

Praktikum
Forschungsprojekt
Anthropomatik praktisch
erfahren
5. Februar 2018

He Jingyu, Martin Miller
Roboterlokalisierung in 2D mittels SE(2) Filterung

Roboterlokalisierung in 2D mittels SE(2) Filterung

– Praktikum: Forschungsprojekt Anthropomatik praktisch erfahren–

He Jingyu, Martin Miller

5. Februar 2018

Zusammenfassung

Einleitung

In der Robotik spielt die Lokalisierung des Roboters eine große Rolle. Dabei sollen die Position sowie die Orientierung des Roboters erfasst werden. In einem zweidimensionalen Szenario kann man die Pose als Element der speziellen Euklidischen Gruppe SE(2) auffassen, welche alle Kombinationen von Rotationen und Translationen in 2D enthält. Um die Pose des Roboters basierend auf verrauschten Sensordaten zu schätzen, kommen häufig Kalman-Filter und ähnliche Verfahren zum Einsatz. Am ISAS wurde eine Methode entwickelt, welche das besondere Verhalten der SE(2), insbesondere die Periodizität der Orientierung, besser behandelt als traditionelle Verfahren.

Aufgabenstellung

In diesem Praktikumsprojekt soll das am ISAS entwickelte Verfahren zur SE(2) Schätzung in einem praktischen Szenario implementiert und evaluiert werden. Dazu kommt der am ISAS entwickelte Crawler zum Einsatz.

Inhaltsverzeichnis

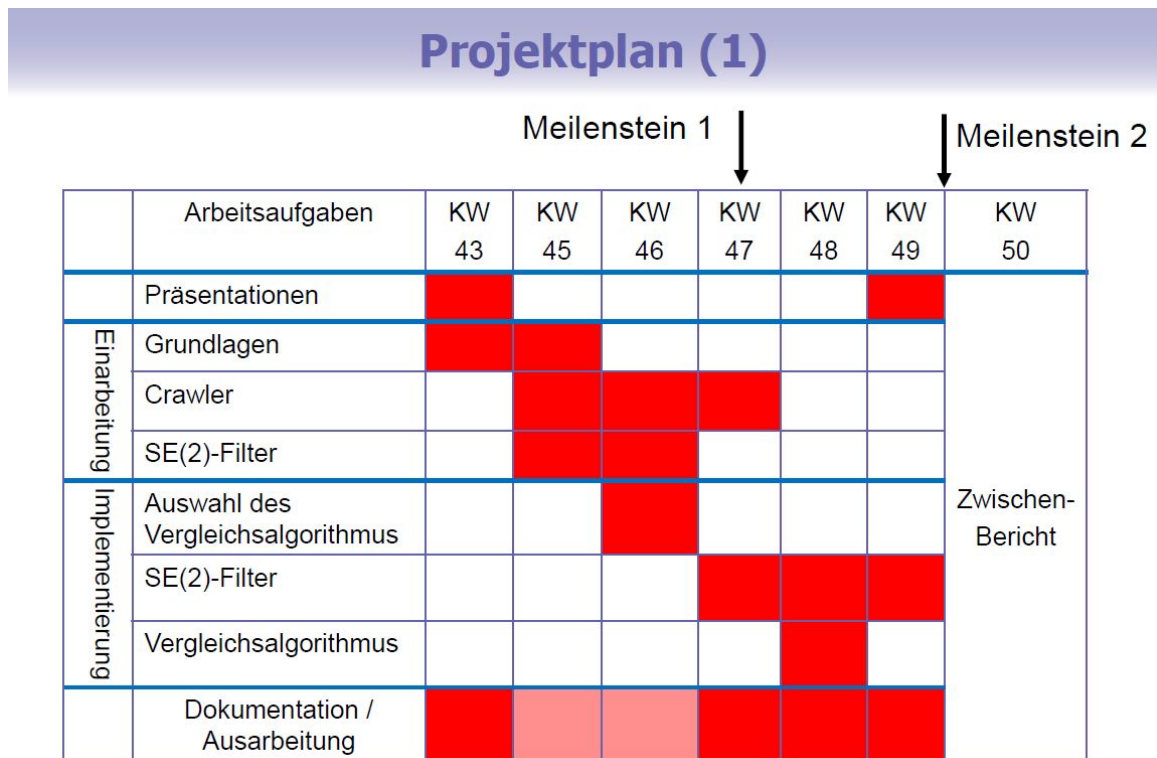
1	Einleitung	3
1.1	Aufgabenstellung	3
2	Projektplan	4
3	Kalman-filter	5
3.1	Prädiktionsschritt	6
3.2	Innovationsschritt	6
4	SE(2) Filter	8
4.1	Lie Gruppe	8
4.2	Bingham-Verteilung	8
4.3	Duale Quaternion	9
4.4	SE(2)Filter mit Crawler	9
5	Crawler	9
5.1	Aufbau	10
6	Simulation	11
6.1	Systemmodell	12
6.2	Messmodell	13
6.3	Progressive Updates	14
7	Evaluation	14
7.1	Ergebnisse	14
8	Zusammenfassung und Ausblick	14
9	Das Filter XY	15
9.1	Grundlagen des Filters XY	15

