Development of an Interface between CAD-Systems and SimMechanics

Prototyping and Optimization of Mechatronic Systems

MATLAB EXPO 2014
9th July 2014

Dipl.-Ing. Christopher Schropp
schropp@tum.de
Current Situation / Motivation

- Very common tools in industry: MATLAB, Simulink, CATIA V5, …
- There is no direct, bidirectional interface
- Separation of the development process of mechatronic systems
- CONSEQUENCE:
  - An overall optimization of the development is difficult.
  - There is a lot of manual work, hence many sources of error.

Current engineering process chain

- Create CAD model
- Create Skeleton model
- Link the two models
- Add missing values
- Check model consistency
- Simulation
- Test of total system
Market Analysis

- Distinction between the interfaces: Real interaction – Exchange file

<table>
<thead>
<tr>
<th>Connectivity to CATIA</th>
<th>SimMechanics Link</th>
<th>CAPRI</th>
<th>CAMAT</th>
<th>Our Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✗</td>
<td>✗</td>
<td>XML</td>
<td>✗</td>
</tr>
<tr>
<td>Bidirectionality</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>SimMechanics Model Creation</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>No manual mapping</td>
<td>-</td>
<td>-</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>
Goal of our Research Project

- **Bidirectional interface** between MATLAB and SimMechanics and a selection of CAD-Tools:
  - Masses, CGs, Moments of Inertia, ...

  CAD-System (e.g. CATIA V5, …)  
  Interface  
  MATLAB (SimMechanics)  
  CAD Parameter changes

- This interface can be used for **optimizations**, for example optimizing CAD parameters for lowest mass, highest forces, …
Goal of our Research Project

Change of the current development process:

Current engineering process chain

1. Create CAD model
2. Create Skeleton model
3. Link the two models
4. Add missing values
5. Check model consistency
6. Simulation
7. Test of total system

New engineering process chain

1. Create CAD model
2. Convert model with our solution
3. Simulation
4. Test of total system
Framework of the Research Project

- **Sponsored by ZIM (Zentrales Innovationsprogramm Mittelstand)**
  
  - Funding program particularly for collaborations of medium-sized companies and research institutions.

- **Our partner company:**

  ![SANEON Logo]
Our Concept

- Several possibilities to connect MATLAB and CATIA V5 (COM, CAA, …)
- We use the COM Server of CATIA V5 together with a .NET DLL

```matlab
myModel = iCADModel( 'SampleCatiaProduct.CATProduct', 1);
myModel.GetCADData;
myModel.CreateSimulinkModel;
```
SimMechanics Model Creation

- Our Algorithm uses only the transferred data for model creation (no compulsory further active connection to the CAD system is necessary).
Additional Features

- **Automatic generation of detailed part screenshots** for an improved presentation of the Simulink diagram. Created JPEGs are used as mask icons.

- **Automatic generation of STL files** for better visualization in *Mechanics Explorer*. 
Use Case
Simulink Model Creation Video
Parameter Changes & Update Functionality

- **Bidirectionality** allows manipulation of CAD parameters (e.g. lengths, diameters, positions, …).

- Resulting changes in the parts and products are calculated by the CAD system.

- After receiving the new data, the MATLAB CAD model data is updated.

- Since the SimMechanics model is parametrized, there are no updates necessary for the diagram.
The manipulation of CAD parameters and the update functionality allows to lay an optimization loop over both tools.

Besides the MATLAB optimization algorithms (e.g. included in the optimization toolbox), the framework allows to easily implement any other optimization algorithm (e.g. SNOPT, IPOPT, ...).
Optimization Video
Benchmark Results

1 Assembly
24 Parts
46 Constraints
31 Joints

Transfer of CAD model data and creation of a SimMechanics model in about 100 seconds.
(Time effort with manual work > 50h!)

- Intel® Core™ i7-2640M CPU 2.80 GHz / 8.00 GB RAM/ 64 Bit OS
Main Advantages of the Developed Software

- Enormous reduction of manual work, working time and in the end development costs.
- Elimination of the error source „human“ in the SimMechanics model creation.
- No separation of the product development process.
- Possibility to use already in MATLAB implemented methods like optimization algorithms, sensitivity analysis, ...
- Attach Simscape subsystems to CAD files.
- Automatic consideration of installation space during optimization.
Outlook

- Support of other popular CAD systems.

- Export of a xml file for documentation purposes.

- Intuitive and user friendly GUI for an easy control of the software.
Thank you very much for your attention!

Christopher Schropp
schropp@tum.de
+49 89 289 16349

Sam Nezhat
sam.nezhat@saneon.de
+49 89 548 426 50
Exemplary way of data: `myPartsData = myModel.GetParts`

- `iCADModel`
  - `CADSystem.GetParts`
  - `myPartsData = {
      ['Part1', 'Product1', 20.5, ...]
      [...]
    }`

- `iCADSystem`
  - `cad.GetParts`
  - `Conversion to: {
      ['char(Part1)', 'char(Product1)', 20.5, ...]
      [...]
    }`

- `.NET DLL`
  - `Foreach Product in RootProduct.Products {
      GetMass(Product)
      [...]}
    }`
  - `{[
      ['Part1', 'Product1', 20.5, ...]
      ['Part2', 'Product1', 14.8, ...]
      [...]
    ]}`

- `CATIA V5`
Calculation of transformation data

- What is needed for SimMechanics: Rotational axis and angle of rotation ($\vec{v}, \alpha$) as well as the translation in Cartesian coordinates ($\vec{t}$).

