D5.3 Design Space Exploration & AI: methods supporting constrained and high-dimensional problems

Author, company:

- Max Baan, ParaPy
- Petter Andersson, GKN Aerospace Engines
- Massimo Panarotto, Chalmers University of Technology
- Alejandro Pradas Gomez, Chalmers University of Technology
- Petter Krus, Linköping University
- Jente Sonneveld, Delft University of Technology

Version:
1.4

Date:
February 5, 2024

Status:
Final / Released

Confidentiality:
Public

Change log

Revision	Date	Prepared by	Checked by	Description
1.0	13/11/2023	Max Baan		First setup of this deliverable
1.1	19/01/2024	Jente Sonneveld	Max Baan	Added multi architecture design exploration contribution
1.2	21/01/2024	Massimo Panarotto	Max Baan	Added ML support for value assessment contribution
1.3	24/01/2024	Petter Andersson, Alejandro Pradas Gomez, Petter Krus	Max Baan	Added contributions on surrogate modelling, EFM – KBE, SVD and NN and generative AI

Table of contents

Acronyms

Acronym	Definition
AI	Artificial Intelligence
CAD	Computer Aided Design
CCM	Configurable Component Modeller (PE Geometry software)
CFR	Code of Federal Regulations
CS	Certification Specification
DEFAINE	Not explicitly defined
DoE	Design of Experiments
DSC	Design Study Configuration
EF-M	Enhanced Function-Means
FEM	Finite Element Model
FOGV	Fan Outlet Guide Vane
KBE	Knowledge Based Engineering
MDAO	Multi-disciplinary Design Analysis and Optimization
RSM	Response Surface Modelling
PIDO	Process Integration and Design Optimization
SV	Surplus Value
SMT	Surrogate Modelling Toolbox
TRS	Turbine Rear Structure
VCS	Value Creation Strategy
VD	Value Drivers
WEM	Whole Engine Model
XML	eXtensible Markup Language

1. Introduction

In the ever-evolving landscape of engineering and design, the quest for optimal solutions often traverses complex and multi-faceted design spaces. Design Space Exploration (DSE) stands at the forefront of this endeavour, serving as a pivotal process in navigating the myriad of possibilities to unearth optimal solutions. However, with the advent of increasingly complex problems characterized by constraints and high-dimensionality, conventional approaches fall short in providing efficient and effective solutions.

This report delves into Design Space Exploration and Artificial Intelligence and the synergy between them, discussing methods and strategies that bolster the exploration of constrained and high-dimensional design spaces. Throughout this report, methodologies and strategies, aimed at enhancing DSE capabilities and employed within the DEFAINE project, are presented, An overview of the industrial applications of these methodologies is included, elucidating their impact on real-world scenarios.

2. Multi-architecture design exploration using dynamic workflows

This section of the report summarizes an approach for the exploration and optimization of the architectural design space. The approach leverages dynamic reformulating workflows to enable the multiple architectures to be evaluated in a single MDAO workflow. The methodology and use-case application are described in detail in **[1]** and deliverable D4.1.3 **[2]**.

2.1. Background

Consolidated strategies and algorithms can be used to perform multidisciplinary and multi-objective optimization of engineering products, as far as the architecture of the system to be explored is fixed. Accounting for the complete system architecture design space in an optimization process is very challenging because of the presence of integer and categorical design variables which lead to a combinatorial explosion of designs to be evaluated. An extra challenge arises because of the hierarchical relationships that may exist between design variables, i.e., the number and/or existence of certain design variables may depend on the value assumed by other design variables. In fact, the hierarchical nature is generally a consequence of the aforementioned categorical variables, as the presence of certain components in the system architecture comes with the necessary design variables to define said components. This makes the design vector of the optimization problem dynamic and unknown a priori.

2.2. Methodology

Within DEFAINE the challenge of dealing with hierarchical variables in multi-architecture design exploration and optimization has been addressed. The methodology is based on the concept of dynamically reformulating sub-workflows. The problem is split up by putting high-level architecture variables in a main workflow and their dependent variables in a dynamically reformulating sub workflow. The methodology consists of 3 main steps, listed below.

- **Design study configuration:** As a means to configure a hierarchical design study, the eXtensible Markup Language (XML)-based Design Study Configuration (DSC) schema was developed. In a DSC file, the hierarchical variable structure of a product can be configured, allowing for unknown variable quantities and types.

- **Nested workflow formulation**: Once a design study is configured by constructing a DSC file, for each configured design step an MDAO workflow can be formulated capable of evaluating all the specified variables. The result is a nested workflow, each level representing a design step. The workflows follow a standard format, an example is given in Figure 1 for a 2-step design study**.**

- **Dynamic workflow reformulation:** during execution, in each iteration of the main workflow, a dynamic reformulation of the sub workflow takes place. Based on the information that becomes available during the execution of the main workflow, the sub workflow formulation can be completed and executed.

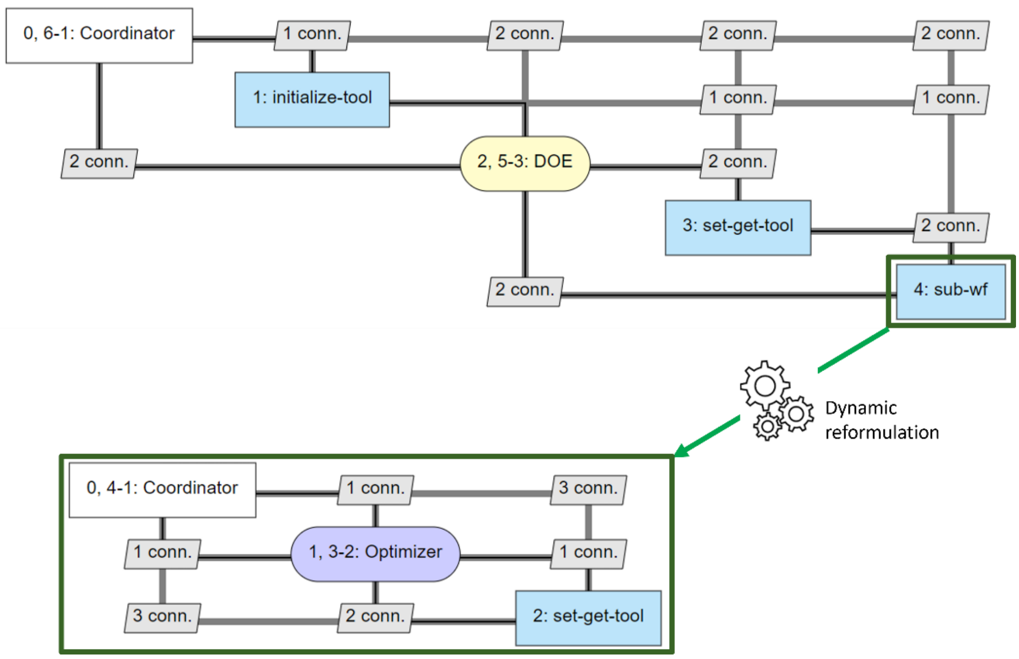


Figure 1: nested workflow featuring a dynamically reformulating sub-workflow

2.3. Industrial application

The methodology has been implemented for a GKN-Fokker Aerostructures' use-case. In this use-case GKN Fokker's modelling and analysis tool MDM was used to explore the architecture design space of a moveable. In Figure 2, an MDM aircraft moveable model instance is shown. In the next sections, the implementation and results are summarized.

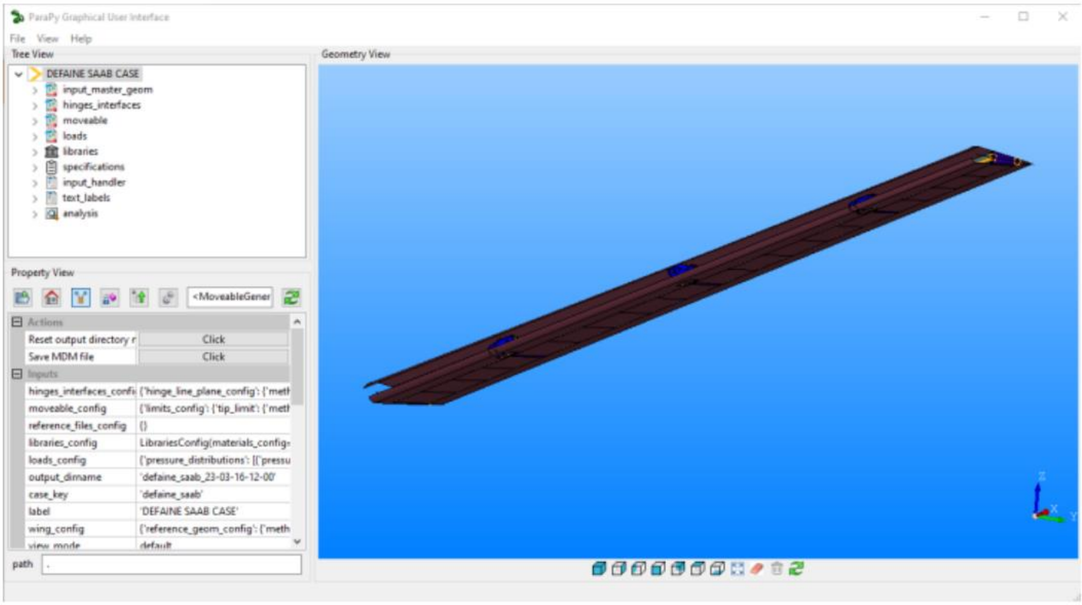


Figure 2: MDM moveable model instance

2.3.1. Implementation

In Figure 3 an overview of the implemented multi-architecture design exploration approach is shown. All the enabling technologies are quickly summarized below.

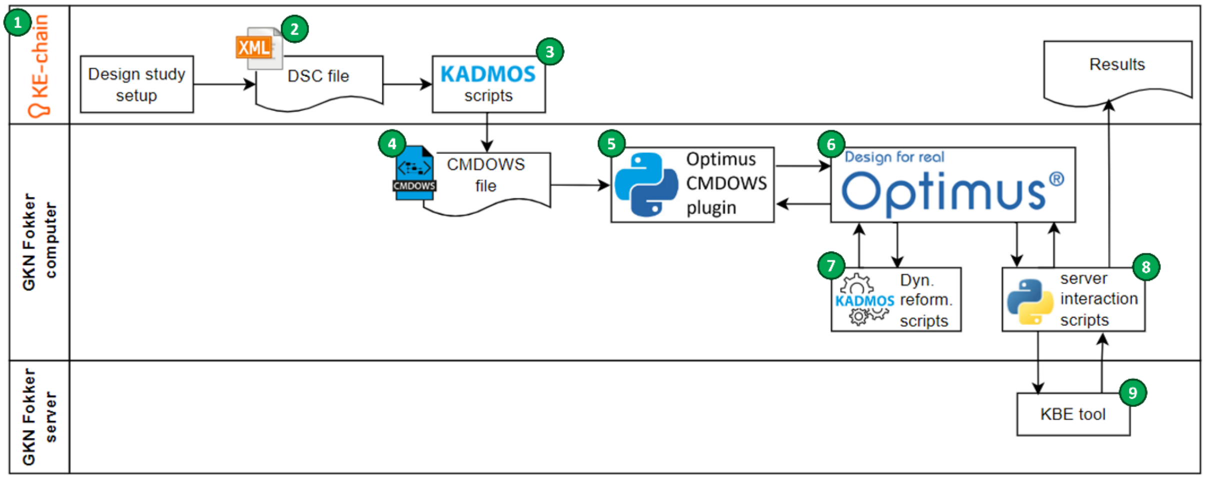


Figure 3: Overview of collaborative architecture optimization design study setup and execution process

1. **KE-chain**: a web-based collaborative environment enabling multiple partners to collaborate on a single design study.
2. **DSC file**: a XML-based file format for specifying multi-level architecture design studies (see D4.3.2 [3]).
3. **KADMOS** based scripts are used to convert the DSC file into a fully formulated main workflow and, depending on the amount of specified design steps, several incomplete sub-workflow formulations (example in Figure 1), all saved in a single CMDOWS file. (see D4.1.x [4] [3] [2])
4. **CMDOWS**, exchange standard for MDAO workflow formulations (see D4.2.1 [5])
5. **CMDOWS-Optimus plugin** A conversion tool that has been developed to convert CMDOWS files into executable Optimus workflows. (see D4.2.1 [5])
6. **Optimus,** a commercial process integration and design optimization (PIDO) tool by NOESIS Solutions[1].
7. **KADMOS** dynamic reformulation scripts that take care of completing the formulation of the sub-workflow based on information that has become available during the execution of the main workflow.
8. **Server interaction scripts,** Python-based scripts that enable communication with a server-based KBE tool using set and get commands to impose inputs and extract outputs
9. **KBE tool**, an instance of a knowledge-based engineering tool living on a server, in this case GKN Fokker's MDM tool based on ParaPy.

2.3.2. Results

A first implementation of the proposed methodology was tested on a GKN-Fokker aileron architecture optimization use-case. After an initial design study setup, the workflows could be automatically generated and executed in Optimus, see Figure 4. The approach proved effective in dealing with the hierarchical mixed-integer design space and enabled a flexible set-up of a nested workflow, consisting of an outer DOE loop and an inner, dynamically formulated, loop for structural optimization. Collaborative trade off studies of various aileron architectures could be set in a time efficient manner due to the high level of achieved process automation. A 72% reduction

[1] https://www.noesissolutions.com/ accessed: 19/01/2024

in simulation workflow setup time and a 96% reduction in simulation workflow update time was observed.

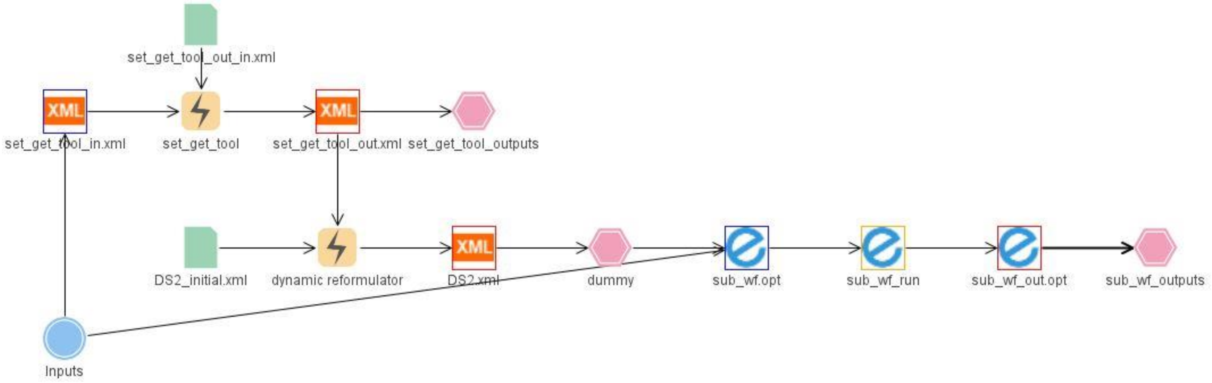


Figure 4 Automatically generated executable Optimus workflow featuring a dynamically reformulating sub-workflow.

3. Functional and physical modelling connection via Enhanced Function-Means and KBE.

This section of the report presents a methodology developed by Chalmers University of Technology on Design Space Exploration. The methodology focuses on combining the functional models (Enhanced Function-Means trees) with physical Models (KBE Applications). A complete description of the methodology will be presented in a journal article at the Aerospace Journal during 2024.

3.1. Background

3.1.1. Functional Modelling and Enhanced Function-Means (EF-M)

Functional Modelling represents the functionality or behaviour of a system of products, providing insights into how it operates. Enhanced Function-Means (EF-M) [1] is one of the methods that focuses on how each functional requirement of the product can be fulfilled by different design solutions. Each design solution alternative represents a variant of the design, and therefore represents a generation of different architectures in the Design Space Exploration. The model is architected in a tree format with Functional Requirements having Design Solution child objects, which in turn can contain further Functional Requirements objects. For more information about the history and main functionalities on these models, See DEFAINE deliverable D5.4.2.
EF-Ms allows for top level design space exploration and evaluation of metrics that do not require the physical embodiment of the product for its evaluation, such as risk management, development process efficiency or integrability [2]. However, this methodology lacks the ability to physically embody the design concept [3]. Some authors have been able to map EF-Ms to physical models via custom User Defined Features in CAD [4], but a mapping to an analysis model needs to be done outside the method. The main advantage of EF-M models with respect to design space exploration is that they can contain many different variants, and with specific software, instantiate all those variants easily.

3.1.2. Knowledge Based Engineering (KBE) Systems

Knowledge based engineering (KBE) stands at the cross point of diverse fundamental disciplines, such as artificial intelligence (AI), computer aided design (CAD) and computer programming, [5]. As such, KBE primitives have direct access to geometrical variables and derived properties. KBE applications therefore are ideal for design space explorations where geometrical and architectural changes are required. In addition, its programmatic and multidisciplinary nature allows for the generation or connection to analysis models tools. Therefore, they are used in industry for the design space exploration of products that heavily rely on geometrical characteristics, such as aerospace.

3.1.3. Combining Functional Models (E-FMs) and KBE Models

During the early stages of the product development process for aerospace components, many alternatives are being considered. Traditionally, the aerospace industry has followed a low-risk approach, favouring incremental architectures changes. However, the increasing global concern regarding climate change has shifted the mindset and increased the interest in innovative and unconventional solutions. To evaluate all design alternatives, the mechanical performance of such structural configurations shall be evaluated, requiring a physical model.

By combining EF-M and KBE models, the design space exploration can be widened. EF-M models can be used to control the architecture, variants, alternatives, and incompatibilities of all design solution options, while KBE models can provide the physical performance metrics.

