



SYSTEMS ENGINEERING GROUP
TU DRESDEN

ROBOLAB

Abschlussbericht

Gruppe L6:

Sergej HAHN
Sebastian KLUGE
Jörg THALHEIM

Betreuer:

Tobias WESCHENFELDER

28. März 2012

1 Ziel

Der Roboter soll in der Lage sein, einen vorgegebenen Parcours mit spezifizierten Hindernissen autonom in einer bestimmten Zeit zu bewältigen.

Die grundlegende Aufgabe hierbei ist es, dass der Roboter einer Linie folgt, welche Spitzkehren und Lücken enthalten kann.

Darüber hinaus soll er in der Lage sein, eigenständig Hindernisse zu umfahren und danach die zu verfolgende Linie wieder aufzunehmen.

2 Entwurf

2.1 Software

Das Programm ist folgendermaßen gegliedert: Beim Start des Programms öffnet sich ein Menü, indem mittels der integrierten Knöpfe zwischen 3 Programm-punkten gewählt werden kann:

- Kalibrierung (linker Menüknopf)
- Programm (mittlerer Menüknopf)
- Test (rechter Menüknopf)

Während der *Kalibrierung* werden halbautomatisch die Lichtsensoren auf die richtigen Kontrastwerte für die Linienerkennung konfiguriert. Dabei wird der arithmetische Mittelwert zwischen dem Untergrund und der Linie ermittelt. Des Weiteren werden die Zeiten gemessen, die für eine Umdrehung auf der Stelle benötigt wird und für einen fest definierten Kreis. Diese wird später zum Lenken benötigt.

Der Punkt *Programm* enthält die eigentliche Logik. Hier werden die Hauptroutinen konfiguriert und gestartet, welche den Roboter autonom fahren lassen.

Test beinhaltet verschiedene in sich geschlossene Module um den Roboter und seine angeschlossenen Sensoren/Motoren zu überprüfen. Während der Entwicklung wurden in diesem Unterprogramm auch neue Funktion/Subroutinen isoliert vom Hauptprogramm getestet.

Abbildung 1: Hauptmenu



2.1.1 Das Hauptprogramm

Das Hauptprogramm unterteilt sich in 2 Tasks: Dem *Linienfolgealgorithmus* und einer *Kollisionserkennung*.

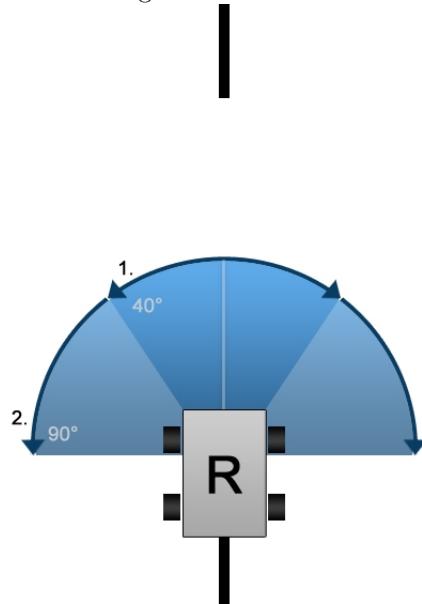
Der *Linienfolgealgorithmus* läuft die meiste Zeit sichtbar im Vordergrund und versucht die Strecke abzufahren.

Die *Kollisionserkennung* fragt im Wesentlichen den Status der beiden Tastsensoren ab. Wird einer der Sensoren gedrückt, wird der Linienfolgealgorithmus unterbrochen und versucht das vermeintliche Hindernis zu umfahren. Dafür wäre Multithreading nicht zwingend nötig gewesen, wenn die Sensoren per Interrupts abfragbar wären.

2.1.2 Linienfolgealgorithmus

Der Roboter fährt solange geradeaus, bis er die Linie verliert. Anschließend sucht der Roboter erst in einem Bereich von 80° auf der Stelle nach der Linie. Wenn keine Linie vorhanden ist, wird der Bereich auf 180° ausgeweitet. Der Roboter kann zudem leichte Kurven erkennen und in diesen mitlenken.

Abbildung 2: Suche an der Stelle



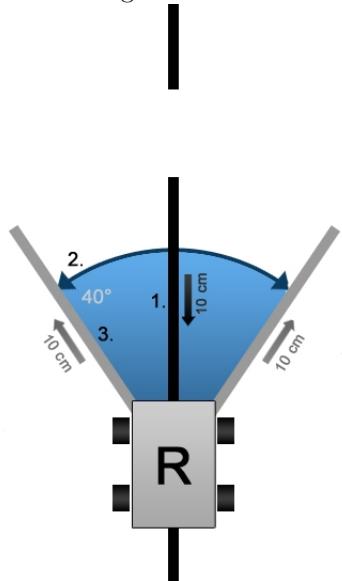
Falls dies keinen Erfolg bringt, sucht der Roboter nach Spitzkehren. Dazu setzt der Roboter 10cm zurück. Er dreht sich um 40° in beide Richtungen und fährt jeweils schräg 10cm vorwärts, bis er auf die Linie trifft.

