Zürcher Hochschule für Angewandte Wissenschaften
Studiengang Informatik

Projektarbeit: Doppelt inverses Pendel

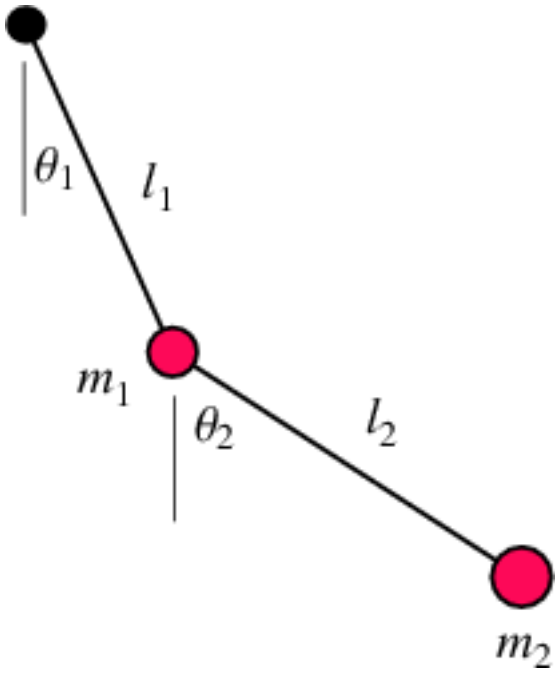


Bild: Wolfram Research

Michael Höhn
hoehnmic@students.zhaw.ch

Stefan Hauenstein
hauenste@students.zhaw.ch

15. Juni 2012

Inhaltsverzeichnis

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 Pendels“ soll untersucht und durch geeignete physikalische und mathematische Beziehungen nachgebildet 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 Computers graphisch dargestellt visualisiert.

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

1.3 Motivation

Physikalische Eigenschaften mit einer objektorientierten Programmiersprache umzusetzen, reizte uns schon immer. Die Aufgabenstellung ermöglichte uns, das theoretisch Gelernte aus dem Studienfach Numerik in Verbindung mit physikalischen Grundlagen in einer praktischen Anwendung umzusetzen. Auch ermöglichte es die algebraischen Zusammenhänge aus vergangenen Vorlesungen besser zu verstehen und Lücken aufzufüllen.

1.4 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 Doppelt inversen Pendels 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.
- Als Ausgangsmaterial des Pendels wird Aluminium 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 aussagekräftigen 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 Projektplanung

Um das Softwareprojekt und die einzelnen Iterationen planen und verwalten zu können, wurde Pivotaltracker [Labs] von Pivotal Labs eingesetzt. Die einzelnen Tasks der Projektplanung können unter [Hoehn and Hauenstein] nachgeschlagen werden.

3.2 Iterationsplan

Iteration	**Termin**	**Story**
1	Mi 04.04.2012	• Evaluierung der Programmiersprache • Physikalische Eigenschaften des Doppelpendels verstehen • Gleichungssystem aufstellen
2	Fr 27.04.2012	• Grafische Benutzeroberfläche • Programmierung des ersten Pendels • Programmierung des zweiten Pendels • Runge Kutta definieren
3	Mi 23.05.2012	• Pendel einstellen • Projektdokumentation
4	SA 16.06.2012	• Projektdokumentation • Präsentation

3.3 Programmiersprache

Das Programm wird mittels der objektorientierten Programmiersprache ActionScript 3 (AS3) und dem Adobe Flex SDK Framework [Adobe] 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 Animationen ausgelegt ist und es dadurch zu einer grossen Zeiteinsparung bei der Entwicklung kommt.

3.4 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 Pendelsimulation passt sich anhand der Fenstergrösse automatisch an.

