

System Analysis and Applications with PySimulator

Andreas Pfeiffer, Matthias Hellerer, Stefan Hartweg,
Martin Otter, Matthias Reiner, Jakub Tobolar

DLR German Aerospace Center Oberpfaffenhofen
Institute of System Dynamics and Control



Knowledge for Tomorrow

Objective

A generic framework to


- **simulate** models (Modelica, FMI) with different engines / tools,
- manage (especially **reading**) the simulation results,
- **visualize** data (plotting) and
- to **analyze** models and result data:

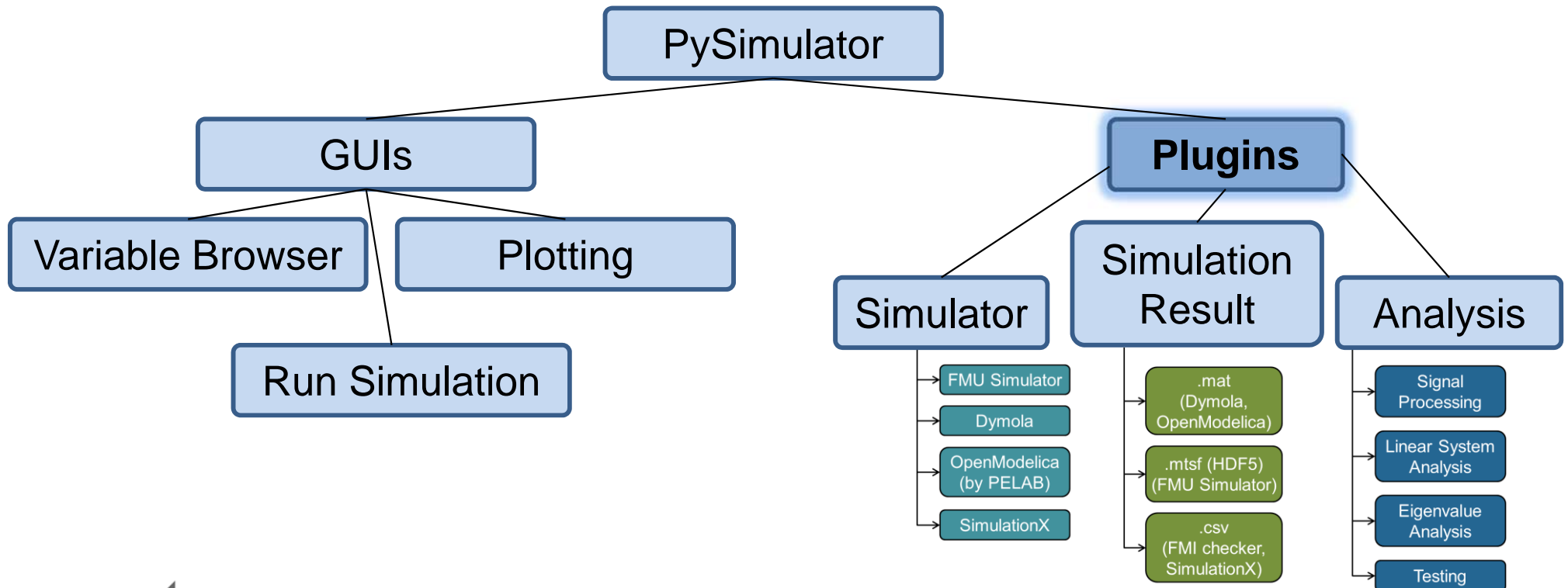
⇒ **PySimulator**





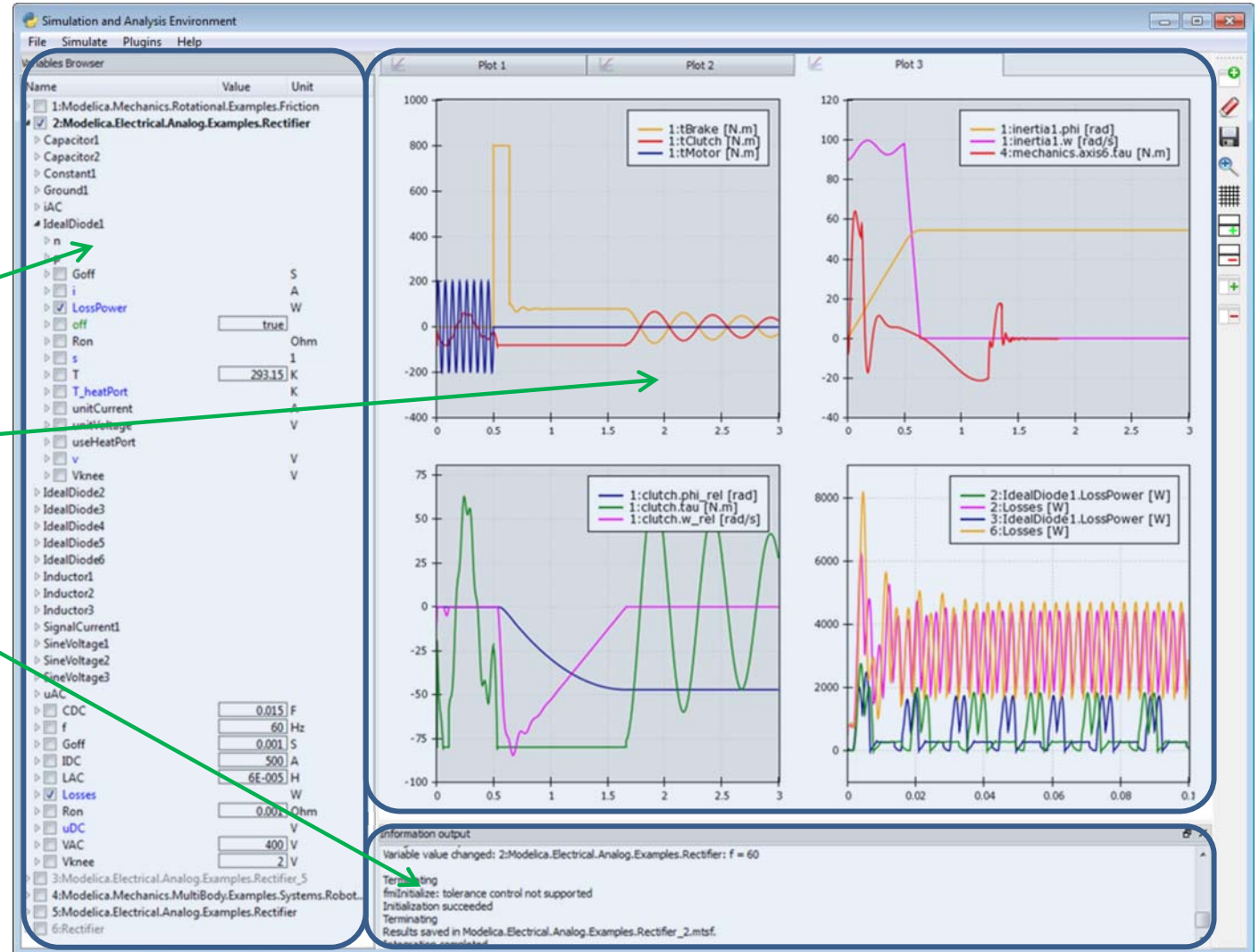
PySimulator

- **Open-Source** software (LGPL) in Python 
- **User friendly** by GUIs and interactivity
- **Modular** with Plugin infrastructure



1 Graphical User Interface

- Qt framework by PySide
- Variables
- Plots
- Console



