Felipe Souza Lima

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

Campina Grande, Paraíba, Brasil

Março, 2020

Felipe Souza Lima

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

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Química da Universidade Federal de Campina Grande, como requisito parcial para obtenção do grau de Mestre. Área de concentração: Engenharia Química.

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

Março, 2020

Felipe Souza Lima

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

93p. : il. (algumas color.) ; 30 cm.

Supervisor: Dr. Antônio Carlos Brandão de Araújo

Dissertação Mestrado – Universidade Federal de Campina Grande
Unidade Acadêmica de Engenharia Química
Programa de Pós-Graduação em Engenharia Química, Março, 2020.

1. Software. 2. Metamodel. 3. Plantwide. 4. Self-optimizing Control. 5. Python. I. Orientador. II. Universidade xxx. III. Faculdade de xxx. IV. Título

Errata sheet

Elemento opcional da **NBR14724:2011**. Exemplo:

FERRIGNO, C. R. A. **Tratamento de neoplasias ósseas apendiculares com reimplantação de enxerto ósseo autólogo autoclavado associado ao plasma rico em plaquetas**: estudo crítico na cirurgia de preservação de membro em cães. 2011. 128 f. Tese (Livre-Docência) - Faculdade de Medicina Veterinária e Zootecnia, Universidade de São Paulo, São Paulo, 2011.

Folha	**Linha**	**Onde se lê**	**Leia-se**
1	10	auto-conclavo	autoconclavo

Felipe Souza Lima

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

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Química da Universidade Federal de Campina Grande, como requisito parcial para obtenção do grau de Mestre. Área de concentração: Engenharia Química.

Trabalho aprovado. Campina Grande, Paraíba, Brasil, XX de março de 2020:

Dr. Antônio Carlos Brandão de Araújo
Orientador

Professor
Dr. Heleno Bispo da Silva Júnior

Professor
Convidado 2

Campina Grande, Paraíba, Brasil
Março, 2020

For those who doubt themselves. Just embrace it.

Acknowledgements

Os agradecimentos principais são direcionados à Gerald Weber, Miguel Frasson, Leslie H. Watter, Bruno Parente Lima, Flávio de Vasconcellos Corrêa, Otavio Real Salvador, Renato Machnievscz[1] e todos aqueles que contribuíram para que a produção de trabalhos acadêmicos conforme as normas ABNT com LaTeX fosse possível.

Agradecimentos especiais são direcionados ao Centro de Pesquisa em Arquitetura da Informação[2] da Universidade de Brasília (CPAI), ao grupo de usuários *latex-br*[3] e aos novos voluntários do grupo *abnTeX2*[4] que contribuíram e que ainda contribuirão para a evolução do abnTeX2.

[1] Os nomes dos integrantes do primeiro projeto abnTeX foram extraídos de `http://codigolivre.org.br/projects/abntex/`
[2] `http://www.cpai.unb.br/`
[3] `http://groups.google.com/group/latex-br`
[4] `http://groups.google.com/group/abntex2` e `http://www.abntex.net.br/`

"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

Segundo a **NBR6028:2003**, o resumo deve ressaltar o objetivo, o método, os resultados e as conclusões do documento. A ordem e a extensão destes itens dependem do tipo de resumo (informativo ou indicativo) e do tratamento que cada item recebe no documento original. O resumo deve ser precedido da referência do documento, com exceção do resumo inserido no próprio documento. (. . .) As palavras-chave devem figurar logo abaixo do resumo, antecedidas da expressão Palavras-chave:, separadas entre si por ponto e finalizadas também por ponto.

Palavras-chave: latex. abntex. editoração de texto.

Abstract

This is the english abstract.

Keywords: latex. abntex. text editoration.

List of Figures

List of frames

List of Tables

List of abbreviations and acronyms

CPU	Compression Purification Unit
CV	Controlled Variable
DOE	Design Of Experiments
DOF	Degree of Freedom
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^y_d	Gain matrix with respect to the disturbances
H	Linear combination matrix
J	Process objective function
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	*Kriging* hyperparameter of variable activity
ε	Residuals or random noise
p	*Kriging* 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

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.

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.

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

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 strucutre 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

$$\Delta y = G^y \Delta u + G_d^y \Delta d \tag{2.3}$$

With

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

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

$$\Delta c = H\Delta y = G\Delta u + G_d \Delta d \tag{2.5}$$

With

$$\begin{aligned} G &= HG^y \\ G_d &= HG_d^y \end{aligned} \tag{2.6}$$

Linearizing the loss function results in Equation 2.7:

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

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

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

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:

$$d - d^* = W_d d' \tag{2.9}$$

$$n = HW_n^y n^{y'} = W_n n^{y'} \tag{2.10}$$

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

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

$$M \triangleq [M_d \quad M_n^y] \tag{2.12}$$

where

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

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

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

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

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

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

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

where

$$\tilde{F} = [F W_d W_n^y] \tag{2.17}$$

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 strucutre 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:

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

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:

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

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.

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

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.

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

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.

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

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

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

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 to given points, arbitrarily named w and x for instance, is defined by Equation 3.5:

$$\mathrm{Cov}\left[z_l(w), z_l(x)\right] = \sigma_l^2 \mathcal{R}\left(\theta_l, w, x\right), \quad l = 1, \ldots, q \tag{3.5}$$

With σ_l^2 being the process variance for the *lth* 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.

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

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 *lth* 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 $Y\left(x_i\right)$ (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 Nymand 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:

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

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 Nymand 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:

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

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:

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

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;

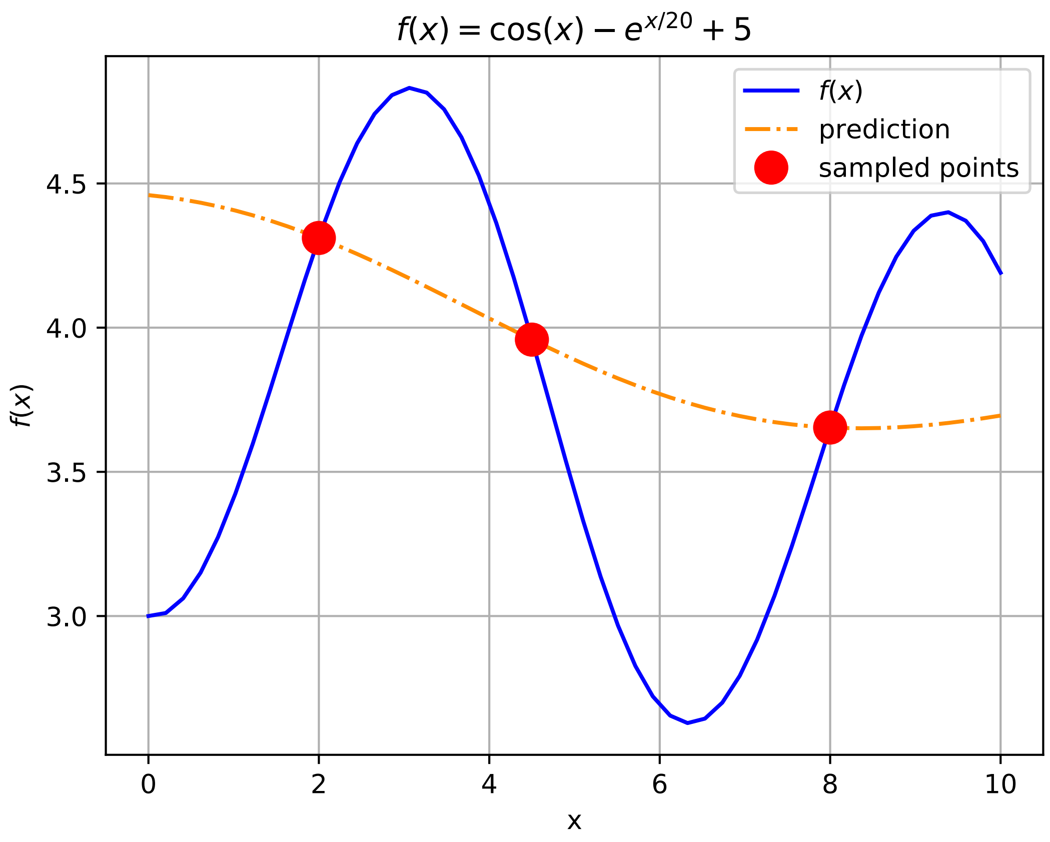
4. If the optimum from the metamodel is close enough (based on a chosen metric) to the original model, then this may be the true optimum. Otherwise, update the *Kriging* model by introducing the value found and return to step 2;

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.



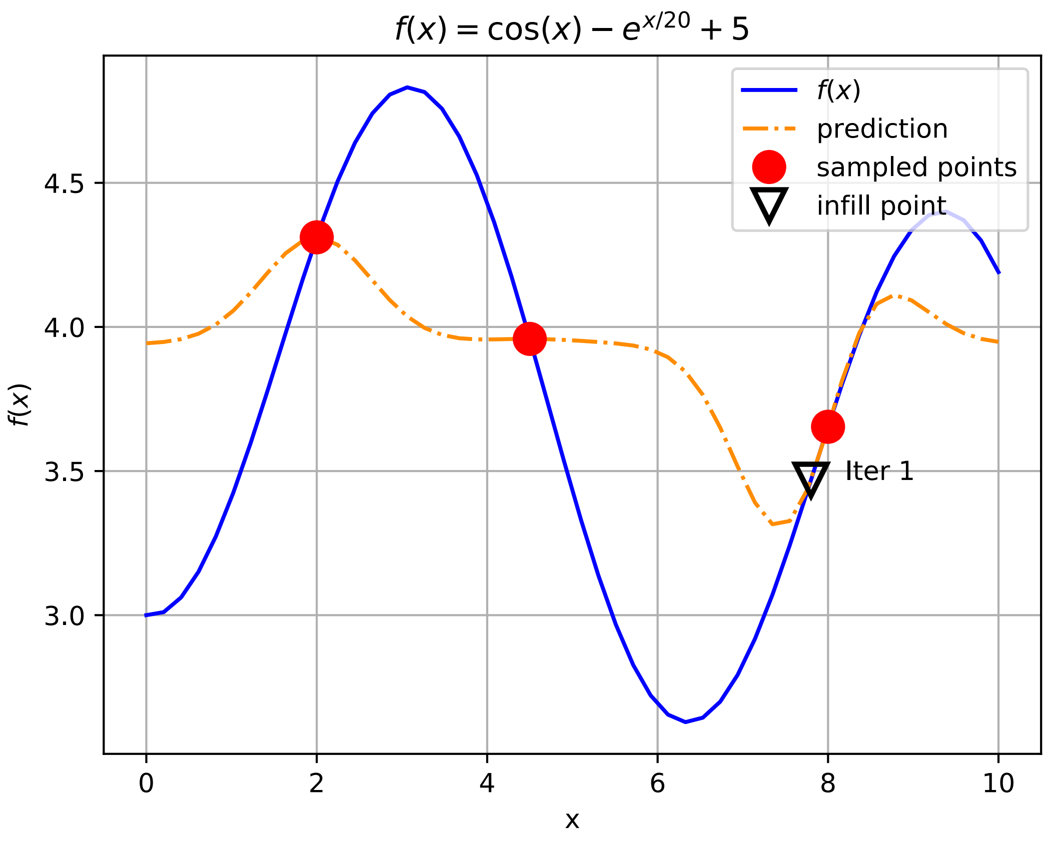
Source: Author.

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.

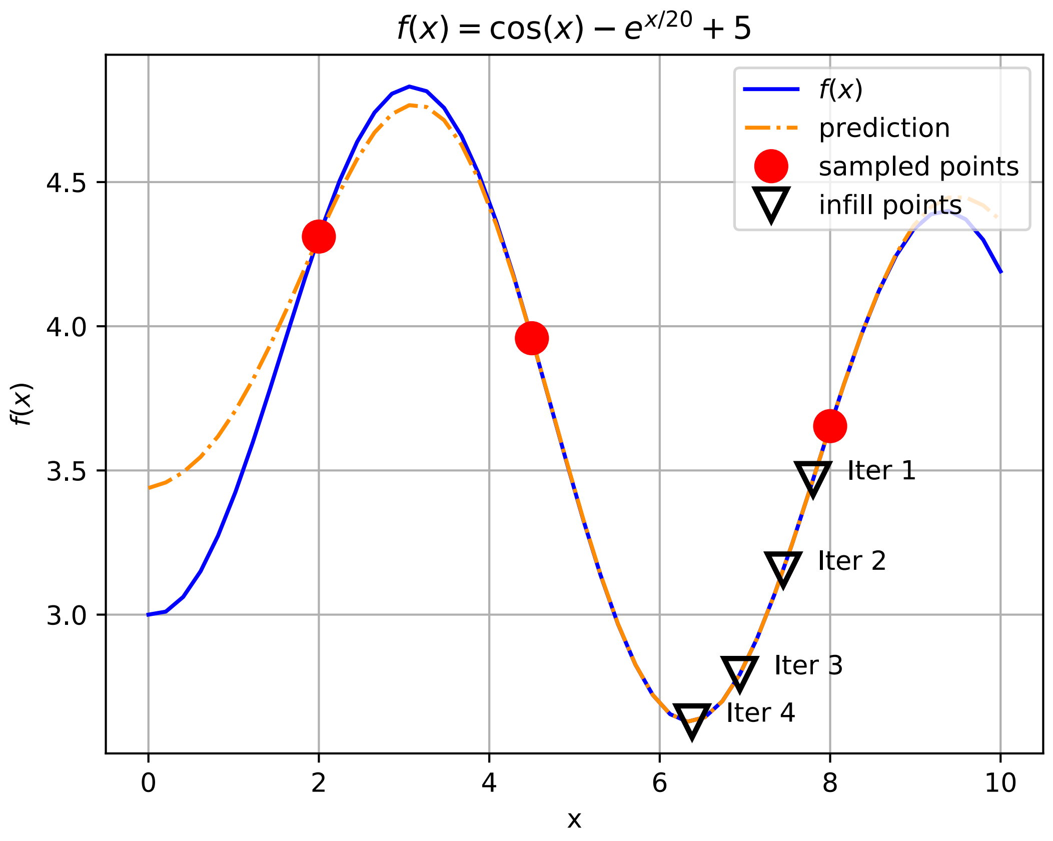


Source: Author.

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).

