CBDO
Cloud Based Design Optimization
User Guide

Martin Fuchs

University of Vienna
Faculty of Mathematics
Nordbergstr. 15
1090 Wien
Austria
email: martin.fuchs@univie.ac.at

This is the user guide of the graphical user interface (GUI) for Cloud Based Design Optimization (CBDO), cf. [2], implemented in MATLAB.

The GUI has been developed using the concept visualized in Figure 1. The designing expert provides an underlying system model – e.g., given as a black box model – and all currently available uncertainty information on the input variables of the model. The information is processed to generate a cloud that provides a nested collection of regions of relevant scenarios parameterized by a confidence level α. Thus we produce safety constraints for the optimization. The optimization minimizes a certain objective function (e.g., cost, mass) subject to the safety constraints to account for the robustness of the design, and subject to the functional constraints which are represented by the system model. The results of the optimization, i.e., the automatically found optimal design point and the worst-case analysis, are returned to the expert, who is given an interactive possibility to provide additional uncertainty information afterwards and rerun the procedure.

The first two steps of the GUI represent the uncertainty elicitation, that is on the one hand, providing the initially available information (Section

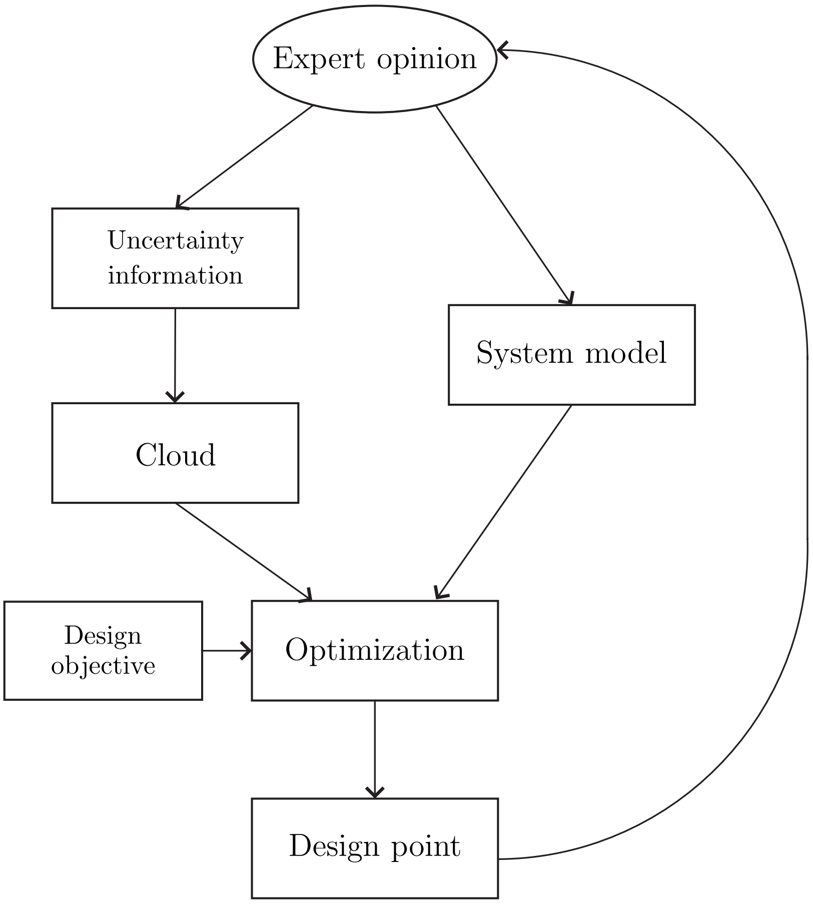


Figure 1: Basic concept.

3), on the other hand, scenario exclusion (Section 4). The next step is the optimization phase (Section 5). Eventually, the user has the possibility of adaptively refining the uncertainty information and rerunning the procedure until satisfaction (Section 6).

To install and set up the GUI follow the instructions in Section 2. Section 1 gives a quickstart guide.

1 Quickstart

1. Download the CBDO archive from `http://www.martin-fuchs.net/cbdo.php`.
2. Extract the archive.
3. Start Matlab.
4. Change to the extracted `cbdo` folder.
5. Run `cvxsetup`.
6. Done! You can now start the GUI with `cbdogui`.

