Robustimizer graphical user interface

# Abstract

Our primary goal is to provide an easy-to-use and cutting-edge optimization software under the influence of uncertainties by providing a graphical user interface which is an accessible platform to the general public. Robustimizer, a graphical user interface is introduced for surrogate-model-based robust optimization. The main features of Robustimizer are introduced such as analytical propagation of noise via the surrogate models, performing principal component analysis on the input noise, automatic update of the initial design of experiments based on multiple criteria, and communication with user-defined scripts and input files.

**Keywords:** Robustimizer, Surrogate modeling, uncertainty propagation

# Introduction

One-stage processes are common in many disciplines such as engineering, physics, biology and economy. For example, stamping is a widely used one-stage manufacturing process for creating metal components such as auto-mobile parts, appliance casings, or metal brackets. In this process a sheet of metal is placed in a stamping press, and a die cuts or forms the material into the desired shape. In many other manufacturing processes the transformation from raw material to finished product occurs in a single step. Chemistry and pharmaceutical industry involve many single-step processes [1, 2]. For instance synthesis of chemical products usually requires controlling the input parameters in which adjusting these parameters, influences the yield and quality of the product. In these processes, the operator selects a set of input parameters based on trial-and-error or experience to control one-stage processes. Those parameters that are controllable are referred to as control parameters.

In many processes, some of the input parameters are not controllable, very expensive to control or hard to measure in exact sense. Such parameters which can be described using a probabilistic distribution are referred to as noise parameters. In the presence of noise parameters, the output is no longer deterministic and therefore variations exist in the output. The optimization of processes under the influence of uncertainties is referred to as robust optimization in many disciplines [3-7]

The presence of disturbances introduce challenges in optimizing the processes. Major challenges are predicting their effects on the uncertainty of the output of the process. More importantly when the process is nonlinear, when there is only partial knowledge about the uncertainty, and when there are correlations among the parameters the robust optimization is more challenging. Computer models or expensive experiments are often used for optimizing processes. Optimizing a process under the influence of uncertainty using computer models requires many evaluations of the model and usually those models are computationally expensive. Experimental procedure for optimization is also very costly. Often an approximate model of the process, a so-called surrogate model, is built from results of computer simulations or experiments. Once the surrogate model is constructed, assessing the output for an un-simulated point becomes significantly cost-effective compared to running simulations or performing experiments. The surrogate model can then be used to represent the process in optimization in the presence of uncertainty.

Despite its potential, robust optimization and optimization under the influence of uncertainties have not yet gained popularity in practice and remain limited to specific research lines [8]. One of the main reasons that hinders the application of robust optimization in practice is that the existing resources does not enable everyone to easily apply the concept of robust optimization. By developing a GUI we provide an easy-to-use and cutting-edge optimization software with the comfort of self-guided interface. In this work, we introduce Robustimizer and we highlight its capabilities. This software holds immense potential for application in numerous fields, making it a valuable tool for diverse industries, including the forming industry, chemical industry, and pharmaceutical industry. Its versatility opens up new avenues for optimizing processes and achieving greater efficiency and effectiveness in these domains. By harnessing its capabilities, practitioners in these fields can unlock innovative solutions and drive advancements in their respective areas of expertise.

# Building blocks of robust optimization

A surrogate model-based robust optimization procedure consists of several building blocks. First a design of experiments (DOE) is generated in the combined control and noise space. Then the responses of the process model are evaluated for the discrete DOE points. In the next step, surrogate model is constructed which is an approximate representation of the process. The search for the robust optimum control parameter consists of uncertainty propagation and the repetitive evaluation of objective function value and constraints, which subsequently leads to the optimal control parameters, *x*opt. Since the reliance on a surrogate model might lead to loss of accuracy in the evaluation of the robust optimum iterative improvement of the surrogate model is applied. In the context of surrogate modelling, exploration and exploitation are two key concepts that relate to how the surrogate model is utilized. Exploration refers to the process of gathering information about the input output

relationship across the entire input space. exploitation, on the other hand, involves utilizing the surrogate model to make predictions or decisions that maximize the desired outcome within the known regions of interest. Striking a balance between exploration and exploitation is crucial in surrogate modelling to ensure that the model is both accurate and efficient in finding the robust optimum. New DOE points are added to the initial DOE to get an improved surrogate model of the process. A new infill point is selected in combined control and noise space and is added to the initial DOE. This procedure can be repeated until the updated surrogate model does not lead to further improvement of the predicted optimum control parameter.

For each building block of robust optimization, various methods exist and a variety of functions and toolboxes can be combined to perform robust optimization [9-11]. Robustimizer presents all these steps in one easy-to-use graphical interface which is presented in the next section.

