

All-Terrain-Ressource-Analysis Rover

Entwicklung eines günstigen und Ressourcen effizienten Robotik Systems zur Kartographie und Effizienz Steigerung in der Forst- und Landwirtschaft

Ein Projekt von Samuel Gorelikov (Klasse 10 Gymnasium Riedberg, Betreuerin: Frau Kleemiß)

Kurzfassung

In dieser Forschungsarbeit untersuche ich die Funktionsfähigkeit kleiner, kostengünstiger Robotik-Systeme. Diese sollen z.B. Landwirten helfen den Wasserverbrauch und die CO2 Emissionen zu senken, indem sie Datensammeln, die helfen die ideale Menge an Wasser zu bestimmen. Hierzu habe ich zwei Roboter-Prototypen mit unterschiedlichem Antrieb, verschiedener Steuerung und seitlich angebrachten Ultraschallsensoren entwickelt. Beide Prototypen wurden in verschiedenen Testsituationen hinsichtlich ihres Verhaltens beim autonomen Fahren untersucht, anschließend eine Fehleranalyse durchgeführt und Ideen zur Verbesserung der Fehler diskutiert.

Darüber hinaus habe ich einen motorisierten Roboterarm gebaut, um künftig einen hoch funktionalen Roboter für wissenschaftliche und z.B. landwirtschaftliche Untersuchungen zu entwickeln (All-Terrain-Ressource-Analysis Rover).

Inhaltsverzeichnis

1. Einleitung	2
1.1. Ursprung der Idee	2
1.2. Wie lässt sich ein solcher Roboter realisieren?	2
2. Vorgehensweise, Materialien und Methode	3
2.1 Bau des ersten Roboters	3
2.1.1 Motorisierung.....	3
2.1.2 Bau der Achse.....	3
2.1.3 Bau der Räder.....	4
2.1.4 Stromversorgung.....	4
2.1.5 Fahrprinzip	4
2.1.6 Sensorik und autonomes Fahren	5
2.1.7 Problematik beim ersten Roboter	5
2.2. Entwicklung einer zweiten Roboterbasis.....	6
2.3 Entwicklung eines Roboterarms	9
3. Versuch zur Testung der Autonomieität des Roboters in engen Räumen	10
4. Versuch zur Testung der Autonomieität des Roboters auf unebenem Gelände im Außenbereich ..	11
5. Versuch zur Ermittlung des Fehlerbereichs der Ultraschallsensoren	11
6. Versuch zur Messung der Präzision des Roboters	12
7. Ergebnisdiskussion	13
Wie wird das Lidar integriert?.....	14
9. Zusammenfassung.....	15
10. Quellen und Literaturverzeichnis	15

1. Einleitung

1.1. Ursprung der Idee

Im Februar 2021 habe die Landung des Mars-Rovers „Perseverance“ live mitverfolgt und mich gefragt, wie die Zukunft der Forschung und Wirtschaft mit Hilfe von Robotern aussehen können. Ich sah erst einmal die Möglichkeit eines Wissenschafts-Robotik-Systems, doch mir wurde bewusst, dass solch ein System auch in der Landwirtschaft oder z.B. in der Forstwirtschaft anwendbar sei.

Zudem kam auch die Frage auf, wie viele Optionen für wissenschaftliche Untersuchungen man auf einem kleineren Roboter unterbringen könnte.

Bei meiner Recherche stieß ich auf den vierbeinigen Roboter „Spot“ des Unternehmens Boston Dynamics, dessen Preis bei rund 66000 Euro liegt, was mir sehr teuer erschien. Mir war klar, dass der hohe Preis unter anderem durch die Verwendung von Beinen zustande kommt. Daher stellte ich mir die Frage: „Lässt sich eine preisgünstige Roboterbasis für Forschung und Alltag entwickeln?“

Aus der Frage heraus setzte ich mir das Ziel, einen Prototyp für eine Roboterbasis zu entwickeln, welcher ausbaufähig für verschiedene Sensoren und andere Messinstrumente ist. Mir war jedoch auch die negative Wirkung der Produktion von elektrischen Komponenten bekannt, daher entschloss ich mich, die Elektronik so zu bauen, dass diese leicht austauschbar ist und nicht der ganze Roboter mit all seinen Bestandteilen auf dem Müll landet.

1.2. Wie lässt sich ein solcher Roboter realisieren?

Um die Basis eines Roboters für Landwirtschafts- und Forschungszwecke und zu entwickeln, waren für mich drei Aspekte zentral:

- die Eigenschaft ihn fernsteuern zu können,
- das autonome Fahren eines Roboters,
- eine Basis, die möglichst modular aufgebaut ist und die Unterbringung von Sensoren und Messinstrumenten erlaubt.

Für die Fernsteuerung eignen sich Radiowellen am besten, da sie zuverlässig sind und auf großen Abständen funktionieren.

Für das autonome Fahren gibt es zwei Möglichkeiten:

- ein einfaches Prinzip, bei welchem der Roboter nur Hindernissen ausweicht,
- oder ein intelligentes Prinzip, bei welchem der Roboter auch Besonderheiten der Umgebungen erkennen und sich an diese anpassen kann.

Beide Prinzipien lassen sich mittels kleiner Computer und Sensoren umsetzen.

Die Modularität beschreibt die Optionen, den Roboter auf den Zweck und die Umstände anzupassen. Dies lässt sich durch austauschbare Sensoren und passende Befestigungen erreichen.

Für die Bewegung des Roboters gibt es drei Möglichkeiten: Räder, Ketten oder Beine. Für einen kleinen und günstigen Roboter scheiden Beine aus, da diese für jedes Gelenk Motoren benötigen, was den Preis des Roboters stark erhöht, daher habe ich im Rahmen dieses Forschungsprojekts sowohl Ketten als auch Räder ausprobiert. Ein weiterer wichtiger Aspekt eines Roboters ist die Sichtbarkeit der Untersuchungsobjekte und der Umgebung, die erforscht werden sollen. Dies kann man durch einen beweglichen Roboterkopf oder Arm erreichen. Diese müssen daher eine Kamera und Sensoren beinhalten.

Es gibt derzeit Computer, die klein und günstig genug sind, um die Steuerung eines kleinen Roboters zu übernehmen, gleiches gilt für Motoren und Kameras. Daraus resultierend lässt sich ein kleiner Roboter auch mit einfach zugänglichen, günstigen Komponenten von einem Schüler bauen.

In dieser Forschungsarbeit habe ich daran gearbeitet, einen solchen kleinen Roboter zu konstruieren und mich mit den Anwendungszwecken dieser Technologie auseinandergesetzt.

