# Global Optimization of Distillation Columns using Explicit and Implicit Surrogate Models

Tobias Keßler[a,*], Christian Kunde[a], Kevin McBride[b], Nick Mertens[c], Dennis Michaels, Kai Sundmacher[a,b], Achim Kienle[a,b]

[a] *Otto-von-Guericke-Universität Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany*
[b] *Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany*
[c] *TU Dortmund, Vogelpothsweg 87, 44227 Dortmund, Germany*

## Abstract

Surrogate-based optimization of distillation columns using an iterative Kriging approach is investigated. To avoid sub-optimal local minima the focus lies on deterministic global optimization. The determination of optimal setups and operating conditions for ideal and non-ideal distillation columns, leading to mixed-integer nonlinear programming (MINLP) problems, serve as case studies. To cope with output multiplicities of the model an implicit surrogate formulation is proposed.

*Keywords:* Global Optimization, Distillation, Surrogate Models, Kriging, Output Multiplicities

### Nomenclature

**Vector-valued:**

| | |
|---|---|
| $\mathbf{a}_i$ | vapor pressure parameters |
| $\boldsymbol{F}$ | interpolation matrix |
| $\boldsymbol{I}$ | unity matrix |
| $\boldsymbol{R}$ | correlation matrix |
| $\boldsymbol{Y}$ | output matrix |
| $\mathbf{1}$ | vector of ones |
| $\boldsymbol{\alpha}$ | relative volatilities |
| $\bar{\boldsymbol{x}}$ | reference points |
| $\boldsymbol{\chi}$ | liquid mole fraction |
| $\boldsymbol{\gamma}$ | vector of vector activity coefficients |
| $\hat{\boldsymbol{f}}(\boldsymbol{x})$ | surrogate model |
| $\boldsymbol{\kappa}$ | cost function parameters |
| $\boldsymbol{\mu}$ | parameter continuation solution |
| $\boldsymbol{\Upsilon}$ | vapor mole fraction |
| $\boldsymbol{\xi}$ | constant trend |
| $\boldsymbol{e}_k$ | $k$th unit vector |
| $\boldsymbol{f}(\boldsymbol{x})$ | original function |
| $\boldsymbol{F}(\boldsymbol{\mu},\eta)$ | parameterized system |
| $\boldsymbol{F}_\mu$ | Jacobian |
| $\boldsymbol{g}$ | inequality constraints |
| $\boldsymbol{h}$ | equality constraints |
| $\boldsymbol{p},\boldsymbol{\theta}$ | Kriging design parameters |
| $\boldsymbol{P}_{\text{vap}}$ | vapor pressure |
| $\boldsymbol{q}(\boldsymbol{x})$ | trend function |
| $\boldsymbol{w}(\boldsymbol{x})$ | weight function |
| $\boldsymbol{z}(\boldsymbol{x})$ | weighted sum of reference points |
| $\boldsymbol{z}^{(k)}$ | solution tangent |
| $\boldsymbol{z}_{\text{feed}}$ | molar feed composition |

**Real-valued:**

| | |
|---|---|
| $\varepsilon$ | relative improvement |
| $\eta$ | parameterization parameter |
| $\log(L)$ | logarithmic likelihood |
| $\sigma$ | continuation step size |
| $B$ | bottom product flow rate |
| $C$ | continuous variable set |
| $c(\boldsymbol{x},\boldsymbol{y})$ | correlation function |
| $D$ | discrete variable set |
| $d$ | output dimension |
| $F$ | feed flow rate |
| $G$ | domain of degrees of freedom |
| $i$, $j$, $k$ | count variable |
| $J(\boldsymbol{x})$ | objective function |
| $l_{\text{col}}$ | column length |
| $l_{\text{r}}$ | length of rectifying section |
| $l_{\text{s}}$ | length of stripping section |
| $m$ | input dimension |
| $N$ | number of reference points |
| $P$ | total pressure |
| $r$ | recovery rate |
| $S$ | sampling region |

[*] Corresponding author
*Email addresses:* `tobias.kessler@ovgu.de` (Tobias Keßler), `christian.kunde@ovgu.de` (Christian Kunde), `mcbride@mpi-magdeburg.mpg.de` (Kevin McBride), `nick.mertens@math.tu-dortmund.de` (Nick Mertens), `kai.sundmacher@ovgu.de` (Kai Sundmacher), `achim.kienle@ovgu.de` (Achim Kienle)

*Preprint submitted to Chemical Engineering Science* *February 15, 2019*

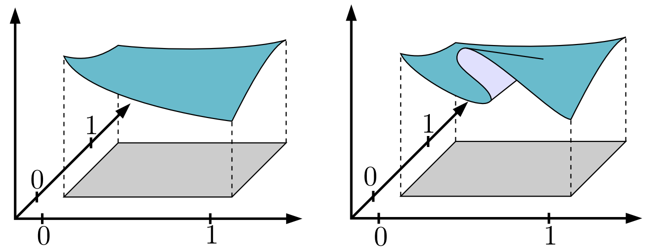
| | |
|---|---|
| $T$ | temperature |
| $TAC$ | total annualized cost |
| $V$ | vapor flow rate |
| $X$ | reference input set |

## 1. Introduction

Rigorous optimization of distillation columns is of major interest in the chemical process industry due to its high economical impact. Due to the presence of discrete and continuous decision variables this leads to mixed-integer nonlinear programs (MINLP). Standard local optimization or stochastic optimization approaches can not guarantee that the optimum found by the optimizer is a global optimum. Alternatively, deterministic global optimization based on convex relaxations within a branch and bound framework is an interesting approach for solving such problems, e.g. see the textbook by Locatelli and Schoen (2013) for an introduction. However, computation times are often prohibitively large when using general purpose global optimization software (Nallasivam et al., 2016).

Exploiting problem-specific properties enables significant reductions of the computational effort, which was demonstrated in (Ballerstein et al., 2015) for binary ideal distillation and extended in (Mertens et al., 2018) to multicomponent mixtures. The monotonicity of concentration variables in distillation columns was exploited to systematically reduce the search space of the problem using specific model reformulations. However, in general, an extension to non-ideal mixtures is not possible (Keßler et al., 2019).

Different solution approaches were recently proposed for the case that a suitable method for deterministic global optimization of the original model is not available. Nallasivam et al. (2016) presented an algorithm for calculating minimum energy requirements for thermally coupled distillation column configurations. The algorithm is based on a shortcut model which is only valid for ideal mixtures under minimum reflux conditions. Quirante et al. (2015) suggested to use surrogate models based on Kriging interpolation for optimization of distillation columns to reduce computational complexity. The main emphasis here was on local optimization but it was also suggested to use Kriging models to reduce computational complexity in deterministic global optimization. Ibrahim et al. (2018) used an artificial neural network in combination with a support vector machine for the optimization of a crude oil distillation unit. The resulting MINLP was solved using a genetic algorithm. Eason and Biegler (2016) presented a hybrid approach for process optimization, where computationally expensive parts of a process are replaced by Kriging models. A novel trust region optimization approach was used to solve their NLP problems. As the choice of the right surrogate model is non-trivial a novel error metric approach for the selection of the best surrogate was presented by Audet et al. (2018). Kieslich et al. (2018) used an iterative surrogate optimization approach with polynomial surrogates, employing a multi-start local search in each iteration. Thorough literature reviews on surrogate based modeling (Bhosekar and Ierapetritou, 2018; Saman et al., 2012) and surrogate assisted optimization (Boukouvala et al., 2016; Forrester and Keane, 2009) can be found in various fields of engineering.



(a) without output multiplicities (b) with output multiplicities

Figure 1: Sketches of response surfaces. The input set is shaded, the output is colored.

The difficulties in solving the original model formulation can be traced back to computationally expensive correlations. In this work, surrogate models are used to describe these expensive correlations and the resulting models are solved to global optimality. By an iterative approach, the surrogate is further refined and the solution is adapted in order to be feasible for the original model. This method is an extension to the algorithm presented in the paper (Keßler et al., 2019). Prior results are extended by formalizing the algorithm and by dealing with more complex case studies. One of the considered mixtures results in a setting with multiple possible product compositions for the same column specifications. Such output multiplicities are given implicitly as the solution set of a highly nonlinear equation system. Standard Kriging approaches are able to deal with smooth, continuous response surfaces without output multiplicities, like depicted in Figure 1a. The figure shows the response surface to the input set $[0,1] \times [0,1]$. However, they are not able to cover problems where output multiplicities may occur and behave poorly for systems with discontinuities or nonlinear trends (Stephenson et al., 2004). One example of such a response surface with output multiplicities is depicted in Figure 1b. Hence, implicit surrogate models are investigated and used for those cases.

The outline of the paper is as follows. The general concept of Kriging interpolation is briefly explained in Section 2. Section 3 defines the considered optimization problems. The method employed in this work to solve the optimization problems and its algorithmic implementation is described in Section 4. Section 5 includes two case studies. The first case study consists of two parts. In the first part, a globally optimal reference solution for an ideal multicomponent distillation process is calculated with the methods described by Mertens et al. (2018). It is demonstrated that the new surrogate-based approach finds a solution close to a global optimum. In the second part, the thermodynamics of the distillation process are assumed to be non-ideal and the surrogate-based approach is used to calculate an optimal solution. By comparing the ideal and non-ideal optimization results it is shown that the process behaves only moderately non-ideally in the region of interest. In the second case study a highly non-ideal multicomponent distillation process is considered, which does not admit an explicit Kriging formulation because of steady-state multiplicities. It is optimized using the surrogate-based approach with implicit Kernel interpolation.