1 Einleitung

Der ISAS-Crawler (kurz: Crawler) ist ein mobiler Agent, der am Institut für Intelligente Sensor-Aktor-Systeme (ISAS) entwickelt wird. Dieser besitzt drei Beine, mithilfe derer er sich fortbewegen kann und vier Ultraschallsensoren, die für die Erkundung der Umwelt verwendet werden können. Weitere Details bzgl. des Aufbaus des Crawlers können Abschnitt 3 entnommen werden. Jedoch existiert noch keine Software, die es ermöglicht, den Crawler zu steuern und die von ihm gelieferten Messdaten zu verwerten. Ein Ansatz, mithilfe dessen diese Problematik gelöst werden kann, wird als SLAM bezeichnet. SLAM steht für Simultane Lokalisierung und Kartographierung (engl. simultaneous localization and mapping) und ermöglicht es, eine Karte der Umwelt zu erstellen, sowie sich selbst in dieser zu lokalisieren. Verwendung findet es vor allem in der Robotik (s. [4], [5]), da es die Grundlage für ein autonomes Verhalten bietet. Ziel dieser Arbeit ist der Entwurf sowie die Umsetzung eines SLAM Verfahrens für den o.g. Crawler und die Erstellung einer Simulationsumgebung für dieses System. ISAS.

1.1 Aufgabenstellung

Einarbeitung in die nötigen Grundlagen (Bayes-Filterung, duale Quaternionen, Bingham Verteilung) Einarbeitung in das am ISAS entwickelte Verfahren Entwurf geeigneter System- und Messmodelle unter Verwendung von Vorarbeiten der letzten Semester Implementierung des Verfahrens und eines Vergleichsverfahrens (z. B. UKF oder Partikelfilter) Evaluation des Verfahrens in Simulationen und mit realen Daten

**Abbildung 1:** Zeitplanung1

2 Projektplan

Zu Beginn des Projektes wurde ein Zeitplan (Abbildung 1 und 2) zwecks besserer Koordinierung und Zeitmanagement erstellt. Hierbei wurden die einzelnen Aufgaben derart gewählt, dass sie zu zweit bzw. zu dritt bearbeitet werden können. Dabei wurde darauf geachtet, dass möglichst kein Stillstand entsteht, d.h. die Teams aus zwei bis drei Personen möglichst parallel arbeiten können. Bei der zeitlichen und organisatorischen Einteilung sind somit drei logische Blöcke entstanden:

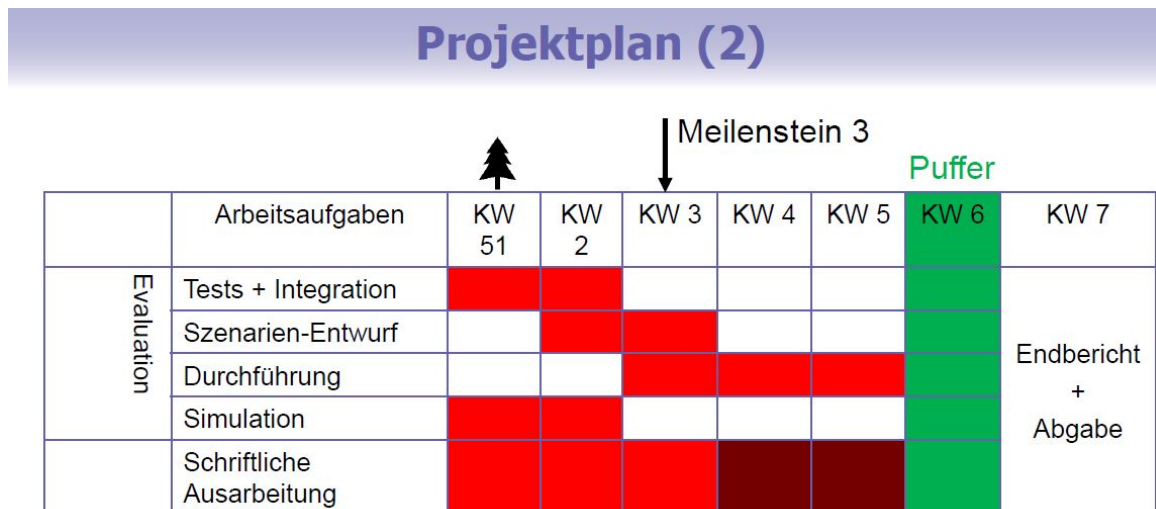


Abbildung 2: Zeitplanung2

3 Kalman-filter

Das Kalman-Filter ist ein mathematisches Verfahren. Benannt ist das Filter nach seinen Entdeckern Rudolf E. Kálmán, Richard S. Bucy und Ruslan L. Stratonovich, die das Verfahren unabhängig voneinander entdeckt bzw. wesentliche Beiträge dazu geliefert haben. Das Kalman-Filter dient dazu, Fehler in realen Messwerten zu reduzieren und Schätzungen für nicht messbare Systemgrößen zu liefern. Voraussetzung dabei ist, dass die interessierenden Werte durch ein mathematisches Modell beispielsweise in Form von Bewegungsgleichungen beschrieben werden können. Die Besonderheit des 1960 von Kálmán vorgestellten Filters bildet dabei seine spezielle mathematische Struktur, die den Einsatz in Echtzeitsystemen verschiedener technischer Bereiche ermöglicht. Dazu zählen u.a. die Auswertung von Radarsignalen oder GPS-Daten zur Positionsbestimmung sich bewegender Objekte (Tracking), aber auch der Einsatz in elektronischen Regelkreisen allgegenwärtiger Kommunikationssysteme wie etwa Radio und Computer.

Das KalmanFilter bestimmt wie das Bayes Filter seine Schätzungen rekursiv. Während das BayesFilter jedoch mit der Belief funktion jeweils die aktuelle Verteilung des Zustands bestimmt, werden beim KalmanFilter in jedem Schritt nur der Erwartungswert des Zustands und die Kovarianz dieser Schätzung bestimmt. Zur Initialisierung der Rekursion wird für $k = 0$ die Anfangsschätzung $\hat{X}_0 = X_0$ mit der Kovarianz P_0 gewählt. Da das Kalman Filter den Zustand eines Systems aus dem Eingangs- und Ausgangssignal schätzt, wird es als Zustandsschätzer bezeichnet. Das Kalman Filter bestimmt zu jedem Zeitpunkt den Erwartungswert und die Kovarianz der zu schätzenden Größe. Die Berechnungsvorschrift wird nachfolgend hergeleitet.

3.1 Prädiktionsschritt

Im Prädiktionsschritt soll die beste Schätzung von X_k und der zugehörigen Kovarianz aus $u_1, \dots, u_k, y_1, \dots, y_{k-1}$ und X_0, P_0 bestimmt werden. Außerdem ist die Schätzung des vorherigen Zeitschritts \hat{x}_{k-1} und deren Kovarianz P_{k-1} verfügbar, welche die beste erwartungstreue Schätzung von X_{k-1} unter Nutzung aller oben genannter Information außer u_k darstellt.

Der Erwartungswert des Systemzustands wird mit Hilfe der Systemgleichung bestimmt wobei der unbekannte Zustand X_{k-1} durch die aktuelle Schätzung \hat{x}_{k-1} ersetzt werden kann.

$$\hat{x}_k^- = EX_k = EAX_{k-1} + Bu_k + R_k \quad (3.1)$$

Die Kovarianz der Prädiktion lautet damit

$$P_k^- = Cov(X_k^-) = AP_{k-1}A^T + \Sigma_R \quad (3.2)$$

In der Umformung wurde genutzt, dass das Systemrauschen zum Zeitpunkt k unkorreliert mit Größen vorheriger Zeitpunkte ist. Der erste Summand der Kovarianz gibt die in Schritt k-1 bestehende Unsicherheit gewichtet mit der Systemdynamik A wieder. Durch den zweiten Summanden kommt noch das neue Systemrauschen R_k hinzu. Gleichungen (3.1) und (3.2) bestimmen somit, wie aus der Schätzung des Systemzustands und deren Kovarianz zum vorherigen Zeitpunkt k-1 eine Prädiktion derselben Größen für den Zeitpunkt k bestimmt werden kann.

