

Zubaidullah Arsalan

Reglung von autonomen Modell-Lastfahrzeugen zum
selbständigen Fahren und Überholen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Angewandte Informatik
am Studiendepartment Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer.nat. Stephan Pareigis
Zweitgutachter : Prof. Dr. rer.nat. Reinhard Baran

Abgegeben am 22.Februar 2006

Zubaidullah Arsalan

Thema der Bachelorarbeit

Reglung von autonomen Modell-Lastfahrzeugen zum selbständigen Fahren und Überholen

Stichworte

Autonome Modell-Fahrzeuge, Sensorik, P-Regler, geregeltes Fahren, Überholmanöver, Subsumptions-Architektur, Aksen-Board

Kurzzusammenfassung

In der vorliegenden Bachelorarbeit wurde ein autonomes Modell-Lastfahrzeugs entwickelt, der selbstständig entlang der Wand fährt, dabei Kollisionen mit Hindernissen vermeidet und in der Lage ist, ein Überholmanöver durchzuführen. Für die Regelung der Fahrtrichtung des Modell-Lastfahrzeugs wurden die Methoden der Regelungstechnik eingesetzt. Ein P-Regler regelt die Fahrtrichtung so, dass das Modell-Fahrzeug mit einem vorgegebenen Sicherheitsabstand parallel zur Wand fährt. Als Steuerungssystem des autonomen Modell-Lastfahrzeugs wurde die Verhaltensbasierte Subsumptions-Architektur von Brooks eingesetzt. Das Steuerungssystem ist in der Programmiersprache C implementiert. Das Rechensystem des autonomen Modell-Fahrzeugs ist der Mikrocontroller Aksen-Board.

Zubaidullah Arsalan

Title of the paper

Controlling of model-trucks for autonomous driving and overhauling

Keywords

Autonomous model-trucks, sensors, P-controller, controlled driving, overhauling, subsumptions-architecture, Aksen-Board.

Abstract

This paper presents the development of a model-truck which is able to autonomously drive, avoid collision with balks, and overhaul. For the controlling of the driving course of the model-truck the methods of the control theory are applied. A P-controller makes the vehicle drive parallel to the wall by keeping a given safety distance. The subsumptions architecture by Brooks is applied as control system for the autonomous model-truck. This control system is implemented in C programming language. As calculation system for the model-truck the microcontroller Aksen-Board is applied.

INHALTSVERZEICHNIS

ABBILDUNGSVERZEICHNIS	V
1. EINLEITUNG.....	1
1.1. Problemstellung	1
1.2. Aufgabenstellung und Abgrenzung.....	2
2. THEORETISCHE GRUNDLAGEN	3
2.1. Autonome mobile Systeme	3
2.1.1. Autonome Fahrzeuge	4
2.1.1.1. Sensorik des autonomen Fahrzeugs	4
2.1.1.2. Sehsysteme	5
2.1.1.3. Radarsysteme.....	5
2.1.1.4. Infrarot-Distanzsensoren	5
2.1.1.5. Ultraschallsensoren	6
2.1.1.6. Robustheit gegen Sensordefekte.....	7
2.1.1.7. Messabweichungen der Sensoren.....	7
2.1.2. Steuerungsarchitektur autonomer Fahrzeuge.....	8
Subsumptions-Architektur	10
2.2. Regelungstechnik.....	14
2.2.1. Regelkreis	14
2.2.2. Regeltypen.....	16
2.3. Ausgleichproblem	20
3. HARDWARE.....	22
3.1. Die Lastwagen.....	22
3.2. Das Aksen-Board	26
3.3. Die Distanzsensoren.....	27
4. LÖSUNGSENTWURF	29
4.1. Wandverfolgung	29
Regler als Bahncontroller	30

4.2. Kollisionsvermeidung	33
4.3. Das Überholmanöver	34
4.4. Steuerungsarchitektur des Systems	35
4.4.1. Entwurf des Systems nach der Subsumptionsarchitektur	36
4.4.2. Definition der Kompetenzebenen.....	37
5. IMPLEMENTIERUNG.....	39
5.1. Systemarchitektur.....	39
5.2. Realisierung des Steuerungssystems	45
6. EXPERIMENTELLE ERGEBNISSE	53
7. ZUSAMMENFASSUNG UND AUSBLICK.....	61
7.1. Zusammenfassung	61
7.2. Ausblick.....	62
LITERATURVERZEICHNIS.....	64
ANHANG.....	66

ABBILDUNGSVERZEICHNIS

Bild 2-1: Ausbreitung und Reflexion der Ultraschallsignale.	6
Bild 2-2: Ultraschall-Entfernungssensormodul SRF05[SRF0-05].	7
Bild 2-3:Funktionsorientierte Steuerungsarchitektur für autonome Fahrzeuge.....	9
Bild 2-4:Parallel laufende Kompetenzebenen[BROO-89].	10
Bild 2-5: Hemmung und Unterdrückung der Ein- und Ausgangssignale.....	11
Bild 2-6:Automatenmodell der Kompetenzebene K_0 [BROO-89].	12
Bild 2-7: Automatenmodell der Kompetenzebene K_0 und K_1 [BROO-89].....	12
Bild 2-8: Blockschaltbild einer Steuerung[SCHU-05].....	15
Bild 2-9: Blockschaltbild einer Regelung[SCHU-05].....	15
Bild 2-10: Blockschaltbild eines Regelkreises.	16
Bild 2-11: a) Sprungantwort des P-Reglers, b) Sprungantwort des I-Reglers.....	18
Bild 2-12: Sprungantwort des PI-Reglers auf plötzliche Änderung der Regeldifferenz.	19
Bild 3-1: Modell-Lastwagen.	22
Bild 3-2: Kennlinie der Geschwindigkeit in Abhängigkeit von der Fahrstufe.	24
Bild 3-3:Das Aksen-Board.....	26
Bild 3-4:Arbeitsweisen der Distanzsensoren GP2D.	27
Bild 3-5:Ausgangsspannung abhängig von der Entfernung des Reflektionspunktes.	28
Bild 4-1: Aufteilung des Erfassungsbereichs des Sensors in verschiedenen Zonen.	29
Bild 4-2: Blockschaltbild des P-Reglers.	31
Bild 4-3: Stellung des Modell-Lastwagens bei der Regelabweichung $e = 0$	32
Bild 4-4: Stellung der Modell-Fahrzeuge vor dem Überholmanöver.	34
Bild 4-5:Sicherheitsabstände zwischen den Fahrzeugen.....	34
Bild 4-6: Parallellaufende Kontrollschichten.	36
Bild 4-7: Automatenmodell des Steuerungssystems.....	38
Bild 5-1: Architektur des Systems	39
Bild 5-2: Schaltungsaufbau der analogen Eingänge.	40
Bild 5-3: Bestimmung der Sensorwerte abhängig von der Entfernung.....	41
Bild 5-4: Kennlinie der Sensorwerte abhängig von der Entfernung.	41
Bild 5-5: Kennlinie der Funktion (5.1.3).....	42
Bild 5-6: Prozesswechsel nach Round-Robin-Algorithmus.	44
Bild 5-7: Spurwechsel zum Überholen.....	48

Bild 5-8: Erfassung des LKW ₁ durch den Sensor von LKW ₂	49
Bild 5-9: Sprung des Istwert von Sseitlich auf S _{Überhol}	49
Bild 6-1: Bahnverlauf des Modell-Lastwagens bei K _p = 4.	53
Bild 6-2: Bahnverlauf des Modell-Lastwagens bei K _p = 6.....	54
Bild 6-3: Bahnverlauf des Modell-Lastwagens bei K _p = 8.....	54
Bild 6-4: Bahnverlauf des Modell-Lastwagens bei K _p = 12.....	55
Bild 6-5: Bahnverlauf des Modell-Lastwagens bei K _p =16.	55
Bild 6-6: Plötzliche Änderung der Regeldifferenz beim Erreichen des Türbereichs.....	57
Bild 6-7: Erfassungspunkte des Sensors falls der LKW schräg zur Wand ist.....	58
Bild 6-8: Erfassung des hinteren Teils des LKW ₁ durch den Sensor von LKW ₂	59
Bild 7-1: Berechnung der Winkel des Fahrzeugs relativ zur Wand mit der Hilfe zwei Distanzsensoren.....	62
Bild 7-2: Spurwechsel nach dem Überholen.	63
Bild A-1: Die Verkablung des Modell-Lastwagens mit dem Aksen-Board.	71
Bild A-2: Wandverfolgung	71
Bild A-3: Spurwechsel zum Überholen.....	72
Bild A-5: die Verzeichnis-Struktur der CD.	73

1. Einleitung

1.1. Problemstellung

Autonome Fahrzeuge haben schon immer die Phantasie der Menschen angeregt. Dank vermehrter Forschungsaktivitäten in letzter Zeit sind jetzt die Voraussetzungen geschaffen, diese Phantasie in die Realität umzusetzen.

In der Automobilindustrie entwickelt sich die Technik für autonome Fahrzeuge rasant. Schon heute sind Fahrzeuge mit so genannten „Fahrassistenzsystemen“ ausgerüstet. Die Fahrassistenzsysteme übernehmen verschiedene Aufgaben und erleichtern die Arbeit des Fahrers, zum Beispiel Abstandssensoren zur Einhaltung des Sicherheitsabstands zwischen Fahrzeugen oder Systeme, die in die Fahrzeugführung eingreifen, wie Elektronisch geregeltes Antiblockiersystem (ABS), Elektronisches Stabilitätsprogramm (ESP) [MAND-06], Adaptive Cruise Control¹ (ACC).

Diese Systeme sind seit Jahren erfolgreich in Betrieb. Aber es wird noch eine gewisse Zeit dauern, bis die voll autonomen Fahrzeuge die Arbeit der Fahrer komplett übernehmen. Ein Meilenstein bei der Entwicklung von voll autonomen Fahrzeugen ist die Entwicklung von Steuerungssystemen, die das Fahrzeug in die Lage versetzen, entlang einer vorgegebenen Spur zu fahren und dabei Kollisionen mit Hindernissen zu vermeiden.

Diese Arbeit soll einen Beitrag zur Entwicklung von solchen Systemen leisten. In Rahmen dieser Arbeit wird ein vollautonomes Modell-Lastfahrzeug entwickelt, das entlang einer vorgegebene Spur fährt, dabei Kollisionen vermeidet und in der Lage ist ein Überholmanöver durchzuführen.

Der Anwendungsbereich solcher voll autonomen Fahrzeuge könnte die Anlieferung und Weiterreichung von Rohmaterialien oder Endprodukten sein. Diese Fahrzeuge wären in der Lage, selbständig zu den Betriebsstätten zu fahren und dabei Kollisionen mit Hindernissen zu vermeiden. Die Betriebsflächen könnten mit bestimmten Merkmalen versehen werden, damit die Fahrzeuge sich präzise orientieren können.

¹ Ein Abstandsregel-Tempomat das vorausfahrende Fahrzeuge erkennt, deren Geschwindigkeiten ermittelt und durch Brems- und Motoreingriffe einen gewünschten Abstand einhält

1.2. Aufgabenstellung und Abgrenzung

Die Welt, in der sich ein im Rahmen dieser Arbeit entwickeltes autonomes Modell-Fahrzeug bewegt, ist eine eindimensionale Welt. Die Bewegungsmöglichkeiten des Fahrzeugs sind auf das Fahren entlang eines Korridors beschränkt. Für die Realisierung des autonomen Modell-Lastfahrzeugs sind folgende Anforderungen dieser Arbeit gestellt:

- ◆ Es soll ein Verfahren aus dem Bereich der Regelungstechnik gefunden werden, mit dessen Hilfe das Lenkgetriebe des autonomen Modell-Lastfahrzeugs so geregelt wird, dass das Fahrzeug möglichst parallel zur Wand fährt.
- ◆ Das autonome Modell-Lastfahrzeug soll Hindernisse erkennen und darauf reagieren können, um Kollisionen zu vermeiden.
- ◆ Das Modell-Fahrzeug soll auch ein Überholmanöver durchführen können, falls ein langsames Fahrzeug vor ihm fährt.

Es soll ein Steuerungsmodell entwickelt werden, das das Modell-Lastwagen die Lage versetzt, die oben genannten Anforderungen gerecht zu werden.

Für diese Anforderungen soll zunächst ein auf theoretische Überlegungen basierendes Lösungskonzept entwickelt werden. Eine besondere Rolle kommt dabei den Methoden der Regelungstechnik zu. Durch das Einsetzen der Regelungstechnik soll die Fahr- richtung der Modell-Fahrzeuge geregelt werden. Das Konzept soll anschließend implementiert und in Experimenten getestet werden.

2. Theoretische Grundlagen

In ersten Abschnitt dieses Kapitels werden die theoretischen Grundlagen der autonomen Fahrzeuge dargelegt. Zunächst erfolgt eine Abgrenzung des Begriffs der autonomen Fahrzeuge. Dabei wird auch auf die Begriffe „autonome mobile Systeme“ und „autonome mobile Roboter“ eingegangen. Anschließend werden die wichtigsten Komponenten der autonomen Fahrzeuge erläutert. Im zweiten Abschnitt wird zuerst das Einsatzgebiet der Regelungstechnik dargelegt, dann der Regelkreis und einige Regeltypen beschrieben. Im dritten Abschnitt wird die „kleinsten Fehlerquadratmethode“ zur Bestimmung einer mathematischen Funktion mit minimalem Fehler für eine Reihe von Messpunkten beschrieben.

2.1. Autonome mobile Systeme

Der Begriff der autonomen mobilen Systeme ist sehr weit gefasst. Eine mögliche Definition lautet:

Autonome Systeme sind technologische Systeme, die selbstständig ihre Umwelt analysieren, sich anpassen, lernen, sich selbst organisieren, fortbewegen und auf Basis eigenständiger Schlussfolgerungen agieren und reagieren, ohne hierzu Eingriffe des Menschen zu benötigen[ARCG-05].

Dagegen ist der Begriff „autonome mobile Roboter“ viel präziser. Nach Hoppen ist ein autonomer mobiler Roboter *eine Maschine, der sich in einer natürlichen Umgebung aus eigener Kraft und ohne Hilfestellung von außen bewegt und dabei ein ihm gestelltes Ziel erreicht. Dabei erfasst der Roboter seine Umgebung selbstständig über eigene Sensoren, speichert das erkannte Wissen über die Umwelt, ermittelt ihre Position innerhalb der Umwelt und entwickelt selbstständig Pläne, sich innerhalb dieser Umwelt zu bewegen. Sie muss in der Lage sein, eine vorgegebene Zielposition sowohl in bekannter als auch in unbekannter Umgebung zu erreichen und auf dem Wege dorthin Kollisionen mit feststehenden oder beweglichen Hindernissen zu vermeiden [HOPP-92].*

Ein autonomes Fahrzeug besitzt im Gegensatz zu einem autonomen mobilen Roboter eingeschränkter Funktionalitäten. Arkin definiert ein autonomes Fahrzeug als eine Maschine, die in der Lage ist, Informationen aus ihrer Umwelt zu extrahieren und Wissen aus ihrer Welt zu nutzen, um sich in einer sinnvollen und zielgerichteten Weise sicher fortzubewegen[ARKI-98].

Die Grundprinzipien der autonomen Fahrzeuge, autonomen mobilen Roboter und anderen autonomen mobilen Maschinen sind gleich. Sie alle besitzen die Fähigkeit, selbstständig ihre Umwelt zu erfassen und zu analysieren, eigenständige Entscheidungen zu treffen und somit ohne externe Hilfe zu agieren und zu reagieren. Im Rahmen dieser Arbeit werden für die Realisierung eines autonomen Modell-Fahrzeugs auch die Techniken aus der Welt der Roboter benutzt.

2.1.1. Autonome Fahrzeuge

Autonome Fahrzeuge werden für unterschiedliche Einsatzgebiete und Anforderungen entwickelt. Im Bereich der Transportsysteme werden die autonomen Fahrzeuge eingesetzt, um in einem fest vorgegebenen Streckennetz sich autonom zu bewegen und Transportaufträge zu erledigen. Man spricht dabei von fahrerlosen Transportsystemen. Das Einsatzgebiet der fahrerlosen Transportsysteme ist die großräumige, flurgebundene Fertigungsumgebung. Fahrerlose Reinigungsfahrzeuge sind zur Pflege von Fußböden in großräumigen Gebäuden konzipiert[GUTS-94]. Autonome Fahrzeuge werden auch für den Einsatz in gefährlichen oder für gesundheitsschädliche Katastrophengebiete entwickelt.

Um ihre Aufgaben zu erfüllen, brauchen die autonomen Fahrzeuge Aktoren und Sensoren, Hardware und Software sowie eine geeignete Systemarchitektur. Die Aktoren der autonomen Fahrzeuge sind zum Beispiel die Lenkung, der Motor und das Schaltgetriebe. Die Funktionen der Aktoren sind vernachlässigenswert, deswegen wird in Rahmen dieser Arbeit die Arbeitsweise der Aktoren nicht näher erläutert.

Intensiv wird in dieser Arbeit auf Sensorik und die Steuerungsarchitektur autonomer Fahrzeuge eingegangen.

2.1.1.1. Sensorik des autonomen Fahrzeugs

Wie der Mensch für die Wahrnehmung seiner Umgebung auf seine Sinnesorgane angewiesen ist, so brauchen die autonomen mobilen Fahrzeuge Sensoren, um Informationen von ihrer Umgebung zu beziehen.

„Sensoren sind Umformer, mit denen sich verschiedene in Allgemeinen nicht elektronische Größen erfassen und auf elektrische Signale abbilden lassen. Den dafür notwendigen Zusammenhang zwischen Messgröße und Messsignal bestimmt das Messprinzip. Die Messsignale können dann mit den Methoden der elektrischen Signalverarbeitung ausgewertet werden“ [SCHR- 90].

Mit Hilfe der Sensoren können bestimmte physikalische oder chemische Eigenschaften, z. B. Entfernung, Temperatur, Feuchtigkeit, Druck oder Helligkeit als Messgrößen erfasst werden.

Die Wahrnehmung der Umwelt ist ein zentraler Baustein der autonomen Fahrzeuge. Um komplexe Anweisungen sicher und aufgabengerecht durchzuführen, müssen die autonomen Fahrzeuge mit Sensoren ausgestattet sein. Die Sensorik der autonomen Fahrzeuge ist in interne und externe Sensoren unterteilt. Interne Sensoren erfassen interne Größen wie zum Beispiel Radgeschwindigkeit und hitzeempfindliche elektronische Subsysteme. Externe Sensoren erfassen die Umgebung des autonomen Fahrzeugs[SOIK-98]. Sie werden eingesetzt, um zum Beispiel Fahrbahnverlauf, Hindernisse und Objekte zu erkennen.

Im Folgenden werden einige externe Sensorsysteme für die Erkennung des Fahrbahnverlaufs und von Hindernissen und Objekten vorgestellt.

2.1.1.2. Sehsysteme

Sehsysteme nehmen mit Kameras Bilder aus ihrer Umgebung auf und verarbeiten die Bilder nach Merkmalen. Aus diesen Merkmalen werden im weitesten Sinne Objekte wie Straßen, Fahrbahnmarkierungen, Verkehrsbeschilderungen und Hindernisse erkannt.

Diese Technik ist für die autonomen Personenfahrzeuge, die in der Stadt zum Einsatz kommen, unverzichtbar. Diese Art von Sensoren sind rechnergestützt und erfordern höchste Rechnerleistung, damit sie den Echtzeitanforderungen gerecht werden [PELL-03].

2.1.1.3. Radarsysteme

Die Radartechnik wird seit langem in der Flugüberwachung eingesetzt. Ein Radarsystem besteht aus einem Sender, einem Empfänger und einer Antenne. Der Sender strahlt elektromagnetische Wellen über die Antenne ab. Treffen die Wellen auf Objekte, werden sie von den Objekten partiell reflektiert. Die Antenne empfängt die reflektierten Strahlen und reicht sie als Signale an den Radarempfänger weiter. Der Empfänger verstärkt die reflektierten Signale und verarbeitet sie. Aus dem entstandenen Signal kann dann ein Rückschluss auf die Position der Objekte gezogen werden. Die Radarsysteme werden seit 1935 für die Flugüberwachung und für die Ermittlung der Entfernung der Flugzeuge eingesetzt. Heute wird dieser Technik auch von Automobilindustrie für Fahrerassistenzsysteme verwendet [DAIM-06]. Die Radarsensoren helfen den Fahrern beim Spurwechsel oder Überholen, beim Einhalten des Sicherheitsabstands zwischen Fahrzeugen (besonders auf Autobahnen) und beim Parken. Sie messen den Anstand des Fahrzeugs relativ zu anderen Fahrzeugen oder Objekten. Der Fahrer wird beim Überschreiten des Sicherheitsabstands gewarnt.

Die Radarsensoren, die in Fahrzeugen eingesetzt werden, arbeiten in 24 und 77 GHz-Bereich. Die Radarsensoren in den 24 GHz-Bereichen sind auf den Nahbereich des Fahrzeugs beschränkt. Sie können eine Entfernung von 25 Metern erfassen und haben eine große Radarkeule², so dass acht Sensoren dieser Art die gesamte Fahrzeugumgebung abdecken können.

Die Radarsensoren im Bereich von ca. 77 GHz können etwa 150 Meter Entfernung abdecken. Sie haben aber im Vergleich zu den oben genannten Radarsensoren eine schmale Radarkeule [DAIM-06].

Einen umfangreichen Überblick über Radarsysteme im Automobilbereich gibt [MIND-99].

2.1.1.4. Infrarot-Distanzsensoren

Die Infrarot-Distanzsensoren erfassen einen einzigen Punkt, der sich senkrecht zum Mittelpunkt zwischen Sender und Empfänger des Sensors befindet. Mit dieser Technik ist die Erfassung der gesamten Umgebung eines Fahrzeugs unmöglich.

² Radarkeule: Die Radarkeule bezeichnet den abgedeckten Winkelbereich des Sensors.

Trotz dieses Nachteils werden die Infrarot-Distanzsensoren in vielen Forschungseinrichtungen im Bereich der autonomen Roboter und Fahrzeugen für Experimentelle Zwecke eingesetzt. Der Grund dafür ist, dass diese Sensoren im Vergleich zur Radarsensoren und Sehsysteme sehr günstig sind. Außerdem sind sie einfach zu bedienen. Sie brauchen wenig Strom, sind nicht rechenintensiv und können an einem einfachen Controller angeschlossen werden. Abhängig von der Entfernung des erfassten Objekts gibt die Auswertelektronik dieser Sensoren die Spannung U_{out} als Analogsignal aus [CONR-05]. Eine Auswertsoftware soll dann aus dem Wert des Ausgangssignals die Entfernung berechnen.

Sehr beliebt sind die Infrarot- Distanzsensoren der Firma Sharp. Sie haben unterschiedliche Reichweiten. Diese Sensoren arbeiten im Nahbereich ziemlich genau aber im Grenzbereich lässt die Zuverlässigkeit jedoch stark nach.

2.1.1.5. Ultraschallsensoren

Die Funktionsweise der Ultraschallsensoren ist den Radarsensoren ähnlich. Da sich die Signale der Ultraschallsensoren mit der Schallgeschwindigkeit ausbreiten, sind sie langsamer als Radarsensoren.

Die Ultraschall-Abstandssensoren arbeiten nach dem Impuls-Echo-Prinzip: Der Sensor sendet in bestimmten Zeitabständen kurze Impulse oder Impuls-Gruppen. Nach der Reflexion am Messobjekt werden die ausgesandten Signale wieder vom Sensor empfangen. Die Laufzeit des Signals wird gemessen. Aus der Laufzeit des Signals und der Schallgeschwindigkeit wird dann die Objektentfernung berechnet.

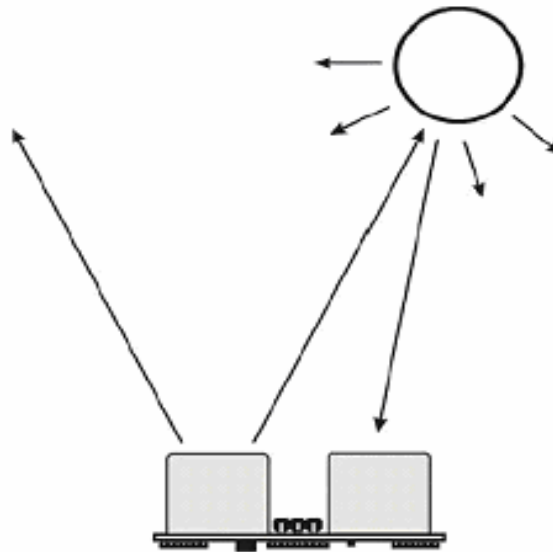


Bild 2-1: Ausbreitung und Reflexion der Ultraschallsignale.

Die Ausbreitungsgeschwindigkeit des Schalls in Luft ist temperaturabhängig und beträgt bei 20°C 343m/s.

Die im Bild 2-2 angezeigte SRF05³ Ultraschallsensor kann eine Reichweite zwischen 3 cm bis 6 m erfassen. Sie kosten so viel wie die Infrarot-Distanzsensoren, somit stellen sie eine gute Alternative zu den Infrarotsensoren dar.

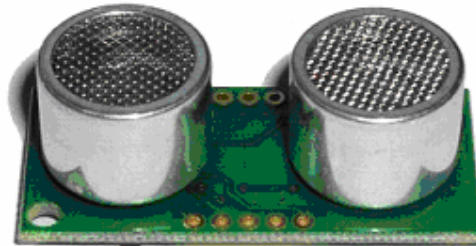


Bild 2-2: Ultraschall-Entfernungssensormodul SRF05[SRF0-05].