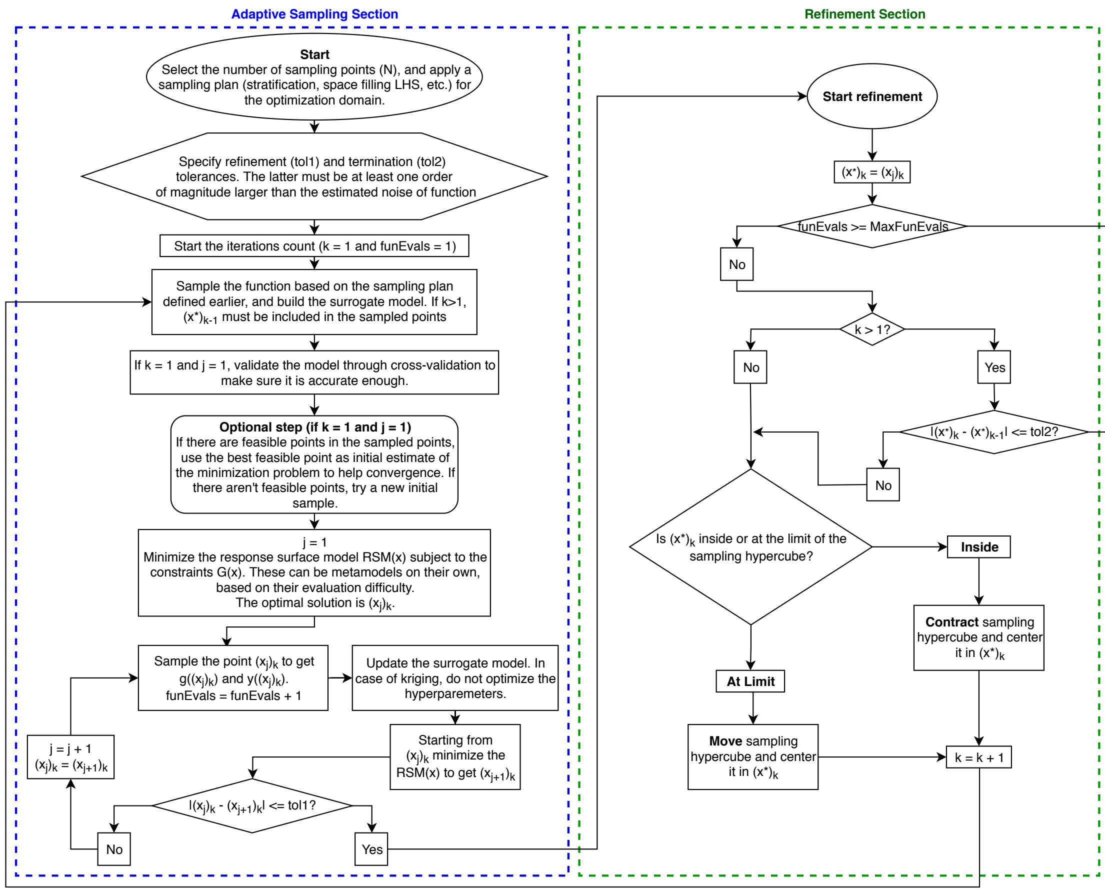
This approach was implemented as a procedure of the Python package *surropt* (from *Surrogate Optimization*). It uses as internal NLP solver a Python wrapper authored by Kummerer and Moore (2019) of the well-established *IpOpt* package (WÄCHTER; BIEGLER, 2006). The *surropt* package is found on `https://github.com/feslima/surropt`.

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 authors of this work and implemented within *Metacontrol*.


Adaptive Sampling Section
Refinement Section
Start
Select the number of sampling points (N), and apply a sampling plan (stratification, space filling LHS, etc.) for the optimization domain.
Specify refinement (tol1) and termination (tol2) tolerances. The latter must be at least one order of magnitude larger than the estimated noise of function
Start the iterations count (k = 1 and funEvals = 1)
Sample the function based on the sampling plan defined earlier, and build the surrogate model. If k>1, (x*)k-1 must be included in the sampled points
If k = 1 and j = 1, validate the model through cross-validation to make sure it is accurate enough.
Optional step (if k = 1 and j = 1)
If there are feasible points in the sampled points, use the best feasible point as initial estimate of the minimization problem to help convergence. If there aren't feasible points, try a new initial sample.
j = 1
Minimize the response surface model RSM(x) subject to the constraints G(x). These can be metamodels on their own, based on their evaluation difficulty.
The optimal solution is (xj)k.
Sample the point (xj)k to get g((xj)k) and y((xj)k). funEvals = funEvals + 1
Update the surrogate model. In case of kriging, do not optimize the hyperparemeters.
Starting from (xj)k minimize the RSM(x) to get (xj+1)k
j = j + 1
(xj)k = (xj+1)k
|(xj)k - (xj+1)k| <= tol1?
No
Yes
Start refinement
(x*)k = (xj)k
funEvals >= MaxFunEvals
No
Yes
k > 1?
No
Yes
|(x*)k - (x*)k-1| <= tol2?
No
Yes
END
Is (x*)k inside or at the limit of the sampling hypercube?
Inside
Contract sampling hypercube and center it in (x*)k
At Limit
Move sampling hypercube and center it in (x*)k
k = k + 1

Source: Alves et al. (2018)

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 Shigueo 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 amd 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:

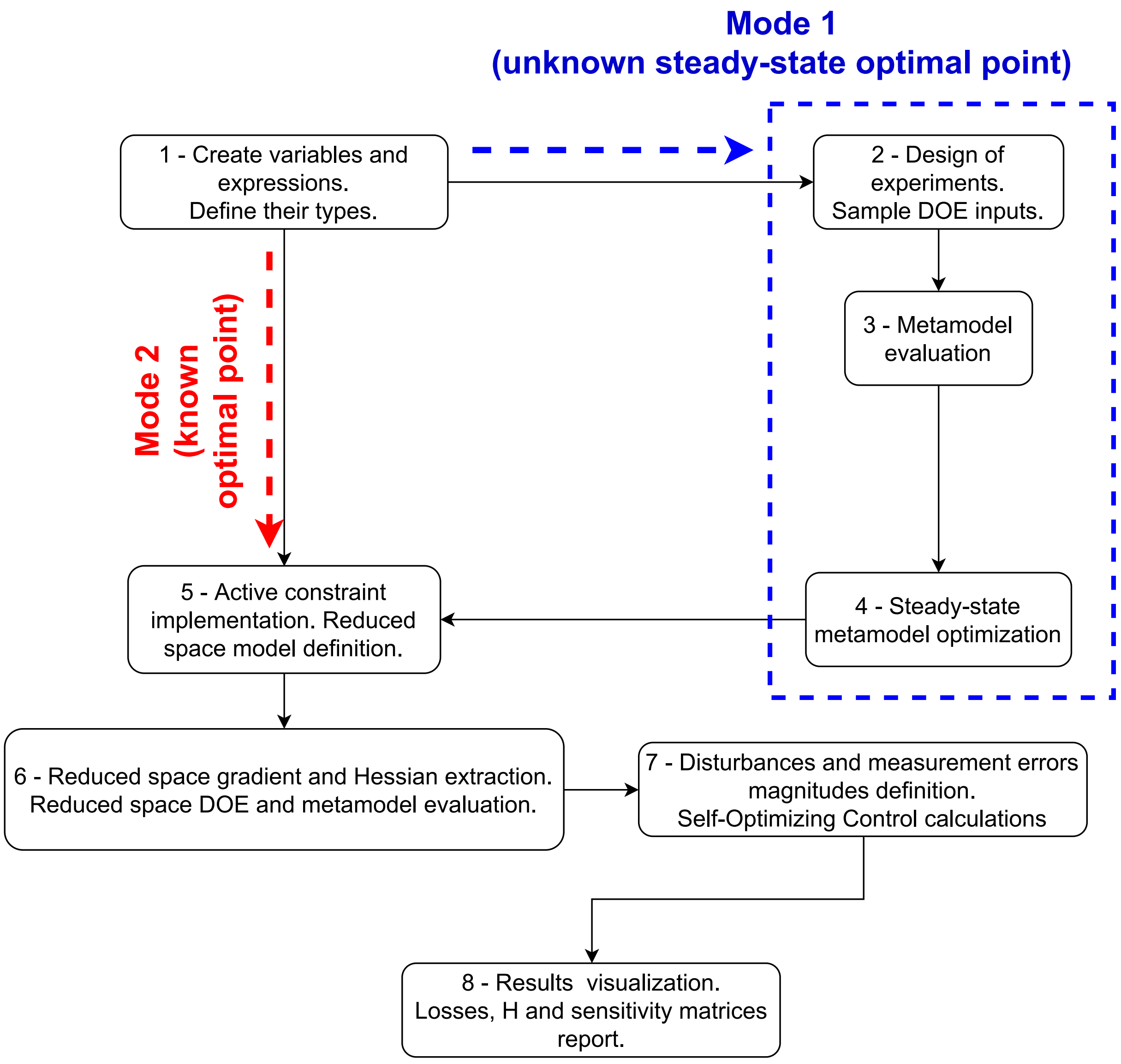
1. The user wishes to apply the methodology proposed by Alves et al. (2018) completely, as seen in Figure 5 represented by the blue dashed arrow and rectangle. That is, he needs to create and define variables and expressions (1), perform a Design of Experiments (DOE) of the industrial process (2), build 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.


Mode 1
(unknown steady-state optimal point)
1 - Create variables and expressions. Define their types.
2 - Design of experiments. Sample DOE inputs.
3 - Metamodel evaluation
4 - Steady-state metamodel optimization
Mode 2 (known optimal point)
5 - Active constraint implementation. Reduced space model definition.
6 - Reduced space gradient and Hessian extraction. Reduced space DOE and metamodel evaluation.
7 - Disturbances and measurement errors magnitudes definition. Self-Optimizing Control calculations
8 - Results visualization. Losses, H and sensitivity matrices report.

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 CO_2 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;

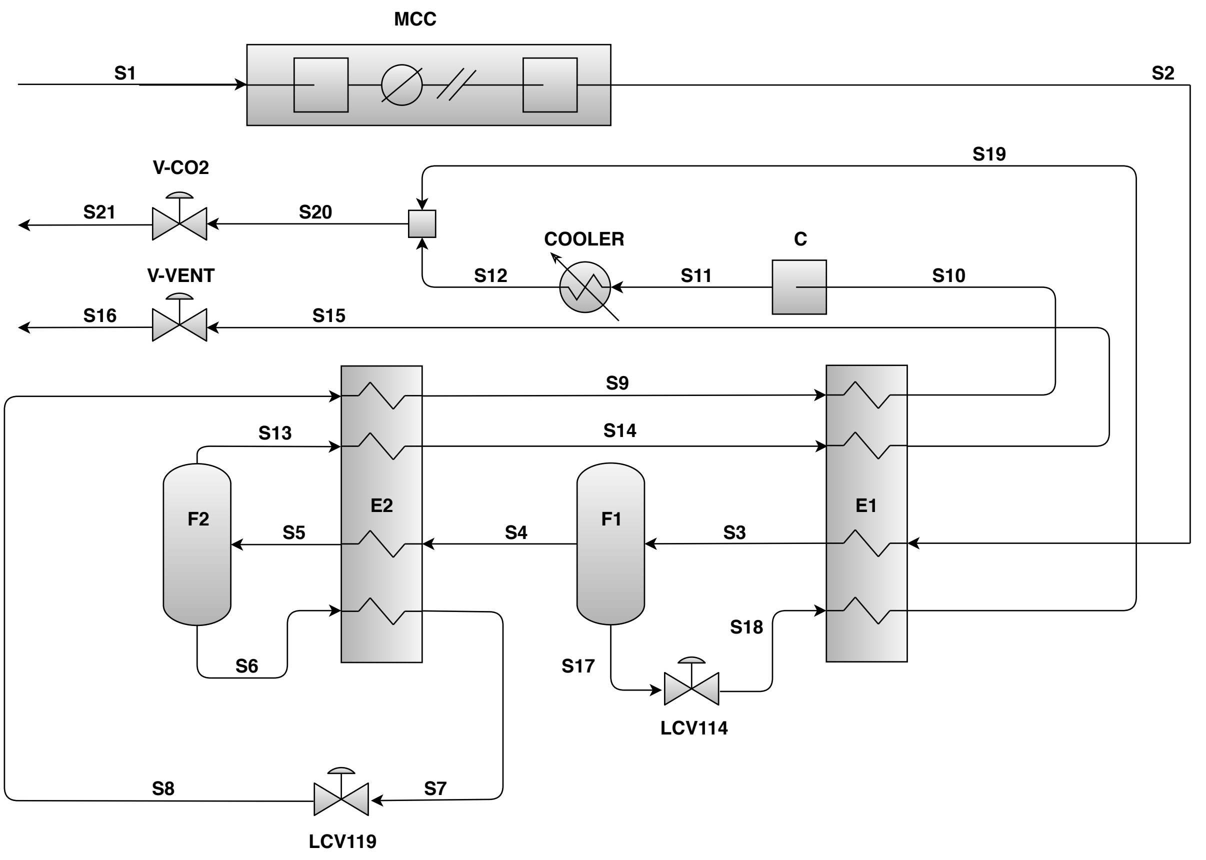
3. An isomerization process that seeks to convert n-butane into isobutane. The objective is maximization of profits;

6.1 The CO_2 Compression and Purification Unit (CPU)

The first case-study to be used as a test-bed in *Metacontrol* consists in a CO_2 compression and purification unit that uses phase separation method to obtain purified CO_2 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 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 CO_2 product streams and the vent gas are reheated on both multi-stream heat exchangers. The CO_2 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).

Figure 6 – CPU process flowsheet



Source: Author

From Liu et al. (2019), the authors of this paper 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 CO_2 flow rate produced. Therefore:

$$J = \frac{W_{\text{MCC}} + W_{\text{C}}}{F_{\text{CO}_2}} \tag{6.1}$$

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

Regarding the CPU process constraints, from Jin, Zhao, and Zheng (2015), Liu et al. (2019) and Dillon et al. (2005), the following apply:

- C-1: CO_2 recovery rate $\geq 90\%$
- C-2: CO_2 purity on product stream $\geq 96\%$
- C-3: Temperature of F2 bottom stream $> -56.6°C$

C-1 aims to meet the environmental requirements (LIU et al., 2019) and reduce CO_2 atmospheric emissions (TOFTEGAARD et al., 2010; BUHRE et al., 2005). C-2 is a result of the demand of CO_2 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 CO_2 solidification in the pipeline, since the value of C-3 corresponds to the CO_2 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: CO_2 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 CO_2 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 CO_2 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 CO_2 CPU process.

Variable (alias used in *Metacontrol*)	**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)

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

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 possibilites 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.

