Prototyphafte Entwicklung eines Postman Roboters

Florian Folger, Henry Tran

\* Fakultät für Geoinformation, Hochschule München, Karlstr. 6, 80333 München, Germany - [folger@hm.edu](mailto:folger@hm.edu) htran@hm.edu

**Modul *Projektstudium Navigation***

**KEY WORDS:** (MobileSim, Matlab, Sonar, Lokalistation, Mapping, OccupancyGrid, autonomes Fahren)

Zusammenfassung:

In dieser Arbeit präsentieren wir einen prototyphaften Postboten Roboter unter Verwendung von Matlab, MobileSim, Aria und dem Roboter P3-DX mit Sonarsensoren. Der Roboter soll sich autonom und kollisionsfrei durch den Raum bewegen und mit verschiedenen Funktionen ausgestattet werden, damit dieser eine Arbeitserleichterung für den Menschen darstellt. Der P3-DX soll selbstständig Briefkästen anfahren und ohne Kollision zum Stehen kommen. Im Anschluss fährt er zu seinem Ausgangspunkt autonom zurück. Während seiner Fahrt entdeckt der Roboter die Umgebung, mit seinen Sonarsensoren, und gibt diese als Karte wieder.

# Einleitung

Die Entwicklung von Postboten Robotern ist in letzter Zeit ein stark diskutiertest Thema. Viele große Firmen aus dem Bereich Logistik entwickeln derzeit derartige Geräte. In dieser Arbeit handelt es sich um eine prototyphafte Entwicklung eines Postboten Roboters. In der Realität werden diese Roboter ebenfalls mit Sonarsensoren ausgestattet, welche Hindernisse erkennen sollen. Die Beine dienen zur Orientierung, damit ist es dem Roboter möglich dem Menschen zu folgen, als eine Art Begleitroboter, geht der Mensch folgt ihm der Roboter. Unsere Arbeit soll zeigen, wie schnell und einfach eine prototyphafte Entwicklung eines Postman Roboters sein kann. Das folgende Kapitel soll erläutern, weshalb Simulationen eine entscheidende Rolle spielen, welche Motivation sich hinter unserem Ansatz verbirgt und wie unsere Implementierung einen Beitrag zur Forschung leistet.

## Simulation in der Robotik

Test und Änderungen können direkt am Roboter durchgeführt werden, dies ist umständlich, nimmt viel Zeit in Anspruch und der Roboter kann nicht verwendet werden. Die Erstellung von Simulationen vereinfachen den Entwicklern die Arbeit, spart Ihnen Zeit und Aufwand.

## Motivation

Die Forschung in der Robotik entwickelt sich rasant, weshalb es einer der spannendsten Forschungsgebiete in der heutigen Zeit darstellt. Der Postboten Roboter bietet einen einfachen Einstieg in die Thematik und liefert erste Einblicke in die Robotik. Für Studierende ist die Entwicklung eines prototyphaften Roboters von Vorteil, diese bekommen erste Eindrücke in ein größeres Projekt und welche Relevanz Simulationen haben.

## Related work

In diesem Abschnitt diskutieren wir verwandte Ansätze aus der Robotik und Navigation.

Ein ähnlicher Ansatz wurde von (Christaller, Thomas, 2013) vorgeschlagen. Dieser diskutiert über die Verwendung von Robotern, welche den Zweck dienen Menschen die Arbeit in verschiedenen Branchen zu erleichtern, diese aber nicht gänzlich zu ersetzten.

(Leonard, J. J., & Durrant-Whyte, H. F. (2012)) verwenden einen Roboter mit Sonarsensoren um sich in einem Gebäude zu lokalisieren. Für die Lokalisation im Gebäude verwenden Sie einen erweiterten Kalmanfilter, welcher die Polarkoordinaten von den Sonarsensoren als Referenzwerte verwendet.

## Beitrag der Arbeit

Durch die Entwicklung der verschiedenen Algorithmen welche unser Roboter ausführen kann, leisten wir einen Beitrag zur Forschung im Bereich der Robotik. Diese wurden in MATLAB implementiert

## Aufbau der Arbeit

Im nachfolgenden Abschnitt werden die Grundlagen beschrieben, auf welchen dieses Thema aufbaut. Abschnitt drei befasst sich mit der Entwicklung und Implementierung des Postman Roboters. Das vierte Kapitel zeigt unsere Ergebnisse und im fünften Abschnitt werden wir unseren Ansatz diskutieren und einen Ausblick geben, wie dieses Projekt weiterbearbeitet werden könnte.

