



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Denis Schetler

Automatischer Ausweichassistent mit einer
Laserscanner - basierten Abstandsregelung für
ein fahrerloses Transportsystem

Denis Schetler

Automatischer Ausweichassistent mit einer
Laserscanner - basierten Abstandsregelung für ein
fahrerloses Transportsystem

Masterarbeit eingereicht im Rahmen der Masterprüfung
im Studiengang Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Ing. B. Schwarz
Zweitgutachter : Prof. Dr. rer. nat. St. Pareigis

Abgegeben am 12. November 2007

Denis Schetler

Thema der Masterarbeit

Automatischer Ausweichassistent mit einer Laserscanner - basierten Abstandsregelung für ein fahrerloses Transportsystem

Stichworte

Fahrerassistenzsystem, Ausweichassistent, Sick-Laserscanner, Abstandsregelung, kinematisches Fahrzeugmodell, symmetrisches Optimum, Frequenzkennlinienverfahren, digitaler Regler, räumliches Mittelwertfilter, Sensor Controlled Vehicle, FAUST-Projekt, QNX-RTOS, PentiumIII

Kurzzusammenfassung

Diese Masterarbeit beschreibt die Entwicklung eines automatischen Ausweichassistenten zur Kollisionsvermeidung mit einer Laserscanner - basierten Abstandsregelung. Das Software-Konzept ist in einer Drei-Ebenen Struktur mit Situationserkennung, Situationsanalyse mit Aktionsentscheidung, sowie Aktionsausführung aufgeteilt. Es werden die Arbeitspakete von der Herleitung des kinematischen Einspurfahrzeugmodells über die Beschreibung des geometrischen Modells zur relativen Abstandsbestimmung bis hin zum Entwurf des Regelkreises vorgestellt. Zur Messwertaufbereitung wird ein räumliches Mittelwertfilter genutzt. Die Dimensionierung des PD-Reglers wird mit den Einstellverfahren *symmetrisches Optimum* und *Frequenzkennlinienverfahren* durchgeführt. Ein über WLAN ferngesteuertes und elektrisch betriebenes Laborfahrzeug dient als Versuchsplattform.

Denis Schetler

Title of the paper

Automatic avoiding assistant, based on a laser-scanner distance regulation, for an automated guided transport system

Keywords

driver assistance system, avoiding assistant, Sick-laser-scanner, distance closed-loop control, kinematical model of a vehicle, symmetrical optimum, frequency response method, digital controller, Sensor Controlled Vehicle, FAUST-Project, QNX-RTOS, PentiumIII

Abstract

This master's thesis describes the development of an automatic collision avoidance assistant based on a laser-scanner distance regulation. The software-concept has a three-layer structure separated into situation detection, situation analysis with action decision and action execution. The working packets contain the derivation of the kinematical model of a vehicle, the description of a geometrical model for the distance identification and the design of a closed-loop control. For processing the measurement readings, a spatial average filter has been used. The dimensioning of the PD-controller is performed with the *symmetrical optimum* and the *frequency response method*. An electrically operated vehicle which is controlled via WLAN represents the experimenting platform.

Inhaltsverzeichnis

1	Einleitung	1
2	Das Sensor Controlled Vehicle (SCV)	5
3	Entwurfskonzept für einen Ausweichassistenten	7
3.1	Der Hardwareaufbau des SCV	9
3.2	Das Software-Konzept	10
4	Das bestehende Software-System	13
4.1	Die Objekterkennung des SCV	13
4.2	Das History-Moving-Average-Filter	14
4.3	Die Aufprallerkennung des SCV	15
4.4	Die Simulation des Umgebungsmodells	16
4.5	Die Visualisierung der gefilterten Laserscannermessung und des Umgebungsmodells	16
5	Kinematisches Einspurfahrzeugmodell des SCV	18
5.1	Herleitung einer kinematischen Fahrzeugbeschreibung	19
5.2	Simulinkmodell der SCV-Kinematik	22
5.3	Diskrete Systemgleichungen des kinematischen Modells SCV	23
5.4	Simulationen des diskreten Einspurfahrzeugmodells	24
5.5	Vergleich von zeitkontinuierlichen mit zeitdiskreten Systemgleichungen	26
5.6	Linearisierung der Systemgleichungen in ausgewählten Arbeitspunkten	26
6	Berechnung der Fahrzeugposition und des Achswinkels mit Laserscannermess- werten	29
6.1	Räumliches Mittelwertfilter zur Unterdrückung des Messrauschens	29
6.2	Bestimmung des Aufenthaltsbereichs des SCV während eines Ausweichvorgangs	32
6.3	SCV vor dem Hindernis	34
6.4	SCV schräg vor dem Hindernis	35
6.5	SCV neben dem Hindernis	37
6.6	Transformation der Abstandswerte von der Sensorposition S in die Vorderrad- position M	37
7	Entwurf des Abstandsregelkreises	39
7.1	Regelkreisübertragungsfunktionen	39
7.1.1	Auswahl des PD-Reglertyps	40
7.1.2	Die Regelgröße y-Abstand	41
7.1.3	Die Übertragungsfunktion des SCV	41
7.1.4	Die Führungsübertragungsfunktion des Regelkreises	42
7.2	Reglerdimensionierung mit dem symmetrischen Optimum	43
7.2.1	Bestimmung der Reglerparameter mit verschiedenen Vorgaben	46
7.2.2	Ergebnissimulationen zum symmetrischen Optimum	47
7.2.3	Betrachtung der Systemstabilität	48
7.3	Reglereinstellung nach dem Frequenzkennlinienverfahren	49
7.3.1	Die Übertragungsfunktion des offenen Regelkreises	50
7.3.2	Analyse der Frequenzkennlinien	50
7.3.3	Ergebnissimulationen zum Frequenzkennlinienverfahren	53

7.4	Auswahl des Reglers	54
7.5	Entwurf des digitalen PD-Reglers	56
8	Software - Modellierung des Ausweichassistenten	59
8.1	Implementierung des räumlichen Mittelwertfilters	59
8.2	Implementierungsdetails der Steuerung des Ausweichassistenten	60
8.3	Bestimmung der Wegdifferenzen mit der Odometrie	65
8.4	Erweiterung der QNX-Photon Testumgebung	66
9	Messungen der SCV-Ausweichvorgänge	68
9.1	Simulationen des SCV-Ausweichvorgangs	68
9.2	Test des Ausweichassistenten mit dem SCV	70
9.3	Realtime Verhalten der Aktionsausführung	73
9.4	Vergleich zu dem Ausweichassistenten mit dem Deichsel-Ansatz	75
10	Alternativer Ansatz zur relativen Ortsbestimmung und weitere Entwicklungsschritte	77
11	Zusammenfassung	79
	Literatur	81
	Abbildungsverzeichnis	83
	Tabellenverzeichnis	85
A	Inbetriebnahme LD-OEM1000	86
B	Mittelungsergebnis des History-MA-Filter bei der Aktivierung des Bypasses	87
C	Simulink Simulationsmodelle	89
D	Matlab Simulationscode	92
E	Gleichungen zur Dreiecksberechnung	99
F	Berechnung eines exemplarischen Arbeitspunktes aus der Linearisierung	100
G	Messreihen des räumlichen Mittelwertfilters	101
H	Transformation der Abstandswerte, Beispielrechnung und Rechts-Ausweichen	103
I	Bode-Diagramme zu Regler 3	104
J	Dymola Simulationsmodell zum Reglervergleich	105
K	Ausgewählter Sourcecode des Ausweichassistenten	106
L	Kalman-Filter zur Sensordatenfusion	122
M	Inhalt der beigelegten CD	127

1 Einleitung

In den 27 EU-Mitgliedstaaten kommen jährlich 40.000 Menschen ums Leben und eine noch viel größere Anzahl von Verkehrsteilnehmern wird verletzt. Der volkswirtschaftliche Schaden, der dadurch entsteht, beträgt über 160 Mrd. € pro Jahr. Das Ziel der Europäischen Kommission ist es, diese Zahl der Unfalltoten im Straßenverkehr EU-weit bis 2010 zu halbieren. Das eSafety Projekt soll durch den Einsatz von *intelligenten* Fahrzeugsystemen die Sicherheit im Straßenverkehr verbessern. Die *Intelligent Car Initiative* wurde gemeinsam von der Europäischen Kommission und europäischen Unternehmen ins Leben gerufen. Das Forschungsprojekt PREVENT, das unfallvermeidende Sicherheitssysteme erforscht, wird durch eSafety mitfinanziert [1].

Das Konsortium für die PREVENT Gruppe setzt sich aus vielen Automobilherstellern, Zulieferer und Forschungseinrichtungen zusammen, wie z. B. Audi, BMW, Peugeot, Citroen, VW, Bosch, IBEO, Siemens VDO, der Fraunhofer Gesellschaft und der Universität Hannover. PREVENT konzentriert sich auf die präventive Unfallvermeidung, die zur aktiven Sicherheit beiträgt. In der ersten Stufe der Unfallvermeidung werden dem Fahrer Informationen und Warnungen mitgeteilt, reagiert der Fahrer nicht und steigt die Unfallwahrscheinlichkeit, greifen Assistenzsysteme ein, um einen Unfall zu vermeiden. Reicht dieser Eingriff nicht aus, so werden umkehrbare Pre-Crash Systeme aktiviert, wie beispielsweise die präventive Straffung der Sicherheitsgurte [2]. Dieser gesamte Bereich zur Unfallvermeidung und Schadensminimierung gehört zu den Forschungsprojekten von PREVENT [3].

Bei Unfällen mit Personenschäden im Straßenverkehr, die durch Fehlverhalten der Fahrzeugführer verursacht werden, sind diese in Deutschland zu 27,7 % auf eine überhöhte Geschwindigkeit oder zu geringen Abstand zurückzuführen (vgl. Abb. 1). Da die Fahrerassistenzsysteme, wie ein Ausweichassistent, ein Umgebungsmodell erstellen, werden eine zu hohe Geschwindigkeit oder ein zu geringer Abstand erkannt und bei einem bevorstehenden Unfall entsprechend eingegriffen. Somit wirken Fahrerassistenzsysteme dieser Art ungefähr einem Viertel aller Unfälle entgegen, die durch Fahrfehlverhalten verursacht werden.

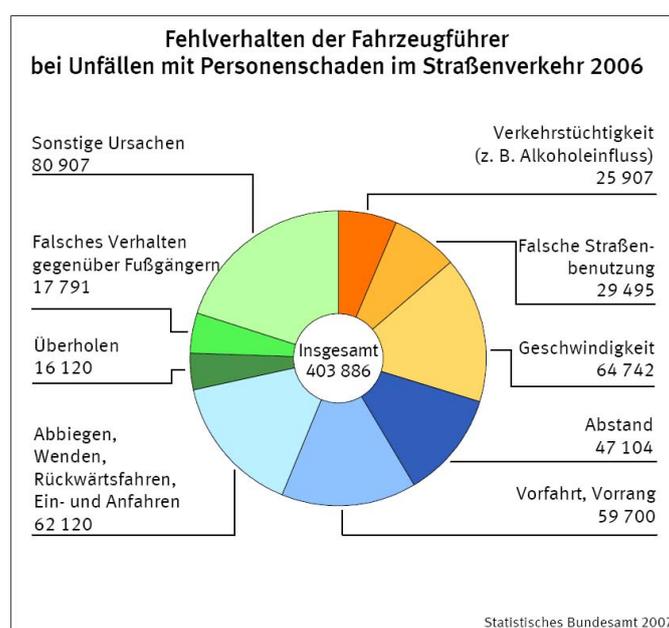


Abb. 1: Personenschäden im Straßenverkehr und Unfallursachen [4]

Im Forschungs- und Entwicklungsprojekt *Fahrerassistenz- und Autonome Systeme* (FAUST) des Departments Informatik an der Hochschule für Angewandte Wissenschaften Hamburg wurde das Versuchsfahrzeug *Sensor Controlled Vehicle* (SCV) geschaffen, welches in Projekten und Abschlussarbeiten weiterentwickelt wird. Das Fahrzeug basiert auf einer AGV¹-Schulungs-Plattform der Firma E&K, mit den folgenden Komponenten:

- zwei Antriebsmotoren
- zwei Lenkmotoren
- zwei ARM7 μC mit 32 MHz / 16 Bit
- Laserscanner SICK LD OEM1000, maximaler Scanbereich: 360°
- Laserscanner SICK LMS200, maximaler Scanbereich: 180°
- zwei Embedded GEME-Rechner mit Pentium III 650 MHz
- WLAN Access Point Asus WL-500g
- einer Kamera D-LINK DCS-2100+

Das SCV kommuniziert über WLAN mit einem Labview basierten Leitstand, an dem sich das Fahrzeug fernsteuern lässt und Fahrerassistenzsysteme aktiviert werden. In vorausgegangenen Master- und Bachelor-Projekten sind bereits eine Fahrsteuerung, ein Leitstand, sowie Assistenten für Ausweich-, Brems- und Einparkvorgänge realisiert worden [5], [6], [7]. Das SCV stellt durch seine verschiedenen Sensoren wie z.B. Laserscanner und Drehmessgeber, sowie die geregelten Aktoren, eine geeignete Erprobungsplattform für Fahrerassistenzsysteme und verteilte Embedded Systeme dar.

Der bisherige Ausweichassistent stellt seinen Soll-Lenkwinkel über eine virtuelle Deichsel ein. Der Lenkwinkel α wird direkt aus der zyklischen Abstandsmessung mit der aktuellen Ausweichbreite y_{dl} und dem Abstand des Hinderrades zum Hindernis P_{ax} berechnet (vgl. Gl. 1) [5].

$$\alpha = \tan^{-1} \left(\frac{y_{dl}}{P_{ax}} \right) \quad (1)$$

Das Ziel dieser Masterarbeit bestand darin, einen alternativen Ansatz für den bisherigen Ausweichassistenten auf dem SCV zu entwickeln. In dem weiterzuentwickelnden System sollte der Soll-Lenkwinkel durch eine übergelagerte Abstandsregelung bestimmt werden. Zu Beginn wurde das bestehende System auf Wiederverwendbarkeit geprüft. Das kinematische Einspurfahrzeugmodell wurde hergeleitet und für die Positionsbestimmung des SCV ein geometrisches Modell erstellt. Zur Unterdrückung des Messrauschens des Laserscanners wurde ein räumliches Mittelwertfilter entwickelt und untersucht. Der Abstandsregler für den Ausweichassistenten wurde mit dem *symmetrischen Optimum* und dem *Frequenzkennlinienverfahren* dimensioniert. Bei Letzterem wurden die Kennlinien mit den Bode-Diagrammen des offenen Regelkreises und des Reglers analysiert. Der PD-Regler ist mit der z-Transformation anschließend digitalisiert worden. Der entworfene Ausweichassistent wurde unter dem Betriebssystem QNX auf einem der Pentium III Prozessoren realisiert. Die Funktionalität ist mit simulierten und realen Ausweichmanövern getestet und ausgewertet worden. Anschließend wurde ein Vergleich der unterschiedlichen Ansätze für einen Ausweichassistenten durchgeführt.

¹AGV: Automatic Guided Vehicle

Der Ausweichassistent nutzt den Laserscanner SICK LD OEM1000, um die Abstandswerte im Bereich 180° vor dem SCV zu messen. Aus den entsprechenden Daten erstellt die Objekterkennung ein Umgebungsmodell. Durch Detektion einer bevorstehenden Kollision mit einem Objekt findet eine Aktivierung des Ausweichvorgangs statt. Während des Manövers werden die Abstände zwischen dem SCV und dem Hindernisobjekt bestimmt. Nach erfolgreichem Ausweichen kommt das SCV parallel zum Hindernis zum Stehen.

Diese Masterarbeit teilt sich in drei Schwerpunkte auf:

- Der analytische Teil dieser Arbeit schafft die Basis für die Entwicklung des Ausweichassistenten. Ein Einspurfahrzeugmodell des SCV wird hergeleitet, mit der Trapezintegration diskretisiert und anschließend mit der Ableitung um den Arbeitspunkt linearisiert. Für das Messwertrauschen aus der Laserscannermessung wird ein neues Filter vorgestellt, das innerhalb eines Scans über räumlich benachbarte Scanwinkel den Mittelwert bildet. Eine Herleitung der geometrischen Modelle für die Bestimmung der Abstände und des Achswinkels rundet dieses Schwerpunktthema ab.
- Der regelungstechnische Teil leitet den Regler für die Abstandsregelung her. Die Eigenschaften des Regelkreises für den Ausweichassistenten werden vorgestellt und anschließend die Reglerparameter mit den Einstellverfahren *symmetrisches Optimum* und *Frequenzkennlinienverfahren* bestimmt. Für die Software-Implementierung in das SCV findet eine Digitalisierung des Reglers in Form einer z-Transformation statt.
- Die Software-Modellierung des Ausweichassistenten auf einem QNX-RTOS-Thread-System mit einem Round-Robin Scheduler wird präsentiert. Die Funktionsblöcke des Systems sind in Module gekapselt und nach der Verarbeitungskette strukturiert. Aus den periodisch gesendeten CAN-Bus-Nachrichten vom Laserscanner werden in der Situationserfassung Objekte aus den Daten extrahiert. Die Situationsanalyse übernimmt die Liste der Objekte und schickt abhängig von der Auswertung Signale an den Entscheidungsautomaten für die Ausweichrichtung. Dieser ist nach dem AIRA-Automatenmodell für hierarchische Zustandsautomaten realisiert. Für die Ausführung des Ausweichvorganges wird ein eigenständiger Thread gestartet, der zyklisch die relativen Abstände zum Hindernis berechnet, den Soll-Lenkwinkel mit einem PD-Regler bestimmt und diesen in Form einer Nachricht an den Steuerungsautomaten schickt.

Der Aufbau dieser Dokumentation ist wie folgt gegliedert:

- Kapitel 2 beinhaltet die Vorstellung des Versuchsfahrzeugs *Sensor Controlled Vehicle* mit seinem Projektkontext.
- Das Konzept des Ausweichassistenten wird in Kapitel 3 vorgestellt. Dazu zählt die grobe Arbeitsweise des Abstandsreglers, sowie der Hardwareaufbau des SCV und das Software-Konzept mit der Threadmodellierung und dem Modulwurf der Software-Struktur.
- Die Darstellung des bestehenden Software-Systems präsentiert das Kapitel 4. Dies beinhaltet die Objekterkennung, das Moving-Average-Filter zur Messrauschunterdrückung und die Aufprallerkennung mit einem Objekt.
- Ein kinematisches Einspurfahrzeugmodell für das SCV beschreibt das Kapitel 5. Die Systemgleichungen für die Radpositionen und den Achswinkel werden hergeleitet und für eine Kreisbahn in Matlab Simulink simuliert. Zusätzlich wird die Lenkverzögerung des SCV bestimmt. Die Gleichungen werden mit der Trapezintegration diskretisiert und anschließend für ausgewählte Arbeitspunkte linearisiert.

- In Kapitel 6 wird ein räumliches Filter zur Messrauschunterdrückung vorgestellt und mit dem Moving-Average-Filter verglichen. Anschließend folgt eine Präsentation des geometrischen Modells für die relative Abstandsberechnung des SCV zum Hindernis.
- Einen Entwurf des Abstandsregelkreises mit einem PD-Regler für den Ausweichassistenten zeigt Kapitel 7. Die Dimensionierung der Reglerparameter erfolgt dabei mit den beiden Einstellverfahren *symmetrisches Optimum* und *Frequenzkennlinienverfahren*. Für das System wird eine Stabilitätsbetrachtung vollzogen und der Ausweichvorgang simuliert, bevor der Regler mit der z-Transformation für die Software-Implementierung digitalisiert wird.
- Die Modellierung der Software-Module ist in Kapitel 8 dokumentiert. Dies beginnt bei der Vorverarbeitung durch das *räumliche Mittelwertfilter* und geht über die *Situationsanalyse und Aktionsentscheidung* bis zu der *Aktionsausführung*. Zusätzlich werden die Threads des Software-Systems mit ihren Prioritäten und ihren blockierenden Funktionen betrachtet.
- Das Kapitel 9 stellt die Testergebnisse des Ausweichassistenten in der Simulation und in der realen Umgebung dar. Dazu wurden verschiedene Szenarien entworfen. Es folgt eine Darstellung unterschiedlicher Messreihen u.a. für den Achswinkel des SCV und für die Regeldifferenz und eine Analyse des Realtime-Verhaltens des Threads für die Aktionsausführung. Ein Vergleich der beiden Ansätze für einen Ausweichassistenten schließt dieses Kapitel ab.
- Für die relative Positionsbestimmung des SCV wird in Kapitel 10 ein alternativer Ansatz vorgestellt, der mit einer geringeren Anzahl von Rechenschritten auskommt und ausschließlich Addition und Subtraktion verwendet. Zusätzlich werden einige Konzepte für die weitere Entwicklung des Ausweichassistenten aufgezeigt.
- Die Präsentation der Ergebnisse bildet den Abschluss dieser Masterarbeit in Kapitel 11.

2 Das Sensor Controlled Vehicle (SCV)

Das Versuchsfahrzeug *Sensor Controlled Vehicle* (SCV) ist im Projekt *Fahrerassistenz- und Autonome Systeme* (FAUST) im Department Informatik an der Hochschule für Angewandte Wissenschaften Hamburg eingegliedert. Das Ziel des Projektes ist es, eine autonom gesteuerte Plattform zu entwickeln, die sich mit Laserscannern und Kameras in der Umgebung orientiert und Fahraufträge abarbeitet. Dazu muss sich das SCV mit seinen Sensoren selbst im Raum lokalisieren, den abzufahrenden Pfad in einer Karte bestimmen und den vorgegebenen Pfad steuern. In [8] werden verschiedene Verfahren für diese Aufgaben mit einem Laserscanner als Sensor vorgestellt.

In der aktuellen Phase des Projekts werden im Rahmen von Master- und Bachelorprojekten Fahrerassistenzsysteme aus der Automobilbranche auf das SCV übertragen. Diese Systeme wie Park-, Brems- und Ausweichassistenten führen eine Objekterkennung durch und orientieren sich im Raum relativ zu den Bezugsobjekten. Zum Beispiel orientiert sich der Parkassistent an der Parklücke, während sich der Ausweichassistent am Hindernisobjekt ausrichtet. Es wird aber keine absolute Lokalisation im Raum durchgeführt.

Ist kein Fahrerassistenzsystem des SCV aktiv, so wird das SCV über den Leitstand mit einem Lenkrad und zwei Pedalen ferngesteuert. Das Fahrzeug ist 1 m breit, verfügt über eine Länge von 1,2 m und besitzt vorne und hinten ein Rad mit Antriebs- und Lenkmotor (vgl. Abb. 2). Auf der linken und rechten Seite befindet sich jeweils mittig ein Stützrad. Der 360°-Laserscanner für die Umgebungsmessung ist vorne-links auf dem SCV montiert.

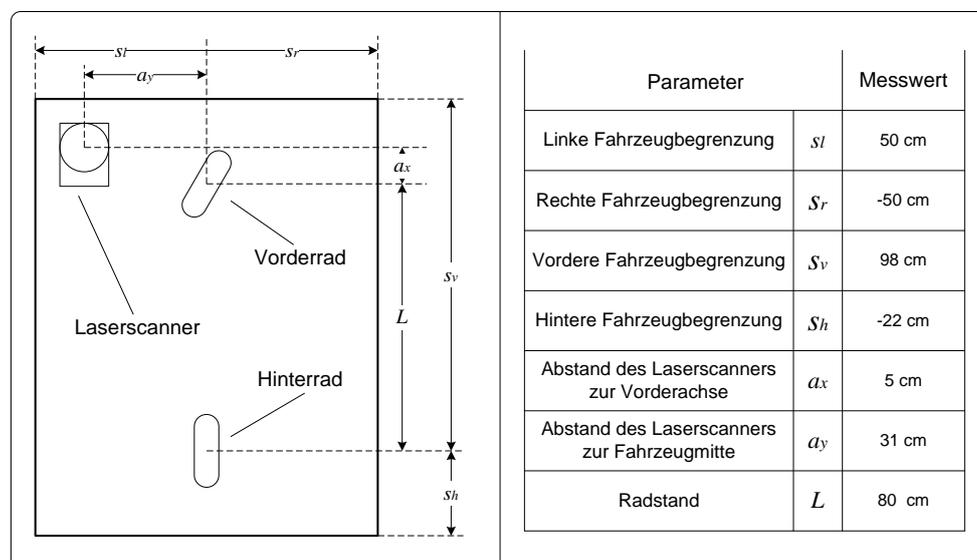


Abb. 2: Abmessungen des Sensor Controlled Vehicle (Vorlage aus [6])

Der Kommunikationsrechner auf dem SCV empfängt die Fahrbefehle über WLAN und leitet diese auf den CAN-Bus weiter. Die Befehle setzt der Koordinierungsrechner entsprechend des aktiven Fahrmodus für die Regelungscontroller vorn und hinten um (vgl. Abb. 3). Das Fahrzeug stellt folgende Fahrmodi zur Verfügung:

- *Front*: Das Vorderrad setzt den kompletten Lenkwinkel um.
- *Rear*: Das Hinterrad setzt den kompletten Lenkwinkel um.
- *Synchron*: Jeweils der halbe Lenkwinkel wird für das Vorder- und Hinterrad eingestellt.

Da sich der Ausweichassistent an der Automobilbranche orientiert, wird das SCV für die Fahrerassistenzsysteme im *Front-Mode* betrieben. Das SCV ist mit verschiedenen Sensoren für die Umgebungserfassung ausgestattet. Mit dem Laserscanner Sick LD-OEM1000² lässt sich die Umgebung mit 360° horizontal und mit dem Sick LMS 200 mit 180° vertikal abscannen. Die Kamera DCS-2100+ liefert Bilder des Bereichs vor dem Fahrzeug.

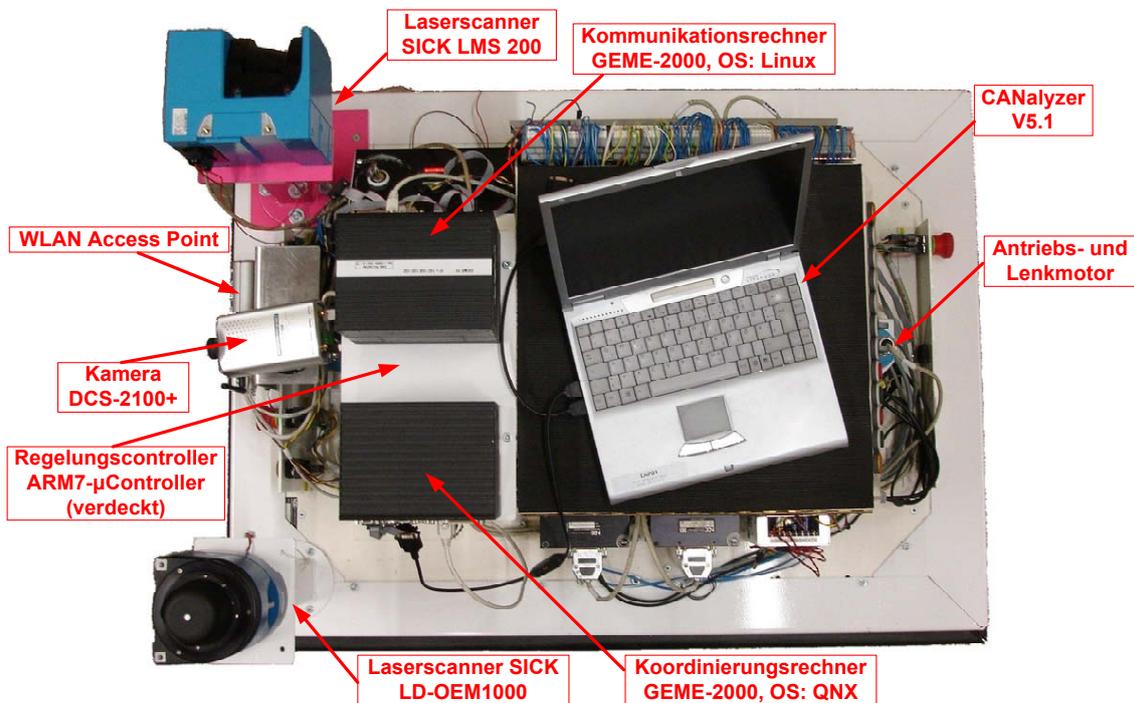


Abb. 3: Aufbau des Sensor Controlled Vehicle

Für das SCV wurde bereits ein Software-Konzept für ein verteiltes, zeitgesteuertes Hard-Real-Time System mit FlexRay entwickelt [9]. Die verteilte Regelungsanwendung für die Antriebs- und Lenkmotoren wurde auf FlexRay-Komponenten umgesetzt. In dieser Masterarbeit nutzt das System keine FlexRay-Komponenten, da die Fahrsteuerung des SCV die ARM7 μC übernehmen.

²LD: Laserdiode

3 Entwurfskonzept für einen Ausweichassistenten

Zur Verbesserung der aktiven Sicherheit wird in dieser Masterarbeit ein Ausweichassistent basierend auf einer Abstandsregelung entwickelt. Das System soll erst eingreifen, wenn eine Kollision mit dem Hindernis unmittelbar bevor steht. Der Fahrzeugbenutzer behält somit längstmöglich die Kontrolle. Mit dem System wird der Bereich Assistenzsysteme des PREVENT Projekts umgesetzt (vgl. Kapitel 1). Die übrigen Bereiche des PREVENT Projekts sind nicht Bestandteil dieser Masterarbeit, d. h. es gibt vor dem automatischen Eingriff keine Kollisionswarnung an den Fahrer, ebenso entfällt eine Aktivierung von umkehrbaren Pre-Crash Systemen.

In diesem Kapitel werden die einzelnen Phasen während des Ausweichens, die aus der Laserscannermessung zu bestimmenden Abstände, der Hardwareaufbau des SCV und die Softwarestruktur des Koordinierungsrechners vorgestellt, auf dem der Ausweichassistent realisiert ist. Die Erläuterungen und Abbildungen beziehen sich dabei auf das Links-Ausweichen, entsprechendes gilt für das Rechts-Ausweichen.

Der Ausweichassistent des SCV soll automatisch einem Hindernis ausweichen. Eine Kollision soll verhindert werden, wenn der Fahrer nicht reagiert. Dazu ist der Bereich vor dem Fahrzeug auf Objekte abzuscanen. Die Detektion einer bevorstehenden Kollision mit einem Hindernis im Gefahrenbereich hat die Auslösung des Ausweichvorgangs zur Folge (Abb. 4). Hierbei wird das lenkbare Vorderrad durch ein M, das starre Hinterrad durch ein P und der 360°-Laserscanner durch ein S bezeichnet.

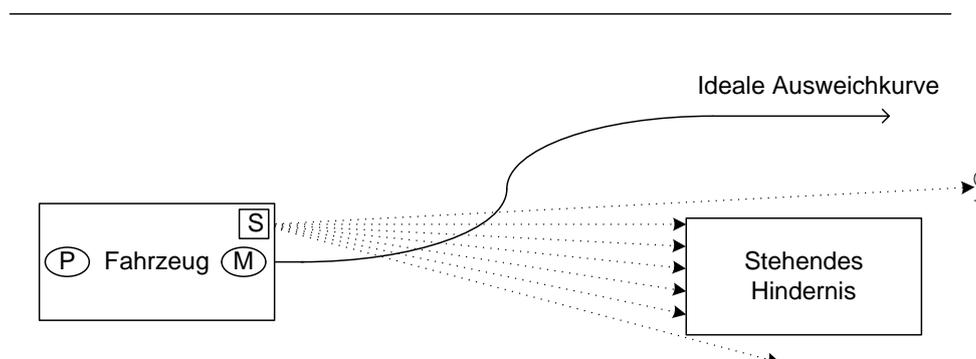


Abb. 4: Scannen vor dem Hindernis, vor dem Ausweichen

Während des Ausweichens werden mit den Messwerten aus der Laserscannermessung die Abstände in x- und y-Richtung zum Hindernis bestimmt (Kapitel 6), um mit einer Abstandsregelung den Vorderrad-Soll-Lenkswinkel vorzugeben (Kapitel 7). Während des Ausweichvorgangs durchfährt das Fahrzeug dabei drei Bereiche, die wegen der geometrischen Berechnungen für die Abstände M_{ax} , M_{ay} und den Achswinkel θ unterschieden werden. Zu Beginn befindet sich das Fahrzeug im Bereich *vor dem Hindernis* (vgl. Abb. 5). Ist der Ausweichvorgang soweit fortgeschritten, dass der Laserscanner nicht mehr auf der Höhe des Hindernisses ist, befindet sich das SCV in dem Bereich *schräg vor dem Hindernis*. Sobald der Sensor S auf der Höhe der Hindernisfront ist, ist das SCV im Bereich *Neben Hindernis*. Die Umgebung eines stehenden Hindernisses teilt sich somit in die drei genannten Bereiche auf. Im letzten Abschnitt des Ausweichvorgangs wird das SCV parallel mit einem Sicherheitsabstand zum Hindernis ausgerichtet und anschließend angehalten. Für die Entwicklung und Durchführung von Tests für den Ausweichassistenten sind die folgenden Rahmenbedingungen realisiert:

- Die Geschwindigkeit, die bei dem automatischen Start des Ausweichvorgangs von dem Fahrer am Leitstand vorgegeben wird, bleibt während des Ausweichmanövers konstant.
- Der Fahrer muss während des Ausweichvorgangs das Vorwärts-Pedal betätigen. Ist dies nicht der Fall, so folgt der Abbruch des Assistenten und das SCV kommt zum Stehen. Dieses Kriterium dient zur Kontrolle über das System bei der Evaluierung des Prototypen.

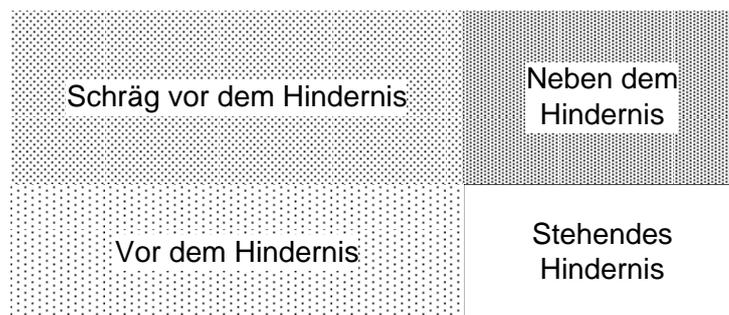


Abb. 5: Unterteilung der Positionsbereiche des SCV während des Ausweichens

Hat ein zu geringer Abstand zum Hindernis M_{ax} einen Ausweichvorgang zur Folge, so wird für den Abstandsregler eine Sprungantwort in der Höhe von $y_{max} + M_{ay}$ generiert. Dies geschieht unter Berücksichtigung des Verhältnisses M_{ax} zu y_{max} . Y_{s1} stellt dabei die Führungsgröße dar. In den Bereichen *vor dem Hindernis* und *schräg vor dem Hindernis* werden die relativen Abstände M_{ax} und M_{ay} zum Hindernis bestimmt, um die Regeldifferenz zwischen Y_{s1} und M_{ay} im Regelkreis zu ermitteln (vgl. Abb. 6).

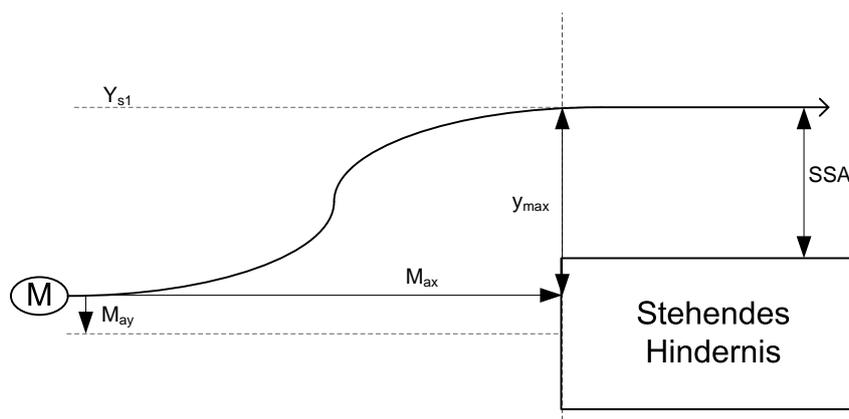


Abb. 6: Abstände des lenkbaren Vorderrads M für die Regelung

Im Bereich *Neben dem Hindernis* ist das Fahrzeug dem Hindernis ausgewichen. Da die Regelung aber für das Vorderrad ausgelegt wurde, ist das SCV zu Beginn in diesem Bereich noch nicht parallel zum Hindernis ausgerichtet. Da der Abstand M_{ay} nicht mehr bezogen auf die halbe Hindernisbreite gemessen werden kann, sondern nur zur Seitenfront des Hindernisses, wird die Führungsgröße des Regelkreises auf den Sicherheitsseitenabstand SSA gesetzt. Somit

ändert sich die Führungsgröße des Regelkreises, die in diesem Bereich als Y_{s2} bezeichnet wird. Das Verhältnis der Regeldifferenz ändert sich dabei nicht, da sowohl die Rückführgröße M_{ay} als auch die Führungsgröße Y_s angepasst wurden (vgl. Abb. 7). Der Ausweichvorgang ist komplett abgeschlossen, wenn sich das SCV parallel zum Hindernis befindet.

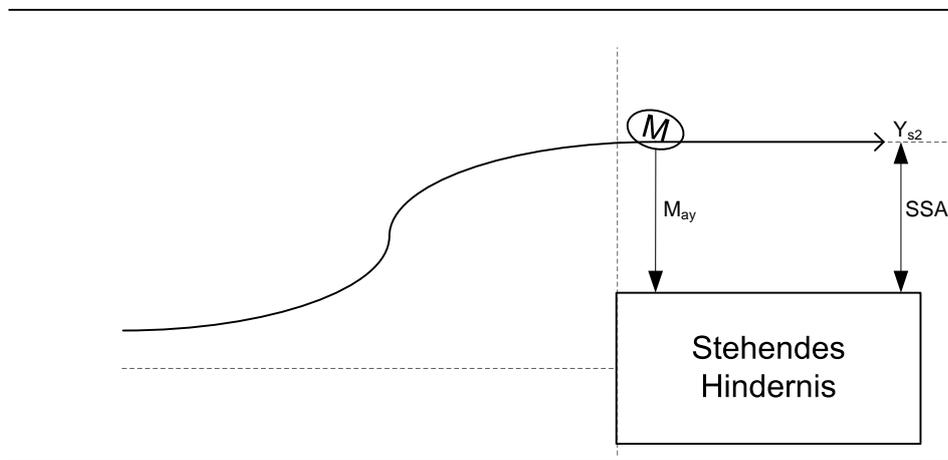


Abb. 7: Abstände des lenkbaren Vorderrads M für die Regelung *neben dem Hindernis*

3.1 Der Hardwareaufbau des SCV

Der Benutzer steuert das Fahrzeug am Leitstand, der über WLAN die Fahrbefehle zum Kommunikationsrechner schickt (Abb. 8). Diese Befehle werden per CAN-Nachrichten weiter an den Koordinierungsrechner gesendet, der die Regelung für das Vorder- und Hinterrad steuert. Der Ausweichassistent ist im Koordinierungsrechner realisiert, indem er die Ansteuerung für die Fahr- und Lenkmotoren übernimmt, wenn ein Ausweichvorgang ausgeführt wird. Die Objekterkennung nutzt zurzeit nicht die Kamera, da bisher auf dem SCV keine Bildverarbeitung realisiert wurde. Der Laserscanner SICK LMS200 scannt den Bereich seitlich des SCV ab. Einem Objekt in dieser Position wird nicht ausgewichen, deshalb verwendet der Ausweichassistent den SICK LMS200 nicht. Vor dieser Masterarbeit war ein Laserscanner des Typs SICK LD OEM mit einem Scanbereich von 360° auf dem SCV installiert. Da andere FAUST-Projekte inzwischen den Laserscanner nutzen, steht dieser nicht mehr für das SCV zur Verfügung. Deshalb wurde der 360° -Scanner durch eine neue Generation des Typs, dem SICK LD OEM1000 ersetzt. Die Winkelauflösung, Funktionalität und Protokollansteuerung ist für den neuen Typ identisch. Die Initialisierung und der Hardwareanschluss mussten angepasst werden, die Details sind im Anhang A dokumentiert. Die Software lässt sich für zwei unterschiedliche Initialisierungs-Modi kompilieren:

- Der Laserscanner wird bei der Initialisierung über CAN-Nachrichten konfiguriert und anschließend ein CPU-Reset ausgeführt.
- Der Laserscanner wird angehalten und in den Idle-Mode gesetzt. Dies hat den Vorteil, dass Schreibzyklen auf dem EEPROM vermieden werden.

