Programmieren in C | Programmentwurf

In diesem Programmentwurf können 10 Punkte erreicht werden. Stellen Sie ihren Source Code zur dieswöchigen Deadline als Github Repository zur Verfügung.

Zur Abgabe Legen Sie die Datein `eulerLib.c`, `eulerLib.h` und `main.c`, welche ihren Lösungscode beinhalten in einem Ordner **HA04** in ihrem persönliches Repository ab. Schreiben Sie ihr Programm so, dass der Compiler weder Errors noch Warnings schmeißt (`gcc eulerLib.c main.c -Wall -o main.exe`)

Theorie: Feder-Dämpfer System

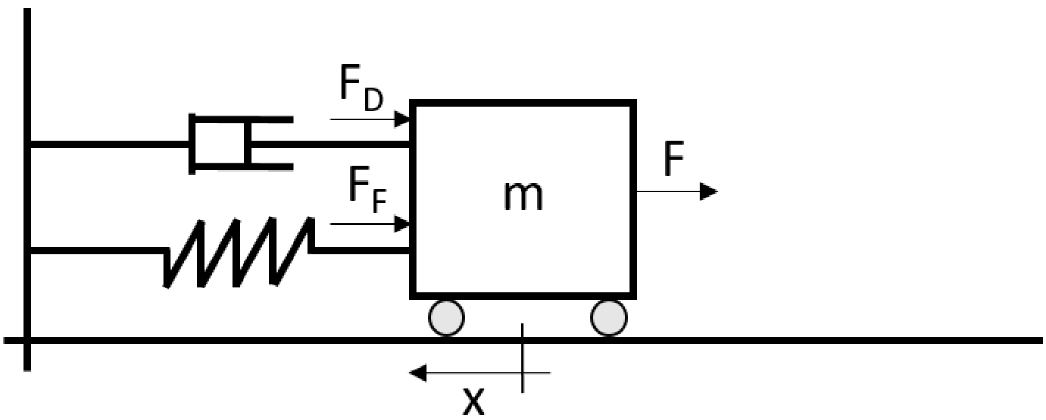

Abbildung 1: Einfaches Masse-Feder-Dämpfer System

In dieser Hausaufgabe soll das Verhalten des in Abbildung 1 dargestellten Masse-Feder-Dämpfer Systems modelliert und simuliert werden. Dazu kann zunächst unter Vernachlässigung von Reibung die folgende Kräftebilanz aufgestellt werden:

$$F_T = -F_D - F_F - F \tag{1}$$

Die Trägheitskraft F_T setzt sich als Summe aus Federkraft F_F Dämpferkraft F_D und einer äußeren angreifenden Kraft F nach Gleichung 1 zusammen.

$$F_T = m \cdot \ddot{x} \tag{2}$$

$$F_D = d \cdot \dot{x} \tag{3}$$

$$F_F = c \cdot x \tag{4}$$

In den beschreibenden Gleichungen bezeichnet m die Masse des Objekts, d die Dämpfungskonstante, c die Federkonstante, $x = x(t)$ den Ort, $\dot{x} = \dot{x}(t)$ die Geschwindigkeit und $\ddot{x} = \ddot{x}(t)$ die Beschleunigung der Masse. Unter Vernachlässigung einer äußeren angreifenden Kraft* und durch Einsetzen von Gleichung 2 bis 4 in 1 erhält folgende gewöhnliche Differentialgleichung zweiter Ordnung:

$$m \cdot \ddot{x} = -d \cdot \dot{x} - c \cdot x \tag{5}$$

Durch normieren von Gleichung 5 nach der Objektmasse m ergibt sich:

$$\ddot{x} = -\left(\frac{d}{m} \cdot \dot{x} + \frac{c}{m} \cdot x\right) \tag{6}$$

Mit Einführung der Variablen v als Geschwindigkeit†, kann Gleichung 6 in ein Diffentialgleichungssystem erster Ordnung überführt werden:

$$\boxed{\begin{aligned} \dot{v} &= -\left(\frac{d}{m} \cdot v + \frac{c}{m} \cdot x\right) && (7) \\ \dot{x} &= v && (8) \end{aligned}}$$

Mit Einführung des Zustandsvektors

$$y = \begin{pmatrix} x \\ v \end{pmatrix} \tag{9}$$

kann das (DGL) Differentialgleichungssystem 7 und 8 in der allgemeinen Form nach Gleichung 10 geschrieben werden

$$\dot{y} = f\left(y\right) = f\left(y\left(x, v\right)\right) \tag{10}$$

* F = 0

† $\dot{x} = v \rightarrow \ddot{x} = \dot{v}$

Aufgabe: Simulation eines Feder-Dämpfer Systems **9 Punkte**

In dieser Hausaufgabe soll das Masse-Feder-Dämpfer beschreibende DGL-System (7) & (8), sowie das in der Vorlesung besprochene explizite Eulerverfahren zum numerischen Lösen von Ersterem implementiert werden. Dazu finden Sie in der Datei *HA04_EulerLib.c* als Kommentar folgende Anweisung:

```
/* YOUR CODE HERE */
/* ------------ */
```

Implementieren Sie hier nach den jeweiligen Anweisungen der Makrostruktur ihren Code. Gehen Sie dabei in den gegebenen Funktionen wie folgt vor:

