UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

END-OF-DEGREE PROJECT

**Degree in Chemical Engineering**

# DEVELOPMENT OF SURROGATE MODELS FOR DISTILLATION TRAINS

## Report and Appendices


**Autor:** Arnau Martínez Mestres
**Director:** Moisès Graells Sobre
**Co-Director**: Ana Somoza Tornos

**Call:** June 2021

# Resum

El temps d'execució necessari per a la resolució de problemes d'optimització en programes de simulació rigorosos no sol ser asequible, fet que promou l'ús de models de substitució. El desenvolupament d'aquests models aproximats comporta la resolució d'una sèrie de reptes com la càrrega computacional i el risc d'excés d'adequació del model. En el treball presentat, les eines i procediments per a crear, entrenar i validar una xarxa neuronal (ANN) son desenvolupats per a l'entrenament de models de simplificació de simulacions rigoroses. Les eines proposades han estat posades a prova en un cas d'estudi que aborda la síntesis de trens de separació per als productes de la pirólisis del polietilé, centrant-se en les columnes de destil·lació del procés simulades en Aspen-HYSYS. Finalment, dos models ANN que simulen el comportament de la columna respecte una funció que considera els costos de la simulació han estat desenvolupats. El comportament i precisió dels dos models és correspon a l'estudiat en la superficie triada.

# Resumen

El tiempo de computación necesario para solucionar problemas de optimización en programas de simulación rigurosos no suele ser asequible, lo que promueve el uso de modelos de sustitución. El desarrollo de estos modelos aproximados conlleva la resolución de una serie de retos como la carga computacional y el riesgo de sobreajuste del modelo. En el presente trabajo, las herramientas y procedimientos para crear, entrenar y validar una red neuronal artificial (ANN), han sido desarrollados para la construcción de modelos simplificados de simulaciones rigurosas. Las herramientas propuestas han sido puestas a prueba en un caso de estudio que aborda la síntesis de trenes de separación para los productos de la pirolisis del polietileno, centrándose en las columnas de destilación del proceso simuladas en Aspen-HYSYS. Finalmente, dos modelos de redes neuronales que simulan el comportamiento de la columna con respecto a una función que considera los costes de la simulación han sido desarrollados. Los dos modelos representan correctamente y con buena precisión la superficie estudiada.

# Abstract

The computational time required to solve optimization problems in rigorous simulation programs is usually unaffordable, raising the need to use surrogate models. The development of these approximate models is a challenge that needs to handle the computational burden and risk of overfitting. In the present work, tools, and procedures to build, train, and validate an Artificial Neural Network (ANN) are developed to build simplified models of rigorous simulations. The proposed tools are tested with a case study that addresses the synthesis of separation trains for the products of polyethylene pyrolysis, focusing on the distillation columns of the process simulated with Aspen-HYSYS. Finally, two ANN models have been developed to simulate the behaviour of the column regarding a function that considers the costs of the simulation. Both models fit correctly and show good accuracies with respect to the surface studied.

# Acknowledgments

First and foremost, I would like to express my deepest gratitude to my project director, Moisès Graells, and project co-director, Ana Somoza, for their guidance and support throughout the entire project and internship at the research group CEPIMA (Centre d'Enginyeria de Processos i Medi Ambient).

I would like to extend my gratitude to all the members from CEPIMA, for their commitment and welcoming attitude towards me when I first join the group as an intern in January. In particular, I would like to express my deepest appreciation to professor Dr. Antonio Espuña, for raising many precious points in our discussion with respect the topics in my project. Moreover, I am grateful for the help and interest of Gerard Campanya, who genuinely gave me advice and knowledge with respect some of the topics developed on the project.

I would also like to thank all the professors that have guided me through my bachelor studies. Specially, professor Dr. Antonio Espuña and professor Dr. Gerard Escudero, for accepting me as a listener in their respective classes during the development of this end-of-degree thesis. I hope that I have manage to address several of the topics discussed in their classes here.

Furthermore, I would like to extend my deepest thanks to my family and friends because this project would not have been possible without their endless patience, love and helpfulness. Their support during the entire studies has given me strength in the most challenging moments.

# Glossary

ANN – Artificial Neural Network

SVM – Support Vector Machines

NaN – Not a Number (unfeasible designs)

NT – Number of Trays

RR – Reflux ratio

$D_{Flow}$ – Distillate Flow/Rate

HK – Heavy Key component

LK – Light Key component

Feed – Inlet matter stream in the distillation column

TOP – Outlet matter stream in the distillation column with the lightest components

BOTTOM – Outlet matter stream in the distillation column with the heaviest components

# Index

# 1. Preface

## 1.1. Project origin

The project continues the work presented in Somoza-Tornos et al. (2020), in which a modelling approach for the joint synthesis of production processes and products from a waste-to-resource perspective is proposed. From a general Process Systems Engineering perspective, this project develops methods and tools to substitute rigorous simulation models for distillation processes by more computationally efficient data-based surrogate models, which could be later used in the simulation and optimization of more complex process systems such as distillation trains and multicomponent separations. Hence, a set of tools will be developed and tested with the same case study presented by Somoza-Tornos et al. (2020), the polyethylene pyrolysis.

## 1.2. Motivation and scope of the project

The use of surrogate models has been receiving increasing attention in the last years (Bhosekar & Ierapetritou, 2018). In particular, its applications in the process systems engineering field have raise in popularity (Swain et al., 1992).

These methods can be applied to the case study reported by Somoza-Tornos et al. (2020): the modelling of valorisation processes upcycling waste from different sources is addressed in the framework of circular economy, specifically in the separation processes required to process the mixtures resulting from the pyrolysis of waste plastic. The circular economy paradigm requires process synthesis to be expanded beyond the consideration of production activities aimed at market needs and to integrate valorisation processes upcycling waste from different sources (industrial and urban). Multicomponent separation processes are generally expensive and potentially hazardous for the environment. Secondary products from these processes are reused as fuels (i.e. waste-to-energy, Honus et al., 2016) or recovered as resources (Hong & Chen, 2017). Hence, decisions are made on the separation and reuse of these products (material reuse vs. energy valorisation).

A recent approach has been done by modelling a system through a superstructure with features from state-task network (i.e. the activation/deactivation of units) and state-equipment network (i.e. multiple tasks in a unit) representations (Somoza-Tornos et al., 2020). The computational time required to run these kind of optimizations is usually unaffordable. Hence, a methodology based on surrogate models is proposed to simulate the process for a later optimization of the superstructure.

# 2. Introduction

The two main alternatives for waste polyethylene (WPE) are its use as fuel (i.e. waste-to-energy) and its recovery as resources (i.e. monomers that can be used to produce new polymers), which it is more in tune with the idea of closing material loops. Each unit and task of the chemical upcycling of WPE has to be studied separately, as the global optimization of the process will decide which units and tasks are activated or deactivated (Somoza-Tornos et al., 2020). The purpose of this work is the development of a method based on artificial neural networks to obtain a surrogate model capable to describe the behaviour of the individual distillation columns so that they can be later used in the solution of the optimization of the separation train recovering the monomers obtained from the polyethylene pyrolysis. These surrogate models will efficiently mimic the results of the rigorous models obtained with Aspen HYSYS.

The approach hereby adopted not only includes the creation and training of the model, but also the sampling of the data and its treatments to fit surrogate's requirements. Deep learning concepts are applied in conjunction with basic distillation knowledge to avoid models with over fitting and under fitting.

## 2.1. Objectives of the project

The main goals of the project are listed below.

- To develop a methodology based on data-base surrogate models to substitute rigorous distillation column models.
    - To characterize the input and output variables of the model.
    - To substitute rigorous simulation models by computationally more efficient surrogate models.
- To design the necessary tools to implement the proposed methodology in an efficient and straightforward way.
    - To connect the simulation environment to an external source.
    - To generalize the tools so they can be used in several distillation column scenarios and users.
- To apply the methodology to a case study and validate the performance of the models.
    - To previously study the process in other simulators.
    - To compare the simulation model with different configurations in terms of variable selection.

# 3. Introduction to machine learning tools. Learning theory

Machine learning tools aim to find a function or model that predicts a target with the lowest possible error. Furthermore, accurate predictions on unseen inputs must be also sought. Hence, regularization is implemented to avoid models having both underfitting or overfitting. A visual comparison of both types of models is shown in Figure 3.1.

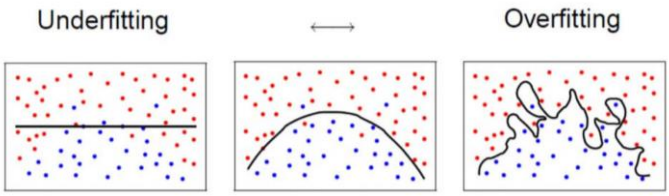


*Figure 3.1. Difference between over fitted and under fitted models (Robertshaw, 2015)*

To understand when each situation occurs a mathematical understanding must be given. The mean square error (MSE) is defined in Equation (1).

$$MSE = \left(E\left[\hat{f}(x)\right] - f(x)\right)^2 + E\left[\hat{f}(x) - E\left[\hat{f}(x)\right]\right]^2 + \sigma_e^2$$
$$MSE = Bias^2 + Variance + Irreductible\ Error \quad (1)$$

The more complex the model the more data points it will capture, and the lower the bias will be. However, complexity will make the model "move" more to capture the data points, and hence its variance will be larger.

Figure 3.2 presents typical predictions of four models with low and high variances and bias respectively.

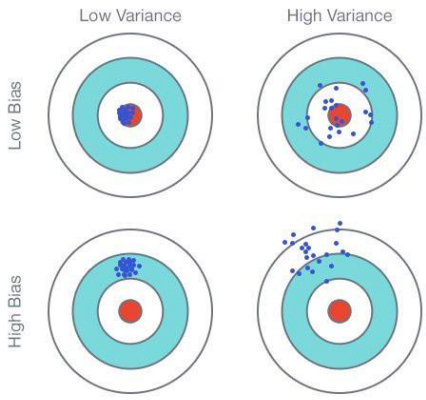


*Figure 3.2. Predictions from models with high and low variances and bias (Medium, 2018)*

Consequently, decreasing the error requires increasing the complexity of the model. However, too much complexity may lead to unpredictability. Although a grade ten polynomial may fit some data perfectly, when more points are added, the model may need to vary a lot to fit them. High complex models show overfitting. Therefore, predictions made with these models will show considerable variances. To avoid overfitting, the following advices are commonly given:

- ✓ Reduce model complexity. All machine learning tools have parameters that can be changed to modify the accuracy of the model. For instance, in artificial neural networks the number of layers and neurons should be decreased if the model shows overfitting.
- ✓ Get more training data, so there are more possible correlations between the data. If more samples cannot be obtained techniques such as data augmentation can be used.
- ✓ Train less, so no redundant information is taken into consideration.

In the other hand, if too low complexity is given to the model, it will not have enough information to accurately represent the reality. Consequently, new points will differ from the predictions. This “bias” in the predictions is common in low complex models. To avoid underfitting, the following advices are commonly given:

- ✓ Increase model complexity.
- ✓ Train longer, so more information is taken from each sample.

There are some rules of thumb to follow when using machine learning tools.

- ✓ Occam’s razor in learning: simpler models are likely to be correct.
- ✓ No free lunch theorem: there is no method which outperforms all others for all data sets.
- ✓ Curse of dimensionality: when the dimensionality increases the amount of data needed to support the result often grows exponentially.

These heuristics will be taken into consideration when designing the surrogate. Reducing the dimensions as well as simplifying the models will be crucial to develop reliable models.

# 4. Overview of surrogate models

Surrogate models are engineering methods used when an output of interest cannot be directly measured, so a model of the outcome is used instead. When designing equipment or optimizing its operation conditions it is often the case that the symbolic form of the function that correlates the inputs with the outputs is unknown or expensive to get (Bhosekar & Ierapetritou, 2018). In this situations, surrogate models offer a computationally efficient alternative to find the objective function.

The development of surrogate models for simulation and optimization has boost parallel to the availability of Machine Learning tools to produce data-based models.

In the past decade's surrogate's popularity has rapidly increase, as well as the different types of surrogates (Xie et al., 2018). Some of the most popular ones are *ANN*, used for process modelling, process control and optimization; *Kriging*, used for process flowsheet simulations, design simulations and feasibility analysis; R*adial basis functions*, used for parameter estimation and feasibility analysis; *Support Vector Machines*, used for classification and feasibility analysis, and *Support Vector Regression,* used for parameter estimation.

It is remarkable to note that with respect the applications, the problems requiring surrogates can be classified in to three classes. The first class of problems is the most fundamental use of surrogates i.e. prediction and modelling. The second class of problems is commonly known as derivative-free optimization (DFO) where the objective function to be optimized is expensive and thus derivative information is unavailable. The third class of problems is feasibility analysis where the objective is also to satisfy design constraints (Bhosekar & Ierapetritou, 2018).

The most popular methods in the literature are next described.

## 4.1. Artificial Neural Networks

### 4.1.1. Elements and parameters to adjust

The basic unit of a neural network is the processing unit, the neuron. The network consists of a highly interconnected numbers of neurons arranged in two or more layers. The structure of layers is divided in two classes: hidden layers and the output layer, which is the last one. The inputs will be given to the network and will cross through the trained hidden layers until they reach the output layer, where data will be displayed.

Each neuron applies the following transformation to the input to get the output. The vector of inputs is multiplied by the weight of the neuron, and the bias is summed up. This value is given to the

activation function, which will apply the final transformation to generate the new vector of inputs for the next layer of neurons.

An adaptation of the process is visually described in Figure 4.1 (Ibrahim et al., 2018).

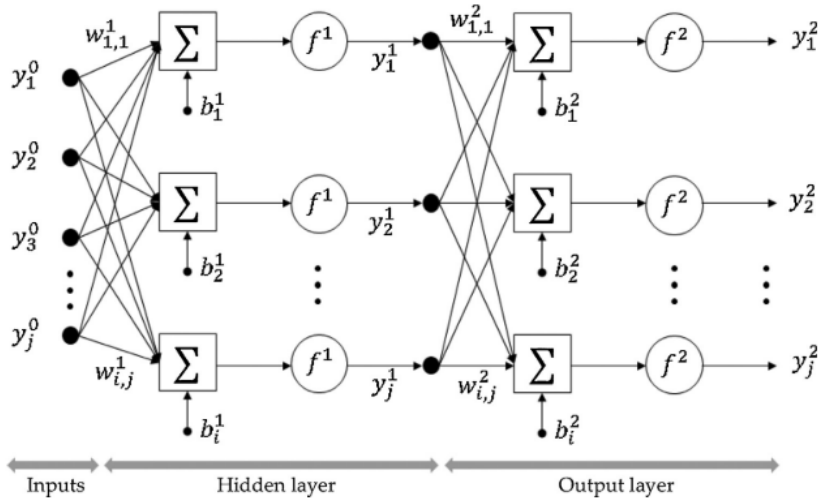


*Figure 4.1 Schematic representation of a multi-layer feed forward neural neuron* (Ibrahim et al., 2018)

Mathematically, this multi-layer feedforward network can be formulated as described in Equation (2).

$$y = f^2(W^2 f^1(W^1 y^0 + b^1) + b^2) \quad (2)$$

Where **W** is the matrix of weights, **b** the vector of biases, **f** the transfer functions, and **y** the inputs/output variables. The exponents represent the two layers that define the multilayer.

The hidden layer, which usually consists on one sub-layer, is where all transformations are made. The number of neurons of each layer can be changed so a better performance is achieved. There are methods to optimize the number of layers and neurons (Dua, 2010), but these are claimed to be time consuming and challenging. Alternatively, the number of layers and neurons can be settled by applying trial and error (Schäfer et al., 2019).

The output layer returns the output variables. The number of neurons that contains depends on the output variables associated to the ANN.

These layers are trained to input-output data points using training algorithms, being "back-propagation" the most used. The two steps in backpropagation are:

1. Computing the prediction error, mean square error (MSE), using fixed values of **W** and **b**.
2. Adjusting **W** and **b** so the prediction error is minimized. The MSE is defined be Equation (3).

$$MSE = \sum_{i} \frac{(t_i - y_i)^2}{N} \tag{3}$$

Where **t** and **y** are the target output and predicted output and N the total number of sample points.

Information passes through layers using an activation function, which modifies the output value of a layer before it moves to the next one. There are several types of activation functions, each of them being recommended for specific tasks or networks (Brownlee, 2021). In multilayer perceptron networks, the Rectified Linear Unit, or ReLU, activation functions are typically used for the hidden layers. The activation function of the output layer will vary regarding the type of problem. For regression problems, linear activations functions are recommended. Whereas in classification problems "softmax" or sigmoid functions are more common.

The accuracy of the model is checked with the coefficient of determination, a dimensionless number that indicates the fraction of the variability of the dependent variable explained by the surrogate model. In order to avoid overfitting or underfitting data must be split in three sets: training set, validation set, and testing set; which are described below:

- Training set (70%): these points are used to construct the model.
- Validation set (15%): these points are used while training to avoid overfitting.
- Testing set (15%): these points are used to check the performance of the model.

### 4.1.2. Types of networks

The most common types of networks are described below. A most detailed description on different neural networks can be found at the bibliography (Leijnen, 2016).

Perceptron: a neural network that is made out of just one neuron.

Feed forward (FF): networks with multiples neurons present as well as one hidden layer. If radial basis is used as activation function is consequently named radial basis function (RBF). Finally, when multiple hidden layers are considered it is a deep feed forward network (DFFN). This last type of ANN's is the most common when facing regression problems.

Recurrent Neural Network (RNN): when networks are feed not only with information from past neurons, but also with information of themselves. In general, recurrent networks are a good choice for advancing or completing information (Elman, 1990).

Convolutional Neural Networks (CNN): are used to identify images and sound, as they focus on learning patterns. Part of the information is given to train some of the neurons, while a second sampling set is directly given to the convolutional neurons (Lecun et al., 1998).

## 4.2. Kernel methods. Support Vector Machines & Regressions

A brief description on Kernel methods is given in the present section. A deeper mathematical background can be found in the bibliography (Hofmann et al., 2008). Traditionally, machine learning algorithms have been focused on the linear case. Real world data analysis problems, on the other hand, often require nonlinear methods to detect the kind of dependencies that allow successful prediction of properties of interest. Kernels methods make us of higher dimensional spaces to hopefully separate or structure the data easily. By using dot products between two vectors, data can be map into a higher dimensional space, where linear algorithm operating in this space will behave non-linearly in the original space.

