**Student Race Car**

The objective of this demo is to provide insight into the typical modeling steps related to the size optimization of a student racing car subjected to bending-, shear-, and torsion loads.

Please note, that the load cases, geometry of the frame, the dimensions of the 1D element cross sections, material parameters etc. are just conceptual.



In case you are new to HyperWorks we recommend viewing the video series: HyperWorks Starter Kit first. The StarterKit vides, provide a basic insight into the work flow and philosophy of HyperWorks and is available for free in the Academic Blog:

<http://www.altairuniversity.com/2011/09/22/hyperworks-starterkit-video-series/>

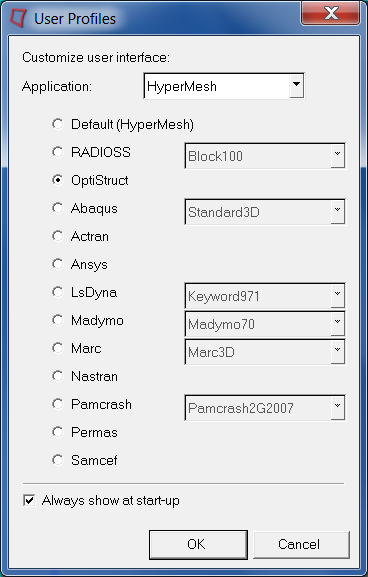
Also this tutorial is available (in a slightly modified version) as a demo video at:

http://www.altairuniversity.com/2012/03/09/simulation-driven-design-of-a-student-race-car-free-video-series/

The following working steps are briefly shown:

* Size Optimization set-up
* Size Optimization Run
* Postprocessing

Right at the beginning of the demo the User Profile is set to OptiStruct



**Linear Static Analysis – Base Design**

**Summary Base Design:**

Total mass: 42.1 kg

ro= 12.5 mm

ri=10.5 mm (Note, the initial radii used in the base design are just a first guess).

|  |  |  |
| --- | --- | --- |
| **Load Step** | **Max. Displacement (mm)** | **Node** |
| Bending | 1.45 | 2764 |
| Shear | 2.03 | 2678 |
| Torsion | 6.29 | 2085 |

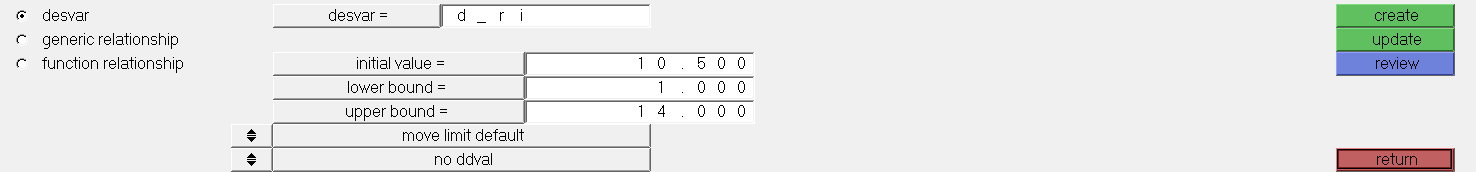
**Size Optimization**

Let’s assume that all the displacements related to the 3 load steps are *acceptable*. In this case we may say: Job done. Or, we may try to find an even lighter designed frame (by employing the same material). This leads us to size optimizing of the cross-sectional properties ro and ri of the tubes cross section. The two parameters ro and ri represent the design variables, and will be varied within the following range:

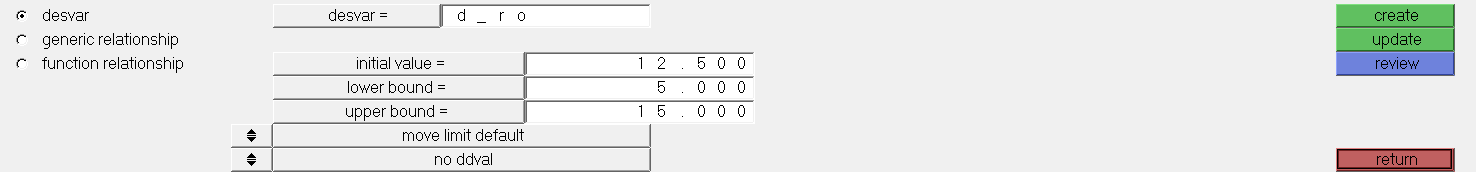
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Lower bound**  **(mm)** | **Upper bound**  **(mm)** | **Initial value**  **(mm)** |
| **ro** | 5 | 15 | 12.5 |
| **ri** | 1 | 14 | 10.5 |

As stated before, the parameters which will be changed is the tubes outer ro and inner radius ri which in turn affect the elements cross-sectional properties i.e. area, moments of inertia, and torsional stiffness.

The design variables are defined via the main menu🡪Optimization🡪Create🡪Size Desvars.