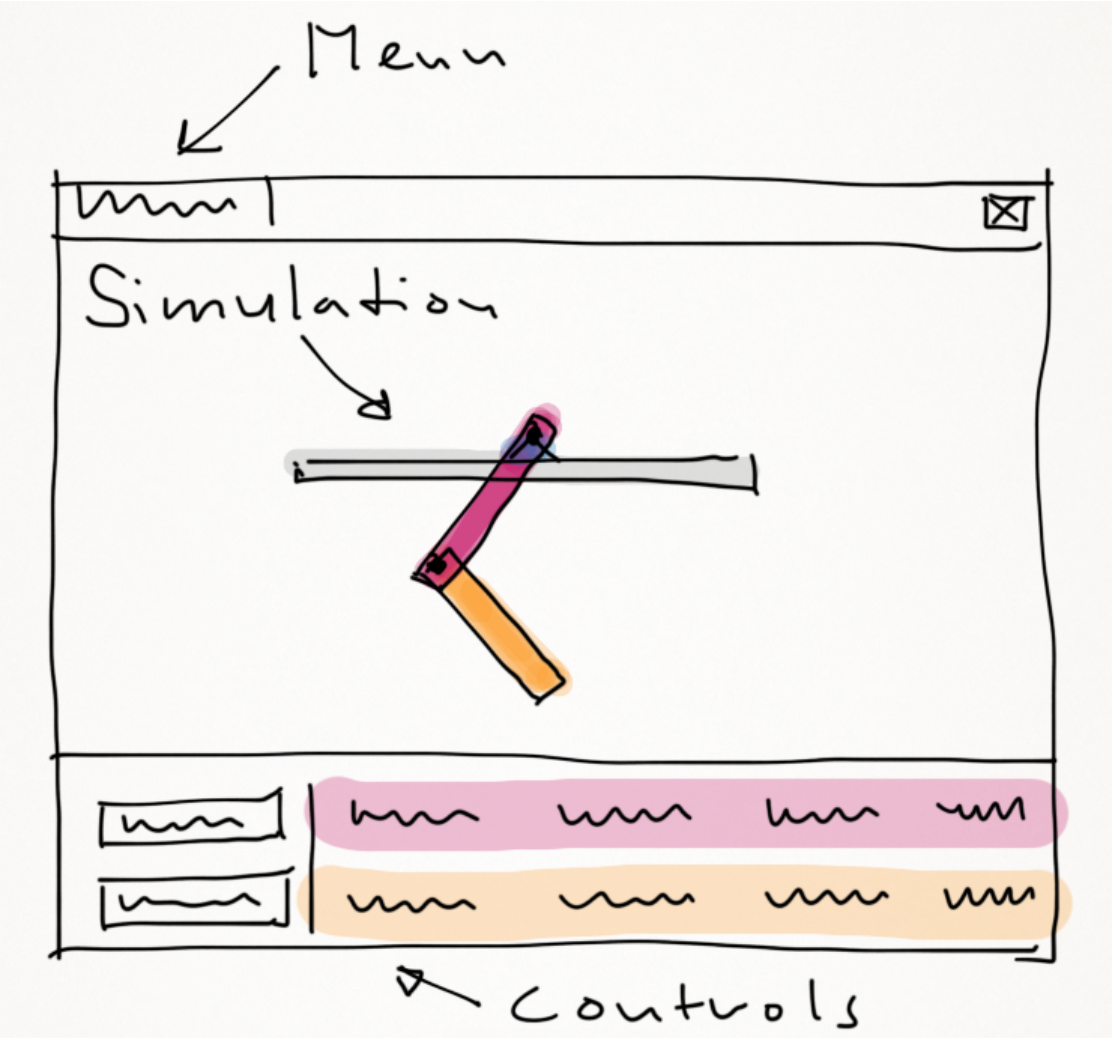


Abbildung 1: Entwurf grafische Oberfläche

4 Physikalische und mathematische Zusammenhänge

4.1 Funktionsprinzip Doppelpendel

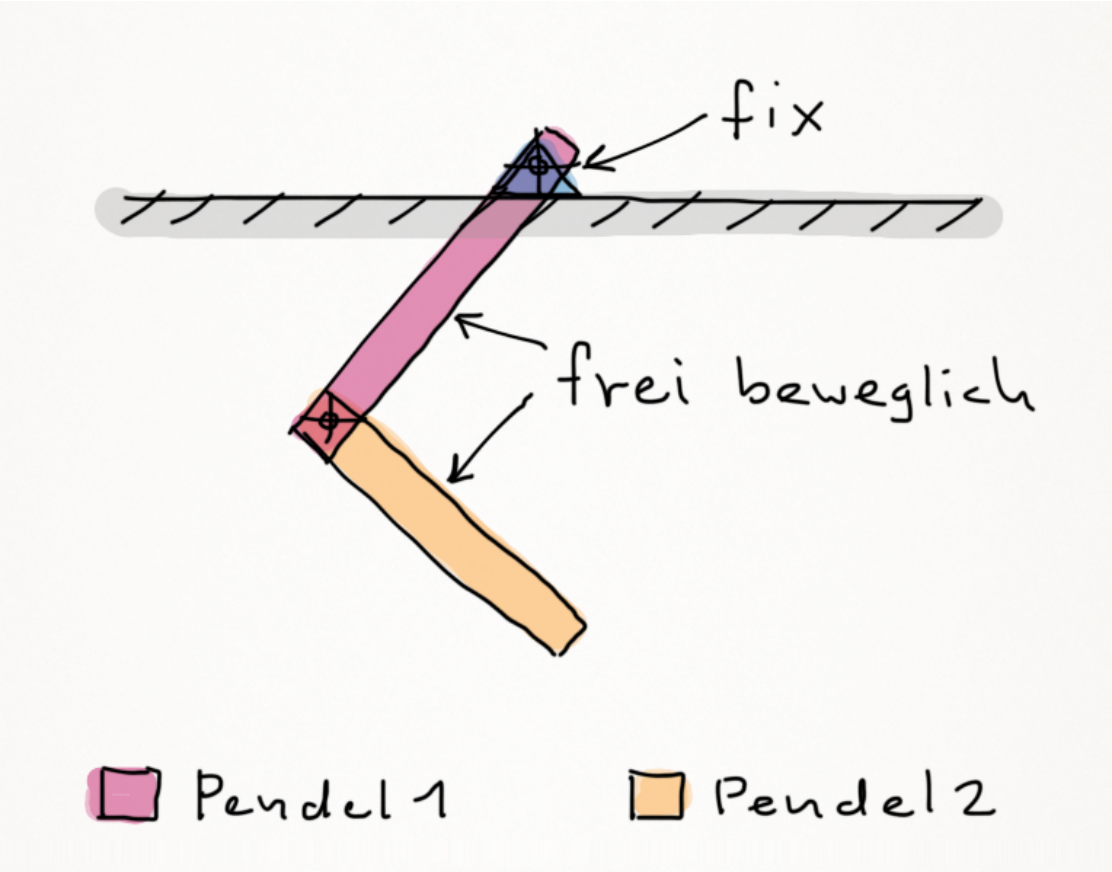


Abbildung 2: Funktionsweise des Doppelpendels

4.2 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, 2004], [Kramann, 2011] und [Weisstein, 2007] konnten die Kinematik und Kräfte des Doppelpendels zum Ausgangsproblem kombiniert werden.

4.2.1 Kinematik des Doppelpendels

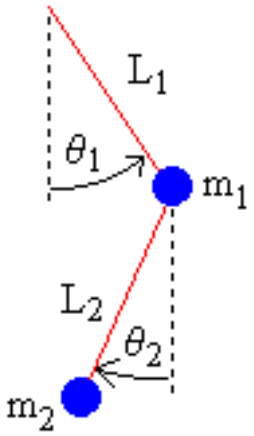


Abbildung 3: Doppelpendel [Neumann, 2004]

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

Ausgangslage für die Berechnung ist das obere Pendel. Das erste Pendel P_1 wird definiert mit x_1 und y_1. Äquivalent dazu wird das zweite Pendel P_2 definiert mit x_2 und y_2.