## 2. Surrogate models

Input-output or response surface models can be used to approximate rigorous process models, trading computational efficiency for accuracy. Numerous approaches such as artificial neural networks, e.g. (Nentwich and Engell, 2016), support vector machines, e.g. (Bennett and Campbell, 2000), polynomial chaos expansions, e.g. (Zhang and Sahinidis, 2013), and more advanced approaches like the ALAMO toolbox (Cozad et al., 2014) for an automated learning of algebraic models for optimization are used in many fields of engineering. In this work Kriging and Kernel interpolation are used, the former gained much interest in chemical engineering over the last years due to its inherent statistics. Note that the other approaches mentioned above may also be utilized instead of Kriging and Kernel Interpolation to generate the surrogate models used in this work.

### 2.1. Kriging Interpolation

Let $\boldsymbol{f} : \mathbb{R}^m \to \mathbb{R}^d$ be the vector-valued function to be approximated. The corresponding Kriging interpolation $\hat{\boldsymbol{f}} : \mathbb{R}^m \to \mathbb{R}^d$ is of the form $\hat{\boldsymbol{f}}(\boldsymbol{x}) := \boldsymbol{q}(\boldsymbol{x}) + \boldsymbol{z}(\boldsymbol{x})$. In general, $\boldsymbol{q}(\boldsymbol{x})$ is a vector-valued polynomial that is fitted to the original function $\boldsymbol{f}(\boldsymbol{x})$. Function $\boldsymbol{z}(\boldsymbol{x})$ is also vector-valued and designed to approximate the remaining difference between $\boldsymbol{q}(\boldsymbol{x})$ and $\boldsymbol{f}(\boldsymbol{x})$. In this work, ordinary Kriging is applied, which means that the polynomial $\boldsymbol{q}(\boldsymbol{x})$ is chosen to be equal to a constant vector $\boldsymbol{\xi}$. As noted by Papalambros and Wilde (2000), this restriction does not impair the accuracy of the resulting Kriging model for smooth functions significantly, as most of its information is contained in $\boldsymbol{z}(\boldsymbol{x})$.

The construction of $\boldsymbol{z}(\boldsymbol{x})$ is derived in the following. It uses the evaluation of function $\boldsymbol{f}(\boldsymbol{x})$ at a certain number of *reference points* $\bar{\boldsymbol{x}}^k \in \mathbb{R}^m$, $k = 1, \dots, N$. The function value of $\boldsymbol{z}(\boldsymbol{x})$ is then given by the weigthed sum of the difference between $\boldsymbol{q}(\boldsymbol{x})$ and $\boldsymbol{f}(\boldsymbol{x})$ at all these reference points, i.e.

$$\boldsymbol{z}(\boldsymbol{x}) = \sum_{k=1}^{N} \left(\boldsymbol{f}\left(\bar{\boldsymbol{x}}^k\right) - \boldsymbol{\xi}\right) w_k(\boldsymbol{x}),$$

with a vector-valued weight function $\boldsymbol{w}(\boldsymbol{x}) = (w_1(\boldsymbol{x}), \dots, w_N(\boldsymbol{x}))^\top$ depending on the distance between $\boldsymbol{x}$ and $\bar{\boldsymbol{x}}^k$. This leads to the following form of the Kriging interpolation:

$$\hat{\boldsymbol{f}}(\boldsymbol{x}) = \boldsymbol{\xi} + \begin{pmatrix} \left(\boldsymbol{f}\left(\bar{\boldsymbol{x}}^1\right) - \boldsymbol{\xi}\right)^\top \\ \vdots \\ \left(\boldsymbol{f}\left(\bar{\boldsymbol{x}}^N\right) - \boldsymbol{\xi}\right)^\top \end{pmatrix}^\top \boldsymbol{w}(\boldsymbol{x}). \tag{1}$$

The weights are calculated using a parameterized function $c : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ and a correlation matrix $\boldsymbol{R} \in \mathbb{R}^{N \times N}$ as

$$\boldsymbol{w}(\boldsymbol{x}) = \boldsymbol{R} \begin{pmatrix} c(\boldsymbol{x}, \bar{\boldsymbol{x}}^1) \\ \vdots \\ c(\boldsymbol{x}, \bar{\boldsymbol{x}}^N) \end{pmatrix}. \tag{2}$$

The choice of $c$ is non-trivial and crucial for the accuracy of the final surrogate model. As the best choice is usually unknown a priori, a parameterized function is chosen and the parameters are optimized to generate the surrogate model. One frequently used function is of the form (Caballero and Grossmann, 2008)

$$c(\boldsymbol{x}^1, \boldsymbol{x}^2) = \exp\left(-\sum_{i=1}^{m} \theta_i \|\boldsymbol{x}_i^1 - \boldsymbol{x}_i^2\|^{p_i}\right).$$

For $\boldsymbol{x}^1 = \boldsymbol{x}^2$ the function equals 1, the parameter vector $\boldsymbol{\theta} \in \mathbb{R}^m$ dictates how fast the function tends to zero for $\boldsymbol{x}^1 \neq \boldsymbol{x}^2$ and the parameter vector $\boldsymbol{p} \in \mathbb{R}^m$ dictates how smooth the function is.

The interpolation matrix $\boldsymbol{F} \in \mathbb{R}^{d \times N}$ is defined as

$$\boldsymbol{F} = \left(\boldsymbol{f}(\bar{\boldsymbol{x}}^1) - \boldsymbol{\xi}, \dots, \boldsymbol{f}(\bar{\boldsymbol{x}}^N) - \boldsymbol{\xi}\right) \boldsymbol{R}, \tag{3}$$

to obtain the final equation for the Kriging interpolation:

$$\hat{\boldsymbol{f}}(\boldsymbol{x}) = \boldsymbol{\xi} + \boldsymbol{F} \begin{pmatrix} c(\boldsymbol{x}, \bar{\boldsymbol{x}}^1) \\ \vdots \\ c(\boldsymbol{x}, \bar{\boldsymbol{x}}^N) \end{pmatrix}. \tag{4}$$

Kriging models interpolate the original function, i.e. the approximation given by the surrogate equals the original function values at each reference point,

$$\hat{\boldsymbol{f}}\left(\bar{\boldsymbol{x}}^k\right) = \boldsymbol{f}\left(\bar{\boldsymbol{x}}^k\right) \quad \text{for all} \quad k \in \{1, \dots, N\}.$$

In order to achieve this, the weight $\boldsymbol{w}(\boldsymbol{x})$ in Equation (1) is set to be the $k$th unit vector for reference point $k$,

$$\boldsymbol{w}\left(\bar{\boldsymbol{x}}^k\right) = \boldsymbol{e}_k, \quad \text{for all} \quad k \in \{1, \dots, N\},$$

or equivalently

$$\begin{pmatrix} \boldsymbol{w}^\top\left(\bar{\boldsymbol{x}}^1\right) \\ \boldsymbol{w}^\top\left(\bar{\boldsymbol{x}}^2\right) \\ \vdots \\ \boldsymbol{w}^\top\left(\bar{\boldsymbol{x}}^N\right) \end{pmatrix} = \boldsymbol{I}.$$

By Equation (2), this, for instance, holds, if

$$\boldsymbol{R} = \begin{pmatrix} c(\bar{\boldsymbol{x}}^1, \bar{\boldsymbol{x}}^1) & \dots & c(\bar{\boldsymbol{x}}^N, \bar{\boldsymbol{x}}^1) \\ \vdots & \ddots & \vdots \\ c(\bar{\boldsymbol{x}}^1, \bar{\boldsymbol{x}}^N) & \dots & c(\bar{\boldsymbol{x}}^N, \bar{\boldsymbol{x}}^N) \end{pmatrix}^{-1} . \tag{5}$$

To save time during the execution of the interpolation of Equation (4), the interpolation matrix $\boldsymbol{F}$ is calculated offline using Equations (3) and (5), that yield

$$\boldsymbol{F}\,\boldsymbol{R}^{-1} = \begin{pmatrix} \left(\boldsymbol{f}(\bar{\boldsymbol{x}}^1) - \boldsymbol{\xi}\right)^\top \\ \vdots \\ \left(\boldsymbol{f}(\bar{\boldsymbol{x}}^N) - \boldsymbol{\xi}\right)^\top \end{pmatrix}^\top .$$

The parameter vectors $\boldsymbol{\theta}$, $\boldsymbol{p}$ and $\boldsymbol{\xi}$ have to be calculated during the so-called fitting process. Following Quirante et al. (2015) the fitting is done by maximizing the logarithmic likelihood function, i.e. minimizing the negative logarithmic likelihood function

$$\begin{aligned} &\min\{-\log(L) \,|\, p_i \in [0, 1.99], \theta_i > 0\}, \\ \log(L) &= -N/2 \left(\log\left(\sigma^2\right) + \log(2\pi)\right) \\ &\quad - 1/2 \log\left(\|\boldsymbol{R}\|\right) \\ &\quad - 1/\left(2\sigma^2\right) \left(\boldsymbol{Y}(\bar{\boldsymbol{x}}) - \boldsymbol{1}\boldsymbol{\xi}^\top\right)^\top \boldsymbol{R}^{-1} \left(\boldsymbol{Y}(\bar{\boldsymbol{x}}) - \boldsymbol{1}\boldsymbol{\xi}^\top\right), \end{aligned}$$

where

$$\sigma^2 = 1/N \left(\boldsymbol{Y}(\bar{\boldsymbol{x}}) - \boldsymbol{1}\boldsymbol{\xi}^\top\right)^\top \boldsymbol{R}^{-1} \left(\boldsymbol{Y}(\bar{\boldsymbol{x}}) - \boldsymbol{1}\boldsymbol{\xi}^\top\right),$$

and the output matrix $\boldsymbol{Y} \in \mathbb{R}^{N \times d}$ is defined as

$$\boldsymbol{Y}(\bar{\boldsymbol{x}}) = \begin{pmatrix} \boldsymbol{f}^\top\left(\bar{\boldsymbol{x}}^1\right) \\ \vdots \\ \boldsymbol{f}^\top\left(\bar{\boldsymbol{x}}^N\right) \end{pmatrix} .$$