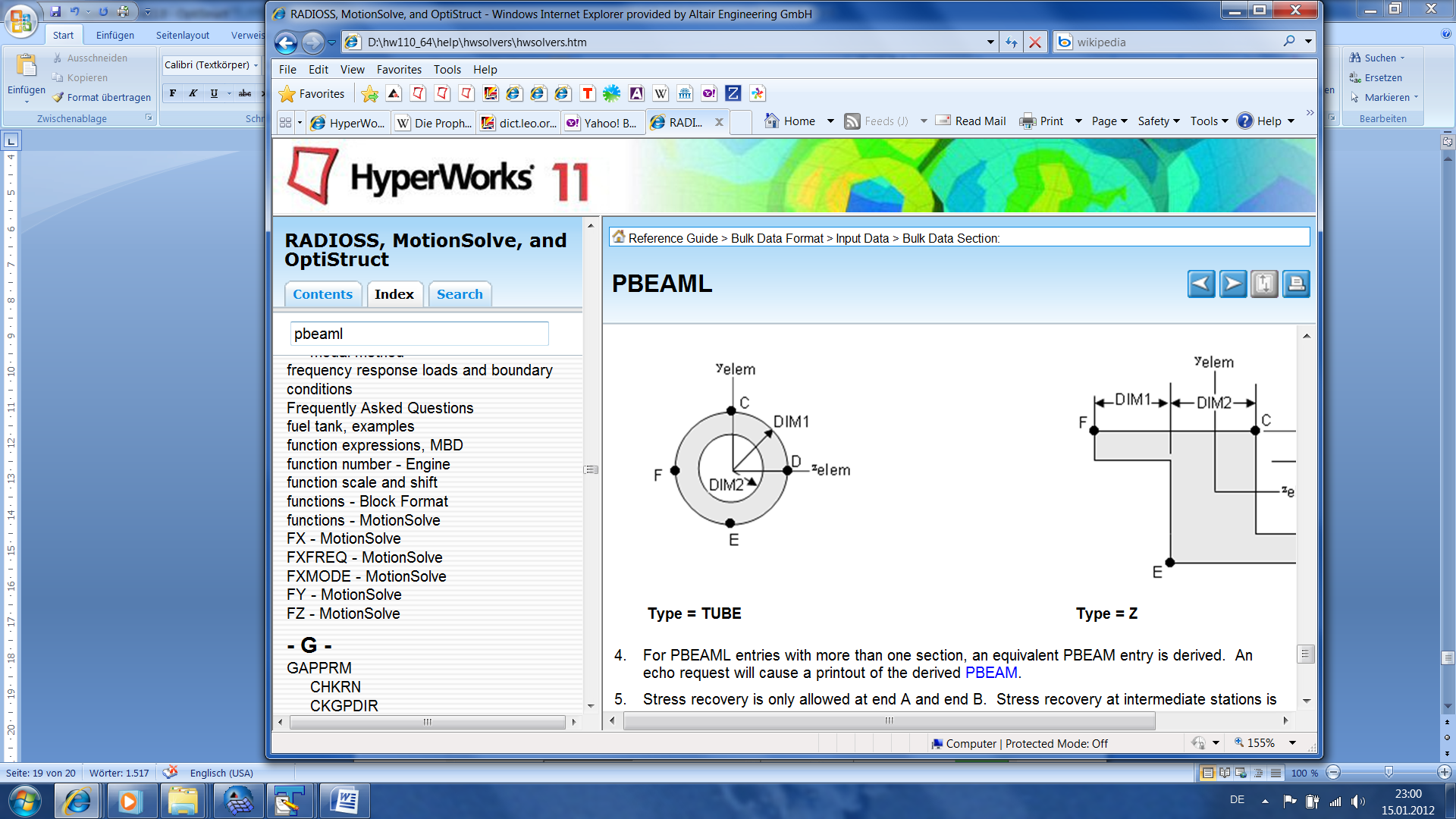


The inner radius ri may vary between its lower bound 1 mm up to its upper bound of 14 mm.



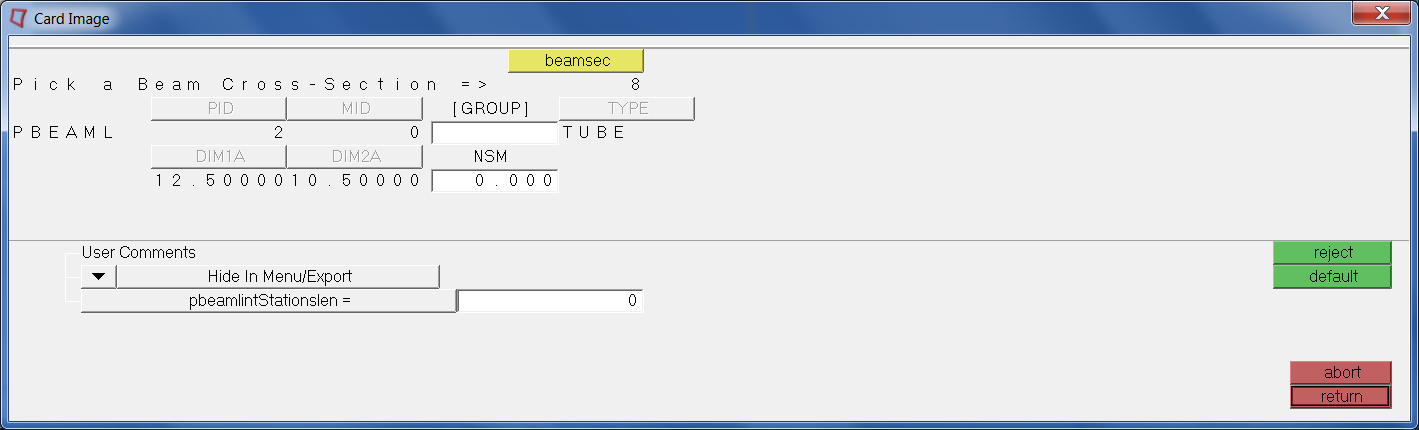
The outer radius ro is allowed to vary from 5 mm up to 15 mm.

Of course, the definition of the design variables is not completed yet as we just specified a name and a range of possible values. In other words, we still need to relate the design variables d\_ri and d\_ro to the cross-sectional properties of the 1D elements. These properties are given as DIM1 and DIM2.



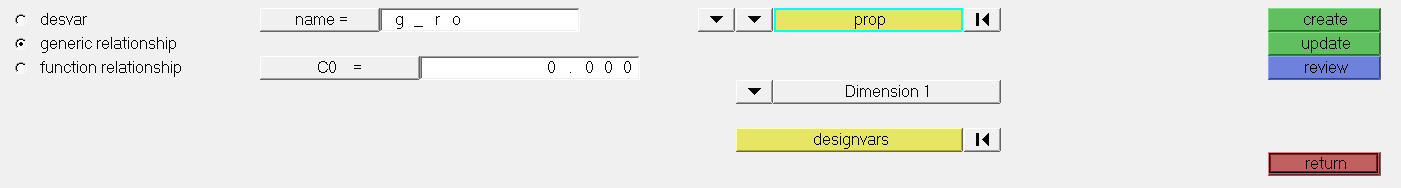
Note that DIM1 corresponds to ro and DIM2 to ri.