# Methodische Grundlagen

Dieser Abschnitt befasst sich mit den grundlegenden Begriffen welche notwendig für dieses Projekt sind.

## Grundlegende Begriffe

*Koppelnavigation:* Beschreibt eine laufende, näherungsweise Ortsbestimmung eines bewegten Objektes aus Kurs und Geschwindigkeit.

*Odometriedaten:* Ist eine Methode zur Schätzung der Position und Orientierung eines mobilen Systems anhand der Daten seines Vortriebsystems. (Joachim Hertzberg, Kai Lingemann, Andreas Nüchter: Mobile Roboter 2012)

*SLAM:* Das Erstellen einer Karte einer unbekannten Umgebung unter gleichzeitiger Lokalisierung des Roboters werden unter dem Begriff SLAM zusammengefasst. (Michael Fürst 2016)

*Occupancy Grid:* bezieht sich auf eine Familie von Computeralgorithmen, in der die Erzeugung von Karten durch die Sensormessdaten des mobilen Roboters entsteht. Dabei sollten Position und Lage des Roboters bekannt sein.

## Mathematische Grundlagen

*Polarpunktberechnung:* Von einem bekannten Standpunkt wird eine Polaraufnahme durchgeführt, dazu benötigt man den Richtungswinkel vom Standpunkt zum Anschlusspunkt bzw. dieser muss zuerst berechnet werden. Dabei ist zu beachten, dass der Richtungswinkel einer Strecke der Winkel gegen Gitternord im Anfangspunkt der Strecke ist.(graphik). Dieser wird mit „t“ bezeichnet. Zur klaren Unterscheidung und Erkennung eines Richtungswinkels werden diese mit Indizes ausgestattet. Als nächstes braucht man die Strecke oder auch Entfernung genannt, vom Standpunkt zum Anschlusspunkt. Ist diese nicht gegeben, so muss man diese vorher mit der Formel ausrechnen (Graphik).

*Koppelnavigation:* Der Roboter nimmt per Sonarsensor Distanzen zur Wand auf, falls die Wand in der Reichweite des Sensors liegt. Die gemessenen Distanzen werden in Wandkoordinaten umgerechnet. Die relative Roboterposition , Sensorpositionen gemessene Distanzen zur Wand , Ausrichtungswinkel und der Sensorwinkel sind bekannt. Damit lassen sich die Wandkoordinaten wie folgt berechnen:

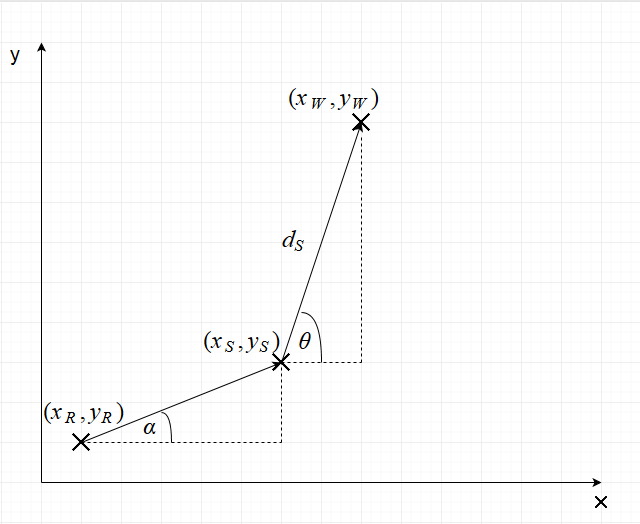


Abbildung 1 – Koppelnavigation

Formel 1

## Technische Grundlagen

***COM:*** Um die Kommunikation zwischen Computer und Roboter herzustellen, wird die COM-Schnittstelle verwendet. COM (Communication Port) ist eine serielle Schnittstelle, die häufig als „USB-Anschluss“ vertreten wird. Durch Abfrage im Gerätemanager und nachdem man den Roboter mit dem Computer verbunden hat, lässt sich der genaue COM-Port bestimmen.  
***Sonarsensor:*** Der Roboter ist mit Sonarsensoren (8) ausgestattet und verwendet dadurch das sogenannte Sonar-Range-System. Sonarsensoren nehmen die Umgebung durch Schallimpulse auf und dadurch lassen sich Objekte und Hindernisse im Raum detektieren. Jeder dieser Sensoren hat eine unterschiedliche Orientierung um mehr Objekte abzutasten.