- What is supplied by the CAD system: Origin and orientation of the parts ($\vec{o}_P, \vec{A}_P$) as well as the origin and the direction (if available) of the joints ($\vec{o}_j, \vec{d}_j$) (e.g. rotational axis).
Calculation of rotation data

Formulate direct cosine matrix (DCM) between two coordinate systems

\[ \overrightarrow{A}_j = \begin{bmatrix} x_j \\ y_j \\ z_j \end{bmatrix} \]

\[ T_{JP} = \begin{bmatrix}
\cos \alpha (\overrightarrow{x}_p, \overrightarrow{x}_j) & \cos \alpha (\overrightarrow{y}_p, \overrightarrow{x}_j) & \cos \alpha (\overrightarrow{z}_p, \overrightarrow{x}_j) \\
\cos \alpha (\overrightarrow{x}_p, \overrightarrow{y}_j) & \cos \alpha (\overrightarrow{y}_p, \overrightarrow{y}_j) & \cos \alpha (\overrightarrow{z}_p, \overrightarrow{y}_j) \\
\cos \alpha (\overrightarrow{x}_p, \overrightarrow{z}_j) & \cos \alpha (\overrightarrow{y}_p, \overrightarrow{z}_j) & \cos \alpha (\overrightarrow{z}_p, \overrightarrow{z}_j)
\end{bmatrix} \]

Convert DCM to quaternion

\[ q = dcm2quat(T) = \begin{bmatrix} S \\ v \end{bmatrix} \]

Formulate corresponding coordinate system for joint (e.g. rotational axis equals z-axis)

\[ \alpha = 2 \times \cos^{-1}(s) \quad \begin{bmatrix} \vec{x} \\ \vec{y} \\ \vec{z} \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \]
The following DMU Kinematics joints are currently supported:

<table>
<thead>
<tr>
<th>Joint Type</th>
<th>Supported Joint</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATKinRigidJoint</td>
<td>Weld Joint</td>
</tr>
<tr>
<td>CATKinRevoluteJoint</td>
<td>Revolute Joint</td>
</tr>
<tr>
<td>CATKinPrismaticJoint</td>
<td>Prismatic Joint</td>
</tr>
<tr>
<td>CATKinCylindricalJoint</td>
<td>Cylindrical Joint</td>
</tr>
<tr>
<td>CATKinSphericalJoint</td>
<td>Spherical Joint</td>
</tr>
</tbody>
</table>

Fixed Parts
Data structure

- **CAD Model Data in MATLAB Workspace**

  - **Workspace**
    - **Name**: Value
    - **myModel**: 1x1 `iCADModel`

  - **Root Product Details**
    - **myModel**: 1x1 `iProduct`
      - **Parent**: NaN
      - **ParentName**: `Product1`
      - **InstanceName**: `Control_Rod_Assembly.CADModel`
      - **Elements**: 1x22 cell
      - **Constraints**: 1x32 `iConstraint`
      - **Parameters**: []
      - **Mechanisms**: 1x14 `iMechanism`

  - **Mechanism**: 1, Freedom = 38
    - **Connections**: 11
      - **Connections 2**: (Stick_Bellcrank_Inner_Box, Stick_Bellcrank_Outer_Box)
        - **Spherical (Connector_Bearing_Sphere, 15x25x10.1, Bearing_under Seat) 1**
        - **Spherical 7 (Connector_Bearing_Sphere, 15x25x10.1, Elevator_Rod_A) 1**
        - **Spherical 6 (Elevator_Rod_B, Elevator_Rod_A) 1**
        - **Rollen 11 (Bell_Crank_Center, Bearing_Center_Bell_Crank) 1**
        - **Spherical 13 (Elevator_Rod_B, Bell_Crank_Center) 1**
        - **Spherical 11 (Product4,1, Bell_Crank_Center) 2**
        - **Spherical 10 (Bell_Crank_Center, Bearing_Center_Bell_Crank) 1**
        - **Spherical 12 (Elevator_Rod_A, Elevator_Surface) 1**
        - **Spherical 13 (Elevator_Rod_A, Elevator_Surface) 1**
        - **Spherical 16 (Stick1, Bolt10x50.1) 1**

  - **Offset 64**: (Bell_Crank_Center, Bearing_Center_Bell_Crank) 1
Framework in MATLAB

iCADSystem

iSimulinkModelBuilder

iMechanism

iParameter

iConstraint

iJoint

iPart

iProduct

iOptimizationProblem

.NET DLL (VB) CAD-System specific

SimMechanics Model
CATIA V5 Design Guideline

- Elements with fixed position may not be part of a FixTogether group.
- Points for spherical joint congruency must be created separately (Generative Shape Design).
- Subassemblies must only contain FixTogethers.
- At the moment, only constraints created as joints (DMU Kinematics) will be detected by our interface.

Currently supported joints are:

- CATKinRigidJoint
- CATKinRevoluteJoint
- CATKinPrismaticJoint
- CATKinCylindricalJoint
- CATKinSphericalJoint
- Weld Joint
- Revolute Joint
- Prismatic Joint
- Cylindrical Joint
- Spherical Joint
- Fixed Parts