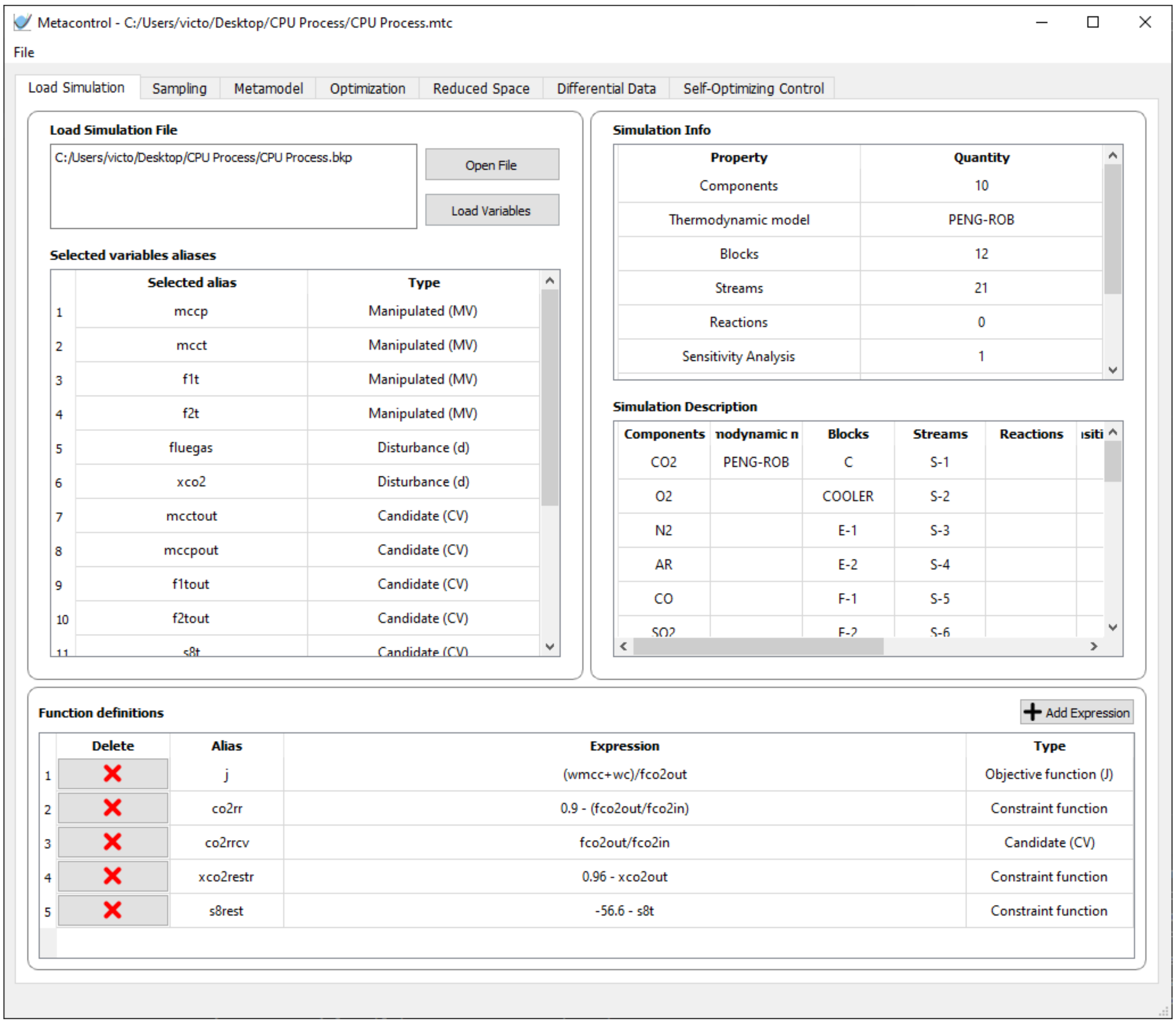
After selecting the relevant variables (Decision variables and process measurements) the user can go back to the main screen, where expressions can be created. This functionality aims to give freedom for the user to build expressions based on variables from the process simulator, such as: Objective functions, CV candidates or constraints. Figure 9 shows the specific power consumption, the CO_2 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.



C:/Users/victo/Desktop/CPU Process/CPU Process.bkp

	Selected alias	Type
1	mccp	Manipulated (MV)
2	mcct	Manipulated (MV)
3	f1t	Manipulated (MV)
4	f2t	Manipulated (MV)
5	fluegas	Disturbance (d)
6	xco2	Disturbance (d)
7	mcctout	Candidate (CV)
8	mccpout	Candidate (CV)
9	f1tout	Candidate (CV)
10	f2tout	Candidate (CV)
11	s8t	Candidate (CV)

Property	Quantity
Components	10
Thermodynamic model	PENG-ROB
Blocks	12
Streams	21
Reactions	0
Sensitivity Analysis	1

Components	nodynamic n	Blocks	Streams	Reactions	siti
CO2	PENG-ROB	C	S-1		
O2		COOLER	S-2		
N2		E-1	S-3		
AR		E-2	S-4		
CO		F-1	S-5		
SO2		F-2	S-6		

	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

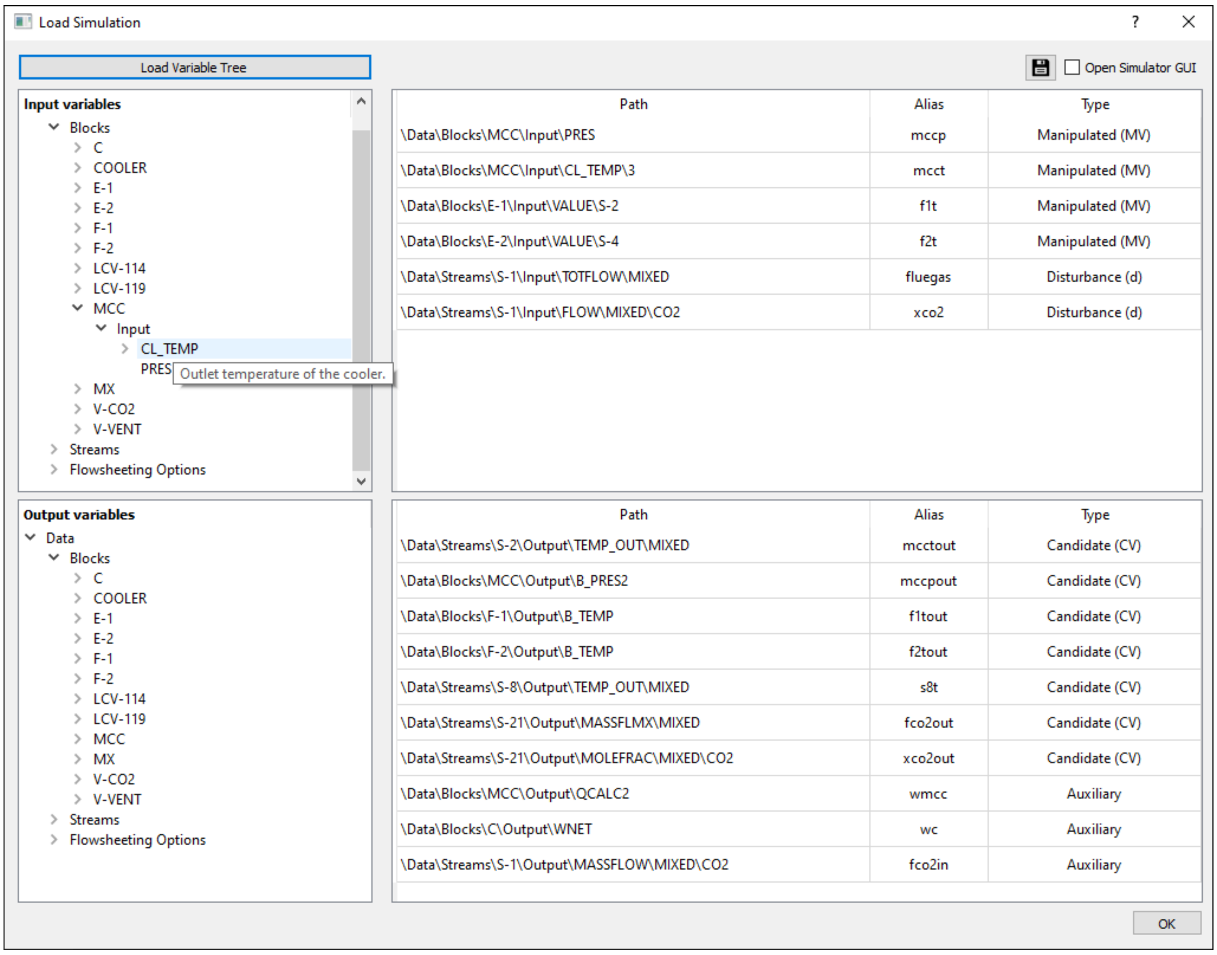
Source: Author.

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.



Path	Alias	Type
\Data\Blocks\MCC\Input\PRES	mccp	Manipulated (MV)
\Data\Blocks\MCC\Input\CL_TEMP\3	mcct	Manipulated (MV)
\Data\Blocks\E-1\Input\VALUE\S-2	f1t	Manipulated (MV)
\Data\Blocks\E-2\Input\VALUE\S-4	f2t	Manipulated (MV)
\Data\Streams\S-1\Input\TOTFLOW\MIXED	fluegas	Disturbance (d)
\Data\Streams\S-1\Input\FLOW\MIXED\CO2	xco2	Disturbance (d)

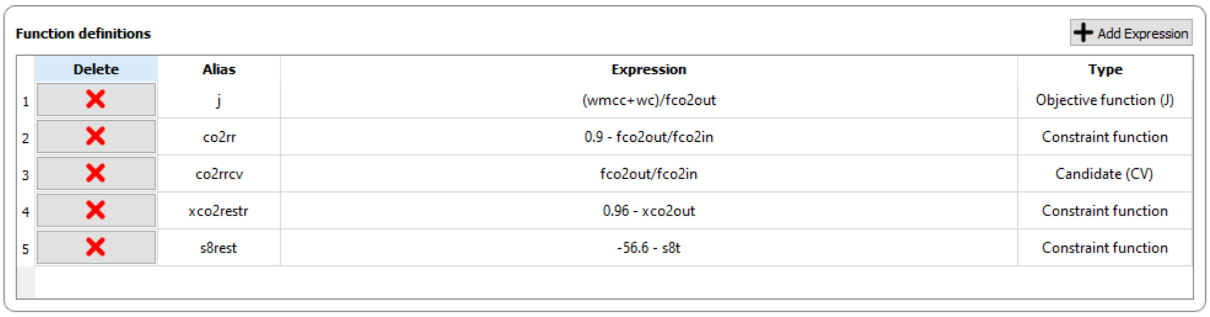
Path	Alias	Type
\Data\Streams\S-2\Output\TEMP_OUT\MIXED	mcctout	Candidate (CV)
\Data\Blocks\MCC\Output\B_PRES2	mccpout	Candidate (CV)
\Data\Blocks\F-1\Output\B_TEMP	f1tout	Candidate (CV)
\Data\Blocks\F-2\Output\B_TEMP	f2tout	Candidate (CV)
\Data\Streams\S-8\Output\TEMP_OUT\MIXED	s8t	Candidate (CV)
\Data\Streams\S-21\Output\MASSFLMX\MIXED	fco2out	Candidate (CV)
\Data\Streams\S-21\Output\MOLEFRAC\MIXED\CO2	xco2out	Candidate (CV)
\Data\Blocks\MCC\Output\QCALC2	wmcc	Auxiliary
\Data\Blocks\C\Output\WNET	wc	Auxiliary
\Data\Streams\S-1\Output\MASSFLOW\MIXED\CO2	fco2in	Auxiliary

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), CO_2 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

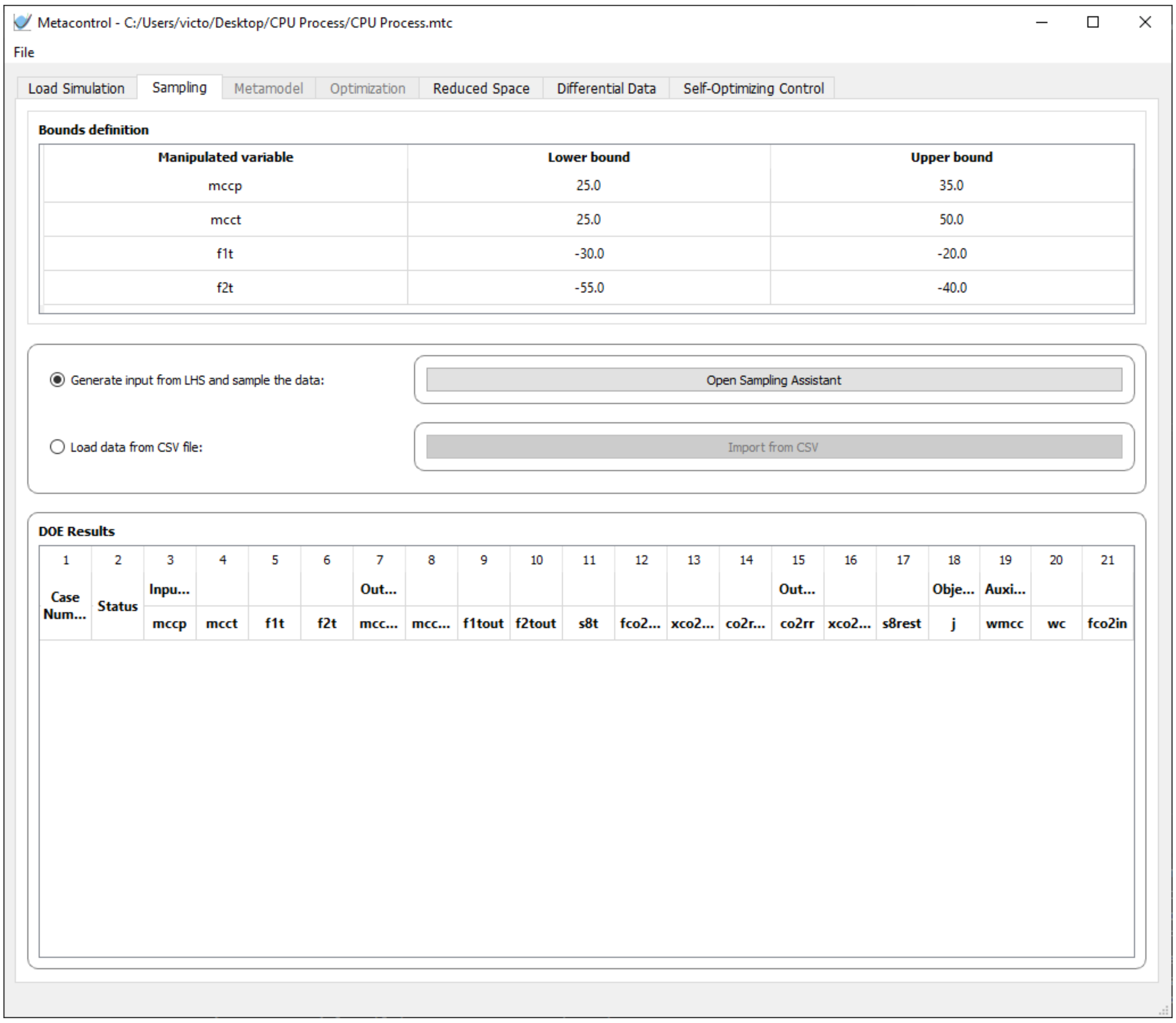
- MCC outlet pressure (bar) = 30.1849
- MCC outlet temperature = 25 °C
- F1 temperature = -30 °C
- F2 temperature = -55°C

That indicates three active constraints that have to be controlled. Regarding stream S8 temperature, CO_2 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.



