Efficient Surrogates Construction of Chemical Processes: Case studies on Pressure Swing Adsorption and Gas-to-Liquids

Zhimian Hao,¹ Chonghuan Zhang¹ and Alexei A. Lapkin¹,²,*

¹Department of Chemical Engineering and Biotechnology, University of Cambridge, Cambridge CB3 0AS, UK
²Cambridge Centre for Advanced Research and Education in Singapore Ltd, 1 Create Way, CREATE Tower #05-05, 138602, Singapore

Abstract

We propose a workflow for reduction in the time required for data generation during generation of statistical digital twins. This methodology is particularly relevant for real-world engineering problems when data generation is expensive. A prerequisite for building surrogates is sufficient input/output data, whereas over-sampling can hardly improve the regression accuracy. The time for data generation can be reduced via (1) reduction of the average time spent on generating individual data points and (2) reduction in the total number of data points, by reducing the sampling rate with the improvement of surrogate quality. Examples of a dynamic process and a steady-state process from the field of carbon capture and utilization are used as two case studies: pressure swing adsorption (PSA) and Gas-to-Liquids (GTL). With the proposed methodology, the time for surrogate generation can be reduced by 88% for PSA and 60% for GTL, respectively.

Keywords: machine learning; digital twin; sampling rate; data effectiveness; pressure swing adsorption; Gas-to-Liquids

1. Introduction

Surrogates, or data-driven models, play an increasingly important role in the evaluation of engineering systems. In recent years, machine learning has offered more types for the ‘surrogate family’ and boosted the application of surrogates.¹-⁴ To build surrogates, one of prerequisites is data, the generation of which can be time-consuming and prohibitively expensive for real-world engineering systems when no preliminary data is available. Conventional sampling methods can lead to under/over-sampling issues.⁵ Our hypothesis is to develop a workflow to reduce the time spent on data generation by reducing the total number

*Corresponding author: Prof. Alexei A. Lapkin. Email: aal35@cam.ac.uk
of the required data points, and improving the data effectiveness (or reducing the average time spent on generating an individual data point). To successfully set up such a methodology, it is beneficial to review prior work on surrogate modelling and smart sampling, which will be elaborated in the remainder of this section.

The evaluation of real-world engineering systems is expensive. With physical models and inputs, computer simulations can accurately deliver information about systems. Although cheaper than experimental or industrial data, simulations can be considerably slow when the systems involve multi-scale, multiphase phenomena and dynamic behaviors. Data-driven surrogates can represent the original physical models by building a relationship between inputs and outputs (responses). For some complex systems, there are even no physical models available, and thus data-driven surrogates together with the design of experiments (DoE) seem to be the only choice. Surrogates are cheap-to-evaluate and can directly be employed for optimization, control and design problems in many engineering fields, e.g. chemical engineering, pharmaceutical manufacturing, supply chain management and aerospace engineering. The traditional formulation of surrogates is polynomials, moving least-squares and radial basis functions, etc., but the low accuracy is often criticized.

In machine learning, artificial neural networks (ANNs) and Gaussian Processes (GPs) have been identified as promising formulations for surrogate generation. ANNs are regarded as universal approximators and allow to fit multiple outputs simultaneously. The flexible structure and various activation functions enable ANNs to accurately fit any linear or nonlinear relationship of inputs/outputs. GPs belongs to a non-parametric model type, which has excellent fitting performance and can predict uncertainty. However, GP is generally implemented for the single output, whereas the formulation for fitting multiple outputs is complex. An alternative solution is to use independent GPs for each output, while the correlations of outputs are missing, which can cause worse predictions for highly-coupled outputs.

Data is one of the prerequisites to train a surrogate. When the available data is limited, data generation is necessary, but it can be extremely expensive for real-world engineering systems. To demonstrate it, we consider an example of a chemical process – pressure swing adsorption (PSA). Details about PSA can be found in Section 4, where we use PSA as one of the case studies. Data is randomly sampled, while the surrogate is trained iteratively. As shown in Figure 1, the time for data generation has a significantly higher order of magnitude than that
for surrogate training, which is one of the common problems in chemical process systems. Insufficient data quantity cannot guarantee good quality for constructing a surrogate, while Garud et al. report that simply increasing the data quantity cannot lead to better performance of a surrogate. Thus, the quality of surrogates should rely on both - data quantity and quality.

To reduce the time for data generation, the first objective is to obtain good-enough surrogates with the minimum amount of data. There are two types of sampling methods: one-shot and sequential (adaptive) methods. The former method samples the design space uniformly in one go and then builds a surrogate, while the latter samples data in batch and refine the surrogate iteratively. The one-shot is straightforward but may result in under/over-sampling, where either poor regression or inefficiency can occur. In recent years, the sequential methods tend to be popular because they are reported to better balance the regression performance and efficiency. However, over-sampling is still hard to avoid by a typical sequential method. To demonstrate this, we still use the example of a PSA process. Mean squared errors (MSE) is employed to evaluate the regression performance. The data is sequentially sampled in a space-filling method, which is stochastic. A stochastic method has anytime behavior, where sampling can be stopped at any time. We plot the fitting performance against time to observe the termination condition as shown in Figure 2. We divide the sequential sampling as two equal parts based on the number of data points (approximately equivalent to time, because the time for surrogate training can be neglected compared to data generation as shown in Figure 1). The plot indicates that regression improvement in the first half is significantly greater than that in the second half. This suggests that too much data is not worth collecting, although the limit of the infinite
number of data points is required to fully fill the searching space theoretically. In other words, due to the anytime behaviour, we should stop any further sampling after achieving a certain fitting performance. Also, we noticed that the MSE values fluctuate all the time. Hence, it is rather challenging to determine an optimal termination criterion.

![Figure 2. Illustration of why too many data points might not be worth sampling.](image)

Meanwhile, with the linear increase in the number of data points generated, time seems to exponentially increase for surrogate training (Figure 1). Consequently, the surrogate training might be extremely time-consuming if the number of sampled data points is high. Therefore, over-sampling brings the unnecessary time costs for data generation and extra effort for surrogate training. To avoid over-sampling, it might be beneficial to spot the non-improvement trend as early as possible.

The second objective in sampling is concerned with the improvement of data quality. The design space for sampling is initially based on limited prior experience or even random guesses, and the infeasible design space is commonly unavoidable. Consequently, some inputs, which happen to be sampled from the infeasible design space, can lead to unexpected outputs, such as non-converged simulation outputs or experimental failures. Such outputs will introduce significant errors to the surrogate construction, and thus they are ineffective data, which should be screened out. To increase data effectiveness, a classifier can be constructed to distinguish between infeasible and feasible design spaces. Such application of a classifier has been successfully demonstrated in prior research works. Ibrahim et al. reported that a support vector machine (SVM) can be used to set a feasibility constraint to filter infeasible design space for
non-converged simulations. Cao et al. adopted a Bayes classifier to improve the design space for the experimental conditions of formulations. Kim et al. applied a combined classification system to increase the quality of design space for computation-based material discovery, which can significantly reduce the number of further samplings. Houben et al. included a classifier into a Bayesian optimisation algorithm to avoid infeasible experiments in emulsion polymerisation.

To further enhance data effectiveness, exploitation-based methods can be considered to identify the promising sample placement. A simple example can be used to explain the importance of sample placement. Suppose a data-driven model fits a model with a simple form $y = \sin(x)$ at the design space $[0, \pi]$. Then the sampling places at $0, \frac{\pi}{2}, \pi$ are more important than other places. Although the real-world problems can be more complex with high dimensionality, such placement can be identified by selecting the data points which can minimize the surrogate uncertainty. Cozad et al. developed a workflow called ALAMO for algebraic model building in a sequential sampling way. For each new data to sample, they apply a derivative-free optimization technique to maximize the error between the surrogate and the original model. To identify one optimal sample placement, many new data points are required to be generated for evaluation during the optimization. Consequently, this method, actually, generates far more data points than the number of optimal data points, as presented in their results. An alternative approach is to employ GP-based surrogates, which contain the uncertainty-related information, and the new points can be sampled to reduce the predicted uncertainty. However, the model-based methods can result in over-sampling in local spaces and lose generality. For example, Gonzalez-Garay et al. reported a surrogate-based optimization method for a chemical flowsheet, and over-sampling was executed to refine the surrogate near the optimal solution. However, the obtained surrogate may not be applicable for other case studies if the objective changes, because only the prior local space near optima is highly exploited.

To develop a more generic surrogate, the model-free sampling strategy can be more advantageous because it can guarantee randomness. In this sense, we should consider homogeneous sampling methods, such as Latin hypercube sampling (LHS), Sobol Sampling, and Monte Carlo sampling. Meanwhile, exploration should be well balanced with exploitation (model-free type) during sequential sampling. On the one hand, exploration samples the entire design space homogeneously. On the other hand, exploitation focuses on sampling the complex/nonlinear space. A general strategy is to divide the overall design space into several
sub-spaces, and then rank them based on certain score criteria, following by exploiting the sub-space with the worst score. Since it is out of the scope of this work, more detailed information can be referred to in the cited literature.35-37

Overall, prior studies focus more on the sampling methods, but little attention is given to reducing the total time for data generation. In this work we aim to obtain good-enough surrogate models via efficient sequential data generation. The efficiency benefits from:

- reduction of the total number of sampling points; the sampling rate gradually slows down with the improvement of regression performance, and thus the non-improvement trend will be spotted at an earlier stage;
- reduction in average time spent on generating an individual datapoint; a classifier is trained to screen out unpromising data inputs, and thus avoid spending time on unnecessary simulations or experiments.