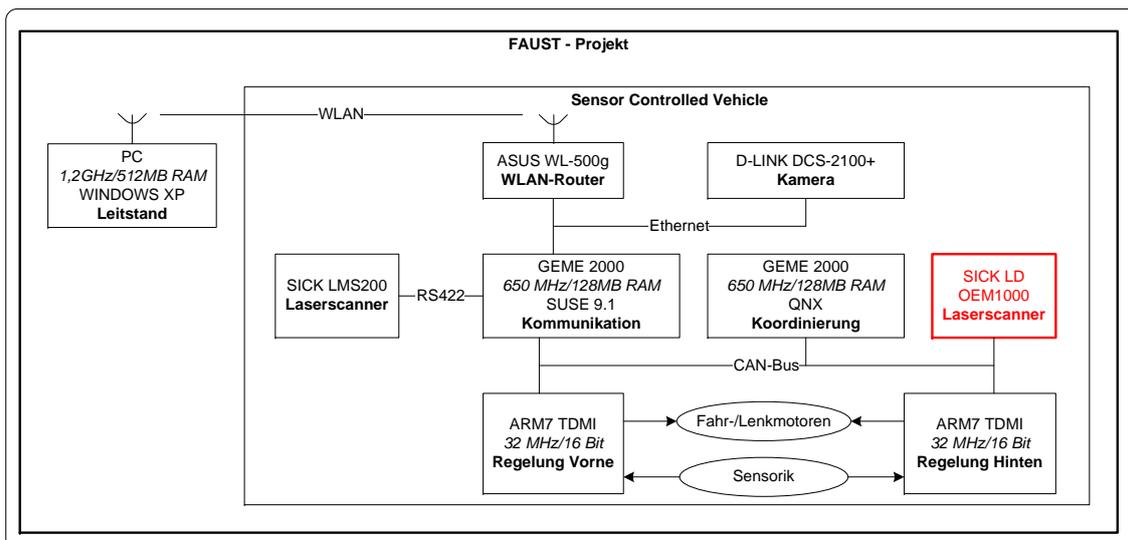


Abb. 8: HW-Aufbau des Sensor Controlled Vehicle

3.2 Das Software-Konzept

Zur Realisierung der Software für den Ausweichassistenten wurde die Softwarestruktur der bestehenden Fahrerassistenzsysteme [5] und [6] analysiert und ein Konzept für die Anpassungen, Erweiterungen und Neuentwicklungen erstellt. Die Drei-Ebenen-Struktur des Assistenzsystems blieb bestehen.

- Situationserkennung
- Situationsanalyse und Aktionsentscheidung
- Aktionsausführung

Die Erfassung der Umgebungs- und Fahrzeugsituation durch die Sensordaten des Laserscanners bildet den ersten Schritt, d.h. er bestimmt die relativen Abstände zum Hindernis und den Achswinkel des Fahrzeugs. Im zweiten Schritt findet die Beurteilung der Situation und die anschließende Bestimmung der Reaktion statt. Ist der Ausweichvorgang beispielsweise noch nicht gestartet und erweist sich der Abstand zum Hindernis als zu gering, so wird der Ausweichvorgang als Aktion ausgewählt. Der letzte Schritt beinhaltet die Ausführung der gewählten Aktion, in diesem Fall hat er das Starten des Assistenten und die Ansteuerung der Lenkmotoren zur Folge. Dieser Eingriff wirkt sich natürlich auf die Umgebungs- und Fahrzeugsituation aus, womit der Kreis geschlossen wird.

Der in dieser Masterarbeit realisierte Ausweichassistent setzt die Situationsanalyse und die Aktionsausführung mit einem anderen Verfahren um, folglich sind diese Threads verändert worden (vgl. Abb. 9). Durch den Wechsel des Laserscanners ist die *LD Initialisierung* und wegen der Implementierung des räumlichen Mittelwertfilters (Kapitel 6) ist die Messwertaufbereitung in der Situationserfassung angepasst worden. Der neu entwickelten *Odometrie*³-Thread bestimmt aus der Geschwindigkeit und dem Lenkwinkel die aktuelle Fahrzeugposition in Form einer xy-Koordinate, relativ zum Ausgangspunkt des zurückgelegten Weges. Der Thread bekommt die Ist-Daten des Vorderrades von dem Thread *CAN-Empfang*. Außerdem wird der Achswinkel berechnet. Diese Odometrie-Daten dienen zur Verifikation der ermittelten Werte aus der Laserscannermessung.

³Odometrie: Wegmessung

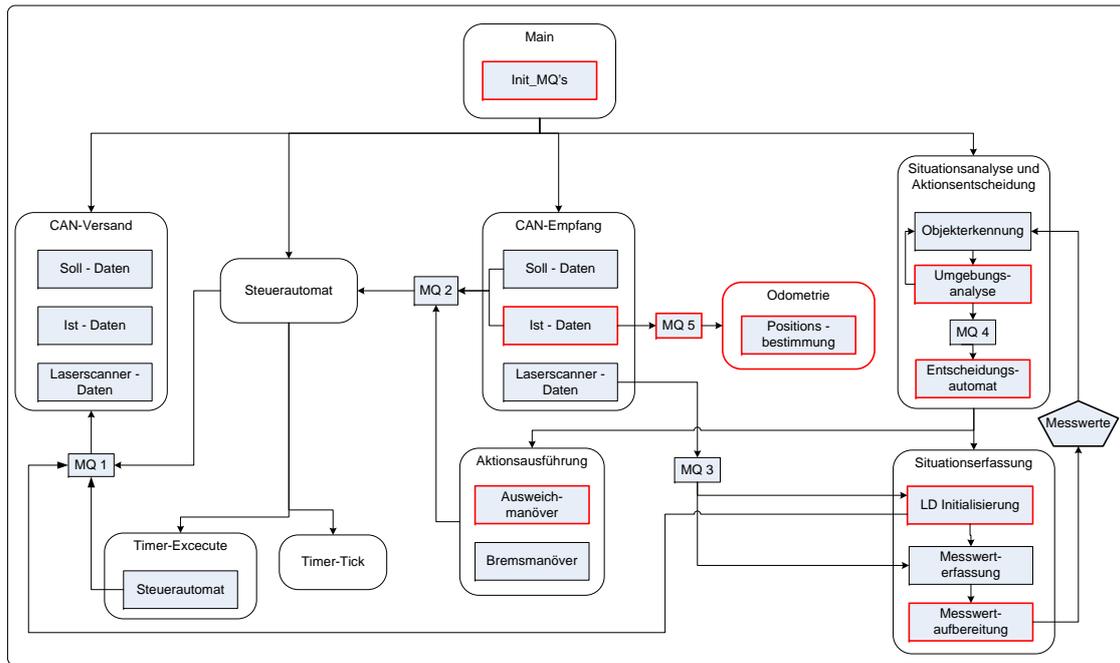


Abb. 9: Threadübersicht des Koordinierungsrechners

Die Drei-Ebenen-Struktur ist in der Softwarestruktur Abb. 10 umgesetzt. Das *Collision Avoidance System* (*CA_System*) fragt zyklisch die erfassten Objekte in der Umgebung von der *Objekterkennung* ab, die die Entfernungswerte von *LD_Comm* abrufen. Die Aufprallerkennung ermittelt anschließend aus den Objekten das Objekt mit der geringsten Aufprallzeit. Die Situationserkennung und die Aufprallerkennung sind, bis auf das neue räumliche Mittelwertfilter zur Unterdrückung des Messrauschens, in *LD_Comm*, aus den Projekten [5] und [6] übernommen. Die nachgelagerten Module sind Neuentwicklungen.

Das *CA_System* holt mit der Identifikationsnummer aus den Aufpralldaten, das Objekt mit der geringsten Aufprallzeit aus der Objektliste und übergibt dies der Statusbestimmung *AusweichStatusAbstRegl*. Diese ermittelt aus den Abstandswerten des Objekts, ob ein Ausweichen eingeleitet werden muss, jeweils für das Rechts- und Links-Ausweichen. In dem Entscheidungsautomaten *EntscheiderAbstReglFSM* wird zum letztmöglichen Zeitpunkt ein Ausweichmanöver gestartet, also wenn R_{on} und L_{on} . Dabei weicht das SCV in die Richtung aus, für die der Status später aktiviert wurde. Löst der Entscheidungsautomat einen Ausweichvorgang aus, so teilt dieser das ermittelte Manöver dem *CA_System* mit. Darauf folgend startet das *CA_System* das Links- bzw. Rechts-Ausweichen für das Hindernis im Modul *AusweichenAbstRegl*. Dieses Modul repräsentiert die Aktionsausführung der Drei-Ebenen-Struktur. Abhängig von dem aktuellen Aufenthaltsbereich des SCV werden die Abstände und der Achswinkel mit den entsprechenden Funktionen berechnet. Der Abstand in y-Richtung wird an den PD-Regler als Rückführgröße des Regelkreises übergeben, der die Stellgröße Soll-Lenkswinkel der Ausweichsteuerung vorgibt. Nach Beendigung des Ausweichvorgangs setzt das Modul *AusweichenAbstRegl* das Manöver im *CA_System* wieder zurück. Der Ausweichassistent ist wieder für einen neuen Ausweichvorgang bereit.

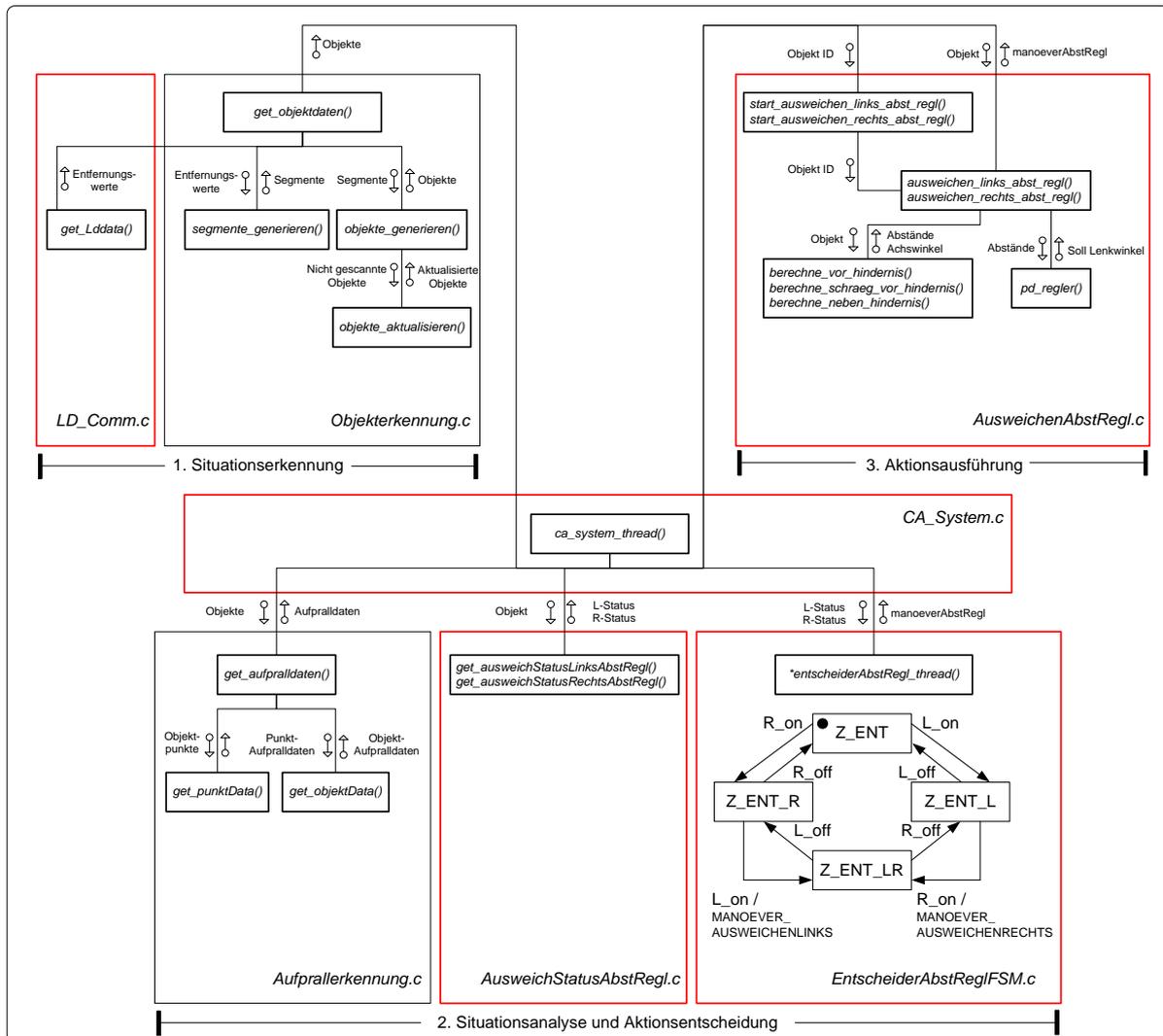


Abb. 10: Softwarestruktur des Ausweichassistenten

4 Das bestehende Software-System

Zur Einordnung der Masterarbeit gibt dieses Kapitel einen Überblick über die Module aus den Vorgängerprojekten [5] und [6]. Die Objekterkennung ermittelt aus der Laserscannermessung die Objekte aus der Umgebung. Das History-Moving-Average-Filter filtert das Messrauschen des Laserscanners, bevor die Übergabe der Abstandswerte an die Objekterkennung stattfindet. Die Aufprallerkennung ermittelt das Objekt mit der geringsten Aufprallzeit aus den Umgebungsobjekten, für den Ausweichassistenten.

4.1 Die Objekterkennung des SCV

Aus den gemessenen Entfernungswerten vom Laserscanner generiert die Objekterkennung aus benachbarten Messpunkten Segmente und vergleicht diese mit den Segmenten aus dem vorigen Scan. Für die Segmente, die eine Überdeckung zwischen zwei Scans aufweisen, erstellt das Modul ein Objekt mit einer eindeutigen Identifikationsnummer und der Objektgeschwindigkeit.

Zur Datenreduktion verringert die Objekterkennung die Kontur des Objekts für die nachgelagerten SW-Module auf drei Messpunkte:

- L : linker Punkt
- K : kürzester Punkt
- R : rechter Punkt

Die drei ermittelten Punkte L , K und R beziehen sich auf die Position des Laserscanners (vgl. Abb. 11). Die Objekterkennung führt nach der Bestimmung der Punkte eine Transformation des Koordinatenursprungs von der Sensorposition S zu der Hinderradposition P durch. Die drei Objektpunkte bleiben dadurch die gleichen, aber die Koordinaten verändert sich (vgl. Abb. 12). Zu beachten ist, dass K_P der kürzeste Objektpunkt bezogen auf P ist, der Punkt K aber bei der Transformation unverändert bleibt. Weiterführende Details sind [5] und [6] zu entnehmen.

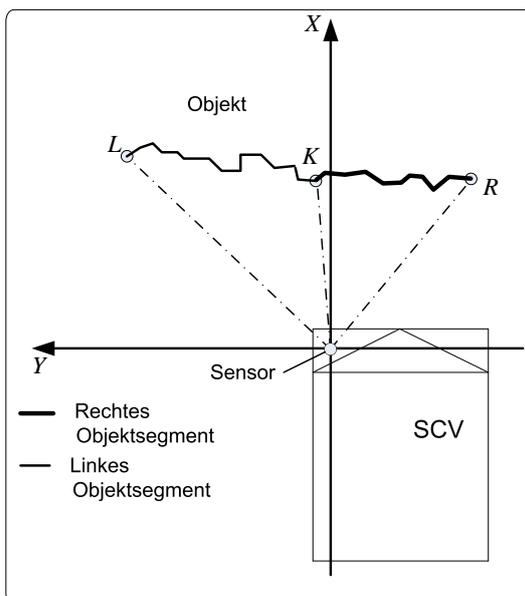


Abb. 11: Objektpunkte bezogen auf S

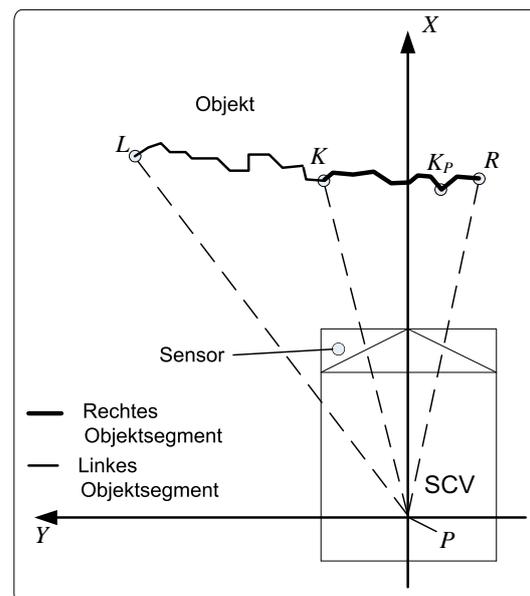


Abb. 12: Objektpunkte bezogen auf P

4.2 Das History-Moving-Average-Filter

Das History-Moving-Average-Filter (History-MA-Filter) bildet einen Mittelwert über den aktuellen und die zeitlich letzten $N - 1$ Messwerte für denselben Scanwinkel (vgl. [6] und [5]). Somit wird für jeden Scanwinkel ein Standard-MA-Filter mit der Filtertiefe N implementiert. Die Messwerte und Mittelwerte werden in einem zweidimensionalen Array verwaltet, die Größe ist:

$$(n \text{ Messungen} + \text{Mittelwert}) \cdot m \text{ Scanwinkel}$$

Durch die Mittelung der letzten $N - 1$ Scanrunden und der aktuellen Scanrunde haben die Objekte eine Verzögerung, wenn sich das SCV auf das Hindernis zubewegt. Tritt für einen Scanwinkel eine erhebliche Abstandsveränderung zwischen zwei Scanrunden auf, die außerhalb des Messrauschens ist, werden alle N Messwerte des History-MA-Filters auf den aktuellen Messwert gesetzt. Dieses in [6] Bypass genannte Verfahren dient zur schnellen Anpassung des Mittelungsergebnisses bei seitlich einfahrenden Objekten.

Der Bypass hat den Effekt, dass es bei einer Abstandsveränderung zwischen Objekt und SCV zu Objektanomalien kommt. Die Objektanomalie entsteht z.B. bei der Fahrt auf das Hindernis zu, da in dem Fall die Anzahl der Scanwinkel zunehmen, die zu dem Objekt gehören. Scanwinkel, die in der Scanrunde zuvor zu weiter entfernten Objekten gehörten, gehen zu dem Objekt über, wodurch eine große Abstandsreduzierung gegeben ist. Somit wird der Bypass für diese Scanwinkel aktiv und es tritt keine Verzögerung für diese Scanwinkel auf, so dass von einem Scan zum nächsten Scan eine erhebliche Abstandsreduzierung des Objekts angezeigt wird. Dieses tritt an der linken und rechten Objektgrenze auf. Im Anhang B ist zur Verdeutlichung das zweidimensionale Array des History-MA-Filter für zwei nacheinander aufgenommene Scanrunden dargestellt, die die Funktionsweise des Bypasses deutlich machen.

Der beschriebene Bypass hat zur Folge, dass alle Messungen für die Scanwinkel, die zu dem Objekt neu hinzugekommen sind, auf den aktuellen Wert gesetzt werden und somit keine zeitliche Verzögerung haben. Bei schmalen Objekten ist dadurch der kürzeste Punkt des Objekts gleich dem rechten oder linken Punkt. Die Folge sind Störungen des geometrischen Modells für die relative Positionsbestimmung zum Hindernis. Die schematische Form des Objekts, die sich aus dem History-MA-Filter ergibt, ist in Abb. 13 dargestellt. Eine Geschwindigkeit von $v_m = 0 \text{ m/s}$ hat eine Erkennung der Hindernisfront als Linie zur Folge und die Objekterkennung stellt somit der Weiterverarbeitung das Objekt in Form der Punkte L , K_1 und R zur Verfügung. Für den Fall, dass $v_m > 0 \text{ m/s}$ ist, wird der Bypass an der linken und/oder rechten Objektgrenze aktiv. Die Häufigkeit ist dabei von der Relativgeschwindigkeit, der Winkelauflösung des Laserscanners und der Größe des Hindernisobjekts abhängig. In Abb. 13 treten links und rechts Objektanomalien auf, d. h. für diese beiden Scanwinkel gibt es keine Verzögerung (blaue gepunktete Linie). Zu diesem Zeitpunkt ist der kürzeste Objektpunkt K_2 und die Längen der Polarkoordinaten von den Punkten L und R haben sich verkürzt.

Mit weiteren Messungen relativieren sich die Ergebniswerte für die beiden äußersten Winkel, d. h. die Objektform geht wieder in eine Linie über und es wird der korrekte Punkt für den kürzesten Abstand bestimmt. Dieses ist solange der Fall, bis erneut Objektanomalien auftreten. Durch diese Variation der Punkte K , L und R des Objekts ist die Wiedererkennung des Objekts als solches nicht mehr gegeben. Aus diesem Grund findet kontinuierlich eine Neueinstufung des Objekts statt.

Die Konsequenz ist, dass das Hindernisobjekt eine neue Identifikationsnummer bekommt, der Ausweichassistent aber mit der alten Identifikationsnummer weiterarbeitet, d. h. die Objektzuordnung der Objekterkennung wird gestört, was wiederum ein nicht korrektes Umgebungsmodell zur Folge hat. In Kapitel 6 wird ein räumliches Mittelwertfilter entwickelt, das keinen Bypass besitzt und folglich keine Objektanomalien erzeugt.

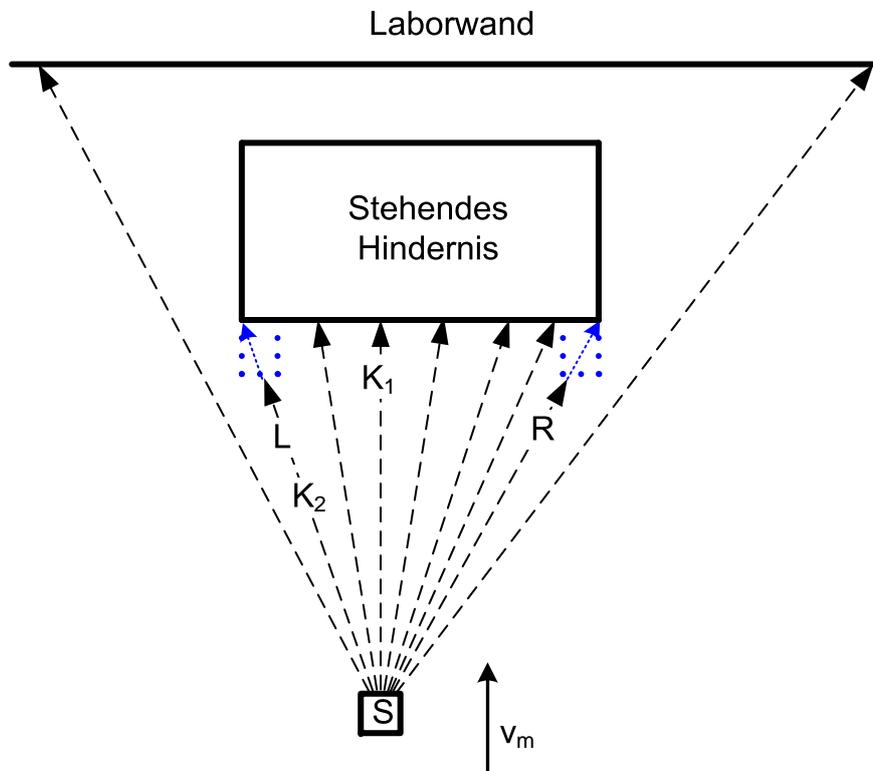


Abb. 13: Schematisches Objekt des History-MA-Filters

Das implementierte History-MA-Filter des SCV in [6] und [5] hat eine Filtertiefe von $N = 10$ und misst 181 Scanwinkel ab.

4.3 Die Aufprallerkennung des SCV

Der Ausweichassistent muss dem Objekt mit der geringsten Aufprallzeit ausweichen, dieses wird in der Aufprallerkennung aus den Umgebungsobjekten von der Objekterkennung bestimmt. Zunächst ermittelt das Modul für jedes Objekt, ob eine Kollision zwischen dem Objekt und dem SCV, in Abhängigkeit der Objektgeschwindigkeit und der Eigenbewegung des SCV, bevorsteht. Anschließend wird das Objekt mit der geringsten Aufprallzeit bestimmt, da für dieses der Ausweichstatus relevant ist.

Die Aufpralldaten beinhalten:

- die Objekt-ID
- die Aufprallzeit
- die Geschwindigkeitsvektoren in x- und y-Richtung des Objekts
- die Abstände der Punkte L , K und R in y-Richtung

4.4 Die Simulation des Umgebungsmodells

Für die Entwicklung von Fahrerassistenzsystemen wurde in [6] eine Simulation auf dem Koordinierungsrechner entwickelt, um dabei von der Hardwareplattform unabhängig zu sein. Es wird die Laserscannermessung, der Lenkwinkel und die Geschwindigkeit des SCV simuliert. Durch die Simulation ist der Ausweichassistent allein auf dem Koordinierungsrechner lauffähig, d. h. ohne die weitere Hardwareplattform des SCV.

Für die Simulation eines Objekts in der Umgebung des SCV muss der Entwickler bzw. der Tester die vier Eckpunkte, die Geschwindigkeitsvektoren und den Achswinkel des Objekts im Scan-Simulator definieren. Das Software-Modul berechnet die Positionsänderung des Objekts mit der Objektgeschwindigkeit und der Eigenbewegung des SCV während der Simulation. Mit diesem Simulationsobjekt wird für jeden Scanwinkel des Messbereiches eine Messung simuliert. Für die Scanwinkel des Objekts werden die Messungen auf den jeweiligen berechneten Abstand gesetzt. Bei Scanwinkeln, die nicht das Objekts repräsentieren, wird der Abstandswert auf einen Maximalwert gesetzt. Diese simulierten Messungen werden wie im Normalbetrieb an die Message Queue für das Modul *LD_Comm* gesendet.

Der Lenkwinkel und die Geschwindigkeit des SCV werden in dem Steuerautomaten simuliert. Der Automat speichert die beide Fahreigenschaften bei einer Sollwertänderungen, so dass sie bei einer Istwertabfrage zur Verfügung stehen. Daraus lässt sich ableiten, dass es in der Simulation zu keiner Differenz zwischen Soll- und Istwert kommt.

4.5 Die Visualisierung der gefilterten Laserscannermessung und des Umgebungsmodells

Eine grafische Oberfläche auf dem Koordinierungsrechner stellt die Visualisierung der Entfernungswerte der gefilterten Laserscannermessung und die Objekte des Umgebungsmodells zur Verfügung. Dieses erlaubt zum Einen die Entwicklung des Ausweichassistenten in Zusammenhang mit der Simulation ohne die komplette Hardwareplattform des SCV und zum Anderen die schnelle Fehlererkennung in der Entwicklung von Fahrerassistenzsystemen (vgl. Abb. 14).

Die Visualisierung beinhaltet folgende Komponenten:

- Koordinatensystem, Ursprung an der Position des Hinterrades P des SCV
- Kontur des SCV und Position des Laserscanners
- Abstandskreise, Radius nimmt jeweils um 0,5 m zu
- gefilterte Messwerte des Laserscanners als Punkte
- Objekte im Scanbereich von 0° bis 180° mit:
 - Identifikationsnummer
 - Strecke L bis K in rot
 - Strecke K bis R in grün
 - Geschwindigkeitsvektor in blau

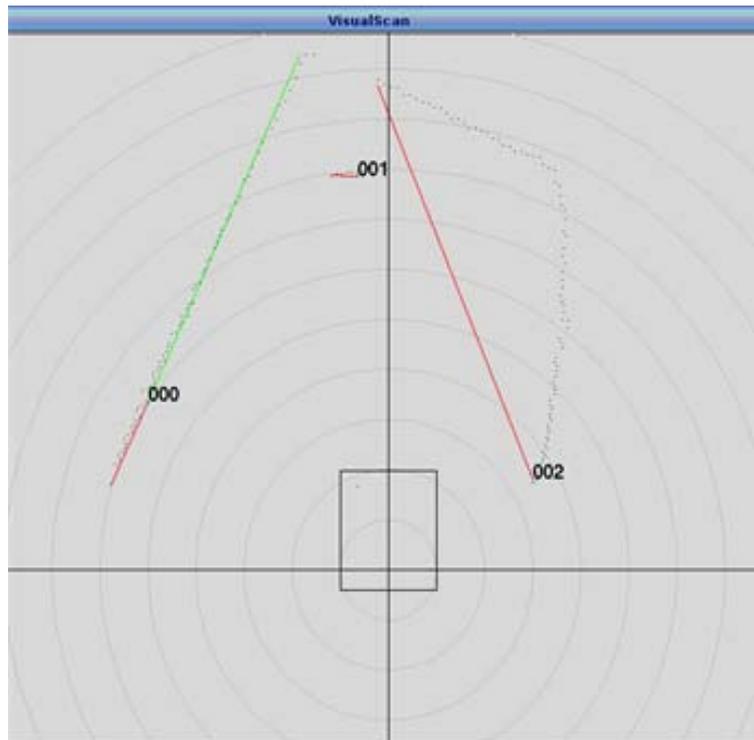


Abb. 14: Visualisierung des Umgebungsmodells (aus [6])

Der eigenständige Thread zur Visualisierung der Umgebungsinformationen wird mit einer Frequenz von 20 Hz aufgerufen, das entspricht der Scanfrequenz des Laserscanners. Die Objekterkennung aktualisiert zyklisch die gefilterten Entfernungswerte und die Objektinformationen mit *set-Funktionen* im Visualisierungs-Thread. Weiterführende Details sind in [6] dokumentiert.

5 Kinematisches Einspurfahrzeugmodell des SCV

In diesem Kapitel wird das kinematische Fahrzeugmodell des SCV vorgestellt, das ein Einspurfahrzeug ist. Für die Fahrzeugsimulation und den Regelkreisentwurf ist dieses Modell erforderlich. An Vorder- und Hinterrad sind Antriebs- und Lenkmotoren installiert. Auf der linken und rechten Seite befindet sich jeweils ein Stützrad. Das Ziel dieser Masterarbeit ist, wie bereits in Kapitel 3 beschrieben, ein Fahrerassistenzsystem für den Automobilbereich zu entwickeln. Folglich besitzt das hier vorgestellte Fahrzeugmodell ein lenkbares und angetriebenes Vorderrad M und ein starres Hinterrad P .

Für die Herleitung der Systemgleichungen wird die Fahrt des SCV auf einer Kreisbahn betrachtet (vgl. Abb. 15). Das Vorderrad ist mit dem Lenkwinkel α nach links eingeschlagen und bewegt sich mit der Geschwindigkeit v_m . Der aktuelle Achswinkel des Fahrzeug ist θ und der Abstand zwischen Vorder- und Hinterrad beträgt $L = 0,8 \text{ m}$. Die Kreisbahnradien r_m und r_p liegen jeweils rechtwinklig zu dem Rad und schneiden sich in dem Bezugspunkt Z .

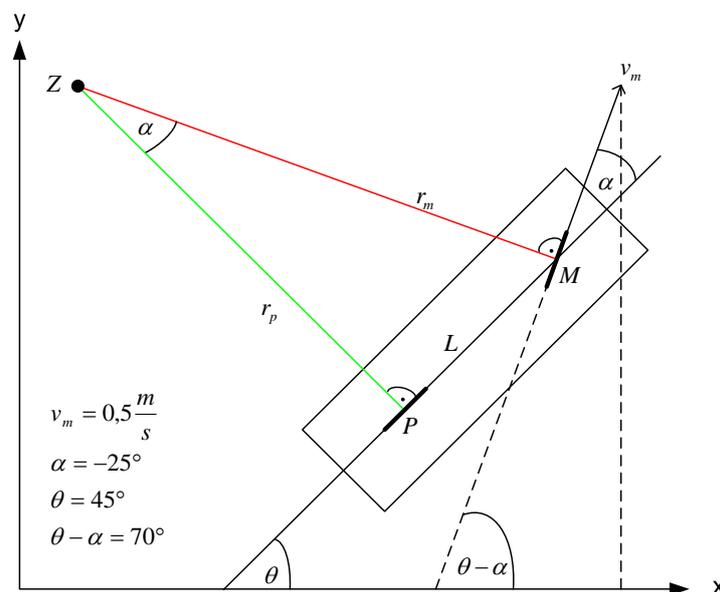


Abb. 15: Skizze: Kinematisches Einspurfahrzeugmodell des SCV

Die Gültigkeitsbereiche der Systemvariablen sind in der Tabelle 1 dokumentiert. Die Bereiche für die Geschwindigkeit und den Lenkwinkel resultieren aus den Fahrzeugeigenschaften. Der Bereich für den Achswinkel ist entsprechend der Berechnungsweise, die folgend vorgestellt wird, gewählt.

Variable	Gültigkeitsbereich
v_m	$-1,4 \text{ m/s} \leq v_m \leq 1,4 \text{ m/s}$
α	$-90^\circ \leq \alpha \leq 90^\circ$
θ	$-180^\circ < \theta \leq 180^\circ$

Tabelle 1: Gültigkeitsbereiche der Systemvariablen des SCV

Die Bedeutung und die Abkürzungen der Systemparameter Vorderradgeschwindigkeit v_m und Lenkwinkel α sind in Tabelle 2 dokumentiert. Dabei ist das Vorzeichen des Lenkwinkels umgekehrt zur Winkelgeometrie des Einheitskreises. Dieses ist begründet durch die Realisierung des Leitstandes und des CAN-Nachrichtenprotokolls, die dieser Masterarbeit vorangegangen sind.

Parameter	Bedeutung
$v_m \geq 0$	Vorwärtsfahrt, positive Geschwindigkeitswerte vom Antriebsmotor
$v_m < 0$	Rückwärtsfahrt, negative Geschwindigkeitswerte vom Antriebsmotor
$\alpha \geq 0$	Rechts eingelenkt, positiver Lenkwinkel vom Lenkmotor
$\alpha < 0$	Links eingelenkt, negativer Lenkwinkel vom Lenkmotor

Tabelle 2: Interpretation der Systemparameter des SCV

5.1 Herleitung einer kinematischen Fahrzeugbeschreibung

Für die Herleitung der Fahrzeugkinematik des SCV wird zunächst die Gleichung für den Achswinkel bestimmt. Mit dem Achswinkel lässt sich anschließend die Position des Vorderrades M in x- und y-Richtung berechnen. Aus der Vorderradgeschwindigkeit v_m und dem Lenkwinkel α wird die Hinterradgeschwindigkeit v_p berechnet, die neben dem Lenkwinkel α für die Berechnung der Hinterradposition genutzt wird.

Berechnung des Winkels der Fahrzeuglängsachse

Zur Herleitung der Gl. 6 für den aktuellen Achswinkel θ wird die Fahrt auf einer Kreisbahn mit einer konstanten Geschwindigkeit [10] betrachtet. In diesem Fall gilt, Winkelgeschwindigkeit $\omega = v/r$. Die Winkelgeschwindigkeit von Vorder- und Hinterrad ist dabei gleich, also gilt Gl. 2. Aus Abb. 15 ist Gl. 3 zu entnehmen. Der Winkel zwischen r_m und r_p im Punkt Z ist gleich dem Lenkwinkel α , da es sich um paarweise senkrechte Schenkel handelt (vgl. [11]).

$$\frac{d\theta}{dt} = \omega_0 = \frac{v_m}{r_m} = \frac{v_p}{r_p} \quad (2) \quad \frac{L}{r_m} = \sin(\alpha) \Rightarrow r_m = \frac{L}{\sin(\alpha)} \quad (3)$$

Durch Einsetzen der Gl. 3 in Gl. 2 ergibt sich Gl. 4. Bei anschließender Integration resultiert Gl. 5, folglich ist der Achswinkel θ mit dem Radabstand L , der Vorderradgeschwindigkeit und dem Lenkwinkel α zu berechnen.

$$\frac{d\theta}{dt} = \frac{v_m}{L} \cdot \sin(\alpha) \quad (4) \quad \theta(t) = \frac{1}{L} \cdot \int v_m(t) \cdot \sin(\alpha) d(t) \quad (5)$$

Abschließend ergibt sich mit der Betrachtung des Vorzeichens von α die Gl. 6. Die Betrachtung ist notwendig, da das Resultat bei Sinus von dem Vorzeichen des Winkels abhängig ist.

$$\theta(t) = \frac{1}{L} \cdot \int v_m(t) \cdot \sin(-\alpha) d(t) \quad (6)$$

Berechnung der Vorderradposition

Die Berechnungsgleichungen für die Vorderradposition in x-Richtung x_m und in y-Richtung y_m im Koordinatensystem lassen sich aus Abb. 15 herleiten. Für $v_m \neq 0$ gilt, die Geschwindigkeit

in x-Richtung ist die Vorderradgeschwindigkeit mal der x-Komponente, sprich $\cos(\theta - \alpha)$. Das $\theta - \alpha$ ergibt sich aus der Vorzeichenbetrachtung des Lenkwinkels. Durch die Integration der Gl. für die Geschwindigkeit in x-Richtung ergibt sich die Gl. 7 für die Position in x-Richtung. Entsprechendes gilt für die Geschwindigkeit und Position in y-Richtung, wobei der Kosinus durch den Sinus ersetzt wird (vgl. Gl. 8).

$$\frac{dx_m}{dt} = v_m(t) \cdot \cos(\theta - \alpha) \Rightarrow x_m = \int v_m(t) \cdot \cos(\theta - \alpha) d(t) \quad (7)$$

$$\frac{dy_m}{dt} = v_m(t) \cdot \sin(\theta - \alpha) \Rightarrow y_m = \int v_m(t) \cdot \sin(\theta - \alpha) d(t) \quad (8)$$

Berechnung der Hinterradgeschwindigkeit

Für die Entwicklung der Hinterradgeschwindigkeit v_p wird im ersten Schritt Gl. 2 als Ausgangspunkt verwendet. Demnach gilt Gl. 9 und 10.

$$\omega_0 = \frac{v_m}{r_m} \quad (9) \quad \omega_0 = \frac{v_p}{r_p} \Leftrightarrow v_p = \omega_0 \cdot r_p \quad (10)$$

Durch einsetzen von Gl. 9 in Gl. 10 ergibt sich Gl. 11. Im zweiten Schritt wird die Beziehung in Gl. 12 aus Abb. 15 abgelesen. Eine Vernachlässigung des Vorzeichens ist an dieser Stelle zulässig, da bei Verwendung des Kosinus das Resultat vom Vorzeichen unabhängig ist.

$$v_p = v_m \cdot \frac{r_p}{r_m} \quad (11) \quad \cos(\alpha) = \frac{r_p}{r_m} \quad (12)$$

Abschließend werden die beiden Entwicklungsschritte zusammengeführt, d.h. durch Einsetzen von Gl. 11 in Gl. 12 ergibt sich die Berechnungsgleichung 13 für die Hinterradgeschwindigkeit.

$$v_p = v_m \cdot \cos(\alpha) \quad (13)$$

Berechnung der Hinterradposition

Die Position für das Hinterrad in x-Richtung x_p und in y-Richtung y_p folgt aus den Gln. 7 und 8. Die Vorderradgeschwindigkeit wird durch die Hinterradgeschwindigkeit aus Gl. 13 ersetzt und dem Kosinus bzw. dem Sinus wird für die Richtungskomponenten der Achswinkel übergeben, statt der Differenz aus Achswinkel und Lenkwinkel, da es sich um ein starres Hinterrad handelt.

$$\frac{dx_p}{dt} = v_m(t) \cdot \cos(\alpha) \cdot \cos(\theta) \Rightarrow x_p = \int v_m(t) \cdot \cos(\alpha) \cdot \cos(\theta) d(t) \quad (14)$$

$$\frac{dy_p}{dt} = v_m(t) \cdot \cos(\alpha) \cdot \sin(\theta) \Rightarrow y_p = \int v_m(t) \cdot \cos(\alpha) \cdot \sin(\theta) d(t) \quad (15)$$

Bestimmung der Lenkverzögerung

Für eine genauere Systembeschreibung des SCV wird in diesem Abschnitt die Lenkverzögerung ermittelt. Dabei handelt es sich um ein Verzögerungsglied erster Ordnung (PT_1), die Übertragungsfunktion ist in Gl. 16 angegeben. Der Endwert K_P ist gleich 1, d.h. nach der Übergangszeit von $3 \cdot T_1$ wird der Lenkwinkel durch das PT_1 -Glied praktisch nicht mehr ver-

ändert, da die Sprungantwort auf 95 % des Endwertes angestiegen ist [12]. Zur Bestimmung der Zeitkonstante T_1 werden verschiedene Messreihen am SCV durchgeführt.

