Kurs 01610 SIMULATION

Einsendeaufgaben zur Kurseinheit 1

Aufgabe 1 (25 Punkte)

In dieser Kurseinheit führen wir gleich die erste Modellbildung und Simulation durch. Wir beobachten einen Trainer, der seinen Blindenhund an einer Leine der Länge l hinter sich herzieht oder vor sich herschiebt. Wir modellieren die zeitabhängige Position der beiden als Punkte in der Ebene mit $T(t) = (x(t), y(t))^\mathsf{T}$ für den Trainer und $H(t) = (X(t), Y(t))^\mathsf{T}$ für den Blindenhund. Der Abstand zwischen den beiden entspricht der Länge l der Leine.
Gegeben sei die Geschwindigkeit des Trainers in x-Richtung mit $v_x(t) = -5\sin(t)$ und in y-Richtung mit $v_y(t) = 5\cos(t)$. Die Länge der Leine sei $l = 5$ und als Anfangswerte wählen wir $T(0) = (5, 0)^\mathsf{T}$ und $H(0) = (10, 0)^\mathsf{T}$.
Für die Kopplung des Systems nehmen wir an, dass sich die Geschwindigkeit des Blindenhundes ergibt, indem man den Geschwindigkeitsvektor des Trainers auf die Richtung der Leine projiziert. Die Richtung der Leine lässt sich wie folgt berechnen:

$$d(t) = \begin{pmatrix} X(t) - x(t) \\ Y(t) - y(t) \end{pmatrix}$$

Die Simulation können Sie mit der ODEINT-Funktion in Python lösen, dazu stellen wir Ihnen ein Jupyter-Notebook[1] zur Verfügung. Alternativ kann die Aufgabe natürlich auch mit anderen Programmiersprachen gelöst werden, zum Beispiel MATLAB oder Java. In diesem Fall kann allerdings keine Korrektur Ihrer Einsendung garantiert werden, da die Korrektoren ein begrenztes Zeitbudget zur Verfügung haben.

1. **Bestimmen Sie die Bahnkurve des Trainers.**
   - Versuchen Sie die Differentialgleichung[2] mit Stift und Papier zu lösen.
   - Lösen Sie die Differentialgleichung mithilfe der ODEINT-Funktion und plotten Sie die Lösung der Bahnkurve in einem x-y-Plot.
2. **Bestimmen Sie die Bahnkurve des Blindenhundes.**
   - Bestimmen Sie das Differenzialgleichungssystem durch Projizierung des Geschwindigkeitsvektors auf den Richtungsvektor der Leine.[3]
   - Lösen Sie das Gleichungssystem mit der ODEINT-Funktion aus dem Paket scipy.integrate.
   - Plotten Sie beide Bahnkurven in einen x-y-Plot und analysieren Sie das Verhalten des Hundes im Vergleich zur Bahnkurve des Trainers.
3. **Validieren Sie das Ergebnis mit Ihrem eigenen Haustier.**

[1] Installieren Sie ANACONDA und öffnen Sie das enthaltene Jupyter-Lab: https://docs.anaconda.com/
[2] Hilfreich ist Kapitel 2 des Basistexts, sowie die Beispielrechnung im Jupyter-Notebook
[3] Sie können diese Gleichung per Hand oder mit der numpy Bibliothek lösen

Aufgabe 2 (15 Punkte)

Zwei Hunde, Paul und Emil, werden beim Spaziergang von einem Radfahrer abgelenkt, der auf dem Fahrradträger eine Brotdose mit Brotzeit dabei hat. Ruft der Hundebesitzer die beiden zurück, haben sie zwei Möglichkeiten: *Gehorchen* und zum Besitzer zurücklaufen oder *Nicht Zuhören* und dem Radfahrer folgen.

- Kommen beide Hunde zurück, erhalten sie jeweils drei Stückchen Käse als Belohnung.
- Gehorcht ein Hund und der andere nicht, erhält der brave Hund 6 Käsewürfel und der andere keinen.
- Rennen beide Hunde dem Radfahrer hinterher, können sie gemeinsam die Brotdose ergattern und erhalten jeweils 4 Käsewürfel.

Wie entscheiden sich Paul und Emil?

