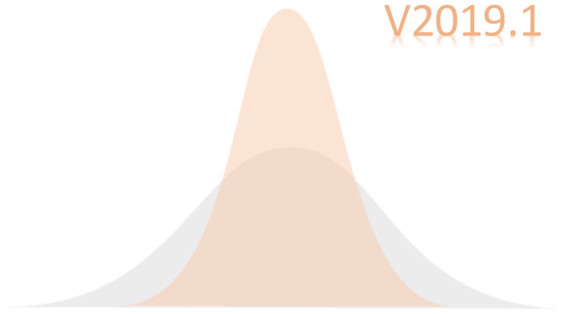
Robustimizer

V2019.1



1. Introduction

Robustimizer is a software tool to perform optimization under the influence of uncertainties. Consider a model that represents a process. The model has some input variables and by changing those inputs the output changes (Figure 1). If some of those inputs are affected by uncertainties or if they are out of control, the uncertainties will be present in the output.

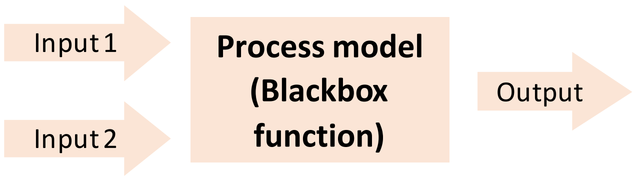


Figure 1. Process model with two inputs and one output

A simple example

A simple example is presented to explain robust optimization. Assume a basketball player is practicing a free throw outdoors. The process of basketball free throw can be modeled using the physics of projectile motion. The input of a simple model of projectile motion is the initial velocity and throw angle and the output is the final coordinates of the basketball and the angle of approach toward the hoop as shown in Figure 2. A criterion can be used to evaluate the success of throw e.g. as a function of the distance of the center of the ball from the center of the hoop and the angle of approach toward the hoop.

This model can be used to find optimum initial velocity and throw angle that will guaranty a successful free throw. As long as the velocity and throw angle remain the same, the basketball will land on the predicted final coordinate with the predicted angle of approach since this problem is deterministic (Figure 2-a). However, if a noisy input exists in the process, e.g. a front wind with a changing magnitude, then final coordinates using the optimum velocity and throw angle will not be the same in each try (Figure 2-b). The height at which the basketball is released can be identified as another noise variable in this process. In this case, it is necessary to include these uncertain inputs in the model and consequently the output will be uncertain. The uncertain input and output can be handled statistically e.g. using a normal probability distribution.

