iv)	$\mathrm{PLS}(\boldsymbol{Z}^{\mathrm{ort}}, \boldsymbol{Y}^{\mathrm{ort}}) \implies \boldsymbol{T}^{(2)}, \boldsymbol{P}^{(2)}$	PLS regression for $\boldsymbol{Z}^{\mathrm{ort}}$ and $\boldsymbol{Y}^{\mathrm{ort}}$
(v)	$\boldsymbol{Y} = \boldsymbol{T}^{(1)}\boldsymbol{Q}^{(1)} + \boldsymbol{T}^{(2)}\boldsymbol{Q}^{(2)} + \boldsymbol{F}$	Least squares for $\boldsymbol{T}^{(1)}, \boldsymbol{T}^{(2)}$ and $\boldsymbol{Y}$

Remark 1: Letting $\boldsymbol{R}^{(1)}$ and $\boldsymbol{R}^{(2)}$ be the weight matrices used to obtain the scores $\boldsymbol{T}^{(1)}$ and $\boldsymbol{T}^{(2)}$ from $\boldsymbol{X}$ and $\boldsymbol{Z}^{\mathrm{ort}}$, the link between Eq. (1) and (3) follows from

$$\begin{aligned} \widehat{\boldsymbol{Y}} &= \boldsymbol{X}\boldsymbol{R}^{(1)}\boldsymbol{Q}^{(1)} + \boldsymbol{Z}^{\mathrm{ort}}\boldsymbol{R}^{(2)}\boldsymbol{Q}^{(2)} \\ &= \boldsymbol{X}\boldsymbol{R}^{(1)}\boldsymbol{Q}^{(1)} + (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{Z}\boldsymbol{R}^{(2)}\boldsymbol{Q}^{(2)} \\ &= \boldsymbol{X}\boldsymbol{R}^{(1)}\boldsymbol{Q}^{(*)} + \boldsymbol{Z}\boldsymbol{R}^{(2)}\boldsymbol{Q}^{(2)}, \end{aligned} \tag{4}$$

with $\boldsymbol{Q}^{(*)} := \boldsymbol{Q}^{(1)} - \left(\left(\boldsymbol{T}^{(1)}\right)^{\intercal} \boldsymbol{T}^{(1)} \right)^{-1} \left(\boldsymbol{T}^{(1)}\right)^{\intercal} \boldsymbol{Z}\boldsymbol{R}^{(2)}\boldsymbol{Q}^{(2)}$. The selection of $\boldsymbol{B}$ and $\boldsymbol{C}$ follows implicitly.

Remark 2: It is worth noting the strong resemblance between SO-PLS and the Frisch–Waugh–Lovell theorem from ordinary least squares. Note also that due to the symmetric and idempotent nature of $\boldsymbol{I} - \boldsymbol{H}$, the model would not change if $\boldsymbol{Y}$ is not orthogonalized. Furthermore, the regression coefficients $\boldsymbol{Q}^{(1)}$ and $\boldsymbol{Q}^{(2)}$ found in step *(v)* coincide with the (transposed) $\boldsymbol{Y}$-loadings that can be extracted from the PLS steps in *(i)* and *(iv)*.

Remark 3: Because the orthogonalization step removes redundant information (information in $\boldsymbol{Z}$ already explained by $\boldsymbol{X}$), the additional explained variance gained by adding the new block $\boldsymbol{Z}$ represents information not included in previous blocks. This makes the method useful in path modeling; a discipline of analyzing direct and indirect effects of different sets of variables on an output [9].

To briefly explain how this works for SO-PLS, we can use the notation in (1) to write

$$\widehat{\boldsymbol{Y}} = \boldsymbol{X}\boldsymbol{B} + \boldsymbol{H}\boldsymbol{Z}\boldsymbol{C} + (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{Z}\boldsymbol{C} = \boldsymbol{X}\widetilde{\boldsymbol{B}} + \boldsymbol{Z}^{\mathrm{ort}}\boldsymbol{C}, \tag{5}$$

