Ausgabe am **25.05.2021**
Abgabe bis zum **06.06.2021 23:59 Uhr**

UNIVERSITÄT OSNABRÜCK

Priv. Doz. Dr. Thomas Wiemann, Alexander Mock

Robotik
Übungsblatt 4

Sommersemester 2021

Aufgabe 4.1 (Bayes-Filter) **(5P)**

Stellen Sie sich vor, Sie seien in einem unterirdischen Hochsicherheitskomplex als Wachposten beschäftigt (je nach persönlicher Präferenz für einen Geheimdienst oder einen Schurken). Auf Grund der herrschenden Sicherheitsvorschriften beschränkt sich Ihr Kontakt zur Außenwelt auf das morgentliche Kommen und das abendliche Gehen des „Direktors“.

Um für ein bisschen Abwechslung in Ihrem Arbeitsalltag zu sorgen (und Ihr unverschämt gutes Gehalt noch weiter aufzubessern), beschließen Sie und ein (Leidens-) Genosse, auf das aktuelle Wetter R_t (Regen oder Nicht-Regen) zu wetten.

Ihre scharfe Beobachtungsgabe und Ihr in der Vorlesung erworbenes Wissen über Bayes-Filter können Sie nun gewinnbringend einsetzen: Anhand der Tatsache, ob der „Direktor“ morgens mit einem Regenschirm U_t (für „umbrella“) erscheint oder nicht, ist es ihnen möglich, Rückschlüsse über die aktuelle Wetterlage zu ziehen. Zudem erscheint es Ihnen plausibel, dass es wahrscheinlicher ist, dass es an einem Tag regnet, wenn es bereits am Vortag geregnet hat. Dieses Wissen ist in Tabelle 1 zusammengefasst.

R_{t-1}	$P(R_t)$
t	0.7
f	0.3

(a)

R_t	$P(U_t)$
t	0.9
f	0.2

(b)

Tabelle 1: Hintergrundwissen zur Wettervorhersage: **(a)** Wahrscheinlichkeit für Regen, wenn es am Vortag geregnet hat; **(b)** Wahrscheinlichkeit, dass ein Regenschirm mitgenommen wird unter der Bedingung, dass es regnet

Initial ($t = 0$) gehen Sie davon aus, dass Regen und Nicht-Regen (bzw. Sonne) gleich wahrscheinlich sind.

(a) Berechnen Sie die a-priori Wahrscheinlichkeitsverteilung für den ersten Tag, an dem Sie Ihre Beobachtungen starten ($t = 1$).

(b) Ausgehend von dieser Verteilung machen Sie die Beobachtung u_1 (der „Direktor“ kommt mit dem Regenschirm). Wie sieht unter Berücksichtigung dieser Information die a-posteriori Wahrscheinlichkeitsverteilung für Tag 1 aus?

(c) Berechnen Sie nun davon ausgehend die Wahrscheinlichkeitsverteilung $P(R_2 \mid u_1)$, also die a-priori Wahrscheinlichkeit, dass es an Tag 2 regnet, wenn Sie an Tag 1 den Regenschirm gesehen haben.

(d) Wie sieht nun die a-posteriori Wahrscheinlichkeit $P(R_2 \mid u_1, u_2)$ aus, wenn Sie den Regenschirm an Tag 2 erneut beobachten?

(e) Nehmen Sie nun an, dass der Regenschirm auch an allen Tagen nach Tag 2 zu sehen ist. Berechnen Sie den Fixpunkt, gegen den die Wahrscheinlichkeit für Regen unter dieser Annahme konvergiert.

(f) Betrachten Sie nun den Fall, dass Sie den „Direktor“ nach Tag 2 nicht mehr sehen. Welche Vorhersage für die Wahrscheinlichkeit von Regen können Sie nun für Tag 3 und 4 machen? Berechnen Sie auch hier den Fixpunkt, gegen den die Wahrscheinlichkeit für Regen konvergiert, wenn die Vorhersage ohne zusätzliche Beobachtungen weiter in die Zukunft gelegt wird.