3.2. Methodology

The Design Space Exploration in this methodology is performed via a design of experiments (DoE). The methodology follows a 3 step process.

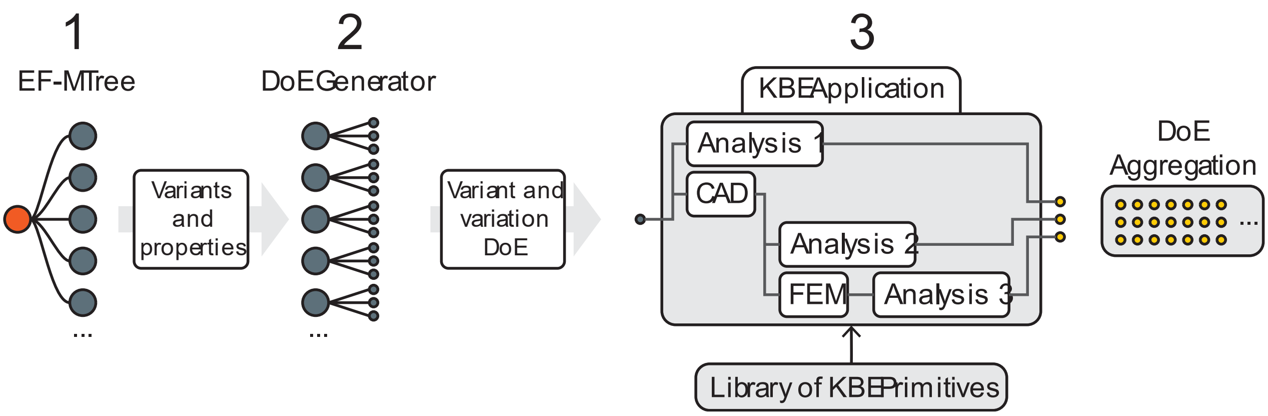


Figure 5: Overview of the EF-M to KBE Methodology steps

The purpose of the first step is to define the architecture and parameters to consider. For that, a traditional EF-M tree is generated containing all design solution relationships. In addition, the ranges or options of the DoE parameters are added as properties of the relevant design solution. Once the tree has been completed, the combinatorial alternatives or design solutions are generated. This step is demonstrated using the CCM tool developed by PE Geometry.

The second step is to generate the DoE table with the desired number of experiments. It considers both the architectural changes as well as the parametric changes within those architectures. This step has been implemented in DEFAINE by developing a python module.

The third step is to execute the DoE cases, for which a KBE application is generated and updated with the specific case. The application iterates though every case in the DoE, that generates the geometrical configurations and evaluates some of the performance metrics. Other performance metrics are sent to external programs. It has been implemented using the KBE System developed by ParaPy.

3.3. Industrial Applications

The methodology is applied to the GKN Aircraft Engine Component use case. A Turbine Rear Structure component has been evaluated using this methodology, see Figure 6.

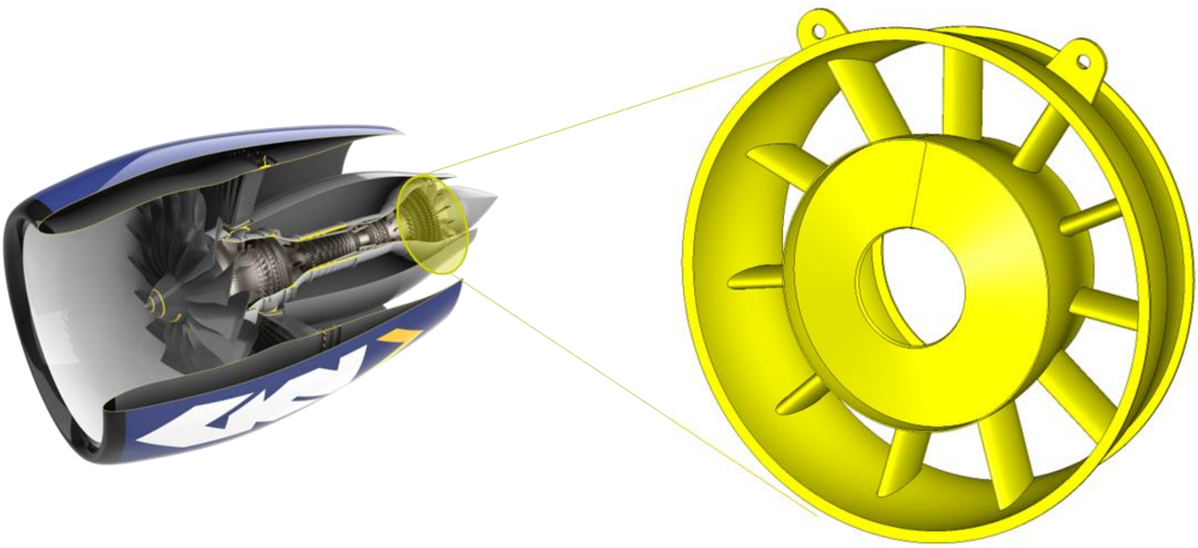


Figure 6: Location of a Turbine Rear Structure (TRS) on a conventional engine. The figure also shows the geometrical model of one of the design use cases generated by the ParaPy KBE application

The architectural variants included in the design space exploration are a wall-integrated containment ring or an independent containment ring. Parametric variables are described in Table 1.

Table 1: Configuration of parametric variables

Variable Name	Feature	Variable type	More information
Lug thickness	Lugs	continuous	Bounds = (10, 20) [mm]
Wall thickness	Outer Case	continuous	Bounds = (1, 15) [mm]
Strut thickness	Vane Assembly	continuous	Bounds = (2, 8) [mm]
Number of Struts	Vane Assembly	discrete	Options = (8,9,10,11,12) [adim]

There are four quantities of interest for this design:

1. **Weight**: It is the default driver for all aerospace components. Calculated using the CAD volume and the material density.
2. **Stiffness**: The component stiffness is important for the system as the mechanical Whole Engine Model (WEM) uses the stiffness of each component to distribute the external load accurately. The calculation method is a 3D element FEM that is run with unitary loads at the interface locations and the displacement measured at the other interface locations. Ten different measurements of stiffness are provided, depending of the loading application and displacement direction. For Example, FX_UX measures the displacement in the X direction when a force in the X direction has been applied.
3. **Lug stress and failure modes**: Lugs on the TRS are considered part of the aircraft system and therefore subject to the Certification Specification for Large Airplanes CS-25 in Europe (14 CFR Part 25 in the United States). In particular CS 25.301 Limit and ultimate analyses. The well established hand calculation method [7] that has been implemented in Python and is part of the KBE primitive lug.

4. **Containment capacity**: Ability of the outer case of the TRS to contain a rotating turbine blade (and disk) that breaks and impacts the TRS, as per CS-E 810 (CFR §33.94). A simple energy-strain model is used and compared to a reference model.

The EF-M tree created for this industrial application is shown in Figure 7.

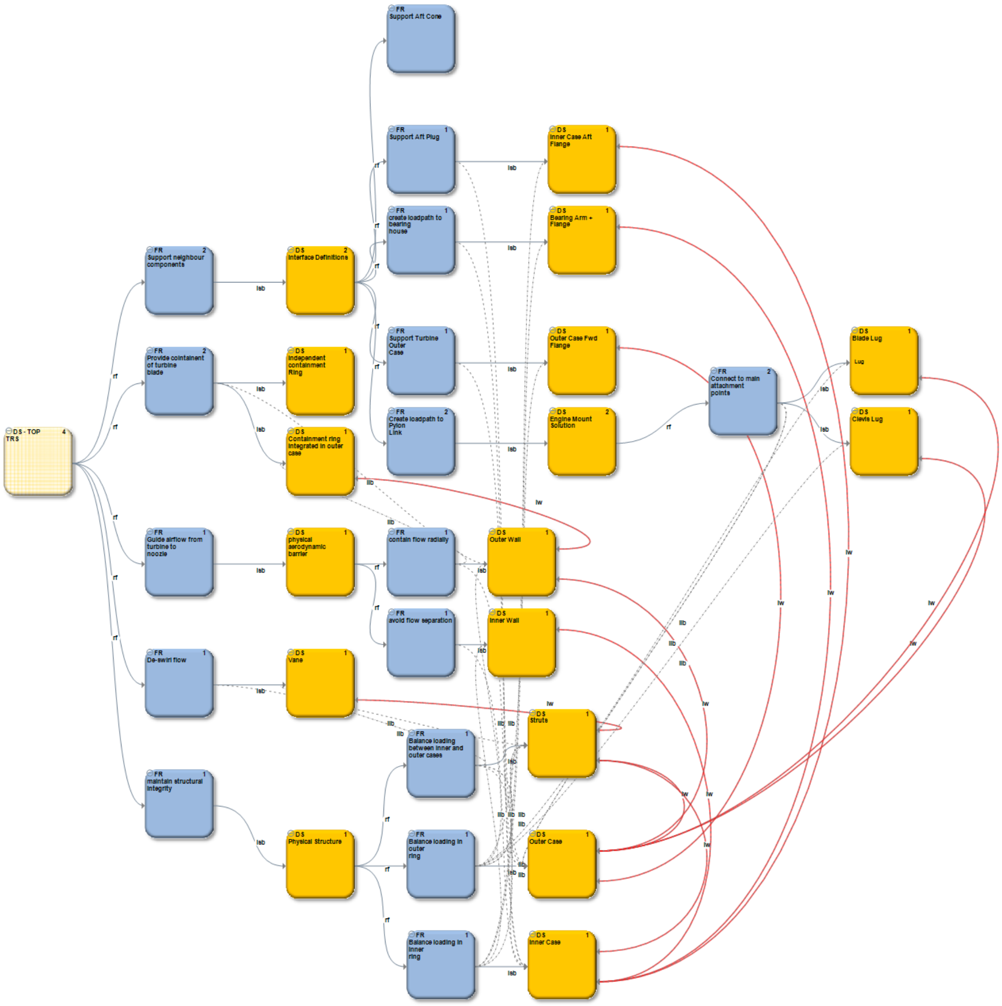


Figure 7: EF-M tree for the TRS industrial application. Screenshot of the CCM tool.

A new library of KBE primitives has been developed to model the geometrical features of the DoE. An example of a TRS use case is shown in Figure 6 and corresponding Finite Element Model (FEM) generated in Figure 8.

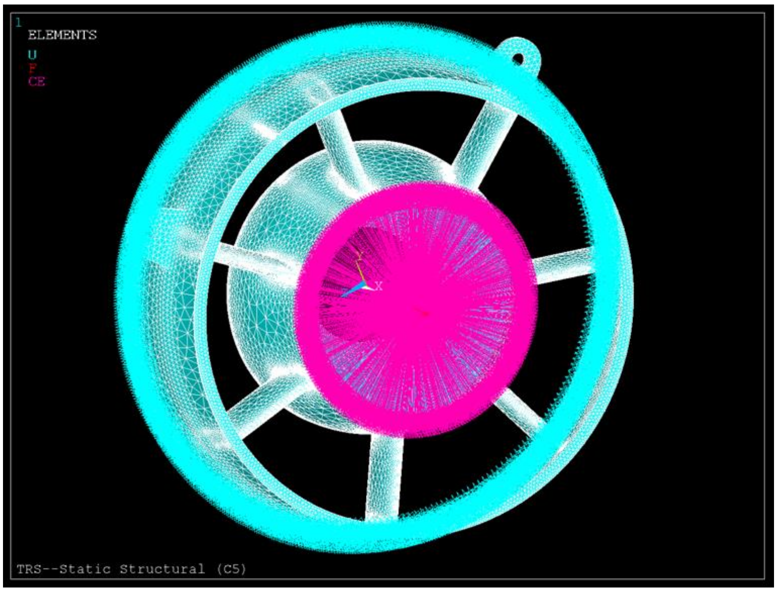


Figure 8: FEM model with loading and boundary condition for a design use case.(view from other side of Figure 6). Screenshot of ANSYS software.

3.4. Results

The methodology has been verified in this industrial use case, ensuring that the Design of Experiments generated is consistent with the configuration stored in the EF-M. In addition, the connection with the Physical KBE model is manually inspected in key design cases to ensure the architectural configurations are updated as expected. Finally, the output results are explored to ensure the design space exploration captures the expected behaviour of the product, like, for example, the stiffness increases with the number of vanes.

-

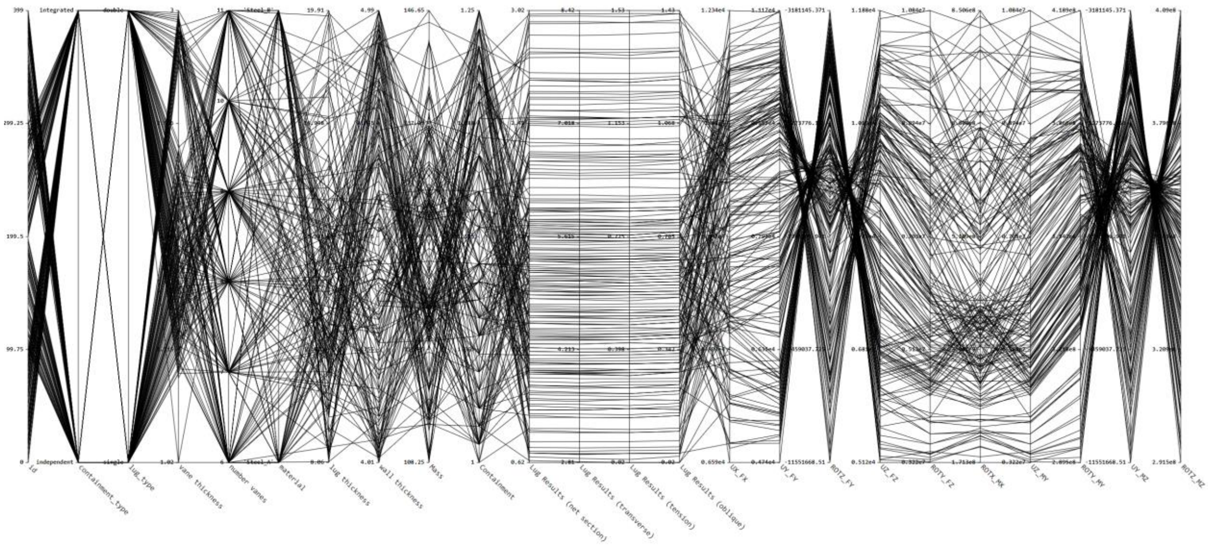


Figure 9: Results of the complete DoE (400 runs) in a Parallel Coordinates Plot.

However, it could benefit from further validation on other applications.

4. Machine learning plug-in based on value impact assessment

This section reports about the software plug-in based on Artificial Intelligence for exploration of the value and cost of different design concepts (part of the software deliverable D5.7 *"Machine learning plug-in based on value impact assessment"*).

4.1. Background

The software Club Design (reported in the deliverable D5.6 *"Set of models for requirements, value and cost"*) has shown how different Design alternatives can be evaluated using a monetary model (Surplus Value, Panarotto et al., 2020).

Club Design consists of a four major functions:

- Capture and define of a specific context for the system in which requirements are generated. It includes initially a set of rank-weighted Needs that have to be satisfied. The VCS is used to define value driven scenarios that is given as input to design studies. this is done in the **"VCS tab" of Club Design**)
- Define Key Engineering Characteristics given a specific Value Creation Strategy (called Value Drivers). They represent proposed directions of investigation since they seem to have a significant influence on the perceived value in a given context. Value Drivers themselves are not attached to a target value or function, but they tend to result in measurable objectives and later, based on these, requirements. Examples of Value Drivers are "Minimum expected life" that impact performance in service, "mass" that impact "take-off weight" or "number of interfaces" that impact how easy a technology/component is to integrated into a system. this is done in the **"VCS" of Club Design**)
- Generate (or import) design alternatives having an impact of the different value drivers (and therefore the value creation strategy). (this is done in the **"design tab" of Club Design**)
- Calculate the quantitative (financial) impact of the design alternatives on the different "Value Creation Strategies". this is done in the **"Surplus Value Simulation" of Club Design**) The assessment is performed using a financial metric called Surplus Value. The Surplus Value Theory provides a simplified equation that is a subset of Net Present Value (NPV) based on several assumptions. NPV is used by economists to describe profit and is a basis for business investment decisions. In accordance with the Surplus Value Theory, the model optimizes the combined profit of the customer, the manufacturer and eventual suppliers. The theory hence strives for the optimization of the combined profit of an imaginary corporation that performs all three roles. The combined Surplus Value is simpler to compute because it is not affected by the actions of competing manufacturers. The Surplus Value model described has been programmed using a Discrete Event Simulation (DES) technique. DES models the operation of a system as a (discrete) sequence of events in time. This means that the whole lifecycle of the system under consideration has been divided into discrete events. This is made possible by the association of the stakeholders needs and expectation to lifecycle "processes" made in the Value Creation Strategy view.

