

Projektarbeit: Doppelt inverses Pendel

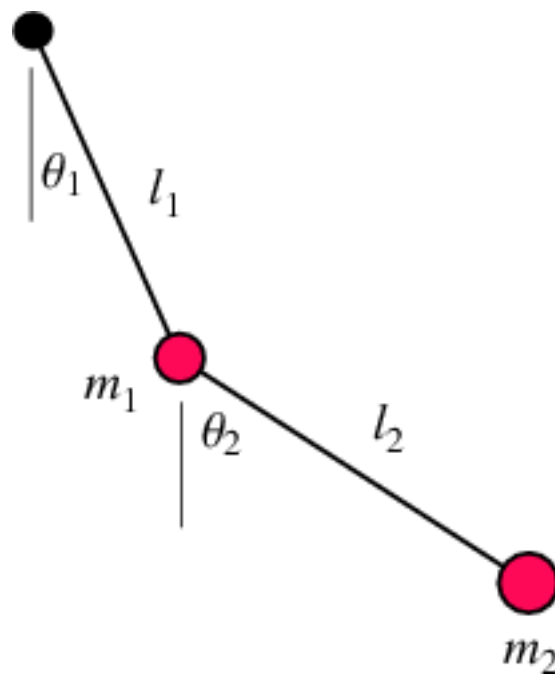


Bild: Wolfram Research

Michael Höhn
hoehnmic@students.zhaw.ch

Stefan Hauenstein
hauenste@students.zhaw.ch

12. Juni 2012

Inhaltsverzeichnis

1	Projektbeschreibung	4
1.1	Danksagung	4
1.2	Aufgabenstellung durch den Experten	4
1.3	Motivation	4
1.4	Iterationsplan	5
1.5	Beteiligte Personen	5
2	Anforderungsanalyse	6
2.1	Zielbestimmung	6
2.2	Rahmenbedingungen und Einschränkungen	6
2.3	Produktfunktionen	6
3	Umsetzung	7
3.1	Funktionsprinzip Doppelpendel	7
3.2	Verwendete Technologien	7
3.2.1	ActionScript 3 mit Flex SDK	7
3.2.2	Entwicklungsumgebung FDT	8
3.3	Numerisches Verfahren	8
3.3.1	Runge-Kutta Verfahren 4. Ordnung (RK4)	8
4	Physikalische und Mathematische Zusammenhänge	10
4.1	Physikalische Eigenschaften	10
4.1.1	Kinematik des Doppelpendels	10
4.1.2	Kräfte im Doppelpendel	11
4.1.2.1	Oberes Pendel	12
4.1.2.2	Unteres Pendel	13
4.1.3	Umformen der Gleichungen	13
4.2	Numerische Lösung	15
4.2.1	Differentialgleichung für Runge-Kutta	15
5	Software	16
5.1	Grafische Oberfläche	16
5.2	MVC-Pattern	17

5.2.1	Model-Controller	17
5.2.2	View	18
5.3	Bootstrapper	18
5.4	Solver Klassen (Runge-Kutta)	18
5.5	Konfiguration des Systems	18
5.5.1	Konfigurationsdateien	18
A	Projektplan	23

1 Projektbeschreibung

1.1 Danksagung

Wir möchten an dieser Stelle Herrn Georg Brügger und Herrn Lukas Eppler danken, ohne ihre wertvolle Unterstützung und Anregungen wäre dieses Software Projekt nur schwer umsetzbar gewesen.

1.2 Aufgabenstellung durch den Experten

Das Schwingverhalten eines idealen „Doppelt Inversen Pendel“ soll untersucht und durch geeignete Physikalische und Mathematische Beziehungen nach gebildet werden. Physikalische Effekte wie Reibung und Luftwiderstand werden vernachlässigt, Abmessungen und Materialien müssen als Eingabegrößen variabel gestaltet werden.

Der Vorgang wird in „Echtzeit“ am Bildschirm des PC's Graphisch dargestellt visualisiert.

Zusätzliche Angaben des Verhaltens können auf der gleichen Bildschirmseite eingeblendet werden.

1.3 Motivation

Physikalische Eigenschaften mit einer Objekt Orientierten Programmiersprache umzusetzen reizte uns schon immer. Die Aufgabenstellung ermöglichte uns das theoretisch gelernte aus dem Studienfach Numerik in Verbindung mit realen Physikalischen Grundlagen in einer praktischen Anwendung umzusetzen.

Auch ermöglichte es die algebraischen Zusammenhänge aus vergangenen Vorlesungen besser zu verstehen und Lücken aufzufüllen.