2. Vorgehensweise, Materialien und Methode

2.1 Bau des ersten Roboters

2.1.1 Motorisierung

Bei dem Bau des ersten Roboters entschied ich mich dazu, vier Räder zu nutzen die alle motorisiert waren. Ich entschloss mich dazu Schrittmotoren zu benutzen, da diese sehr präzise zu steuern sind. Ich ordnete die Räder und Motoren in einem Rechteck an, mit dem Ziel, die Motoren in entgegengesetzte Richtungen fahren zu lassen, um den Roboter zu drehen. Als ich dann den so konstruierten Roboter testete, zeigte sich, dass die Motoren zu schwach waren, um den Roboter so zu drehen, wie ich es geplant hatte.

2.1.2 Bau der Achse

Mein Budget war aber schon erschöpft und ich konnte keine anderen Motoren kaufen. Daher suchte ich nach einer anderen Lösung und teilte ich das selbstgebaute Chassis in zwei Segmente und montierte in die Mitte zwischen den beiden Hälften ein Gelenk. Nun musste ich nur den Winkel der beiden Hälften zueinander verändern, wodurch sich der Roboter drehen konnte.

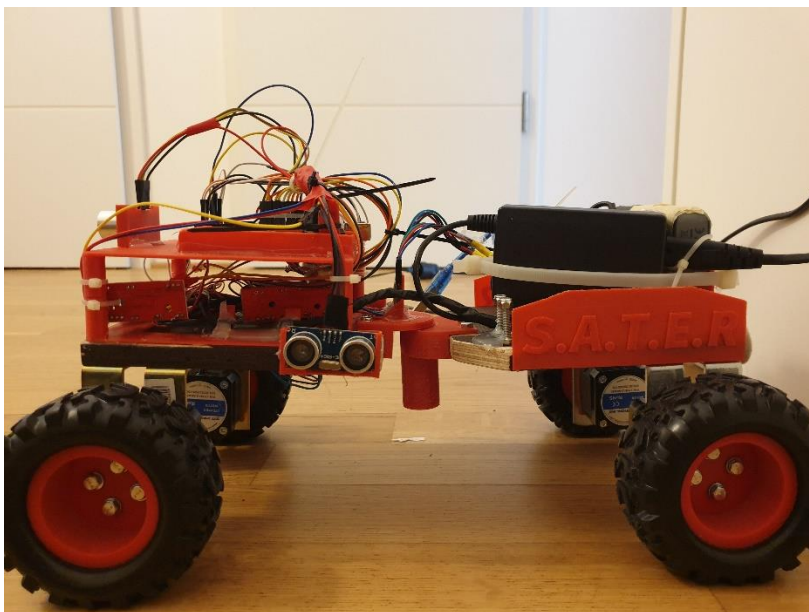


Abbildung 1: Roboter 1 - Seitenansicht

2.1.3 Bau der Räder

Ich wollte für den Roboter Räder haben, mit denen man auch unwegsames Terrain überwinden kann, dabei betrachtete ich ein alten Spielzeug-RC-Truck, dessen Felgen einen großen Durchmesser hatten und der mit Reifen ausgestattet war, die eine gute Struktur besaßen. Jedoch ließen sich diese Räder nicht auf den Schaft des Schrittmotors montieren. Ich entwarf passende Felgen, druckte sie mit einem 3-d-Drucker aus und befestigte die vier Reifen des Spielzeug-Trucks auf diesen Felgen.

2.1.4 Stromversorgung

Die Motoren des Roboters benötigten 12V. Um diese Spannung bereitzustellen, verwendete ich ein altes Ladegerät, wodurch der Roboter kabelgebunden an die Stromquelle war.

Der Arduino, den ich zur Steuerung nutzte, benötigte 5V. Diese Spannung stellte ich dem Roboter mit einer Powerbank zur Verfügung.

2.1.5 Fahrprinzip

Damit der Rover um Kurven fahren kann, muss er sich drehen, so dass die beiden Hälften ungefähr in einem 45 Grad Winkel zu einander stehen.

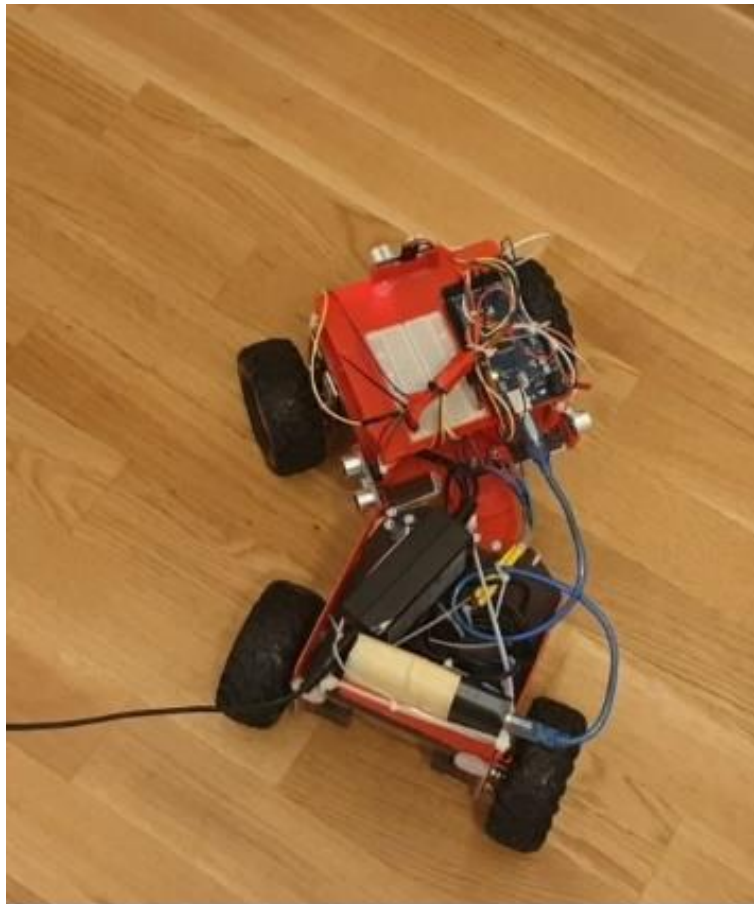
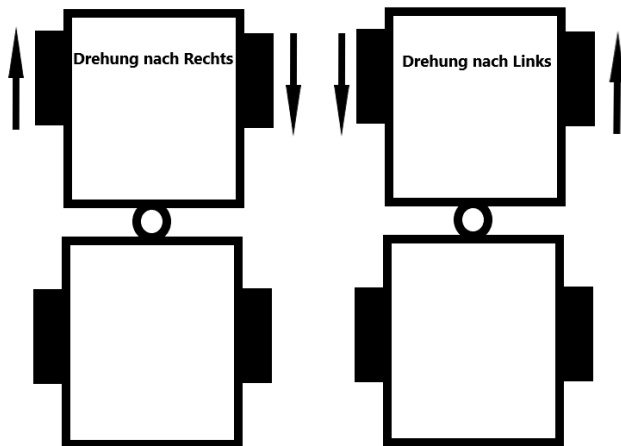
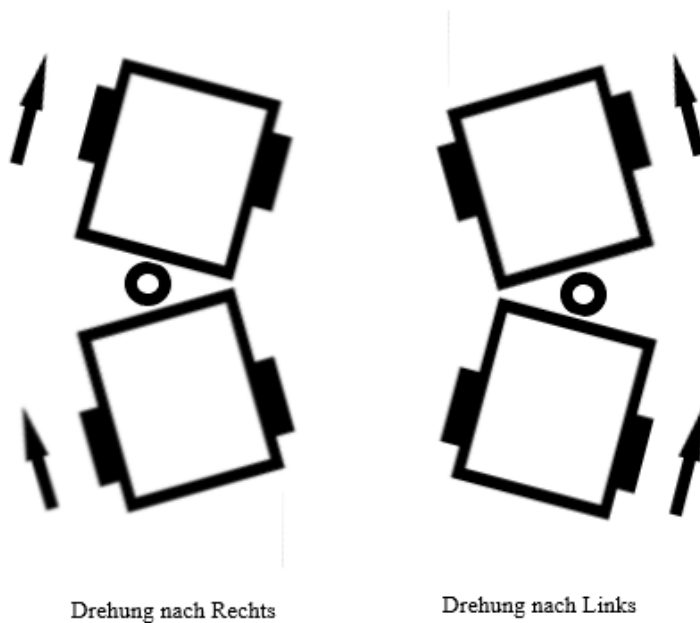