2.1.1.6. Robustheit gegen Sensordefekte

Damit die autonomen Fahrzeuge ihren Aufgaben gerecht werden, sind sie auf ihre Sensordaten angewiesen. Mit Hilfe der vorliegenden Sensordaten bestimmen sie ihre Aktionen. Das Ausfall eines Sensors oder mehrere Sensoren führen dazu, dass das Fahrzeug ein falsches Bild von der Umgebung bekommt.

Das könnte dazu führen, dass die Hindernisse nicht erkannt werden und das Fahrzeug mit einem Hindernis kollidiert. Die Folgen sind dann erheblichen Sachschäden oder sogar Verletzung von Personen. Das Steuerungssystem des autonomen Fahrzeugs muss in der Lage sein, die Sensoren des Fahrzeugs auf ihre Funktionalität zu überprüfen. Bei einem Sensordefekt soll das Steuerungssystem des Fahrzeugs abhängig von der Wichtigkeit des defekten Sensors eine situationsgerechte Aktion wie Sperren oder Einschränken der Funktionen des Fahrzeugs durchführen können. Die Überprüfung der Sensoren soll in Echtzeit stattfinden, damit die Fehler der Sensoren rechtzeitig erkannt und dementsprechend schnell reagiert werden kann.

2.1.1.7. Messabweichungen der Sensoren

Die Messungen physikalischer Größen sind immer mit gewissen Messfehlern behaftet. Abhängig von verwendeten Messgeräten, Messverfahren und der physikalischen Messgröße fällt die Messabweichung

$$\Delta x = x - x_w \quad (2.1.1.7.1)$$

x : der gemessene Wert

x_w : der wahre Wert

³ SRF: Ultrasonic Range Finder.

unterschiedlich stark aus. Die autonomen Fahrzeuge nehmen die Umgebung mit Hilfe ihrer Sensoren wahr. Die Messdaten der Sensoren sind mit Fehlern behaftet. Wie im Abschnitt 2.1.1.4 erläutert, sind zum Beispiel die Abweichungen der Messwerte des Infrarot-Distanzsensors im Nahbereich des Sensors klein und nehmen im Grenzbereich des Sensors zu.

Werden diese Messfehler der Sensoren ignoriert, könnte sich das autonome Fahrzeug ein falsches Bild von seiner Umgebung machen und dies könnte bei größeren Messabweichungen zu Kollisionen oder nicht gewünschtem Verhalten des autonomen Fahrzeugs führen. Um das zu vermeiden, müssen die Messdaten der Sensoren auf ihre Genauigkeit untersucht werden. Liegt eine Messabweichung vor, müssen diese, falls möglich, kompensiert werden. Sind die Messabweichungen nicht kompensierbar, dürfen dann nur die Sensorwerte benutzt werden, deren Messabweichungen noch tolerierbar sind.

Die Messabweichungen sind allgemein in zwei Klassen aufgeteilt, nämlich systematische und zufällige Messabweichungen. Bei der systematischen Messabweichung ist jeder Messwert x_i von dem wahren Wert x_w in die gleiche Richtung und um den gleichen Betrag verschoben. Die systematische Messabweichung wird wie folgt berechnet: $\Delta x = x - x_w$ ist die absolute systematische Messabweichung, und $\frac{\Delta x}{x}$ ist die relative systematische Messabweichung. Die systematische Messabweichung ist kompensierbar, das heißt, der Messfehler kann beseitigt werden, wenn man die Messabweichung nach der Gleichung (2.1.1.7.1) für einen bekannten Wert x_w berechnet und dementsprechend die Messwerte x_i korrigiert.

Die zufälligen Messabweichungen können nicht kompensiert werden. Somit ist es nicht möglich, den wahren Wert x_w genau zu berechnen. Man kann nur angeben, dass er in dem Intervall zwischen $x - \Delta x$ und $x + \Delta x$ liegt, anders ausgedrückt: $x_w = x \pm \Delta x$. Bei der zufälligen Messabweichung ist $\pm \Delta x$ die zufällige absolute Messabweichung und $\pm \frac{\Delta x}{x}$ die relative zufällige Messabweichung.

2.1.2. Steuerungsarchitektur autonomer Fahrzeuge

Ein autonomes Fahrzeug muss in der Lage sein, selbständig von einer Position A nach einer Zielposition B zu fahren. Das Fahrzeug soll z.B. selbst den Weg zum Ziel finden und dabei auf der Fahrspur bleiben, Hindernisse erkennen und umfahren und eventuell mit anderen Fahrzeugen kooperieren, um eine eventuelle Kollision zu vermeiden. All diese Aufgaben sollen parallel und in Echtzeit durchgeführt werden, sonst kann das Fahrzeug auf Hindernisse nicht rechtzeitig reagieren. Ein autonomes Fahrzeug soll auch im Falle eines Sensorausfalls reaktionsfähig bleiben, ein Steuerungsmodell muss also all den oben genannten Anforderungen gerecht werden.

Im Folgenden werden zwei Steuerungsmodelle, die oft in der Roboterwelt Verwendung finden, erläutert.

- ◆ Die Funktionsorientierte Architektur, auch hierarchische Architektur genannt, ist die klassische Architektur für autonome mobile Roboter. Bei diesem Modell wird die Steuerung in mehrere Teilfunktionen zerlegt, die dann sequentiell ausgeführt werden[HAUT-05]:

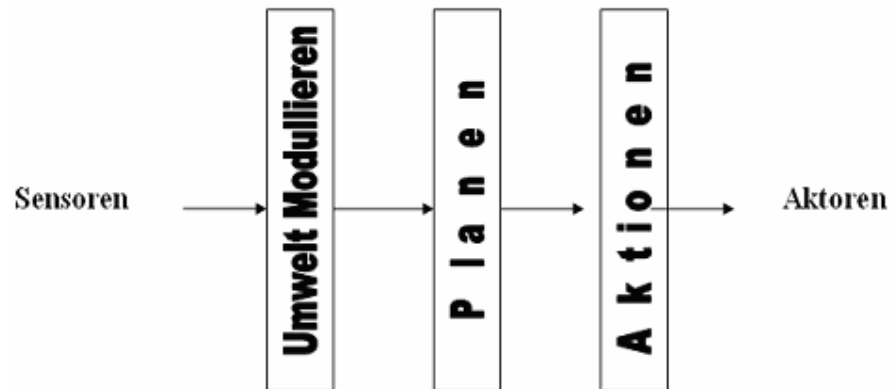


Bild 2-3:Funktionsorientierte Steuerungsarchitektur für autonome Fahrzeuge.

Setzt man die Funktionsorientierte Architektur für die Steuerung autonomer Fahrzeuge ein, wird die Modulierung der Umgebung, der Planung und der Aktionen des autonomen Fahrzeugs wie folgt durchgeführt: Mit Hilfe der Sensordaten des autonomen Fahrzeugs wird ein globales Modell der Umgebung des autonomen Fahrzeugs erstellt. Anschließend werden basierend auf diesem Modell die Aktionen des Steuerungssystems geplant und durchgeführt (vgl. Bild 2-3). Somit sind die Reaktionen solcher Systeme auf eintretende Ereignisse vorhersehbar, da die Informationen sequentiell verarbeitet werden.

Der Nachteil der Funktionsorientierten Architektur ist, dass das autonome Fahrzeug zu spät auf eintretende Ereignisse seiner Umgebung reagiert. Diese Architektur wird den Echtzeitanforderungen, die an ein autonomes Fahrzeug gestellt werden, nicht gerecht, z.B. die Durchführung eines Überholvorgangs unter gleichzeitiger Vermeidung von Kollisionen. Außerdem kann ein Fehler innerhalb einzelner Teilfunktionen das ganze Steuerungssystem des autonomen Fahrzeugs lahm legen.

- ◆ Verhaltensbasierte Steuerungsarchitektur: Die Grundidee dieses Modells ist, die Steuerung des autonomen Fahrzeugs in verschiedenen Verhaltensweisen zu realisieren. Alle Verhaltensweisen laufen parallel und haben sofortigen Zugriff auf die Sensordaten. Jede Verhaltensweise greift auf die Sensordaten zu, die für ihre Ausführung notwendig sind und kann anschließend sofort ihre Aktionen ausführen. Fällt eine Verhaltensweise aus, können die anderen Verhaltensweisen weiterarbeiten. Das Gesamtverhalten des autonomen Fahrzeugs entsteht durch das Zusammenspiel der einzelnen Verhaltensweisen.

Das Steuerungssystem der im Rahmen dieser Arbeit zu entwickelnden autonomen Modell-Fahrzeuge wird nach der Verhaltensbasierten Steuerungsarchitektur, genauer der „Subsumptions-Architektur“ von Brooks realisiert. Im Folgenden wird das Konzept der Subsumptions-Architektur vorgestellt

Subsumptions-Architektur

Nach der Subsumptions-Architektur wird das Gesamtverhalten des autonomen Systems in Verhaltensweisen aufgeteilt [BROO-89]. Diese laufen alle parallel und haben direkten Zugriff auf Sensordaten. Abhängig von den aktuellen Sensordaten und inneren Zuständen des Systems werden bestimmte Verhaltensweisen aktiviert oder deaktiviert.

Den einzelnen Verhaltensweisen werden die so genannten Kompetenzebenen (K_0, \dots, K_n) zugeordnet. In jeder Kompetenzebene wird eine Klasse von Verhaltensmustern definiert. Die höheren Kompetenzebenen können, falls sie aktiv sind, die Aktionen der unteren Kompetenzebenen unterdrücken (vgl. Bild 2.4).

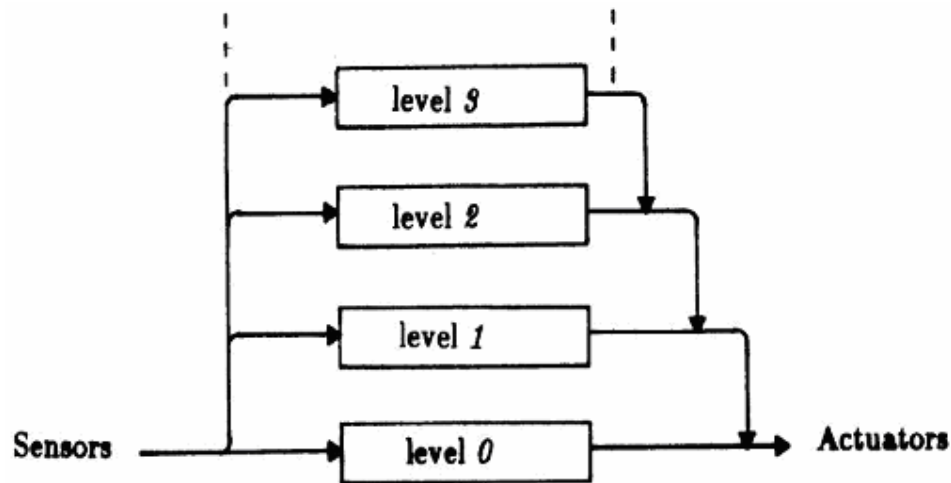


Bild 2-4: Parallel laufende Kompetenzebenen[BROO-89].

Je höher die Kompetenzebenen, desto spezifischer ist ihr Verhalten. Mit zunehmender Anzahl der Kompetenzebenen wird das Verhalten des Systems komplexer.

Bei der Implementierung muss folgende Regel beachtet werden: Zuerst wird die Kompetenzebene K_0 implementiert. Wenn diese Ebene funktioniert, wird sie nicht mehr geändert. Als nächstes wird die Kompetenzebene K_1 implementiert. Die Kompetenzebene K_1 kann die Daten (Ausgaben) von K_0 benutzen und Daten an den Schnittstellen von K_0 eingeben. Die Aktionen bzw. Ausgaben von K_0 können durch K_1 unterdrückt werden. Analog werden an Kompetenzebenen (K_2, \dots, K_n) implementiert.

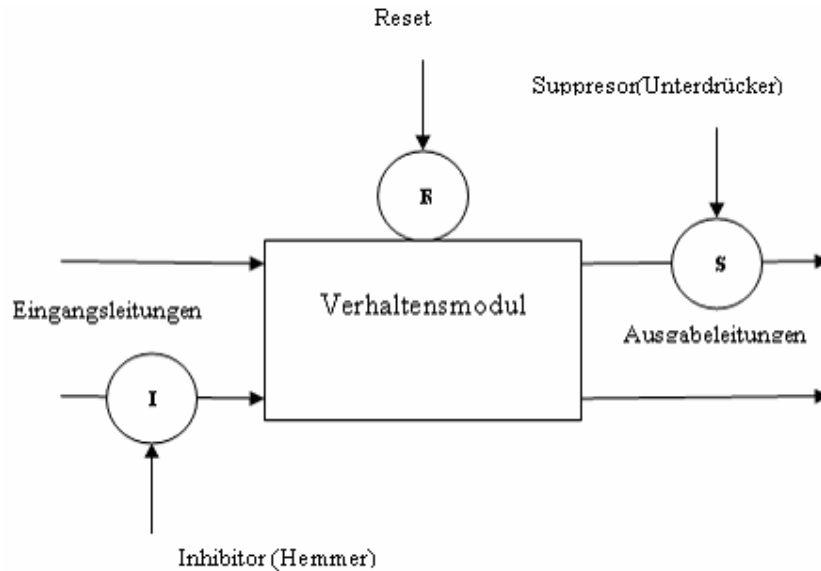


Bild 2-5: Hemmung und Unterdrückung der Ein- und Ausgangssignale.

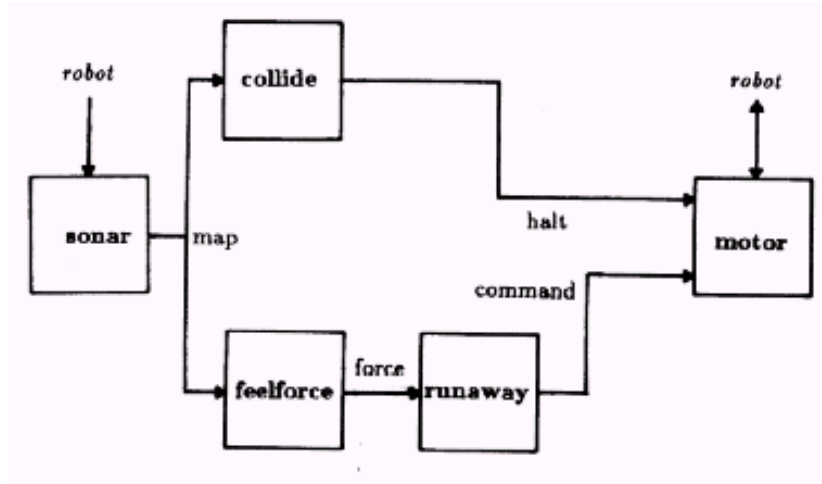
Wie das Bild 2-5 verdeutlicht, können die Eingangssignale eines Moduls durch die Module höherer Ebenen ersetzt, die Ausgangssignale unterdrückt oder das Modul neu gestartet werden. Die Hemmung und die Unterdrückung werden grafisch durch Kreise an die jeweilige Eingangs- und Ausgangsleitungen des Moduls dargestellt. Das Reset wird wie im Bild 2-5 durchgeführt.

Im Folgenden werden einige nach Brooks beispielhafte Verhaltensweisen aufgelistet.

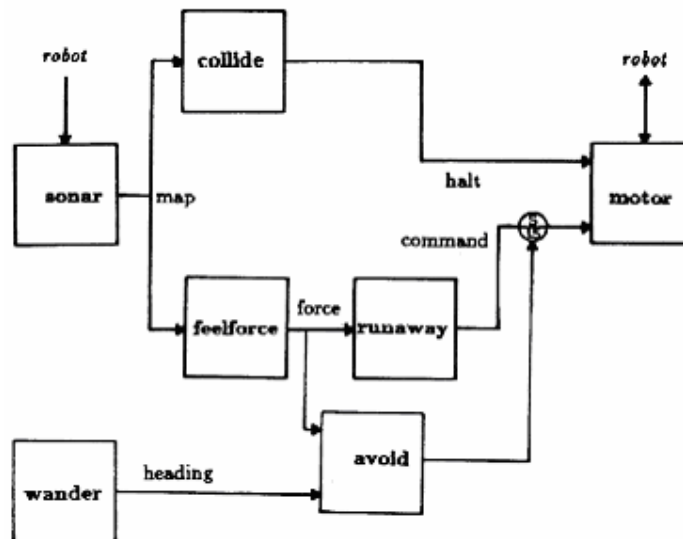
- ◆ Vermeidung von Kollisionen mit sich bewegenden oder stationären Objekten.
- ◆ Wandern und dabei Hindernisse umfahren.
- ◆ Erkundung entfernter Ziele, die erreichbar sind und diese ansteuern.

Als Beispiel werden im Folgenden die von Brooks entworfenen Kompetenzebenen für diese Verhaltensweisen erläutert.

Die Kompetenzebene K_0 : Diese Kompetenzebene ist für die Kollisionsvermeidung zuständig. Das Modul "sonar" (vgl. Bild 2-6) macht aus den Sensordaten ein Bild von der Umgebung des Systems. Das Modul "feelforce" nimmt die Hindernisse wahr und ermittelt abhängig davon eine abstoßende Kraft und summiert diese auf. Das Modul "runaway" sorgt dafür, dass der Roboter vor den Hindernissen ausweicht und von ihnen weg bewegt. Das Modul "collide" wird dann aktiv, wenn sich ein Hindernis in unmittelbarer Nähe des Roboters befindet und stoppt den Roboter.

Bild 2-6: Automatenmodell der Kompetenzebene K₀ [BROO-89].

Die Kompetenzebene K₁: Bei K₁ wird ein zielloses Fahren realisiert. Das Modul "wanderer" generiert zufällig Befehle für die Fahrrichtungen des Roboters. Das Modul "avoid" berücksichtigt auch das Ergebnis des Moduls "feelforce" aus K₀. Die Ausgabe des Moduls "runaway" von K₀ wird durch "avoid" unterdrückt.

Bild 2-7: Automatenmodell der Kompetenzebenen K₀ und K₁ [BROO-89].

Die wichtigsten Vorteile, die die Subsumptions-Architektur mit sich bringt, sind

- ◆ Schnelle und immer vorhandene Reaktionsfähigkeit.
- ◆ Weitere Kompetenzebenen können einfach in das System hinzugefügt werden und somit das Verhalten des Systems erweitern.

Ein wesentlicher Nachteil dieses Modells ist, dass das System nur auf die aktuell vorliegenden Sensordaten reagiert und sich kein Bild aus seiner Umwelt macht. Soll ein auto-

nomes Fahrzeug komplexere Aufgaben durchführen, z.B. den Weg zum Ziel selbst planen, wäre für diese Aufgabe eine funktionsorientierte Architektur eher geeignet. Für solche autonomen Fahrzeuge wäre eine Kombination aus beiden Architekturen die beste Lösung. Diese Kombination wird auch Hybrid-Architektur genannt. Bei diesem Architektur-Modell werden die reaktiven Ebenen unten angesiedelt. Sie sollen z.B. eine schnelle Reaktion auf Hindernisse ermöglichen und das Fahrzeug auf der Fahrbahn halten. Bei diesem Modell ist die hierarchische Architektur in den höheren Ebenen angesiedelt. Sie plant, kontrolliert und koordiniert die Vorgänge im Steuerungssystem.

2.2. Regelungstechnik

In allen Systemen, bei denen es auf die Automatisierung der Prozesse im System ankommt, müssen die Prozesse permanent beobachtet werden, um im Falle einer Abweichung vom geforderten Verhalten so zu reagieren, dass die geforderte Betriebsart wiederhergestellt wird.

Für solche Art der Automatisierung wird die Regelungstechnik eingesetzt. In Rahmen dieser Arbeit werden die Methoden aus der Regelungstechnik benutzt, um die Fahrtrichtung der Modell-Fahrzeuge so zu regeln, dass das Fahren parallel zur Wand ermöglicht wird. Im diesem Kapitel wird auf den Aufgabenbereich der Regelungstechnik eingegangen, dann die Entstehung eines „Regelkreises“ beschrieben, und die am häufigsten eingesetzten Regeltypen vorgestellt.

„Die Regelungstechnik beschäftigt sich mit der Analyse von dynamischen Systemen, die meist technischer Art sind. Ein dynamisches System besitzt einen oder mehrere Eingänge, über die auf das System in kontrollierbarer Weise eingewirkt werden kann[SCHU-05]“.

Die Eingangsgrößen, die direkt auf das Verhalten des Systems einwirken, nennt man Stellgrößen. Die Wirkung des Eingriffs auf die Stellgrößen findet nicht unmittelbar statt, sondern in einer zeitlichen Verzögerung, sodass eine gewisse Zeitspanne zwischen dem Eingriff und seinen beobachtbaren Auswirkung liegt.

Die Regelungstechnik findet man sowohl in natürlichen als auch in technischen Systemen. In den meisten höheren Lebewesen gibt es eine Vielzahl von Regelkreisen. Einige Beispiele sind im Folgenden aufgeführt[SCHU-05]:

- ◆ Pupille im Auge, Lichteinfall auf die Netzhaut
- ◆ Blutzucker, Blutdruck, CO₂-Gehalt im Blut
- ◆ Körpertemperatur

Im Bereich der Technik wird die Regelungstechnik z. B. für die Regelung einer Klimaanlage eingesetzt. Die Aufgabe der Klimaanlage ist, eine gewünschte Raumtemperatur konstant zu halten. Die Regeleinrichtung der Klimaanlage erhöht die Wärmezufuhr, wenn die Außentemperatur fällt und dadurch der Raum an Wärme verliert, oder sie kühlt den Raum, falls die Außentemperatur sich so erhöht, dass die Raumtemperatur einen bestimmten Wert überschreitet. Damit wird erreicht, dass die Raumtemperatur unabhängig von der Außentemperatur immer gleich bleibt.

2.2.1. Regelkreis

In der Regelungstechnik wird zwischen Regelung und Steuerung unterschieden. Das Steuern eines Systems besteht darin, das Eingangs- und Ausgangsverhalten des Systems zu analysieren und dem System eine Steuereinheit vorzuschalten, die dafür sorgt, dass die gewünschten Werte der Ausgangssignale erzielt werden. In der Regelungstechnik wird das als eine Steuerung definiert [BOET-02].

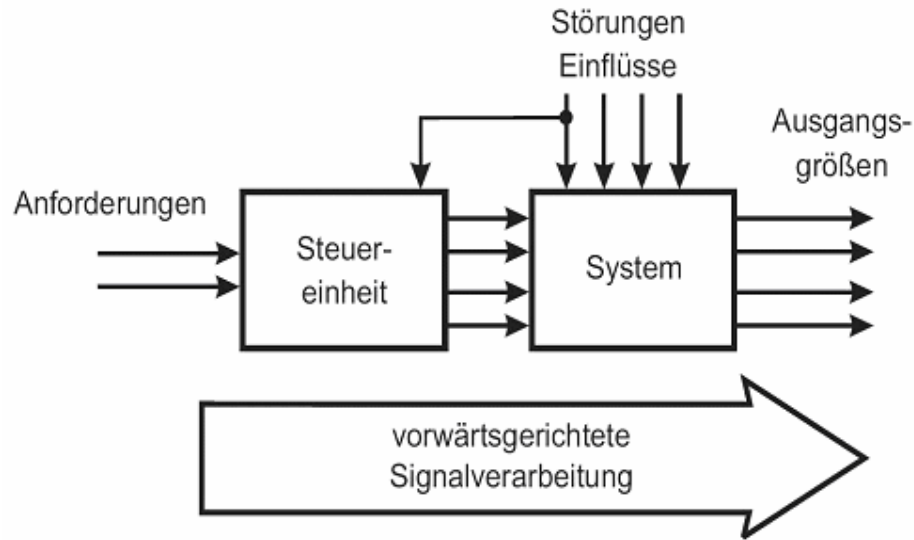


Bild 2-8: Blockschaltbild einer Steuerung[SCHU-05].

Eine Regelung erfasst im Unterschied zur Steuerung auch das Ausgangsverhalten des zu regelnden Systems für die Rückkopplung. Die Ausgangsinformation fließt somit wieder in das System ein.

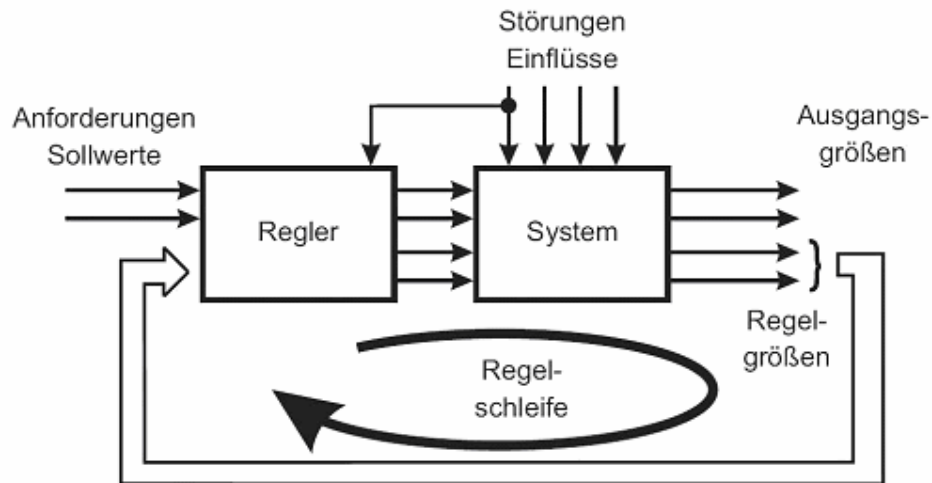


Bild 2-9: Blockschaltbild einer Regelung[SCHU-05].