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

File

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

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

(●) Generate input from LHS and sample the data: [Open Sampling Assistant]

( ) Load data from CSV file: [Import from CSV]

DOE Results

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Case Num...	Status	Inpu...				Out...								Out...			Obje...	Auxi...		
		mccp	mcct	f1t	f2t	mcc...	mcc...	f1tout	f2tout	s8t	fco2...	xco2...	co2r...	co2rr	xco2...	s8rest	j	wmcc	wc	fco2in

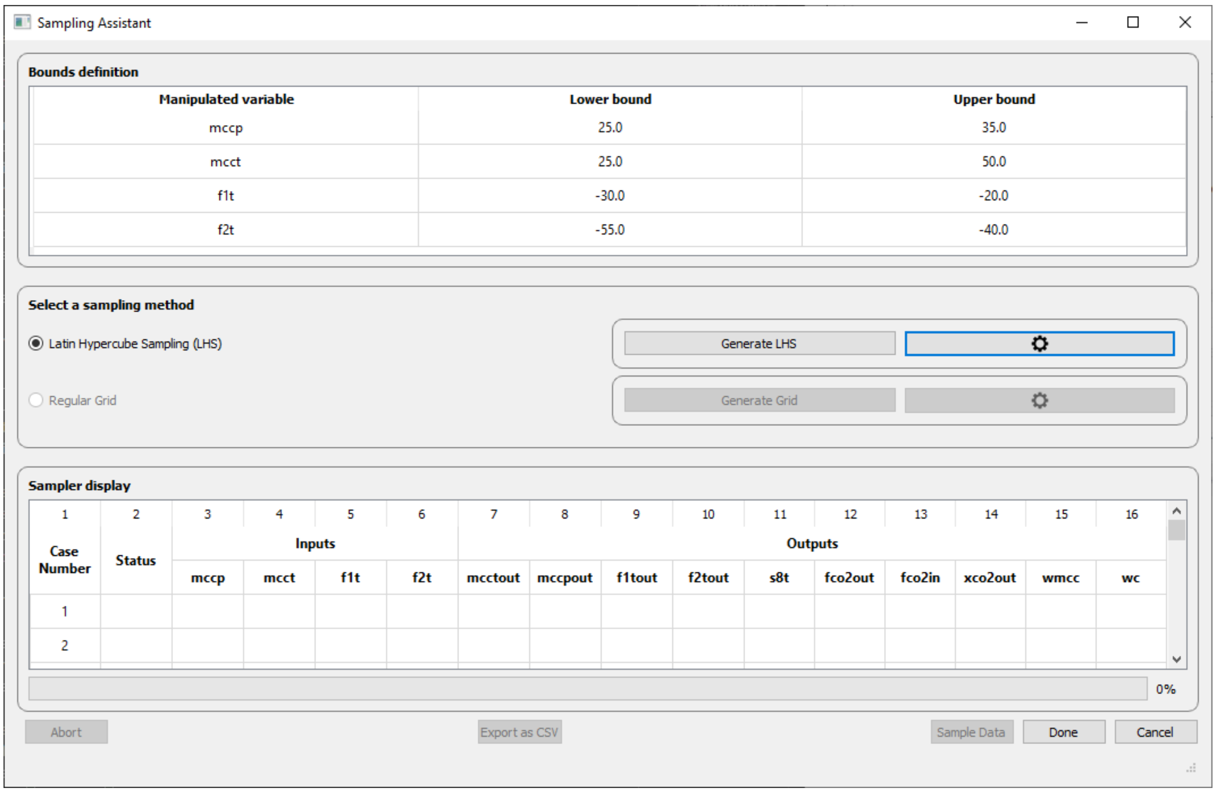
Table 2 – Optimization runs: *Aspen Plus vs Metacontrol* - Decision variables and objective function - CPU Process

	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 *.bkp, 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)



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)

○ Regular Grid

Sampler display

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Case Number	Status	Inputs				Outputs									
		mccp	mcct	f1t	f2t	mcctout	mccpout	f1tout	f2tout	s8t	fco2out	fco2in	xco2out	wmcc	wc
1															
2															

0%

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

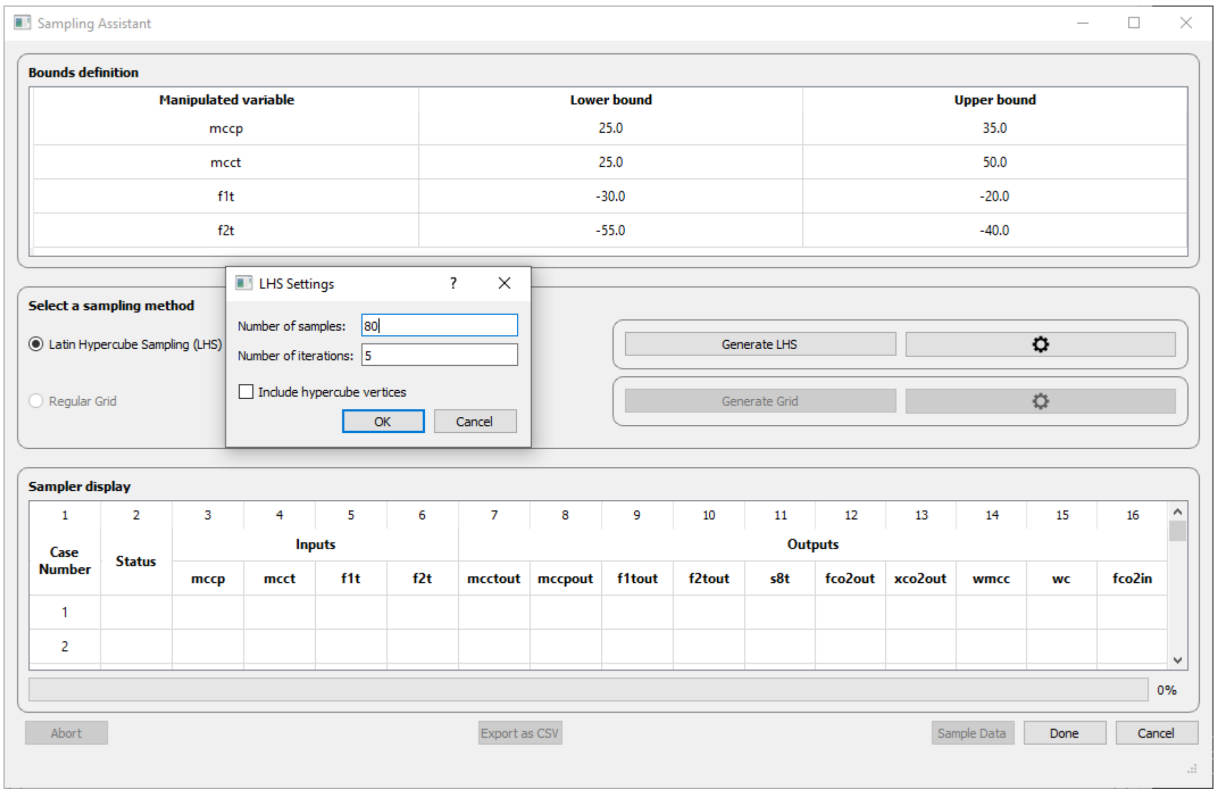
	Stream S8 temperature °C	CO_2 molar fraction	CO_2 recovery rate
Aspen Plus	-55.8201	0.9674	0.9658
Metacontrol	-55.4859	0.9666	0.9671

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 (±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.



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

Number of samples: 80

Number of iterations: 5

☐ Include hypercube vertices

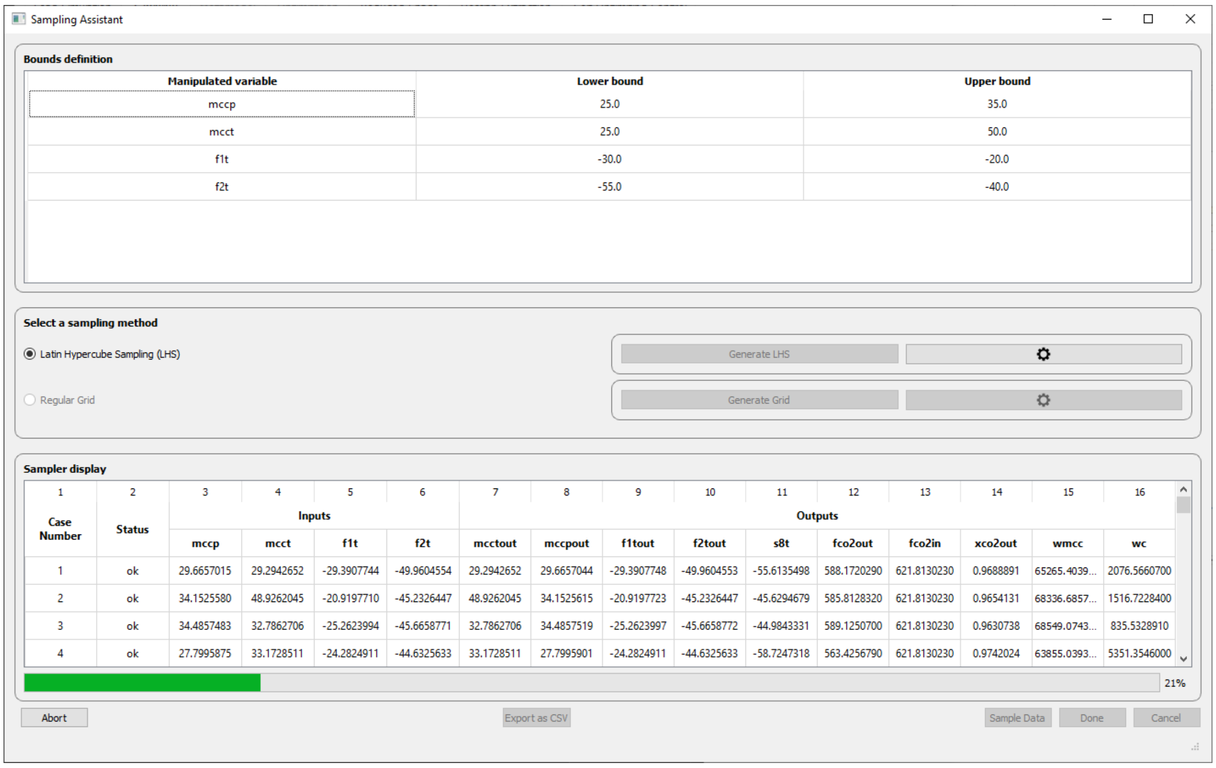
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Case Number	Status	Inputs				Outputs									
		mccp	mcct	f1t	f2t	mcctout	mccpout	f1tout	f2tout	s8t	fco2out	xco2out	wmcc	wc	fco2in
1															
2															

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 (*numdifftools* 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

[● Latin Hypercube Sampling (LHS)] Regular Grid

Sampler display

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
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%

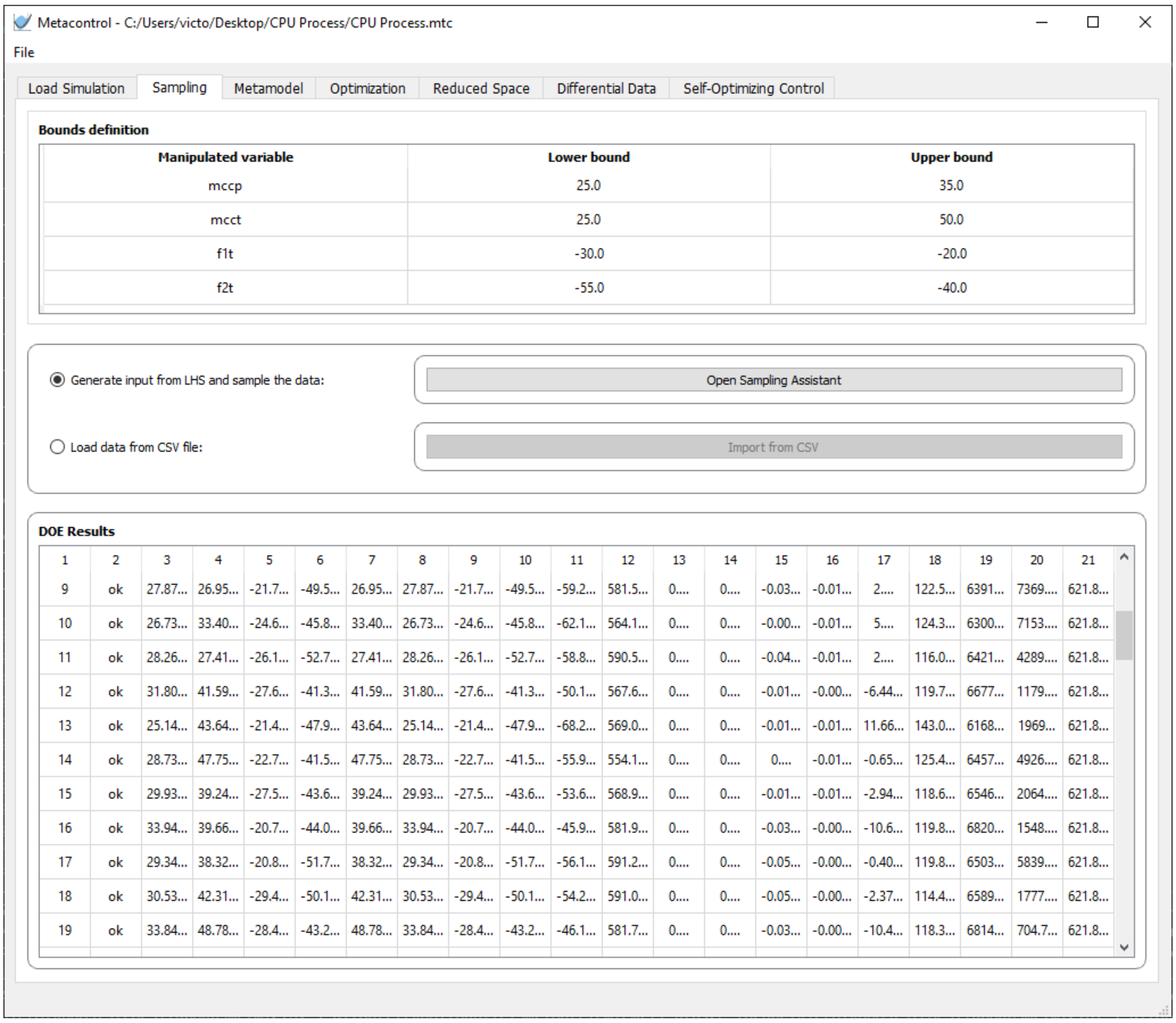
and generate the estimation of the gradients and hessians necessary to carry on the Self-Optimizing Control study. These results are displayed on Figure 24.

