Exercise 22: Adding Simulink Signals to Record Frame Movements

2016-12-18: 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. Creating a new SimMechanics System
- 3. Create two links with length 50 and 80 and one or two mounting holes
- 4. Create SimMechanics models for the four links and four joints in different colors
- 5. Create a video of the movements
- 6. Analyze the simulation for 3 Seconds
- 7. Create Simulink signals for all the frames of the four links
- 8. Simulate and record those signals too
- Final remarks on toolbox version and execution date

1. Summary of the already explained aspects of the VLFL-Toolbox

The previous examples of this series covered the following topics:

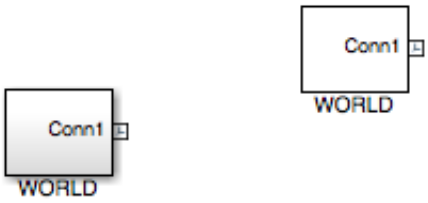
- Example 01: First Steps Using the VLFL-Toolbox for Solid Object Design
- Example 02: Using the VLFL-Toolbox for STL-File Export and Import
- Example 03: Closed 2D Contours and Boolean Operations in 2D
- Example 04: 2.5D Design Using Boolean Operators on Closed Polygon Lists (CPL)
- Example 05: Creation, 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: Adding Simulink Signals to Record Frame Movements

2. Creating a new SimMechanics System

```
smbNewSystem ('SG_LIB_EXP_22');           % Creates the mechansim diagramm
```

```
smbDrawNow;
```

```
Creating temporary directory '/Users/lueth/Desktop/tmp_SG_LIB_EXP_22/'
```

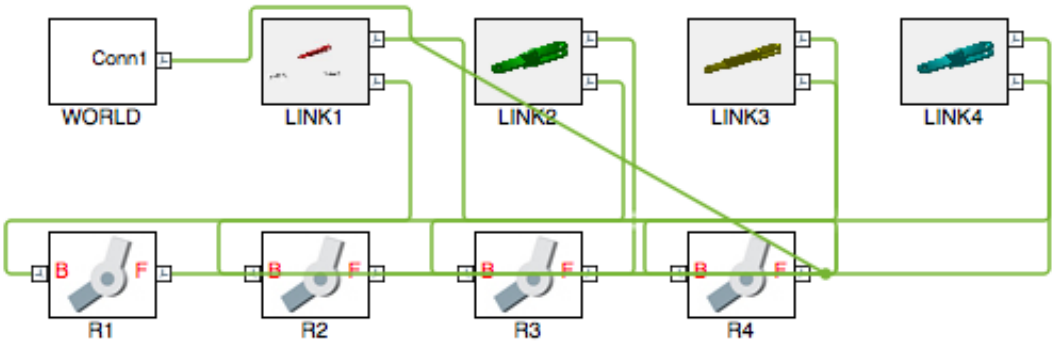


3. Create two links with length 50 and 80 and one or two mounting holes

```
SG1=SGmodelLink(80,'',1,2);              % Creates a long rod with flange
SG2=SGmodelLink(50,'',1,2);              % Creates a short rod with flange
```

4. Create SimMechanics models for the four links and four joints in different colors

```
smbCreateSG (SG1,'LINK1','r');             % Add long rod as LINK1
smbCreateSG (SG2,'LINK2','g');             % Add short rod as LINK2
smbCreateSG (SG1,'LINK3','y');             % Add long rod as LINK3
smbCreateSG (SG2,'LINK4','c');             % 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
smbDrawNow;
```



5. Create a video of the movements

smbVideoSimulation(3); % Show a 3 seconds video

6. Analyze the simulation for 3 Seconds

The result of a simulation is a strucutre that contains SimMultiBody states (xout) and recorded Simulink signals (sim). If there are no Simulink signals, sout is empty.

```
simOut=smbSimulate(3)
```

```
Simulink.SimulationOutput:

    simlog: [1×1 simscape.logging.Node]
      tout: [209×1 double]
      xout: [1×1 Simulink.SimulationData.Dataset]

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

The states contain the parameter = angles/velocity of the joints

