MAKING MEMS DESIGN EASY

www.i-rom.de

USER MANUAL

Version 2023-2

Table of Contents

Part A - MODELBUILDER Release Notes

The i-ROM MODELBUILDER is a software product for modeling and simulation of Micro-Electro-Mechanical-Systems (MEMS). The software is tailored for the design of electro-mechanical transducers such as accelerometers, angular rate sensors, micro actuators (e.g. micro mirrors) or similar products. In particular, the MODELBUILDER is focused on MEMS where the mechanical domain consists of seismic masses which are connected by beam-like suspension springs to anchors and the electrostatic domain consists of comb-cells or plate-like capacitors for sensing and actuation. The current release supports one functional layer with a constant thickness. Further layers are used for bottom or top plate capacitors.

The MODELBUILDER represents the mechanical domain by rigid-body masses which are linked by Timoshenko beam elements to other masses or anchors. Rigid bodies of seismic masses are assembled from primitives and Boolean operations. Perforation patterns of circular and rectangular shape can either be assigned in Cartesian or in cylinder coordinates.

Timoshenko beams are defined by connecting points and linking spring elements. Multiple springs of arbitrary orientation can share the same connecting point. Curved springs as well as springs with uniform or varying beam width are supported by the MODELBUILDER. The mechanical domain is considered linear.

Corner fillets of different radiuses, mask undercut, and orientation dependent etch sidewall slope values can be assigned to all model components. A non-zero sidewall slope allows for modeling of beams, comb-cells and masses with trapezoidal cross-sections what is important to evaluate crosstalk of accelerometers or quadrature effects of angular rate sensors.

The electrostatic domain is represented by a series of comb-cell library elements and plate-like capacitances of arbitrary shape. Comb-cells and plate-like capacitances consider all motion components of seismic masses to calculate the capacitance-stroke-functions. Fringing fields are ignored, and the tilt angle should be $< 20°$ since trigonometric functions are simplified.

Contact elements (stopper) are utilized to limit the travel range. Contact elements are necessary to model pull-in and hysteresis release effects. The contact behavior is defined by a contact stiffness value to adjust the penetration depth and a damping factor to tune bouncing effects.

Damping can be assigned by the alpha-beta-damping approach (Rayleigh damping), by a constant damping ratio and by modal damping ratios assigned to individual modes (for modal superposition).

The MODELBUILDER supports static, modal, harmonic and transient simulations, whereby electro-mechanical interactions are considered for all types of analysis. Interfaces are available for model export to ANSYS®, COMSOL® and SIMULINK®.

The MEMS design process

The MODELBUILDER provides a comprehensive design support with an automated model generation procedure and a consistent data exchange process during all phases of the MEMS design process.

MEMS design starts with a Conceptual design phase where new ideas are explored, is followed by a highly accurate finite-element based model evaluation phase (Interface to ANSYS®, Interface to COMSOL®) and finally, the system design phase where the functional behavior of MEMS is tested in a user-defined system environment (Interface to SIMULINK®).

Fast and accurate system models are the goal of MEMS design. System models combine micro-mechanical sensor and actuator models with controller or circuit representations in order to simulate the interactions of the whole system at different load, operating and environmental conditions. System modeling is a big challenge and requires fast and accurate reduced-order-models (ROM) of all involved domains.

The MODELBUILDER and the interfaces to ANSYS®, COMSOL® and SIMULINK® support MEMS designers to shorten the product development cycle and to deliver sensor and actuator components with optimized performance by advanced simulation features. In academia, the MODELBUILDER supports lecturers to train students with realistic examples and make classroom theory come alive. Students and engineers can prototype new designs in shortest time and visualize or animate the simulation results in 3D. Even the behavior of complicated MEMS can be explained with ease. The software components are sophisticated, practical and easy-to-use.

Conceptual design

Conceptual design is the first phase of the MEMS design process where different ideas for new MEMS products must be analyzed, optimized and compared to each other. Most important for conceptual design are fast procedures to generate parametric models at different levels of abstraction (model-order-reduction).

Fast models with a low number of governing equations are required to extract characteristic properties and to simulate the dynamic response of devices in a reasonable time.

Typical MEMS devices are complicated structures and it is difficult to find proper dimensions and physical parameters which fulfill the product requirements.

Fast parametric models are essential for dimensional design and structural optimization because a very large number of design variants must be analyzed. During optimization, design parameters are varied in a wide range whereby Taylor series approximations or function fit procedures are not applicable.

After proper dimensions have been found, designers must evaluate the influence of manufacturing tolerances and imperfections on the systems performance. Often several hundreds of Monte-Carlo-simulations are be carried out with different process data combinations and slightly modified parameter sets. Finite element simulations would be too cumbersome for conceptual design.

Efficient modeling and simulation during conceptual design is realized through a rigid-body approximation, using Timoshenko beam theory instead of time-consuming 3D-field solver, applying different model reduction levels (Guyan condensation) and the use of a modal superposition solver for the mechanical domain. Those modeling methods reduce the complexity and the number of the governing equations what is referred reduced-order-modeling (ROM) in literature. Users can choose different reduction levels to adjust accuracy versus speed.

The accuracy of reduced-order-models depends on the layout and the chosen approximations. In most cases, results are appropriate for the conceptual design phase. In the next design phase follow detailed investigations of the favorite model with preferred dimensions using highly accurate but also time-consuming simulation techniques.

Finite-element model export (Interface to ANSYS®)

The MODELBUILDER provides an Interface to ANSYS®. All model components including corner fillets, mask undercut and etch sidewall slope data are transferred automatically to the finite-element-program ANSYS® in order to create a very precise 3D-representation of the current design. The generated model exchange file contains not only solid model items but also material properties, boundary conditions and mesh settings needed for full-order finite-element-models (FEM). Specific commands to run a modal analysis in ANSYS® are added at the end of the file for demonstration. The model export is based on APDL-commands (ANSYS Parametric Design Language).

Finite-element simulations are highly accurate. Nevertheless, transient coupled-domain simulations of MEMS based on FEM are very time-consuming. They are rarely used in practice for complex microsystems. Rather, FE-simulations are used to verify and improve characteristic properties of the preferred MEMS design. Typical data of interest are eigenfrequencies, stiffness data, capacitances and capacitance derivatives.

It is easier and more efficient to adapt model parameters obtained from conceptual design in ROM-simulations instead of running time-consuming full-order FE-simulations. Improved reduced-order-models are utilized for the final phase of design where the behavior of MEMS components will be assessed in a User-defined system environment. The Interface to Simulink® is important to accomplish this task.

Finite-element model export (Interface to COMSOL®)

The MODELBUILDER provides an [Interface to COMSOL®](#) which is similar to the Interface to ANSYS®. Here, the data transfer is based on the COMSOL Application Builder. The COMSOL® interface supports the export of all solid model items including material data, corner fillets, mask undercut and the etch sidewall slope. Mesh settings and special assignments for physical finite element simulations are very model specific. In some cases, it requires manual corrections of assigned data in the COMSOL user interface to create optimal finite element models.

Export of mask layout data (GDS Interface)

Geometric data of the functional layers including information of bottom and top parallel plate capacitors are exported in GDSII text format. The exported mask layout is based on nominal data without mask undercut.

Model export for system simulations (Interface to SIMULINK®)

The MODELBUILDER provides an [Interface to SIMULINK®](#). All model components are automatically transferred to a signal-flow graph for system simulations in a customized environment. Input signals in the mechanical domain are forces and moments, translational and rotational accelerations as well as angular rates needed for Coriolis effects. Input terms in the electrical domain are voltage functions. Output signals are displacements and tilts, the electrical currents and capacitances between conductors.

The generated Simulink models are valuable for a comprehensive investigation of the functional behavior of MEMS. All internal signals as electrostatic forces, contact forces, Coriolis terms or capacitances and capacitance derivatives are accessible after system simulation runs. Most important is that the Simulink models can be directly inserted in a customized model environment to simulation the interaction with controller units or circuit representations.

Simulink model parameters obtained from conceptual design are usually replaced by finite-element simulation results or by data obtained from measurements on test structures to improve the accuracy (Fig. 1). Parameter adaptation becomes possible, since MEMS models can be exported at different levels of abstraction. Easiest way is choosing a modal superposition representation prior model export. In that case, eigenfrequencies can be modified in SIMULINK blocks to adapt stiffness terms in the mechanical domain and capacitances or capacitance derivatives can be replaced by fit-functions or lookup tables in the electrostatic domain. In the current release, model adaptation must be done manually.

Graphical model generation (SKETCHER Interface)

All model components can either be defined in a command-based model file or in graphical way using the SKETCHER interface. The command-based modeling approach is very fast and efficient but requires some experiences with the i-ROM MODELBUILDER design language. Part D of the manual explains the command-based approach on several examples and Part E the command syntax. The command file is directly evaluated by the MODELBUILDER main program.

Alternatively, all model components can be defined and modified in a graphical way using the SKETCHER interface. The graphical approach is very intuitive and easy to learn. The user interface of the SKETCHER is similar to a common CAD environment. Alphanumerical settings, design variables and mathematical terms can be assigned to all shape elements in property taps. New model components can be selected from library elements, placed in the model window, changed in size or aligned to other items by mouse manipulations.

The SKETCHER interface is fully integrated into the MODELBUILDER main program. It supports a bidirectional data exchange between the command-based approach and the internal graphical data structure.

In the current release, the data exchange of a few shape elements such as elliptical areas, Bezier-shaped springs and user defined areas with arbitrary curved or Bezier-shaped outlines is not implemented yet. Also, Bezier-shaped springs and the parametric mirror, copy and array operations follow soon.

Model simplifications and approximations

The MODELBUILDER uses model simplifications and approximations to create fast parametric models during conceptual design.

Seismic masses are considered rigid. In most cases, seismic masses are designed to be very stiff in order to avoid bending of masses and unwanted vibrations in the operating range. For this reason, MEMS usually consist of thick functional layers manufactured by High Aspect Ratio Micromachining (HARM) technologies.

The advantage of using rigid-body approximations is the tremendous reduction of degrees of freedom, since masses can be described exactly by only six degrees of freedom (DOF).

Suspension springs are represented by Timoshenko beam elements. Beam models are applicable for springs which are long compared to the thickness or width what happens in most cases. Beam models are accurate and represent the mechanical behavior with a very low number of elements for each spring.

During conceptual design, capacitances are calculated from the quasi-homogeneous electrostatic field between conductors. Fringing fields are ignored, and tilt should be < 20°. However, capacitances depend on all six motion DOF of the attached rigid bodies. Both, the change of the overlapping area and the change of the electrode gap are evaluated by the implemented algorithms. Obviously, ignoring fringing fields leads to deviations. Capacitances are smaller as real capacitances, but capacitance derivatives are still accurate.

Deviations of capacitance values are not critical. Most parameters of capacitive transducers depend mainly on the capacitance derivatives. The output signal of sensors is proportional to the capacitance change for given loads and forces of actuators depend likewise on capacitance derivatives.

In practice, the comb and plate capacitances are superimposed by parasitic capacitances caused by connecting wires, the bond pads, the package and the printed circuit board. Those values are much larger as the fringing field of comb or plate capacitances and, in any case, must be added to the system model by additional lumped elements before the simulations are performed.

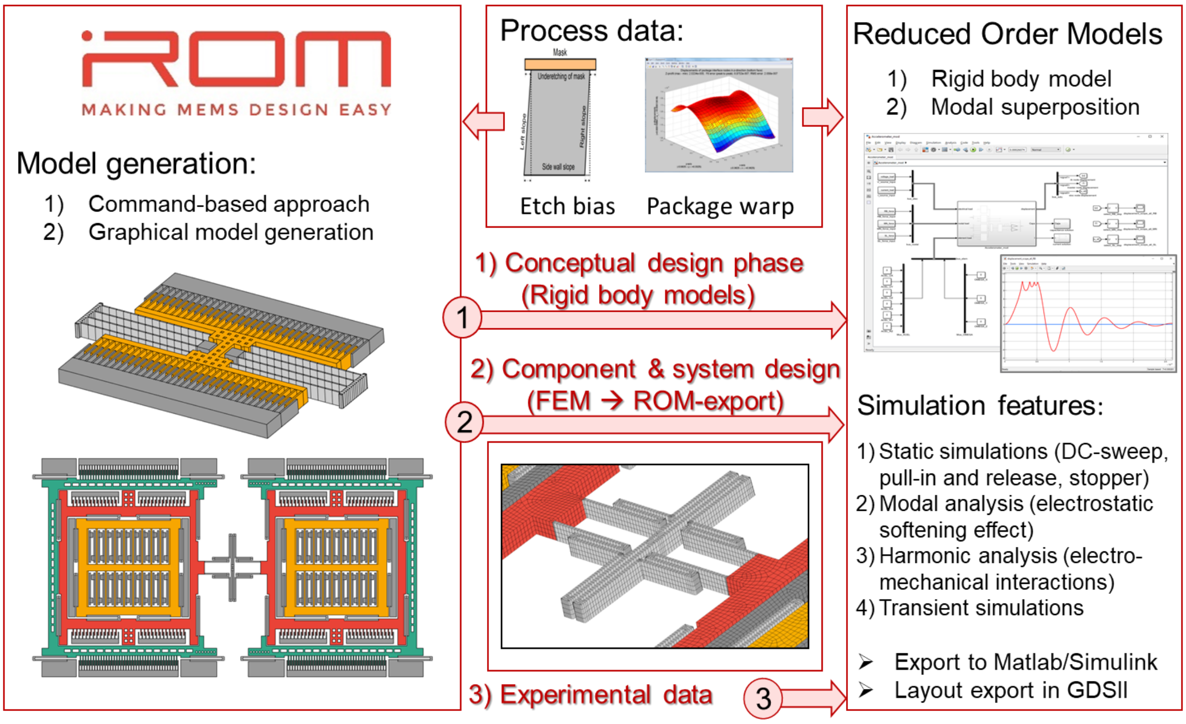
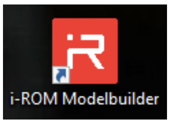


Fig. 1: MEMS-design flow supported by the i-ROM MODELBUILDER

Part B - Quick start

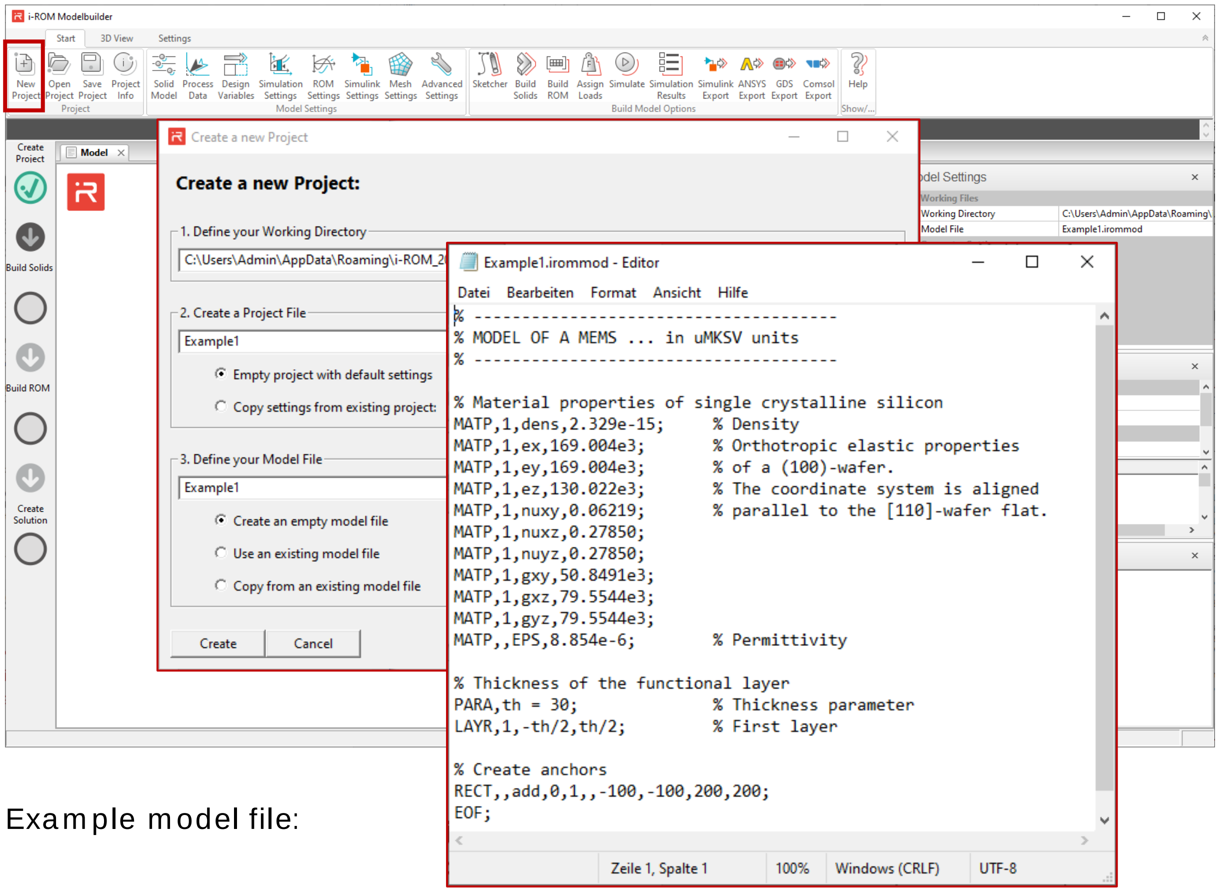
1 Model generation procedure

➢ Start the i-ROM MODELBUILDER from the desktop or the source code folder



➢ Prepare a new project “Example1” in the “Test_case1” folder

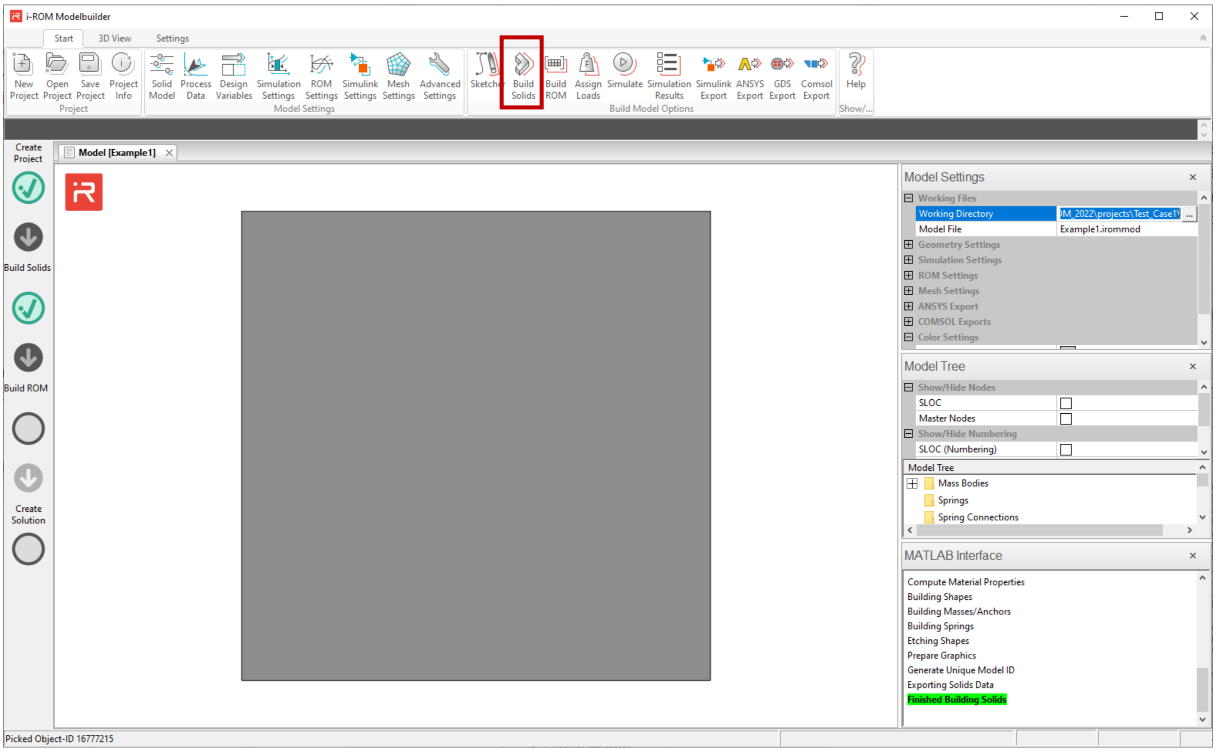
- Click on the “New Project” icon.
- Enter or select all items in the “Create a new Project” window.
- A new model input file “Example1.irommod” with material properties and a dummy anchor appears in the working directory.



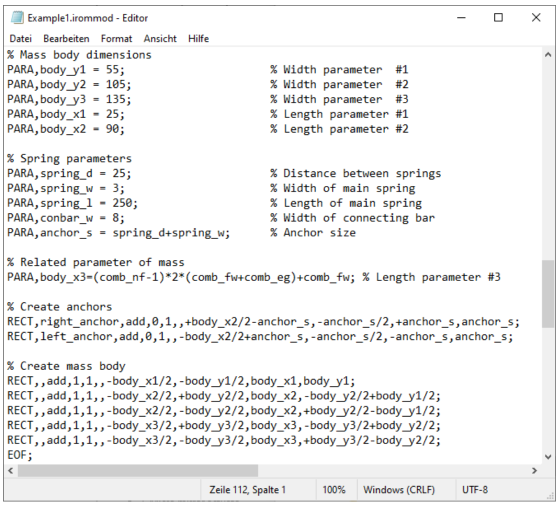
Example model file:

All example files discussed in this “User Manual” can be found in the “project” folder of the MODELBUILDER. In order to run an existing example, open a project file, click “Build Solids” to create a 3D-model, then click “Build ROM” to create a ROM-model and finally click “Simulate” to run a modal analysis.

- Click on “Build Solids” to show the initial settings
  - The dummy anchor with 200 µm edge length is shown in dark gray.

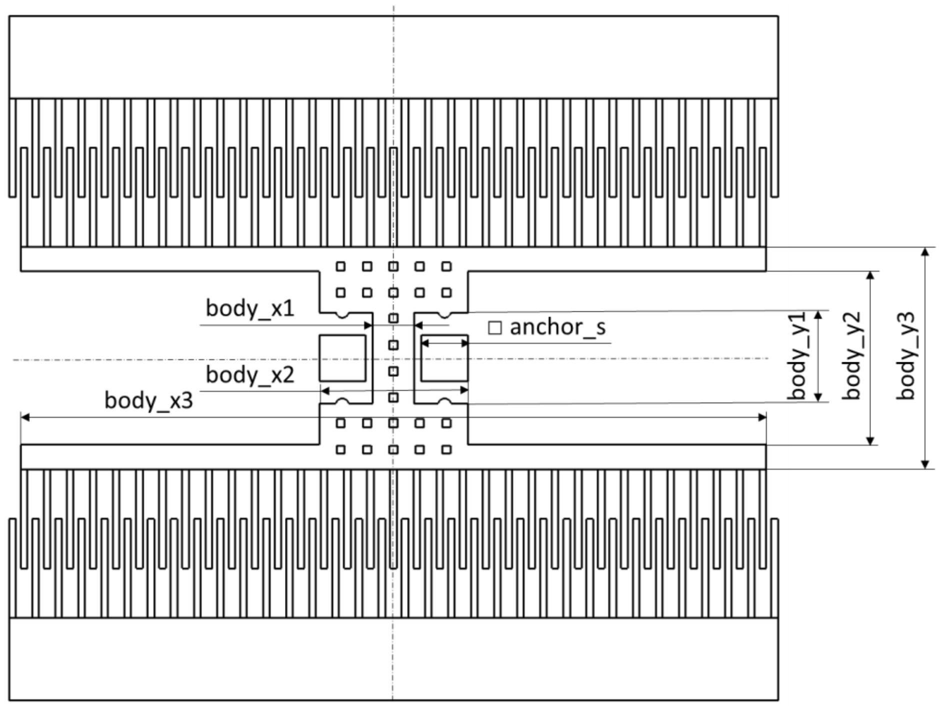


- Plan you modeling approach and add further commands to the model file
  - To avoid time-consuming editing, the content of “Example1_save.irommod” can be copied into “Example1.irommod”.
  - The model file starts with design parameters defined by PARA commands.
  - Then follow commands to define anchors and seismic masses. In the given example, the mass body is assembled from five rectangular primitives.
  - Parametric modeling is helpful to optimize MEMS designs. Alternatively, all dimensions of the RECT command can directly be defined by numbers.
  - A break point in the model file can be set by the EOF command.

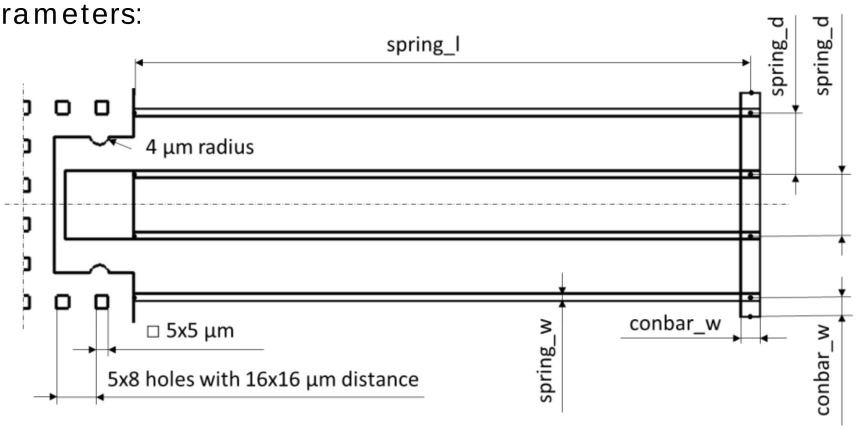



```
% Mass body dimensions
PARA,body_y1 = 55;                    % Width parameter  #1
PARA,body_y2 = 105;                   % Width parameter  #2
PARA,body_y3 = 135;                   % Width parameter  #3
PARA,body_x1 = 25;                    % Length parameter #1
PARA,body_x2 = 90;                    % Length parameter #2

% Spring parameters
PARA,spring_d = 25;                   % Distance between springs
PARA,spring_w = 3;                    % Width of main spring
PARA,spring_l = 250;                  % Length of main spring
PARA,conbar_w = 8;                    % Width of connecting bar
PARA,anchor_s = spring_d+spring_w;    % Anchor size

% Related parameter of mass
PARA,body_x3=(comb_nf-1)*2*(comb_fw+comb_eg)+comb_fw; % Length parameter #3

% Create anchors
RECT,right_anchor,add,0,1,,+body_x2/2-anchor_s,-anchor_s/2,+anchor_s,anchor_s;
RECT,left_anchor,add,0,1,,-body_x2/2+anchor_s,-anchor_s/2,-anchor_s,anchor_s;

% Create mass body
RECT,,add,1,1,,-body_x1/2,-body_y1/2,body_x1,body_y1;
RECT,,add,1,1,,-body_x2/2,+body_y2/2,body_x2,-body_y2/2+body_y1/2;
RECT,,add,1,1,,-body_x2/2,-body_y2/2,body_x2,+body_y2/2-body_y1/2;
RECT,,add,1,1,,-body_x3/2,+body_y3/2,body_x3,-body_y3/2+body_y2/2;
RECT,,add,1,1,,-body_x3/2,-body_y3/2,body_x3,+body_y3/2-body_y2/2;
EOF;
```

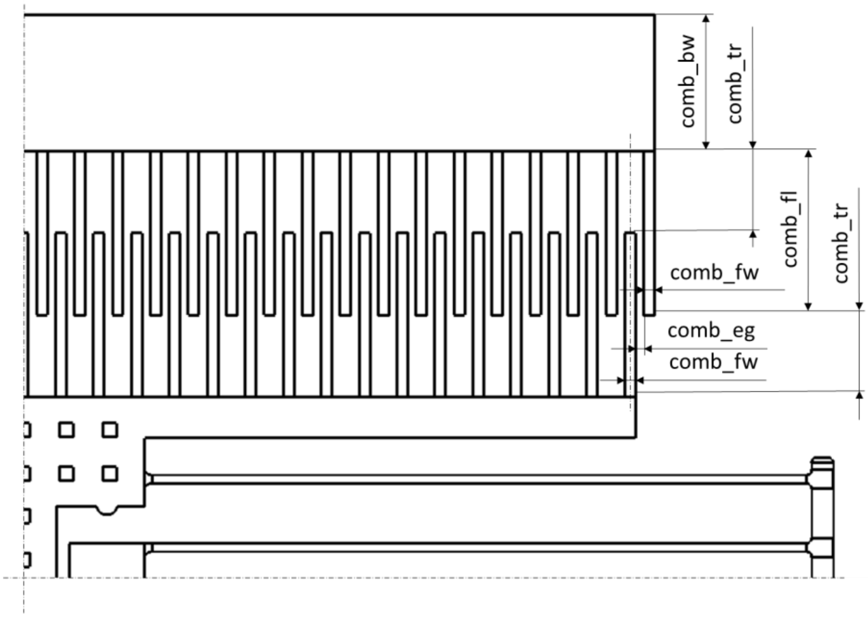
Mass and anchor parameters:



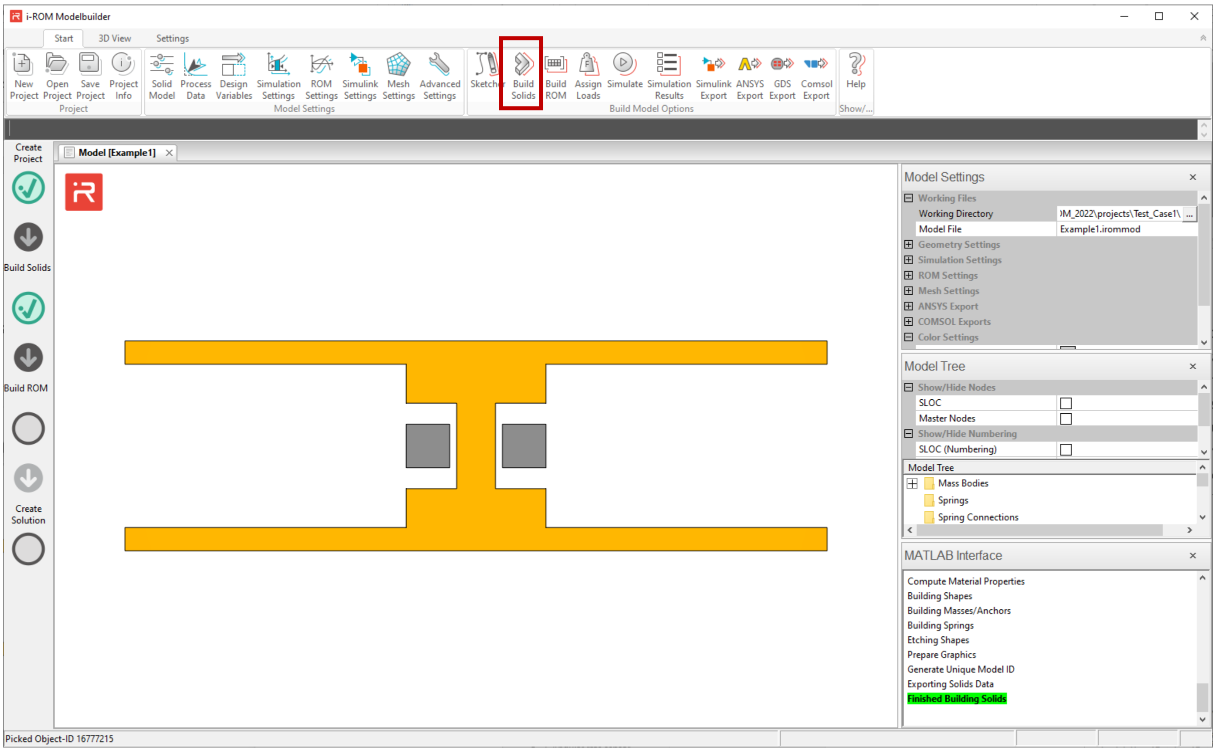
Spring parameters:



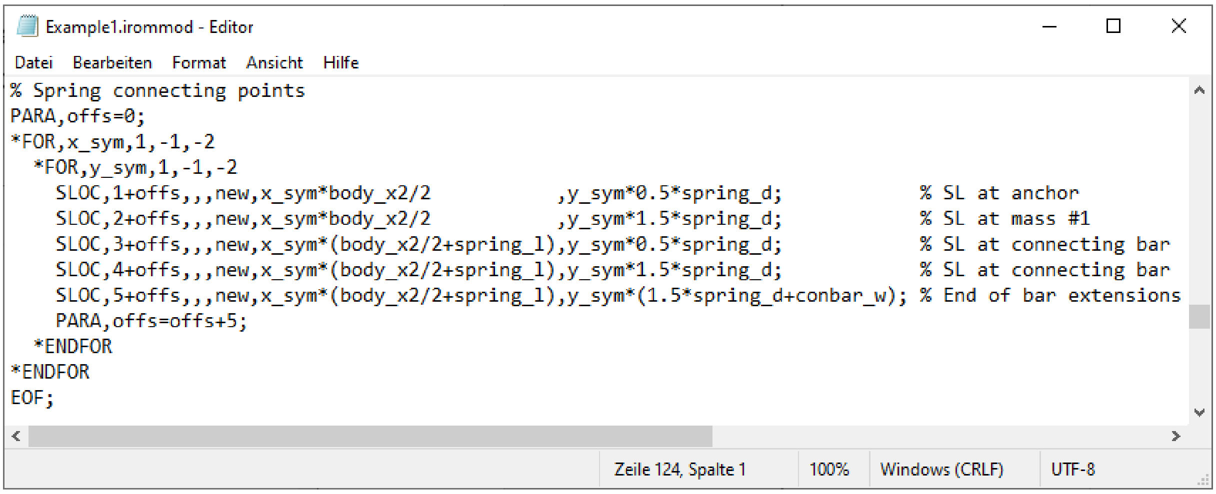
Comb cell parameters:



- Click on “Build Solids” to update the model
  - Two anchors shown in dark gray and the seismic mass shown in orange appear on the screen.



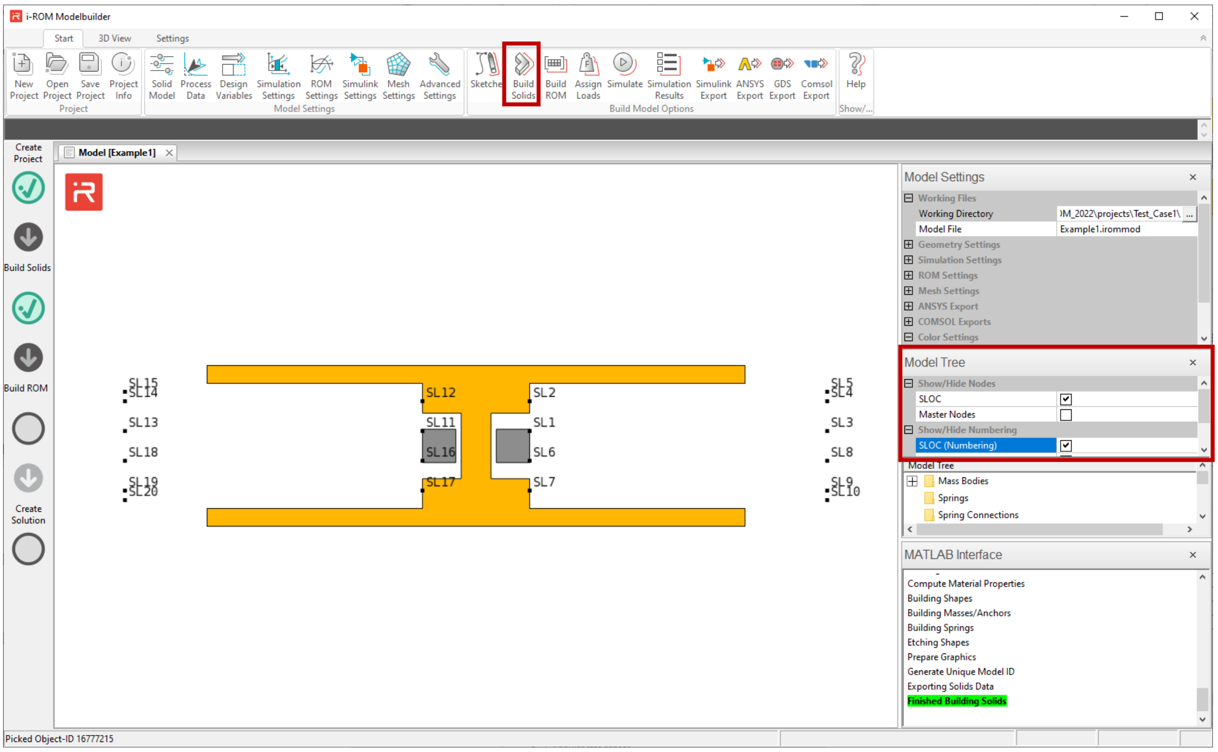
- Define connecting points for suspension springs (SL - Spring Location point)
  - Five connecting points are defined by the SLOC command.
  - The connecting points are mirrored in x- and y-direction by *FOR-loops.
  - In total 20 spring location points appear on the screen which are utilized for a subsequent spring design.



```
% Spring connecting points
PARA,offs=0;
*FOR,x_sym,1,-1,-2
  *FOR,y_sym,1,-1,-2
    SLOC,1+offs,,,new,x_sym*body_x2/2          ,y_sym*0.5*spring_d;           % SL at anchor
    SLOC,2+offs,,,new,x_sym*body_x2/2          ,y_sym*1.5*spring_d;           % SL at mass #1
    SLOC,3+offs,,,new,x_sym*(body_x2/2+spring_l),y_sym*0.5*spring_d;           % SL at connecting bar
    SLOC,4+offs,,,new,x_sym*(body_x2/2+spring_l),y_sym*1.5*spring_d;           % SL at connecting bar
    SLOC,5+offs,,,new,x_sym*(body_x2/2+spring_l),y_sym*(1.5*spring_d+conbar_w); % End of bar extensions
    PARA,offs=offs+5;
  *ENDFOR
*ENDFOR
EOF;
```

➢ Click on “Build Solids” to update the model

- Markers and numbers of connecting points can be switched on in the “Model Tree” window shown in the right of the figure below.



➢ Define suspension springs between connecting points

- Nine suspension springs are defined on the right and on the left side of the sensor by the SPRI command.
- The corner fillets at SL1, SL6, SL11, and SL16 are automatically switched off in the current release.

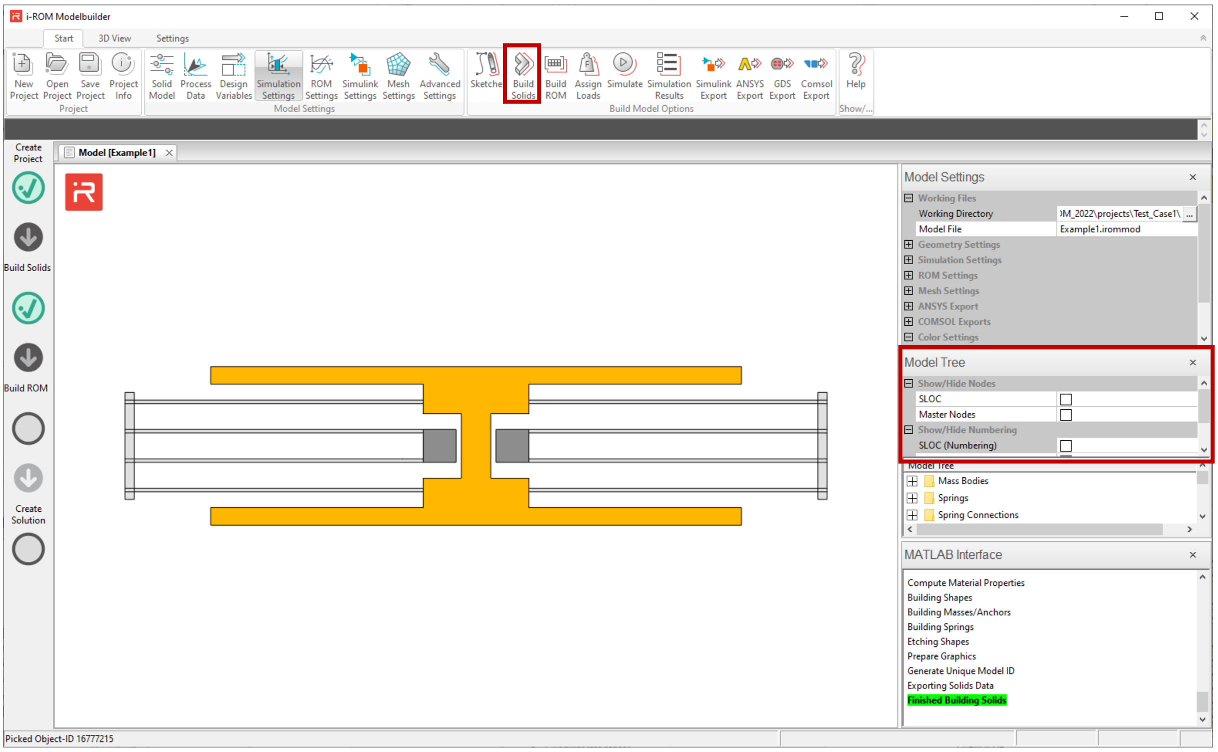
Example1.irommod - Editor

Datei Bearbeiten Format Ansicht Hilfe

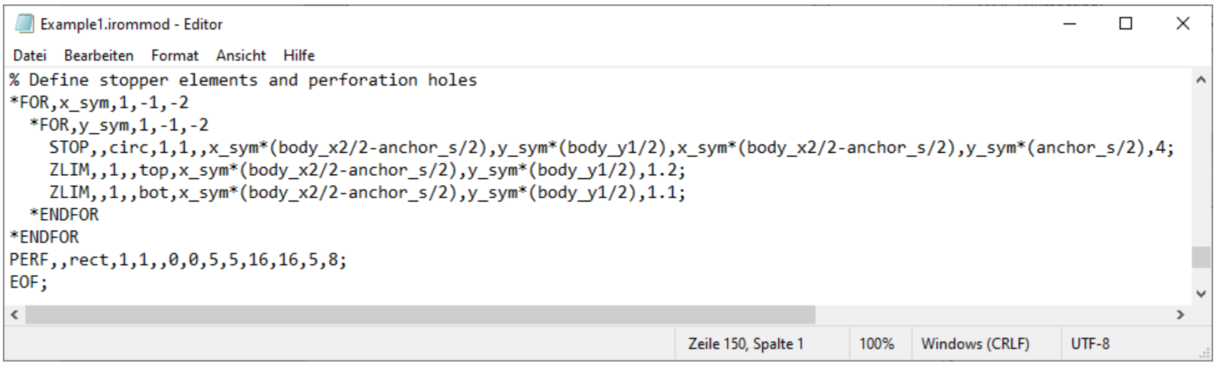
```
*FOR,x_sym,0,1
  PARA,offs=10*x_sym;
  SPRI, 1+offs,3+offs,,,spring_w;      % No corner fillet SL1 left
  SPRI, 6+offs,8+offs,,,spring_w;      % No corner fillet SL6 right
  SPRI, 2+offs,4+offs,,,spring_w;      % Spring SL2 to SL4
  SPRI, 7+offs,9+offs,,,spring_w;      % Spring SL7 to SL9
  SPRI,10+offs,9+offs,,,conbar_w;      % Connecting bar
  SPRI, 4+offs,5+offs,,,conbar_w;      % Connecting bar
  SPRI, 9+offs,8+offs,,,conbar_w;      % Connecting bar
  SPRI, 8+offs,3+offs,,,conbar_w;      % Connecting bar
  SPRI, 3+offs,4+offs,,,conbar_w;      % Connecting bar
*ENDFOR
EOF;
```

Zeile 139, Spalte 1 | 100% | Windows (CRLF) | UTF-8

- Click on “Build Solids” to update the model
  - The suspension springs on both sides are shown in light gray.



- Define stopper elements and a perforation pattern of the seismic mass
  - Four circular stopper elements are defined by the STOP command.
  - Stopper are described by the contact and target node location.
  - In addition, there are four vertical stopper elements at the top and at the bottom face defined by the ZLIM command.
  - A perforation pattern with rectangular holes is defined by the PERF command.



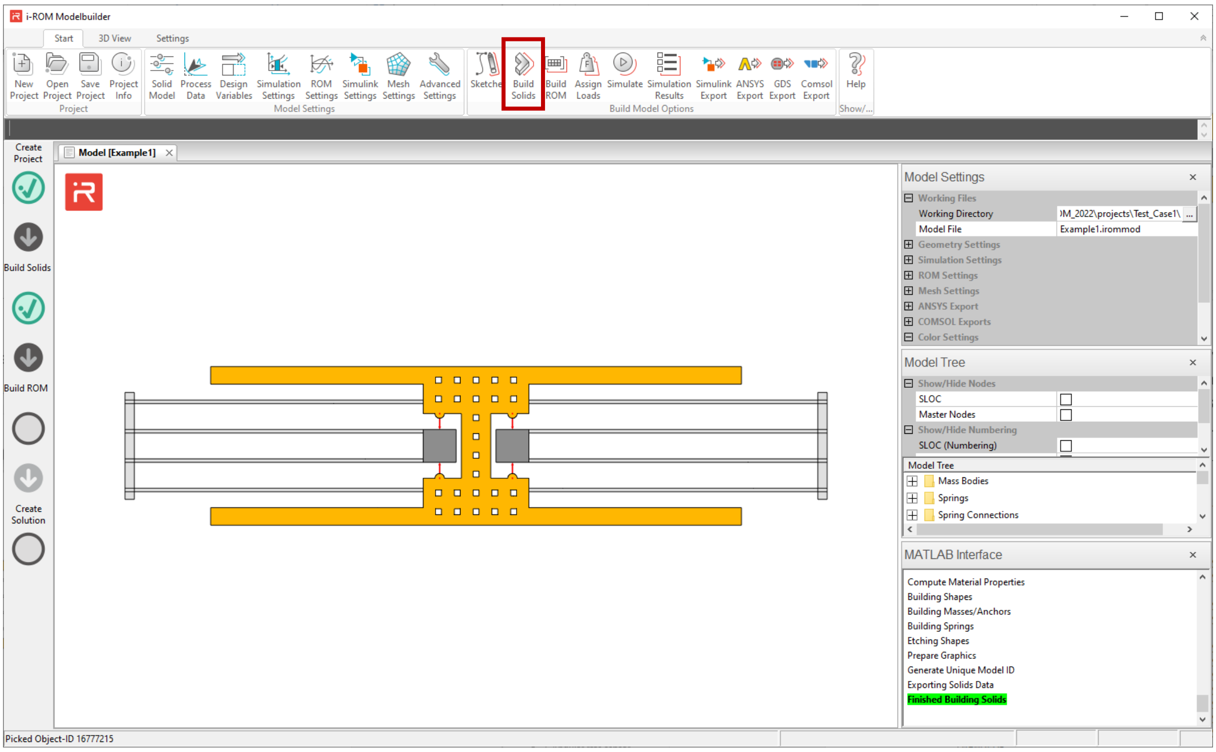
Example1.irommod - Editor

Datei Bearbeiten Format Ansicht Hilfe

```
% Define stopper elements and perforation holes
*FOR,x_sym,1,-1,-2
  *FOR,y_sym,1,-1,-2
    STOP,,circ,1,1,,x_sym*(body_x2/2-anchor_s/2),y_sym*(body_y1/2),x_sym*(body_x2/2-anchor_s/2),y_sym*(anchor_s/2),4;
    ZLIM,,1,,top,x_sym*(body_x2/2-anchor_s/2),y_sym*(body_y1/2),1.2;
    ZLIM,,1,,bot,x_sym*(body_x2/2-anchor_s/2),y_sym*(body_y1/2),1.1;
  *ENDFOR
*ENDFOR
PERF,,rect,1,1,,0,0,5,5,16,16,5,8;
EOF;
```

Zeile 150, Spalte 1 | 100% | Windows (CRLF) | UTF-8

- Click on “Build Solids” to update the model
  - Stopper, z-limiter and a perforation pattern appear at the seismic mass.



- Define comb cells for capacitive sensing and electrostatic actuation
  - Each comb cell is defined by a single COMB command.
  - Two master nodes (MN) are defined at the upper and lower part of the seismic mass by the MAST command.
  - Master nodes can be visualized by activating markers and numbers (labels) in the “Model Tree” window.

Example1.irommod - Editor

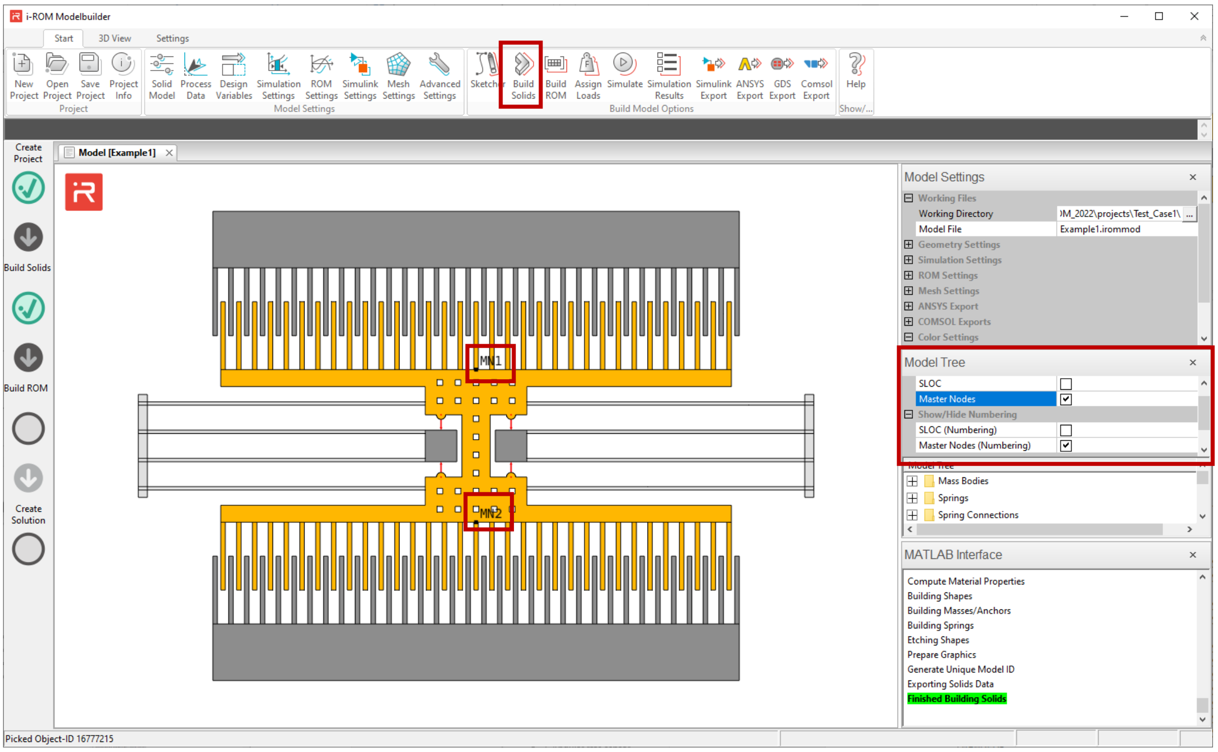
Datei Bearbeiten Format Ansicht Hilfe

```
% Define capacitors, comb cell with area variation
COMB,sense+,area,1,1,, 90,comb_nf,1,comb_fw,comb_fl,comb_bw,comb_tr,comb_eg,,,0,+body_y3/2;
COMB,sense-,area,1,1,,270,comb_nf,1,comb_fw,comb_fl,comb_bw,comb_tr,comb_eg,,,0,-body_y3/2;

% Master nodes
MAST,,1,1,1,0,+body_y3/2,-th/2;
MAST,,2,1,1,0,-body_y3/2,-th/2;

EOF;
```

Zeile 149, Spalte 1 | 100% | Windows (CRLF) | UTF-8

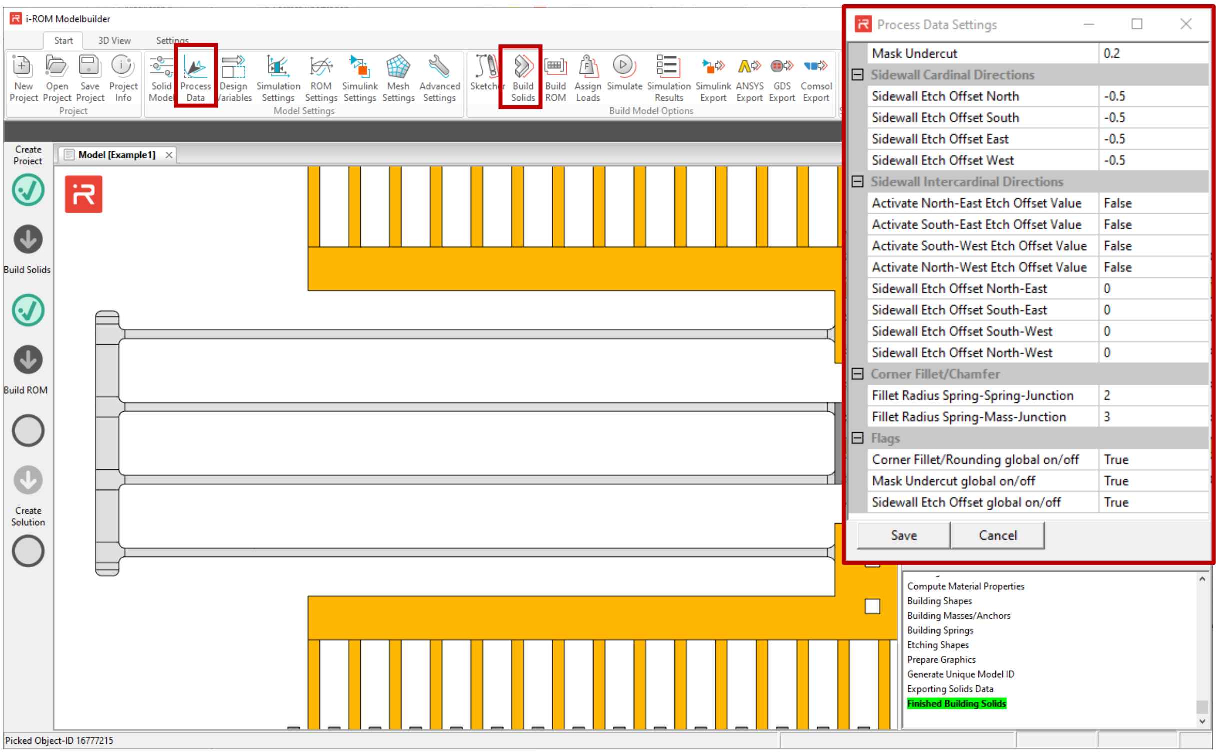
➢ Click on “Build Solids” to update the model

- Comb cells are attached at the upper and lower edge of the seismic mass.



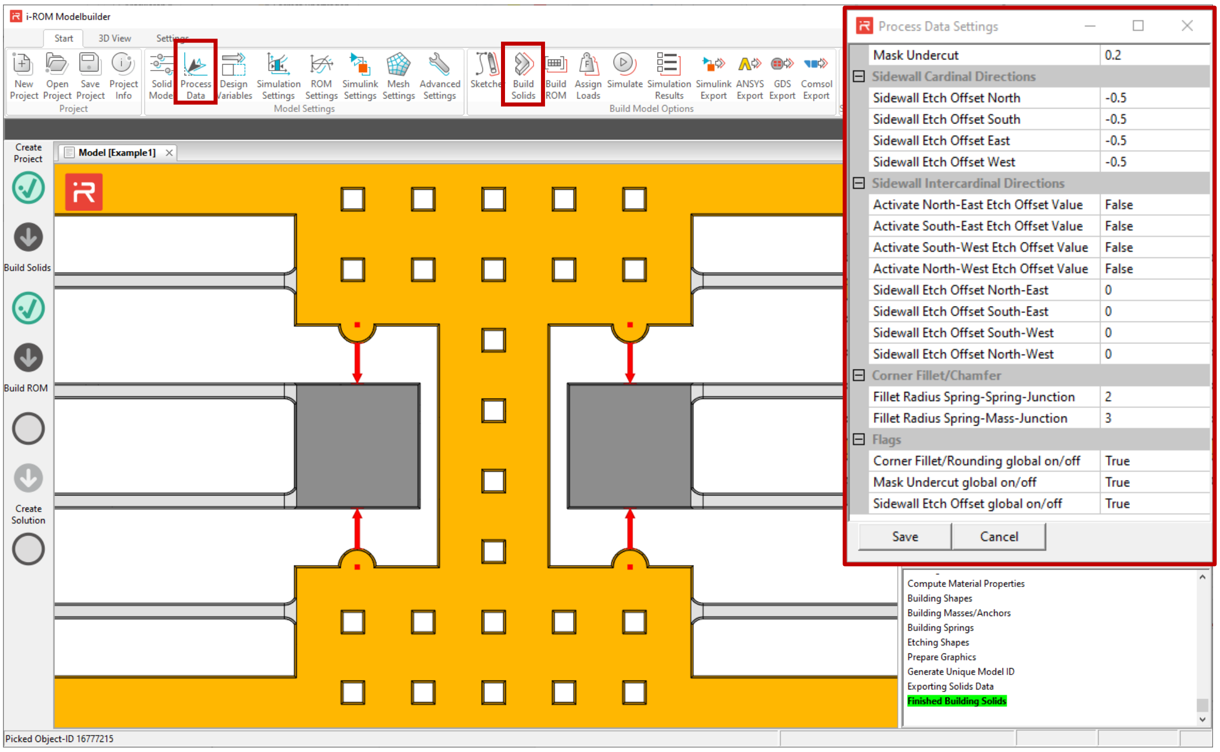
➢ Assign corner fillets at spring junctions to reduce stress concentrations

- Two different corner radiuses are assigned to spring-spring- and spring-mass-junctions in the [Process Data Settings]() window.
- The tilt angle at junctions to masses is 30°. It can be changed in the [Solid Model Settings]() window. Click on “Build Solids” to update the model.



➢ Assign mask undercut and sidewall etching data

- A “Mask Undercut” value of 0.2 µm is assigned to all outer edges.
- Enlarged “Sidewall Etch Offset” data are assigned to faces pointing north, south, east and west. All bottom edges move 0.5 µm outwards.
- Click on “Build Solids” to update the model.



➢ Click on the “3D View” panel to inspect the model from different orientations

Use orthogonal view buttons or manipulate the orientation of the model by mouse buttons:

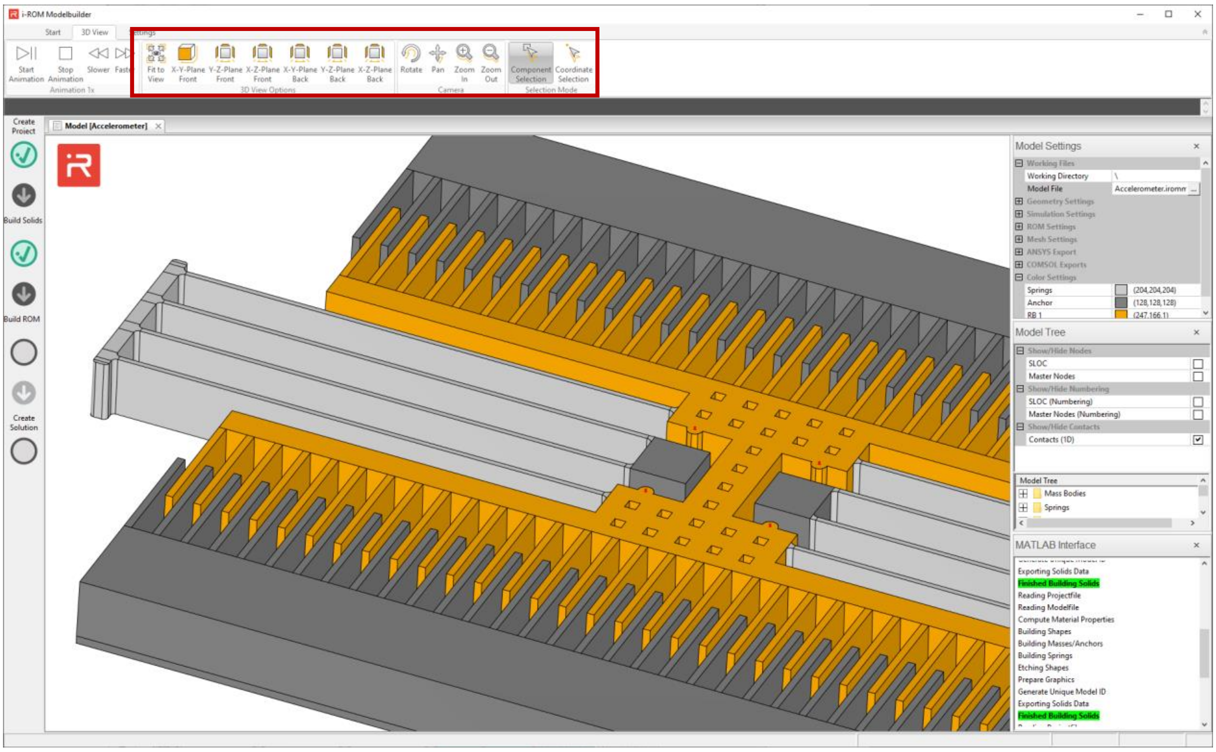
- Press “Ctrl + left mouse button” to move the structure in x- or y-directions,
- Press “Ctrl + scroll the mouse wheel” to zoom in or out, and
- Press “Ctrl + right mouse button” to tilt the structure around x- and y-axes.

The User can either tilt the model around its Center Point (CP-mode) or around a User-defined Pivot Point (PP-mode) located on the model.

- To activate the “CP-mode”, move the mouse pointer into the open space. Press “Ctrl + right mouse button” and tilt the model around x- or y-axes.
- For the “PP-mode”, move the mouse pointer to the pivot point of interest. Press “Ctrl + right mouse button” and tilt the model around x- or y-axes.

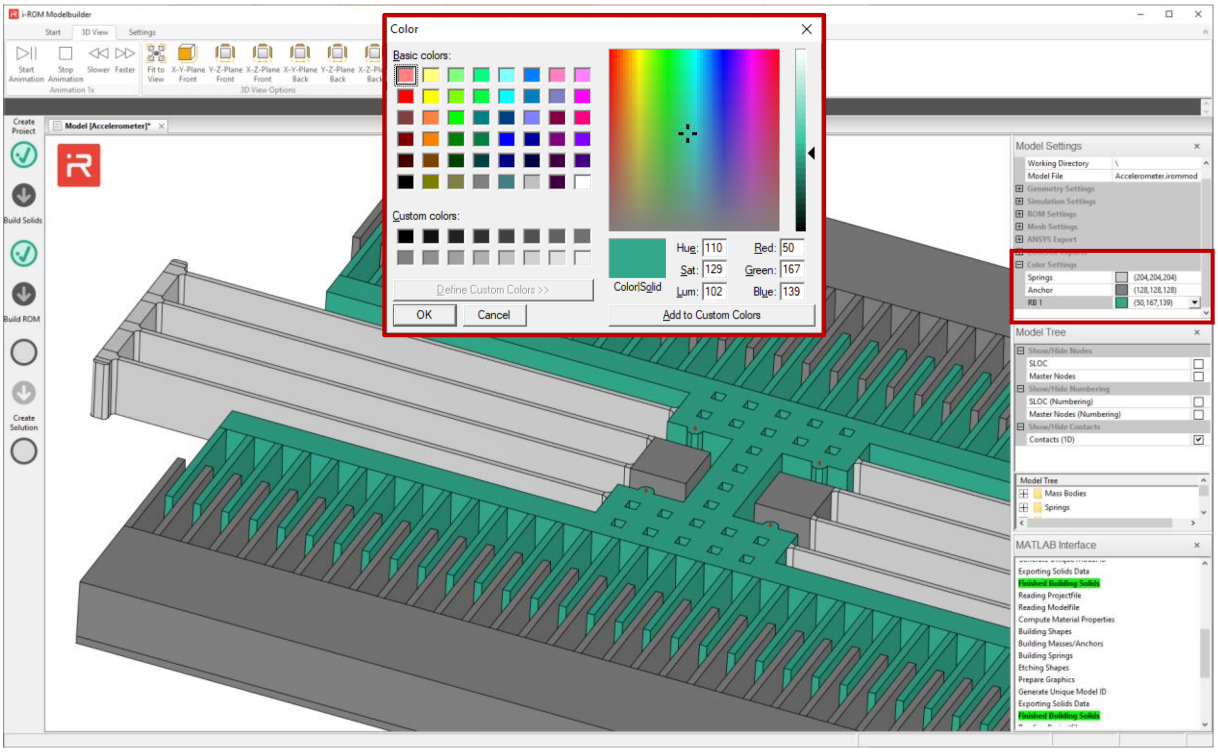
A rectangular section of the model can be selected by the “Box zoom” mode. Press the right mouse button and move the mouse from the upper left to the lower right position in the model window.

➢ The “Fit to View” button shows the model at full size on the display screen.



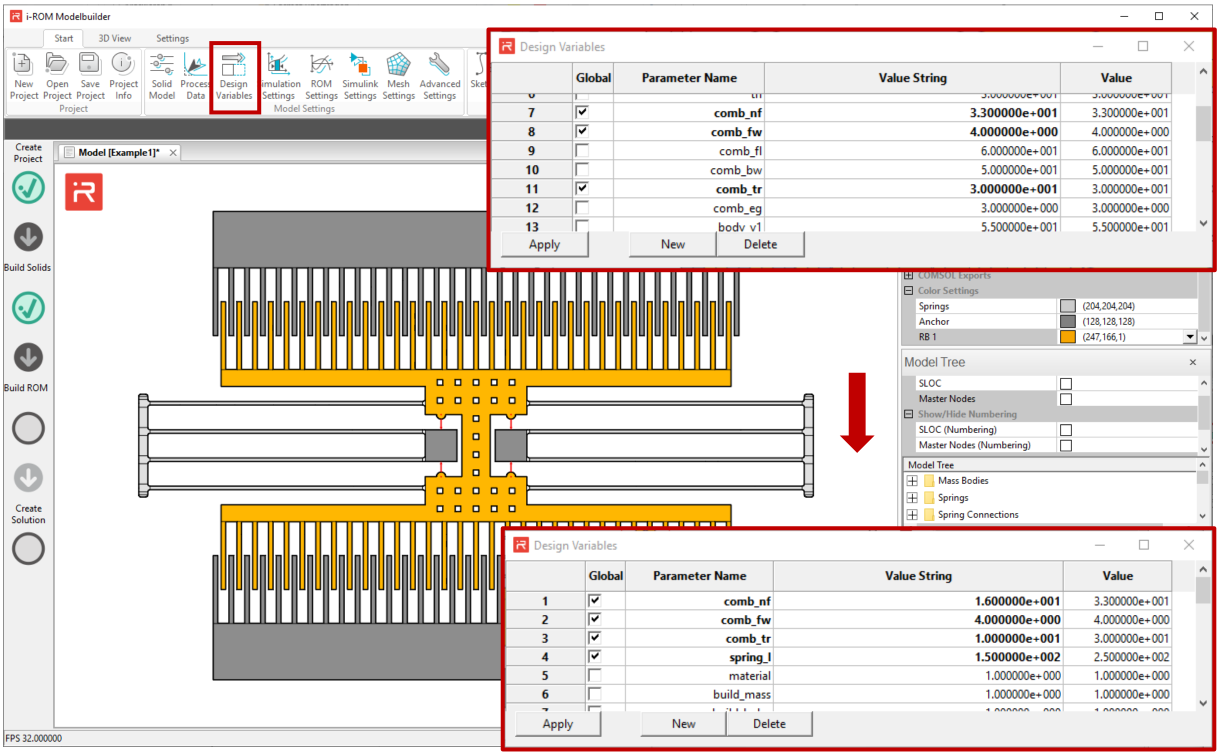
➢ Change “Color Settings” of mass bodies in the “Model Settings” window

- Default and customized colors can be assigned to all mass bodies.
- Exemplarily, the color of mass #1 is set to green as shown below.
- Click on “Save Project” to store the color data settings.



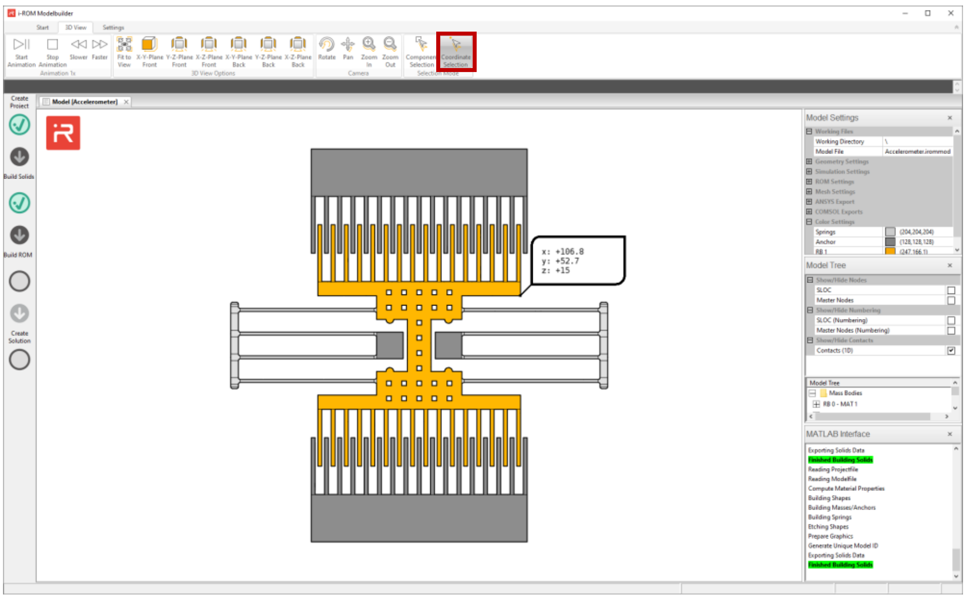
➢ Assign “Design Variables” to the model for case studies and optimization

- The number of comb fingers, the comb travel range, the spring length and comb finger width have been set “global” in the Design Variables window. Global values can be changed in the GUI and overwrite settings in the model file. Modified values apply for the next “Build Solids” command. All other variables are taken from the model file.
- “New” variables must be “global” or they are defined in the model file.

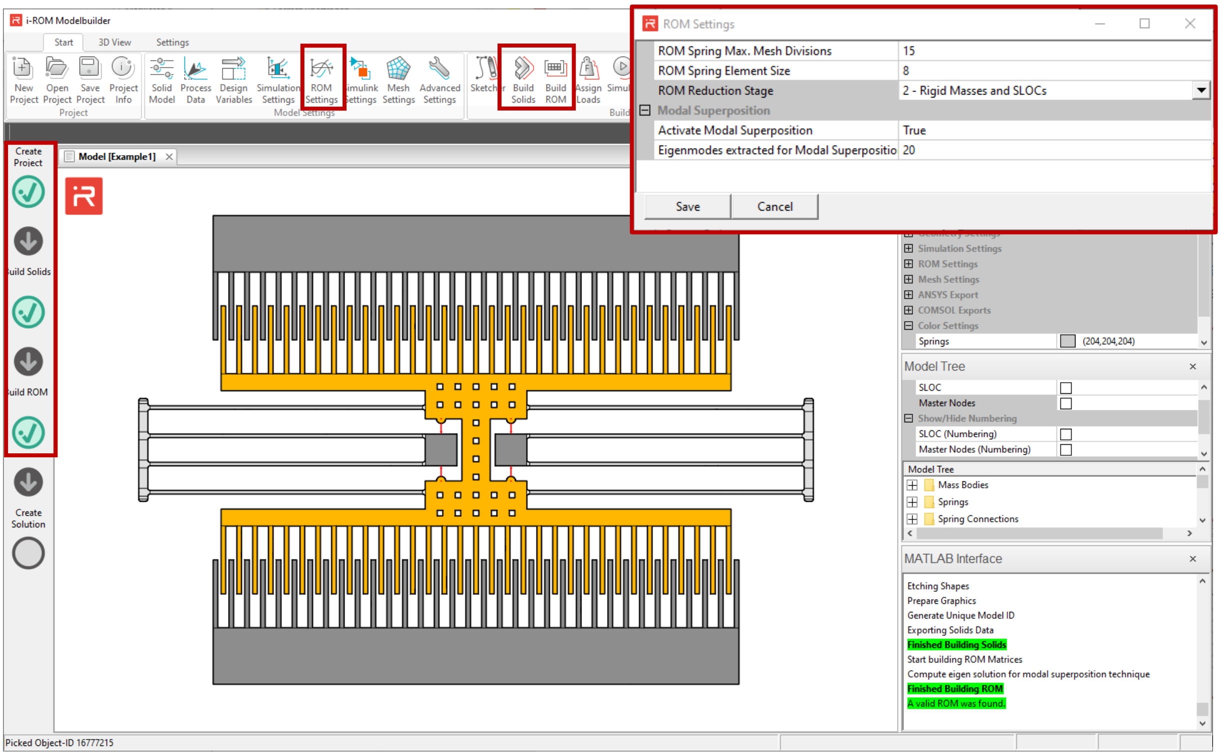


➢ Click on “Build Solids” to update the model

- Click on the “Coordinate Selection” icon in order to check several dimensions of the model. Click into the open space to remove the marker.



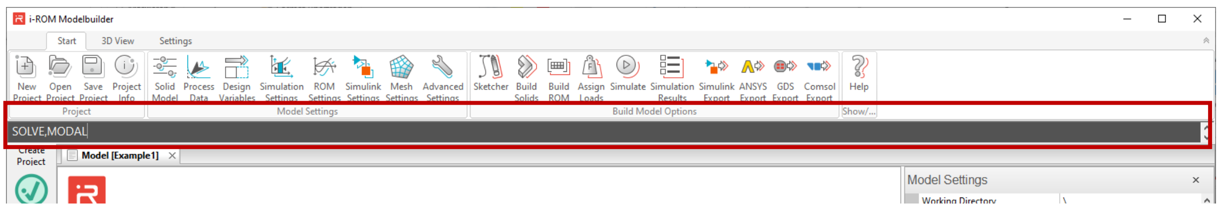
- Create a numerical reduced-order model (ROM) for subsequent simulations
  - Reassign the original settings in the Design Variables window.
  - Click on “Build Solids” to update the original solid model.
  - Open the ROM Settings window.
  - Change “Spring Max. Mesh Divisions” to 15 and “Spring Element Size” to 8 µm (the current model uses micrometer units → see µMKSV-system).
  - Click on “Build ROM” to create a numerical simulation model.
  - The gray circles between “Build ROM” and “Create Solution” turn green after the routines are finished.



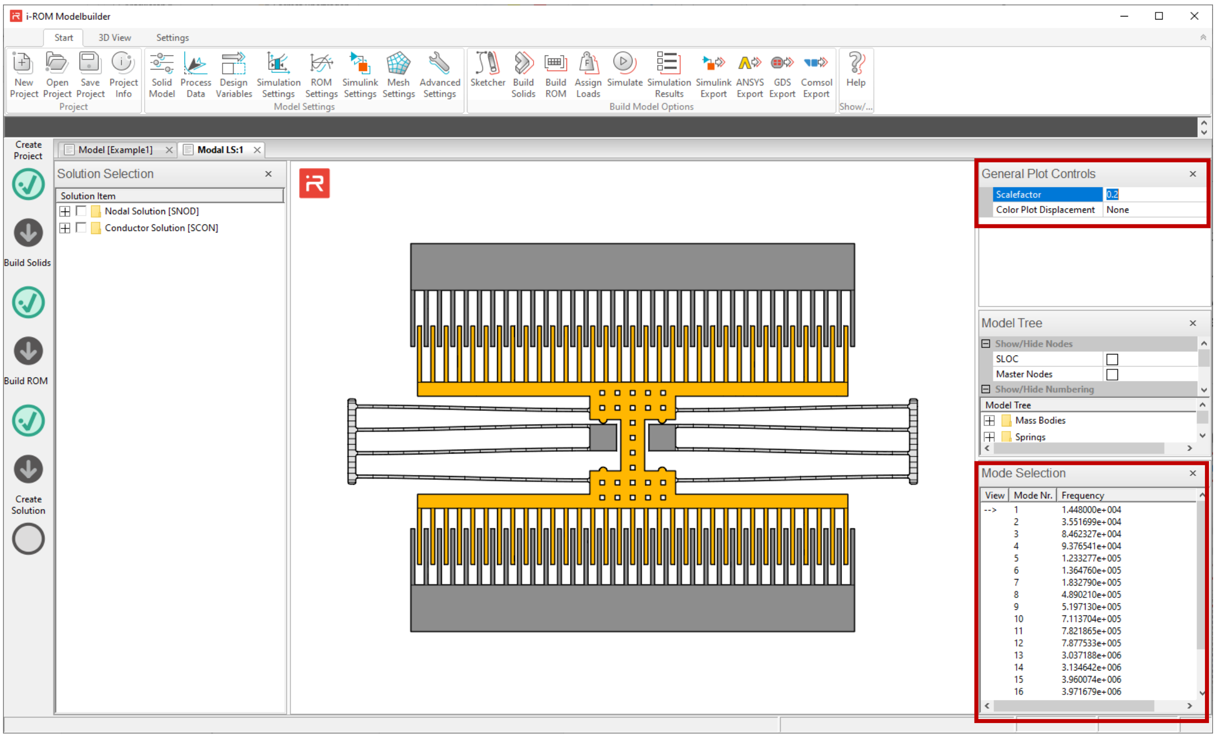
- Run a “Modal Analysis” and calculate twenty eigenmodes
  - Assign the number of modes in the command line: Enter “MOPT, 20”.
  - Start a modal analysis: Enter “SOLV, modal”.
  - There must be a semicolon at the end of each command if multiple commands should be entered by copy and paste on the command line:

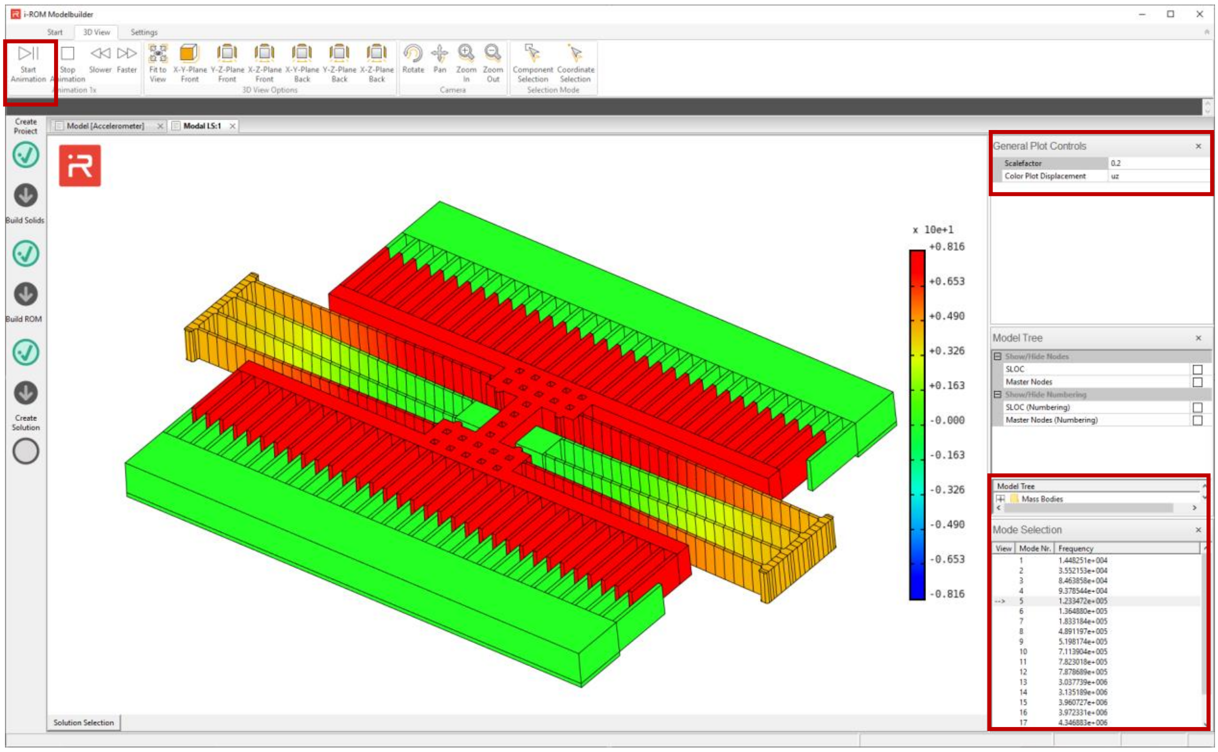
```
MOPT, 20;
SOLV, modal;
```

  - The command line window is marked by the red box below.



