Theorie der Sensorfusion

U. Frese

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

Übungsblatt 3

Abgabe: Mo, 14.01.2013, 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 11 Axiomatische Schnipsel (5 Punkte)

Beweist die folgenden Eigenschaften von Erwartungswerten und Vektoren und Matrizen sowie der Kovarianzmatrix von Vektoren. Tipp: Oft (!) ist ein Rückführen auf den eindimensionalen Fall durch Betrachten der einzelnen Komponenten ein guter Ansatz. Manchmal funktioniert ein Betrachten der Matrix- bzw. Vektorterme selbst besser. Die Liste erscheint lang, fast alle Beweise sind aber Einzeiler. Die meisten der Eigenschaften von $\mathrm{Cov}(X)$ folgen direkt aus den Eigenschaften von $\mathrm{Cov}(X,Y)$ mit $Y = X$. Dann reicht ein kurzer Verweis auf die entsprechende Eigenschaft, z.B. „nach (17) mit $Y = X$“.

1. $\mathrm{E}(U+V) = \mathrm{E}(U) + \mathrm{E}(V)$ für $U, V : \Omega \to \mathbb{R}^{m\times n}$
2. $\mathrm{E}(\lambda U) = \lambda\,\mathrm{E}(U)$ für $U : \Omega \to \mathbb{R}^{m\times n}, \lambda \in \mathbb{R}$
3. $\mathrm{E}(A) = A$ für $A \in \mathbb{R}^{m\times n}$
4. $\mathrm{E}(U^T) = \mathrm{E}(U)^T$ für $U : \Omega \to \mathbb{R}^{m\times n}$
5. $\mathrm{E}(UV) = \mathrm{E}(U)\,\mathrm{E}(V)$ für $U : \Omega \to \mathbb{R}^{m\times n}, V : \Omega \to \mathbb{R}^{n\times p}$, unabhängig

6. $\mathrm{Cov}(X,Y) = \mathrm{E}(XY^T) - \mathrm{E}(X)\,\mathrm{E}(Y)^T$ für $X : \Omega \to \mathbb{R}^n, Y : \Omega \to \mathbb{R}^m$
7. $\mathrm{Cov}(X) = \mathrm{E}(XX^T) - \mathrm{E}(X)\,\mathrm{E}(X)^T$ für $X : \Omega \to \mathbb{R}^n$
8. $\mathrm{Cov}(\lambda X, \mu Y) = \lambda\mu\,\mathrm{Cov}(X,Y)$ für $\lambda, \mu \in \mathbb{R}, X : \Omega \to \mathbb{R}^n, Y : \Omega \to \mathbb{R}^m$
9. $\mathrm{Cov}(\lambda X) = \lambda^2\,\mathrm{Cov}(X)$ für $\lambda \in \mathbb{R}, X : \Omega \to \mathbb{R}^n$
10. $\mathrm{Cov}(X,Y) = \mathrm{Cov}(Y,X)^T$ für $X : \Omega \to \mathbb{R}^n, Y : \Omega \to \mathbb{R}^m$
11. $\mathrm{Cov}(X) = \mathrm{Cov}(X)^T$ für $X : \Omega \to \mathbb{R}^n$
12. $\mathrm{Cov}(X+Y, Z) = \mathrm{Cov}(X,Z) + \mathrm{Cov}(Y,Z)$ für $X, Y : \Omega \to \mathbb{R}^n, Z : \Omega \to \mathbb{R}^m$
13. $\mathrm{Cov}(X, Y+Z) = \mathrm{Cov}(X,Y) + \mathrm{Cov}(X,Z)$ für $X : \Omega \to \mathbb{R}^n, Y, Z : \Omega \to \mathbb{R}^m$
14. $\mathrm{Cov}(X+Y) = \mathrm{Cov}(X) + \mathrm{Cov}(Y) + \mathrm{Cov}(X,Y) + \mathrm{Cov}(X,Y)^T$ für $X, Y : \Omega \to \mathbb{R}^n$
15. $\mathrm{Cov}(X,Y) = 0$ für $X : \Omega \to \mathbb{R}^n, Y : \Omega \to \mathbb{R}^m$ unabhängig
16. $\mathrm{Cov}(X+Y) = \mathrm{Cov}(X) + \mathrm{Cov}(Y)$ für $X, Y : \Omega \to \mathbb{R}^n$ unabhängig
17. $\mathrm{Cov}(AX, BY) = A\,\mathrm{Cov}(X,Y)B^T$ für $X : \Omega \to \mathbb{R}^n, Y : \Omega \to \mathbb{R}^m, A \in \mathbb{R}^{p\times n}, B \in \mathbb{R}^{q\times m}$
18. $\mathrm{Cov}(AX) = A\,\mathrm{Cov}(X)A^T$ für $X : \Omega \to \mathbb{R}^n, A \in \mathbb{R}^{p\times n}$