# Prototyphafte Entwicklung eines Postman Roboters

In diesem Kapitel wird die Implementierung des Postboten Roboters beschrieben, und unsere Vorgehensweise. Das Ziel ist es den Roboter manuell anzusteuern, Briefkästen ab einer bestimmten Distanz autonom anzusteuern und selbstständig zu seinem Startpunkt zurückzukehren.

## Übersicht

*z.B. ein Überblick über Ihren Ablauf als state machine*

## MobileSim

Damit ein guter Start in dieses Projekt gelingt, wird zuerst der Roboter durch eine Simulation ersetzt. „MobileSim“ erlaubt es den Nutzer durch Eingabe von Befehlen, in der Eingabeaufforderung, verschiedene Objekte zu erzeugen, beispielsweise einen Roboter oder Briefkasten. Damit die Objekte in einer Karte dargestellt werden können, wurden uns Daten eines Flur Planes der Hochschule München bereitgestellt. In der „MobileSim“-Dokumentation befinden sich alle weiteren Commands. Für die Simulation wird ein weiteres Tool benötigt, damit kann der Roboter über MATLAB angesteuert werden. Wir verwenden „ARIA“, in dessen „ReadMe“-File befinden sich die Anweisungen, mit denen der Roboter bewegt und gedreht werden kann. Es ist ratsam „MobileSim“ durch die Eingabeaufforderung zu starten, da MATLAB abstürzen wird solange die Verbindung zum Roboter nicht unterbrochen ist.

## Robotersteuerung

Die Robotersteuerung wird hier in zwei Vorgänge eingeteilt, der erste Vorgang ist das manuelle Steuerung durch einen Nutzer, der den Roboter aktiv navigiert. Dies geschieht durch Eingabe in die Kommandozeile, die nach jeder abgeschlossenen Interaktion zur Verfügung steht. Um den Roboter vorwärts zu bewegen wird „W“, rückwärts „S“, „A“ und „D“ für eine jeweilige Drehung nach links bzw. nach rechts um jeweils 30°. Der zweite Vorgang ist die autonome Steuerung, der durch den Nutzer einmal ausgelöst werden muss. Der Roboter bietet unterschiedliche Befehle zur Fortbewegung, die man in der READ.ME in ARIA findet. Von der Befehlsliste wird „arrobot\_setvel(x)“ verwendet um den Roboter um ‚x‘-Einheiten (in diesem Falle Einheit in mm) vorwärts zu bewegen. Um die Orientierung bzw. „Blickrichtung“ des Roboters zu beeinflussen, lässt sich mit dem Befehl „arrobot\_setrotvel(x)“ (x Einheit in °) sowohl nach rechts (negativ x) als auch nach links (x positiv) drehen. Jeder Bewegungsvorgang, dazu gehören auch Drehbewegungen, benötigt eine gewisse Zeit durchgeführt zu werden, weshalb nach jedem Vorgang eine Arbeitszeit zugeordnet wird, diese vollständig durchzuführen, bevor man mit dem nächsten Befehl beginnt (sogenannte Pause). Daraufhin folgt ein relevanter Befehl „arrobot\_stop“ um den Roboter anzuhalten. Die autonome Steuerung wird mit Hilfe von implementierten Methoden („homing()“, „automove()“) realisiert. Diese benötigen Koordinaten von dem Ziel oder dem Objekt, worauf der autonome Bewegungsvorgang sich bezieht. Die Objekte oder auch Ziele werden mit einem Schwellenwert belegt, sodass der Roboter ab diesem gegebenen Wert autonom darauf zusteuert. Bei der autonomen Bewegung wird die Distanz zum Zielobjekt mit einer Abtastrate schrittweise abgefragt, damit sich dieser ständig aktualisiert. Mit einem zweiten festgelegten Schwellenwert lässt sich der Roboter mit einer verringerten Bewegungsgeschwindigkeit auf das Objekt näherungsweise zusteuern.

