

Felipe Souza Lima

***Metacontrol*: a Python based software for
self-optimizing control structure selection using
metamodels**

Campina Grande, Paraíba, Brasil

Abril, 2020

Felipe Souza Lima

***Metacontrol: a Python based software for self-optimizing
control structure selection using metamodels***

Dissertation submitted to the Graduate Program in Chemical Engineering of the Federal University of Campina Grande in fulfillment for the degree Master of Chemical Engineering. Area: Chemical Engineering.

Universidade Federal de Campina Grande

Unidade Acadêmica de Engenharia Química

Programa de Pós-Graduação em Engenharia Química

Supervisor: Dr. Antônio Carlos Brandão de Araújo

Campina Grande, Paraíba, Brasil

Abril, 2020

L732m

Lima, Felipe Souza.

Metacontrol: a Python based software for self-optimizing control structure selection using metamodels / Felipe Souza Lima. – Campina Grande, 2020.

111 f. : il. color.

Dissertação (Mestrado em Engenharia Química) – Universidade Federal de Campina Grande, Centro de Ciências e Tecnologia, 2020.

"Orientação: Prof. Dr. Antônio Carlos Brandão Araújo".

Referências.

1. Engenharia Química. 2. Modelagem e Simulação. 3. Python. 4. Self-Optimizing Control. 5. Kriging. 6. Software. 7. Plantwide. I. Araújo, Antônio Carlos Brandão. II. Título.

CDU 66.01(043)

Felipe Souza Lima

Metacontrol: a Python based software for self-optimizing control structure selection using metamodels

Dissertation submitted to the Graduate Program in Chemical Engineering of the Federal University of Campina Grande in fulfillment for the degree Master of Chemical Engineering. Area: Chemical Engineering.

Trabalho aprovado. Campina Grande, Paraíba, Brasil, 27 de Abril de 2020:



Orientador

Dr. Antônio Carlos Brandão de Araújo



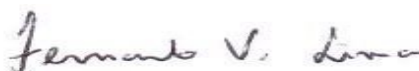
Professor

Dr. Heleno Bispo da Silva Júnior



Professor

Dr. Sidinei Kleber Silva



Professor

Dr. Fernando Vines de Lima

Campina Grande, Paraíba, Brasil
Abril, 2020

For those who doubt themselves. Face and embrace it. It is part of what defines you. Use the doubt as a weapon and you will be surprised of how far you can go.

Acknowledgements

To my mother Lígia, my brother Gustavo and my father Enildo. You are the why I remember things are going to be okay in my best and direst moments.

To Victor and Paloma, for the moments, either good or bad. I will remember and cherish them all. Seriously, thank you.

To my supervisor Brandão, for the ideias and guidance.

To Rodolpho, Ranny and Claudemi, for being awesome friends I have met during this journey.

*“To love the journey is to accept no such end.
I have found, through painful experience, that the most important step a person can take is
always the next one.
(Dalinar Kholin, Oathbringer.)*

Resumo

Neste trabalho é apresentado uma ferramenta computacional baseada em Python que permite uma rápida implementação da metodologia de Controle Auto-Otimizante (do inglês Self-Optimizing Control) com o auxílio de modelos surrogados. A dissertação mostra as possibilidades e o potencial do *software Metacontrol* através de estudos de caso de processos já estabelecidos na indústria. Como resultado, obteve-se uma análise profunda destes processos por uma perspectiva de controle em escala de planta (do inglês *plantwide*), e também são discutidas recomendações de uso da ferramenta. Além disso, os dados, exemplos e o código fonte do *software Metacontrol* estão disponíveis no link <https://github.com/feslima/metacontrol>.

Palavras-chave: Python, Self-Optimizing Control, *Kriging*, Software, Plantwide.

Abstract

In this work, it is presented a Python based software tool that enables fast implementation of a Self-Optimizing Control methodology with the help of surrogate models. The dissertation outlines the potential uses of the *Metacontrol* software through cases studies of well established industrial processes. As a result, an in-depth analysis from a plantwide perspective of these processes is discussed, along with recommendations of use. Furthermore, the data, examples and the *Metacontrol* source code shown here are available to download at <https://github.com/feslima/metacontrol>.

Keywords: Python, Self-Optimizing Control, *Kriging*, Software, Plantwide.

List of Figures

Figure 1 – Initial plot of our complex model. The solid blue line represents the function behavior. The dashed line is the <i>Kriging</i> metamodel of the three sampled points (red circles) available.	42
Figure 2 – The <i>Kriging</i> model after one update.	43
Figure 3 – The <i>Kriging</i> model after four updates. Notice how the <i>Kriging</i> model adjusts to the true function.	44
Figure 4 – Flowchart of Caballero and Grossmann (2008) algorithm, translated to Python by the author of this work and implemented within <i>Metacontrol</i>	45
Figure 5 – Flowchart describing how <i>Metacontrol</i> works.	51
Figure 6 – CPU process flowsheet	54
Figure 7 – <i>Metacontrol</i> main screen with CPU process simulation file loaded.	57
Figure 8 – Loading variables for the CPU from Aspen Plus simulation and adding alias to them. At the top right corner of this screen, the user is able to select the option to reveal the GUI from Aspen Plus. This features allows the user to inspect inside the process simulator interface to remember any stream or block names. This can be helpful when one is selecting the variables using the COM technology and there are several unit operations blocks and streams, for instance. Another feature that was implemented in order to ease the search of the variables, regards the description of each variable: Hovering the mouse over a COM variable will show its description, extracted directly from the process simulator.	58
Figure 9 – Creating expressions for specific power consumption (objective function), CO ₂ recovery rate and S8 Temperature (constraint functions/CV candidates).	59
Figure 10 – <i>Metacontrol</i> “Sampling panel”. The user can perform the sampling using the process simulator or import a .CSV file.	60
Figure 11 – <i>Metacontrol</i> Sampling assistant. The limits for the decision variables used in the CPU process are the same from Jin, Zhao, and Zheng (2015) and Liu et al. (2019)	61
Figure 12 – <i>Metacontrol</i> Latin Hypercube Sampling settings. 80 samples were generated and 5 iterations were performed in order to try to maximize the minimum distance between the points (<i>maxmin</i> criterion). The user can also add the vertices of the design of experiments.	62
Figure 13 – <i>Metacontrol</i> Sampling for the CPU process.	63
Figure 14 – Sampling results, where the user can inspect convergence status and the values of the selected variables for each case.	64

Figure 15 – <i>Kriging</i> configuration and validation metrics results.	65
Figure 16 – Fitness for each metamodel.	66
Figure 17 – Refinement algorithm configuration and results screen.	67
Figure 18 – Refinement algorithm control panel output.	67
Figure 19 – “Variable activity” panel, where the user is capable of highlighting which variables are active constraints and inputting values for them. If an active constraint is a nonlinear constraint, the user must pair this variable with a decision variable (MV) to consume a degree of freedom.	68
Figure 20 – Loading a *.csv file in containing design of experiments data in <i>Metacontrol</i> : if the user chooses this option, he must provide a file containing all variables selected from the first step (“Load variables” under “Load simulation” tab). the convergence flag is used as a header to map the *.csv, and the software asks the user to select it.	68
Figure 21 – Associating each alias created in <i>Metacontrol</i> to each column of the *.csv data.	69
Figure 22 – “Differential data” input screen: Reduced space model training, Differentiation method, and gradient/hessian evaluation.	69
Figure 23 – Generating reduced space metamodel for CPU process: to avoid redundancy, the variables “f1tout”, “f2tout”, and “mcctout” were not chosen in the reduced space problem since they correspond to the decision variables that were found as active constraints. In general, if the user decides to remove any variable previously set in the problem, he must uncheck the undesired variable.	70
Figure 24 – Differential data estimated in <i>Metacontrol</i>	71
Figure 25 – Input screen in <i>Metacontrol</i> "Self-Optimizing Control " tab - CPU Process	72
Figure 26 – Best control structure in worst-case loss ascending order, for subsets of size 1 (single measurement policy)	73
Figure 27 – Best control structure in worst-case loss ascending order, for subsets of size 2 (linear combinations of measurements)	74
Figure 28 – C3 Splitter Column Process flowsheet	76
Figure 29 – C3 Splitter Column - Temperature profile.	77
Figure 30 – C3 Splitter Column Process - loading simulation. Process constraint “c2” is multiplied by 4.184 to convert simulation reboiler duty from $GCal/h$ to GJ/h	79
Figure 31 – C3 Splitter Column Process - loading variables from Aspen Plus.	80
Figure 32 – K-fold validation metric for constant (poly0) regression model.	81
Figure 33 – K-fold validation metric for linear (poly1) regression model.	82
Figure 34 – K-fold validation metric for quadratic (poly2) regression model.	83

Figure 35 – Indirect control index objective function being minimized using surrogate metamodel.	84
Figure 36 – Reduced space problem - sampling using a *.bkp file.	85
Figure 37 – Reduced space problem - pointing the *.bkp file location.	86
Figure 38 – Reduced space problem - Sampling assistant: Identical to the Sampling Assistant that exists under the “Sampling” tab, in order to keep consistency of interface across <i>Metacontrol</i> . Number 1 indicates the button to open the Assistant, 2 consists in the main screen, 3 is the button that opens the settings of the sampling technique that will generate the input data; 4 generates the data. In addition, number 5 depicts the control of the sampling procedure: Sample data, cancel, close screen (“Done” button). Lastly, the user can abort the sampling at any time using the “Abort” button (number 6) or export the design of experiments as a *.csv (number 7).	87
Figure 39 – “Differential Data” tab - C3 Splitter indirect control.	88
Figure 40 – Input screen in <i>Metacontrol</i> “Self-Optimizing Control” tab - C3 Splitter column: Here, all 190 possible control structures for a single measurement policy were considered to be evaluated by <i>Metacontrol</i> . For linear combinations of measurements as CV candidates, the 50 best ones of each possible subset size were evaluated, when possible. For subset sizes of 19 and 20, all combinations were considered (20 and 1, respectively).	91
Figure 41 – Best control structures for single measurement policy: Stages with significant temperature deviation between them associated with flow and flow ratios - namely boilup, reflux, boilup to feed ratio and reflux to feed ratio.	92
Figure 42 – Worst control structures for single measurement policy using exclusively temperature measurements: Stages with small temperature deviation between them. One can easily note that the inspection of the best and worst control structures is simple in <i>Metacontrol</i> : The user is capable of sorting, using the graphical user interface built, the control structures in ascending or descending order of worst-case loss, average-case loss and conditional number.	92
Figure 43 – Best control structures using linear combinations of measurements as CV candidates - Subset of size 3.	93
Figure 44 – Best control structures using linear combinations of measurements as CV candidates - Subset of size 6.	93
Figure 45 – Best control structures using linear combinations of measurements as CV candidates - Subset of size 9.	93

Figure 46 – Best control structures using linear combinations of measurements as CV candidates - All available measurements.	93
Figure 47 – C4 Isomerization process flowsheet.	95
Figure 48 – C4 Isomerization process - loading simulation. The cooling water price is positive due to signal convention inside the process simulator - heat removed from the system has a negative sign.	96
Figure 49 – C4 Isomerization process - loading variables.	97
Figure 50 – C4 Isomerization process - loading variables.	98
Figure 51 – C4 Isomerization process - High-order data obtainment.	98
Figure 52 – C4 Isomerization process - Self-Optimizing Control input.	99
Figure 53 – C4 Isomerization process - Single measurements policy: Best CV candi- dates.	99
Figure 54 – C4 Isomerization process - Single measurements policy: Best CV candi- dates not using compositions.	99
Figure 55 – ψ function value - named “perf” within <i>Metacontrol</i>	104

List of Tables

Table 1 – CV Candidates for CO_2 CPU process.	55
Table 2 – Optimization runs: <i>Aspen Plus vs Metacontrol</i> - Decision variables and objective function - CPU Process	60
Table 3 – Optimization runs: <i>Aspen Plus vs Metacontrol</i> - Process constraints - CPU Process	61
Table 4 – High-order data obtainment: <i>Aspen Plus vs Metacontrol</i>	66
Table 5 – Mean-squared error of high-order data obtainment: <i>Aspen Plus vs Meta-</i> <i>control</i> - CPU Process	66
Table 6 – Best Self-Optimizing Control variables found by <i>Metacontrol</i> for a single measurement policy.	69
Table 7 – CV Candidates for C3 Splitter composition indirect control.	78
Table 8 – Optimization runs: <i>Aspen Plus vs Metacontrol</i> - Decision variables and objective function - C3 Splitter Indirect control	79
Table 9 – High-order data obtainment: <i>Aspen Plus vs Metacontrol</i> - C3 Splitter column case study.	89
Table 10 – Mean-squared error of high-order data obtainment: <i>Aspen Plus vs Meta-</i> <i>control</i> - C3 Splitter column	90
Table 11 – C4 Isomerization process optimization summary.	96
Table 12 – CV Candidates for C_4 Isomerization process.	97

List of abbreviations and acronyms

CPU Compression Purification Unit

CV Controlled Variable

DOE Design Of Experiments

DOF Degree of Freedom

GUI Graphical User Interface

IEAGHG International Agency Greenhouse Gas

LHS Latin Hypercube Sampling

MV Manipulated Variable

NLP Non Linear Problem

RTO Real Time Optimization

SOC Self-Optimizing Control

List of symbols

Chapter 2

c	Controlled variable
d	Process disturbances
F	Optimal measurement sensitivity matrix with respect to the disturbances
G^y	Gain matrix with respect to the measurements
G_d^y	Gain matrix with respect to the disturbances
H	Linear combination matrix
J	Process objective function
J_0	Process objective function in unrestricted space
J_{opt}	Optimal value of J
J_{ud}	Hessian of cost function with respect to the disturbance variables $\left(\frac{\partial^2 J}{\partial u \partial d}\right)$
J_{uu}	Hessian of cost function with respect to the manipulated variables $\left(\frac{\partial^2 J}{\partial^2 u}\right)$
L	Loss
n	Implementation error
$n^{y'}$	Implementation error with respect to the measurements
u	Manipulated variable
u_0	Process degrees of freedom
W_d	Diagonal magnitude matrix of disturbances
W_n	Diagonal magnitude matrix of measurement errors
x	Process states
y	Measurements
z	Loss variable

Chapter 3

\hat{y}	Metamodel (approximation) of y
$\mathcal{F}, f(x)$	Polynomial regression function

σ_l^2	Process variance
θ	<i>Kriging</i> hyperparameter of variable activity
ε	Residuals or random noise
p	<i>Kriging</i> hyperparameter of correlation smoothness
x	Input of a process, or sample
y	Function that calculates the output of a process
z	Stochastic departure function

Contents

1	INTRODUCTION	27
1.1	Publications	27
2	THE SELF-OPTIMIZING CONTROL OVERVIEW	29
3	<i>KRIGING</i> REASONING	37
4	NON LINEAR OPTIMIZATION AND INFILL CRITERIA	41
5	THE <i>METACONTROL</i> FRAMEWORK	47
5.1	<i>Metacontrol</i> workflow	49
6	CASE STUDIES APPLIED IN <i>METACONTROL</i>	53
6.1	The CO ₂ Compression and Purification Unit (CPU)	53
6.2	Indirect composition control of a C3 Splitter column	75
6.3	The <i>C</i> ₄ Isomerization Process	94
7	<i>METACONTROL</i> PRACTICE AND RECOMMENDATIONS	101
7.1	The process simulator sampling general good practice: Towards maximizing cases' convergence while keeping precision	101
7.2	The initial sample: Dimensionality and the choice of the degrees of freedom towards the optimization feasible region	102
7.3	The adaptive sampling optimization algorithm parameters	103
7.4	The reduced-space problem sample: Dimensionality and the effect of the <i>Kriging</i> hyperparameters objective function in gradient/hessian evaluation.	104
8	CONCLUSION	105
	References	107

1 Introduction

This dissertation is about an assembly of several methodologies into a software tool, called *Metacontrol*, which enables a fast implementation of the Self-Optimizing Control (SOC) technique. This assembly consist of three major methodologies: *Kriging* metamodels, optimization through infill criteria and SOC. The dissertation is organized as follows:

Chapter 2 gives a brief summary of the key concepts involving SOC methodology and the main the reason why this research and software tool development was needed.

Chapter 3 presents a discussion of *Kriging* metamodels and its reasoning.

Chapter 4 introduces the process of constrained nonlinear optimization using *Kriging* metamodels. This process is also known as infill criteria.

Chapter 5 demonstrates how the assembly of the methodologies shown in chapters 2, 3 and 4 are combined to form the core concept behind *Metacontrol*.

Chapter 6 is dedicated to case-studies using *Metacontrol*. In addition, there is a brief discussion on good practices involving the use of the software tool.

The *Metacontrol* software is publicly available at <https://github.com/feslima/metacontrol>. There, the reader can find instructions on how to install the open-source tool. Also, for each technique discussed in chapters 2, 3 and 4, there is an open-source Python package as result. Their links are found in their respective chapters.

1.1 Publications

As result of this research the following paper was submitted:

Lima, F. S., Alves, V. M. C., Araujo, A. C. B.: *Metacontrol*: a Python based software for self-optimizing control structure selection using metamodels. Submitted for publication to Computers & Chemical Engineering.

2 The Self-Optimizing Control overview

Every industrial process is under limitations ranging from design/safety (e.g. temperature or pressure which an equipment can operate, etc.), environmental (e.g. pollutant emissions), to quality specifications (e.g. product purity), and economic viability. More often than not, these constraints are applied all at once and can be conflicting. Therefore, it is mandatory to operate such processes optimally (or, at least, close to its optimal point) in order to attain maximum profits or keep expenses at minimum while still obeying these specifications.

One way to achieve this is through the application of plantwide control methodologies. In particular, Self-Optimizing Control ([MORARI; STEPHANOPOULOS, 1980](#); [SKOGESTAD, 2000](#); [ALSTAD; SKOGESTAD; HORI, 2009](#)) is a practical way to design a control structure of a process following a criterion (for instance: economic, environmental, performance) considering a constant set-point policy ([ALVES et al., 2018](#)). The SOC methodology is advantageous in this scenario because there is no need to reoptimize the process every time that a disturbance occurs.

However, the review presented here contains merely the paramount elements needed to understand the main concepts and expressions that translate the ideas behind the method. The author recommends them if the reader needs a more detailed explanation ([SKOGESTAD, 2000](#); [HALVORSEN et al., 2003](#); [HORI; SKOGESTAD; ALSTAD, 2005](#); [HORI, Eduardo S.; SKOGESTAD, 2007](#); [ALSTAD; SKOGESTAD; HORI, 2009](#); [ALVES et al., 2018](#); [KARIWALA; CAO; JANARDHANAN, 2008](#); [KARIWALA; CAO, 2009](#); [UMAR et al., 2012](#)).

The main concept of Self-optimizing control consists in the pursue of a control structure that is based on a constant setpoint policy, leading to near-optimal operation. From [Skogestad \(2004\)](#):

“Self-optimizing control is when one can achieve an acceptable loss with constant setpoint values for the controlled variables without the need to reoptimize when disturbances occur.”

It is assumed the process objective function, assumed scalar, is influenced by its steady-state operation. Therefore, the optimization problem described in [Equation 2.1](#) is formed, with u_0 being the degrees of freedom available, x and d representing the states

and the disturbances of the system, respectively.

$$\begin{aligned}
 & \underset{u_0}{\text{minimize}} && J_0(x, u_0, d) \\
 & \text{subject to} && g_1(x, u_0, d) = 0 \\
 & && g_2(x, u_0, d) \leq 0
 \end{aligned} \tag{2.1}$$

Regarding the disturbances, these can be: change in feed conditions, prices of the products and raw materials, specifications (constraints) and/or changes in the model. Using NLP solvers, the objective function can be optimized considering the expected disturbances and implementation errors.

Since the whole technology considers near-optimal operation, as a result of keeping constant setpoints (differently from RTO, for instance), there will always exist a (positive) loss, given by [Equation 2.2](#)

$$L = J_0(d, n) - J_{opt}(d) \tag{2.2}$$

Metacontrol focus on the first four steps of the Self-Optimizing Control technology, named by [Skogestad \(2000\)](#) as “top-down” analysis. In these steps, the variable selection seeking the usage of the steady-state degrees of freedom is the main problem to be addressed with the systematic procedure proposed. It is possible to search for a Self-Optimizing Control structure basically using two methods:

1. Manually testing each CV candidate, reoptimizing the process for different disturbances’ scenarios, and choosing the structure that yields the lowest (worst-case or average-case) loss;
2. Using local methods based on second-order Taylor series expansion of the objective function, that are capable of easily and quickly “pre-screening” the most promising CV candidates.

The manual nature of method 1 and the possibility of creating an automated framework using method 2 motivated the creation of *Metacontrol* itself. Applying, comprehensively, the second method in a software was also a key motivation for this work. Therefore, it is logical that the usage of the linear methods will be discussed in this section, since they are the ones implemented within *Metacontrol*.

A linear model with respect to the plant measurements can be represented as [Equation 2.3](#)

$$\Delta y = G^y \Delta u + G_d^y \Delta d \quad (2.3)$$

With

$$\begin{aligned} \Delta y &= y - y^* \\ \Delta u &= u - u^* \\ \Delta d &= d - d^* \end{aligned} \quad (2.4)$$

G^y and G_d^y are the gain matrices with respect to the measurements and disturbances, respectively. Regarding the CVs, linearization will give [Equation 2.5](#)

$$\Delta c = H \Delta y = G \Delta u + G_d \Delta d \quad (2.5)$$

With

$$\begin{aligned} G &= H G^y \\ G_d &= H G_d^y \end{aligned} \quad (2.6)$$

Linearizing the loss function results in [Equation 2.7](#):

$$\begin{aligned} L &= J(u, d) - J_{opt}(d) = \frac{1}{2} \|z\|_2^2 \\ z &= J_{uu}^{\frac{1}{2}} (u - u_{opt}) = J_{uu}^{\frac{1}{2}} G^{-1} (c - c_{opt}) \end{aligned} \quad (2.7)$$

Later, [Halvorsen et al. \(2003\)](#) developing the exact local method, showed that the loss function can be rewritten as in [Equation 2.8](#)

$$z = J_{uu}^{\frac{1}{2}} \left[\left(J_{uu}^{-1} J_{ud} - G^{-1} G_d \right) \Delta d + G^{-1} n \right] \quad (2.8)$$

With J_{ud} and J_{uu} corresponding to the hessian with respect to the disturbances and manipulated variables $\left(\frac{\partial^2 J}{\partial u \partial d} \right)$ and with respect to the manipulated variables $\left(\frac{\partial^2 J}{\partial^2 u} \right)$, respectively. If one assumes that W_d is a (diagonal) magnitude matrix that considers the disturbances and W_n^y the magnitude matrix that takes into account the measurement error, and considering that both are 2-norm-bounded ([Halvorsen et al. \(2003\)](#) and [Alstad,](#)

Skogestad, and Hori (2009) contains a discussion and justification for using 2-norm), Equations 2.9 to 2.11 can be defined to scale the system:

$$d - d^* = W_d d' \quad (2.9)$$

$$n = HW_n^y n^{y'} = W_n n^{y'} \quad (2.10)$$

$$\left\| \begin{pmatrix} d' \\ n^{y'} \end{pmatrix} \right\|_2 \leq 1 \quad (2.11)$$

The loss function from Equation 2.7 can be also written in a more appropriate way considering the definition of (ALSTAD; SKOGESTAD; HORI, 2009) of the uncertainty variables regarding the contribution of the disturbances and measurement error on the incurred loss, Equation 2.12 and considering the scaled system from Equations 2.9 to 2.11

$$M \triangleq [M_d \quad M_n^y] \quad (2.12)$$

where

$$\begin{aligned} M_d &= -J_{uu}^{1/2} (HG^y)^{-1} HFW_d \\ M_{n^y} &= -J_{uu}^{1/2} (HG^y)^{-1} HW_{n^y} \end{aligned} \quad (2.13)$$

with F corresponding to the optimal measurement sensitivity matrix with respect to the disturbances.

Finally, if one uses all the definitions described so far, the worst-case loss for the effect of the disturbances and measurement error is given by Equation 2.14

$$L_{\text{worst-case}} = \max_{\left\| \begin{pmatrix} d' \\ n^{y'} \end{pmatrix} \right\|_2 \leq 1} = \frac{\bar{\sigma}(M)^2}{2} \quad (2.14)$$

Equation 2.14 shows that in order to minimize the worst-case loss, it is necessary to minimize $\bar{\sigma}(M)$, Equation 2.15:

$$H = \arg \min_H \bar{\sigma}(M) \quad (2.15)$$

This optimization problem was initially solved using a numerical search, as proposed by Halvorsen et al. (2003). Fortunately, Alstad, Skogestad, and Hori (2009) derived an explicit solution that gives the optimal linear combination of measurements coefficient matrix (H) that minimize the worst-case loss that exists due to the effect of the disturbances and measurement errors, in Equation 2.16

$$H^T = \left(\tilde{F} \tilde{F}^T \right)^{-1} G^y \left(G^{yT} \left(\tilde{F} \tilde{F}^T \right)^{-1} G^y \right)^{-1} J_{uu}^{1/2} \quad (2.16)$$

where

$$\tilde{F} = [FW_d W_n^y] \quad (2.17)$$

Assuming that $\tilde{F} \tilde{F}^T$ is full rank.

Equation 2.16 has three interesting properties proved by Alstad, Skogestad, and Hori (2009):

1. It applies to any number of measurements (n_y).
2. The solution for H was proved to minimize not only the worst-case, but also the average-case loss. Therefore, if one uses Equation 2.16 seeking the determination of a control structure that minimizes the loss at the worst-case scenario, he is also minimizing the loss for the average-case scenario. This was called as a “super-optimality” by Alstad, Skogestad, and Hori (2009).
3. The solution proposed minimizes the *combined* effect of the disturbances and the measurement errors, simultaneously.

Therefore, the usage of the explicit solution will give both the minimized worst and average case losses using a single evaluation, and will also consider the combined effect of the disturbances and measurement errors of the problem. Therefore, this solution it is the default one used in *Metacontrol*.

Another way of solving the optimization problem from Equation 2.15 is to use the Extended nullspace method (ALSTAD; SKOGESTAD; HORI, 2009). Differently from Equation 2.16, this solution does not consider the combined effect of the disturbances and measurement errors simultaneously. Instead, the problem is solved in two steps. The first regards “disturbance rejection”: The loss is minimized with respect to disturbances. If there are remaining degrees of freedom, then the effect of the measurement errors can be minimized. The extended nullspace, differently from the exact local method, is not an optimal solution, instead being considered sub-optimal. (ALSTAD; SKOGESTAD, 2007; ALSTAD; SKOGESTAD; HORI, 2009). However, the authors of Alves et al. (2018) also

translated the mathematical formulations of the extended nullspace method into Python, and it is intended to be implemented within *Metacontrol* GUI in future releases merely as a secondary feature, giving its sub-optimality. The solution using the extended nullspace method is depicted in Equation 2.18:

$$H = M_n^{-1} \tilde{J} \left(W_{n^y}^{-1} \tilde{G}^y \right)^\dagger W_{n^y}^{-1} \quad (2.18)$$

Since Equation 2.16 also minimizes the worst-case loss, its evaluation was also considered inside *Metacontrol*: the user can inspect the expected average-case loss for each control structure that can exist in the combinatorial problem. The expression for the average-case loss is a result of the work of Kariwala, Cao, and Janardhanan (2008) and is described in Equation 2.19:

$$L_{\text{average}} = \frac{1}{6(n_y + n_d)} \left\| J_{uu}^{\frac{1}{2}} (H G^y)^{-1} H \tilde{F} \right\|_F^2 \quad (2.19)$$

Lastly, it was necessary to implement within *Metacontrol* a branch-and-bound algorithm capable of quickly searching the best control structures for each possible subset of a given process, using the incurred loss as metric. This was considered by the authors of Alves et al. (2018) as an obligatory feature, since when *Metacontrol* is being used, it was understood that the main idea was to, in a comprehensive software, the user operating it should be capable of inspecting the most promising control structures, and discarding the unnecessary evaluation of the unpromising structures (i.e.: With a high incurred loss - both average of worst-case scenario) to save time and effort. It is important to remember that there is an evident combinatorial problem that grows in an explosive fashion, as the number of the unconstrained degrees of freedom of the reduced space problem and the number of available measurements both increases. Without a search method that is capable of quickly discarding undesired solutions, the usability of *Metacontrol* would be seriously compromised. Luckily, there are several implementations of branch-and-bound algorithms tailored for Self-Optimizing Control studies purposes, such as in Cao and Saha (2005), Cao and Kariwala (2008) and Kariwala and Cao (2009).

From the aforementioned works, Kariwala and Cao (2009) it is of particular interest: the monotonic criterion implemented consists of the exact local method from Halvorsen et al. (2003) and derived explicitly by Alstad, Skogestad, and Hori (2009), which is used as the default methodology to pre-screen the most promising self-optimizing CV candidates in *Metacontrol*. Therefore, the usage of the proposed branch-and-bound algorithm by Kariwala and Cao (2009) it is not only convenient, making the software more effective, but also keeps the “calculation engine” from *Metacontrol* using the same criterion. It would not make any sense, for instance, using a branch-and-bound algorithm that outputs the index of the most promising CVs using the maximum singular value rule from Skogestad

and Postlethwaite (2007) and use the CV index sequence from this algorithm to evaluate the worst-case loss. Fundamentally speaking, the orders of “best” control structures would not be the same, simply because the search method would be using an different criterion from the linear method implemented to evaluate the H matrix.

The Branch-and-Bound algorithm developed by Kariwala and Cao (2009) that was originally implemented in MATLAB[®] by them was translated to Python by the main author of Alves et al. (2018). The same is true for equations of Exact Local and Extended Nullspace methods described by Alstad, Skogestad, and Hori (2009). Those Python routines were packaged under the name of *pySOC* (Python-based Self-Optimizing Control), and can be found in <https://github.com/feslima/pySOC>, with the code being freely available for inspection, revision and suggestions.

3 *Kriging* reasoning

Metamodels are a way to represent the world in simpler terms. Think of them as a photograph, they do not capture the moment as whole but can represent it good enough. In this analogy, the moment is a complex process that it is too cumbersome to explain it completely in mathematical terms, and metamodels, as photographs, may serve the purpose of capturing the core trends of this process without being too unwieldy and not losing too much information.

There is a family of metamodeling methodologies, ranging from a simple linear regression to complex neural networks. However, this chapter will be dedicated to discuss *Kriging* surrogates.