# Introduction to Robustimizer

The GUI of Robustimizer is shown in Figure 1 and steps of robust optimization are listed in the various tabs discussed in this section. The user starts from the first tab and naturally proceeds to the next tabs after providing the required input. Throughout the use of GUI, the user is provided guidance. In general, a problem can be formulated by following these generic steps in each tab:

1. Define number of control parameters, noise parameters and outputs

Provide bounds of control parameters and statistical information about the noise parameters

1. Create the initial DOE

Import the results of experiments or simulations

1. Fit surrogate model and validate it
2. Define desired optimization settings
3. Perform optimization
4. Add additional DOE points for improving results (Optional)

In this section the theoretical aspects of implementation of each building block is presented.

## Initializing problem

The vector of control parameters is referred to as **x** and the vector of noise parameters is referred to as **z**. The vector **v** = (**x***,***z**) is input for the process model. The process model is evaluated for a selected number of different values of **v**, the Design of Experiments (DOE) made using Fractional or full factorial design (FFD).

The robust optimization starts by deciding on the number of control and noise parameters with the help of sensitivity analysis step. Robustimizer supports not only independent noise parameters but also performs principal component analysis (PCA) on correlated noise data imported from measurements.

The correlations among noise parameters are evaluated for a better description of the input noise. For this purpose, the covariance principal component analysis is used to find

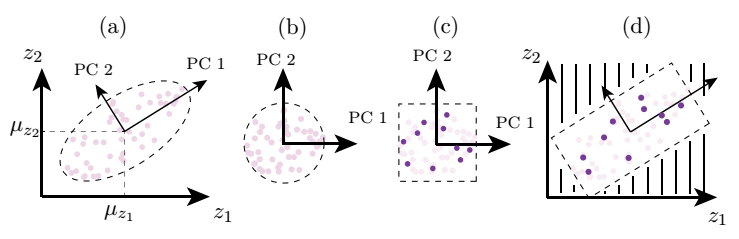


Figure 2: Schematic of sampling from correlated parameters

the principal components (PCs) of the noise parameters as described by [12]. The purpose of PCA analysis is to get uncorrelated subspace, to possibly reduce the parameter space and to avoid sampling of physically-unlikely parameter combinations when the DOE is created.

The orthogonal transformation from the correlated data to linearly uncorrelated PCs is shown schematically in Figures 1(a) and 1(b). The DOE is then created in the PC space (Figure 1(c)) avoiding physically-unlikely parameters sets, the hashed area shown in Figure 1(d). A set of physically-unlikely parameters can lead to an overestimation of the response scatter during the robust optimization [12]. For evaluation of the process model, a back transformation from the PC domain to the physical parameter domain is performed. The mutual linear correlation coefficient for various material parameters are calculated by:



where **X** and **Y** are two random variables centered about their mean, *σ*X and *σ*Y are their standard deviations, and Cov(**X***,***Y**) is the covariance, the degree of linear relationship between **X** and **Y**. The value for the correlation coefficient ranges from -1 to 1 and it indicates the amount of correlation between two parameters.

## Making a DOE and process model evaluation

In robust optimization, The DOE can be made using various schemes such as factorial design, central composite, random sampling, Latin hypercube sampling (LHS), and orthogonal sampling [13, 14]. It is desirable to have a small number of DOE points since the process model is often expensive to evaluate. In addition, the goal is to sample uniformly to get an accurate approximation of the underlying relationship between input and output in the whole control-noise domain.

Figure 2 provides a schematic representation of various sampling techniques for two input variables. In a full factorial or fractional factorial sampling (Figure 2-(a,b)), extreme values of each parameter are used. For the control parameters, the lower and the upper bound are selected as extreme values. The extreme values for a noise parameter are generally *µ*z ± 3*σ*z in which *µ*z is the mean and *σ*z is the standard deviation of that noise parameter.

A group of black squares with black dots

Description automatically generated

Figure 2: Various methods for generating a DOE

In random sampling the DOE points are generated without considering the previously generated points (Figure 2(c)). Therefore, it is very probable that the points are not evenly distributed. A central composite design (Figure 2(d)) consists of three types of points including FFD, the center point, and axial (star) points. In LHS the range of each parameter is divided into *n*DOE equiprobable bins and each bin is sampled once (Figure 2(e)). Orthogonal sampling is an extension to LHS in which the sampling domain is divided into sub-domains and, similarly to LHS, the domain is sampled such that each subdomain has the same density of points (Figure 2(f)). The LHS method can be implemented in such a way that the minimum distance between the points is maximized (Maximin). In that case, a uniform and disperse sample can be obtained (Figure 2(g)). The Maximin approach can also be used for combination of LHS and FFD (Figure 2(h)).

The choice of the size of the sample to build the DOE directly influences the accuracy of the surrogate model and the computational effort required to build the surrogate model. As a rule of thumb, it is proposed to select the number of sampling points equal to 10 times the number of input parameters for moderately complex functions. This number can be altered if a highly nonlinear process response is expected. Nevertheless, it is feasible to start using a small number of sample points and add infill points to the initial DOE at later stages of the robust optimization to improve the accuracy of the surrogate model where necessary. Robustimizer not only supports many standard methods such as LHS or factorial design, but also offers the flexibility to import any custom-made DOE in a text file.