1.1 Variable Browser

- Models and variables
 - Intuitive **tree-view**
 - Context **menus** from plugins

- Models
 - **Select** for simulation
 - Select for analysis on results
 - Unique **result file**

- Variables
 - Change **parameters**
 - **Select** for plotting
 - **Colors** for different **variability**
 - Detailed information on **lowest level**

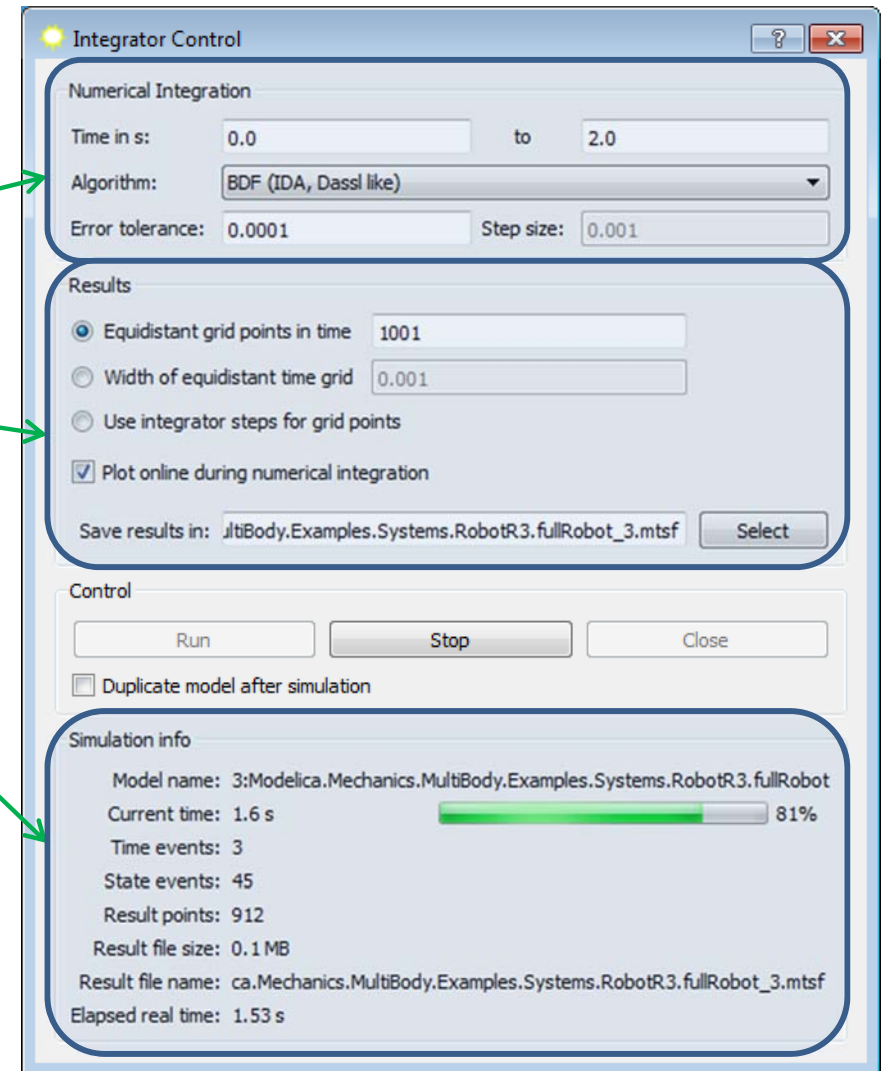
The screenshot shows the Variable Browser interface with a tree view on the left and a table on the right. The tree view shows a hierarchy of models and variables, with '5:M' selected. A context menu is open over '5:M', showing options: 'Close Model', 'Duplicate Model', 'Compare Results', and 'Results'. The 'Results' option is expanded, showing a file named 'FullRobot_1.mtsf' with a size of 6.5 MB. The table on the right shows the following data:

Name	Value	Unit
1:FullRobot		
2:M		
3:M		
4:Re		
5:M		
6:Fr		
brake		
flange_a		
flange_b		
muc_pos		
support		
a		rad/s ²
a_relfric		rad/s ²
Backward		
cgeo	1	
fn		N
fn_max	1600	N
Forward		
Free		
free		
f_normalized		1
locked	false	



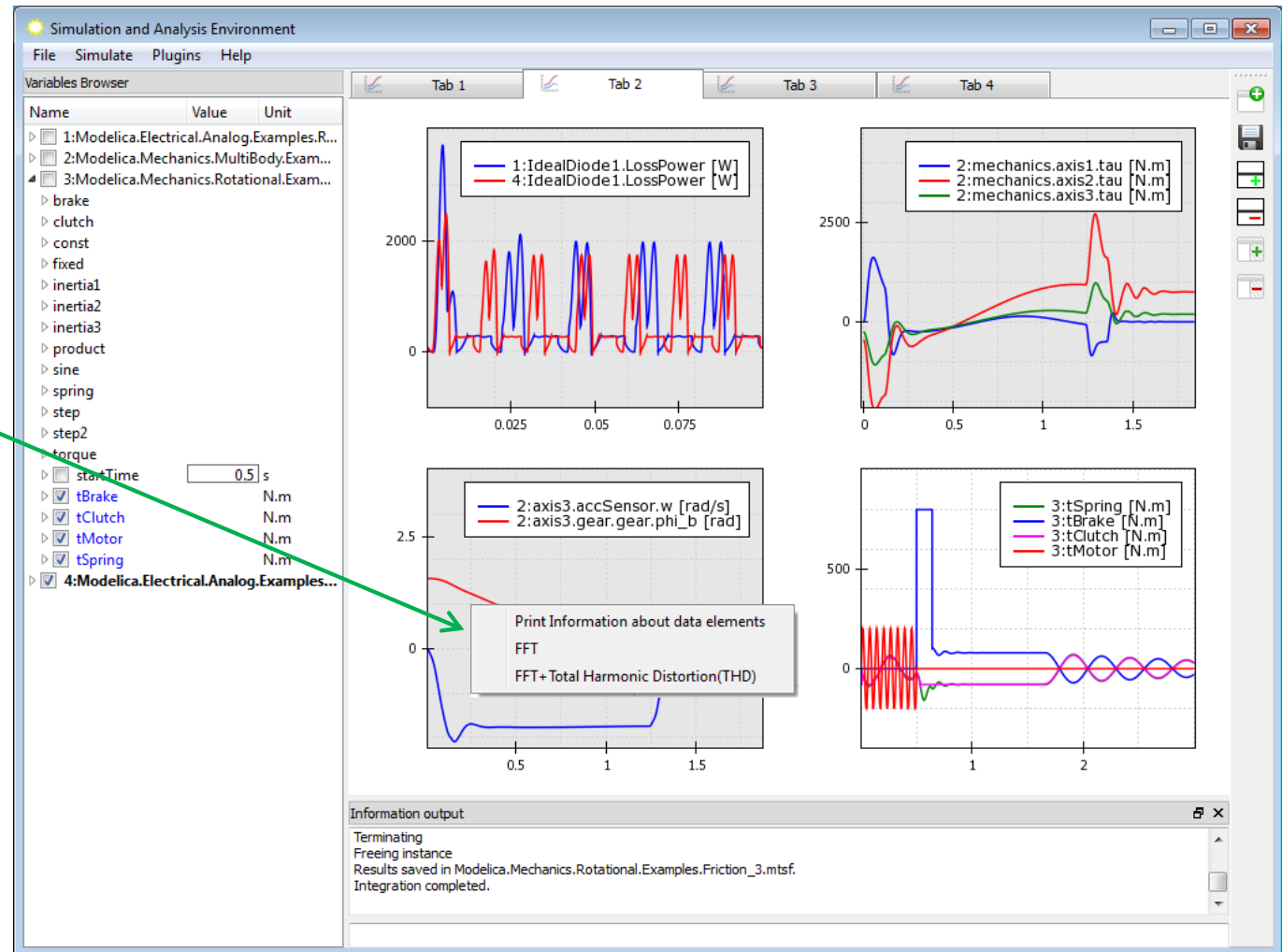
1.2 Run Simulation

- **Generic** settings for
 - Simulation parameters and **integration algorithm**
 - **Results**: time grid, file name
 - Simulation progress and integration **statistics**
- Simulator **plugin**:
 - Provides **concrete** information