In this paper the parameter vector $\boldsymbol{\xi}$ is set to be the mean of the output, i.e.

$$\boldsymbol{\xi} = \frac{1}{N} \sum_{j=1}^{N} \boldsymbol{f}\left(\bar{\boldsymbol{x}}^j\right) .$$

### *2.2. Kernel Interpolation*

Calculating statistical properties of Kriging interpolation requires knowledge of the correct covariance function $c$. If the chosen correlation $c$ does not have the properties of a covariance, the method presented above is called Kernel interpolation. As noted by Scheuerer et al. (2013) Kernel interpolation is mathematically equivalent to Kriging interpolation, but allows more general correlation functions.

For the implicit surrogate formulations used in the second case study presented in this work a linear correlation function is employed, therefore it will be called Kernel interpolation. The kernel function used for fitting the surrogate model is

$$c(\boldsymbol{x}^1, \boldsymbol{x}^2) = 10 \prod_{i=1}^{m} 1 - \left|\boldsymbol{x}_i^1 - \boldsymbol{x}_i^2\right| .$$

The interpolation itself remains the same.

## 3. Definition of the Optimization Problems

The original optimization problems dealt with in this work are mixed integer nonlinear programming (MINLP) problems of the following form:

$$\begin{array}{ll} \min & J(\tilde{\boldsymbol{x}}), \\ \text{s. t.} & \boldsymbol{h}(\tilde{\boldsymbol{x}}) = 0, \\ & \boldsymbol{g}(\tilde{\boldsymbol{x}}) \leq 0, \\ & \tilde{\boldsymbol{x}} \in G, \quad G \subseteq \mathbb{R}^n, \\ & \tilde{x}_i \in Z, \quad Z \subseteq \mathbb{Z}, \quad \text{for all } i \in I, \end{array} \tag{OP}$$

where $J(\tilde{\boldsymbol{x}})$ is the objective function, $\boldsymbol{h}(\tilde{\boldsymbol{x}})$ are the equality constraints, such as mass balances, and $\boldsymbol{g}(\tilde{\boldsymbol{x}})$ are the inequality constraints, such as purity requirements. Variables $\tilde{x}_i$ with $i \in I$ have to fulfill integrality restrictions.

For the applications considered in the computational study, the variables $\tilde{x}_i$ can be interpreted as either model input or designated model output. Hence, the variable vector $\tilde{\boldsymbol{x}}$ is divided into two parts denoted by $\boldsymbol{x}$ (input) and $y$ (output), i.e. $\tilde{\boldsymbol{x}} \mathrel{\widehat{=}} (\boldsymbol{x}, y)$. Note that, depending on the problem, different choices of $\boldsymbol{x}$ and $y$ out of $\tilde{\boldsymbol{x}}$ may be possible.

In a next step, an approximation of Problem (OP) is constructed by reorganizing the prior equality constraints $\boldsymbol{h}(\tilde{\boldsymbol{x}})$. Computationally expensive correlations between $\boldsymbol{x}$ and $y$ are treated as a black-box and replaced by a surrogate function $\hat{h}(\boldsymbol{x}, y)$, while remaining equality constraints are captured by $\bar{\boldsymbol{h}}(\boldsymbol{x}, y) = 0$. Furthermore, the variables $\boldsymbol{x}$ are assumed to be bounded and scaled onto the interval $[0, 1]$ in order to fit the black-box model.

The resulting approximation problem is given by

$$\begin{array}{ll} \min & J(\boldsymbol{x}), \\ \text{s. t.} & \hat{h}(\boldsymbol{x}, y) = 0, \\ & \bar{\boldsymbol{h}}(\boldsymbol{x}, y) = 0, \\ & \boldsymbol{g}(\boldsymbol{x}, y) \leq 0, \\ & \boldsymbol{x} \in S, \quad S \subseteq [0, 1]^m, \\ & x_i \in Z, \quad \text{for all } i \in I, \\ & y \in K, \quad K \subseteq \mathbb{R}. \end{array} \tag{AP}$$

For this work, it is assumed that all functions are continuous and that there exists a feasible solution to both Problems (OP) and (AP).

In the following sections, an algorithm is described and applied that iteratively solves Problem (AP) in order to find good approximate solutions for Problem (OP). In each iteration, the feasible set is adapted and the quality of the surrogate function $\hat{h}(\boldsymbol{x}, y)$ is enhanced.

Two separate cases are considered. The first case deals with the so-called *explicit model*, for which the computationally expensive correlation between input $\boldsymbol{x}$ and output $y$ can be described explicitly by a function $y := f(\boldsymbol{x})$. Replacing $f(\boldsymbol{x})$ by a surrogate function $\hat{f}(\boldsymbol{x})$ as described in Section 2 gives rise to the constraint function

$$\hat{h}(\boldsymbol{x}, y) := \hat{f}(\boldsymbol{x}) - y$$

for Problem (AP).

The second case deals with the so-called *implicit model* that is used when output multiplicities, i.e. multiple solutions $y$ for the same input $\boldsymbol{x}$, occur. These solutions are given implicitly by an equation $f(\boldsymbol{x}, y) = 0$. In this case, a surrogate function $\hat{f}(\boldsymbol{x}, y)$ is used to derive the contraint function

$$\hat{h}(\boldsymbol{x}, y) := \hat{f}(\boldsymbol{x}, y)$$

for Problem (AP).

To the best of the authors' knowledge, capturing output multiplicities of real processes with Kriging or Kernel interpolation, as in the latter case, is not yet discussed in the literature.

## 4. Algorithmic Implementation

The algorithm used in this work is implemented into an automatic framework in Matlab and calls GAMS/BARON (Tawarmalani and Sahinidis, 2005) as a global subsolver. The algorithm is displayed as pseudo-code in Algorithm 1 and is described in more detail, next. For the sake of simplicity the presentation of the algorithm is restricted to the explicit model in this section. The adaptation for the implicit model is mentioned in Section 5.

**Algorithm 1** Optimization Algorithm (stated for the explicit model)

```
Input: Optimization problem (OP). Desired number of reference points $N$.
Ouput: solution vector $\boldsymbol{x}^{\text{opt}}$
set initial sampling region $S = [0,1]^m$ (corresponds to the initial domain for Problem (AP))
set reference input set $X = \{0,1\}^m$ (all vertices of $S$)
set relative improvement $\varepsilon = 1$
set iteration counter $i = 0$
repeat
    calculate $i = i + 1$
    use MATLAB function HALTONSET$(N, S)$ to generate reference inputs $\bar{\boldsymbol{x}}^1, \ldots, \bar{\boldsymbol{x}}^N$
    for all $\bar{\boldsymbol{x}} \in (X \cap S) \cup \{\bar{\boldsymbol{x}}^1, \ldots, \bar{\boldsymbol{x}}^N\}$ do
        evaluate $\boldsymbol{f}(\bar{\boldsymbol{x}})$
    use MATLAB/FMINCON to solve maximum likelihood for optimal Kriging parameters
    generate surrogate function $\hat{f}$ and resulting Problem (AP)
    if exist$(\boldsymbol{x}^{\text{opt}})$ then
        set optcr = $tol_2$
        call GAMS/BARON to solve (AP) for $\bar{\boldsymbol{x}}^{\text{opt},i}$ with initial value $\boldsymbol{x}_0 = \boldsymbol{x}^{\text{opt}}$
    else
        set optcr = $tol_1$
        call GAMS/BARON to solve (AP) for $\bar{\boldsymbol{x}}^{\text{opt},i}$ without initial value $\boldsymbol{x}^{\text{opt},i}$
    use MATLAB function FMINCON to solve (OP) for $\boldsymbol{x}^{\text{opt},i}$ with initial value $\boldsymbol{x}_0 = \bar{\boldsymbol{x}}^{\text{opt},i}$
    if exist$(\boldsymbol{x}^{\text{opt},i})$ then
        add $\boldsymbol{x}^{\text{opt},i}$ to sampling set $X = X \cup \{\boldsymbol{x}^{\text{opt},i}\}$
        if exist$(\boldsymbol{x}^{\text{opt}})$ then
            if $J(\boldsymbol{x}^{\text{opt},i}) < J(\boldsymbol{x}^{\text{opt}})$ then
                calculate $\varepsilon = \left|1 - J(\boldsymbol{x}^{\text{opt},i})/J((\boldsymbol{x}^{\text{opt}})\right|$
                set new best known solution $\boldsymbol{x}^{\text{opt}} = \boldsymbol{x}^{\text{opt},i}$
                define new sampling region $S$ (contracting the domain for Problem (AP) with $1/(2 \cdot i)$, centered around $\boldsymbol{x}^{\text{opt}}$)
        else
            set new best known solution $\boldsymbol{x}^{\text{opt}} = \boldsymbol{x}^{\text{opt},i}$
    else
        add $\bar{\boldsymbol{x}}^{\text{opt},i}$ to sampling set $X = X \cup \{\bar{\boldsymbol{x}}^{\text{opt},i}\}$
until iteration $i == iter_{\max}$ or $\varepsilon \leq reltol$
```

Note that the number $N$ and the choice of reference points $\bar{\boldsymbol{x}}$ may greatly influence the accuracy of the surrogate function $\hat{\boldsymbol{f}}(\boldsymbol{x})$. For a given number $N$ a Halton sequence (Kocis and Whiten, 1997) is used as a space filling approach to generate the sampling points. This prevents clustering of reference points in unimportant regions of the search space $S$. For a limited number of samples clustering in unimportant regions increases the chance that the surrogate model does not represent the search space $S$ correctly, because certain important features are not captured.