3.2 Innovationsschritt

Im Innovationsschritt soll zusätzlich die aktuelle Beobachtung y_k berücksichtigt werden. In dieser Schritt wird erst die Verstärkungsmatrix (Kalman Gain) berechnet. Es lautet

$$K = \hat{P}_k^- C^T (C \hat{P}_k^- C^T + \Sigma_S)^{-1} \quad (3.3)$$

Entsprechend erfolgt die Innovation linear, als gewichtete Summe der aktuellen Schätzung und einer Innovation, die proportional zum aktuellen Beobachtungsresiduum ist

$$\hat{x}_k = \hat{x}_k^- + K(y_k - C\hat{x}_k^-) \quad (3.4)$$

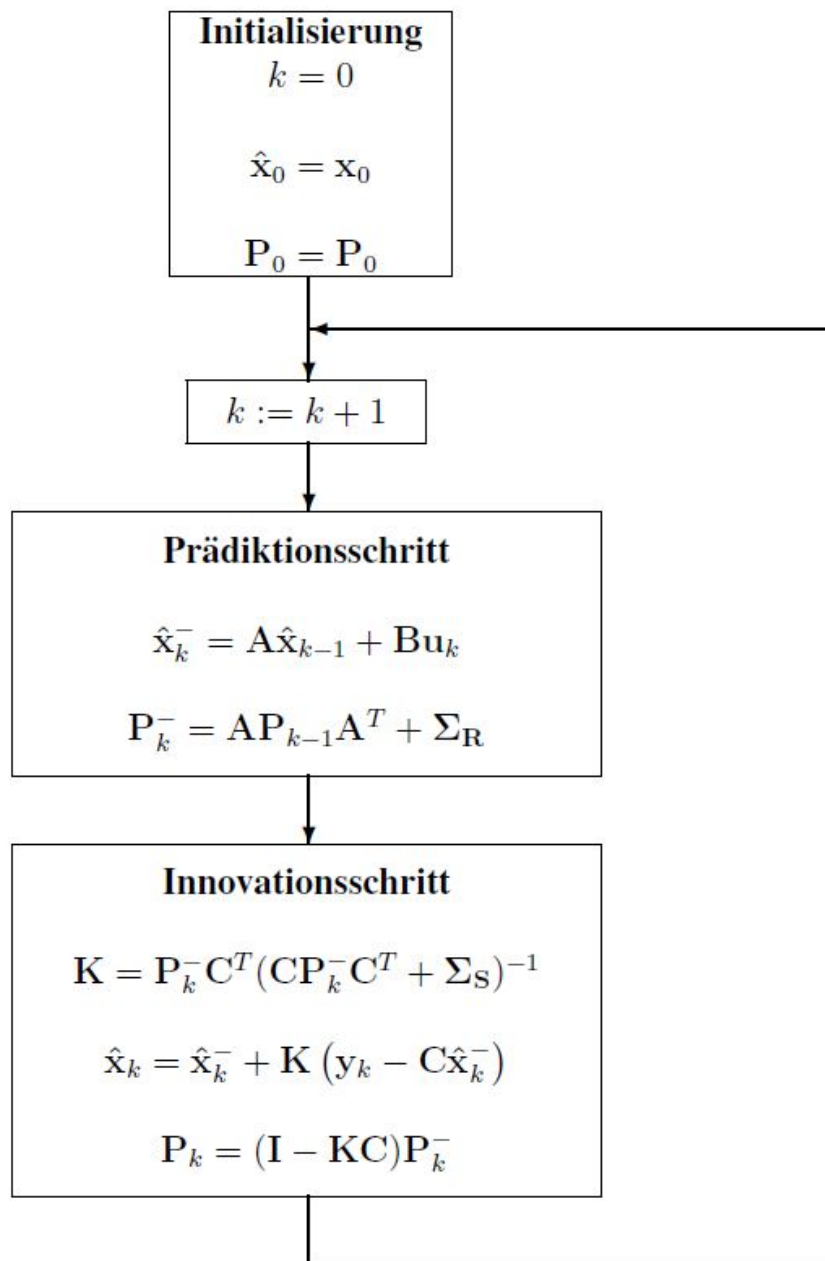
Unter Nutzung der Voraussetzung, dass das Beobachtungsrauschen S_k unkorreliert ist mit der Prädiktion, lässt sich die Kovarianz der Innovation berechnen gemäß

$$P_k = (I - KC)P_k^-(I - KC)^T + K\Sigma_S K^T \quad (3.5)$$

Wie noch gezeigt wird, lässt sich dieser Ausdruck für die optimale Verstärkungsmatrix des KalmanFilters K weiter vereinfachen zu

$$P_k = (I - KC)P_k^- \quad (3.6)$$

Die ganze KalmanFilter kann durch folgende Diagramm beschrieben werden.

