Application of Surrogate Models as an Alternative to Process Simulation for Implementation of Self-Optimizing Control Procedure on Large-Scale Process Plants – Natural Gas to Liquids (GTL) Case Study

Journal:	*Industrial & Engineering Chemistry Research*
Manuscript ID	ie-2020-05715n.R1
Manuscript Type:	Article
Date Submitted by the Author:	n/a
Complete List of Authors:	Khezri, Vahid; Ferdowsi University of Mashhad Faculty of Engineering, Chemical Engineering Panahi, Mehdi; Ferdowsi University of Mashhad Faculty of Engineering, Chemical Engineering Yasari, Elham; Ferdowsi University of Mashhad Faculty of Engineering, Chemical Engineering Skogestad, Sigurd; Norwegian University of Science and Technology, Chemical Engineering

Application of Surrogate Models as an Alternative to Process Simulation for Implementation of Self-Optimizing Control Procedure on Large-Scale Process Plants – Natural Gas to Liquids (GTL) Case Study

Vahid Khezri[a], Mehdi Panahi[a]*, Elham Yasari[a], Sigurd Skogestad[b]
[a] Chemical Engineering Department, Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran
[b] Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), 7491 Trondheim, Norway
*corresponding author: Dr. Mehdi Panahi, mehdi.panahi@um.ac.ir

Abstract

High computational load, time-consuming convergence and simulation crashes are common when using process simulators for flowsheet optimization. In this paper, by replacing the large-scale physical process simulations by surrogate models, the optimization time and computational load are reduced significantly along with maintaining the accuracy and reliability. A gas to liquids (GTL) plant was used as a large-scale process plant case study. Multi-layer perceptron neural network (MLP-ANN), radial basis function neural network (RBF-ANN), support vector machine (SVM), and adaptive neuro-fuzzy inference system (ANFIS) models were selected as alternative surrogate models. These alternatives were investigated in implementation of the self-optimizing control procedure on the above case study to find the best individual and combined self-optimizing controlled variables (CVs). The MLP-ANN surrogate model showed the best performance in predicting the optimum points and for selecting the best self-optimizing CVs. In fact, it even performed better than using the full process simulator.

Keywords: Plantwide control, MLP-ANN, Fischer-Tropsch synthesis, Process flowsheet optimization, Controlled variable selection

1. Introduction

A chemical process requires the use of a control system for a variety of reasons including safety limitations, environmental policies, operational requirements, process economics, and the achievement of desirable product characteristics [1]. In recent years, with the advent of far more complex processes with heat and mass integrations, the need for optimal control systems is felt more than ever.

In a process plant, various equipment are inter-connected. It is essential to address the plantwide control issues as a subset of the process control system design. Plantwide control focuses on the structural decision-making methods for selecting controlled variables (CVs) and the pairing them with manipulated variables (MVs) [2].

The choice of CVs in a chemical plant is important both for economic optimality and for the stability of the process.

This paper is in continuation to our previous work [3], which addressed modeling and optimization of a gas-to-liquids (GTL) process plant by using of Multi-layer perceptron neural network (MLP-ANN). A comparison between simulation and MLP-ANN models in predicting the optimum point was carried out. In addition, a brief comparison between four surrogate models in terms of prediction was conducted. In the present paper, we investigate the application of surrogate models as an accurate and reliable alternative to process simulation in self-optimizing control of a GTL process plant. This is performed in two parts. Firstly, a detailed comparison is made between surrogate models and detailed simulations in terms of their optimization performance, which shows the ability of surrogate models to predict the output value and the location of optimum points. In the second part, the best surrogate model, the MLP-ANN model, is used as a fast and precise alternative in the self-optimizing procedure. By doing this, the same or even better results can be obtained with much less time and computational cost for the multiple process re-optimizations cases which are needed to calculate self-optimizing matrices.

This paper focuses on self-optimizing method as the core part of the plantwide control procedure, defined by Skogestad as follows: "*Self-optimizing control is when we can achieve an acceptable loss with constant setpoint values for the controlled variables (without the need to re-optimize when disturbances occur)*". More generally, the objective is that the use of self-optimizing control should keep the process close to optimal on the shorter time scale and thus reduce the need for reoptimizing the setpoints by the slower real-time optimization (RTO) layer.

Self-optimizing methods have been previously applied to several processes in order to find proper CVs. In the work of Panahi and Skogestad [4], the best CVs were determined by the application of self-optimizing control on a post-combustion CO_2 capturing process plant. The selected CVs and their optimum setpoints were reported for achieving optimum energy consumption in different operational regions. In another work[5], the maximization of the variable income of a GTL process plant was investigated in two operating modes. Individual and combined optimum CVs were determined using the self-optimizing method. Reduction of the worst-case loss to near zero was achieved when measurements were combined. The HDA process was explored as another large-scale process plant using the self-optimizing method [6]. Accordingly, the optimum CVs were determined and evaluated using the Aspen Plus and Aspen dynamics simulation software. The Tennessee Eastman process was another large-scale process plant, which was used to demonstrate the ability of the self-optimizing method for the selection of the best CVs. For this purpose, acceptable loss was achieved in the presence of disturbances[7].

The optimal selection of linear measurement combinations (H-matrix) for self-optimizing method was developed based on a quadratic expansion of the cost function around the optimal nominal point, together with a linear measurement model [8]. Some parameters based on the process model need to be calculated to apply the self-optimizing method to a specific process. One of these parameters is the optimum sensitivity matrix (F), which is calculated for the main disturbances in the process plant. To obtain F, it is necessary to re-optimize the whole process plant for each of the disturbances. This can be very complicated and time-consuming for a large-scale process plant when traditional process optimization methods are used[4, 5]. These optimization results in high computational loads and a high probability of simulation crashes especially when recycles are present [9]. In addition, the problems related to the flowsheet convergence, numerical solver settings and disconnecting between process simulators and computing software (e.g. Matlab) should be considered. For these reasons, the current paper investigates the use of surrogate models to reduce the computational load of optimization and subsequently speed up self-optimizing procedure, while maintaining accuracy.

Four alternative surrogate models are considered in this study:

- Multilayer perceptron neural network (MLP-ANN) using Bayesian regularization (BR)
- Radial basis function neural networks (RBF-ANN) using genetic algorithm (GA)
- Support vector machine (SVM) using particle-swarm optimization (PSO)
- Adaptive neuro-fuzzy inference system (ANFIS) using particle-swarm optimization (PSO).

There are three main reasons for this selection. Firstly, these surrogate models are among the most common data-based models used in regression problems. These models are referred in many articles in the chemical engineering field and particularly in data driven models of equipment or process plants and flowsheets. Secondly, in most cases, these models show a good performance of interpolation and reach the desired goals of prediction accuracy. Thirdly, these models are fairly easy to implement.

A wastewater treatment plant was modeled by the MLP-ANN in [10]. To this end, the neural network was trained using a data set from a wastewater treatment plant, and the resulting model was introduced as a tool for evaluating the performance. The superiority of RBF-ANN compared to the MLP-ANN was reported by [11] to predict the temperature rise of a multi-stage flash in a seawater desalination process plant. For this purpose, a single-layer ANN with twelve neurons was recommended as the optimum structure. The resulting model was also recommended as a proper alternative for the multi-stage flash process model due to high accuracy and desirable computational performance. In [12], MLP-ANN and ANFIS were used as surrogate models for columbic efficiency (CE) and power density parameters in a microbial fuel cell. The MLP-ANN model was preferred due to a simpler structure and easier training process. In [13], MLP-ANN, RBF-ANN, ANFIS, and Least Square SVM (LS-SVM), were compared to predict the amount of CO_2 solubility in ionic liquids. The performance of LS-SVM model was the best and a suitable replacement for complex interpolation methods. In addition, an innovative particle swarm optimization (PSO) algorithm was employed to optimize the parameters of the LS-SVM and ANFIS. In addition to the surrogate-assisted modeling cases above, this effective method was recently used to make self-optimizing implementation much easier [14].