Als erstes werden die trigonometrischen Gleichungen aufgestellt:

$$x_1 = L_1 \sin\theta_1 \tag{4.2.1}$$

$$y_1 = -L_1 \cos\theta_1 \tag{4.2.2}$$

$$x_2 = x_1 + L_2 \sin\theta_2 \tag{4.2.3}$$

$$y_2 = y_1 - L_2 \cos\theta_2 \tag{4.2.4}$$

Danach wird aus den Gleichungen (4.2.1 - 4.2.4) die erste und zweite Ableitung gebildet.

Erste Ableitung (Geschwindigkeit):

$$x_1' = \theta_1' L_1 \cos\theta_1 \tag{4.2.5}$$

$$y_1' = \theta_1' L_1 \sin\theta_1 \tag{4.2.6}$$

$$x_2' = x_1' + \theta_2' L_2 \cos\theta_2 \tag{4.2.7}$$

$$y_2' = y_1' + \theta_2' L_2 \sin\theta_2 \tag{4.2.8}$$

Zweite Ableitung (Beschleunigung):

$$x_1'' = -\theta_1'^{\,2} L_1 \sin\theta_1 + \theta_1'' L_1 \cos\theta_1 \tag{4.2.9}$$

$$y_1'' = \theta_1'^{\,2} L_1 \cos\theta_1 + \theta_1'' L_1 \sin\theta_1 \tag{4.2.10}$$