Für diese Messreihen muss der Soll-Lenkwinkel von einer CAN-Nachricht zur nächsten auf einen definierten Winkel gesetzt werden, um so die reine Verzögerung von dem Lenkmotor zu ermitteln. Dieses ist nicht mit dem Lenkrad machbar, da in dem Fall schon durch die Lenkbewegung eine Verzögerung auftritt, d. h. es wurde dafür eine spezielle Software verwendet. Anwendung hierfür findet eine modifizierte Software des Einparkassistenten [7], mit der per Knopfdruck am Lenkrad der Soll-Lenkwinkel sofort auf einen definierten Winkel gesetzt wird.

$$G(s) = \frac{K_P}{(T_1 s + 1)} \quad (16)$$

In dieser Masterarbeit wird für die weiteren Betrachtungen von einer Lenkverzögerung ausgegangen, welche aus den Messungen für den Soll-Lenkwinkel-Sprung von 0° zu 30° resultiert. Dieser Sprung liegt in dem Bereich, der häufig bei den Ausweichsimulationen vorkommt. Die Bestimmung der Zeitkonstante T_1 erfolgt über die Ermittlung der Zeitdauer, welche erforderlich ist, um 63 % des Endwertes zu erreichen. Folglich wird in diesem konkreten Fall der Soll-Lenkwinkel von 30° vorgegeben und die Zeit abgelesen, sobald der Lenkwinkel ca. $18,9^\circ$ beträgt. In der Tabelle 3 sind die sieben durchgeführten Messungen aufgelistet und abschließend der berechnete Durchschnittswert angegeben, der in den weiteren Berechnungen und Simulationen genutzt wird.

Die Lenkverzögerung für den Soll-Lenkwinkel-Sprung von 0° auf 30° in Form eines PT_1 -Glieds ist in Abb. 16 dargestellt.

Messung Nr.	t [ms]
1	245, 220
2	273, 920
3	275, 470
4	226, 790
5	238, 640
6	239, 960
7	241, 290
Ø	248, 756

Tabelle 3: Messungen der Zeiten für die Bestimmung der Lenkverzögerung

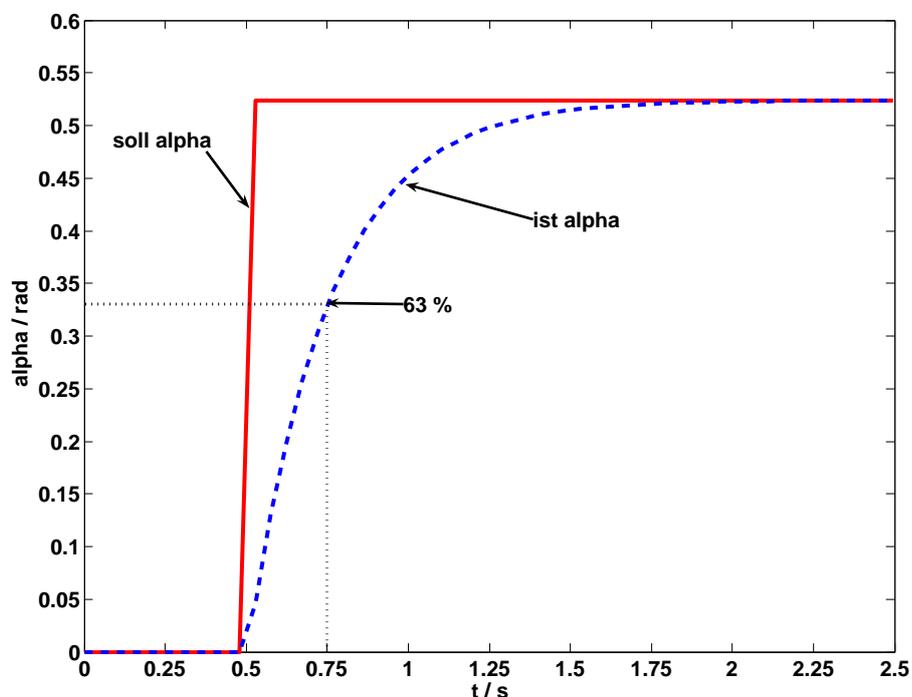


Abb. 16: Sprungantwort der Lenkverzögerung des SCV

5.2 Simulinkmodell der SCV-Kinematik

Die Fahrzeugbewegungen des SCV lassen sich mit den Gleichungen des kinematischen Fahrzeugmodells mit Matlab simulieren. Dazu werden die Gleichungen zu einem Blockschaltbild in Simulink zusammengesetzt (vgl. Abb. 68 in Anhang C). Die Simulationskurven zu dem Simulink Modell in Abb. 68 sind in den folgenden Abbildungen dargestellt. Der Lenkwinkel α beträgt -30° und die Vorderradgeschwindigkeit ist $v_m = 0,5 \text{ m/s}$. In Abb. 17a sind die x-Koordinaten der Räder über die Zeit aufgetragen. Es ergibt sich eine sinusförmige Kurve, ebenso bei der y-Koordinate der Räder über die Zeit in Abb. 17b. Wird beispielsweise das Vorderrad stärker eingeschlagen, so verringert sich die Amplitude, während eine Geschwindigkeitserhöhung eine Reduzierung der Schwingdauer zur Folge hat. In Abb. 18 sind die y-Koordinaten über die x-Koordinaten der Räder aufgetragen, auf diese Weise ergibt sich eine Kreisbahn.

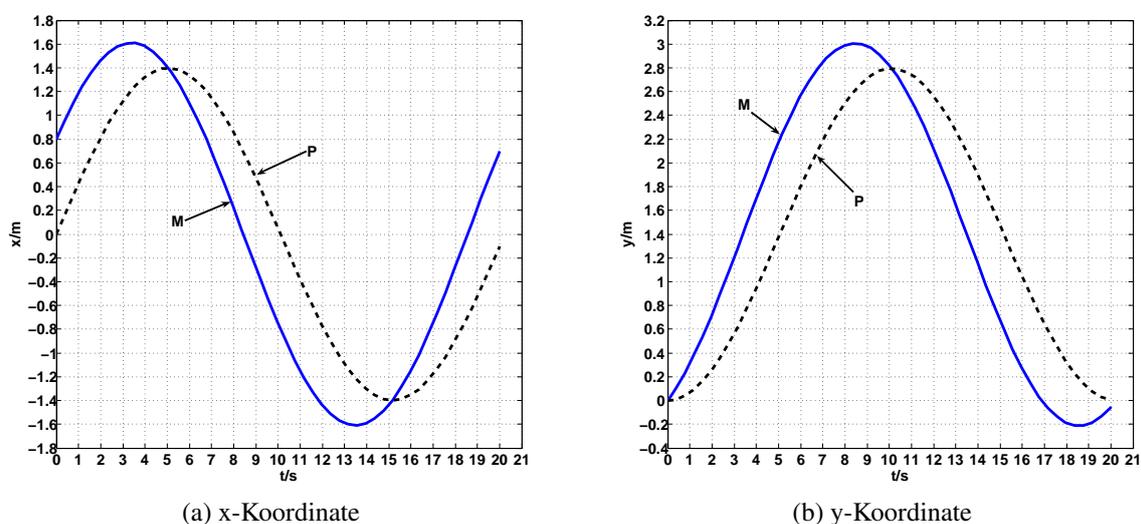


Abb. 17: Kinematisches Modell des SCV über die Zeit

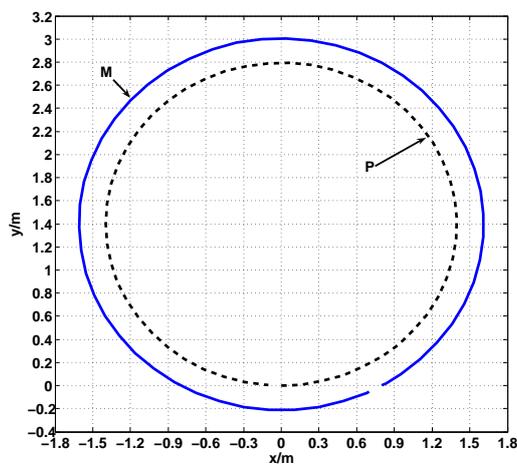


Abb. 18: Kinematisches Modell des SCV, y über x

5.3 Diskrete Systemgleichungen des kinematischen Modells SCV

Für die zeitdiskreten Simulationen der Systemeigenschaften Achswinkel und Radpositionen werden in diesem Abschnitt die zeitdiskreten Systemgleichungen mit der Trapezintegration hergeleitet. Für den Achswinkel θ wird von der Gl. 6 (17) ausgegangen. Die gesamte Fläche unter der Geraden in Abb. 19 entspricht dabei $\theta(k+1)$ in der Gl. 18, wobei Δt das Abtastintervall ist. Folgend sind die Flächen aus Abb. 19 und die entsprechenden Summanden aus Gl. 18 aufgeführt:

- (1) : $\theta(k)$
- (2) : $\frac{1}{L} \cdot \Delta t \cdot v_m(k) \cdot \sin(-\alpha(k))$
- (3) : $\frac{1}{2} \Delta t \cdot \{v_m(k+1) \cdot \sin(-\alpha(k+1)) - v_m(k) \cdot \sin(-\alpha(k))\}$

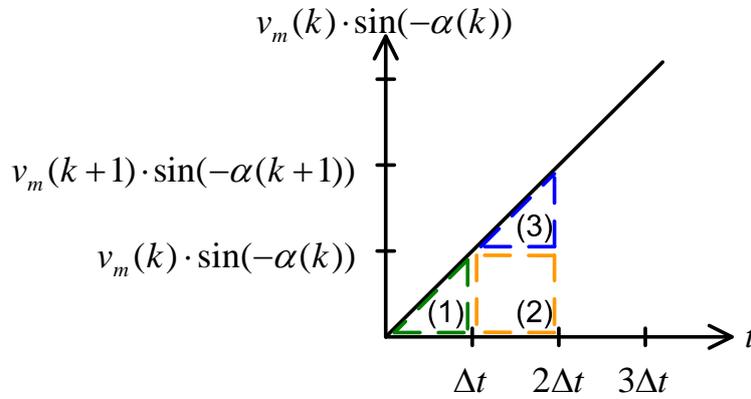


Abb. 19: Trapezintegration für den Achswinkel

$$\theta(t) = \frac{1}{L} \cdot \int v_m(t) \cdot \sin(-\alpha) d(t) \quad (17)$$

$$\begin{aligned} \theta(k+1) &= \theta(k) \\ &+ \frac{1}{L} \cdot [\Delta t \cdot v_m(k) \cdot \sin(-\alpha(k)) \\ &+ \frac{1}{2} \Delta t \cdot \{v_m(k+1) \cdot \sin(-\alpha(k+1)) \\ &- v_m(k) \cdot \sin(-\alpha(k))\}] \\ &= \theta(k) \\ &+ \frac{1}{L} \cdot [\frac{1}{2} \Delta t \cdot v_m(k) \cdot \sin(-\alpha(k)) \\ &+ \frac{1}{2} \Delta t \cdot v_m(k+1) \cdot \sin(-\alpha(k+1))] \end{aligned} \quad (18)$$

Die weiteren Systemeigenschaften des SCV sind entsprechend herzuleiten. Angegeben werden nachfolgend die Gleichungen für das Vorderrad M in x-Richtung Gl. 19, in y-Richtung Gl. 20 und für das Hinterrad P in x-Richtung Gl. 21 und in y-Richtung Gl. 22.

$$\begin{aligned} x_m(k+1) &= x_m(k) \\ &+ \frac{1}{2} \cdot \Delta t \cdot v_m(k) \cdot \cos(\theta(k) - \alpha(k)) \end{aligned}$$

$$+ \frac{1}{2} \cdot \Delta t \cdot v_m(k+1) \cdot \cos(\theta(k+1) - \alpha(k+1)) \quad (19)$$

$$\begin{aligned} y_m(k+1) &= y_m(k) \\ &+ \frac{1}{2} \cdot \Delta t \cdot v_m(k) \cdot \sin(\theta(k) - \alpha(k)) \\ &+ \frac{1}{2} \cdot \Delta t \cdot v_m(k+1) \cdot \sin(\theta(k+1) - \alpha(k+1)) \end{aligned} \quad (20)$$

$$\begin{aligned} x_p(k+1) &= x_p(k) \\ &+ \frac{1}{2} \Delta t \cdot v_m(k) \cdot \cos(\alpha(k)) \cdot \cos(\theta(k)) \\ &+ \frac{1}{2} \Delta t \cdot v_m(k+1) \cdot \cos(\alpha(k+1)) \cdot \cos(\theta(k+1)) \end{aligned} \quad (21)$$

$$\begin{aligned} y_p(k+1) &= y_p(k) \\ &+ \frac{1}{2} \Delta t \cdot v_m(k) \cdot \cos(\alpha(k)) \cdot \sin(\theta(k)) \\ &+ \frac{1}{2} \Delta t \cdot v_m(k+1) \cdot \cos(\alpha(k+1)) \cdot \sin(\theta(k+1)) \end{aligned} \quad (22)$$

Für die zeitdiskrete Simulation des Ausweichvorgangs ist die diskrete Übertragungsfunktion der Lenkverzögerung erforderlich. Dazu wird in der zeitkontinuierlichen Funktion die Laplace-Transformierte s mit der Tustin Formel $s \approx \frac{2}{T} \frac{z-1}{z+1}$ ersetzt und anschließend zusammengefasst (vgl. Gl. 23 und Gl. 23). Dadurch wird das Verzögerungsglied in den z -Bereich transformiert, für weitere Details vgl. Kapitel 7.5 und [13].

$$\begin{aligned} G(s) &= \frac{K_P}{(T_1 s + 1)} \\ &\approx \frac{1}{T_1 \cdot \frac{2}{T} \cdot \frac{z-1}{z+1} + 1} \\ &\approx \frac{T(z+1)}{T_1 \cdot 2 \cdot (z-1) + (z+1) \cdot T} \\ &\approx \frac{Tz + T}{2 \cdot T_1 z - 2 \cdot T_1 + Tz + T} \\ &\approx \frac{Tz + T}{(2T_1 + T) \cdot z + (T - 2T_1)} \end{aligned} \quad (23)$$

5.4 Simulationen des diskreten Einspurfahrzeugmodells

Für die Verifikation der diskreten Systemgleichungen des SCV dienen die Matlab Simulationen in diesem Abschnitt (vgl. Anhang D Listing 1). Es gelten die Voraussetzungen aus der folgenden Auflistung:

- Der Lenkwinkel α ist konstant.
- Der Startwert der Fahrzeugachse θ wird zu Simulationsbeginn gesetzt.
- Die Geschwindigkeit des Vorderrades v_m ist konstant.

- Die Abtastrate ist $\Delta t = 0,05 \text{ s}$, das entspricht 20 Hz und deshalb der Senderate der CAN-Nachrichten für die Geschwindigkeit und des Winkels im SCV.
- Die Koordinaten des Vorderrades M und des Hinterrades P werden zu Simulationsbeginn gesetzt.

Dabei gilt:

$$M = (x - \text{Koordinate} | y - \text{Koordinate})$$

$$P = (x - \text{Koordinate} | y - \text{Koordinate})$$

- Der Abstand der Räder ist $L = 0,8 \text{ m}$

In der ersten Simulation ist der Lenkwinkel $\alpha = -10^\circ$, der Achswinkel θ am Anfang 0° , die Geschwindigkeit $v_m = 0,5 \text{ m/s}$ und die Startpositionen der Räder $M = (0,8|0)$ und $P = (0|0)$. Die resultierende Kreisbahn, die das SCV abfährt, ist in Abb. 20a für die x-Koordinaten und in Abb. 20b für die y-Koordinaten dargestellt, jeweils über die Zeit aufgetragen.

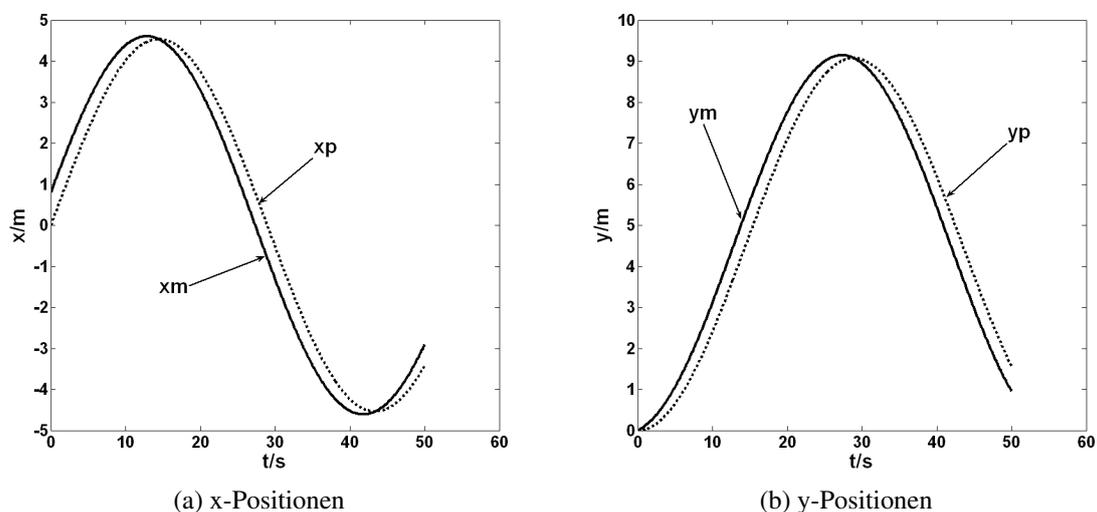


Abb. 20: Simulation Räder über die Zeit

In der zweiten Simulation ist der Lenkwinkel $\alpha = -30^\circ$, die weiteren Simulationsparameter sind unverändert. Das Simulationsergebnis ist in Abb. 21 dokumentiert, in dem Diagramm sind die y-Koordinaten von Vorder- und Hinterrad über die jeweiligen x-Koordinaten aufgetragen. Es ergeben sich zwei Kreisbahnen, da das Hinterrad auf einer kleineren Kreisbahn als das Vorderrad fährt.

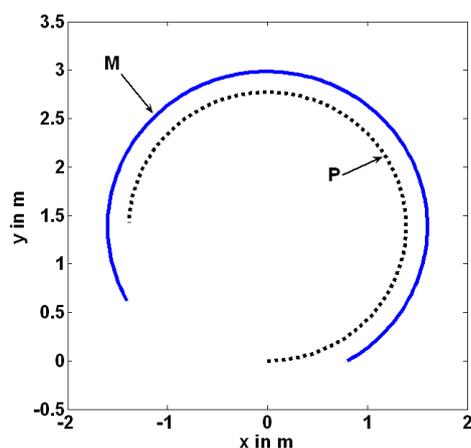


Abb. 21: Simulation y-Positionen der Räder über jeweilige x-Positionen

5.5 Vergleich von zeitkontinuierlichen mit zeitdiskreten Systemgleichungen

Zur Beurteilung der Näherung der zeitdiskreten Systemgleichungen werden die Simulationsergebnisse beider Gleichungssätze miteinander verglichen. Die Abtastrate, für die zeitdiskreten Systemgleichungen, ist $\Delta t = 0,1s$ und folglich größer, als die Senderate der CAN-Nachrichten für Geschwindigkeit und Winkel im SCV. Die Simulation in Abb. 22 hat die Simulationsparameter:

$$\begin{aligned} \alpha &= -30^\circ & M &= (0,8|0) \\ \theta &= 0^\circ & P &= (0|0) \\ L &= 0,8\text{ m} \end{aligned}$$

Die Kurven sind in diesem Fall praktisch äquivalent, d. h. die zeitdiskreten Systemgleichungen sind für die Simulation des Ausweichvorgangs geeignet.

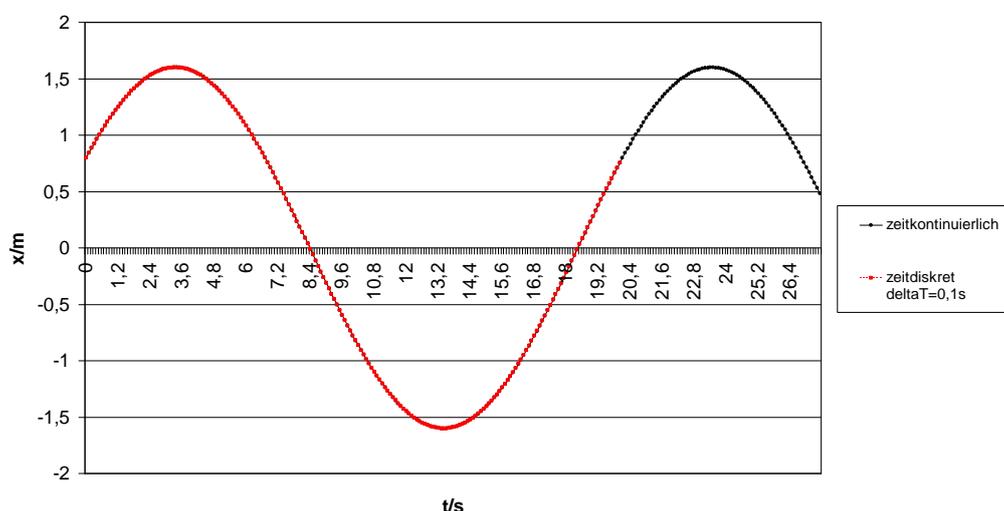


Abb. 22: Vergleich zeitkontinuierlicher / zeitdiskreter Systemgleichungen

5.6 Linearisierung der Systemgleichungen in ausgewählten Arbeitspunkten

In diesem Abschnitt werden die Systemgleichungen des SCV für verschiedene Arbeitspunkte linearisiert, da für lineare Systemgleichungen einfachere Verfahren zur Bestimmung der Reglerparameter im Regelkreis existieren. Für jeden Arbeitspunkt wird eine Definition von Lenkwinkel α , Achswinkel θ und Vorderradgeschwindigkeit v_m vorgenommen. Der Radabstand ist in allen Arbeitspunkten wegen des Fahrzeugaufbaus $L = 0,8\text{ m}$. Mit diesen resultierenden, linearisierten Systemgleichungen werden im Folgenden die Reglerparameter für das *symmetrische Optimum* berechnet, d. h. mit den Gleichungen für einen Arbeitspunkt wird ein Parametersatz für den Regler berechnet. Es ergeben sich daher für verschiedene Arbeitspunkte unterschiedliche Reglerparameter.

Die Gl. 4 (24) für die Winkelgeschwindigkeit des Achswinkels wird für einen konkreten Arbeitspunkt (AP) linearisiert [14] und der Koeffizient von $\Delta\alpha$ zu $k_\theta\alpha$ substituiert.

$$\dot{\theta} = \frac{v_m}{L} \cdot \sin(-\alpha) \quad (24)$$

$$\begin{aligned} \Rightarrow \Delta\dot{\theta} &\approx \frac{v_m}{L} \cdot \frac{d}{d\alpha} \sin(-\alpha) \Big|_{AP} \cdot \Delta\alpha \\ &\approx \frac{v_m}{L} \cdot \cos(-\alpha) \Big|_{AP} \cdot \Delta\alpha \\ &\approx k_{\theta}\alpha \cdot \Delta\alpha \end{aligned} \quad (25)$$

Zur Kontrolle der Gl. 25 wird die entsprechende Einheitengleichung in Gl. 26 betrachtet. Der Quotient aus Geschwindigkeit und Strecke ergibt die Winkelgeschwindigkeit, d. h. das Ergebnis aus Gl. 25 ist bestätigt.

$$[k_{\theta}\alpha] = \frac{m/s}{m} = \frac{1}{s} \quad (26)$$

Für die Linearisierung der Radgeschwindigkeiten werden die Additionstheoreme für Sinus und Kosinus sowie der *Satz des Pythagoras* verwendet (vgl. [11] bzw. Anhang E).

Die Linearisierung für die Geschwindigkeit des Vorderrades in x-Richtung ist in den Gln. 27 und 28 dokumentiert. Zunächst wird die Differenz $\theta - \alpha$ mit dem Additionstheorem für Kosinus umgeformt und anschließend die Gl. im AP linearisiert. Der Koeffizient von $\Delta\theta$ wird zu $k_x\theta$ und der von $\Delta\alpha$ zu $k_x\alpha$ substituiert.

$$\begin{aligned} \dot{x}_m &= v_m \cdot \cos(\theta - \alpha) \\ &= v_m \cdot [\cos(\theta) \cdot \cos(\alpha) + \sin(\theta) \cdot \sin(\alpha)] \\ \Rightarrow \Delta\dot{x}_m &\approx v_m \cdot \left\{ \frac{d}{d\theta} [\cos(\theta) \cdot \cos(\alpha) + \sin(\theta) \cdot \sin(\alpha)] \Big|_{AP} \cdot \Delta\theta \right. \\ &\quad \left. + \frac{d}{d\alpha} [\cos(\theta) \cdot \cos(\alpha) + \sin(\theta) \cdot \sin(\alpha)] \Big|_{AP} \cdot \Delta\alpha \right\} \\ &\approx v_m \cdot \left\{ [-\sin(\theta) \cdot \cos(\alpha) + \cos(\theta) \cdot \sin(\alpha)] \Big|_{AP} \cdot \Delta\theta \right. \\ &\quad \left. + [\cos(\theta) \cdot (-\sin(\alpha)) + \sin(\theta) \cdot \cos(\alpha)] \Big|_{AP} \cdot \Delta\alpha \right\} \\ &\approx k_x\theta \cdot \Delta\theta + k_x\alpha \cdot \Delta\alpha \end{aligned} \quad (27)$$

Die Einheitengleichung 29 bestätigt die Gl. 28, da beide Koeffizienten eine Geschwindigkeitskomponente haben.

$$[k_x\theta] = [k_x\alpha] = \frac{m}{s} \quad (29)$$

Die Geschwindigkeit in y-Richtung des Vorderrades wird entsprechend, wie in x-Richtung, linearisiert. Der Unterschied der beiden Gln. 30 und 31 besteht darin, dass statt der Winkelfunktion Kosinus, die Winkelfunktion Sinus mit dem dazugehörigen Additionstheorem Verwendung findet. Die Einheitengleichung 32 ist vergleichbar mit Gl. 29. Auf die Darstellung der Linearisierungen für das Hinterrad wird verzichtet, da diese ähnlich zu denen des Vorderrades ist.

$$\begin{aligned} \dot{y}_m &= v_m \cdot \sin(\theta - \alpha) \\ &= v_m \cdot [\sin(\theta) \cdot \cos(\alpha) - \cos(\theta) \cdot \sin(\alpha)] \\ \Rightarrow \Delta\dot{y}_m &\approx v_m \cdot \left\{ \frac{d}{d\theta} [\sin(\theta) \cdot \cos(\alpha) - \cos(\theta) \cdot \sin(\alpha)] \Big|_{AP} \cdot \Delta\theta \right. \end{aligned} \quad (30)$$

$$\begin{aligned}
& + \frac{d}{d\alpha} [\sin(\theta) \cdot \cos(\alpha) - \cos(\theta) \cdot \sin(\alpha)] \Big|_{AP} \cdot \Delta\alpha \} \\
\approx & v_m \cdot \{ [\cos(\theta) \cdot \cos(\alpha) - (-\sin(\theta)) \cdot \sin(\alpha)] \Big|_{AP} \cdot \Delta\theta \\
& + [\sin(\theta) \cdot (-\sin(\alpha)) + \cos(\theta) \cdot \cos(\alpha)] \Big|_{AP} \cdot \Delta\alpha \} \\
\approx & k_y \theta \cdot \Delta\theta + k_y \alpha \cdot \Delta\alpha
\end{aligned} \tag{31}$$

$$[k_y \theta] = [k_y \alpha] = \frac{m}{s} \tag{32}$$

In der Tabelle 4 sind die Ergebnisse für verschiedene exemplarische Arbeitspunkte aufgelistet. Die Vorderradgeschwindigkeit v_m ist für alle berechneten Arbeitspunkte 0,5 m/s. Ein konkretes Beispiel ist in Anhang F dokumentiert. Die Koeffizienten wurden mit dem Matlab-Code in Listing 2 in Anhang D berechnet.

Parameter	$k_{\theta\alpha}$	$k_x\theta$	$k_x\alpha$	$k_y\theta$	$k_y\alpha$
$\alpha = -30^\circ ; \theta = 0^\circ$	$0,541 \frac{1}{s}$	$-0,25 \frac{m}{s}$	$0,25 \frac{m}{s}$	$0,433 \frac{m}{s}$	$0,433 \frac{m}{s}$
$\alpha = -30^\circ ; \theta = 10^\circ$	$0,541 \frac{1}{s}$	$-0,321 \frac{m}{s}$	$0,321 \frac{m}{s}$	$0,383 \frac{m}{s}$	$0,47 \frac{m}{s}$
$\alpha = 10^\circ ; \theta = 20^\circ$	$0,616 \frac{1}{s}$	$-0,087 \frac{m}{s}$	$0,87 \frac{m}{s}$	$0,492 \frac{m}{s}$	$0,433 \frac{m}{s}$
$\alpha = -2,87^\circ ; \theta = 20,44^\circ$	$0,624 \frac{1}{s}$	$-0,198 \frac{m}{s}$	$0,198 \frac{m}{s}$	$0,459 \frac{m}{s}$	$0,477 \frac{m}{s}$
$\alpha = 9,65^\circ ; \theta = 14,57^\circ$	$0,616 \frac{1}{s}$	$-0,086 \frac{m}{s}$	$0,086 \frac{m}{s}$	$0,498 \frac{m}{s}$	$0,456 \frac{m}{s}$
$\alpha = 4,23^\circ ; \theta = 1,66^\circ$	$0,623 \frac{1}{s}$	$0,022 \frac{m}{s}$	$-0,022 \frac{m}{s}$	$0,5 \frac{m}{s}$	$0,497 \frac{m}{s}$

Tabelle 4: Exemplarische Arbeitspunkte

6 Berechnung der Fahrzeugposition und des Achswinkels mit Laserscannermesswerten

Die Istwerte der Fahrzeugpositionsregelung sollen aus den Abstandsmesswerten des Laserscanners zum Hindernis abgeleitet werden. Das Ziel dieses Kapitels ist es, die Abstände zum Hindernis in x - und y -Richtung, sowie den Achswinkel aus der Laserscannermessung zu berechnen. Im ersten Abschnitt wird die Messwertfilterung mit einem räumlichen Mittelwertfilter vorgestellt und ein Vergleich zu dem History-MA-Filter in [6] und [5] bzw. 4.2 durchgeführt. Des Weiteren wird die relative Fahrzeugposition des SCV zum Hindernis in den Abschnitten eines Ausweichvorgangs mit speziellen geometrischen Modellen bestimmt. Die Klassifizierung der Abschnitte des Ausweichvorgangs *vor dem Hindernis*, *schräg vor dem Hindernis* bzw. *neben dem Hindernis* wird zuvor beschrieben. Der Achswinkel gilt als Kriterium für das Beenden des Ausweichassistenten. Sämtliche Szenarien und Gleichungen in diesem Kapitel sind für das Links-Ausweichen ausgelegt.

6.1 Räumliches Mittelwertfilter zur Unterdrückung des Messrauschens

Bei Sensoren in technischen Systemen, wie dem SCV, treten Messfehler in Form von Signalmessrauschen auf. Zur Unterdrückung des Rauschens in der Abstandsmessung wird ein räumliches Mittelwertfilter (RM-Filter) eingesetzt, das einen Mittelwert über räumlich benachbarte Messpunkte bildet. Der Laserscanner Sick LD-OEM1000 liefert die Messwerte mit einer Messabweichung von $\pm 2,5$ cm bei 20 ... 90% Reflektion ab 0,5 m (vgl. [15]). In der Masterarbeit [6] wurde durch verschiedene Entfernungsmessungen eine Standardabweichung im Bereich von 1,8 cm für das Vorgängermodell Sick LD-OEM ermittelt.

Das RM-Filter in dieser Masterarbeit hat eine räumliche Filtercharakteristik, da die Messwerte mit aufeinanderfolgenden Scanwinkeln innerhalb eines Scans gemittelt werden. Es stellt die gefilterten Daten in Form eines kompletten 180° Scans der nachgelagerten Objekterkennung zur Verfügung. Das Filter glättet Ausreißer in einem Scan, indem ein Fenster mit einer definierten Breite über den Scan verschoben wird. Das Fenster mit der ungeraden Fensterbreite N beinhaltet den aktuellen Messpunkt und die räumlichen Messpunkte links und rechts von diesem. N muss dabei ungerade sein, damit ein Wert mit einer mittleren Position vorliegt.

Der Scanpunkt für den aktuellen Winkel wird dem Mittelwert aus den räumlich benachbarten Messpunkten gleichgesetzt (vgl. Gl. 33 und Abb. 23). Das RM-Filter nutzt diese Rück- und Vorgriffstechnik für die Messwerte, da räumlich gemittelt wird. Die zeitliche Verzögerung des Filters mit $(N - 1)/2$ Messwerten innerhalb einer Scanrunde ist zu vernachlässigen, da der Objekterkennung immer ein kompletter Scan mit 180 Messwerten übergeben wird. Die ersten und letzten $(N - 1)/2$ Messwerte werden nicht gefiltert, da bei diesen die Rohdaten aus den räumlich benachbarten Messpunkten nicht die volle Fensterbreite des Filters füllen. Das ist aber tolerierbar, da der Assistent für ein Objekt, das sich 90° seitlich neben dem SCV befindet, keinen Ausweichvorgang startet.

$$y_n = \frac{1}{N} \cdot \sum_{i=-((N-1)/2)}^{(N-1)/2} x_{n+i} \quad (33)$$

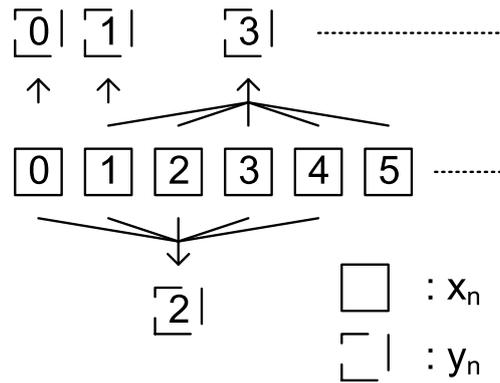


Abb. 23: Das RM-Filter zu Beginn einer Scanrunde

Für das Filter gilt: Je größer das Fenster, desto größer die Glättung. Die zwei Nachteile des Filters sind:

- verschliffene Ecken
- allein stehende Messpunkte in Form einer Rampe zwischen den Objekten

Zur Verdeutlichung des räumlichen Mittelwertfilters wurde ein Scan im Laborraum aufgezeichnet (vgl. Abb. 24). Die gefilterten Messwerte und die Objekte aus dem Umgebungsmodell des Scans sind in Abb. 25 visualisiert. Das Filter liefert durch die räumliche Mittelung allein stehende Messpunkte zwischen den Objekten, in Abb. 25 sind dies die Messpunkte zu den Scanwinkeln 74° bis 77° und 101° bis 104° . Die nachgelagerte Objekterkennung verwirft diese allein stehenden Messpunkte, da sie nicht zu einem Segment hinzugefügt werden. Dies hat eine Verkleinerung der Objekte zur Folge. Bei einer Filterbreite von $N = 5$, wie im Ausweichassistenten eingesetzt, gehören bei einem Objektübergang in den ungefilterten Messwerten zwei Punkte zu Objekt eins und zwei Punkte zu Objekt zwei. In dem Beispiel in Abb. 25, gehören die Abstandsmesspunkte zu den Scanwinkeln 74° , 75° , 103° und 104° zu der Laborwand (Objekt 210 und 002) und 76° , 77° , 101° und 102° zu dem Hindernisobjekt 199. Das Objekt 199 in der Abb. 25 ist somit durch das Filter vier Messpunkte kleiner.

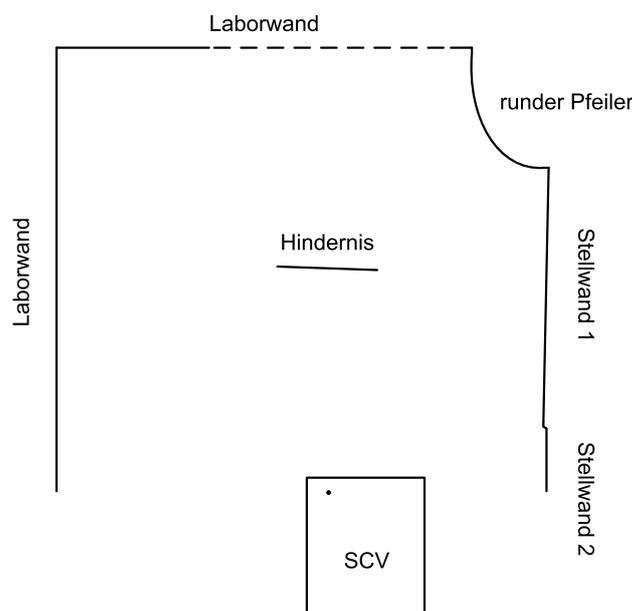


Abb. 24: Umgebungsplan des SCV mit Hindernisobjekt

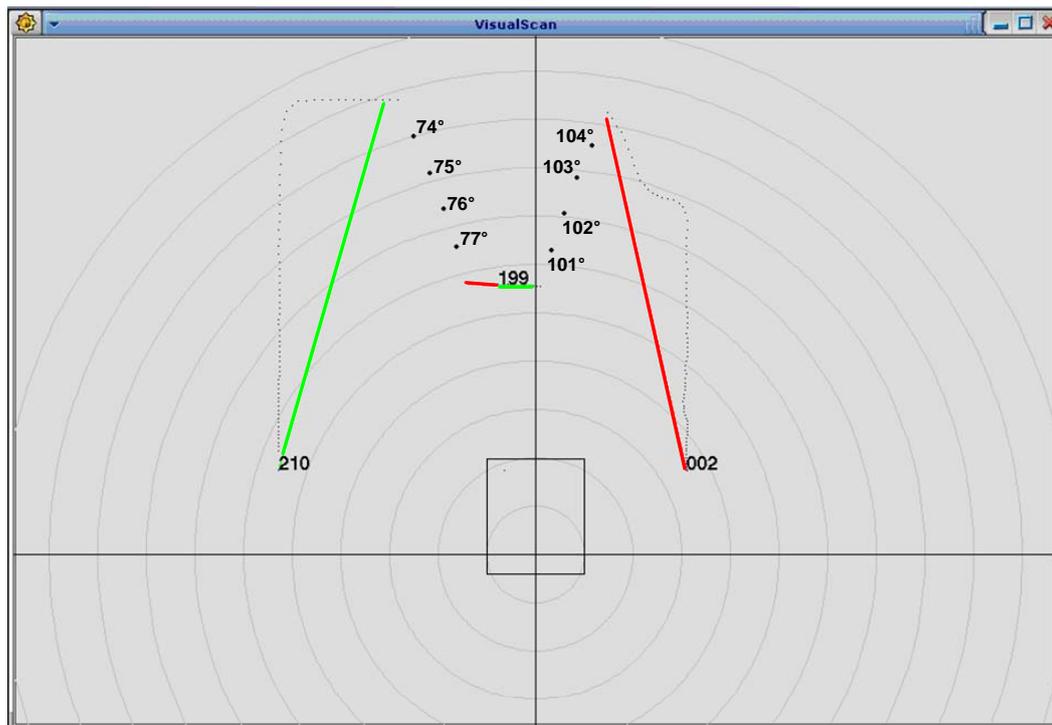


Abb. 25: Objektübergänge bei der Hinderniserkennung, rot Strecke L bis K und grün Strecke K bis R eines Objekts

In der Tabelle 5 sind die relevanten Messpunkte des Objektübergangs von der Wand zum Hindernis aus Abb. 25 aufgeführt. Die Rohdaten des Laserscanners sind als x_n und die Ergebnisse des räumlichen Mittelwertfilters als y_n angegeben. Das Ergebnis des Filters y_n ist bis zu dem Winkel 73° nicht von einem Messpunkt des Hindernisses beeinflusst. Bei den zwei folgenden Mittelungsergebnissen überwiegen die Messpunkte der Wand, werden aber vom Hindernis beeinflusst. Bei 76° , 77° kehrt sich dieses Verhältnis um und ab 78° setzt sich y_n ausschließlich aus den Messpunkten des Hindernisobjekts zusammen.

n	71	72	73	74	75	76	77	78	79	80
$x_n \left[\frac{m}{256} \right]$	1016	1006	1006	997	991	504	491	493	496	491
$y_n \left[\frac{m}{256} \right]$	1014	1008	1003	900	797	695	595	495	491	490

Tabelle 5: Die gegenseitige Beeinflussung der Objektgrenzen des räumlichen Mittelwertfilters, $N = 5$

Die Ergebnisse für einen Objektübergang mit den Filterbreiten $N = 3$, $N = 5$ und $N = 7$ sind im Anhang G dokumentiert. Es gilt:

- Je kleiner die Filterbreite,
 - desto geringer die Glättung des Filters.
 - desto weniger wird das Objekt verkleinert.
 - desto höher ist die Rauschempfindlichkeit.