The process model - sometimes referred to as black-box function - is evaluated in each DOE point to obtain the output. The response can be the result of a computer simulation, evaluation of the analytical model, or experimental results. The models and simulations are an approximation of a real process and therefore are not exact. They must be accurate enough to capture the influence of variation in the input parameters. If the difference between the predicted response and the real response is large, the process model must be improved. Furthermore, it is recommended to perform a study on the numerical noise of the model before running the optimization procedure, to make sure that the order of variation due to numerical noise is lower than the order of variation caused by noise parameters [12].

Process response can be imported via GUI using two different methods. The response can be evaluated in DOE points using any simulation tools and then imported via GUI in a text file. Alternatively, an executable script file can be selected which is capable of reading the DOE points as a text file and write the output in another text file. This feature offers a significant advantage as it enables the iterative addition of new DOE points in the later stages of optimization. Additionally, it allows for the automatic import of process model simulation results into the software. This will be covered in next sections.

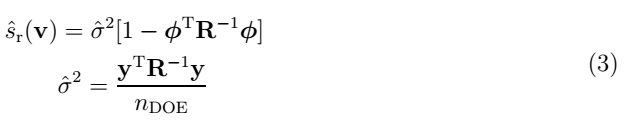
## Surrogate modeling and validation

The surrogate model describes the relationship between the response of the process model *r* (or constraint response, *r*c) and the input vector **v** = (**x***,***z**). The surrogate model is an approximate representation of the output obtained from process model evaluation. It is possible to assess the prediction error at each data point, enabling the iterative enhancement of the surrogate model’s prediction by utilizing this error evaluation.

A surrogate model built on the discrete responses obtained from evaluations of a process model is required to search for the robust optimum control. One can choose from several methods of surrogate modeling, such as Gaussian process (GP) models, radial basis function networks, neural networks or regression models [15]. The choice of the surrogate model depends on the complexity of the response with respect to input parameters. GPs are a widely used method for surrogate modeling and therefore it is implemented in Robustimizer by describing the response of the process model using:



In this equation, the vector *ϕ* contains the correlations between the point **x** and the DOE points, **R** is an *n*DOE × *n*DOE matrix that contains the spatial correlations between all DOE points, and **y** is the vector containing the responses of the process model on the DOE points. The error estimate, ˆ*s*r at every point can be calculated by:



The main assumption is that the data is a realization of a Gaussian random field which means that the responses are spatially correlated. The distance between the DOE points is used to determine the correlation between them.

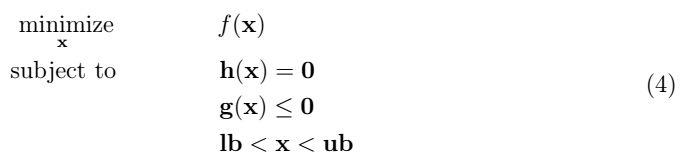
In surrogate modeling, validation techniques play a crucial role in assessing the accuracy and reliability of the surrogate models. These techniques help ensure that the surrogate models accurately represent the underlying system or process being modeled. One commonly used validation technique is cross-validation. Cross-validation involves dividing the available data into multiple subsets or folds. The surrogate model is then trained on a subset of the data and tested on the remaining subset. This process is repeated multiple times, with different subsets used for training and testing in each iteration. The results from each iteration are then averaged to provide an overall estimate of the surrogate model performance. One of the most commonly used cross-validation techniques is leave-one-out cross-validation (LOOCV). LOOCV is a special case of k-fold cross-validation where k is equal to the number of data points in the dataset. In each iteration of LOOCV, one data point is withheld as the test set, and the surrogate model is trained on the remaining data points. This process is repeated for each data point in the dataset, and the results are averaged to obtain the final performance estimate. Using the Surrogate model tab in Robustimizer, the user is able to fit, plot and validate the surrogate models based on GPs.

## Optimization settings

The optimization settings tab is shown in Figure 4. In this tab the solution methods, objective function and constraints and the uncertainty propagation methods can be chosen using various techniques that are discussed in this section.

## Solution methods

Generally, a robust optimization problem in the presence of constraints can be expressed as:



where *f*(**x**) is the objective function, **h**(**x**) are the equality constraints, **g**(**x**) are the inequality constraints, **lb** are the lower bounds of **x** and **ub** are the upper bounds of **x**. This formulation can be used for both deterministic and probabilistic optimization. The difference is in the definition of the objective function and constraints. The definition of the objective function and constraints for a robust optimization approach will be presented in the next section. The focus in this section is to introduce the methods that are used to solve such an optimization problem specifically when the objective function, the constraints, or both are nonlinear. In that case, Equation (4) is referred to as a constrained nonlinear problem. To solve such a problem, constrained nonlinear optimization algorithms can be employed [16]. There are two main classes of algorithms namely derivative-free or stochastic techniques such as genetic algorithm (GA), and iterative methods which require the derivatives such as iterative quadratic programming (SQP). A genetic algorithm maintains a large population of candidate solutions [17] in contrast to iterative search methods in which a single potential solution is generated at each iteration. GA is on the basis of bio-inspired operators (e.g. mutation, crossover and selection) and a population of candidate solutions evolves toward better solutions. Both methods are available to use with various classes of problems in Robustimizer.

