

Modellierung dynamischer Systeme

Abgabe der Praktikumsaufgabe 3

Maria Lüdemann und Birger Kamp

May 11, 2016

Contents

1	Teilaufgabe Schiefer Flipper	2
1.1	Gegebene Formeln und Konstanten	2
1.2	Funktionen	3
1.2.1	Init	3
1.2.2	Acc	3
1.2.3	Wand1Refl	3
1.2.4	Wand2Refl	3
1.2.5	Wand3Refl	4
1.2.6	Wand1Kontakt	4
1.2.7	Wand2Kontakt	4
1.2.8	Wand3Kontakt	4
1.2.9	HndKontakt	4
1.3	Stateflow-Modell	4

1 Teilaufgabe Schiefer Flipper

Im Vorlauf der Praktikumsaufgabe 3 haben wir uns dafür entschieden den schiefen Flipper zu simulieren.

Dafür soll die Bewegung einer Kugel auf einer geneigten Ebene simuliert werden. Die Ebene ist von drei Wänden begrenzt und hat in der Mitte ein zylindrisches Hindernis. Das Abprallen der Kugel von den Wänden und dem Hindernis verändert die Richtung ihrer Bewegung, dabei wird auch ihre Beschleunigung und somit Geschwindigkeit simuliert.

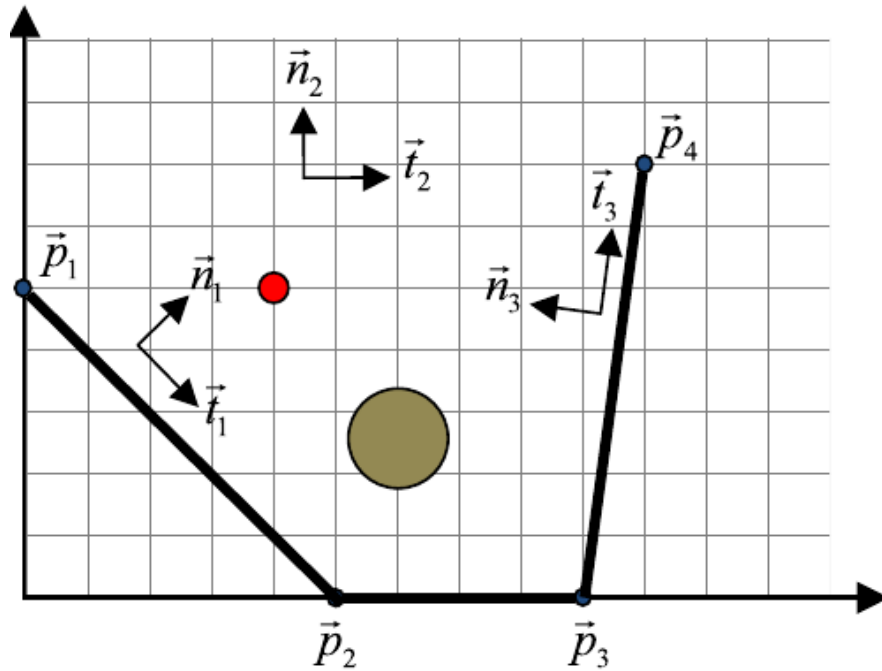


Figure 1: Skizze zum schiefen Flipper

1.1 Gegebene Formeln und Konstanten

Die begrenzenden Wände, durch Vektoren dargestellt definieren sich durch:

$$\vec{p}_1 = (0, 5)^T \quad \vec{p}_2 = (5, 0)^T \quad \vec{p}_3 = (9, 0)^T \quad \vec{p}_4 = (10, 7)^T \quad (1)$$

Die Position und der Umfang des zylindrischen Hindernisses entsprechen:

$$\vec{p}_Z = (6, 2.5)^T \quad RHnd = 0.8 \quad (2)$$

Startposition der Kugel

$$\vec{x}a_0 = (4, 5)^T \quad (3)$$

Startgeschwindigkeit der Kugel

$$\vec{v}_{10} = (0, 0)^T \quad (4)$$

Radius der Kugel

$$R = 0.25 \quad (5)$$

1.2 Funktionen

Im Folgenden werden die verwendeten MatLab-Funktionen dokumentiert.

