

Abgabe zum Praktikum Mess- und Regelungstechnik

Revision 1

Simon Klüpfel, Lukas Zeller
Robotik und Telematik
Universität Würzburg
Am Hubland, D-97074 Würzburg

`simon.kluepfel@stud-mail.uni-wuerzburg.de`, `lukas.zeller@stud-mail.uni-wuerzburg.de`

17. November 2022

1 Einleitung

1020 PS auf 2162kg, und von 0 auf 100 km/h in 2,1 Sekunden[7]. Dazu noch ein Level an autonomen Fahren das uns wie von Zauberhand durch den Verkehr bringt und die kognitiven Fähigkeiten des Menschen teilweise bereits übersteigt. Die Rede ist vom Tesla Model S. Die enormen Fahrleistungen lassen sich durch modernsten Maschinenbau und Elektrotechnik sowie jahrelanger Entwicklung erreichen. Spannender ist nach Meinung der Autoren schon das (teil-)autonome Fahren und die Fortbewegung eines zwei Tonnen schweren Fahrzeugs in einer dynamischen Umgebung wie dem Strassenverkehr, ohne einen Unfall zu verursachen oder im Worst-Case Menschen zu verletzen oder gar zu töten. Dass dies nicht passiert spricht einmal für das Know-How der Tesla-Ingenieure als auch der Zuverlässigkeit der verwendeten Technik. In dieser Arbeit wird ein kleiner Einblick in die Technologie der Lokalisierung gegeben und ein Vergleich angestellt zwischen zwei Methoden, der AMCL- und Odometrie-Lokalisierung.

2 Zur Lokalisierung

Aus [4], Folie 5-2, lassen sich drei Grundvoraussetzungen sowie das Ziel für unsere Roboterlokalisierung definieren: Er bewegt sich in der Umgebung mittels bekannter Steuerbefehle, nimmt seine Umwelt mit Sensoren wahr und hat eine Karte seiner Umgebung zum Abgleich gespeichert. Das Ziel ist eine Positionsabschätzung zum Zeitpunkt t in X- und Y-Position sowie ein Heading welches die Richtung angibt. Als Referenz ist ein in der Umgebung verankertes Koordinatensystem gegeben. Weiter unterscheidet man zum Beginn der Lokalisierung zwischen lokaler und globaler Selbstlokalisierung. Im ersten Fall ist die Initialposition des Roboters grob bekannt, das Ziel ist es bei Bewegung des Roboters eine Neuberechnung der Position durchzuführen, dieses Verfahren nennt man *position tracking*. Bei der schwierigeren globalen Selbstlokalisierung kennen wir unsere Initialposition dagegen nicht, sie muss aus der Roboterbewegung und neugewonnenen Sensordaten errechnet werden. Hier kann das *kidnapped robot problem* eine Rolle spielen, bei der der Roboter in Zuge einer nicht abgeschlossenen Lokalisierung umgesetzt wird. Bei der Sensorik die zur Lageabschätzung verwendet wird unterscheidet man zwischen *propriozeptiven*, d.h. internen, und *exterozeptiven*, externen, Sensoren. Ersteres misst die lokal am Roboter anfallenden *onboard* [1] Eigenschaften wie die Drehzahl der Radmotoren oder die Temperatur, die exterozeptiven Sensoren liefern Messungen aus der

Umgebung des Roboters, als Beispiel nennt [1] Distanzmessungen wie Ultraschall, Laser, oder eine Kameraaufnahme der Umgebung.

3 Kurze Einführung in den Volksbot und ROS

In dieser Arbeit wurde das Robot Operating System, kurz *ROS*, auf einem Roboter, dem *Volksbot* verwendet. Der verwendete Roboter ist der *RT3-2* mit zwei bzw. einem (*RT-3*) passiven Rad und jeweils zwei angetriebenen Räder die vorne rechts und links angebracht sind. Entwickelt wurden diese vom *Fraunhofer Institut IAIS* aus Sankt Augustin. Die technischen Daten sind wie folgt:

Abmessungen	580x520x315mm (L x B x H)
Gewicht	17kg
Raddurchmesser	260x85mm (aktive Räder) 200mm (passive Räder)
Maximale Geschwindigkeit	2,2 $\frac{m}{s}$
Maximale Zuladung	25kg

Daten aus [3]

Auf dem Roboter ist ein Laserscanner sowie Hardware zur Verbindung mit dem als ROS-Plattform verwendeten Laptop installiert. Neben der Steuerung beziehungsweise Regelung ueber *ROS* lässt sich der Roboter über einen Joystick, ähnlich eines Gamecontrollers, manuell steuern. Das *Robot Operating System* entstand aus dem Bedarf fuer ein Open-Source-Roboter-Framework in der Robotercommunity. Es entstand Mitte der 2000er-Jahre aus Projekten an der *Stanford University* wie dem *Stanford AI Robot - STAIR*. 2007 konnte mithilfe von *Willow Garage Inc*, einer amerikanischen Robotikfirma, und derer Ressourcen die ROS-Konzepte in erste Implementationen umgesetzt werden. Heute hat *ROS* zehntausende Nutzer auf der ganzen Welt, die Anwendungen reichen von Amateurrobotern bis zu grossen automatisierten Industrieanlagen. *ROS* an sich besteht dabei aus vielen kleinen Programmen die ueber Nachrichten, die *Messages*, miteinander kommunizieren. Die Nachrichten werden dabei untereinander direkt verschickt ohne einen zentralen Bus oder aehnliches. Insgesamt folgt *ROS* damit einer Graphenarchitektur mit Knoten (englisch *nodes*) und Kanten (*edges*), sie entsprechen den Nachrichten. [6] Es werden sowohl vorgegebene ROS-Nodes, die vom Institut für Robotik und Telematik zur Verfügung gestellt wurden, als auch angepasste Nodes die die Autoren selbst erstellt beziehungsweise geändert haben, verwendet.

4 Odometrie, GMapping und AMCL im Überblick

Die Odometrie

Die Odometrie errechnet die Bewegung des Roboters ueber die Radgeschwindigkeit und -bewegung des *Volksbot*. Im Zentrum dieser Rechnung stehen die mathematischen Zusammenhaenge im Kreisbogen. Aus [2] wurde dazu folgende Grafik entnommen:

[5] stellt die Grundzuege dieses Verfahrens vor. Hierzu betrachte man zuerst folgende Abbildung und die geometrischen Zusammenhaenge:

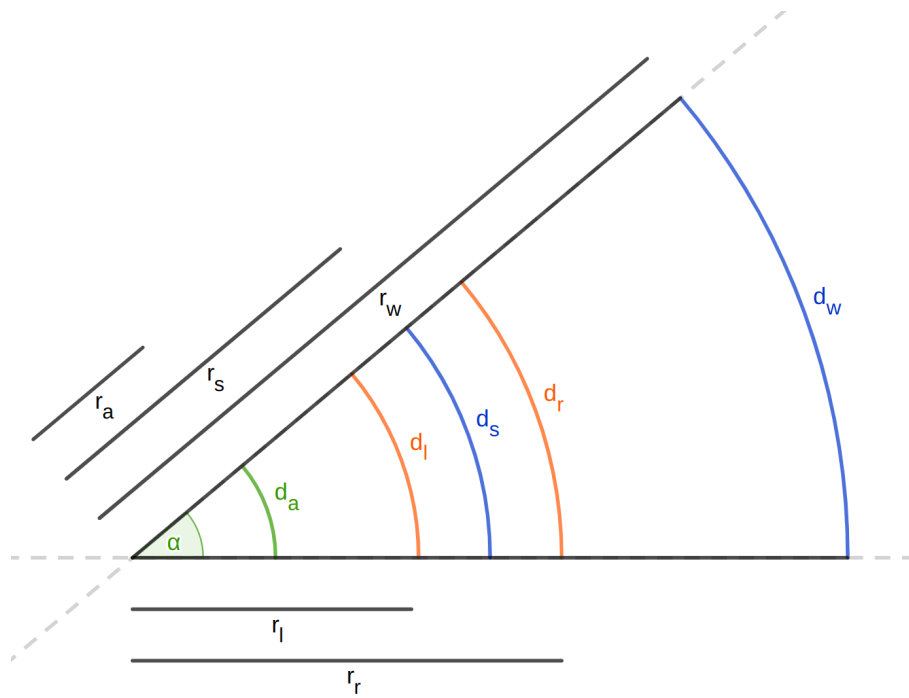


Abbildung 1: Ein Kreissegment mit verschiedenen eingezeichneten Bögen bei unterschiedlichen Radien. Die Radienlängen sind an der oberen Kreiskante aufgetragen. Der Winkel α des Kreissegments ist an der Spitze eingezeichnet

