Online Model Maintenance in Real-time Optimization Methods

José Matias[a], Johannes Jäschke[a]

[a]*Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway*

Abstract

The performance of model-based optimization methods, like Real-time Optimization (RTO), relies on the model accuracy and adequacy. However, features of the process may be unknown and/or the system behavior can drastically change with time (e.g. system degradation). Therefore, even if we have a perfect model in the beginning, we may end up making decisions based on a poor model. This paper proposes a method that adapts the model structure online, based on an available model set, while simultaneously estimates the model parameters. The problem is presented in a superstructure framework and solved using a mixed-integer nonlinear formulation. Then, the updated model is combined with Output Modifier Adaptation, an RTO variant, for economic optimization. Our method is tested in a continuous stirred-tank reactor and a gas lifted oil well network. The results show that we can select the correct model structure, update its parameters and, simultaneously, converge to the plant optimum.

Keywords: Model-based optimization, Modifier Adaptation, Model-plant mismatch, Model correction, Symbolic regression

1. Introduction

Process models are very useful whenever it is necessary to interact with a given system. They are fundamental for process design, process analysis and

*Corresponding author
Email address: johannes.jaschke@ntnu.no (Johannes Jäschke)

process operation [1]. In Real-time Optimization (RTO), we use a rigorous steady-state model to find the best operating strategy given an economic criterion [2]. Usually, RTO is implemented using the two-step (TS) approach [3], which is based on solving two sequential problems. First, the model parameters are adjusted in order to match the current plant conditions. Next, the updated model is used for computing the new optimal operating point.

The performance of a TS implementation relies heavily on the model, which needs to be accurate enough to predict the true plant optimum [4]. If this requirement is not met, the RTO decisions based on this model are not reliable and can lead to significant economic loss. Obtaining a good model can be challenging and expensive, especially if the inner workings of the process are partially unknown (e.g. in complex reaction networks), different system operation modes (e.g. different batch campaigns), and/or the system behavior changes drastically over time (e.g. system degradation) [2]. In these cases, even after adapting the model parameters, the model can fail to represent the actual plant behavior.

In this paper, we present an approach that allows the model structure to evolve in time while simultaneously determining the best model parameters. Our solution combines online model structure selection [5] and RTO using Output Modifier Adaptation (MAy) [6] as shown in Figure 1. MAy is a TS variant that guarantees convergence to the plant optimum even if the best available model is structurally and parametrically wrong. In our method, we propose several model structures for describing the process and select online the best model with updated parameters for representing the plant. In order to avoid chattering between model structures, as a consequence of noisy measurements for instance, we only perform the model structure selection step if a goodness of fit criterion is not met. Next, the updated model is used for optimizing the process operation via the classical MAy framework.

Note that MAy does not require model adaptation to converge to the plant optimum. However, updating the model to the current plant measurements is important for model maintenance and provides valuable information about

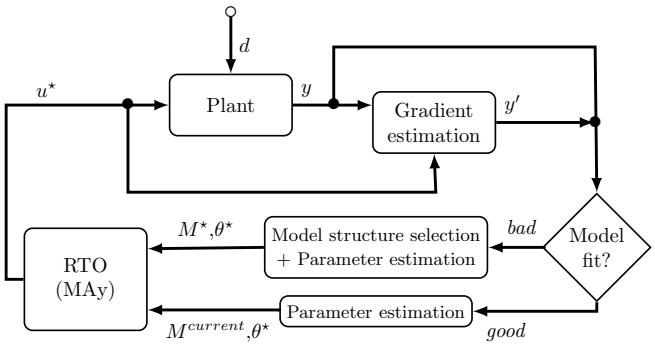


Figure 1: Schematic representation of the online model maintenance approach. The variable d represents disturbances that affect the process. y and y' are, respectively, the plant measurements and gradients, which are the derivative of the outputs with respect to the inputs u. $\theta^\star$ are the model parameters and $M^\star$ is the selected model structure, updated only if a goodness of fit criterion is not met. $u^\star$ is the output computed via MAy using the chosen model structure and the updated parameters.

the process, which can be used for other purposes, such as process condition monitoring. This paper fits well into this special issue for Prof. Engell because of his group important contributions to adapting the optimization problem based on measurements [7] and gradients [8].

Another advantage of our method appears when there are several candidate model structures available but no previous evidence for which one best represents the actual process. Using our method, one does not need to know the model structure *a priori* and can propose several model structures based on different hypothesis. Under the operation, the best model structure will be revealed, while at the same time the process is driven to the optimal operation point. Therefore, the model now can track different disturbances and "self-adapt" to more drastic changes in the process structure, e.g. equipment degradation, flow regime changes, among others.

The paper is organized as follows. First, we indicate how our method is related to previous studies in Section 2. Then, Section 3 introduces our method in detail, explaining its implementation with a toy example. In Section 4, we

formally define the problem and show the algorithm for its implementation. In Section 5, we discuss possible implementation issues. Next, Sections 6 and 7 give computational results of our method in the two case studies, a simple continuous stirred tank reactor (CSTR) reactor and a gas lift oil well network. Finally, Section 8 concludes the paper.

2. Related work

Previously, we proposed a 2-step online model maintenance method in the context of RTO [5]. First, we solved a model identification problem, in which several model structures were candidates for adequately describing the current behavior of the plant. Then, we used the updated model structure to optimize process operation with an an economic cost function. The main drawbacks of the method in [5] are: (1) there is no explicit mechanism to avoid chattering between consecutive iterations; and (2) the parameter values are not adapted, only the best model structure is selected. In this paper, we extend the previous approach to overcome them. We update the model parameters at every iteration, similarly to the TS approach, and include a goodness of fit test to check whether or not the model structure needs to be adapted.

If the discrepancy between the predictions and the measured values is acceptable, we do not update the model structure, only its parameters. This goodness of fit test avoids excessive switching between model structures, which may affect the method's convergence to the plant optimum. The parameter estimation is included because it increases the predictive capability of the individual structures and, as a consequence, of the whole system model. On the other hand, estimating the model parameters leads to a mixed-integer nonlinear optimization problem with issues regarding identifiability and non-unique parameters, which are discussed in Section 5.

Ahmad et al. [9] also apply a combination of modifier adaptation and parameter estimation. The goal of the authors is to enforce that the model is adequate in a modifier adaptation sense (i.e. it is locally convex in the vicinity

of the plant optimal inputs [6]), which guarantees convergence to the plant optimum. Since the model with adapted parameters provides better second order information, they also speed up the convergence. Instead of estimating the parameters, the same research group in Gao et al. [10] proposed a method where they switch between the rigorous model and local quadratic approximations of the plant. Here, the main idea is also to guarantee convergence of the MAy to the plant optimum since the quadratic is always adequate in a modifier adaptation sense. However, the cost of using the local approximations is the lower rate of convergence to the optimum due to the poor second order information [11]. Despite the fact that our method shares characteristics with the works above (i.e. we estimate the model parameters *and* switch between model structures to improve the fitting to the plant), our main goal is not limited to guaranteeing convergence to the plant optimum, which is achieved by restricting the available model set to modifier adaptation adequate models. Our main objective is to online maintain a good model by combining a "population of submodels" that is consistent with our current knowledge about the system.

Our method is also similar in spirit to the approach proposed by Hille and Budman [12], which modifies the estimation problem in order to simultaneously identify a proper structure for optimization and optimize the process. The authors define a model-update criterion online in order to estimate the model parameters for satisfying the conditions of optimality.

In a more general sense, the idea of simultaneously defining the functional form of the model and estimating its parameters (which is also known as symbolic regression, function identification or empirical design) has already been applied in chemical systems [13, 14] and control [15, 16]. The goal in these papers is also to increase the flexibility of the model in cases that the underlying phenomenon is not completely known. However, instead of using deterministic optimization approaches, the optimization problem is solved via Genetic Programming (GP). Despite being able to provide good solutions, GP does not provide any guarantees of local and global optimality. Recently, Cozad and Sahinidis [17] presented an MINLP formulation to solve the symbolic regression

problem. The paper uses a basic set of operators and operands (e.g. $+$, $\times$, $\exp(\cdot)$, $\sqrt{\cdot}$, etc.) instead of blocks of more complex structures as used in our paper. As such, we have the ability to include more *a priori* knowledge into our problem. Also, in our method, we use not only plant measurements but also plant gradients in the regression problem. As a consequence, we reduce problems with the non-uniqueness of model structures.

3. Model Maintenance using "modifiers" to quantify model accuracy in the Real-time Optimization context

RTO aims at optimizing a non-linear steady-state model based on an economic objective [2]. The process model plays a critical role in the standard RTO implementation. Traditionally, RTO models employ rigorous phenomenological relations to describe the process behavior. For a given model structure, the model parameters (or a subset of them) are adapted with new plant information obtained with sensors and analyzers [3]. The parameter adaptation step is carried out by solving an optimization problem, which minimizes the offset between the measurements and model prediction by changing the model parameters. However, this standard updating strategy may not suffice in cases of structural plant-model mismatch, where the RTO model is not adequate to find the true plant optimum even with updated parameters [18].

3.1. Alternatives for addressing plant-model mismatch

Instead of relying on model parameter adaptation, one could apply the iterative gradient-modification optimization (IGMO) scheme proposed by Gao and Engell [8]. Here, the model is corrected by adding terms for bias and gradient correction that shift the cost and constraint functions without adapting the model parameters. Consequently, the model locally matches the plant optimality conditions. Thus, the optimization routine is able to drive the operation to the plant optimum upon convergence without any constraint violation, even in presence of plant-model mismatch [8]. Marchetti et al. [6] also applied the bias

and gradient correction approach, which they labeled as "modifiers" and introduced the Modifier Adaptation (MA) method. In the same paper, the authors proposed an MA variant called Output Modifier Adaptation (MAy) that has the same convergence properties but modifies the model directly,

$$\mathbf{y}_{ad,k}(\mathbf{u}) := \mathbf{y}(\mathbf{u}) + \boldsymbol{\epsilon}_k + (\boldsymbol{\lambda}_k)^T(\mathbf{u} - \mathbf{u}_k). \tag{1}$$

Here, $\mathbf{y} : \mathcal{R}^{n_u} \to \mathcal{R}^{n_y}$ is the original steady-state model; $\mathbf{u} \in \mathcal{R}^{n_u}$ the model inputs; and k is the subscript representing the k^{th} MAy iteration. The adaptation terms $\boldsymbol{\epsilon}$ and $\boldsymbol{\lambda}$, as known as the modifiers, are calculated by

$$\boldsymbol{\epsilon}_k = \mathbf{y}_p(\mathbf{u}_k) - \mathbf{y}(\mathbf{u}_k), \qquad \boldsymbol{\lambda}_k = \Big(\mathbf{y}'_p(\mathbf{u}_k) - \mathbf{y}'(\mathbf{u}_k)\Big)^T, \tag{2}$$

where, $\mathbf{y}_p$ are the plant measurements and $\mathbf{y}'_p$ the plant gradients, which are the derivative of the outputs with respect to the inputs. They need to be estimated. The modifiers contain valuable information about the differences between the model and the plant. For example, $\boldsymbol{\epsilon}$ indicates the difference between the model predictions and plant measurements, and $\boldsymbol{\lambda}$ the difference between the model and plant gradients.

Hence, the modifiers can be considered a good model quality indicator in the context of RTO and can be also used for discriminating between different models. Matias and Jäschke [5] aggregated their information by introducing the "total model modifier", which is simply the sum of the Frobenius norm of the modifiers. In this paper, we introduce a weighted total modifier:

$$\psi := \frac{1}{2}\Big(\boldsymbol{\epsilon}_k^T \Sigma_\epsilon \boldsymbol{\epsilon}_k + \boldsymbol{\lambda}_k^T \Sigma_\lambda \boldsymbol{\lambda}_k\Big) \tag{3}$$

in which, Σ_ϵ and Σ_λ are weighting matrices of appropriate dimensions. The weights can be chosen simply as scaling factors or as the inverse of the covariance matrix of $\boldsymbol{\epsilon}$ and $\boldsymbol{\lambda}$ in the case that enough measurements are available to compute them. If the number of inputs is larger than 1, we stack the $\boldsymbol{\lambda}$ values for each input, i.e. $\boldsymbol{\lambda}_k = [\boldsymbol{\lambda}_{k,u_1}^T, \boldsymbol{\lambda}_{k,u_2}^T, \cdots, \boldsymbol{\lambda}_{k,u_{n_u}}^T]^T$, where n_u is the number of inputs. A small value of ψ indicates that the model matches the plant data well at the current operating point.

3.2. Online model maintenance approach

To illustrate the approach and how the total modifier ψ can be applied for comparing different models, we consider the problem of modeling and optimizing a reaction process (Figure 2). In this example, there are 3 possible hypotheses to describe the initially unknown reaction set. Therefore, we can propose three different models for representing the system.