The remaining sections are structured as follows. Section 2 proposes the overall workflow for the surrogate construction. Section 3 demonstrates the state-of-art of two principles for efficient data generation. Section 4 presents two case studies on chemical processes, following by conclusions and outlooks in the final section.

2. **Workflow for surrogate construction**

This section presents the workflow for surrogate generation. To enhance the surrogate quality globally, this work is focused on exploration-based sampling (exploitation-based method may affect the evaluation on the influence of the sampling rate). We select Latin Hypercube Sampling (LHS), because it does not lose generality and can deliver a well-distributed sampling result.5 As shown in Figure 3, the algorithm samples initial data by LHS. Then, simulations or experiments generate the corresponding outputs. With the initial data points (or together with a few iterations), an SVM classifier is trained to separate the feasible design space from the infeasible one. The data inputs from the infeasible region are deleted, while inputs in the feasible one are passed to the simulator for outputs. To fit multiple outputs simultaneously, ANN is selected as the surrogate type. In the following iterations, data is sampled in batch by LHS for surrogate refinement, with which the sampling rate gradually slows down.
Figure 3. Proposed workflow for surrogate generation.

We briefly demonstrate the procedure for data generation. Process simulators – Aspen Plus, Dymola, or gPROMS are powerful tools for process modelling. Still, they are not flexible for data storage and are limited in their capacity to access high-level statistic packages. Therefore, it is necessary to establish an interface between process simulators and high-level programming languages (e.g., MATLAB, Python, etc.). Advantages of MATLAB over other programming languages are: reliable optimization and its machine-learning toolboxes, which have been well-established for commercial use. The automation of MATLAB to process simulators has
previously been reported in process design and control. As shown in Figure 4, the inputs are sampled by LHS in MATLAB and passed to simulators, e.g., Dymola for dynamic simulations and AspenPlus for steady-state process simulations. The obtained outputs are sent back to MATLAB for data collection.

![Diagram](Image)

**Figure 4. Data generation by interfacing MATLAB with process simulators.**

The obtained dataset is divided into three parts: training, validation, and test at a ratio 70% / 20% / 10%. Random search is used to optimize structure of networks, e.g., the number of layers and the number of neurons. We select the hyperbolic tangent (tanh) as the activation function. For each iteration, the training dataset is fitted 60 times by the different random network structures, based on which the MSE values of the validation dataset are calculated. The network with the minimal validation MSE value is then selected as the surrogate model in the corresponding iteration. Performance of the obtained surrogate is evaluated by the test dataset.

3. **State-of-art for efficient data generation**

The two essential elements, the classifier and slow-down sampling, are detailed in this section.

### 3.1 Classifier SVM

The sampled points might fall in the infeasible design space due to extreme operating conditions for experiments (e.g., unexpected reactions occur at high temperature) or non-converged recycle streams, or integration failure on stiff models during computational simulations. A classifier can be trained to pre-treat the data inputs. Only the selected data inputs can be passed into the simulation or experiment stage, thus saving the average time spent on a single data point.

SVM is a machine learning technique primarily for classification. SVM was initially proposed as a linear classifier, while Vapnik et al. expanded its application as a nonlinear classifier in 1995. It is a mature and reliable method, the success of which was proven in the fields of pattern recognition and computer vision problems. For a typical chemical process, the high-
dimensional features (or multiple inputs) and the nonlinearity are unavoidable. Ibrahim has demonstrated its successful application in chemical process engineering. A toolbox of SVM can be accessed in MATLAB, so SVM is selected as the classifier in this work.

The training process for SVM is similar to the steps for surrogate training. Two differences are specified here. Firstly, only the dataset in the initial several iterations is used to train the classifier. This is because the classifier in this work is expected to give a rough classification between infeasible and feasible design spaces, so the iterative refinement for the classifier is not necessary. Secondly, the output for the classifier is binary, 0 and 1: set 0 if the simulation outputs fall on the infeasible space, while setting 1 if the simulation outputs fall on the feasible area. Following this, the data inputs together with the classifier outputs are used to train the SVM.

3.2 Slow-down sampling

As illustrated in Figure 2, too much data is not worth sampling, while data generation is usually more time-demanding than surrogate training. Herein, we define the sampling rate as the newly added number of samples per iteration, \( N_{\text{added}} \). A slow-down sampling strategy is considered: slow down the sampling rate while the quality of the iteratively refined surrogates improves. To achieve this, the improvement rate of surrogate quality needs to be quantified.

MSE can be used to quantify the regression performance. We use the moving mean (\( \hat{MSE} \)) to smooth the fluctuation of the MSE curve. The improvement rate of surrogates can be defined as the MSE decrease per data added, or we call it the slope of MSE as Equation (1). A scaling function can scale the slope values to a relative value, relative slope as Equation (2). This value is generally between -1 and 1 (due to the fluctuation, the value of the slope can be positive). The added ratio function can convert the relative slope to the value of the added ratio (typically between 0 and 1). Following this, the number of added samples is calculated by Equation (4), where \( N_{\text{upper}} \) is the maximum adding number.

\[
\text{slope}_{\text{MSE}} = \frac{MSE_{i+1} - MSE_i}{N_{\text{added}_i}} \quad (1)
\]
\[
\text{relative slope}_i = \frac{\text{slope}_{\text{MSE}_i}}{\text{slope}_{\text{MSE}_2}} \quad (2)
\]
\[
\text{added ratio}_i = \text{ratio}(\text{relative slope}_i) \quad (3)
\]
\[
N_{\text{added}_i} = N_{\text{upper}} \cdot \text{added ratio}_i \quad (4)
\]
The ratio function, Equation (3), can be directly proportional to the absolute value of relative slope, \( i \). To further decrease the sampling rate near the optimal surrogate, we use a trigonometric-type ratio function as shown in Equation (5). With the assistance of trigonometric-type ratio function, the sampling rate can drop extensively when the relative slope is approaching 0, compared to direct proportionality (Figure 5).

\[
added\ ratio_i = \begin{cases} 
-S \cdot \cos \left( \frac{\pi}{2} \cdot |relative\ slope| \right) + S & , |relative\ slope| < 1 \\
S & , |relative\ slope| \geq 1
\end{cases} 
\] (5)

Figure 5. Added ratio function for sampling rate: trigonometric type (Scaling factor, \( S = 1\) ) vs. proportional type.

To better demonstrate the slow-down principle, we use a simple example of two reactions in series \( A \xrightarrow{k_1} B \xrightarrow{k_2} C \). The two reactions are assumed to obey first-order kinetics, as written in Eqs (6-8).

\[
\frac{dA}{dt} = -k_1 A 
\] (6)

\[
\frac{dB}{dt} = k_1 A - k_2 B 
\] (7)

\[
\frac{dC}{dt} = k_2 B 
\] (8)

Based on this physical model, the concentration profiles of the three species are simulated. An ANNs-based surrogate is iteratively refined by sequential sampling. As Figure 6a indicates, with more data added, the fitting performance improves (MSE decreases). Meanwhile, the decreasing rate of MSE becomes slower (Figure 6a) and the absolute value of relative slope
tends to be smaller (Figure 6b). Following this, the added ratio decreases (Figure 6c) as well as the same trend is indicated for the sampling rate (Figure 6d). The algorithm is terminated when $|\text{relative slope}_i| < 0.02$, and 25 data points in total are sampled until the final iteration.

Figure 6. ‘Slow-down’ principle for sampling rate for surrogate construction of $A \xrightarrow{k_1} B \xrightarrow{k_2} C$ (Scaling factor for the added ratio function is set as $S = 1.2$).

To evaluate the performance of surrogate in a more direct way, we simulate the concentration profiles of three species using physical model and surrogate model, respectively. Figure 7 shows that the regression performance of the iteratively-refined surrogate gradually improves with iteration. The surrogate obtained in the final iteration (Iteration 8) can perfectly model the original concentration profiles of the three species.
Figure 7. The regression performance of ANN surrogate for the concentration profiles of three species regarding the $A \xrightarrow{k_1} B \xrightarrow{k_2} C$ reaction. Each iteration adds new data points to refine the surrogate. The performance of surrogate gradually improves from Iteration 1, 2, 4 to Iteration 8. Solid lines for the simulation by the physical model, while dashed lines for the simulation by the surrogate model. Surrogate is built based on the sampled data points.

For each iteration, the training data is fitted 60 times with the random structures of ANNs. The boxplot is used to show the distribution of the MSE values of these ANN structures based on the validation dataset. The selected ANN is used to predict the outputs of the test dataset. As shown in Figure 8, the values of MSE are decreasing for both validation and test dataset. In other words, the fitting performance gradually improves with the more sampled data added in.
The slow-down principle can be well reproduced, which can be referred to an example of peaks function in Supplementary Information (SI, Figure S1-S2). Sequential sampling is performed four times on the peaks: the sampling trends are similar for the four times and the number of total sampled data points are close to each other (between 190 and 220).

