Tutorial 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) - Last Change: 2017-07-07

Contents

Complete List of all Tutorials with Publishable MATLAB Files of this Solid-Geoemtries Toolbox

The following topics are covered an explained in the specific tutorials:

- Tutorial 01: First Steps Using the VLFL-Toolbox for Solid Object Design
- Tutorial 02: Using the VLFL-Toolbox for STL-File Export and Import
- Tutorial 03: Closed 2D Contours and Boolean Operations in 2D
- Tutorial 04: 2½D Design Using Boolean Operators on Closed Polygon Lists (CPL)
- Tutorial 05: Creation, Relative Positioning and Merging of Solid Geometries (SG)
- Tutorial 06: Relative Positioning and Alignment of Solid Geometries (SG)
- Tutorial 07: Rotation of Closed Polygon Lists for Solid Geometry Design
- Tutorial 08: Slicing, Closing, Cutting and Separation of Solid Geometries
- Tutorial 09: Boolean Operations with Solid Geometries
- Tutorial 10: Packaging of Sets of Solid Geometries (SG)
- Tutorial 11: Attaching Coordinates Frames to Create Kinematik Models
- Tutorial 12: Define Robot Kinematics and Detect Collisions
- Tutorial 13: Mounting Faces and Conversion of Blocks into Leightweight-structures
- Tutorial 14: Manipulation Functions for Closed Polygons and Laser Cutting (SVG)
- Tutorial 15: Create a Solid by 2 Closed Polygons
- Tutorial 16: Create Tube-Style Solids by Succeeding Polygons
- Tutorial 17: Filling and Bending of Polygons and Solids
- Tutorial 18: Analyzing and modifying STL files from CSG modeler (Catia)
- Tutorial 19: Creating drawing templates and dimensioning from polygon lines
- Tutorial 20: Programmatically Interface to SimMechanics Multi-Body Toolbox
- Tutorial 21: Programmatically Convert Joints into Drives (SimMechanics)
- Tutorial 22: Adding Simulink Signals to Record Frame Movements
- Tutorial 23: Automatic Creation of a Missing Link and 3D Print of a Complete Model
- Tutorial 24: Automatic Creation of a Joint Limitations
- Tutorial 25: Automatic Creation of Video Titels, Endtitels and Textpages
- Tutorial 26: Create Mechanisms using Universal Planar Links
- Tutorial 27: Fourbar-Linkage: 2 Pose Syntheses and Linkage Export for 3D Printing
- Tutorial 28: Fourbar-Linkage: 3 Pose Syntheses and Linkage Export for 3D Printing
- Tutorial 29: Create a multi body simulation using several mass points
- Tutorial 30: Creating graphical drawings using point, lines, surfaces, frames etc.
- Tutorial 31: Importing 3D Medical DICOM Image Data and converting into 3D Solids
- Tutorial 32: Exchanging Data with a FileMaker Database
- Tutorial 33: Using a Round-Robin realtime multi-tasking system
- Tutorial 34: 2D Projection Images and Camera Coordinate System Reconstruction
- Tutorial 35: Collection of Ideas for Tutorials
- Tutorial 36: Creating a Patient-Individual Arm-Skin Protector-Shell

Motivation for this tutorial: (Originally SolidGeometry 3.3 required)

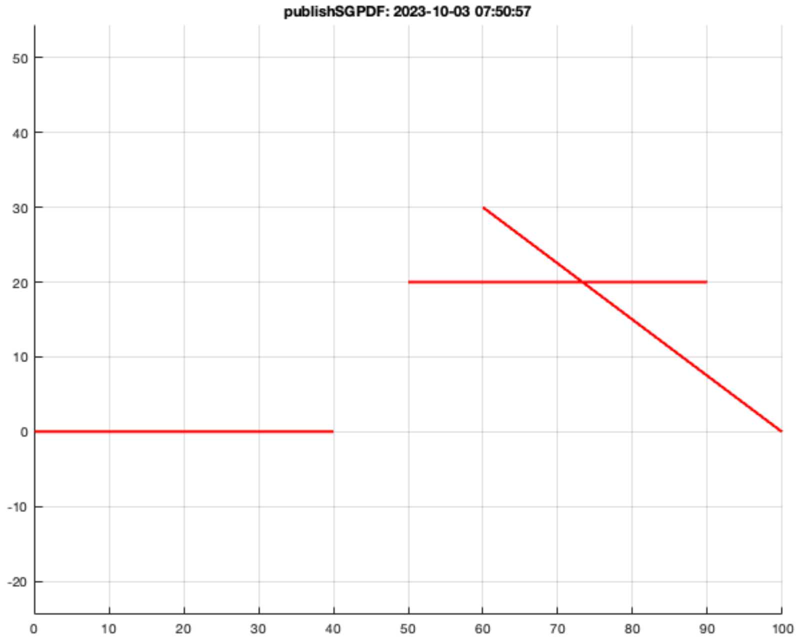
```
smbNewSystem ('SG_LIB_EXP_28',[+5 +5 +5]);           % Creates the mechansim diagramm
```

```
Creating temporary directory '/Users/timlueth/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);
```