The designs are often created by "brainstorming" potential solutions and based on expert judgements or using simulation tools driven by KBE methodologies. The objective is to find the best design that increases the Surplus Value (calculated for example in k€). However, there may be situations in which the design team wants to extend the dataset created in Club Design:

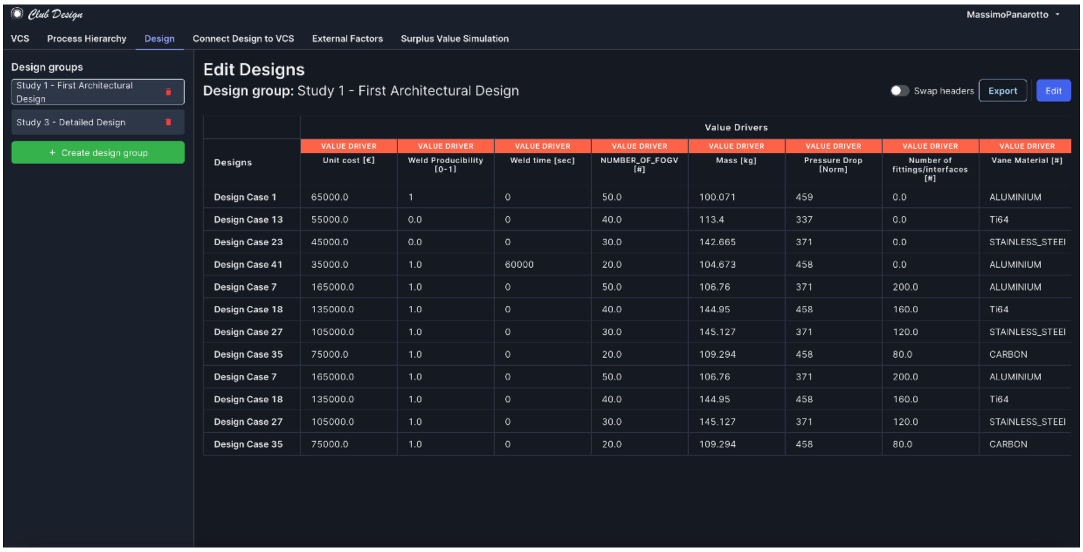
1. When running the Surplus Value simulation takes considerable time
2. When the design team wants to receive a suggestion about what are the driving factors that make the SV to increase (i.e. suggest a new design)

Therefore, there is the potential to use Machine Learning and Artificial Intelligence to

4.2. Methodology

The plug-in is based on a Kriging-based Surrogate model (using the SMT toolbox) on a Club Design exported dataset, It features designs on the rows, and design objectives and SV on the columns which generates new data points via interpolation. The results are visualized in an Interactive enabled interface for Parallel Coordinates plots.

4.3. Industrial application



	Value Drivers							
Designs	VALUE DRIVER Unit cost [€]	VALUE DRIVER Weld Producibility [0-1]	VALUE DRIVER Weld time [sec]	VALUE DRIVER NUMBER_OF_FOGV [#]	VALUE DRIVER Mass [kg]	VALUE DRIVER Pressure Drop [Norm]	VALUE DRIVER Number of fittings/interfaces [#]	VALUE DRIVER Vane Material [#]
Design Case 1	65000.0	1	0	50.0	100.071	459	0.0	ALUMINIUM
Design Case 13	55000.0	0.0	0	40.0	113.4	337	0.0	Ti64
Design Case 23	45000.0	0.0	0	30.0	142.665	371	0.0	STAINLESS_STEEI
Design Case 41	35000.0	1.0	60000	20.0	104.673	458	0.0	ALUMINIUM
Design Case 7	165000.0	1.0	0	50.0	106.76	371	200.0	ALUMINIUM
Design Case 18	135000.0	1.0	0	40.0	144.95	458	160.0	Ti64
Design Case 27	105000.0	1.0	0	30.0	145.127	371	120.0	STAINLESS_STEEI
Design Case 35	75000.0	1.0	0	20.0	109.294	458	80.0	CARBON
Design Case 7	165000.0	1.0	0	50.0	106.76	371	200.0	ALUMINIUM
Design Case 18	135000.0	1.0	0	40.0	144.95	458	160.0	Ti64
Design Case 27	105000.0	1.0	0	30.0	145.127	371	120.0	STAINLESS_STEEI
Design Case 35	75000.0	1.0	0	20.0	109.294	458	80.0	CARBON

Figure 8: 12 Design alternatives defined in the software tool Club Design.

Figure 8 shows 12 design alternatives for a Fan Outlet Guide Vane (coming from the GKN Aero Engines use case). The designs are characterized by very different objectives, representing both functional (e.g. weight, pressure drop) as well as non-functional requirements (e.g., weld accessibility, Number of fittings/interfaces for easier maintenance, material criticality score).

The Surplus Value model implemented in Club Design allows to aggregate these different objectives into a single monetary function, representing the cost and revenue profile over time. Figure 9 shows the SV profiles for the 12 alternatives, showing how Design Case # 35 is the one that has the highest SV (in k€).

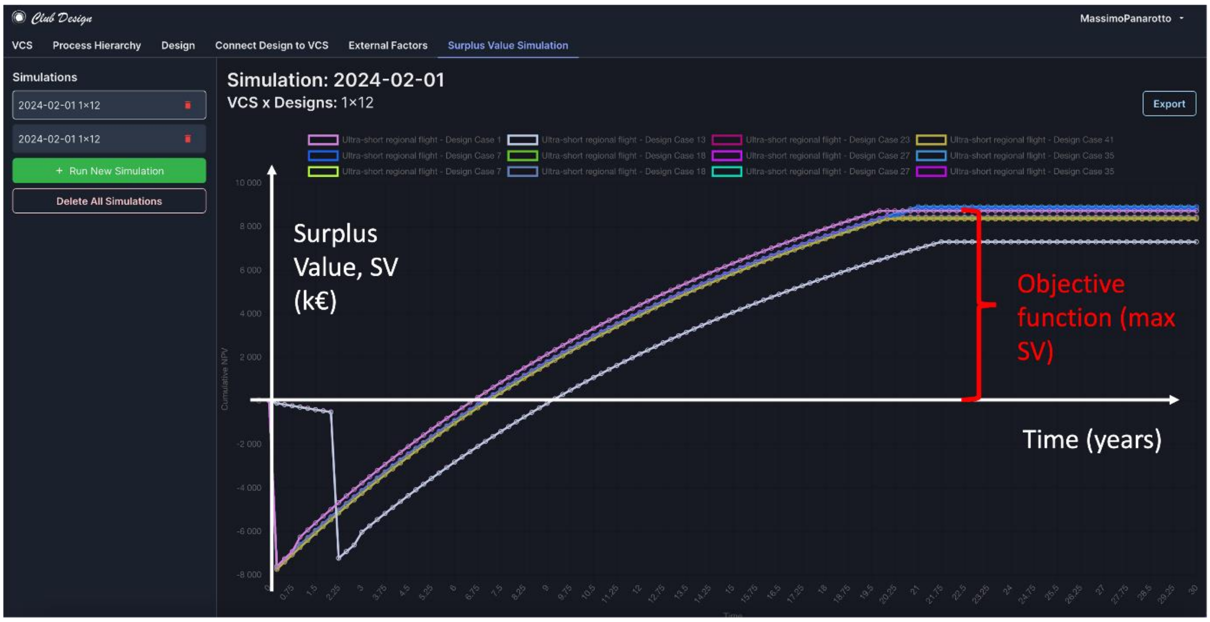


Figure 9: Alternatives evaluated in Club Design using the Surplus Value model. Design Case #35 is the one with highest Surplus Value (in k€). The simulation with only 12 designs took 34 seconds.

The Surplus Value Simulation with only 12 designs took 34 seconds. Therefore, it is appealing to use a Machine Learning algorithm to reduce the computational time. At the same time, the ML algorithm would inform designers about the best design to further increase the SV. This is particularly relevant for objectives that are very different in nature, and for a decision making team that is composed by experts coming from different disciplines

Using the “export button” on the Surplus Value Simulation tab, a new table is created, where the design objectives and the SV for each of the designs can be exported. This is the input for the Kriging based Surrogate model. Figure 10 shows a simplified example of a Kriging-based surrogate model from 5 design alternatives. The model is used to generate a new design and to predict the resulting SV.

.

```
 Training

   Training ...
   Training - done. Time (sec):  0.8160009
Optimal 'vd' values for maximizing SPV: [40837.63     0.50     43.08     0.00     1.01 24754.34     30.30     33.90    100.23    33
7.60   129.64]
Predicted 'spv' value: 8881.788690439374
RMS error 5.940028572016205e-15
Mean Squared Error (MSE) for validation points: 35545.743693667944
R-squared for validation points: 0.9297990369910556
```

Figure 10: Kriging-based surrogate model of 5 design alternatives. The model is used to generate a new design and to predict its SV.

The result shows how new design points can be generated without the SV simulation, reducing the amount of time taken (0.81 seconds compared to 34 seconds). Also, the prediction can be based on an accurate prediction. In this simple example, with only 5 designs, the R-square is almost 0.93 which is considered very accurate for this case.

The results can be visualized using an interactive parallel coordinate plot, where the exploration among the ML “suggested” design. Figure 11 shows the parallel plot comparing the ML predicted design against the other 5 designs contained in the dataset.

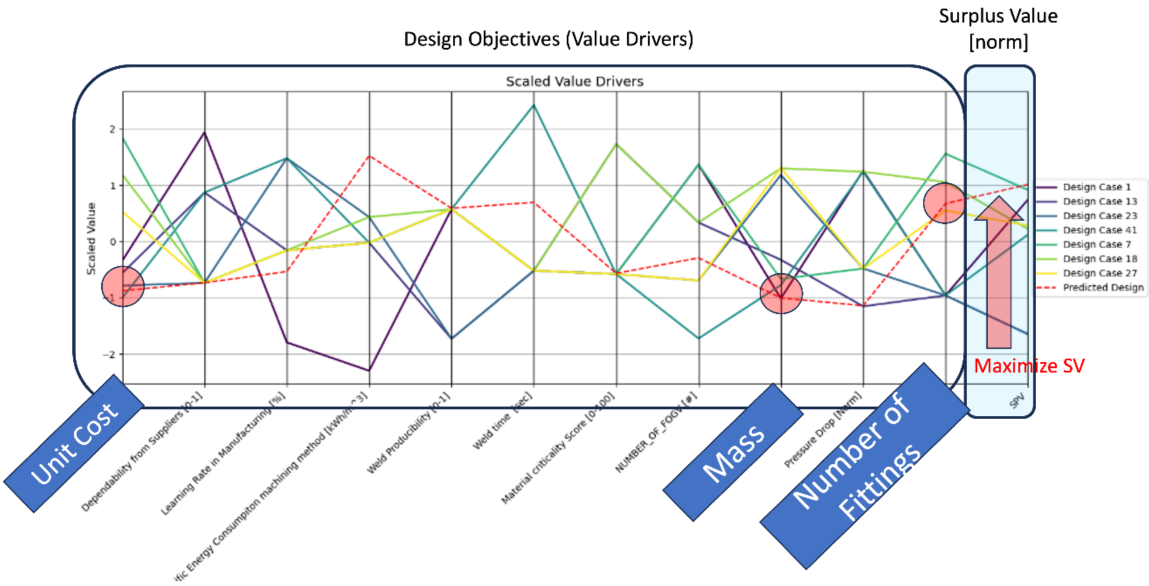


Figure 11: Parallel plot comparing the predicted design against the other 5 designs. The dashed line is the predicted dataset.

Each line in the parallel plot is a different design. The dashed line is the predicted design. All the scales have been normalized.
From the predicted design (dashed line in Figure 11), some considerations can be drawn:

- The predicted design does not feature the lowest possible cost. This means that some investment in the product is justified, and an aggressive cost reduction strategy is not favourable.
- The predicted design features the lowest mass. This confirms mass being one of the main value drivers for the product.
- The predicted design features a high number of fittings (although not the highest). This is where the investment made (reflected in the unit cost) is going to be dedicated to. A higher number of fittings means a higher cost of manufacturing (higher cost of manufacturing the fittings). However, the fittings allow the product to be easily maintained and upgraded over time, which increases the value over time for the product, and hence the Surplus Value. This conclusion can support an expert coming from the organizational function of “maintenance & upgrading” to discuss with the other members of the multi-disciplinary team about the need to have a relatively high number of fittings. This points at the benefit of this type of models to act as boundary objects within a multi-disciplinary design team (Panarotto et al., 2019).

4.4. Discussion and Conclusion

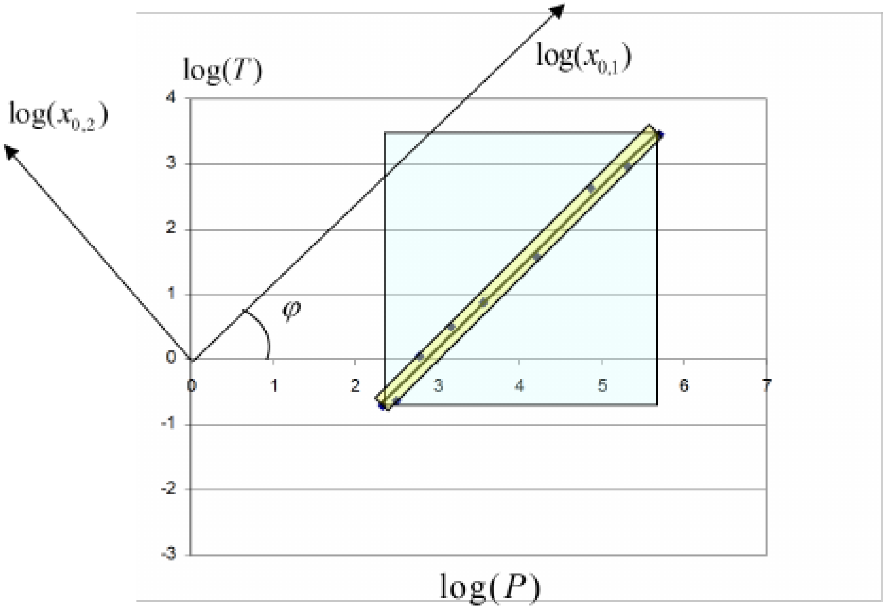
While machine learning has been applied in the context of Engineering Design and Value Assessment (e.g., Piotrowski, 2019), so far it has not been applied on a heterogenous set of design objectives (combining functional and non-functional requirements) aggregated using a single monetary value function. The plug-in made available in Club Design allows to predict new design points, and this can be particularly benefits in cases which:

- When running the Surplus Value simulation takes considerable time (in this case study, the time taken for simulating 12 designs was 34 seconds, the time to generate a new design with ML was 0.86 seconds).
- When the design team wants to receive a suggestion about what are the driving factors that make the SV to increase (i.e. suggest a new design)

5. SVD Modelling and Correction Neural Networks

This section is about the using Principal Component Analysis (or SVD) to create prediction models based on statistical data, such as data sets of a range existing similar products in such a way that the properties of new products can be estimated from a set of requirements on the product attributes.

Multiple regression analysis can be used to create models that often show good agreement around the data sets, used to establish the model. However, when the product attributes are highly correlated, it is an advantage to use Principal Component Analysis (PCA). This is mostly done using the Singular Value Decomposition (SVD) hence these two acronyms are used more or less interchangeably. Using PCA the coordinate system is rotated in such a way that two new parameters, the principal components, become uncorrelated. First, the statistical properties become more sound, and secondly and perhaps more important, the explicit constraint can be set on the parameters that now provide a better fit around the statistical data set, such that the design space gets a shape more consistent with the space spanned by the data set. This is especially useful when used for system optimisation.



Rotating the coordinate system means that the explicit limitation of the variables can have a much tighter fit towards the measurement data. Using \gls{PCA} it is possible also to include the distribution of data in the design space, into the model.

Using Singular Value Decomposition, SVD, introduced in [11], it is possible to do the PCA and create a model that has a few synthetic parameters as inputs and all the attribute of the design as outputs. The output of the model can include both design parameters and functional characteristics. It is then possible to quickly estimate a design from given requirements, by solving the resulting system of equations. Interestingly, it can also be used to estimate performance and other characteristics from limited data. In [12], SVD was used to reduce the number of variables for optimization of an industrial robot.

Another very useful application is for modelling of components and subsystems. In [13] and [14] this was used for models based on statistical data of components and systems, such as aircraft and aero engines. In this way models with high accuracy can be produced that can relate e.g., engine dimensions (such as diameter, length, etc.), to attributes such as weight, bypass ratio, trust and specific fuel consumption. It is also possible to include year of introduction as one variable and in this way also have a mechanism for technology evolution over time, although looking at times beyond the present will take us outside the dataset that the model is based on.

In a design situation the SVD model can be used in the role of a surrogate model. Instead of making a parametric design, of a higher fidelity, which is optimized for each situation, It is possible to optimize for a few situations and then build an SVD-model based on these. In this way a meta model with high accuracy can be obtained. Ones an optimal solution has been reached it can be recalculated and be added to the set of data points the SVD model is based on.

Finally, SVD analysis can be used to evaluate a given parametrization by studying the correlation with the ideal SVD parameter set. This is useful since it sometimes is an advantage to have a parametrization that have a clearer interpretation than the synthetic SVD parameter set can provide. This was shown in [14]. Interestingly, it is also possible to derive the number of driving requirement in a design by studying a number of instances of a particular kind of product.