➢ Evaluate results of the “Modal Analysis” and create animations
- Select mode #1 in the “Mode Selection” window (use a double click).
- Reduce the scale factor from 1.0 to 0.2 in “General Plot Controls”.
- Click on the “Start Animation” button and re-orientate the structure.
- Select other modes during the animation in the “Mode Selection” window.
- You can change the “Scale factor” and activate “Color Plot Displacement” in the “General Plot Controls” window during the animation.

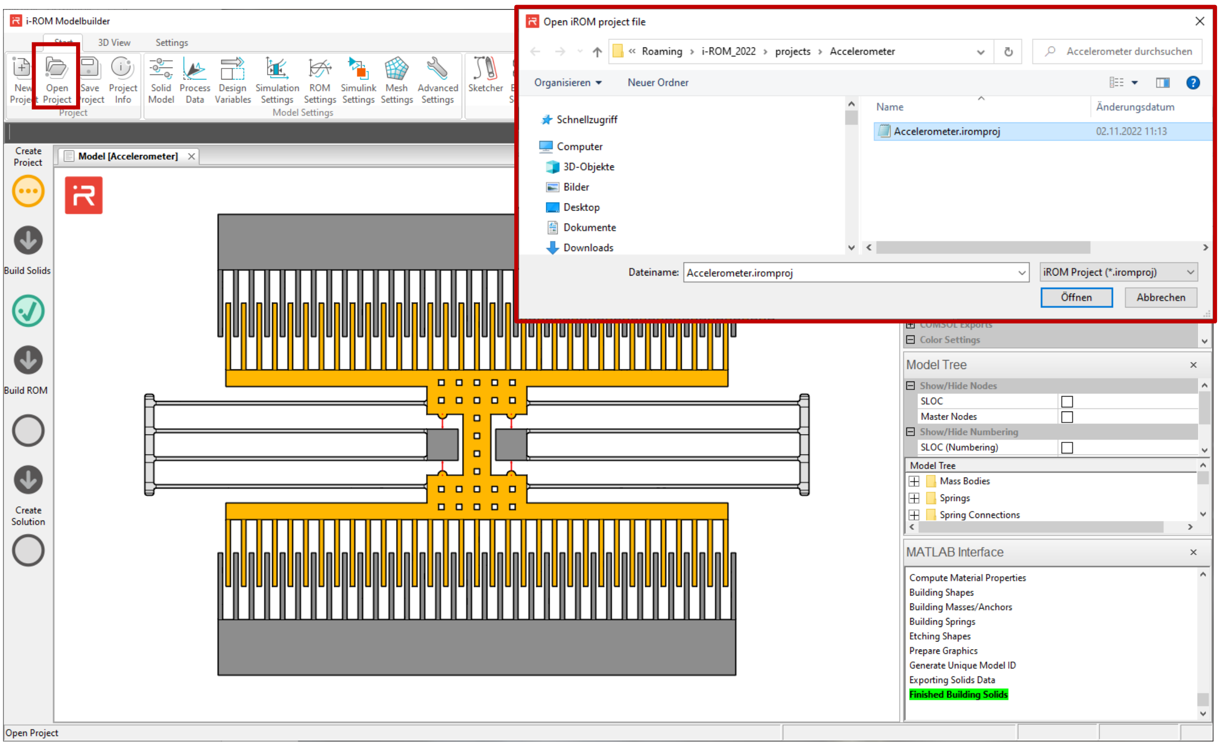




2 Simulation features of the MODELBUILDER

The simulation features will be explained on the Acceleration sensor example discussed in the Example Manual section. The model is similar to the previously discussed example but is extended by additional layout components (e. g. capacitors with gap variation, bottom plate capacitors). The example is given in µMKSV-units (project folder "Accelerometer") and in SI-units (Standard International units are prepared in the project folder "Accelerometer_SI").

➢ Open the existing acceleration sensor example in **µ**MKSV-units
- Click on the "Open Project" icon.
- Open the folder "projects/Accelerometer".
- Select the "Accelerometer.iromproj" file and open it.

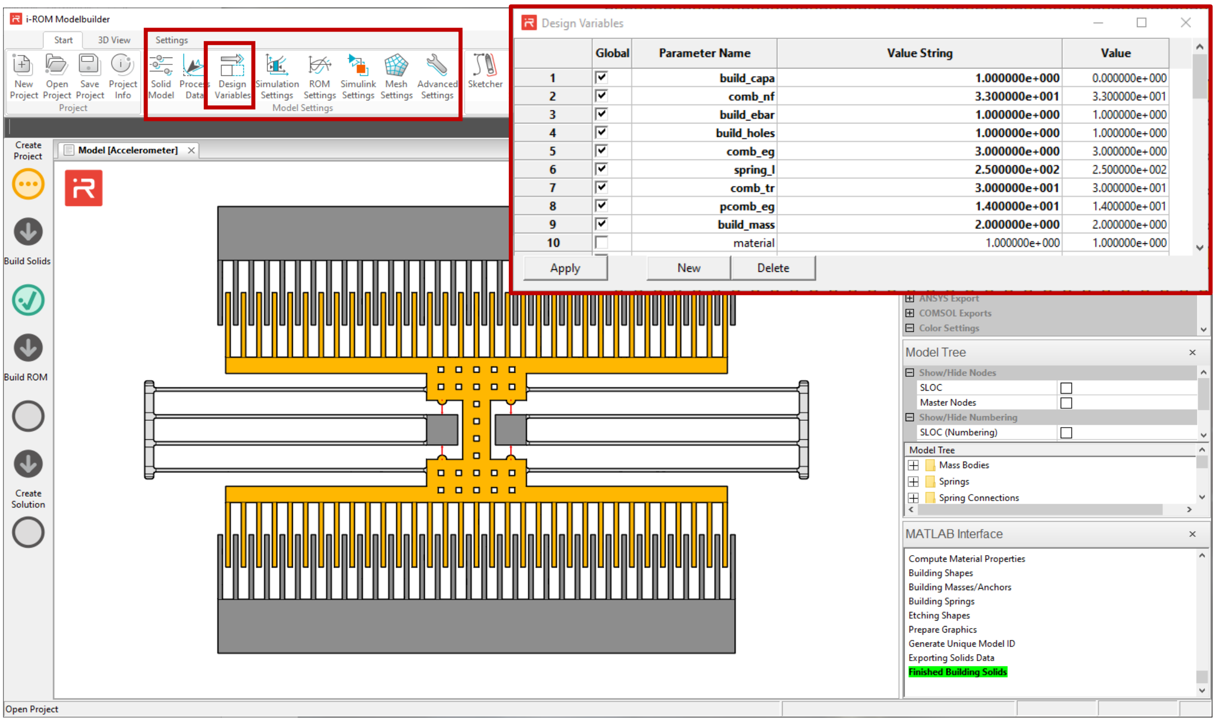


All four types of simulations will be demonstrated on the acceleration sensor model. Combs with gap varying capacitances are assigned first. Then follow several simulation examples which are taken from the Example Manual to demonstrate the required simulation commands and post-processing features:

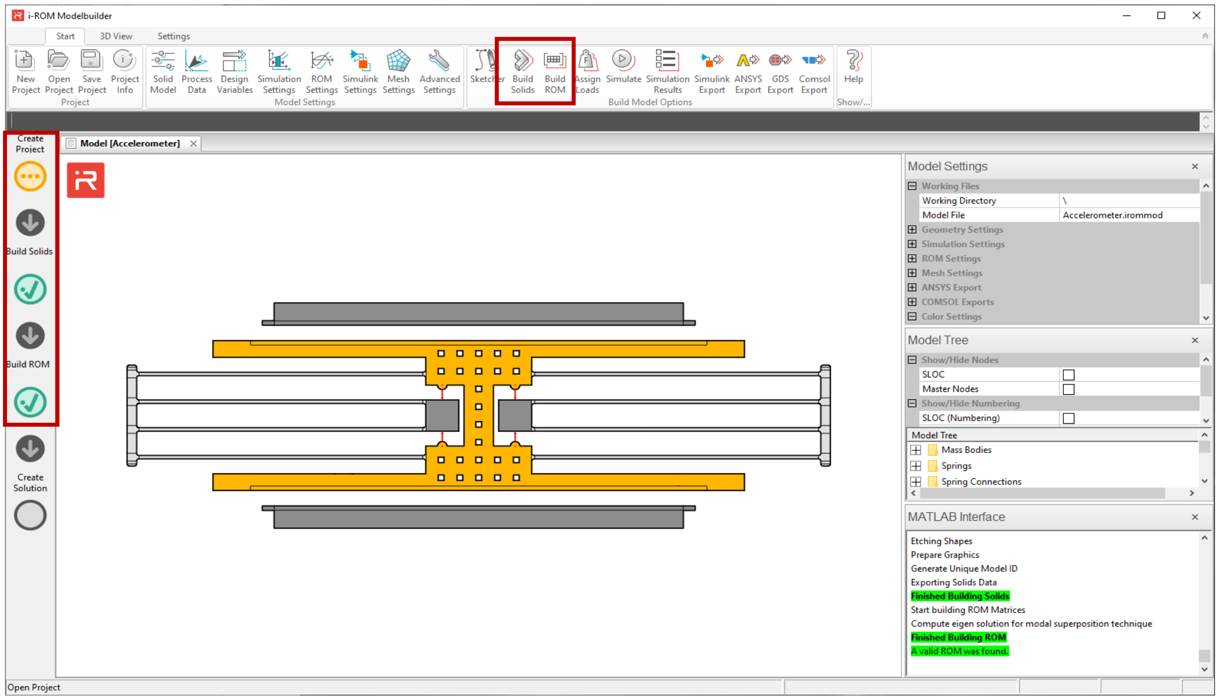
a) Modal analysis, Modal analysis with electrostatic softening (Fig. 15).
b) Static simulations, DC-voltage sweep with pull-in and pull-out (Fig. 16).
c) Harmonic response analysis, AC-sweep simulations (Fig. 18).
d) Transient simulations with pull-in and contact bouncing (Fig. 20).

2.1 Assign combs with gap varying capacitances

- Modify model parameters and simulation settings
  - Model parameters can be set or changed in the “Model Settings” panel.
  - Click on “Design Variables” and change “build_capa” from 0 to 1 (0 comb with area variation, 1 gap variation, 2 bottom plate capacitor).



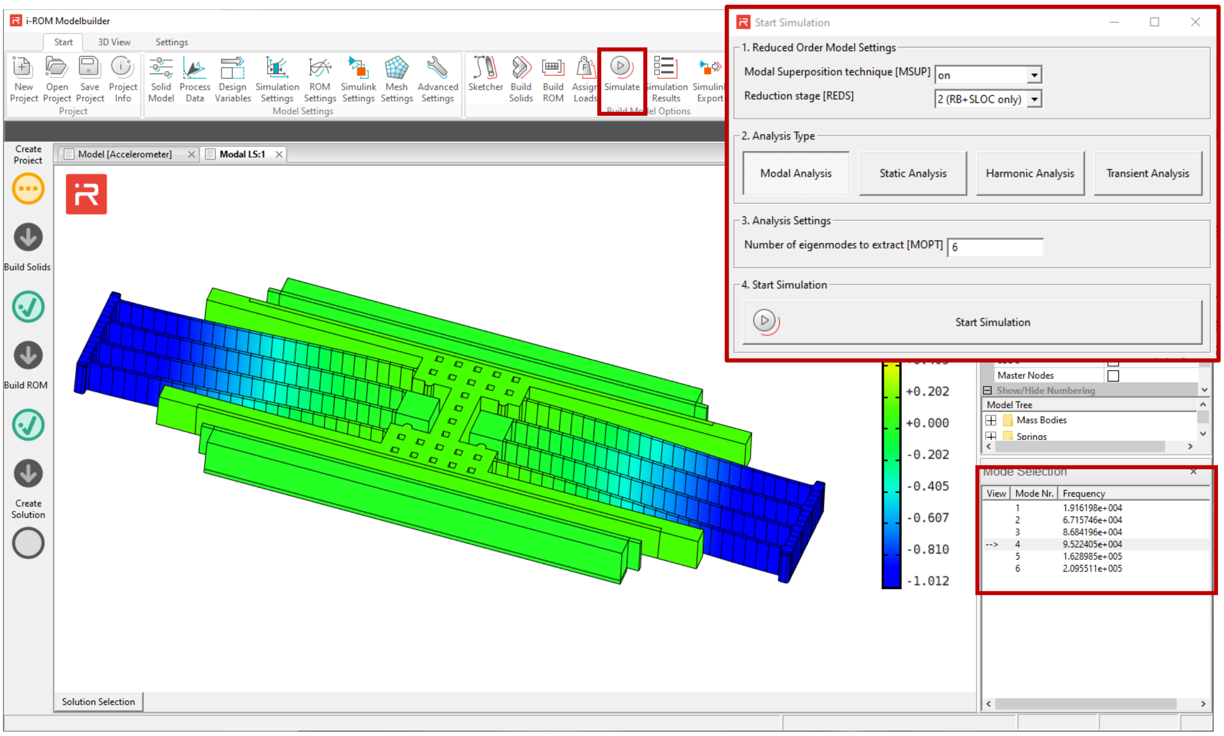
- Click on “Build Solids” and “Build ROM” to generate a new simulation model
  - The commands are finished after the marked circles on the left turn green.



2.2 Modal analysis, Modal analysis with electrostatic softening

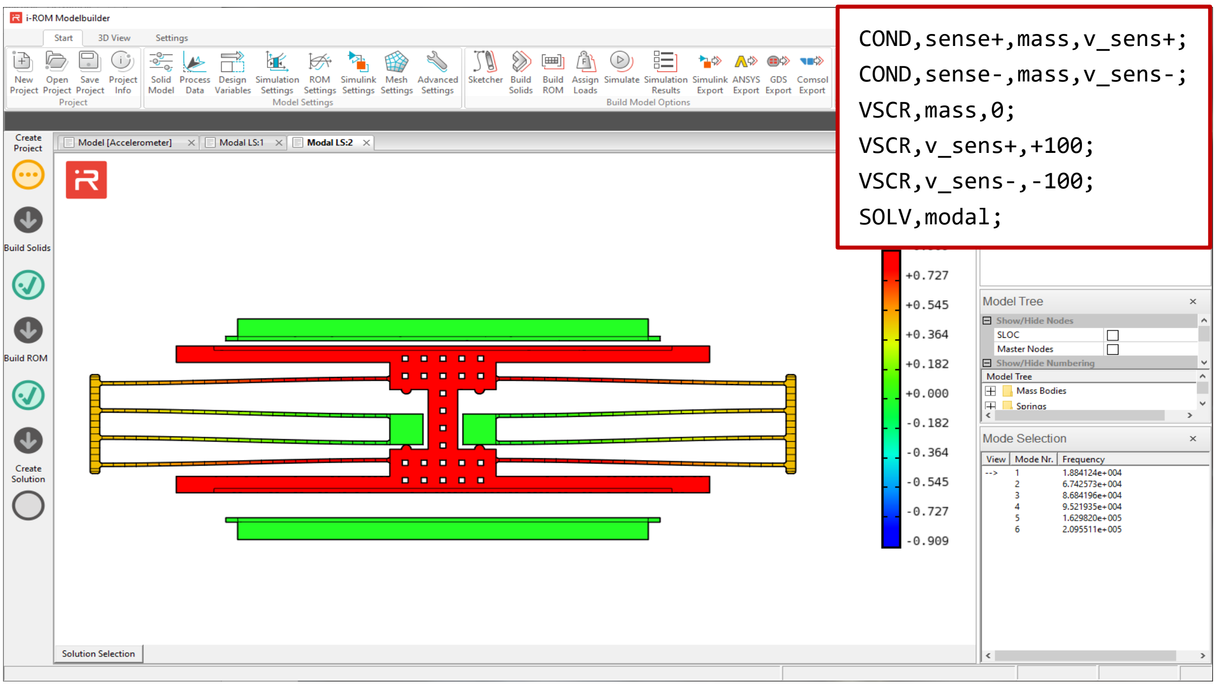
➢ Run a “Modal analysis” in the mechanical domain

- Enter “SOLV, modal” (command-based approach) or click on the “Simulate” icon and “Start Simulation” (GUI-based approach). Compare results with Fig. 15 of the Example manual.
- Select mode #4 and apply a scale factor of 0.4.



➢ Run a “Modal analysis with electrostatic softening” effects

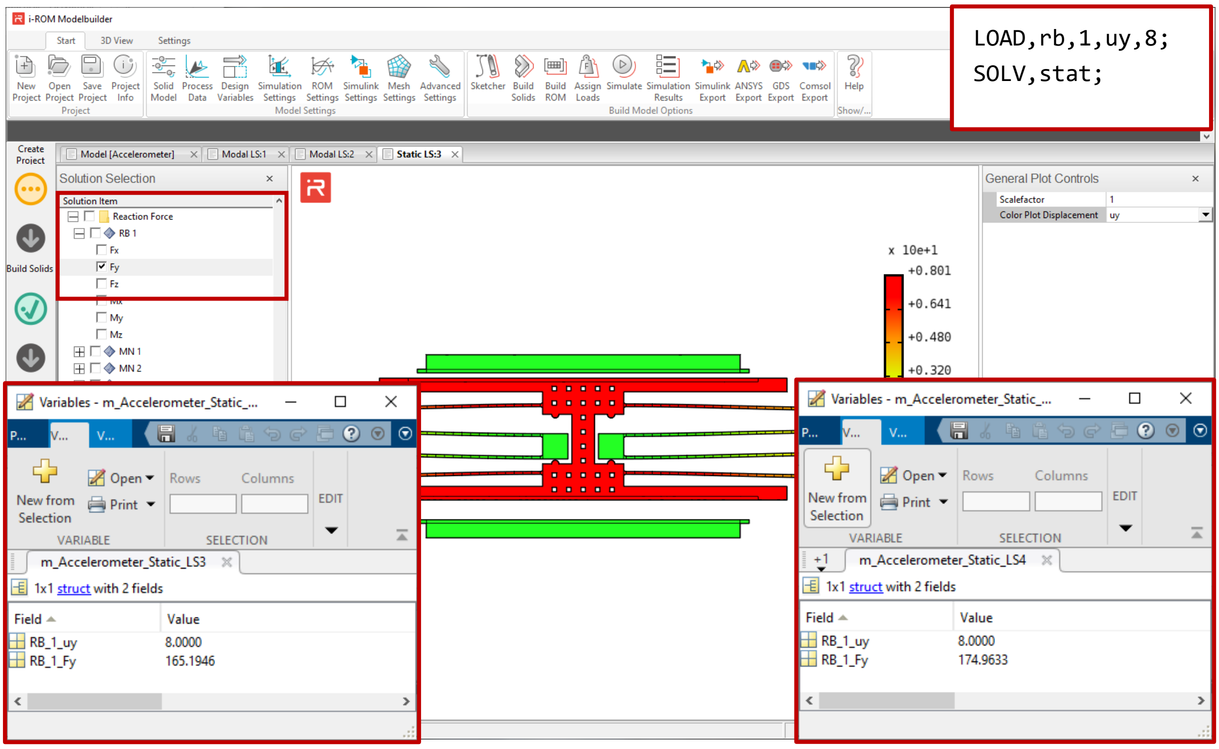
- Enter the commands from the red box and compare results with Fig. 15.



2.3 Static simulations, DC-sweep simulations (pull-in, pull-out)

➢ Run a “Static simulation” in the mechanical domain

- Calculate the stiffness of rigid body #1 in uy-direction according to Fig. 16.
- The reacting force is 165 µN if voltage loads are still applied from the previous load step (electrostatic softening). Otherwise the force is 175 µN.

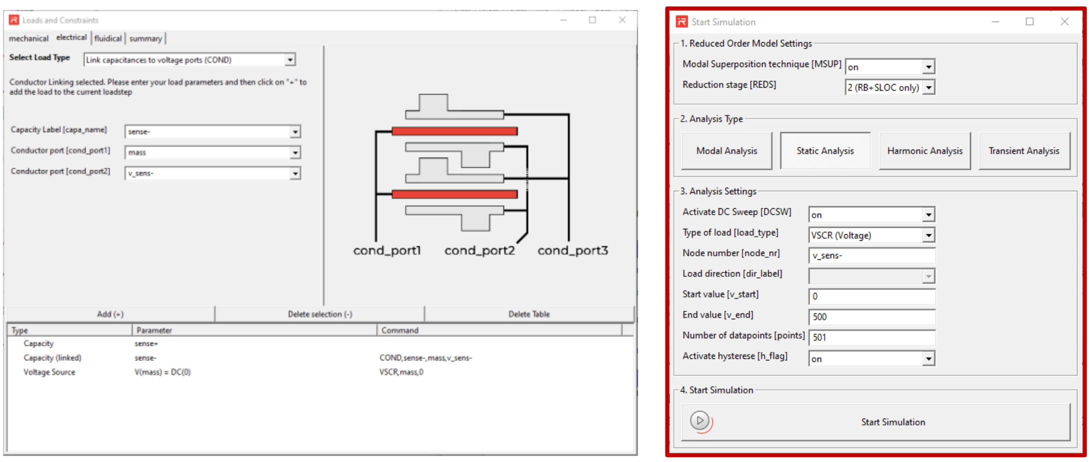


➢ Run a “DC-voltage sweep” with pull-in and pull-out effects

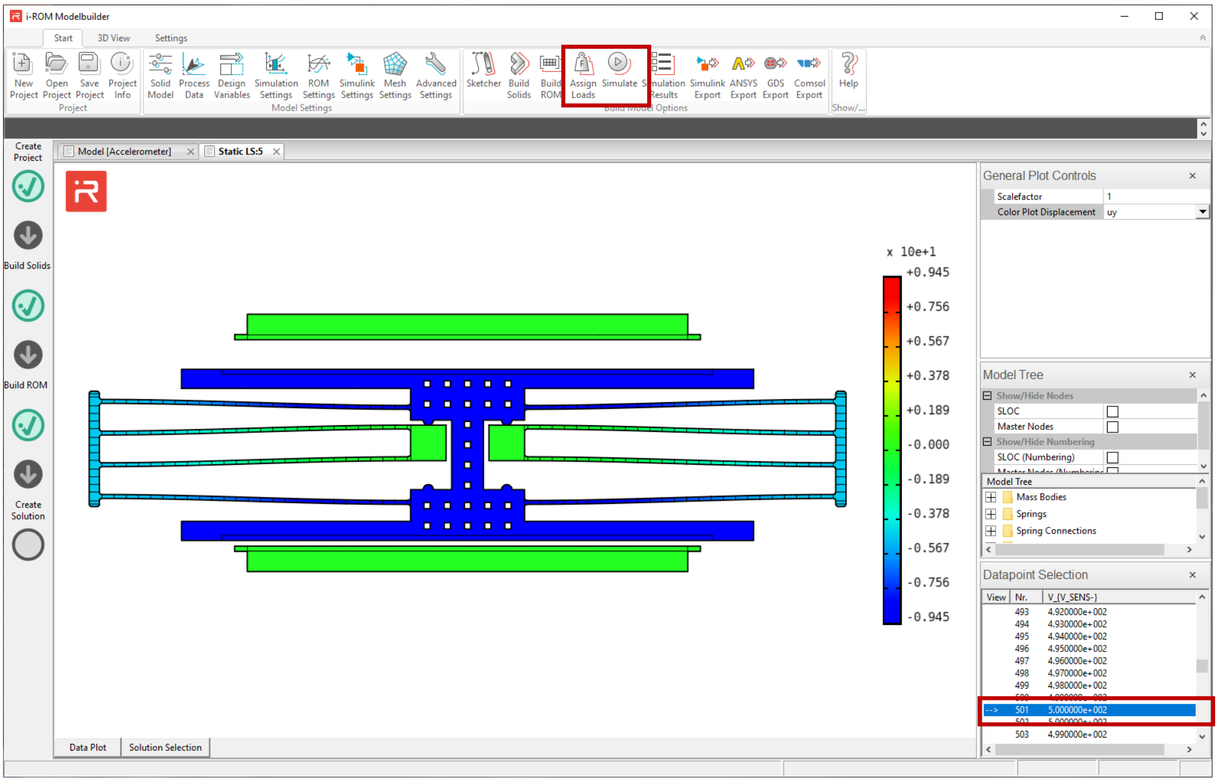
- Enter the following commands and compare results with Fig. 16.

```
COND,sense-,mass,v_sens-;
VSCR,mass,0;
DCSW,vscr,v_sens-,,0,500,501,1;
SOLV,stat;
```

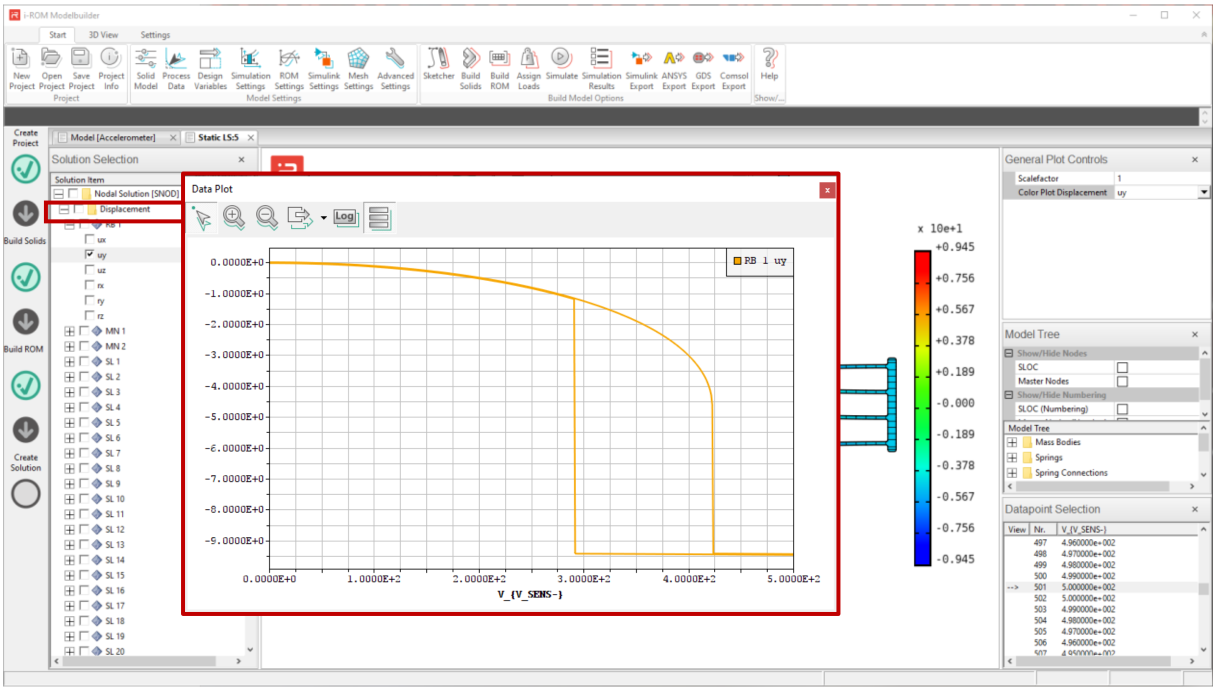
- The commands can also be assigned with the Loads and Constraints window and Start Simulation window as shown in the figures below.



- Select sample #501 in the “Datapoint Selection” window.

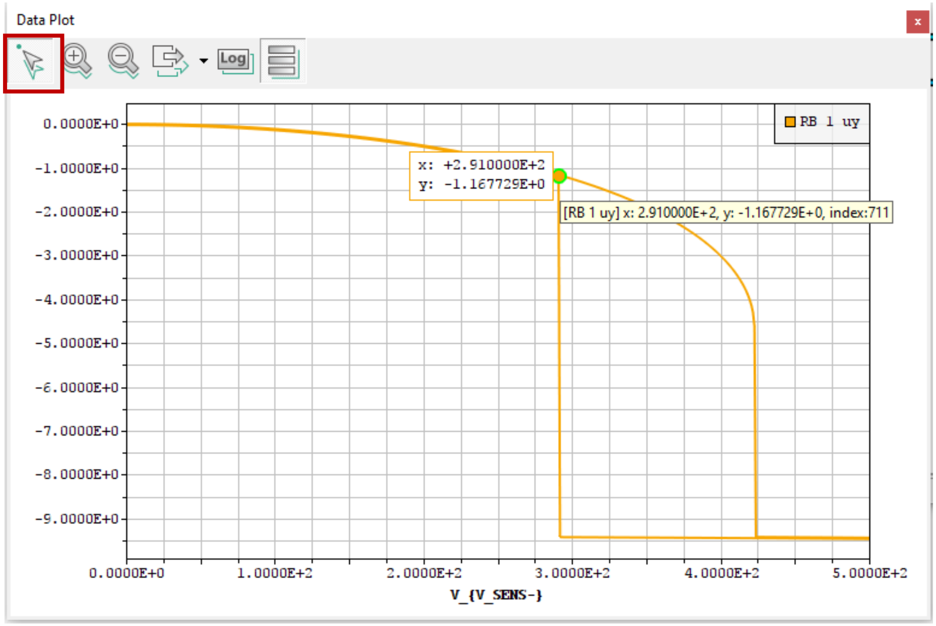


➢ Plot the voltage-displacement-relationship
  - Select uy-DOF at RB 1 in the “Solution Selection” window for a curve plot.
  - Pull the “Data Plot” window into the center of the screen for better visualization.

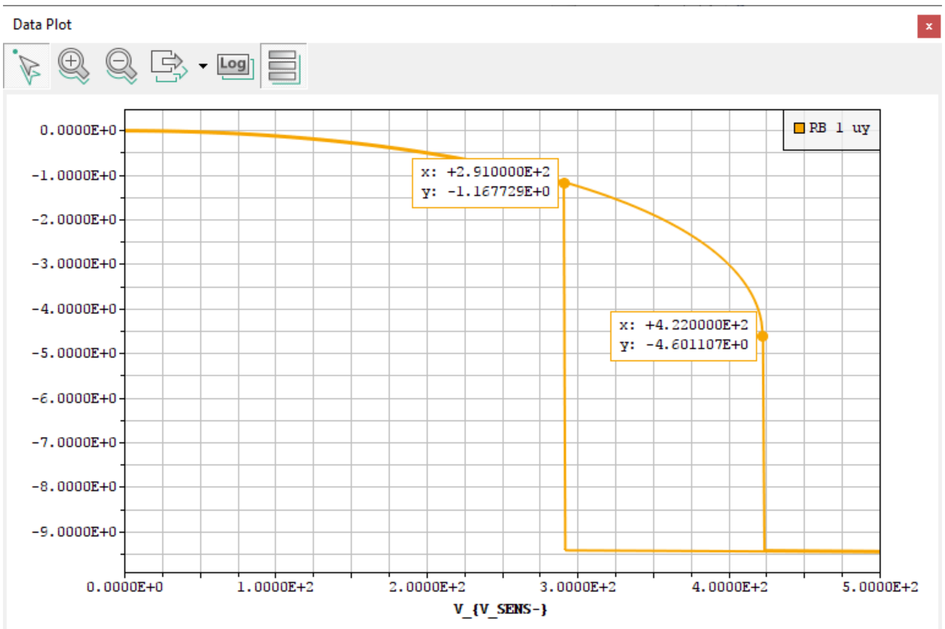


➢ Set data point markers at the pull-in and pull-out voltages

- The left data point selection icon of the “Data Plot” window must be active.
- Move the mouse along the orange curve. Datapoint coordinates follow the mouse pointer.
- Stop at the data point of interest. The yellow label box appears.
- Click on the right mouse button and select “Create Datapoint”.
- A permanent datapoint appears at the pull-out voltage of 291 V.



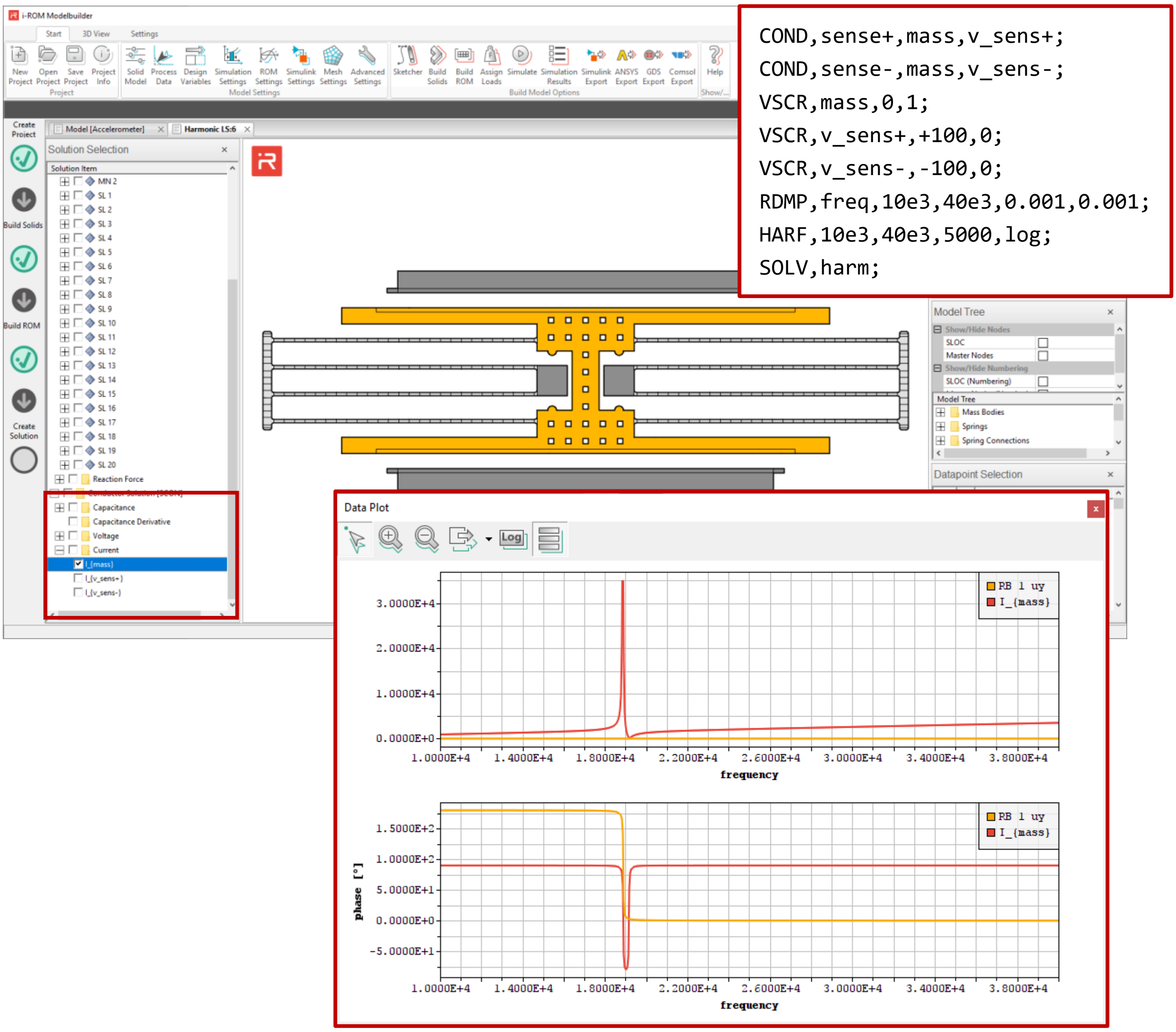
- Select the next data point at the pull-in voltage of 422 V.



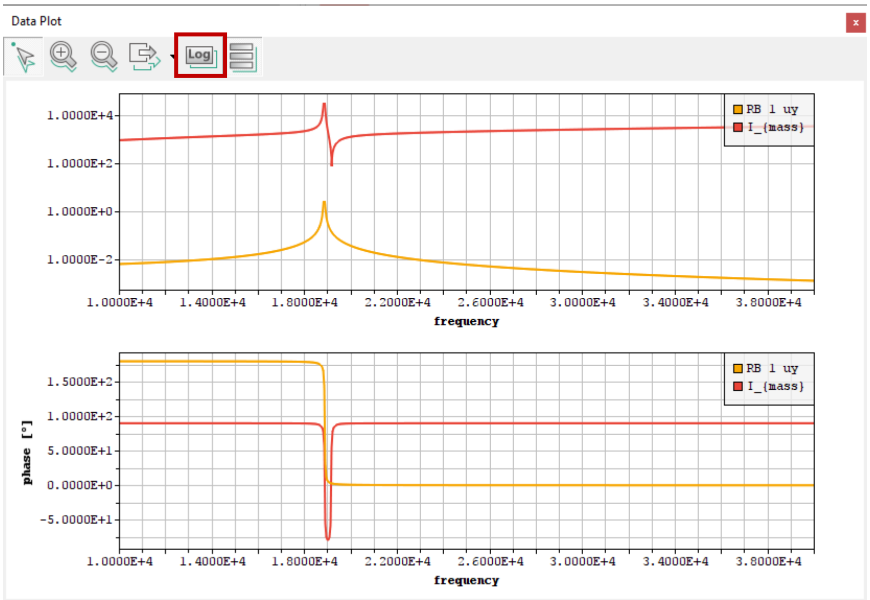
- Delete a data point: Go to the first data point, click on the right mouse button and select “Delete Datapoint”.
- Click on “Start Animation” of the “3D View” panel to animate the pull-in and pull-out behavior of the 3D-model.
- The pull-in and pull-out curve consist of 1002 samples. You can use the “Slower” and “Faster” button in the GUI to adapt the speed of the animation. On average, 30-50 samples are processed per second on a PC.

2.4 Harmonic response analysis, AC-sweep simulations

➢ Run a “Harmonic response analysis” of the electro-mechanical system
  - Enter the commands of the red box and compare results with Fig. 18.
  - Select uy-DOF at RB 1 and the electrical current at the “mass” port in the “Solution Selection” window for visualization in the “Data Plot” window.

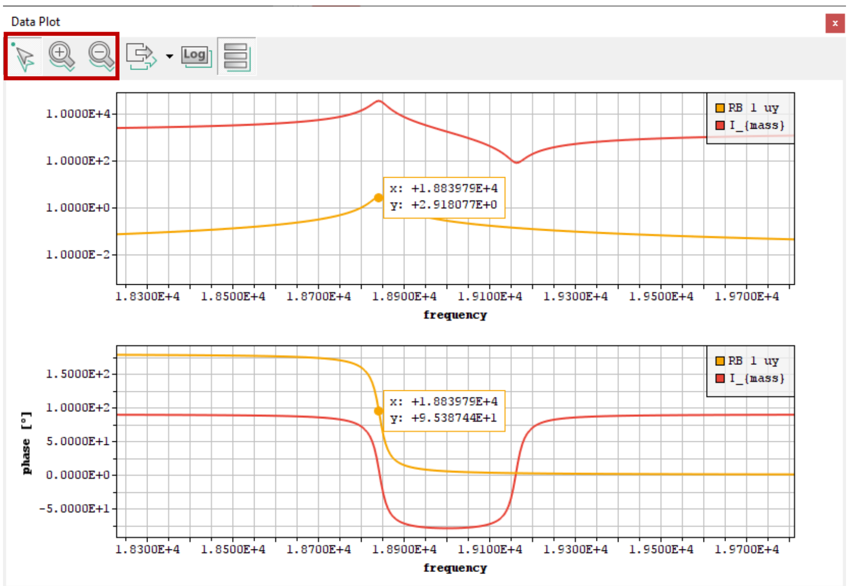


- Select logarithmic x- and y-axes scaling. Click multiple times in the "LOG" icon to toggle between linear and logarithmic x-y-axes.

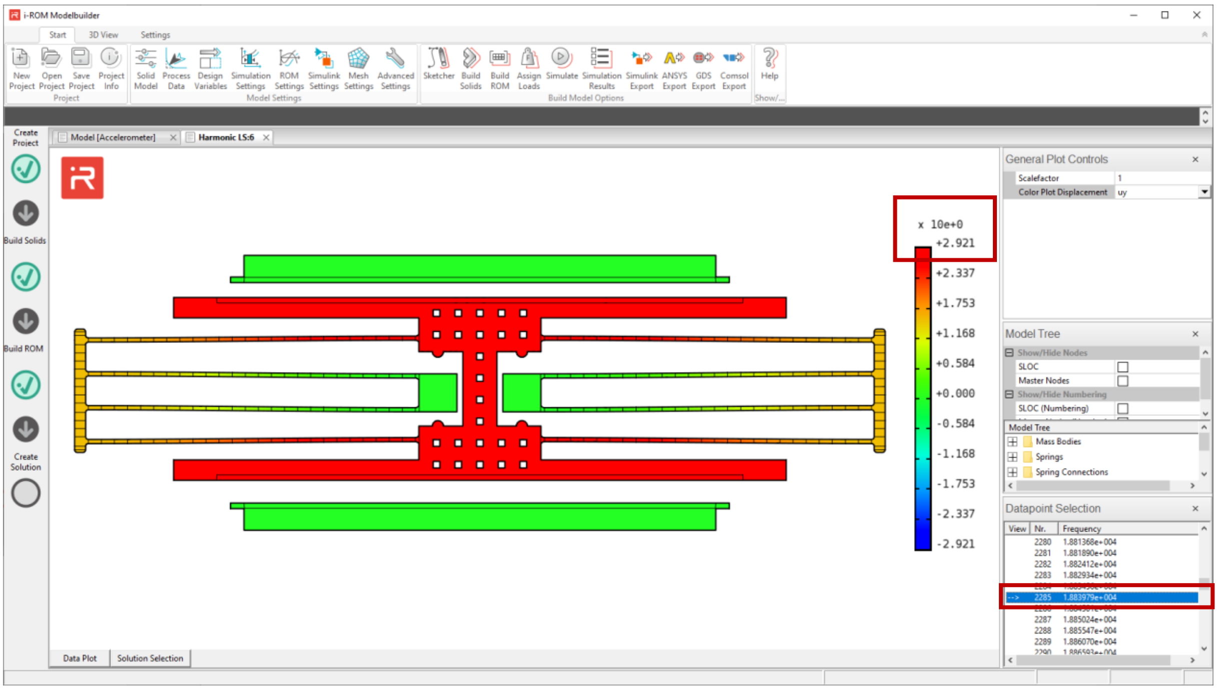


➢ Manipulate the curves (zoom, pan) of the "Data Plot" window

- The left "SELECT" button of the "Data Plot" window must be active.
- ZOOM: Move the mouse pointer into the region of the 2D-plot where you want to zoom. Scroll the mouse wheel to zoom in or out.
- PAN: Zoom into the 2D-plot. Keep the left mouse button pressed and move the mouse to pan the curves in x- or y-direction.
- Alternatively, you can use the "Zoom-In" and "Zoom-Out" icons.

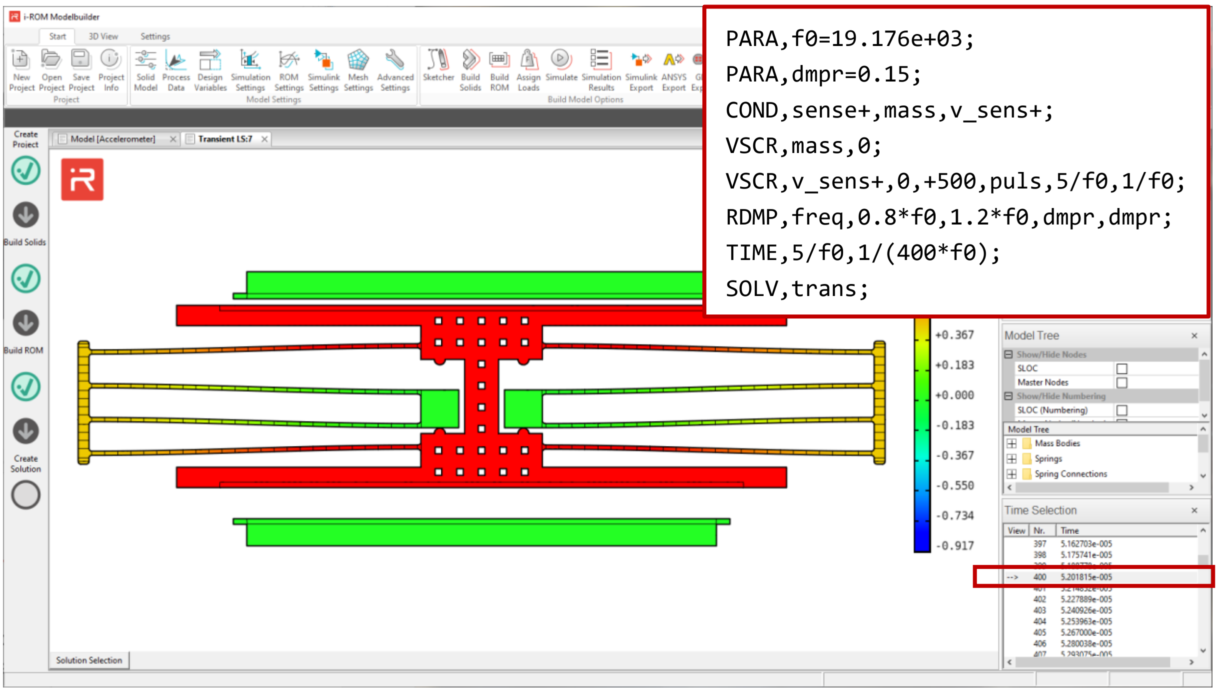


➢ Create a colored amplitude plot of the sample at resonance
- Select sample #2285 at 18.839 kHz in the “Datapoint Selection” window.
- The amplitudes of all sample frequencies can be animated.



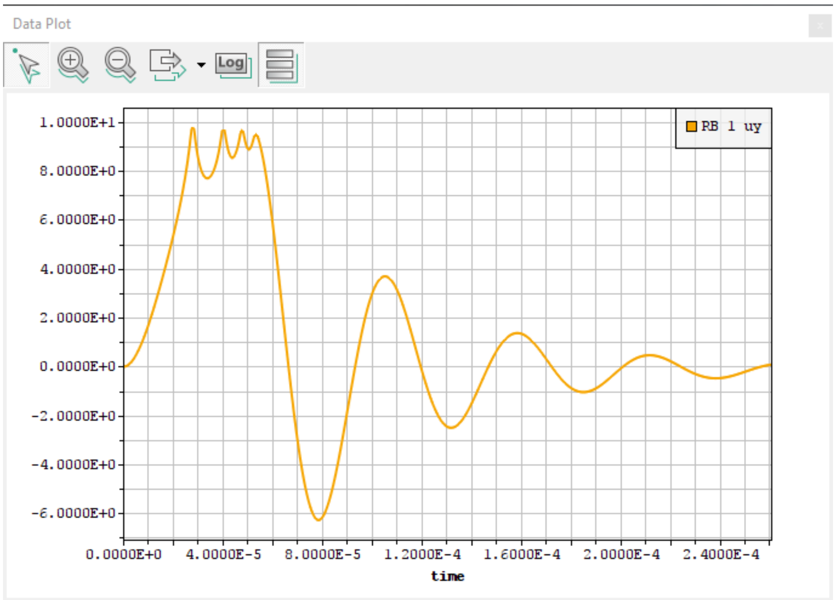
2.5 Transient simulations with pull-in and contact bouncing

➢ Run a “Transient simulation” of a voltage pulse with contact bouncing
- Enter the commands of the red box and compare results with Fig. 20.

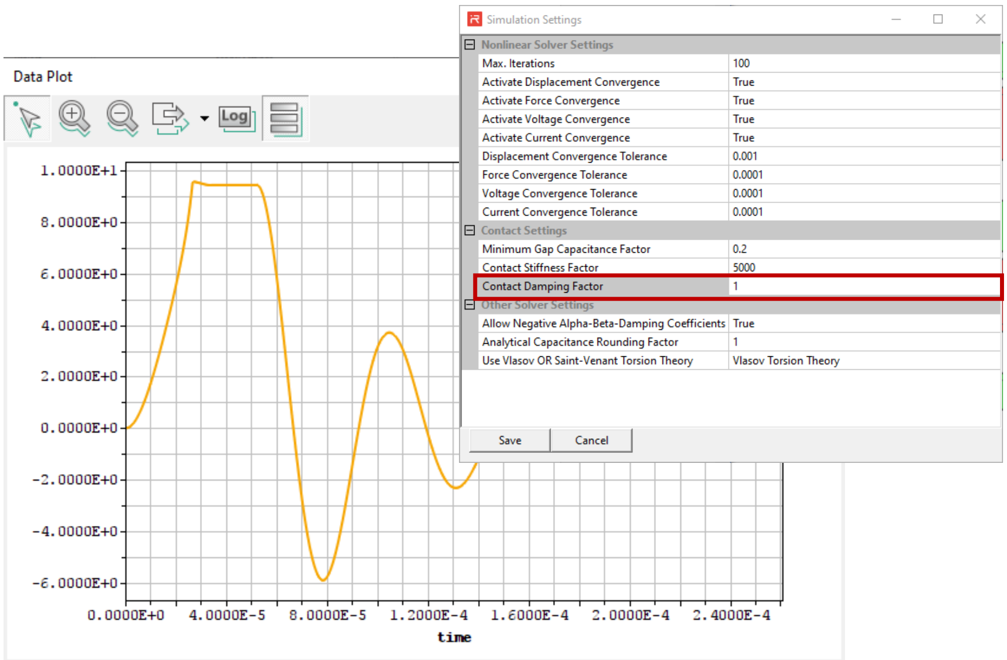



```
PARA,f0=19.176e+03;
PARA,dmpr=0.15;
COND,sense+,mass,v_sens+;
VSCR,mass,0;
VSCR,v_sens+,0,+500,puls,5/f0,1/f0;
RDMP,freq,0.8*f0,1.2*f0,dmpr,dmpr;
TIME,5/f0,1/(400*f0);
SOLV,trans;
```

- The contact stiffness and damping factor can be changed in the Simulation Settings window of the GUI or directly in the model input file. Stopper data (see STOP command) specified in the user model file have a higher priority.
- Plot uy-DOF results of rigid body #1 over time.



- Enlarge the damping factor from 0.0 to 1.0 to eliminate bouncing.
- Click on "Build ROM" to create a new simulation model with changed contact settings and re-run the previous simulation.

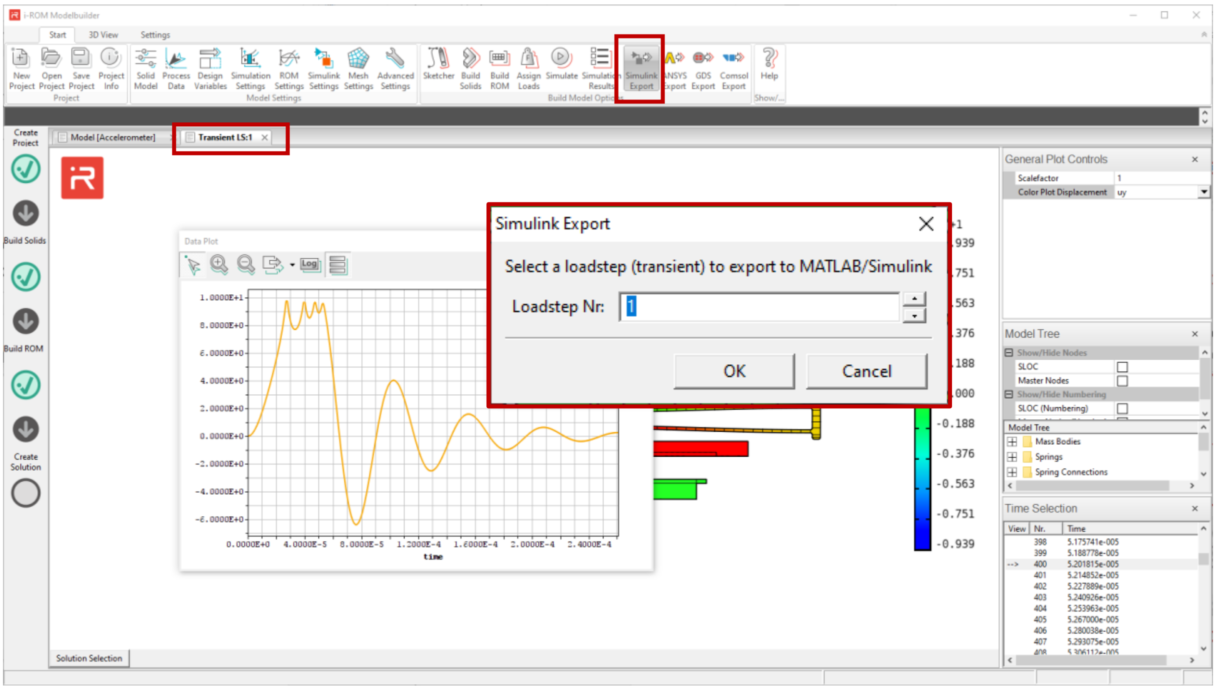


➢ Animate the transient response with contact bouncing effects

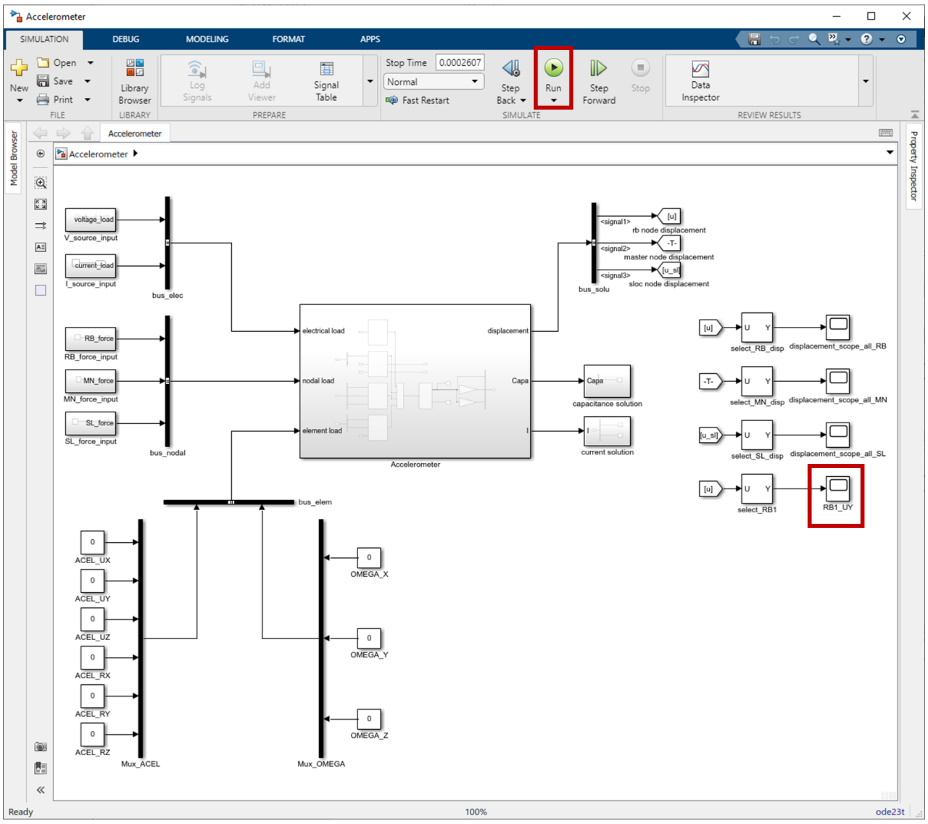
3 Model export to Matlab/Simulink® (Interface to SIMULINK®)

➤ Create a "SIMULINK model" of the last transient simulation run

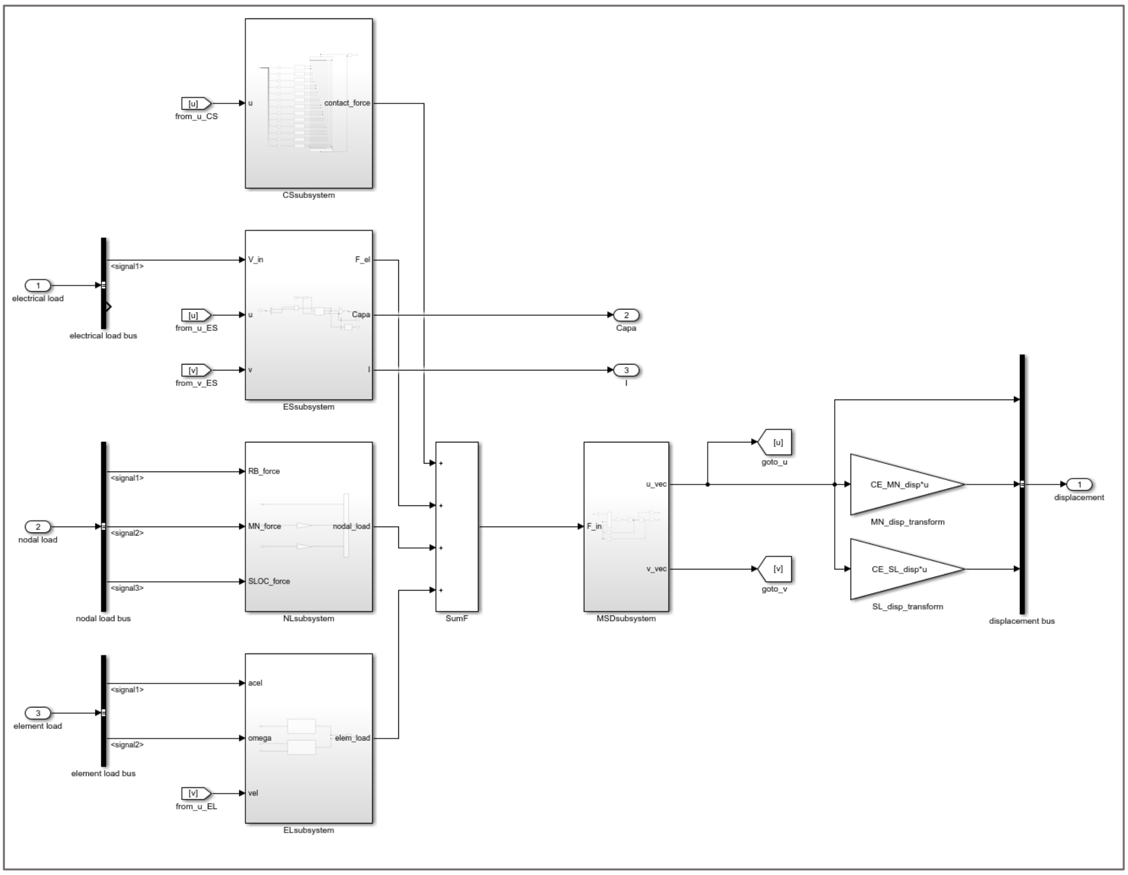
- Click on the "Simulink Export" icon. Select the last load step specified in the results window (Transient LS:1, → specify 1).



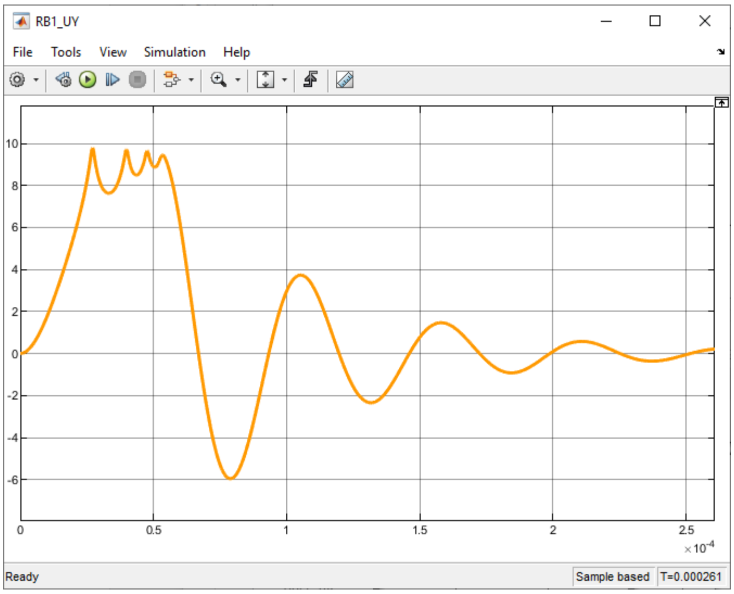
➤ Run the SIMULINK model and plot rigid body displacements



The following figure shows details of the accelerometer subsystem



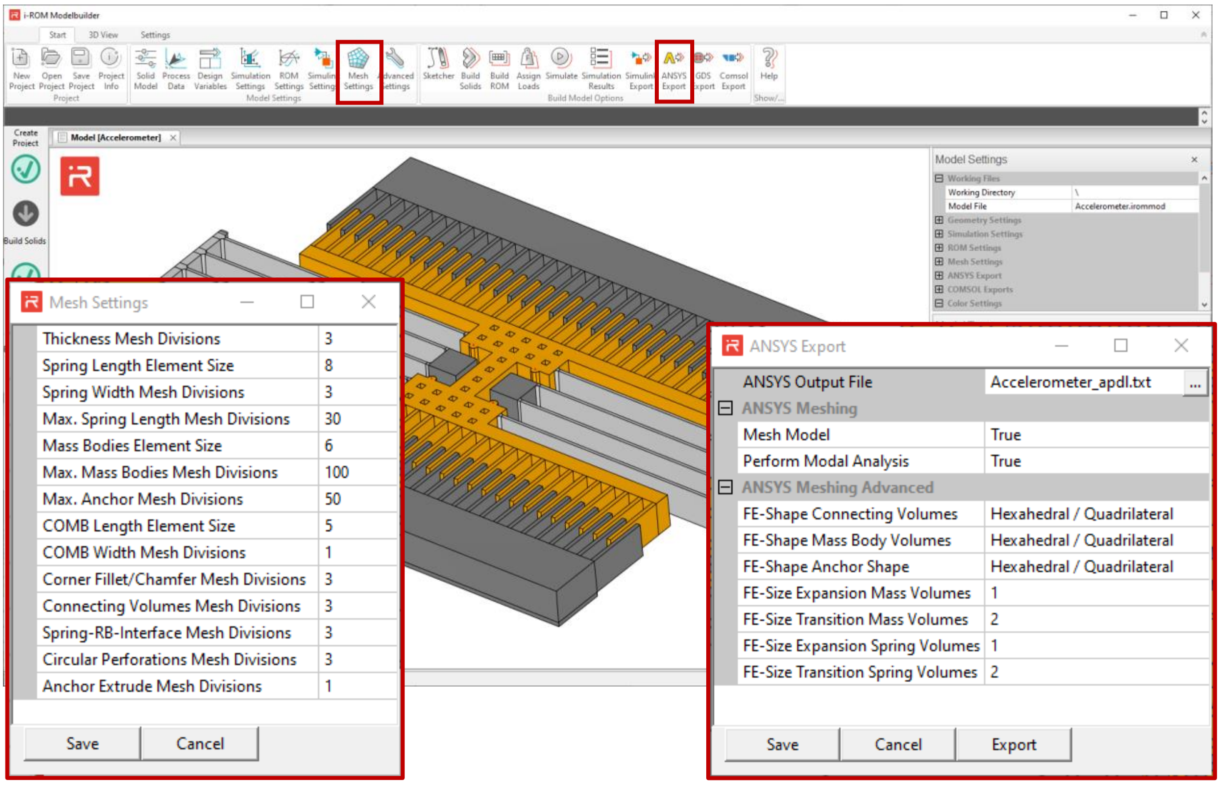
- Visualize transient Simulink results data
  - Mechanical domain results are available for rigid bodies (RB), master nodes (MN) and spring location points (SL). The capacitances and the current flow can be observed for all ports in the electrical domain.
  - Click on the “RB1_UY” scope block to plot the displacement response.



4 Model export to ANSYS® (Interface to ANSYS®)

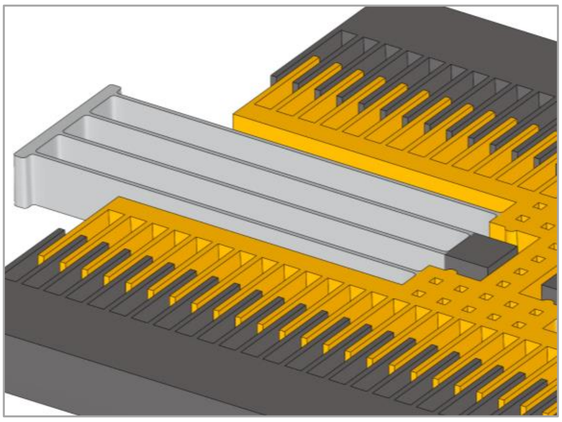
➢ Create an “ANSYS model” of the accelerometer example

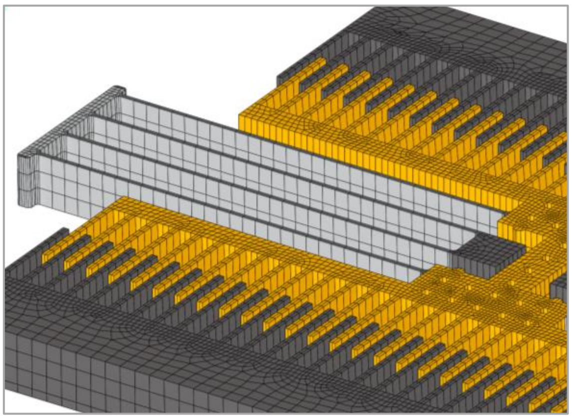
- Set the “build_capa” parameter to zero in the Design Variables window.
- Build a new solid model with comb fingers (click on “Build Solids”).
- Default settings for the ANSYS mesh can be changed in the Mesh Settings window.



➢ Export the acceleration sensor model to the finite element tool ANSYS

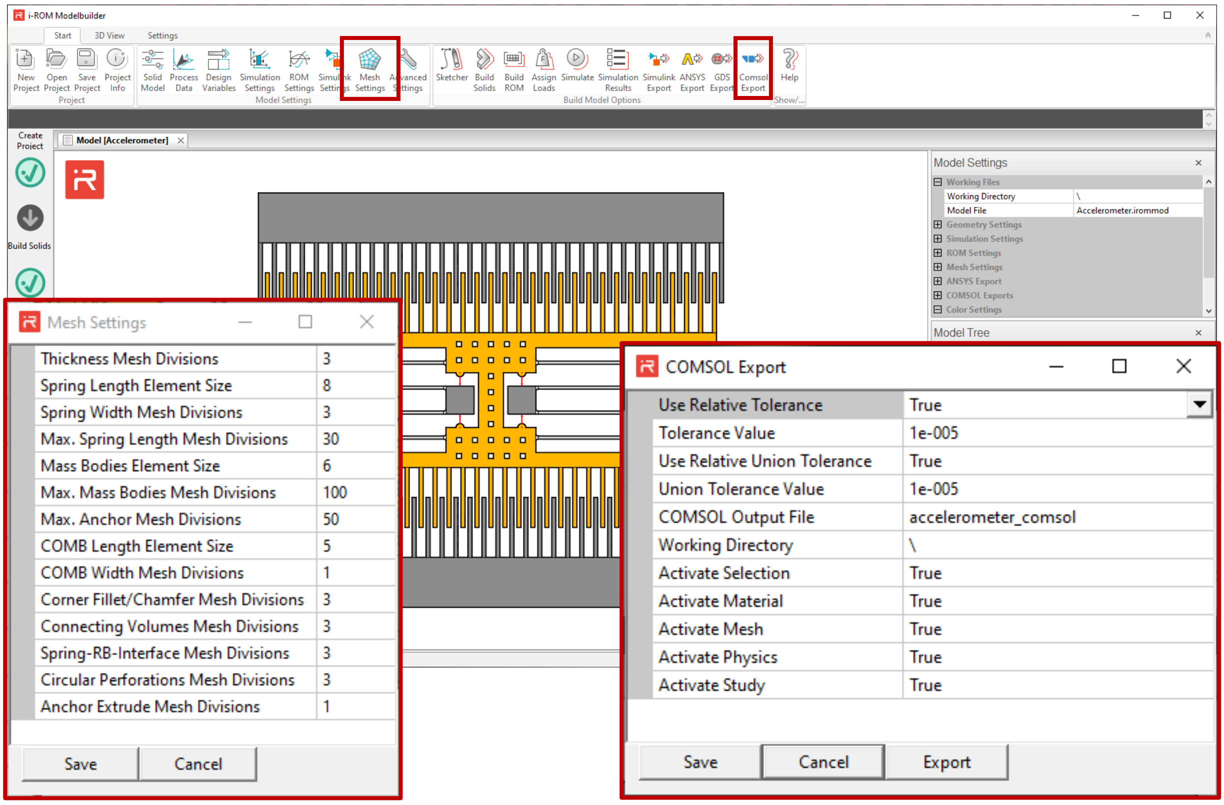
- Click on the “ANSYS Export” icon.
- A text file “Accelerometer_apdl.txt” appears in the working directory.
- Run the User model input file in ANSYS. The following screen shoots show the exported solid model and the finite element model in ANSYS.



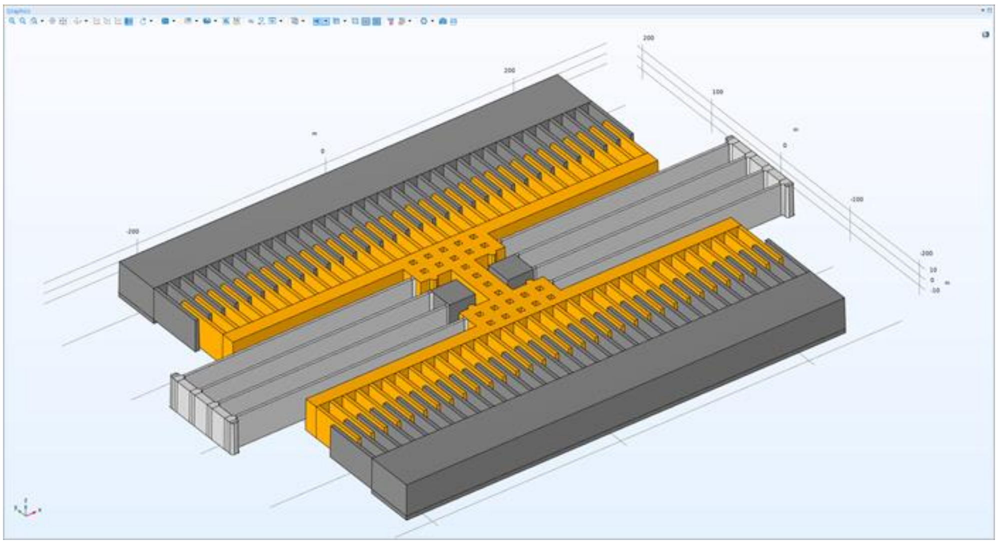


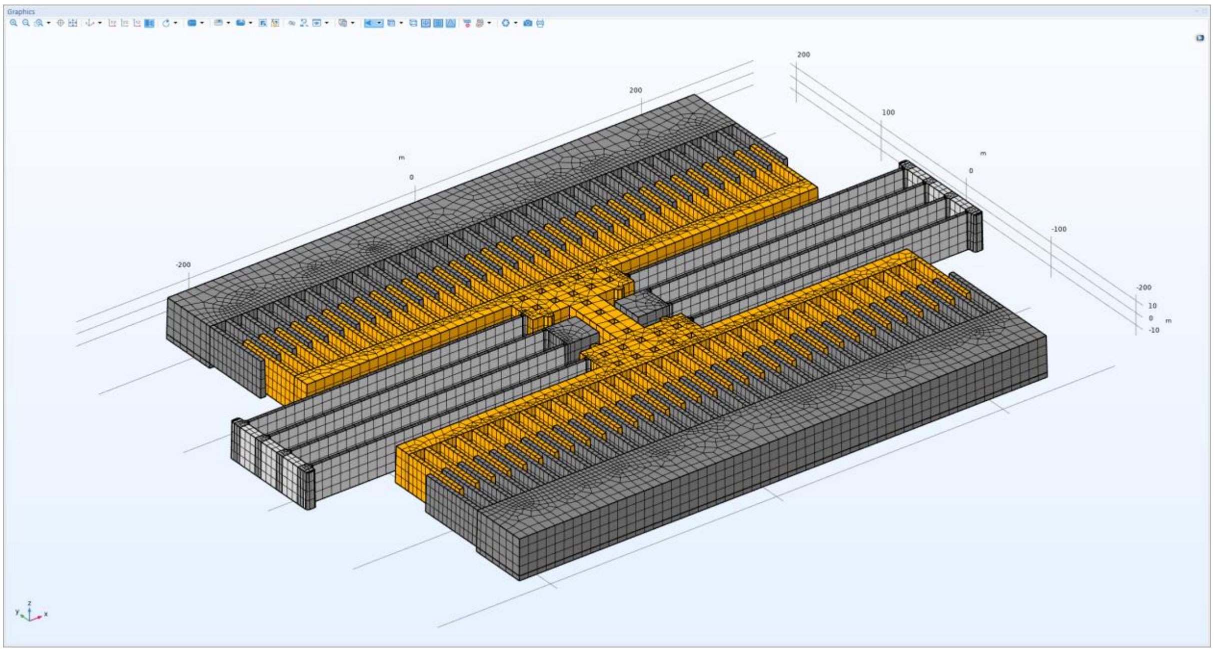
5 Model export to COMSOL® (Interface to COMSOL®)

➢ Create an "COMSOL model" of the accelerometer example
- Default settings for the COMSOL mesh can be changed in the Mesh Settings window.



➢ Export the acceleration sensor model to the finite element tool COMSOL®
- Click on the "COMSOL Export" icon.
- Three files with the file name "accelerometer_comsol" appear in the working directory. Extensions are *.class, *.java and *.txt.
- Open the *.class file in COMSOL® and follow the instructions.