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 ±5% for the disturbances. For the CO_2 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.



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

[✓ Generate input from LHS and sample the data:]

Load data from CSV file:

DOE Results

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
9	ok	27.87...	26.95...	-21.7...	-49.5...	26.95...	27.87...	-21.7...	-49.5...	-59.2...	581.5...	0....	0....	-0.03...	-0.01...	2....	122.5...	6391...	7369....	621.8...
10	ok	26.73...	33.40...	-24.6...	-45.8...	33.40...	26.73...	-24.6...	-45.8...	-62.1...	564.1...	0....	0....	-0.00...	-0.01...	5....	124.3...	6300...	7153....	621.8...
11	ok	28.26...	27.41...	-26.1...	-52.7...	27.41...	28.26...	-26.1...	-52.7...	-58.8...	590.5...	0....	0....	-0.04...	-0.01...	2....	116.0...	6421...	4289....	621.8...
12	ok	31.80...	41.59...	-27.6...	-41.3...	41.59...	31.80...	-27.6...	-41.3...	-50.1...	567.6...	0....	0....	-0.01...	-0.00...	-6.44...	119.7...	6677...	1179....	621.8...
13	ok	25.14...	43.64...	-21.4...	-47.9...	43.64...	25.14...	-21.4...	-47.9...	-68.2...	569.0...	0....	0....	-0.01...	-0.01...	11.66...	143.0...	6168...	1969...	621.8...
14	ok	28.73...	47.75...	-22.7...	-41.5...	47.75...	28.73...	-22.7...	-41.5...	-55.9...	554.1...	0....	0....	0....	-0.01...	-0.65...	125.4...	6457...	4926....	621.8...
15	ok	29.93...	39.24...	-27.5...	-43.6...	39.24...	29.93...	-27.5...	-43.6...	-53.6...	568.9...	0....	0....	-0.01...	-0.01...	-2.94...	118.6...	6546...	2064....	621.8...
16	ok	33.94...	39.66...	-20.7...	-44.0...	39.66...	33.94...	-20.7...	-44.0...	-45.9...	581.9...	0....	0....	-0.03...	-0.00...	-10.6...	119.8...	6820...	1548....	621.8...
17	ok	29.34...	38.32...	-20.8...	-51.7...	38.32...	29.34...	-20.8...	-51.7...	-56.1...	591.2...	0....	0....	-0.05...	-0.00...	-0.40...	119.8...	6503...	5839....	621.8...
18	ok	30.53...	42.31...	-29.4...	-50.1...	42.31...	30.53...	-29.4...	-50.1...	-54.2...	591.0...	0....	0....	-0.05...	-0.00...	-2.37...	114.4...	6589...	1777....	621.8...
19	ok	33.84...	48.78...	-28.4...	-43.2...	48.78...	33.84...	-28.4...	-43.2...	-46.1...	581.7...	0....	0....	-0.03...	-0.00...	-10.4...	118.3...	6814...	704.7...	621.8...

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

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

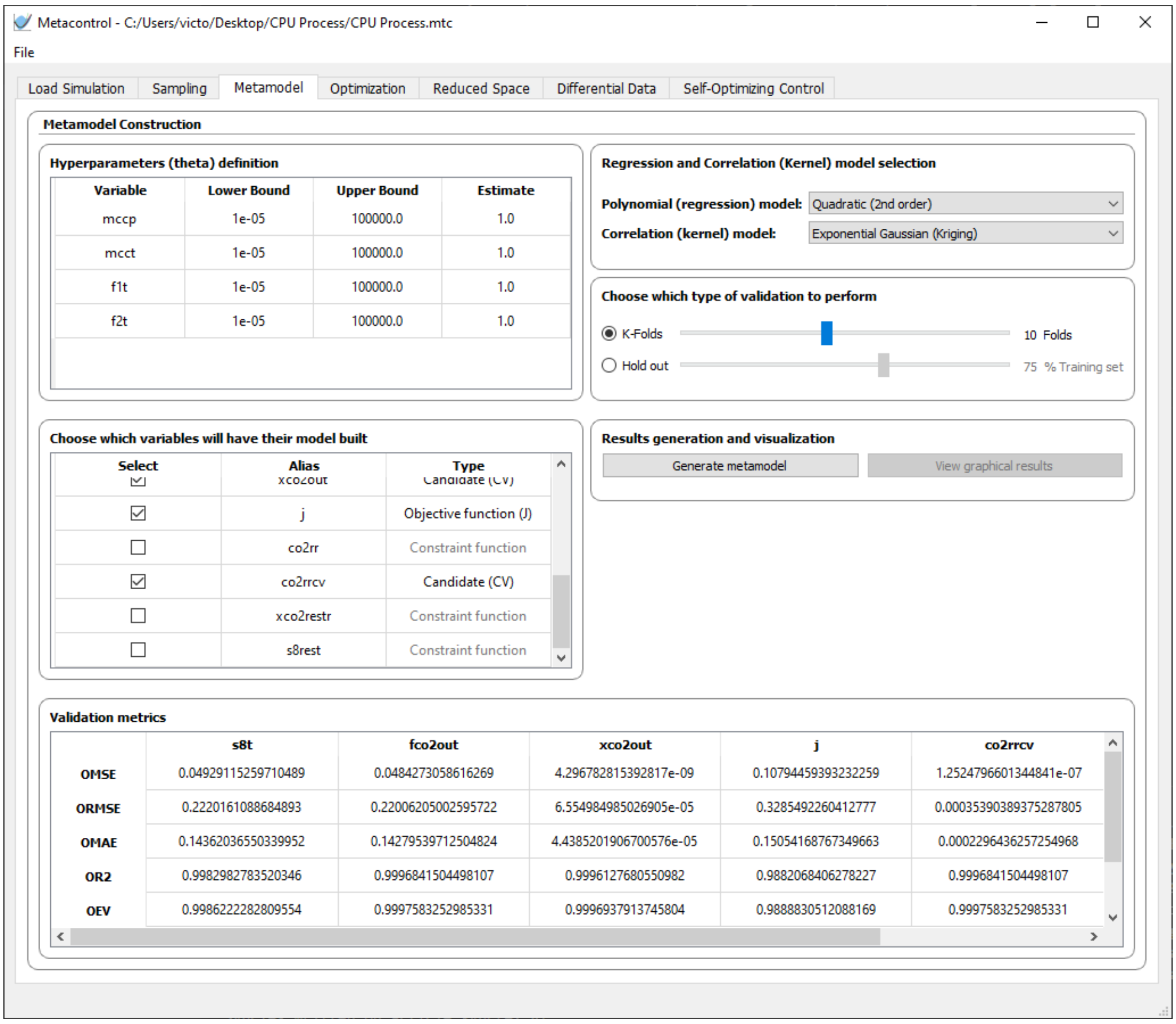
In addition, for the measurement errors, it was considered ±0.5°C for temperature measurements, 0.01 for pressure and flow measurements and 0.001 for ratios (CO_2 recovery rate and product purity). These assumptions generated by Equation 6.4:

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

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.


Metacontrol - C:/Users/victo/Desktop/CPU Process/CPU Process.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
mccp	1e-05	100000.0	1.0
mcct	1e-05	100000.0	1.0
f1t	1e-05	100000.0	1.0
f2t	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
✓	xco2out	Candidate (CV)
✓	j	Objective function (J)
☐	co2rr	Constraint function
✓	co2rrcv	Candidate (CV)
☐	xco2restr	Constraint function
☐	s8rest	Constraint function

Results generation and visualization

Generate metamodel | View graphical results

Validation metrics

	s8t	fco2out	xco2out	j	co2rrcv
OMSE	0.04929115259710489	0.0484273058616269	4.296782815392817e-09	0.10794459393232259	1.2524796601344841e-07
ORMSE	0.2220161088684893	0.22006205002595722	6.554984985026905e-05	0.3285492260412777	0.00035390389375287805
OMAE	0.14362036550339952	0.14279539712504824	4.4385201906700576e-05	0.15054168767349663	0.0002296436257254968
OR2	0.9982982783520346	0.9996841504498107	0.9996127680550982	0.9882068406278227	0.9996841504498107
OEV	0.9986222282809554	0.9997583252985331	0.9996937913745804	0.9888830512088169	0.9997583252985331

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 *nth* 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.

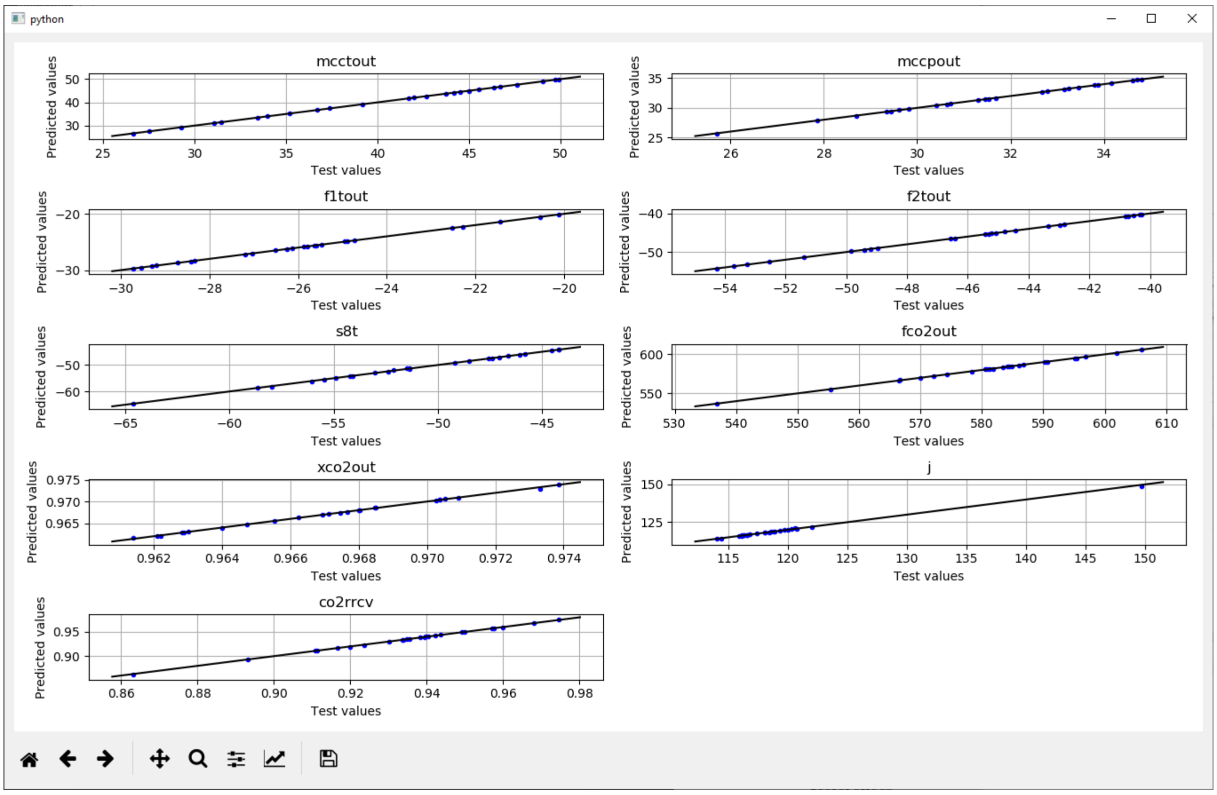


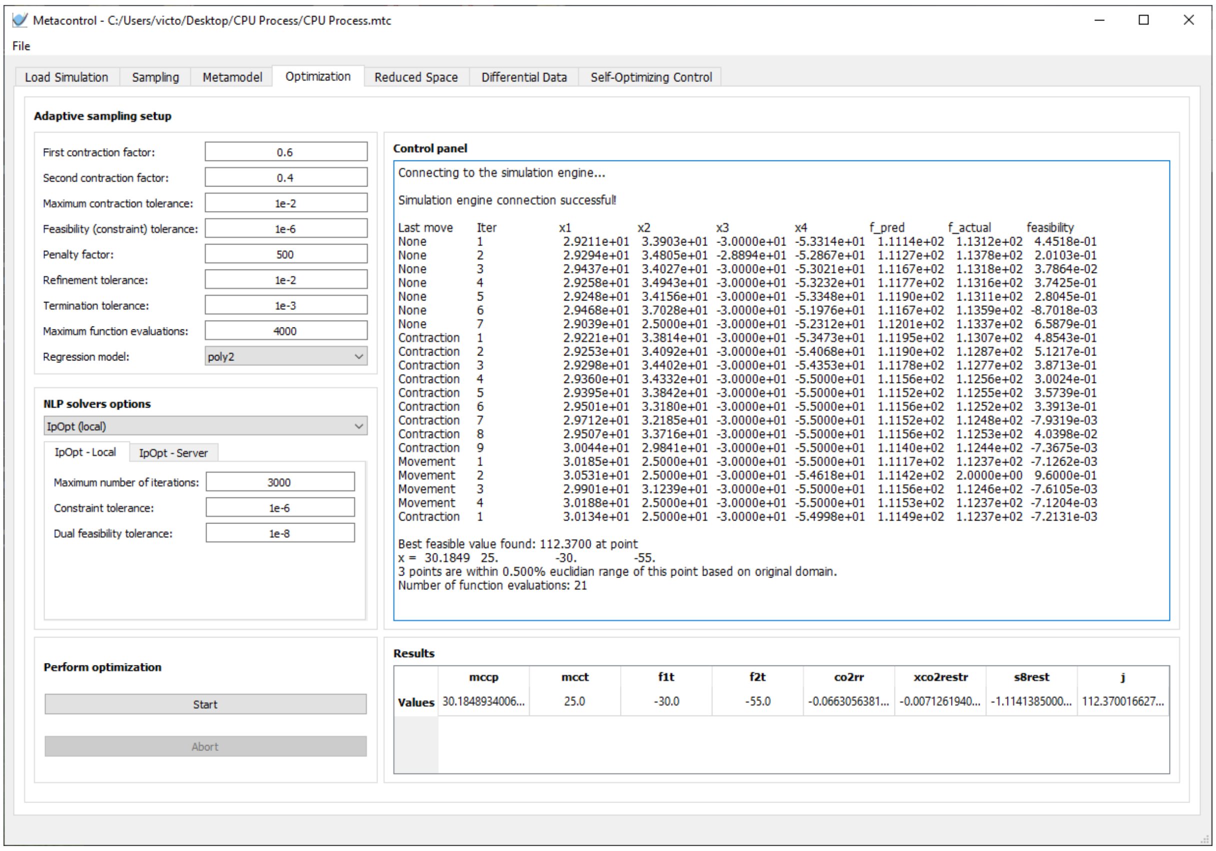
Table 4 – High-order data obtainment: *Aspen Plus vs Metacontrol*

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

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

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

Figure 17 – Refinement algorithm configuration and results screen.



