Automatische Generierung von Simscape Multibody Modellen auf der Grundlage von Gelenkgetrieben mit der Matlab-Toolbox SG-Library

Tim, Lüth*; Franz, Irlinger**

* Technische Universität München,
Lehrstuhl für Mikrotechnik und Medizingerätetechnik (MIMED)
tim.lueth@tum.de

** Technische Universität München,
Lehrstuhl für Mikrotechnik und Medizingerätetechnik (MIMED)
franz.irlinger@tum.de

Kurzfassung

Der Beitrag stellt erstmals die automatische Erzeugung von Mehrkörper-Simulationsmodellen aus automatisiert konstruierten Mechanismen mit Hilfe von MATLAB-Simscape Multibody vor. Damit lässt sich etwa die Bewegung des Massenschwerpunkts von Mechanismen optimieren. Die dafür notwendige Toolbox wird im Internet zur Verfügung gestellt. Die Autoren haben auf den GTK Tagungen in 2013 und 2015 erstmal vorgestellt, wie man die Glieder und Gelenke von Mechanismen vollständig aus formalen Beschreibungen als Oberflächenmodelle für die additive Fertigung (3D-Druck) mit Hilfe von MATLAB konstruieren kann. Dies ist der Schlüssel für die automatische Optimierung von diversen kinematischen Kennwerten. Mit der Ankopplung an Simscape-Multibody können nun auch die dynamischen Parameter optimiert werden.

Abstract

In this article the automatic generation of dynamic multi-body simulation models from automatically constructed mechanisms with the use of MATLAB-Simscape Multibody is presented for the first time. Thus, for example, the movement of the mass center point of a mechanisms can be optimized. The authors provide the necessary toolbox on the Internet. At the GTK meetings in 2013 and 2015, the authors first presented how to construct the links and joints of mechanisms completely from formal descriptions as surface models for additive manufacturing (3D printing) using MATLAB. This is the key to the automatic optimization of various kinematic characteristics of mechanisms. With the coupling to Simscape-Multibody, the dynamic parameters can now also be optimized.

1 Motivation und Zugang zur Bibliothek

Seit 2010 arbeiten die beiden Autoren an dem Konzept des automatisierten Entwurfs von Mechanismen und Roboter [1, 2]. Dabei soll von der Kinematiksynthese über die Konstruktion bis hin zur Ansteuerung der Gelenkgetriebe eine einheitliche Entwicklungsumgebung verwendet werden. Nur so können automatisierte Optimierungsprozesse realisiert werden. Alle Entscheidungen des Konstrukteurs müssen formal beschreibbar werden. Nachdem leider der SFB-989-Antrag "Modulare Roboter" Anfang 2013 vom DFG-Senat nicht bewilligt wurde, haben wir begonnen, die Forschungsergebnisse im Rahmen einer MATLAB-Toolbox "SG-Library" (solid geometry) zur Verfügung zu stellen. In diese Bibliothek werden nach und nach auch Methoden anderer Gruppen integriert [3] und Ergebnisse von abgeschlossenen Dissertationen aufgenommen [4].

Alle in diesem Artikel vorgestellten Schritte können vom Leser nachvollzogen werden, wenn die SG-Library-Toolbox heruntergeladen und durch Doppelklick installiert wird [5].

2 Entwurf der Körper als Oberflächenmodellen

Wie bereits in [2] vorgestellt, enthält die SG-Lib zahlreiche Funktionen für die Modellierung von festen Körpern und Gelenken. Für planare Mechanismen bieten sich Funktionen an, die geschlossenen Polygonzüge "extrudieren" und auf

diese Weise die notwendigen Koppelglieder erzeugen. Für Gelenkglieder gibt es vordefinierte Funktionen, die auch bereits Zapfen oder andere Verbindungselemente wie Flanschbilder vorsehen, sowie Ebenen für eine Schichtung:

```
A=SGmodelLink2(40); SGfigure; SGplot(A); view(-30,30); % Creates a link with distance 40 mm
A=SGmodelLink2(40,-1,-1); SGfigure; SGplot(A); view(-30,30); % Creates a link with distance 40 mm
A=SGmodelLink2(40,-1,-1,'BLFU'); SGfigure; SGplot(A); view(-30,30); % Creates a link with 40 mm
```