So entsteht eine Regelschleife (vgl. Bild 2-9), die in der Regelungstechnik als Regelkreis bezeichnet wird.

Ein Regelkreis besteht aus einer Vergleichsstelle, einem Regler und einer Regelstrecke (vgl. Bild 2-10). Die Regelstrecke ist der Teil des zu regelnden Systems, der beeinflusst werden soll.

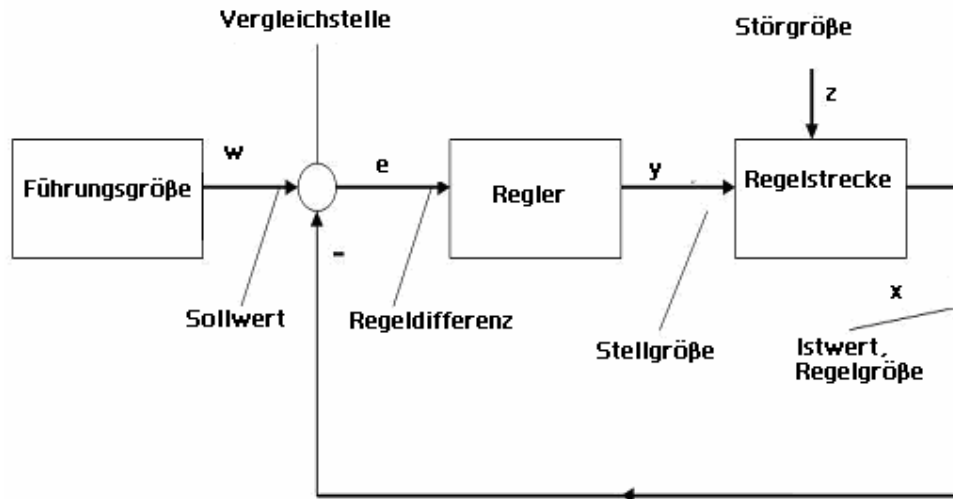


Bild 2-10: Blockschaltbild eines Regelkreises.

- ◆ **Führungsgröße:** Die Führungsgröße w (vgl. Bild 2-10) eines Regelkreises ist eine von der betreffenden Regelung nicht beeinflusste Größe, auch Sollwert genannt, die dem Regelkreis von außen zugeführt wird. Bei der oben beschriebene Klimaanlage ist der Sollwert die gewünschte Raumtemperatur.
- ◆ **Vergleichsstelle:** Die Vergleichsstelle ist eine Funktionseinheit, die die Regeldifferenz (auch Regelabweichung genannt) $e = (w - x)$ zu Verfügung stellt, wobei mit x die Regelgröße gemeint ist.
- ◆ **Regler:** Der Regler bildet aus der Einganggröße (Regeldifferenz e) die Ausganggröße (Stellgröße y) so, dass sich die Regelgröße x so schnell und genau wie möglich den Sollwert anpasst.
- ◆ **Regelstrecke:** Die Regelstrecke beinhaltet das Stellglied, d.h. die Stelle, an der die Regelung durchgeführt wird, bzw. an dem die Stellgröße einwirkt, und den Messort, an dem die Regelgröße x gemessen wird. Die Einganggröße der Regelstrecke ist die Stellgröße y und die Ausganggröße der Regelstrecke ist die Regelgröße x , auch Istwert genannt. Auf die Regelstrecke wirkt auch die Störgröße z . Das sind alle Größen, die die Regelung von außen beeinflussen (außer der Sollwert w). Im Beispiel der Klimaanlage ist die Störgröße z die Außentemperaturschwankung.

2.2.2. Regeltypen

Die Wahl eines bestimmten Reglertyps richtet sich nach dem geforderten Zeitverhalten und der geforderten Regelgenauigkeit der Regelstrecke. Die wichtigsten klassischen Reglertypen werden hier kurz zusammengefasst[BOET-02]:

- ◆ P-Regler
- ◆ I-Regler
- ◆ D-Regler

Bei dem P-Regler ist die Stellgröße y proportional zu der Regeldifferenz $e = (w - x)$. Die klassische P-Regler-Gleichung stellt den Zusammenhang zwischen der Regelabweichung e und der Stellgröße y dar.

$$y(t) = K_p \cdot e(t) \quad (2.2.2.1)$$

K_p ist der Proportionalitätsfaktor des P-Reglers. Er ist der einzige Einstellparameter des P-Reglers und bestimmt die Regelstärke des P-Reglers. Wird K_p ein kleiner Wert zugewiesen, erzeugt der P-Regler ein schwaches Ausgangssignal (Stellgröße y) und demzufolge wird auf die Regelabweichung sehr langsam reagiert. Bei einem großen K_p greift der Regler schon bei kleinerer Regelabweichung zu stark auf das zu regelnde System ein. Der Nachteil des P-Reglers ist, dass er eine bleibende Regelabweichung aufweist. Das heißt, er ist nicht in der Lage, den Istwert genau auf den Sollwert einzustellen.

Die bleibende Regelabweichung entsteht aus folgenden Gründen: wenn die Regelabweichung $e = (w - x) = 0$ wird, ergibt sich die Stellgröße $y = 0$ (siehe Gleichung (2.2.2.1)). Das bedeutet, dass der Regler nicht mehr auf das zu regelnde System eingreift, das System wird nicht mehr geregelt. Im Beispiel der Klimaanlage wird die Regeleinrichtung der Anlage den Temperatur nicht mehr regeln, wenn die gewünschte Raumtemperatur erreicht ist ($w = x$), weil der P-Regler kein Stellsignal mehr liefert. Demzufolge ändert sich die Raumtemperatur abhängig von der Außentemperatur, und es entsteht wieder eine Regelabweichung. Am Anfang ist aber die Reaktion der Anlage auf die entstehende Regelabweichung so schwach, dass es nicht ausreicht, die Regelabweichung wieder abzubauen. Erst bei größeren Regelabweichungen reagiert die Regeleinrichtung der Anlage stark genug, sodass die gewünschte Raumtemperatur wieder hergestellt wird. Wie dieses Beispiel verdeutlicht, hat der P-Regler immer eine bleibende Regelabweichung.

Um die bleibende Regelabweichung des P-Reglers möglichst klein zu halten, muss der Proportionalitätsfaktor K_p so groß wie möglich gewählt werden, damit der Regler schon bei kleineren Regelabweichungen stark genug auf diese Regelabweichungen reagiert, in dem er eine große Stellgröße y erzeugt. (vgl. Gleichung (2.2.2.1)). Theoretisch könnte der K_p Wert sehr groß gewählt werden, damit die bleibende Regelabweichung gegen null tendiert. Ein zu große K_p würde aber in der Praxis das zu regelnde System in Schwankung bringen. Im Beispiel Klimaanlage würde die Anlage die Wärmezufuhr zu stark erhöhen, falls die Raumtemperatur niedriger ist als die gewünschte Temperatur, sodass die Raumtemperatur den gewünschten Wert dann überschreiten würde. Es entsteht eine negative Regelabweichung ($x > w$). In diesem Fall würde dann die Anlage den Raum stark kühlen, so dass die Raumtemperatur wieder unter den gewünschten Wert sinken würde. Somit gerät das System in Schwankung.

Da K_p auf Grund des erläuterten Sachverhalts nicht beliebig groß gewählt werden kann, kann auch die bleibende Regelabweichung nicht komplett beseitigt werden. Somit weist der P-Regler immer eine bleibende Regelabweichung auf. Ob die bleibende Regelabweichung des P-Reglers vernachlässigbar klein ist oder nicht, hängt von dem Einsatzgebiet des P-Reglers und geforderten Genauigkeit an das Regelsystem ab.

Bei dem I-Regler werden die Regelabweichungen über die Zeit aufsummiert und die Summe mit dem Faktor K_i multipliziert (siehe Gleichung (2.2.2.2)).

$$y = (t)K_i \cdot \int e(t)dt \quad (2.2.2.2)$$

Die Dauer der Regelabweichung geht somit auch in die Berechnung der Stellgröße ein. Der I-Regler hat keine bleibende Regelabweichung, ist aber langsamer als der P-Regler. Das Bild 2-11 zeigt die Sprungantwort⁴ des P- und I-Reglers abhängig von der sprunghaften Änderung der Regeldifferenz.

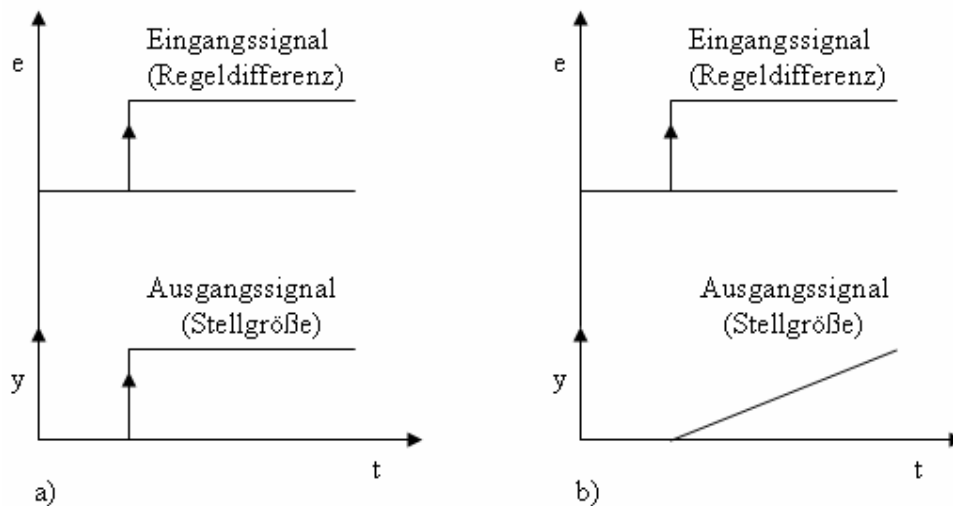


Bild 2-11: a) Sprungantwort des P-Reglers, b) Sprungantwort des I-Reglers.

Wie das Bild 2-11 zeigt, reagiert der P-Regler schnell auf die sprunghafte Änderung der Regeldifferenz, indem er ein entsprechend starkes Ausgangssignal erzeugt. Bei dem I-Regler ergibt sich am Ausgang eine stetige Signaländerung. Das Ausgangssignal des I-Reglers ist am Anfang schwach, es wird aber umso stärker, je länger die Regelabweichung vorhanden ist. Selbst eine kleine Regelabweichung kann somit ein kräftiges Ausgangssignal hervorrufen.

Durch Kombination des P- und I-Reglers können die Vorteile beider Regeltypen miteinander verbunden werden. Der entstandene Regler wird als PI-Regler bezeichnet. Bei einem Signalsprung am Eingang des PI-Reglers folgt am Ausgang unmittelbar der P-Anteil (vgl. Bild 2.12), während der I-Anteil linear mit der Zeit wächst. Die Zeit, die

⁴ Sprungantwort: Verlauf der Ausgangsgröße (Stellgröße) eines Reglers abhängig von der sprunghaftigen Änderung des Eingangssignals (Regeldifferenz).

vom Signalsprung an vergeht, bis der I-Anteil die Größe des P-Anteils erreicht hat, wird die Nachstellzeit T_n genannt.

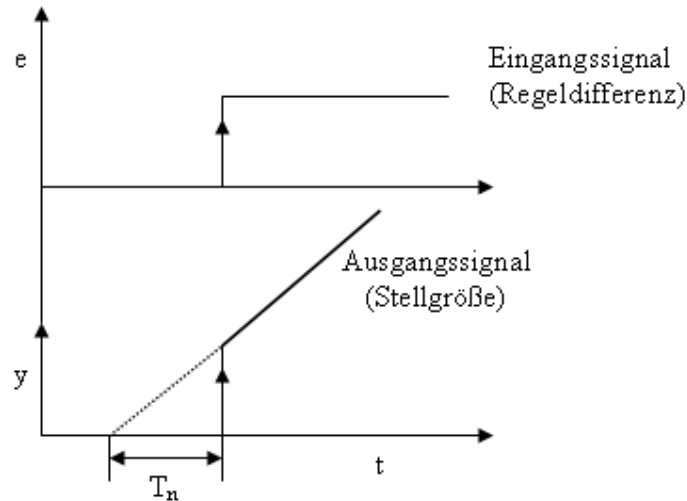


Bild 2-12: Sprungantwort des PI-Reglers auf plötzliche Änderung der Regeldifferenz.

Der PI-Regler besteht aus der Addition der P-Regler und I-Regler.

$$y(t) = K_p \cdot e(t) + K_I \cdot \int_0^t e(t) dt \quad (2.2.2.3)$$

Diese Gleichung ist äquivalent zu

$$y(t) = K_p \cdot \left(e(t) + \frac{1}{T_n} \int_0^t e(t) dt \right) \quad (2.2.2.4)$$

Zwischen K_p und K_I besteht folgender Zusammenhang:

$$K_I = \frac{K_p}{T_n}$$

Der D-Regler sorgt für eine schnelle Ausregelung des Systems bei einer schnellen Änderung der Regelabweichung.

$$y(t) = K_D \cdot \frac{e(t)}{dt} \quad (2.2.2.5)$$

Falls das zu regelnde System zu große Störungen ausgesetzt ist, kann dem Regler ein D-Anteil hinzugefügt werden, um auf Störung schnell zu reagieren.

2.3. Ausgleichproblem

In der Technik tritt häufig das Problem auf, für eine Reihe von Messpunkten einen funktionalen Zusammenhang $y = g(x)$ so zuzuordnen, dass es möglich wird, auch für beliebige x -Werte, für die keine Messwerte vorliegen, einen Funktionswert $g(x)$ zu errechnen. Für die Bestimmung der Funktion $g(x)$ aus den Messpunkten, auch Stützstellen genannt, gibt es verschiedene mathematische Ansätze. Interpolationspolynome⁵ sind die oft verwendeten Hilfsmittel zur Bestimmung der Funktion $g(x)$.

Im Bereich der autonomen Fahrzeuge stellen die Sensoren Informationen über die Umgebung des autonomen Fahrzeugs in Form von Rohdaten bereit. Diese Informationen können nicht ohne weiteres benutzt werden. Die Sensordaten müssen zuerst in eine Weltmodellierung überführt werden. Es soll eine mathematische Funktion mit Hilfe der experimentell ermittelten Messpunkte $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ bestimmt werden, die aus den Rohdaten der Sensoren die physikalischen Größen, wie z.B. den Abstand des Fahrzeugs relativ zu einem Objekt, berechnet. Die durch Experimente ermittelten Messpunkte sind aber oft mehr oder weniger mit Fehlern behaftet. Der Nachteil des Interpolationsverfahrens für solche Fälle ist, dass die durch die Interpolation gefundene Funktion $g(x)$ die Stützstellen exakt beschreibt und den Messfehler als solchen nicht behandelt.

Die lineare Ausgleichrechnung „kleinste Fehlerquadratmethode“ [HERM-01] minimiert die Fehler, die durch die Messung entstehen. Mit Hilfe dieser Methode ist es möglich, eine Näherungsfunktion zur Überführung der Sensordaten in für das autonome Fahrzeug brauchbare Größen mit minimalem Fehler zu bestimmen.

Approximation durch kleinste Fehlerquadratmethode

Die Näherungsfunktion wird mit Hilfe der Ausgleichrechnung wie folgt berechnet:

Gegeben sei eine Menge von Messpunkten $\{(x_i, y_i), i = 0, \dots, n\}$, mit $x_i \neq x_j$ für $i \neq j$ sowie die Ansatzfunktionen f_0, \dots, f_m . Zu bestimmen ist eine stetige Funktion $f = \sum_{i=0}^m \lambda_i \cdot f_i$, die den funktionalen Zusammenhang $f(x_i) = y_i$ hinreichend genau approximiert, wobei $\lambda = [\lambda_0, \dots, \lambda_m]$ die Parameter der Funktion f sind. Die Methode der kleinsten Fehlerquadrate bestimmt das Minimum der Fehlerfunktion

$$E(\lambda_0, \dots, \lambda_m) = \sum_{i=0}^n (y_i - f(x_i))^2 = \sum_{i=0}^n \left(y_i - \sum_{j=0}^m \lambda_j \cdot f_j(x_i) \right)^2 \quad (2.3.1)$$

bezüglich der Parameter $(\lambda_0, \dots, \lambda_m)$.

$$\text{Definiert man } A = \begin{pmatrix} f_0(x_0) & \dots & f_m(x_0) \\ \dots & \dots & \dots \\ f_0(x_n) & \dots & f_m(x_n) \end{pmatrix}, \lambda = \begin{pmatrix} \lambda_0 \\ \dots \\ \lambda_m \end{pmatrix}, y = \begin{pmatrix} y_0 \\ \dots \\ y_n \end{pmatrix} \quad (2.3.2)$$

Ergibt sich so die Gleichung

⁵ Siehe das Buch von M. Hermann: Numerische Mathematik, München: Oldenburg, 2001, Seite 299.

$$A \cdot \lambda = y$$

Diese Gleichung ist für $m < n$ nicht lösbar.

Das (globale) Minimum der Funktion E lässt sich dann über die folgenden notwendigen und hinreichenden Bedingungen bestimmen (partiellen Ableitungen)

$$\frac{\partial}{\partial \lambda_0} E(\lambda_0, \dots, \lambda_m) = 0, \dots, \frac{\partial}{\partial \lambda_m} E(\lambda_0, \dots, \lambda_m) = 0 \quad (2.3.3)$$

Die Gleichung (2.3.3) ist gleichbedeutend mit

$$A^T \cdot A \cdot \lambda = A^T \cdot y \quad (2.3.4)$$

wobei und A^T die transponierte Matrix von A ist. Die Lösung der Gleichung (2.3.4) liefert dann die gesuchten Parameter $\lambda = [\lambda_0, \dots, \lambda_m]$ der Funktion f .

3. Hardware

3.1. Die Lastwagen

Für die Durchführung der Arbeit stehen zwei Modell-Lastwagen der Firma Wedico² zur Verfügung. Sie sind im Maßstab 1:16 gebaut. Beide sind technisch baugleich und bestehen aus einer 3-achsigen Sattelzugmaschine und einem 2-achsigen Auflieger.



Bild 3-1: Modell-Lastwagen.

Aus der Bauanleitung [WEDI-01] kann die Verdrahtung der elektrischen Fahrzeugkomponenten entnommen werden. Der elektrische Antriebsmotor wirkt über ein 3-Gang-Schaltgetriebe auf die beiden Hinterachsen der Zugmaschine.

Die Zugmaschine ist außer mit der zum Fahren erforderlichen Elektrik auch mit Beleuchtung und einem Geräuschgenerator ausgestattet. Diese Funktionen werden von der Fernsteuerung bedient.

Der Blinker für die linke und rechte Seite wird in Abhängigkeit vom Lenkeinschlag der Vorderräder automatisch geschaltet. Auch die Bremslichter und der Rückfahrcheinwerfer werden automatisch gesteuert. Der Auflieger ist mit Glühlampen für Blinker, Rücklicht und Bremslicht ausgestattet.

Diese werden durch ein mehradriges Kabel von der Zugmaschine mit Strom versorgt. Der Fahrmotor, das Schaltgetriebe und die Lenkung können direkt mit dem Servoport eines Mikrocontrollers verbunden werden.

Der Fahrmotor ist mit einem Fahrregler verbunden. Der Fahrregler versorgt den Fahrmotor mit elektrischer Energie. Dabei wird die angelegte Spannung an Fahrmotor nicht kontinuierlich sondern stufenweise durch den Fahrregler verändert. Diese Abstufungen werden als Fahrstufen⁶ bezeichnet.

Jeder Fahrstufe wird eine Nummer $n \in [-40, 45]$ zugeordnet. Positive Nummern gehören zu Fahrstufen, bei denen das Fahrzeug vorwärts fährt. Zum Rückwärtsfahren werden Fahrstufen mit negativer Nummer verwendet. Mit der Fahrstufe $n = 0$ wird der Motor ausgeschaltet. Mit steigendem Betrag n vergrößert sich auch die Spannung an den Fahrregler. Folgende Gleichung zeigt die Zuordnung der Fahrstufen in Abhängigkeit von der Impulsbreite des Servosignals b [HÜBN-04].

$$\text{Fahrstufe } n = \left. \begin{array}{ll} 45 & \text{wenn } 2000\mu\text{s} \leq b \leq 1974\mu\text{s} \\ 0 & \text{wenn } b = 1500\mu\text{s} \\ -40 & \text{wenn } 1020\mu\text{s} \leq b \leq 1000\mu\text{s} \end{array} \right\} \quad (3.1.1)$$

Für die Impulsbreite $1526 < b < 1974$ gilt:

$$\text{Fahrstufe } n = \frac{1}{10}(b - 1526) \quad (3.1.2)$$

Und für die Impulsbreite $1020 < b < 1435$ gilt:

$$\text{Fahrstufe } n = \frac{1}{10}(b - 1435) \quad (3.1.3)$$

Das Ergebnis der Gleichungen (3.1.2) und (3.1.3) muss dann auf Ganzzahlen gerundet werden. Es ist nicht bekannt, welche Geschwindigkeiten den einzelnen Fahrstufen entsprechen. Um die Geschwindigkeit des Modell-Lastwagens für einzelne Fahrstufen zu ermitteln, wurde in einem Versuch der Modell-Lastwagen mit Fahrstufe n_i die Strecke S mit dem Gang 2 gefahren und dabei die benötigte Zeit t_i ermittelt. Anschließend wurde die Geschwindigkeit $v_i = \frac{S}{t_i}$ berechnet. Das Bild 3-2 zeigt die Kennlinie der Geschwindigkeit, abhängig von der Fahrstufe.

⁶ Die Fahrstufen wurden in Rahmen einer Studienarbeit von B. Hübener experimentell ermittelt [HÜBN-04].

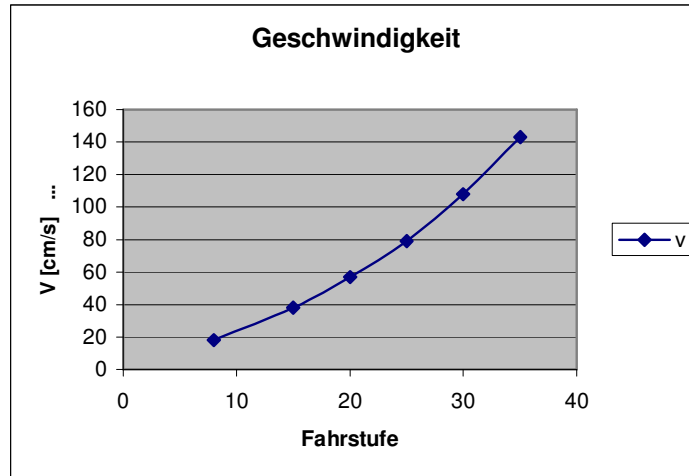


Bild 3-2: Kennlinie der Geschwindigkeit in Abhängigkeit von der Fahrstufe.

Um die Geschwindigkeit für beliebige Fahrstufen zu berechnen, soll eine Funktion $f(n)$ gesucht werden, die die Kennlinie (vgl. Bild 3-2) am besten beschreibt. Der Verlauf der Kennlinie legt nah, einen quadratischen Zusammenhang zu wählen:

$$f(n) = a \cdot n^2 + b \cdot n + c \quad (3.1.4)$$

Der Wert des dritten Parameter ist $c = 0$, da bei der Fahrstufe $n = 0$ der Fahrmotor aus ist. Die Parameter a und b wurde mit Hilfe der kleinsten Fehlerquadratmethode (siehe Abschnitt 2.3) wie folgt berechnet: zuerst wurden die gemessenen Stützstellen in die Gleichung (3.1.4) eingesetzt.

$$f(n_0) = a \cdot n_0^2 + b \cdot n_0 = y_0$$

...

$$f(n_m) = a \cdot n_m^2 + b \cdot n_m = y_m$$

Die Gleichungen wurden in Matrizen-Schreibweise aufgestellt.

$$\begin{pmatrix} n_0^2 & n_0 \\ \dots & \dots \\ n_m^2 & n_m \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} y_0 \\ \dots \\ y_m \end{pmatrix} \quad (3.1.5)$$

$$\begin{array}{ccc} \downarrow & \downarrow & \downarrow \\ A & c & y \end{array}$$

Es muss gelten:

$$A \cdot c = y \quad (3.1.6)$$

Da die Gleichung (3.1.6) eine unsymmetrische Koeffizientenmatrix besitzt, ist diese Gleichung nicht lösbar. Diese Gleichung wurde mit Hilfe der Normalgleichung gelöst.

$$B \cdot c = Y \quad (3.1.7)$$

Wobei $B = A^T \cdot A$ und $Y = A^T \cdot y$ ist. Die Lösung der Gleichung liefert das Ergebnis: $a = 0,08$ und $b = 1,3$. Somit lautet die gesuchte Gleichung

$$f(n) = 0,08 \cdot n^2 + 1,13 \cdot n . \quad (3.1.8)$$

Abhängig von der Impulsbreite b des Servoports berechnet sich der Lenkeinschlag des Lenkgetriebes α wie folgt:

$$f(b) = \begin{cases} \alpha_{\text{voll_links}} & \text{wenn } b \geq 1845\mu\text{s} \\ 0 & \text{wenn } b = 1500\mu\text{s} \\ \alpha_{\text{voll_rechts}} & \text{wenn } b \leq 1155\mu\text{s} \end{cases}$$

Der Lenkeinschlag des Lenkgetriebes α kann für $1155\mu\text{s} < b < 1845\mu\text{s}$ stufenlos zwischen $\alpha_{\text{voll_links}}$ und $\alpha_{\text{voll_rechts}}$ variiert werden.