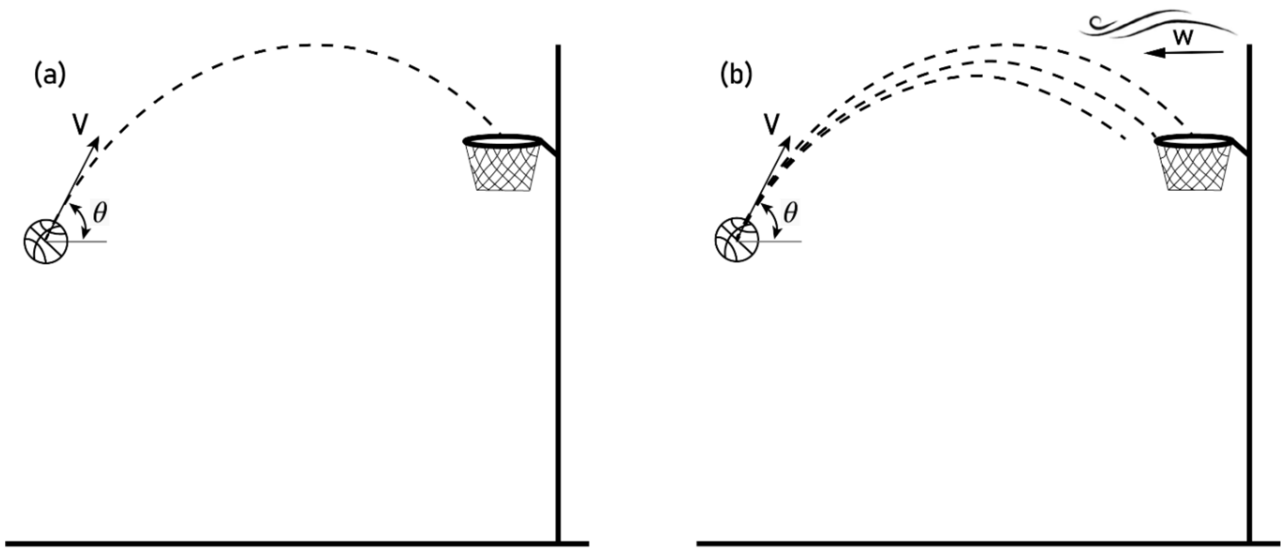


Figure 2. Basketball free throw

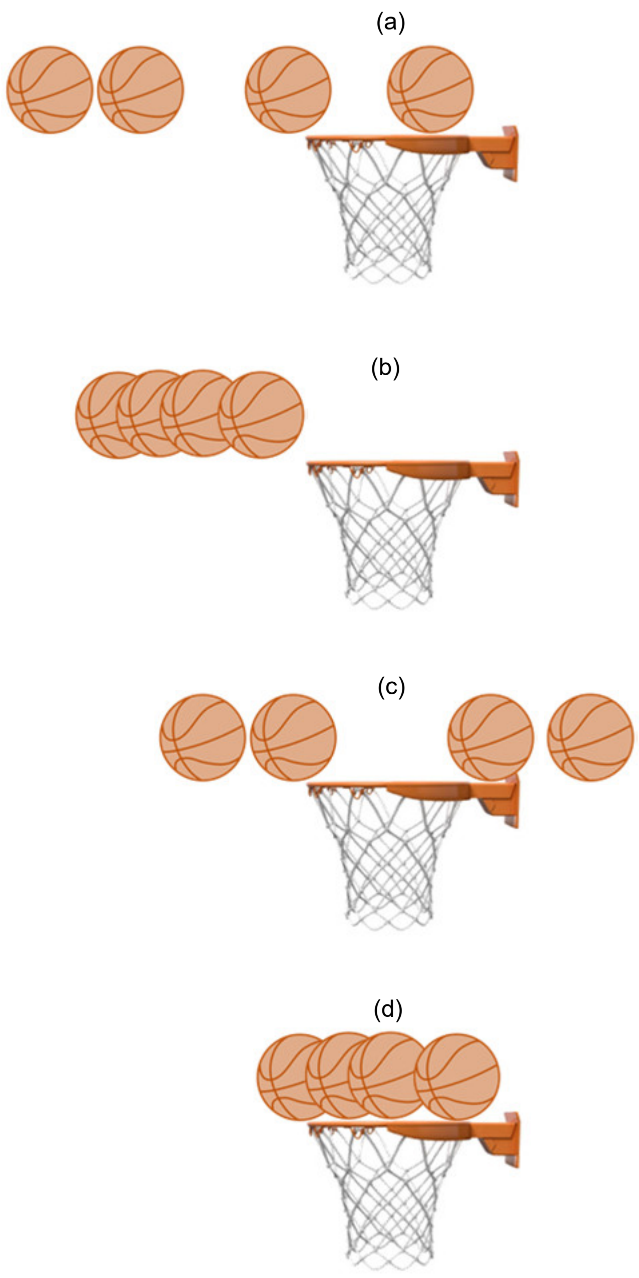


Figure 3. Side view of the hoop (a) mean off target and large scatter (b) mean off target and small scatter (c) mean on target and large scatter (d) mean on target and small scatter

Robust optimization can be used in this case to obtain an initial velocity and a throw angle that increases the success rate while the unwanted inputs exist. It must be noted that t he result for deterministic optimization and robust optimization are not necessarily the same.
The variables that can be adjusted in the process are called design variables. The variables that are out of control or hard to control are called noise variables. In this example, the initial velocity and the throw angle are design variables and wind magnitude and throw height are considered as noise variables.
To perform robust optimization, a proper definition of robustness is required and it can be defined in many different ways. A common approach in robust optimization is to reduce the scatter of output and set the mean on the target. To show why this objective is a good measure for robustness, the basketball free throw is used as an example. Figure 3 schematically shows the side views of the final coordinates of few basketball throws including uncertainties. In Figure 3-a not only the mean is off target but also a large scatter is observed. In Figure 3-b although the scatter is reduced, the mean is still off target. Figure 3-c shows a case in which the mean is on target while scatter is large. Figure 3-d is a desirable situation in which mean is on target and scatter is small. In a desirable situation the throw is both accurate and precise.

