

Article

Cloud-based CAD parametrization for design space exploration and design optimization in numerical simulations

Joel Guerrero ¹, Luca Mantelli ² and Sahrish B. Naqvi ¹

¹ DICCA, Dipartimento di Ingegneria Civile, Chimica e Ambientale. Università degli Studi di Genova. Via Montallegro 1, 16145 Genova, Italy;

² DIME, Dipartimento di Ingegneria Meccanica, Energetica, Gestionale e dei Trasporti. Thermochemical Power Group (TPG). Via Montallegro 1, 16145 Genova, Italy

* Correspondence author: joel.guerrero@unige.it

Version February 2, 2020 submitted to Journal Not Specified

Abstract: In this manuscript, an automated framework dedicated to design space exploration and design optimization studies is presented. The framework integrates a set of numerical simulation, computer-aided design, numerical optimization, and data analytics tools using scripting capabilities. The tools used are open-source and freeware, and can be deployed on any platform. The main feature of the proposed methodology is the use of a cloud-based parametrical computer-aided design application, which allows the user to change any parametric variable defined in the solid model. We demonstrate the capabilities and flexibility of the framework using computational fluid dynamics applications; however, the same workflow can be used with any numerical simulation tool (e.g., a structural solver or a spread-sheet) that is able to interact via a command line interface or using scripting languages. We conduct design space exploration and design optimization studies using quantitative and qualitative metrics, and to reduce the high computing times and computational resources intrinsic to these kinds of studies, concurrent simulations and surrogate-based optimization are used.

Keywords: CFD; numerical optimization; CAD parametrization; cloud-based; design space exploration, SSIM

1. Introduction

Consumer demand, government regulations, competitiveness, globalization, better educated end-users, environmental concerns, market differentiation, social media trends, and even influencers, they are all driving products manufacturers and industry to reduce production expenditures and final cost of goods, and at the same time improving the quality and reliability of the products with the lowest environmental impact. To reach these goals and to develop revolutionary products, the manufacturing sector is relying more on virtual prototypes, computer simulations, and design optimization.

Computational fluid dynamics (CFD), computational structural dynamics (CSD), computer-aided manufacturing (CAM), computer-aided design (CAD), multi-physics simulations, digital twins, the internet-of-things (IoT) and the cloud, are among many of the tools increasingly being used to simulate and certify products by analysis and simulation before going into production and commercialization. Even before reaching the market, modern products have undergone some kind of heuristic or methodological optimization. Though the optimization might take different forms in different fields (e.g., finance, health, construction, operations, manufacturing, transportation, construction, engineering design, sales, public services, mail, and so on), the ultimate goal is always getting the

³¹ best out of something under given circumstances, either by minimizing, maximizing, equalizing, or
³² zeroing a quantity of interest (QoI).

³³ Product optimization can be undertaken in two different ways, by using design space exploration
³⁴ (DSE) or by using design optimization (DO). Even a combination of both methodologies is possible.
³⁵ In DSE, we simply explore the design space in a methodological way, and while doing so, we extract
³⁶ knowledge. DSE is the process of discovering, expanding, evolving, and navigating the design space
³⁷ in order to extract knowledge to support better decision making [1]. It is not difficult to recognize that
³⁸ in DSE, we are not converging to an optimal value, we are only exploring the design space, but in
³⁹ doing so, we are gathering valuable information about the global behavior, and this information can
⁴⁰ be used to get a better design. Moreover, this knowledge can also be used to conduct surrogate-based
⁴¹ optimization (SBO) studies. The SBO method consists of constructing a mathematical model (also
⁴² known as a surrogate, response surface, meta-model, emulator) from a limited number of observations
⁴³ (CFD simulations, physical experiments, or any quantifiable metric) [2–5]. After building the surrogate,
⁴⁴ it can be explored and exploited. Conducting the optimization at the surrogate level is orders of
⁴⁵ magnitude faster than working at the high-fidelity level [2].

⁴⁶ Design optimization strategies, on the other hand, consist on formulating an optimization problem
⁴⁷ and converging to the optimal design. Here, it is assumed that the problem can be formulated before
⁴⁸ the search and convergence begin. A typical optimization problem can be formulated as follows,

$$\text{Find } \mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} \quad (1)$$

⁴⁹ which minimizes, maximizes, equalizes, or zeroed,

$$f_j(\mathbf{X}), \quad j = 1, 2, \dots, q \quad (2)$$

⁵⁰ subject to design constraints (linear and non-linear),

$$\begin{aligned} g_j(\mathbf{X}) &\leq 0, \quad j = 1, 2, \dots, m \\ l_j(\mathbf{X}) &= 0, \quad j = 1, 2, \dots, p \end{aligned} \quad (3)$$

⁵¹ and variables bounds,

$$x_i^{lb} \leq x_i \leq x_i^{ub} \quad j = 1, 2, \dots, n \quad (4)$$

⁵² where \mathbf{X} is a n -dimensional vector called the design vector, $f_j(\mathbf{X})$ is the objective function or QoI, $g_j(\mathbf{X})$
⁵³ are the inequality constraints, $l_j(\mathbf{X})$ are the equality constraints, and x_i^{lb} and x_i^{ub} are the variables lower
⁵⁴ and upper bounds, respectively. To find the optimal value we can use gradient-based methods or
⁵⁵ derivative-free methods [5–10]. Also, the problem formulation can be single-objective (one QoI to be
⁵⁶ optimized) or multi-objective (more than one QoI to be optimized simultaneously). Things can get
⁵⁷ even more complicated, as in some cases we might need to deal with design optimization problems
⁵⁸ incorporating many disciplines (e.g., aerodynamics, propulsion, structures, and performance). In this
⁵⁹ case, we say we are dealing with a multi-disciplinary design optimization problem (MDO) [11–16].
⁶⁰ MDO allows designers and engineers to incorporate all relevant disciplines simultaneously. The
⁶¹ optimum of the simultaneous problem is superior to the design found by optimizing each discipline

62 sequentially since it can exploit the interactions between the disciplines. However, including all
63 disciplines simultaneously significantly increases the complexity of the problem [7].

64 The field we are concerned with in this manuscript is that of engineering design; nevertheless,
65 this by no means limits the range of applicability of the current work; it simply reflects the authors'
66 interests and fields of expertise.

67 In engineering design, we are often interested in optimizing the geometry. To do so, two
68 approaches are available, direct modeling and parametric modeling. In direct modeling, we modify
69 the geometry by pushing and pulling points, lines, and surfaces (like working with clay). This gives
70 designers and engineers a lot of flexibility when it comes to shape the geometry; however, in the
71 process of doing so, we give up geometry parametrization in favor of creating organic shapes that
72 might be difficult to manufacture. In parametric modeling, the user defines relationships, constraints,
73 parametric variables, and configurations when creating the solid model. Then, by changing these
74 variables, the user can easily create endless variations on the original geometry with complete control
75 and millimetric precision.

76 However, when conducting fully automatic DSE or DO studies, introducing the CAD tools is not
77 very straightforward. Most of the times the CAD applications are not compatible with the operating
78 system (OS) where the numerical simulations are being performed (usually Unix-like OS), or simply, it
79 is not possible to connect the optimization loop with the CAD tool due to the fact that the user can
80 only interact with it using a graphical user interface (GUI), which cannot be used in an automatic
81 optimization loop driven by a command line interface (CLI).

82 To overcome this problem, many commercial simulation frameworks are adding a monolithic
83 design environment to integrate all the applications needed to conduct design space exploration and
84 design optimization studies, namely, CAD, multi-physics solver, optimizer, and post-processing. While
85 commercial frameworks have proven to be reliable, they come with a price tag that often is unreachable
86 by small and medium-sized enterprises (SMEs), hobbyists, researchers or personal users. Hereafter,
87 we propose the integration of open-source and freeware tools to conduct DSE and DO studies.

88 To perform the numerical simulations, we use the multi-physics solver OpenFOAM (version 7.0)
89 [17,18] or the programming language Python. The optimization algorithms and the code coupling
90 interface is provided via the Dakota library [19,20] (version 6.10). All the real-time data analytics,
91 quantitative and qualitative post-processing, and data analytics are performed using Python, VTK
92 [21], and bash scripting. Finally, to create and modify the geometry we use Onshape [22], which
93 is a cloud-based parametric CAD and product development application. Onshape's application
94 programming interface (API) is open-source; therefore, it can be deployed in any platform with an
95 internet connection. The API is implemented in Python, and the calls to Onshape's server are done
96 using RESTful requests. Onshape offers two subscription plans, a pay-up plan and a free one. Both
97 subscriptions plans have the same professional capabilities, the only difference is the level of product
98 support offered and the access to enterprise options.

99 The purpose of this manuscript is two-fold. First, we want to utilize the cloud to support CAD
100 parametrization in DSE or DO design loops, which undoubtedly will give users enormous flexibility
101 as the CAD application does not need to be installed locally, and there is no need for a monolithic
102 CAD/Simulation software integration. Secondly, we want to deploy fully automatic, fault-tolerant,
103 and scalable engineering design loops using in-house computational resources, the cloud, or HPC
104 centers; and everything based on open-source and freeware tools. We hope that this contribution
105 will offer guidelines to designers and engineers working with design optimization and design space
106 exploration, will help them at implementing their own optimization loops, and to some extend, it
107 will help to address some of the findings and recommendations listed in the NASA contractor report
108 "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences" [23], where it is stated the
109 following: "*Included in this desired set of capabilities is a vision for how CFD in 2030 will be used: a vision*
110 *of the interaction between the engineer/scientist, the CFD software itself, its framework and all the ancillary*
111 *software dependencies (databases, modules, visualization, etc.), and the associated HPC environment. A single*

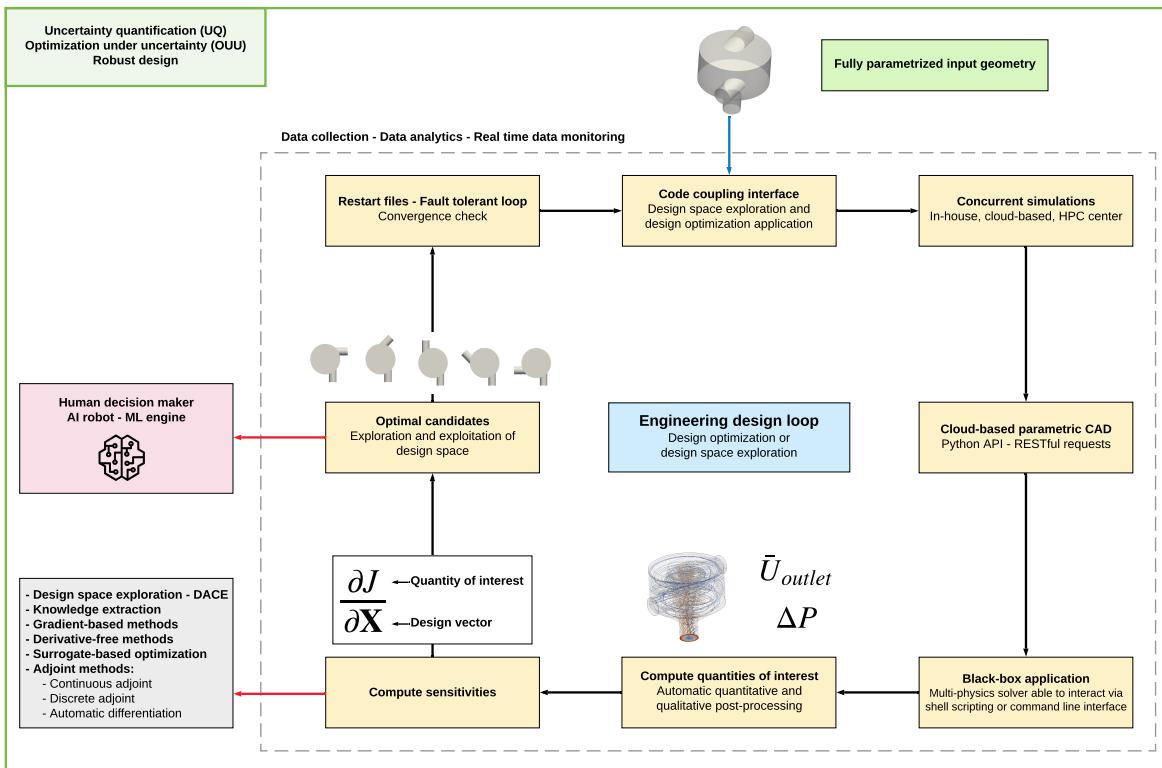


Figure 1. Graphical summary of the engineering design loop.