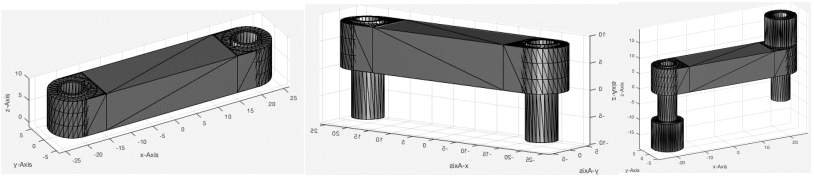

Abb. 1: Drei Beispiele für Gelenke aus geschichteten Gliedern mit und ohne Zapfen

Für alle mit der SG-Lib erzeugten Oberflächenmodelle der Glieder gilt, dass sie in STL-Dateien umgewandelt und danach im additiven Fertigungsverfahren hergestellt werden können.

3 Nutzung von Koordinatensystem-Frames zur Bildung kinematischer Ketten

Die SG-Lib unterstützt eine große Vielfalt der relativen Anordnung von Körper über räumliche Relationen wie beispielsweise "center", "above" oder "aligntop". Für kinematische Ketten spielen jedoch vor allem Koordinatensysteme eine Rolle, die relativ zu einem Körper definiert und an diesem "fixiert" sind, d.h. sie werden bei jeder räumlichen Veränderung des Körpers mitbewegt. Häufig spricht man von *Base-Frame* und *Follower-Frame*, wenn man den Befestigungspunkt zum vorhergehenden Glied bzw. zum nächsten Glied beschreiben will. Diese Frames werden beispielsweise von der Funktion SGmodeLink miterzeugt.

```
SGmodelLink2(40,0,0);          % Base Frame and Follower Frame are on Level 0
SGmodelLink2(40,0,1);          % Base Frame on Level 0 and Follower Frame on Level 1
SGmodelLink2(40,0,1,'BL');     % Base Frame has a button on lower side
```

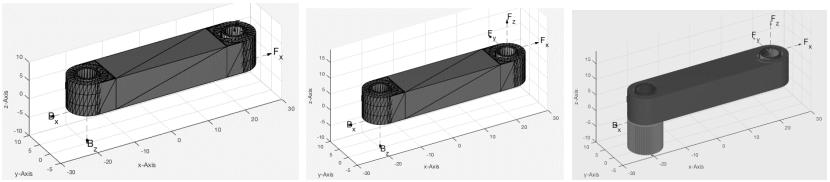

Abb. 2: Drei Beispiele für Glieder für geschichtete Glieder mit Koordinatensysteme **B**ase (B) und **F**ollower (F)

Mit der Nutzung dieser Frames lassen sich dann sehr einfach kinematische Ketten beschreiben:

```
A=SGmodelLink2(40,0,1,'BL'); B=SGmodelLink2(40,0,1); SGTchain({A,B})
% create a F-B fram chain
```

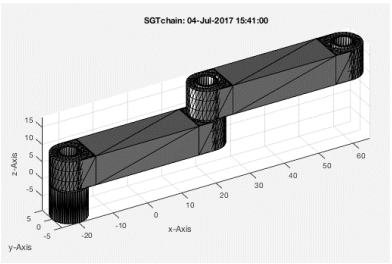

Abb. 3: Verkettung der Koordinatensysteme **F**ollower mit **B**ase

Es gibt auch Funktionen, mit denen automatisiert Flanschflächen in komplexen STL-Geometrien gefunden und selektiert werden können.

4 Grundlage von Simscape Multibody

MATLAB unterstützt mit der *Simulink*-Umgebung das Programmieren mit Funktionsblöcken. Die Linien zwischen den Blöcken stehen für die übergebenen Variablen zwischen den Funktionen. Die Blöcke selbst stehen für die numerisch-mathematische Funktionen zur Verarbeitung der Eingangssignale in die Ausgangsignale. Typischerweise wird eine definierte Abtastfrequenz zum wiederholten Aufrufen der Blockfunktionen verwendet.

Zusätzlich bietet die *Simscape*-Umgebung die Möglichkeit auch physikalische Modelle als Blöcke zu verwenden. In diesem Fall stehen die Blöcke für physische Objekte (und deren Position, Geschwindigkeit, Beschleunigung) und die Linien zwischen den Blöcken für physikalische Wechselwirkungen wie Kräfte und Momente. Diese Darstellung hat den bestechenden Vorteil, dass die Differentialgleichungssysteme von Simscape automatisch erstellt und gelöst werden. Zur Ansteuerung der physikalischen Simscape-Modelle werden oft Signale aus Simulink verwendet, die über einen Block in physikalische Wechselwirkungen gewandelt werden. Aus einem reinen Signalzahlenwert 1 wird beispielsweise ein Drehmoment 1 [Nm].

