Theorie der Sensorfusion

U. Frese
`http://www.informatik.uni-bremen.de/agebv/de/VeranstaltungTDS12/`

Übungsblatt 4

Abgabe: Mo, 28.01.2013, im Kurs

Bitte bearbeitet die Übungszettel in Gruppen zu 2-3 Teilnehmern und gebt Eure Ausarbeitung am 14.01.2013 im Kurs ab. Schreibt Namen und Email aller Gruppenmitglieder auf die Abgabe. Die Ausarbeitungen können nach freier Wahl handschriftlich oder mit dem Computer gesetzt oder gemischt sein.

Als Hilfe könnt Ihr das Mathematik Merkblatt in Kapitel 1 des Skriptes verwenden.

Aufgabe 15 Rodriguez in Rotation (5 Punkte)

Beweist, dass nach der Definition aus der Vorlesung die unten stehende Rodriguez Formel eine Rotation eines Punktes p um die Achse $(x, y, z)^T, x^2 + y^2 + z^2 = 1$ mit Winkel α darstellt.

$$\operatorname{Rot}\left(\begin{pmatrix} x \\ y \\ z \end{pmatrix}, \alpha, p\right) = \begin{pmatrix} (1-c)x^2 + c & (1-c)xy - sz & (1-c)xz + sy \\ (1-c)xy + sz & (1-c)y^2 + c & (1-c)yz - sx \\ (1-c)xz - sy & (1-c)yz + sx & (1-c)z^2 + c \end{pmatrix} p, \tag{1}$$

$$\text{mit} \quad c = \cos\alpha, \quad s = \sin\alpha \tag{2}$$

Tipp: Man kann sich die Schreibarbeit vereinfachen, indem man ausnutzt, dass eine in p lineare Gleichung für alle p gilt, wenn sie nur für $p = (1, 0, 0)^T$, $p = (0, 1, 0)^T$ und $p = (0, 0, 1)^T$ gilt.

Aufgabe 16 Was peilt der Kohlenkarren in 2D (5 Punkte)

Betrachtet noch einmal den zweidimensionalen Kohlenkarren aus Aufgabe 13. Wenn Eure Lösung funktioniert, könnt Ihr sie verwenden, ansonsten die Musterlösung. Beobachtet, wie sich Positions- und Orientierungsunsicherheit verändern, wenn der Karren umherfährt, besonders, wenn längere Zeit keine Messung verfügbar ist.

Schreibt aus Euren Beobachtungen einen kleinen Aufsatz (ca. 1 Seite), in dem Ihr erklärt, wie die Unsicherheit entsteht, welche Struktur sie hat, wie sie von der Bewegung abhängt, und was allgemein an der Unsicherheit der Karrenposition/orientierung bemerkenswert ist.

Schreibt den Text für einen Leser, der keine Vorkenntnisse in Sensorfusion hat. Ihr könnt Eure Beobachtungen mit dem Programm, Erkenntnisse aus den Gleichungen, oder anschauliche Argumente über fahrende Karren verwenden.

Aufgabe 17 Der Kohlenkarren fliegt! (10 Punkte)

Zielsetzung

In dieser Aufgabe soll ein Unscented Kalman Filter (UKF) die Orientierung (nicht Position) eines Körpers (des fliegenden Kohlenkarren) in 3D verfolgen. Der Kohlenkarren ist mit einem sogenannten Inertialsensor ausgestattet, der aus einem 3D-Gyrometer und einem 3D-Akzelerometer besteht.

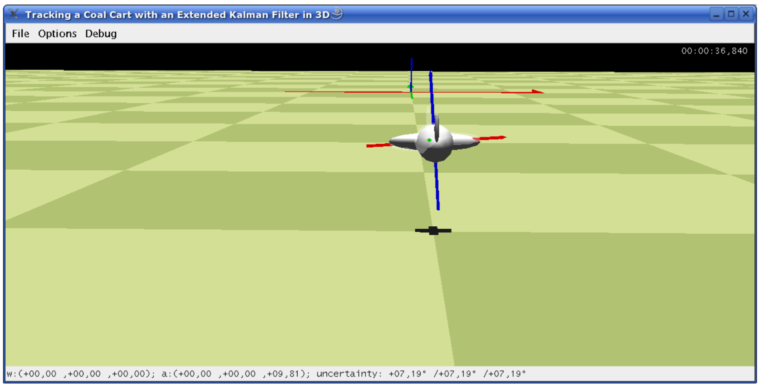

Abbildung 1: Screenshot aus dem Simulationsrahmen `CoalCart3D.java`. Im Bild sieht man den wahren Kohlenkarren mit Flügeln und an der selben Position transparent überlagert die Schätzung der Orientierung des Kohlenkarren aus dem Unscented Kalman Filter. Die Koordinatensysteme von Welt und Schätzung sind angegeben (Rot-X, Grün-Y, Blau-Z)

Das Gyrometer misst, wie sich die Orientierung des Körpers ändert. Sein Messwert ist ein Vektor in Körperkoordinaten, dessen Richtung die Achse der momentanen Drehung des Körpers angibt. Seine Länge gibt die Drehgeschwindigkeit um jene Achse in Radian/s an. Durch Akkumulation dieser Information im Dynamikschritt soll der UKF die Orientierung des Körpers verfolgen.