A surrogate-assisted tool for fast implementation of the self-optimizing procedure was presented recently by Lima et al. [14]. In this application, which was developed in Python, the Kriging method was used as a surrogate model. In addition to demonstrating the performance of the proposed software, three industrial-scale process plants were investigated [14].

The use of surrogate models and meta-heuristic algorithms for optimizing complex processes with a high computational load that is predicted by a surrogate model has been employed in numerous papers. In [15], Nandi et al. performed a comparative investigation on the performance of two hybrid models of multilayer perceptron-genetic algorithm (MLP-GA) and support vector regression-genetic algorithm (SVR-GA) for modeling and optimizing benzene isopropylation in the Hbeta catalytic process. The GA meta-heuristic algorithm was used to optimize the input variables of each surrogate model. A significant improvement in the yield and selectivity of the desired product of process (cumene) was reported. In this study, the results from optimizing the MLP-GA hybrid model had high accuracy compared with the validation data. In [16], the prediction of the sludge volume index (SVI) was performed using MLP-ANN and RBF-ANN. The parameters of the two surrogate models (weights and biases) were optimized by GA using the available experimental data, and the results were presented in different states, including different numbers of inputs. Eventually, the performance of the MLP-ANN was reported to be better than the RBF-ANN. In [17], Ma et

al. conducted the optimization of a high-sulfur natural gas purification plant using ANN-GA hybrid method for reducing the time of computation and solving the non-convergence error in simulation software. To this end, after training and testing of the MLP-ANN model with simulations, GA was applied to the model to optimize the energy consumption of the plant.

In summary, the use of surrogate models in combination with meta-heuristic optimization algorithms is suitable for complex processes. The present paper uses this technique to re-optimize the process for each disturbance in the self-optimizing control procedure. Initially, datasets of input-output variables are created for each of the key disturbances using the existing Aspen-Hysys simulation of the gas-to-liquids (GTL) process. The parameters of the surrogate models are optimized using PSO or GA meta-heuristic algorithms. For validation, the data is divided into several categories by the *k-fold cross-validation* method. Next, surrogate models are trained on the data and validated at each step. The obtained models are optimized by the PSO algorithm, and the optimum value of the objective function is compared with the value obtained by the simulation. Finally, the best surrogate model is chosen for replacing the process simulation of the GTL flowsheet for applying the self-optimizing method.

The paper is structured as follows: Each surrogate model is first described briefly, and a description of the PSO optimization algorithm is presented. The next section briefly discusses the self-optimizing method and further explains the GTL process. Then the modeling and optimization in order to select the best self-optimizing CVs is presented, and finally the results are discussed and concluded.

2. Methods

2.1. Surrogate models

2.1.1. Multi-layer perceptron neural network (MLP-ANN)

The multilayer perceptron neural network is the most common type of neural network in the field of data-based regression and modeling. In general, MLP–ANN consists of three types of layers: the input layer, the output layer, and one or more hidden layers. The neuron is the main computational unit in a neural network. Neurons are located in different layers of the network and play a major role in the performance of a network. In addition, the weights for each connection, the value of bias, and the type of activity function in each layer affect the output value of the network. Typically, considering one hidden layer with an adequate number of neurons in the neural network structure lead to an acceptable performance. The adjustable parameter in this kind of neural network is the number of neurons in the hidden layer, whose correct determination has a significant effect on the network performance.

The behavior of a neuron can be expressed in mathematical form using equations (1) and (2):

$$u_i = \sum_{j=1}^{n} W_{i_j} X_j \tag{1}$$

$$y_i = \Theta(u_i + b_i) \tag{2}$$

Where X_j is the input vector, W_{ij} is the weight vector of the *i*th neuron, u_i is the output of the linear combiner, b_i is the bias, Θ is the activity function, and y_i is the output of the *i*th neuron.

In this work, the MLP-ANN was trained by the Bayesian regularization (BR) algorithm using the Matlab R2016b software. It has a linear transfer function in the output layer and a tangent sigmoid transfer function for Θ in its hidden layer.

2.1.2. Radial basis function neural network (RBF-ANN)

This kind of neural network has also three layers; input, output, and usually one hidden layer. The activity functions Θ in the hidden layer are of the radial basis type, and the output layer has linear activity functions. As can be seen in equations (3) and (4), in RBF-ANN, the activation function is applied on the input vector firstly and then the summation is done on the product of weight vector and output of the activity function. However, it is different in MLP-ANN based on equations (1) and (2). In MLP-ANN the activation function is applied on the result of summation, which is done on the product of the weight vector and input vector. The RBF-ANN is considered as a good alternative to the MLP-ANN due to the high training speed and simpler structure [18]. This paper used Matlab R2016b software to create the RBF-ANN.

The equations for the output of the radial basis neural network used in this paper are given in the equations (3) and (4):

$$y_i = \sum_{j=1}^{n} W_{ij} u_j + b_i \tag{3}$$

$$u_i = \Theta(X_j) \tag{4}$$

Where X_j is the input vector, W_{ij} is the weight vector of the ith neuron, u_i is the output of the activity function, the b_i stands for bias, Θ is the radial basis activity function, and y_i is the output of the ith neuron.

2.1.3. Support vector machine (SVM)

Support vector regression (SVR) for fitting data is a subset of support vector machine (SVM) which was proposed by Vapnik in 1995 [19]. This method has a high potential for nonlinear fitting and has shown to have a good performance for small data sets [20]. The main rule used in the SVR method is to minimize the upper limit of the generalization error [15]. In the SVR method, the function (x) will be formed to predict the output (y) associated with the inputs (x) with the estimation error (ε) and remove the data with violation (ξ) by a parameter called C. The parameter (ε) is a precision parameter representing the radius of the intended area for fitting the training data [21]. The parameter C also represents a trade-off between the complexity of the resulting model and the model estimation error [15].

In this paper, the Gaussian function was used as the kernel function. The code of SVM is developed in the Matlab R2016b.

2.1.4. Adaptive neuro fuzzy inference system (ANFIS)

The adaptive neuro-fuzzy inference system (ANFIS) was first introduced by Jang in 1993 [22]. This technique was improved later by aggregating the ability of neural networks. The fuzzy inference system (FIS) is a basic model based on fuzzy rules in which each fuzzy rule represents the local behavior of the system [12, 23, 24]. The Gaussian membership function in the Matlab R2016b software was used in this paper.

Regularly, the ANFIS is trained using error propagation algorithms. This training contains the adjustment of parameters related to the membership functions. However, the use of these methods for optimizing the parameters is slow and easily gets trapped in local optimum [25]. On the other hand, the use of evolutionary optimization algorithms is more reliable. [13]. Therefore, this paper uses the particle swarm optimization (PSO) evolutionary algorithm to perform the optimization on the ANFIS parameters.

2.2. Particle swarm optimization (PSO) algorithm

The PSO optimization algorithm is a population-based and the iterative approach was first proposed by Kennedy and Eberhart in 1995 [26]. Each particle represents a possible response in the optimization space and has a velocity parameter that is responsible for updating responses in each generation. The two parameters gbest and pbest, affect the orientation of the particle swarm movement. The best personal response (pbest) is the best response that each particle has obtained so far in all generations. The best general response (gbest) is the best response that has been obtained between all particles in all generations.