The simplest form to represent a real world process (y) through a metamodel (\hat{y}) and its error (ε) is done through [Equation 3.1](#).

$$y(x) = \hat{y}(x) + \varepsilon \quad (3.1)$$

The error ε is associated with the unmodeled effects of the inputs x and random noise (i.e. it cannot be explained in detail but cannot be ignored as well.). When using the *Kriging* methodology as metamodel, this error is assumed to be a probabilistic function of x , or in other words, this error is assumed to be *not* independent and identically distributed. The specific probabilistic function is represented by a Gaussian distribution with mean zero and variance σ^2 .

$$\varepsilon = \varepsilon(x) \sim \mathcal{N}(0, \sigma^2) \quad (3.2)$$

As from [Søren Nymand Lophaven, Hans Bruun Nielsen, and Jacob Søndergaard \(2002\)](#), a *Kriging* metamodel is comprised of two parts: a polynomial regression \mathcal{F} and departure function z of stochastic nature, as can be seen in [Equation 3.3](#).

$$\hat{y}_l(x) = \mathcal{F}(\beta_{:,l}, x) + z_l(x), \quad l = 1, \dots, q \quad (3.3)$$

The regression model, considered as a linear combination of t functions ($f_j : \mathbb{R}^n \rightarrow \mathbb{R}$), as defined in [Equation 3.4](#).

$$\mathcal{F}(\beta_{:,l}, x) \equiv f(x)^T \beta_{:,l} \quad (3.4)$$

The most common choices for $f(x)$ are polynomials with orders ranging from zero (constant) to two (quadratic). It is assumed that z has mean zero, and the covariance between two given points, arbitrarily named w and x for instance, is defined by Equation 3.5:

$$\text{Cov}[z_l(w), z_l(x)] = \sigma_l^2 \mathcal{R}(\theta_l, w, x), \quad l = 1, \dots, q \quad (3.5)$$

With σ_l^2 being the process variance for the l th response component, and $\mathcal{R}(\theta, w, x)$ defined as the correlation model. In *Metacontrol*, the correlation model used is described in Equation 3.6.

$$\mathcal{R}(\theta_l, w, x) = \exp\left(-\sum_{i=1}^m \theta_l (w - x_i)^p\right), \quad (\theta_l \geq 0, p_l \in [0, 2]) \quad (3.6)$$

Two important concepts must be addressed at this point: The first regards the meaning of the hyperparameter θ , being interpreted as the “activity” of variable x , meaning that, a low value of θ indicates that the points are highly correlated (ALVES et al., 2018). In addition, the value of θ also indicates how fast the correlation goes to zero as the process moves in the l th direction, as discussed by Caballero and Grossmann (2008). The second concept regards the parameter p in Equation 3.6, that represents the “smoothness” of the correlation. As its value reduces, the rate of the initial correlation drops as the distance between w and x_i increases. When $p \approx 0$, there is a discontinuity between $Y(w)$ and $Y(x_i)$ (FORRESTER; SOBESTER; KEANE, 2008) and there is no immediate correlation between the given points.

The hyperparameters θ are degrees of freedom available for optimization purposes, seeking the improvement of the metamodel fitness. In Søren Nyman Lophaven, Hans Bruun Nielsen, and Jacob Søndergaard (2002), the optimal set of hyperparameters θ^* corresponds to the maximum likelihood estimation. Assuming a Gaussian process (LOPHAVEN, S.; NIELSEN, H.; SØNDERGAARD, Jacob, 2002), the optimal values of the hyperparameters solves Equation 3.9:

$$\min_{\theta} \left\{ \psi(\theta) \equiv |R|^{\frac{1}{m}} \sigma^2 \right\} \quad (3.7)$$

Where $|R|$ is the determinant of the correlation matrix. The internal optimizer used in *DACE* corresponds to a modified version of the *Hooke & Jeeves* method, as showed by S. N. Lophaven, H. B. Nielsen, and J. Søndergaard (2002).

As stated before, high-order data obtainment it is an obligatory step in the proposed methodology implemented in *Metacontrol*. Fortunately, Søren Nyman Lophaven, Hans Bruun Nielsen, and Jacob Søndergaard (2002) also derived expressions for Jacobian

evaluation of a *Kriging* prediction (for full demonstration, consult Søren Nymand Lophaven, Hans Bruun Nielsen, and Jacob Søndergaard (2002)), given in Equation 3.8:

$$\hat{y}'(x) = J_f(x)^T \beta^* + J_r(x)^T \gamma^* \quad (3.8)$$

The expression for Hessian evaluation was derived by Alves et al. (2018) (full demonstration in appendix A of their work), and it is depicted in Equation 3.9:

$$\hat{y}''(x) = H_f(x) \beta^* + H_r(x) \gamma^* \quad (3.9)$$

Equations 3.8 and 3.9, differently from numeric/automatic differentiation, are not approximations and, instead, are analytical expressions derived by Søren Nymand Lophaven, Hans Bruun Nielsen, and Jacob Søndergaard (2002) and Alves et al. (2018). Therefore, it is expected a reduced error when one is using these expressions, if compared to techniques based in numerical approximation, considering that the *Kriging* metamodel used is precise enough.

For the design of experiments part, it was decided to implement the Latin Hypercube Sampling (LHS) because it allows to better sample the optimization domain without introducing ill-conditioning in the spatial correlation matrix calculated by the *Kriging* builder.

Lastly, both the LHS function and *Kriging* model builder/predictor were implemented as a separated package in Python under the name of *pydace* (from *Python toolbox for Design and Analysis of Experiments*). This package is a partial code translation from the MATLAB® toolbox implemented by Søren Nymand Lophaven, Hans Bruun Nielsen, and Jacob Søndergaard (2002) named *DACE* to the Python programming language. The link to the open-source code is <https://github.com/feslima/pydace>. There the reader can find a brief documentation on how to install and example of usage.

4 Non linear optimization and infill criteria

When dealing with a non linear problem, such as in [Equation 2.1](#), typically it is resorted to classical solvers (e.g. SQP, trust-region-dogleg, genetic algorithms, simulated annealing, etc.) to obtain its solution, depending on the nature of the NLP (e.g. presence of discontinuities, whether or not the function is differentiable, etc.).

There is a entire field of study dedicated to find these NLP solutions with *Kriging* surrogates. In the works of [Jones \(2001\)](#), [Sasena \(2002\)](#), [Forrester, Sobester, and Keane \(2008\)](#) and [Alexandrov et al. \(2000\)](#), there are entire discussions and frameworks on how to solve non linear problems and comparisons of several metrics involved in the optimization process with metamodels.

The premise of performing a optimization using surrogates is that the model to be optimized is too time consuming or computationally expensive to be solved with classical solvers. To circumvent this, the following steps are proposed:

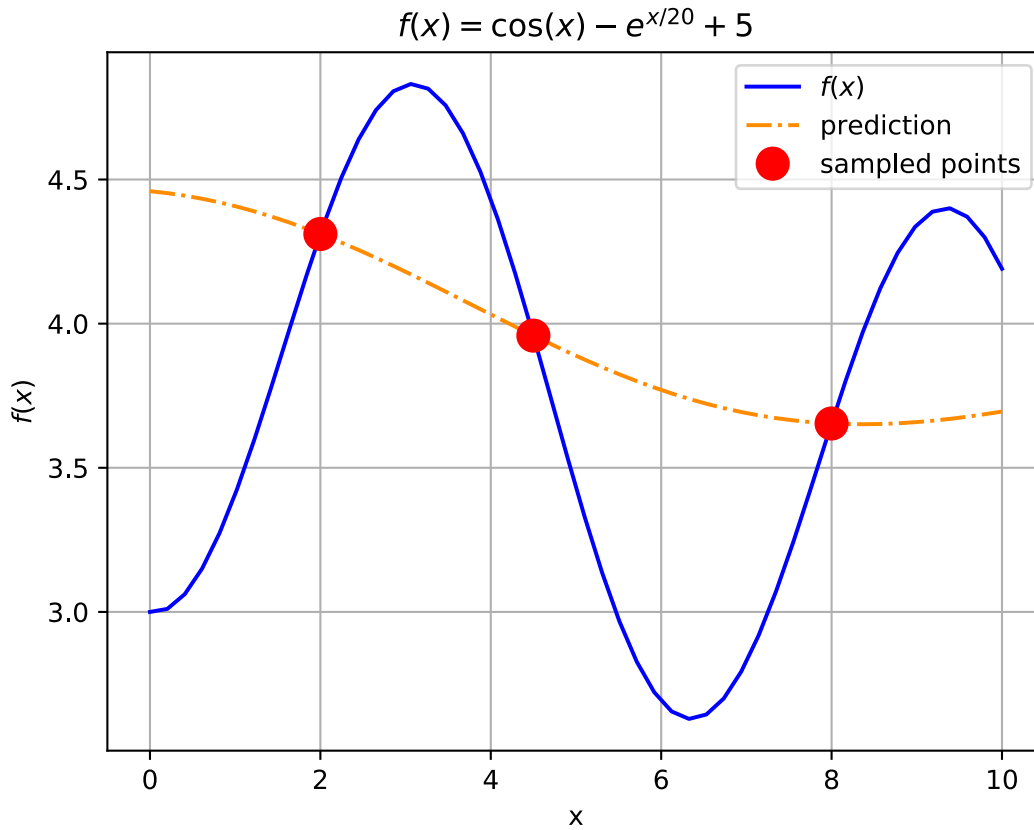
1. Build an approximation model with *Kriging* surrogates using a limited number of initial samples. This approximation is a “generalistic” enough representation of the real model;
2. Perform a optimization of the approximation model using classical NLP solvers and an infill criteria. The surrogate model reduces the “search area” needed by the solver;
3. Compare the surrogate optimum found in step 2 with the result from original model. In other words: feed the results from the *Kriging* metamodel optimum into the original model and see if they are close enough;
4. If the optimum from the metamodel is close enough (based on a chosen metric) to the original model, then this may be the true optimum. Otherwise, update the *Kriging* model by introducing the value found and return to step 2;

This process is basically “filling holes” (hence the name *infill*) in our *Kriging* metamodel until original model optimum is found. To illustrate this in the simplest way, suppose a complex process that we need to optimize that is represented by the following function:

$$f(x) = -\cos(x) - e^{\frac{x}{20}} + 5$$

Assuming that we only have three initial points sampled from this model function, we build our *Kriging* model. As can be seen in Figure 1.

Figure 1 – Initial plot of our complex model. The solid blue line represents the function behavior. The dashed line is the *Kriging* metamodel of the three sampled points (red circles) available.

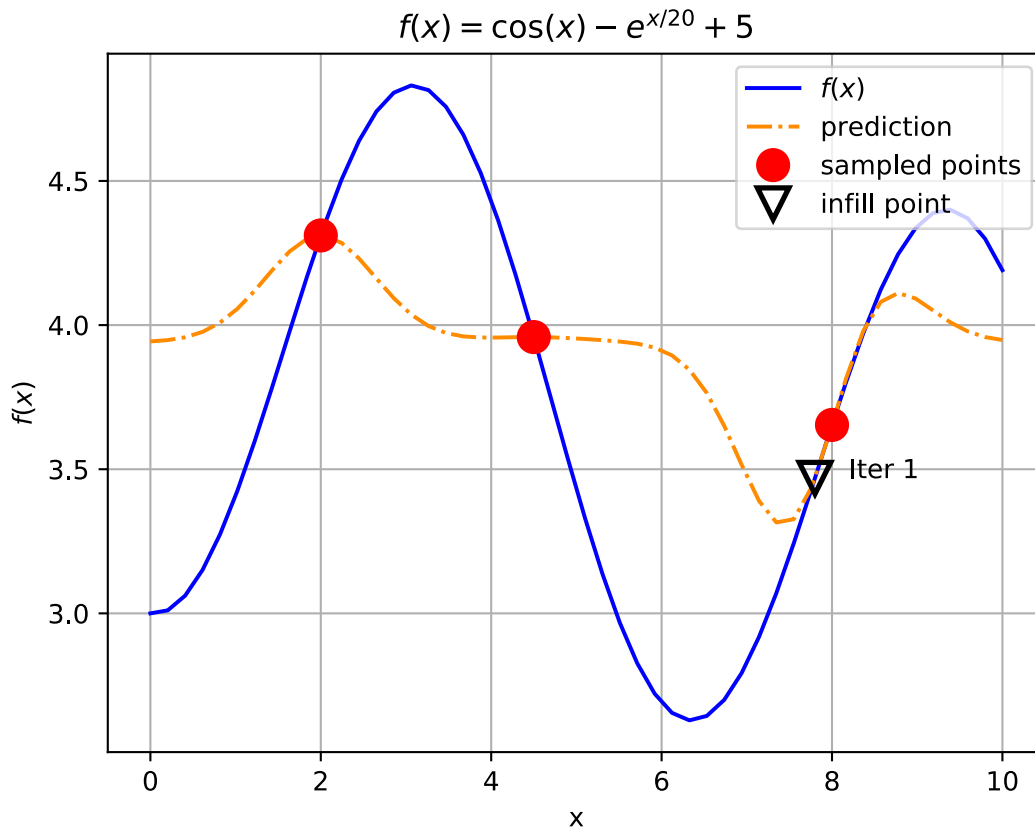


Source: Author.

When applying an optimization solver on the *Kriging* model, we get a new optimal value for x near 7.8 (3.47 for $f(x)$ when we consult the original model). Now, we include these values of $(x, f(x))$ in the sample and rebuild the *Kriging* metamodel. The result is shown Figure 2. We keep repeating this procedure until we get the result in Figure 3.

This example is a trivial one because the problem involves a single input variable and infill criteria is the own *Kriging* prediction of the model. As discussed in Jones (2001), this criteria has its pitfalls if used without other precautions.

Caballero and Grossmann (2008) presented an algorithm, based on the “method 2” in the work of Jones (2001), referred as a gradient matching technique where the gradient of the surrogate is forced to match with the true function gradient, this is done through trust-region approach to ensure local convergence which was proven in the work of Alexandrov et al. (2000). The basic idea of this approach is: minimize the NLP problem

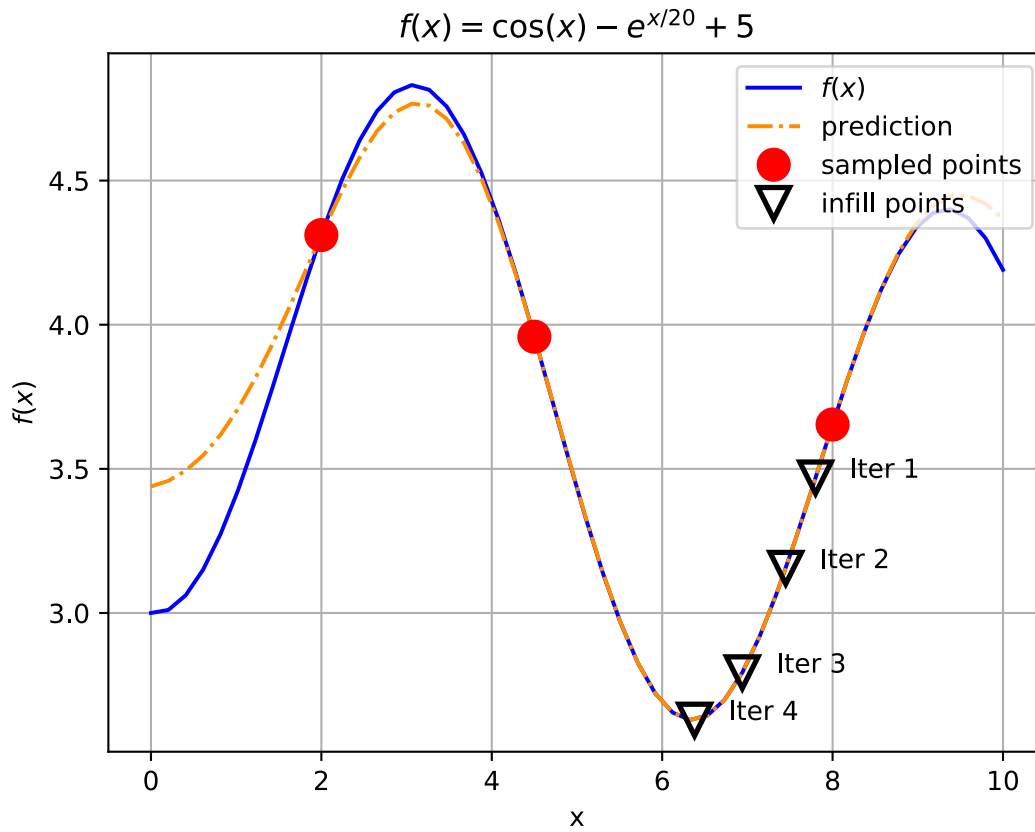
Figure 2 – The *Kriging* model after one update.

Source: Author.

metamodel, consult the original function at the minimum found in the metamodel, update the sample matrix used to build the surrogate. Repeat this until a convergence criteria is met. The flowchart depicting the whole procedure is defined in Figure 4. For detailed explanation of each step of the proposed algorithm, one must refer to Caballero and Grossmann (2008) and Alves et al. (2018).

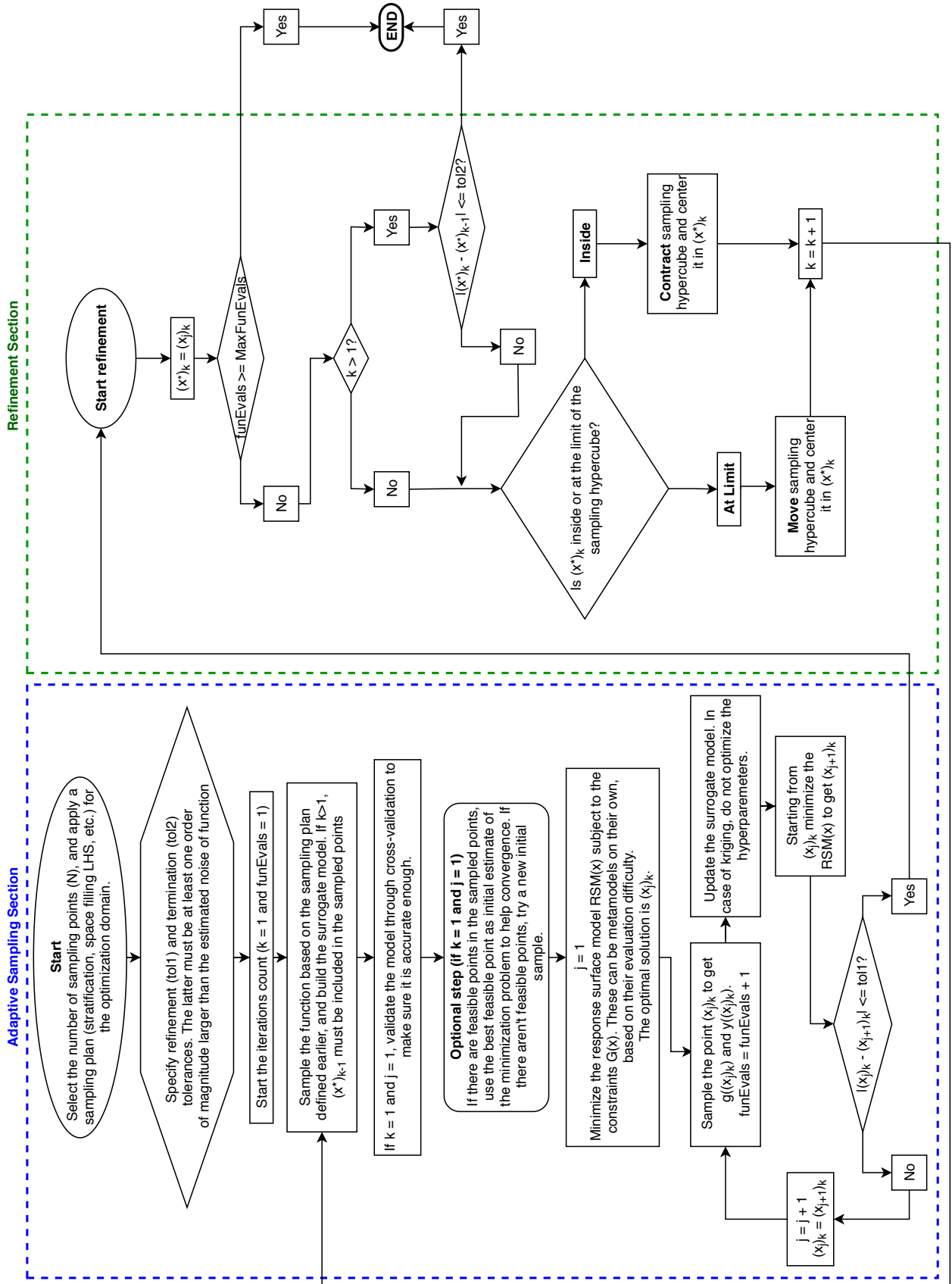
This approach was implemented as a procedure of the Python package *surropt* (from *Surrogate Optimization*). It uses as internal NLP solver a Python wrapper authored by Kummerer and Moore (2019) of the well-established *IpOpt* package (WÄCHTER; BIEGLER, 2006). The *surropt* package is found on <https://github.com/feslima/surropt>.

Figure 3 – The *Kriging* model after four updates. Notice how the *Kriging* model adjusts to the true function.



Source: Author.

Figure 4 – Flowchart of Caballero and Grossmann (2008) algorithm, translated to Python by the author of this work and implemented within *Metacontrol*.



5 The *Metacontrol* framework

To apply the “top-down” part (SKOGESTAD, 2000) of the SOC methodology the conventional way (ALVES et al., 2018; ALSTAD; SKOGESTAD; HORI, 2009; SKOGESTAD, 2000), the following steps are typically involved:

1. Identify the relevant process variables: manipulated variables, disturbances and potential CV candidates (process measurements) in order to perform a Degree of Freedom (DOF) analysis (taking into account both steady and dynamic state of the process);
2. Define optimal operation: Define the objective function to be used in order to seek an optimal operating point;
3. Modeling of the industrial process (using a process simulator or or any numerical environment, for instance) as close as possible to the reality;
4. Optimize the process model;
5. Implement the control loops of active constraints found in the previous step - “active constraint control” (SKOGESTAD, 2000);
6. Evaluate the loss (result of a constant setpoint policy as showed by Skogestad (2000) and Halvorsen et al. (2003)) for each possible control structures for the remaining (unconstrained) degrees of freedom available: This can be done manually, evaluating each possible control structures one at a time (“brute-force” approach (UMAR et al., 2012)), which is, very often, an impracticable approach due to combinatorial explosion (ARAÚJO; GOVATSMARK; SKOGESTAD, 2007). Therefore, it is more efficient to “pre-screen” the most promising CV candidates using local (linear) methods that have been developed and applied by several authors such as Halvorsen et al. (2003), Hori, Skogestad, and Alstad (2005), Eduardo Shiguelo Hori and Skogestad (2008) and Alstad, Skogestad, and Hori (2009). For the latter approach, it is necessary to obtain the reduced-space problem (unconstrained) differential information (gradient with respect to CV candidates and disturbances, and also the objective function Hessian) evaluated at the optimal point found in step 4;
 - a) When using the local methods, it is necessary to define disturbances magnitudes and the measurement errors of the candidates of step 1;
 - b) To evaluate the loss using local methods, one have to apply the mathematical formulations involved in these methods to obtain the candidates variables

combinations and their respective losses. The mathematical formulations that can be used are mainly: The maximum gain rule authored by Skogestad and Postlethwaite (2007), the exact local method derived by Halvorsen et al. (2003) and analytically solved by Alstad, Skogestad, and Hori (2009) or the nullspace method derived by Alstad and Skogestad (2007);

7. Perform a controllability analysis based on the results from step 6 in order to determine the most efficient MV-CV pairings.

Even though it is possible to describe the methodology in steps, its application is not so simple. That is, many of those steps take place in different environments. For instance:

- In steps 1 and 2, it is the closest as engineers, have to a brainstorming session, where it is considered the variables that best describe the process, which of these will yield better convergence in the process simulator; which of them will be realistic enough when designing a control system, etc. Then perform a degree of freedom analysis for both steady state and dynamic state (i.e. the DOF analysis of one seldom is the same as the other). In addition, there is the performance criteria decision in the objective function it is going to be optimized.
- In step 3, it is necessary to simulate with a software package, the process to a minimum satisfaction standard. Or in other words, is this simulation a realistic enough representation of the process?
- Sometimes, the process simulator (e.g. Aspen Plus, Aspen HYSYS, Unisim, etc.) optimization routines are not capable of solving the nonlinear problem that has been defined. So, in step 4, it may be needed to resort an external optimization package (e.g. IpOpt, GAMS, MATLAB® optimization toolbox, etc). This is another environment to work with. In other words, an additional “layer” of complexity.
- If it is necessary the usage of an external NLP solver, then one have to go back to the process simulator and implement the active constraints, as required by the SOC methodology.
- In step 6, to obtain the differential information required, there are different approaches in order to do so:
 1. Extracting manually from the simulator (i.e. performing a first and second order numerical differentiation by applying the differentiation steps and collecting the output from the simulator);

2. Using another external package to extract the gradient and hessian (i.e. Automatic differentiation packages);
3. Using a surrogate approximation of the process simulation in the optimum region, and extracting the differential information from this metamodel;

Option 3 was proposed as solution in the work of [Alves et al. \(2018\)](#), and is implemented in *Metacontrol*. Options 1 and 2 are not implemented due to difficult nature inherent to them. For example, option 1 is a tedious task, even impossible depending on the number of variables to apply the differentiation steps required, and human-error prone since each step applied is done manually. In both options, another limitation that one faces regards the physical meaning of the variables involved in the numeric differentiation process. Strictly speaking: the simulation package does not accept negative values for variables such as flowrates. Compositions are limited to values between 0 and 1. Temperatures may or not accept negative values depending on their units (from 0 to infinity for Kelvin, or -273 to infinity in degree Celsius), etc. Thus, if the numeric differentiation package try to step into outside of these valid value ranges, the simulation software will simply not converge.

- In Step 6 and 7, one have to use another numeric environment to implement algorithms and equations from [Kariwala and Cao \(2009\)](#), [Alstad, Skogestad, and Hori \(2009\)](#) and [Alves et al. \(2018\)](#) in order to perform the calculations necessary to obtain the controlled variables candidates combinations and analyze the controllability of the process studied, respectively.

As can be seen, this is a methodology implementation that goes back-and-forth between several numeric computation (e.g. MATLAB®, Octave, Microsoft Excel, etc.) and simulation (e.g. Aspen Plus, Aspen HYSYS, Unisim, etc.) environments. Therefore, *Metacontrol* was created as a software package that allows all of these steps to be done in a single environment (or at least, keep the necessity of transition to a minimum) for the sake of convenience to apply the SOC methodology.

5.1 *Metacontrol workflow*

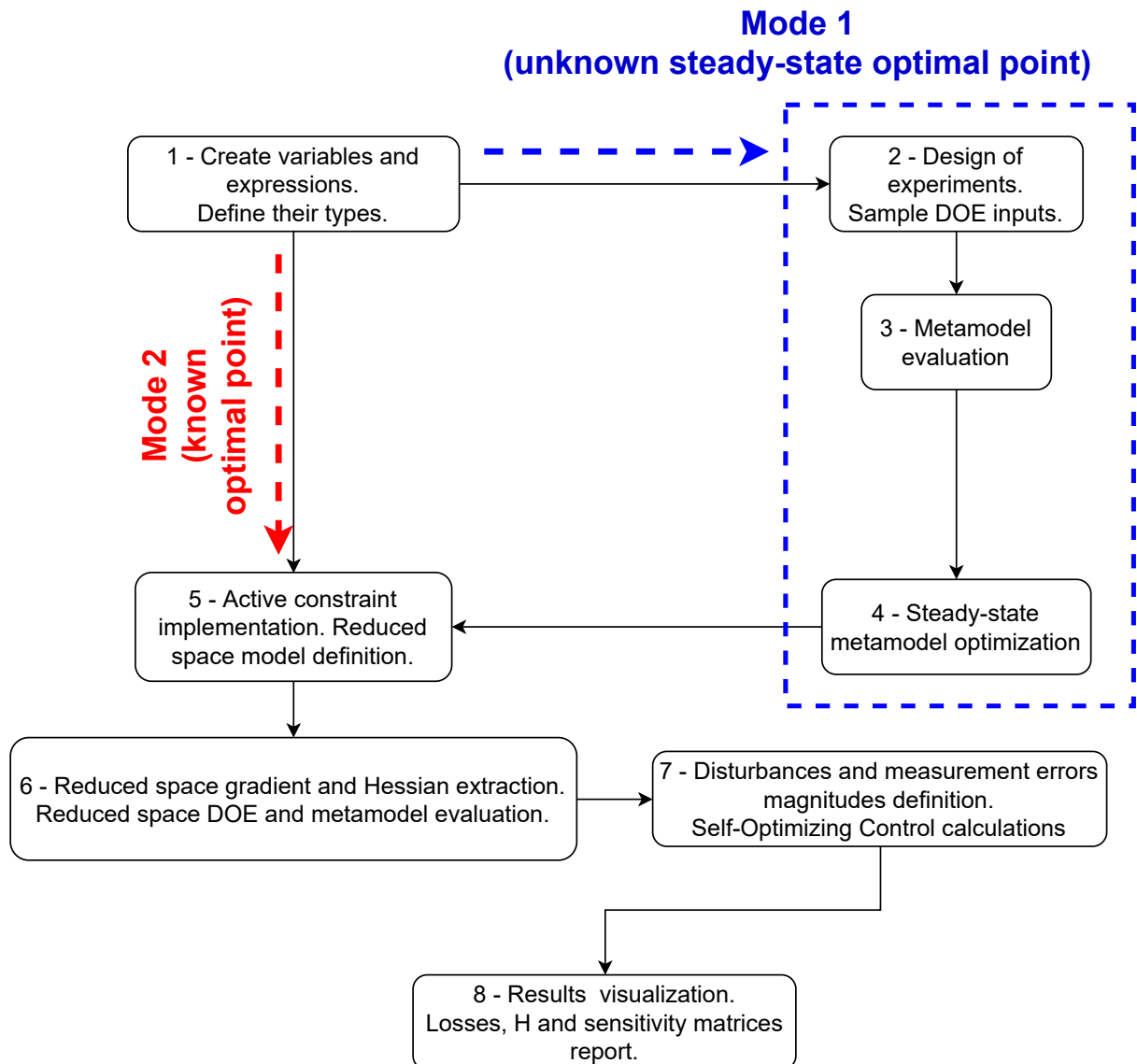
The tool has two modes of operation:

1. The user wishes to apply the methodology proposed by [Alves et al. \(2018\)](#) completely, as seen in [Figure 5](#) represented by the blue dashed arrow and rectangle. That is, he needs to create and define variables and expressions (1), perform a Design of Experiments (DOE) of the industrial process (2), build its NLP constraints and objective function metamodels (3), optimize this NLP metamodel (4). If the

optimization process is successful, implement its active constraints and obtain the reduced space model (5), build another metamodel of the objective function and controlled candidates variables in reduced space, then extract the gradient and Hessian (6). Finally, apply the SOC methodology described by [Alstad, Skogestad, and Hori \(2009\)](#) (7-8).