3.2. Das Aksen-Board

Das Aksen-Board ist für die Robotik und Embedded-Systeme von der Fachhochschule Brandenburg entwickelt worden. Das Herz des Aksen-Board ist ein Mikrocontroller der 8051-Familie.

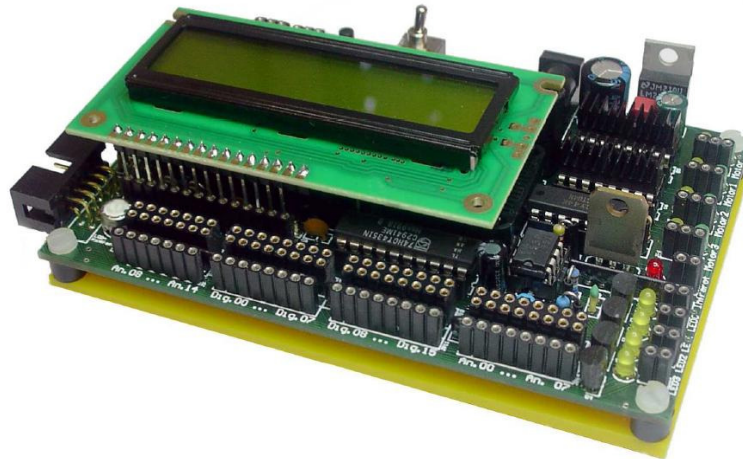


Bild 3-3:Das Aksen-Board.

Es besteht aus folgenden Komponenten

- ◆ Mikroprozessor SAB 80C515A.
- ◆ 64 KB Flash, 8 KB Flash.
- ◆ 4 Motortreiber (in Drehzahl und Richtung variierbar).
- ◆ 3 Servo-Ausgänge, Sie können durch Software auf maximal 8 Servo-Ausgänge erweitert werden.
- ◆ 15 analoge Eingänge. Unterschiedliche Arten von Sensoren, z. B. Sensoren für Abstand, Infrarot, Licht können angeschlossen werden. Die analogen Eingänge sind mit einem Pull-up-Widerstand verbunden.
- ◆ 16 digitale Ports. Sie können als Ein- bzw. Ausgang konfiguriert werden.
- ◆ 4 schaltbare Leistungstreiber, z.B. Infrarotsender, Lämpchen und LED.
- ◆ 1 Infrarotausgang mit Leistungstreiber
- ◆ 3 Encoder-Eingänge zum Erfassen von Drehzahlen.
- ◆ DIP-Schalter (vierfach). Der DIP-Schalter eignet sich gut zur Auswahl von verschiedenen Programmteilen oder Betriebsarten.
- ◆ Zweizeiliges LCD , je 16 Zeichen.
- ◆ CAN-Interface 1 Mbit (optional).
- ◆ Bluetooth-Verbindung zum PC (optional).

Das Aksen-Board ist mit einer Seriellen Schnittstelle ausgestattet. Diese Schnittstelle ermöglicht den Datenaustausch zwischen dem Aksen-Board und ein Computer bzw. ein Peripheriegerät. Für die Kommunikation zwischen Aksen-Board und anderer Hardware besitzt das Aksen-Board ein CAN-Bus. Die Kommunikation ist paketorientiert. Es kann mit Geschwindigkeiten von bis zu einem Megabit gesendet werden.

Programme für das Aksen-Board werden in der Sprache C geschrieben. Um das Aksen-Board zu programmieren, werden zwei Programme benötigt. Das ist zum einen der Compiler, der den Quelltext in für das Board verständlichen Maschinencode übersetzt und zum anderen der Aksen-Flasher, der diesen Maschinencode auf das Board überträgt.

Die Fachhochschule Brandenburg stellt alle Software und Informationen zur Verfügung, die zu Programmierung des Aksen-Boards benötigt werden [FHBR -05].

3.3. Die Distanzsensoren

Zu Verfügung stehen die Distanzsensoren GP2D12⁷ und GP2D15 der Firma Sharp. Für die Messung der Entfernung hat Sharp eine ganz Serie von Bausteinen herausgebracht. Sie haben unterschiedliche Reichweiten. Der in Rahmen dieser Arbeit eingesetzte Sharp-Sensor der Type GP2D12 arbeitet im Intervall [10cm, 80cm]. Dieser Sensor wird für die Erfassung der Wand eingesetzt. Der Sharp-Sensor GP2D15 erfasst Objekte im Intervall [30cm, 150cm], und wird für die Erkennung der Hindernisse auf der Fahrbahn eingesetzt.

Diese Sensoren arbeiten nach dem Triangulationsprinzip⁸. Er besteht aus einem IR-Sender und einer LED-Zeile als Empfänger. Der IR-Sender schickt einen modulierten IR-Strahl gerade aus dem Sensor. Wird dieses Licht reflektiert fällt es in einem bestimmten Winkel durch eine Linse auf die LED-Zeile. Das Bild 2-4 verdeutlicht das Prinzip.

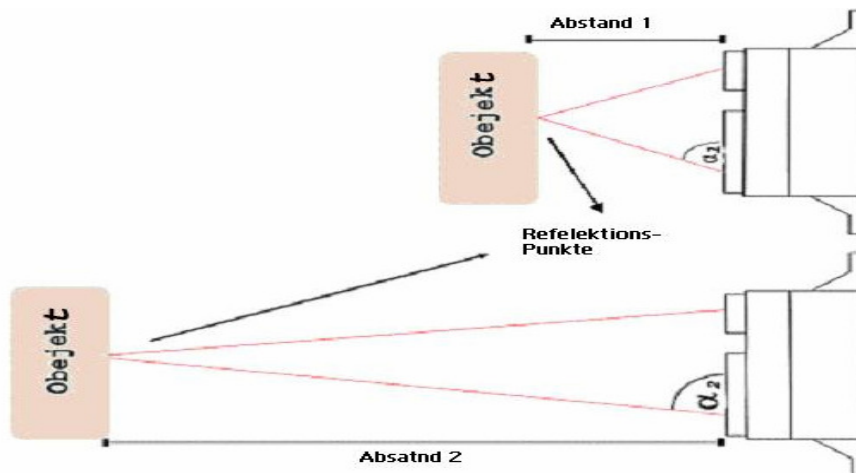


Bild 3-4:Arbeitsweisen der Distanzsensoren GP2D.

⁷ general Purpose type distance measuring sensors.

⁸ Triangulationsprinzip: Von Enden einer Standlinie (A-B) aus misst man die Winkel zu dem Punkt (C), von dem man die Entfernung bestimmen will.

Je nach Entfernung zu dem Objekt ändert sich der Einfallswinkel. Dadurch verschiebt sich das einfallende IR-Licht auf der LED-Zeile. Aus der Position des IR-Lichtes ermittelt der Sensor die Entfernung und gibt sie über U_{out} als Analogsignal aus. Dabei entsprechen 10 cm etwa 2,6 V und 80 cm etwa 0,4 V. Die Ausgangsspannung ist aber nicht linear zur Entfernung. Das Bild 3-5 zeigt die Kennlinie der Ausgangsspannung:

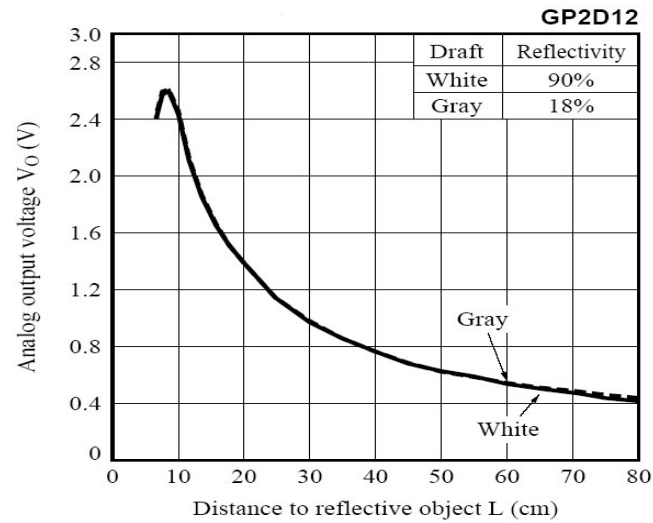


Bild 3-5: Ausgangsspannung abhängig von der Entfernung des Reflektionspunktes.

4. Lösungsentwurf

Das im Rahmen dieser Arbeit eingesetzte Modell-Fahrzeug ist mit einem Distanzsensor ausgestattet. Der Sensor soll Objekte auf der Fahrbahn des Fahrzeugs erkennen und deren Entfernung messen. Der Erfassungsbereich des Sensors wird in Verschiedene Bereiche unterteilt (vgl. Bild 4-1).

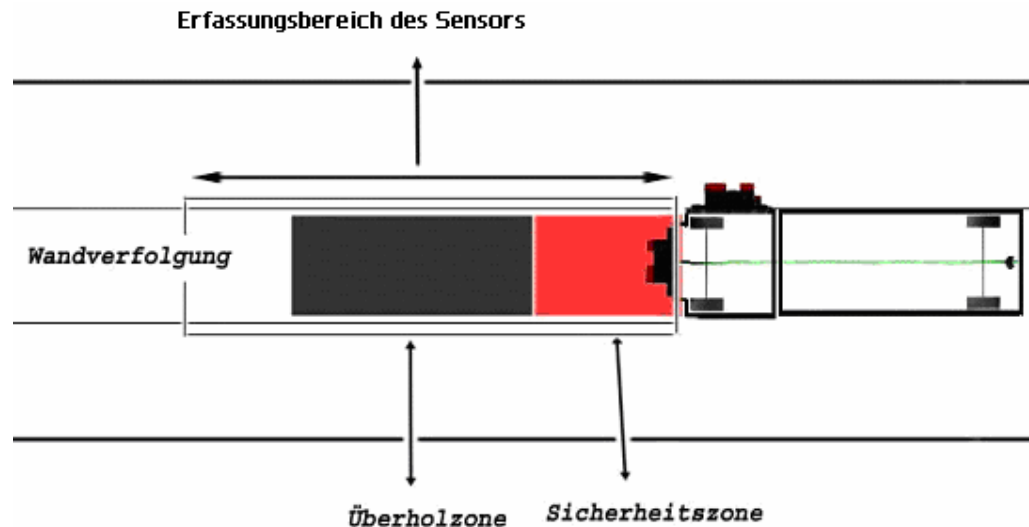


Bild 4-1: Aufteilung des Erfassungsbereichs des Sensors in verschiedenen Zonen.

Abhängig von der Entfernung des Objekts sollen folgende Fähigkeiten des Fahrzeugs aktiviert werden.

- ◆ Wandverfolgung: Ist die Fahrbahn frei, soll der Modell-Lastwagen entlang der Wand fahren.
- ◆ Überholmanöver: Wird ein Fahrzeug in der Überholzone entdeckt, soll ein Überholmanöver gestartet werden.
- ◆ Kollisionsvermeidung: Überschreitet ein Objekt den Sicherheitsbereich des Fahrzeugs, soll das Fahrzeug gestoppt werden.

Im diesem Kapitel wird ein Lösungsentwurf für die genannten Fähigkeiten erstellt. Diese Fähigkeiten entsprechen die Anforderungen, die im Abschnitt 1.2 dem autonomen Modell-Lastfahrzeug, das im Rahmen dieser Arbeit entwickelt werden soll, gestellt sind.

4.1. Wandverfolgung

Für ein autonomes Fahrzeug ist die Erkennung des Fahrbahnverlaufs eine wichtige Voraussetzung, um die ihm gestellten Aufgaben erfolgreich durchzuführen. Hierzu ist ein Mechanismus zur Bahnkontrolle erforderlich. Es muss ein Bahncontroller entwickelt werden, der dafür sorgt, dass das Fahrzeug auf der Fahrbahn bleibt. Dazu muss der

Bahncontroller die aktuelle Position des Fahrzeugs relativ zur Fahrbahn ermitteln. Abhängig von der Abweichung zwischen der aktuellen Position des Fahrzeugs und der Fahrbahn soll er dann entsprechend die Fahrtrichtung des Fahrzeugs korrigieren.

Der im Rahmen dieser Arbeit zu entwickelnde autonome Modell-Lastwagen soll mit einem vorgegebenen Sicherheitsabstand entlang der Wand fahren. Die Aufgabe des Bahncontrollers ist in diesem konkreten Fall, den Abstand des Modell-Fahrzeugs relativ zur Wand zu ermitteln. Ist der Abstand des Fahrzeugs zur Wand größer als der vorgegebene Sicherheitsabstand, soll der Bahncontroller die Fahrtrichtung des Fahrzeugs in Richtung der Wand korrigieren. Bei Überschreitung des Sicherheitsabstands soll er das Fahrzeug von der Wand weg steuern. Der Abstand des Fahrzeugs relativ zur Wand wird aus den externen Sensordaten berechnet.

Regler als Bahncontroller

Für die Bahnkontrolle des Modell-Lastwagens werden im Rahmen dieser Arbeit die Methoden der Regelungstechnik eingesetzt. Ein Regler soll die Fahrtrichtung des Modell-Lastwagens so regeln, dass das Fahrzeug in einem vorgegebenen Abstand möglichst parallel zur Wand fährt.

In Rahmen dieser Arbeit wird das Lenkgetriebe des Fahrzeugs an einem Servoport des im Abschnitt 3.2 beschriebenen Mikrocontrollers Aksen-Board angeschlossen. Den Abstand zur Wand liefert ein Distanzsensor.

Die Regelbedingungen des Reglers sind für diesen speziellen Fall wie folgt festgelegt:

- ◆ Der Sicherheitsabstand des Fahrzeugs von der Wand ist regelungstechnisch der Sollwert w .
- ◆ Die aktuelle Entfernung des Fahrzeugs von der Wand stellt die Regelgröße Istwert x dar.
- ◆ Der Vergleich zwischen Soll- und Istwert ergibt die Regeldifferenz e .
- ◆ Die Impulsbreite des Servoport ist regelungstechnisch die Stellgröße y . Die Stellgröße wirkt dann direkt auf die Lenkung des Fahrzeugs (Stellglied).
- ◆ Die Lenkung des Fahrzeugs (Stellglied) und der Sensor, der den Abstand des Fahrzeugs zur Wand misst (Messort) bilden die Regelstrecke (vgl. Bild 4-2).

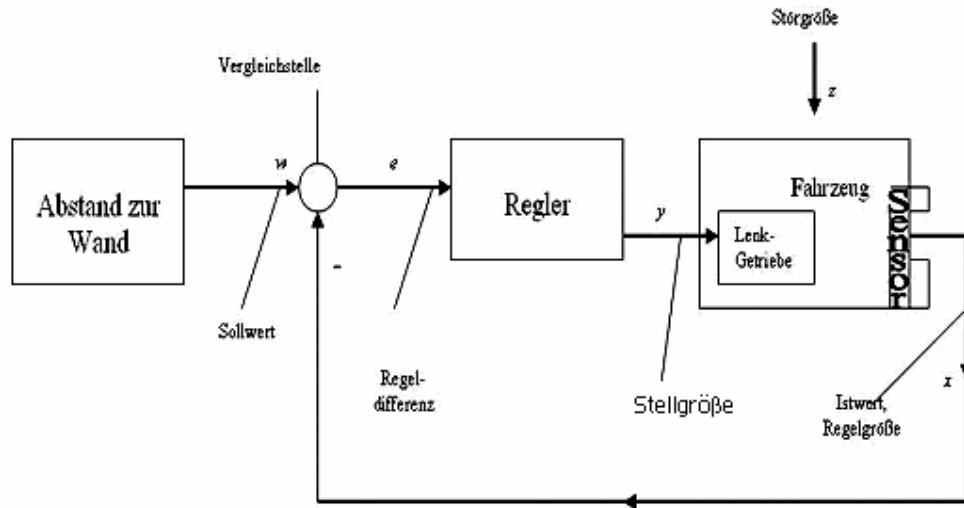


Bild 4-2: Blockschaltbild des P-Reglers.

Wie im Abschnitt 3.1 beschrieben, ist die Lenkung des Fahrzeugs für eine Impulsbreite $b = 1500\mu\text{s}$ gerade ($\alpha = 0$). Der Wertebereich von b liegt im Intervall $[1155\mu\text{s}, 1845\mu\text{s}]$. Demzufolge darf die Stellgröße das Intervall $[1155, 1845]$ nicht verlassen.

Setzt man einen P-Regler für die Regelung des Lenkgetriebes ein, liefert der Regler z.B. die Stellgröße $y = 0$, falls die Regelabweichung $e(t) = 0$ ist. Das ergibt sich aus der Gleichung des P-Reglers:

$$y(t) = K_p \cdot 0 = 0 \quad (4.1.1.1)$$

Die Gleichung des P-Reglers für diesen konkreten Fall muss so angepasst werden, dass der Regler bei der Regelabweichung $e = 0$ die Stellgröße $y = 1500$ liefert. Damit wird erreicht, dass bei der Regeldifferenz $e = 0$ das Fahrzeug gerade fährt. Es also muss eine Konstante y_0 (Offset) in der Gleichung des P-Reglers eingeführt werden, damit die folgende Bedingung erfüllt ist:

$$y(t) = y_0 + K_p \cdot 0 := 1500 \quad (4.1.1.2)$$

In diesem Fall ist $y_0 = 1500$. Wie schon im Abschnitt 2.2.2 beschrieben, bleibt bei dem P-Regler eine gewisse Regelabweichung. Je nach Einsatzgebiet fällt diese Regelabweichung unterschiedlich aus. Für die Berechnung der bleibenden Regelabweichung muss das Verhalten des Reglers für $t = \infty$ untersucht werden, da sich das Regelsystem für $t = \infty$ stabilisiert. Die Stellgröße y wird für $t = \infty$ mathematisch wie folgt berechnet:

$$\lim_{t \rightarrow \infty} y(t) = y_0 + K_p \cdot e(t) \quad (4.1.1.3)$$

Bei dem P-Regler nähert sich die Reglerabweichung für $t = \infty$ einem Grenzwert:

$$\lim_{t \rightarrow \infty} e(t) = c$$

c ist die bleibende Regelabweichung. Setzt man das Ergebnis in die Gleichung (4.1.13) ein, ergibt sich für $t = \infty$:

$$y = y_0 + K_p \cdot c \quad (4.1.1.4)$$

Beim Einsetzen des P-Reglers für die Regelung der Fahrtrichtung des Modell-Lastwagens entsteht die Regelabweichung c dadurch, dass beim Erreichen des Sollwerts ($e = 0$) das Fahrzeug aus folgendem Grund nicht genau parallel zu der Sollbahn steht: Gibt es eine Differenz zwischen Soll- und Istwert, korrigiert der P-Regler die Fahrtrichtung des Fahrzeugs, indem er den Lenkeinschlag des Lenkgetriebes entsprechend anpasst. Das Fahrzeug erreicht die Sollbahn mit einem bestimmten Winkel zur ihr. (Siehe Bild 4-3.)

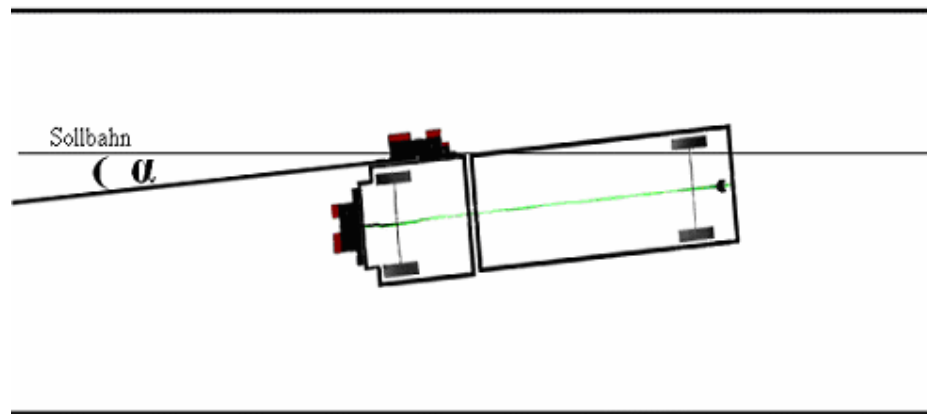


Bild 4-3: Stellung des Modell-Lastwagens bei der Regelabweichung $e = 0$.

Dann stellt der P-Regler die Lenkung gerade ($y(t) = y_0 + K_p \cdot 0 = y_0$). Auf Grund der Bewegung des Fahrzeugs entsteht wieder eine Regeldifferenz ($e \neq 0$). Ist die Regelabweichung gering, reagiert der Regler zu langsam. Denn für geringe Regelabweichung liefert der P-Regler auch geringe Werte für die Stellgröße y (siehe die Gleichung (2.2.2.1)).

Löst man die Gleichung ((4.1.1.4)) nach c , ergibt sich

$$c = \frac{y - y_0}{K_p} \quad (4.1.1.5)$$

Wie die Gleichung (4.1.1.5) zeigt, ist c indirekt proportional zu K_p . Um die bleibende Regelabweichung c zu minimieren, muss der Proportionalitätsfaktor des P-Reglers so groß wie möglich gewählt werden, damit auch bei kleinerer Regelabweichung der Regler die Lenkung gut beeinflusst. Der Regler soll aber die Stabilitätsgrenze⁹ nicht überschreiten. Für die Bestimmung des größtmöglichen Werts für K_p kann das Verfahren

⁹Die Grenze, ab der der Regler nicht mehr zuverlässig arbeitet, und das zu regelnde System in starken Schwankungen gerät.

von Ziegler/Nichols angewendet werden[GEER-04]. Das Verfahren funktioniert wie folgt:

- ◆ K_p einen kleinen Wert zuweisen
- ◆ Das Verhalten des Systems beobachten.
- ◆ Reagiert der Regler zu langsam, K_p -Wert erhöhen und das Verhalten des Systems beobachten.
- ◆ Den Punkt 3 solange durchführen, bis das System anfängt zu schwingen.
- ◆ Ablesen von $K_{P,Kr}$ (kritische K_P , der Wert, bei dem das zu regelnde nicht mehr stabil ist) und die Periodendauer der Schwingung T_{Kr} bei $K_{P,Kr}$.

Um festzustellen, ob die bleibende Regelabweichung c beim Einsetzen des P-Reglers vernachlässigbar klein ist, muss das Fahrverhalten des Modell-Lastwagens experimentell untersucht werden. Falls der P-Regler die Fahrtrichtung des Fahrzeugs nicht gut genug regelt, weil die bleibende Regelabweichung zu groß ist, muss dann ein PI-Regler (siehe Abschnitt 2.2.2) eingesetzt werden, um die Regelabweichung zu beseitigen. Anhand folgender Tabelle kann die Parameter der PI-Regler berechnet werden.

	K_P	T_n
P-Regler	$K_P = K_{P,Kr} * 0.5$.
PI-Regler	$K_P = K_{P,Kr} * 0.45$	$T_n = T_{Kr} * 0.85$

Tabelle 4-1: Berechnung der PI-Regler Parameter[GEER-04]

4.2. Kollisionsvermeidung

Die Sicherheit hat während der Bewegungsoperation bei autonomen Fahrzeugen die höchste Priorität, weil diese Fahrzeuge auf sich allein gestellt sind. Im Falle einer Kollisionsgefahr mit Hindernissen müssen die Fahrzeuge in der Lage sein, schnell zu reagieren, um die Sicherheit in allen Situationen gewährleisten zu können.

Für die Kollisionsvermeidung ist ein robustes Verhalten des Systems beim Umgang mit unsicherer und ungenauer Information über das Vorhandensein von Hindernissen erforderlich. Die Unsicherheit ist auf ungenaue oder fehlerhafte Messdaten bei der Erfassung der Umgebung durch Sensoren zurückzuführen. Die Realisierung einer umfangreichen Hindernisvermeidungs-Strategie würde den Rahmen dieser Arbeit sprengen. Deswegen wurde im Rahmen dieser Arbeit die Vermeidung von Kollisionen auf einem Notstopp beschränkt.

4.3. Das Überholmanöver

Die Realisierung des Überholvorgangs im Rahmen dieser Arbeit beschränkt sich auf nicht-kooperatives Überholen. Das überholende Fahrzeug soll den Überholvorgang so durchführen, dass keine Kollision mit dem überholten Fahrzeug entsteht. Für die Realisierung des Überholvorgangs wird zuerst folgendes Szenario vorgestellt.

Der zu überholende Modell-Lastwagen LKW_1 hat die Länge l_1 und fährt mit der Geschwindigkeit v_1 entlang der Wand. Der zweite Modell-Lastwagen LKW_2 hat die Länge l_2 und folgt dem LKW_1 mit der Geschwindigkeit v_2 . Es gilt $v_2 > v_1$.



Bild 4-4: Stellung der Modell-Fahrzeuge vor dem Überholmanöver.

Das LKW_2 soll mit einem Distanzsensoren für die Erkennung des vorausfahrenden Fahrzeugs LKW_1 ausgestattet sein. Falls der LKW_1 im Erfassungsbereich des Sensors erscheint, soll die Geschwindigkeit des LKW_2 relativ zum LKW_1 berechnet werden, indem das Rechensystem des LKW_2 die Abstandsdifferenz Δx zwischen den Fahrzeugen in der Zeit Δt berechnet.

Der LKW_2 soll dann zum Überholen die „Fahrspur“ wechseln und soll die Strecke zurücklegen, die der LKW_1 während der Überholzeit zurücklegt. Es soll zusätzlich noch den Sicherheitsabstand s_1 , die eigene Länge l_2 und die Länge l_1 des LKW_1 berücksichtigen. Der Sicherheitsabstand soll doppelt zurückgelegt werden (vor und nach dem Überholen). (Siehe Bild 4-5).

