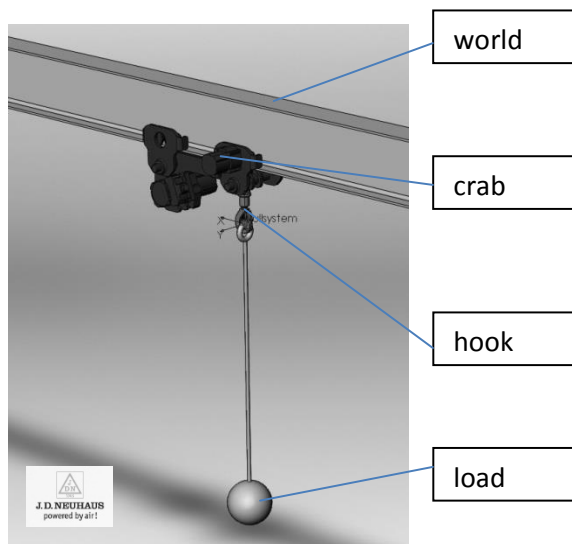


Einführungsbeispiel PyMbs

Mit Hilfe von Python wird eine typische Aufgabenstellung bei der Analyse des dynamischen Verhaltens räumlicher mechanischer Strukturen bearbeitet. Grundlage der Arbeiten bildet das an den beteiligten Instituten entwickelte Python-Paket **PyMbs** (Python Multibody Systems). Die Software dient der Modellbildung und Simulation von Starrkörpersystemen.

Ausgehend von einem einfachen, gegebenen Codebeispiel werden grundlegende sprachliche Konstrukte erläutert. Durch die selbständige Erweiterung des Quellcodes soll das Arbeiten mit Skripten und dem Interpreter erlernt und geübt werden. Durch die graphische Darstellung des Beispielsystems ist jederzeit eine Erfolgskontrolle gegeben.



Aufgaben:

1. Führen Sie das gegebene Skript „Laufkatze.py“ in Spyder aus und prüfen Sie die Kinematik des Systems.
2. Erweitern Sie das mechanische System um einen Körper *Last*, der am Haken hängt. Die Last soll am Haken um die y-Achse drehbar sein. Nutzen Sie die Datei '*Last_01.stl*' für die Visualisierung. Prüfen Sie das Ergebnis in der Simulation auf Plausibilität.
Parameter der Last: Masse: 38kg, Schwerpunkt: [0,0,-1.28], Trägheit: $\text{diag}([2.28, 2.28, 0.15])$
3. Führen Sie im Skript eine boolsche Variable ein, die festlegt, ob der Körper *Last* im System vorhanden sein soll oder nicht. Erfragen Sie die Auswahl vom Benutzer und zeigen Sie einen kurzen Text zur getroffenen Auswahl in der Konsole an.
4. Laden Sie die Datei '*param.txt*', die die Werte der Massen enthält. Lesen Sie die Werte der Massen aus und nutzen Sie diese in `world.addParam`

Zusatzaufgaben

1. Bauen Sie ein Seil, indem Sie eine variable Anzahl von Körpern ('*Segment_01.stl*') hinzufügen.

Benutzen Sie eine Schleife, in welcher jeweils ein nachfolgender Körper (*segment*) an den vorhergehenden Körper (*vorgaenger*) angeschlossen wird.

Werte für Segmente: $mass=0.1$, $cg=[0,0,-0.1]$,
 $inertia=diag([0.01, 0.01, 0.01])$

Definieren Sie Anschlusspunkte mit dem Befehl

```
segment.addFrame('joint', p=[0,0,-0.2])  
segment.addFrame('visual', p=[0,0,-0.1], R=rotMat(pi/2, 'x'))
```

und das Gelenk mit

```
world.addJoint(vorgaenger.joint, segment, ['Rx', 'Ry'])
```

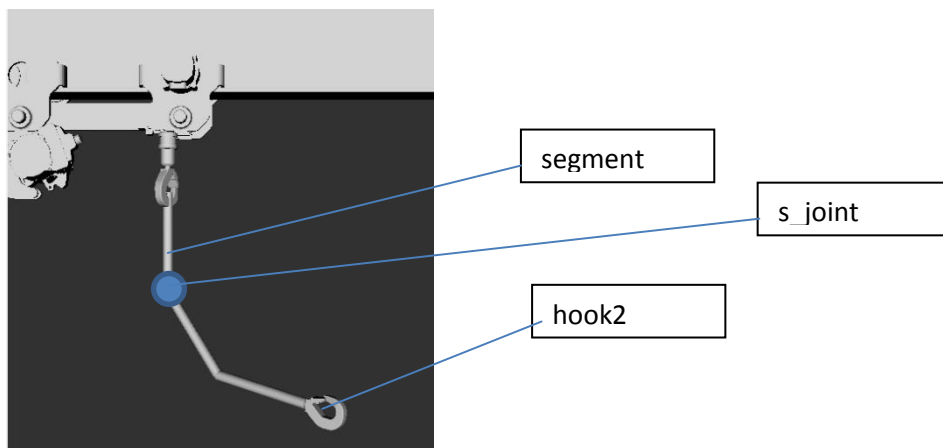
Visualisieren Sie die Seilsegmente.

```
world.addVisualization.Cylinder(segment.visual, radius=0.02, height=0.2)
```

Schließen Sie das Seil mit einem Körper Seilhaken ('*Seilhaken_01.stl*') ab.

Visualisieren Sie den Seilhaken.

2. Kapseln Sie das Hinzufügen eines Seilsegmentes als Funktion. Die Funktion erhält als Argumente das letzte Segment und die Länge des hinzuzufügenden Seilsegmentes. Prüfen Sie das Argument Länge auf Gültigkeit (Zahl größer Null).



Weitere Befehle

```
abfrage = raw_input()  
if abfrage:  
    ...  
print ...  
raise ValueError(...)
```

Stringformatierung 'hallo %s' % 'welt'

Strings zerschneiden

```
a = 'pi = 3.14'  
b = a.split('=')  
print b[0]  
print b[1]
```