

# Implementierung des Stick-Slip Effekts mit SCILAB/SCICOS(nicht fertig)

Dipl.-Ing. Klaus Weichinger Autor

snaky.1@gmx.at

20. September 2017

## Kurzfassung

Reibung zwischen gleitenden Flächen findet in vielen Anwendungen statt und sorgt durch den Stick-Slip Effekt zu unerwünschten Nebeneffekten wie z.B. eine bleibende Regelabweichung oder eine Oszillation um die Endlage bei einer Positionsregelung eines linearen Antriebs. Hier wird nun die Realisierung des Stick-Slip Modells mit SCILAB/SCICOS behandelt und anhand einiger Anwendungsbeispiele demonstriert.

## 1 Einführung

Die die Simulation realer Systeme am Computer ist heute ein nicht mehr wegzudenkendes Werkzeug bei der Analyse von Systemen oder beim Entwurf von Regelungen und Steuerungen. Um die Auswirkungen des Stick-Slip Effekts bereits in der Simulation berücksichtigen zu können wird in diesem Beitrag nun dessen Implementierung mit der Open-Source-Software SCILAB/SCICOS behandelt.

Der Stick-Slip Effekt tritt bei sich berührenden und gegeneinander bewegten Festkörpern an der Kontaktfläche statt und kann somit bei den unterschiedlichsten Maschinenelementen, Lagern und Führungen beobachtet werden.

In der Regelungstechnik wirkt sich der Stick-Slip Effekt besonders bei der Positionsregelung (Winkelregelung) von Linearantrieben (Rotationsantrieben) negativ aus und es kann zu bleibenden Regelfehlern in der Positionsendlage (Winkelendlage) bzw. zu einer Oszillation um die Endlage kommen.

Im Folgenden wird ausschließlich auf lineare Bewegungen Bezug genommen. Die erzielten Ergebnisse können aber genau so auf rotatorische Bewegungen angewandt werden.

## 2 Bewegungsgleichung und Stick-Slip Effekt

Es wird nun das System nach Abbildung 1 betrachtet, dafür ein mathematisches Modell hergeleitet und im nächsten Schritt ein SCICOS-Simulationsblock erstellt. Mit den Modellparametern nach Tabelle 1

ist das System paramitriert, wobei einige Parameter (z.B.  $sf$ ) auch eine Schaltfunktion für Modellannahmen übernehmen.

Symbol	Description
$m$	Masse
$sf$	Haftreibung (static friction)
$kf$	Gleitreibung (kinetic friction)
$ldc$	lineare Dämpfung (linear damping coefficient)
$sv$	Stribeckgeschwindigkeit
$F$	Antriebskraft (Eingang)
$x$	Massenposition (Ausgang)
$\dot{x}$	Massengeschwindigkeit (Ausgang)

Tabelle 1: Modellparameter, -größen

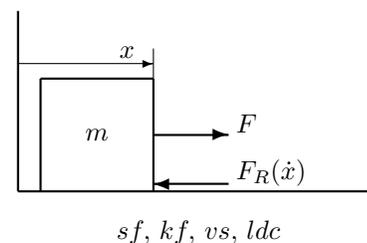


Abbildung 1: Stick-Slip Modell

Gilt  $sf = 0$ , ist der Stick-Slip Effekt deaktiviert und das Modell nach 1 wird verwendet.

$$m \ddot{x} = F - ldc \dot{x} \quad (1)$$

Gilt  $sf \neq 0$ , wird zwischen Haften und Gleiten der Kontaktfläche unterschieden. Findet ein Gleiten zwischen den beiden Körpern statt (d.h.  $\dot{x} \neq 0$ ), gilt das Modell nach 2. Haften die beiden Körper (d.h.  $\dot{x} = 0$  und  $|F| \leq sf$ ) wird das Modell nach 3 verwendet.

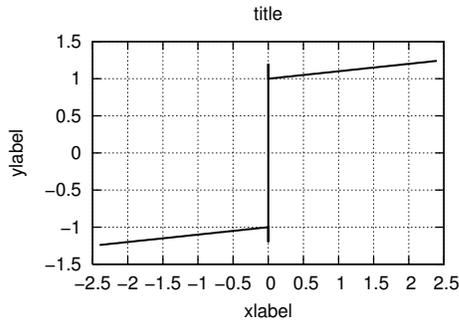
$$m \ddot{x} = F - F_R(\dot{x}) \quad (2)$$

$$\ddot{x} = \dot{x} = 0 \quad (3)$$

Für die Reibkraft  $F_R(\dot{x})$  werden nun abhängig von der Stribeckgeschwindigkeit folgende Reibmodelle verwendet:

