Exercise 28: Fourbar-Linkage: 3 Pose Syntheses and Linkage Export for 3D Printing

2017-01-08: Tim C. Lueth, MIMED - Technische Universität München, Germany (URL: http://www.mimed.de)

Contents

- 1. Summary of the already explained aspects of the VLFL-Toolbox
- 2. Motivation for this tutorial
- 3. Show a Two Pose Problem
- 4. Find a General Solutions for the Three Pose Problem
- 5. Find a special Solution for the 4Bar-Linkage wit A0 and B0 on same level
- 6. Create a SimMultiyBody System for the calculated solution
- 7. Show the Video of the Simulation
- 8. Now Create the Solid Geoemtry at time 0.78 seconds
- Final Remarks

1. 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_EXP28
```

2. Motivation for this tutorial

```
smbNewSystem ('SG_LIB_EXP_28',[+5 +5 +5]);           % Creates the mechansim diagramm
```

```
Creating temporary directory '/Users/lueth/Desktop/tmp_SG_LIB_EXP_28/'
```



3. Show a Two Pose Problem

Poses are described by the start point and end point of a link. The first point of the coupling defines a point, the second point also the direction. In the simplest case, two poses for the design of a four-bar are given.

```
d=[0 0];
C1=[ 0 0];
D1=[40 0];
C2=[60 30];
D2=[100 0];
C3=[90 20];
D3=[50 20];

SGfigure;
PLplot([C1;D1],'r-',2);
PLplot([C2;D2],'r-',2);
PLplot([C3;D3],'r-',2);
```



4. Find a General Solutions for the Three Pose Problem

As a solution, there are two straight lines on each of which the frame point A0 or the frame point B0 may be located. The intersection point of these two lines is the pole point P12. In a special case, both frame points are located at this point and a triangle is formed from the four-bar. In any case, the rack points can be displaced such that other secondary conditions can also be fulfilled.

```
l=500

imageFigureMovie('record');
synth4Bar3Pose(C1,D1,C2,D2,C3,D3,d);
% exp_2017_01_08(C1,D1,C2,D2,C3,D3,l,d);
[~,FN2]=imageFigureMovie('write',smbFilename('synth4Bar3Pose.avi'));
```