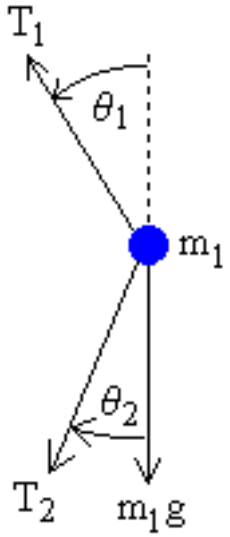
$$x_2'' = x_1'' - \theta_2'^{\,2} L_2 \sin\theta_2 + \theta_2'' L_2 \cos\theta_2 \tag{4.2.11}$$

$$y_2'' = y_1'' + \theta_2'^{\,2} L_2 \cos\theta_2 + \theta_2'' L_2 \sin\theta_2 \tag{4.2.12}$$

4.2.2 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:

4.2.2.1 Oberes Pendel



T_1 = Kraft im oberen Pendel
m_1 = Masse des oberen Pendels
g = Gravitationskonstante

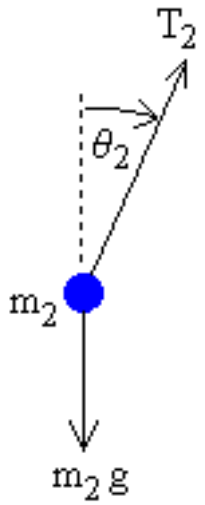
Abbildung 4: Kräfte im oberen Pendel [Neumann, 2004]

Anhand von Abbildung 4 können die folgenden Gleichungen aufgestellt werden.

$$m_1 x_1'' = -T_1 \sin\theta_1 + T_2 \sin\theta_2 \tag{4.2.13}$$

$$m_1 y_1'' = T_1 \cos\theta_1 + T_1 \cos\theta_2 - m_1 g \tag{4.2.14}$$

4.2.2.2 Unteres Pendel



T_2 = Kraft im unteren Pendel
m_2 = Masse des unteren Pendels
g = Gravitationskonstante

Abbildung 5: Kräfte im unteren Pendel
[Neumann, 2004]

Anhand von Abbildung 5 können die folgenden Gleichungen aufgestellt werden.

$$m_2 x_2'' = -T_2 \sin\theta_2 \tag{4.2.15}$$

$$m_2 y_2'' = T_2 \cos\theta_2 - m_2 g \tag{4.2.16}$$

4.2.3 Umformen der Gleichungen

Um zu der endgültigen differenziellen Gleichung für die Berechnung zu gelangen, müssen die einzelnen Gleichungen algebraisch umgeformt werden. Die Kräftegleichungen (4.2.15) und (4.2.16) werden nach $T_2 \sin\theta_2$ bzw. nach $T_2 \cos\theta_2$ aufgelöst und in die Gleichungen (4.2.13) und (4.2.14) eingesetzt.

$$m_1 x_1'' = -T_1 \sin\theta_1 - m_2 x_2'' \tag{4.2.17}$$

$$m_1 y_1'' = T_1 \cos\theta_1 - m_2 y_2'' - m_2 g - m_1 g \tag{4.2.18}$$

Gleichung (4.2.17) wird mit cos θ_1 und Gleichung (4.2.18) mit sin θ_1 erweitert und nach $T_1 \sin\theta_1 \cos\theta_1$ aufgelöst.

$$T_1 \sin\theta_1 \cos\theta_1 = -\cos\theta_1(m_1 x_1'' + m_2 x_2'') \tag{4.2.19}$$

$$T_1 \sin\theta_1 \cos\theta_1 = \sin\theta_1(m_1 y_1'' + m_2 y_2'' + m_2 g + m_1 g) \tag{4.2.20}$$

Daraus ergibt sich der erste Teil der Bewegungsgleichung.

$$\sin\theta_1(m_1 y_1'' + m_2 y_2'' + m_2 g + m_1 g) = -\cos\theta_1(m_1 x_1'' + m_2 x_2'') \tag{4.2.21}$$

Um den zweiten Teil der Gleichung zu erhalten, wird die Gleichung (4.2.15) mit cos θ_2 und die (4.2.16) mit sin θ_2 erweitert.

$$T_2 \sin\theta_2 \cos\theta_2 = -\cos\theta_2(m_2 x_2'') \tag{4.2.22}$$

$$T_2 \sin\theta_2 \cos\theta_2 = \sin\theta_2(m_2 y_2'' + m_2 g) \tag{4.2.23}$$

Der zweite Teil der Gleichung ergibt somit

$$\sin\theta_2(m_2 y_2'' + m_2 g) = -\cos\theta_2(m_2 x_2'') \tag{4.2.24}$$

x_2'', x_2'', y_1'' und y_2'' werden durch die Gleichungen (4.2.9 – 4.2.12) ersetzt und mit Hilfe des Lagrange-Formalismus nach θ_1'' und θ_2'' aufgelöst.

