A New Termination Criterion for Sampling for Surrogate Model Generation using Partial Least Squares Regression

Julian Straus, Sigurd Skogestad

Department of Chemical Engineering, Norwegian Univ. of Science & Technology (NTNU), NO-7491 Trondheim

Abstract

This paper proposes a new incremental sampling method for the generation of surrogate models based on the application of partial least squares regression (PLSR) as a termination criterion. Compared to existing incremental and adaptive methods, the proposed method allows the sampling algorithm to stop without needing to fit a surrogate model at each iteration step. The proposed procedure was applied to a motivating pipe model and two case studies; the reaction and the separation section of an ammonia synthesis loop. In all cases, the new sampling method allows a small number of sampling points, corresponding to a regular grid with less than two points in each independent variable. The two surrogate models of the ammonia loop are combined for overall optimization. The optimum for the combined surrogate models is close to the optimum obtained with the original model.

Keywords: Partial Least Squares Regression, Iterative Sampling for Surrogate Model, Optimization of Integrated Processes

1. Introduction

Surrogate models, frequently called response surfaces or reduced-order models, are emerging as an engineering tool with many applications (Forrester et al., 2008). They are simplified mathematical representations of complex models. Their application reduces the computational cost. Queipo et al. (2005) provide an extensive review of surrogate-based analysis and optimization, with a focus on aerospace systems. The even more complex models used in process systems engineering has sparked the interest for the application of surrogate models also in this field. Bhosekar and Ierapetritou (2018) give a detailed overview of the application of surrogate models in process systems engineering. One application is multi-scale modeling (Biegler et al., 2014; Karolius et al., 2016). Surrogate models are in this approach, for example, used to include computational fluid dynamics. A second emerging field for surrogate models is process optimization using black-box models (Caballero and Grossmann, 2008; Eason and Biegler, 2016; Forrester and Keane, 2009; Grimstad et al., 2016; Quirante and Caballero, 2016). Commercial process simulators generally do not provide derivative information. This reduces their applicability in optimization. However, the fitting of surrogate models allows for the use in derivative-based optimization algorithms. Boukouvala et al. (2016) extensively discuss the application of surrogate models in constrained derivative-free optimization (CDFO) and draw a connection between CDFO and mixed integer nonlinear programming (MINLP).

In the field of optimization, surrogate models can be directly integrated into the optimization routine. The SOMI framework developed by Müller et al. (2013) is used for solving expensive MINLP problems. Similarly, the ARGONAUT algorithm developed by Boukouvala and Floudas (2017) incorporates grey-box surrogate modelling and couples it with global optimization for nonlinear problems. It was further improved through introduction of parallel computing for sampling and applied to both optimization of energy systems (Beykal et al., 2018a) and oil-field operation (Beykal et al., 2018b). Kieslich et al. (2018) utilized Smolyak (sparse) grids and combined them with Chebyshev polynomials for the optimization of black-box functions. Their approach combines surrogate model generation and optimization and utilizes bound refinement for improved accuracy.

The performance of surrogate models is influenced by two factors. First, the chosen basis functions for the surrogate model affect the achievable accuracy of the surrogate model to represent the nonlinear response surface. Common basis functions include B-splines (Grimstad et al., 2015), Kriging models (Krige, 1951; Caballero and Grossmann, 2008; Quirante and Caballero, 2016; Eason and Biegler, 2016), individual chosen basis function in the ALAMO approach (Cozad et al., 2014, 2015; Wilson and Sahinidis, 2017), and artificial neural networks (Eason and Cremaschi, 2014). Davis et al. (2017) provide an overview of the different methods and compare their performance on

[☆]J.S. gratefully acknowledges the financial support from YARA International ASA.

Email address: sigurd.skogestad@ntnu.no (Sigurd Skogestad)

Abbreviations

PLSR	Partial least squares regression
RMSE	Root-mean-squared error

Variables

ΔW	Difference of the significant weights
n_{add}	Number of additional sampling points
n_{ini}	Number of initial sampling points
n_f	Number of sampling steps for averaging
n_p	Number of sampling points
n_s	Number of significant latent variables
n_u	Number of independent variables
n_y	Number of dependent variables
$\mathbf{U}$	Sample set
$\mathbf{U}'$	Sample set in latent variables
s	Sample standard deviation
$\mathbf{W}$	Weights from PLSR
$\mathbf{W}_S$	Significant weights from PLSR
$\mathbf{w}_i$	Weight of latent variable i from PLSR
$\mathbf{Y}$	Response set

Greek variables

β	Threshold for significant latent variables, see (8)
γ	Threshold for termination of sampling, see (10)
$\boldsymbol{\epsilon}$	Model fit error with surrogate model, see (15)

Superscripts

k	Iteration in the sampling algorithm

Subscripts

i	Index of latent variable i
j	Index of dependent variable j
m	Index of the validation space

47 challenge functions.

The sampling method is the second major influence on the performance of the surrogate model. In addition to the fitting of the surrogate model, the sampling of points from the detailed model is the main computational cost. Hence, the aim of sampling is to sample as few points as possible while achieving satisfactory accuracy of the surrogate model. The overall concept is called *design of computer experiments*. Garud et al. (2017b) provide an extensive review of the different sampling approaches. They can be differentiated between 1. predefined (static) 2. incremental, and 3. adaptive sampling. In the first sampling method, the sampling points are generated and sampled in one iteration. In the second sampling method, points are added incrementally and surrogate models are usually fitted in each iteration step until satisfactory performance is achieved. The third sampling methods use the surrogate model fit to also decide on placement of the new sampling points.

Predefined (static) sampling is the simplest approach. Monte Carlo sampling (Metropolis and Ulam, 1949) is an early method based on pseudo-random numbers. The key idea of Monte Carlo sampling is that the randomness in sampling will result in space filling. This is however not guaranteed and may require a large number of sampling points n_p.

Hence, space-filling methods are frequently considered instead. The simplest space-filling method is regular grid sampling. It is applied for surrogate modeling (Grimstad et al., 2016), but it has an exponential increase in sampling points,

$$n_p = n_g^{n_u} \tag{1}$$

where n_g is the number of points per dimension in the regular grid. Therefore it is only useful for a small number of independent variables n_u.

Several other methods have been developed to overcome this *curse of dimensionality*. Latin hypercube sampling (LHS) (McKay et al., 1979) is probably the most popular method today. It is applied by *e.g.* Ochoa-Estopier et al. (2014) for a heat-integrated crude oil distillation system for $n_u = 10$ and $n_p = 3000$ which corresponds to $n_g = 2.3$ in a regular grid. LHS may, however, not explore the whole space as shown for a simple 2-dimensional case study by Garud et al. (2017b).

Independent of the chosen sampling method, static approaches have in addition the inherent problem of selecting how many points to sample. Another approach for overcoming the *curse of dimensionality* are sparse grids. Sparse grids, as applied by Kieslich et al. (2018) for surrogate-based optimization, reduce the the number of sampling points to $n_g \log(n_g)^{n_u - 1}$. Bungartz and Griebel (2004) give an extensive review of sparse grids whereas Pflüger et al. (2010) extend the concepts to high dimensional data. As it is not known *a-priori* how many sample points are needed for a desired surrogate model accuracy, both under- and oversampling can occur. Especially oversampling can result in increased computational cost due to the sampling of unnecessary points.

The problem with oversampling can be alleviated by incremental sampling. Nuchitprasittichai and Cremaschi (2013) use such an incremental approach based on LHS. Surrogate models are fitted after each additional sampling step and the procedure is stopped upon reaching a termination criterion based on boots trapping. Quirante and Caballero (2016) use the maxmin approach in which the points are placed so that the minimum distance between sampling points is maximized. Depending on the performance of the surrogate model, more points are sampled, again using the maxmin approach.

Adaptive sampling methods were developed as an improvement to incremental approaches. They are generally based on two concepts, exploration and exploitation (Garud et al., 2017b). The former tries to achieve point placement in regions which are poorly represented in the sampling space. This is similar to the incremental approaches. However, adaptive approaches utilize the surrogate fit for identifying highly nonlinear regions. Correspondingly, new points are placed in these nonlinear regions. The smart

sampling algorithm developed by Garud and co-workers is one of the adaptive sampling methods (Garud et al., 2017a, 2018). Through the application of two metrics, one for exploitation and one for exploration, they identify new optimal points. Cozad et al. (2014) developed a combined surrogate model fitting and sampling algorithm which aims at sampling points which have a maximum error with the surrogate model. The resulting surrogate models have a simple structure allowing an easy calculation of derivatives. Eason and Cremaschi (2014) combined space filling through incremental LHS with exploitation through jackknifing.

The need for repeated fitting of a surrogate model in both incremental and adaptive approaches can be computationally expensive. The development of a termination criterion for incremental sampling without the need of fitting a surrogate model is attractive, and is the focus of this paper. One possible approach is to apply partial least squares regression (PLSR), which has a very low computational cost, and use this as a termination criterion.