4. Case studies

Two case studies come from two processes in carbon capture and utilization (CCU): pressure swing adsorption (PSA) and Gas-to-Liquids (GTL), which starts from combined reforming (steam + CO₂) of natural gas.

4.1 Case study 1: surrogate generation for PSA, a dynamic process built in Dymola

Pressure swing adsorption (PSA) is a cyclic dynamic process for gas separation. As to achieve the objective of net-zero 2050, PSA is regarded as a promising technology for CO₂ capture from fossil fuel-based processes. Through continuously varying pressure, adsorption switches with desorption for all the process periods (Figure 9). Eventually, PSA reaches a cyclic steady state (CSS), where consecutive cycles have the same profile.
The physical model, dynamic simulation and a typical input-output data point of PSA are presented in the Section S2 of Supplementary Information. The physical model of PSA is complex due to its stiff and nonlinear partial differential equations (PDEs). The simulation based on the PSA physical model is time-demanding because the simulation result is only meaningful under CSS. In other words, one PSA simulation usually needs to be executed for a long time to reach CSS and then obtain one meaningful simulation output. A surrogate can solve the issues mentioned above, but we still need to minimize the time cost of data generation for the surrogate establishment. We program the physical model of PSA in Dymola and use MATLAB to run Dymola to collect inputs/outputs data automatically. Table 1 describes the inputs and outputs for the PSA system.

Table 1. Description of input and output variables for PSA surrogate.

<table>
<thead>
<tr>
<th>Input variables</th>
<th>Range</th>
<th>Unit</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{\text{ads}} )</td>
<td>20-100</td>
<td>[s]</td>
<td>Duration of adsorption stage</td>
</tr>
<tr>
<td>( t_{\text{bd}} )</td>
<td>30-200</td>
<td>[s]</td>
<td>Duration of blowdown stage</td>
</tr>
<tr>
<td>( t_{\text{evac}} )</td>
<td>30-200</td>
<td>[s]</td>
<td>Duration of evacuation stage</td>
</tr>
<tr>
<td>( P_{I} )</td>
<td>0.07-0.5</td>
<td>[bar]</td>
<td>Setpoint of intermediate pressure</td>
</tr>
<tr>
<td>( P_{L} )</td>
<td>0.005-0.05</td>
<td>[bar]</td>
<td>Setpoint of low pressure</td>
</tr>
<tr>
<td>( v_{\text{feed}} )</td>
<td>0.1-2</td>
<td>[m/s]</td>
<td>Inlet flowrate</td>
</tr>
<tr>
<td>( y_{\text{CO}_2} )</td>
<td>0.02-0.06</td>
<td>[-]</td>
<td>Inlet molar fraction of ( \text{CO}_2 )</td>
</tr>
</tbody>
</table>
Output variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recovery</td>
<td>[-]</td>
<td>Recovery rate of CO₂</td>
</tr>
<tr>
<td>Purity</td>
<td>[-]</td>
<td>Purity of CO₂ in the product flow</td>
</tr>
<tr>
<td>Energy</td>
<td>[kWh/ton-CO₂]</td>
<td>Energy usage per ton CO₂ captured</td>
</tr>
</tbody>
</table>

In this case, PSA is applied to capture the CO₂ from the flue gas of a natural gas power plant. Due to the low CO₂ concentration in the flue gas (~4%), one PSA unit cannot guarantee the required purity (90% for carbon capture). PSA in series can be an option. In this work, we mainly focus on performance of the first PSA unit, where recovery of CO₂ is supposed to be high enough. The purity of CO₂ should improve as well. A trade-off relationship is reported between recovery and purity, so the CO₂ purity cannot be too high given the priority on recovery. Therefore, we trained an SVM classifier to choose the samples with high recovery (higher part, >50%) and purity (middle part, 25%-75%). The classifier's performance can be referred to in Figure 10, and eventually, only 23% of the initial-sampled data is selected to fall in the desired space.

![Figure 10. Classification performance for PSA.](image)

The slow-down sampling is applied to collect data iteratively. Figure 11 indicates that the regression improvement is not significant after 10 iterations, and the corresponding sampling rate decreases in the meanwhile. The relative slope fluctuates significantly along with the iteration (Figure 11b), while the added ratio function helps smooth the fluctuation (Figure 11c). Eventually, we terminate the algorithm after 50 iterations, since the MSE value seems to hardly decrease after the 40th iteration.
Figure 11. ‘Slow-down’ principle for the sequential sampling of PSA.

The efficiency of the proposed workflow can be compared to the case with equal sampling and no classifier. Figure 12a indicates that slow-down sampling has a much higher possibility for earlier termination. When similar fitting performance is reached (e.g., MSE = 2.3e-3), much fewer data points are collected by the slow-down sampling (7372 data points) than by equally sampling (10500 data points). Figure 12b illustrates the effect of the SVM classifier. The classifier is trained by the dataset in the first iteration. The average time per data generated is assumed to be kept the same as the 1st iteration if no classifier applies (as the dashed line in Figure 12b). Herein, the data effectiveness without a classifier is around 23%. By contrast, the classifier can significantly improve the data effectiveness by avoiding undesired inputs for the data generation, thus reducing the average time per individual data from 375 s (1st iteration) to 65 s (50th iteration).
The effect of the two principles can be merged to improve efficiency of surrogate generation for PSA. As shown in Figure 13, if equally sampling without classifier is applied, the time spent on surrogate generation for PSA is 3.9e6 s, which can be reduced by 88% if the two principles apply (4.8e5 s).

A separate dataset was used to test the performance of the iteratively refined surrogate for PSA. We employed the boxplot for the relative errors between the surrogate predictions and the rigorous simulations for the three outputs – recovery of CO₂, purity of CO₂ in the product flow,
and energy consumption of the system. As shown in Figure 14, all outputs can be well predicted with relative errors smaller than 5%.

Figure 14. Prediction performance of the final surrogate for PSA.

4.2 Case study 2: surrogate generation for GTL, a steady-state flowsheet in AspenPlus

Gas-to-Liquids is a classical chemical process for production of fuels.\textsuperscript{47,48} We built a flowsheet in AspenPlus (detailed information in SI, Figure S3). The process starts with the combined reforming (steam + CO\textsubscript{2}) of natural gas to syngas, followed by Fischer-Tropsch (FT) synthesis for fuels. A recycle stream is split: one for reforming, the other for FT reactor.

The combined reforming, FT (kinetics and chain growth probability for products distribution), simulation (convergence) and a typical input-output data point of GTL system are presented in the Section S3 of Supplementary Information. To seek the optimal operating condition, we may optimize some decision variables under uncertainty (input for surrogate) to evaluate the corresponding process performance (output for surrogate) as shown in Table 2.

Table 2. Description of input and output variables for the surrogate building of GTL

<table>
<thead>
<tr>
<th>Input variables</th>
<th>Range</th>
<th>Unit</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>F\textsubscript{CO\textsubscript{2}}</td>
<td>72-8200</td>
<td>[kmol/h]</td>
<td>Inlet flowrate of CO\textsubscript{2}</td>
</tr>
<tr>
<td>F\textsubscript{CH\textsubscript{4}}</td>
<td>*design spec.</td>
<td>[kmol/h]</td>
<td>Inlet flowrate of CH\textsubscript{4}</td>
</tr>
<tr>
<td>x\textsubscript{CH\textsubscript{4}}</td>
<td>0.94-0.96</td>
<td>[-]</td>
<td>Inlet molar fraction of CH\textsubscript{4}</td>
</tr>
<tr>
<td>T\textsubscript{FT}</td>
<td>215-265</td>
<td>[°C]</td>
<td>Temperature in FT reactor</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
<td>Unit</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>-------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>$P_{\text{FT}}$</td>
<td>15-50</td>
<td>[bar]</td>
<td>Pressure in FT reactor</td>
</tr>
<tr>
<td>$N_{\text{trays}}$</td>
<td>45-65</td>
<td>[-]</td>
<td>No. of trays in distillation column</td>
</tr>
<tr>
<td>$T_{\text{reformer}}$</td>
<td>750-1000</td>
<td>[°C]</td>
<td>Temperature in reformer reactor</td>
</tr>
<tr>
<td>$P_{\text{reformer}}$</td>
<td>3-7</td>
<td>[bar]</td>
<td>Pressure in reformer reactor</td>
</tr>
<tr>
<td>Split&lt;sub&gt;vent&lt;/sub&gt;</td>
<td>0.001-0.2</td>
<td>[-]</td>
<td>Split fraction to vent stream (the other to recycle)</td>
</tr>
<tr>
<td>Split&lt;sub&gt;FT&lt;/sub&gt;</td>
<td>0.01-0.99</td>
<td>[-]</td>
<td>Split fraction to FT (the other to reformer)</td>
</tr>
</tbody>
</table>

