Exercise 26: Create Mechanisms using Universal Planar Links

2017-01-20: Tim C. Lueth, MIMED - Technische Universität München, Germany (URL: http://www.mimed.de)

Contents

- Summary of the already explained aspects of the VLFL-Toolbox
- Motivation for this tutorial
- 1. Create a SimMultiBody System for a Fourbar-Linkage
- 2. Now Create a Specific Configuration (Pose) and Write a STL-Files
- 3. Now Analyze the Stucture and Group the Solids to Parts
- 4. Now Arrange all Parts for Printing as Separated Solids
- 5. Now Write the Separated Parts into Different STL Files
- 6. Create a Video of the Linkage Simulation
- Final Remarks

Summary of the already explained aspects of the VLFL-Toolbox

The previous examples of this series covered the following topics:

- Example 01: First Steps Using the VLFL-Toolbox for Solid Object Design
- Example 02: Using the VLFL-Toolbox for STL-File Export and Import
- Example 03: Closed 2D Contours and Boolean Operations in 2D
- Example 04: 2.5D Design Using Boolean Operators on Closed Polygon Lists (CPL)
- Example 05: Creation, Relative Positioning and Merging of Solid Geometries (SG)
- Example 06: Relative Positioning and Alignment of Solid Geometries (SG)
- Example 07: Rotation of Closed Polygon Lists for Solid Geometry Design
- Example 08: Slicing, Closing, Cutting and Separation of Solid Geometries
- Example 09: Boolean Operations with Solid Geometries
- Example 10: Packaging of Sets of Solid Geometries (SG)
- Example 11: Attaching Coordinates Frames to Create Kinematik Models
- Example 12: Define Robot Kinematics and Detect Collisions
- Example 13: Mounting Faces and Conversion of Blocks into Leightweight-structures
- Example 14: Manipulation Functions for Closed Polygons and Laser Cutting (SVG)
- Example 15: Create a Solid by 2 Closed Polygons
- Example 16: Create a Solid along a contour
- Example 20: Programmaticaly Interface to SimMechanics Multi-Body Toolbox
- Example 21: Programmaticaly Design of SimMechanisms Mechanisms and Simulink Signals
- Example 22: Creation of fixed nodes for the assembly of joints
- Example 23: Automatic Creation of a Missing Link and 3D Print of a Complete Model
- Example 24: Automatic Creation of a Joint Limitations
- Example 25: Automatic Creation of Video Titels, Endtitels and Textpages
- Example 26: Create Mechanisms using Universal Planar Links

- Example 27: Fourbar-Linkage: 2 Pose Syntheses and Linkage Export for 3D Printing
- Example 28: Fourbar-Linkage: 3 Pose Syntheses and Linkage Export for 3D Printing

```
% function VLFL_EXP26
```

Motivation for this tutorial

Ein Mechanismus besteht aus zwei Grundelementen: a) Gelenke und b) Glieder, die diese Gelenke verbinden. Bei der "automatischen Konstruktion" von Mechanismen ist es hilfreich sich auf eine der beiden Elemente zu beschränken. Dies wurde in den vorherigen Tutorials bereits genutzt. In diesem Tutorial wird nun ein neues Verfahren vorgestellt. Es sind "Universelle Planare Glieder". Diese bestehen aus einem einfachen Gelenkglied sowie zwei Hälften eines rotatorischen Gelenks. Werden zwei Glieder an einem der Endpunkte miteinander verbunden, dann verbinden sich die beide Hälften aufgrund einer räumlichen Überlappung zu einer Drehachse und es wird automatisch ein Drehgelenk gebildet. Wird ein Gliedenden nicht verbunden, dann bleibt dort dennoch eine Drehachse erhalten. Jede Drehachse kann mit "Knöpfen oder Antriebsmechanismen relativ zum Gelenk verbunden werden und in ihrem Winkelbereich eingeschränkt werden. Die Glieder können in festen Ebenen verbunden werden. Dies erlaubt sowohl eine kollisionsfrei Bewegung unter Berücksichtigung der Glieder als auch die Berücksichtigung von Antriebselementen sofern diese in Vielfachen der Gliederhöhe ausgelegt werden. Dieses Tutorial zeigt nun die Verwendung der Universelle Planbaren Glieder an einem einfachen Beispiel.