Abbildung 2: Roboter 1 – Sicht von oben

Eine geeignete Drehung erreicht der Roboter, indem er die Motoren vorne in entgegengesetzte Richtung drehen lässt:



Um danach die Richtung der Drehung beizubehalten, drehen sich danach die Motoren nur auf einer Seite, wodurch der Roboter mit konstantem Winkel dreht.



Damit er geradeaus fahren kann, aktiviert der Roboter zuerst nur die Vorderräder, wodurch er sich ausglättet. Dann werden die Hinterräder dazugeschaltet und er fährt im Allrad-Modus.

2.1.6 Sensorik und autonomes Fahren

Den Rover habe ich mit drei HC-SRO4 Ultraschallsensoren ausgestattet. Diese ermöglichen es dem Roboter, selbständig Objekten auszuweichen. Die seitlichen Sensoren sind in einem 90 Grad Winkel zu dem Frontsensor angeordnet, da der Roboter nach links oder rechts ausweichen können muss.

2.1.7 Problematik beim ersten Roboter

Beim ersten Roboter ergaben sich mehrere Probleme. Ein Problem war, dass der Roboter nur autonom fuhr und man selbst keinen Einfluss auf das Fahren hatte. Ebenfalls war der Roboter zu schwach und

hat Probleme mit der Bewältigung von unwegsamem Terrain, da die Motoren nicht leistungsfähig genug waren. Das dritte Problem des Roboters war, dass die Elektronik den kompletten Platz beanspruchte und dadurch kein Platz für die Montage weiterer Komponenten zur Funktionalisierung des Roboters vorhanden war. Zudem besitzt der Prototyp 1 keinen Akku, was den Roboter für einen Einsatz in weiterer Entfernung nutzlos macht.

Um diese Probleme zu beheben, entwickelte und baute ich eine zweite Roboterbasis.

2.2. Entwicklung einer zweiten Roboterbasis

Bei der Planung des Projekts suchte ich nach einer bereits fertigen Roboterbasis mit Ketten, da ich Ketten als durchaus gute Lösung für die zuvor aufgetretenen Probleme beim Prototyp 1 betrachtete. Ich entschied mich für ein fertiges Paket, da dies preisgünstiger als eine selbstgebaute Roboterbasis war. Ich stieß bei der Suche auf eine Roboterbasis, die zwei Motoren mit Getriebe, ein Chassis und die Ketten beinhaltet. Beim Aufbau der Basis fielen mir mehrere Problemstellen auf, die man anpassen musste. Zum einen besaß das Paket keine Steuerung. Diese musste ich daher selbst bauen. Zum anderen war der Kettenantrieb von schlechter Qualität, so dass ich die Ketten zusätzlich noch besser spannen musste. Außerdem war die Basis des Roboters zu klein, um einen Roboterarm, Kopf und Sensoren montieren zu können. Also entschloss ich mich dazu, selbst die Basis zu überarbeiten.

Folgende Punkte musste ich anpassen, um einen funktionalen Roboter zu bauen.

1. Wahl der Steuerkarte: TB6612FNG H-Bridge oder L298N

Für die Steuerung des Roboters musste ich mich zwischen zwei Steuerkarten entscheiden, der TB6612FNG H-Bridge oder der L298N. Hierbei waren für mich drei Aspekte zentral, die Größe der Steuerkarten, die Effizienz und die maximale Stromstärke der Steuerkarten. Da ich meinen Roboter möglichst viel Platz für Instrumente geben wollte, suchte ich nach einer möglichst kleinen Steuerkarte. Die TB6612FNG H-Bridge -Steuerkarte ist deutlich kleiner (ungefähr ein Viertel der Größe der L298N Steuerkarte) und in dieser Hinsicht besser geeignet. Die Effizienz der Steuerkarte war mir ebenfalls wichtig, da ich aus Platzgründen keine große Kühlung verwenden wollte. Zudem wollte ich die Akkuladung maximieren, wobei eine hohe Effizienz sehr praktisch und wichtig ist. Die TB6612FNG H-Bridge besitzt eine Effizienz von 91-95% im Vergleich zu der Effizienz der L298N Steuerkarte mit 40-70%, somit ist die TB6612FNG H-Bridge deutlich effizienter als die L298N und dadurch auch in dieser Hinsicht besser geeignet. Bei der maximalen Stromstärke kommt die TB6612FNG H-Bridge auf 3,2 Ampere im Vergleich zu 3 Ampere bei der L298N Steuerkarte.