**Abbildung 3:** Kalman Filter - Zusammenfassung

4 SE(2) Filter

4.1 Lie Gruppe

Eine Lie-Gruppe G ist eine Mannigfaltigkeit G , welche zusammen mit einer glatten Operation

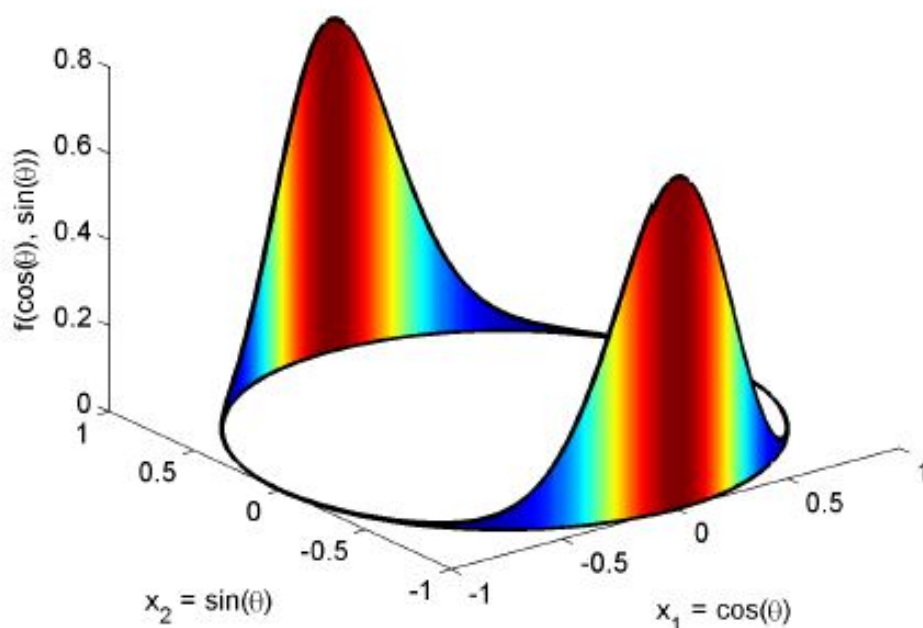
$$G \times G \ni (g_1, g_2) \rightarrow g_1 \cdot g_2 \in G$$

eine Gruppe ist.

Ein Vorteil der abstrakten Definition von Lie-Gruppe besteht darin, dass sie wesentlich flexibler ist. Zum Beispiel gilt folgendes: Ist G eine Lie-Gruppe und N ein abgeschlossener Normalteiler, dann ist G/N wie der eine Lie-Gruppe.

4.2 Bingham-Verteilung

In diesem Projekt leiten wir einen Filter, der auf der Bingham-Verteilung für zwei-dimensionale Zufallszahlen Vektoren der Längeneinheit basiert. Die Bingham-Verteilung kann zur Darstellung der Orientierungen auf der Ebene verwendet werden. Anstatt sich auf die auf Gaußsche-Verteilung basierende Annäherungen zu verlassen, haben wir uns entschieden, alle auftretenden Wahrscheinlichkeiten durch Bingham-Verteilung darzustellen. Die Bingham-Distribution ist definiert auf der Hypersphäre und ist antipodisch symmetrisch. Unsere Verwendung der Bingham-Distribution ist durch ihre Bequemlichkeit motiviert. Darstellung von hypersphärischen Zufallsvektoren,



In unsere Fall ist die Bingham-Verteilung einfacher für die Darstellung der Orientierung der Laufroboter. Als eine Folge der obigen Motivation kann man sehen, dass die Bingham-Wahrscheinlichkeitsdichte (pdf) sieht genau aus, wie sein zwei-Gaussian-Verteilungen die gegeneinander steht. Außerdem die Parametermatrix des Bingham-Verteilung, die in der Exponentialfunktion erscheint (was die Inverse ist Kovarianzmatrix im Gaußschen Fall) wird üblicherweise zerlegt in eine orthogonale und eine diagonale Matrix, die ein ergibt, intuitive Interpretation der Matrizen. Dies führt zu der folgende Definition.