The velocity of all particles is initially set at zero. A random population of particles is created in the response space, and the values of gbest and pbest are determined after evaluating the value of the objective function for each particle. Accordingly, the particle velocity would be calculated for the next generation. Finally, the location of all particles is updated using the new values of velocity.

2.3. Gas to liquids process (GTL)

The GTL process combines reforming of natural gas to produce synthesis gas (CO and H_2) with a Fischer-Tropsch (FT) process to produce transportable liquid products such as diesel, naphtha, and wax [27].

There are several reasons why there might be an increase in the willingness to implement large-scale GTL plants and the growth of research on this issue. These may include [28]:

1. The desire to use stranded gas reserves
2. The desire to increase access to new gas resources by big companies in the field of energy
3. Market demand for cleaner fuels and cheaper sources of raw materials
4. The rapid development of technology by design corporations
5. The interest of states with rich gas reserves

The GTL process has three main steps: 1) Reforming of natural gas to generate syngas (a mixture of hydrogen and carbon monoxide), 2) Fischer-Tropsch reactions to create hydrocarbon chains, 3) upgrading (including cracking and hydro-processing) of the produced hydrocarbons from the FT reactions to achieve final products specifications [29].

The basic simulation model used in this paper is presented in [30]. A simple flowsheet of the process is presented in Figure 1. Further details of this simulation model, as well as the main reactions for the GTL process, are described in [5].

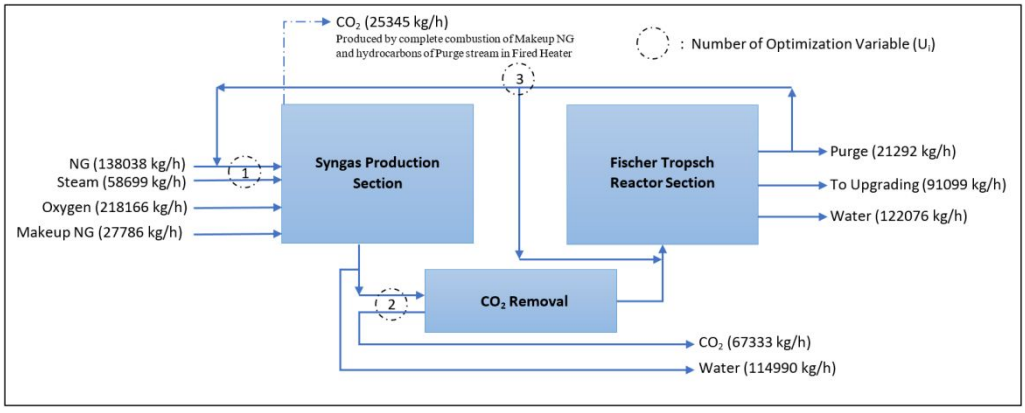


Figure 1 Simple flowsheet of the GTL process

2.4. Self-optimizing control

A proper control structure is essential in a process plant in order to achieve optimum economic performance. Therefore, various methods have been proposed for designing a suitable control structure including selection of suitable variables for control, measurement, and adjustment. One systematic method for control structure design is based on the self-optimizing method of Skogestad.

The key in this method is to select good controlled variables (CVs). In the ideal case, the setpoints of these variables can be fixed and maintained at constant values in the presence of disturbances. That is, with the CVs fixed at constant setpoints, the process operates with an acceptable loss for objective function compared to the re-optimized case. It should be noted that this method does not relate to the characteristics of the controllers, but is related to the steady-state economic performance of a process [31].

The steps to apply the self-optimizing method and selecting the optimum CVs in a process plant are as follows:

1. Definition of cost (objective) function and constraints for optimization of the process
2. Identification of degrees of freedom (DOFs) for optimization (u)
3. Identification of the major disturbances (d)
4. Steady-state optimization of the nominal case and for all disturbances
5. Identification of candidate CVs as individual measurements (y) or combination of measurements (CV = Hy)
6. Evaluation of the objective function and loss compared to the optimum when keeping alternative CVs at constant setpoints and selection of the best CV candidate.

Based on the details given in each of the above steps in [32], it is observed that in the fourth step one needs to obtain the optimum sensitivity matrix F. Typically, there are two ways to calculate the F matrix. One of them is to use equation (5), which is obtained analytically [5, 8].

$$F = G^y J_{uu}^{-1} J_{ud} - G_d^y \tag{5}$$

Here G^y is the gain of the candidate measurements (y) from the inputs (u), J_{uu} is the Hessian of the objective function for unconstraint DOFs, J_{ud} is the second derivatives of the objective function for unconstraint DOFs and disturbances, and G_d^y is also the gain of the disturbances to the candidate measurements.

In another method, used in this paper, the optimum sensitivity matrix is obtained numerically by re-optimizing the process for each disturbance. A small change is initially made for each disturbance and the new optimal values for the candidate measurements (y) are obtained numerically. The matrix F, which has dimensions of n_y (the number of candidate measurements) multiplied by n_d (the number of important disturbances) is obtained according to equation (6).

$$F = \frac{\Delta y^{opt}}{\Delta d} \tag{6}$$

where Δy^{opt} is the optimal change in y relative to the optimum nominal value and Δd is the small change applied to each of the disturbances.

The worst-case loss needed for evaluating alternative CV sets in step 6 can be obtained from Equations (7) and (8).

$$L_{wc} = worst - case\ loss = \frac{1}{2}\bar{\sigma}(M)^2 \tag{7}$$

$$M = J_{uu}^{\frac{1}{2}}(HG^{y})^{-1}(H[FW_d \quad W_n]) \tag{8}$$

where, $\bar{\sigma}(M)$ is the maximum singular value of matrix M, W_d is the diagonal matrix of the maximum expected values of the disturbances, W_n is the implementation error, J_{uu} is the Hessian of the objective function for unconstrained DOFs, G^y is the gain of the selected measurements from the inputs and H is selection or combination matrix, CV = Hy.

Because of the sensitivity of the second-order derivatives to error and the existence of several matrices in equation (5), the results of the first method for calculating F may not be reliable [4]. In addition, the simplicity of using the numerical method compared to the analytical method, makes its use more practical [33].

The numerical methods can alternatively use the data from experiments on a real process plant. However, the accuracy seems not to be sufficiently good in this case [34]. The most common option is to optimize the process in conventional softwares, such as Aspen-Hysys or Aspen Plus. In this way, the built-in optimizer plugins can be used or the software simulator might be linked to an external optimizing software like Matlab optimization toolbox. However, in this paper we instead propose the use of surrogate models for calculating F.

2.4.1 Review the implementation of self-optimizing method for the GTL process

The application of the self-optimizing method on the GTL process plant was investigated in [5]. The difference between products sale revenue and variable costs was considered as the optimization objective function. Also, the equality and inequality constraints of the equipment and process operation were discussed. The number of DOFs of the process was 15. The Mixed method was utilized in the optimizer of the UniSim process simulator for optimization[5]. There were nine equality constraints and three active constraints in the optimal nominal point. Three unconstrained DOFs were left for self-optimizing procedure.

In the present paper, the wax production rate (Y) which is the main product of the FT reactor is defined as the objective function [5, 30]. The optimization problem in terms of the three unconstraint DOFs is as presented in equation (9):

$$\begin{gathered}\textbf{Maximize Y= wax production rate (kg/h)} \\ \textbf{where} \\ \textbf{desicion variables: } \mathbf{u} = (\mathbf{u_1},\mathbf{u_2},\mathbf{u_3}) \\ \boldsymbol{u_1}\textbf{: } \frac{\boldsymbol{H_2O}}{\boldsymbol{C}} \textbf{ entering the syngas section, } \boldsymbol{u_2}\textbf{:}\boldsymbol{CO_2} \textbf{ removal percentage, } \boldsymbol{u_3}\textbf{:} \\ \textbf{Recycled tail gas to FT ratio}\end{gathered} \tag{9}$$

The 18 available measurements proposed in [5] are presented in Table 1. These measurements may be viewed as candidate CVs if individual measurements are used.

