

Musculo-skeletal simulation with Simulink and SimMechanics

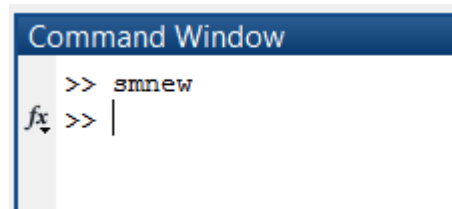
Gruppenmitglieder: Julia Schaal, Tabea Pomes, Valentin Claus, Marc Jacob

Einführung

Das hier gezeigte Wiki soll ein How-To darstellen, wie man mittels einem bereits bestehenden Muskelmodell von Häufle et al. [1] eine obere Extremität simulieren kann. Dazu bedienen wir uns eines einfachen Modells, einem Oberarm mit jeweils einem Freiheitsgrad in Schulter- sowie Ellbogengelenk. Aufgrund dieses ersten, einfachen Ansatzes ist es schließlich möglich, kompliziertere Sachverhalte zu modellieren und der Veranschaulichung zugänglich zu machen. Dabei lässt sich diese Grundstruktur auf weitere Elemente, wie zum Beispiel die unteren Extremitäten, übertragen, indem man entsprechende Module austauscht beziehungsweise ersetzt.

Simulink starten

Simulink wird über die Matlaboberfläche im 'Command Window' mittels dem Befehl 'smnew' gestartet.



```
Command Window
>> smnew
fx >> |
```

Abbildung 1: Start von Simulink über den Befehl 'smnew'

Bei Start sehen wir zwei Fenster, einmal die Library, die die verschiedenen Bausteine enthält und zum Anderen das aktuelle Projekt, mit dem wir uns beschäftigen wollen. Dort sind schon gewisse Grundstrukturen vordefiniert, wie z.B. das Weltkoordinatensystem. Im nächsten Schritt können wir nun die passiven Strukturen unseres Armmodells über die vordefinierte Grundelemente integrieren.