112 engineer/scientist must be able to conceive, create, analyze, and interpret a large ensemble of related simulations
 113 in a time-critical period (e.g., 24 hours), without individually managing each simulation, to a pre-specified level
 114 of accuracy".

115 The rest of the manuscript is organized as follows. Section 2 gives an overview of the methodology
 116 used. In section 3 we describe the numerical experiments carried out to demonstrate the usability and
 117 flexibility of the framework. Finally, in section 4 we present the conclusions and future perspectives.

118 2. Description of the workflow – Methodology

119 In figure 1, we illustrate a graphical summary of the methodology used in this work. The
 120 engineering design loop starts with a fully parametrized geometry, then new candidates are generated
 121 by changing the parametrical variables. It is important to stress that our starting point is the
 122 parametrical variables and not the solid model; that is, we are allowed to start from any possible
 123 geometry that can be generated using the parametrical variables. Hereafter, we use Onshape [22] as
 124 solid modeler, which is a cloud-based CAD application. The fact that Onshape is cloud-based gives us
 125 the flexibility to deploy the framework in any platform without the need to install the application. The
 126 only requirement is to have a working internet connection.

127 The whole workflow is controlled by the library Dakota [19,20], which serves as the numerical
 128 optimizer and code coupling interface tool. The Dakota library provides a flexible and extensible
 129 interface between simulation codes and iterative analysis methods. The library is software agnostic, in
 130 the sense that it can interface any application that is able to parse input/output files via a CLI. The
 131 library also has extensive design optimization and design space exploration capabilities. It comes
 132 with many gradient-based methods and derivative-free methods for design optimization. It also
 133 contains many design and analysis of computer experiments (DACE) methods to conduct design space
 134 exploration studies. And to obtain faster turn-around times, Dakota supports concurrent function
 135 evaluations.

The engineering design loop illustrated in figure 1 is orchestrated by using Dakota's configuration input file. In this input file, all the steps to follow in the engineering design loop are defined. As previously stated, the only requirement is that the applications involved in the loop are able to interact via the CLI. In references [3,24–33], few examples using Dakota to control complex engineering design loops are discussed. However, none of them addressed the use of a fully parametric cloud-based CAD tool to generate the solid geometry or the use of the cloud to deploy the loop.

After defining Dakota's configuration file, the engineering design loop can be launched sequentially or concurrently using local resources, on the cloud, or remotely in an HPC center. All the tools involved in the loop are black-box applications that are connected using Dakota. An essential step of every optimization loop is that a QoI must be provided to compute the sensitivities; this is also controlled using Dakota's configuration file. This step is critical and is the user's responsibility to define all the quantities of interests to monitor. After computing the QoI, Dakota will compute the sensitivities using the method selected by the user. With Dakota, the user is not obliged to use the optimization and space exploration methods implemented on it; one can easily interface Dakota with a third-party optimization library.

At this point, we can rely on a human decision-maker or a machine learning engine to pick up the best design or set of optimal solutions. During the whole process, data is collected and monitored in real-time. Dakota also offers restart capabilities, so in the event of an unexpected failure of the system (hardware or software), the user can restart from a previously saved state.

In this work, we use the design loop illustrated in figure 1 for DO and DSE studies. In DO, the user starts from an initial design or guess, and the optimization algorithm will make it slightly better. That is, in DO we are making suboptimal guesses incrementally better. This by no means is negative, and the chances are that the results are a substantial improvement over the initial guess. In essence, DO is an iterative-converging process that requires a starting point (or a set of points) and a set of constraints. On the other hand, in DSE we do not need to define an initial guess or a set of constraints (except for the bounds of the design space). We generate new solutions sequentially or concurrently, that might be better or worse than a baseline, but in the process of doing so, we are exploring and exploiting the design space. DSE gives more information to engineers than DO, and this information can be used for decision making, knowledge extraction, and anomalies detection. All the information gathered during the design loop can also be used to construct reduced order models, surrogate models, or to interrogate the data using exploratory data analysis and machine learning techniques.

3. Numerical experiments

3.1. Cylinder optimization problem - Minimum surface and fixed volume

This problem is also known as the soda can optimization problem. We aim at finding the optimal dimensions of a right cylinder that minimize the total surface area of the cylinder, which holds a given volume. This problem can be formulated as follows,

$$\text{minimize } S_{tot} \quad (5)$$

subject to,

$$\begin{aligned} V &= 355 \text{ cm}^3 \\ 0 < r, h &< \infty \end{aligned} \quad (6)$$

where,

$$\begin{aligned} S_{tot} &= 2\pi r^2 + 2\pi rh \\ V &= \pi r^2 h \end{aligned} \tag{7}$$

¹⁷⁴ in equations 5-7, S_{tot} is the cylinder's total surface, V its volume, r its radius, and h its height. The
¹⁷⁵ solution to this problem is the following,

$$\begin{aligned} r &= 3.837 \text{ cm} \\ h &= 7.675 \text{ cm} \\ S_{min} &= 277.54 \text{ cm}^2 \end{aligned} \tag{8}$$

¹⁷⁶ This is a classic problem that is frequently posed to freshman calculus students. Therefore, we
¹⁷⁷ will not go into details on how to find the analytical solution (equation 8). Instead, we will use this
¹⁷⁸ case to illustrate how the cloud-based design loop works.

¹⁷⁹ In figure 2, we illustrate the general workflow. In steps 1-2, we define all the configuration
¹⁸⁰ variables and measurements (e.g., area, volume, length, and so on). In these steps, we also check that
¹⁸¹ we are obtaining the desired output by changing manually the parametrical variables. In figure 3, we
¹⁸² show the screen-shot of how this case was setup in Onshape (the document is available at the following
¹⁸³ link ¹); in the figure, it can be observed that all the configurations, bounds, and measurements have
¹⁸⁴ been defined. All these variables can be accessed or modified using Onshape's Python API ². In
¹⁸⁵ step 3 we proceed to test the connection with Onshape's server, this is illustrated in figure 4. In the
¹⁸⁶ figure, we use the API client to encode the changes to the model configurations and evaluation of the
¹⁸⁷ measurements. Then, using OAuth authentication, a RESTful request is sent to Onshape's server, which
¹⁸⁸ sends a response back to the client. The response can be the new geometry or the evaluation of the
¹⁸⁹ volume of the new solid model. After testing the configurations and communication with Onshape's
¹⁹⁰ server, we proceed to define the problem in Dakota's configuration file and to create any additional
¹⁹¹ scripts needed to parse input/output files (step 4). This step includes choosing the optimization or
¹⁹² space exploration method and defining the bounds, constraints, and objective functions. At this point,
¹⁹³ we can proceed to deploy the case sequentially or concurrently using local resources, the cloud, or
¹⁹⁴ HPC center resources (step 5). Finally, in step 6, we can visualize the optimal solid model. Additionally,
¹⁹⁵ we can use exploratory data analysis to study the collected data. During the whole process, restart
¹⁹⁶ files are generated and data is monitored in real-time.

¹⁹⁷ In listing 1, we show an excerpt of the Python code used to change the configuration variables. In
¹⁹⁸ the listing, the keywords **height_to_update**, **dia1_to_update**, and **dia2_to_update** are the parametric
¹⁹⁹ variables, and each one was defined in the Onshape document. Their values are substituted
²⁰⁰ automatically by Dakota, and their bounds are defined in Dakota's configuration file. The function
²⁰¹ **part_studio_stl_conf** is responsible for exporting the geometry using the current values of the
²⁰² configuration variables (in this case the geometry is exported in STL format but any supported
²⁰³ CAD exchange format can be used). The exported geometry is then used with the black box solver. The
²⁰⁴ **did**, **wid**, and **eid** keywords in listing 1 are referred to the document id, workspace id, and element id
²⁰⁵ of the Onshape document (refer to figure 3). In listing 2, we show an excerpt of the Python code used
²⁰⁶ to evaluate the measurements (the structure is similar to that of listing 1). In the listing, the line of
²⁰⁷ code "function(context, queries) return getVariable(context, 'volume');" evaluates the measurement,
²⁰⁸ as defined in the Onshape document. In this case, we are evaluating the volume of the solid model. As
²⁰⁹ for the configuration variables, all the measurements need to be defined in the Onshape document.

¹ <https://cad.onshape.com/documents/448249f25f37397d1823feb6/w/33bca1cf858efd73dc35ab4f/e/2ec99af57f87dd94045affd>
² <https://github.com/onshape-public/apikey/tree/master/python>

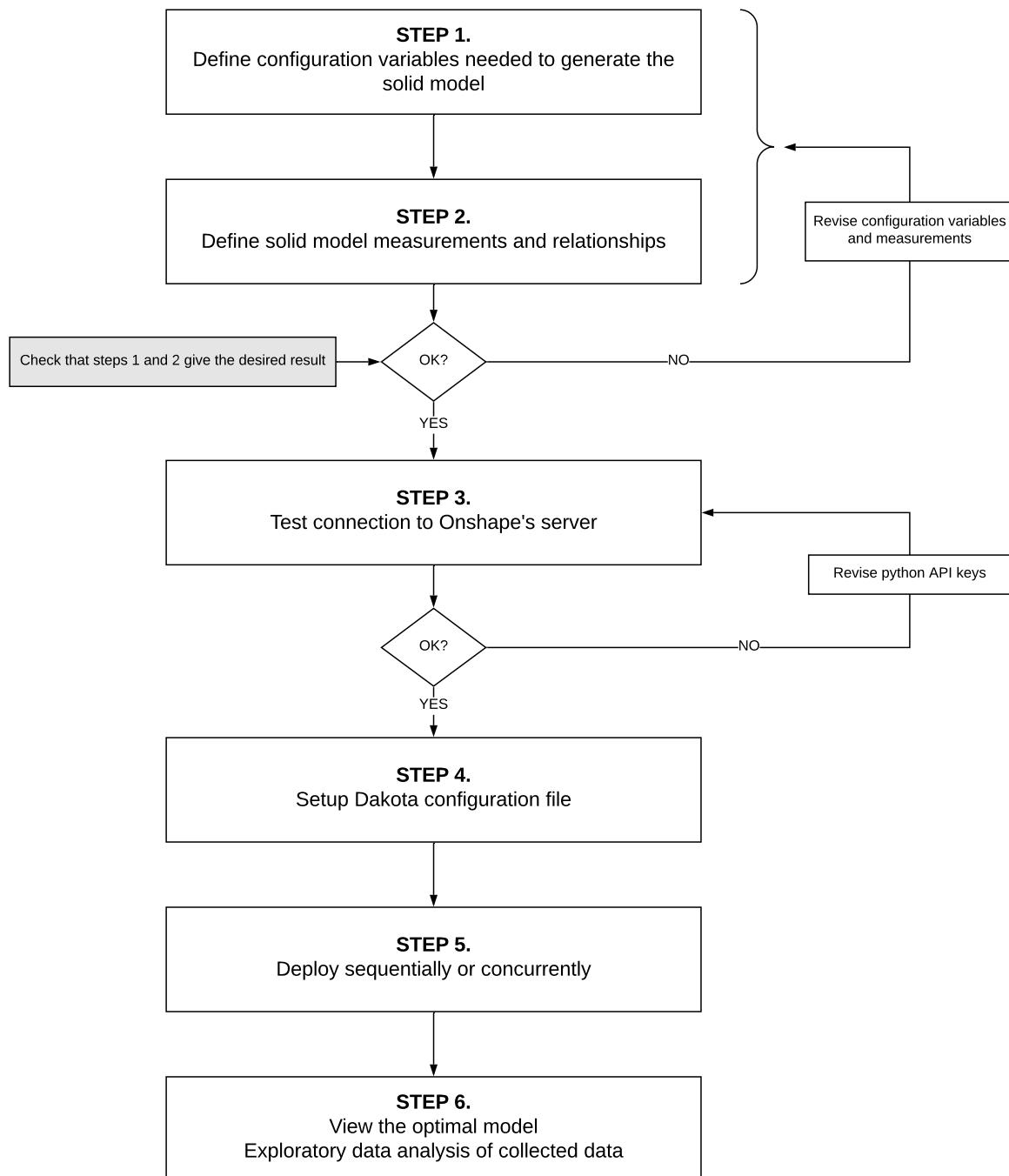


Figure 2. Workflow of the problem setup using the proposed cloud-based framework.