```
xout = simOut.get('xout')
```

```
xout =

  Simulink.SimulationData.Dataset
  Package: Simulink.SimulationData

  Characteristics:
              Name: 'xout'
    Total Elements: 8

  Elements:
    1 : 'SG_LIB_EXP_22.R1.Rz.q'
    2 : 'SG_LIB_EXP_22.R1.Rz.w'
    3 : 'SG_LIB_EXP_22.R2.Rz.q'
    4 : 'SG_LIB_EXP_22.R2.Rz.w'
    5 : 'SG_LIB_EXP_22.R3.Rz.q'
    6 : 'SG_LIB_EXP_22.R3.Rz.w'
    7 : 'SG_LIB_EXP_22.R4.Rz.q'
    8 : 'SG_LIB_EXP_22.R4.Rz.w'

  -Use 'get' or 'getElement' to access elements by index or name.
  -Use 'addElement' or 'setElement' to modify elements.
```

There is no Simulink signals yet

```
sout = simOut.get('sout')
```

```
sout =
```

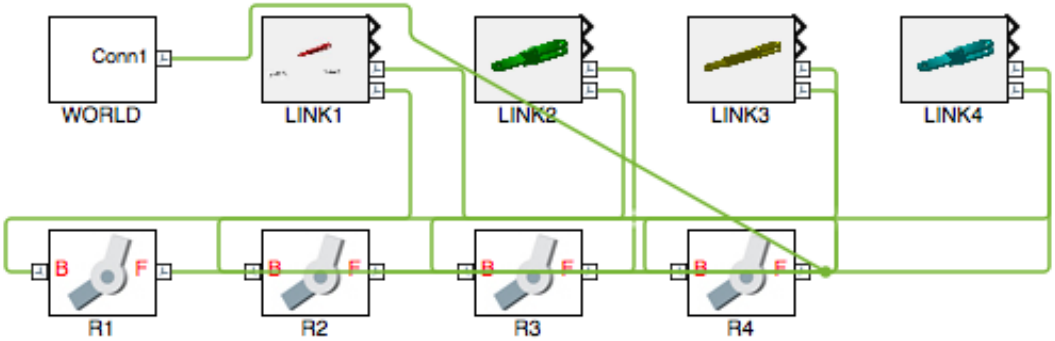
```
[ ]
```

7. Create Simulink signals for all the frames of the four links

```
smbAddFrameSensor ('LINK1.RF');
smbAddFrameSensor ('LINK2.RF');
smbAddFrameSensor ('LINK3.RF');
smbAddFrameSensor ('LINK4.RF');
```

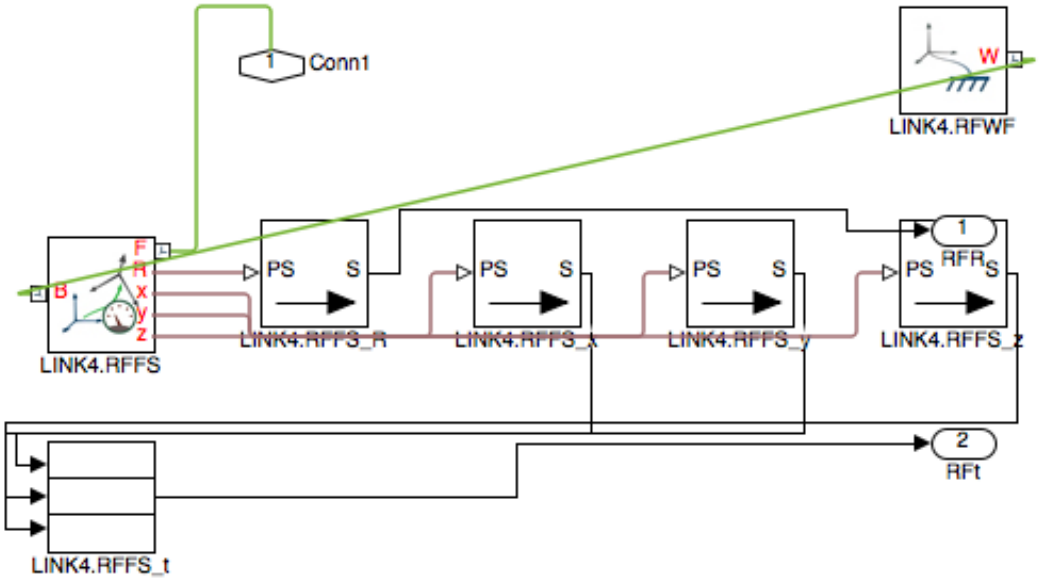
Now, all links have simulink signals and signal output for R and T of the reference frame