1.4 Iterationsplan

Iteration	Termin	Task
1	Mi 04.04.2012	<ul style="list-style-type: none">• Physikalische Eigenschaften des Doppelpendels aufstellen• Evaluierung der Programmiersprache• Gleichungssystem aufstellen
2	Fr 27.04.2012	<ul style="list-style-type: none">• Grafische Benutzeroberfläche• Programmierung des ersten Pendels• Programmierung des zweiten Pendels• Runge Kutta definieren
3	Mi 23.05.2012	<ul style="list-style-type: none">• Pendel einstellen• Projektdokumentation• Präsentation
4	Präsentationstermin	<ul style="list-style-type: none">• Präsentation• Projektdokumentation

1.5 Beteiligte Personen

Funktion	Name
Projekt Team	Stefan Hauenstein Michael Höhn
Experte	Georg Brügger
Scrum Master	Lukas Eppler

2 Anforderungsanalyse

2.1 Zielbestimmung

Es soll ein Programm entwickelt werden das die physikalischen und chaotischen Eigenschaften des inversen Doppelpendels grafisch darstellt.

Die einzelnen Segmente des Pendels sollen durch Benutzereingaben über Tastatur und/oder Maus einstellbar sein.

2.2 Rahmenbedingungen und Einschränkungen

- Die Umgebung des Doppelpendels soll als ideal gelten.
- Das Ausgangsmaterial wird als Aluminium Stab definiert (Dichte $2,7[g/cm^3]$)
- Die Gravitation wird mit $9.81[m/s^2]$ definiert
- Wegen der chaotischen Eigenschaft des Pendels können keine aussagenkräftigen Tests implementiert werden.

2.3 Produktfunktionen

- Der Benutzer kann eine Konfiguration als XML-File mit der Erweiterung IDP einlesen.
- Der Benutzer kann die einzelnen Glieder des Pendels mit der Maus positionieren.
- Während das Pendel in Bewegung ist werden Winkel und Geschwindigkeit angezeigt.
- Der Benutzer kann den Ablauf starten, jederzeit stoppen und zurückstellen.

3 Umsetzung

3.1 Funktionsprinzip Doppelpendel

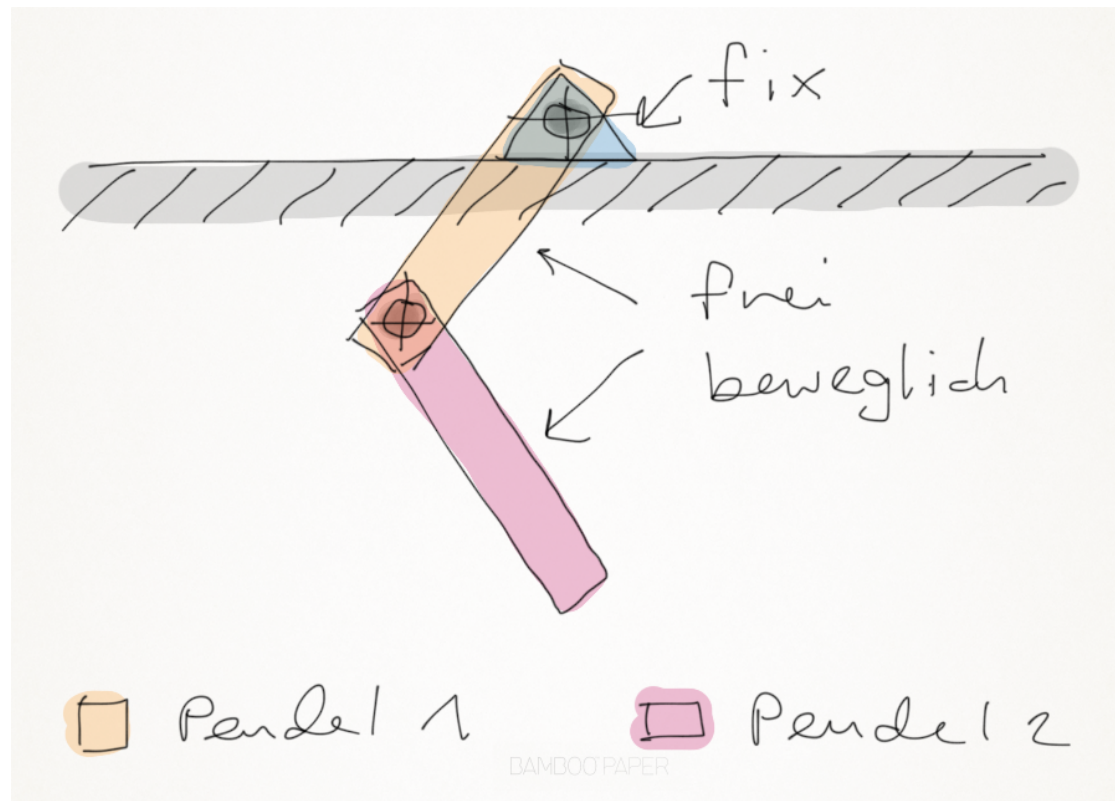


Abbildung 1: Funktionsweise des Doppelpendels

3.2 Verwendete Technologien

3.2.1 ActionScript 3 mit Flex SDK

Das Programm wird mittels der Objektorientierten Programmiersprache ActionScript 3 (AS3) und dem Adobe Flex SDK Framework[Adobe, 2012] implementiert. Es ist dank dem Adobe Flashplayer auf fast jedem Betriebssystem und Browser lauffähig.