Simscape Multibody ist eine Blocksammlung von physikalischen Modellen, die ständig ergänzt wird. Es gibt Körper und Gelenke und einen einfachen CSG-Modellierer sowie die Möglichkeit STEP oder STL-Daten zu importieren. Die Trägheitsmomente oder Massenzentren müssen jedoch manuell nachgetragen werde; ebenso Koeffizienten wie Reibung, Dämpfung, Federkonstanten.

Simscape Multibody ist jedoch nicht nur in der Lage Bewegungen der modellierten Körper auf der Basis von wirkenden Kräften und Momenten durch Freischneiden zu berechnen, sondern auch gültige geschlossene kinematische Ketten zu erkennen und deren Schlussbedingungen zu ermitteln. Diese Aufgabe übernimmt ein spezieller Simscape-Block, der *Mechanismen-Configuration* genannt wird. Ein weiterer wichtiger Block ist ein Simscape-Block zum automatisierten Aufstellen und iterativen Lösen der Differentialgleichungssysteme. Dieser Block wird *Solver-Configuration* genannt wird. Ein dritter absolut elementarer Block ist das Ursprungskoordinatensystem, das benötigt wird, um die Richtung wirkender Kräfte angegeben zu können. Es wird *World-Frame* genannt.

Ein minimales Simscape-Multibody-Modell besteht also mindestens aus diesen drei Blöcken: *Mechanismen-Configuration*, *Solver-Configuration* und *World-Frame*.

Nun können an diese drei Blöcke weitere Blöcke angehängt werden, wobei die Blöcke für die Körper der Verbindungsglieder zwischen, oder für die Gelenke selbst, also für die Einschränkung von Bewegungsfreiheitsgraden, stehen. Mit der SG-Lib kann ein solches System per Befehl erzeugt werden.

```
smbNewSystem('GTK_2017')
```

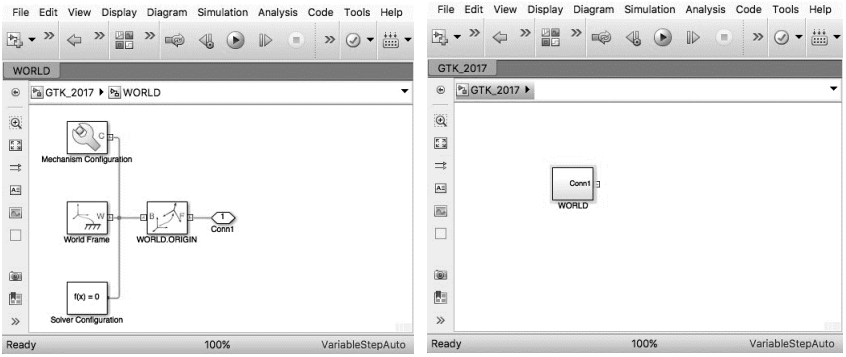

Abb. 4: links) Die drei Minimalblöcke von Simscape-Multibody.
rechts) Die drei Blöcke wurden zu einem Subsystem "WORLD" zusammengefasst, um die Diagramme übersichtlicher zu gestalten.

5 Automatisierte Erzeugung von Gliedern und Gelenken

Nachdem ein Simscape-Multibody System erzeugt ist, können wir die Gelenkglieder als Körper modellieren und anschließend als physikalische Modellblöcke erzeugen. Die folgenden vier Zeilen erzeugen vier Glieder unterschiedlicher Länge und Farbe:

```
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';
```

Es handelt sich um einfache Oberflächenmodelle, die wir mit einigen Befehlen darstellen oder als STL-Datei schreiben können.

```
SGplot(A);                              % Draw solid A
SGTplot(A);                             % Draw solid A with coordinate frames
SGTchain({A,B,C,D};                     % Draw the solids in a kinematic chain
SGwriteMultipleSTL({A,B,C,D})           % Writes the four links as STL-Files
```

Um diese jetzt als Simscape-Multibody-Blöcke nutzen zu können, benötigen wir die folgenden Befehle. Danach sehen wir die Blöcke im Simscape-Diagramm:

```
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
```