5 Die Funktionsweise der Pfadverfolgung und ihre Güte

6 Experimenteller Vergleich der Qualität der Pfadverfolgung bei Verwendung von AMCL- bzw. Odometrie-Lokalisierung

Nun soll betrachtet werden wie genau die Lokalisierungsmethoden einen vorgegebenen Pfad verfolgen. Desweiteren soll ein Vergleich zwischen beiden Methoden angestellt werden. Demnach ist die Lokalisierungsmethode besser, welche in der Verwendung eine geringere Abweichung vom vorgegebenen Pfad aufweist. Wir gaben dem Controller hier einen Pfad vor, welcher mit Hilfe der 3D-Design-Applikation *Blender* so erstellt wurde, dass er sowohl gerade Linien als auch weiche und harte Kurven enthält.

Start und Ende sind hier der Cursor unten im Bild, Anfangsrichtung ist gerade nach oben. Die verwendete Pfaddatei ist in `catkin_ws` als `Pfad.dat` zu finden. Dieser Pfad wurde auf dem Boden des Testgeländes mit Wollfäden markiert. Der Controller hat die Koordinaten des erstellten Pfades als relativ interpretiert. Dadurch konnten wir den Roboter am Startpunkt des Pfades platzieren, und in die Geradeaus-Richtung des Pfades orientieren. Dies wurde mithilfe durch ein Maßband, das durch die gerade Mittellinie des Pfades bis zum Roboter führt, und diesen dann symmetrisch teilt, möglichst exakt getan. Dazu wurde im Ursprung des Roboters in der Mitte der Vorderradachse ein Schraubenzieher befestigt, der gerade nach unten zeigt um die Position auf dem Boden leichter einsehen zu können. Es ergibt sich dennoch ein systematischer Fehler, da die Startausrichtung und -Platzierung des Roboters nicht perfekt mit der des Pfades am Boden übereinstimmt. Der geschätzte Maximalfehler für die Position des Roboters beträgt 2cm, der der Ausrichtung 2°.

Dadurch ergibt sich im maximalen Abstand vom Startpunkt ein maximaler Fehler von:

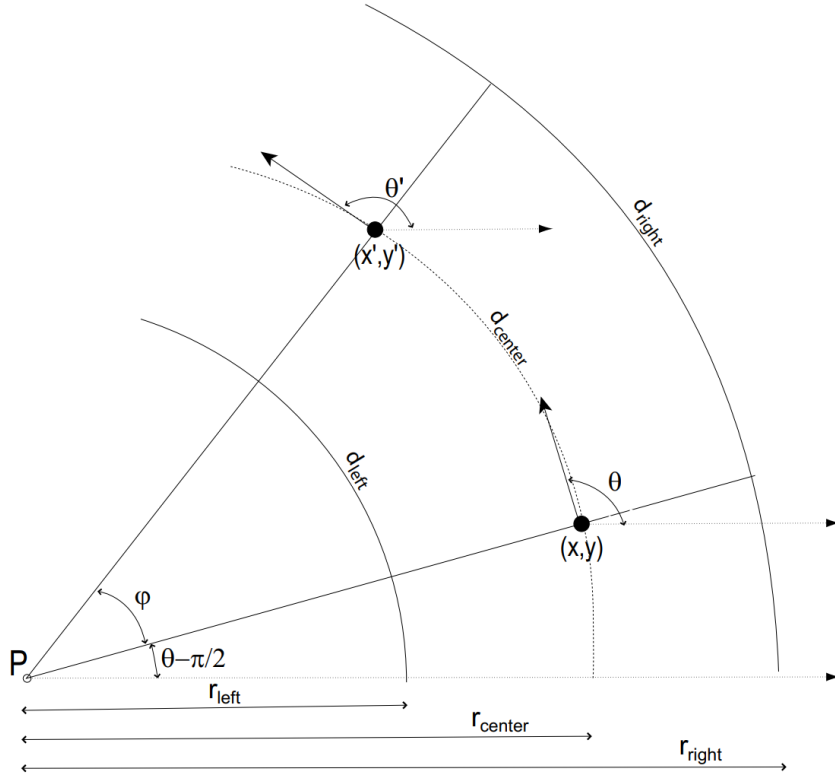


Abbildung 2: Die Geometrie der Odometrie, aus [5]. OLSON schreibt dazu: *Detailed odometry geometry. We compute the center of the circle P by forcing d_{left} and d_{right} to be two arcs with the same inner angle ϕ . From this, we can compute the new robot position (x', y', θ')*

$$0,02m + \tan(2^\circ) * 3m = 0,12m$$

An allen anderen Orten ist der systematische Fehler durch den geringeren Abstand vom Startpunkt kleiner. Nachdem der Roboter platziert wurde, wurde ein Node gestartet das den benannten Pfad abfährt, und dabei alle 2 Sekunden anhält, sodass seine Position auf dem Boden markiert werden kann. Dies wird für beide Lokalisierungsmethoden durchgeführt. Danach wird der senkrechte Abstand jedes Punktes zum markierten Pfad gemessen und in einer Tabelle eingetragen. Die Messergebnisse sind in Anhang¹ einzusehen.

7 Beschreibung der Ergebnisse

Der auf dem Boden markierte Pfad und die markierten Punkte, an denen der Roboter anhielt. Diese wurden jeweils auf einem Bild in Blau bei der Pfadverfolgung unter Verwendung von AMCL, und in Rot bei der Verwendung von Odometrie, zusätzlich zur besseren Sicht nachbearbeitet, eingekreist und mit der jeweiligen Nummer des Anhaltens gekennzeichnet, startend bei 0 für die Initialposition. Man beobachtet das der Pfad anfangs, bis etwa Punkt 7, mit beiden Lokalisierungsmethoden etwa gleich gut verfolgt wird. Der Durchschnitt der Abweichungen vom Pfad von Haltepunkt 1 bis zu Haltepunkt 7 liegt bei Odometrie bei 8cm, bei AMCL 4,2cm. Die folgenden Punkte zeigen bei beiden Lokalisierungsmethoden eine deutlich größere Abweichung auf: Punkte 8-13 bei der Verwendung von Odometrie haben eine durch-

¹Siehe Tabelle 1

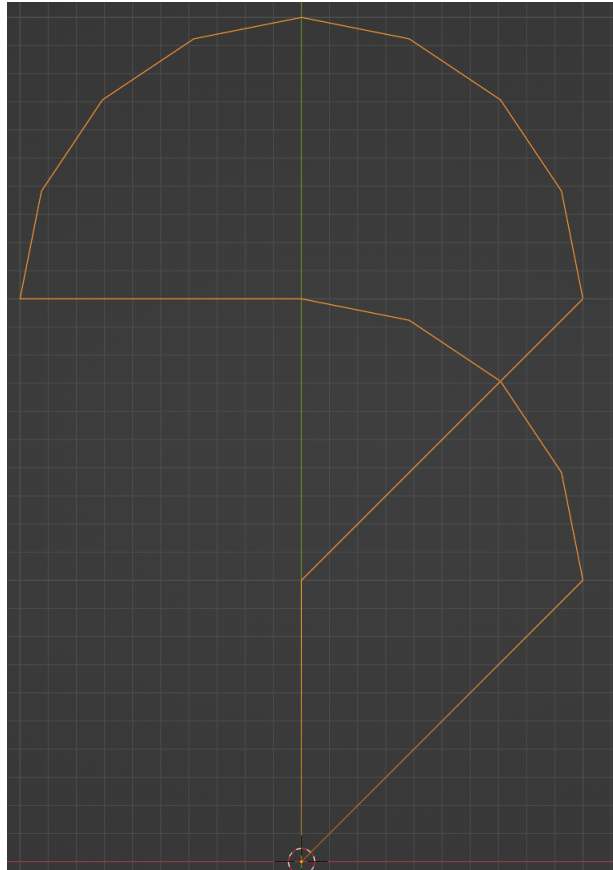


Abbildung 3: Der Pfad, wie er in Blender erstellt wurde. Man erkennt eine Linienstruktur bestehend aus gerade und gekurvten Teilstrecken, die einen geschlossenen Pfad bilden