2. The user already knows the process steady-state optimum, and wishes to apply the methodology partially, see the red dashed arrow in [Figure 5](#). He needs to create the variables/expressions (1), then he implements the active constraints and generate the reduced space metamodels (5). Extract the gradients and hessian needed (6), define the rest of inputs needed to perform the SOC analysis (7) and analyze its results (8).

The only difference between modes 1 and 2, is that the user, when opting for mode 2, skips steps (2) to (4) in mode 1. Everything else is the same.

Figure 5 – Flowchart describing how *Metacontrol* works.

Source: Author

6 Case studies applied in *Metacontrol*

In this chapter will be presented three case studies of Self-Optimizing Control that are applied in the *Metacontrol* software. Each one of them use different objective function criteria with varying complexity. They are:

1. A CO₂ compression purification unit (CPU). The optimization criteria here is the performance enhancement of the process through the reduction in the energy consumption;
2. A hydrocarbon distillation column. The optimization criteria is the minimization of nominal setpoint deviation;
3. An isomerization process that seeks to convert *n*-butane into isobutane. The objective is maximization of profits;

6.1 The CO₂ Compression and Purification Unit (CPU)

The first case-study to be used as a test-bed in *Metacontrol* consists in a CO₂ compression and purification unit that uses phase separation method to obtain purified CO₂ from oxy-fuel combustion. This process is one of the several that exist in the industry that are capable of reducing the greenhouse effect on climate change (JIN; ZHAO; ZHENG, 2015). The process and its simulation are based on the work of Liu et al. (2019). In addition, their unit is based on the prototype proposed by the International Agency Greenhouse Gas (IEAGHG) R&D program study (DILLON et al., 2005).

The process is depicted in Figure 6. Flue gas is compressed by a three-stage after-cooled compressor before being sent to the cold box, where two multi-stream heat exchangers (E1 and E2) and two separators (F1 and F2) take place. In the base case from Liu et al. (2019), the flue gas is first cooled to -24.51°C and sent to F1, with its bottom stream being the first product of the process. Afterwards, The top stream from F1 is sent to the second multi-stream heat-exchanger (E2) being cooled to -54.69°C before going to separator F2. The bottom stream from this separator consists in the second product of the process, and the top stream from F2 is discarded as vent. Both CO₂ product streams and the vent gas are reheated on both multi-stream heat exchangers. The CO₂ product streams are mixed and become ready for storage. The reader can consult Jin, Zhao, and Zheng (2015) and Liu et al. (2019) for more information about the simulation (i.e.: Raw flue gas conditions, detailed stream and equipment conditions, etc).

From Liu et al. (2019), the author of this work have selected the objective function described in Equation 6.1, that consists in the specific energy consumption, defined as the ratio of energy used in both compressors (MCC and C) to total CO₂ flow rate produced. Therefore:

The units of specific energy consumption of the objective function are kWh/tCO_2 .

- C-1: CO₂ recovery rate $\geq 90\%$
- C-2: CO₂ purity on product stream $\geq 96\%$
- C-3: Temperature of F2 bottom stream $> -56.6^{\circ}\text{C}$

C-1 aims to meet the environmental requirements (LIU et al., 2019) and reduce CO₂ atmospheric emissions (TOFTEGAARD et al., 2010; BUHRE et al., 2005). C-2 is a

result of the demand of CO₂ storage and transportation (LIU et al., 2019). In addition, according to POSCH and HAIDER (2012), the purity addressed in this constraint would realize acceptable energy consumption. Lastly, C-3 exists to avoid CO₂ solidification in the pipeline, since the value of C-3 corresponds to the CO₂ three-phase freezing point (POSCH; HAIDER, 2012; KOOHESTANIAN et al., 2017).

As stated by Liu et al. (2019), the main disturbances in the CPU process are:

- D-1: Flue gas flow rate
- D-2: CO₂ concentration in the flue gas

D-1 and D-2 are a result of the oxy-fuel combustion boiler island (LIU et al., 2019), given the variation of the boiler operation. Load changes in the boiler island and variations of the combustion conditions can generate D-1 and D-2, as also stated by Liu et al. (2019). It is considered a $\pm 5\%$ disturbance amplitude for CO₂ feed composition and flue gas flow rate of the base-case, similarly as Jin, Zhao, and Zheng (2015) and Liu et al. (2019).

The number of degrees of freedom for the CPU process is 4 (JIN; ZHAO; ZHENG, 2015; LIU et al., 2019) for Mode I (Given feed). For the sake of simplicity and without loss of generality, the same DOFs from Jin, Zhao, and Zheng (2015) were used here:

1. MCC outlet pressure (bar)
2. MCC outlet temperature (°C)
3. F1 temperature (°C)
4. F2 temperature (°C)

Using this information and based on the review of control configurations for CO₂ CPU process (LIU et al., 2019), the CV candidates in Table 1 that were considered in this case study are listed.

Table 1 – CV Candidates for CO₂ CPU process.

Variable (alias used in <i>Metacontrol</i>)	Description
mccp/mccpout	Compressor outlet pressure (bar)
mcct/mcctout	Compressor outlet temperature (°C)
f1t/f1tout	F1 temperature (°C)
f2t/f2tout	F2 temperature (°C)
s8t	S8 stream temperature (°C)
fco2out	CO ₂ product flowrate (t/h)
xco2out	CO ₂ product molar fraction
co2rr	CO ₂ recovery rate

With 4 degrees of freedom and 8 CV candidates, there are (Equation 6.2)

$$\binom{8!}{4!} = \frac{8!}{4! \times (8 - 4)!} = 70 \quad (6.2)$$

possible control structures for a single measurement policy (excluding the possible ways of controlling the regulatory layer and the possibility of using linear combinations of measurements). Therefore, the manual evaluation of all possibilities is impracticable and also would need the usage of different software environments. This tedious evaluation, however, can be mitigated by *Metacontrol*.

With the problem defined and the possible CV candidates being listed, it is possible to start to use the capabilities of *Metacontrol* in order to aid the search for a Self-Optimizing Control structure for this case-study. Initially, it is necessary to seek for the variables of the process simulator (Aspen Plus) using the COM interface between the *Metacontrol* software and the process simulator.

Figures 7 and 8 illustrate the process of selecting a *.bkp file, selecting the relevant variables, and adding alias to them.

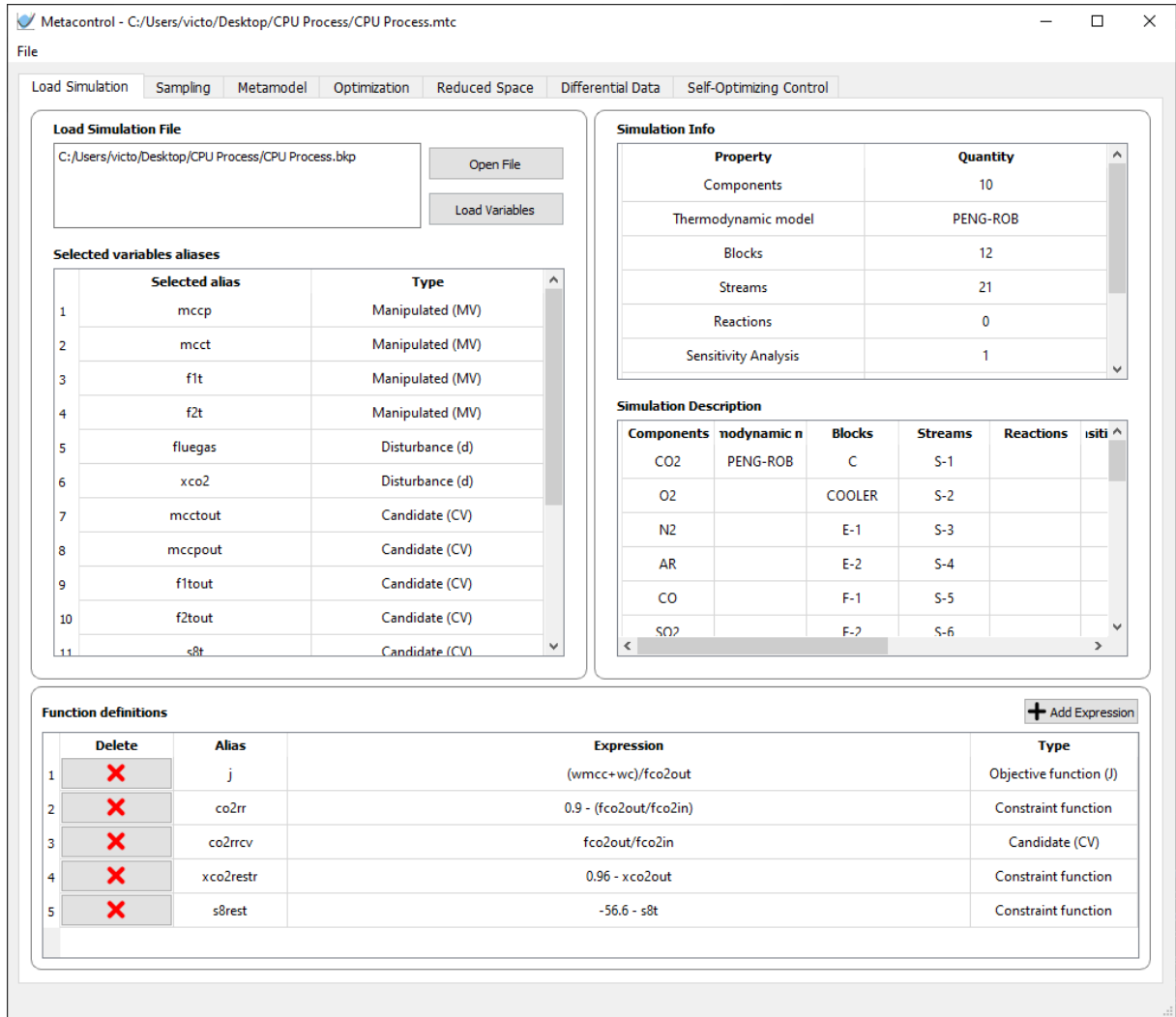
From Figure 7 the user can see that *Metacontrol* shows on its main screen some relevant information: Block names, flowsheet operations (i.e.: Optimizations, sensitivities, calculators), the components selected in the process simulated and the thermodynamic package used. These are enumerated on “Simulation Info” panel and the name of each object is present on “Simulation Description” panel.

After selecting the relevant variables (Decision variables and process measurements) the user can go back to the main screen, where expressions can be created. This functionality aims to give freedom for the user to build expressions based on variables from the process simulator, such as: Objective functions, CV candidates or constraints. Figure 9 shows the specific power consumption, the CO₂ recovery rate expressions being built, based on the auxiliary variables selected on Figure 8.

With the procedure aforementioned being completed, the user can generate the design of experiments (DOE) in order to build *Kriging* responses of the objective function, CV candidates and process constraints. The ranges for each decision variable are taken from Jin, Zhao, and Zheng (2015) (Table 3 from their paper), and this step is illustrated in Figures 10 to 12.

Figure 13 shows the sampling process running. After running all cases, the user can inspect the results of the design of experiments, as can be seen in Figure 14.

With the results of the sampling procedure, the user can go to the “Metamodel” Panel, and select which variables will have *Kriging* responses built, the bounds for the *Kriging* hyperparameters optimization, and the regression and correlation models to be

Figure 7 – *Metacontrol* main screen with CPU process simulation file loaded.

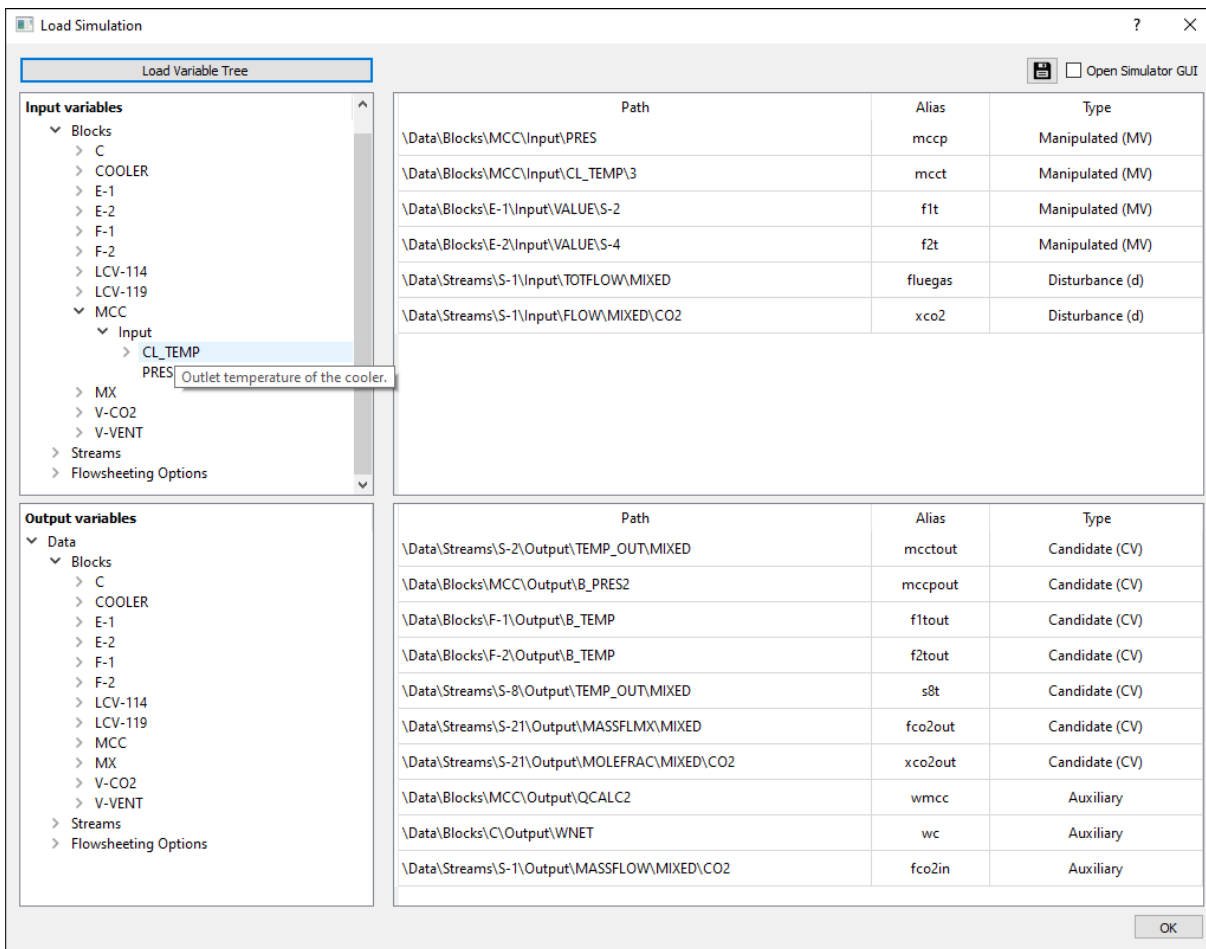
Source: Author.

used. This procedure can be depicted in Figure 15.

The user can also choose which type of validation is going to be performed: *Hold-out* or *K-fold* validation. It is important to point out that this first metamodel generation is performed only to give a quick view of the initial sampling. In other words, to check if the initial sampling is acceptable to be refined by the implementation of the algorithm proposed by Caballero and Grossmann (2008) that is bundled in *Metacontrol*. In addition, if the user chooses *Hold-out* validation, it is possible to view the graphical results (fitness of training set to the metamodel) of each *Kriging* interpolator generated, as can be seen in Figure 16.

In Figure 15, the reader can also inspect under the panel “Validation metrics”, several metrics are used to evaluate reduced models performance, such as: Mean squared error (MSE), Root mean squared error (RMSE), Mean absolute error (MAE), R^2 linear

Figure 8 – Loading variables for the CPU from Aspen Plus simulation and adding alias to them. At the top right corner of this screen, the user is able to select the option to reveal the GUI from Aspen Plus. This features allows the user to inspect inside the process simulator interface to remember any stream or block names. This can be helpful when one is selecting the variables using the COM technology and there are several unit operations blocks and streams, for instance. Another feature that was implemented in order to ease the search of the variables, regards the description of each variable: Hovering the mouse over a COM variable will show its description, extracted directly from the process simulator.









Source: Author.

coefficient, Explained variance (EV), the Sample mean and also its standard deviation.

In order to try to improve the initial sampling for optimization purposes, we go to the “Optimization” tab where the refinement algorithm proposed by [Caballero and Grossmann \(2008\)](#) is implemented. [Figure 17](#) shows the parameters that can be tuned in order to attempt to improve the *Kriging* interpolator using the automated refinement procedure, with further discussion and details regarding each parameter can be found on [caballero2008](#) and in the previous work from the author of this dissertation ([ALVES](#)

Figure 9 – Creating expressions for specific power consumption (objective function), CO₂ recovery rate and S8 Temperature (constraint functions/CV candidates).

Function definitions				 Add Expression
	Delete	Alias	Expression	Type
1		j	(wmcc+wc)/fco2out	Objective function (J)
2		co2rr	0.9 - fco2out/fco2in	Constraint function
3		co2rrcv	fco2out/fco2in	Candidate (CV)
4		xco2restr	0.96 - xco2out	Constraint function
5		s8rest	-56.6 - s8t	Constraint function

et al., 2018). In addition, NLP solvers parameters can also be changed in this screen.

In Figure 17, the user can see the final result of the refinement algorithm: on the “Results” panel, the final results for the decision variables, constraints expressions defined previously and the objective function. In addition, a control panel showing the operations of contraction and movement of the hyperspace performed by the algorithm (and how many iterations on each operation) can be inspected. Figure 18 shows the control panel details of the procedure.

Inspecting Figures 17 and 18, it is clear that the optimal operating point found is

- MCC outlet pressure (bar) = 30.1849
- MCC outlet temperature = 25 °C
- F1 temperature = -30 °C
- F2 temperature = -55°C

That indicates three active constraints that have to be controlled. Regarding stream S8 temperature, CO₂ product purity and recovery rate, these were inactive constraints. Therefore, the reduced space problem has one degree of freedom left for Self-Optimizing Control.

In order to prove the effectiveness of the proposed software and the procedures and algorithms used, an optimization using the process simulator (Aspen Plus) SQP implementation (an optimization block) was performed, and the results can be found in Tables 2 and 3, compared with the results found by *Metacontrol*. They identical, quantitatively and qualitatively (the active constraints found in both approaches). The constraints in *Metacontrol* are written internally in the form $g(x) \leq 0$, and showed in the GUI in the same way, due to NLP solvers and refinement algorithm syntaxes.

After determining the nominal optimal operating point, the active constraints must be implemented in the simulation file externally using the process simulator (Using design

Figure 10 – *Metacontrol* “Sampling panel”. The user can perform the sampling using the process simulator or import a .CSV file.

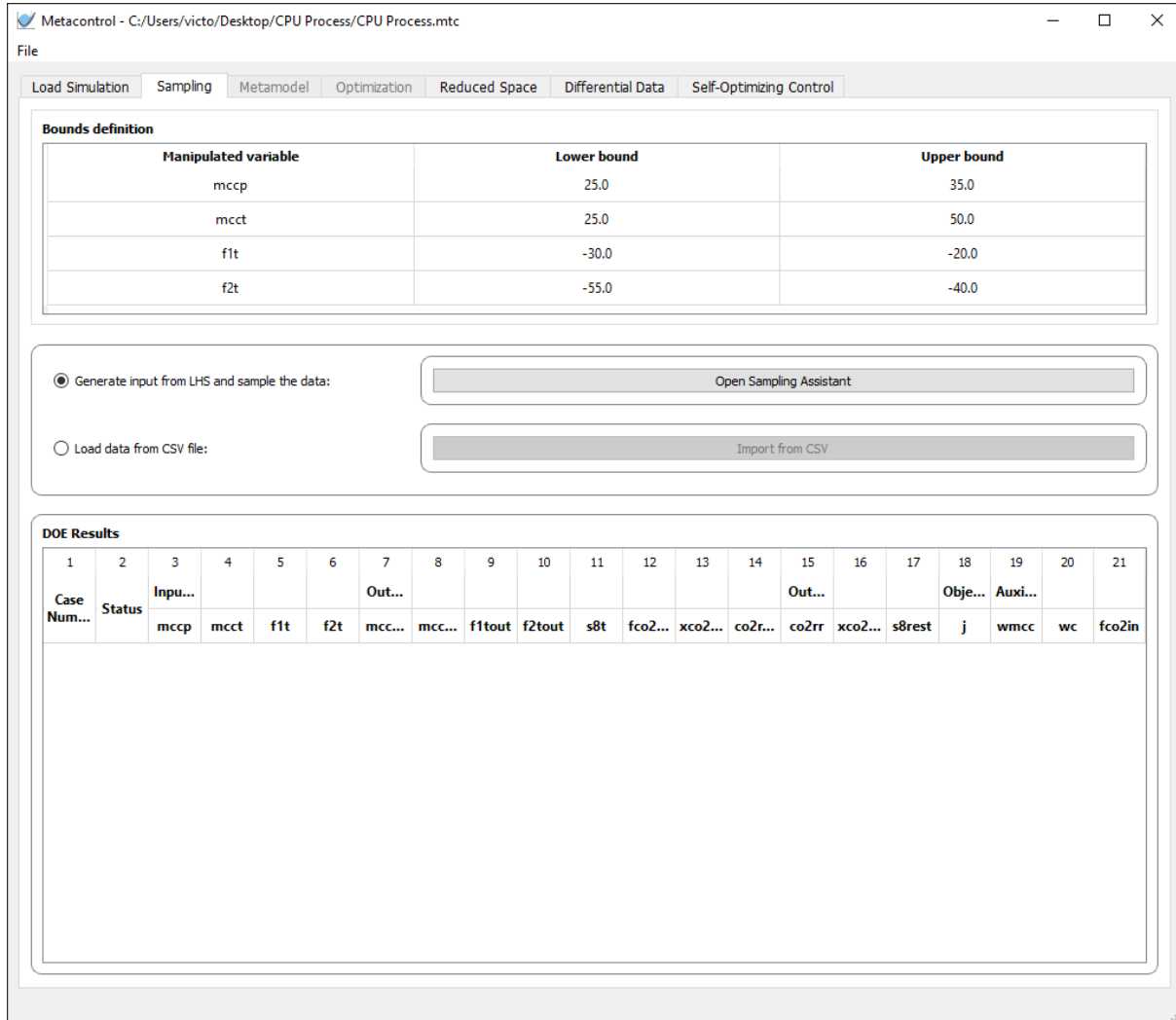


Table 2 – Optimization runs: *Aspen Plus* vs *Metacontrol* - Decision variables and objective function - CPU Process

	Objective function J (kWh/tCO ₂)	MCC Pressure (bar)	MCC outlet temperature (°C)	F1 temperature (°C)	F2 temperature (°C)
Aspen Plus	112.3690	30.0316	25	-30	-55
Metacontrol	112.3691	30.1849	25	-30	-55

specifications for instance) and go back to *Metacontrol*, in order to generate the reduced space *Kriging* metamodel, seeking the obtainment of differential data (e.g.: the gradients G_y, G_y^d , and the Hessians J_{uu} and J_{ud}). The reduced space problem can be sampled using the process simulator linked with *Metacontrol* directly, or importing a *.csv file. Both options mentioned are similar to the initial sampling procedure.

Over the tab “Differential data”, the user is capable of checking which variables are active constraints (either decision variables or nonlinear constraints), inserting the values for the optimal operating point found on the previous step (refined surrogate optimization), and the value for the nominal disturbances. If the user sample the reduced space problem using the *.bcp, he must also input the range for the remaining decision variables to be

Figure 11 – *Metacontrol* Sampling assistant. The limits for the decision variables used in the CPU process are the same from [Jin, Zhao, and Zheng \(2015\)](#) and [Liu et al. \(2019\)](#)

Manipulated variable	Lower bound	Upper bound
mccp	25.0	35.0
mcct	25.0	50.0
f1t	-30.0	-20.0
f2t	-55.0	-40.0

Case Number	Status	Inputs				Outputs									
		mccp	mcct	f1t	f2t	mcctout	mccpout	f1tout	f2tout	s8t	fco2out	fco2in	xco2out	wmcc	wc
1															
2															

Table 3 – Optimization runs: *Aspen Plus* vs *Metacontrol* - Process constraints - CPU Process

	Stream S8 temperature °C	CO ₂ molar fraction	CO ₂ recovery rate
Aspen Plus	-55.8201	0.9674	0.9658
<i>Metacontrol</i>	-55.4859	0.9666	0.9671

sampled and for the disturbances. The range for the remaining degrees of freedom and for the disturbances are suggested to be a small percentage of the nominal values ($\pm 0.5\%$, for instance) in order to train a surrogate model accurate enough at the optimal region, guaranteeing robust high-order data (gradients and Hessians) obtainment, as suggested previously in [Alves et al. \(2018\)](#).

In the CPU process, the MCC operating pressure, MCC temperature, F1 temperature and F1 temperature are active constraints as mentioned previously. Therefore, they should be marked as “active” under the “Variable activity” panel, as shown in [Figure 19](#).

Since in the initial sampling the process simulator was directly linked with *Metacontrol*, in [Figure 19](#) under the “Data source” panel a *.csv file was imported, originated from a sensitivity analysis run done in a *.bkp file of the reduced space problem for the CPU process, in order to show this supplementary feature of *Metacontrol*. This is illustrated in [Figures 20](#) and [21](#).

Figure 12 – *Metacontrol* Latin Hypercube Sampling settings. 80 samples were generated and 5 iterations were performed in order to try to maximize the minimum distance between the points (*maxmin* criterion). The user can also add the vertices of the design of experiments.

The screenshot shows the 'Sampling Assistant' window. At the top, the 'Bounds definition' table lists manipulated variables and their bounds:

Manipulated variable	Lower bound	Upper bound
mccp	25.0	35.0
mcct	25.0	50.0
f1t	-30.0	-20.0
f2t	-55.0	-40.0

Below this, the 'Select a sampling method' section has 'Latin Hypercube Sampling (LHS)' selected. An 'LHS Settings' dialog box is open, showing 'Number of samples: 80' and 'Number of iterations: 5'. There is an unchecked checkbox for 'Include hypercube vertices'. Buttons for 'Generate LHS' and 'Generate Grid' are visible.

The 'Sampler display' section shows a table with 16 columns: Case Number, Status, and 14 output variables (mccp, mcct, f1t, f2t, mcctout, mcpout, f1tout, f2tout, s8t, fco2out, xco2out, wmcc, wc, fco2in). The first two rows are empty. A progress bar at the bottom indicates 0% completion. Buttons for 'Abort', 'Export as CSV', 'Sample Data', 'Done', and 'Cancel' are at the bottom.

After importing the design of experiments from the external source (*.csv) and associating each variable created in *Metacontrol* with the data (as shown in Figure 21), the user can go to “Differential data” tab, in order to generate the reduced space metamodel. Under the panel “Reduced space metamodel training” the button “Open training dialog” allows the modification of the *Kriging* parameters, similarly as done previously in the step of generating the first metamodel to inspect the initial sampling.

The method of high-order data obtainment currently implemented in *Metacontrol* is based on the analytical expressions for the gradients and Hessian derived by Søren Nymand Lophaven, Hans Bruun Nielsen, and Jacob Søndergaard (2002) and Alves et al. (2018), respectively. In future releases of *Metacontrol* there will be two more methods of differentiation as secondary features. These will be based on numeric and automatic differentiation (*numdiff*tools and *autograd* Python toolboxes, respectively), using the surrogate model as source of high-order data obtainment. However, it is strongly recommended the usage of the *Kriging* predictor analytical expressions to ensure results robustness, as stated before in Alves et al. (2018), and also stressed previously on this work.

After opening the training dialog (Figure 23) and configuring the reduced metamodel settings, the user can generate the metamodel, click on “ok”, go back to the main screen

Figure 13 – *Metacontrol* Sampling for the CPU process.