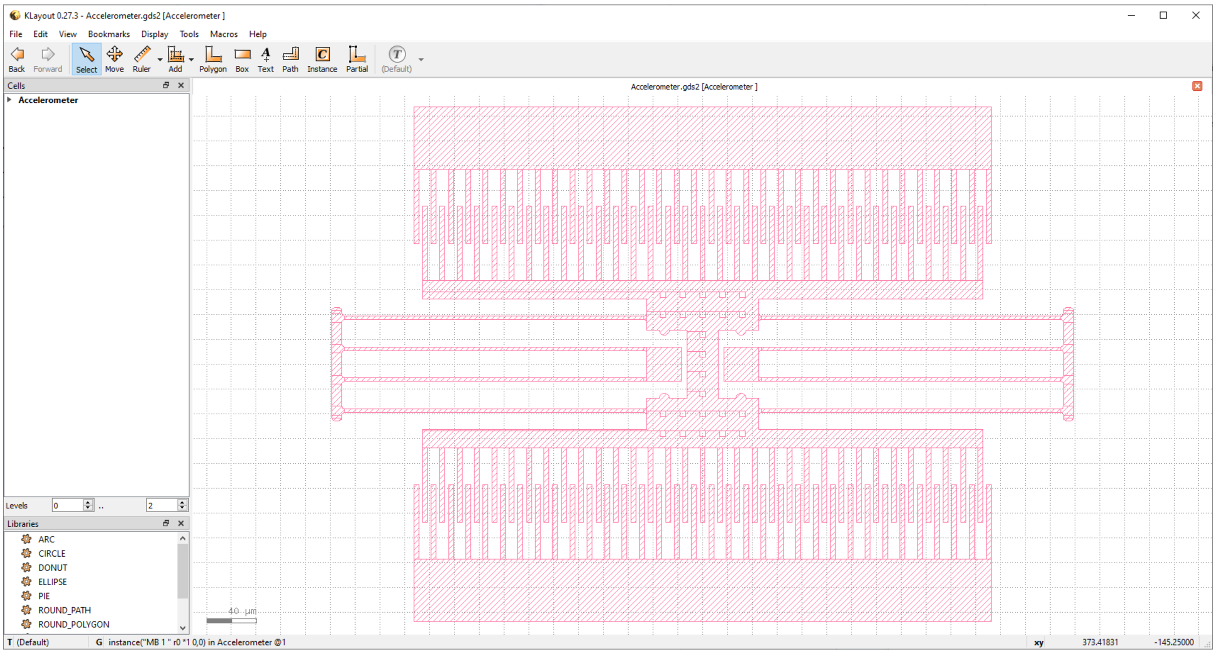




6 Create a Mask Layout (GDS Interface)

➢ Create a mask layout of the accelerometer example

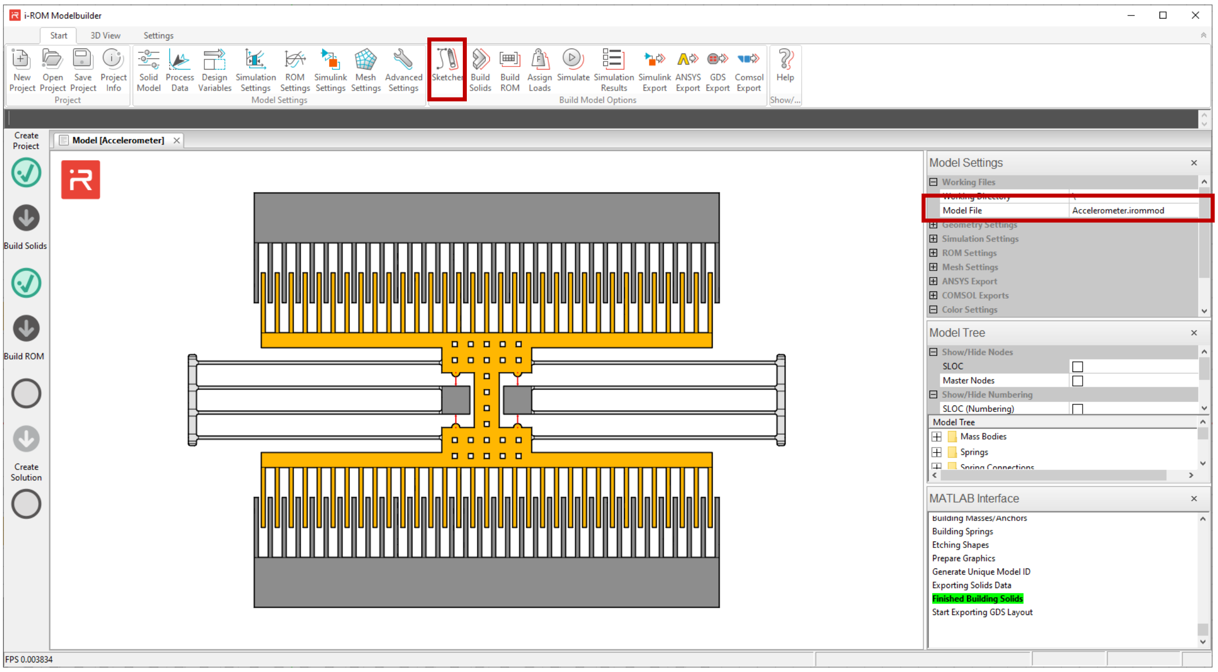
- Click on the “GDS Export” icon. Assign a system of units for the GDS-data and define the number of divisions for curved lines. Curved lines are represented by polygonal lines in the GDS2 output format.



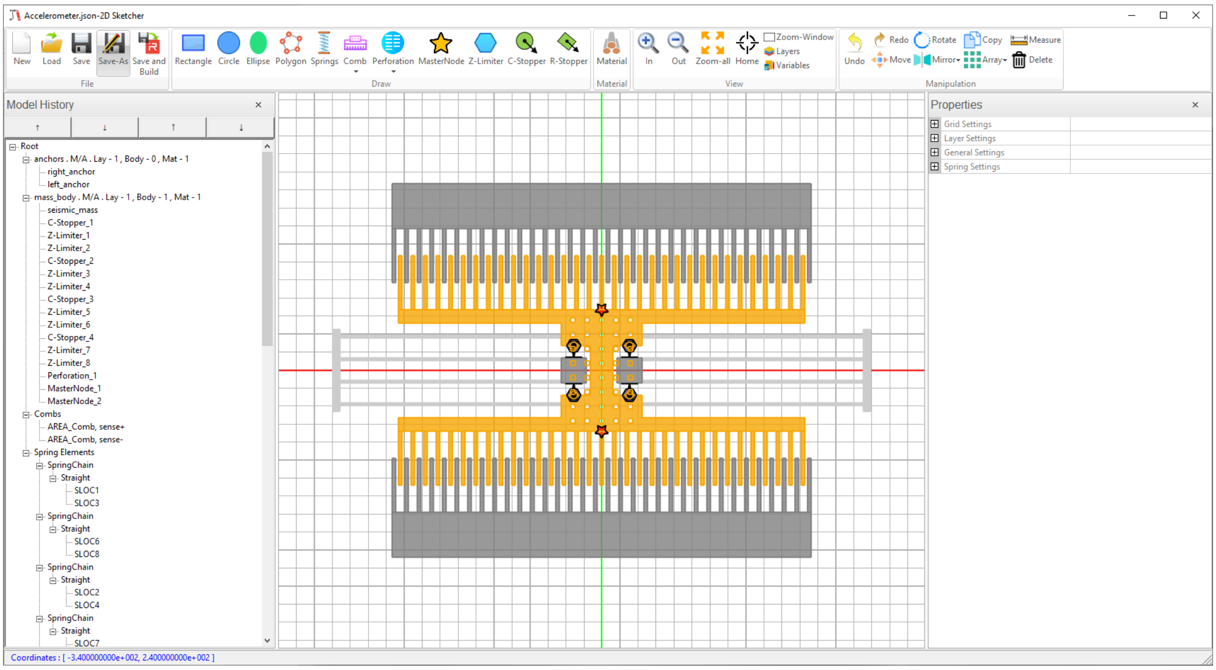
7 Graphical Model manipulation (SKETCHER Interface)

➢ Graphical model manipulation

- The existing accelerometer model can by automatically transferred the 2D-Sketcher. Click on the “Sketcher” icon.

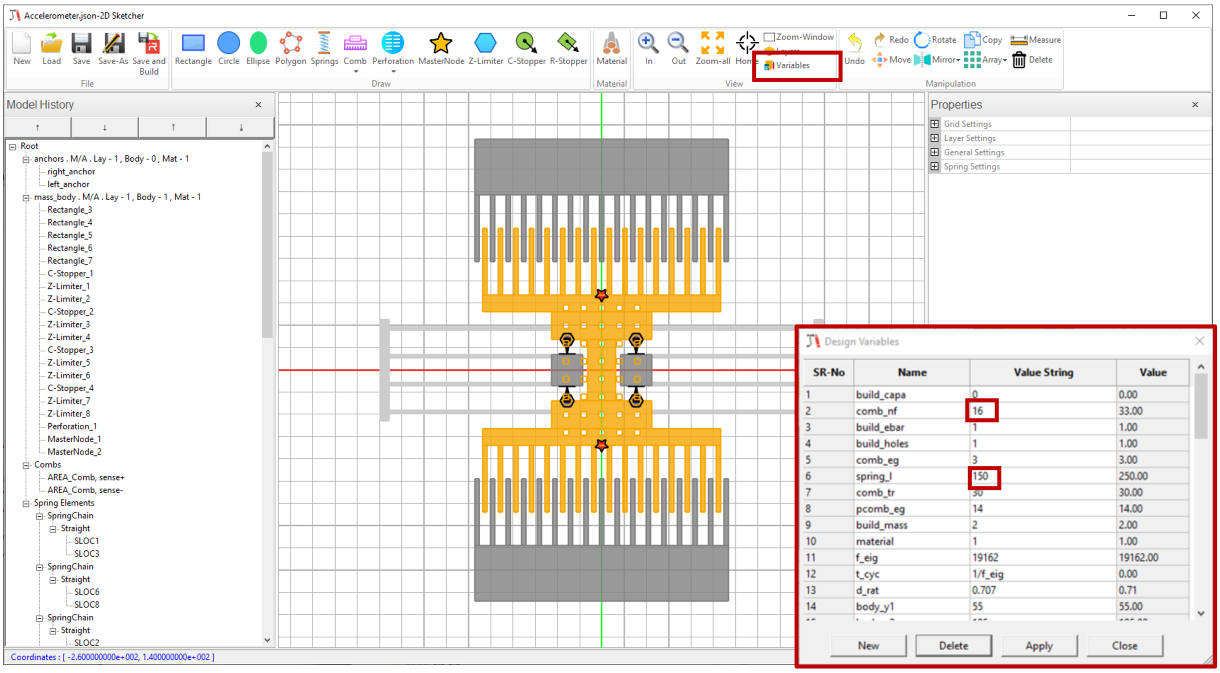


- All model items are translated to an “accelerometer.json“ file and the 2D-Sketcher window appears on the screen.
- Numerical values and design parameters for geometrical dimensions are retained in the new model. Conditional model items defined by *IF commands are replaced by the current settings of the MODELBUILDER.

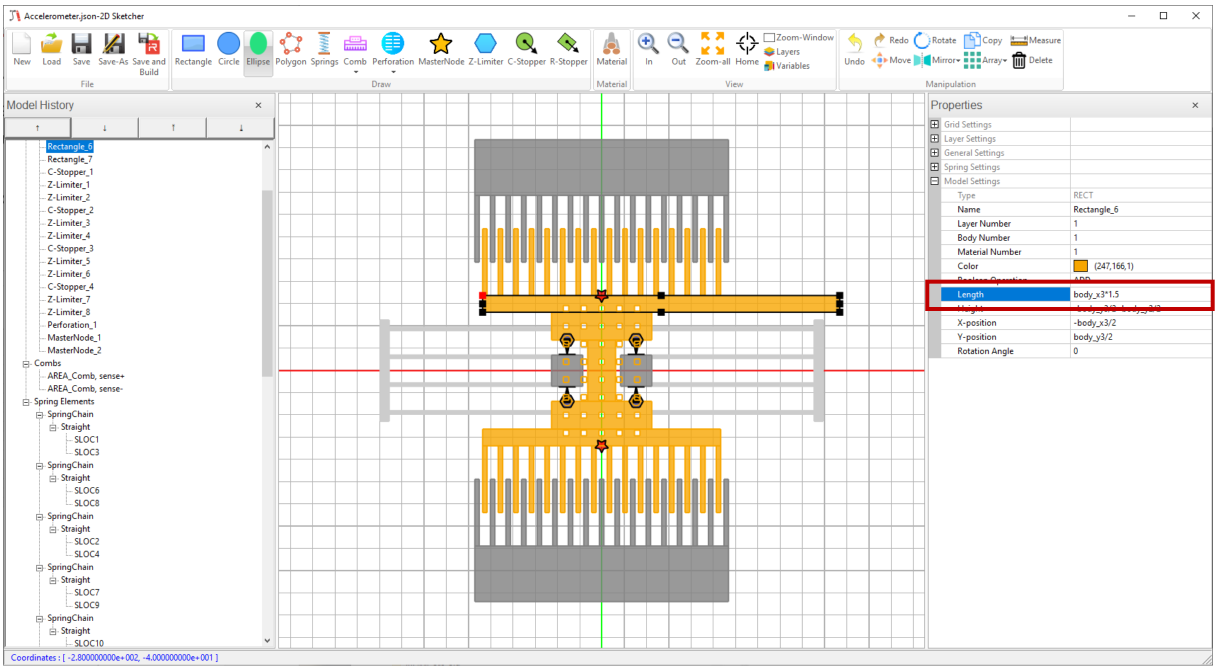


➢ Change design variables

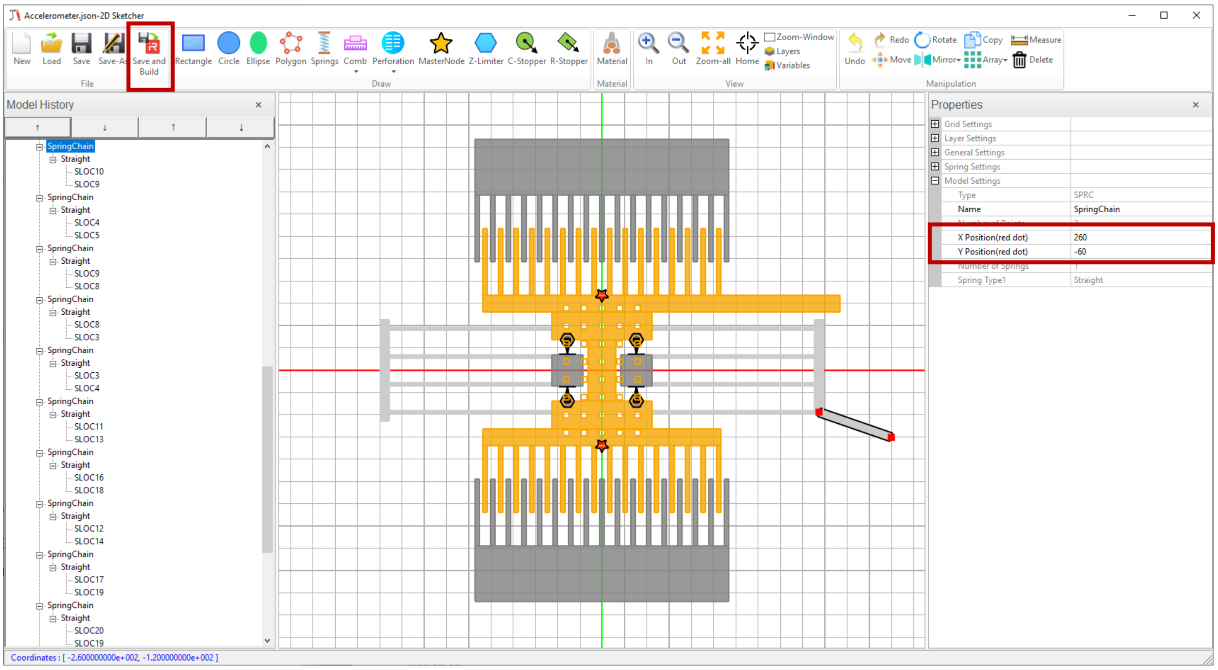
- Click on the “Variables” icon and change “comb_nf” to 16 and “spring_l” to 150. Press the “Apply” button.



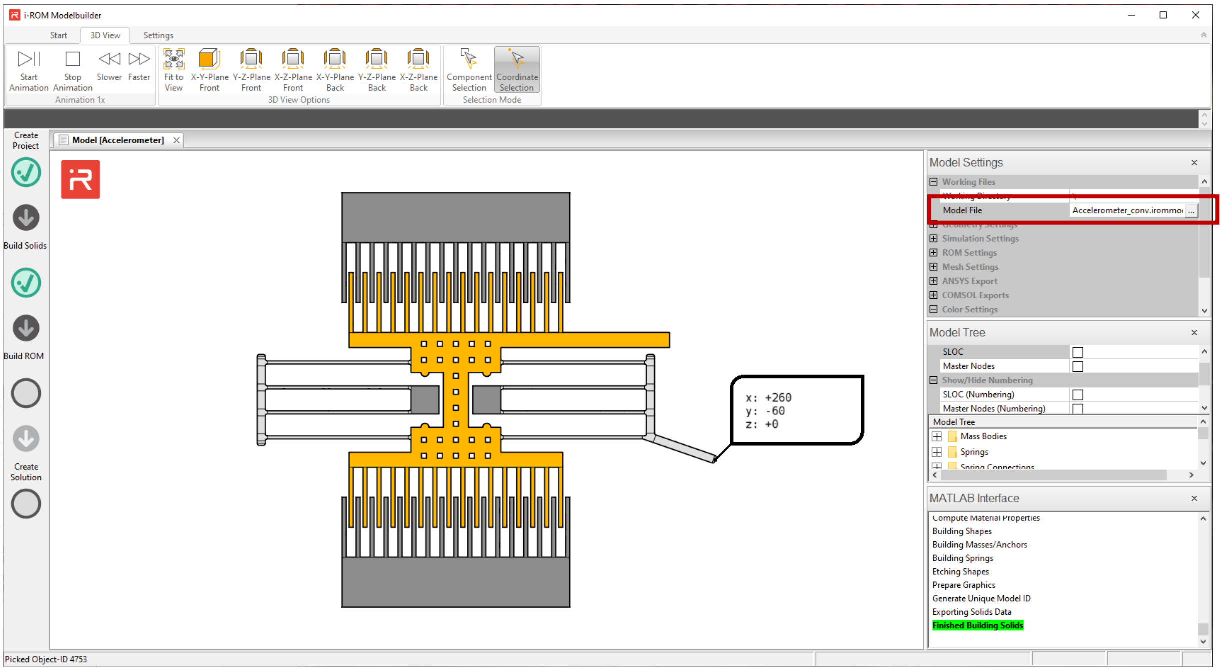
- Click on the upper rectangle of the mass body in the graphic window. The model item is highlighted in the model tree and settings appear in the “Properties” tab on the right. The x-y-position of the rectangle is marked by the red dot in the upper left corner and the size by the black handles.
- Current settings can either be changed in the “Properties” tab or by mouse manipulation. Change the length of the rectangle to “body_x3*1.5”.



- Click on the lower right spring and move the red spring location point with the mouse to another position. Click on "Save and Build" to transfer all changes to the MODELBUILDER main program.



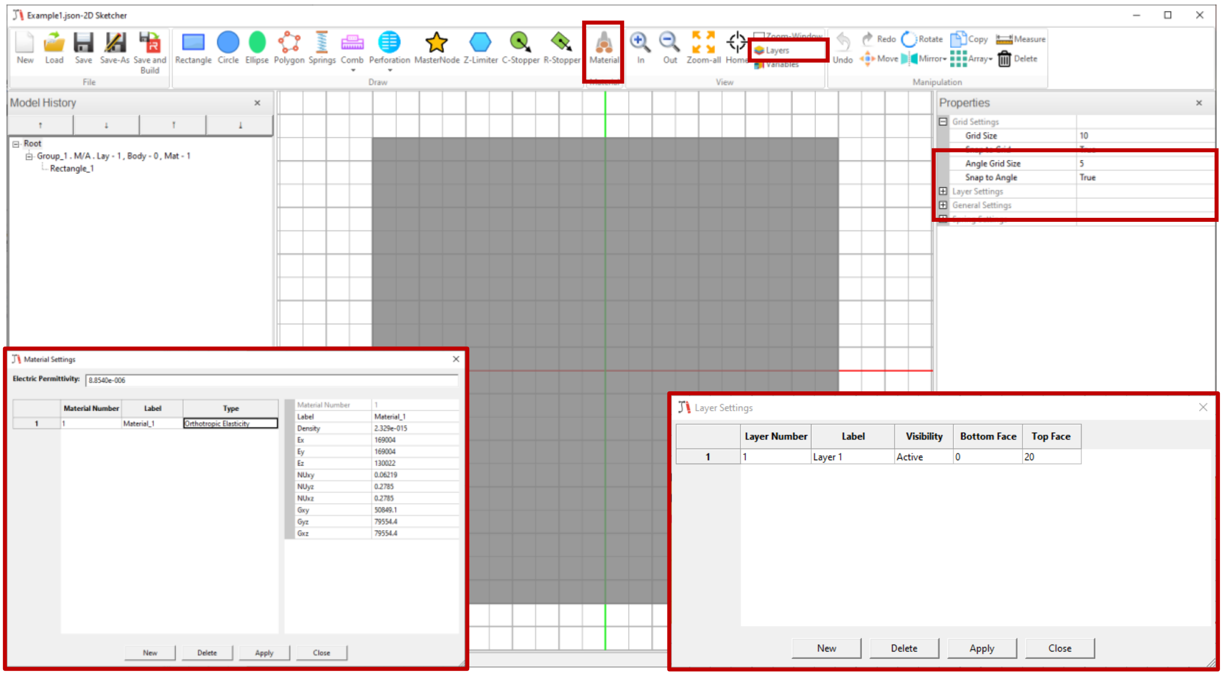
- The modified design appears on the screen. The "Model File" name is changed to "Accelerometer_conv.irommod" to avoid overwriting the original model data.
- Click on the "Coordinate Selection" icon in the "3D View" panel to check the data of the modified connecting point location.



8 Graphical Model generation (SKETCHER Interface)

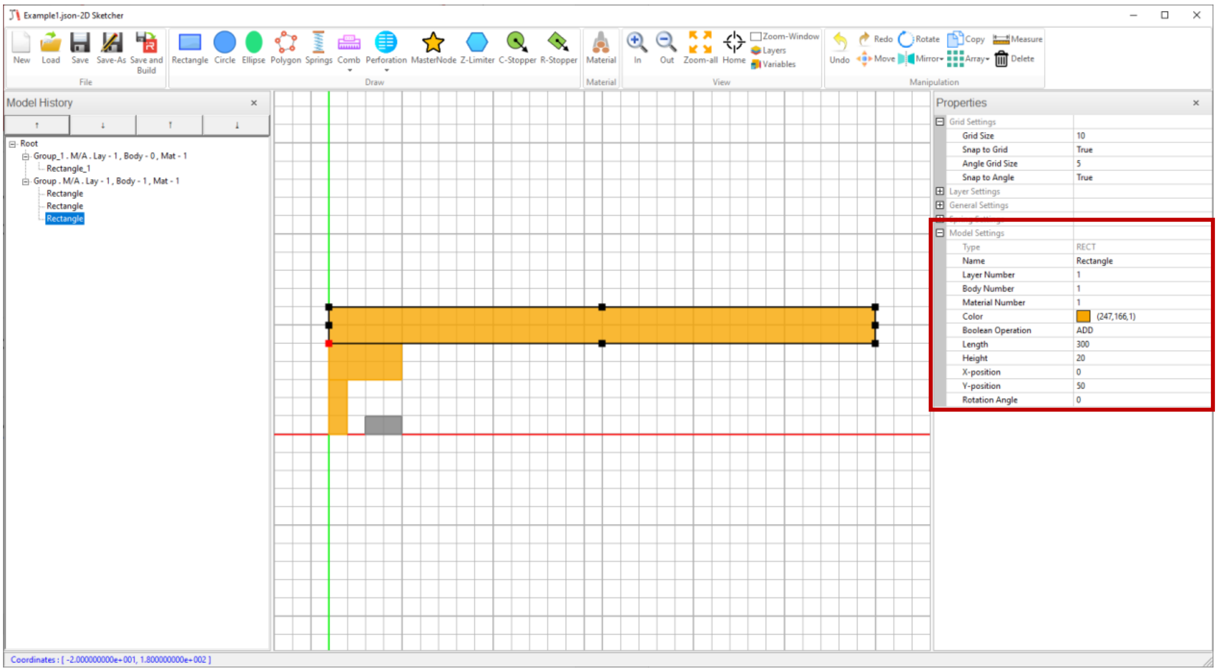
➢ Graphical model generation (new model)

- Create a new model in the Sketcher interface. Click on the “New Project” icon, create a “Test_case1” folder and select it as working directory.
- Click on “Build Solids” and then click “Sketcher”. An empty model with an anchor block appears on the screen. The default grid settings are shown in the “Properties” window.
- Click on the “Material” icon and select default properties for single crystalline silicon. Click on “Orthotropic Elasticity” and the “Apply” button.
- Click on the “Layers” icon to specify the bottom and top face z-coordinates of the functional layer.



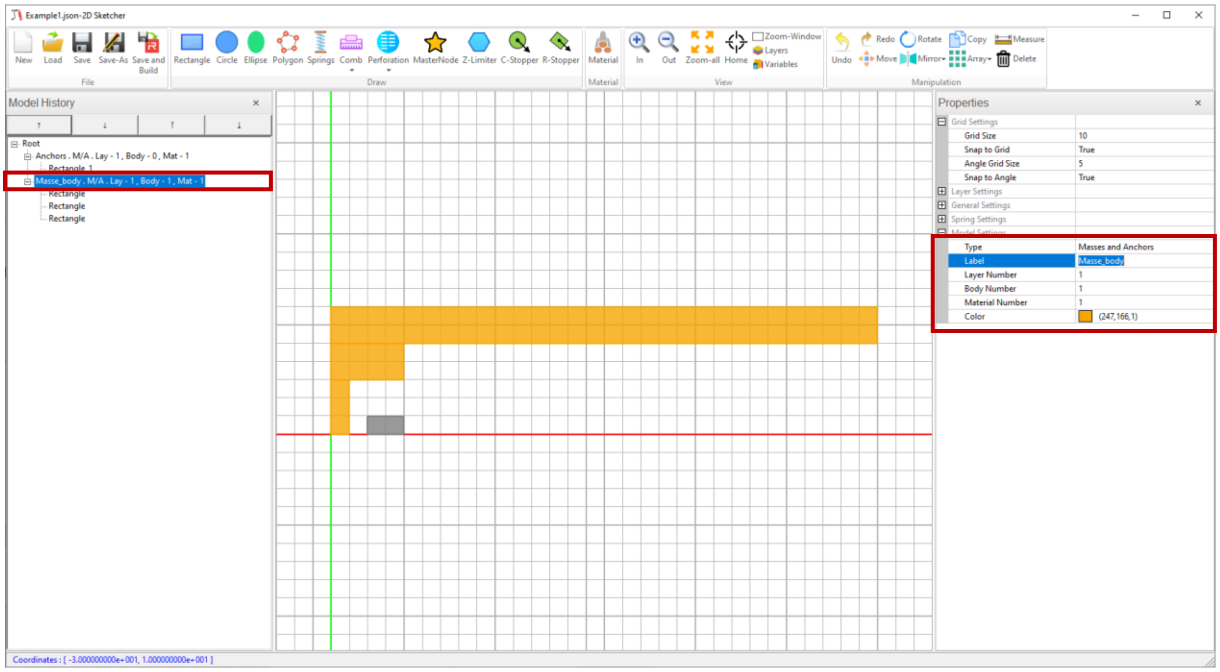
➢ Create a quarter model of the mass body and anchor block

- Click on the gray anchor and move the rectangle to the desired position. The current data can be seen from the grid. The body reference number for anchors must be zero.
- Next, create tree rectangles which belong to the mass body, set body reference number 1. Click on the “Rectangle” icon and draw the shape elements.
- Rectangles can be moved by mouse manipulations or by specifying the X- and Y-position in the “Properties” window.



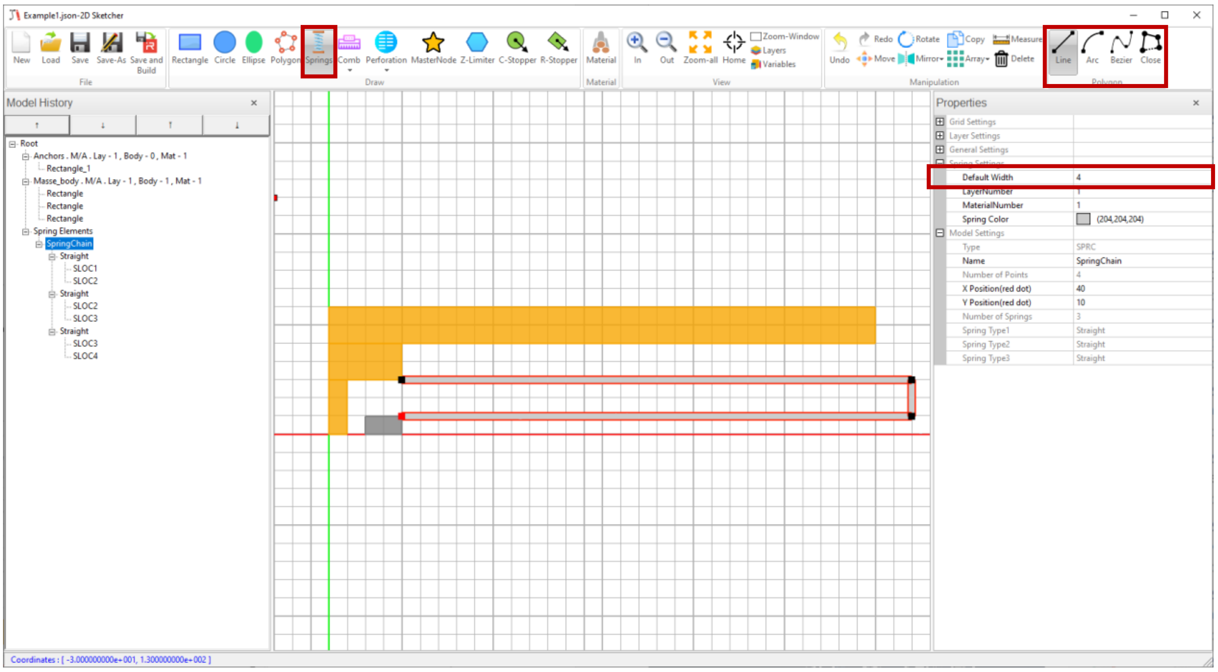
➢ Assign group properties

- All items with the same layer, body and material reference number belong to the same group. Group names and colors can be assigned in the “Properties” window. Click on the group in the “Model History” window to change group settings.

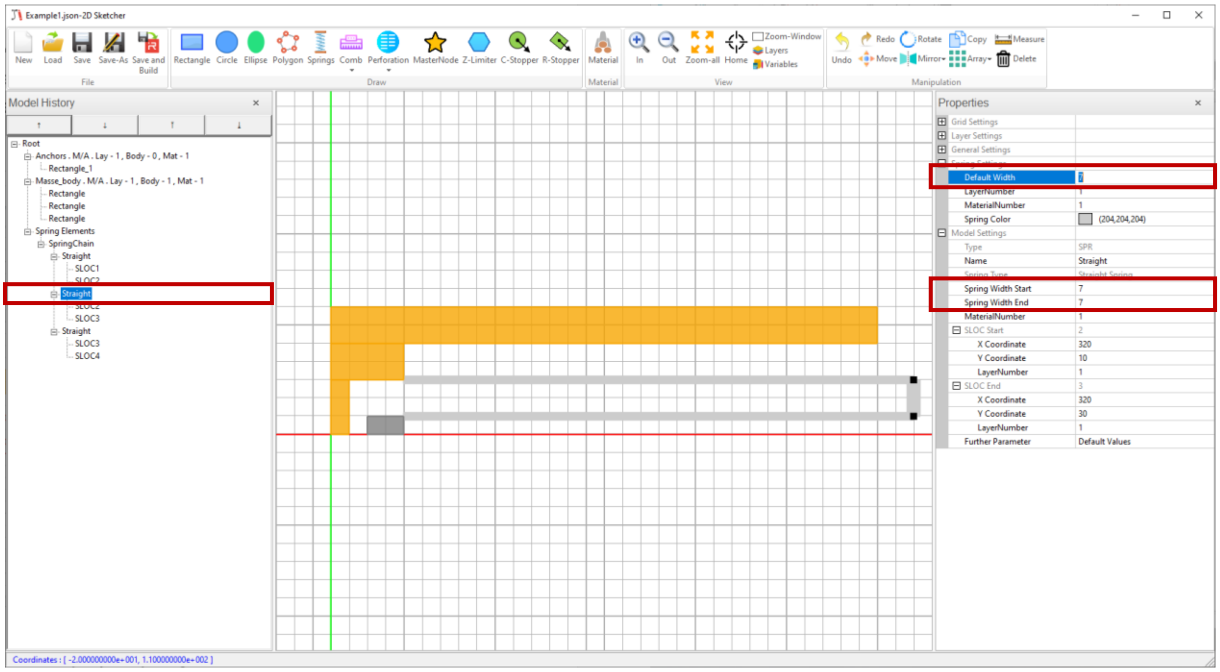


➢ Draw spring elements

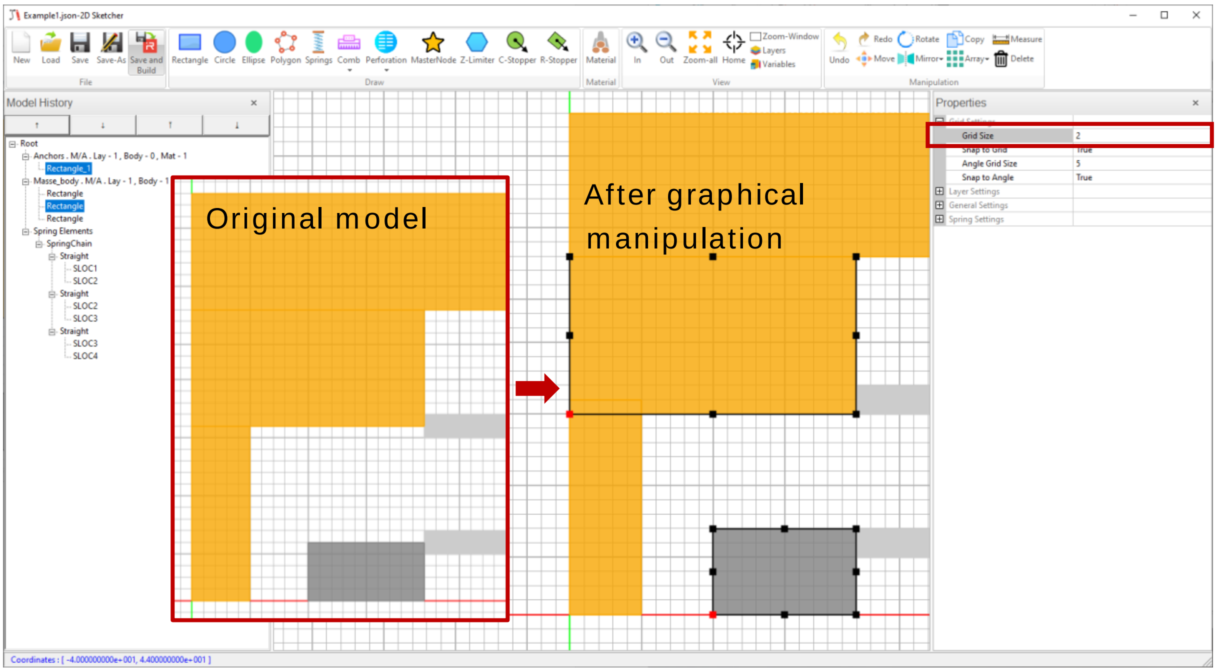
- Set a default spring width of 4 µm.
- Click on the “Springs” icon, select a straight “Line” and draw tree springs as shown below. Press the “Close” icon or ESC to leave the spring menu.



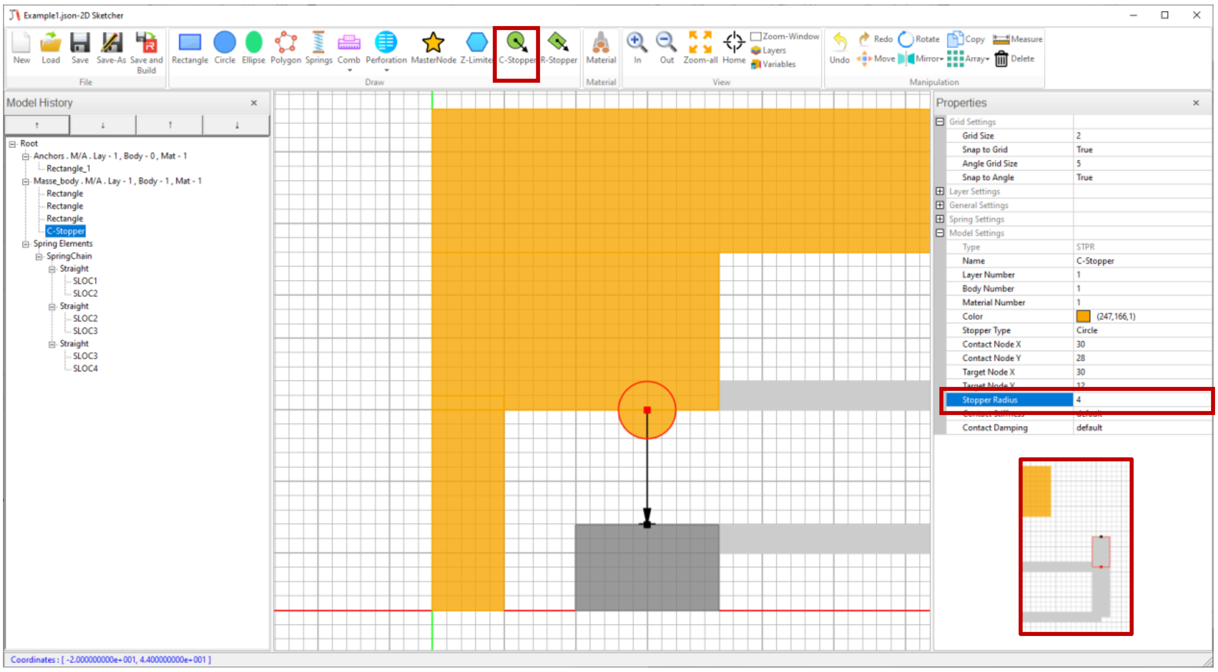
- Change the spring width of the vertical spring from 4 to 7 µm. Furthermore, change the default spring width to 7 µm for further spring elements.



➢ Adjust the size of the mass body and anchor to the spring width
- Change the grid size from 10 to 2 µm.
- Align the upper edge of the anchor and the lower edge of the second orange rectangle to outer edges of the spring.

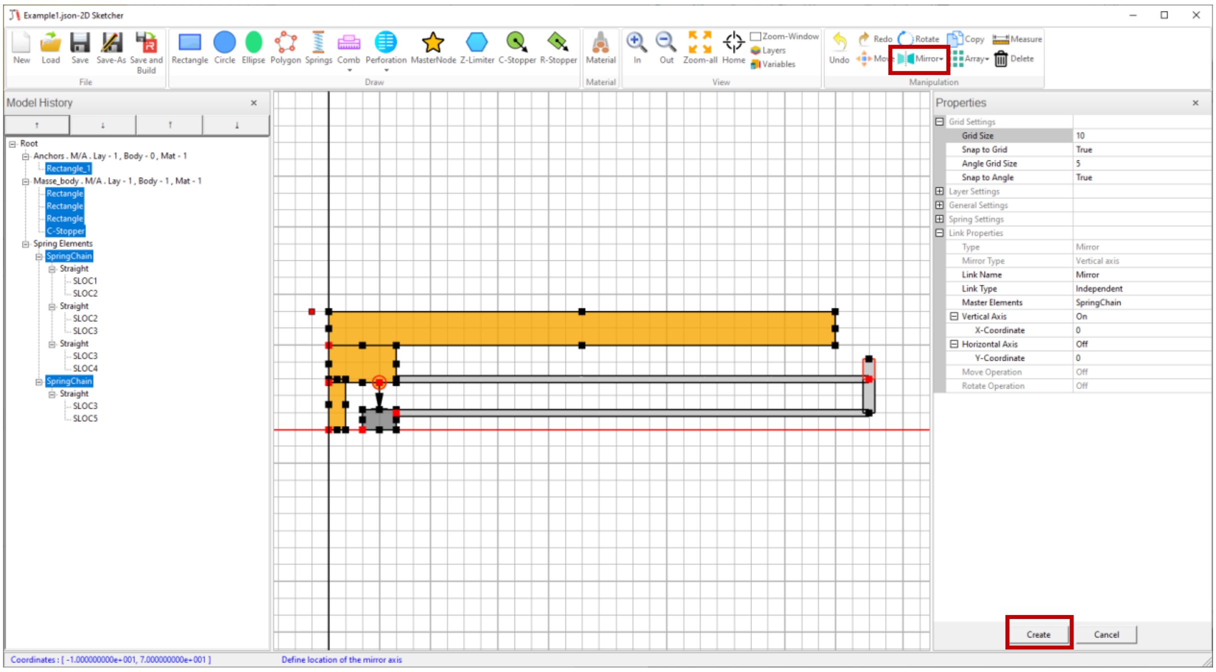


➢ Add a circular stopper element
- Click on the “C-Stopper” icon and draw a stopper from the orange mass body to the anchor block. Press ESC to leave the stopper menu.
- Change the stopper radius to 4 µm.
- Create another spring element at the upper right connecting point.



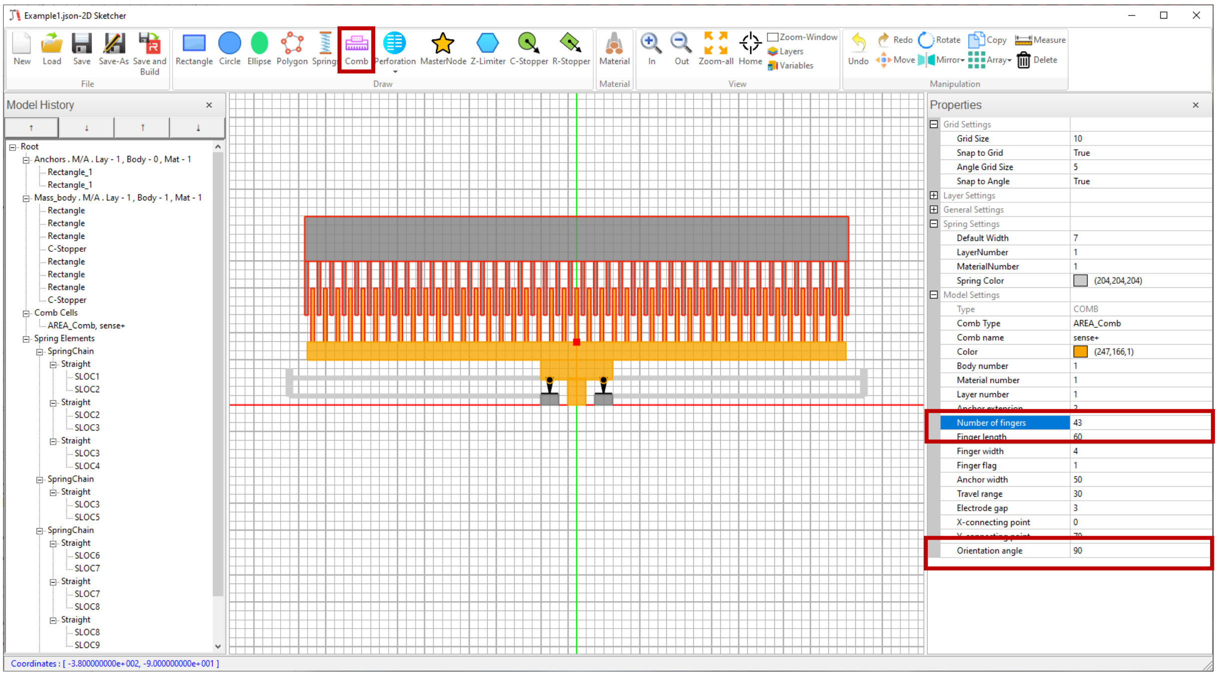
➢ Mirror all structural items around the y-axis

- Select all shape elements:
  - Draw a box with pressed left mouse button or
  - Select items with the Shift key and pressed left mouse button.
- Click on the “Mirror” icon, click “Vertical axis” and press “Create”.

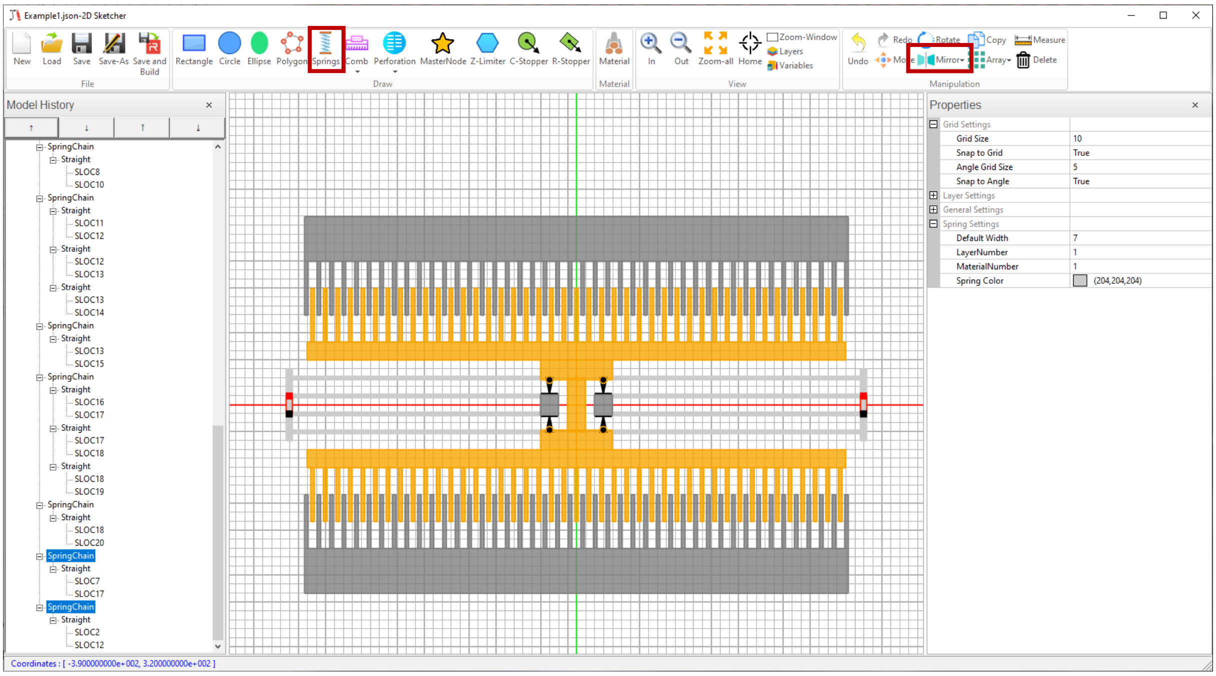


➢ Add the upper comb capacitor

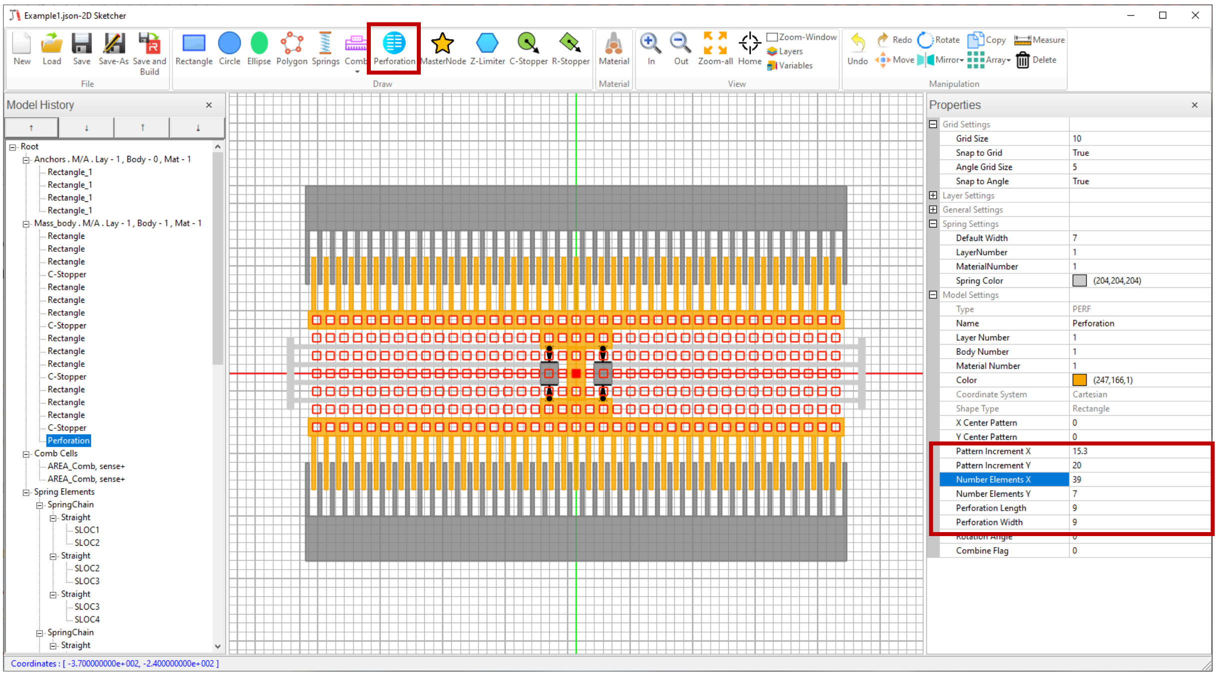
- Click on the “Comb” icon, select “AREA comb” and place the red connecting point on the upper edge of the mass body.
- Change the “Orientation angle” to 90 degrees and adjust the “Number of fingers” from 33 to 43.



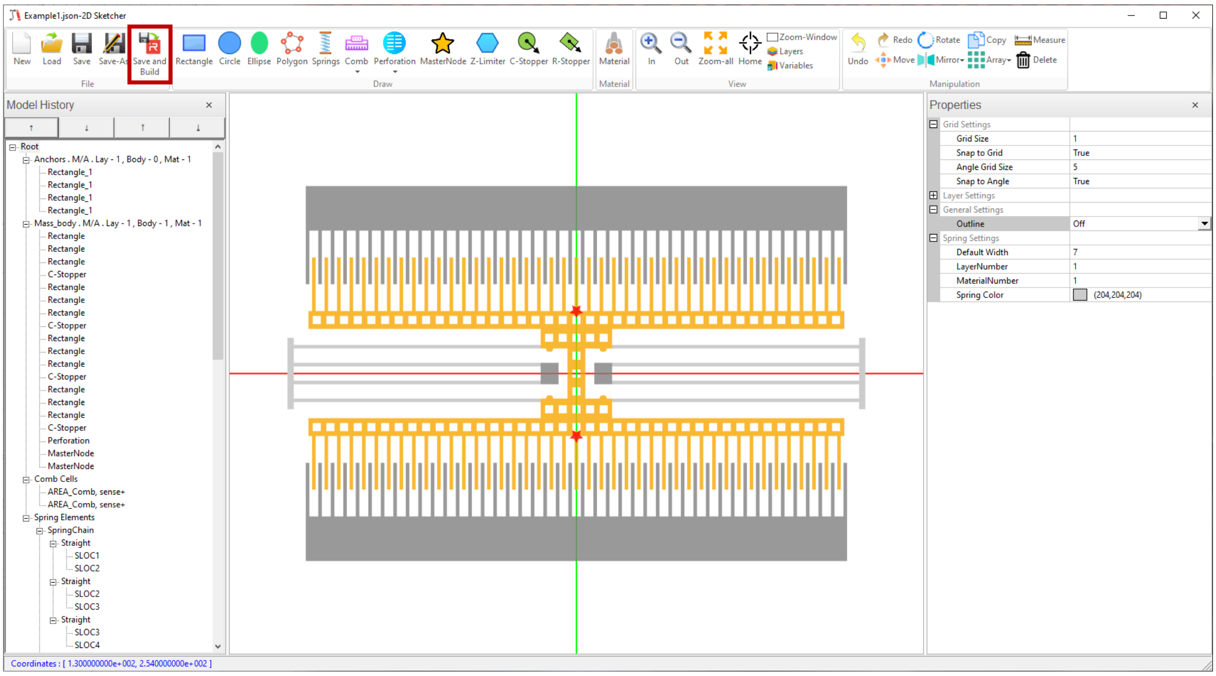
- ➢ Mirror all structural items around the x-axis
- ➢ Add two connecting springs between the upper and lower spring chains



- ➢ Add a perforation pattern in the center of the mass body
  - Click on the “Perforation” icon, select “Cartesian pattern of rectangles” and assign correct pattern parameters.

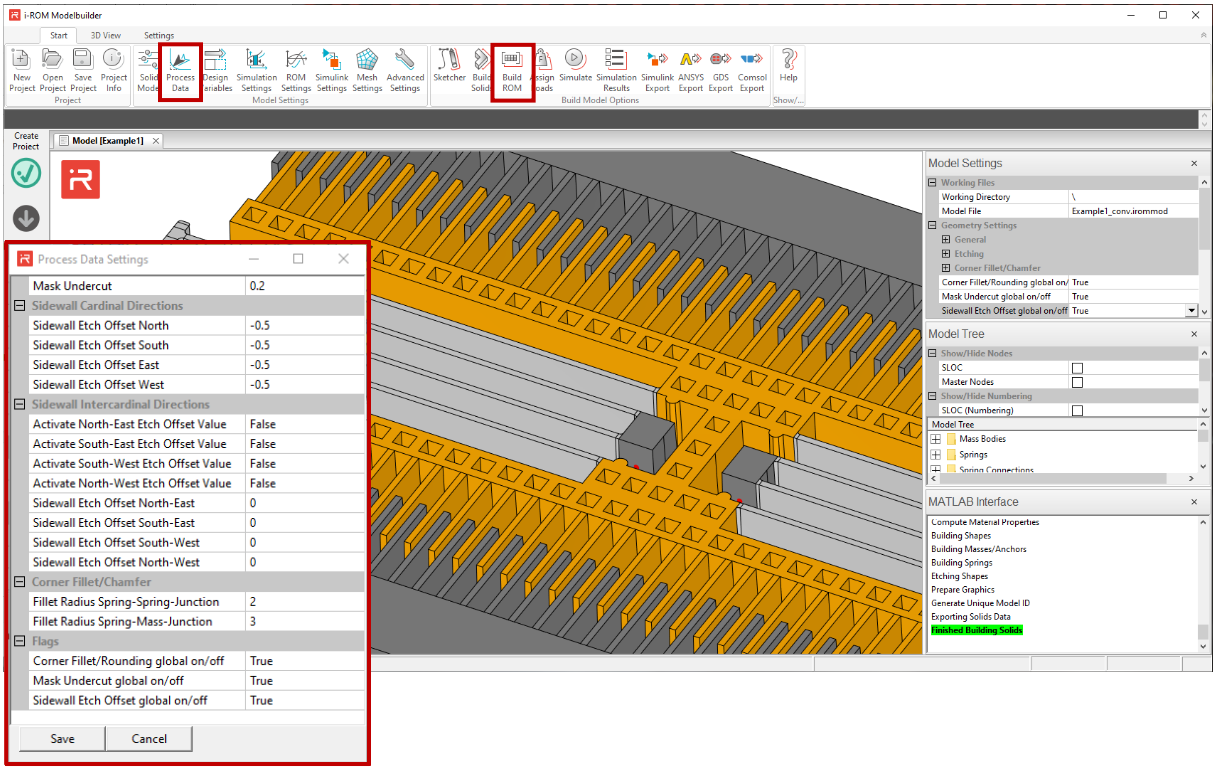


- ➢ Add master nodes on the mass body
  - Click on the “MasterNode” icon and place two master nodes at the upper and lower edge of the mass body.
  - You can deactivate outlines of shape elements in the general settings tap and assign a small grid size to erase grid lines on the screen.

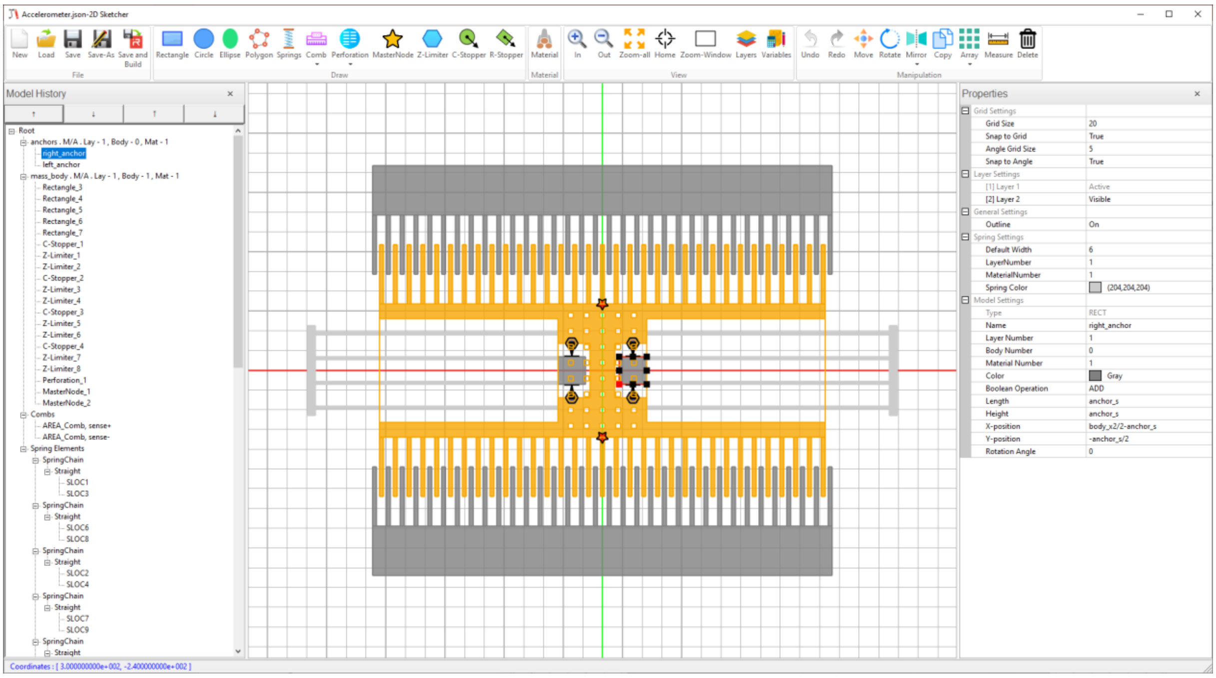


➢ Transfer the 2D-Sketcher model to the MODELBUILDER

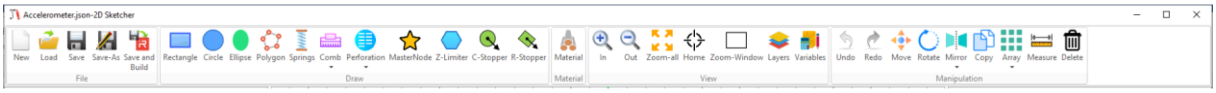
- Click on the “Save and Build” icon in the Sketcher window.
- The Sketcher window closes and the model opens automatically in the MODELBUILDER main program.
- Add process data settings and rebuild the model.



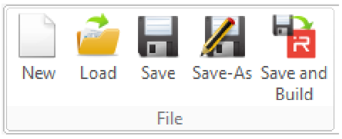
Part C - Graphical User Interface of the SKETCHER



Top bar GUI elements



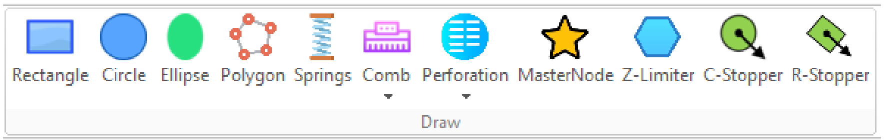
➢ File:



- New: Create a new SKETCHER model.
- Load: Load an existing SKETCHER model.
- Save: Save the current SKETCHER model.
- Save As: Save the current SETCHER model under a specific file name.
- Save and Build: Save the current model and export the file to the MODEL-BUILDER main program.

SKETCHER model files are saved in a JavaScript object notation (extension *.json) and MODELBUILDER files are based on the i-ROM Programming and Design Language (extension *.irommod). Translated MODELBUILDER files have the same prefix but are given the extension “*_conv.irommod” to avoid overwriting existing files.

- Draw:

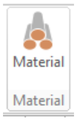


- Rectangle: Creates rectangular primitives for mass bodies, anchors, and plate capacitances.
- Circle: Creates circular primitives for mass bodies, anchors, and plate capacitances.
- Ellipse: Creates elliptical primitives for mass bodies, anchors, and plate capacitances.
- Polygon: Creates custom primitives for mass bodies, anchors, and plate capacitances. A pop-up menu appears with:
  - Line: Creates a straight outer line segment.
  - Arc: Creates a circular outer line segment.
  - Bezier: Creates an outer line segment of a Bezier curve.
  - Close: Closes the outer line segment from the current to the start point (you can also click on the start point of the polygon to close the custom primitive).
- Springs: Creates suspension springs with straight line, curved, and Bezier lines*. Click the Close icon to exit the spring chain menu.
- Comb: Creates a comb cell from the comb cell library. The current release supports 5 different comb cell library elements (see COMB command).
- Perforation: Creates a perforation pattern inside mass bodies or anchors. The current release supports 4 different types of perforations (see PERF command).
- Master Node: Creates master nodes on mass bodies (see MAST command).
- Z-Limiter: Creates z-limiter for out-of-plane motion components (see ZLIM command).
- C-Stopper and R-Stopper: Creates circular or rectangular stopper elements for in-plane motion components (see STOP command).

The active drawing functions can be used multiple times to create model components. The active icon is visualized by a dark gray background color in the top GUI window. Press ESC or click on the active icon to exit the current drawing mode.

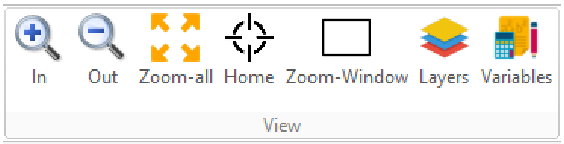
**The data exchange of the marked points with the MODELBUILDER is not supported in the current release.*

➢ Material:



Defines isotropic or orthotropic material properties for mass bodies or spring elements. Spring elements must have material reference number 1. Material properties must be defined prior using material reference numbers for model components.

➢ View:



- In: Zoom in on the model.
- Out: Zoom out of the model.
- Zoom all: Fit all model components in the graphics window.
- Home: Pan the origin of the coordinate systems to the center of the graphics window.
- Zoom window: Select the area to be displayed by dragging a mouse box.
- Layers: Assigns layer settings (new layers, modify layers, delete layers). In the current release, the first layer defines the functional layer and the other layers are used for the top and bottom plate capacitances. Plate capacitances are assembled from rectangular, circular, elliptical, and polygonal primitives.
- Variables: Assigns new design variables or changes existing design variables. Design variables are constant values or mathematical expressions. Design variable can be applied to dimensional parameters of model components. Other elements such as body, material or layer reference numbers, material properties or settings for manipulations (mirroring, copying, array operations) must not be defined by design variables.

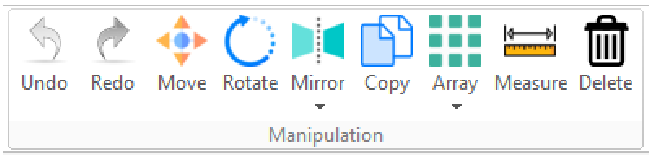
View settings can also be realized by mouse manipulations:

- Press Ctrl and roll the mouse wheel to zoom in or out.
- Press Ctrl and left mouse button to pan in x- and y-direction.

Select model items:

- Click on the model component on the screen or in the Model History window.
- Multiple model elements can be selected by:
  - Press Shift and select multiple items with the mouse.
  - Drag a mouse box to selected all model components inside the box.
  - Press Shift and select multiple model components in the model tree.
- Click in the open space or press ESC to unselect all items.

➢ Manipulation:



- Undo: Undo the previous edit. It can be applied multiple times.
- Redo: Restore a previously deleted edit.
- Move: Move selected objects in x- or y-direction.
  - Select one or more model components.
  - Click on the Move icon.
  - Click on the screen to defined the starting point of the motion vector.
  - Click on the screen to select the endpoint of the motion vector.

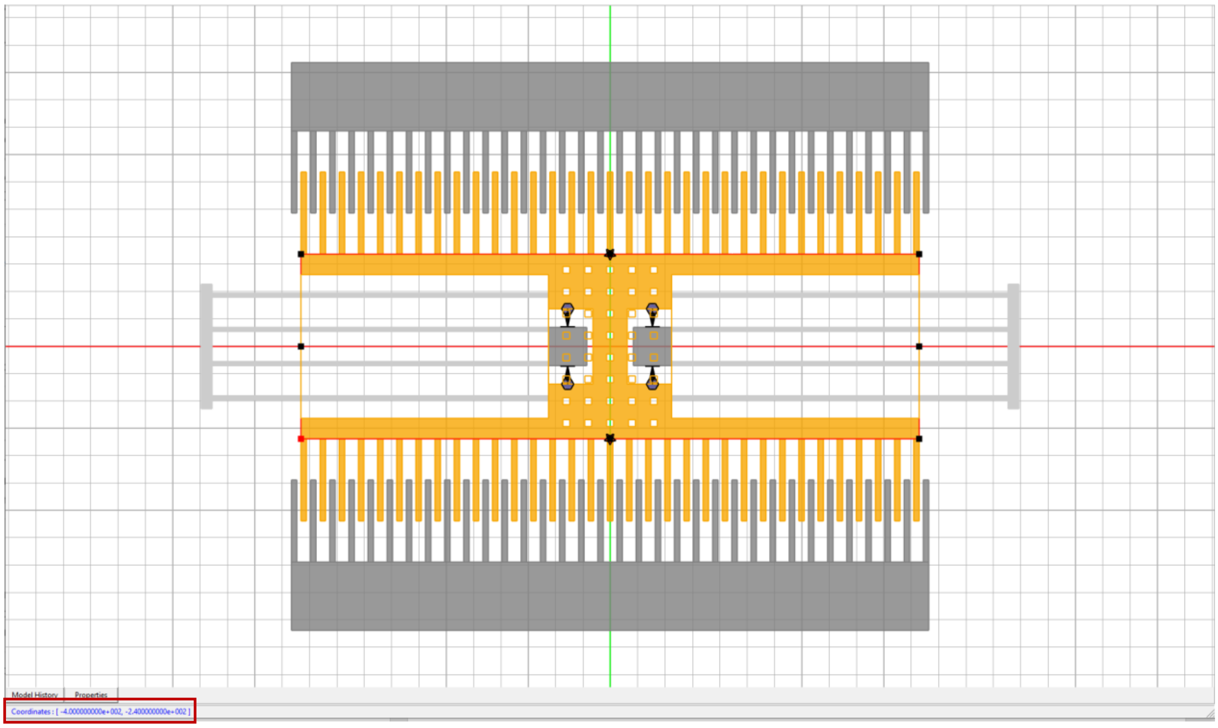
  You can also click directly on one or more model components and move them while holding down the left mouse button.
- Rotate: Rotate selected objects.
  - Select one or more model components.
  - Click on the Rotate icon.
  - Click on the screen to defined the center point for the rotation operation. A cylinder coordinate system appears on the screen.
  - Move the mouse to rotate model components. Press the left mouse button to apply the rotate operation.
- Mirror: Mirror selected objects around the horizontal, vertical or both axes.
  - Select one or more model components.
  - Click on the Mirror icon and select the mirror axis.

- Click on the screen to set the horizonal or vertical position of the mirror axis. The position of the mirror axis can be changed by numerical values in the Properties tab.
- Click Create to mirror all model components.

The original model components can be deleted if only a mirrored copy was intended.

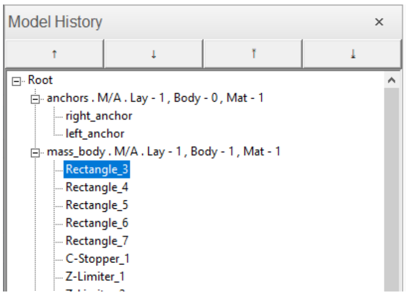
- Copy: Creates a copy of selected objects in x- and y-direction.
  - Select one or more model components.
  - Click the Copy icon.
  - Click on the screen to defined the starting point of the distance vector.
  - Click on the screen to select the endpoint of the distance vector. The distance vector can be modified by numeric values in the Properties tab.
  - Click Create to finish the copying operation.
- Array: Creates an array of selected objects in Cartesian or cylinder coordinates.
  - Select one or multiple model components.
  - Click the Array icon and select Cartesian or cylinder coordinated based array operations.
  - Enter or change array data in the Properties tab and click Create to complete the Array operation.
- Measure: Measures distances between model components. Distances can be measured between connecting points, handles, and grid values of all model components.
- Delete: Use the Delete icon or the Delete button of the keyboard to delete one or more selected items.

Main Graphics window



- The Main Graphics window is used to display all model components on the screen.
- Model components can be selected either in the Main Graphics window or in the Model History window. Selected model components are highlighted with a red connecting point and black handles. The red connecting point can be used to rotate model components and the black handles to change dimension parameters. Model components placed behind other elements can only be selected in the model tree.
- Properties and dimensional parameters of selected model components appear automatically in the Properties tab on the right side of the SKETCHER user interface.
- The current cursor position is displayed in the lower left corner.

Model History window

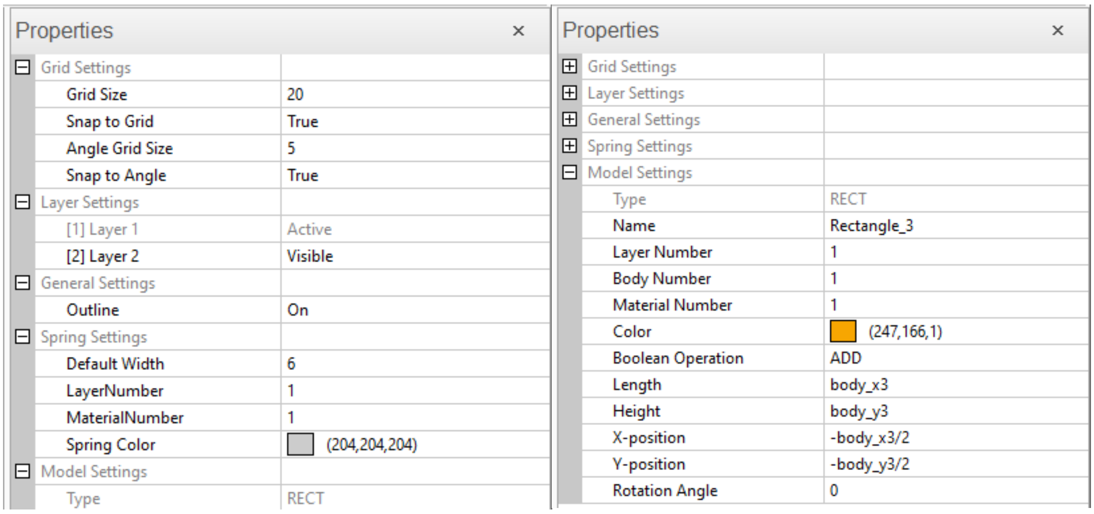


- The Model history window lists all model components in a model tree.

- The model tree can be used to select one or more (Shift-click) model components.
- The order of the model components of the same group can be changed with the upper arrow buttons:
  - First arrow button: Move the selected model components one step up.
  - Second arrow button: Move the selected model components one step down.
  - Third arrow button: Move the selected model components to the top of the group list.
  - Fourth arrow button: Move the selected model components to the end of the group list.

  Arrow buttons are required to change the order of model components for Boolean operations.

Properties window



➢ Grid Settings tab: Assign grid settings for model components.
- Grid size: Size to snap to x- and y-coordinate values in a Cartesian Coordinate system or the radial direction in a cylindrical coordinate system.
- Snap to grid: True → grid snapping is active.
  False → grid snap is deactivated.
- Angle grid size: Grid size in degrees for angular coordinates in a cylindrical coordinate system.
- Snap to angle: True → Snap angle is active.
  False → Snap angle is disabled.

Grid parameters are transferred from the MODELBUILDER to the SKETCHER interface using the SKET command. For new SKETCHER models, the default grid parameters are taken from the .\settings\init-file.json file.

The grid display in the main graphic window is disabled when there are too many grid lines in the current view. This happens when visualizing model components that are much larger than the grid size (>100x). On the other hand, you can assign a very small grid size to disable the grid lines of the main graphics window.

- Layer Settings tab: Changes the visibility of layers.
  - Active: Only one layer can be set active. The active layer is displayed in full color.
  - Visible: Visible layers appear semi-transparent.
  - Hidden: Hidden layers are not visible in the Main Graphics window.

  Layer data settings can be changed either via the Layers icon or on the Layer settings tab. New layers must be defined in the top GUI window. The first layer is the functional layer, all other layers are used for top and bottom plate capacitors.

- General Settings tab: Turns model component outlines on or off.
  - On: Model component outlines are drawn on the screen.
  - Off: Model outlines are disabled.

  The Off state is typically used when you want to delete perforation pattern outlines and mass body outlines based on Boolean subtraction.

- Spring Settings tab: Assigns default spring settings.
  - Default width: Default spring width. The value is adopted for all subsequently defined spring elements.
  - Layer Number: All springs must be in the first layer (functional layer).
  - Material Number: Material reference number. Material 1 is used for all springs in the current release.
  - Spring Color: Assigns a color to all spring elements.

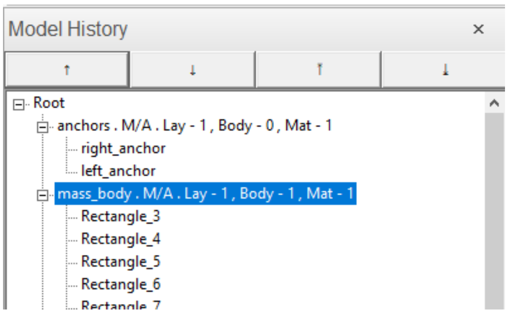
- Model Settings tab: Assigns and changes settings for model components.

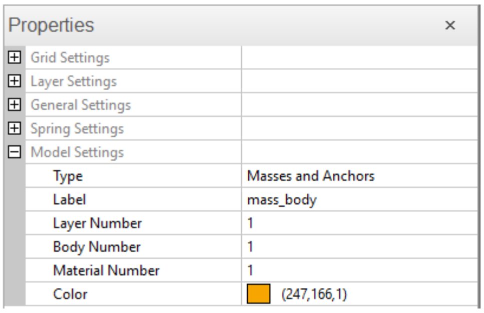
  For example, the following points apply to rectangular primitives:
  - Type: Displays a model component of type RECT.

- Name: The default name is Rectangle. The name appears in the model tree and can be changed on the Model Settings tab.
- Layer Number: Integer, 1 → functional layer, >1 → layers for plate capacitances
- Body Number: Integer, 0 → Anchors, >0 Rigid body reference number
- Material Number: Integer, >0. Before the reference number can be set, the material data must be assigned.
- Color: Opens a Colors pop-up window to assign default and custom colors.
- Boolean Operation: ADD → Adds the primitive to group elements, SUB → Subtracts a primitive from group elements.
- Length: Length of the rectangular primitive in x-direction. Can be negative.
- Height: Height of the rectangular primitive in y-direction. Can be negative.
- X-position: X-position of the connecting point.
- Y-position: Y-position of the connecting point.
- Rotation angle: Rotation angle of the rectangle at the connecting point.

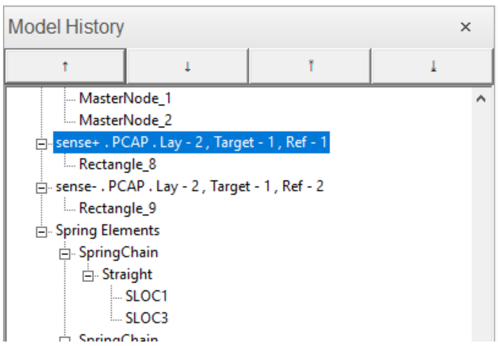
Boolean operations apply to the model components of the same group. Group elements have the same layer, body and material reference number. The group name can be changed in the Properties window when the group is selected in the Model History window.

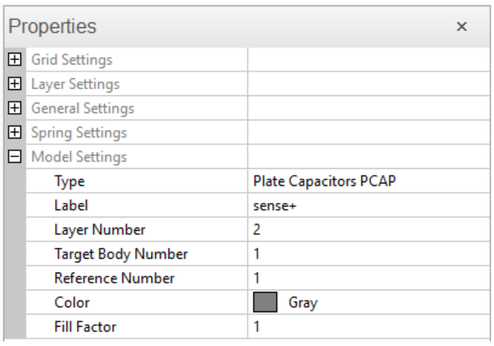
Groups can be of either “Masses and Anchors (M/A)” type





or “Plate Capacitors (PCAP)” type.





PCAP group elements are:

- Label: Capacitance label (see PCAP command). The capacitance label is used in the MODELBUILDER to link capacitances with voltage ports (see COND command). Comb or plate capacitances with the same label are merged into a single capacitance in MODELBUILDER simulations.
- Layer Number: The layer number must be >1 for plate capacitances. Layers settings defined the electrode gap between moving and fixed capacitances at the initial position.
- Target body number: Defines the reference number of the moving mass body. For capacitance calculations, the overlapping area of all group elements and the target body is evaluated in the MODELBUILDER main program.
- Reference number: The reference number is necessary to assign different capacitances that should refer to the same layer and the same target body. Each capacitance can be assigned to a different capacitance label allowing for different voltage ports in MODELBUILDER simulations.
- Fill factor: Capacitance values can be scaled with a constant fill factor. The fill factor can be increased (>1) or reduced (<1) in order to take into account the influence of fringing fields at outer edges or within perforation holes.

Part D - Example Manual

1 Acceleration sensor example

The following example demonstrates how to model and simulate a MEMS acceleration sensor in the i-ROM MODELBUILDER environment. All model items are defined in a User model input file. In the present case, the model input file is called “Accelerometer.irommod” and can be found in the “Accelerometer” project folder.

The single-axis accelerometer of Fig. 2 consists of a seismic mass with perforations, four stopper elements, two anchors, connecting springs and two comb-cells which form a differential capacitor. The upper comb is referred “sense+” and the lower one **“sense–”**.