schnittliche Abweichung von 51,2cm, bei der Verwendung von AMCL ist bei den Punkten 8-14 eine durchschnittliche Abweichung von 22,9cm aufgetreten. Hierbei ist zu bemerken dass durch das beiden Ansätze in der Realität stark verschiedene Pfade verfolgt wurden (siehe Abbildung) und es dazu kam dass AMCL einen längeren Weg zurückgelegt hat, und somit einen Haltepunkt mehr in der Messtabelle aufweist als Odometrie. Anhand der Markierungen auf dem Boden erkennt man auch dass der Odometrie-Ansatz sich nach Haltepunkt 7 durchgehend auf einer Seite des zu verfolgenden Pfades befindet, und immer weiter in diese Richtung von diesem abdriftet, während der AMCL Ansatz die Seite wechselt, in welche der Positionsfehler vorliegt.

8 Diskussion und Auswertung der Fahrtwege unter AMCL und Odometrie

Die durchschnittliche Abweichung aller Fehlerpunkte von AMCL liegt bei 12,6cm, bei der Odometrie sind es 25,9cm. Zu Beginn der Verfolgung des Pfades lagen beide Methoden etwa gleich nahe am Zielpfad, jedoch fing die Odometrie-Lokalisation ab der ca. Hälfte des Pfades an eine immer größer werdende Abweichung vom Zielpfad aufzubauen bis zu einem maximal gemessenem Fehler von 1,13m, während bei AMCL keine größere Pfadabweichung als 36cm gemessen wurde. Das bereits beschriebene Abdriften der Haltepunkte des Odometrie Ansatz lässt darauf schließen das die sich akkumulierenden Fehler in den Radumdrehungen, welche

das alleinige Lokalisierungsmerkmal der Odometrie ist, einen Drift in der Positionierung verursachen, was bewirkt das Odometrie für eine Lokalisierung auf Pfaden, die nicht kurz sind, keine der Realität akkuraten Ergebnisse liefern kann. Da AMCL nicht allein auf die Radumdrehungen angewiesen ist, kann es die Fehler dieser Informationsquelle ausgleichen und so auch auf längeren Pfaden bis zu einem gewissen Grad die Position genau bestimmen.

9 Zusammenfassung und Ausblick

Literatur

- [1] Dr. Mohamed Oubbati. Einführung in die Robotik - Vorlesung 3 Universität Ulm. https://www.uni-ulm.de/fileadmin/website_uni_ulm/iui.inst.130/Arbeitsgruppen/Robotics/Robotik/Vorlesung_03.pdf. abgerufen: 27.11.2022.
- [2] TU Dresden. ODOMETRIE. <https://robofab.inf.tu-dresden.de/spring/task/odometry/>. abgerufen: 26.11.2022.
- [3] Fraunhofer Institut IAIS. Volksbot. <https://www.volksbot.de/rt3-de.php>. abgerufen: 24.11.2022.
- [4] O. Bittel. Lokalisierung HTWG Konstanz. http://www-home.htwg-konstanz.de/~bittel/ain_rob/Vorlesung/05_Lokalisierung.pdf. abgerufen: 27.11.2022.
- [5] Edwin Olson. A primer on odometry and motor control. *Electronic Group Discuss*, 12, 2004.
- [6] Morgan Quigley, Brian Gerkey, and William D Smart. *Programming Robots with ROS: a practical introduction to the Robot Operating System*. Ö'Reilly Media, Inc.", 2015.
- [7] Tesla Deutschland. Model S Plaid. http://www.tesla.com/de_de/models. abgerufen: 26.11.2022.