Kernel methods just replace an existing inner product with the inner product from some other suitable space. Hence, choosing the right kernel as well as tuning its parameters is a crucial task that will depend on the data of the problem. Automatic kernel selection is discussed in the bibliography (Howley & Madden, 2006). The most common kernel types are: linear, polynomial, Gaussian, exponential, circular, spherical, spline, and Bayesian.

***Support vector machines***

Support vector machines are a set of related supervised learning methods used for classification or regression. By creating a hyperplane that maximizes the distance between the sets of points the SVM can predict whether a new example falls into one category or another. The variety of Kernels allow super vector machines to easily separate different sets of data (Souza, 2010). Figure 4.2 shows how this tool splits the data regarding its category: red or green.

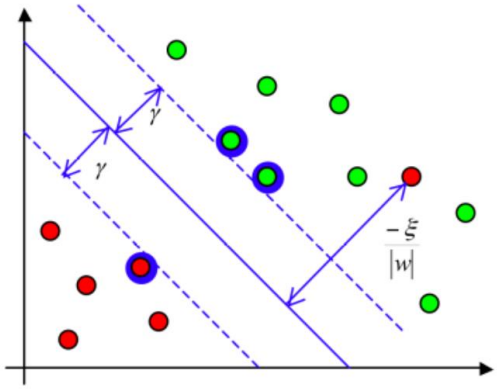


*Figure 4.2. Representation of the hyperplane generated by a SVM*

As well as in neural networks, SVM's must be validate during the training phase and then tested. The accuracy is checked with the error (ξ), as not all the samples will be correctly classified.

Support Vector Machine can also be used as a regression method, maintaining all the main features that characterize the algorithm (maximal margin). However, as data is a real number it is more challenging to predict its value, as it has infinite possibilities. The Support Vector Regression (SVR) uses the same principles as the SVM for classification, with only a few minor differences. Kernels can be also used to perform other regressions than linear, as it is shown in Figure 4.3.

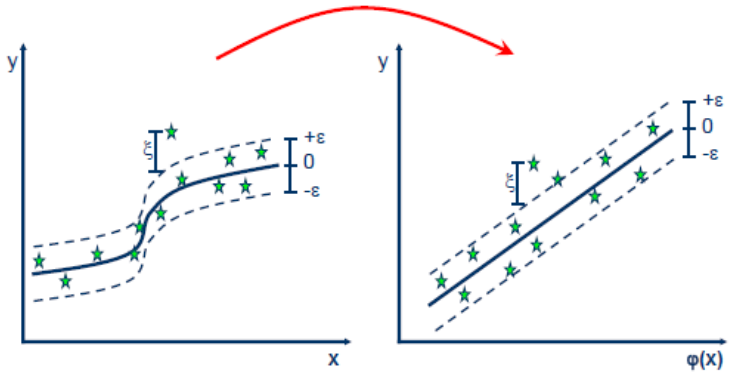


*Figure 4.3. Support Vector Regression representation with non-linear Kernels (Sayad, 2010)*

# 5. Separation processes. Distillation

## 5.2. Overview in distillation

Distillation is a separation process based on the difference in volatilities of the components. The liquid mixture at its boiling point will produce a vapour phase richer on the most volatile components, which are usually named light components. The vapour phase will be later condensed and separated a part. This condensed stream is usually called the distillate, but it can be also referred as TOP stream of the column. The less volatile components will not evaporate and will exit the distillation unit in a stream usually named BOTTOM.

As in all kinds of separations, no mass is transformed, just distribute in a more uniformed way. However, the Second Law of Thermodynamics claims that the total entropy of the universe must remain constant or increase in a process. A purer stream has a lower level of entropy than a uniform mixture of components, as there are more possible microstates that configure a uniform mixture than a pure one. Consequently, in order to evolve to a purer mixture, another component must degrade at least as much as it. This component is called the separation agent, and varies in each kind of separation process. In distillation, the separation agent is the energy that must be given to the reboiler in order to enrich the vapour phase with the light components. The energy will be degraded as a consequence of exiting at a lower temperature. Heat integration systems are used to reduce the degradation of the energy, reusing the energy produced in the condenser unit thanks to the TOP stream to heat the reboiler.

***Volatility***

One of the parameters that influences the amount of heat required is the difficulty in separating the components. The relative volatility between components can be used to quantitatively determine the difficultness. The relative volatility is defined as the ratio of absolute volatilities of two components, which is described by Equation (4).

$$\alpha_i = \frac{P_i}{x_i} = \frac{y_i}{x_i} P \tag{4}$$

In the previous definition Dalton's Law has been applied to correlate the volatility to the molar fraction of a component in both phases. The simplification will then facilitate the definition of the relative volatility, shown in Equation (5):

$$\alpha_{ij} = \frac{y_i / x_i}{y_j / x_j} = \frac{k_i}{k_j} \tag{5}$$

Where "$k_i$" is defined as the equilibrium ratio of a component. This is the value that simulation software's like Aspen HYSYS use to define the volatilities, as it just depends on liquid and vapour molar fractions of a component at a specific temperature and pressure.

Both absolute and relative volatility directly depend on the pressure. Further details on how a distillation process can be affected by the pressure will be given in Section 5.3.

***Equilibrium and degrees of freedom***

Not only mass and energy conservation must be kept, but also the equilibrium laws. In the equilibrium, the temperature (T), the pressure (T), and the fractions in the liquid and vapour phase ($x_i$ and $y_i$) must remain constant. The equilibrium law is described in Equation (6).

$$f(x_i, y_i, T, P) = 0 \tag{6}$$

The number of independent variables that must be defined is determined by Gibbs Phase Rule, defined in Equation (7).

$$F + L = C + 2 \tag{7}$$

Where F is the number of phases, L the degrees of freedom, and C the number of components. For binary mixtures the resultant L is 2.

***Distillation column variables***

A basic representation of a distillation columns is shown in Figure 5.1. When performing an optimization of the column some key variables have to be taken into consideration. A short definition of these variables is given in the next paragraph.

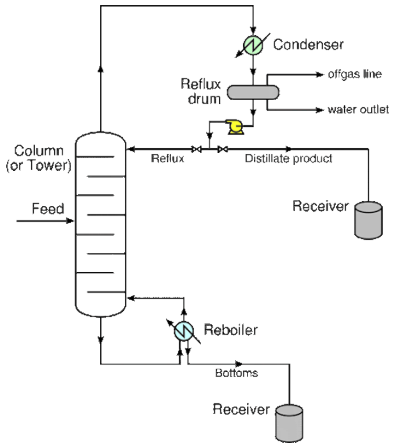


*Figure 5.1 Distillation column picture with its basic streams and elements (Wermac, 2021)*

The number of plates (NN) that conforms the column defines how many equilibrium stages will achieve the feed stream before exiting the tower from top or bottom. The feed tray represents at which tray of the column the initial mixture is fed. A first estimation of this tray can be done by considering the composition of the Feed. For instance, if a 0.9 purity at TOP and a 0.1 purity at BOTTOM is pursued from a 0.5 purity mixture, the Feed will be located at half of the tower, to procure a smooth concentration profile in the tower. Each tray has a cost, so the bigger the tower the more expensive it is. Consequently, the objective function will take into consideration the costs of the trays. Therefore, the number of trays will be a decision variable when optimizing the design of columns.

The reflux ratio (RR) is defined as the ratio between the stream that is recycled to the tower after condensing at top ($L_D$) and the exiting stream (TOP or distillate). The purpose of recycling part of the distillate to the tower is to improve the separation that is done in each tray. The boiling Feed will face a richer stream in volatile components ($L_D$) in each tray. Consequently, the vapour stream will become richer in these volatile components due to equilibrium laws. For this reason, the higher the reflux ratio the less trays will be required to achieve a certain purity. However, increasing the reflux ratio has its own costs associated. The bigger the recycle the less product it is obtained. If the same amount of distillate wants to be produced a bigger amount of energy at the condenser unit will be required. Hence, the reflux ratio will be a decision variable when optimizing both the design and the operation of a column.

From these two variables the others can be defined. In a distillation column there are in total five variables: number of trays, feed tray, condenser duty, reboiler duty, and reflux ratio. Giving a Feed stream, by defining these five variables the outlet streams can be calculated. However, some simplifications can be done the reduce the number of variables to specify.

From the mass and energy balances we obtain that the feed stream is split in TOP and BOTTOM. Hence, in order to obtain saturated liquid, the condenser and reboiler duties must be similar as the enthalpy of both streams will not differ that much. Columns usually work with saturated liquids because subcooled ones increase the amount of liquid recycled into the tower. As a consequence, the purity of the components increase, as well as the duties. If the optimum RR is one in particular, working with subcooled streams means not working at optimal conditions. In this particular case, it would mean produce a purer product than what it is required. Astonishingly, having heat losses in the column will result in the same situation, as losses produce a purer reflux.

As it was explained before, the feed tray can be estimated by considering the concentration profiles over the column and the feed composition. From these simplifications it is conclude that distillation columns have three degrees of freedom. Further discussion regarding simplifications will be done in Section 5.4.

***Definition of optimization problems***

Depending on what wants to be optimized the decision variables and the objective function will vary. When designing a process different optimizations will be carried out with regard the stage of the process. First of all, columns must be designed to fulfil specific demands. Once the columns are bought, more accurate operating conditions must be found to obtain the best possible benefit. Finally, more regular optimizations will be done when unforeseen situations happen. These three types of optimization problems are described in more detail in the following paragraphs.

There are three variables which are linked. By knowing the reflux ratio and the number of trays, the outlet conditions can be calculated from a defined Feed stream. Alternatively, the design of the column (NN) can be calculated from the outlet conditions and a reflux ratio.

When performing design optimizations strategic decisions are applied. The scope of the optimization is years, as the optimum designs are going to be used until the project ends. Consequently, design optimizations are done once in a while. When performing this kind of optimizations, the outlet composition is usually specified. Therefore, the decision variables are the reflux ratio and the number of trays. The optimization will lead to a proper design to minimize or maximize an objective function.

In contrast, operation optimization scope is shorter, usually weeks or months. In order to make this kind of optimization a clear design must be known. Consequently, the number of trays will be specified, while the outlet conditions and the reflux ratio will be the decision variables.

Finally, control optimizations represent the decisions that must be taken in a short interval of time, usually minutes or hours. The framework for this kind of optimization problems are unforeseen events, which cause the equipment work at different conditions from the ones specified by the operation optimization. Dynamic systems are out of the scope in the present research, as many of the equations used to develop the research assume steady state conditions.

## 5.3. Multicomponent separation

In multicomponent separation the different components usually are required to be separated individually, in order to obtain n-streams of high purity from a n-mixture of components. Considering that each column ends in a distillate stream and a bottom stream, a total of n-1 columns will be required to separate the n-mixture. However, multiple scenarios show up in multicomponent distillations. For instance, a mixture of four components (A-D) can be distillate in five possible ways:

1. $A - BCD \rightarrow B - CD \rightarrow C - D$
2. $A - BCD \rightarrow BC - D \rightarrow B - C$
3. $AB - CD \rightarrow A - B \rightarrow C - D$

4. $ABC - D \rightarrow A - BC \rightarrow B - C$
5. $ABC - D \rightarrow AB - C \rightarrow A - B$

When the number of components is five the possible number of sequences increases to 14, and so on.

Therefore, some criteria must be chosen to select which sequence will be the best one. On the following paragraphs different approaches are going to be explain. All of them use simplifications, so there is no absolute answer. But knowing these methods will lead to more accurate and reasonable-based decisions.

***Heuristics***

Heuristics regarding separation processes can be defined to make a fast first analysis about the characteristics of the mixture that needs to be distillate. These Heuristics should take into consideration the decision variables that were previously described, as well as other factors that affect the objective function. Four heuristics are proposed as basis to analyse a mixture given.

Heuristic 1. Split components of low relative volatility at the end of the sequence. The lower the relative volatility the more complicated the separation it is. Therefore, higher reflux ratios will be required, increasing the duties. Consequently, if the separation of these components is done at the beginning, where no other components have been already split, the costs will be even higher, as big streams with high reflux ratios demand huge amounts of energy to condense.

Heuristic 2. Give preference to the direct sequence, which means splitting the components from the most volatile to the less one. This heuristic emphasizes the economic value of the lighter components, which are usually sold at higher prizes. As distillation columns are not perfect, the later the lighter components are split the higher the chances of losing them as secondary products of the previous distillation units. For this reason, separating them the first should be highly considered when the prize at which the products are sold is not despicable.

Heuristic 3. Splitting the most abundant component the first. Consequently, later columns will have to deal with lower quantities of product, decreasing the duties required.

Heuristic 4. Equimolar separation of the components at TOP and BOTTOM. As in Heuristic 3, by splitting the initial stream in two halves', the next columns will not have to deal with unbalanced amounts of Feed.

Not always will these heuristics be in tune, so different weights must be assigned to the different heuristics regarding the situation. For instance, if he relative volatility of all the components is quite similar, the first heuristic will have less relevance as all the distillations will be similarly challenging.

### *Underwood's simplified equation*

A second approach to find the optimal sequence is simplifying Underwood's equation, which is defined in Equation (8). The first simplification is approximating the value of Φ to the average of the heavy key and light key component (HK & LK). These terms refer to the main components that are split on the tower. All the components lighter than the light key (LLK) are completely separated on TOP, while the heavier than the heavy key (HHK) are split it on the BOTTOM's. As distillation columns are not perfect, there will always be presence of the LK component at BOTTOM as well as presence of the HK at TOP.

$$1 - q = \sum_{N_C} \frac{\alpha_i \cdot x_{F,i}}{\alpha_i - \Phi} = \sum_{N_C} \frac{\alpha_i \cdot x_{F,i}}{\alpha_i - \frac{\alpha_{LK} + \alpha_{HK}}{2}} \tag{8}$$

Underwood's equation can be rewritten to find the minimum reflux ratio ($r_{min}$), which represents the lowest amount of reflux that must be given to the tower to make the desired separation. Key performance indicators propose that the reflux ratio is between 1.2-1.5 times the minimum reflux ratio. The KPI takes into consideration the different costs of the distillation process, so the interval must be taken just as a reference, as costs can vary from different countries or bibliography. From the $r_{min}$ the minimum amount of vapours required can be calculated, which will directly affect the utility costs of the column. The derived expression is presented in Equation (9).

$$V_{min} = D(r_{min} + 1) = D \sum_{i} \frac{\alpha_i \cdot x_{D,i}}{\alpha_i - \Phi} = V_{min}^{bin} + \sum_{LLK} \frac{\alpha_i}{\alpha_i - \frac{\alpha_{LK} + \alpha_{HK}}{2}} \cdot F \cdot x_{F,i} \tag{9}$$

The $V_{min}$ can be then split it in two parts: the minimum vapours required to separate the key components, and the vapours required to separate the other components ones. When designing a particular column both values should be taken into consideration. However, as this method is going to be used to choose the best possible sequence, all the columns are considered. Consequently, no matter which sequence is chosen, the $V_{min}^{bin}$ of all the components will appear, as A will be at some point split it from B, as well as B will be separated from C and so on. To conclude, if this method is used to compare sequences, the only values that must be calculated are the vapours associated to the non-key components ($V_{extra}$). Finally, from a specific Feed stream all the $V_{extra}$ can be calculated, showing which sequence requires less amount of vapours. As it was clarified before, all these methods tackle the problem from different perspectives, and all of them lack part of the information. For instance, by using Underwood's simplified method only the amount of vapours required is being considerate to choose the optimal sequence. The accuracy of the method can be checked later using simulators such as Aspen HYSYS.

***Effect of the pressure***

Changes in pressure can lead to different optimal sequences, as varying the boiling points change the volatility of the components. Decreasing the pressure will decrease the boiling point of the substances, therefore increasing their volatilities. Consequently, the relative volatility between components increase, intensifying the effect of it. Therefore, those sequences that prioritize first the separation of the most volatile components will be promoted. By increasing the pressure, the different separations become equally challenging, meaning that there is no point in considering Heuristic 1.

## 5.4. Shortcut and rigorous distillation models

Two common types of distillation units that can be found in process simulation software's are typically named "shortcut" and the "column". The two of them perform differently and are used for different purposes, as they are not defined with the same equations.

### 5.4.1. Shortcut

Shortcut methods aim to reduce the number of equations to describe the behaviour of the process, thus requiring minor calculus time and memory. Although these methods are not as accurate as rigorous columns which can consider the dynamics of the complete column, they can be used to design columns and obtain the limiting conditions as minimum reflux ratio or minimum number of stages. Most commercially available process simulators offer a Shortcut model based on Fenske-Underwood-Gilliland equations, or FUG, equations. Underwood's equation has already been presented, but not the simplifications or hypothesis that are made to get it. FUG's framework assumes the following conditions (Canyon Hydro et al., 2013).

- Constant molar flow along the column.
- Constant relative volatilities throughout the process.
- Negligible fluid and vapour accumulation within the column.

The first hypothesis is that the enthalpy of vaporization is the same for all components, valid simplification when separating similar components. Considering constant relative volatilities simplifies the calculus, as liquid-vapour equilibrium no longer applies. However, this assumption reduces the accuracy of the model when equilibrium is crucial. Finally, the fluid and vapour's accumulation within the column can be neglected as it is less than the liquid accumulated in the reboiler.

In the following paragraph's FUG's method is explained in detail.

