Tutorial 26: Create Mechanisms using Universal Planar Links

2017-01-20: 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)

```
% function VLFL_EXP26
```

Motivation for this tutorial

A mechanism consists of two basic elements: a) joints and b) links that connect these joints. In the "automatic construction" of mechanisms, it is helpful to limit one of the two elements. This has already been used in the previous tutorials. In this tutorial a new procedure is presented. They are "universal planar links". These consist of a simple joint member and two halves of a rotary joint. If two links are connected to each other at one of the end points, the two halves of the joints are connected to an axis of rotation due to a spatial overlap, and a swivel joint is automatically formed. If a member is not connected, an axis of rotation is still retained there. Each axis of rotation can be connected with "knobs or drive mechanisms relative to the joint and its angular range can be restricted, the links can be

connected in fixed planes, allowing a collision-free movement considering the links as well as the consideration of drive elements. This tutorial now shows you how to use the universal planar links in a simple example.

```
% clear all;
```

1. Create a SimMultiBody System for a Fourbar-Linkage

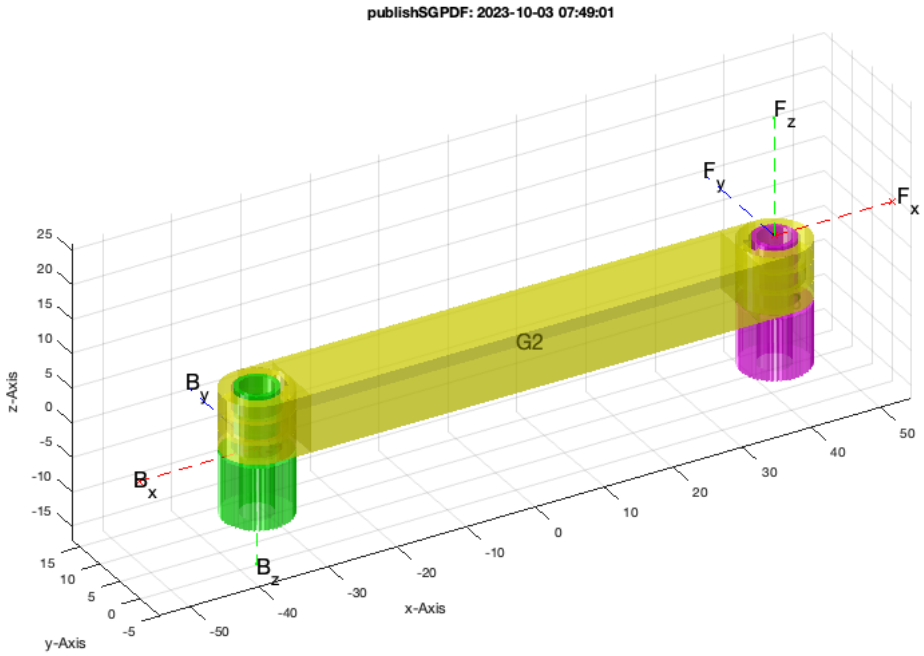
```
smbNewSystem ('SG_LIB_EXP_26')   % Creates the mechansim diagramm

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

L1=75; A=SGmodelLink2(L1,0,1,'BL,FL'); A.col='r';
L2=60; B=SGmodelLink2(L2,0,1);         B.col='g';
L3=50; C=SGmodelLink2(L3,0,-1);        C.col='y';
L4=50;D=SGmodelLink2(L4,0,-1);         D.col='m';
```