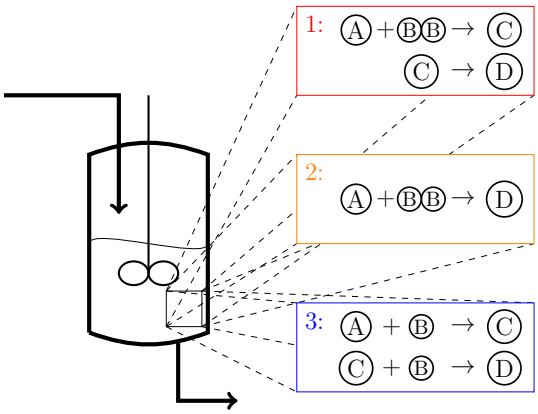


Figure 2: Illustration of the application of our method to a simple CSTR reactor with an unknown reaction set.

We initialize the method by choosing one of the model options. Then, when plant data come, in the form of measurements $y_{p,k}$ and gradients $y'_{p,k}$, we can check how well this information fits to the chosen model using the goodness of fit criterion. In Section 4.3, we describe this test in detail and present different options of how to perform it.

If the chosen model does not meet the criterion, we start the model refinement step. The idea here is to compare all available models in terms of difference between prediction and measurements (full dots) and model and plant gradients (arrows). Note that, even if the previously chosen model does not pass the test, we analyze it in this step because it still may be the best plant representation. As shown in Figure 3, we conclude that the best local plant approximation in this iteration is the model m_3 with the reaction set 3 because the model matches

both plant measurements and gradients best.

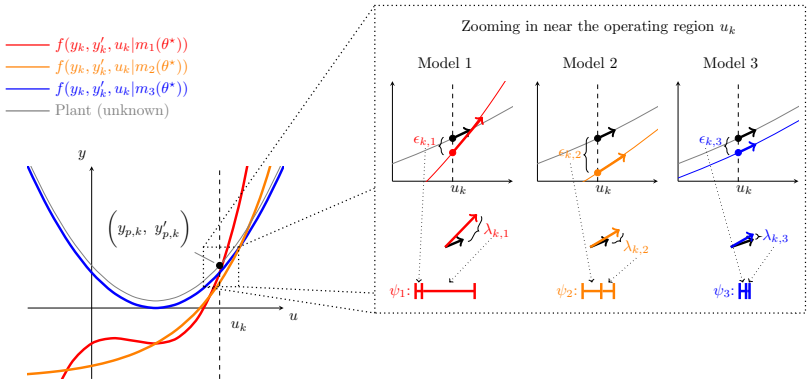


Figure 3: Comparing models response with plant information. We show how the total modifier ψ is computed and how it is used for choosing between different models. The modifier ϵ_k is the difference between model prediction and the plant measurements at u_k, and λ_k the difference between model and plant gradients evaluated at u_k. Based on the total modifier, we conclude that $\psi_3 < \psi_1 < \psi_2$. Thus, model 3 is the best approximation at u_k

Analyzing the example models at u_k in Figure 3, we see that the zero and first order modifiers have complementary information. For example, if we just analyze the model prediction capacity (ϵ - zero order modifier) at u_k, models 1 and 3 are locally very similar. However, when the first order information (λ) is also used for comparing the models, it is clear that model 3 describes the plant better than model 1 at u_k. Note that using ψ to differentiate between models is basically an extension of the least-squares criterion, where we add first-order information.

3.2.1. Using first-order information in ψ : numerical example

In a simple example, we show how the total modifier ψ improves model discrimination compared to the commonly used criterion, which we refer to as the least-square criterion:

$$\psi^{LS} := \frac{1}{2}\left((\mathbf{y}_p(\mathbf{u}) - \mathbf{y}(u))^T \Sigma_\epsilon (\mathbf{y}_p(\mathbf{u}) - \mathbf{y}(u))\right) = \frac{1}{2}\left(\boldsymbol{\epsilon}_k^T \Sigma_\epsilon \boldsymbol{\epsilon}_k\right) \tag{4}$$

Let us say we want to optimize the process represented by the following

model, i.e. $\phi = y_p$, where

$$y_p(u) = 40 - 40u + 10u^2 \tag{5}$$

The optimum is obtained when $u = 2$ and $y_p(u) = 0$. Assume that we have three candidate models to describe the process but we do not know which one best represents $y_p(u)$ beforehand. The three models have the same functional form $y(u) = \theta_1 + \theta_2 u + \theta_3 u^2$ but different parameter values:

$$\begin{aligned} \theta_1 &= [0.4, -0.4, 0.1] \\ \theta_2 &= [28, -28, 7] \\ \theta_3 &= [320, -320, 80] \end{aligned} \tag{6}$$

Clearly, the three models are different from $y_p(u)$. However, their optimum is equal to the plant optimum $u^\star_{\text{Model 1}} = u^\star_{\text{Model 2}} = u^\star_{\text{Model 3}} = u^\star_{\text{Plant}}$. We compare the three models with the "plant" in Figure 4.

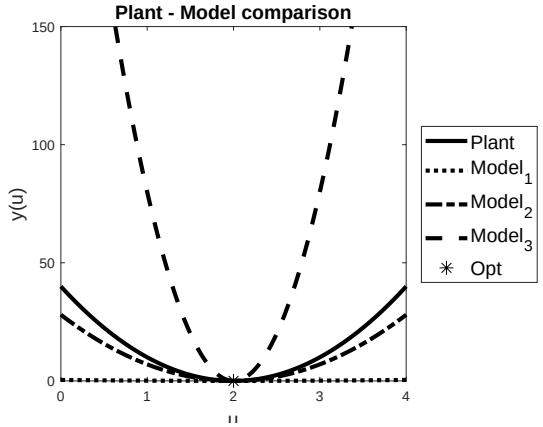
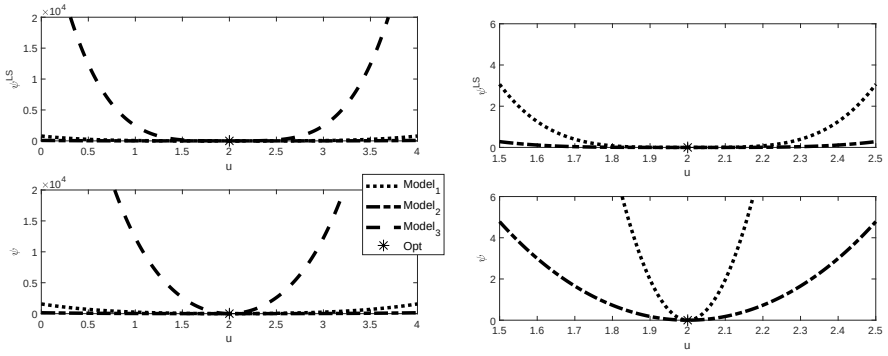


Figure 4: Comparison between "plant" and models. The three models are not perfect (plant-model mismatch), but they are able to predict the plant optimum correctly

Now, we use both ψ and ψ^{LS} to decide which model is the best for a given value of u. In Figure 5, we plot the both criteria values, using $\Sigma_\epsilon = \Sigma_\lambda = 1$, for all models in the region defined by $u \in [0, 4]$. By analyzing Figure 5a, we readily discard Model 3 using either ψ or ψ^{LS}, despite the fact that is relatively accurate near the optimum. Figure 5b zooms in close to the optimal point, however it



(a) All 3 models (b) Models 1 and 2 in the optimum vicinity

Figure 5: Comparing the value of the estimation objective function ψ^{LS} and ψ. By including the gradient information, it is easier to discriminate between the two models.

only takes into account Models 1 and 2. Considering both ψ and ψ^{LS}, we conclude that Model 1 is the best representation of the plant. However, ψ^{LS} is much more flat near the optimum and the difference between the values $\psi^{LS}_{\text{Model 1}}$ and $\psi^{LS}_{\text{Model 2}}$ is much smaller in this input range, which can be a problem if we have noisy measurements. On the other hand, ψ is a better indicator of the difference between between Model 1 and 2 due to the fact that we included extra information (gradients) in the analysis. In previous work [5], the outcomes of choosing a model with the total modifiers were compared with commonly used model selection criterion (R-square, Akaike information criterion (AIC) and the Bayes information criterion (BIC)) yielded similar results.

3.3. Potential benefits of including gradients in the estimation problem

We present a general approach to quantify the changes in the estimated parameters due to the addition of the gradients in the estimation problem. We assume that the gradient estimates are independent from the plant measurements used in the estimation problem. For example, if the gradients are estimated via Central Difference Approximation, we do not use the same measurements for estimating the parameters and for approximating the gradients. Consequently,

the plant measurements and gradients are statistically independent.

3.3.1. Using only plant measurements

First, we analyze the case in which only measurements are used for estimating the parameters. In this case, the objective function of the estimation problem is only the sum of squared residuals ψ^{LS} instead of the total modifier ψ. Given the inputs $\mathbf{u}$ and plant measurements $\mathbf{y}_p$, ψ^{LS} can be written as (note that we explicitly show the dependence on the parameters):

$$\psi^{LS}(\boldsymbol{\theta}) = \frac{1}{2}\left(\boldsymbol{\epsilon}(\boldsymbol{\theta})^T \ \Sigma_\epsilon \ \boldsymbol{\epsilon}(\boldsymbol{\theta})\right) \tag{7}$$

By differentiating ψ^{LS} w.r.t. θ and using the chain rule, we find that:

$$\frac{\partial \psi^{LS}}{\partial \boldsymbol{\theta}} = \frac{\partial \psi^{LS}}{\partial \boldsymbol{\epsilon}} \frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{\theta}} = -\frac{\partial \psi^{LS}}{\partial \boldsymbol{\epsilon}} \frac{\partial \mathbf{y}}{\partial \boldsymbol{\theta}} \tag{8}$$

If we evaluate the expression above at $\theta^{LS,\star}$, which is the optimal solution for the unconstrained estimation problem, we get:

$$\frac{\partial \psi^{LS}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^{LS,\star}) = -\frac{\partial \psi^{LS}}{\partial \boldsymbol{\epsilon}}(\boldsymbol{\theta}^{LS,\star}) \frac{\partial \mathbf{y}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^{LS,\star}) = \mathbf{0} \tag{9}$$

3.3.2. Using plant measurements and gradients

On the other hand, if we use the total modifier ψ as the objective function, it can be written as (same as Equation (3)):

$$\psi(\boldsymbol{\theta}) = \frac{1}{2}\left(\boldsymbol{\epsilon}(\boldsymbol{\theta})^T \ \Sigma_\epsilon \ \boldsymbol{\epsilon}(\boldsymbol{\theta}) + \boldsymbol{\lambda}(\boldsymbol{\theta})^T \ \Sigma_\lambda \ \boldsymbol{\lambda}(\boldsymbol{\theta})\right) \tag{10}$$

Note that:

$$\frac{\partial \psi^{LS}}{\partial \boldsymbol{\epsilon}} \frac{\partial \mathbf{y}}{\partial \boldsymbol{\theta}} = \frac{\partial \psi}{\partial \boldsymbol{\epsilon}} \frac{\partial \mathbf{y}}{\partial \boldsymbol{\theta}} \tag{11}$$

Then, we differentiate ψ w.r.t. θ and evaluate it at $\theta^{LS,\star}$, i.e. the parameters

estimated using ψ^{LS}. We assume that Σ_λ is not a function of θ and is symmetric:

$$\begin{aligned}
\frac{\partial \psi}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^{LS,\star}) &= -\frac{\partial \psi}{\partial \boldsymbol{\epsilon}}(\boldsymbol{\theta}^{LS,\star})\frac{\partial \mathbf{y}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^{LS,\star}) - \frac{\partial \psi}{\partial \boldsymbol{\lambda}}(\boldsymbol{\theta}^{LS,\star})\frac{\partial \mathbf{y}'}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^{LS,\star}) \\
\frac{\partial \psi}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^{LS,\star}) &= \cancelto{\mathbf{0}}{-\frac{\partial \psi}{\partial \boldsymbol{\epsilon}}(\boldsymbol{\theta}^{LS,\star})\frac{\partial \mathbf{y}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^{LS,\star})} - \frac{\partial \psi}{\partial \boldsymbol{\lambda}}(\boldsymbol{\theta}^{LS,\star})\frac{\partial \mathbf{y}'}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^{LS,\star}) \\
&= -\boldsymbol{\lambda}^T(\boldsymbol{\theta}^{LS,\star}) \cdot \Sigma_\lambda \cdot \frac{\partial \mathbf{y}'}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^{LS,\star}) \\
&= -\underbrace{(\mathbf{y}'_p - \mathbf{y}'(\boldsymbol{\theta}^{LS,\star}))^T}_{\text{T1}} \cdot \underbrace{\Sigma_\lambda}_{\text{T2}} \cdot \underbrace{\frac{\partial \mathbf{y}'}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^{LS,\star})}_{\text{T3}}
\end{aligned} \tag{12}$$

To assess the magnitude of change due to the inclusion of the gradients, we use the gradient of the estimation problem ψ evaluated at $\theta^{LS,\star}$. If its value deviates from $\mathbf{0}$, it means that the information added by the gradients in the problem have influence in the parameters estimates. Our conclusion on how the new data (gradients) change the results of the estimation depend on the three terms of the right-hand side of the last equation:

T1) If $\mathbf{y}'_p \approx \mathbf{y}'(\boldsymbol{\theta}^{LS,\star})$ (i.e. the gradients of the model updated only with the measurements $\mathbf{y}'(\boldsymbol{\theta}^{LS,\star}$ can described the plant gradients $\mathbf{y}'_p$), including gradient information does not contribute to the estimation problem.