1. Use Fenke's equation to calculate the minimum number of trays ($N_{min}$). From a known Feed stream and outlet conditions (HK & LK), the molar fractions at TOP and BOTTOM can be

determined ($x_D$ & $x_B$). Relative volatilities between the components can be calculated with Equation (10). By knowing these values, Equation (11) can be used to determine $N_{min}$.

$$\alpha_{i,j} = \frac{K_i}{K_j} \tag{10}$$

$$N_{\min} + 1 = \frac{\ln\left[\left(\frac{x_{D_{LK}}}{x_{D_{HK}}}\right)\left(\frac{x_{B_{LK}}}{x_{B_{HK}}}\right)\right]}{\ln(\alpha_{LK,HK})} \tag{11}$$

2. Solve Underwood's equation to find the minimum reflux ratio ($r_{min}$). A n-grade equation must be solved, where n is the number of components to separate. By knowing the feed composition ($z_i$) and state (q), and the volatilities of the pure components the parameter Φ can be calculated from Equation (12). Then, Φ can be introduced in Equation (13) to calculate $r_{min}$.

$$\sum_{i}^{NC} \frac{\alpha_i \cdot z_i}{\alpha_i - \Phi} = 1 - q \tag{12}$$

$$\sum_{i}^{NC} \frac{\alpha_i \cdot x_{D,i}}{\alpha_i - \Phi} = 1 + r_{min} \tag{13}$$

3. Once the reflux ratio is defined from the minimum reflux ratio Gilliland's correlation can be applied to find the actual number of tray. Gilliland's correlation is shown in Equation (*14*).

$$\frac{N - N_{min}}{N + 1} = 0.75\left[1 - \left(\frac{R - R_{min}}{R + 1}\right)^{0.5688}\right] \tag{14}$$

4. Finally, if the feed tray position wants to be determined more accurately, Kirkbride's correlation can be used. The required variables are: the compositions at the outlet and inlet conditions ($x_i$ & $z_i$) and the TOP and BOTTOM streams (D & B). Kirkbride's correlation is shown in Equation (*15*).

$$\frac{N_R}{N_S} = \left[\left(\frac{z_{HK}}{z_{LK}}\right)\left(\frac{x_{B,LK}}{x_{D,HK}}\right)^2\left(\frac{B}{D}\right)\right]^{0.206} \tag{15}$$

To sum up, by giving a Feed stream and the outlet composition, FUG's method determines $r_{min}$ and $N_{min}$. If the actual reflux ratio is approximated, for instance from KPI's, the actual number of trays and feed tray are also calculated. However, this is just an approximation. If more accurate designs are

required, a proper optimization of the shortcut should be done to fine the best reflux ratio, from which then calculate the actual number of trays.

The shortcut has in total five degrees of freedom. The outlet conditions can be summarized with the HK composition at the distillate and the LK composition at bottom's. The pressure at the reboiler and condenser is required so the volatilities can be calculated. Finally, the correlation between the $r_{min}$ and the reflux ratio must be given to predict the actual design conditions. Consequently, shortcuts are commonly used to firstly approximate column designs. Then, more rigorous column designs can be simulated based on this approximation so operation problems can be solved.

### 5.4.2. Rigorous distillation models

In contrast to shortcuts, most commercially available process simulators rigorous distillation models take into consideration the liquid-vapour equilibrium equations, which are applied at each tray to find the find outlet vapours and liquids. Consequently, variables such as pressure drops, reboiling ratios or distillate flow rates can be specified. The complexity of the model is higher than in the shortcut. Consequently, convergence errors may occur when simulating with columns.

When implementing a Column model in HYSYS, the following variables must be specified: number of trays, feed tray position, type of column, and condenser and reboiler pressures. Therefore, there are two degrees of freedom to converge the column. For operation problems, variables such as reflux ratio and distillate flowrate are usually chosen. Other feasible options are reboiling ratios or fractions at outlet conditions, for example.

The main upside of simulating with columns is the accuracy, as liquid-vapour equilibrium is considerate at each tray. On the other, column models required the number of trays to be fixed, not being proper models when designing columns then. Furthermore, convergence errors may lead to unfeasible designs.

On the other side, shortcuts avoid convergence errors by reducing the number of equations implemented. Additionally, they can be used to design columns as the number of trays can be estimated with FUG's method. However, they lack accuracy and flexibility.

# 6. State of the art. Surrogates in distillation problems

In this section two of the most cited papers on the field are going to be individually studied. Both tackle optimization problems regarding distillation columns, though they use different surrogates in different situations. The main points I am going to focus are:

- For which purpose are the surrogate's used.
- The selection of the surrogate.
- Type of optimization problem presented.
- Variables taken into consideration to train the model.
- Other tasks done apart from training the model.

## 6.1. Rigorous Design of Distillation Columns Using Surrogate Models Based on Kriging Interpolation (Quirante et al., 2015)

In the paper (Quirante et al., 2015), Kriging models are used to solve design optimization problems based of five different examples, that represent different situations where distillation units are required. The surrogates are used to mimic the behaviour of the column, and can be explicitly or implicitly introduced on the optimizer. While the surrogate is implemented on MatLab, the optimization tool is the GAMS-BARON modelling system.

In the introduction of the paper a brief explanation of the state of the art is presented. The reasons why surrogates are becoming more popular when approaching distillation problems are presented, as well as a classification of the different types of surrogates.

Then, Kriging models are presented so the mathematical background can be understood. A methodology to tackle global optimization problems based on data augmentation is described when there is a limit of sampling points. The main steps to follow are:

1. Sampling: the sampled points must be separated enough to ensure that the noise generated by the simulation does not significantly affect the Kriging model.
2. Train the model: Kriging does not incorporate cross-correlation between the different simulation outputs, univariate Kriging models are fitted.
3. Test the model using cross-validation for a set of test simulations, validate the accuracy of the model. If the error is small enough the Kriging model can be used to substitute the actual one without further considerations. However, in general this is not the case.

4. Substitute the actual model by the Kriging surrogate and perform the (MI)NLP optimization. If the Kriging model is considered a good approximation of the actual model, then finish. Otherwise continue with the next point.
5. Add the optimal point obtained in the previous step to the set of sampled points. Update Kriging and re-optimize. If in two consecutive iterations, there is no improvement then test for optimality by going to the next step.
6. Contract the feasible search region within a trust region around the incumbent solution and repeat steps 4 to 6. In a completely free of error simulation system, this approach can be repeated until we can guarantee that the error in the gradient is below a given tolerance and test the Karush– Kuhn–Tucker optimality conditions. In a noisy system it is not possible to follow this approach, and we must finish with the non-improvement criterion in a small but large enough region.

Finally, after defining the objective function that will be minimized, different examples of distillation columns are presented. In Table 1 the main objectives and variables used to train the Kriging models are presented for all the examples. The nomenclature used can be found at the paper. The main dependent variables that are taken into consideration are the outlet compositions, dimensions of the tower and duties. The most relevant comments regarding the examples presented are summarized in the next paragraphs.

*Table 1. Main objectives and dependent and independent variables*

| | Objective | Dependent Variables | Independent variables |
|---|---|---|---|
| **Example 1.a** Fixed trays | Operative conditions (RR and BR) | $[x_{benzene}^{TOP}, D_{benzene}^{TOP}, Q_{reb}, Q_{Cond}, D]$ | $Kr(RR, BR)$ |
| **Example 1.b** Variable trays | Total number of trays and feed location | $[x_{benzene}^{TOP}, D_{benzene}^{TOP}, Q_{reb}, Q_{Cond}, D]$ | $Kr(RR, BR, N_S, N_R)$ |
| **Example 2** Divided wall column | Total number of trays and feed location | $[D_1]$<br>$[x_i^{TOP,C2}, F_i^{TOP,C2}, x_i^{BOT,C2}, F_i^{BOT,C2}, Q_{cond}, D_2]$<br>$[x_i^{TOP,C3}, F_i^{TOP,C3}, x_i^{BOT,C3}, F_i^{BOT,C3}, Q_{reb}, D_3]$ | $Kr_1(N_{S1}, N_{R1})$<br>$Kr_2(N_{S2}, N_{R2})$<br>$Kr_3(N_{S3}, N_{R3})$ |
| **Example 3** Extraction | Total number of trays and feed location | $[D_1, Q_{reb}^{C1}, Q_{cond}^{C1}]$<br>$[D_2, Q_{reb}^{C2}, Q_{cond}^{C2}]$ | $Kr_1(N_{S1}, N_{R1}, N_{M1}, E)$<br>$Kr_2(N_{S2}, N_{R2}, E)$ |
| **Example 4** Distillation sequences | Total number of trays and feed location | $TAC = TAC_{C1} + TAC_{C2}$ | |
| **Example 5** Demethanizer | Heat exchange to precool the feed and the flux that is sent overhead | $[D, Q_{cooler}, TAC]$ | $Kr(MF, Q_1, Q_2)$ |

**Example 1.a. Conventional distillation column with fixed number of trays**

The Kriging models are not explicitly introduced on the optimizer; they act as a black box to it. By doing this, software's like MatLab, which can easily work with matrixes, can be used. Only a local minimum can be assured. It is an operation problem, as the column is already designed. To improve the convergence variables such as the reflux ratio and the reboiling ratio should be chosen above other variables such as the outlet conditions. The Kriging model is trained with 100 points and tested with cross-validation.

**Example 1.b. Conventional distillation column with variable number of trays**

The model can be explicitly introduced to the optimizer, as it is an accurate representation of the model. Consequently, a global minimum can be found. This is recommended when the number of Kriging models as well as the sample length is short.

**Example 2. Divided wall column (DWC)**

This type of columns reduce the amount of energy required to heat the N-1 columns, where N are the number of components to split from the initial mixture. The condensers and reboilers are switch to thermocouples. Divided wall columns are simulated as two conventional distillation columns where the energy and mass streams are connected. All the columns are considered as one in the mathematical model.

**Example 3. Extractive distillation system**

Extractive distillation systems are used when the boiling points of the initial stream components are similar. Entrainers are added to it so a high volatile component helps the separation of the mixture. The process is simulated as two conventional distillation columns.

The first column is feed with the entrainer and the mixture, splitting one of the components at TOP and the rest of the mixture plus the entrainer at BOTTOM. This stream will feed the second column, which will recycle the entrainer to the first one through TOP.

**Example 4. Distillation sequences. Non Sharp separations**

Instead of splitting individually the components, several multicomponent product streams are required. The proposed superstructure contains all possible alternatives to get the desired products. The optimal configuration wants to be determined, so the problem complexity increases. The problem can be solved as an MINLP using a Big-M approach.

The resulting Kriging model is not accurate enough to ensure that the optimum is real. The steps 5 and 6 from the methodology described above are then applied to confirm whether the value is the optimum or not.

**Example 5. Demethanizer column**

In this process methane want to be recover from an initial stream. Several columns are used. The first one is a cryogenic high pressure column, where methane is separated overhead and the other heavier hydrocarbons are taken as bottom. The Feed stream is pre-cooled by heat exchangers.

## 6.2. Optimization-based design of crude oil distillation units using surrogate column models and a support vector machine (Ibrahim et al., 2018)

In the paper two different surrogate models are used. A set of ANN's (7 in total) is used to model the column (inputs: column structure and operating conditions; outputs: column performance), including

the heat recovery (done with pitch analysis). To reduce computational time a SVM is used to study feasibility constraints, so unfeasible designs are filtered when optimizing (implemented on MatLab). The optimization is done by a genetic algorithm, which is stochastic, minimizing the objective function (implemented on MatLab).

In the paper the following sequence is followed to tackle the problem.

1. Generating the data (selection of variables and simulation)
2. Creating the model (ANN's)
3. Neglecting points that doesn't converge (SVM)
4. Optimizing the model (genetic algorithm)

A detailed description of these steps is given below.

**1. Data generation**

To begin with relevant inputs and outputs are chosen (independent and dependent variables). Table 2 lists the variables.

*Table 2. List of dependent and independent variables*

| Input variables | | Output variables[*1] |
|---|---|---|
| Structure variables | Operating conditions | |
| Location of the feed tray<br>Number of trays in each section<br>Pump-around stream<br>Side-stripper draw streams. | Feed inlet T<br>T drop<br>Pump-around duties<br>Stripping steam flow rates<br>Reflux ratio. | ASTM D86 T5 and T95[*2]<br>F (product flow rates)<br>TS (supply T)<br>TT (target T)<br>E (enthalpies)<br>D (column section diameters) |

[*1] Each output variables will have its own ANN associated

[*2] ASTM D86 (method used to measure the batch distillation curve of petroleum) boiling temperatures of product i at 5% and 95% of vaporization

Samples are then generated using "Latin hypercube sampling". It consists on dividing the input variable space into intervals, where samples are created randomly from each interval.

The simulation of the samples is done with and interface between Aspen HYSYS v8.6 and MatLab R2015a (Peng-Robinson as the thermodynamic property package). Results of the simulation are recorded and labelled as feasible if the simulation converged and unfeasible if it does not.

From the 7000 samples generate, 59% (4130) simulations converged; for the remaining 41% (2870), the simulations did not converge. The sampling is carried out on a HP desktop PC with Intel(R) Core i5 processor running at 3.2 GHz, and 8 GB of RAM. It took around 1.5 h to generate the set of 7000 samples.

**2. Artificial Neural Networks (ANN's).**

The hidden layer contains 10 neurons. Uses a sigmoid function. The output layer size depends on the output variables associated to each ANN. It uses linear transfer function. The number of layers and neurons was settle applying trial and error.

These layers are trained to input-output data points using “back-propagation”.

The total number of feasible sampling points is divided into 3 sets:

- Training set (70%)
- Validation set (15%)
- Testing set (15%)

Model’s accuracy is checked with the coefficient of determination, being approximately 0.9999 for all ANN’s.

### 3. Super Vector Machines (SVM)

Application of SVM for binary classification of samples: feasible of unfeasible. An optimal hyperplane will be defined so both classes are optimally separated. Feasible samples are the ones that converged in the simulation, while those leading to unconverged simulations are defined as unfeasible. The SVM is defined in Equation(16).

$$y(x) = sign(w \cdot x + b) \tag{16}$$

Where **y(x)** can be +1 or -1 depending on the feasibility of the sample, **w** is the normal vector of the hyperplane and **b** the bias. Given a new instance (x), the equation allows classification of the sample as feasible (+1) or unfeasible (-1).

In the paper the SVM has a third-order polynomial function with outputs +1 and -1.

The total number of sampling points will be divided into 3 sets:

- Training set (75%): these points are used to construct the model.
- Validation set (25%): these points are used while training to avoid overfitting.

The function “fitcsvm”, which is implemented in MatLab 2015a, is applied to train and validate the SVM. When no filter is applied only 70% of the designs are feasible. Other papers use ANN’s to filter data. However, in the paper it’s claimed that SVM demonstrated to be more efficient eliminating unfeasible designs, though ANN’s retained slightly more feasible designs.

### 4. Genetic algorithm

The framework applies a stochastic optimization algorithm (genetic algorithm, GA), to search for the optimal conditions that minimize the objective function, the total annual costs (TAC). Even though the surrogate models used can be easily optimized, the sigmoid function used in the ANN’s as well as the objective function given rise to a nonlinear, non-convex optimization problem. Gradient-based searches are unlikely to locate the global optimum. Therefore, a stochastic optimization method, GA, is used. The framework is optimized using Matlab.

# 7. Techno-economic analysis

In this section the main costs and revenues will be presented, so the economic analysis of the project can be done. The main source of information used to write this section is the book Chemical Engineering Design, SI Edition. In particular, Chapter 6, where the costing and project evaluation is done (Sinnott & Towler, 2020).

## 7.1. Costs

Costs can be split it in four types: fixed capital investment (FCI), working capital (WC), variables costs of production (VCOP), and fixed costs of production (FCOP). A brief description of each of these costs is given below.

### 7.1.1. Fixed capital investment (FCI)

Represents the total costs required to design, construct and install the plant. Elements to consider are:

- Inside battery limits (ISBL): procurement and installation of all the process equipment. Includes the direct field costs (price of the equipment) and the indirect field costs (insurances, social security and field expenses).
- Offsite costs (OSBL): investment costs that must be made to accommodate adding a new plant. Includes: electric main substations, power generation plants, steam or water lines, air separation plants, emergency services, etc. Offsite costs are usually estimated as a proportion of ISBL costs. An initial estimation can be 40% of ISBL, but it can range from 10%-100%, depending on the project's scope and its impact on the infrastructure.
- Engineering costs: include the costs of detailed design and other engineering services required to carry out the project. They are estimated as a 30% of ISBL+OSBL for small projects and 10% for large ones.
- Contingency charges: extra costs added into the project budget to allow from variation from the cost estimated. For instance, changes in prices, project's scope, subcontractor problems, etc. A minimum contingency charge of 10% of ISBL+OBSL should be considered, increasing up to 50% if the technology used is uncertain.

### 7.1.2. Working capital (WC)

The working capital considers the additional money needed to start the plant up and run it until it starts earning money. Consequently, the following items are considered:

- Value of raw material inventory.

- Value of product and by-product inventory.
- Cash on hand.
- Accounts receivable: products shipped but not yet paid for.
- Credit for accounts payable: products received but not yet paid for.

It typically represents between the 5-30% of the FCI.

### 7.1.3. Variable costs of production (VCOP)

Variables costs of production are proportional to the plan output or operation rate. Variable costs can usually be reduced by more efficient design or operation of the plant. For instance, heat integration reduces the utilities required, as heat is reused. VCOP typically include:

- Raw materials consumed by the process.
- Utilities.
- Consumables.

### 7.1.4. Fixed costs of production (FCOP)

Fixed production costs are costs that are incurred regardless of the plant operation rate or output. If the plant cuts back its production these costs are not reduced. Fixed costs include:

- Operating labour.
- Supervision: it usually takes 25% of operating labour.
- Direct salary overhead: payroll taxes, health insurances, etc. Usually 40%-60% of Operating Labour + Supervision.
- Maintenance: usually estimated as 3%-5% of ISBL, depending on the expected plant reliability.
- Rent of land: usually estimated as 1%-2% of ISBL+OBSL. If the land is purchased, the cost will be added to FCI and will be recovered at the end of the plant life.
- Taxes and insurance: usually 1%-2% pf ISBL.
- General plant overhead: charges to cover corporate overhead functions such as human resources, research and development (R&D), information technology, finance, legal, etc. It is usually estimated as 65% of total labour (including supervision and direct overhead) plus maintenance.

## 7.2. Objective function

When defining an objective function, different criteria's and factors will apply regarding what it wants to be minimized or maximized. Typically, objective functions express to monetary costs of a project over a period of time, being €/year or other similar units the most common ones. Consequently, all the

costs and revenues must be considerate to perform an accurate analysis of the project. The key factors to consider have already been described in section 7.1. Applied in a distillation process, the different factors are described below.

- Fixed Capital Costs: price of the column, which depends on the number of trays for distillation. Also the price of the exchangers, which depends on their surface.
- Variable Costs of Production: which can be summarized in the price of the utilities required to cool and heat the outlet streams.
- Revenues: the purity of the separated components will vary the price at which the outlet streams can be sold, directly impacting on the objective function. Perhaps by increasing some costs the revenues obtained are higher than the costs increase.