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

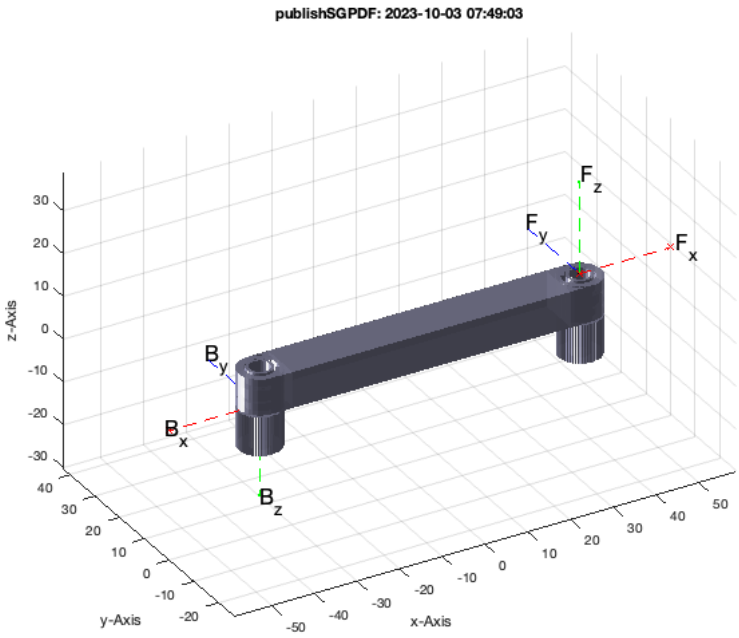
Show the components of the link

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

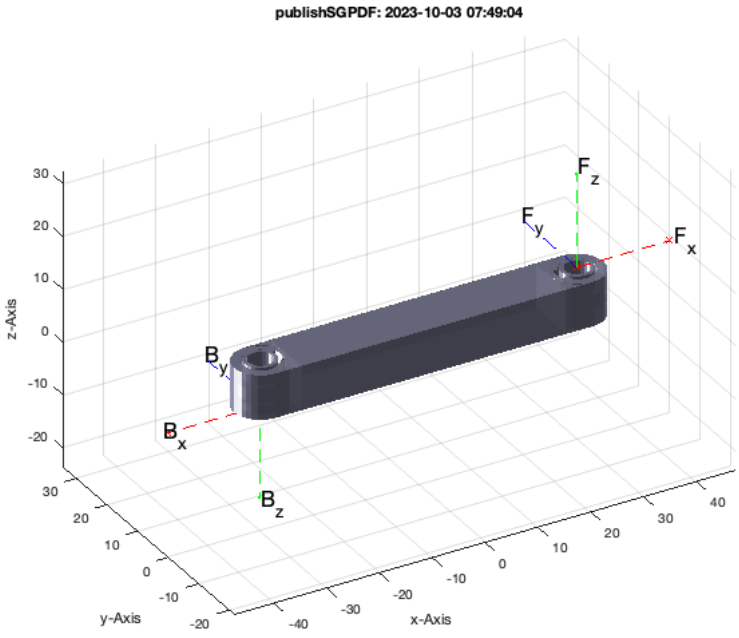
```
8% 12% 16% 20% 24% 28% 32% 36% 40% 44% 48% 52% 56% 60% 64% 68% 72% 76% 80% 84% 88% 92% 96% 100%
SGanalyzeGroupParts: 3 separated parts found
```