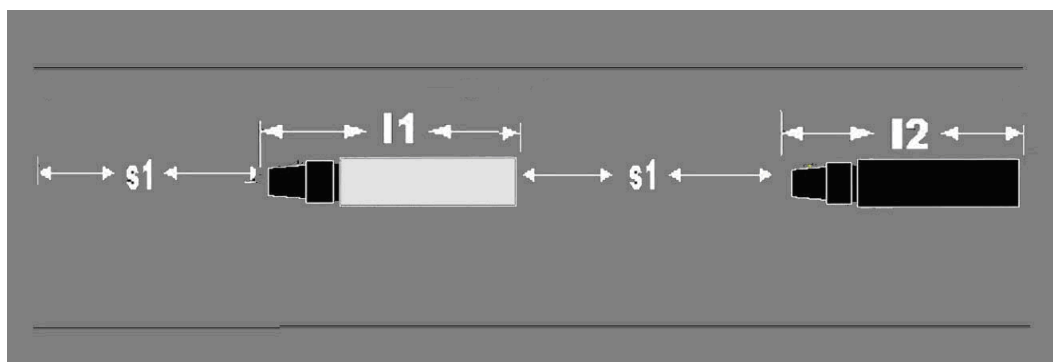


Bild 4-5: Sicherheitsabstände zwischen den Fahrzeugen.

Als gesamte Überholstrecke ergibt sich die Strecke S_{gesamt} :

$$S_{gesamt} = v_1 \cdot \Delta t + 2 \cdot S_1 + l_1 + l_2$$

S_{gesamt} : Überholstrecke

v_1 : Geschwindigkeit von LKW₁

Δt : Überholzeit

S_1 : Sicherheitsabstand

l_1 : Länge von LKW₁

l_2 : Länge von KW₂

Die Überholzeit Δt berechnet sich wie folgt:

$$\Delta v = \frac{2 \cdot S_1 + l_1 + l_2}{\Delta t}$$

$$\Delta v \cdot \Delta t = 2 \cdot S_1 + l_1 + l_2 \quad (\text{Beide Seiten mit } \Delta t \text{ multipliziert.})$$

$$\Delta t = \frac{2 \cdot S_1 + l_1 + l_2}{\Delta v} \quad (\text{Beide Seiten durch } \Delta v \text{ dividiert.})$$

Bei dem Überholvorgang soll zuerst der Sollwert des P-Reglers, der für die Bahnkontrolle des Modell-Lastwagens zuständig ist, so verändert werden, dass KW₂ ohne Kollisionsgefahr das LKW₁ überholt.

Nach Vergehen der Überholzeit soll dem Sollwert des P-Reglers wieder der vordefinierte Wert zugewiesen werden.

4.4. Steuerungsarchitektur des Systems

Als Steuerungsarchitektur dieser Arbeit wird die im Abschnitt 2.1.2 beschriebene Subsumptions-Methode von Brooks gewählt. Das Konzept ermöglicht die parallele Verarbeitung der einzelnen Aktivitäten (Verhaltensweisen) des autonomen Modell-Fahrzeugs. Das erhöht die Robustheit des autonomen Fahrzeugs in kritischen Fällen und das Steuerungssystem kann die an es gestellten Echtzeitanforderungen, zum Beispiel Vermeidung von Kollisionen, gerecht werden.

Das Steuerungssystem des in Rahmen dieser Arbeit zu entwickelnden Modell-Fahrzeugs ist in verhaltensbasierte Subkomponenten geteilt. Diese Subkomponenten werden in Rahmen dieser Arbeit als Kontrollschichten bezeichnet. Die einzelnen Kontrollschichten können schrittweise entwickelt werden. Dies ermöglicht es, jede Kontrollschicht einzeln zu entwickeln und zu testen. Auf diese Weise werden die Fehlerquellen schnell gefunden und behoben.

Außerdem kann ein Fehler in einer Kontrollschicht nicht die Funktionalität der anderen Schichten stören. Das Steuerungssystem kann durch Hinzufügen weiterer Kontrollschichten erweitert werden.

Das Einsetzen der Subsumptions-Architektur bringt folgende Vorteile mit sich:

- ◆ Einzelnes autonomes Verhalten.
- ◆ Schneller Zugriff auf Sensordaten.
- ◆ Geringer Rechenaufwand.
- ◆ Robustheit gegen Sensorausfall.
- ◆ Einfach erweiterbar.
- ◆ Aufgrund der Unterteilung des Gesamtverhaltens in einzelne Verhaltensweisen einfach implementierbar.

4.4.1. Entwurf des Systems nach der Subsumptions-Architektur

Für den Entwurf des Steuerungssystems nach der Subsumptions-Architektur wird wie folgt vorgegangen:

Zuerst wird die Kontrollschicht K_0 definiert. Dann werden die Module (Einzelverhalten) von K_0 festgelegt. Danach die Kontrollschicht K_1 und deren Module definiert. Analog werden alle weiteren Kontrollschichten (K_2, \dots, K_n) und deren Module definiert. Die Kontrollschichten werden angefangen von unteren bzw. von einfacheren bis hin zu komplexeren Kontrollschichten entworfen (Siehe Bild 4-6).

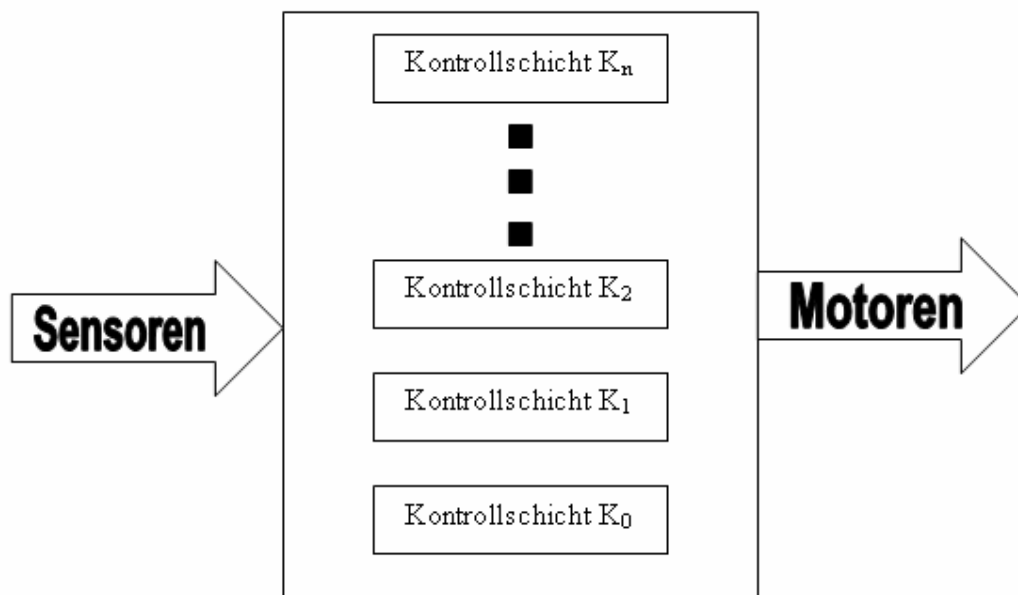


Bild 4-6: Parallelllaufende Kontrollschichten.

4.4.2. Definition der Kompetenzebenen

Der Kontrollschicht K_0 : „Kollisionsfreies Fahren“

In dieser Schicht wird das Fahren unter Vermeidung von Kollisionen realisiert. Das Fahrzeug fährt mit der Geschwindigkeit v . Befindet sich ein Hindernis in der Sicherheitszone des Fahrzeugs (siehe Abschnitt 4) oder liefern die Sensoren keine Daten, werden alle Motoren ausgeschaltet. Das verhindert die Beschädigung des Fahrzeugs und der Objekte seiner Umgebung im Falle einer Kollisionsgefahr.

Folgende Module (Verhalten) beinhaltet die Kontrollschicht K_0 :

- ◆ *notstopp*: Es stoppt das Fahrzeug im Falle einer Kollisionsgefahr oder wenn die Sensoren ausfallen.
- ◆ *fahren*: Es setzt das Fahrzeug in Bewegung mit der Geschwindigkeit v und geradeaus Lenkung.

Das Kontrollschicht K_1 : „Wandverfolgung“

In dieser Schicht wird das geregelte Fahren entlang der Wand realisiert. Das Fahrzeug soll entlang der Wand fahren und einen Sicherheitsabstand (Sollwert) einhalten. Ein Regler soll die Lenkung des Fahrzeugs so regeln, dass die Differenz aus dem Istwert (momentanem Abstand) und Sollwert möglichst gering bleibt.

Befindet sich ein langsamerer Fahrzeug vor ihm, soll ein Überholmanöver durchgeführt werden. Die Kontrollschicht K_1 besteht aus folgenden Modulen:

- ◆ *wandverfolgung*: Aktiviert das Modul *regler*, falls die Fahrbahn frei ist, damit der Regler die Fahrtrichtung entlang der Wand führt. Ist die Fahrbahn nicht frei, wird das Modul *ueberholen* aktiviert, damit ein Überholmanöver durchgeführt wird.
- ◆ *regler*: Regelt die Lenkgetriebe.
- ◆ *ueberholen*: Dieses Modul beinhaltet die Realisierung eines Überholvorgangs. Das Überholmanöver wird, wie im Abschnitt 4.3 beschrieben, durchgeführt.

Die einzelnen Kontrollschichten und deren Verbindungen zu einander werden durch einen endlichen Automaten dargestellt.

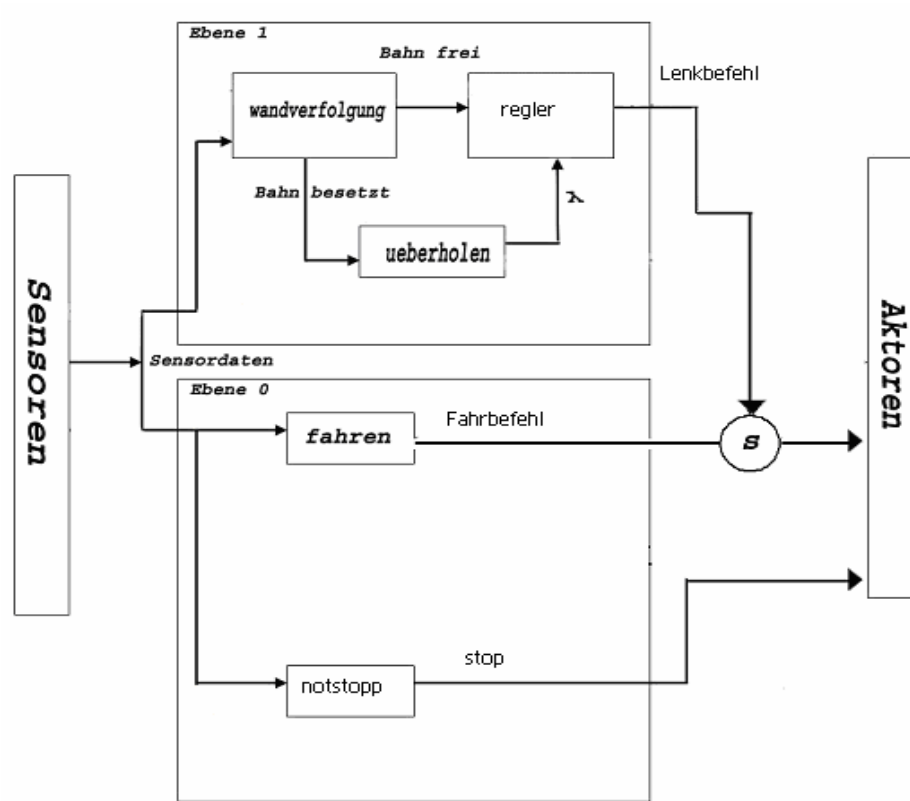


Bild 4-7: Automatenmodell des Steuerungssystems.

- ◆ *sensoren*: Ermittelt aus den Sensordaten die Entfernung der Hindernisse und stellt sie den anderen Prozessen zur Verfügung.

Zu den Aktoren zählen die Methoden, die Motoren des Modell-Lastwagens steuern. Wie das Bild 4-7 verdeutlicht, wird die Ausgabe des Moduls *notstopp* durch die Kontrollschicht K_1 unterdrückt.

5. Implementierung

5.1. Systemarchitektur

Die Hauptkomponenten des Systems sind die im Kapitel 3 beschriebenen Modell-Lastwagen, das Aksen-Board und die Infrarot-Distanzsensoren. Weiterhin beinhaltet das System einen Spannungsverstärker für das Anschließen der Sensoren an das Aksen-Board, einen Bleiakku (12 V) für die Stromversorgung des Modell-Lastwagens und ein NiCd Akku (7.2 V und 2400 mAh) für die Stromversorgung des Aksen-Boards und des Spannungsverstärkers.

Alle Komponenten und deren Verkabelung sind im Bild 5-1 aufgezeichnet.

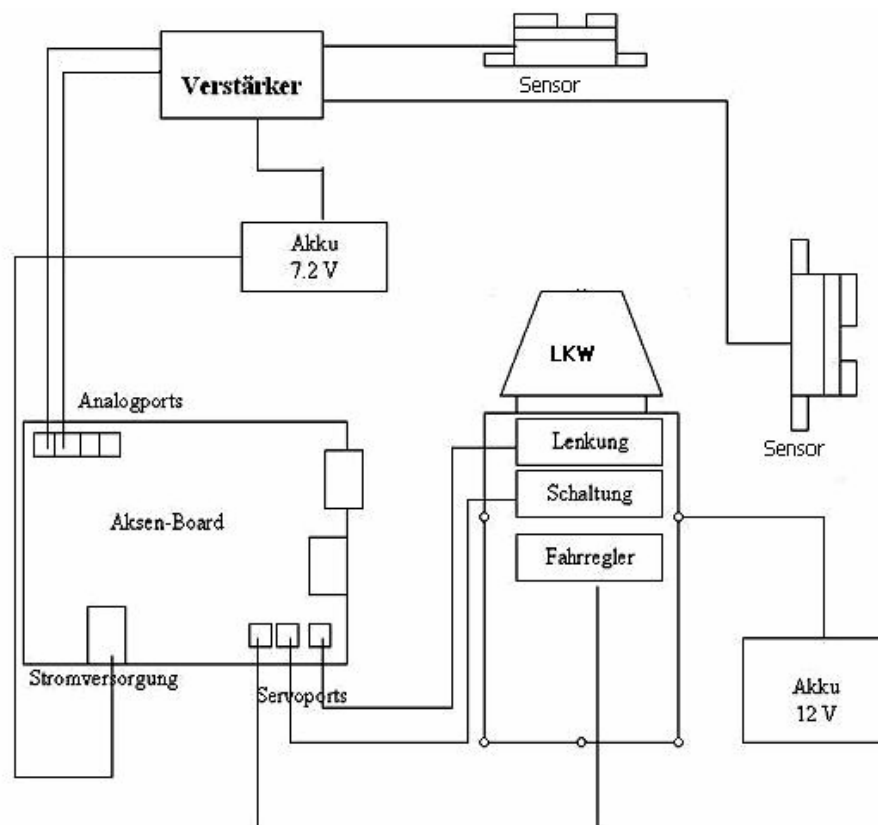


Bild 5-1: Architektur des Systems

Die Motoren des Modell-Lastwagens sind wie folgt an das Aksen-Board angeschlossen: Die Lenkung des Modell-Lastwagens wurde an den ersten Servo-Ausgang, der Fahrregler an den zweiten Servo-Ausgang und die Schaltung an den dritten Servo-Ausgang des Aksen-Boards angeschlossen.

Die Analog-Eingänge des Aksen-Boards sind mit einem Pull-up-Widerstand verbunden. Die Stromstärke von der Spannung, die der eingesetzte Infrarot-Distanzsensor sendet, ist zu schwach, um die Spannung, die durch diesen Pull-up-Widerstand auf den analo-

gen Eingang anliegt, runter zu ziehen. Um das Problem zu umgehen, wurde ein Spannungsverstärker benutzt (vgl. Bild 5-2).

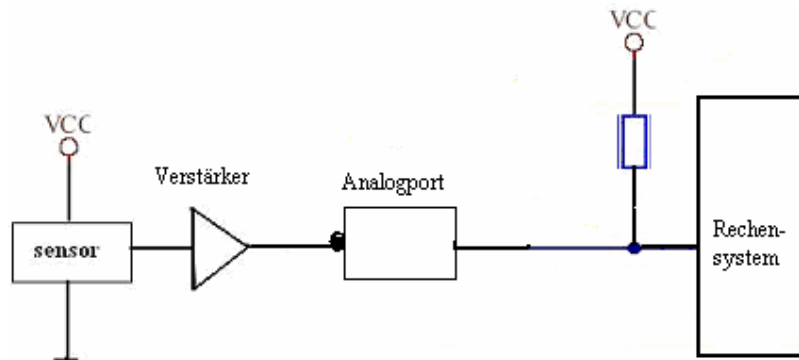


Bild 5-2: Schaltungsaufbau der analogen Eingänge.

Der Verstärker besetzt vier Ports. Maximal können vier Sensoren an den Verstärker angeschlossen werden. Die Distanzsensoren wurden an den Verstärker angeschlossen und der Verstärker wiederum an den analogen Eingang des Aksen-Boards. An den Modell-Lastwagen wurden zwei Distanzsensoren montiert.

Ein Sharp GP2D12 an der rechten Kotflügel für die Ermittlung der Distanz zwischen dem Modell-Lastwagen und der Wand sowie ein GP2D15 an der vorderen Stoßstange.

Wie im Kapitel 3 schon erläutert, liefern die Sharp Distanzsensoren abhängig von der Entfernung eine Ausgangsspannung im Bereich 0,4 bis 2,5 V. Diese Spannung wird mittels des Aksen-Boards in ein digitales Signal umgewandelt. Das Signal besteht aus 8 Bits, somit können die Zahlen von 0 bis 255 kodiert werden. Dabei liefert das Aksen-Board bei einer minimalen Spannung von 0 V die Zahl 0 und bei einer maximalen Spannung von 5 V die Zahl 255 ab. Diese Zahlenwerte werden im Rahmen dieser Arbeit als Sensorwerte bezeichnet.

Da die Ausgangsspannung des Sensors nicht linear zu der Entfernung ist, wurden in einem Versuch abhängig von den einzelnen Entfernungen die zugehörigen Sensorwerte ermittelt. Die Ergebnisse wurden anschließend in einer Tabelle erfasst. Das Bild 5-3 zeigt die Versuchsanordnung und das Bild 5-4 gibt die Ergebnisse des Versuchs wieder.

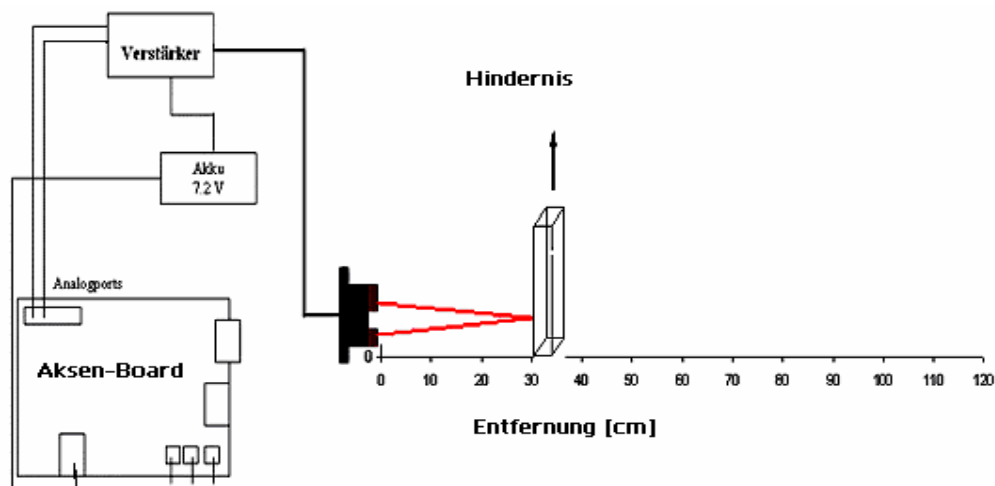


Bild 5-3: Bestimmung der Sensorwerte abhängig von der Entfernung.

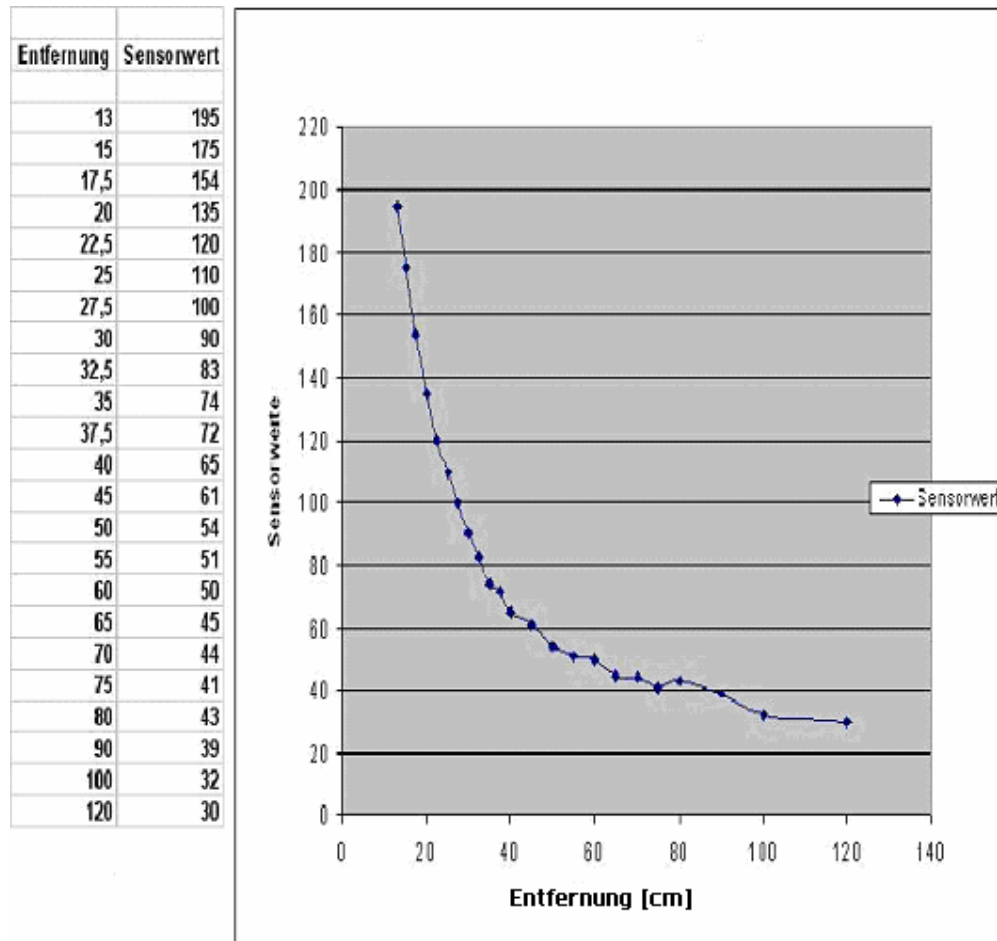


Bild 5-4: Kennlinie der Sensorwerte abhängig von der Entfernung.

Je nach Objektentfernung liefert der Sensor Werte zwischen 30 bis 195. Somit ist der Definitionsbereich der Sensorwerte auf den Intervall $[30,195]$ beschränkt. Es wurde auch festgestellt, dass beim Ausfall des Sensors, der Analogport den Wert 255 liefert. Demzufolge kann die Software den Ausfall des Sensors erkennen, in dem sie die Werte der Analogports, an denen die Sensoren angeschlossen sind, analysiert.

Im folgendem soll die Kennlinie des Sensors (vgl. Bild 5-4) im Bereich $x \in [10\text{cm}, 100\text{cm}]$ durch eine Funktion $f(x)$ näherungsweise abgebildet werden. Wie in diesem Graph zu sehen ist, handelt es sich um eine Hyperbel der Form

$$f(x) = c_1 \cdot \frac{1}{x} + c_2 \quad (5.1.1)$$

Mit Hilfe der Umkehrfunktion von (5.1.1) kann dann die Entfernung für beliebige Sensorwerte im Bereich $[30,195]$ berechnet werden. Zuerst werden die zwei unbekanntenen Größen c_1 und c_2 der Funktion (5.1.1) ermittelt. Die zwei unbekanntenen Größen c_1 und c_2 müssen so gewählt werden, dass der Messfehler minimal bleibt. Um das zu erreichen, wurde für die Ermittlung der Unbekannten c_1 und c_2 die im Abschnitt 2.3 beschriebene „kleinste Fehlerquadratmethode“ angewendet. Für die Bestimmung der Parameter der Funktion (5.1.1) wurde wie im Abschnitt 3.1 vorgegangen. Dies Lösung liefert das Ergebnis: $c_1 = 2556$ und $c_2 = 7$ (Die Zahlen sind auf Ganzzahlen gerundet). Die gesuchte Gleichung lautet somit

$$f(x) = 2556 \cdot \frac{1}{x} + 7 \quad (5.1.2)$$

Da aber die Entfernung aus den Sensorwerten berechnet wird und nicht umgekehrt, soll die Umkehrfunktion von (5.1.2) berechnet werden.

$$x(f) = 2556 \cdot \frac{1}{(f - 7)} \quad (5.1.3)$$

Mit Hilfe der Gleichung (5.1.3) kann dann aus den Sensorwerten die Entfernung berechnet werden.