As the number of trays will vary in each sample, the amount of steel required will also vary. Contrary, the equipment required is constant, as all the columns considered have a condenser and a reboiler no matter which operating conditions are chosen. However, the size of these exchangers will depend on the duties. Consequently, exchanger duties affect both FCC and VCOP. The simplify the calculations the pressure has been considered constant, so no pressure drops affect the column. Finally, the outlet conditions will also vary, giving higher or lower revenues.

In the particular case study of this project, columns are studied separately, though future projects seek a global optimization of the whole process to decide which columns must be activated or deactivated. Although the global optimization is out of the scope of this project, the objective function defined will take it into consideration. Consequently, only the costs are being considerate.

The objective function equation, which is describe by the total annual costs (TAC), is defined by Equations (17).

$$TAC\left(\frac{\$}{kg\ TOP}\right) = \frac{FCC\left(\frac{\$}{year}\right) + VCOP\left(\frac{\$}{year}\right)}{Flow_{TOP}(\frac{kg\ TOP}{year})}$$

$$FCC = \left(C_{Reb} + C_{Cond} + NT \cdot C_{Tray} + C_{Col}\right) \cdot \frac{i(1+i)^t}{(1+i)^t - 1} \tag{17}$$

$$VCOP = C_{HU} \cdot HU + C_{CU} \cdot CU$$

Where "i" is the interest rate of the project, "t" the life time of the project, "$C_i$" the costs of the different elements, "Pj" the price at which the streams are sold, "HU" refers to the hot utilities and "CU" the cold utilities and "NT" is the number of trays of the column. These last three variables will be optimized by HYSYS given the objective function. The different costs must be evaluated regarding other variables.

$$C_{Reb} = 3.4 \cdot (24000 + 46 \cdot A(m^2)^{1.2})$$

$$C_{Cond} = 3.5 \cdot (24000 + 46 \cdot A(m^2)^{1.2})$$

$$C_{Trays} = 6 \cdot (110 + 380 \cdot D_c(m)^{1.8})$$

$$C_{Column} = 4 \cdot \rho_{steel} \cdot \pi \cdot (Trays + 2) \cdot h_{Tray} \cdot \left(\Delta V_{lat} - \frac{4}{3}\Delta V_{base}\right) \quad (18)$$

$$C_{CU} = 0.6 \cdot Q_{Cond}^{-0.9}\left(\frac{kJ}{s}\right) \cdot T(K)^{-3} \cdot CEPCI + 1.1 \cdot 10^{6} \cdot T(K)^{-5} \cdot 4.5$$

$$C_{HU} = 7 \cdot 10^{-7} \cdot Q_{Reb}^{-0.9}\left(\frac{kJ}{s}\right) \cdot T(K)^{0.5} \cdot CEPCI + 6 \cdot 10^{-8} \cdot T(K)^{0.5} \cdot 4.5$$

For equipment costs, which include the reboiler, the condenser, the column and trays, Equation (19) is used (Sinnott & Towler, 2020). For utility costs, which include hot and cold utilities, Equation (20) is considered (Ulrich & Vasudevan, 2006).

$$C_e = IF \cdot (a + b \cdot S^n) \quad (19)$$

Where $C_e$ is the purchased equipment cost on a U.S. Gulf Coast basis, Jan. 2007; parameters a, b, and n are cost constants; S is the size parameter and IF is the installation factor.

$$C_{s,u} = a \cdot CEPCI + b \cdot C_{s,f} \quad (20)$$

Where $C_{s,u}$ is the price of the utility, a and b the utility cost coefficients, $C_{s,f}$ is the price of the fuel in $/GJ and CEPCI the inflation parameter for projects (CE Plant Cost Index), with a current value of 638,8.

The parameters can be found at the cited bibliography. The final equations for each cost item are described above, in Equation (18).

# 8. Methodology

The general methodology designed to tackle surrogate modelling problem can be summarised with the steps listed below. Its implementation in this project is defined afterwards.

1. Study the separation with simplified models (i.e. FUG equations).
2. Define the sampling variables and intervals.
3. Eliminate the unfeasible designs.
4. Treat the data previous to fit the model. Some treatments include: the calculation of the objective function for each sample, the filtering of data, the shuffling, the normalization, and the division in sets.
5. Train and validate the model.

From the same dataset two surrogate models that represent the same column of the process are going to be compared.

Both models simulate the behaviour of the column. The main difference between them is the amount of data used for the training and validation. While the first one, named Model 1, includes the entire sampling, the second one, named Model 2, filters part of the data by purity and recovery.

A comparison between both of the models with shortcuts and columns will highlight the possible improvements of applying this approach. Other factors to be considered are the time consumed, the amount of unfeasible designs and the complexity of the models.

For both models the following steps, summarised in Figure 8.1, will be carried out:

1. Model the separation with the shortcut distillation unit in Aspen-HYSYS.
2. Define the sampling variables required to calculate the costs.
3. Define the procedure, including the number of samples and the order in which will be simulated to improve flowsheet convergence.
4. Study the non-converged cases and eliminate unfeasible designs.
5. Calculate the costs. Study of the data with a heat map.
6. Filter the data by purity and recovery (just in the Model 2).
7. Pre-treatment of the data, which includes the shuffle of data for a later normalization, and then the division into training and testing set.
8. Define the neural network properties and fit the data.
9. Plot results and calculate accuracy coefficients.

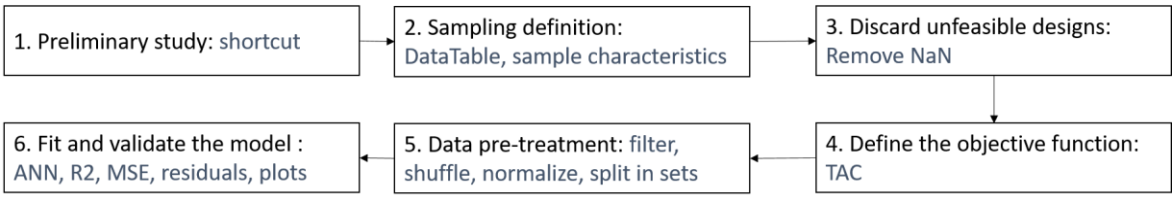


*Figure 8.1. Summary of the methodology steps*

A different way to explain this methodology is depicted in Figure 8.2. Each data point that feeds the surrogate model is defined by a previously implemented rigorous simulation. From this simulation, noise is reduced to study the process with respect the costs, instead of the vast variety of properties and calculations made by the simulation. The simplified points are eventually used to train and validate the surrogate model.

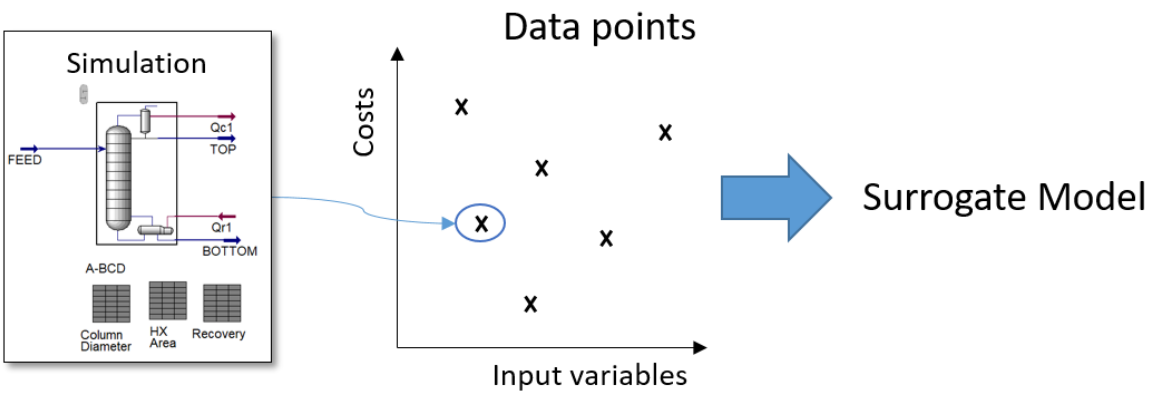


*Figure 8.2. Summary of the methodology by looking at the transference of the data*

# 9. Tools

## 9.1. Simulation of the process

The surrogate model is defined by data points. Their characteristics are direct consequence of the simulation they were sampled from. Thus, a correct definition of the simulation is an indispensable step in the process.

Commercial process simulators are powerful calculous tools with several amounts of data, not only containing the physical properties of thousands of components, but also thermodynamic packages, equipment calculations, costs analysis, and operation conditions, among others. The most common software's are listed below.

- Aspen-HYSYS
- Aspen-PLUS
- UniSim
- Flesxim
- ChemSep
- CHEMCAD
- ProModel

The software chosen to simulate the process is Aspen-HYSYS, a process modelling environment for conceptual design and operations improvements in the context of chemical processes. As many other commercial simulators, Aspen-HYSYS tools allow the user to estimate mass and energy balances, physicochemical properties, liquid-vapour equilibrium and the simulation of several equipment in chemical engineering. The reason why Aspen-HYSYS has been chosen for the development of this project is the fact that some of his tools are able to connect to external sources, for instance Python. Moreover, the previous experience with this software was a plus when the decision was taken.

To begin with, steady state conditions and the Peng-Robinson thermodynamic property package (PR) are the initial hypothesis considered. PR is the industry standard for simulating gas streams, especially in hydrocarbons processes.

As explained in Section 5.4, two typical distillation column representations are the "shortcut" model and the "column" model. The first one is commonly used as initial template of the process due to its simplicity and easy convergence. Figure 9.1 illustrates the shortcut model interface in the Aspen-HYSYS software.

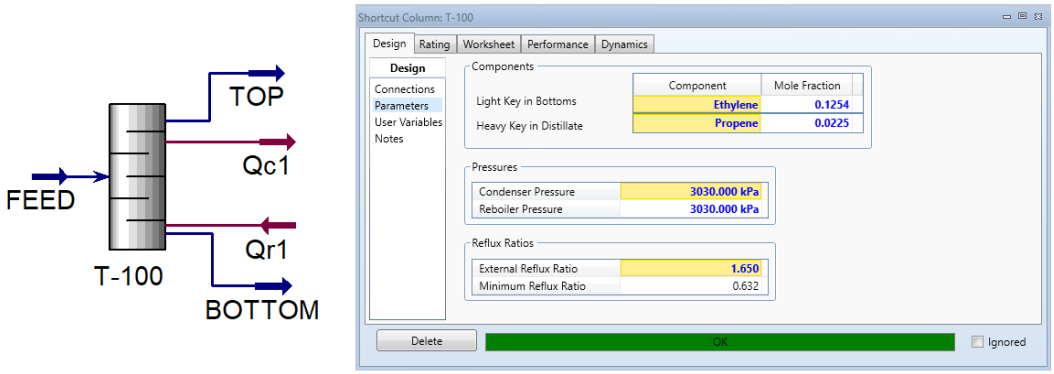


*Figure 9.1. Aspen-HYSYS shortcut model interface*

To define a shortcut model in Aspen-HYSYS the following information is required. A defined FEED stream, the LK & HK in BOTTOM and TOP respectively, the condenser and reboiler pressure, and the external reflux ratio, which has to be greater than the minimum reflux ratio. The simulation converges when this information is given. The value of the HK & LK variables determines the outlet mass flows, while the external reflux ratio defines the actual number of trays required to achieve the separation. Boundaries for the sampling are obtained by changing these variables from high to low accurate separations, in terms of TOP purity and reflux ratio.

The simulation of the "column" model requires more steps to be done. When this object is added to the spreadsheet, the user must complete several interface screens to firstly converge the simulation. The information required in these screens includes: the mass and energy streams, the number of trays and feed tray, the fraction of liquid condensed at TOP, the reboiler configuration, the pressure and pressure drops at the condenser and reboiler, the distillate rate and the reflux ratio (which are the initial degrees of freedom), and an optional estimation of the temperatures. Figure 9.2 illustrates the column model interface in the Aspen-HYSYS software.

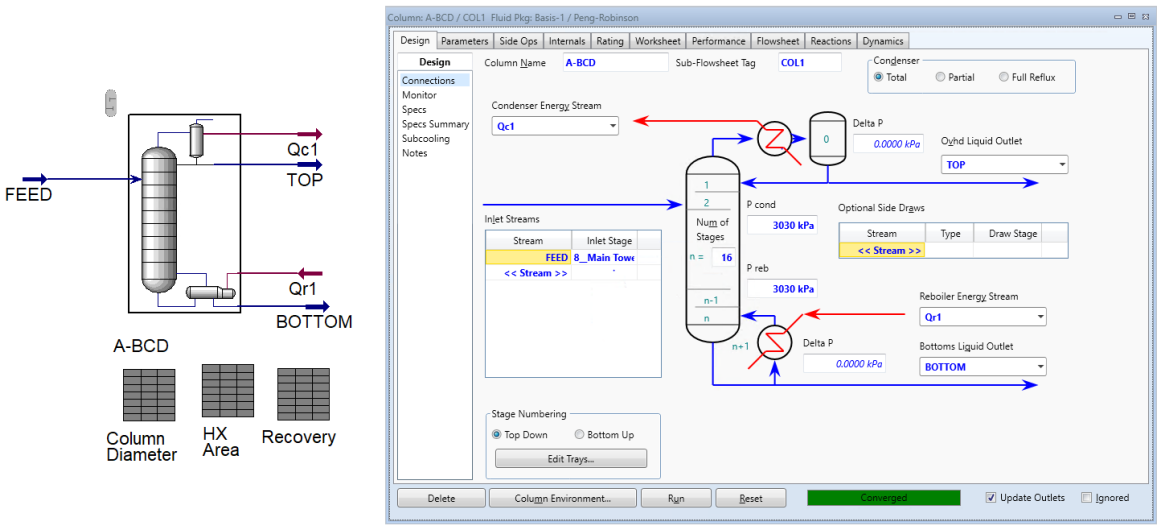


*Figure 9.2. Aspen-HYSYS column model interface*

Although other degrees of freedom could be selected to converge the column, the reflux ratio (RR) and the distillate rate ($D_{flow}$) enable straightforward estimations of the separation.

For instance, in the distillation of the lightest component (A) from a mixture of four (ABCD) the $D_{flow}$ should be close to the initial amount of A in the feed stream. If $D_{flow}$ is considerably lower, the recovery of A will be poor. The reason is that not all A is distillate, so part of it exits the column at the BOTTOM conditions. Contrary, if $D_{flow}$ is rather higher, the top fraction of A will be negatively affected, as not only A is distillate, reducing the purity of the TOP stream.

The reflux ratio value can be estimated with KPI, being approximately 1.2 times the minimum reflux ratio, calculated by the shortcut model.

The convergence of the column model implies the calculation of the liquid-vapour equilibrium, the mass and energy balances, and several other calculations in each tray of the column. Figure 9.3 represents the composition profile for each tray, being 0 the tray below the condenser and 18 the tray above the reboiler. The outlet streams are defined by the first and last tray, though HYSYS calculates the entire column.

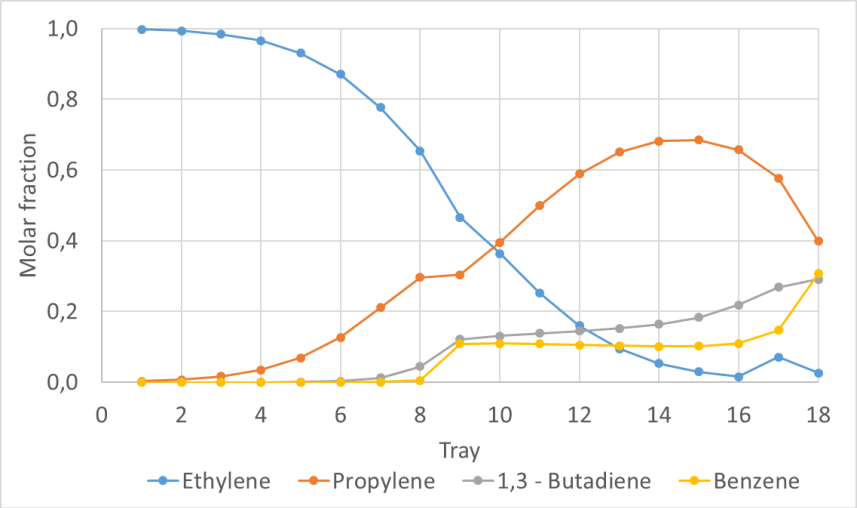


*Figure 9.3. Composition profiles in the column model with respect the tray.*

Once the simulation is defined, the next step is the sampling, which it is possible due to the Aspen-HYSYS object DataTable. It allows the user to read and write variables in the simulation. This Aspen-HYSYS object can be connected to external environments, such as Python, enabling the user to export data from it.

The sampling shall include the other required variables to calculate the objective function defined in Section 7.2. The list of variables includes: the condenser and reboiler areas, duties, and temperatures, the column diameter and tray spacing, and the component mass flows, which are not necessary to define the costs but will be handful the study of the recoveries and purity.

The "spreadsheet" object in Aspen-HYSYS allows the user to export variables from other objects to a spreadsheet similar to Excel, which enables the calculation of some of them, for instance the area of the condenser and reboiler or the diameter of the column.

For the condenser and reboiler area KPI's have been used. The minimum $\Delta T$ required in heat exchangers is typically considered as 10ºC. From this value the area can be calculated using Equation (*21*). Where U is the overall heat transfer coefficient, Q the condenser or reboiler duty, and A the area of exchange.

$$Q = U \cdot A \cdot \Delta T \tag{21}$$

The column diameter has also been settled as a constant, as making it variable means having to consider the flooding in the objective function, which it is out of the scope of the project. Consequently, it is determined by KPI's. Typically, the vapour speed along the column is about 1 m/s (Seider et al., 2010), thus the column diameter (d) is defined depending on the vapour flow (V), as it is described in Equation (*22*).

$$V\left(\frac{mol}{s}\right) \cdot \frac{8.31 \cdot T\,(K)}{P\,(Pa)} = 1\frac{m}{s} \cdot \left(\pi \cdot \frac{d^2}{4}\right) \tag{22}$$

Finally, the key variables to calculate the costs are exported to a DataTable. Table 3 columns summarises the DataTable characteristics, while the rows, which represent the several variables selected, will be later explained with the Case study.