Output variables

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{\text{gasoline}}$</td>
<td>[kmol/h]</td>
<td>Product flowrate of gasoline</td>
</tr>
<tr>
<td>$F_{\text{diesel}}$</td>
<td>[kmol/h]</td>
<td>Product flowrate of diesel</td>
</tr>
<tr>
<td>$F_{\text{gas}}$</td>
<td>[kmol/h]</td>
<td>Product flowrate of light HCs [C&lt;sub&gt;1&lt;/sub&gt;-C&lt;sub&gt;4&lt;/sub&gt;]</td>
</tr>
<tr>
<td>$F_{\text{H}<em>2\text{O}</em>\text{net}}$</td>
<td>[kmol/h]</td>
<td>Net flowrate of process water</td>
</tr>
<tr>
<td>$\text{vent}_\text{CO}_2$</td>
<td>[kmol/h]</td>
<td>Flowrate of CO&lt;sub&gt;2&lt;/sub&gt; in the vent</td>
</tr>
<tr>
<td>Electricity</td>
<td>[Gcal/h]</td>
<td>Electricity usage for pumps and compressors</td>
</tr>
<tr>
<td>$U_{\text{air}}$</td>
<td>[Gcal/h]</td>
<td>Cooling utility by air</td>
</tr>
<tr>
<td>$U_{1000}$</td>
<td>[Gcal/h]</td>
<td>Heating utility by 1000 °C fuel gas</td>
</tr>
<tr>
<td>$U_{\text{Steam}}$</td>
<td>[Gcal/h]</td>
<td>Heating utility by high pressure steam</td>
</tr>
<tr>
<td>$U_{\text{water}}$</td>
<td>[Gcal/h]</td>
<td>Cooling utility by cooling water</td>
</tr>
</tbody>
</table>

* In Aspen Plus, $F_{\text{CH}_4}$ is determined by a flowsheet option (design specification). Varying $F_{\text{CH}_4}$ and steam flowrate can achieve the desired syngas ratio, where $\text{H}_2:\text{CO} = 2-2.2$ is preferred for the FT reaction.

Aspen simulations suffer from non-convergence issues when improper operating conditions are given, or the recycle stream is set too tight. Such problems also occur in our case. Initially, MATLAB is used to run the Aspen model in the first two iterations, where nearly 20% of simulations crashed due to non-convergence issues (Figure 15). The inputs/outputs in the first two iterations are used to train an SVM classifier. The obtained classifier was then applied here to screen out the potential non-converged inputs in the following iterations, thus improving the percentage of effective data (Figure 15). In the final iteration, less than 9% of inputs resulted in ineffective data, with the pre-treatment of SVM. Notably, the non-converged simulations in Aspen Plus usually takes a long time to stop but deliver invalid outputs. The trained classifier lowered the possibility for the simulator to take the ‘bad’ inputs and tremendously reduced the average time on generating an individual data output from 61 s (1~2 iteration) to 33 s (40<sup>th</sup> iteration).
Figure 15. Improvement of data effectiveness by the classifier.

Figure 16. ‘Slow-down’ principle for the sequential sampling of GTL.

The slow-down sampling is applied to collect data iteratively. The relative slope in Figure 16b fluctuates more significantly than the case study of PSA, see Figure 11. That is probably because the GTL has more outputs to fit, and the regression is more complex than the case of
PSA. The observed trend in Figure 16a indicates that the regression improvement is not significant after the 25th iteration. Eventually, we terminate the algorithm after 40 iterations to avoid the unnecessary computation cost.

The two principles can separately improve the sampling efficiency for building surrogates for GTL. As shown in Figure 17a, the slow-down sampling possesses a higher chance for an earlier termination than the equally-sampling, to achieve a similar fitting performance (MSE=1e-4) with fewer data points (slow-down for 13112 data points vs. equally sampling for 18287 data points). The trend in Figure 17b shows that the SVM classifier is able to reduce the average time spent on individual points by nearly 50%.

![Figure 17](image)

Figure 17. The contribution of slow-down sampling and classifier for the efficiency improvement of GTL surrogate construction: slow-down sampling has much a higher possibility of collecting fewer data points than equal sampling; a classifier can reduce the average time per data generated. (Clarification for the dashed line in b: since no classifier is used in the first two iterations, we assume that their average value for a single data generation will be the time in the following iterations.)

Overall, the effect of two principles can be merged to improve the efficiency of surrogate generation for GTL. As shown in Figure 18, the total time spent on surrogate generation for GTL can be reduced by 60% compared to equally sampling without a classifier.
Figure 18. Comparison of total time spent on surrogate generation for GTL between [equally sampling + no classifier] and [slowdown sampling + classifier]. Total time is the sum of time spent on data generation and surrogate training.

A separate test dataset is used to evaluate the performance of the surrogate obtained in the final iteration. We employ the boxplot for the relative errors between the surrogate predictions and the rigorous simulations for the 10 outputs. As shown in Figure 19, most outputs can be well predicted with relative errors smaller than 5%, and some are even smaller than 1%, e.g., the mass flowrate for the fuel products. The fitting for the utility is not ideal, and the relative error of the electricity consumption can go up to 15%. This is probably due to the insufficient feature selection for utility fitting. For example, the electricity consumption is related to the units of pumps and compressors, while no relevant features are selected into the inputs for the surrogate training. Meanwhile, no features related to heat exchangers are chosen, so the fitting performance of the utility is not as good as the mass flowrates. However, the motivation behind surrogate is to build a reduced-order model to replace the original full-order physical model, and thus sacrificing partial accuracy is unavoidable but acceptable.
6 Conclusions and outlook

This work developed an efficient workflow for surrogate generation for engineering systems (typically \( t_{data} \gg t_{training} \)) regarding data quality and quantity. The proposed workflow can reduce the time for data generation via two principles: (1) a classifier is trained to avoid the undesired design space for data generation and improve the data quality, (2) a slow-down principle is applied to data generation, and meanwhile, more time is allocated to surrogate training (but still negligible compared to data generation). The slow-down sampling significantly lowers the possibility for over-sampling (data quantity). With the proposed workflow, the time of surrogate generation is shown to be reduced by 88% for PSA and 60% for GTL case studies.

The primary goal of this work was to investigate the influence of the sampling rate for the surrogate generation. Sampling was desired to be of a homogenous type, which may be disturbed by exploitation-based methods. As a result, we only considered exploration-based methods in our current methodology workflow. In the future, to further enhance sampling efficiency, the exploration in sampling the global design space should be well balanced with the exploitation in sampling the nonlinear/complex local areas. Another work that can be done is to determine better termination criteria: we tried to stop the algorithm when the MSE difference between two consecutive iterations approached 0, or the slope approached 0, but the fluctuation of MSE values always existed for the case study of GTL or PSA, which made the tolerance value for termination hard to set. One solution could be to apply feature selection techniques (i.e., automatically adjust input variables) to improve fitting performance and reduce its fluctuation during sequential sampling.
Additionally, this work lays foundation for the superstructure optimization of an extensive CCU system. The two case studies presented in this work belong to its two process options. The sub-processes of CCU are usually modelled in different process simulators, which cause inconvenience for overall simulation or optimization. This work can replace the original sub-process models of CCU with the machine learning-based surrogates, following by overall optimization in a high-level interactive platform.

Competing Interests

Authors declare no competing financial interest.

Acknowledgments

ZH and CZ acknowledge financial support from Cambridge Trust and the Chinese Scholarship Council. ZH’s final-year PhD was partially funded by the Sustainable Reaction Engineering research group of Prof. Lapkin. AAL acknowledges funding from the National Research Foundation (NRF), Prime Minister’s Office, Singapore, under its Campus for Research Excellence and Technological Enterprise (CREATE) program as a part of the Cambridge Centre for Advanced Research and Education in Singapore Ltd (CARES Ltd).

References


Supplementary Information

Efficient Surrogates Construction of Chemical Processes: Case studies on Pressure Swing Adsorption and Gas-to-Liquids

Zhimian Hao,1 Chonghuan Zhang1 and Alexei A. Lapkin1,2

1Department of Chemical Engineering and Biotechnology, University of Cambridge, Cambridge CB3 0AS, UK
2Cambridge Centre for Advanced Research and Education in Singapore Ltd, 1 Create Way, CREATE Tower #05-05, 138602, Singapore

Table of Contents of the Supporting Information

S1. Reproducibility of the slow-down sampling for peaks function ........................................28

S2. Description of a PSA System ..........................................................................................30
  S2.1. Model equation ........................................................................................................30
  S2.2. Boundary conditions ................................................................................................33
  S2.3. Simulation of PSA ....................................................................................................33
  S2.4. An example of one input-output data point for PSA surrogate.................................35

S3. The description of GTL process ......................................................................................37
  S3.1. Why combined reforming? .......................................................................................38
  S3.2. Fischer-Tropsch (FT) reaction ..................................................................................38
    S3.2.1. FT kinetics ........................................................................................................38
    S3.2.2. Product distribution ...........................................................................................39
  S3.3. Flowsheet establishment of GTL in Aspen Plus.........................................................42
    S3.3.1. Combined reforming section ..............................................................................42
    S3.3.2. Fischer-Tropsch section ....................................................................................43
    S3.3.3. Separation and recovery ....................................................................................43
    S3.3.4. Utility and its integration ...................................................................................43
    S3.3.5. Convergence strategy .........................................................................................44
  S3.4. Simulation result of Aspen Plus for GTL ..................................................................45
  S3.5. An example of one input-output data point for GTL surrogate ..................................46

References ................................................................................................................................46
S1. Reproducibility of the slow-down sampling for peaks function

Peaks function is used as a numerical example to demonstrate the reproducibility of the slow-down sampling. Peaks function consists of two variables (Figure S1), which can be accessed by the function ‘peaks’ in MATLAB. The mathematical form is as equation S.1.

\[
Z = 3(1 - x)^2 \exp(-x^2) - (y + 1)^2 - 10\left(\frac{x}{5} - x^3 - y^5\right)\exp(-x^2 - y^2) - \frac{1}{3}\exp(-(x + 1)^2 - y^2)
\]  

S.1

The slow-down sampling is applied to generate data sequentially for the surrogate construction (Figure S2). The termination criterion is set as \(\text{relative slope} < 0.02\), when the slope is regarded as 0. With four trails, the total number of sampled data falls between 190 and 220. The fitting for the peaks function is relatively more manageable than a typical engineering system, so the termination criterion of peaks function can be well determined. By contrast, we terminate the sampling when the algorithm reaches a maximum iteration for the two case studies used in this work.

