**AEP Gruppe B5 – SoSe2017**

Beschreibung der Praktikumsaufgabe

Das Modul Informatik 2 im zweiten Semester des Studiengangs Mechatronik beinhaltet ein Praktikum, das sich mit dem Thema des autonomen Fahrens beschäftigt. Dazu wurden die Aufgaben gestellt ein Fahrzeug zu konstruieren (Lego) und einen Algorithmus zu implementieren, der dieses Fahrzeug autonom in einer Längsparklücke einparken lässt. Als Grundlage des technischen Aufbaus diente die Lego Mindstorms Produktserie. Die Programmierung wurde mit der Software Matlab realisiert.

Team und Aufgabenverteilung

Anna Blankenstein: Programmierung, Dokumentation

Felix Bruchhage: Programmierung, Konstruktion, Lego Digital Designer

Luca Di Lillo: Programmierung, Konstruktion, Lego Digital Designer

Hardware

Zunächst musste ein PKW-Chassis mit Hilfe der Lego Mindstorm Komponenten aufgebaut werden. Dabei soll sich die Konstruktion möglichst nah an einem realen PKW orientieren. Das Fahrzeug verfügt über eine lenkbare Vorderachse, dessen maximales Lenkwinkel 40 ° beträgt. Als Antriebsachse wurde die Hinterachse gewählt. Zudem sollen ein Ultraschallsensor sowie ein Gierratensensor sinnvoll am Auto angebracht werden.

Die Lenkung wurde über eine direkte Übersetzung mit einem Motor an der Lenkachse realisiert. Diese Lösung wurde gewählt um ein möglichst geringes Lenkspiel zu gewährleisten und um in der späteren Programmierung keine Umrechnung zwischen Motordrehung und Lenkwinkel berücksichtigen zu müssen.

Für einen kleineren Wendekreis wurde an der hinteren, angetriebenen Achse ein Differential angebracht. Dieses ermöglicht es, dass die Räder mit unterschiedlichen Geschwindigkeiten drehen können.

Der Ultraschallsensor wurde auf der rechten Seite im unteren Bereich des Chassis mittig zwischen den Achsen befestigt. Dieser wird genutzt um Abstände zu Objekten erfassen zu können. Er wird dazu eingesetzt, eine Parklücke zu erkennen. Er registriert, wenn der seitliche Abstand groß genug ist um einzuparken und vermisst diesen Bereich automatisch beim Vorbeifahren. So kann er zudem erkennen, wann eine Parklücke ausreichend lang ist, bzw. wann sie zu schmal für den Einparkvorgang ist.

Der Gyrosensor ist waagerecht auf der anderen Seite möglichst nah am Fahrzeug-Schwerpunkt befestigt. Er detektiert geringste Winkeländerungen um die vertikale Achse und ermöglicht es, eine geregelte Geradausfahrt zu realisieren und zu erkennen in welchem Winkel das Fahrzeug zur Fahrspur bzw. Parklücke steht.

Die Motoren und Sensoren wurden mit dem Hauptrechner des Roboters, dem sog. NXT, verbunden. Dieser ist mittig auf dem Fahrzeug montiert, um ihn leicht zugänglich zu machen.

Software

Als Software um den Parkalgorithmus zu programmieren wurde die Toolbox der RWTH Aachen MATLAB genutzt. Dieses Programm ermöglicht es, Algorithmen zu entwickeln, die sowohl mit spezifischen Befehlen für den Lego-Rechner, als auch mit bekannten Kontrollstrukturen wie „for“- oder „while“-Schleifen aus Programmiersprachen wie C++ arbeiten. Im Vergleich zu den klassischen NXT Programmen, ist hier eine theoretisch unbegrenzte Programmgröße möglich. Des Weiteren bietet Matlab (Toolbox der RWTH Aachen) bereits implementierte komplexe numerische Funktionen und Algorithmen an, wie z.B. eine Routine für das Kalibrieren und Initialisieren des Gyroskops. Die erforderlichen Toolboxen und Bibliotheken lassen sich ganz einfach in der Programmierumgebung integrieren und nutzen. Hilfreich sind dabei auch die gut dokumentierten Befehlsbeschreibungen, die sich per Rechtsklick öffnen lassen, sowie die detailliert erläuterten Erklärungen und Beispiele, die sich auf der Internetseite Mathworks.com oder mindstorms.rwth-aachen.de finden lassen.