*Table 3. Process DataTable*

| Object | Variable | Value | Units | Tag | Access Mode |
|---|---|---|---|---|---|
| Main Tower | Number of Stages | 16 | -- | NT | Read/Write |
| A-BCD | Spec Value (Reflux Ratio) | 0.597 | -- | RR | Read/Write |
| A-BCD | Spec Value (Distillate Rate) | 6.124 | kg/h | Distillate_Rate | Read/Write |
| A-BCD | Diameter (Diameter_1) | 0.4164 | m | Column_Diameter | Read |
| A-BCD | Tray Spacing (Tray Spacing_1) | 0.35 | m | Column_Spacing | Read |
| A-BCD | Product Stream Comp Mass Flows (Pure_A-Ethylene) | 5.527 | kg/h | TOP_Flow_A | Read |
| A-BCD | Product Stream Comp Mass Flows (Pure_A-Propene) | 592.9 | kg/h | TOP_Flow_B | Read |
| A-BCD | Product Stream Comp Mass Flows (Pure_A-13-Butadiene) | 4.017 | kg/h | TOP_Flow_C | Read |
| A-BCD | Product Stream Comp Mass Flows (Pure_A-Benzene) | 2.23E-6 | kg/h | TOP_Flow_D | Read |
| A-BCD | Product Stream Comp Mass Flows (BCD-Ethylene) | 617.6 | kg/h | BOTTOM_Flow_A | Read |
| A-BCD | Product Stream Comp Mass Flows (BCD-Propene) | 2.917 | kg/h | BOTTOM_Flow_B | Read |
| A-BCD | Product Stream Comp Mass Flows (BCD-13-Butadiene) | 3.266 | kg/h | BOTTOM_Flow_C | Read |

| | | | | | |
|---|---|---|---|---|---|
| A-BCD | Product Stream Comp Mass Flows (BCD-Benzene) | 4.994 | kg/h | BOTTOM_Flow_D | Read |
| A-BCD | Condenser Outlet Temperature | -9.332 | Cº | T_Condenser | Read |
| A-BCD | Reboiler Outlet Temperature | 91.77 | Cº | T_Reboiler | Read |
| A-BCD | Duties Summary (Condenser) | -3.68E6 | kJ/h | Duty_Condenser | Read |
| Qr1 | Heat Flow | 3.57E6 | kJ/h | Duty_Reboiler | Read |
| HX Area3 | Cell Matrix (B-5) | 123.9 | $m^2$ | Area_Condenser | Read |
| HX Area3 | Cell Matrix (C-5) | 26.41 | $m^2$ | Area_Reboiler | Read |

The input variables that will run the simulation are those marked as “Write” or “Read/Write”, while those with access mode “Read” are the other variables required to calculate the costs. All kinds of variables can be added to a DataTable, though some of them report errors when running the code. Variables from spreadsheets can be added too (i.e. last row in Table 3), facilitating the later treatment of the data.

The feed tray is a variable that DataTables cannot read, though its proportion with the NT variable remains constant if NT > 5. Consequently, the feed tray will be manually set by approximation, considering the inlet conditions of the stream and the separation that will occur.

## 9.2. Communication interface

The communication interface as well as the following tools designed in this project have been developed in Python programming language. Not only is Python an open source coding language, but it also has a huge community daily working with it, which eases the programming. Other programming languages considered were C, C++, Java, MatLab, and VBA.

The data gathering based on the sampling selection is done through a connection with Aspen HYSYS from Python. Not only does the code allow to write and read data, but also to run simulations and call the optimizer. Thanks to the code, thousands of samples can be made and exported to an Excel file in hours. A later treatment of the data will define the input and output variables necessary to train the surrogate model. The “pandas” library allows the user to read and export csv or Excel files. The function “DataTable” makes the treatment of the data straightforward. Figure 9.4 summarises the main steps of the connection.

To begin with, the code requires the location of the simulation as well as the location where HYSYS is installed. Then, a first python function opens the HYSYS case and connects to the desired DataTable. The simulation of the sampling points is based on an object class that has three basic functions defined.

1. The function “Read” exports all the data from the DataTable into a python dictionary. This data can be later transformed into a panda’s
   data frame. It requires the HYSYS DataTable object as input.

2. The function “Write” uses a panda Data Frame as sampling points to change the values of the input variables in the given DataTable. Consequently, the dimension of the Data Frame must match the dimension of input variable in the DataTable. For instance, if the HYSYS Data Frame allows three variables to be modified or written, the sampling points given by the Data Frame must have three dimensions too. Each variable in the DataTable has to be defined with a TAG and a read or write option, so the code can distinguish which variables must be changed or exported.
3. The function “Run” runs the simulation once the new sampling point is written in the DataTable. It does not need any input variable.

For each sampling point the first three functions will be always called. Iterating over the Data Frame that represents all the sampling points a second Data Frame can be obtained with the simulation results. Finally, results are exported to an Excel file.

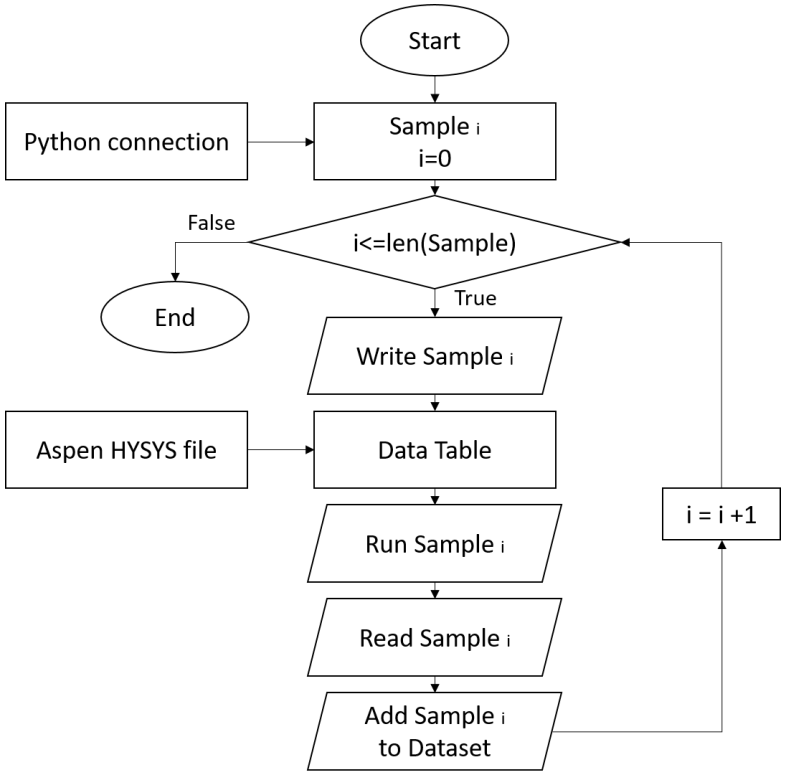


*Figure 9.4.Flowchart of the connection Aspen HYSYS - Python.*

## 9.3. Sampling and Data Processing

### Latin Hypercube Sampling

One of the most common sampling methods is the Latin Hypercube sampling (Sheikholeslami & Razavi, 2017). Latin hypercube sampling (LHS) is inspired by the concept of "Latin square" from combinatorial mathematics, where an n-by-n matrix is filled with n different numbers such that each number occurs exactly once in each row and exactly once in each column (i.e. Figure 9.5). Like Latin squares, the basic idea of LHS for a 2-dimensional space and a sample size of n is partitioning each dimension into n disjoint intervals (levels) with equal marginal probability of 1/n and then randomly sampling once from each interval to ensure that there is only one point at each level.

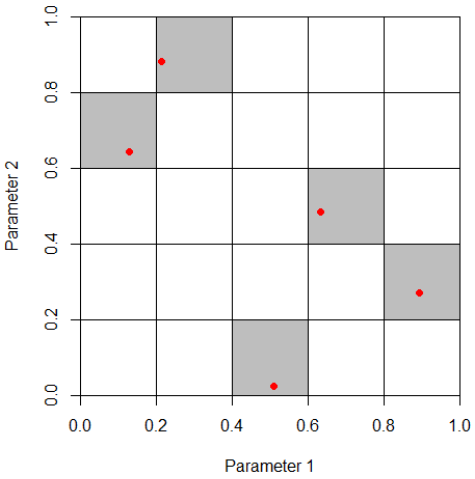


*Figure 9.5. Two dimensions of a uniform random LHS with 5 samples (Carnell, 2020).*

The python coded written to make the sample is based on the "uniform.random" function available in the "NumPy" library. It draws samples from a uniform distribution. Samples are uniformly distributed over the half-open interval [low, high]. In other words, any value within the given interval is equally likely to be drawn by uniform. However, this values are normalized from 0-1, so a later scaling is done to get "n" random samples uniformly distributed.

### Treatment of unfeasible samples

Data exported from the sampling defines those designs that did not converge, from now on called unfeasible designs, with a value of -32457. Changing the value to "NaN" allows the user to print and drop the unfeasible designs with the pandas functions "isna().any(axis=1)" and "dropna()". A study of the unfeasible designs will be helpful to check whether there are possible sampling mistakes or just the well-known HYSYS hysteresis. Figure 9.6 depicts the distribution of NaN's for a random set of data.

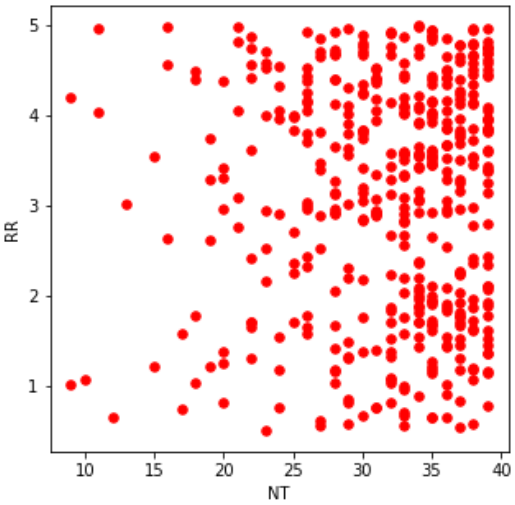


*Figure 9.6. Unfeasible design distribution for the NT and RR variables of a random set of data*

### Costs calculation

The "Read" variables of the DataTable will define the different costs described by Equations (17) and (18), so data can be reduced to the input variables, which are the "Write" variables in the DataTable, and the total costs. The cost percentage of each of the elements can be calculated to study how costs are distributed. By plotting heat maps the global cost can be easily studied for the sampling region chosen.

### Data filtering

The data filtering is only applied in Model 2. Moreover, it is the main difference with respect to Model 1, as not all the data is used in the modelling.

The random sampling will lead to several data points with different input variables but the same outlet conditions. Consequently, the same outputs can be achieved with several designs and operative conditions, which results in different costs. Therefore, data can be filtered by eliminating those points that lead to the same outputs but at higher costs. The aim of Model 2 is check whether a previous filtering of the data affects the accuracy and results of the eventual surrogate.

Figure 9.7 describes how the filtering is implemented. The code makes use of the function "groupby()", available in pandas DataFrames, to pick filtered values.

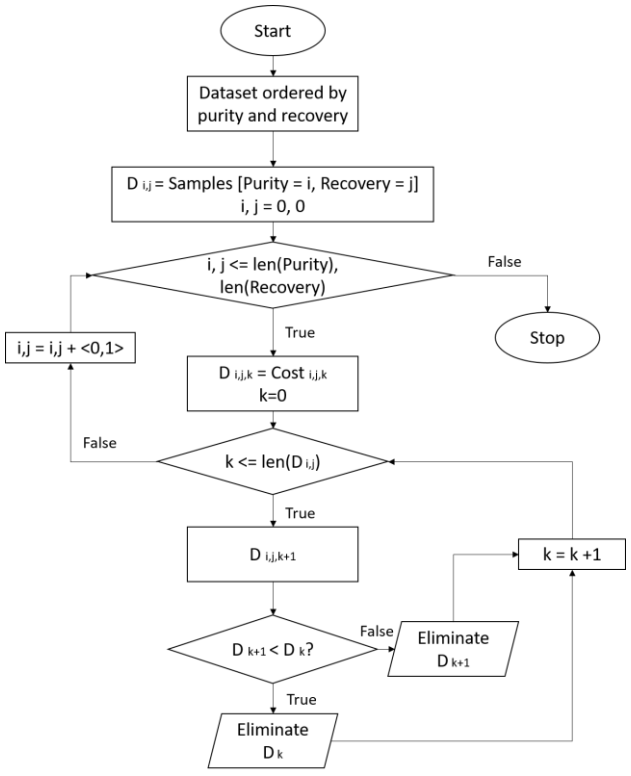


*Figure 9.7. Filtering flowchart*

It can be also summarised by Equation (*23*).

$$Dataset_{Filtered} = \sum_{i=0}^{Purity} \sum_{j=0}^{Recovery} \min\left( \sum_{k=0}^{\frac{Cost}{kgTOP}} Dataset_{i,j,k} \right) \quad (23)$$

**Data preparation to fit the surrogate model**

Panda's library offers several functions to summarize large datasets. The function "info()" indicates the type of values that each data column has, even the number of NaN's is indicated. The function "boxplot()" plots dataset columns characteristics such as the mean, standard deviation, and percentiles. It is useful to determine whether normalization is required or not. Finally, the function "describe()" summarises the information plotted by "boxplot()" with numbers.

Figure 9.8 represents the main steps in the preparation of data to fit the model.

Even though data may have been run randomly, it is always advisable to shuffle data before even normalising it. From the "sklearn" library the function "shuffle(DataFrame)" can be imported.

Once data is shuffled, it can be normalised. Again, "sklearn" library offers several tools to normalize data. While "MinMaxScaler" is typically recommended in classification problems, the normalization method implemented as "StandardScaler" is usually chosen in regression ones. The parameters of this normalization, which are the mean and the variance, can be saved for later transformations with the functions "mean_" and "var_" that "StandardScaler" incorporates.

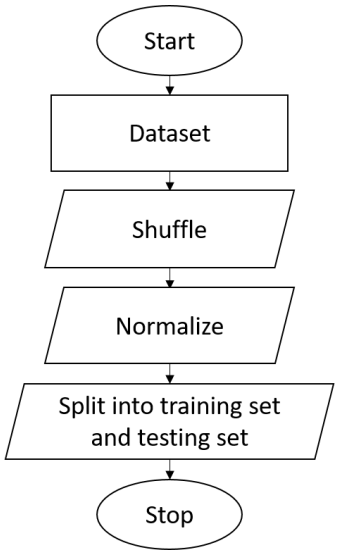


*Figure 9.8. Pre-treatment flowchart*

Data can be then split into a training set and a testing set. The division is defined by a threshold. It is crucial for data to be shuffled before it is split, as training set must contain data from the whole surface, not just one particular part. Finally, the training set is ready to fit the surrogate model. Validation set will be later used to check the model's performance.

## 9.4. Model. Artificial Neural Network

The python library Keras is a user interface for deep learning that it is based on the TensorFlow library, which uses infrastructure layers for differentiable programming. To simplify, the library TensorFlow allows the manipulation of tensors, as N-dimensional arrays, while Keras makes the creating of deep

learning tools straightforward, using the calculations of TensorFlow. Consequently, some of the commands are quite similar to NumPy library. Moreover, Keras has been developed to enable fast experimentation, which means that it is user friendly.

The three main functions of Keras that allows the used to create an ANN are described below and summarised in Figure 9.9.

The function “Sequential”, in which not only are the number of hidden layers and neurons defined, but also the several activation functions. Once “Sequential” has been run, the function “summary” allows the visualization of the empty neural network, with all the training parameters. As explained in Section 4.1.1, hidden layers commonly use the “ReLu” activation function, while the output layers of regression problems require the “linear” one. Section 3 also claimed the preference in low complexity models. Therefore, a couple of hidden layers with low number of neurons are initially proposed. If underfitting occurs, these parameters will be increased.

The function “compile”, where the loss function to minimize and the optimized are decided. More than one loss function can be calculated and later plotted, though just the one defined in the “loss” variable will be used to train the network parameters. The optimizer “Adam” and the loss function mean squared error (MSE) are traditionally recommended in regression problems.

The function “fit”, in which the training set is specified, as well as the number of epochs, the batch size and the validation set. These fitting parameters define the computational time that the network will require to be trained. The “epochs” variable refers to the number of times that data is fit to the network, training its parameters. The “batch_size” is the number of training points fitted simultaneously in the network. Finally, the validation set, which is not the same as the testing set, separates part of the training set to check later the performance and accuracy of the fit.

The “TicToc” function is imported from the “pytictoc” library to keep the computational time of each model run.

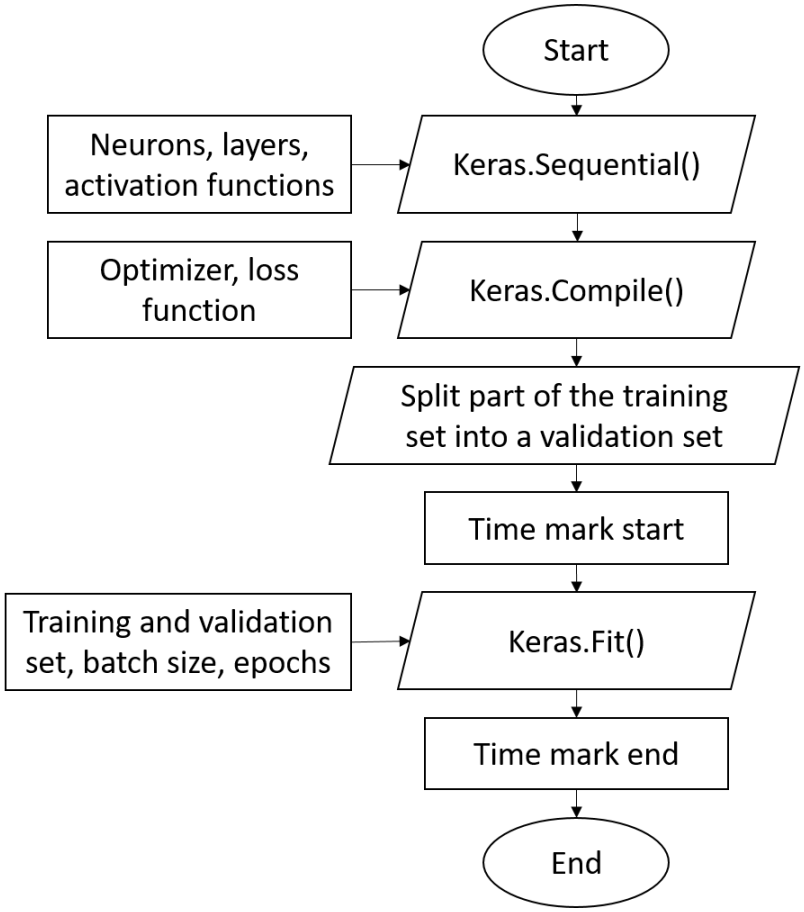


*Figure 9.9. Flowchart of the model based on Keras ANN's.*

# 10. Case Study

The proposed framework has been applied to the synthesis of the process of polyethylene pyrolysis for the recovery of hydrocarbons. The outlet conditions of the pyrolysis furnace are taken from the bibliography (Kannan et al., 2014), as well as the case study presented in this project (Somoza-Tornos et al., 2020). The outlet of the pyrolysis reactor is cooled and compressed to enter the distillation sequence where the different hydrocarbons may be recovered. For the sake of simplicity and due to the different boiling point of methane compared to the rest, the stream is demethanized before entering the distillation sequence, leading to a feed composition of: 47% ethylene, 20% propene, 16% 1,3-butadiene, and 17% benzene. Secondary components such as propyne and 1-butene are recovered with 1,3-butadiene since their low concentration does not justify two extra separation stages.