5.1. Singular Value Decomposition

Singular Value decomposition, SVD, is a technique that is related to PCA. The result is essentially the same, but it involves an elegant mathematical method to obtain a model that is aligned with the main axis of the data set. Consider the data set **X** which is a matrix. Then there exists a decomposition of the form:

$$\mathbf{X} = \mathbf{U} \times \mathbf{W} \times \mathbf{V}^{\mathbf{T}}$$

where W is diagonal. This is the Singular Value Decomposition matrix. This can look like this:

$$\begin{pmatrix} x_{11} & x_{12} & x_{11} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \\ x_{41} & x_{42} & x_{43} \end{pmatrix} =$$

$$= \begin{pmatrix} u_{11} & u_{12} & u_{11} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \\ u_{41} & u_{42} & u_{43} \end{pmatrix} \times \begin{pmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & w_3 \end{pmatrix} \times \begin{pmatrix} v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \\ v_{31} & v_{32} & v_{33} \end{pmatrix}^{\mathbf{T}}$$

The consequence of this operation is that if each row in the **X** and **U**- matrix represents a data set of the entity that should be modelled. Any point in **U** is mapped onto **X** trough the matrix product. Usually, the resulting matrices are arranged in such a way the diagonal elements of the **W**-matrix are in descending order. Hence the influence of the **U** variables is in descending order in each row, which means that the last ones can be omitted in order to get a simpler model without too much loss in accuracy. However, for this to be valid the dataset should first be centred around the mean value. This can be done by subtracting the average of each column in the **X**-vector from the values of each column. An interesting property of the **U**-matrix is then that the sum of the variance of each column is one. That is:

$$\sum_{i=1}^{n} u_{ij}^2 = 1$$

This means that all columns (that is parameters) have the same deviation. The W matrix is then a weight matrix with only diagonal elements, and $\mathbf{V}^{\mathbf{T}}$ is a matrix that rotates the coordinate system from the main axis into **X**.
The meaning of the matrices is the following. If the **U** -matrix is a dataset that is uniform in all directions, **W** -matrix stretches this in the different dimensions as indicated by the diagonal elements. Finally, the $\mathbf{V}^{\mathbf{T}}$-matrix rotates the dataset to produce the final mapping into the original coordinates as depicted below

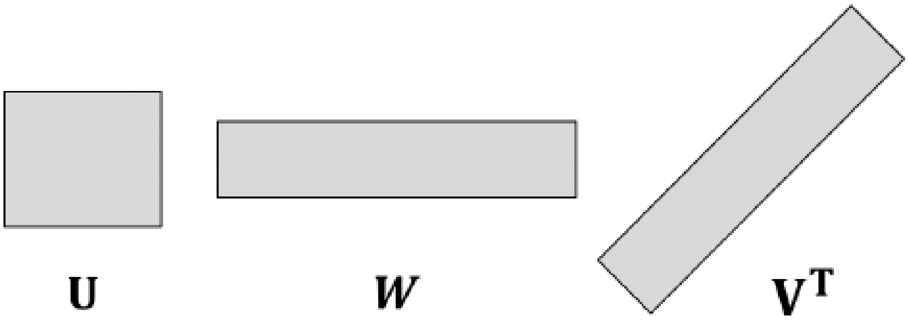


Fig.6.1. The meaning of the matrix operations in SVD.

Using a model based on SVD to estimate parameters and properties, the following equation is used:

$$\mathbf{x}^{\mathbf{T}} = \mathbf{s}^{\mathbf{T}} \times \mathbf{W} \times \mathbf{V}^{\mathbf{T}}$$

Here, the **x** and **s** are vectors. **s** is the input vector with SVD-parameters that are orthogonal, and **x** is the estimated values of parameters and properties. Introducing **K** is the loading matrix defined as:

$$\mathbf{K}^{\mathbf{T}} = (\mathbf{W} \times (\mathbf{V}^{\mathbf{T}})$$

we can now write

$$\mathbf{x} = \mathbf{K} \times \mathbf{s}$$

We have now an expression with a reduced vector s as input and all the product attributes in x, both design parameters $\mathbf{x_D}$ and functional characteristics (or quantities of interests) **y**. Note that since **s** have no physical meaning it has to be found through some equation solving or optimization process. The advantage is, however, that any variables in design parameters and y can be set and the model will then give the other variables.

Example: Transport Aircraft

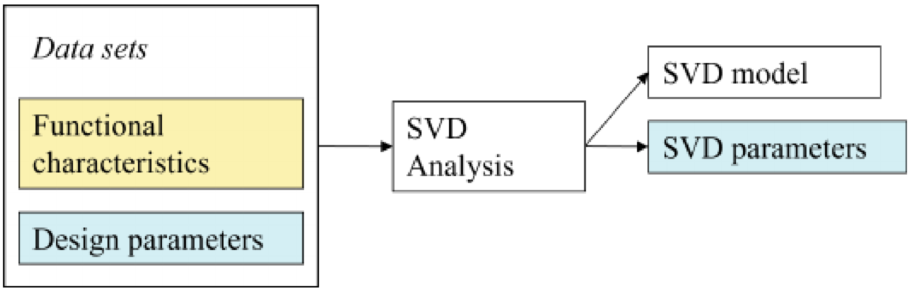


Fig 6.2. Generation of SVD parameter set and model, from sets of data.

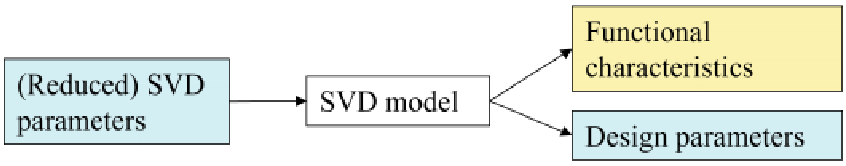


Fig 6.3.. Using a SVD model to predict data from a a reduced set of SVD parameters. This requires finding the SVD variables that are fulfilling the functional requirements. This will then also give the design parameter

5.2. Correction Neural Network

The SVD model is always a linear approximation based on the underlying data set, even though it can be pre- and processed to introduce some non-linearity. Typically, the log transformation where the data is transformed so that the SVD operates ton the lo of the values rather than the values themselves. However, using a Correction Neural Network, different kind of shapes can automatically be adjusted to the dataset. An advantage with combining the NN with the SVD is that it be used on correct already rather well fitted model, so the problem is well conditioned.

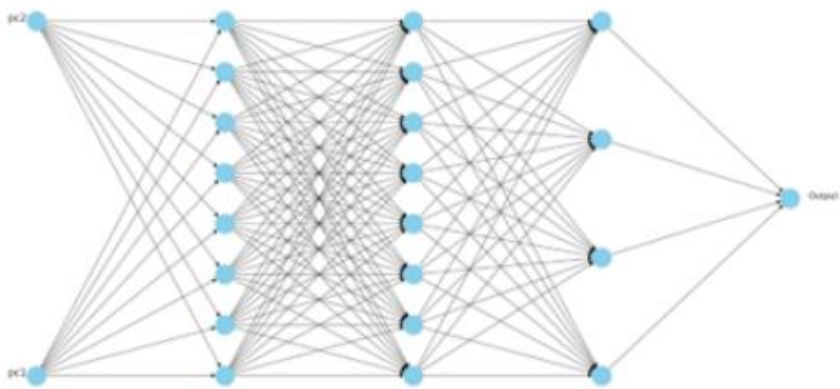


Figure 10 Neural Network with one input layer, three hidden layers and one output layer.

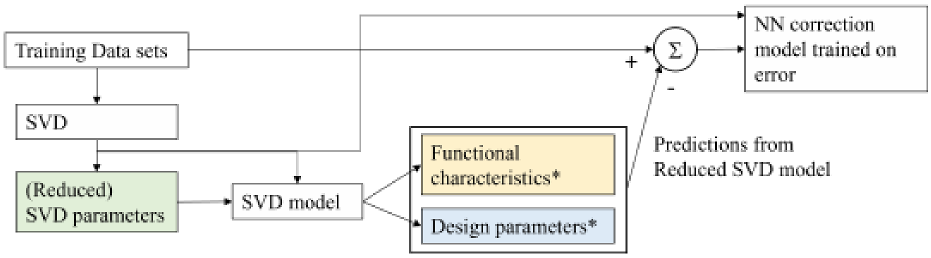


Fig 6.4.. Training of a correction network. The correction network can be trained to remove the error for the predictions of the training data.

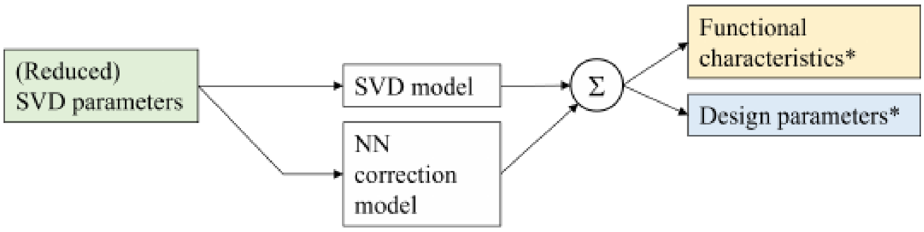


Fig 6.5. The SVD model with the correction model, making it possible to have accurate predictions with few SVD variables.

5.3. Example: Modelling a dataset of Aero engines

As an example a dataset of 158 civil aero engines was used. It includes ten key data for each engine.
It is a very wide range of engines from the smallest with a thrust of 267 N up to 435kN, i.e. three orders of magnitude.

Name	Bpr+1	Ts [kn]	Tc [kn]	SfcS [1/hr]	SfcC [1/hr]	M	[h]	m [kg]	d [m]	l [m]
TF1000	9.14	4.44822	1.201019	0.4	0.64	0.6	9144	129.5588	0.5842	1.4224
TF1200	8.59	5.729307	1.779288	0.41	0.66	0.6	9144	129.5588	0.5842	1.4224
TF1400	8.59	6.227508	1.779288	0.41	0.66	0.6	9144	136.3777	0.5842	1.4224
TF1500	8.59	6.67233	1.765943	0.41	0.68	0.6	9144	136.3777	0.5842	1.4224
TJ60	1.5	0.266893	0.133447	1.24	1.5	0.9	6096	4.091331	0.127	0.3429
TJ75	1.44	0.333617	0.213515	1.24	1.5	0.9	6096	4.545924	0.1778	0.3429
TJ80	1.5	0.320272	0.213515	1.24	1.5	0.9	6096	4.091331	0.127	0.3429
ALF502L	6.2	33.36165	9.563673	0.428	0.73	0.75	9144	595.9706	1.27	1.66624
ALF502R-5	6.6	31.00409	10.0085	0.408	0.72	0.7	7620	607.3354	1.27	1.62052
CFE738-1-1B	6.3	26.32457	6.512194	0.369	0.645	0.8	12192	602.3349	1.2192	2.5146
CFM56-2C2	8.47	106.7573	22.15214	0.36	0.671	0.8	10668	1723.814	1.8288	2.43078
CFM56-3B1	7	88.9644	20.68422	0.38	0.667	0.8	10668	1950.201	1.6002	2.36474
CFM56-3B2	6.9	97.86084	22.41903	0.39	0.667	0.8	10668	1955.202	1.6002	2.36474
CFM56-3C1	7	104.5332	23.88694	0.33	0.667	0.8	10668	1955.202	1.6002	2.36474
CFM56-5A1	7	111.2055	22.2411	0.331	0.596	0.8	10668	2270.689	1.8288	2.51206
CFM56-5A3	7	117.8778	22.2411	0.331	0.596	0.8	10668	2270.689	1.8288	2.51206

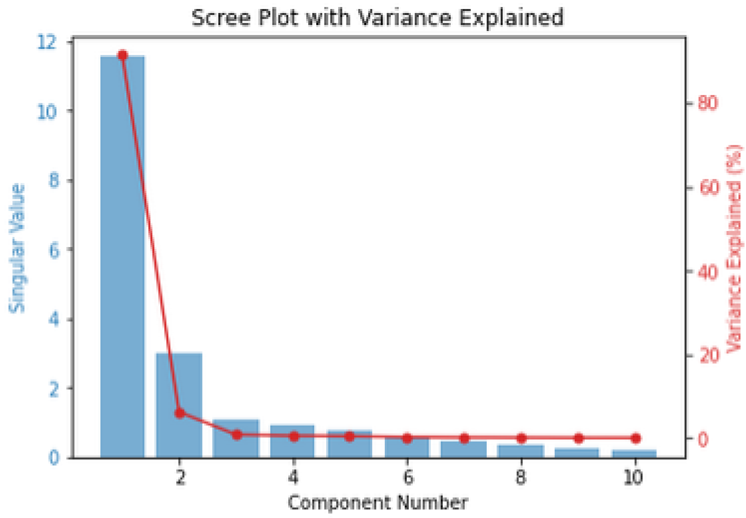


Fig 6.6. Singular values and explained variance. Showing the tapering of the importance of the SVD-components.

Fig 6.7. The loading matrix showing the dominance of the PC variables (SVD) with low numbers.

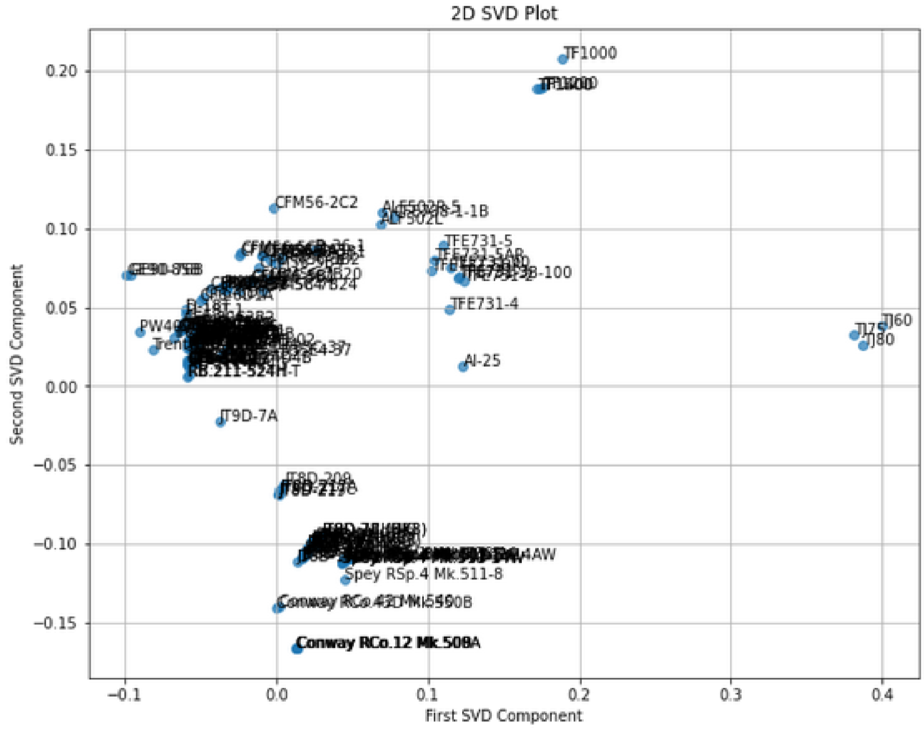


Fig 6.8. The distribution of data as function of the first two principal components (SVD variables).

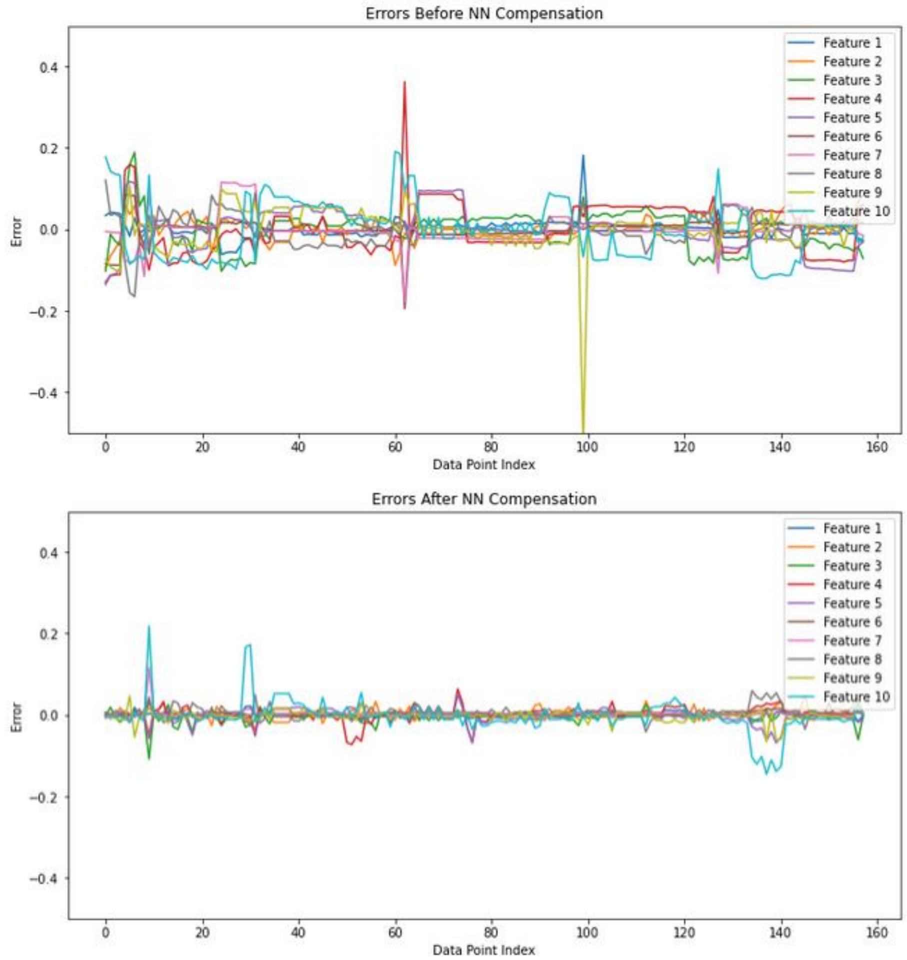


Fig6.9. The top graphs show the errors between the original and the reconstructed data set using two SVD variables. The error is in the data before retransforming from log data and cantering. The bottom shows the error after correction network has been added. The data set shows both the training set and the validation set.

This means that a model with only two parameters that can predict 10 attributes, can be used to represent all 158 aero engines in the data set with an accuracy that is likely more than enough in a conceptual design.

5.4. Summary

The process to establish a prediction model based on SVD and NN can be summarized in the figure below. The SVD model can be used as it is but there is also the possibility to combine it with a NN correction network. Ones established the use of the models is similar. With a minimum of input variables, the whole set of product attributes can be predicted.