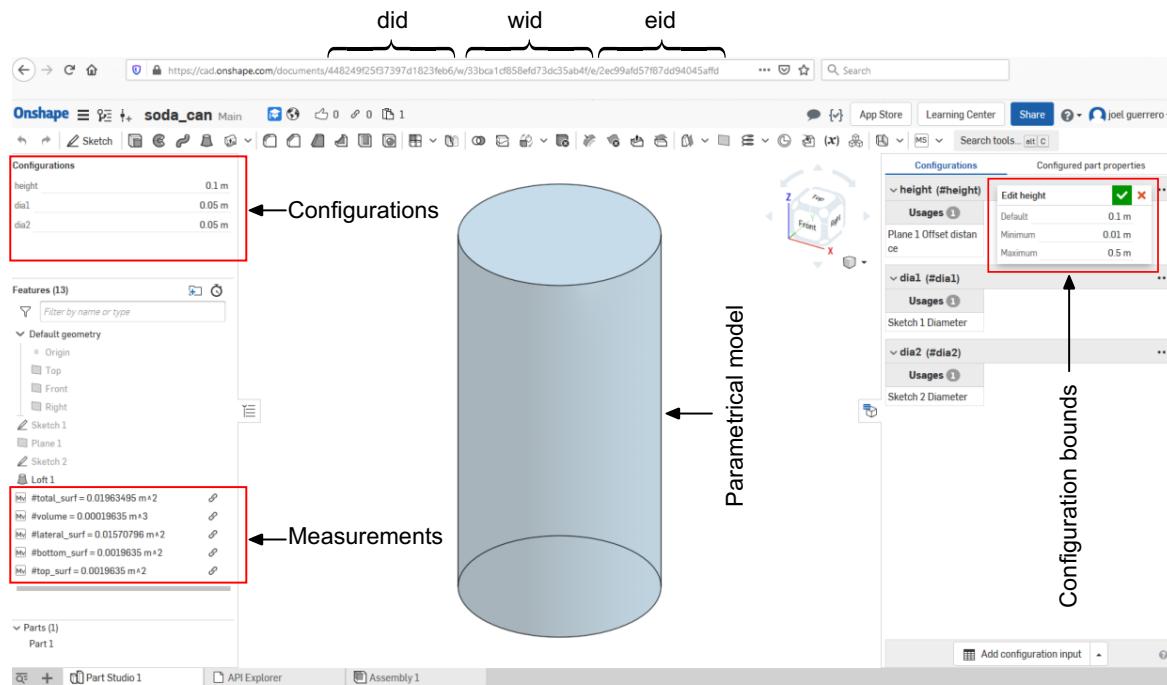


Figure 3. Definition of configuration variables and measurements in the Onshape document.

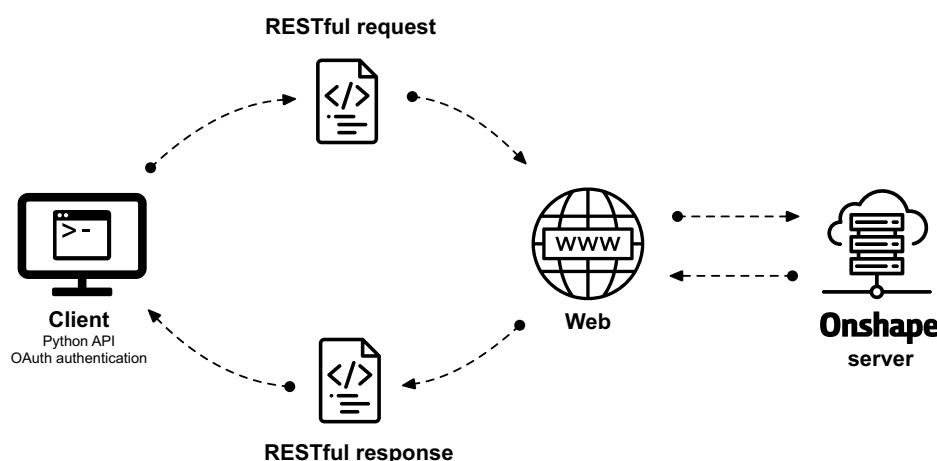


Figure 4. Onshape's cloud-based client-server communication using RESTful API. The client communicates with Onshape's server using Python API keys and OAuth authentication.

```

configuration = {
    'units': 'meter',
    'scale': 1.0,
    'configuration' :
        'height=[[height_to_update]]+m;'
        'dia1=[[dia1_to_update]]+m;'
        'dia2=[[dia2_to_update]]+m'
}
stl = c.part_studio_stl_conf(did, wid, eid, configuration)

```

Listing 1: Excerpt of the Python code used to setup the parametric configuration variables.

```

body_feature = {
    "script" :
        "function(context,_queries){return_getVariable(context,_'volume');}"
}

configuration = {
    'units': 'meter',
    'scale': 1.0,
    'configuration' :
        'height=[[height_to_update]]+m;'
        'dia1=[[dia1_to_update]]+m;'
        'dia2=[[dia1_to_update]]+m'
}
out = c.featurescript_conf(did, wid, eid, body_feature, configuration)

```

Listing 2: Excerpt of the Python code used to evaluate the measurements.

210 In the listing, the function **featurescript_conf** takes the configuration values and the measurement
 211 function definition and gives as output the evaluation of the measurement for the given configuration.
 212 For the interested reader, the working case with all the scripts can be downloaded at this link³). These
 213 scripts can be used as a starting point for more complex cases. It is worth mentioning that the Python
 214 API works with Python 2 (2.7.9+).

215 Let us discuss the outcome of a DO study using a gradient-based method (method of feasible
 216 directions or MFD [34,35]). As we are optimizing a right cylinder, we set the diameters of the top
 217 and bottom surfaces to the same value, we also started to iterate from two different initial conditions.
 218 In table 1, we show the outcome of this study. As can be observed, in both situations we arrived
 219 at the optimal value, and any deviation from the analytical solution is due to numerical precision
 220 and convergence tolerances. It is also interesting to note that depending on the starting conditions,
 221 different convergence rates can be achieved. The closer we are to the optimal solution, the faster the
 222 convergence will be. This put in evidence that the formulation of an optimization problem using
 223 gradient-based methods requires certain knowledge on the behavior of the design space; otherwise,
 224 the convergence rate to the optimal value will be slow.

225 During the DO study, we also used a derivative-free method (mesh adaptive direct search
 226 algorithm or MADS [36]), which also converged to the optimal solution but with a slow convergence
 227 rate, as shown in table 2. As a side note, even if the derivative-free method exhibited a slow convergence

³ https://github.com/joelguerrero/cloud-based-cad-paper/tree/master/soda_can/

228 rate, it was faster than the gradient-based method with a poor guess of the starting point (MFD-2 in
 229 table 1). In general, derivative-free methods do not require the definition of the starting point, and
 230 they are insensitive to numerical noise.

| | MFD-1 | MFD-2 | Analytical solution |
|--------------------------------------------------|---------|---------|---------------------|
| Starting point - Height (height_to_update) - cm | 4 | 2 | - |
| Starting point - Diameter (dia1_to_update) - cm | 8 | 12 | - |
| Optimal value - Height (height_to_update) - cm | 7.617 | 7.607 | 7.675 |
| Optimal value - Diameter (dia1_to_update) - cm | 7.692 | 7.697 | 7.674 |
| QoI (S_{tot}) - cm ² | 277.026 | 277.027 | 277.54 |
| Non-linear constraint (Volume) - cm ³ | 354.001 | 354.000 | 354.98 |
| Function evaluations | 88 | 405 | - |

Table 1. Outcome of the optimization study using a gradient-based method (MFD [34,35]).

| | MFD | MADS | Analytical solution |
|--------------------------------------------------|---------|---------|---------------------|
| Optimal value - Height (height_to_update) - cm | 7.617 | 7.699 | 7.675 |
| Optimal value - Diameter (dia1_to_update) - cm | 7.692 | 7.655 | 7.674 |
| QoI (S_{tot}) - cm ² | 277.026 | 277.236 | 277.54 |
| Non-linear constraint (Volume) - cm ³ | 354.001 | 354.406 | 354.98 |
| Function evaluations | 88 | 256 | - |

Table 2. Outcome comparison of the gradient-based method (MFD [34,35]) and the derivative-free method (MADS [36]). In the table, MFD refers to the gradient-based method (same as MFD-1 in table 1), and MADS refers to the derivative-free method.

231 In table 3, we compare the results of the same DO study but this time using two and three design
 232 variables. Again, we obtain results close to the analytical one, and surprisingly, the convergence rate of
 233 both cases was similar. The main reason for the similarity of the convergence rate is that the starting
 234 points of the design variables are close to the optimal value. This evidence the importance of choosing
 235 good starting points in order to get a good convergence rate; gradient-based methods can be very
 236 sensitive to this choice. Regarding the case setup, the main difference is that we need to add additional
 237 scripts to compute the area of the top and bottom surfaces of the cylinder, independently.

238 Let us run the same case using a design space exploration method. We remind the readers that
 239 when using DSE, we are not explicitly converging to an optimal solution; we are just exploring the
 240 design space. Then, the outcome of this study can be used for knowledge extraction, anomalies
 241 detection, or to construct a surrogate model. To conduct this DSE study, we used a full-factorial
 242 experiment with 21 experiments equally spaced for each design variable (for a total of 441 observations).
 243 In figure 5, we show one of the many plots that can be used to visualize the data coming from DSE
 244 studies [3,37]. This plot is called scatter plot matrix, and in one single illustration, it shows the
 245 correlation information, the data distribution (using histograms and scatter plots), and regression
 246 models of the responses of the QoI.