Vergleich der Eigenschaften:

TB6612FNG H-Bridge	L298N Steuerkarte
Motor Spannung: 2,5-13,5 Volt DC	Motorspannung: 4,5-46 Volt DC
Maximale Stromstärke: 3,2 Ampere	Maximale Stromstärke: 3 Ampere
Kein Kühlblock	Kühlblock
Effizienz: 91-95%	Effizienz 40-70%
Spannungsabfall: 0,05-0,13 Volt DC	Spannungsabfall: 1,4 V DC
Standby Modus	Kein Standby Modus
Größe: 23mm*28mm*8,5mm	Größe: 43*43*26mm

2. Korrektur der Ketten

Als ich den Roboter fahren ließ, stellte ich fest, dass der Roboter ungleichmäßig fuhr. Das lag an den unterschiedlichen Spannungen der Ketten auf beiden Seiten. Um dieses Problem zu lösen, untersuchte ich die Ketten. Diese konnte man jedoch nicht kürzen, daher druckte ich mit einem 3-d-Drucker ein Rad mit einem größeren, geeigneten Durchmesser aus. Das neue Rad montierte ich dann an ein Kugellager und befestigte dieses an der Achse, wodurch die Ketten gespannt wurden und die freie Bewegung des Roboters wieder möglich wurde.



Abbildungen 3-5: -Kette vor der Anpassung/3d gedrucktes Rad und Kugellager/ Kette nach der Anpassung

3. Finden einer passenden Stromquelle

Für den Roboter benötigte ich eine leichte, kleine, leistungsstarke und bezahlbare Energiequelle. Ein Bohrmaschinen-Akku weist diese Eigenschaften auf. Dementsprechend entschloss ich mich dazu, einen Bosch-18V Bohrmaschinen-Akku zu verwenden. Jedoch musste noch ein Problem vor der Verwendung gelöst werden: Der Akku gibt 18V aus, der Arduino hingegen benötigt 9V und die Motoren 12V. Mit zwei parallel geschalteten Spannungswandlern löste ich dieses Problem und regelte die Spannung für den Arduino auf 9V und die Spannung für die Motoren auf 12V. Ein weiteres Problem war die Befestigung des Akkus an der Roboterbasis. Hierfür entwarf ich eine Halterung, die den Akku flexibel und leicht wechselbar machte und druckte sie mit einem 3-d-Ducker aus.

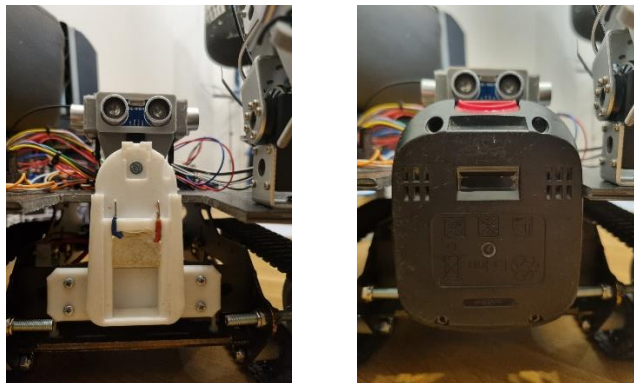
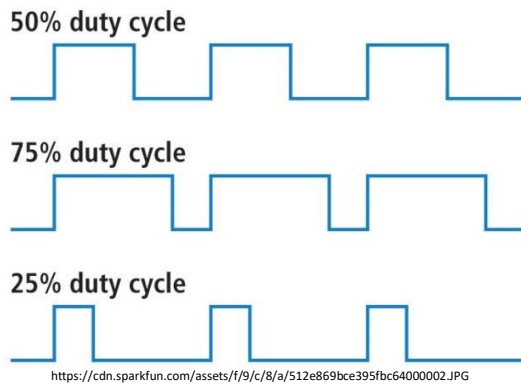