```
smbDrawNow;
```



The model of link4 is extendend by a transformation sensor

```
smbDrawNow ('LINK4.RF_T');
```



8. Simulate and record those signals too

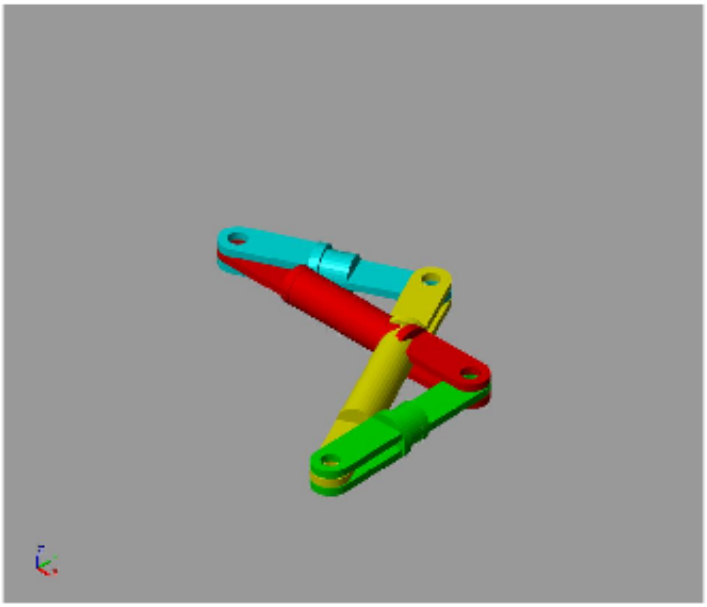
```
simOut=smbSimulate(3)
smbVideoSimulation(3);
```

```
Simulink.SimulationOutput:

    simlog: [1×1 simscape.logging.Node]
      sout: [1×1 Simulink.SimulationData.Dataset]
      tout: [209×1 double]
      xout: [1×1 Simulink.SimulationData.Dataset]

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

........
```



The states contain the parameter = angles/velocity of the joints

```
xout = simOut.get('xout')
```

```
xout =

  Simulink.SimulationData.Dataset
  Package: Simulink.SimulationData

  Characteristics:
              Name: 'xout'
    Total Elements: 8

  Elements:
    1 : 'SG_LIB_EXP_22.R1.Rz.q'
    2 : 'SG_LIB_EXP_22.R1.Rz.w'
    3 : 'SG_LIB_EXP_22.R2.Rz.q'
    4 : 'SG_LIB_EXP_22.R2.Rz.w'
    5 : 'SG_LIB_EXP_22.R3.Rz.q'
    6 : 'SG_LIB_EXP_22.R3.Rz.w'
    7 : 'SG_LIB_EXP_22.R4.Rz.q'
    8 : 'SG_LIB_EXP_22.R4.Rz.w'

  -Use 'get' or 'getElement' to access elements by index or name.
  -Use 'addElement' or 'setElement' to modify elements.
```

TheSimulink signals are related to the reference rotation and translation

```
sout = simOut.get('sout')
T1=smbTofSimOut(simOut,'LINK1.RF'); VL1=squeeze(T1(1:3,4,:))';
```