*Table 4. Possible distillation columns in a four-component separation*

| 1st process column | 2nd process column | 3rd process column |
|---|---|---|
| A-BCD | A-BC | A-B |
| AB-CD | AB-C | B-C |
| ABC-D | B-CD | C-D |
| | BC-D | |

Table 2 summarizes all the possible distillation columns present in a four-component mixture, being A the lightest component (ethylene), and D the heaviest one (benzene). The proposed methodology seeks to be applied to all of them, though just preliminary results on the first one has been taken.

The separation process studied here is then the distillation of ethylene, from a mixture that includes propene, 1,3-butadiene, and benzene. Figure 10.1 depicts the main variables of this distillation column, where black, blue and orange colours represent the dependent variables, the independent variables and the parameters chosen, respectively.

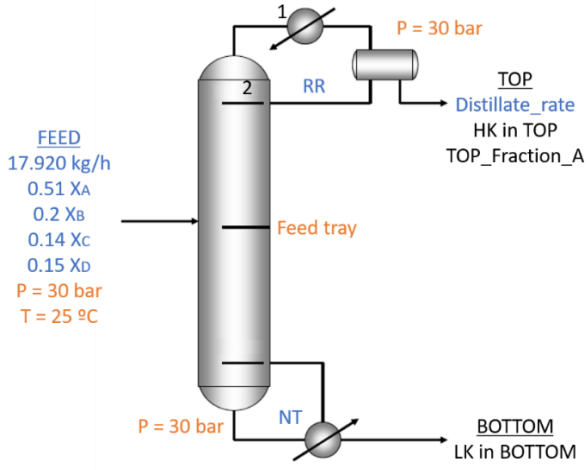


*Figure 10.1. Process flowsheet of the A-BCD column*

# 11. Results and discussion

## 11.1. Choosing the sequence

To begin with, a study of the constant feed flow has been done to determine which column model the first. Table 5 presents its characteristics, not only indicating the molar fraction, but also the boiling temperatures and K-values essential to determine the volatilities. The molar feed rate is 426.80 kmol/h.

*Table 5. Feed flow properties*

| | Component | X | Tb | K | α | $\alpha_{adjacent}$ |
|---|---|---|---|---|---|---|
| A | Ethylene | 0.51 | 104.00 | 1.851 | 84.37 | 3.44 |
| B | Propylene | 0.20 | 225.00 | 0.539 | 24.56 | 2.81 |
| C | 1,3-butadiene | 0.14 | 268.75 | 0.192 | 8.75 | 8.75 |
| D | Benzene | 0.15 | 353.20 | 0.022 | 1.00 | |

From this information, heuristics can be applied to first guess the likeliest optimum sequence. Heuristic 2 claims the importance of promoting direct sequences, as lightest components are usually the most valuable ones. In this particular case this is in tune not only with Heuristic 3, as component A is the most abundant one, but also Heuristic 4, as separation A-BCD is the closest one to equimolarity, being A's molar fraction up to 0.51.

Underwood's simplified equation can also be applied, to check whether the conclusions taken from heuristics agree with other simplified methods. Table 6 summarizes the results from applying the equation, ordering the possible sequences regarding the vapour minimum requirements.

*Table 6. Underwood's simplified method to classify each possible sequence regarding the vapour's minimum requirements*

| Case | Sequences | | | V extra | Order |
|---|---|---|---|---|---|
| 1 | A-BCD | B-CD | C-D | 16.84 | 1 |
| 2 | A-BCD | BC-D | B-C | 116.80 | 2 |
| 3 | AB-CD | A-B | C-D | 276.93 | 3 |
| 4 | ABC-D | AB-C | A-B | 613.39 | 5 |
| 5 | ABC-D | A-BC | B-C | 349.22 | 4 |

Direct sequence is claimed to be the best choice again. These results match the conclusions from other papers in which the same separation process was considered (Somoza-Tornos et al., 2020). Consequently, the first column of the "Case 1" sequence is chosen as the case study of this project.

## 11.2. Model 1

### Sampling and NaN's

The sampling includes three variables: number of trays (NT), reflux ratio (RR), and distillate rate ($D_{flow}$). The sampling boundaries are taken from shortcut results, applying high and low HK & LK in TOP and BOTTOM. The final boundaries are 5 – 20 for the NT (natural numbers), and 0.3 – 3 for the RR (real numbers). For the distillate rate, boundaries have been chosen considering that 51% of the feed flow should be ideally evaporated. A total of 10.000 samples are run, as done in similar papers (Ibrahim et al., 2018). Figure 11.1 shows the resulting sampling map, for the RR and distillate rate variables.

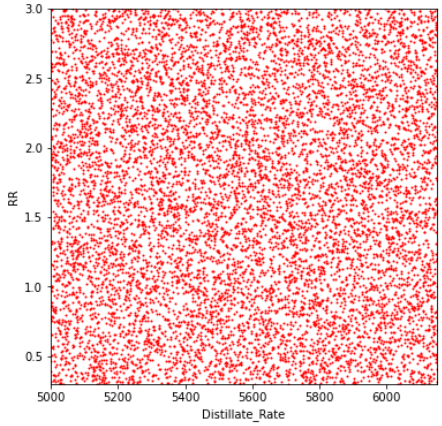


*Figure 11.1. Representation of the 10.000 samples resulting from the LHS method.*

The 10.000 samples are randomly created, so samples with high NT's can be followed by samples with low NT's. This becomes more tedious for the simulator as the number of samples increases, leading to large computational times and more unfeasible designs. By ordering the sample points before its simulation, not only does Aspen HYSYS require less time, but also the feasibility of the proposed designs is improved. Table 7 depicts several simulations run under different conditions of ordering.

*Table 7. Computational time of samplings with different ordering methods.*

| Ordering method | Number of samples | Unfeasible samples (%) | Computational time* |
|---|---|---|---|
| Random | 1.000 | 13.2 | 1 h 43 min 47 sec |
| $RR \rightarrow NT \rightarrow D_{Flow}$ | 1.000 | 14.9 | 1 h 48 min 47 sec |
| $D_{Flow} \rightarrow NT \rightarrow RR$ | 1.000 | 11.8 | 56 min 52 sec |
| $NT \rightarrow D_{Flow} \rightarrow RR$ | 1.000 | 3.8 | 34 min 23 sec |

*Double processor: Intel(R) Xeon(R) Silver 4114 CPU @ 2.20 GHz 2.19 GHz, 128 GB of RAM

It can be noticed how the lack of order increases the computational time required, as well as the percentage of unfeasibility, describe by the number of NaN's. It seems that prioritizing the order in the NT variables is the most effective way to run the sampling, which it is comprehensible as the number

of trays is the key variable that defines the distillation column in HYSYS. Figure 11.2 depicts the NaN's distribution for the "Random" ordered sampling described above.

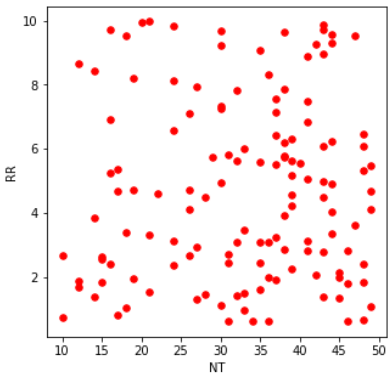


*Figure 11.2. NaN distribution for the "Random" sampling.*

Studying the properties of these NaN's may lead to a better comprehension of the unfeasibility of the simulations. Hence, samples are plotted in the order they were simulated, so gradients in the three input variables can be spotted (Figure 11.3).

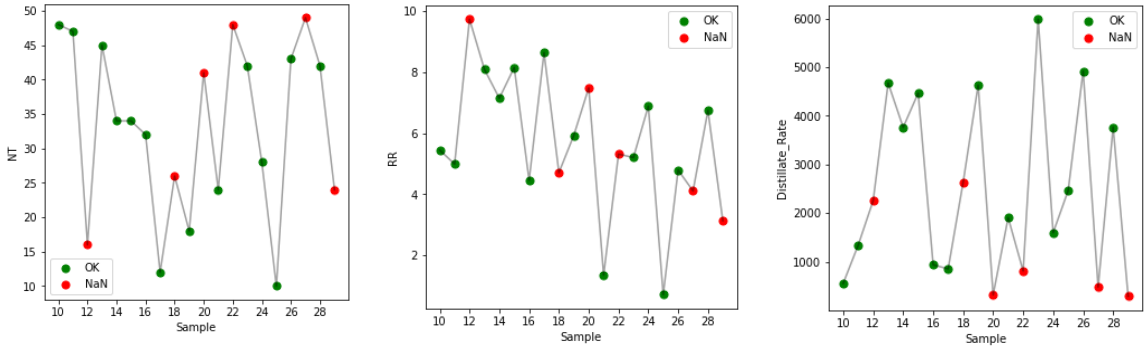


*Figure 11.3. Evolution of NaN's.*

The figure depicts how NaN's usually appear when the NT variable is modified, as changes in the column design increase the chances of infeasibility. Heavy changes in the distillate rate also lead to NaN's, though its relevance has not been noticed to be as significant as for the NT. The final sampling has then followed the configuration in Table 7 with lowest computational time. Its main properties are described in Table 8.

*Table 8. Computational time of the actual sampling.*

| Ordering method | Number of samples | NaN's (%) | Computational time* |
|---|---|---|---|
| $NT \rightarrow D_{Flow} \rightarrow RR$ | 10.000 | 0 | 1 h 43 min 47 sec |

*Double processor: Intel(R) Xeon(R) Silver 4114 CPU @ 2.20 GHz 2.19 GHz, 128 GB of RAM

**Data processing**

The nonexistence of unfeasible designs reduces considerable the computational time required. The next step is the calculation of the objective function for each sample. Table 9 summarizes the percentage of each element in it. A comparison with the shortcut values has been made seeking for differences in both approaches.

*Table 9. Costs percentage for each element in the objective function for the sampling points and a shortcut model.*

| Cost / Mg TOP | Sampling | Shortcut |
|---|---|---|
| **Column** | 0.84 (3.96 %) | 0.74 (3.45 %) |
| **Trays** | 0.11 (0.53 %) | 0.07 (0.31 %) |
| **Hot Utilities** | 15.50 (72.41 %) | 15.53 (72.69 %) |
| **Cold Utilities** | 2.84 (12.94 %) | 2.84 (13.31 %) |
| **Reboiler** | 0.80 (3.75 %) | 0.80 (3.75 %) |
| **Condenser** | 1.38 (6.41 %) | 1.39 (6.48 %) |
| **TOTAL** | 21.47 | 21.37 |

Utilities represent 85 % of the total cost, while the size of the column and the cost of the trays just represents 4.5 % of it. Shortcut values are quite similar from the ones in the sampling. The differences can be explained due to the simplifications of the FUG method implemented in the shortcut, as the temperatures, duties and NT estimated are slightly different compared to the column model in HYSYS.

Figure 11.4 depicts several heat maps that represent the three independent variables of each sample in the labels and a target variable with the colour show the sampling distribution in terms of purity (TOP molar fraction of component A, the LK), recovery (TOP kg of A / Feed kg of A), and costs (objective function).

Not all the samples with high purities have good recoveries. The highest recoveries can be found at the upper distillate rate values, as the sampling interval has been chosen to maximize this separation. The bottom heat map of the figure depicts a direct dependence of the cost with the reflux ratio, while the distillate rate or the NT seem to have no relevance on it. The remarkable importance of the hot utility costs may explain the distribution in this figure. Consequently, it exists sampling points with high recoveries at considerably low RR, which have lower costs.

Afterwards, data is shuffled, normalized, and split into a training and testing set. Part of the training set will be later separated as a validation set.

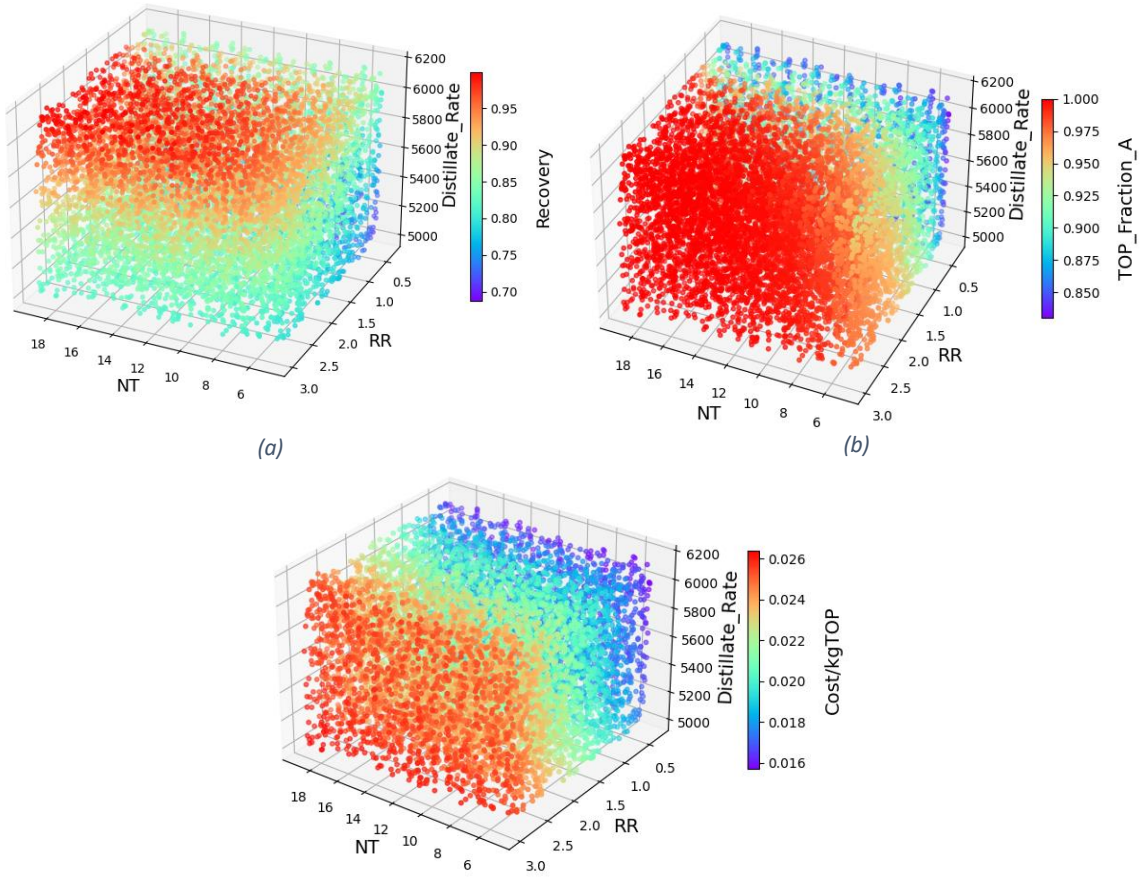


*Figure 11.4. Heat Maps of the entire sampling (10.000 samples). a: Recovery, b: TOP_Fraction_A, c: Cost/kgTOP*

**Fitting the model**

The training set fits the model as shown in Table 10. The testing set represents 10 % of the total sample points, while the validation set is the 15 % of the remaining training set. The total time required to train and validate the model is 21.1 seconds.

*Table 10. ANN simulation model properties.*

| | | | |
|---|---|---|---|
| Training set: | 7650 samples | Neurons 1st layer: | 6 |
| Validation set: | 1350 samples | Neurons 2nd layer: | 3 |
| Testing set: | 1000 samples | Epochs: | 75 |

The evolution of the error with respect the epochs is plot in Figure 11.5 to check whether the model has been trained enough or requires the data to be fed more times. The resulting plot shows how MSE reaches an apparent minimum as the number of epochs increases. Although the MSE of the validation set is higher when compared to the training set, the proper disminution of it stresses the lack of overfitting in the model, as new data is predicted properly.

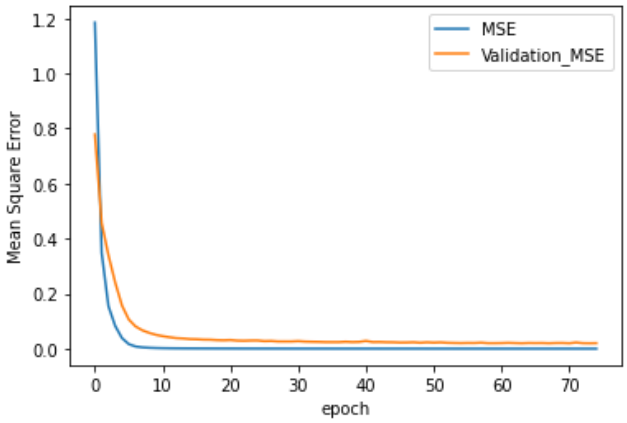


*Figure 11.5. Evolution of the MSE in the training and validation set with respect the epochs for the simulation model.*

By reversing the previous normalization residuals can be plotted. The comparison between the model predictions and the testing set is shown in Figure 11.6 and Figure 11.7.

The first of these two plots compares the model predictions with the actual values in the testing set. The ideal values line is a linear function with a slope equal to 1. If model predictions are distant to the ideal value, it means the costs predicted by the model do not match the actual values. The coefficient of determination is 0.9996, indicating that the model is able to explain 99.96 % of the variance of the outputs, which indicates the goodness of the fit.