Es wurde eine Filterbreite von $N = 5$ für den Ausweichassistenten gewählt, da diese einen Kompromiss zwischen Objektverkleinerung durch allein stehende Messpunkte und Rauschempfindlichkeit repräsentiert.

Das in dieser Masterarbeit verwendete RM-Filter, welches über räumlich benachbarte Messpunkte den Mittelwert bildet, hat die folgenden Vorteile gegenüber dem History-MA-Filter:

- Keine Verzögerung des Hindernisobjekts
- Kein Bypass, da nur die aktuelle Scanrunde betrachtet wird
- Keine Objektanomalien durch den Bypass
- $(N_{hma} - 1) \cdot n_{swi} \cdot d$ weniger Speicherplatz
 N_{hma} : Filterbreite „History-MA-Filter“
 n_{swi} : Anzahl Scanwinkel
 d : benötigter Speicherplatz für einen Abstandswert
- Kürzere Berechnungszeit, da keine Fallunterscheidung für den Bypass

Abschließend ist zu beachten, dass den verschliffenen Objektecken und dem resultierenden verkleinerten Hindernisobjekt durch einen entsprechenden Sicherheitsabstand entgegenwirkt werden kann.

Das RM-Filter in dieser Masterarbeit mittelt räumlich innerhalb eines Scans, d. h. die vorigen Scanrunden werden nicht betrachtet. Für diesen Anwendungsfall bietet sich das Kalman-Filter nicht an, da dieses zeitlich filtert, indem es optimale Schätzwerte für die Systemzustandsgrößen liefert und mit der aktuellen Messung vergleicht [16]. Das ist allerdings innerhalb einer Scanrunde nicht realisierbar, denn von dem Abstandswert des Scanwinkels n lässt sich nicht auf den Scanwinkel $n + 1$ schließen.

6.2 Bestimmung des Aufenthaltsbereichs des SCV während eines Ausweichvorgangs

In diesem Abschnitt wird vorgestellt, wie aus der Laserscannermessung bestimmt wird, in welchem Bereich sich das SCV während des Ausweichens gerade befindet. Im Kapitel 3 wurden die folgenden Bereiche schon diskutiert.

- *Vor dem Hindernis*
- *Schräg vor dem Hindernis*
- *Neben dem Hindernis*

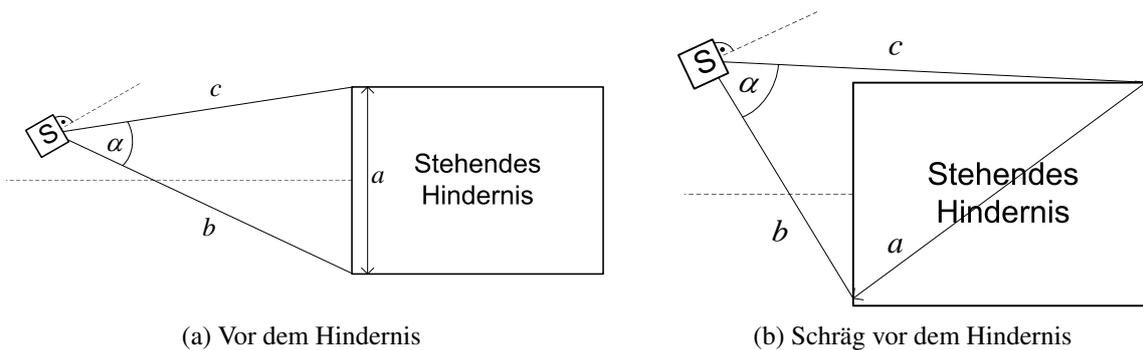
Das SCV bewegt sich während des Ausweichens stetig auf einer Ausweichkurve vorwärts. Folglich werden die Bereiche nacheinander durchquert und jedes Gebiet nur einmal befahren. Somit muss nach der Berechnung für die aktuelle Position eine Prüfung vorgenommen werden, ob die Bereichszuordnung noch korrekt ist. Trifft dieses nicht zu, so wird zum Nächsten gewechselt und die Berechnungen für diesen Bereich vorgenommen.

Zu Beginn des Ausweichvorgangs wird der Startbereich zwischen *vor dem Hindernis* und *schräg vor dem Hindernis* ermittelt. Ist die Differenz der x-Koordinaten der Objektpunkte L und R kleiner als eine definierte Tiefe, so befindet sich das SCV *vor dem Hindernis*. Dabei gilt die Annahme, dass es sich um ein rechteckiges Objekt handelt und der Achswinkel des SCV zum Hindernis 0° beträgt. Im anderen Fall befindet sich das SCV *schräg vor dem Hindernis*.

Es werden die Kriterien für die Bereichswechsel vorgestellt, die nach einer Berechnung geprüft werden, wobei diese für jedes Gebiet unterschiedlich sind.

Vor dem Hindernis \Rightarrow *Schräg vor dem Hindernis*

Solange die Differenz der Breite des Hindernisses a aus der vorigen und der aktuellen Messung kleiner als eine definierte Toleranz für die Hindernisbreite ist, befindet sich das SCV *vor dem Hindernis*. Ist a größer als die Toleranz, wird in den Bereich *schräg vor dem Hindernis* gewechselt. Die Breite des Hindernisses wird mit dem linken und rechten Objektunkt ermittelt (vgl. Abschnitt 6.3). Bewegt sich das SCV in das Gebiet *schräg vor dem Hindernis*, so ändert sich der linke Objektunkt und folglich die Breite (Abb. 26a und 26b).



(a) Vor dem Hindernis

(b) Schräg vor dem Hindernis

Abb. 26: Erster Bereichsübergang während des Ausweichens

Schräg vor dem Hindernis \Rightarrow *Neben dem Hindernis*

Das SCV befindet sich solange *schräg vor dem Hindernis*, bis bei dem Links-Ausweichen der rechte und der kürzeste Punkt nicht demselben Messpunkt entsprechen. Bei *schräg vor dem Hindernis*, Abb. 27 sind die Punkte unterschiedlich. In diesem Beispiel ist der rechte Punkt vom Hindernis nicht mehr gleich dem rechten gemessenen Punkt, da der Laserscanner im Bereich 180° vor dem SCV misst, das Fahrzeug befindet sich aber noch schräg vor dem Hindernis. Ist der rechte gemessene Punkt gleich dem kürzesten gemessenen Punkt, wegen des Scanbereichs von 180° und der Position, so befindet sich das SCV neben dem Hindernis (vgl. Abb. 28).

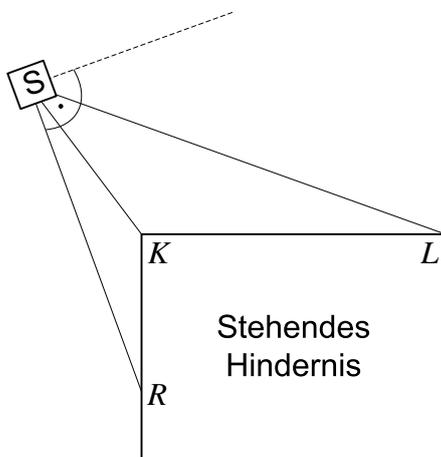


Abb. 27: Schräg vor dem Hindernis

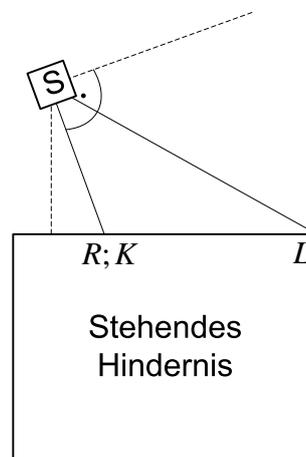


Abb. 28: Neben dem Hindernis

Neben dem Hindernis \Rightarrow Ausweichen Abgeschlossen

Der Ausweichvorgang ist abgeschlossen, sobald sich das SCV parallel zum Hindernis ausgerichtet hat. Das ist der Fall, wenn sich der Betrag des Achswinkels in einem definierten Toleranzband für 0° befindet.

6.3 SCV vor dem Hindernis

Befindet sich das Fahrzeug zu der Zeit der Laserscannermessung *vor dem Hindernis*, gelten die Berechnungsszenarien aus Abb. 29 für S_{ax} und S_{ay} und Abb. 30 für θ . In Abb. 29 ist für das Dreieck 1 \cup 2 der Winkel zwischen linkem und rechten Objektpunkt α , der Abstand zum rechten Objektpunkt b und der Abstand zum linken Objektpunkt c gegeben. Mit dem Kosinussatz wird die Breite des Hindernisses a in Gl. 34 berechnet.

$$a = \sqrt{b^2 + c^2 - 2bc \cdot \cos(\alpha)} \tag{34}$$

In Dreieck 2 in Abb. 29 ist neben c auch der geringste Abstand zum Hindernis b_1 gegeben. Das Abstandsminimum ist b_1 im Polarkoordinatensystem, in dem Messbereich von c bis b , einschließlich der Randpunkte in der Laserscannermessung und entspricht somit dem Punkt K des Objekts. Die Kante a_1 wird mit dem Satz des Pythagoras in Gl. 35 berechnet.

$$a_1 = \sqrt{c^2 - (b_1)^2} \tag{35}$$

Der Abstand in y -Richtung zum Hindernis entspricht dem Abstand zum Punkt K , also der Strecke b_1 (Gl. 36). S_{ay} wird aus den Strecken a und a_1 bestimmt, vgl. Gl. 37.

$$S_{ax} = b_1 \tag{36} \qquad S_{ay} = \frac{a}{2} - a_1 \tag{37}$$

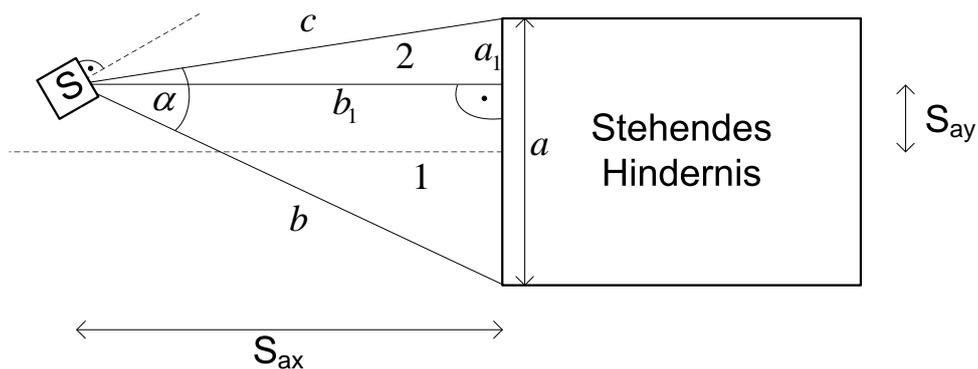
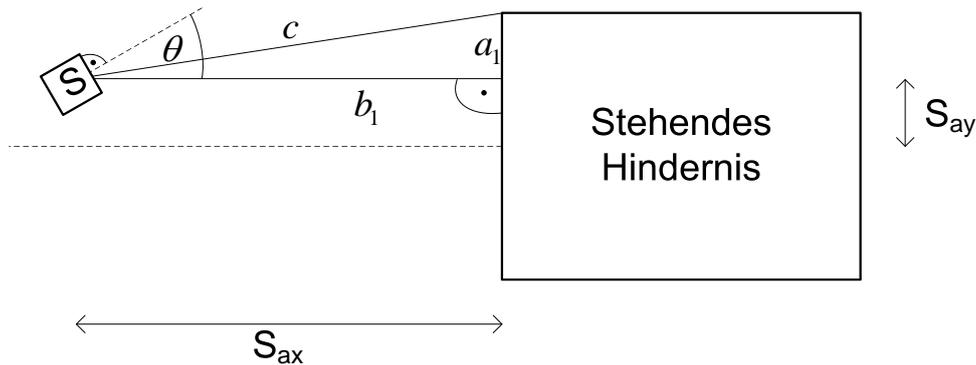


Abb. 29: Dreiecksberechnungen zur Bestimmung von S_{ax} und S_{ay}

Der Laserscanner liefert die Messwerte in Form von Polarkoordinaten, d.h. es ist der Winkel zu der Strecke b_1 in Abb. 30 bekannt, daher lässt sich nach Gl. 38 der Achswinkel θ berechnen.

$$\theta = \angle b_1 \tag{38}$$

Abb. 30: Bestimmung von θ bei SCV vor dem Hindernis

6.4 SCV schräg vor dem Hindernis

In dem Bereich *schräg vor dem Hindernis* werden für die Bestimmung der Abstände S_{ax} und S_{ay} zwei Dreiecke betrachtet (vgl. Abb. 31). Im ersten Schritt ist im Dreieck 1 die Strecke a_1 und der Winkel β_1 zu bestimmen. Die Strecken b_1 und c sind aus der Laserscannermessung bekannt und der Winkel α_1 ist die Differenz der Winkel ϕ von den Polarkoordinaten der Punkte *Rechts* und *Kurz*. Die Strecke a_1 wird wie in Abschnitt 6.3 nach dem Kosinussatz berechnet (vgl. Gl. 39). Für die Bestimmung des Winkels β_1 muss zunächst der Winkel γ_1 bestimmt werden, ausgehend vom Kosinussatz. Dazu wird dieser nach γ_1 aufgelöst (vgl. Gl. 40 bis 43). Anschließend ist β_1 mit der Winkelsumme eines Dreiecks zu berechnen (Gl. 44).

$$a_1 = \sqrt{(b_1^2 + c^2 - 2 \cdot b_1 \cdot c \cdot \cos(\alpha_1))} \quad (39)$$

$$c^2 = a^2 + b^2 - 2 \cdot a \cdot b \cdot \cos(\gamma_1) \quad (40)$$

$$2 \cdot a \cdot b \cdot \cos(\gamma_1) = a^2 + b^2 - c^2 \quad (41)$$

$$\cos(\gamma_1) = \frac{a^2 + b^2 - c^2}{2 \cdot a \cdot b} \quad (42)$$

$$\gamma_1 = \cos^{-1} \left(\frac{a^2 + b^2 - c^2}{2 \cdot a \cdot b} \right) \quad (43)$$

$$\beta_1 = 180^\circ - \alpha_1 - \gamma_1 \quad (44)$$

Im zweiten Schritt werden die Strecken a_2 und b_2 im Dreieck 2 ermittelt. Aus der Messung ist die Strecke c bekannt und in dem ersten Schritt wurde der Winkel β_1 berechnet.

Zunächst sind die Winkel β_2 und α_2 mit den Gln. 45 und 46 zu bestimmen, um die Strecke b_2 in Gl. 47 zu berechnen, die auch den Abstand in x-Richtung repräsentiert. In Gl. 48 wird mit dem *Satz des Pythagoras* a_2 ermittelt (Gl. 158). Der Abstand in y-Richtung wird mit beiden Dreiecken in Form von a_1 und a_2 in Gl. 49 berechnet.

$$\beta_2 = 180^\circ - \beta_1 \quad (45)$$

$$\alpha_2 = 90^\circ - \beta_2 \quad (46)$$

$$b_2 = \cos(\alpha_2) \cdot c = S_{ax} \quad (47)$$

$$a_2 = \sqrt{c^2 - (b_2)^2} \quad (48)$$

$$S_{ay} = \frac{a_1}{2} + a_2 \quad (49)$$

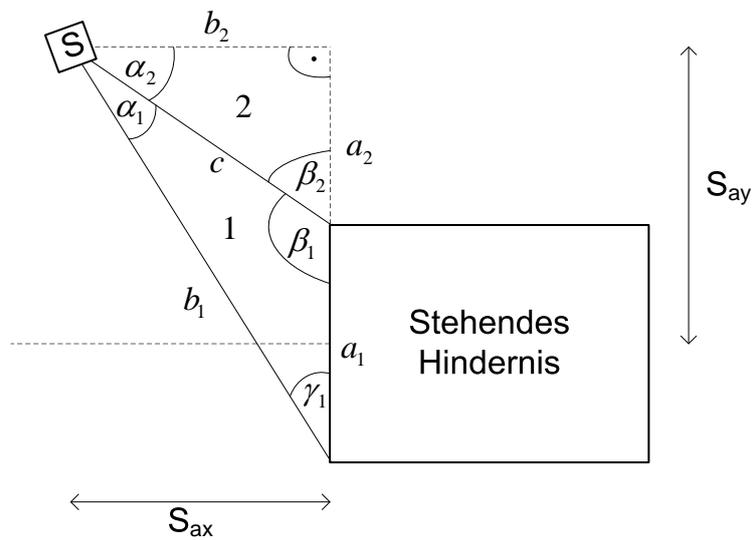


Abb. 31: Dreiecksberechnungen zur Bestimmung von S_{ax} und S_{ay}

Der Achswinkel θ wird mit der geometrischen Betrachtung aus Abb. 32 bestimmt. Der Winkel δ ist aus der Messung bekannt (siehe 6.3), während der Winkel α_2 durch die Bestimmung von S_{ax} und S_{ay} ermittelt wurde. Folglich ist θ mit einer Winkeldifferenz zu berechnen (vgl. Gl. 50).

$$\theta = \delta - \alpha_2 \quad (50)$$

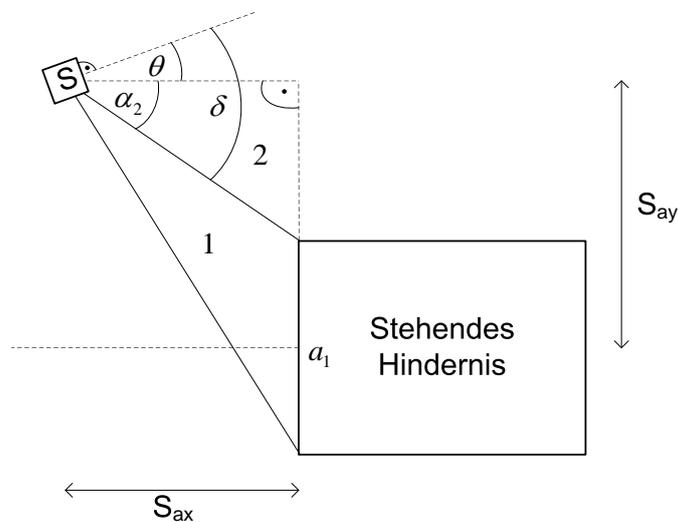


Abb. 32: Winkelberechnung zur Bestimmung von θ

6.5 SCV neben dem Hindernis

In dem Bereich *neben dem Hindernis* ist die Führungsgröße der Sicherheitsabstand, da sich das SCV parallel zum Hindernis mit diesem Abstand ausrichten soll (vgl. Abb. 33). Zu diesem Zeitpunkt ist kein Abstand zum Hindernis in x-Richtung mehr vorhanden, deshalb gilt Gl. 51. In der hier betrachteten Fahrzeugposition wird der y-Abstand näherungsweise bestimmt, da der Scanbereich auf 180° eingestellt ist. Die Strecke vom gemessenen rechten Objektpunkt ist der Abstand S_{ay} (Gl. 52). Der Winkel β in Abb. 33 steht wegen des Scanbereiches im Verhältnis zum Achswinkel (vgl. Gl. 53). Zur Bestimmung des Winkels β wird die Strecke a berechnet und mit dieser der Winkel γ und schließlich β (Gln. 54 bis 56). Es ist der Grenzfall Objektpunkt $K = L$ zu beachten, bei dem die Berechnung einen Achswinkel von 0° ergibt, auch wenn sich das SCV nicht parallel zum Hindernis befindet. Wird das Ende des Hindernisses erreicht, so tritt dieser Fall näherungsweise auf.

$$s_{ax} = 0 \quad (51) \quad a = \sqrt{b^2 + c^2 - 2 \cdot b \cdot c \cdot \cos(\alpha)} \quad (54)$$

$$s_{ay} \approx c \quad (52) \quad \gamma = \cos^{-1} \left(\frac{a^2 + b^2 - c^2}{2 \cdot a \cdot b} \right) \quad (55)$$

$$\theta = \beta - \frac{\pi}{2} \quad (53) \quad \beta = \pi - \gamma - \alpha \quad (56)$$

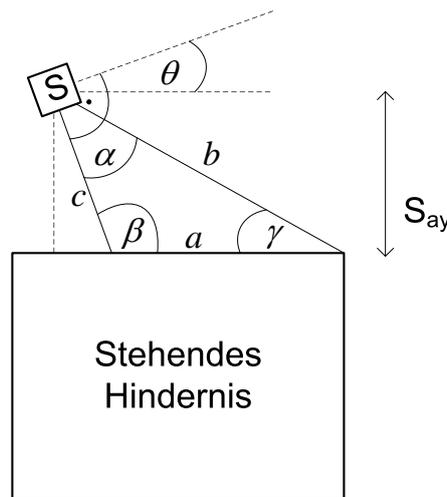


Abb. 33: Geometrisches Modell zur Bestimmung von S_{ay} und θ

6.6 Transformation der Abstandswerte von der Sensorposition S in die Vorderradposition M

Die geometrischen Berechnungen für S_{ax} , S_{ay} und θ sind sensorbezogen S . Der Regelkreis arbeitet aber vorderradbezogen M . Folglich müssen die Abstände abhängig vom Achswinkel transformiert werden. Durch den Fahrzeugaufbau des SCV gelten die folgenden Abstände zwischen Laserscanner und Vorderrad: $a_x = 5 \text{ cm}$ und $a_y = 31 \text{ cm}$. Laut dem *Satz des Pythagoras* resultiert daraus die Strecke a (Gl. 57) und mit dem Arccos der Winkel α (Gl. 58).

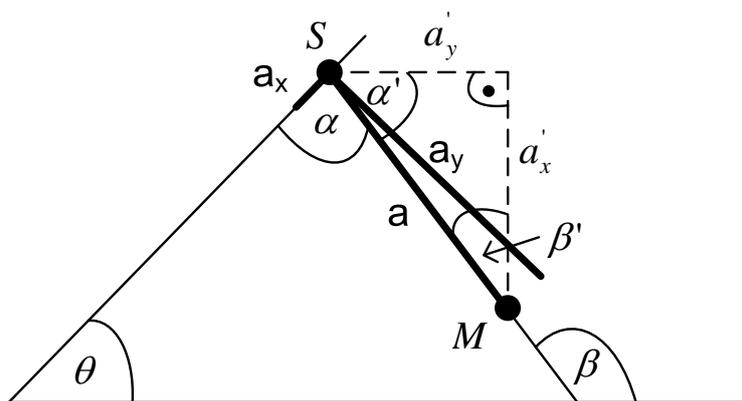


Abb. 34: Transformation der Abstandswerte von der Sensorposition zu der Vorderradposition (bei θ gleich 0° bis 90°)

$$a = \sqrt{a_x^2 + a_y^2} = 31,4 \text{ cm} \quad (57)$$

$$\alpha = \cos^{-1} \left(\frac{a_x}{a} \right) = 80,8^\circ \quad (58)$$

Für die Bestimmung der Strecken a'_x und a'_y ist der Winkel α' erforderlich, der in den Gln. 59 bis 61 berechnet wird. Dadurch sind die geforderten Strecken mit Gl. 62 und Gl. 63 zu bestimmen.

$$\beta = \theta + \alpha \quad (59)$$

$$\beta' = \beta - 90^\circ \quad (60)$$

$$\alpha' = 90^\circ - \beta' \quad (61)$$

$$a'_x = \cos(\alpha') \cdot a \quad (62)$$

$$a'_y = \sin(\alpha') \cdot a \quad (63)$$

Mit der Sensorposition und den Strecken a'_x und a'_y , sind die Vorderradabstände M_{ax} und M_{ay} während des Ausweichens nach links mit Gl. 64 und 65 zu berechnen. Eine Beispielrechnung für die Transformation und das Rechts-Ausweichen ist im Anhang H dokumentiert.

$$M_{ax} = S_{ax} - a'_x \quad (64)$$

$$M_{ay} = S_{ay} - a'_y \quad (65)$$

7 Entwurf des Abstandsregelkreises

Das SCV soll einem Hindernis durch eine Abstandsregelung gesteuert ausweichen. Der Regelkreis dazu wird in diesem Kapitel entworfen. Der Regelkreis für das System wird vorgestellt und die Übertragungsfunktion des SCV, sowie die Führungsübertragungsfunktion des Regelkreises berechnet. Diese gelten unabhängig von den Verfahren zur Bestimmung der Reglerparameter, d. h. sie sind für das System allgemein gültig. Folgend wird das *symmetrische Optimum* angewendet, um die Reglerparameter zu bestimmen. Anschließend werden die Frequenzkennlinien, ausgehend von dem Ergebnis des *symmetrischen Optimums*, im Bode-Diagramm verschoben. Dadurch kommt es zu einer Verbesserung des Regelverhaltens.

7.1 Regelkreisübertragungsfunktionen

Der Regelkreis des Ausweichassistenten wird im Folgenden vorgestellt, dieser regelt den y -Abstand des Vorderrades. Die Führungsgröße Soll-Abstandswert Y_s und der gemessene Ist-Abstandswert M_{ay} ergeben die Regeldifferenz e . Der Regler korrigiert das dynamische Verhalten der Regelung durch Vorgabe des Lenkwinkels α an das Fahrzeugmodell (vgl. Abb. 35). Das Ziel beim Entwurf eines Regelkreises ist es, einen stabilen, mit einer stationären Genauigkeit behafteten, Regelkreis zu erhalten. Zudem soll das Einschwingverhalten bei einer Sprungantwort eine kleine Ausregelzeit haben und außerdem hinreichend gedämpft sein.

Für den Regelkreis wird zunächst der Reglertyp ausgewählt und anschließend erläutert, aus welchem Grund eine y -Abstandsregelung ausreicht. Abschließend folgt die Berechnung der Übertragungsfunktion des SCV, sowie der Führungsübertragungsfunktion für die Reglereinstellverfahren.

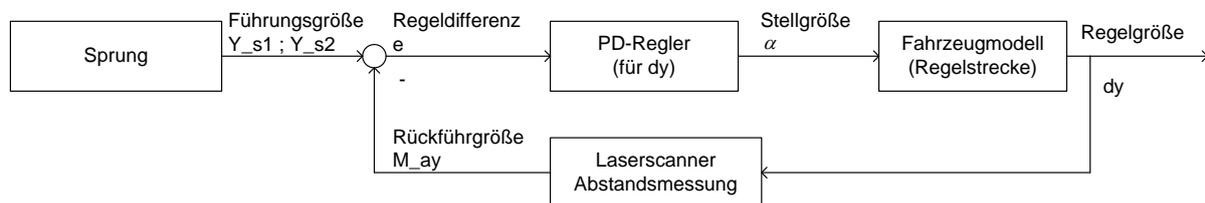


Abb. 35: Schema Regelkreis

Der Sprung der Führungsgröße Y_{s1} , ist einmalig zu Ausweichbeginn aus dem Sicherheitsseitenabstand SSA , der y -Koordinate des linken Objektpunktes $L.y$ und der relativen Position des SCV M_{ay} zu berechnen (vgl. Gl. 66). Da sich die relative Positionsbestimmung des SCV während des Ausweichmanövers auf die mittlere Hindernisbreite bezieht, muss diese zu Beginn berücksichtigt werden. Die Strecken $L.y$ und SSA repräsentieren die Ausweichbreite $y_{max,l}$ (vgl. Abb. 36).

$$Y_{s1} = M_{ay} + L.y + SSA \quad (66)$$

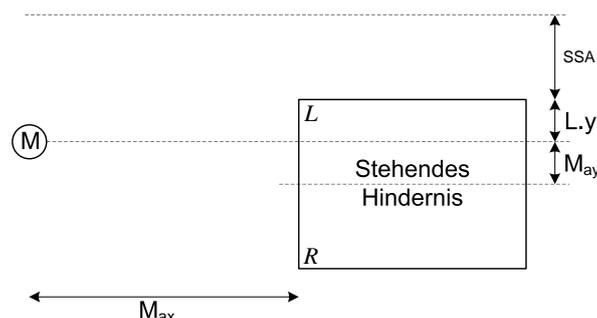


Abb. 36: Berechnung des Sprungs für die Führungsgröße

7.1.1 Auswahl des PD-Reglertyps

In diesem Abschnitt wird die Wahl des PD-Reglers im Regelkreis begründet. Das System des SCV hat durch die Lenk- und Antriebsmotoren ein I-Glied, das den Übergang von Geschwindigkeit zum Weg repräsentiert. Würde ein PID-Regler verwendet werden, so hätte das System zwei I-Glieder im offenen Regelkreis in Reihe geschaltet und das System würde nach dem Nyquist-Kriterium instabil werden. Das System kann in dem Fall durch die Einführung von Zählerzeitkonstanten stabilisiert werden, allerdings kann es dabei dazu kommen, dass die Forderungen, *Der Regelkreis soll hinreichend gedämpft sein* und *Der Regelkreis soll genügend schnell sein*, nicht erfüllt werden [12].

Da der offene Regelkreis des SCV bereits ein I-Glied enthält, ist die stationäre Genauigkeit gesichert. In Folge dessen wird für den Ausweichassistenten ein PD-Regler ausgewählt [12].

Der PD-Regler wird durch eine Parallelschaltung von Proportionalglied und Differenzierglied und einer Reihenschaltung mit einem weiteren Proportionalglied realisiert, welches den Faktor K_P beinhaltet (vgl. Abb. 37).

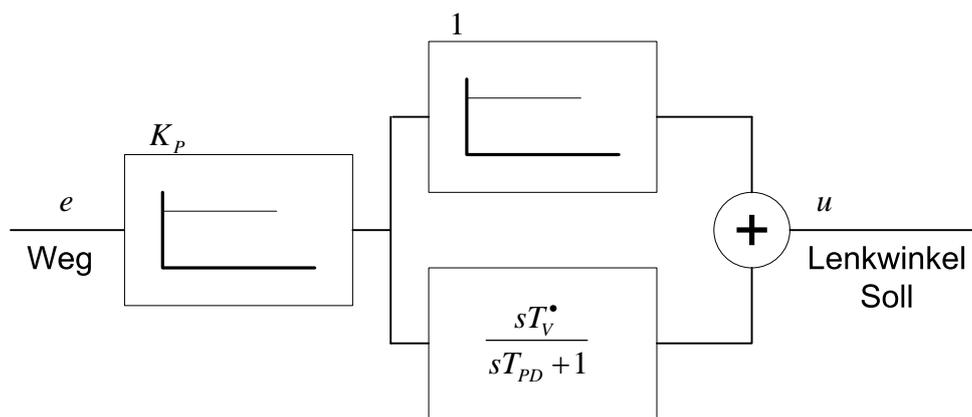


Abb. 37: Blockschema des PD-Reglers

Die Übertragungsfunktion des PD-Reglers Gl. 67 lässt sich aus der Abb. 37 ablesen. Bei dem Ausweichassistenten ist die Eingangsgröße die Regeldifferenz e , die sich aus der Differenz von Soll-y-Abstand und Ist-y-Abstand darstellt. Die Ausgangsgröße u ist der Soll-Lenk Winkel für das Vorderrad M . Die Gl. 67 ist durch Umformung und Substitution auf die Form in Gl. 70 zu bringen. Diese wird für die Regelbestimmung nach dem *symmetrischen Optimum* verwendet.

$$u = e \left[K_p \left(1 + \frac{sT_v^*}{sT_{PD} + 1} \right) \right] \quad (67)$$

$$= e \cdot K_p \frac{sT_v^* + (sT_{PD} + 1)}{sT_{PD} + 1} \quad (68)$$

$$= e \cdot K_p \frac{s(T_v^* + T_{PD}) + 1}{sT_{PD} + 1} \quad (69)$$

$$\text{mit } T_v = T_v^* + T_{PD}$$

$$= e \cdot K_p \frac{sT_v + 1}{sT_{PD} + 1} \quad (70)$$

7.1.2 Die Regelgröße y-Abstand

Der Ausweichassistent wird aktiviert, wenn der x-Abstand zum Hindernis im Verhältnis zu der Ausweichbreite zu gering ist. Daraus folgt, je größer die Ausweichbreite ist, desto größer ist der x-Abstand zum Hindernis, wenn das *CA_System* den Ausweichvorgang startet. Das *CA_System* startet ein Manöver, indem ein Sprung auf die Führungsgröße gegeben wird. Die Höhe des Sprungs ist dabei abhängig von der Ausweichbreite, auf den der Regler das System einregelt. Der Regler muss demnach nur den y-Abstand regeln, um auf die Ausweichbreite in y-Richtung zu kommen. Folglich bestimmen die Reglerparameter und das Startkriterium des Ausweichens den Abstand in x-Richtung.

7.1.3 Die Übertragungsfunktion des SCV

Die Übertragungsfunktion $G_S(s)$ des SCV wird im Folgenden bestimmt. Diese ist für die Verfahren zur Einstellung der Reglerparameter erforderlich. Zunächst wird das Blockscheema des Fahrzeugmodells aus den linearisierten Gleichungen für θ und für y_m erstellt (vgl. Abb. 38). Die Gleichungen wurden in Kapitel 5.6 hergeleitet.

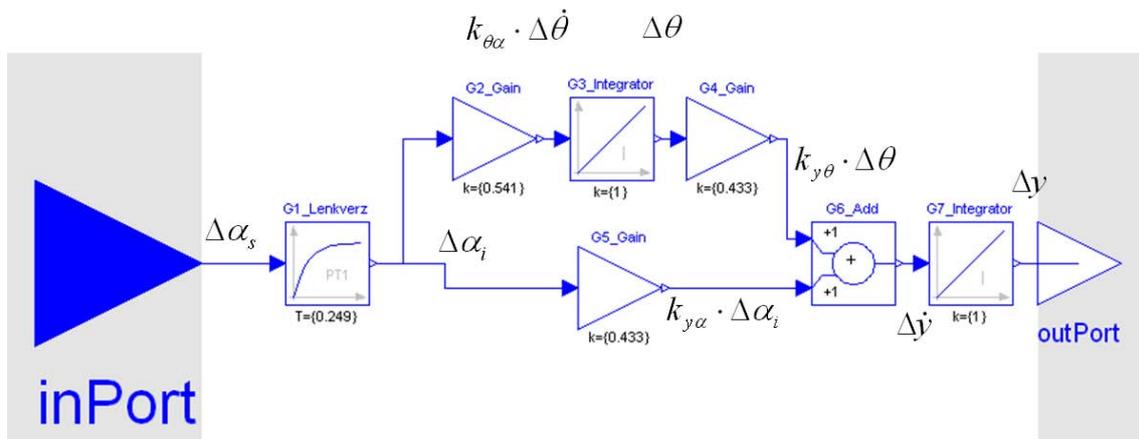


Abb. 38: Blockscheema der linearisierten Übertragungsfunktion des SCV

Aus dem Strukturbild entsteht mit den Zusammenfassungsregeln, Parallel- und Reihenschaltung von Gliedern, die Übertragungsfunktion für das SCV (vgl. Gl. 76). In den Gln. 71 bis 76 sind die Übertragungsfunktionen der einzelnen Glieder und deren schrittweise Zusammenfassung angegeben. Die Indizes der Übertragungsfunktionen entsprechen dabei den Blöcken aus Abb. 38.

$$\begin{aligned} G_1(s) &= \frac{K_{pl}}{(T_L s + 1)} \\ &= \frac{1}{T_L s + 1} \end{aligned} \quad (71)$$

$$\begin{aligned} G_{234}(s) &= K_{p2} \cdot K_3 \frac{1}{s} \cdot K_{p4} \\ &= k_{\theta\alpha} \cdot \frac{1}{s} \cdot k_{y\theta} \\ &= k_{\theta\alpha} \cdot k_{y\theta} \cdot \frac{1}{s} \\ &= k_{sub} \cdot \frac{1}{s} \end{aligned} \quad (72)$$

$$G_5(s) = k_{y\alpha} \quad (73)$$

$$G_6(s) = G_{234} + G_5 = k_{sub} \frac{1}{s} + k_{y\alpha} \quad (74)$$

$$\begin{aligned} G_{16}(s) &= \frac{1}{T_L s + 1} \cdot \left(k_{sub} \frac{1}{s} + k_{y\alpha} \right) \\ &= \frac{k_{sub} \frac{1}{s} + k_{y\alpha}}{T_L s + 1} = \frac{k_{y\alpha} s + k_{sub}}{T_L s^2 + s} \end{aligned} \quad (75)$$

$$\begin{aligned}
G_{17}(s) &= \frac{k_{y\alpha}s + k_{sub}}{T_L s^2 + s} \cdot \frac{1}{s} = \frac{k_{y\alpha}s + k_{sub}}{T_L s^3 + s^2} \\
&= \frac{\left(\frac{k_{y\alpha}}{k_{sub}}s + 1\right) \cdot k_{sub}}{(T_L s + 1) \cdot s^2} = G_S(s)
\end{aligned} \tag{76}$$

Die Gl. 76 wird mit der Einheitengleichung in Gl. 77 überprüft. Die Formelzeichen sind mit den entsprechenden Einheiten und Bedeutungen in Tabelle 6 aufgelistet. Die Regelstrecke SCV setzt die Stellgröße Lenkwinkel in die Regelgröße y-Abstand um. Der Winkel ist einheitenlos und ein Abstand ist eine Strecke und hat daher die Einheit m . Die Einheitengleichung 77 bestätigt also die Übertragungsfunktion des SCV (Gl. 76).

$$[G_S(s)] = \frac{\left(\frac{m/s}{m/s^2} \frac{1}{s} + 1\right) \cdot (m/s^2)}{\left(s \frac{1}{s} + 1\right) \cdot \frac{1}{s^2}} = \frac{\frac{m}{s^2} + (m/s^2)}{\frac{1}{s^2}} = m \tag{77}$$

Zeichen	Einheit	Bedeutung
s	$1/s$	Laplace Transformierte
T_L	s	Zeitkonstante Lenkverzögerung
$k_{\theta\alpha}$	$1/s$	Faktor aus Linearisierung von θ
$k_{y\theta}$	m/s	Faktor aus Linearisierung von y
$k_{y\alpha}$	m/s	Faktor aus Linearisierung von y
k_{sub}	m/s^2	Faktor substituiert; $k_{sub} = k_{\theta\alpha} \cdot k_{y\theta}$

Tabelle 6: Formelzeichen und Einheiten zur Übertragungsfunktion des SCV

7.1.4 Die Führungsübertragungsfunktion des Regelkreises

In diesem Abschnitt wird die Führungsübertragungsfunktion $G(s)$ für die anschließende Reglerparameterbestimmung mit dem *symmetrischen Optimum* im Abschnitt 7.2 bestimmt. Die Übertragungsfunktion $G_0(s)$ von dem Regler $G_R(s)$ und dem Fahrzeugmodell $G_S(s)$ berechnet sich mit Gl. 78. Dazu werden außerdem drei Substitutionen vorgenommen.