with $\widetilde{\boldsymbol{B}} := \boldsymbol{B} + \boldsymbol{R}^{(1)}\boldsymbol{Q}^{(1)} \left(\left(\boldsymbol{T}^{(1)}\right)^{\intercal} \boldsymbol{T}^{(1)} \right)^{-1} \left(\boldsymbol{T}^{(1)}\right)^{\intercal}$. Note that by reordering the blocks and orthogonalizing on $\boldsymbol{Z}^{\mathrm{ort}}$ instead, a similar expression, $\widehat{\boldsymbol{Y}} = \boldsymbol{X}^{\mathrm{ort}}\boldsymbol{B} + \boldsymbol{Z}\widetilde{\boldsymbol{C}}$, can be obtained. Recall that the sum of squares of $\widehat{\boldsymbol{Y}}$ can be calculated as $\mathrm{tr}\left(\widehat{\boldsymbol{Y}}^{\intercal}\widehat{\boldsymbol{Y}}\right)$. Due to the orthogonality between blocks, we obtain the decomposition

$$\mathrm{tr}\left(\widehat{\boldsymbol{Y}}^{\intercal}\widehat{\boldsymbol{Y}}\right) = \mathrm{tr}\left(\widetilde{\boldsymbol{B}}^{\intercal}\boldsymbol{X}^{\intercal}\boldsymbol{X}\widetilde{\boldsymbol{B}}\right) + \mathrm{tr}(\boldsymbol{C}^{\intercal}(\boldsymbol{Z}^{\mathrm{ort}})^{\intercal}\boldsymbol{Z}^{\mathrm{ort}}\boldsymbol{C}). \tag{6}$$

Eq. (6) is referred to as the *overall effect*. The first term, $\mathrm{tr}\left(\widetilde{\boldsymbol{B}}^{\intercal}\boldsymbol{X}^{\intercal}\boldsymbol{X}\widetilde{\boldsymbol{B}}\right)$, is known as the *total effect* of $\boldsymbol{X}$, and the second term, $\mathrm{tr}(\boldsymbol{C}^{\intercal}(\boldsymbol{Z}^{\mathrm{ort}})^{\intercal}\boldsymbol{Z}^{\mathrm{ort}}\boldsymbol{C})$, is known as the *additional* effect of $\boldsymbol{Z}$. The total effect can be further decomposed into a *direct effect*, which represents the influence of $\boldsymbol{X}$ that is uncorrelated with $\boldsymbol{Z}$, i.e., $\mathrm{tr}(\boldsymbol{B}^{\intercal}(\boldsymbol{X}^{\mathrm{ort}})^{\intercal}\boldsymbol{X}^{\mathrm{ort}}\boldsymbol{B})$, and an *indirect effect,* which is the difference between the total and the direct effect. This variance decomposition will be useful for measuring how much of the variance is explained by the first-principles model ($\boldsymbol{X}$) and how much is corrected by the data ($\boldsymbol{Z}$). Due to the assumption of unit variance in data, the effects can be interpreted as relative explained variances.

PROPOSED METHODOLOGY

This section introduces a systematic methodology for incorporating first-principles process knowledge into the SO-PLS model, as depicted in Fig. 2. The framework utilizes first-principles models that describe the unit operations under study, typically consisting of nonlinear ordinary or partial differential and algebraic equations (PDAEs), which are validated through (small-scale) experiments. Furthermore, measurements from multiple (large-scale) historical batches are employed.

Step 1: Data augmentation by first-principles: In this step, the goal is to construct two data blocks using measurement data from the observed system and additional data generated from the first-principles model, initialized with the initial conditions ($\boldsymbol{X}_0$) from the process. Following the structure in [13], the measurement data include manipulated inputs ($\boldsymbol{U}$) and observed system output states ($\boldsymbol{X}_{\mathrm{UO}}$) (e.g., reaction temperatures). The sampled quality variables ($\boldsymbol{Y}$) (e.g., component concentrations) constitute the model output. Note that we only assume access to steady-state data, not full trajectories. However, the method can be extended to trajectory data using techniques like unfolding [13]. The synthetic output

states generated from the first-principles model are denoted $\boldsymbol{X}_{\mathrm{FP}}$. The model may include expressions for states not measured in the full-scale system, thereby augmenting the data matrix with additional information. The initial conditions of the unmeasured states are assumed to be known. Ultimately, we obtain the following data blocks:

- *Mechanistic block:* $\boldsymbol{X} = [\boldsymbol{X}_0, \boldsymbol{U}, \boldsymbol{X}_{\mathrm{FP}}]$.
- *Measured block:* $\boldsymbol{Z} = \boldsymbol{X}_{\mathrm{UO}}$.
- *Output block:* $\boldsymbol{Y}$.