Wenn keine Linie durch den Roboter erkannt wird, versucht der Roboter als letztes Mittel einen Kreis zu fahren. Dabei sollen Lücken und Sackgassen erkannt werden.

2.1.3 Hindernisumfahrung

Wird einer der Tastsensoren betätigt, versucht der Roboter dieses Hindernis zu umfahren. Hierfür fährt er ein Stück zurück und dreht danach um nach links.

Abbildung 3: Scharfe Kurve

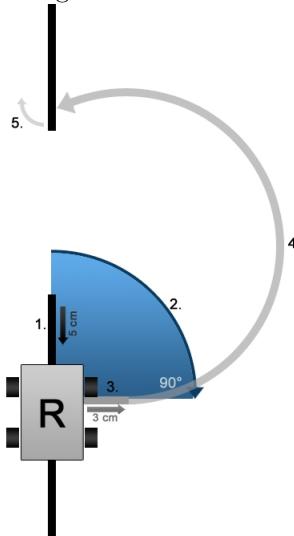


Danach fährt er vorwärts und schlägt danach einen Kreis nach rechts ein.

Werden nun die Tastsensoren wieder betätigt, wiederholt sich dieser Zyklus. Durch dieses Vorgehen tastet er sich um das Hindernis herum, bis er eine Linie erkennt.

In diesem Fall fährt der Roboter in einer Linkskurve bis er wieder auf die Linie trifft. Auf dieser bringt er sich wieder in Fahrtrichtung und führt seinen Linienfolgealgorithmus fort.

Abbildung 4: Lücke



2.2 Hardware

Der Roboter besteht aus:

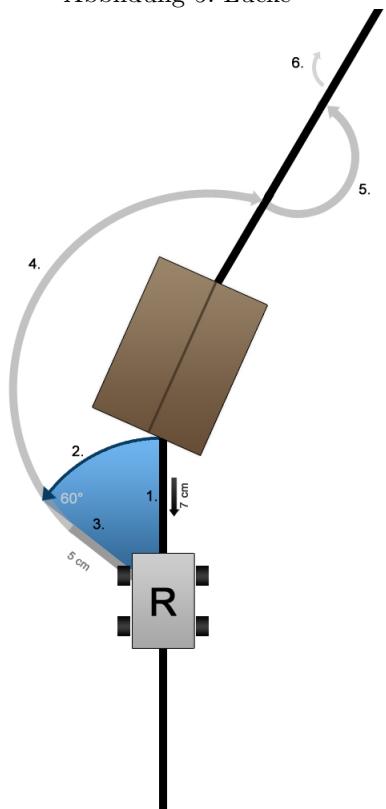
- Steuereinheit
- zwei Servomotoren
- zwei Tastsensoren
- Ultraschallsensor
- Helligkeitssensor

Die *Tastsensoren* sind, jeweils links und rechts, am vorderen Teil des Roboters angebracht. An ihnen ist eine breite, abgerundete *Stoßstange* befestigt. Diese sorgen dafür, dass der Roboter *Hindernisse* aus allen Winkeln schnell erfassen und somit umfahren kann.

Der *Lichtsensor* sitzt mittig nah an der *Antriebsachse*. Er befindet sich zirka einen halben Zentimeter über dem Boden, so dass etwaige Erschütterungen und Unebenheiten des Roboters keinen Einfluss auf die Messwerte des Sensors haben. Auf eine zusätzliche Abschirmung wurde verzichtet, da der Sensor im Raw-Modus schon gute Kontrastwerte liefert.

Die *Antriebsachse* mit den zwei Motoren befindet sich unten mittig am Roboter. Damit der Roboter sowohl stabil als auch wendig ist, wurde hinten mittig eine *Kugelhalterung* angebaut. Die eingefasste Kugel ist starr und dreht sich beim Fahren nicht. Diese Kugel hat gegenüber eines Stützrades den Vorteil, dass diese wendiger ist und sich in jeder Situation gleich verhält.