```
T2=smbTofSimOut(simOut,'LINK2.RF'); VL2=squeeze(T2(1:3,4,:))';
T3=smbTofSimOut(simOut,'LINK3.RF'); VL3=squeeze(T3(1:3,4,:))';
T4=smbTofSimOut(simOut,'LINK4.RF'); VL4=squeeze(T4(1:3,4,:))';
SGfigure; axis on; view(0,90); grid on;
VLplot(VL1,'r.-');
VLplot(VL2,'g.-');
VLplot(VL3,'y.-');
VLplot(VL4,'c.-');
drawnow;
```

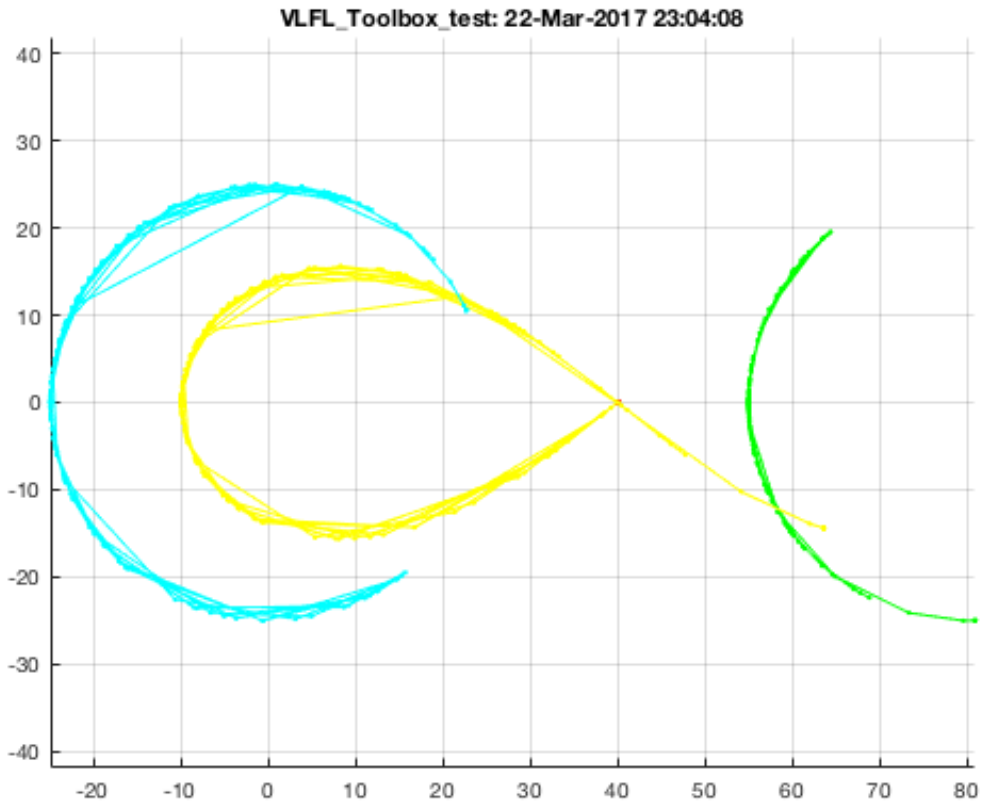
```
sout =

  Simulink.SimulationData.Dataset
  Package: Simulink.SimulationData

  Characteristics:
              Name: 'sout'
    Total Elements: 8

  Elements:
    1 : 'SG_LIB_EXP_22/LINK1/LINK1.RF_T.RFR'
    2 : 'SG_LIB_EXP_22/LINK1/LINK1.RF_T.RFt'
    3 : 'SG_LIB_EXP_22/LINK2/LINK2.RF_T.RFR'
    4 : 'SG_LIB_EXP_22/LINK2/LINK2.RF_T.RFt'
    5 : 'SG_LIB_EXP_22/LINK3/LINK3.RF_T.RFR'
    6 : 'SG_LIB_EXP_22/LINK3/LINK3.RF_T.RFt'
    7 : 'SG_LIB_EXP_22/LINK4/LINK4.RF_T.RFR'
    8 : 'SG_LIB_EXP_22/LINK4/LINK4.RF_T.RFt'

  -Use 'get' or 'getElement' to access elements by index or name.
  -Use 'addElement' or 'setElement' to modify elements.
```



Final remarks on toolbox version and execution date

```
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 23:04:11 by 'lueth' using Matlab 9.1.0.441655 (R2016b) on a MACI64
======================================== Used Matlab products: ============================
==========
map_toolbox
matlab
robotics_system_toolbox
simmechanics
simscape
simulink
================================================================================================
==========
```

- *Tim Lueth, tested and compiled on OSX 10.11.6 with Matlab 2016b on 2016-12-18*
- _____, executed and published on 64 Bit PC using Windows with Matlab 2015a on 2015-xx-xx_

Published with MATLAB® R2016b