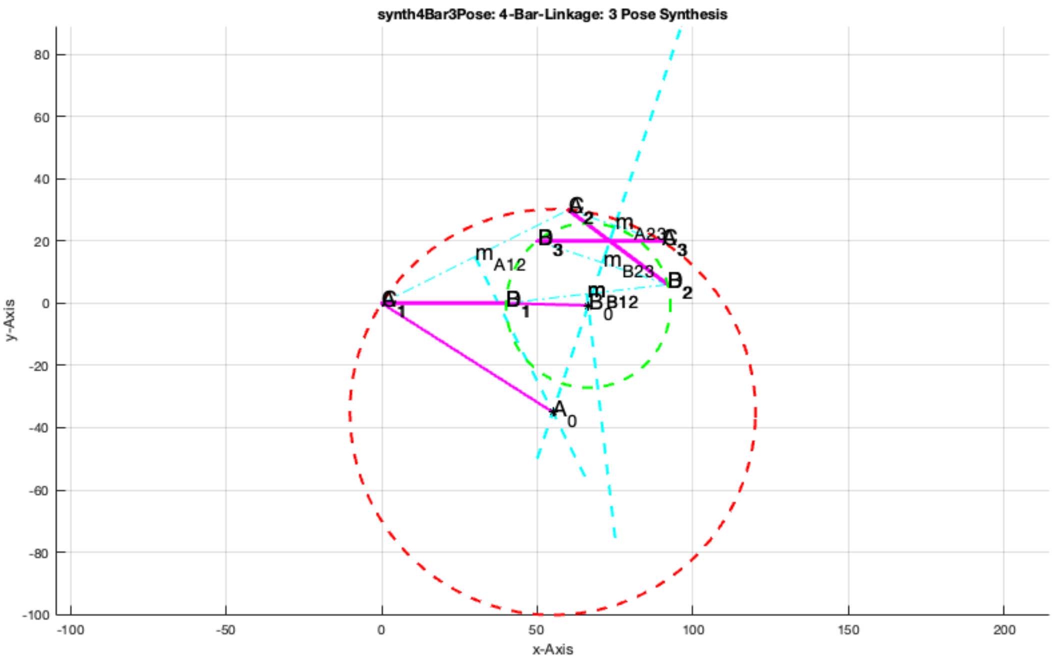


4. Find a General Solutions for the Three Pose Problem

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');                          % Start Image Stack
synth4Bar3Pose(C1,D1,C2,D2,C3,D3,d);
% exp_2017_01_08(C1,D1,C2,D2,C3,D3,l,d);
imageFigureMovie(gcf);                               % RECORD ONE IMAGE
[~,FN2]=imageFigureMovie('write',smbFilename('synth4Bar3Pose.avi')); % SAVE IMAGE AS VIDEO
```

```
l =
   500
imageFigureSaveMovie: Writing figure movie with 1 frames in file: /Users/timlueth/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

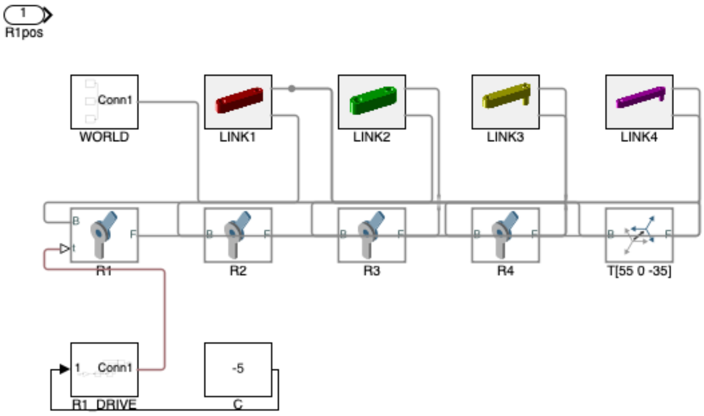
```
[A0,B0,A1,B1]=synth4Bar3Pose(C1,D1,C2,D2,C3,D3,d);
% [A0,B0,A1,B1]=exp_2017_01_08(C1,D1,C2,D2,C3,D3,l,d);
A0
B0

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 Linkage to World Frame
smbCreateDrive ('R1');
smbSetJointInputTorque('R1');
smbCreateBlockConst('C','R1_DRIVE/1',-5)
ID=smbDrawNow;
smbSimulate(4);
```