Also recall: DIM1 (at node A) was used earlier already – just view the property collector again:



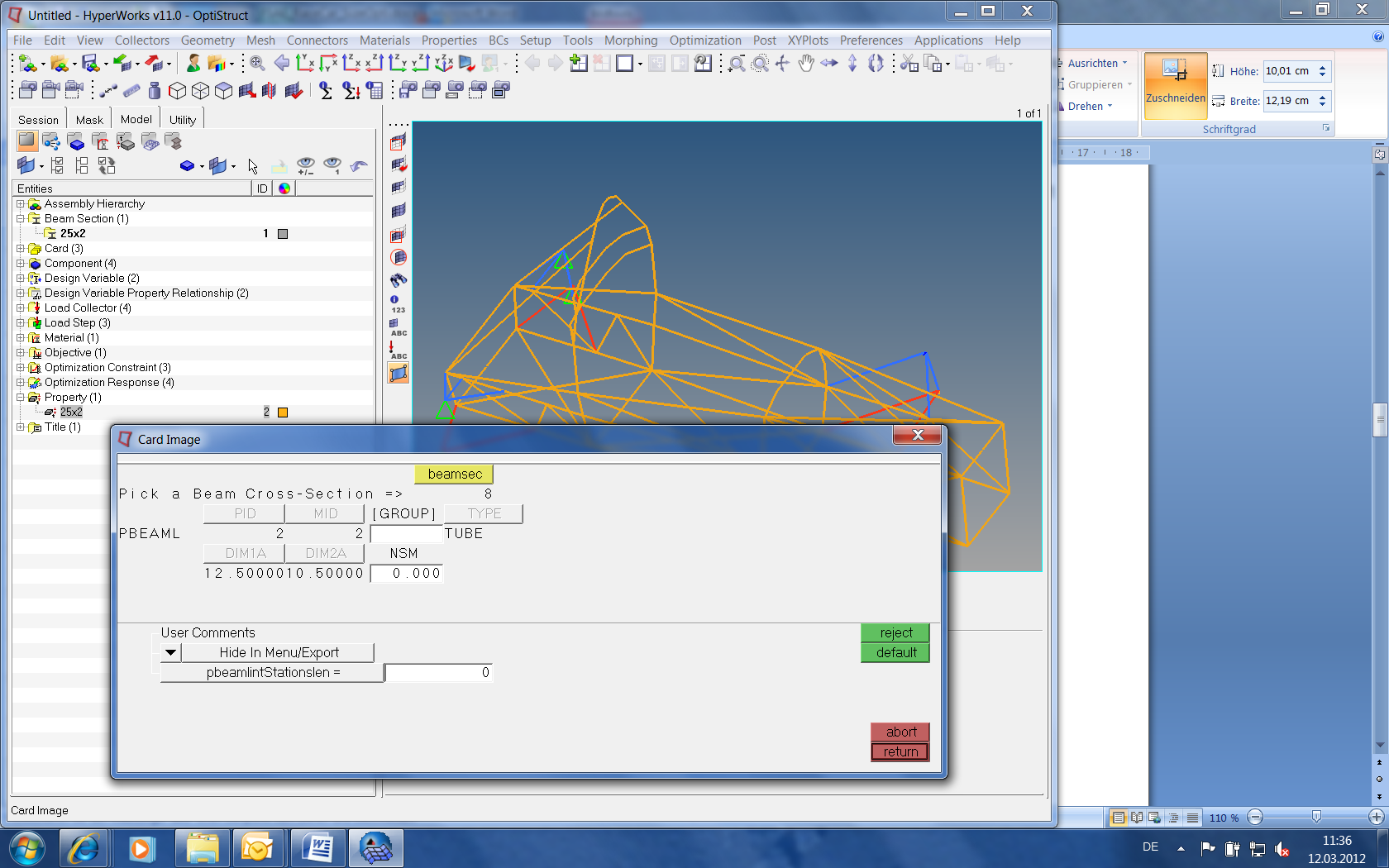
To”link” the design variables to DIM1 and DIM2 to d\_ro and d\_ri, we need to activate the generic relationship option.

At 🡪Optimization🡪Create🡪Desvar Relationship🡪Generic

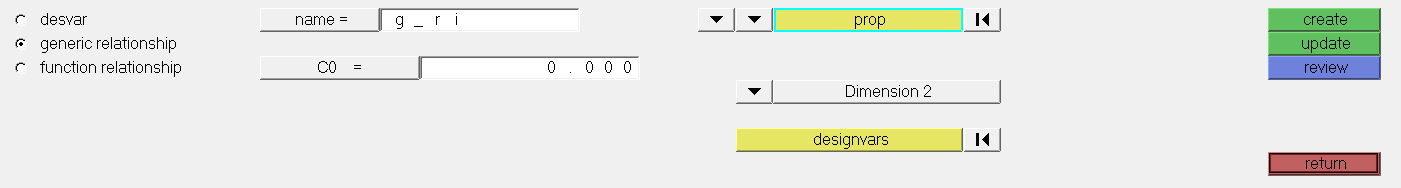


In here the design variable d\_ro becomes related to Dimension 1 (DIM1) which in turn corresponds to the intrinsic definition of the outer radius of the tube.

(designvars = d\_ro); prop = property collector, here with ID =2)



The same steps apply to the inner radius which is defined as Dimension 2.



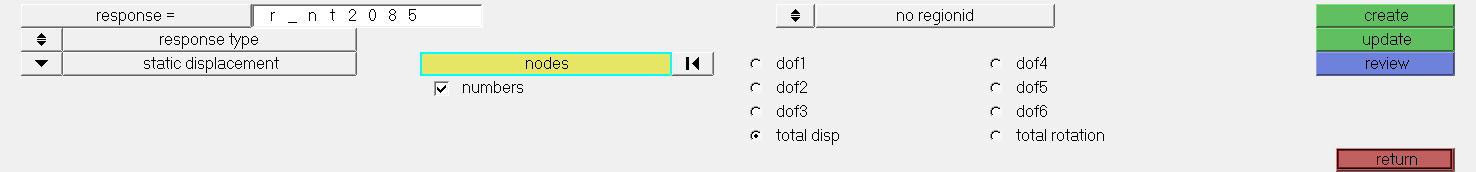
This completes the definition of the Design Variable Property Relationship (and thus the design variables)



While during the size optimization these design parameters are varied, we need to observe the mass of the frame (=objective is lighter frame) and the nodal displacements (i.e. we want to know how the changes of the radii will affect the displacements at certain nodes of the structure).