Adaptive sampling setup

First contraction factor:	0.6
Second contraction factor:	0.4
Maximum contraction tolerance:	1e-2
Feasibility (constraint) tolerance:	1e-6
Penalty factor:	500
Refinement tolerance:	1e-2
Termination tolerance:	1e-3
Maximum function evaluations:	4000
Regression model:	poly2

NLP solvers options

IpOpt (local)

IpOpt - Local | IpOpt - Server

Maximum number of iterations:	3000
Constraint tolerance:	1e-6
Dual feasibility tolerance:	1e-8

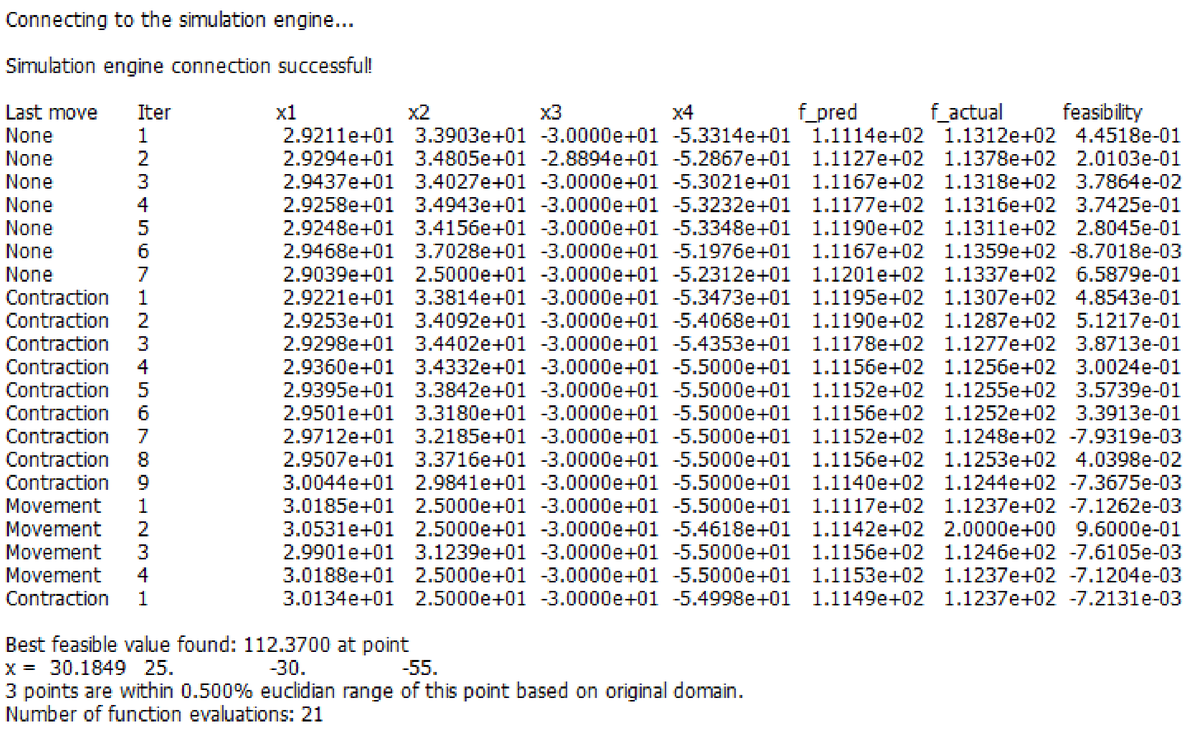
Control panel

```
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
```

Results

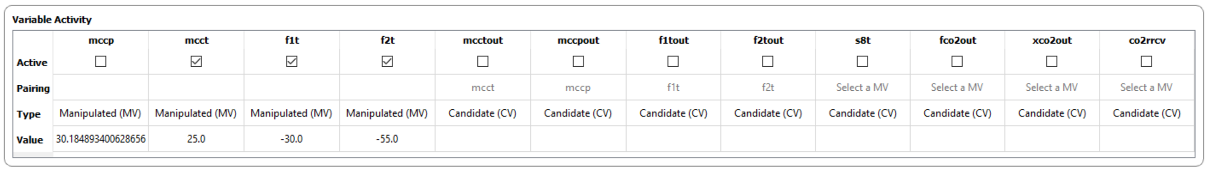
	mccp	mcct	f1t	f2t	co2rr	xco2restr	s8rest	j
Values	30.1848934006...	25.0	-30.0	-55.0	-0.0663056381...	-0.0071261940...	-1.1141385000...	112.370016627...

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

	mccp	mcct	f1t	f2t	mcctout	mccpout	f1tout	f2tout	s8t	fco2out	xco2out	co2rrcv
Active	☐	☑	☑	☑	☐	☐	☐	☐	☐	☐	☐	☐
Pairing					mcct	mccp	f1t	f2t	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.

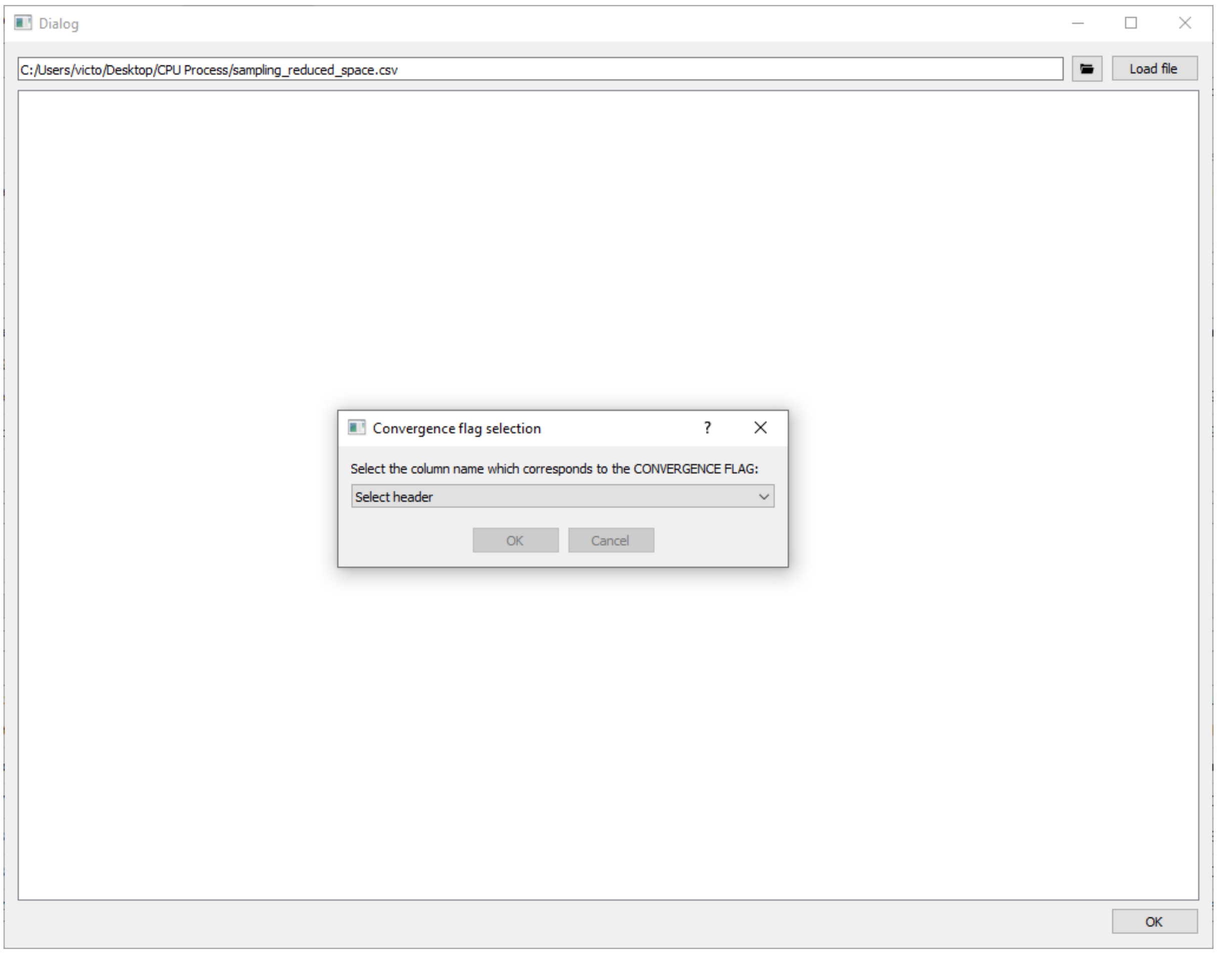
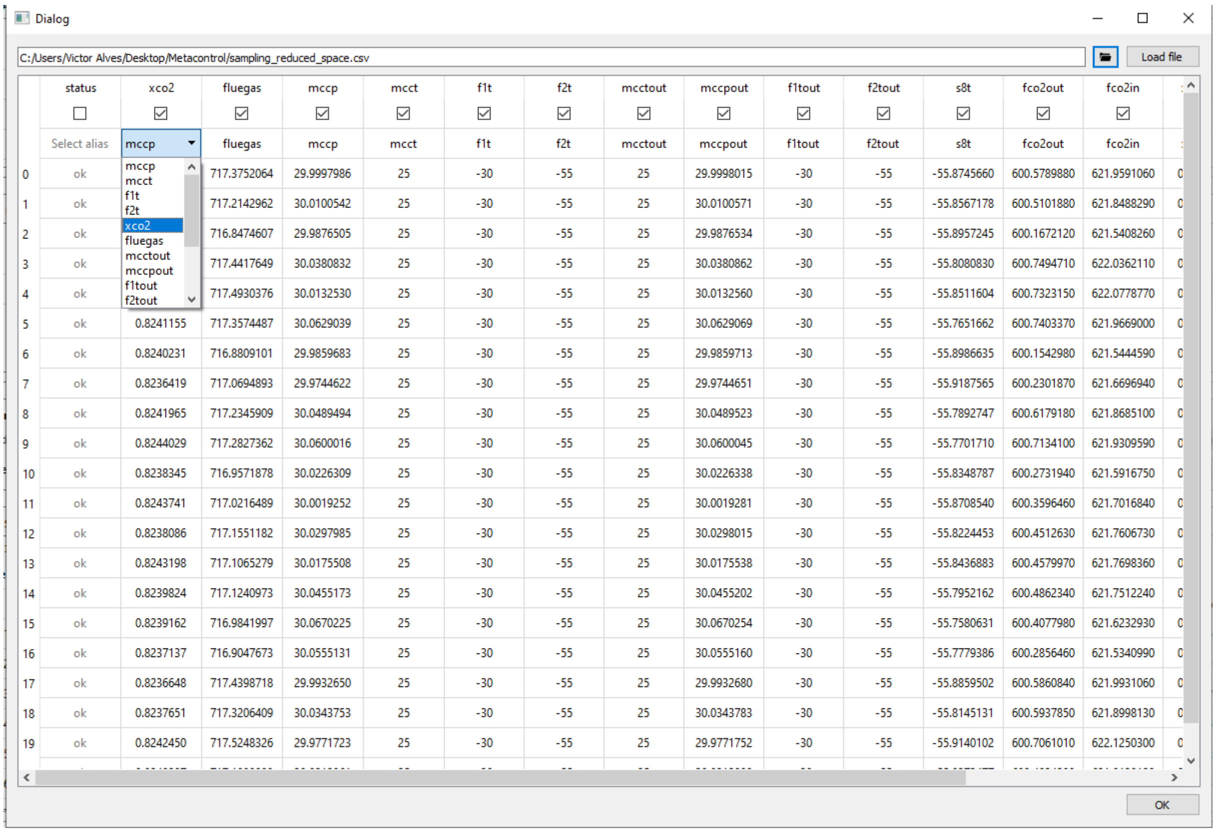

Dialog
C:/Users/victo/Desktop/CPU Process/sampling_reduced_space.csv
Load file
Convergence flag selection
Select the column name which corresponds to the CONVERGENCE FLAG:
Select header
OK
Cancel
OK

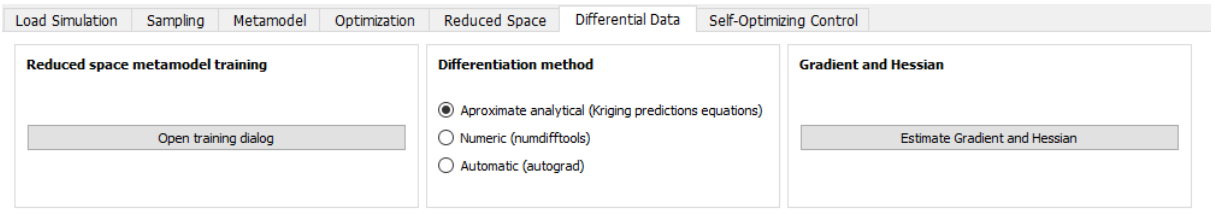
Figure 21 – Associating each alias created in *Metacontrol* to each column of the *.csv data.