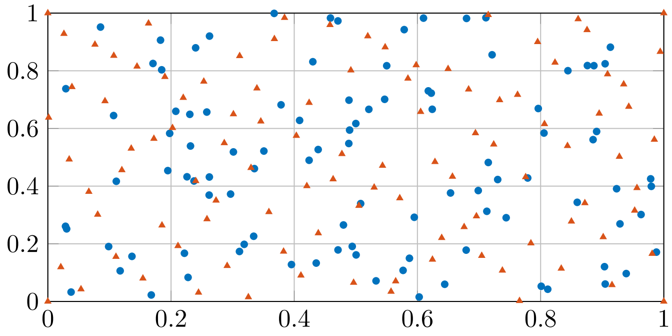


Figure 2: Difference between Halton squence (orange triangles) and random set (blue dots).(For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

Figure 2 illustrates the difference between a space filling approach and a purely random approach in two dimensions. The orange triangles represent the points chosen using a Halton sequence, whereas the blue dots are chosen randomly. Clustering of the blue dots can be observed in multiple places, e.g. in the upper right corner.

This work deals with mixed-integer problems, thus integer variables as well as continuous variables have to be chosen for each $\bar{\boldsymbol{x}}^k$. A Halton set $C$ is generated for the domain of continuous variables and another Halton set $D$ is generated for the domain of the discrete variables, where the single entries are rounded to the nearest discrete value. The reference set $X$ then comprises each combination of discrete and continuous coordinates $X = C \times D$, e.g. for 10 samples in the continuous domain and 12 samples in the integer domain there are a total of $10 \cdot 12 = 120$ reference points $\bar{\boldsymbol{x}}$. In order to prevent Kriging from extrapolating beyond the given samples, the corners of the sampling region are always included in $X$, i.e. the number of reference points $N$ is at least $2^m$. Therefore a rather large number of reference points is needed, but it is made sure that each hypersurface defined by a set of integer variables is approximated equally well. Furthermore, it was found that this approach leads to smoother hypersurfaces, which is preferable as this leads to fewer local minima being found, reducing the computational effort during global optimization.

After the sampling is done, the original function $\boldsymbol{f}(\boldsymbol{x})$ is evaluated at each $\bar{\boldsymbol{x}}^k$ to generate the output matrix $\boldsymbol{Y}$. The reference points and the outputs are scaled such that they lie in the interval $[0, 1]$. With the scaled $\boldsymbol{Y}$ and $\bar{\boldsymbol{x}}$ the Kriging parameters $\boldsymbol{p}$ and $\boldsymbol{\theta}$ are calculated, as well as the mean of the outputs $\boldsymbol{\xi}$ and the interpolation matrix $\boldsymbol{F}$. The parameters are then written into a file readable by GAMS.

In each iteration $i$, a globally optimal solution $\bar{\boldsymbol{x}}^{\text{opt},i}$ of the Surrogate Problem (AP) is obtained by applying the deterministic global optimization software GAMS/BARON. Note that surrogate problem and the Original Problem (OP) may not have the same feasible set. Mismatch between the surrogate model and the original model leads to the possibility of $\bar{\boldsymbol{x}}^{\text{opt},i}$ being suboptimal or infeasible for the original model. Therefore, the original model is subsequently solved using a local optimization method with $\bar{\boldsymbol{x}}^{\text{opt},i}$ as a starting point to obtain a solution $\boldsymbol{x}^{\text{opt},i}$ that is feasible and locally optimal for the original problem.

If a feasible $\boldsymbol{x}^{\text{opt},i}$ is found, it is checked whether the objective function $J\left(\boldsymbol{x}^{\text{opt},i}\right)$ has a smaller value than the currently best known solution $\boldsymbol{x}^{\text{opt}}$. If the value is smaller, the point is saved as the new best, known solution. Furthermore, $\boldsymbol{x}^{\text{opt},i}$ will be added to the set of reference points $X$.

To save computation time, $\boldsymbol{x}^{\text{opt}}$ is given to GAMS/BARON as a starting point, i.e. GAMS/BARON has a valid upper bound after the first iteration. An initially available upper bound usually accelerates convergence for global optimization by allowing to cut off additional search regions. For the compensation of differences between the models, Matlab and GAMS, an additional optimization step within the neighborhood of $\boldsymbol{x}^{\text{opt}}$ (±5% in the continuous variables) is conducted to find a valid starting point for the GAMS/BARON model.

As proposed by Caballero and Grossmann (2008), the search space $S$ is contracted around the optimal solution $\boldsymbol{x}^{\text{opt}}$ and a corresponding new sampling set is generated in each iteration, i.e. the sampling region $S$ gets successively smaller and is centered around $\boldsymbol{x}^{\text{opt}}$. In general, this strategy leads to a better approximation of the original problem. With an increase of the approximation quality, it is also more rewarding for the global optimization solver to reduce its error tolerance. Hence, the limit on the desired optimality gap (optcr) is also reduced after the first iteration.

Table 1: Chemical species

| No. | Name |
|---|---|
| 1 | Dimethylformamide |
| 2 | Decane |
| 3 | Dodecane |
| 4 | Dodecene |
| 5 | Tridecanal |

The entire procedure is repeated until either the iteration count $i$ reaches a predefined limit or the relative improvement of the successive feasible solution drops below a given threshold.

In this work the parameters of Algorithm 1 are set as follows. The highest allowable iteration count $iter_{\max} = 5$, the improvement threshold $\varepsilon = 0.02$ and the parameters for the relative optimality gap are set to $tol_1 = 0.1$ and $tol_2 = 0.01$.

All Kriging models constructed through our computations are implemented as MINLPs and solved using the GAMS 24.6.1 framework with the deterministic global optimization software BARON 15.9.22., Cplex 12.6.3 is used as a LP/MIP subsolver and CONOPT 3.17A is utilized as a NLP subsolver. The calculations are carried out on a Linux PC with 3.40 GHz Intel Core i7-6700 CPU and 16 GB memory.

## 5. Case Studies

This section is divided into two parts. The first part deals with the optimization of a distillation column which is moderately non-ideal in the region of interest. In this part two settings are considered. In the first setting, the non-ideal thermodynamic properties are approximated by an ideal model that is solved by the optimization Algorithm 1. A globally optimal reference solution is calculated to validate the proposed Algorithm 1 in terms of the approximation quality (see Section 5.1.1). In the second setting, the thermodynamic properties are modeled as non-ideal and Algorithm 1 is applied to obtain an approximate solution (see Section 5.1.2).

In the second part the distillation of a highly non-ideal mixture is investigated, where, due to steady-state multiplicities, the use of implicit surrogate models is necessary (see Section 5.2).

### 5.1. Explicit Surrogate Model

The process considered is a novel hydroformylation process of dodecene to tridecanal using a thermomorphic solvent system as described by Schäfer et al. (2012). A flowchart of the process is given in Figure 3. In a previous work by Keßler et al. (2017) the simultaneous rigorous global optimization of the reactor and the catalyst recycling was

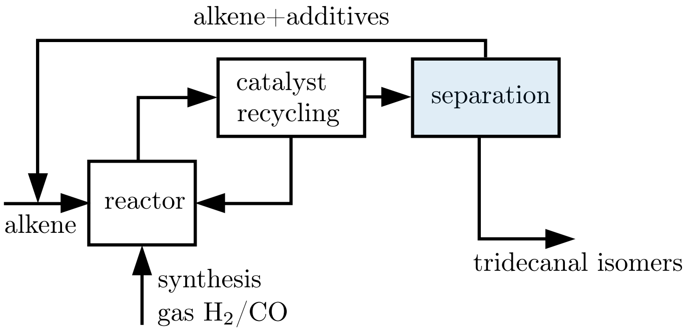


Figure 3: Process flowchart, the shaded separation is investigated in this work.

Table 2: Values and units for cost function parameters $\boldsymbol{\kappa}$

| Parameter | Value | Unit |
|---|---|---|
| $\kappa_1$ | 22824 | s/mol · \$/a |
| $\kappa_2$ | 2556.2 | \$/a |
| $\kappa_3$ | 7.2 | 1 |
| $\kappa_4$ | 0.81 | 1 |
| $\kappa_5$ | 0.525 | 1 |
| $\kappa_6$ | 163.5543 | \$/a |
| $\kappa_7$ | 0.8 | 1 |
| $\kappa_8$ | 0.97 | 1 |
| $\kappa_9$ | 0.725 | 1 |

already presented. Global optimization of the product separation, shaded in Figure 3, is considered in this work.