Figure S1. Sampling for the surrogate building of peaks function.
Figure S2. Reproducibility of slow-down sampling for peaks function.
S2. Description of a PSA System

![Figure S3. Description of a four-stage PSA for CO$_2$ capture.](image)

S2.1. Model equation

The PSA process is operated in a cyclic mode between absorbing desired gas species at a higher pressure and releasing them at a lower pressure. A typical PSA consists of four stages: the gas mixture flows into the column at the high pressure ($P_H$); then the undesired gas species are extracted out due to their weaker interactions with absorbents, while the column pressure decreases to an intermediate pressure – the blowdown pressure ($P_L$); further, the column continues to be evacuated to an even lower pressure – evacuation pressure ($P_E$) and the desired product is expected to be extracted in the meanwhile; following this, the column is fed with the gas mixture until the high pressure ($P_H$). In some cases, the high pressure ($P_H$) is set to vacuum level to pursue a higher absorption capacity and lower energy consumption, thus resulting in vacuum swing adsorption (VSA). These four stages make up one cycle of PSA, and the repeating cycles purify the desired product in a cyclic way. This work is mainly focused on a four-stage pressure swing adsorption (PSA) process model for CO$_2$ capture (Figure S3) as reported in Haghpanah’s work. A column packed with solid adsorbent is considered, and the following assumptions are used to derive the balance equations (Equations S.2 - S.20):

1. A one-dimensional dispersed plug flow model is applied to simulate the bulk fluid flow in the axial direction.

2. No mass, temperature, or pressure gradient exists in the radius direction.
(3) Ideal gas law is applied for the state of the gas phase.

(4) Ergun equation is used for the pressure drop in the axial direction.

(5) The thermal equilibrium between the gas and solid phase is established instantaneously.

(6) Diffusion through adsorbent pores is considered as molecular diffusion in the macropores.

(7) Multisite Langmuir model is applied to calculate the solid phase saturation loading.

Total mass balance in the gas phase:

\[
\frac{1}{P} \frac{\partial P}{\partial t} - \frac{1}{T} \frac{\partial T}{\partial t} - \frac{T}{P} \frac{\partial (P/T)}{\partial z} = \frac{RT}{P} \frac{1 - \varepsilon}{\varepsilon} \sum_{i=1}^{n_{\text{comp}}} q_i \frac{\partial q_i}{\partial t} \tag{S.2}
\]

Component mass balances \((n_{\text{comp}} - 1)\) in gas phase:

\[
\frac{\partial y_i}{\partial t} + y_i \frac{\partial P}{\partial t} - y_i \frac{\partial T}{\partial t} = \frac{T}{P} D_r \frac{\partial (P \partial y_i}{\partial z} \frac{\partial y_i}{\partial z} - \frac{RT}{P} \frac{1 - \varepsilon}{\varepsilon} \frac{\partial q_i}{\partial t} \tag{S.3}
\]

Component mass balance in the solid phase:

\[
\frac{\partial q_i}{\partial t} = k_i (q_i^* - q_i) \tag{S.4}
\]

Energy balance inside column:

\[
\left[ \frac{1 - \varepsilon}{\varepsilon} \left( \rho_s C_{p,s} + C_{p,a} \sum_{i=1}^{n_{\text{comp}}} q_i \right) \right] \frac{\partial T}{\partial t} = \frac{K_z}{\varepsilon} \frac{\partial^2 T}{\partial z^2} - \frac{C_{p,g}}{R} \frac{\partial (vP)}{\partial t} - \frac{C_{p,g}}{R} \frac{\partial P}{\partial t} - \frac{1 - \varepsilon}{\varepsilon} C_{p,a} T \sum_{i=1}^{n_{\text{comp}}} \frac{\partial q_i}{\partial t}\right] + \frac{1 - \varepsilon}{\varepsilon} \sum_{i=1}^{n_{\text{comp}}} \left( -\Delta H_i \frac{\partial q_i}{\partial t} \right) - \frac{2h}{\varepsilon r_{\text{in}}} (T - T_w) \tag{S.5}
\]

Energy balance in the column wall:

\[
\rho_w C_{p,w} \frac{\partial T_w}{\partial t} = K_w \frac{\partial^2 T_w}{\partial z^2} + \frac{2 r_{\text{in}} h_{\text{in}}}{r_{\text{out}}^2 - r_{\text{in}}^2} (T - T_w) - \frac{2 r_{\text{out}} h_{\text{out}}}{r_{\text{out}}^2 - r_{\text{in}}^2} (T_w - T_a) \tag{S.6}
\]

Pressure drop by Ergun Equation:

\[
\frac{\partial P}{\partial z} = -\frac{150}{4} \frac{1 - \varepsilon}{\varepsilon} \left( 1 - \frac{\varepsilon}{\mu} \right)^2 \mu v \tag{S.7}
\]

The mass transport coefficient is given by
\[ k_i = \frac{c_i}{q_i^*} \frac{15\varepsilon_p D_p}{r_p^2} \]  \hspace{1cm} (S.8)

\( q_i^* \) is obtained from a dual-site Langmuir model:

\[ q_i^* = \frac{q_{sd,i} b_i c_i}{1 + \sum_{l=1}^{n_{comp}} b_l c_i} + \frac{q_{sd,i} d_i c_i}{1 + \sum_{l=1}^{n_{comp}} d_i c_i} \]  \hspace{1cm} (S.9)

where \( b_i, d_i \) are the solid phase saturation loadings of sites 1 and 2, respectively. They can be calculated based on Arrhenius-type temperature dependence:

\[ b_i = b_{0,i} \exp \left( -\frac{U_{b,i}}{RT} \right) \]  \hspace{1cm} (S.10)

\[ d_i = d_{0,i} \exp \left( -\frac{U_{d,i}}{RT} \right) \]  \hspace{1cm} (S.11)
S2.2. Boundary conditions

The model equations for four stages of PSA are identical, while different stages are distinguished from each other by their boundary conditions of pressure: Adsorption (Ads), Blowdown (Bd), Evacuation (Evac), and Feed pressurization (Press). In the following equations, $t_{cycle}$ refers to the time duration for a PSA cycle. $t_{cycle}$ is always (re-)initialized as 0 when starting a new PSA cycle.

\begin{align*}
\text{Ads} & \quad t_{cycle} \in [0, t_{ads}] \\
\frac{\partial P}{\partial z}|_{z=0} & = -\frac{150}{4} \frac{1 - \varepsilon^2}{\varepsilon} \mu v_{feed} \quad \text{S.12} \\
P|_{z=L} & = P_{atm} \quad \text{S.13}
\end{align*}

\begin{align*}
\text{Bd} & \quad t_{cycle} \in [t_{ads}, t_{ads} + t_{bd}] \\
\frac{\partial P}{\partial z}|_{z=0} & = 0 \quad \text{S.14} \\
P|_{z=L} & = \frac{1}{P_0} [P_L + (P_H - P_L) \exp^{-\lambda t_{cycle}}] \quad \text{S.15}
\end{align*}

\begin{align*}
\text{Evac} & \quad t_{cycle} \in [t_{ads} + t_{bd}, t_{ads} + t_{bd} + t_{evac}] \\
\frac{\partial P}{\partial z}|_{z=0} & = 0 \quad \text{S.16} \\
P|_{z=L} & = \frac{1}{P_0} [P_L + (P_I - P_L) \exp^{-\lambda t_{cycle}}] \quad \text{S.17}
\end{align*}

\begin{align*}
\text{Press} & \quad t_{cycle} \in [t_{ads} + t_{bd} + t_{evac}, t_{ads} + t_{bd} + t_{evac} + t_{press}] \\
\frac{\partial P}{\partial z}|_{z=0} & = -\frac{150}{4} \frac{1 - \varepsilon^2}{\varepsilon} \mu v_{feed} \quad \text{S.18} \\
\frac{\partial P}{\partial z}|_{z=L} & = 0 \quad \text{S.19}
\end{align*}

S2.3. Simulation of PSA

The model equations are discretized using a finite volume method, and partial differential equations (PDEs) are turned into a set of differential-algebraic equations (DAEs). We build this DAEs system in Dymola, a Modelica platform. If the column is discretized as 30 equal volumes (“30” is recommended based on both accuracy and efficiency by Haghpanah), 1220 equations are generated. In numerical solving, Modelica is an objected-oriented modelling language, and all equations are solved simultaneously. Eventually, the DAEs are implemented and numerically integrated by the Dassl solver using the Modelica language in Dymola 2018. The full model of PSA can be the sum of four stages multiplied by corresponding binary variables (Equation S.20). Through varying binary variables $[Y_1, Y_2, Y_3, Y_4]$, the four stages of the PSA model can be simulated continuously.
\[ PSA = Y_1S_{ads} + Y_2S_{bd} + Y_3S_{evac} + Y_4S_{press} \]

<table>
<thead>
<tr>
<th></th>
<th>( S_{ads} )</th>
<th>( S_{bd} )</th>
<th>( S_{evac} )</th>
<th>( S_{press} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y_1 )</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( Y_2 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( Y_3 )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( Y_4 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table S1. Binary variables for four stages.