```
l =

   500


ans =

    20     0     0


ans =

    76    18     0
```

```
ans =

    70    20     0

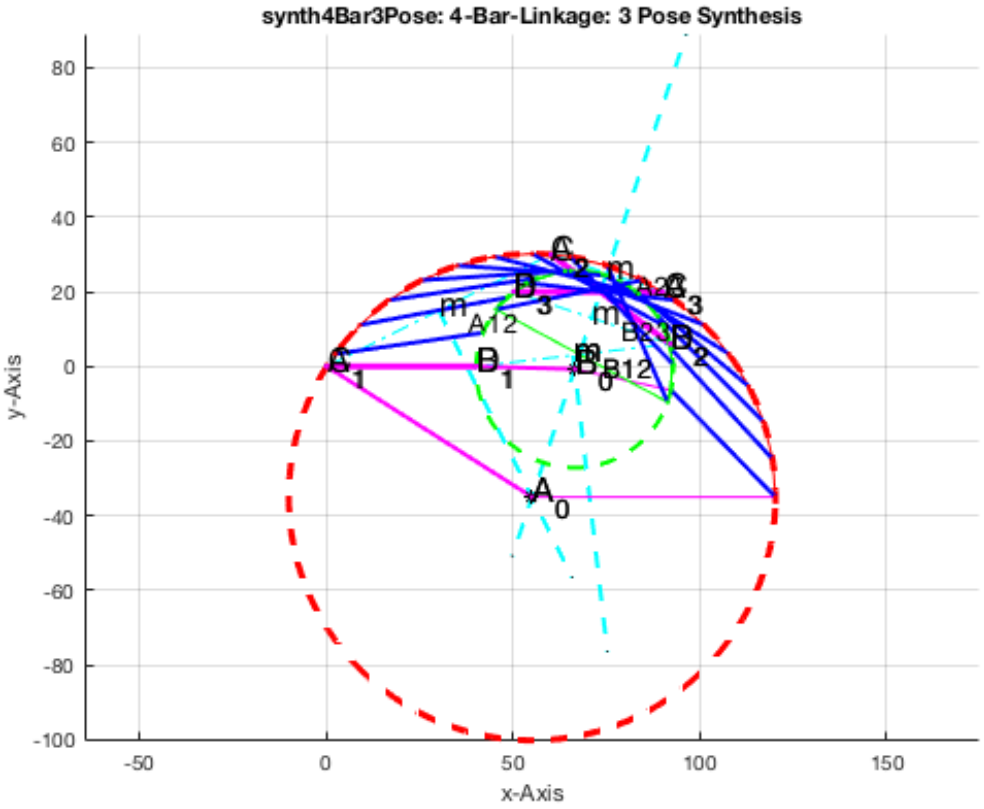

ans =

   27.5000  -17.5000         0


ans =

   53.2143   -0.3571         0

imageFigureSaveMovie: Writing figure movie with 42 frames in file: /Users/lueth/Desktop/tmp_SG
_LIB_EXP_28/synth4Bar3Pose.avi.
```



5. Find a special Solution for the 4Bar-Linkage wit A0 and B0 on same level

If A0 and B0 are to lie on the same plane, B0+k*(P12-B0) must correspond to the Y coordinate of the Y coordinate of A0 in the y coordinate

```
[A0,B0,A1,B1]=synth4Bar3Pose(C1,D1,C2,D2,C3,D3,d);
% [A0,B0,A1,B1]=exp_2017_01_08(C1,D1,C2,D2,C3,D3,l,d);
A0
B0

L1=norm(B0-A0)
L2=norm(B1-B0)
L3=norm(A1-B1)
```

```
L4=norm(A0-A1)

% L1=500; L2=400; L3=500; L4=400

L=[L1 L2 L3 L4]
LMax=find((L==max(L))); LMax=LMax(1);
LMin=find((L==min(L))); LMin=LMin(1);
l1=L(LMin)+L(LMax);
l2=sum(L)-l1;
l3=l1-l2
```