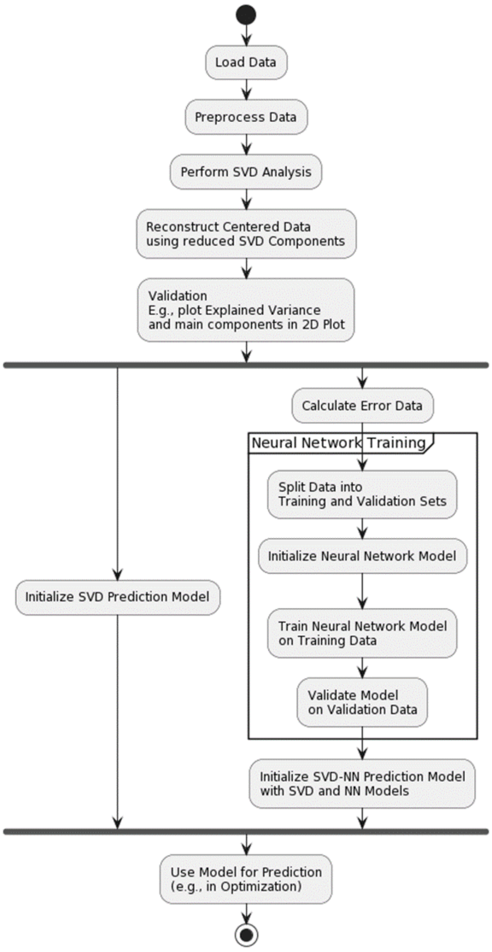


Fig. Process to establish prediction model based on SVD and/or neural network.

6. Generative AI for design space exploration activities

An early exploration of the capabilities of Large Language Models was performed by Chalmers University of technology and Linköping University. A summary of those explorations will be presented on the 18th International Design Conference, May 20th-23rd 2024 in Dubrovnik, Croatia. The Paper: LARGE LANGUAGE MODELS IN COMPLEX SYSTEM DESIGN will be part of the conference proceedings. The paper explores the reported applications of Large Language Models LLMs on Engineering Design tasks. The paper provides two examples where they can support the design process and finally the opportunities and challenges of this technology are highlighted.

6.1. Exploration result summary

Two use cases were explored, one at the system level and one at the component level.
On the system architecture use case, a hydraulic system UML diagram was performed with the support of the LLM. See Figure 11 for the generated output.

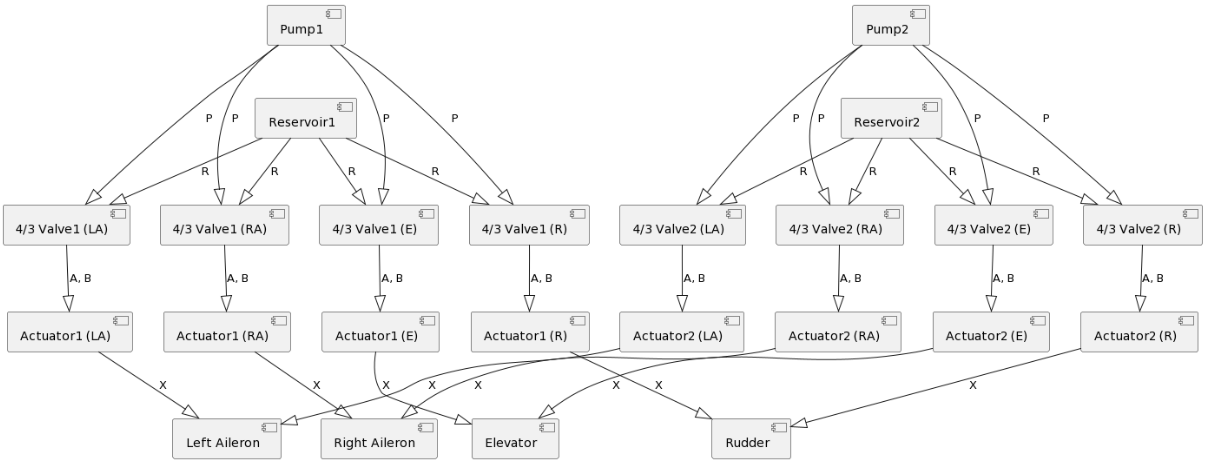


Figure 11: UML diagram (in PlantUMLTM) of an aircraft actuation system with four functions and two circuits, each with a pump and reservoir connected to a valve and actuator for each function.

For the component use case, the LLM support the editing of a CAD model based on the a description of the sketch to update. See Figure 12 and Figure 13 for the process description.

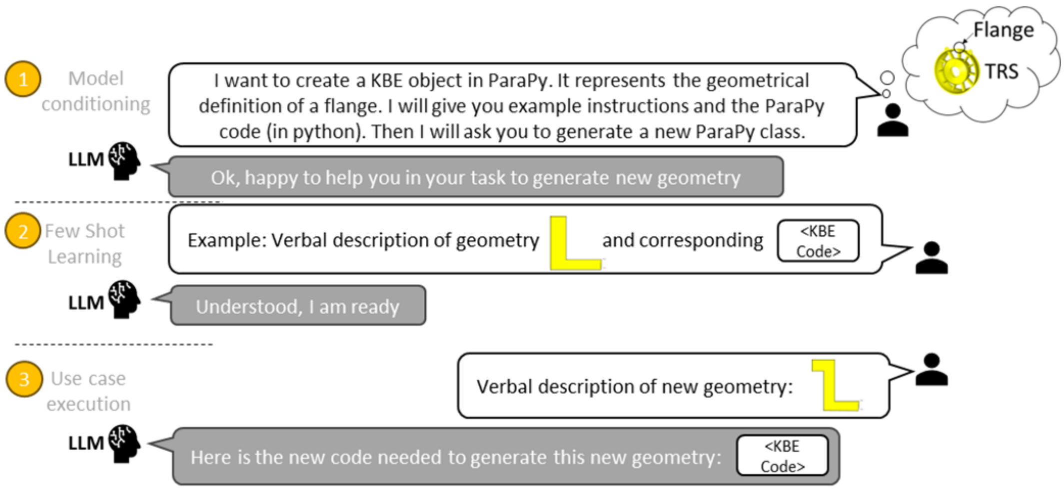


Figure 12: Diagram representing the design process steps. The three steps on the left represent the conversational support of LLMs

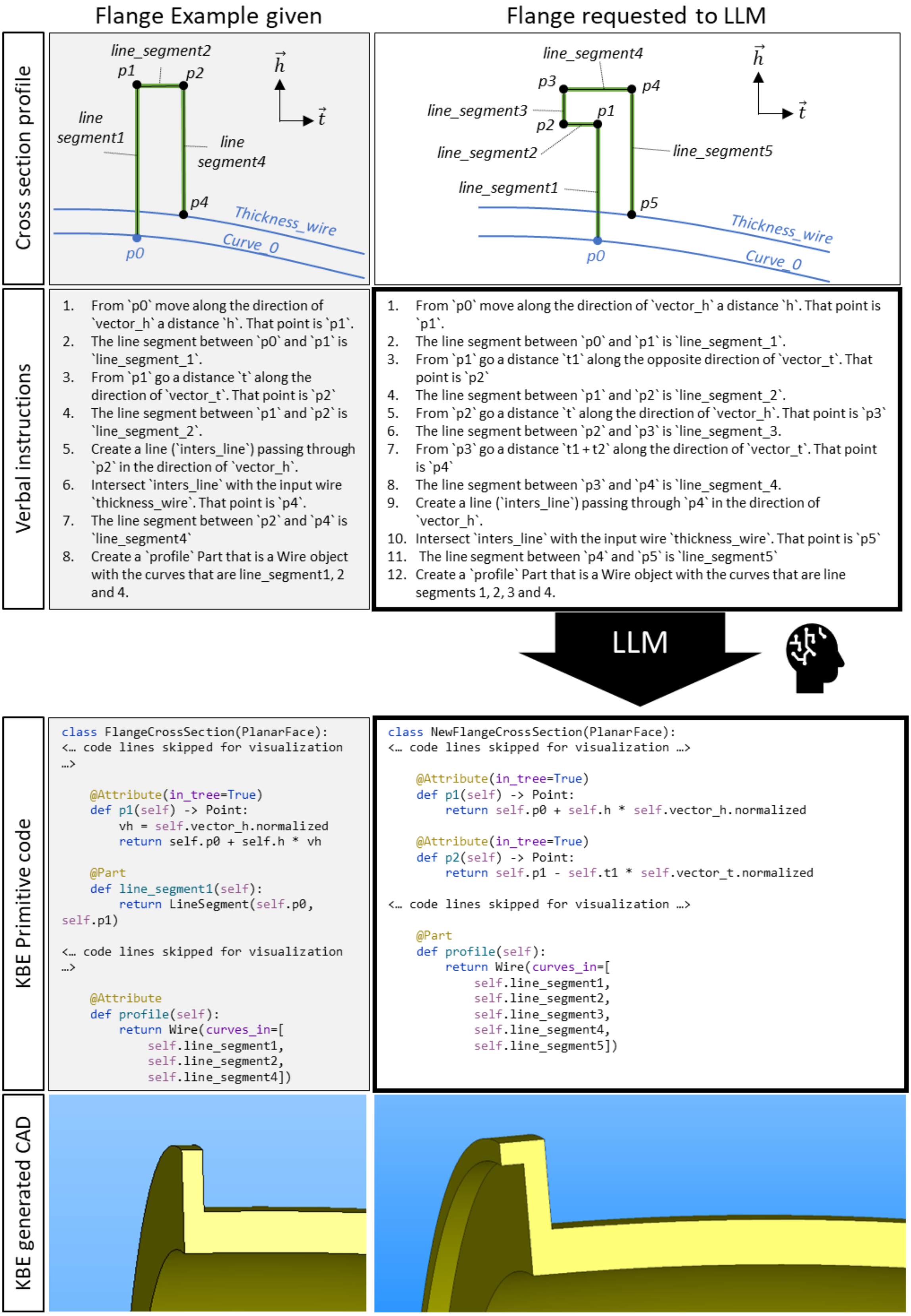


Figure 13: Column comparison between the pre- and post-design change geometry expressed visually and verbally. Prompt and response of the LLM for step 3 are highlighted in the right.

Based on the geometrical description and a one shot example, the model can update the python code in the KBE application to represent the new geometry. When the code generated by the LLM is executed, the CAD correctly renders the new geometry, and the rest of the KBE application can be used to analyze the new configuration, as shown in

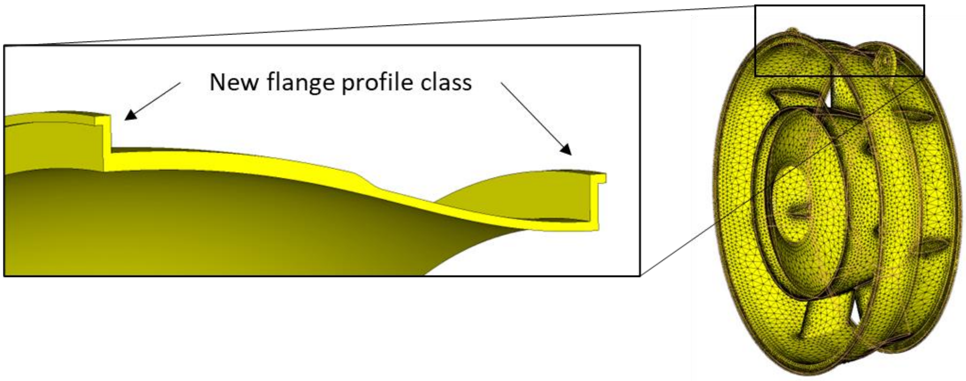


Figure 14: Cross section of the TRS showing the new geometry of the flange. Right: Complete TRS automatically meshed my using the same KBE functions.

6.2. Conclusions

The paper shows the potential to use generative AI technology on Engineering Design, where results are deterministic. The paper highlights opportunities, such as the reduced amount of training data (an example) required to train the model. Some of the challenges identified are the need for careful inspection of results and the need to improve the reliability of the LLM based applications.

On the topic of reliability of LLM based systems, Chalmers University of Technology is further working on exploring the state of the art multi-agent framework such as [15] applied to an engineering design application. The work was initiated in the DEFAINE project, and it is continued after the project ends. It is expected to be presented in the NordDesign conference in August 2024.

7. Surrogate modelling

In the complex and computationally demanding field of aircraft design, surrogate modelling has emerged as a powerful tool to accelerate Design Space Exploration (DSE) and optimization processes. While computational advancements have aided design engineers in early-stage decision-making, the multidisciplinary nature of aircraft design often leads to extended project timelines, with simulations alone consuming weeks or even months. This challenge is further exacerbated by the multi-faceted nature of DSE, which often involves hundreds or thousands of simulations. Surrogate modelling addresses this bottleneck by constructing approximation models, also known as surrogates, of the underlying system. These surrogates, trained on a dataset of inputs and corresponding outputs, can effectively mimic the behaviour of the original system, enabling rapid and cost-effective exploration of the design space.
The effectiveness of surrogate modelling within aerospace has been demonstrated in several cases, as it reduces computational expenses associated with DSE. However, it is crucial to critically evaluate the trustworthiness of surrogate predictions to ensure informed decision-making.
This chapter presents an in-depth investigation into the performance of various surrogate models on a range of data sets. To objectively assess their capabilities, the predicted responses of surrogate models have been compared against the computed results, employing established quality metrics.

7.1. Background

The findings in study 1 and 2 are based on two research efforts: Elias Nilsson's MSc. thesis at Chalmers University, which focused on establishing a benchmark framework and evaluation strategy for surrogate modelling algorithms, utilizing synthetic data for controlled experimentation; and Petter Andersson's research, which utilized data from a Design Space Exploration study conducted within the DEFAINE project, providing insights into the performance of surrogate models on real-world aircraft design data. This evaluation provides valuable insights into the strengths and limitations of various surrogate modelling techniques. The results highlight the potential of surrogate modelling to streamline DSE and optimization processes while maintaining the accuracy and reliability essential for making informed design decisions. In addition, an industrial application from GKN Fokker Aerospace BV that consists of an aileron, which can be varied in dimensions, structural layout, materials and loading provides an example where surrogate modelling is used for optimisation.

There are many models to choose from like kriging and random forests, which have their respective positives and negatives. In this study a number of surrogate models are tested to better understand when a particular surrogate model is feasible or not. The benchmark study follows an experimental approach where different datasets are defined and described as mathematical formulas. The characteristic of each dataset is described in terms of Linearity, Polynomial, continuity, discrete, mixed etc. A typical data set can have both numeric and categorical parameters. The datasets are used to test a number of response surface algorithms. The result is presented in a tabular format where the problem datasets are described per row and the algorithms are presented in the columns.
Numeric data can be integers such as the number of vanes or real numbers such as lengths or thickness. There are also have categorical data like the type of material that is used – or a binary value that indicates if a feature is included or not. A typical data set can have both numeric and categorical parameters. Some models are purely numerical and have difficulties

handling categorical data out of the box and therefore have to use pre-processing methods like one-hot encoding.

7.2. Methodologies and Experimental setup.

We are using Surrogate Modeling Toolbox [6] , Scikit-learn [7] and GKN Aerospace in-house data. The Surrogate Modeling Toolbox (SMT) is an open-source Python package consisting of libraries of surrogate modelling methods (e.g., radial basis functions, kriging), sampling methods, and benchmarking problems. Scikit-learn is open source machine learning in python. A set of tools for predictive data analysis built on NumPy, SciPy, and matplotlib.
Bellows follows a short description of the surrogate models used in this study.

7.2.1. Kriging

Kriging is an interpolating model that is a linear combination of a known function which is added to a realization of a stochastic process. The algorithm used in this study is found in the SMT Toolbox.

7.2.2. Random Forest

The Random Forest algorithm is an averaging algorithm based on randomized decision trees. Here, a diverse set of classifiers is created by introducing randomness in the classifier construction. The prediction of the ensemble is given as the averaged prediction of the individual classifiers. The algorithm used in this study is found in the Scikit-learn library.

7.2.3. Multi-layer Perceptron

Multi-layer Perceptron is a supervised neural network learning algorithm that learns a function by training on a dataset, where is the number of dimensions for input and is the number of dimensions for output. The algorithm used in this study is found in the Scikit-learn library.

7.2.4. Radial basis functions

The radial basis function (RBF) algorithm represents the interpolating function as a linear combination of basis functions, one for each training point. The algorithm used in this study is found in the SMT Toolbox.

7.2.5. Nearest Neighbors

Neighbors-based regression is a supervised learning method that can be used in cases where the data labels are continuous rather than discrete variables. The label assigned to a query point is computed based on the mean of the labels of its nearest neighbours. The algorithm used in this study is the K Neighbors Regressor based on the k nearest neighbors of each query point, where k is an integer value specified by the user and can be found in the Scikit-learn library.

7.2.6. Support Vector Regression

Is a subset of the Support Vector Machines supervised learning methods for regression. The implementation is based on libsvm. The fit time complexity is more than quadratic with the number of samples which makes it hard to scale to datasets with more than a couple of 10000 samples. The algorithm used in this study is found in the Scikit-learn library.

7.2.7. One-hot encoding

One-hot encoding is a method used to represent categorical data in a numerical form that a computer can process. It is often used as a pre-processing step in machine learning models. In one-hot encoding, each category is represented as a binary vector with a "1" in the position that corresponds to the category and "0" in all other positions. One-hot encoding allows us to use categorical data as input to machine learning algorithms, which typically require numerical input.

7.3. Evaluation metrics

The following evaluation metrics from the scikit learn library has been used to evaluate the performance of the different algorithms.