C:/Users/Victor Alves/Desktop/Metacontrol/sampling_reduced_space.csv

	status	xco2	fluegas	mccp	mcct	f1t	f2t	mcctout	mccpout	f1tout	f2tout	s8t	fco2out	fco2in
	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
	Select alias	mccp	fluegas	mccp	mcct	f1t	f2t	mcctout	mccpout	f1tout	f2tout	s8t	fco2out	fco2in
0	ok		717.3752064	29.9997986	25	-30	-55	25	29.9998015	-30	-55	-55.8745660	600.5789880	621.9591060
1	ok		717.2142962	30.0100542	25	-30	-55	25	30.0100571	-30	-55	-55.8567178	600.5101880	621.8488290
2	ok		716.8474607	29.9876505	25	-30	-55	25	29.9876534	-30	-55	-55.8957245	600.1672120	621.5408260
3	ok		717.4417649	30.0380832	25	-30	-55	25	30.0380862	-30	-55	-55.8080830	600.7494710	622.0362110
4	ok		717.4930376	30.0132530	25	-30	-55	25	30.0132560	-30	-55	-55.8511604	600.7323150	622.0778770
5	ok	0.8241155	717.3574487	30.0629039	25	-30	-55	25	30.0629069	-30	-55	-55.7651662	600.7403370	621.9669000
6	ok	0.8240231	716.8809101	29.9859683	25	-30	-55	25	29.9859713	-30	-55	-55.8986635	600.1542980	621.5444590
7	ok	0.8236419	717.0694893	29.9744622	25	-30	-55	25	29.9744651	-30	-55	-55.9187565	600.2301870	621.6696940
8	ok	0.8241965	717.2345909	30.0489494	25	-30	-55	25	30.0489523	-30	-55	-55.7892747	600.6179180	621.8685100
9	ok	0.8244029	717.2827362	30.0600016	25	-30	-55	25	30.0600045	-30	-55	-55.7701710	600.7134100	621.9309590
10	ok	0.8238345	716.9571878	30.0226309	25	-30	-55	25	30.0226338	-30	-55	-55.8348787	600.2731940	621.5916750
11	ok	0.8243741	717.0216489	30.0019252	25	-30	-55	25	30.0019281	-30	-55	-55.8708540	600.3596460	621.7016840
12	ok	0.8238086	717.1551182	30.0297985	25	-30	-55	25	30.0298015	-30	-55	-55.8224453	600.4512630	621.7606730
13	ok	0.8243198	717.1065279	30.0175508	25	-30	-55	25	30.0175538	-30	-55	-55.8436883	600.4579970	621.7698360
14	ok	0.8239824	717.1240973	30.0455173	25	-30	-55	25	30.0455202	-30	-55	-55.7952162	600.4862340	621.7512240
15	ok	0.8239162	716.9841997	30.0670225	25	-30	-55	25	30.0670254	-30	-55	-55.7580631	600.4077980	621.6232930
16	ok	0.8237137	716.9047673	30.0555131	25	-30	-55	25	30.0555160	-30	-55	-55.7779386	600.2856460	621.5340990
17	ok	0.8236648	717.4398718	29.9932650	25	-30	-55	25	29.9932680	-30	-55	-55.8859502	600.5860840	621.9931060
18	ok	0.8237651	717.3206409	30.0343753	25	-30	-55	25	30.0343783	-30	-55	-55.8145131	600.5937850	621.8998130
19	ok	0.8242450	717.5248326	29.9771723	25	-30	-55	25	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.



Reduced space metamodel training

Differentiation method

● Aproximate analytical (Kriging predictions equations)
○ Numeric (numdifftools)
○ Automatic (autograd)

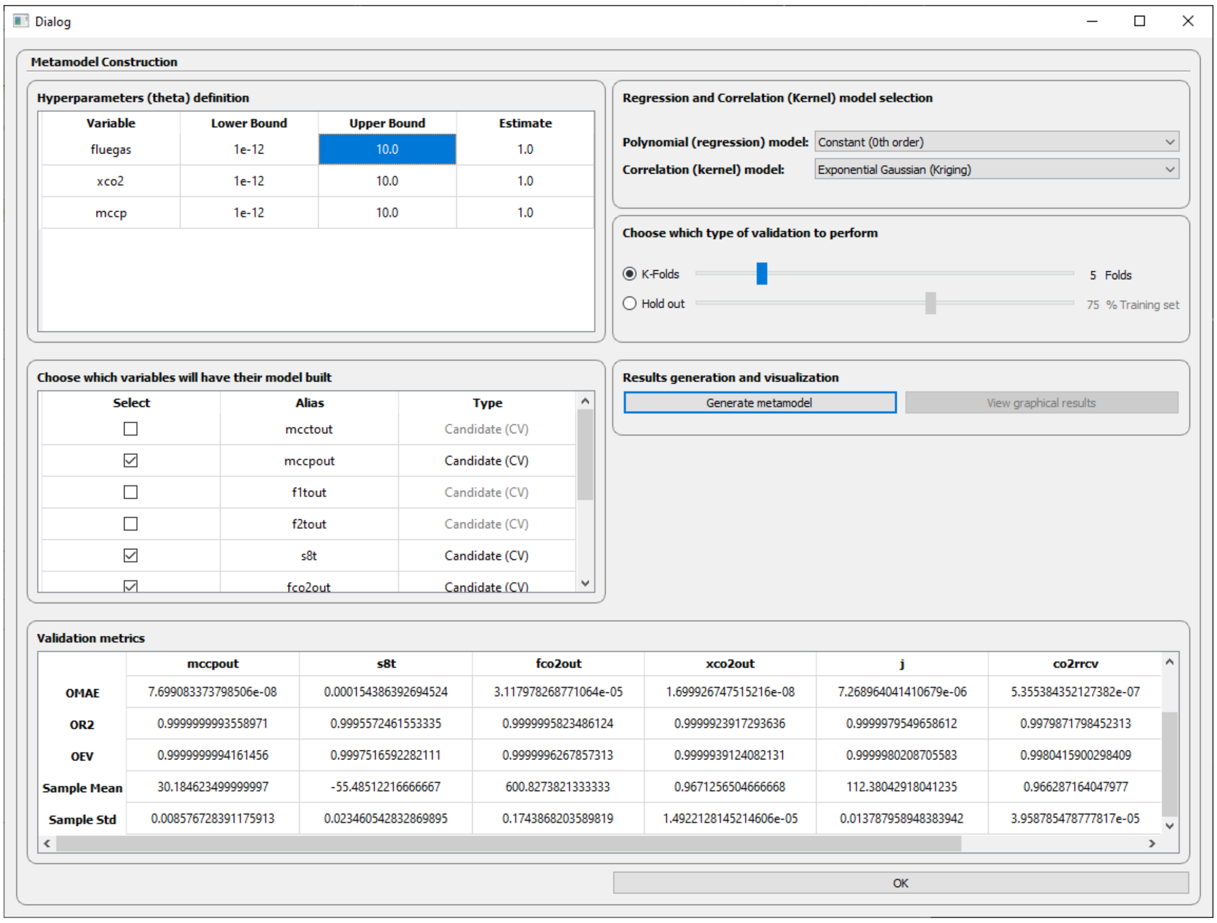
Gradient and Hessian

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

CV Candidate *alias*	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.

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.



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

Choose which variables will have their model built

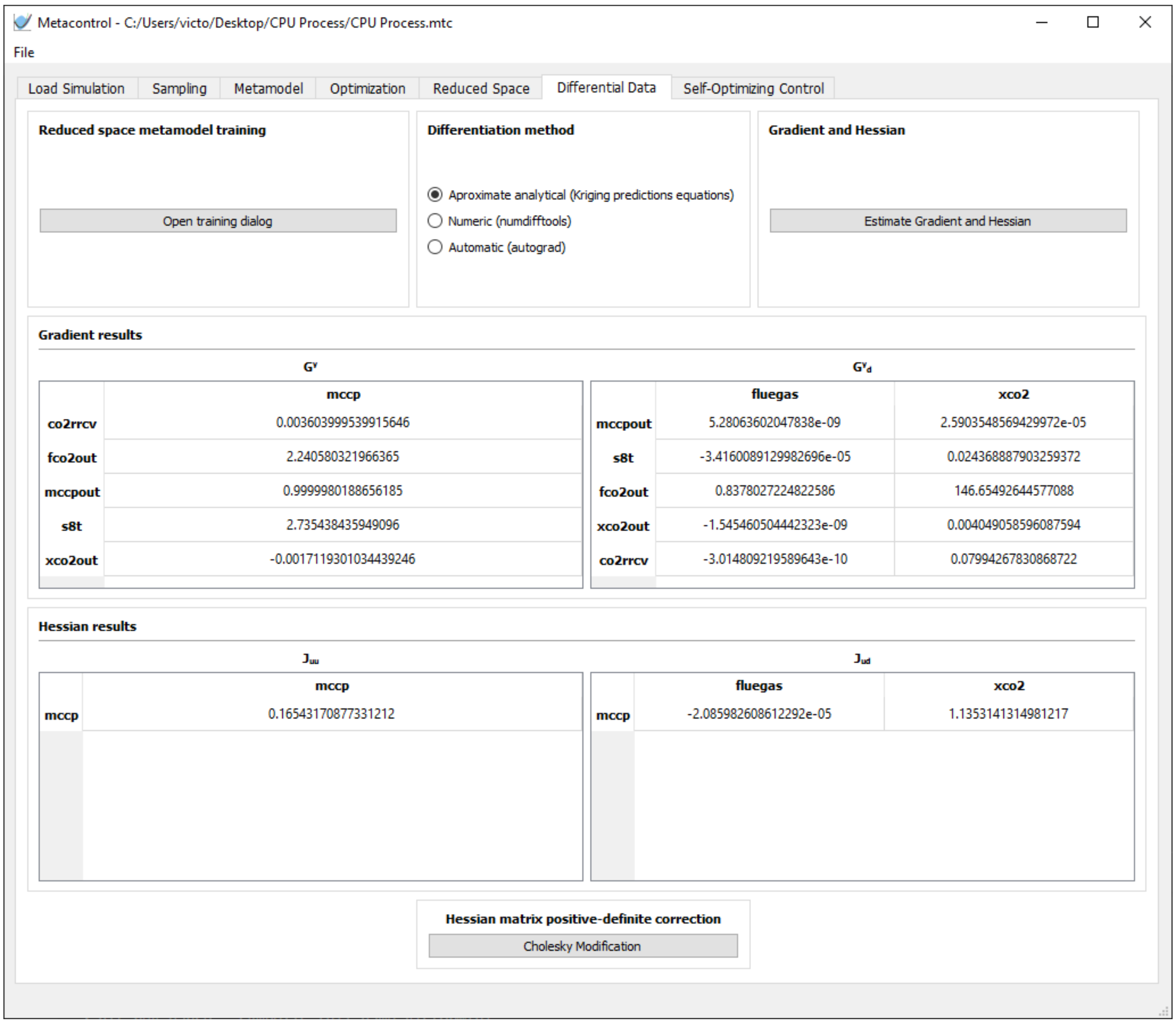
Select	Alias	Type
☐	mcctout	Candidate (CV)
☑	mccpout	Candidate (CV)
☐	f1tout	Candidate (CV)
☐	f2tout	Candidate (CV)
☑	s8t	Candidate (CV)
☑	fco2out	Candidate (CV)

Results generation and visualization

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	0.9999999993558971	0.9995572461553335	0.9999995823486124	0.9999923917293636	0.9999979549658612	0.9979871798452313
OEV	0.9999999994161456	0.9997516592282111	0.9999996267857313	0.9999939124082131	0.9999980208705583	0.9980415900298409
Sample Mean	30.184623499999997	-55.48512216666667	600.8273821333333	0.9671256504666668	112.38042918041235	0.966287164047977
Sample Std	0.008576728391175913	0.023460542832869895	0.1743868203589819	1.4922128145214606e-05	0.013787958948383942	3.958785478777817e-05

Figure 24 – Differential data estimated in *Metacontrol*



Reduced space metamodel training

Differentiation method

[✓ Aproximate analytical (Kriging predictions equations)]
Numeric (numdifftools)
Automatic (autograd)

Gradient and Hessian

Gradient results

G^y

	mccp
co2rrcv	0.003603999539915646
fco2out	2.240580321966365
mccpout	0.9999980188656185
s8t	2.735438435949096
xco2out	-0.0017119301034439246

G^y_d

	fluegas	xco2
mccpout	5.28063602047838e-09	2.5903548569429972e-05
s8t	-3.4160089129982696e-05	0.024368887903259372
fco2out	0.8378027224822586	146.65492644577088
xco2out	-1.545460504442323e-09	0.004049058596087594
co2rrcv	-3.014809219589643e-10	0.07994267830868722