T2) If we set $\Sigma_\lambda = \mathrm{Cov}(y_p, y_p)^{-1}$ (i.e. as the inverse of the covariance matrix of the gradient estimates), we weight the deviation between $\mathbf{y}'_p$ and $\mathbf{y}'(\boldsymbol{\theta}^{LS,\star})$ based on the "trust" we have in the measurement. Thus, if the estimation of the plant gradient in a given direction has a small variance, it has a large influence in ψ and, consequently, in the parameter estimates. On the other hand, gradients with large variance do not have a major influence in the estimation process. Note that, if we use an approximation of the covariance matrix, we would also take into account the correlation between the gradient directions in the estimation problem.

T3) The final term represents the model structure. If a set of parameters has no influence in the model gradients, it is clear that no information about that parameter can be by including the gradients. For instance, in a linear

model $y = \theta_0 + \theta_1 u$ (where $y' = \theta_1$), including the gradient measurement does not improve the estimation of θ_0.

This criterion ψ enables us to take into account both structural characteristics of the model ($T3$) and practical aspects ($T1$ and $T2$). Hence, by using $T1$, $T2$ and $T3$, we can systematically assess how the new information, related to the gradient, affects the parameter estimation.

4. Combining RTO and Model Selection

A summary of our method is given in Algorithm (1). It consists of three main tasks. First, we check if we need to refine the model structure (*Task 1*). If yes, we solve a simultaneous model structure/parameter adaptation problem. Otherwise, we only update the model parameters (*Task 2*). Next, we run an operation economic optimization with the updated model (*Task 3*). To facilitate the understanding, we first describe *Task 2* and *Task 3*. Then, we show *Task 1* in details.

4.1. Model structure selection (Task 2)

For representing the available model combinations, we use a block based superstructure representation proposed by Yeomans and Grossmann [19]. Basically, we divide our model in blocks and use a collection of sub-models for representing the modeling uncertainty of each block. For example, in the system shown in Figure 2, we set up a block in the model for the reaction set and we propose three sub-models for representing it. The assignment of the sub-models for different blocks, the determination of how they are connected and the parameter estimation are performed simultaneously by solving a mixed integer non linear problem (MINLP). In order to formulate the optimization problem, we used a generalized disjunctive programming approach [20].

First, we divide our model in n_b blocks $M_i,\ i = 1, \cdots, n_b$, leading to a superstructure model formulation represented by $Y := M_1(\theta_1) \cup M_2(\theta_2) \cup ... \cup$

Input : Set of possible models structures

Output: 1. Best model structure with updated parameters

2. New operating point

Data: Process measurements

begin

Estimate plant gradients;

Task *1*. *Goodness of fit test*

Check if the model structure needs to be adapted ;

Task *2*. *Model structure selection*

Select a combination of models (model structure) and update model parameters OR only update model parameters ;

Task *3*. *Optimizing operation*

Use Output Modifier Adaptation to compute new operating point;

Implement the new operating point in the plant;

end

Algorithm 1: Model Adaptation via Output Modifier Adaptation

$M_{n_b}(\theta_{n_b})$. Several sub-models can be used to describe a given block. For example, the b^{th} block is described by n_{M_b} different models $M_b(\theta_b) := \{m_{b,1}(\theta_{b,1}) \vee m_{b,2}(\theta_{b,2}) \vee \ldots \vee m_{b,n_{M_b}}(\theta_{b,n_{M_b}})\}$, where $\theta_b := \{\theta_{b,1} \vee \theta_{b,2} \vee \ldots \vee \theta_{b,n_{M_b}}\}$ represents the model parameters for block b[1]. The model structure selection and

[1]Note that the models may share some parameters in M_b, i.e. $\exists\ i, j \in [1, n_{M_b}]$ such that $\theta_{b,i} \cap \theta_{b,j} \neq \emptyset$.

parameter estimation problem becomes:

$$
\begin{aligned}
\mathbf{z}^{\star}_{k+1}, \mathbf{\Theta}^{\star}_{k+1} = \ & \underset{\mathbf{z},\mathbf{\Theta}}{\arg\min} \ \psi := \frac{1}{2}\left(\boldsymbol{\epsilon}_k^T \Sigma_\epsilon \boldsymbol{\epsilon}_k + \boldsymbol{\lambda}_k^T \Sigma_\lambda \boldsymbol{\lambda}_k\right) \\
& \text{s.t. } \boldsymbol{\epsilon}_k = \mathbf{y}_p(\mathbf{u}_k) - \mathbf{y}(\mathbf{u}_k | M_1(\boldsymbol{\theta}_1), \ldots, M_{n_b}(\boldsymbol{\theta}_{n_b})), \\
& \quad (\boldsymbol{\lambda}_k)^T = \mathbf{y}'_p(\mathbf{u}_k) - \mathbf{y}'(\mathbf{u}_k | M_1(\boldsymbol{\theta}_1), \ldots, M_{n_b}(\boldsymbol{\theta}_{n_b})) \\
& \quad \left[\bigvee_{j=1,\ldots,n_{M_1}} \begin{pmatrix} \text{if } z_{1,j} = 1 \\ M_1 := m_{1,j}(\mathbf{u}_k | \boldsymbol{\theta}_{1,j}) \end{pmatrix}\right] \wedge \ldots \wedge \left[\bigvee_{j=1,\ldots,n_{M_{n_b}}} \begin{pmatrix} \text{if } z_{n_b,j} = 1 \\ M_{n_b} := m_{n_b,j}(\mathbf{u}_k | \boldsymbol{\theta}_{n_b,j}) \end{pmatrix}\right] \\
& \quad \sum_{j=1}^{n_{M_b}} z_{b,j} = 1 \quad \text{where } b = 1, \ldots, n_b \quad \text{and } z_{b,j} \in \{0, 1\}
\end{aligned}
\tag{13}
$$

where,

$$
\begin{aligned}
\mathbf{z} &:= \left[[z_{1,1}, z_{1,2}, \ldots, z_{1,n_{M_1}}], \ldots, [z_{n_b,1}, z_{n_b,2}, \ldots, z_{n_b,n_{M_{n_b}}}]\right] \\
\boldsymbol{\theta}_i &:= [\theta^T_{i,1}, \ldots, \theta^T_{i,n_{M_i}}] \qquad i = 1, \ldots n_b \\
\mathbf{\Theta} &:= \mathbf{z} \odot [\boldsymbol{\theta}_1 \ldots \boldsymbol{\theta}_{n_b}]
\end{aligned}
$$

where, $\odot$ is the element-wise multiplication operator. The variables of Equation (13) were previously defined. Except for z_{bj}, which indicates that the j^{th} model is assigned (or not) to the block M_b, and $\mathbf{z}$, which is the is the vector of all z_{bj}. Using this formulation for the optimization problem, any mixed-integer nonlinear programming solver can be used to solve it. Upon convergence, the solution of Equation (13) will provide the model $M(\theta^\star)$ and parameter values $\theta^\star$ that best match the plant state, in a sense of minimizing ψ.

4.2. Optimizing operation (Task 3)

The updated model structure and parameters found from solving Equation (13), which are represented by $M^\star(\theta^\star)$, are used for computing the optimal operating point via Output Modifier Adaptation:

$$
\begin{aligned}
\mathbf{u}^{\star}_{k+1} = \ & \underset{\mathbf{u}}{\arg\min} \ \phi(\mathbf{u}, \mathbf{y}_{ad,k}(\mathbf{u} | M^{\star}_k(\theta^\star)) \\
& \text{s.t. } G := \{g_i(\mathbf{u}, \mathbf{y}_{ad,k}(\mathbf{u} | M^{\star}_k(\theta^\star))) \leq 0, \quad i = 1, \cdots, n_g\}
\end{aligned}
\tag{14}
$$

in which,

$$\mathbf{y}_{ad,k}(\mathbf{u}|M_k^\star(\theta^\star)) := \mathbf{y}(\mathbf{u}|M_k^\star(\theta^\star)) + \boldsymbol{\epsilon}_k + (\boldsymbol{\lambda}_k)^T(\mathbf{u} - \mathbf{u}_k)$$

and G is the set of all g_i constraints, with $i = 1, \cdots, n_g$. This set usually includes lower and upper bounds for measurements as well as operation and safety constraints.

Remark. *Note that the plant gradients $\boldsymbol{y}'_p(\boldsymbol{u}_k)$ need to be estimated to solve Equations* (13) *and* (14)*. Computing gradient estimates based on plant measurements can be challenging. There are several strategies available for estimating $\boldsymbol{y}'_p(\boldsymbol{u}_k)$ in the literature, for a detailed review refer to Marchetti et al. [21].*

4.3. Goodness of fit test (Task 1)

Before performing *Task 2* and *Task 3*, we run a goodness of fit test. When a set of new measurements $y_{p,k}$ and $y'_{p,k}$ is available, the test evaluates the prediction quality of the model structure obtained at the previous iteration $k-1$, namely $M^\star_{k-1}(\theta^\star_{k-1})$. For the test, we use the predicted (*a priori*) total modifier:

$$\psi_k^{ap} := \frac{1}{2}\left(\boldsymbol{\epsilon}^T_{k|k-1}\Sigma_\epsilon\boldsymbol{\epsilon}_{k|k-1} + \boldsymbol{\lambda}^T_{k|k-1}\Sigma_\lambda\boldsymbol{\lambda}_{k|k-1}\right) \tag{15}$$

where,

$$\boldsymbol{\epsilon}_{k|k-1} = \mathbf{y}_p(\mathbf{u}_k) - \mathbf{y}(\mathbf{u}_k|M^\star_{k-1}(\theta^\star_{k-1}))$$
$$(\boldsymbol{\lambda}_{k|k-1})^T = \mathbf{y}'_p(\mathbf{u}_k) - \mathbf{y}'(\mathbf{u}_k|M^\star_{k-1}(\theta^\star_{k-1}))$$

Since fluctuations in the plant information can affect this index, we filter the value of ψ_k^{ap} by a first order element, similarly to Hebing et al. [22]:

$$\psi_k^{ap,filt} = \frac{\psi_k^{ap} + \psi_{k-1}^{ap,filt}\Delta_{filt}}{\Delta_{filt} + 1} \tag{16}$$

in which, Δ_{filt} is a tuning parameter. Small values ($\to 0$) indicate a high level of confidence in the current measurements. After computing $\psi_k^{ap,filt}$, its value is compare to a goodness of fit threshold ψ^{gof}. If $\psi_k^{ap,filt} < \psi^{gof}$, we assume that the model structure $M^\star_{k-1}$ is able to satisfactorily represent the current plant state. Hence, we fix the model structure and use Equation 13 only to estimate

the model parameters. If $\psi_k^{ap,filt} > \psi^{gof}$, we updated the model structure and its parameters. As mentioned earlier, even if the model structure $M_{k-1}^{\star}$ does not pass the test, it is still included in the model structure adaptation problem, because it still may be the best structure available for representing the plant.

Note that the use of $\psi^{ap,filt}$ averages the decision of updating the model structure over information obtained in several iterations, instead of relying only on local measurements. As a consequence, there is a time-scale separation between the model structure adaptation and the operation optimization, which avoids model chattering. If the chosen model structure keeps switching between consecutive iterations, it may lead to noisy parameter estimates and affect the MAy convergence properties.

4.3.1. Alternatives goodness of fit tests

Other tests can be applied instead of the one previously presented. For example, the prediction accuracy criterion of Gao et al. [10] compares the capacity of the adapted model $M_{k-1}^{\star}(\theta_{k-1}^{\star})$ to represent the plant measurements in consecutive iterations:

$$\rho_k = \max \left\{ \left| 1 - \frac{y_{ad,i}(\mathbf{u}_k | M_{k-1}^{\star}(\theta_{k-1}^{\star})) - y_{ad,i}(\mathbf{u}_{k-1} | M_{k-1}^{\star}(\theta_{k-1}^{\star}))}{\mathbf{y}_{p,i}(\mathbf{u}_k) - \mathbf{y}_{p,i}(\mathbf{u}_{k-1})} \right| : i = 1, \ldots, n_y \right\} \tag{17}$$

By comparing the maximum deviation among all measurements to an allowable threshold, this model quality index can be used as goodness of fit indicator. A second option is to use a statistical test such as the Wald test [23], where we test the nonlinear hypothesis of $H_0 := \psi(M_{k-1}^{\star}(\theta_{k-1}^{\star})) = 0$ with a chosen statistical significance. One of the challenges with this alternative is the low number of samples to consistently estimate the parameter estimates variance matrix.