- `RHS_MSD`: Berechnung des DGL-System (7) & (8) mit Rückgabe der Werte in `rhs[0]` ($\dot{x}$) und `rhs[1]` ($\dot{v}$)
- `eulerSettings_MSD`: Initialisierung des simHandles.
  - `numOfStates` entspricht `(int)NUMOFSTATES`.
  - `f` wird `RHS_MSD` zugewiesen.
  - `stateVecInit` wird auf dynamisch allokierten Speicher der Größe `sizeof(double) * (handle->numOfStates)` gesetzt.
  - `simTime`, `stepSize`, sowie `stateVecInit` (`x_0` und `v_0`) vom user via Terminal entgegennehmen.
  - `stateVec` wird dynamisch allokierter Speicher zugewiesen. Der Speicher muss genau so groß sein, dass die Werte des Zustandsvektors zu jeder generierten Lösung abgelegt werden kann.[‡]
  - `derivStateVec` wird dynamisch allokierter Speicher in gleicher Größe wie `stateVec` zugewiesen.
- `eulerForward`: Berechnung der Lösungen des DGL Systems auf den Zeitgitter, welches sich durch die Simulationszeit und der Zeitschrittweite ergibt.
- `showResults_MSD`: Visualisierung der Ergebnisse
  1.) Schreiben Sie die das vom Integrator generierte Zeitgitter, sowie die Lösungsvektoren von Position und Geschwindigkeit in eine Textdatei mit der Bezeichnung `simData.txt`. Vergessen Sie nicht zu überprüfen, ob `fopen NULL` zurückgibt und abschließend den Filestream wieder zu schließen.
  2.) Plotten Sie die Ergebnisse, welche nun in `simData.txt` abgelegt sind via Gnuplot. Der plot muss x-Achsenbeschriftung, Titel und eine Legende enthalten[§] Verwenden Sie für den Gnuplot Aufruf `popen`.

[‡]Sie benötigen die gesamte Historie der Lösungen um diese anschließen visualisieren zu können.
[§]Zb. wie in Abbildung 2 dargestellt.

Aufgabe: Untersuchung des Aperiodischen-Grenzfalls **1 Punkt**

Bestimmen Sie anschließend, ab welcher ganzzahligen Dämpferkonstante d sich bei ansonsten fixen Parametern ($m = 1$, $c = 2$, $x_0 = 1$, $v_0 = 0$) der Aperiodische Grenzfall, welcher in Abbildung 2[¶] dargestellt ist, einstellt. Implementieren Sie den entsprechenden Zahlenwert in der `RHS_MSD`-Funktion.

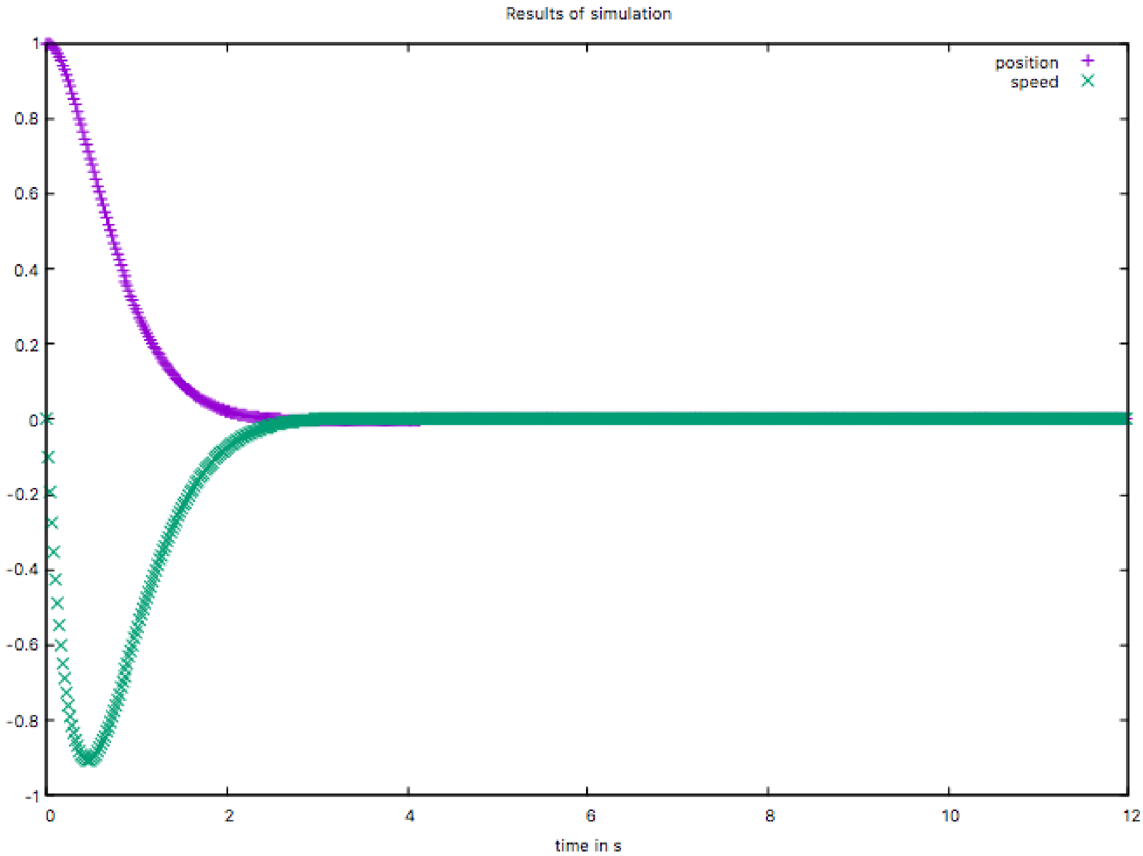

Abbildung 2: Aperiodischer Grenzfall

[¶]Kleinstmögliche Dämpfung, welche gerade so stark ist, dass das schwingungsfähige System ohne Überschwingen in die Nulllage zurückgeht.