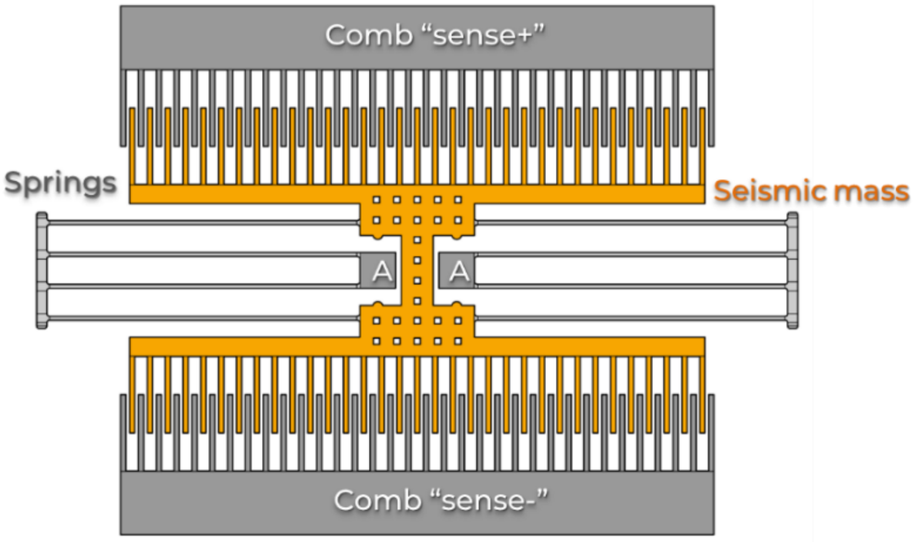


Fig. 2: Schematic view of the single-axis accelerometer and its components.

For MEMS, it is recommended to set up models in micrometer units to avoid numerical problems with ill-conditioned matrices. A so-called **µ**MKSV-system is widely used with micrometer, kilogram, second, volt and picoampere as basis units. Related units become micronewton, megapascal, picofarad and picocoulomb. Exemplarily, the free-space permittivity changes from 8.854e-12 F/m to 8.854e-6 pF/µm and gravitation of earth from 9.81 m/s^2 to 9.81e6 µm/s^2. The given µMKSV accelerometer model is also defined in SI-units for comparison (“Accelerometer_SI” project folder).

Several dimensions of the given example are highly enlarged for better visualization on the computer screen and in the figures of this documentation.

Usually the spring width and electrode gaps are about two micrometers and etch tolerances are a few nanometers. Enlarged gaps are helpful to plot and to animate the functional behavior with larger amplitudes for demonstration.

One consequence is that the drive voltages, eigenfrequencies and all loads (forces, acceleration, angular rates) assume values that are much higher than reality.

In the following sections, the modeling and simulation process of the accelerometer will be discussed in detail. It starts with the model generation process (pre-process), then follow commands and features to perform simulation runs (solution-process). Finally, typical steps for results evaluation and graphical visualization will be shown on several examples (post-process).

Creating seismic masses and anchors

In the MODELBUILDER, seismic masses can be defined quickly by parametric 2D-primitives and Boolean operations. The mass body of Fig. 2 can either be defined by rectangles which are combined by Boolean “add” (a) or “subtract” (b) operations or simply by a polygon (c) as shown in Fig. 3. The resulting entity is a single mass body containing no internal lines, except different material properties have been defined for involved primitives. The thickness and vertical location of the functional layer with its mass, springs and anchors is defined by a LAYR command.

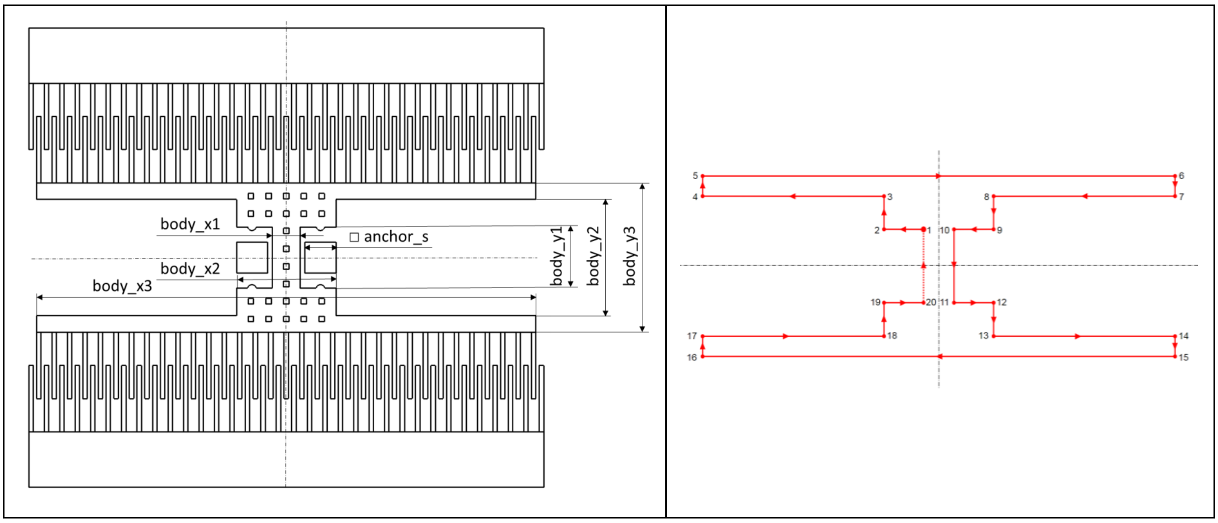


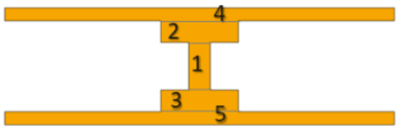
Fig. 3: Defining the seismic mass of the accelerometer by a single polygon.

Appropriate MODELBUILDER commands for all three cases are listed in Fig. 4. The geometrical dimensions are defined by parameters (PARA command) which can be modified in the Design Variables window of the Graphical User Interface.

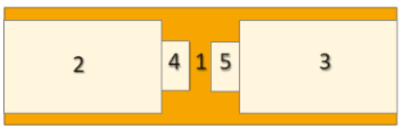
Commands to define masses and anchors are RECT, CIRC, TRIA and POLY. Each seismic mass is related to a body reference number of type integer. In the given example, just one mass is necessary (body reference number is 1). Primitives of the same mass body may have different material reference numbers which are linked to material properties defined by the MATP command.

A variety of material properties are often used to assign efficient density values to different regions of seismic masses if resource consuming perforations are neglected. Otherwise, a reasonable number of perforations can be added by the PERF command.

```
% Define a seismic mass by Boolean add operations (a)
RECT,,add,1,1,,-body_x1/2,-body_y1/2,body_x1,body_y1;
RECT,,add,1,1,,-body_x2/2,+body_y2/2,body_x2,-body_y2/2+body_y1/2;
RECT,,add,1,1,,-body_x2/2,-body_y2/2,body_x2,+body_y2/2-body_y1/2;
RECT,,add,1,1,,-body_x3/2,+body_y3/2,body_x3,-body_y3/2+body_y2/2;
RECT,,add,1,1,,-body_x3/2,-body_y3/2,body_x3,+body_y3/2-body_y2/2;
```




```
% Define a seismic mass by subtract operations (b)
RECT,,add,1,1,,-body_x3/2,-body_y3/2,+body_x3,+body_y3;
RECT,,sub,1,1,,-body_x2/2,-body_y2/2,-body_x3/2+body_x2/2,+body_y2;
RECT,,sub,1,1,,+body_x2/2,-body_y2/2,+body_x3/2-body_x2/2,+body_y2;
RECT,,sub,1,1,,-body_x1/2,-body_y1/2,-body_x2/2+body_x1/2,+body_y1;
RECT,,sub,1,1,,+body_x1/2,-body_y1/2,+body_x2/2-body_x1/2,+body_y1;
```




```
% Defined by a polygon (c)
POLY,,add,1,1,,...
-body_x1/2,+body_y1/2,...
-body_x2/2,+body_y1/2,...
-body_x2/2,+body_y2/2,...
-body_x3/2,+body_y2/2,...
-body_x3/2,+body_y3/2,...
+body_x3/2,+body_y3/2,...
+body_x3/2,+body_y2/2,...
+body_x2/2,+body_y2/2,...
+body_x2/2,+body_y1/2,...
+body_x1/2,+body_y1/2,...
+body_x1/2,-body_y1/2,...
+body_x2/2,-body_y1/2,...
+body_x2/2,-body_y2/2,...
+body_x3/2,-body_y2/2,...
+body_x3/2,-body_y3/2,...
-body_x3/2,-body_y3/2,...
-body_x3/2,-body_y2/2,...
-body_x2/2,-body_y2/2,...
-body_x2/2,-body_y1/2,...
-body_x1/2,-body_y1/2;
```

Fig. 4: Creating a seismic mass by primitives and Boolean operations.

The body reference number must be set to zero for all anchors. The anchor extension to the substrate surface is defined by the “Anchor Extrude Length” parameter in the Solid Model Settings window of the GUI.

Graphical manipulation of the model

The “3D View” panel shown in Fig. 5 allows to realign the model into the six orthogonal views, to select model items, to attach coordinate selection markers and to start and stop animation sequences after simulation runs. The slower and faster icons are used to change the speed of animations.

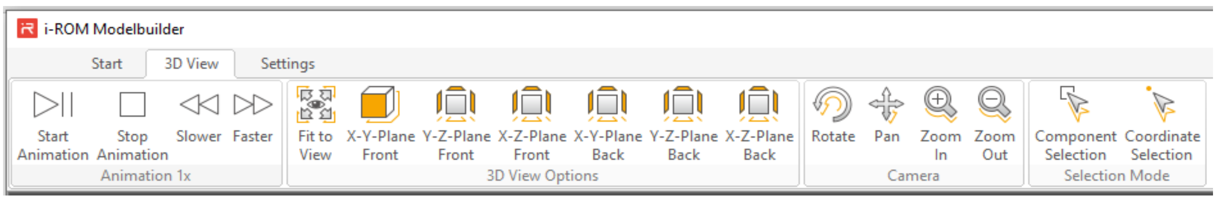


Fig. 5: 3D View panel to verify model items in the pre- and post-process.

Alternatively, the model orientation can quickly be changed by mouse buttons:

- Press “Ctrl + left mouse button” to move the structure in x- or y-directions,
- Press “Ctrl + scroll the mouse wheel” to zoom in or out, and
- Press “Ctrl + right mouse button” to tilt the structure around x- and y-axes.

In the latter case, the User can either tilt the model around its Center Point (CP-mode) or around a User-defined Pivot Point (PP-mode) located on the model.

To activate the “CP-mode”, move the mouse pointer into the open space. Press “Ctrl + right mouse button” and tilt the model around x- or y-axes.

For the “PP-mode”, move the mouse pointer to the pivot point of interest. Note, the pivot point must be on the model and may not be in the open space. Press “Ctrl + right mouse button” and tilt the model around x- or y-axes.

A rectangular section of the model can be selected by the “Box zoom” mode. Press the right mouse button and move the mouse from the upper left to the lower right position in the model window.

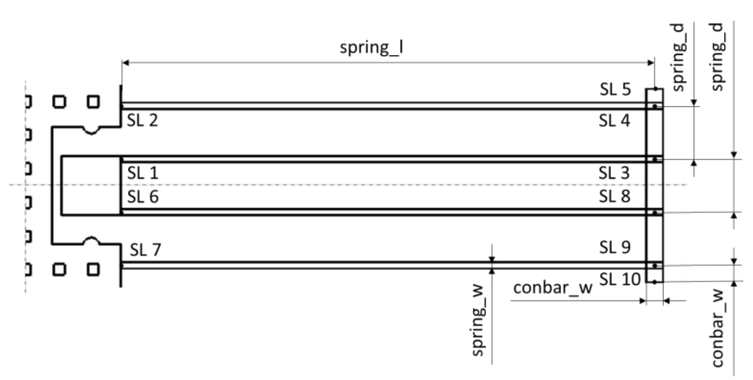
The “Fit to View” button shows the model at full size on the display screen.

Creating suspension springs

Timoshenko beam elements are implemented for modeling of suspension springs. Beams either connect seismic masses to anchors or seismic masses to other masses. A straight spring is described by two and a circular spring by three connecting points (SLOC command) which are linked by a line element (SPRI command) to form a spring. Connecting points are either located at the interface to masses (the connecting points #2 and #7), at the interface to anchors (the connecting points #1 and #6) or they are in the open space. The latter case happens if the connecting point links different springs together (points #3, #4, #8 and #9) or defines the open end of a beam (points #5 and #10 in Fig. 6). The outer corner fillets at connecting points #1 and #6 are automatically disabled because those springs are aligned at the upper and lower edge of the anchor.

Looping and condition commands as *FOR or *IF allow for simple and efficient spring designs in larger models. The spring width can be assigned to the begin and to the end of each line and allows for uniform and tapered beams. Beams can also be subdivided into several lines to model a sudden or a smooth transition of the spring width. The thickness is taken from the LAYR command.

```
% Spring connecting points
PARA,offs=0;
*FOR,x_sym,1,-1,-2
 *FOR,y_sym,1,-1,-2
  SLOC,1+offs,,,new,x_sym*body_x2/2          ,y_sym*0.5*spring_d;           % SL at anchor
  SLOC,2+offs,,,new,x_sym*body_x2/2          ,y_sym*1.5*spring_d;           % SL at mass #1
  SLOC,3+offs,,,new,x_sym*(body_x2/2+spring_l),y_sym*0.5*spring_d;           % SL at connecting bar
  SLOC,4+offs,,,new,x_sym*(body_x2/2+spring_l),y_sym*1.5*spring_d;           % SL at connecting bar
  SLOC,5+offs,,,new,x_sym*(body_x2/2+spring_l),y_sym*(1.5*spring_d+conbar_w); % End of bar extensions
  PARA,offs=offs+5;
 *ENDFOR
*ENDFOR
```




```
% Define springs between SLOCs
*FOR,x_sym,0,1
 PARA,offs=10*x_sym;
 SPRI, 1+offs,3+offs,,,spring_w;
 SPRI, 6+offs,8+offs,,,spring_w;
 SPRI, 2+offs,4+offs,,,spring_w;
 SPRI, 7+offs,9+offs,,,spring_w;
 SPRI,10+offs,9+offs,,,conbar_w;
 SPRI, 9+offs,8+offs,,,conbar_w;
 SPRI, 8+offs,3+offs,,,conbar_w;
 SPRI, 3+offs,4+offs,,,conbar_w;
 SPRI, 4+offs,5+offs,,,conbar_w;
*ENDFOR
```

Fig. 6: Defining springs by connecting points and linking spring elements.

Creating comb-cells and plate-like capacitors

Comb-cells and plate-like capacitors are necessary for capacitive sensing and electrostatic actuation of MEMS. The COMB command supports different types of comb capacitors and links the moving part of the comb to an outer face of the seismic mass. Each comb capacitor is associated to a custom name that is later used to map the comb capacitances to the voltage ports (COND command).

Only one command is necessary for each comb. Combs are parametric library elements with a specific layout, an arbitrary number of fingers and a series of dimensional parameters. Both, the x-y-location and the orientation angle of the comb are defined regarding a connecting point which is placed at the center of the interface to the seismic mass (see COMB command). Likewise, the thickness is taken from the LAYR command. Fixed conductors are automatically anchored with the specified anchor distance (see Solid Model Settings window) to the substrate surface.

Exemplarily, three different types of capacitors have been defined in Fig. 7. A desired type can be activated by the “build_capa” parameter in the Design Variables window of the Graphical User Interface as shown later.

Plate-like capacitors are defined by PCAP commands. Plate-like capacitors are assembled from primitives and Boolean operations in the same way as seismic masses. The overlapping area is automatically recognized for capacitance and force calculations. Details are discussed in the Micro mirror actuator section.

```
% Define capacitors, comb cells with area variation
*IF,build_capa,eq,0
 COMB,sense+,area,1,1,, 90,comb_nf,1,comb_fw,comb_fl,comb_bw,comb_tr,comb_eg,,,0,+body_y3/2;
 COMB,sense-,area,1,1,,270,comb_nf,1,comb_fw,comb_fl,comb_bw,comb_tr,comb_eg,,,0,-body_y3/2;
*ENDIF

% Define capacitors, comb cells with gap variation
*IF,build_capa,eq,1
COMB,sense+,dif2,1,1,,180,1,-1,pcomb_w,pcomb_l,pcomb_bw,pcomb_tr,pcomb_eg,,30,0,+body_y3/2-pcomb_w/2
COMB,sense-,dif2,1,1,,  0,1,-1,pcomb_w,pcomb_l,pcomb_bw,pcomb_tr,pcomb_eg,,30,0,-body_y3/2+pcomb_w/2
*ENDIF

% Define capacitors, bottom plate capacitors
*IF,build_capa,eq,2
    PCAP,,sense+,rect,add,1,1,2,-plate_x/2,+body_y3/2-plate_y/2,plate_x,+plate_y;
    PCAP,,sense-,rect,add,1,2,2,-plate_x/2,-body_y3/2+plate_y/2,plate_x,-plate_y;
*ENDIF
```

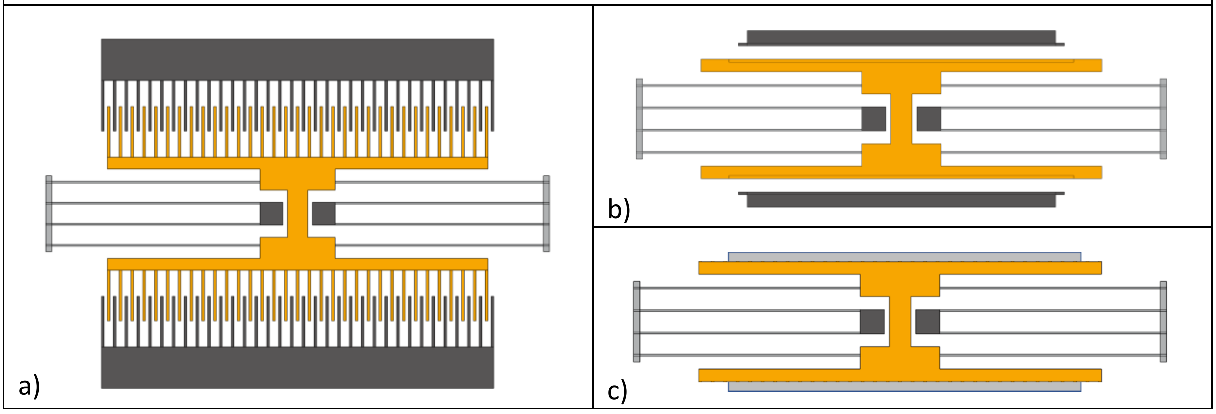


Fig. 7: Defining comb- and plate-like capacitors for the accelerometer example.

Creating perforations and stopper elements

The PERF command is a powerful routine to create perforation patterns in seismic masses. It supports circular and rectangular perforations arranged either in Cartesian or cylinder coordinates. Perforations which are partially or completely outside of mass bodies are automatically ignored. Alternatively, Boolean operations for intersections among different perforations or intersections with outer lines may be activated by the “c_flag” of the PERF command. Results for both values of the “c_flag” are illustrated with enlarged perforations in Fig. 8.

The STOP command defines contact elements to limit the travel range of seismic masses. The command creates stopper of circular or rectangular shape on masses with custom defined contact stiffness and damping values. Both parameters tune the penetration depth and bouncing effect at the mechanical contact. In this example, four circular stopper elements have been attached to the seismic mass in order to limit the travel range in uy-direction. In Fig. 8 the stoppers are located above and below of the anchors.

Out-of-plane motion components at certain points can be limited by the ZLIM command. It creates contact elements above or below of seismic masses.

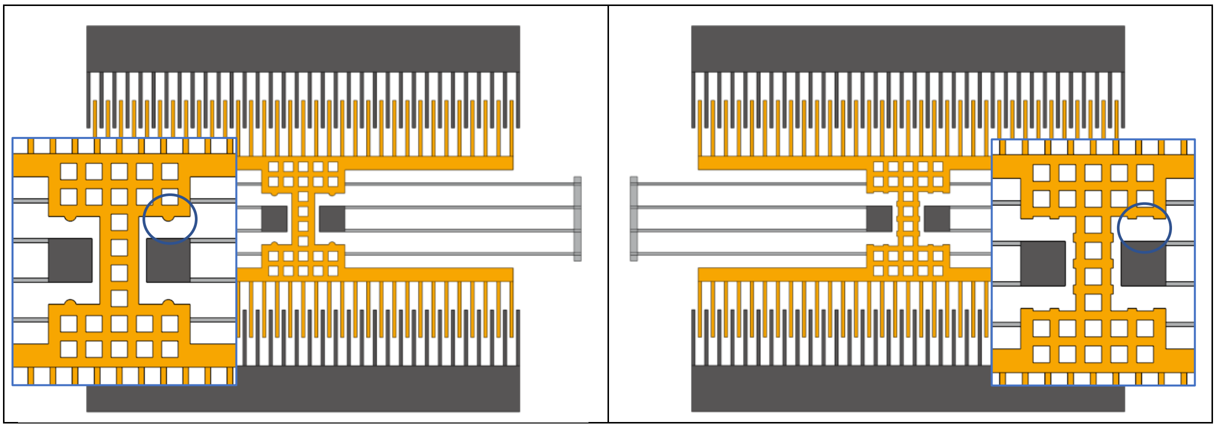


Fig. 8: Deactivated and activated intersections of perforations at outer edges (c-flag).

Creating corner fillets at clamps and spring connections

Sharp corners at springs cause stress concentrations known as notch effect. For reliable designs, corners at the clamp and between springs should be replaced by fillets with a radius of a few micrometers. Two different values for the fillet radiuses can be set in the Process Data window of the GUI. One radius is used for the spring connections to masses and anchors referred as “Fillet Radius Spring-Mass-Junction” and the other radius for the connections of springs to other springs referred as “Fillet Radius Spring-Spring-Junction”.

It is also possible to assign a specific fillet radius to individual springs or to remove the fillet (assign zero values) at some corners with the help of the SPRI command. Fillets are automatically removed at anchor or mass connections if there is not enough room for corner fillets (see Fig. 9).

Fillets create tangential lines to the adjacent model items. At the clamp to masses and anchors appear new volumes with sharp angles which are difficult to mesh in finite-element tools. To avoid a poor finite-element mesh quality, one can assign a tilt angle in the Solid Model Settings window. It is essential for applications in which a model export to ANSYS® or COMSOL® is planned.

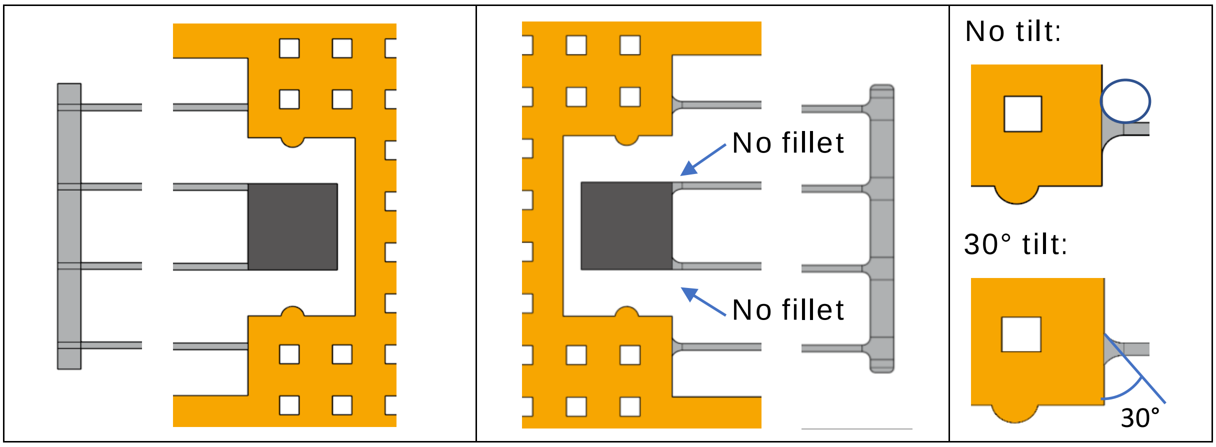


Fig. 9: Top view before and after adding fillets to clamps and spring connections.

Define mask undercut and etch sidewall offset

In the MODELBUILDER, all dimensions of the microsystems are related to the mask layout. During manufacturing, masks are under-etched, and the silicon structure shrinks according to the “Mask Undercut” value specified in the Process Data Settings window of the Graphical User Interface. All outer dimensions of the functional layer (masses, anchors, springs, combs, perforations and stopper) are affected by the value. For instance, the width of springs gets smaller and the gaps of combs or perforations get larger by twice of the specified value.

In addition, the vertical etch profiles are not ideal after manufacturing. The bottom face is usually not congruent to the top face of the functional layer. In contrast to the mask undercut, which always makes silicon structures thinner, the etch sidewall slope can point inwards or outwards. Furthermore, the sidewall slope deviation strongly depends on the orientation angle of the mask edges on the wafer surface. Therefore, different etch sidewall values can be assigned in the Process Data Settings window for vertical faces which are pointing “north”, “south”, “east” and “west”. Further values could be defined for directions in between (e.g. “north-east”). The etch sidewall offset of all other faces are interpolated.

The assigned etch sidewall values are lateral dimensions and specify the shift of the bottom edge compared to the same edge at the top face. Zero means it is an ideal vertical etch profile. Positive offset values make the silicon thinner as known from mask undercut and negative values make the silicon at the bottom edge larger compared to the edge at the top.

The mask offsets will be explained on the accelerometer example. The width of all horizontal beams is set to 3 µm. Exemplarily, the mask undercut is set to +0.2 µm and the etch sidewall offsets “north” and “south” are -0.5 µm. Horizontal beams get a trapezoidal cross-section with 2.6 µm beam width at the top and 3.6 µm beam width at the bottom. If the etch sidewall offset of all faces pointing “south” is changed from -0.5 µm to +0.5 µm, the cross-section becomes a parallelogram with 2.6 µm beam width at the top and at the bottom. The bottom face is moved 0.5 µm north compared to the top face. The asymmetric cross-section creates elliptical fillets at the bottom face corners. Results for both cases are shown in Fig. 10 Cross-section properties can be visualized by the BSEC command.

General etch sidewall offset data from the Process Data Settings window can also be redefined by other values for individual springs (see SPRI command). Analyzing the influence of etch tolerances is very important to evaluate the performance of microsystems. Values for etch tolerances in Fig. 10 have been highly enlarged for better visualization.

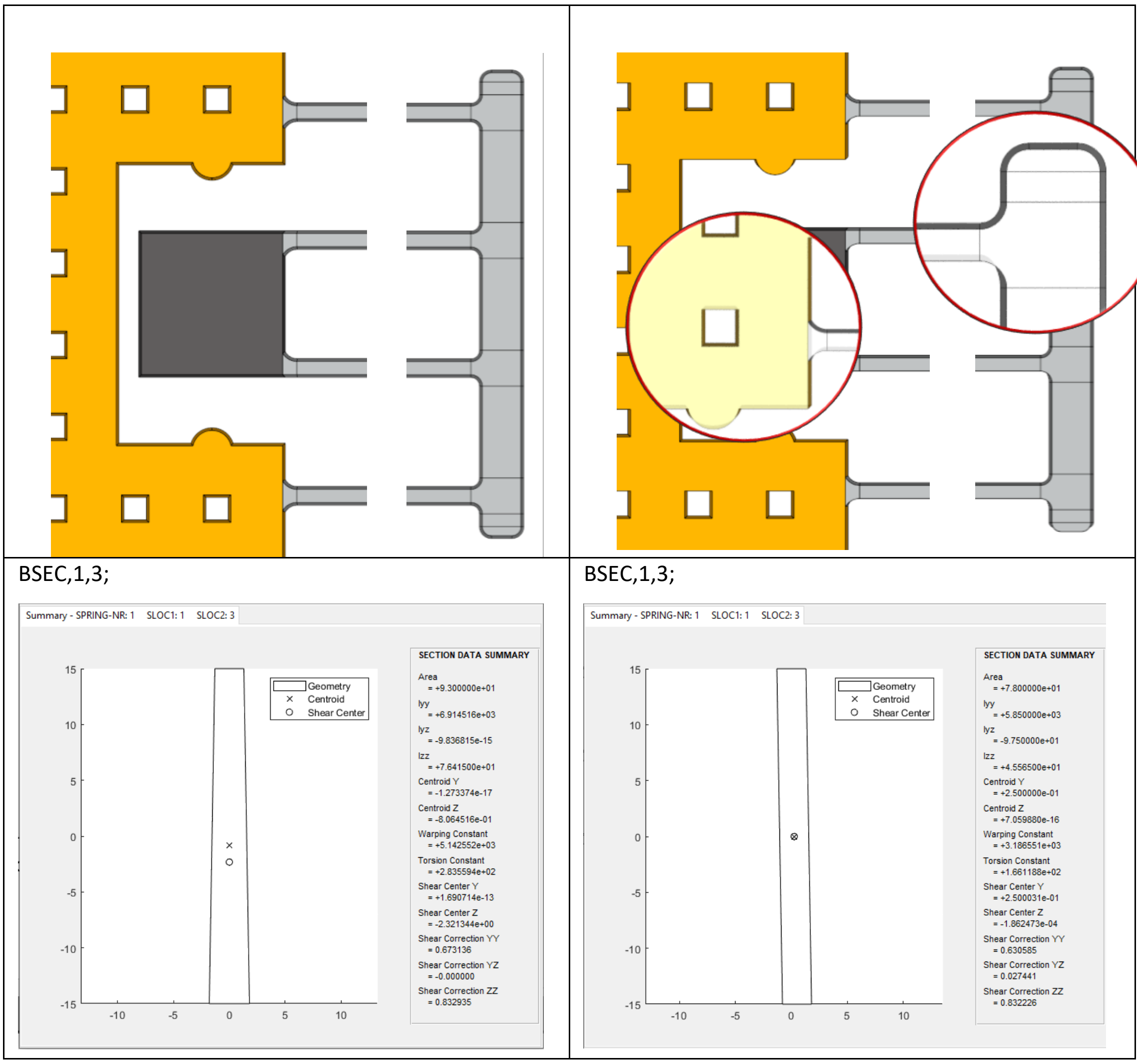


Fig. 10: Top view on springs and cross-section properties at different sidewall etch offsets.

Define material properties

The MODELBUILDER supports isotropic and orthotropic elastic material properties for MEMS. The latter one should be used for single crystalline silicon. Material properties are defined by the MATP command and orientation dependent values can be visualized by the MPLO command. Since masses and anchors are considered rigid, the elastic material properties are solely applied for Timoshenko beam elements.

Orthotropic material properties are either defined in the Global Coordinate System (GCS) or in a rotated Material Coordinate System (MCS) with specified “orientation angle”. Exemplarily, the acceleration sensor is etched from a (100)-wafer. The model coordinate system with mask edges is aligned parallel to the wafer flat which points in [110]-direction according to the Miller schema. Hence, the material coordinate system is rotated 45° regarding the global coordinate system. Material properties defined in MCS- or GCS-coordinates of Fig. 11 are identical.

```
% Properties of Si with 45° MCS-rotation
*IF,material,eq,1
   MATP,1,dens,2.329e-15;
   MATP,1,ex,130.02e3;
   MATP,1,ey,130.02e3;
   MATP,1,ez,130.02e3;
   MATP,1,nuxy,0.2785;
   MATP,1,nuxz,0.2785;
   MATP,1,nuyz,0.2785;
   MATP,1,gxy,79.554e3;
   MATP,1,gxz,79.554e3;
   MATP,1,gyz,79.554e3;
   MATP,,eps,8.854e-6;
   MATP,,mcs,pi/4;
*ENDIF
```

```
% Properties of Si in the GCS-system
*IF,material,eq,2
   MATP,1,dens,2.329e-15;
   MATP,1,ex,169.00e3;
   MATP,1,ey,169.00e3;
   MATP,1,ez,130.02e3;
   MATP,1,nuxy,0.06219;
   MATP,1,nuxz,0.27850;
   MATP,1,nuyz,0.27850;
   MATP,1,gxy,50.849e3;
   MATP,1,gxz,79.554e3;
   MATP,1,gyz,79.554e3;
   MATP,,eps,8.854e-6;
*ENDIF
```

```
MPLO,1;
```

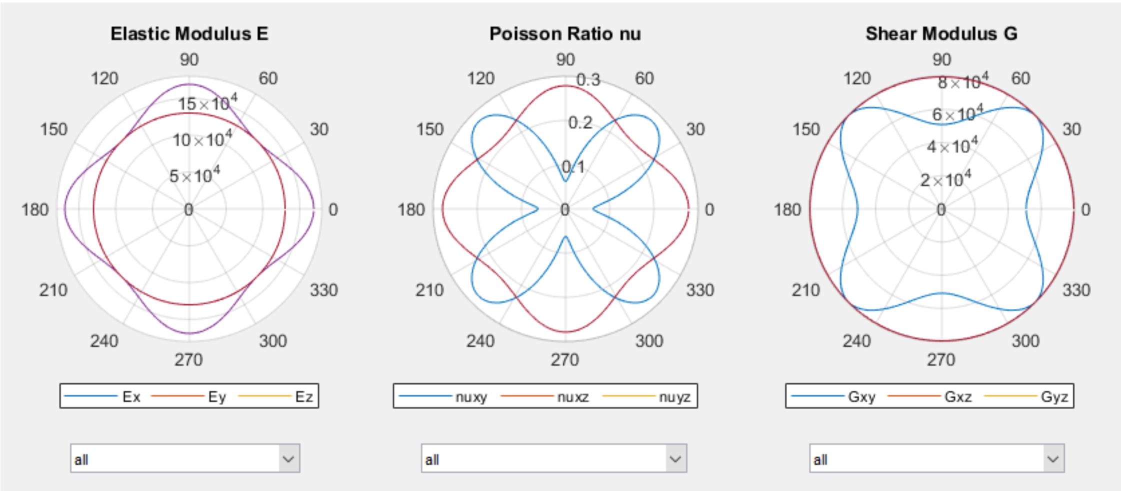


Fig. 11: Orthotropic material properties of single crystalline silicon.

Define master nodes

Master nodes (MN) are characteristic points on or in seismic masses where nodal loads can be applied, and displacement results can be observed after simulation runs. Master nodes can be used to extract stiffness data. For instance, the ratio of applied test loads and monitored displacements are the stiffness of a mechanical structure with regard to the master node location.

Master nodes are defined by the MAST command. Beside to master nodes, the rigid body's center of gravity (RB) and the spring connecting points (SLOC) are further points which allow for mechanical loads and displacement constraints. The position and reference number of master nodes and spring connecting points can be visualized by markers and labels as shown in Fig. 12. The annotation symbols are activated or deactivated in the “Model Tree” window.

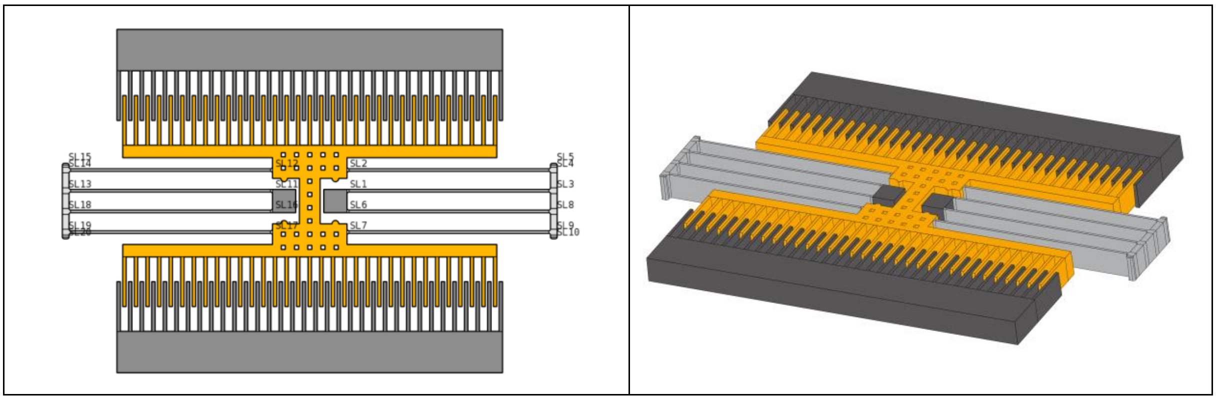


Fig. 12: Top view with annotation symbols and the final 3D-model of the sensor.

Creating different model variants

User defined parameters of the model input file (see [PARA](#) command) are listed in the [Design Variables](#) window of the Graphical User Interface. Design variables can be defined as “global”. Global parameters can directly be modified in the design variable window. Global parameters have a higher priority as other design variable. Global parameters can be assigned to geometrical dimensions or physical properties for structural optimization and case studies. After changing global design variables, a new 3D-model must be generated by clicking on the “Build Solids” icon of the GUI.

Fig. 13 shows four examples of model variants:

a) The extension bar is removed by setting the “build_ebar” flag to zero.
b) The electrode gap of combs is changed from 3 µm to 5 µm by “comb_eg”.
c) The number of fingers is reduced from 33 to 25 by the “comb_nf” parameter.
d) The capacitance type is changed by modifying the “build_capa” flag from zero to one or two.

Now, a 3D-model has been generated and is ready to use. The pre-process is finished and allows for numerical simulations and results evaluation.

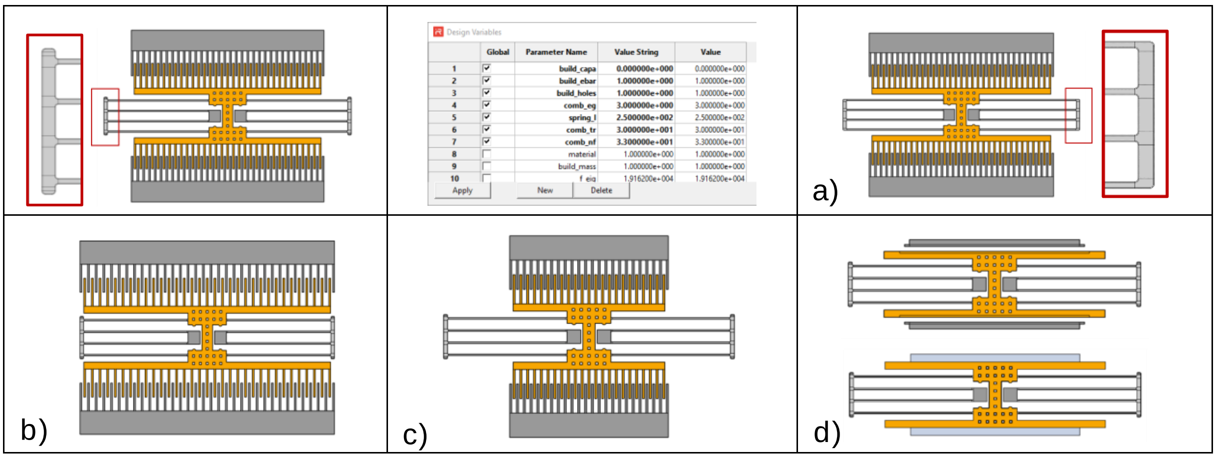


Fig. 13: Creating design variants by changing design variables in the GUI.

Settings for the reduced-order-models (ROM)

Prior running simulations, a few parameters must be assigned in the Simulation Settings and the ROM Settings window of the GUI. The parameters are:

- ROM mesh sizing: In the MODELBUILDER, connecting springs are represented by a series of Timoshenko beam elements. The mesh sizing can be controlled by two parameters which are "Spring Max. Mesh Divisions" and "Spring Element Size". The first parameter specifies the maximum number of beam elements allowed for springs and the second one the default length of beam elements for all springs of the model. The number of elements can be modified for individual springs by the SPRI command.
- ROM reduction stage: The parameter describes what motion degrees of freedom (DOF) are considered for simulation runs. Motion DOF are the rigid body center points (RB), spring connecting points (SL) and internal nodes (IN) between Timoshenko beam elements.
  Considering all motion DOF (level 0) is most time consuming but provides the highest accuracy. In contrast, considering only rigid body motion DOF (level 1) is the fastest approach but less accurate. Level 2 is recommended for most applications. Level 2 takes RB and SL motion DOF into account. All other DOF are eliminated by a Guyan condensation method. The default reduction level can be changed prior simulations by the REDS command.
- Modal superposition: A modal superposition solver (MSUP) can by applied for all three reduction levels. If MSUP is activated, the mechanical system is represented by a superposition of the lowest eigenvectors. The default number of modes is 20 but can be redefined in the ROM Settings window.

  Modal superposition is recommended for most applications. It is fast and numerically stable. Especially for transient simulations, the MSUP solver avoids tiny time-steps since high frequency motion components are eliminated automatically. The modal superposition solver can be activated or deactivated prior simulations by the MSUP command.
- Vlasov-torsion: Saint-Venant torsion theory is typically implemented for Euler-Bernoulli or Timoshenko beam elements and assumes that warping deformations (axial displacements) caused by torsion are not restrained. The assumption is only valid for torsion beams with uniform cross-sections and open ends referred "free warping".
  Restrained warping occurs at clamps, at beam connections and at jumps of the cross-sectional areas. The Vlasov-theory considers "restrained warping" and the related stiffness change of torsion beams. The effect is relevant for short beams and affects the tilt mode and the out-of-plane mode of the accelerometer shown in Fig. 14.

After the reduced-order-model settings have been assigned it is necessary to create a numerical ROM-model from the solid model representation. The system information for all three reduction levels and for activated and deactivated modal superposition are created by clicking on the "Build ROM" icon of the GUI. The system matrices are generated, and the model is ready for simulations.

Modal analysis and electrostatic softening effects

The "Modal analysis" calculates eigenfrequencies (natural frequencies) and eigenvectors of the electro-mechanical system. A modal analysis is performed after executing "SOLV, modal" in the GUI command line window. The default number of modes is set to six but can be changed by the MOPT command prior simulations.

For all types of analyses, load settings and solver commands can either be entered or copied into the command line or retrieved from a text file in the Assign Loads and Constraints window (go to "summary" and "Import from file"). Furthermore, all setting can be assigned in the graphical user interface.

After simulations are finished, a post-processor window "MODAL LS:1" of load step 1 appears automatically on the screen. In the GUI, modes can be selected, can be scaled in their amplitude and contour plots can be activated or deactivated. The "3D View" panel allows to realign the model, to attach coordinate selection markers and to "start" and "stop" animation sequences of modes. Fig. 14 shows some eigenmodes of the acceleration sensor.

Numerical values for eigenvectors at rigid body center points (RB), master nodes (MN) and spring location points (SL) can be listed by SNOD commands or by selecting solution items in the "Solution Selection" window of the GUI.

In a modal analysis, zero displacement constrains can be applied by the LOAD command to fix motion DOF of rigid bodies and spring connecting points. Displacements of anchors and fixed comb or parallel plate conductors are automatically set to zero. Inertial properties of masses, spring properties and modal analysis results are written to files in the working directory.

The shift of eigenfrequencies due to deflection dependent electrostatic forces is also considered in a modal analysis. Deflection dependent electrostatic forces appear in models with comb-cells and plate-like capacitors which change the electrode gap during operation. The frequency shift referred "Electrostatic softening" is widely used to modify eigenfrequencies during operation for resonant vibration sensors or angular rate sensors referred as "Mode-matching".

In the given example, plate-like comb capacitors are activated by setting the design variable "build_capa" to 1.

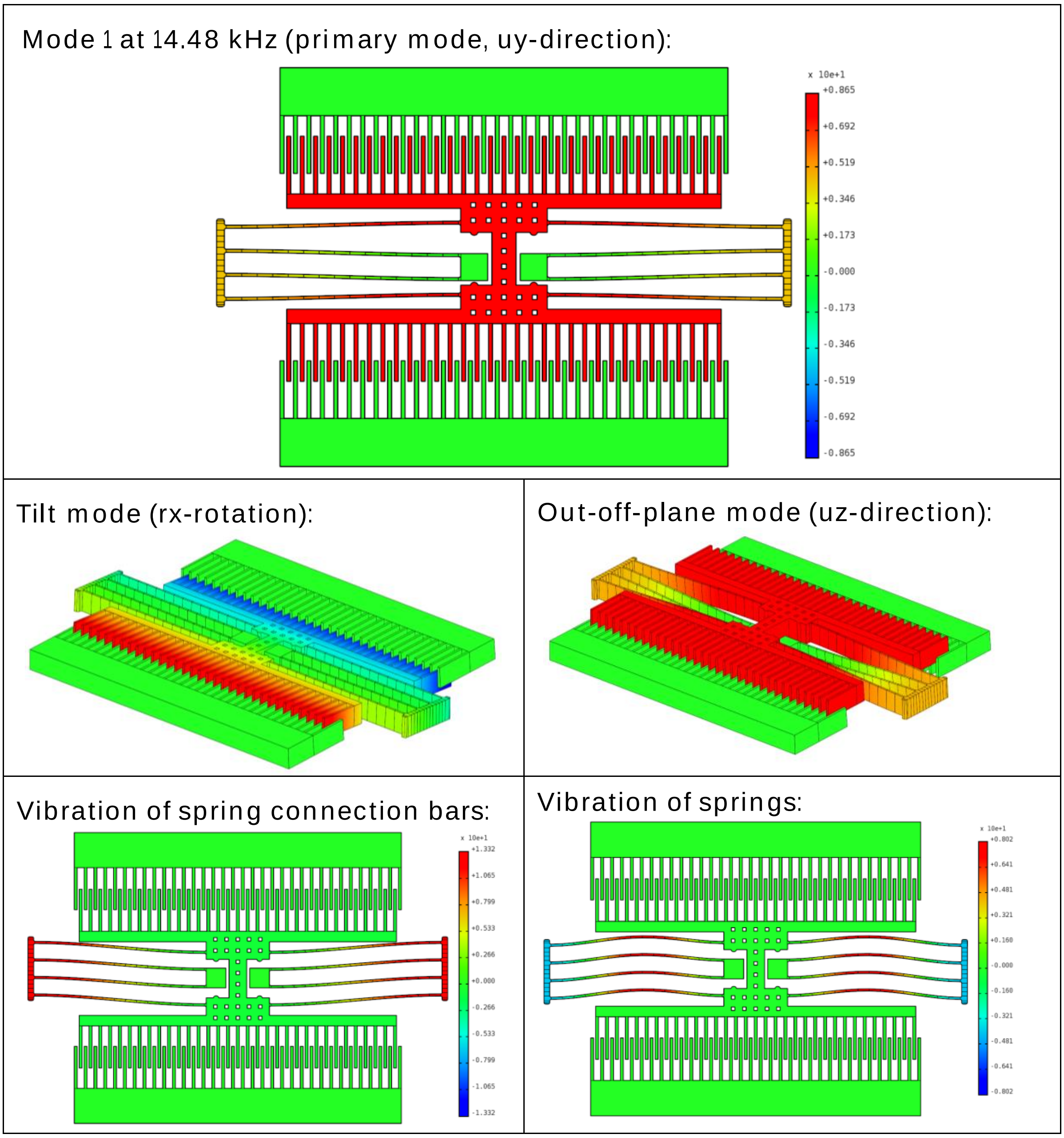
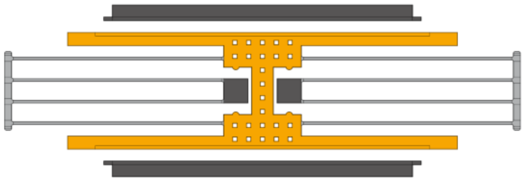


Fig. 14: Modal analysis results of the acceleration sensor (all motion DOF: RED,0).

Fig. 15 shows the command sequence and modal analysis results of the accelerometer with and without electrostatic softening effects. The applied DC-voltage on the upper and lower comb-cell must be smaller than the pull-in voltage. Otherwise, the system gets instable and imaginary eigenvalues appear whereby the character “i” is written right behind of the frequency value in the “Mode Selection” window.

In the given example, in-plane modes in uy-direction get a lower eigenfrequency and modes with out-of-plane components get a slightly higher eigenfrequency. The modal analysis evaluates second derivatives of capacitances at displaced positions referred operating point. In the given example, the tuning voltage is much higher than usual because of the highly enlarged electrode gaps.



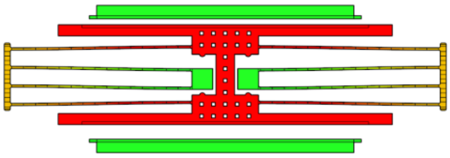
SOLV, modal; % No frequency tuning 	% Modal analysis with frequency tuning COND,sense+,mass,v_sens+; COND,sense-,mass,v_sens-; VSCR,mass,0; VSCR,v_sens+,+100; VSCR,v_sens-,-100; SOLV,modal;
% Mechanical eigenfrequencies 1: 1.9162e+04 (uy operating mode) 2: 6.7157e+04 (rx mode) 3: 8.6842e+04 (anti-phase spring mode) 4: 9.5224e+04 (in-phase spring mode) 5: 1.6290e+05 (uz mode) 6: 2.0955e+05 (z-tilt)	% Tuned eigenfrequencies 1: 1.8841e+04 (uy operating mode) 2: 6.7426e+04 (rx mode) 3: 8.6842e+04 (anti-phase spring mode) 4: 9.5219e+04 (in-phase spring mode) 5: 1.6298e+05 (uz mode) 6: 2.0955e+05 (z-tilt)

Fig. 15: Frequency tuning of eigenmodes due to DC-voltages on combs.

Static simulations and DC-sweep

"Static simulations" calculate characteristic properties of the system such as stiffnesses in certain directions or capacitances and their derivatives for a single load case. In practice, DC-sweep operations are preferred because sweep operations determine the static response of the system to a varying input parameter in a specified range and not just for a single load step. Possible sweep parameters are mechanical loads (forces, moments, transversal or angular acceleration) and degree-of-freedom constraints (displacements, rotations and voltages).

A DC-sweep is important to analyze nonlinear effects related to electrostatic quantities or electrostatic interactions with the mechanical domain. Examples are capacitance-stroke-functions and voltage-displacement-functions.

Constant mechanical nodal loads and DOF constraints are defined by the LOAD command, inertial loads by the ACEL command and DC-voltages sources by VSCR command. Varying loads and constraints are specified by the DCSW command. Constant and varying loads can be superimposed. All load and solution commands can likewise be defined in the graphical user interface. Furthermore, it is possible to change model settings by REDS and MSUP commands.

The solution routine with all defined loads and DOF constraints is started by "SOLV, static". A post-processor window "STATIC LS:1" of load step 1 appears automatically on the screen in the same way as discussed in the "Modal analysis" section. Several examples with commands and results are shown in Fig. 16.

The upper line of Fig. 16 illustrates how stiffness data can be extracted from a static simulation. In the first case, a displacement constraint of 8 µm in uy-direction has been applied by the LOAD command. The reacting force is 174.97 µN and leads to a stiffness of 21.87 µN/µm.

In the second case, the obtained reacting force has been applied by the LOAD command and displacements have been evaluated. As expected, results are identical. It should be noticed, that the peak displacements of the color plot are slightly larger as displacements at the center of gravity which comes from rx-rotations caused by the assigned etch sidewall slope.

Trapezoidal cross-sections of beams cause additional rx-tilt and parallelogram shaped cross-sections cause additional uz-shift (out-of-plane motion) which are superimposed to the primary uy-displacements. The uz-contour plot can be used to visualize rx-tilt and uz-shift of the acceleration sensor.

The next two examples of Fig. 16 show simulation commands and result plots for acceleration and voltage loads. Both loads are much higher than usual because of the enlarged spring width and electrode gaps of the model. Voltage loads generate deflection dependent electrostatic forces which are solved iteratively. In the current example, the travel range of comb-cells in closing gap directions is limited to 80 % of the electrode gap measured at the center of the overlapping area. The minimal remaining gap as well as the contact stiffness can be changed in the Simulation Settings window of the GUI. Furthermore, the travel range is limited by stopper elements.

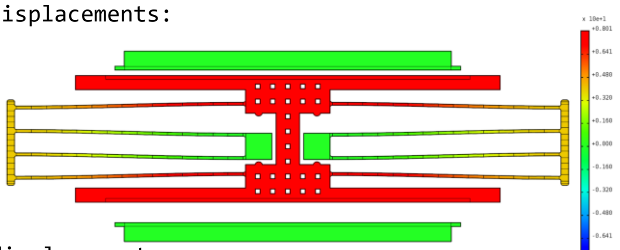
Commands of example four of Fig. 16 demonstrate how capacitance-stroke-functions can be calculated by DCSW commands. All comb and plate capacitances to be considered must be linked to voltage ports by the COND command. All other capacitances are ignored what speeds up simulation runs, especially for transient simulations.

In the given example, voltages are set to zero to avoid electrostatic forces. In practice, different local comb and plate capacitances can be merged to the same voltage ports and form a single global capacitance. Global capacitances are accessible for results evaluation by the SCON command or in the Graphical Users Interface using the features of the “Solution Selection” window.

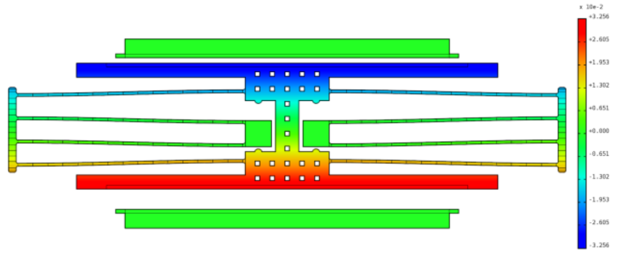
The last example of Fig. 16 illustrates how voltage-displacement-functions can be simulated by DCSW commands. If the sweep range is set high enough, pull-in and release (pull-out) voltages can be determined for system evaluations. The pull-in voltage is a critical or upper bound of the voltage where the systems gets instable. The release voltage depends on the minimal remaining gap at the contact elements and is the lower voltage bound where the system returns into the stable operating range. Both curves show a hysteresis for capacitors with varying electrode gaps.

```
% Displacement constraints
LOAD,rb,1,uy,8;  % µm units
%LOAD,rb,1,rx,0;
SOLV,stat;
SNOD,rb,1,fy;
SNOD,rb,1,uy;

    Results:
    RB1    FY = +174.963 µN
    RB1    UY = +8.000 µm
    STIFFNESS in y-direction
        FY/UY = 21.87 µN/µm

% Delete previous constraints
LOAD,rb,1,uy,del;
% Nodal forces (inverse problem)
LOAD,rb,1,fy,174.963;
SOLV,stat;
SNOD,rb,1,uy;
```

uy-displacements:

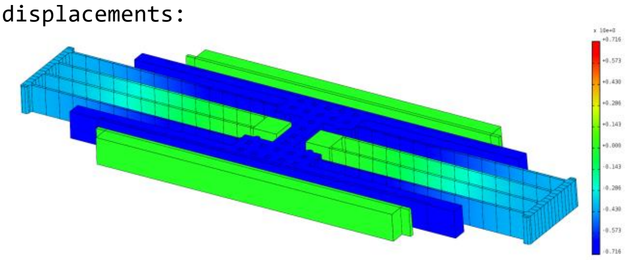


uz-displacements:



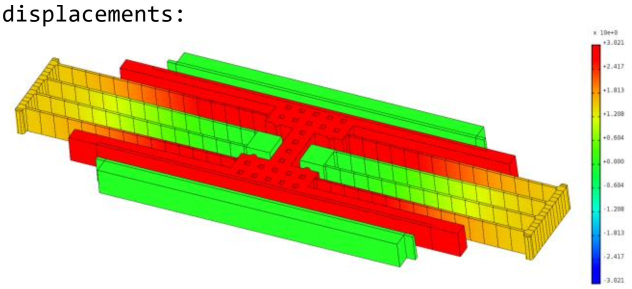
```
% Acceleration in uy-direction
LOAD,rb,1,fy,del; % Delete load
ACEL,uy,9.81e9;   % 1000xgrav.
SOLV,stat;
SNOD,rb,1,uy;
```

uy-displacements:

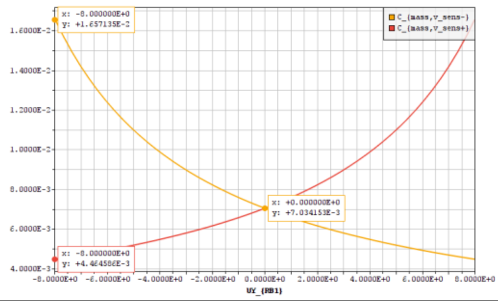


```
% Electrostatic forces
ACEL,uy,del;        % Delete load
COND,sense+,mass,v_sens+;
VSCR,mass,0;
VSCR,v_sens+,400;
SOLV,stat;
SNOD,rb,1,uy;
```

uy-displacements:



```
% Capacitance-stroke-function
COND,sense+,mass,v_sens+;
COND,sense-,mass,v_sens-;
VSCR,mass,0;
VSCR,v_sens+,0;
VSCR,v_sens-,0;
DCSW,rb,1,uy,-8,+8,51,0;
SOLV,stat;
SCON,capa,mass,v_sens+;
SCON,capa,mass,v_sens-;
```



```
% Voltage-displacement-function
COND,sense-,mass,v_sens-;
VSCR,mass,0;
DCSW,vscr,v_sens-,,0,500,501,1;
SOLV,stat;
SNOD,rb,1,uy;
```

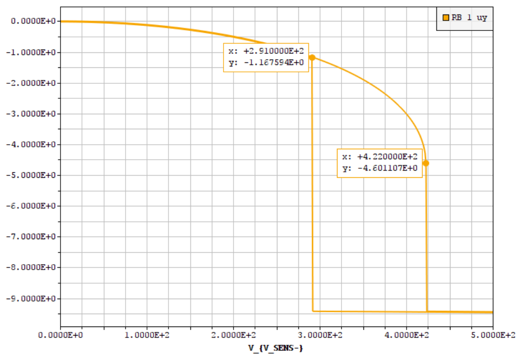


Fig. 16: Static and DC-sweep simulation commands with obtained result plots.

Coriolis forces caused by angular rates require mechanical components with non-zero velocities. This contradicts the application of static simulations to calculate the response caused by angular rates. However, static simulations are time and resource efficient.

The MODELBUILDER supports Coriolis forces by a combination of the DRIV and OMEG command. The first command selects an eigenvector from a modal analysis with a specific amplitude and frequency. Both values are necessary to calculate all velocity DOF for the mechanical domain. The second command applies angular rates and calculates the Coriolis force vector.

Finally, the “SOLV, stat” command starts the simulation process for the quasi-static mechanical response. Resonant amplification due to quality factors can be considered by scaling result plots in the GUI. An example is shown in Fig. 17.

```
% Coriolis forces
VSCR,,del; % Delete loads
COND,sense+; COND,sense-;
DCSW,off; % Delete sweep
REDS,0;   % Better accuracy
DRIV,1,19163,5,rb,1,uy;
OMEG,x,1000;
SOLV,stat;
SNOD,rb,1,uz;

STIFF UY:      21.87 µN/µm
FREQ  #1:      19163 Hz
Modal mass: 1.51e-9 kg
Coriolis force:-1.816 µN

% Results evaluation
DRIV,del; OMEG,x,del;
LOAD,rb,1,fz,-1.816;
SOLV,stat;  SNOD,rb,1,uz;
```

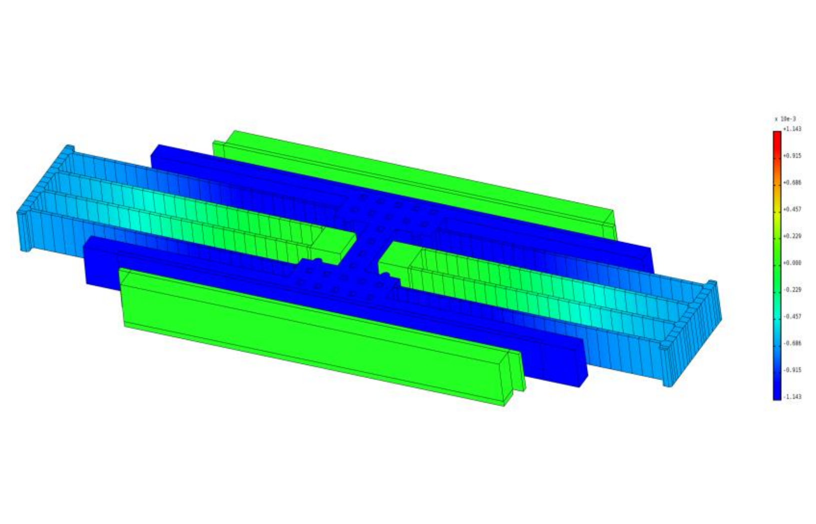


Fig. 17: Quasi-static displacements due to Coriolis forces for given angular rates.

Harmonic response analysis (AC-sweep)

“Harmonic simulations” are utilized to determine the steady-state response of a linearized system at sinusoidal loads. Results are amplitudes and phase angles or the real and imaginary parts of the output quantity in the specified frequency range.

The solver starts with calculating the operating point from specified DC-loads, linearizes the system equations and finally applies AC-loads in order to simulate the harmonic response. The terms DC- and AC-loads are adopted from electrical engineering and represent time independent und sinusoidal (harmonic) load components which are superimposed. Nonlinear solution schemas are applied for DC-loads in order to calculate the operating point, but the impact of AC-load components is linearized. This approach is known as small-signal case (SSC) in electronics.

Nonlinear effects should be simulated with care in a harmonic response analysis. For example, electrostatic forces are proportional to the square of the applied voltage. Linearization is only applicable if the AC-voltage is much smaller than the DC-voltage. Otherwise, second harmonics strongly contribute to the response but are ignored by theory what leads to wrong results. Furthermore, capacitances of gap varying combs and plate-like capacitors are inverse proportional to displacements. A linearization is only applicable for displacement amplitudes which are small compared to the electrode gap.

For the mechanical domain, DC- and AC-loads or constraints are defined by LOAD, ACEL, OMEG and for the electrical domain by VSCR and ISCR commands. The frequency range, number of datapoints and linear or logarithmic frequency spacing are defined by the HARF command. Finally damping must be defined for the mechanical domain to avoid singularities at resonances as shown later.

The solution-process is started by the "SOLV, harm" command. After simulations have been finished a post-processing window "Harmonic LS:1" appears on the screen.

The "Data Plot" window provides access to amplitude- and phase-frequency-response curves. The response curves can be plotted with linear or logarithmic axes scaling or the obtained data are listed in several output formats. Result items are either selected graphically in the "Solution Selection" window of the GUI or by SNOD and SCON commands in batch mode.

In the MODELBUILDER, damping can be defined in three different ways:

- Alpha-beta-damping (Rayleigh damping): The general idea of the Rayleigh approach is to superimpose a damping matrix from a weighted combination of the mass and stiffness matrix whereby the mass matrix is scaled by "alpha" and the stiffness matrix by a "beta" multiplier. It can be shown that the "beta" multiplier linearly increases damping regarding frequencies and the "alpha" multiplier decreases damping inverse proportional.
  For MEMS design, the "alpha" and "beta" multipliers are often adjusted in such a way that the damping ratios of two characteristic frequencies (eigenmodes) are captured precisely (e.g. drive and sense modes of angular rate sensors). Rayleigh damping can be assigned by the RDMP command and is applicable for harmonic and transient simulations.
  Drawbacks of the Rayleigh approach are that the damping ratios can only be controlled for two frequencies (two eigenmodes) and negative multipliers might occur. Negative multipliers can cause negative damping ratios for high order modes. For this reason, the obtained damping ratio over frequency response calculated from RDMP settings should be visualized by the DPLO command before simulations are performed.

- Constant damping ratio: The CDMP command can be used to set a constant damping ratio for the whole frequency range of the harmonic response analysis. However, the approach is not applicable for transient simulations. Assigning a constant damping ratio is a simple and widely used approach and represents the amplitudes well for low frequencies (quasi-static cases) and close to the eigenfrequencies of all modes.
  Drawback is that the amplitude response between eigenfrequencies is approximated and deviations occur for moderate and high damping ratios.

- Modal damping ratio: The MDMP command assigns damping ratios to the eigenmodes of the system referred as modal damping. It is a very powerful approach because individual damping ratios can be assigned to modes and the dynamic response is represented with high accuracy. Modal damping is applicable for both, harmonic response analyses and transient simulations if the modal superposition solver is activated (see MSUP command).
  Modal damping is widely used for MEMS design. Appropriate damping ratios for modes can be obtained from fluid-flow simulations, from analytical assessments or simply from measurements on similar structures.

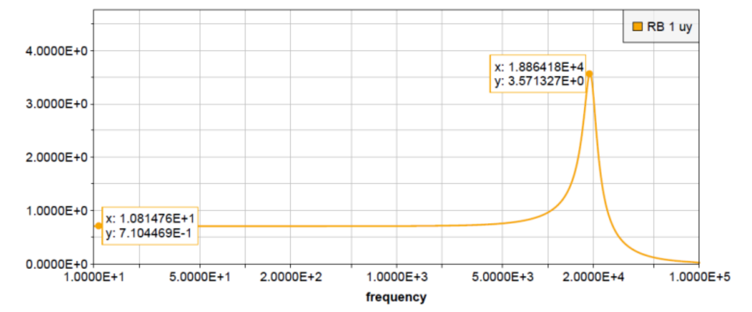
Fig. 18 shows examples of harmonic response analyses. In the first case, an acceleration load with a DC- and an AC-component of 9819 m/s^2 is applied for demonstration (see ACEL command). The DC-acceleration shifts the seismic mass -0.719 µm in uy-direction. Hence, capacitances and the capacitance derivatives of the upper and lower comb become different values at the operating point.

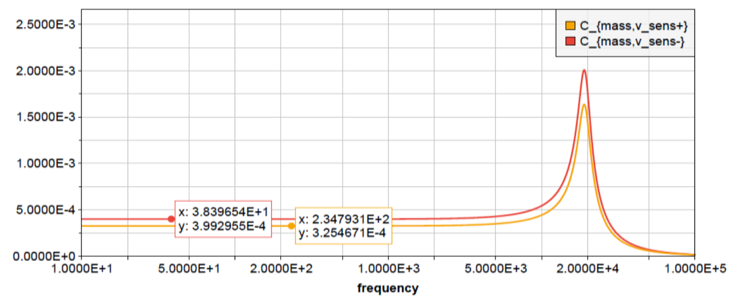
The motion amplitudes at low frequencies are equal to the static displacements. A damping ratio of 0.1 leads to about five times larger amplitudes at the eigenfrequency. Note, exactly five times larger values occur only for single-DOF systems where the reduction level has been set to one (see REDS command). For multi-DOF systems, the displacements or the mode shape at the eigenfrequency are not simply scaled static displacement states. For this reason, harmonic response analyses are vital to capture the performance of complex systems even if they operate at resonance.

The capacitance responses of the SCON commands in Fig. 18 show capacitance amplitudes. Total capacitances consist of the DC-capacitances at the operating point which are calculated in static simulations and the sinusoidal capacitance change calculated in the harmonic response analysis. Capacitance amplitudes depend on the capacitance derivatives at the operating point and motion amplitudes. The electrical current depends furthermore on comb or plate velocities and the applied voltages.

Take care to erase loads and constraints from previous examples. Erase commands will not be shown in following input files. Click “Loads and Constraints”, select “summary”, click on “Delete table” to erase all existing loads and constraints.

```
% DC- and AC-acceleration load
COND,sense+,mass,v_sens+;
COND,sense-,mass,v_sens-;
VSCR,mass,0,0;
VSCR,v_sens+,+10,0;
VSCR,v_sens-,-10,0;
ACEL,uy,9.81e9,9.81e9;
MDMP,all,0.1;
HARF,10,100e3,2000,log;
SOLV,harm;

% Amplitudes in uy-direction
SNOD,rb,1,uy;

% Capacitance amplitude
SCON,capa,mass,v_sens+;
SCON,capa,mass,v_sens-;

% Current amplitude
SCON,curr,mass;
```






```
% DC- and AC-voltage loads
COND,sense+,mass,v_sens+;
COND,sense-,mass,v_sens-;
VSCR,mass,0,1;
VSCR,v_sens+,+100,0;
VSCR,v_sens-,-100,0;
RDMP,freq,1e4,4e4,0.001,0.001;
HARF,10e3,40e3,2000,log;
SOLV,harm;
SNOD,rb,1,uy;
SCON,curr,mass;
```

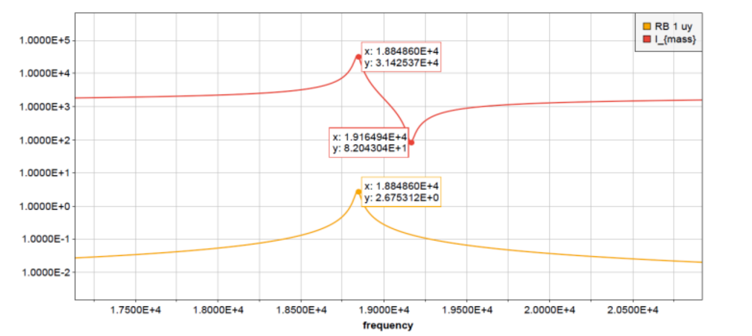


Fig. 18: Harmonic response analysis for applied acceleration and voltage loads.

The second example of Fig. 18 shows the admittance of the transducer. A DC-voltage has been set at the fixed combs and an AC-voltage of much lower amplitude is applied at the moving mass. The current flow (admittance) shows a resonance at the tuned eigenfrequency (electrostatic softening) and an anti-resonance at the original mechanical eigenfrequency. This effect is widely used for experimental characterization of MEMS to quantify the electrostatic softening effect.

Transient simulations

“Transient simulations” determine the model response in the time domain. The solution process is based on a Newmark time integration method with internal Newton-Raphson equilibrium iterations to account for non-linear effects related with electro-mechanical interactions. The TIME command specifies two essential parameters, the total simulation time and the time-step size for the numerical time integration (time increment). A proper time-step size should be chosen carefully. Small steps are time consuming and large steps lead to inaccurate results or even instabilities.

The time-step size depends on the highest frequency which significantly contributes to the response for given load situations. Vibration frequencies of accelerometers or the drive and sense frequencies of angular rate sensors clearly contribute most. Thinking in terms of a modal superposition or a Fourier decomposition, several high order modes might be relevant too. In general, the highest frequency of interest should be resolved by at least 50 steps per cycle. Systems with low damping (high quality factors) need smaller time-steps for accurate results. A quality factor of 1000 needs nearly 150 steps as shown for the Micro mirror actuator example.

The MODELBUILDER switches from a fixed time-step size defined by the TIME command to an automatic time stepping algorithm if contact elements tend to close. If a contact gap is smaller than 20 % of the initial gap, the time intervals are gradually reduced. At the moment of impact, small time steps are necessary for accurate results since contact elements strongly increases the response frequency of the system.

The contact behavior can be controlled by two parameters, the contact stiffness and the contact damping factor defined in the Simulation Settings window. Appropriate values should be assigned from the following considerations:

- Contact stiffness: Too large contact stiffness values lead to very small time-steps and the solver could fail to converge. Too small contact stiffness values lead to a large penetration depth and conductors of comb or plate capacitors might touch each other. The contact stiffness should be calculated from the expected impact forces at the contact divided by the accepted penetration depth. When difficulties arise, proper contact stiffness data can also be estimated in static simulations which are faster to run for testing.
  In transient simulations, the contact stiffness leads to an elastic impact and bouncing effects might occur. Therefore, a damping element is required to control the velocity ratio before the impact and after the contact release.
- Contact damping factor: Contact damping reduces the kinetic energy and bouncing can be lowered or even eliminated. In the latter case, the movable mass rests at the contact until external forces are smaller than restoring forces of suspension springs.
  The contact damping factor is not simply a fixed coefficient of a damper element since damping forces of the contact would jump strongly at the moment of impact. To avoid numerical problems, the implemented contact model leads to a smooth increase of damping forces depending on the velocity of the contact element, the penetration depth and the assigned “Contact damping factor”. If difficulties arise to predict a proper damping value, one should start with a very small or zero damping factor and increase the value until the expected behavior occurs.

Transient loads can be defined by LOAD, ACEL, OMEG, VSCR and ISCR commands. In practice, a few characteristic load-time-functions are required for transient simulations as known from electronic design automation. Above commands support “sine”, “step”, “pulse” and “ramp” functions. Arbitrary user defined functions can be defined in the MATLAB/SIMULINK environment (see Interface to SIMULINK®).

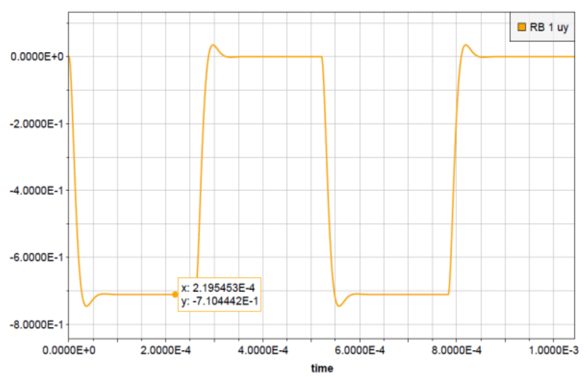
Multiple loads which are applied at different degrees of freedom can be superimposed. The solution process is started by “SOLV, trans” and a post-processing window “Transient LS:1” appears on the screen.

The transient response can be visualized and animated in the post-processing window by the “3D View” panels. Displacement and contour plots are available for each time step. Furthermore, “Data Plots” are utilized to plot time dependent result curves which can be chosen graphically in the “Solution Selection” window which appears after each simulation run.

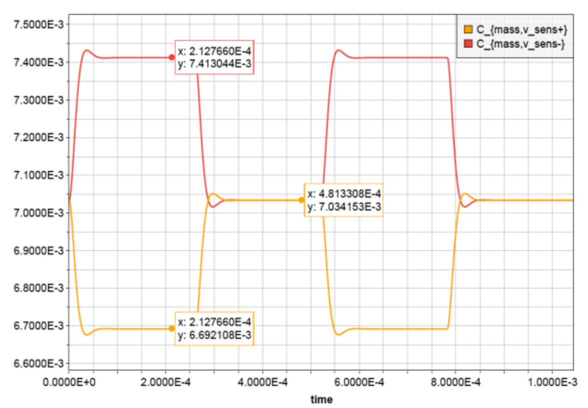
In Fig. 19 the transient response of an accelerometer to an acceleration pulse is simulated for demonstration. The total simulation time is 20 periods (inverse of eigenfrequency) and 100 timesteps are set for each period. The acceleration pulse is 9819 m/s^2 with a signal period of 10 cycles and 50 % pulse width. Optimal damping (dmpr=0.707) is assigned by the Rayleigh approach.

```
PARA,f0=19.163e+03;
PARA,dmpr=0.707;
COND,sense+,mass,v_sens+;
COND,sense-,mass,v_sens-;
VSCR,mass,0;
VSCR,v_sens+,+10;
VSCR,v_sens-,-10;
ACEL,uy,0,+9.81e9,puls,10/f0;
RDMP,freq,0.8*f0,1.2*f0,dmpr,dmpr;
TIME,20/f0,1/(100*f0);
SOLV,trans;
SNOD,rb,1,uy;
SCON,capa,mass,v_sens+;
SCON,capa,mass,v_sens-;
SCON,curr,mass;
```

uy-displacements:



Comb capacitances:



Current at mass port:

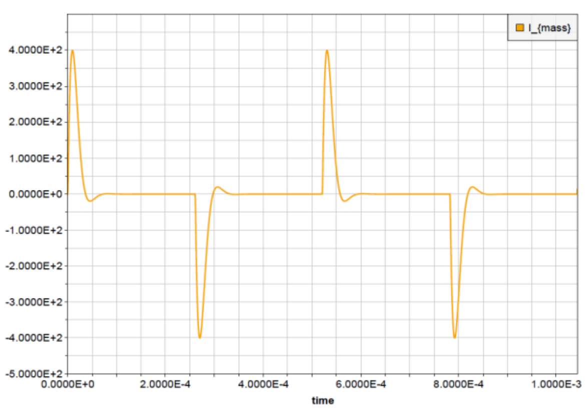


Fig. 19: Transient response of the accelerometer to an acceleration pulse.

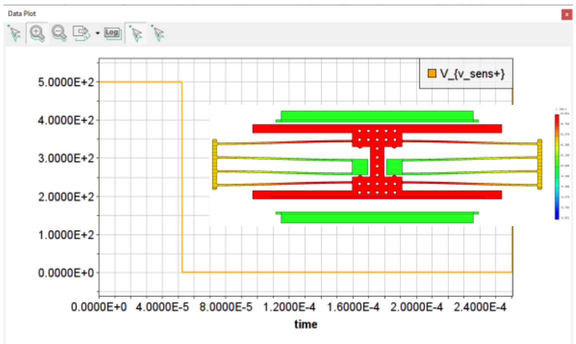
The uy-displacement plot shows a short transient overshoot as known from theory. The quasi-static state at the high-pulse corresponds to results from static simulations. Below are the capacitance-time-functions for the upper and lower comb and the current response at the mass port. In this example, the electrical current depends solely on capacitance changes caused by the moving mass since applied voltages are constant. 2000 time steps take about 5 seconds on a PC.

Fig. 20 shows another example where a voltage pulse is applied at the upper comb-cell. The voltage pulse is set to a value which is larger as the pull-in voltage calculated by the DCSW command.