5. Handling Identification Issues

In this section, we briefly address issues related to the definition of the available model structures and the correspondent parameter vectors. We also discuss how these problems can be related to our method. In our case we are

not only trying to estimate parameters from a given structure but also trying to select the best structure in an available set. Thus, we have to take some precautions when choosing the model structures in the set and the parameters of which are to be estimated. Two structural properties are important when choosing the models in the set, distinguishability and identifiability [24].

5.1. Distinguishability

Since we have different model structures for representing specific phenomena and/or assumptions, it is possible that the parameter vectors associated with different model structures are not the same. Therefore, given a set of measurement $\mathbf{y}_p$ and estimated gradients $\mathbf{y}'_p$, it may not be possible to adjust the model parameters of a given model structure M_i to fit the plant information well. On the other hand, there may be a model structure M_j that can be adjusted to $\mathbf{y}_p$ and $\mathbf{y}'_p$.

In this case, we call models M_i and M_j as structurally output distinguishable (s.o.d.) [25], and we can eliminate structure M_i in favor of structure M_j. Formally, two models are s.o.d. if, for all feasible inputs and a given parametrization θ_i of M_i, there are no parameters θ_j of M_j such that $y(u|M_i(\theta_i)) = y(u|M_j(\theta_j))$.

By using the total modifier ψ instead of the least-square ψ^{LS}, the s.o.d. criterion requires not only $y(u|(M_i(\theta_i)) = y(u|M_j(\theta_j))$ (zeroth-order distinguishability) but also $y'(u; M_i, \theta_i) = y'(u; M_j, \theta_j)$. (*first order* distinguishability). Consequently, we relax the requirements for two models to be considered s.o.d., as illustrated in Section 3.2.1. Such characteristic is important because it is not possible to select one model in favor of the other, based only on input-output experiments, if the models are not s.o.d..

5.2. Identifiability

For all the model structures in the set, it is important to understand how reliably we can estimate its parameters θ from y_p and y'_p. We need to study the model *structural* properties to detect problems that may occur even before we collect data. The idea is to draw general conclusions about the chosen

parametrization, for example, if the planned measurements have enough information for estimating θ [24].

If one of the available structure combinations in the set leads to an unidentifiable model, the parameter estimates are prone to form an arbitrary set of estimates that can overfit the observation data [26]. As a consequence, we may choose the wrong model structure in favor of the best one [27]. Moreover, the estimated parameters of an unindentifiable model may have no physical meaning, which can be problematic if we are using the estimated model for different purposes, like condition monitoring.

We have to guarantee that all models in the available set have identifiable parameter sets. Various techniques for studying model identifiability are available in the literature (see. Miao et al. [26]). In the appendix, we show the method that we applied in the models of Case Study 2.

5.3. Identifiability and distinguishability in our method

Since in our method we do not choose the next plant operating point neither to discriminate most efficiently between two models nor to improve the parameter estimation, it is important to guarantee that the individual models have identifiable parameter sets and are distinguishable in the operating range. Otherwise, the parameter estimates are not only imprecise but also highly correlated, with a high variance [28]. Additionally, model chattering may occur between two sequential iterations if the structures are not distinguishable [5].

6. Case Study 1: Tank Reactor

To illustrate that our approach can drive the process to the true plant optimum while identifying the best model, we use a modified version of the tank reactor case study shown in Matias and Jäschke [5]. The main difference here is that the plant behavior changes with time. This case study shows that, by adding parameter estimation to the algorithm, we do not affect the capacity of the method to find the best model structure while optimizing the system operation. Even in the case that the "true" model structure is not included within the

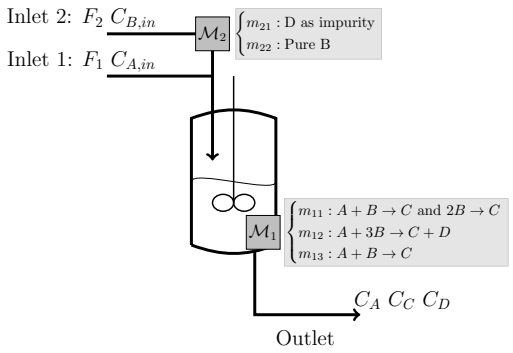


Figure 6: CSTR flowsheet. The system has only one manipulated variable, the flowrate of inlet 1, F_1, which contains pure A. The outlet concentrations C_A, C_C, and C_D are measured as well as the flowrate of inlet 2, F_2. The highlighted blocks $\mathcal{M}_1$ and $\mathcal{M}_2$ indicate the model uncertainty related to the reactor set and the concentration of D in inlet 2, respectively.

set of candidate models, which is the most likely situation to occur in practice. Clearly, the parameter estimation step may lead to overfitting and/or identifiability issues. Such issues are discussed in the context of the next case study, in Section 7.

The process is shown in Figure 6. The objective is to maximize the concentration of the product C at the reactor outlet. It is known that C is formed from the reaction of A and B but the complete reaction set is assumed to be unknown. The system has two inlets 1 and 2. The first one contains pure A with concentration $C_{A,in}$ and its flowrate F_1 can be manipulated freely. The second one comes from an upstream process with fixed flowrate F_2 and concentration $C_{B,in}$. There is also a fourth component D that is either a byproduct of the reactions or comes as a residual in inlet 2. Therefore, there are three possible settings for the process:

1. Inlet 2 contains pure B. The reaction is first-order in A and B, and there is a side reaction that generates D by consuming B:
   $\mathrm{A} + \mathrm{B} \xrightarrow{k_1} \mathrm{C}$ and $2\,\mathrm{B} \xrightarrow{k_2} \mathrm{D}$

2. Inlet 2 contains pure B and there is only one reaction, first-order in A and

third-order in B, that generates both C and D:
$\mathrm{A} + 3\,\mathrm{B} \xrightarrow{k_3} \mathrm{C} + \mathrm{D}$

3. Only the first reaction of Option (1) takes place and D enters the system as an impurity in inlet 2:
$\mathrm{A} + \mathrm{B} \xrightarrow{k_1} \mathrm{C}$

Given these three options, we summarize the model using two blocks $\mathcal{M}_1$ and $\mathcal{M}_2$, which are indicated in Figure 6. We further apply some standard assumptions: perfect mixing, perfect level control, and isothermal operation. The resulting steady-state model is:

$$\underline{\mathbf{0}} = \begin{bmatrix} \frac{F_1 C_{A,in}}{V} \\ \frac{F_2 C_{B,in}}{V} \\ 0 \\ \boxed{\mathcal{M}_2} \end{bmatrix} - \frac{(F_1 + F_2)}{V} \begin{bmatrix} C_A \\ C_B \\ C_C \\ C_D \end{bmatrix} + \boxed{\mathcal{M}_1} \tag{18}$$

where,
$\mathcal{M}_1 :=$

$$\left\{ \left(m_{11} : \begin{bmatrix} -1 & 0 \\ -1 & -2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} k_1 C_A C_B \\ k_2 C_B^2 \end{bmatrix} \right) \bigvee \left(m_{12} : \begin{bmatrix} -1 \\ -3 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} k_3 C_A C_B^3 \end{bmatrix} \right) \bigvee \left(m_{13} : \begin{bmatrix} -1 \\ -1 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} k_1 C_A C_B \end{bmatrix} \right) \right\} \tag{19}$$

in which, $m : \ldots$ indicates that the sub-model m can be defined by the equations $[\ldots]$. k_1, k_2 and k_3 are the rate constants of the chemical reactions and we need to estimate their values. For $\mathcal{M}_2$:

$$\mathcal{M}_2 := \left\{ \left(m_{21} : 0 \right) \bigvee \left(m_{22} : \frac{(F_1 + F_2) C_D}{V} \right) \right\} \tag{20}$$

According to the three possible process settings, we then define three model options $M_1 = \{m_{11}, m_{21}\}$, $M_2 = \{m_{12}, m_{21}\}$, and $M_3 = \{m_{13}, m_{22}\}$ to fit the problem to the framework of Equation (13). Also, we define the parameter set to be estimated for the model structures as $\boldsymbol{\theta}_1 = [k_1, k_2]^T$, $\boldsymbol{\theta}_2 = [k_3]^T$, and $\boldsymbol{\theta}_3 = [k_1]^T$.

6.1. Simulation set-up

First, we define the "real" process. We consider that inlet 2 contains only B. Regarding the reaction set, we consider that, at the beginning of the simulation, there are two reactions, a first order reaction in A and B that generates C and a side reaction, second order in B, that generates D. Then, for illustrative purposes, we consider that the "true" reaction set start to change linearly to $\mathrm{A} + 3\,\mathrm{B} \xrightarrow{k_3} \mathrm{C} + \mathrm{D}$, which can be represented by $\mathcal{M}_1 = \alpha\ M_1 + (1-\alpha)M_2$, where $\alpha \in [0, 1]$. Hence, M_1 is the "true" model at the beginning ($\alpha = 1$) of the simulation and M_2 at the end ($\alpha = 0$). Despite hypothetical, this disturbance can represent mixing problems inside the reactor and also catalyst deactivation changing the reaction pathway. For the parameter set, we assume that the "true" parameters of M_1 are $k_1 = 0.75$ [L/(mol min)] and $k_2 = 1.5$ [L/(mol min)], while the parameter of M_2 is $k_3 = 1.5$ [$\mathrm{L}^3/(\mathrm{mol}^3$ min)].

For estimating the parameters, we assume that only C_B, C_C and C_D are measured in the outlet stream ($\mathbf{y}_p = [C_B, C_C, C_D]^T$). To further simplify the problem, we consider perfect measurements (no noise). In turn, we assume that the gradients are not directly measured. We use plant experiments with Forward Finite Differencing (FFD), as in Marchetti et al. [21]. with a step size of Δh. The economic optimization objective is to maximize the production of C and there are no operation constraints (thus, $\phi := C_C$ and $G := \emptyset$ in Equation (14)). The degree of freedom for the operation is assumed to be the flowrate of pure A, F_1. The initial condition and the model parameters are summarized in Table 1.

We use an input filter, $\mathbf{u}_{k+1} = \mathbf{u}_k + K_u(\mathbf{u}^\star_{k+1} - \mathbf{u}_k)$, to include some robustness due to the possible presence of noise measurements and gradient estimation error in the modifier computation. Values of K_u near 1 are chosen if measurement noise and the gradient estimation error are small. A consolidated method for choosing the filter values remains an open question. However, Papasavvas and François [29] proposed an interesting method for automatic selection of the filter gain that preserves MAy's convergence properties while minimizing conservatism. Here, since the case study is relatively simple, the filter gain value is chosen by a trial-and-error process. Regarding the total modifier weights, we

chose the values of Σ_ϵ and Σ_λ as scaling factors.

The proposed method is simulated in the MATLAB R2019b programming environment (Mathworks Inc., Natick, MA, USA) using the CasADi v3.4.5 extension [30] for algorithmic differentiation. The NLP problem is solved using IPOPT version 3.12.2 [31] and the MINLP problem is solved using BARON [32].

Table 1: Initial condition and parameters for the CSTR simulation. We also show the weights of the total modifier (*diag* indicates a diagonal matrix). The two last values are the perturbation used in the Forward Finite Difference approach and the input filter parameter.

Description	Symbol	Value	Unit
Plant initial condition	$x_0 = \begin{bmatrix} C_{A,0} \\ C_{B,0} \\ C_{C,0} \\ C_{A,0} \end{bmatrix}$	$\begin{bmatrix} 0.7385 \\ 0.0231 \\ 0.4922 \\ 0.0308 \end{bmatrix}$	[mol/L]
F_1 initial flow rate	$F_{1,0}$	8	[L/min]
Feed concentration of A	$C_{A,in}$	2	[mol/L]
Feed concentration of B	$C_{B,in}$	1.5	[mol/L]
F_2 flow rate	F_2	5	[L/min]
Reactor volume	V	500	[L]
ϵ weight in ψ	Σ_ϵ	$diag(1,1,1)$	[-]
λ weight in ψ	Σ_λ	$0.1 diag(1,1,1)$	[-]
Goodness of fit threshold	ψ^{gof}	0.01	[-]
Perturbation step size	Δh	0.01	[mol/L]
Input filter	K_u	0.75	[-]

6.2. Results

The goal of the case study is to show that we can use our method to converge to the true plant optimum while identifying the best model. We start by discussing the convergence to the optimal operating point. Next, we show the behavior of the total modifier and its use as a plant-model mismatch indicator.

Then, we show the connection between the total modifier behavior with the model choice and the goodness of fit test. Finally, we conclude the case study analysis by analyzing the parameter estimation results.