Substitutionen:

$$\begin{aligned}
\frac{k_{y\alpha}}{k_{sub}} &\Rightarrow T_{S1} \\
T_L &\Rightarrow T_{S2} \\
k_{sub} &\Rightarrow K_{S1}
\end{aligned}
\quad
\begin{aligned}
G_0(s) &= G_R(s) \cdot G_S(s) \\
&= \underbrace{\text{PD-Regler}}_{K_p \cdot \frac{sT_v + 1}{sT_{PD} + 1}} \cdot \frac{(T_{S1}s + 1) K_{S1}}{(T_{S2}s + 1) \cdot s^2} \tag{78}
\end{aligned}$$

In die allgemeine Form für die Führungsübertragungsfunktion Gl. 79 wird die Übertragungsfunktion Gl. 78 eingesetzt (Gl. 80). Nach der Umformung ergibt sich Gl. 81, wobei s durch $j\omega$ substituiert wurde. Da für das *symmetrische Optimum* der Nenner relevant ist, werden in der Zwischenbetrachtung die höheren Potenzen der imaginären Einheit j umgewandelt (vgl. Gln. 83 bis 86). Der Zähler wird in dem Verfahren nicht genutzt und deshalb nicht näher betrachtet.

$$G(s) = \frac{G_0(s)}{1 + G_0(s)} \quad (79)$$

$$= \frac{K_p(sT_v + 1)(T_{S1}s + 1)K_{S1}}{(sT_{PD} + 1)(T_{S2}s + 1) \cdot s^2 + K_p(sT_v + 1)(T_{S1}s + 1)K_{S1}} \quad (80)$$

$$= \frac{K_pK_{S1}(T_v s + T_{S1}s + T_v T_{S1}s^2 + 1)}{(T_{PD}T_{S2}s^2 + T_{PD}s + T_{S2}s + 1)s^2 + K_pK_{S1}(T_v s + T_{S1}s + T_v T_{S1}s^2 + 1)}$$

$$= \frac{K_pK_{S1}(T_v s + T_{S1}s + T_v T_{S1}s^2 + 1)}{T_{PD}T_{S2}s^4 + T_{PD}s^3 + T_{S2}s^3 + s^2 + K_pK_{S1}T_v s + K_pK_{S1}T_{S1}s + K_pK_{S1}T_v T_{S1}s^2 + K_pK_{S1}}$$

$$= \frac{K_pK_{S1}}{K_pK_{S1}} \cdot \frac{1 + (T_v + T_{S1})s + T_v T_{S1}s^2}{1 + (T_v + T_{S1})s + \left(T_v T_{S1} + \frac{1}{K_pK_{S1}}\right)s^2 + \left(\frac{T_{PD} + T_{S2}}{K_pK_{S1}}\right)s^3 + \frac{T_{PD}T_{S2}}{K_pK_{S1}}s^4}$$

$$= \frac{1 + (T_v + T_{S1})j\omega + T_v T_{S1}(j\omega)^2}{1 + (T_v + T_{S1})j\omega + \left(T_v T_{S1} + \frac{1}{K_pK_{S1}}\right)(j\omega)^2 + \left(\frac{T_{PD} + T_{S2}}{K_pK_{S1}}\right)(j\omega)^3 + \frac{T_{PD}T_{S2}}{K_pK_{S1}}(j\omega)^4} \quad (81)$$

$$= \frac{1 + (T_v + T_{S1})j\omega + T_v T_{S1}(j\omega)^2}{N(j\omega)} \quad (82)$$

Zwischenbetrachtungen:

$$(T_v + T_{S1})j\omega \Rightarrow (T_v + T_{S1})\omega \cdot j \quad (83)$$

$$\left(T_v T_{S1} + \frac{1}{K_pK_{S1}}\right)(j\omega)^2 \Rightarrow -\left(T_v T_{S1} + \frac{1}{K_pK_{S1}}\right)\omega^2 \quad (84)$$

$$\left(\frac{T_{PD} + T_{S2}}{K_pK_{S1}}\right)(j\omega)^3 \Rightarrow -\left(\frac{T_{PD} + T_{S2}}{K_pK_{S1}}\right)\omega^3 \cdot j \quad (85)$$

$$\frac{T_{PD}T_{S2}}{K_pK_{S1}}(j\omega)^4 \Rightarrow \frac{T_{PD}T_{S2}}{K_pK_{S1}}\omega^4 \quad (86)$$

Aus der Zwischenbetrachtung ergibt sich für den Nenner $N(j\omega)$ die Gl. 87, sortiert nach Realteil und Imaginärteil. Die Koeffizienten der ω -Potenzen werden zur Vereinfachung der folgenden Rechnungen in Abschnitt 7.2 durch a, b, c und d substituiert, Gl. 88.

$$\Rightarrow N(j\omega) = 1 - \underbrace{\left(T_v T_{S1} + \frac{1}{K_pK_{S1}}\right)}_a \omega^2 + \underbrace{\frac{T_{PD}T_{S2}}{K_pK_{S1}}}_b \omega^4 + \left(\underbrace{(T_v + T_{S1})}_c \omega - \underbrace{\left(\frac{T_{PD} + T_{S2}}{K_pK_{S1}}\right)}_d \omega^3 \right) \cdot j \quad (87)$$

$$= 1 - a \cdot \omega^2 + b \cdot \omega^4 + (c \cdot \omega - d \cdot \omega^3) \cdot j \quad (88)$$

7.2 Reglerdimensionierung mit dem symmetrischen Optimum

Folgend sollen mit der Einstellregel *symmetrisches Optimum* die Reglerparameter bestimmt werden. Das Verfahren stammt von C. Kessler aus dem Jahr 1958 und ist in der Antriebstechnik verbreitet. Das Fahrzeugmodell wird als Verzögerungssystem mit reellen Polen vorausgesetzt

und es wird angenommen, dass sich die Zeitkonstanten in eine Gruppe großer und in eine Gruppe kleiner Zeitkonstanten aufteilt [12].

Der Führungsfrequenzgang $F_w(j\omega)$ soll in einer großen Umgebung von $\omega = 0$ gleich 0 sein. Dieses soll durch geeignete Wahl der Reglerparameter erreicht werden, dem vor allem die absenkende Tendenz der Nennerzeitkonstante entgegen wirkt. D.h. es soll in einer großen Umgebung von $\omega = 0$ folgendes gelten: $|N(j\omega)| = 1$ (vgl. [12]).

Das Ziel ist die Gl. 89 zu erfüllen, indem die Faktoren $k_{\omega 2}$, $k_{\omega 4}$, $k_{\omega 6}$ und $k_{\omega 8}$ zu Null gemacht werden.

$$N(j\omega)N(-j\omega) = 1 + k_{\omega 2}\omega^2 + k_{\omega 4}\omega^4 + k_{\omega 6}\omega^6 + k_{\omega 8}\omega^8 \stackrel{!}{=} 1 \quad (89)$$

Die unbekannt Reglerparameter K_p , T_v und T_{PD} sind demnach zu bestimmen. Dazu wird der Nenner $N(j\omega)$ mit dem konjugiert Komplexen multipliziert (vgl. Gl. 90). Die Koeffizienten der ω -Potenzen werden aus der Gl. 90 bestimmt und gleich Null gesetzt, Gln. 91, 92 bzw. 93, 94 und 95.

$$\begin{aligned} N(j\omega)N(-j\omega) &= (1 - a \cdot \omega^2 + b \cdot \omega^4)^2 + (c \cdot \omega - d \cdot \omega^3)^2 \\ &= (1 - a\omega^2 + b\omega^4 - a\omega^2 + a^2\omega^4 - ab\omega^6 + b\omega^4 - ab\omega^6 + b^2\omega^8) \\ &\quad + (c^2\omega^2 - 2cd\omega^4 + d^2\omega^6) \end{aligned} \quad (90)$$

ω -Potenzen gleich Null setzen und Umformen:

$$\begin{aligned} \omega^2 &\Rightarrow -2a + c^2 = 0 \\ a &= \frac{c^2}{2} \end{aligned} \quad (91) \quad \begin{aligned} \omega^6 &\Rightarrow -2(a \cdot b) + d^2 = 0 \\ 2(a \cdot b) &= d^2 \end{aligned}$$

$$\begin{aligned} \omega^4 &\Rightarrow a^2 + 2b - 2(c \cdot d) = 0 \\ a^2 + 2b &= 2(c \cdot d) \end{aligned} \quad (92) \quad \begin{aligned} 2\left(\frac{c^2}{2} \cdot b\right) &= d^2 \\ c^2 \cdot 2b &= d^2 \end{aligned}$$

mit ω^2 :

$$\left(\frac{c^2}{2}\right)^2 + 2b = 2(c \cdot d) \quad c \cdot \sqrt{2b} = d \quad (94)$$

$$\frac{1}{4}c^4 + 2b = 2(c \cdot d) \quad \omega^8 \Rightarrow b^2 = 0 \quad b = 0 \quad (95)$$

$$\frac{1}{8}c^3 + \frac{b}{c} = d \quad (93)$$

Das Verhältnis zwischen a und c ergibt sich aus der Umformung des Koeffizienten von ω^2 . Die Koeffizienten von ω^4 und ω^6 werden nach d aufgelöst, um sie im Folgenden gleich zu setzen. Der Koeffizient von ω^8 wird nur Null, wenn auch $b = 0$ ist. Unter dieser Voraussetzung werden aber auch alle anderen Koeffizienten gleich Null. Das Ziel $k_{\omega 8} = 0$ wird also nicht erreicht. Denn sobald $b = 0$ gilt, folgt daraus Gl. 94 $d = 0$, Gl. 93 $c = 0$ und Gl. 91 $a = 0$. Deshalb werden im Weiteren die Koeffizienten $k_{\omega 2}$, $k_{\omega 4}$ und $k_{\omega 6}$ betrachtet.

Verhältnisse der Reglerparameter bestimmen:

Im ersten Schritt wird aus dem Verhältnis Gl. 96 die Gl. 97 für K_p bestimmt. Die Zeitkonstante T_v aus dem Regler muss vorgegeben werden, T_{S1} und K_{S1} sind aus der Übertragungsfunktion $G_S(s)$ bekannt.

$$a = \frac{c^2}{2} \quad (96)$$

$$T_v T_{S1} + \frac{1}{K_p K_{S1}} = \frac{(T_v + T_{S1})^2}{2}$$

$$\frac{1}{K_p K_{S1}} = \frac{(T_v + T_{S1})^2}{2} - \frac{2 \cdot T_v T_{S1}}{2}$$

$$K_p K_{S1} = \frac{2}{(T_v + T_{S1})^2 - 2 \cdot T_v T_{S1}}$$

$$\text{mit } (T_v + T_{S1})^2 = T_v^2 + 2T_v T_{S1} + T_{S1}^2$$

$$K_p = \frac{2}{(T_v^2 + T_{S1}^2) \cdot K_{S1}} \quad (97)$$

Im zweiten Schritt werden die Gl. für ω^4 und ω^6 in 98 gleichgesetzt, um die Gl. 113 für T_v^\bullet zu bestimmen. T_v^\bullet wurde in Abschnitt 7.1.1 hergeleitet. Zunächst muss die Gl. 98 auf die passende Form für die p-q-Formel gebracht werden (Gl. 102). Anschließend ergibt die Formel die zwei Faktoren b_1 und b_2 (vgl. Gln. 103 bis 108). In den Gln. 109 bis 113 wird resubstituiert und nach T_v^\bullet umgestellt. Aus der Gl. 113 lässt sich die Reglerzeitkonstante T_{PD} mit $T_{PD} = T_v - T_v^\bullet$ bestimmen.

$$\frac{1}{8}c^3 + \frac{b}{c} = c \cdot \sqrt{2b} \quad (98) \quad \text{mit } p = -\frac{7}{4}c^4 \text{ und } q = \frac{1}{64}c^8 \quad (103)$$

$$\frac{1}{8}c^2 + \frac{b}{c^2} = \sqrt{2b} \quad (99) \quad b_{1,2} = \frac{7}{8}c^4 \pm \sqrt{\frac{49}{64}c^8 - \frac{1}{64}c^8} \quad (104)$$

$$\left(\frac{1}{8}c^2 + \frac{b}{c^2}\right)^2 = 2b \quad (100) \quad b_{1,2} = \frac{7}{8}c^4 \pm \sqrt{\frac{3}{4}c^8} \quad (105)$$

$$\frac{1}{64}c^4 + 2\frac{1}{8}c^2 \cdot \frac{b}{c^2} + \frac{b^2}{c^4} = 2b \quad (101) \quad b_{1,2} = \frac{7}{8}c^4 \pm \sqrt{\frac{3}{4}c^4} \quad (106)$$

$$b^2 - \frac{7}{4}bc^4 + \frac{1}{64}c^8 = 0 \quad (102) \quad b_1 = 0,00897c^4 \quad (107)$$

$$b_2 = 1,74c^4 \quad (108)$$

$$\frac{T_{PD}T_{S2}}{K_p K_{S1}} = b_{fak} \cdot (T_v + T_{S1})^4 \quad (109)$$

$$T_{PD} = b_{fak} \cdot \frac{K_p K_{S1} (T_v + T_{S1})^4}{T_{S2}} \quad (110)$$

$$\text{mit } T_v = T_v^\bullet + T_{PD} \quad (111)$$

$$T_v - T_v^\bullet = T_{PD} = b_{fak} \cdot \frac{K_p K_{S1} (T_v + T_{S1})^4}{T_{S2}} \quad (112)$$

$$T_v^\bullet = T_v - b_{fak} \cdot \frac{K_p K_{S1} (T_v + T_{S1})^4}{T_{S2}} \quad (113)$$

Einheitenprüfung:

Die Einheiten der Formelzeichen aus den Gln. 113 und 97 sind in 114 bis 119 aufgelistet. Die Einheitengleichung 120 bestätigt die Verhältnisgleichung für K_p (Gl. 113). Die Überprüfung der Bestimmungsgleichung für T_v^\bullet (Gl. 113) erfolgt mit der Einheitengleichung 121.

$$[K_P] = \frac{\text{Grad}}{m} \quad (114)$$

$$[T_v] = s \quad (115)$$

$$[T_{S1}] = \frac{m/s}{\frac{1}{s} \cdot (m/s)} = s \quad (116)$$

$$[K_{S1}] = \frac{m/\text{Grad}}{s^2} \quad (117)$$

$$[T_v^\bullet] = s \quad (118)$$

$$[T_{S2}] = s \quad (119)$$

$$\frac{\text{Grad}}{m} = \frac{1}{(s^2 + s^2) \cdot \frac{m/\text{Grad}}{s^2}} = \frac{\text{Grad}}{m} \quad (120)$$

$$s = s - \frac{(\text{Grad}/m) \cdot \frac{(m/\text{Grad})}{s^2} \cdot s^4}{s} = s \quad (121)$$

7.2.1 Bestimmung der Reglerparameter mit verschiedenen Vorgaben

Mit dem *symmetrischen Optimum* wurden zwei Verhältnisse der Reglerparameter bestimmt. Für die Berechnung der Parameter müssen der Arbeitspunkt und die Zeitkonstante T_v vorgegeben werden, um mit den Gln. 97 und 113 die weiteren Reglerparameter zu berechnen. Es ist allerdings noch zu klären, welcher Faktor für b_{fak} bei der Bestimmung von T_{PD} gültig ist. Die beiden Lösungen aus der p-q-Formel werden in Tabelle 7 miteinander verglichen, wobei die Vorgaben für den Arbeitspunkt und T_v identisch sind. Bei b_1 sind die bestimmten Zeitkonstanten positiv und die Koeffizienten von ω^4 und ω^6 sind kleiner als eins und deshalb dem Ziel Null nahe. Bei dem Faktor b_2 wird die Zeitkonstante T_v^\bullet negativ und $k_{\omega 4}$ und $k_{\omega 6}$ sind weit von dem Ziel entfernt, die Koeffizienten Null zu machen. Also ist der Faktor $b_1 = 0,00897$ für die Berechnung von T_{PD} gültig und b_2 ungültig.

b_{fak}	$\alpha [^\circ]$	$\theta [^\circ]$	$T_v [s]$	$T_{PD} [s]$	$T_v^\bullet [s]$	$K_p [^\circ/m]$	$k_{\omega 2}$	$k_{\omega 4}$	$k_{\omega 6}$
b_1	10	20	0,5	0,4354	0,0646	2,8801	0	0,6836	0,1527
b_2	10	20	0,5	84,458	-83,909	2,8801	0	-322,69	9327,7

Tabelle 7: Vergleich Faktor b_1 und b_2

Die Reglerparameter für verschiedene Arbeitspunkte und unterschiedliche T_v sind in Tabelle 8 aufgelistet. In den Fällen 1 bis 3 wurde T_v und in den Fällen 4 und 5, der Arbeitspunkt, variiert. Bei 1 ist die Zeitkonstante T_v^\bullet negativ, deswegen wurde zu Fall 2, T_v vergrößert, da T_v und T_v^\bullet in Beziehung zueinander stehen. Die Zeitkonstanten sind dadurch in Fall 2 gültig und die ω -Koeffizienten gleich Null bzw. in der Nähe. In Fall 3 ist zu sehen, dass ein weiteres Erhöhen von T_v auch die Koeffizienten vergrößert und dadurch dem Ziel entgegen wirkt. Deswegen wird in dem folgenden Fall 4 der AP verändert, ausgehend von $T_v = 0,5$ s. Durch die Veränderung des APs wird jedoch die Zeitkonstante T_v^\bullet wieder negativ. Durch ein Erhöhen von T_v wird die Zeitkonstante wieder gültig, die Koeffizienten $k_{\omega 4}$ und $k_{\omega 6}$ werden aber stark ungleich Null. Diese fünf Fälle zeigen exemplarisch die Variation von AP und T_v , wobei auf die Auflistung von weiteren Fällen verzichtet wird. Im weiteren Verlauf der Masterarbeit werden die Reglerparameter aus der Berechnung zwei verwendet und als Regler 1 bezeichnet. Der Matlab-Code für die Berechnung mit dem *symmetrischen Optimum* ist im Anhang D zu finden, Listing 2.

Nr.	α [°]	θ [°]	T_v [s]	T_{PD} [s]	T_v^\bullet [s]	K_p [°/m]	$k_{\omega 2}$	$k_{\omega 4}$	$k_{\omega 6}$
1	10	20	0,3	0,3021	-0,0021	3,0963	0	0,3627	0,1053
2	10	20	0,5	0,4354	0,0646	2,8801	0	0,6836	0,1527
3	10	20	1	0,8247	0,1753	2,1698	0	1,3923	0,8238
4	-10	10	0,5	0,5493	-0,0493	2,135	0	0,85	0,57
5	-10	10	2	1,9942	0,0058	0,9895	0	-6,6545	37,3220

Tabelle 8: Vergleich von verschiedenen Vorgaben für AP und T_v

7.2.2 Ergebnissimulationen zum symmetrischen Optimum

Die Simulation des Ausweichvorgangs mit dem Fahrzeugmodell des SCV ist in Abb. 39 dokumentiert. Dazu wurde zu dem Zeitpunkt $t = 1$ s ein Sprung von 0,5 m auf die Führungsgröße des Regelkreises gegeben. Das Vorderrad hat ein Überschwingen von ca. 0,15 m. Das Hinterrad hat dabei ein sehr geringes Überschwingen. Die Ausregelzeit, das ist die Zeit, bis die Räder im Toleranzbereich der Führungsgröße sind, beträgt bei dem Vorderrad, sowie bei dem Hinterrad ungefähr 6 s.

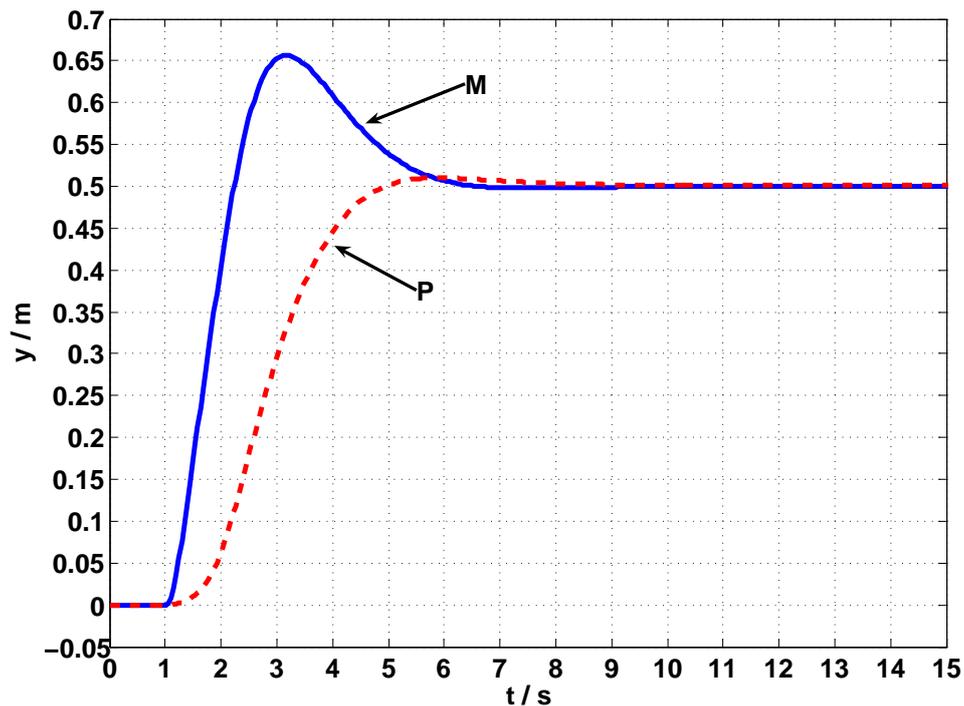


Abb. 39: Vorder- und Hinterradposition des SCV zu Regler 1

Der Lenkwinkel α des SCV hat einen Gültigkeitsbereich von -90° bis 90° . Dieser wird vom Soll-Lenkwinkel zu Beginn des Ausweichens verletzt, durch die Lenkverzögerung ist der Ist-Lenkwinkel aber innerhalb des Bereichs (vgl. Abb. 40). Der Achswinkel θ steigt auf ca. 30° und fällt anschließend wieder ab. Das SCV richtet sich dann parallel zum Hindernis aus (Abb. 41).

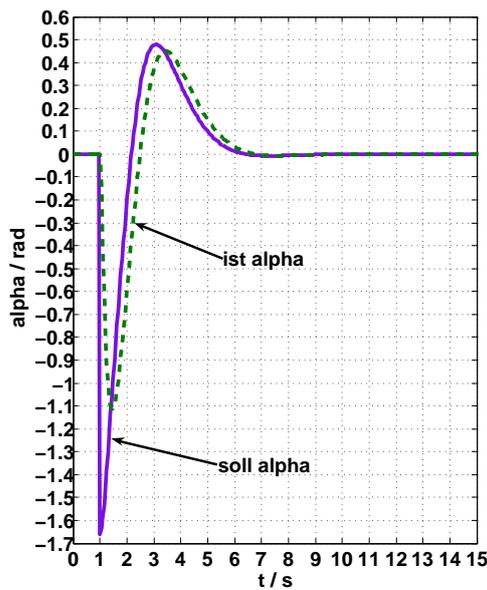


Abb. 40: Lenkwinkel des SCV zu Regler 1

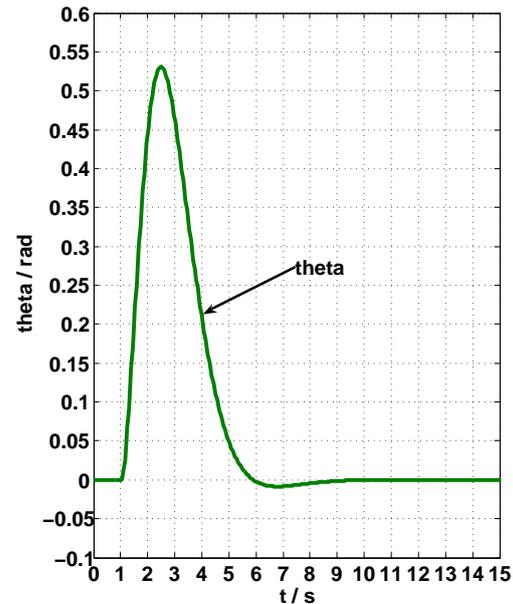


Abb. 41: Achswinkel des SCV zu Regler 1

7.2.3 Betrachtung der Systemstabilität

Zur Regelung eines Systems muss es stabil sein, vgl. Abschnitt 7.1. Dieses ist erfüllt, wenn alle Nullstellen des Nennerpolynoms von der Übertragungsfunktion einen negativen Realanteil besitzen, d.h. die Gesamtfunktion des Systems klingt ab. Im ersten Schritt wird dies mit dem *notwendigem Kriterium* überprüft. Sind ein oder mehrere Koeffizienten des Nennerpolynoms Null oder negativ, so ist der Realteil von mindestens einer Nullstelle positiv oder gleich Null und folglich instabil. Sind alle Koeffizienten positiv und ungleich Null, so ist das *notwendige Kriterium* erfüllt. Das bedeutet aber nicht, dass das System stabil sein muss, da es ein notwendiges Kriterium ist. Für das Nennerpolynom der Übertragungsfunktion resultiert aus Gl. 122 mit den eingesetzten Parametern aus der Regelstrecke und des Reglers Gl. 123. Die Koeffizienten a_0 bis a_4 sind in Gl. 123 alle positiv und ungleich Null, das *notwendige Kriterium* für die Übertragungsfunktion ist daher erfüllt.

Parameter der Regelstrecke und des Reglers:

$$\begin{aligned} T_v &= 0,5s & T_{PD} &= 0,4354s & K_p &= 2,8801^\circ/m \\ T_{S1} &= 1,4287s & T_{S2} &= 0,249s & K_{S1} &= 0,3031 \frac{m}{s^2} \end{aligned}$$

$$1 + (T_v + T_{S1})s + \left(T_v T_{S1} + \frac{1}{K_p K_{S1}} \right) s^2 + \left(\frac{T_{PD} + T_{S2}}{K_p K_{S1}} \right) s^3 + \frac{T_{PD} T_{S2}}{K_p K_{S1}} s^4 \quad (122)$$

$$\underbrace{1}_{a_4} + \underbrace{1,9287s}_{a_3} + \underbrace{1,8599s^2}_{a_2} + \underbrace{0,7840s^3}_{a_1} + \underbrace{0,1242s^4}_{a_0} \quad (123)$$

Im zweiten Schritt werden die Koeffizienten mit dem *Kriterium von Hurwitz* überprüft. Es erlaubt eine Stabilitätsaussage, wenn alle Koeffizienten ungleich Null und positiv sind. Dazu werden die Koeffizienten a_0 bis a_4 in die Hurwitz-Determinante (Gl. 124) eingesetzt und die Unterdeterminanten bis H_3 ausgewertet. Sind alle Unterdeterminanten positiv, so haben alle

Nullstellen einen negativen Realanteil und das *Kriterium von Hurwitz* ist erfüllt, in dem Fall ist das System stabil. Die Unterdeterminanten des SCV werden in Gl. 125 bis Gl. 127 berechnet.

$$H = \begin{pmatrix} a_1 & a_3 & a_5 & a_7 \\ a_0 & a_2 & a_4 & a_6 \\ 0 & a_1 & a_3 & a_5 \\ 0 & a_0 & a_2 & a_4 \end{pmatrix} \quad (124)$$

$$H_1 = 0,7840 > 0 \quad (125)$$

$$H_2 = \begin{pmatrix} 0,7840 & 1,9287 \\ 0,1242 & 1,8599 \end{pmatrix} = 1,4582 - 0,2395 = 1,2187 > 0 \quad (126)$$

$$\begin{aligned} H_3 &= \begin{pmatrix} 0,7840 & 1,9287 & 0 \\ 0,1242 & 1,8599 & 1 \\ 0 & 0,7840 & 1,9287 \end{pmatrix} \\ &= 2,8124 - 0,6147 - 0,4620 = 1,7357 > 0 \end{aligned} \quad (127)$$

Das *notwendige Kriterium* ist erfüllt und alle drei Unterdeterminanten vom *Kriterium von Hurwitz* sind positiv, d. h. das System ist stabil. Die beiden Kriterien gehören zu den *Algebraischen Stabilitätskriterien*, mit denen sich die Systemstabilität überprüfen lässt, ohne die Nullstellen des Polynoms berechnen zu müssen. Für weitere Details zur Systemstabilität wird auf [17] verwiesen.

7.3 Reglereinstellung nach dem Frequenzkennlinienverfahren

In diesem Abschnitt wird der bestimmte Regler aus Kapitel 7.2 durch die Analyse der Frequenzkennlinien des offenen Regelkreises und des Reglers optimiert. Bei dem Ausgangsregler hat die Sprungantwort des Vorderrads M ein Überschwingen von ca. 0,15 m, bei einem Sprung von 0,5 m. Ein großes Überschwingen bei dem Links-Ausweichen hat z. B. auf einer zweispurigen Autobahn die Folge, dass das Fahrzeug mit der Mittelleitplanke kollidiert. Das Ziel der Frequenzkennlinienanalyse ist die Reduktion des Überschwingens, wobei die Ausregelzeit hinreichend klein bleiben soll. In Kapitel 7.4 werden die Ergebnisse der Reglerbestimmung aus den beiden Verfahren *symmetrisches Optimum* und *Frequenzkennlinien* verglichen und beurteilt.

Bei dem *Frequenzkennlinienverfahren* wird der Frequenzgang in die zwei Bestandteile Betrag $|G(j\omega)|$ und Phase $\angle G(j\omega)$ aufgeteilt. Die Betragskennlinie und die Phasenkennlinie werden im Bode-Diagramm über der ω -Achse logarithmisch aufgetragen. Dadurch werden die Parameter des offenen Regelkreises durch markante, geometrische Punkte im Diagramm dargestellt. Das dynamische Verhalten der Regelung ist hiermit veranschaulicht und es lassen sich gezielte Maßnahmen zur Verbesserung der Dynamik vornehmen [12]. Zwei Regelkreiseigenschaften, die sich geometrisch bestimmen lassen, sind z. B.:

- Die Phasenreserve ϕ_r dient als Maß für die Dämpfung des Einschwingvorgangs der Sprungantwort des geschlossenen Kreises. Je größer ϕ_r , umso gedämpfter ist der Einschwingvorgang. Der Winkel, der bei ω_c in der Phasenkennlinie oberhalb von -180° liegt, entspricht ϕ_r . [18]

- Die Durchtrittsfrequenz ω_c dient als Maß für die Schnelligkeit des Führungsverhaltens. Je größer ω_c , um so schneller der Zeitvorgang und umgekehrt. Die Durchtrittsfrequenz entspricht der Frequenz, bei der die Betragskennlinie des offenen Regelkreises die $0-dB$ -Linie schneidet.[18]

7.3.1 Die Übertragungsfunktion des offenen Regelkreises

Für die Bestimmung der Übertragungsfunktion des offenen Regelkreises wird der geschlossene Regelkreis vor dem Soll-Ist-Vergleich aufgetrennt. Die resultierende Reihe an Übertragungsglieder entspricht dem offenen Regelkreis [12]. Demnach wird die Gl. 78 bzw. Gl. 128 zu Gl. 130 weiterentwickelt.

$$F_0(s) = G_R(s) \cdot G_S(s) \quad (128)$$

$$= K_p \cdot \frac{sT_v + 1}{sT_{PD} + 1} \cdot \frac{(T_{S1}s + 1) K_{S1}}{(T_{S2}s + 1) \cdot s^2} \quad (129)$$

$$= K_p \cdot K_{S1} \cdot \frac{(T_v \cdot T_{s1})s^2 + (T_v + T_{s1})s + 1}{(T_{PD} \cdot T_{S2})s^4 + (T_{PD} + T_{S2})s^3 + s^2} \quad (130)$$

Die Verstärkungsfaktoren K_P und K_{S1} sind vor den Quotienten gezogen und werden für die folgenden Analysen der Frequenzkennlinien nicht näher betrachtet, da sie im Bode-Diagramm eine Verschiebung der Kurve auf der y-Achse bewirken, an der Charakteristik der Kurve aber nichts ändern.

7.3.2 Analyse der Frequenzkennlinien

Durch Einsetzen der Strecken- und Reglerparameter aus dem *symmetrischen Optimum* in die Gl. 130 für den offenen Regelkreis ergibt sich Gl. 131. Wie bereits in Abschnitt 7.3.1 erwähnt, werden die Verstärkungsfaktoren K_P und K_{S1} vernachlässigt. Daraus ergibt sich aus dem Bode-Diagramm eine Durchtrittsfrequenz von $\omega_c = 0,97 \text{ rad/sec}$ und eine Phasenreserve von $\phi_r = 17^\circ$ (vgl. Abb. 42).

$$F_0(s) = \frac{0,714s^2 + 1,929s + 1}{0,108s^4 + 1,864s^3 + s^2} \quad (131)$$

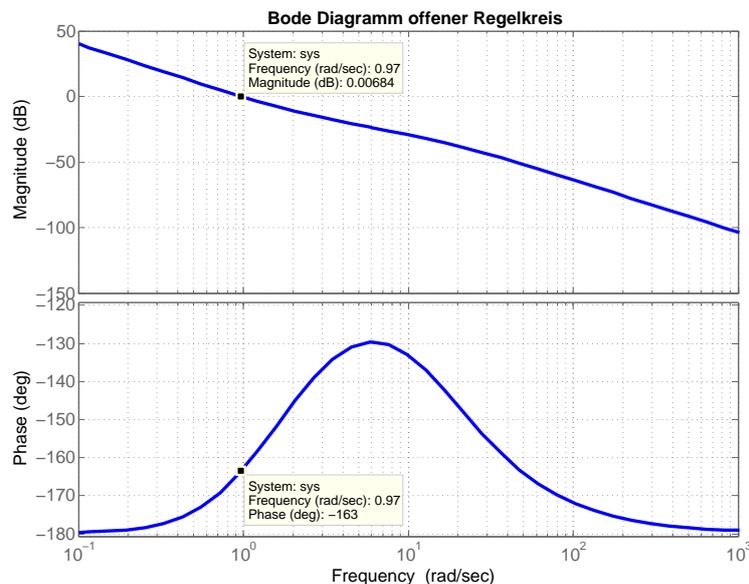


Abb. 42: Offener Regelkreis nach dem *symmetrischen Optimum*

Mit den berechneten Reglerparametern aus dem *symmetrischen Optimum* ergibt sich die Gl. 132 für den Regler. Das Bode-Diagramm ist in Abb. 43 dargestellt. Die maximale Verstärkung V beträgt 1,15 dB und die Phasenhebung 3,96°.

$$F_R(s) = \frac{0,5s + 1}{0,435s + 1} \quad (132)$$

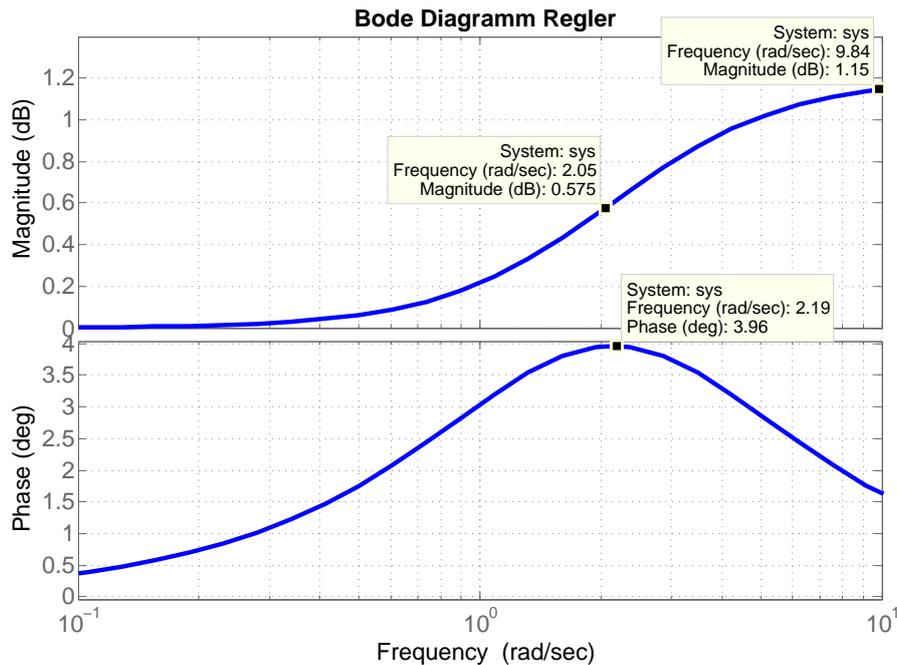


Abb. 43: Regler nach dem *symmetrischen Optimum*

Die Ergebnissimulationen zum *symmetrischen Optimum* in 7.2.2 haben ein Überschwingen des Vorderrades von 0,15 m aufgezeigt, welches u.a. mit der niedrigen Phasenreserve des offenen Regelkreises von 17° zusammenhängt. Diese Phasenreserve resultiert aus der maximalen Verstärkung des Reglers, die abhängig von den Zeitkonstanten des Reglers Gl. 132 ist. Es gilt, je größer die Differenz der Zeitkonstanten, desto größer die maximale Verstärkung des Reglers.

