

## Tutorial 65: Solid Geometry Cut and Cross Section Path extrusion

2021-03-15: Tim C. Lueth, Professor at Technische Universität München, Germany (URL: <http://www.SG-Lib.org>) - Last Change: 2021-03-16

### Contents

---

- [Complete List of all Tutorials with Publishable MATLAB Files of this Solid-Geoemtries Toolbox](#)
- [Motivation for this tutorial: \(Originally SolidGeometry 5.1 required\)](#)
- [Concept of Cutting Solid by a Frame](#)
- [Twist Just elongate](#)
- [TWIST Elongate and turn 90 degree](#)
- [TWIST Elongate and turn 180 degree](#)
- [TWIST Elongate and turn 270 degree](#)
- [TWIST Elongate and turn -90 degree](#)
- [Twist Just elongate](#)
- [STRAIGHT Just elongate](#)
- [STRAIGHT Elongate and turn 90 degree](#)
- [STRAIGHT Elongate and turn 180 degree](#)
- [STRAIGHT Elongate and turn 270 degree](#)
- [STRAIGHT Elongate and turn -90 degree](#)
- [STRAIGHT IN Z AND ROTATE](#)
- [STRAIGHT IN Y AND ROTATE](#)
- [STRAIGHT IN +X AND ROTATE](#)
- [STRAIGHT IN -Z AND ROTATE](#)
- [STRAIGHT IN -Y AND ROTATE](#)
- [STRAIGHT in -X AND ROTATE](#)
- [THE COBRA POSE](#)

### Complete List of all Tutorials with Publishable MATLAB Files of this Solid-Geoemtries 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: Create a Solid by two arbitrary CPLs and a distance
- Tutorial 48: Gear Pairings by Yannick Krieger
- Tutorial 49: Generation of non circular gear pairs by Yannick Krieger/Sebastian Baumgartner
- Tutorial 50: CVLof2CPLzcorrelate and SGof2CPLzcorrelate
- Tutorial 51: Creating Parallel Tasks for batch processing
- 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
- Tutorial 56: Checking Functions for Solids
- Tutorial 57: Processing Stacks of Slices = CVLz
- Tutorial 58: Integrating joints into solids
- Tutorial 59: Integrating arbitrary joints into solids
- Tutorial 60: Facet generation for arbitrary contours in 3D space
- Tutorial 61: FeeTech Servo Toolbox
- Tutorial 62: Design of Monolithic Snake-like Manipulators
- Tutorial 63: Generation of STL archives through the analysis of assemblies
- Tutorial 64: Relative spatial arrangement of CPL contours
- Tutorial 65: Solid Geometry Cut and Cross Section Path extrusion

#### Motivation for this tutorial: (Originally SolidGeometry 5.1 required)

```
function VLFL_EXP65
```

```
load Yannick_robot.mat      % or loadweb JACO_robot.mat
```

```
T=[
-0.0019    0.0000    1.0000   -19.1082
  0.0000    1.0000   -0.0000    8.1038
 -1.0000    0.0000   -0.0019   37.6292
         0         0         0    1.0000];
```

```
SG=YKLower
```

```
SG =
struct with fields:
```

```

VL: [18280x3 double]
FL: [36600x3 double]
NL: [109800x3 double]
stampname: '/Volumes/LUETH-WIN/LCL Robot/2020-11-02 Simon Schiele LCL_Robot V2 STLs/2020-10-05_LCLtest_Unterarm_1.stl'
stampdate: '30-Nov-2020 16:51:47'
    
```

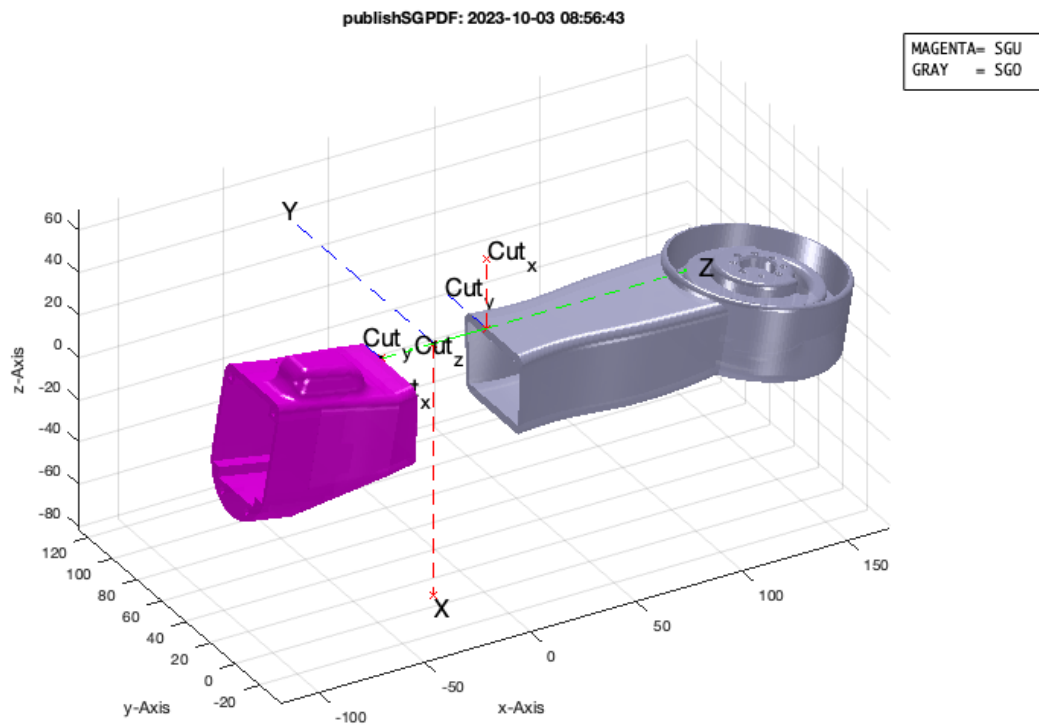
### Concept of Cutting Solid by a Frame

```
SGcutT(SG,T,'',50);
```

```
TYPE='twist'
```

```

TYPE =
' twist'
    
```



### Twist Just elongate

```
SGcutTexttrudeT(SG,T,[0 0 100],'',TYPE); SGN=ans;
```

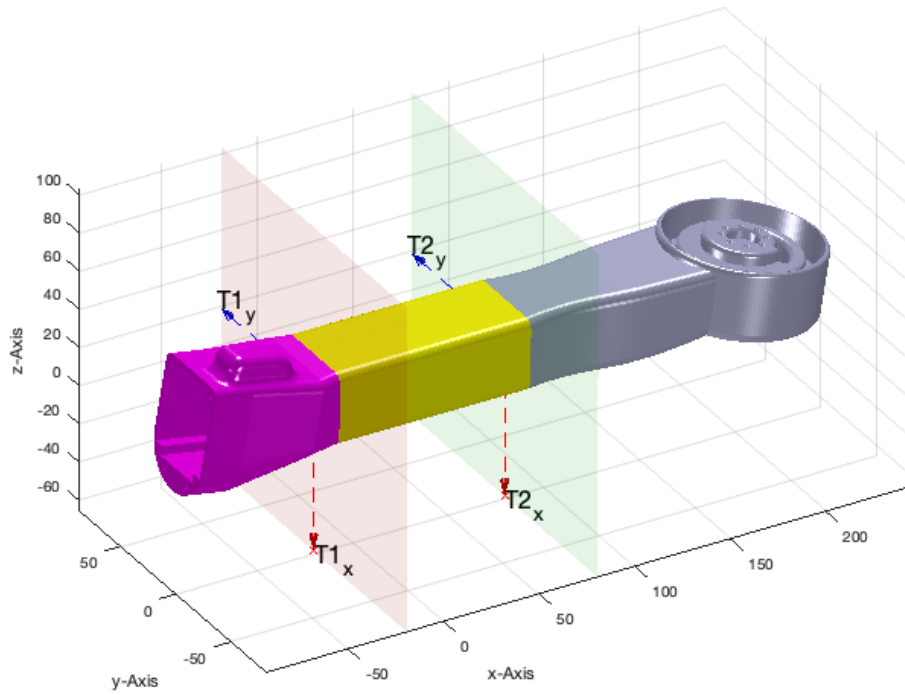
SGcutTexttrudeT: The extension frame is Td:

```

Td =
     1     0     0     0
     0     1     0     0
     0     0     1    100
     0     0     0     1
    
```

```
SGcutTexttrudeT: twist
```

publishSGPDF: 2023-10-03 08:56:45



### TWIST Elongate and turn 90 degree

```
SGcutTextrudeT(SG,T,TofPez([0 0 100],[0 0 1],pi/2),'',TYPE); SGN=ans;
```

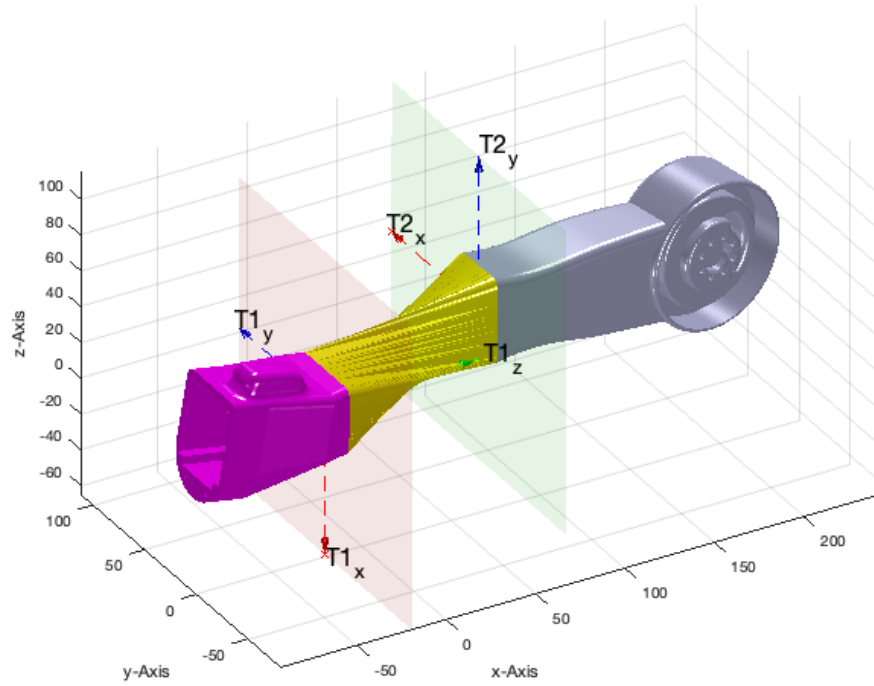
SGcutTextrudeT: The extension frame is Td:

Td =

0.0000	-1.0000	0	0
1.0000	0.0000	0	0
0	0	1.0000	100.0000
0	0	0	1.0000

SGcutTextrudeT: twist

publishSGPDF: 2023-10-03 08:56:47



### TWIST Elongate and turn 180 degree

```
SGcutTextrudeT(SG,T,TofPez([0 0 100],[0 0 1],pi),'',TYPE); SGN=ans;
```

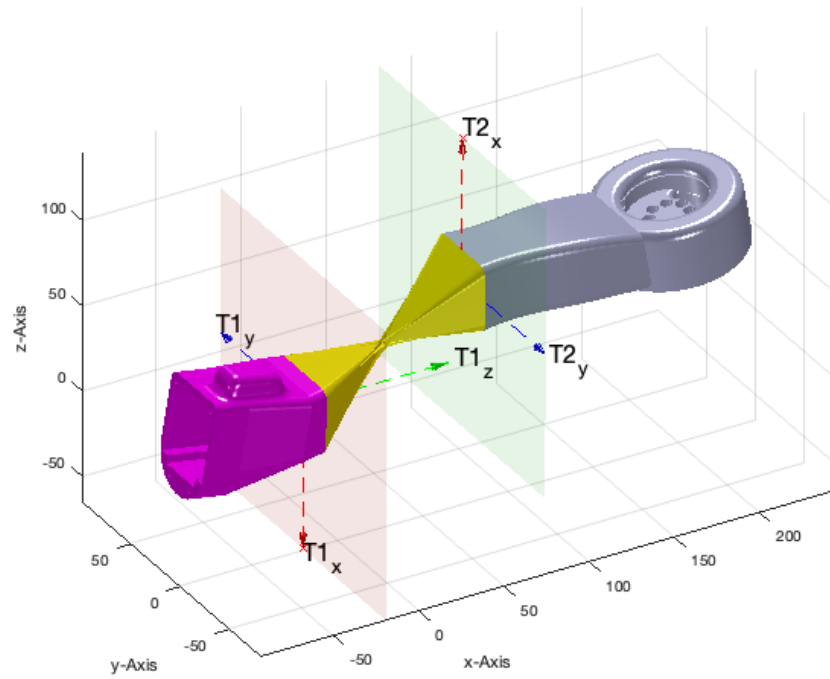
SGcutTextrudeT: The extention frame is Td:

Td =

-1.0000	-0.0000	0	0
0.0000	-1.0000	0	0
0	0	1.0000	100.0000
0	0	0	1.0000

SGcutTextrudeT: twist

publishSGPDF: 2023-10-03 08:56:50



### TWIST Elongate and turn 270 degree

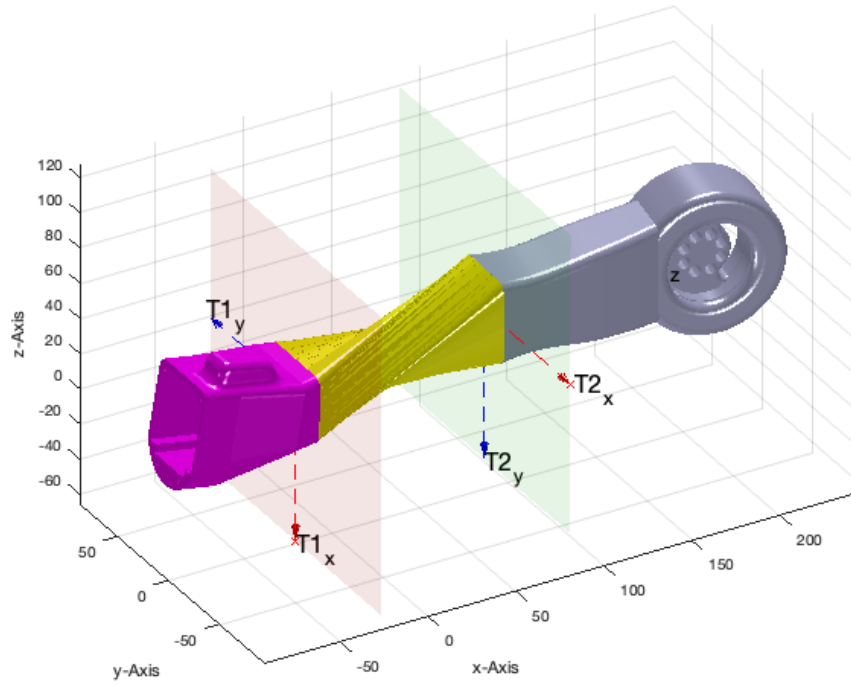
```
SGcutTextrudeT(SG,T,TofPez([0 0 100],[0 0 1],pi*1.5),'',TYPE); SGN=ans;
```

SGcutTextrudeT: The extention frame is Td:

```
Td =
-0.0000    1.0000         0         0
-1.0000   -0.0000         0         0
         0         0    1.0000   100.0000
         0         0         0    1.0000
```

SGcutTextrudeT: twist

publishSGPDF: 2023-10-03 08:56:52

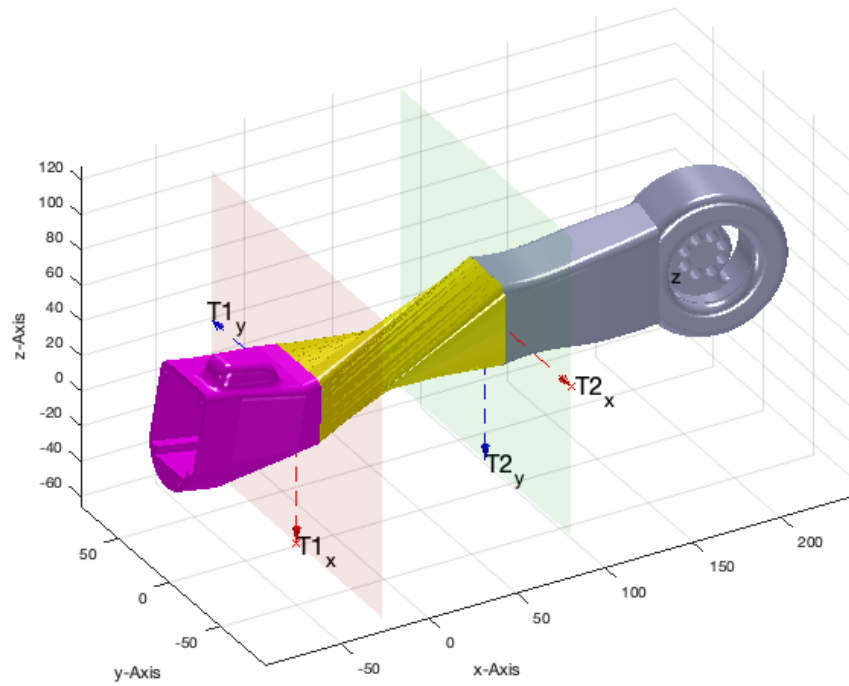


**TWIST Elongate and turn -90 degree**

```
SGcutTextrudeT(SG,T,TofPez([0 0 100],[0 0 1],[-pi/2],'',TYPE); SGN=ans;
```

SGcutTextrudeT: The extention frame is Td:  
Td =  
0.0000 1.0000 0 0  
-1.0000 0.0000 0 0  
0 0 1.0000 100.0000  
0 0 0 1.0000  
SGcutTextrudeT: twist

publishSGPDF: 2023-10-03 08:56:54



### Twist Just elongate

```
TYPE='twist'; SGcutTextrudeT(SG,T,[-100 100 100],'',TYPE); SGN=ans;
```

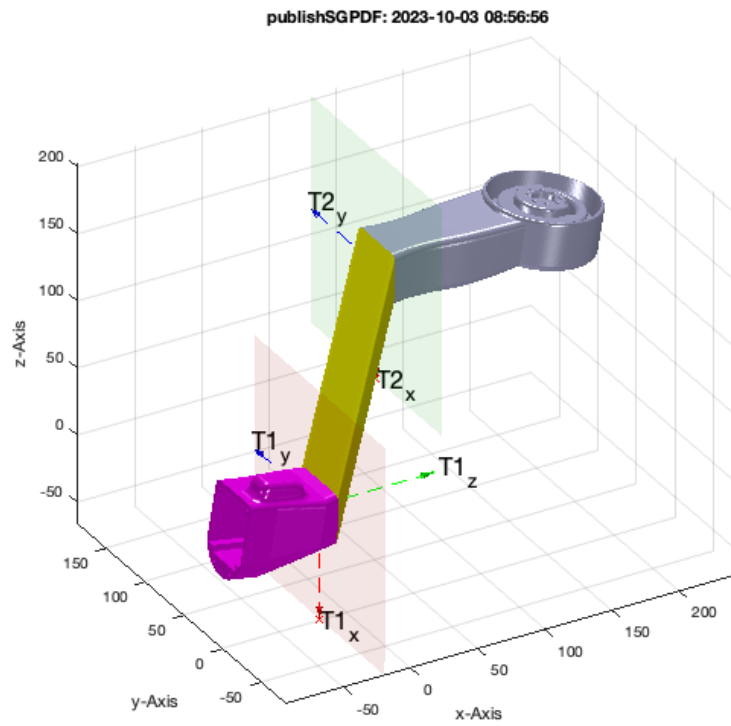
SGcutTextrudeT: The extension frame is Td:

Td =

1	0	0	-100
0	1	0	100
0	0	1	100
0	0	0	1

SGcutTextrudeT: twist





### STRAIGHT Just elongate

```
TYPE='straight'; SGcutTextrudeT(SG,T,[0 0 100],'',TYPE); SGN=ans;
```

SGcutTextrudeT: The extention frame is Td:

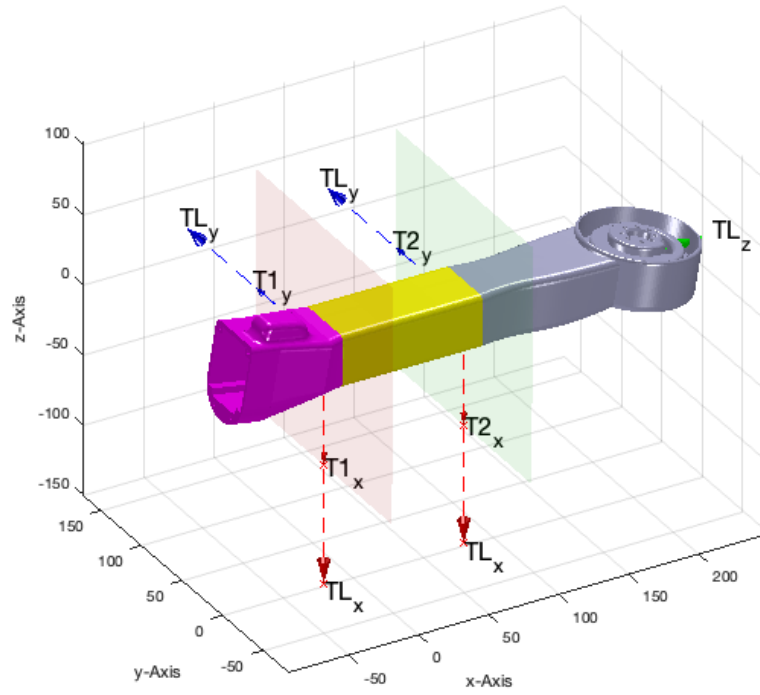
Td =

1	0	0	0
0	1	0	0
0	0	1	100
0	0	0	1

SGcutTextrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY 66 degree

publishSGPDF: 2023-10-03 08:56:59



### STRAIGHT Elongate and turn 90 degree

```
TYPE='straight'; SGcutTextrudeT(SG,T,TofPez([0 0 100],[0 0 1],pi/2),'',TYPE); SGN=ans;
```

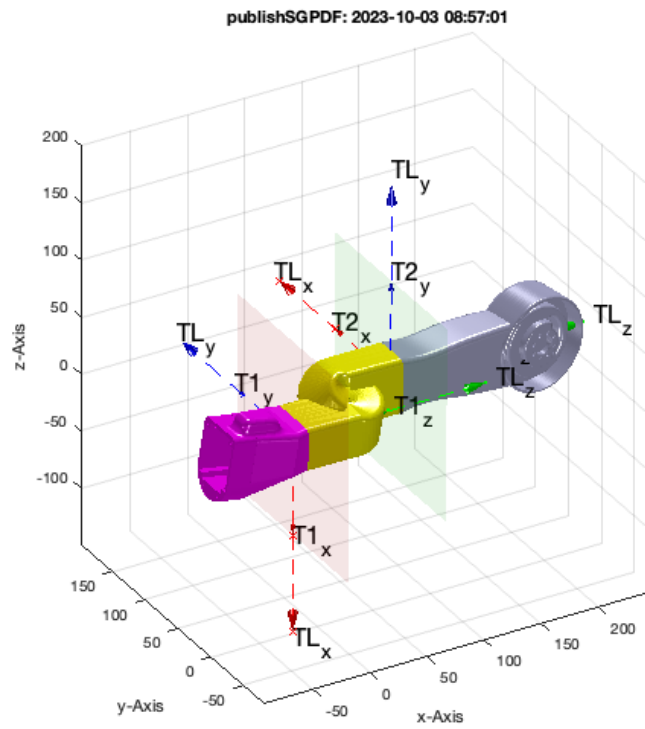
SGcutTextrudeT: The extention frame is Td:

```
Td =
    0.0000    -1.0000         0         0
    1.0000     0.0000         0         0
         0         0     1.0000    100.0000
         0         0         0     1.0000
```

SGcutTextrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY -118 degree

TLadjustR: NEED TO ROTATE ENDFRAME BY 90 degree



### STRAIGHT Elongate and turn 180 degree

```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([0 0 100],[0 0 1],pi),'',TYPE); SGN=ans;
```

SGcutTexttrudeT: The extention frame is Td:

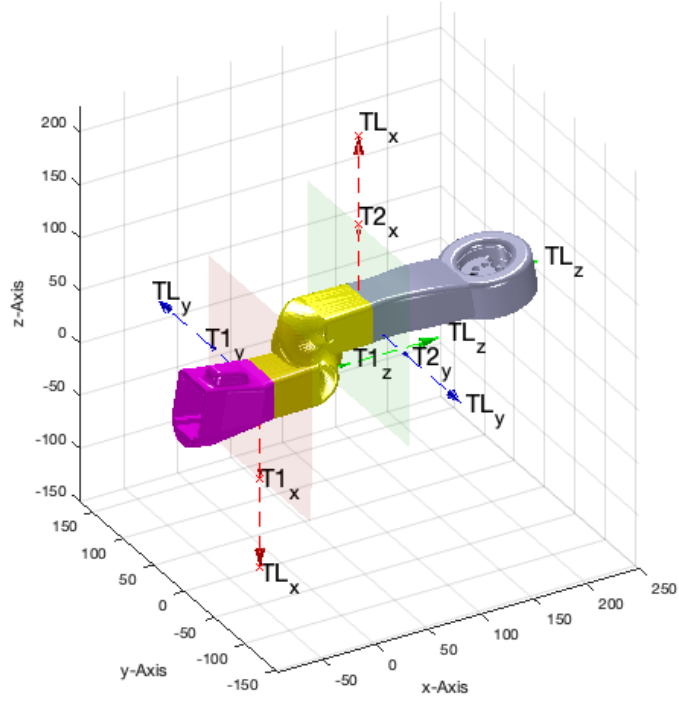
```
Td =
-1.0000  -0.0000   0   0
 0.0000  -1.0000   0   0
 0   0   1.0000  100.0000
 0   0   0   1.0000
```

SGcutTexttrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY -162 degree

TLadjustR: NEED TO ROTATE ENDFRAME BY -180 degree

publishSGPDF: 2023-10-03 08:57:04



**STRAIGHT Elongate and turn 270 degree**

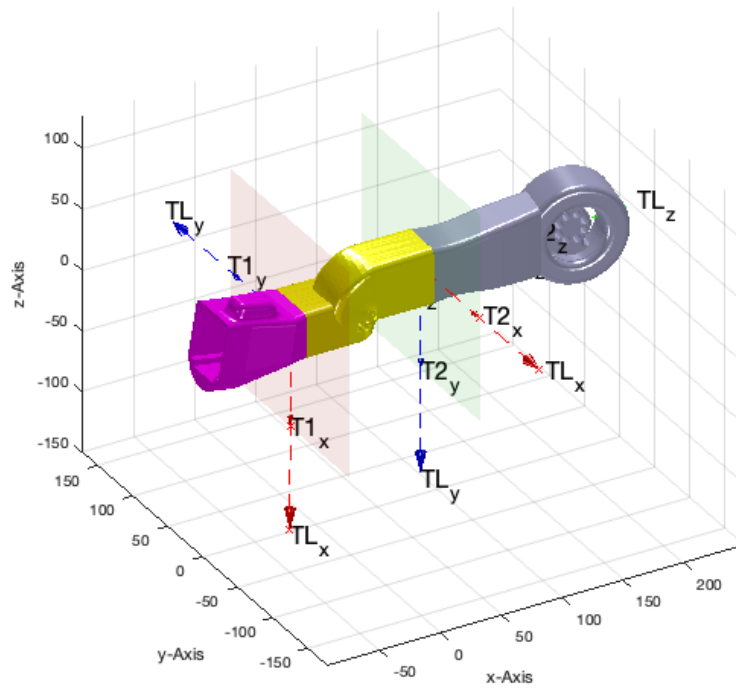
```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([0 0 100],[0 0 1],pi*1.5),'',TYPE); SGN=ans;
```

SGcutTexttrudeT: The extention frame is Td:

```
Td =
-0.0000    1.0000         0         0
-1.0000   -0.0000         0         0
         0         0    1.0000   100.0000
         0         0         0    1.0000
```

```
SGcutTexttrudeT: straight
TLadjustR: NEED TO ROTATE STARTFRAME BY 153 degree
TLadjustR: NEED TO ROTATE ENDFRAME BY -90 degree
```

publishSGPDF: 2023-10-03 08:57:07



### STRAIGHT Elongate and turn -90 degree

```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([0 0 100],[0 0 1],[-pi/2),'',TYPE); SGN=ans;
```

SGcutTexttrudeT: The extention frame is Td:

Td =

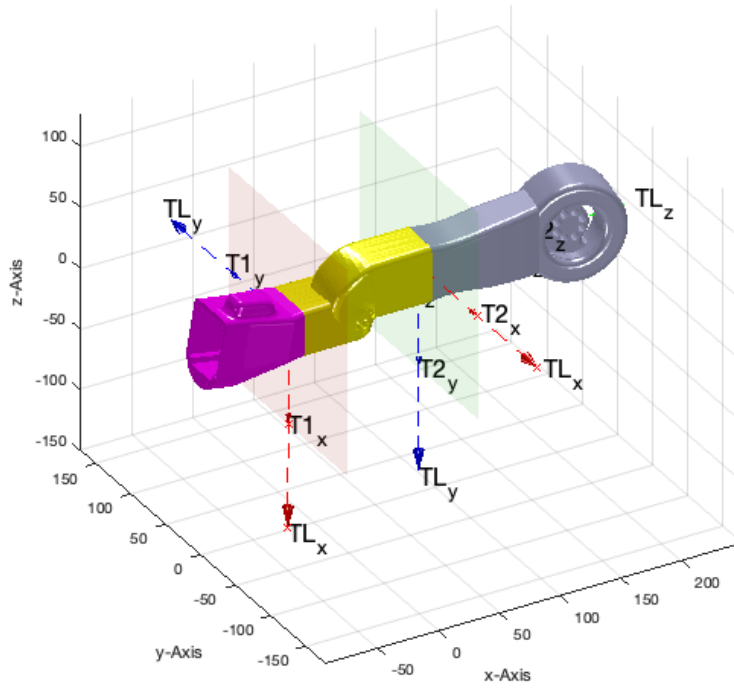
0.0000	1.0000	0	0
-1.0000	0.0000	0	0
0	0	1.0000	100.0000
0	0	0	1.0000

SGcutTexttrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY 153 degree

TLadjustR: NEED TO ROTATE ENDFRAME BY -90 degree

publishSGPDF: 2023-10-03 08:57:09



### STRAIGHT IN Z AND ROTATE

```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([-50 50 50],[0 0 1],0),'',TYPE); SGN=ans;
```