$$f(\underline{x}) = \frac{1}{F} \cdot \exp(\underline{x}^T M Z M^T \underline{x}) \quad (1)$$

4.3 Duale Quantinion

In unsere Projekt wird Duale Quantinion zur Darstellung der Plannere rigid body motion verwendet. Die Duale Quantinion wird durch folgende Forme darstellt

$$\sigma = p + \epsilon q \quad (2)$$

Unter Ihnen die p steht für jeweilige Quantinion

$$p = p_0 + \underline{p} = p_0 + p_1 \underline{i} + p_2 \underline{j} + p_3 \underline{k} \quad (3)$$

$$q = q_0 + \underline{q} = q_0 + q_1 \underline{i} + q_2 \underline{j} + q_3 \underline{k} \quad (4)$$

Die Rotationsmatrix R , Transformationsmatrix t kann kompakt durch eine doppelte Quaternion dargestellt werden. Das Translationsvektor t ist eine reine Quaternion. Wir kombinieren es mit der Rotationsquaternion r in die folgende duale Quaternion

$$\sigma = r + \frac{\epsilon}{2} \underline{t} r = \cos \frac{\theta}{2} + \underline{\hat{u}} \sin \frac{\theta}{2} + \frac{\theta}{2} \left(-\sin \frac{\theta}{2} (\underline{t} \cdot \underline{\hat{u}}) + \cos \frac{\theta}{2} \underline{t} + \sin \frac{\theta}{2} \underline{t} \times \underline{\hat{u}} \right) \quad (5)$$

In unseren fall wird die Bewegung der Roboter hauptsächlich auf eine Eben durchgeführt werden. Deshalb stehen $\underline{t}_x \underline{t}_y$ für Translation der Roboter und θ für die Rotation der Roboter. Um Rotationen und Translationen für 2D zu kombinieren, wird hier duale Quaternion-Multiplikation verwendet. Eine Rotation mit eine folgende Translation wird durch die Multiplikation beschrieben

$$\left[1 + \epsilon \frac{1}{2} (\underline{t}_x \underline{i} + \underline{t}_y \underline{j}) \right] \cdot \left[\cos \left(\frac{\alpha}{2} \right) + \sin \left(\frac{\alpha}{2} \right) \underline{k} \right] \quad (6)$$

4.4 SE(2)Filter mit Crawler

5 Crawler

In unsere Projekt wird ein Laufroboter gebraucht und getestet, Das ist von Labor für Intelligente Sensor-Aktuator-Systeme (ISAS) selbst aufgebaut und es Besteht viele verschiedene Versionen des Crawlers. Diese Roboter werden für mehrere Zwecke verwendet. Darüber hinaus ist das Chassis von diesen Roboter werden komplett in 3D gedruckt und können daher sehr schnell gebaut werden.