This leads us to the definition of optimization responses:

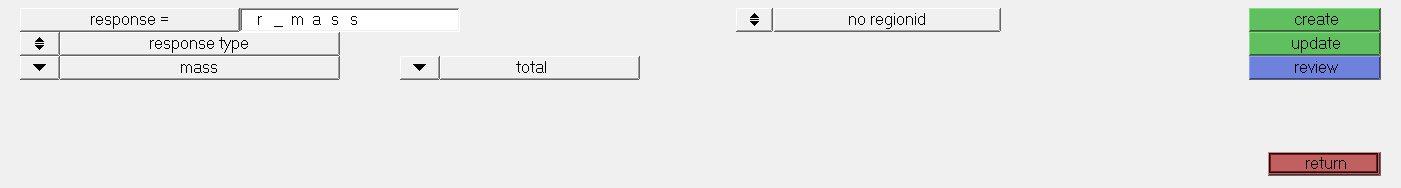
Main Menu🡪Optimization🡪Create🡪Responses



Next we define the response regarding the total displacement of node 2085. Type in a name used to reference this response, then specify the node ID, and define whether a single translational displacement component or its magnitude should be “tracked”.

The responses for nodes 2678 and 2764 are defined just the same way.

As the objective of the size optimization is to find a lighter design, the mass needs to tracked during the optimization. Just change the “response type” to mass.

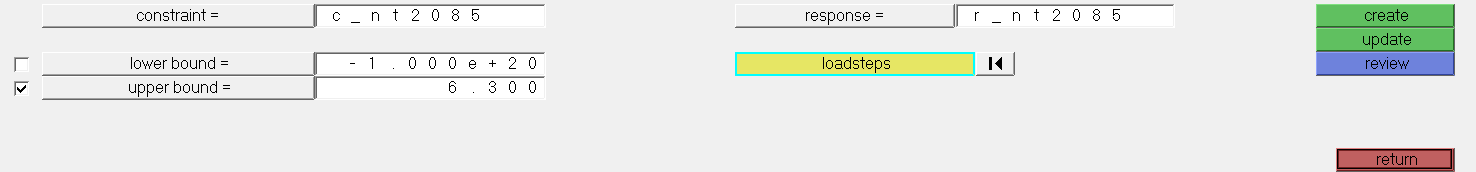


Note: If we ask the optimizer to minimize the frames’ mass the result is quite predictable – Any ideas already?

The outer radius will be set to its lower boundary whereas the inner radius will be at its upper bound - which is not exactly the solution we are looking for.

Thus, we need to define optimization constraints. As design constraints the displacements from the base design, as listed in the table above will be applied.

Main Menu🡪Optimization🡪Create🡪Constraints



The maximum allowed displacement (magnitude) at node 2085 (regarding the load step torsion) is 6.3 mm. The displacement constraints regarding the loadsteps Bending and Shear are defined in the same way.

And finally, we need to define the objective of the size optimization study.