### Objective function and constraints

One of the earliest methods for robust process design originates from the work of Taguchi [18]. The Taguchi method is generally used to classify robust design problems. The objective in robust design is one of the following:

* The smaller the better
* The larger the better
* On target is best

When the focus is mainly on the *performance* of the process (mean of the response), the first two approaches are employed. In the third approach, the Taguchi method typically involves a two-step procedure. In the first step, certain control parameters are identified to reduce the variability of the response. In the second step, different control parameters are utilized to shift the mean towards the target value.

A common approach, employed in Robustimizer is to simultaneously minimize the response variability while setting the mean at the target value. This approach must also satisfy certain constraints under the influence of uncertainties. Various methods can be employed to quantify the response variability and handle the constraints. In robust optimization, the satisfaction of constraints is directly associated with the probability of failure, often referred to as chance constraints in the literature. The main challenge in analyzing the constraints lies in evaluating the probability of failure since the response probability density function is unknown. To address this, a *moment matching* technique is frequently used, which approximates the reliability based on the statistical moments of the response. More specifically, the mean and standard deviation (or variance) are used to estimate the reliability of the satisfaction of a constraint using:



where *µ*rc(**x**) and *σ*rc(**x**) are the mean and standard deviation of the response of a constraint. The mean and standard deviation are simple to compute using limited stochastic data and therefore this method is widely used in the literature [19, 20]. The choice of *n* is related to the probability of constraint satisfaction assuming a normal distribution for the response of the constraint.

For a lower bound constraint:



There are several approaches to define the objective function, and one common measure for assessing the variability of the response is the standard deviation. The standard deviation provides an indication of the data spread around the mean value. For a robustness measure that minimizes the variation of response in addition to the difference between mean and target value, *C*r, several expressions are proposed in such as:

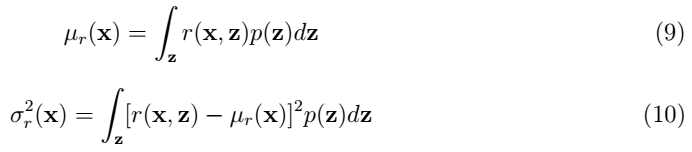


|  |  |
| --- | --- |
| Or: |  |
|  |  |

In Equations (7) and (8), *w* is a weighting factor to adjust the optimization objective between mean on target and response variation. By varying *w*, this weighted sum formulation can lead to a set of optimal solutions. Then a set of Pareto optimal solutions is obtained which indicates the trade-off between the deviation of the mean from the target and the variation of the response.

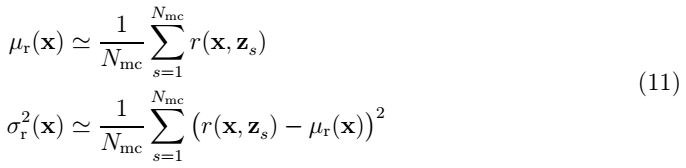
### Noise propagation

The criteria used for evaluation of objective function and constraints require the statistical moments of the response (mean and standard deviation). Finding the statistical moments of the response from the noise parameters is referred to as noise propagation. Several methods for noise propagation have been developed over past decades. Monte Carlo (MC) and its variations, perturbation methods, Gaussian Quadrature (GQ), polynomial chaos, Bayesian statistical modeling, method of moments (Taylor-series expansion) and stochastic collocation have been widely used [21]. Dimensionality of the problem [22] and fidelity of the model [23] determine the efficiency and effectiveness of these methods. The mean and variance of response are expressed by the following integrals in the noise space:

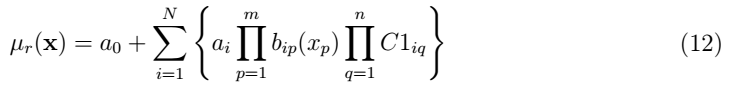


Monte Carlo (MC) analysis is one of the most widely used methods in the literature for propagation of noise and for approximating the above-mentioned moments [24, 25]. Sampling from the noise parameter involves selecting values from its probability distribution. Several methods can be used to perform this sampling, and the concept is akin to the sampling techniques described in Section 3.2. The primary distinction lies in the objective, while Section 3.2 focuses on uniform sampling, here the samples are drawn from a probability distribution. To perform the sampling, random numbers between 0 and 1 are chosen from the cumulative probability distribution (CDF) of a normal distribution and then mapped to the noise domain. With random sampling, the previously sampled points are not taken into account, and the points can belong to any subset of the sampling domain. In Latin Hypercube Sampling (LHS), the CDF is divided into sub-domains of equal size, and sampling is conducted by adding new points while ensuring that more than one point is not selected in each domain.

