

Tutorial 55: Automated Design of Manipulators with Screws or Ball Bearing

2020-08-07: Tim C. Lueth, Professor at Technische Universität München, Germany (URL: <http://www.SG-Lib.org>) - Last Change: 2020-08-07

Contents

- [Complete List of all Tutorials with Publishable MATLAB Files of this Solid-Geometries Toolbox](#)
- [Motivation for this tutorial: \(Originally SolidGeometry 4.9 required\)](#)
- [Automated Design of a rotating joint with a ball bearing and a DIN912-985 fixation with a nut-tunnel](#)
- [Final Remarks](#)

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

The following topics are covered and 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 Lightweight-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: Creation of Kinematic Chains and Robot Structures
- Tutorial 36: Creating a Patient-Individual Arm-Skin Protector-Shell
- Tutorial 37: Dimensioning of STL Files and Surface Data
- Tutorial 38: Some more solid geometry modelling function
- Tutorial 39: HEBO Modules robot design
- Tutorial 40: JACO Robot Simulation and Control
- Tutorial 41: Inserting Blades, Cuts and Joints into Solid Geometries
- Tutorial 42: Performing FEM Stress and Displacement Analysis and Structural Optimization of Solids
- Tutorial 43: Performing FEM Structural Optimization (CAO) and Topological Optimization (SKO) of Solids
- Tutorial 44: Creation of solids and kinematics from 3D curves and transformation matrices
- Tutorial 45: Creation of Solids using the SG-Coder - SGofCPLcommand
- Tutorial 46: Creating Fischertechnik compatible gear boxes using SGofCPLcommand
- Tutorial 47: Creating four-joints by 3 pose synthesis
- Tutorial 52: CPL Buffers and cw/ccw Orientation
- Tutorial 53: SKOL - Soft Kill Option for Large Displacement by Yilun Sun
- Tutorial 54: Automated Design of Precision Joints by Screws or Ball Bearings
- Tutorial 55: Automated Design of Manipulators with Screws or Ball Bearing

Motivation for this tutorial: (Originally SolidGeometry 4.9 required)

The task of this tutorial is to show the automated construction of kinematic chains, which can be designed with standardized bolts, nuts and washers (DIN or ISO).

```
dbprintf('The task of this tutorial is to show the automated construction of kinematic chains, which can be designed with standardized bolts, nuts and
```

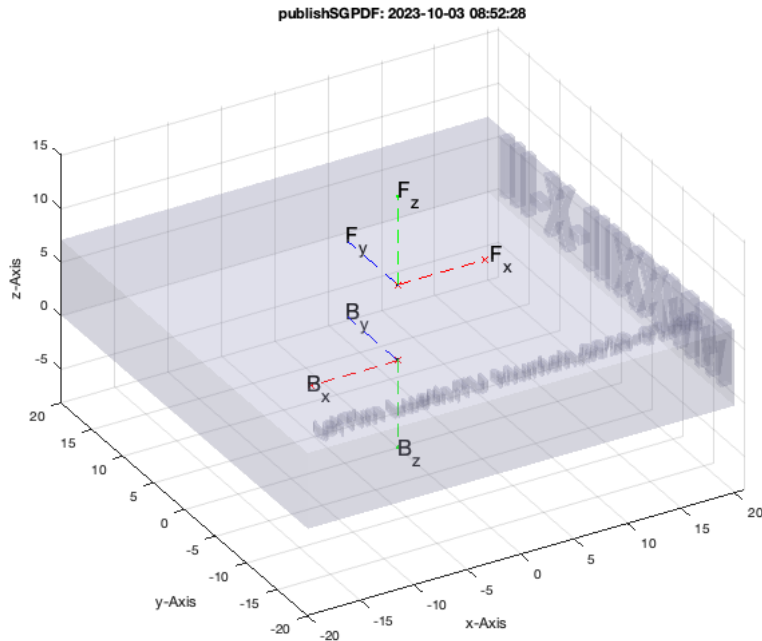
```
VLFL_EXP55: The task of this tutorial is to show the automated construction of kinematic chains, which can be designed with standardized bolts, nuts and
```