Step 2: Model fitting by SO-PLS: In the next step, an SO-PLS model is fitted, with the number of components for each block selected based on the minimal root mean squared error obtained through K-fold cross-validation (RMSECV). The SO-PLS architecture, shown in Fig. 2, resembles that of Fig. 1.

Step 3: Variance decomposition by SO-PLS: Once the model is fitted, the overall variance in $\widehat{\boldsymbol{Y}}$ is decomposed into total (direct + indirect) and additional effects:

- *Total:* Variance explained by mechanistic block.
- *Direct:* Variance explained by part of mechanistic block uncorrelated with measured block.
- *Indirect:* Variance explained by part of mechanistic block correlated with measured block.
- *Additional:* Variance explained by measured block.

It is worth noting that with this methodology, the first-principles model does not necessarily need to be recalibrated to process data, as the data is projected into a latent space. Consequently, there can be biases between the true and synthetic states. The key point to consider is that the synthetic states, although biased, exhibit correlation structures comparable to the real system. This aspect of the methodology presents an opportunity to incorporate small-scale knowledge with full-scale data.

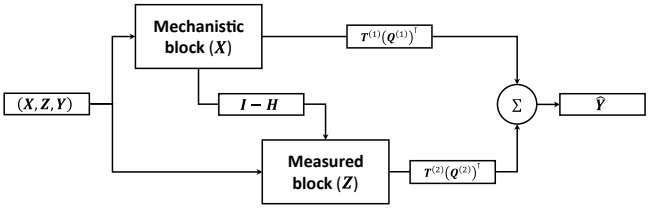

Figure 2. SO-PLS architecture. $\boldsymbol{I} - \boldsymbol{H}$ is the *residual-maker matrix*, like in parallel hybrid architectures.

TOY EXAMPLE

The proposed methodology is tested on a simulated case study involving a temperature-controlled, constant-volume continuous stirred tank reactor (CSTR) [10]. The reaction network consists in conversion of component A into component B and then into component C. A side reaction also produces component D which, along with component C, is an undesired by-product.

$$A \overset{k_1}{\rightarrow} B \overset{k_2}{\rightarrow} C,$$

$$2A \overset{k_3}{\rightarrow} D.$$

Let $\mathcal{C}$ and $\mathcal{R}$ denote the set of components and reactions respectively. Then $\mathcal{C} = \{A, B, C, D\}$ and $\mathcal{R} = \{1,2,3\}$. Further, let $\boldsymbol{r} \in \mathbb{R}^{|\mathcal{R}|}$ be a vector of rates of reactions. Then

$$r_1 = r_1(C_A, T) = k_1(T)C_A,$$
$$r_2 = r_2(C_B, T) = k_2(T)C_B,$$
$$r_3 = r_3(C_A, T) = k_3(T)C_A^2,$$

where C_A, C_B (and C_C, C_D) are component concentrations and T is the temperature. All variables are implicitly time dependent. The specific reaction rates are defined by the Arrhenius expression

$$k_i(T) = k_{i0} \exp\left(-\frac{E_i}{RT}\right), \quad i = 1,2,3.$$

We define the production rate vector $\boldsymbol{R} \in \mathbb{R}^{|\mathcal{C}|}$ using the stoichiometry of the reaction network

$$R_A = R_A(C_A, T) = -r_1(C_A, T) - 2r_3(C_A, T),$$
$$R_B = R_B(C_A, C_B, T) = r_1(C_A, T) - r_2(C_B, T),$$
$$R_C = R_C(C_B, T) = r_2(C_B, T),$$
$$R_D = R_D(C_A, T) = r_3(C_A, T).$$

By using the stoichiometric matrix $\boldsymbol{S} \in \mathbb{R}^{|\mathcal{R}| \times |\mathcal{C}|}$, this can be written compactly as $\boldsymbol{R} = \boldsymbol{S}^{\intercal}\boldsymbol{r}$, with

$$\boldsymbol{S} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ -2 & 0 & 0 & 1 \end{bmatrix}.$$