Aufgabe 4.2 (Modellierung mit Kalmanfilter) (7P)

Für einen hypothetischen mobilen Roboter bestehe der Zustandsraum aus folgenden Komponenten:

- die Koordinaten x, z
- die Orientierung θ
- der Luftdruck in Osnabrück, λ
- die Anzahl der deutschen Bundesländer, β

Der Roboter hat die beiden möglichen Aktionen:

- $rot(\Delta\theta)$: im Stand rotieren
- $tra(\Delta x, \Delta z)$: um Δx und Δz bewegen (bei Beibehaltung der Orientierung bis auf Rauschen, s.u.), der Roboter ist also *holonom* in der Ebene (aber **nicht** in seinem Zustandsraum).

Unabhängig von den Aktionen des Roboters kann sich der Luftdruck von einem zum nächsten Zustand von selbst mit einer Standardabweichung von 0.001 hPa ändern. Die Roboteraktionen ändern weder den Luftdruck, noch die Anzahl der Bundesländer. Der Roboter kann die aktuellen Werte von x, z, θ und λ messen. Definieren Sie sich für diese Messungen ein geeignetes Sensormodell Σ_z.

Die Aufgabe besteht nun darin, mit einem Kalman-Filter den Zustand nach einer Rotation und Translation des Roboters abzuschätzen. Im Einzelnen:

(a) Geben Sie die Matrizen **A**, **B** und **H** für den Kalman-Filter für die beschriebene Domäne an! (2P)

(b) Modellieren Sie die Aktionen rot und tra:

- rot rotiert um den vorgegebenen Wert bei einer Standardabweichung von 0.02 und lässt x, z unverändert.
- tra translatiert den Roboter um die vorgegebenen Werte bei einer Standardabweichung von 0.5 in beiden Dimensionen. Als Nebeneffekt ändert es möglicherweise die Orientierung mit einer Standardabweichung von 0.01

Hinweis: Modellieren Sie hier so, dass sie für beide Aktionen die gleiche Matrix **B** verwenden und der Aktionsvektor **u** entsprechend der jeweiligen Aktion gefüllt wird. (2P)

(c) Im Startzustand befinde sich der Roboter an $(0, 0, 0)$, der Luftdruck sei 1000 hPa, und die Anzahl der Bundesländer sei 16. Berechnen Sie den Zustand (mehrdimensionaler Mittelwert und Standardabweichung), der sich ergibt, wenn der Roboter erst um 0.3 rad rotiert und anschließend um 2 Einheiten x und 3 Einheiten z translatiert. Geben Sie für alle vorkommenden Zustände die a-priori-Werte für Mittelwert und Varianz sowie die entsprechenden Kalman-Gewinnmatrizen an. (3P)

Aufgabe 4.3 (Fusion von Sensordaten) **(8P)**

In dieser Aufgabe werden Sie selbst einen Kalmanfilter analog zur Vorlesung implementieren, um Sensordaten eines Roboters zu fusionieren und damit eine möglichst genaue relative Lokalisierung zu ermöglichen.

(a) Die Pose ihres Roboters soll zur Vereinfachung in der xy-Ebene liegen: (x,y,θ). Überlegen Sie sich, wie Ihr Zustand in diesem Fall aussehen kann. Nun bekommen Sie von zwei Sensoren Informationen über die Roboterbewegung:

- IMU. Topic: `/imu/data`
- Odometrie. Topic: `/odom`

Erklären Sie, was die Felder beider Messages beschreiben. Kategorisieren Sie die Felder zusätzlich mithilfe der folgenden Tabelle:

	Ort	Geschwindigkeit	Beschleunigung
Lin. Bewegung			
Drehung			

(2P)

(b) Subscriben Sie sich auf die beiden Topics, sodass Ihnen im Callback eine aktuelle Odometrie und IMU-Message zur Verfügung steht. Implementieren Sie nun - gemäß der Vorlesungsfolien - einen linearen Kalman-Filter, der den Zustand Ihres Roboter gegeben der Sensordaten anpassen soll. Tipp: Sie können für Matrix-Berechnungen die Bibliothek `Eigen` verwenden. Diese ist bereits mit ROS auf Ihrem System installiert.

Publishen Sie die geschätzte Pose als globale Transformation über TF. Das Resultat können Sie sich in Rviz angucken. Stellen Sie dazu den `Fixed Frame` in RViz auf den TF-Frame, über das Sie Ihre Ergebnisse publishen. Stellen Sie auch die DecayTime des LaserScans hinreichend groß ein ($> 10\,\mathrm{s}$). Ist Ihre Implementierung korrekt, sollten die verschiedenen Scans einigermaßen konsistent zusammengesetzt werden. (6P)