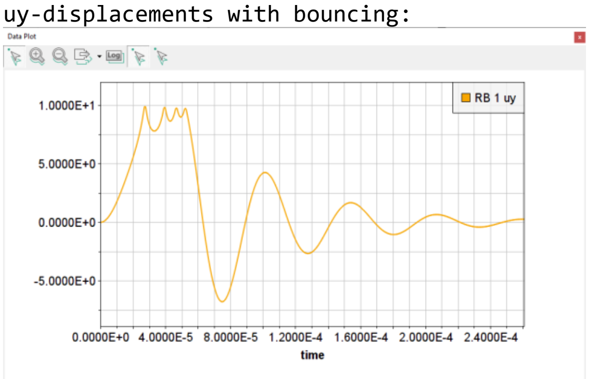
Consequently, contact elements at the stoppers limit the travel range to about 10 µm. In this model, the damping ratio is reduced to 0.15 in order to show the ring-down oscillations after contact release.

```
% Voltage pulse with contact
PARA,f0=19.163e+03;
PARA,dmpr=0.15;
COND,sense+,mass,v_sens+;
VSCR,mass,0;
VSCR,v_sens+,0,+500,puls,5/f0,1/f0;
RDMP,freq,0.8*f0,1.2*f0,dmpr,dmpr;
TIME,5/f0,1/(400*f0);
SOLV,trans;
SNOD,rb,1,uy;
SCON,curr,v_sens+;

Contact stiffness: 5000
Contact damping:   0.0 (1.0 no bounding)
```

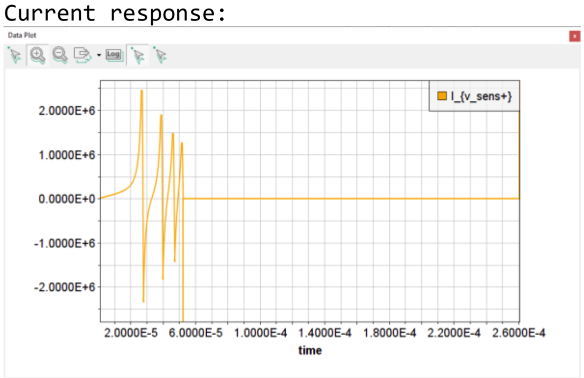
Voltage-pulse-function:



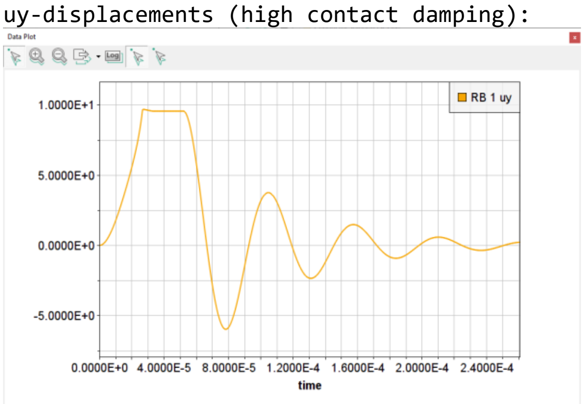
uy-displacements with bouncing:



Current response:



uy-displacements (high contact damping):



Current response:

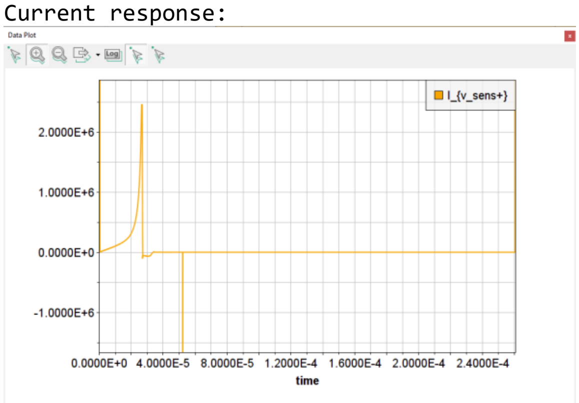


Fig. 20: Transient response to a voltage pulse with low and high contact damping.

In the following, the influence of contact damping will be shown for demonstration. In the first case (second line of Fig. 20), a low “contact damping_factor” with a zero value was applied for the transient simulation. As expected, strong bouncing can be observed in uy-direction. In the right plot, the electrical current at the voltage port is added to the displacement plot for comparison. The electrical current depends on the velocity of the seismic mass. At the moment of impact, bouncing creates almost ideal velocity jumps which are reproduced in the current response. Two strong peaks in the current response are originated from the sudden rise and fall of the voltage signal in the time domain.

The third line shows the transient response of the same model at a high “contact damping factor” with a value of 1.0. Bouncing is strongly reduced, and oscillations of the displacement and current functions disappear. At the moment of impact, one can see a small penetration of the moving part into the contact target. The value can be controlled by the contact stiffness. More advanced contact models with specific features (e.g. adhesion) can be added in the MATLAB/SIMULINK environment (see Interface to SIMULINK®).

Model export to ANSYS® (Interface to ANSYS®)

For a consistent design flow, it is necessary to create finite-element models automatically. The MODELBUILDER provides an Interface to ANSYS® where MEMS models with given dimensional parameters and process data can be transferred to ANSYS® for highly accurate FE-simulations. The interface is an additional feature of the MODELBUILDER and requires a special license.

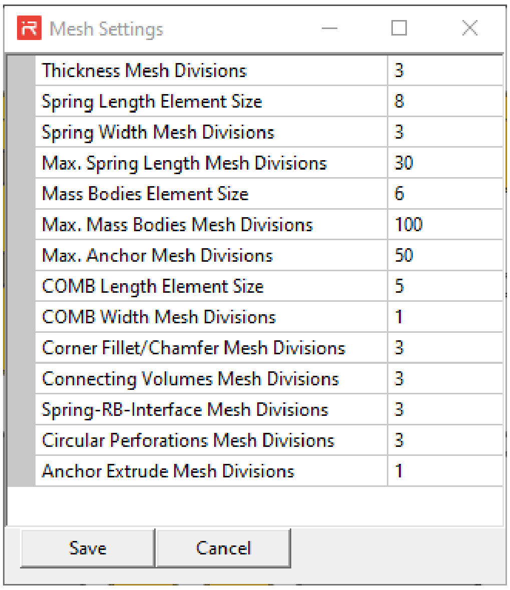
The Interface to ANSYS® transfers the current MEMS layout to ANSYS® by APDL commands (APDL - Ansys Parametric Design Language). For solid modeling, either a “bottom-up” or a “top-down” approach can be utilized (see ANSYS Users Guide) whereby the first approach is implemented in the MODELBUILDER software.

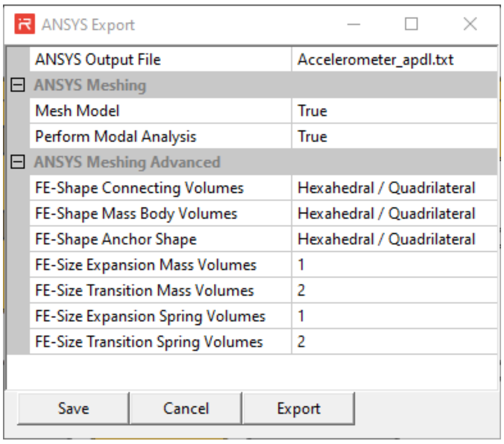
The “bottom up” approach starts from the lowest hierarchical order of solid model entities. 3D-models are generated from vertices which are referred keypoints, followed by lines, surfaces and volumes, to ultimately create a 3D-representation of the entire MEMS device. Individual solid model entities as rigid bodies, anchors, springs and combs can be selected by component names or reference numbers.

Next step is the finite-element mesh generation process. The APDL-model export file contains all information about the finite-element mesh density and required meshing operations which are performed in ANSYS®. Mesh density parameters can be set or modified in the Mesh Settings window of the Graphical User Interface as shown in Fig. 21.

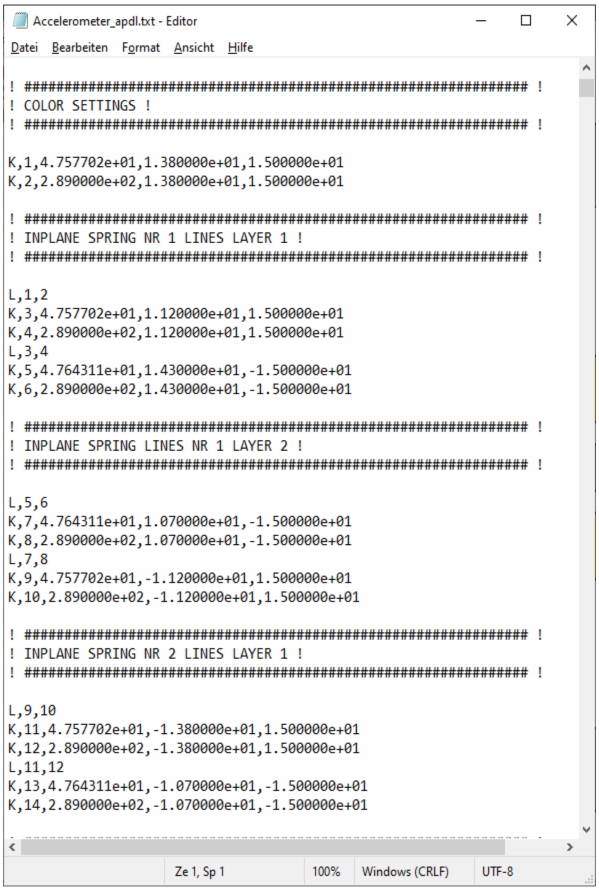
The APDL-model export file is generated by clicking on the “ANSYS Export” icon of the Graphical User Interface. At the end of the APDL-file are commands to run a finite-element modal analysis for demonstration.

ANSYS mesh settings in the MODELBUILDER:

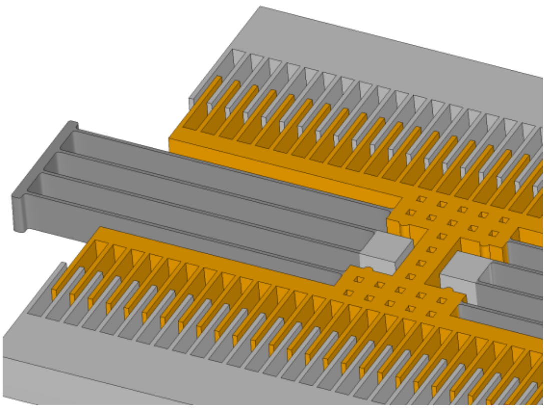




ANSYS-APDL model file:



ANSYS solid model:



ANSYS finite-element mesh:

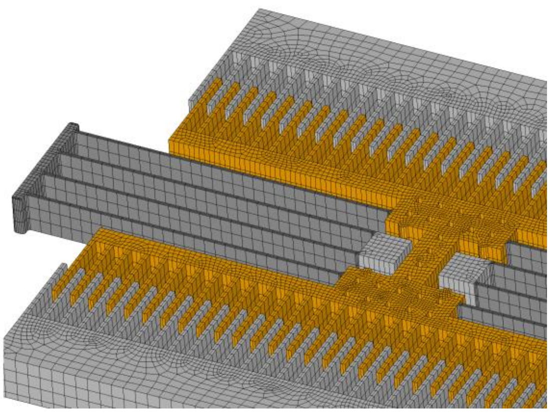
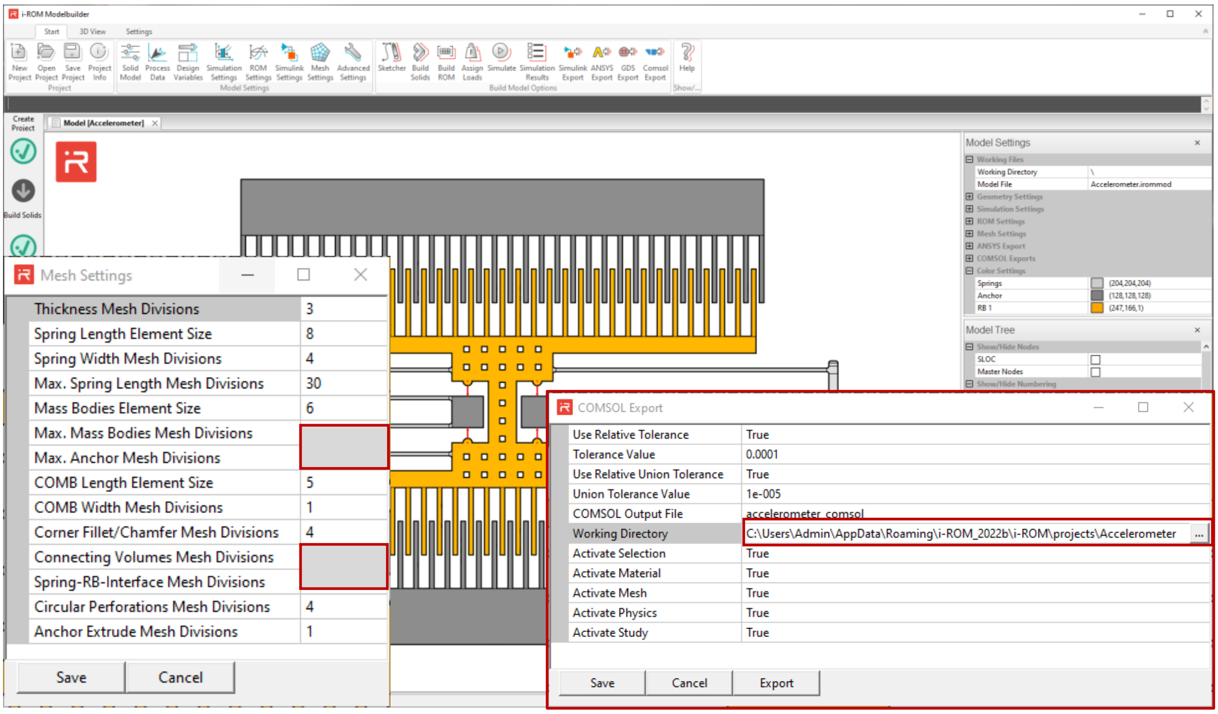


Fig. 21: Finite-element model export to ANSYS based on APDL-commands.

Model export to COMSOL® (Interface to COMSOL®)

The MODELBUILDER also provides an Interface to the Finite Element Program COMSOL®. It requires a special license. The Interface to COMSOL® supports the export of all solid model features including corner fillets, mask undercut and etch sidewall slope settings for subsequent finite elements simulations. Other optional features are:

- Activate Selection: Solid models are grouped into multiple components such as Anchor bodies, Springs, Comb conductors or Mass bodies. These selections are useful for assigning physics simulation data. Other selections apply to interface lines of springs, combs, perforations and fillets and are required for mesh settings and mesh creation routines.
- Activate Material: Assigns material properties to solid model components.
- Activate Mesh: Applies mesh settings to solid model components. The meshing process must be started manually in COMSOL as shown in the following steps.
- Activate Physics: Applies physics simulations settings (e.g. fixed constraints on anchor bodies). Additional settings based on preselected components can be assigned manually in COMSOL.
- Activate Study: Assigns simulation settings for a modal analysis. The simulation routine must be started manually in COMSOL.



The left part of figure above shows suitable mesh settings for the accelerometer example (see Mesh Settings window). Some mesh settings are not applicable in COMSOL such as maximum mesh divisions for mass bodies, anchors or connecting volumes. For these model components, COMSOL applies default settings (fine, normal, coarse) that can be modified in the COMSOL user interface. Spring width and corner fillet mesh divisions should be an even number to achieve better mesh quality with symmetric meshes.

COMSOL applies Boolean operations to connect areas to volumes (Convert to Solid) and to connect volumes to a solid model (Form union). A so-called repair tolerance must be applied to avoid gaps at the interfaces between geometrical objects caused by numerical tolerances.

Absolute or relative repair tolerances can be assigned for both types of Boolean operations. Tolerance settings can also be changed in the COMSOL user interface.

It is very important to specify the correct working directory in the COMSOL export window. The working directory is the folder where COMSOL reads the exported model files. There are two possibilities:

- Enter a backslash: Both, the MODELBUILDER working directory and the COMSOL working directory are the same.
- Enter or select a different folder: COMSOL model files must be opened from the specified folder on the same or a different computer. Click Save to save changed settings and click Export to create the COMSOL model files.

Three COMSOL model files appear in the MODELBUILDER project folder. The files are not transferred to the assigned COMSOL working directory!

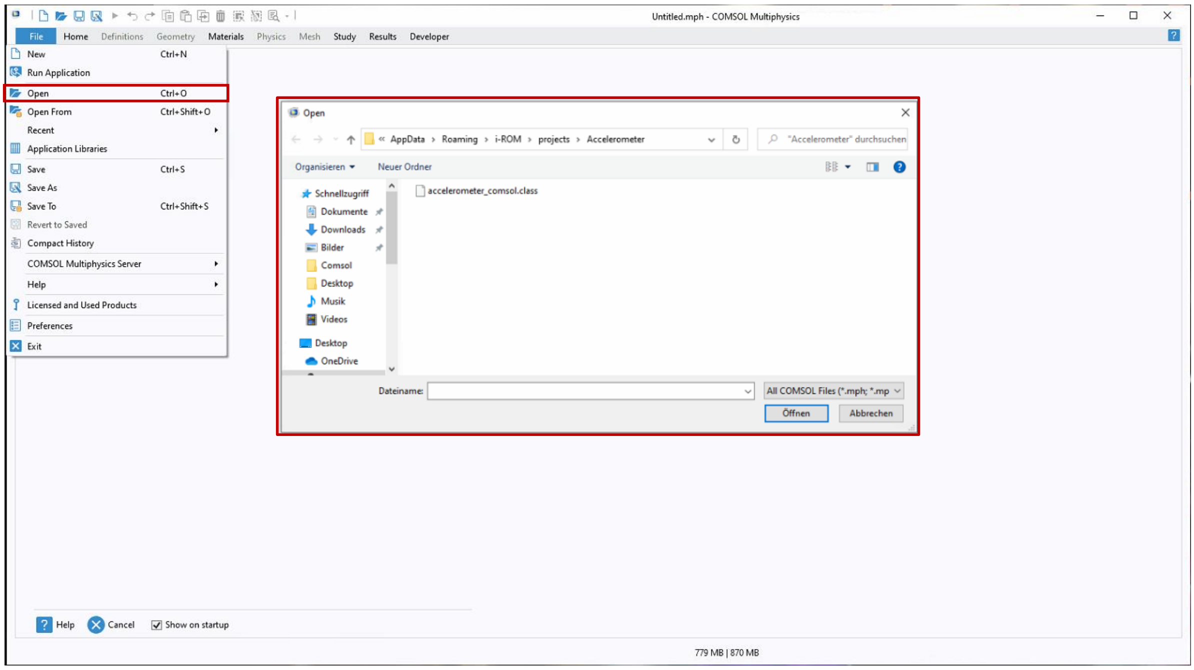
- accelerometer_comsol.txt: The "*.txt" files contain all solid model components and settings for selection, meshing, physics and study functions.
- accelerometer_comsol.java: The Java model file "*.java" contains all instructions for reading the model components from the "*.txt" file.
- accelerometer_comsol.class: Compiled COMSOL Application Builder file. The "*.class" file is created from the "*.java" and "*.txt" file. It can be opened directly in COMSOL.

The files can be copied to another folder on the same or another computer for subsequent finite element simulations. However, COMSOL needs to be installed on the computer to create a compiled "*.class" file. Otherwise a warning appears. Alternatively, "*.class" files can also be created on other computers using the "ComsolConvert.bat" file located in the ext-folder on the MODELBUILDER software:

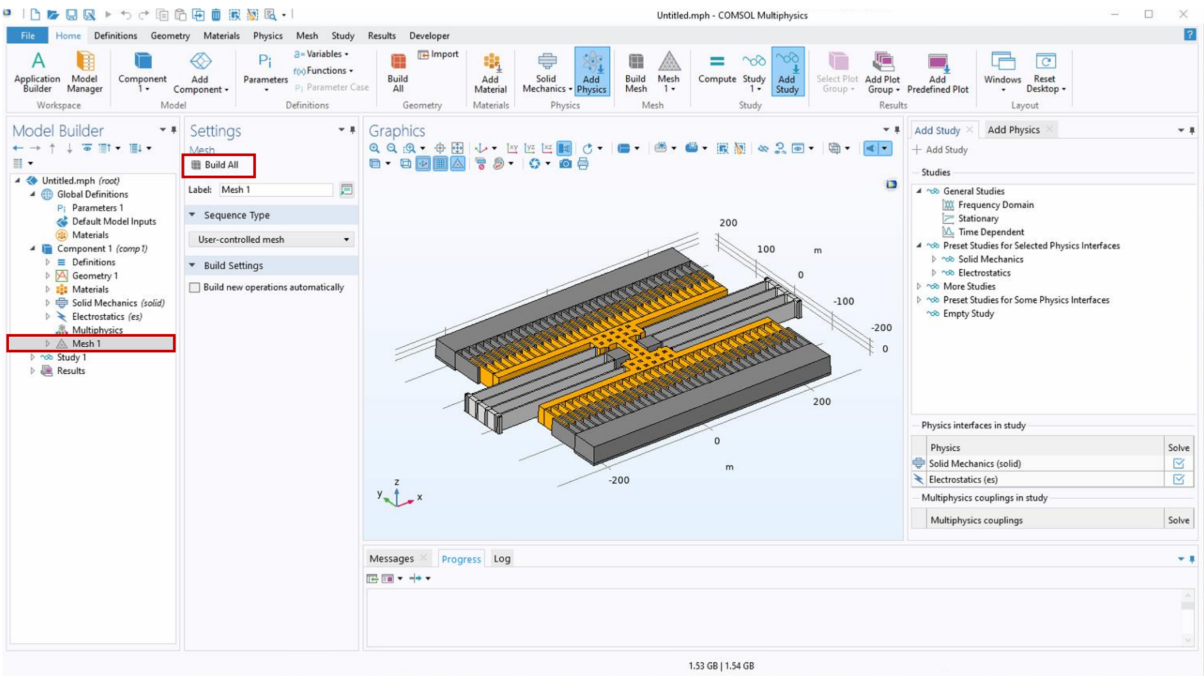
- Copy the "*.java" and "*.txt" files on the computer where COMSOL should be used.

- Copy the “.\ext\ComsolConvert.bat” file from the MODELBUILDER source code into the working directory. The directory name must be the same as the folder specified in the COMSOL export window.
- Run “ConvertComsol.bat” to create a “*.class” file for the COMSOL Application builder. Please note that the Application Builder requires the JAVA Programming Language to be installed on the computer. See the “COMSOL Multiphysics Programming Reference Manual” for details.

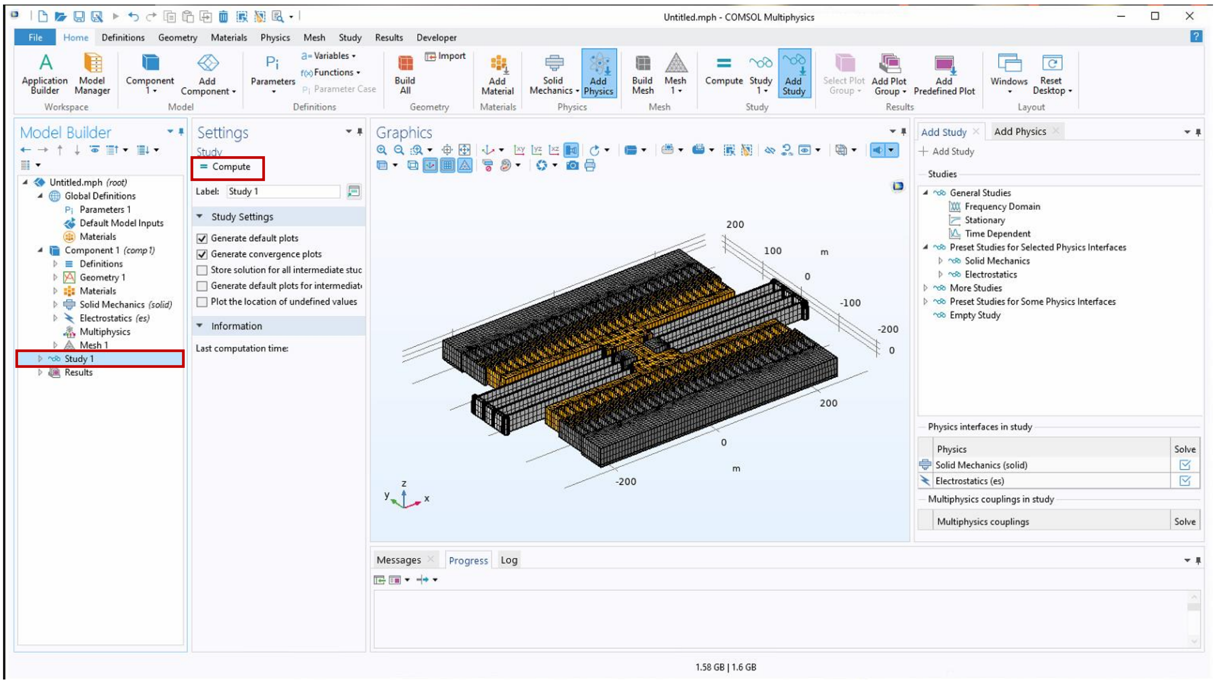
Start COMSOL and open “accelerometer_comsol.class”.



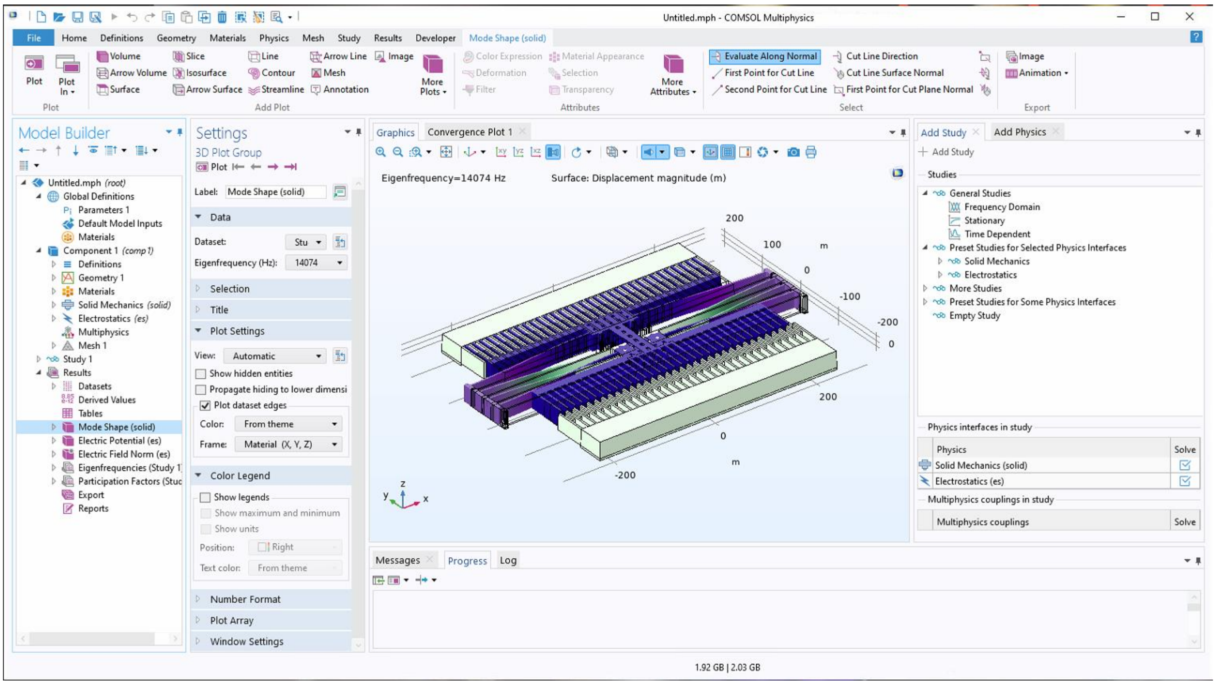
A COMSOL solid model appears in the Graphics window. Click on “Mesh 1” in the model tree and then on “Build All” to start the meshing process.



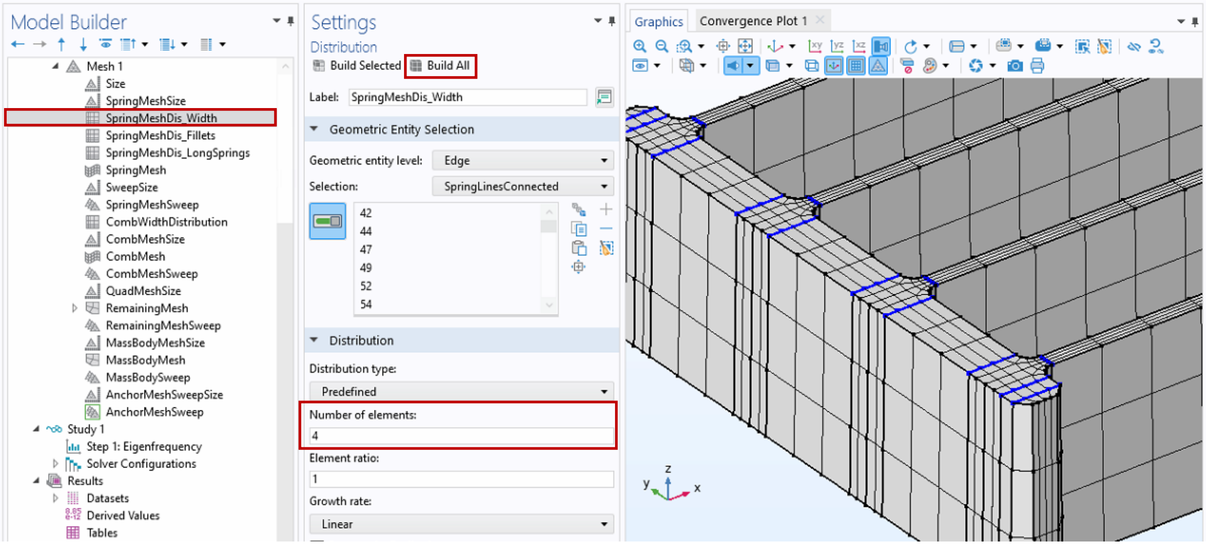
A finite element model appears on the screen. Click on “Study 1” in the model tree and then on “Compute” to start a modal analysis.



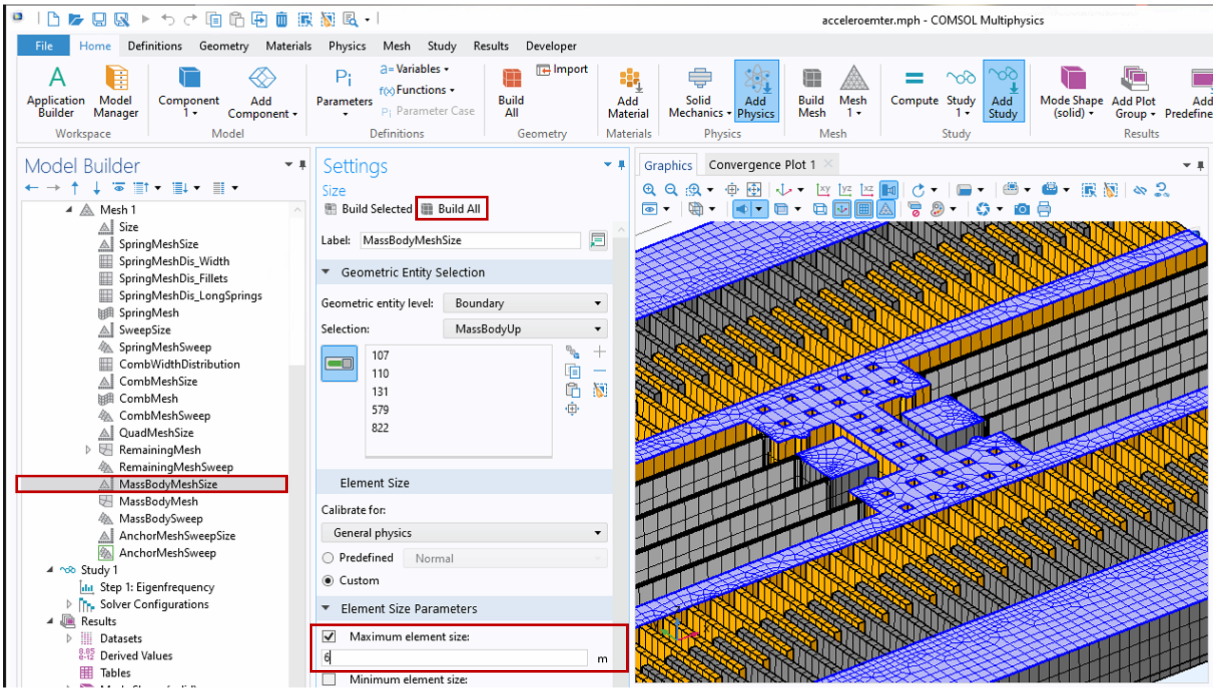
The results of the modal analysis are plotted on the screen. The first twelve modes are calculated according to the “Study 1” settings. The finite element model can be saved as COMSOL application file “*.mph” file. Application files open faster but take up much more disk space than “*.class” files.

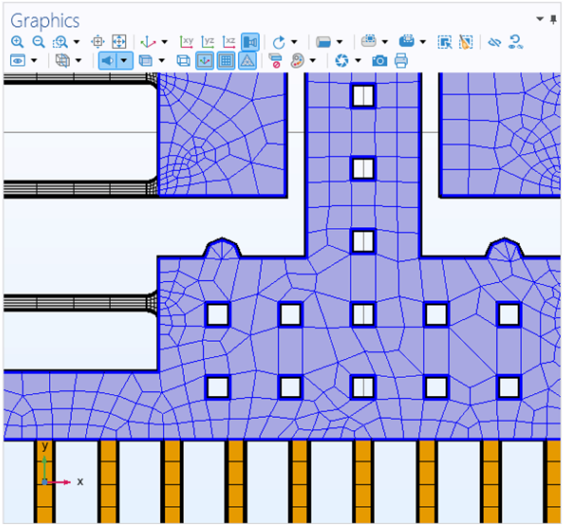
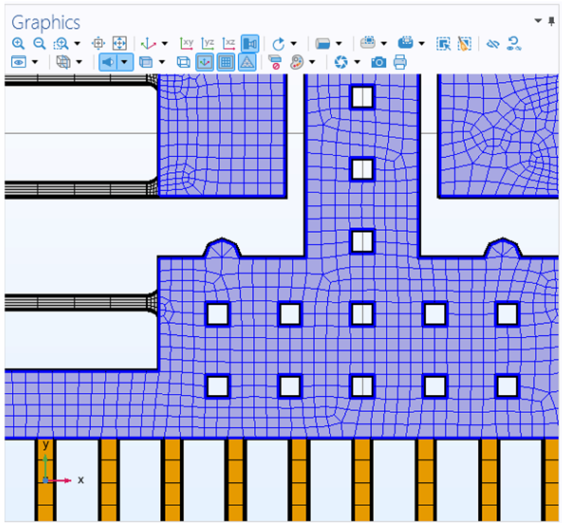


Appropriate mesh settings are usually model specific. Mesh settings can be easily changed in the COMSOL model tree. Click on “Mesh 1” and open the mesh settings and generation sequence assigned by the MODELBUILDER export interface. Click “SpringMeshDis_Width”. A settings menu appears that visualizes the assign number of elements for the spring width. The value can be changed and will take effect after clicking on “Build all”.

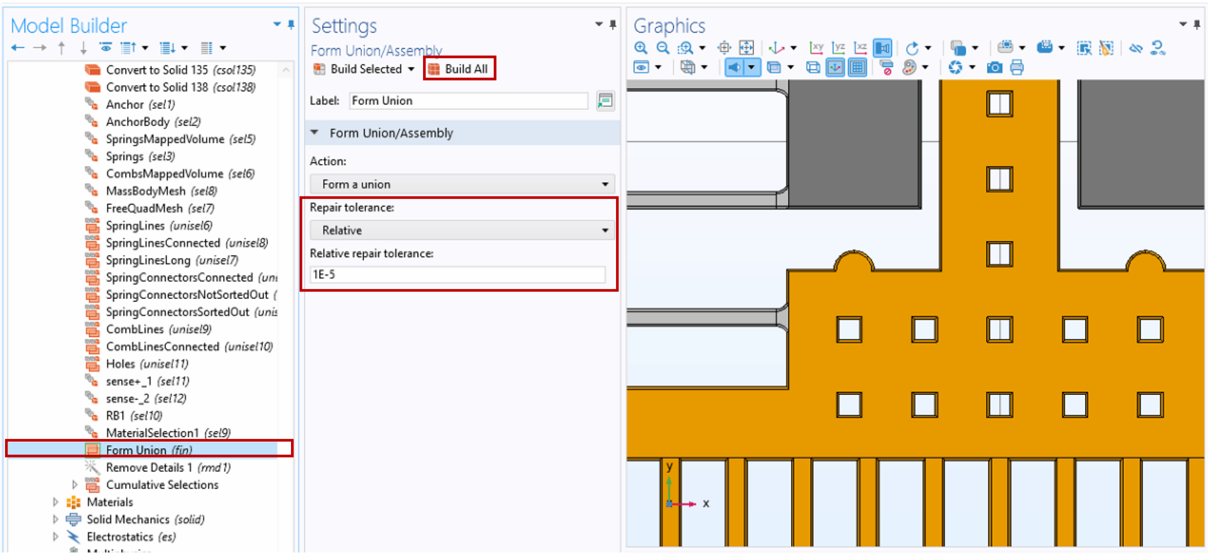


For demonstration, the element size of mass bodies and anchors should be changed from 6 to 3 in the “MassBodyMeshSize” settings sequence.



Maximum element size: 6	Maximum element size: 3
	

The assigned relative repair tolerance of 1e-5 can also be changed in the solid model settings menu “Form union”.



Model export to SIMULINK® (Interface to SIMULINK®)

The i-ROM MODELBUILDER provides an optional [Interface to Simulink®]() for advanced system simulations. The interface is an additional feature of the MODELBUILDER and requires a special license.

Simulink models are based on a signal-flow graph and allow for transient simulations. Signal-flow graphs are characterized by a unidirectional signal flow between library elements referred “blocks”**.** The exported MEMS model is a “subsystem” with “inputs” and “outputs”. The subsystem block contains a group of Simulink library elements which are connected by signal lines.

At least, MEMS models consist of a mechanical domain described by a force-displacement-relationship and an electrical domain characterized by a voltage-current-relationship. For signal-flow models, each domain must be decomposed whereby one quantity becomes “input” and the other one “output”. Most practical is to assign forces and voltages to “input” signals and displacements and currents to “output” signals.

Input quantities of the signal-flow model in SIMULINK are:

- Forces and moments on Master nodes (MN), spring connecting points (SL) and rigid body center points (RB) defined by the LOAD command.
- Linear and rotational acceleration loads defined by the ACEL command.
- Angular rates (velocities) around coordinate axes defined by the OMEG command. Angular rates are utilized for Coriolis forces and moments.
- Voltage loads at conductor ports defined by the VSCR command.

Input quantities and simulation settings (REDS, MSUP, TIME) from the last transient simulation performed in the MODELBUILDER are automatically transferred to the SIMULINK model by clicking on the “Simulink Export” icon of the Graphical User Interface. Alternatively, the SIML command can be utilized in a batch mode to create a Simulink model of the current design.

In general, all output quantities such as displacements at rigid bodies, master nodes and spring connecting points in the mechanical domain as well as capacitances and the electrical current in the electrostatic domain are available for results evaluation. Furthermore, the current settings for “Data Plots” are additional “pre-selected” output quantities of the SCOPE-blocks for automated results monitoring in system simulations.

Exemplarily, a Simulink model with the same loads and simulation settings as listed in Fig. 20 is generated and analyzed for comparison. The curves in the right lower part of Fig. 22 show the transient displacement response in uy-direction and the electrical current of the voltage port “v_sens+”. Simulink results agree with results of the MODELBUILDER main program.

The Interface to Simulink® is a very important feature of the i-ROM MODELBUILDER. It supports a comprehensive scientific investigation of the current MEMS design because all internal model quantities are accessible for a detailed model evaluation.

Internal signals such as capacitance derivatives, different kinds of forces (spring forces, damping forces, inertial forces, Coriolis forces, contact forces, electrostatic forces), current components (motion induced current, current due to the voltage change) and other data provide essential information on the functional behavior and related physical effects of the model.

Internal parameters, load functions, Simulink blocks and the signal routing can be modified for case studies in order to verify the performance at different load situations and environmental conditions. In general, the exported Simulink models can be linked to other User-defined model components for controller and system design.

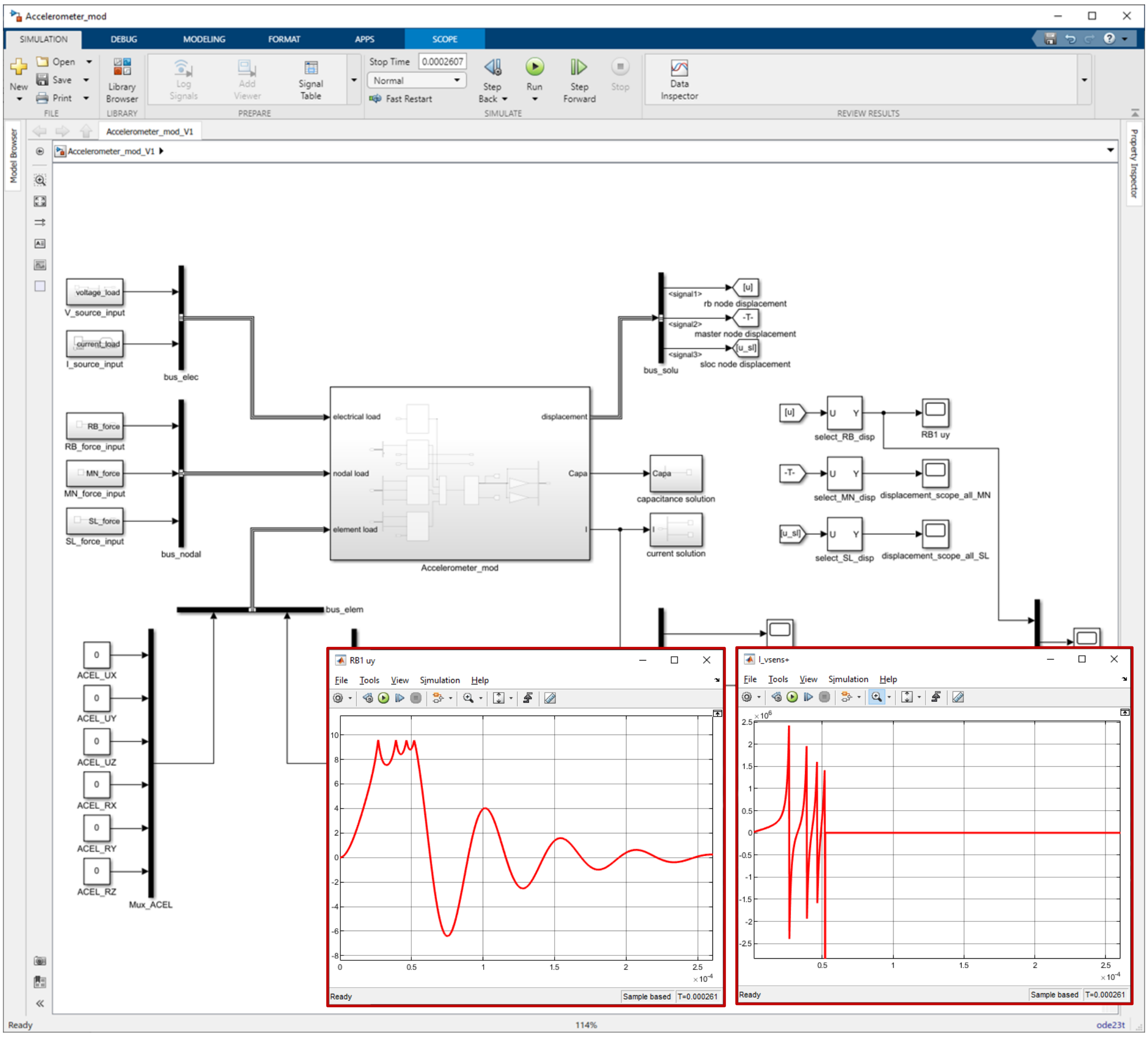


Fig. 22: Transient response to a voltage pulse with low damping in Simulink.

2 Angular rate sensor example

The next example demonstrates how to model and simulate a single-axis angular rate sensor in the i-ROM MODELBUILDER environment. All model items are defined in the “Gyroscope.irommod” input file what can be found in the “Gyroscope” project folder. Angular rate sensors are often referred gyroscopes.

The angular rate sensor in Fig. 23 consists of 6 mass bodies, 12 anchors, 132 connecting springs and 16 comb-cells for electrostatic actuation, capacitive sensing, and quadrature compensation. It is a z-axis gyroscope, what means the sensor detects angular rates (angular velocities, speed of rotation) around the vertical axis. In case of positive angular rates, the sensor rotates counterclockwise with its housing as marked by the red arrow. The speed of rotation will be measured and provides the output signal of the MEMS sensor product.

Capacitive micro-mechanical sensors transform the quantity to be measured into mechanical forces, forces create mechanical displacements and displacements provide the capacitance change for the electronic signal evaluation unit (ASIC). Inertial forces are utilized for accelerometers and Coriolis forces are widely used for angular rate sensors.

In rotating systems, Coriolis forces appear on moving masses which have non-zero velocities. For this reason, angular rate sensors need actuator components which permanently drive one or multiple mass bodies at a constant vibration amplitude. Therefore, MEMS based angular rates sensors are often referred “Vibratory gyroscope” in literature.

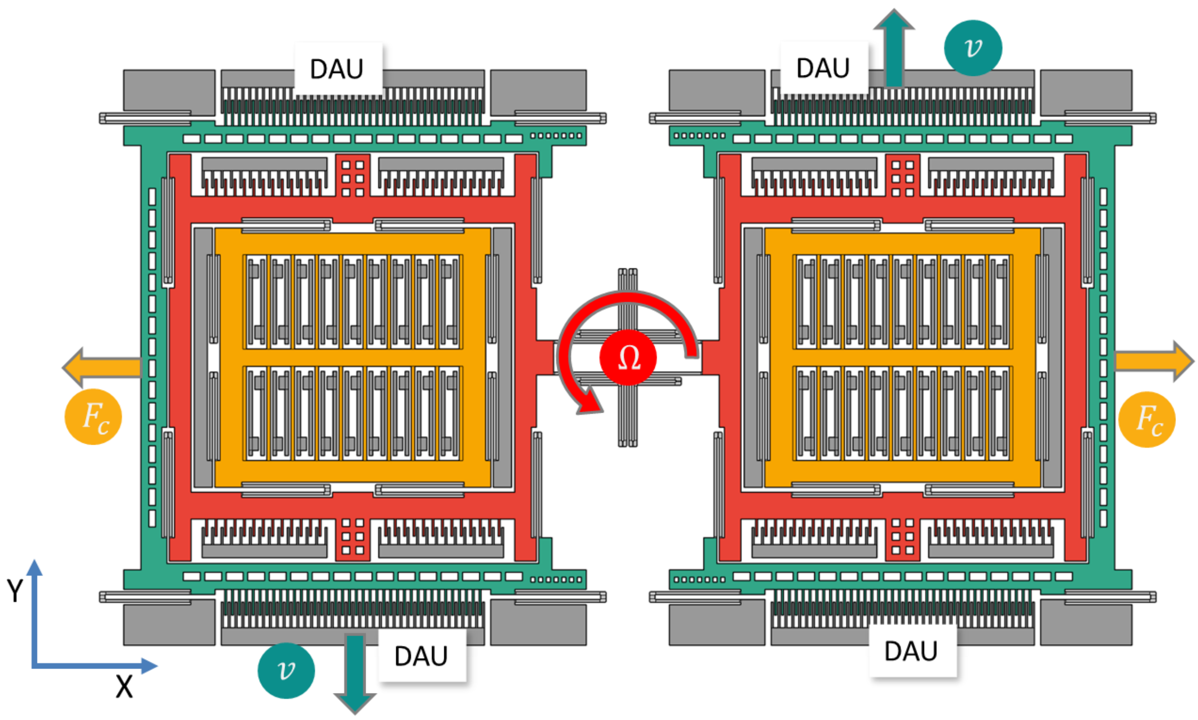


Fig. 23: Single-axis angular rate sensor with drive velocities and Coriolis forces.

From the physical point of view, the Coriolis force vector $\vec{F}_C$ acting on a seismic mass m is proportional to the angular rate vector $\vec{\Omega}$ (input signal) and depends on the velocity vector $\vec{v}$ of the oscillating mass.

$$\vec{F}_C = 2 \cdot m \cdot \vec{v} \times \vec{\Omega} \qquad \text{Eq. (1)}$$

In the example of Fig. 23, the green “Drive frames” are stimulated by electrostatic forces at the attached comb-cells referred “DAU - Drive Actuation Units”. The DAU create an anti-phase oscillation in uy-direction. Fig. 24 shows a snapshot in time where the right drive frame moves upwards, and the left drive frame moves downwards. Caused by the spring design, the red “Coriolis frames” in Fig. 23 follow the drive motion with almost identical amplitudes but the orange “Sense frames” rest at the initial position.

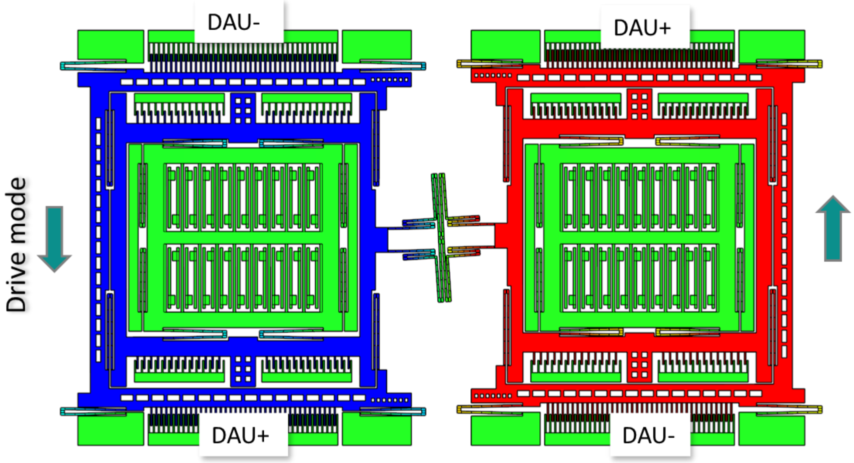


Fig. 24: Scaled drive motion amplitudes of the angular rate sensor.

If the gyroscope rotates around the z-axis, an angular rate Ω_z appears which creates tiny Coriolis forces at the “Drive frames” and at the “Coriolis frames”. In the same snapshot in time, the Coriolis forces point anti-phase in ux-direction as marked by the orange arrows in Fig. 23 and create motion as shown in Fig. 25.

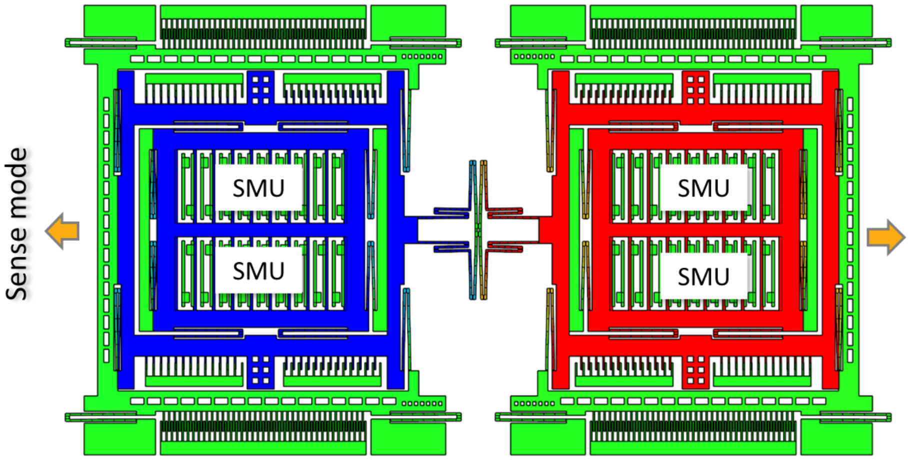


Fig. 25: Scaled sense motion amplitudes of the angular rate sensor.

Due to the spring design, the Coriolis forces at the “Drive frames” are without effect. The “Drive frames” are constraint in sense direction and can’t move in ux-direction.

In contrast, the Coriolis forces generated at the “Coriolis frames” drive both, the “Coriolis frames” and the “Sense frames” with almost identical amplitudes in sense direction. Ideally, the “Drive frames” move only in drive direction and the “Sense frames” move only in sense direction. Crosstalk between drive and sense is strongly reduced by the six masses design of the given example.

Finally, the sense motion is transformed into a capacitance change at the inner comb-cells of the “Sense frames”. Those comb-cells are referred “SMU – Sense Measurement Units” in the following sections.

The anchors of the SMU-combs are linked by interconnects to the voltage ports “V_SMU+” and “V_SMU-” which provide the differential sensor output ports for the electronic signal evaluation unit (see Fig. 26).

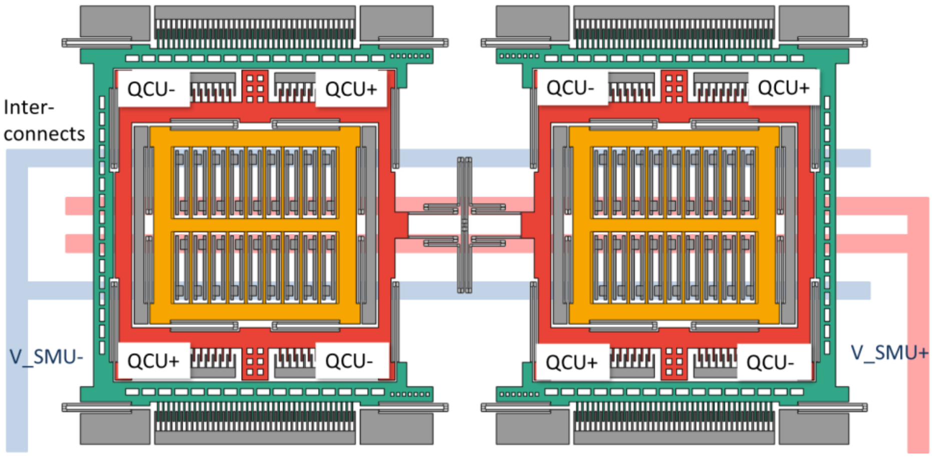


Fig. 26: Interconnects for sensing and comb-cells for quadrature compensation.

In real MEMS devices, the drive motion in uy-direction couples slightly to the sense motion in ux-direction. This crosstalk is mainly caused by asymmetries originated from manufacturing tolerances such as etch sidewall slopes, misaligned springs or misaligned comb-cells.

Even in the absence of angular rates, the crosstalk of real-world gyroscopes produces a signal offset known as “Quadrature”. The offset can be compensated by electrostatic forces at asymmetric combs referred “QCU – Quadrature Compensation Units” (see Fig. 26). Several other layouts are published in literature.

QCU-combs generate electrostatic forces in sense direction which depend linearly on displacements in drive direction. The magnitude of forces can be scaled by DC-voltages. Depending on the sign of the offset, DC-voltages are either applied at the “QCU+” or at the “QCU-” combs. In practice, the voltages are slowly increased until the signal offset disappears.

Model generation procedure

The model generation procedure is similar to the acceleration sensor example discussed earlier. Details of the current design can be found in the model input file “Gyroscope.irommod”.

The angular rate sensor consists of six mass bodies. The reference numbers are shown in the left and right upper corner of Fig. 27. Furthermore, three different types of comb-cells are used for the DAU, the SMU und the QCU-capacitances.

In practice, drive motion amplitudes stimulated at the DAU-combs are designed to be relatively large. Typical values are between 4 and 20 µm. Comb-cells of type “area” (area variation) are widely used for actuation of gyroscopes.

Sense motion amplitudes detected at the SMU are small, typically less than 0.1 µm. Comb-cells of type “dif1” or “dif2” (gap variation) are preferred for differential sensing of angular rates.

Finally, QCU-capacitors need to be asymmetric to create the desired force-displacement-relationship. The type “asym” has been set for demonstration (see COMB command).

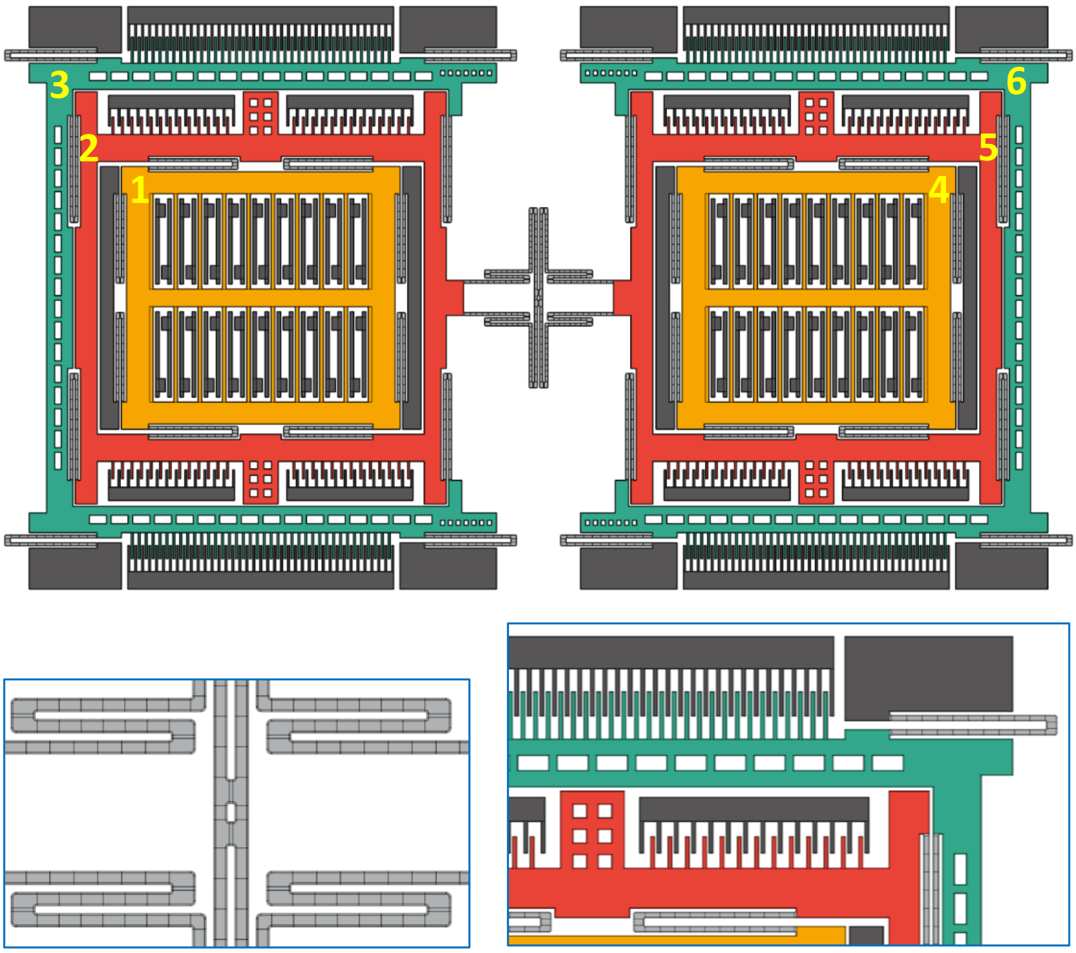


Fig. 27: Top view on the angular rates sensor, the coupling springs and the comb cells.

Default and user defined colors can be assigned to each seismic mass in the "Model Settings" window of the GUI. The coloring helps to clarify the operating principle in simulation models and result plots. Further annotation symbols can be activated in the "Model Tree" window. The colors are likewise transferred to the finite element tool ANSYS® or COMSOL® if a model export is activated by the "ANSYS Export" or "COMSOL Export" icon.

In Fig. 27, enlarged dimensions have been set for the spring width, electrode gap, corner fillet and process data. For subsequent simulations, realistic dimensions are applied as listed in the "Gyroscope1" project folder. Switch to the "Gyroscope1" folder in order to run the following examples.

Modal analysis and electrostatic softening effects

Simulations usually start with a "Modal analysis" to adjust the eigenfrequencies into a range of interest. In this example, the drive and sense frequencies should be between 20 kHz and 25 kHz whereby the sense frequency is often designed to be a few percent higher compared to the drive frequency. Finding appropriate dimension is a major challenge in conceptual design.

After some changes of geometrical parameters, the drive mode frequency has been set to 21.2 kHz and the sense mode frequency to 22.0 kHz. Obviously, the length of horizonal beams affect the frequency of the drive mode and the length of vertical beams the frequency of the sense mode in ux-direction.

A modal analysis of the mechanical domain shows that mode #1 and #2 are in-phase (common) modes and mode #3 and #4 are anti-phase operating modes (see Fig. 28). The frequency spacing between in-phase and anti-phase modes can be adjusted by the stiffness of the cross-shaped coupling spring at the center.

Frequently, angular rate sensors make use of "Mode-matching" principles of operation. "Mode-matching" means that both operating modes must have the same eigenfrequency in order to exploit resonance amplification for drive and sense motion at the same time.

Identical frequencies are difficult to realize because of manufacturing tolerances. To overcome the problem, the sense mode frequency is usually designed a few percent higher as the drive frequency. During operation, the sense mode is tuned down by electrostatic forces until the eigenfrequencies are nearly identical.

For demonstration, the SMU-combs are exploited for frequency tuning. The command sequence to calculate the tuned eigenfrequencies is listed in the right of Fig. 28. Mode-matching occurs at DC-voltage of about 10.4 V. In practice, separate tuning combs are usually attached to the system which are referred "FTU – Frequency Tuning Unit".

SOLV,modal;	COND,sen+,mass,v_sen-,v_sen+; COND,sen-,mass,v_sen-,v_sen+; VSCR,mass,0; VSCR,v_sen+,+10.4; VSCR,v_sen-,-10.4; SOLV,modal;
Mechanical eigenfrequencies: 1: 1.8628e+04 (in-phase sense mode) 2: 1.9896e+04 (in-phase drive mode) 3: 2.1248e+04 (anti-phase drive mode) 4: 2.2019e+04 (anti-phase sense mode) 5: 1.0074e+05 (anti-phase rx-tilt) 6: 1.0387e+05 (in-phase rx-tilt)	Tuned eigenfrequencies: 1: 1.7722e+04 2: 1.9896e+04 3: 2.1248e+04 4: 2.1249e+04 5: 1.0074e+05 6: 1.0387e+05
Highly enlarged anti-phase sense mode:	

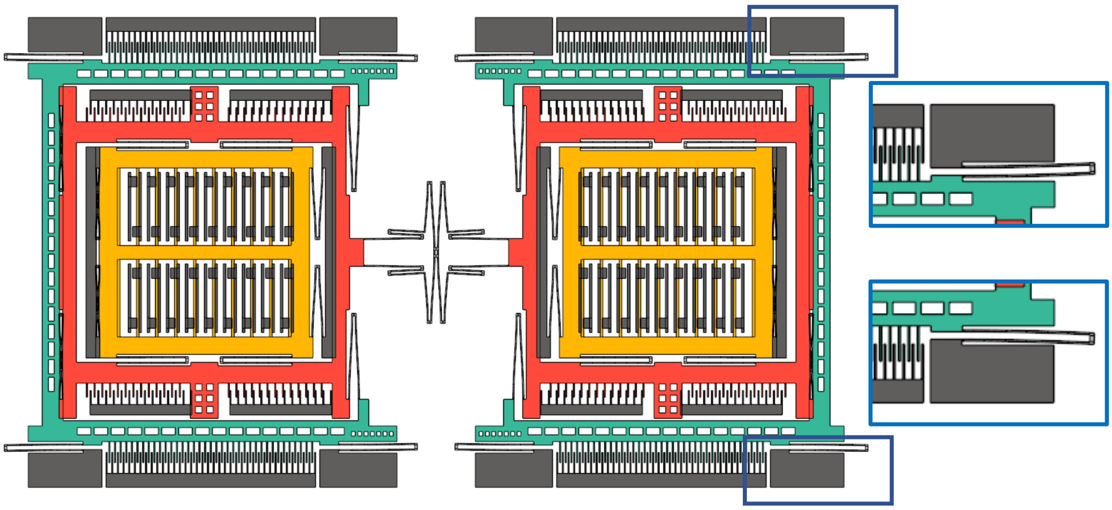


Fig. 28: Modal analysis results of the angular rate sensor with tuning effects.

Result plots in Fig. 28 indicate that “Drive frames” are not perfectly constraint in ux-direction. The horizontal beams in the blue boxes move a little up and down. It allows the green “Drive frames” to move slightly in sense direction and affects the sensor performance. It illustrates that numerical simulations are important to quantify imperfections of sensor and actuator components. A connecting bar between the ends of the upper and lower springs would strongly reduce this unwanted motion component and could improve the sensor performance.

The first four eigenfrequencies deviate about 2 % to full-order finite element simulations. Fig. 29 shows the finite element model which has been generated by ANSYS® export features to validate the accuracy (see Interface to ANSYS®).

High-order modes are designed to be about 5 times larger as the drive mode to avoid modal interactions due to non-linearities. From the physical point of view, even tiny non-linear effects stimulate harmonics (2f, 3f, ...) of the drive frequency. Disturbing resonant oscillations can occur if such a frequency is equal to a high order eigenmode. Therefore, the first harmonics of the drive mode should be compared carefully with other eigenfrequencies obtained in a modal analysis. The higher the frequencies the smaller are the amplitudes and consequently the disturbing effect.

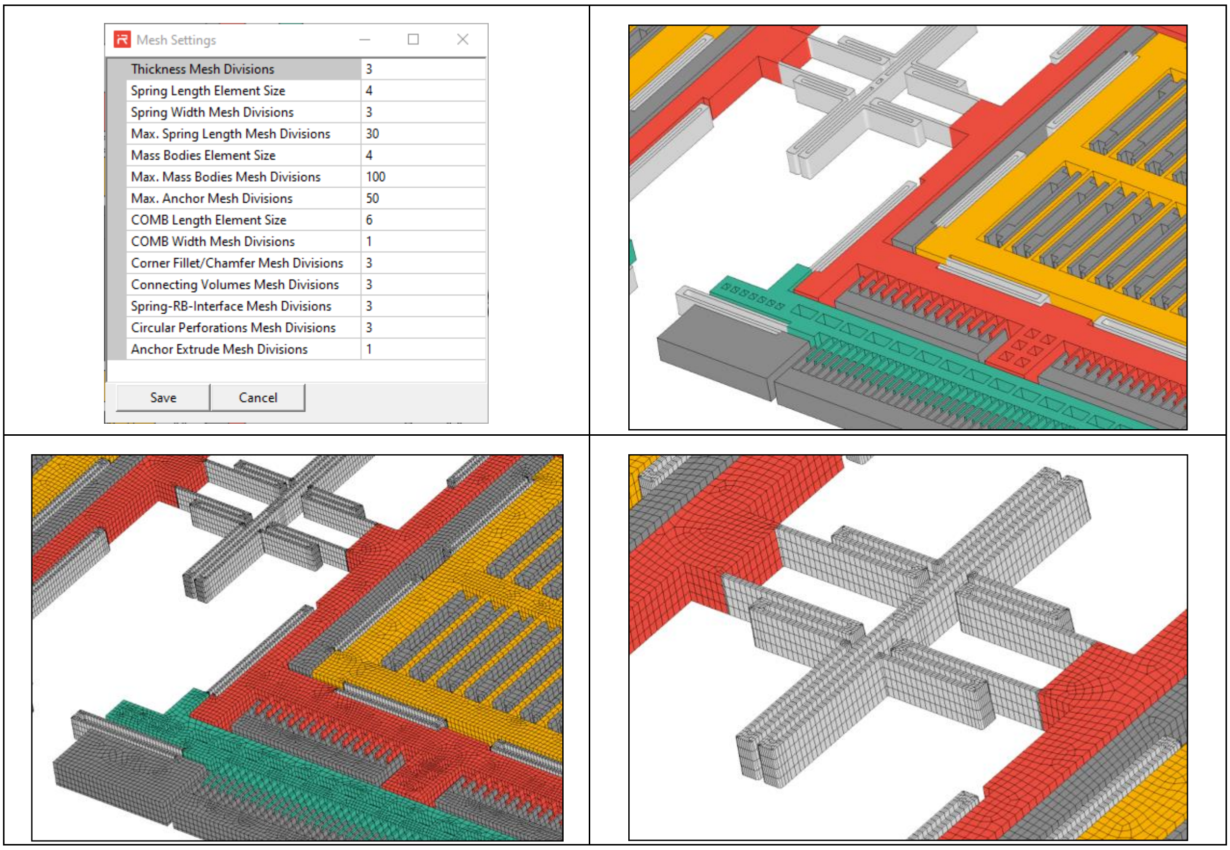


Fig. 29: Finite element model export of the angular rate sensor to ANSYS.

Static analysis and DC-sweep

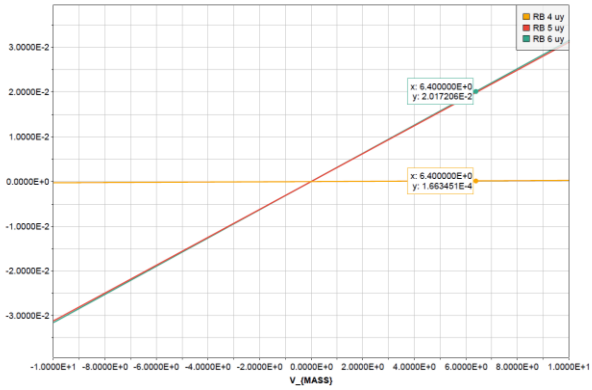
“Static simulations” are necessary to calculate primary sensor parameters such as the required drive voltages, the expected sense displacements and the related capacitance change for given angular rates. Furthermore, static simulations are helpful to estimate the sensitivity of the sensor to external accelerations, to vibrations or overload situations.

During operation, angular rate sensors oscillate continuously with a well-defined and permanently controlled amplitude in drive direction. Exemplarily, an amplitude of 4 µm should be reached in resonance. Assuming a quality factor of 200, it requires a quasi-static displacement amplitude of 20 nm at the “Drive frames”. To actuate the system, DC- and AC-voltages must be applied at the DAU-combs to create anti-phase drive oscillations.

The DC-voltages are set to ±20 V. The required AC-voltage is calculated by a sweep operation as shown in the right column of Fig. 30. A sweep range from -10 V to +10 V is specified by the DCSW command. About 6.4 V are necessary to reach 20 nm amplitudes at the “Drive frames” and “Coriolis frames” as shown by the green marker in the voltage-displacement-relationship. The contour plot below is scaled by a factor of 200 in order to visualize the drive amplitudes in resonance.

```
% Voltage for DAU
COND,drv+,mass,v_drv+;
COND,drv-,mass,v_drv-;
VSCR,v_drv+,-20;
VSCR,v_drv-,+20;
DCSW,vscr,mass,,-10,10,201,0;
SOLV,stat;
SNOD,rb,4,uy;
SNOD,rb,5,uy;
SNOD,rb,6,uy;

Drive voltages: DC:20 V, AC:6.4 V
Static results: UY = 20 nm
In resonance (Q=200): 4 µm
```

Voltage-displacement-relationship:



uy-displacements at 6.4 V (sample #165, 200 times enlarged):

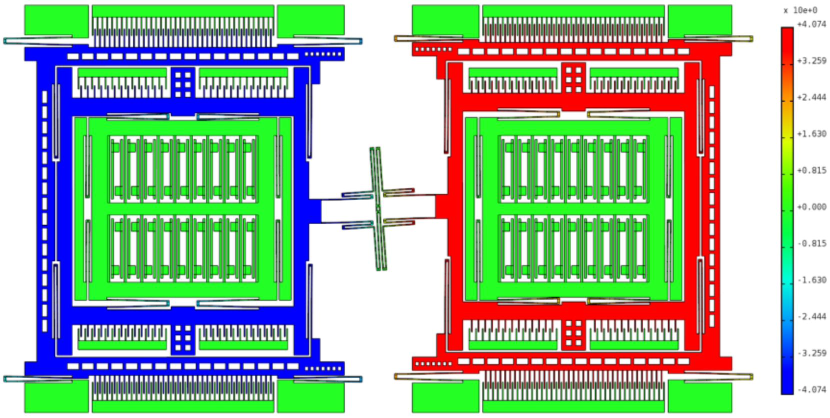


Fig. 30: Drive mode simulation results (voltage-displacement-relationship).

In a next simulation run, the sense mode amplitudes and the capacitance change shall be analyzed for a nominal angular rate of 20 rad/s. It corresponds to 1150 degrees per second or about 3.2 turns per second.

The DRIV command is utilized to assign 4 µm amplitudes to the drive eigen-mode. Next, the OMEG command defines the nominal angular rate around the z-axis and finally, “SOLV, stat” calculates the quasi-static displacement amplitudes. The “Sense frames” move about 0.62 nm. Assuming a quality factor of 100, the displacements become 62 nm in resonance.

```
% Coriolis forces
COND,sen+,mass,v_sen-,v_sen+;
COND,sen-,mass,v_sen-,v_sen+;
VSCR,mass,0;
VSCR,v_sen+,+10.4;
VSCR,v_sen-,-10.4;
DRIV,3,21.2e3,4,rb,6,uy;
OMEG,z,20;
SOLV,stat;
SNOD,rb,4,ux;
SNOD,rb,6,ux;
SCON,capa,mass,v_sen+;
SCON,capa,mass,v_sen-;
```

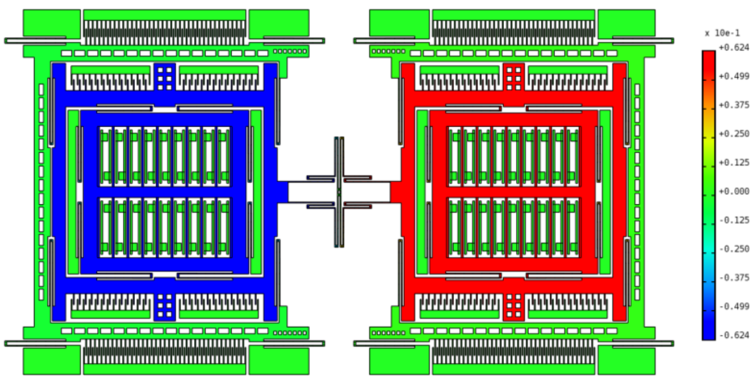


Fig. 31: Quasi-static sense mode simulation results for given angular rates.

The fixed fingers of the SMU-combs form a differential capacitor to transduce sense motion amplitudes into capacitance changes. It can be seen in Fig. 26, that the SMU-combs are mirrored around the y-axis to superimpose the capacitance change of anti-phase motion components. The comb fingers which are connected to the “V_SMU+” voltage ports are referred “SMU+” capacitances and the comb fingers connected to the “V_SMU-” ports are referred “SMU-” capacitances. The initial capacitances for both are about 270.8 fF.

In contrast to displacements, the capacitance changes obtained in quasi-static simulations can’t simply be multiplied by the quality factor to get results at resonance. Comb-cell capacitances based on gap variation scale non-linear with displacements.

For this reason, a second simulation run has been performed where the angular rate signal was multiplied with the quality factor. The resulting capacitance changes at 61.2 nm displacement amplitude become +7.40 fF at the “V_SMU+” and -7.00 fF at the “V_SMU-” ports. Both values are calculated for positive amplitudes and switch values for negative amplitudes of the sinusoidal oscillation. The average capacitance change is 7.20 fF.

Harmonic response analysis (AC-sweep)

While the static analysis covers only the low-frequency range, the “Harmonic analysis” precisely describes amplitudes and phase angles for all frequencies. In the following example, the results of the quasi-static simulations are compared to the harmonic response for the “Mode-matching” (a) and the “Mode-splitting” (b) approach.

Fig. 32 lists the command sequence for the harmonic response analysis. On the right is the amplitude-response for the drive mode (RB6 UY), the sense mode (RB4 UX) and the capacitance amplitudes at voltage ports “v_sen+” and “v_sen-” with regard to “mass”. Values for DC- and AC-voltages and the value for the angular rate are taken from the previous quasi-static simulation runs.

The modal superposition solver must be activated by the MSUP command because the quality factors for drive and sense strongly differ. A quality factor of 200 for the drive mode and 100 for the sense mode correspond to damping ratios of 0.0025 and 0.005 respectively. The quality factors for all other modes are set to 100. Damping ratios are assigned to modes by the MDMP command. Rayleigh damping defined by RDMP could fail because frequencies with strongly different damping ratios for drive and sense are very close together. Negative alpha-beta-multiplier might appear and can cause problems. Constant damping defined by CDMP does not allow different damping for drive and sense modes.

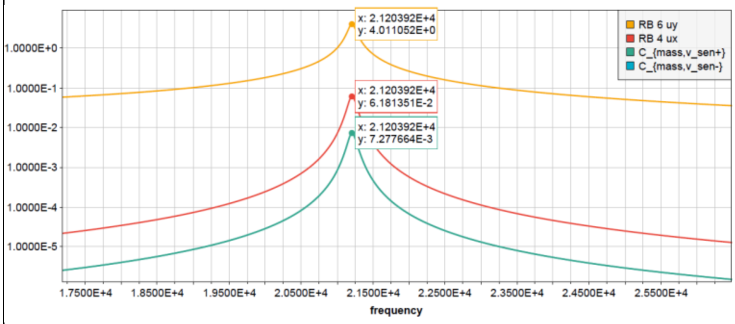
The lower bound of the frequency range is 10 kHz and the upper bound is 30 kHz. 5000 steps are calculated in between with a logarithmic data point spacing. Simulations are started by the “SOLV, harm” command and take just a few seconds.

The response for the “Mode-matching” approach agrees well with quasi-static results. The reason for small deviations between quasi-static and harmonic simulation results was already discussed for the accelerometer example.