Um das Überschwingen des Vorderrades zu dämpfen, wird die Differenz der Reglerzeitkonstanten erhöht (Gl. 133). In dem Bode-Diagramm in Abb. 44 ist die maximale Verstärkung $V = 30,6 \text{ dB}$ und die Phasenhebung von 70,5° für den veränderten Regler nach dem *Frequenzkennlinienverfahren* abzulesen. Der Ausweichvorgang wurde mit verschiedenen Zeitkonstanten simuliert und $T_v = 85 \text{ s}$ und $T_{PD} = 2,5 \text{ s}$ bestimmt. Der Matlab-Code für die Erstellung der Bode-Diagramme für das *Frequenzkennlinienverfahren* ist im Listing 3 im Anhang D präsentiert.

$$F_R(s) = \frac{85s + 1}{2,5s + 1} \quad (133)$$

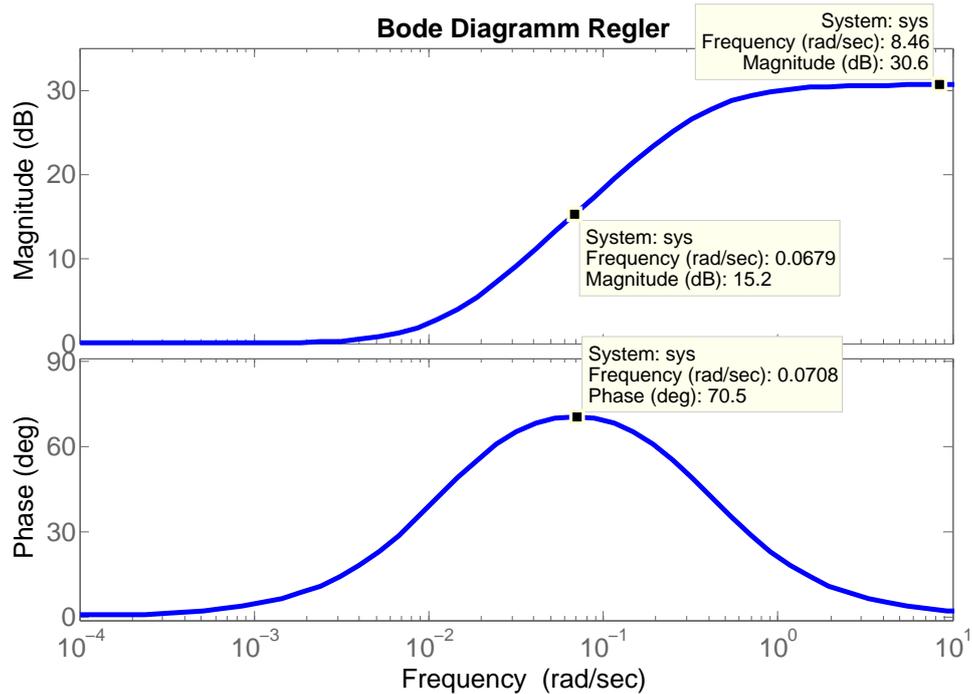


Abb. 44: Regler mit veränderten Zeitkonstanten nach der Analyse der Frequenzkennlinien

Mit dem Regler aus Gl. 133 wird auch der offene Regelkreis verändert (vgl. Gl. 134). Die Phasenreserve beträgt nun 23° und die Durchtrittsfrequenz $13,4 \text{ rad/sec}$ (Abb. 45).

$$F_0(s) = \frac{121,44s^2 + 86,429s + 1}{0,623s^4 + 3,929s^3 + s^2} \quad (134)$$

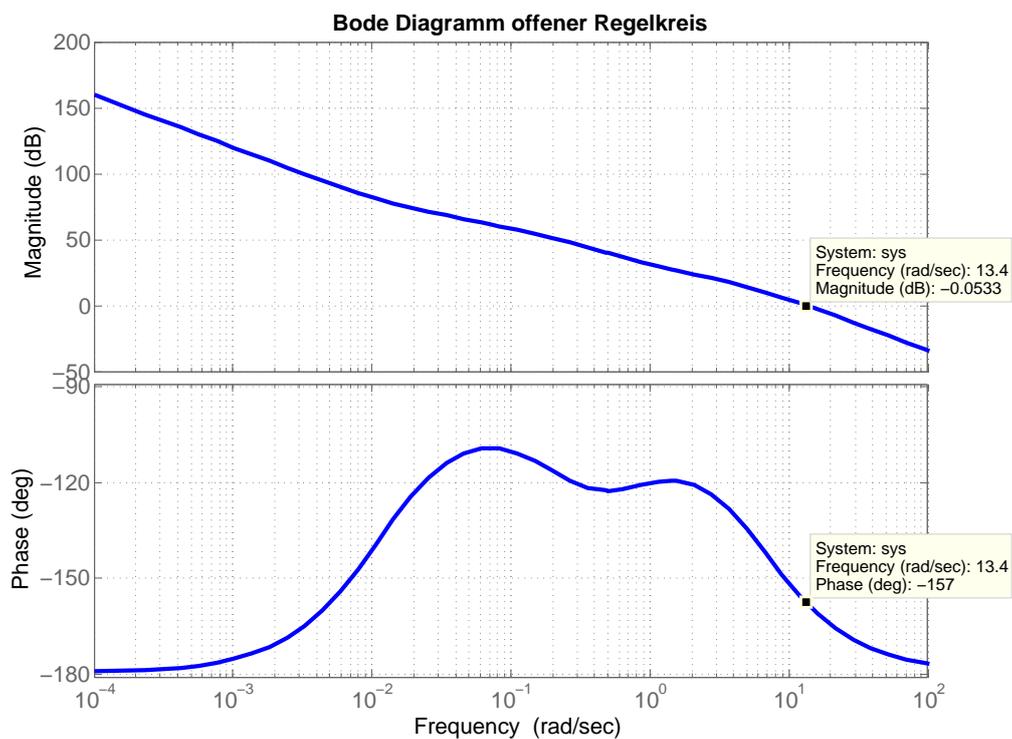


Abb. 45: Offener Regelkreis mit modifiziertem Regler

Durch die Veränderung der Zeitkonstanten des Reglers nach der Analyse der Frequenzkennlinien wurde die Phasenreserve und die Durchtrittsfrequenz erhöht. Dadurch findet eine Dämpfung des Einschwingvorgangs des geschlossenen Regelkreises für eine Sprungantwort, sowie eine Beschleunigung des Erreichens des Führungsverhaltens statt.

Die Verstärkungsfaktoren K_P und K_{S1} sind bisher in dem *Frequenzkennlinienverfahren* nicht berücksichtigt worden, diese werden im Folgenden betrachtet. Der Faktor K_{S1} gehört zu der Regelstrecke und ist bei der Linearisierung des Fahrzeugmodells, abhängig vom Arbeitspunkt, berechnet worden und ist daher nicht veränderbar. K_P ist der Verstärkungsfaktor des Reglers und lässt sich an den offenen Regelkreis in Gl. 134 anpassen, da dieser die Verstärkungsfaktoren nicht beinhaltet.

Die maximale Reglerverstärkung ist durch die Veränderung der Zeitkonstanten von 1,15 dB auf 30,6 dB gestiegen. Das bedeutet, dass der Verstärkungsfaktor K_P vom Regler angepasst werden muss, da sonst der Regler übersteuert und einen zu großen Soll-Lenkswinkel außerhalb des Gültigkeitsbereiches einstellt. Durch Simulationen des Ausweichvorgangs mit verschiedenen Verstärkungsfaktoren wurde K_P so lange verkleinert, bis der Soll-Lenkswinkel zu Beginn des Ausweichvorganges im Gültigkeitsbereich geblieben ist. Mit diesem Verfahren wurde $K_P = 0,04^\circ/m$ ermittelt.

Durch die Verkleinerung der Durchtrittsfrequenz w_c des Reglers, wird ein weiterer Regler als Alternative mit dem *Frequenzkennlinienverfahren* entworfen. Die Zeitkonstante T_{PD} wurde auf 4 s vergrößert, wodurch die Durchtrittsfrequenz w_c auf 10,5 rad/sec sinkt. Daher erreicht der Regelkreis die Führungsgröße langsamer. Dies hat eine weichere Lenkbewegung des SCV während des Ausweichvorganges zur Folge. Die resultierenden Gleichungen für den Regler und den offenen Regelkreis sind in Gl. 135 und Gl. 136 angegeben. Die Bode-Diagramme sind im Anhang I dokumentiert.

$$F_R(s) = \frac{85s + 1}{4s + 1} \quad (135)$$

$$F_0(s) = \frac{121,44s^2 + 86,429s + 1}{0,996s^4 + 5,429s^3 + s^2} \quad (136)$$

7.3.3 Ergebnissimulationen zum Frequenzkennlinienverfahren

Die Simulation des Ausweichvorgangs mit dem ersten Regler nach dem *Frequenzkennlinienverfahren* mit der Überschwingweite-Optimierung ist in Abb. 46 dargestellt und als Regler 2 bezeichnet. Der Sprung der Führungsgröße beträgt 0,5 m bei $t = 1$ s, genau so wie bei 7.2.2. Das Überschwingen des Vorderrades beträgt ungefähr 0,04 m und die Ausregelzeit von Vorder- und Hinterrad ca. 9s.

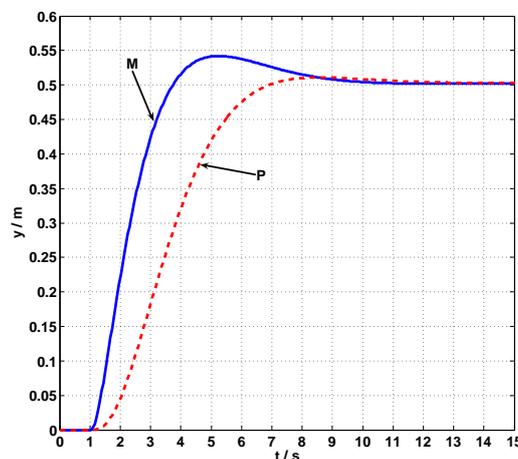
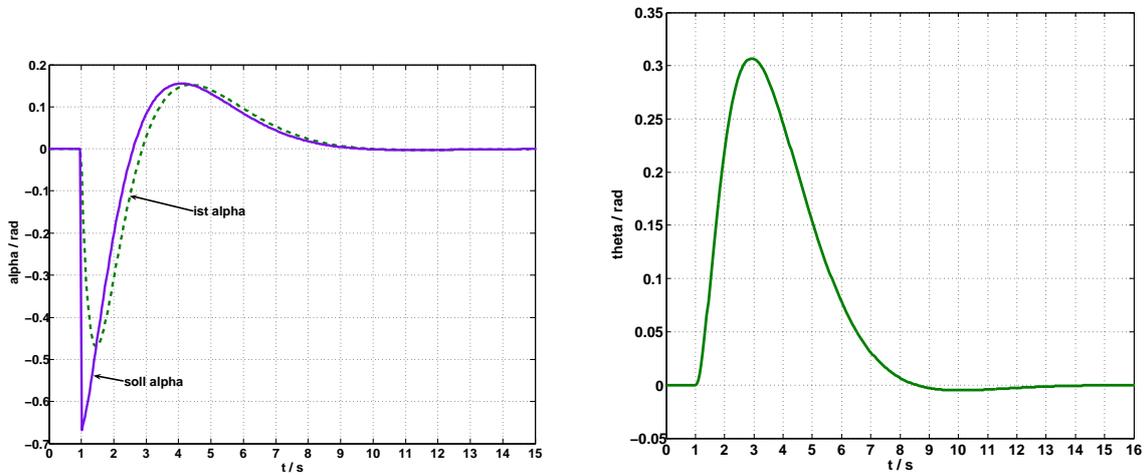


Abb. 46: Regler 2: Vorder- und Hinterrad während des Links-Ausweichens

Der Soll-Lenkwinkel liegt zu Beginn des Ausweichvorgangs bei -37° und der Ist-Lenkwinkel steigt maximal bis ca. -26° an (vgl. Abb. 47a). Damit liegen sowohl Soll- als auch Ist-Lenkwinkel innerhalb des Gültigkeitsbereiches von -90° bis 90° . Der Achswinkel θ des SCV steigt maximal bis 18° an (Abb. 47b).

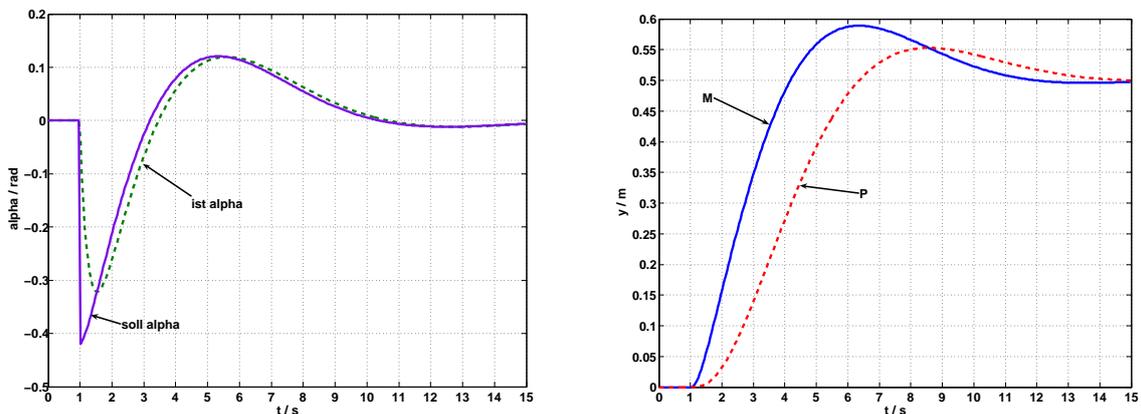


(a) Regler 2: Lenkwinkel während des Ausweichens

(b) Regler 2: Achswinkel während des Ausweichens

Abb. 47: Regler 2

Die Absenkung der Durchtrittsfrequenz w_c mit dem Regler 3 im Regelkreis erreicht eine weichere Lenkbewegung während des Ausweichvorganges und der Soll-Lenkwinkel hat einen geringeren Maximalwert. Im Gegensatz dazu erhöht sich das Überschwingen des Vorderrades auf fast 0,1 m und die Ausregelzeiten der Räder vergrößern sich.



(a) Regler 3: Lenkwinkel während des Ausweichens

(b) Regler 3: Vorder- und Hinterrad während des Ausweichens

Abb. 48: Regler 3

7.4 Auswahl des Reglers

Die bestimmten Regler nach dem *symmetrischen Optimum* und dem *Frequenzkennlinienverfahren* werden mit Kennwerten der Ausweichsimulation verglichen und eine Auswahl getroffen. Es wird dazu die Überschwingweite des Vorderrades sowie die Anregel- und Ausregelzeit

des Vorder- und Hinterrades bewertet. Außerdem muss der Lenkwinkel während des Ausweichvorganges innerhalb des Gültigkeitsbereiches bleiben und ein kleinerer Lenkausschlag ist einem Größeren zu Beginn vorzuziehen, um ruckartige Lenkbewegungen zu vermeiden. Diese Kennwerte sind teilweise gegensätzlich zueinander, z. B. bei einer sehr kurzen Anregelzeit für das Vorderrad ist das Überschwingen desselbigen groß, da der Achswinkel des SCV zum Anregelzeitpunkt hoch ist. Es muss also ein Kompromiss zwischen den Kennwerten gefunden werden, die in Tabelle 9 aufgelistet sind.

Für die Bestimmung der Ausregelzeit wird ein Toleranzband von $\pm 0,01 m$ definiert, in dem sich das Vorder- und Hinterrad nach dem Ausweichvorgang befinden muss. Eine Grundforderung für das System ist neben der Stabilität eine genügende stationäre Genauigkeit. Dabei gilt, dass die Größe des Toleranzbandes von den definierten Anforderungen abhängt. Um eine genügende stationäre Genauigkeit zu erzielen, ist entweder ein I-Glied im Regler einzusetzen oder der Verstärkungsfaktor K_P anzupassen, so dass die Kreisverstärkung erhöht wird. Da in dieser Masterarbeit ein PD-Regler Verwendung findet, musste der Faktor K_P in den Verfahren *symmetrisches Optimum* und *Frequenzkennlinienverfahren* entsprechend gewählt werden. Für weitere Details zur stationären Genauigkeit wird auf [12] verwiesen.

Kennwerte	Abkürzung
Soll- α max.	$\alpha(0)$ [°]
Ist- α max.	α_{ist} [°]
Überschwingweite M	Δy [cm]
Anregelzeit M	t_{an_M} [s]
Anregelzeit P	t_{an_P} [s]
Ausregelzeit M	t_{aus_M} [s]
Ausregelzeit P	t_{aus_P} [s]

Tabelle 9: Die Abkürzungen der Kennwerte

Die Kennwerte für die Beurteilung des Ausweichvorganges wurden mit dem zeitkontinuierlichen Fahrzeugmodell in Dymola-Simulationen gemessen und in der Tabelle 10 dokumentiert (Modell in Anhang J). Mit dem Regelkreis nach dem *symmetrischen Optimum* ist $\alpha(0)$ außerhalb des Gültigkeitsbereiches der Lenkung, wobei dieser durch die Lenkverzögerung α_{ist} nicht verletzt wird. Der Lenkausschlag α_{ist} ist mehr als doppelt so groß wie bei dem Regelkreis mit dem Regler 2 nach dem *Frequenzkennlinienverfahren*. Die Überschwingweite Δy ist um den Faktor 3,7 größer. Im Gegensatz dazu sind die Anregel- und Ausregelzeiten mit dem *symmetrischen Optimum* geringer als die des *Frequenzkennlinienverfahrens*, z. B. ist die Ausregelzeit des Vorderrades das 1,5 fache kleiner.

Verfahren	$\alpha(0)$ [°]	α_{ist} [°]	Δy [cm]	t_{an_M} [s]	t_{an_P} [s]	t_{aus_M} [s]	t_{aus_P} [s]
sym. Opt. 1	-94,8	-60,8	14	2,2	5,0	5,9	4,7
Freq. Ken. 2	-39,0	-26,5	3,8	3,8	7,1	8,7	6,6
Freq. Ken. 3	-24,0	-18,2	8,3	4,2	6,4	11,0	12,8

Tabelle 10: Kennwerte der Ausweichsimulation mit zwei Reglerentwurfverfahren

Für den Regelkreis des Ausweichassistenten wird der Regler 2 nach dem Frequenzkennlinienverfahren ausgewählt, da dieser im Verhältnis zu dem Regler 1 einen geringen Lenkausschlag und eine kleine Überschwingweite hat. Das überwiegt, denn der Faktor zwischen den beiden Reglern bei diesen Kennwerten ist größer, als bei den Anregel- und Ausregelzeiten. Als Alternative für eine weichere Lenkbewegung wird der Regler 3 bei den Tests des Ausweichassistenten ebenfalls berücksichtigt.

Die Eigenschaften der Regler und der Regelkreise und den daraus folgenden dynamischen Aussagen sind in der folgenden Liste benannt:

- *Symmetrisches Optimum*
 - kleinere Phasenreserve \Rightarrow kleinere Dämpfung
 - höhere Verstärkung $(K_p \cdot V) = 3,3 \frac{\circ}{m} \cdot dB \Rightarrow$ höhere Überschwinger
 - offener Regelkreis nur im schmalen Bereich nicht -180°
 \Rightarrow der Regler erzeugt quasi keine Phasenhebung
- *Frequenzkennlinienverfahren*
 - größere Phasenreserve \Rightarrow größere Dämpfung
 - kleinere Verstärkung $(K_p \cdot V) = 1,2 \frac{\circ}{m} \cdot dB$ mit dem Regler 2
 \Rightarrow kleinere Überschwinger
 - offener Regelkreis ist im breiten Frequenzbereich nicht -180°
 \Rightarrow Phasenhebung von Regler und Regelstrecke
 - Durchtrittsfrequenz
 - Regler 2: größere Durchtrittsfrequenz \Rightarrow kleinere Ausregelzeit
 - Regler 3: kleinere Durchtrittsfrequenz \Rightarrow größere Ausregelzeit

7.5 Entwurf des digitalen PD-Reglers

Bei der Ermittlung der Reglerparameter durch die Verfahren *symmetrisches Optimum* und *Frequenzkennlinien* wurden bisher kontinuierliche Systeme betrachtet. Für die Realisierung des Reglers in der Software wurde dieser digitalisiert. In diesem Abschnitt sind die Arbeitsschritte dazu dokumentiert. In dem Verfahren wird die z-Transformation angewendet, bei der die Verzögerung um eine Abtastperiode mathematisch durch z^{-1} beschrieben wird und eine Differenzgleichung entsteht (vgl. [13]).

Die z-Übertragungsfunktion, die aus einem kontinuierlichen System mit der z-Transformation ermittelt wird, ist immer eine Approximation. In dem hier angewendeten Verfahren wird die Substitutionsbeziehung aus Gl. 137 genutzt. Diese entsteht mit der Differenzgleichung aus dem Euler Verfahren und wird in die Übertragungsfunktion für den D-Anteil eingesetzt (vgl. Gl. 138) ([19]).

$$s = \frac{z-1}{Tz} \quad (137) \quad \left. \frac{T_v \bullet s}{1 + T_{PD}s} \right|_{s=\frac{z-1}{Tz}} = \frac{T_v \bullet}{T} \cdot \frac{z-1}{z(1 + T_{PD}/T) - T_{PD}/T} \quad (138)$$

Die z -Übertragungsfunktion des Reglers setzt sich aus dem P- und D-Anteil zusammen. Die Gl. 139 wird durch Erweiterung, Ausmultiplizieren und Zusammenfassen umgeformt. Abschließend folgt eine Multiplikation mit z^{-1} , da z dem nächsten Iterationsschritt in der Zukunft entspricht, dieser aber im System nicht verfügbar ist. Der aktuelle Zeitpunkt z^0 und der letzte Iterationsschritt z^{-1} werden gemessen bzw. vom letzten Iterationsschritt gespeichert.

$$\begin{aligned}
 G_{PD}(z) &= K_p \left(1 + \frac{T_v^\bullet}{T} \cdot \frac{z-1}{z(1+T_{PD}/T) - T_{PD}/T} \right) \\
 &= K_p \left(\frac{T(z(1+T_{PD}/T) - T_{PD}/T)}{T(z(1+T_{PD}/T) - T_{PD}/T)} + \frac{T_v^\bullet}{T} \cdot \frac{z-1}{z(1+T_{PD}/T) - T_{PD}/T} \right) \\
 &= K_p \left(\frac{z(T+T_{PD}) - T_{PD} + zT_v^\bullet - T_v^\bullet}{z(T+T_{PD}) - T_{PD}} \right) \\
 &= K_p \left(\frac{z(T+T_{PD}+T_v^\bullet) - T_{PD} - T_v^\bullet}{z(T+T_{PD}) - T_{PD}} \right) \\
 &= K_p \left(\frac{(T+T_{PD}+T_v^\bullet) + z^{-1}(-T_{PD} - T_v^\bullet)}{(T+T_{PD}) + z^{-1}(-T_{PD})} \right) \tag{139}
 \end{aligned}$$

Aus der z -Übertragungsfunktion Gl. 139 für den PD-Regler sind die Koeffizienten für den digitalen Regler zu bestimmen (vgl. Gln. 140 bis 143).

$$b_0 = (T + T_{PD} + T_v^\bullet) \cdot K_P \tag{140} \qquad a_0 = T + T_{PD} \tag{142}$$

$$b_1 = (-T_{PD} - T_v^\bullet) \cdot K_P \tag{141} \qquad a_1 = -T_{PD} \tag{143}$$

Die Übertragungsfunktion wird in eine Differenzgleichung überführt, wobei die indizierten Variablen e (Eingangssignal) und α (Ausgangssignal) und die Koeffizienten benutzt werden (vgl. Gl. 144). Der aktuelle Zeitpunkt entspricht dem Index n .

$$\alpha_n = (b_0 \cdot e_n + b_1 \cdot z^{-1} \cdot e_{n-1} - a_1 \cdot z^{-1} \cdot \alpha_{n-1}) \cdot \frac{1}{a_0} \tag{144}$$

Das Blockschaltbild für den digitalen PD-Regler, das sich aus der Gl. 144 ergibt, ist in Abb. 49 für den Regler 2 dargestellt. Das Eingangssignal ist die Regeldifferenz e und das Ausgangssignal ist der Lenkwinkel α . Die Abtastzeit T des digitalen PD-Reglers beträgt 0,05 s, dieses entspricht der maximalen Scanfrequenz des Laserscanners von 20 Hz.

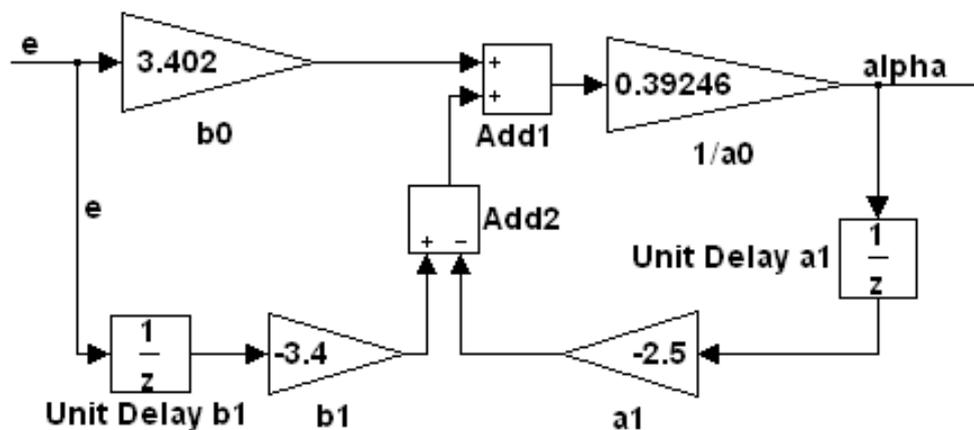


Abb. 49: Blockschaltbild digitaler PD Regler

Für den Regler 3, der eine weichere Lenkbewegung vollzieht, ergeben sich die Koeffizienten aus Gl. 145 bis Gl. 148 für den digitalen Regler.

$$b_0 = 3,402s \cdot ^\circ/m \quad (145)$$

$$b_1 = -3,4s \cdot ^\circ/m \quad (146)$$

$$\frac{1}{a_0} = 0,24704 \frac{1}{s} \quad (147)$$

$$a_1 = -4s \quad (148)$$

8 Software - Modellierung des Ausweichassistenten

In diesem Kapitel werden die Modellierungsdetails der SW-Module, die entwickelt bzw. modifiziert wurden, vorgestellt. Dabei wurde das Konzept aus Kapitel 3 für die Thread- und Softwarestruktur umgesetzt. Durch die Vorstellung der SW-Module sollen deren Arbeitsweise näher gebracht werden. Ausgewählter Sourcecode des Ausweichassistenten ist in Anhang K zu finden.

8.1 Implementierung des räumlichen Mittelwertfilters

Der Laserscanner sendet die Messwerte der Laserscannermessung per CAN-Nachricht an den Koordinierungsrechner, in der jeweils drei Messwerte enthalten sind. Wird eine Messnachricht von der CAN-Schnittstelle empfangen, so wird eine Laserscanner-Nachricht mit dem kopierten Dateninhalt aus der CAN-Nachricht an das SW-Modul *LDComm* gesendet, in dem die Messwerte nacheinander bearbeitet werden. Für das räumliche Mittelwertfilter in der Funktion *auswertung* im Modul *LDComm* stehen somit der aktuelle Messwert und die Messwerte, die zuvor empfangen wurden, zur Verfügung. Die vorigen Messwerte entsprechen dabei den Messwerten links neben dem Aktuellen. Die Messpunkte rechts des aktuellen Messpunktes stehen noch nicht zur Verfügung. Deshalb ist in der Implementierung des Filters der aktuelle Messwert, der empfangen wurde, der räumlich rechteste Messwerte, der zu der aktuellen Mittelung genutzt wird. Bei dem Empfang des Messwerts x_n wird der Mittelwert aus den Messwerten $x_{n-(N-1)}$ bis x_n berechnet und in $y_{n-(N-1)/2}$ gespeichert (vgl. Abb. 50). Werden die Messwerte x_2 und x_3 bei einer Filterbreite von $N = 5$ empfangen, so werden zunächst die Messwerte für y_n übernommen und überschrieben, wenn der räumlich rechteste Punkt für die jeweilige Mittelung empfangen wird.

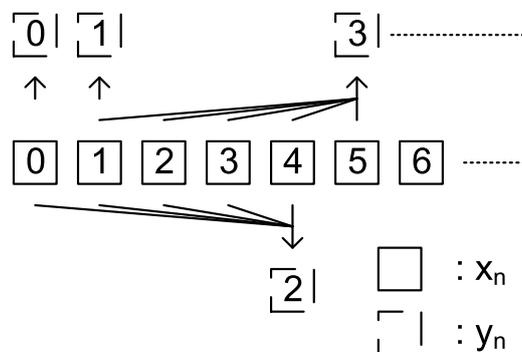


Abb. 50: Softwarerealisierung des RM-Filters

Der Speicherplatzbedarf für das Array zur Mittelwertberechnung der beiden Filtermethoden ist:

- History-MA-Filter: $11 \text{ Messwerte} \cdot 181 \text{ Scannwinkel} \cdot 16 \text{ Bit} = 31,856 \text{ kB}$
- Räumliches Mittelwertfilter: $2 \text{ Messwerte} \cdot 181 \text{ Scannwinkel} \cdot 16 \text{ Bit} = 5,792 \text{ kB}$

Das bedeutet, dass das räumliche Mittelwertfilter ca. 26 kB weniger Speicher verwendet als das History-MA-Filter.

8.2 Implimentierungsdetails der Steuerung des Ausweichassistenten

Der Start des Ausweichvorgangs und die Entscheidung der Ausweichrichtung

Der Ausweichassistent muss aus der Umgebungsinformationen des SCV extrahieren, ob ein Ausweichvorgang gestartet werden muss und im positiven Fall die Ausweichrichtung vorgeben. Das Collision Avoidance System (*CA_System*) ruft zyklisch die Objekte der Umgebung von der Objekterkennung ab und lässt von der Aufprallerkennung das Objekt mit der geringsten Aufprallzeit bestimmen (vgl. Kapitel 3). Mit dem potentiellen Hindernisobjekt wird in dem Modul *AusweichStatusRegl* der Ausweichstatus für das Links- und Rechts-Ausweichen ermittelt. Bei Unterschreitung eines definierten Abstands zum Hindernisobjekt oder wenn das Verhältnis Ausweichbreite zu Abstand zum Hindernis zu gering ist, wird der Status für die jeweilige Richtung aktiviert. Dieses ist nur der Fall, wenn sich das Objekt in einem bestimmten Winkelbereich vor dem SCV befindet. Die empirisch bestimmten Kriterien für die Statusaktivierung sind:

$$- K_{ax} < 1,5 m \ \&\& \ |K_{phi}| < 0.17^\circ$$

- Links-Ausweichen

$$K_{ax} < (L_{ay} + SSA + S_l) \ \&\& \ |K_{phi}| < 0.17^\circ$$

- Rechts-Ausweichen

$$K_{ax} < |(R_{ay} - SSA + S_r)| \ \&\& \ |K_{phi}| < 0.17^\circ$$

SSA : Sicherheitsseitenabstand

S_l : Linke Grenze des SCV

S_r : Rechte Grenze des SCV

Das *CA_System* sendet für die Bestimmung der Ausweichrichtung die Signale L_{on} bzw. L_{off} und R_{on} bzw. R_{off} an den Entscheidungsautomaten für die Ausweichrichtung, abhängig von den Ausweichstatus Links und Rechts.

Der Automat ist mit dem *AIRA*⁴-Konzept realisiert worden und nutzt die *airaRT* Bibliothek. Diese stellt eine Laufzeitumgebung für die Implementierung von hierarchischen Zustandsautomaten zur Verfügung. Für weitere Details wird auf [5] und [20] verwiesen.

In dem Fall, dass sich kein Hindernis vor dem SCV befindet, ist der Entscheidungsautomat *EntscheiderAbstReglFSM* in dem Zustand Z_{ENT} . Bei dem Empfang einer L_{on} -Nachricht, findet ein Wechsel zu dem Zustand Z_{ENT_L} statt. Folgt die R_{on} -Nachricht, wird zu dem Zustand Z_{ENT_LR} gewechselt und dem *CA_System* ein Ausweichmanöver nach Rechts vorgeben (vgl. 51). Für das Links-Ausweichen gilt Entsprechendes. Das Konzept des Entscheidungsautomaten ist [5] entnommen.

⁴AIRA: Automaten in reaktiven Anwendungen

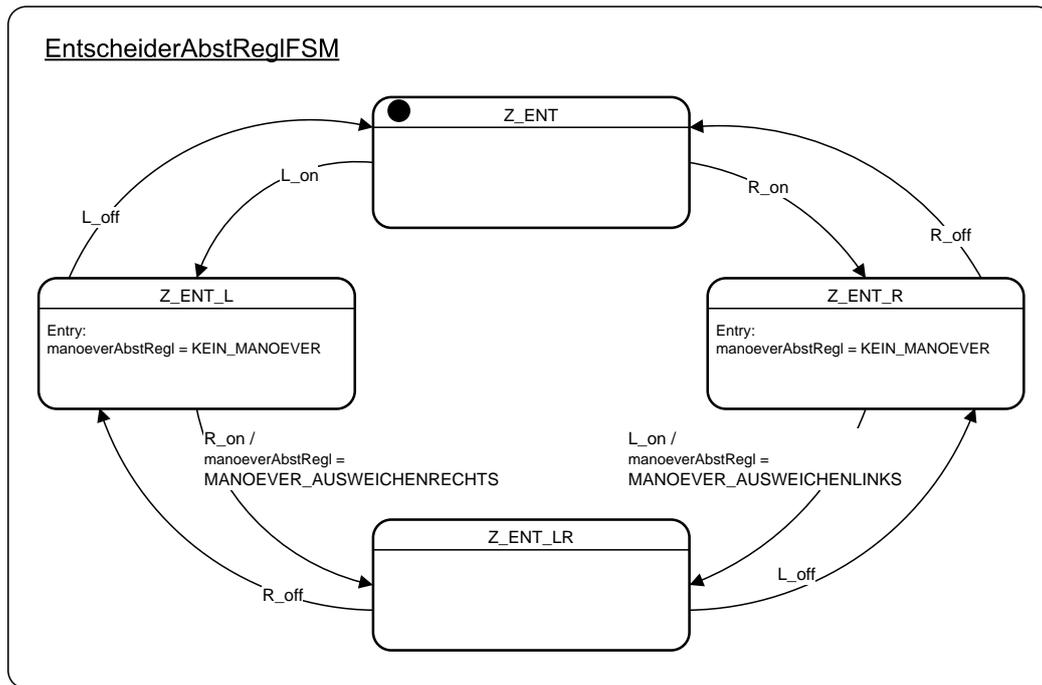


Abb. 51: Entscheidungsautomat für die Ausweichrichtung

Der Ausweichassistent während des Ausweichvorgangs

Ein Ausweichvorgang wird vom Entscheidungsautomaten ausgelöst. Dadurch startet das *CA_System* einen eigenständigen Thread für das Ausweichmanöver mit der Identifikationsnummer des Hindernisobjekts. Zu Beginn werden die folgenden Einstellungen vom Ausweichassistenten vorgenommen:

- Die Führungsgröße des Regelkreises wird auf die Ausweichbreite gesetzt.
- Die Soll-Geschwindigkeit des SCV für das Manöver wird auf die aktuelle Ist-Geschwindigkeit gesetzt.
- Die Fahrzeugposition des SCV wird auf *vor dem Hindernis* bzw. *schräg vor dem Hindernis* gesetzt.
 - *Vor dem Hindernis* $\Rightarrow |L.x - R.x| < 0,1 m$
 - *Schräg vor dem Hindernis* $\Rightarrow |L.x - R.x| \geq 0,1 m$
- Der Steuerautomat wird in den *CA_System*-Mode versetzt, d. h. die Soll-Geschwindigkeit und der Soll-Lenkwinkel werden vom Ausweichassistenten vorgegeben.

Die Verarbeitungskette des Ausweichassistenten ist in Abb. 52 verdeutlicht. Das *CA_System* ruft in zyklischen Abständen von 50 ms die aktuelle Position des Hindernisobjekts ab, dies entspricht der Abtastfrequenz des Laserscanners und der Sendefrequenz der Nachrichten auf dem CAN-Bus für den Soll-Lenkwinkel. Die Objekterkennung liefert die Punkte *K*, *L* und *R* in kartesischen Koordinaten, die auf das Hinterrad *P* bezogen sind (vgl. Abschnitt 4.1). Das geometrische Modell für die relative Positionsbestimmung des SCV ist aber für Polarkoordinaten auf den Sensor *S* bezogen. Deshalb werden die Koordinaten vor der Berechnung der Abstandswerte und des Achswinkel transformiert. Die Berechnung ist abhängig vom aktuellen Bereich, in dem sich das SCV befindet (vgl. Abschnitt 6.2). Während des Ausweichvorgangs durchfährt das SCV die Bereiche *vor dem Hindernis*, *schräg vor dem Hindernis* und *neben dem Hindernis*. Für

den PD-Regler, der für das Vorderrad M des SCV ausgelegt ist, werden anschließend die Abstandswerte zu M_{ax} und M_{ay} transformiert. Der Abstand M_{ay} wird dem Regelkreis als aktuelle Messung zugeführt und der Soll-Lenkswinkel wird für den Ausweichfortschritt genutzt. Als Annahme gilt, dass das Hindernisobjekt eine rechteckige Form hat, denn in dem Fall steht das SCV parallel zum Hindernis, wenn sich der Achswinkel nach der Ausweichkurve im Toleranzbereich von 0° befindet. Nach der Beendigung des Ausweichmanövers wird der CA_System -Mode im Steuerautomaten wieder deaktiviert.

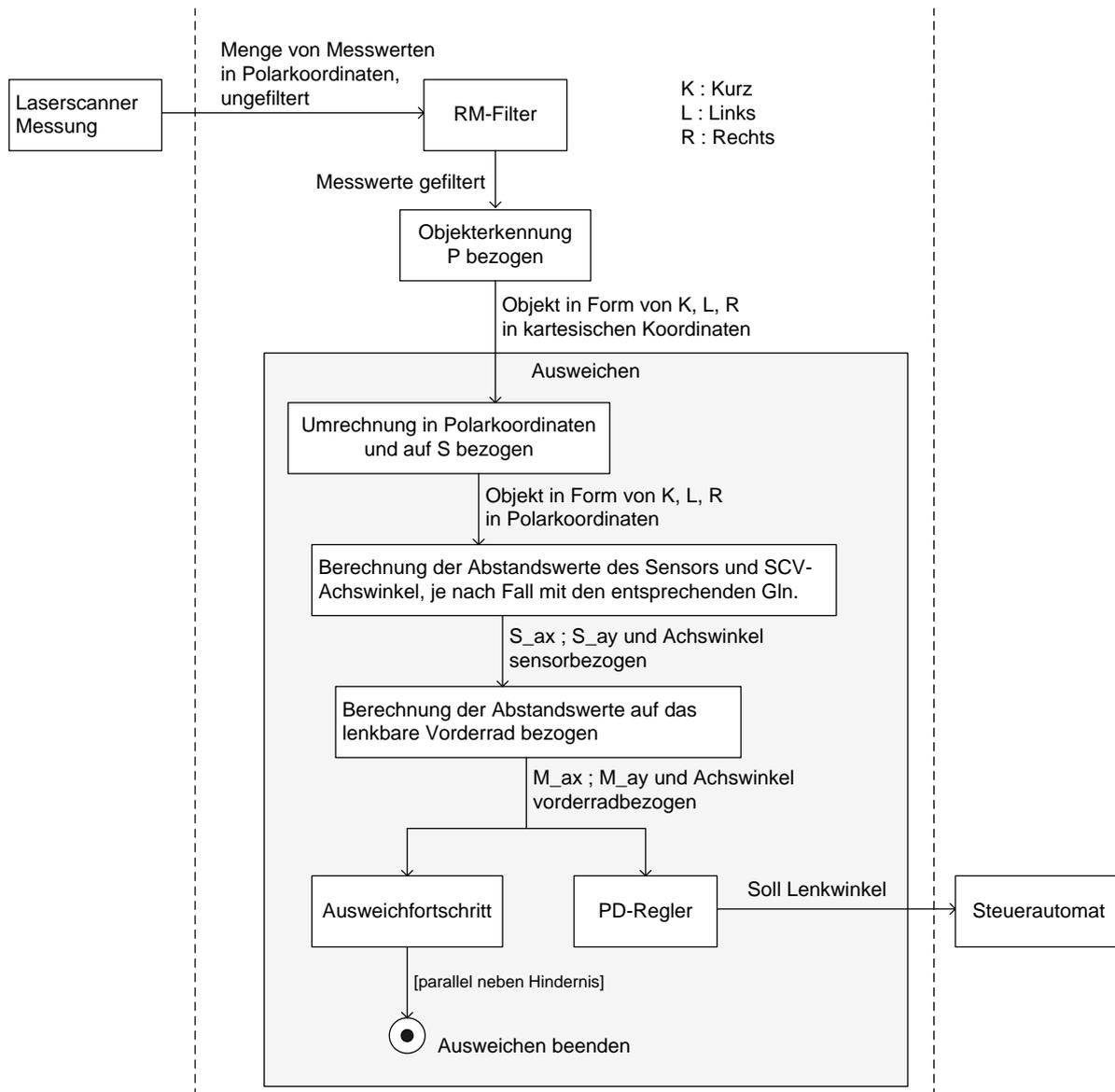


Abb. 52: Verarbeitungskette des Ausweichassistenten während des Ausweichvorgangs

Der PD-Regler

Der digitale PD-Regler ist in einem eigenen Modul entsprechend dem Abschnitt 7.5 umgesetzt. Der Funktion pd_regler werden die Führungsgröße und die relativen Positionsdaten des SCV übergeben, sprich die Abstände zum Hindernis in x - und y -Richtung. Die $1/z$ -Glieder wurden durch Variablen umgesetzt, die die Regeldifferenz und den Soll-Lenkswinkel zwischen zwei Reglerzyklen zwischenspeichern. Durch die Diskretisierung des Reglers in Abschnitt 7.5

lässt sich dieser durch Addition und Multiplikation abbilden. Mit der Funktion *reset_regler* wird der Regler zu Beginn des Ausweichvorgangs zurückgesetzt, dazu werden die $1/z$ -Glieder auf 0 gesetzt.