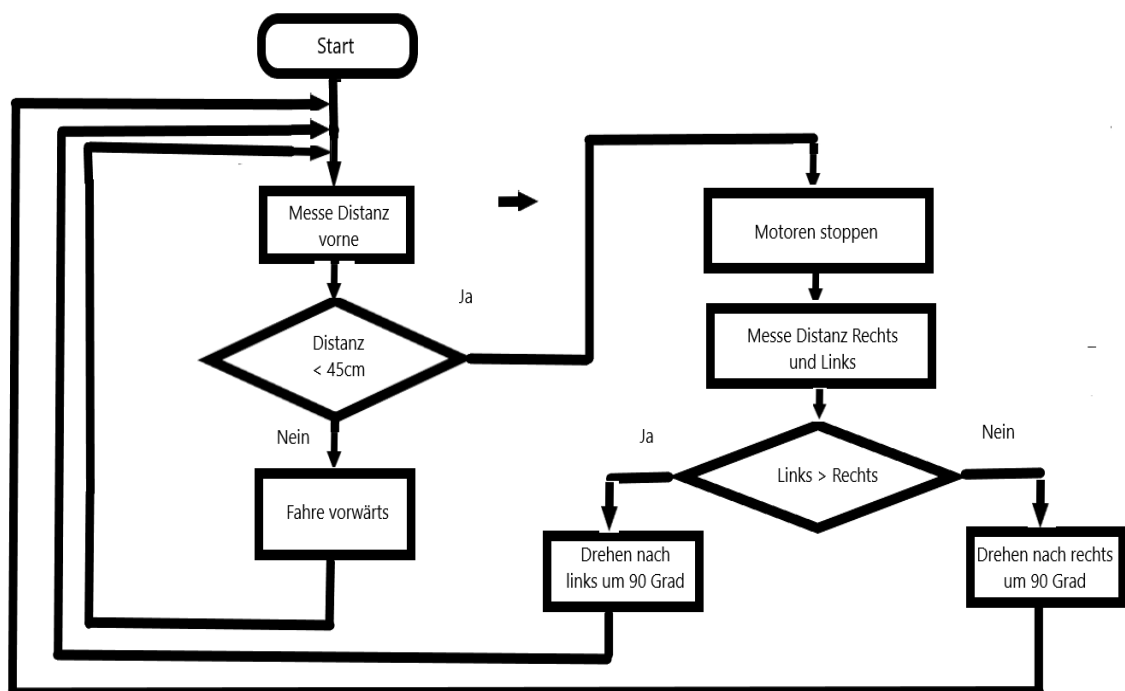
Abbildung 6: Akku-Halterung

4. Steuerung des Roboters

Um den Roboter manuell zu steuern, wollte ich eine 4-Kanal RC- Fernbedienung, die zur Steuerung von RC Flugzeugen verwendet wird, nutzen. Dabei stieß ich jedoch auf das Problem, dass ich durch die Verwendung von zwei Motoren zwei Steuerhebel zur Steuerung benötigen würde und jeweils manuell die Geschwindigkeiten beider Motoren hätte anpassen müssen (z.B., wenn der Roboter Fahrtrichtung hätte ändern sollen). Dies hielt ich für zu umständlich und ineffizient. Dementsprechend entschloss ich mich dazu, einen Arduino UNO zu benutzen, um die Kanäle der Steuerung digital zu kombinieren. Um dies möglich zu machen, verband ich den Receiver der Fernbedienung mit dem Arduino und las das Signal des Receivers aus. Es handelt sich dabei um ein PWM-Signal. Ein PWM-Signal (PWM: Pulseweitenmodulation) ist ein Signal mit konstanter Periodendauer, bei welchem jedoch sich die Pulsdauer verändert. Bei diesem Signal ist die Länge des Rechtecksignals wichtig. Dieses Signal lässt sich demodulieren und dann zur Steuerung anwenden. Ich las die PWM-Signale des rechten Steuerhebels (zwei Kanäle) aus, kombinierte diese im Arduino und sendete das Signal an die Steuerkarten der Motoren weiter.



Die Steuerung des Roboters über die RC-Fernbedienung funktionierte ohne Probleme und ich beschloss, dazu den Roboter intelligenter zu machen. Hierzu nahm ich drei Ultraschall-Sensoren und nutzte sie, um einen Algorithmus zu schreiben, der dem Roboter ermöglicht, automatisch auftauchenden Objekten auszuweichen. In diesem Code messe ich mit den Ultraschallsensoren sowohl die Distanz zwischen dem Roboter und dem nächsten Objekt als auch die Abstände nach rechts und links. Die Ultraschallsensoren messen die Distanz, indem sie eine Ultraschallwelle losschicken und diese wieder auffangen. Dabei wird die Zeit t_{Laufzeit} gemessen. Um daraus die Distanz zwischen Roboterbasis und Objekt d_{Objekt} zu berechnen, ließ ich den Arduino die Formel: ($d_{\text{Objekt}} = t_{\text{Laufzeit}} \cdot 0,034/2$) lösen. Diese Formel ergibt sich aus der Schallgeschwindigkeit, welche gleichförmig ist und in trockener Luft bei 20 Grad Celsius 343,2m/s entspricht. Möchte man daraus eine Strecke s errechnen, muss man die Formel umstellen: $s = v \cdot t$, eingesetzt ergibt sich $s = \frac{343,2m}{s} \cdot t_{\text{Laufzeit}}$. Der Faktor 0,034 ergibt sich, da man die Distanz d_{Objekt} in Zentimetern haben möchte. Dieses Ergebnis wird dann durch zwei geteilt, da die Strecke in der Zeit t_{Laufzeit} von den Schallwellen zweimal zurückgelegt wird, hin und zurück. Mit dem durch den Code und die Sensoren ermittelten Abstand kann der Algorithmus entscheiden, in welche Richtung der Roboter sich bewegen muss, damit er nicht auf Objekte stößt.



5. Roboterbasis

Die gekaufte Roboterbasis, die ich zunächst verwendete, hatte nur eine kleine Plattform. Diese bot nicht genug Platz für die Montage eines Roboterarms und mehrerer Sensoren. Um dieses Problem zu beheben, verwendete ich eine Platte aus gepressten Karton. Diese Platte war aber nicht stabil genug und begann durch die Vibration der Ketten zu oszillieren. Die Platte aus gepressten Karton war somit keine gute Lösung. Ich suchte nach anderen, geeigneteren Materialien und verglich dazu Holz, Aluminium und die Kunststoffe Plexiglas und Polyvinylchlorid (kurz „Vinyl“). Bei der Entscheidung, welche dieser Materialien ich nutzen könnte, waren für mich drei Aspekte relevant: Stabilität, Gewicht und der Aufwand der Verarbeitung.

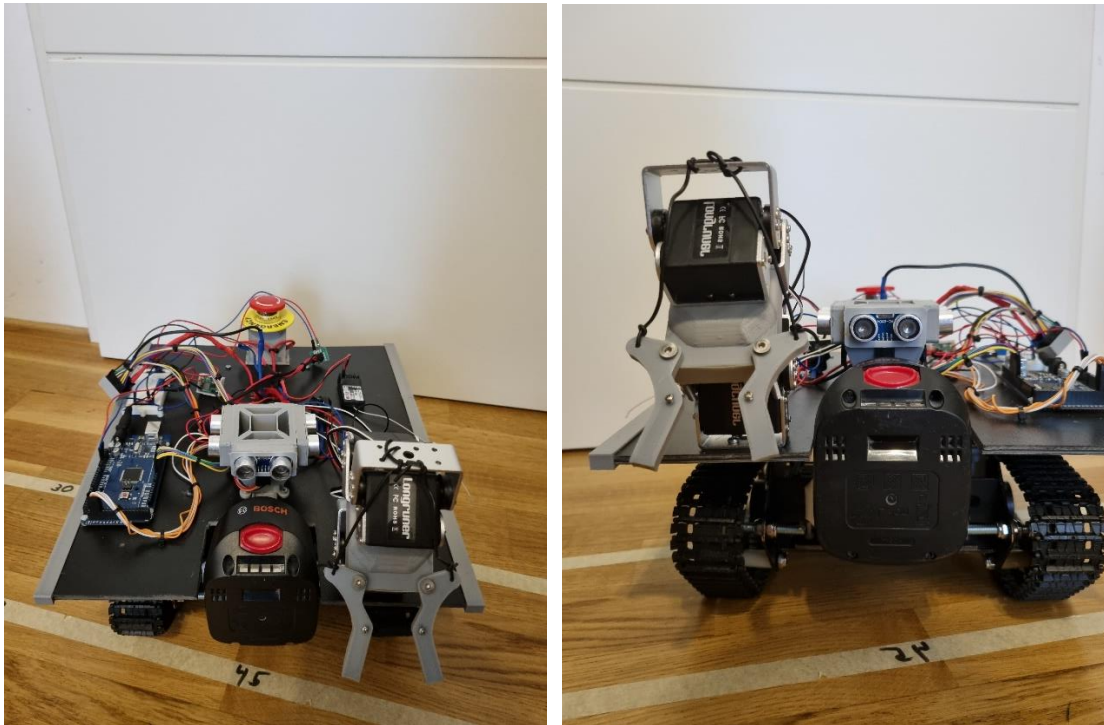


Abbildung 7: Roboterbasis

	Holz	Aluminium	Plexiglas	Vinyl
Verarbeitung	leicht	mittel	mittel	leicht
Stabilität	oszilliert stark	oszilliert stark	oszilliert stark	schwache Oszillation
Stärke	5mm	2mm	3mm	4mm
Dichte	870kg/m ³	2700kg/m ³	1180kg/m ³	1380kg/m ³

Die Vinylbodenplatte war von den vier Materialien am stabilsten und wackelte am wenigsten.