Bounds definition

Manipulated variable	Lower bound	Upper bound
mccp	25.0	35.0
mcct	25.0	50.0
f1t	-30.0	-20.0
f2t	-55.0	-40.0

Select a sampling method

☒ Latin Hypercube Sampling (LHS) Generate LHS ⚙️

☐ Regular Grid Generate Grid ⚙️

Sampler display

Case Number	Status	Inputs				Outputs									
		mccp	mcct	f1t	f2t	mcctout	mccpout	f1tout	f2tout	s8t	fco2out	fco2in	xco2out	wmcc	wc
1	ok	29.6657015	29.2942652	-29.3907744	-49.9604554	29.2942652	29.6657044	-29.3907748	-49.9604553	-55.6135498	588.1720290	621.8130230	0.9688891	65265.4039...	2076.5660700
2	ok	34.1525580	48.9262045	-20.9197710	-45.2326447	48.9262045	34.1525615	-20.9197723	-45.2326447	-45.6294679	585.8128320	621.8130230	0.9654131	68336.6857...	1516.7228400
3	ok	34.4857483	32.7862706	-25.2623994	-45.6658771	32.7862706	34.4857519	-25.2623997	-45.6658772	-44.9843331	589.1250700	621.8130230	0.9630738	68549.0743...	835.5328910
4	ok	27.7995875	33.1728511	-24.2824911	-44.6325633	33.1728511	27.7995901	-24.2824911	-44.6325633	-58.7247318	563.4256790	621.8130230	0.9742024	63855.0393...	5351.3546000

21%

Abort Export as CSV Sample Data Done Cancel

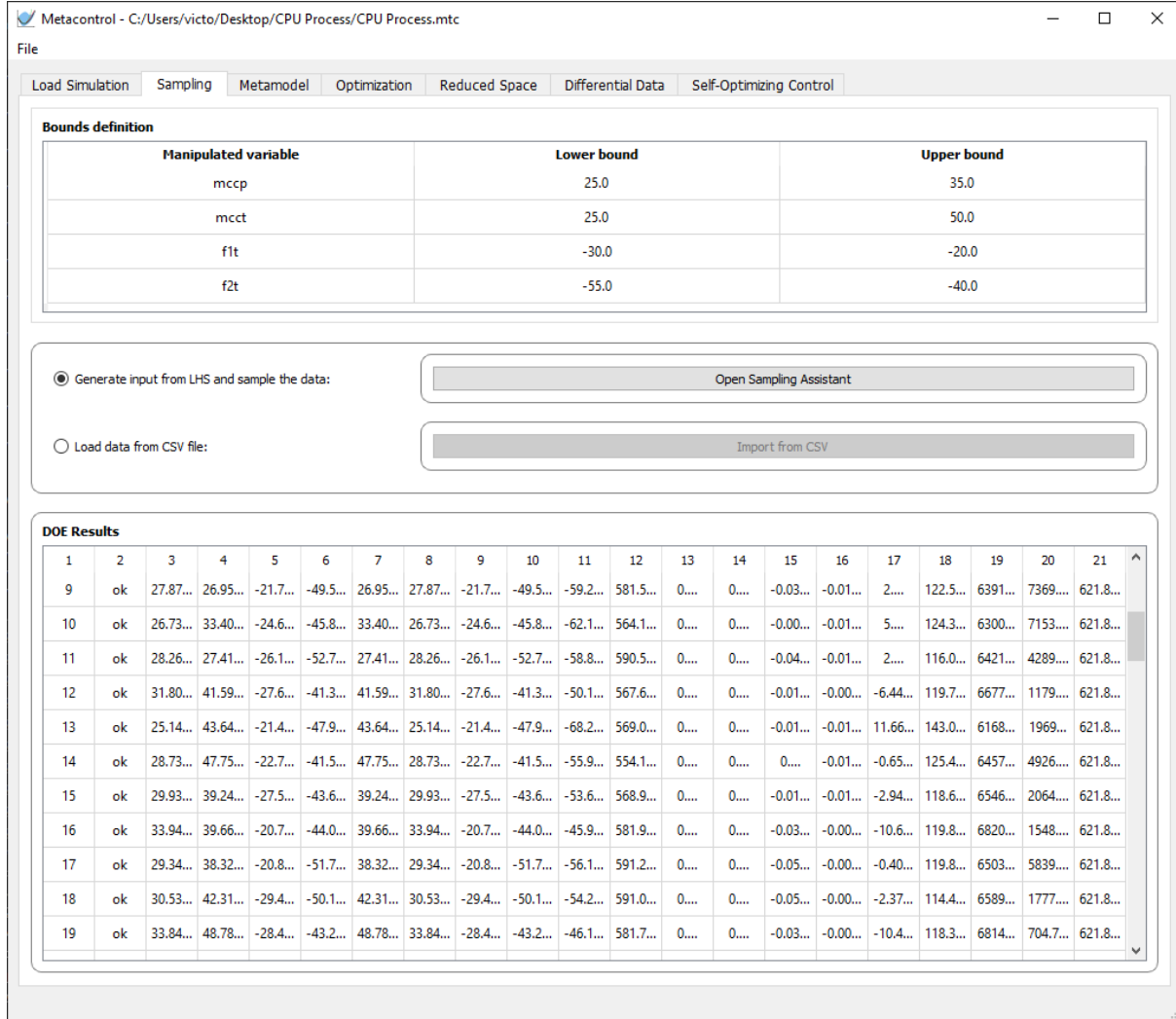
and generate the estimation of the gradients and hessians necessary to carry on the Self-Optimizing Control study. These results are displayed on [Figure 24](#).

In order to prove the effectiveness of the analytical expressions derived by [Søren Nymand Lophaven, Hans Bruun Nielsen, and Jacob Søndergaard \(2002\)](#) and already used in [Alves et al. \(2018\)](#), the gradients obtained using surrogate models in *Metacontrol* were compared against the ones generated in Aspen Plus (Equation-Oriented sensitivity mode). The process simulator does not provide the hessian of any function natively, and therefore J_{uu} and J_{ud} could not be compared. However, the excellent agreement between the values between the gradients found in both procedures can be considered as a sufficiently robust result.

Through inspection of [Table 4](#), the reader can see how robust the results of the gradients obtained by the methodology proposed in the previous work from [Alves et al. \(2018\)](#) and now are automated in *Metacontrol*. The matrix mean-squared error in [Table 5](#) also corroborates this affirmation.

After inspecting the gradients and hessians generated, the user can go to the “Self-Optimizing Control” tab, where the disturbances and measurement error magnitudes will be inserted. As stated previously in this case study, was considered a magnitude of $\pm 5\%$ for the disturbances. For the CO₂ inlet composition, it was considered the absolute value,

Figure 14 – Sampling results, where the user can inspect convergence status and the values of the selected variables for each case.



and for the flue gas flow rate, $\pm 5\%$ of the nominal flow rate. Therefore, in Equation 6.3 :

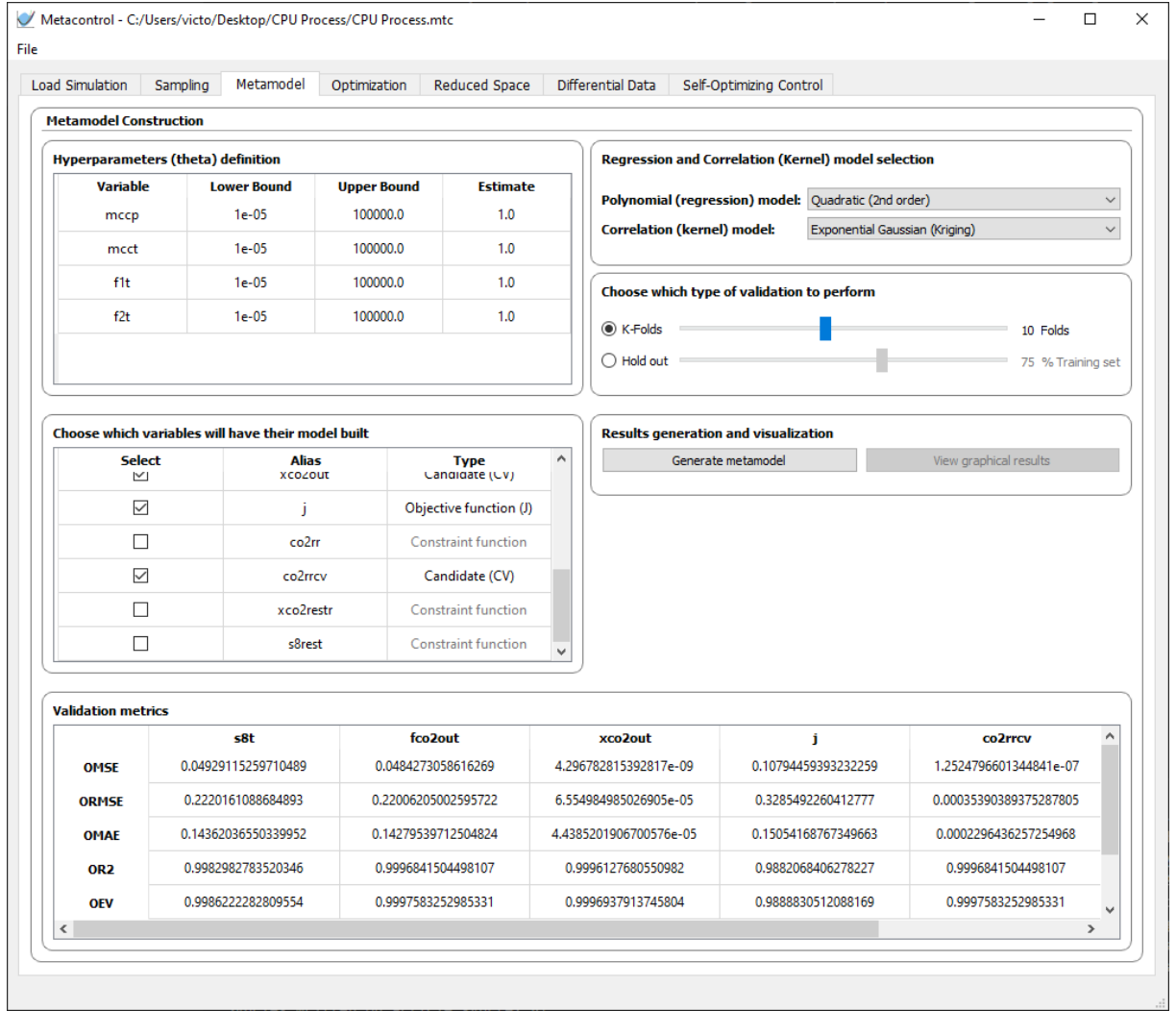
$$W_d = \text{diag}(0.05, 35.8595) \quad (6.3)$$

In addition, for the measurement errors, it was considered $\pm 0.5^\circ\text{C}$ for temperature measurements, 0.01 for pressure and flow measurements and 0.001 for ratios (CO_2 recovery rate and product purity). These assumptions generated by Equation 6.4:

$$W_n^y = \text{diag}(0.001, 0.01, 0.01, 0.5, 0.001) \quad (6.4)$$

The order for Equations 6.3 and 6.4 it is the same from the column order from Figure 24. Figure 25 shows the magnitude matrix data being inserted in *Metacontrol*.

Under the “Subsets sizing options” panel, by default the best control structure for each subset size is evaluated by *Metacontrol*, but the user can change how many subsets

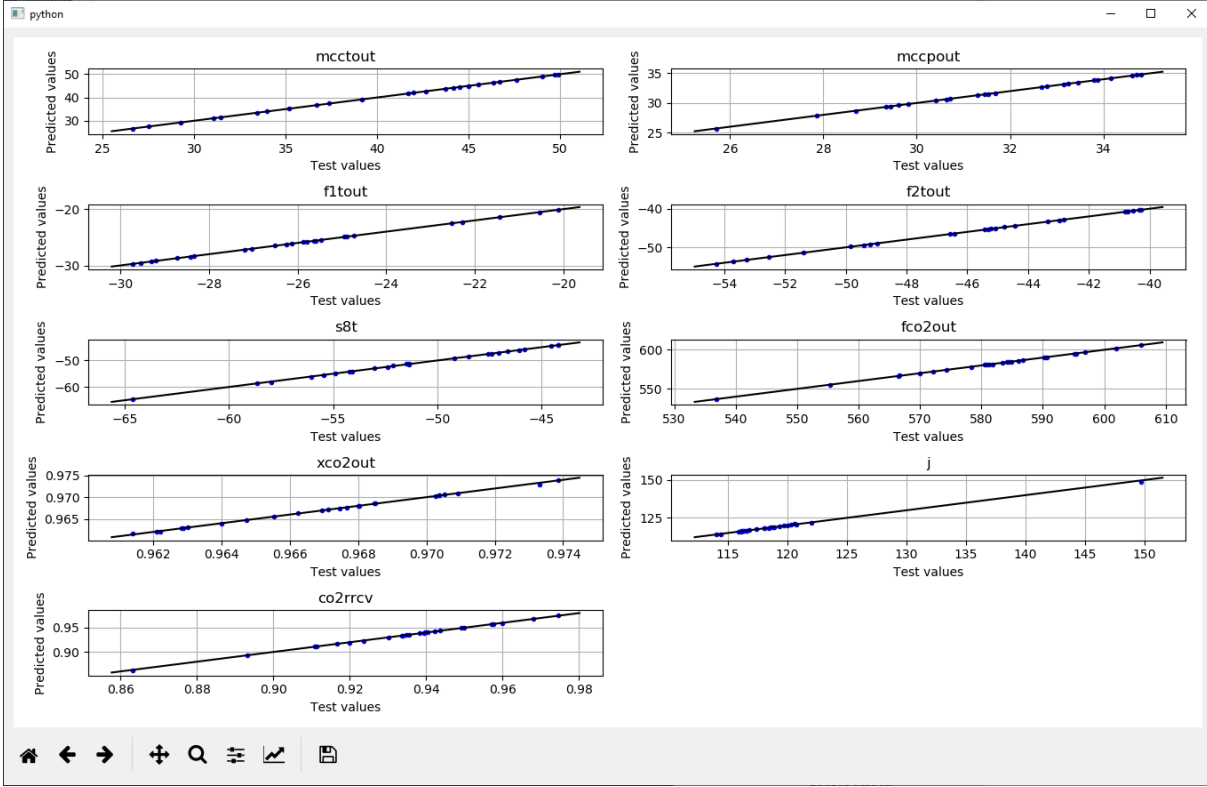
Figure 15 – *Kriging* configuration and validation metrics results.

he wants to evaluate, until the maximum number for each subset size.

After providing all the necessary inputs (magnitude matrices and number of best sets to be evaluated for each subset size), clicking in “Generate results” will show the n th best Self-Optimizing Control structures for each subset size, as can be seen for demonstration purposes in Figures 26 and 27, the results for a single measurement policy, and for linear combinations using 2 measurements at a time. The user can also inspect the H matrix (that will be of ones and zeros for single measurements and a full matrix for linear combinations) and the optimal sensitivity matrix for each subset evaluated.

For instance, considering a single measurement policy for the unconstrained degrees of freedom, Table 6 depicts the best CV candidates in worst-case loss ascending order. The best Self-Optimizing Control variable for the considered case consists in the multi-stage compressor (MCC) outlet pressure. This result can be related to the previous finding Liu et al. (2019).

Figure 16 – Fitness for each metamodel.

Table 4 – High-order data obtainment: *Aspen Plus vs Metacontrol*

	G^y	G_d^y
<i>Metacontrol</i>	$\begin{bmatrix} 0.0036 \\ 2.2406 \\ 1.0000 \\ 2.7354 \\ -0.0017 \end{bmatrix}$	$\begin{bmatrix} -3.0148 \times 10^{-10} & 0.0799 \\ 0.8378 & 146.6549 \\ 5.2806 \times 10^{-9} & 2.5904 \times 10^{-5} \\ -3.4160 \times 10^{-5} & 0.0244 \\ -1.5455 \times 10^{-9} & 0.0040 \end{bmatrix}$
<i>Aspen Plus</i>	$\begin{bmatrix} 0.0036 \\ 2.2403 \\ 1 \\ 2.7330 \\ -0.0017 \end{bmatrix}$	$\begin{bmatrix} 1.3472 \times 10^{-7} & 0.0798 \\ 0.8378 & 146.6124 \\ 0 & 0 \\ 3.3797 \times 10^{-15} & 0.0250 \\ 1.6912 \times 10^{-16} & 0.0040 \end{bmatrix}$

Table 5 – Mean-squared error of high-order data obtainment: *Aspen Plus vs Metacontrol* - CPU Process

	G^y	G_d^y
Mean-squared error	1.1659×10^{-6}	1.8088×10^{-4}

Figure 17 – Refinement algorithm configuration and results screen.

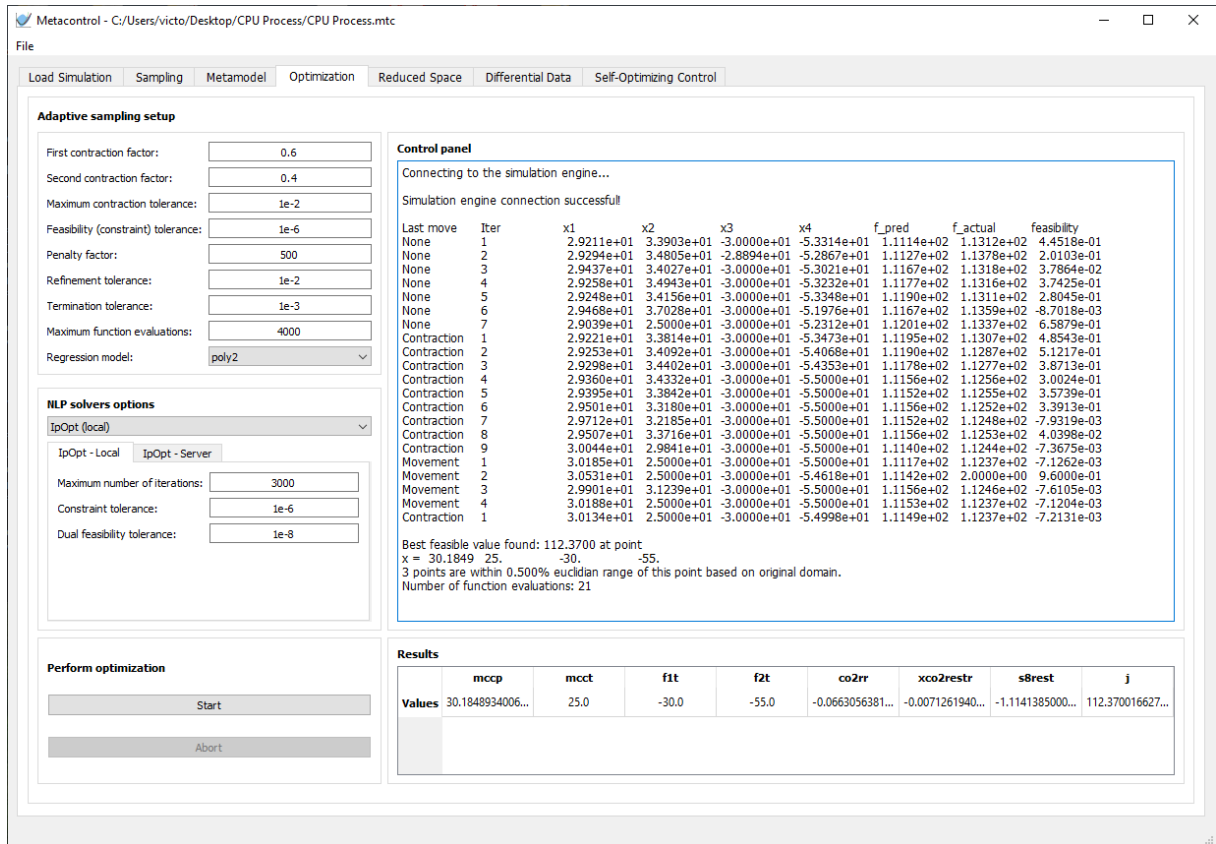


Figure 18 – Refinement algorithm control panel output.

Connecting to the simulation engine...

Simulation engine connection successful!

Last move	Iter	x1	x2	x3	x4	f_pred	f_actual	feasibility
None	1	2.9211e+01	3.3903e+01	-3.0000e+01	-5.3314e+01	1.1114e+02	1.1312e+02	4.4518e-01
None	2	2.9294e+01	3.4805e+01	-2.8894e+01	-5.2867e+01	1.1127e+02	1.1378e+02	2.0103e-01
None	3	2.9437e+01	3.4027e+01	-3.0000e+01	-5.3021e+01	1.1167e+02	1.1318e+02	3.7864e-02
None	4	2.9258e+01	3.4943e+01	-3.0000e+01	-5.3232e+01	1.1177e+02	1.1316e+02	3.7425e-01
None	5	2.9248e+01	3.4156e+01	-3.0000e+01	-5.3348e+01	1.1190e+02	1.1311e+02	2.8045e-01
None	6	2.9468e+01	3.7028e+01	-3.0000e+01	-5.1976e+01	1.1167e+02	1.1359e+02	-8.7018e-03
None	7	2.9039e+01	2.5000e+01	-3.0000e+01	-5.2312e+01	1.1201e+02	1.1337e+02	6.5879e-01
Contraction	1	2.9221e+01	3.3814e+01	-3.0000e+01	-5.3473e+01	1.1195e+02	1.1307e+02	4.8543e-01
Contraction	2	2.9253e+01	3.4092e+01	-3.0000e+01	-5.4068e+01	1.1190e+02	1.1287e+02	5.1217e-01
Contraction	3	2.9298e+01	3.4402e+01	-3.0000e+01	-5.4353e+01	1.1178e+02	1.1277e+02	3.8713e-01
Contraction	4	2.9360e+01	3.4332e+01	-3.0000e+01	-5.5000e+01	1.1156e+02	1.1256e+02	3.0024e-01
Contraction	5	2.9395e+01	3.3842e+01	-3.0000e+01	-5.5000e+01	1.1152e+02	1.1255e+02	3.5739e-01
Contraction	6	2.9501e+01	3.3180e+01	-3.0000e+01	-5.5000e+01	1.1156e+02	1.1252e+02	3.3913e-01
Contraction	7	2.9712e+01	3.2185e+01	-3.0000e+01	-5.5000e+01	1.1152e+02	1.1248e+02	-7.9319e-03
Contraction	8	2.9507e+01	3.3716e+01	-3.0000e+01	-5.5000e+01	1.1156e+02	1.1253e+02	4.0398e-02
Contraction	9	3.0044e+01	2.9841e+01	-3.0000e+01	-5.5000e+01	1.1140e+02	1.1244e+02	-7.3675e-03
Movement	1	3.0185e+01	2.5000e+01	-3.0000e+01	-5.5000e+01	1.1117e+02	1.1237e+02	-7.1262e-03
Movement	2	3.0531e+01	2.5000e+01	-3.0000e+01	-5.4618e+01	1.1142e+02	2.0000e+00	9.6000e-01
Movement	3	2.9901e+01	3.1239e+01	-3.0000e+01	-5.5000e+01	1.1156e+02	1.1246e+02	-7.6105e-03
Movement	4	3.0188e+01	2.5000e+01	-3.0000e+01	-5.5000e+01	1.1153e+02	1.1237e+02	-7.1204e-03
Contraction	1	3.0134e+01	2.5000e+01	-3.0000e+01	-5.4998e+01	1.1149e+02	1.1237e+02	-7.2131e-03

Best feasible value found: 112.3700 at point
 x = 30.1849 25. -30. -55.
 3 points are within 0.500% euclidian range of this point based on original domain.
 Number of function evaluations: 21

Figure 19 – “Variable activity” panel, where the user is capable of highlighting which variables are active constraints and inputting values for them. If an active constraint is a nonlinear constraint, the user must pair this variable with a decision variable (MV) to consume a degree of freedom.

Variable Activity												
Active	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pairing					mcct	mccp	flt	fzt	Select a MV	Select a MV	Select a MV	Select a MV
Type	Manipulated (MV)	Manipulated (MV)	Manipulated (MV)	Manipulated (MV)	Candidate (CV)	Candidate (CV)	Candidate (CV)	Candidate (CV)	Candidate (CV)	Candidate (CV)	Candidate (CV)	Candidate (CV)
Value	30.184893400628656	25.0	-30.0	-55.0								

Figure 20 – Loading a *.csv file in containing design of experiments data in *Metacontrol*: if the user chooses this option, he must provide a file containing all variables selected from the first step (“Load variables” under “Load simulation” tab). the convergence flag is used as a header to map the *.csv, and the software asks the user to select it.

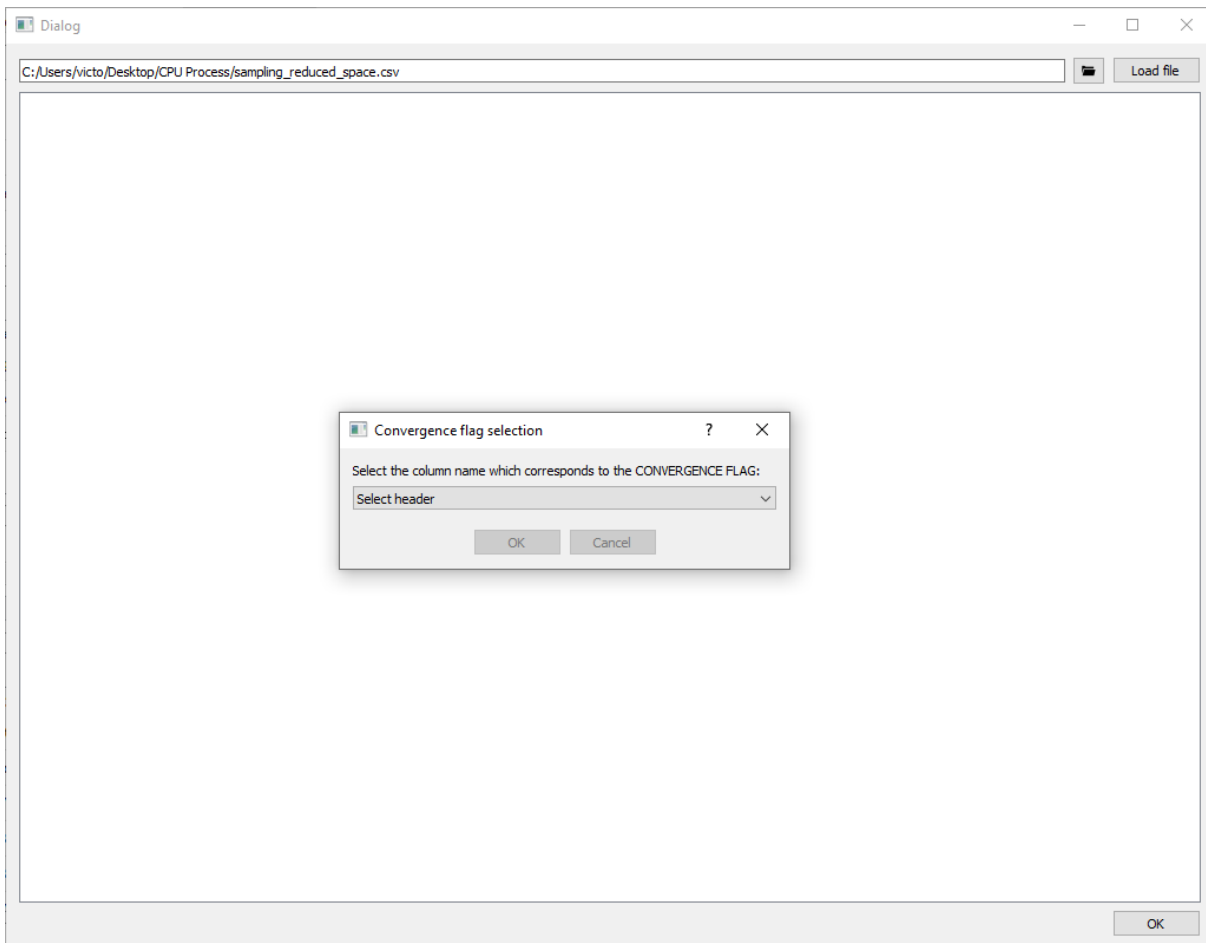


Figure 21 – Associating each alias created in *Metacontrol* to each column of the *.csv data.

	status	xco2	fluegas	mccp	mcct	flt	f2t	mcctout	mcppout	fltout	f2tout	s8t	fco2out	fco2in
0	ok	mccp	717.3752064	29.9997986	25	-30	-55	25	29.9998015	-30	-55	-55.8745660	600.5789880	621.9591060
1	ok	mcct	717.2142962	30.0100542	25	-30	-55	25	30.0100571	-30	-55	-55.8567178	600.5101880	621.8488290
2	ok	flt	716.8474607	29.9876505	25	-30	-55	25	29.9876534	-30	-55	-55.8957245	600.1672120	621.5408260
3	ok	f2t	717.4417649	30.0380832	25	-30	-55	25	30.0380862	-30	-55	-55.8080830	600.7494710	622.0362110
4	ok	mcctout	717.4930376	30.0132530	25	-30	-55	25	30.0132560	-30	-55	-55.8511604	600.7323150	622.0778770
5	ok	mccpout	0.8241155	717.3574487	30.0629039	25	-30	-55	30.0629069	-30	-55	-55.7651662	600.7403370	621.9669000
6	ok	mcctout	0.8240231	716.8809101	29.9859683	25	-30	-55	29.9859713	-30	-55	-55.8986635	600.1542980	621.5444590
7	ok	fltout	0.8236419	717.0694893	29.9744622	25	-30	-55	29.9744651	-30	-55	-55.9187565	600.2301870	621.6696940
8	ok	f2tout	0.8241965	717.2345909	30.0489494	25	-30	-55	30.0489523	-30	-55	-55.7892747	600.6179180	621.8685100
9	ok	s8t	0.8244029	717.2827362	30.0600016	25	-30	-55	30.0600045	-30	-55	-55.7701710	600.7134100	621.9309590
10	ok	fco2out	0.8238345	716.9571878	30.0226309	25	-30	-55	30.0226338	-30	-55	-55.8348787	600.2731940	621.5916750
11	ok	fco2in	0.8243741	717.0216489	30.0019252	25	-30	-55	30.0019281	-30	-55	-55.8708540	600.3596460	621.7016840
12	ok		0.8238086	717.1551182	30.0297985	25	-30	-55	30.0298015	-30	-55	-55.8224453	600.4512630	621.7606730
13	ok		0.8243198	717.1065279	30.0175508	25	-30	-55	30.0175538	-30	-55	-55.8436883	600.4579970	621.7698360
14	ok		0.8239824	717.1240973	30.0455173	25	-30	-55	30.0455202	-30	-55	-55.7952162	600.4862340	621.7512240
15	ok		0.8239162	716.9841997	30.0670225	25	-30	-55	30.0670254	-30	-55	-55.7580631	600.4077980	621.6232930
16	ok		0.8237137	716.9047673	30.0555131	25	-30	-55	30.0555160	-30	-55	-55.7779386	600.2856460	621.5340990
17	ok		0.8236648	717.4398718	29.9932650	25	-30	-55	29.9932680	-30	-55	-55.8859502	600.5860840	621.9931060
18	ok		0.8237651	717.3206409	30.0343753	25	-30	-55	30.0343783	-30	-55	-55.8145131	600.5937850	621.8998130
19	ok		0.8242450	717.5248326	29.9771723	25	-30	-55	29.9771752	-30	-55	-55.9140102	600.7061010	622.1250300

Figure 22 – “Differential data” input screen: Reduced space model training, Differentiation method, and gradient/hessian evaluation.

Load Simulation	Sampling	Metamodel	Optimization	Reduced Space	Differential Data	Self-Optimizing Control
<div> <div> Reduced space metamodel training Open training dialog </div> <div> Differentiation method <input checked="" type="radio"/> Aproximate analytical (Kriging predictions equations) <input type="radio"/> Numeric (numdifftools) <input type="radio"/> Automatic (autograd) </div> <div> Gradient and Hessian Estimate Gradient and Hessian </div> </div>						

Table 6 – Best Self-Optimizing Control variables found by *Metacontrol* for a single measurement policy.

CV Candidate <i>alias</i>	Worst-Case Loss (kWh/tCO_2)	Average-Case Loss (kWh/tCO_2)
mccpout	0.0097	0.0011
s8t	0.0125	0.0014
xco2out	0.0458	0.0051
co2rrcv	0.0549	0.0061
fco2out	15.5916	1.7324

Note: Description for the variables aliases present in [Table 1](#).