Nehmen wir zuerst an, der Körper würde sich nur drehen und seine Position nicht ändern. Das Akzelerometer misst die auf den Körper einwirkende Beschleunigung als 3D-Vektor in Körperkoordinaten in m/s^2. Für einen Körper in Ruhe ist dies ausschließlich die negative Erdbeschleunigung, d.h. ein Vektor von $9.81 m/s^2$ Länge, der nach *oben* zeigt. Damit gibt der Messwert absolute Information über die Orientierung des Körpers. Er kann im Messschritt ausgewertet werden, um die wachsenden Fehler aus der Akkumulation der Gyrometer zurückzuführen.

In dem Szenario dieser Aufgabe steht der Körper allerdings nicht still, sondern bewegt sich. Dadurch überlagern die Beschleunigungen aus der Bewegung die Information aus der Erdanziehung. Trotzdem können die Akzelerometer als Informationsquelle für "oben" benutzt werden, indem man die echten Beschleunigungen als ziemlich grosses "Rauschen" auf den Messwerten der Akzelerometer betrachtet. Das ist zwar nicht strikt richtig (warum?), funktioniert aber, weil bei beschränkter Geschwindigkeit im zeitlichen Mittel die Gravitation die Messwerte der Akzelerometer dominiert.

Implementiert nach dieser Idee einen Unscented Kalman Filter zur Schätzung der Orientierung des Kohlenkarrens.

Rahmen

Verwendet den Rahmen `CoalCart3D.java` (Abb. 1) und programmiert den UKF in der Klasse `UnscentedKalmanFilter3D.java`. Parameter finden sich in `CoalCartParameter.java`. Der fliegende Kohlenkarren wird gesteuert mit `w` / `s` für Beschleunigen / Abbremsen und den Pfeiltasten für Rollen und Nicken. Der Blickwinkel kann durch Ziehen mit der Maus und den drei Maustasten verändert werden. Das Programm kann kontinuierlich (25 Schritte je Sekunde) oder mit einzelnen Schritten auf Tastendruck betrieben werden (Taste: `ENTER`, Menü `Options/run step by step P`). Bei jedem Schritt wird die Gyrometermessung an den UKF als Dynamikschritt weitergereicht. Die Akzelerometer-Messwerte werden auf Tastendruck (Leertaste) oder automatisch (Menüpunkt `Options/Auto measure A`) als Messung an den UKF weitergereicht.

Ihr könnt Euch die in der Kovarianz repräsentierte Unsicherheit anzeigen lassen (`Options/Show sampled posterior uncertainty U`). In diesem Modus wird statt μ im schnellen Wechsel eine zufällig aus $N(\mu, \Sigma)$ gezogene Orientierung gezeigt. Stärke und Art des Springens der angezeigten Orientierung zeigen dann die repräsentierte Verteilung.

Der Algorithmus ist nicht groß, aber schwierig und voller Möglichkeiten, schwer zu erkennende Fehler zu programmieren. Geht deshalb schrittweise vor und testet möglichst kleine Teile nacheinander. Überlegt Euch immer kritisch, ob das Ergebnis wirklich das ist, was man erwartet. Konkrete Tipps:

- Arbeitet zuerst im Einzelschrittmodus (Menü: `Options/run step by step`).
- Programmiert zuerst die `State.plus` $\boxplus$ und `State.minus` $\boxminus$ Routinen und testet, dass $S_1 \boxplus (S_2 \boxminus S_1) = S_2$ mit dem Menüpunkt `Debug/Check State.plus State.minus`.
- Programmiert das Dynamikmodell und implementiert den Dynamikschritt als einfaches Anwenden der Modellfunktion auf μ.
- Programmiert die Sigma-Punkt-Konversionen und testet sie mit dem Menüpunkt `Debug/Check mean/covariance <---> Sigma point conversion`. Nach Hin- und Rückkonversion müssen ursprünglicher Mittelwert und Kovarianz herauskommen.
- Programmiert und testet den Dynamikschritt noch ohne Messrauschen (`Options/Active measurement noise N`) und überprüft, ob sich das System so verhält, wie Ihr erwartet.
- Schaltet nun das Messrauschen an.
- Programmiert jetzt den Messschritt mit einzelnen Messungen (Leertaste). Probiert ihn zuerst ohne Messrauschen. Im Stillstand (!) sollte die Innovation 0 sein.
- Aktiviert nun das Messrauschen.
- Aktiviert nun automatische Messungen.
- Achtung: Im Normalfall ist der Mittelwert ein Vektor und die Kovarianz eine Matrix. In dem speziellen UKF zur Schätzung von Orientierungen (und anderen allgemeinen Strukturen) ist die Kovarianz eine Matrix, aber der Mittelwert ein allgemeiner Zustand. In diesem Fall ist der Mittelwert also eine Starrkörpertransformation und damit auch eine 3×3 Matrix.

Aufgabe 18 Drehungen in 4D (2 Bonuspunkte)

In drei Dimensionen lässt sich jede Starrkörpertransformation als Drehung um eine Achse darstellen. Dies werden wir in der letzten Vorlesung beweisen. Zeigt hier, dass das 4-dimensionale Analogon dieser Aussage nicht gilt.

Aufgabe 19 Gemeinsam drehen und sich doch nicht näher kommen (2 Bonuspunkte)

Beweist aus den Rotationsaxiomen, dass $p \mapsto \mathrm{Rot}(v, \alpha, p)$ eine Isometrie ist.