```
A0 =

   55.0000  -35.0000


B0 =

   66.4286   -0.7143


L1 =

   36.1403


L2 =

   26.4382


L3 =

    40


L4 =

   65.1920


L =

   36.1403   26.4382   40.0000   65.1920


l3 =

   15.4899
```

6. Create a SimMultiyBody System for the calculated solution

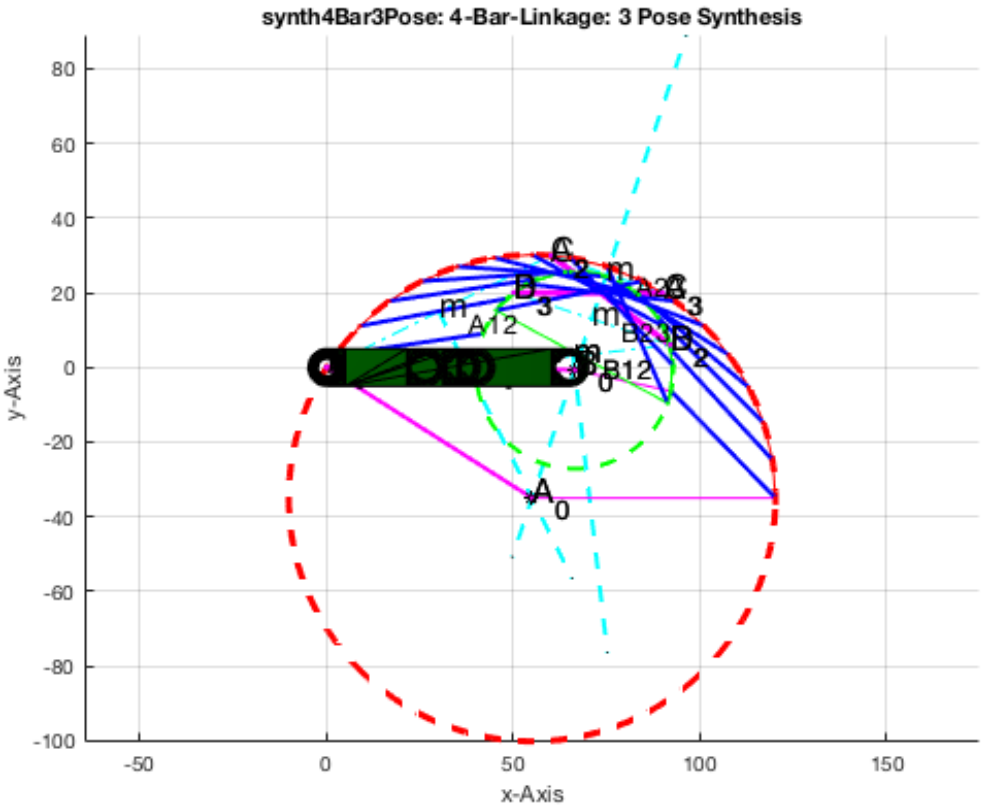
```
A=SGmodelLink(L1,'',1,2); A=SGmodelLink2(L1,0,1);
B=SGmodelLink(L2,'',1,2); B=SGmodelLink2(L2,0,1);
C=SGmodelLink(L3,'',1,2); C=SGmodelLink2(L3,0,-1);
```

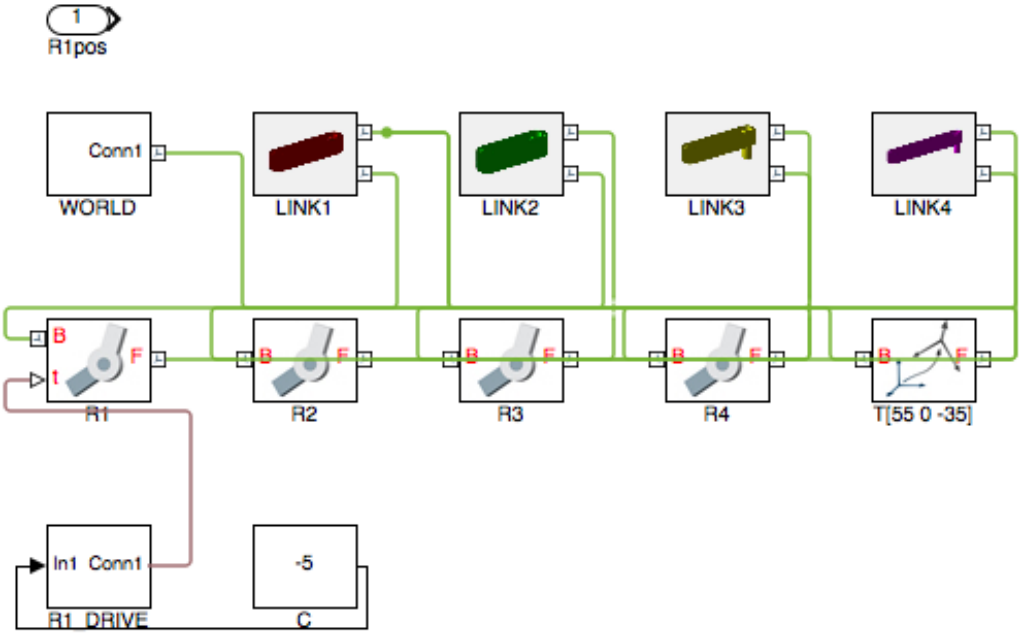
```
D=SGmodelLink(L4,'',1,2); D=SGmodelLink2(L4,0,-1);

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
phi=atan2(B0(2)-A0(2),B0(1)-A0(1))
% phi=0;
smbCreateConnection('WORLD.ORIGIN','LINK1.B',TofR(rot(0,0,phi),[A0(1) 0 A0(2)])); % Connect Li
nkage to World Frame
smbCreateDrive ('R1');
smbSetJointInputTorque('R1');
smbCreateBlockConst('C','R1_DRIVE/1',-5)
ID=smbDrawNow;
smbSimulate(4);
```