PLSR is a method from chemometrics, developed for the analysis of high-dimensional data. It was previously applied in the calculation of surrogate models (Straus and Skogestad, 2017a,b) to reduce the number of independent variables n_u in the fitting through the introduction of latent variables. The new latent variables $\mathbf{u}'$ were calculated using the weights $\mathbf{W}$ given by PLSR. In this paper, we use this information instead as a termination criterion for sampling, without the need to fit a surrogate model.

This paper is structured as follows; Section 2 first explains the main properties of partial least squares regression (PLSR). Section 3 describes the procedure for sampling for surrogate model generation without the necessity of fitting a surrogate model in each iteration. Section 4 illustrates the steps in the procedure using a simple pipe model as motivating example. Section 5 applies this method to two case studies, the reaction and the separation sections of a simplified ammonia synthesis loop. These two submodels are then combined with the original synthesis gas makeup section for the respective submodels and evaluated in comparison to the original model. Section 6 then discusses the properties of the proposed method.

2. Background - partial least squares regression

Partial least squares regression (PLSR) is a linear regression tool widely used in chemometrics (*e.g.* Wold et al. (2001)). It also has many other application, for example in the analysis of high-dimensional genomic data (Boulesteix and Strimmer, 2007). In many applications, the number of independent variables n_u, *e.g.* spectroscopy frequencies and genes, exceed the number of samples n_p, which results in problems with classical multivariate regression models. Furthermore, problems may arise in the multivariate regression if independent variables are noisy or strongly correlated. A detailed review of PLSR can be found in (Boulesteix and Strimmer, 2007) and (Wold et al., 2001). The former explains various algorithms for the calculation of the latent variables.

The aim of PLSR is a variable reduction in the independent variables resulting in new latent variables. PLS may also mean projection to latent structures (Wold et al., 2001). PLSR is similar to principal component regression (PCR) (*e.g.* Martens (2001)). However, in contrast to PCR, it considers in the calculation of the latent variables their impact on the dependent variables. The variable reduction is given through the transformation of the original independent variable space $\mathbf{U} \in \mathbb{R}^{n_p \times n_u}$ into a space of n_c latent variables $\mathbf{U}' \in \mathbb{R}^{n_p \times n_c}$.

$$\mathbf{U}' = \mathbf{U}\mathbf{W} \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{n_u \times n_c}$. In PLSR, $\mathbf{W}$ is calculated to maximize the covariance between $\mathbf{U}'$ and the dependent variable space $\mathbf{Y}' \in \mathbb{R}^{n_p \times n_c}$.

Several algorithms exist for computing $\mathbf{W}$. An overview is given by Boulesteix and Strimmer (2007). In this paper, we use the *Statistically Inspired Modification of PLS* algorithm (SIMPLS) (de Jong, 1993), which obtains the weights for each component $i = 1, \ldots, n_c$ sequentially according to

$$\mathbf{w}_i = \arg\max_{\mathbf{w}} \mathbf{w}^\mathsf{T}\mathbf{U}^\mathsf{T}\mathbf{Y}\mathbf{Y}^\mathsf{T}\mathbf{U}\mathbf{w} \tag{3}$$

with the following constraints

$$\begin{aligned} \mathbf{w}_i^\mathsf{T}\mathbf{w}_i &= 1 \\ \mathbf{w}_i^\mathsf{T}\mathbf{U}^\mathsf{T}\mathbf{U}\mathbf{w}_j &= 0 \qquad \forall j = 1, \ldots, i-1 \end{aligned} \tag{4}$$

$\mathbf{w}_i$ denotes the columns of the weight matrix $\mathbf{W}$. It gives the coefficients of the original variables in the calculation of the new latent variables. The first constraint normalizes the weights, whereas the second constraint results in orthogonality of the latent variables.

Depending on the implemented algorithm (*e.g. plsregress* in MATLAB and *simpls* in R (Boulesteix and Strimmer, 2007)), $\mathbf{u}'_i$ corresponding to a column of $\mathbf{U}'$ may have a length of 1, *i.e.*

$$\mathbf{u}_i'^\mathsf{T}\mathbf{u}'_i = 1 \tag{5}$$

This is contrary to the constraints (4). The proposed method however utilizes weights $\mathbf{w}_i$ with unit length. Hence, it requires the transformation of the weights $\mathbf{w}_i$ to have unit length.

PLSR is sensitive to scaling (Wold et al., 1983). The standard score gives equal variance for each independent variable and is given by

$$\mathbf{U}_{scaled} = (\mathbf{U} - \boldsymbol{\mu}_\mathbf{U}) \circ \boldsymbol{\sigma}_\mathbf{U}^{-1} \tag{6}$$

The operator $\circ$ corresponds to the Schur product which is element-wise multiplication. $\boldsymbol{\mu}_\mathbf{U}$ is the mean value and $\boldsymbol{\sigma}_\mathbf{U}$ the standard deviation in the matrix $\mathbf{U}$ with respect to each of the independent variables $\mathbf{u}$. This scaling was found to improve the performance when PLSR is used for independent variable reduction (Straus and Skogestad, 2017b) and will be applied in the sampling procedure as well.

3. Proposed sampling procedure utilizing PLSR

The idea is to compute the weight matrix $\mathbf{W}$ after each sampling, or after a block of n_{add} samplings, and consider the convergence of $\mathbf{W}^k$. This may be done by monitoring the difference

$$\Delta\mathbf{W}^k = \mathbf{W}^k - \mathbf{W}^{k-1} \tag{7}$$

at iteration k. The norm, $\left\|\Delta\mathbf{W}^k\right\|$, can then be utilized as termination criterion for the sampling procedure. However, $\mathbf{W}^k$ should only include the weights corresponding to the significant latent variables

$$\mathbf{W}_s^k = \begin{bmatrix}\mathbf{w}_1^k & \cdots & \mathbf{w}_{n_s}^k\end{bmatrix} \tag{8}$$

where n_s is the number of significant weights as defined by the threshold β. The first omitted weight vector $\mathbf{w}_{n_s+1}^k$ explains less than β % of the variance of the dependent variable y.

The initialization of the procedure consists of sampling n_{ini} points. The sampling of the initial points can be performed using any method, *e.g.* Latin hypercube sampling (McKay et al., 1979) or Sobol sampling (Sobol, 1967). PLSR is then applied to calculate the initial weights $\mathbf{W}_s^1$. In the subsequent iterative procedure, n_{add} points are sampled at each iteration step k. This corresponds to a so-called *arithmetic sampling*, as defined by Provost et al. (1999), and can be written as

$$n_p(k) = n_{ini} + k \cdot n_{add} \tag{9}$$

Similar to the initially sampled points, any sampling method can be used for the additional sampling points for the incremental sampling. The additional points n_{add} should however be sampled using the same sampling method as the initial sampling points n_{ini}. When using Latin hypercube sampling, it is possible to either augment the existing Latin hypercube, that is considering the previous points so that the new sampling set is in itself a Latin hypercube, or sample n_{add} new additional sampling points which form a Latin hypercube. If the original Latin hypercube is not augmented, then the resulting set is not necessarily a Latin hypercube.

We use the Frobenius norm, and monitor $\left\|\Delta\mathbf{W}_s^k\right\|_F$ as the incremental sampling progresses. The reason behind choosing the Frobenius norm is discussed in the discussion in Section 6. Although we found that the norm eventually converges to a fixed value, it can temporarily increase and decrease. This *noise* may terminate the procedure before reaching a satisfactory accuracy. To avoid a preemptive termination, we propose to average the norm of the last n_f steps resulting in the calculation of the averaged norm

$$\left\|\Delta\mathbf{W}_s^k\right\|_F^{av} = \frac{\sum_{l=k-n_f+1}^{k} \left\|\Delta\mathbf{W}_s^l\right\|_F}{n_f} \tag{10}$$

Algorithm 1 Sampling procedure.

1: For a given subprocess $\mathbf{g}$ with independent variables $\mathbf{u} \in \mathbb{R}^{n_u}$ and dependent variables $\mathbf{y} \in \mathbb{R}^{n_y}$, define upper and lower bounds for the independent variables.
2: Sample n_{ini} initial points.
3: Select the threshold β and calculate $\mathbf{W}_s^1$ according to Eq. (8).
4: Initialize with $k = 1$.
5: **while** $\left\|\Delta\mathbf{W}_s^k\right\|_F^{av} > \gamma$ **do**
6: Sample n_{add} additional points.
7: Scale the sampled space using the standard score (6).
8: Perform PLS regression.
9: Obtain the number of significant weights n_s using the selected β and calculate $\Delta\mathbf{W}_s^k$ according to Eqs. (7) and (8).
10: Calculate the averaged norm $\left\|\Delta\mathbf{W}_s^k\right\|_F^{av}$ in Eq. (10).
11: Set the iteration number $k = k + 1$.
12: **end while**
13: Fit the surrogate models.

The averaged norm is compared to a threshold γ and, if it is below γ, the iterative procedure is stopped and a surrogate model is fitted to the sampling space $\mathbf{U}$.