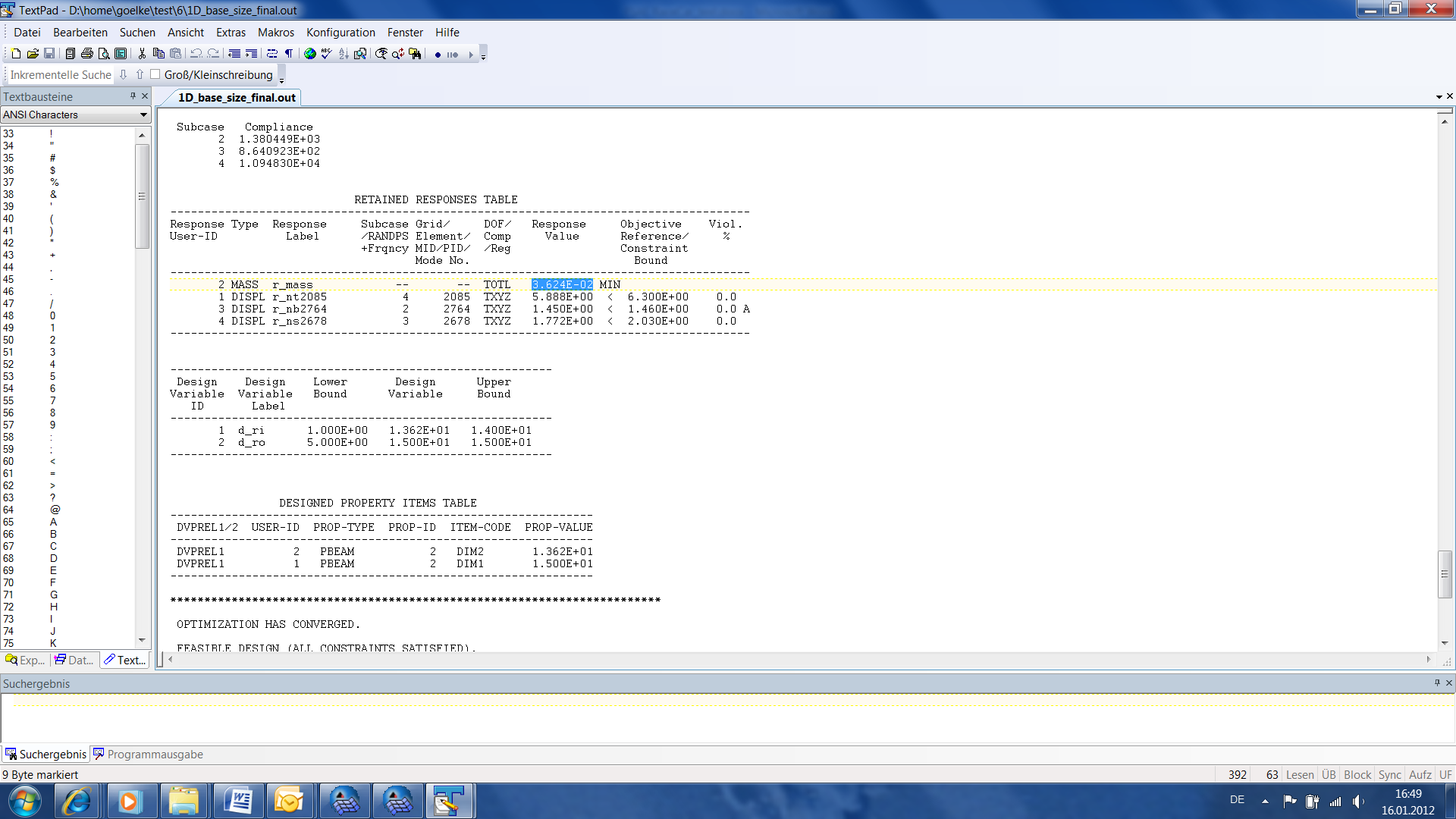
Main Menu🡪Optimization🡪Create🡪Objective



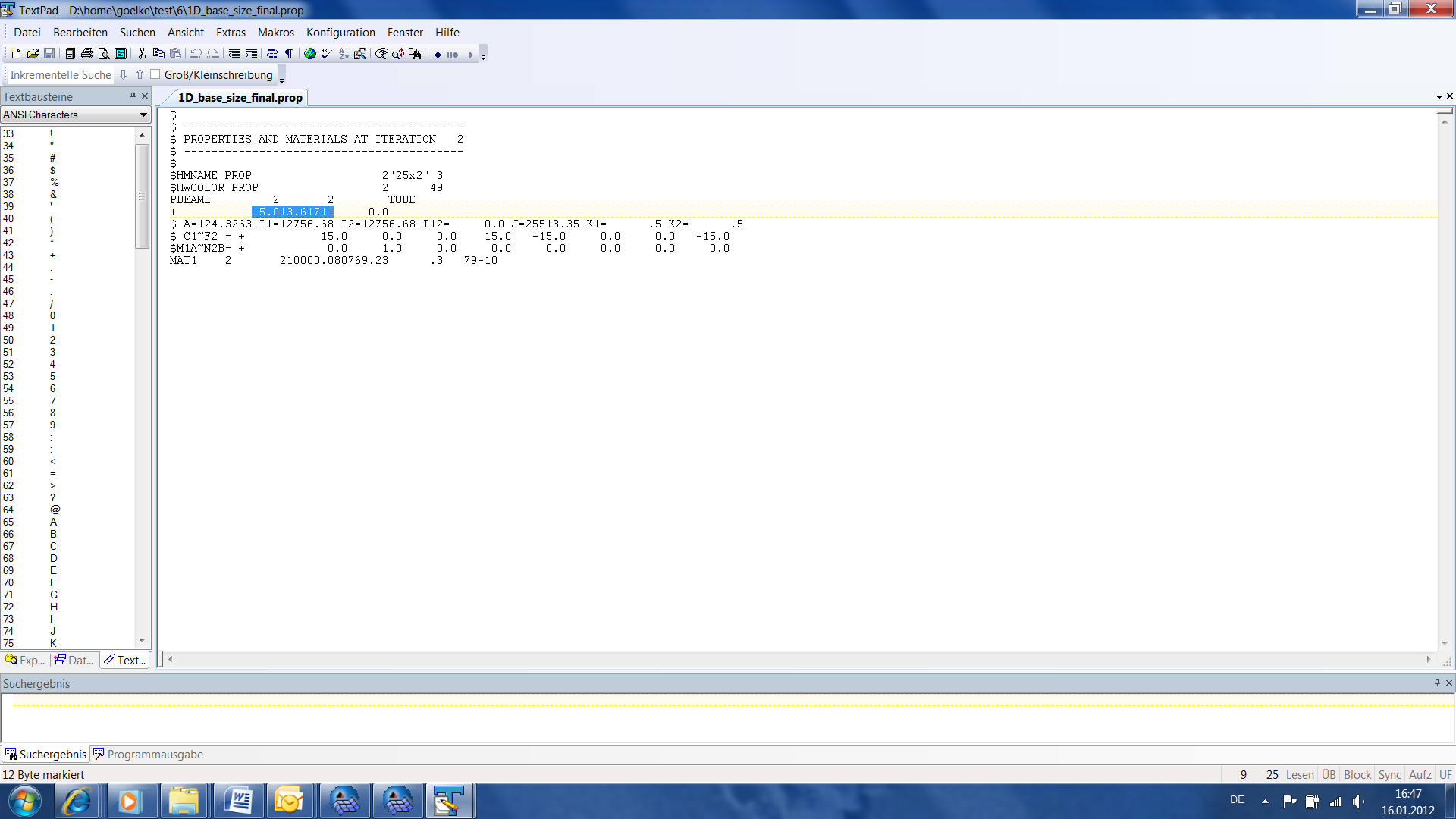
That completes the size optimization set up.

**Postprocessing**

In the \*.out file we learn that the mass of the frame is now down at 36.24 kg which corresponds to weight savings of 16% (base design: 42.12 kg) while the displacements match the values of the base design.