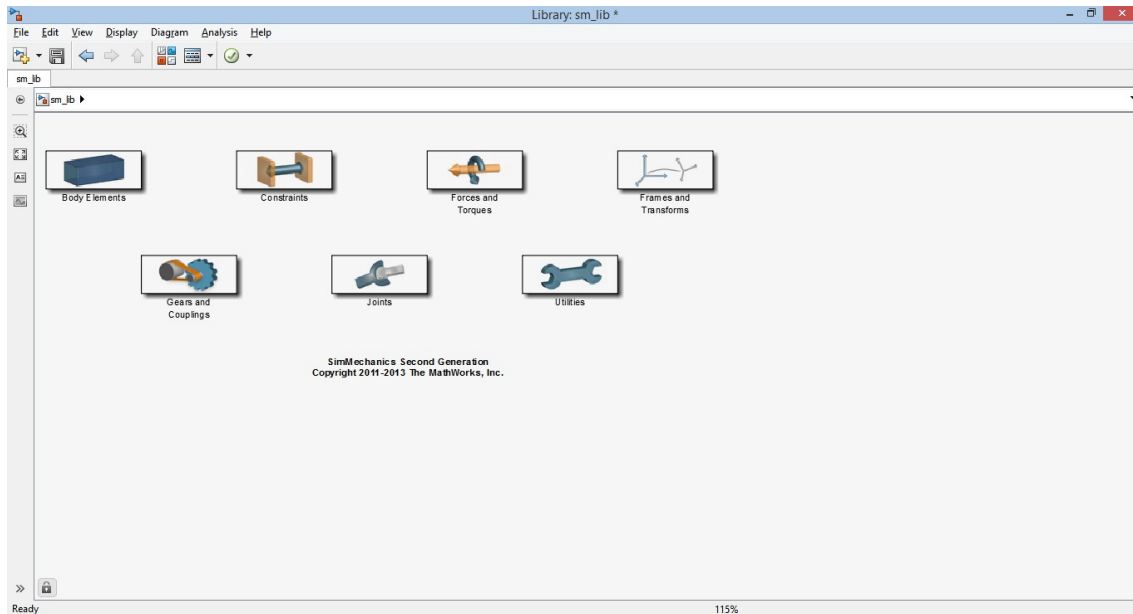


Abbildung 2: Die Library

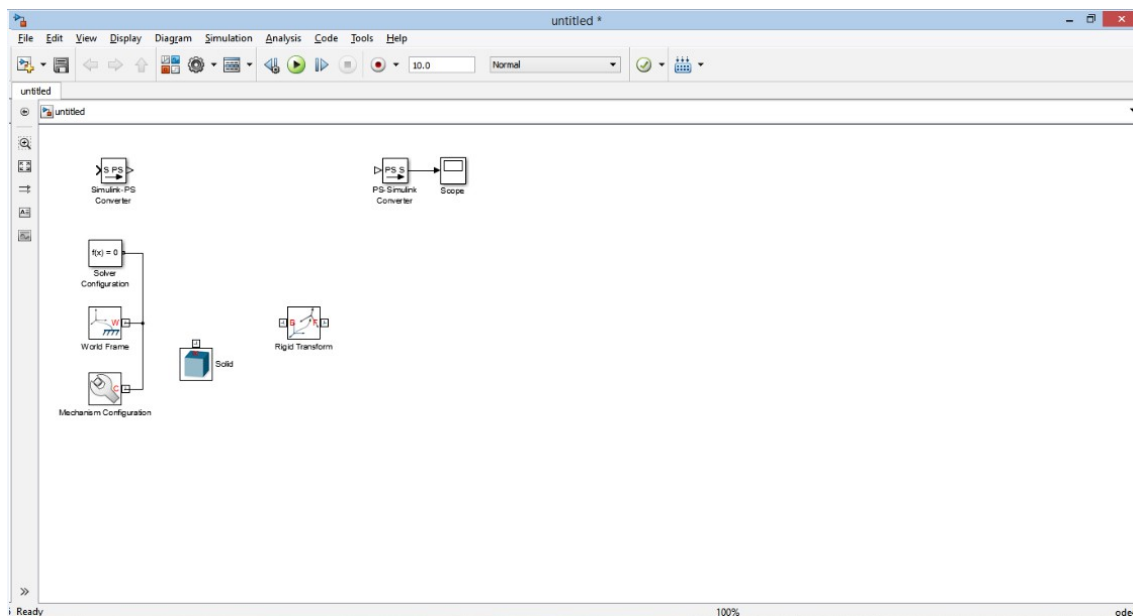


Abbildung 3: Die Projektoberfläche

Modellierung der passiven Strukturen

Zunächst benötigen wir das Schultergelenk. Dazu wählen wir als Vereinfachung ein Gelenk mit einem Freiheitsgrad. Das entspricht der Bewegungsmöglichkeit des Schultergelenks zwar nicht im geringsten, reicht für unser Modell jedoch zunächst einmal vollkommen aus. Dazu gehen wir in die **Library** -> **Joints** und wählen das Element **revolute joint** aus. Dieses ziehen wir per drag-and-drop in das Fenster des aktuellen Projekts und setzen es neben das vordefinierte Weltkoordinatensystem. Dieses Gelenk verbinden wir nun mit dem Weltkoordinatensystem, indem wir eine Linie von der

linken Verbindungsstelle zu der bereits existierenden Verbindungsstelle des Weltkoordinatensystems mit dem restlichen Elementen ziehen.

Dann gehen wir im nächsten Schritt vom Ursprung zu der Koordinate des Schwerpunktes des Oberarms. Dazu wählen wir in der **Library** -> **Frames and Transforms** das Element **Rigid Transform** aus setzen es neben das zuvor ausgewählte Gelenk und verbinden es mit diesem über die bereits genannte Methode. Macht man nun einen **Doppelklick** auf dieses Element, so erscheint ein extra Fenster, in dem man die Parameter definieren kann. Dort geben wir unter **Translation** -> **Standard Axis** im Parameter **+Y** einen Offset von **-0.13** (m) ein. Dies entspricht einer Verschiebung von 13cm, der Hälfte der Länge unseres modellierten Oberarms von 26cm. Schließlich wird an diese Position das Modell des Oberarms gesetzt. Zur Vereinfachung nehmen wir dabei einen Klotz. Dazu wählen wir in der **Library** -> **Body Elements** das Element **Solid** aus und setzen es an das Modell an. Wiederum wird es mit einer Verbindung mit dem **Rigid Transform** verbunden. Mittels **Doppelklick** auf dieses Element können wir die Parameter dafür definieren. Dazu lassen wir bei **Geometry** die voreinstellung **Brick** und wählen für die **Dimensions** -> **[0.07 0.26 0.07]** (m). Unser vereinfachtes Oberarmmodell besteht also als einem Klotz der Kantenlänge 7*26*7cm.