In the simulations, we do not consider the dynamic behavior of the system, only the steady-state is analyzed. The simulations were carried out for 20 steady-state periods. For the first 5 iterations, the plant is represented by M_1, then the plant behavior changes linearly, until the 15^{th} iteration, where M_3 becomes the plant model. Therefore, between iterations 5 and 15, there is no "true" model in the available set. The simulation starts from a suboptimal operational condition, $u_0 := F_{1,0} = 4$ [L/min]. The initial model guess for the model structure is M_3 (i.e. D as impurity and C is generated by the first order reaction $\mathrm{A} + \mathrm{B} \xrightarrow{k_1} \mathrm{C}$). The initial guess for k_1 is 1 [L/(mol min)].

6.2.1. Convergence to plant optimum

The economic objective is to maximize the production of C by adjusting the input F_1, the feed stream that contains the reactant A. Since this is a simulation study and we know the "plant" optimum, we can check if the method is able to converge to it.

In Figure 7, we show the concentration of C as a function of the input F_A and the steady-state period. We plot the input trajectories calculated using Equation (14) (blue) as well as the chosen model at the corresponding iteration. To benchmark our method, we compare this trajectory with the input sequence computed via MAy using M_1. We see that, even though we start with a wrong model M_3, our method chooses the best model structure while driving the system to its optimum. The method chooses the correct model at the first iteration that M_3 does not pass in the goodness of fit test (iteration 3) (see Figure 9). When the plant structure begins to change, both input sequences deviate from the true optimum due to plant-model mismatch, even though we are optimization via modifier adaptation methods in both cases. This happens because modifier adaptation methods are able to reach the true plant optimum only after the convergence of the modifiers ϵ and λ [6]. After iteration 15, our

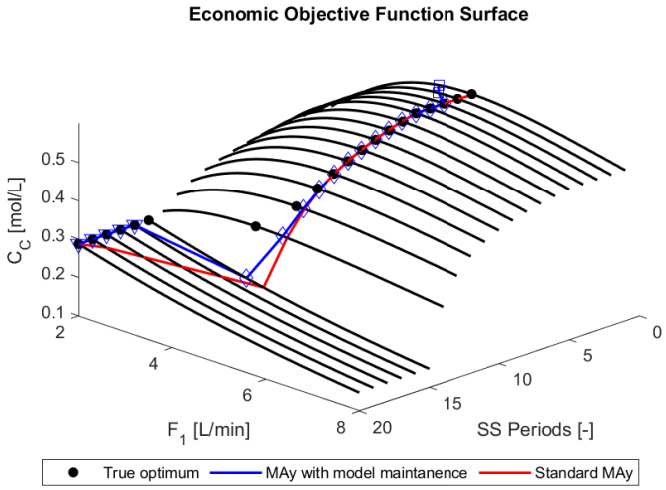


Figure 7: Optimum Tracking. The input trajectories F_1 computed by Equation (14) (blue) and by the standard MAy (red) are compared along the simulation periods. We also show the model structure chosen in the iteration ($M_1 = \Diamond$, $M_2 = \bigtriangledown$ and $M_3 = \square$) and the true optimum. Note that plant behavior changes between iterations 5 and 15. In the beginning of the simulation, M_1 is the true model. Then, the plant behavior changes between iterations 5 and 15. After this period, M_2 is the true model.

method readily identifies the correct model structure and reaches the plant optimum, while MAy takes a few iterations to drive the system to its optimum (until the modifiers converge to their true value).

Therefore, we see that estimating the model structure and its parameters is not necessary for MAy to reach the plant optimum. MAy is conceptually designed to assure the convergence even in cases where the model $\mathbf{y}(\mathbf{u})$ used in the economic optimization (Equation (14)) is structurally and parametrically wrong. The only requirement is that the model follows some mild adequacy criteria [6]. However, by refining the model structure and updating its parameters, we can potentially improve the optimization performance as seen in this case study. On the other hand, if model chattering happens between consecutive iterations, it can affect the MAy convergence, limiting the advantages of our

method. That is the reason that we included a goodness of fit test to trigger the model structure selection. The test criterion is based on a filtered version of the total modifier, which is used here as a model quality indicator.

6.2.2. Total modifier as model quality indicator

In Figure 8, we show the results of the total modifier ψ for the 3 model options, M_1, M_2, and M_3. The total modifier ψ in our method is used as an indicator of the plant-model mismatch. Note that instead of using just the model predictions residual $(y_p - y)$, we also add the deviation in the gradient prediction $(y'_p - y')$. Hence, ψ is influenced by both terms, which means that we want to match not only the plant measurements but also the plant first order information. This roughly means that we want our model not only to show precisely where we are, but also where are we going.

On the left-hand side of Figure 8, we compare the values of ψ at every iteration for the three models. On the right hand side, we show the composition of the total modifier for each individual model. As shown in Equation (3), the total modifier (full line) is the summation of $0.5\ \boldsymbol{\epsilon}^T\ \Sigma_\epsilon\ \boldsymbol{\epsilon}$ (dotted line) and $0.5\ \boldsymbol{\lambda}^T\ \Sigma_\lambda\ \boldsymbol{\lambda}$ (dashed line).

The results of Figure 8 show us why the total modifier can be used as model quality indicator. We can conclude that M_1 is the "true" model in the beginning of the simulations because the total modifier is equal to zero. The same case applies to M_2 after iteration 15. Moreover, in the end of the simulation, we see that the modifiers for all models converge to their stationary point when the plant behavior ceases to change (as expected from the standard MAy optimization results in Figure 7). When we analyze the contribution of the first order terms λ to ψ, we see that, although its magnitude changes during the simulation, we could have reached the same conclusion without this information. The main reason is that the models assumptions are very different in M_1, M_2 and M_3. However, since we are applying the method in the MAy-RTO context, having good first order information is important to optimize the system using the modifier adaptation scheme. In the next case study, the gradient-related

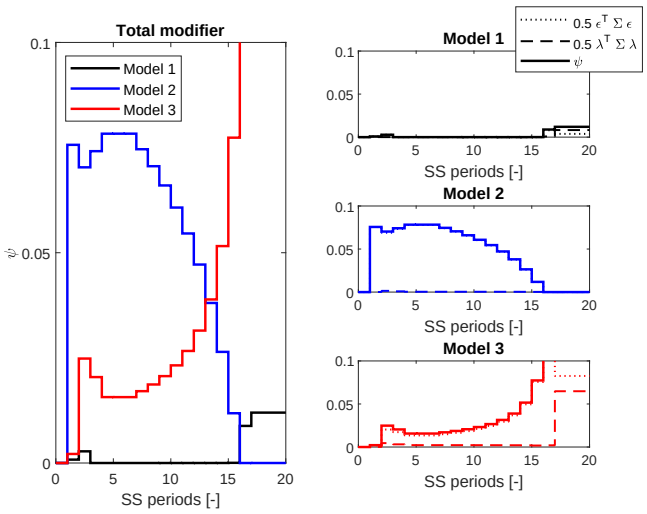


Figure 8: The total modifier for the 3 models is compared during the simulation on the left-hand side plot. On the right-hand side, the values of the total modifier are plotted individually as well as the contribution of the zeroth (ϵ) and first (λ) order modifiers, which compose the value of ψ according to Equation (3)

term of ψ plays an important role in the model selection step of our method.

6.2.3. Model Structure Selection and goodness of fit criterion

After computing the total modifier at the current information, we can calculate the goodness of fit criterion using Equation (16). In Figure 9, we show how the model choice (top plot) is triggered by the value of $\psi_k^{ap,filt}$ (second plot from top to bottom). Whenever $\psi_k^{ap,filt}$ is bigger than the threshold ψ^{gof}, which is represented by the dotted line, we update the model structure.

As shown in Figure 9, the filter is initialized at the first iteration. At iteration 2, $\psi^{ap,filt}$ is updated and already indicates that the model structure needs to be refined. Then, we update the model structure to M_1, which meets the goodness of fit criterion until iteration 15 when we update the model structure again. By using $\psi^{ap,filt}$ instead of ψ, we consider information from several iterations to trigger the model structure refinement step. The main advantage of this step is to avoid switching back and forth between two or more model candidates

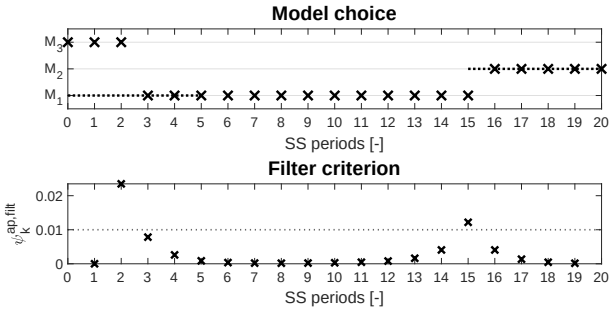


Figure 9: The top plot shows the model choice at each iteration. The dashed line corresponds to the true model at a given iteration. In the second plot, we show $\psi^{ap,filt}$, computed using Equation (16), and the threshold value for the test.

in consecutive iterations, which can affect the performance of the operation optimization.

6.2.4. Parameter Estimation

Next, Figure 10 shows the parameters estimates using ψ as criterion. We start with Model 3 and $k_1 = 1$ [mol/L]. We compare the value of the parameter estimates (markers) with the parameter value (dotted line). Since M_1 and M_3 share the parameter k_1, we used different colors to indicate when the parameter of M_1 is estimated and when the parameter of M_3 is estimated. We used the same color code as in Figure 8, i.e. black - M_1, blue - M_2 and red - M_3.

By analyzing Figure 10, we see that we can estimate the model parameters without any deviations from their "plant" counterparts when we use the correct model structure. When there is no true model (between iterations 5 and 15), we can estimate the parameters of the model structure currentyly being used. Even though the simulation setting is ideal (without noise and the model set contains the true model in some of the iterations), this case study shows the capacity of the total modifier to be used as a model performance indicator, and illustrates important concepts used in our method.

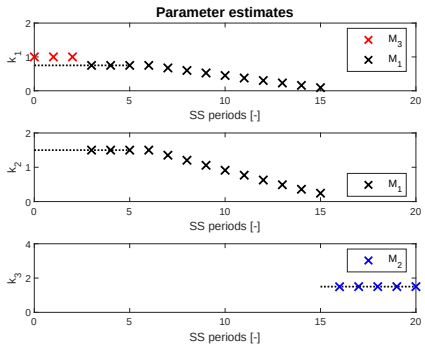


Figure 10: Results for the simultaneous model structure and parameter estimation according to Equations (13). Each model structure has its own estimable parameter set ($\boldsymbol{\theta}_1 = [k_1, k_2]^T$, $\boldsymbol{\theta}_2 = [k_3]^T$, and $\boldsymbol{\theta}_3 = [k_1]^T$). Note that both M_1 and M_3 contain k_1, therefore its value is estimated whenever one of these two structures is chosen.

7. Case Study 2: Gas lift oil well network

In the second case study, we implement our method in a more complex process, a gas lift oil well network to illustrate some issues that are to be expected in a more realistic setting including noise and poor model information.

7.1. Process description

To extract oil from a subsea reservoir, a well is drilled into the seabed. The reservoir natural pressure is responsible for lifting the fluid, which usually contains a mixture of oil, gas, sand and water, to the topside facilities. There, the oil is treated (handling the water that is mixed with the oil, for example) and stored. If the reservoir pressure is not high enough to transport the fluids, artificial lifting methods need to be applied. Among the possibilities, artificial gas lifting is frequently used [33]. The idea is to inject gas close to the bottom of the well in order to reduce the fluid mixture density. Consequently, the hydrostatic pressure loss is reduced and the production from the reservoir increases. On the other hand, by adding more gas into the pipeline, the frictional pressure drop increases, which may reduce the well production, such that an optimal gas injection that maximizes oil production must be found.

One possibility to deal with this trade-off is to determine the gas lift flowrate using RTO. However, since RTO is a model-based optimization, the challenge shifts to capture accurately not only the multi-phase flow inside the pipelines but also the reservoir behavior in a model. Moreover, even if we have a good model in the beginning of the production, reservoir characteristics change with time leading to plant-model mismatch. The problem becomes more challenging when dealing with a network of wells.

We propose to optimize the operation of a two-well network shown in Figure 11. The main goal is to maximize the oil extraction of the reservoir while considering the processing capacity constraints on the gas lift flowrates and on the maximum gas processing at the topside facilities. The manipulated variables of the system are the gas lift flowrate of each well.