Beendigungskriterien des Ausweichassistenten

Der Ausweichassistent soll sich selbst beenden, wenn sich das SCV parallel neben dem Hindernis befindet, denn zu dem Zeitpunkt hat der Ausweichvorgang erfolgreich stattgefunden und das SCV befindet sich in einer definierten Position. Dafür wird der Achswinkel des SCV aus der Berechnung in Kapitel 6 genutzt. Der Benutzer hat am Leitstand die Möglichkeit, den Ausweichassistenten abzubrechen, um in der Entwicklungsphase eine Kollision mit Umgebungsobjekten bei Fehlverhalten zu vermeiden. Die Kriterien, die den Ausweichassistenten beenden bzw. abbrechen, werden im Folgenden genannt.

- Das SCV befindet sich in dem Bereich neben dem Hindernis und der Achswinkel θ aus der Laserscannermessung ist im Toleranzband von 0° .
- Der Benutzer am Leitstand betätigt nicht mehr das Vorwärtspedal

Die Thread-Prioritäten und das Scheduling Verfahren des Softwaresystems

Die Threads in dem Softwaresystems des Ausweichassistenten nutzen das Round-Robin Scheduling Verfahren, d. h. jeder Thread bekommt einen bestimmten Zeitschlitz an Rechenzeit zugeteilt. Ist der Auftrag vor Ende des Zeitschlitzes abgearbeitet, so blockiert sich der Thread selbst und gibt das System wieder frei. Im anderen Fall erfolgt eine Preemption nach Ablauf der Zeit [21]. Die Prioritäten der Threads, die auf dem Koordinierungsrechner laufen, sind in Tabelle 11 aufgelistet. Die Standardeinstellung ist eine Priorität von 10, davon weichen zwei Threads ab. Zum Einen der *Timer-Tick*-Thread, der die höchste Priorität von 20 hat, da dieser für das zeitliche Verhalten des Steuerungsautomaten verantwortlich ist. Zum Zweiten der *Entscheider* für die Ausweichrichtung mit einer Priorität von 19, damit die Statusänderung für eine Ausweichrichtung mit einer kurzen Bearbeitungszeit berücksichtigt wird.

Thread	Funktion	Datei	Priorität
Main	main	main.c	10
CAN-Empfang	can_recv_thread	CANComm.c	10
CAN-Senden	can_send_thread	CANComm.c	10
Steuerautomat	automat_thread	automat.c	10
Timer-Expired	expired_timers_thread	automat.c	10
Timer-Tick	timer_tick_thread	automat.c	20
Situationserfassung	lddata_recv_thread	LDComm.c	10
Situationsanalyse	ca_system_thread	CA_System.c	10
Entscheider	entscheiderAbstRegl_thread	EntscheiderAbstReglFSM.c	19
Aktionsausführung	ausweichen_links_abst_regl	LDComm.c	10

Tabelle 11: Die Thread-Prioritäten des Softwaresystems

Die Funktionen, die die Threads blockieren und dadurch das System vor Ablauf des Zeitschlitzes freigeben, sind in Tabelle 12 dokumentiert. Der *Main*-Thread läuft in einer Endlosschleife, da er nach der Systeminitialisierung keine weiteren Aufgaben hat. Folglich ist seine Sleep-Dauer lang. Der Thread *CAN-Empfang* wird über einen Interrupt aktiv, wenn an der CAN-Schnittstelle eine neue Nachricht anliegt. Die Threads mit der blockierenden Funktion *msg_dequeue()* warten auf Nachrichten von anderen Softwaremodulen. Wird eine Nachricht empfangen, so wird der entsprechende Auftrag abgearbeitet. In zyklischen Zeitabständen wird ein Auftrag bei einem *nanosleep()* zur Ausführung gebracht. Im Falle der Aktionsausführung soll eine Frequenz von 20 Hz erreicht werden, durch Sleep-Dauer, Rechenzeit und Wartezeit für den Prozessorzugriff. Diese entspricht der Frequenz der Laserscannermessung und der CAN-Nachrichten für den Soll-Lenkwinkel. Das Timing der Aktionsausführung wird in Kapitel 9.3 näher betrachtet.

Thread	blockierende Funktion	Sleep-Dauer
Main	sleep()	36000 s
CAN-Empfang	InterruptWait()	
CAN-Senden	msg_dequeue()	
Steuerautomat	msg_dequeue()	
Timer-Expired	nanosleep()	8 ms
Timer-Tick	nanosleep()	8 ms
Situationserfassung	msg_dequeue()	
Situationsanalyse	nanosleep()	1 ms
Entscheider	msg_dequeue()	
Aktionsausführung	nanosleep()	48 ms

Tabelle 12: Blockierende Funktionen der Threads

Die Erstellung verschiedener Softwareversionen

Das Softwareprojekt des Ausweichassistenten stellt Versionen mit unterschiedlichen Funktionalitäten für den jeweiligen Bedarf zur Verfügung. Der Entwickler kann z. B. den Laserscanner deaktivieren und die Simulation einschalten und somit unabhängig vom SCV den Ausweichassistenten testen. Die jeweiligen Einstellungen sind zentral in der Header-Datei *version* auszuwählen. Die SW-Module inkludieren diese Headerdatei und passen sich dadurch den geforderten Einstellungen an.

- *FAUST_ON*
Aktiviert den Steuerautomaten für die Lenk- und Antriebsmotoren.
- *LASER_ON*
Schaltet den Laserscanner ein und startet die Messung.
- *SEND_LD_SET_MSG*
Senden der Konfigurationsnachrichten für den Laserscanner bei der Initialisierung und Ausführung eines CPU Resets zur Aktivierung der Einstellungen.

- Startwinkel Scanbereich
- Endwinkel Scanbereich
- Baudrate des CAN-Buses
- Scanfrequenz
- Winkelauflösung

Werden die Konfigurationsnachrichten nicht gesendet, wird der Laserscanner während des Systemstarts angehalten und in den *IDLE*-Mode versetzt.

- *SIMULIERE_SCANDATEN*
Simuliert das im *Simulator* definierte Hindernisobjekt und generiert die entsprechenden Laserscanner-Nachrichten für das Modul *LDComm*.
- *DEF_ENTSCHEIDER_AUSWEICH_ABSTREGL*
DEF_ENTSCHEIDER_BREMS_AUSWEICH
Auswahlmöglichkeiten zwischen dem Ausweichassistenten mit der Abstandsregelung aus dieser Masterarbeit und dem Ausweich- und Bremsassistent aus [5] und [6].
- *THREAD_TIMINGS*
Aktiviert die Ansteuerung des Parallelports für die Messung der Ausführungszeiten der Threads.

8.3 Bestimmung der Wegdifferenzen mit der Odometrie

Die Odometrie bestimmt die relative Position seit dem Startpunkt durch die Beobachtung der Fahrzeugräder, indem die Wegdifferenzen aufaddiert werden. Mit der Odometrie lässt sich die Messung der Abstandswerte zum Hindernis und des Achswinkels des SCV durch das Kalman-Filter durch eine Sensordatenfusion präzisieren (vgl. Kapitel 11). Der Begriff Odometrie stammt aus dem griechischen von *hodós* - *Weg* und *métron* - *Maß*, zusammengesetzt ergibt sich: *Wegmessung* [22]. Die Odometrie eignet sich für kurze Strecken, sie wird allerdings bei langen Strecken ungenauer, da die Wegdifferenzen aufaddiert werden. Zusätzlich entsteht bei dem Verfahren Messrauschen durch das Überfahren einer Bodenwelle oder den Schlupf der Räder [8].

Im SW-Modul *Odometrie* werden nicht direkt das Vorder- und Hinterrad beobachtet, sondern die Ist-Nachricht des CAN-Buses für die Geschwindigkeit und den Lenkwinkel genutzt. So lassen sich die Gleichungen aus der Trapezintegration von Abschnitt 5.3 nutzen, um die Wegdifferenzen zu berechnen. Die Odometrieberechnungen wurden in einem eigenen Thread realisiert. Die Ist-Daten des SCV sendet die CAN-Schnittstelle an das Modul.

Das *Odometrie*-Modul stellt folgende Funktionen nach außen zur Verfügung:

- *resetOdometrieDaten*
Es werden die Odometriedaten und alle Hilfsvariablen der Gleichungen initialisiert.
- *getOdometrieDaten*
Gibt die Odometriedaten mit Δx_m , Δy_m und θ zurück.
- *setOdometrieAchswinkel*
Setzen des Achswinkels auf den übergebenen Wert.

- *printOdometrieDaten*

Die aktuellen Odometriedaten werden zu Testzwecken auf der Kommandozeile ausgegeben.

8.4 Erweiterung der QNX-Photon Testumgebung

Für den Test des Ausweichassistenten wird die Visualisierung auf dem Koordinierungsrechner genutzt (vgl. Abschnitt 4.5). In dieser Masterarbeit wurde die Testumgebung erweitert, um zum Einen die Lage des Hindernisobjekts bei der Simulation zu erkennen und zum Anderen, um eine Orientierung für die Ausweichkurve zu erhalten. Dazu werden folgende Elemente zusätzlich visualisiert:

- Die idealisierte Ausweichkurve in gelb.
- In der Simulation der Umriss des Hindernisobjekts in schwarz.

Die idealisierte Ausweichkurve wird durch eine Kosinuskurve berechnet, die durch die Ausweichbreite $y_{abstand}$ und den Abstand zum Hindernis $x_{abstand}$ an das jeweilige Ausweichmanöver angepasst wird. Die Kosinuskurve lässt sich mit den Parametern in Gl. 149 anpassen, die in Tabelle 13 beschrieben sind. Für die konkrete Berechnung der idealisierten Ausweichkurve wird die Gl. 150 genutzt. Der Thread für das Ausweichmanöver übergibt mit der Funktion *set_ideal_kurve* zu Beginn die Ausweichkurve des Ausweichvorgangs dem Visualisierungsthread und aktualisiert danach zyklisch den Ausweichfortschritt mit *set_ausgabe_parameter*. Mit den Ausgabeparametern wird dem Visualisierungsthread mitgeteilt, welcher Restanteil der Kurve noch nicht abgefahren wurde und folglich ausgegeben werden muss. Außerdem findet die Übergabe des Achswinkels statt, da der Koordinatenursprung im Hinderrad des SCV liegt und deshalb die visualisierte Kurve während des Ausweichmanövers in der Größe des aktuellen Achswinkels gedreht werden muss.

$$y = a \cdot \cos(b \cdot x + c) + d \quad (149)$$

Abk.	Bedeutung	Beschreibung
a	Amplitudenänderung	abhängig von der Ausdehnung des Hindernisses
b	Frequenzänderung	abhängig von der Entfernung des Hindernisses
c	Phasenänderung	Verschiebung \cos auf der x -Achse
d	y-Achsenabschnitt	Verschiebung \cos auf der y -Achse

Tabelle 13: Parameter für ideale Ausweichkurve

$$kurve(i + 1) = \frac{y_{abstand}}{2} \cdot \cos\left(\frac{1}{x_{abstand}/2} \cdot x(i + 1) + \pi\right) + \frac{y_{abstand}}{2} \quad (150)$$

Bei der Simulation des Hindernisobjekts aktualisiert der Scan-Simulator zyklisch die vier Eckpunkte mit der Funktion *set_simulation_objekt* in dem Visualisierungsthread. Die Objektpunkte des Hindernisobjekts werden in x- und y-Richtung um -10 Pixelpunkte verschoben und in der grafischen Oberfläche angezeigt, damit die rote und grüne Linie für die linke bzw. rechte Objekthälfte des Hindernisobjekts nicht verdeckt werden. Die Visualisierung einer Ausweichsimulation ist in Abb. 53 dokumentiert.

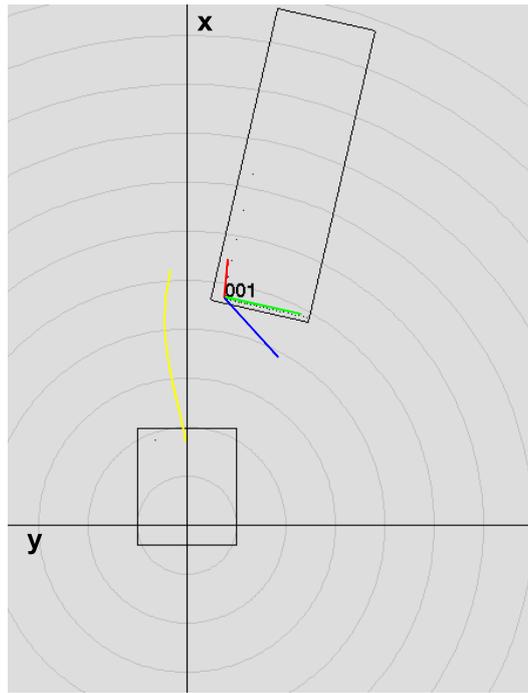


Abb. 53: Visualisierung der erweiterten Simulationsumgebung

9 Messungen der SCV-Ausweichvorgänge

In diesem Kapitel erfolgt der Nachweis über die Funktionsweise des Ausweichassistenten in der Simulation und in der realen Umgebung. Dazu werden für beide Ausweichrichtungen und für unterschiedliche Hindernisobjekte Testszenarien definiert und durchgeführt.

9.1 Simulationen des SCV-Ausweichvorgangs

Der Ausweichassistent wird zuerst in der Simulation getestet, damit die Hardware des SCV im Falle einer Fehlfunktion keinen Schaden nimmt. Die Fahrzeugeigenbewegung und das Umgebungsmodell werden simuliert und in der Photon-GUI visualisiert (Kapitel 4, 8.4).

Im ersten Testszenario wird ein Rechts-Ausweichen simuliert. Das SCV befindet sich zu Beginn in dem Bereich *vor dem Hindernis*, da der Laserscanner auf der Höhe des Objekts ist. Für die Ausweichbreiten gilt: $y_{max_r} < y_{max_l}$, deshalb weicht das SCV nach rechts aus. Das Hindernisobjekt ist 2 m breit und 3 m lang und steht zu Simulationsstart in 4,85 m Entfernung (vgl. Abb. 54). Die Geschwindigkeit des SCV beträgt 0,4 m/s.

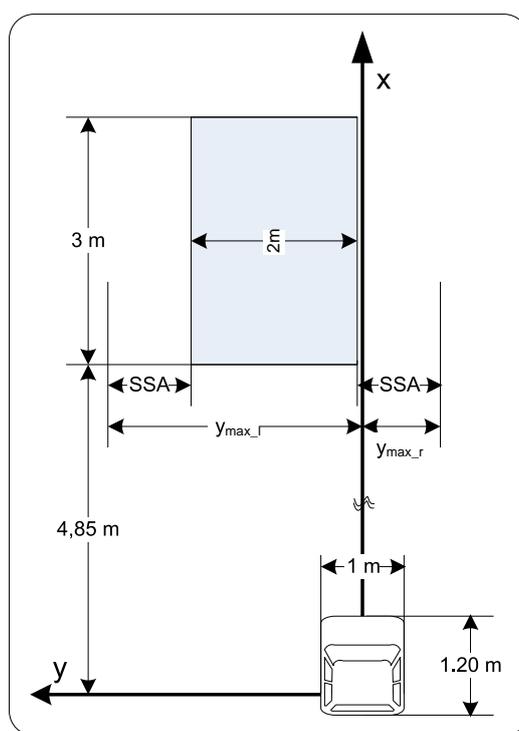
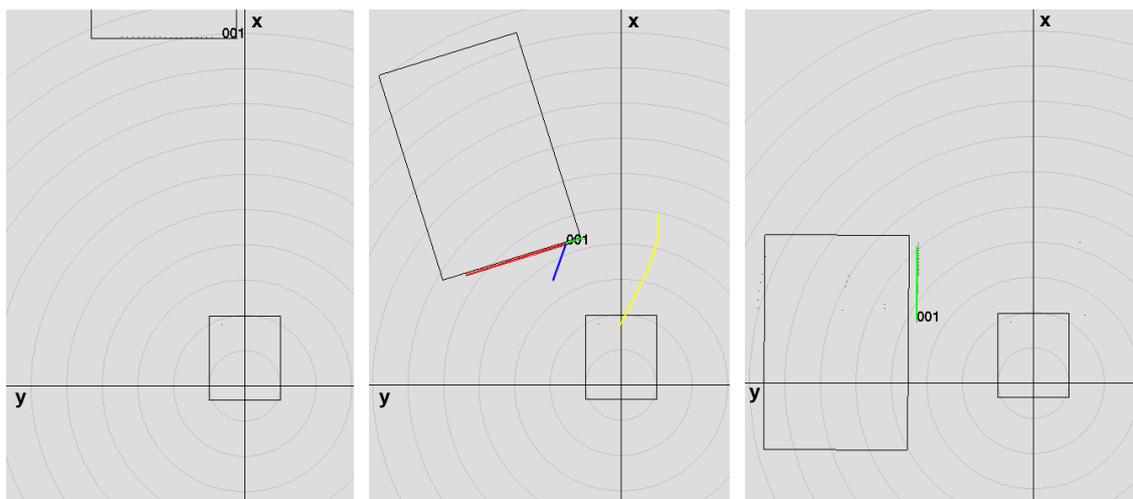


Abb. 54: Testszenario 1: Rechts-Ausweichen

Das Ergebnis der Simulation ist mit drei Screenshots der Visualisierung dokumentiert (vgl. Abb. 55). Zu Beginn befinden sich das Hindernisobjekt und das SCV in der Ausgangsposition (Abb. 55 links). Detektiert der Ausweichassistent eine Kollision, so startet dieser einen Ausweichvorgang und die *idealisierte Kurve* wird in gelb angezeigt. Da der Koordinatenursprung im Hinterrad des SCV liegt, bewegt sich das Hindernisobjekt bei dem Rechts-Ausweichen in die linke Hälfte des Koordinatensystems (Abb. 55 mitte). Nach dem Beenden des Ausweichvorgangs steht das SCV parallel zum Hindernis mit einem Seitenabstand von ca. 1,5 m (Abb. 55 rechts).

Abb. 55: Simulation Rechts-Ausweichens mit Start *vor dem Hindernis*

Für das zweite Testszenario ist das Hindernisobjekt nach links um 3 m verbreitert worden, also ist es insgesamt 5 m breit. Die anderen Simulationsparameter wurden nicht verändert. Durch dieses Szenario wird der Ausweichvorgang für breitere Hindernisse getestet, da sich die geometrischen Modelle für die relative Positionsbestimmung in den Bereichen *vor dem Hindernis* und *schräg vor dem Hindernis* auf die halbe Hindernisbreite beziehen. Zusätzlich wird der Abstand zwischen halber Hindernisbreite und der Vorderradposition für die Berechnung des Sprungs der Führungsgröße im Regelkreis berücksichtigt. Der Ausweichvorgang für dieses Szenario ist in Abb. 56 dokumentiert. In diesem Test kommt das SCV ebenfalls mit einem Seitenabstand von 1,5 m zum Stehen.

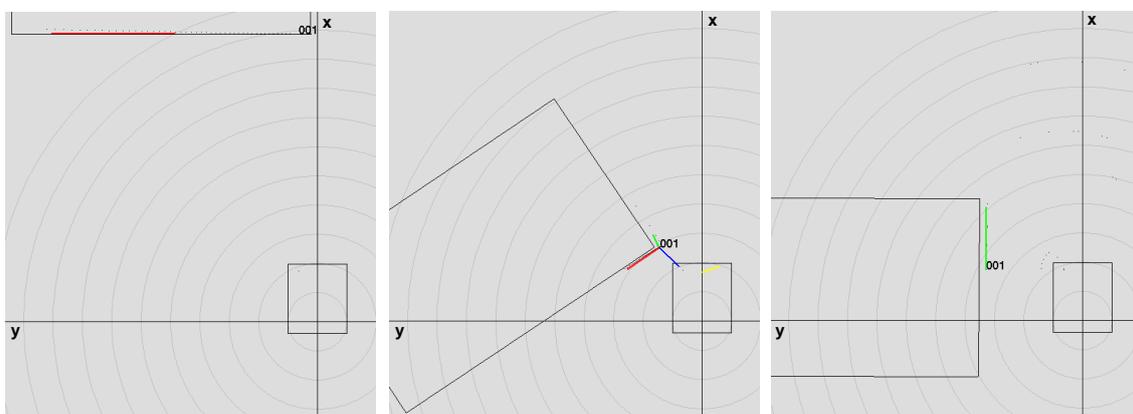


Abb. 56: Simulation des Rechts-Ausweichen mit einem breiten Hindernisobjekt

Das Links-Ausweichen mit Start des Ausweichmanövers in dem Bereich *schräg vor dem Hindernis* wird in Szenario drei getestet. Dazu befindet sich das Hindernisobjekt mit den Außenmaßen 1 m x 3 m in einer Entfernung von 4,85 m (vgl. Abb. 57). In dem Fall gilt für die betrachteten Ausweichbreiten $y_{max_l} < y_{max_r}$, d.h. es wird ein Links-Ausweichen vollzogen. Da die Position des Laserscanners nicht auf der Höhe des Hindernisses ist, wird das geometrische Modell zur relativen Ortsbestimmung im Bereich *schräg vor dem Hindernis* gestartet. Am Ende des Ausweichvorgangs steht das SCV neben dem Hindernis (vgl. Abb. 58).

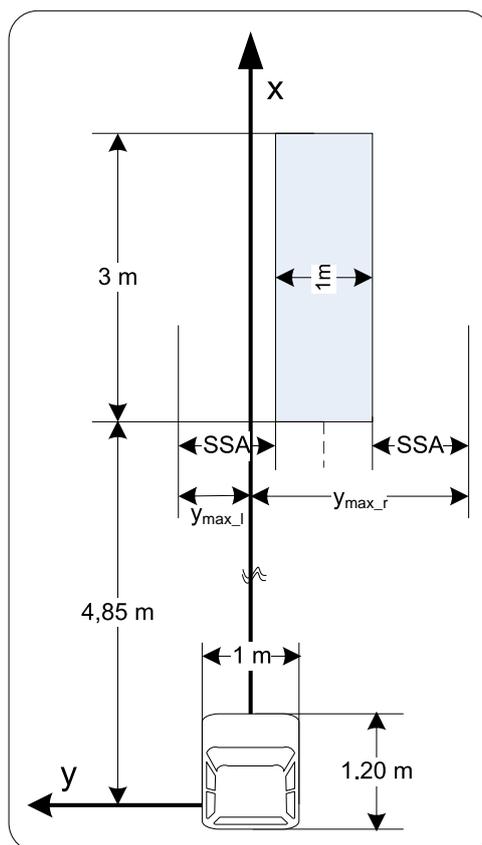


Abb. 57: Testszenario 3: Links-Ausweichen

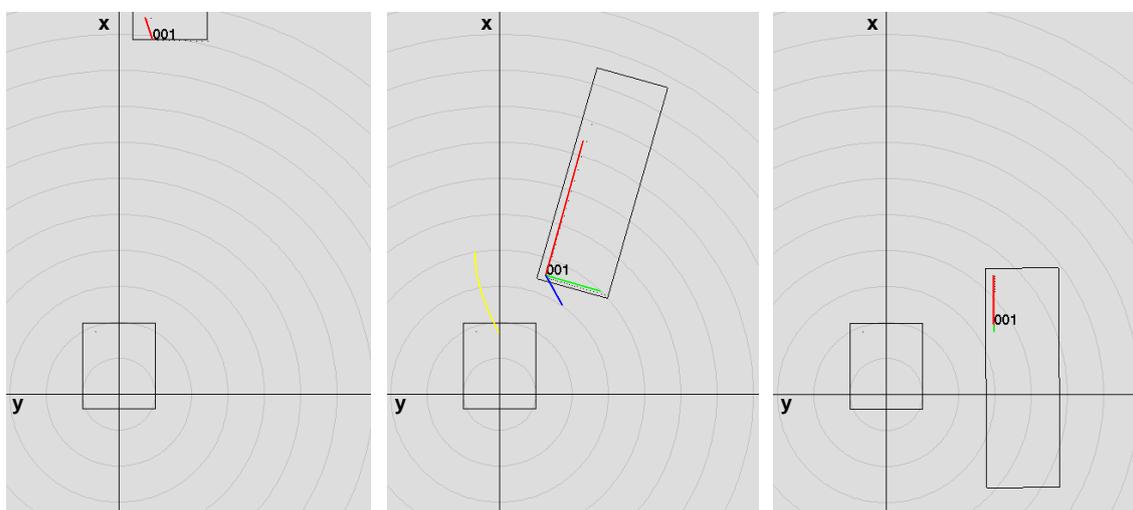
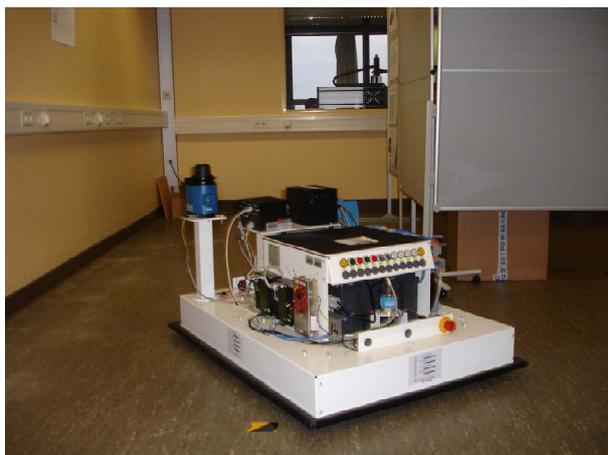


Abb. 58: Simulation des Ausweichvorgangs

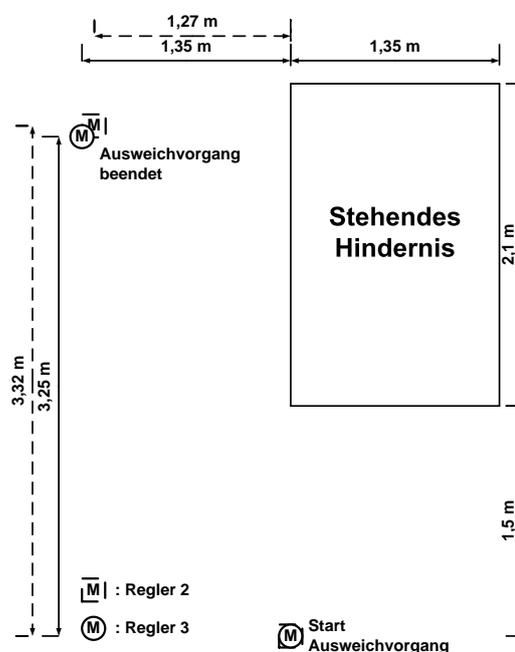
9.2 Test des Ausweichassistenten mit dem SCV

Der Test in der realen Umgebung mit dem SCV wurde im Laborraum der Hochschule mit drei Stellwänden für das Hindernis durchgeführt (vgl. Abb. 59a). Das SCV fuhr gesteuert vom Leitstand mit einem Soll-Lenkwinkel von 0° gerade auf das Hindernis zu, d. h. näherungsweise mit einem Achswinkel von 0° . Die Soll-Geschwindigkeit vom Leitstand wird zu Ausweichbeginn übernommen und während des kompletten Ausweichmanövers konstant beibehalten.

Das Szenario für die Verifikation des Links-Ausweichens entspricht prinzipiell dem aus der Simulation (Abb. 57). Das SCV befindet sich zu Beginn bereits in dem Bereich *schräg vor dem Hindernis*. Allerdings sind die Außenmaße des Hindernisses und die relativen Abstände den Räumlichkeiten des Labors angepasst. Die Position des SCV bei Ausweichbeginn und bei Beendigung des Ausweichvorgangs sind in Abb. 59b für den Regler 2 und den Regler 3 dokumentiert. Die Situationsanalyse des Assistenten löst bei einem Abstand in x-Richtung von 1,5 m aus, dies ist unabhängig von dem gewählten Regler. Nach Beendigung des Ausweichvorgangs steht das SCV mit dem Regler 2 1,27 m seitlich neben dem Hindernis. Mit dem Regler 3 beträgt der Abstand 1,35 m. Das Fahrzeug steht zu diesem Zeitpunkt nicht parallel zu dem Hindernisobjekt, da der Regelvorgang auf die Führungsgröße nicht abgeschlossen ist. Der Ausweichvorgang wird allerdings dennoch beendet, da das Ende des Hindernisobjekts fast erreicht ist und somit ein Achswinkel im Toleranzbereich von 0° bestimmt (vgl. 6.5). Durch die längere Ausregelzeit des Reglers 3 folgt der größere seitliche Abstand gegenüber Regler 2 bei der Beendigung des Ausweichassistenten.



(a) SCV während des Links-Ausweichens



(b) Start- und Endpunkte des Ausweichvorgangs

Abb. 59: Links-Ausweichen mit Beginn im Bereich *schräg vor dem Hindernis*

Der Test für das Rechts-Ausweichen ist vergleichbar mit dem ersten Simulations-Testszenario (vgl. Abb. 54), d. h. das SCV befindet sich zu Manöverstart in dem Bereich *vor dem Hindernis*. Das Ergebnis des Ausweichvorgangs in der realen Umgebung ist für dieses Szenario mit dem bestimmten Achswinkel der Laserscannermessung und mit dem berechneten Achswinkel aus der Odometrie dokumentiert.

Im Testlauf mit Regler 2 fällt der Achswinkel auf maximal -45° , bevor er durch das Gegenlenken wieder ansteigt, um sich parallel zum Hindernis auszurichten (vgl. Abb. 60). Folgend sind die Zeitpunkte der Bereichsumschaltungen aufgelistet:

- 0,45 s: *vor dem Hindernis* \Rightarrow *schräg vor dem Hindernis*
- 2,4 s: *schräg vor dem Hindernis* \Rightarrow *neben dem Hindernis*

Der Vergleich zwischen den ermittelten Achswinkeln aus der Laserscannermessung und der Odometrie ergibt:

- Laserscannermessung
 - Zu Beginn in dem Gebiet *vor dem Hindernis* variiert der Achswinkel stark, da der Achswinkel des SCV laut Berechnung gleich dem Messwinkel des Objektpunktes K ist.
 - Bei der zweiten Bereichsumschaltung sind zwei Messfehler außerhalb des Toleranzbereiches aufgetreten. Zu diesem Zeitpunkt ist das geometrische Modell im Grenzbereich von *schräg vor dem Hindernis*, dieses wirkt sich auf die Berechnung aus.
 - Die Steigung zwischen den Messwerten ab 6,3 s ist größer als die zuvor, da das Ende des Hindernisobjekts erreicht wird (vgl. 6.5).
- Odometrie
 - Das Gegenlenken des Ausweichassistenten bei 3 s wurde von der Odometrie nicht im vollen Ausmaß erfasst, folglich laufen die beiden gemessenen Achswinkel ab diesem Zeitpunkt auseinander.
- Insgesamt
 - Die Steigung des Achswinkels θ sinkt ab 3,9 s, d. h. der Soll-Lenkwickelausschlag verringert sich (ausgenommen die Steigung der Laserscannermessung bei Erreichen des Endes vom Hindernis).

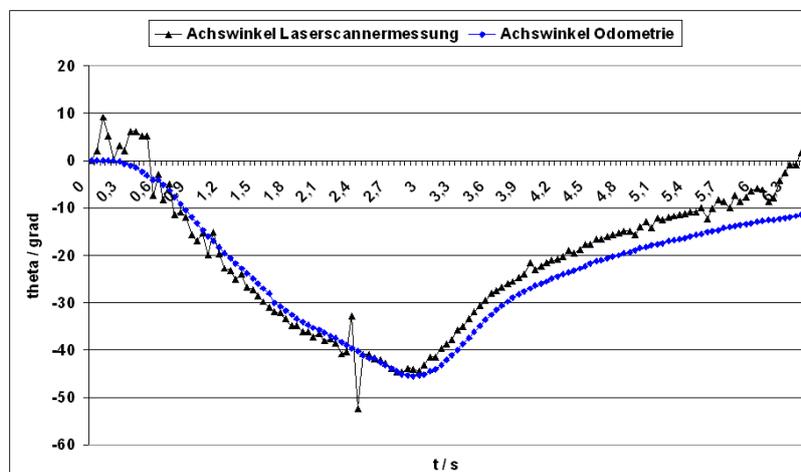


Abb. 60: SCV während des Links-Ausweichens Regler 2

Mit dem Regler 3 fällt der Achswinkel maximal auf -41° und steigt nach dem Gegenlenken nicht so stark wie bei Regler 2. Dieses Beides bedeutet, dass die Lenkbewegungen mit dem Regler 3 weicher sind. Der Vergleich zwischen den beiden Messmethoden in Abb. 61 bestätigt die Erkenntnisse aus Abb. 60. Die Zeitpunkte für die Bereichsumschaltung sind:

- 0,5 s: *vor dem Hindernis* \Rightarrow *schräg vor dem Hindernis*
- 2,25 s: *schräg vor dem Hindernis* \Rightarrow *neben dem Hindernis*

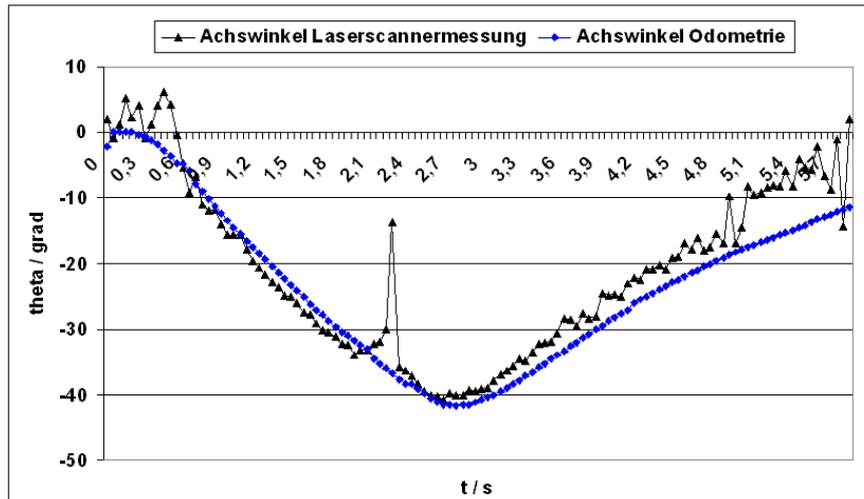


Abb. 61: SCV während des Links-Ausweichens Regler 3

Die Regeldifferenz für den Bereich *neben dem Hindernis* ist in Abb. 62 präsentiert. Der vorgegebene Soll-Lenkswinkel vom Regler ist wegen der zu geringen Reglerdifferenz am Ende des Ausweichmanövers zu gering, um das Parallelausrichten des SCV in einer kurzen Strecke zu vollziehen. Deshalb blieb das SCV bei den Tests in der realen Umgebung mit einem Achswinkel von ungefähr -10° stehen. Dieses ließe sich mit einem der beiden folgenden Ansätze beheben:

- Unterschreitet der Achswinkel θ eine definierte Schwelle, so wird der Soll-Lenkswinkel abhängig vom Achswinkel θ geregelt. In dem Fall wird davon ausgegangen, dass der Seitenabstand zum Hindernis näherungsweise erreicht ist.
- Unterschreitet der Achswinkel θ eine definierte Schwelle, so wird der P-Anteil des Reglers erhöht. Dadurch steigt die Verstärkung des Reglers und ein größerer Soll-Lenkswinkelausschlag wird vorgegeben.

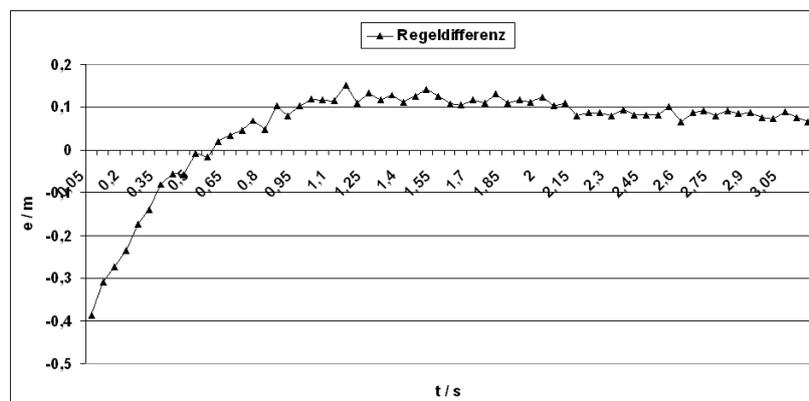


Abb. 62: Regeldifferenz

9.3 Realtime Verhalten der Aktionsausführung

Die Software des Ausweichassistenten läuft unter dem Echtzeitbetriebssystem QNX und ist in mehrere Threads aufgeteilt (vgl. 8). In diesem Abschnitt wird das Timing des Threads zur

Aktionsausführung betrachtet, da dieser zyklisch den Soll-Lenkwinkel für das Ausweichmanöver vorgibt. Das Ziel ist es, den Thread alle 50 ms zur Ausführung zu bringen, da dies der Scanfrequenz des Umgebungsmodells und der CAN-Nachricht für den Soll-Lenkwinkel entspricht. D.h., sowohl auf Sensor- als auch auf Aktoreseite wird mit einer Frequenz von 20 Hz gearbeitet.

Das zeitliche Verhalten des Threads *Aktionsausführung* zeichnet sich durch die Rechenzeit und die Sleep-Dauer von 48 ms aus (vgl. Abb. 63). Hinzu kommt die variable Wartezeit nach einem Sleep, bis dem Thread der Prozessor zugeteilt wird. Die Rechenzeit teilt sich im Wesentlichen in die vier folgenden Blöcke auf:

- Vom Umgebungsmodell die aktuellen Daten des Hindernisobjekts übernehmen.
- Die aktuellen Abstände zum Hindernis mit dem geometrischen Modell bestimmen.
- Den Soll-Lenkwinkel mit dem digitalen PD-Regler berechnen.
- Eine Nachricht mit dem Soll-Lenkwinkel an den Steuerungsautomaten versenden.

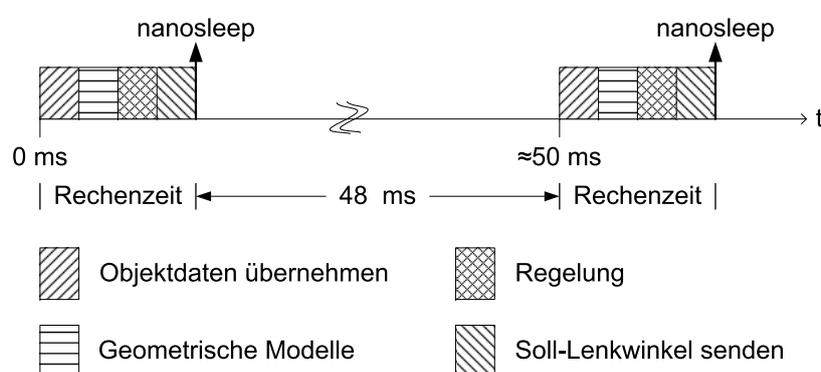


Abb. 63: Schematisches Timing des Threads Aktionsausführung

Für die Timing-Messungen wurde ein digitaler Ausgang des Parallelports genutzt, der während der Rechenzeit auf *high* gesetzt wird [5]. Mit einem Oszilloskop wurde eine Rechenzeit von $5,13 \mu\text{s}$ gemessen (Abb. 64). Die Zeit zwischen zwei Ausführungszyklen beträgt nach der Messung $49,48 \text{ ms}$ (Abb. 65). Aus dieser Messung ergibt sich, dass der Thread $20 \mu\text{s}$ zu früh mit der Bearbeitung seiner Aufgabe beginnt, um die Frequenz von 20 Hz einzuhalten. Allerdings ist dabei zu beachten, dass u.a. die Wartezeit auf die Prozessorressource variiert und folglich sich auch das Timing des Threads verändert.