It is important to mention that the surrogate models are not fitted to the set of latent variables $\mathbf{U}'$ obtained *via* PLSR but to the set of original independent variables $\mathbf{U}$. Hence, the variable transformation is only used in the sampling itself for the calculation of the termination criterion and not in the surrogate model fitting or the application of the surrogate models. The advantage of using the independent variable transformation is then given by the calculation of a termination criteria without the necessity of fitting a nonlinear surrogate model. Furthermore, the number of significant weights does not influence the used sampling points for fitting of the surrogate model. All sampled points are used for the fitting of the surrogate model.

Algorithm 1 summarizes the procedure. In the case of multiple dependent variables $\mathbf{y}$, it is either possible to perform PLS regression for all dependent variables independently or simultaneously. The former is computationally more expensive, albeit only marginally. If the latent variables are used to fit the surrogate model, we found earlier that it is best to perform PLS regression independently (Straus and Skogestad, 2017b). However, here we are looking at the differences and do not use the latent variables for the fit of the surrogate model. We therefore use the simultaneous approach. This will be further discussed in the case studies in Section 5.

4. Description of the sampling procedure

4.1. Motivating example - pipe model

The sampling procedure is now explained in detail using the pressure drop over a an isothermal pipe as a moti-

vating example. The independent variables $\mathbf{u}$ are the inlet pressure p_{in}, the temperature T, and the molar flows $\dot{n}_i$. The dependent variable y is the pressure drop. The model is

$$0 = \left(p_{in}^2 - p_{out}^2\right) - 4f\frac{L}{D}\frac{RT\overline{M}}{A^2}\dot{n}^2 \tag{11}$$

This model allows for changing the number of independent variables n_u through changing the number of gas components n_{gas} in the stream. These influence the average molar mass

$$\overline{M} = \frac{\sum_{i=1}^{n_{gas}} M_i\dot{n}_i}{\dot{n}} \tag{12}$$

and the total flow

$$\dot{n} = \sum_{i=1}^{n_{gas}} \dot{n}_i \tag{13}$$

The investigated case has 5 gas components ($i = H_2$, N_2, NH_3, Ar, and CH_4) resulting in $n_u = 7$. One surrogate model has to be fitted for the pressure difference $y = p_{in} - p_{out}$ ($n_y = 1$). The points were sampled using $\mathbf{u} = \begin{bmatrix} p_{in} & T & \dot{\mathbf{n}}^\mathsf{T} \end{bmatrix}^\mathsf{T}$. We found previously (Straus and Skogestad, 2017a) that it is beneficial to use intensive variables for PLS regression. Hence, molar fractions x_i are used as independent variables in the fitting of the surrogate model and calculation of the PLSR weights. The data of the pipe are given in Table 1. The nominal value and the bounds (lower and upper bound) of the sampling domain can be found in Table 2. Table 3 gives the parameters for the sampling procedure (Algorithm 1), including the parameters for choosing significant weights ($\beta = 2$ %) and for terminating the sampling ($\gamma = 0.05$).

4.2. Evaluation of the norm of the weights

We only include the significant weights $\mathbf{w}_i$ in $\mathbf{W}_s$, see Eq. (8). To understand this better, Figure 1 shows the convergence of all the seven weights $\mathbf{w}_i$ for an increasing sampling space $n_p(k)$. Note the log scale for the norm. For illustration purposes, we oversample using 5000 points sampled as a Latin hypercube. The 5000 sample points were obtained by sampling $n_{ini} = 25$ sample points and subsequently augmenting the Latin hypercube by $n_{add} = 5$ sampling points, that is, we obtain a new Latin hypercube in each iteration. PLSR was performed every 5 sampling points ($n_{add} = 5$) after initialization with 25 sampling points. The last 5 calculated norms were used for averaging in (10) ($n_f = 5$). The colour code shows the three significant weights, $\mathbf{w}_1$, $\mathbf{w}_2$, and $\mathbf{w}_3$, (black) and the four insignificant weights with an explained variance less than $\beta = 2$ % (red). As we can see, all weights are converging. However, it is possible to see a clear difference between the significant and insignificant weights. $\mathbf{w}_1$ and $\mathbf{w}_2$ are similar in convergence and hard to distinguish. The third significant weight $\mathbf{w}_3$ converges at a slightly slower rate and has a value in-between the significant and insignificant weights. The insignificant weights converge at a much slower rate. Especially $\mathbf{w}_5$, $\mathbf{w}_6$, and $\mathbf{w}_7$ experience frequent changes in the norm resulting in noisy bumps, even with the applied filtering. This is especially pronounced in the close-up of the first 1000 point in Figure 1 b).

It has to be noted that the number of significant weights n_s (with $\beta \geq 2$ %) decreases with increasing n_p in this case study. Initially, $\mathbf{w}_{4,k}$ explains between 2 % and 4 % of the variance in y, so $n_s = 4$. However, it settles to around 0.5 % after around 300 sampled points, giving $n_s = 3$. As a result, the number of significant weights n_s can change in the course of the sampling.

Figure 2 shows how the important combined averaged norm of the change of significant weights $\left\|\Delta\mathbf{W}^k\right\|_F^{av}$ develops for the first 1000 sampling points, but here using a linear scale for the norm. As we can see, the reduction in the norm is especially pronounced in the first 100 to 150 sampling points and is less pronounced with increasing sampling points. This threshold $\gamma = 0.05$ is reached after 230 sampling points. The norm of the change of the combined significant weights, $\left\|\Delta\mathbf{W}_s^k\right\|_F^{av}$, is less susceptible to the *noise* in the calculation compared to the individual weights shown in Figure 1. Hence, it is not necessary to use a large n_f for averaging.

4.3. Error of the surrogate model

We found in Figure 2 that the significant weights $\mathbf{W}_s^k$ converge after about 200-300 sampling points. How does this reduction correspond to the accuracy of a fitted surrogate model?

To this end, we investigate the correlation between the norm of the difference, $\left\|\Delta\mathbf{W}_s^k\right\|_F^{av}$, and the accuracy of the surrogate model. The surrogate model structure is a 2-layer cascade forward neural network with 5 hidden neurons in each layer. The surrogate models were fitted after each 5 additional points starting at initially 25 points. After 100 sampled points, the interval is increased to every 25 points and to every 100 points after 1000 sampled points. Each time, 10 neural networks were fitted to aver-

Table 1: Parameters for the pipe example.

Parameter	L/D [-]	A [m^2]	f [-]
Value	8.8×10^4	0.2	0.003

Table 2: Upper and lower bounds and the nominal value of the independent variables ($\mathbf{u}$) (pipe example).

Variable	Unit	Nominal Value	Lower Bound	Upper Bound
p_{in}	[bar]	23	27	31
T	[°C]	0	10	20
$\dot{n}_{H_2,in}$	[mol/s]	700	1400	2100
$\dot{n}_{N_2,in}$	[mol/s]	230	460	690
$\dot{n}_{NH_3,in}$	[mol/s]	50	100	150
$\dot{n}_{Ar,in}$	[mol/s]	10	20	30
$\dot{n}_{CH_4,in}$	[mol/s]	10	20	30

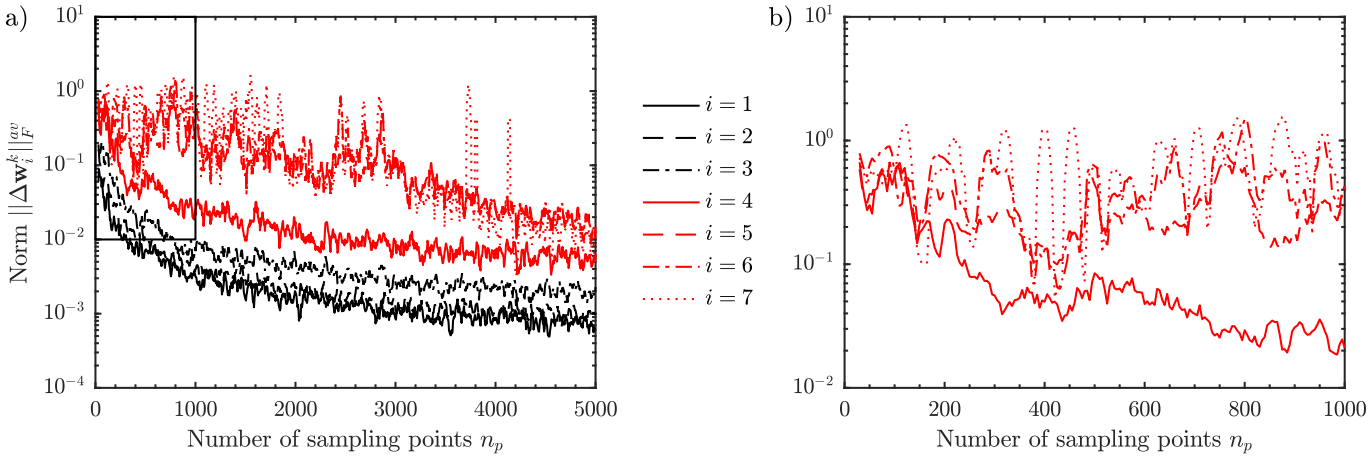

Figure 1: Development of the norm of the change of a) all individual weights, $||\Delta \mathbf{w}_i^k||_F^{av}$ for $i = 1 \dots n_u$ (significant weights $\mathbf{w}_1$, $\mathbf{w}_2$, and $\mathbf{w}_3$ in black, insignificant weights in red), and b) a close-up of the insignificant weights, $||\Delta \mathbf{w}_i^k||_F^{av}$ for $i = n_s + 1 \dots n_u$ for the first 1000 sampling points (pipe example).