In den letzten beiden Terminen wurde die Programmier-Software geändert. Anstatt dem dateiorientierten modularen Aufbau des Codes in MATLAB wurde nun das Simulationswerkzeug Simulink verwendet. Simulink ist eine Toolbox aus der Umgebung Matlab, die ein Editierfenster als Oberfläche bietet. Dort können sogenannte Funktionsblöcke per Drag and Drop eingefügt werden und logisch untereinander verbunden werden. Der Vorteil liegt darin, dass Simulink speziell konzipiert und daher gut geeignet ist für die Simulation dynamischer Systeme. Das Aussehen der programmierten Gesamtsysteme ähnelt den, aus der Systemtechnik bekannten, Blockschaltbildern, die zur Veranschaulichung von technischen Zusammenhängen verwendet werden. Da eine ergebnisreiche Realisierung mit Simulink aus Gründen der geringen Zeitfenster nicht mehr möglich war, entschieden wir unser Programm weiterhin mit Matlab zu gestalten. So konnten wir einige Verbesserungen umsetzen und den Algorithmus so zuverlässiger arbeiten lassen.

Einparkkonzept

Strategie zur Geradeausfahrt und Lückenerkennung

Das Fahrzeug fährt zu Beginn parallel zur geraden Fahrspur. Damit keine ungewollten Querabweichungen entstehen, wird die Geradeausfahrt über Werte des Gyroskops geregelt und bei Bedarf gegengelenkt. Zur Ermittlung der Lenkgeschwindigkeit und -richtung, wird ein KI-Regler eingesetzt. Dabei detektiert es über den Ultraschallsensor den Abstand zu Objekten, die rechts neben ihm sind. Wird der Abstand groß genug für das seitliche Einparken, so wird angenommen, dass eine potentielle Parklücke beginnt. Zu diesem Zeitpunkt wird der interne Motor-Drehgeber zurückgesetzt und misst die gefahrene Strecke bis der zurückgelegte Weg 55 cm beträgt. Ist dieser Fall eingetreten hat es eine passende Lücke zum Einparken gefunden. Um das eventuell neben ihm stehende Fahrzeug nicht zu touchieren fährt das Auto um ca. die halbe Fahrzeuglänge zusätzlich nach vorn. Wird vorher ein nahestehendes Objekt erkannt, wird die Parklücken-Suche unterbrochen und der Algorithmus startet von vorn.

Strategie zum Einparken

Nach der oben genannten ersten Phase des Einparkvorgangs wird bis zum maximalen Lenkeinschlag nach rechts eingelenkt. Im Anschluss beginnt die Rückwärtsfahrt. Hat das Fahrzeug einen Winkel von 40 ° bezogen auf seine Ausgangsposition erreicht, wird während des Fahrens maximal nach links eingelenkt. Das Fahrzeug stoppt, wenn es einen Winkel von 0 ° erreicht hat, also parallel zur Fahrspur in der Parklücke steht. Zum Schluss wird die Lenkung gerade ausgerichtet und bis zur Mitte der Lücke vorgefahren. Zu diesem Zeitpunkt endet das Programm.

Fazit

Durch das Informatik-Praktikum 2 konnten die Fertigkeiten in der Programmierung vertieft und an einem realitätsnahen Objekt angewendet werden. Außerdem flossen in dem Projekt mehrere Teilgebiete der Mechatronik zusammen. Es musste darauf geachtet werden, dass der mechanische Aufbau mit den elektronischen Komponenten und den später implementierten Algorithmen verträglich war. Weiterhin mussten wir uns bereits in Teile der Regelungstechnik einarbeiten. Die Lösungen für Teilprobleme und Aufgaben waren nicht immer leicht zu finden. Doch diese Schwierigkeiten verhalfen uns schließlich zu einem tieferen Verständnis und auch der Austausch mit Kommilitonen trug zu weiteren Verbesserungen bei. Durch die intensive Arbeit an dem Projekt konnten vielfältige Kenntnisse gewonnen werden. Diese Grundkenntnisse werden uns für die folgenden Semester weiterhin hilfreich sein. Auch die genutzten Programme werden häufig in der Industrie eingesetzt und können uns in unserem zukünftigen Beruf nützlich sein.