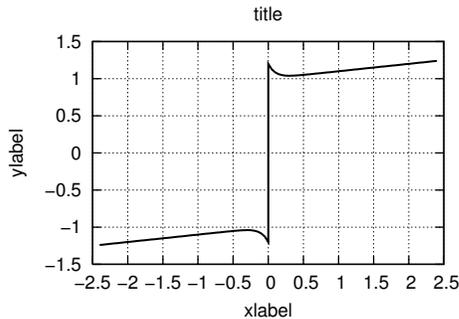
- $sv = 0$

$$F_R(\dot{x}) = ldc \dot{x} + kf \operatorname{sgn}(\dot{x}) \quad (4)$$



- $sv > 0$

$$F_R(\dot{x}) = ldc \dot{x} + \left[ kf + (sf - kf) e^{-\left| \frac{\dot{x}}{sv} \right|} \right] \operatorname{sgn}(\dot{x}) \quad (5)$$



### 3 Implementierung in Scilab/Scicos

Der Übergang zwischen Haften und Gleiten an der Kontaktfläche findet immer bei der Geschwindigkeit  $\dot{x} = 0$  statt. Das Problem ist nun, dass der Solver den Nulldurchgang von  $\dot{x} = 0$  durch die Zeitschrittweite überspringen kann und somit dieser Zeitpunkt nicht behandelt wird.

Mit Hilfe der Nulldurchgangserkennung in SCICOS kann dieses Problem nun gelöst werden. Dabei werden sogenannte Surfaces – das sind Funktionen deren Nulldurchgang erkannt werden soll – berechnet. Findet bei dieser Surface ein Vorzeichenwechsel statt (d.h.

ein Nulldurchgang hat stattgefunden) wird der Solver zurückgesetzt und versucht, den Nullpunkt mit den Toleranzangaben der Simulation zu erreichen. Ist nun dieser Punkt erreicht, wird ein Nulldurchgangsereignis erkannt und die zugehörige Funktion ausgeführt, die in diesem Fall die Systemumschaltung durchführt.

Listing 1: Computational Function

```

1 function block=cfStickSlipSys(block, flag)
2 m=block.rpar(1);
3 sv=block.rpar(2);
4 sf=block.rpar(3);
5 kf=block.rpar(4);
6 ldc=block.rpar(5);
7
8 phase=phase_simulation();
9
10 if flag==0 then
11 // Aktualisierung des Zustands
12 x=block.x(1);
13 v=block.x(2);
14 F=block.inptr(1);
15 stickslip=block.x(3);
16
17 if (sf==0) then
18 // System ohne Reibmodell aber mit
19 // linearer Dämpfung
20 block.xd(1)=v;
21 block.xd(2)=1/m*(F-ldc*v);
22 else
23 if (stickslip==0) then
24 // System haftet
25 block.xd(1)=0;
26 block.xd(2)=0;
27 else
28 // System gleitet
29 // Reibmodelle
30 if (sv>0) then
31 Fr=ldc*v+(kf+(sf-kf))*..
32 exp(-(abs(v)/sv))*sign(v);
33 else
34 Fr=ldc*v+kf*sign(v);
35 end
36 block.xd(1)=v;
37 block.xd(2)=1/m*(F-Fr);
38 end
39 end
40 block.xd(3)=0;
41 end
42
43 if flag==1 then
44 // Berechnung der Ausgänge
45 x=block.x(1);
46 v=block.x(2);
47 block.outptr(1)=block.x(1);
48 block.outptr(2)=block.x(2);
49 if (v==0) then
50 block.outptr(3)=0;
51 else
52 block.outptr(3)=1;
53 end
54 end
55
56 if (flag==2)&(block.nevprt==-1) then
57 // Nulldurchgang hat stattgefunden
58 if (block.jroot(1)<>0) then
59 // Nulldurchgang der Geschwindigkeit
60 F=block.inptr(1);
61 if (abs(F)<=abs(sf)) then
62 block.x(2)=0;
63 block.x(3)=0;
64 end
65 end
66 if (block.jroot(2)>0) then
67 // Nulldurchgang: Antriebskraft

```

```

68 // größer als die Haftreibung
69 block.x(3)=1;
70 end
71 if (block.jroot(3)<0) then
72 // Nulldurchgang: Antriebskraft
73 // größer als die Haftreibung
74 block.x(3)=1;
75 end
76 end
77
78 if flag==4 then
79 // Initialisierung
80 block.x(3)=0;
81 if (block.x(2)<>0) then
82 block.x(3)=1;
83 end
84 end
85
86 if flag==5 then
87 // Simulationsende
88 end
89
90 if flag==9 then
91 // Berechnung der Nulldurchgänge
92 // (Surfaces)
93 F=block.inptr(1);
94 v=block.x(2);
95 if (block.x(3)==0) then
96 block.g(2)=F-sf;
97 block.g(3)=F+sf;
98 block.g(3)=1;
99 else
100 block.g(1)=v;
101 end
102 end
103 endfunction

```

Das Listing 1 stellt nun die Implementierung der benötigten Simulationsfunktion dar. Neben dieser Funktion (auch Computational Function genannt) ist auch eine Funktion, die den graphischen Block definiert (siehe `StickSlipSys.sci`), nötig. Auf die Implementierung wird hier nicht mehr weiter eingegangen. Hingegen wird noch dessen Verwendung beschrieben.