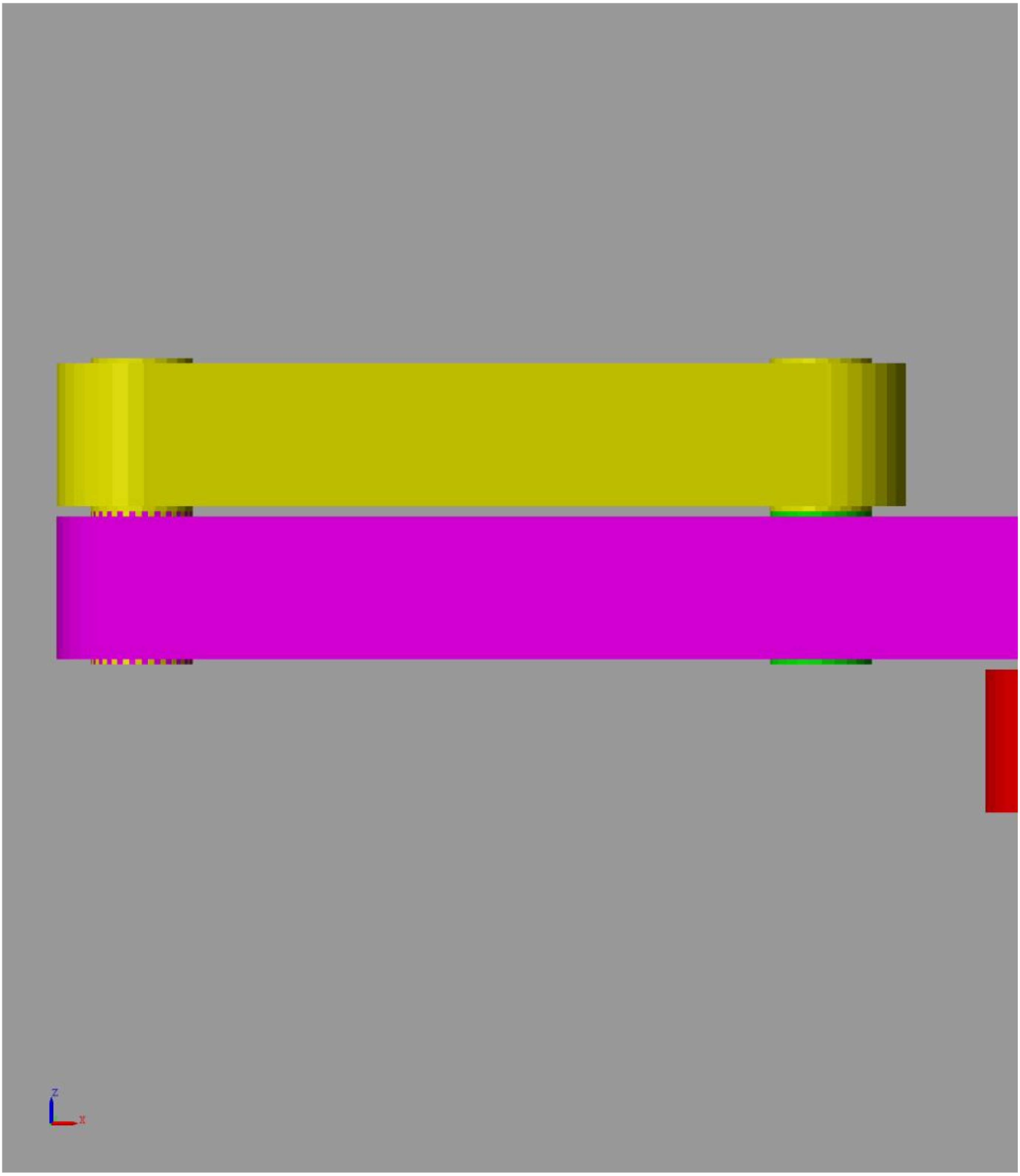
```
Warning: z(2) < z(1) ===>inverse faces
Warning: z(2) < z(1) ===>inverse faces
phi =
    1.2490
```