## Map Building

Der Roboter in der Simulation besitzt 16 Sonarsensoren, diese werden verwendet, um den Grundriss der Karte darzustellen. Durch Bewegung des Roboters werden die einzelnen Wandpunkte abgetastet. Die Sensordaten liefern lediglich Distanzen, weshalb diese durch die Koppelnavigation, siehe Formel 1, und polarem Anhängen in Wandkoordinaten umgerechnet werden. Während der Bewegung des Roboters wird dauerhaft ein Plot erzeugt, welcher mit den neu berechneten Daten befüllt wird. Diese zweidimensionale erzeugte Karte lässt sich auch als Binäres Occupancy Grid darstellen. Die Umgebung wird aufgenommen und mit Werten 0, für freie Flächen und 1 für belegte Flächen, beispielsweise eine Wand, befüllt. Das Map Building nimmt am meisten Zeit in Anspruch, weil ständig die Sensordaten grafisch dargestellt werden. Sinnvoller wäre es die Karte einmal aufzunehmen, zu speichern und im neuen durchlauf zu laden.

## Pathfollowing

Durch das „Pathfollowing“ ist es möglich die abgefahrene Strecke bzw. Route des Roboters grafisch darzustellen. Die Position des Roboters wird mit einem gegebenen Interval abgefragt und in einem Textdokument abgespeichert (X- und Y-Koordinaten relativ zum Roboter). Während die Umgebung des Roboters durch Sensoren in einem GRID dargestellt wird, kann mit der Eingabe „p“ in der Kommandozeile die abgefahrene Strecke des Roboters zusätzlich in das GRID abgebildet werden**.**

## Package Drop

Der Postboten Roboter sollte als Basisziel die Möglichkeit besitzen sich autonom einem Briefkasten anzunähern, sobald er eine gewisse Distanz erreicht hat. Es wurden dafür drei Briefkästen auf der Karte mit festgelegten Koordinaten hinzugefügt. Leider ist es nicht möglich die Briefkästen durch die Eingabe eines Befehls in „MobileSim“ auf die Koordinaten zu legen, deswegen werden diese manuell verschoben. Das Ziel ist es, dass der Briefkasten dem Roboter mitteilt wie weit er noch von ihm entfernt ist. Es werden zu allen Boxen die Distanzen zum relativen Standpunkt des Roboters berechnet.

Formel 2

Wenn ein Schwellwert von 1,5 Metern erreicht wird, teilt die Box dem Roboter dies mit. Dieser orientiert sich in die Richtung des Briefkastens und bewegt sich autonom darauf zu. Damit der Roboter zum Stehen kommt, werden Fehler in auf die X-und Y-Koordinaten aufaddiert. Nach seiner Ankunft kann er wieder manuell gesteuert werden.

## Homeing

Die Methode „Homing“ hat die Funktion den Roboter zum Startpunkt zurückzusetzen. Hierbei wird der Startpunkt entweder festgelegt durch Koordinaten oder es wird der Punkt verwendet beim Start des Roboters. Durch die Verwendung von geeigneten Stützpunkten (in dem Fall 4) werden zu jedem dieser relative Distanzen zu der aktuellen Position berechnet und ausgewertet. Diese zugehörigen Distanzen zu den vier Stützpunkten werden unterschiedlich gewichtet, damit der Roboter zu dem nächstliegenden Punkt hinsteuert. Dabei wird der Richtungswinkel zu diesem Punkt ausgerechnet und beim Roboter justiert, sodass er gerade drauf zusteuert. Von dem erreichten Stützpunkt werden wieder die Distanzen zu dem Startpunkt und den Stützpunkten berechnet. Sobald der Roboter den Stützpunkt mit dem geringsten Abstand zum Startpunkt erreicht, steuert der Roboter autonom mit demselben Algorithmus (siehe 3.3) auf den Startpunkt zu und hält dort an.

Formel 3

## Collision Avoidance

Beim autonomen Fahren des Roboters kann es dazukommen, dass dieser an einer Wand oder anderem Hindernis zum Stehen kommt. Damit wir solche Kollisionen vermeiden werden sie frühzeitig erkannt und vermieden. Die Sensoren benötigen unterschiedliche Schwellwerte, ansonsten wird der Roboter durch enge Gänge und Türen nicht mehr durchfahren können. Es werden die gemessenen Distanzen der Sensordaten betrachtet und anhand dieser ein geeigneter Schwellwert festgelegt. Sollte ein Hindernis detektiert werden stoppt der Roboter abrupt und dreht sich Vorerst um 180°. Im Nachhinein bewegt er sich selbstständig um 0.15 Meter um erneut manuell gesteuert werden zu können. Für die zwei Vorderen und Hinteren Sensoren wurde eine Grenze von 0.3 Meter festgelegt, die seitlichen Sensoren reagieren ab 0.10 Meter.