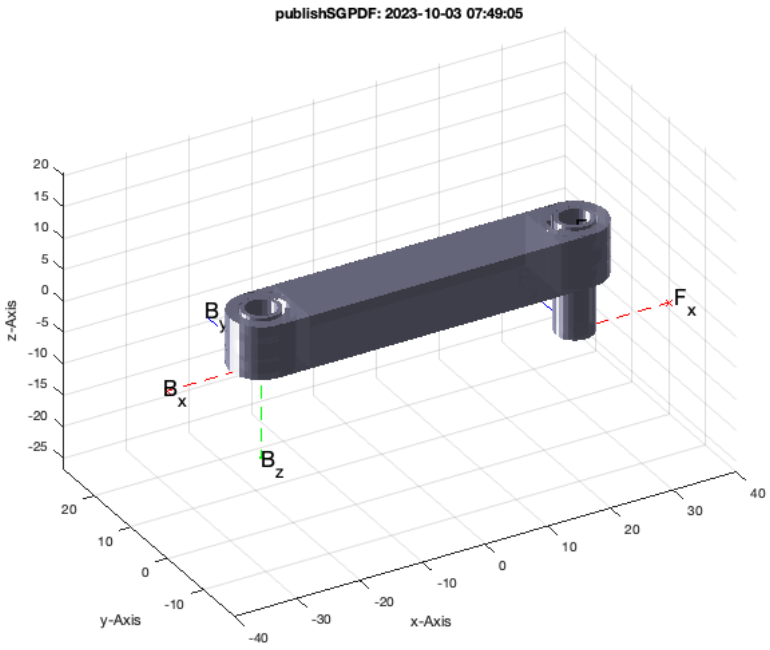

```
SGfigure; SGTplot(A); view(-30,30);
```



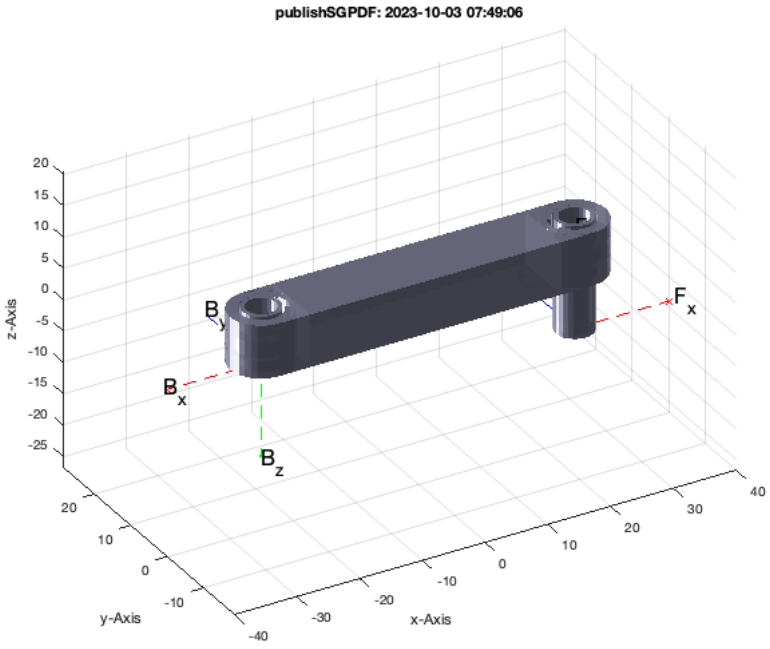

```
SGfigure; SGTplot(B); view(-30,30);
```



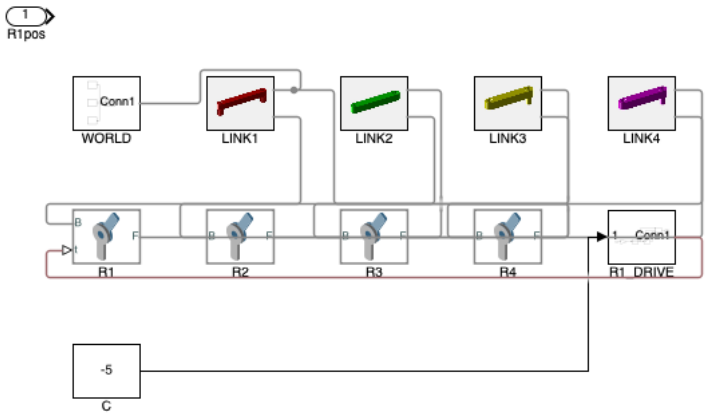

```
SGfigure; SGTplot(C); view(-30,30);
```