Eine Implementierung des Doppelpendels mittels AS3 und dem Flex SDK ist empfehlenswert, da das Flex Framework speziell für Grafische Applikationen mit Animatio-

nen ausgelegt ist und es dadurch zu einer grossen Zeiteinsparung bei der Entwicklung kommt.

3.2.2 Entwicklungsumgebung FDT

Die FDT (Flash Development Tool)[Powerflasher, 2012] Entwicklungsumgebung benutzt Eclipse als Basisframework. FDT stellt für die Entwicklung von AS3 und Flex-Projekten nützliche Funktionen wie Fehlererkennung, Debugger und Autocomplete zur Verfügung. Ein AS3 Projekt, das mit dem Flex SDK erstellt wurde kann aber auch ohne FDT verändert und kompiliert werden.

3.3 Numerisches Verfahren

Um die Fehlerkomulierten Abweichungen der Berechnung auf möglichst lange Zeit klein zu halten wird das Numerische Verfahren von Runge-Kutta in der 4. Ordnung verwendet. Der Rechenaufwand für dieses Verfahren ist höher als bei anderen Verfahren wie z.B. dem Euler-Verfahren. Seine Genauigkeit auf grosse Zeitintervalle gleichen diesen Mehraufwand wieder aus.

3.3.1 Runge-Kutta Verfahren 4. Ordnung (RK4)

Das klassische Runge-Kutta-Verfahren (nach Carl Runge und Martin Wilhelm Kutta) ist ein spezielles explizites 4-stufiges Runge-Kutta-Verfahren zur numerischen Lösung von Anfangswertproblemen. Runge hat als erster (1895) ein mehrstufiges Verfahren angegeben und Kutta die allgemeine Form expliziter s-stufiger Verfahren.

Das klassische Runge-Kutta-Verfahren verwendet den Ansatz, Ableitungen durch Differenzenquotienten zu approximieren. Die dabei bei nichtlinearen Funktionen notwendigerweise auftretenden Fehler können durch geeignete Kombinationen verschiedener Differenzquotienten reduziert werden. Das RK4 verfahren ist eine solche Kombination, die Diskretisierungsfehler bis zur dritten Ableitung kompensiert.

$$y(t)' = f(t, y(t)) \quad \text{mit} \quad y(0) = y_0$$

$$y^{k+1} = y^k + h \left(\frac{1}{6}k_1 + \frac{2}{6}k_2 + \frac{2}{6}k_3 + \frac{1}{6}k_4 \right) \quad (3.3.1)$$

$$\text{mit} \quad k_1 = f(t_k, y^k)$$

$$k_2 = f\left(t_k + \frac{h}{2}, y^k + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(t_k + \frac{h}{2}, y^k + \frac{h}{2}k_2\right)$$

$$k_4 = f(t_k + h, y^k + hk_3)$$

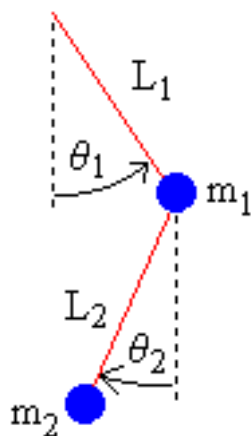
4 Physikalische und Mathematische Zusammenhänge

4.1 Physikalische Eigenschaften

Um die Gleichungen des Ausgangsproblems zu definieren, muss man sich erst mit den physikalischen Eigenschaften eines Pendels beschäftigen. Kräfte welche aus der Bewegung entstehen kombinieren sich mit Statischen. Kräfte des ersten Pendels beeinflussen die des Zweiten. Im Folgenden werden die Kinematik und die Kräfte des Doppelpendels zum gesuchten Ausgangsproblem kombiniert.

Auf Basis der Webseiten [Neumann, 2004a], [Kramann, 2011] und [Weisstein, 2007] konnten die Kinematik und Kräfte des Doppelpendels zum Ausgangsproblem kombiniert werden.

4.1.1 Kinematik des Doppelpendels



x = Horizontale Position des Massenschwerpunkts
 y = Vertikale Position des Massenschwerpunkts
 θ = Winkel
 L = Pendellänge

Abbildung 2: Doppelpendel [Neumann, 2004a]

Ausgangslage für die Berechnung ist das obere Pendel. Das erste Pendel wird als Subskript 1 definiert mit x_1 und y_1 . Äquivalent dazu wird das zweite Pendel als Subskript 2 definiert mit x_2 und y_2 .