Table 1 Candidate CVs [5]

No.	Measurements	No.	Measurement
1	O_2/C (y_1)	10	CH_4 mole fraction in fresh syngas (y_{10})
2	feed H_2O/C ($y_2=u_1$)	11	H_2 mole fraction in tail gas (y_{11})
3	CO_2 removal % ($y_3=u_2$)	12	CO mole fraction in tail gas (y_{12})
4	recycled tail gas ratio to FT reactor ($y_4=u_3$)	13	CH_4 mole fraction in tail gas (y_{13})
5	H_2/CO in fresh syngas (y_5)	14	H_2 mole fraction into FT reactor (y_{14})
6	H_2/CO in tail gas (y_6)	15	CO mole fraction into FT reactor (y_{15})
7	H_2/CO into FT reactor (y_7)	16	fresh syngas flow rate (y_{16})
8	H_2 mole fraction in fresh syngas (y_8)	17	tail gas flow rate to syngas unit (y_{17})
9	CO mole fraction in fresh syngas (y_9)	18	tail gas flow rate to FT reactor (y_{18})

Seven major disturbances were identified; see Table 2where the maximum expected disturbance is shown. In the present research, three major disturbances were selected as following, the feed natural gas flow (D1), the outlet temperature of the ATR reactor (D4), and the purge flow ratio (D6).

Table 2 Disturbances of GTL process from an industrial point of view [5]

No.	Disturbance	Maximum expected value
D1	natural gas (NG) flow rate	10 %
D2	NG hydrocarbon composition: is an increase in N_2 mole fraction and corresponding decrease in hydrocarbons	In the analysis: 10 % In the simulation: 3 %
D3	fired heater outlet temperature	30 °C
D4	ATR outlet temperature	40 °C
D5	FT reaction rate constant	10 %
D6	purge ratio	15 %
D7	NG price	10 %

The number of possible individual measurement sets was equal to 816 according to equation (10).

$$\binom{18}{3} = \frac{18!}{3!15!} = 816 \tag{10}$$

In the sixth step of the self-optimizing procedure, the "exact local method" was used to determine the best set of measurements combinations as CVs (CV = Hy). In this method, the worst-case loss is computed for each candidate CV set using equations (7) and (8). The set with the smallest worst-case loss will be selected as the best set for pairing with unconstrained DOFs.

2.5. Data processing

2.5.1. Data sets generating

To develop surrogate models, it is first necessary to create a data set for use in the training and test steps. Each set contains data connecting the input and output variables of the process. A uniform distribution of points in the response space was used. The high computational cost and time-consuming evaluation of each point using the GTL process simulation makes it desirable to use few data points. To capture the nonlinearity of the process needed for describing the behavior around the optimal operation point, each input variable was examined at three levels as given in Table 3. With uniform gridding, the total number of points in the input space is 27 ($3 \times 3 \times 3 = 27$).

Table 3 Levels and ranges of input variables for optimization

Variable	Level 1	Level 2	Level 3
u_1	0.400	0.600	0.800
u_2	0.320	0.560	0.800
u_3	0.550	0.675	0.800

In addition, three major disturbances (D1, D4 and D6) were selected, but these were only examined at two levels (nominal and individually perturbed by 10%). With uniform gridding this gives 27 additional data points for each disturbance. Therefore, a total of 4×27=108 data points were generated using the detailed GTL simulation model.

2.5.2. Pre-processing

The data was refined to improve the performance of surrogate models for the later estimation. First, all data (inputs, disturbances) are normalized in the range of 0 to 1 by equation (11).

$$X = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{11}$$

Here X is the normalized value of x, and x_{max} and x_{min} are the maximum and minimum values of x, respectively.

Furthermore, the data is shuffled randomly to eliminate the effect of the data order on model performance.

2.5.3. Performance indicators

The current paper uses the root mean square error (RMSE) for selecting the optimum structure of the surrogate models. The mathematical form of this parameter is given in equation (12):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_{Sim,i} - Y_{Pred,i})^2} \tag{12}$$

Here i represents a data point, N is the number of input-output data of the training or test set, Y_{Sim} is the value of the objective function (cost) for the simulation and Y_{Pred} is the value of the estimated value of the objective function by surrogate models.

2.5.4. Overfitting check

Overfitting is when a surrogate model shows excellent performance in prediction of the training data, but performs weakly in predicting the test data or any new point in the response space. In this paper, k-fold cross validation was used to see if over-fitting has occurred. The data set for each disturbance consisting of 27 data points, was split into 24 data points for training and 3 points for testing, and then these sets were rotated, corresponding to using k=9 distinct test data sets. The results of this check showed that there is no evidence of overfitting in the training procedure. More details and box plots are available in the final part of the support information file.

2.6. Surrogate model development

2.6.1 Hyper-parameter optimization of surrogate models

In this paper, four surrogate models were investigated, including MLP-ANN, RBF-ANN, SVM, and ANFIS. To perform a fair comparison of the four surrogate models, the parameters affecting the performance of each model were optimized to maximize their capability.

The MLP-ANN used in this paper is a three-layer type, including an input layer, an output layer, and one hidden layer. The adjustable parameters in this surrogate model will be the number of neurons in the hidden layer. The number of neurons in the input and output layers is equal to the number of input and output variables, respectively.

For the RBF-ANN, the number of neurons in the single hidden layer is considered as the adjustment parameter. In addition, the value of the spread parameter relative to the radial basis function is also specified as another adjustment parameter.

The structural parameters for the SVM method include the penalty parameter C, the insensitive parameter ε, and the parameter corresponding to the kernel function σ^2.

The structural parameter in the ANFIS method is the number of membership functions.

The intervals for optimizing the above parameters are given in Table 1 of the support information file.

The value of the RMSE parameter was calculated for each integer value of the optimization variables in both the MLP-ANN and ANFIS models. The number of neurons/membership functions which gave the smallest RMSE was selected as the best parameter. In the RBF-ANN model, the optimization of the resulting MINLP problem was performed using the genetics algorithm from the optimization toolbox of the Matlab R2016b software. Finally, the parameters optimization of the SVM method was performed by the PSO algorithm. The objective function for all optimization problems was the value of RMSE in estimating the test data. The parameters used for the two PSO and GA algorithms are presented in Table 2 of the support information file.

2.6.2 Surrogate model generation

The data used to train and evaluate the models includes three inputs (u_1, u_2, u_3) and one output value to be maximized (the wax production rate Y on kilogram per hour).

In order to train and optimize the weights of the MLP-ANN and RBF-ANN, the Matlab R2016b software was used. The code corresponding to the SVM method was also implemented in this software. The PSO optimization algorithm was used to train the FIS used in the ANFIS method. The Matlab codes, models and simulations are available for interested researchers.

3. Results and discussion

3.1. Optimum structure of surrogate models

3.1.1. MLP-ANN

The number of neurons in the hidden layer was varied from 10 to 20, and the MLP-ANN was trained on the data. The data used was normalized and randomized to avoid the effect of their order. The estimation error of the test data set was calculated as RMSE. Hence, for the best performance, the optimum parameter was regarded as the number of neurons that showed the least amount of error. In this study, the optimal number of neurons for the MLP-ANN was found to be 17, with an RMSE value of 0.0222 in investigating disturbance D1. The comparison diagram of the RMSE value for any integer value of neurons in the hidden

layer associated with disturbance D1 is shown in Figure 2. In the case of disturbances D4 and D6, 10 neurons were obtained as the optimum parameter.