Abbildung 5: Lücke



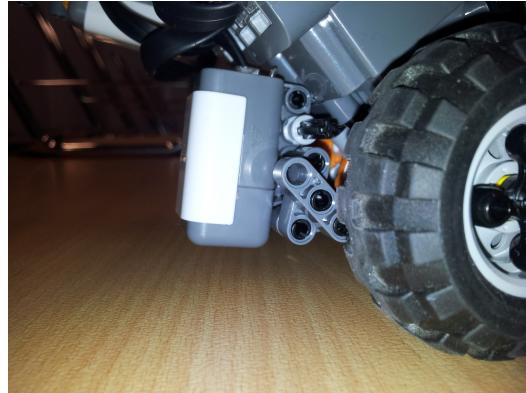
Der *Ultraschallsensor* sitzt mittig auf dem Roboter. Eine Funktion für diesen war geplant, wurde aber während der Entwicklung verworfen. Er dient also nur als Ergänzung des optischen Erscheinungsbildes.

Die *Steuereinheit* sitzt zentral auf dem Roboter, sodass der Roboter einerseits optimal ausbalanciert ist und andererseits die Tasten der *Steuereinheit* gut zugänglich sind.

Abbildung 6: Tastsensoren mit Stoßstange



Abbildung 7: Lichtsensor



Insgesamt ist der Roboter 30 cm lang, 21 cm breit und 19 cm hoch.

3 Projektverlauf und Erfahrungen

3.1 Lösung der Aufgabe

Als erstes haben wir die grundlegenden Aufgaben, welche an den Roboter gestellt wurden, *zusammengefasst*. Wir analysierten die Videos des ersten Kurs-Durchgangs. Dadurch bekamen wir einen guten Einblick, mit welchen Problemen die Roboter zu kämpfen haben und welche Lösungsansätze/Bauweisen sich bewährten.

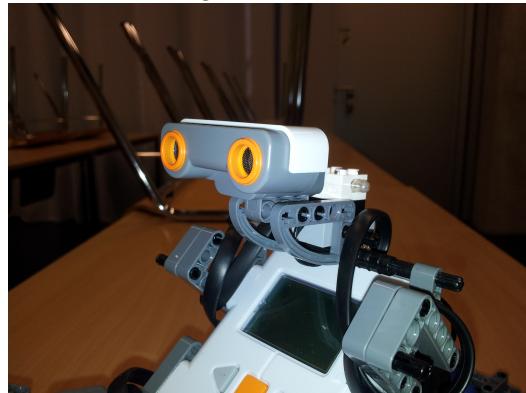
Nebenbei baute ein Teil der Gruppe aus den gegebenen Teilen des Bauatzes den Roboter, wobei *eigene Ideen* im gleichen Maße Einfluss fanden wie vorgegebene Bauanleitungen aus dem Bausatz oder dem Internet. Es wurden Funktionen für grundlegende Bewegungsformen wie Kurven, Kreisfahrt oder das Fahren einer bestimmten Länge programmiert.

Im Anschluss arbeiteten wir Schritt für Schritt die zu erfüllenden Aufgaben in das Programm ein, wobei *grundlegende Aufgaben* wie dem *folgen einer Linie* Vorrang vor Aufgaben wie der *Bewältigung von Lücken* oder *Spitzkehren* hatten.

Abbildung 8: Antriebsachse



Abbildung 9: Ultraschallsensor



Parallel hierzu testen wir permanent die eingearbeiteten Funktionen auf diversen Teststrecken, um etwaige Fehler im Programmcode schnell zu erkennen und zu beheben.

Dieser Teil war mit Abstand der zeitaufwendigste Teil unserer Arbeit, da erst hier auch kleine Fehler im Programm sichtbar wurden, welche erst beim intensiven Testen offensichtlich wurden.

Nach Fertigstellung des Roboters wurde dieser nochmals permanent getestet, um sicherzugehen, dass dieser *zuverlässig* und *fehlerfrei* die vorgegebenen Strecken absolvieren kann. Hierbei wurden nochmals auftretende Fehler ausgebessert.

3.2 Welche eurer Entscheidungen waren gut und welche nicht?

Es hat sich als positiv herausgestellt, dass wir den Roboter nicht stur nach einer vorgegebenen Bauanleitung bauten, sondern auch eigene Ideen mit einfließen ließen. Hierdurch konnten wir den Roboter vom Design her ganz nach unserem Geschmack bauen. Darüber hinaus war es ein großer Vorteil, dass wir permanent während der Entwicklung der Software nicht nur diese an den Ro-

Abbildung 10: Steuereinheit

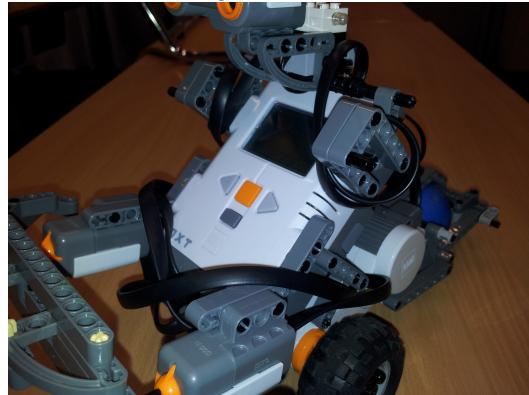


Abbildung 11: gesamter Roboter



boter anpassen konnten, sondern auch auftretende funktionelle Probleme durch bauliche Änderungen behoben werden konnten.