Hessian results

J_{uu}

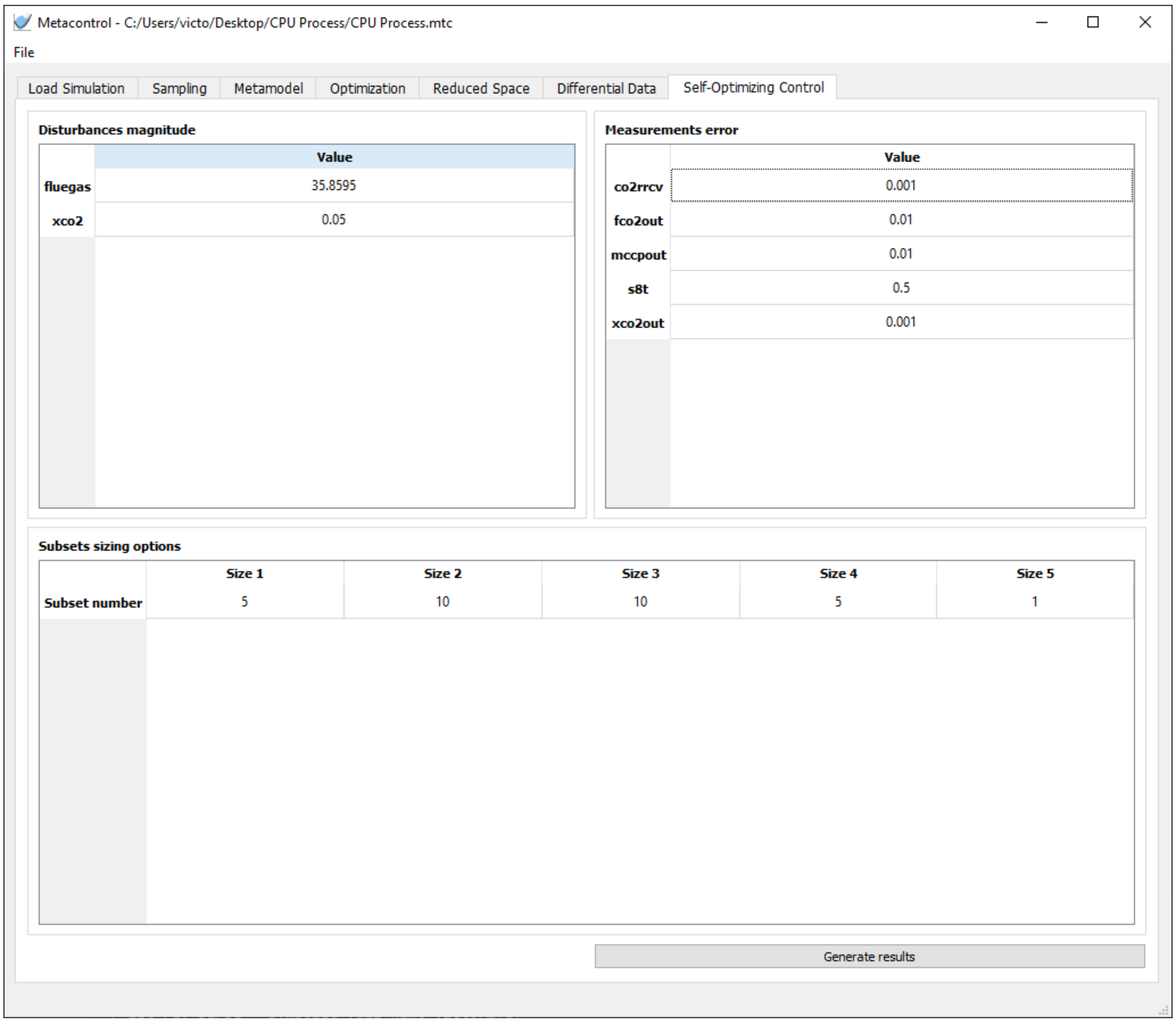
	mccp
mccp	0.16543170877331212

J_{ud}

	fluegas	xco2
mccp	-2.085982608612292e-05	1.1353141314981217

Hessian matrix positive-definite correction

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



Disturbances magnitude

	Value
fluegas	35.8595
xco2	0.05

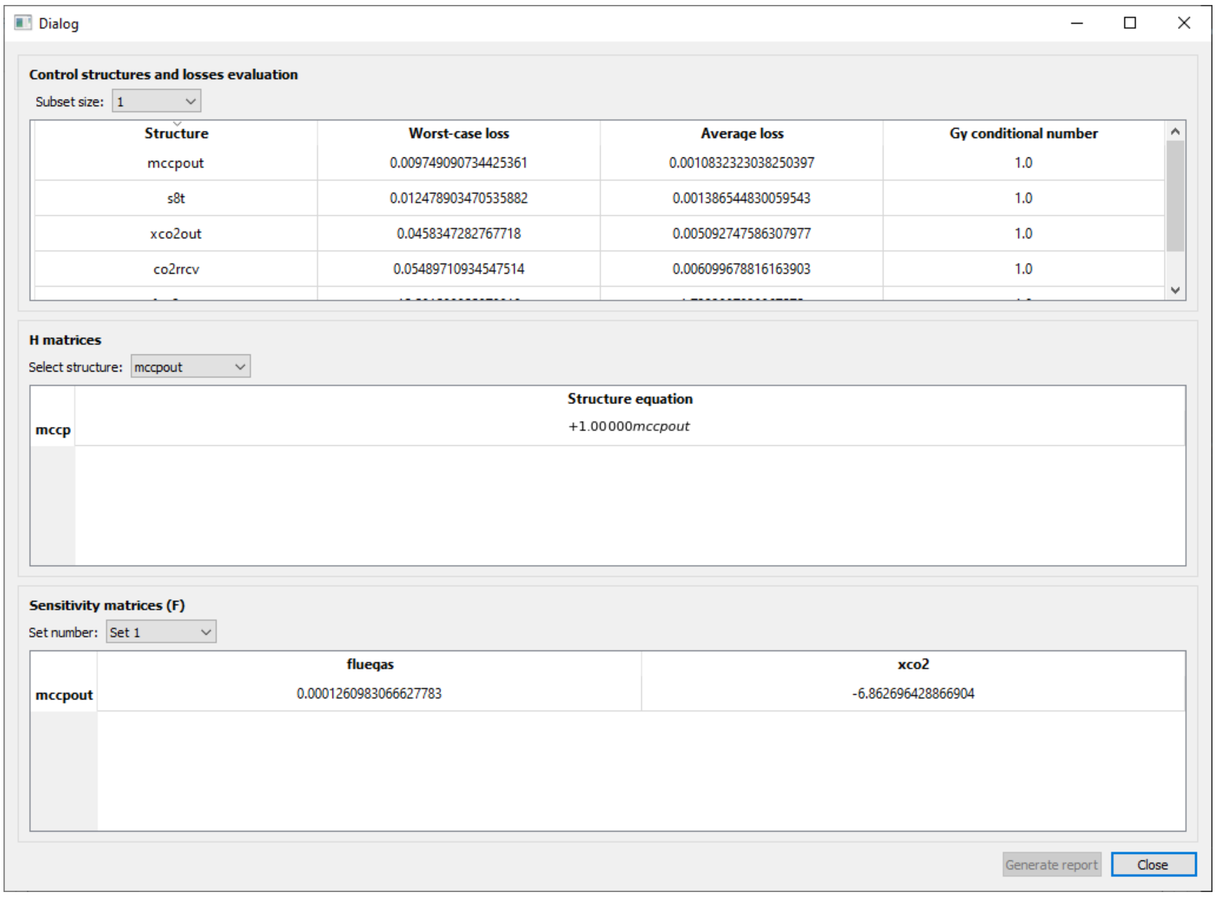
Measurements error

	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

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



Dialog

Control structures and losses evaluation

Subset size: 1

Structure	Worst-case loss	Average loss	Gy conditional number
mccpout	0.009749090734425361	0.0010832323038250397	1.0
s8t	0.012478903470535882	0.001386544830059543	1.0
xco2out	0.0458347282767718	0.005092747586307977	1.0
co2rrcv	0.05489710934547514	0.006099678816163903	1.0

H matrices

Select structure: mccpout

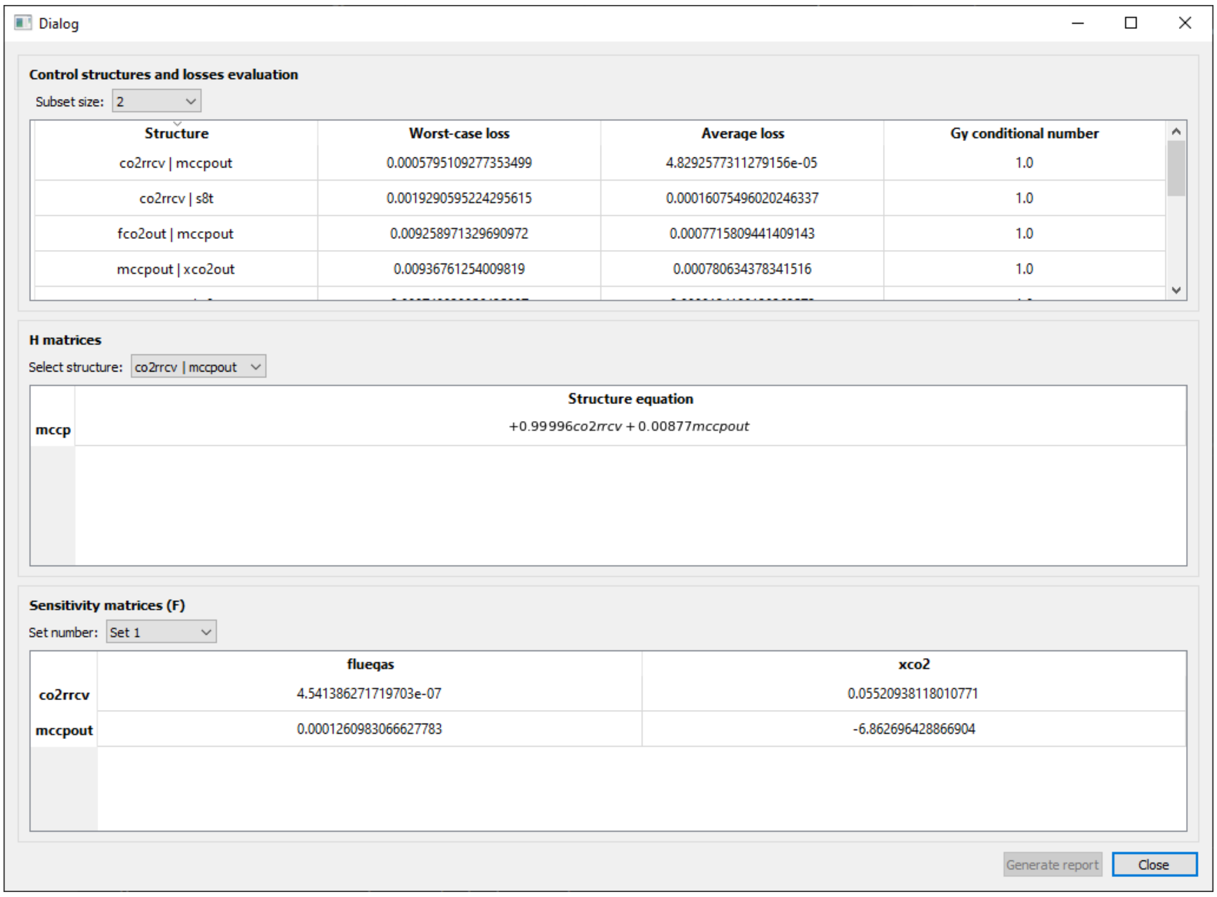
	Structure equation
mccp	$+1.00000 mccpout$

Sensitivity matrices (F)

Set number: Set 1

	fluegas	xco2
mccpout	0.0001260983066627783	-6.862696428866904

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



Control structures and losses evaluation

Subset size: 2

Structure	Worst-case loss	Average loss	Gy conditional number
co2rrcv | mccpout	0.0005795109277353499	4.8292577311279156e-05	1.0
co2rrcv | s8t	0.0019290595224295615	0.00016075496020246337	1.0
fco2out | mccpout	0.009258971329690972	0.0007715809441409143	1.0
mccpout | xco2out	0.00936761254009819	0.000780634378341516	1.0

H matrices

Select structure: co2rrcv | mccpout

	Structure equation
mccp	$+0.99996 co2rrcv + 0.00877 mccpout$

Sensitivity matrices (F)

Set number: Set 1

	fluegas	xco2
co2rrcv	4.541386271719703e-07	0.05520938118010771
mccpout	0.0001260983066627783	-6.862696428866904

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+.

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.

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.

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>.

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. 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.

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.

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.

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.

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.

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/ie049736l.

HORI, Eduardo Shigueo; 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. eprint: https://doi.org/10.1021/ie0711978. Available from: <https://doi.org/10.1021/ie0711978>.

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.

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, Esmaeil 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 optimzer 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.

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.

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>.

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.

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.

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.

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. Available from: <https://doi.org/10.1007/s10107-004-0559-y>.

Appendix

APPENDIX A – Quisque libero justo

Quisque facilisis auctor sapien. Pellentesque gravida hendrerit lectus. Mauris rutrum sodales sapien. Fusce hendrerit sem vel lorem. Integer pellentesque massa vel augue. Integer elit tortor, feugiat quis, sagittis et, ornare non, lacus. Vestibulum posuere pellentesque eros. Quisque venenatis ipsum dictum nulla. Aliquam quis quam non metus eleifend interdum. Nam eget sapien ac mauris malesuada adipiscing. Etiam eleifend neque sed quam. Nulla facilisi. Proin a ligula. Sed id dui eu nibh egestas tincidunt. Suspendisse arcu.

APPENDIX B – Nullam elementum urna vel imperdiet sodales elit ipsum pharetra ligula ac pretium ante justo a nulla curabitur tristique arcu eu metus

Nunc velit. Nullam elit sapien, eleifend eu, commodo nec, semper sit amet, elit. Nulla lectus risus, condimentum ut, laoreet eget, viverra nec, odio. Proin lobortis. Curabitur dictum arcu vel wisi. Cras id nulla venenatis tortor congue ultrices. Pellentesque eget pede. Sed eleifend sagittis elit. Nam sed tellus sit amet lectus ullamcorper tristique. Mauris enim sem, tristique eu, accumsan at, scelerisque vulputate, neque. Quisque lacus. Donec et ipsum sit amet elit nonummy aliquet. Sed viverra nisl at sem. Nam diam. Mauris ut dolor. Curabitur ornare tortor cursus velit.

Morbi tincidunt posuere arcu. Cras venenatis est vitae dolor. Vivamus scelerisque semper mi. Donec ipsum arcu, consequat scelerisque, viverra id, dictum at, metus. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut pede sem, tempus ut, porttitor bibendum, molestie eu, elit. Suspendisse potenti. Sed id lectus sit amet purus faucibus vehicula. Praesent sed sem non dui pharetra interdum. Nam viverra ultrices magna.

Aenean laoreet aliquam orci. Nunc interdum elementum urna. Quisque erat. Nullam tempor neque. Maecenas velit nibh, scelerisque a, consequat ut, viverra in, enim. Duis magna. Donec odio neque, tristique et, tincidunt eu, rhoncus ac, nunc. Mauris malesuada malesuada elit. Etiam lacus mauris, pretium vel, blandit in, ultricies id, libero. Phasellus bibendum erat ut diam. In congue imperdiet lectus.

Annex

ANNEX A – Morbi ultrices rutrum lorem.

Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper. Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit amet, scelerisque sit amet, vehicula pellentesque, sapien.

ANNEX B – Cras non urna sed feugiat cum sociis natoque penatibus et magnis dis parturient montes nascetur ridiculus mus

Sed consequat tellus et tortor. Ut tempor laoreet quam. Nullam id wisi a libero tristique semper. Nullam nisl massa, rutrum ut, egestas semper, mollis id, leo. Nulla ac massa eu risus blandit mattis. Mauris ut nunc. In hac habitasse platea dictumst. Aliquam eget tortor. Quisque dapibus pede in erat. Nunc enim. In dui nulla, commodo at, consectetuer nec, malesuada nec, elit. Aliquam ornare tellus eu urna. Sed nec metus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

ANNEX C – Fusce facilisis lacinia dui

Phasellus id magna. Duis malesuada interdum arcu. Integer metus. Morbi pulvinar pellentesque mi. Suspendisse sed est eu magna molestie egestas. Quisque mi lorem, pulvinar eget, egestas quis, luctus at, ante. Proin auctor vehicula purus. Fusce ac nisl aliquam ante hendrerit pellentesque. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi wisi. Etiam arcu mauris, facilisis sed, eleifend non, nonummy ut, pede. Cras ut lacus tempor metus mollis placerat. Vivamus eu tortor vel metus interdum malesuada.