The only difference to quasi-static simulation results is that the “SMU+” and “SMU-” capacitance functions in Fig. 32 show the same amplitudes. A harmonic response analysis linearizes all system equations at the operating point. Hence, capacitance amplitudes are calculated from the capacitance derivatives multiplied by displacement amplitudes. In a harmonic response analysis, the capacitance amplitudes for positive and negative displacement amplitudes are inherently the same.

```
% Harmonic analysis (a)
COND,drv+,mass,v_drv+;
COND,drv-,mass,v_drv-;
COND,sen+,mass,v_sen-,v_sen+;
COND,sen-,mass,v_sen-,v_sen+;
VSCR,mass,0;
VSCR,v_drv+,-20,6.4;
VSCR,v_drv-,+20,6.4;
VSCR,v_sen+,+10.4;
VSCR,v_sen-,-10.4;
MDMP,all,0.005;
MDMP,3,0.0025;
OMEG,z,20;
HARF,10e3,30e3,5000,log;
SOLV,harm;
SNOD,rb,6,uy;
SNOD,rb,4,ux;
SCON,capa,mass,v_sen+;
SCON,capa,mass,v_sen-;


b) no frequency tuning:
…
VSCR,v_sen+,+10.4/5;
VSCR,v_sen-,-10.4/5;
…
```

a) Mode-matching (frequency tuning):



b) Separated drive and sense mode (Mode-splitting):

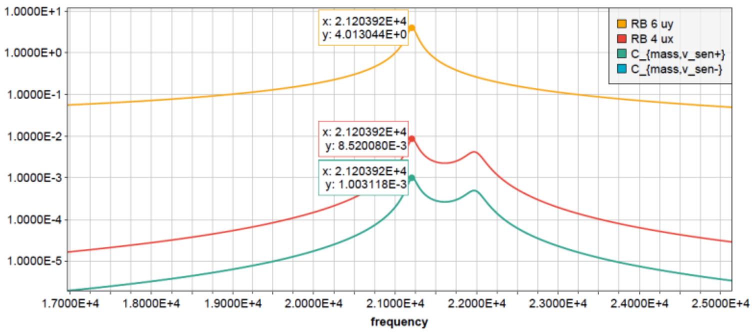


Fig. 32: Harmonic response analysis results with and without frequency tuning.

In the next simulation run, the tuning voltages have been reduced from 10.4 V to 2.08 V. Frequency tuning is strongly reduced and the drive and sense mode frequencies are separated by a few percent referred “Mode-splitting” approach. The harmonic response for the “Mode-splitting” condition is shown in case b). The sense mode shows two peaks, one at the drive frequency and one at the sense frequency.

At the marked drive frequency of 21.2 kHz, the sense mode is not in resonance and much smaller amplitudes and capacitance changes can be observed. As a result, the sensitivity of the gyroscope is much lower compared to case a).

“Mode-matching” allows for a very high sensitivity but usually the bandwidth of the sensor is small caused by high quality factors. The bandwidth can be enlarged if the response curve of the sense mode shows a high but also flat plateau. It is very difficult to realize in practice. In Fig. 33, the displacements are reduced from 62 nm to 13 nm (factor of 5) but the flat plateau spans now about 100 Hz. Angular rates with a frequency of about 50 Hz can be measured with high accuracy if the drive frequency is set at the center of the plateau.

```
COND,drv+,mass,v_drv+;
COND,drv-,mass,v_drv-;
COND,sen+,mass,v_sen-,v_sen+;
COND,sen-,mass,v_sen-,v_sen+;
VSCR,mass,0;
VSCR,v_drv+,-20,6.4;
VSCR,v_drv-,+20,6.4;
VSCR,v_sen+,+10.4/1.2;
VSCR,v_sen-,-10.4/1.2;
MDMP,all,0.005;
OMEG,z,20;
HARF,20.75e3,22.0e3,1000,lin;
SOLV,harm;
SNOD,rb,4,ux;
```

Sense amplitude for mode-matching (frequency tuning):

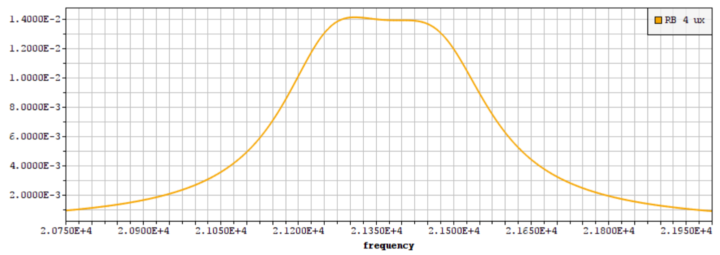


Fig. 33: Sense motion response showing a large amplitude and bandwidth.

Transient simulations

The harmonic response analysis calculates the stationary amplitude and phase angle of system quantities for given frequencies. “Transient simulations” in the time domain provide supplementary information about the system’s response but are more resource consuming.

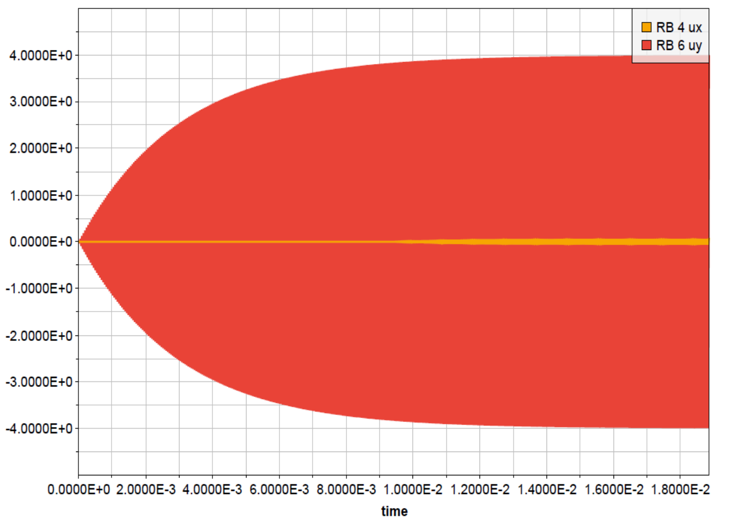
For sine-, step-, pulse-, or ramp-functions, the transient simulations calculate how long it takes until the stationary state is reached. For angular rate sensors, engineers are interested in the settling time of the drive and sense mode after load functions act on the system or change their values. For low damped systems, the settling time (95 % of the peak value) is roughly the quality factor divided by the eigenfrequency of the mode. For the given example, the drive mode takes about 10 ms and the sense mode about 5 ms to reach the stationary state. The following transient simulation reproduces the expected response.

In Fig. 34, the drive mode is stimulated at the DAU-ports with a sinusoidal voltage load at its eigenfrequency. Due to a quality factor of 200, the expected settling time is about 200 cycles. The angular rate function is defined by a pulse function. The high-pulse starts after 200 cycles (at about 9.5 ms) and it takes 100 cycles to reach the stationary state (at about 15 ms). The total simulation time is 400 cycles whereby 100 time-steps have been set for each period (see TIME command).

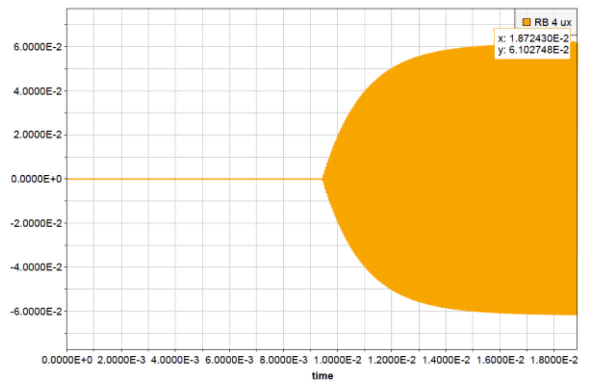
Stationary displacement amplitudes for drive and sense correspond to the results of the harmonic response analysis. The capacitance-time-functions at the SMU-combs show the expected asymmetry as discussed in the static simulation section. Transient simulations are solved iteratively and provide accurate results for non-linear effects. In this example, the initial SMU-capacitances are 270.8 fF, capacitances for positive sense amplitudes become 278,2 fF and for negative amplitudes 263.7 fF. Hence, the capacitance changes are +7.4 fF and -7.1 fF.

```
PARA,f0=21.249e+03;
COND,drv+,mass,v_drv+;
COND,drv-,mass,v_drv-;
COND,sen+,mass,v_sen-,v_sen+;
COND,sen-,mass,v_sen-,v_sen+;
VSCR,mass,0;
VSCR,v_drv+,-20,6.4,sine,f0,0;
VSCR,v_drv-,+20,6.4,sine,f0,0;
VSCR,v_sen+,+10.4;
VSCR,v_sen-,-10.4;
OMEG,z,0,+20,step,200/f0;
MDMP,all,0.005;
MDMP,3,0.0025;
TIME,400/(f0),1/(100*f0);
SOLV,trans;
SNOD,rb,6,uy;
SNOD,rb,4,ux;
SCON,capa,mass,v_sen+;
SCON,capa,mass,v_sen-;
SCON,curr,v_sen+;
SCON,curr,v_sen-;
```

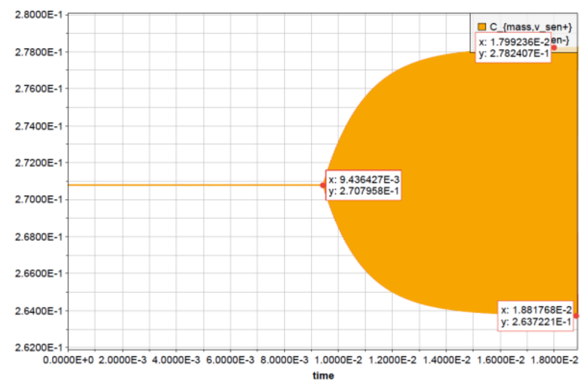
Drive and sense displacements over time:



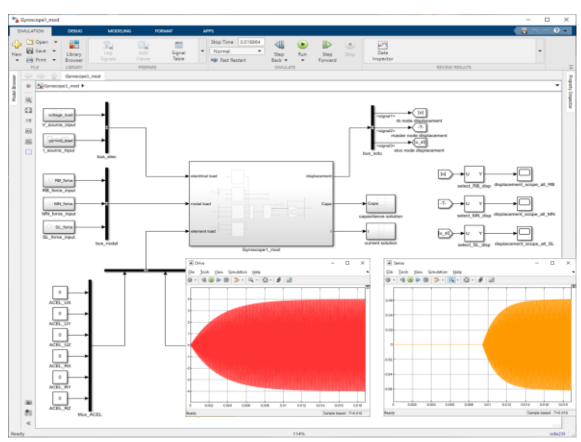
Sense displacements at mass body #4:



Capacitance-time-functions at SMU:



Simulink model export:



Sense displacements for mode-splitting:

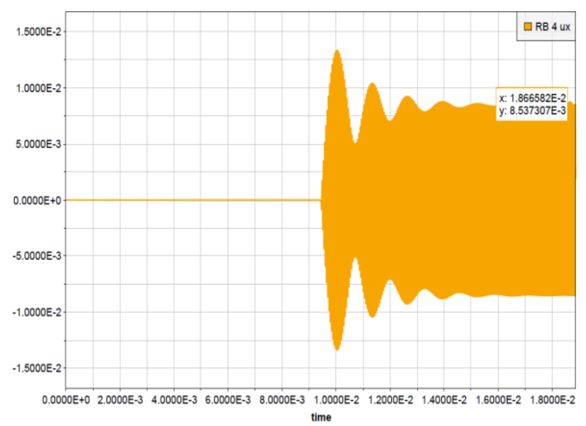


Fig. 34: Transient simulation results for mode-matching and mode-splitting.

In a next step, the transient simulation model has been exported to Simulink (see Inferface to SIMULINK®). All simulation settings are automatically transferred to the signal-flow models. The Simulink export is activated by clicking on the "Simulink Export" icon of the Graphical User Interface. Simulink simulation results are identical. Simulink models are more flexible to use. It is possible to apply arbitrary load functions, to modify system model features and to monitor internal simulation quantities as discussed earlier.

The right lower picture of Fig. 34 shows sense displacements for the "Mode-spitting" condition. In this graph, the SMU-voltages have been reduced from 10.4 V to 2.08 V. Amplitudes correspond to results of the harmonic response analysis. So far, the ideal system has been considered without imperfections.

In the next simulation run, a system with quadrature was modeled for demonstration. Open the model in the "Gyroscope2.irommod" in the same project folder.

For demonstration, the DAU-combs are misaligned with an offset of 10 nm by the "asym_g" parameter in the Design Variables window of the GUI. The fixed fingers of the upper DAU-combs are shifted 10 nm to the right, and the fixed fingers of the lower DAU-combs are shifted 10 nm to the left. At zero angular rates, the sense mode is slightly stimulated due to the asymmetry. In the given design, the misaligned DAU-combs create small electrostatic forces in sense direction depending on displacements in drive direction.

The unwanted sense motion components can be suppressed by electrostatic forces acting in opposite direction. In this example, a quadrature compensation voltage of 3.0 V has been applied at the "QCU-" combs shown in Fig. 26. The voltage at the "V_QUD-" ports create an anti-resonance at 20.61 kHz and cancels sense motion components in ux-direction. Commands for the harmonic response analysis and simulations results can be seen in Fig. 35. The higher the voltage has been set, the more moves the anti-resonance to lower frequencies.

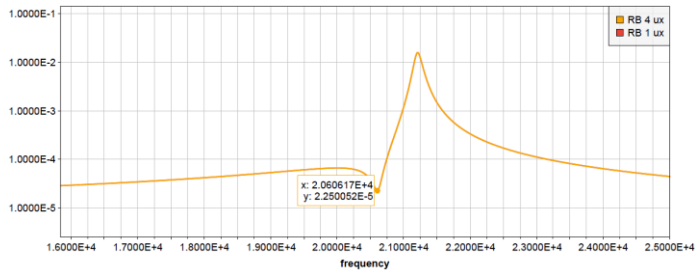
The frequency of the anti-resonance is the new operating point where quadrature signals are canceled out for a great part. Consequently, the drive frequency has be set at 20.61 kHz in subsequent transient simulations. In the first run, the "QCU-" voltage is set to zero in order to visualize the transient response without quadrature compensation. The sense signal amplitude due to quadrature is 0.28 nm and the signal with additional angular rate becomes 1.13 nm. Sense amplitudes are much smaller because both, the drive mode and the sense mode are not in resonance.

In the next transient simulation run, the “QCU-” voltage is lowly ramped from 0 V to 3.0 V. The sense motion component caused by quadrature is continuously reduced to about 0.028 nm. If the “QCU+” would be ramped by a DC-voltage, the sense motion amplitudes continuously increase, and the signal offset would worsen even more.

Quadrature compensation at resonance is more difficult to analyze because the phase shift must be considered too. More details can be found in literature.

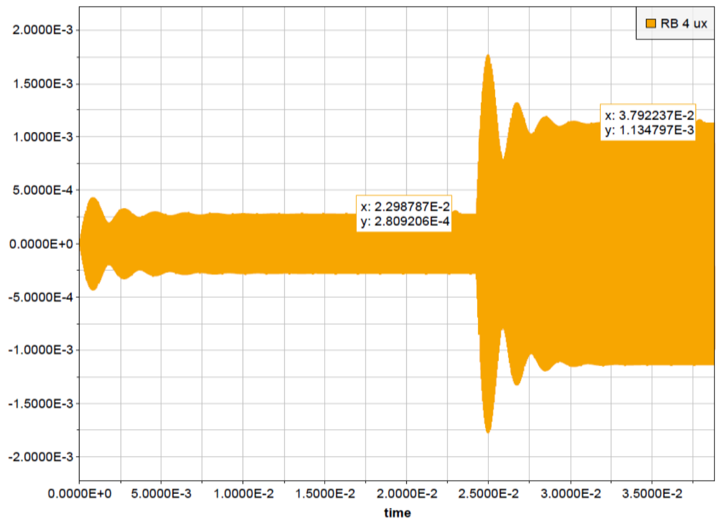
```
% Harmonic response analysis
COND,drv+,mass,v_drv+;
COND,drv-,mass,v_drv-;
COND,sen+,mass,v_sen-,v_sen+;
COND,sen-,mass,v_sen-,v_sen+;
COND,qud+,mass,v_qud+;
COND,qud-,mass,v_qud-;
VSCR,mass,0;
VSCR,v_drv+,-20,6.4;
VSCR,v_drv-,+20,6.4;
VSCR,v_sen+,+10.4;
VSCR,v_sen-,-10.4;
VSCR,v_qud+,0.0;
VSCR,v_qud-,3.0;
MDMP,all,0.005;
MDMP,3,0.0025;
HARF,15e3,25e3,1000,log;
SOLV,harm;
SNOD,rb,4,ux;
SNOD,rb,1,ux;
```

Harmonic response with quadrature compensation:




```
% Transient simulations
PARA,f0=20.61e+03;
COND,drv+,mass,v_drv+;
COND,drv-,mass,v_drv-;
COND,sen+,mass,v_sen-,v_sen+;
COND,sen-,mass,v_sen-,v_sen+;
COND,qud+,mass,v_qud+;
COND,qud-,mass,v_qud-;
VSCR,mass,0;
VSCR,v_drv+,-20,6.4,sine,f0,0;
VSCR,v_drv-,+20,6.4,sine,f0,0;
VSCR,v_sen+,+10.4;
VSCR,v_sen-,-10.4;
VSCR,v_qud+,0;
VSCR,v_qud-,0,3.0,ramp,500/f0;
MDMP,all,0.005;
MDMP,3,0.0025;
OMEG,z,0,+20,step,500/f0;
TIME,800/(f0),1/(50*f0);
SOLV,trans;
SNOD,rb,4,ux;
SNOD,rb,1,ux;
```

Transient response without quadrature compensation:



Transient response with quadrature compensation:

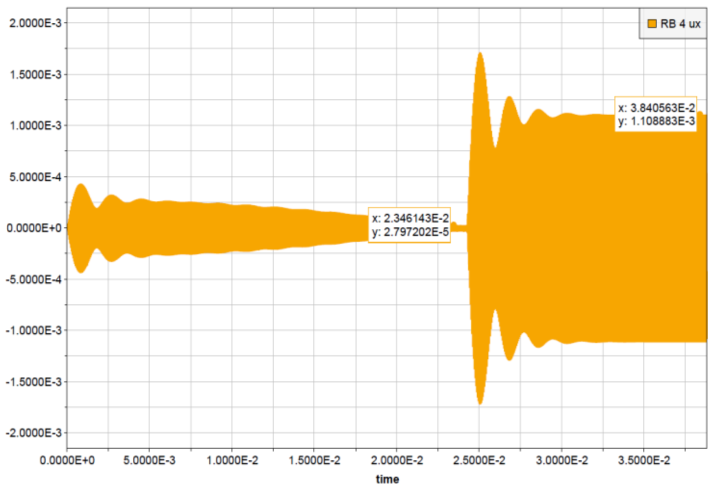


Fig. 35: Transient simulation schema to compensate quadrature effects.

3 Micro mirror actuator example

The next example demonstrates how to model and simulate a micro mirror actuator in the i-ROM MODELBUILDER environment. All model items are defined in the “Mirror.irommod” input file what can be found in the “Micro_mirror” project folder. Micro mirrors and micro mirror arrays are widely used for image projection systems, for spectrometer applications and optical object recognition or position tracing systems.

The micro mirror example shown in Fig. 36 is designed for resonant light deflection applications and can be used for the line-scan in image projection devices. The resonant frequency of the optical area is 24 kHz and the optical scan angle is about ±3.5°. The micro mirror is based on a two-mass system whereby the red mass body forms the “Drive Actuator Unit (DAU)” with two parallel plate capacitors underneath and the orange mass forms the optical area for light deflection.

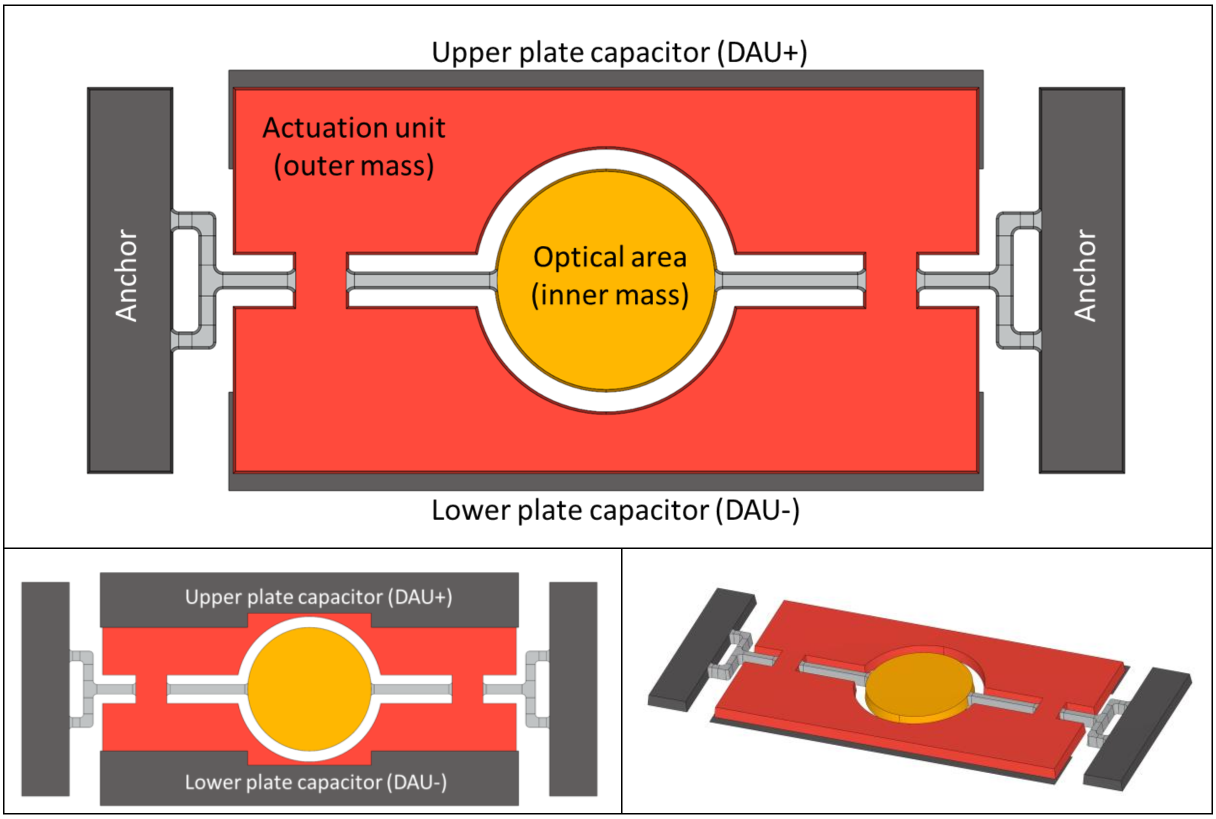


Fig. 36: Schematic view on the model components of the micro mirror actuator.

In order to achieve large scan frequencies, an anti-phase torsion mode is used for the given example. The first eigenmode of the micro mirror is the common mode whereby the inner and outer mass bodies tilt almost synchronously around the rx-axis. The second eigenmode is the anti-phase operating mode. Its frequency is about 4 times larger as the frequency of the common mode. The tilt amplitudes of the inner and outer masses are designed to be strongly different.

For the given dimensions, the tilt amplitude of the optical area is about 26 times larger as the tilt amplitude of the outer mass used for actuation. The ratio of the tilt amplitudes can be controlled by the mass moments of inertia of the moving masses.

A large amplitude ratio of the tilt angles is very important for the performance. A small tilt at the actuator mass allows for relatively narrow electrode gaps to the underlaying parallel plate capacitors. At narrow gaps, the applied voltages create large electrostatic forces which strongly actuate the anti-phase oscillation mode. In practice, the amplitudes of the drive mass can exploit almost the entire electrode gap without showing any pull-in effects since the system is driven in resonance. The small tilt amplitudes of the actuator mass couple through the inner springs to the orange mirror plate and create 26 times enlarged amplitudes. The amplitude ratio is defined by the rx-tilt DOF results of the anti-phase eigenvector. Underneath of the optical mirror must be a deep cavity.

A deep cavity is necessary for two reasons. First, to avoid a mechanical contact to the substrate surface and second to lower damping in order to obtain a significant resonant amplification of the anti-phase operating mode. In the present example, squeeze-film damping occurs mainly at moving plates with large amplitudes. For the anti-phase operating mode, it is the inner mass which contributes most to viscose damping. In that region, the gap to the substrate is large and a quality factor of 1000 is assigned to the anti-phase tilt mode in the following.

For the common mode, it is mainly the outer mass which has large amplitudes. Since the electrode gaps at the outer mass are much smaller, a quality factor of 100 has been assigned to the common mode. The quality factors of all other modes are likewise set to 100 for demonstration. Proper damping data can be obtained from squeeze-film simulations in finite-element tools or from analytical equations published in literature.

Model generation procedure

The actuator mass shown in red is defined by five primitives (see "Mirror.irommod" file). It starts with a rectangle defining the outer dimensions of the mirror plate. The second command cuts a circular hole at the center by Boolean subtraction. Finally, three rectangular primitives are used to cut the openings for the connecting springs. The inner mass is defined by a single circular primitive.

The suspension springs are modeled by 16 connecting points and 12 linking spring elements. The C-shaped connections at the anchors are valuable to reduce the influence of package stress on the micro mirror. Two master nodes are assigned at the actuator mass and two master nodes at the mirror mass as shown in Fig. 37. The master nodes are utilized to monitor displacements at the outer edges.

Both, the upper and the lower “Drive Actuator Unit (DAU)” are defined by three rectangular plate capacitors using the PCAP command. The “DAU+” and “DAU-” capacitances are automatically combined to a single capacitance since the same label has been set for all three plate elements.

Parallel plate capacitors can be larger as the moving masses. Furthermore, Boolean operations are utilized for complicated shapes of plate capacitors. The MODELBUILDER recognizes the overlapping area and considers in-plane and out-of-plane motion components for accurate capacitance calculations.

The right lower picture in Fig. 37 shows the exported finite element mesh of the mirror example (see Interface to ANSYS®).

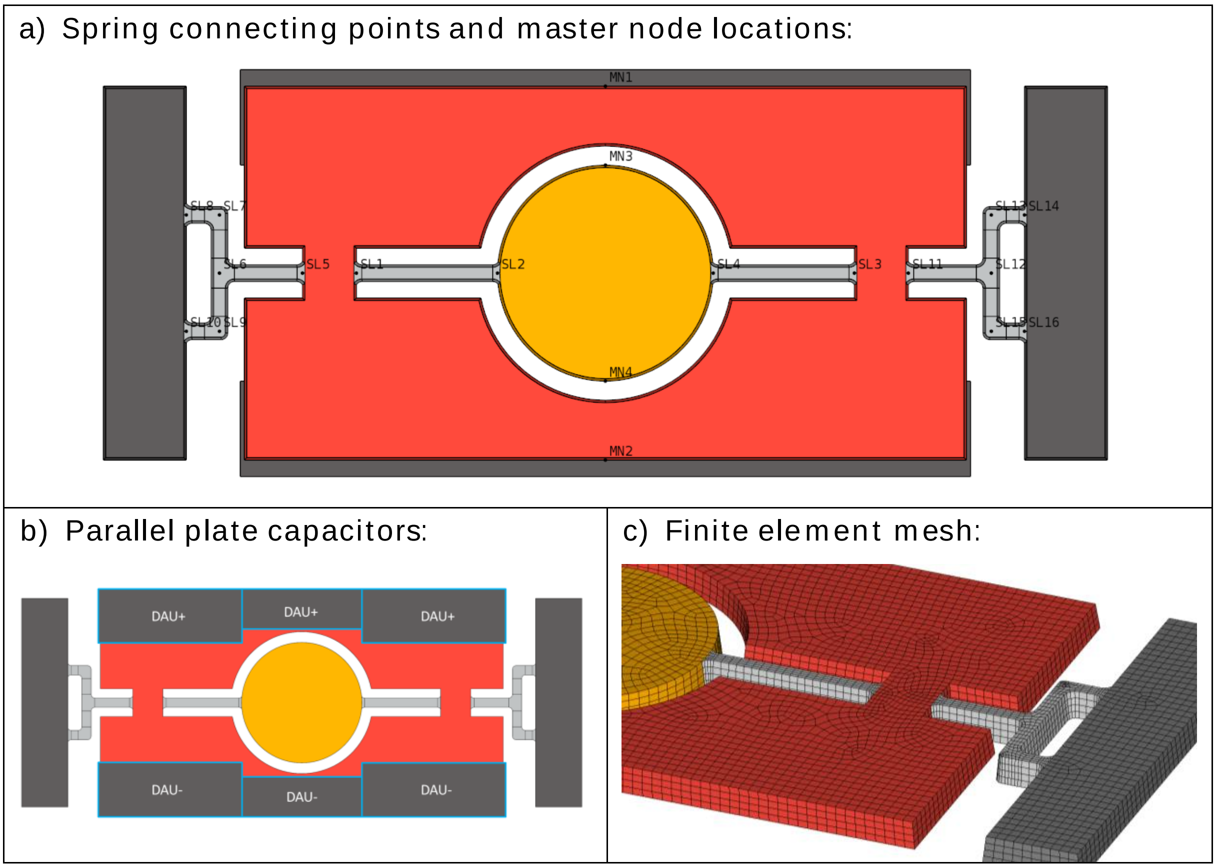


Fig. 37: Graphical visualization of characteristic model items and the FE-mesh.

Modal analysis and electrostatic softening effects

The “Modal analysis” calculates the eigenfrequencies and eigenvectors of the micro mirror example. In practice, a high DC-voltage is applied at the bottom plate capacitors and the mirror is driven by a superimposed AC-voltage applied at the functional layer with its masses. The DC-voltage is set +200 V at the “V_DAU+” port and -200 V at the “V_DAU-” port.

Fig. 38 shows the eigenfrequencies of the mechanical domain and the tuned frequencies at applied DC-voltages. It can be seen, that modes with out-of-plane motion components at the actuator mass change their eigenfrequencies. It is mainly mode #1 but also mode #4 has a slightly reduced eigenfrequency. Mode #1 is the common mode with a tuned frequency of about 5.4 kHz. The frequency shift is about -15 %. The anti-phase operating mode #2 has a frequency of 24 kHz.

The amplitude ratio for rx-tilt is about 26 and can be calculated from the eigenvectors of both mass bodies which are listed in the output plot of the “Solution Selection” window.

Mode #3 is the common mode in uy-direction. The colors in the contour plot show uz-displacements which are caused by the highly enlarged etch sidewall angles. The right lower picture shows mode #4 with out-of-plane motion components.

The geometrical dimensions have been adjusted to realize 24 kHz oscillation frequencies. If the deflected light beam of the mirror illuminates two lines in each cycle, one in the forward direction and one in backward direction, it leads to 48 thousand scanlines per second. Hence, images with 1000 lines can be projected with a frame rate of 48 picture per second.

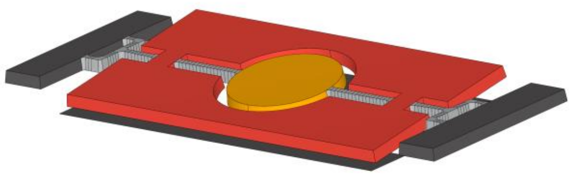
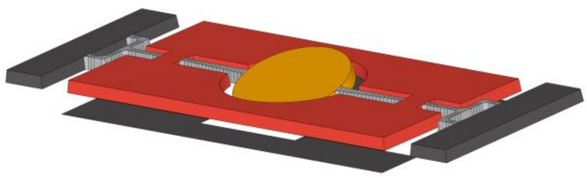
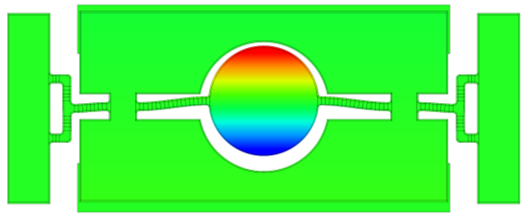
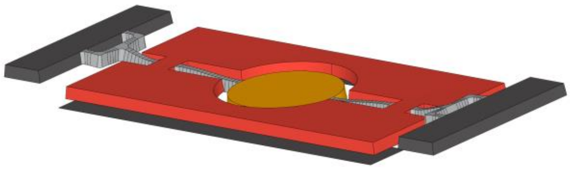
SOLV,modal;	COND,dau+,mass,v_dau+; COND,dau-,mass,v_dau-; VSCR,mass,0; VSCR,v_dau+,+200; VSCR,v_dau-,-200; SOLV,modal;
Mechanical eigenfrequencies: 1: 6.3550e+03 (common mode, rx-tilt) 2: 2.4036e+04 (anti-phase mode, rx-tilt) 3: 2.5341e+04 (common mode, uy-disp.) 4: 2.5818e+04 (common mode, uz-disp.) 5: 3.8836e+04 (anti-phase mode, uy-disp) 6: 4.3802e+04 (common mode, rz-tilt)	Tuned eigenfrequencies: 1: 5.4131e+03 2: 2.4027e+04 3: 2.5341e+04 4: 2.5717e+04 5: 3.8836e+04 6: 4.3802e+04
Mode #1, common mode, rx-tilt: 	Mode #2, anti-phase mode, rx-tilt: 
Mode #3 with uz-displacement contours: 	Mode #4, common mode, uz-displacements: 

Fig. 38: Modal analysis results with and without electrostatic softening effects.

Static analysis and DC-sweep

For micro mirror applications, “Static and DC-sweep simulations” are used to calculate the voltage-tilt-relationship. There are two different ways to drive the micro mirror during operation. In the simplest case, a voltage is applied at only one conductor as shown in the left column of Fig. 39.

Independent on the sign of the applied voltage at the “DAU+” port, the parallel plate capacitor creates attracting forces and the mirror tilts in just one direction. The “DAU-” port must be activated to tilt the mirror in the opposite direction what is costly in practice. Another drawback is that the voltage-tilt-relationship is a parabolic function even for small deflection angles. The voltage-tilt-relationship is non-linear and requires additional effort for controller circuits.

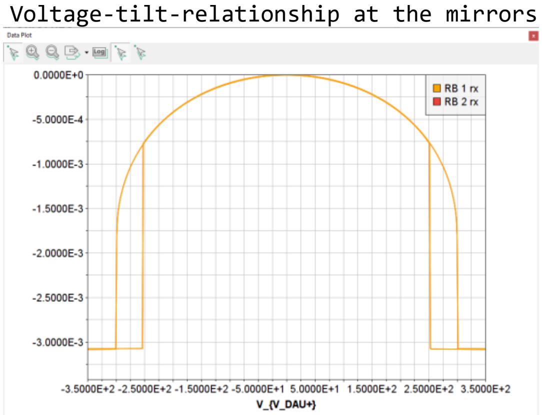
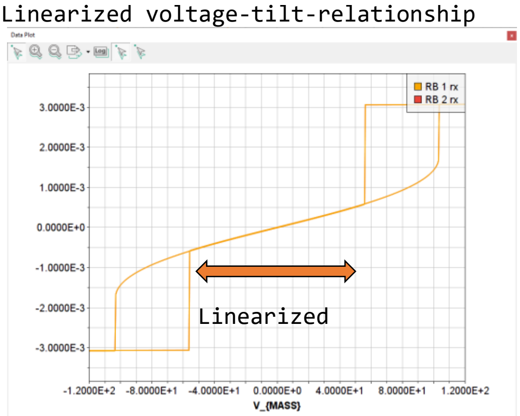
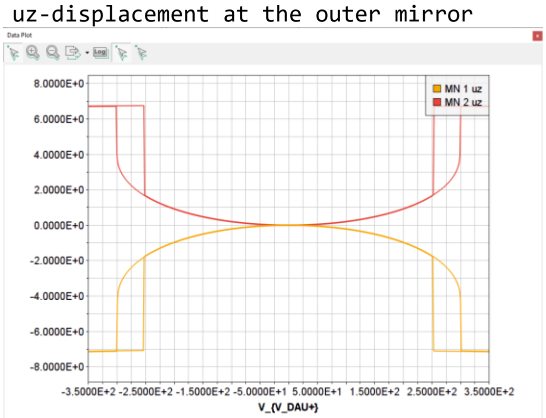
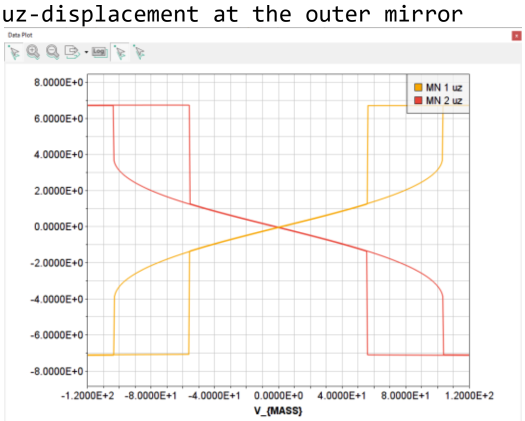
<table>
<tr>
<td>% Static voltage sweep
COND,dau+,mass,v_dau+;
VSCR,mass,0;
DCSW,vscr,v_dau+,,-350,350,501,1;
SOLV,stat;
SNOD,rb,1,rx;
SNOD,rb,2,rx;
SNOD,mn,1,uz;
SNOD,mn,2,uz;</td>
<td>% Linearized deflection range
COND,dau+,mass,v_dau+;
COND,dau-,mass,v_dau-;
VSCR,v_dau+,+200;
VSCR,v_dau-,-200;
DCSW,vscr,mass,,-120,120,501,1;
SOLV,stat;
SNOD,rb,1,rx;
SNOD,rb,2,rx;
SNOD,mn,1,uz;
SNOD,mn,2,uz;</td>
</tr>
<tr>
<td>% Voltage-tilt-relationship at the mirrors

</td>
<td>% Linearized voltage-tilt-relationship

</td>
</tr>
<tr>
<td>% uz-displacement at the outer mirror

</td>
<td>% uz-displacement at the outer mirror

</td>
</tr>
</table>

Fig. 39: Voltage-tilt-relationship of the mirror at different operating conditions.

Alternatively, both capacitors are utilized for electrostatic actuation as shown in the right column of Fig. 39. Two DC-voltages of different sign are applied at the bottom plate conductors referred bias voltages. The drive voltage is applied at the functional layer and forms a differential capacitor. DC-sweep results show that the voltage-tilt-function is linearized for driving voltages between -40 V and +40 V. Another advantage is that the mirror tilts in both directions depending on the sign of the applied voltage. The system is linearized in the operating range.

The voltage-tilt-functions in Fig. 39 are created by the “Data Plot” features of the GUI. Motion DOF results can be visualized at the rigid body center points (RB), at master nodes (MN) and spring connecting points (SL). Furthermore, DC-sweep results can be animated with the “3D View” panel icons. For above examples, an amplitude scale factor of 30 should be set in the “General Plot Control” window.

Static simulations analyze the quasi-static response at low frequencies. It is also possible to estimate the behavior at resonance for the first eigenmode by scaling data with the quality factor. However, the current example operates at the second eigenfrequency which can only be analyzed in a harmonic or transient simulation as shown in the following.

Harmonic response analysis (AC-sweep)

The “Harmonic response analysis” in Fig. 40 calculates tilt amplitudes and the phase angle in a frequency range from 1 kHz to 30 kHz with 5 thousand steps. It takes just a few seconds on a PC.

The right upper amplitude-frequency-plot shows the strong resonant amplification close to the first and second eigenmode. Linear axis scaling is usually not appropriate for high quality factors. The response curves below use logarithmic scaling for the y-axis. Logarithmic axis can be activated by the “LOG” icon in the “Data Plot” window of the Graphical User Interface. Clicking on the LOG-icon toggles between four combinations of the x-y-axes scaling (lin-lin, log-lin, lin-log and log-log referred as Bode plot).

The orange curve in Fig. 40 shows the frequency response of the optical mirror (RB 1, rx-DOF) and the red curve the response of the actuator mass (RB 2, rx-DOF). Tilt amplitudes at the eigenfrequency of 24 kHz are 31.3 mrad for the mirror mass and 1.20 mrad for the actuator mass. The mirror tilts ±1.79° mechanically and the optical scan angle is ±3.58°.

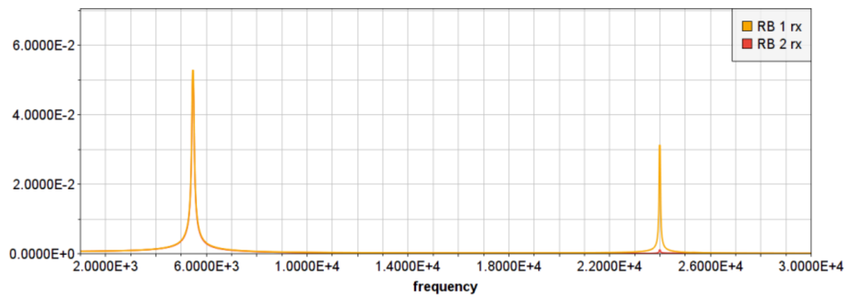
An anti-resonance can be observed at 23.58 kHz which cancels tilt-displacements at the actuator mass by about 60 dB (3 decades) compared to the tilt at the operating frequency. There are ideas to use anti-resonance for driving mirrors because the amplitude ratio between inner and outer mirror is extremely high.

Unfortunately, there is no resonance amplification and the overall performance is not as good as using an eigenmode for operation. The anti-resonance can be considered as the cut-off-frequency where the micro mirror turns from the in-phase oscillations (0° phase difference between rx-DOFs) to the anti-phase oscillations (180° phase shift) what can be seen in the phase-frequency-response of Fig. 40.

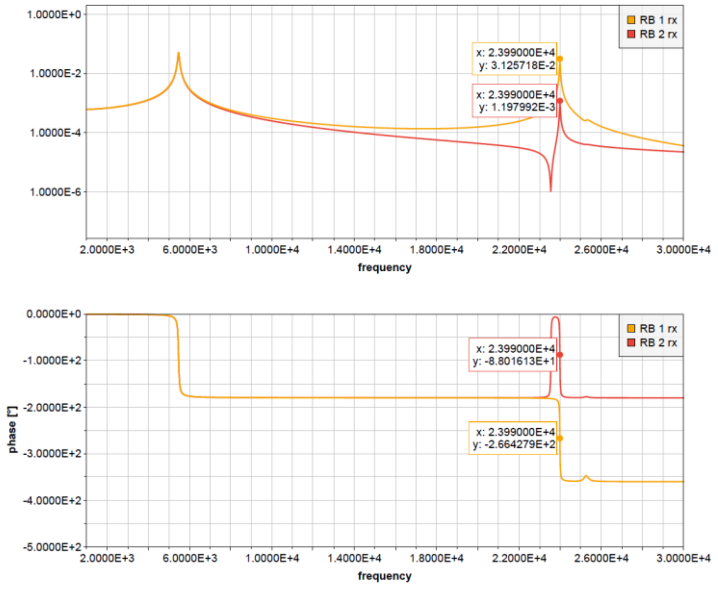
Displacements at the master nodes are shown in the right lower graph. The outer edge of the optical mirror has an amplitude of about 41 µm in uz-direction and the outer actuator mass moves 2.7 µm.

```
COND,dau+,mass,v_dau+;
COND,dau-,mass,v_dau-;
VSCR,mass,0,60;
VSCR,v_dau+,+200;
VSCR,v_dau-,-200;
MDMP,all,1/(2*100);
MDMP,2,1/(2*1000);
HARF,1e3,30e3,5000,lin;
SOLV,harm;
SNOD,rb,1,rx;
SNOD,rb,2,rx;
SNOD,mn,1,uz;
SNOD,mn,3,uz;
```

Tilt amplitudes at the inner and outer mass verse frequency:



Logarithmic plot of the tilt amplitudes:



Master node displacements:

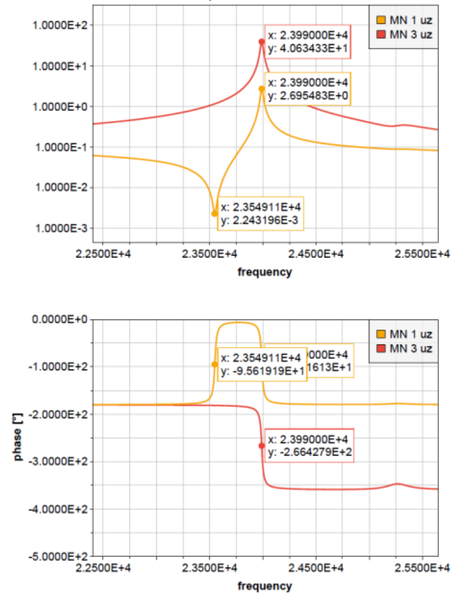


Fig. 40: Harmonic response analysis results of the micro mirror example.

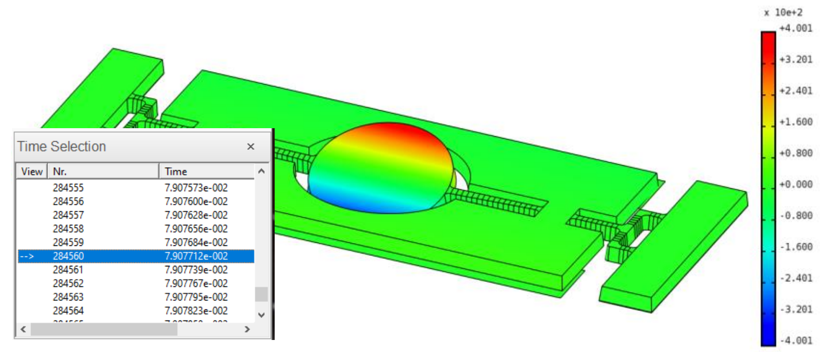
Transient simulations

“Transient simulations” analyze the structural response in the time domain. The example in Fig. 41 uses the same input parameters as set in the previous harmonic response analysis. The mirror is driven at 24 kHz by sinusoidal voltages and stationary state amplitudes are compared with AC-sweep result data.

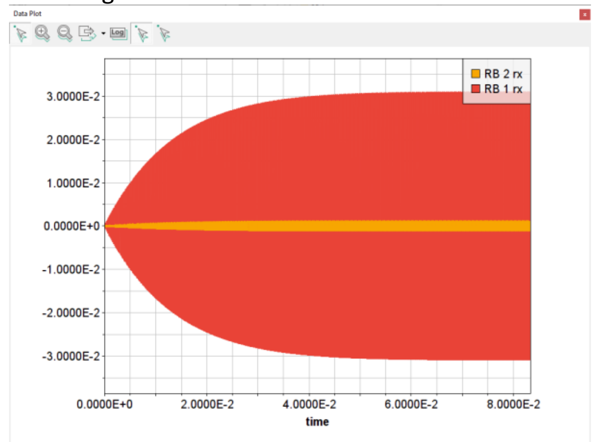
The quality factor of the operating mode is 1000. The settling time (95 % of peak value) is about 1000 cycles. In order to reach the stationary state, 2000 cycles will be analyzed in the following example. Hence, the total simulation time is 83 ms. Oscillators with high quality factors need smaller time steps compared to systems with moderate or optimal damping. For this reason, 150 steps are set for each cycle. It corresponds to a time-step size of 0.28 µs.

```
PARA,f0=24.03e+03;
COND,dau+,mass,v_dau+;
COND,dau-,mass,v_dau-;
VSCR,mass,0,60,sine,f0,0;
VSCR,v_dau+,+200;
VSCR,v_dau-,-200;
MDMP,all,1/(2*100);
MDMP,2,1/(2*1000);
TIME,2000/(f0),1/(150*f0);
SOLV,trans;
SNOD,rb,1,rx;
SNOD,rb,2,rx;
```

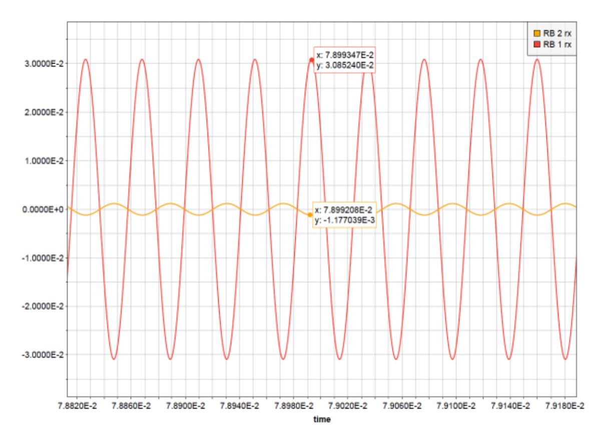
Displacements in uz-direction at the specified sample:



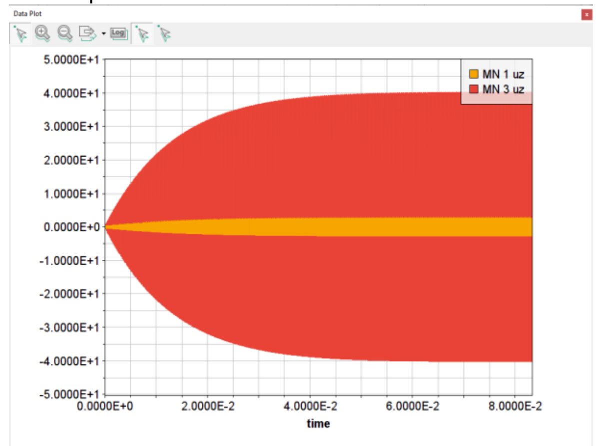
Tilt angle over time for both mass bodies:



Zoom into the transient response:



uz-displacements at master nodes over time:



Zoom into the transient response:

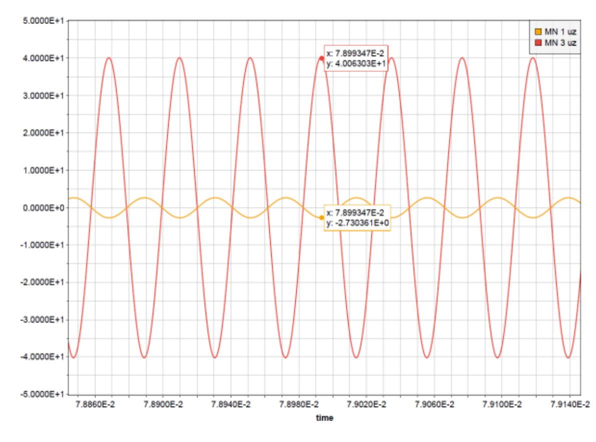


Fig. 41: Transient simulations of the settling time at sinusoidal voltage loads.

In total, 300 thousand time-steps are necessary to run the example in Fig. 41. It takes about 10 minutes on a PC. Exported Simulink models are usually faster because special solver provide an excellent performance for stiff differential equations which usually occur for MEMS applications (see Interface to Simulink®).

The second row in Fig. 41 shows the transient response for rx-tilt of the inner and outer mirror. Amplitudes are 30.8 mrad and 1.17 mrad what corresponds to results of the harmonic response analysis. Master node amplitudes are 40.1 µm and 2.73 µm at the outer edges of the mirror. The red and orange data point markers of the "*Data Plot*" panel show the DOF-label (RB 2 rx), the x-axis value what is time, the y-axis value what is the tilt angle in rad and the sample index.

Sample #284,560 with peak displacements has been selected in the "3D View" panel to show the structural displacements at a snapshot in time. The upper right picture of Fig. 41 shows uz-displacements at 79.07 ms which have been enlarged by a scale factor of 10 for better visualization.

Interesting is the time domain animation of the structural response. It takes several hundred cycles until the anti-phase motion can clearly being observed. The left plot in Fig. 42 shows the rx-tilt response for the inner and outer mass. At the beginning of the simulation are strong interference patterns which slowly disappear. Likewise, there are out-of-plane motion components caused by the DC-bias voltages which are switched on at the initial time-step. Vertical electrostatic forces pull the masses downwards and cause oscillations in uz-direction as shown in the right plot of Fig. 42.

In order to animate the transient response, select the first time-step in the "Time Selection" window, then set the scale factor of the "General Plot Controls" window to 1000 to enlarge motion components for better visualization and finally adjust the orientation of the mirror model to get the best view. Click on "Start animation" in the "3D View" window. After larger deflections have reached, the scale factor can be reduced accordingly to a final value of about 30.

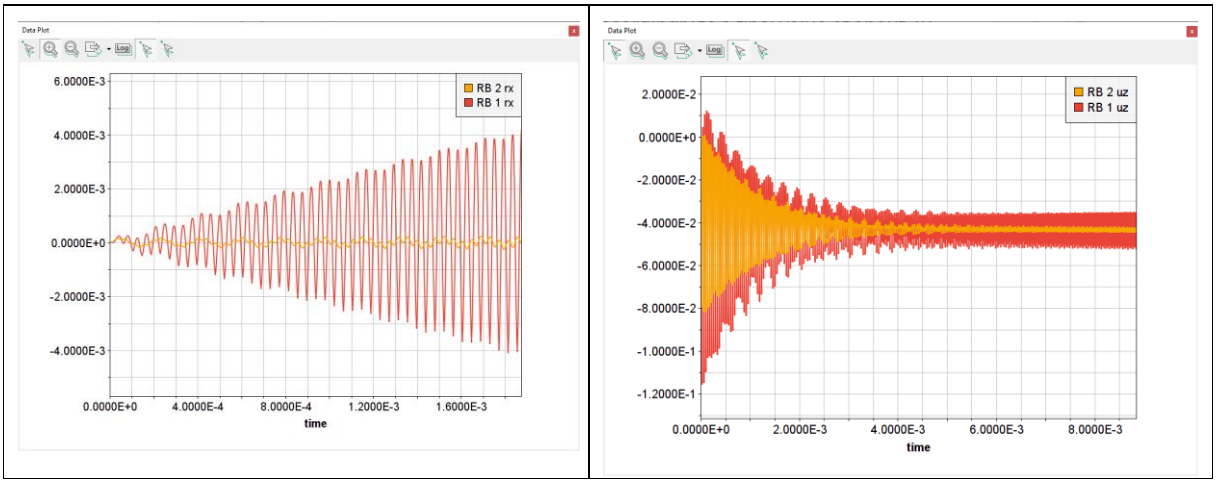


Fig. 42: Interferences of motion components at the beginning of the simulation.

Part E – Command Manual

1 Model file commands

General Syntax

The “User Model Input File” defines all model component of the MEMS devices to be analyzed. In particular, it defines the solid model items consisting of mass bodies with perforations and stopper elements, anchors, suspension springs, comb cells and plate capacitors. Additional commands are used to assign material properties and master node locations. These commands are referred “pre-processing” commands.

Other commands are “solution” and “post-processing” commands.

Solution commands are used to apply loads, displacement constraints, simulation settings and finally to execute the simulation run.

“Post-processing” commands are used to list, plot and evaluate simulation results.

Both, “solution” commands and “post-processing” commands can either be entered on the command line of the Graphical User Interface or the commands are added to “Load Case Files” which can be loaded in the Assign Loads and Constraints window.

Model file commands are defined in a text file with the following syntax:

- Each command consists of a “command name”, is followed by several “parameters” and ends with a semicolon.
- A “comment” starting with a %-sign can be written after the commands or in separate lines.
- Parameters can be defined by mathematical terms using the MATLAB-syntax (e. g. cos(x)).

```
% Mass body dimensions
PARA,body_x    = 90;          % Length parameter
PARA,anchor_s = 28;           % Anchor size

% Create anchors
RECT,,add,0,1,,+body_x/2,+body_x/2-anchor_s,-anchor_s/2,anchor_s/2;
RECT,,add,0,1,,-body_x/2,-body_x/2+anchor_s,-anchor_s/2,anchor_s/2;
```

*FOR-loops

```
*FOR, index, init_val, end_val, incr
*ENDFOR
```

*FOR-loops repeat a group of commands a specified number of times

- *FOR-loops can only be utilized in the user model input file. The command sequence to be processed starts with *FOR and ends with *ENDFOR.
- *FOR-loops can be nested as shown in the following example.

```
*FOR,x_sym,1,-1,-2
  *FOR,y_sym,1,-1,-2
    PARA, x_loc=x_sym*(body_x2/2-anchor_s/2);
    STOP,,circ,1,1,,x_loc,y_sym*body_y1/2,x_loc,y_sym*anchor_s/2,4;
  *ENDFOR
*ENDFOR
```

Input Data

Parameter	Description
index	Loop index - Alphanumeric name, has to start with a letter
init_val	Initial value of the loop index - Real or integer
end_val	Ends the loop if index is greater than end_val - Real or integer
incr	Index increment, can be positive or negative - If empty → Increment is set to 1 - Real or integer (≠0)

*IF-conditions

```
*IF, value1, oper, value2
*ELSEIF, value1, oper, value2
*ELSE
*ENDIF
```

*IF-conditions process a group of commands conditionally

- *IF-conditions can be utilized to process a group of commands in the user model input file conditionally.
- The command sequence to be processed starts with *IF, *ELSEIF or *ELSE depending on the condition and ends with *ENDIF or *ELSE. *IF-conditions can be nested.

```
*IF,x_sym,eq,0
   SPRI,1+offs,3+offs,,,spring_w,,0;
*ELSE
   SPRI,1+offs,3+offs,,,spring_w,,,0;
*ENDIF
```

Input Data

Parameter	Description
val1	First numerical value to be compared - Real, integer or a mathematical term in MATLAB-syntax
oper	Operation label for the Boolean condition ▪ eq → Equal ▪ ne → Not equal ▪ lt → Less than ▪ le → Less than or equal ▪ ge → Greater than or equal ▪ gt → Greater than
val2	Second numerical value to be compared - Real, integer or a mathematical term in MATLAB-syntax

EOF

```
EOF;
```

Terminates the execution of the user model input file

2 Pre-processing commands

PARA

PARA, `param1=value1, param2=value2, param3=value3;`

Creates user defined model parameters

- One or more parameters can be defined by a single **PARA**-command.

Input Data

Parameter	Description
`param1 | param2 | ...`	Parameter name - Alphanumeric name, has to start with a letter
`value1 | value2 | ...`	Value of the parameter or a mathematical term in MATLAB-syntax (see General Syntax), character values are not supported

LAYR

LAYR, `layer_nr, bot_loc, top_loc;`

Defines the vertical location of functional layers

- The layer command defines the z-coordinates of the bottom and top faces of functional layers. The functional layer contains mass bodies, anchors, springs, combs and stopper elements. The current release supports one functional layer. The layer number must be set to 1.
- The anchor extension to the substrate surface is specified by "Anchor extrude length" defined in the Simulation Settings window.
- Further layers are used to define the vertical location for top and bottom plate capacitors (see PCAP Command). The coordinates `bot_loc` and `top_loc` can be identical.

Input Data

Parameter	Description
`layer_nr`	Layer number (integer) - If 1, defines the location of the functional layer - If > 1, defines the location of parallel plates
`bot_loc`	Defines the z-coordinate of the bottom face
`top_loc`	Defines the z-coordinate of the top face

SKET

`SKET, par_type, par_name, value1, value2, value3;`

Defines settings for the SKETCHER interface

- The **SKET** commands assign grid settings and group reference names.
- All Sketcher items with the same body, material and layer reference number belong to the same group. A group reference name can be assigned to make the model structure more understandable. Reference names of individual components are directly assigned with other pre-processor commands.
- **SKET** settings are model-specific and support the data exchange with the SKETCHER interface. Default values are applied if no **SKET** command is defined.
- Default grid settings can be set in `./settings/init-file.json`.

Input Data

Parameter	Description
par_type	Sketcher parameter type to be assigned → grid = Assign grid settings → group = Assign group reference names
par_name	If par_type=grid: → g_size = Assign grid size → g_snap = Activates/deactivates grid snap → a_size = Assign angle size → a_snap = Activates/deactivates angle snap If par_type=group: → Alphanumeric name, has to start with a letter
value1	f par_type=grid: → g_size = Grid size value (double) → g_snap = 1 grid ON, 0 grid OFF → a_size = Angle grid size value in degree (double) → a_snap = 1 grid ON, 0 grid OFF If par_type=group: → Body reference number of the group
value2	If par_type=grid: unused If par_type=group: → Material reference number of the group

value3	If par_type=grid: unused If par_type=group: → Layer reference number of the group

Example of typical SKET commands:

```
SKET, grid,     g_size,10;
SKET, grid,     g_snap, 1;
SKET, grid,     a_size, 5;
SKET, grid,     a_snap, 1;
SKET, group,   anchors, 0,1,1;
SKET, group,mass_body, 1,1,1;
```

MATP

MATP, mat_nr, type, value;

Assigns material properties to material reference numbers

- Model items such as primitives of masses or anchors, springs and moving fingers of comb cells are related to a material reference number.
- Material reference number 1 is automatically assigned to springs but can be used for other model items as well.
- Isotropic or orthotropic elastic properties are considered for suspension springs.
- Isotropic elastic properties are defined by ex and nuxy parameters only.
- Mass bodies are considered rigid and take just density values into account.
- The ance flag defines which areas of the anchors should reach the substrate surface. Anchors are defined by a zero body reference number (see RECT, CIRC, TRIA or POLY command). Anchor primitives must be assigned different material reference numbers if only parts of the anchor region should be extended to the substrate surface. The material properties can be the same. If the ance flag is set to true, the associated anchor primitives are extended to the surface, otherwise not.
- Materials used inside of other material areas must have a higher material reference number.

Input Data

Parameter	Description
mat_nr	Material reference number - > 0 - Empty if type = esys - Empty if type = eps
type	Material parameter type: - name → Material reference name - ance → Anchor extension (true, false) - dens → Density - ex → Elastic moduli x-direction - ey → Elastic moduli y-direction - ez → Elastic moduli z-direction - nuxy → Minor Poisson's ratio - nuxz - nuyz - gxy → Shear moduli - gxz - gyz - eps → Permittivity - esys → Orientation angle of the Material Coordinate System (MCS) in degrees
value	**name** → Alphanumeric name, has to start with a letter- ance → 1 or true, Anchor extension (default) → 0 or false, No anchor extension Otherwise: Value of the material parameter

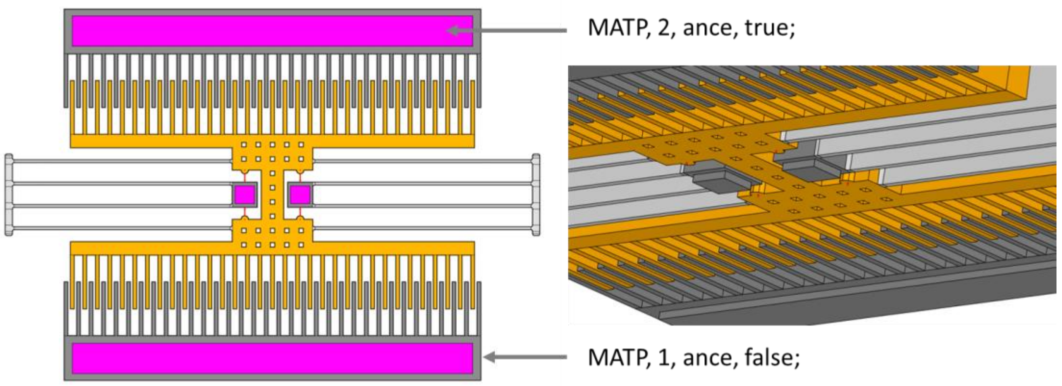


Fig. 43: Modeling of anchor extension areas

RECT

```
RECT, rect_name, b_opt, body_nr, mat_nr, layer_nr, x, y, length,
height, orient_ang;
```

Creates a rectangular primitive for mass bodies or anchors

- The rectangular primitive is defined by a connecting point, length in x-direction and height in y-direction. The orientation angle can be used to rotate the rectangle around the connecting point.
- Mass bodies and anchors may be assembled from several primitives with different material reference numbers.

Input Data

Parameter	Description
rect_name	Reference name of the primitive (can be empty) ➔ Alphanumeric name, has to start with a letter
b_opt	Type of the Boolean operation - add → Add a primitive to the mass body or anchor - sub → Subtract a primitive from a mass body or anchor
body_nr	Reference number of related mass body or anchor - If 0 → The primitive is related to anchors - If > 0 → The primitive is related to the specified mass body
mat_nr	Reference number of the material properties - Use existing material properties only (see MATP command)
layer_nr	Layer number (the current version supports only one layer) - If empty, the primitive is part of the first layer
x	X-coordinate of the connecting point for rectangular primitives
y	Y-coordinate of the connecting point for rectangular primitives
length	Length (x-direction) of the rectangular primitive (can be negative)
height	Height (y-direction) of the rectangular primitive (can be negative)

orient_ang	Orientation angle in degrees at the connecting point: 0 ≤ orient_ang < 360

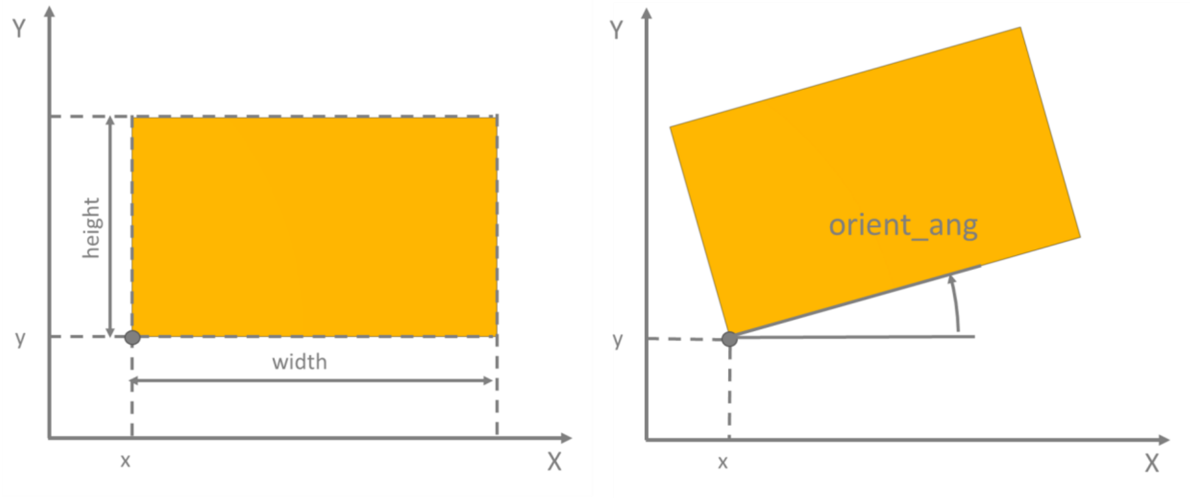


Fig. 44: Dimensional parameters of a rectangular primitive

CIRC

```
CIRC, circ_name, b_opt, body_nr, mat_nr, layer_nr, xm, ym, r1, r2,
  phi1, phi2, orient_ang;
```

Creates a circular primitive for mass bodies or anchors

- The primitive can be a circular area, a ring with inner and outer radiuses or a sector of both defined by the starting and span angle.

Input Data

Parameter	Description
circ_name	Reference name of the primitive (can be empty) ➔ Alphanumeric name, has to start with a letter
b_opt	Type of the Boolean operation - add → Add a primitive to the mass body or anchor - sub → Subtract a primitive from a mass body or anchor
body_nr	Reference number of related mass body or anchor - If 0 → The primitive is related to anchors - If > 0 → The primitive is related to the specified mass body
mat_nr	Reference number of the material properties - Use existing material properties only (MATP commands)

`layer_nr`	Layer number (the current version supports only one layer) - If empty, the primitive is part of the first layer
`xm`	X-coordinate of the center point of the circular primitive
`ym`	y-coordinate of the center point of the circular primitive
`r1`	First radius of circular primitive
`r2`	Second radius of circular primitive - If empty or `r2` = 0, it defines a solid circular primitive
`phi1`	Starting angle of the circular segment, 0 ≤ `phi1` < 360
`phi2`	Span angle of the circular segment, 0 < `phi2` ≤ 360
`orient_ang`	Orientation angle in degrees at the connecting point: 0 ≤ `orient_ang` < 360

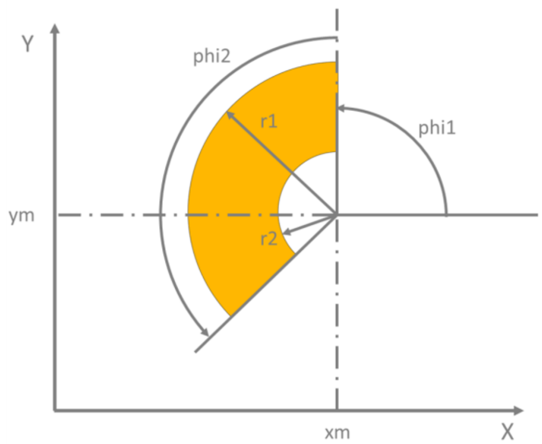


Fig. 45: Dimensional parameters of a circular primitive

ELPS

```
ELPS, elps_name, b_opt, body_nr, mat_nr, layer_nr, xm, ym, r_x, r_y,
   phi1, phi2, orient_ang;
```

Creates an elliptical primitive for mass bodies or anchors

- The elliptical primitive is defined by the center point coordinates, the radiuses in x- and y-direction and the start- and span angles.
- The orientation angle can be used to rotate an elliptical primitive around its connecting point at the center.

- Mass bodies and anchors may be assembled from several primitives with different material reference numbers.

Input Data

Parameter	Description
`elps_name`	Reference name of the primitive (can be empty) ➔ Alphanumeric name, has to start with a letter
`b_opt`	Type of the Boolean operation - add → Add a primitive to the mass body or anchor - sub → Subtract a primitive from a mass body or anchor
`body_nr`	Reference number of related mass body or anchor - If 0 → The primitive is related to anchors - If > 0 → The primitive is related to the specified mass body
`mat_nr`	Reference number of the material properties - Use existing material properties only (MATP commands)
`layer_nr`	Layer number (the current version supports only one layer) - If empty, the primitive is part of the first layer
`xm`	X-coordinate of the center point of the elliptical primitive
`ym`	y-coordinate of the center point of the elliptical primitive
`r_x`	Radius in x-direction of the non-rotated ellipse
`r_y`	Radius in y-direction of the non-rotated ellipse
`phi1`	Starting angle of the elliptical segment, 0 ≤ phi1 < 360
`phi2`	Span angle of the elliptical segment, 0 < phi2 ≤ 360
`orient_ang`	Orientation angle in degrees at the elliptical primitive: 0 ≤ `orient_ang` < 360

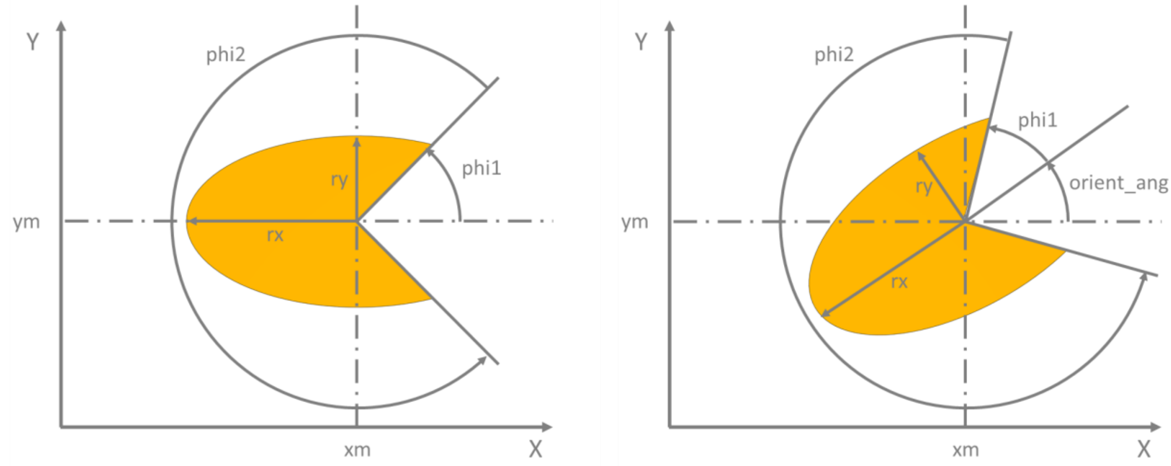


Fig. 46: Dimensional parameters of an elliptical primitive (non-rotated an rotated)

TRIA

```
TRIA, tria_name, b_opt, body_nr, mat_nr, layer_nr, x1, y1, x2, y2,
x3, y3, orient_ang;
```

Creates a triangular primitive for mass bodies or anchors

Input Data

Parameter	Description
tria_name	Reference name of the primitive (can be empty) ➔ Alphanumeric name, has to start with a letter
b_opt	Type of the Boolean operation - add → Add a primitive to the mass body or anchor - sub → Subtract a primitive from a mass body or anchor
body_nr	Reference number of related mass body or anchor - If 0 → The primitive is related to anchors - If > 0 → The primitive is related to the specified mass body
mat_nr	Reference number of the material properties - Use existing material properties only (see MATP command)
layer_nr	Layer number (the current version supports only one layer) - If empty, the primitive is part of the first layer
x1	X-coordinate of the first point (connecting point)

y1	Y-coordinate of the first point (connecting point)
x2	Second x-coordinate of the primitive
y2	Second y-coordinate of the primitive
x3	Third x-coordinate of the primitive
y3	Third y-coordinate of the primitive
orient_ang	Orientation angle in degrees at the connecting point: $0 \leq$ orient_ang < 360

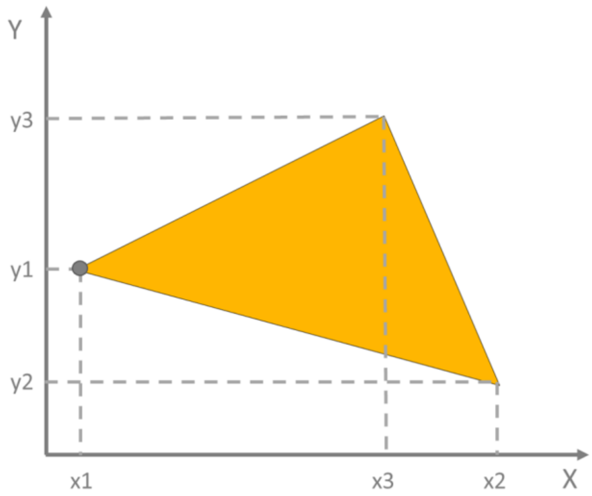


Fig. 47: Dimensional parameters of a triangular primitive

POLY

```
POLY, poly_name, b_opt, body_nr, mat_nr, layer_nr, x1, y1, x2, y2,
   ... xn, yn, orient_ang;
```

Creates a polygonal primitive for mass bodies or anchors

Input Data

Parameter	Description
poly_name	Reference name of the primitive (can be empty) ➔ Alphanumeric name, has to start with a letter
b_opt	Type of the Boolean operation - add → Add a primitive to the mass body or anchor

	- sub → Subtract a primitive from a mass body or anchor
body_nr	Reference number of related mass body or anchor - If 0 → The primitive is related to anchors - If > 0 → The primitive is related to the specified mass body
mat_nr	Reference number of the material properties Use existing material properties only (see MATP commands)
layer_nr	Layer number (the current version supports only one layer) - If empty, the primitive is part of the first layer
x1	First x-coordinate of the primitive
y1	First y-coordinate of the primitive
x2	Second x-coordinate of the primitive
y2	Second y-coordinate of the primitive
...	Further pairs of x-y-coordinates
xn	n-th x-coordinate of the primitive
yn	n-th y-coordinate of the primitive
orient_ang	Orientation angle in degrees at the connecting point: 0 ≤ orient_ang < 360

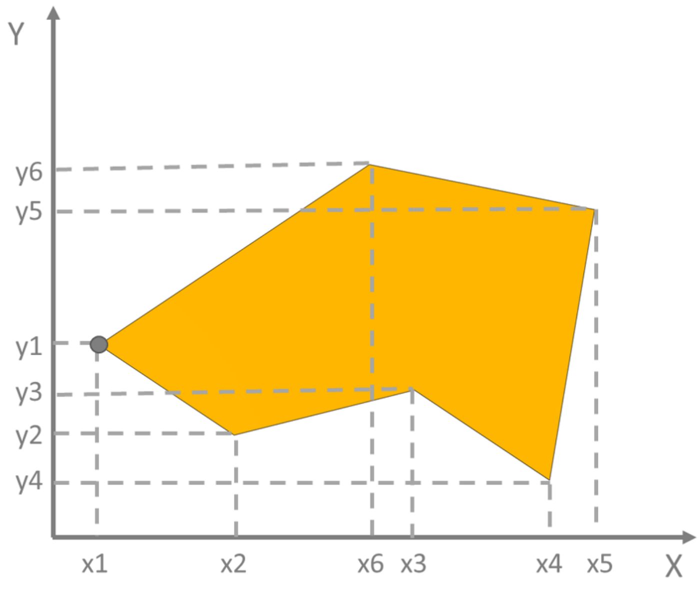


Fig. 48: Dimensional parameters of a polygonal primitive

UDAR

```
*UDAR, udar_name, b_opt, body_nr, mat_nr, layer_nr, x, y, ori-
ent_ang;
```

User defined area for primitives of mass bodies and anchors

- User defined areas are defined within an ***UDAR** and ***ENDUDAR** command sequence.
- The ***UDAR** command defines primitive settings, the start point coordinate and the orientation angle.
- If follow a series of commands for the outer lines which are either:
  - Straight lines (see **LINE** command),
  - Circular arcs (see **CARC** command) or
  - Bezier curves (see **BCUR** command).
- User defined areas are closed by the ***ENDUDAR** command. The command creates a straight line to the starting point except the endpoint of the last line is identical to the start point.
- ***FOR**-loops and ***IF**-conditions are supported as shown in the following example.