The reactor can be modelled by six mass and energy balances, yielding the ordinary differential equation system

$$\begin{aligned}
\frac{\mathrm{d}C_A}{\mathrm{d}t} &= \frac{F}{V}\left(C_{A,\mathrm{in}} - C_A\right) + R_A, \\
\frac{\mathrm{d}C_B}{\mathrm{d}t} &= -\frac{F}{V}C_B + R_B, \\
\frac{\mathrm{D}C_C}{\mathrm{d}t} &= -\frac{F}{V}C_C + R_C, \\
\frac{\mathrm{D}C_D}{\mathrm{d}t} &= -\frac{F}{V}C_D + R_D, \\
\frac{\mathrm{d}T}{\mathrm{d}t} &= \frac{F}{V}(T_{\mathrm{in}} - T) + \frac{UA}{\mu V c_p}(T_c - T) \\
&\quad - \frac{1}{\mu c_p}\frac{F}{V}(r_1\Delta H_1 + r_2\Delta H_2 + r_3\Delta H_3), \\
\frac{\mathrm{d}T_c}{\mathrm{d}t} &= \frac{1}{m_c c_{pc}}\left(Q_c + UA(T - T_c)\right),
\end{aligned} \tag{7}$$

where T_c is the temperature inside the cooling jacket. The parameters and measured variables of the system are presented in Tbl. 2. To introduce input uncertainty, the input variables are treated as random

variables.

Table 2: Values of the parameters and measured variables used to generate the dataset

Parameter	Description	Value
V	Volume	10 L
A	Heat transfer area	0.215 m^2
U	Heat transfer coef.	4032 kJ/(h m^2 K)
c_p	Heat capacity (mix.)	3.01 kJ/(kg K)
c_{pc}	Heat capacity (cool.)	2 kJ/(kg K)
μ	Density of mix.	0.9342 kg/L
m_c	Mass of cool. fluid	5 kg
ΔH_1	Heat of reaction 1	4.2 kJ/mol
ΔH_2	Heat of reaction 2	−11 kJ/mol
ΔH_3	Heat of reaction 3	−41.85 kJ/mol
k_{10}	Reaction rate const. 1	$1.287 \cdot 10^{12}$ h^{-1}
k_{20}	Reaction rate const. 2	$1.287 \cdot 10^{12}$ h^{-1}
k_{30}	Reaction rate const. 3	$9.043 \cdot 10^{9}$ L/(h mol)
E_1/R	Activation energy 1	9758.3 K
E_2/R	Activation energy 2	9758.3 K
E_3/R	Activation energy 3	8560 K
Variable	**Description**	**Distribution**
$C_{A,\mathrm{in}}$	Inflow concentration A	$\mathcal{N}(5.1\ \mathrm{mol/L}, 1^2)$
T_{in}	Inflow temperature	$\mathcal{N}(378.05\ \mathrm{K}, 5^2)$
F	Flowrate	$\mathcal{N}(141.9\ \mathrm{L/h}, 5^2)$
Q_c	Heat transfer rate	$\mathcal{N}(-1113.5\ \mathrm{kJ/h}, 50^2)$
Error	**Description**	**Value**
$\boldsymbol{\Sigma}$	Diffusion term	diag[0.01,0.01,0.01,0.01,0.1,0.1]
$\boldsymbol{E}$	Measurement noise	diag[0.02,0.02,0.02,0.02,0.5,0.5]

To introduce additional variability across different simulations, the deterministic system in Eq. (7) is augmented by a stochastic term. (7) can be represented as

$$\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = f\big(\boldsymbol{x}(t)\big),$$

where $\boldsymbol{x} = [C_A, C_B, C_C, C_D, T, T_c]^\intercal$. We can then write the corresponding system of stochastic differential equations as

$$\mathrm{d}\boldsymbol{x}(t) = f\big(\boldsymbol{x}(t)\big)\mathrm{d}t + \mathrm{d}\boldsymbol{\Sigma}\boldsymbol{W}(t), \tag{8}$$

where $\boldsymbol{W}(t) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}\mathrm{d}t)$ is a multivariate Wiener process, while the diffusion term $\boldsymbol{\Sigma}$ is given in Tbl. 2.