After completing four stages, re-initialize the cycle time \( t_{cycle} \) as 0 and then start the simulation of another one cycle of PSA. Consequently, the PSA cycle is simulated iteratively until a cyclic steady state (CSS) is reached. Theoretically, when a CSS is reached, the column profile is expected to be the same between the same step in two subsequent cycles. In the mathematical language, when \[ |x(t) - x(t + t_{cycle})| < \delta \], PSA is deemed to be under CSS. The dynamic simulation of PSA can be found in Figure S4.
Figure S4. The dynamic simulation of PSA.

A full dynamic simulation can contain much more results than what has been shown here, but those results might not be relevant in the process analysis and optimization. This is a motivation to develop a reduced-order surrogate.

S2.4. An example of one input-output data point for PSA surrogate

According to the simulation result as shown in Figure S4, we can collect one input-output data point as shown in Table S2. Obviously, this data point has much less information than a dynamic simulation result in Dymola, but this data point has contained the essential features for the PSA system regarding the
operating condition and the corresponding process output. A sufficient number of such effective data points can be used to train a surrogate for the PSA process.

Table S2. An example of one input-output data point for PSA surrogate.

<table>
<thead>
<tr>
<th>Input variables</th>
<th>Value</th>
<th>Unit</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{\text{ads}}$</td>
<td>94.89</td>
<td>[s]</td>
<td>Duration of adsorption stage</td>
</tr>
<tr>
<td>$t_{\text{bd}}$</td>
<td>122.84</td>
<td>[s]</td>
<td>Duration of blowdown stage</td>
</tr>
<tr>
<td>$t_{\text{evac}}$</td>
<td>189.46</td>
<td>[s]</td>
<td>Duration of evacuation stage</td>
</tr>
<tr>
<td>$P_1$</td>
<td>0.18</td>
<td>[bar]</td>
<td>Setpoint of intermediate pressure</td>
</tr>
<tr>
<td>$P_L$</td>
<td>0.02</td>
<td>[bar]</td>
<td>Setpoint of low pressure</td>
</tr>
<tr>
<td>$v_{\text{feed}}$</td>
<td>0.58</td>
<td>[m/s]</td>
<td>Inlet flowrate</td>
</tr>
<tr>
<td>$y_{\text{CO}_2}$</td>
<td>0.15</td>
<td>[-]</td>
<td>Inlet molar fraction of CO$_2$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output variables</th>
<th>Value</th>
<th>Unit</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recovery</td>
<td>0.8967</td>
<td>[-]</td>
<td>Recovery rate of CO$_2$</td>
</tr>
<tr>
<td>Purity</td>
<td>0.8934</td>
<td>[-]</td>
<td>Purity of CO$_2$ in the product flow</td>
</tr>
<tr>
<td>Energy</td>
<td>150.22</td>
<td>[kWh/ton-CO$_2$]</td>
<td>Energy usage per ton CO$_2$ captured</td>
</tr>
</tbody>
</table>
S3. The description of GTL process

As shown in Figure S5, the Gas-to-Liquid (GTL) process is modelled in Aspen Plus, containing combined reforming, FT synthesis, and product upgrading section. This process starts with the combined reforming (CO$_2$ + H$_2$O) of natural gas to syngas, followed by FT synthesis for fuels. Since the upgrading section has little influence on the overall performance, we use a distillation column to simplify it. In this work, more attention is given to the reforming and FT, as well as the flowsheet building in Aspen Plus and its simulation convergence.

Figure S5. Flowsheet for GTL built in Aspen Plus.
S3.1. Why combined reforming?

Reforming is a process of converting natural gas (NG) to syngas (H₂ and CO). The relevant approaches range from partial oxidation reforming, autothermal reforming, steam reforming to dry reforming.⁵ The main difference among them is the oxidants. The former two approaches require O₂ as the oxide agents, while the latter two technologies employ steam and CO₂. The most mature technology is steam reforming, which has been commercialized for H₂ production for decades. However, none of the individual reforming technology can provide the desired syngas ratio for FT synthesis. Thus, separation technology is required to remove the excessive component. Alternatively, these reforming options can be combined to adjust the ratio between H₂ and CO. Baltrusaitis et al. compare five combination configurations and deliver the conclusion that the combination of steam reforming and dry reforming is economically favored over other options.⁶ Also, such a combination can reduce carbon footprint by 67% over conventional steam reforming.⁷ Meanwhile, the existence of steam is preferred because the dry reforming suffers severe coke formation.⁸ Another novelty of combined reforming (CO₂ + steam) is that CO₂ is regarded as one of the raw materials. Thus, unlike the conventional process (autothermal reforming or steam reforming), a CO₂ separation unit is not required before the reforming unit in the process design. Based on the thermodynamic analysis,⁹ the combined reforming has the potential to fully convert the methane while maintaining a high yield of H₂ at 88%. Therefore, we consider the combined reforming (CO₂ + steam) for syngas production in this work.

\[
\begin{align*}
\text{Combined reforming} & \quad \text{CH}_4 + \text{H}_2\text{O} \rightarrow 3\text{H}_2 + \text{CO} \\
& \quad \text{CH}_4 + \text{CO}_2 \rightarrow 2\text{H}_2 + 2\text{CO}
\end{align*}
\]

S3.2. Fischer-Tropsch (FT) reaction

Fischer-Tropsch (FT) has a long industrial history of producing high-quality fuels.¹⁰,¹¹ The FT products are commercially favored mainly due to S-free and N-free. FT can be classified into two types: low-temperature FT (LTFT) based on cobalt catalysts and high-temperature FT (HTFT) based on iron catalysts. The FT reaction is highly exothermic, and a lower temperature can improve the final conversion regarding the thermodynamic equilibrium. By contrast, the high temperature tends to terminate the carbon chains, resulting in shorter hydrocarbons.¹² In recent years, LTFT is more popular because it can produce long-chain hydrocarbons and is less energy-intensive.¹² Additionally, most latest research works focus on the cobalt catalysts,⁴,¹¹,¹³–¹⁶ which are more active and widely used in industry. Hence, LTFT is considered in this work. For LTFT, the desired syngas ratio falls in the range of 2 – 2.2 for H₂ : CO⁴,¹⁰,¹¹ and the reaction temperature is reported to range from 200 to 270 °C¹⁰,¹⁷,¹⁸ as well as pressure in 15 – 50 bar.¹⁷,¹⁹

\[
(2n + 1)\text{H}_2 + n\text{CO} \rightarrow C_n\text{H}_{2n+2} + n\text{H}_2\text{O}
\]

S3.2.1. FT kinetics

For FT kinetics, Yates et al. developed a simple but reliable expression 30 years ago for the consumption rate of CO as Equation S.21.²⁰

\[
r_{\text{CO}} = \frac{F \ a \ P_{\text{CO}} \ P_{\text{H}_2}}{(1 + b \ P_{\text{CO}})^2}
\]  

S.21

where, \( F \) : catalyst improvement factor since Yates’ kinetics proposed in 1991;

\( P_i \) : partial pressure of component \( i \);
a: reaction rate coefficient;  
b: adsorption coefficient.

Vervloet et al. suggested the catalyst improvement factor could fall in the range of $1 - 10^{15}$ while, a decade ago, Guettel et al. claimed that a more promising value could be up to 20 in the future.$^{14}$ Herein, we assume that the value of the catalyst improvement factor is 10 for this work. The variables a and b can be expressed by the following equations,

$$
a = a_0 \exp \left[ \frac{E_a}{R} \left( \frac{1}{493.15} - \frac{1}{T} \right) \right] \quad \text{S(1)}$$

$$
b = b_0 \exp \left[ \frac{\Delta_b H}{R} \left( \frac{1}{493.15} - \frac{1}{T} \right) \right] \quad \text{(9)}$$

where, $a_0$: pre-exponential factor;  
$E_a$: activation energy;  
$b_0$: adsorption coefficient;  
$\Delta_b H$: adsorption enthalpy.