2.3 Entwicklung eines Roboterarms

Damit der Roboter seine Aufgaben (Objekte untersuchen oder verändern) erfolgreich erledigen kann, benötigt er einen Roboterarm, um sich passend zur Umgebung auszurichten. Dieser Aspekt blieb für mich jedoch zunächst zweitrangig, da der Fokus dieser Forschungsarbeit auf der Roboterbasis lag und der Arm nicht als Teil dieser Basis geplant war. Um aber eine bessere Vorstellung der Fähigkeiten des Roboters zu entwickeln und zeigen zu können, dass die Roboterbasis tatsächlich funktional ist, entwickelte ich aber ein Prototyp für einen Roboterarm mit motorgesteuerter Mechanik.

1. Bau des Arms

Beim Bau des Arms musste ich auf zweierlei achten. Zum einen sollte sich der Arm frei bewegen können, zum anderen musste ich aber auch darauf achten, dass der Arm bei seiner Bewegung nicht die Sensoren beeinträchtigt oder gar stört. Ich entschloss mich dazu den Roboterarm aus drei Motoren zu bauen:

- ein Motor ist direkt an der Basis des Roboters montiert und fungiert als Schulter
- einer zweiter ist darüber montiert und fungiert als Ellbogen
- ein dritter Motor öffnet und schließt die Finger des Arms.

2. Auswahl der Motoren

Bei der Auswahl der Motoren musste ich mich zwischen Servomotoren, Schrittmotoren und Getriebemotoren entscheiden. Meine Entscheidung fiel auf die Servomotoren, da diese sich präzise steuern lassen und keine Steuerkarte benötigen, um vom Arduino gesteuert zu werden.

3. Integration des Arms

Der Roboterarm sollte direkt durch den Nutzer des Roboters steuerbar sein. Um dies zu erreichen, nutze ich den gleichen RC Empfänger wie für die manuelle Steuerung der Basis.



Abbildung 8: Roboterarm

3. Versuch zur Testung der Autonomie des Roboters in engen Räumen

Test 1: L-förmiger Raum mit einer Breite von 140cm und einer maximalen Länge von 400cm. Beide Roboter bewältigten den Raum ohne große Probleme, jedoch hatte Roboter 1 (radbetriebener Prototyp 1) Probleme sich rechtzeitig gerade auszurichten, wodurch er oft nah an die Wände kam. Roboter 2 (kettengeschriebener Prototyp 2) hatte keine Kollisionen mit den Wänden und hielt den programmierten Abstand meist ein, er zeigte jedoch Schwierigkeiten bezüglich der Orientierung, wenn die Sensoren nicht parallel zur Wand standen.

Test 2: Rechteckiger Raum mit in zufälliger Konstellation verteilten, kleinen Objekten, Raumbreite und Länge 290 cm · 250 cm. Beide Roboter fuhren erfolgreich durch diesen Raum, jedoch war Roboter 1 nicht in der Lage den Ausgang innerhalb von zwei Minuten zu finden, und fuhr für zwei Minuten und 45 Sekunden an den Wänden entlang, bis er dann den

Raum verließ. Robotertyp 2 absolvierte diesen Test hingegen perfekt, verließ den Raum nach 37 Sekunden und ohne Fehler der Sensorik.

Auch hier zeigten sich aber vor allem bei Roboter 1 Schwierigkeiten bezüglich der Orientierung, wenn die Sensoren nicht parallel zur Wand standen.

4. Versuch zur Testung der Autonomie des Roboters auf unebenem Gelände im Außenbereich

Test 1: Roboter 1 fuhr erfolgreich und ohne Kollisionen durch das Gelände und wich Bäumen erfolgreich aus. Er hatte jedoch Probleme, größere Steine zu überwinden aufgrund der zu schwachen Motoren.

Mit Roboter 2 konnte dieser Test nicht durchgeführt werden, da dieser erst im Dezember 2021 fertig gestellt wurde und das Wetter nicht für den Test geeignet war, jedoch plane ich diesen Test mit Roboter 2 im Frühling durchzuführen.

5. Versuch zur Ermittlung des Fehlerbereichs der Ultraschallsensoren

Während der Tests des autonomen Fahrens der Roboter bemerkte ich, dass die Ultraschallsensoren Probleme hatten, die korrekte Distanz zu messen, wenn der Sensor nicht parallel zur Wand oder zu den Oberflächen der Objekte war. Um dieses Problem zu lösen, bestimmte ich die maximale Distanz und den maximalen Winkel der Sensoren, für die der Sensor noch zuverlässig arbeitet.

Aufbau und Durchführung:

In einem Abstand von 15cm zum Ultraschallsensor wurde ein Stück Holz positioniert. Der Winkel, in dem das Holz zum Sensor steht, wird verändert und der Wert, den der Sensor ausgibt, notiert.

Winkel	Distanz
0 Grad	15cm
5 Grad	14cm
10 Grad	13cm
15 Grad	14cm
20 Grad	60cm
25 Grad	57cm
30 Grad	60cm
35 Grad	1182 cm
40 Grad	1182cm