Die endgültige Bewegungsgleichung für das Doppelpendel lautet somit:

$$\theta_1'' = \frac{-g(2m_1 + m_2)\sin\theta_1 - m_2 g \sin(\theta_1 - 2\theta_2) - 2\sin(\theta_1 - \theta_2)m_2(\theta_2'^2 L_2 + \theta_1'^2 L_1 \cos(\theta_1 - \theta_2))}{L_1(2m_1 + m_2 - m_2\cos(2\theta_1 - 2\theta_2))} \tag{4.2.25}$$

$$\theta_2'' = \frac{2\sin(\theta_1 - \theta_2)(\theta_1'^2 L_1(m_1 + m_2) + g(m_1 + m_2)\cos\theta_1 + \theta_2'^2 L_2 m_2 \cos(\theta_1 - \theta_2))}{L_2(2m_1 + m_2 - m_2\cos(2\theta_1 - 2\theta_2))} \tag{4.2.26}$$

5 Numerisches Verfahren

Um den Fehler der numerischen Lösung pro Berechnungsschritt 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 jedoch wieder aus.

5.1 Runge-Kutta Verfahren 4. Ordnung (RK4)

Das klassische Runge-Kutta-Verfahren (nach Carl Runge und Martin Wilhelm Kutta) ist ein spezielles explizites 4-stufiges Verfahren zur numerischen Lösung von Anfangswertproblemen. Runge hat als erster (1895) ein mehrstufiges Verfahren angegeben und Kutta (1905) die allgemeine Form expliziter s-stufiger Verfahren definiert.

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 Differenzenquotienten reduziert werden. Das RK4 verfahren ist eine solche Kombination, die Diskretisierungsfehler bis zur dritten Ableitung kompensiert.

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

$$y^{k+1} = y^k + h\left(\frac{1}{6}k_1 + \frac{2}{6}k_2 + \frac{2}{6}k_3 + \frac{1}{6}k_4\right) \tag{5.1.1}$$

$$\begin{aligned}
\text{mit} \quad k_1 &= f(t_k, y^k) \\
k_2 &= f\left(t_k + \frac{h}{2}, y^k + \frac{h}{2}k_1\right) \\
k_3 &= f\left(t_k + \frac{h}{2}, y^k + \frac{h}{2}k_2\right) \\
k_4 &= f\left(t_k + h, y^k + hk_3\right)
\end{aligned}$$

5.2 Differentialgleichung für Runge-Kutta

Die Bewegungsgleichung (4.2.25 und 4.2.26) lässt sich 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 (5.1.1) folgende Schlussgleichungen:

$$\theta_1' = \omega_1 \tag{5.2.1}$$

$$\theta_2' = \omega_2 \tag{5.2.2}$$

$$\omega_1' = \frac{-g(2m_1 + m_2)\sin\theta_1 - m_2 g \sin(\theta_1 - 2\theta_2) - 2\sin(\theta_1 - \theta_2)m_2(\omega_2^2 L_2 + \omega_1^2 L_1 \cos(\theta_1 - \theta_2))}{L_1(2m_1 + m_2 - m_2 \cos(2\theta_1 - 2\theta_2))} \tag{5.2.3}$$

$$\omega_2' = \frac{2\sin(\theta_1 - \theta_2)(\omega_1^2 L_1(m_1 + m_2) + g(m_1 + m_2)\cos\theta_1 + \omega_2^2 L_2 m_2 \cos(\theta_1 - \theta_2))}{L_2(2m_1 + m_2 - m_2 \cos(2\theta_1 - 2\theta_2))} \tag{5.2.4}$$

6 Software

6.1 Verwendete Technologien

6.1.1 Flash Development Tool (FDT)

Als Entwicklungsumgebung wird FDT [Powerflasher] eingesetzt. FDT stellt für die Entwicklung von AS3 und Flex-Projekten nützliche Funktionen wie Fehlererkennung, Debugger und Autocomplete zur Verfügung.

6.1.2 Versionskontrolle

Die Versionskontrolle des Quellcodes wurde mit git [Git] umgesetzt. Das Git repository des Doppelt inversen Pendels ist auf GitHub [GitHub] verfügbar. Im Git repository befindet sich auch die Dokumentation und eine Kopie der GNU Public License, unter der diese Software veröffentlicht ist.