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



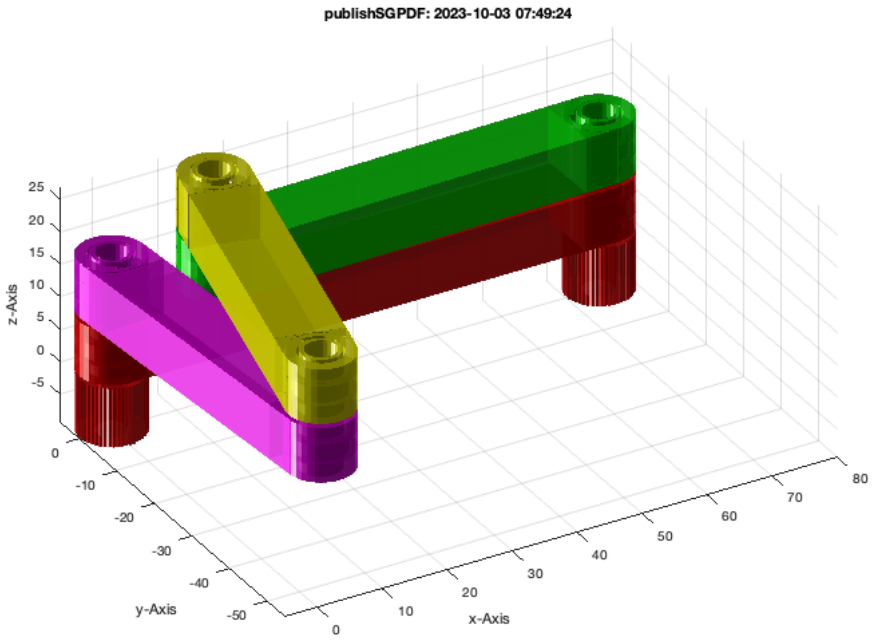

```
smbCreateSG (A,'LINK1','r');              % Add long rod as LINK1
smbCreateSG (B,'LINK2','g');              % Add short rod as LINK2
smbCreateSG (C,'LINK3','y');              % Add long rod as LINK3
smbCreateSG (D,'LINK4','m');              % Add short rod as LINK4
smbCreateJoint ('R','R1','LINK1.F','LINK2.B');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

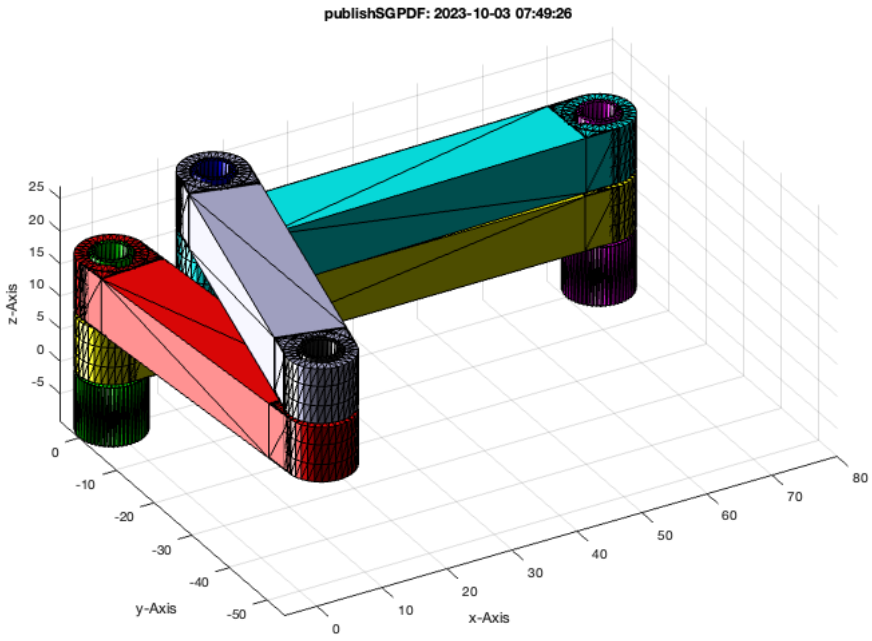
```
CREATING A FULL SOLID-MOVEMENT SIMULATION-MODEL 'SG_LIB_EXP_26' THAT RUNS At LEAST 5.00 SECONDS
=================================================================================================
Adding frame sensors for all solids of the model
Add frame sensors for 'LINK1.SG'
Add frame sensors for '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: [1184x1 double]
                   xout: [1x1 Simulink.SimulationData.Dataset]