2 Installation

Follow the instruction steps from the Quickstart Section 1. Afterwards the GUI can be started by changing to the `cbdo` folder in Matlab and running `cbdogui`. To prepare the Matlab file containing the underlying system model one has to use the some conventions. Choose a name for the model file, e.g., `model.m`, and create the model file to have the form

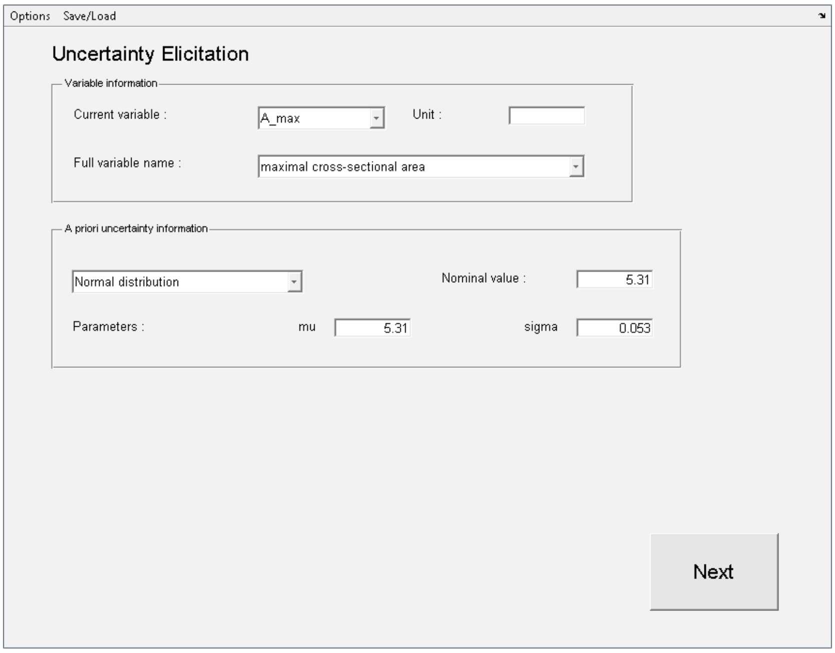
```
[obj,rekvdiff]=model(rekvars,uncvars,designvars,fixedvars)
```

The inputs of the model file have to be declared in a Matlab script file, say, `varinit.m`. Both `model.m` and `varinit.m` have to be created in the `/model` subfolder. This subfolder also contains templates for the inputs declaration file, i.e., `/model/samplevarinit.m`, and for the underlying model file, i.e., `/model/samplemodel.m`. The subfolder `examples` contains further example models and input declaration files, e.g., [3].

3 Uncertainty elicitation

After starting the GUI with `cbdogui` it asks whether to load the last state to the workspace unless it is run for the first time. In the latter case one should first configure the options to set up the model file and inputs declaration file names, and other user-defined parameters after clicking *Options/Edit Options*. Tooltips are given for each option in the GUI to guide the user through the configuration.

Having set up the options one returns to the uncertainty elicitation clicking *Back*. The initially available information specified in the inputs declaration file can be modified choosing a variable's name and specifying its associated marginal CDF, or interval bound, respectively, cf. Figure 2.



Current variable : A_max

Unit :

Full variable name : maximal cross-sectional area

Normal distribution

Nominal value : 5.31

Parameters : mu 5.31 sigma 0.053

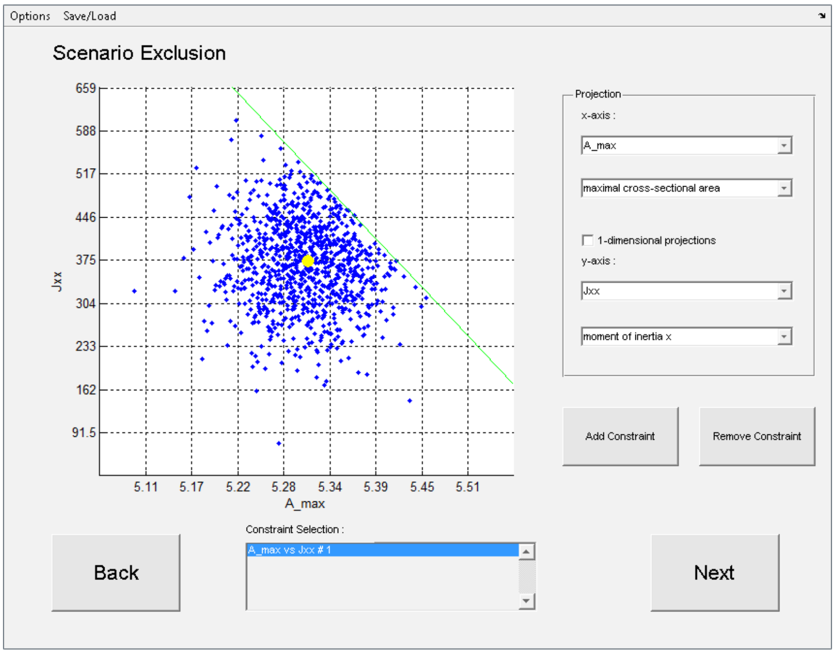
Figure 2: Example for the uncertainty elicitation GUI.