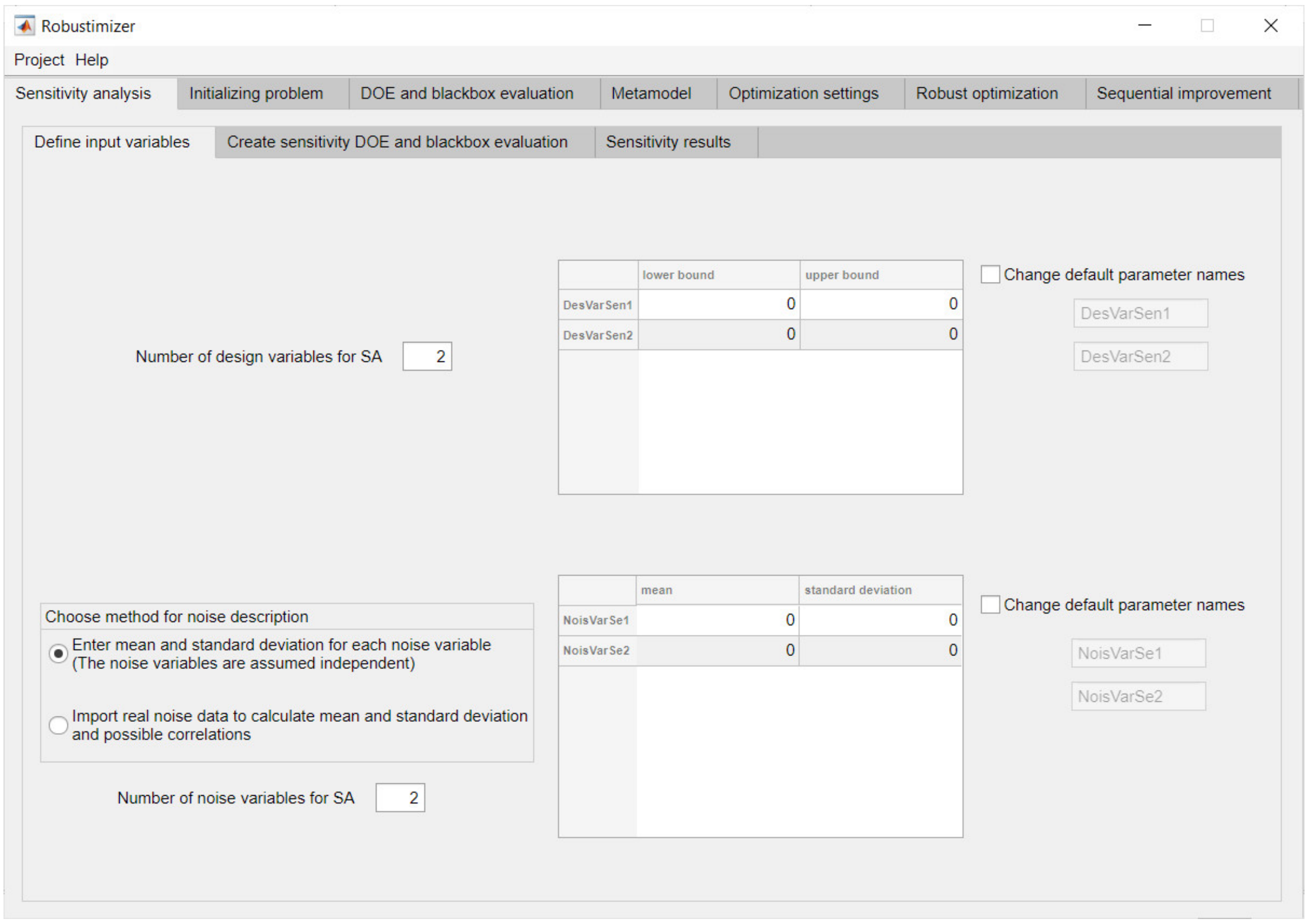
In the next section the features included in Robustimizer is introduced to show the capabilities of this software.