247 By conducting a quick inspection of the scatter plot matrix displayed in figure 5, we can evidence
 248 that the data is distributed uniformly in the design space (meaning that the sampling plan is unbiased),
 249 and this is evidenced in the diagonal of the plot (the plots corresponding to the design variables).
 250 By looking at the scatter plot of the experiments (lower triangular part of the matrix), we see the
 251 distribution of the data in the design space. If, at this point, we detect regions in the design space

| | MFD-2DV | MFD-3DV | Analytical solution |
|---------------------------------------------------|---------|---------|---------------------|
| Starting point - Height (height_to_update) - cm | 4 | 4 | - |
| Starting point - Diameter 1 (dia1_to_update) - cm | 8 | 8 | - |
| Starting point - Diameter 2 (dia2_to_update) - cm | - | 5 | - |
| Optimal value - Height (height_to_update) - cm | 7.617 | 7.648 | 7.675 |
| Optimal value - Diameter 1 (dia1_to_update) - cm | 7.692 | 7.686 | 7.674 |
| Optimal value - Diameter 2 (dia2_to_update) - cm | - | 7.666 | - |
| QoI (S_{tot}) - cm ² | 277.026 | 277.026 | 277.54 |
| Non-linear constraint (Volume) - cm ³ | 354.001 | 354.004 | 354.98 |
| Function evaluations | 88 | 114 | - |

Table 3. Outcome of the optimization study using a gradient-based method (MFD [34,35]). In the table, MFD-2DV refers to the case with two design variables. MFD-3DV refers to the case with three design variables. The case MFD-2DV uses the same diameter for the top and bottom surfaces.

that remain unexplored, we can add new training points to cover those areas. In the case of outliers (anomalies), we can remove them from the dataset with no significant inconvenience. However, we should be aware that outliers are telling us something, so it is a good idea to investigate the cause and effect of the outliers. In the upper triangular part of the plot, the correlation information is shown (Spearman correlation in this case). This information tells us how correlated the data is. For example, and by looking at the last row of the plot that shows the response of the QoI, if we note here a strong correlation between two variables, it is clear that these variables cannot be excluded from the study. As can be seen, this simple plot can be used to gather a deep understanding of the problem.

The data gathered from the DSE study can also be used to construct a meta-model, and then conduct the optimization at the surrogate level. In figure 6, we illustrate the response surface, which was constructed using Kriging interpolation (universal Kriging). The implementation details of the method can be found in references [2,4,20,38–42]. To conduct the optimization at the surrogate level, we used the MFD gradient-based method (method of feasible directions [34,35]). However, any optimization method (gradient-based or derivative-free) can be used as working at the surrogate level is inexpensive; we do not need to perform high fidelity function evaluations.

In figures 5–6, we plot a two-variable design space. In general, a design space will be n -dimensional, where n is the number of design variables of which the objective is a function. We deliberately used a two-variable design space to help visualize the response surface, the design space, and the various concepts related to DO and DSE. For completeness, we extended this problem to three design variables, and we obtained similar results by using the same methodology. We want to point out that all the results discussed in this section were obtained using Python scripting as black-box solver, and the volume and surfaces were computed using Onshape’s API.

We would like to highlight that the optimized can dimensions presented in this section significantly differ from actual soda cans. We should ask ourselves, is the shape of this soda can truly optimal? From a mathematical point of view, yes. But from a point of view of going through the whole process of manufacturing the can, is not. This simple example shows that optimization is very subjective. Sometimes manufacturers are trying to optimize something a little bit more abstract, like, how the can is manufactured?, packing factor?, opening mechanism?, costumer satisfaction?, aluminum cost? And these abstract questions are better answered using design space exploration and by visualizing and interacting with the results in real-time, as is possible to do by using the proposed cloud-based engineering design framework.

To close the discussion of this introductory case, we would like to reiterate that the optimization loop implemented is fault-tolerant, so in the event of hardware or software failure, the optimization

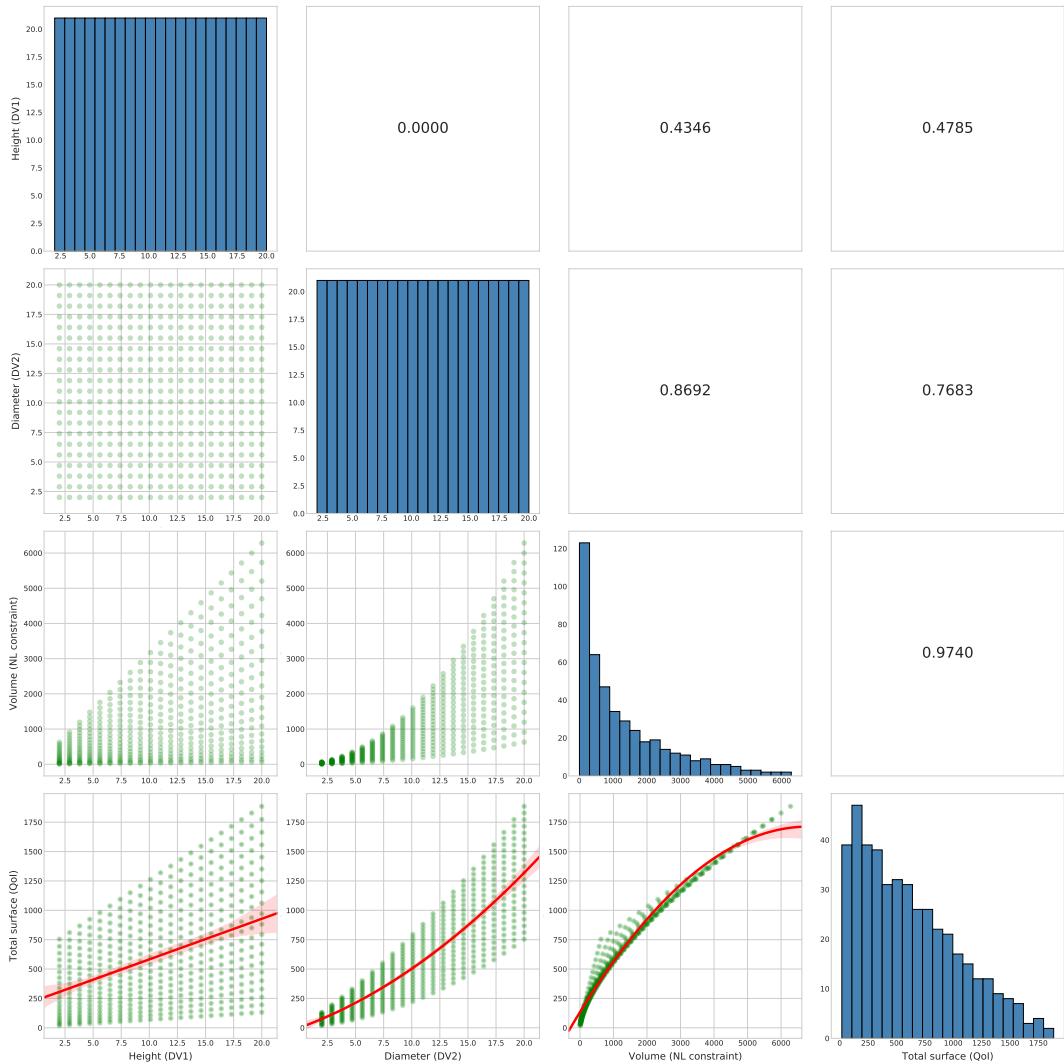


Figure 5. Scatter plot matrix of the cylinder optimization case using two design variables. In the upper triangular part of the plot, the Spearman correlation is shown. In the diagonal of the matrix, the histograms showing the data distribution are displayed. In the lower triangular part of the matrix, the data distribution is shown using scatter plots. In the last row of the matrix plot, the response of the QoI in function of the design variables and the non-linear (NL) constraint is illustrated, together with a quadratic regression model.

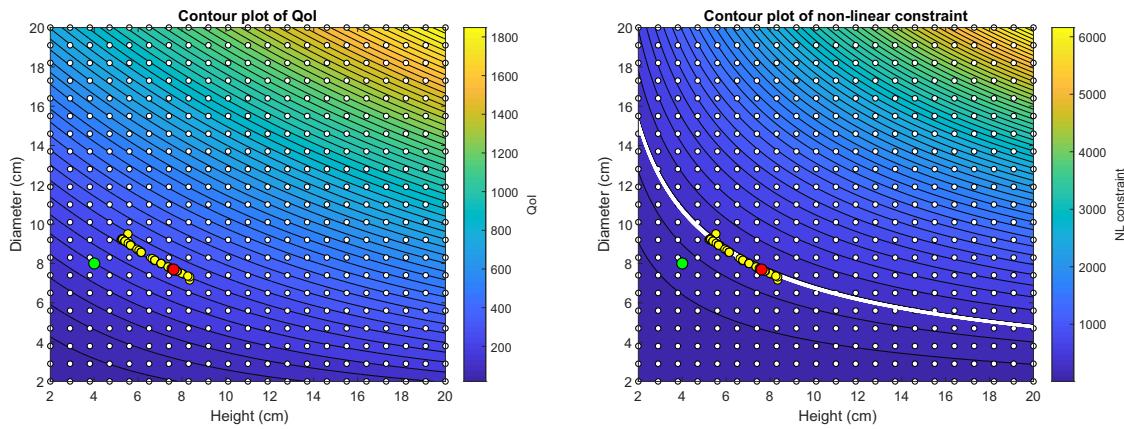


Figure 6. **Left image:** contour plot of the QoI (total surface). **Right image:** contour plot of the non-linear constraint (volume); in the image, the white line represents the range where the volume is $354 \text{ cm}^3 < \text{Volume} < 356 \text{ cm}^3$. In both images, the green circle represents the starting point, the red circle represents optimal value, the yellow circles represent the path followed by the optimization algorithm (note that the gradient evaluations are not plotted), and the white circles represent the sampling points.

task can be restarted from the last saved state. During the design loop, all the data is made available immediately to the user, including the geometry, even when running multiple simulations at the same time. Moreover, the data is monitored in real-time; therefore, anomalies and trends can be detected in real-time, and corrections/decisions can be taken. Finally, when it comes to engineering design studies, DO will converge to the optimal value, but formulating the problem requires some knowledge about the design space. Also, DO does not give valuable information about the global behavior of the QoI. Design space exploration, on the other hand, provides a lot of information about the design space without converging to the optimal value. Still, these studies might be expensive to conduct due to the high number of function evaluations often required to construct a reliable estimator. An added benefit of DSE is that the outcome can be used to conduct SBO studies, where the cost of evaluating the QoI and derivatives is zero as we are working at the surrogate level. Ultimately, the choice of the method to use is to the user, and likely based on the computational resources available and in the difficulty to formulate the optimization problem.

3.2. Static mixer optimization case

In this case, we introduce the use of a qualitative metric to conduct the engineering design study. We also compare the outcome of a DO study and a DSE study. The geometry used in this case is shown in figure 7, and it corresponds to a static mixer with two inlets and one output. The goal, in this case, is to obtain a given velocity distribution at the outlet by changing the angle of the inlet pipe 1 (refer to figure 7). The velocity distribution field at the outlet was designed in such a way that the velocity normal to the outlet surface has a paraboloid distribution. Then, by using the SSIM index method (refer to appendix A for an explanation), we compared the target image and the image of the current configuration (refer to figure 8). The closer the SSIM index is to one, the more similar the images are; therefore, we aim at maximizing the QoI.

The simulations were conducted using OpenFOAM (version 7.0) [17,18]. To find the approximate solution of the governing equations, the SIMPLE pressure-velocity coupling method was used, together with the $k - \epsilon$ turbulence model with wall functions, and a second-order accurate and stable discretization method for the convective, diffusive, and gradient terms. The Onshape document with

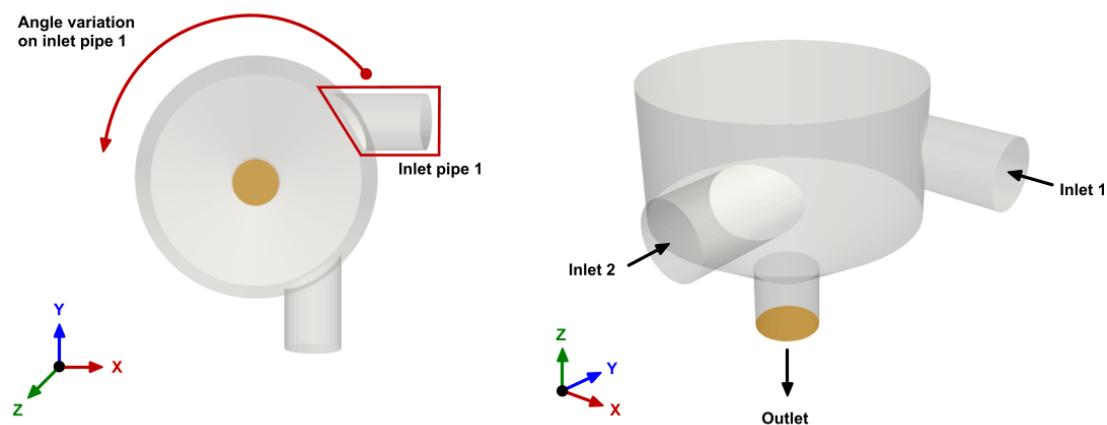


Figure 7. Static mixer geometry.