The stochastic system in (8) is simulated using the Euler-Maruyama method with a step size of 0.001 h and initial conditions $\boldsymbol{x}_0 = \big[C_{A,\mathrm{in}}, 0, 0, 0, T_{\mathrm{in}}, 375\big]^\intercal$. A total of 50 simulations are performed. To create the dataset from these simulations, the system is run until it reaches steady state. The final values of $\boldsymbol{x}(t_s)$ at steady state t_s are then recorded as data points. Additionally, measurement uncertainty is added by assuming that $\boldsymbol{x}(t_s)$ is observed with a noise component, $\boldsymbol{x}(t_s) + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{E})$ with $\boldsymbol{E}$ defined in Tbl. 2. As a result, the final dataset generated from Eq. (8) is an $\mathbb{R}^{50\times 7}$ matrix. The columns correspond to the measured variables and states ($C_{A,\mathrm{in}}$, T_{in}, F, Q_c, C_A, C_B, C_C, C_D, T).

Application of methodology

To illustrate the SO-PLS method, two case studies are considered: ***Case (i)*** with a structural error due to a missing reaction in the first-principles model, and ***Case (ii)*** with a severe parametric error. The goal is to assess how well the measured data corrects these errors.

Case (i): In *Step 1*, we assume access to a calibrated first-principles process model for the CSTR system. However, unlike the real system, we assume no knowledge of the third reaction, $2A \rightarrow D$. I.e., the model is similar to Eq. (7) with $r_3 = 0$. We collect the synthetic system states and process inputs in the mechanistic block $\boldsymbol{X} = \big[\boldsymbol{X}_0, \boldsymbol{U}, \widetilde{\boldsymbol{C}}_A, \widetilde{\boldsymbol{C}}_B, \widetilde{\boldsymbol{C}}_C, \widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{T}}_c\big] \in \mathbb{R}^{50\times 7}$, where $\big[\widetilde{\boldsymbol{C}}_A, \widetilde{\boldsymbol{C}}_B, \widetilde{\boldsymbol{C}}_C, \widetilde{\boldsymbol{T}}, \widetilde{\boldsymbol{T}}_c\big] \in \mathbb{R}^{50\times 5}$ are generated from the mis-specified first-principles model over 50 batches. For the output block, we assume access only to samples of components A and B, so $\boldsymbol{Y} = [\boldsymbol{C}_A, \boldsymbol{C}_B] \in \mathbb{R}^{50\times 2}$. The measured block contains observed system states, $\boldsymbol{Z} = [\boldsymbol{T}, \boldsymbol{T}_c] \in \mathbb{R}^{50\times 2}$.

In *Step 2*, the SO-PLS model is fitted, and the number of latent variables is chosen using 10-fold cross-validation, giving an optimal choice of $\{6,1\}$ latent variables for the $\boldsymbol{X}$ and $\boldsymbol{Z}$ blocks, respectively. The performance is compared to the first-principles model and the Hybrid PLS model proposed by [13] (where PLS is applied to the augmented matrix $[\boldsymbol{X}, \boldsymbol{Z}]$), shown in the first two rows of Tbl. 3. The Hybrid PLS model is cross-validated, yielding a model of 7 latent variables. The first-principles model shows low performance ($R^2 = 37.3\%$). The Hybrid PLS fit shows significant information contained in $\boldsymbol{X}$ and $\boldsymbol{Z}$ when projecting the columns into a latent space ($R^2 = 97.2\%$). The SO-PLS model performs similarly ($R^2 = 97.1\%$) and exhibits slightly higher Q^2 than Hybrid PLS, indicating less overfitting. This is likely due to the sequential selection of latent variables [7]. Additionally, SO-PLS offers an extra layer of interpretation by measuring the incremental contribution of the $\boldsymbol{Z}$ block.

Table 3. Calibration and validation metrics averaged over output states ($\boldsymbol{Y} = [\boldsymbol{C}_A, \boldsymbol{C}_B]$). The parameters of the FP model are assumed to be validated upon receival of the model, hence no new validation is performed.