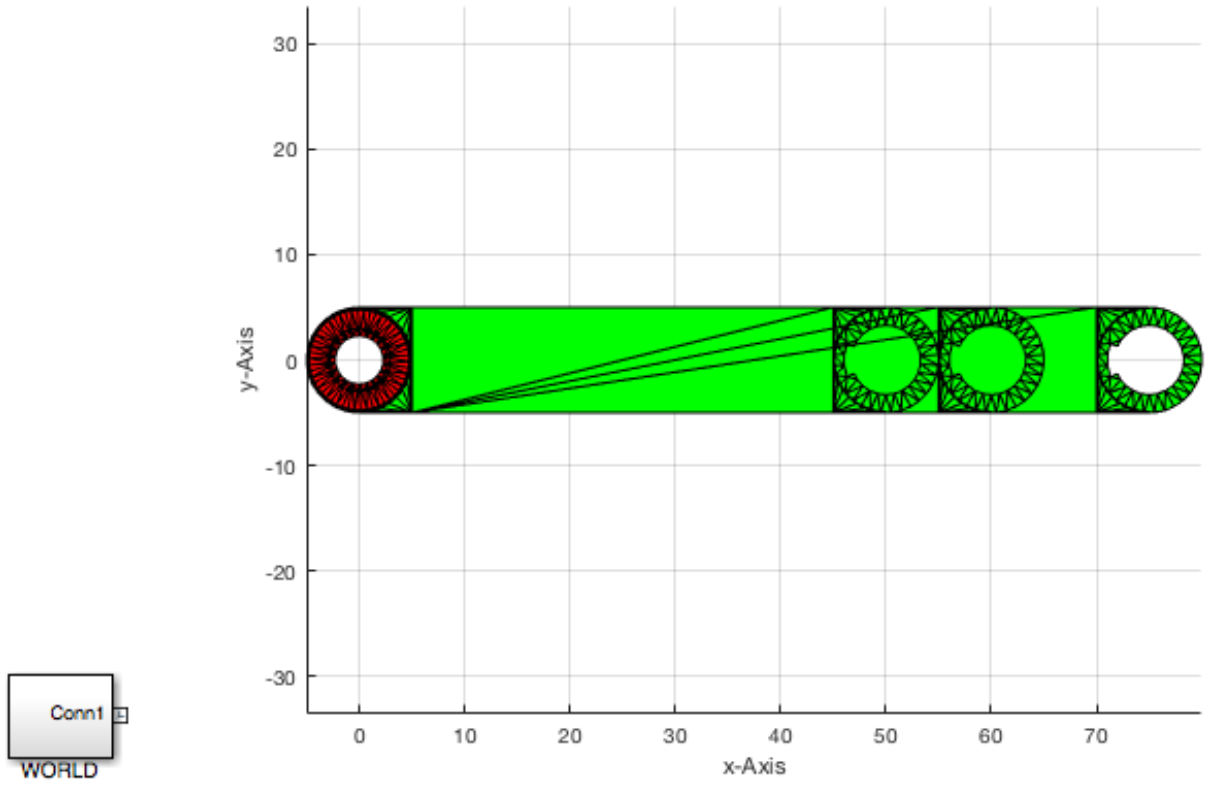
1. Create a SimMultiBody System for a Fourbar-Linkage

```
smbNewSystem ('SG_LIB_EXP_26')   % Creates the mechansim diagramm

L1=75;
L2=60;
L3=50;
L4=50;

A=SGmodelLink2(L1,0,1,'BL,FL'); A.col='r';
B=SGmodelLink2(L2,0,1);         B.col='g';
C=SGmodelLink2(L3,0,-1);        C.col='y';
D=SGmodelLink2(L4,0,-1);        D.col='m';
```

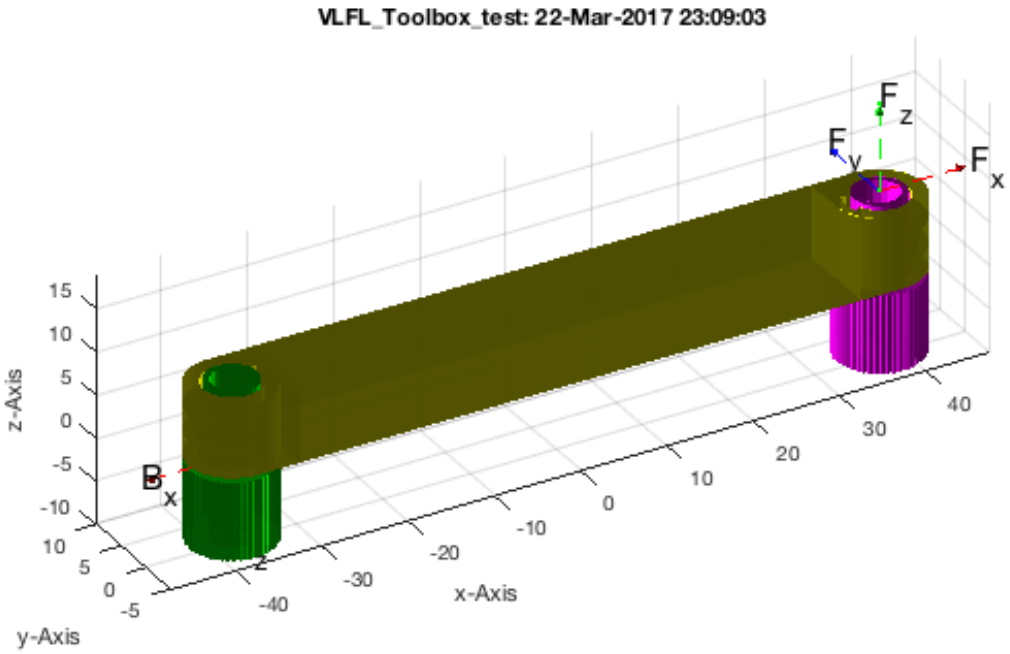
```
Creating temporary directory '/Users/lueth/Desktop/tmp_SG_LIB_EXP_26/'
```