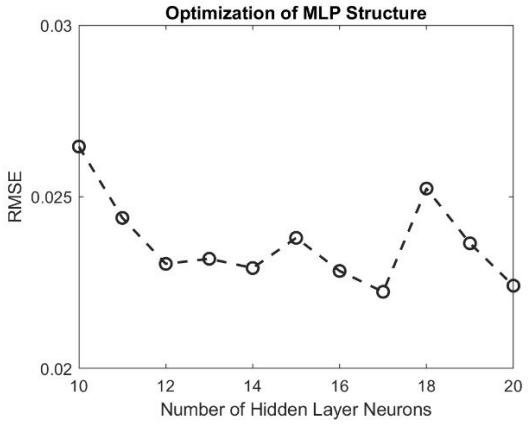


Figure 2 Optimization of MLP-ANN structure (D1)

3.1.2. RBF-ANN

The optimization of the RBF-ANN structure was performed using the GA in the optimization toolbox of the Matlab R2016b software. In this optimization, the number of neurons in the hidden layer and the spread parameter related to radial basis functions were optimized in the specified intervals. The value of the objective function was equal to the RMSE value in estimating the test data set. The optimization process and the RMSE value for each iteration associated with disturbance D1 are shown in Figure 3.

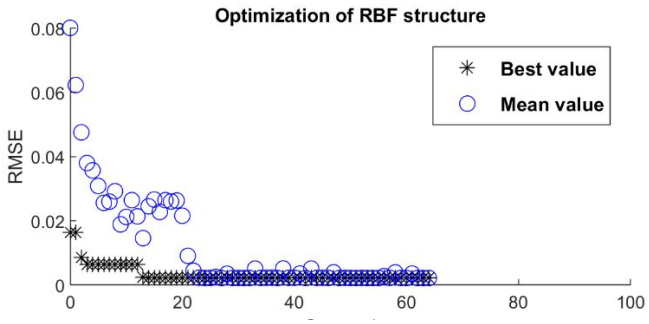


Figure 3 Optimization of RBF-ANN structure (D1)

For D1, the optimum amount of neurons in the hidden layer was 20, and the optimum spread parameter for radial basis functions was 1.87. For disturbances D4 and D6, the number of neurons in the hidden layer was 19 and 11 and the spread parameter was 1.61 and 1.31, respectively.

3.1.3. SVM

Three parameters related to the performance of the SVM were optimized in the continuous space by the PSO algorithm. For this purpose, the PSO optimization algorithm was coded in Matlab R2016b. The objective function for this optimization was also determined as the RMSE in estimating the test data set. The diagram for the optimization of the SVM model and the value of RMSE for each iteration associated with disturbance D6 is shown in Figure 4.

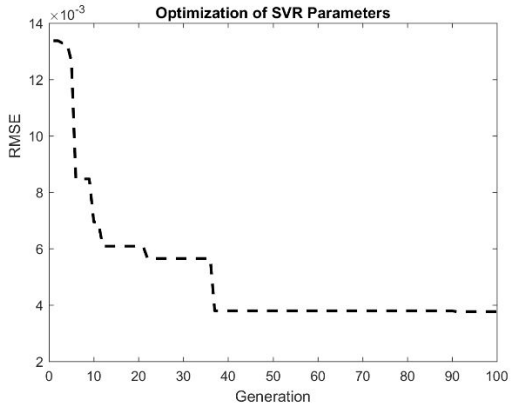


Figure 4 Optimization of SVM parameters (D6)

For disturbance D6, the optimum value of the insensitive parameter was 0.0078, the penalty parameter was 83.18e6, and the spread parameter value for the kernel function was 1.5346. For disturbance D1, the values of insensitive parameter, penalty, and spread parameter were 0.0004, 46.05e6, and 1.8254, respectively. These parameters for disturbance D4 were 0.02, 15.17e6, and 0.7748, respectively.

3.1.4. ANFIS

To determine the number of membership functions in the structure of the ANFIS model, the integer values were selected in the range of 10 to 20, and the RMSE was determined for each one. Therefore, the number of membership functions with the lowest RMSE would be the optimum value. The RMSE for each integer value of the number of membership functions for disturbance D4 are shown in Figure 5.

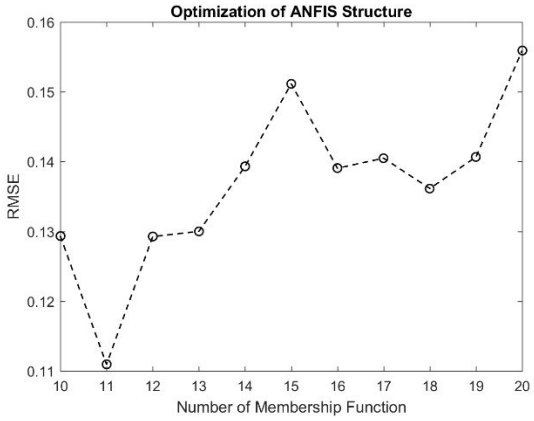


Figure 5 Optimization of ANFIS structure (D4)

As shown in Figure 5, the lowest RMSE was in the case of 11 membership functions. Therefore, in examining disturbance D4, the number of 11 membership functions was considered for the use of the optimum structure. The number of membership functions for disturbances D1 and D6 were also 13 and 10, respectively.

3.2. Surrogate models comparison

As mentioned, for each disturbance, 24 data points were used for training and 3 data points were used for testing, and these were rotated to give k=9 data sets. During the training, the surrogate model predicts the objective Ypred for each data set. A model can predict the optimal point with low error when its generalization is acceptable. Therefore it is important for a model to predict any point in response space precisely. The generalization plots of surrogates are presented in the final section of SI. In order to compare the surrogate models, the optimum point was calculated for each surrogate model.

With the k-fold cross validation methods, this gives k=9 distinct optimum results for each surrogate model for each disturbance. The surrogate models were optimized by PSO and the characteristics of the PSO algorithm are presented in Table 4.

Table 4 Characteristics of PSO optimization algorithm

PSO Optimization Algorithm	Population size: 80
	Max iteration: 1000
	Personal inertia: 2
	Social inertia: 2

To compare the accuracy of four surrogate models, the optimal inputs (values of u1, u2 and u3) found from optimizing the surrogate models were entered as input to the detailed Aspen Hysys simulation model and used to compute the corresponding simulated cost Ysim. The relative error in the cost prediction is defined as

$$Relative\ Error = \frac{Y*_{Pred} - Y_{Sim}}{Y_{Sim}} \times 100 \quad (13)$$

Here, Y^*_{pred} is the optimal cost predicted by the surrogate model and Y_{sim} is the corresponding simulated cost obtained with the Hysys process simulator using the same input values. Because of the cross validation, for each surrogate model, there are 9 values of Y^*_{pred} (and 9 corresponding values of Y_{sim}) for each disturbance. The resulting 9 relative errors are displayed in the box plot in Figure 6 for disturbance D1 for each of the four surrogate models. Figure 6 shows the high ability of the MLP-ANN surrogate model to estimate the optimum points. The very small relative error and very low variance indicate the robustness and reliability of this model. The SVM model shows similar results to the MLP-ANN model with a little more variance. The RBF-ANN model has a mean relative error close to the SVM and MLP-ANN models, but it has an outlier with about 10% relative error. The ANFIS model shows the poorest performance with a high variance and a mean relative error of about 15%. An outlier with over 100% relative error percentage further contributes to this problem.