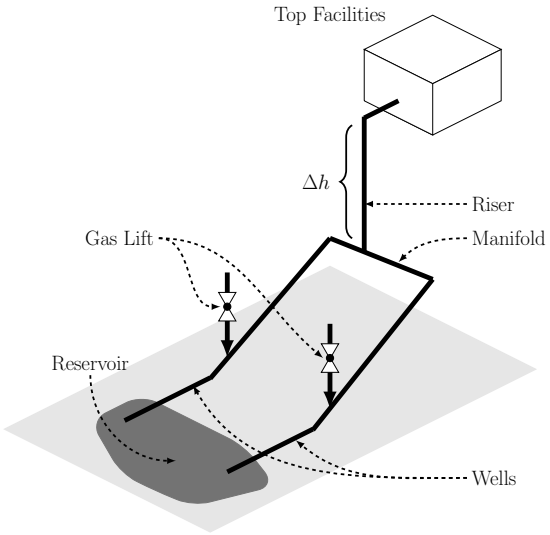


Figure 11: Network containing two gas lifted wells. A manifold connects the two wells to the riser, a pipeline that transports the gas/oil mixture from the seabed to the top facilities. The system has two inputs that can be manipulated, the gas-lift flowrate of each well. The pressures along the wells and riser, as well as the total oil and gas production are measured at the top facilities.

7.2. Model

In order to obtain the model of the gas lift oil well network, we make some simplifying assumptions: constant temperatures along the wells, ideal gas behavior, and a simple linear relations to calculate the reservoir outlet flows. The developed model is based on Krishnamoorthy et al. [34]. The modeling uncertainty here is included as different assumptions regarding pressure loss due to friction along the riser (M_1) and the wells (M_2), as well as the reservoir model (M_3):

1. Pressure loss due to friction along the riser/manifold:

$$M_1 := \left\{ \left(m_{1,1} : \begin{bmatrix} \text{pressure drop} \\ \text{is negligible} \end{bmatrix} \right) \bigvee \left(m_{1,2} : \begin{bmatrix} \text{pressure drop calculated by} \\ \text{Darcy-Weisbach equation} \end{bmatrix} \right) \right\}$$

2. Pressure loss due to friction along the along the wells:

$$M_2 := \left\{ \left(m_{2,1} : \begin{bmatrix} \text{pressure drop} \\ \text{is negligible} \end{bmatrix} \right) \bigvee \left(m_{2,2} : \begin{bmatrix} \text{pressure drop calculated by} \\ \text{Darcy-Weisbach equation} \end{bmatrix} \right) \right\}$$

3. Reservoir liquid outflow model. The reservoir is represented by a linear equation that relates oil outlet, w_{ro}, with the pressure difference between well bottom hole, p_{bh}, and reservoir, p_r. The reservoir model can be written as $w_{ro,i} = PI(i)(p_r - p_{bh,i})$ where $i = 1, 2$ (one model for each well). For each sub-model, different values of PI are used:

$$M_3 := \left\{ \left(m_{3,1} : PI = \begin{bmatrix} 5 & 5 \end{bmatrix}^T \right) \bigvee \left(m_{3,2} : PI = \begin{bmatrix} 7 & 7 \end{bmatrix}^T \right) \right\}$$

The resulting model contains around 40 equations and it is not shown here for the sake of brevity. Please refer to Krishnamoorthy et al. [34] for the complete model equations. Figure 12 summarizes the model and the modeling uncertainty. Using 3 blocks, we end up with $2^3 = 8$ different combinations in the available model set. This exponential increase in the number of options shows the importance of elaborating the model structure decision as an optimization problem like in Equation (13), especially when including more candidate submodels.

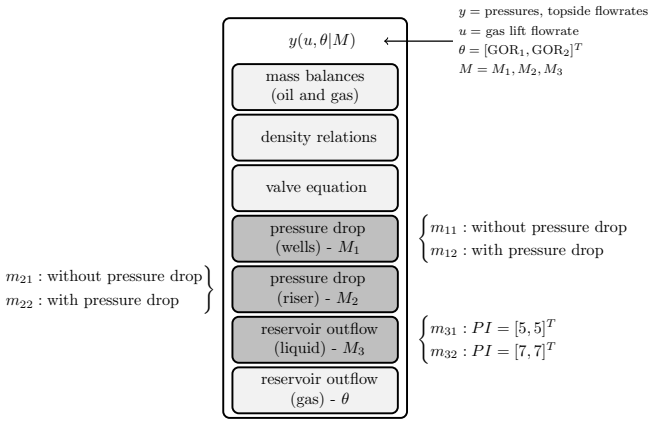


Figure 12: Overview of the steady-state model. The uncertain model blocks are highlighted in dark gray.

In Figure 12, we also show the model parameters that need to be estimated, gas oil ratio GOR of the wells, $\theta = [\mathrm{GOR}_1, \mathrm{GOR}_2]^T$. This parameter determines the ratio between the oil, w_{ro}, and gas, w_{rg}, outflows of the reservoir (i.e. $\mathrm{GOR} = \frac{\text{gas flowrate}}{\text{oil flowrate}}$). Note that, in this case study, the parameters are not directly included in the blocks M_1, M_2 and M_3, but they are part of a common set of equations for all models. The choice of the estimable parameter set is important to avoid identifiability issues (i.e. guaranteeing that the planned measurements contain enough information for reliably estimating the model parameters). An identifiability analysis of the available models is given in the Appendix.

7.3. Simulation set-up

In the simulation, we consider that a model represents the plant as in Case Study 1. However, the plant model behavior changes as a function of the gas lift flowrate, as shown in Figure 13, instead of time. In Region A (dark gray), both riser and wells have pressure loss due to friction, whereas in Region B (light gray) frictional pressure drop is not significant (i.e. only the liquid column hydrostatic pressure affects the difference between the top and bottom pressures). In order

to simulate this behavior, we use two different models to represent the plant and the transition between them is calculated by a 2-D sigmoidal function as in Matias and Jäschke [5]. This transition strategy leads to a third region, which we called intermediary. This region is also shown in Figure 13 using an intermediary gray color. Using this "plant" configuration, we cannot predict the plant behavior in both regions with only one model. Plant-model mismatch is inherent to this system if we change regions during operation and use a single model. Therefore, we want to use our method to optimize the system while we track the changes in the plant behavior.

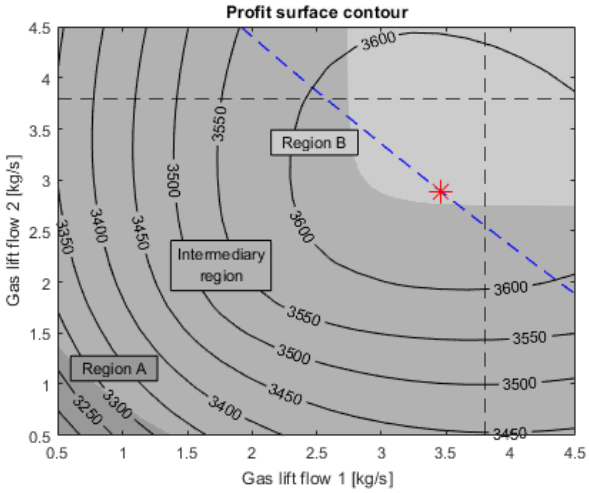


Figure 13: Profit as a function of the inputs. We indicate the plant optimum, the constraints on the maximum gas processing at the topside facilities (blue dashed line), and the maximum gas lift flowrate in each well (black dashed line). In the contour plot, Region A is indicated by the dark gray region and Region B by the light gray. The transition between the two regions is also indicated.

The economic optimization problem aims at maximizing the profit, which we computed as the difference between the squared oil production revenue and the squared cost related to compression of the gas for artificial lifting. We also consider constraints on the processing capacity (g_1) and on the availability of gas lift (g_2 and g_3). For the 2 well network, the optimization problem is written

as:

$$\begin{aligned} \max_{\mathbf{u}=[w_{gl,1},w_{gl,2}]^T} \quad & \phi := \alpha_o w_{oTot}^2 - \alpha_{gl} \sum_{i=1}^{2} w_{gl,i}^2 \\ \text{s. t.} \quad & G := \left\{ \begin{array}{l} g_1 : w_{gTot} - w_{gM} \leq 0 \\ g_2 : w_{gl,1} - w_{glM} \leq 0 \\ g_3 : w_{gl,2} - w_{glM} \leq 0 \end{array} \right\} \end{aligned} \tag{21}$$

where, w_{oTot} and w_{gTot} are the well network oil and gas production; $w_{gl,i}$ is the gas lift flow rate of well i; w_{glM} is the maximum gas lift flowrate; and w_{gM} is the maximum gas processing capacity of the upstream processes. α_o and α_{gl} are the price of oil and the cost associated with the gas compression, respectively. We specify $\alpha_0 = 1$ and $\alpha_{gl} = 0.5$. Note that the total oil and gas production of the well are computed by the the input-output mapping, $[w_{oTot}, w_{gTot}]^T = \mathbf{y}(w_{gl,1}, w_{gl,2})$ shown in Figure 12, which is adapted by the Output Modifier Adaptation method (i.e. $\mathbf{y} \mapsto \mathbf{y}_{ad}$) like in Equation (1).

The pressures in the system and the total oil and gas production at topside are the system measurements. To make the case study more realistic, we add noise to the plant model outputs $y_{noise} = y_{model} + r$, where r is drawn from a standard normal distribution with mean zero and the variance adapted according to the current value of the measurement (1% of the current value for the pressures and 0.5% for the flows). The noise r is neither correlated in time nor between measurements.

The case study optimization and simulation are carried out using the same solvers and software as in the previous case study. The gradients are not directly measured, they are estimated via plant experiments with Central Finite Differencing (CFD) using a perturbation step size $\Delta h = 0.05$ for both inputs. This method is similar to Forward Finite Differences (FFD) used in the first case study and perform better in presence of noise, for more details refer to Brekelmans et al. [35]. For the total modifier (Equation (3)), we set $\Sigma_\epsilon = I(n_y)$ and $\Sigma_\lambda = 0.01I(2n_y)$, where I is the identity matrix and the value in parenthesis indicate its dimension. These values were chosen based on the order of magnitude of the zero and first order modifiers. We set the goodness of fit threshold ψ^{gof}

as 0.9, also based on the total modifier order of magnitude. The input filter gain in MAy optimization is chosen as $K_u = 0.25$ after some exploratory experiments. The goal was to guarantee a smooth operation while preserving the ability of MAy to converge only at the true plant optimal. Despite conservative, this filter value avoids excessive input changes due to noise measurements and gradient estimation errors, making the iterations safer. We could have applied filters to the modifiers (ϵ and λ) instead, but input filtering has been shown to be more effective [29]. The initial condition and the values of the nominal model parameters are not shown here for the sake of brevity. They can be found in Matias and Jäschke [5].

7.4. Results

The results are shown in the same order as in the previous case study. First, we discuss the scheme's ability to converge to the optimal plant operational point. Then, we show the model structure selection, goodness of fit, and parameter estimation results.

In the simulations, we run the system for 50 steady-state periods. The initial point is at $w_{gl} = [0.5, 0.5]^T$ [kg/s], which is in Region A, and the plant optimum is in Region B. Therefore, in this case study, the method needs to track both the optimum and changes in the plant behavior. In the figures, we use different markers for each one of the 8 available model structures discussed previously. The correspondence between markers and models is shown in Table 2. To avoid ambiguities, we refer to the whole system model (combination of all blocks) as complete model (CM).

7.4.1. Economic Optimization of plant operation

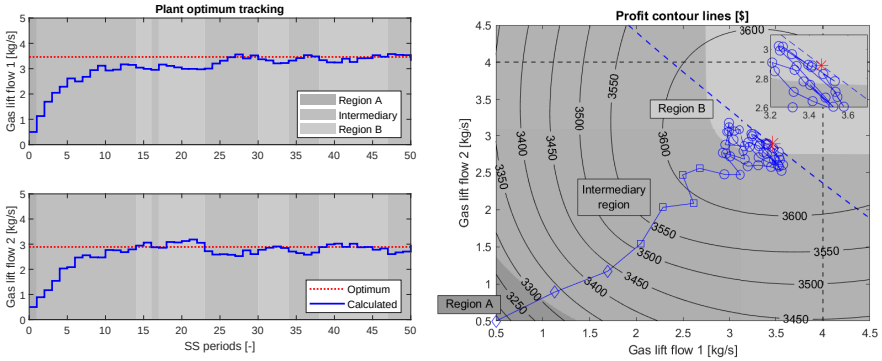
The results are shown in Figure 14. On the left-hand side, we plot the calculated inputs by Equation (14) (blue) and the plant optimum (red dotted line). Also, we show the region of the current steady-state period. We use the same color correspondence as in Figure 13, where the dark gray shows that we are in Region A, the light gray in Region B. The intermediary region is also

Table 2: Marker and model correspondence. Complete model 8 (CM_8) is the correct model for plant region A and complete model 2 (CM_2) for region B

Symbol	Model	M_1 - Well frictional ΔP	M_2 - Riser frictional ΔP	M_3 - Reservoir parameters (PI)
$\triangleleft$	$CM_1 \rightarrow \{m_{11}, m_{21}, m_{32}\}$	without	without	$[7; 7]$
$\bigcirc$	$CM_2 \rightarrow \{m_{11}, m_{21}, m_{31}\}$	without	without	$[5; 5]$
$\bigstar$	$CM_3 \rightarrow \{m_{11}, m_{22}, m_{32}\}$	without	with	$[7; 7]$
$\triangledown$	$CM_4 \rightarrow \{m_{11}, m_{22}, m_{31}\}$	without	with	$[5; 5]$
$\triangle$	$CM_5 \rightarrow \{m_{12}, m_{21}, m_{32}\}$	with	without	$[7; 7]$
$\square$	$CM_6 \rightarrow \{m_{12}, m_{21}, m_{31}\}$	with	without	$[5; 5]$
$\triangleright$	$CM_7 \rightarrow \{m_{12}, m_{22}, m_{32}\}$	with	with	$[7; 7]$
$\lozenge$	$CM_8 \rightarrow \{m_{12}, m_{22}, m_{31}\}$	with	with	$[5; 5]$

indicated. On the right-hand side, we plot the same information, the computed inputs, but now we plot them on the plant profit contour surface. In this plot, we also indicate the different plant regions as well as the optimization problem constraints. The dashed blue line is the maximum gas production constraint and dashed black lines are the maximum gas lift flow rate for each well. Additionally, in the right-hand side figure, we use markers to indicate the chosen model structure in the correspondent steady-state period. Note that $\lozenge$ is the correct model for Region A and $\bigcirc$ for Region B. In this plot, we do not show the probing plants to compute the plant gradients via CFD.