7.3.1. Cross validation

A common way to validate surrogate models is to split the data set in a training set and a set that is used for validating the performance. One way is to split the data % wise, often 20% and test once. To get a more comprehensive validation of the complete data set this can be repeated by dividing the set in a new set of training and testing points, until you have used all points as test data. In this work the cross validation is by dividing the dataset in 5 folds and repeat 10 times.

7.3.2. $R^{\wedge 2}$ (coefficient of determination) regression score function.

Best possible score is 1.0 and 0 is bad. It can be negative (because the model can be arbitrarily worse). In the general case when the true y is non-constant, a constant model that always predicts the average y disregarding the input features would get a score of 0.0.

In the particular case when y_true is constant, the score is not finite: it is either NaN (perfect predictions) or -Inf (imperfect predictions). To prevent such non-finite numbers to pollute higher-level experiments such as a grid search cross-validation, by default these cases are replaced with 1.0 (perfect predictions) or 0.0 (imperfect predictions) respectively. You can set force_finite to False to prevent this fix from happening.

7.3.3. Symmetric Mean Absolute Percentage Error (SMAPE)

For the first study the Symmetric mean absolute percentage error was used to measure the accuracy of the surrogate models performance for different synthetic data sets.
SMAPE is defined as follows at Wikipedia.

$$\text{SMAPE} = \frac{100}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{|A_t| + |F_t|}$$

Where At is the actual value and Ft is the forecast value. Here the version providing a measurement between 0 and 100% is used.

7.3.4. Mean Absolute Percentage Error (MAPE)

The mean absolute percentage error (MAPE), is an evaluation metric for regression problems in the scikit learn library [7]. The idea of this metric is to be sensitive to relative errors. It is for example not changed by a global scaling of the target variable.

7.3.5. Explained variance score

Explained variance score is an evaluation metric for regression problems in the scikit learn library. The best possible score is 1.0, lower values are worse. The difference between the explained variance score and the R^2 score is that the explained variance score does not account for systematic offset in the prediction. For this reason, the R^2 score, the coefficient of determination should be preferred in general.

7.4. Result of the benchmark analysis

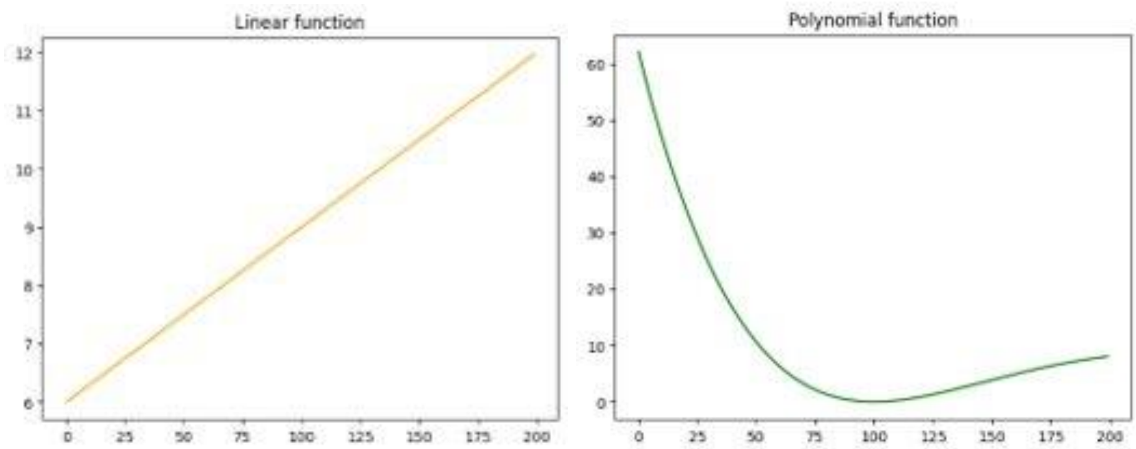
Two studies have been conducted to better understand how well different surrogate models perform on different data sets. In this chapter the analysis results from the study are presented in tables with the outcome from evaluation of a number of surrogate models.

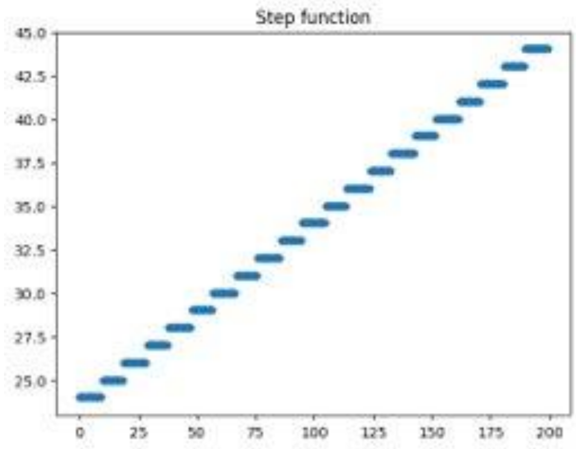
7.5. Study 1, benchmark of synthetic data.

From experience of typical engineering DSE studies, synthetic problems are defined as mathematical functions that mimic similar behaviour in order to provide different challenges for the surrogate models. This data can be used to try available response surfaces to see how they perform. Data are described by mathematical functions, which formulates the characteristics of the data and enables the creation of large data sets. Different response surfaces are used on the data sets to see how they perform.

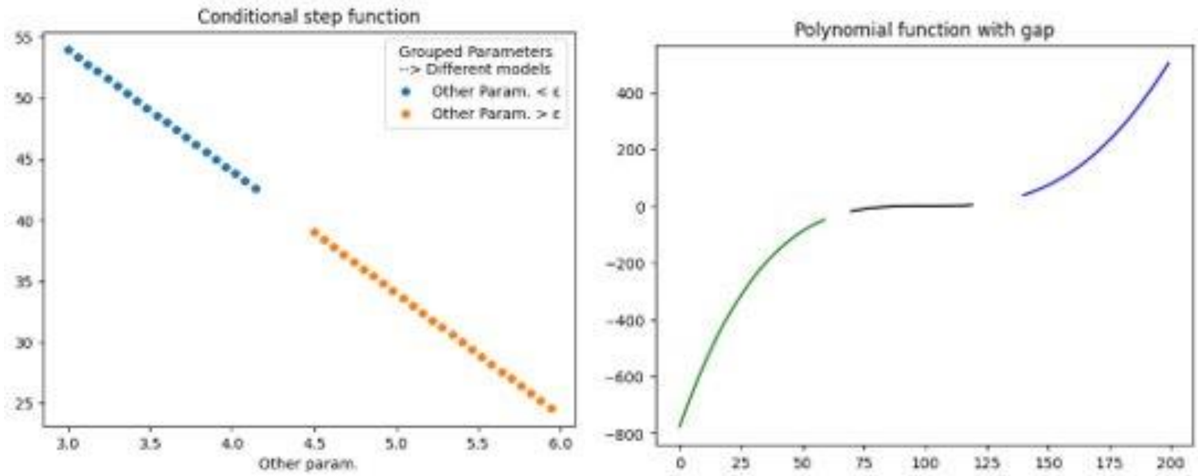
7.5.1. Distribution of numeric data

Parameters may have simple distribution functions while others can be more complicated.
Here we have three parameters with different distribution functions that are common in our data





The functions can also be more complex and conditionally dependent on other parameters. For example - the number of vanes on a model can have different sets of possible values depending on a related thickness parameter.



The models performances is measured on 50 linearly sampled data points using leave-one-out cross validation with 30 iterations for predicting the left out data point. Categorical features, whenever included, are subjected to one-hot encoding, a common method of representing categorical features in a numerical form that models can handle.

The result from the benchmark is presented in the Appendix in a tabular format, providing the mathematical formula for the problem together with boundary conditions. The table shows the evaluation of different surrogate models perform with respect to the *Symmetric Mean Absolute Percentage Error* and the $R^{\wedge 2}$ for each problem.

7.5.2. Conclusion

The results are divided into two sections, the first dealing with "*Only numeric parameters"* and the second dealing with "*Numeric and categorical data".* The details are explained before each subsection. The problems are deliberately designed to capture specific phenomena's or challenges like undefined areas or gaps in an otherwise continues function. Some problems were defined as steps and mixtures of continues and categorical input such a polygonal function and "material".

An important detail is that due to stochasticity, model performance is subject to some variation – thus, the exact numbers in the result tables can vary between iterations. It should also be noted that further adjustments of the parameters for the algorithms tested could improve the performance of the algorithms, hence this study should only serve as a guideline when comparing different response surface models. With respect to the included algorithms and the problems presented in the study, Kriging is performing best for almost all circumstances, including a mixture of continues, none continues and categorical data. The only situation where Kriging did not perform well was with enormous exponential responses >10^10. Thus, if the response feature has these magnitudes, an idea could be to rescale the input features. There are several interesting ideas to dive into and improvements to be done. E.g. non-uniform sampling of the input data. As of now, data is sampled uniformly, but it might be interesting to sample from a skewed distribution to see how the models perform. Improved analytics and automatic visualization; perhaps given some inputs. This is something that could be improved to better understand the qualities of different models. Iterating models via hyperparameter tuning, e.g., RandomizedSearchCV and providing parameters to tune. As of now, the model uses the default parameters.

7.6. Study 2, benchmark of GKN Aerospace engines data

Results from a DSE study performed in the DEFAINE project. The DSE support the development of a component in an Electric Fan Thruster, a novel solution for more sustainable aviation [8]. For the Fan Outlet Guide Vane (FOGV) assembly, the design objective is to provide a lightweight, high performance and cost efficient solution. This study builds on the work reported in the ICAS conference in Stockholm, Sweden 2022 [9] and CEAS conference in Lausanne, Switzerland 2023 [10].

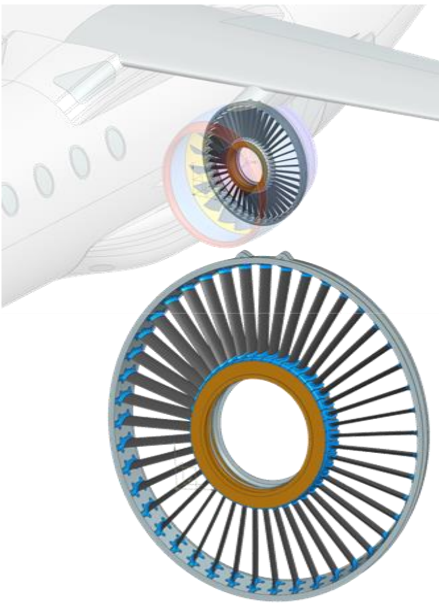
Parameters used in this DSE study are the same as reported in Lausanne although somewhat other ranges.

Name	Type	Kind	Range
Fogv hub aft wall thk	Real	Continuous	1.5:4
Fogv hub attachemnt delta	Real	Continuous	-4:4
Fogv hub fwd wall thk	Real	Continuous	1.5:4
Fogv hub ic thk	Real	Continuous	1.5:4
Fogv hub include stiffner rib	Boolean	Nominal discrete	No;Yes
Fogv hub oc thk	Real	Continuous	1.5:4
Fogv mnt lug ang pos	Real	Continuous	5:10
Fogv oc aft stiff rib height	Real	Continuous	04:30

Name	Type	Kind	Range
Fogv oc attachemnt delta	Real	Continuous	-4:4
FOGV oc fwd stiff rib height	Real	Continuous	04:29.8
Fogv oc thickness	Real	Continuous	1.5:5
Fogv thrust lug angular pos	Real	Continuous	15:35
Material	String	Nominal discrete	Alu, Ti64,Steel
Fogv vane t max	Real	Continuous	60:100
Vane thickness	Real	Continuous	1.5:5
Number of fogv	Integer	Discrete by value	20:50
Use vane core	Boolean	Nominal discrete	No;Yes

Parameter: Number of FOGV:s

The FOGV assembly described here is located at the rear of the rotating fan. The rotating fan does work on the flow by turning it, and the FOGVs subsequently align the flow in the axial direction. The FOGV assembly also provides a load path from the mount lugs to the core of the engine. The number of vanes is a design variable which can be varied, which has an impact on aerodynamic, aeroacoustic and structural aspects of the fan. To keep the aerodynamic impact low while varying the vane count, a parameter called *solidity* can be kept constant. Solidity is defined as the chord (distance from leading edge to trailing edge) divided by the circumferential distance between two adjacent vanes.

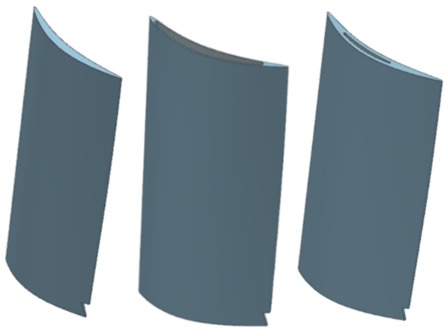


Parameter: Vane Material.

The vane can be manufactured in metal using Aluminium, Titanium or Steel.

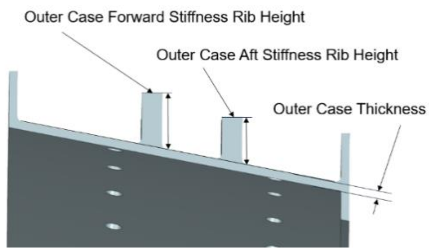
Parameter: Vane max thickness, Vane wall thickness, vane hollowness.

When reducing the number of vanes in the FOGV assembly, the total weight is increased. Mainly due to that vanes become longer and thicker so that the combined volume of the vanes is increased. When the vane is hollow the vane wall thickness is a parameter. If the vane configuration is of the three piece type and manufactured of carbon material there is a mid-foam material added.



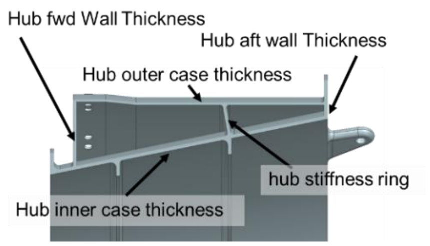
Parameter: Outer Case Forward and aft Stiffness Rib Height, Outer Case Thickness.

There are three parameters that can be varied independently. By varying the stiffness rib height the structure can be adopted to meet different stiffness and strength requirements.



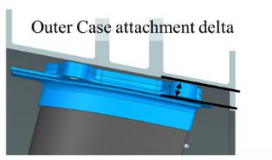
Parameters: Hub fwd Wall Thickness, Hub aft wall Thickness, Hub outer case thickness, Hub inner case thickness, optional hub stiffness ring.

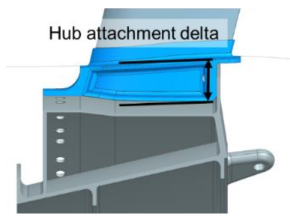
The hub part of the assembly plays an important role of providing a load path from the vane ring and the thrust lugs to the engine core. The hub also provides attachment for the electrical engine and the gearbox.



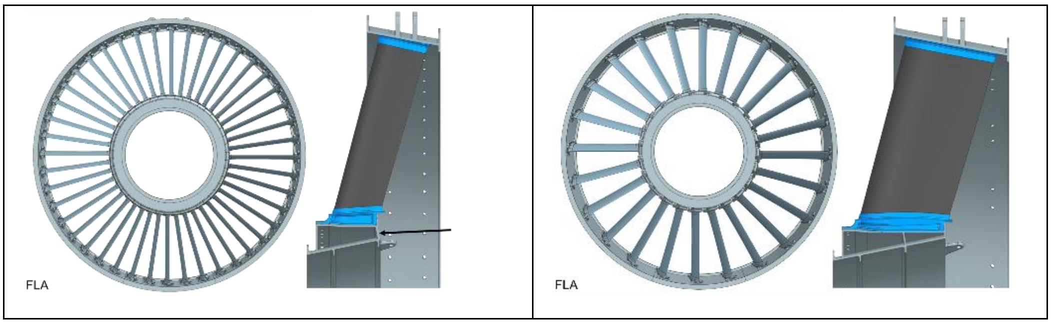
Parameter: HUB attachment delta, Outer Case attachment delta.

The attachment between the vane and outer case or inner hub is realised either with a fitting feature as illustrated in the picture to the right or integrated in the vane as illustrated in the one piece type of vane. Hence the thickness parameter, used here, is a delta parameter that either increase or decrease the thickness in relation to the reference design.





Below are two different vane counts illustrated. 50 Vane Count and 20 vane count.



The 16 parameters are included in a DOE for a Design Space Exploration study. The parameters are a mix of continues real, Nominal discrete and discrete by value, see **Error! Reference source not found.**. The distribution used is Latin Hypercube and 132 designs. The number of designs were decided on time available and amount of access to computer cluster.

For this study we are only evaluating the two resulting outputs for prediction, *Mass* and *supplier*. Where the *mass* becomes a fairly continues function that is based on all parameters, and supplier is non-liner and based on if then else rules coupled to the parameters; optional hub stiffness ring, Material, Vane max thickness, the number of vanes and the use of vane core. 5 different suppliers in total.

7.6.1. Predicting Mass

M*ass* is a continues function that is based on all parameters presented above.

Radial Basis Function

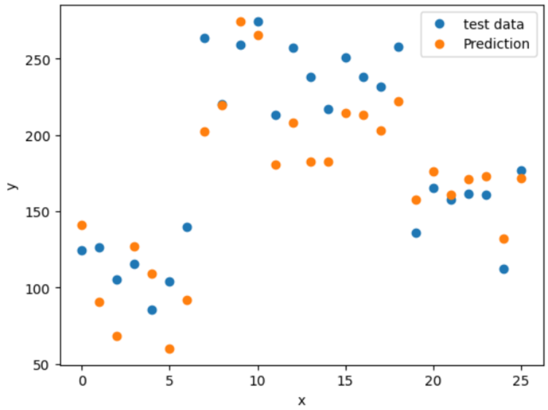


Figure illustrating the variation between test data and the predicted data.