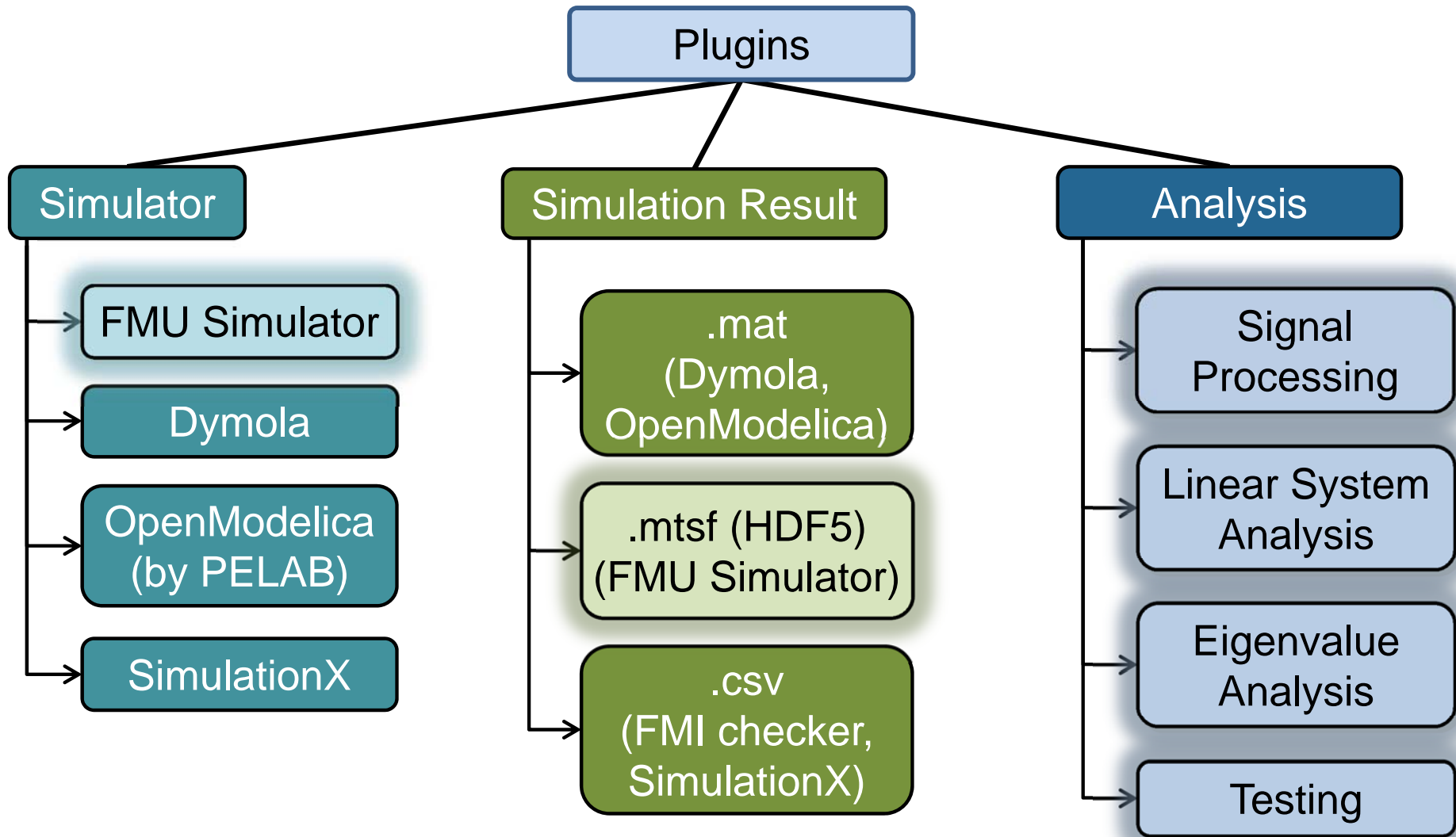


1.3 Plotting of Variables

- Plot engine: **Chaco**
- **Tabs** and **Matrix plots**
- Context **menu** (plugins)
- **Interactivity:**
 - Panning
 - Zooming (all axes, per axis, selection)
 - Time selection



2 Plugin Structure



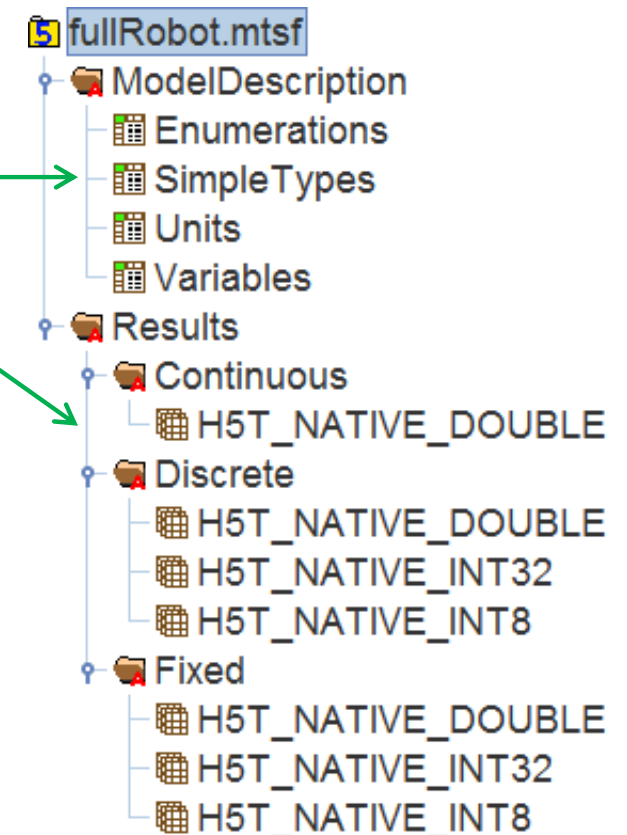
2.1 FMU Simulator

- **FMU** = Functional Mockup Unit based on FMI
- Parse FMU model **description**
- Interface to FMU **binary**
- **Integrator** based on Assimulo
(includes interface to Sundials Solver Suite)
- **Results** stored in MTSF format
 - Full robot model from the Modelica Standard Library, 30 Mio. result points in time
→ **171 GBytes** result file