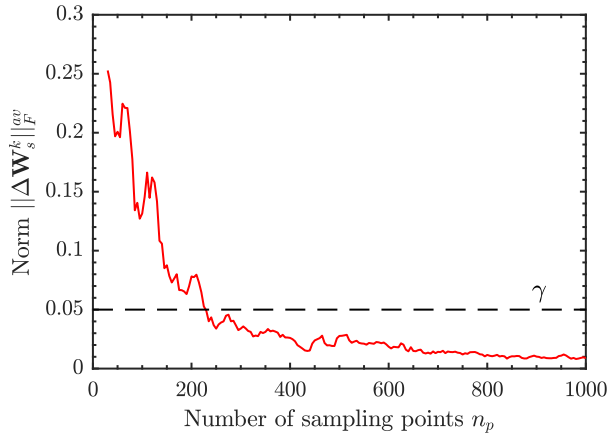

Figure 2: Development of the averaged norm of the change in the combined significant weights, $||\Delta \mathbf{W}_s^k||_F^{av}$ (pipe example).

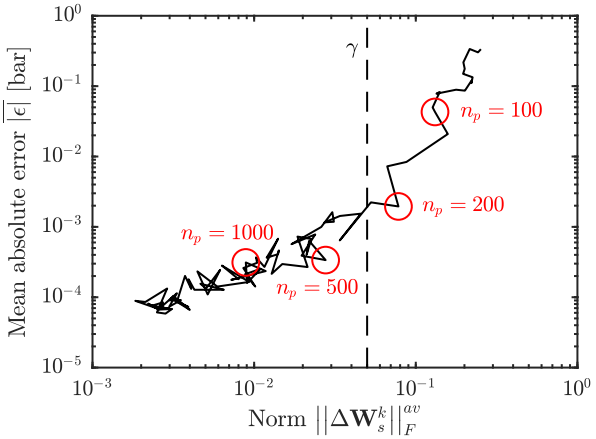

Figure 3: Mean absolute error of the surrogate model $\overline{|\epsilon|}$ as a function of the averaged Frobenius norm of the change in the combined significant weights (pipe example).

age the randomness in the initial seed to the neural networks. The dependent variable of the surrogate model fit ($y = p_{in} - p_{out}$) is then calculated as the average of the resulting 10 neural networks. The validation space is given by 10^4 randomly sampled points. Note that the neural networks were not fitted to the latent variables $\mathbf{u}'$, but to the initial independent variables $\mathbf{u}$. This is different to the results reported in (Straus and Skogestad, 2017a) and (Straus and Skogestad, 2017b).

Figure 3 shows the mean absolute error $\overline{|\epsilon|}$ of the pressure difference $y = p_{in} - p_{out}$ as a function of $\left\|\Delta \mathbf{W}_s^k\right\|_F^{av}$. Here, ϵ is the difference between the exact value for y and the one obtained from the surrogate model (*i.e.* the neural network). The threshold $\gamma = 0.05$ used in the previous section is also indicated. From this figure, where we used log-scale for $\overline{|\epsilon|}$, we see that sampling more than 1000 points does not reduce the error further. Increasing the sampling space above $n_p \approx 300 - 500$ only marginally reduces the error in the fitted neural network. This corresponds to the concept of learning curves as described by Provost et al. (1999), which says that an increase in sampling points does not improve the accuracy of the surrogate model. The threshold γ corresponds to the point in which the decrease in the averaged norm $\left\|\Delta \mathbf{W}_s^k\right\|_F^{av}$ in Figure 2 flattens and is at

$$\left\|\Delta \mathbf{W}_s^k\right\|_F^{av} \approx 0.02\text{-}0.05 \tag{14}$$

Since we want to avoid the fitting of the surrogate model (neural networks) during the sampling, this can be used as the termination criterion in the sampling for surrogate

Table 3: Tuning parameters of the proposed sampling procedure (all examples and case studies).

Parameter	n_{ini}	n_{add}	n_f	β	γ
Value	25	5	5	2 %	$n_{add} \times 10^{-2}$

model generation, that is, γ should be between 0.02 and 0.05.

4.4. Results of the applied sampling procedure

The above results were based on an augmented, oversampled Latin hypercube with 5000 points. The application of the method with the tuning parameters given in Table 3 ($\gamma = 0.05$) and the proposed incremental Latin hypercube sampling results in a termination after 210 sampled points. This is similar to the previous oversampling shown in Figure 2, where the threshold γ is crossed after 230 sampling points. The resulting surrogate model shows a maximum absolute error $|\epsilon|_{\max} = 0.045$ bar and an average absolute error $\bar{\epsilon} = 5 \times 10^{-4}$ bar. The 3 significant weights explain 94.60 % of the variance in the dependent variable $y = p_{in} - p_{out}$. All 7 weights explain in total only 94.92 % of the variance in the dependent variable due to the nonlinearity of the pipe model. Consequently, the 4 insignificant weights explain combined only 0.32 % of the variance in y. The relatively high maximum absolute error is caused by neglecting the corner points of the independent variables, *i.e.* the points given by constructing a 2-point regular grid using the bounds in Table 2. Hence, the surrogate model is extrapolating close to the corners. With $n_u = 7$, it would be possible to incorporate the corner points as they only correspond to $2^7 = 128$ points. However, if $n_u > 10$, the incorporation of the corner points would require a large number of sampled points. In this situation, it is best to apply the surrogate model first. If necessary, it is then possible to add only the corner points in which the subsequent application of the surrogate model is moving. This reduces the points which one has to sample.

5. Ammonia synthesis loop case studies

So far, the method was applied to an example with $n_y = 1$. Now, two additional case studies are used for testing the sampling procedure with $n_y > 1$ and to evaluate whether similar conclusion can be drawn. Both case studies are part of the ammonia synthesis loop shown in Figure 4. The first case study is the reaction section (marked red), as previously described in Straus and Skogestad (2017a,b). The second case study is the separation section (marked green) of the same synthesis loop.

The error of the dependent variable j is given by

$$\boldsymbol{\epsilon}_j = \mathbf{y}_{surr,j} - \mathbf{y}_{val,j} \tag{15}$$

in which $\mathbf{y}_{val,j}$ corresponds to the exact values from the detailed model and $\mathbf{y}_{surr,j}$ to the value given by the surrogate model. The maximum absolute error $|\boldsymbol{\epsilon}|_{\max}$ and the root-mean-squared error (RMSE)

$$\mathrm{RMSE}_j = \frac{\sum_{m=1}^{n_{val}} \left(y_{surr,j,m} - y_{val,j,m}\right)^2}{n_{val}} \tag{16}$$

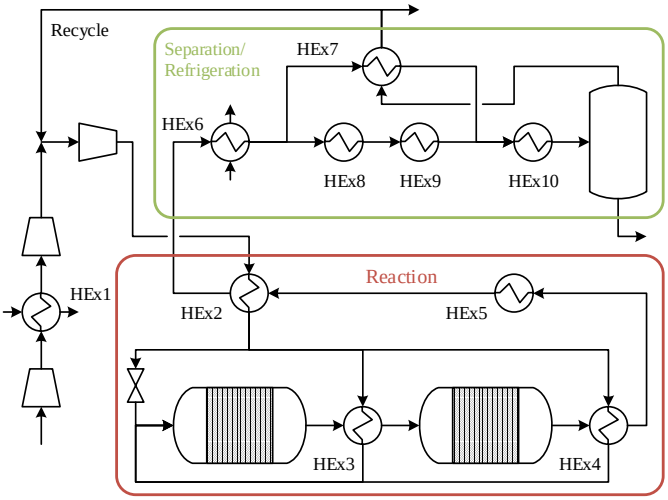

Figure 4: Ammonia synthesis loop with the submodels *Reaction Section* and *Separation Section*.

are used to assess the performance of the surrogate models. In addition, the relative error is calculated using the range of the dependent variables of the validation space, $\mathbf{y}_{val}$, *i.e.*

$$\boldsymbol{\epsilon}_{j,rel} = \frac{\boldsymbol{\epsilon}_j}{\max \mathbf{y}_{val,j} - \min \mathbf{y}_{val,j}} \tag{17}$$

5.1. Case study 1: Reaction section of an ammonia synthesis loop

The reaction section of the ammonia synthesis loop was previously applied in the introduction of new latent variables $\mathbf{u}'$ (Straus and Skogestad, 2017a,b). It is interconnected to the compressor train and the separation section through the overall mass recycle. It consists of two consecutive reactor beds with interstage heat integration (HEx3). Furthermore, the reaction heat is used for the generation of high pressure stream (HEx5) and heating the inlet flow to the first bed (HEx2 and HEx4). It is shown in Figure 4.