Input Data

Parameter	Description
udar_name	Reference name of the primitive (can be empty) ➔ Alphanumeric name, has to start with a letter
b_opt	Type of the Boolean operation - add → Add a primitive to the mass body or anchor - sub → Subtract a primitive from a mass body or anchor
body_nr	Reference number of related mass body or anchor - If 0 → The primitive is related to anchors - If > 0 → The primitive is related to the specified mass body
mat_nr	Reference number of the material properties Use existing material properties only (see MATP commands)
layer_nr	Layer number (the current version supports only one layer) - If empty, the primitive is part of the first layer

x	X-coordinate of the connecting point (starting point) for the user defined area
y	Y-coordinate of the connecting point (starting point) for the user defined area
orient_ang	Orientation angle in degrees at the connecting point: 0 ≤ orient_ang < 360

```
LINE, x, y;
```

Parameter	Description
x	X-coordinate of the end point of a straight line
y	Y-coordinate of the end point of a straight line

```
CARC, xm, ym, phi;
```

Parameter	Description
xm	X-coordinate of the center point of a circular arc
ym	Y-coordinate of the center point of a circular arc
phi	Span angle of the circular segment, -360 < phi < +360 → > 0, counter clockwise arc → < 0, clockwise arc

```
BCUR, x, y, x_cp1, y_cp1, x_cp2, y_cp2;
```

Parameter	Description
x	X-coordinate of the end point of the Bezier curve
y	Y-coordinate of the end point of the Bezier curve
x_cp1	X-coordinate of the first control point → If empty, default values will be used

y_cp1	Y-coordinate of the first control point → If empty, default values will be used
x_cp2	X-coordinate of the second control point → If empty, default values will be used
x_cp2	Y-coordinate of the second control point → If empty, default values will be used

***ENDUDAR**

Example of a user defined area with ***FOR**-loops:

```
PARA, numb = 4;
PARA, ampl =10;
*UDAR,,add,1,1,1,-100,150,0          % Start point
 LINE, -50,150;                      % Straight line
 CARC, -50,130,-90;                  % Circular arc
 BCUR,   0,120,-10,130,-20,120;      % Bezier curve
 LINE,  30,130;
 LINE,  30,100;
 *FOR,lines,1,numb                   % Loop of Bezier curves
  BCUR, 30-(100+30)*lines/numb,100+ampl*sin(2*pi*lines/numb);
 *ENDFOR
*ENDUDAR                             % Close the area
```

The same example without ***FOR**-loops:

```
*UDAR,, add,1,1,1,-100,150,0      % Start point
 LINE,  -50,150;                  % Straight line
 CARC,  -50,130,-90;              % Circular arc
 BCUR,    0,120,-10,130,-20,120;  % Bezier curve with
 LINE,   30,130;                  % control points
 LINE,   30,100;
 BCUR, -2.5,110;                  % Bezier curves
 BCUR,  -35,100;                  % no control points
 BCUR,-67.5, 90;
 BCUR, -100,100;
*ENDUDAR
```

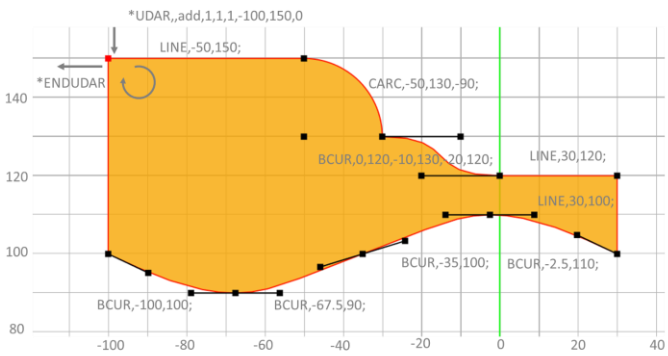


Fig. 49: Example of a user defined area in the SKETCHER Interface

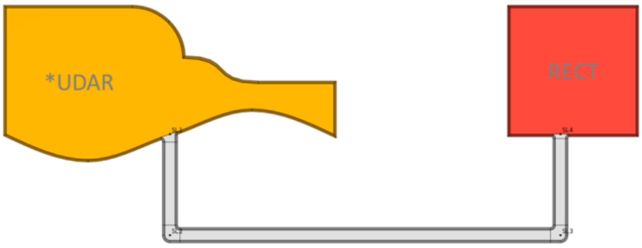


Fig. 50: Example of a user defined area in the MODELBUILDER

SLOC

SLOC, ref_nr, body_nr*, layer_nr, flag, x/p_start, y/p_cent, span_ang;

Defines the location of a connecting point for suspension springs (SLOC)

- The **SLOC** command defines “spring connecting points” and “auxiliary points” for spring elements.
- “Spring connecting points” are placed at the starting and ending point of straight, circular and Bezier springs (see SPRI command). Several springs can share the same connecting point when three or more springs intersect at the same point (e.g. T- or X-junctions). A connecting point is also necessary if springs suddenly change their width.
- “Auxiliary points” define the center points of circular springs and control points of Bezier springs. Several circular or Bezier springs can share the same auxiliary point. Auxiliary points may not be used to define starting and ending points of springs. However, “spring connecting points” and “auxiliary points” can have the same location if springs should start or end at center points or Bezier control points.

- "Spring connecting points" are defined by the flag parameter new (or empty) and "auxiliary points" by the flag parameter aux. It follows the x- and y-coordinates for the SLOC locations.
- For circular springs, the distances between starting and center point (radius 1) must be the same as the distance ending point and center point (radius 2). Hence, the ending point location is related to the starting and center point coordinates and the span angle in between. To guarantee the same radiuses, the ending point of circular springs is defined by the flag parameter rel (related point). It follows the SLOC numbers of the starting point p_start, the center point p_cent and the span angle span_ang. The ending point position is then calculated automatically.
- Both, starting and center point SLOCs must be defined prior assigning related points for circular springs. Positive span angles calculate end point positions counter clockwise and negative values clockwise.
- Avoid a direct connection of two ending points which come from different circular springs. The second spring chain will overwrite the data from the first one and non-consistent models can appear.
- For some applications such as closed loop spring designs, the related ending point of the last circular spring becomes the same coordinate as the first starting point of the loop. In this very special case, the SLOC must be defined twice with the same reference number. At the begin of the loop, a starting point is defined with the flag parameter new. Then follow four related SLOC with 90° span angle. The last SLOC becomes the same position as the first SLOC of the loop. In order to avoid warnings, the last SLOC must be defined with the flag parameter dup (duplicated point). An example is shown in the figure below.
- The body_nr* parameter is automatically determined in the current release. Leave the parameters blank.

Input Data

Parameter	Description
ref_nr	Reference number of the spring connecting point (SLOC) ≥ 1 [integer]
body_nr*	Body reference number Determined automatically, leave the parameter blank
layer_nr	Layer number (the current version supports only one functional layer)

	- If empty, the **SLOC** is at the center of the first layer
`flag`	Flag of the spring connecting point - new (or empty) = Defines a spring connecting point - aux = Defines an auxiliary point - rel = Defines an ending point of circular springs - dup = Defines a duplicated ending point of circular springs
`x/p_start`	X-coordinate of the spring connecting point (if `flag`=rel or aux) OR Starting point **SLOC** reference number (if `flag`=rel or dup)
`y/p_cent`	Y-coordinate of spring connecting point (if `flag`=rel or aux) OR Center point **SLOC** reference number (if `flag`=rel or dup)
`span_ang`	Span angle in degrees (if `flag`=rel or dup) If > 0, counter clockwise span angle (0 < span_ang < 360) If < 0, clockwise span angle (-360 < span_ang < 0)

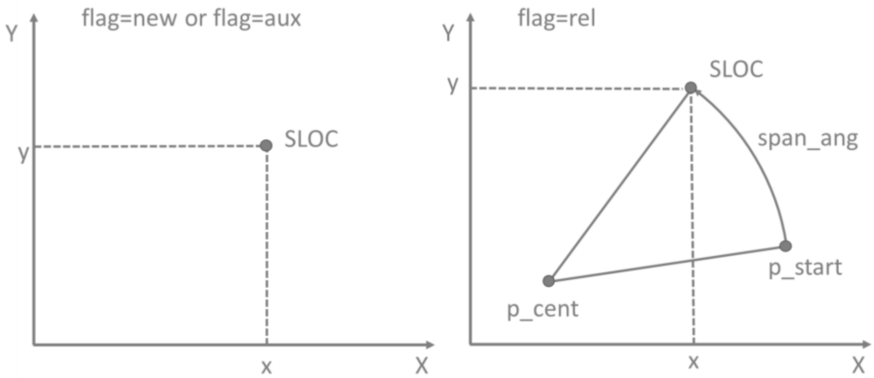
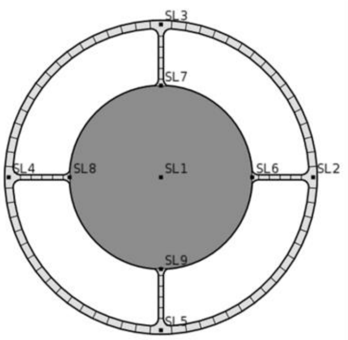


Fig. 51: Dimensional parameters of a spring location point




```
PARA,r_anchor=60;
PARA,rad_ring=100;
SLOC,1,,1,aux,0,0;
SLOC,2,,1,new,rad_ring,0;
SLOC,3,,1,rel,2,1,90;
SLOC,4,,1,rel,3,1,90;
SLOC,5,,1,rel,4,1,90;
SLOC,2,,1,dub,5,1,90;
SLOC,6,,1,new,r_anchor,0;
SLOC,7,,1,new,0,r_anchor;
SLOC,8,,1,new,-r_anchor,0;
SLOC,9,,1,new,0,-r_anchor;
```

```
PARA,spr_w=3;
SPRI,2,3,1,,2*spr_w;
SPRI,3,4,1,,2*spr_w;
SPRI,4,5,1,,2*spr_w;
SPRI,5,2,1,,2*spr_w;
SPRI,6,2,,,spr_w;
SPRI,7,3,,,spr_w;
SPRI,8,4,,,spr_w;
SPRI,9,5,,,spr_w;
```

Fig. 52: Example of SLOC and SPRI settings for circular springs

SPRI

```
SPRI, sloc1, sloc2, sloc3, sloc4, width1, width2, cr_left1,
cr_right1, cr_left2, cr_right2, mc_left, mc_right, es_left1,
es_right1, es_left2, es_right2, lmdiv, wmdiv, romdiv;
```

Creates a straight, a circular or a Bezier spring between two connecting points

- "Straight springs" are defined by the first two connecting points (sloc3 and sloc4 must be empty).
- "Circular springs" are defined by the first three connecting points (sloc4 must be empty). Sloc1 is the starting point, sloc2 the ending point and sloc3 the center point (auxiliary point).
- "Bezier springs" are defined by all four connecting points. Sloc1 is the starting point, sloc2 the ending point, sloc3 the first control point and sloc4 the second control point. Sloc3 and sloc4 are auxiliary points (see SLOC command). Bezier springs are not available in the current release.
- Springs have either a constant or a linear varying beam width (uniform or tapered beams). The thickness is taken from the LAYR command.
- Default values for the corner fillet radiuses can be overwritten by parameters cr_left1, cr_right1, cr_left2, cr_right2. A zero value disables the corner fillet at the specified location.
- Default mask undercut values can be overwritten by parameters mc_left, mc_right of above command.
- Default etch sidewall offset data can be overwritten by parameters es_left1, es_right1, es_left2, es_right2 of above command.
- Parameters lmdiv, wmdiv overwrite default FEM mesh settings.
- Parameter romdiv overwrites the number of beam elements used for the springs in the MODELBUILDER.

Input Data

Parameter	Description
sloc1	Spring connecting point at the begin of the spring - Number of the SLOC (integer)
sloc2	Spring connecting point at the end of the spring - Number of the SLOC (integer)
sloc3	Auxiliary point for circular and Bezier springs - If empty, a straight spring will be generated

	- Circular springs: sloc3 is the center point - Bezier springs: sloc3 is the first control point
sloc4	Auxiliary point Bezier springs - Bezier springs: sloc3 is the second control point - Must be empty for straight and curved springs
width1	Width of the beam at the begin of the spring
width2	Width of the beam at the end of the spring - If empty, width2 = width1
cr_left1	Corner fillet radius at the left side of the spring at sloc1 - If empty, the default fillet radius is used
cr_right1	Corner fillet radius at the right side of the spring at sloc1 - If empty, the default fillet radius is used
cr_left2	Corner fillet radius at the left side of the spring at sloc2 - If empty, the default fillet radius is used
cr_right2	Corner fillet radius at the right side of the spring at sloc2 - If empty, the default fillet radius is used
mc_left	Mask undercut value (top face edge) left side → If empty, default mask undercut value is used
mc_right	Mask undercut value (top face edge) right side → If empty, default mask undercut value is used
es_left1	Etch sidewall value (bottom edge) left side of sloc1 - If empty, default etch sidewall value is used
es_right1	Etch sidewall value (bottom edge) right side of sloc1 - If empty, default etch sidewall value is used
es_left2	Etch sidewall value (bottom edge) left side of sloc2 - If empty, default etch sidewall value is used
es_right2	Etch sidewall value (bottom edge) right side of sloc2 - If empty, default etch sidewall value is used
lmdiv	User defined spring length mesh division for FEM export - If empty, default FEM mesh settings data are used
wmdiv	User defined spring width mesh division for FEM export

	- If empty, default FEM mesh settings data are used
romdiv	User defined number of beam elements in the MODEL-BUILDER - If empty, default ROM settings data are used

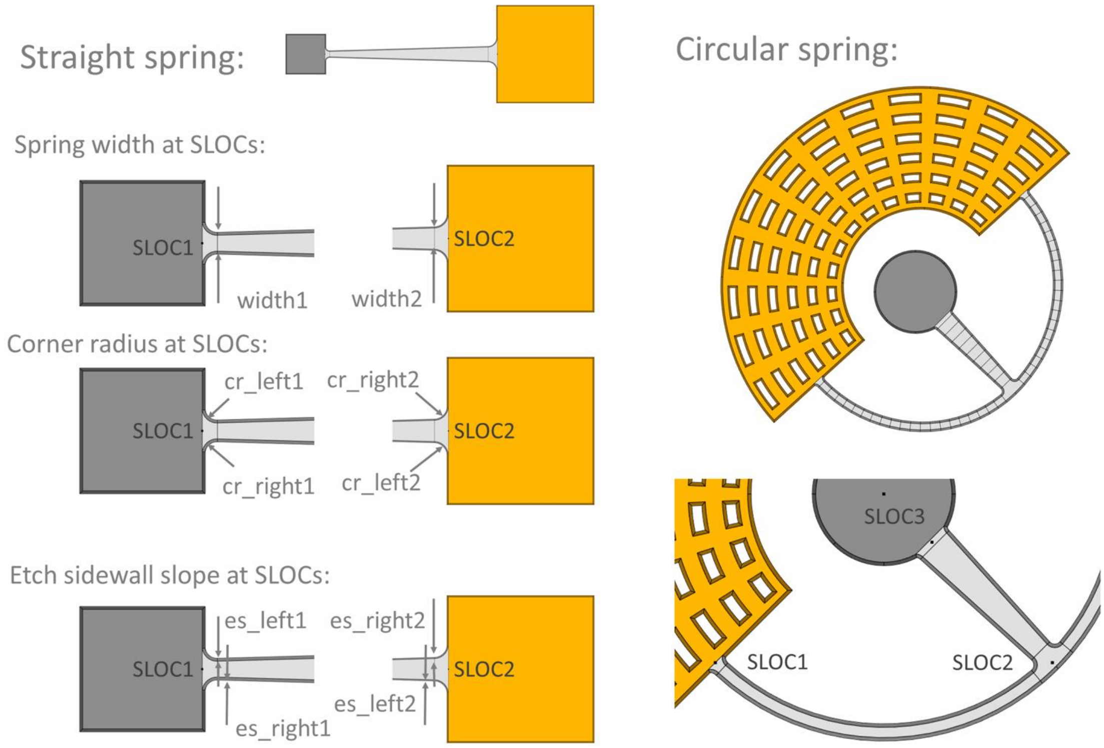


Fig. 53: Dimensional parameters of spring elements

PERF

```
PERF, perf_name, type, body_nr, mat_nr, layer_nr, loc_x,loc_y,
length, height, dx/dr, dy/dphi, n_x/n_r, n_y/n_phi, csys, xm,
ym, c_flag, orient_ang;
```

Creates a pattern of perforations in seismic masses or anchors

- Rectangular and circular perforation patterns are either defined in a Cartesian or a cylinder coordinate system.
- Perforations are created in primitives with the same body and material reference number.
- Boolean operations for intersections of perforations with other perforations or intersections with outer lines of masses can by activated or deactivated by the `c_flag`.

Input Data

Parameter	Description
perf__name	Reference name of the primitive (can be empty) ➔ Alphanumeric name, has to start with a letter
type	Type of the perforation pattern - rect → Create rectangular perforations - circ → Create circular perforations
body_nr	Reference number of related mass body - If > 0 → The perforations are related to the specified mass body - If 0 → The perforations are related to anchors
mat_nr	Reference number of the material properties - Use existing material properties only (MATP command)
layer_nr	Layer number (the current version supports only one layer) - If empty, the primitive is part of the first layer
loc_x	X-coordinate of the center of the perforation pattern
loc_y	Y-coordinate of the center of the perforation pattern
length	First dimensional parameter of the pattern element - type = rect ▪ csys=0 → Length of the pattern element ▪ csys=1 → Radial width of the pattern element - type = circ ▪ Radius of the pattern element
height	Second dimensional parameter of the pattern element - type = rect ▪ csys=0 → Width of the pattern element ▪ csys=1 → Span angle of the pattern element - type = circ (Unused)
dx/dr	Distance of pattern elements in x-direction/rad-direction
dy/dphi	- csys=0 → Distance between pattern elements in y-direction - csys=1 → Span angle between pattern elements in phi-direction

nx/nr	Number of elements in x-direction/rad-direction
ny/nphi	Number of elements in y-direction/phi-direction
csys	Type of the coordinate system 0 → Cartesian coordinate system (default) 1 → Cylinder coordinate system
xm	X-coordinate of the center point of the cylinder coordinate system
ym	Y-coordinate of the center point of the cylinder coordinate system
c_flag	Consider intersections of perforation patterns - If 0 → do not consider intersections of perforations (default) - If 1 → consider intersections with other perforation patterns and outer lines of mass bodies or anchors
orient_ang	Orientation angle in degrees at the connecting point: 0 ≤ orient_ang < 360

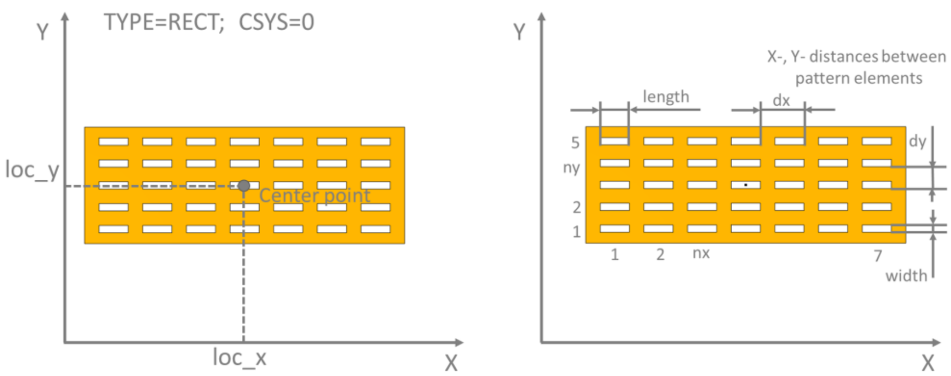


Fig. 54: Rectangular pattern in Cartesian coordinates

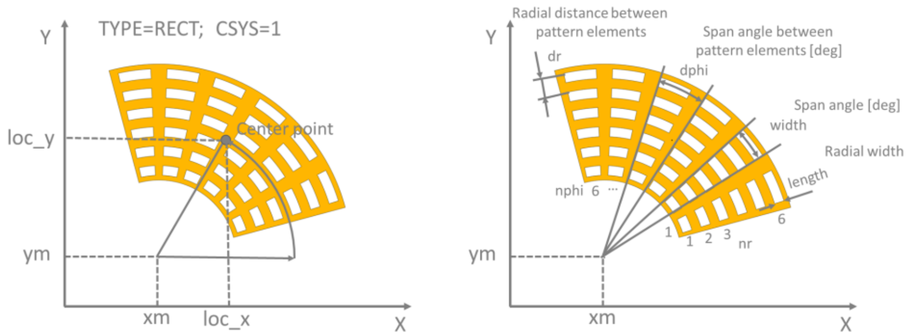


Fig. 55: Rectangular pattern in cylinder coordinates

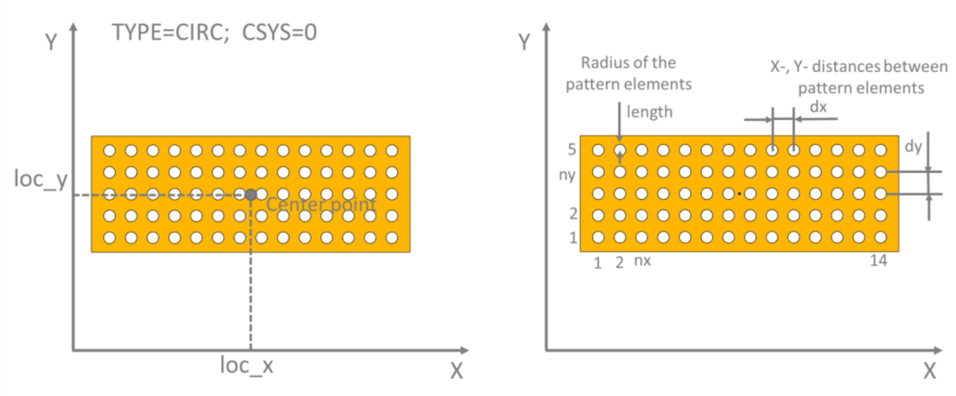


Fig. 56: Circular pattern in Cartesian coordinates

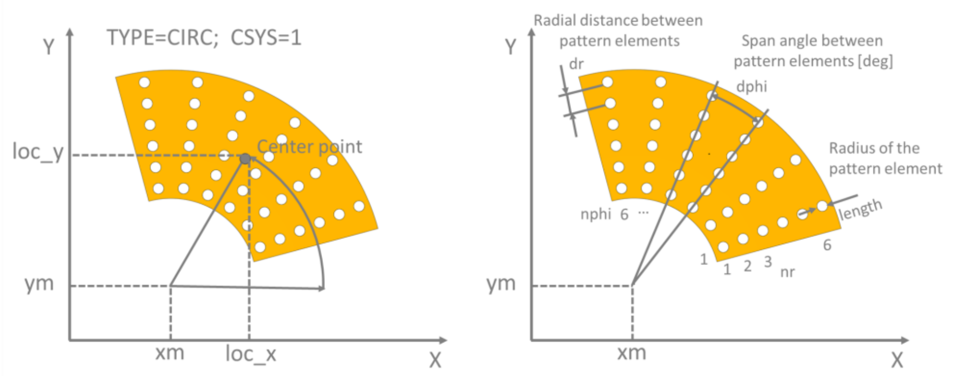


Fig. 57: Circular pattern in cylinder coordinates

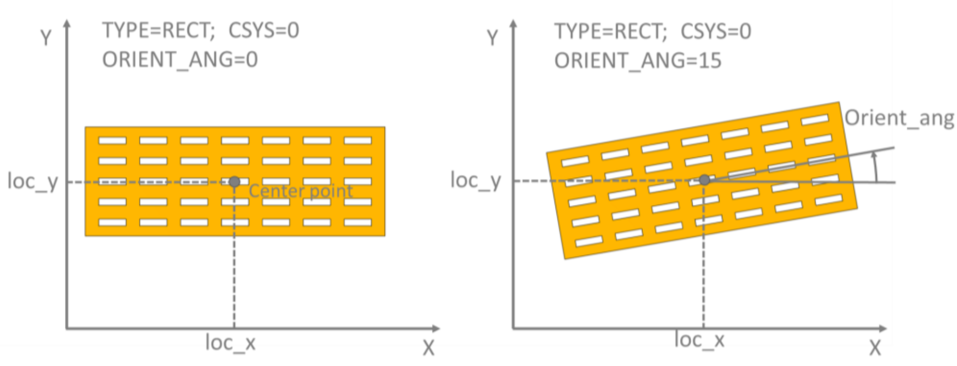


Fig. 58: Rotated rectangular pattern in Cartesian coordinates

STOP

```
STOP, stop_name, type, body_nr, mat_nr, layer_nr, xc, yc, xt, yt,
height/radius, length, kn, d_fact;
```

Creates a stopper element to limit the in-plane travel range

- Creates a contact element between a movable contact and a fixed target node. The contact node is defined by the xc- and yc-coordinates and the target node by the xt- and yt-coordinates.

- The stopper limits the travel range in the direction from the contact to the target node. Tangential motion components (slide) are not restricted.
- The default contact stiffness and damping factor can be overwritten by the kn- and d_fact-parameters. The travel range between contact nodes and target nodes will be reduced by the stopper size defined by the height/radius-parameter.
- The movable contact domain is either represented by a rectangular or a circular stopper with the center at the contact node position.

Input Data

Parameter	Description
stop_name	Reference name of the primitive (can be empty) ➔ Alphanumeric name, has to start with a letter
type	Type of the stopper element - rect → Creates a rectangular stopper element - circ → Creates a circular stopper element
body_nr	Reference number of related mass body - > 0 → The contact is related to the specified mass body
mat_nr	Reference number of the material properties - Use existing material properties only (see MATP commands)
layer_nr	Layer number (the current version supports only one layer) - If empty, the primitive is part of the first layer
xc	X-coordinate of the movable contact node
yc	Y-coordinate of the movable contact node
xt	X-coordinate of the fixed target node
yt	Y-coordinate of the fixed target node
height/radius	First dimensional parameter of the contact domain - rect → Height of the rectangular stopper element - circ → Radius of the circular stopper element
length	Second dimensional parameter of the contact domain - rect → Length of the rectangular stopper element - circ → Unused

kn	Contact stiffness - If empty, use the default contact stiffness
d_fact	Contact damping factor - If empty, use the default contact damping factor

ZLIM

`ZLIM, zlim_name, body_nr, layer_nr, flag, xc, yc, zt, kn, d_fact;`

Limits the travel range in vertical direction (out-of-plane motion)

- Creates an out-of-plane contact element with default or user defined stiffness and contact damping factors.
- The `flag` parameter specifies weather the travel range should be limited upwards at the top face of the functional layer (`flag=top`) or downwards at the bottom face (`flag=bot`).
- The out-of-plane travel range is limited to the `zt`-value at the contact node position defined by the xc- and yc-coordinate.

Input Data

Parameter	Description
zlim_name	Reference name of the primitive (can be empty) ➔ Alphanumeric name, has to start with a letter
body_nr	Reference number of related mass body - If > 0 → The primitive is related to the specified mass body
layer_nr	Layer number (the current version supports only one layer) - If empty, the primitive is part of the first layer
flag	Direction flag top → Limits upwards motion bot → Limits downwards motion
xc	X-coordinate of the movable contact node
yc	Y-coordinate of the movable contact node
zt	Travel range in z-direction
kn	Contact stiffness - If empty, use the default contact stiffness

d_fact	Contact damping factor - If empty, use the default contact damping factor

COMB

COMB, capa_label, type, body_nr, mat_nr, layer_nr, alpha, n_comb, flag, f_w, f_l, f_bl, tr, gap1, gap2, gap3, x, y, xm, ym;

Creates a comb cell for capacitive sensing and electrostatic actuation

- Several comb cell library elements can be linked to seismic masses.
- Comb cells with the same capa_label are automatically combined to a single capacitance.
- Comb cells consist of local conductors which can be linked to voltage ports by the COND command. The first local conductor is always the moving comb, the other local conductors are the fixed combs.
- If two fixed combs exist (see type dif1 and dif2), the second conductor is on the right of the moving fingers and the third conductor is on the left of the moving fingers.

Input Data

Parameter	Description
capa_label	Capacitance label of the comb cell - Alphanumeric label, has to start with a letter
type	Type of the comb cell library element - available types: ▪ area → Comb with area variation ▪ circ → Circular comb with area variation ▪ asym → Asymmetric comb with gap variation ▪ dif1 → Differential comb (clamped on one side) ▪ dif2 → Differential comb (clamped on both sides)
body_nr	Reference number of related mass body - If > 0 → the primitive is related to the specified mass body
mat_nr	Reference number of the material properties - Use existing material properties only (see MATP commands)
layer_nr	Layer number (the current version supports only one layer) - If empty, the primitive is part of the first layer

`alpha`	Orientation angle of straight comb cells - Ignored for circular combs
`n_comb`	Number of fingers of the comb cell at the seismic mass
`flag`	Fixed finger flag (see graphical explanation) → Defines different modifications of fixed comb fingers
`f_w`	Finger width
`f_l`	Finger length (see graphical explanation) - Straight comb cells → Length - Curved comb cells → Span angle
`f_bl`	Fixed anchor block length - Straight comb cells → Length - Curved comb cells → Span angle
`tr`	Travel range of comb cell fingers - Straight comb cells → Travel range - Curved comb cells → Span angle of the travel range
`gap1`	Gap between opposite fingers
`gap2`	Second gap (see graphical explanation)
`gap3`	Third gap (see graphical explanation)
`x`	X-coordinate of comb cell connecting point
`y`	Y-coordinate of comb cell connecting point
`xm`	X-coordinate of the center point if `type` = circ
`ym`	Y-coordinate of the center point if `type` = circ

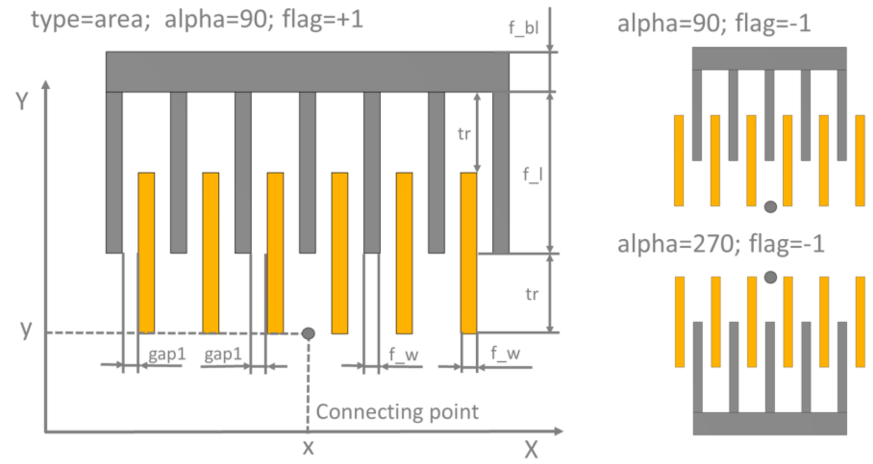


Fig. 59: Comb cell parameters of type: “area”

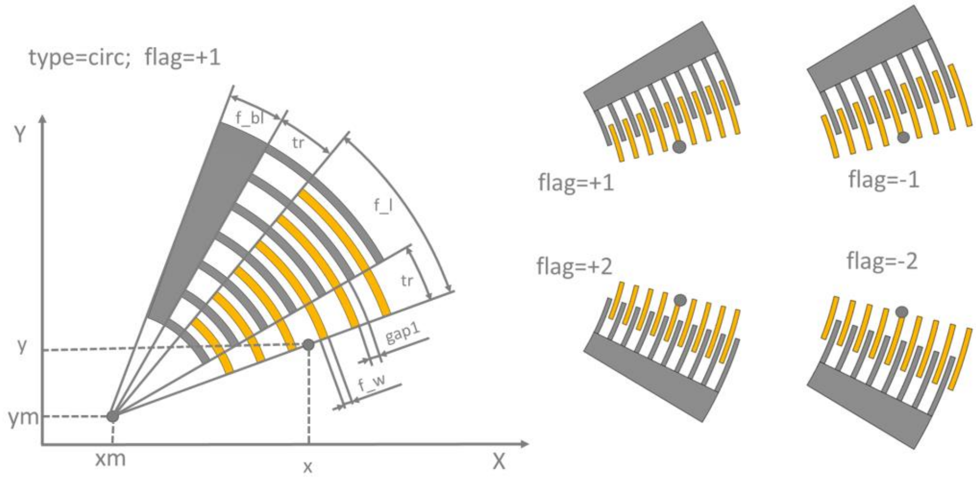


Fig. 60: Comb cell parameters of type: “circ” (all angles are defined in degrees)

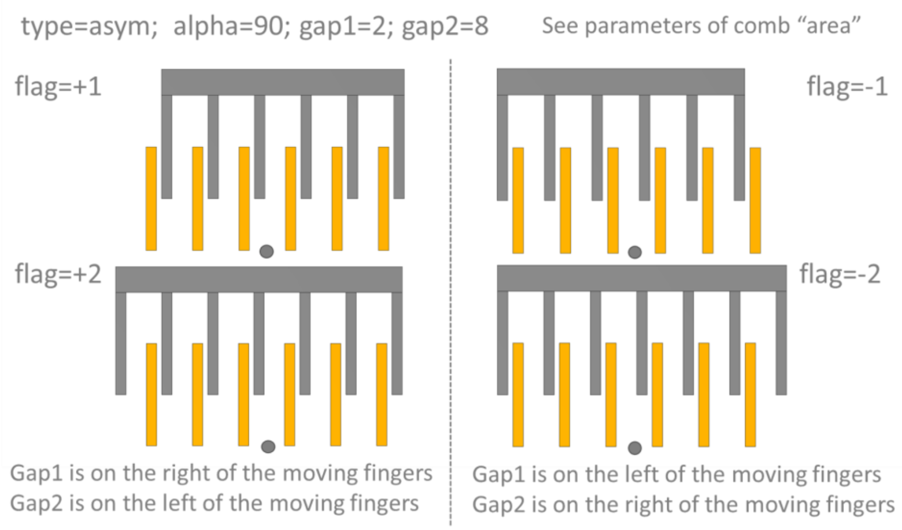


Fig. 61: Comb cell parameters of type: “asym”

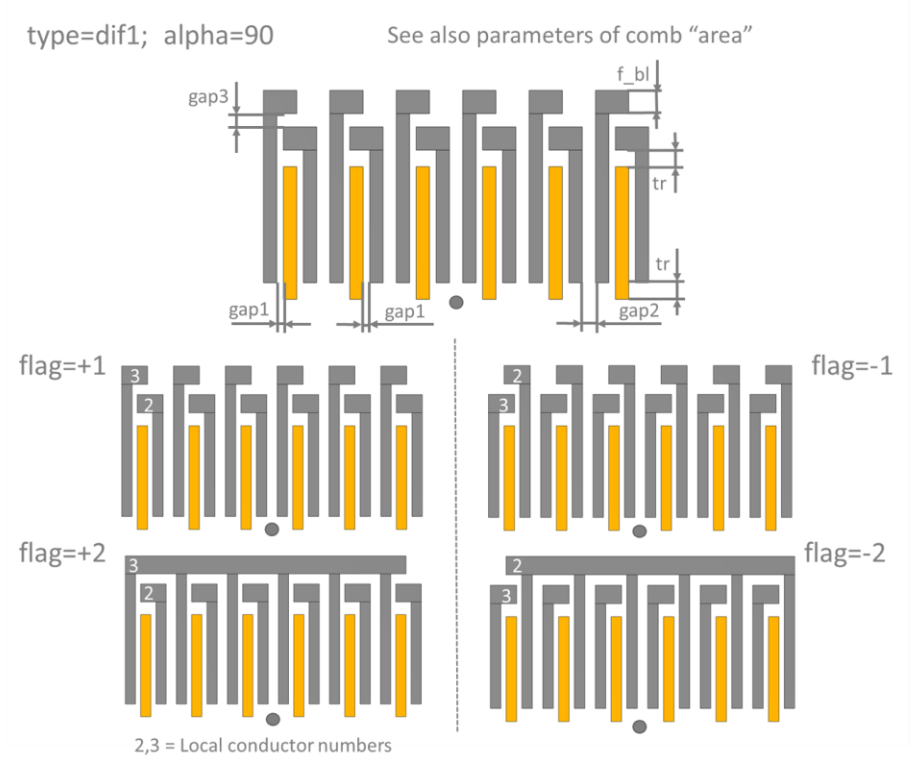


Fig. 62: Comb cell parameters of type: "dif1"

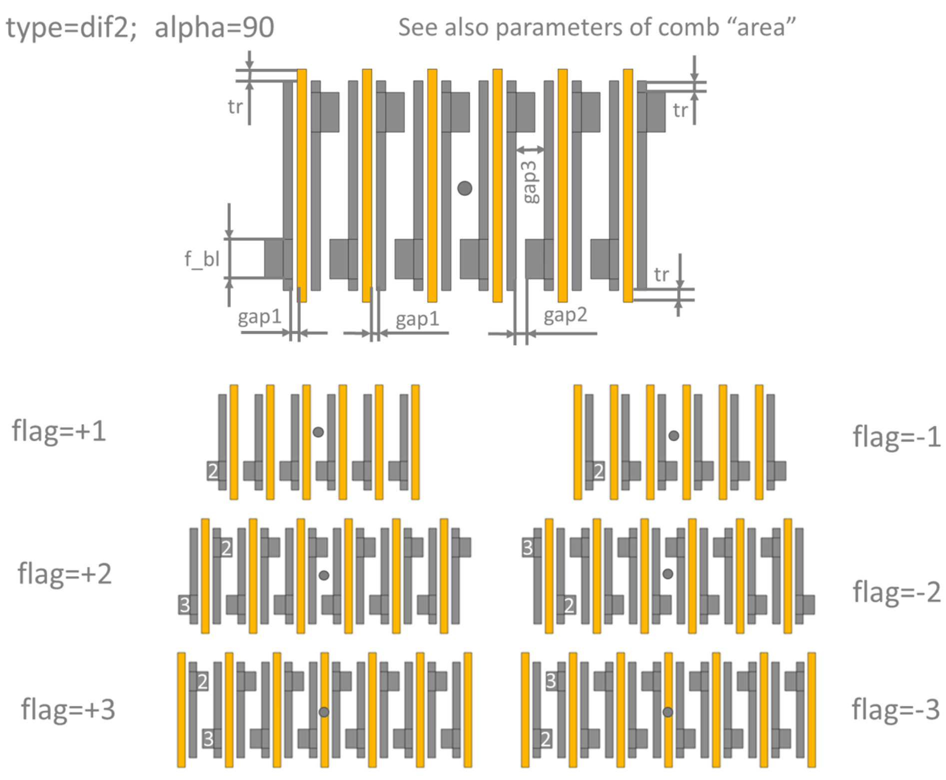


Fig. 63: Comb cell parameters of type: "dif2"

PCAP

PCAP, pcap_name, capa_label, type, b_opt, target_body, ref_nr, layer_nr, p1, p2, p3 ... pn, orient_ang, fill_fact;

Creates a primitive for a parallel plate capacitor

- Parallel plate capacitors are assembled from several primitives (rect, circ, tria, poly) defined by the type-parameter.
- Boolean operations are applied to all elements in the same group. Group elements have the same target_body number, ref_nr and layer_nr. The pcap_name of all group elements must be the same.
- If the capa_label of different groups is the same, capacitances are combined to a single capacitance value.
- The target_body number refers to the mass body reference number of the moving conductor.
- The ref_nr is used to create individual capacitances acting in the same target_body and have the same layer_nr.
- The layer_nr is used to calculate the electrode gap to moving mass bodies at the initial position (see LAYR command).
- Parallel plate primitives consider out-of-plane and in-plane motion components. The capacitance change of in-plane motion components is linearized at the initial position. The influence of perforations holes (see PERF command) is taken into account but fringing fields in holes and outside the plate area are ignored.
- Perforations holes reduce the plate area and associated capacitances. An area fill factor fill_fact can be used to scale the capacitances of the primitive. Area fill factors less than one are commonly used to defined an effective area in models where the perforations are ignored. Alternatively, area fill factors greater than one are used to account for the influence of fringing fields in perforation holes, which increased the capacitances.

Input Data

Parameter	Description
pcap_name	Reference name of the parallel plate capacitor - Alphanumeric name, has to start with a letter
capa_label	Capacitance label of the parallel plate primitive - Alphanumeric label, has to start with a letter
type	Type of the parallel plate element (primitive)

	- Available types: ▪ rect → Rectangular primitive ▪ circ → Circular primitive ▪ elps → Elliptical primitive ▪ tria → Triangular primitive ▪ poly → Polygonal primitive
b_opt	Type of the Boolean operation - add → Add the area of the primitive to a capacitance - sub → Subtract the area of the primitive from a capacitance
target_body	Target mass body number to define the moving conductor → Integer > 0, related to the specified mass body
ref_nr	Reference number to create different capacitances on the same target mass and the same layer → Integer > 0
layer_nr	Layer reference number to define the electrode gap → Integer > 1, related to the specified layer

type = rect (see RECT-command)

Parameter	Description
p1	X-coordinate of the connecting point for rectangular primitives
p2	Y-coordinate of the connecting point for rectangular primitives
p3	Length (x-direction) of the rectangular primitive (can be negative)
p4	Height (y-direction) of the rectangular primitive (can be negative)
orient_ang	Orientation angle in degrees at the connecting point: 0 ≤ orient_ang < 360
fill_fact	Area fill factor of the primitive → > 0, Default = 1

type = circ (see CIRC-command)

Parameter	Description
p1	X-coordinate of the center point (connecting point)
p2	Y-coordinate of the center point (connecting point)
p3	First radius of circular primitive
p4	Second radius of circular primitive - If empty or p4 = 0, it defines a solid circle
p5	Starting angle of the circular segment: 0 ≤ phi1 < 360
p6	Ending angle of the circular segment: 0 < phi2 ≤ 360
orient_ang	Orientation angle in degrees at the connecting point: 0 ≤ orient_ang < 360
fill_fact	Area fill factor of the primitive → > 0, Default = 1

type = elps (see ELPS-command)

Parameter	Description
p1	X-coordinate of the center point (connecting point)
p2	Y-coordinate of the center point (connecting point)
p3	Radius in x-direction of the non-rotated ellipse
p4	Radius in y-direction of the non-rotated ellipse
p5	Starting angle of the circular segment: 0 ≤ phi1 < 360
p6	Ending angle of the circular segment: 0 < phi2 ≤ 360
orient_ang	Orientation angle at the elliptical primitive: 0 ≤ orient_ang < 360
fill_fact	Area fill factor of the primitive → > 0, Default = 1

type = tria (see TRIA-command)

Parameter	Description
p1	X-coordinate of the first point (connecting point)
p2	Y-coordinate of the first point (connecting point)
p3	Second x-coordinate of the primitive
p4	Second y-coordinate of the primitive
p5	Third x-coordinate of the primitive
p6	Third y-coordinate of the primitive
orient_ang	Orientation angle in degrees at the connecting point: 0 ≤ orient_ang < 360
fill_fact	Area fill factor of the primitive → > 0, Default = 1

type = poly (see POLY-command)

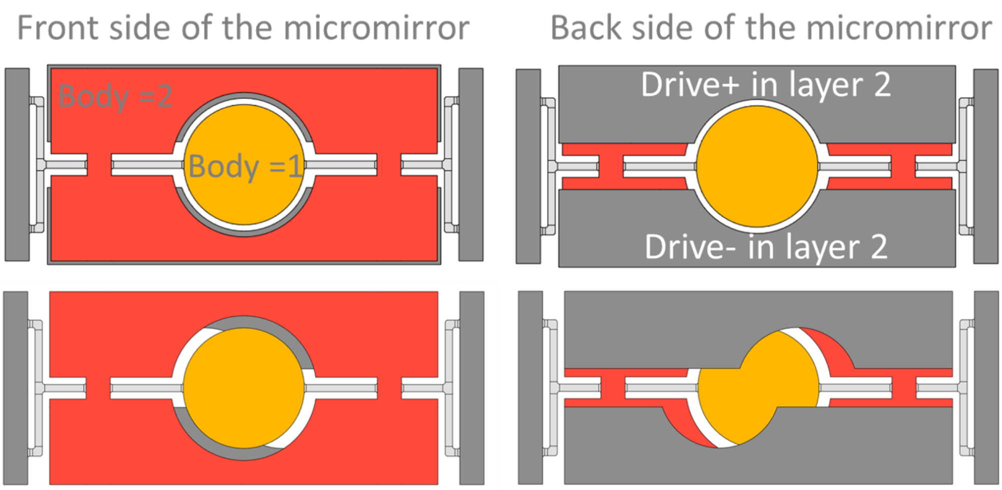
Parameter	Description
p1	Total number of polygon points
p2	X-coordinate of the first point (connecting point)
p3	Y-coordinate of the first point (connecting point)
p4	Second x-coordinate of the primitive
p5	Second y-coordinate of the primitive
…, pn	Further pairs of x-y-coordinates
orient_ang	Orientation angle in degrees at the connecting point: 0 ≤ orient_ang < 360
fill_fact	Area fill factor of the primitive → > 0, Default = 1

The following figure shows an example where different reference numbers (`ref_nr`) are necessary for modeling of bottom plate capacitors.

The micromirrors consists of two mass bodies, the inner mirror mass with body reference number 1 and the outer actuation mass with body number 2. The upper drive capacitor (drive+) acts on mass body 2 (target mass) and is placed in layer 2.

The lower drive capacitor (drive-) acts on the same mass body and is placed in the same layer. Each capacitor is modeled be a rectangular primitive and a circular cut.

In the lower figures, the circular cuts are shifted to the left and to the right to demonstrate that Boolean operations are only applied to primitive in the same group.




```
% PCAP: drive+; Target Body number: 2; Reference number: 1; Layer number: 2
PCAP,drive+,rect,add,2,1,2,-m_length/2,m_width/2,m_length,-m_width*0.4,0;
PCAP,drive+,circ,sub,2,1,2,-m_length*0.1,0,0,m_rad,0,360;

% PCAP: drive-; Target Body number: 2; Reference number: 2; Layer number: 2
PCAP,drive-,rect,add,2,2,2,-m_length/2,-m_width/2,m_length,m_width*0.4;
PCAP,drive-,circ,sub,2,2,2,m_length*0.1,0,0,m_rad,0,360;
```

Fig. 64: Parallel plate capacitors with different reference numbers

MAST

```
MAST, mast_prefix, ref_nr, body_nr, layer_nr, x, y, z;
```

User defined master node (MN) for loads and displacement monitoring

- Master nodes are located on or in seismic masses.
- Master nodes are used to apply loads, displacement constraints or master nodes are used to monitor displacement results at certain points of mass bodies.

Input Data

Parameter	Description
mast_prefix	Reference name of the primitive (can be empty) ➔ Alphanumeric name, has to start with a letter
ref_nr	Reference number of the master node ➔ ≥ 1 [integer]
body_nr	Reference number of related mass body ➔ > 0, the primitive is related to the specified mass body
layer_nr	Layer number (the current version supports only one functional layer)
x	X-coordinate of master node location
y	Y-coordinate of master node location
z	Z-coordinate of master node location

3 Solution commands

MSUP

`MSUP, flag;`

Activates or deactivates the modal superposition solver for the following load steps

- The modal superposition solver is activated or deactivated in the ROM Settings window of the GUI. The number of modes for the solver are specified in the same window.
- Independent on the default settings, the command activates or deactivates the MSUP-solver for the following load steps.

Input Data

Parameter	Description
`flag`	Flag to activate/deactivate the modal superposition solver - 1 or true → Activates modal superposition - 0 or false → Deactivates modal superposition

REDS

`REDS, flag;`

Defines the reduction stage of simulation models for the following load steps

- The default reduction stage is defined in the ROM Settings window.
- The ROM reduction stage defines what motion degrees of freedom (DOF) are considered for simulation. Motion DOF are displacements (ux, uy, uz) and rotations (rx, ry, rz) at the rigid body center points (RB), the spring connecting points (SLOC) and internal nodes (IN) between Timoshenko beam elements.
- Deactivated motion DOF are eliminated by a Guyan condensation method.

Input Data

Parameter	Description
`flag`	Flag to change the ROM reduction stage for the following load steps - 0 → All motion degrees of freedom - 1 → Only rigid body motion degrees of freedom (RB) - 2 → Rigid body (RB) and spring connecting points (SLOC)

LOAD

`LOAD, node_type, node_nr, dir_label, dcval, acval, sigtype, sigpar1, sigpar2;`

Defines nodal loads and DOF-constraints for the mechanical domain

- Nodal loads are forces and moments acting on the rigid body center points, at master nodes and at spring connecting points.
- Nodal DOF-constraints are given displacements and rotations at the rigid body center points, at master nodes and at spring connecting points.
- The command supports constant (DC), harmonic (AC) and transient load functions of type sine, step, puls and ramp.

Input Data

Parameter	Description
node_type	Available node types (enter a two-character string) - rb → Rigid body center point (center of gravity) - mn → Master node (MAST) - sl → Spring location point (SLOC)
node_nr	Number of the node associated with above node type - ≥ 1 [integer]
dir_label	Available load types and directions (enter a two-character string) - Forces or moments: → fx or mx → fy or my → fz or mz - Displacements or rotations (in radian): → ux or rx → uy or ry → uz or rz
dcval	DC-value of the load, used for all types of simulations - Real value - del → Delete specified load
acval	AC-value of the load, used in harmonic and transient simulations - Real value

sigtype	Available transient load types (enter a 4-character string) - sine → Sinusoidal function - step → Step function - puls → Periodic pulse function - ramp → Ramp function - If empty → Only dcval for transient simulations
sigpar1	- sigtype= sine: ▪ sigpar1 → Frequency > 0 - sigtype= step: ▪ sigpar1 uz Step time ≥ 0 - sigtype= puls: ▪ sigpar1 → Period time > 0 - sigtype= ramp: ▪ sigpar1 → Ramp time till acval is reached - The high-values for all load types are taken from acval
sigpar2	- sigtype= sine: ▪ sigpar2 → Phase (default = 0) - sigtype= step or ramp: ▪ sigpar2 unused - sigtype= puls: ▪ sigpar2 → High-pulse time (default = 0.5)

ACEL

```
ACEL, dir_label, dcval, acval, sigtype, sigpar1, sigpar2;
```

Defines translational or rotational acceleration loads

- Acceleration loads are used for inertial effects. Centrifugal forces are ignored.

Input Data

Parameter	Description
dir_label	Available directions (enter a two-character string) → ux or rx → uy or ry → uz or rz
dcval	DC-value of the load, used for all types of simulations

	- Real value - del → Delete specified acceleration
acval	AC-value of the load, used in harmonic and transient simulations - Real value
sigtype	Available transient load types (enter a 4-character string) - sine → Sinusoidal function - step → Step function - puls → Periodic pulse function - ramp → Ramp function - if empty → Only dcval for transient simulations

sigpar1	- sigtype= sine:
	  ▪ sigpar1 → Frequency > 0
	- sigtype= step:
	  ▪ sigpar1 → Step time ≥ 0
	- sigtype= puls:
	  ▪ sigpar1 → Period time > 0
	- sigtype= ramp:
	  ▪ sigpar1 → Ramp time till acval is reached
	- The high-values for all load types are taken from acval

sigpar2	- sigtype= sine:
	  ▪ sigpar2 → Phase (default = 0)
	- sigtype= step or ramp:
	  ▪ sigpar2 unused
	- sigtype= puls:
	  ▪ sigpar2 → High-pulse time (default = 0.5)

OMEG

```
OMEG, axis, dcval, acval, sigtype, sigpar1, sigpar2;
```

Defines angular rates to model Coriolis forces and moments

- Coriolis effects are considered in static, harmonic and transient simulations.
- In a static simulation, the drive mode must be defined by the DRIV-command.

Input Data

Parameter	Description
axis	Global rotation axis (enter a one-character string) → X → Y → Z
dcval	DC-value of the load, used for all types of simulations - Real value - del → Delete specified angular rate
acval	AC-value of the load, used in transient simulations only - Real value
sigtype	Available transient load types (enter a 4-character string) - sine → Sinusoidal function - step → Step function - puls → Periodic pulse function - ramp → Ramp function - if empty → Only dcval for transient simulations
sigpar1	- sigtype= sine: ▪ sigpar1 → Frequency > 0 - sigtype= step: ▪ sigpar1 → Step time ≥ 0 - sigtype= puls: ▪ sigpar1 → Period time > 0 - sigtype= ramp: ▪ sigpar1 → Ramp time till acval is reached - The high-values for all load types are taken from acval
sigpar2	- sigtype= sine: ▪ sigpar2 → Phase (default = 0) - sigtype= step or ramp: ▪ sigpar2 unused - sigtype= puls: ▪ sigpar2 → High-pulse time (default = 0.5)

VSCR

VSCR, cond_port, dcval, acval, sigtype, sigpar1, sigpar2;

Defines a voltage signal at a conductor port

- Voltage loads can be applied for all simulation types.

Input Data

Parameter	Description
cond_port	Name of the conductor port - Alphanumeric name, has to start with a letter
dcval	DC-value of the load, used for all types of simulations - Real value - del → Delete specified voltage source
acval	AC-value of the load, used in harmonic and transient simulations - Real value
sigtype	Available transient load types (enter a 4-character string) - sine → Sinusoidal function - step → Step function - puls → Periodic pulse function - ramp → Ramp function - if empty → Only dcval for transient simulations
sigpar1	- sigtype= sine: ▪ sigpar1 → Frequency > 0 - sigtype= step: ▪ sigpar1 → Step time ≥ 0 - sigtype= puls: ▪ sigpar1 → Period time > 0 - sigtype= ramp: ▪ sigpar1 → Ramp time till acval is reached - The high-values for all load types are taken from acval
sigpar2	- sigtype= sine: ▪ sigpar2 → Phase (default = 0) - sigtype= step or ramp: ▪ sigpar2 unused - sigtype= puls: ▪ sigpar2 → High-pulse time (default = 0.5)

ISCR

ISCR, cond_port, dcval, acval, sigtype, sigpar1, sigpar2;

Defines a current signal at a conductor port

- Current loads can be applied to harmonic and transient simulations.
- Current loads are not supported for the SIMULINK model export.

Input Data

Parameter	Description
cond_port	Name of the conductor port - Alphanumeric name, has to start with a letter
dcval	AC-value of the load, used in harmonic and transient simulations - Real value - del → Delete specified current source
acval	AC-value of the load, used in harmonic and transient simulations - Real value
sigtype	Available transient load types (enter a 4-character string) - sine → Sinusoidal function - step → Step function - puls → Periodic pulse function - ramp → Ramp function - if empty → Only dcval for transient simulations
sigpar1	- sigtype= sine: ▪ sigpar1 → Frequency > 0 - sigtype= step: ▪ sigpar1 → Step time ≥ 0 - sigtype= puls: ▪ sigpar1 → Period time > 0 - sigtype= ramp: ▪ sigpar1 → Ramp time till acval is reached - The high-values for all load types are taken from acval
sigpar2	- sigtype= sine: ▪ sigpar2 → Phase (default = 0) - sigtype= step or ramp: ▪ sigpar2 unused - sigtype= puls: ▪ sigpar2 → High-pulse time (default = 0.5)

COND

`COND, capa_label, cond_port1, cond_port2, cond_port3, cond_port4, cond_port5, cond_port6, cond_port7, cond_port8;`

Assigns conductor ports to comb and plate capacitances

- Conductors ports are used to apply voltage and current loads by VSCR and ISCR commands. Capacitances which are not linked to conductors are ignored.
- The number of conductor ports for comb cells depends on the selected type of the comb cell library element. The first port is always on the moving mass. The order of other ports is defined by the COMB command.
- Plate capacitances are related to two conductor ports whereby the first one is on the moving mass and the second one is the fixed bottom plate conductor.

Input Data

Parameter	Description
capa_label	COMB or PCAP capacitance label (must already exist) - Alphanumeric label, has to start with a letter
cond_port1 … cond_port8	Conductor port reference name - Alphanumeric name, has to start with a letter - If empty → The capacitance is detached from voltage ports

CONT

`CONT, capa_name, contact_flag;`

Activates or deactivates contact elements at comb and plate capacitances

- Contact elements at comb and plate capacitances are automatically generated to prevent short circuits if the moving masses hit the fixed conductors.
- Contact parameters are defined in the Simulation Settings window of the GUI.
- Simulations are faster if contact elements at capacitances are deactivated. Contacts at stopper (STOP command) and z-motion limiters (ZLIM command) still exist.

Input Data

Parameter	Description
`capa_name`	[COMB](#) or [PCAP](#) capacitance reference name - Alphanumeric name, has to start with a letter - If empty, apply the `contact_flag` to all capacitances
`contact_flag`	Flag to activate or deactivate contact elements on capacitances - 1 OR true → Activate contacts (default) - 0 OR false → Deactivate contacts

CDMP

```
CDMP, dr;
```

Defines a constant damping ratio for harmonic simulations

- The constant damping ratio is only applicable in a harmonic response analysis. If the modal superposition solver (see MSUP command) is active, the damping ratio is applied to all modes (see MDMP command).
- The constant damping ratio will by superimposed by Rayleigh damping (see RDMP command).

Input Data

Parameter	Description
`dr`	Damping ratio - ≥ 0 → Default = 0

RDMP

```
RDMP, type, val1, val2, dr1, dr2;
```

Defines Rayleigh damping for harmonic and transient simulations

- Rayleigh damping is superimposed to the constant damping ratio (if the modal superposition solver is off) and to the modal damping ratios (if the modal superposition solver is on).
- Rayleigh damping is defined by the damping ratios at two characteristic frequencies. Negative alpha-beta-multiplies can be set to zero in the Simulation Settings window.

Input Data

Parameter	Description
`type`	Type of frequency specifications - freq → Frequencies are specified by numbers - mode → Frequencies are taken from eigenmodes
`val1`	- If `type` = freq → First frequency - If `type` = mode → First eigenmode
`val2`	- If `type` = freq → Second frequency - If `type` = mode → Second eigenmode
`dr1`	Damping ratio for the first frequency or eigenmode
`dr2`	Damping ratio for the second frequency or eigenmode

MDMP

```
MDMP, mode_nr, m_dr;
```

Defines modal damping ratios for harmonic or transient simulations

- The modal superposition solver must be active (see MSUP command). Different individual modal damping ratios can be assigned to eigenmodes.
- The modal damping ratios will by superimposed by Rayleigh damping (see RDMP command).

Input Data

Parameter	Description
`mode_nr`	Mode number - > 0 (integer) → Number of the eigenmode - all → Assigns the modal damping ratio to all modes
`m_dr`	Modal damping ratio - ≥ 0 → Default = 0

DPLO

DPLO, freq1, freq2, points, division;

Creates a plot of frequency dependent damping ratios defined by the RDMP command

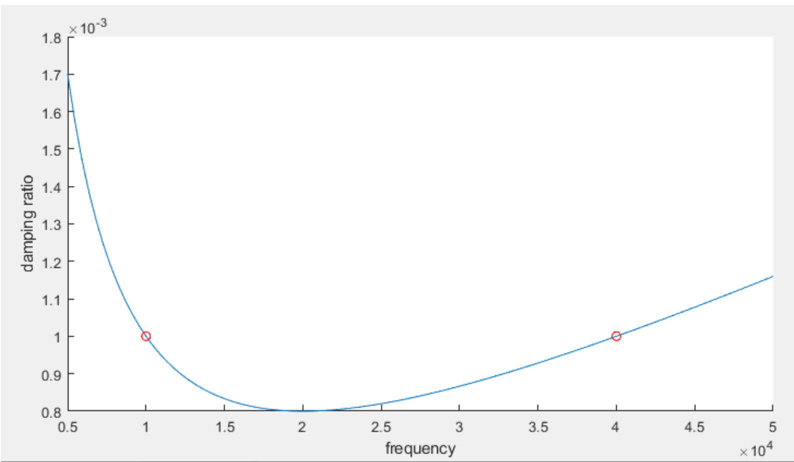
- The **DPLO** command must directly be entered after the RDMP command. The curve shows the damping ratio for all frequencies in the specified range. The damping ratios should not be negative (see Rayleigh damping).
- Damping ratios and frequencies of the RDMP command are marked by a red circle.

Input Data

Parameter	Description
freq1	Starting frequency - ≥ 0
freq2	Ending frequency - > freq1
points	Number of data points - ≥ 2
division	Type of the data point spacing - lin → Linear spacing - log → Logarithmic spacing (default)

Example:

```
RDMP,10e3,40e3,0.001,0.001;     % Rayleigh damping
DPLO,5e3,5e4,100,lin;           % Damping ratios verse frequency
```



MPLO

```
MPLO, mat_nr;
```

Create a plot of the orientation dependent orthotropic material properties

Input Data

Parameter	Description
mat_nr	Material reference number

BSEC

```
BSEC, sloc1, sloc2;
```

Provides cross-section properties of Timoshenko beam elements

- Result properties are the area of the cross-section, the area moments of inertia, the coordinates of the centroid, the warping and torsion constants, the coordinates of the shear center and the shear center correction factors.
- For non-uniform beams (tapered beams), properties are listed for the first connecting point (the order sclo1 and sloc2 can be changed to get results of the other end).

Input Data

Parameter	Description
sloc1	First spring connecting point (SLOC)
sloc2	Second spring connecting point (SLOC)

DRIV

```
DRIV, drive_mode, drive_freq, drive_ampl, node_type, node_nr,
dir_label;
```

Defines a drive mode for Coriolis effects in quasi-static simulations

- The drive mode is used to calculate a velocity vector for Coriolis forces and moments in quasi-static simulations.
- The drive mode amplitude is defined at a user defined degree of freedom on rigid bodies (rb) or master nodes (mn).
- The drive frequency is either the eigenfrequency of the mode or a user defined frequency value.

Input Data

Parameter	Description
drive_mode	Number of the drive mode - ≥ 1 [integer]
drive_freq	Frequency of drive mode - > 0 [real] - Frequency - If empty – eigenfrequency of the drive mode
drive_ampl	Amplitude of the drive mode at the specified DOF
node_type	Available node types (enter a two-character string) - rb → Rigid body center point (center of gravity) - mn → Master node
node_nr	Number of the associated rigid body or master node - ≥ 1 [integer]
dir_label	Available directions (enter a two-character string) → ux or rx → uy or ry → uz or rz

DCSW

```
DCSW, load_type, node_nr, dir_label, v_start, v_end, points, h-flag;
```

Defines a load parameter for DC-sweep simulations

- The response of the load is analyzed in a series of static simulations.
- The hysteresis flag activates the forward-backward mode of the sweep.

Input Data

Parameter	Description
load_type	Available node types (enter a string) - rb → Rigid body center point (center of gravity) - mn → Master node - sl → Spring location point (SLOC) - acel → Acceleration load - vscr → Voltage source - off → Deactivates all assigned sweep settings

node_nr	Number of the node associated with above node type - ≥ 1 [integer] for load_type = rb, mn, sl - → Unused for load_type = acel - [string] Name of voltage source for load_type = vscr
dir_label	Available load types and directions (enter a two-character string), unused for VSCR sources - Forces or moments: → fx or mx → fy or my → fz or mz - Displacements, rotations (in radian) or accelerations: → ux or rx → uy or ry → uz or rz
v_start	Value of load at start of DC-sweep
v_end	Value of load at the end of the DC-sweep
points	Number of sample points
h-flag	Activates/deactivates the forward-backward mode - If 1 or true → Forward-backward mode - If 0 or false → Only forward-mode (default)

MOPT

```
MOPT, eigenmodes;
```

Number of eigenmodes to be extract during a modal analysis

- The default number of modes is set to 6.

Input Data

Parameter	Description
eigenmodes	Number of eigenmodes to be analyzed

HARF

```
HARF, freq1, freq2, points, spacing;
```

Defines the frequency range for a harmonic response analysis

- The command defines the start and end frequency, the number of data points within this range and the type of data point spacing in a harmonic response analysis.

Input Data

Parameter	Description
freq1	Start of the frequency range - ≥ 0
Freq2	End of the frequency range - > freq1
points	Number of data points (frequencies) to be simulated - ≥ 2
spacing	Type of data point spacing - lin → Linear spacing - log → Logarithmic spacing (default)

TIME

```
TIME, sim_time, tstep;
```

Defines the total simulation time and the time-step size for transient simulations

- The specified time-step size is reduced if contact elements tend to close.

Input Data

Parameter	Description
sim_time	Total simulation time - > 0
tstep	Maximum time-step size - > 0 - < sim_time

SOLV

```
SOLV, antype, simulink_flag;
```

Starts a specified simulation

- Loads, displacement constraints and simulation settings must be defined prior starting the simulation run.

Input Data

Parameter	Description
antype	Available types of analysis: - moda → Starts a modal analysis - stat → Starts a static simulation or a DC-sweep - harm → Starts a harmonic simulation (AC-sweep) - tran → Starts a transient simulation
simulink_flag	Used for transient simulations only - 1 or true → Skip transient solution procedure, prepare model items for the SIMULINK export - 0 or false → Start the transient simulation (default)

SIML

```
SIML, load_step;
```

Builds a Simulink Model of the specified load step

- The specified load step must be a transient simulation.

Input Data

Parameter	Description
load_step	Number of the load step to be exported

4 Post-processing commands

SNOD

SNOD, node_type, node_nr, dir_label, del_flag;

Defines nodal results of the mechanical domain for plotting or tabular lists

- Results with multiple samples (DC-sweep, AC-sweep and transient data) are plotted as curves in the “Data Plot” window. Related curves can be exported in several formats.
- Results of a static or modal analysis are listed in a special result variables window.

Input Data

Parameter	Description
node_type	Available node types (enter a two-character string) - rb → Rigid body center point (center of gravity) - mn → Master node (MAST) - sl → Spring location point (SLOC)
node_nr	Number of the node associated with above node type - ≥ 1 [integer]
dir_label	Available load types and directions (enter a two-character string) - Forces or moments: → fx or mx → fy or my → fz or mz - Displacements, rotations (in radian): → ux or rx → uy or ry → uz or rz
del_flag	Delete previous defined nodal solutions → del – if node_type & node_nr & dir_label empty – delete all nodal solutions → del - else → delete a specific nodal solution

SCON

```
SCON, type, cond_port1, cond_port2, del_flag;
```

Defines nodal results of the electrostatic domain for plotting or tabular lists

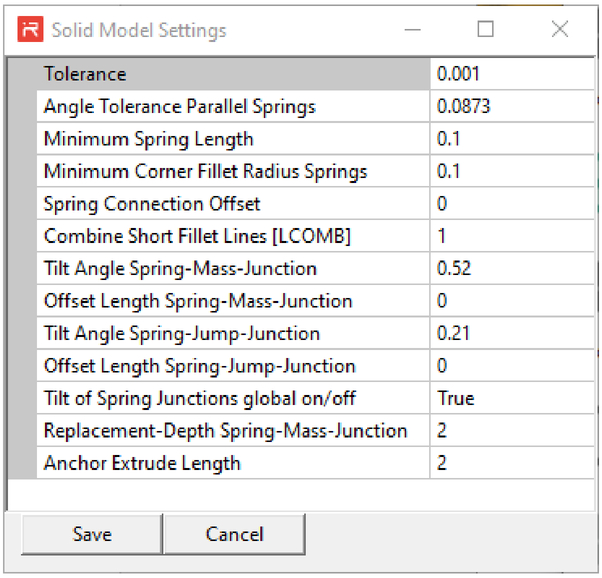
- Results with multiple samples (DC-sweep, AC-sweep and transient data) are plotted as curves in the “Data Plot” window. Related curves can be exported in several formats.
- Results of a static or modal analysis are listed in a special result variables window.

Input Data

Parameter	Description
type	Available result types (enter a 4-character string) - capa → Capacitances - volt → Voltages - curr → Current
cond_port1	First conductor port name
cond_port2	Second conductor port name, if type = capa
del_flag	Delete previous defined nodal solutions → del – if type empty → Delete all electrical solutions → del – else → Delete a specific electrical solution

Part F - Parameters and Settings of the GUI

1 Solid Model Settings window



Tolerance

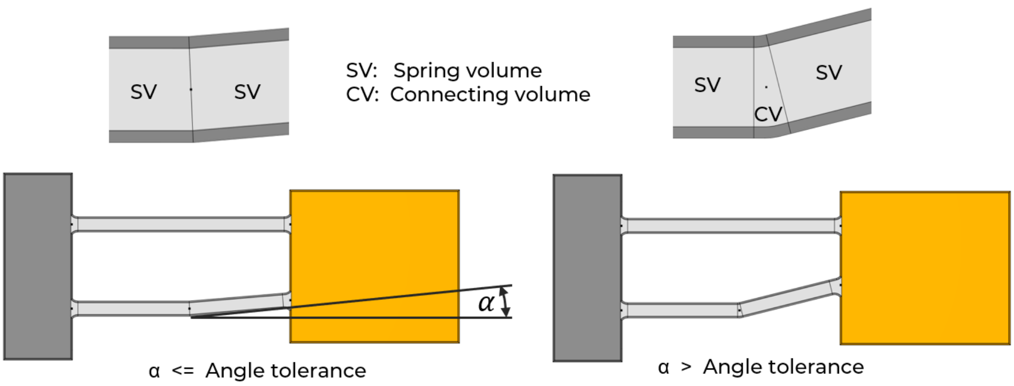
Specifies the tolerance for building solid model items and Boolean operations

- Solid model items within this tolerance are considered coincident.
- > 0, default = 0.001 µm (Units of length, e.g. µm)

Angle Tolerance Parallel Springs

Specifies the angle between spring elements to create connecting volumes

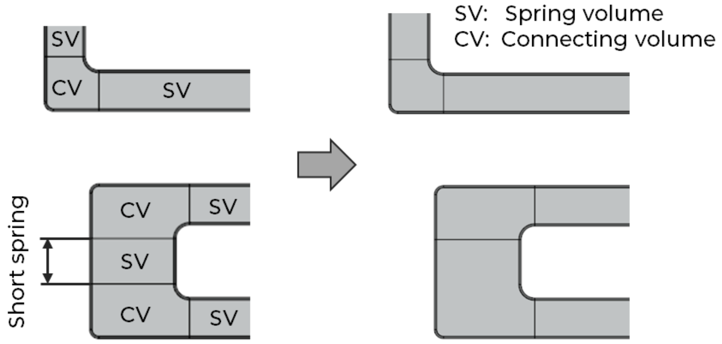
- A connecting volume appears between spring elements if the angle is larger as the specified value.
- > 0, default = 0.0873 (angle in radian; about 5 °)



Minimum Spring Length

Specifies the minimal spring length of the model

- Corner fillets and process data settings can create springs of very short length which cause meshing problems for the finite element model export to ANSYS® and COMSOL® .
- Spring volumes (SV) with a short length are combined with an adjacent connecting volume (CV).
- ≥ 0, default = 0.1 (Units of length, e.g. µm)

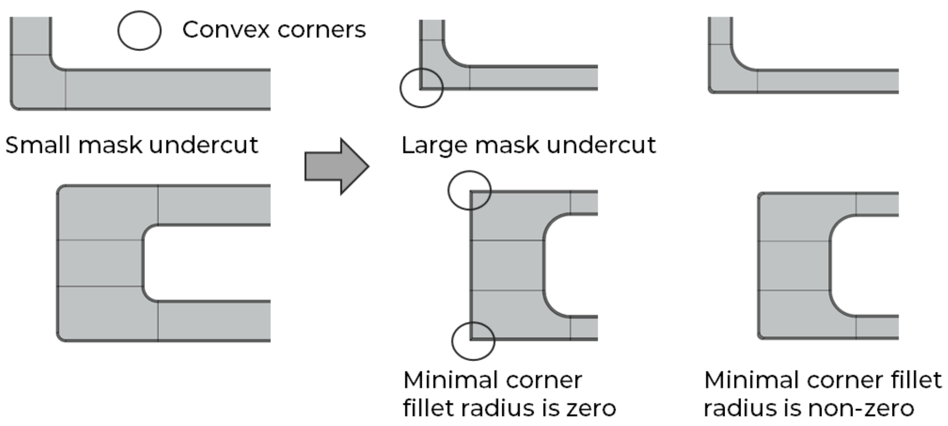


The short spring is smaller as the Minimal Spring Length parameter.

Minimum Corner Fillet Radius Springs

Specifies the minimal corner fillet radius of the model

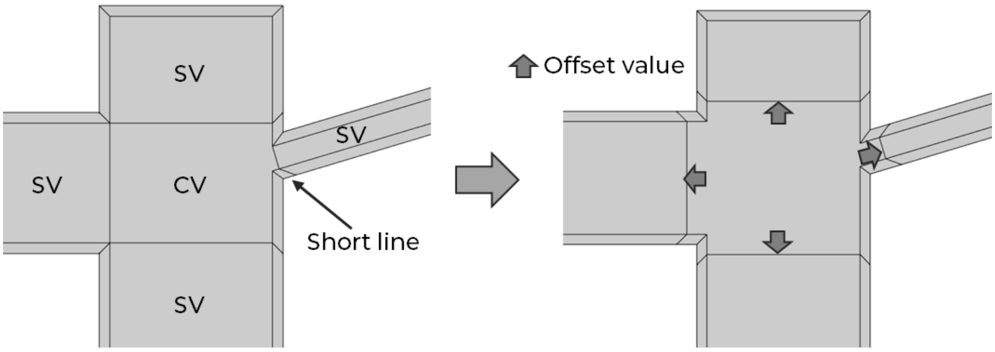
- The specified corner fillet radius in the [Process Data Setting](#) window is enlarged or reduced by the mask undercut settings. Concave corners get larger radiuses and convex corners get smaller radiuses. If the undercut is bigger as the corner radius, are sharp tip appears at the convex corners.
- A minimal corner radius can be defined to force corner fillets on all spring elements.
- > 0, default = 0.1 (Units of length, e.g. µm)



Spring Connection Offset

Extends the size of the spring connection volumes in spring directions

- Connecting volumes can obtain short lines if complicated spring designs are linked at one connecting point and corner fillets are switched off.
- The connecting volumes can be slightly extended by a spring connection offset to improve the mesh quality for a finite element model export to ANSYS® or COMSOL®.
- ≥ 0, default = 0

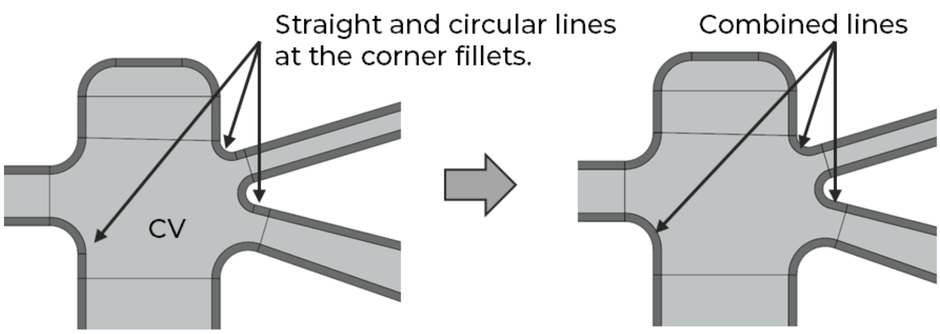


The connecting volumes is extended by the spring connection offset.

Combine Short Fillet Lines

Combines circular and straight lines at the fillet of spring connecting volumes

- Corner fillets at opposite sides of the springs can have different sizes in case of asymmetric structures. Straight and circular or elliptical lines appear in the corner region.
- Both lines can be combined to improve the finite element mesh quality in ANSYS® and COMSOL®.
- The parameter defines the length of straight lines which are combined with circular lines.
- ≥ 0, default = 1 (Units of length, e.g. µm)



Short lines are combined with circular lines at the connecting volume.