1.2.1 Init

```
1 function Init
2
3 % Hindernis und Pfoften fuer Waende
4 Hnd = [6;2.5];
5 p1 = [0;5];
6 p2 = [5;0];
7 p3 = [9;0];
8 p4 = [10;7];
9
10 % Kugel
11 x1 = [4;5];
12 v1 = [0;0];
13
14 % Normale und Tangentiale der Waende
15 t1 = (p2-p1) / norm(p2-p1);
16 n1 = [-t1(2,1); t1(1,1)];
17 t2 = (p3-p2) / norm(p3-p2);
18 n2 = [-t2(2,1) ; t2(1,1)];
19 t3 = (p4-p3) / norm(p4-p3);
20 n3 = [-t3(2,1) ; t3(1,1)];
```

1.2.2 Acc

```
1 function [ax,ay] = Acc
2 ax = 0;
3 ay = -1;
```

1.2.3 Wand1Refl

```
1 function Wand1Refl()
2
3 vt = v1' * t1;
4 vn = v1' * n1;
5
6 % Reflektierte Geschwindigkeit (mit Vektorisierung der Geschwindigkeit)
7 v1 = vt*t1 + (-vn*n1);
```

1.2.4 Wand2Refl

```

1 function Wand2Refl()
2
3 vt = v1' * t2;
4 vn = v1' * n2;
5
6 % Reflektierte Geschwindigkeit (mit Vektorisierung der Geschwindigkeit)
7 v1 = vt*t2 + (-vn*n2);

```

1.2.5 Wand3Refl

```

1 function Wand3Refl()
2
3 vt = v1' * t3;
4 vn = v1' * n3;
5
6 % Reflektierte Geschwindigkeit (mit Vektorisierung der Geschwindigkeit)
7 v1 = vt*t3 + (-vn*n3);

```

1.2.6 Wand1Kontakt

```

1 function kontakt = Wand1Kontakt
2 kontakt = (((x1-p1)' * n1) <= R);

```

1.2.7 Wand2Kontakt

```

1 function kontakt = Wand2Kontakt
2 kontakt = ((x1-p2)' * n2) <= R;

```

1.2.8 Wand3Kontakt

```

1 function kontakt = Wand3Kontakt
2 kontakt = ((x1-p3)' * n3) <= R;

```

1.2.9 HndKontakt

```

1 function kontakt = HndKontakt
2 kontakt = norm(x1 - Hnd) <= RHnd + R;

```

1.3 Stateflow-Modell

Das folgende Modell veranschaulicht das Stateflow-Modell, das die Simulation durchführt.

Es gibt in diesem Modell nur einen Zustand, in dem erfährt die Kugel ihre Beschleunigung in Richtung *unten*. Anschließend wird für jede Wand/Hindernis geprüft, ob die Kugel in Kontakt damit gekommen ist. Falls ja, dann wird berechnet in welche Richtung die Kugel umgelenkt wird.

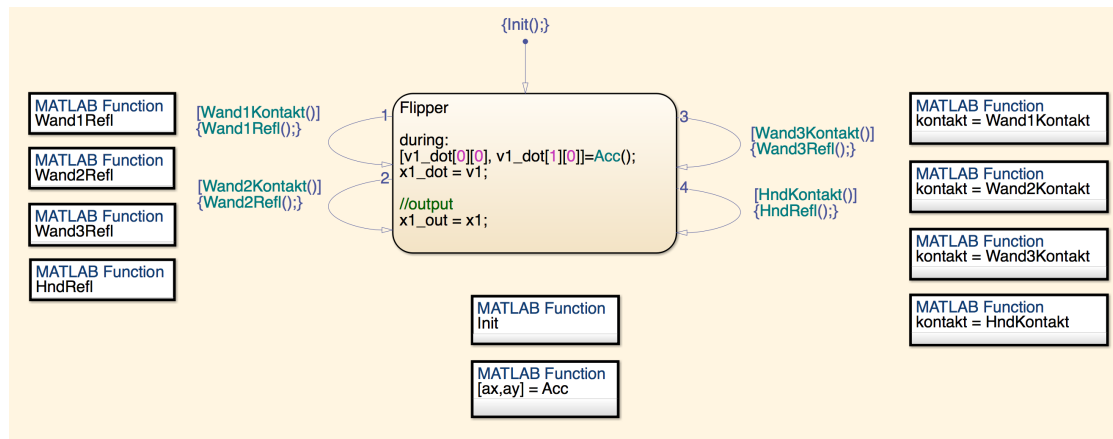


Figure 2: Stateflow Modell für den Flipper