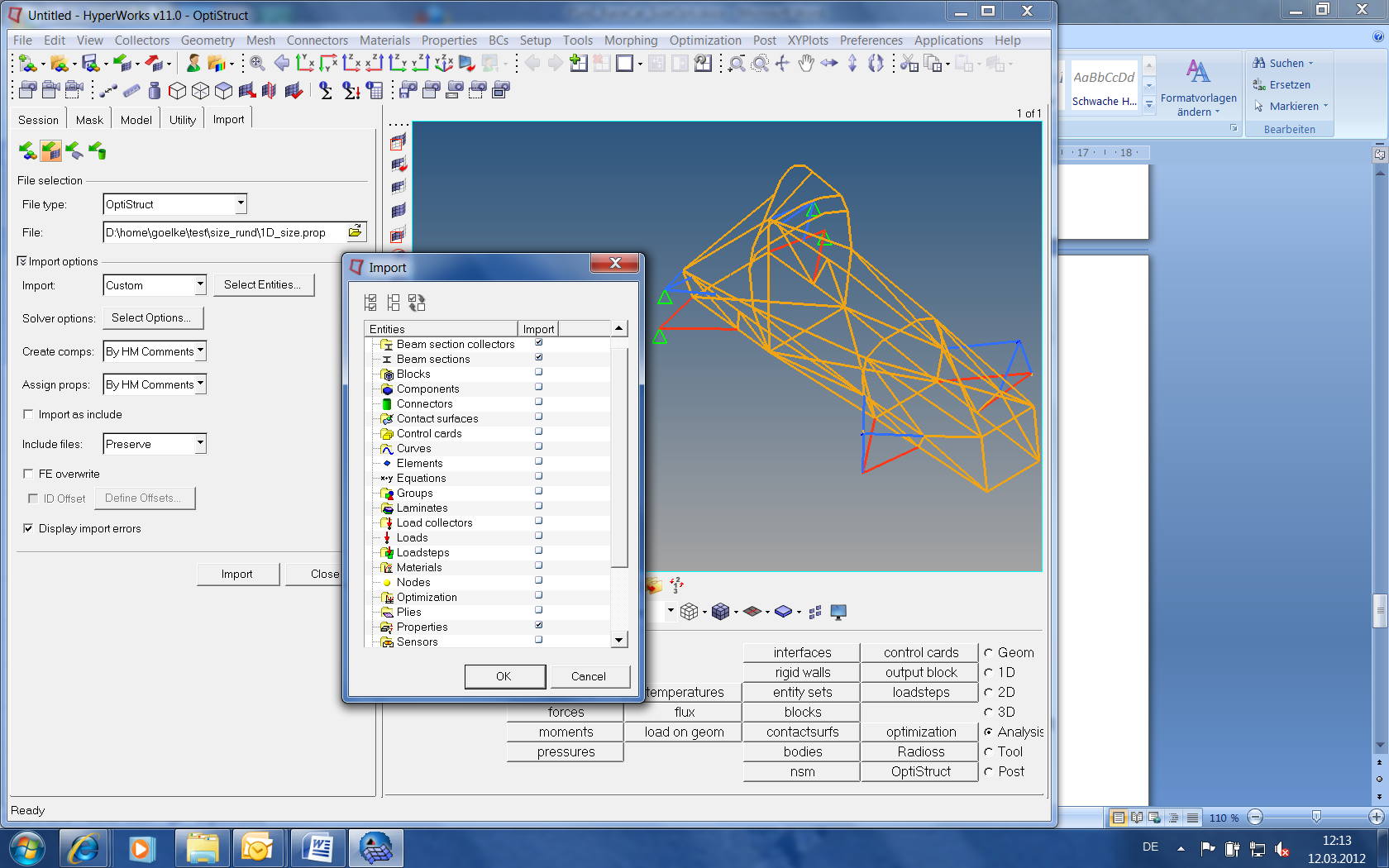
In the working directory we need to open the file with the extension\*.prop in order to view the optimized PBEAML properties



The final dimensions (which are continuous) could be rounded off to:

* Outer tube radius (DIM1): 15 mm (base value: 12.5 mm)
* Inner tube radius (DIM2): 13.6 mm (base value 10.5 mm)

This property file can be read into HyperMesh (as a solver deck) with the settings: Import 🡪 Custom (then specify to import beamsection, beamsection collector and property). Eventually, check the model and assign the property collector again …