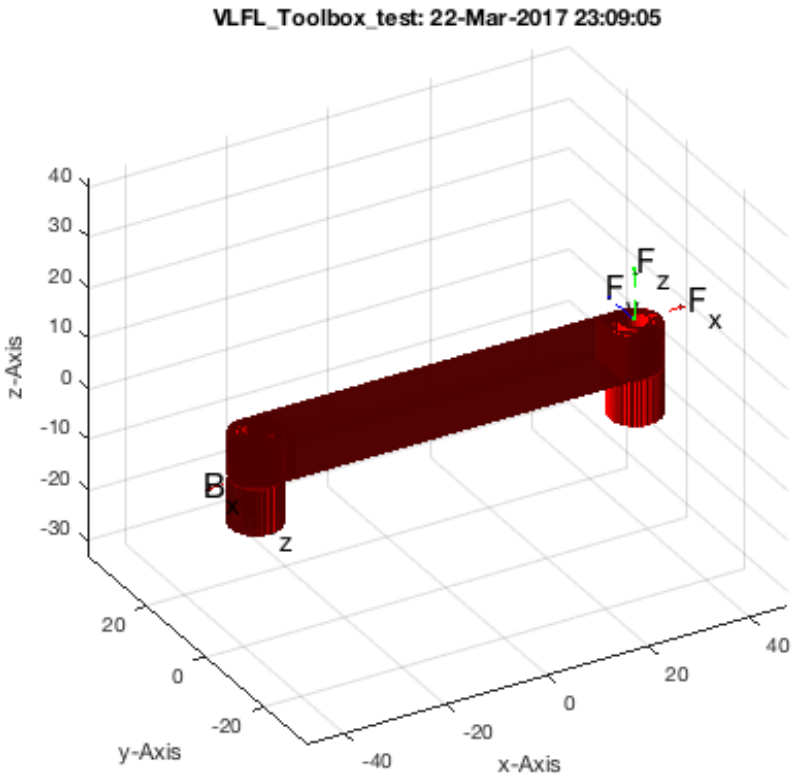
Show the components of the link

```
SGanalyzeGroupParts(A); SGTframeplot(A);
```

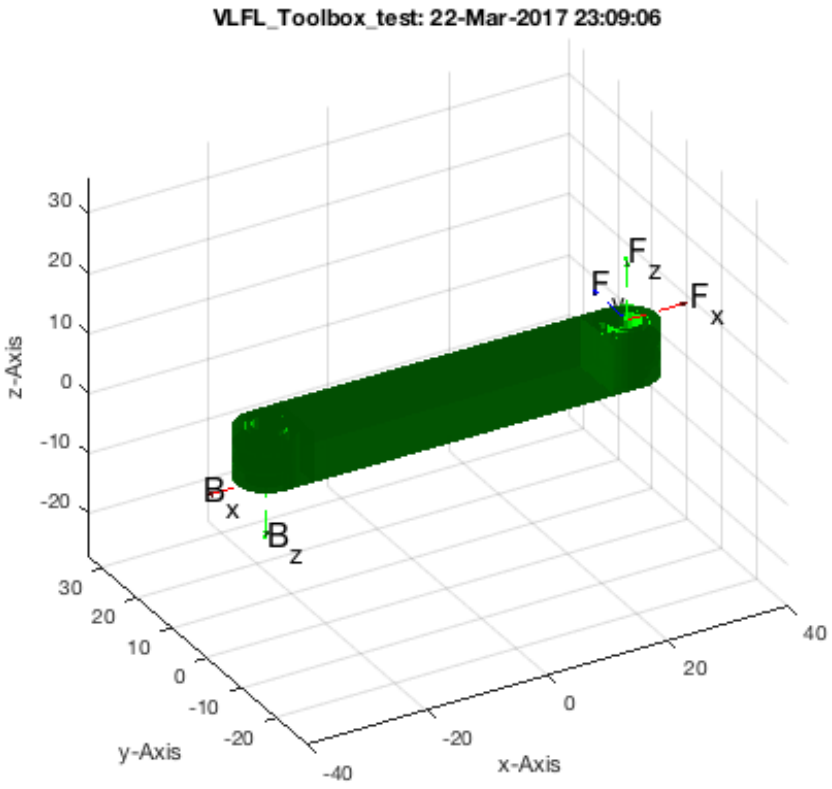
```
5% 10% 15% 20% 25% 30% 35% 40% 45% 50% 55% 60% 65% 70% 75% 80% 90% 95% 100%
SGanalyzeGroupParts: 3 separated parts found.
```