Als erstes werden die trigonometrischen Gleichungen aufgestellt:

$$x_1 = L_1 \sin \theta_1 \quad (4.1.1)$$

$$y_1 = -L_1 \cos \theta_1 \quad (4.1.2)$$

$$x_2 = x_1 + L_2 \sin \theta_2 \quad (4.1.3)$$

$$y_2 = y_1 - L_2 \cos \theta_2 \quad (4.1.4)$$

Danach wird aus den Gleichungen (4.1.1 - 4.1.4) die erste und zweite Ableitung gebildet.

Erste Ableitung (Geschwindigkeit):

$$x'_1 = \theta'_1 L_1 \cos \theta_1 \quad (4.1.5)$$

$$y'_1 = \theta'_1 L_1 \sin \theta_1 \quad (4.1.6)$$

$$x'_2 = x'_1 + \theta'_2 L_2 \cos \theta_2 \quad (4.1.7)$$

$$y'_2 = y'_1 + \theta'_2 L_2 \sin \theta_2 \quad (4.1.8)$$

Zweite Ableitung (Beschleunigung):

$$x''_1 = -\theta'^2_1 L_1 \sin \theta_1 + \theta''_1 L_1 \cos \theta_1 \quad (4.1.9)$$

$$y''_1 = \theta'^2_1 L_1 \cos \theta_1 + \theta''_1 L_1 \sin \theta_1 \quad (4.1.10)$$

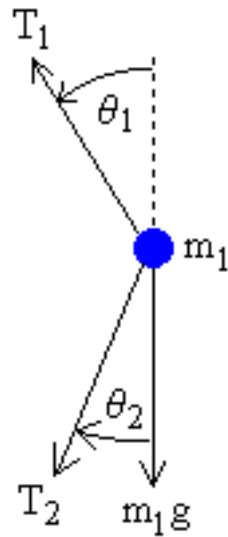
$$x''_2 = x''_1 - \theta'^2_2 L_2 \sin \theta_2 + \theta''_2 L_2 \cos \theta_2 \quad (4.1.11)$$

$$y''_2 = y''_1 + \theta'^2_2 L_2 \cos \theta_2 + \theta''_2 L_2 \sin \theta_2 \quad (4.1.12)$$

4.1.2 Kräfte im Doppelpendel

Die Kräfte werden pro Pendel berechnet und in einzelne Gleichungen gesetzt. Auf Grundlage des zweiten newtonschen Gesetzes ($F = m \cdot a$) können folgende Gleichungen für das Doppelpendel erstellt werden:

4.1.2.1 Oberes Pendel



T_1 = Kraft im oberen Pendel
 m_1 = Masse des oberen Pendels
 g = Gravitations Konstante

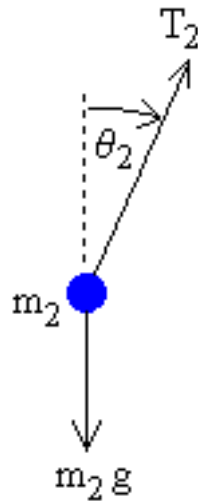
Abbildung 3: Kräfte im oberen Pendel [Neumann, 2004a]

Anhand von Abbildung 3 können die folgenden Gleichungen aufgestellt werden.

$$m_1 x_1'' = -T_1 \sin \theta_1 + T_2 \sin \theta_2 \quad (4.1.13)$$

$$m_1 y_1'' = T_1 \cos \theta_1 + T_2 \cos \theta_2 - m_1 g \quad (4.1.14)$$

4.1.2.2 Unteres Pendel



T_2 = Kraft im unteren Pendel
 m_2 = Masse des unteren Pendels
 g = Gravitations Konstante

Abbildung 4: Kräfte im unteren Pendel
 [Neumann, 2004a]

Anhand von Abbildung 4 können die folgenden Gleichungen aufgestellt werden.

$$m_2 x_2'' = -T_2 \sin \theta_2 \quad (4.1.15)$$

$$m_2 y_2'' = T_2 \cos \theta_2 - m_2 g \quad (4.1.16)$$

4.1.3 Umformen der Gleichungen

Um zu der endgültigen Differenziellen Gleichung für die Berechnung zu gelangen müssen die einzelnen Gleichungen algebraisch umgeformt werden. Angefangen bei den Kräftegleichungen. Gleichung (4.1.15) und (4.1.16) werden nach $T_2 \sin \theta_2$ bzw. nach $T_2 \cos \theta_2$ aufgelöst und in die Gleichungen (4.1.13) und (4.1.14) eingesetzt.

$$m_1 x_1'' = -T_1 \sin \theta_1 - m_2 x_2'' \quad (4.1.17)$$

$$m_1 y_1'' = T_1 \cos \theta_1 - m_2 y_2'' - m_2 g - m_1 g \quad (4.1.18)$$

Gleichung (4.1.17) wird mit $\cos \theta_1$ und Gleichung (4.1.18) mit $\sin \theta_1$ erweitert und nach $T_1 \sin \theta_1 \cos \theta_1$ aufgelöst.