Kriging

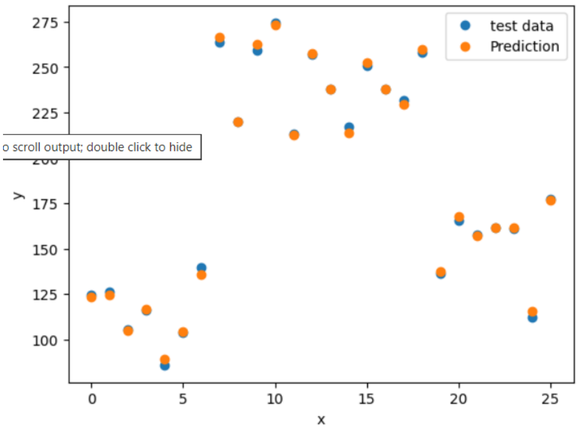


Figure illustrating the variation between test data and the predicted data.

Random Forest Regression

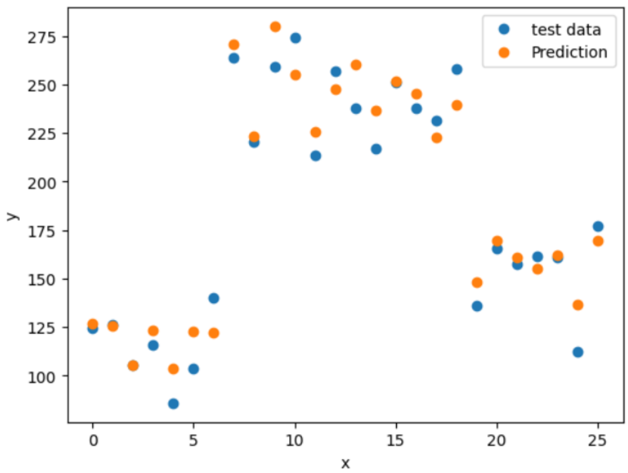


Figure illustrating the variation between test data and the predicted data.

Summary of results for response surfaces predicting the mass of the component.

Evaluation metric	Radial Basis Function	Kriging	Random Forest
Absolute percentage error	0.20	0.010	0.07
Explained variance score	0.40	1	0.89
R^2 score	0.28	1	0.88

The evaluation shows that Kriging is performing best and almost spot on of the surrogate models evaluated. Random forest is also performing good and although Radial Basis Function captures the trend it is not able to predict with any accuracy.

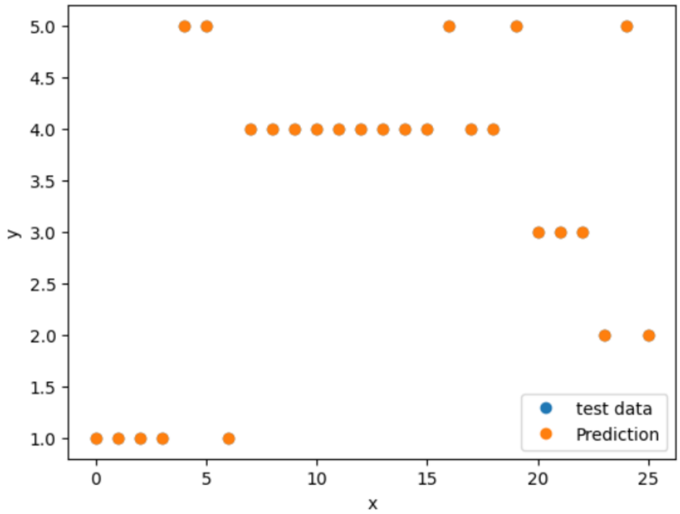
7.6.2. Predicting supplier

Supplier is non-liner and based on if then else rules coupled to 6 parameters; optional hub stiffness ring, Material, Vane max thickness, the number of vanes and the use of vane core. 5 different suppliers in total.

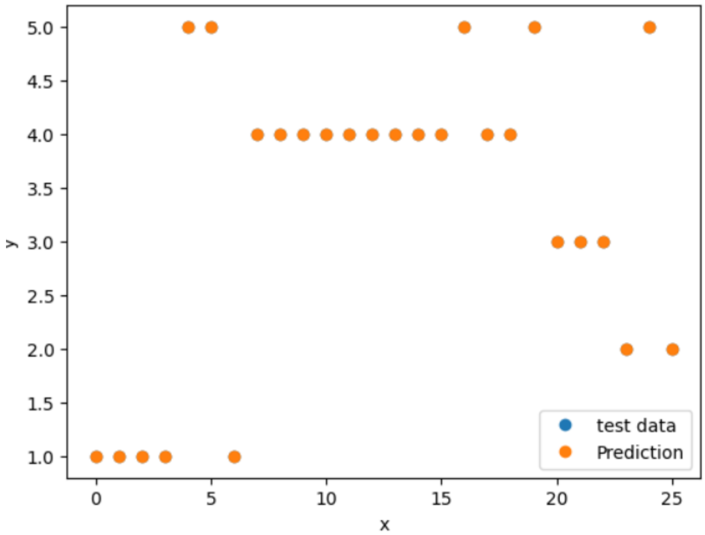
Radial Basis Function

It was not possible to use the Radial Bases Function provided in the SMT toolbox on this data set.

Kriging



Random Forest Regression (RFR)



Summary of results for response surfaces predicting the mass of the component.

Evaluation metric	Kriging	Random Forest
Absolute percentage error	1.73e-07	0.00261
Explained variance score	1	1
R^2 score	1	1

From the test performed it is clear that both Kriging and Random forest perform well on this type of data. It was not possible to use the Radial Bases Function, hence excluded from the table.

7.6.3. Conclusion

Similar to the first benchmark, Kriging is performing best for the datasets tested. This industrial use case included categorical data such as material and an optional feature, an extra stiffness rib in the hub. Also, the number of vanes represents countable but discrete values. This proved to be a challenge for the Radial Basis Function model although it managed to capture the trend when predicting the mass. The second output to predict was the preferred supplier, a categorical data output based on "*if then else"* statements resulting in 5 possible outcomes. This is not possible for Radial Basis Function and here both Kriging and Random Forest perform well on this type of data. Further benchmarking is needed to better understand the difference in performance between Random Forest and Neural network.

7.7. Industrial application – Aileron use case

The GKN Fokker Aerospace BV use case consists of an aileron, which can be varied in dimensions, structural layout, materials and loading. A surrogate model or response surface model (RSM) is created, which is validated on quality and subsequently used in typical concept study analysis. Toolsets used are GKN Fokker Aerospace BV's in house KBE applications combined with COTS NOESIS Optimus.

7.7.1. Approach

A surrogate model or RSM is based on a data set created with native toolsets. As this is time consuming, creation of the RSM is done in a frontloading phase. This starts with an initial Design of Experiments (DOE), aimed at deducing each design variable (DV) contribution to the objective, and to decide which DV will be taken into account for the RSM. Next an adaptive DOE algorithm is used, combined with an initial start data set to create the complete data set to be used for the RSM. Next a RSM algorithm evaluation is done, using multiple RSM algorithms and compare their mathematical quality indicators. These mathematical quality indicators are a good initial measure, however shown to be not sufficient to be fully confident on quality. Therefore, an additional check data set is created, with which the native result is compared with the RSM prediction. In case the percentage difference is lower than a certain acceptable percentage the RSM is accepted.

7.7.2. Application

The RSM is targeted to be used in design phases where time is limited. In this case in conceptual trade studies before contract award. Two typical examples are shown. First one is a small aileron with low loading (SLL) and second a larger aileron with higher loading (LHL). For both a multi-objective optimization (MOO) with cost and mass as objectives is run, the plots in Figure 15 and Figure 16 below show the pareto front combined with the iteration results. MOO results are verified with a single objective optimization (SOO) run per objective, which confirms the MOO results. And finally a native run is done for the optimum points to check possible differences. These differences are lower than 2.5%, which is highly acceptable for the active design phase. Also results show that depending on the inputs different options become optimal for different areas in the design space, which shows different product behavior is supported by the use case. Next the RSM is used to explore the sensitivity of the objectives to a certain standard deviation in input. In this case it is assumed the OEM specifies the loading with a certain standard deviation. Using a Monte Carlo analysis the resulting standard deviation of the objectives is found, see Figure 17. In this case, the MC analysis is using 100 experiments, which cannot be done with native runs within an acceptable time frame, showing the added value of the RSM.

7.7.3. Evaluation

In this case the RSM is used in a phase with limited time. Which means it needs to be fast but also have an acceptable quality level. The initial quality checks during creation as well as the native run checks of the optima show <2.5% differences, which is very acceptable.
The creation of the native dataset took 27 hours to run, RSM creation + validation took 0.5 hr. Running 1 RSM call is < 1 sec, 1 native run is 201 sec.
Running the MOO takes 36 iterations, so with RSM it takes: 14 sec, with native runs it takes 7236 sec, which is a performance enhancement factor of 516. Including the native run check in the RSM runs, this factor becomes 33.
So the performance enhancement is well achieved, facilitating the opportunity to do an increased number of concept evaluations compared to the traditional way of working. Additionally the RSM offers the capability to run sensitivity studies not possible in acceptable lead times with native runs.

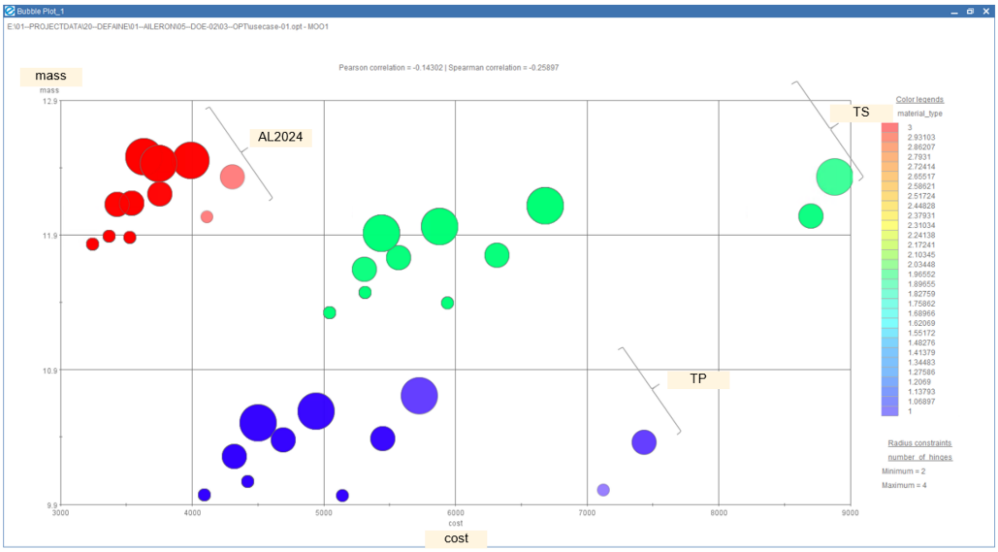


Figure 15 MOO analysis SLL aileron

Figure 16 MOO analysis LHL aileron

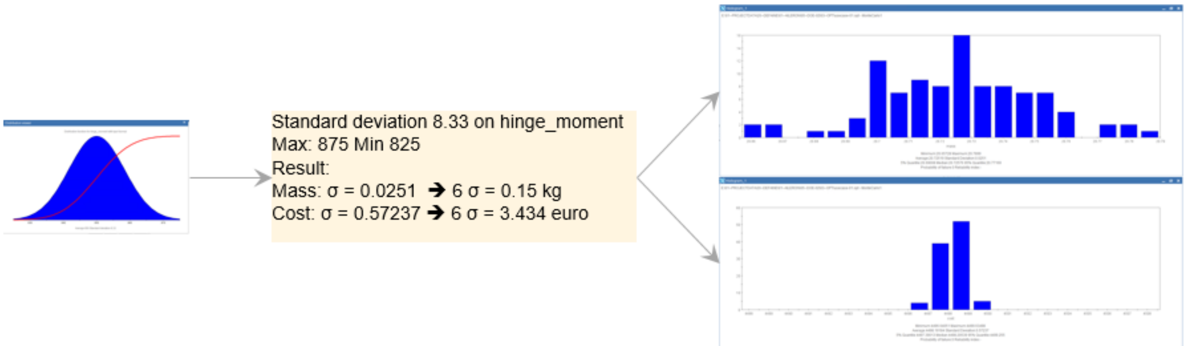


Figure 17 Monte Carlo analysis on standard deviation on hinge moment

7.8. Discussion and conclusion

Two benchmark studies and an industrial use case has been presented to better understand how well different surrogate models perform on different data sets and to provide an example of how the surrogate methodology can be applied.

References

[1] J. Sonneveld, T. van den Berg, G. la Rocca, S. Valencia Ibanez, B. van Manen, A. Bruggeman and B. Beijer, “Dynamic workflow generation applied to aircraft moveable architecture optimization,” in *CEAS*, Lausanne, 2023.

[2] DEFAINE Consortium, “D4.1.3. Final release of workflow (re-) formulation tool(s),” ITEA3 Call 6, 2023.

[3] DEFAINE Consortium, “D4.1.2. Second release of Workflow (re-)formulation tool(s),” ITEA3 Call 6, 2022.

[4] DEFAINE Consortium, “D4.1.1. First release of Workflow (re-)formulation tool(s),” ITEA3 Call 6, 2022.

[5] DEFAINE Consortium, “D4.2.1. CMDOWS,” ITEA3 Call 6, 2022.

[6] “SMT 2.3.0 documentation,” [Online]. Available: https://smt.readthedocs.io/en/latest/index.html#. [Accessed 12 1 2024].

[7] “scikit-learn - Machine Learning in Python,” [Online]. Available: https://scikit-learn.org/stable/index.html. [Accessed 21 11 2023].

[8] GKN Aerospace, “GKN aerospace to lead development of electric fan thruster for electric aircraft,” 2021. [Online]. Available: https://www.gknaerospace.com/en/newsroom/news-releases/2021/gkn-aerospace-to-lead-development-of-electric-fan-thruster-for-electric-aircraft/. [Accessed 30 01 2023].

[9] P. Andersson, M. Lejon, A. Pradas and M. Jacobson, “Demonstrating an approach for multidisciplinary set-based design within an aerospace research project - DEFAINE,” in *33RD Congress of the international council of the aeronatical sciences*, Stockholm, 2022.

[10] P. Andersson, M. Lejon, S. Chidambaranathan, S. Dasari Wejletorp, M. Panarotto and M. Jacobson, “Multidisciplinary design space exploration: An electric fan thruster component design use case,” in *Aerospace Europe Conference 2023 - 10:th EUCASS - 9:th CEAS*, 2023.

1. Schachinger P, Johannesson HL. Computer modelling of design specifications. Journal of engineering design. 2000 Dec 1;11(4):317-29. doi: 10.1080/0954482001000935.

2. Raudberget, D., Levandowski, C., Isaksson, O., Kipouros, T., Johannesson, H. and Clarkson, J., 2015. Modelling and assessing platform architectures in pre-embodiment phases through set-based evaluation and change propagation. Journal of Aerospace Operations, 3(3-4), pp.203-221., doi: 10.3233/AOP-150052.
3. Müller, J.R., Siiskonen, M.D.I. and Malmqvist, J., 2020, May. Lessons learned from the application of enhanced function-means modelling. In Proceedings of the Design Society: DESIGN Conference (Vol. 1, pp. 1325-1334). Cambridge University Press. doi: 10.1017/dsd.2020.87.
4. Müller, J.R., Isaksson, O., Landahl, J., Raja, V., Panarotto, M., Levandowski, C. and Raudberget, D., 2019. Enhanced function-means modeling supporting design space exploration. AI EDAM, 33(4), pp.502-516.
5. La Rocca, G., 2012. Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design. Advanced engineering informatics, 26(2), pp.159-179.
6. van den Berg, T. and van der Laan, T., 2021. A multidisciplinary modeling system for structural design applied to aircraft moveables. In AIAA AVIATION 2021 FORUM (p. 3079).
7. AD0759199 Stress Analysis Manual, US Air Force, https://apps.dtic.mil/sti/citations/AD0759199 [Last accessed on December 2023]
8. Panarotto, M., Isaksson, O., Habbassi, I., & Cornu, N. (2020). Value-Based development connecting engineering and business: A case on electric space propulsion. *IEEE Transactions on engineering management*, *69*(4), 1650-1663.
9. Panarotto, M., Bertoni, M., & Johansson, C. (2019). Using models as boundary objects in early design negotiations: analysis and implications for decision support systems. *Journal of Design Research*, *17*(2-4), 214-237.
10. Piotrowski, W., Kipouros, T., & Clarkson, P. J. (2019, September). Enhanced interactive parallel coordinates using machine learning and uncertainty propagation for engineering design. In *2019 15th International Conference on eScience (eScience)* (pp. 339-348). IEEE.
11. Mandel, J. (1982). Use of the singular value decomposition in regression analysis. Amer. Statist., 36(1), 15–24.
12. Feng, X., Sander-Tavallaey, S., & Ölvander, J. (2007). Cycle-Based Robot Drive Train Optimization Utilizing SVD Analysis. 33rd Design Automation Conference, Las Vegas, Nevada, USA.
13. Krus, P. (2017). Design Space Configuration Trough Analytical Parametrization. In A. Chakrabarti & D. Chakrabarti (Eds.), Research into Design for Communities, Volume 1. Proceedings of ICoRD 2017 (pp. 15–24). Springer Singapore. https://doi.org/10.1007/978-981-10-3518-0
14. Krus, P. (2017). Design Space Configuration Trough Analytical Parametrization. In A. Chakrabarti & D. Chakrabarti (Eds.), Research into Design for Communities, Volume 1. Proceedings of ICoRD 2017 (pp. 15–24). Springer Singapore. https://doi.org/10.1007/978-981-10-3518-0
15. Qian, C., Cong, X., Yang, C., Chen, W., Su, Y., Xu, J., Liu, Z. and Sun, M., 2023. Communicative agents for software development. arXiv preprint arXiv:2307.07924.