	Calibration			Validation		
	FP	Hyb-PLS	SO-PLS	FP	Hyb-PLS	SO-PLS
R^2, Q^2 *(i)*	37.3%	97.2%	97.1%	-	94.5%	96.0%
RMSE *(i)*	0.74	0.15	0.16	-	0.23	0.21
R^2, Q^2 *(ii)*	60.5%	94.3%	94.4%	-	86.3%	88.9%
RMSE *(ii)*	0.57	0.27	0.27	-	0.37	0.35

In *Step 3*, the overall variance of $\hat{\boldsymbol{Y}}$ model is decomposed according to $\boldsymbol{X}$ and $\boldsymbol{Z}$. While similar precision could be achieved by concatenating $\boldsymbol{X}$ and $\boldsymbol{Z}$ into one block and fitting them against $\boldsymbol{Y}$ in a regular PLS model, the SO-PLS model allows for block-wise interpretations. As shown in Tbl. 4, the measured data improve the overall fit by 3%. The variance directly influenced by the mechanistic block is 44.9%, while 48.1% is indirectly influenced through the measured block, mainly due to the shared states $\boldsymbol{T}$ and $\boldsymbol{T}_c$. An additional feature of the SO-PLS model is the residual output block $\boldsymbol{Y}^{\mathrm{ort}}$, which encodes the information

not explained by $\boldsymbol{X}$. Fig. 3(a) shows that the model fails to capture some quadratic behavior, aligning with the structural error introduced in the first-principles model.

Table 4. Decomposition of overall explained variance (Q^2) with standard errors (based on bootstrapping).

	Direct	Indirect	Additional	Overall
Case *(i)*	44.9% (0.6%)	48.1% (0.4%)	3.0% (0.6%)	96.0% (0.5%)
Case *(ii)*	44.4% (1.5%)	37.8% (1.7%)	6.6% (2.2%)	88.9% (1.4%)

Case (ii): In *Step 1*, we use a calibrated first-principles process model structurally equivalent to (7), but with parameters $k_{i0}, E_i/R$, $i = 1,2,3$, perturbed randomly by up to 50%. The construction of $\boldsymbol{X}$, $\boldsymbol{Z}$, and $\boldsymbol{Y}$ remains the same, with the addition of state $\widetilde{\boldsymbol{C}}_D$ to $\boldsymbol{X}$.

In *Step 2*, optimal $\{6,1\}$ latent variables are chosen for $\boldsymbol{X}$ and $\boldsymbol{Z}$, respectively. Performance is compared to the first-principles model and a Hybrid PLS model fitted to $([\boldsymbol{X}, \boldsymbol{Z}], \boldsymbol{Y})$ with optimal 7 latent variables in the two last rows of Tbl. 3. Again, SO-PLS shows less overfitting, evident from the smaller difference between R^2 and Q^2.

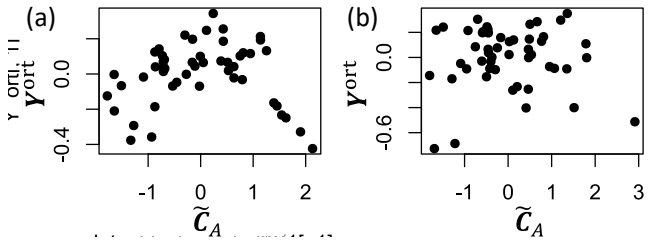

Figure 3. $\widetilde{\boldsymbol{C}}_A$ vs $\boldsymbol{Y}^{\text{ort}}$ for *Case (i)* (a) and *(ii)* (b).

In *Step 3*, as shown in Tbl. 4, measured data improve the overall fit by 6.6%. Compared to *Case (i)*, the mechanistic block explains less total variance (82.2%), despite the correct model formulation. The results show that measured data improve the model fit, but also highlight a limit to how much the first-principles model can be miscalibrated (given the lower overall R^2 at 88.9%). Furthermore, as expected, no structural errors are observed in the residuals (Fig. 3(b)).

CONCLUSION

In this work, we present a methodology based on SO-PLS for incorporating process knowledge into latent-variable models. This approach efficiently integrates first-principles models with measured process data without needing to recalibrate the first-principles model to the full-scale system. The methodology is demonstrated using a CSTR case example. The results show that it can decouple the variance explained by first-principles from that explained purely by data and capture mismatches.

Current work aims to apply this procedure in an industrial setting to identify root causes of disturbances in bioprocesses, using variance decomposition to pinpoint sources of abnormalities. Future work seeks to integrate process-wide interactions with first-principles knowledge into a unified SO-PLS model, extending previous research [7]. Increased complexity in the first-principles model can increase computational costs, necessitating the use of a surrogate model for generating synthetic data. Furthermore, the SO-PLS model's repeated deflation steps can be computationally demanding, especially in process-wide models with many blocks, due to the need to explore all latent variable combinations during cross-validation. The response-oriented sequential alternation (ROSA) model offers a viable alternative to address these challenges [15].