```
phi =

    1.2490
```

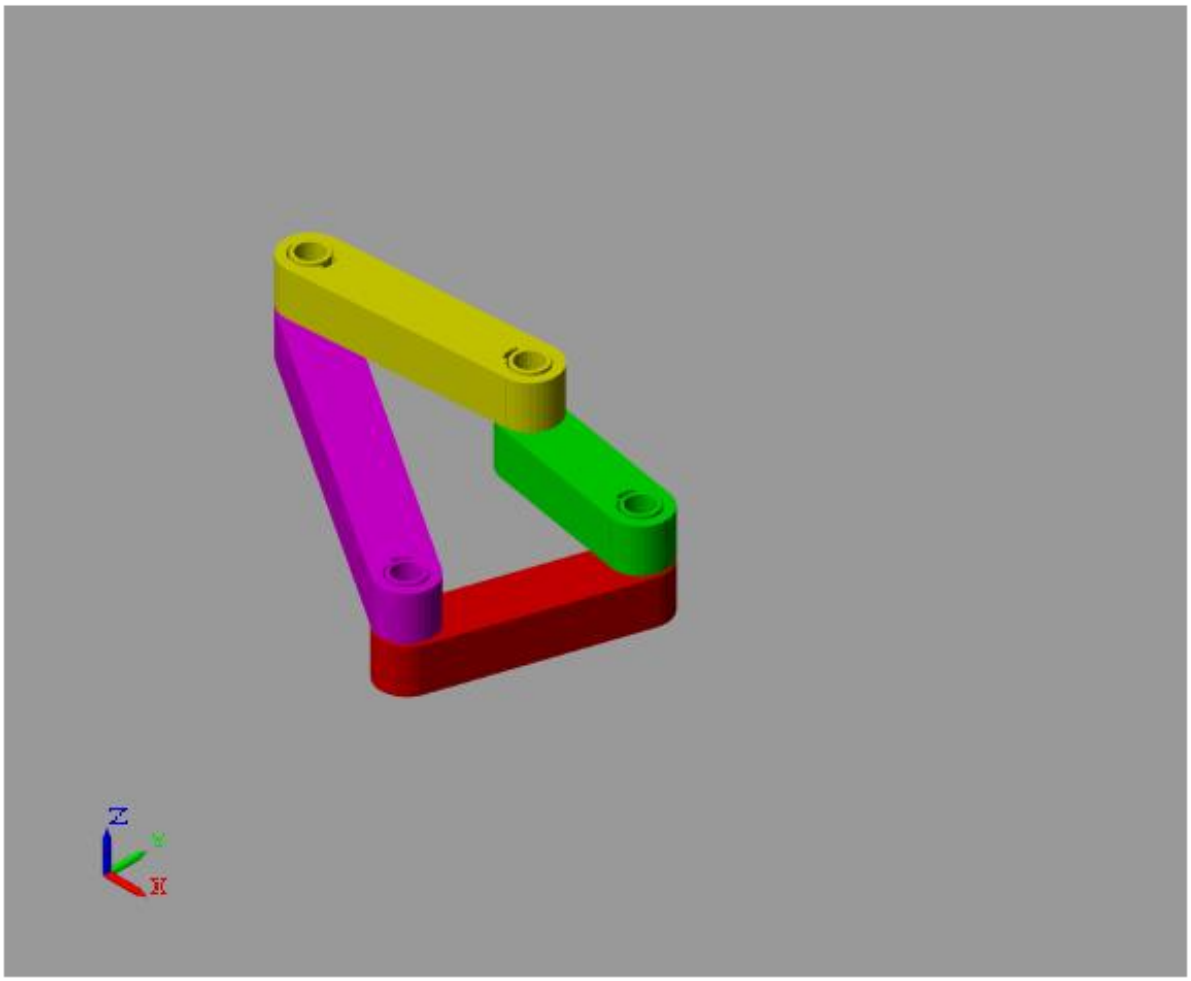




7. Show the Video of the Simulation

```
[I1,FN]=smbVideoSimulation (4);    % Simulate for 1 second
IT=imageVideoTitle(FN,{'SG-Lib Tutorial #27','3 Pose Syntheses','Tim C. Lueth','$date'},'',[0.
44 0.74 1.14]);
IE=imageVideoEndtitle(FN);
videoWriteClipMovie(smbFilename('3 Pose Syntheses SimMultiBody.avi'),IT,2,FN2,ID,1,FN,IE,1);
imshow(I1);
```

```
......Creating a new video file (NO SOUND/2016b): '/Users/lueth/Desktop/tmp_SG_LIB_EXP_28/3 Po
se Syntheses SimMultiBody.avi'
5% 10% 15% 20% 25% 30% 35% 40% 45% 50% 55% 60% 65% 70% 75% 80% 85% 90% 95% 100% 5% 10% 15% 20%
 25% 30% 35% 40% 45% 50% 55% 60% 65% 70% 75% 80% 85% 90% 95% 100%
```



8. Now Create the Solid Geoemtry at time 0.78 seconds

```
SG=smbFullModelSimulation(0.74);
% SG=SGmagnifyVL(SG,'',[100 100 100]);
SGwriteSTL(SG,smbFilename('3-Pose-Synth'));
```

```
CREATING A FULL SOLID-MOVEMENT SIMULATION-MODEL 'SG_LIB_EXP_28' THAT RUNS At LEAST 0.74 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'
================================================================================================
=====