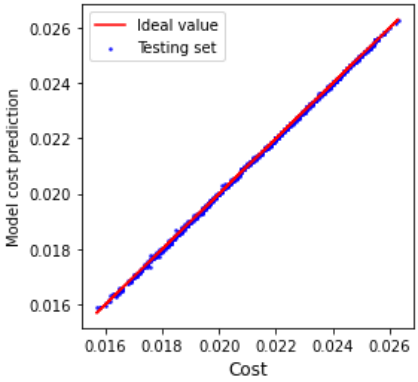


*Figure 11.6. Model predictions of the cost compared to the sampling values of it.*

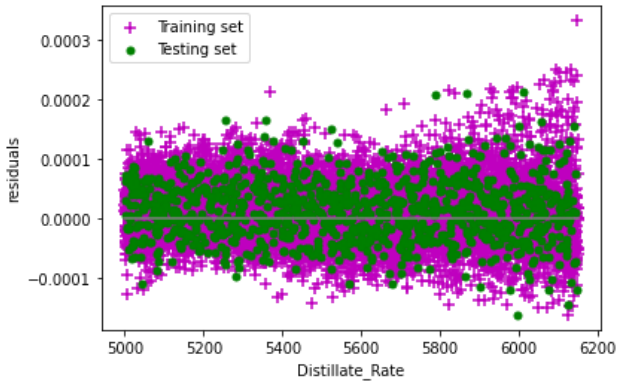


*Figure 11.7. Residuals of the training and testing set with respect the RR.*

The residuals distribution in Figure 11.7 shows no pattern, which reinforces the lack of overfitting or underfitting in the model. If the number of neurons decreases, for instance 6 neurons in only one layer, the residuals distribution adopts what is commonly known as the “banana pattern” (Figure 11.8). Pattern distributions in the residuals imply that the model is not able to entirely describe the behaviour of the process, as errors are not random.

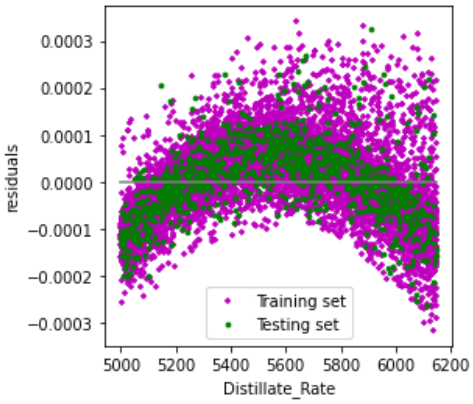


*Figure 11.8. Residuals distribution with a lower number of neurons (6 - 0 in the first and second layer respectively).*

To gain further confidence in the ANN model, a comparison between the samples and the model prediction at a constant number of trays is plot in Figure 11.9 and Figure 11.10, and no remarkable differences can be appreciated.

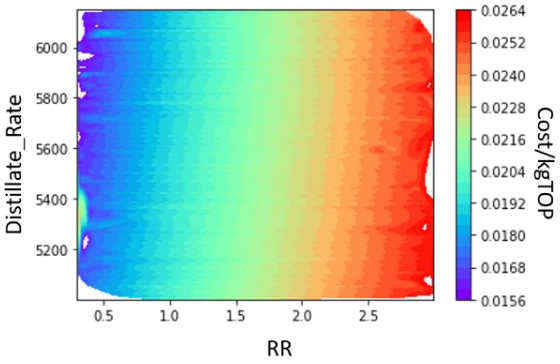


*Figure 11.9. Heat Map of the sampling points at NT = 8*

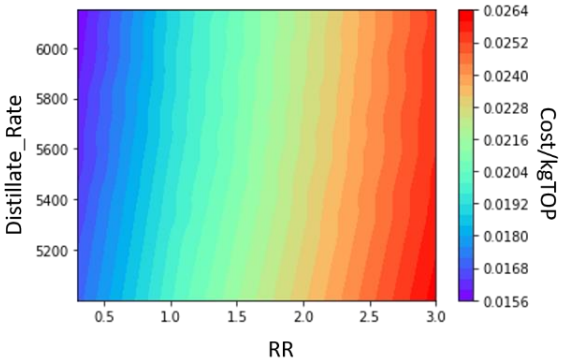


*Figure 11.10. Heat Map of the model predictions at NT = 8*

## 11.3. Model 2

Model 2 aims to find whether modelling with just part of the data, which has been previously filtered, would lead to different results in terms of accuracy, model complexity and computational time. Thus, the same dataset is used, though a previous filtering is done before the shuffling, normalization and separation of the data.

**Data filtering**

Table 11 and Table 12 summarize the shape of the dataset before and after the filtering is implemented. Only 6.355 samples remain from the previous 10.000. From Table 11 it can be noticed how the first three samples have the same purity (TOP_Fraction_A) and recovery, despite having different costs. Table 12 only keeps the lowest cost sample for each range of purity and recovery.

*Table 11. Dataset before the filtering. 10000 samples.*

| NT | RR | Distillate_Rate | HK_top | LK_bottom | TOP_Fraction_A | Recovery | Cost/kgTOP |
|---|---|---|---|---|---|---|---|
| 18 | 2.654 | 6136.613 | 0.0 | 0.002 | **1.0** | **0.998** | **0.0247** |
| 18 | 2.904 | 6133.669 | 0.0 | 0.002 | **1.0** | **0.998** | **0.0255** |
| 17 | 2.976 | 6134.568 | 0.0 | 0.002 | **1.0** | **0.998** | **0.0257** |
| 17 | 2.707 | 6128.726 | 0.0 | 0.003 | 1.0 | 0.997 | 0.0249 |
| 17 | 2.693 | 6127.136 | 0.0 | 0.003 | 1.0 | 0.997 | 0.0248 |

*Table 12. Dataset after the filtering. 6335 samples.*

| NT | RR | Distillate_Rate | HK_top | LK_bottom | TOP_Fraction_A | Recovery | Cost/kgTOP |
|---|---|---|---|---|---|---|---|
| 18 | 2.654 | 6136.613 | 0.0 | 0.002 | **1.0** | **0.998** | **0.0247** |
| 17 | 2.693 | 6127.136 | 0.0 | 0.003 | 1.0 | 0.997 | 0.0248 |
| 17 | 2.282 | 6119.888 | 0.0 | 0.005 | 1.0 | 0.996 | 0.0236 |
| 18 | 2.233 | 6117.945 | 0.0 | 0.005 | 1.0 | 0.995 | 0.0235 |
| 19 | 2.165 | 6109.282 | 0.0 | 0.006 | 1.0 | 0.994 | 0.0233 |

Figure 11.11 is the updated version of Figure 11.4, with the filtered data instead of the entire dataset. Even though the behaviour is similar, it can be seen how most of the samples with high RR but low Distillate Rate are discarded, as these points lead to high costs without good recoveries. However, the similar pattern suggests that both models may not have remarkable differences.

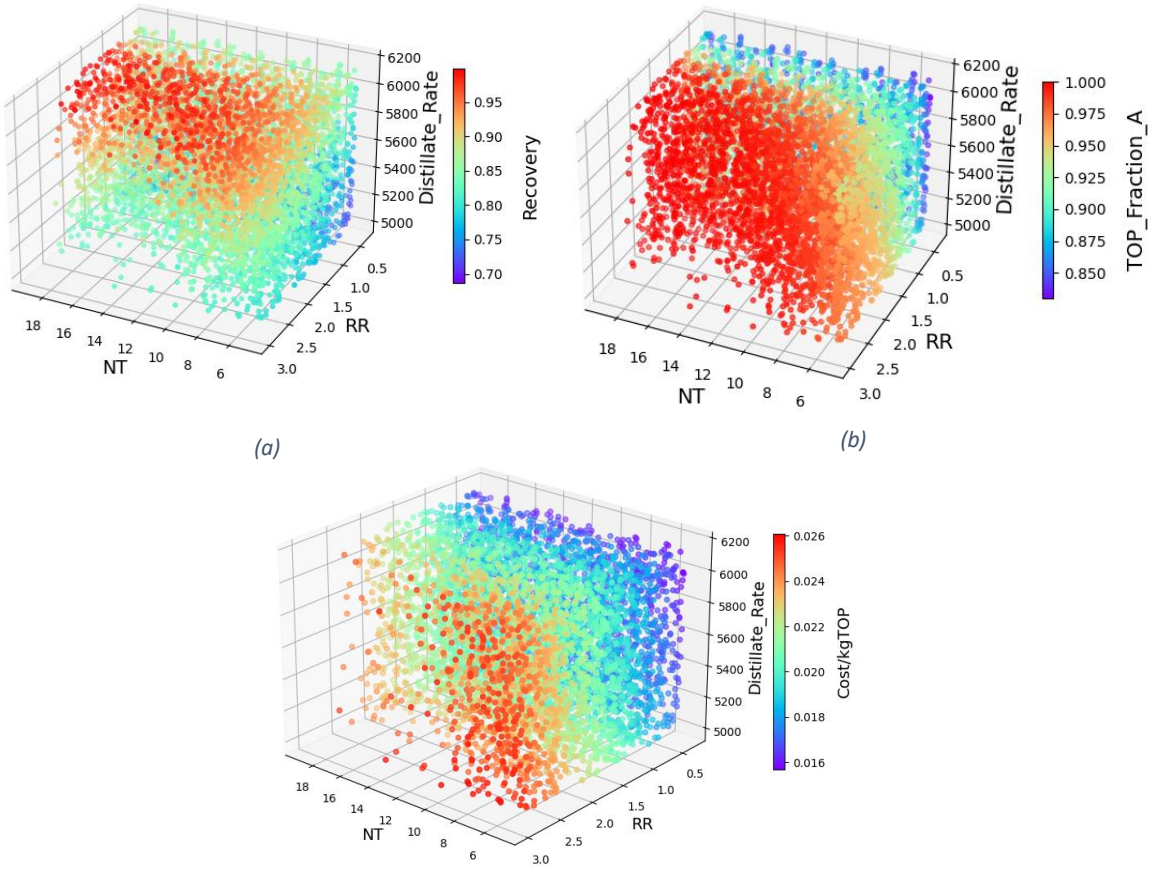


*Figure 11.11. Heat Maps of the filtered sampling (6.355 samples).* a: Recovery, b: TOP_Fraction_A, c: Cost/kgTOP

### Comparison with the shortcut

Before training the model, results have been compared again with the shortcut model, by simulating those samples above some purity and recovery conditions in it. To define the shortcut, the HK in TOP, the LK in BOTTOM, and the RR variables are taken from the sample with lowest cost that satisfies the minimum requirements of purity and recovery. These requirements have been settled at the 90 %, 95 % and 99 % of both variables. Results from the sampling and the shortcut can be found at Table 13, where blue cells represent the independent variables specified at the shortcut.

*Table 13. Sampling-shortcut comparison at different purity and recovery minimum requirements*

| **Optimum values** | **Recovery & Purity 0.9** | | **Recovery & Purity 0.95** | | **Recovery & Purity 0.99** | |
|---|---|---|---|---|---|---|
| | **Samples** | **Shortcut** | **Samples** | **Shortcut** | **Samples** | **Shortcut** |
| **NT** | 16 | 9 | 15 | 12 | 19 | 17 |
| **RR** | 0.597 | | 0.919 | | 1.192 | |
| **Distillate rate (kg/h)** | 6124 | 6094 | 6126 | 6112 | 6136 | 6135 |
| **TOP fraction A (mass)** | 0.902 | | 0.956 | | 0.991 | |
| **Recovery** | 0.900 | | 0.953 | | 0.990 | |
| **Cost ($/kgTOP)** | 0.0174 | 0.0173 | 0.0189 | 0.0189 | 0.0202 | 0.0202 |
| **LK in BOTTOM** | 0.107 | | 0.050 | | 0.011 | |
| **HK in TOP** | 0.065 | | 0.029 | | 0.006 | |

The shortcut cost estimations closely match the expected values from the sampling. However, the required distillate rates and NT to satisfy the RR established are not that similar. As expected, shortcut simplifications lead to good results but cannot be used to accurately estimate the behaviour of a real column. For instance, when the shortcut points are sought on the filtered data, no matches are found, meaning that those variables do not lead to minimum cost designs.

Data is finally shuffled, normalized, and split into a training and testing set like in the simulation model.

### Fitting the model

The model characteristics are summarized in Table 14. The computational time required to train and validate the model is 16,1 seconds.

*Table 14. ANN optimization model properties*

| | | | |
|---|---|---|---|
| Training set: | 4861 samples | Neurons 1$^{st}$ layer: | 3 |
| Validation set: | 858 samples | Neurons 2$^{nd}$ layer: | 2 |
| Testing set: | 636 samples | Epochs: | 75 |

Figure 11.12 depicts the evolution of the error in both training and validation sets. The low values of the MSE in both sets claims the good performance of the model.

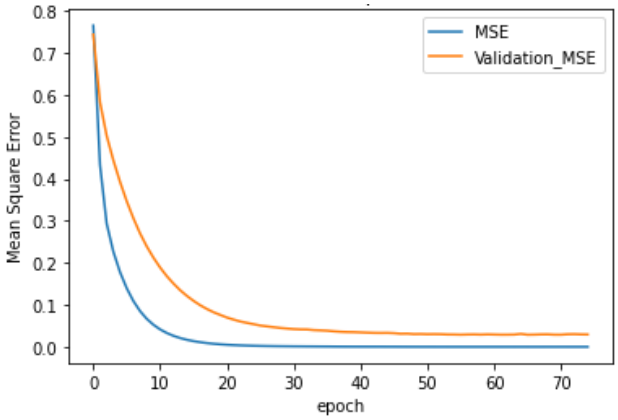


*Figure 11.12. Evolution of the MSE in the training and validation set with respect the epochs for the optimization model*

The representation of the residuals for the model also emphasizes the lack of overfitting. The coefficient of determination ($R^2$) for Figure 11.13 is 0.9993.

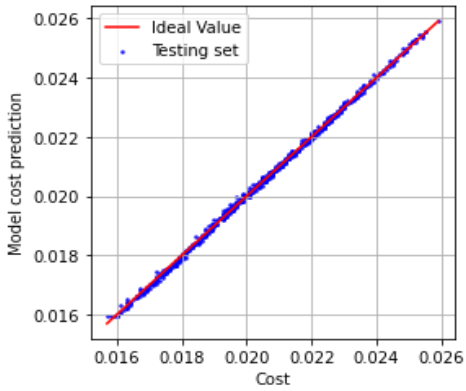


*Figure 11.13. Model predictions of the cost compared to the sampling values of it*

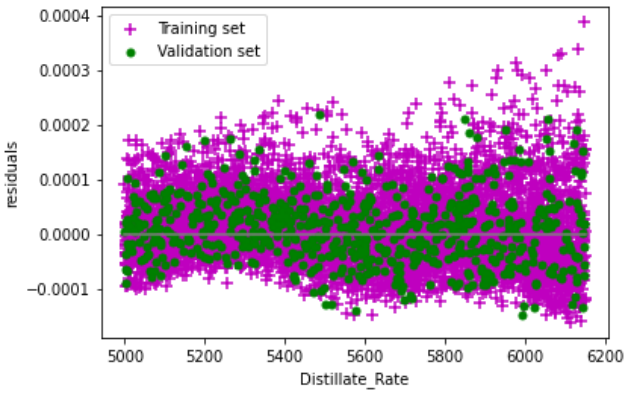


*Figure 11.14. Residuals of the training and testing set with respect the RR*

## 11.4. Model's comparison.

A first comparison of the two models can be found at Table 15. The training of both models shows no patterns on the residuals, good coefficients of determination and low losses. The complexity of the Model 1, and thus the computational time required to train it, is bigger, as more data has been given to it.

Figure 11.15 and Figure 11.16 depict the Heat Map for the predictions of both models for a new set of points, barely showing any difference between them.

*Table 15. Models comparison with regard of the ANN characteristics*

| | **Model 1** | **Model 2** |
|---|---|---|
| **Layers & Neurons** | 6 – 3 | 3 – 2 |
| **Data points** | 10000 | 6335 |
| **Fitting Computational Time** | 21.1 seconds | 16.1 seconds |
| **MSE** | 0.0004 | 0.0008 |
| **R2** | 0.9996 | 0.9993 |
| **Pattern in the residuals** | No | No |

Therefore, both predictions are plotted in a unique graphic in Figure 11.17, where the distance to the red line indicates how different the predictions between Model 1 and Model 2 are. The $R^2$ obtained is 0.9958, indicating the similarity in the models prediction. In Section 3 it is claimed that if both models describe the same surface, the model with less complexity should be chosen among the others. Consequently, Model 2 should get over Model 1, though they are not as different in terms of complexity to clearly discard Model 1.