5.1.1. Model description

The model has 10 independent variables ($\mathbf{u}$): the inlet pressure p_{in}, the inlet temperature T_{in}, 5 inlet molar flows $\dot{n}_{i,in}$ (H_2, N_2, NH_3, Ar, and CH_4), 2 split ratios, and the outlet temperature of the steam generation heat exchanger 5, $T_{HEx5,out}$. There are 4 dependent variables ($\mathbf{y}$): the pressure drop Δp [mbar], the temperature change ΔT [mK], the extent of reaction $\dot{\xi}$ [mol/s], and the duty of heat exchanger 5, Q_{HEx5} [kW]. We use a "grey-box" model by introducing exact mass balances using $\dot{\xi}$ and the stoichometric coefficients ν_i

$$\dot{n}_{i,out} = \dot{n}_{i,in} + \nu_i \dot{\xi} \tag{18}$$

In our previous work, a two-point regular grid and 5000 points defining a Latin hypercube were used (Straus and Skogestad, 2017a,b). This resulted in reasonable errors for the dependent variables Δp, ΔT, and $\dot{\xi}$ through the introduction of latent variables $\mathbf{u}'$. We want to see if we

can use fewer points, even with the heat duty of the heat exchanger (Q_{HEx5}) as a new dependent variable. A two-point regular grid corresponds to $2^{10} = 1024$ sampling points, but we want to see if we can terminate the sampling with even fewer points.

The upper and lower bounds of the parameters in the sampling domain can be found in Table 4. The feed mole fractions x_i and the total molar flow $\dot{n}_{in}$ are used as independent variables in the surrogate model generation and application of PLSR instead of the molar flows $\dot{n}_{i,in}$ (Straus and Skogestad, 2017a). This requires omitting the mole fraction of hydrogen. Furthermore, the molar ratio H_2/N_2 is used instead of the mole fraction of nitrogen as independent variable in surrogate model fitting.

The surrogate model structure is a two-layer cascade forward neural network with 5 hidden neurons in each layer. The validation space consists of $n_{val} = 10^4$ randomly sampled points.

5.1.2. Results

The data for the proposed sampling procedure are the same as in the pipe case study, see Table 3. PLSR was applied to all dependent variables simultaneously. With augmented Latin hypercube sampling and $\gamma = 0.05$, the proposed sampling procedure terminated after 370 sampled points. Figure 5 shows the evaluation of the norm of the significant weights. Similar to the pipe section, we can observe a steep decrease in $\left\|\Delta \mathbf{W}_s^k\right\|_F^{av}$ for the first 100 sample points. This decrease is reduced with an increasing sampling space. We have $n_s = 5$ weights in $\mathbf{W}$ which explain more than 99.06 % of the variance in the dependent variables $\mathbf{y}$ after 370 sampling points. The first insignificant weight $\mathbf{w}_6$ explains only 0.19 % of the variance in $\mathbf{y}$. During the sampling procedure, n_s changed twice in the first 100 points but remained constant at $n_s = 5$ from 300 points onwards.

Repeating the sampling procedure 20 times, results in a mean number of sampling points $\overline{n_p} = 393$ with a standard deviation of $s = 49.0$. This shows that the proposed sampling procedure is consistent in its termination. The variation in the number of sampling points is caused by the randomness in the new sampling points. The performance measures $|\boldsymbol{\epsilon}_j|_{\max}$ and RMSE_j for the four dependent variables ($\mathbf{y}$) can be found in Table 5. Again, the corner points were not sampled. This results in extrapolation for certain values of the independent variables $\mathbf{y}$. The maximum absolute normalized error $|\boldsymbol{\epsilon}_{j,rel}|_{\max}$ is 0.08 %, 0.22 %, 0.28 %, and 0.28 % for Δp, ΔT, $\dot{\xi}$, and Q_{HEx5} respectively using augmented Latin hypercube sampling.

In addition to Latin hypercube sampling, Monte Carlo and Sobol sampling were used. Monte Carlo sampling (MC) terminated after $n_p = 390$, whereas Sobol sampling terminated after $n_p = 335$ which is similar to Latin hypercube sampling ($n_p = 370$). We also compared the resulting fit in the dependent variables (Table 5). As we can see, the errors have the same order of magnitude. Differences in the surrogate model fit can be caused by the randomness in neural network generation as the initial seed for neural network generation was not common for all sampling schemes.

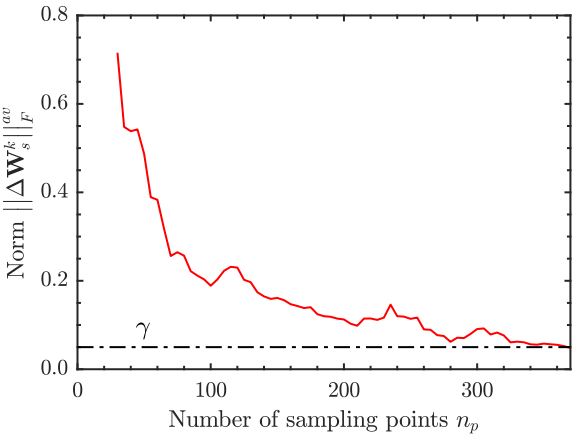

Figure 5: Development of the Frobenius norm $\left\|\Delta \mathbf{W}_s^k\right\|$ as a function of the number of sampling points with $\gamma = 0.05$ (case study 1).

Table 4: Upper and lower bounds of the independent variables ($\mathbf{u}$) (case study 1).

Variable	Unit	Lower Bound	Upper Bound
p_{in}	[bar]	−10	+10
T	[K]	−20	+20
$\dot{n}_{\mathrm{H_2},in}$	[%]	−20	+20
$\frac{\dot{n}_{\mathrm{H_2},in}}{\dot{n}_{\mathrm{N_2},in}}$	[%]	−10	+10
$\dot{n}_{\mathrm{NH_3},in}$	[%]	−20	+20
$\dot{n}_{\mathrm{Ar},in}$	[%]	−20	+20
$\dot{n}_{\mathrm{CH_4},in}$	[%]	−20	+20
$T_{HEx5,out}$	[K]	−20	+20
Split Ratio 1	[pp]	−5	+5
Split Ratio 2	[pp]	−20	+20

Table 5: Comparison of model fit errors with Latin hypercube (LHS), Monte Carlo (MC), and Sobol sampling (case study 1).

$\mathbf{y}$	Unit ϵ	Design	$\lvert\boldsymbol{\epsilon}_j\rvert_{\max}$	RMSE_j
Δp	[mbar]	LHS	7.3	0.5
		MC	2.8	0.3
		Sobol	6.4	0.9
ΔT	[mK]	LHS	12.5	0.9
		MC	16.4	1.5
		Sobol	11.8	1.1
$\dot{\xi}$	[mol/s]	LHS	0.78	0.09
		MC	0.38	0.03
		Sobol	0.84	0.07
Q_{HEx5}	[kW]	LHS	87.8	6.5
		MC	72.8	7.8
		Sobol	182.6	14.3

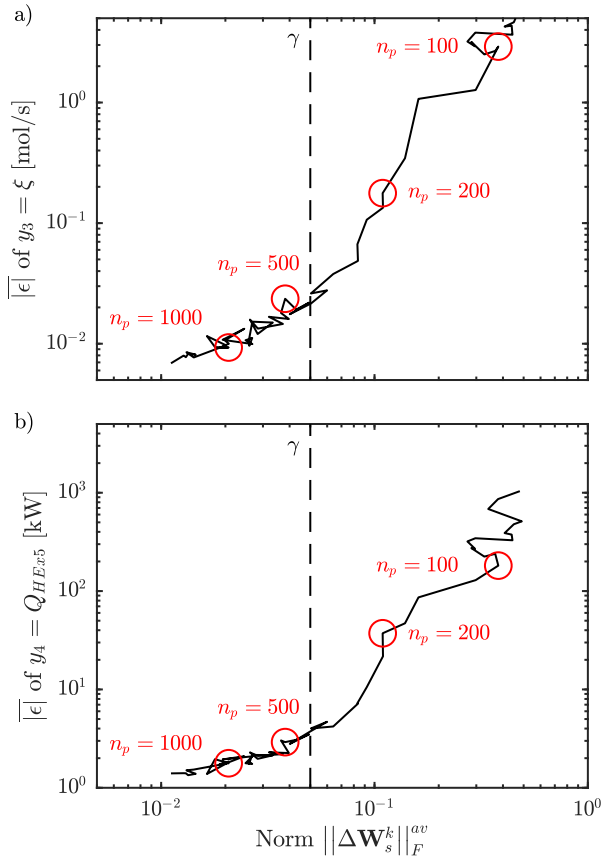

Figure 6: Mean absolute error for y_3 and y_4 of the surrogate model $\overline{|\epsilon|}$ as function of the averaged Frobenius norm of the significant weights $\mathbf{W}_s^k$ (case study 1).