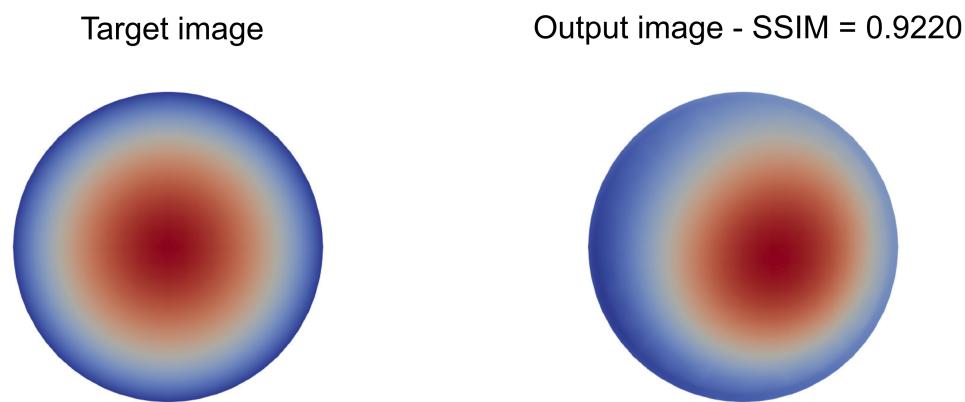


Figure 8. Velocity distribution normal to the outlet surface. **Left image:** reference velocity distribution or target image. **Right image:** image of the velocity distribution for a non-optimal case. To determine if the images are similar, we used the SSIM index method. The closer the SSIM index of the output image is to one, the more similar the images are.

312 all the dimensions is available at the following link⁴. The case setup with the boundary conditions
 313 and initial conditions can be found at this link⁵.

314 In figure 9, we plot the outcome of the DO study using a gradient-based method (method of
 315 feasible directions or MFD [34,35]), and the DSE study using a uniform sampling for the inlet pipe
 316 angle (from 0 to 180 degrees). For the DO case, we used as starting point 0 degrees, and the case
 317 converged to the optimal value (pipe angle equal to 111.0549 degrees and SSIM index equal to 0.9660)
 318 in 31 function evaluations. In the DSE case, we explored the design space from 0 to 180 degrees, in
 319 steps of 5 degrees, so roughly speaking, we used the same number of function evaluations as for the
 320 DO case. From figure 9, we can evidence that the DSE study, while not formerly converging to the
 321 optimal solution, gives more information about the design space than the DO method. From the DSE
 322 results, we can see that there is a plateau of the SSIM value for pipe angle values between 90 and 135
 323 degrees. This information is not available when conducting DO studies, as the goal of these methods is

⁴ <https://cad.onshape.com/documents/8f1312fafb3aac0f7bd3ed38/w/72a43b7cd8ca686e908ef122/e/33c606cd59a53e2b8532a94a>

⁵ https://github.com/joelguerrero/cloud-based-cad-paper/tree/master/static_mixer

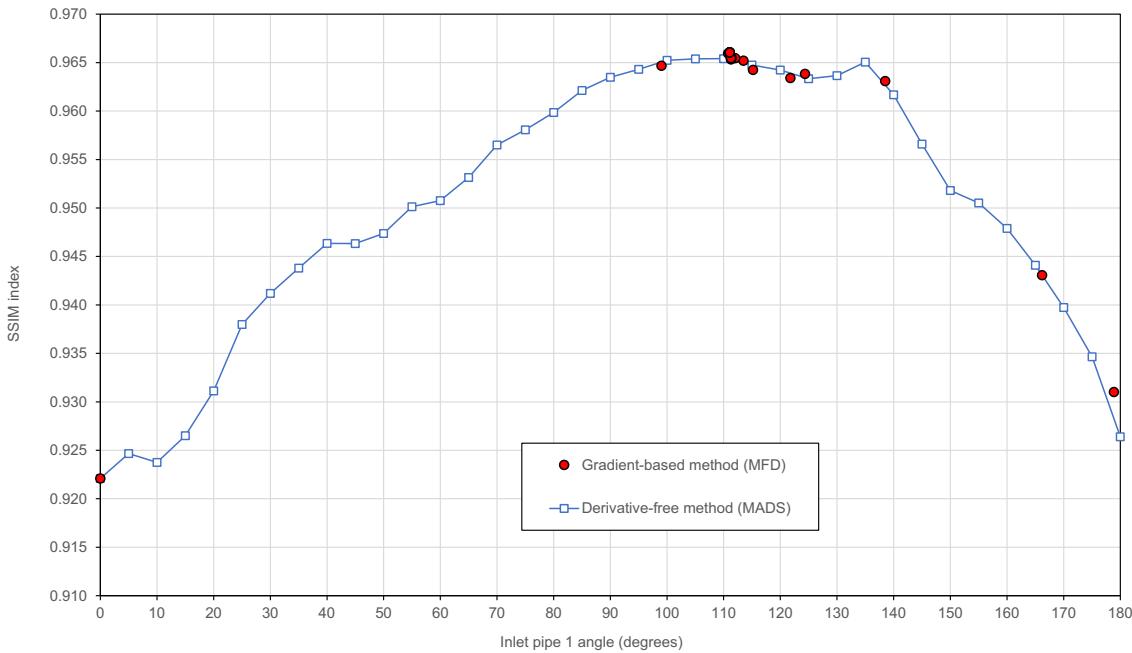


Figure 9. Comparison of the outcome of the DO and DSE studies. The QoI used was the SSIM index.

324 to convergence to the optimal solution in an iterative fashion, and in doing so, some areas of the design
 325 space may remain unexplored. Using the data of the DSE study, we can also get a good estimate of
 326 the maximum value of the SSIM index, or we can use the data to construct a meta-model, and then
 327 use any DO method to find the optimal value. Both methods, DO and DSE, have their advantages
 328 and drawbacks and often is a good practice to use a combination of both. That is, we first explore the
 329 design space in an inexpensive way, and then we use the information gathered from the DSE study to
 330 start a refined DO study.

331 In figure 10, we show the velocity distribution at the outlet surface for five cases of the DSE study.
 332 In this figure, we also show the SSIM index value, the geometry layout, and the target image. As
 333 previously stated, the goal of this study was to obtain a given velocity distribution at the outlet (target
 334 image) by changing the angle of the inlet pipe. Then, by using the SSIM index method (appendix A),
 335 we compare the target image and the image of the current configurations (as shown in figure 10). The
 336 closer the SSIM index is to one, the more similar the images are. We highlight that we are using a
 337 qualitative metric instead of the traditional quantitative metrics used in engineering design studies.
 338 We designed beforehand the desired appearance of the field at the outlet, and then, by comparing the
 339 images in the design loop, we found the best match for our qualitative metric.

340 Again, we stress the fact that the loop is fully automatic and fault-tolerant, and it can be run
 341 concurrently and on the cloud. For the DSE case, we run eight simulations concurrently, each one using
 342 four cores. For the DO case, we were limited by the number of derivatives that can be computed at the
 343 same time. As this case only has one design variable, only one derivative can be computed. Therefore,
 344 the maximum number of concurrent simulations achievable in this DO case was two (one function
 345 evaluation and one gradient evaluation using forward differences), and each concurrent evaluation
 346 was conducted using eight cores.

347 Let us now conduct a DSE study using three design variables, namely, the diameter of the inlet
 348 pipe one, the diameter of the inlet pipe two, and the angle of the inlet pipe one. Again, all the
 349 parametrical variables were defined in the Onshape's document and modified using the Python API.
 350 This study was conducted using 170 experiments, generated using the space filling latin hypercube

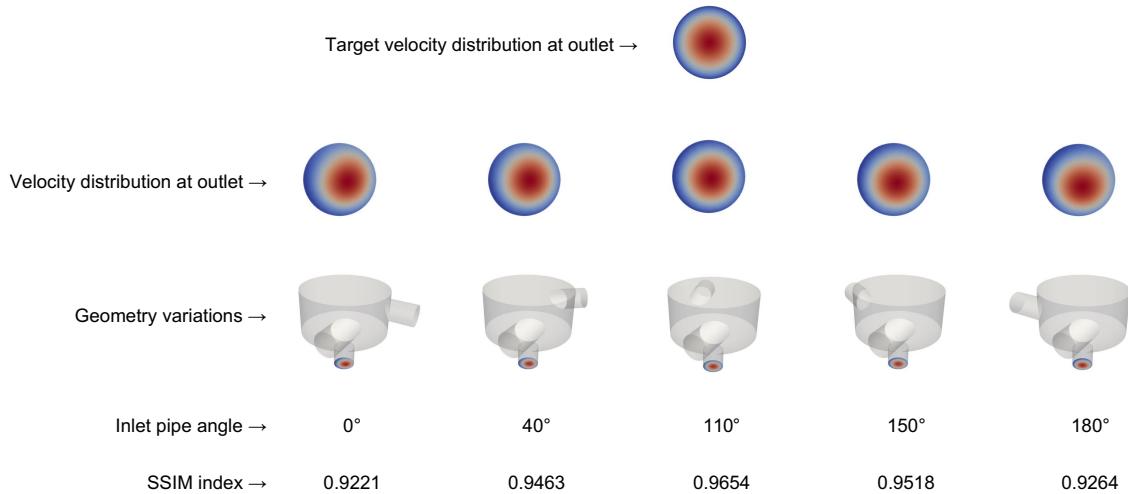


Figure 10. Qualitative comparison of the velocity distribution at the outlet. The SSIM method was used to compare the images. In the SSIM method, a value of 1 means that the images are identical. The target image is shown in the first row of the figure.

351 sampling method (LHS) [2]. The simulations were run concurrently (eight simulations at the same
 352 time), and each simulation was run in parallel using four cores.

353 In figure 11, we show another way to visualize high-dimensional data by using the parallel
 354 coordinates plot [43]. This kind of plot is extremely useful when visualizing and analyzing multivariate
 355 data, as it lets us identify how all variables are related. The highlighted line in figure 11 represents
 356 the best solution (maximum SSIM index value), and also shows the respective values of the design
 357 variables. In this DSE case, we can see that solutions that are better than the solution obtained using
 358 one design variable (SSIM = 0.9660), can be obtained by also changing the diameters of the inlet
 359 pipes. These solutions are shown in figure 12. It worth mentioning that the parallel coordinates plots
 360 implemented are interactive; this allows us to isolate a range of values in real-time. We can even
 361 change the order of the columns interactively and compare the slopes between variables. The scripts
 362 used for the parallel coordinates plots, as well as the data, are available at the following link⁶. The
 363 interactive parallel coordinates plot can be accessed at the following link⁷.

364 3.3. Two Ahmed bodies in platoon

365 In this case, we use the engineering design loop to conduct a parametric study. We compare the
 366 numerical results obtained with the current framework, against the experimental results obtained
 367 in references [44,45]; therefore, this is also a validation case. The simulations were conducted using
 368 OpenFOAM (version 7.0) [17,18]. To find the approximate solution of the governing equations, the
 369 SIMPLE pressure-velocity coupling method was used, together with the $k - \omega$ SST turbulence model
 370 with wall functions, and a second-order accurate and stable discretization method for the convective,
 371 diffusive, and gradient terms.