The input profile in Figure 14a shows that the method is able to track the plant optimum despite the change in the plant behavior and the presence of noise. The profile approaches the optimum smoothly, as a direct consequence of the small value of the input filter $K_u = 0.25$. Additionally, the operating constraints are always satisfied. The reasons are the convergence of the model structure estimation problem and the constraint adaptation using the modifiers ϵ and λ. The modifiers correct the eventual plant-model mismatch in case of deviation between the estimated parameters and their plant counterparts as well as in cases when the wrong model structure is selected.



(a) Optimum tracking

(b) Plant profit contour surface and optimal input sequence $w_{gl}^{\star}$.

Figure 14: Optimum Tracking. We start the simulation with the system at a suboptimal operational condition $w_{gl} = [0.5, 0.5]^T$ in Region A. Then, we optimize the system (Equation (21)) for 50 steady-states periods. The value of the inputs is compared to the optimal inputs (left) and plotted on the plant profit surface (right) to show how we track the optimum while changing the operating Regions from A to B. We plot the optimization problem constraints on the on the Figure (b), dashed blue line - maximum gas production constraint, and dashed black lines - maximum gas lift flow rate. We also include an inset zooming in near the plant optimum to show that the constraints are not violated.

In Figure 14a, we do not compare the results with the standard MAy input trajectory. Because, in this case study, these trajectories are almost overlapped. This shows an important advantage of the method. We can explore and rely on the nice convergence properties of the MAy while identifying the model structure, increasing our knowledge about the real process.

7.4.2. Model Structure Selection

In Figure 15, we show the model structure selected during the SS periods by Equation (13) (top) and the value of the goodness of fit test (bottom). In the top plot, we use the marker-model correspondence of Table 2 but the markers have two different colors. When the method chooses the correct model for the region, we use the *blue* marker and the *white* marker otherwise. To explicitly show the correspondence between the SS period and the plant region, we used

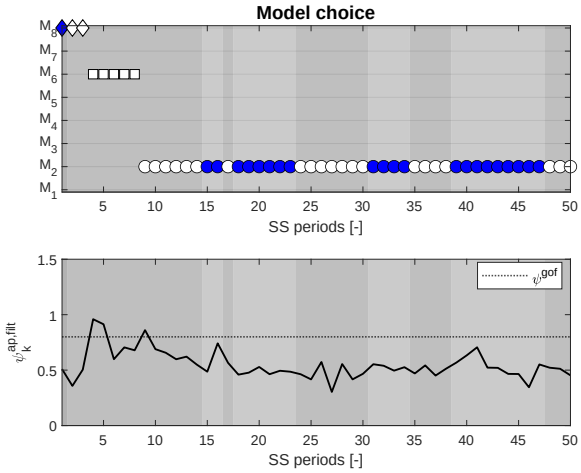


Figure 15: Results for the model structure selection (Equations (13)). Each marker corresponds to one of the 8 available model structure. The correspondence is shown in Table 2. The markers have two colors, white and blue. Blue indicates that the correct model structure is chosen in the SS period. The plant regions and SS period correspondence is shown on the background. In order to facilitate visualization, we also plotted the model markers in three different levels according to the plant region. In the bottom plot, we show the evolution of the goodness of fit criterion $\psi_k^{ap,filt}$ and its threshold ψ^{gof}. If $\psi_k^{ap,filt} > \psi^{gof}$, we trigger the model structure selection.

the gray color map of Figure 14. Also, we plotted the markers in different levels according to the plant region to facilitate visualization.

In the bottom plot, we show the value of $\psi_k^{ap,filt}$ during the simulation. We compare it against the threshold ψ^{gof}. If $\psi_k^{ap,filt} < \psi^{gof}$, we assume that the model structure is able to represent the plant well enough. Hence, we do not need to refine the model structure, only update its parameters.

Figure 15 shows that the method is able to choose the correct structure in Regions A and B. Additionally, it selects a model in the intermediary region even when there is no "correct" model available. Note that, after iteration 15, we switch between the intermediary region and Region B several times. In this situation, the use of $\psi_k^{ap,filt}$ to trigger the model structure refinement is specially interesting. Here, the filter "averages" this decision and avoids

chattering between consecutive iterations.

In the bottom plot, we can see that the the goodness of fit criterion is not met in two consecutive iterations (4 and 5). In this case, we continue estimating the model structure. However, it may happen that none of the models is a good enough representation and the best model keeps failing at the test consecutively. If this occurs, a policy to systematically increase/decrease the threshold ψ^{gof} needs to be created.

These results present a significant improvement when compared to the ones in Matias and Jäschke [5]. Since the authors used an unweighted version of the total modifier, the effect of the wrong model assumption m_{12} could not be identified in every iteration inside Region B. Depending on the noise realization and/or on the gradient estimation error, the method was not able to distinguish between models CM_6 ($\square$) and CM_2 ($\bigcirc$). Even though, in their case study, the model parameters were fixed and CM_2 contained the "true" parameter values.

On the other hand, here, we weight the total modifier using Σ_ϵ and Σ_λ. As a consequence, we scale the error between different measurements and also in the gradient estimation, which improves the model selection step. The disadvantage is that we need to tune more parameters to implement our method. Note that the gradients estimation uncertainty plays a major role in this step. If it is too large when compared to the measurement prediction residual ϵ, it may dominate ψ leading to a model structure selection based only on approximation error, which is completely undesirable. However, if it is properly scaled, as shown in $T2$ in Equation (12), the inclusion λ can improve model structure distinguishability. Moreover, since the gradient uncertainty is additive and bounded [21], the approximation error is "shared" by all structures.

7.4.3. Total modifier as model quality indicator

We can see the contribution of both ϵ and λ in ψ in Figure (16). It shows the composition of the total modifier for the individual model structures, where the total modifier (full line) is the summation of $0.5\epsilon\Sigma_\epsilon\epsilon$ (dotted line) and $0.5\lambda\Sigma_\lambda\lambda$ (dashed line). Instead of presenting the value for all model structures, we show

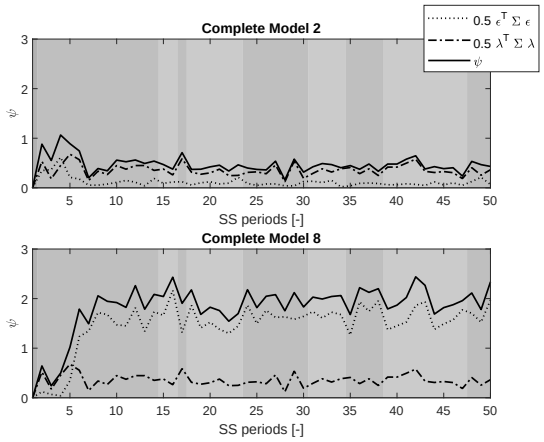


Figure 16: The total modifier composition for CM_2 and CM_8 is compared during the simulation. The values of the total modifier (full line) are plotted along with the contribution of the zeroth-order ($0.5\epsilon\Sigma_\epsilon\epsilon$ - dotted line) and first-order ($0.5\lambda\Sigma_\lambda\lambda$ - dashed line) modifiers. The plant regions are also indicated with the same color scheme as in the previous figures.

only for CM_8 and CM_2, the best model for Region A and B, respectively.

We see that both terms contribute to the total modifier, which shows that they provide complementary information to the model selection algorithm. Including the gradients in ψ can be seen simply as the addition of new observations (either correlated or independent, depending on the gradient estimation approach) in order to improve the model structure selection and parameter estimation. The main advantage is that these new observations are specifically tailored to improve the proper identification of the system curvature and not taken with a different purpose (e.g. optimization).

It is important to note that including λ in ψ is not mandatory. If the competing model structures are distinghishable by considering only zeroth-order information, the method works similarly if we consider only the ϵ related term in ψ, see Case Study 1, for instance.

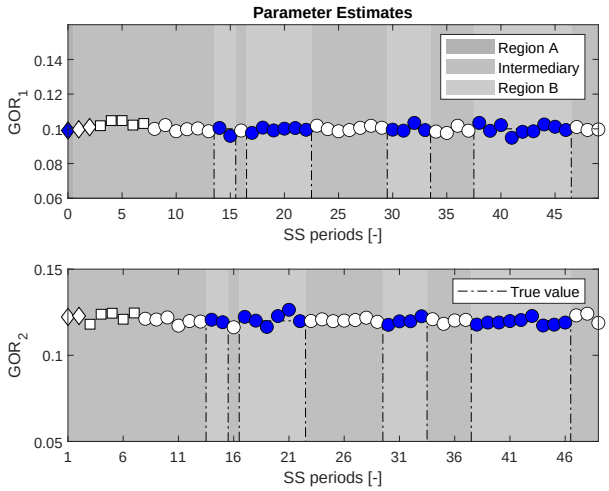


Figure 17: Results for the simultaneous model structure and parameter estimation according to Equations (13). All the model structures share the same estimable parameter set $\boldsymbol{\theta} = [\mathrm{GOR}_1, \mathrm{GOR}_2]^T$.

7.4.4. Parameter Estimation

Finally, the estimated parameters are show in Figure 17. The actual value is plotted as a dashed black line, except in the intermediary region, where we do not have true parameter values. The markers indicate the value of the estimated parameters and the chosen model structure for a given SS period. The marker correspondence is the same as in the previous figures. Once again, if the correct model is chosen, the marker is shown in blue. We also plot the corresponding plant region.

Note that, before estimating the parameters online, we need to perform an identifiability analysis on the models (shown in the Appendix). Hence, the estimation results are not sensitive to perturbations in the problem data and, consequently, we avoid problems with overparametrization and ill-conditioned parameter estimation problems.

Given that we have a reasonable estimable parameter set, it is expected that the parameter estimation problem converges to the true parameter values. We see that, even if we choose the incorrect model structure, CM_1 for example,

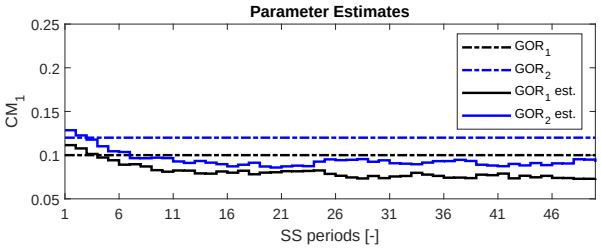


Figure 18: Results for the estimation of $\boldsymbol{\theta} = [\mathrm{GOR}_1, \mathrm{GOR}_2]^T$, given that we have an incorrect model structure, CM_1 in this case. Since we performed an identifiability analisys, discussed in Appendix B, the parameters can be uniquely identified despite the fact that we have plant-model mismatch. Clearly, they converge to the incorrect values because we are not able to represent the underlying process with CM_1.

we are able to uniquely identify a set of parameters as shown in Figure 18. This highlights the importance of performing an identifiability analysis before implementing our method. If this previous step is set aside, one of the models could overfit the measurements and be chosen by the method even if it is not the best representation of the plant.

8. Conclusion

We presented a method that adapts the model structure and parameters online while optimizing the process. The method is implemented using the framework depicted in Figure 1 and latter detailed in Algorithm (1). Basically, it can be represented by a two-level structure, where: 1. the zeroth order modifier, $\epsilon = y_p(u_k) - y(u_k)$, and first order modifier, $\lambda = (y_p'(u_k) - y'(u_k))^T$ are used for solving an identification problem, in which the total modifier ψ, Equation (3), is minimized. The goal is to select the best model with updated parameters in the model set; and 2. a more classical Output Modifier Adaptation (MAy) with the updated model is performed.