Table S3 gives all the relevant values used in this work.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
<th>Unit</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F$</td>
<td>10</td>
<td>-</td>
<td>Catalyst improvement factor</td>
</tr>
<tr>
<td>$a_0$</td>
<td>8.9e-3</td>
<td>mol $\cdot$ s$^{-1} \cdot$ kg$^{-1} \cdot$ bar$^{-2}$</td>
<td>Pre-exponential factor</td>
</tr>
<tr>
<td>$E_a$</td>
<td>3.7e4</td>
<td>J $\cdot$ mol$^{-1}$</td>
<td>Activation energy</td>
</tr>
<tr>
<td>$b_0$</td>
<td>2.2</td>
<td>bar$^{-1}$</td>
<td>Adsorption coefficient</td>
</tr>
<tr>
<td>$\Delta_b H$</td>
<td>-6.8e4</td>
<td>J $\cdot$ mol$^{-1}$</td>
<td>Adsorption enthalpy</td>
</tr>
</tbody>
</table>

S3.2.2. Product distribution

Yates’ kinetics only describes the consumption rate of CO, but no more information is given concerning the products. The Anderson-Schulz-Flory mechanism can be used to explain the distribution of FT products.$^{11}$ Since most FT products are linear hydrocarbons, the reaction can be regarded as the polymerization process to grow a long chain. Figure S6a demonstrates how a small carbon chain extends to a long chain. The chain growth probability, $\alpha$, is a parameter to denote whether the chain continues to grow or not: $\alpha$ for growth, whereas $(1 - \alpha)$ for termination. Figure S6b shows how $\alpha$ can determine the product distribution.
Figure S6. (a) Chain growth by the Anderson-Schulz-Flory mechanism. (b) The product distribution based on the $\alpha$.\textsuperscript{11}

Typically, FT is used to produce long hydrocarbons, which can be used as waxes or be easily cracked for short chains. Hence, the value of $\alpha$ should be high enough (>0.9). In our work, we use 0.93 for it. We plot the corresponding product distribution as Figure S7.

Since the properties of hydrocarbons (HCs) are similar, we use several components to represent the whole range of HCs for simplification during the simulation as the Table S4. Based on the distribution of FT products, the molar fractions of representative HCs can be calculated from the sum of molar fraction of corresponding FT products. In this work, we use C$_6$H$_{18}$ to approximate gasoline, while C$_{16}$H$_{34}$ is regarded as diesel. For further simplification, wax is not taken into consideration in the system.

![Figure S7. Molar distribution of FT products when $\alpha = 0.93$.](image)

**Table S4. Components used in the simulation.**

<table>
<thead>
<tr>
<th>C range</th>
<th>Molar fraction</th>
<th>Representations</th>
</tr>
</thead>
<tbody>
<tr>
<td>C$_1$</td>
<td>0.07</td>
<td>CH$_4$</td>
</tr>
<tr>
<td>C$_2$~C$_4$</td>
<td>0.18</td>
<td>C$_3$H$_8$</td>
</tr>
<tr>
<td>C$<em>5$~C$</em>{12}$</td>
<td>0.30</td>
<td>C$<em>6$H$</em>{18}$ (gasoline)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td>---------</td>
</tr>
<tr>
<td></td>
<td>0.45</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$C_{16}H_{34}$ (diesel)</td>
</tr>
<tr>
<td>Total</td>
<td>1.00</td>
<td></td>
</tr>
</tbody>
</table>
S3.3. Flowsheet establishment of GTL in Aspen Plus

We establish the Aspen Plus process model of GTL mainly by referring to the prior works of Ha et al., Lee et al., and Zhang et al. To deal with a petrochemical system like GTL, Peng-Robinson is recommended as the thermodynamic method.

S3.3.1. Combined reforming section

In the reforming section, GTL starts with NG, water and CO₂. A typical composition of NG can be referred to Bao’s work (Table S5). The water is heated to steam before reforming. In the pre-reformer, all the carbon components are converted to CO and CH₄. In the reformer, CH₄ is converted to syngas with the assistance of CO₂ and steam. The reforming is performed at a high temperature between 700 and 1000 °C, so it is assumed to reach equilibrium. In Aspen Plus, the reformer is modelled by an RGibbs reactor, where the total Gibbs energy is minimized to reach the equilibrium (‘Restricted Chemical Equilibrium’ is set for the combined reforming reactions). A flowsheet option is set to vary the flowrate of H₂O and NG to guarantee the ratio of CO : H₂ falls in a range of 2 – 2.2 in the reformer outlet. Before the FT section, the mixed stream is cooled down and split into a gas stream (mainly syngas and CO₂) and a liquid stream (primarily water).

Table S5. The compositions of NG (CH₄% is regarded as uncertain in this work. *When CH₄% changes, the ratio of C₂H₆ : C₃H₈ : CO₂ : N₂ is assumed to keep unchanged.).

<table>
<thead>
<tr>
<th>Components</th>
<th>Mol %</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH₄</td>
<td>95.39</td>
<td>94 – 96</td>
</tr>
<tr>
<td>C₂H₆</td>
<td>3.91</td>
<td>*</td>
</tr>
<tr>
<td>C₃H₈</td>
<td>0.03</td>
<td>*</td>
</tr>
<tr>
<td>CO₂</td>
<td>0.59</td>
<td>*</td>
</tr>
<tr>
<td>N₂</td>
<td>0.08</td>
<td>*</td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>
S3.3.2. Fischer-Tropsch section

In the FT section, the syngas is pressurized before entering the FT reactor. RPlug is chosen to simulate the multi-tubular fixed-bed reactor for FT in Aspen Plus. Yates’ kinetics is used for the overall consuming rate of CO ($r_{CO}$). Since the properties of hydrocarbons (HCs) are similar, we employ four reactions (R1~R4) to represent the whole range of HCs for simplification during the simulation as the Table S6. The sum of CO consuming rates of individual reactions should be equal to the overall consuming rate of CO ($r_{CO}$). Then the individual reaction rates are obtained from the reaction stoichiometry. CO$_2$ is reported not to react on the Cobalt-based catalyst and can be regarded as an inert gas in the FT reaction.\cite{16,23} Thus, no reaction associated with CO$_2$ is listed in the FT reactor. Based on the information mentioned above, we inserted the kinetics into RPlug.

<table>
<thead>
<tr>
<th>C range</th>
<th>Molar fraction</th>
<th>Representations</th>
<th>Representative reactions</th>
<th>$R_{CO}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>C$_1$</td>
<td>0.07</td>
<td>CH$_4$</td>
<td>R1: 3H$_2$ + CO $\rightarrow$ CH$_4$ + H$_2$O</td>
<td>0.007$r_{CO}$</td>
</tr>
<tr>
<td>C$_2$~C$_4$</td>
<td>0.18</td>
<td>C$_3$H$_8$</td>
<td>R2: 7H$_2$ + 3CO $\rightarrow$ C$_3$H$_8$ + 3H$_2$O</td>
<td>0.053$r_{CO}$</td>
</tr>
<tr>
<td>C$<em>5$~C$</em>{12}$</td>
<td>0.30</td>
<td>C$<em>8$H$</em>{18}$ (gasoline)</td>
<td>R3: 17H$_2$ + 8CO $\rightarrow$ C$<em>8$H$</em>{18}$ + 8H$_2$O</td>
<td>0.234$r_{CO}$</td>
</tr>
<tr>
<td>C$_{13+}$</td>
<td>0.45</td>
<td>C$<em>{16}$H$</em>{34}$ (diesel)</td>
<td>R4: 33H$<em>2$ + 16CO $\rightarrow$ C$</em>{16}$H$_{34}$ + 16H$_2$O</td>
<td>0.706$r_{CO}$</td>
</tr>
<tr>
<td>Total</td>
<td>1.00</td>
<td></td>
<td></td>
<td>$r_{CO}$</td>
</tr>
</tbody>
</table>

S3.3.3. Separation and recovery

Following the FT reactor, a three-phase flash is used to split the mixed stream into gas, liquid HCs and wastewater. As a simplification to the upgrading system, we use a distillation column (RadFrac) to separate gasoline from diesel. For the gas mixture in the simulation, we use an ideal separator (in the real world, PSA can be an option) to recycle all the C$_1$ components. The GTL system contains inert gas (N$_2$), which must be purged (otherwise, it will gradually accumulate in the recycle stream, making the convergence impossible to achieve). Herein, the recycle stream is split to vent (to purge) and C$_1REC$, which will be followed by splitting into reforming section and FT sections, respectively.

S3.3.4. Utility and its integration

The heating utility is supplied by high-pressure steam and fuel gas over 1000 °C. The cooling utility is provided by air and cooling water at room temperature. Pump and compressor are powered by electricity.

The reforming is highly endothermic, and the reforming section can account for over 50% of energy consumption in the overall GTL process. A high temperature is required for the reforming reaction. Thus, the reformer outflow has an extremely high temperature and needs to be cooled before the FT process. We built three heat exchangers to gradually cool down the reformer outflow, while the recycled heat is used to pre-heat the mixed stream to the gas form (>100 °C), an intermediate temperature for pre-reformer (~500 °C) and a high temperature for reformer (700 ~ 1000 °C).

Additionally, the purge stream contains CO and CH$_4$, which will bring in considerable greenhouse emissions if the direct emissions apply. With the assistance of air, a burner is used to deal with these C$_1$ components. An RGibbs reactor @ 600 °C is used to simulate the burner. Due to the exothermic reactions, the burner will release heat, while the waste heat recovery technology\cite{24,25} can be used to
recover the part of burner heating (utilization efficiency is assumed at $\eta_{\text{burner}} = 60\%$) to reduce the heating utility of steam or fuel gas.