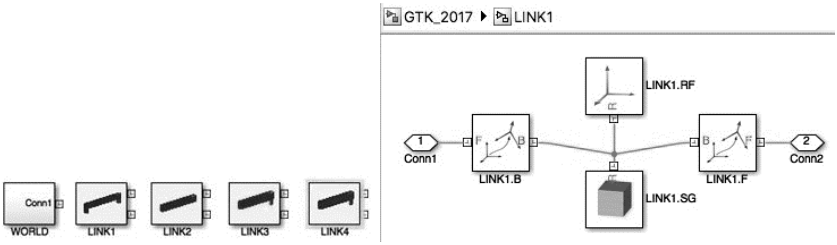

Abb. 5: links) Das Diagramm enthält vier Blöcke als Gliedmodelle.
rechts) Jeder Körper besteht aus einer Geometriebeschreibung LINK.SG, einem Base-Frame LINK.B und einem Follower-Frame LINK.F sowie einem Referenzkoordinatensystem LINK.RF für die Geometrie.

Nun wollen wir die Glieder über Drehgelenke, d.h. rotatorische Freiheitsgrade verbinden. Dazu gibt es einen Befehl, der es auch erlaubt die Frames der Blöcke anzugeben, die verbunden werden sollen:

```
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
```

Im Diagramm sind nun die Verbindungen der geschlossenen kinematischen Kette zu sehen.

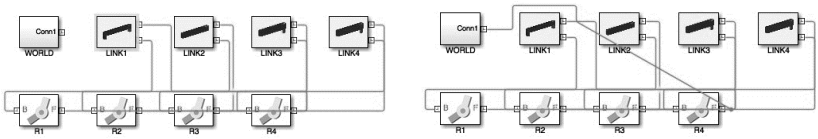

Abb. 6: links) Das Diagramm enthält nun vier Blöcke und vier verbindende Einschränkungen der Freiheitsgrade, hier als Rotationgelenk. rechts) Das Modell ist mit dem WORLD-Block (*Mechanismen-Configuration*, *Solver-Configuration* und *World-Frame*) verbunden.

Mit einem weiteren Kommando ergänzen wir die Lage der Kette zum Koordinatenursprung.

```
smbCreateConnection('WORLD.ORIGIN','LINK1.B'); % Connect Linkage to World Frame
```

Jetzt können wir das Modell unter dem Einfluss der Schwerkraft als Mehrkörpersystem simulieren. Dabei können Grenzen der Bewegung aufgrund der geometrischen Einschränkungen der Schlussbedingung erkannt werden. Eine Kollisionsberechnung findet in Simscape Multibody jedoch nicht statt. Diese wird von der SG-Lib extern anboten.

```
smbSimulate(4);              % Simulation für 4 Sekunden
smbVideoSimulation (4);      % Video erzeugung der Simulation für 4 Sekunden
```

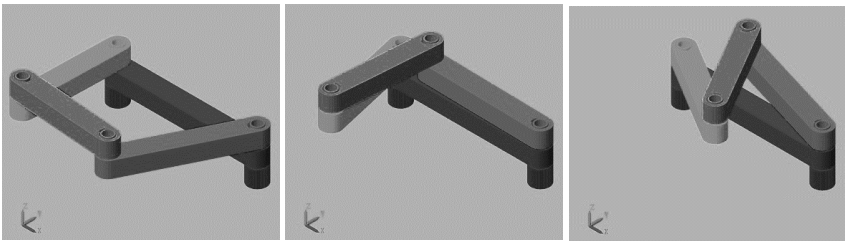

Abb. 7: Das Diagramm enthält nun vier Blöcke und vier verbindende Einschränkungen der Freiheitsgrade, hier als Rotationgelenk. Das Modell ist mit dem WORLD-Block (*Mechanismen-Configuration*, *Solver-Configuration* und *World-Frame*) verbunden.

6 Simulation von Positions- oder Drehmomentvorgaben

Möchte man das Mechanismenmodell nicht nur durch Schwerkrafteinfluss bewegen, dann müssen wir entweder einen Antrieb mit Positionsvorgabe oder Drehmomentvorgabe modellieren. Diese Blöcke waren in Matlab 2017a noch nicht vorhanden und wurden daher mit der SG-Lib realisiert.

Mit einem Kommando wird nun ein Motormodell erzeugt, das Position, Geschwindigkeit und Beschleunigung aus einem Positionsvorgabensignal als Simulink-Block übernehmen kann und in physikalische Simscape-Wirkungen wandelt.