$$T_1 \sin \theta_1 \cos \theta_1 = -\cos \theta_1 (m_1 x_1'' + m_2 x_2'') \quad (4.1.19)$$

$$T_1 \sin \theta_1 \cos \theta_1 = \sin \theta_1 (m_1 y_1'' + m_2 y_2'' + m_2 g + m_1 g) \quad (4.1.20)$$

Daraus ergibt sich die erste Teil der Bewegungsgleichung.

$$\sin \theta_1 (m_1 y_1'' + m_2 y_2'' + m_2 g + m_1 g) = -\cos \theta_1 (m_1 x_1'' + m_2 x_2'') \quad (4.1.21)$$

Um den zweiten Teil der Gleichung zu erhalten wird die Gleichung (4.1.15) mit $\cos \theta_2$ und die (4.1.16) mit $\sin \theta_2$ erweitert.

$$T_2 \sin \theta_2 \cos \theta_2 = -\cos \theta_2 (m_2 x_2'') \quad (4.1.22)$$

$$T_2 \sin \theta_2 \cos \theta_2 = \sin \theta_2 (m_2 y_2'' + m_2 g) \quad (4.1.23)$$

Der zweite Teil der Gleichung ergibt somit

$$\sin \theta_2 (m_2 y_2'' + m_2 g) = -\cos \theta_2 (m_2 x_2'') \quad (4.1.24)$$

x_2'' , x_1'' , y_1'' und y_2'' werden durch die Gleichungen (4.1.9 – 4.1.12) ersetzt und mit Hilfe des Lagrange-Formalismus nach θ_1'' und θ_2'' aufgelöst.

Die endgültige Bewegungsgleichung für das Doppelpendel lautet somit:

$$\theta_1'' = \frac{-g(2m_1 + m_2) \sin \theta_1 - m_2 g \sin(\theta_1 - 2\theta_2) - 2 \sin(\theta_1 - \theta_2) m_2 (\theta_2'^2 L_2 + \theta_1'^2 L_1 \cos(\theta_1 - \theta_2))}{L_1(2m_1 + m_2 - m_2 \cos(2\theta_1 - 2\theta_2))} \quad (4.1.25)$$

$$\theta_2'' = \frac{2 \sin(\theta_1 - \theta_2) (\theta_1'^2 L_1 (m_1 + m_2) + g(m_1 + m_2) \cos \theta_1 + \theta_2'^2 L_2 m_2 \cos(\theta_1 - \theta_2))}{L_2(2m_1 + m_2 - m_2 \cos(2\theta_1 - 2\theta_2))} \quad (4.1.26)$$

4.2 Numerische Lösung

4.2.1 Differentialgleichung für Runge-Kutta

Die Bewegungsgleichung (4.1.25 und 4.1.26) lässt sich aber noch nicht mittels des Runge-Kutta Algorithmus numerisch annähern. Um dies zu ermöglichen muss bedacht werden, dass die erste Ableitung von θ die Winkelgeschwindigkeit ω ist.

Daraus ergibt sich:

$$\theta'_1 = \omega_1 \quad \text{und} \quad \theta'_2 = \omega_2$$

damit erhält man für den Runge-Kutta Algorithmus 3.3.1 folgende Schlussgleichungen:

$$\theta'_1 = \omega_1 \tag{4.2.1}$$

$$\theta'_2 = \omega_2 \tag{4.2.2}$$

$$\omega'_1 = \frac{-g(2m_1 + m_2) \sin \theta_1 - m_2 g \sin(\theta_1 - 2\theta_2) - 2 \sin(\theta_1 - \theta_2) m_2 (\omega_2^2 L_2 + \omega_1^2 L_1 \cos(\theta_1 - \theta_2))}{L_1(2m_1 + m_2 - m_2 \cos(2\theta_1 - 2\theta_2))} \tag{4.2.3}$$

$$\omega'_2 = \frac{2 \sin(\theta_1 - \theta_2) (\omega_1^2 L_1 (m_1 + m_2) + g(m_1 + m_2) \cos \theta_1 + \omega_2^2 L_2 m_2 \cos(\theta_1 - \theta_2))}{L_2(2m_1 + m_2 - m_2 \cos(2\theta_1 - 2\theta_2))} \tag{4.2.4}$$