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);
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/timlueth/Desktop/tmp_SG_LIB_EXP_28/3 Pose Syntheses SimMultiBody.avi'
100% 5% 10% 15% 20% 25% 30% 35% 40% 45% 50% 55% 60% 65% 70% 75% 80% 85% 90% 95% 100%
```

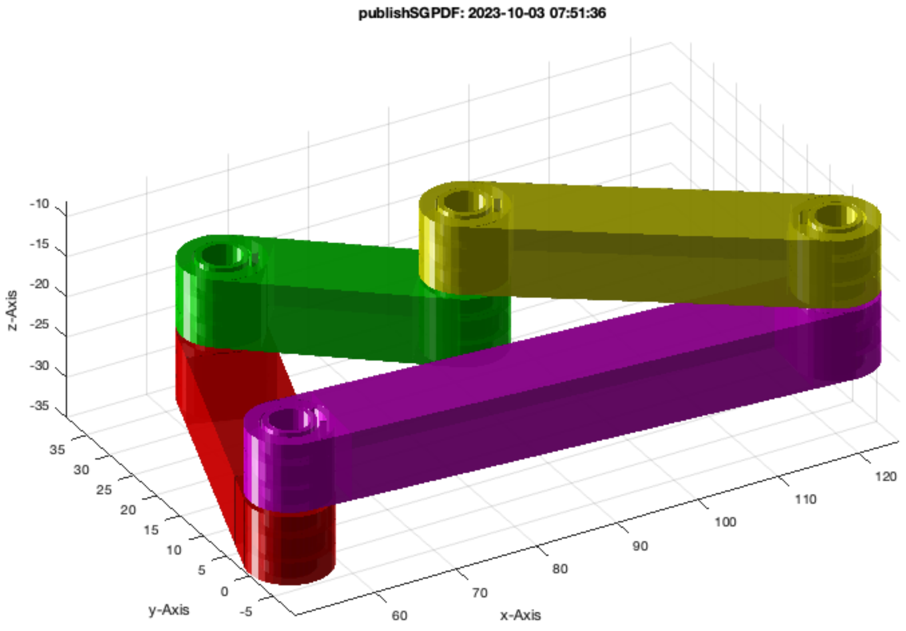


8. Now Create the Solid Geoemtry at time 0.78 seconds

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

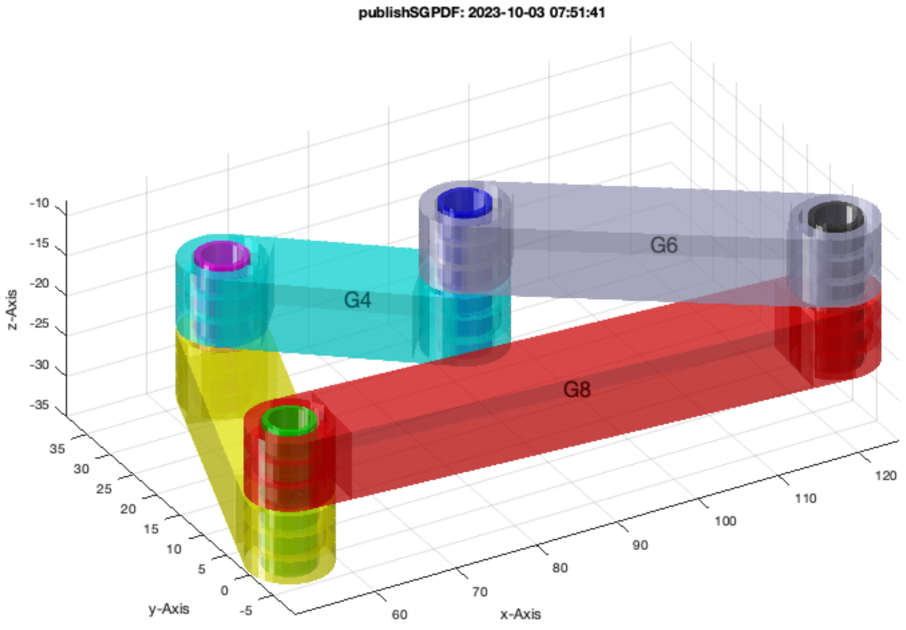
```
CREATING A FULL SOLID-MOVEMENT SIMULATION-MODEL 'SG_LIB_EXP_28' THAT RUNS At LEAST 0.74 SECONDS
===================================================================================================
Adding frame sensors for all solids of the model
```