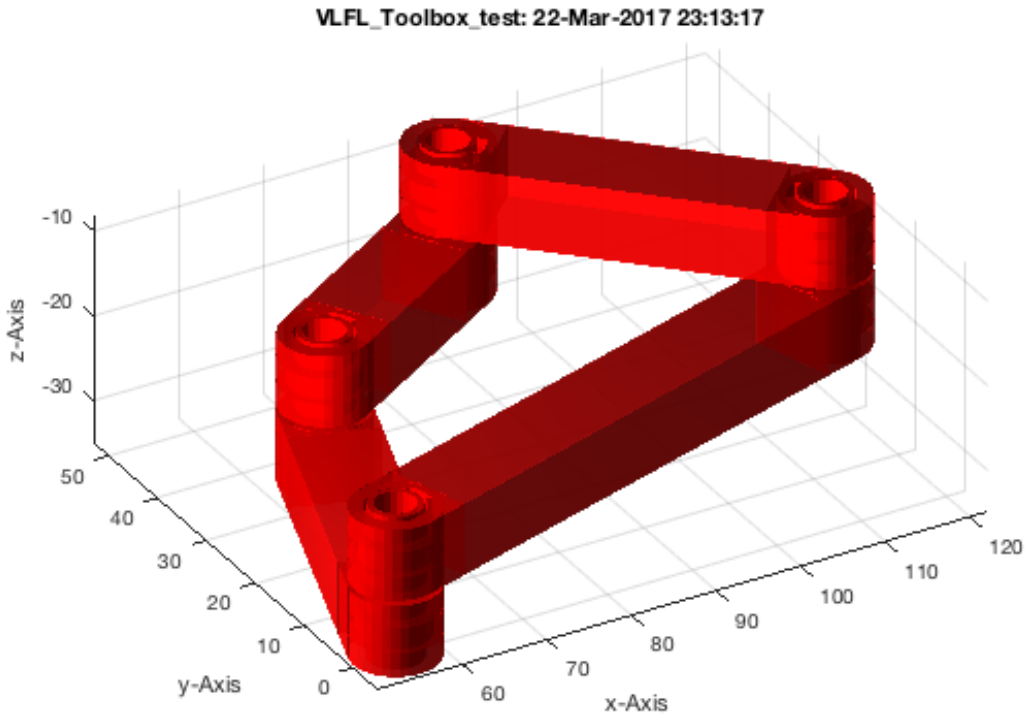
Simulink.SimulationOutput:

    simlog: [1×1 simscape.logging.Node]
      sout: [1×1 Simulink.SimulationData.Dataset]
```

```
      tout: [222×1 double]
      xout: [1×1 Simulink.SimulationData.Dataset]

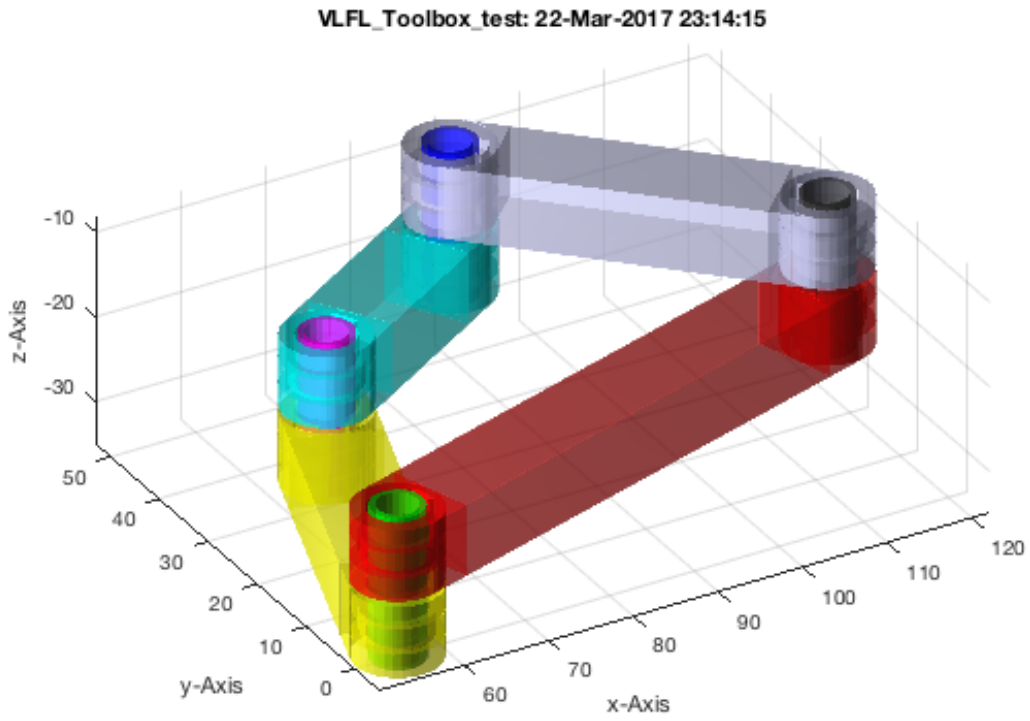
Use 'get' to access a variable by name.
Use 'getSimulationMetadata' to access metadata about the simulation.