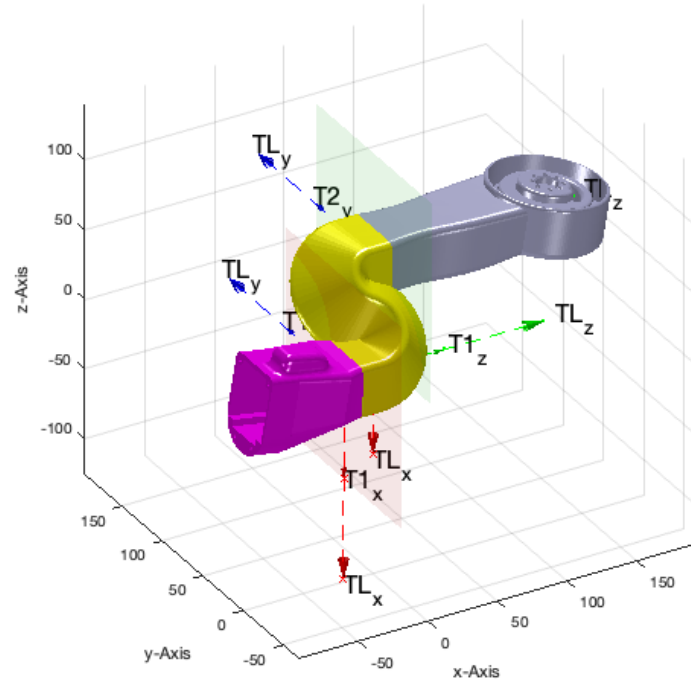
SGcutTexttrudeT: The extention frame is Td:

```
Td =
    1     0     0   -50
    0     1     0    50
    0     0     1    50
    0     0     0     1
```

SGcutTexttrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY -135 degree

publishSGPDF: 2023-10-03 08:57:12



```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([-50 50 50],[0 0 1],+pi/2),'',TYPE); SGN=ans;
```

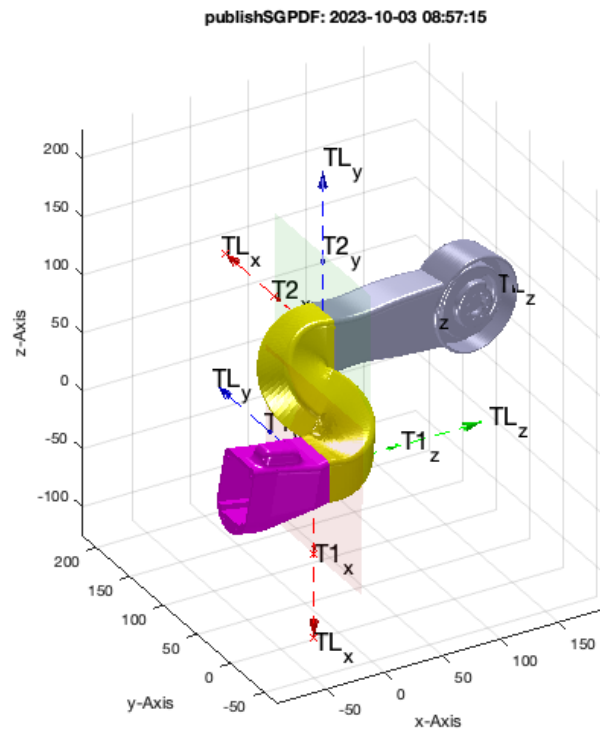
SGcutTexttrudeT: The extention frame is Td:

Td =  
 0.0000 -1.0000 0 -50.0000  
 1.0000 0.0000 0 50.0000  
 0 0 1.0000 50.0000  
 0 0 0 1.0000

SGcutTexttrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY -130 degree

TLadjustR: NEED TO ROTATE ENDFRAME BY 90 degree



```
TYPE='straight'; SGcutTextrudeT(SG,T,TofPez([-50 50 50],[0 0 1],[-pi/2),'',TYPE); SGN=ans;
```

SGcutTextrudeT: The extension frame is Td:

Td =

0.0000	1.0000	0	-50.0000
-1.0000	0.0000	0	50.0000
0	0	1.0000	50.0000
0	0	0	1.0000

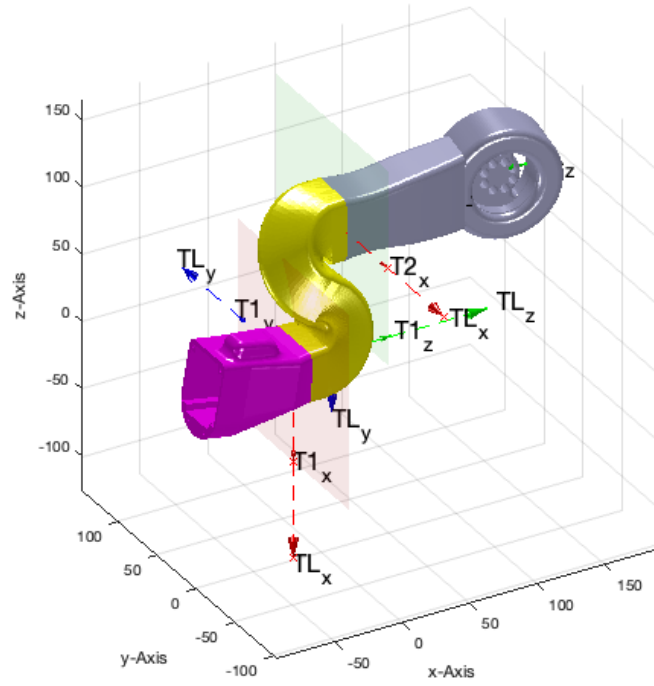
SGcutTextrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY -153 degree

TLadjustR: NEED TO ROTATE ENDFRAME BY -90 degree



publishSGPDF: 2023-10-03 08:57:17



```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([-50 50 50],[0 0 1],+pi),'',TYPE); SGN=ans;
```

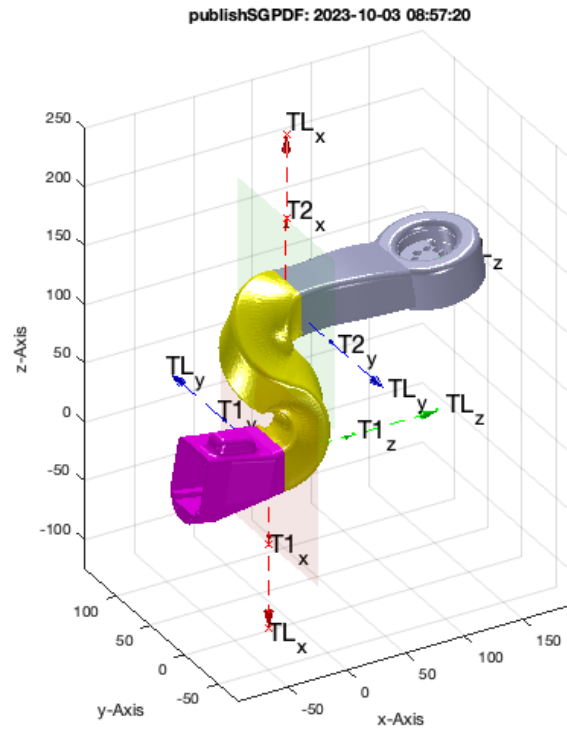
SGcutTexttrudeT: The extention frame is Td:

```
Td =
-1.0000  -0.0000   0 -50.0000
 0.0000  -1.0000   0  50.0000
 0         0   1.0000  50.0000
 0         0   0   1.0000
```

SGcutTexttrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY -145 degree

TLadjustR: NEED TO ROTATE ENDFRAME BY -180 degree



**STRAIGHT IN Y AND ROTATE**

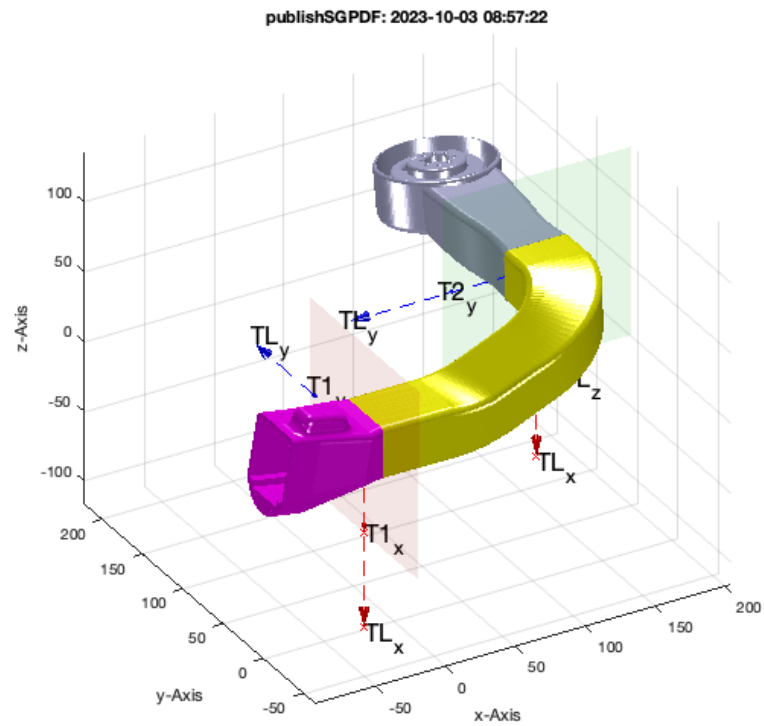
```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([-50 50 150],[0 1 0],0),'',TYPE); SGN=ans;
```

SGcutTexttrudeT: The extention frame is Td:

Td =

1	0	0	-50
0	0	1	50
0	-1	0	150
0	0	0	1

SGcutTexttrudeT: straight  
 TLadjustR: NEED TO ROTATE STARTFRAME BY -132 degree  
 TLadjustR: NEED TO ROTATE ENDFRAME BY 34 degree



```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([-50 50 150],[0 1 0],+pi/2),'',TYPE); SGN=ans;
```

SGcutTexttrudeT: The extention frame is Td:

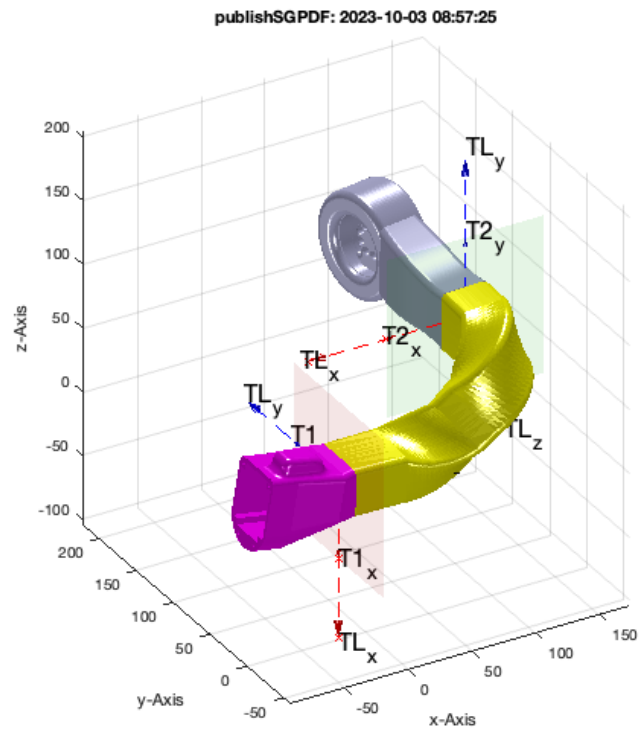
Td =

0.0000	-1.0000	0	-50.0000
0	0	1.0000	50.0000
-1.0000	-0.0000	0	150.0000
0	0	0	1.0000

SGcutTexttrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY -138 degree

TLadjustR: NEED TO ROTATE ENDFRAME BY 141 degree



```
TYPE='straight'; SGcutTextrudeT(SG,T,TofPez([-50 50 150],[0 1 0],[-pi/2],'',TYPE); SGN=ans;
```

SGcutTextrudeT: The extension frame is Td:

Td =

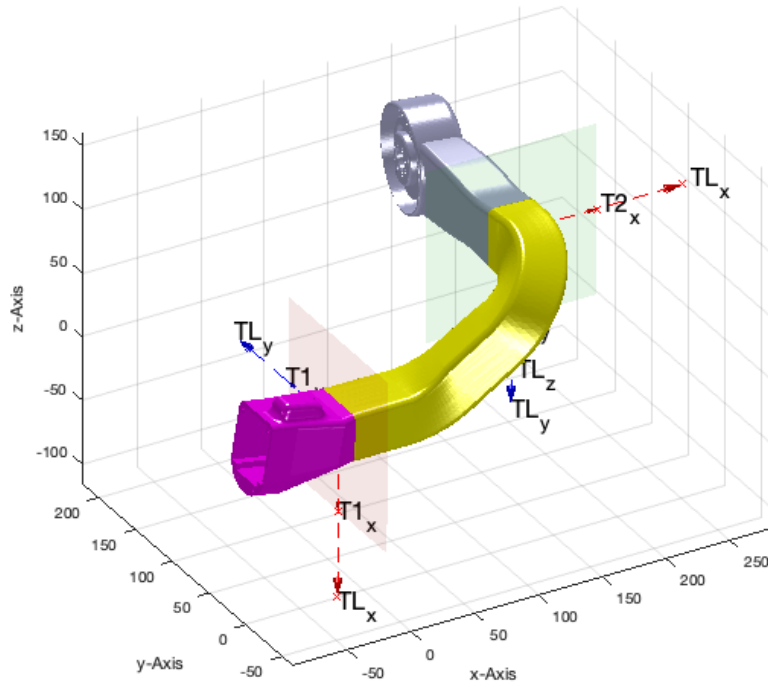
0.0000	1.0000	0	-50.0000
0	0	1.0000	50.0000
1.0000	-0.0000	0	150.0000
0	0	0	1.0000

SGcutTextrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY -143 degree

TLadjustR: NEED TO ROTATE ENDFRAME BY -50 degree

publishSGPDF: 2023-10-03 08:57:27



```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([-50 50 150],[0 1 0],+pi),'',TYPE); SGN=ans;
```

SGcutTexttrudeT: The extention frame is Td:

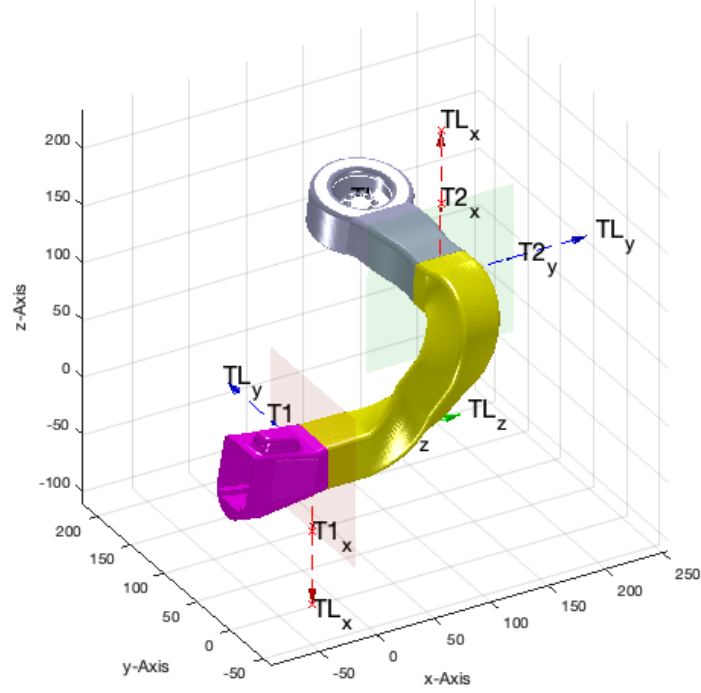
Td =  
 -1.0000   -0.0000   0   -50.0000  
           0        0    1.0000   50.0000  
 -0.0000   1.0000   0   150.0000  
           0        0        0    1.0000

SGcutTexttrudeT: straight

TladjustR: NEED TO ROTATE STARTFRAME BY -147 degree

TladjustR: NEED TO ROTATE ENDFRAME BY -126 degree

publishSGPDF: 2023-10-03 08:57:30



### STRAIGHT IN +X AND ROTATE

```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([-50 50 150],[1 0 0],0),'',TYPE); SGN=ans;
```

SGcutTexttrudeT: The extention frame is Td:

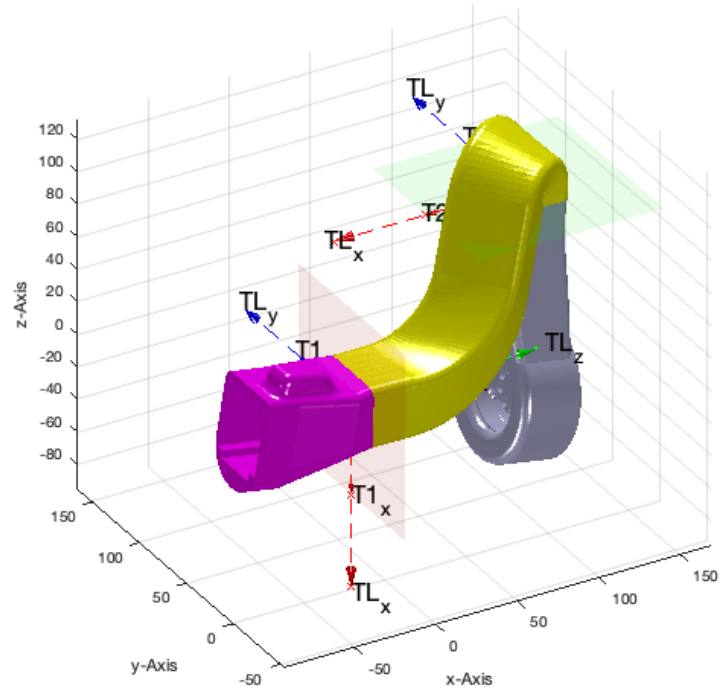
```
Td =
    0     0     1   -50
    0     1     0    50
   -1     0     0   150
    0     0     0     1
```

SGcutTexttrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY -144 degree

TLadjustR: NEED TO ROTATE ENDFRAME BY 59 degree

publishSGPDF: 2023-10-03 08:57:32



```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([-50 50 150],[1 0 0],+pi/2),'',TYPE); SGN=ans;
```

SGcutTexttrudeT: The extention frame is Td:

Td =

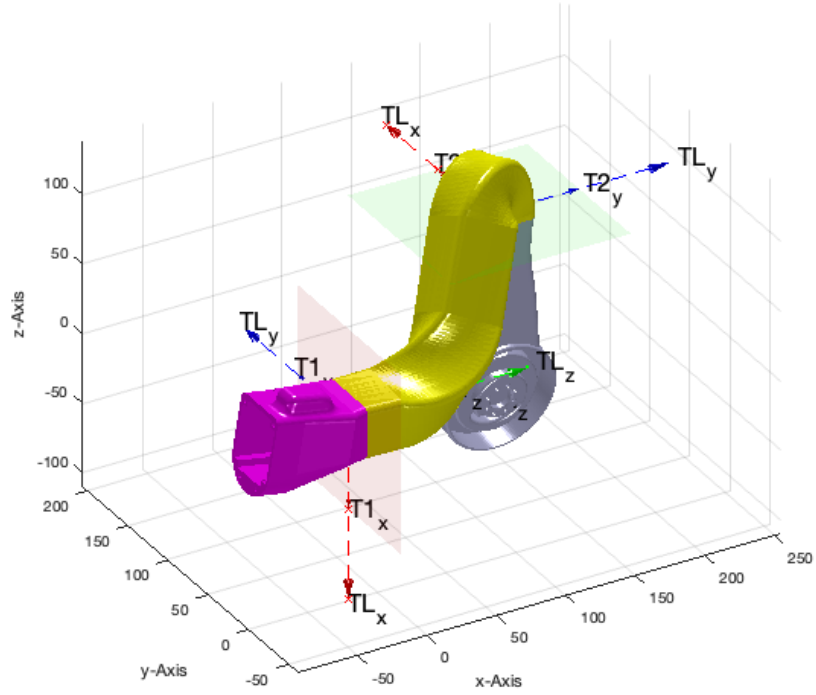
0	0	1.0000	-50.0000
1.0000	0.0000	0	50.0000
-0.0000	1.0000	0	150.0000
0	0	0	1.0000

SGcutTexttrudeT: straight

TladjustR: NEED TO ROTATE STARTFRAME BY -133 degree

TladjustR: NEED TO ROTATE ENDFRAME BY 156 degree

publishSGPDF: 2023-10-03 08:57:35



```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([-50 50 150],[1 0 0],[-pi/2),'',TYPE); SGN=ans;
```

SGcutTexttrudeT: The extention frame is Td:

```
Td =
    0         0         1.0000   -50.0000
   -1.0000    0.0000         0         50.0000
   -0.0000   -1.0000         0        150.0000
    0         0         0         1.0000
```