372 The study was conducted at different inter-vehicle spacing, an Ahmed body slant angle equal
 373 to 25 degrees, and an inlet velocity equal to 40 m/s. The QoI to measure is the normalized drag
 374 in platooning. In figure 13, we depict a sketch of the computational domain and the definition of
 375 the inter-vehicle spacing S . From the parametrization used when creating the solid model, the two

⁶ https://github.com/joelguerrero/cloud-based-cad-paper/tree/master/parallel_coordinates_dse_case

⁷ http://joelguerrero.github.io/parallel_coordinates_dse_case/

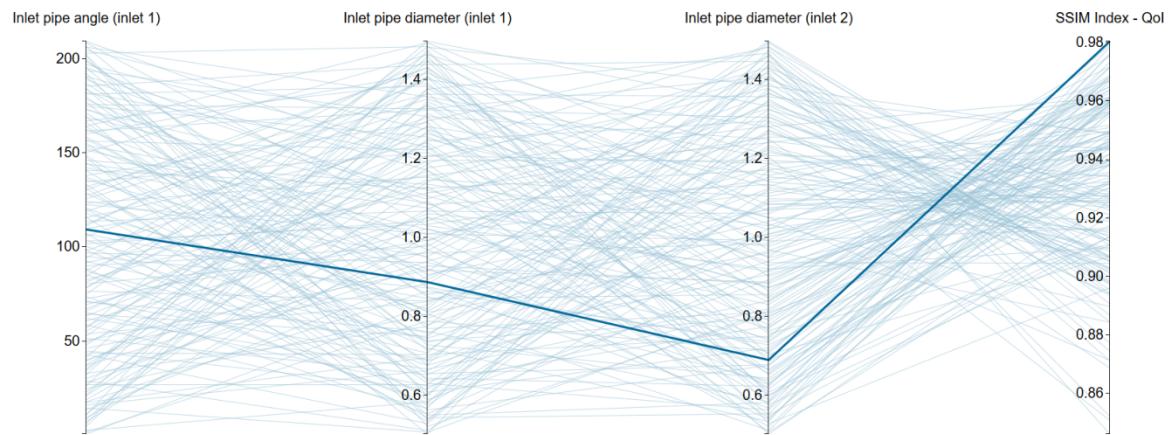


Figure 11. Parallel coordinates plot of the outcome of the DSE study using three design variables. The highlighted line represents the best solution.

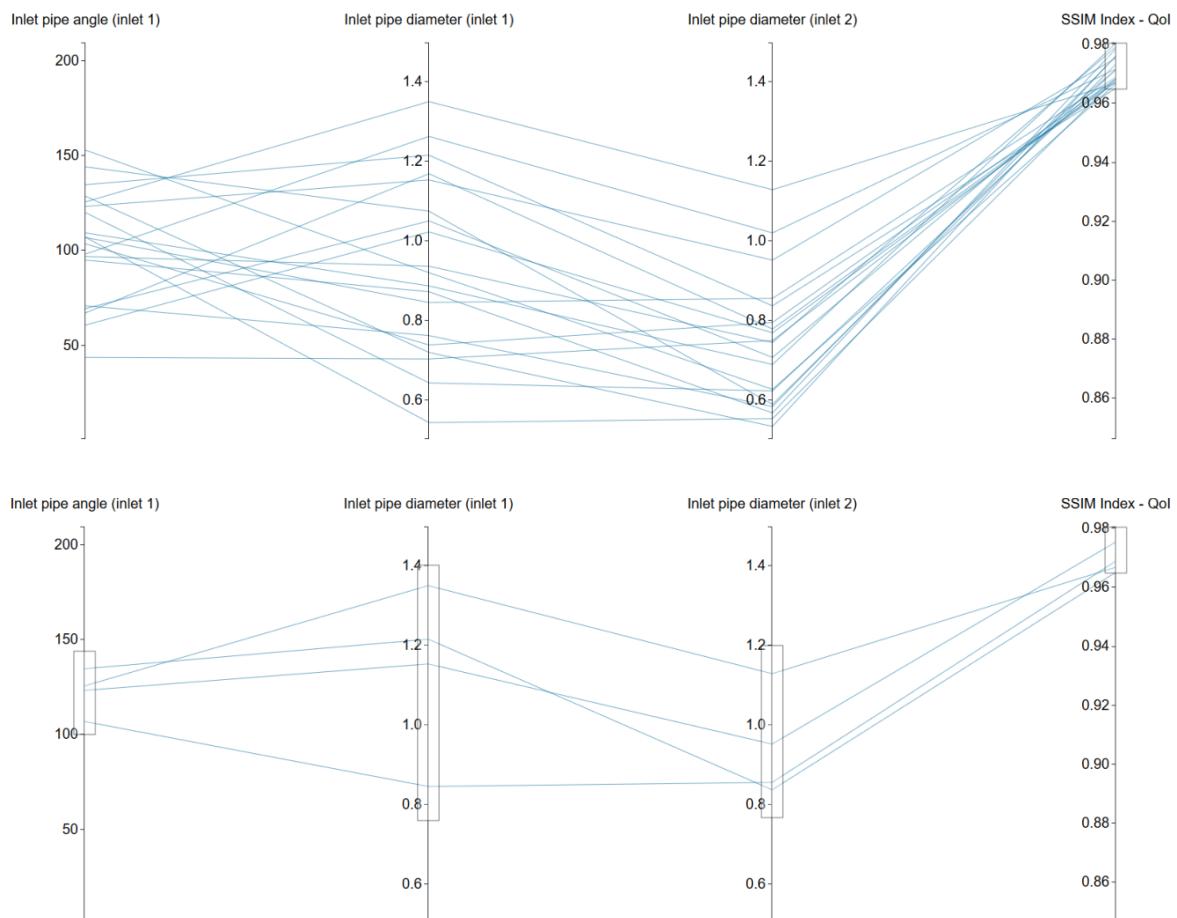


Figure 12. Parallel coordinates plot with filters. In the top image, the QoI has been filtered ($0.9660 \leq SSIM \leq 1$). In the bottom figure, we apply additional filters to the design variables.

³⁷⁶ Ahmed bodies can be simulated in any formation with different slant angles, where everything can be
³⁷⁷ controlled using configuration variables. The Onshape document with all the dimensions is available
³⁷⁸ at the following link ⁸.

³⁷⁹ In figure 14, we plot the outcome of this parametric study, where the normalized drag coefficient
³⁸⁰ in platooning is computed as follows,

$$C_{D_{Platooning}} = \frac{C_{D1}}{C_{D2}} \quad (9)$$

³⁸¹ in this equation, C_{D1} is the drag coefficient of the Ahmed body in a platoon position (front, back,
³⁸² sideways, or any combination), and C_{D2} is the drag coefficient of the single Ahmed body. From the
³⁸³ results presented in figure 14, it can be observed a satisfactory agreement between the numerical
³⁸⁴ and experimental values. It is worth mentioning that the simulations were run concurrently (four
³⁸⁵ simulations at the same time), and each simulation was run in parallel using six cores.

³⁸⁶ In this final application, we only conducted a parametrical study with one design variable.
³⁸⁷ However, this study served to demonstrate the usability of the framework for complex validation cases.
³⁸⁸ The reader should be aware that this case can be extended to more complex scenarios; for example, we
³⁸⁹ could simulate one Ahmed body overtaking the other one.

³⁹⁰ 4. Conclusions and future perspectives

³⁹¹ In this manuscript, we presented an engineering design framework to perform design
³⁹² optimization and design space exploration studies. The engineering design loop implemented, allows
³⁹³ for sequential and concurrent simulations (i.e., many simulations can be run at the same time), and
³⁹⁴ each simulation can be run in parallel; this allows to reduce the output time of the design loop
³⁹⁵ considerably. The optimization loop is fault-tolerant and software agnostic, and it can be interfaced
³⁹⁶ with any application able to interact using input/output files via a command-line interface. The code
³⁹⁷ coupling capabilities were provided by the library Dakota, and all the tools used in this work are
³⁹⁸ open-source and freeware.

³⁹⁹ Two novel features were introduced in the workflow. First, the use of a cloud-based parametric
⁴⁰⁰ CAD tool that gives engineers and designers complete control over the geometry during the design
⁴⁰¹ loop. This feature allows users to deploy the design loop in any platform as the installation is not
⁴⁰² required. It also lets the designers interact with the parametric CAD model using a programmatic
⁴⁰³ API. Introducing the CAD tool into the design loop has been traditionally a problem because most
⁴⁰⁴ of the CAD applications run in Windows OS. In contrast, the simulation software runs in Unix-like
⁴⁰⁵ OS. Furthermore, in traditional CAD tools is not possible to interact with the parametric model using
⁴⁰⁶ a programmatic environment; they take all the inputs via a graphical user interface that cannot be
⁴⁰⁷ controlled in an automatic design loop. The use of the cloud-based CAD tool allowed us to circumvent
⁴⁰⁸ these problems.

⁴⁰⁹ Secondly, the use of the SSIM index method to drive the design study. By using this metric, it is
⁴¹⁰ possible to compare images instead of integral quantities. We can now design beforehand how the
⁴¹¹ field will look like in a given location of the domain, and the design loop will try to find the best match
⁴¹² for that qualitative metric.

⁴¹³ From the numerical experiments presented, it was demonstrated the flexibility and usability
⁴¹⁴ of the proposed workflow to tackle engineering design problems using different approaches. As
⁴¹⁵ for the optimization strategy concerns, we used gradient-based methods, derivative-free methods,
⁴¹⁶ surrogate-based optimization, and design space exploration techniques. All the methods delivered
⁴¹⁷ satisfactory results. The SSIM index method also proved to be very robust and easy to implement.

⁸ <https://cad.onshape.com/documents/b691f01f6fadba22433180ad/w/28165b21b45b4fee07e761b8/e/93c2ec3a1d01f9149d0557b1>

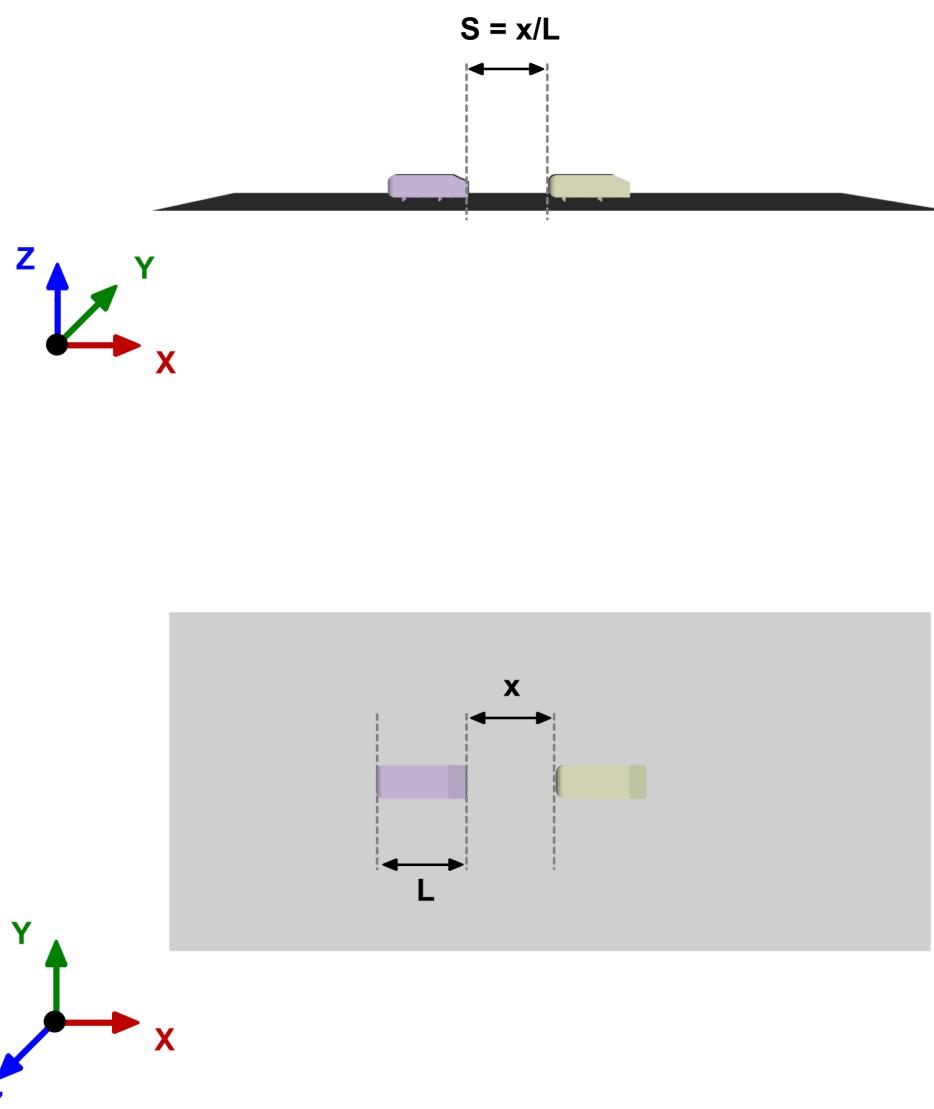


Figure 13. Spacing definition of the two Ahmed bodies, where x is the distance between the two bodies, L is the Ahmed body length, and S is the non-dimensional inter-vehicle spacing ($S = x/L$).