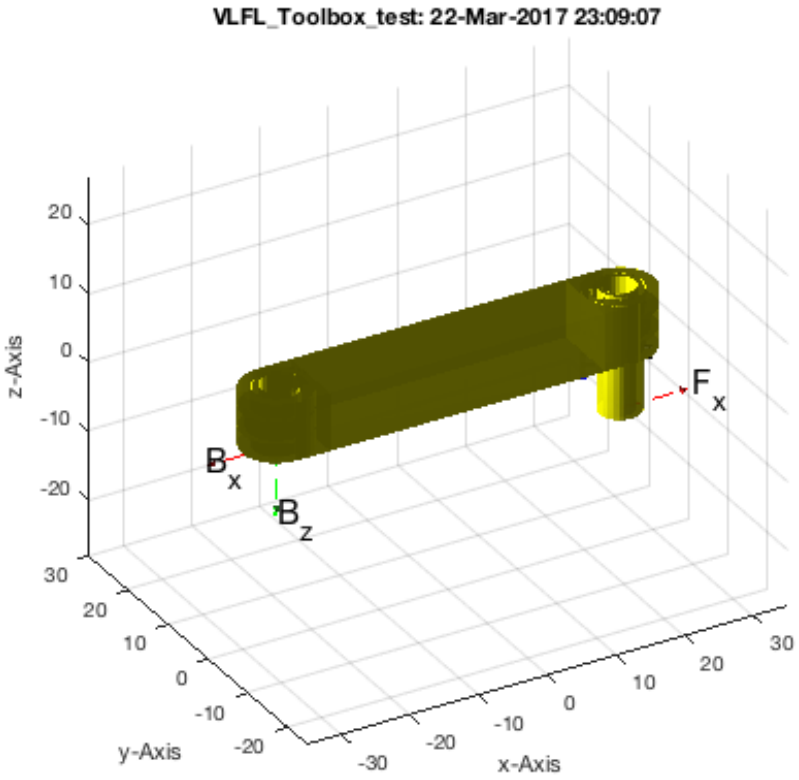

```
SGfigure; SGTplot(A); view(-30,30);
```



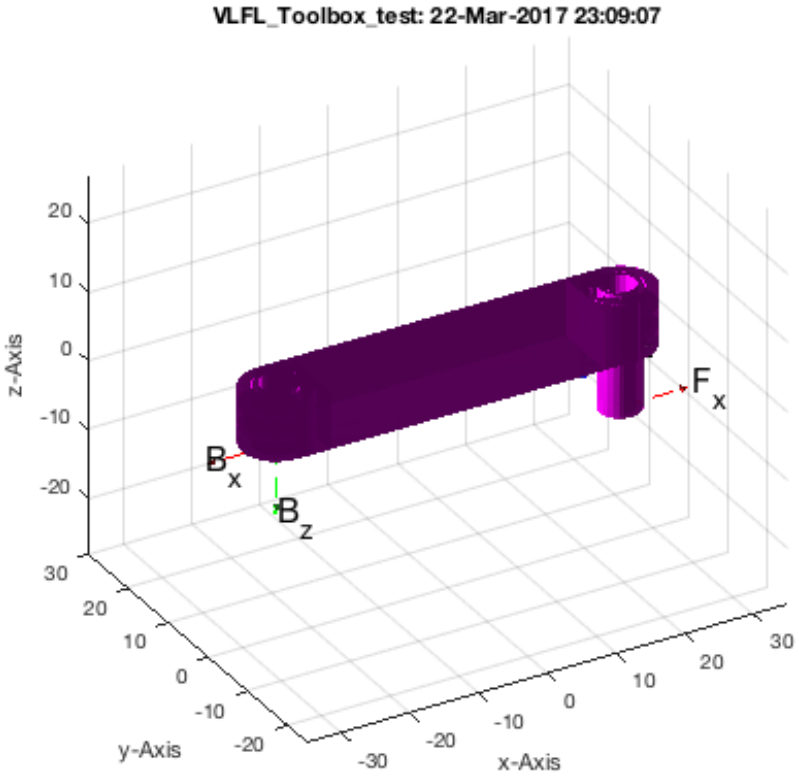
```
SGfigure; SGTplot(B); view(-30,30);
```




```
SGfigure; SGTplot(C); view(-30,30);
```