5 Software

5.1 Grafische Oberfläche

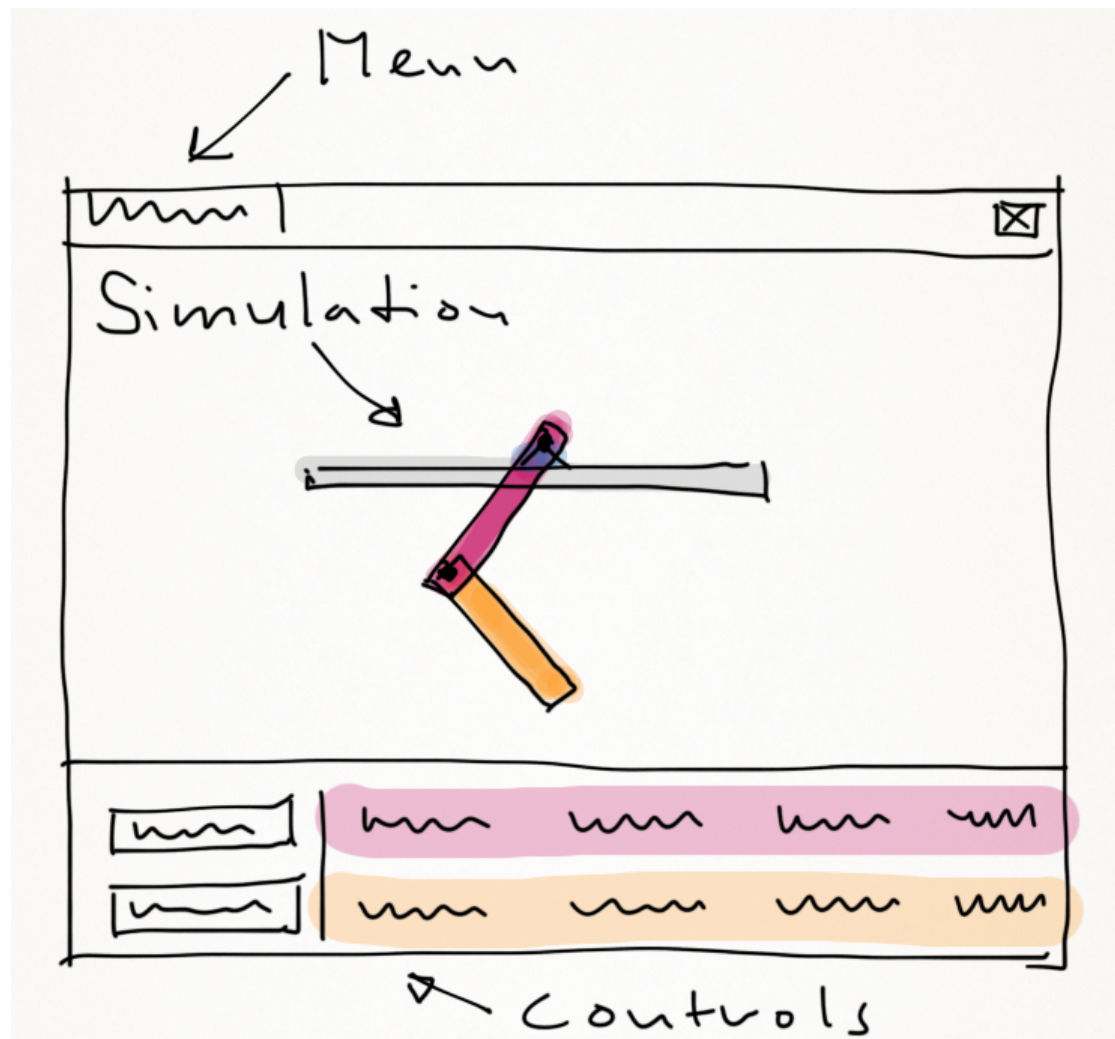


Abbildung 5: Entwurf Grafische Oberfläche

Die Grafische Oberfläche wird als skalierbares Modell aufgebaut. Der Vorteil in diesem Aufbau besteht darin, dass die Applikation auf jeder Bildschirmauflösung optimal dargestellt wird. Die Grösse der Pendel Simulation passt sich anhand der Fenstergrösse automatisch an.