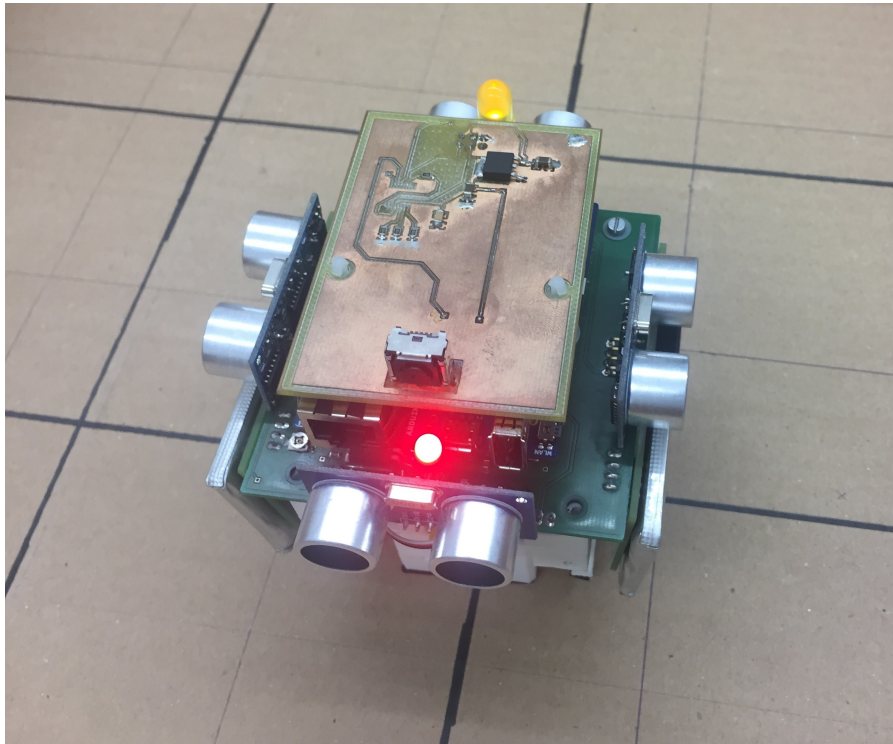


Abbildung 4

5.1 Aufbau

Der Crawler, der in die Projekt benutzt wird, ist die neueste Version. Die Hauptkomponente der Version in dieser Arbeit ist ein Arduino Yún. Die Arduino Yún Funktionen zwei verschiedene CPUs.

Der Arduino Yun verfügt über ein WLAN-Modul, mit dem ein WLAN generiert oder in ein WLAN eingefügt wird vorhanden kann gewählt werden. Darüber hinaus verfügt es über vier verschiedene Beine, wobei jedes einzeln von vier gesteuert werden kann Servomotor. Dies ermöglicht dem Roboter, sich gerade, seitlich und omnidirektional zu bewegen.

Der Crawler enthält vier Ultraschallsensoren und eine IMU. Es sind insgesamt vier Ultraschallsensoren installiert, die jeweils aus einem bestehen Empfänger und ein Sender existieren. Die Sender und Empfänger werden über einen Multiplexer (MUX) gesteuert. Die Sensoren sind in 90-Graden zueinander angeordnet, wobei sich nicht überschneidendes Erkunden der Bewegungsachsen ist möglich. Ferner ist eine IMU installiert, die spezifischere Bewegungsdaten, z.B. die Beschleunigung. Dies wird derzeit nicht verwendet.

LEDs Zusätzlich befinden sich oben zwei LEDs (gelb und rot), mit denen der Roboter kann einfacher, zum Beispiel von einer Deckenkamera, verfolgt werden.

Die Roboter bestehen aus vier Hauptelementen, drei Beinen und einem obersten Segment mit einer Länge von 17 cm und einer Breite von 5 cm das bildet den Körper. Jedes Bein hat zwei unabhängige lineare Freiheitsgrade (DOF), die zu insgesamt sechs. Jeder lineare DOF hat eine Reichweite von 2 cm. Der erste DOF jedes Bein erlaubt es, einzeln und das zweite angehoben zu werden. man lässt das vordere und hintere Bein seitlich und das mittlere Bein, um sich vorwärts

und rückwärts zu bewegen. Die Höhe des Roboters liegt zwischen 5 und 7 cm, abhängig von der Haltung der Beine. Aufgrund des symmetrischen Designs der mechanische Struktur des Roboters, gibt es keine dedizierte Front oder zurück, wenn also der Roboter die Richtung ändert, wird der Schwanz der Kopf und umgekehrt. Das Design der Roboter ermöglicht eine Vielzahl unterschiedlicher Grundfunktionen Bewegungsmuster, die alle aufgrund der Unabhängigkeit aller sechs Gelenke wie im beigefügten Video gezeigt. Ein wichtiges Bewegungsmuster für die Vorwärtsbewegung ist als Folgendes dargestellt. Hier wird zuerst der Mittelfuß angehoben, bewegt vorwärts und dann wieder gesenkt. Dann ist die vordere und hintere Füße werden gleichzeitig angehoben und die Basis wird relativ zum mittleren Fuß vorwärts bewegt wo der Roboter steht. Endlich die Vorder- und Rückseite die Füße werden wieder abgesenkt. Der Roboter befindet sich in seiner ursprünglichen Konfiguration und der nächste Schritt kann direkt folgen.

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \Phi_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \Phi_k \end{bmatrix} + \begin{bmatrix} \cos(\Phi_k) & -\sin(\Phi_k) & 0 \\ \sin(\Phi_k) & \cos(\phi_k) & 0 \\ 0 & 0 & \Phi_k \end{bmatrix} \begin{bmatrix} \hat{s}_k^F \\ \hat{s}_k^S \\ \hat{\omega}_k \end{bmatrix} + \begin{bmatrix} \omega_k^x \\ \omega_k^y \\ \omega_k^\Phi \end{bmatrix}$$

6 Simulation

6.1 Systemmodell

Vorwissen über das System findet im Systemmodell Anwendung. Mit der Überföhrungsfunktion $a : \mathbb{R}^2 \wedge S$ um den Folgezustand \underline{x}_{t+1} aus dem aktuellem Zustand abzuschätzen.