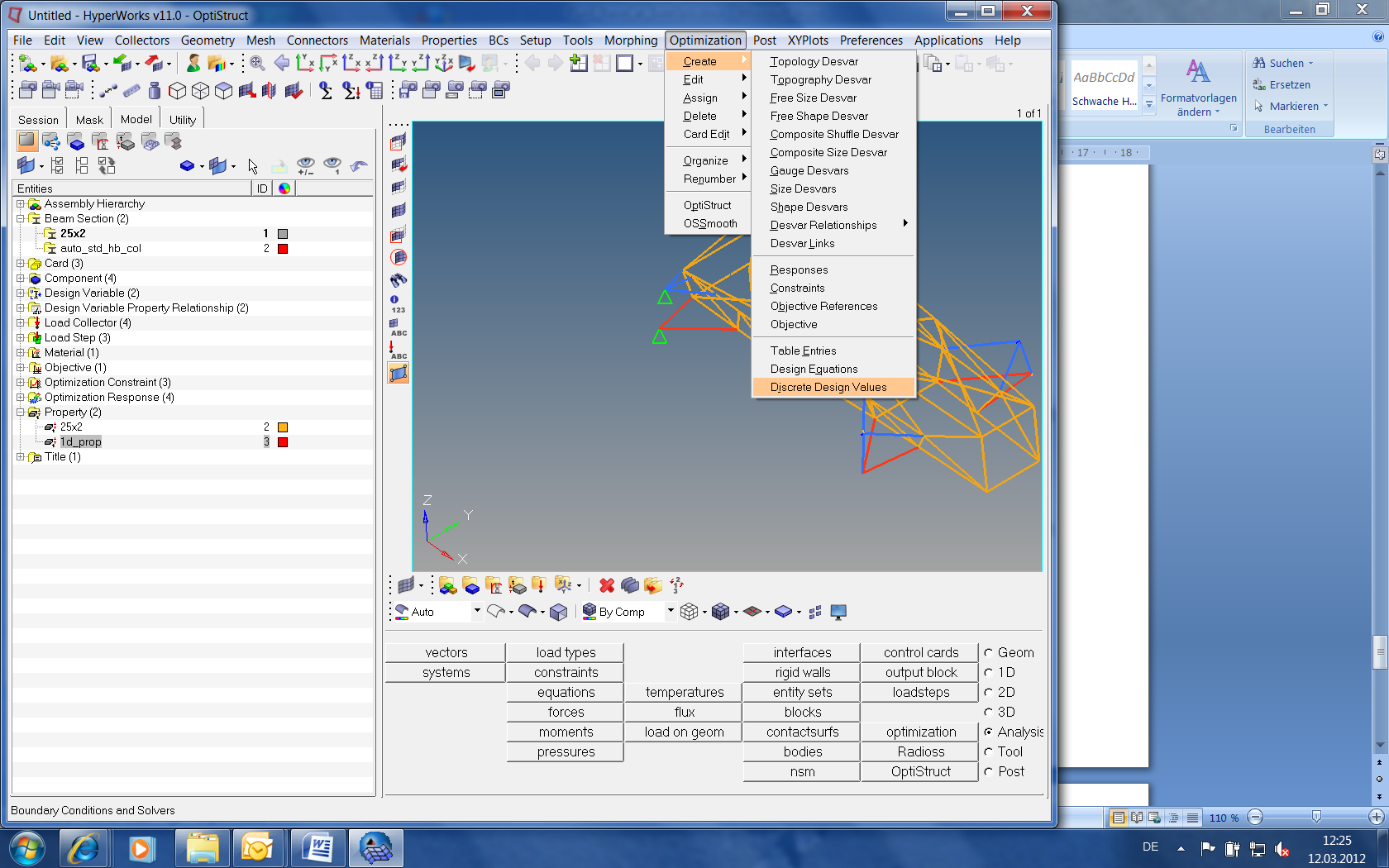
\*.prop

In the above shown size optimization study the displacements of the base design have been “artificially” used as design constraints to better compare base design and optimized design. Naturally, the design constraints may be more relaxed or tighter depending on your requirements.

**Discrete Size Optimization**

Due to the simple set up of the size optimization, the optimization results for ro and ri had to be manually rounded off. Of course, we can run the same optimization study aiming for discrete values for ro and ri.

In addition to the optimization set-up from above one fundamental addition is required: We need to define discrete design values



🡪Optimization🡪Create🡪Discrete Design Values

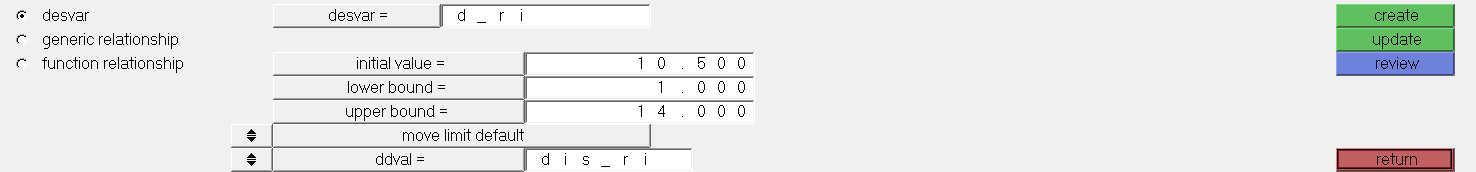


Assign a name, specify lower and upper bound (here the outer radius varies between 5 mm and 15 mm) and a “suited” increment of e.g. 0.1 mm.

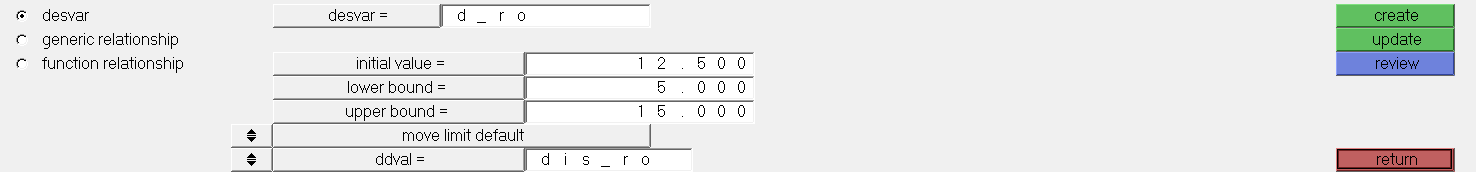
The same process is applied to the design values of the inner radius



Next, the discrete design values are assigned/linked to the design variables, respectively. This is accomplished in the “desvar” panel: select the design variable of interest (=desvar) and activate the “ddval” option in here. In this subpanel the discrete design value (as defined before) is referenced. Don’t forget to click on “update”.



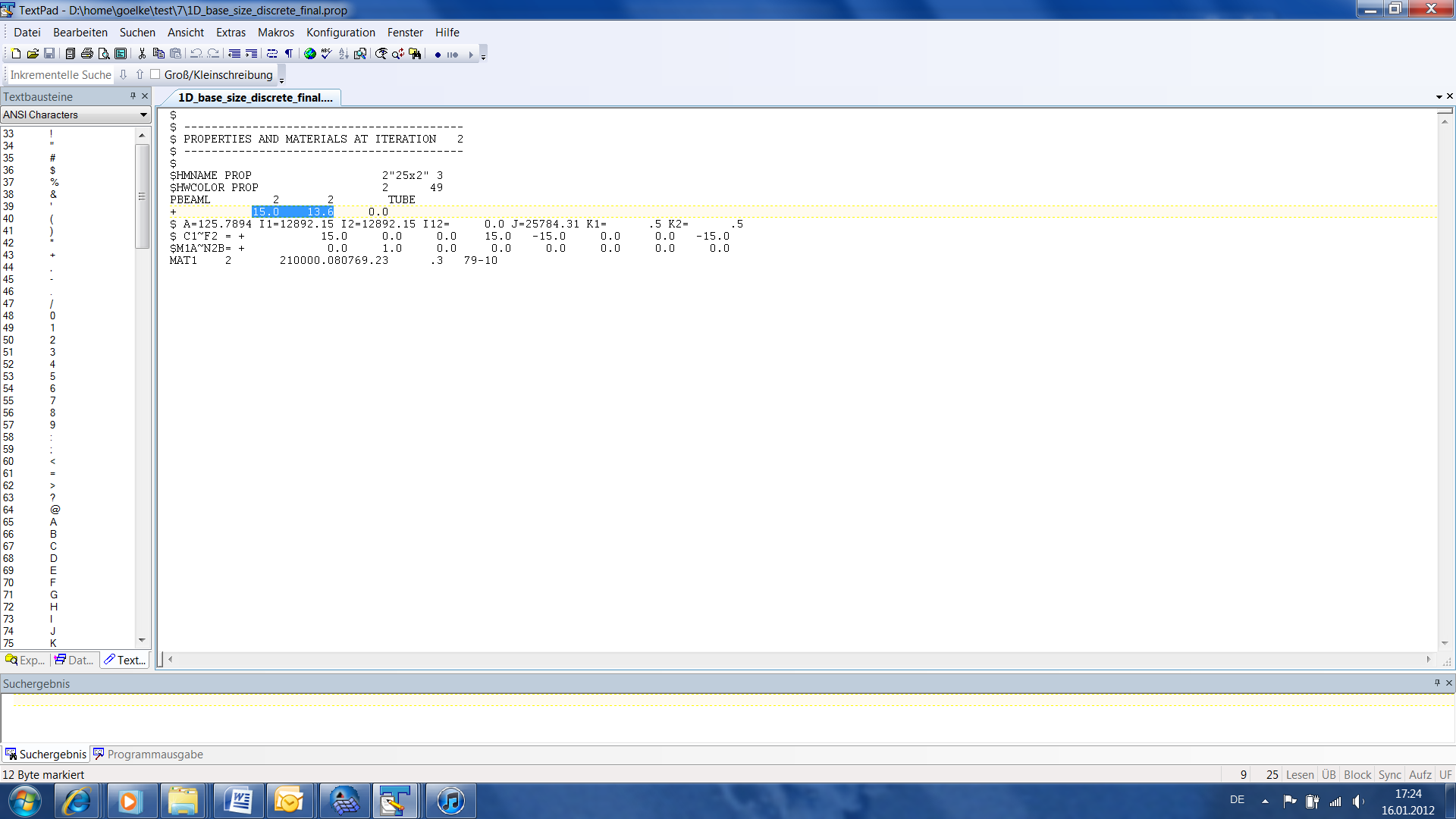
The same working steps are then applied with respect to the design variable of the outer radius.