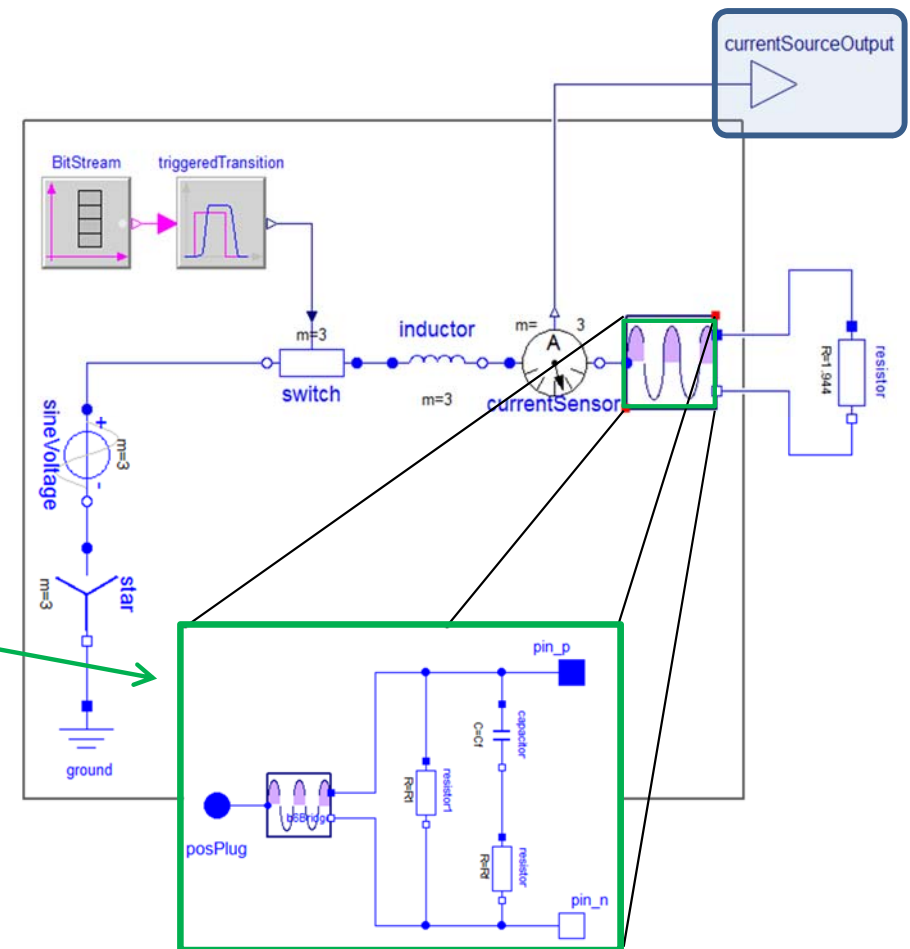
2.2 Standard Time Series File Format (MTSF)

- Naming: **M**odelica Association **T**ime **S**eries **F**ile Format
- Proposal for an **Open Standard** (DLR, Bausch-Gall)
- Basic File Format: **HDF5**
- Main concept:
 - **Model Description** (meta info) in few lists
 - Numeric data in **Time Series** matrices
- Test-Implementation in Python:
 - Writing and Reading of **large files**: proved.
 - Application in **PySimulator**
- **Participate** in the development of the Standard?
→ Contact us!



3.1 Application: Energy Grid of an Airplane

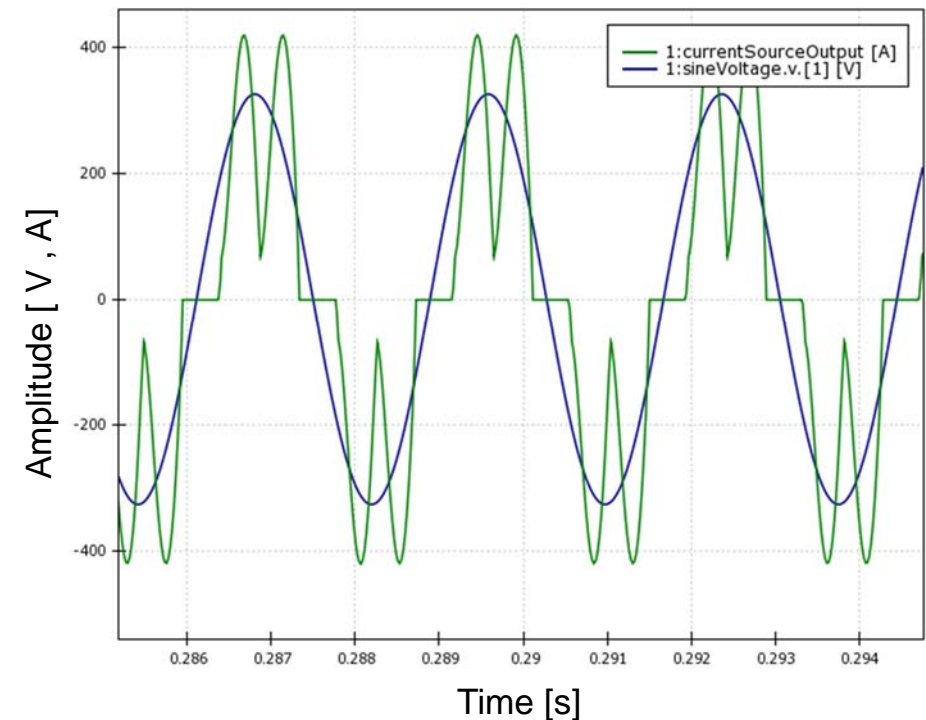
- **Virtual testing** of aircraft power systems:
 - Standalone model-based test of an aircraft AC network
- **Rectifier unit** as a 230 V AC load