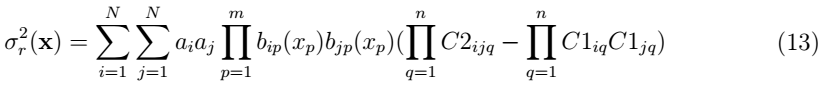
After choosing the samples, the approximate mean and standard deviation are evaluated by:

**

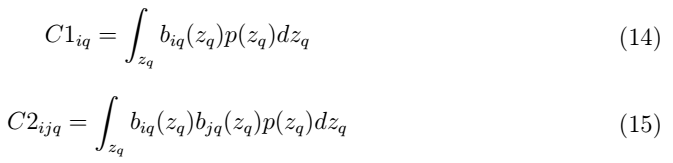
where *N*mc is the number of sample points drawn from the noise probability distribution, *p*(**z**). It has been shown by Nejadseyfi et al [39,40] that mean, standard deviation, higher order moments and their derivatives can be efficiently obtained by using analytical formulas for GPs and correlated/uncorrelated noise parameters by:



|  |  |  |
| --- | --- | --- |
| and: |  |  |

**

where *C*1 and *C*2 depend on the choice of surrogate model and probability distribution:



where *biP* (*XP* ) are basis functions for that surrogate model and *p*(**z**) is the input probability distribution. Once *C*1 and *C*2 are known for combination of surrogate model and probability distribution, mean and standard deviation of output can be evaluated at each control point. This method is programmed using vectorized operations which is computationally efficient compared to numerical integration e.g. by MC method and it is one of the main features that contribute to efficiency and speed of Robustimizer. Moreover, in gradient-based optimization, the gradients can also be calculated efficiently using:

A group of mathematical equations

Description automatically generated

(17)

(18)

## Evaluating the robust optimum and improving surrogate model accuracy

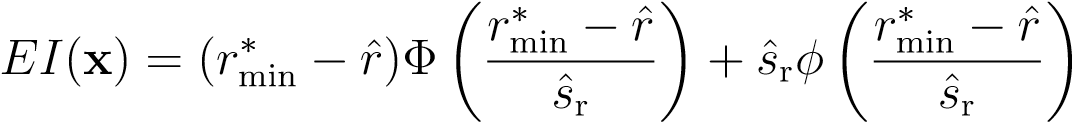
An optimum set of control parameters found using the surrogate model is not always equal to the optimum of the underlying process model. This occurs when the prediction behaviour of the surrogate model around the predicted optimum is poor when there are no sampling points around the optimum. Therefore, it is necessary to use an update procedure to improve the surrogate model and subsequently obtain an accurate robust optimum. Moreover, the optimum might be in the areas that are less explored due to the initial DOE design.

Gaussian processes provide a flexible framework for modeling the uncertainty associated with data points. They define a prior distribution over functions and, after observing data, produce a posterior distribution that incorporates the data information. When using Gaussian processes for Bayesian optimization or active learning, the goal is to find the optimal points in a high-dimensional search space with limited observations or evaluations. Exploration involves gathering more information about the unknown regions of the search space by adding more points into the design of experiments, while exploitation focuses on exploiting the regions that appear to be optimal based on the current knowledge. The

exploration-exploitation trade-off is often formalized using acquisition functions in the context of Gaussian processes. An acquisition function provides a measure of how valuable it is to evaluate a candidate point in the search space. The selection of the acquisition function depends on the specific problem and the trade-off between exploration and exploitation.

Unlike space-filling methods that the new points are added to the initial DOE in the sparsely-sampled regions, adaptive sampling techniques require a criterion to add an infill point where it is needed most. In some cases, the infill point is added at the predicted optimum. However, in surrogate-model-based robust optimization uncertainties the infill criterion is based on the surrogate model estimation error, ˆ*s*. Using this potential error, various methods can be developed to add infill points. A simple approach is to add an infill point where ˆ*s* has the biggest value.

One of the most used adaptive methods is based on *expected improvement [26]* that has been proposed to take into account both local and global search for new infill points in deterministic optimization. The expected improvement method has a closed-form of the acquisition function, which effectively uses the existing sampling points to improve the optimization accuracy. The expected improvement is defined by:

 (16)

In this equation, *ϕ* is the standardized normal distribution, Φ is the cumulative distribution of a standardized normal distribution and  is the minimum value of the response at the DOE points examined so far. ˆ*r* and ˆ*s*r are the predicted value and uncertainty of the predicted response. The procedure is to search for a point that has the highest *EI* value to add it to the initial DOE.

Equation (16) is used to optimize the response of a process (deterministic optimization).