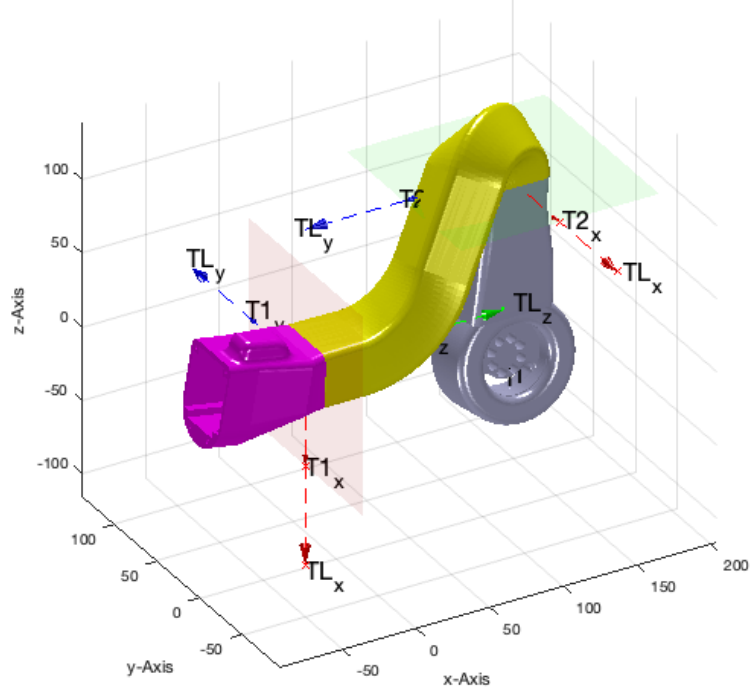
SGcutTexttrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY -151 degree

TLadjustR: NEED TO ROTATE ENDFRAME BY -57 degree



publishSGPDF: 2023-10-03 08:57:37



```
TYPE='straight'; SGcutTextrudeT(SG,T,TofPez([-50 50 150],[1 0 0],+pi),'',TYPE); SGN=ans;
```

SGcutTextrudeT: The extention frame is Td:

Td =

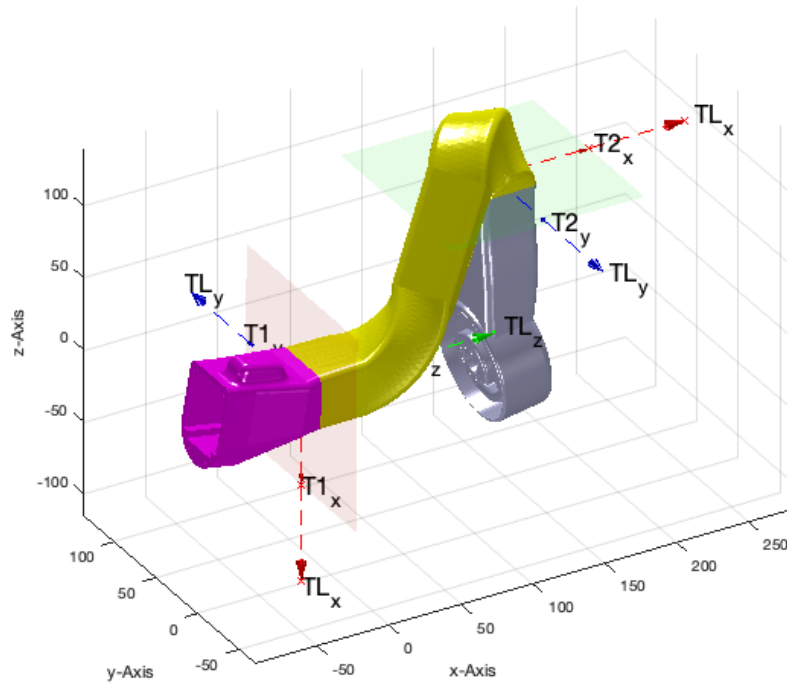
0	0	1.0000	-50.0000
0.0000	-1.0000	0	50.0000
1.0000	0.0000	0	150.0000
0	0	0	1.0000

SGcutTextrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY -138 degree

TLadjustR: NEED TO ROTATE ENDFRAME BY -136 degree

publishSGPDF: 2023-10-03 08:57:40



### STRAIGHT IN -Z AND ROTATE

```
TYPE='straight'; SGcutTextrudeT(SG,T,TofPez([-50 50 50],[0 0 -1],0),'',TYPE); SGN=ans;
```

SGcutTextrudeT: The extension frame is Td:

Td =

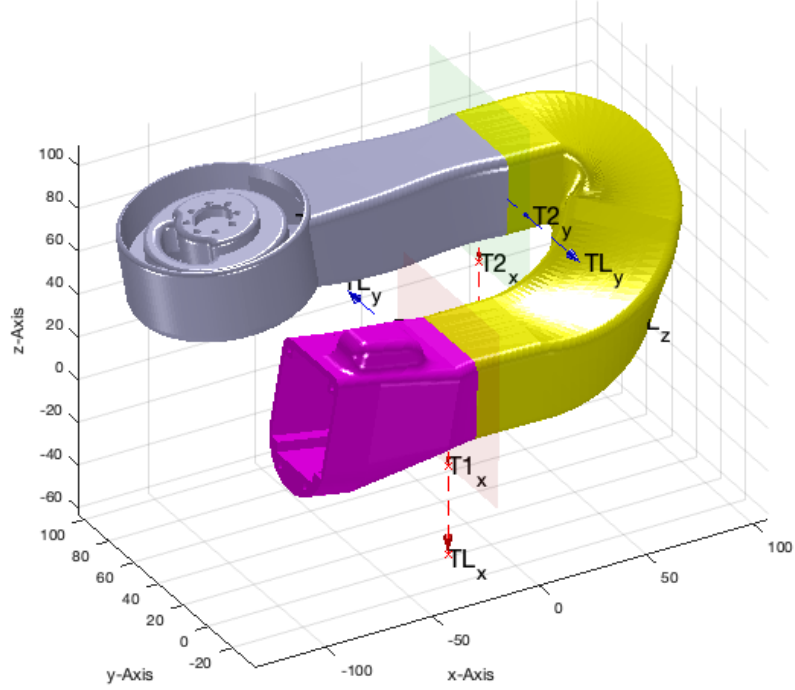
1	0	0	-50
0	-1	0	50
0	0	-1	50
0	0	0	1

SGcutTextrudeT: straight

TladjustR: NEED TO ROTATE STARTFRAME BY -129 degree

TladjustR: NEED TO ROTATE ENDFRAME BY 78 degree

publishSGPDF: 2023-10-03 08:57:42



```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([-50 50 50],[0 0 -1],+pi/2),'',TYPE); SGN=ans;
```

SGcutTexttrudeT: The extention frame is Td:

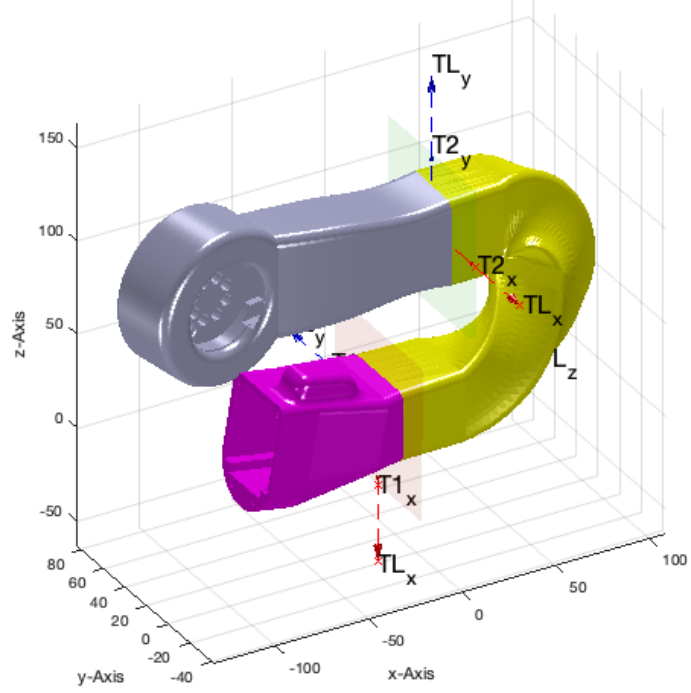
Td =  
 0.0000 -1.0000 0 -50.0000  
 -1.0000 -0.0000 0 50.0000  
 0 0 -1.0000 50.0000  
 0 0 0 1.0000

SGcutTexttrudeT: straight

TladjustR: NEED TO ROTATE STARTFRAME BY -149 degree

TladjustR: NEED TO ROTATE ENDFRAME BY -152 degree

publishSGPDF: 2023-10-03 08:57:45



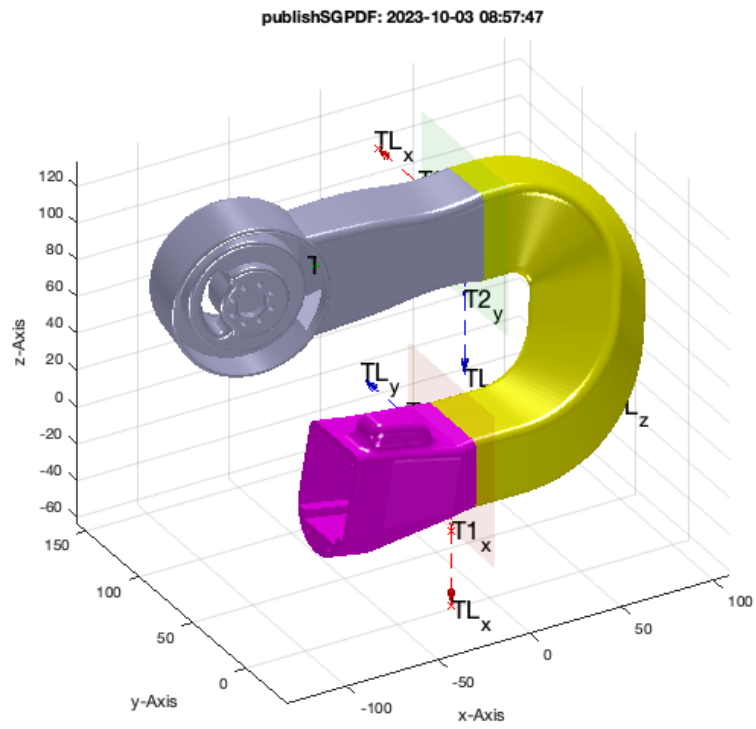
```
TYPE='straight'; SGcutTextrudeT(SG,T,TofPez([-50 50 50],[0 0 -1],[-pi/2),'',TYPE); SGN=ans;
```

SGcutTextrudeT: The extention frame is Td:

```
Td =
    0.0000    1.0000         0   -50.0000
    1.0000   -0.0000         0    50.0000
         0         0   -1.0000    50.0000
         0         0         0     1.0000
```

SGcutTextrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY -135 degree



```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([-50 50 50],[0 0 -1],+pi),'',TYPE); SGN=ans;
```

SGcutTexttrudeT: The extention frame is Td:

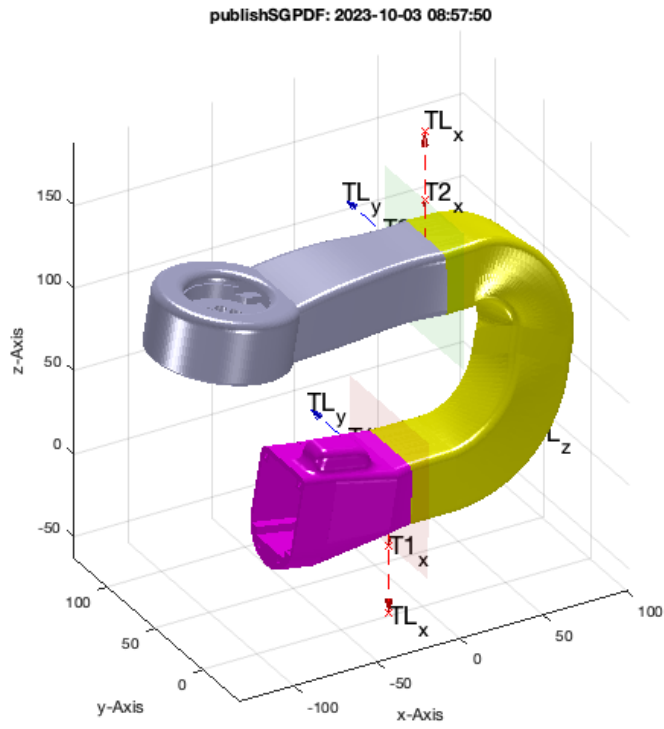
Td =  

-1.0000	-0.0000	0	-50.0000
-0.0000	1.0000	0	50.0000
0	0	-1.0000	50.0000
0	0	0	1.0000

SGcutTexttrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY -150 degree

TLadjustR: NEED TO ROTATE ENDFRAME BY -60 degree



### STRAIGHT IN -Y AND ROTATE

```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([-50 50 150],[0 -1 0],0),'',TYPE); SGN=ans;
```

SGcutTexttrudeT: The extention frame is Td:

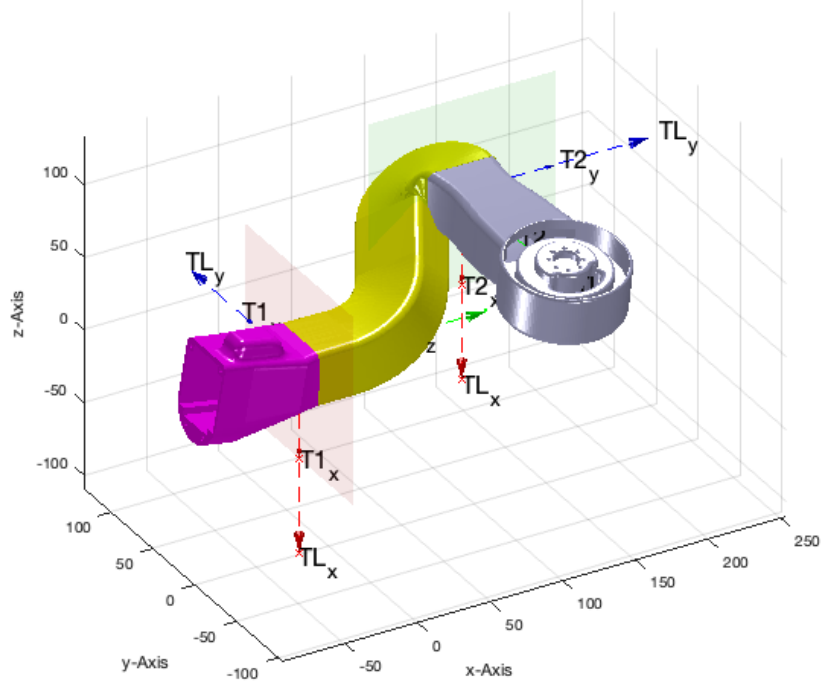
```
Td =
    1     0     0   -50
     0     0    -1    50
     0     1     0   150
     0     0     0     1
```

SGcutTexttrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY -132 degree

TLadjustR: NEED TO ROTATE ENDFRAME BY -49 degree

publishSGPDF: 2023-10-03 08:57:52



```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([-50 50 150],[0 -1 0],+pi/2),'',TYPE); SGN=ans;
```

SGcutTexttrudeT: The extention frame is Td:

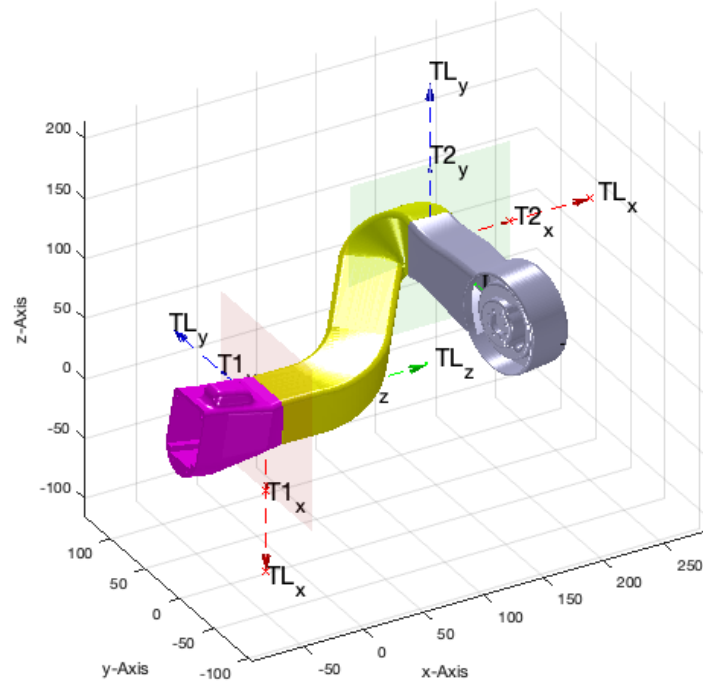
```
Td =
    0.0000    -1.0000         0   -50.0000
         0         0   -1.0000    50.0000
    1.0000    0.0000         0   150.0000
         0         0         0    1.0000
```

SGcutTexttrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY -138 degree

TLadjustR: NEED TO ROTATE ENDFRAME BY 46 degree

publishSGPDF: 2023-10-03 08:57:55



```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([-50 50 150],[0 -1 0],[-pi/2),'',TYPE); SGN=ans;
```

SGcutTexttrudeT: The extention frame is Td:

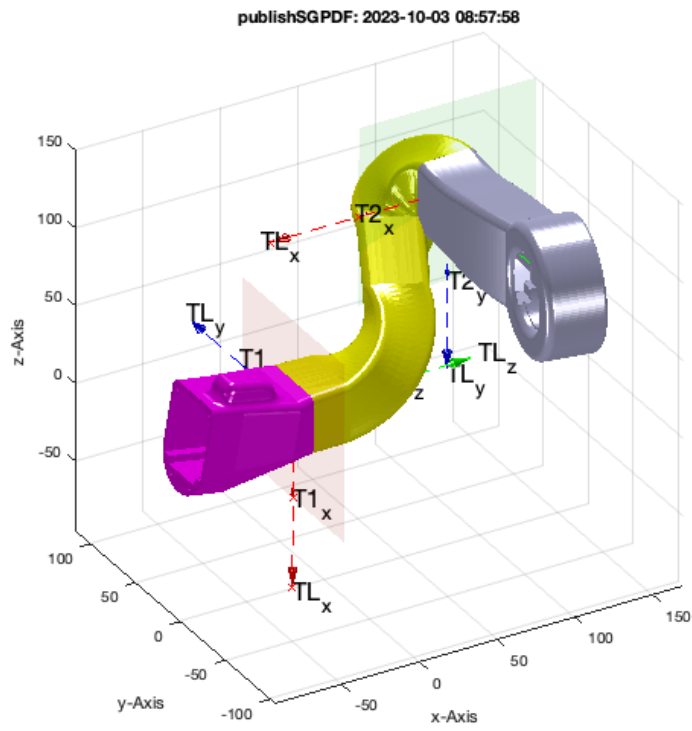
```
Td =
    0.0000    1.0000     0 -50.0000
         0         0   -1.0000  50.0000
   -1.0000    0.0000     0 150.0000
         0         0     0   1.0000
```

SGcutTexttrudeT: straight

TladjustR: NEED TO ROTATE STARTFRAME BY -143 degree

TladjustR: NEED TO ROTATE ENDFRAME BY -165 degree





```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([-50 50 150],[0 -1 0],+pi),'',TYPE); SGN=ans;
```

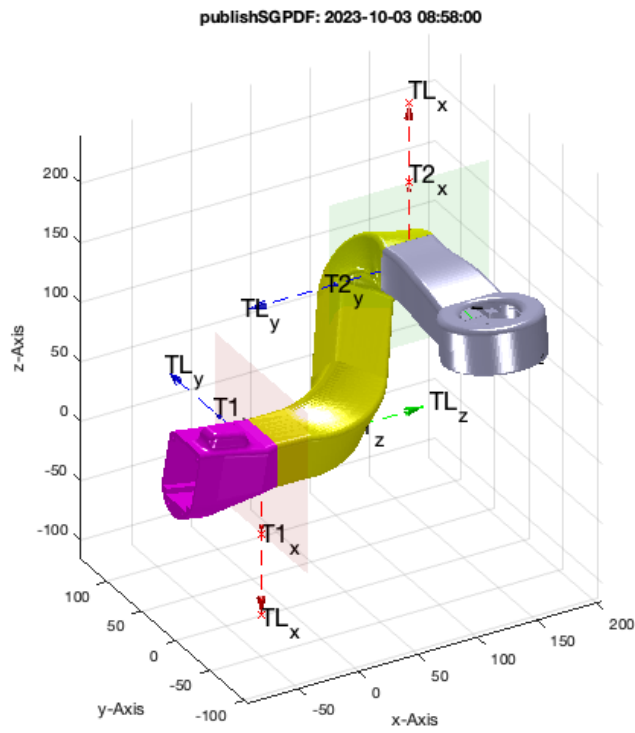
SGcutTexttrudeT: The extention frame is Td:

Td =  
 -1.0000   -0.0000   0   -50.0000  
 0   0   -1.0000   50.0000  
 0.0000   -1.0000   0   150.0000  
 0   0   0   1.0000

SGcutTexttrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY -147 degree

TLadjustR: NEED TO ROTATE ENDFRAME BY 116 degree



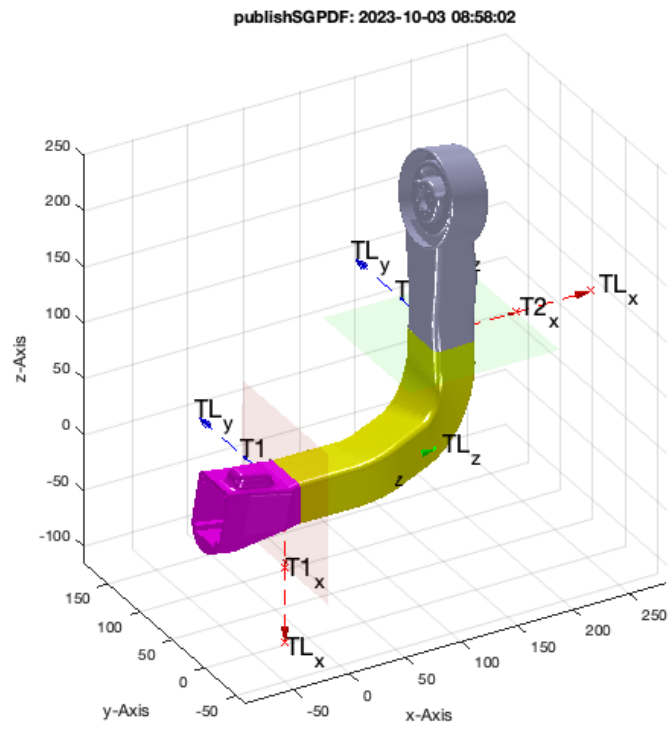
**STRAIGHT in -X AND ROTATE**

```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([-50 50 150],[-1 0 0],0),'',TYPE); SGN=ans;
```

SGcutTexttrudeT: The extention frame is Td:

Td =  
 0 0 -1 -50  
 0 1 0 50  
 1 0 0 150  
 0 0 0 1

SGcutTexttrudeT: straight  
 TLadjustR: NEED TO ROTATE STARTFRAME BY -144 degree  
 TLadjustR: NEED TO ROTATE ENDFRAME BY -28 degree



```
TYPE='straight'; SGcutTextrudeT(SG,T,TofPez([-50 50 150],[-1 0 0],+pi/2),'',TYPE); SGN=ans;
```

SGcutTextrudeT: The extension frame is Td:

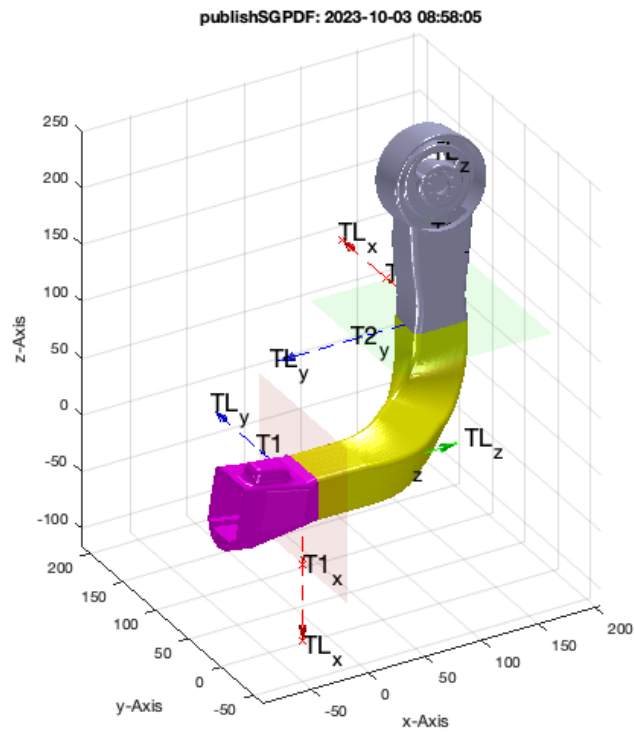
Td =

0	0	-1.0000	-50.0000
1.0000	0.0000	0	50.0000
0.0000	-1.0000	0	150.0000
0	0	0	1.0000

SGcutTextrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY -133 degree

TLadjustR: NEED TO ROTATE ENDFRAME BY 48 degree



```
TYPE='straight'; SGcutTexttrudeT(SG,T,TofPez([-50 50 150],[-1 0 0],[-pi/2),'',TYPE); SGN=ans;
```

SGcutTexttrudeT: The extention frame is Td:

Td =

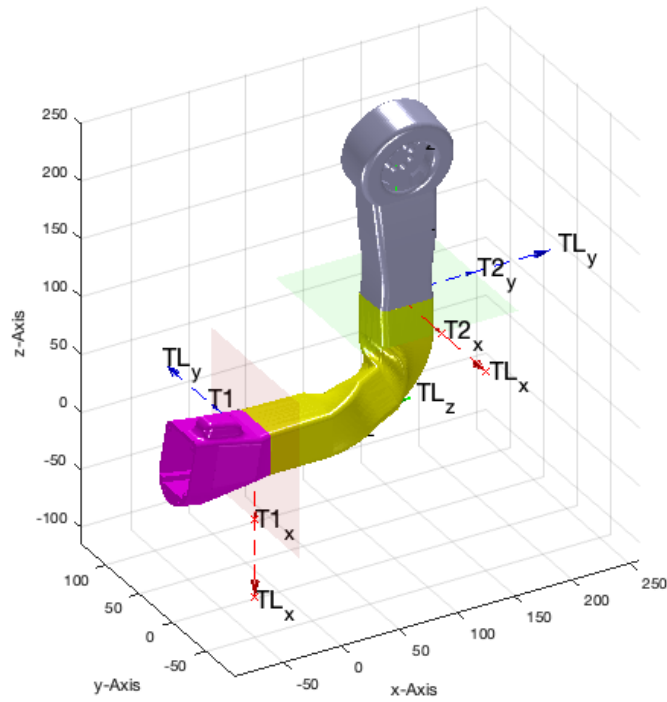
0	0	-1.0000	-50.0000
-1.0000	0.0000	0	50.0000
0.0000	1.0000	0	150.0000
0	0	0	1.0000

SGcutTexttrudeT: straight

TladjustR: NEED TO ROTATE STARTFRAME BY -151 degree

TladjustR: NEED TO ROTATE ENDFRAME BY -118 degree

publishSGPDF: 2023-10-03 08:58:08



```
TYPE='straight'; SGcutTextrudeT(SG,T,TofPez([-50 50 150],[-1 0 0],+pi),'',TYPE); SGN=ans;
```

SGcutTextrudeT: The extension frame is Td:

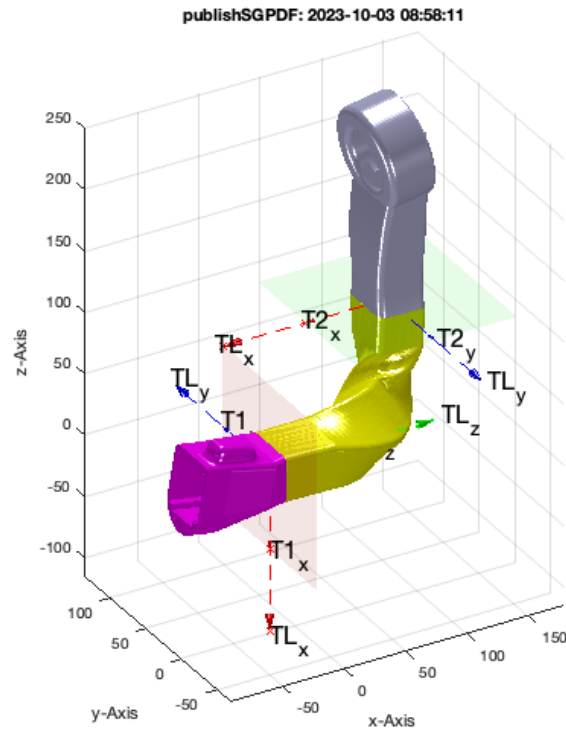
Td =  

0	0	-1.0000	-50.0000
0.0000	-1.0000	0	50.0000
-1.0000	-0.0000	0	150.0000
0	0	0	1.0000

SGcutTextrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY -138 degree

TLadjustR: NEED TO ROTATE ENDFRAME BY 133 degree



### THE COBRA POSE

```
TYPE='straight'; SGcutTextrudeT(SG,T,TofPez([-40 40 40],[-1 0 0],+pi),'',TYPE); SGN=ans;
```

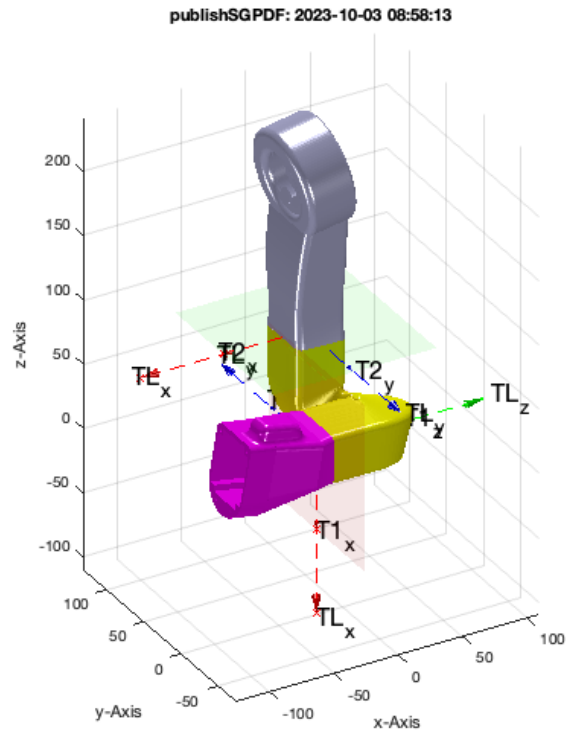
SGcutTextrudeT: The extention frame is Td:

```
Td =
    0         0   -1.0000  -40.0000
    0.0000  -1.0000    0   40.0000
   -1.0000  -0.0000    0   40.0000
    0         0     0     1.0000
```

SGcutTextrudeT: straight

TLadjustR: NEED TO ROTATE STARTFRAME BY -139 degree

TLadjustR: NEED TO ROTATE ENDFRAME BY 26 degree



---

Published with MATLAB® R2023a