3.1 Application: Energy Grid of an Airplane

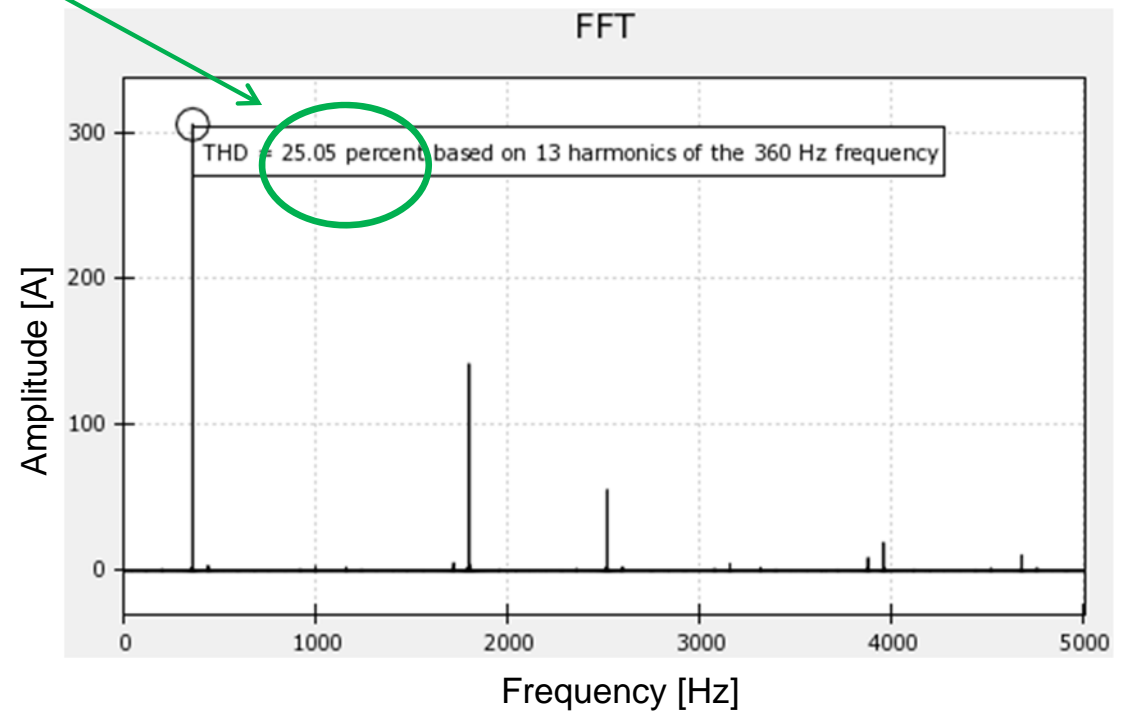
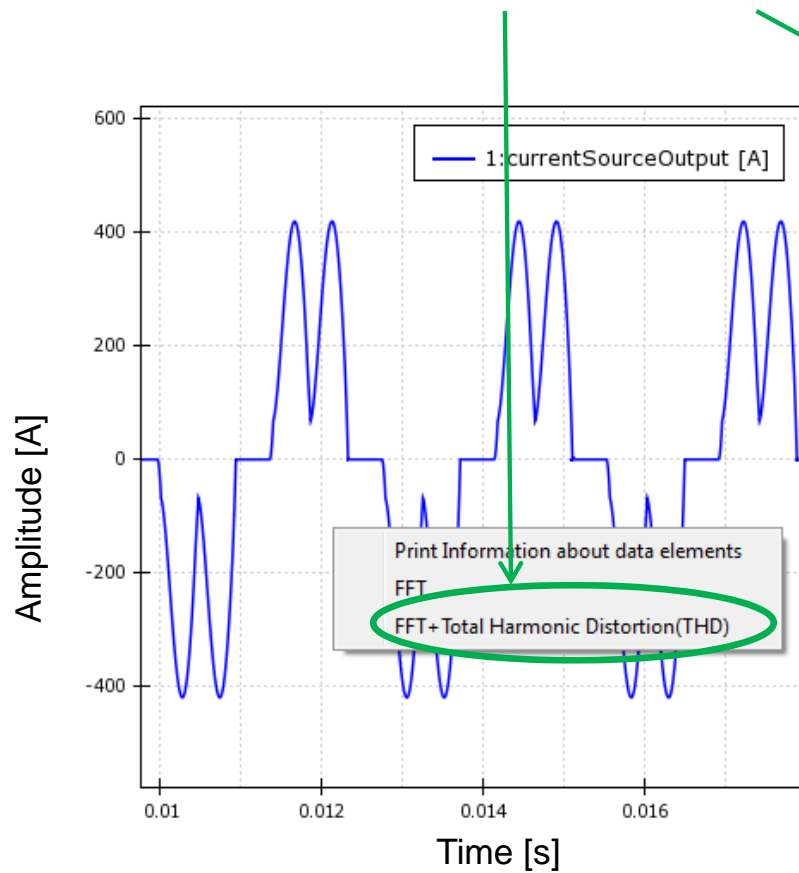
- Challenge: Nonlinear parts
 → **Distortion** in power systems
- One measure:
 Total Harmonic Distortion (**THD**)