Quick Tips

To start the application, install the package after downloading Robustimizer. Run Robustimizer.exe in the directory that you have installed the program. Note that to open this standalone application, it is required to install MATLAB Runtime. After loading screen, the main window appears. Currently Robustimizer is only available for windows.

2. Running the program

There are seven tabs on top which include the steps to formulate and perform robust optimization. Each tab is explained in detail in the following sections.



Robustimizer

Project Help

Sensitivity analysis | Initializing problem | DOE and blackbox evaluation | Metamodel | Optimization settings | Robust optimization | Sequential improvement

Define input variables | Create sensitivity DOE and blackbox evaluation | Sensitivity results

Number of design variables for SA: 2

	lower bound	upper bound
DesVarSen1	0	0
DesVarSen2	0	0

☐ Change default parameter names

DesVarSen1

DesVarSen2

Choose method for noise description

● Enter mean and standard deviation for each noise variable (The noise variables are assumed independent)

○ Import real noise data to calculate mean and standard deviation and possible correlations

Number of noise variables for SA: 2

	mean	standard deviation
NoisVarSe1	0	0
NoisVarSe2	0	0

☐ Change default parameter names

NoisVarSe1

NoisVarSe2

Figure 4. Main window of Robustimizer

2.1. Sensitivity analysis

Before starting the optimization procedure, the effects of design variables and noise variables on each response can be analyzed to reduce the number of inputs. Since models generally have many inputs, it is recommended to perform sensitivity analysis to determine the importance of the model inputs and to eliminate the variables that are not affecting the results significantly. This tab includes three other tabs which are the steps towards analyzing sensitivities:

2.1.1. Define input variables

In this part the number of design and noise parameters can be defined as shown in Figure 4. For design variables, the number of design variables must be entered. When this field is changed, the size of the respective table will be adjusted and the user enters the lower bound and upper bound for each variable. To change the default names of the variables, select a checkbox and modify the name in that textbox.

Save a project at any time and load it later using the menu items on top left

Project -> Save As & Project -> Load

To define noise variables there are two methods. If there are no measurements or available noise data the first option in *choose method for noise description* panel must be selected. After entering the number of noise variables, mean and standard deviation for each variable is entered. If the second option in *choose method for noise description* panel is selected, the user must provide a *txt* file to calculate the statistics of noise variables and possible correlations between them. The data for each noise variable must be included in one column and the columns must be tab separated for multiple noise variables. If this option is selected, the textbox named *number of noise variables for SA* and *noise data* table will be disabled and the number of noise variables will be assigned from the number of columns in the file. In addition, the statistics of the noise variables will be evaluated from the provided data.

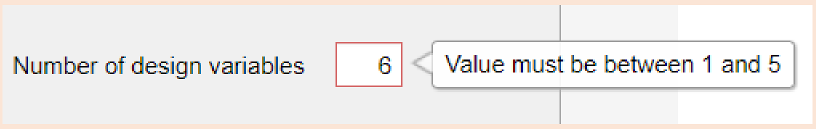
2.1.2. Create sensitivity DOE and blackbox evaluation

In this tab by clicking on *Create screening DOE* (Figure 5) a design of experiments (DOE) is created at which the response must be evaluated. This DOE will be shown in the table and it can be saved in a txt file as tab-separated columns. The user must evaluate the response(s) on these DOE points (e.g. Finite element simulations, mathematical formulations, solving PDEs, etc) and import the results using one txt file. The number of rows for this file should be equal to the number of generated DOE points and the number of columns which are separated by tab must be equal to the number of responses. After importing the file, the responses will be shown in the respective table.

2.1.3. Sensitivity results

In the dropdown menu (Figure 6), the user selects the response for which the sensitivity must be evaluated. After pressing *Run sensitivity analysis* button, the sensitivity plot with the effects of each variable on specific response is plotted.

The maximum number of inputs is limited to 5 design and 5 noise variables for an optimum performance.