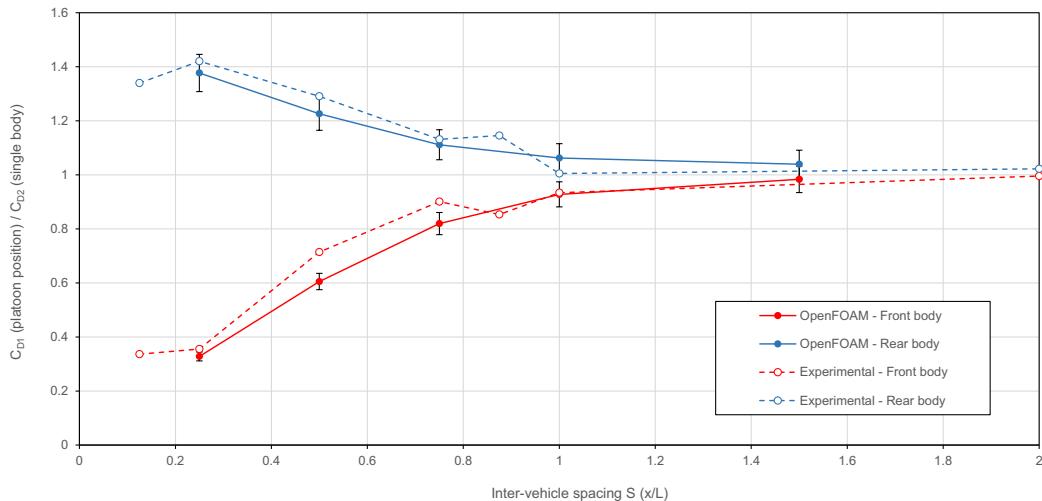


Figure 14. Normalized drag coefficient against inter-vehicle spacing S . The numerical results (continuous line) are shown with 5% error bars. The experimental results (dashed line) were taken from references [44,45].

418 This tool, together with reduced-order models and surrogate models, has the potential to open
 419 the door to generative design in CFD. We look forward to working in this field, together with machine
 420 learning techniques and more advanced image recognition algorithms.

421 **Author Contributions:** Conceptualization, J.G.; Methodology, J.G., L.M. and S.N.; Software, J.G., L.M. and S.N.;
 422 Validation, J.G.; Formal Analysis, J.G.; Investigation, J.G., L.M. and S.N.; Resources, J.G.; Data Curation, J.G.;
 423 Writing—Original Draft Preparation, J.G. and L.M.; Writing—Review & Editing, J.G., L.M. and S.N.; Visualization,
 424 J.G.; Supervision, J.G.

425 **Funding:** This research received no external funding.

426 **Acknowledgments:** The use of the computing resources at CINECA high performance computing center was
 427 possible thanks to the ISCRA grant, project IsC45 DO4EnD2. We acknowledge the support provided by the AWS
 428 Cloud Credits for Research program. This work was conducted as part of the "Computational Optimization in
 429 Fluid Dynamics" course held by Jan Pralits and Joel Guerrero at the University of Genoa.

430 **Conflicts of Interest:** The authors declare no conflict of interest.

431 Appendix A

432 Hereafter, we briefly describe the Structural Similarity Index (SSIM) method used in section 3.2 to
 433 measure the similarity between images. The SSIM is a method for predicting the perceived quality of
 434 digital television and cinematic pictures, as well as other kinds of digital images and videos.

435 Referring to a grey-scale image, a similarity index can be computed considering it as a
 436 bi-dimensional function of intensity [46]. The simplest and most commonly used similarity index is
 437 the mean squared error (MSE), which is obtained averaging the squared intensity difference between
 438 two pictures on each pixel [47]. However, the MSE, like many other mathematically defined indexes,
 439 is not able to take into account subjective quality measures (*i.e.*, human perception-based criteria, such
 440 as image structure comparison) [48]. For this reason, it can be misleading when it is necessary to find
 441 the image that is more similar to a reference one.

442 To avoid the problems related to the MSE, the SSIM index can be used. Based on how it is defined,
 443 the SSIM takes into account the structured information and the neighborhood dependencies that
 444 are usually present in natural images. The SSIM has been used with success in different research
 445 fields; for example, in reference [49], the authors used it to detect disturbances or blurring effects in
 446 a set of pictures. The authors also reported that it was not possible to do the same with the MSE. In
 447 reference [50], the SSIM index of flame images was used as a measure of the burning state in a sintering

process. By using a small number of samples, the authors were able to recognize the burning state with satisfactory accuracy thanks to the SSIM index. In reference [51], a hand gesture recognition study based on both MSE and SSIM was presented, and it was concluded that both techniques could be used for gesture recognition. In addition, it was also found that the SSIM was superior to the MSE, as it was insensitive to small imperfections in the reconstructed image caused by thresholding.

Considering two different image discrete signals, let us say x and y , the similarity evaluation is based on three characteristics: luminance, contrast, and structure [47]. The luminance μ_x of each signal is computed as the mean intensity, as follows,

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i \quad (\text{A1})$$

where N is the number of pixels.

The luminance comparison between x and y is then performed defining the function $l(x, y)$,

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (\text{A2})$$

where C_1 is a constant used to avoid instabilities when the denominator is close to zero.

The contrast σ_x is estimated as the standard deviation of the image signal, and is computed as follows,

$$\sigma_x = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2} \quad (\text{A3})$$

The contrast comparison function $c(x, y)$ is similar to equation A2, and it also includes a constant to avoid instabilities (C_2).

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (\text{A4})$$

The structure comparison can be performed by defining the function $s(x, y)$,

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (\text{A5})$$

where σ_{xy} is specified as follows,

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y) \quad (\text{A6})$$

Finally, by combining equations A2, A4 and A5, it is possible to obtain the SSIM index between x and y , as follows,

$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma \quad (\text{A7})$$

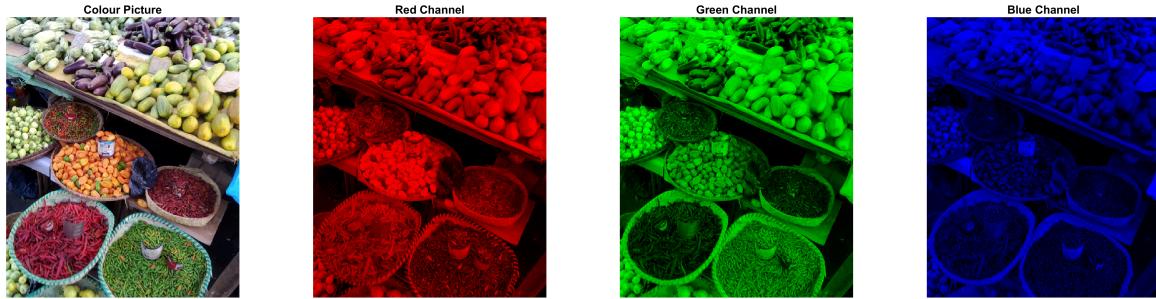


Figure A1. Separation of red, green and blue channels of a color picture. Image courtesy of Diego Rattazzi (diego.rattazzi@edu.unige.it).

where α , β , and γ are positive parameters used as weights factors to set the importance of $l(x, y)$, $c(x, y)$ and $s(x, y)$ when computing the SSIM index. A simplified expression of equation A7 can be obtained by setting $l(x, y)$, $c(x, y)$, $s(x, y)$, and C_3 to the following values [47],

$$\alpha = 1 \quad \beta = 1 \quad \gamma = 1 \quad C_3 = \frac{C_2}{2} \quad (\text{A8})$$

thus obtaining the following expression for SSIM (which is the form of the equation A7 used in this work),

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \quad (\text{A9})$$

To analyze the images, we use the Python library scikit-image [52], which is a collection of algorithms for image processing. The images to compare are saved as color images in digital format (e.g., Portable Network Graphics or PNG format). However, this procedure was designed for grey-scale images, as stated at the beginning of this section. Thus, it is necessary to separate the three different color channels (red, green, and blue), as shown in figure A1. This is done by using the Python function `imread` to import the digital image (in PNG format) as a `uint8` three-dimensional array. At this point, each channel is a monochrome picture so that it can be treated as a grey-scale picture, and its SSIM index can be computed by using equation A9. The SSIM of the original digital image can be finally obtained as the average of the SSIMs of the three color channels. The computation of the SSIM of the separate channels and their averaging is performed using the `compare_ssim` function implemented in the Python library scikit-image. The SSIM index value is a number between 0 and 1, where 1 means a perfect matching between the images. That is, the closer the value is to 1, the more similar the images are. A sample python script can be found at the following link ⁹.

485

486 References

- 487 1. Mattson, C.A. Design exploration. <https://design.byu.edu/blog/design-exploration-presentation-given-stanford-university-15-jan-2014>, Last accessed: February 2020.
- 488 2. Forrester, A.; Sobester, A.; Keane, A. *Engineering Design via Surrogate Modeling. A Practical Guide*; Wiley, 2008.