Given that several model structures are available for adequately describing the plant behavior, the total modifier ψ indicates not only the difference between plant measurement and model prediction ϵ, which is traditionally considered in

identification problems, but also the difference between plant and model gradients λ. By including first order information about the differences between the plant and model, we avoid issues such like non-uniqueness of model structures as shown in Section 3.2.1.

Before solving the structure identification problem, we perform a goodness of fit test. If the current structure is a good representation of the plant, we do not update the model structure, only its parameters. The goal is to avoid chattering between consecutive iterations, which may hinder the MAy convergence to the optimal operating point.

The method was tested in two case studies, a CSTR and a gas lift well network (Sections 6 and 7, respectively). In our simulation studies, the method converges reasonably fast to the plant optimum and updates the model structure and parameters accurately. We also show how it behaves in cases that there is no "true" model. The method, then, chooses the model structure combination with updated parameters that minimized the criterion ψ. Note that, before running our method online, we need to perform a sensitivity analysis on our models in order to rule out problems with parameters identifiability.

Our method is somewhat structurally similar to the standard two-step RTO approach. The main differences are: (i) the model structure can change and, (ii) by combining the model identification with Output Modifier Adaptation, we can deal with plant model mismatch provided that our model is adequate. Hence, even if we do not have the "true" model in our model set, we are able to reach the plant optimum via MAy. The method presented is useful when there is uncertainty in the modeling step or in cases where the real process change with time, like when an equipment is degrading. However, to apply our method, we must have information about the plant gradients, which can be a limitation. Additionally, probing the plant in a way that enables us to select the best model can be useful, for example choosing a sampling strategy that increases our knowledge about a particular model (or set of models). A future research interest is to find an optimal sampling plan that allies feasible region exploration, to get better model updates, and model exploitation (fast

convergence to the real plant optimum).

9. APPENDIX: Identifiability analysis of the models used in Case Study 2

The appendix is divided in two parts. First, we introduce a method for model parameter identifiability checking used by Yang et al. [36]. Next, we show how we applied it to the oil well network model of Case Study 2. The method is used for avoiding problems with overparametrization and ill-conditioned parameter estimation problems.

9.1. Model parameter identifiability checking

There are different methods for testing if the model structure is identifiable (see, e.g. Walter and Pronzato [24], Miao et al. [26], and Yang et al. [36]). Here we give a brief overview of the method presented in Yang et al. [36]. The idea is to find parameters that cannot be estimated separately and group them in clusters. For example, let us say that our system model is $y = \theta_1\theta_2 u$. Both parameters have similar effect on the model output. If we try to estimate both at the same time, we have infinite solutions to the data-fitting problem. However, if we cluster them in a unique parameter $\bar{\theta}$, its value can be easily determined.

For more complicated models, it is hard to determine this similarity between parameters. One alternative is to use the parameters output sensitivity vector s_i, which is defined as $s_i = \frac{\partial y}{\partial \theta_i}$. We use the cosine distance between the vectors s_i and s_j to analyze the colinearity between the pair of parameters i and j:

$$d_{ij} = 1 - |\cos \omega_{ij}| = 1 - \frac{|s_i \cdot s_j|}{\|s_i\| \, \|s_j\|} \tag{22}$$

If the cosine distance is lower than a threshold τ_θ, the parameters cannot be estimated separately in practice, and are clustered (or one of the parameter is set to some nominal value). By sequentially checking the pairwise cosine distance of all available parameters combinations, we obtain an estimable parameter set that can be accurately updated with the plant information.

9.2. Parameter subsection selection applied to Case Study 2

Table 3 shows the parameters of the gas lift oil well model, showed in Section 7.2, that can be adapted.

Table 3: Parameters of the gas lift oil well network

	Parameter	Description
θ_1	GOR	Reservoir gas-oil ratio
θ_2	p_{res}	Reservoir pressure
θ_3	ρ_{oil}	Density of oil in the reservoir
θ_4	μ_{oil}	Viscosity of oil in the reservoir

In order to choose the estimable parameter set, we use the following procedure based on Yang et al. [36]:

1. We generate 100 data points by randomly probing the plant in the feasible input space (Regions A, B and intermediary region shown in Figure 13, using the same noise range as in the Case Study 2);

2. For all models, we estimate the parameters θ_i where $i = 1, \cdots, 4$ in Table 3. Then, we compute the correspondent s_i for each model;

3. Next, we average s_i over all data points. At this stage, each model structure has one average value of s_i, with $i = 1, \cdots, 4$ (i.e. for each parameter in Table 3);

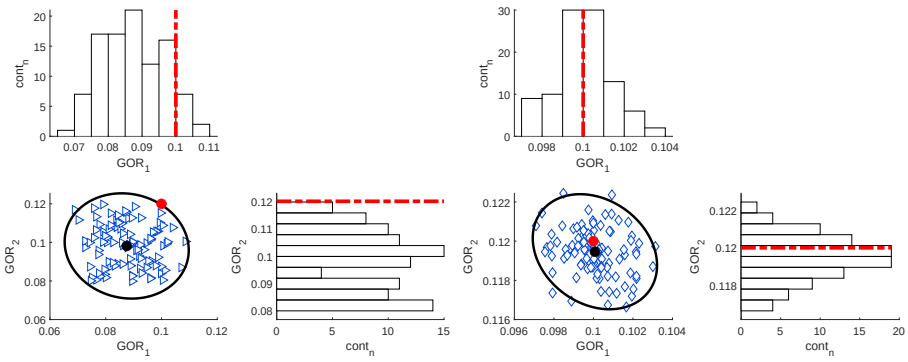
4. For each model, we compute d_{ij} for all the possible parameter combinations and set $\tau_\theta = 0.7$. If $d_{ij} < \tau_\theta$, we set one of the parameters to its nominal value.

By applying Step 4 sequentially, we obtain the estimable parameter set as:

$$\boldsymbol{\theta} = \begin{bmatrix} \text{GOR}_1 & \text{GOR}_2 \end{bmatrix}^T \tag{23}$$

In Figure 19, we show the estimation results for the 100 probing points for CM_7 and CM_8 (Model 8 is the "true" model in Region A and model 7 has the

incorrect structure in every region in the feasible space). The estimate mean is shown as a black dot, and the confidence interval with $\alpha = 0.95$ is shown as the black ellipsoid. Histograms containing of the individual parameters estimates are also plotted. Finally, the actual value of the parameter is shown in red.



(a) CM_7 (b) CM_8

Figure 19: Parameter estimation for 100 samples. The 95% confidence interval as well as the mean (as a black dot) are shown. Histograms containing of the individual parameters estimates are also plotted. The actual value of the parameter is shown in red.

Since we do not know *a priori* which structure is correct, we do not analyze the bias or the variance of the estimation. Instead, we focus on the fact that for all models the parameters show a low correlation between them, which indicates that they, in fact, bring complementary information to the estimation process and can be estimated separately.

References

[1] D. Bonvin, C. Georgakis, C. Pantelides, M. Barolo, M. Grover, D. Rodrigues, R. Schneider, D. Dochain, Linking models and experiments, Industrial and Engineering Chemistry Research 55 (2016) 6891–6903.

[2] M. Darby, M. Nikolaou, J. Jones, D. Nicholson, RTO: An overview and

assessment of current practice, Journal of Process Control 21 (2011) 874–884.

[3] C. Y. Chen, B. Joseph, On-line optimization using a two-phase approach: An application study, Industrial and Engineering Chemistry Research 26 (1987) 1924–1930.

[4] A. G. Marchetti, T. de Avila Ferreira, S. C. Costello, D. Bonvin, Modifier adaptation as a feedback control scheme, Industrial and Engineering Chemistry Research (2020).

[5] J. Matias, J. Jäschke, Online model maintenance via output modifier adaptation, Industrial and Engineering Chemistry Research (2019).

[6] A. Marchetti, B. Chachuat, D. Bonvin, Modifier-adaptation methodology for real-time optimization, Industrial and Engineering Chemistry Research 48 (2009) 6022–6033.

[7] S. Engell, Feedback control for optimal process operation, Journal of Process Control 17 (2007) 203–219.

[8] W. Gao, S. Engell, Iterative set-point optimization of batch chromatography, Computers and Chemical Engineering 29 (2005) 1401–1409.

[9] A. Ahmad, W. Gao, S. Engell, A study of model adaptation in iterative real-time optimization of processes with uncertainties, Computers and Chemical Engineering 122 (2019) 218–227.

[10] W. Gao, S. Wenzel, S. Engell, A reliable modifier-adaptation strategy for real-time optimization, Computers and Chemical Engineering 91 (2016) 318–328.

[11] T. Faulwasser, D. Bonvin, On the use of second-order modifiers for real-time optimization, IFAC Papers online 47 (2014) 7622–7628.

[12] R. Hille, H. M. Budman, Simultaneous identification and optimization of biochemical processes under model-plant mismatch using output uncertainty bounds, Computers & Chemical Engineering 113 (2018) 125–138.

[13] B. McKay, M. Willis, G. Barton, Steady-state modelling of chemical process systems using genetic programming, Computers and Chemical Engineering 21 (1997) 981–996.

[14] B. McKay, M. Willis, D. Searson, G. Montague, Non-linear continuum regression using genetic programming, in: Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 2, Morgan Kaufmann Publishers Inc., 1999, pp. 1106–1111.

[15] P. Balasubramaniam, A. V. A. Kumar, Solution of matrix riccati differential equation for nonlinear singular system using genetic programming, Genetic Programming and Evolvable Machines 10 (2009) 71–89.

[16] M. A. Keane, J. R. Koza, J. P. Rice, Finding an impulse response function using genetic programming, in: 1993 American Control Conference, IEEE, 1993, pp. 2345–2350.

[17] A. Cozad, N. V. Sahinidis, A global minlp approach to symbolic regression, Mathematical Programming 170 (2018) 97–119.

[18] J. Forbes, T. Marlin, J. MacGregor, Model adequacy requirements for optimizing plant operations, Computers and Chemical Engineering 18 (1994) 497–510.

[19] H. Yeomans, I. E. Grossmann, A systematic modeling framework of superstructure optimization in process synthesis, Computers and Chemical Engineering 23 (1999) 709–731.

[20] R. Raman, I. E. Grossmann, Modelling and computational techniques for logic based integer programming, Computers and Chemical Engineering 18 (1994) 563–578.

[21] A. G. Marchetti, G. François, T. Faulwasser, D. Bonvin, Modifier adaptation for real-time optimizationmethods and applications, Processes 4 (2016) 55.

[22] L. Hebing, F. Tran, H. Brandt, S. Engell, Robust optimizing control of fermentation processes based on a set of structurally different process models, Industrial and Engineering Chemistry Research 59 (2020) 2566–2580.

[23] W. H. Greene, Econometric analysis, Pearson Education India, 2003.

[24] E. Walter, L. Pronzato, Identification of parametric models, volume 8, Springer, 1997.

[25] E. Walter, Y. Lecourtier, J. Happel, On the structural output distinguishability of parametric models, and its relations with structural identifiability, IEEE Transactions on Automatic Control 29 (1984) 56–57.

[26] H. Miao, X. Xia, A. S. Perelson, H. Wu, On identifiability of nonlinear ODE models and applications in viral dynamics, SIAM review 53 (2011) 3–39.

[27] G. E. Box, W. G. Hunter, The experimental study of physical mechanisms, Technometrics 7 (1965) 23–42.

[28] Y. Bard, Nonlinear parameter estimation, Academic press, 1974.

[29] A. Papasavvas, G. François, Output modifier adaptation with filter-based constraints, Journal of Process Control 87 (2020) 37–53.

[30] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, M. Diehl, CasADi – A software framework for nonlinear optimization and optimal control, Mathematical Programming Computation 11 (2019) 1–36.

[31] A. Wächter, L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Mathematical programming 106 (2006) 25–57.

[32] M. Tawarmalani, N. V. Sahinidis, A polyhedral branch-and-cut approach to global optimization, Mathematical Programming 103 (2005) 225–249.

[33] A. J. Peixoto, D. Pereira-Dias, A. F. Xaud, A. R. Secchi, Modelling and extremum seeking control of gas lifted oil wells, IFAC-PapersOnLine 48 (2015) 21–26.

[34] D. Krishnamoorthy, B. Foss, S. Skogestad, Real-time optimization under uncertainty applied to a gas lifted well network, Processes 4 (2016) 52.

[35] R. Brekelmans, L. Driessen, H. Hamers, D. Den Hertog, Gradient estimation schemes for noisy functions, Journal of Optimization Theory and Applications 126 (2005) 529–551.

[36] C. Yang, K. Wang, Z. Shao, L. T. Biegler, Integrated parameter mapping and real-time optimization for load changes in high-temperature gas-cooled pebble bed reactors, Industrial and Engineering Chemistry Research 57 (2018) 9171–9184.