It can be altered to be used in a robust optimization procedure [27]. A new infill point

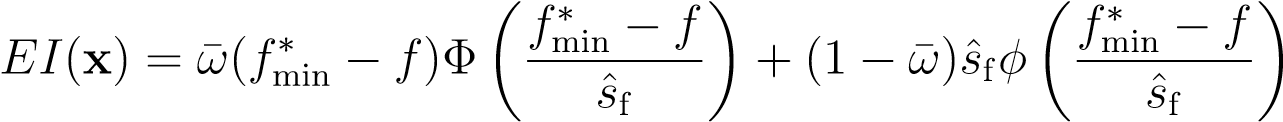
A black background with a black square

Description automatically generated with medium confidence) must be selected in the combined control and noise space. The objective function value, *f*(**x**), replaces the response, *r*(**x**). Moreover, to evaluate *f*(**x**) and get the minimum value of the objective function at the DOE points, A black background with a black square

Description automatically generated with medium confidence, one needs to calculate *µ*r(**x**) and *σ*r(**x**). Therefore, the objective function values are not a result of evaluations of the process model, but of a prediction using a surrogate model. Thus, there is a prediction uncertainty at the current best point. The minimum objective function value at the DOE points has an uncertainty of ˆ*s*∗. A suitable estimation for the prediction error at any control point, *s*ˆ(**x**), is also required. The uncertainty measure on surrogate model (ˆ*sr*) is dependent on both control parameters and noise parameters. To obtain the uncertainty of the objective function value (ˆ*sf*) an integral over noise space is usually evaluated (calculating mean value of mean square error (MSE)) [27,28]:



The influence of uncertainty of the best point, ˆ*s*∗, is ignored and expected improvement is evaluated using [43]:

 (18)

More details about including the influence of ˆ*s*∗ can be found elsewhere [28]. The first term in Equation (18) is related to local search (Exploitation near the predicted optimum) and the second term is related to global search (Exploration). One can adjust the search in the global and local domain by choosing a proper weight factor such that 0 *< ω <*¯ 1. Maximizing expected improvement using (18) leads to an infill point in the control space, **x**′. At that control setting, a point in noise space must be selected to be able to evaluate the process model. For this purpose **z**′ = argmax**z** A black background with a black square

Description automatically generated with medium confidence)) is employed. The point ( A black background with a black square

Description automatically generated with medium confidenceis then added to the initial DOE, a new surrogate model is fitted and the robust optimum is evaluated again using the updated surrogate model.

Since exploration-exploitation is an adaptive method, Robustimizer supports external scripts which provide the function evaluation on a newly added infill point. Using this method it is convenient to update the DOE automatically, evaluate the function output in the new DOE, fit a new surrogate model and repeat robust optimization sequentially. This feature and its settings are shown in he next section.

# Running Robustimizer

There are seven tabs on top which include the steps to formulate and perform robust optimization. Each tab shown in figure 3 is explained in detail in this section.

A screenshot of a computer

Description automatically generated

Figure 3: Main window of Robtimizer

## Sensitivity analysis

Tab has been removed

## Initializing problem

The user defines the main design and noise variables as shown in Figure 7. This tab is similar to Section ‎2.1.1 and the user enters the number of design variables, upper and lower bounds for design variables, and the statistics of noise variables. Similar to Section ‎2.1.1, there are two methods to define noise variables. If there are no measurements or available noise data the first option in choose method for noise description must be selected. After entering the number of noise variables, mean and standard deviation for each variables must be provided.

If the second option in choose method for noise description is selected, the user must provide a txt file to calculate the statistics of noise variables and possible correlations. The data for each noise variable must be included in one columns and the columns must be tab separated for multiple noise. If this option is selected, the textbox number of noise variables for SA and noise data table will be disabled and the number of noise variables will be assigned as the number of columns in the file. In addition, the statistics of the noise variables will be evaluated from the provided data.

It is optional to assign a name for each variable. The main difference of this tab with Section ‎2.1.1 is that the user must enter the number of constraints response. If there are no constraints this field can remain unchanged. The number of main responses is not editable and Robtimizer only allows one main response to optimize. If there are multiple responses, the user is capable of involving them as constraint response.

There are two options to evaluate the response(s) on these DOE points which are available under *Import results method* button group. The user can perform the blackbox evaluation outside Robtimizer, save them in a txt file and import the results using one txt file. Alternatively, an executable script file can be introduced to Robtimizer. In this case, Robtimizer will automatically read the results file when it is generated. This executable file should be provided by user (e.g. C++ or Fortran script) and this script must be able to read *in.txt* filein the running folder and generate *out.txt* file in the same folder. The input text file, *in.txt,* includes the DOE points in each row and the *out.txt* contains the main response and constraint responses (if applicable). After clicking on *Run script and import results* button, Robtimizer will wait for blackbox evaluation and generation of *out.txt*. The user can set a time limit for this operation using the *Timeout* field. When the output file is generated, the responses will be automatically imported and shown in the respective table.