⁹ <https://github.com/joelguerrero/cloud-based-cad-paper/tree/master/SSIM>

- 491 3. Guerrero, J.; Cominetti, A.; Pralits, J.; Villa, D. Surrogate-Based Optimization Using an Open-Source
492 Framework: The Bulbous Bow Shape Optimization Case. *Math. Comput. Appl.* 2018, **24**.
493 doi:<https://doi.org/10.3390/mca23040060>.
- 494 4. Romero, V.J.; Swiler, L.P.; Giunta, A.A. Construction of response surfaces based on
495 progressive lattice-sampling experimental designs. *Structural Safety* 2004, **26**, 201–219.
496 doi:<https://doi.org/10.1016/j.strusafe.2003.03.001>.
- 497 5. Kochenderfer, M.; Wheeler, T. *Algorithms for Optimization*; MIT Press, 2019.
- 498 6. Gen, M.; Cheng, R. *Genetic Algorithms and Engineering Optimization*; Wiley-Interscience, 2000.
- 499 7. Vanderplaats, G. *Multidiscipline Design Optimization*; Vanderplaats Research & Development, Inc,
500 2007.
- 501 8. Papalambros, P.; Wilde, D. *Principles of Optimal Design. Modeling and Computation*; Cambridge
502 University Press, 2017.
- 503 9. Nocedal, J.; Wright, S. *Numerical Optimization*; Springer, 2006.
- 504 10. Chong, E.; Zak, S. *An Introduction to Optimization*; Wiley, 2013.
- 505 11. Sobiesczanski-Sobieski, J.; Morris, A.; van Tooren, M.; Rocca, G.L.; Yao, W. *Multidisciplinary Design
506 Optimization Supported by Knowledge Based Engineering*; Wiley, 2015.
- 507 12. Chauhan, S.; Hwang, J.; Martins, J. An automated selection algorithm for nonlinear solvers in MDO.
508 *Structural and Multidisciplinary Optimization* 2018, **58**, 349–377.
- 509 13. Martins, J. The adjoint method in multidisciplinary design optimization - Special session in honor of
510 Antony Jameson's 85th birthday. AIAA SciTech Forum, 2020.
- 511 14. Vassberg, J.C.; Jameson, A. *Introduction to Optimization and Multidisciplinary Design Part I:
512 Theoretical Background for Aerodynamic Shape Optimization*; Lecture Series 2016-03, Von Karman
513 Institute, Brussels, Belgium, 2016.
- 514 15. Keane, A.J.; Nair, P.B. *Computational Approaches for Aerospace Design: The Pursuit of Excellence*; John
515 Wiley & Sons, 2005.
- 516 16. Martins, J.R.R.A.; Lambe, A.B. Multidisciplinary Design Optimization: A Survey of Architectures.
517 *AIAA Journal* 2013, **51**, 2049–2075. doi:<https://doi.org/10.2514/1.J051895>.
- 518 17. The OpenFOAM Foundation. <http://www.openfoam.org>, Accessed: February 2020.
- 519 18. Weller, H.G.; Tabor, G.; Jasak, H.; Fureby, C. A tensorial approach to computational
520 continuum mechanics using object-oriented techniques. *Computer in Physics* 1998, **12**, 620–631.
521 doi:<https://doi.org/10.1063/1.168744>.
- 522 19. Dakota Web Page. <https://dakota.sandia.gov/>, 2018. Accessed: February 2020.
- 523 20. Adams, B.M.; Eldred, M.S.; Geraci, G.; Hooper, R.W.; Jakeman, J.D.; Maupin, K.A.; Monschke, J.A.;
524 Rushdi, A.A.; Stephens, J.A.; Swiler, L.P.; Wildey, T.M.; Bohnhoff, W.J.; Dalbey, K.R.; Ebeida, M.S.;
525 Eddy, J.P.; Hough, P.D.; Khalil, M.; Hu, K.T.; Ridgway, E.M.; Vigil, D.M.; Winokur, J.G. *Dakota,
526 A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation,
527 Uncertainty Quantification, and Sensitivity Analysis: Version 6.10 User Manual*. Sandia National
528 Laboratories, 2019.
- 529 21. The Visualization Toolkit (VTK). <http://www.vtk.org>, Accessed: February 2020.
- 530 22. Onshape product development platform. <http://www.onshape.com>, Accessed: February 2020.
- 531 23. Slotnick, J.; Khodadoust, A.; Alonso, J.; Darmofal, D.; Gropp, W.; Lurie, E.; Mavriplis, D. CFD Vision
532 2030 Study: A Path To Revolutionary Computational Aerosciences. Technical report, NASA, 2014.
- 533 24. Daymo, E.; Tonkovich, A.L.; Hettel, M.; Guerrero, J. Accelerating reactor development with accessible
534 simulation and automated optimization tools. *Chemical Engineering and Processing - Process
535 Intensification* 2019, **142**, 107582. doi:<https://doi.org/10.1016/j.cep.2019.107582>.
- 536 25. Xia, C.C.; Gou, Y.J.; Li, S.H.; Chen, W.F.; Shao, C. An Automatic Aerodynamic Shape Optimisation
537 Framework Based on DAKOTA. *IOP Conference Series: Materials Science and Engineering* 2018,
538 **408**, 012021. doi:[10.1088/1757-899x/408/1/012021](https://doi.org/10.1088/1757-899x/408/1/012021).
- 539 26. Byrne, J.; Cardiff, P.; Brabazon, A.; O'Neill, M. Evolving parametric aircraft
540 models for design exploration. *Journal of Neurocomputing* 2014, **142**, 39–47.
541 doi:<https://doi.org/10.1016/j.neucom.2014.04.004>.
- 542 27. Ohm, A.; Tetursson, H. Automated CFD optimization of a small hydro turbine for water distribution
543 networks. Technical report, Chalmers University of Technology, 2017.

- 544 28. Sousa, J.; Gorlé, C. Computational urban flow predictions with Bayesian inference:
545 Validation with field data. *Building and Environment* 2019, **154**, 13 – 22.
546 doi:<https://doi.org/10.1016/j.buildenv.2019.02.028>.
- 547 29. Kiani, H.; Karimi, F.; Labbafi, M.; Fathi, M. A novel inverse numerical modeling
548 method for the estimation of water and salt mass transfer coefficients during ultrasonic
549 assisted-osmotic dehydration of cucumber cubes. *Ultrasonics Sonochemistry* 2018, **44**, 171 – 176.
550 doi:<https://doi.org/10.1016/j.ulstsonch.2018.02.003>.
- 551 30. Habla, F.; Fernandes, C.; Maier, M.; Densky, L.; Ferras, L.; Rajkumar, A.; Carneiro, O.;
552 Hinrichsen, O.; Nobrega, J.M. Development and validation of a model for the temperature
553 distribution in the extrusion calibration stage. *Applied Thermal Engineering* 2016, **100**, 538 – 552.
554 doi:<https://doi.org/10.1016/j.applthermaleng.2016.01.166>.
- 555 31. Khamlaj, T.A.; Rumpfkeil, M.P. Analysis and optimization of ducted wind turbines. *Energy* 2018,
556 **162**, 1234 – 1252. doi:<https://doi.org/10.1016/j.energy.2018.08.106>.
- 557 32. Montoya, M.C.; Nieto, F.; Hernandez, S.; Kusano, I.; Alvarez, A.; Jurado, J. CFD-based aeroelastic
558 characterization of streamlined bridge deck cross-sections subject to shape modifications using
559 surrogate models. *Journal of Wind Engineering and Industrial Aerodynamics* 2018, **177**, 405 – 428.
560 doi:<https://doi.org/10.1016/j.jweia.2018.01.014>.
- 561 33. Kelm, S.; Müller, H.; Hundhausen, A.; Druska, C.; Kuhr, A.; Allelein, H.J. Development of a
562 multi-dimensional wall-function approach for wall condensation. *Nuclear Engineering and Design*
563 2019, **353**, 110239. doi:<https://doi.org/10.1016/j.nucengdes.2019.110239>.
- 564 34. Zoutendijk, G. *Methods of Feasible Directions : a Study in Linear and Non-Linear Programming*;
565 Elsevier, 1960.
- 566 35. Vanderplaats, G.N. An efficient feasible directions algorithm for design synthesis. *AIAA Journal* 1984,
567 **22**, 1633–1640. doi:<https://doi.org/10.2514/3.8829>.
- 568 36. Le Digabel, S. Algorithm 909: NOMAD: Nonlinear Optimization with the MADS Algorithm. *ACM
Trans. Math. Softw.* 2011, **37**. doi:<10.1145/1916461.1916468>.
- 570 37. Guerrero, J. Opportunities and challenges in CFD optimization: Open Source technology and the
571 Cloud. The sixth symposium on OpenFOAM® in Wind Energy (SOWE) - Gotland, Sweden, 2018.
- 572 38. Oliver, M.; Webster, R. Kriging: A Method of Interpolation for Geographical Information Systems.
573 *International Journal of Geographic Information Systems* 1990, **4**, 313–332.
- 574 39. Adams, B.M.; Eldred, M.S.; Geraci, G.; Hooper, R.W.; Jakeman, J.D.; Maupin, K.A.; Monschke, J.A.;
575 Rushdi, A.A.; Stephens, J.A.; Swiler, L.P.; Wildey, T.M.; Bohnhoff, W.J.; Dalbey, K.R.; Ebeida, M.S.;
576 Eddy, J.P.; Hough, P.D.; Khalil, M.; Hu, K.T.; Ridgway, E.M.; Vigil, D.M.; Winokur, J.G. *Dakota,
577 A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation,
578 Uncertainty Quantification, and Sensitivity Analysis: Version 6.10 Theory Manual*. Sandia National
579 Laboratories, 2014.
- 580 40. Dalbey, K.R.; Giunta, A.A.; Richards, M.D.; Cyr, E.C.; Swiler, L.P.; Brown, S.L.; Eldred, M.S.; Adams,
581 B.M. *Surfpack User's Manual Version 1.1*. Sandia National Laboratories, 2013.
- 582 41. Dalbey, K.R. Efficient and Robust Gradient Enhanced Kriging Emulators. Technical report, Sandia
583 National Laboratories, 2013.
- 584 42. Giunta, A.A.; Watson, L. A comparison of approximation modeling techniques: polynomial versus
585 interpolating models. 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and
586 Optimization. NASA Langley Technical Report Server, 1998.
- 587 43. Inselberg, A. *Parallel Coordinates, Visual Multidimensional Geometry and Its Applications*; Springer,
588 2009.
- 589 44. Pagliarella, R.M.; Watkins, S.; Tempia, A. Aerodynamic Performance of Vehicles in Platoons: The
590 Influence of Backlight Angles. SAE World Congress & Exhibition. SAE International, 2007.
591 doi:<https://doi.org/10.4271/2007-01-1547>.
- 592 45. Pagliarella, R.M. On the aerodynamic performance of automotive vehicle platoons featuring pre and
593 post-critical leading forms. Technical report, RMIT University, 2009.
- 594 46. Sampat, M.P.; Wang, Z.; Gupta, S.; Bovik, A.C.; Markey, M.K. Complex wavelet structural
595 similarity: A new image similarity index. *IEEE Transactions on Image Processing* 2009, **18**, 2385–2401.
596 doi:<10.1109/TIP.2009.2025923>.

- 597 47. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error
598 Visibility to Structural Similarity. *IEEE Transactions on Image Processing* 2004, *13*, 600–612.
599 doi:10.1109/TIP.2003.819861.
- 600 48. Eskicioglu, A.M.; Fisher, P.S. Image Quality Measures and Their Performance. *IEEE Transactions on
601 Communications* 1995, *43*, 2959–2965. doi:10.1109/26.477498.
- 602 49. Ndajah, P.; Kikuchi, H.; Yukawa, M.; Watanabe, H.; Muramatsu, S. SSIM image quality metric for
603 denoised images. International Conference on Visualization, Imaging and Simulation, 2010, pp. 53–57.
- 604 50. Lin, Y.; Chai, L.; Zhang, J.; Zhou, X. On-line burning state recognition for sintering process using SSIM
605 index of flame images. Proceedings of the 11th World Congress on Intelligent Control and Automation,
606 2014, p. 2352–2357. doi:10.1109/WCICA.2014.7053089.
- 607 51. Priyal, S.P.; Bora, P.K. A study on static hand gesture recognition using moments. 2010
608 International Conference on Signal Processing and Communications (SPCOM), 2010, pp. 1–5.
609 doi:10.1109/SPCOM.2010.5560535.
- 610 52. Van der Walt, S.; Schönberger, J.L.; Nunez-Iglesias, J.; Boulogne, F.; Warner, J.D.; Yager,
611 N.; Gouillart, E.; Yu, T. scikit-image: image processing in Python. *PeerJ* 2014, *2*, e453.
612 doi:<https://doi.org/10.7717/peerj.453>.

613 © 2020 by the authors. Submitted to *Journal Not Specified* for possible open access
614 publication under the terms and conditions of the Creative Commons Attribution (CC BY) license
615 (<http://creativecommons.org/licenses/by/4.0/>).