Dialog

Metamodel Construction

Hyperparameters (theta) definition

Variable	Lower Bound	Upper Bound	Estimate
fluegas	1e-12	10.0	1.0
xco2	1e-12	10.0	1.0
mccp	1e-12	10.0	1.0

Regression and Correlation (Kernel) model selection

Polynomial (regression) model:

Constant (0th order)

Correlation (kernel) model:

Exponential Gaussian (Kriging)

Choose which type of validation to perform

☒ K-Folds

5 Folds

☐ Hold out

75 % Training set

Results generation and visualization

Generate metamodel

View graphical results

Choose which variables will have their model built

Select	Alias	Type
<input type="checkbox"/>	mcctout	Candidate (CV)
<input checked="" type="checkbox"/>	mccpout	Candidate (CV)
<input type="checkbox"/>	f1tout	Candidate (CV)
<input type="checkbox"/>	f2tout	Candidate (CV)
<input checked="" type="checkbox"/>	s8t	Candidate (CV)
<input checked="" type="checkbox"/>	fco2out	Candidate (CV)

Validation metrics

	mccpout	s8t	fco2out	xco2out	j	co2rrcv
OMAE	7.699083373798506e-08	0.000154386392694524	3.117978268771064e-05	1.699926747515216e-08	7.268964041410679e-06	5.355384352127382e-07
OR2	0.999999993558971	0.9995572461553335	0.9999995823486124	0.9999923917293636	0.9999979549658612	0.9979871798452313
OEI	0.999999994161456	0.9997516592282111	0.9999996267857313	0.9999939124082131	0.9999980208705583	0.9980415900298409
Sample Mean	30.18462349999997	-55.48512216666667	600.8273821333333	0.9671256504666668	112.38042918041235	0.966287164047977
Sample Std	0.008576728391175913	0.023460542832869895	0.1743868203598919	1.4922128145214606e-05	0.013787958948383942	3.95878478777817e-05

OK

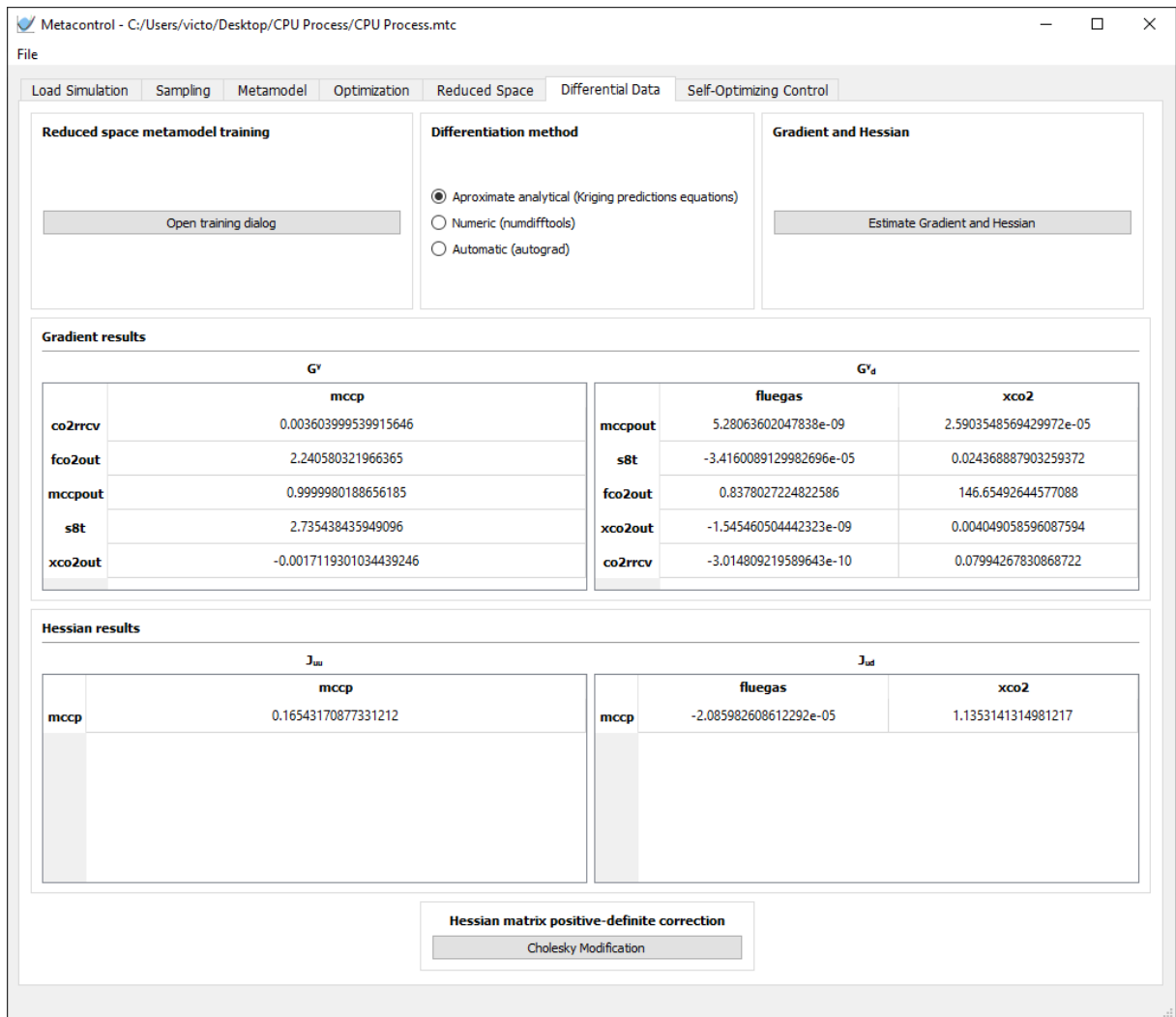
Figure 24 – Differential data estimated in *Metacontrol*

Figure 25 – Input screen in *Metacontrol* "Self-Optimizing Control " tab - CPU Process

Metacontrol - C:/Users/victo/Desktop/CPU Process/CPU Process.mtc

File

Load Simulation Sampling Metamodel Optimization Reduced Space Differential Data Self-Optimizing Control

Disturbances magnitude

	Value
fluegas	35.8595
xco2	0.05

Measurements error

	Value
co2rrcv	0.001
fco2out	0.01
mccpout	0.01
s8t	0.5
xco2out	0.001

Subsets sizing options

	Size 1	Size 2	Size 3	Size 4	Size 5
Subset number	5	10	10	5	1

Generate results

Figure 26 – Best control structure in worst-case loss ascending order, for subsets of size 1 (single measurement policy)

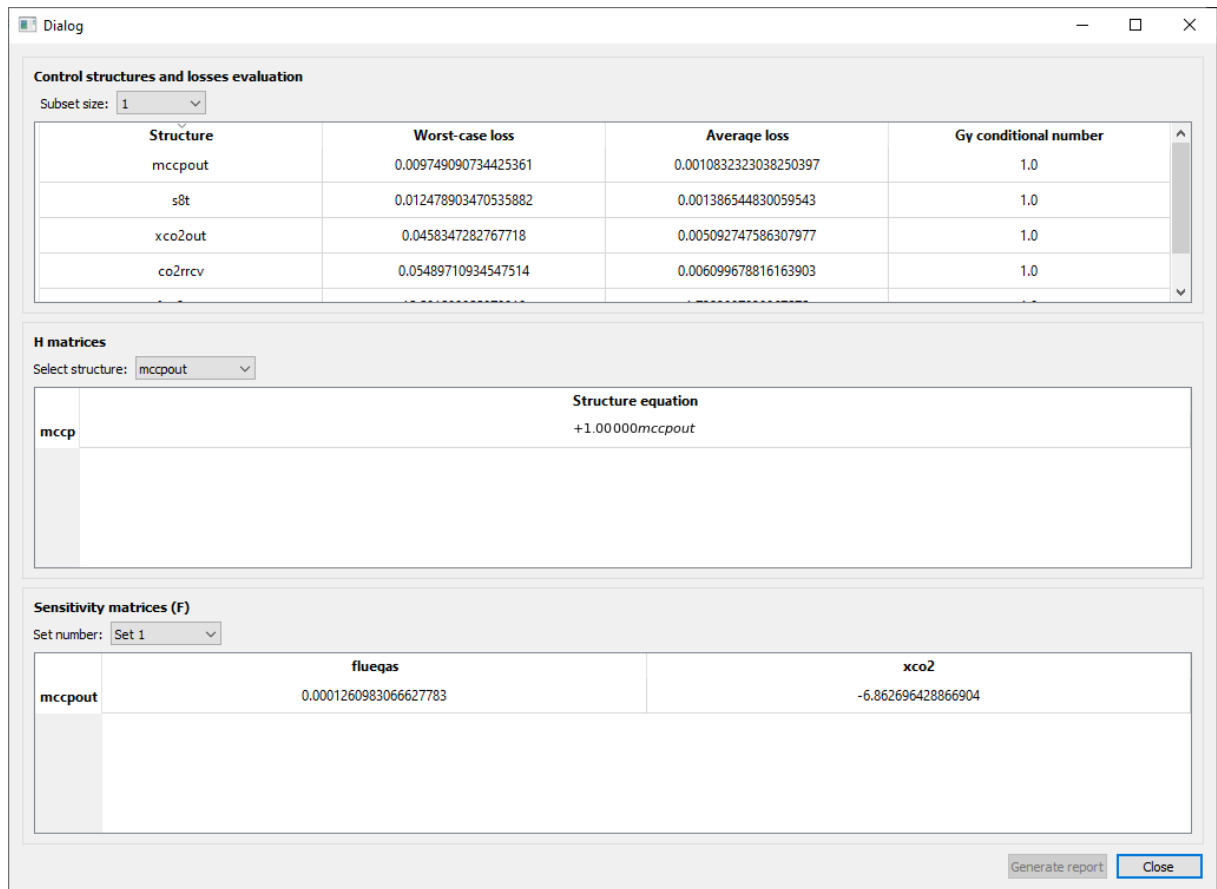
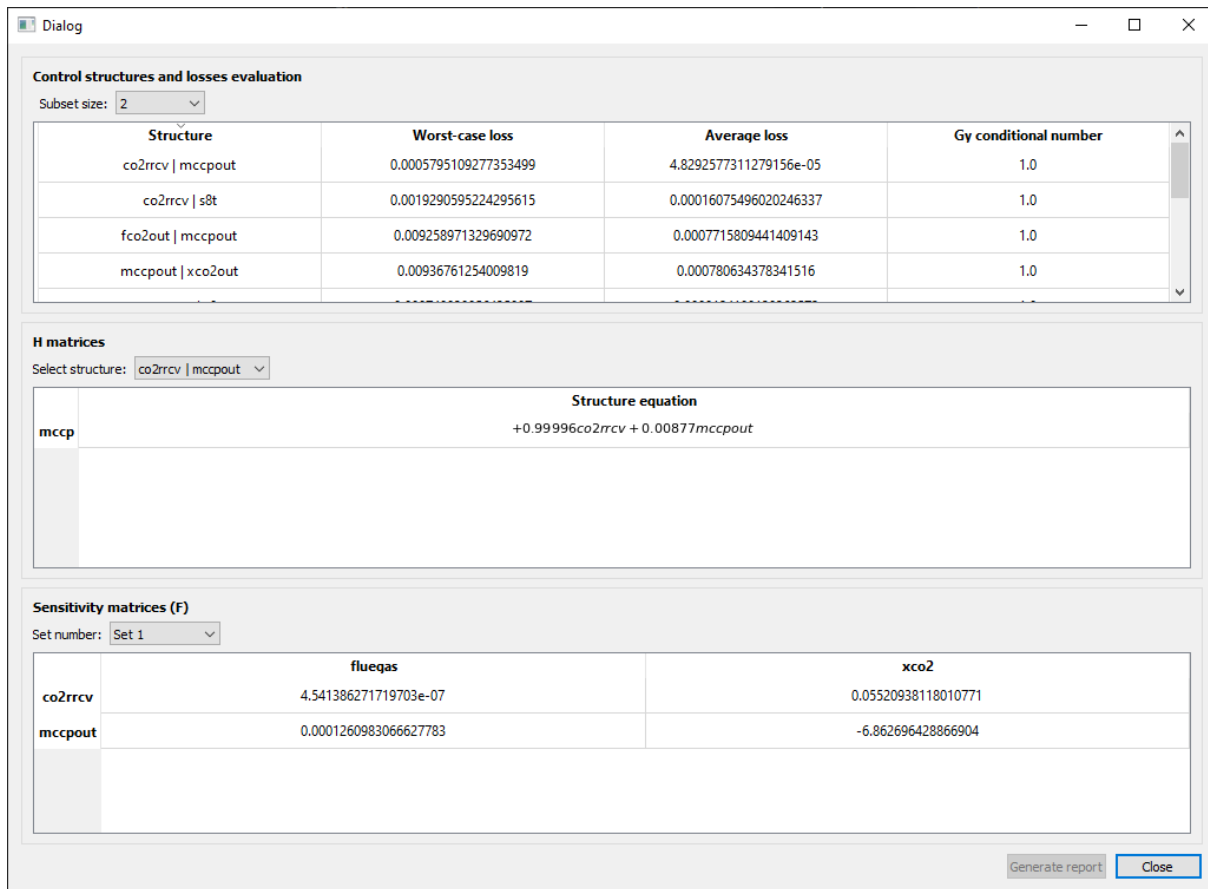


Figure 27 – Best control structure in worst-case loss ascending order, for subsets of size 2 (linear combinations of measurements)



6.2 Indirect composition control of a C3 Splitter column

The second example to be tested in *Metacontrol* consists is a particular case of Self-Optimizing Control structure selection: Indirect control. As stated before (HORI; SKOGESTAD; ALSTAD, 2005), indirect control consists when, for some reason, one is unable to control directly one or more variables (denominated by Hori, Skogestad, and Alstad (2005) as “primary” variables), and consequently tries to control a set of “secondary” variables, that will (hopefully) maintain the “primary” ones at their desired setpoints. As demonstrated in the past (HORI; SKOGESTAD; ALSTAD, 2005; ALSTAD; SKOGESTAD; HORI, 2009), the selection of a control structure for indirect control is a special case of the exact local method derived by Halvorsen et al. (2003) and that had an explicit solution derived by Alstad, Skogestad, and Hori (2009).

In order to use the exact local method for the special case of indirect control, the control strucutre designer must only impose a objective function that will minimize the error between the “primary” variables and their desired setpoints. In fact, this approach has been successfully done previously (HORI, Eduardo Shigueo; SKOGESTAD, 2008). As mentioned before, *Metacontrol* uses the exact local method with explicit solution from Alstad, Skogestad, and Hori (2009) to report to the user the best sets of controlled variables with Self-Optimizing Control properties.

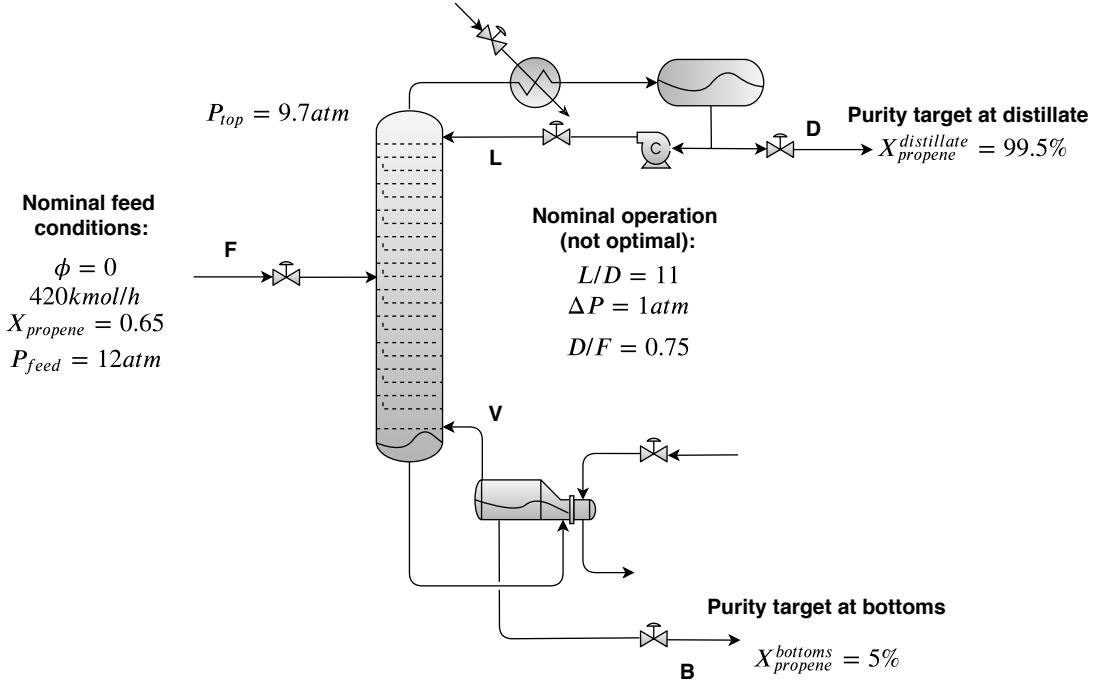
A didactic case of indirect control that happens on the industry consists in composition control of distillation columns. Due to unreliability and slow dynamics of online analyzers, the direct control of compositions becomes infeasible, or at least, very difficult to be done. On the other hand, the market demand and/or environmental legislation will, very often, impose purity levels to key components at the distillation columns. Therefore, there is a conflict between process instrumentation and market/environmental restrictions.

The case used as the second test-bed in *Metacontrol* derives directly from a previous work of the author (ALVES et al., 2018), which is a propylene-propane splitter, being depicted in Figure 28. The previous publication from Alves et al. (2018) showed that, from an economic plantwide control perspective, the best set of controlled variables to be chosen would be the composition of propene at the top stream and the composition of propene lost at the bottoms stream, as can be seen in Equation 6.5.

$$c = \begin{pmatrix} x_{top}^{propene} \\ x_{bottom}^{propene} \end{pmatrix} \quad (6.5)$$

Given the fact that the cited control structure uses two compositions measurements that are difficult to be directly controlled, *Metacontrol* will be used to find a Self-Optimizing Control structure for this problem. The objective function in question will be the relative steady-state deviation (HORI, Eduardo Shigueo; SKOGESTAD, 2008) from the nominal

Figure 28 – C3 Splitter Column Process flowsheet



optimal setpoint found by [Alves et al. \(2018\)](#), depicted in [Equation 6.6](#).

$$\Delta x^2 = \left(\frac{x_{\text{top}}^{\text{propene}} - x_{\text{setpoint}}^{\text{propene}}}{x_{\text{setpoint}}^{\text{propene}}} \right)^2 + \left(\frac{x_{\text{bottom}}^{\text{propene}} - x_{\text{setpoint}}^{\text{propene}}}{x_{\text{setpoint}}^{\text{propene}}} \right)^2 \quad (6.6)$$

The economically optimal values for the setpoints at the top and bottom streams of the C3 splitter are 0.995 (active constraint for economic plantwide control problem) and 0.05, respectively. For the latter, the setpoint was rounded to 0.05. The constraint that exists in this problem regards the reboiler duty, unable to surpass the limit of 80 GJ/h . The economically optimal values for the compositions and the reboiler duty constraint are taken from [Alves et al. \(2018\)](#).

- C-1: Reboiler duty $\leq 80 \text{ GJ/h}$

The main process disturbances considered are the same from [Alves et al. \(2018\)](#):

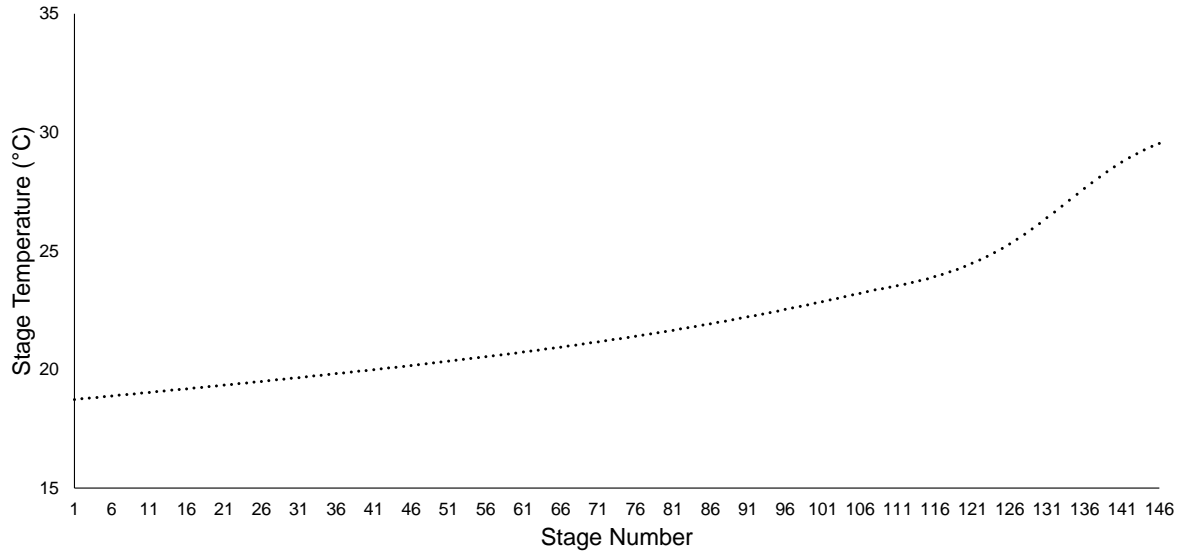
- D-1: Propylene flow rate
- D-2: Propane flow rate
- D-3: Feed vapor fraction ϕ

The number of degrees of freedom for this process is two ([ALVES et al., 2018](#); [SKOGESTAD, 2000](#)) and without loss of generality, they are the same from [Alves et al. \(2018\)](#):

1. Reflux ratio
2. Distillate to feed ratio

The distillation column studied has 146 stages. As [Alves et al. \(2018\)](#) did, to test the Self-Optimizing Control theory, some temperature measurements will be selected by the slope criterion ([LUYBEN, 2006](#)) as promising CV candidates and will also choose (based on the same criterion) poor candidates. To illustrate, the temperature profile for the C3 splitter is shown in [Figure 29](#). In addition, several flows and flow ratios were considered as CV candidates to be tested.

Figure 29 – C3 Splitter Column - Temperature profile.



There are 20 CV candidates and 2 degrees of freedom for this case study. For a single measurement policy, for illustration purposes, there are 190 possible control structures ([Equation 6.7](#)), being clear the impracticability of evaluating all of these control structures manually. As stated before in the previous example, the problem becomes even larger if one begins to consider linear combinations as CV candidates.

$$\binom{20!}{2!} = \frac{20!}{2! \times (20 - 2)!} = 190 \quad (6.7)$$

With all preliminary information emphasized so far, it is possible to use the first tab of *Metacontrol*. Similarly as the first case study, the objective function, process constraints and CV candidates can be created at the “Function definitions” panel, as can be seen in [Figure 30](#). [Figure 31](#) shows the variables being added to the *.mtc file, in order to be used for the study.

Table 7 – CV Candidates for C3 Splitter composition indirect control.

Variable (alias used in <i>Metacontrol</i>)	Description
bf	Bottoms to feed ratio
vf	Boilup to feed ratio
lf	Reflux to feed ratio
rrcv	Reflux ratio
dfcv	Distillate to feed ratio
l	Reflux rate ($kmol/h$)
v	Boilup rate ($kmol/h$)
t8	Stage 8 temperature ($^{\circ}C$)
t9	Stage 9 temperature ($^{\circ}C$)
t10	Stage 10 temperature ($^{\circ}C$)
t11	Stage 11 temperature ($^{\circ}C$)
t12	Stage 12 temperature ($^{\circ}C$)
t129	Stage 129 temperature ($^{\circ}C$)
t130	Stage 130 temperature ($^{\circ}C$)
t131	Stage 131 temperature ($^{\circ}C$)
t132	Stage 132 temperature ($^{\circ}C$)
t133	Stage 133 temperature ($^{\circ}C$)
t134	Stage 134 temperature ($^{\circ}C$)
t135	Stage 135 temperature ($^{\circ}C$)
t136	Stage 136 temperature ($^{\circ}C$)

For this case study, only 60 initial points were sampled. These points were refined by the algorithm from [Caballero and Grossmann \(2008\)](#) implemented in *Metacontrol* in order to find the optimal nominal operating point. Using a K -fold validation metric, one of the features that are implemented in *Metacontrol*, it was found that using the quadratic regression polynomial yielded the most desirable metrics, as can be seen in [Figures 32 and 34](#). This is a valuable feature: it systematically informs, for the problem being studied, which regression model will yield the most accurate results.

The optimization results found are described in [Figure 35](#). There are no active constraints for this problem, and therefore there are 2 unconstrained degrees of freedom left for self-optimizing control purposes. The values for the decision variables are virtually the same ones found by [Alves et al. \(2018\)](#). This was expected since the direct control structure proposed came from an economic plantwide structure proposal (controlling propene distillate and bottoms compositions). The difference between the optimal decision variables values found previously by [Alves et al. \(2018\)](#) and now, can be associated to the rounded setpoint of the composition of propene at the bottom stream.

An optimization was performed using the process simulator internal optimizer, to make the reader able to compare with the optimization using the surrogate model with the refinement algorithm from [Caballero and Grossmann \(2008\)](#). As in the first case study, the results are nearly identical ([Table 8](#)).

Figure 30 – C3 Splitter Column Process - loading simulation. Process constraint “c2” is multiplied by 4.184 to convert simulation reboiler duty from $GCal/h$ to GJ/h .

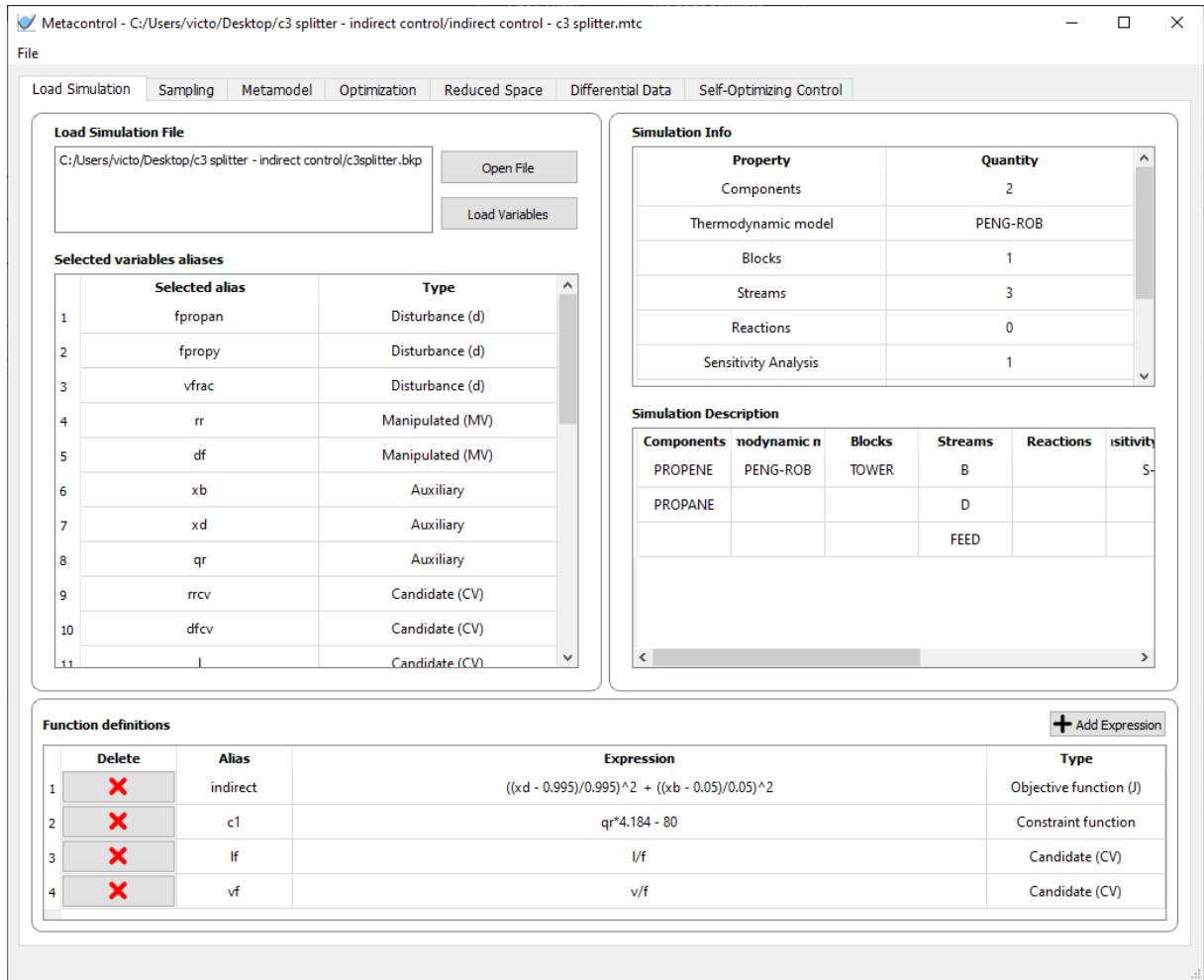


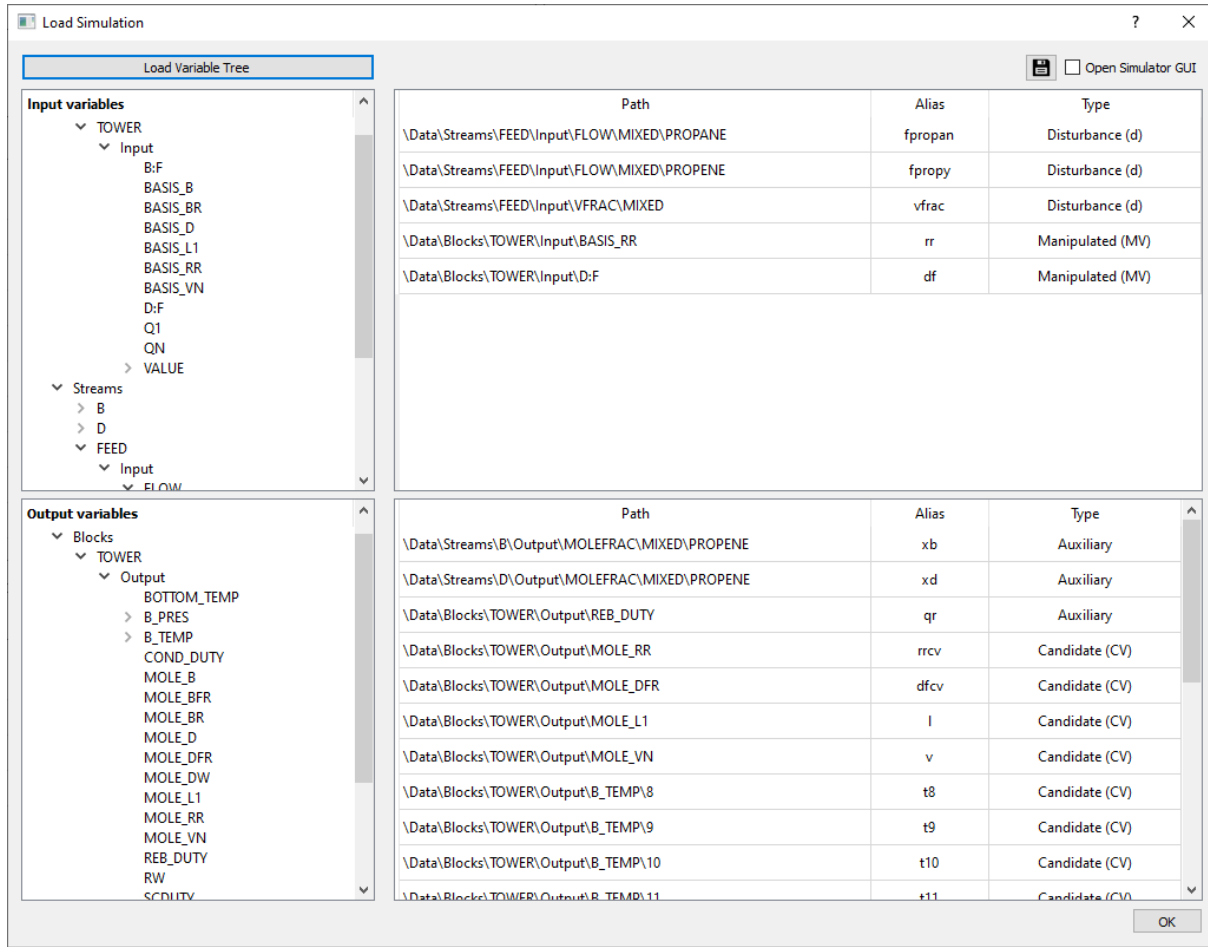
Table 8 – Optimization runs: *Aspen Plus* vs *Metacontrol* - Decision variables and objective function - C3 Splitter Indirect control

	Objective function “indirect”	Reflux Ratio	Distillate to feed ratio
Aspen Plus	7.47×10^{-15}	13.5246	0.6349
Metacontrol	5.92×10^{-10}	13.5159	0.6349

As stated before, 20 CV candidates were chosen to be tested by *Metacontrol*. Differently from the first case study, where it was used the *.csv import feature (merely to show the capability of the software), on this example the reduced space problem sampling was done internally in *Metacontrol*. Since there are no active constraints, the same *.bkp file can be used (i.e. It is not necessary to implement any design specifications to consume the degrees of freedom for active constraints).