Number of design variables 6 Value must be between 1 and 5

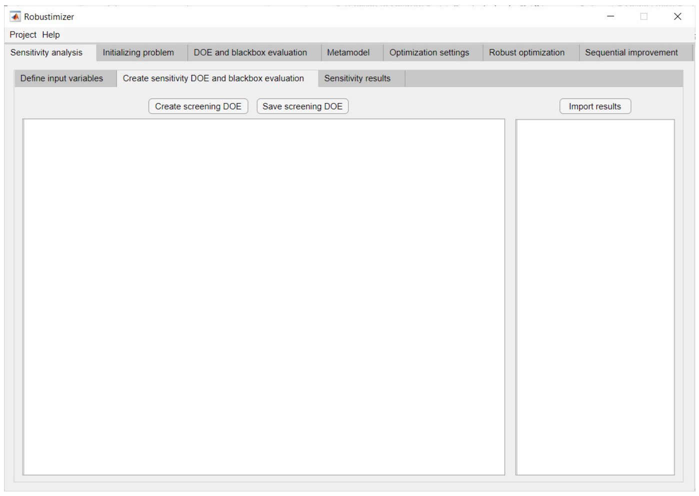


Figure 5. Creating design of experiments for sensitivity analysis and entering the results

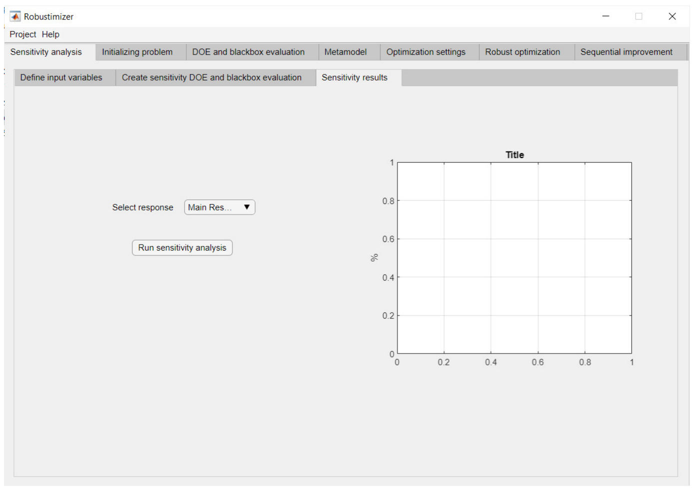


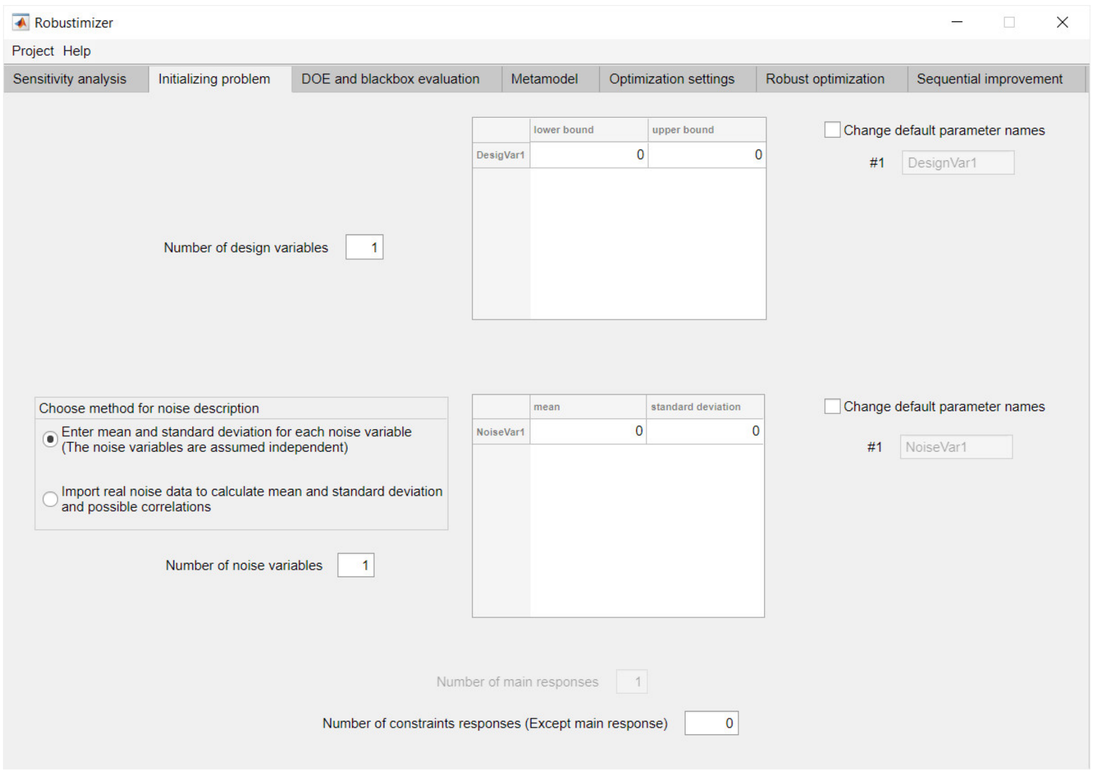
Figure 6 Results of sensitivity analysis