Automated Design of a rotating joint with a ball bearing and a DIN912-985 fixation with a nut-tunnel

```
LR=5.1; LH=7.1

D=SGManipulatorLink(40,true,LR,LH);
[A,B,C,U,E]=SGManipulatorBase(40,true,LR,LH);
SG=A; SGN=SG;
%
close all; SGfigure; view(-30,30); SGTplotalpha(SG,'w',0.2);
```

```
LH =
    7.1000
FLrepair: (SGremsurfedgepoints) No open boundaries finally exist!
FLrepair: (SGsubtract) No open boundaries finally exist!
SGboolh: "SGboolh('-', 'A', 'B')" called during "SGsubtract" failed: Solid A is unchanged
Warning: "SGsubtract(K,BX)" FAILED in "<a href = "matlab:
SGopentoline('SGManipulatorBase.m',124);">SGManipulatorBase</a>"
FLrepair: (SGremsurfedgepoints) No open boundaries finally exist!
FLrepair: (SGsubtract) No open boundaries finally exist!
```



```
[H,N,S]=SGdesignDIN912DIN985radialbearing([2.5 7 3.5],[8 -1],'T',2*7);
[~,NP]=SGdesignDIN912DIN985radialbearing([2.5 7 3.5],[8 -1],'TP',2*7);
if ~isequal(E,SG);
    SGN=SGsubtract(SG,H,'alignT',{'C','F'}); % Subtract the head side
end
if ~isequal(A,SG);
    if isequal(E,SG);
        SGN=SGsubtract(SGN,NP,'alignT',{'C','B'},'rotz',pi/1000,'roty',pi/1000); % Subtract the thread nut side
    else
        SGN=SGsubtract(SGN,N,'alignT',{'C','B'},'rotz',pi/1000,'roty',pi/1000); % Subtract the thread nut side
    end
end
```

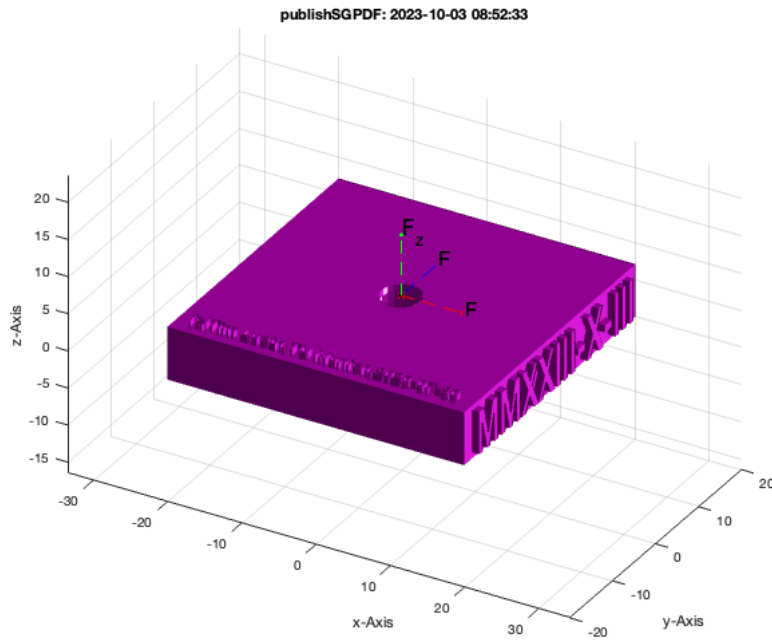
```
SGdesignDIN912DIN985radialbearing: Creating Nut M2.5
SGdesignDIN912DIN985radialbearing: Creating Screw M2.50 x 8mm
SGdesignDIN912DIN985radialbearing: Number of DIN433 washers for a minimal wall size 1.0mm is n=2
SGdesignDIN912DIN985radialbearing: The exact wall size on bearing size is: 1.0mm. The thread length on bearing side is 4.5mm
SGdesignDIN912DIN985radialbearing: Creating Head Side Bore 2.6mm x 1.2mm
SGdesignDIN912DIN985radialbearing: Creating Bearing Insertion Tunnel 2.50mm x 7.00mm
SGdesignDIN912DIN985radialbearing: Creating Head Insertion Pocket 4.6mm x 10.6mm x 14mm
```

```

SGdesignDIN912DIN985radialbearing: Creating Nut Side Bore 2.5mm x 3.7mm
SGdesignDIN912DIN985radialbearing: Creating Nut Insertion Tunnel 5.0mm x 14.0mm
SGdesignDIN912DIN985radialbearing: Creating Head Insertion Pocket 5.0mm x 3.6mm x 14mm
SGdesignDIN912DIN985radialbearing: =====
SGdesignDIN912DIN985radialbearing: Creating Nut M2.5
SGdesignDIN912DIN985radialbearing: Creating Screw M2.50 x 8mm
SGdesignDIN912DIN985radialbearing: Number of DIN433 washers for a minimal wall size 1.0mm is n=2
SGdesignDIN912DIN985radialbearing: The exact wall size on bearing size is: 1.0mm. The thread length on bearing side is 4.5mm
SGdesignDIN912DIN985radialbearing: Creating Head Side Bore 2.6mm x 1.2mm
SGdesignDIN912DIN985radialbearing: Creating Bearing Insertion Tunnel 2.50mm x 7.00mm
SGdesignDIN912DIN985radialbearing: Creating Head Insertion Pocket 4.6mm x 10.6mm x 14mm
SGdesignDIN912DIN985radialbearing: Creating Nut Side Bore 2.5mm x 3.7mm
SGdesignDIN912DIN985radialbearing: Creating Nut Insertion Tunnel 5.0mm x 14.0mm
SGdesignDIN912DIN985radialbearing: Creating Head Insertion Pocket 5.0mm x 3.6mm x 14mm
SGdesignDIN912DIN985radialbearing: =====