Figure 36 shows the interface built to select the reduced space problem simulation file, where the user can simply point to the *.bkp file using the GUI and the sampling assistant will use the limits that were imposed under the “Range of disturbances” panel in

Figure 31 – C3 Splitter Column Process - loading variables from Aspen Plus.



order to generate the data for the reduced space problem. The latter can be inspected in Figure 38.

Using the sampled data from the simulation file, the user can go to “Differential Data” tab and generate the gradients and the hessians, exactly as done at the first case study. Figure 39 shows the gradients G^y, G_d^y and the hessians J_{uu}, J_{ud} calculated using *Metacontrol*.

Similarly to the first case study, the gradients obtained by *Metacontrol* were compared against the ones generated by the process simulator. Not surprisingly, they were virtually identical, which is an evidence of the robustness of the previously proposed methodology Alves et al. (2018) that is embedded in the *Metacontrol* software.

Figure 32 – K-fold validation metric for constant (poly0) regression model.

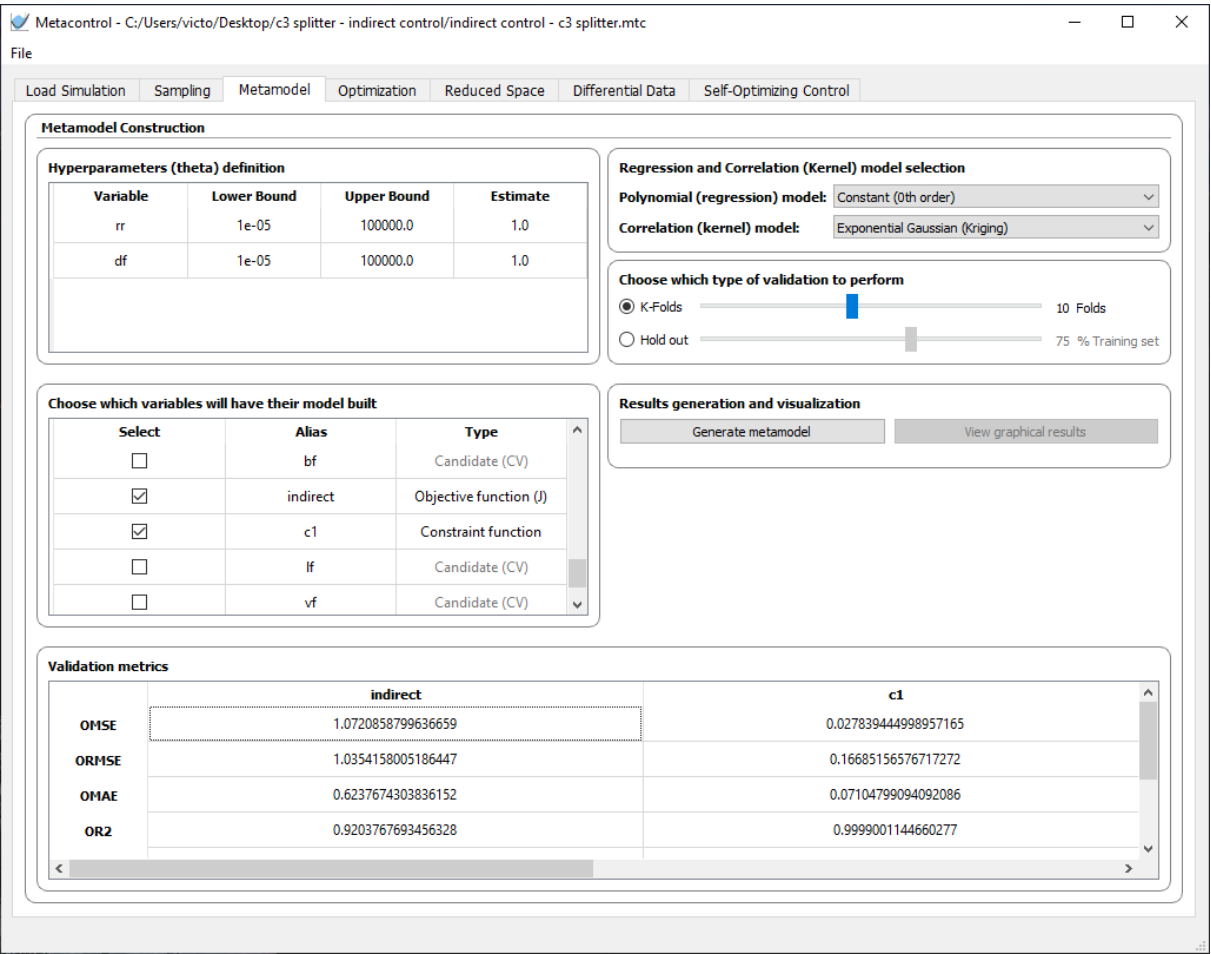


Figure 33 – K-fold validation metric for linear (poly1) regression model.

Metacontrol - C:/Users/victo/Desktop/c3 splitter - indirect control/indirect control - c3 splitter.mtc

File

Load Simulation Sampling **Metamodel** Optimization Reduced Space Differential Data Self-Optimizing Control

Metamodel Construction

Hyperparameters (theta) definition

Variable	Lower Bound	Upper Bound	Estimate
rr	1e-05	100000.0	1.0
df	1e-05	100000.0	1.0

Regression and Correlation (Kernel) model selection

Polynomial (regression) model: Linear (1st order) ▼

Correlation (kernel) model: Exponential Gaussian (Kriging) ▼

Choose which type of validation to perform

☒ K-Folds 10 Folds

☐ Hold out 75 % Training set

Choose which variables will have their model built

Select	Alias	Type
<input type="checkbox"/>	bf	Candidate (CV)
<input checked="" type="checkbox"/>	indirect	Objective function (J)
<input checked="" type="checkbox"/>	c1	Constraint function
<input type="checkbox"/>	lf	Candidate (CV)
<input type="checkbox"/>	vf	Candidate (CV)

Results generation and visualization

Generate metamodel View graphical results

Validation metrics

	indirect	c1
OMSE	0.42958260951538574	0.01728828790505161
ORMSE	0.6554255179006885	0.13148493413715356
OMAE	0.42842736403688575	0.048979662225269824
OR2	0.9355323953196946	0.9999425996258667

Figure 34 – K-fold validation metric for quadratic (poly2) regression model.

Metacontrol - C:/Users/victo/Desktop/c3 splitter - indirect control/indirect control - c3 splitter.mtc

File

Load Simulation Sampling **Metamodel** Optimization Reduced Space Differential Data Self-Optimizing Control

Metamodel Construction

Hyperparameters (theta) definition

Variable	Lower Bound	Upper Bound	Estimate
rr	1e-05	100000.0	1.0
df	1e-05	100000.0	1.0

Regression and Correlation (Kernel) model selection

Polynomial (regression) model: Quadratic (2nd order)

Correlation (kernel) model: Exponential Gaussian (Kriging)

Choose which type of validation to perform

☒ K-Folds 10 Folds

☐ Hold out 75 % Training set

Choose which variables will have their model built

Select	Alias	Type
<input type="checkbox"/>	bf	Candidate (CV)
<input checked="" type="checkbox"/>	indirect	Objective function (J)
<input checked="" type="checkbox"/>	c1	Constraint function
<input type="checkbox"/>	lf	Candidate (CV)
<input type="checkbox"/>	vf	Candidate (CV)

Results generation and visualization

Generate metamodel View graphical results

Validation metrics

	indirect	c1
OMSE	0.2987788527754601	2.50350932753648e-06
ORMSE	0.5466066709942902	0.0015822481877178687
OMAE	0.362119359032828	0.0009642026030455646
OR2	0.9901886159736323	0.9999999883847966

Figure 35 – Indirect control index objective function being minimized using surrogate metamodel.

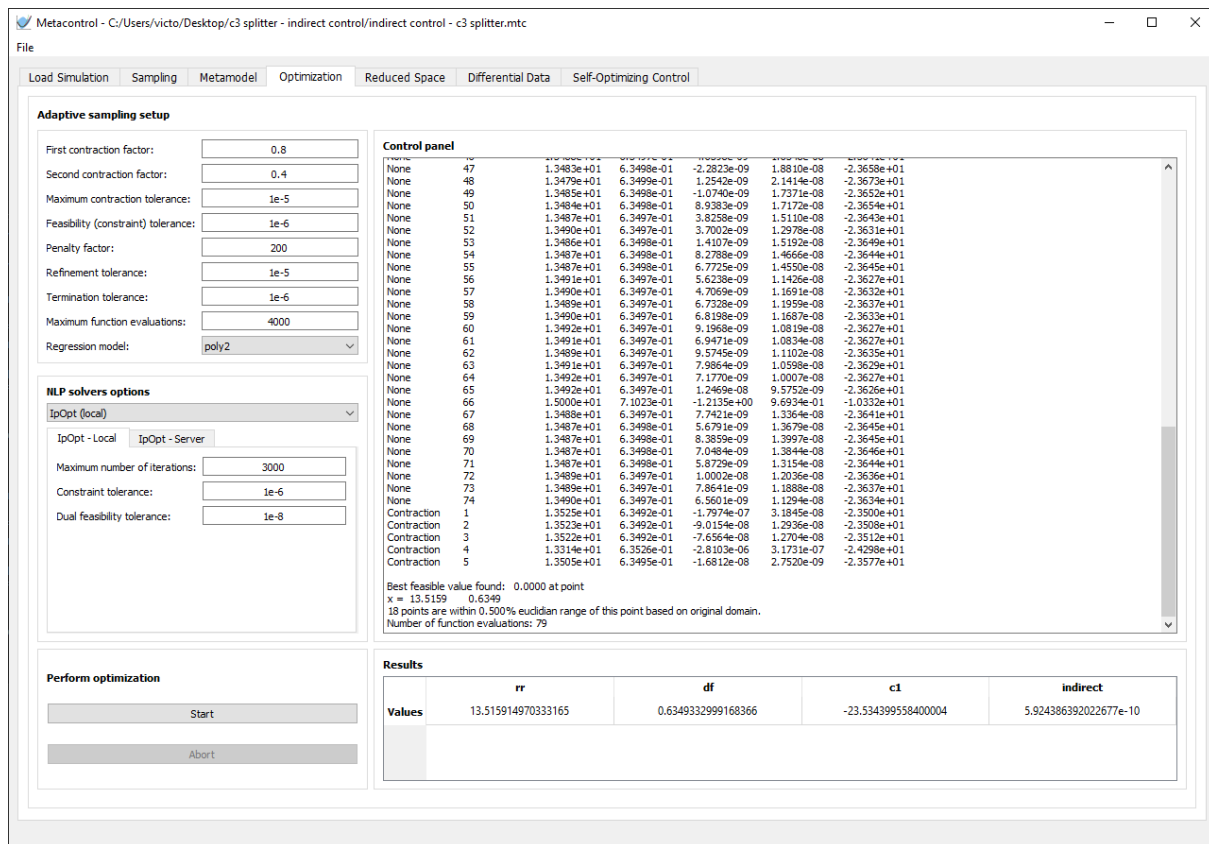


Figure 36 – Reduced space problem - sampling using a *.bkp file.

Metacontrol - C:/Users/victo/Desktop/c3 splitter - indirect control/indirect control - c3 splitter.mtc

File

Load Simulation Sampling Metamodel Optimization **Reduced Space** Differential Data Self-Optimizing Control

Variable Activity

	rr	df	rrcv	dfcv	l	v	t8	t9	t10	t11	t12	t129	t130	t131	t132	t133	t134	t135	t136	bf	lf	vf
Active	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pairing			Sele...	Sele...	Sele...	Sele...	Sele...	Sele...	Sele...	Sele...	Sele...	Sele...	Sele...	Sele...	Sele...	Sele...	Sele...	Sele...	Sele...	Sele...	Sele...	Sele...
Type	Man...	Man...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...
Value	13.5...	0.63...																				

Range of Disturbances

Disturbance variable	Lower bound	Upper bound	Nominal Value
fpropan	146.0	148.0	147.0
fpropy	272.0	274.0	273.0
vfrac	0.0	0.001	0.0
rr	13.4	13.6	13.515914970333165

Data Source

☒ Simulation file C:/Users/victo/Desktop/c3 splitter - indirect control/c3splitter.bkp

☐ CSV file

Reduced Space Sampled Data

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	Cas...	Sta...	Inp...		Out...																		
2			rr	df	rrcv	dfcv	l	v	t8	t9	t10	t11	t12	t129	t130	t131	t132	t133	t134	t135	t136	bf	lf
3	1	ok	13.5...	0...	13.5...	0...	360...	390...	18.9...	18.9...	19.0...	19.0...	19.0...	26.1...	26.3...	26.6...	26.8...	27.1...	27.3...	27.6...	27.8...	0...	8...
4	2	ok	13.4...	0...	13.4...	0...	358...	388...	18.9...	18.9...	19.0...	19.0...	19.0...	26.1...	26.3...	26.6...	26.8...	27.1...	27.3...	27.6...	27.8...	0...	8...
5	3	ok	13.4...	0...	13.4...	0...	360...	390...	18.9...	18.9...	19.0...	19.0...	19.0...	26.1...	26.4...	26.6...	26.8...	27.1...	27.3...	27.6...	27.8...	0...	8...

Figure 37 – Reduced space problem - pointing the *.bkp file location.

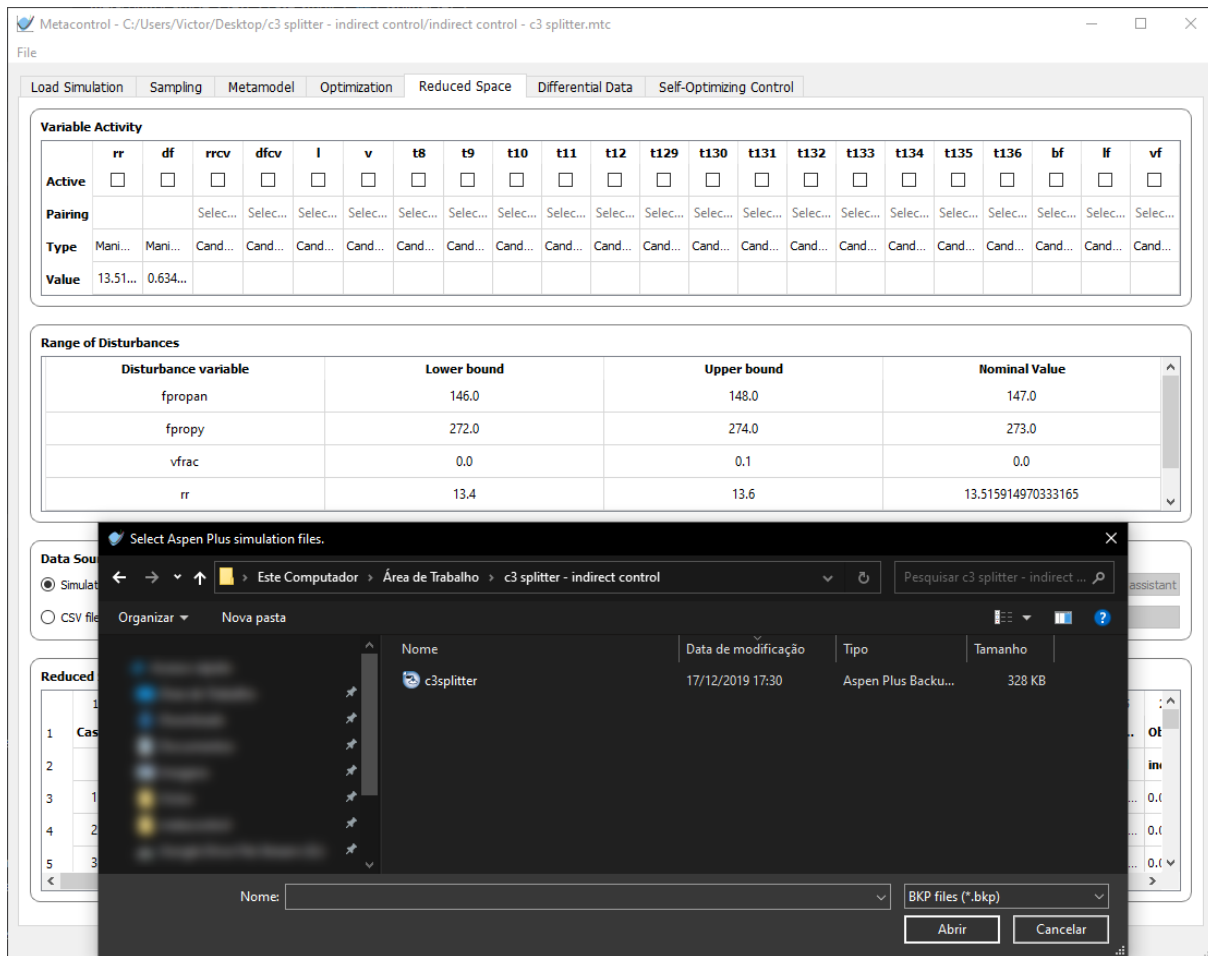


Figure 38 – Reduced space problem - Sampling assistant: Identical to the Sampling Assistant that exists under the “Sampling” tab, in order to keep consistency of interface across *Metacontrol*. Number 1 indicates the button to open the Assistant, 2 consists in the main screen, 3 is the button that opens the settings of the sampling technique that will generate the input data; 4 generates the data. In addition, number 5 depicts the control of the sampling procedure: Sample data, cancel, close screen (“Done” button). Lastly, the user can abort the sampling at any time using the “Abort” button (number 6) or export the design of experiments as a *.csv (number 7).

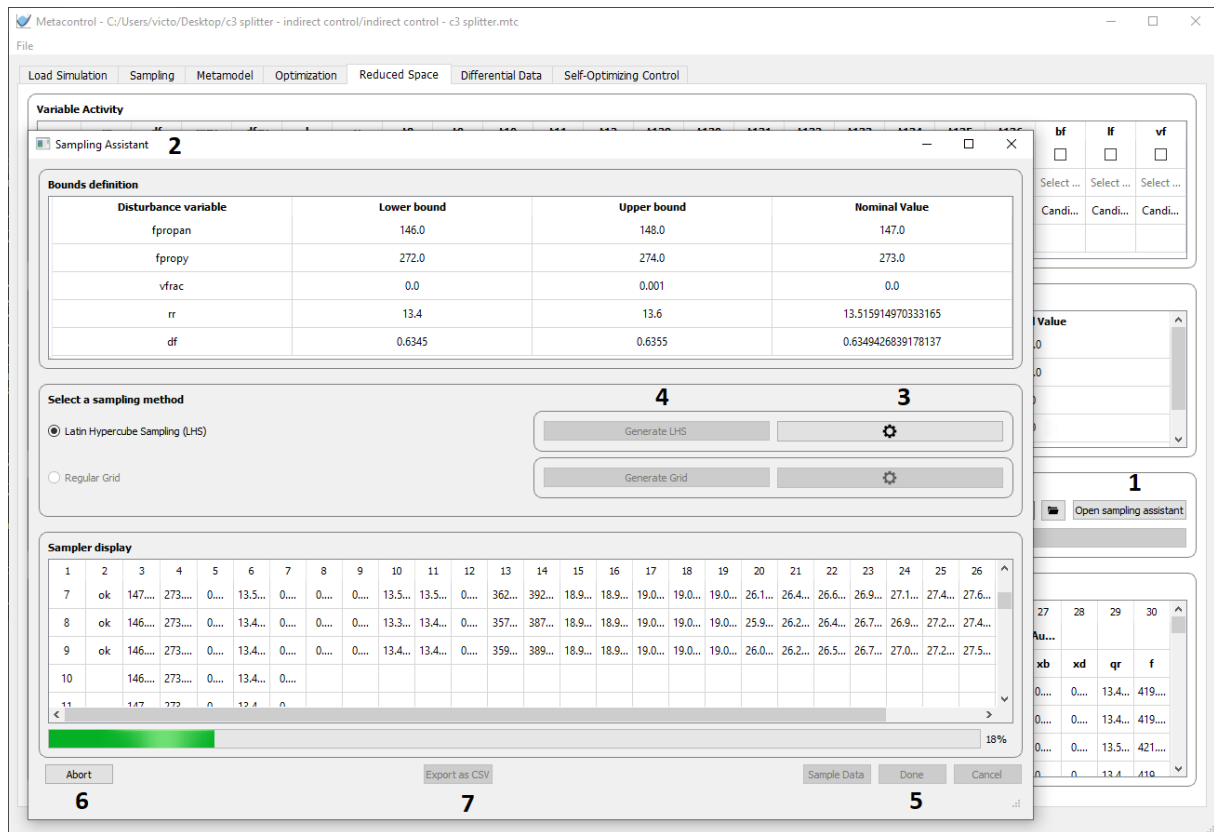


Figure 39 – “Differential Data” tab - C3 Splitter indirect control.

Metacontrol - C:/Users/Victor/Desktop/c3 splitter - indirect control/indirect control - c3 splitter.mtc

File

Load Simulation Sampling Metamodel Optimization Reduced Space Differential Data Self-Optimizing Control

Reduced space metamodel training

Open training dialog

Differentiation method

☒ Approximate analytical (Kriging predictions equations)

☐ Numeric (numdiff/fools)

☐ Automatic (autograd)

Gradient and Hessian

Estimate Gradient and Hessian

Gradient results

G'

	rr	df
rrcv	1.0000000659688437	-6.016626128830726e-07
dfcv	5.8224506514232136e-09	1.0000000064778083
l	266.6719608586177	5676.6818711050055
v	270.45839614783404	6086.935860127169
ts	-0.01764493328227118	0.2686594589791405

G'_d

	fpropan	fpropy	vfrac
rrcv	1.5006001852184657e-08	-1.7690262049478601e-09	-9.403592256628808e-09
dfcv	-6.818247074084155e-11	1.363532578303847e-10	-2.995086098128421e-08
l	8.581712894408655	8.581710560862737	7.550223831587095e-05
v	9.153598792160464	9.370186996338427	-406.2084406512939
ts	0.0006752757354098762	-0.0003669134589952631	0.0038650595160064174

Hessian results

J_{rr}

	rr	df
rr	0.0114197075315572	7.640438166009123
df	7.640438166009123	5126.867988240707

J_{rd}

	fpropan	fpropy	vfrac
rr	0.01235820754969243	-0.006672185525298593	-0.002437001447956368
df	8.297687727781545	-4.467318414009255	-1.6427849408197508

Hessian matrix positive-definite correction

Cholesky Modification

Table 9 – High-order data obtainment: *Aspen Plus* vs *Metacontrol* - C3 Splitter column case study.

	G^y		G_d^y		
<i>Metacontrol</i>	1.0000	-1.6084×10^{-6}	-3.2986×10^{-9}	1.2500×10^{-9}	-9.5679×10^{-8}
	1.5380×10^{-10}	1.0000	-6.1220×10^{-11}	4.5431×10^{-12}	9.7279×10^{-10}
	266.6720	5676.6833	8.5817	8.5817	-2.6062×10^{-5}
	270.4588	6086.9436	9.1536	9.3702	-406.2089
	-0.0177	0.2700	0.0007	-0.0004	0.0038
	-0.0189	0.2894	0.0007	-0.0004	0.0041
	-0.0201	0.3094	0.0008	-0.0004	0.0044
	-0.0213	0.3299	0.0008	-0.0004	0.0047
	-0.0226	0.3511	0.0009	-0.0005	0.0050
	0.0527	74.2428	0.1257	-0.0678	0.0586
	0.0613	76.6816	0.1291	-0.0697	0.0507
	0.0691	78.3864	0.1314	-0.0709	0.0429
	0.0759	79.2567	0.1323	-0.0713	0.0351
	0.0813	79.2233	0.1317	-0.0710	0.0276
	0.0853	78.2564	0.1297	-0.0699	0.0205
	0.0877	76.3710	0.1261	-0.0680	0.0139
	0.0891	73.6324	0.1214	-0.0654	0.0104
	-1.5380×10^{-10}	-1.0000	6.1220×10^{-11}	-4.5431×10^{-12}	-9.7279×10^{-10}
	0.6349	13.5159	-5.0048×10^{-10}	-1.1641×10^{-9}	-7.1020×10^{-8}
	0.6439	14.4928	-0.0003	0.0002	-0.9672

Table 9 – (continued)

G^y			G_d^y		
Aspen Plus	1	0	0	0	0
	8.2510×10^{-19}	1	1.5116×10^{-7}	1.5116×10^{-7}	0
	266.6720	5676.6840	8.5817	8.5817	0
	270.4590	6086.9530	9.1536	9.3701	-406.2095
	-0.0177	0.2689	0.0007	-0.0004	0.0038
	-0.0188	0.2882	0.0007	-0.0004	0.0041
	-0.0200	0.3081	0.0008	-0.0004	0.0044
	-0.0213	0.3286	0.0008	-0.0004	0.0047
	-0.0225	0.3497	0.0009	-0.0005	0.0050
	0.0561	74.1792	0.1261	-0.0679	0.0650
	0.0644	76.6318	0.1295	-0.0697	0.0567
	0.0718	78.3508	0.1318	-0.0709	0.0483
	0.0781	79.2344	0.1326	-0.0714	0.0398
	0.0831	79.2129	0.1320	-0.0711	0.0316
	0.0867	78.2563	0.1299	-0.0700	0.0238
	0.0887	76.3785	0.1264	-0.0680	0.0166
	0.0891	73.6377	0.1214	-0.0654	0.0100
	-4.7860×10^{-17}	-1	8.6912×10^{-8}	8.6912×10^{-8}	-4.3560×10^{-17}
	0.6349	13.5159	2.0431×10^{-6}	2.0431×10^{-6}	0
	0.6439	14.4927	-0.0003	0.0002	-0.9672

Table 10 – Mean-squared error of high-order data obtainment: *Aspen Plus* vs *Metacontrol* - C3 Splitter column

	G^y	G_d^y
Mean-squared error	1.6738×10^{-5}	3.1706×10^{-8}

From Table 9 it is shown that the gradients generated using the metamodels built by *Metacontrol* are extremely close to the evaluation of the gradients directly using the nonlinear model from the process simulator.

Since all required high-order data it is available, the user can go to the last step of the top-down procedure, which consists in the loss evaluation for the control structures with Self-Optimizing Control properties. The last required information consists of the disturbances and measurement error matrices. Similarly as Alves et al. (2018), 10% of the nominal feed component flow rates were considered as the expect magnitudes, and for the feed vapor fraction it was also considered a 10% disturbance magnitude (Equation 6.8). Regarding the measurement errors, for flows and flow ratios, it was considered a 0.001 magnitude representing the accuracy of flow meters. For temperature measurements, 0.5°C, a value that can realistically represent thermocouples and RTD sensors accuracies (both are typically used in the industry, specially

distillation), was used. The order of Equations 6.8 and 6.9 are the same from the *Metacontrol* user interface: An alphabetical order of the aliases given at the first tab. The insertion of all information regarding the magnitude matrices can be seen in Figure 40.

$$W_d = \text{diag}(14.7, 27.3, 0.1) \quad (6.8)$$

$$W_n^y = \text{diag}(0.001, 0.001, 0.001, 0.001, 0.001, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.001, 0.001) \quad (6.9)$$

Figure 40 – Input screen in *Metacontrol* “Self-Optimizing Control” tab - C3 Splitter column: Here, all 190 possible control structures for a single measurement policy were considered to be evaluated by *Metacontrol*. For linear combinations of measurements as CV candidates, the 50 best ones of each possible subset size were evaluated, when possible. For subset sizes of 19 and 20, all combinations were considered (20 and 1, respectively).

The screenshot shows the Metacontrol software interface with the following components:

- Disturbances magnitude table:**

	Value
fpropan	14.7
fpropy	27.3
vfirac	0.1
- Measurements error table:**

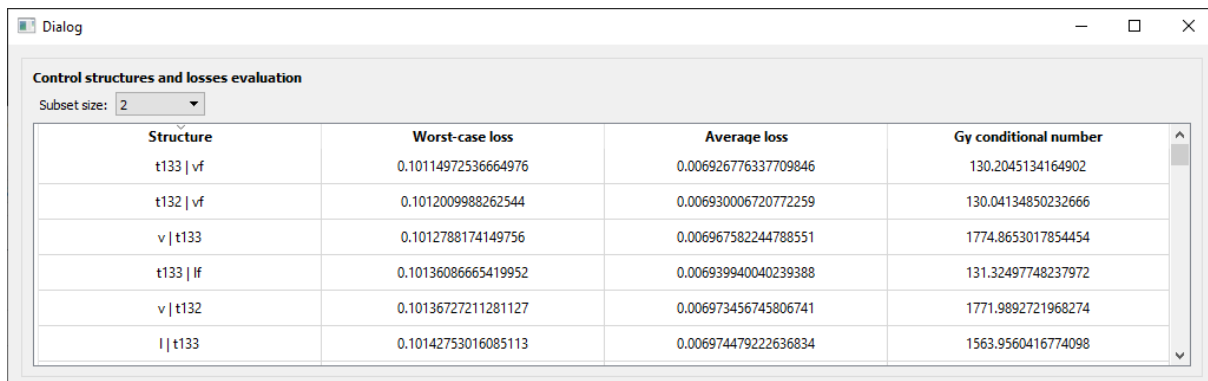
	Value
t132	0.5
t133	0.5
t134	0.5
t135	0.5
t136	0.5
t8	0.5
t9	0.5
v	0.001
vf	0.001
- Subsets sizing options table:**

	Size 2	Size 3	Size 4	Size 5	Size 6	Size 7	Size 8	Size 9	Size 10	Size 11	Size 12	Size 13	Size 14	Size 15	Size 16	Size 17	Size 18	Size 19	Size 20
Subset number	190	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	20	1
- Generate results button:** A button at the bottom right of the interface.