The five components of the mixture are given in Table 1. Note that n- and iso- species of components have similar volatilities and are therefore treated as single components in this model. Tridecanal is the high-boiler of the mixture and it gathers at the bottom of the distillation column. The distillation column is assumed to be operated at 60 mbar at the top with a linear pressure drop of 30 mbar along the column, i.e. vacuum distillation is employed.

The column model is implemented in Matlab as a dynamic model. The steady state solution for each reference point $\bar{\boldsymbol{x}}^k$ is obtained using dynamic simulation with ode15s. The model assumes: Thermodynamic equilibrium between the vapor and the liquid phase, constant molar overflow, a single feed and a total condenser.

The degrees of freedom include the vapor flow rate $V \in [18, 50]$ mol/s, the bottom product flow rate $B \in [1.233, 4.133]$ mol/s, and the number of stages in the rectifying and stripping sections $l_{\mathrm{r}}$, $l_{\mathrm{s}} \in [1, 15]$, respectively. The feed stream is assumed to be $F = 24.233$ mol/s, with a feed composition $\boldsymbol{z}_{\text{feed}} = [0.0634,\ 0.7014,\ 0.1533,\ 0.005,\ 0.0769]$.

A simplified cost function with lumped parameters based on the cost model of McBride and Sundmacher (2015) is used. The diameter of the column is assumed to be equal for the rectifying and the stripping section, fixed estimates for the density $\rho$, the vaporization enthalpy $\Delta h_{\text{vap}}$, and the molar mass $M$ from simulation data are used as simplifications to the original cost function. The total annualized cost ($TAC$) is calculated as

$$\begin{aligned} TAC = {} & \kappa_1 \cdot V + \kappa_2 \cdot (l_{col} + \kappa_3)^{\kappa_4} \cdot (V)^{\kappa_5} \\ & + \kappa_6 \cdot (l_{\text{col}} - \kappa_7)^{\kappa_8} \cdot (V)^{\kappa_9}\ \$\text{a}^{-1}, \end{aligned}$$

and takes the investment costs (length $l_{\text{col}}$ and diameter of the column) as well as the operating costs (cooling water at the condenser and vapor at the reboiler) into account. The parameters $\boldsymbol{\kappa}$ are specific to the mixture and can be found in Table 2.

Note that the separation of tridecanal from the other components is rather simple. Thus, a high purity of tridecanal in the bottom product $\chi_{5,\mathrm{B}} \geq 0.995$ is imposed. In both, the ideal and the non-ideal setting, two test instances are considered. They differ in the desired recovery rate $r$ of tridecanal in the bottom product of (a) 95% and (b) 99%.

The sampling points are chosen as described in Section 4. The number of sampling points in the initial optimization step is 47 in the continuous variables and 25 in the integer variables, i.e. a total number of 1175 sampling points is used for both case studies. Note that, as with every surrogate model, the choice of the sampling points itself and the number of sampling points are tuning parameters. The number of samples in each dimension has to be chosen depending on how complex the hypersurfaces are. In the present case the number was determined heuristically as it yielded a good coverage of the response surface. For different applications these numbers should be reevaluated, as they reflect a trade-off between the accuracy and the speed of the algorithm.

### 5.1.1. *Ideal Distillation*

In this setting the phase behavior is described using constant relative volatilities $\alpha_i$ , with $\boldsymbol{\alpha} = [51.35, 27, 8.43, 7.7, 1]$. The order of the components is the same as in Table 1.

Table 3 shows the optimization results for test instance (a) for each iteration. On the left side of the table the design variables and the computation times are stated, followed by the outputs of the Kriging models. The right side of the table shows the output of the original model using the design variables. Instances marked with an asterisk are solutions obtained via local optimization. The values given in the “time” column are the times needed in seconds for running the proposed Algorithm. They are listed separately for every single iteration.

The row labeled by “reference” shows a reference solution obtained by rigorous global optimization of the corresponding ideal distillation column model. As shown in a previous work by Mertens et al. (2018) these models can be solved to global optimality in a short amount of time using model reformulations and a problem specific bound tightening strategy. These methods are implemented in a SCIP 5.0 framework (Gleixner et al., 2017) and used to solve the rigorous column model to global optimality.

The solution obtained in the first iteration is nearly

18 % more expensive than the reference solution, but also overfulfills the desired specifications. The outputs of the Kriging model and the Matlab model are identical, meaning that the Kriging model is accurate in the region of the solution. The following local optimization reduces the cost significantly and brings the outputs to the desired specifications. In the second iteration the cost is reduced by 2.39 %, thus a third iteration is done. The solutions of the second and third iteration are identical and the algorithm terminates. In the end, the Kriging algorithm yielded a solution which is 2.38 % more expensive than the reference solution.

Table 4 shows the results for test case (b) and is structured in the same way as Table 3. Again, the solution of the first iteration is much more expensive than the reference solution with +14.13 % annual costs and the consecutive local optimization lowers this gap significantly by approximately 10 percentage points. In the second iteration the cost is further lowered. As the third iteration yields the same solution as the second, the algorithm is terminated. The final solution is only 0.68 % more expensive than the reference solution.

For both test instances, the proposed optimization algorithm generates a solution which lies in a close neighborhood to a global optimum of the rigorous distillation column model. The deviation is given by 2.4 % for test instance (a) and by 0.7 % for test instance (b). Regarding the computation times, the proposed method takes 943 seconds for all three iterations of instance (a) and 1259 seconds for all three iterations of instance (b). However, this case study is not meant to compare the computation times but to show the suitability of the Kriging approach in terms of the quality of the solution.

### *5.1.2. Non-ideal Distillation*

In the non-ideal case study the vapor liquid equilibrium is described with an ideal gas phase and a non-ideal liquid phase. The gas phase is described by Dalton's law $\boldsymbol{P}_{\text{vap}} = \boldsymbol{\Upsilon} \cdot P$ and the non-ideal liquid phase is modeled using an extended Raoult's law $\boldsymbol{P}_{\text{vap}} = \boldsymbol{\chi} \cdot \boldsymbol{\gamma} \cdot \boldsymbol{P}^*_{\text{vap}}$, where $\boldsymbol{\chi}$ and $\boldsymbol{\Upsilon}$ are the liquid and vapor mole fractions, respectively. For the calculation of the vapor pressures the vapor pressure correlation

$$\log_{10}(\boldsymbol{P}^*_{\text{vap}}) = \mathbf{a}_1 + \mathbf{a}_2 \cdot T^{-1} \\ + \mathbf{a}_3 \cdot \log_{10}(T) + \mathbf{a}_4 \cdot T + \mathbf{a}_5 \cdot T^2$$

is used, with parameters $\mathbf{a}_i$ taken from McBride and Sundmacher (2015), temperature $T$ in K and $\boldsymbol{P}^*_{\text{vap}}$ in mmHg. To model non-ideal phase behavior in the activity coefficients $\boldsymbol{\gamma}$ different thermodynamic models can be used, such as UNIQUAC or Wilson (Poling et al., 2000). In this work a modified UNIFAC (Dortmund) model (Weidlich and Gmehling, 1987) with parameters from McBride and Sundmacher (2015) is used.

Figure 4 shows the non-ideal VLE as well as the VLE with $\boldsymbol{\gamma} = 1$, i.e. the ideal VLE, for the two binary mix-

Table 3: Computation results ideal model, 95 % recovery.
Left side: Surrogate model / GAMS results - right side: Matlab model

| Iteration | $B$ | $V$ | $l_r$ | $l_s$ | $TAC$ | time | $\chi_{5,\text{B}}$ | $r$ | $\chi_{5,\text{B}}$ | $r$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.777 | 25.633 | 2 | 8 | 748582(+17.85%) | 921 | 1 | 0.9537 | 1.0000 | 0.9537 |
| 1* | 1.779 | 22.481 | 2 | 8 | 665353(+4.75%) | | | | 0.9950 | 0.9500 |
| 2 | 1.779 | 22.481 | 2 | 6 | 650155(+2.36%) | 14 | 1 | 0.9548 | 0.9948 | 0.9498 |
| 2* | 1.779 | 22.486 | 2 | 6 | 650286(+2.38%) | | | | 0.9950 | 0.9500 |
| 3 | 1.779 | 22.486 | 2 | 6 | 650287(+2.38%) | 8 | 1 | 0.9548 | 0.9950 | 0.9500 |
| 3* | 1.779 | 22.486 | 2 | 6 | 650286(+2.38%) | | | | 0.9950 | 0.9500 |
| Reference | 1.779 | 22.817 | 1 | 4 | 635182 | | | | 0.9950 | 0.9500 |