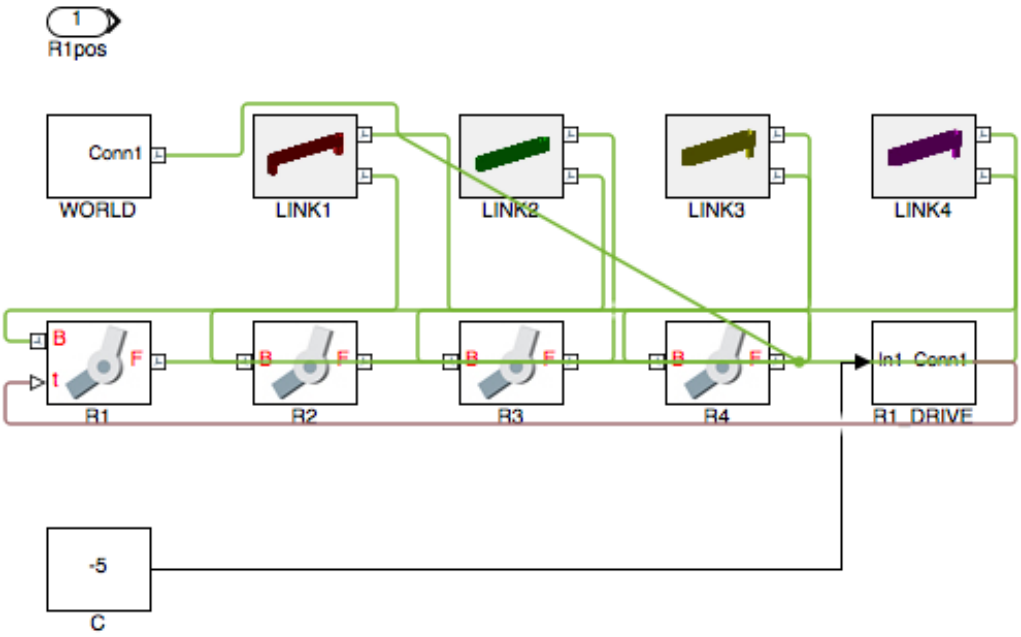



```
SGfigure; SGTplot(D); view(-30,30);
```



```
smbCreateSG (A,'LINK1','r');              % Add long rod as LINK1
smbCreateSG (B,'LINK2','g');              % Add short rod as LINK2
smbCreateSG (C,'LINK3','y');              % Add long rod as LINK3
smbCreateSG (D,'LINK4','m');              % Add short rod as LINK4
smbCreateJoint ('R','R1','LINK1.F','LINK2.B'); % Add a RR Joint
smbCreateJoint ('R','R2','LINK2.F','LINK3.B'); % Add a RR Joint
smbCreateJoint ('R','R3','LINK3.F','LINK4.B'); % Add a RR Joint
smbCreateJoint ('R','R4','LINK4.F','LINK1.B'); % Add a RR Joint

smbCreateConnection('WORLD.ORIGIN','LINK1.B'); % Connect Linkage to World Frame
smbCreateDrive ('R1');
smbSetJointInputTorque('R1');
smbCreateBlockConst('C','R1_DRIVE/1',-5)
ID=smbDrawNow;
smbSimulate(4);
```



2. Now Create a Specific Configuration (Pose) and Write a STL-Files

```
SG=smbFullModelSimulation(5);
% SG=SGmagnifyVL(SG,'',[100 100 100]);
SGwriteSTL(SG,smbFilename('Universal Planar Link'));
```

```
CREATING A FULL SOLID-MOVEMENT SIMULATION-MODEL 'SG_LIB_EXP_26' THAT RUNS At LEAST 5.00 SECOND
S
================================================================================================
=====
Adding frame sensors for all solids of the model
Add frame sensors for 'LINK1.SG'
Add frame sensors for 'LINK2.SG'
Add frame sensors for 'LINK3.SG'
```