#### 4 Verwendung

Der Stick-Slip Simulationsblock besteht nun aus den Dateien

- `StickSlipSys.sci`
- `cfStickSlipSys.sci`

die vor der Verwendung in SCICOS mit den Befehlen

```

exec StickSlipSys.sci
exec cfStickSlipSys.sci

```

in SCILAB geladen werden. Der Block kann nun in SCICOS mit dem Menüeintrag `Edit->Add new block` und der Eingabe von `StickSlipSys` eingefügt werden. Öffnet man den Block nun mit `Open/Set`, wird eine Auflistung von möglichen Aktionen angeführt (siehe Abbildung 2).

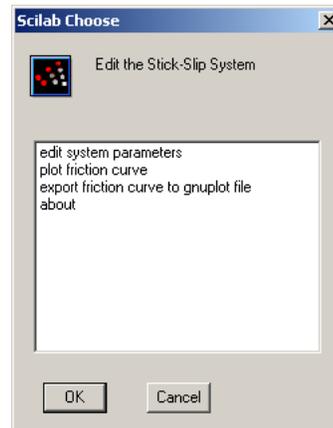


Abbildung 2: Aktionsauflistung

In dieser Auflistung wird nun die gewünschte Aktion ausgewählt. Die erste Aktion ermöglicht die Änderung der Modellparameter (siehe Abbildung 3),

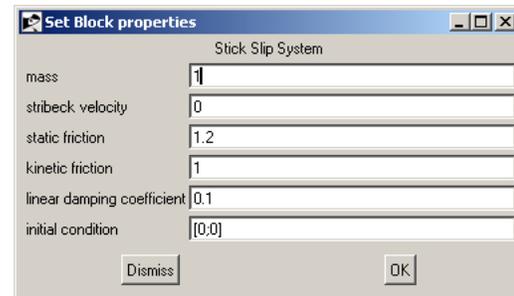


Abbildung 3: Modellparameter Dialog

wobei mit der zweiten Aktion die eingestellte Reibkennlinie direkt dargestellt werden (vgl. Abbildung 4) kann.

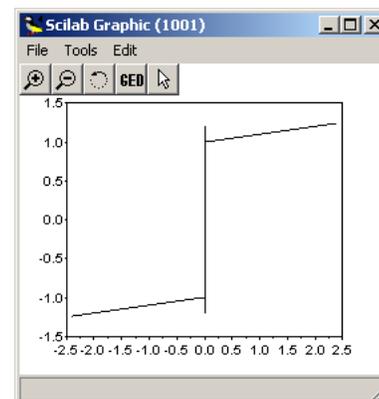


Abbildung 4: Darstellung der Reibkennlinie

Für Dokumentationszwecke wurde auch eine Exportfunktion für GNUPLOT realisiert, wobei man dabei die Anzahl der Stützstellen des Plots und den Exportnamen angeben kann. Beim GNUPLOT-Export werden zwei Dateien angelegt. Eine \*.gp Datei, die die GNUPLOT-Anweisungen zur Generierung eines EPS-Bildes beinhaltet und eine \*.dat Datei, in der die Punkte der Reibkennlinie abgelegt sind. Mit dieser Datei ist es auch möglich mit anderen Programmen die Reibkennlinie darstellen zu lassen. Die Exporteinstellungen können über den angezeigten Dialog nach Abbildung 5 eingegeben werden.

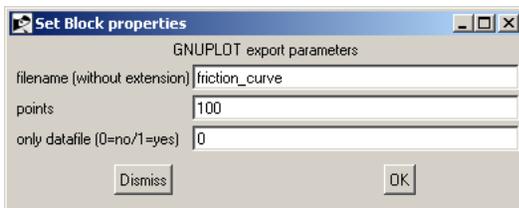


Abbildung 5: GNUPLOT Exporteinstellungen

### Hinweis 1

Wird die \*.dat Datei in anderen Programmen verwendet (z.B. den bekannten Tabellenkalkulationsprogrammen) ist zu beachten, dass diese meist ein Komma statt einem Punkt für die Zahlendarstellung benötigen. Eine einfache Lösungsmöglichkeit wäre, mit einem Texteditor alle Punkte in der \*.dat Datei durch Kommas ersetzen zu lassen.