Aufgabe 3 (20 Punkte)

Stellen Sie das Kinderspiel Schere-Stein-Papier spieltheoretisch in strategischer Normalform, d.h. mit Strategiemengen und Nutzenmatrix, dar, wobei die Auszahlungen bei

- Gewinn: 1,
- Unentschieden: 0,
- Niederlage: -1

betragen sollen.
Gibt es eine dominante Strategie?

Aufgabe 4 (15 Punkte)

Lösen Sie das folgende Problem mit einem List Scheduler (LS): Auf p gleichartige Ressourcen werden n gleichartige Tasks verteilt, wobei für jede Task i die Laufzeit t_i bekannt ist. Die Tasks werden nacheinander verplant, wobei die Ressource gewählt wird, die am frühesten zur Verfügung steht.[4]

1. Zeigen Sie, dass der LS eine Sequenz von 4 Tasks mit Laufzeiten 5, 1, 1, 5 nicht optimal auf 2 gleichartige Ressourcen verteilt. Geben Sie eine alternative Reihung der Tasks an, die zu einer optimalen Verteilung führt.
2. Implementieren Sie einen LS. Verteilen Sie die Taskliste t-1-4.txt mit n=1000 Tasks auf p=2,4,8 Ressourcen. Wann beendet jeweils die letzte Ressource die Bearbeitung?

[4]Bei mehreren Ressourcen mit den gleichen Laufzeiten wählt man die mit dem kleinsten Index.

Aufgabe 5 (25 Punkte)

Sie machen ein Photo von Emil und Paul mit einer Stereokamera, die zwei Bilder erzeugt (left und right). Daraufhin implementieren Sie die folgende image processing pipeline, um eine *depth map* mittels *stereo depth estimation* zu erhalten:

Aktion	Abkürzung	Laufzeit
Bild einlesen	imgread	$t_{\text{imgread,left}} = t_{\text{imgread,right}} = 1$
Konvertierung in Graustufen	grayscale	$t_{\text{grayscale,left}} = t_{\text{grayscale,right}} = 2$
sobel filter anwenden	sobel	$t_{\text{sobel,left}} = t_{\text{sobel,right}} = 6$
sum of absolute differences berechnen	SAD	$t_{\text{SAD}} = 10$
Ausgabebild schreiben	imgwrite	$t_{\text{imgwrite}} = 1$

Der task graph der Anwendung sieht wie folgt aus:

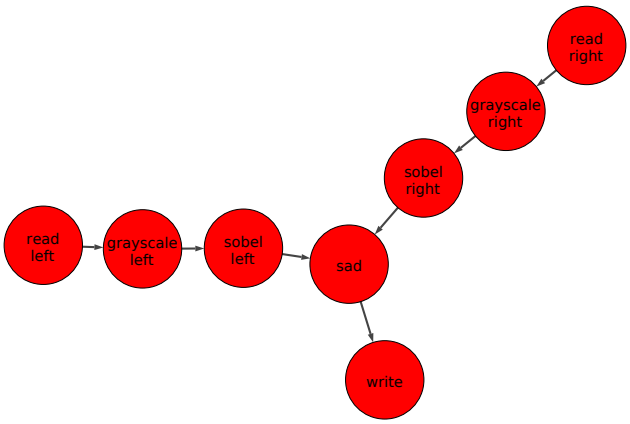

Abbildung 1: Task graph für stereo depth estimation

1. Berechnen Sie die makespan für deterministische Laufzeiten, d.h. die tatsächlichen Laufzeiten entsprechen den erwarteten Laufzeiten aus der Tabelle,
   - bei sequentieller Ausführung in der oben genannten Reihenfolge,
   - bei paralleler Ausführung auf einer Maschine mit 2 CPUs und folgendem mapping:
     - P_0: imgread left, grayscale left, sobel left,
     - P_1: imgread right, grayscale right, sobel right, SAD, imgwrite.
2. Berechnen Sie die makespan für stochastische Laufzeiten, wobei die Abweichungen $[-1, 0, +1]$ mit der Wahrscheinlichkeitsverteilung $[40\%, 10\%, 50\%]$ eintreten. Simulieren Sie dazu die sequentielle und die parallele Ausführung über je 1000 Runden. Wie stark weicht die tatsächliche makespan im Mittel von der prognostizierten ab?