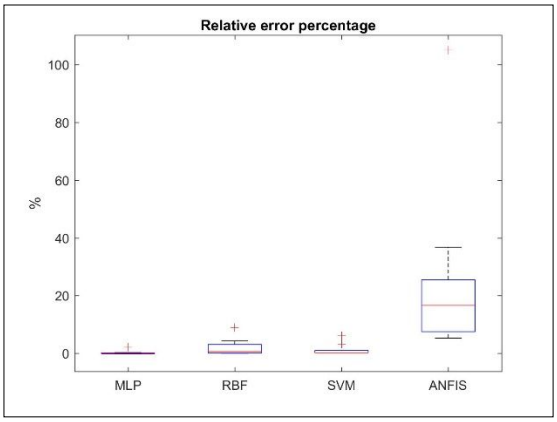


Figure 6 Relative error between optimum wax production rate predicted by surrogate models (Y^*pred) and corresponding simulated output values (Ysim) for D1

A similar behavior of the surrogate models can be seen for disturbances D4 and D6 in figure 1 of support information file. For disturbance D4 and especially D6, the MLP-ANN model has a better performance than the other surrogate models.

Table 5 Optimum points, obtained from simulation-optimization for each disturbance

disturbance	u_1	u_2	u_3	Y^*sim = wax production rate (kg/h)
D1	0.400	0.320	0.749	77777.55
D4	0.400	0.320	0.757	77769.83
D6	0.400	0.320	0.705	77890.35

It is also interesting to compare with the optimal cost Y^*_{sim} obtained for each disturbance from the GTL process simulation using the internal optimizer of Aspen Hysys v9. These results are shown in Table 5. For disturbance D1, the relative error between Y^*_{pred} and Y^*_{sim} are shown in the box plots in Figure 7. The plot is very similar to Figure 6, and again the MLP-ANN surrogate model is the best, and ANFIS is the worst.

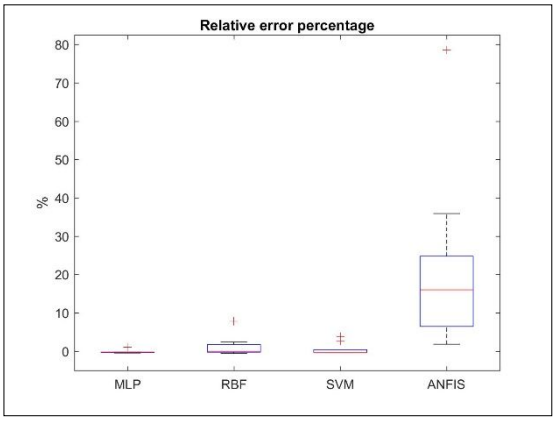


Figure 7 Relative error between optimum wax production rate predicted by surrogate models (Y^*_{pred}) and optimum point obtained from simulation-optimization (Y^*_{sim}) for D1

Similar comparisons is shown for disturbances D4 and D6 in Figure 2 of support information file.

Note that Y^*_{sim} is the optimal cost obtained from the detailed process simulator. There is only one value of Y^*_{sim} for each disturbance. Y_{sim} is the simulated cost obtained by using the input values predicted by

optimizing the surrogate models. Since Y is the wax production rate which should be maximized, it should always hold that $Y^*_{sim} > Y_{sim}$. However, because of errors in the surrogate models, it may happen that $Y^*_{pred} > Y^*_{sim}$.

The location of the input values for the predicted optimum points by the surrogate models for disturbance D1 is shown in figure 8. Note that there are 9 points for each surrogate model because of the cross validation. In addition to these predicted optimum points, the optimum point obtained from optimization of the GTL process simulation with output value Y^*_{sim}=77777.55 kg/h and input values u_1=0.4, u_2=0.32, u_3=0.749 (Table 5) is shown with a green filled circle in Figure .

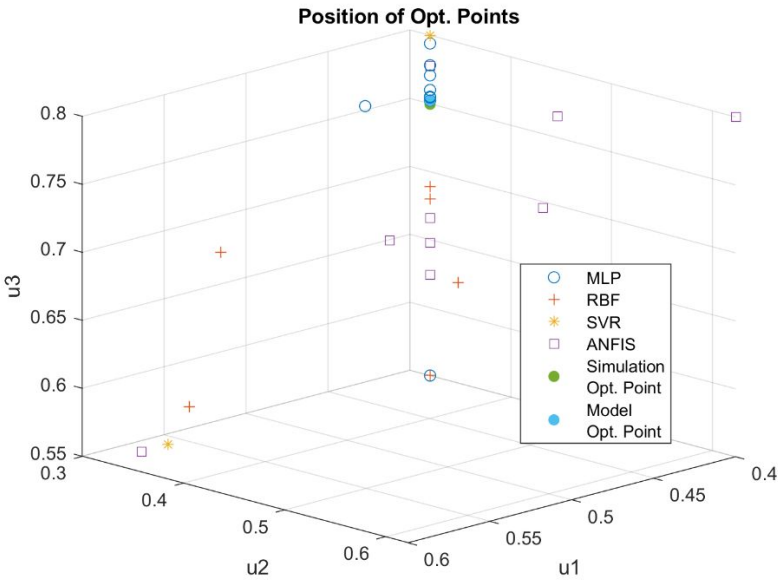


Figure 8 Position of predicted optimum input value for D1

To compare, the best (maximum) predicted output value of the wax production rate is Y^*_{pred}=77614.28 kg/h. This is for the MLP-ANN model with input values of u_1=0.4, u_2=0.32, and u_3=0.752. The corresponding cost validated by the simulation was Y_{sim}= 77649.65 kg/h, which gives a loss compared to the optimal value Y^*_{sim}=77777.55 kg/h of 0.16%.

This comparison was also made for disturbances D4 and D6. The results of these two disturbances are shown in Figure 3 of support information file.

To conclude, MLP-ANN shows the best performance. It gives a more accurate prediction with a very low error in the estimation of the output value of optimum points.

3.3. Selection of self-optimizing CVs

Based on the previous section, the MLP-ANN surrogate model is selected as the best alternative to detailed simulation of the GTL process, when performing the self-optimizing procedure. To perform a comparison, all steps were conducted in parallel using both the detailed process simulator and the MLP_ANN surrogate model.

In the self-optimizing control procedure, one needs to compute three matrices. These are the optimal sensitivity matrix (F), the Hessian of the objective function for unconstrained DOFs (J_{uu}), and gain matrix [Gy Gyd]. To calculate F, the optimal values of the candidate outputs y at nominal point and in presence of the disturbances are required. For simulation, the nominal optimum point reported in [30] was used. For the surrogate model case, the optimal nominal point found by MLP-ANN was used. These nominal optimum points and their validated output values by simulation are presented in Table 6.

Table 6 Optimal nominal points used in self-optimizing procedure calculations

	u_3=Recycle ratio to FT (-)	u_1= H_2O/C (-)	u_2= CO_2 Removal (-)	wax production rate (kg/h)
Simulation case reported in ref. 28	0.61	0.4	0.32	Y^*_{sim}=77753
Optimized MLP-ANN surrogate case	0.71	0.4	0.32	Y^*_{pred}=77878

As it was mentioned, the exact local method was used for selection of the best self-optimizing CV combination, CV = Hy. The CVs were found for both cases of individual measurements and the combination of measurements. To select three individual CVs, a branch and bound algorithm [35] was used. Among all possible CVs sets, the set with the smallest worst-case loss was selected as the best set. The best two sets of individual measurements for the two cases of conventional simulation-optimization and MLP-ANN surrogate model are presented in Table 7.