Ein zweites Kommando erzeugt einen Simulink-Block, der eine Cosinus-Signalvorgabe für den Motor erzeugt.

```
smbCreateDrive ('R1');                              % Creates a Driver for Simscape
smbCreateSineWave ('Cosinus','R1_DRIVE/1');         % Creates a Simulink Cosinus Signal
smbSimulate(1);                                     % Simulate for 1 Second
```

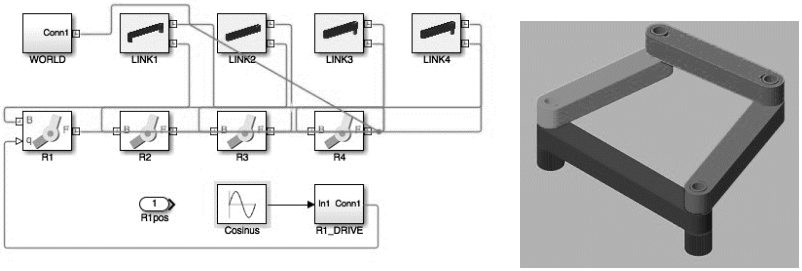

Abb. 8: links) Im Diagramm ist nun ein Antrieb mit einer Positionsvorgabe für das Gelenk 1 zu sehen. rechts) In der Simulation bewegt sich der Körper nun gleichförmig. Die Simulation bricht ab, wenn das Gelenk der Winkelvorgabe nicht mehr folgen kann.

Die resultierende Bewegung sieht ähnlich aus, doch wird die Simulation jetzt beim Anschlag aufgrund geometrischer Randbedingungen abbrechen, da das Gelenk der Positionsvorgabe nicht mehr folgen kann.

Alternativ können wir einen Antrieb mit Drehmomentvorgabe erzeugen, in dem wir den Antrieb mit der Positionsvorgabe umwandeln. Anstatt einer Cosinusbewegung erzeugen wir einen Simulinkblock, der ein konstantes Drehmoment vorgibt.

```
smbSetJointInputTorque('R1');              % Convert the position into a torque drive
smbCreateBlockConst('C','R1_DRIVE/1',-5);  % Constant torque: -5 Nm
smbSimulate(1);                            % Simulate for 1 Second
```

In der Anschlagssituation wird ein Gegenmoment aufgebracht, so dass die Simulation nicht abbricht, sondern das Gelenk zwar zurückschlägt aber dann wieder in die gewünschte Bewegung eintritt.

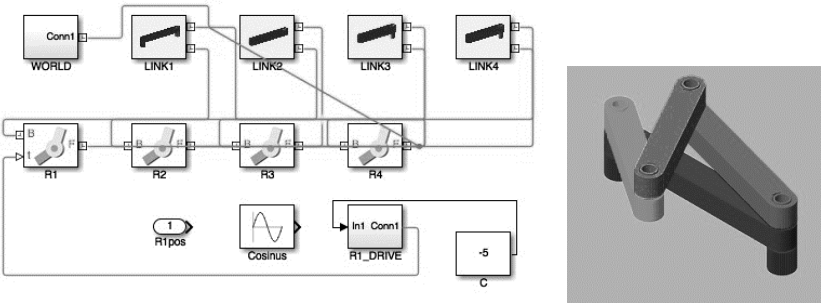

Abb. 9: links) Im Diagramm ist nun ein Antrieb mit Drehmomentvorgabe für das Gelenk 1 zu sehen. Rechts) In der Simulation bewegt sich der Körper nun gleichförmig.

7 Aufzeichnung der Bewegung bei Schwerkraft oder Antrieb

Die visuelle Darstellung der Bewegung ist sehr anschaulich und hilfreich. Für eine automatische Optimierung der Gliedbewegung ist es jedoch notwendig, die simulierten Bewegungen aufzuzeichnen und nachträglich auswerten zu können. Aus diesem Grund wurde eine Routine erstellt, die automatisch ein Simscape-Multibody-Modell analysiert, nach Körpern sucht und für diese "Simulink-Sensoren" an die Referenzframes der Körpergeometrien bindet. Die Funktion liefert über einen definierten Zeitraum die Koordiantenframes der Körper zurück.

```
[~,BNi,SGi,Ti,tm]=smbFullModelSimulation(1);          % Body Names, Solid Geometry, Frames, time

BNi =
  4×1 cell array
    'LINK1.SG'
    'LINK2.SG'
    'LINK3.SG'
    'LINK4.SG'
Ti =
  4×1 cell array
    [4×4×223 double]
    [4×4×223 double]
    [4×4×223 double]
    [4×4×223 double]
```