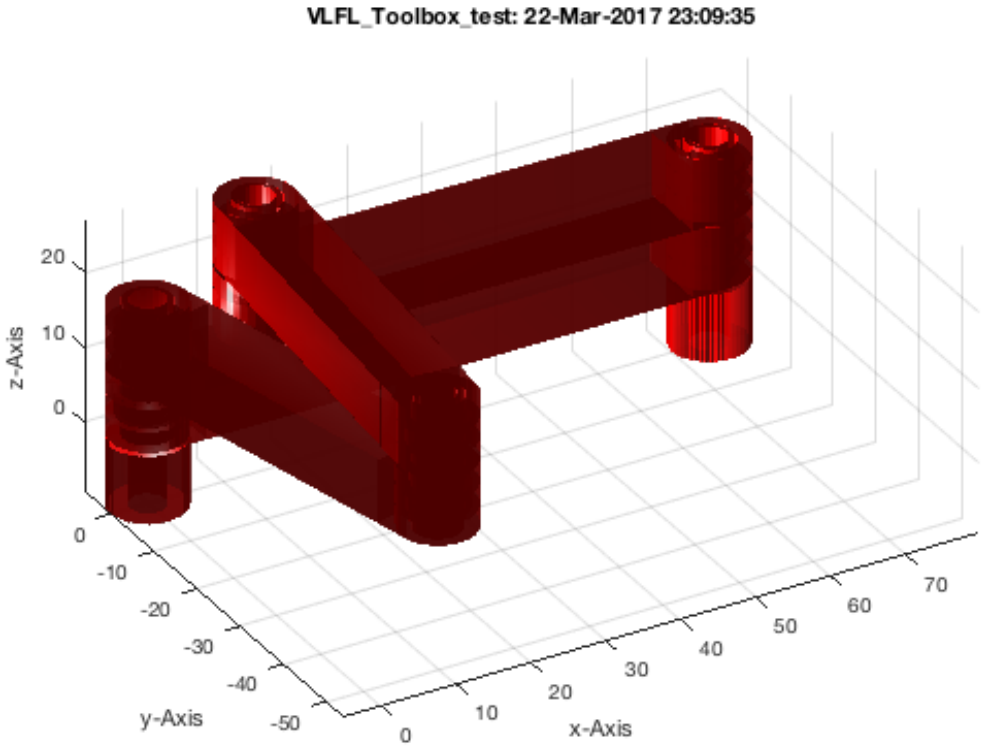
```
Add frame sensors for 'LINK4.SG'
================================================================================================
=====

Simulink.SimulationOutput:

    simlog: [1×1 simscape.logging.Node]
      sout: [1×1 Simulink.SimulationData.Dataset]
      tout: [1097×1 double]
      xout: [1×1 Simulink.SimulationData.Dataset]

Use 'get' to access a variable by name.
Use 'getSimulationMetadata' to access metadata about the simulation.

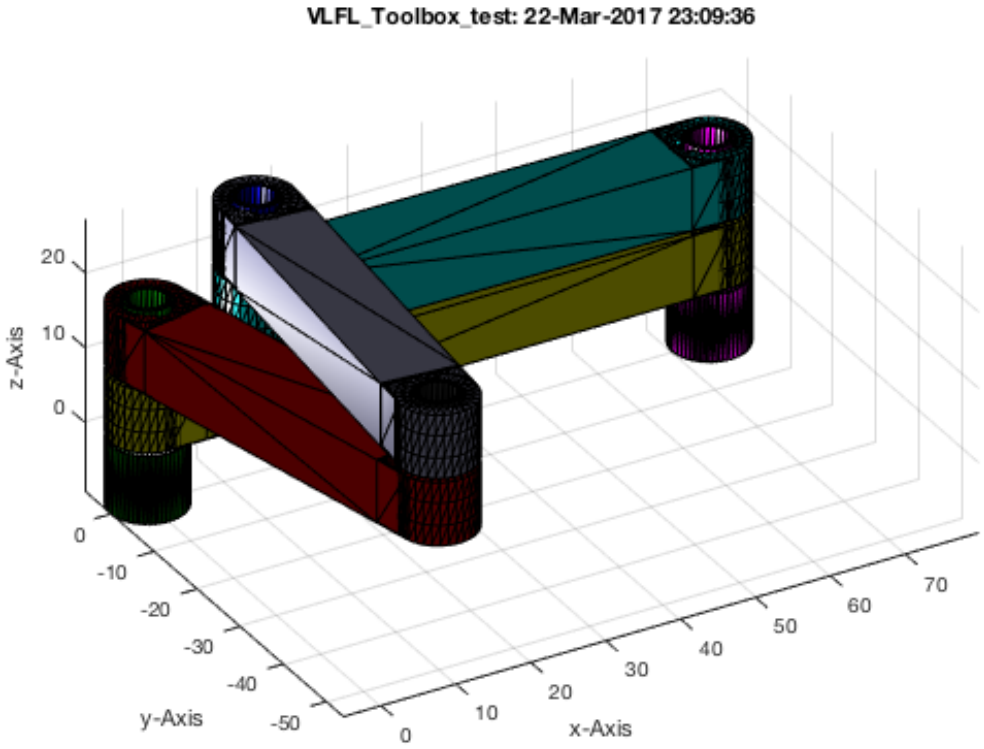
LOADING BINARY STL-File: /Users/lueth/Desktop/tmp_SG_LIB_EXP_26/sbm_temp_LINK1.stl
Header:
Number of facets: 3756
0..LOADING BINARY STL-File: /Users/lueth/Desktop/tmp_SG_LIB_EXP_26/sbm_temp_LINK2.stl
Header:
Number of facets: 2404
0..LOADING BINARY STL-File: /Users/lueth/Desktop/tmp_SG_LIB_EXP_26/sbm_temp_LINK3.stl
Header:
Number of facets: 2584
0..LOADING BINARY STL-File: /Users/lueth/Desktop/tmp_SG_LIB_EXP_26/sbm_temp_LINK4.stl
Header:
Number of facets: 2584
0..
CREATED A SOLID GEOMETRY OF THE FULL SIMULATION-MODEL 'SG_LIB_EXP_26' AT TIME: 5.00 SECONDS
================================================================================================
=====
1000..2000..3000..4000..5000..6000..7000..8000..9000..10000..11000..
```