Table 7 Best two sets of self-optimizing CVs found from conventional simulation-optimization and MLP-ANN surrogate model

		best sets of CVs	worst-case loss (obtained from equation 7), kg/h
conventional simulation-optimization	1	y_2 = feed H_2O/C y_3 = CO_2 removal % **y_{14} = H_2 mole fraction into FT reactor**	**1470.8**
	2	y_2 = feed H_2O/C y_3 = CO_2 removal % **y_{13} = CH_4 mole fraction in tail gas**	**4619.9**
MLP-ANN surrogate model	1	y_2 = feed H_2O/C y_3 = CO_2 removal % **y_{10} = CH_4 mole fraction in fresh syngas**	**565.5**
	2	y_2 = feed H_2O/C y_3 = CO_2 removal % **y_{14} = H_2 mole fraction into FT reactor**	2105.3

As it is observed in Table 7, the first and second measurements (y_2 and y_3) are similar among all the best CV sets, but the third measurement is different. There are actually only three different sets in Table 7, as the best set from the simulation optimization (with y_{14}) is the second-best set from the surrogate model optimization. From the worst-case loss σ(M)=565,5 kg/h, it seems that the best CV set (with y_{10}) is obtained with the surrogate model. Indeed this is correct, but the values of the worst-case losses from the simulation and surrogate model in Table 7 cannot be compared directly, because of the differences in the obtained F, J_{uu} and G^y used to compute σ(M). This is clearly seen by difference in the loss-values 1470.8 and 2105.3 for the same CV set in Table 7.

To make sure that the set with y10 is indeed the best, detailed simulated Aspen-Hysys cost calculations were carried out, keeping constant self-optimizing CVs in presence of disturbances. The calculated loss in cost function (comparised to the re-optimized cases in Table 5) for the two best CVs sets (with y10 and y14, respectively) are presented in Tables 8 and 9 respectively. The results indeed confirm that the CV set with y_{10} is significantly better than the CV set with y_{14}. The CV set with y_{10} has an average loss of 432 kg/h for the three disturbances (Table 8), whereas the average loss for y_{14} is 3364 kg/h (Table 9). This is interesting because it shows that the surrogate model is better than the detailed simulation model in selecting the best CVs. The reason is most likely inaccuracies in computing the optimal points when using the Aspen-Hysys internal optimizer.

Table 8 Exact Aspen-Hysys loss calculations for CV-set with y10.

Y_{sim}=Cost function value with implementation CV set (**y_2, y_3 and y_{10}**) in presence of different disturbances		
D1	D4	D6
77707.4	77247.2	77493.6
loss		
70.1	613.6	613.0

Table 9 Exact Aspen Hysys loss calculations for CV set with y14.

Y_{sim}=Cost function value with implementation CV set (**y_2, y_3 and y_{14}**) in presence of different disturbances		
D1	D4	D6
77316.8	73150.5	73184.3
loss		
460.7	4710.3	4922.3

The above is with single measurements. In order to select the best combination of measurements as candidate CVs, a partial branch and bound algorithm was used to select the set with the least amount of worst-case loss among possible combinations. Optimal losses for different numbers of CVs combinations are presented in Figure for both the surrogate model and simulation-optimization cases.

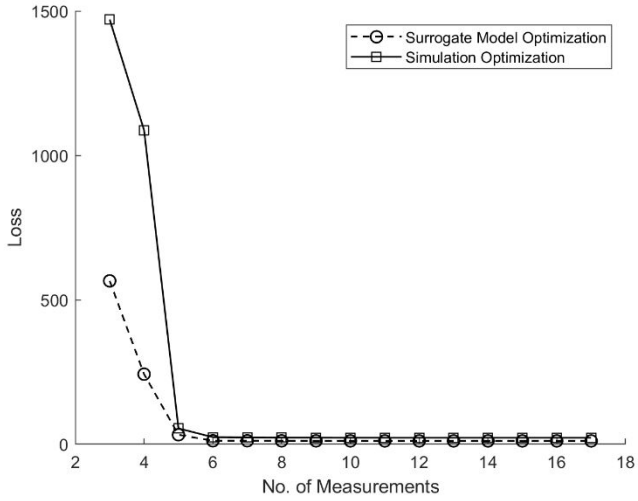


Figure 9 Minimum worst-case loss σ(M) with different number of measurements to be combined as CVs for both simulation-optimization and MLP-ANN cases

Figure 9 illustrates that with combination of five or more measurements, the worst-case loss will be zero from an engineering point of view.

In summary, it was shown in this work that with application of MLP-ANN surrogate models in self-optimizing procedure, the required time for implementation of this procedure reduces significantly specially when this method is applied on large scale processes where several adjustments, recycles and so forth should be iterated and solved to get the whole flowsheet converged. As optimization of surrogate models is quite simple, efficient and very fast, replacement of process flowsheets with well-trained surrogate models could improve the accuracy of required matrices in self-optimizing procedure and this consequently result in getting more reliable self-optimizing CVs.

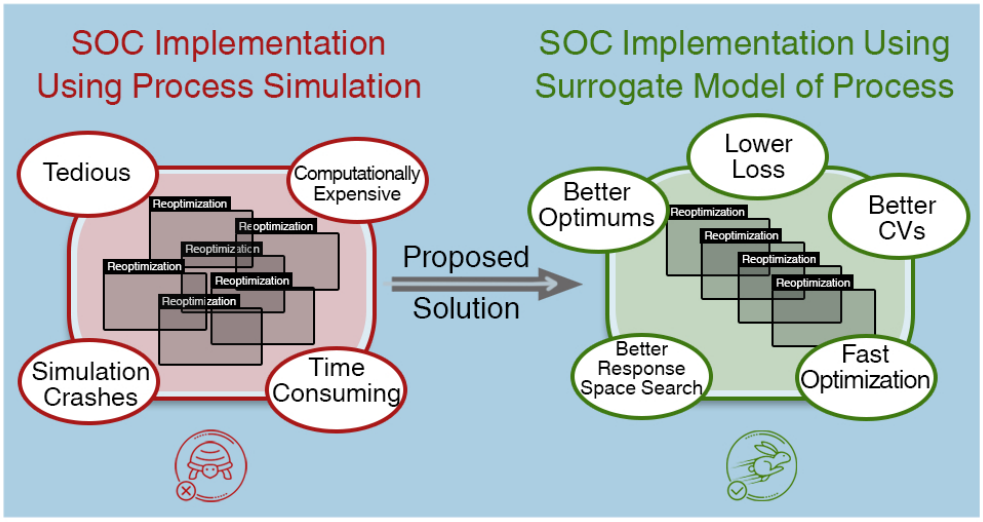
4. Conclusion

In this study, the application of surrogate models was investigated to avoid the need to use time-consuming and non-robust detailed simulation for process optimization in the self-optimizing control procedure. The four surrogate models include multi-layer perceptron (MLP-ANN), radial basis function (RBF-ANN), support vector machine (SVM) and adaptive neuro-fuzzy inference system (ANFIS). A natural gas to liquids (GTL) process was selected as a large-scale process plant with several recycles. The surrogate models were compared in terms of predicting the output value as well as the optimal operating points. The MLP-ANN surrogate model showed the best performance in almost all cases and was found to be a good replacement for a detailed GTL process flowsheet simulator in the self-optimizing procedure. The results revealed that the best self-optimizing CV set obtained from the MLP-ANN surrogate model was more reliable than the best set obtained from the detailed simulation-optimization method. Therefore, surrogate models, which are simpler and very much faster, could be good alternatives to process flowsheet simulation when the self-optimizing procedure is applied. The use of these models can significantly reduce computational load and time for selecting the best CVs.