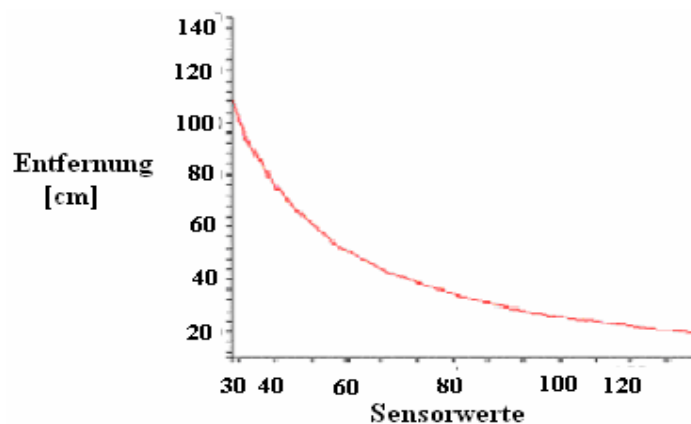


Bild 5-5: Kennlinie der Funktion (5.1.3).

Der Sharp-Sensor GP2D15 erfasst Objekte in Bereich 30 bis 150cm. Analog zu dem Sensor GP2D12 wurde die Funktion (5.1.3) für die Berechnung der Entfernung aus den Sensorwerten von GP2D15 ermittelt.

$$x(f) = (5927 \cdot \frac{1}{(f-1)}) \quad (5.1.7)$$

Messabweichungen der Sensoren

Die beiden Distanzsensoren GP2D12 und GP2D15 erweisen zufällige Messabweichungen. Diese Abweichungen fallen je nach Entfernung des Messobjekts unterschiedlich aus. Die absolute zufällige Messabweichung Δx (siehe Abschnitt 2.1.1.7) des Sensors GP2D12 abhängig von der Entfernung beträgt

$$\Delta x \leq \left\{ \begin{array}{l} \pm 3 \text{ bei } 10\text{cm} \leq x_w \leq 60\text{cm} \\ \pm 7 \text{ bei } 60\text{cm} < x_w \leq 70\text{cm} \end{array} \right\}$$

und

$$\Delta x \geq \left\{ \pm 10 \text{ bei } 70\text{cm} < x_w < 100\text{cm} \right\}$$

Die absolute zufällige Messabweichung des Sensors GP2D15 abhängig von der Entfernung beträgt

Messabweichung

$$\Delta x \leq \left\{ \begin{array}{l} \pm 3 \text{ bei } 30\text{cm} \leq x_w \leq 70\text{cm} \\ \pm 7 \text{ bei } 70\text{cm} < x_w \leq 100\text{cm} \end{array} \right\}$$

Und

$$\Delta x \geq \left\{ \pm 10 \text{ bei } 100\text{cm} < x_w < 150\text{cm} \right\}$$

- ◆ x_w : reale Entfernung des Test-Objekts.
- ◆ Δx : absolute zufällige Messabweichung.

In Rahmen dieser Arbeit beträgt der Sicherheitsabstand des Modell-Lastwagens zur Wand 40cm (das ist die mittlere Erfassungsreichweite des Sensors GP2D12). Die Messabweichungen des Sensors kann in Rahmen dieser Arbeit vernachlässigt werden, da die Messabweichungen des Sensors bei 40cm unter 1cm liegen. Im Grenzbereich arbeitet der Sensor nur für kurze Zeit, somit haben die Messabweichungen des Sensors in Grenzbereich keine großen Auswirkungen.

Der Sensor GP2D15 wird für die Erkennung der Objekte oder andere Fahrzeuge auf der Fahrbahn eingesetzt. Die zufälligen Messabweichungen dieses Sensors sind bis zu einer Entfernung von 100cm noch tolerierbar. Bei Entfernungen größer 100cm sind die zufälligen Messabweichungen zu groß. So kann zum Beispiel die Überholstrecke und somit die Überholzeit falsch berechnet werden.

Programmiersprache

Die Verwaltungsinstanz des autonomen Modell-Lastwagens ist das Aksen-Board. Wie im Abschnitt 3.2 schon erwähnt, werden die Programme für das Aksen-Board in C geschrieben. Das Aksen-Board unterstützt auch das Multitasking.

Das Multitasking ist nach Round-Robin-Algorithmus ohne Prioritätsvergabe realisiert. Bei diesem Algorithmus werden für die Prozesse eine Liste erstellt. Jeder Prozess bekommt eine Zeitscheibe. Der Prozessor wird den Prozessen der Reihe nach zu Verfügung gestellt. Die Prozesse nehmen den Prozessor in Anspruch, solange ihre Zeitscheibe nicht überschritten ist. Ist ihre Zeitscheibe abgelaufen, werden sie am Ende der Liste angehängt (vgl. Bild 5-6).

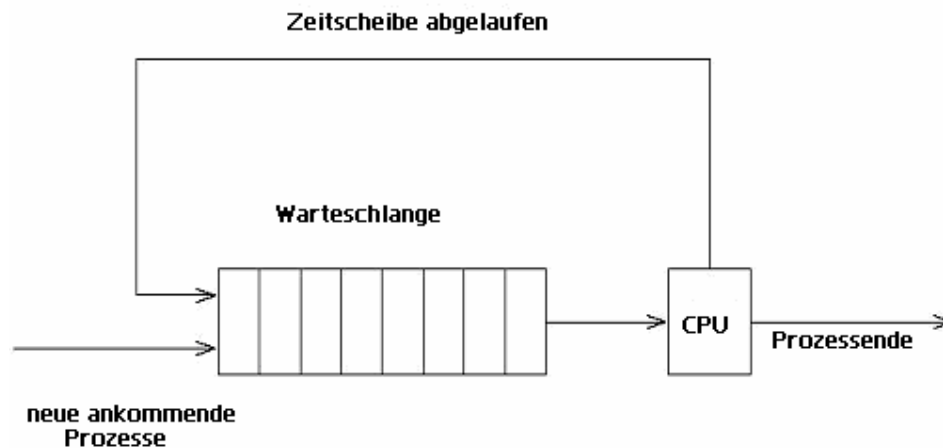


Bild 5-6: Prozesswechsel nach Round-Robin-Algorithmus.

Das Aksen-Board unterstützt insgesamt 20 „parallelaufende“ Prozesse. Ein Prozess wird wie folgt erzeugt:

```
process_start( void (*funktionszeiger)(), unsigned char zeit)
```

Im ersten Parameter wird ein Zeiger auf eine ganz normale Funktion ohne Parameter erwartet, im zweiten Parameter wird die Verweilzeit¹⁰ in Millisekunden erwartet. Als Rückgabewert wird die ID des erzeugten Prozesses zurückgegeben. Wird als Rückgabewert eine Zahl größer 19 geliefert, handelt es sich um eine Fehlersituation, und der Prozess wird nicht bearbeitet. Durch den Aufruf von

```
process_kill(unsigned char pid)
```

¹⁰ Verweilzeit: Ausführungszeit. Die Zeit, die der Prozessor einem Prozess zur Verfügung steht.

kann ein Prozess beendet bzw. aus der Prozessliste entfernt werden. Als Übergabeparameter wird die Prozess-ID erwartet. Gibt es keinen Prozess mit der übergebenen Prozess-ID, wird die Konstante `PROCESS_ERROR` zurückgegeben. Durch den Aufruf von `process_get_pid()`; kann ein Prozess seine eigene Prozess-ID ermitteln. Mit dem Aufruf der `-`Methode

```
process_kill(processprocess_get_pid());
```

kann ein Prozess sich selbst beenden. mittels `process_defer()` kann ein Prozess eine vorzeitige Abgabe seiner Zeitscheibe und somit einen Prozesswechsel erzwingen. Für detaillierte Erläuterung des Multitasking mit dem Aksen-Board siehe bitte [HAND-05].

Das Aksen-Board liefert eine gute Grundlage für die Realisierung des im Kapitel 4 entworfenen Verhaltensbasierten Steuerungssystems. Das Steuerungssystem wird in C programmiert. Für jede Kontrollschicht wird ein Prozess erzeugt. Die Verweilzeiten werden experimentell festgelegt.

5.2. Realisierung des Steuerungssystems

Die Implementierung des Steuerungssystems des Modell-Lastwagens erfolgt nach der im Abschnitt 4.4.2 entworfenen Steuerungsarchitektur (vgl. Bild 4-7). Für die Steuerung der Motoren des Modell-Lastwagens wurden die drei Funktionen

```
void fahr(unsigned char geschwindigkeit, signed char richtung );
void lenk(unsigned int servosignal);
void schalt(unsigned char gang);
```

implementiert. Die Funktion `fahr(...)` steuert den Fahrmotor des Modell-Lastwagens. Der erste Parameter dieser Funktion ist die Geschwindigkeit, mit der der Modell-Lastwagen betrieben werden soll. Mit Hilfe der Umkehrfunktion der Gleichung (3.1.8) im Abschnitt 3.1 wird aus der übergebenen Parameter `geschwindigkeit` die zugehörige Fahrstufe berechnet.

$$n(f) = \left(\frac{-1,129 + \sqrt{(1,274 + 0,334 \cdot f)}}{0,167} \right) \quad (5.2.1)$$

f : die Geschwindigkeit

Da das Aksen-Board die Bibliothek `Math.h` nicht unterstützt, wurde die Methode `wurzel(..)` implementiert. Diese Methode erwartet als Übergabeparameter eine Zahl x und liefert als Rückgabewert die Wurzel von x .

Der zweite Parameter der Funktion `fahr(...)` `richtung` bestimmt die Richtung der Fahrt. Für `richtung = 1` wird vorwärts gefahren und für `richtung = (-1)` rückwärts.

Die Funktion `lenk (...)` steuert das Lenkrad. Als Übergabeparameter wird die Größe der Impulsbreite des Servosignals erwartet. Mit Hilfe der dritten Funktion `schalt(...)` wird das Schaltgetriebe des Modell-Lastwagens gesteuert. Der Modell-Lastwagen kann in

drei Gängen betrieben werden. Der Übergabeparameter *gang* dieser Funktion soll demzufolge $1 \leq \text{gang} \leq 3$ sein.

Die im Abschnitt 4.4.2 entworfene Kontrollschicht K_0 „Kollisionsfreies Fahren“ wurde in zwei Modulen implementiert. Diese Kontrollschicht kann drei Zustände [*FAHREN*, *NOTSTOPP*, *NONE*] annehmen. In dem Modul *kontroll_0* wird abhängig vom Wert der Variablen der Struktur *Entfernung* einer dieser drei Zustände gewählt, und dann abhängig vom Zustand wird eins der Module *fahren* oder *notstopp* aktiviert. Falls der gewählte Zustand *NONE* ist, wird durch den Aufruf von *process_defer()*; ein Prozesswechsel erzwungen.

Damit das Hinzufügen weiterer Kontrollschichten ohne zusätzliche Anpassung der existierenden Kontrollschichten erfolgen kann, wurde der Zugriff der Kontrollschichten auf die Motoren des Fahrzeugs nicht direkt realisiert, sondern mittels des Prozesses *entscheider*. Dieser Prozess ist allein für das Auswählen und die Ausführung der Aktionen zuständig.

Das verhindert einerseits die widersprüchlichen Befehle, die durch gleichzeitige Aktivierung mehrerer Verhaltensweisen entstehen könnten, und andererseits ermöglicht es die Unterdrückung der Ausgaben der unteren Kontrollschichten durch höhere Kontrollschichten. Aus diesem Grund werden die Befehle der Kontrollschicht K_0 nicht direkt an die Motoren des Modell-Lastwagens gesendet, sondern in der Struktur

```
struct Kontroll_0
{
    int lenk_kontroll_0;
    int fahr_kontroll_0;
    int schalt_kontroll_0;
} k_0;
```

gespeichert. Diese Befehle werden dann von dem Modul *entscheider* ausgeführt.

Die Kontrollschicht K_1 „Wandverfolgung“ soll das geregelte Fahren entlang der Wand ermöglichen und einen Überholvorgang realisieren. Die Kontrollschicht K_1 besetzt die vier Zustände[WANDVERFOLGUNG, UEBERHOLEN, REGLER, NONE]. In dem Modul *kontroll_1* wird abhängig vom aktuellen Zustand von K_1 eins der im Abschnitt 4.4.2 entworfenen drei Modulen aktiviert.

```
void kontroll_1(){
    while(!stop){
        switch(zustand_1)
        {
            case WANDVERFOLGUNG :
                wand_Verfolgung();
                break;
```

```

case REGLER :
    pRegler();
    break;
case UEBERHOLE N :
    ueberholen ();
    break;
case NONE :
    process_de fer();
    break;
    }
    }

```

Wie aus dem Code zu sehen ist, beinhaltet das Modul *kontroll_1* eine While-Schleife. Die While-Schleife ist aktiv, solange die Variable *stopp*¹¹ nicht auf eins gesetzt ist. Fall der Zustand *NONE* ist, wird der Prozesswechsel erzwungen, damit der Prozessor nicht unnötig besetzt bleibt.

Der Programmcode dieser Kontrollschicht ist in drei Modulen implementiert. Das Modul *p-Regler* berechnet abhängig von der Differenz der Variablen *sollWert* und *istWert* die Impulsbreite des Servosignals für das Lenkgetriebe (Stellgröße). Die globalen Variablen des Moduls *p-Regler* sind in der Struktur

```

struct Regler
{
    char kp;
    char sollWert;
    unsigned char *istWert; // Zeiger auf wandAbstand
} regler;

```

definiert. Das Modul *wandverfolgung* aktiviert den *P-Regler*, falls der Wert der globalen Variable *hindernisAbstand* ≥ 100 (wenn bis zu einer Entfernung von 100cm kein Hindernis zu erkennen ist). Bei

$SICHERHEITSSZONE \leq \text{hindernisAbstand} \leq 100$

wird das Modul *ueberholen* aktiviert.

Das Modul *ueberholen* ermittelt durch den Aufruf der Methode *v_relativ()* die relative Geschwindigkeit des LKW₂ zu dem LKW₁. Danach berechnet es mit Hilfe der Methode *u_zeit()* die Überholzeit und setzt durch den Aufruf der Methode *clear_time()* den Ti-

¹¹ Wenn das Modul *notstopp* aktiv wird, setzt es die globale Variable *stop*=1. Damit wird erreicht, dass alle Prozesse beendet werden.

mer des Axsen-Boards auf Null. Mit Hilfe dieses Timers wird dann das Ende der Überholzeit erkannt. Dann setzt es den Sollwert des P-Reglers auf

$$\text{sollWert} = 2 \cdot s_{\text{seitlich}} + b = S_{\text{ueberhol}}$$

b: Breite des Modell-Lastwagens

S_{seitlich} : seitlicher Sicherheitsabstand (vgl. Bild 5-7)

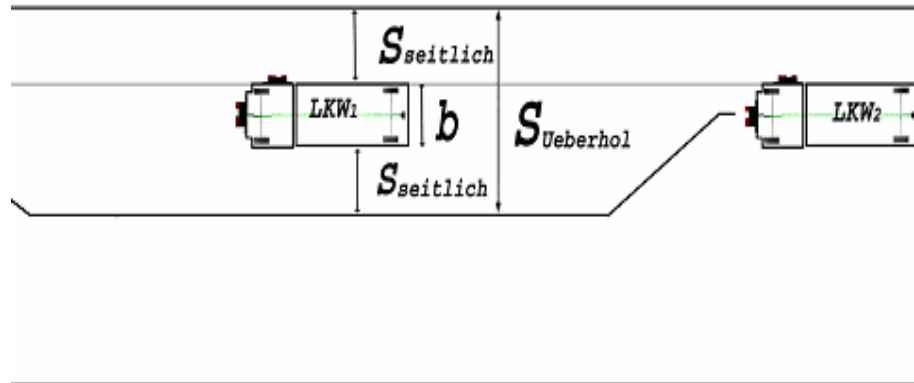


Bild 5-7: Spurwechsel zum Überholen.

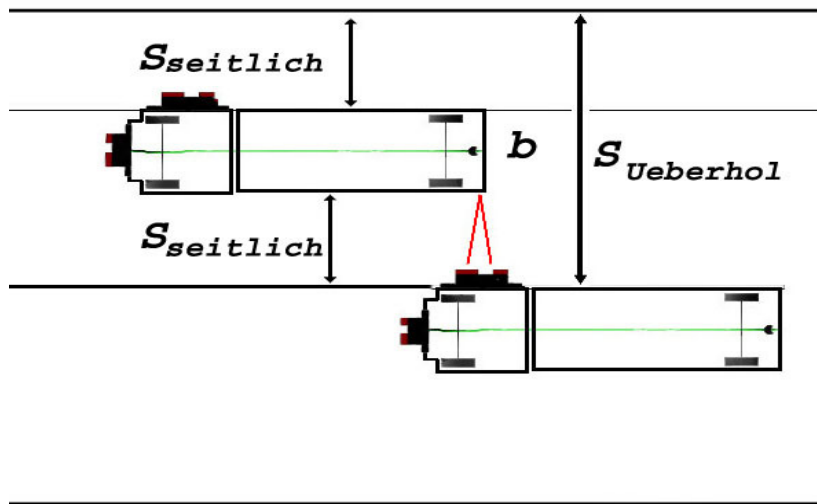
Der Sollwert wird durch das Modul *p-Regler* wieder auf den ursprünglichen Wert zurückgesetzt, wenn die Zeit des Timers, deren Wert vor dem Überholvorgang auf Null gesetzt wurde, größer als die Überholzeit ist. Das heißt, wenn die Überholzeit abgelaufen ist.

```
if ( regler.sollWert == UEBERHOL_ABSTAND && akt_time >= kont_1.ueberholZeit)
    regler.sollWert = SICHERHEITS_ABSTAND;
```

Während des Überholvorgangs kommt es zu einem plötzlichem Anstieg der Regeldifferenz von $e = 0$ auf

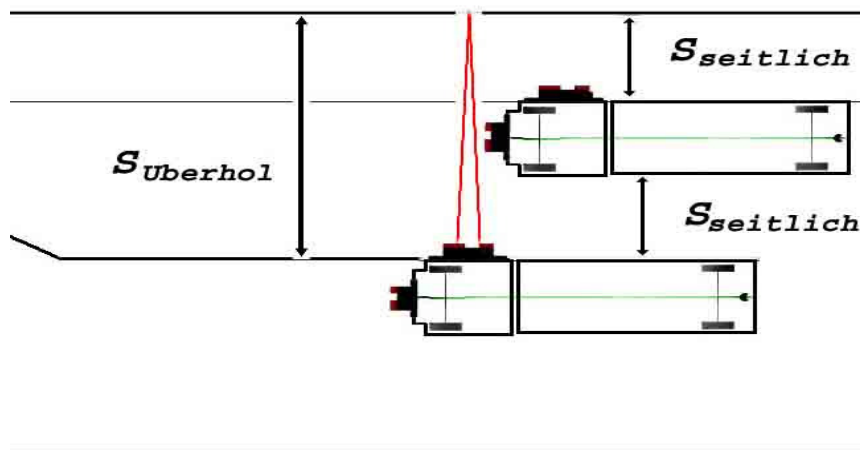
$$e = S_{\text{ueberhol}} - (s_{\text{seitlich}} + b) = 2 \cdot s_{\text{seitlich}} + b - (s_{\text{seitlich}} + b) = s_{\text{seitlich}}$$

wenn der LKW₂ den LKW₁ erreicht. Der Grund liegt darin, dass der Sensor am LKW₂ beim Erreichen des LKW₁ nicht mehr die Wand erfasst, sondern die Karosserie des LKW₁ (vgl. Bild 5-8).

Bild 5-8: Erfassung des LKW₁ durch den Sensor von LKW₂

Hier wurde der P-Regler so angepasst, dass bei einer plötzlichen Änderung der Regelgröße von $istWert = S_{Überhol}$ auf $istWert = S_{seitlich}$ der Regler den Sollwert auf den Wert $sollWert = s_{seitlich}$ setzt. Damit wird erreicht, dass die Regeldifferenz $e = 0$ bleibt und der LKW₂ mit Sicherheitsabstand $s_{seitlich}$ an LKW₁ vorbeifährt.

Überholt der LKW₂ den LKW₁, so erfasst der Sensor des LKW₂ wieder die Wand. Dabei fällt die Regelgröße von $istWert = S_{seitlich}$ auf $istWert = S_{Überhol}$. (vgl. Bild 5-9)

Bild 5-9: Sprung des Istwertes von $S_{seitlich}$ auf $S_{Überhol}$.

In dem Fall setzt der P-Regler den Sollwert wieder auf $sollWert = S_{überhol}$.

Neben den Modulen der Kontrollschichten K_0 und K_1 wurden noch zwei Module *entscheider* und *sensoren* implementiert. Das Modul *entscheider* ist für die Ausführung der von den Kontrollschichten gesetzten Befehle verantwortlich. Das Modul *entscheider* ist wie folgt implementiert.

```

1 void entscheidener(){
2   int stell_g = 0;
3   unsigned char v = 0;
4   int gang = 0;
5   while(!stop){
6     lcd_Ausgabe();
7     stell_g = *ent.lenkMotor;
8     fahrstufe = *ent.fahrMotor;
9     gang     = *ent.schaltMotor;
10    if(zustand_0 != NONE){
11      ent.lenk_Motor = & cont_0.lenk_contrlo_0;
12      ent.fahr_Motor = & cont_0.fahr_contrlo_0;
13      ent.schaltMotor = & k_0.schaltBefehl_0;
14    }
15    if(zustand_1 != NONE){
16      ent.lenk_Motor = & cont_1.lenk_control_1;
17      ent.fahr_Motor = & cont_1.fahr_control_1;
18    }
19    if (stell_g != *ent.lenkMotor)
20      lenk(*ent.lenkMotor);
21    if (fahrstufe != *ent.fahrMotor)
22      fahr(*ent.fahrMotor,1);
23    if (gang != *ent.schaltMotor)
24      schalt(*ent.schaltMotor);
25    process_defer();
26  }
27  process_kill( process_get_pid());
28}

```

In der ersten if-Abfrage der While-Schleife des Moduls *entscheider* wird geprüft, ob eines der Module der Kontrollschicht K_0 aktiv ist. Das ist der Fall, falls der aktuelle Zustand von K_0 nicht *NONE* ist (`if(zustand_0 != NONE)`). Ist der aktuelle Zustand von K_0 nicht *NONE*, werden die Befehle von K_0 übernommen (Zeile 11, 12, 13).

Bei der zweiten if-Abfrage werden die Befehle der Kontrollschicht K_1 übernommen, falls `Zustand_1 != NONE` ist. Hier können die Befehle von K_0 überschrieben werden. (Zeile 16, 17). Am Ende der While-Schleife werden dann die übernommenen Befehle ausgeführt¹² (Zeile 19, 24), und durch den Aufruf von *process_defer()*; (Zeile 25) ein

¹² Auf die Motoren wird nur dann zugegriffen, wenn die Signale sich geändert haben.

Prozesswechsel erzwungen. Wenn die While-Schleife beendet ist (*stop* = 0), beendet sich der Prozess *entscheider* selbst (Zeile 27).

Das Modul *sensoren* rechnet durch den Aufruf der Methoden *wand_Sensor()* und *bahn_Sensor()* die Sensorwerte in Entfernung um, und speichert das Ergebnis in der globalen Struktur.

```
struct Senser
{
    char wandAbstand;
    char hindernisAbstand;
    unsigned char wandSensorWert;
    unsigned char bahnSensorWert;
}abstand;
```

In der Main-Methode werden die Module *sensoren*, *kontroll_0*, *kontroll_1* und *entscheider* als einzelne Prozesse gestartet.

Zuerst wird der Prozess *sensoren* erzeugt. Entsteht bei der Prozesserzeugung ein Fehler, werden die anderen Prozesse nicht gestartet, da sonst der Modell-Lastwagen orientierungslos bleibt. War die Prozesserzeugung erfolgreich, wird als nächstes der Prozess *entscheider* erzeugt. Schlägt die Prozesserzeugung fehl, wird der Prozess *sensoren* auch beendet und die Main-Methode verlassen, ohne die Module *kontroll_0* und *kontroll_1* als Prozesse zu starten, denn nur der Prozess *entscheider* hat Zugriff auf die Motoren des Fahrzeugs. Folgender Programmcode zeigt die Erzeugung der Prozesse *sensoren* und *entscheider*

```
1  if((pid_Sen = process_start(sensoren, VERWEILZEIT)) > 19) {
2      lcd_setxy(0,0);
3      lcd_puts("Sensoren Prozess nicht erzeugt");//Fehler Meldung
4  }
5  else if ((pid_Ent = process_start
           (entscheider, VERWEILZEIT_ENTSCHEIDER)) > 19) {
6      lcd_setxy(0,0);
7      lcd_puts("Entscheider Prozess nicht erzeugt");
8      process_kill(pid_Sen);
9  }
```

Wenn der Rückgabewert der Methode *process_start* größer ist als 19 (Zeile 1), dann ist der Prozesserzeugung fehlgeschlagen, und die *elseif*-Anweisung (Zeile 5) wird nicht ausgeführt.

Nach erfolgreicher Erzeugung der Prozesse *sensoren* und *entscheider* wird das Modul *kontroll_0* erzeugt. Beim Fehlschlagen der Prozesserzeugung werden alle anderen ge-

startete Prozesse ebenfalls beendet, weil nur der Prozess *kontroll_0* den Modell-Lastwagen in Bewegung setzt. Ohne diesen Prozess fährt das Fahrzeug nicht. Zuletzt wird dann das Modul *kontroll_1* als Prozess gestartet. Wird dieser Prozess nicht erzeugt, bleiben die Verhaltensweisen der Kontrollschicht K_1 aus, aber das Fahrzeug ist schon in der Lage, autonom zu fahren und dabei Kollisionen zu vermeiden.