Clicking on “Generate results” button, the user can inspect the number of best control structures that he entered in the previous screen. Analyzing Figure 41, one can easily see that controlling sensitive temperatures associated together with flows and flow ratios generates a control structure capable of indirect controlling both distillate and bottom streams compositions with a small incurred loss. On the other hand, stages with small temperatures deviation between

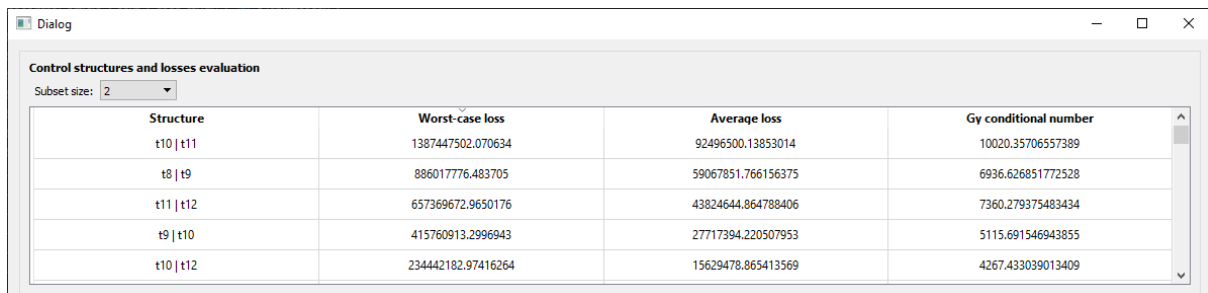
them gave unacceptable losses, similarly as found by different authors [Alves et al. \(2018\)](#) and [Eduardo S. Hori and Skogestad \(2007\)](#), for instance. The latter result can be found in [Figure 42](#). More generally, this result is also a confirmation that the slope criterion from [Luyben \(2006\)](#) it is a good starting assumption when one it is deciding which variable should be controlled. The main difference when one is using Self-Optimizing Control is that the mathematical formulation derived by the author of the technology already translated desired robust control and near-optimal operation from a heuristic and qualitative perspective to a mathematical one, making the whole procedure systematic.

Figure 41 – Best control structures for single measurement policy: Stages with significant temperature deviation between them associated with flow and flow ratios - namely boilup, reflux, boilup to feed ratio and reflux to feed ratio.



Structure	Worst-case loss	Average loss	Gy conditional number
t133 vf	0.10114972536664976	0.006926776337709846	130.2045134164902
t132 vf	0.1012009988262544	0.006930006720772259	130.04134850232666
v t133	0.1012788174149756	0.006967582244788551	1774.8653017854454
t133 lf	0.10136086665419952	0.006939940040239388	131.32497748237972
v t132	0.10136727211281127	0.006973456745806741	1771.9892721968274
l t133	0.10142753016085113	0.006974479222636834	1563.9560416774098

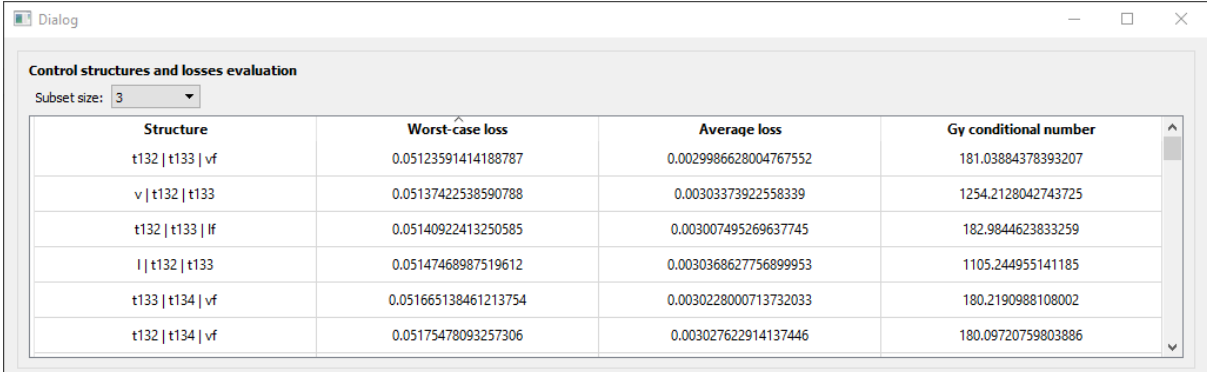
Figure 42 – Worst control structures for single measurement policy using exclusively temperature measurements: Stages with small temperature deviation between them. One can easily note that the inspection of the best and worst control structures is simple in *Metacontrol*: The user is capable of sorting, using the graphical user interface built, the control structures in ascending or descending order of worst-case loss, average-case loss and conditional number.



Structure	Worst-case loss	Average loss	Gy conditional number
t10 t11	1387447502.070634	92496500.13853014	10020.35706557389
t8 t9	886017776.483705	59067851.766156375	6936.626851772528
t11 t12	657369672.9650176	43824644.864788406	7360.279375483434
t9 t10	415760913.2996943	27717394.220507953	5115.691546943855
t10 t12	234442182.97416264	15629478.865413569	4267.433039013409

The analysis of the incurred loss when one is using linear combinations of measurements as CV candidates can be inspected in [Figures 43 and 45](#), and the usage of all measurements is depicted in [Figure 46](#). As stated by [Kariwala, Cao, and Janardhanan \(2008\)](#), the usage of all available measurements it is often not necessary. Actually, a good tradeoff between the number of measurements used and the value of the loss generally exists in most cases. For instance, the worst-case loss when one uses all 20 measurements is approximately 0.0126 ([Figure 46](#)), while using a simpler combination of 9 measurements gives a worst-case loss of 0.0141, only approximately 11.9% higher, but with less measurements forming the linear combination.

Figure 43 – Best control structures using linear combinations of measurements as CV candidates - Subset of size 3.



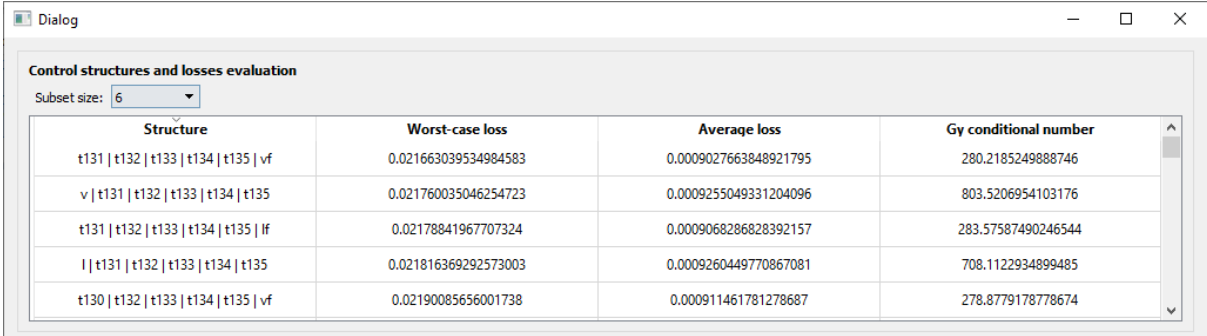
Dialog

Control structures and losses evaluation

Subset size: 3

Structure	Worst-case loss	Average loss	Gy conditional number
t132 t133 vf	0.05123591414188787	0.0029986628004767552	181.03884378393207
v t132 t133	0.05137422538590788	0.00303373922558339	1254.2128042743725
t132 t133 lf	0.05140922413250585	0.003007495269637745	182.9844623833259
l t132 t133	0.05147468987519612	0.0030368627756899953	1105.244955141185
t133 t134 vf	0.051665138461213754	0.0030228000713732033	180.2190988108002
t132 t134 vf	0.05175478093257306	0.003027622914137446	180.09720759803886

Figure 44 – Best control structures using linear combinations of measurements as CV candidates - Subset of size 6.



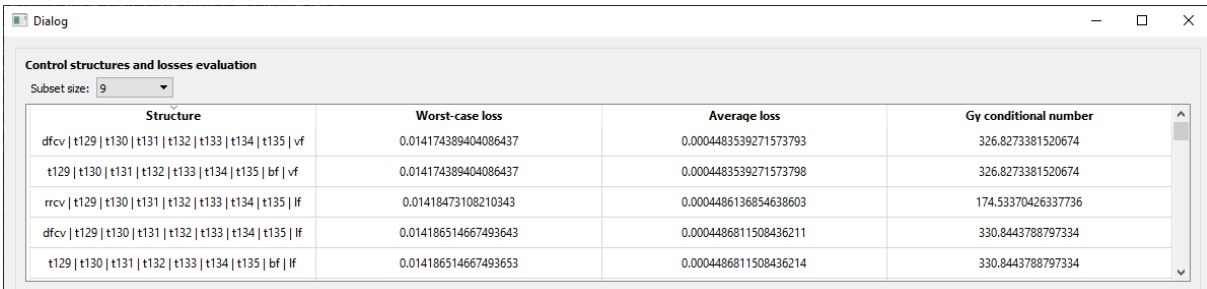
Dialog

Control structures and losses evaluation

Subset size: 6

Structure	Worst-case loss	Average loss	Gy conditional number
t131 t132 t133 t134 t135 vf	0.021663039534984583	0.0009027663848921795	280.2185249888746
v t131 t132 t133 t134 t135	0.021760035046254723	0.0009255049331204096	803.5206954103176
t131 t132 t133 t134 t135 lf	0.02178841967707324	0.0009068286828392157	283.57587490246544
l t131 t132 t133 t134 t135	0.021816369292573003	0.0009260449770867081	708.1122934899485
t130 t132 t133 t134 t135 vf	0.02190085656001738	0.000911461781278687	278.8779178778674

Figure 45 – Best control structures using linear combinations of measurements as CV candidates - Subset of size 9.



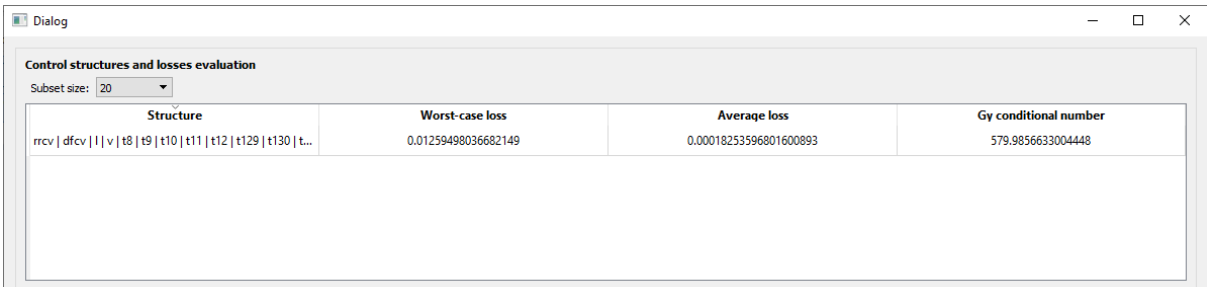
Dialog

Control structures and losses evaluation

Subset size: 9

Structure	Worst-case loss	Average loss	Gy conditional number
dfcv t129 t130 t131 t132 t133 t134 t135 vf	0.014174389404086437	0.0004483539271573793	326.8273381520674
t129 t130 t131 t132 t133 t134 t135 bf vf	0.014174389404086437	0.0004483539271573798	326.8273381520674
rrcv t129 t130 t131 t132 t133 t134 t135 lf	0.01418473108210343	0.0004486136854638603	174.53370426337736
dfcv t129 t130 t131 t132 t133 t134 t135 lf	0.014186514667493643	0.0004486811508436211	330.8443788797334
t129 t130 t131 t132 t133 t134 t135 bf lf	0.014186514667493653	0.0004486811508436214	330.8443788797334

Figure 46 – Best control structures using linear combinations of measurements as CV candidates - All available measurements.



Dialog

Control structures and losses evaluation

Subset size: 20

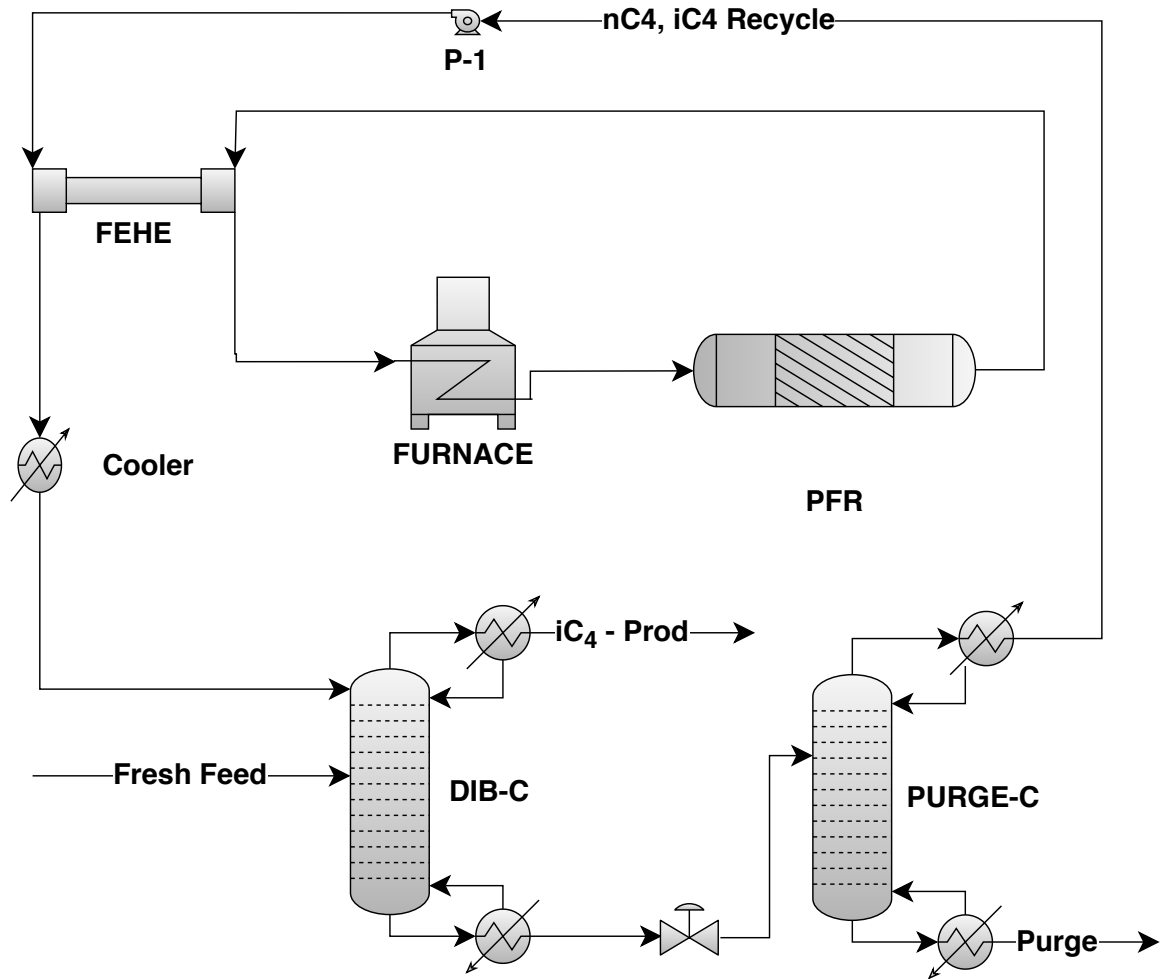
Structure	Worst-case loss	Average loss	Gy conditional number
rrcv dfcv l v t8 t9 t10 t11 t12 t129 t130 t...	0.01259498036682149	0.00018253596801600893	579.9856633004448

6.3 The C_4 Isomerization Process

The last case-study presented consists in a C_4 isomerization process, that aims to convert n -butane ($n - C_4$) into isobutane ($i - C_4$). The latter can be used as an octane-enhancing gasoline blending agent, and also it is an precursor for isobutyl alcohol production [Jagtap and Kaistha \(2012\)](#). The process described in this case-study it is based on the work of ([JAGTAP; KAISTHA, 2012](#)): Base operating conditions and optimal operating ones. The idea of this case study is to depict to reader the second mode of operation that can be used in *Metacontrol*, described in [Section 5.1](#) when the optimal operating point it is known. This was implemented within *Metacontrol* because there is a plethora of papers and discussions over the several years that addresses the optimization of several processes (([JAGTAP; KAISTHA, 2012](#); [JAGTAP; PATHAK; KAISTHA, 2013](#); [ARAÚJO; GOVATSMARK; SKOGESTAD, 2007](#); [ARAÚJO; SKOGESTAD, 2008](#); [GERA et al., 2013](#); [LIU et al., 2019](#); [SKOGESTAD, 2004](#)), just to name a few), and when one is dealing with economic plantwide control specially, there are several results that can be anticipated regarding active constraints. For a deeper understanding, the reader should refer to [Chapter 7, Section 7.2](#).

Thus, it is understood that there is a relevant number of experienced researchers that have interest in using the the local methods derived by [Halvorsen et al. \(2003\)](#) and [Alstad, Skogestad, and Hori \(2009\)](#) in order to find self-optimizing variables (or linear combinations of measurements), but already know constraints that must be controlled on their particular applications, specially when this task can be done in a comprehensive software environment, which is the case for *Metacontrol*. In such cases, there is no need to used mode 1 from *Metacontrol*, and the user can simply build a metamodel of the reduced-space problem, merely providing the simulation file of the process with the active constraints already implemented, and sample the process using the unconstrained degrees of freedom, in order to generate the necessary high-order data, to finally obtain the most promising CV candidates.

The process flowsheet can be found on figure [47](#). The process was already optimized by [Jagtap and Kaistha \(2012\)](#) as stated before. Therefore, it was used the previously found optimal point from the aforementioned work, and it is described in [Table 11](#). For this process, [Jagtap and Kaistha \(2012\)](#) kept the composition of n -butane on the bottoms of the purge column constant, and the other variables that are fixed were active constraints of the optimization problem: Either anticipated or calculated, except regarding the cooler temperature, that was alleged to have little impact on the objective function, and it was kept constant. Therefore, the aforementioned fixed composition will be considered as an unconstrained degree of freedom, differently from [Jagtap and Kaistha \(2012\)](#), and the reason is to evaluate if there is another variable that is easier to control than a composition that can be used in the control structure.

Figure 47 – C_4 Isomerization process flowsheet.

In Figure 48, it can be seen that the only expression built for this problem was the economic objective function, due to the fact that the problem is already unconstrained (the active constraints are already known). Similarly to the first and second case studies, the user must identify the process disturbances, CV candidates and degrees of freedom (Figure 49).

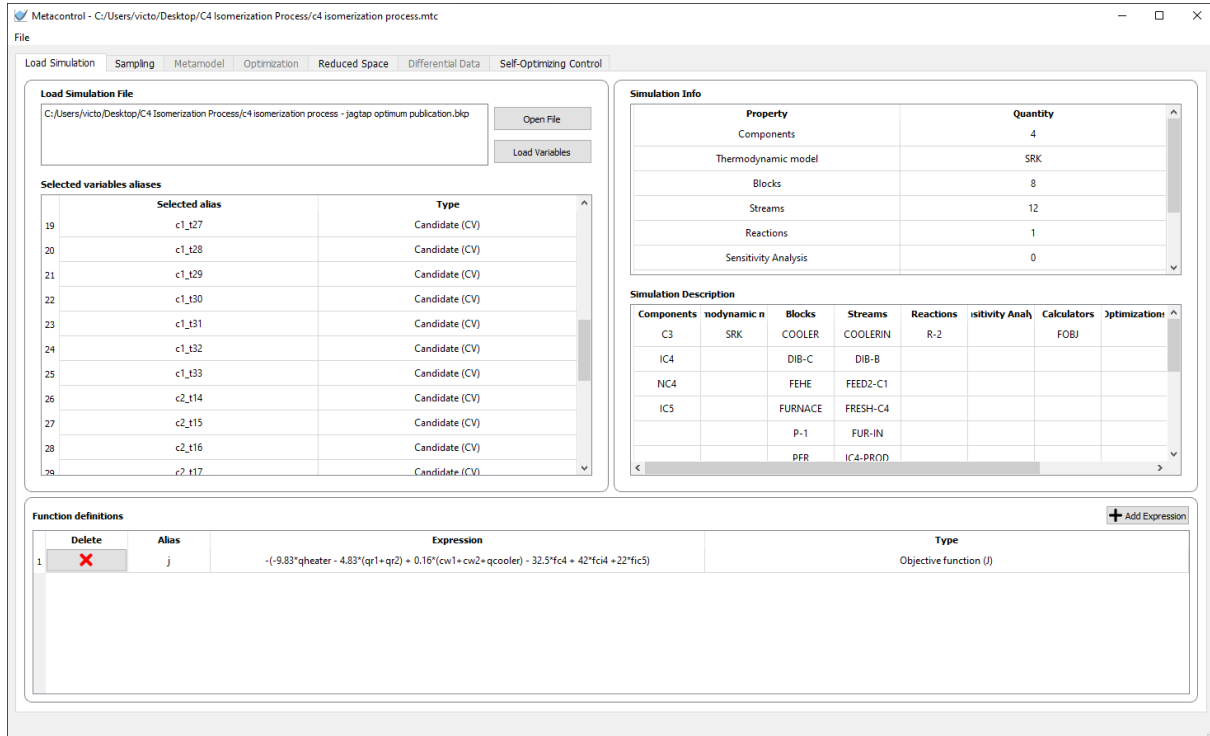
For the expected disturbances, the values come from Jagtap and Kaistha (2012), and disturbances for the amounts of isobutane and n -butane in the feed were considered, with a range of 10% of the nominal values. However, instead of considering the compositions, the values of the individual component flow rates were used in the design of experiments. Regarding CV candidates, sensitive temperatures at the optimal operating point were inspected for both columns, and the most sensitive ones were considered as CV candidates. The full list of CV candidates can be seen in Table 12.

50 points were sampled with an amplitude of $\pm 0.5\%$ around the optimal point (Figure 50), and the gradients and Hessians could be extracted (Figure 51). Lastly, Similarly as the previous cases, the implementation error for temperatures was considered as 0.5°C , 10^{-3} for flow rates and 10^{-6} for compositions. All the aforementioned data was inserted

Table 11 – C4 Isomerization process optimization summary.

Objective Function: Profit [\$/h]	
$J = -9.83 \times Q_{furnace} - 4.83 \times (Q_{DIB-C}^{reboiler} + Q_{PURGE-C}^{reboiler})$ $-0.16 \times (Q_{DIB-C}^{condenser} + Q_{PURGE-C}^{condenser})$ $-32.5 \times F_{C_4} + 42 \times F_{i-C_4} + 22 \times F_{i-C_5}$	
Process constraints	
$T_{reactor} = 200^\circ\text{C}$ (active)	$0 \leq Q_{furnace} \leq 1.3$ (base-case)
$0 \leq V_1 \leq 1.3$ (base-case)	$P_{reactor} = 45$ bar (active)
$0 \leq V_2 \leq 1.5$ (base-case)	
$T_{cooler} = 53^\circ\text{C}$ (fixed)	
Unconstrained degrees of freedom	
$x_{i-C_4}^{B1} = 0.0565$	$x_{i-C_5}^{D2} = 0.02$
$x_{n-C_4}^{B2} = 0.01$	

Figure 48 – C4 Isomerization process - loading simulation. The cooling water price is positive due to signal convention inside the process simulator - heat removed from the system has a negative sign.



inside *Metacontrol*, as can be seen in Figure 52.

For the sake of brevity only the single measurement policy was considered in this analysis. Figure 53 shows that, not surprisingly, the control of sensitive temperatures and the composition of the pollutant (in this case, $i - C_5$), yielded the lowest losses. However, keeping temperatures and flow rates with constant setpoints instead of using compositions are also promising control structures, as can be seen in Figure 54

Figure 49 – C_4 Isomerization process - loading variables.

The screenshot shows a software window titled 'Load Simulation' with a 'Load Variable Tree' tab. It contains two main sections: 'Input variables' and 'Output variables', each with a tree view on the left and a table of variables on the right.

Input variables:

Path	Alias	Type
\Data\Streams\FRESH-C4\Input\FLOW\MIXED\IC4	fic4_dist	Disturbance (d)
\Data\Streams\FRESH-C4\Input\FLOW\MIXED\NC4	fnc4_dist	Disturbance (d)
\Data\Blocks\DIB-C\Input\VALUE\2	xic4b1	Manipulated (MV)
\Data\Blocks\PURGE-C\Input\VALUE\1	xic5d2	Manipulated (MV)
\Data\Blocks\PURGE-C\Input\VALUE\2	xnc4b2	Manipulated (MV)

Output variables:

Path	Alias	Type
\Data\Blocks\FURNACE\Output\QCALC	qheater	Auxiliary
\Data\Blocks\DIB-C\Output\REB_DUTY	qr1	Auxiliary
\Data\Blocks\DIB-C\Output\COND_DUTY	cw1	Auxiliary
\Data\Blocks\PURGE-C\Output\REB_DUTY	qr2	Auxiliary
\Data\Blocks\PURGE-C\Output\COND_DUTY	cw2	Auxiliary
\Data\Streams\FRESH-C4\Output\MOLEFLMX\MIXED	fc4	Auxiliary
\Data\Streams\IC4-PROD\Output\MOLEFLMX\MIXED	fc4	Auxiliary
\Data\Blocks\DIB-C\Output\B_TEMP\21	c1_t21	Candidate (CV)
\Data\Blocks\DIB-C\Output\B_TEMP\22	c1_t22	Candidate (CV)
\Data\Blocks\DIB-C\Output\B_TEMP\23	c1_t23	Candidate (CV)
\Data\Blocks\DIB-C\Output\B_TEMP\24	c1_t24	Candidate (CV)

Table 12 – CV Candidates for C_4 Isomerization process.

Variable (alias used in <i>Metacontrol</i>)	Description
c1_t“x’	1st column stage X temperature (stages 21-33) (°C)
c2_t“x’	2nd column stage X temperature (stages 14-20) (°C)
x_ic4_b1	1st column $i - C_4$ bottoms composition
x_ic5_d2	2nd column $i - C_5$ distillate composition
x_nc4_b2	2nd column $n - C_4$ bottoms composition
c1_v	1st column boilup rate ($kmol/h$)
c2_v	2st column boilup rate ($kmol/h$)
c1_l	1st column reflux rate ($kmol/h$)
c1_l	2nd column reflux rate ($kmol/h$)

Figure 50 – C4 Isomerization process - loading variables.

Metacontrol - C:/Users/victo/Desktop/C4 Isomerization Process/c4 isomerization process.mtc

File

Load Simulation Sampling Metamodel Optimization Reduced Space Differential Data Self-Optimizing Control

Variable Activity

	xic4b1	xic5d2	xic4b2	c1_t21	c1_t22	c1_t23	c1_t24	c1_t25	c1_t26	c1_t27	c1_t28	c1_t29	c1_t30	c1_t31	c1_t32	c1_t33	c2_t14	c2_t15	c2_t16	c2_t17	c2_t18	c2_t19	c2_t20	ic4_b	ic5_d	nc4_b	c1_v	c1_l	c2_l	c2_v
Active	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Pairing				Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	Selec...	
Type	Mani...	Mani...	Mani...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	Can...	
Value	0.0565	0.02	0.01																											

Range of Disturbances

Disturbance variable	Lower bound	Upper bound	Nominal Value
fic4_dist	62.82828	63.45972	63.144
fnc4_dist	180.631305	182.446695	181.539
xic4b1	0.0562175	0.0567825	0.0565
xic5d2	0.0199	0.0201	0.02

Data Source

☒ Simulation file: C:/Users/victo/Desktop/C4 Isomerization Process/c4 isomerization process - jagtap optimum publication.bkp

☐ CSV file:

Reduced Space Sampled Data

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
1	Cas...	Sta...	Inp...			Out...																													
2		xic...	xic...	xic...	xic...	c1...	c1...	c1...	c1...	c1...	c1...	c1...	c1...	c1...	c1...	c1...	c1...	c1...	c2...	c2...	c2...	c2...	c2...	c2...	c2...	xj...	xj...	xj...	xj...	c1_v	c1_l	c2_l	c2_v	j	qh...
3	1	ok	0...	0...	0...	49.9...	50.1...	50.5...	51.0...	51.5...	52.1...	52.7...	53.3...	53.9...	54.4...	55.0...	55.6...	55.9...	51.6...	55.5...	60.9...	66.5...	71.2...	74.3...	76.2...	0...	0...	0...	126...	104...	185...	335...	-18...	0...	
4	2	ok	0...	0...	0...	49.8...	50.0...	50.5...	51.0...	51.5...	52.1...	52.7...	53.3...	53.9...	54.4...	55.0...	55.6...	55.9...	51.6...	55.5...	60.9...	66.5...	71.1...	74.3...	76.2...	0...	0...	0...	126...	104...	185...	334...	-18...	0...	
5	3	ok	0...	0...	0...	49.9...	50.1...	50.5...	51.0...	51.5...	52.1...	52.7...	53.3...	53.9...	54.4...	55.0...	55.6...	55.9...	51.5...	55.5...	60.8...	66.5...	71.1...	74.3...	76.2...	0...	0...	0...	126...	105...	185...	335...	-18...	0...	

Figure 51 – C4 Isomerization process - High-order data obtainment.