The *Next* button leads to the next step which is scenario exclusion.

4 Scenario exclusion

From the information given in the first step the program generates a sample. The second step enables the user to exclude scenarios by linear constraints as described in [2], [5].

To this end the user selects a 1-dimensional or 2-dimensional projection of the generated sample using the field *Projection* on the right. To add a constraint one hits the *Add constraint* button and defines the linear exclusion



x-axis : A_max
maximal cross-sectional area
□ 1-dimensional projections
y-axis : Jxx
moment of inertia x

Constraint Selection : A_max vs Jxx # 1

Figure 3: Example for scenario exclusion.

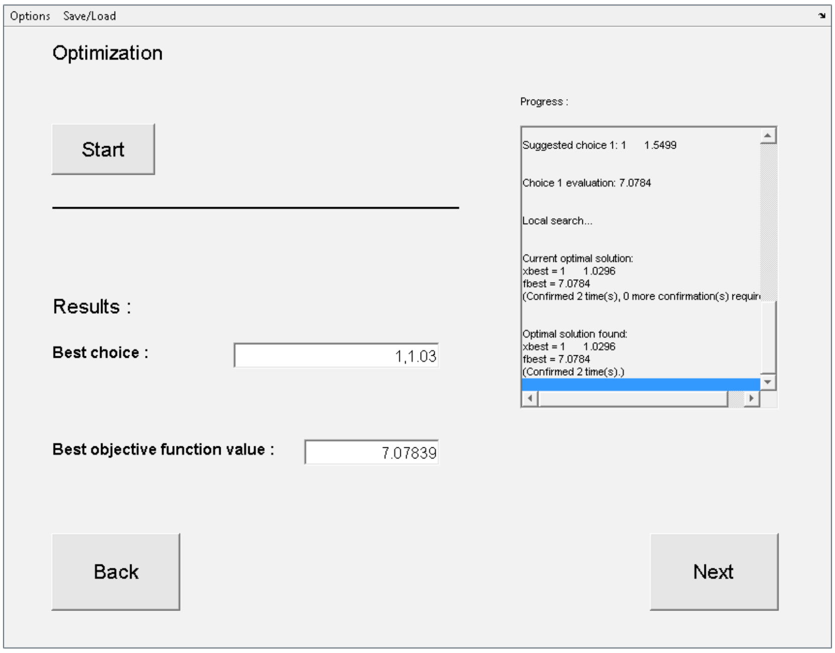
by two clicks into the sample projection on the left. All linear constraints can be selected from the *Constraint Selection* box to revise and possibly remove them via the *Remove constraint* button. Figure 3 shows a possible exclusion in two dimensions.

After the exclusion the *Next* button leads to the optimization phase.

5 Optimization

The *Start* button initiates two computations: potential cloud generation, which includes weight computation if necessary, cf. [5], and optimization, cf. [4]. As a result one gets the optimal design point found by the program, and the associated objective function value, cf. Figure 4. It should be remarked that the workspace of the optimization including all results is stored as .mat files in the `cbdo` directory.

The user now has the possibility for the adaptive analysis of the results. Thus the *Next* button leads back to the uncertainty elicitation which is waiting for refinement.