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

LOADING BINARY STL-File: /Users/timlueth/Desktop/tmp_SG_LIB_EXP_26/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: 3862
Number of vertices: 1942
SGN2SGT: 2 Frames ('B', 'F') decoded in STL format.

LOADING BINARY STL-File: /Users/timlueth/Desktop/tmp_SG_LIB_EXP_26/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_26/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_26/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_26' AT TIME: 5.00 SECONDS
=================================================================================================
publishSGPDF:<a href = "matlab: openbydoubleclick ('/Users/timlueth/Desktop/tmp_SG_LIB_EXP_26')">/Users/timlueth/Desktop/tmp_SG_LIB_EXP_26/</a><a href
```



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

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

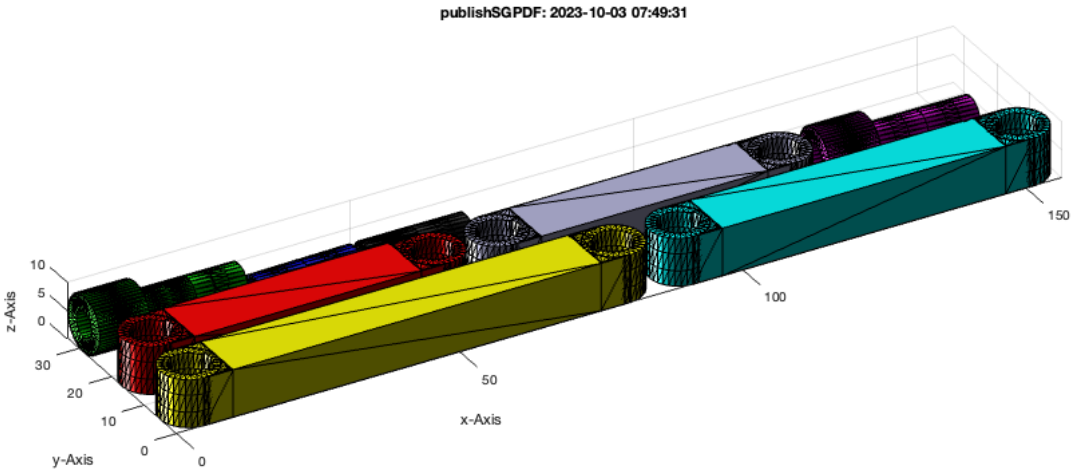
```
4% 8% 12% 16% 20% 24% 28% 32% 36% 40% 44% 48% 52% 56% 60% 64% 68% 72% 76% 80% 84% 88% 92% 96% 100%
```



4. Now Arrange all Parts for Printing as Separated Solids

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