Metacontrol - C:/Users/victo/Desktop/C4 Isomerization Process/c4 isomerization process.mtc

File

Load Simulation Sampling Metamodel Optimization Reduced Space Differential Data Self-Optimizing Control

Reduced space metamodel training

Differentiation method

☒ Approximate analytical (Kriging predictions equations)

☐ Numeric (numdiff/fools)

☐ Automatic (autograd)

Gradient and Hessian

Gradient results

	xic4b1	xic5d2	xic4b2
c1_t21	-12.836326388972429	8.63504478279614	-0.042249254165221524
c1_t22	-13.481756601851558	13.06146593641944	-0.04496062893030825
c1_t23	-16.50294434834123	12.409967896821918	-0.010403480020199085
c1_t24	-19.260461006875143	11.671789201269613	0.02444776783794345
c1_t25	-21.589055611608494	10.880268350928633	0.057182395244758856

	fic4_dist	fnc4_dist
c1_t21	0.002573767427431414	0.012481551977264933
c1_t22	0.00240712749556012	0.013202372540307289
c1_t23	0.0014494087633839	0.01199761211671464
c1_t24	-4.39062871172138e-05	0.010712591050155817
c1_t25	-0.0020369084245337285	0.009447731699025203

Hessian results

	xic4b1	xic5d2	xic4b2
xic4b1	50.6041338420517	-0.11366576111820838	0.1294628249983691
xic5d2	-0.11366576111820838	14.720208793163307	-0.0021408586497470143
xic4b2	0.1294628249983691	-0.0021408586497470143	21.13888162564711

	fic4_dist	fnc4_dist
xic4b1	0.7183106061448671	-0.008061791068073582
xic5d2	0.14466285763950526	0.0012800061926166487
xic4b2	0.04442315698802312	-0.0006800829448780119

Hessian matrix positive-definite correction

Figure 52 – C4 Isomerization process - Self-Optimizing Control input.

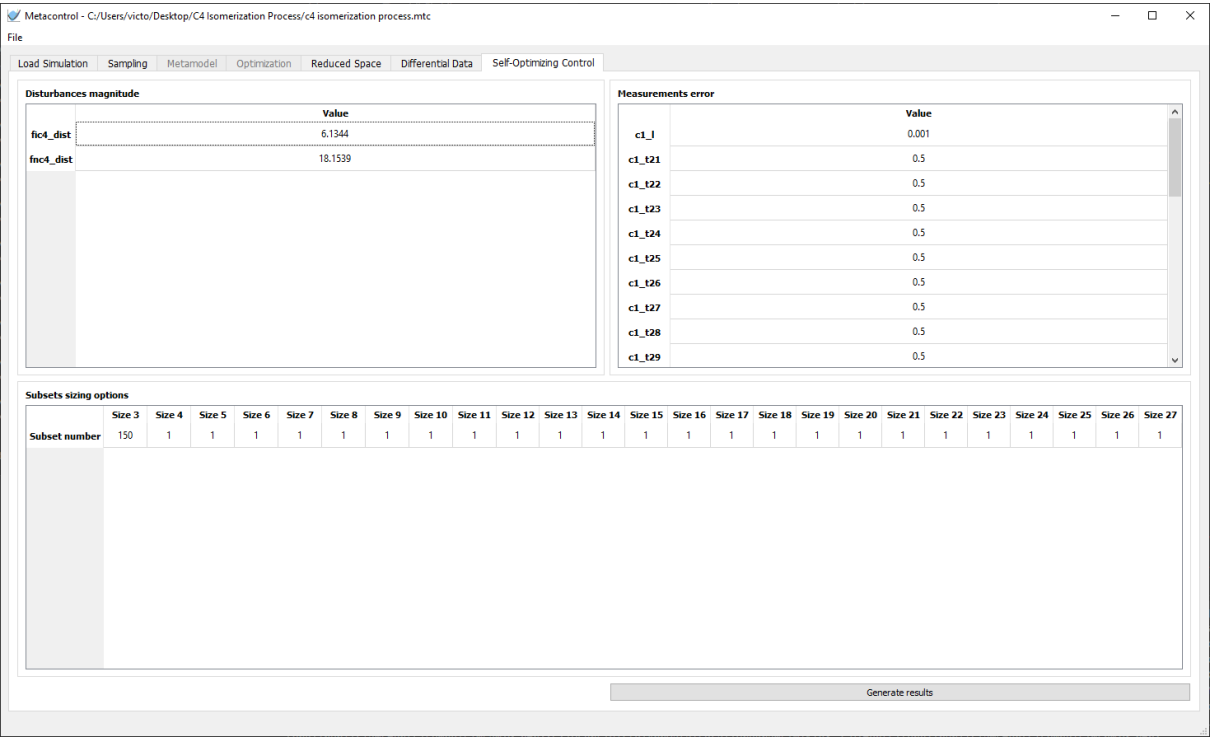


Figure 53 – C4 Isomerization process - Single measurements policy: Best CV candidates.

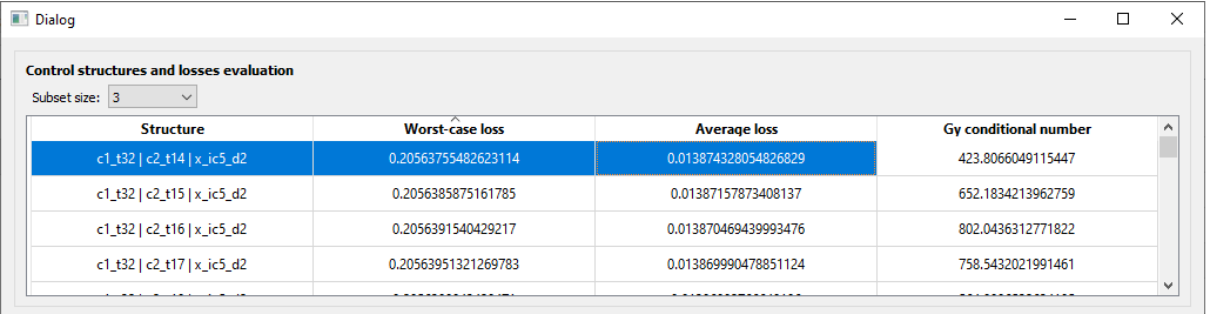
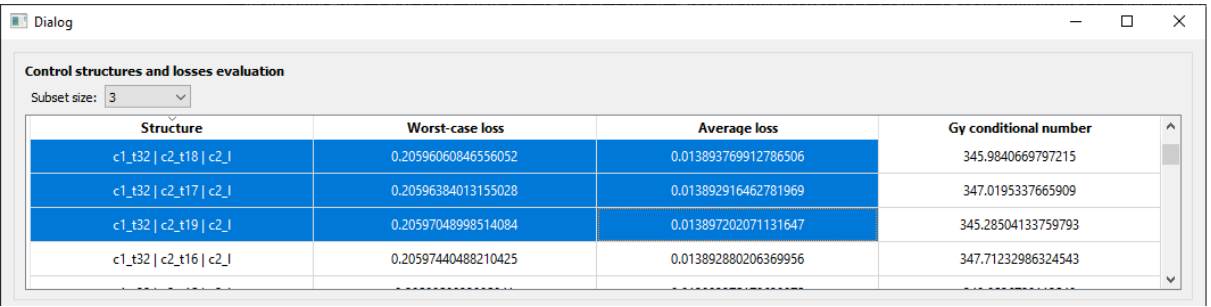


Figure 54 – C4 Isomerization process - Single measurements policy: Best CV candidates not using compositions.



7 *Metacontrol* practice and recommendations

In this chapter will be done a thorough discussion of usage aspects of *Metacontrol*, recommendations and good-practices when using the proposed software. The main idea is to pass to the reader a concise set of recommendations and the justification for these. The covered aspects will be mainly:

1. The process simulator sampling general good practice: Towards maximizing cases convergence while keeping precision;
2. The initial sample: Dimensionality and the choice of the degrees of freedom towards the optimization feasible region;
3. The adaptive sampling optimization algorithm parameters;
4. The reduced-space problem sample: Dimensionality and the effect of the [Søren Nymand Lophaven, Hans Bruun Nielsen, and Jacob Søndergaard \(2002\)](#) hyperparameters objective function in gradient/hessian evaluation.

7.1 The process simulator sampling general good practice: Towards maximizing cases' convergence while keeping precision

When one is using a modular flowsheet process simulator, such as Aspen Plus, problems will eventually arise based on model complexity and the necessity of realistically representing a process becomes of utmost importance. Unit operations such as distillation and chemical reactors are sources of numerical noise, result of termination criteria in the algorithms that solve these unit operations blocks, or even rounding numbers, as discussed by [Caballero and Grossmann \(2008\)](#). With this problem, some workarounds are suggested in order to try to reduce the number of unconverged cases and to enhance precision. The problem gets even more complicated when recycle streams are present in the process, acting as noise-amplifiers ([QUIRANTE; CABALLERO, 2016](#)). With the aforementioned problems being stated, some recommendations that worked well based on the study of previous publications are:

1. Tighten as much as possible the following convergences' tolerances: tear stream-related, Inside/Out algorithm for distillation blocks-related, and chemical reactors'

- convergence related. Mixers, pumps and heat-exchangers were previously shown to not introduce - or introduce irrelevant - noise (QUIRANTE; CABALLERO, 2016).
2. AspenTech (2017) recommends that when a tear streams/recycle streams are present in the flowsheet interacting with distillation column blocks, the convergence tolerance of the former should be at least a order of magnitude lower (tighter) than the latter. This will avoid the numerical noise from the distillation block to impede convergence.
 3. Reconcile the tear streams. When these streams are reconciled, there are input specifications inside those that will aid the convergence. In fact, the number of cases converged increased when the author of this work reconciled the flowsheet with a feasible operating point. The procedure to reconcile the tear streams can be found in AspenTech (2017).

7.2 The initial sample: Dimensionality and the choice of the degrees of freedom towards the optimization feasible region

In order to successfully obtain a nominally optimal operating point, one must ensure that the initial sample provided to the algorithm proposed by Caballero and Grossmann (2008) it is capable of representing the basic trends of the objective function and the nonlinear constraints created by the user. However, the so called “curse of Dimensionality” Forrester, Sobester, and Keane (2008) still looms over big-data and machine-learning fields of study. The dimensionality can be reduced using engineering common sense, as different authors did in the past for instance (ARAÚJO; GOVATSMARK; SKOGESTAD, 2007; ARAÚJO; SKOGESTAD, 2008; GERA et al., 2013).

The idea is to keep constant the degrees of freedom that have little or no effect on the objective function, and, instead, use only the degrees of freedom that are “dominant” in the process. Another way of reducing the number of degrees of freedom is to anticipate results that are expected to be active constraints, specially when one is dealing with economic Self-Optimizing Control. Strictly speaking: one can expect active constraints in liquid-phase reactor’s maximum holdups and temperatures, when the kinetics are simple; maximum operating pressure for gas-phase reactors, minimum acceptable purity for valuable products (avoiding product “giveaway” (JACOBSEN; SKOGESTAD, 2011)) and maximum impurity levels in contaminants restrictions all expected to occur, for instance. These results were repeatedly found over several years of Self-Optimizing Control studies, and were synthesized in the work of Minasidis, Skogestad, and Kaistha (2015).

Regarding the number of points in the initial sample, this problem it is a heuristic one. If the number of points is too low, the prediction capability of the metamodel decreases, and the optimal solution found before the contraction steps could be far from the actual

solution, and several movement steps would be necessary, consuming time (CABALLERO; GROSSMANN, 2008). In addition, the number of points as discussed by Caballero and Grossmann (2008) is case-dependent: For instance, given a function with a sharp peak, it will require a large number of points around it. However, using an excessively amount of points can be time consuming (if each model evaluation takes considerable time and due to matrix inversion operations performed by the *pydace* toolbox) or even worse, starting ill-conditioning the correlation matrix R , if the large amount of points are not separated enough (new points introduced can be clustered and making R ill-conditioned). This can also jeopardize the *Kriging* metamodel construction and consequently, its prediction capability.

In order to balance accuracy and number of points, the author of this work use as a starting point the heuristic proposed by (CABALLERO; GROSSMANN, 2008): around 30-50 points for two or three variables, around 70-80 for four/five variables, and increasing by 10 points for each additional independent variable. The work of Caballero and Grossmann (2008) uses the maximum practical limit of 100 points. However, some cases might need more points than this limit, and the reader should not feel inhibited of trespassing this limit. Nevertheless, he must be careful with the excess of points, due to the reasons aforementioned.

Lastly, regarding the degrees of freedom chosen to perform the design of experiments, it has been shown by previous authors Hori, Skogestad, and Alstad (2005) and Kariwala, Cao, and Janardhanan (2008) that any variable can be chosen as a degree of freedom, since the problem is evaluated in steady-state, there is no loss of generality. Therefore, one can try to convert the nonlinear constraints, for instance, to decision variables (box constraints). This will enforce that all the sampled (and converged) cases, to be inside the feasible region of the optimization problem. To use design specifications/constraints as decision variables is not a new approach and it has been done in the past Gera et al. (2013) and Jagtap, Pathak, and Kaistha (2013) when the cited authors were using a NLP solver coupled with the process simulator, trying to ensure robust convergence of the optimization problem. As an example, if one chooses to sample the purity of a valuable product from a distillation column instead of a reflux rate, he not only knows automatically the bounds of that variable $x_{min} \leq x \leq 1$, but all of the sampled cases would be within the optimization problem feasible region.

7.3 The adaptive sampling optimization algorithm parameters

The contraction factors (namely *first* and *second*), refinement, termination and *maximum contraction* tolerances can be tuned in order to achieve a quicker optimization convergence. As discussed by Caballero and Grossmann (2008), using a large value for the

first contraction factor might result in a increased amount of hypercube movements, even though the final result will be the same. As a general setup, the parameters values are 0.6 and 0.4 for *first* and *second* factors, respectively.

The termination parameter value, as a rule of thumb, is set to be at least a order of magnitude higher than the tightest convergence parameter set in the process simulator, so that the optimization procedure does not adjust the numerical noise introduced by the simulation software. Then the refinement parameter is set to be at least one order of magnitude higher than the termination value.

However, care must be taken when specifying the *maximum contraction* factor, since this parameter tells the algorithm the minimum hypercube size to contract. This a safeguard parameter to prevent the introduction of ill-conditioning in the *Kriging* input matrix. Consequently, this parameter cannot be higher than the refinement value.

7.4 The reduced-space problem sample: Dimensionality and the effect of the *Kriging* hyperparameters objective function in gradient/hessian evaluation.

When one is obtaining high-order data using kriging metamodels, he must be aware of the value of the ψ objective function that is minimized using the hyperparameters as degrees of freedom (Equation 3.9). It has been found that convergence values that are acceptable for general prediction purposes (i.e.: 10^{-3} to 10^{-4}) are not suitable to predict the gradients and hessians necessary to the SOC study. Regarding the hessians ($(\frac{\partial^2 J}{\partial u \partial d}), (\frac{\partial^2 J}{\partial u \partial u})$), they are even more sensitive to the hyperparameters estimation (an expected result, due to the fact that they are second derivatives). It was found that a good value for the ψ function that was capable of generating metamodels that are capable of predicting the gradients and hessians with precision, would be around 10^{-5} (at least). This variable can be inspected within *Metacontrol*, under the panel “Validation metrics”, for the reduced-space metamodel, as can be seen in Figure 55. Its value can be used as a metric of how precise the gradient and hessian evaluation will be. As showed in the previous examples when for illustration purposes the original gradient was evaluated, the agreement between the metamodel-based gradient and the original one is excellent.

Figure 55 – ψ function value - named “perf” within *Metacontrol*.

Validation metrics						
	mccpout	sSt	fco2out	xco2out	j	co2rcv
Perf	8.629183E-12	2.156030E-06	6.844434E-09	4.671179E-08	1.086412E-07	7.619089E-08
OMSE	4.418632E-15	7.393014E-09	6.604627E-10	1.352940E-17	5.744203E-11	2.121798E-16
ORHSE	6.647279E-08	8.598264E-05	2.569947E-05	3.678233E-09	7.579052E-06	1.456639E-08
OMAE	5.634071E-08	7.447521E-05	2.071772E-05	2.734540E-09	6.290715E-06	1.246374E-08
OR2	1.000000E+00	9.999783E-01	1.000000E+00	9.999999E-01	9.999997E-01	9.999997E-01

8 Conclusion

The software developed (*Metacontrol*) aims to become a tool that enables the easy deployment of the methodology for Self-Optimizing Control structure selection through the use of surrogate models. The dissertation explains the complete workflow of the technology implemented in the software. Also, the functionalities and capability of *Metacontrol* were demonstrated through 3 case studies, showing how the tool can be used for performance enhancement in the first case (reduction in the energy consumption in a CO_2 compression process), indirect control in the second case (minimization of nominal setpoint deviation of a hydrocarbon separation process) and economic plantwide control in the third case.

In addition, a discussion was done of good practices on how to set the simulations, how to specify parameters in a surrogate optimization and what to expect of metrics used to estimate gradients and Hessian.

All the data, example files and the *Metacontrol* source code presented in this work can be found at <https://github.com/feslima/metacontrol>. The tool acts as a resource to the scientific community to implement, analyse or improve current control strategies of industrial processes.

Therefore, the author invites the readers to test and give feedback on the tool and methodology.

References

- ALEXANDROV, N. et al. Optimization with variable-fidelity models applied to wing design. In: 38TH aerospace sciences meeting and exhibit. [S.l.: s.n.], 2000. P. 841.
- ALSTAD, Vidar; SKOGESTAD, Sigurd. Null space method for selecting optimal measurement combinations as controlled variables. **Industrial & engineering chemistry research**, v. 46, n. 3, p. 846–853, 2007. DOI: [10.1021/ie060285+](https://doi.org/10.1021/ie060285+).
- ALSTAD, Vidar; SKOGESTAD, Sigurd; HORI, Eduardo S. Optimal measurement combinations as controlled variables. **Journal of Process Control**, Elsevier BV, v. 19, n. 1, p. 138–148, Jan. 2009. DOI: [10.1016/j.jprocont.2008.01.002](https://doi.org/10.1016/j.jprocont.2008.01.002).
- ALVES, Victor M. C. et al. Metamodel-Based Numerical Techniques for Self-Optimizing Control. **Industrial & Engineering Chemistry Research**, American Chemical Society (ACS), v. 57, n. 49, p. 16817–16840, Nov. 2018. DOI: [10.1021/acs.iecr.8b04337](https://doi.org/10.1021/acs.iecr.8b04337).
- ARAÚJO, Antonio; SKOGESTAD, Sigurd. Control structure design for the ammonia synthesis process. **Computers & Chemical Engineering**, v. 32, n. 12, p. 2920–2932, 2008. ISSN 0098-1354. DOI: <https://doi.org/10.1016/j.compchemeng.2008.03.001>. Available from: <http://www.sciencedirect.com/science/article/pii/S0098135408000392>.
- ARAÚJO, Antonio Carlos Brandão de; GOVATSMARK, Marius; SKOGESTAD, Sigurd. Application of plantwide control to the HDA process. I—steady-state optimization and self-optimizing control. **Control Engineering Practice**, v. 15, n. 10, p. 1222–1237, 2007. Special Issue - International Symposium on Advanced Control of Chemical Processes (ADCHEM). ISSN 0967-0661. DOI: <https://doi.org/10.1016/j.conengprac.2006.10.014>. Available from: <http://www.sciencedirect.com/science/article/pii/S0967066106001997>.
- ASPENTECH. **Aspen Plus User Guide**. English. Version Version 10. [S.l.], 2017.
- BUHRE, B. J. P. et al. Oxy-fuel combustion technology for coal-fired power generation. **Progress in Energy and Combustion Science**, v. 31, n. 4, p. 283–307, 2005. ISSN 0360-1285. DOI: <https://doi.org/10.1016/j.pecs.2005.07.001>. Available from: <http://www.sciencedirect.com/science/article/pii/S0360128505000225>.
- CABALLERO, José A.; GROSSMANN, Ignacio E. An algorithm for the use of surrogate models in modular flowsheet optimization. **AIChE Journal**, v. 54, n. 10, p. 2633–2650, 2008. DOI: [10.1002/aic.11579](https://doi.org/10.1002/aic.11579). eprint: <https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/aic.11579>. Available from: <https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.11579>.

- CAO, Yi; KARIWALA, Vinay. Bidirectional branch and bound for controlled variable selection. **Computers & Chemical Engineering**, Elsevier BV, v. 32, n. 10, p. 2306–2319, Oct. 2008. DOI: [10.1016/j.compchemeng.2007.11.011](https://doi.org/10.1016/j.compchemeng.2007.11.011).
- CAO, Yi; SAHA, Prabirkumar. Improved branch and bound method for control structure screening. **Chemical Engineering Science**, Elsevier BV, v. 60, n. 6, p. 1555–1564, Mar. 2005. DOI: [10.1016/j.ces.2004.10.025](https://doi.org/10.1016/j.ces.2004.10.025).
- DILLON, D. J. et al. Oxy-combustion processes for CO₂ capture from power plant. **Engineering investigation report**, v. 9, 2005.
- FORRESTER, Alexander; SOBESTER, Andras; KEANE, Andy. **Engineering design via surrogate modelling: a practical guide**. [S.l.]: John Wiley & Sons, 2008.
- GERA, Vivek et al. Economic Plantwide Control of the Cumene Process. **Industrial & Engineering Chemistry Research**, v. 52, n. 2, p. 830–846, 2013. DOI: [10.1021/ie301386h](https://doi.org/10.1021/ie301386h). eprint: <https://doi.org/10.1021/ie301386h>. Available from: <https://doi.org/10.1021/ie301386h>.
- HALVORSEN, Ivar J. et al. Optimal Selection of Controlled Variables†. **Industrial & Engineering Chemistry Research**, American Chemical Society (ACS), v. 42, n. 14, p. 3273–3284, July 2003. DOI: [10.1021/ie020833t](https://doi.org/10.1021/ie020833t).
- HORI, Eduardo S.; SKOGESTAD, Sigurd. Selection of Control Structure and Temperature Location for Two-Product Distillation Columns. **Chemical Engineering Research and Design**, Elsevier BV, v. 85, n. 3, p. 293–306, Jan. 2007. DOI: [10.1205/cherd06115](https://doi.org/10.1205/cherd06115).
- HORI, Eduardo S.; SKOGESTAD, Sigurd; ALSTAD, Vidar. Perfect Steady-State Indirect Control. **Industrial & Engineering Chemistry Research**, American Chemical Society (ACS), v. 44, n. 4, p. 863–867, Feb. 2005. DOI: [10.1021/ie0497361](https://doi.org/10.1021/ie0497361).
- HORI, Eduardo Shiguelo; SKOGESTAD, Sigurd. Selection of Controlled Variables: Maximum Gain Rule and Combination of Measurements. **Industrial & Engineering Chemistry Research**, v. 47, n. 23, p. 9465–9471, 2008. DOI: [10.1021/ie0711978](https://doi.org/10.1021/ie0711978). eprint: <https://doi.org/10.1021/ie0711978>. Available from: <https://doi.org/10.1021/ie0711978>.
- JACOBSEN, Magnus G.; SKOGESTAD, Sigurd. Active Constraint Regions for Optimal Operation of Chemical Processes. **Industrial & Engineering Chemistry Research**, v. 50, n. 19, p. 11226–11236, 2011. DOI: [10.1021/ie2012196](https://doi.org/10.1021/ie2012196). eprint: <https://doi.org/10.1021/ie2012196>. Available from: <https://doi.org/10.1021/ie2012196>.

- JAGTAP, Rahul; KAISTHA, Nitin. Economic Plantwide Control of a C4 Isomerization Process. **Industrial & Engineering Chemistry Research**, v. 51, n. 36, p. 11731–11743, 2012. DOI: [10.1021/ie3001293](https://doi.org/10.1021/ie3001293). eprint: <https://doi.org/10.1021/ie3001293>. Available from: <https://doi.org/10.1021/ie3001293>.
- JAGTAP, Rahul; PATHAK, Ashok S; KAISTHA, Nitin. Economic plantwide control of the ethyl benzene process. **AIChE Journal**, v. 59, n. 6, p. 1996–2014, 2013. DOI: [10.1002/aic.13964](https://doi.org/10.1002/aic.13964). eprint: <https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/aic.13964>. Available from: <https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.13964>.
- JIN, Bo; ZHAO, Haibo; ZHENG, Chuguang. Optimization and control for CO₂ compression and purification unit in oxy-combustion power plants. **Energy**, v. 83, p. 416–430, 2015. ISSN 0360-5442. DOI: <https://doi.org/10.1016/j.energy.2015.02.039>. Available from: <http://www.sciencedirect.com/science/article/pii/S0360544215001942>.
- JONES, Donald R. A taxonomy of global optimization methods based on response surfaces. **Journal of global optimization**, Springer, v. 21, n. 4, p. 345–383, 2001.
- KARIWALA, Vinay; CAO, Yi. Bidirectional branch and bound for controlled variable selection. Part II: Exact local method for self-optimizing control. **Computers & Chemical Engineering**, Elsevier BV, v. 33, n. 8, p. 1402–1412, Aug. 2009. DOI: [10.1016/j.compchemeng.2009.01.014](https://doi.org/10.1016/j.compchemeng.2009.01.014).
- KARIWALA, Vinay; CAO, Yi; JANARDHANAN, Sivaramakrishnan. Local self-optimizing control with average loss minimization. **Industrial & Engineering Chemistry Research**, v. 47, n. 4, p. 1150–1158, 2008. DOI: <https://doi.org/10.1021/ie070897+>.
- KOOHESTANIAN, Esmail et al. Sensitivity analysis and multi-objective optimization of CO₂CPU process using response surface methodology. **Energy**, v. 122, p. 570–578, 2017. ISSN 0360-5442. DOI: <https://doi.org/10.1016/j.energy.2017.01.129>. Available from: <http://www.sciencedirect.com/science/article/pii/S0360544217301366>.
- KUMMERER, Matthias; MOORE, Jason K. **Cython interface for the interior point optimizer IPOPT**. [S.l.: s.n.], Dec. 2019. Available from: <https://github.com/matthias-k/cyipopt>.
- LIU, Kaile et al. Self-Optimizing Control Structure and Dynamic Behavior for CO₂ Compression and Purification Unit in Oxy-fuel Combustion Application. **Industrial & Engineering Chemistry Research**, v. 58, n. 8, p. 3199–3210, 2019. DOI:

10.1021/acs.iecr.9b00121. eprint: <https://doi.org/10.1021/acs.iecr.9b00121>.

Available from: <<https://doi.org/10.1021/acs.iecr.9b00121>>.

LOPHAVEN, S. N.; NIELSEN, H. B.; SØNDERGAARD, J. **Aspects of the Matlab toolbox DACE**. Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2002. (IMM-TR-2002-13).

LOPHAVEN, Søren Nymand; NIELSEN, Hans Bruun; SØNDERGAARD, Jacob. **DACE - A Matlab Kriging Toolbox, Version 2.0**. [S.l.: s.n.], 2002.

LUYBEN, William L. Evaluation of criteria for selecting temperature control trays in distillation columns. **Journal of Process Control**, v. 16, n. 2, p. 115–134, 2006. ISSN 0959-1524. DOI: <https://doi.org/10.1016/j.jprocont.2005.05.004>. Available from: <<http://www.sciencedirect.com/science/article/pii/S0959152405000569>>.

MINASIDIS, Vladimiros; SKOGESTAD, Sigurd; KAISTHA, Nitin. Simple Rules for Economic Plantwide Control. In: GERNAEY, Krist V.; HUUSOM, Jakob K.; GANI, Rafiqul (Eds.). **12th International Symposium on Process Systems Engineering and 25th European Symposium on Computer Aided Process Engineering**. [S.l.]: Elsevier, 2015. v. 37. (Computer Aided Chemical Engineering). P. 101–108. DOI: <https://doi.org/10.1016/B978-0-444-63578-5.50013-X>. Available from:

<<http://www.sciencedirect.com/science/article/pii/B978044463578550013X>>.

MORARI, Manfred; STEPHANOPOULOS, George. Studies in the synthesis of control structures for chemical processes: Part II: Structural aspects and the synthesis of alternative feasible control schemes. **AIChE Journal**, Wiley, v. 26, n. 2, p. 232–246, Mar. 1980. DOI: [10.1002/aic.690260206](https://doi.org/10.1002/aic.690260206).

POSCH, Sebastian; HAIDER, Markus. Optimization of CO₂ compression and purification units (CO₂CPU) for CCS power plants. **Fuel**, v. 101, p. 254–263, 2012. 8th European Conference on Coal Research and Its Applications. ISSN 0016-2361. DOI:

<https://doi.org/10.1016/j.fuel.2011.07.039>. Available from:

<<http://www.sciencedirect.com/science/article/pii/S0016236111004364>>.

QUIRANTE, Natalia; CABALLERO, José A. Large scale optimization of a sour water stripping plant using surrogate models. **Computers & Chemical Engineering**, v. 92, p. 143–162, 2016. ISSN 0098-1354. DOI:

<https://doi.org/10.1016/j.compchemeng.2016.04.039>. Available from:

<<http://www.sciencedirect.com/science/article/pii/S0098135416301351>>.

SASENA, Michael James. **Flexibility and efficiency enhancements for constrained global design optimization with kriging approximations**. 2002. PhD thesis – University of Michigan Ann Arbor, MI.

- SKOGESTAD, Sigurd. Control structure design for complete chemical plants. **Computers & Chemical Engineering**, Elsevier BV, v. 28, n. 1-2, p. 219–234, Jan. 2004. DOI: [10.1016/j.compchemeng.2003.08.002](https://doi.org/10.1016/j.compchemeng.2003.08.002).
- SKOGESTAD, Sigurd. Plantwide control: the search for the self-optimizing control structure. **Journal of Process Control**, Elsevier BV, v. 10, n. 5, p. 487–507, Oct. 2000. DOI: [10.1016/s0959-1524\(00\)00023-8](https://doi.org/10.1016/s0959-1524(00)00023-8).
- SKOGESTAD, Sigurd; POSTLETHWAITE, Ian. **Multivariable feedback control: analysis and design**. [S.l.]: Wiley New York, 2007. v. 2.
- TOFTEGAARD, Maja B. et al. Oxy-fuel combustion of solid fuels. **Progress in Energy and Combustion Science**, v. 36, n. 5, p. 581–625, 2010. ISSN 0360-1285. DOI: <https://doi.org/10.1016/j.pecs.2010.02.001>. Available from: <http://www.sciencedirect.com/science/article/pii/S0360128510000201>.
- UMAR, Lia Maisarah et al. Selection of Controlled Variables using Self-optimizing Control Method. In: PLANTWIDE Control. [S.l.]: John Wiley & Sons, Ltd, Mar. 2012. P. 43–71. DOI: [10.1002/9781119968962.ch4](https://doi.org/10.1002/9781119968962.ch4).
- WÄCHTER, Andreas; BIEGLER, Lorenz T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. **Mathematical Programming**, v. 106, n. 1, p. 25–57, Mar. 2006. ISSN 1436-4646. DOI: [10.1007/s10107-004-0559-y](https://doi.org/10.1007/s10107-004-0559-y). Available from: <https://doi.org/10.1007/s10107-004-0559-y>.