6. Experimentelle Ergebnisse

Bestimmung des Verstärkungsfaktors K_p des P-Reglers:

Für die Bahnkontrolle des Modell-Lastwagens kommt ein P-Regler zum Einsatz. Der P-Regler soll die Fahrt parallel zur Wand ermöglichen. Da das Fahrverhalten des Modell-Lastwagens nicht bekannt ist, kann nicht festgelegt werden, welcher Wert dem Proportionalitätsfaktor des P-Reglers K_p zugewiesen werden soll, damit das Fahrzeug möglichst parallel zur Wand fährt.

Der Proportionalitätsfaktor K_p wird im Folgenden experimentell ermittelt. Dabei wird das Verfahren von Ziegler/Nichols (vgl. Abschnitt 4.1) angewandt. Da der Programmspeicher des Axsen-Boards auf 32KB eingeschränkt ist, wird nur mit Char- und Integer Zahlen gearbeitet.

Zuerst wurde $K_p = 1$ zugewiesen. Das Modell-Fahrzeug wurde in einer Entfernung $x_{\max} = 80\text{cm}$ parallel zur Wand gestellt. Diese Entfernung ist der maximale Erfassungsbereich des Sensors. Der Sollwert (Sicherheitsabstand zur Wand) beträgt $\text{sollWert} = 40\text{cm}$. Das Modell-Fahrzeug wurde mit einer konstanten Geschwindigkeit $v = 37 \frac{\text{cm}}{\text{s}}$ im Gang 2 gefahren und das Fahrverhalten des Modell-Fahrzeugs beobachtet. Das Fahrzeug schwang periodisch um die Sollbahn. Es wurden dabei die einzelnen Amplituden dieser Schwankungen für den Zeitraum $0\text{s} \leq t \leq 30\text{s}$ gemessen. Analog wurde das Fahrverhalten des Fahrzeugs für $K_p = 2, \dots, 16$ durchgeführt.

Für $K_p = 1, \dots, 4$ reagiert der P-Regler so langsam, dass nach 25s immer noch eine Regelabweichung $e = \pm 10\text{cm}$ existiert (vgl. Bild 6-1).

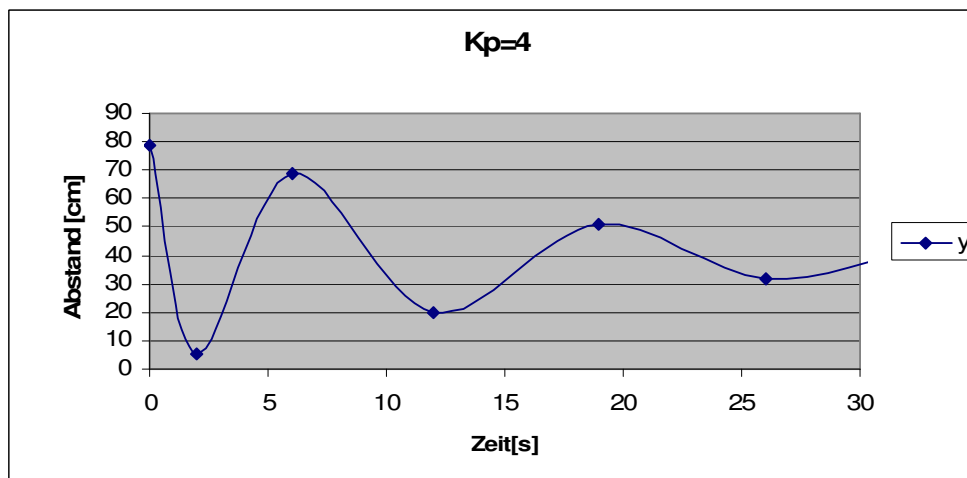
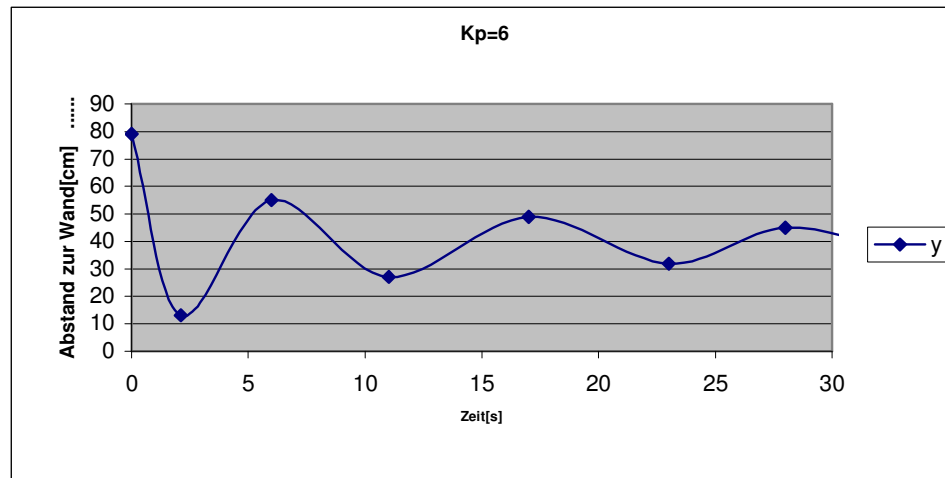
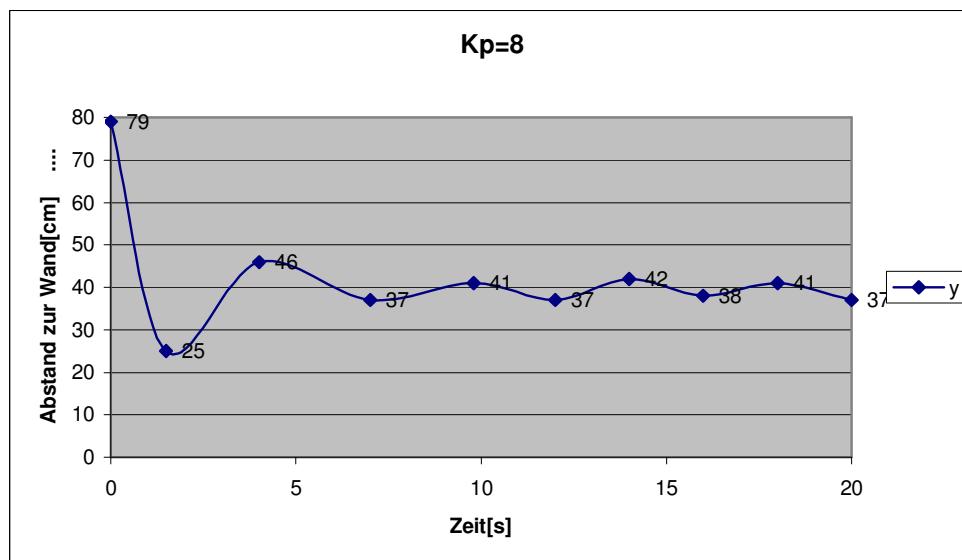


Bild 6-1: Bahnverlauf des Modell-Lastwagens bei $K_p = 4$.

Bei $K_p = 6$ wurde die Amplituden deutlich kleiner.

Bild 6-2: Bahnverlauf des Modell-Lastwagen bei $K_p = 6$.

Bei $K_p = 8$ stabilisiert sich das Fahren entlang der Wand. Die Regelabweichung ist so klein, dass die Schwankungen beim Beobachten des Fahrzeugs während der Fahrt nicht wahrgenommen werden.

Bild 6-3: Bahnverlauf des Modell-Lastwagen bei $K_p = 8$.

Bei $K_p = 12$ ist die Amplitude der Bahnverlauf des Modell-Lastwagens noch kleiner und die Periodendauer der Schwankung des Fahrzeugs hat sich auf $t \approx 400ms$ s verkleinert. (vgl. Bild 6-4).

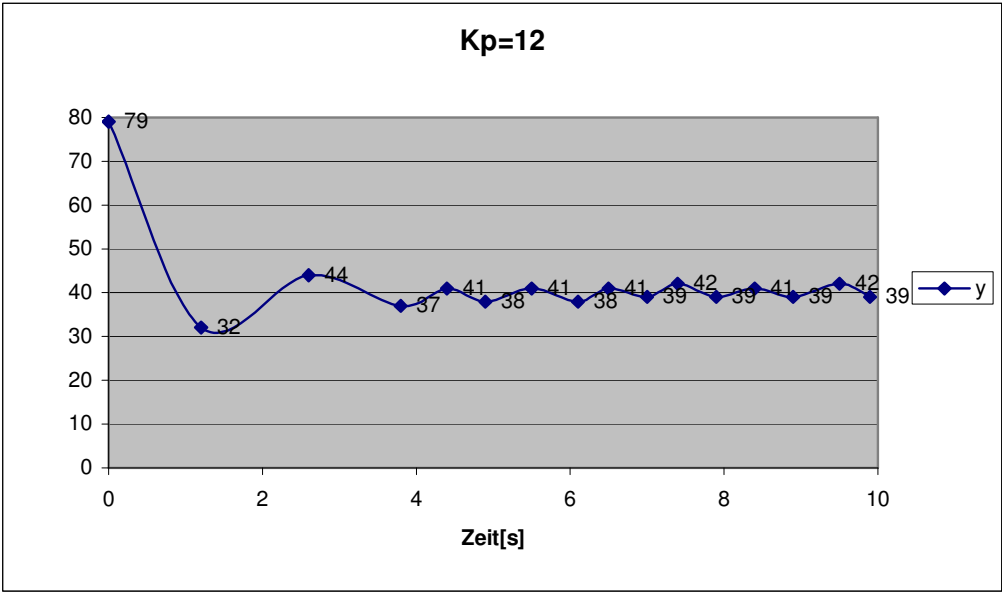


Bild 6-4: Bahnverlauf des Modell-Lastwagens bei $K_p = 12$.

Bei $K_p > 16$ ist die Regelstärke so groß, dass die Stabilitätsgrenze überschritten wird. In dem Fall liefert der Regler schon bei minimaler Regelabweichung $e = 1\text{cm}$ eine zu große Stellgröße y und somit reagiert zu stark auf die Regelabweichung. Als Folge erreicht das Fahrzeug mit einem zu großen Winkel die Sollbahn. Demzufolge entfernt sich das Fahrzeug von der Sollbahn, weil es in Bewegung ist. Somit werden die Abweichungen auch größer. Die Periodendauer der Schwankung sinkt auf etwa 200ms (vgl. Bild 6-5), und die Vorderräder des Fahrzeugs anfangen zu zittern.

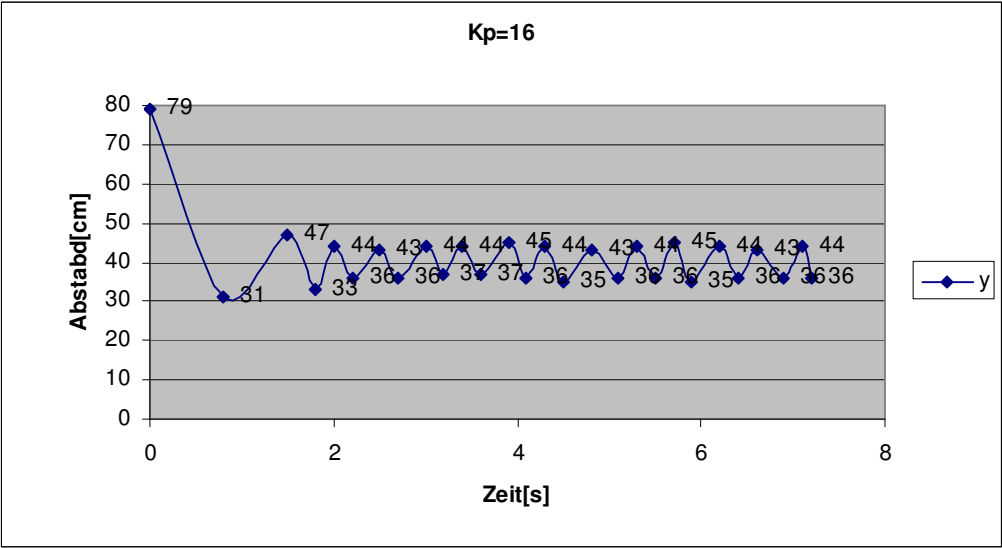


Bild 6-5: Bahnverlauf des Modell-Lastwagens bei $K_p=16$.

Nach Ziegel wird das System bei $K_p = \frac{K_{p \max}}{2}$ mit minimaler Regelabweichung geregelt.

Demzufolge ergibt sich ein Verstärkungsfaktor $K_p = \frac{16}{2} = 8$. Die experimentellen

Beobachtungen haben gezeigt, dass für $K_p = 8$ der Modell-Lastwagen so stabil entlang der Wand fährt, dass sich die bleibende Regelabweichung nicht bemerkbar macht.

Als nächster Schritt wurde das Fahrverhalten des Fahrzeugs für $K_p=8$ näher untersucht. Das Fahrzeug braucht die Zeit $t \approx 7s$ um seine Fahrt entlang der Wand zu stabilisieren, falls das Fahrzeug beim Start etwa 80cm entfernt von der Wand ist. Es bleibt eine durchschnittlich bleibende Regelabweichung $e \approx \pm 2cm$. Das heißt, auf einer Seite der Sollbahn mehr und auf der anderen Seite der Sollbahn weniger. Diese nicht symmetrische Regelabweichung um die Sollbahn kommt zustande, weil das Lenkgetriebe des Modell-Lastwagen nicht exakt auf $\alpha = 0$ eingestellt werden kann.

Folgende Tabelle zeigt die Regelstärke y in Abhängigkeit von Regeldifferenz e . Der Sollwert hat den Wert 40cm und der Proportionalitätsfaktor $K_p = 8$;

Für diesen konkreten Fall ist die Gleichung der P-Regler:

$$y(t) = 1500 + 8 \cdot e(t)$$

wobei $e(t)$ die Regeldifferenz $e = (\text{sollWert} - \text{istWert})$ zum Zeitpunkt t ist. Die folgende Tabelle zeigt die Werte von Stellgröße y abhängig von der Regeldifferenz.

Sollwert/cm	Istwert/cm	Regeldifferenz/cm	Stellgröße/ μs
40	0	40	1820
	10	30	1740
	20	20	1660
	30	10	1580
	36	4	1532
	37	3	1524
	38	2	1516
	39	1	1508
	40	0	1500
	41	-1	1492
	42	-2	1484
	43	-3	1476
	50	-10	1420
	60	-20	1340
	70	-30	1260
	80	-40	1180

Tabelle 6-1: Stellgröße y abhängig von der Regeldifferenz.

Wie schon im Abschnitt 3.1 erwähnt, schlägt das Lenkgetriebe bei einer Impulsbreite $b = 1845$ voll nach links und bei der Impulsbreite $b = 1155$ voll nach rechts aus. Der Erfassungsbereich des eingesetzten Sharp-Sensors ist aber auf 10 bis 80cm eingeschränkt. In Folge dessen wird (vgl. Tabelle 6-1) das Fahrzeug nicht voll einschlagen, auch wenn das Fahrzeug sehr nahe an der Wand ist. Um eine schnelle Reaktion des Fahrzeugs zu erreichen, falls das Fahrzeug zu weit weg von der Wand oder sehr nah an der Wand ist, wurde der P-Regler durch zwei Programmzeile wie folgt angepasst:

```
if (Istwert >= 80) stellgroesse = 1155;
else if(Istwert <= 10) stellgroesse = 1845;
```

Wie aus der Tabelle 6-1 zu lesen ist, wird bei einer Regeldifferenz von $e = \pm 2\text{cm}$ die Stellgröße $y = 1516$ bzw. $y = 1484$ angenommen. Nach experimenteller Beobachtung reagiert der Regler bei $y \geq 1516$ bzw. $y \leq 1484$ stark genug, um das Fahrzeug schnell wieder auf die Sollbahn zu lenken, sodass die Regelabweichung 2cm nicht überschreitet.

In Rahmen dieser Arbeit kann die Abweichung vernachlässigt werden, da diese Regelabweichung sich beim Fahren des Fahrzeugs nicht bemerkbar macht und die Sicherheit des Fahrzeugs in keinem Fall gefährdet ist. Infolgedessen wurde auf das Hinzufügen eines zusätzlichen I-Anteils im Regler verzichtet. In einem weiteren Versuch wurde der Modell-Lastwagen mit $v = 79 \frac{\text{cm}}{\text{s}}$ und $v = 90 \frac{\text{cm}}{\text{s}}$ gefahren. Dem Regelverstärker des P-Reglers K_p wurde der Wert 8 zugewiesen und das Fahrverhalten des Modell-Lastwagens entlang der Wand beobachtet. Der P-Regler reagiert auch bei diesen Geschwindigkeiten optimal. Es wurde aber festgestellt, dass bei Geschwindigkeiten $v \geq 90 \frac{\text{m}}{\text{s}}$ das Fahrzeug gegen die Wand fährt, falls es zu weit weg von der Wand ist und stark in Richtung der Wand lenkt. Falls das Fahrzeug mit der Geschwindigkeit $v \geq 90 \frac{\text{m}}{\text{s}}$ betrieben wird, muss entsprechend der Abstand zur Wand vergrößert werden.

Ein weiterer Störfaktor für das geregelte Fahren entlang der Wand waren die Türen, die sich auf beiden Seiten des Korridors befinden. Die Türen sind etwa 10cm tiefer als die Wandebene (vgl. Bild 6-6).

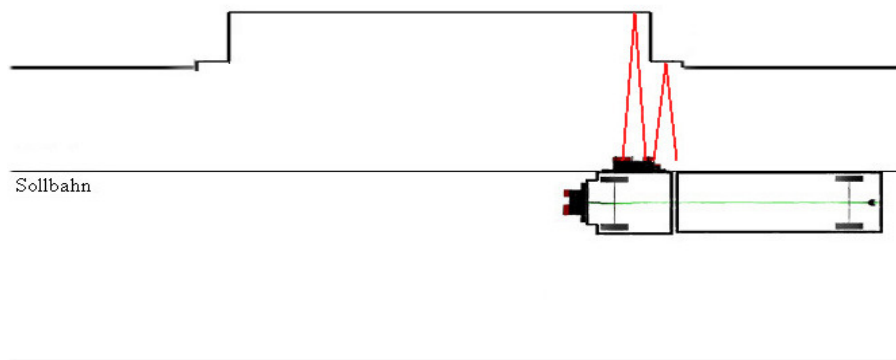


Bild 6-6: Plötzliche Änderung der Regeldifferenz beim Erreichen des Türbereichs.

Das Fahrzeug braucht etwa 7s, um seine Fahrt mit $v = 37 \frac{cm}{s} v$ entlang der Wand zu stabilisieren. Tauchen die Türen auf, findet plötzlich eine Regeldifferenz $e = -10cm$ statt.

Der Regler führt eine Korrektur durch und regelt das Modell-Fahrzeug in Richtung der Tür. Beim Verlassen des Türbereichs findet wieder eine Abweichung $e = 10$ von der Sollbahn statt. Der Regler steuert wieder gegen. Somit wird das parallele Fahren entlang der Wand durch die Türbereiche gestört. Nach Beobachtungen braucht das Fahrzeug etwa 4s, um die Schwingungen abzubauen und seine Fahrt entlang der Wand wieder zu stabilisieren. Da die Türen entlang der Wand in unterschiedlichen Abständen immer wieder auftauchen, kann das Fahrzeug nicht wie gewünscht parallel zur Wand fahren.

Um diese Störung zu beseitigen, wurde der P-Regler so angepasst, dass bei einer plötzliche Änderung der Regelgröße $e_{istWert} = 10$ der Sollwert um $solWert = sollWert + 10$ erhöht wird. Dadurch bleibt die Regeldifferenz gleich und der Regler wird auf die Änderung der Regelgröße nicht reagieren. Aufgrund der Messabweichung des Distanzsensors, die durch plötzliche Abstandsänderung zustande kommt, entsteht eine Regelabweichung ($-7 \geq e_{istWert} \geq -14$). Der Intervall $[-7, -14]$ wurde experimentell festgelegt. In diesem Intervall liefern die Tests zufriedenstellende Ergebnisse. (Fahrzeug fährt gerade, ohne in Richtung Wand zu lenken). Analog wird bei plötzlicher Änderung der Regelgröße ($7 \leq e_{istWert} \leq 14$) der Sollwert um 10 verringert. Dieser Fall kommt vor, wenn das Fahrzeug den Türbereich verlassen hat.

Die Messabweichung des Distanzsensors ist bei einer plötzlichen Abstandsänderung nicht konstant. Die Messabweichungen liegen zum Teil außerhalb des festgelegten Intervalls. Sie sind so zerstreut, dass daraus kein verlässlicher Mittelwert gebildet werden kann. Aus diesem Grund kann das gewünschte Verhaltens des Fahrzeugs in Türbereichen nicht garantiert werden.

Hat das Fahrzeug seine Fahrt entlang der Wand vor Erreichen des Türbereichs nicht stabilisiert, bzw. fährt das Fahrzeug mit einem bestimmten Winkel α entlang der Wand, werden die Türen nicht erkannt, da in diesem Fall keine sprunghafte Änderung der Regelgröße erfolgt (vgl. Bild 6-7).

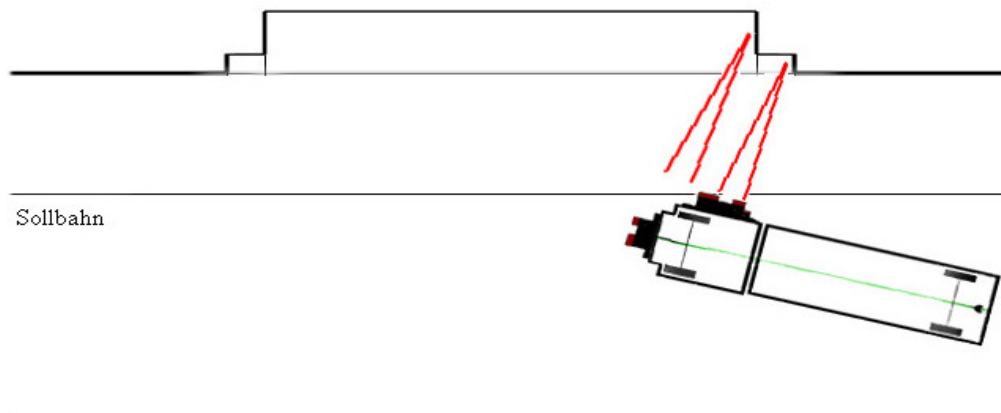


Bild 6-7: Erfassungspunkte des Sensors falls der LKW schräg zur Wand ist.

Das Fahrzeug muss vor Erreichen des Türbereichs parallel zur Wand fahren. Das ermöglicht eine schwingungsfreie Fahrt in einem Korridor mit Türen.

Der Überholvorgang

Für den Test des Überholvorgangs wurde das Fahrzeug LKW₁ mit der Geschwindigkeit $v_1 = 37 \frac{cm}{s}$ gefahren. Das Fahrzeug LKW₂ folgt LKW₁ mit der Geschwindigkeit $v_2 = 79 \frac{cm}{s}$. Ist der Abstand a zwischen LKW₁ und LKW₂ $a \leq 120cm$, soll der LKW₂ zuerst seine Geschwindigkeit relativ zu LKW₁ berechnen. Das geschieht, indem der LKW₂ nach der Zeit $\Delta t = 1s$ den Abstand Δs berechnet. Mit Hilfe Δt und Δs soll dann die Überholstrecke und Überholzeit wie im Abschnitt 4.3 berechnet werden. Dann soll der LKW₂ zum Überholen die Spur wechseln, die Überholstrecke fahren und sich wieder auf seine alte Fahrbahn einordnen. Die Spurwechsel erfolgen durch entsprechende Änderung des Sollwerts (vgl. Bild 5-7 im Abschnitt 5.2).

Nach dem Test wurde festgestellt, dass der Sensor des LKW₂ während des Spurwechsels den hinteren Teil des LKW₁ erfasst. Dadurch entsteht eine plötzliche Änderung der Regelgröße $e_{istWert}$, die aufgrund der Bewegungen der beiden Fahrzeuge immer unterschiedlich ausfällt, sodass diese nicht durch entsprechende Änderung des Sollwerts kompensiert werden konnte. Das Bild (6-8) zeigt die Fahrtrichtung des Fahrzeugs nach dem Setzen des Sollwert auf $sollWert = S_{ueberhol}$.

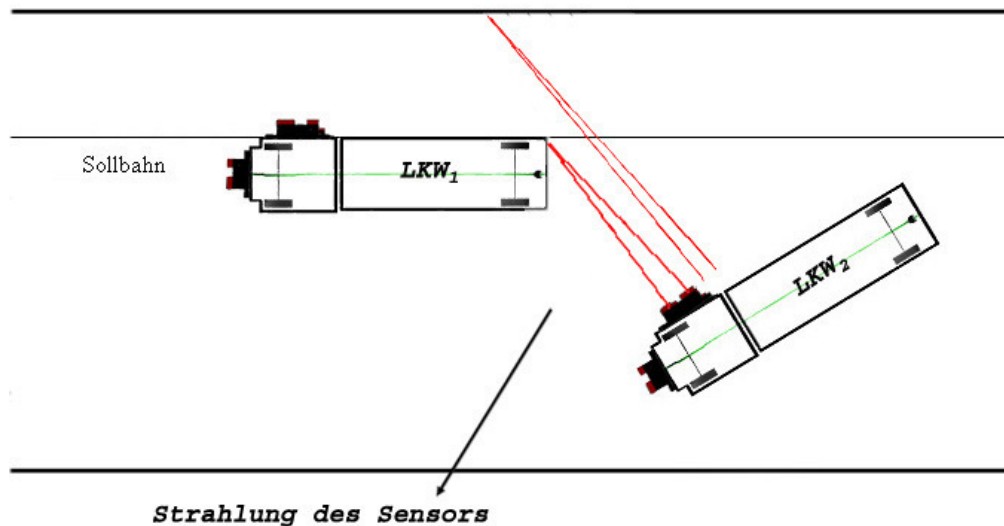


Bild 6-8: Erfassung des hinteren Teils des LKW₁ durch den Sensor von LKW₂.