6.2 Aufbau

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

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

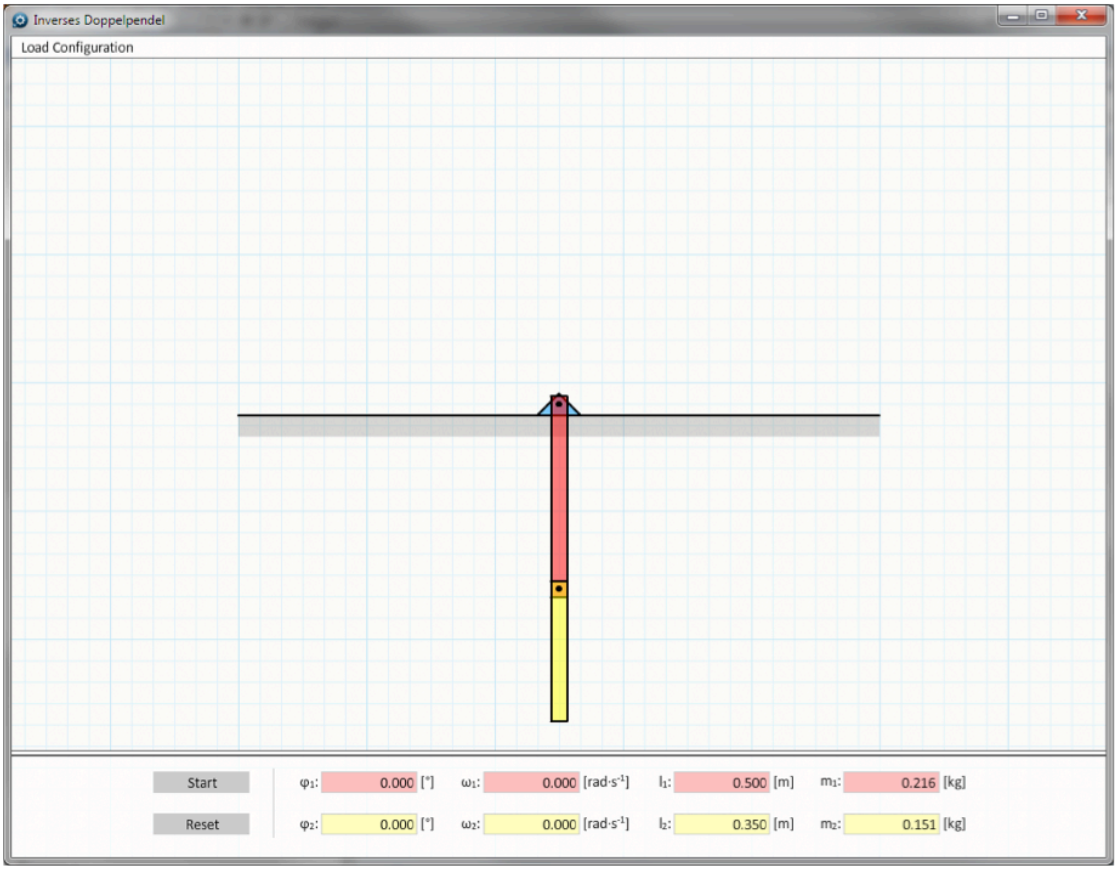


Abbildung 6: Grafische Oberfläche Doppelt inverses Pendel

6.3 MVC-Pattern

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

6.3.1 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.