Results :

Best choice : 1,1.03

Best objective function value : 7.07839

Progress :

```
Suggested choice 1: 1      1.5499

Choice 1 evaluation: 7.0784

Local search...

Current optimal solution:
xbest = 1      1.0296
fbest = 7.0784
(Confirmed 2 time(s), 0 more confirmation(s) requir

Optimal solution found:
xbest = 1      1.0296
fbest = 7.0784
(Confirmed 2 time(s).)
```

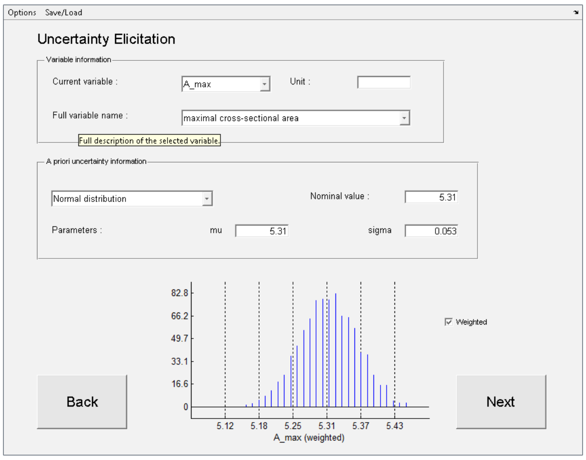
Figure 4: Example for the optimization phase.

6 Adaptive step

The GUI determining the *a priori* information is not modifiable anymore at this stage of the program. Meanwhile, observe that in the lower part of the GUI a histogram illustrates the weighted marginal distributions of the sample, which respect both the *a priori* information and the scenario exclusion, cf. Figure 5.

Hitting the *Next* button makes the scenario exclusion appear again and enables the *a posteriori* adaption of the uncertainty information. For example, the expert who uses the program can consider the worst-case scenario (highlighted with a red dot) to be too pessimistic and exclude it, cf. Figure 6.

The *Next* button leads to the optimization phase again and the user can rerun the procedure until satisfaction.



Uncertainty Elicitation

Variable information

Current variable : A_max

Unit :

Full variable name : maximal cross-sectional area

A priori uncertainty information

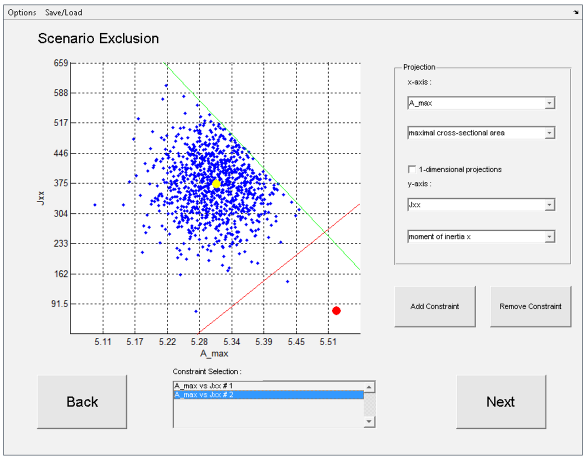
Normal distribution

Nominal value : 5.31

Parameters : mu 5.31 sigma 0.053

☑ Weighted

Figure 5: Example for uncertainty elicitation in the adaptive step.



Scenario Exclusion

Projection

x-axis : A_max

maximal cross-sectional area

□ 1-dimensional projections

y-axis : Jxx

moment of inertia x

Constraint Selection :

A_max vs Jxx # 1

A_max vs Jxx # 2

Figure 6: Example for *a posteriori* scenario exclusion.

References

[1] CBDO. Cloud Based Design Optimization webpage `http://www.martin-fuchs.net/cbdo.php`, 2008.

[2] M. Fuchs. *Uncertainty modeling in higher dimensions: Towards robust design optimization.* PhD thesis, University of Vienna, Faculty of Mathematics, 2008.

[3] M. Fuchs, D. Girimonte, D. Izzo, and A. Neumaier. *Robust Intelligent Systems*, chapter Robust and automated space system design. Springer, 2008. In press, preprint available on-line at: `http://www.martin-fuchs.net/publications.php`.

[4] M. Fuchs and A. Neumaier. Autonomous robust design optimization with potential clouds. *International Journal of Reliability and Safety, Special Issue on Reliable Engineering Computing*, 2008. Accepted, preprint available on-line at: `http://www.martin-fuchs.net/publications.php`.

[5] M. Fuchs and A. Neumaier. Potential based clouds in robust design optimization. *Journal of Statistical Theory and Practice, Special Issue on Imprecision*, 2008. Accepted, preprint available on-line at: `http://www.martin-fuchs.net/publications.php`.