Aufgabe 12 Eine Gegenposition (5 Punkte)

Was wäre, wenn man für $X : \Omega \to \mathbb{R}^n$ statt der Kovarianzformel $\mathrm{Cov}(X) = \mathrm{E}\big((X - \mathrm{E}(X))(X - \mathrm{E}(X))^T\big)$ umgekehrt definiert hätte

$$\mathrm{Q}(X) = \mathrm{E}\big((X - \mathrm{E}(X))^T(X - \mathrm{E}(X))\big)? \tag{1}$$

Formal ist der Term gültig, und er stellt ja auch eine n-dimensionale Generalisierung des Produktes im eindimensionalen Fall dar. Untersucht im Einzelnen die Eigenschaften dieses Begriffes (analog zu Aufgabe 11) und bringt abschliessend in einem kurzen Text den Unterschied auf den Punkt. Lässt sich eine explizite Relation zwischen $\mathrm{Q}(X,Y)$ und $\mathrm{Cov}(X,Y)$ angeben?

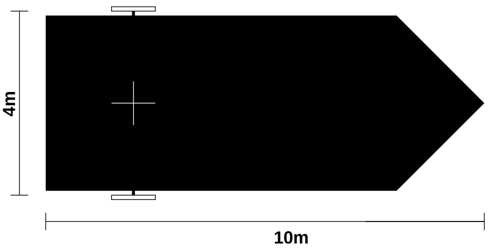

Abbildung 1: Kinematik und Aufbau des Kohlenkarrens. Der weisse Punkt in der Mitte ist der Referenzpunkt des Fahrzeuges, in dem sich auch der Peilsensor befindet. Die Räder befinden sich oberhalb und unterhalb des Peilsensors, mit einer Spurbreite von $4m$.

Aufgabe 13 Peil den Kohlenkarren 2D (10 Punkte)

Ein $10m$ langer und $4m$ breiter Kohlenkarren (Abb. 1) bewegt sich auf einem $100m \times 100m$-Platz umher. An den beiden Hinterrädern sind Drehsensoren angebracht, die die Strecke messen, welche das jeweilige Rad zurücklegt. Außerdem befindet sich über der Mitte der Hinterachse ein Peilgerät, das den horizontalen Winkel zu drei auf dem Platz befindlichen Baken misst. Der Sensor sieht nur Baken, die vor dem Karren $\pm 60°$ liegen; welche Bake gesehen wird, kann sich also ändern.

Implementiert einen Extended Kalman Filter, der Position und Orientierung des Karrens auf dem Platz verfolgt. Verwendet dazu die Vorlage `KalmanFilter2D.sample.java` (Abb. 2). Die Parameter, insbesondere Masse des Karrens, Rauschwerte und die Positionen der Baken finden sich in `CoalCartParameter.java`. Für die Radsensoren gilt dasselbe bewegungsabhängige Rauschen wie in Aufgabe 8. Der Programmrahmen ruft die Funktionen `KalmanFilter2D.initialize`, `KalmanFilter2D.dynamic` und `KalmanFilter2D.measurement` automatisch auf. Die letztere Routine wird für jede gepeilte Bake einzeln aufgerufen, nicht mit einem Vektor aller Peilungen als eine Messung.

Der Rahmen verwendet das `Jama` Matrix Paket (in der `.zip` Datei enthalten). Dessen Dokumentation findet sich unter `http://math.nist.gov/javanumerics/jama/doc/`.

Tipps:

- Implementiert zuerst einmal Dynamik und Messmodell. Ihr könnt ganz ohne Kovarianz (`covX=null` lassen) und die EKF-Formeln das System laufen lassen. Wenn Ihr dann das Messrauschen abschaltet (Options/Activate Measurement Noise) sollte der geschätzte Karren exakt dem wahren Karren folgen.
- Für das Dynamikmodell überlegt zuerst, wie viel sich die beiden Räder drehen, wenn der Karren eine bestimmte Strecke nach vorne fährt oder sich um einen bestimmten Winkel dreht. Für Kurvenfahrten würde der Karren eigentlich einer Kreisbahn folgen. Ihr könnt aber in Näherung annehmen, dass jede Messung von einer kurzen Geradeausfahrt gefolgt von einer kleinen Drehung stammt.
- Beachtet, dass die Messungen des Peilsenders Winkel sind und Winkel eine 2π-Periodizität haben.
- Ruft die `checkConsistency`-Routine auf. Sie überprüft die Kovarianz auf Format, Symmetrie und positive Definitheit. Schlägt sie fehl, ist meist eine der Matrixformeln falsch.
- Oft stecken Fehler in den Jacobi-Matrizen. Ihr könnt dies überprüfen, indem Ihr `Debug/...` als Menüpunkt auswählt. Dann werden die jeweiligen Jacobis sowohl numerisch also auch über Eure Routine berechnet, und Ihr könnt die Ergebnisse vergleichen. Sie müssen nicht exakt, aber ungefähr gleich sein (Fehler < 0.001).