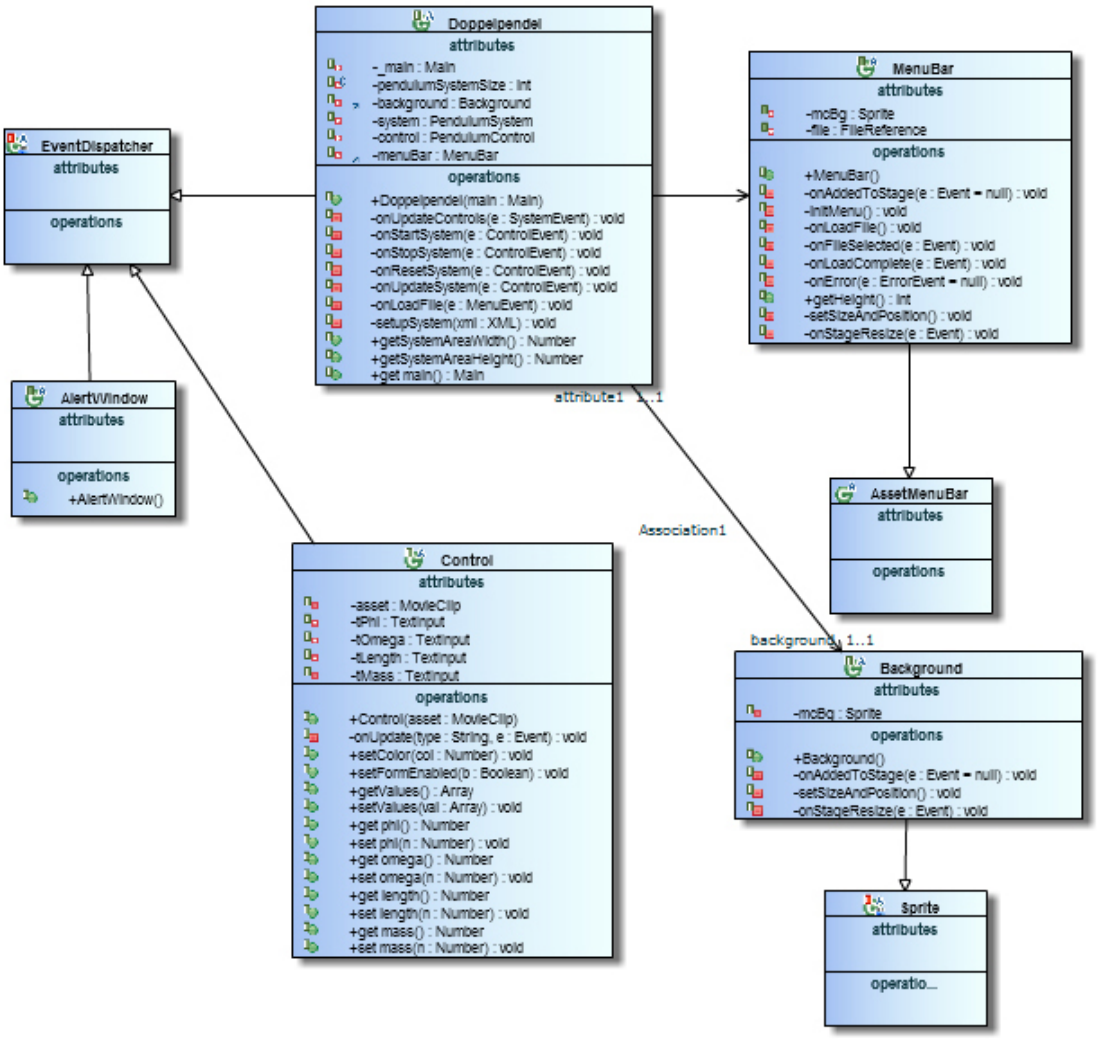


Abbildung 7: Klassendiagramm Doppelpendel

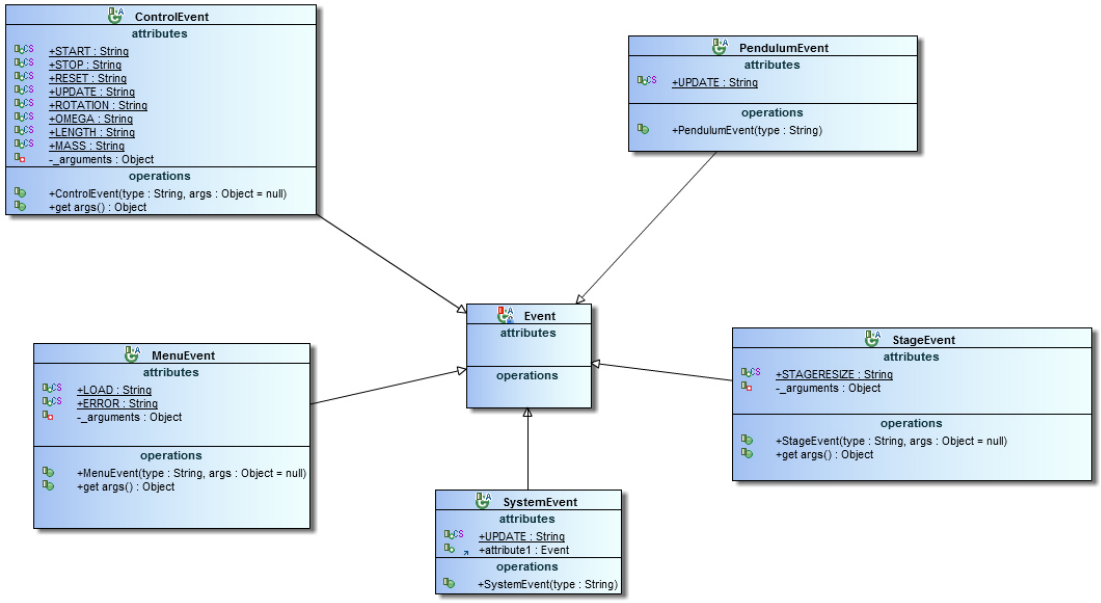


Abbildung 8: Klassendiagramm Eventsystem

6.3.2 View

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

6.4 Solver Klassen (Runge-Kutta)

Die Bewegungsgleichung des Doppelpendels wird in drei abgetrennten 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 9: Klassendiagramm Pendulumsystem

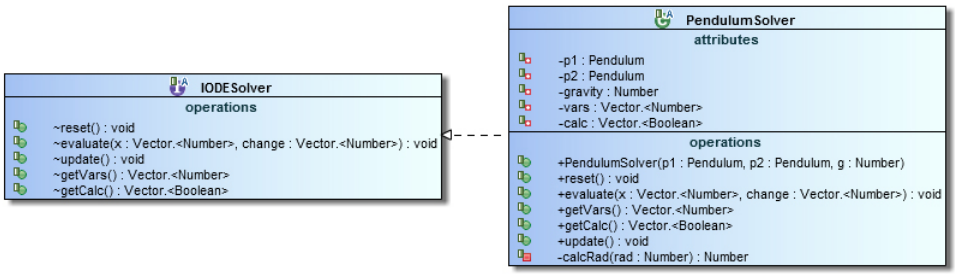


Abbildung 10: Klassendiagramm PendulumSolver/IODESolver

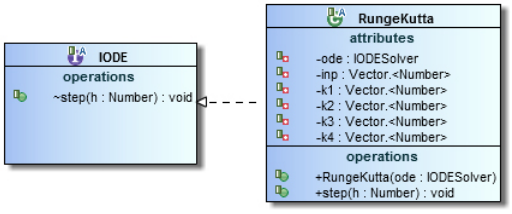


Abbildung 11: Klassendiagramm RungeKutta/IODE

6.5 Konfiguration des Systems

6.5.1 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 Dateiendung verwendet. Die IDP-Datei (IDP steht für Inverses Doppel Pendel) kann in jedem Texteditor bearbeitet werden.

Beispiel einer IDP-Konfigurationsdatei:

```
<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 12: IDP Konfigurationsdatei

7 Fazit Projektarbeit

Die Entscheidung die Projektarbeit mit AS3 und dem FlexSDK zu entwickeln, erwies sich schnell als richtig. Die grundlegenden Strukturen der Applikation konnten in kurzer Zeit umgesetzt werden und dem Experten zur Ansicht vorgelegt werden. Um den Effekt der Pendelbewegung möglichst schnell zu visualisieren, wurde die Bewegung zu beginn als lineare Animation implementiert. Die Animation der Pendel konnte dank der losen Kopplung der AS3 Strukturen leicht gegen die finale Bewegungsgleichung ausgetauscht werden.

Obschon wir mit dem Resultat der Implementierung in AS3 sehr zufrieden sind, mussten wir feststellen, dass die AS3 Programmiersprache bezüglich Performance von Realtime Berechnungen gewisse Nachteile mit sich brachte. Die Animation des Doppelpendels ruckelte zu Beginn relativ Stark, was uns zu umfangreichen Optimierungen im Renderer des Pendelsystems gezwungen hat.

Die Projektarbeit über das Doppelt inverse Pendel ermöglichte uns wertvolle Einblicke in physikalische Problemstellungen und deren numerische Lösungsverfahren. Insgesamt war diese Projektarbeit für uns sehr spannend und aufschlussreich.

Abbildungsverzeichnis

Literatur

Adobe. Flex sdk. URL http://sourceforge.net/adobe/flexsdk/wiki/.

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

Git. Git - fast version control. URL http://git-scm.com/.

GitHub. Github - social coding. URL https://github.com/minimih/Doppelpendel.

Michael Hoehn and Stefan Hauenstein. Doppelt inverses pendel projektplanung. URL https://www.pivotaltracker.com/projects/505419.

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

Pivotal Labs. Pivotal tracker. URL https://www.pivotaltracker.com/.

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

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

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

A Projektplan