Table 4: Computation results ideal model, 99 % recovery.
Left side: Surrogate model / GAMS results - right side: Matlab model

| Iteration | $B$ | $V$ | $l_\mathrm{r}$ | $l_\mathrm{s}$ | $TAC$ | time | $\chi_{5,\mathrm{B}}$ | $r$ | $\chi_{5,\mathrm{B}}$ | $r$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.854 | 26.299 | 4 | 8 | 782383(+14.13%) | 1228 | 1 | 0.9950 | 1.0000 | 0.9950 |
| 1* | 1.854 | 23.689 | 4 | 8 | 712698(+3.97%) | | | | 0.9950 | 0.9900 |
| 2 | 1.854 | 23.689 | 4 | 5 | 689538(+0.59%) | 21 | 1 | 0.9950 | 0.9946 | 0.9897 |
| 2* | 1.854 | 23.713 | 4 | 5 | 690175(+0.68%) | | | | 0.9950 | 0.9900 |
| 3 | 1.854 | 23.713 | 4 | 5 | 690175(+0.68%) | 10 | 1 | 0.9950 | 0.9950 | 0.9900 |
| 3* | 1.854 | 23.713 | 4 | 5 | 690175(+0.68%) | | | | 0.9950 | 0.9900 |
| Reference | 1.854 | 24.149 | 3 | 4 | 685507 | | | | 0.9950 | 0.9900 |

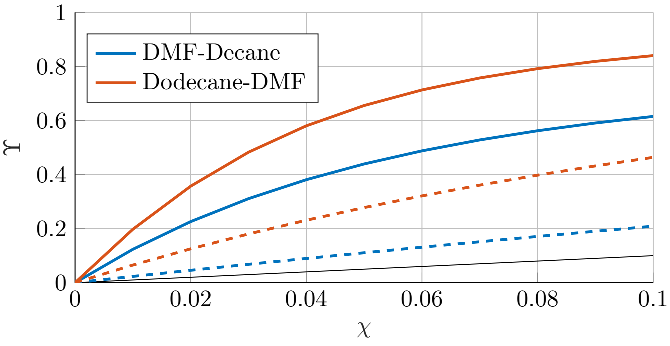


Figure 4: Binary phase equilibria for the mixtures DMF-Decane (blue) and Dodecane-DMF (orange). The solid lines show the non-ideal VLE, the dashed lines the ideal ($\boldsymbol{\gamma} = 1$) VLE. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

tures DMF-Decane and Dodecane-DMF. In previous works dealing with the hydroformylation process the column was assumed to be ideal (Steimel et al., 2013) and modeled using shortcut methods (McBride and Sundmacher, 2015; McBride et al., 2017). This case study aims at validating the previous works and demonstrating the power of the presented approach.

Algorithm 1 is applied to obtain a solution to this problem, which, based on the computational results from Section 5.1.1, is expected to lie in the neighborhood of a globally optimal solution.

The results for test instance (a) in the non-ideal setting can be found in Table 5. Due to the more complex topology of the problem as a result of the non-ideal phase equilibrium, the optimization of the first iteration takes more time than in the ideal case. The local optimization step is able to find a solution which exactly fulfills the desired specifications and is able to reduce the cost by 7 %. In the second iteration the cost is not reduced any further, thus the algorithm terminates.

The results for test instance (b) in the non-ideal setting can be found in Table 6. Like in the ideal setting, the time needed to solve this problem is much higher than for test instance (a), because the feasible region is smaller, i.e. the specification is stricter. The local refinement generates a solution exactly on the specifications. In the second iteration a solution is generated, which is slightly (0.25 %) more expensive than the first solution. Therefore the algorithm terminates.

It turns out that the results in the ideal and the non-ideal setting are very similar. The ideal optimal solutions are a little bit more expensive, i.e. more conservative.

### 5.2. Implicit Surrogate Model

An implicit surrogate formulation for systems with output multiplicities as in Figure 1b is proposed in this work. Instead of building an explicit surrogate $\hat{\boldsymbol{f}}(\boldsymbol{x}) \approx y$ the new surrogate is postulated as $\hat{\boldsymbol{f}}(\boldsymbol{x}, y) \approx 0$. It is well known

Table 6: Computation results non-ideal model, 99 % recovery.
Left side: Surrogate model / GAMS results - right side: Matlab model

| Iteration | $B$ | $V$ | $l_\mathrm{r}$ | $l_\mathrm{s}$ | $TAC$ | time | $\chi_{5,\mathrm{B}}$ | $r$ | $\chi_{5,\mathrm{B}}$ | $r$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.852 | 28.705 | 4 | 4 | 811677 | 60792 | 1 | 0.9937 | 0.9997 | 0.9934 |
| 1* | 1.854 | 23.412 | 4 | 4 | 674449 | | | | 0.9950 | 0.9900 |
| 2 | 1.855 | 23.081 | 2 | 4 | 650082 | 186 | 1 | 0.9955 | 0.9875 | 0.9830 |
| 2* | 1.854 | 24.096 | 2 | 4 | 676158 | | | | 0.9950 | 0.9900 |

Table 5: Computation results non-ideal model, 95 % recovery.
Left side: Surrogate model / GAMS results - right side: Matlab model

| Iteration | $B$ | $V$ | $l_\mathrm{r}$ | $l_\mathrm{s}$ | $TAC$ | time | $\chi_{5,\mathrm{B}}$ | $r$ | $\chi_{5,\mathrm{B}}$ | $r$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.779 | 22.728 | 3 | 4 | 694306 | 1253 | 1 | 0.9500 | 0.9992 | 0.9538 |
| 1* | 1.779 | 22.728 | 3 | 4 | 648854 | | | | 0.9950 | 0.9500 |
| 2 | 1.779 | 22.728 | 3 | 4 | 646537 | 290 | 1 | 0.9500 | 0.9934 | 0.9532 |
| 2* | 1.779 | 22.728 | 3 | 4 | 648854 | | | | 0.9950 | 0.9500 |

from bifurcation analysis (Seydel, 2009) that algebraic systems of this form are able to capture output multiplicities.

Although the single steps of Algorithm 1 remain the same, i.e. sampling, fitting, global optimization and local optimization, for each of these steps a different strategy than in the explicit case is needed.

In this particular example the sampling is done using a parameter continuation (Rheinboldt, 1986; Seydel, 2009). After an initial sampling to generate different column designs, i.e. vapor flow, length of the column and feed stage, for each unique column design a parameter continuation along the distillate product $D$ is conducted.

Parameter continuation methods are used to approximate a solution of an underdetermined equation system of the form

$$\boldsymbol{f}(\boldsymbol{x}, y, \omega) = 0, \tag{6}$$

where $\omega$ is a free parameter. In the present case $\omega$ is set to be the distillate flow rate $D$. Solutions of Equation (6) are denoted as $\boldsymbol{\mu}$. Starting from a known solution $\mu^{(k)}$ the free parameter gets shifted and the corresponding other states, $\boldsymbol{x}^{(k+1)}$ and $y^{(k+1)}$, are calculated until a predefined value of $D$ is reached. To achieve this an iterative predictor-corrector approach is used, which will be discussed in the following. An illustration of the discussed predictor-corrector approach can be found in Figure 5.
In the present work a so called *tangent predictor* (Seydel, 2009) is used. A tangent to the current solution $\mu^{(k)}$ is calculated by solving the linear algebraic system

$$\boldsymbol{F}_\mu \left( \boldsymbol{f} \left( \mu^{(k)} \right) \right) \boldsymbol{z}^{(k)} = 0, \tag{7}$$

where $\boldsymbol{F}_\mu \in \mathbb{R}^{m+1 \times m+2}$ is the Jacobian and $\boldsymbol{z}^{(k)}$ is the tangent vector of the $k$th solution. In order to fix the length and orientation of $\boldsymbol{z}^{(k)}$ in Equation (7), the Jacobian needs to be extended by a normalizing equation. One possible choice is $\boldsymbol{e}_j^\top \boldsymbol{z}^{(k)} = 1$, which yields the following tangent equation:

$$\boldsymbol{z}^{(k)} = \begin{pmatrix} \boldsymbol{F}_\mu \left( \boldsymbol{f} \left( \mu^{(k)} \right) \right) \\ \boldsymbol{e}_j \end{pmatrix}^{-1} \boldsymbol{e}_{m+2}.$$

Using the tangent it is possible to calculate a predicted new solution $\hat{\boldsymbol{\mu}}^{(k+1)} = \boldsymbol{\mu}^{(k)} + \sigma^{(k)} \boldsymbol{z}^{(k)}$, where $\sigma^{(k)}$ is the step size of the current predictor step.

The predicted solution will lie in the neighborhood of an actual solution, but it needs to be corrected, which is why the corrector step is necessary. In this work the *local parameterization* (Seydel, 2009) is used, where the solution is parameterized along the locally most rapidly changing state, denoted by index $i$,

$$\max \left( \left| \boldsymbol{z}^{(k)} \right| \right) = z_i^{(k)},$$

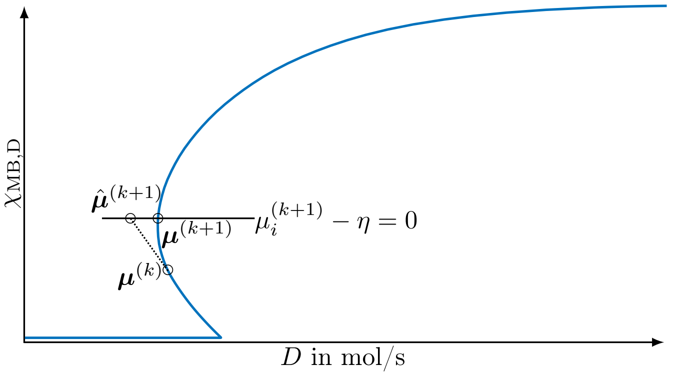

$\chi_{\mathrm{MB,D}}$
$\hat{\boldsymbol{\mu}}^{(k+1)}$
$\boldsymbol{\mu}^{(k+1)}$
$\mu_i^{(k+1)} - \eta = 0$
$\boldsymbol{\mu}^{(k)}$
$D$ in mol/s

Figure 5: Illustration of a predictor-corrector step of the used parameter continuation algorithm

and the parameterized system,

$$\boldsymbol{F}(\boldsymbol{\mu}, \eta) = \begin{pmatrix} \boldsymbol{f}\left(\boldsymbol{\mu}\right) \\ \mu_i - \eta \end{pmatrix} = 0,$$
$$\hat{\mu}_i^{(k+1)} - \left( \hat{\mu}_i^{(k+1)} - \mu_i^{(k)} \right) \sigma^{(k)} = \eta,$$

is solved using the predicted solution $\hat{\boldsymbol{\mu}}^{(k+1)}$ as a starting value for a Newton iteration to generate $\boldsymbol{\mu}^{(k+1)}$. The main tuning parameters of the presented parameter continuation algorithm are the step sizes of the predictor step and the Newton iteration. If they are not chosen properly, the algorithm may not be able to converge.