3. Now Analyze the Stucture and Group the Solids to Parts

```
SGfigure; view(-30,30); SGplot(SG,'m');  SG=SGanalyzeGroupParts(SG); SGplot(SG);
```

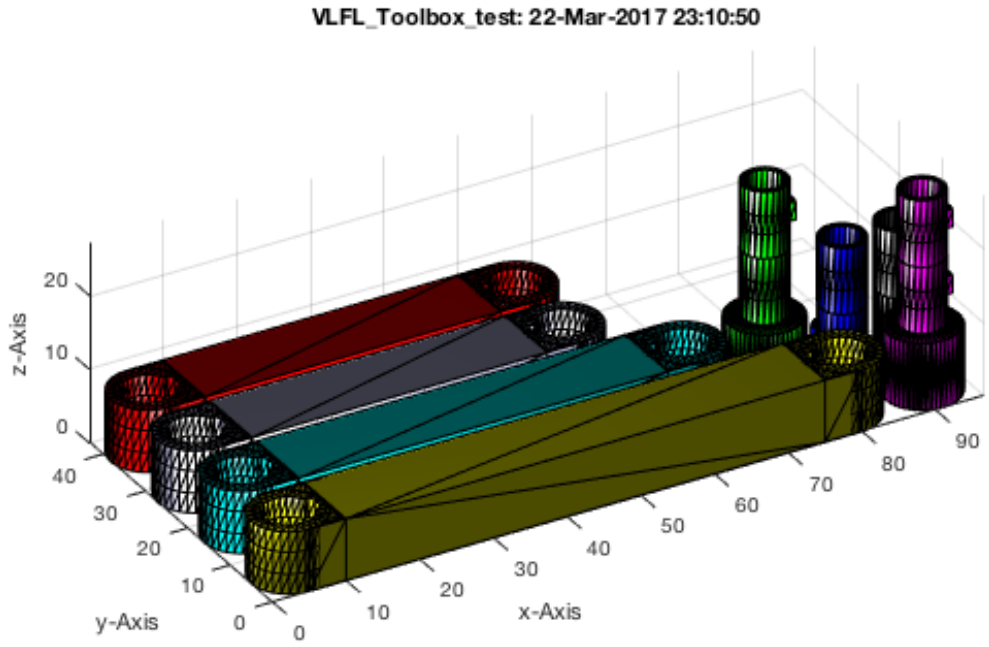
```
5% 10% 15% 20% 25% 30% 35% 40% 45% 50% 55% 60% 65% 70% 75% 80% 85% 90% 95% 100%
```



4. Now Arrange all Parts for Printing as Separated Solids

```
[~,SG]=SGpacking(SG); SGfigure; view(-30,30); SGplot(SG);
```

```
Packing 8 objects (h=61):
```