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support and data access from Novo Nordisk A/S.

REFERENCES

1. Destro F, Inguva PK, Srisuma P, Braatz RD. Advanced methodologies for model-based optimization and control of pharmaceutical processes. *Curr Opin Chem Eng* 45:101035 (2024) https://doi.org/10.1016/j.coche.2024.101035
2. Rathore AS, Garcia-Aponte OF, Golabgir A, ..., Herwig C. Role of knowledge management in development and lifecycle management of biopharmaceuticals. *Pharm Res* 34:243-256 (2017) https://doi.org/10.1007/s11095-016-2043-9
3. Içten E, Maloney AJ, Beaver MG, Shen DE, Zhu X, Graham LR, Robinson JA, Huggins S, Allian, A, Hart R, Walker SD, Braatz RD. A virtual plant for integrated continuous manufacturing of a carfilzomib drug substance intermediate, part 1: cdi-promoted amide bond formation. *Org Process Res Dev* 24:1861-1875 (2020) https://doi.org/10.1021/acs.oprd.0c00187
4. Zahel T, Hauer S, Mueller EM, Murphy P, Abad S, Vasilieva E, Maurer D, Brocard C, Reinisch D, Sagmeister P, Herwig C. Integrated process modeling-a process validation life cycle companion. *Bioengineering* 4:86 (2017) https://doi.org/10.3390/bioengineering4040086
5. Diab S, Christodoulou C, Taylor G, Rushworth P. Mathematical modeling and optimization to inform impurity control in an industrial active pharmaceutical ingredient manufacturing process. *Org Process Res Dev* 26:2864-2881 (2022) https://doi.org/10.1021/acs.oprd.2c00208
6. Pantelides CC, Renfro JG. The online use of first-principles models in process operations: Review, current status and future needs. *Comput Chem Eng*

51:136-148 (2013) https://doi.org/10.1016/j.compchemeng.2012.07.008
7. Zhu Q, Facco P, Zhao Z, Barolo M. Capturing connectivity information from process flow diagrams by sequential-orthogonalized PLS to improve soft-sensor performance. *Chemom Intell Lab Syst* 252:105192 (2024) https://doi.org/10.1016/j.chemolab.2024.105192
8. Cattaldo M, Ferrer A, Måge I. Dynamic multiblock regression for process modelling. *J Chemom* 38:e3618 (2024) https://doi.org/10.1002/cem.3618
9. Overgaard T, Bertran MO, Jorgensen JB, Nielsen BF. (in press). Identifying drivers of downstream yield variability using integrated process models: an application to api manufacturing. *IFAC Proc Vol* (2025)
10. Næs T, Romano R, Tomic O, Måge I, Smilde A, Liland KH. Sequential and orthogonalized PLS (SO-PLS) regression for path analysis: order of blocks and relations between effects. *J Chemom* 35:e3243 (2021) https://doi.org/10.1002/cem.3243
11. van de Vusse JG. Plug-flow type reactor versus tank reactor. *Chem Eng Sci* 19:994-996 (1964) https://doi.org/10.1016/0009-2509(64)85109-5
12. Höskuldsson A. PLS regression methods. *J Chemom* 2:211-228 (1988) https://doi.org/10.1002/cem.1180020306
13. Ghosh D, Mhaskar P, MacGregor JF. Hybrid partial least squares models for batch processes: Integrating data with process knowledge. *Ind Eng Chem Res* 60:9508-9520 (2021) https://doi.org/10.1021/acs.iecr.1c00865
14. von Stosch M, Oliveira R, Peres J, Feyo de Azevedo S. Hybrid semi-parametric modeling in process systems engineering: past, present and future. *Comput Chem Eng* 60:86-101 (2014) https://doi.org/10.1016/j.compchemeng.2013.08.008
15. Liland, KH, Næs, T, Indahl, UG. ROSA – a fast extension of partial least squares regression for multiblock data analysis. *J Chemom* 30: 651–662 (2016) https://doi.org/10.1002/cem.2824