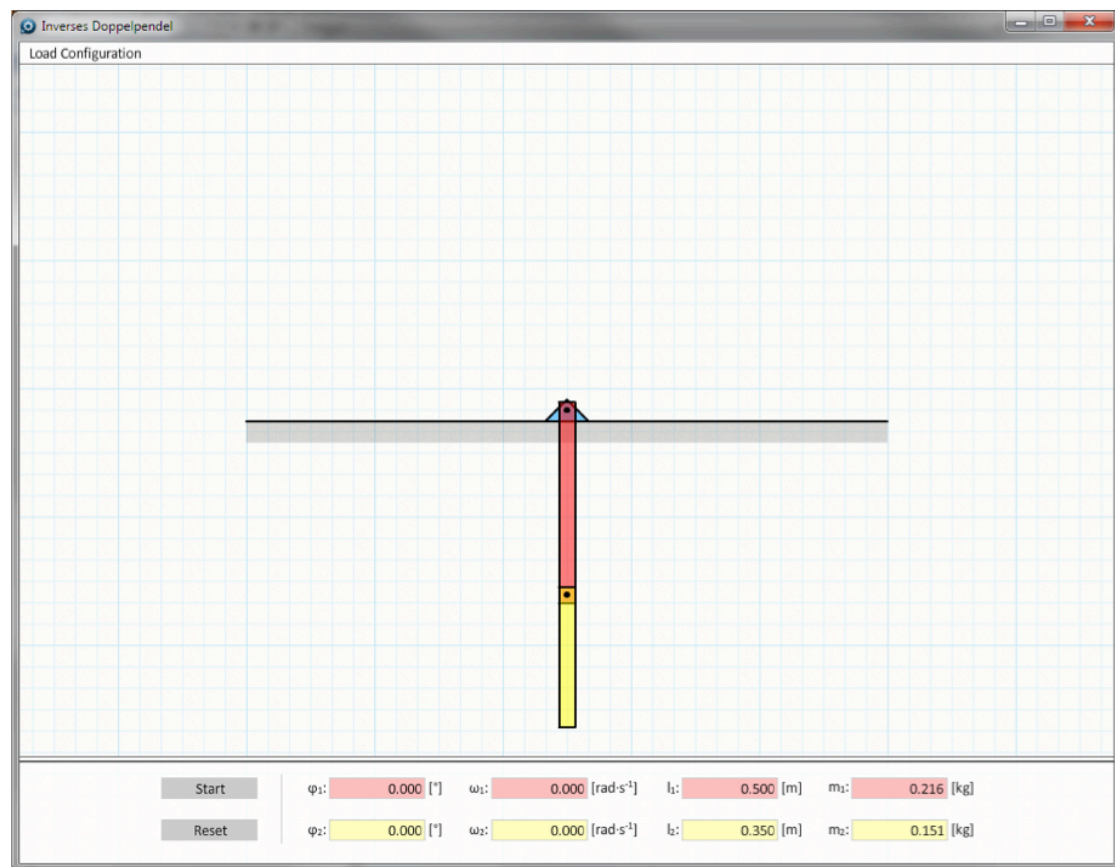


Abbildung 6: Realisierte Grafische Oberfläche GUI

5.2 MVC-Pattern

In einer ersten Version dieser Applikation wurde der Controller getrennt vom Model aufgebaut. Leider führte dies zu ungewollten Verdoppelung von Aufgaben was zu einer schlechteren Kohäsion führte als die Koppelung von Model und Controller.

5.2.1 Model-Controller

Die Funktion des Model-Controllers wird durch die Klasse *Doppelpendel* übernommen. Sie nimmt sämtliche Event-Aufrufe und Daten entgegen und leitet sie an die entsprechenden Klassen weiter.

5.2.2 View

Die View verteilt sich auf die jeweiligen GUI-Klassen *Pendulum*, *PendulumControls*, *Background* und *MenuBar*.

5.3 Bootstrapper

Die Klasse *Main* initialisiert die Applikation und platziert sie auf der *Stage* (Hauptcontainer des GUIs), auf die sie durch die Vererbung der AS3 Klasse *SSprite* Zugriff hat. Über die *Main* Klasse wird die Hauptklasse *Doppelpendel* instanziiert und gestartet.

Die Sprite-Klasse ist ein AS3-Grundbaustein der Anzeigeliste: ein Knoten der Anzeigeliste, der Grafiken anzeigen und auch untergeordnete Objekte enthalten kann.

5.4 Solver Klassen (Runge-Kutta)

Die Bewegungsberechnung des Doppelpendels werden in 3 Klassen berechnet. Die Klasse *PendulumSystem* nimmt die aktuellen Werte der Pendel entgegen und leitet sie zum Berechnen an die Klassen *RungeKutta* und *PendulumSolver* weiter. Die Klasse *RungeKutta* nähert die Differentialgleichung, die in der Klasse *PendulumSolver* definiert ist, über das RK4 verfahren an. Die Lösung wird als Vektor im *PendulumSolver* gespeichert.