The chosen threshold

$$\gamma = n_{add} \times 10^{-2} = 0.05 \tag{19}$$

was based on the threshold in the pipe case study. Hence, we want to analyze the the correlation of $\left\|\Delta \mathbf{W}_s^k\right\|_F^{av}$ with the surrogate model fit. 2000 points were sampled using augmented Latin hypercube sampling and were used in the following analysis. 10 neural networks were fitted every 5 points from 25 to 100 points, every 25 points to 1000 points and subsequently every 100 points. The used value of the dependent variable in the calculation of the error is the average value of these 10 values. PLSR was applied to all dependent variables simultaneously. Figure 6 shows the mean absolute error for the dependent variables $\dot{\xi}$ and Q_{HEx5} as a function of $\left\|\Delta \mathbf{W}_s^k\right\|_F^{av}$. The two dependent variables correspond to the variables with the highest maximum absolute relative error according to Eq. (17). If we compare Figure 6 (this case study) to Figure 3 (pipe model), we can directly see that the correlation between $\overline{|\epsilon|}$ and $\left\|\Delta \mathbf{W}_s^k\right\|_F^{av}$ is similar. In both cases, increasing the number of sampling points does not improve the fit from a certain point onward and gives a similar threshold γ.

5.2. Case study 2: Separation section of an ammonia synthesis loop

The task of the separation section of the ammonia synthesis loop is to separate ammonia from the synthesis gas. This is achieved by several sequential and parallel heat exchangers followed by a separator. A heat exchanger using water as coolant (HEx6) cools the gas stream leaving the reaction section before it is split into two parallel heat exchanger trains. The first cooling train uses the gas stream leaving the separator for heat integration (HEx7) whereas the second cooling train uses liquid ammonia as refrigerator in two separate heat exchangers (HEx8 and HEx9). The two streams are subsequently mixed and cooled (HEx10) with liquid ammonia. Ammonia is then separated in a separator in which the liquid stream is considered as product stream and the gas stream is heat-integrated with the first parallel heat exchanger (HEx7).

5.2.1. Model description

HEx6 and HEx7 are modelled using the Number of Transfer Units Method. HEx8, HEx9, and HEx10 are heat exchangers with fixed outlet temperatures $T_{HEx8,out}$, $T_{HEx9,out}$, and $T_{HEx10,out}$. The duties of the heat exchangers are calculated using the mass enthalpy of the gas streams as a function of the temperature, pressure, and composition. The mass enthalpy was calculated using a surrogate model based on cubic B-splines (Grimstad et al., 2015). This surrogate model was fitted to points sampled in the commercial flowsheet simulator Aspen HYSYS. This is a simplified approach, but rather accurate. The separator is calculating the vapour-liquid equilibrium using Raoult's law for NH_3 and Henry's law for the other gas components (Alesandrini et al., 1972). It has to be noted that heat exchangers 8 and 9 are redundant in this model structure as heat exchanger 10 is cooling the stream to a fixed outlet temperature. However, in a real plant, the cooling in heat exchangers 8,9, and 10 is achieved using an ammonia refrigeration loop. The different heat exchangers correspond then to a liquid ammonia refrigerant at different pressure levels.

The separation section has 13 independent variables ($\mathbf{u}$). These are the inlet pressure p_{in}, the inlet temperature T_{in}, 5 molar flows $\dot{n}_{i,in}$ (H_2, N_2, NH_3, Ar, and CH_4), the inlet flow rate $\dot{n}_{H_2O,in}$ and temperature $T_{H_2O,in}$ of the cooling water in HEx6, 1 split ratio, and the outlet temperatures of the heat exchangers $T_{HEx8,out}$, $T_{HEx9,out}$, and $T_{HEx10,out}$. The 12 dependent variables ($\mathbf{y}$) are the stream variables of the two streams leaving the section (Δp, ΔT, and $\dot{n}_i$) corresponding to the product (subscript P) and the recycle (subscript R) stream, the temperature change of the water stream in heat exchanger 6, ΔT_{H_2O} , and the heat duties in the heat exchangers 8, 9, and 10 (Q_{HEx8}, Q_{HEx9}, and Q_{HEx10}). Note, that the temperature difference between the liquid outlet stream and the feed stream as dependent variable can be calculated using the two in-

Table 6: Upper and lower bounds of the independent variables ($\mathbf{u}$) (case study 2).

Variable	Unit	Lower Bound	Upper Bound
p_{in}	[bar]	-10	$+10$
T	[K]	-25	$+25$
$\dot{n}_{\mathrm{H_2},in}$	[%]	-15	$+15$
$\frac{\dot{n}_{\mathrm{H_2},in}}{\dot{n}_{\mathrm{N_2},in}}$	[%]	-10	$+10$
$\dot{n}_{\mathrm{NH_3},in}$	[%]	-20	$+20$
$\dot{n}_{\mathrm{Ar},in}$	[%]	-20	$+20$
$\dot{n}_{\mathrm{CH_4},in}$	[%]	-20	$+20$
$\dot{n}_{\mathrm{H_2O},in}$	[%]	-20	$+20$
$T_{\mathrm{H_2O},in}$	[K]	-5	$+5$
$T_{HEx8,out}$	[K]	-4	$+4$
$T_{HEx9,out}$	[K]	-4	$+4$
$T_{HEx10,out}$	[K]	-8	$+8$
Split Ratio	[pp]	-5	$+5$

dependent variables T_{in} and $T_{HEx10,out}$ as

$$\Delta T_P = T_{in} - T_{HEx10,out} \tag{20}$$

We proposed to use a "grey-box" modelling approach where exact component mass balances are introduced to avoid the creation or destruction of mass through the introduction of surrogate models (Straus and Skogestad, 2017b). This can be achieved through defining a separation factor α_i for each chemical component i:

$$\dot{n}_{i,R} = \alpha_i \dot{n}_{i,in} \tag{21}$$
$$\dot{n}_{i,P} = (1-\alpha_i)\, \dot{n}_{i,in} \tag{22}$$

Consequently, 12 surrogate models have to be fitted. The upper and lower bounds of the independent variables are given in Table 6. The parameters used are the same as in the reaction section and for the pipe model (Table 3). The surrogate model structure is a 2-layer cascade forward neural network with 5 hidden neurons in each layer. The validation space consists of $n_{val} = 10^4$ randomly sampled points.

5.2.2. Results

We apply PLSR to all dependent variables $\mathbf{y}$ simultaneously because with $n_y = 12$ it is computationally more expensive to perform PLSR independently. With the selected threshold $\gamma = 0.05$ and augmented Latin hypercube sampling, the method terminated after $n_p = 625$. With the selected value $\beta = 2$ %, we find that $n_s = 5$ significant weights explain 90.81 % of the variance in the dependent variables $\mathbf{y}$. The first neglected insignificant weight $\mathbf{w}_6$ explains 1.59 % whereas $\mathbf{w}_7$ explains 1.30 % which is only slightly below β. Figure 7 shows the evaluation of the averaged norm for the simultaneous approach.

The model fit measures, $|\boldsymbol{\epsilon}_j|_{\max}$ and RMSE_j are given in Table 7. As the splitting factors for H_2, N_2, Ar, and CH_4 are all around 99 %, it is not useful to calculate the error directly. Hence, their errors are calculated as the error in the recycle stream $\dot{n}_{i,R}$. The relative error according to Eq. (17) results in a maximum absolute relative error of around 0.1 % for the first 9 dependent variables in Table 7. The last three variables (heat exchanger duties) have however a relative error of around 1 %. This can be explained by the phase change occurring in the heat exchangers through the condensation of ammonia. This phase change is not captured perfectly using the surrogate model approach. Applying the method 20 times gives an average number of sampling points of $n_p = 639$ and a standard deviation $s = 59.6$. Similar to the reaction section case study, Sobol and Monte Carlo sampling were also tested. Monte Carlo sampling terminated after $n_p = 660$ whereas Sobol sampling required $n_p = 585$ which is similar to the value $n_p = 630$ with Latin hypercube sampling. We also compared the resulting model fit in the dependent variables (Table 7). Again, the errors are of the same order of magnitude. The differences depends mainly on the seed of the neural network fitting.

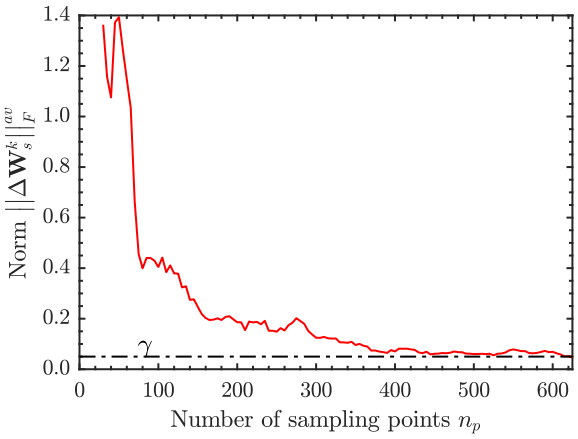

Figure 7: Development of the Frobenius norm $||\Delta \mathbf{W}_s^k||$ as a function of the number of sampling points with $\gamma = 0.05$ (case study 2).