### Hinweis 2

Der Punkt zwischen Gleiten und Haften kann oftmals zu längeren Simulationsdauern führen. Um die Simulationsgeschwindigkeit zu erhöhen, empfiehlt es sich die Toleranzen der Simulation zu verändern wobei hier die Einträge

- Integrator absolute tolerance
- Integrator relative tolerance

ausschlaggebend sind (vgl. Abbildung 6).

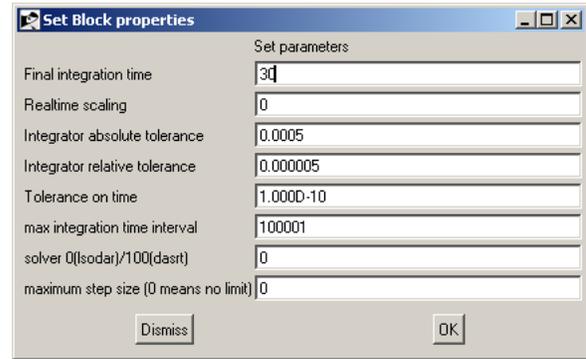


Abbildung 6: Simulationseinstellungen

## 5 Anwendungsbeispiele

In diesem Abschnitt wird nun das erstellte Simulationsmodell anhand einiger Beispiele getestet. Bereits das erste Beispiel stellt ein bekanntes Experiment dar. Für die folgenden Versuche werden die Parameter  $m = 1 \text{ kg}$ ,  $sv = 0.01 \frac{\text{m}}{\text{s}}$ ,  $sf = 1.2 \text{ N}$ ,  $kf = 1 \text{ N}$  und  $ld = 0.1 \frac{\text{N}\cdot\text{s}}{\text{m}}$  verwendet.

### 5.1 Beispiel - Experiment

In diesem Beispiel wird eine Block mit der Masse  $m$  mit einer Feder gezogen (siehe Abbildung 7). Die

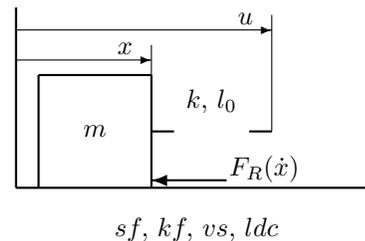


Abbildung 7: Stick-Slip Experiment

Bewegungsgleichung für dieses Experiment ist nun mit

$$m \ddot{x} = (u - x - l_0) k - F_R(\dot{x}) \quad (6)$$

für den gleitenden Block und mit

$$\dot{x} = \ddot{x} = 0 \quad (7)$$

für den haftenden Block gegeben.

Dieses Experiment wurde nun mit Hilfe des Stick-Slip Simulationsmodells der SCICOS-Simulation nach Abbildung 8 implementiert

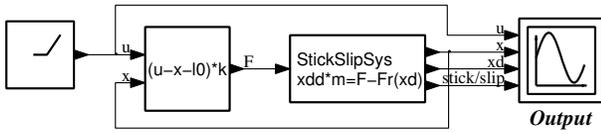


Abbildung 8: SCICOS Simulation `experiment.cos`

und liefert die in der Abbildung 9 dargestellten Ergebnisse.

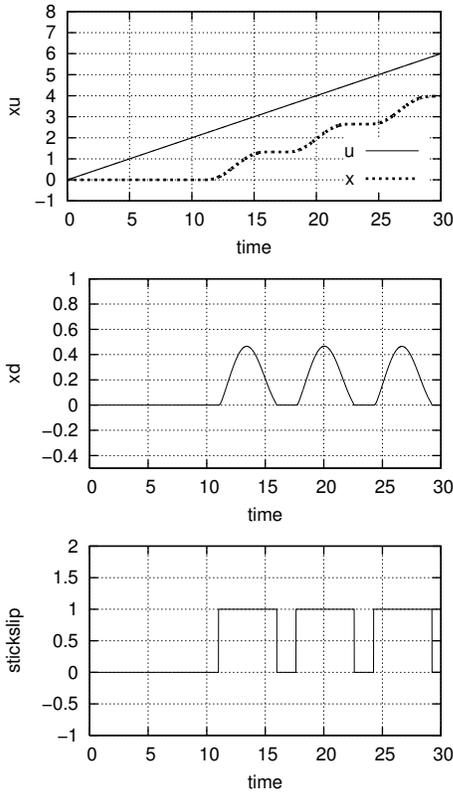


Abbildung 9: Experimentergebnisse

## 6 Zusammenfassung und Folgerungen

Todo... (and never done)

Gewidmet unserem Sohn 2012-2015.

### Acknowledgements

Todo... (and never done)