```
Add frame sensors for 'LINK1.SG'
Add frame sensors for 'LINK2.SG'
Add frame sensors for 'LINK3.SG'
Add frame sensors for 'LINK4.SG'
===================================================================================================
simOut =
  Simulink.SimulationOutput:
                 simlog: [1x1 simscape.logging.Node]
                   sout: [1x1 Simulink.SimulationData.Dataset]
                   tout: [250x1 double]
                   xout: [1x1 Simulink.SimulationData.Dataset]

     SimulationMetadata: [1x1 Simulink.SimulationMetadata]
           ErrorMessage: [0x0 char]

LOADING BINARY STL-File: /Users/timlueth/Desktop/tmp_SG_LIB_EXP_28/sbm_temp_LINK1.stl
Binary Header: COLOR=RGBA,MATERIAL=AAAABBBBCCCCDDDD;SOLID "/Users/timlueth/Desktop/tmp_SG_LIB_E
Color of solid defined as: "k"
Alpha of solid defined as: 65.00
Number of facets: 2494
Number of vertices: 1258
SGN2SGT: 2 Frames ('B', 'F') decoded in STL format.

LOADING BINARY STL-File: /Users/timlueth/Desktop/tmp_SG_LIB_EXP_28/sbm_temp_LINK2.stl
Binary Header: COLOR=RGBA,MATERIAL=AAAABBBBCCCCDDDD;SOLID "/Users/timlueth/Desktop/tmp_SG_LIB_E
Color of solid defined as: "k"
Alpha of solid defined as: 65.00
Number of facets: 2494
Number of vertices: 1258
SGN2SGT: 2 Frames ('B', 'F') decoded in STL format.

LOADING BINARY STL-File: /Users/timlueth/Desktop/tmp_SG_LIB_EXP_28/sbm_temp_LINK3.stl
Binary Header: COLOR=RGBA,MATERIAL=AAAABBBBCCCCDDDD;SOLID "/Users/timlueth/Desktop/tmp_SG_LIB_E
Color of solid defined as: "k"
Alpha of solid defined as: 65.00
Number of facets: 2678
Number of vertices: 1350
SGN2SGT: 2 Frames ('B', 'F') decoded in STL format.

LOADING BINARY STL-File: /Users/timlueth/Desktop/tmp_SG_LIB_EXP_28/sbm_temp_LINK4.stl
Binary Header: COLOR=RGBA,MATERIAL=AAAABBBBCCCCDDDD;SOLID "/Users/timlueth/Desktop/tmp_SG_LIB_E
Color of solid defined as: "k"
Alpha of solid defined as: 65.00
Number of facets: 2678
Number of vertices: 1350
SGN2SGT: 2 Frames ('B', 'F') decoded in STL format.


CREATED A SOLID GEOMETRY OF THE FULL SIMULATION-MODEL 'SG_LIB_EXP_28' AT TIME: 0.74 SECONDS
===================================================================================================
SG =
  4×1 cell array
    {1×1 struct}
    {1×1 struct}
    {1×1 struct}
    {1×1 struct}
publishSGPDF:<a href = "matlab: openbydoubleclick ('/Users/timlueth/Desktop/tmp_SG_LIB_EXP_28')">/Users/timlueth/Desktop/tmp_SG_LIB_EXP_28/</a><a href
```



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

```
4% 8% 12% 16% 20% 24% 28% 32% 36% 40% 44% 48% 52% 56% 60% 64% 68% 72% 76% 80% 84% 88% 92% 96% 100%
SGanalyzeGroupParts: 8 separated parts found
```



Final Remarks

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2023-Oct-03), is for limited non commercial educational use only!
Licensee: Tim Lueth (Development Version)!
Please contact Tim Lueth, Professor at TU Munich, Germany!
WARNING: This VLFL-Lib (Rel. ) license will exceed at 06-Jul-2078 07:51:42!
Executed 03-Oct-2023 07:51:44 by 'timlueth' on a MACI64 using Mac OSX 13.6 | R2023a Update 5 | SG-Lib 5.4
======================================= Used Matlab products: =====================================
distrib_computing_toolbox
```

```
fixed_point_toolbox
map_toolbox
matlab
simmechanics
simscape
simulink
=====================================================================================================
```

Published with MATLAB® R2023a