$$\text{THD} = \frac{\sqrt{I_2^2 + I_3^2 + \dots + I_n^2}}{I_1}$$



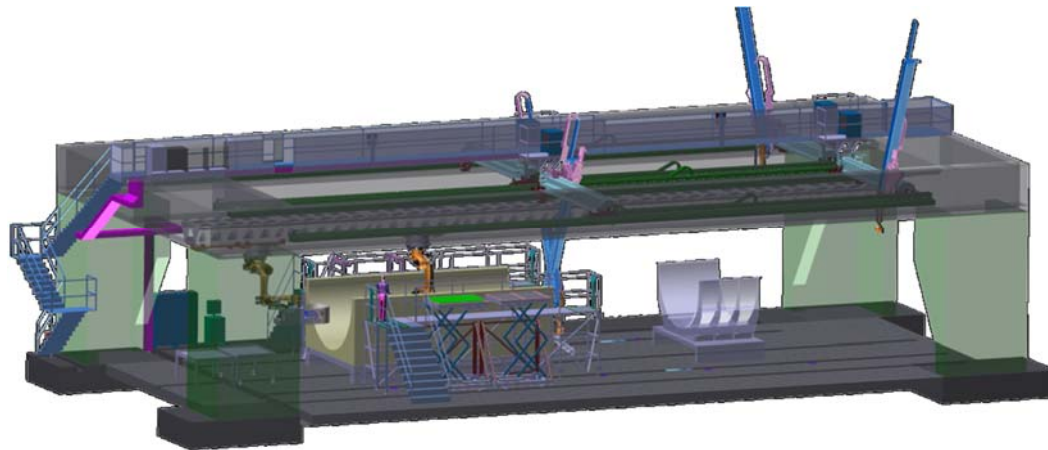
3.1 Application: Energy Grid of an Airplane

- Small **plugin** for THD based on FFT plugin



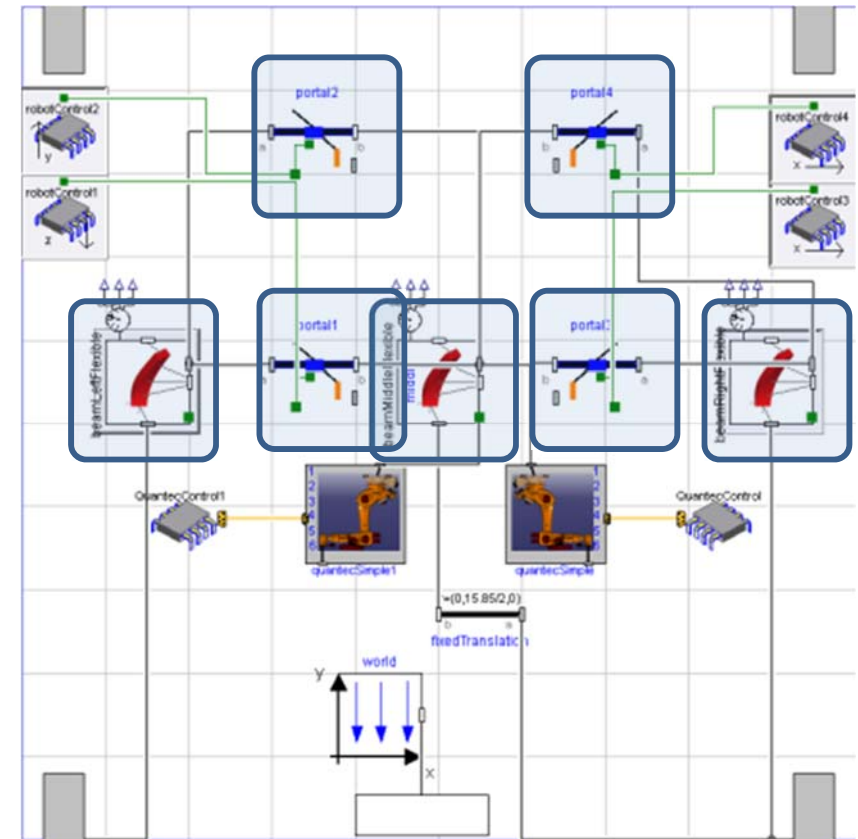
3.2 Application: Vibration Analysis

- **Vibration Analysis** of a mechanical system
- DLR Augsburg: Manufacturing of large carbon-fiber composite structures for airplanes by robots