5. References

(1) Stephanopoulos, G., *Chemical process control*. Prentice hall Englewood Cliffs, NJ: 1984; Vol. 2.
(2) Larsson, T.; Skogestad, S., Plantwide control-A review and a new design procedure. *Modeling, Identification and Control* **2000,** 21, (4), 209-240.
(3) khezri, v.; Yasari, E.; Panahi, M.; Khosravi, A., A Hybrid ANN− GA-based Technique to Optimize a Steady-State GTL Plant. *Industrial & Engineering Chemistry Research* **2020**.
(4) Panahi, M.; Skogestad, S., Economically efficient operation of CO2 capturing process part I: Self-optimizing procedure for selecting the best controlled variables. *Chemical Engineering and Processing: Process Intensification* **2011,** 50, (3), 247-253.
(5) Panahi, M.; Skogestad, S., Selection of controlled variables for a natural gas to liquids process. *Industrial & engineering chemistry research* **2012,** 51, (30), 10179-10190.
(6) de Araújo, A. C.; Govatsmark, M.; Skogestad, S., Application of plantwide control to the HDA process. I—steady-state optimization and self-optimizing control. *Control engineering practice* **2007,** 15, (10), 1222-1237.
(7) Larsson, T.; Hestetun, K.; Hovland, E.; Skogestad, S., Self-optimizing control of a large-scale plant: The Tennessee Eastman process. *Industrial & engineering chemistry research* **2001,** 40, (22), 4889-4901.
(8) Halvorsen, I. J.; Skogestad, S.; Morud, J. C.; Alstad, V., Optimal Selection of Controlled Variables. *Industrial & Engineering Chemistry Research* **2003,** 42, (14), 3273-3284.
(9) Straus, J.; Skogestad, S., Minimizing the complexity of surrogate models for optimization. In *Computer Aided Chemical Engineering*, Elsevier: 2016; Vol. 38, pp 289-294.
(10) Mjalli, F. S.; Al-Asheh, S.; Alfadala, H., Use of artificial neural network black-box modeling for the prediction of wastewater treatment plants performance. *Journal of Environmental Management* **2007,** 83, (3), 329-338.
(11) Aminian, A., Prediction of temperature elevation for seawater in multi-stage flash desalination plants using radial basis function neural network. *Chemical Engineering Journal* **2010,** 162, (2), 552-556.
(12) Esfandyari, M.; Fanaei, M. A.; Gheshlaghi, R.; Mahdavi, M. A., Neural network and neuro-fuzzy modeling to investigate the power density and Columbic efficiency of microbial fuel cell. *Journal of the Taiwan Institute of Chemical Engineers* **2016,** 58, 84-91.
(13) Baghban, A.; Mohammadi, A. H.; Taleghani, M. S., Rigorous modeling of CO2 equilibrium absorption in ionic liquids. *International Journal of Greenhouse Gas Control* **2017,** 58, 19-41.
(14) Lima, F. S.; Alves, V. M. C.; Araujo, A. C. B., Metacontrol: A Python based application for self-optimizing control using metamodels. *Computers & Chemical Engineering* **2020,** 140, 106979.
(15) Nandi, S.; Badhe, Y.; Lonari, J.; Sridevi, U.; Rao, B.; Tambe, S. S.; Kulkarni, B. D., Hybrid process modeling and optimization strategies integrating neural networks/support vector regression and genetic algorithms: study of benzene isopropylation on Hbeta catalyst. *Chemical Engineering Journal* **2004,** 97, (2-3), 115-129.
(16) Bagheri, M.; Mirbagheri, S. A.; Bagheri, Z.; Kamarkhani, A. M., Modeling and optimization of activated sludge bulking for a real wastewater treatment plant using hybrid artificial neural networks-genetic algorithm approach. *Process Safety and Environmental Protection* **2015,** 95, 12-25.
(17) Ma, L.; Hu, S.; Qiu, M.; Li, Q.; Ji, Z., Energy consumption optimization of high sulfur natural gas purification plant based on back propagation neural network and genetic algorithms. *Energy Procedia* **2017,** 105, 5166-5171.
(18) Girosi, F.; Poggio, T., Networks and the best approximation property. *Biological cybernetics* **1990,** 63, (3), 169-176.
(19) Vladimir, N. V., Statistical learning theory. *Xu JH and Zhang XG. translation. Beijing: Publishing House of Electronics Industry, 2O04* **1998**.
(20) Wang, J.; Li, L.; Niu, D.; Tan, Z., An annual load forecasting model based on support vector regression with differential evolution algorithm. *Applied Energy* **2012,** 94, 65-70.

(21) Golkarnarenji, G.; Naebe, M.; Badii, K.; Milani, A. S.; Jazar, R. N.; Khayyam, H., Support vector regression modelling and optimization of energy consumption in carbon fiber production line. *Computers & Chemical Engineering* **2018,** 109, 276-288.
(22) Jang, J.-S., ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics* **1993,** 23, (3), 665-685.
(23) Yilmaz, I.; Kaynar, O., Multiple regression, ANN (RBF, MLP) and ANFIS models for prediction of swell potential of clayey soils. *Expert systems with applications* **2011,** 38, (5), 5958-5966.
(24) Loukas, Y. L., Adaptive neuro-fuzzy inference system: an instant and architecture-free predictor for improved QSAR studies. *Journal of medicinal chemistry* **2001,** 44, (17), 2772-2783.
(25) Yingjie, L.; Baoshu, W., Study on the control course of ANFIS based aircraft auto-landing. *Journal of Systems Engineering and Electronics* **2005,** 16, (3), 583-587.
(26) Kennedy, J.; Everhart, R., A new optimizer using particle swarm theory. In proceedings of the sixth international symposium on micro machine and human science. Nagoya Japón. *IEEE service center Piscataway, NJ* **1995**.
(27) Keshav, T. R.; Basu, S., Gas-to-liquid technologies: India's perspective. *Fuel Processing Technology* **2007,** 88, (5), 493-500.
(28) Heisch, T.; Sills, R.; Briscoe, M., Emergence of the gas-to-liquid industry: A review of global GTL development. *Journal ofNatural Gas Chemistry* **2002,** 11, 1-4.
(29) Bao, B.; El-Halwagi, M. M.; Elbashir, N. O., Simulation, integration, and economic analysis of gas-to-liquid processes. *Fuel Processing Technology* **2010,** 91, (7), 703-713.
(30) Rafiee, A.; Panahi, M., Optimal Design of a Gas-to-Liquids Process with a Staged Fischer-Tropsch Reactor. *Chemical Engineering & Technology* **2016,** 39, (10), 1778-1784.
(31) Jäschke, J.; Cao, Y.; Kariwala, V., Self-optimizing control–A survey. *Annual Reviews in Control* **2017,** 43, 199-223.
(32) Skogestad, S., Plantwide control: The search for the self-optimizing control structure. *Journal of process control* **2000,** 10, (5), 487-507.
(33) Straus, J.; Skogestad, S., Self-Optimizing Control in Chemical Recycle Processes. *IFAC-PapersOnLine* **2018,** 51, (18), 536-541.
(34) Alstad, V.; Skogestad, S., Null space method for selecting optimal measurement combinations as controlled variables. *Industrial & engineering chemistry research* **2007,** 46, (3), 846-853.
(35) Kariwala, V.; Cao, Y., Bidirectional branch and bound for controlled variable selection. Part II: Exact local method for self-optimizing control. *Computers & chemical engineering* **2009,** 33, (8), 1402-1412.


SOC Implementation
Using Process Simulation
Tedious
Computationally
Expensive
Reoptimization
Simulation
Crashes
Time
Consuming
Proposed
Solution
SOC Implementation Using
Surrogate Model of Process
Lower
Loss
Better
Optimums
Better
CVs
Better
Response
Space Search
Fast
Optimization