Nun brauchen wir noch den Ellbogen sowie den Unterarm. Dazu kommt wieder eine Verschiebung in den Schwerpunkt des Ellbogens, wobei wir das bereits bestehende '**Rigid Transform 1**' kopieren können (dabei kopieren wir auch die enthaltenen Parameter). Dies setzen wir also nun neben das bereits bestehende Modell und verbinden es wieder über den linken Eingang mit dem restlichen Modell. Als nächstes kopieren wir das Gelenk ('**Revolute Joint**') und bauen es wiederum an. Zum Schluss kommt nochmal eine letzte Verschiebung (**Rigid Transform** wieder kopieren), dem ein weiteres **Solid** (ebenfalls kopierbar) folgt. Für das zuletzt eingefügte 'Solid 2' passen wir nun noch die Parameter über einen **Doppelklick** auf das Element an und setzen für die **Dimensions** -> **[0.05 0.26 0.05]** (m)

Definieren der Randbedingungen

In diesem Schritt müssen wir noch die Umgebung, wie zum Beispiel die Schwerkraft definieren. Dazu doppelklicken wir **Mechanism Configuration** und setzen bei **Gravity** -> **[0 -9.80665 0]** (m/s²). Die Schwerkraft soll somit in Y-Richtung wirken. Dies ist notwendig, da die Gelenke jeweils um die Z-Achse drehen. Das Modell sollte nun erst einmal wie in Abb. 4 dargestellt aussehen.