# Ergebnisse

*Beschreiben Sie hier Ihre Ergebnisse unter Verwendung von Statistiken, Diagrammen und Grafiken. Die Ergebnisse werden in diesem Kapitel noch nicht diskutiert!*

# Diskussion und AUsblick

In dieser Arbeit haben wir unseren Ansatz für eine prototyphafte Entwicklung eines Postboten Roboters mit Hilfe von Sonarsensoren erläutert und werden in diesem Abschnitt die oben genannten Ergebnisse diskutieren. Unser System hat demonstriert, dass die Funktionen sowohl in der Simulation als auch am realen Roboter ausgeführt werden können.

Es ist möglich den Roboter via WASD schrittweise zu bewegen.

Die manuelle Steuerung könnte im nächsten Schritt durch die Beschleunigungs- und Neigungssensoren eines Smartphones übernommen werden. Damit wäre auch eine leichtere Handhabung für den Benutzer gegeben. Beim Verwenden des realen Roboters war zu beobachten, dass dieser einen Rechtsdrall aufweist. Damit kommt es zu Problemen beim autonomen fahren, falls die Collision Avoidance nicht aktiv ist. Ebenfalls ändert sich durch diesen Drall das Mapping bzw. die erstelle Karte bei jeder neuen Fahrt. Damit die Funktionen Homeing und Package Drop funktionieren ist es sinnvoll jedes Mal dieselbe Startkoordinate zu wählen und nach mehreren Fahrten die Stützkoordinaten zu prüfen. Um diesen Rechtdrall auszugleichen könnte einerseits das Closed-Loop Verfahren angewandt werden oder eine Art Kalibrierungsdatei, durch den errechneten Fehler, dem Roboter mitgegeben werden.

Die Briefkästen haben in dieser Arbeit feste Koordinaten bekommen und können nur angefahren werden solange der Roboter in der Nähe des Schwellwerts liegt, kein Hindernis im Weg steht oder der Briefkasten sich in keinem Raum befindet. Es könnte ein Bug-Algorithmus implementiert werden welcher dem Roboter gestattet sich an den Wänden entlang zu hangeln, Räume abzufahren und als besucht zu taggen, falls der Erkundete Raum den Briefkasten nicht beinhaltet. Damit könnten Briefkästen, welche sich ein einem Raum befinden, ebenfalls detektiert und autonom angefahren werden. Das Homeing funktioniert momentan nur durch implementieren von Stützpfeilern an denen sich der Roboter ausrichtet. Damit ist es ihm nicht möglich aus einem Raum selbstständig herauszufahren und seinen Startpunkt zu erreichen. Auch hier könnte der Bug-Algorithmus Verwendung finden oder der Postboten Roboter könnte die gespeicherte Trajektorie als Referenz verwenden und diese bis zu seinem Startpunkt zurückfahren, wie eine Art rote Linie auf dem Boden. Außerdem wäre es sinnvoll den Roboter zu einer Ladestation selbstständig fahren zu lassen, falls dieser einen geringen Akkustand erreicht hat.

# Acknowledgements (optional)

Wir danken dem Labor für Navigation für die Bereitstellung der Hardware und Software und Herrn Abmayr für die Unterstützung während des Projektes.

# References

<https://www.logistik-watchblog.de/neuheiten/1790-zustellroboter-die-deutsche-post-arbeitet-am-postbot-2-0.html>

<https://www.golem.de/news/postbot-die-post-testet-zustellungsroboter-1710-130446.html>

Christaller, T., Decker, M., Gilsbach, J. M., Hirzinger, G., Lauterbach, K., Schweighofer, E., ... & Sturma, D. (2013). *Robotik: Perspektiven für menschliches Handeln in der zukünftigen Gesellschaft* (Vol. 14). Springer-Verlag.

Leonard, J. J., & Durrant-Whyte, H. F. (2012). *Directed sonar sensing for mobile robot navigation* (Vol. 175). Springer Science & Business Media.

<http://www.vermessung-und-geoinformation.de/wp-content/uploads/Auflage_12_Leseprobe_Kap_14.pdf>

<https://kamaro-engineering.de/ekf-slam-erklaert-wie-sieht-ein-roboter-die-welt/>

Joachim Hertzberg, Kai Lingemann, Andreas Nüchter: Mobile Roboter 2012