```
binpacking3D: Packing 8 objects (h=66):
```



5. Now Write the Separated Parts into Different STL Files

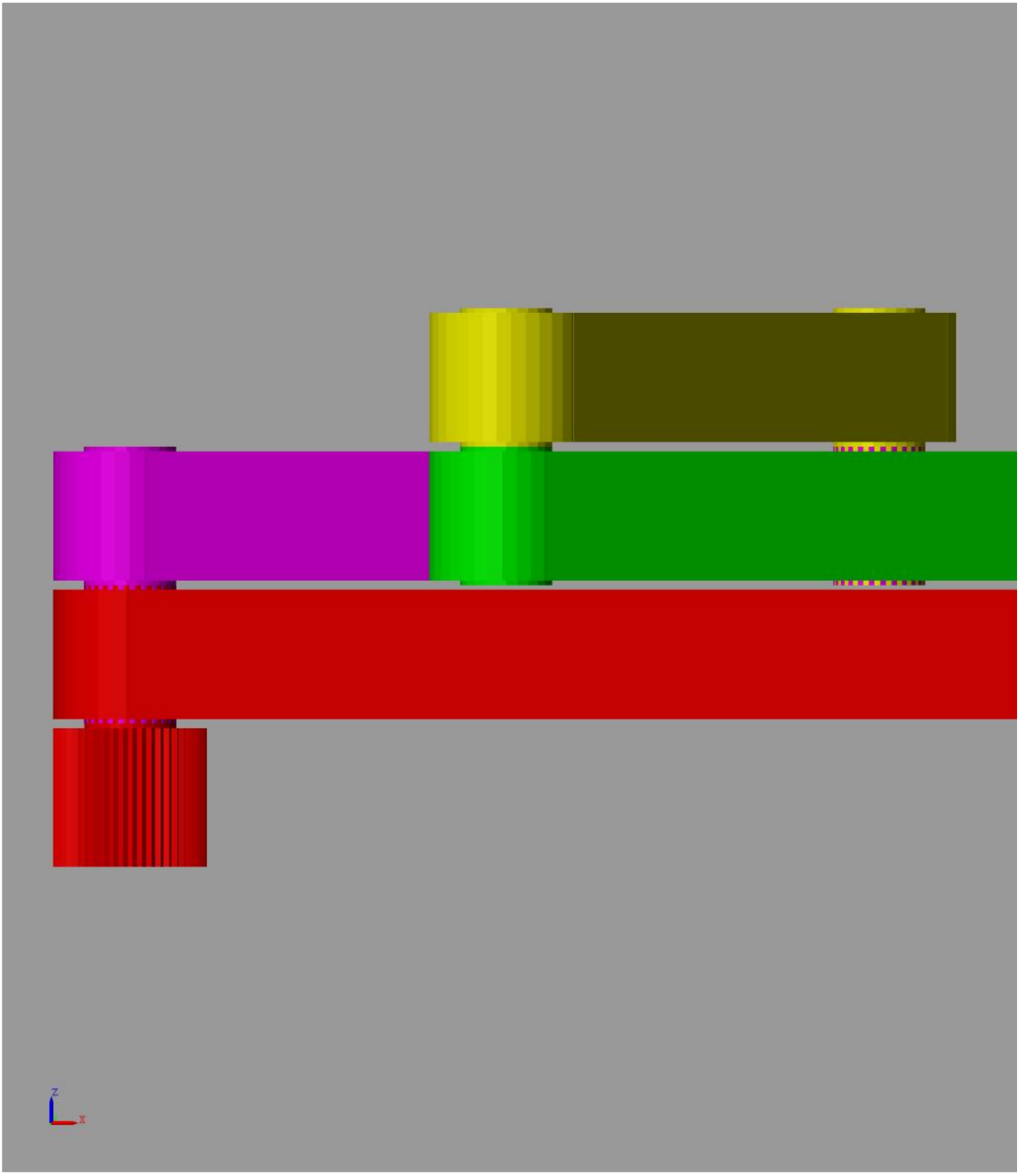
```
SGwriteSeparatedSTL(SG);
```

```
SGwriteSeparatedSTL: Writing 1 STL files in /Users/timlueth/Desktop/STLsep EXP-2023-10-03/
publishSGPDF:<a href = "matlab: openbydoubleclick ('/Users/timlueth/Desktop/STLsep EXP-2023-10-03')">/Users/timlueth/Desktop/STLsep EXP-2023-10-03/</a>
```

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/timlueth/Desktop/tmp_SG_LIB_EXP_26/Universal 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.  (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 07:49:53!
```

```
Executed 03-Oct-2023 07:49:55 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