5. Now Write the Separated Parts into Different STL Files

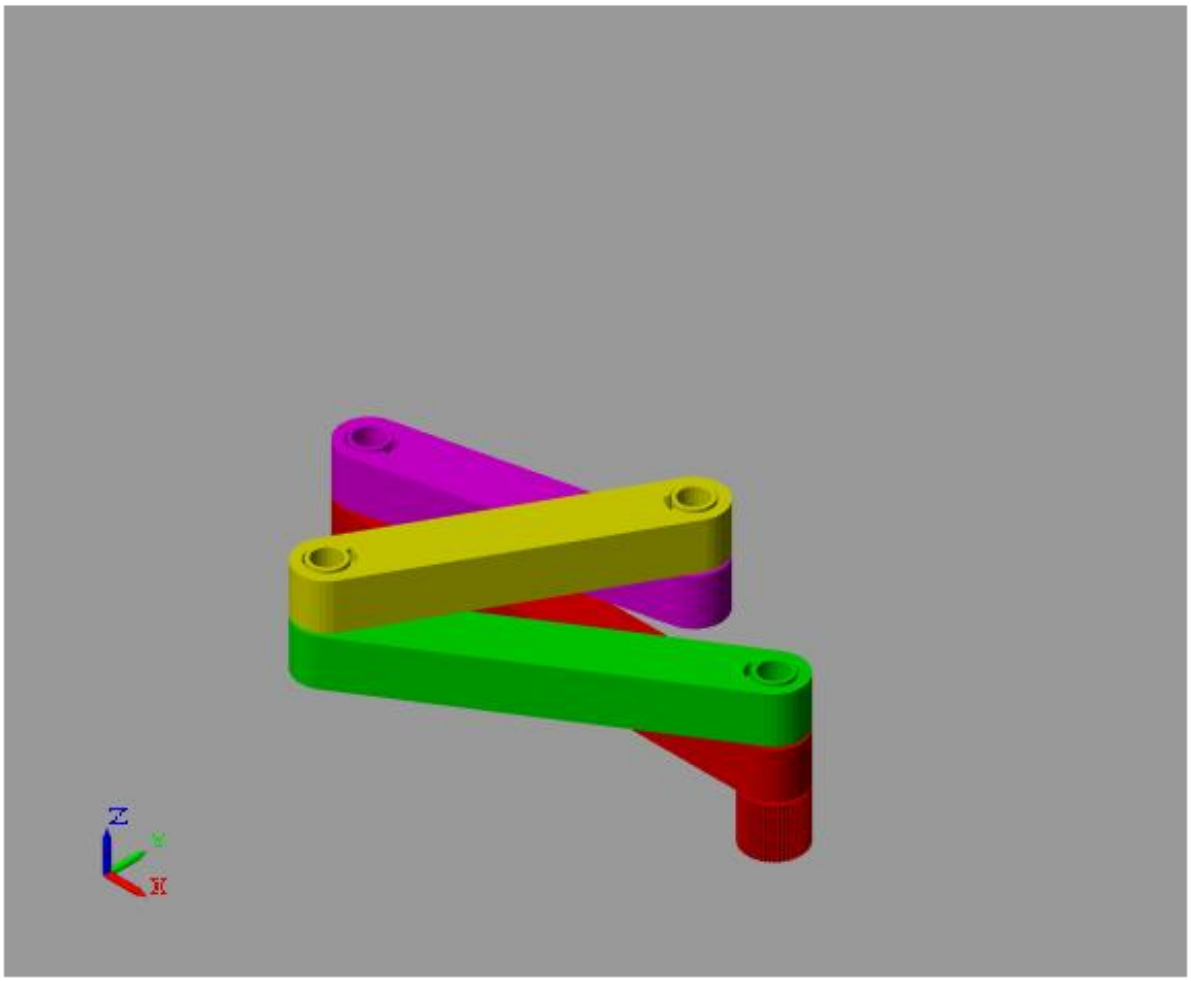
```
SGwriteSeparatedSTL(SG);
```

```
SGwriteSeparatedSTL: Writing 8 STL files in /Users/lueth/Desktop/EXP-2017-03-22/
```

6. Create a Video of the Linkage Simulation

```
[I1,FN]=smbVideoSimulation (4);    % Simulate for 1 second
IT=imageVideoTitle(FN,{'SG-Lib Tutorial #26','Universal Planar Links','Tim C. Lueth','$date'},
'',[0.1 0.2 0.3]);
IE=imageVideoEndtitle(FN);
videoWriteClipMovie(smbFilename('Universal Planar Links SimMultiBody.avi'),IT,2,ID,1,FN,IE,1);
imshow(I1);
```

```
.....Creating a new video file (NO SOUND/2016b): '/Users/lueth/Desktop/tmp_SG_LIB_EXP_26/Unive
rsal Planar Links SimMultiBody.avi'
5% 10% 15% 20% 25% 30% 35% 40% 45% 50% 55% 60% 65% 70% 75% 80% 85% 90% 95% 100%
```



Final Remarks

```
close all
VLFLlicense
```

```
This VLFL-Lib, Rel. 3.6 (2017-Mar-23), is for limited non commercial educational use only!
Licensee: 3.6 Tim Lueth for academic and education 05-Mar-2017 !
Please contact Tim Lueth, Professor at TU Munich, Germany!
WARNING: This VLFL-Lib (Rel. 3.6) license will exceed at 18-Mar-2018!
Executed 22-Mar-2017 23:11:12 by 'lueth' using Matlab 9.1.0.441655 (R2016b) on a MACI64
======================================== Used Matlab products: ================================
=======
image_toolbox
map_toolbox
matlab
robotics_system_toolbox
simmechanics
simscape
simulink
================================================================================================
=======
```

Published with MATLAB® R2016b