A screenshot of a computer

Description automatically generatedFigure 6: *Initializing the problem*

## DOE and blackbox evaluation

In this tab, the user creates a DOE to evaluate the blackbox function. By default Robtimizer creates a Latin hypercube sample (LHS) after 1000 iterations using maximized minimum distance criterion. The user can enter the number of DOE points and decide whether or not to combine LHS with factorial design to improve DOE. By clicking on *Create DOE* button (Figure 7)a DOE is created at which the response must be evaluated. This DOE will be shown in the respective table and it can be saved in a txt file with tab-separated columns and it can be loaded later. The user can also provide a custom-made DOE made outside Robtimizer. Importing DOE is not possible when choosing the second option of *choose method for noise description* as DOE must include the possible correlations.

A screenshot of a computer

Description automatically generated

Figure 7: *Creating DOE and importing results of blackbox evaluation*

## Metamodel

Use this tab to create, validate and visualize the metamodel. Under *Fit metamodel* panel you can choose the type of metamodel and fit a metamodel. *Cross validation* panel can be used to validate the metamodel. Plotting and visualizing the metamodel is also part of this tab. It must be noted that plotting is performed only in 3 dimensions. One input on X axis and another input on Y axis. The response value will be shown in Z axis. The user can change the variables and responses and update the plot. If more than 2 input are present, the values for other variables will be set in their mean value and the user can change these values using the sliders in front of the name of each variable. Modification of X-axis variable and Y-axis Variable is possible using the drop-down menus. (Figure 8)

A screenshot of a computer

Description automatically generated

Figure 8: *Creating, validating and visualizing the metamodel*

## Optimization settings

Default optimization settings can be used to proceed with robust optimization. However, the user has more control on the methods of optimization in this tab shown in Figure 9. Optimization method, noise propagation method, objective function definition and constraints definition can be adjusted in this tab and the relevant edit fields will be enabled or disabled.

The objective function to minimize can be chosen in different ways as listed under Objective function for main response panel. If the user is willing to set the mean on target while minimizing the standard deviation, the target value must be set. For a non-normally-distributed main response it is possible to improve the predictions by considering the skewness of the output [29].

If there are constraints, the user can define a target value for the constraint and the type of the constraint for each constraint separately. For a non-normally-distributed constraint response it is possible to improve the predictions by considering the skewness of the response.

A screenshot of a computer

Description automatically generated

Figure 9: *Optimization settings tab*

## Robust optimization

After performing all the previous steps and defining the problem the user can proceed with robust optimization. By pressing the Perform optimization button, the optimization procedure starts and when the results are generated they will be shown in the text box in Figure 10.

To visualize the scatter of each response at the predicted optimum the user can perform a Monte Carlo analysis at the optimum under the Visualize scatter on optimum panel. The robust optimum obtained using the existing metamodel. The user can improve the metamodel to obtain a more accurate optimum. This can be performed in the next tab.

A screenshot of a computer

Description automatically generated

Figure 10: *Robust optimization tab*

## Sequential improvement

In this tab the user can choose a method to add an infill point to the current DOE and perform the optimization again. The default method is Jones criterion with a weighting factor of 0.5. Adding a new infill points can be done either manually or automatically. If the user has defined a script to automatically obtain the results in Section ‎0‎2.3 it is recommended to use the automatic option. In that case by entering the number of sequential improvement steps Robtimizer will automatically find a best infill point, add it to the existing DOE, run the script, fit a new metamodel, find the robust optimum and repeat this procedure. The best infill point and the optimization results will be shown in the text box in the panel for each trial.

For the manual case, the user will obtain a recommended infill point and add it to the existing DOE using the buttons in *Manual infill and optimization* panel. Subsequently the user returns to *DOE and blackbox evaluation* tab to import the results of the blackbox function evaluation using *Read results file* in that tab. It must be noted that the results of all DOE points must be imported. After that the user continues with fitting a new metamodel and finding the robust optimum. The user repeats this cycle as many times as required. The stop criteria can be determined based on the value of expected improvement criterion or change in the optimum design or objective function.

A screenshot of a computer

Description automatically generated

*Figure 11. Sequential improvement of the metamodel*

# Availability

Robustimizer is available to download and use at www.robustimizer.com

**References:**

1. Zhiqiang, L., et al., *A multi-objective robust optimization scheme for reducing optimization performance deterioration caused by fluctuation of decision parameters in chemical processes.* Computers & Chemical Engineering, 2018. **119**: p. 1-12.

2. Han, H.S. and K.Y. Choi, *Advances in Nanomaterial-Mediated Photothermal Cancer Therapies: Toward Clinical Applications.* Biomedicines, 2021. **9**(3).

3. Kann, A. and J.P. Weyant, *Approaches for performing uncertainty analysis in large-scale energy/economic policy models.* Environmental Modeling & Assessment, 2000. **5**(1): p. 29-46.

4. Myklebust, O., *Zero Defect Manufacturing: A Product and Plant Oriented Lifecycle Approach.* Procedia CIRP, 2013. **12**: p. 246-251.