In diesem Fall gibt es 223 Zeitpunkte innerhalb der 1 Sekunde für die das Modell als Mehrkörpersystem simuliert wurde. Für die 4 Körper LINK1 bis LINK4 gibt

es jeweils 223 homogene Transformationsmatrizen relativ zur Geometrie SGi. Für den Startpunkt lauten die Matrizen von Link 1:

```
T2=Ti{2};  T2(:,:,1)

ans =
-0.8428    0.5382   0         49.7163
-0.5382   -0.8428   0        -16.1473
 0         0        1.0000     9.0000
 0         0        0          1.0000
```

Mit Hilfe dieser Bewegungsmatrizen können Geschwindigkeit, Beschleunigung der Körper sowie potentielle Kollisionen und Geometrieanpassungen der Glieder nachvollzogen werden. Jeder beliebige Zeitpunkt kann nachgestellt werden.

8 Erzeugung einer speziellen Pose zu einer Momentaufnahme

Der Aufruf des Kommandos:

```
SGofsmbFullModelSimulation(SGi,Ti,tm,5)   % Create the Solid configuration for time 5 s
```

Erzeugt für die gegebenen Solids in der Liste SGi eine Darstellung der Pose zu dem jeweiligen Moment.

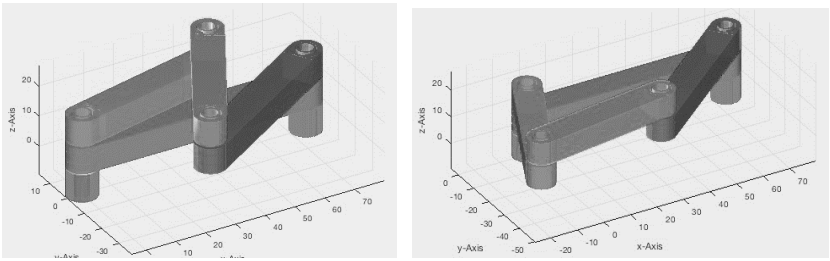

Abb. 10: links) Startzeitpunkt t=0s, rechts) Zeitpunkt t=1s.

Es ist leicht verständlich, dass es nun natürlich leicht möglich ist, die Geometrien der Gelenke zu variieren und auf Kollisionen mit sich selbst oder mit einer Umgebung zu prüfen. Solange die Bedingungen für die Abstände von Base-

Frame und Follower-Frame eingehalten werden sind die berechneten Matrizensequenzen allgemein gültig und können zur Berechnung der Bewegung und der Kollisionen herangezogen werden.

Die Optimierung verläuft in zwei Richtungen bzw. hat zwei Optionen:

- Modifikation der räumlichen Gestaltung der Gliedgeometrien zur Kollisionsvermeidung auf der Basis unveränderter der Abstände oder Orientierung der Frames.
- Modifikation der Abstände oder Orientierung der Frames. Dies führt zu einer Notwendigkeit der erneuten Simulation und Bewegungsaufzeichnung der Mehrkörpersimulation.

9 Additive Fertigung des simulierten Gelenks: 3D Druck

Für die Analyse des Mechanismus kann es sinnvoll sein, universelle Glieder mit integrierten Gelenkanteilen für die Simulation zu benutzen. Es ist dann möglich erst nach Abschluss der Planungsaufgabe zu analysieren, welche endgültigen Teile des Mechanismus sich ergeben.

Die Ausgangsbasis für die Bewegungsplanung waren 4 Glieder mit integriertem Gelenkanteil. Die aus der Bewegung sich ergebenden Elemente bestehen jedoch aus 8 Elementen.