The graphics can be saved or copied by moving the cursor on the graphic and selecting the save option appearing on the graphic. In addition to that, the view can be changed by moving or magnifying the graphics. To reset the view, press home icon.

2.2. Initializing problem

After sensitivity analysis and deciding which variables to consider in the 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* panel 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 column and the columns must be tab separated for multiple noise variables. 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 by 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 Robustimizer only allows one main response. If there are multiple responses, the user is capable of introducing them as constraint responses.



Number of design variables: 1

	lower bound	upper bound
DesigVar1	0	0

☐ Change default parameter names
#1 DesignVar1

Choose method for noise description
(●) Enter mean and standard deviation for each noise variable (The noise variables are assumed independent)
( ) Import real noise data to calculate mean and standard deviation and possible correlations

Number of noise variables: 1

	mean	standard deviation
NoiseVar1	0	0

☐ Change default parameter names
#1 NoiseVar1

Number of main responses: 1

Number of constraints responses (Except main response): 0

Figure 7. Initializing the problem

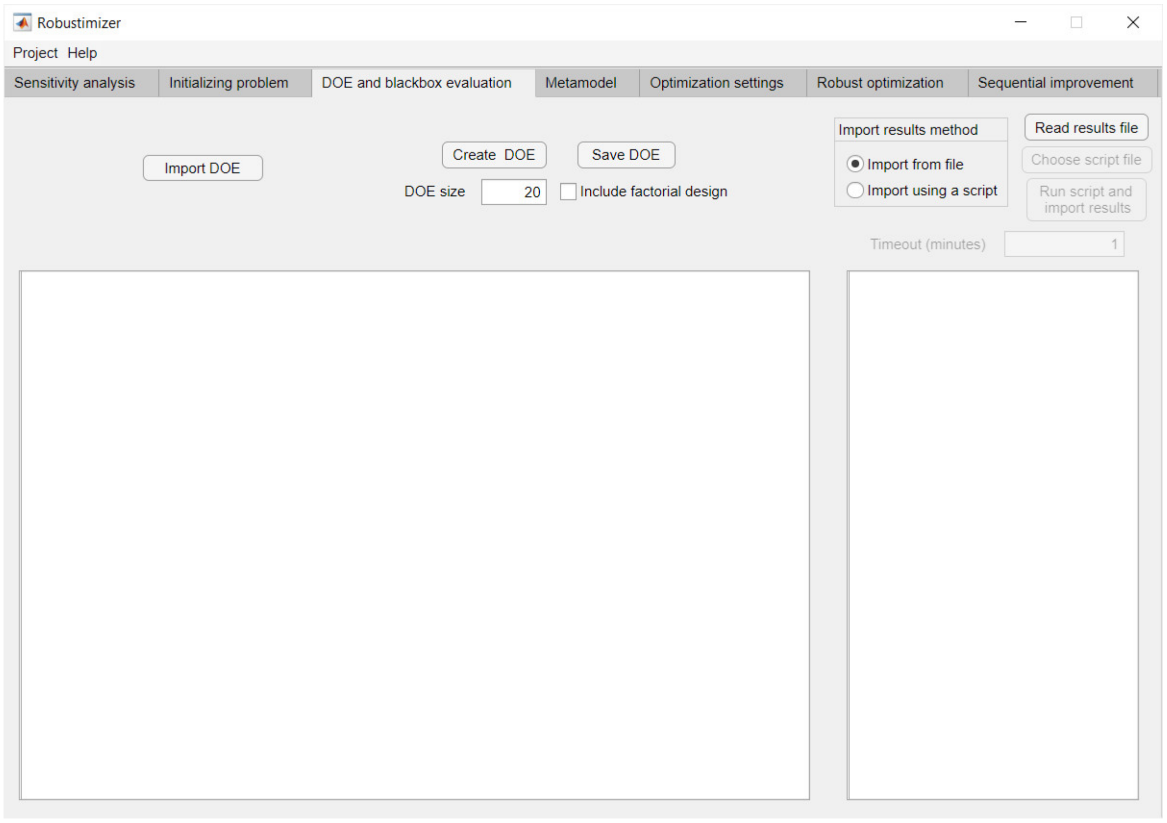
2.3. DOE and blackbox evaluation

In this tab, the user creates a DOE to evaluate the blackbox function. By default Robustimizer 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 8) 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 which is prepared outside Robustimizer. Importing DOE is not possible when choosing the second option of *choose method for noise description* as DOE must include the possible correlations.

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 Robustimizer, save them in a txt file and import the results using one txt file. Alternatively, an executable script file can be introduced to Robustimizer. In this case, Robustimizer will automatically read the results file when it is generated. This executable file should be provided by user (e.g. C++ or Fortran executable) and this script must be able to read *in.txt* file in 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, Robustimizer 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 to the table.

Quick Tips

RobOpt is a single objective robust optimization software tool. However, for multiple objectives the user is capable of including them as implicit reliability constraints.



DOE size: 20
Include factorial design: [ ]
Import results method: [● Import from file] ○ Import using a script
Timeout (minutes): 1

Figure 8. Creating DOE and importing results of blackbox evaluation

2.4. 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.

2.5. 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 10. 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 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 prediction of statistics by considering the skewness of the output [1].

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.

2.6. 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.

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.

2.7. 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 [2] 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 2.3 it is recommended to use the automatic option. In that case by entering the number of sequential improvement steps Robustimizer 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 boxes in the panel for each trial.

For the manual case, the user will obtain a recommended infill point. The user can 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. In that case the result of new evaluation must be added to the end of the existing result file. 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.

[1] O. Nejadseyfi, H.J.M. Geijselaers, A.H. van den Boogaard, (2019). Evaluation and assessment of non-normal output during robust optimization. *Structural and multidisciplinary optimization*, *59*(6), 2063-2076.

[2] D.R. Jones, M. Schonlau, W.J. Welch, (1998). Efficient global optimization of expensive black-box functions. Journal of Global optimization, 13(4), 455-492.