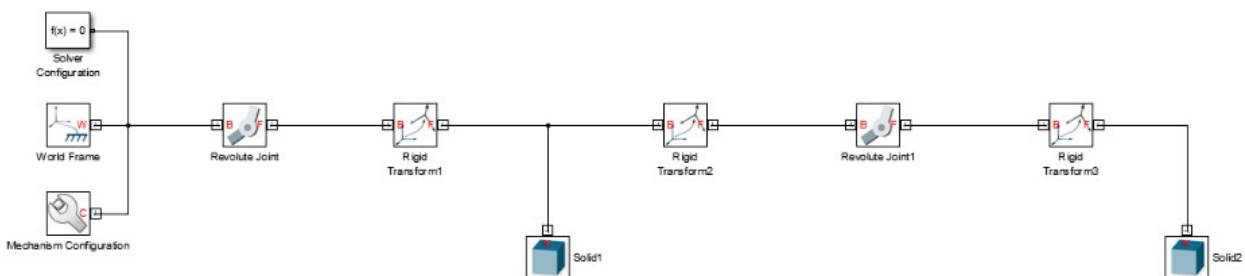


Abbildung 4: Erste Implementierung der passiven Strukturen

Erweiterung des Modells und Implementierung des Muskelmodells

Zuerst müssen wir ein paar zusätzliche Parameter für das Ellbogengelenk aktivieren. Dafür gehen wir mittels **Doppelklick** auf das entsprechende Element in die Optionen und aktivieren bei **Z Revolute Primitive (Rz)** -> **Actuation** in dem Dropdown-System für **Torque** -> **Provided by input**. Ebenso setzen wir bei **Sensing** einen Haken bei **Position** und **Velocity**. Bei Bestätigung erscheinen nun zusätzliche Ein- und Ausgänge an dem Gelenk, die wir für die Implementierung des Muskelmodells benötigen. Dazu kopieren wir das von Häufle et al. [1] erstellte Muskelmodell in unser Projekt. Nun müssen wir noch die Bedingungen, wie im Podcast besprochen, in die Simulink Umgebung einprogrammieren. Dazu müssen wir die Variablen zuerst mittels den vordefinierten **SPS-convertern** umwandeln. Dazu kann man die entsprechenden Bausteine kopieren. Wenn man einen Baustein anwählt und die Tastenkombination **Strg+r** drückt, so kann man Bauelemente rotieren. Wir benötigen nun zwei mal den **PS->S-Converter** und ein mal den **S->PS-Converter**. Die Gleichungen, wie im Podcast besprochen, lassen sich dann über die Elemente '**Gain**' sowie einer Konstanten '**MPI_MTC_init**' verwirklichen. Das **Gain** entspricht dabei dem Ansatzradius des Muskels am Unterarm, der zur Vereinfachung konstant als 2cm angenommen wurde. Die entsprechenden Symbole dazu findet man im **Library Browser**.



Abbildung 5: Symbol für den Library Browser

Das fertige Modell könnte schließlich wie folgt aussehen

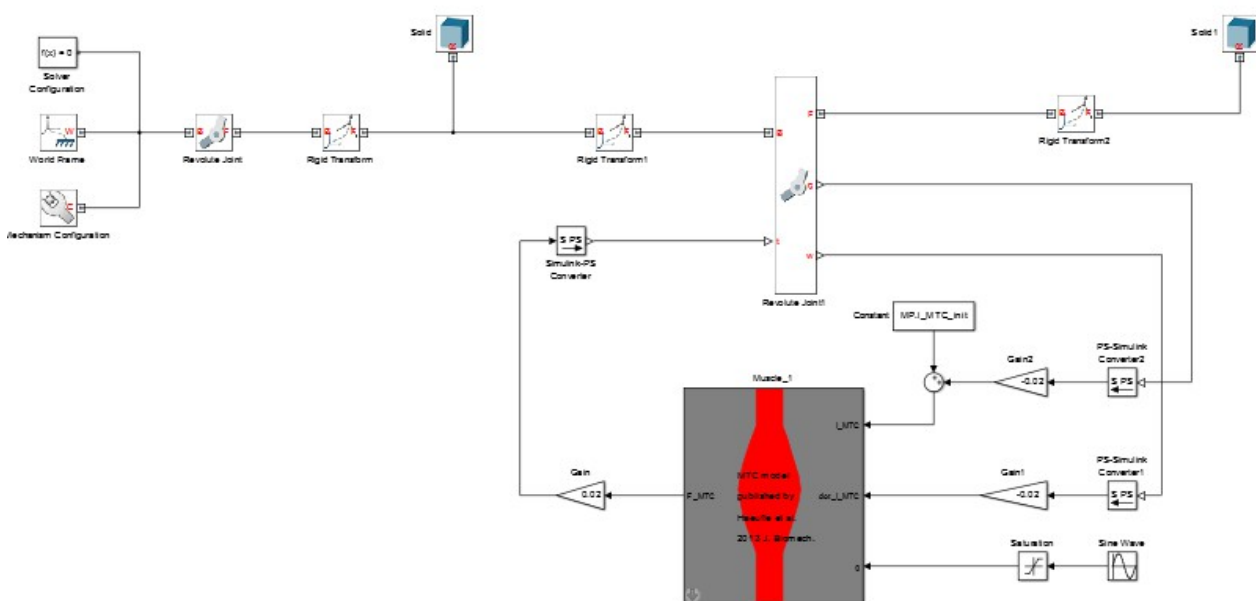


Abbildung 6: Fertiges Arm-Muskel-Modell

Die Konstante *MPI_MTC_init* muss schließlich noch über das Muskelmodell in Matlab eingeladen werden.

Die Ansteuerung (*q*) wurde in diesem Modell mit einer Sinus-Kurve verwirklicht. Die Amplitude wurde dazu auf 0.2 gesetzt und die Frequenz auf 1 rad/sec. Dem wurde noch eine Sättigung (Saturation) mit 1 als Maximalwert (100% Aktivierung) und 0.001 (0.1% Aktivierung) als Minimalwert angefügt.

Quellen

[1] D.F.B. Haeufle et al. / Journal of Biomechanics 47 (2014) 1531–1536