S3.3.5. Convergence strategy

The modelled GTL system has two flowsheet options (design specifications) to vary the flow rates of $\text{H}_2\text{O}$ and NG, as well as two recycle streams flowing to reforming/FT sections. As a result, the flowsheet convergence can be pretty tricky. After trials and errors, we would suggest a two-stage strategy for a fast convergence for this case. In the first stage, the convergence of mass flows is performed. *Newton* algorithm is selected to converge the two design specifications, while *Broyden* or *Wegstein* algorithm is used to tear the stream *FTIN*. Because the recycle loop and design specifications can interact with each other in our case, the tear stream has to be solved simultaneously with the two design specifications (*Outside-Simultaneous* is recommended for sequencing). After the mass flow is converged, the heat flow is then converged in the second stage. *Wegstein* algorithm is used to tear the heating streams $H1$, $H2$ and $H3$, which can be made as a convergence block. The heating convergence block is sequenced after the first-stage convergence. Meanwhile, this explains why we use *HXFlux* to model the heat exchangers. Unlike a typical *HeatX* involved with mass flows (solid lines), *HXFlux* only deals with heat flows (dashed lines as shown in Figure S5).
S3.4. Simulation result of Aspen Plus for GTL

We present a typical simulation result of flowrates and utility consumptions as shown in Table S7 and Table S8. For the inlet flow for FT, the ratio of $\text{H}_2 : \text{CO} = 2.16$ (FTIN) falls in the desired range for the FT reaction. A full Aspen Plus simulation can contain much more results than what has been shown here, but those results might not be relevant in the process analysis and optimization. This is a motivation to develop a reduced-order surrogate.

<table>
<thead>
<tr>
<th>Units</th>
<th>CO2</th>
<th>NG</th>
<th>H2O</th>
<th>AIR</th>
<th>C2-C4</th>
<th>GASOLINE</th>
<th>DIESEL</th>
<th>VENTOUT</th>
<th>H2OOUT</th>
<th>FTIN</th>
<th>FTOUT</th>
<th>RECYCLE</th>
<th>C1RECYCLE</th>
<th>TOFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
<td>0.00</td>
<td>24911.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>H2</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>53701.40</td>
<td>1437.11</td>
<td>1436.92</td>
<td>1313.24</td>
<td>522.06</td>
</tr>
<tr>
<td>CO2</td>
<td>6922.51</td>
<td>101.15</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>2.31</td>
<td>0.00</td>
<td>1773.30</td>
<td>1.02</td>
<td>18056.40</td>
<td>18056.40</td>
<td>17998.20</td>
<td>16449.00</td>
<td>6539.08</td>
</tr>
<tr>
<td>C2H6</td>
<td>0.00</td>
<td>670.31</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>442.89</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>C3H8</td>
<td>0.00</td>
<td>5.14</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>159.16</td>
<td>553.86</td>
<td>0.00</td>
<td>0.00</td>
<td>462.56</td>
<td>25374.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>H2O</td>
<td>0.00</td>
<td>0.00</td>
<td>50639.00</td>
<td>0.00</td>
<td>0.00</td>
<td>19.66</td>
<td>19.59</td>
<td>0.00</td>
<td>358.97</td>
<td>61414.30</td>
<td>97436.90</td>
<td>19911.30</td>
<td>18197.40</td>
<td>7234.14</td>
</tr>
<tr>
<td>C8H18</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>C16H34</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>CH4</td>
<td>0.00</td>
<td>17996.30</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>158.45</td>
<td>158.45</td>
<td>158.45</td>
<td>158.45</td>
<td>158.45</td>
<td>158.45</td>
<td>158.45</td>
<td>158.45</td>
<td></td>
</tr>
<tr>
<td>N2</td>
<td>0.00</td>
<td>13.71</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>O2</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>339.53</td>
<td>0.00</td>
<td>0.00</td>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>6922.51</td>
<td>18786.60</td>
<td>50639.00</td>
<td>0.00</td>
<td>0.00</td>
<td>615.54</td>
<td>582.63</td>
<td>1093.19</td>
<td>3423.27</td>
<td>61414.30</td>
<td>97436.90</td>
<td>47614.50</td>
<td>19911.30</td>
<td>18197.40</td>
</tr>
<tr>
<td>$\bar{M}_W$</td>
<td>44.01</td>
<td>16.71</td>
<td>18.02</td>
<td>28.85</td>
<td>61.44</td>
<td>111.61</td>
<td>226.45</td>
<td>35.25</td>
<td>18.02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Utility ID</th>
<th>Unit</th>
<th>ELE</th>
<th>Steam</th>
<th>U1000</th>
<th>WATER</th>
<th>FT reactor</th>
<th>AIR</th>
<th>Burner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility type</td>
<td>Electricity</td>
<td>Steam</td>
<td>Fuel gas</td>
<td>Cooling water</td>
<td>Cooling water</td>
<td>Cooling air</td>
<td>Waste heat recovery</td>
<td></td>
</tr>
<tr>
<td>Duty</td>
<td>Gcal/h</td>
<td>324.23</td>
<td>46.65</td>
<td>1078.69</td>
<td>560.92</td>
<td>869.40</td>
<td>7.32</td>
<td>16.05</td>
</tr>
</tbody>
</table>
S3.5. An example of one input-output data point for GTL surrogate

According to the simulation result as shown in Table S7 and Table S8, we can collect one input-output data point as shown in Table S9. Obviously, this data point has much less information than an Aspen Plus simulation result, but it has contained the essential features for the GTL process regarding the operating condition and corresponding process output. A sufficient number of such effective data points can be used to train a surrogate for the GTL process.

Table S9. An example of one input-output data point for GTL surrogate.

<table>
<thead>
<tr>
<th>Input variables</th>
<th>Value</th>
<th>Unit</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>F_CO2</td>
<td>6922.51</td>
<td>[kmol/h]</td>
<td>Inlet flowrate of CO_2</td>
</tr>
<tr>
<td>F_CH4</td>
<td>18786.60</td>
<td>[kmol/h]</td>
<td>Inlet flowrate of CH_4</td>
</tr>
<tr>
<td>x_CH4</td>
<td>0.958</td>
<td>[-]</td>
<td>Inlet molar fraction of CH_4</td>
</tr>
<tr>
<td>T_FT</td>
<td>253.24</td>
<td>[°C]</td>
<td>Temperature in FT reactor</td>
</tr>
<tr>
<td>P_FT</td>
<td>19.79</td>
<td>[bar]</td>
<td>Pressure in FT reactor</td>
</tr>
<tr>
<td>N_trays</td>
<td>55</td>
<td>[-]</td>
<td>No. of trays in distillation column</td>
</tr>
<tr>
<td>T_reformer</td>
<td>947.66</td>
<td>[°C]</td>
<td>Temperature in reformer reactor</td>
</tr>
<tr>
<td>P_reformer</td>
<td>4.07</td>
<td>[bar]</td>
<td>Pressure in reformer reactor</td>
</tr>
<tr>
<td>Split_vent</td>
<td>0.086</td>
<td>[-]</td>
<td>Split fraction to vent stream (the other goes to recycle)</td>
</tr>
<tr>
<td>Split_FT</td>
<td>0.398</td>
<td>[-]</td>
<td>Split fraction to FT reactor (the other goes to reformer)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output variables</th>
<th>Value</th>
<th>Unit</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>F_gasoline</td>
<td>713.02</td>
<td>[kmol/h]</td>
<td>Sum of [C_8H_18] in outflows</td>
</tr>
<tr>
<td>F_diesel</td>
<td>1098.81</td>
<td>[kmol/h]</td>
<td>Sum of [C_16H_34] in outflows</td>
</tr>
<tr>
<td>F_gas</td>
<td>437.98</td>
<td>[kmol/h]</td>
<td>Sum of [C_3H_8] in outflows</td>
</tr>
<tr>
<td>F_H2O_net</td>
<td>11172.41</td>
<td>[kmol/h]</td>
<td>Outflow [H_2O] – inflow [H_2O]</td>
</tr>
<tr>
<td>ventCO2</td>
<td>1773.30</td>
<td>[kmol/h]</td>
<td>Flowrate of CO_2 in VENTOUT</td>
</tr>
<tr>
<td>Electricity</td>
<td>324.23</td>
<td>[Gcal/h]</td>
<td>Electricity consumption for pumps and compressors</td>
</tr>
<tr>
<td>U_air</td>
<td>7.32</td>
<td>[Gcal/h]</td>
<td>Cooling by air</td>
</tr>
<tr>
<td>U_1000</td>
<td>1069.06</td>
<td>[Gcal/h]</td>
<td>Heating by 1000 °C fuel gas – waste heat recovery η_{burner}</td>
</tr>
<tr>
<td>U_Steam</td>
<td>46.65</td>
<td>[Gcal/h]</td>
<td>Heating by high-pressure steam</td>
</tr>
<tr>
<td>U_water</td>
<td>1430.32</td>
<td>[Gcal/h]</td>
<td>Cooling by cooling water</td>
</tr>
</tbody>
</table>

References