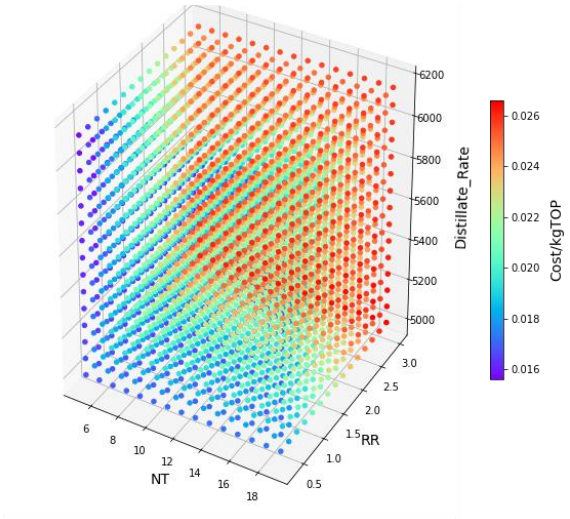


*Figure 11.15. Heat Map of Model 1*

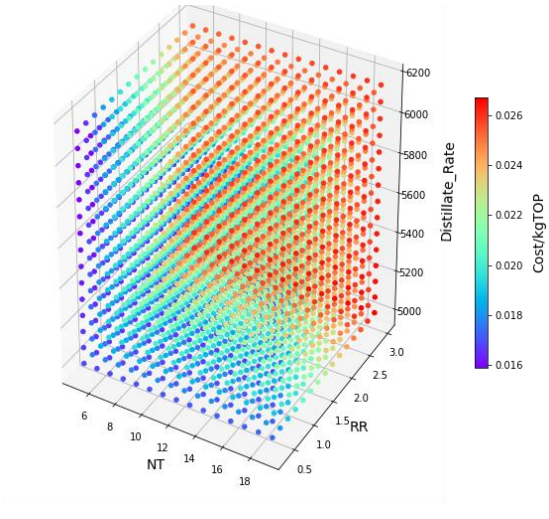


*Figure 11.16. Heat Map of Model 2*

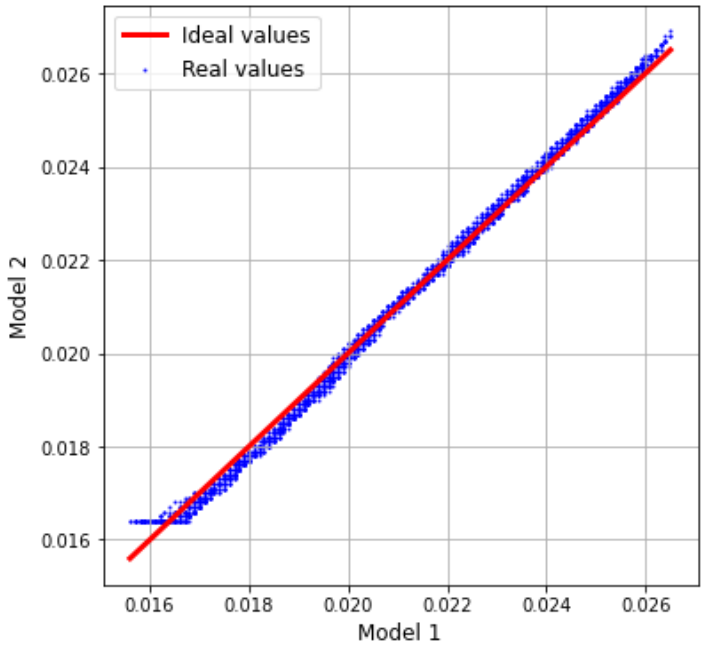


*Figure 11.17. Model 2 predictions of the cost compared to Model 1 predictions of it*

To conclude, not only do both models report similar predictions, but also similar levels of complexity. Consequently, the filtering step that differentiates both models has little effect in the training and validation of the models.

Once the model is trained, any point from the studied surface can be predicted instantaneously, facilitating the application of optimization tools. Therefore, optimizations can be carried out in external environments within minutes and a good level of accuracy, as each point of the desired surface is rapidly predicted. Moreover, the costs function can be easily updated or changed to a different one, considering other variables, so the same dataset can be used with different purposes.

Further studies could consider studying the dependence of the surrogate's accuracy and complexity with respect the sampling size, which is the step that requires more computational time. Instead of simulating the process, future surrogates could focus on optimizing it, by predicting the costs and design and operation conditions from a given purity and recovery.

# 12. Environmental impact

Environmental impact analysis is nowadays a key aspect in every scientific project, as satisfying 0environmental impact targets has become a must in all technologies under development. Even more in those related to information technology (IT) and engineering, for their positive impact on new technologies, products or processes.

Despite the project has been completely done digitally, there are several aspects that should be acknowledged when considering its environmental impact. For instance, the manufacturing of the assets utilized. A list of the possible damages generated is given below:

- The hardware manufacturing directly generates waste, which can be reduced by a correct recycle of the assets at the time that become outdated.
- The transport and manufacturing of the assets leads to carbon emissions and the generation of greenhouse gases, which increase the global warming.
- Manufacturing these devices usually requires factories, whose installation may disrupt ecosystems by allowing pollution to affect its natural cycles.
- The use of non-renewable resources such as metals or the coal used to produce electricity.
- Hazardous materials for the environment that are difficult to treat for their further disposal.

On the other side, this project does also aim to improve the environment. In the “Project origin” section it has been explained the sustainable background of the project, which seeks to optimize distillation trains by considering the reuse of secondary products as selling components or fuel. The fact of conceiving secondary products, sometimes known as waste, as potential resources, reduces the consumption of raw materials and thus, their environmental impact.

# Conclusions

Once the proposed methodology has been implemented in a case study and the discussion of the results has taken place, the following conclusions with regard the initial objectives are listed below.

- The state of the art research for the application of surrogate models in distillation trains not only has led to the election of ANN's to substitute rigorous distillation column models, but also it has been useful to determine which input-output variables should be considered in the process.
- The implementation of the general methodology in the proposed case study enables the training and validation of two surrogate models with the same purpose: the simulation of the behaviour of the column.
- The required tools to implement the methodology have been designed in Python so they can be later used in external environments. The Python-HYSYS connection enables the user to extract the desired data from the Aspen HYSYS simulation to an Excel file, where it can be easily manipulated.
- The Python code eases the manipulation of the objective function, as well as other required pre-treatments of the data, allowing the implementation of the tools in several scenarios and users.
- The previous study of the distillation process in shortcuts and columns in Aspen HYSYS has led to a first approximation of the desired variable boundaries. Other comparisons such as the FUG simplified model or Heuristics gave more details about the studied process.
- The order of the samples before its sampling is crucial to minimize the computational time of the sampling as well as the amount of unfeasible designs. The number of trays (NT) has been claimed as the most influential variable with respect the sampling, thus the whole sampling is ordered from higher to lower NT.
- The analysis of the residuals, the evolution of the error, and the comparisons with the initial data show the good performances of both final models at predicting the cost for new configurations. The similar model complexity results in close predictions in both models, claiming that the datasets used are not different enough.
- The ease of the model to instantaneously predict new data points when it is trained helps the user to carry out optimizations in external environments without having to simulate each point in rigorous distillation models.

On the basis of the conclusions taken above and the results discussion, some recommendations and future work are proposed below.

- To implement the proposed methodology to all the columns of the process and run an external optimization with all the surrogates.
- To increase the complexity of the objective function to avoid simplifications such as the constant tray diameter, the lack of flooding or the constant temperature increments in the exchangers. Other reasonable non-economic objectives such as safety or environmental impact could also be considered.
- To take into consideration other thermodynamic tools such as pinch analysis and energy integration to determine the column pressures and the heat exchangers networks, so the amount of heat required can be reduced.
- To explore other types of surrogate models where to apply the proposed methodology. Other configurations of ANN may also be explored seeking less-complex models.
- To evaluate if the accuracy of the model changes as a function of the data samples sizes.
- To study a surrogate model that optimizes the column instead of simulating its behaviour. This model could consider as input variables the recovery and purity of the outlet stream, while the costs, and design and operation conditions would be the outputs of the model.
- To implement a logarithmic filtering of the data, so more points at high purity and recovery are taken into consideration.

# Economic evaluation

The present section consists of the economic analysis of the project, including the costs associated to the project, and the discussion of potential profits from it. The costs that have been taken into account are: the hardware, the software, and personnel costs.

First, the hardware costs are summarized in the table below (Table 16).

*Table 16. Economic evaluation. Hardware costs.*

| **Hardware** | |
|---|---|
| Computer and peripherals [€] | 65.00 € |
| Total [€] | **65.00 €** |

The hardware costs consider the physical material used in the project, which consists in a computer and its peripherals, namely the screen, the keyboard and the mouse. The final value of 65 € is computed from the amortization of the equipment, described by Equation (*24*). A purchase price of 1000€, with a lifespan of 5 years, and a salvage value of 350 €, lead to the amortization per year, which has been multiplied by the duration of the project, approximately half a year. The usage factor for the hardware used is 1.

$$\text{Amortization}\left(\frac{€}{\text{year}}\right) = \frac{Purchase\ price\ (€) - Salvage\ value\ (€)}{Lifespan\ (years)} \tag{24}$$

Second, the software costs are summarized in the table below (Table 17).

*Table 17. Economic evaluation. Software costs.*

| **Software** | |
|---|---|
| Microsoft Office Professional 2016 [€] | 149.00€ |
| ASPEN-HYSYS v11 [€] | 4306.81 € |
| Total [€] | **4455.81 €** |

The software costs include the licenses and programs used during the project. As Python is an open source programming language, it does not represent cost to their users.

On the other side, the writing editor license cost has been obtained from the official Microsoft Store (Microsoft, 2021).

With respect to ASPEN-HYSYS®, just an approximate cost of 20000-50000 $/year has been found. Thus, the average value in euros has been considered. The conversion of 1.219 $/€ has been obtained from

the newspaper “El Economista”(ElEconomista, 2021), on 27/05/21. Additionally, a usage factor of 0.14 has been applied to the ASPEN-HYSYS, as it has been shared by 7 people.

Third, the personnel costs are described in the table below (Table 18).

*Table 18. Economic evaluation. Personnel costs.*

| **Personnel** | |
|---|---|
| Process Engineer [€] | 5156.32 € |
| Social Security (23.6 %) [€] | 1216.89 € |
| Total [€] | **6373.21 €** |

The personnel costs, which consider the cash compensations and taxes of the employees, has been assumed as if the project was developed by a process engineer with no working experience. Thus, the annual gross salary for this position is taken as 20000€. Considering 52 weeks per year and 40 hours per week the resulting salary per hour is 9.62 €/h. The total dedication in hours for this project is computed considering an average work of 5 h per day, starting the 16 of January of 2021 and finishing the 16 of June of 2021. Thus, the total time invested on the project is 536 h. Then, the total cost designated to the process engineer is computed as a product of these values. Additionally, the social security percentage of 23.6 % has been stablished according to Spanish Government (España, 2021).

Finally, an overhead of 10 % of the total cost is added to consider intangible assets such as electricity, maintenance, internet connection, or literature. The total costs computation is defined in 1.

*Table 19. Economic evaluation. Total costs computation.*

| **Total cost** | |
|---|---|
| Hardware [€] | 65.00 € |
| Software [€] | 4455.81 € |
| Personnel [€] | 6373.21 € |
| Total without overhead [€] | 10894.02 € |
| Overhead (10 %) [€] | 1089.40 € |
| Total with overhead [€] | **11983.42 €** |

Figure 0.1 Figure 0.1depicts the costs distribution of the project. The most remarkable input are the personnel costs, being 58.5 % of the total cost. In the same order of magnitude there are the software costs, representing 40.9 % of the total cost, while hardware expenses are negligible.

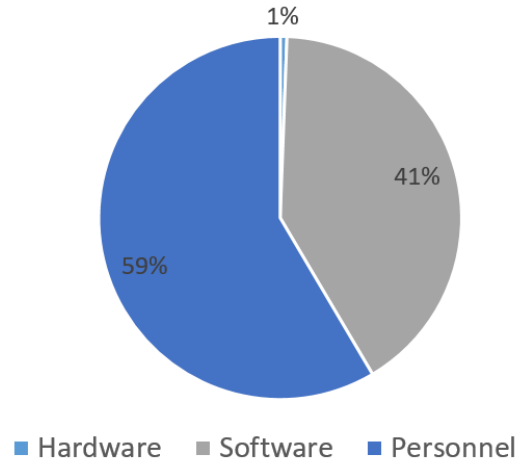


*Figure 0.1. Economic evaluation. Total cost distribution (%).*

# Bibliography

Bhosekar, A., & Ierapetritou, M. (2018). Advances in surrogate based modeling, feasibility analysis, and optimization: A review. *Computers and Chemical Engineering*, *108*, 250–267. https://doi.org/10.1016/j.compchemeng.2017.09.017

Brownlee, J. (2021). *Maching learning mastery*. Machinglearningmastery.Com. https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/#:~:text=for Output Layers-,Activation Functions,a layer of the network.

Canyon Hydro, Summary, E., Of, F., Potential, T. H. E., Ferreres, X. R., Font, A. R., Ibrahim, A., Maximilien, N., Lumbroso, D., Hurford, A., Winpenny, J., Wade, S., Sataloff, R. T., Johns, M. M., Kost, K. M., State-of-the-art, T., Motivation, T., Norsuzila Ya'acob1, Mardina Abdullah1, 2 and Mahamod Ismail1, 2, Medina, M., … Masuelli, M. (2013). We are IntechOpen , the world ' s leading publisher of Open Access books Built by scientists , for scientists TOP 1 %. *Intech*, *32*(July), 137–144. http://www.intechopen.com/books/trends-in-telecommunications-technologies/gps-total-electron-content-tec- prediction-at-ionosphere-layer-over-the-equatorial-region%0AInTec%0Ahttp://www.asociatiamhc.ro/wp-content/uploads/2013/11/Guide-to-Hydropower.pdf

Carnell, R. (2020). *The Comprehensive R Archive Network*. Basic Latin Hypercube Samples and Designs with Package Lhs.

Dua, V. (2010). A mixed-integer programming approach for optimal configuration of artificial neural networks. *Chemical Engineering Research and Design*, *88*(1), 55–60. https://doi.org/10.1016/j.cherd.2009.06.007

ElEconomista. (2021). *ElEconomista*. Eleconomista.Com. https://www.eleconomista.es/cruce/EURUSD

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211. https://doi.org/10.1016/0364-0213(90)90002-E

España, G. de. (2021). *Seguridad Social*. Seg-Social.Es. https://www.seg-social.es/wps/portal/wss/internet/Trabajadores/CotizacionRecaudacionTrabajadores/36537

Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *Annals of Statistics*, *36*(3), 1171–1220. https://doi.org/10.1214/009053607000000677

Hong, M., & Chen, E. Y. X. (2017). Chemically recyclable polymers: A circular economy approach to sustainability. *Green Chemistry*, *19*(16), 3692–3706. https://doi.org/10.1039/c7gc01496a

Honus, S., Kumagai, S., Němček, O., & Yoshioka, T. (2016). Replacing conventional fuels in USA, Europe, and UK with plastic pyrolysis gases – Part I: Experiments and graphical interchangeability methods. *Energy Conversion and Management*, *126*, 1118–1127. https://doi.org/10.1016/j.enconman.2016.08.055

Howley, T., & Madden, M. G. (2006). *An Evolutionary Approach to Automatic Kernel Construction*. 417–426.

Ibrahim, D., Jobson, M., Li, J., & Guillén-Gosálbez, G. (2018). Optimization-based design of crude oil distillation units using surrogate column models and a support vector machine. *Chemical Engineering Research and Design*, *134*, 212–225. https://doi.org/10.1016/j.cherd.2018.03.006

Kannan, P., Al Shoaibi, A., & Srinivasakannan, C. (2014). Temperature effects on the yield of gaseous olefins from waste polyethylene via flash pyrolysis. *Energy and Fuels*, *28*(5), 3363–3366. https://doi.org/10.1021/ef500516n

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). A B7CEDGF HIB7PRQTSUDGQICWVYX HIB edCdSISIXvg5r ` CdQTw XvefCdS. *Proc. OF THE IEEE*. http://ieeexplore.ieee.org/document/726791/#full-text-section

Leijnen, S. (2016). *The Asimov Institute*. Asimovinsitute.Org. https://www.asimovinstitute.org/neural-network-zoo/

Medium. (2018). *Towards Data Science*. Towards Data Science. https://towardsdatascience.com/regularization-the-path-to-bias-variance-trade-off-b7a7088b4577

*Metamodels for simulation input-output relations.pdf*. (n.d.).

Microsoft. (2021). *Microsoft Official Store*. Microsoft.Com. https://www.microsoft.com/es-ES/microsoft-365/buy/compare-all-microsoft-365-products

Quirante, N., Javaloyes, J., & Caballero, J. A. (2015). Rigorous design of distillation columns using surrogate models based on Kriging interpolation. *AIChE Journal*, *61*(7), 2169–2187. https://doi.org/10.1002/aic.14798

Robertshaw, T. (2015). *Introduction to Machine Learning with Naive Bayes*. Tom Robertshaw. https://tomrobertshaw.net/2015/12/introduction-to-machine-learning-with-naive-bayes/

Sayad, D. S. (2010). *Support Vector Machine - Regression (SVR)*. Saedsayad.Com. https://www.saedsayad.com/support_vector_machine_reg.htm

Schäfer, P., Caspari, A., Kleinhans, K., Mhamdi, A., & Mitsos, A. (2019). Reduced dynamic modeling approach for rectification columns based on compartmentalization and artificial neural networks. *AIChE Journal*, *65*(5). https://doi.org/10.1002/aic.16568

Seider, W. D., Seader, J. ., Lewin, D. R., & Widagdo, S. (2010). Product and Process Design Principles Synthesis Analysis and Design Third Edition. In *John Wiley & Son, Inc.* (Vol. 91, Issue 2). https://www.academia.edu/16568227/208468464-Product-and-Process-Design-Principles-Synthesis-Analysis-and-Design-Third-Edition_1_

Sheikholeslami, R., & Razavi, S. (2017). Progressive Latin Hypercube Sampling: An efficient approach for robust sampling-based analysis of environmental models. *Environmental Modelling and Software*, *93*, 109–126. https://doi.org/10.1016/j.envsoft.2017.03.010

Sinnott, R., & Towler, G. (2020). Costing and Project Evaluation. In *Chemical Engineering Design* (Issue 2019). https://doi.org/10.1016/b978-0-08-102599-4.00006-0

Somoza-Tornos, A., Chen, Q., Graells, M., Espuña, A., & Grossmann, I. E. (2020). Modeling Framework for Joint Product and Process Synthesis with Material Recovery Opportunities. *Computer Aided Chemical Engineering*, *48*, 823–828. https://doi.org/10.1016/B978-0-12-823377-1.50138-5

Souza, C. (2010). *Kernel Support Vector Machines for Classification and Regression in C#*. Crsouza.Com. http://crsouza.com/2010/04/27/kernel-support-vector-machines-for-classification-and-regression-in-c/

Swain, J. J., Goldsman, D., Crain, C., & Barton, R. R. (1992). *for Simulation*. *9*(x), 49–54.

Ulrich, G. D., & Vasudevan, P. T. (2006). How to estimate utility costs. *Chemical Engineering*, *113*(4), 66–69.

Wermac. (2021). *Wermac Distillation Columns*. WERMAC. http://www.wermac.org/equipment/distillation_part1.html

Xie, Q., Liu, H., Bo, D., He, C., & Pan, M. (2018). Data-driven Modeling and Optimization of Complex Chemical Processes Using a Novel HDMR Methodology. *Computer Aided Chemical Engineering*, *44*(January), 835–840. https://doi.org/10.1016/B978-0-444-64241-7.50134-8

# Appendixes

The appendixes are uploaded in the linked GitHub folder. They contain, the Python code for the several tools described in the project, the data sets and the ANN for both models and the Python code for the Heat Map plots.

https://github.com/arnaumm3/TFG