Eine negative Entscheidung war, dass wir unsere Änderungen bei der Behebung von Fehlern nicht dokumentierten, da somit nicht nachvollziehbar war, wodurch eventuell neu auftretende Fehler verursacht wurden. In diesem Fall mussten wir im Ernstfall komplexe Änderungen verwerfen und Teile des Programms überarbeiten, um den ursprünglich kleinen Fehler zu beheben.

3.3 Eigene Erkenntnisse

Mit unserem jetzigen Kenntnisstand würden wir uns vor dem Projekt einen *Arbeitsplan* aufstellen, in welchem die Verteilung der anstehenden Aufgaben auf die *einzelnen Gruppenmitglieder* festgelegt ist. Dies hätte dann den Vorteil, dass man anstatt gemeinsam alle Aufgaben zu bewältigen sich jeder Einzelne

auf bestimmte Aufgaben *spezialisieren* kann und somit im Endeffekt *besser* und *effizienter* gearbeitet werden kann.

3.4 Implementierung und Verbesserung

Zum einen wäre es möglich, die Lösung einzelner spezieller Aufgaben *effizienter* zu gestalten, insbesondere die der *Hindernisumfahrung*. Es besteht hierbei die Möglichkeit, neben den *Tastsensoren* auch den *Ultraschallsensor* mit einzubeziehen, da man durch diesen die Hindernisse nicht immer anstoßen muss, um sie zu erfassen, sondern diese bereits in einer gewissen Distanz erfasst und umfahren werden können. Hierdurch würde man die zu fahrende Strecke und auch Zeit minimieren, welche der Roboter zur Hindernisumfahrung benötigt. Ein Ultraschallsensor könnte zudem Spezialfälle wie Hindernisse, welche als Sackgassen aufgebaut sind, abdecken.

Zum anderen könnte man die *Kugelhalterung* verbessern, da diese in ihrer jetzigen Fassung die Kugel einklemmt und diese somit nicht rollen kann, sondern schleift. Eine ideale Kugelhalterung würde einerseits gewährleisten, dass die Kugel nicht aus ihr *herausspringt*, und andererseits dass diese rollen kann und nicht schleift.

Eine weitere Verbesserungsmöglichkeit wäre, die Position der *Steuereinheit* zu verändern, da in ihrer jetzigen Position der *Ladekabelanschluss* verdeckt ist und somit beim Laden des Roboters ein *Auseinanderbauen* unumgänglich ist.

4 Bewertung

4.1 Einschätzung und Nutzen des Projektes

Das Projekt war eine gute Gelegenheit, unsere ersten *Programmierkenntnisse* praktisch anzuwenden. Es war insbesondere sehr befriedigend, dass man die Auswirkung einer jeden Änderung im Programm sofort am Roboter sah, und somit ein direktes *Feedback* der eigenen *Erfolge* und *Misserfolge* bekam. Dadurch erhöhte sich auch der *Spaßfaktor*, weil ein stetiger Erfolg gut für die eigene Motivation war.

Darüberhinaus war es auch eine gute Gelegenheit, das gemeinsame Arbeiten an einem Projekt zu erlernen und dabei im Umgang mit Versionswaltungen wie *Git* sicherer und vertrauter zu werden.

Weiterhin wurde man beim Anfertigen des Abschlussberichtes vertrauter im Umgang mit *LATEX*. Dies ist von Vorteil, da Latex im *universitären Umfeld* sehr verbreitet ist und sich gut für Dokumentationen/Abschlussarbeiten eignet.

4.2 Verbesserungsvorschläge

Da den Meisten der Umgang mit *NXC*, *LATEX* und *Git* nicht vertraut war, und es zudem keine Einführung in diesem Sinne gab, wurde man mehr oder weniger „ins kalte Wasser geschmissen“. Den Umgang mit diesen Programmen und Programmiersprachen mussten wir uns selber aneignen, wodurch ein Teil der zur Verfügung stehenden Zeit verloren ging. Somit wäre eine kurzer *Crashkurs* zum Beispiel im Rahmen eines *Workshop* wünschenswert. Diese könnte in Form von

einzelnen Gruppen durch die jeweiligen *Tutoren* am Anfang des Kurses stattfinden. Da hier gegenüber Vorlesungen der Vorteil besteht, dass *Unklarheiten* und *Fragen* sofort diskutiert werden können.