Abbildung 7: Klassendiagramm Pendulumsystem

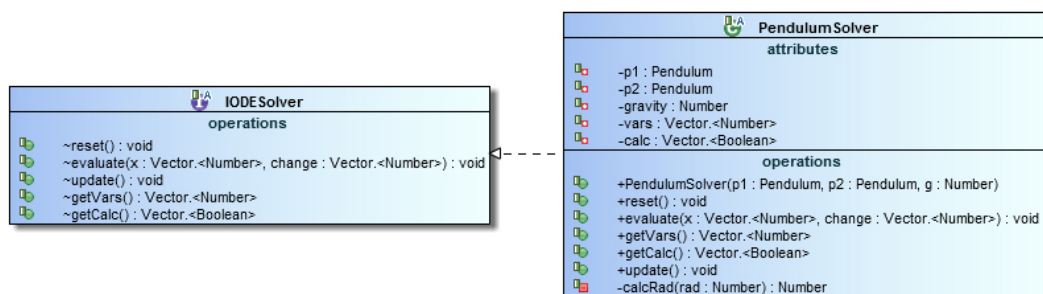


Abbildung 8: Klassendiagramm PendulumSolver/IODESolver

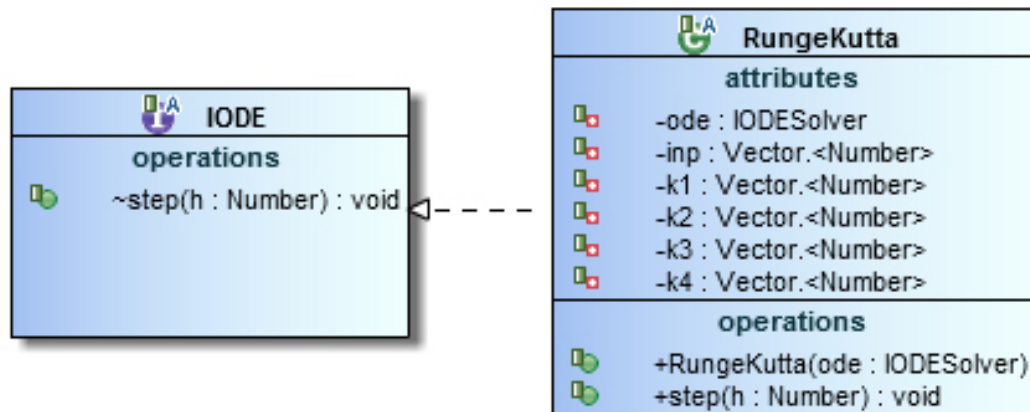


Abbildung 9: Klassendiagramm RungeKutta/IODE

5.5 Konfiguration des Systems

5.5.1 Konfigurationsdateien

Die Software arbeitet mittels Konfigurationsdateien um die verschiedenen Anfangswerte eines Doppelpendels laden zu können. Die Konfigurationsdateien sind im XML Format definiert. Es wird für die Dateien eine eigene Dateieindung verwendet. Die IDP-Datei (IDP steht für Inverses Doppel Pendel) kann in jedem Texteditor bearbeitet werden.

Beispiel einer IDP-Konfiguration:

```
<data>
  <!--
    gravity in m/s^2 (9.81 = earth)
    density in kg/m^3 (2700 = aluminium)
  -->
  <system gravity="9.81" density="2700">
    <!--
      length in m
      phi in grad
      omega in rad*s^(-1)
      color in hex
    -->
    <pendulum length="0.5" phi="180" omega="0" color="0xFF0000" />
    <pendulum length="0.35" phi="180" omega="0" color="0xFFFF00" />
  </system>
</data>
```

Abbildung 10: IPD Konfigurationsdatei

6 xxx

Abbildungsverzeichnis

1	Funktionsweise des Doppelpendels	7
2	Doppelpendel	10
3	Kräfte im oberen Pendel	12
4	Kräfte im unteren Pendel	13
5	Entwurf Grafische Oberfläche	16
6	Realisierte Grafische Oberfläche GUI	17
7	IPD Konfigurationsdatei	19

Literatur

Adobe. Flex sdk, 2012. URL <http://sourceforge.net/adobe/flexsdk/wiki/>.

aont. Double pendulum euler, Juli 2011. URL <http://wonderfl.net/c/lvvN>.

Git. Git - fast version control, 2012. URL <http://git-scm.com/>.

GitHub. Github - social coding, 2012. URL <https://github.com/>.

Helga Dankert Jürgen Dankert. *Technische Mechanik: Statik, Festigkeitslehre, Kinematik/Kinetik*. Vieweg + Teubner, 2009. ISBN 9783835101777. URL http://www.tm-aktuell.de/TM5/Doppelpendel/doppelpendel_grenzen.html.

G Kramann. Simulationsgleichungen eines doppelpendels, 2011. URL http://www.kramann.info/61_Kinematik/02_NewtonEuler/04_Doppelpendel/index.php.

Pivotal Labs. Pivotal tracker, 2012. URL <https://www.pivotaltracker.com/>.

Erik Neumann. Double pendulum, 2004a. URL http://www.myphysicslab.com/beta/dbl_pendulum.html.

Erik Neumann. Runge-kutta algorithm, 2004b. URL http://www.myphysicslab.com/runge_kutta.html.

Powerflasher. Fdt - actionscript, flash, flex, mxml, haxe - ide, editor, 2012. URL <http://fdt.powerflasher.com/>.

Eric W. Weisstein. Double pendulum, 2007. URL <http://scienceworld.wolfram.com/physics/DoublePendulum.html>.

A Projektplan