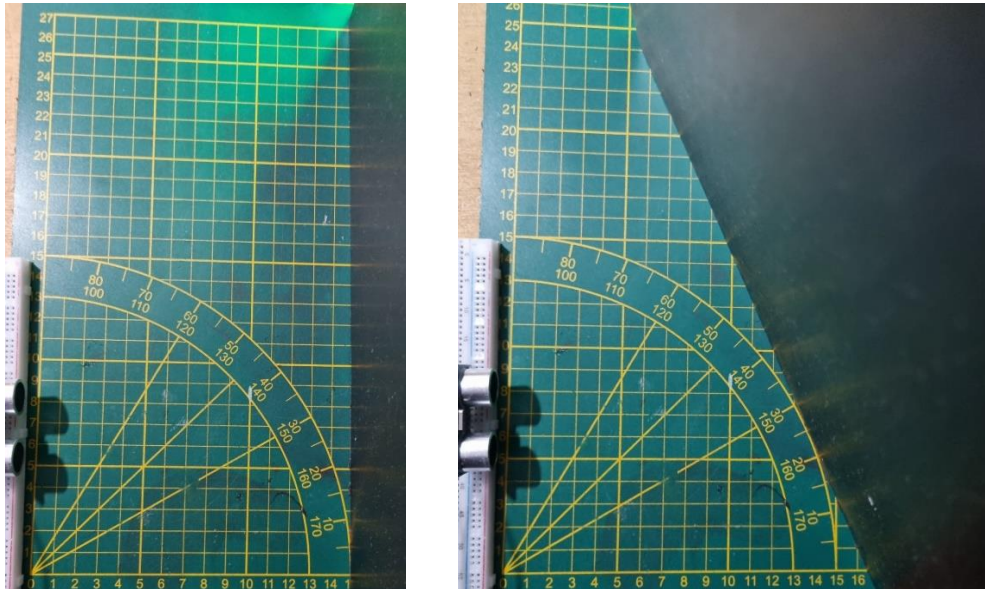


Abbildung 9: Sensoren Test (rechts gerade, links 20Grad Winkel)

Deutung: Der Test zeigt, dass bereits kleine Winkeländerungen das Ergebnis verfälschen. Kritisch wird es für den Roboter, wenn der Winkel zur Wand 20 Grad oder mehr beträgt. Ab diesem Punkt sind die Ergebnisse des Sensors nicht mehr brauchbar und sorgen für Probleme bei der Navigation des Roboters. Die maximale messbare Distanz der Sensoren beträgt ungefähr 400 cm, ab einem 401cm Abstand fängt der Arduino an, einen Abstand von 1182cm auszugeben. Solche Ungenauigkeiten stellen ein großes Problem für Roboter mit Ultraschallsensoren dar, was eine Änderung des Konzepts nötig macht.

6. Versuch zur Messung der Präzision des Roboters

Um das Ausweichen des Roboters zu verbessern, habe ich mit einem Versuch die Präzision des Roboters gemessen.

Aufbau und Durchführung: In dem einprogrammierten Abstand von 45cm von der Wand wird eine Markierung gemacht, der Roboter fährt auf die Wand und die Markierung zu, dabei wird der Abstand gemessen, den der Roboter noch zurücklegt. Der reale Abstand muss dabei noch korrigiert werden, da die Ultraschallsensoren 6 cm innerhalb des Roboters liegen.

Messungen	programmierter Abstand	realer Abstand	korrigierter Abstand
1	50cm	35cm	41cm
2	50cm	35cm	41cm
3	45cm	30cm	36cm
4	45cm	30cm	36cm
5	35cm	25cm	31cm
6	35cm	27cm	33cm
7	30cm	20cm	26cm
8	30cm	18cm	24cm

Auswertung: Betrachtet man die korrigierten Ergebnisse, erhält man eine durchschnittliche Abweichung des programmierten zum korrigierten Abstand von 6,5 cm. Diese Abweichung entsteht vermutlich durch die nicht absolut gerade Montage der Sensoren, wodurch die Strecke der Schallwellen etwas länger ist als der tatsächliche Abstand des Roboters zur Wand.

Interessanterweise ist diese Abweichung bei kleineren Abständen geringer. Dies liegt vermutlich daran, dass der Arduino ein anderes Timing erhält. Diese Ungenauigkeit des Timings entsteht meiner Ansicht nach durch den Code, da dieser keine Unterbrechung nutzt, sondern eine „while-Schleife“, welche selbst eine gewisse Zeit benötigt, bis sie sich wiederholt. Somit gibt es auch hier noch Bedarf zur Optimierung.

7. Ergebnisdiskussion

Nach derzeitigem Stand sind die von mir gebauten Roboter-Prototypen noch nicht für Landwirtschafts- und Forschungszwecke geeignet. In der nahen Zukunft möchte ich meinen Roboter für solche komplexen Aufgaben aber ausstatten und vorbereiten, weil ich eine große Zukunft für kleine Robotik Systeme sehe.

Derzeit beruht die Navigation meiner Roboter ausschließlich auf der stark limitierten Leistung des Arduino Mega und den drei Ultraschallsensoren. Wie ich in Abschnitt 6 und 7 zu zeigen versucht habe, liegt das größte Problem des Roboters in der Ultraschall-Sensorik, da diese recht unpräzise ist und Probleme mit ungeraden Oberflächen, bzw. verschiedenen Winkeln hat.

Ich bin jedoch zuversichtlich, dass ich in naher Zukunft ein Lidar verwenden kann. Bei einem Lidar handelt es sich um ein Raummessungsgerät, welches Laser verschickt, misst und dabei rotiert. Durch die Rotation werden die Abstände rund um den Roboter gemessen, diese Messungen werden vom Computer interpretiert und ein Algorithmus erstellt darauf basierend eine Karte. Dieses würde die aktuellen Probleme mit der Sensorik beheben, da es alle Abstände in einem Winkel von 360 Grad misst. Ein Lidar wird es dem Roboter erlauben intelligent zu navigieren und auch zu vorgegeben Orten zu fahren, da sich die Abstandsdaten gut speichern lassen und ihm damit eine intelligente Navigation erlauben.

Passend zum Lidar wäre eine weitere sinnvolle Modifikation, beziehungsweise Entwicklung, den Roboter mit einem Raspberry Pi oder einem Jetson Nano, statt mit einem Arduino auszustatten. Diese beiden Minicomputer sind um einiges leistungsstärker und das Robotik Betriebssystem ROS 2 lässt sich auf diesen verwenden, dieses Betriebssystem und die zusätzliche Leistung würden in Kombination mit einem Lidar intelligentes autonomes Fahren ermöglichen.

Des Weiteren möchte ich eine etwas größere Roboterbasis mit stärkeren Motoren konstruieren. Denn während des Baus des Roboters kam mir die Idee, dass der optimale Einsatzzweck von kleinen Robotern in der Aufklärung und Hilfe liegt. Ein etwas größerer, leistungsstärkerer Roboter könnte in zerstörte Gebäude oder mit Giftstoffen oder durch radioaktive Stoffe kontaminierte Gebiete fahren, dort Messungen durchführen oder Unterstützung leisten.