Using this parameter continuation approach, unstable solution branches can also be calculated in contrast to the dynamic simulation used in the previous section. Depending on the product specifications, the optimal solution may lie in the unstable region and neglecting it would yield false globally optimal solutions. Note that depending on the algorithm and its parameterization, the parameter continuation may generate more points than necessary for fitting the surrogate model. In the present test cases a total number of 1145 sampling points is used.

The mixture investigated is the ternary mixture of toluol, methanol and methylbutyrate presented by Dorn et al. (1998). The mixture is highly non-ideal and has an azeotrope between the heavy boiler toluol and the light boiler methanol. A residue curve map including the azeotrope is shown in Figure 6. The rigorous model is implemented in Matlab, the activity coefficients of the mixture are calculated using a Wilson model. Note, that due to the highly non-ideal phase behavior an approximation of the process using constant relative volatilities is not applicable.

The product specification is 91 % methanol in the distillate stream. This specification lies well above the azeotrope shown in Figure 6, which lies at around 88 % methanol in the distillate stream, but is still far from the highest achievable purity. Therefore the space of valid solutions is rather large.

See Figure 7 for illustrative continuation results. The output multiplicities occur in all three plots. The first two plots show the full hysteresis whereas in the third plot

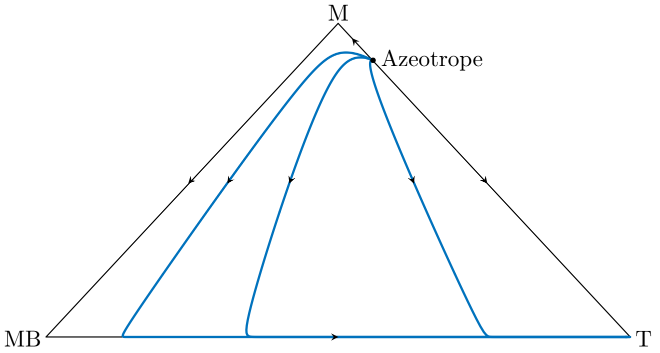


Figure 6: Residue curve map for the mixture methanol (M), methylbutyrate (MB) and toluol (T).

the multiplicities coincide. All relevant solutions lie in the region after the right limit point in the bifurcation diagram is passed. By exploiting this insight the sampling region can be reduced by neglecting the first solution branch, i.e. the region before the right limit point is encountered in the direction of increasing $D$. Thereby the number of output multiplicities reduces from 3 to 2.

Degrees of freedom, $\boldsymbol{x}$, are the vapor flow rate $V \in [2, 13]$ mol/s, the head product flow rate $D \in [1.1, 1.5]$ mol/s and the number of stages in the rectifying and stripping section $l_{\mathrm{r}}$, $l_{\mathrm{s}} \in [2, 25]$. The feed stream is assumed to be $F = 1.8$ mol/s, with a feed composition $\boldsymbol{z}_{\mathrm{feed}} = [0.2806, 0.6566, 0.0628]$ of toluol, methanol and methylbutyrate, respectively. Inputs of the surrogate model are the degrees of freedom, $\boldsymbol{x}$, as well as two mole fractions, $y$, $\chi_{\mathrm{MB,D}}$ and $\chi_{\mathrm{T,D}}$. The outputs of the surrogate model are the mole fraction of the third component $\chi_{\mathrm{M,D}}$ and a logic variable which equals zero if the solution is feasible.

For the sake of simplicity the cost function is the same as in the above case studies.

The kernel function used for fitting the surrogate model is

$$c(\boldsymbol{x}^1, \boldsymbol{x}^2) = 10 \cdot \prod_{i=1}^{m} 1 - \left|\boldsymbol{x}_i^1 - \boldsymbol{x}_i^2\right| .$$

It is chosen because preliminary investigations have shown that it works well for the present case study.

To close the gap between the surrogate and the rigorous model a problem specific line search algorithm is used for the local optimization step, which is used to generate an upper bound for global optimization of each surrogate. Expert knowledge of the process yields the insight that a change in the feed position as well as a change in the vaporization rate have a strong influence on the achievable product purity. If $V$ is increased the turning point is shifted in the direction of decreasing $D$, which results in a purer product stream. The same behavior can be achieved by shifting the feed position in the direction of the bottom of the column. Because shifting the feed position is more cost efficient, this measure is the first thing the algorithm tries. If there is no significant change in product purity anymore $V$ is increased. If a setup yields a purer product than specified, $V$ will be reduced until the specification is reached. An illustration of the influence of the feed position and vaporization rate is shown in Figure 7. Note that the dashed lines do not represent the envelope, but only possible realizations of the column. The specifications used in this case study always give a cost optimal solution in the turning point of the composition profiles of Figure 7. This can be explained with the topology of the bifurcation diagrams in Figure 7. Any increase in column length or vaporization rate corresponds to an increase of operation or investment costs. This will shift the left limit point of the bifurcation diagram to higher methanol concentrations above specification. Therefore at the optimal solution the limit point and the specification have to coincide. To calculate the turning point an underlying parameter continuation has to be used during the local optimization.

The results are presented in Table 7. In the first iteration the surrogate solution and the solution of the local optimization algorithm differ only slightly. The feed position was shifted by one stage and the vaporization rate was slightly increased to fulfill the purity requirement, thereby making the column a little more expensive. In the second iteration the generated surrogate model is already able to capture the rigorous model quite well and there is only a small deviation in the vaporization rate after the local optimization step is finished. Because the $TAC$ was reduced by 2.93% a third iteration is conducted, which yields the same result as the second iteration and the algorithm terminates.

## 6. Conclusion

In this work an approach for the global optimization of distillation columns using explicit and implicit surrogate models was proposed. For the global deterministic optimization state of the art software is used, upper bounds were generated with tailor made local optimization strategies.

To prove the effectiveness of the presented approach, two case studies were conducted. The first case study deals with a five component distillation column and is split into two test settings. In the first test setting ideal phase behavior is assumed. The results show that solutions obtained by the new approach are close to the solutions calculated with a rigorous global optimization approach presented earlier by the authors. The second test setting shows the applicability of the approach to an extended problem with non-ideal phase behavior. Both test cases allow the use of explicit surrogate models because both yield unique stationary solutions.

The second case study deals with a highly non-ideal three component distillation column and is used to show the applicability of the approach under the presence of steady state multiplicities. To cope with the steady state

Table 7: Computation results Toluol column.
Left side: GAMS result - right side: Matlab model

| Iteration | $D$ | $V$ | $l_\mathrm{r}$ | $l_\mathrm{s}$ | $TAC$ | time | $\chi_{\mathrm{M,D}}$ | $\chi_{\mathrm{M,D}}$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 1.29 | 2.25 | 14 | 6 | 115263 | 671 | 0.918 | 0.9 |
| 1* | 1.3 | 2.7 | 15 | 5 | 132210 | 306 | | 0.91 |
| 2 | 1.3 | 2.34 | 19 | 5 | 126495 | 1470 | 0.91 | 0.909 |
| 2* | 1.3 | 2.387 | 19 | 5 | 128340 | 275 | | 0.91 |
| 3 | 1.3 | 2.34 | 19 | 5 | 126497 | 4635 | 0.91 | 0.909 |
| 3* | 1.3 | 2.385 | 19 | 5 | 128340 | 277 | | 0.91 |

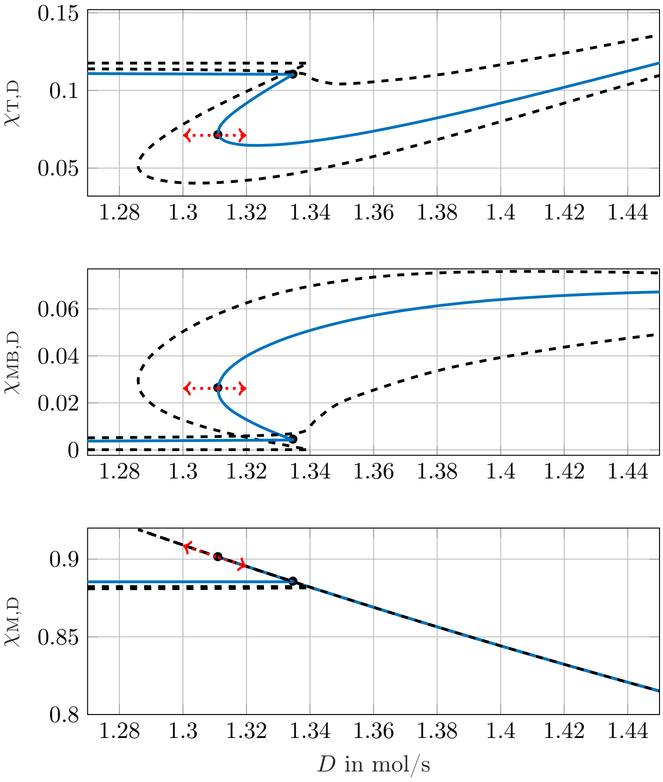


Figure 7: Liquid mole fractions $\chi$ of the three products at the top of the column w.r.t. product stream $D$ generated via parameter continuation. The continuous blue line shows one possible continuation result, where the right black dot denotes the right limit point and the left black dot denotes the left limit point. Red arrows show the influence of $V$ and the feed position. If $V$ is increased or the feed position is shifted downwards along the column the solutions will follow the red arrow pointing to the left, whereas if $V$ is lowered or the feed position is shifted upwards along the column the solutions will follow the red arrow pointing to the right. One example for each direction is shown as a dashed line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

multiplicities in this special case an implicit surrogate formulation is necessary.