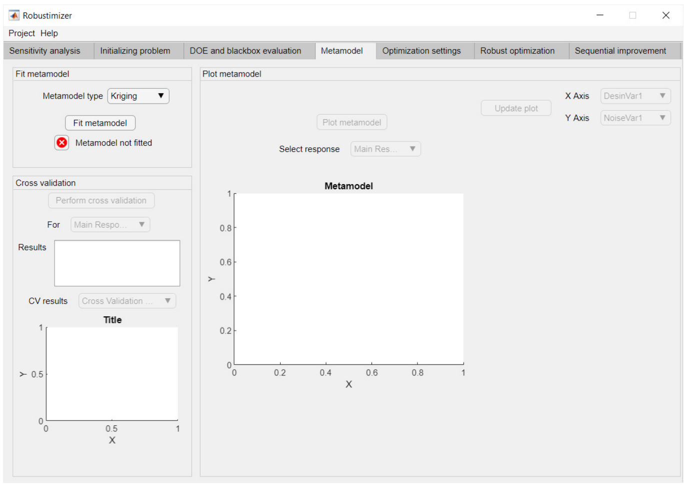


Figure 9. Creating, validating and visualizing the metamodel

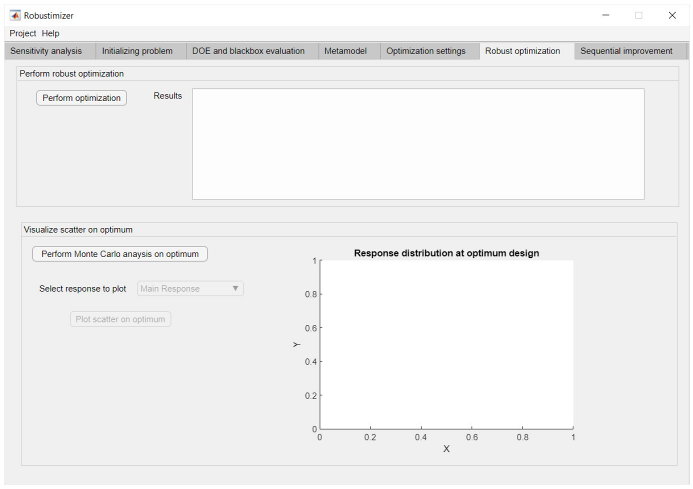

Robustimizer
Project Help
Sensitivity analysis
Initializing problem
DOE and blackbox evaluation
Metamodel
Optimization settings
Robust optimization
Sequential improvement
Perform robust optimization
Perform optimization
Results
Visualize scatter on optimum
Perform Monte Carlo anaysis on optimum
Select response to plot Main Response
Plot scatter on optimum
Response distribution at optimum design
X
Y

Figure 11. Robust optimization tab

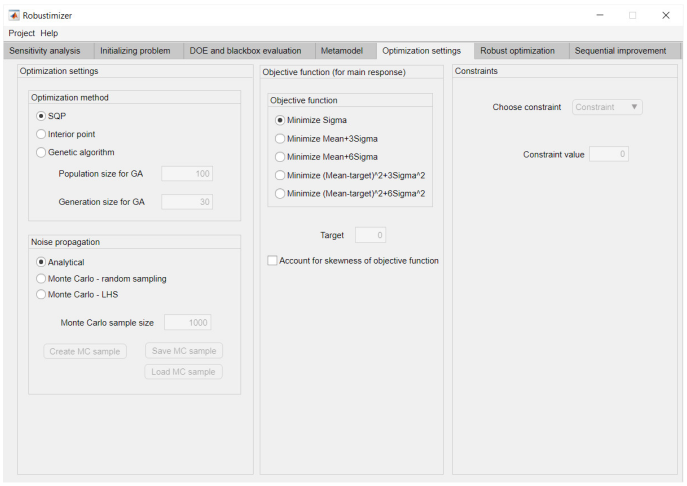


Figure 10. Optimization settings tab

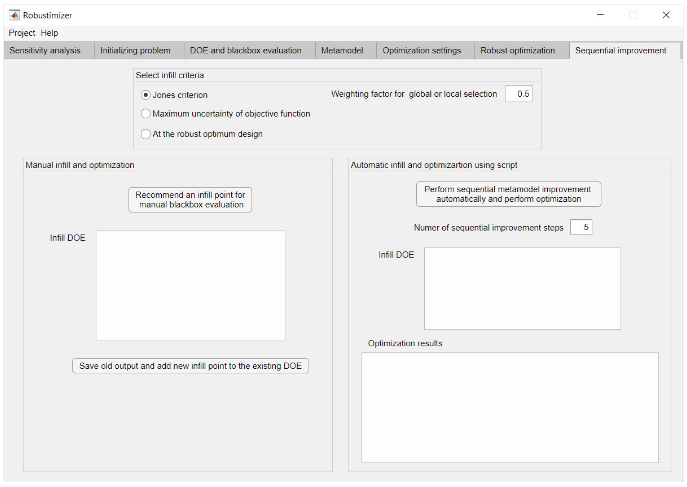

Robustimizer
Project Help
Sensitivity analysis
Initializing problem
DOE and blackbox evaluation
Metamodel
Optimization settings
Robust optimization
Sequential improvement
Select infill criteria
Jones criterion
Weighting factor for global or local selection 0.5
Maximum uncertainty of objective function
At the robust optimum design
Manual infill and optimization
Recommend an infill point for manual blackbox evaluation
Infill DOE
Save old output and add new infill point to the existing DOE
Automatic infill and optimizartion using script
Perform sequential metamodel improvement automatically and perform optimization
Numer of sequential improvement steps 5
Infill DOE
Optimization results

Figure 12. Sequential improvement of the metamodel

For sequential improvement of the metamodel, it is possible to use different criterion for each step of metamodel improvement. This is applicable for both manual and automatic cases of adding infill point(s).

If blackbox function evaluation is computationally expensive, a large number of DOE is not recommended as it will require significant calculation time. For expensive models take advantage of using sequential improvement of the metamodel.