5.3. Combination of surrogate models for optimization

So far, we fitted individual surrogate models to the reaction and separation section in two separate case studies. The resulting validation errors of the resulting surrogate models are small. However, in the real process the two models are combined using a recycle (Figure 4), but good individual fits do not guarantee that the combined model converges to the correct optimum. To study this, the reaction and separation sections are combined with the models of the purge split and the compressor train to form the flowsheet in Figure 4 which has 9 operational degree of freedom. The surrogate models for the dependent variables $\mathbf{y}$ are explicit. The introduction of a grey-box model structure through the separation factors $\boldsymbol{\alpha}$ and the rate of extent of reaction $\dot{\xi}$ guarantees mass consistency in the recycle system. The overall flowsheet is then given by a nonlinear system of equations with fewer states than the

Table 7: Comparison of model fit errors with Latin hypercube (LHS), Monte Carlo (MC), and Sobol sampling (case study 2).

$\mathbf{y}$	Unit ϵ	Design	$\lvert\boldsymbol{\epsilon}_j\rvert_{\max}$	RMSE_j
Δp_R	[mbar]	LHS	2.23	0.21
		MC	7.5	0.9
		Sobol	30.3	0.33
ΔT_R	[K]	LHS	0.17	0.02
		MC	0.08	0.01
		Sobol	0.12	0.01
Δp_P	[mbar]	LHS	6.9	0.7
		MC	1.13	0.12
		Sobol	1.6	0.09
$\Delta T_{\mathrm{H_2O},out}$	[mK]	LHS	5.7	0.3
		MC	4.32	0.3
		Sobol	1.6	0.15
$\alpha_{\mathrm{H_2}}$	[mmol/s]	LHS	1.14	0.15
		MC	2.60	0.32
		Sobol	22.0	2.0
$\alpha_{\mathrm{N_2}}$	[mmol/s]	LHS	0.43	0.06
		MC	0.07	0.01
		Sobol	0.42	0.04
$\alpha_{\mathrm{NH_3}}$	[mmol/s]	LHS	115.4	13.5
		MC	783.8	34.4
		Sobol	409.0	38.1
α_{Ar}	[mmol/s]	LHS	2.0	0.19
		MC	0.21	0.02
		Sobol	0.7	0.07
$\alpha_{\mathrm{CH_4}}$	[mmol/s]	LHS	0.62	0.06
		MC	1.81	0.17
		Sobol	0.45	0.05
Q_{HEx8}	[kW]	LHS	188	23.3
		MC	284	36.6
		Sobol	323	29.8
Q_{HEx9}	[kW]	LHS	71	8.1
		MC	56	5.3
		Sobol	99	7.1
Q_{HEx10}	[kW]	LHS	84	11.2
		MC	122	13.8
		Sobol	177	11.9

original model.

The economic cost function to minimize is

$$\begin{aligned} J = &- p_P \dot{n}_P - p_{purge} \dot{n}_{Purge} - p_S Q_{HEx5} \\ &+ p_{feed} \dot{n}_{feed} \\ &+ p_C \left(Q_{Comp1} + Q_{Comp2} + Q_{Comp3} \right) \\ &+ p_{HEx} \left(Q_{HEx8} + Q_{HEx9} + Q_{HEx10} \right) \end{aligned} \tag{23}$$

The prices for the feed, product, and purge stream as well as the compressor duties are adopted from (Arajo and Skogestad, 2008) with $p_{feed} = 0.704$ \$/kmol, $p_P = 3.4$ \$/kmol, $p_{purge} = 0.0112$ \$/kmol, and $p_C = 0.072$ \$/kWh. The heat duty in heat exchanger 5 has a cost term of $p_S = 0.036$ \$/kWh whereas the cooling in heat exchangers 8, 9, and 10 has a cost term of $p_{HEx} = 0.027$ \$/kWh. The cooling water flow and temperature to heat exchangers 1 and 6 are considered to be at a fixed value.

The operational constraints are given by the bounds in the decision variables for surrogate model generation. In addition, there are bounds on the purge split ratio and the compressor speed. The duties of heat exchanger 8 and 9 may be different between the surrogate model and the original model. This is caused by the redundancy of both heat exchangers.

Both the original model and the surrogate-based model are subsequently optimized for a given feed. The results are very similar. 8 degrees of freedom are at constrained operation. The compressor speed is unconstrained and the error with respect to the original model is 0.07 %. The resulting relative error in the cost function is 0.3 %. hanging the initial values of the operational degrees of freedom does not change the results.

The application of a different initial sampling design does not have a large influence on the results of the optimization. Using the surrogate models obtained *via* Sobol sampling and Monte Carlo sampling gave a similar small error in the optimization results compared to the surrogate models obtained *via* Latin hypercube sampling. All active constraints are identified whereas the error in the compressor speed with respect to the original model is given by 0.09 % and 0.07 % for Sobol and Monte Carlo sampling respectively.

6. Discussion

6.1. Comparison with other methods

The proposed incremental sampling procedure does not require the fitting of a surrogate model. In this respect, it differs from the ALAMO approach (Cozad et al., 2014), the smart sampling algorithm (Garud et al., 2017a), and the adaptive sampling approach of Eason and Cremaschi (2014). One advantage with our approach is that the decision about the surrogate model basis function is separated from the sampling. This allows to choose the best basis function based on the sampled space and does not require that both steps are done by the same person/group. Depending on the detailed model, the number of independent variables n_u, the number of dependent variables n_y, and the computational expense of fitting a surrogate model, it can be furthermore advantageous to avoid fitting a surrogate model at each iteration step. However, it is not possible to draw a general conclusion as the applications vary.

Although the developed procedure results only in $n_g = 1.8$ and $n_g = 1.6$ points (see (1)) in a regular grid for the reaction and separation section case studies respectively, it can result in a larger number of sampling points compared to existing adaptive procedures as it does not fit a surrogate model, and hence exploit, its fit. The question remaining is how expensive it is to fit surrogate models compared to sample additional points. If the number of independent variables n_y is large and solving the detailed

model is not very expensive, then it may be advantageous to avoid the fitting of n_y surrogate models at each sampling step. Contrary, if n_y is small and the computation expense of sampling one point is large, then other adaptive procedures could be preferable.

6.2. Choice of tuning parameters

Several tuning parameters have to be chosen. The most important tuning parameter is the threshold γ for termination. It is possible to continue the procedure by lowering the threshold γ if one is not satisfied with the performance of the surrogate model. Hence, it may be good to start with a high threshold to avoid oversampling. The results of the three case studies indicate that a threshold of approximately $\gamma = 0.05$ works for several cases, where PLSR is performed after every fifth sampled point.

A second important tuning parameter is the threshold β used to select the significant weights in $\mathbf{W}_s^k$. Depending on the definition of the independent variables, this threshold may exclude the majority of the weights (Straus and Skogestad, 2017a). For example, if molar flows are used as independent variables in the pipe case study, then only 1 weight is significant. On the other hand, using mole fraction as independent variables results in the presented 3 significant weights. Furthermore, using a hard bound β may result in frequent switching of n_s. This results in large changes in the norm as illustrated for $\Delta\mathbf{W}_s^k$ in Figure 7. This was less significant in the presented case studies by using the minimum value of n_s of the last two steps. As an alternative, it is possible to choose n_s directly after sampling a certain number of points instead of choosing the threshold β. In all case studies, n_s did not change after a certain number of sampled points, with $\left\|\Delta\mathbf{W}_s^k\right\|_F^{av}$ still much larger than the threshold γ.

Other tuning parameters are the number of sampled points in each iteration, n_{add}, and the past horizon n_f for averaging the norm. It is advisable to have a small value for n_{add} to avoid problems in the calculation of the differences $\Delta\mathbf{W}_s^k$. However, if n_{add} is chosen too small, it can be that the sampling space is not properly filled if the original Latin hypercube is not augmented. Provost et al. (1999) proposed an alternative to the arithmetic sampling approach. In this geometric approach, the number of sampled points increases with increasing step number. They showed that the computational load is reduced as the termination criterion does not have to be evaluated as frequently. Applying this approach for the PLSR-based termination criterion can however be problematic as the procedure is relying on the difference in the weights . As the calculation of the weights is not computationally expensive, at least if applied simultaneously, the arithmetic approach used in this paper seems reasonable.

The past horizon n_f is important to remove problems with oscillatory behaviour of the norm. Oversampling can be the result if it is chosen too high. The value $n_f = 5$ used in the case studies seems to be reasonable. It avoids oversampling while preventing preemptive termination of the sampling.

Table 8: Simultaneous *vs.* individual application of PLSR.

	Case Study	Approach	$\overline{n_p}$	s
1.	Reaction	Individual	406	38.6
		Simultaneous	393	49.0
2.	Separation	Individual	646	91
		Simultaneous	639	59.6

6.3. Simultaneous and individual application of PLSR

If $n_y > 1$, one has to decide whether PLSR is applied individually to each dependent variable y_i or simultaneously to all dependent variables. We used the simultaneous approach in both case studies. The advantage of applying PLSR individually is that it is possible to see which of the dependent variables requires the most sampling points.

Both of the ammonia case studies were repeated 20 times to see if there is a difference if PLSR is applied simultaneously or individually. The resulting average number of sample points and standard deviations can be found in Table 8. The difference in the average number of sampling points is not significant in either case study. Hence, we conclude that is advantageous to apply PLSR simultaneously to all dependent variables as it reduces the computational load in calculating the weights.