Tilt Angle Spring-Mass-Junction

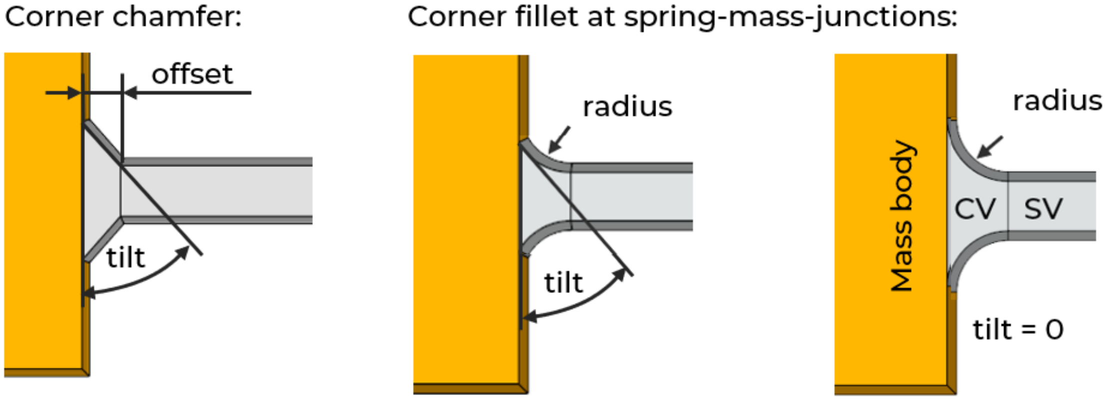
Defines the tilt angle at spring-mass-junctions

- The interface between springs and masses can be designed by an additional connection volume with a straight chamfer or a circular corner fillet to avoid stress concentrations (see “Offset Length Spring-Mass-Junction”).
- 0 ≤ tilt_angle < π/2 [rad], default = 0.52 (about 30°)

Offset Length Spring-Mass-Junction

Defines the offset length at spring-mass-junctions

- The offset length is necessary for a straight bond chamfer.
- ≥ 0, default = 0



tilt	Tilt angle at spring-mass-junctions
offset	Offset length at spring-mass-junctions
radius	Corner fillet radius spring-mass-junctions from the ***Process Data*** window

Tilt Angle Spring-Jump-Junction

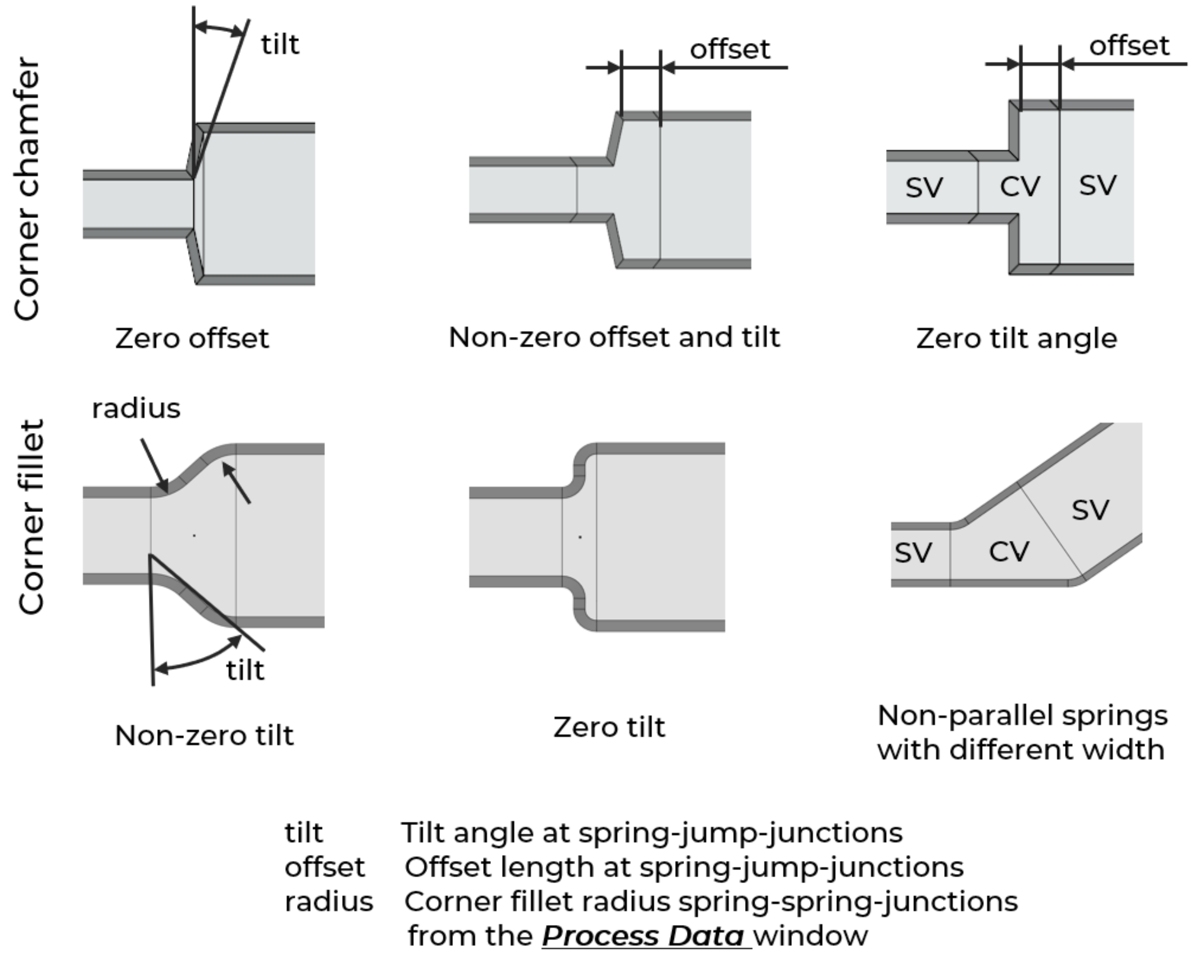
Defines the tilt angle at spring-jump-junctions (sudden change of the spring)

- The interface between springs with different beam width (spring-jump-junction) is designed by an additional connection volume with a straight chamfer or a circular corner fillet to avoid stress concentrations.
- 0 ≤ tilt_angle < π/2 [rad], default = 0.21 (about 12°)

Offset Length Spring-Jump-Junction

Defines the offset length at spring-jump-junctions

- The offset length defines the extension of the connecting volume in beam axis direction for straight chamfer.
- ≥ 0, default = 0



Tilt of Spring Junctions global on/off

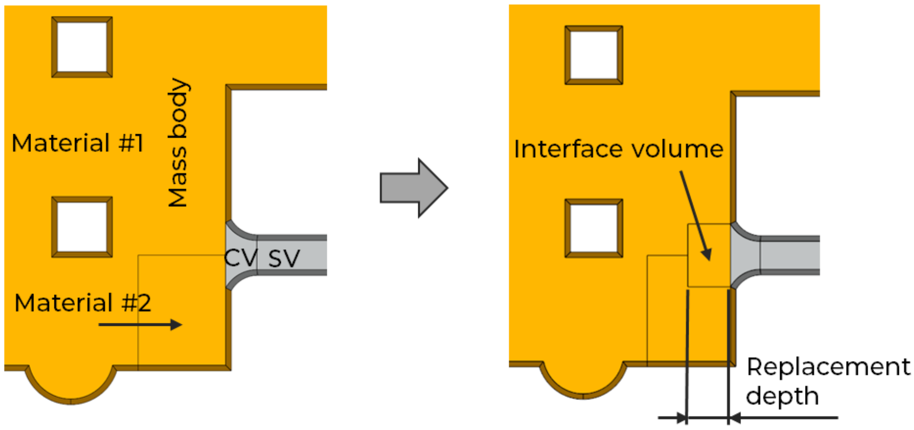
Activates or deactivates the tilt angle at spring-mass- and spring-jump-junctions for subsequent “Build Solids” operations

- Default = true

Replacement-Depth Spring-Mass-Junction

Defines the depth of an interface volume at the connection of mass bodies to spring elements

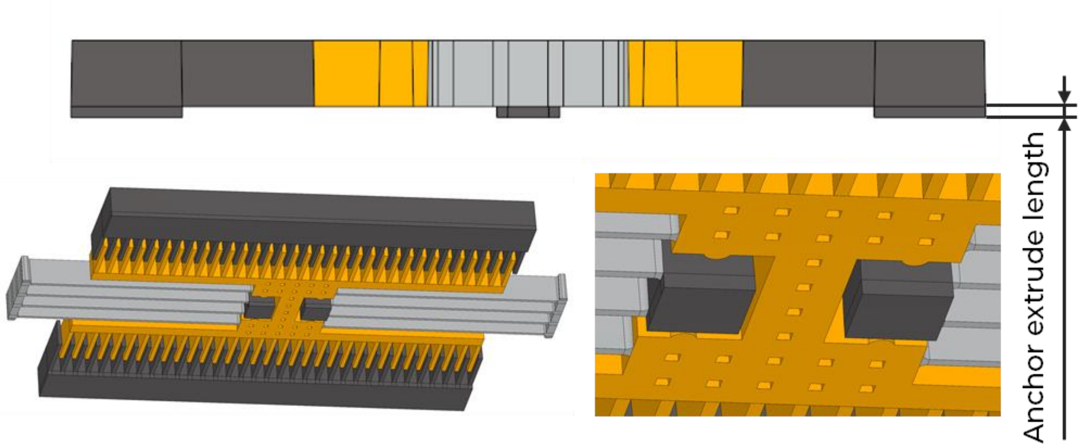
- Additional interface volumes (IV) at masses are necessary for bodies which consist of two or more material domains at the interface region to springs.
- Different material domains are separated by internal lines in mass bodies. As a consequence, attached springs are difficult to meshed with hexahedral elements after the finite element model export to ANSYS® or COMSOL®.
- If the replacement depth parameter is non-zero, an additional volume with the specified depth appears at the interface and allows for hexahedral elements in adjacent springs.
- ≥ 0, default = 2 µm



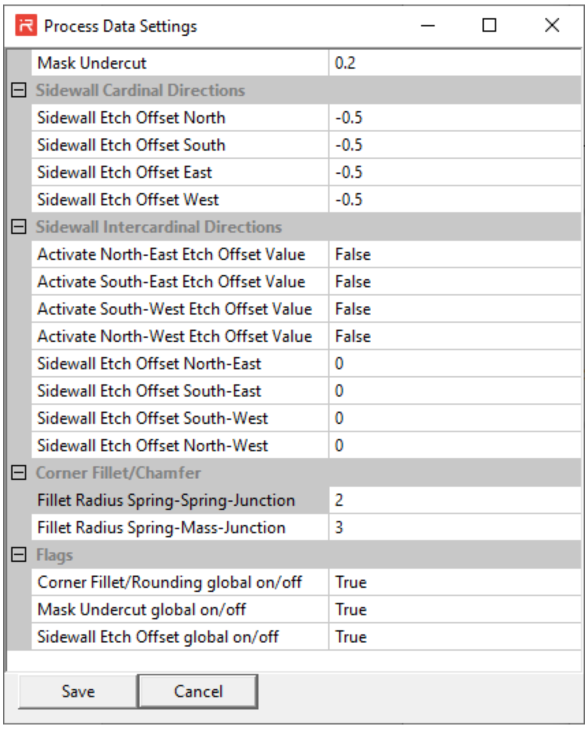
Anchor Extrude Length

Defines the length of the anchor extrusion to the substrate surface

- The anchor extrude length is the distance between the bottom face of the functional layer and the substrate surface (thickness of sacrificial layer).
- ≥ 0, default = 2 (Units of length, e.g. µm)



2 Process Data Settings window



Mask Undercut

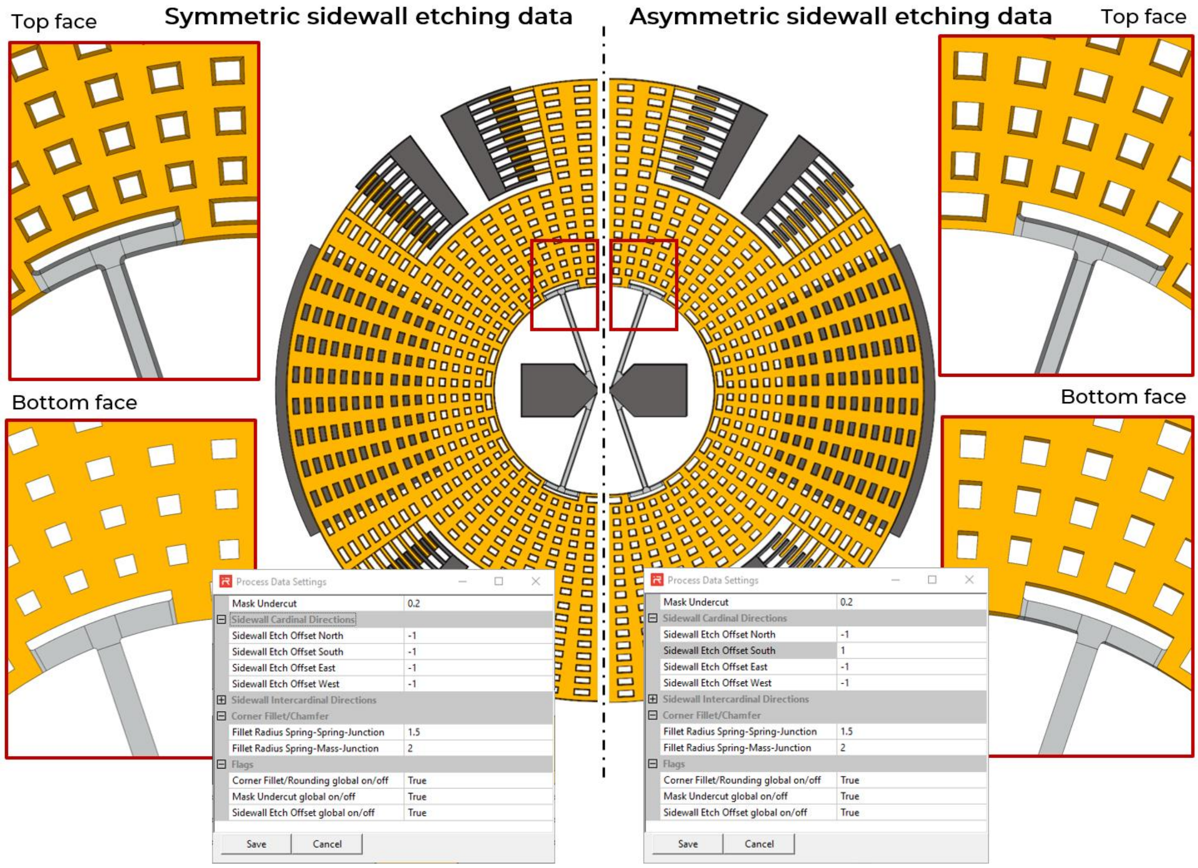
Defines the mask undercut of the functional layer

- If the mask undercut value is positive, all outer edges of the functional layer shrink according to the specified value. Negative values enlarge the structure which is supported by the tool but there is no physical reason.
- Default = 0 (Units of length, e.g. µm)

Sidewall Etch Offset North, South, East and West

Defines the sidewall etch offset of the functional layer

- The sidewall etching parameters define the offset of the bottom edge compared to the edge at the top face of the functional layer. Positive values make the functional layer at the bottom edge smaller and negative values wider.
- Different values can be assigned to faces pointing “north”, “south”, “east” and “west”. Interpolations are applied for faces pointing in other directions except values for “north-east”, “south-east”, “south-west” and “north-west” have been set.
- Specific etch offsets can also be assigned to individual springs (see SPRI command).
- Default = 0 (Units of length, e.g. µm)



Sidewall Etch Offset North-East, South-East, South-West ...

Defines the sidewall etch offset of the functional layer

- Sidewall etching parameters “north”, “south”, “east” and “west” are interpolated in all other directions. Additional sidewall etching parameters can be set for “north-east”, “north-west”, “south-east” and “south-west” which allow for complicated etching profiles.
- Default = deactivated

Fillet Radius Spring-Spring-Junction

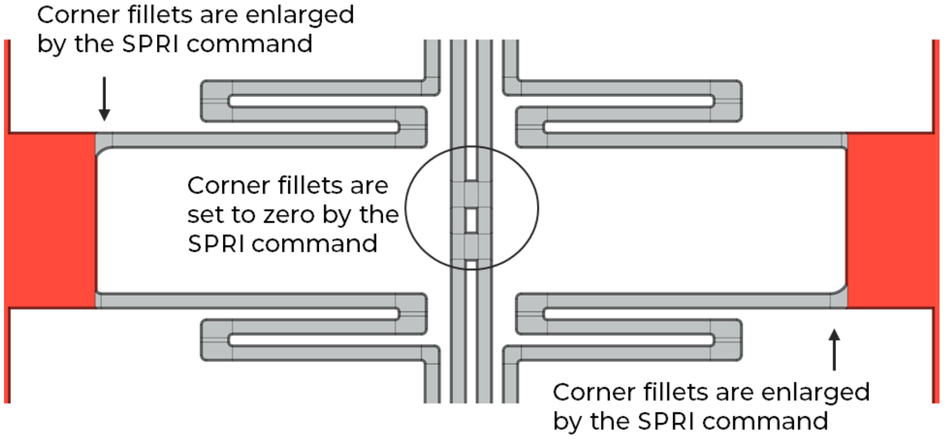
Defines the radius of corner fillets at spring-spring-junctions

- The rounding radius is applied to all spring-spring-junctions except specific values have been set to spring elements by the SPRI command.
- The rounding radius is defined for the nominal layout. The corner fillets can be deactivated in the Simulation Settings window.
- Mask undercut and sidewall etch offsets change the radius and create elliptical lines for asymmetric process data settings.
- ≥ 0, default = 0 (Units of length, e.g. µm)

Fillet Radius Spring-Mass-Junction

Defines the radius of corner fillets at spring-mass-junctions

- The rounding radius is applied to all spring-mass-junctions (including anchors) except specific values have been set to spring elements by the SPRI command.
- The rounding radius is defined for the nominal layout. The corner fillets can be deactivated in the Simulation Settings window. If there is not enough space at mass bodies or anchors, the fillets are automatically removed.
- Mask undercut and sidewall etch offsets change the radius and create elliptical lines for asymmetric process data settings.
- ≥ 0, default = 0 (Units of length, e.g. µm)



Corner Fillet/Rounding global on/off

Activates or deactivates the corner fillets at spring-mass- and spring-jump-junctions for subsequent “Build Solids” operations

- Default = on

Mask Etching global on/off

Activates or deactivates the “Mask Undercut” for subsequent “Build Solids” operations

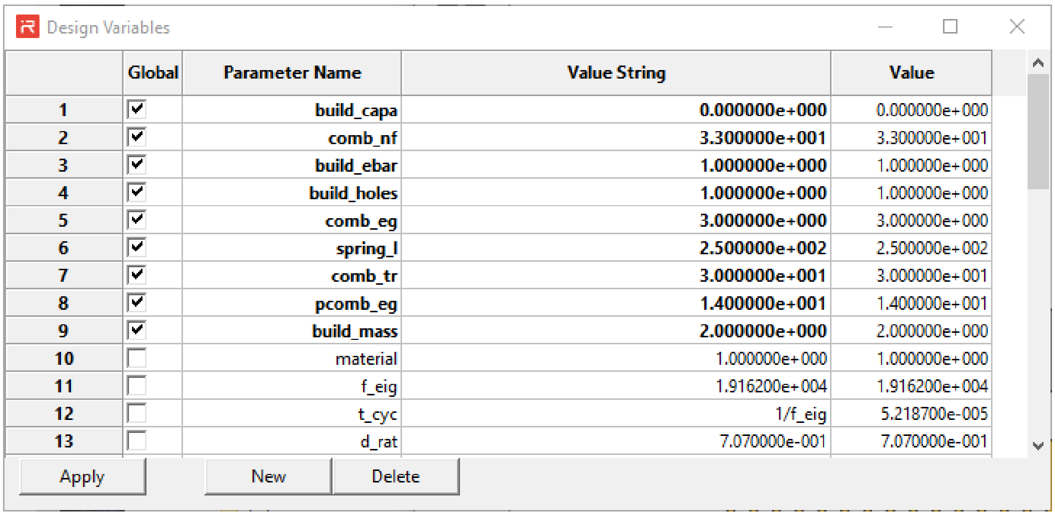
- Default = on

Sidewall Etching global on/off

Activates or deactivates the “Sidewall etching” values for subsequent “Build Solids” operations

- Default = on

3 Design Variables window


Design Variables

	Global	Parameter Name	Value String	Value
1	☑	build_capa	0.000000e+000	0.000000e+000
2	☑	comb_nf	3.300000e+001	3.300000e+001
3	☑	build_ebar	1.000000e+000	1.000000e+000
4	☑	build_holes	1.000000e+000	1.000000e+000
5	☑	comb_eg	3.000000e+000	3.000000e+000
6	☑	spring_l	2.500000e+002	2.500000e+002
7	☑	comb_tr	3.000000e+001	3.000000e+001
8	☑	pcomb_eg	1.400000e+001	1.400000e+001
9	☑	build_mass	2.000000e+000	2.000000e+000
10	☐	material	1.000000e+000	1.000000e+000
11	☐	f_eig	1.916200e+004	1.916200e+004
12	☐	t_cyc	1/f_eig	5.218700e-005
13	☐	d_rat	7.070000e-001	7.070000e-001

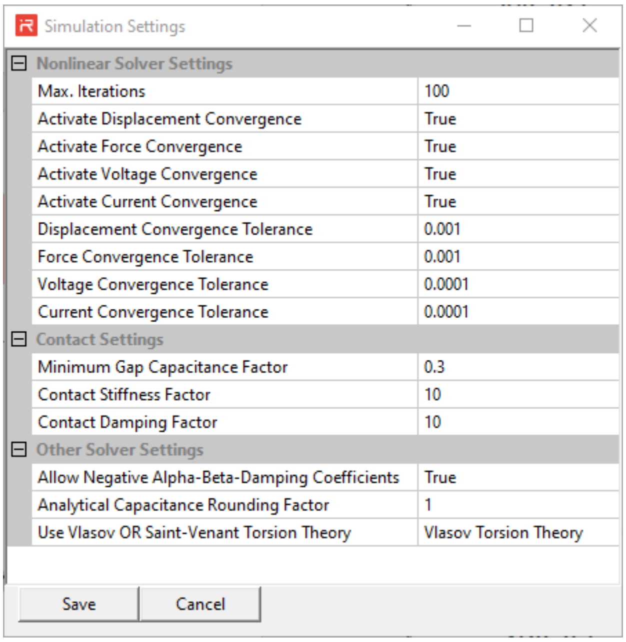
Apply New Delete

Design Variables

Assigns design variables for cases studies and optimization cycles

- User defined parameters (see PARA command) defined in the model input file can be modified in the Design Variables window. Those parameters must be marked in the first column to become a “global” parameter. Then, the parameter value can be changed in the value string column.
- “Global” design variables have a higher priority as parameters in the user model file and substitute values for case studies and optimization cycles.
- “NEW” design variables must be assigned “global”. The values will be applied after “Build solids”.
- A new solid model must be built after changing the values. “Global” design parameters appear at the top of the design variables list. The order of global design variables can be manually changed on the Project file.

4 Simulation Settings window



Max. Iterations

Specifies the maximum number of equilibrium iterations for nonlinear simulations

- Default = 100

Activate Displacement Convergence

Activates displacements (translational and rotational DOF) for convergence tolerance monitoring in nonlinear simulations

- Default = on

Activate Force Convergence

Activates forces for convergence tolerance monitoring in nonlinear simulations

- The residual vector between applied forces and restoring forces will be evaluated in the implemented Newton-Raphson method.
- Default = on

Activate Voltage Convergence

Activates voltages for convergence tolerance monitoring in nonlinear simulations

- Default = on

Activate Current Convergence

Activates currents for convergence tolerance monitoring in nonlinear simulations

- The residual vector between external and internal current will be evaluated in the implemented Newton-Raphson method.
- Default = on

Displacement Convergence Tolerance

Defines the displacement convergence tolerance for nonlinear simulations

- The infinite norm (maximum absolute row sum) of the displacement increment vector is compared with the infinite norm of the accumulated displacement vector. Convergence is achieved if the ratio of both norms is smaller as the specified tolerance.
- Default = 0.001

Force Convergence Tolerance

Defines the force convergence tolerance for nonlinear simulations

- The L2-norm (square root of the sum of the squares) of the force residual vector is compared with the L2-norm of the applied force vector. Convergence is achieved if the ratio of both norms is smaller as the specified tolerance.
- Default = 0.001

Voltage Convergence Tolerance

Defines the voltage convergence tolerance for nonlinear simulations

- The same approach as used for displacements.
- Default = 0.0001

Current Convergence Tolerance

Defines the current convergence tolerance for nonlinear simulations

- The same approach as used for forces.
- Default = 0.0001

Minimum Gap Capacitances Factor

Defines the minimal gap for contact elements at comb and plate capacitors

- The gap change of comb and parallel plate capacitances is limited by contact element.
- Above factor specifies the remaining gap at which contact elements become effective.
- The default value is 0.3 what is 30 % of the initial electrode gap.

Contact Stiffness Factor

Defines the default contact stiffness factor of contact elements

- The default contact stiffness factor is used for comb and plate capacitances, for in-plane stopper elements (see STOP command) and out-of-plane contacts (see ZLIM commands).
- The contact stiffness factors defined by STOP or ZLIM commands have a higher priority.
- Default = 10 (Units of forces/units of length)

Contact Damping Factor

Defines the default contact damping factor of contact elements

- The default contact damping factor is used for comb and plate capacitances, for in-plane stopper elements (see STOP command) and out-of-plane contacts (see ZLIM commands).
- The contact damping factors defined by STOP or ZLIM commands have a higher priority.
- Default = 10

Allow Negative Alpha-Beta-Damping Coefficients

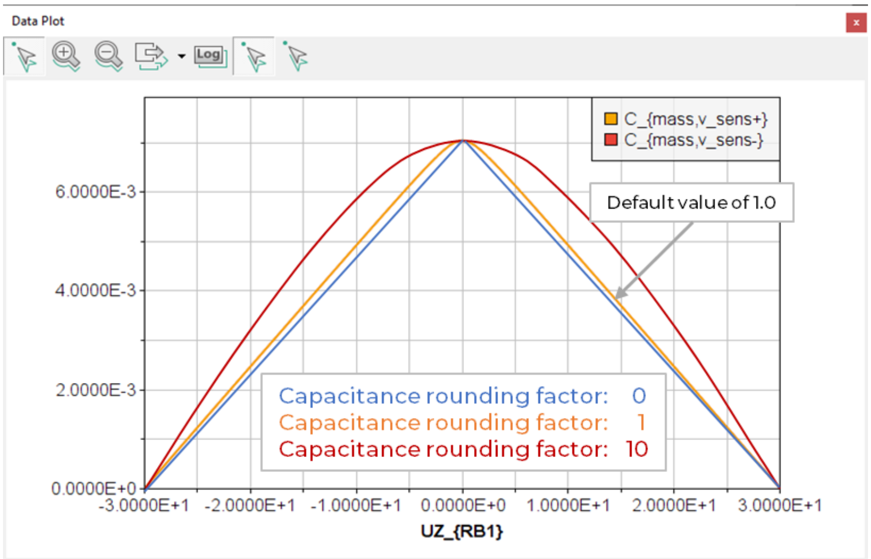
Allows negative alpha-beta-damping multipliers for Rayleigh-Damping

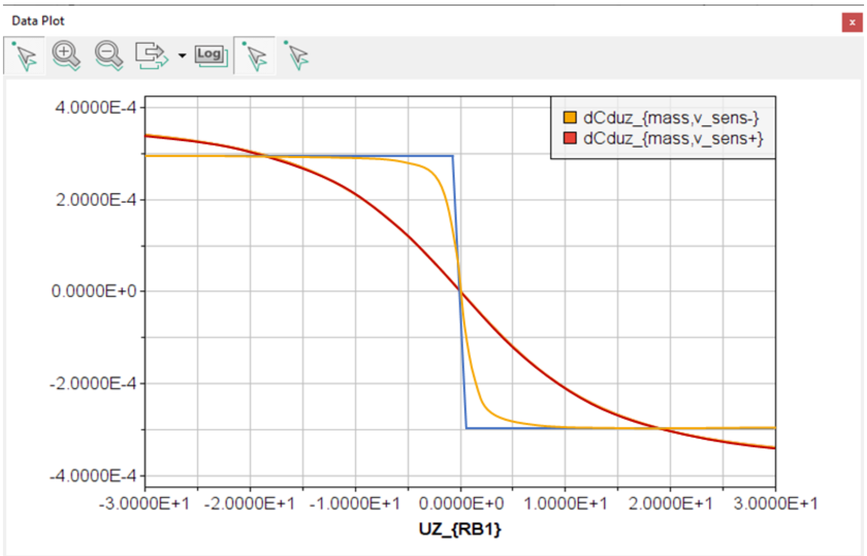
- Negative alpha-beta-damping multipliers can occur for some settings of the RDMP command.
- If the flag is deactivated, negative alpha and beta coefficients are set to zero.
- Default = on

Analytical Capacitance Rounding Factor

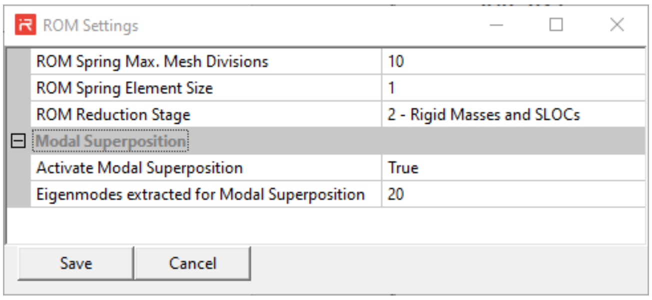
Defines a capacitance rounding factor for out-of-plane motion of comb-cells

- Capacitances of comb cells consider the quasi-homogeneous field between conductors (fringing field is ignored).
- The capacitance-stroke-relationship for out-of-plane motion components leads to linear functions for positive and negative uz-displacements. The function is continuous but not differentiable at the initial position. A small rounding at the peak values has been introduced to solve the problem.
- A capacitance rounding factor of 1 smoothens the capacitance-stroke-function as shown by the orange curve. Larger values increase and smaller values decrease the rounding range independent from other dimensions.
- > 0, default = 1





5 ROM Settings window



ROM Spring Max. Mesh Divisions

Defines the maximum number of beam elements used for suspension springs

- A very small number of beam elements is usually sufficient to calculate mechanical properties.
- A larger number is often applied to plot smooth bending lines of deformed springs for documentations.
- Default = 10

ROM Spring Element Size

Defines the default length of beam elements in suspension springs

- The number of elements can be changed for individual springs by the SPRI command.
- Default = 1 (Units of length, e.g. µm)

ROM Reduction Stage

Defines the default reduced-order-model (ROM) reduction stage for simulations

- Possible settings are: Consider all motion DOF (0), consider only rigid body motion DOF (1) or consider rigid body and spring connecting points (2) for simulations.
- The reduction stage can be changed by the REDS command for subsequent simulation runs.
- Default = 2

Activate Modal Superposition

Activates the modal superposition solver (MSUP) for simulations

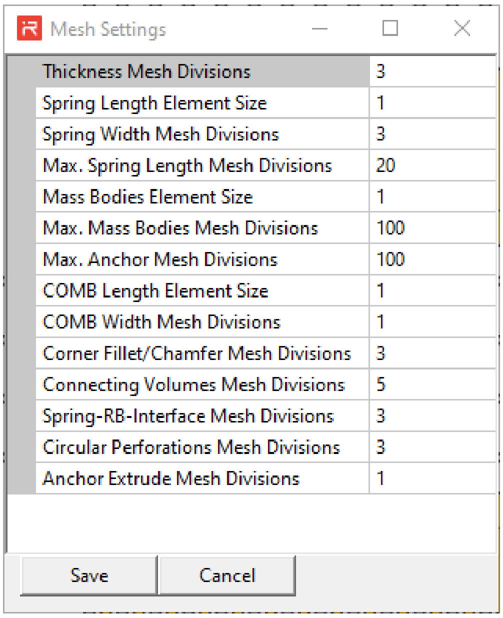
- The modal superposition solver can also be activated by the MSUP command for subsequent simulation runs.
- Default = on

Eigenmodes extracted for Modal Superposition

Specifies the number of eigenmodes for the modal superposition solver

- The lowest modes in the specified set will be basis functions of the simulation model.
- Default = 20

6 Mesh Settings window



Thickness Mesh Divisions

Defines the number of finite elements for the thickness of the functional layer

- Default = 3

Spring Length Element Size

Defines the element size for meshing of springs in length direction

- Default = 1 (Units of length, e.g. 1 µm)

Spring Width Mesh Divisions

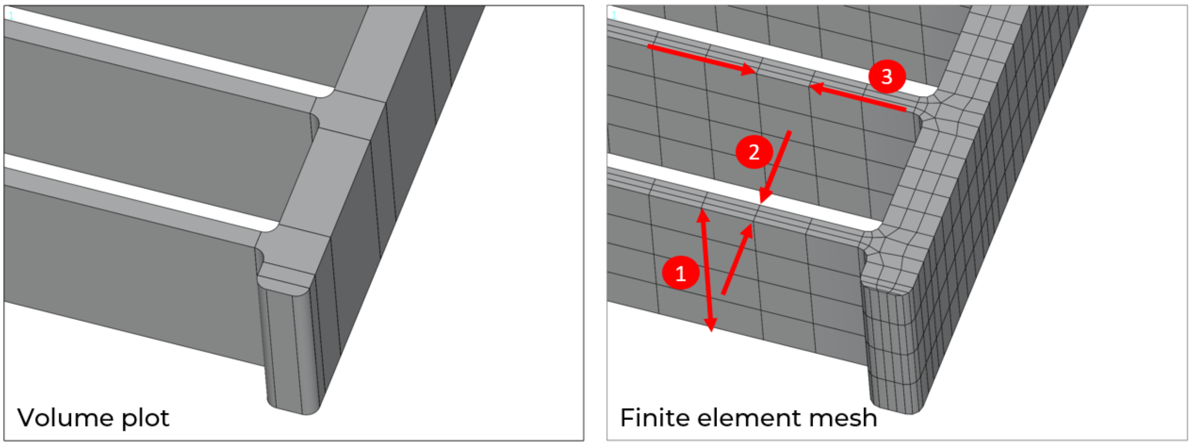
Defines the number of elements for the spring width

- The value can be changed for individual springs by the SPRI command.
- Default = 3

Max. Spring Length Mesh Divisions

Defines the maximum number of elements for the spring length

- The value can be changed for individual springs by the SPRI command.
- Default = 20



1 **Thickness mesh division (4)**

2 **Spring width mesh division (3)**

3 **Spring element size** (8 µm, not more than **Max. Spring Length Mesh Division** elements)

Mass Bodies Element Size

Defines the element size for meshing of seismic masses and anchors

- Default = 1 µm (Units of length, e.g. 1 µm)

Max. Mass Bodies Mesh Divisions

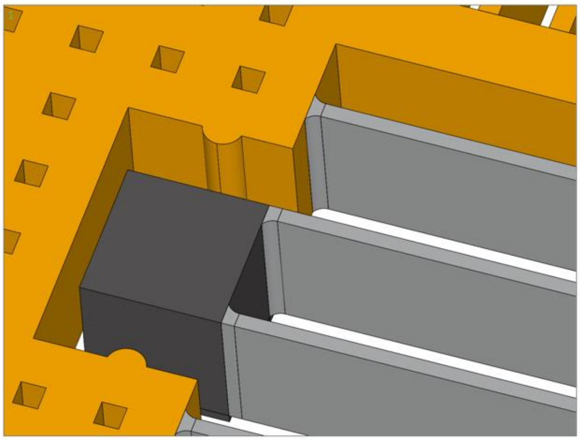
Defines the maximum number of elements along boundary lines of mass bodies

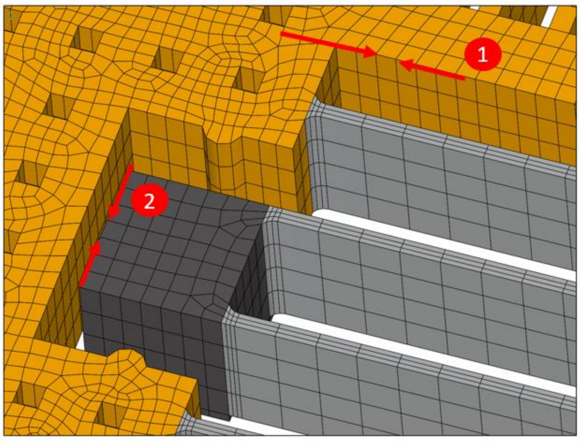
- Default = 100

Max. Anchor Mesh Divisions

Defines the maximum number of elements along boundary lines of anchors

- Default = 100





1 **Mass bodies element size (6 μm)**

Boundary lines of masses may not have more than **Max. Mass Mesh Division** elements

2 **Anchor element size (6 μm)**

Boundary lines of anchors may not have more than **Max. Anchor Mesh Division** elements

Comb Length Element Size

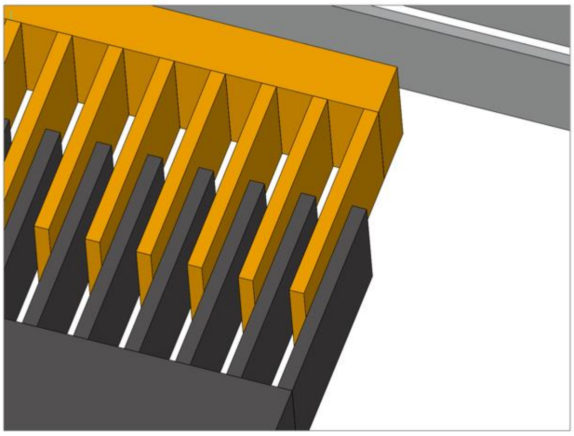
Defines the element size for meshing of comb fingers in length

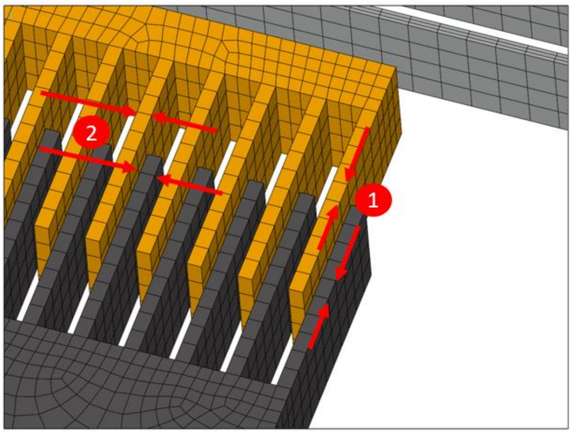
- Default = 1 (Units of length, e.g. 1 μm)

Comb Width Mesh Divisions

Defines the number of elements for the finger width of comb cells

- Default = 1





1 **Comb element size (5 μm)**

2 **Comb width mesh divisions (1)**

Corner Fillet/Chamfer Mesh Divisions

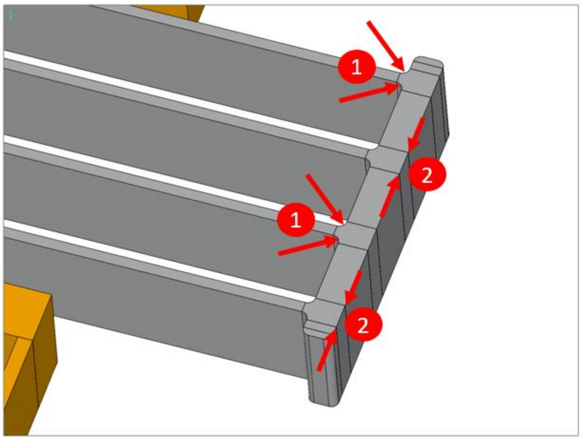
Defines the number of elements for the corner fillet or chamfer mesh divisions

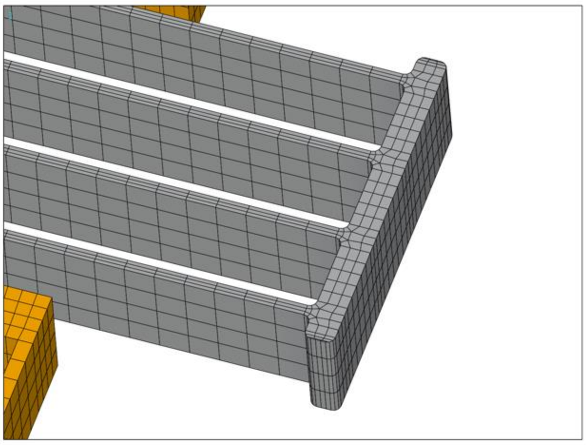
- Default = 3

Connecting Volumes Mesh Divisions

Defines the number of elements along straight lines of connecting volumes

- Default = 5





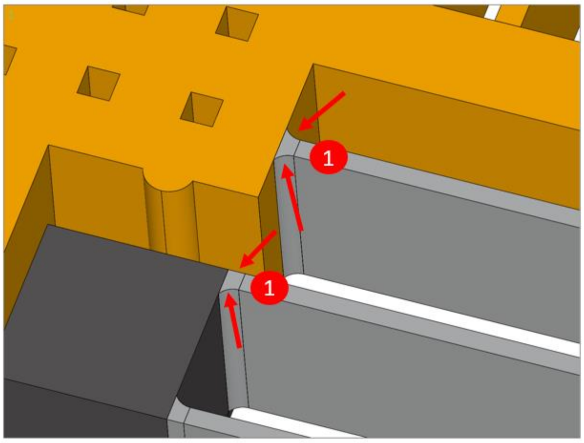
1 Corner fillet/chamfer mesh divisions (3)

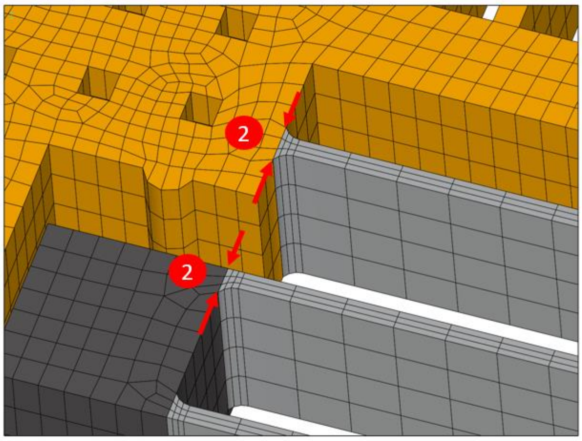
2 Connecting volume mesh divisions (3)

Spring-RB-Interfaces Mesh Divisions

Defines the number of elements at the interface between springs and rigid bodies

- The spring-RB-interface mesh division can only be set for models with corner chamfer or corner fillets. In all other cases, the “Spring Width Mesh Divisions” will be used for the interface to guarantee hexahedral elements in suspension springs.
- Default = 3





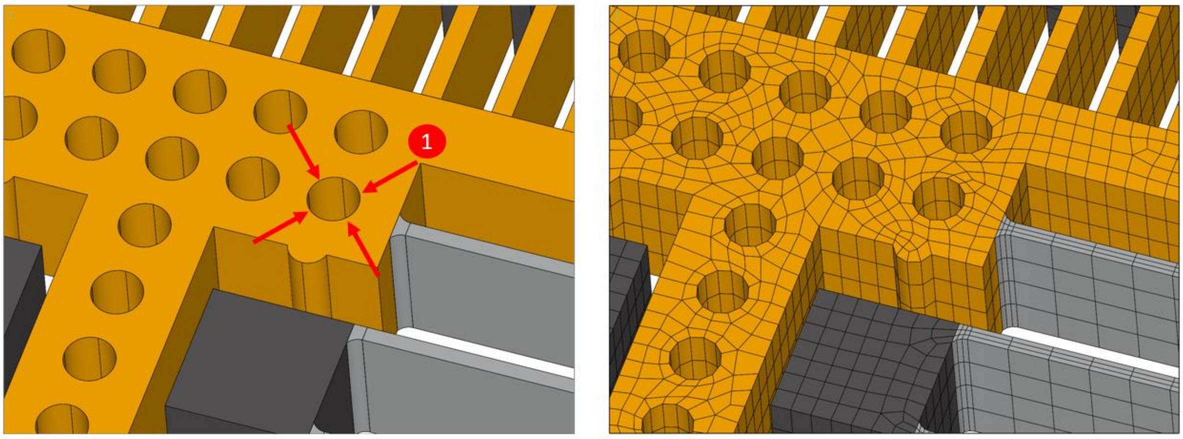
1 Corner fillet/chamfer mesh divisions (3)

2 Spring-RB-interfaces mesh divisions (3)

Circular Perforations Mesh Divisions

Defines the minimal number of elements at circular perforation patterns

- Circular perforations are defined by 4 circular lines which span a 90 degrees arc.
- The specified number of elements is applied to each 90° arc.
- Default = 3

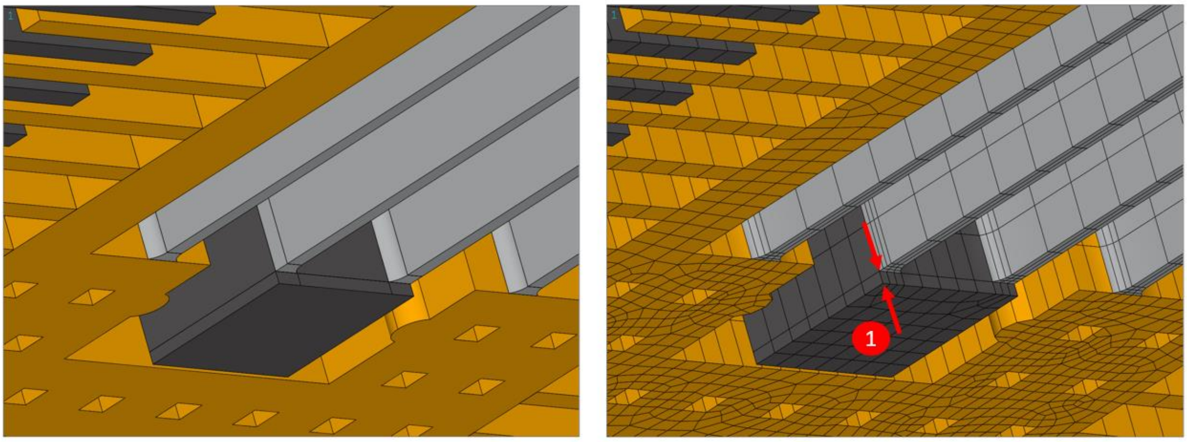


1 **Circular perforations mesh divisions** (3, the number is specified for a 90° arc, in total appear 12 element around the circle**)**

Anchor Extrude Mesh Divisions

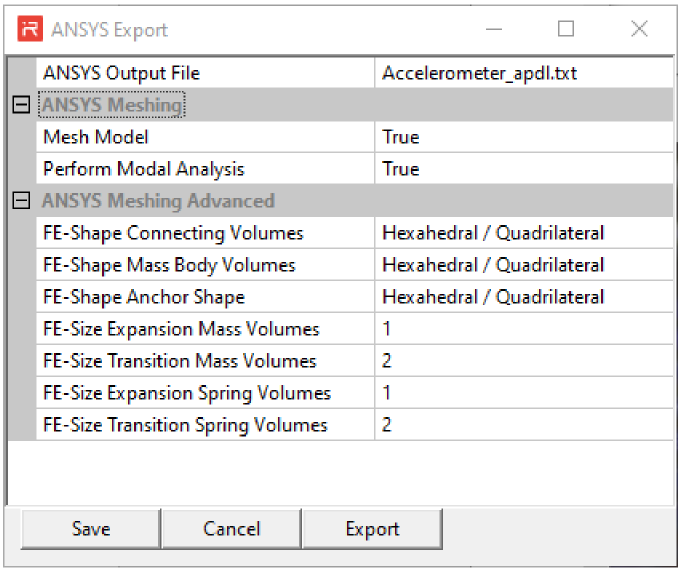
Defines the number of elements for the anchor extrusion to the substrate surface

- Default = 1



1 **Anchor extrude mesh divisions (1)**

Further settings are specific to the ANSYS® finite element model export:



FE-Shape Connecting Volumes

Defines the shape of the finite elements for spring connecting volumes

- Either “Hexahedral/Quadrilateral” or “Tetrahedral/Triangular” finite elements can be selected.
- Default = “Hexahedral/Quadrilateral”

FE-Shape Mass Body Volumes

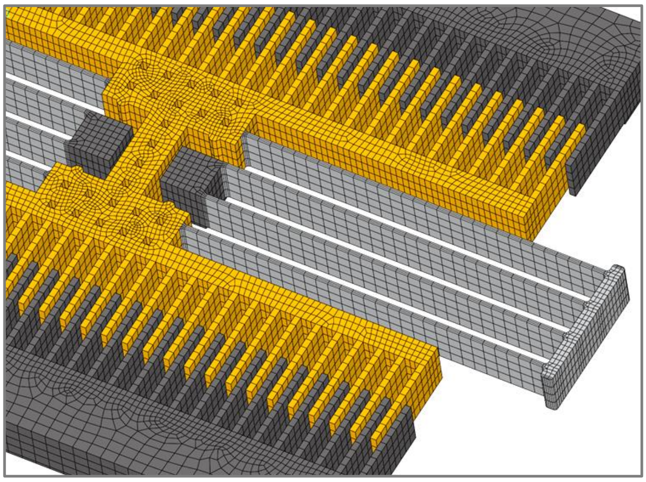
Defines the shape of the finite elements for volumes of mass bodies

- Either “Hexahedral/Quadrilateral” or “Tetrahedral/Triangular” finite elements can be selected.
- Default = “Hexahedral/Quadrilateral”

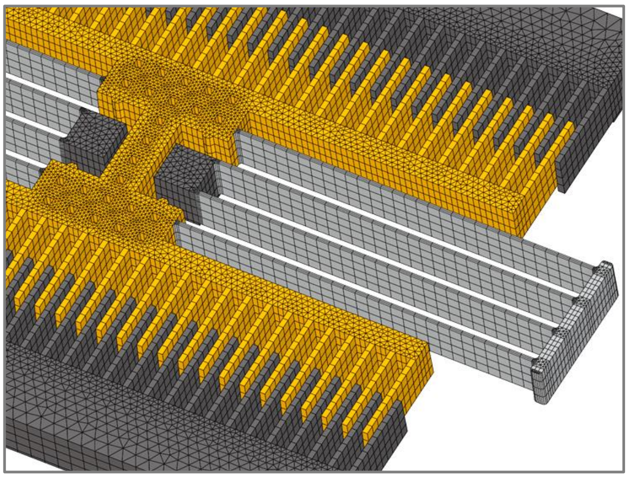
FE-Shape Anchor Volumes

Defines the shape of the finite elements for volumes of anchors

- Either “Hexahedral/Quadrilateral” or “Tetrahedral/Triangular” finite elements can be selected.
- Default = “Hexahedral/Quadrilateral”



- Hexahedral finite elements for all structural items (default)



- Hexahedral finite elements for springs and comb fingers
- Prism shaped elements are used for connecting volumes, masses and anchors

FE-Size Expansion Mass Volumes

Defines the size of internal finite elements compared to the size at boundary lines

- Elements at the interior domain can either contract (factor is smaller than 1) or expand (factor is larger than 1) compared to the element size defined at boundary lines.
- The command affects mass bodies and anchors.
- Default = 1, 0.5 ≤ expansion factor ≤ 4.0

FE-Size Transition Mass Volumes

Defines how rapidly internal elements change their size compared to outer elements

- A transition factor of 1 is the fastest transition and a factor of 4 changes the element size smoothly (see MOPT command in the ANSYS® Users Guide).
- The command affects mass bodies and anchors.
- Default = 2, 1.0 ≤ transition factor ≤ 4.0

FE-Size Expansion Spring Volumes

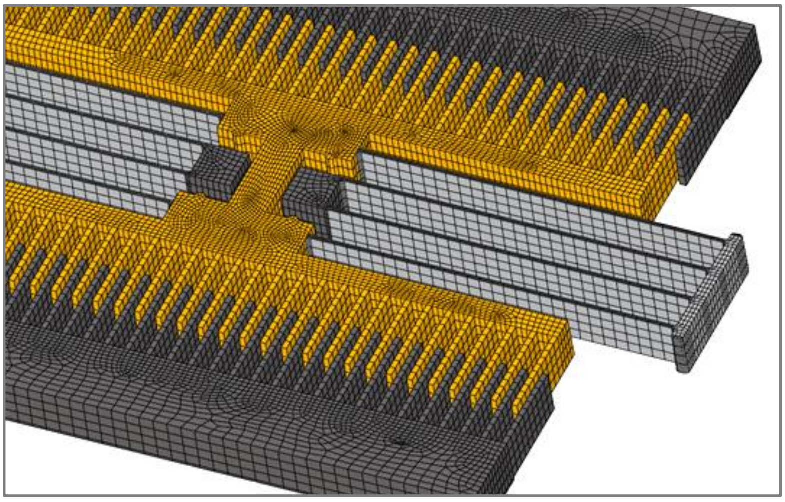
Defines the size of internal finite elements compared to the size at boundary lines

- Elements at the interior domain can either contract (factor is smaller than 1) or expand (factor is larger than 1) compared to the element size defined at boundary lines.
- The command affects connecting volumes of springs.
- Default = 1, 0.5 ≤ expansion factor ≤ 4.0

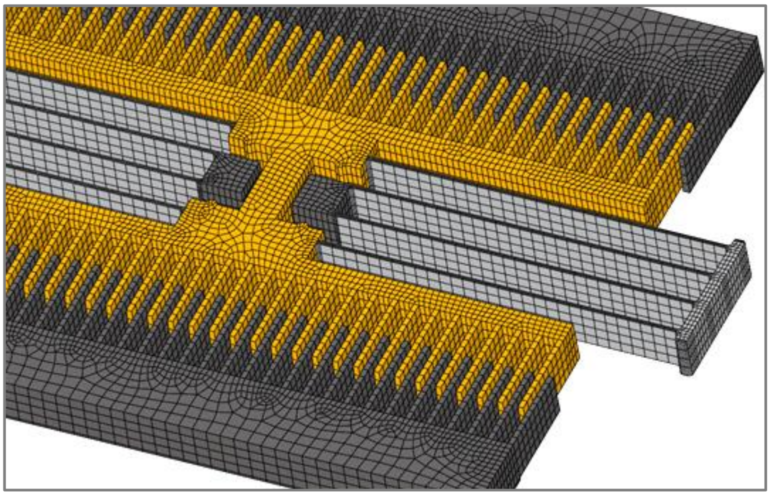
FE-Size Transition Spring Volumes

Defines how rapidly internal elements change their size compared to outer elements

- A transition factor of 1 is the fastest transition and a factor of 4 changes the element size smoothly (see MOPT command in the ANSYS® Users Guide).
- The command affects connecting volumes of springs.
- Default = 2, 1.0 ≤ transition factor ≤ 4.0

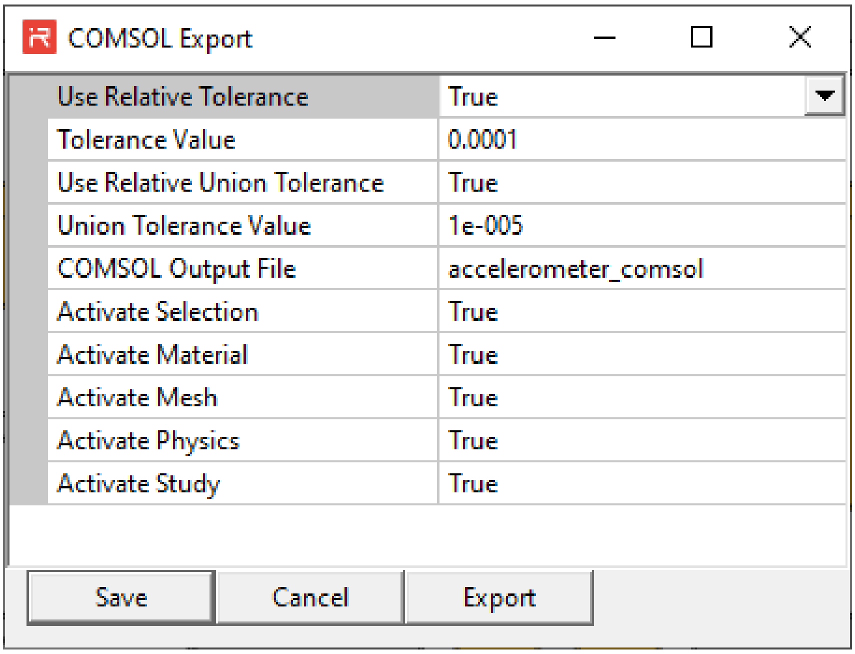


- Mesh expansion factor is set to 0.5 (size reduction).
- Perforations are removed for demonstration.



- Mesh expansion factor is set to 3 (size expansion).
- Expansion is used to create a coarse mesh in larger mass bodies.

Further settings are specific for the COMSOL® finite element model export:



Use Relative Tolerance

Activate absolute and relative tolerance settings for combining areas to volumes

- Tolerance settings in COMSOL® and model specific. The export windows assigns global settings. Data of individual items can be assigned to area and volumes in COMSOL®.
- Default = true, (relative tolerance)

Tolerance Value

Default tolerance value for all COMSOL® areas and volumes

- Default = 1e-5, (relative tolerance)

Use Relative Union Tolerance

Activate absolute and relative tolerance settings for combining volumes to a solid model

- Tolerance settings in COMSOL® and model specific. The COMSOL® export window assigns global settings. The "union" operation combines all volumes to a final solid model.
- Default = true, (relative tolerance)

Union Tolerance Value

Default tolerance value for the Union operation in COMSOL®

- Default = 1e-6, (relative tolerance)

COMSOL® Output File

Defines the name of the COMSOL® output files

- The "Interface to COMSOL" creates three output files of the same name and the extensions *.class, *.java and *.txt. The *.class file can be opened in the COMSOL graphical user interface.
- The files are stored in the MODELBUILDER working directory if no path to a named folder is specified. Extensions are automatically assigned to the file name.

Activate Selection

Activates assigned names for model items

- Default = true

Activate Material

Activates assigned material properties

- Default = true

Activate Mesh

Activates mesh settings and build mesh commands

- Default = true

Activate Physics

Activates mechanical and electrostatic domain settings

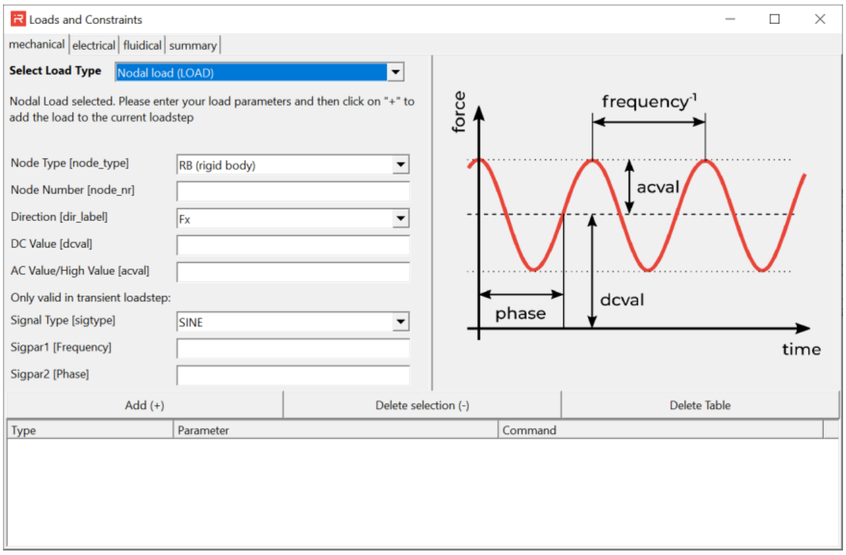
- Default = true

Activate Study

Activates settings for a modal analysis

- Default = true

7 Assign Loads and Constraints window

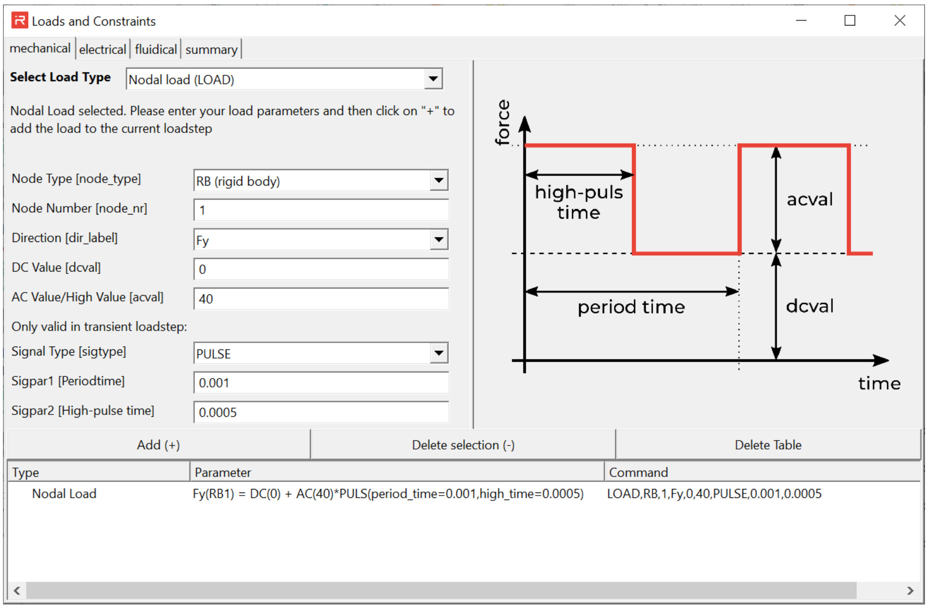


Loads and constraints can either be assigned by [Solution commands] or in a graphical way using the loads and constraints window.

It consists of sub-windows which are used to assign mechanical, electrical and fluidic loads or constraints. The fourth sub-window provides a summary and can be used to read or write load cased from a file.

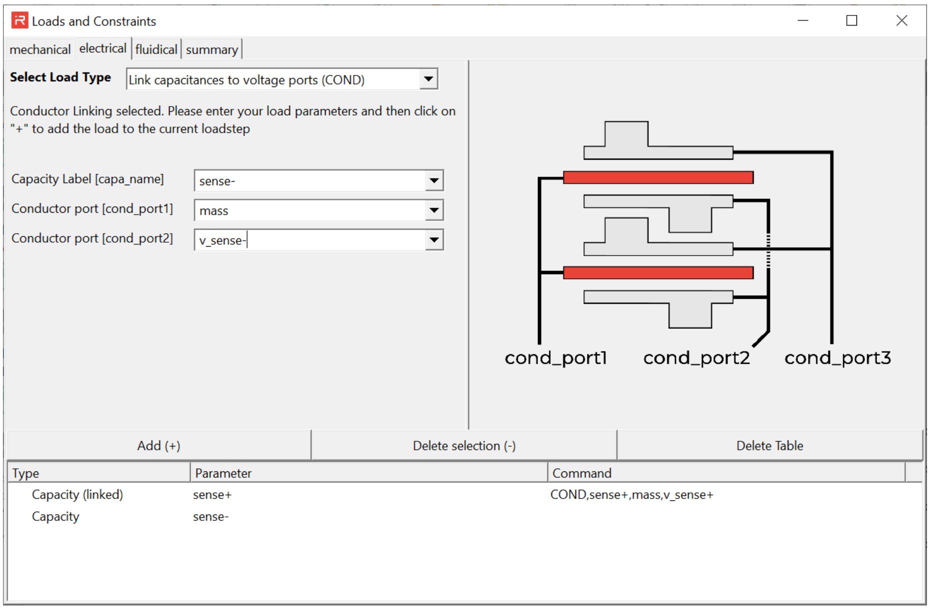
Mechanical domain

The mechanical domain sub-window covers the LOAD, ACEL, OMEG and DRIV command features. The following figure shows a pulse force signal in y-direction at rigid body #1. Click on the Add (+) button do assign the specified command. Individual commands can be erased by the Delete selection (-) button and all mechanical loads and constraints are erased by the Delete Table button.

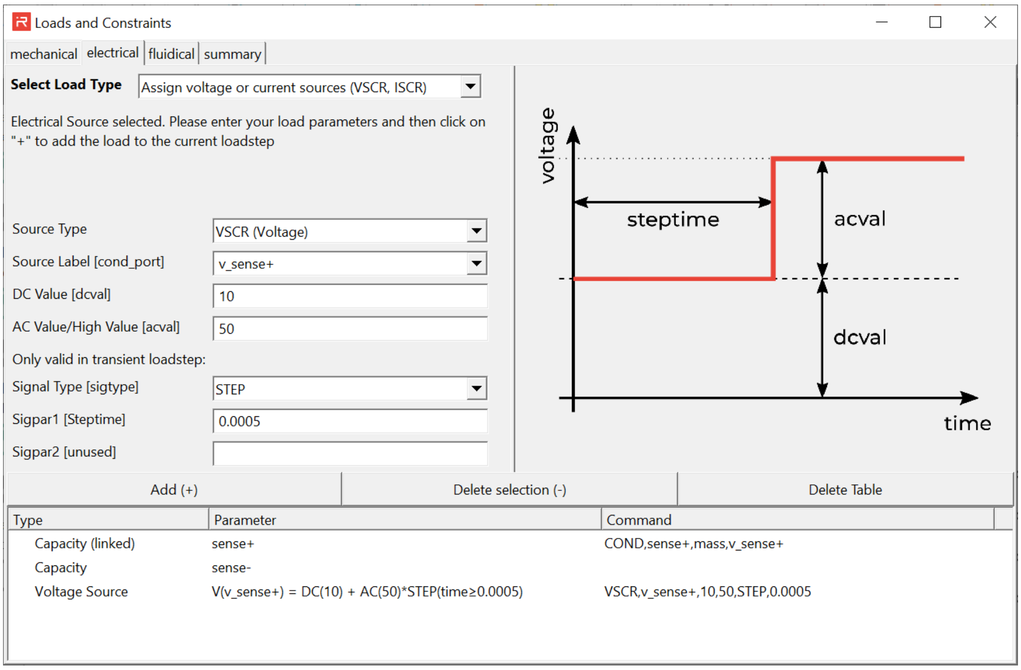


Electrical domain

The electrical domain sub-window covers the COND, VSCR and ISCR command features. In a first step, conductor port names are assigned to the comb cells and plate capacitances. Existing comb and plate capacitance labels from the model file appear automatically. The conductor port names are assigned by users. Click on Add (+) to link the capacitance to conductor ports.

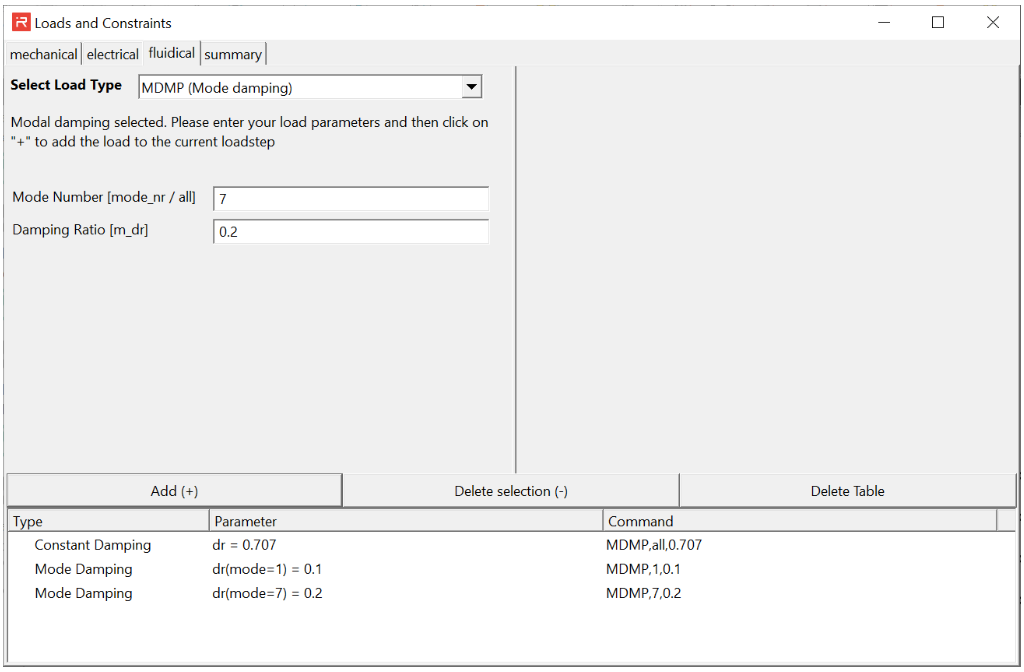


In a second step, different voltage or current sources can be assigned to the conductor ports.



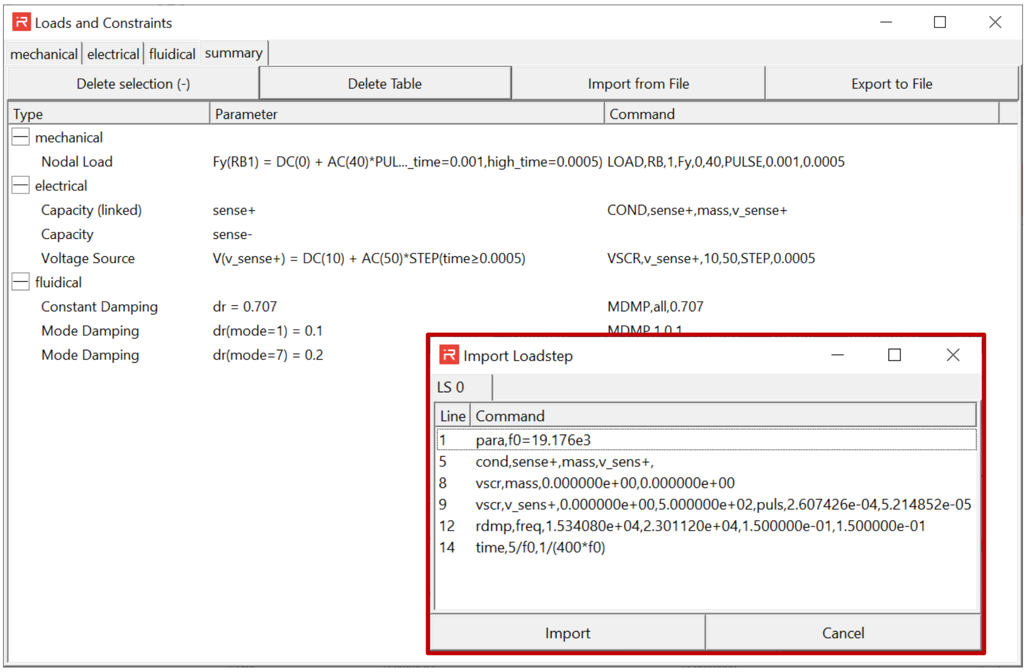
Fluidic domain

The fluidic domain sub-window covers the CDMP, RDMP and MDMP command features. The following figure illustrates the assignment of different modal damping ratios. Click on Add (+) to activate the commands.



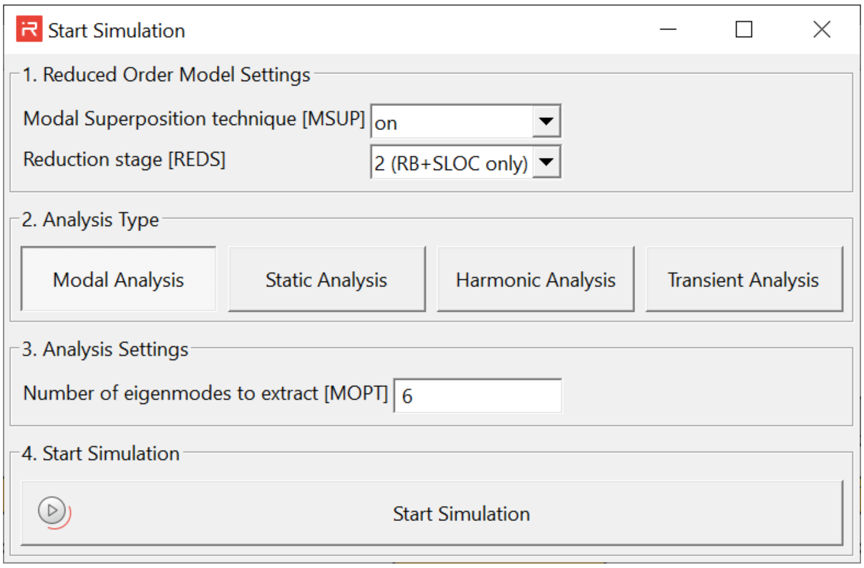
Import and export of load cases

The last sub-window shows a summary of all assigned loads and constrains. The currently define load case can be exported to file.



The import from file button is used to retrieve existing load cases. The file content can be added by other solution commands such as MOPT, DCSW, HARF, TIME, to assign solver settings.

8 Simulate window

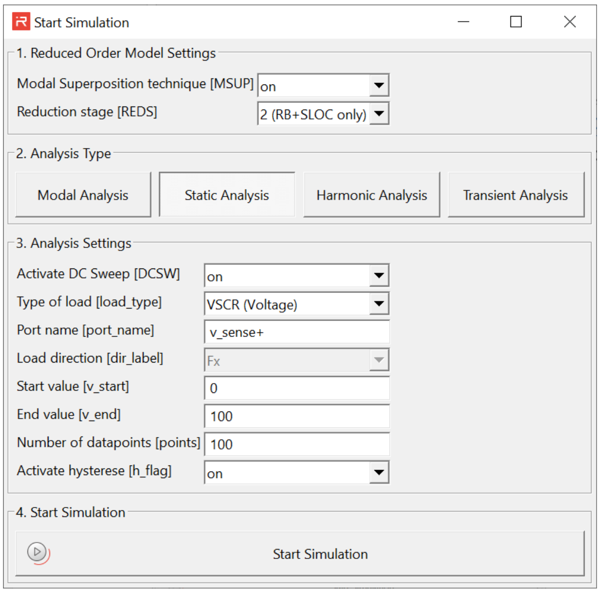


Simulation control parameters can either be assigned by Solution commands or in a graphical way using the simulate window.

It consists of four sub-windows which are used to activate and control modal analyses (MOPT), static analyses (DCSW), harmonic analyses (HARF) and transient analyses settings (TIME). MSUP and REDS model types can be temporarily changed for the following simulation. Finally, the simulation run is initiated by clicking on the start simulation button.

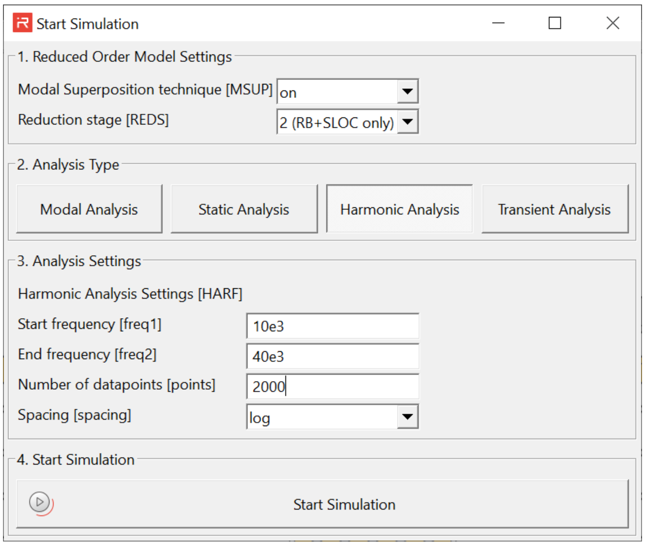
Static analyses settings

Static analysis settings are only necessary for DC-sweep operations. The following figure shows parameter settings for a voltage sweep on conductor "v_sense+".



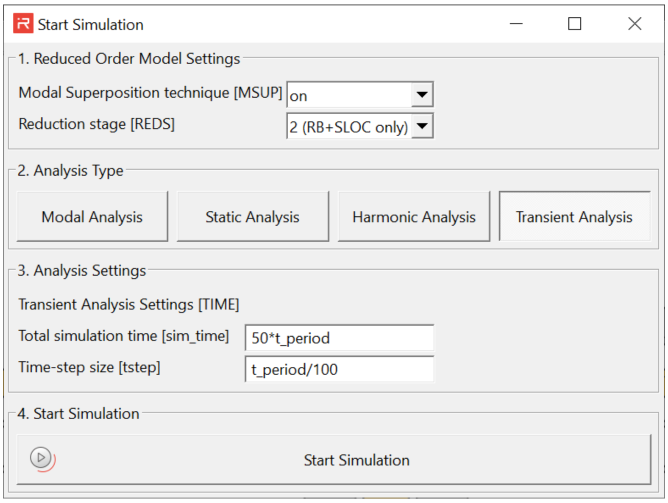
Harmonic response analyses settings

The following figure shows parameters for a harmonic response analyses which covers a frequency range from 10 kHz to 40 Hz with 2000 frequency steps. The frequency spacing is logarithmic.

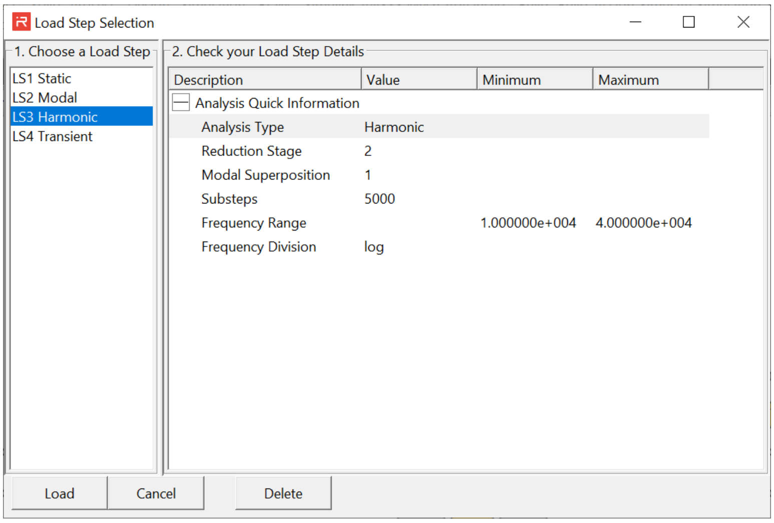


Transient analyses settings

The following figure shows typical parameters for transient analyses. The total simulation time is 50 periods and each period will be analyzed with 100 time-steps. “t_period” is a used defined parameter which corresponds to the inverse of the first eigenfrequency in the given example.



9 Simulation results window



The simulation results window can be used to select existing load steps for results evaluation. Furthermore, the window provides a quick summary of each load step and can be used to delete load cased from the project folder.

All load cases are automatically erased by generating a new model with “Build solids” or “Build ROM”. The results should be copied to a new project folder to save the data permanently.

Part G – Files and results export features

1 Content of Files

The i-ROM MODELBUILDER creates the following model files in the working directory:

File name	Description
`*.iromproj`	Project file with all model settings of the graphical user interface (text file) ➔ Changes take effect after opening models
`*.irommod`	User model input file defining the MEMS model (text file) ➔ Changes take effect after "Build Solids"
`*.iromcore`	Contains all simulation model data (binary file)
`*.iromgobj`	Contains user model graphics data (binary file)
`*.iromintl`	Contains internal simulator parameters (binary file)
`*.iromlnfo`	Contains reference data for all load step (binary file)
`*.iromromd`	Contains graphics data for postprocessing (binary file)
`*.iromlsxx`	Contains individual results of load steps (binary file)

Further user defined result files are:

- Files for SIMULINK® model export:
  - *.slx → Simulink model file
  - *_ini_simulink.m → Simulink model parameters

- Files for ANSYS® model export:
  - *_apdl.txt → ANSYS-APDL file for model exchange

- Files for COMSOL® model export:
  - *.class → Compiled COMSOL model file
  - *.java → Temporary Java file for COMSOL export
  - *.txt → COMSOL model components

➢ General model information files:
- *_MASS_MATRIX.txt → Inertial properties of rigid bodies
- *_SECTIONPROP.txt → Cross-section properties for springs

2 Result export features

The “Data Plot” windows supports several output formats for simulation results. Clicking on the red “Data export” icon opens a MATLAB workspace window with all alphanumerical data for the selected result items.

The arrow on the right of the “Data export” icon (marked in blue) opens a selection window:

- Data export: Saves the simulation results in a MATLAB formatted binary file (MAT-file) which is stored in the current working directory.
- Graph export: Saves the simulation results to a graphic file into the current working directory. Supported formats are “pdf”, “png”, “jpg”, “gif”, “bmp” and “svg”.