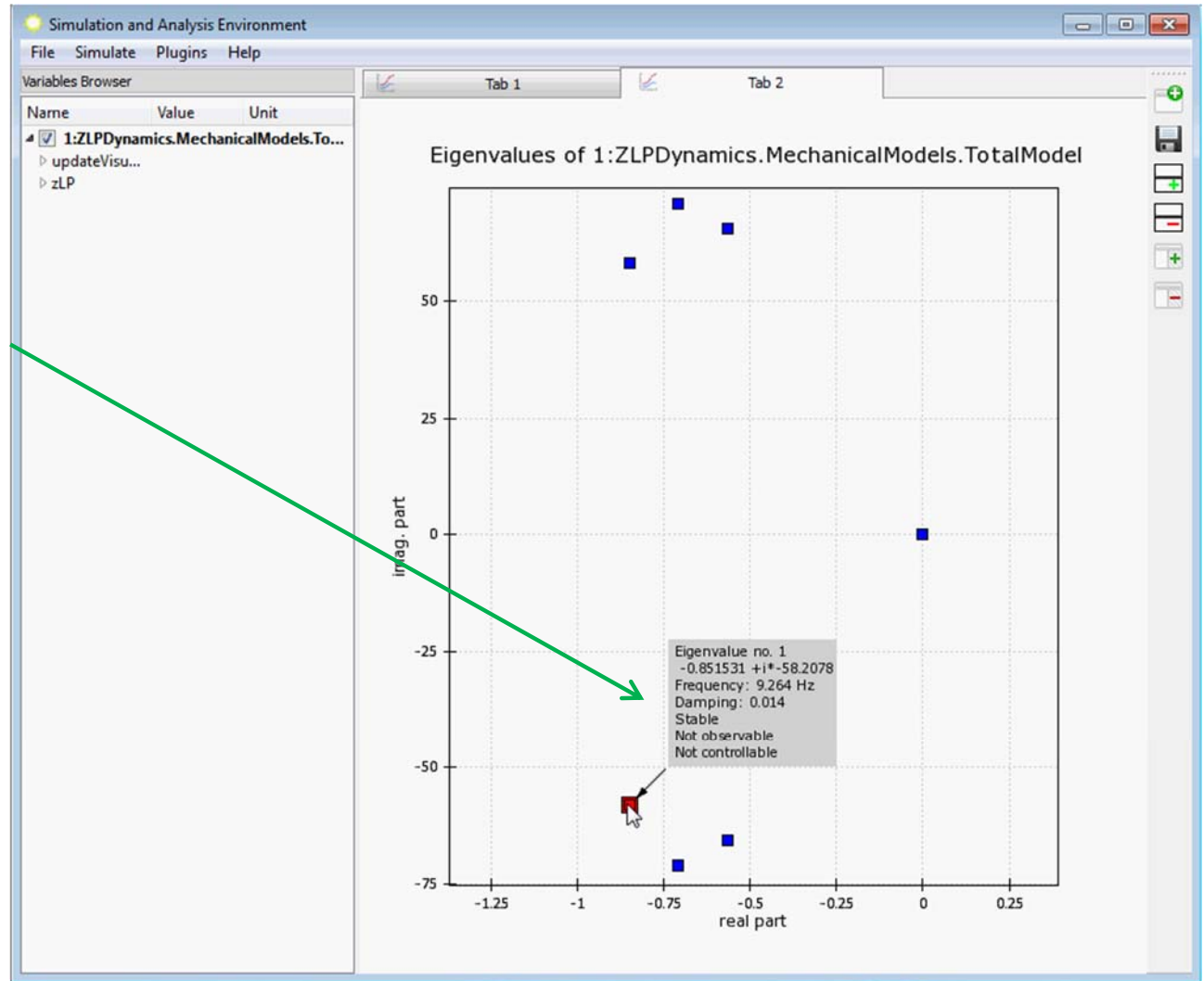
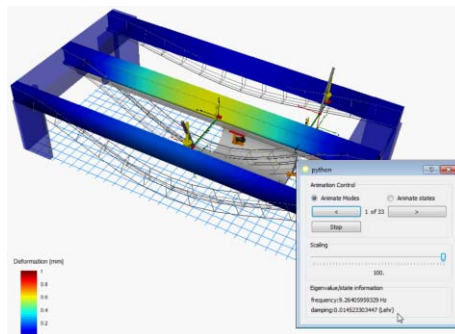
Dimensions: 30 m x 15 m x 7 m

Modelica model:



3.2 Application: Vibration Analysis

- Modelica → **FMU**
- **Linearization** at steady state
- **Eigenvalue** Analysis
- **Mode animation** (requires visualization of model, e.g. DLR SimVis)



3.3 Application: Automated Testing

- **Comparing** results:
 - for a Modelica model simulated by different tools
 - for a Modelica model and its corresponding FMU
 - for an FMU generated by different tools for the same model
 - for different versions of a model (same tool)
 - ...
- **Example:** PlanarMechanicsStandard.Examples.TwoTrackWithDifferentialGear (Modelica model) simulated by
 - OpenModelica → mat-file
 - Dymola → mat-file
 - SimulationX → csv-file
 - FMUSimulator (with FMU) → mtsf-file
- Under development:
 - Plugin for **automated numerical comparison** of all variables



4 Summary

- PySimulator: **Open Source** software in **Python**
- Brings together **simulation results** of different simulation tools:
 - Plotting
 - Analysis
 - Testing
- **Applications** from aerospace, electrics, robotics, mechanics, automated testing
- Simply extendable by **own plugins** – try it!
- **Download** and information: www.pysimulator.org