```

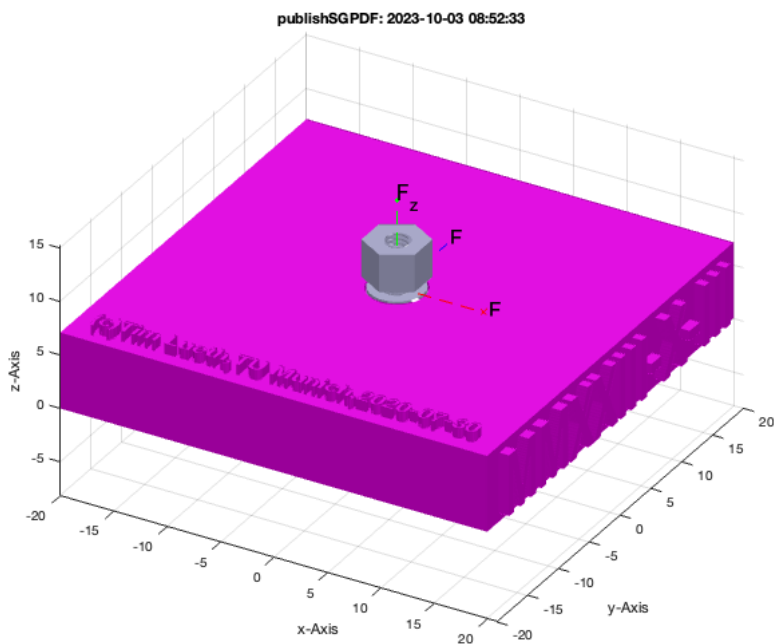
```
SGfigure; SGTplotalpha(SGN, 'm'); view(+30,30);
```



```

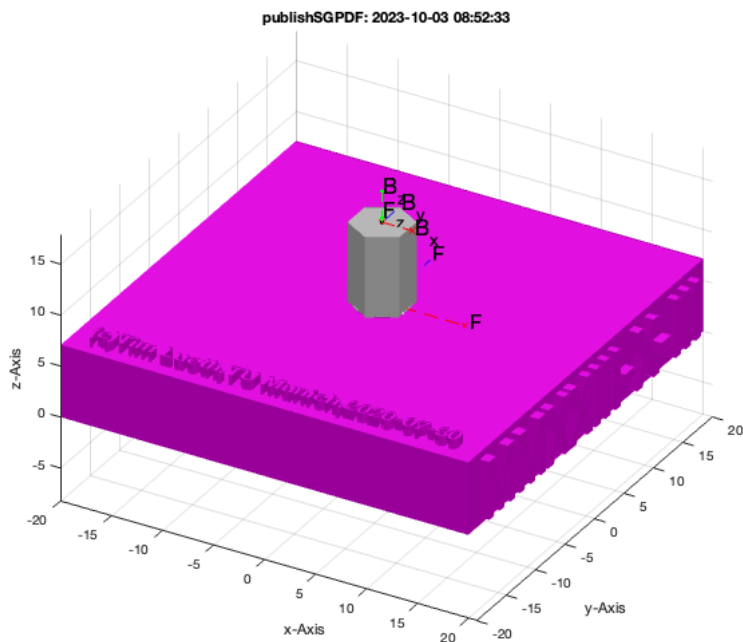
SGTplotalpha(S, 'w', ' ', ' ', SGN, 'alignT', {'C', 'F'});
if ~isequal(A,SG); SGTplotalpha(S, 'w', ' ', ' ', SGN, 'matchT', {'C', 'B'}); end;

```



```
SGTplotalpha(H, 'r', 'r', 'r', SGN, 'alignT', {'C', 'F'});
SGTplotalpha(N, 'r', 'r', 'r', SGN, 'alignT', {'C', 'B'});
SGwriteSTL(SGN)
```

```
publishSGPDF:<a href = "matlab: openbydoubleclick ('/Users/timlueth/Desktop')"/>Users/timlueth/Desktop/</a><a href = "matlab: openbydoubleclick ('/Userans = '/Users/timlueth/Desktop/publishSGPDF_SGN_(2023-10-03).STL'
```



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 08:52:35!  
Executed 03-Oct-2023 08:52:37 by 'timlueth' on a MACI64 using Mac OSX 13.6 | R2023a Update 5 | SG-Lib 5.4  
===== Used Matlab products: =====  
database_toolbox  
distrib_computing_toolbox  
fixed_point_toolbox  
image_toolbox  
map_toolbox  
matlab  
optimization_toolbox  
pde_toolbox  
simmechanics  
simscape  
simulink  
=====
```

Published with MATLAB® R2023a