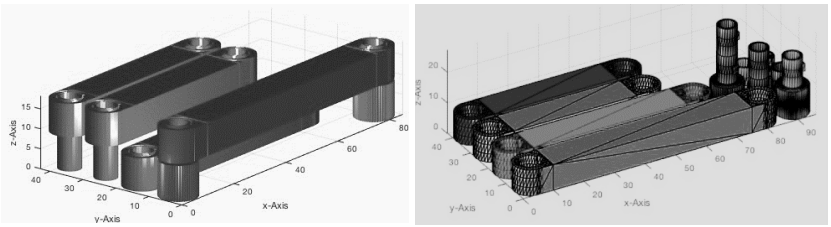

Abb. 11: links) 4 Gelenke als Ausgangsbasis für die Bewegungsplanung rechts) Automatisiert gefundene Komponenten des Bewegungsmechanismus. Man erkennt die selbständig verbundenen verlängerten Achsen

Abschließend sollte noch einmal darauf hingewiesen werden, dass die Bibliothek natürlich auch für die Mehrkörpersimulation und die additive Fertigung offener Mechanismen/Roboter verwendet werden kann.

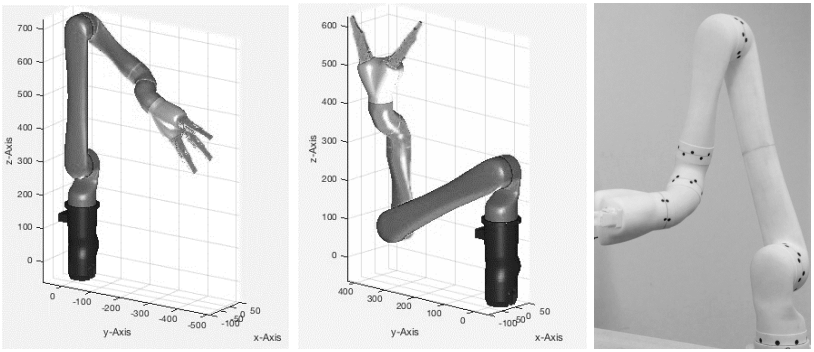

Abb. 12: Zwei Posen für den Roboter JACO der Firma KINOVA und additiv gefertigter JACO (Sep/2013)

Ausblick

Wir hoffen in den nächsten Jahren zu zeigen, dass auch eine integrierte FEM-Berechnung für die Gestaltung der Glieder verwendet werden kann. Ebenfalls sollen die Verfahren aus [3] weitgehend in die SG-Lib aufgenommen werden.

Danksagung

Bedanken möchten wir uns bei den wiss. Angestellten Dr.-Ing Daniel Roppenecker, Dr.-Ing. Mattias Träger, Dipl.-Ing. Yannick Krieger, M.S. und Dipl.-Ing. Christina Hein, M.S. für ihre Beiträge zur Bibliothek oder zum Entwurfsknowhow sowie bei den Studierenden Gweni-Viani Alonso Aruffo, Florian Schleich und Christopher Schlichter für das Evaluieren der Funktionen.

Literatur

[1] Lueth, T.C. und Irlinger, F. (2013). "Berechnete Erzeugung von dreidimensionalen Oberflächenmodellen im STL-Format aus der Beschreibung planarer Mechanismen fu□r die Generative Fertigung durch Selektives-Lasersintern [Computational 3D Surface Generation of Planar Me-

chanismus using STL File Format for Generative Manufacturing by Selective Laser Sintering]". Beitrag im Konferenzband *10. Kolloquium Getriebetechnik*. TU Ilmenau. Sep. 11-13, 2013. pp. 267–284.

[2] Lueth, T. (2015). SG-Library. "Entwicklung einer konstruktiven MATLAB-Toolbox zur räumlichen Modellierung von Körpern, Gelenken und Getrieben". *11. Getriebetechnik-Kolloquium 2015*. TU Mu□nchen. September 2015. pp. 183–203.

[3] Kevin Russell, Qiong Shen, Raj S. Sodhi (2013). *Mechanism Design - Visual and Programmable Approaches.* CRC Press.

[4] Träger, M.F., Krohmer, E., Krieger, Y.S. und Lueth, T.C. (2015). "Automatisierte Konstruktion von Zahnradgetrieben für die Herstellung mittels Rapid-Prototyping-Verfahren.". In Lueth, T.C., Irlinger, F. und Abdul-Sater, K. (Eds.). *11. Kolloquium Getriebetechnik*. Garching b. München. Deutschland. 28.-30. September 2015. pp. 235–254.

[5] Lueth, T. (2017). "Solid-Geometry-Library MATLAB-Toolbox 3.9". download at:
https://www.mimed.mw.tum.de/fileamin/w00bhh/www/Matlab_Toolboxes/SolidGeometry_3.9.mltbx
or https://www.mimed.mw.tum.de/research/tim-lueths-research,
available since 2017-July-03.