6.4. Choice of norm

The choice of the norm for $\left|\left|\Delta\mathbf{W}_s^k\right|\right|$ is in general not very important. It only has an influence on the defined threshold. The 1-norm will correspond to the the 1-norm of the weight $\mathbf{w}_{n_s}^k$ as the difference is usually largest in the last significant weight. The contribution from the other weights $\mathbf{w}_i^k$ with $i < n_s$ are then neglected. As a result, the termination threshold has to be higher than in the case of other norms. The infinity-norm on the other hand calculates the maximum absolute row sum. As the individual weights $\mathbf{w}_i^k$ are the columns of the matrix $\mathbf{W}_s^k$, this approach seems counter intuitive. The 2-norm and the Frobenius norm both incorporate all entries in the difference $\Delta\mathbf{W}_s^k$. The Frobenius norm was eventually chosen due to the similarity of the Frobenius norm to the vector 2-norm and its performance in the application.

6.5. Design of experiment

The termination criterion was applied to three different *design of experiment* methods; Latin hypercube, Monte Carlo and Sobol sampling. It is interesting to note that the differences are small. One would expect that Sobol and Latin hypercube sampling are superior to Monte Carlo sampling because of better space-filling properties (Garud et al., 2017b), but somewhat surprisingly our results do not show this.

7. Conclusion

A new termination criterion for incremental sampling based on partial least squares regression was introduced. It predicts when sufficient points are sampled. This termination criterion is independent of the surrogate model basis functions and does not require the fitting of a surrogate model at each sampling step. This is advantageous if the fitting of the surrogate model is computationally expensive and/or the number of dependent variables, n_y, is large. Furthermore, it allows for the separation between the sampling and surrogate model generation tasks. It can however result in an increased number of sample points compared to the existing adaptive sampling methods, as it does not utilize exploitation for the identification of new sampling points. The two case studies showed that the application of the termination criterion allows a reduction in sampling points compared to predefined sampling. For the ammonia process, the combination of the surrogate models with the compressor train and the mass recycle stream of the original model resulted in very good results in the subsequent optimization when compared to using the original detailed model.

References

Alesandrini, C. G., Lynn, S., Prausnitz, J. M., 1972. Calculation of vapor-liquid equilibria for the system NH_3-N_2-H_2-Ar-CH_4. Industrial & Engineering Chemistry Process Design and Development 11 (2), 253–259.

Arajo, A., Skogestad, S., 2008. Control structure design for the ammonia synthesis process. Computers & Chemical Engineering 32 (12), 2920 – 2932.

Beykal, B., Boukouvala, F., Floudas, C. A., Pistikopoulos, E. N., 2018a. Optimal design of energy systems using constrained grey-box multi-objective optimization. Computers & Chemical Engineering.

Beykal, B., Boukouvala, F., Floudas, C. A., Sorek, N., Zalavadia, H., Gildin, E., 2018b. Global optimization of grey-box computational systems using surrogate functions and application to highly constrained oil-field operations. Computers & Chemical Engineering 114, 99 – 110, FOCAPO/CPC 2017.

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

Biegler, L. T., dong Lang, Y., Lin, W., 2014. Multi-scale optimization for process systems engineering. Computers & Chemical Engineering 60, 17 – 30.

Boukouvala, F., Floudas, C. A., Jun 2017. Argonaut: Algorithms for global optimization of constrained grey-box computational problems. Optimization Letters 11 (5), 895–913.

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.

Boulesteix, A.-L., Strimmer, K., 2007. Partial least squares: a versatile tool for the analysis of high-dimensional genomic data. Briefings in Bioinformatics 8 (1), 32–44.

Bungartz, H.-J., Griebel, M., 2004. Sparse grids. Acta Numerica 13, 147269.

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

Cozad, A., Sahinidis, N. V., Miller, D. C., 2014. Learning surrogate models for simulation-based optimization. AIChE Journal 60 (6), 2211–2227.

Cozad, A., Sahinidis, N. V., Miller, D. C., 2015. A combined first-principles and data-driven approach to model building. Computers & Chemical Engineering 73, 116 – 127.

Davis, S. E., Cremaschi, S., Eden, M. R., 2017. Efficient surrogate model development: Optimum model form based on input function characteristics. In: Espua, A., Graells, M., Puigjaner, L. (Eds.), 27th European Symposium on Computer Aided Process Engineering. Vol. 40 of Computer Aided Chemical Engineering. Elsevier, pp. 457 – 462.

de Jong, S., 1993. SIMPLS: An alternative approach to partial least squares regression. Chemometrics and Intelligent Laboratory Systems 18 (3), 251 – 263.

Eason, J., Cremaschi, S., 2014. Adaptive sequential sampling for surrogate model generation with artificial neural networks. Computers & Chemical Engineering 68, 220 – 232.

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

Forrester, A., Sobester, A., Keane, A., 2008. Engineering Design via Surrogate Modelling: A Practical Guide. Wiley.

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

Garud, S. S., Karimi, I., Kraft, M., 2017a. Smart sampling algorithm for surrogate model development. Computers & Chemical Engineering 96, 103 – 114.

Garud, S. S., Karimi, I. A., Brownbridge, G. P., Kraft, M., 2018. Evaluating smart sampling for constructing multidimensional surrogate models. Computers & Chemical Engineering 108, 276 – 288.

Garud, S. S., Karimi, I. A., Kraft, M., 2017b. Design of computer experiments: A review. Computers & Chemical Engineering 106, 71 – 95, ESCAPE-26.

Grimstad, B., Foss, B., Heddle, R., Woodman, M., 2016. Global optimization of multiphase flow networks using spline surrogate models. Computers & Chemical Engineering 84, 237 – 254.

Grimstad, B., et al., 2015. SPLINTER: a library for multivariate function approximation with splines. `http://github.com/bgrimstad/splinter`, accessed: 2017-11-26.

Karolius, S., Preisig, H. A., Rusche, H., 2016. Multi-scale modelling software framework facilitating simulation of interconnected scales using surrogate-models. In: Kravanja, Z., Bogataj, M. (Eds.), 26th European Symposium on Computer Aided Process Engineering. Vol. 38 of Computer Aided Chemical Engineering. Elsevier, pp. 463 – 468.

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

Krige, D. G., 1951. A statistical approach to some mine valuations and allied problems at the witwatersrand. Master's thesis, University of Witwatersrand, South Africa.

Martens, H., 2001. Reliable and relevant modelling of real world data: a personal account of the development of PLS regression. Chemometrics and Intelligent Laboratory Systems 58 (2), 85 – 95, pLS Methods.

McKay, M. D., Beckman, R. J., Conover, W. J., 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21 (2), 239–245.

Metropolis, N., Ulam, S., 1949. The monte carlo method. Journal of the American Statistical Association 44 (247), 335–341.

Müller, J., Shoemaker, C. A., Pich, R., 2013. SO-MI: A surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems. Computers & Operations Research 40 (5), 1383 – 1400.

Nuchitprasittichai, A., Cremaschi, S., 2013. An algorithm to determine sample sizes for optimization with artificial neural networks. AIChE Journal 59 (3), 805–812.

Ochoa-Estopier, L. M., Jobson, M., Smith, R., 2014. The use of reduced models for design and optimisation of heat-integrated crude oil distillation systems. Energy 75, 5 – 13.

Pflüger, D., Peherstorfer, B., Bungartz, H.-J., 2010. Spatially adap-

tive sparse grids for high-dimensional data-driven problems. Journal of Complexity 26 (5), 508 – 522, SI: HDA 2009.

Provost, F., Jensen, D., Oates, T., 1999. Efficient progressive sampling. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '99. ACM, New York, NY, USA, pp. 23–32.

Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., Tucker, P. K., 2005. Surrogate-based analysis and optimization. Progress in Aerospace Sciences 41 (1), 1 – 28.

Quirante, N., Caballero, J. A., 2016. Large scale optimization of a sour water stripping plant using surrogate models. Computers & Chemical Engineering 92 (Supplement C), 143 – 162.

Sobol, I., 1967. On the distribution of points in a cube and the approximate evaluation of integrals. USSR Computational Mathematics and Mathematical Physics 7 (4), 86 – 112.

Straus, J., Skogestad, S., 2017a. Use of latent variables to reduce the dimension of surrogate models. In: Espua, A., Graells, M., Puigjaner, L. (Eds.), 27th European Symposium on Computer Aided Process Engineering. Vol. 40 of Computer Aided Chemical Engineering. Elsevier, pp. 445 – 450.

Straus, J., Skogestad, S., Jan 2017b. Variable reduction for surrogate modelling. In: Proceedings of Foundations of Computer-Aided Process Operations 2017, Tucson, AZ, USA, 8-12 Jan. 2017.

Wilson, Z. T., Sahinidis, N. V., 2017. The ALAMO approach to machine learning. Computers & Chemical Engineering 106, 785 – 795, ESCAPE-26.

Wold, S., Martens, H., Wold, H., 1983. The multivariate calibration problem in chemistry solved by the PLS method. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 286–293.

Wold, S., Sjöström, M., Eriksson, L., 2001. PLS-regression: a basic tool of chemometrics. Chemometrics and Intelligent Laboratory Systems 58 (2), 109 – 130.