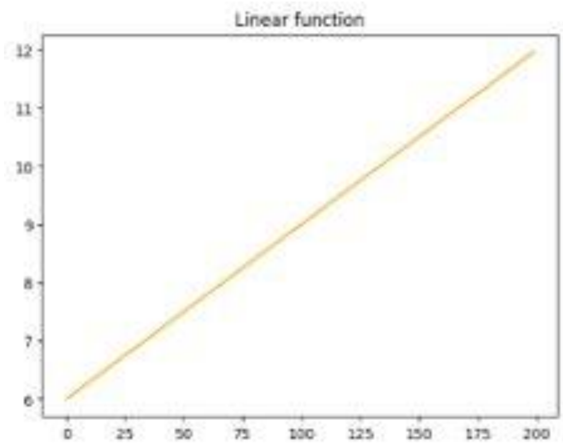
8. Appendix

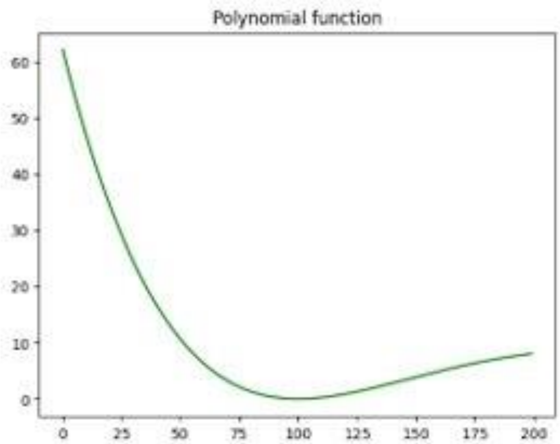
8.1. Benchmark of synthetic problems

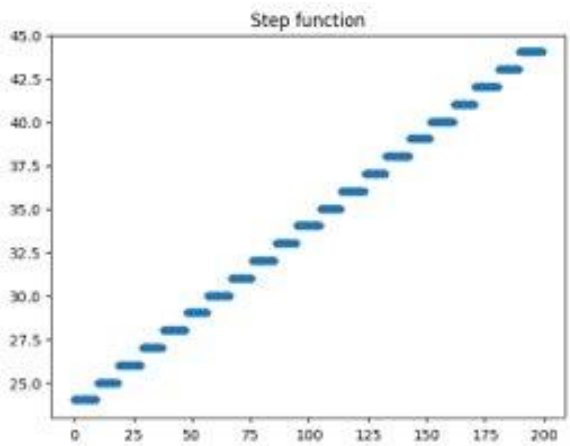
The bulk of the result from the first study is presented in this chapter to make the document easier to read. Hence, some of the information is repeated here to provide the context.
From experience of typical engineering DSE studies, synthetic problems are defined as mathematical functions that mimic similar behaviour in order to provide different challenges for the surrogate models. This data can be used to try available response surfaces to see how they perform. Data are described by mathematical functions, which formulates the characteristics of the data and enables the creation of large data sets. Different response surfaces are used on the data sets to see how they perform.

8.1.1. Distribution of numeric data

Parameters may have simple distribution functions while others can be more complicated.
Here we have three parameters with different distribution functions that are common in our data

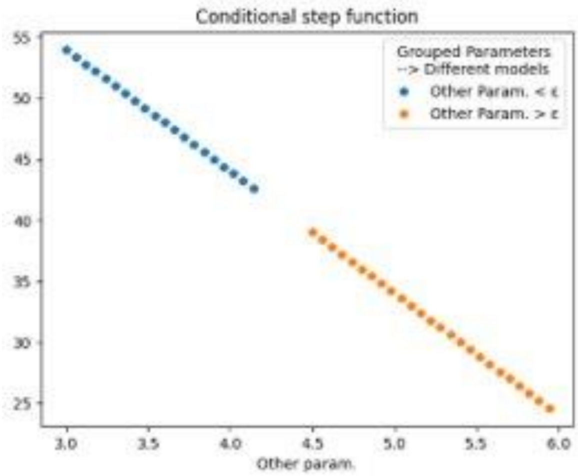


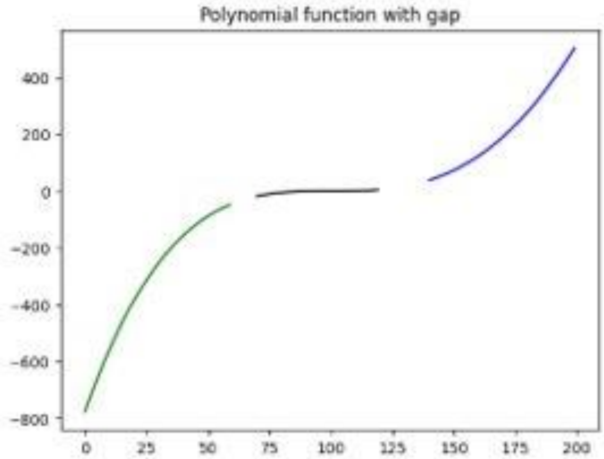




The functions can also be more complex and conditionally dependent on other parameters.

For example - the number of vanes on a model can have different sets of possible values depending on a related thickness parameter.





The models performances is measured on 50 linearly sampled data points using leave-one-out cross validation with 30 iterations for predicting the left out data point. Categorical features, whenever included, are subjected to one-hot encoding, a common method of representing categorical features in a numerical form that models can handle.

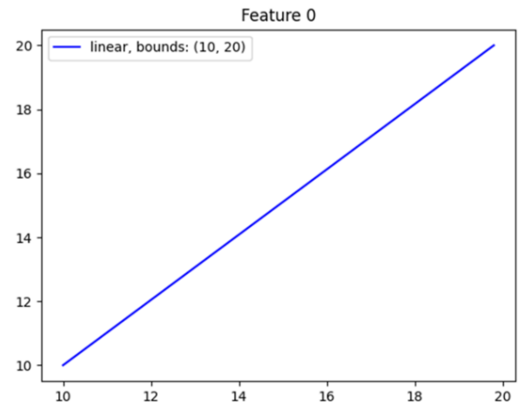
The results are divided into two sections, *Only numeric parameters* and *Numeric and categorical data.* The details are explained before each subsection.

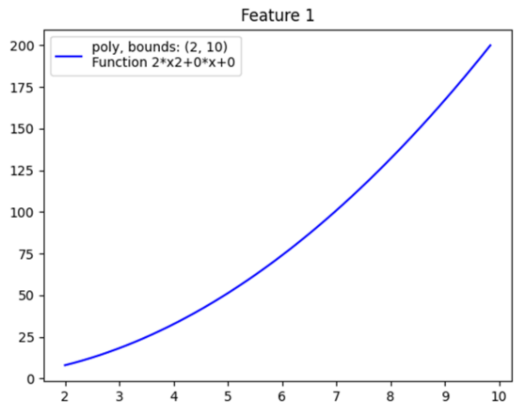
1. Only numeric parameters:
   Continuous input data
   Non-continuous input data

2. Numeric and categorical data
   Continuous numeric input data and ordinal feature
   Continuous numeric input data and categorical feature
   Non-continuous numeric input data and ordinal feature
   Non-continuous numeric input data and categorical feature

8.1.2. Continuous input data

Continuous input data and Non-continuous input data
First parameter – Linear within (10, 20)
Second parameter – Polynomial $f(x) = x^2$ with domain in (2, 10)

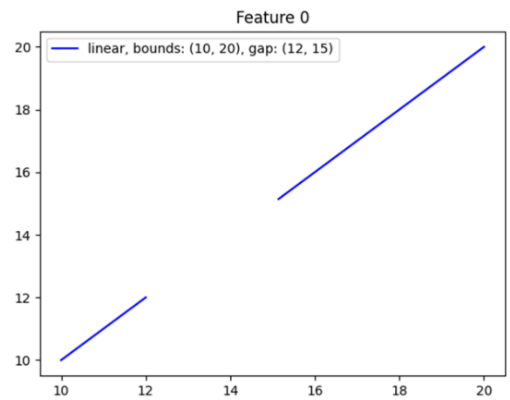


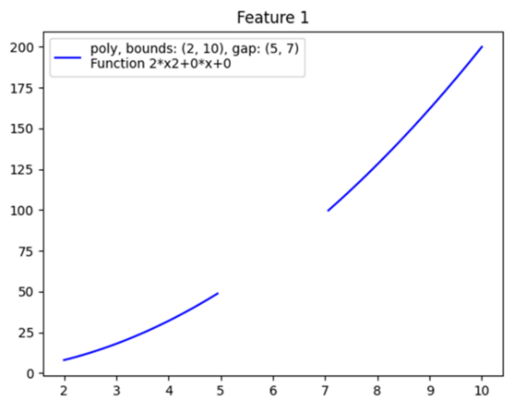


Problem SMAPE / R^2	**Kriging**	**Random Forest**	**Multi-layer Perceptron**	**Nearest Neighbors**	**Support vector machine**
Linear 2x + 7y	0.0% / 1.0	2.1% / 1.0	21.1% / 0.48	2.2% / 1.0	32.0% / 0.03
Polynomial x^2 + 2y^2	0.0% / 1.0	3.1% / 1.0	86.6% / -0.75	3.7% / 0.99	55.3% / -0.23
Exponential e^((x+y)/15)	0.21% / 1.0	5.0% / 0.85	73.9% / -0.20	3.53% / 0.66	80.1% / -0.23
Poly- & Exp e^(x^2+y^2)/120	100.0% / < 0	75% / -0.09	100.0% / < 0	80.1 % / < 0	100.0% / < 0

8.1.3. Only numeric parameters Non-continuous input data

First parameter – Linear within (10, 12) and (15, 20)
Second parameter – Polynomial with domain in (2, 5) and (7, 10)





Problem SMAPE / R^2	Kriging	Random Forest	Multi-layer Perceptron	Nearest Neighbors	Support vector machine
Linear 2x + 7y	0.0% / 1.0	1.32% / 1.0	19.8% / 0.53	2.01% / 1.0	50.2% / <0
Polynomial x^2 + 2y^2	0.0% / 1.0	2.5% / 1.0	84.7% / <0	3.6% / 0.99	80.3% / <0
Exponential e^((x+y)/15)	0.6% / 1.0	5.5% / 0.92	86.9% / <0	5.4% / 0.78	98.1% / <0
Poly- & Exp e^(x^2+y^2)/120	100.0% / < 0	66% / <0	89.0% / < 0	69.1 % / < 0	100.0% / < 0

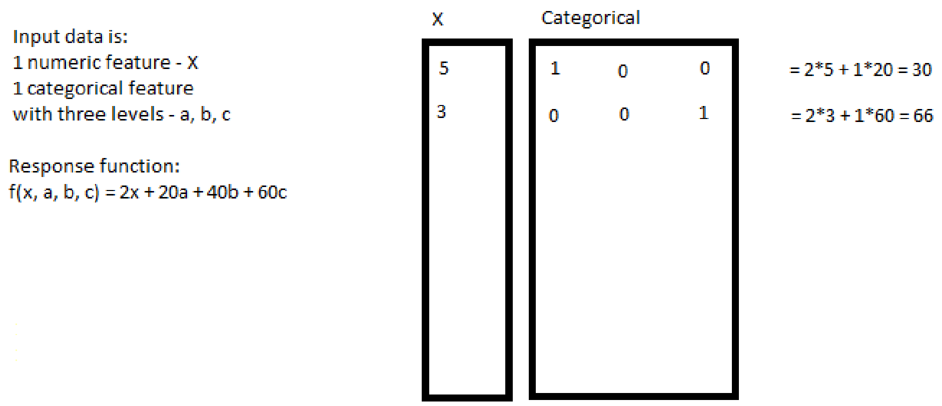
8.1.4. Continuous numeric and ordinal feature

First parameter – Linear within (10, 20)
Second parameter – ordinal with 3 levels – transformed numbers 1, 2, 3

Problem SMAPE / R^2	Kriging	Random Forest	Multi-layer Perceptron	Nearest Neighbors	Support vector machine
Linear 2x + 20y	0.0% / 1.0	0.50% / 0.99	13.8% / <0	4.1% / 0.82	9.5% / 0.26
Polynomial x^2 + 20y^2	0.0% / 1.0	3.4% / 0.94	75.9% / <0	2.9% / 0.96	15.1% / 0.02
Exponential e^((x+20y)/120)	0.0% / 1.0	1.1% / 1.0	45.3% / 0.05	19.0% / 0.77	45.2% / <0
Poly- & Exp e^(x^2+20y^2)/120	0.0% / 1.0	12.6% / 0.57	44.6% / 0.52	11.4 % / 0.55	50.3% / <0

8.1.5. Continuous numeric and categorical feature

First parameter – Linear within (10, 20)
Second parameter – Categorical with 3 levels – with one-hot encoding



Problem SMAPE / R^2	Kriging	Random Forest	Multi-layer Perceptron	Nearest Neighbors	Support vector machine
Linear 2x + 20y	0.0% / 1.0	0.38% / 1	15% / < 0	5.2% / 0.65	10.6% / 0.01
Polynomial x^2 + 20y^2	0.0% / 1.0	2.4% / 0.97	77.3% / < 0	3.5% / 0.91	15.5% / < 0
Exponential e^((x+20y)/120)	0.0% / 1.0	0.22% / 1.0	9.6% / < 0	1.6% / 0.79	3.4% / 0.59
Poly- & Exp e^(x^2+20y^2)/120	0.2% / 1.0	15.6% / 0.68	59.3% / < 0	24.7 % / 0.59	59.3% / < 0

8.1.6. Non-continuous numeric and ordinal feature

First parameter – Polynomial within (2, 5) and (7, 10)
Second parameter – Ordinal with 3 levels – transformed numbers 1, 2, 3

Problem SMAPE / R^2	Kriging	Random Forest	Multi-layer Perceptron	Nearest Neighbors	Support vector machine
Linear 2x + 20y	0.0% / 1.0	3.7% / 0.99	10.5% / 0.96	6.7% / 0.98	34.0% / <0
Polynomial x^2 + 20y^2	0.0% / 1.0	3.3% / 1.0	89.6% / <0	3.7% / 0.99	81.1% / <0
Exponential e^((x+20y)/120)	83% / 0.71	36% / 1.0	75% / <0	51% / 0.51	100% / <0

Problem SMAPE / R^2	**Kriging**	**Random Forest**	**Multi-layer Perceptron**	**Nearest Neighbors**	**Support vector machine**
Poly- & Exp e^(x^2+20y^2)/120	12% / 1.0	4.4% / 0.73	83% / <0	5.8 % / 0.49	66% / <0

8.1.7. Non-continuous numeric and categorical feature

First parameter – Polynomial within (2, 5) and (7, 10)
Second parameter – Categorical with 3 levels – with one-hot encoding

Problem SMAPE / R^2	**Kriging**	**Random Forest**	**Multi-layer Perceptron**	**Nearest Neighbors**	**Support vector machine**
Linear 2x + 20y	0.0% / 1.0	3.2% / 0.99	11.8% / 0.96	6.7% / 0.98	34.5% / <0
Polynomial x^2 + 20y^2	0.0% / 1.0	5.0% / 1.0	89.4% / <0	7% / 0.99	81.1% / <0
Exponential e^((x+20y)/16)	87% / 0.26	38% / 0.13	79.2% / <0	48.3% / 0.14	99.7% / <0
Poly- & Exp e^(x^2+20y^2)/6000	24.8% / 0.85	5.6% / 0.65	83.1% / <0	6.7 % / 0.49	71.3% / <0

8.1.8. Four polynomial parameters and a categorical feature with two levels

This case involves more parameters and higher complexity.
With response function Σ p_i^2 + 1000 * l_1 for polynomial parameters p_i and levels l_j

Problem SMAPE / R^2	**Kriging**	**Random Forest**	**Multi-layer Perceptron**	**Nearest Neighbors**	**Support vector machine**
Defined above	0.0% / 1.0	3.4% / 0.99	95.8% / <0	4.7% / 0.99	81.6% / <0

8.1.1. Conclusion

The results are divided into two sections, the first dealing with "*Only numeric parameters"* and the second dealing with "*Numeric and categorical data".* The details are explained before each subsection. The problems are deliberately designed to capture specific phenomena's or challenges like undefined areas or gaps in an otherwise continues function. Some problems were defined as steps and mixtures of continues and categorical input such a polygonal function and "material". An important detail is that due to stochasticity, model performance is subject to some variation – thus, the exact numbers in the result tables can vary between iterations. It should also be noted that further adjustments of the parameters for the algorithms tested could improve the performance of the algorithms, hence this study should only serve as a guideline when comparing different response surface models. With respect to the included algorithms and the problems presented in the study, Kriging is performing best for almost all circumstances, including a mixture of continues, none continues and categorical data. The only situation where Kriging did not perform well was with enormous exponential responses >10^10. Thus, if the response feature has these magnitudes, an idea could be to rescale the input features. There are several interesting ideas to dive into and improvements to be done. E.g. non-uniform sampling of the input data. As of now, data is sampled uniformly, but it might be interesting to sample from a skewed distribution to see how the models perform. Improved analytics and automatic visualization; perhaps given some inputs. This is something that could be improved to better understand the qualities of different models. Iterating models via hyperparameter tuning, e.g., Randomized Search CV and providing parameters to tune. As of now, the model uses the default parameters.