LOADING BINARY STL-File: /Users/lueth/Desktop/tmp_SG_LIB_EXP_28/sbm_temp_LINK1.stl
Header:
Number of facets: 2404
0..LOADING BINARY STL-File: /Users/lueth/Desktop/tmp_SG_LIB_EXP_28/sbm_temp_LINK2.stl
Header:
Number of facets: 2404
0..LOADING BINARY STL-File: /Users/lueth/Desktop/tmp_SG_LIB_EXP_28/sbm_temp_LINK3.stl
Header:
Number of facets: 2584
0..LOADING BINARY STL-File: /Users/lueth/Desktop/tmp_SG_LIB_EXP_28/sbm_temp_LINK4.stl
Header:
Number of facets: 2584
0..
CREATED A SOLID GEOMETRY OF THE FULL SIMULATION-MODEL 'SG_LIB_EXP_28' AT TIME: 0.74 SECONDS
================================================================================================
=====
```




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

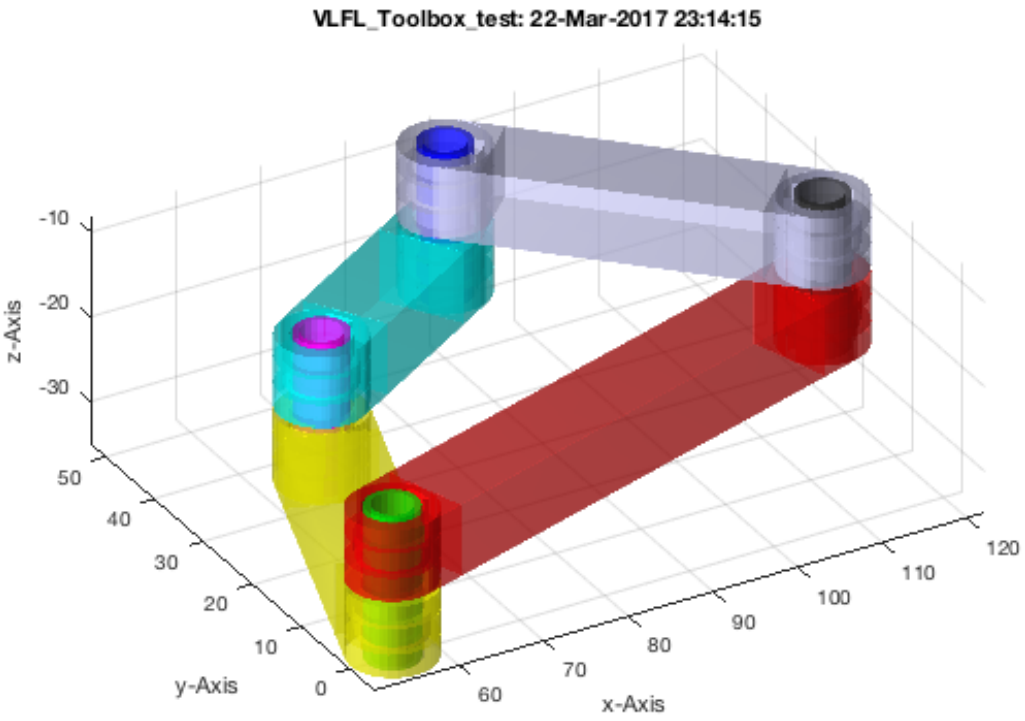
```
5% 10% 15% 20% 25% 30% 35% 40% 45% 50% 55% 60% 65% 70% 75% 80% 85% 90% 95% 100%
SGanalyzeGroupParts: 8 separated parts found.
```



Final Remarks

```
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 23:14:18 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