5. Gao, L. and Z. Zhang, *Robust Optimization for Managing Pavement Maintenance and Rehabilitation.* Transportation Research Record, 2008. **2084**(1): p. 55-61.

6. Palmer, T.N., *Predicting uncertainty in forecasts of weather and climate.* Reports on Progress in Physics, 2000. **63**: p. 71-116.

7. Dettinger, M.D. and J.L. Wilson, *First order analysis of uncertainty in numerical models of groundwater flow part: 1. Mathematical development.* Water Resources Research, 1981. **17**: p. 149-161.

8. Nejadseyfi, O., et al., *From specified product tolerance to acceptable material and process scatter: an inverse robust optimization approach.* International Journal of Material Forming, 2020. **13**(3): p. 467-478.

9. Huang, D., et al., *Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models.* Journal of Global Optimization, 2006. **34**(3): p. 441-466.

10. Marzat, J., E. Walter, and H. Piet-Lahanier, *Worst-case global optimization of black-box functions through Kriging and relaxation.* Journal of Global Optimization, 2013. **55**(4): p. 707-727.

11. Kitayama, S. and K. Yamazaki, *Sequential approximate robust design optimization using radial basis function network.* International Journal of Mechanics and Materials in Design, 2014. **10**(3): p. 313-328.

12. Wiebenga, J.H., et al., *Effect of material scatter on the plastic behavior and stretchability in sheet metal forming.* Journal of Materials Processing Technology, 2014. **214**(2): p. 238-252.

13. Uy, M. and J.K. Telford. *Optimization by Design of Experiment techniques*. in *2009 IEEE Aerospace conference*. 2009.

14. Tang, B., *Orthogonal Array-Based Latin Hypercubes.* Journal of the American Statistical Association, 1993. **88**(424): p. 1392-1397.

15. Luo, J. and W. Lu, *Comparison of surrogate models with different methods in groundwater remediation process.* Journal of Earth System Science, 2014. **123**(7): p. 1579-1589.

16. Baginski, M.E., D.L. Faircloth, and M.D. Deshpande, *Comparison of two optimization techniques for the estimation of complex permittivities of multilayered structures using waveguide measurements.* IEEE Transactions on Microwave Theory and Techniques, 2005. **53**(10): p. 3251-3259.

17. Homaifar, A., C.X. Qi, and S.H. Lai, *Constrained Optimization Via Genetic Algorithms.* SIMULATION, 1994. **62**(4): p. 242-253.

18. Taguchi, G., *System of experimental design: engineering methods to optimize quality and minimize costs*. Vol. 1. 1987: UNIPUB/Kraus International Publications.

19. Scholar, V., X. Du, and W. Chen, *A Methodology for Managing the Effect of Uncertainty in Simulation-Based Design.* AIAA Journal, 2000. **38**.

20. Du, X. and W. Chen, *Efficient uncertainty analysis methods for multidisciplinary robust design.* AIAA Journal, 2002. **40**: p. 545-552.

21. Lee, S.H. and W. Chen, *A comparative study of uncertainty propagation methods for black-box-type problems.* Structural and Multidisciplinary Optimization, 2009. **37**(3): p. 239-253.

22. Fuchs, M. and A. Neumaier. *Handling Uncertainty in Higher Dimensions with Potential Clouds towards Robust Design Optimization*. in *Soft Methods for Handling Variability and Imprecision*. 2008. Berlin, Heidelberg: Springer Berlin Heidelberg.

23. Ng, L.W.T. and K.E. Willcox, *Multifidelity approaches for optimization under uncertainty.* International Journal for Numerical Methods in Engineering, 2014. **100**(10): p. 746-772.

24. Zhou, Q., et al., *Robust optimization for reducing welding-induced angular distortion in fiber laser keyhole welding under process parameter uncertainty.* Applied Thermal Engineering, 2018. **129**: p. 893-906.

25. Putko, M., et al., *Approach For Uncertainty Propagation And Robust Design In Cfd Using Sensitivity Derivatives.* Journal of Fluids Engineering, 2001. **124**.

26. Jones, D.R., M. Schonlau, and W.J. Welch, *Efficient Global Optimization of Expensive Black-Box Functions.* Journal of Global Optimization, 1998. **13**(4): p. 455-492.

27. Ur Rehman, S. and M. Langelaar, *Efficient infill sampling for unconstrained robust optimization problems.* Engineering Optimization, 2016. **48**(8): p. 1313-1332.

28. Sóbester, A., S.J. Leary, and A.J. Keane, *A parallel updating scheme for approximating and optimizing high fidelity computer simulations.* Structural and Multidisciplinary Optimization, 2004. **27**(5): p. 371-383.

29. Nejadseyfi, O., H.J.M. Geijselaers, and A.H. van den Boogaard, *Evaluation and assessment of non-normal output during robust optimization.* Structural and Multidisciplinary Optimization, 2019. **59**(6): p. 2063-2076.