UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA
OBERÖSTERREICH

Autonome Roboter Systeme

Übung 3 — Navigation, Lokalisierung und Kartierung

Dorit Borrmann

Winter 2023/2024

Ziele: In dieser Übung beschäftigen Sie sich mit Navigation, Lokalisierungs- und Kartierungsverfahren.

Voraussetzungen: Wahrscheinlichkeitsbasierte Robotik, Partikelfilter, ROS, Linux, und grundlegende C/C++/Python Kenntnisse.

Teil 1 – Navigation

Die Navigation eines mobilen Roboters orientiert sich üblicherweise am Ziel des Roboters. In dieser Aufgabe soll die Navigation eines mobilen Roboters ohne Ziel implementiert werden (vgl. Kapitel - Navigation, Folie 29-31).

Schreiben Sie einen ROS-Node, der die Navigation für den Turtlebot implementiert. Hierfür benötigen Sie einen `ros::Publisher`, der auf dem Topic `cmd_vel` eine Message vom Typ `geometry_msgs::Twist` publiziert. Um die Geschwindigkeit `msg.linear.x` und den Winkel `msg.angular.z` anhand der Laserscandaten zu bestimmen, implementieren Sie dazu eine Callback Funktion für das Topic `scan`.

Sei $\theta = 0$ die Vorwärtsrichtung des Roboters.

Zur Berechnung der Geschwindigkeit v gilt:

- Bestimmen Sie die minimale Distanz d in einem Bereich von $-30° < \theta < 30°$.
- Verwenden Sie die Parameter $d_{\min} = 0.2$, $d_{\max} = 1.0$ und $v_{\max} = 0.5$
- $v = 0.0$, wenn $d < d_{\min}$
- $v = v_{\max}$, wenn $d > d_{\max}$
- $v = v_{\max} \cdot d/d_{\max}$, sonst.

Zur Berechnung des Winkels α gilt:

- Verwenden Sie den Bereich von $-90° \leq \theta \leq 90°$.
- Für jede Lasermessung $l_i(\theta_i, d_i)$ innerhalb dieses Bereichs berechnen Sie ein Gewicht

$$w(l_i) = w_{\text{ori}}(l_i) \cdot w_{\text{dist}}(l_i)$$

- $w_{\text{dist}}(l_i)$ `= (1.0/(1.0 + exp(-(`d_i `- 1.50)/0.20)))` beschreibt das Gewicht basierend auf der Distanzmessung d_i. Setzen Sie $w_{\text{dist}}(l_i) = 1.0$ wenn d_i den Messbereich des Laserscanners überschreitet.

- $w_{\text{ori}}(l_i) = \cos(\varphi_i/1.2)$ bevorzugt die aktuelle Orientierung.
- Der Winkel berechnet sich durch

$$\alpha = \texttt{atan2}\left(\sum_i \sin\theta_i \cdot w(l_i), \sum_i \cos\theta_i \cdot w(l_i)\right)$$

- Setzen sie den berechneten Winkel α (in rad) als Winkelgeschwindigkeit `msg.angular.z`.

Testen Sie Ihre Implementierung systematisch. Passen Sie wenn nötig die Parameter an, um das Verhalten des Roboters zu verbessern.

Teil 2 – Kartierung In dieser Aufgabe verwenden Sie ein probabilistisches SLAM-Verfahren für Rasterkarten. `GMapping` (`http://wiki.ros.org/gmapping`) verwendet einen Partikelfilter um Rasterkarten aus Laserscandaten zu erzeugen. Verwenden Sie die Navigation aus Aufgabe 3.1 um mit dem Roboter durch das Labor zu fahren und Laserscans, Odometriedaten sowie die tf-Transformationen in einem `.bag`-file aufzunehmen.

Verwenden Sie den `GMapping` Node um eine 2D-Karte aus den aufgenommenen Daten zu generieren. Passen Sie die (Default-) Parameter an um gute Ergebnisse zu erzielen. Speichern Sie die Karte mit dem `map_saver` aus dem Package `map_server`. Verwenden Sie `rviz` um einen Screenshot der Karte mit der Trajektorie zu erstellen.

Im Package `turtlebot3_slam` befinden sich Launch-Files und Konfigurationsparameter für `GMapping`. In ROS werden Transformationen zwischen Koordinatensystemen durch tf-Transformationen repräsentiert (vgl. `http://wiki.ros.org/tf`). Stellen Sie sicher, dass eine Transformation zwischen `odom` und `base_scan` existiert, da `GMapping` diese benötigt um die Pose des Roboters in der Karte basierend auf Odometrie und Laserscandaten zu bestimmen.

Teil 3 – Monte Carlo Lokalisierung: In dieser Aufgabe werden Sie sich mit der Monte Carlo Lokalisierung mit adaptivem oder KLD-Sampling (wie von Dieter Fox beschrieben) in der in Aufgabe 3.2 erstellten Karte lokalisieren. AMCl (`http://wiki.ros.org/amcl`) verwendet Partikelfilter um die Pose des Roboters in einer bekannten Karte zu bestimmen. Im Package `turtlebot3_navigation` befindet sich ein Launch-File für AMCL. Weiterhin muss die Karte durch den `map_server` bereitgestellt werden. Welche Parameter müssen Sie anpassen um gute Ergebnisse zu erzielen? Dokumentieren Sie die Lokalisierung und die Entwicklung der Partikel mit mehreren Bildern oder einem Video. Schaffen Sie, die Parameter so zu setzen, dass das Kidnapped-Robot-Problem gelöst wird?

Abgabe: Reichen Sie den Code zu Teil 1, die Screenshots/Videos sowie die Erklärungen zu den Parametern bis angegebenen Abgabedatum um 23:59 Uhr auf `https://elearning.fh-ooe.at` ein. Verspätete Abgaben werden nicht akzeptiert.