Ich sehe viele Anwendungszwecke für den Roboterarm, bei diesem kann der Greifer auch durch einen Feuchtigkeitssensor ausgetauscht werden. Dies ermöglicht es dem Roboter, Bodenproben zu nehmen und daher dabei zu helfen, die Ideale Feuchtigkeit des Bodens zu erreichen. Ebenfalls lässt sich auf dem Roboter durch eine 3d gedruckte Lochplatte leicht weiteres Messwerkzeug montieren wie z.B. eine Kamera oder Meteorologische Instrumente.

Mit den Resultaten dieser Forschungsarbeit bin ich zufrieden. Der Bau der Roboter hat mich zwar vor einige Herausforderungen gestellt, diese konnte ich aber gut bewältigen. Die Tests mit meinen beiden Prototypen haben interessante Informationen geliefert und neue Fragen aufgeworfen.

Mit meiner Forschungsarbeit an diesen beiden Prototypen habe ich mir eine gute Grundlage

für den Bau eines dritten Prototyps erarbeitet, der hoffentlich ein vollautonomer Roboter werden wird und der der ursprünglichen Idee gerecht wird.

8. ROS und Lidar Integration als nächster Entwicklungsschritt

Wie funktioniert ROS?

ROS ist ein Framework für Robotik-Programmierung auf Basis der Programmiersprachen Python und C++. Die Programmierung arbeitet auf Basis eines Publish and Subscribe Prinzips. Bei diesem Prinzip gibt es mehrere Nodes. Diese erfüllen bestimmte Aufgaben und können die Daten ausgeben, die von den anderen Nodes abonniert werden können. Dadurch können die Daten einer Funktion auf eine andere übertragen werden. Dies erlaubt es, eine komplexe und vielseitige Steuerung für den Roboter zu programmieren, die aus vielen einzelnen Systemen besteht.

Wie wird ROS integriert?

Zur Steuerung des Roboters nutze ich zwei Raspberry Pi's und zwei Arduino's. Auf den Raspberry Pi's läuft das Betriebssystem Ubuntu, auf welchem ROS läuft. Der erste Raspberry Pi auf dem Roboter ist mit einem Arduino verbunden. Der Arduino wird zur Steuerung der Motoren und Encoder genutzt. Die Daten der Encoder so wie die Daten für den Motor werden durch den Arduino umgewandelt und durch den Raspberry Pi gesteuert und interpretiert. Der Arduino funktioniert daher als ein Art Übersetzer zwischen den Raspberry Pi's und den elektrischen Komponenten des Roboters. Bei der Fernbedienung wird der zweite Arduino genutzt, um die Signale der Steuerung-Sticks für den zweiten Raspberry Pi zu übersetzen. Der Raspberry Pi interpretiert die Signale der Steuerungsticks und wandelt diese in ROS-Befehle um und veröffentlicht diese. Der Raspberry Pi auf dem Roboter hat diese Node abonniert und empfängt dadurch die Befehle der Steuerung.

Wie funktioniert ein Lidar?

Ein Lidar ist ein rotierender Laser-Sensor, welcher durch die Rotation Messungen in 360° machen kann. Diese Daten können vom Raspberry Pi ausgelesen und interpretiert werden.

Wie wird das Lidar integriert?

Die Daten des Lidars werden mit den Daten der Rotary Encoder kombiniert. Diese messen die Rotation der Motoren und die Messungen werden dann vom Arduino in Distanzen und Winkel umgerechnet. Die Kombination der Positionsdaten und der Lidar-Daten nennt sich SLAM(Simultaneous-Localization-and-Mapping). Hierbei wird das ROS-Paket RVIZ genutzt. Dieses ermöglicht die Kombination der Daten und eine Visualisierung. Durch den Nutzen von ROS und RVIZ lässt sich eine Karte der Umgebung erstellen und Bewegungsbefehle geben.

9. Zusammenfassung

Die Entwicklung von autonomen Robotern schreitet immer weiter voran, wobei diese immer günstiger werden. Dennoch sind diese immer noch teuer und meist recht groß. Der Bau von autonomen kleinen Robotern hingegen ist auch für Schüler wie mich möglich.

Derzeit plane ich mein absolutes Ziel eines vollautomatisierten Roboters mit intelligenter Autonomie zu realisieren und nutze dabei Jugend Forscht als Meilenstein dem Ziel näher zu kommen.

Ich begann mein Projekt mit der Prämisse einen Roboter zu bauen, der Objekten ausweichen kann und für die Forschung einsetzbar ist. Auch wenn ich das zweite Ziel nicht erreichte, ist der Roboter funktional und autonom, wodurch ich mein Hauptziel erreichte.

Zusätzlich sehe ich aber für die dritte Generation meiner Roboter mehr Einsatzzwecke, diese variieren von Aufklärung in Krisengebieten, der Rettung von Menschen, Fabriküberwachung zu z.B. Landwirtschafts-Unterstützung und Minen-Aufklärung.

10. Quellen und Literaturverzeichnis

- Q1: https://www.youtube.com/watch?v=gm0b_ijaYMQ
18.02.2021 20:15
- Q2: <https://www.sparkfun.com/tutorials/400>
22.02.2021 17:35
- Q3: https://www.uniulm.de/fileadmin/website_uni_ulm/iui.inst.050/vorlesungen/wis_e1011/les/Uebung_2.pdf
08.08.2021 17:48
- Q4: <https://learn.sparkfun.com/tutorials/pulse-width-modulation/all>
08.08.2021 18:12
- Q5: <https://dronebotworkshop.com/tb6612fng-h-bridge/>
04.09.2021 19:17
- Q6: <https://www.bostondynamics.com/products/spot>
03.04.2021 22:40
- Q7: <https://www.exp-tech.de/motoren/motorsteuerung/dc-motortreiber/8386/sparkfun-motor-driver-dual-tb6612fng-1a>
04.01.2022 12:29
- Q8: <https://www.jpl.nasa.gov/missions/mars-2020-perseverance-rover>
21.12.2021 23:14
- Q9: <https://gist.github.com/ShawnHymel/ccc28335978d5d5b2ce70a2a9f6935f4>
Basis für Radiowellen Code 07.08.2021
- Q10: <https://www.youtube.com/watch?v=IN8tjTk8ExI&t=312s>
Inspiration für Roboterarm 09.12.2021
- Q11: <https://wiki.ros.org/>
ROS Wiki
- Q13: <http://wiki.ros.org/rviz>
Rviz Dokumentation

10. Anmerkungen

3D Drucker: Anycubic I3 Mega

Schrittmotoren: Nema 17

Servos: Longrunner LDX218

Chassis und

Motoren: https://www.amazon.de/gp/product/B077Z4GYLB/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1