$$\underline{x}_{t+1} = a(\underline{x}_t)$$

Zusätzlich wurde ein normalverteiltes additives Rauschen $w \sim N(0, \sigma)$ angenommen.

$$\underline{x}_{t+1} = a(\underline{x}_t) + w$$

Die Folge von Bewegungsbefehlen an den Crawler ist bekannt. Wir können eine Fallunterscheidung für alle drei Bewegungsrichtungen durchführen.

$$a(\underline{x}_t, d) = a_{fwd}, f_{rd} == fwd \quad (7)$$

Die Schrittlängen und Rotationen, die bei der Ausführung einer Bewegung wurden in einem Vorexperiment gemessen. Dazu wurden mehrfach die Verschiebung und Rotation nach einer Bewegung aufgezeichnet und gemittelt. Die gesamte Systemmodell kann als Folgendes dargestellt werden.

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \Phi_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \Phi_k \end{bmatrix} + \begin{bmatrix} \cos(\Phi_k) & -\sin(\Phi_k) & 0 \\ \sin(\Phi_k) & \cos(\phi_k) & 0 \\ 0 & 0 & \Phi_k \end{bmatrix} \begin{bmatrix} \hat{s}_k^F \\ \hat{s}_k^S \\ \hat{\omega}_k \end{bmatrix} + \begin{bmatrix} \omega_k^x \\ \omega_k^y \\ \omega_k^\Phi \end{bmatrix}$$

Die Systemzustandmatrix beinhaltet 3 Variable $\underline{x}, \underline{y}$ und $\underline{\Phi}$.

6.2 Messmodell

Mit dem Messmodell wird beschrieben, welche Messergebnisse bei einem Systemzustand zu erwarten sind.

$$z_t = h(x_t)$$

Da die Ultraschallsensoren des Crawler nur grobe Abstandsmessungen ermöglichen, wurde auch hier ein normalverteiltes additives Rauschen $w \sim N(0, \sigma)$ angenommen:

$$\underline{z}_t = h(\underline{x}_t) + w$$

Die Messfunktion h kann für einen Zustand \underline{x}_t ausgewertet werden, indem ein Crawler der von x_t festgelegten Lage und Orientierung auf einer virtuellen Karte simuliert wird. Die Karte muss dazu bekannt sein.

6.3 Progressive Updates

Schrittweise Fusionierung der Messwerte in den geschätzten Zustand Vermeidet kleine Likelihoodwerte Eingabe nur Messwerte, keine Positionsschätzung durch Ultraschalldistanzen

$$\begin{aligned} f(\underline{x}|\underline{z}) &\propto f(\underline{z}|\underline{x}) \cdot f(\underline{x}) \\ &= (f(\underline{z}|\underline{x})^{\lambda_1} \cdot \dots \cdot f(\underline{z}|\underline{x})^{\lambda_s}) \cdot f(\underline{x}) \end{aligned}$$

7 Evaluation

7.1 Ergebnisse

8 Zusammenfassung und Ausblick

9 Das Filter XY

9.1 Grundlagen des Filters XY

Vektoren und Matrizen

$$\underline{x}, \mathbf{A}$$

Mengenzeichen

$$\mathbb{R}, \mathbb{N}$$

Zufallsvariablen, etc...

$$\mathbf{y}, \mathbf{z}, \text{Var}, \text{E}, \text{Cov}$$

Bitte nur Gleichungen nummerieren, auf die sich auch später bezogen wird

$$a = b + c \quad . \quad (8)$$

Laut (8) ist $a = b + c$.

Mehrzeiliger Formelsatz mit *align*

$$\begin{aligned} a &= b + c \quad , \\ a_{ij} &= b_{ij} + c_{ij} \quad . \end{aligned}$$

oder mit *multline*

$$\begin{aligned} a_{2343443} = \\ b + c + \frac{3464421}{32455767567567575677} \cdot (b_{ij} + c_{ij}) \cdot \int_{x=55}^{88} x^{67823+x} \frac{x}{32455767567567575677} dx \quad . \end{aligned}$$

So werden Bilder eingebunden (als pdf, jpg oder png)

Auf diese Abbildung wird dann mit Abb. 5 verwiesen.

Literatur

- [1] S. RUSSEL AND P. NORVIG *Artificial Intelligence - A Modern Approach*, Second Edition, Prentice Hall, 2003.



Abbildung 5: Hier kommen weitere Erklärungen zum Bild