Der Regler regelt entgegen und fährt gegen die Wand (vgl. Bild 6-8). Damit der Sensor des LKW₂ während des Spurwechsels nicht den LKW₁ erfasst, wurde der Sensor für die Erfassung der Wand 2cm höher als die Höhe des LKWs montiert. Somit konnte der

Überholvorgang erfolgreich durchgeführt werden, ohne dass Störungen beim Erreichen des LKW_1 eintraten.

7. Zusammenfassung und Ausblick

7.1. Zusammenfassung

Die Entwicklung von autonomen Fahrzeugen gewinnt für die Industrie immer stärker an Bedeutung. Dies liegt zum einen an den erfolgreichen Forschungen, die in den letzten Jahren auf diesem Gebiet durchgeführt wurden und zum anderen liegt es an den vielfältigen Einsatzmöglichkeiten für autonome Fahrzeuge. Diese Arbeit leistet daher einen Beitrag zur Entwicklung solcher Fahrzeuge.

Sie hat die Entwicklung eines autonomen Fahrzeugs zum Ziel, das selbständig entlang der Wand fährt, dabei Kollisionen mit Hindernissen vermeidet und in der Lage ist, ein Überholmanöver durchzuführen.

Zunächst wurden in dieser Arbeit die Grundlagen und der Aufbau von autonomen Fahrzeugen dargelegt. Dabei wurde auf das Zusammenspiel zwischen der Sensorik/Aktorik sowie Hard- und Software eines autonomen Fahrzeugs eingegangen. Der Schwerpunkt wurde auf die Sensorik und das Steuerungssystem der autonomen Fahrzeuge gelegt.

Als nächstes wurde die Rolle der Regelungstechnik in Bezug auf autonome Fahrzeuge erläutert. Dabei wurden die technischen Grundlagen sowie die wichtigsten Regeltypen beschrieben.

Im Kapitel 4 wurde dann ein Lösungskonzept entworfen. Das Konzept sieht die Nutzung der Methoden der Regelungstechnik für die Steuerung der Fahrtrichtung vor. Nach dem Konzept ermöglicht ein P-Regler eine parallele Fahrt zur Wand. Ein Steuerungssystem nach der Subsumptions-Architektur von Brooks sorgt für ein reaktives Verhalten des Modell-Lastwagens. So kann der Modell-Lastwagen ein Überholmanöver durchführen und gleichzeitig auf Hindernisse reagieren. Das Modell ist auch für Erweiterungen des Systems gut geeignet.

Die Reaktion des Modell-Lastwagens auf unmittelbare Hindernisse oder Sensordefekte wurde auf das Ausschalten des Fahrmotors eingeschränkt. Die Realisierung eines komplexen Verhaltens hätte die Rahmen dieser Arbeit gesprengt.

Bei einem Überholvorgang schert der Modell-Lastwagen zum Überholen aus, fährt die Überholstrecke geradeaus und schert wieder ein. Dabei werden die Sicherheitsabstände eingehalten. Im letzten Schritt wurde das Lösungskonzept implementiert. Im Rahmen von Experimenten wurde der Parameter für den P-Regler ermittelt und das gewünschte Verhalten des Modell-Lastwagens bestätigt. Als Ergebnis ist ein autonomes Modellfahrzeug mit folgenden Fähigkeiten entstanden:

- ◆ Geregelttes Fahren entlang der Wand: Der Modell-Lastwagen kann bei Einhalten eines Sicherheitsabstand w die Wand verfolgen. Der eingesetzte P-Regler zeigt eine bleibende Regelabweichung $e \in [-2cm, 2cm]$.
- ◆ Reaktion in Echtzeit auf Kollisionsgefahren und Sensordefekte.
- ◆ Die Durchführung eines Überholvorgangs.

7.2. Ausblick

Zukünftige Arbeiten könnten darauf abzielen, dass der Modell-Lastwagen mit weiteren Störfaktoren zurecht kommt und stabil seine Fahrspur verfolgt.

Im Rahmen der Experimente wurde z.B. festgestellt, dass der LKW nicht stabil entlang einer Wand fährt, wenn die Wand große Unebenheiten aufweist.

Beim Durchführen der Versuche in einem Korridor stellten die Türen solche Unebenheiten dar. Sie sind entlang des Korridors in unterschiedlichen Abständen platziert. Die Türbereiche sind etwa 10cm tiefer als die Wandebene. Beim Erreichen einer Tür entsteht eine Regeldifferenz von ± 10 cm. Das Fahrzeug reagiert auf diese Regelabweichung und lenkt in Richtung der Wand. Am Ende eines Türbereichs entsteht eine Regelabweichung von $e = 10$ cm. Das Fahrzeug lenkt daher von der Wand weg. Dieses Verhalten stört das wandparallele Fahren. Das Fahrzeug braucht die Zeit $t \approx 4s$, um seine Fahrt wieder zu stabilisieren.

Mit der im Kapitel 6 eingesetzten Lösung konnte die Störung nicht beseitigt werden. Grund hier für war, dass der Sensor bei plötzlichen Abstandsänderungen keine zuverlässigen Daten sendet.

Die Realisierung folgender Idee, die aus zeitlichen Gründen im Rahmen dieser Arbeit nicht eingesetzt werden konnte, könnte die oben genannte Störung beseitigen.

Falls das Fahrzeug eine Tür in der Zeit t_1 passiert, soll festgestellt werden, wie das Fahrzeug in der Zeit t_0 (vor dem Erreichen der Tür) relativ zur Wand stand. Stand das Fahrzeug quasi parallel zur Wand, dann soll das Fahrzeug auf die Regelabweichung, die durch das Auftauchen der Tür entstand, nicht reagieren und geradeaus weiterfahren, sonst soll das Fahrzeug auf die Regelabweichung normal reagieren.

Im Rahmen dieser Arbeit wurde die Entfernung des Modell-Lastwagens zur Wand nur mit Hilfe der auf den rechten Kotflügel montierten Distanzsensors gemessen. Daher war es nicht möglich, die Lage des Modell-Lastwagens relativ zur Wand zu bestimmen.

Für die Bestimmung des Winkels zwischen dem Modell-Lastwagen und der Wand soll ein zusätzlicher Sensor wie im Bild 7-1 montiert werden.

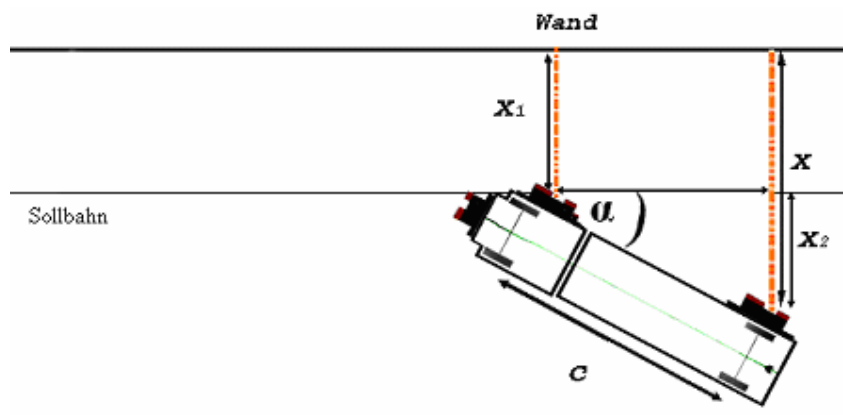


Bild 7-1: Berechnung der Winkel des Fahrzeuges relativ zur Wand mit der Hilfe zwei Distanzsensoren.

Wie das Bild (7-1) zeigt, messen die beiden Sensoren die Entfernung des Fahrzeugs relativ zur Wand. Sind die gemessenen Anstände der beiden Sensoren gleich, steht das Fahrzeug parallel zur Wand. Falls das nicht der Fall ist, dann steht das Fahrzeug schräg zur Wand. Die Winkel α zwischen dem Fahrzeug und der Wand berechnet sich dann wie folgt:

$$\alpha = \arcsin\left(\frac{x_2}{c}\right)$$

x_1 : Der Wert des vorderen Sensors

x_2 : Der Wert des hinteren Sensors

x_2 : $(x - x_1)$

c : Der Abstand zwischen den Sensoren

α : Der Winkel zwischen dem Modell-Lastwagen und der Wand

Im Rahmen dieser Arbeit wurde für die Erfassung der Hindernisse der Sensor GP2D15 eingesetzt. Wie schon im Abschnitt 5.1 erwähnt, liegt der Erfassungsbereich dieses Sensors im Intervall [30cm, 100cm]. Dieser Sensor ist für die Erkennung der Hindernisse nicht gut geeignet, da die minimale Entfernung, die der Sensor misst, beträgt 30cm, auch wenn das Hindernis beispielsweise 15cm entfernt ist. Demzufolge musste das Modul *notstopp* aktiv werden und das Fahrzeug stoppen, wenn der Sensor eine Entfernung von 30cm meldet, weil die tatsächliche Entfernung des Hindernisses noch kleiner sein könnte. Das hatte zur Folge, dass der Modell-Lastwagen gestoppt wurde, auch wenn dessen Abstand $20 \leq s \leq 30\text{cm}$ relativ zur Wand war.

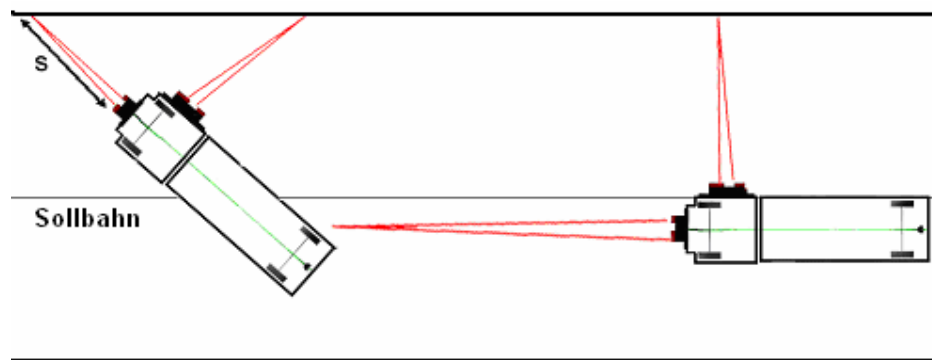


Bild 7-2: Spurewechsel nach dem Überholen.

Beim Einscheren verkleinert sich der Abstand des Fahrzeugs relativ zur Wand auf unter 30cm (vgl. Bild 7-2). In dem Fall stoppt das Fahrzeug, obwohl es genügend Raum gibt, wieder in Richtung der Sollbahn zu lenken, ohne mit der Wand zu kollidieren.

Bei der Erweiterung dieser Arbeit kann der Sensor GP2D15 für die Erkennung anderer Fahrzeuge eingesetzt werden. Es soll ein zusätzlicher Distanzsensor im vorderen Stoßstange des Modell-Lastwagens montiert werden, der im Intervall [4cm, 30cm] arbeitet, und Hindernisse, die sich in diesem Intervall befinden, meldet. Damit wird erreicht, dass wenn ein Hindernis sich unmittelbar vor dem Modell-Lastwagen befindet das Fahrzeug stoppt, sonst nicht.

LITERATURVERZEICHNIS

- [ARKI-98] Arkin, R. C.: Behaviour-Based Robotics. MIT Press, Cambridge, USA, 1998
- [BOET-02] Böttle P., Boy G., Clausing H.: Elektrische Meß- und Regelungstechnik. Die Meisterprüfung. Würzburg: Vogel, 2002
- [BROO-89] R. A. Brooks: A Robust Layered Control System For A Mobile Robot. IEEE Journal of Robotics and Automation, RA-2, 04.1989
- [CONR-05] www.conrad.de, Zugriffsdatum: 27.11.2005
- [DAIM-06] DaimlerChrysler Hightech Report: <http://www.daimlerchrysler.de>, Zugriffsdatum: 2.01.2006
- [FHBR-05] Labor für Künstliche Intelligenz an der FHB: <http://ots.fh-brandenburg.de/index.php>, Zugriffsdatum: 4.12.2005
- [GEER-04] Peter Geering: Regelungstechnik. Mathematische Grundlagen, Entwurfsmethoden, Beispiele. Springer-Verlag Berlin Heidelberg 2004
- [GUTS-94] Ralf Gutsche: Fahrlose Transportsysteme, Automatische Bahnplanung in dynamischen Umgebungen. Braunschweig/Wiesbaden, 1994
- [HAND-05] Das Handbuch des Aksen-Boards: http://ots.fhbrandenburg.de/downloads/aksen/CD1.04_B00.965_FW1.4_FL0.3_H1.04/Handbuch/Handbuch.pdf
- [HAUT-05] M.Hauth. autonome Roboter: <http://tech-www.informatik.uni-hamburg.de/lehre/ss2005/proseminar/robotik/pdf/2005-06-06.pdf>
Zugriffsdatum: 20.11.2005
- [HERM-01] M. Hermann: Numerische Mathematik. München: Oldenburg, 2001
- [HOPP-92] Hoppen P.: Mobile Roboter. Echtzeitnavigation in bekannter und unbekannter Umgebung. Mannheim, 1992.
- [HÜBN-04] B. Hübener: Remotesteuerung eines Modellbaulastwagens -Schaffung einer Erprobungsplattform für Experimente mit Fahrerassistenzsystemen. Hamburg, 12.2004
- [MAND-05] Thomas Mandl: <http://www.uni-hildesheim.de/~mandl/Lehre/ubi/Fahrerassistenzsysteme.pdf>, Zugriffsdatum: 28.01.2006
- [MEND-99] Mende, R.: Radarsysteme zur automatischen Abstandsregelung in Automobilen. Doktorarbeit, Fakultät für Maschinenbau und Elektrotechnik, Technische Universität Carolo-Wilhelmina, Braunschweig, 1999.
-

-
- [PELL-03] Pellkofer M.: Verhaltensentscheidung für autonome Fahrzeuge mit Blickrichtungssteuerung. Doktorarbeit, Fakultät für Luft- und Raumfahrttechnik, Universität der Bundeswehr, München, 2003
- [SCHR-90] E. Schröder: Signalverarbeitung. Numerische Verarbeitung digitaler Signale. München, Carl Hanser Verlag, 1990
- [SCHU-05] Prof. Dr.-Ing. Schumacher W., Prof. em. Dr.-Ing. Leonhard W.: Grundlagen der Regelungstechnik. Technischen-Universität Braunschweig. <http://www.ifr.ing.tu-bs.de/lehre/vorlesungen> Zugriffsdatum: 26.11.2005
- [SOIK-98] Dipl.-Ing. Martin Soika und Landhut: Meß-, Steuerungs- und Regelungstechnik. Robustheit autonomer mobile Roboter gegenüber Sensordefekten. VDI Verlag GmbH, Düsseldorf 1998
- [WEDI-01] Wedico 2001 WEDICO: Wedico Bauanleitung Elektrische Anlage, Art.-Nr. 782. 17.04.2001
- [WEDI-03] Wedico 2003 WEDICO: Wedico Bauanleitung Fahrtregler, Art.-Nr. 143 / 191.10.09.2003
- [SRF0-05] Ultra sonic range finder: <http://www.robot-electronics.co.uk/shop/UltrasonicRangers1999.htm>, Zugriffsdatum: 13.11.2005
-

Anhang

Datei LKW_2.h

```

//Standard-Include-Files
#include <stdio.h>
#include <regc515c.h>
//Aksen-Board Bibliothek
#include <stub.h>
#include <math.h>
//Servo- Anschlüsse:
// S0 Lenkung
// S1 Fahrtregler
// S2 Getriebe
#define LENK_SERVO 0
#define FAHR_SERVO 1
#define SCHALT_SERVO 2

// Die Werte sind durch Experiments festgelegt
#define LENK_RECHTS_VOLL 1155 // lenke voll nach rechts
#define LENK_GERADE 1500 //1487 lenke grade
#define LENK_LINKS_VOLL 1845 //lenke voll nach links
#define FAHR_KONSTANTE 1500
#define FAHRSTUFE_MAX 45 //maximale Fahrstufe

#define INITIAL_ZEIT 7000 /* Die Zeit, die der Fahrmotor braucht, um
                             fahrbereit zu werden.*/
#define STABILITAETSZEIT 5000 /* Die Stabilisierung der Fahrt entlang der
                             Wand nach dem Überholvorgang */
#define UEBERHOL_STRECKE 180 /* 2 * Sicherheitsabstand + 2 * Länge des
                             Modell-Lastwagen */
#define SICHERHEITS_ABSTAND 40 // Seitliche Sicherheitsabstand zur Wand
#define UEBERHOL_ABSTAND 40 // Seitliche Sicherheitsabstand beim Überholen
/* (2 * SICHERHEITS_ABSTAND + Länge
   des Modell-Lastwagen) */

#define MIN_ABSTAND 20
#define SENSORWERT_MAX 196 // Wertebereich der Sensorwerte [29, 196]
#define SENSORWERT_MIN 29
#define SICHERHEITS_ZONE 11 /* Der Abstand, bei dessen Unterschreitung
                             das Modell-Lastwagen gestoppt wird. */

#define REGEL_DIFFERENZ_MAX 40

// Zustände
#define NONE 0
#define NOTSTOP 1
#define FAHREN 2
#define WANDVERFOLGUNG 3
#define REGLER 4
#define UEBERHOLEN 5

//Globale Variablen
char vLKW ; // Geschwindigkeit[cm/s] vom LKW
unsigned char stop =0;
// ID der Prozesse,
unsigned char pid_Kontroll_0=25, pid_Kontroll_1=25, pid_Ent=25, pid_Sen=25;

```

```

//Parameter der P-Regler
struct PRegler
{
    char kp;
    char sollWert;
    char *istWert;    // Zeiger auf wandAbstand
}regler;

//variablen der Kontrollschicht K0
struct Kontroll_0
{
    unsigned char zustand_0;
    int lenkBefehl_0;
    char fahrBefehl_0;
    unsigned char schaltBefehl_0;
} k_0;

//variablen des Moduls sensor
struct Senser
{
    char wandAbstand;           // Abstand zur Wand
    char hindernisAbstand;     // Hindernisabstand
}abstand;

//variablen der Kontrollschicht K1
struct Kontroll_1
{
    unsigned char zustand_1;
    char *soll_Wert;           // Zeiger auf sollWert.
    int lenkBefehl_1;
    char fahrBefehl_1;
    long ueberholZeit ;
    char ueberhol ;
} k_1;

struct Entscheider
{
    int *fahrMotor;
    int *lenkMotor;
    int *schaltMotor;
} ent;

/*****
* fahr(..); steuert den Fahrregler des Modell-Lastwagen.      *
* IN --> Geschwindigkeit[cm/s], Fahrriichtung                *
*      (1 --> vorwaerts, (-1) --> rueckwaerts).                *
* OUT --> void                                                *
*****/
void fahr(unsigned char geschwindigkeit, signed char richtung);

/*****
* lenk(..); steuert das Lenkgetriebe des

```

```

*      Modell-Lastwagen.
* IN --> Impulsbereite
* OUT --> void
*****/
void lenk(unsigned int lenkRichtung);

/*****/
* schalt(..); steuert das Schaltgetriebe des
*      Modell-Lastwagen.
* IN --> Gang [1,2,3]
* OUT --> void
*****/
void schalt(unsigned char gang);

/*****/
* entscheidener(); führt die Befehle der einzelnen
*      Kontrollschichten aus.
* IN --> void
* OUT --> void
*****/
void entscheidener();

/*****/
* kontroll_1(); Abhaegig von den Zustand des
*      Kontrollschichts K1 aktiviert die einzelnen
*      Modulen
* IN --> void
* OUT --> void
*****/
void kontroll_1();

/*****/
* wand_Verfolgung(); Abhängig von dem Hindernisabstand
*      veraendert den Zustand des K1
*      entsprechend
* IN --> void
* OUT --> void
*****/
void wandverfolgung();

/*****/
* ueberholen(); vergroeuert denn Anstand zur Wand, um
*      den Ueberholen zu ermoeöglichen
* IN --> void
* OUT --> void
*****/
void ueberholen();

/*****/
* p_Regler(); berechnet das Servosignal für das
*      Lenkgetriebe abhaengig von der
*      Regeldifferenz
* IN --> void
* OUT --> void
*****/
void p_Regler();

/*****/

```

```

* kontroll_1(); Abhaengig von dem Hindernisabstand          *
*      veraendert den Zustand des K0 und                  *
*      aktiviert anschliessend die einzelnen              *
*      Modulen                                           *
* IN --> void                                             *
* OUT --> void                                           *
*****/
void kontroll_0();

/*****
* init_Fahrmotor(); Initailisiert den Fahrmotor des LKWs  *
* IN --> void                                             *
* OUT --> void                                           *
*****/
void init_Fahrmotor();

/*****
* fahren(); setzt das LKW in Bewegung mit der Geschwindigkeit *
*      vLKW.                                             *
* IN --> void                                             *
* OUT --> void                                           *
*****/
void fahren();

/*****
* notstopp(); stoppt das LKW.                             *
*      vLKW2.                                           *
* IN --> void                                             *
* OUT --> void                                           *
*****/
void notstopp();

/*****
* abstand_Sensoren(); Stellt die durch den Sensoreng emessene *
*      Entfernung fuer die anderen Modulen zu Verfuegung *
* IN --> void                                             *
* OUT --> void                                           *
*****/
void sensoren();

/*****
* wand_Sensor(..); wandelt die Werte des Sensors GP2D12   *
*      in CM um                                         *
* IN --> Sensorwert                                     *
* OUT --> Entfernung in [cm]                             *
*****/
unsigned char wand_Sensor(unsigned char analog);

/*****
* bahn_Sensor(..); wandelt die Werte des Sensors GP2D15   *
*      in CM um                                         *
* IN --> Sensorwert                                     *
* OUT --> Entfernung in [cm]                             *
*****/
unsigned char bahn_Sensor(unsigned char analog);

/*****

```


Bilder der Modell-Lastfahrzeugen

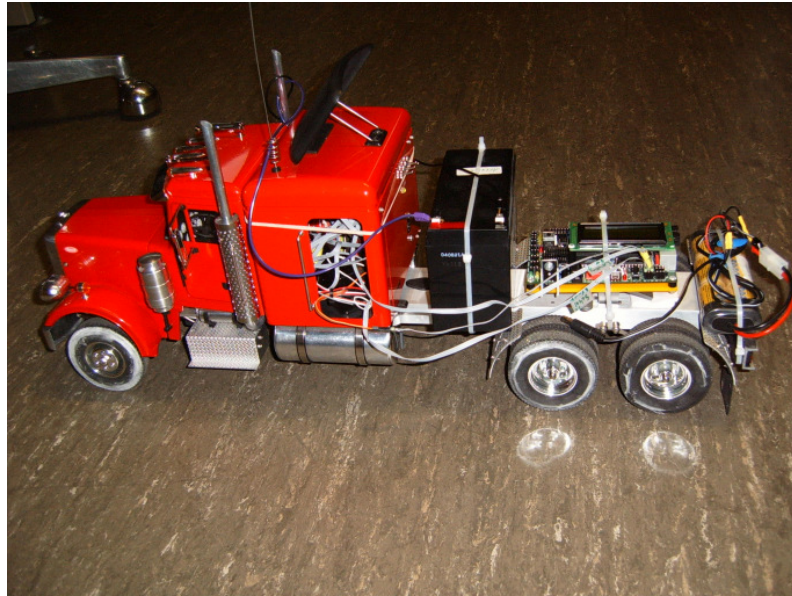


Bild A-1: Die Verkablung des Modell-Lastwagen mit dem Axen-Board.



Bild A-2: Wandverfolgung.



Bild A-3: Spurwechsel zum Überholen.



Bild A-4: Überholvorgang.

Inhalt der CD

Zu dieser Arbeit wird eine CD beigelegt. Die Verzeichnis-Struktur dieser CD ist im Bild A-5 dargestellt.



Abbildung 7-3Bild A-5: die Verzeichnis-Struktur der CD.

- ◆ Das Verzeichnis **Aksen-Board** enthält die Bibliotheken die Aksen-Boards, das Handbuch und alle benötigte Programme zum Programmierung des Aksen-Boards.
- ◆ Das Verzeichnis **Bachelorarbeit** enthält diese Arbeit im PDF-Format.
- ◆ Im Verzeichnis **Film** befinden sich die Filmaufnahmen, die während des Tests der autonomen Modell-Lastwagen aufgenommen wurden.
- ◆ Das Verzeichnis **Literatur** beinhaltet die elektronische Literatur, die in Rahmen dieser Arbeit benutzt wurde.
- ◆ Im Verzeichnis **LKW_Bilder** befinden sich die Bilder der Modell-Lastwagen
- ◆ Das Verzeichnis **src** enthält der Programmcode der autonomen Modell-Lastwagen.

Versicherung über die Selbständigkeit

Hiermit versichere ich, dass die vorgelegte Arbeit im Sinne der Prüfungsordnung nach §22(4) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, den _____

Arsalan