With the proposed methods global optimization of highly non-ideal distillation processes becomes feasible within a reasonable time. This is a promising approach if good initial values are not available or difficult to obtain. The latter would be a necessary prerequisite for finding good solutions with established local optimization strategies. Furthermore, established local optimization strategies may fail in the vicinity of bifurcation points, because the Jacobian becomes singular.

Future work will be concerned with more advanced sampling techniques and multi regional search approaches. Furthermore a more generally suitable local optimization technique for the implicit surrogate models and a comparison of the algorithm using different surrogate formulations, e.g. the ALAMO approach, are also of interest.

## Acknowledgement

This work is part of the Collaborative Research Center "Integrated Chemical Processes in Liquid Multiphase Systems - InPROMPT". Financial support by the Deutsche Forschungsgemeinschaft (DFG) is gratefully acknowledged through SFB/TRR 63.

## References

Audet, C., Kokkolaras, M., Le Digabel, S., Talgorn, B., 2018. Order-based error for managing ensembles of surrogates in mesh adaptive direct search. Journal of Global Optimization 70 (3), 645–675.

Ballerstein, M., Kienle, A., Kunde, C., Michaels, D., 2015. Deterministic global optimization of binary hybrid distillation/melt-crystallization processes based on relaxed MINLP formulations. Optimization and Engineering 16, 409–440.

Bennett, K. P., Campbell, C., 2000. Support vector machines: Hype or hallelujah? ACM SIGKDD Explorations 2 (2), 1–13.

Bhosekar, A., Ierapetritou, M., 2018. Advances in surrogate based modeling, feasibility analysis, and optimization: A review. Computers and Chemical Engineering 108, 250–267.

Boukouvala, F., Misener, R., Floudas, C. A., 2016. Global optimization advances in mixed-integer nonlinear programming, minlp, and constrained derivative-free optimization, cdfo. European Journal of Operational Research 252 (3), 701–727.

Caballero, J. A., Grossmann, I. E., 2008. An algorithm for the use of surrogate models in modular flowsheet optimization. American Institute of Chemical Engineers Journal 54 (10), 2633–2650.

Cozad, A., Sahinidis, N. V., Miller, D. C., 2014. Learning surrogate models for simulation-based optimization. American Institute of Chemical Engineers Journal 60, 2211–2227.

Dorn, C., Güttinger, T. E., Wells, G. J., Morari, M., Kienle, A., Klein, E., Gilles, E. D., 1998. Stabilization of an unstable distillation column. Industrial & Engineering Chemistry Research 37, 506–515.

Eason, J. P., Biegler, L. T., 2016. A trust region filter method for glass box/black box optimization. American Institute of Chemical Engineers Journal 62 (9), 3124–3136.

Forrester, A. I. J., Keane, A. J., 2009. Recent advances in surrogate-based optimization. Progress in Aerospace Sciences 45 (1), 50–79.

Gleixner, A., Eifler, L., Gally, T., Gamrath, G., Gemander, P., Gottwald, R. L., Hendel, G., Hojny, C., Koch, T., Miltenberger, M., Müller, B., Pfetsch, M. E., Puchert, C., Rehfeldt, D., Schlösser, F., Serrano, F., Shinano, Y., Viernickel, J. M., Vigerske, S., Weninger, D., Witt, J. T., Witzig, J., 2017. The scip optimization suite 5.0. Tech. Rep. 17-61, ZIB, Takustr.7, 14195 Berlin.

Ibrahim, D., Jobson, M., Li, J., Guillén-Gosálbez, G., 2018. Optimization based design of crude oil distillation units using surrogate column models and a support vector machine. Chemical Engineering Research and Design 134, 212–225.

Keßler, T., Kunde, C., Mertens, N., Michaels, D., Kienle, A., 2019. Global optimization of distillation columns using surrogate models. SN Applied Sciences 1 (11).

Keßler, T., Mertens, N., Kunde, C., Nentwich, C., Michaels, D., Engell, S., Kienle, A., 2017. Efficient global optimization of a novel hydroformylation process. Computer Aided Chemical Engineering 40, 2113–2118.

Kieslich, C. A., Boukouvala, F., Floudas, C. A., 2018. Optimization of black-box problems using smolyak grids and polynomial approximations. Journal of Global Optimization.

Kocis, L., Whiten, W. J., 1997. Computational investigations of low-discrepancy sequences. ACM Transactions on Mathematical Software 23 (2), 266–294.

Locatelli, M., Schoen, F., 2013. Global Optimization. Theory, Algorithms, and Applications. MOS-SIAM Series on Optimization. Cambridge University Press, Cambridge.

McBride, K., Kaiser, N. M., Sundmacher, K., 2017. Integrated reaction–extraction process for the hydroformylation of long-chain alkenes with a homogeneous catalyst. Computers & Chemical Engineering 105, 212–223.

McBride, K., Sundmacher, K., 2015. Data driven conceptual process design for the hydroformylation of 1-dodecene in a thermomorphic solvent system. Industrial & Engineering Chemistry Research 54, 6761–6771.

Mertens, N., Kunde, C., Kienle, A., Michaels, D., 2018. Monotonic reformulation and bound tightening for global optimization of ideal multi-component distillation columns. Optimization and Engineering 19 (2), 479–514.

Nallasivam, U., Shah, V. H., Shenvi, A. A., Huff, J., Tawarmalani, M., Agrawal, R., 2016. Global optimization of multicomponent distillation configurations: 2. enumeration based global minimization algorithm. American Institute of Chemical Engineers Journal 62 (6), 2071–2086.

Nentwich, C., Engell, S., 2016. Application of surrogate models for the optimization and design of chemical processes. In: Proceedings of the International Joint Conference on Neural Networks. IEEE World Congress of Computational Intelligence. pp. 1291–1296.

Papalambros, P. Y., Wilde, D. J., 2000. Principles of Optimal Design: Modeling and Computation, 2nd Edition. Cambridge University Press, Cambridge.

Poling, B. E., Prausnitz, J. M., O'Connell, J. P., 2000. The Properties of Gases and Liquids, 5th Edition. Mcgraw-Hill Professional, New York Citylo.

Quirante, N., Javaloyes, J., Caballero, J. A., 2015. Rigorous design of distillation columns using surrogate models based on Kriging interpolation. American Institute of Chemical Engineers Journal 61 (7), 2169–2187.

Rheinboldt, W. C., 1986. Numerical Analysis of Parametrized Nonlinear Equations. Wiley-Interscience.

Saman, R., Tolson, B. A., Burn, D. H., 2012. Review of surrogate modeling in water resources. Water Resources Research 48 (7).

Schäfer, E., Brunsch, Y., Sadowski, G., Behr, A., 2012. Hydroformylation of 1-dodecene in the thermomorphic solvent system dimethylformamide/decane. phase behavior-reaction performance-catalyst recycling. Industrial & Engineering Chemistry Research 51, 10296–10306.

Scheuerer, M., Schaback, R., Schlather, M., 2013. Interpolation of spatial data – a stochastic or a deterministic problem? European Journal of Applied Mathematics 24 (4), 601–629.

Seydel, R., 2009. Practical Bifurcation and Stability Analysis. Springer.

Steimel, J., Harrmann, M., Schembecker, G., Engell, S., 2013. Model-based conceptual design and optimization tool support for the early stage development of chemical processes under uncertainty. Computers & Chemical Engineering 59, 63–73.

Stephenson, J., Gallagher, K., Holmes, C. C., January 2004. Beyond kriging: dealing with discontinuous spatial data fields using adaptive prior information and bayesian partition modelling. Special Publications 239, 195–209.

Tawarmalani, M., Sahinidis, N. V., 2005. A polyhedral branch-and-cut approach to global optimization. Mathematical Programming 103 (2), 225–249.

Weidlich, U., Gmehling, J., 1987. A modified unifac model. 1. prediction of vle, h$^e$ and $\gamma^\infty$. Industrial & Engineering Chemistry Research 26 (7), 1372–1381.

Zhang, Y., Sahinidis, N. V., 2013. Uncertainty quantification in co2 sequestration using surrogate models from polynomial chaos expansion. Industrial & Engineering Chemistry Research 52 (9), 3121–3132.