That is all it takes to create a discrete design variable.

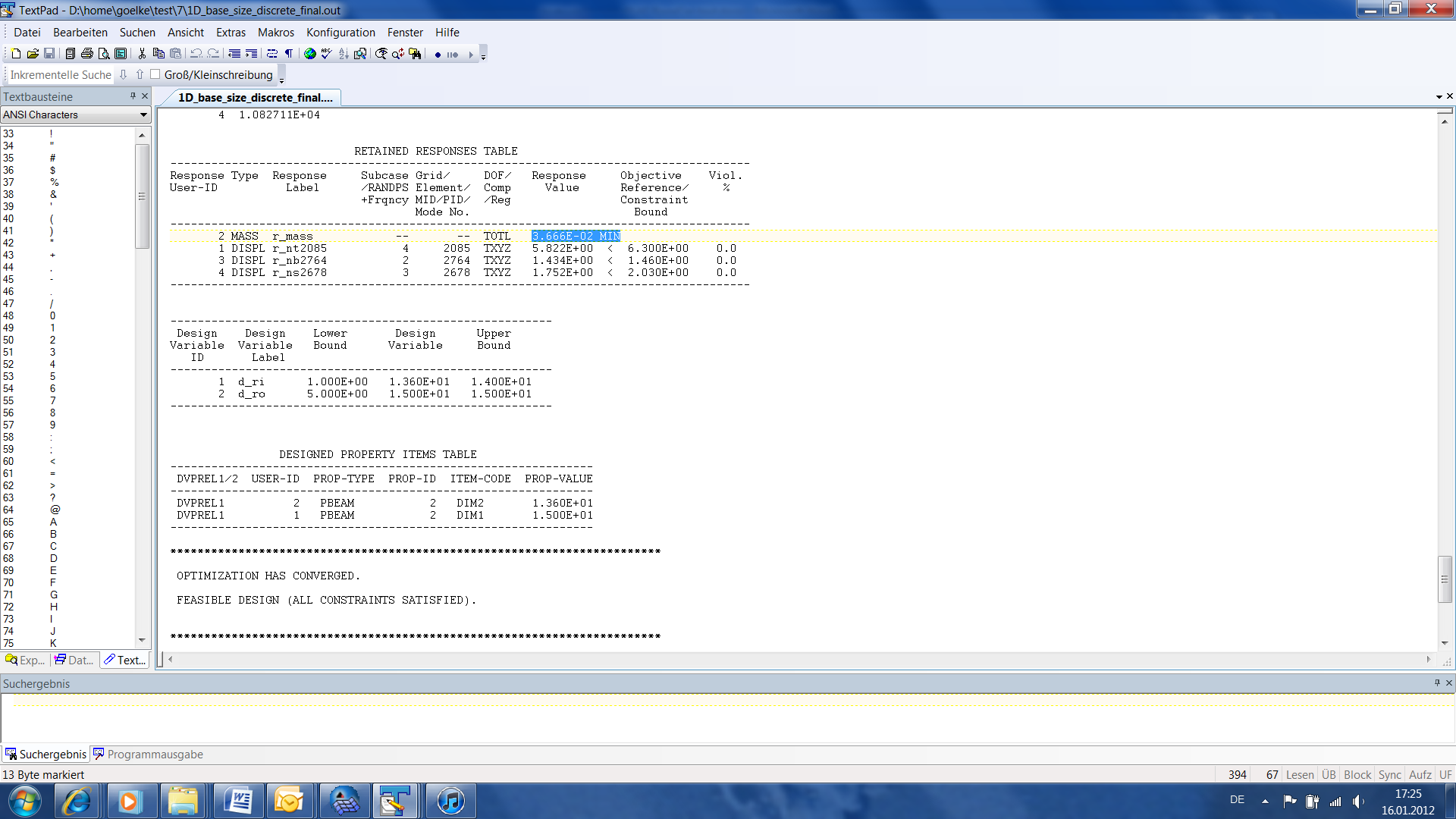
Time to start the size optimization over.

In this example the \*.prop file lists the following information:



Outer radius (DIM1) = 15 mm, inner radius (DIM2) = 13.6 mm.

From the outfile we can readily extract the information about the mass of the frame, which is slightly increased to 36.66 kg.



Of course in a real model you have many different cross-sectional properties. Some parts of the frame may even not be part of the design space etc. So there is lots of space for improvements.

At the bottom line, this simple tutorial should help you getting started.

Good luck and enjoy CAE with HyperWorks