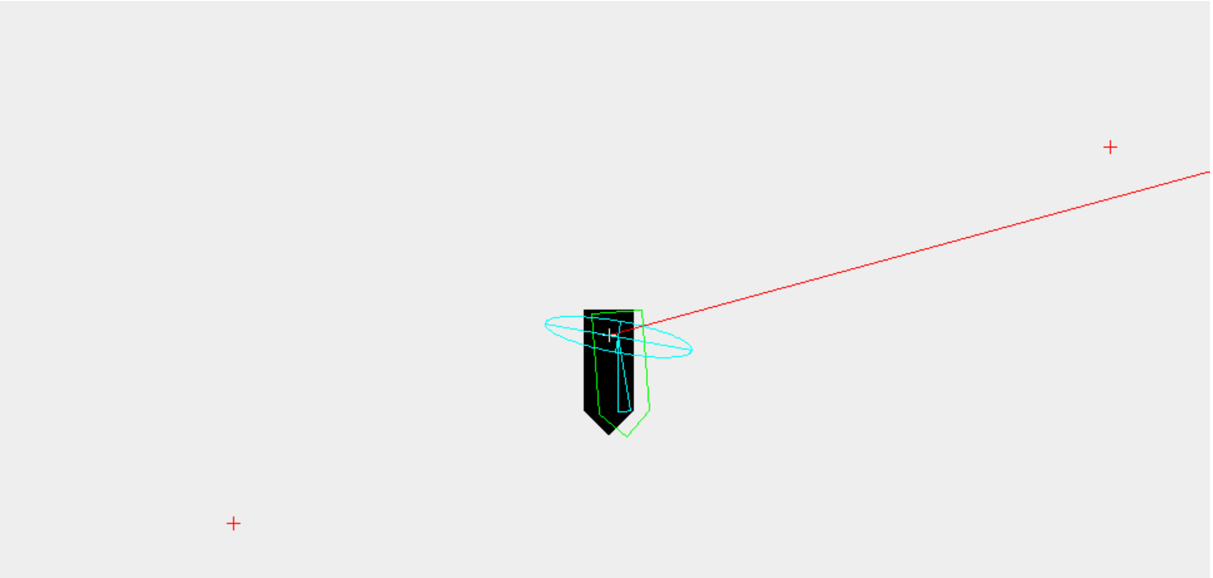

Abbildung 2: Screenshot aus dem Simulationsrahmen `CoalCart2D.java`. Der schwarze Kasten zeigt die wahre Position und Orientierung des Kohlenkarrens mit der Spitze nach vorne. Die roten Kreuze sind die Baken, die der Karren peilen kann. Wenn eine Bake gepeilt wird, wird die Messung als roter Strahl gezeigt. Der grüne Rahmen zeigt die geschätzte Position und Orientierung (`muX`). Das türkise Kreisstück zeigt die Winkelunsicherheit (`covX`) an, die türkise Ellipse die Positionsunsicherheit. Die Korrelation zwischen X und Y zeigt sich aus der Lage der Ellipse, die Korrelation mit der Orientierung ist graphisch nicht zu sehen.

- Wer ohne Debugger programmiert, ist selbst schuld! Eine schöne Java-Umgebung und weitere Tipps finden sich unter `http://www.informatik.uni-bremen.de/~roefer/pi2-06/`.

Aufgabe 14 Vom Maximum zum Minimum (3 Bonuspunkte)

Sei A eine symmetrische Matrix (z.B. eine Kovarianzmatrix). Seien für $x, y \in \mathbb{R}^n$ mit $||x|| = ||y|| = 1$ der Wert von $x^T Ax$ maximal und der von $y^T Ay$ minimal (im Falle einer Kovarianzmatrix sind dies die Richtungen mit der größten bzw. geringsten Unsicherheit). Zeige, dass x und y Eigenvektoren von A sind, d.h. es gilt

$$Ax = \lambda x \text{ und } Ay = \mu y$$

für geeignete $\lambda, \mu \in \mathbb{R}$. Zeige ferner für diese λ, μ, dass $x^T Ax = \lambda$, $y^T Ay = \mu$. Folgere, dass $\lambda \neq \mu$, falls sich Maximum und Minimum unterscheiden (hierfür gibt es aber keinen Punkt...) und dass dann $x \perp y$ (d.h. $x^T y = 0$).

Hinweis: der erste Teil der Aufgabe (Extrema unter Nebenbedingungen) benötigt die Methode der Lagrangeschen Multiplikatoren. Eine gute Einführung in diese Methode bietet der englische (!) Wikipedia-Artikel zu *Lagrange multipliers*.