Als Vergleich für die Einhaltung der Frequenz von 20 Hz wird der Laserscanner betrachtet. Laut [15] beträgt die Scanfrequenz des Laserscanners $20 \text{ Hz} \pm 5 \%$. Demnach liegt die zeitliche Varianz bei $\pm 2,5 \text{ ms}$. Daraus folgt, ist im Grenzfall die Wartezeit des Threads *Aktionsausführung* nach dem Sleep null, so ist die Frequenz weiterhin im Toleranzbereichs des Laserscanners.

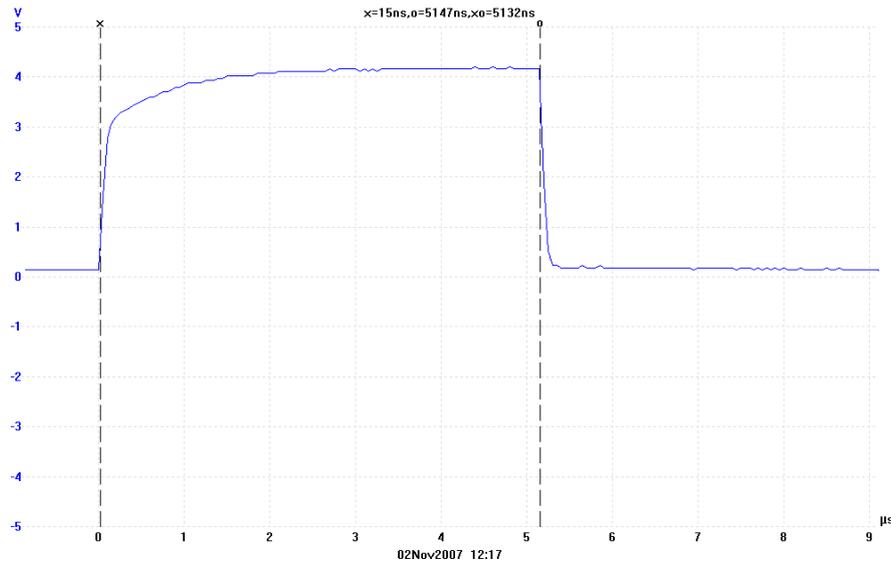


Abb. 64: Timingmessung des Threads Aktionsausführung: Ausführungsdauer

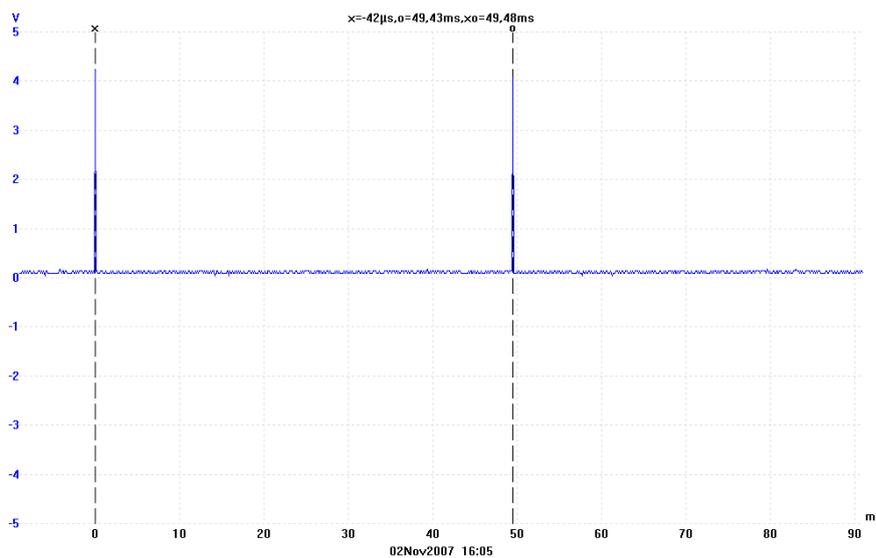


Abb. 65: Timingmessung des Threads Aktionsausführung: Frequenz

9.4 Vergleich zu dem Ausweichassistenten mit dem Deichsel-Ansatz

In der Masterarbeit [5] wurde ein Ausweichassistent nach dem Ansatz einer *virtuellen Deichsel* realisiert. Wie bereits in den Abschnitten 4.2 und 6.1 offen gelegt, hat das History-MA-Filter folgende Auswirkungen auf das System des Ausweichassistenten:

- Es treten Objektanomalien auf.
- Das Filter erzeugt eine zeitliche Verzögerung.
- Die Identifikationsnummer des Objekts ändert sich während des Ausweichens.

Die Objektanomalien treten in unregelmäßigen Abständen in der Vorverarbeitung auf. Sie sind abhängig von der Relativgeschwindigkeit, der Größe des Hindernisobjekts und dem Winkel mit dem sich das SCV auf das Hindernis zubewegt. Es wird angenommen, dass sich das Resultat

des *Deichsel*-Ausweichassistenten in der realen Umgebung zwischen verschiedenen Testläufen wegen der Vorverarbeitung nicht reproduzieren lässt. Außerdem reicht nach [5] die Fläche des Laborraums nicht für ein reales Ausweichmanöver des *Deichsel*-Ausweichassistenten aus, da die benötigten Ausweichbreiten nicht zur Verfügung stehen. Deshalb wird in dieser Masterarbeit auf einen messtechnischen Vergleich der beiden Ausweichassistenten verzichtet.

Die konzeptionellen Vor- und Nachteile des auf einer Abstandsregelung basierenden Ausweichassistenten sind:

- Vorteile
 - Die Laserscanner Abstandmessung M_{ay} geht als Rückführgröße in den Regelkreis ein.
 - Die Ausweichcharakteristik lässt sich durch die Reglerparameter verändern.
- Nachteil
 - Die relative Abstandsbestimmung bezieht sich auf die halbe Hindernisbreite. Diese wird im letzten Teil des Bereichs *schräg vor dem Hindernis* nicht mehr korrekt detektiert.

Zur Behebung des Nachteils sind zwei Lösungsansätze denkbar:

- Das geometrische Modell zur relativen Positionsbestimmung wird nach Kapitel 10 angepasst
- Der Scanbereich des Laserscanners wird erhöht.

10 Alternativer Ansatz zur relativen Ortsbestimmung und weitere Entwicklungsschritte

Das geometrische Modell für die relative Ortsbestimmung des SCV lässt sich mit einem alternativen Ansatz vereinfachen. Die Ressourcen auf dem Koordinierungsrechner, einem Pentium III mit 650 Mhz, sind für die Rechenschritte aus 6 ausreichend. Bei Prozessoren, die in Embedded Systemen eingesetzt werden, ist die Rechenzeit aber sehr wertvoll und es stehen nicht immer Einheiten für Floating-Point Berechnungen und trigonometrischen Funktionen zur Verfügung. Die Vorteile des Ansatzes in diesem Kapitel sind:

- Weniger Rechenschritte
- Keine trigonometrischen Funktionen
- Keine Multiplikation oder Division

Für die Berechnungen der relativen Positionen werden direkt die kartesischen Koordinaten der drei markanten Punkte L , K und R von der Objekterkennung genutzt. Im Folgenden wird das Links-Ausweichen betrachtet. Der Abstand zum Hindernis in x-Richtung S_{ax} entspricht bei dem Start des Ausweichvorgangs dem Punkt $K.x$ (vgl. Gl. 151). Die relative Positionsbestimmung in y-Richtung bezieht sich bei diesem Ansatz auf die vordere linke Ecke des Hindernisses, deshalb gilt Gl. 152. Der Sprung für die Führungsgröße des Regelkreises setzt sich aus dem y-Abstand zu Ausweichbeginn S_{ay_init} und dem Sicherheitsseitenabstand SSA zusammen (vgl. Gl. 153 und Abb. 66).

$$S_{ax} = K.x \quad (151) \quad S_{ay} = L.y \quad (152) \quad Y_{s1} = S_{ay_init} + SSA \quad (153)$$

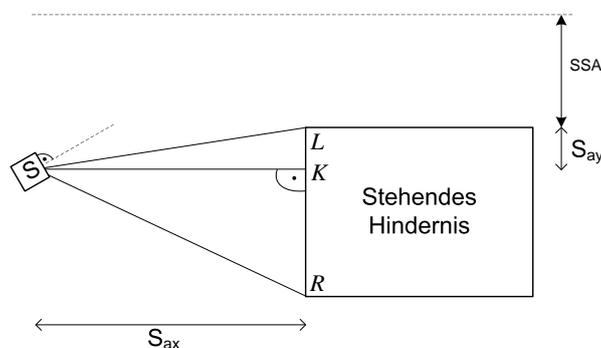


Abb. 66: Berechnung des Sprungs für die Führungsgröße

Während des Ausweichmanövers findet keine Bereichsumschaltung mehr statt, unter der Voraussetzung, dass es sich um ein rechteckiges Hindernisobjekt handelt und sich das SCV auf einer geraden Bahn auf die Front des Hindernisses zubewegt (Abb. 67). Für die relative Positionsbestimmung sind in diesem Fall durchgängig die Gln. 151 und 152 anzuwenden, wobei *neben dem Hindernis*, d.h. $R = K$ folgendes gilt: $S_{ax} = 0 \text{ m}$. In den Regelkreis geht der berechnete Abstand S_{ay_reg} als Messung ein (vgl. Gl. 154). Im ursprünglichen Bereich *vor dem Hindernis* ist S_{ay} bei diesem Ansatz negativ und in dem Gebiet *schräg vor dem Hindernis* positiv.

$$S_{ay_reg} = Y_{s1} - (S_{ay_init} - S_{ay}) \quad (154)$$

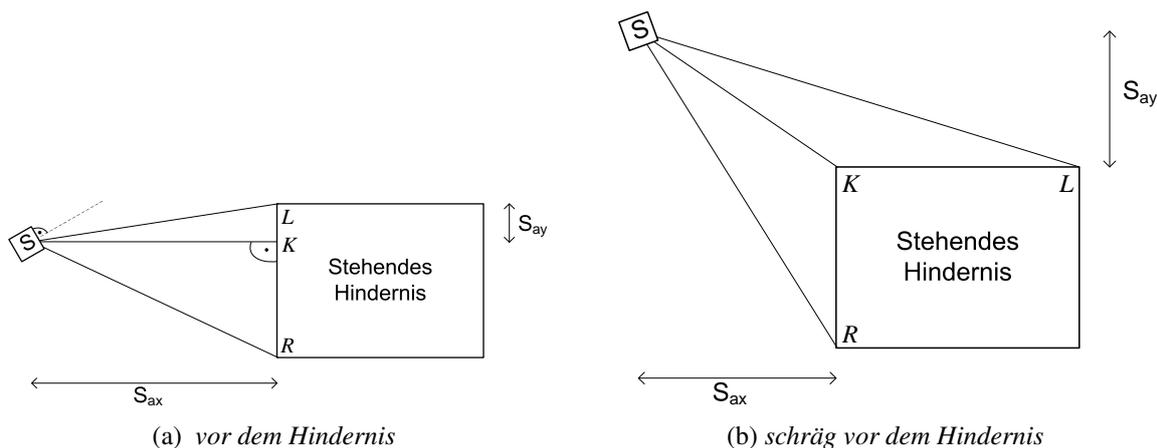


Abb. 67: Relative Positionsbestimmung

Das Kalman Filter eignet sich innerhalb eines Scans nicht zur Unterdrückung des Messrauschens (vgl. Kapitel 6). Allerdings ließe sich eine *optimale Schätzung* für die Abstandswerte und den Achswinkel zwischen kompletten Scans bestimmen. Des Weiteren kann das Kalman Filter zur Sensordatenfusion zwischen Odometrie und Laserscannermessung genutzt werden (vgl. Anhang L).

Für das räumliches Mittelwertfilter innerhalb eines Scans ist das Medianfilter aus [8] als Alternative zu testen. Dieses verschiebt ebenfalls ein Fenster über den kompletten Scan, ersetzt dabei allerdings den aktuellen Achswinkel durch den mittleren Messwert der räumlich benachbarten Messwerte. Dadurch entstehen keine allein stehenden Messpunkte zwischen den Objektübergängen.

Das RM-Filter innerhalb eines Scans ließe sich intern auf dem DSP des Laserscanners SICK LD OEM1000 implementieren [15]. Dieses spart Rechenzeit des Prozessors, die in einem Embedded-Prozessor im Kfz-Bereich begrenzt ist.

Der Ausweichassistent kann im nächsten Entwicklungsschritt auf FlexRay-Knoten portiert werden. Damit wäre ein Einsatz auf Hardwarekomponenten des Kfz-Bereichs mit einem zeitgesteuerten Hard-Realtime System gewährleistet [9].

Zur Verfeinerung des Umgebungsmodells kann in Zukunft eine lokale Lokalisierung genutzt werden. Dazu wird der Laserscanner mit einem Scanbereich von 360° betrieben und aus aufeinanderfolgenden Scans die Messpunkte zur Überdeckung gebracht. Das *Verfahren von Cox* ist z. B. ein Algorithmus dafür. Bei dem Verfahren werden die extrahierten Linien aus dem aktuellen Scan einem apriori Linienmodell zugeordnet. Mit der Anfangsschätzung der Aufnahmeposition, die aus der Odometrie ermittelt wird, lässt sich die Verschiebung und Drehung des Scans bestimmen. Mit den Informationen aus der lokalen Lokalisation können weitere Aspekte bei der Aktionsentscheidung, als auch bei der Aktionsausführung berücksichtigt werden. Zum Beispiel, wenn sich neben dem Hindernis ein weiteres Objekt befindet oder sich während des Manövers in den Ausweichbereich bewegt. Für weitere Algorithmen und Details wird auf [8] verwiesen.

11 Zusammenfassung

Das Ziel dieser Masterarbeit war, einen automatischen Ausweichassistenten auf einer Laserscanner-basierten Abstandsregelung für das SCV des FAUST-Projekts zu entwickeln. Der Laserscanner SICK LD OEM1000 liefert die Abstandswerte über den CAN-Bus mit einer Frequenz von 20 Hz. Das räumliche Mittelwertfilter unterdrückt das Messrauschen des Laserscanners in der Vorverarbeitung. Die Objekterkennung erstellt aus den Messwerten ein Umgebungsmodell. Die Situationsanalyse und Aktionsentscheidung überprüft zyklisch, ob eine Kollision bevorsteht und löst im positiven Fall einen Ausweichvorgang für die Richtung mit der kürzeren Ausweichbreite aus. Die Aktionsausführung des Ausweichmanövers läuft in einem dafür gestarteten Thread, der die aktuellen Hindernisdaten von der Objekterkennung abrufen, das geometrische Modell für die relative Positionsbestimmung berechnet und den Soll-Lenkwinkel mit einem PD-Regler einstellt. Der Ausweichassistent beendet den Vorgang selbstständig, sobald das SCV parallel zum Hindernis steht. Zu Entwicklungs- und Testzwecken können die gefilterten Messwerte des Laserscanners und die extrahierten Objekte des Umgebungsmodells in einer QNX-Photon Testumgebung visualisiert werden.

Der Ausweichassistent besteht aus einer Drei-Ebenen-Struktur mit *Situationserkennung*, *Situationsanalyse / Aktionsentscheidung* und *Aktionsausführung*. Die Software läuft auf dem Koordinierungsrechner des SCV, einem PentiumIII 650 Mhz mit dem Echtzeitbetriebssystem QNX. Der Ausweichassistent und der Steuerautomat für die Fahr- und Lenkmotoren betreiben kontinuierlich neun Threads auf dem Koordinierungsrechner. Für den Zeitraum eines Ausweichvorgangs wird ein weiterer Thread erstellt. Die Threads nutzen das Round-Robin Scheduler-Verfahren. Das Timing mit einer Frequenz von 20 Hz für die Aktionsausführung wird durch ein 48 ms Sleep des Threads plus Rechenzeit und Wartezeit für den Prozessor erreicht. Die Messung für die Zeit zwischen zwei Ausführungszyklen ergab 49,48 ms. Für die Entscheidung der Ausweichrichtung des Assistenten wurde ein hierarchischer Zustandsautomat nach dem AIRA-Konzept realisiert.

Das räumliche Mittelwertfilter mit der Filterbreite $N = 5$ bildet innerhalb eines Scans aus den räumlich benachbarten Messwerten einen Mittelwert. Dazu wird ein Fenster mit der Filterbreite N über den kompletten Scan verschoben und für jeden Scanwinkel der Mittelwert bestimmt.

Mit dem hergeleiteten kinematischen Einspurfahrzeugmodell des SCV lässt sich der Achswinkel, sowie die Vorder- und Hinterradposition abhängig vom Lenkwinkel und der Vorderradgeschwindigkeit berechnen. Die Systemgleichungen stehen für die kontinuierliche und die diskrete Betrachtung zur Verfügung und wurden in ausgewählten Arbeitspunkten linearisiert.

Das entwickelte geometrische Modell dient zur relativen Positionsbestimmung des SCV. Der Abstand in x- und y-Richtung zum Hindernis und der Achswinkel werden auf Basis der Objektpunkte *Links*, *Kurz* und *Rechts* des Hindernisobjekts bestimmt.

Der entworfene Abstandsregelkreis dient zur Steuerung des SCV während des Ausweichvorgangs. Die Parameter des PD-Reglers wurden im ersten Schritt mit dem *symmetrischen Optimum* bestimmt und ein Ausweichvorgang simuliert. Mit dem Regler erreicht der Ist-Lenkwinkel α_{ist} beim Links-Ausweichen einen Maximalauschlag von $-60,8^\circ$, das Überschwingen des Vorderrads Δy beträgt 14 cm und die Ausregelzeit t_{aus_M} des Vorderrades ist 5,9 s in der Simulation. Der Frequenzgang des Reglers wurde in einem Bode-Diagramm visua-

lisiert und nach dem *Frequenzkennlinienverfahren* im zweiten Schritt analysiert. Die Erhöhung der Phasenreserve ergibt eine größere Dämpfung des Einschwingverhaltens. Dadurch folgt: $\alpha_{ist} = -26,5^\circ$, $\Delta y = 3,8 \text{ cm}$ und $t_{aus_M} = 8,7 \text{ s}$. Als Alternative wurde die Durchtrittsfrequenz im dritten Schritt verkleinert, um eine weichere Lenkbewegung durch eine höhere Ausregelzeit zu erreichen. Der maximale Lenkwinkelausschlag sinkt von $-26,5^\circ$ auf $-18,2^\circ$.

Die Funktionsweise des Ausweichassistenten wurde durch verschiedene Szenarien in der Testumgebung des SCV und in der realen Umgebung nachgewiesen. In der Testumgebung wird die Laserscannermessung und die Fahrzeugeigenbewegung simuliert. Die Szenarien beinhalten beide Ausweichrichtungen, Hindernisbreiten von 1 m bis 5 m und die Startpositionen *vor dem Hindernis* und *schräg vor dem Hindernis*.

Literatur

- [1] COMMISSION, European: *eSafety - moderne Informations- und Kommunikationstechnologien bringen mehr Sicherheit im Straßenverkehr*, European Commission, eSafety Merkblatt, 2007
- [2] LEXUS: *Lexus*. 2007. – www.lexus.de
- [3] PREVENT: *The PReVENT Integrated Project: Approach and Organisation*, Statistisches Bundesamt, PReVENT Presentation v20, 2007
- [4] DESTATIS: *Unfallgeschehen im Straßenverkehr*, Statistisches Bundesamt, Unfallstatistiken 2006, 2007
- [5] PRÖHL, Andreas: *Automatischer Ausweichassistent auf Basis einer Laserscanner-Abstandserfassung für ein fahrerloses Transportsystem*, Hochschule für Angewandte Wissenschaften Hamburg, Masterarbeit, 2006
- [6] CORDES, Stefan: *Automatischer Bremsassistent auf Basis einer Laserscanner-Abstandserfassung für ein fahrerloses Transportsystem*, Hochschule für Angewandte Wissenschaften Hamburg, Masterarbeit, 2006
- [7] YEFREMOV, John Alberts; Ingmar Gründel; Ning Liu; Denis Schetler; Y.: *Einparkassistent*, Hochschule für Angewandte Wissenschaften Hamburg, Projektarbeit, 2007
- [8] GUTMANN, Jens-Steffen: *Robuste Navigation autonomer mobiler Systeme*. Akademische Verlagsgesellschaft Aka, 2000. – ISBN 3–89838–241–9
- [9] SELLENTIN, Jörn: *Ein zeitgesteuertes, verteiltes SW-Konzept implementiert auf FlexRay-Komponenten für ein fahrerloses Transportsystem*, Hochschule für Angewandte Wissenschaften Hamburg, Masterarbeit, 2006
- [10] MIROW, Bernd: *Physik Formeln*. Dümmlers Verlag, 1996. – ISBN 3–427–41770–0
- [11] BARTSCH, Hans-Jochen: *Taschenbuch Mathematischer Formeln*. Fachbuchverlag Leipzig, 1999. – ISBN 3–446–21048–2
- [12] FÖLLINGER, Otto: *Regelungstechnik*. Hüthig, 1994. – ISBN 3–7785–2336–8
- [13] FÖLLINGER, Otto: *Laplace-, Fourier- und z-Transformation*. Hüthig, 2003. – ISBN 3–7785–2911–0
- [14] MEISEL, Andreas: *Hilfsblätter 'Modellierung zeitkontinuierlicher und hybrider Systeme'*, Hochschule für Angewandte Wissenschaften Hamburg, Hilfsblätter für Modellierungsvorlesung, 2006
- [15] SICK: *Lasermesssystem LD-OEM*, Sick, Betriebsanleitung, 2004
- [16] SCHETLER, Denis: *Kalman Filter zur Rekonstruktion von Messsignalen*, Hochschule für Angewandte Wissenschaften Hamburg, Seminararbeit, 2006
- [17] MEISEL, Andreas: *MO 6.3 Regelung dyn. Systeme - Regelkreise*, Hochschule für Angewandte Wissenschaften Hamburg, Vorlesungsskript, 2006
- [18] RÖRIG, Christof: *Frequenzkennlinienverfahren*, FH Dortmund, Vorlesungsskript, 2004
- [19] UNBEHAUEN, Heinz: *Regelungstechnik 2*. Vieweg, 2000. – ISBN 3–528–73348–9

-
- [20] KALTENHÄUSER, Heiner: *Implementierung reaktiver Software*, Hochschule für Angewandte Wissenschaften Hamburg, Vorlesungsskript, 2006
- [21] FÄHNDEERS, Erhard: *Betriebssysteme - Prozess Scheduling*, Hochschule für Angewandte Wissenschaften Hamburg, Vorlesungsskript, 2002
- [22] FÜR MASCHINENBAU FACHGEBIET MECHATRONIK, Technische Universität Ilmenau F.: *Komplexpraktikum Mechatronik Autonomes mechatronisches Fahrsystem*, Technische Universität Ilmenau, Komplexpraktikum Mechatronik, 2006
- [23] COX, I. ; WILFONG, G.: *Autonomous Robot Vehicles*. Springer-Verlag, 1990. – ISBN 3-540-97240-4

Abbildungsverzeichnis

1	Personenschäden im Straßenverkehr und Unfallursachen [4]	1
2	Abmessungen des Sensor Controlled Vehicle (Vorlage aus [6])	5
3	Aufbau des Sensor Controlled Vehicle	6
4	Scannen vor dem Hindernis, vor dem Ausweichen	7
5	Unterteilung der Positionsbereiche des SCV während des Ausweichens	8
6	Abstände des lenkbaren Vorderrads M für die Regelung	8
7	Abstände des lenkbaren Vorderrads M für die Regelung <i>neben dem Hindernis</i>	9
8	HW-Aufbau des Sensor Controlled Vehicle	10
9	Threadübersicht des Koordinierungsrechners	11
10	Softwarestruktur des Ausweichassistenten	12
11	Objektpunkte bezogen auf S	13
12	Objektpunkte bezogen auf P	13
13	Schematisches Objekt des History-MA-Filters	15
14	Visualisierung des Umgebungsmodells (aus [6])	17
15	Skizze: Kinematisches Einspurfahrzeugmodell des SCV	18
16	Sprungantwort der Lenkverzögerung des SCV	21
17	Kinematisches Modell des SCV über die Zeit	22
18	Kinematisches Modell des SCV, y über x	22
19	Trapezintegration für den Achswinkel	23
20	Simulation Räder über die Zeit	25
21	Simulation y-Positionen der Räder über jeweilige x-Positionen	25
22	Vergleich zeitkontinuierlicher / zeitdiskreter Systemgleichungen	26
23	Das RM-Filter zu Beginn einer Scanrunde	30
24	Umgebungsplan des SCV mit Hindernisobjekt	30
25	Objektübergänge bei der Hinderniserkennung, rot Strecke L bis K und grün Strecke K bis R eines Objekts	31
26	Erster Bereichsübergang während des Ausweichens	33
27	Schräg vor dem Hindernis	33
28	Neben dem Hindernis	33
29	Dreiecksberechnungen zur Bestimmung von S_{ax} und S_{ay}	34
30	Bestimmung von θ bei SCV vor dem Hindernis	35
31	Dreiecksberechnungen zur Bestimmung von S_{ax} und S_{ay}	36
32	Winkelberechnung zur Bestimmung von θ	36
33	Geometrisches Modell zur Bestimmung von S_{ay} und θ	37
34	Transformation der Abstandswerte von der Sensorposition zu der Vorderradpo- sition (bei θ gleich 0° bis 90°)	38
35	Schema Regelkreis	39
36	Berechnung des Sprungs für die Führungsgröße	39
37	Blockschema des PD-Reglers	40
38	Blockschema der linearisierten Übertragungsfunktion des SCV	41
39	Vorder- und Hinterradposition des SCV zu Regler 1	47
40	Lenkwinkel des SCV zu Regler 1	48
41	Achswinkel des SCV zu Regler 1	48
42	Offener Regelkreis nach dem <i>symmetrischen Optimum</i>	50
43	Regler nach dem <i>symmetrischen Optimum</i>	51
44	Regler mit veränderten Zeitkonstanten nach der Analyse der Frequenzkennlinien	52
45	Offener Regelkreis mit modifiziertem Regler	52
46	Regler 2: Vorder- und Hinterrad während des Links-Ausweichens	53

47	Regler 2	54
48	Regler 3	54
49	Blockschaltbild digitaler PD Regler	57
50	Softwarerealisierung des RM-Filters	59
51	Entscheidungsautomat für die Ausweichrichtung	61
52	Verarbeitungskette des Ausweichassistenten während des Ausweichvorgangs	62
53	Visualisierung der erweiterten Simulationsumgebung	67
54	TestszENARIO 1: Rechts-Ausweichen	68
55	Simulation Rechts-Ausweichens mit Start <i>vor dem Hindernis</i>	69
56	Simulation des Rechts-Ausweichens mit einem breiten Hindernisobjekt	69
57	TestszENARIO 3: Links-Ausweichen	70
58	Simulation des Ausweichvorgangs	70
59	Links-Ausweichen mit Beginn im Bereich <i>schräg vor dem Hindernis</i>	71
60	SCV während des Links-Ausweichens Regler 2	72
61	SCV während des Links-Ausweichens Regler 3	73
62	Regeldifferenz	73
63	Schematisches Timing des Threads Aktionsausführung	74
64	Timingmessung des Threads Aktionsausführung: Ausführungsdauer	75
65	Timingmessung des Threads Aktionsausführung: Frequenz	75
66	Berechnung des Sprungs für die Führungsgröße	77
67	Relative Positionsbestimmung	78
68	Kinematisches Modell in Simulink	89
69	Simulink Simulationsmodell für den kontinuierlichen Regelkreis	90
70	Simulink Simulationsmodell für den diskreten Regelkreis	91
71	MA-Filter, ein Scan in Polarkoordinaten, Winkel von 0° bis 109°	101
72	MA-Filter, ein Scan in Polarkoordinaten, Winkel von 0° bis 56°	101
73	Transformation der Abstandswerte bei einem Achswinkel von 0° bis -90° (Rechts-Ausweichen)	103
74	Bode-Diagramm des Reglers 3	104
75	Bode-Diagramm des offenen Regelkreises zu dem Regler 3	104
76	Kontinuierlicher und diskreter Regelkreis zum Reglervergleich	105
77	Kalman-Filter mit zwei Sensoren	122

Tabellenverzeichnis

1	Gültigkeitsbereiche der Systemvariablen des SCV	18
2	Interpretation der Systemparameter des SCV	19
3	Messungen der Zeiten für die Bestimmung der Lenkverzögerung	21
4	Exemplarische Arbeitspunkte	28
5	Die gegenseitige Beeinflussung der Objektgrenzen des räumlichen Mittelwert- filters, $N = 5$	31
6	Formelzeichen und Einheiten zur Übertragungsfunktion des SCV	42
7	Vergleich Faktor b_1 und b_2	46
8	Vergleich von verschiedenen Vorgaben für AP und T_v	47
9	Die Abkürzungen der Kennwerte	55
10	Kennwerte der Ausweichsimulation mit zwei Reglerentwurfsverfahren	55
11	Die Thread-Prioritäten des Softwaresystems	63
12	Blockierende Funktionen der Threads	64
13	Parameter für ideale Ausweichkurve	66
14	History-MA-Filter: aktueller Scan $n = 6$	87
15	History-MA-Filter: aktueller Scan $n = 7$	88
16	Räumliches Mittelwertfilter $N = 3$	102
17	Räumliches Mittelwertfilter $N = 5$	102
18	Räumliches Mittelwertfilter $N = 7$	102

A Inbetriebnahme LD-OEM1000

Der LD-OEM1000 ist das Nachfolgermodell von dem LD-OEM, deshalb sind die beiden Laserscanner vom Typ gleich, der Laserscanner wurde aber in einigen Punkten, wie z. B. den Hardwareschnittstellen und der Softwareansteuerung weiter entwickelt.

Der Laserscanner wird statt einem 9-poligen D-Sub-HD-Stecker mit einem 15-poligen D-Sub-HD-Stecker angeschlossen. Das neue Kabel mit dem 15-poligen Stecker für den Laserscanner verfügt auf der anderen Seite über einen 9-poligen Stecker für den CAN-Bus und einen 9-poligen Stecker für die serielle Schnittstelle RS-232.

Das Konfigurationstool *LD-Config* für den Laserscanner Sick LD-OEM wurde für den Sick LD-OEM1000 durch das Programm *SOPAS* ersetzt. Spezielle Einstellung für die Verbindung des Laserscanners mit dem Tool SOPAS von der Firma Sick über die Schnittstelle RS-232 sind:

- CoLa-Dialekt → binär
- Baudrate → 115 kBd
- Port-Einstellungen → 8 Datenbits, 1 Stoppbit, keine Parität

Die Einstellungen zu CoLa-Dialekt und Baudrate sind keine default-Einstellungen. Die Standardeinstellung für den CoLa Dialekt ist ASCII, bei der die 0 durch *0x30* dargestellt wird. Zur Konfiguration des Laserscanners muss das Tool SOPAS an dem Laserscanner mit den folgenden Einstellungen angemeldet werden:

- Benutzername „AUTORISIERTER KUNDE“
- Kennwort „client“

Die Reset-Wartezeit des Laserscanners wurde in der Funktion *send_DO_RESET* im Modul *LD-Comm.c* auf 4 s erhöht, da bei einem CPU-Reset die Antwortnachricht vor dem Reset gesendet wird und dieser Reset bei dem Sick LD-OEM1000 länger als bei dem Vorgängermodell LD-OEM dauert. Die Wartezeit von 4 s wurde durch eine Testreihe ermittelt, eine genauere Analyse wurde nicht durchgeführt, da ein Reset des Laserscanners nur während der Startphase des Systems durchgeführt wird.

B Mittelungsergebnis des History-MA-Filter bei der Aktivierung des Bypasses

In den Tabellen 14 und 15 sind nur die Messpunkte des Hindernisobjekts angegeben.

n	84°	85°	86°	87°	88°	89°	90°	91°	92°	93°	94°	95°	96°	97°
1	944	943	941	940	940	939	939	939	940	940	941	943	944	946
2	939	937	936	935	934	934	934	934	934	935	936	937	939	941
3	934	932	931	930	929	929	929	929	929	930	931	932	934	936
4	929	927	926	925	924	924	924	924	924	925	926	927	929	931
5	924	922	921	920	919	919	919	919	919	920	921	922	924	925
6	918	917	916	915	914	914	913	914	914	915	916	917	918	920
7	965	963	962	961	960	960	960	960	960	961	962	963	965	967
8	960	958	957	956	955	955	954	955	955	956	957	958	960	962
9	954	953	952	951	950	949	949	949	950	951	952	953	954	956
10	949	948	946	945	945	944	944	944	945	945	946	948	949	951
11	941	940	938	937	937	936	936	936	937	937	938	940	941	943

Tabelle 14: History-MA-Filter: aktueller Scan $n = 6$

n	83°	84°	85°	86°	87°	88°	89°	90°	91°	92°	93°	94°	95°	96°	97°	98°
1	915	944	943	941	940	940	939	939	939	940	940	941	943	944	946	917
2	915	939	937	936	935	934	934	934	934	934	935	936	937	939	941	917
3	915	934	932	931	930	929	929	929	929	929	930	931	932	934	936	917
4	915	929	927	926	925	924	924	924	924	924	925	926	927	929	931	917
5	915	924	922	921	920	919	919	919	919	919	920	921	922	924	925	917
6	915	918	917	916	915	914	914	913	914	914	915	916	917	918	920	917
7	915	913	912	911	910	909	908	908	908	909	910	911	912	913	915	917
8	915	960	958	957	956	955	955	954	955	955	956	957	958	960	962	917
9	915	954	953	952	951	950	949	949	949	950	951	952	953	954	956	917
10	915	949	948	946	945	945	944	944	944	945	945	946	948	949	951	917
11	915	936	934	933	932	931	931	931	931	931	932	933	934	936	938	917

Tabelle 15: History-MA-Filter: aktueller Scan $n = 7$

C Simulink Simulationsmodelle

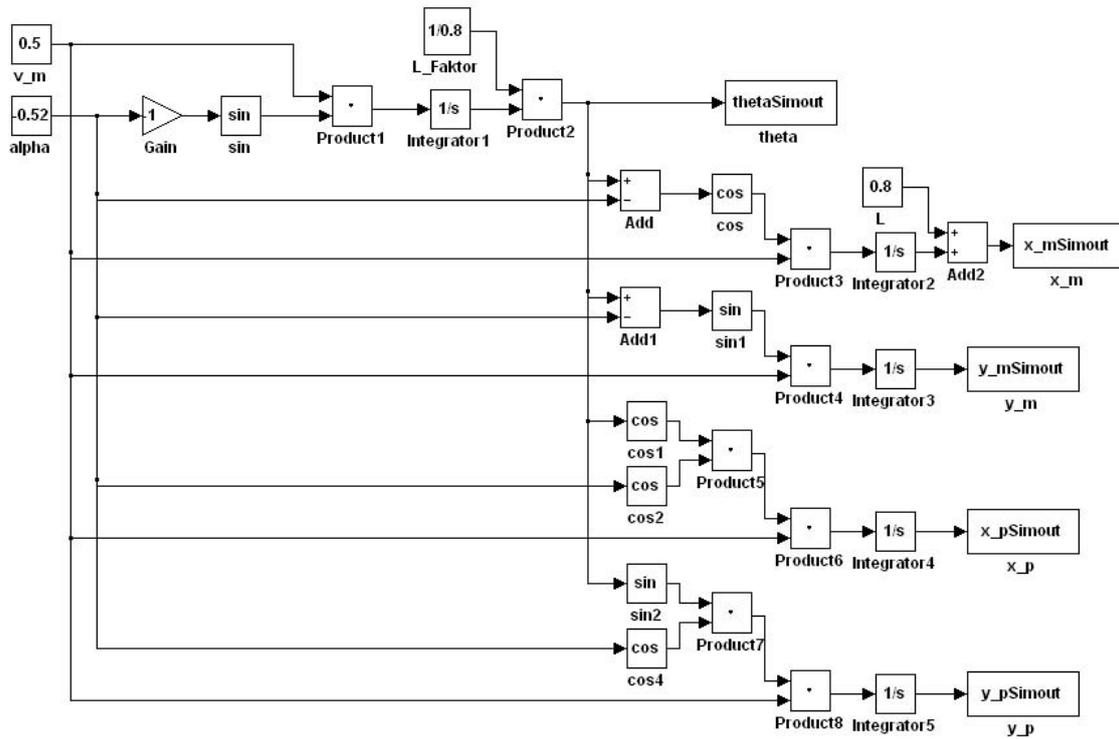


Abb. 68: Kinematisches Modell in Simulink

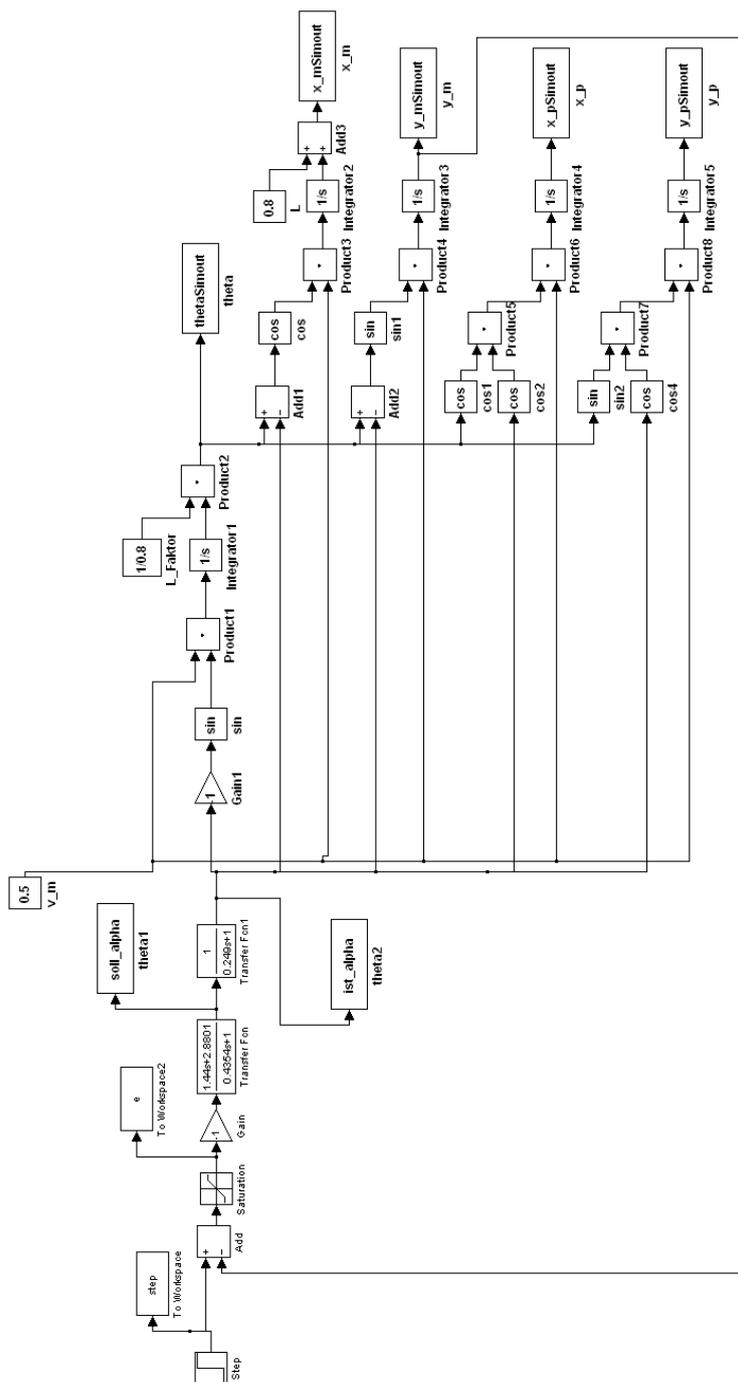


Abb. 69: Simulink Simulationsmodell für den kontinuierlichen Regelkreis

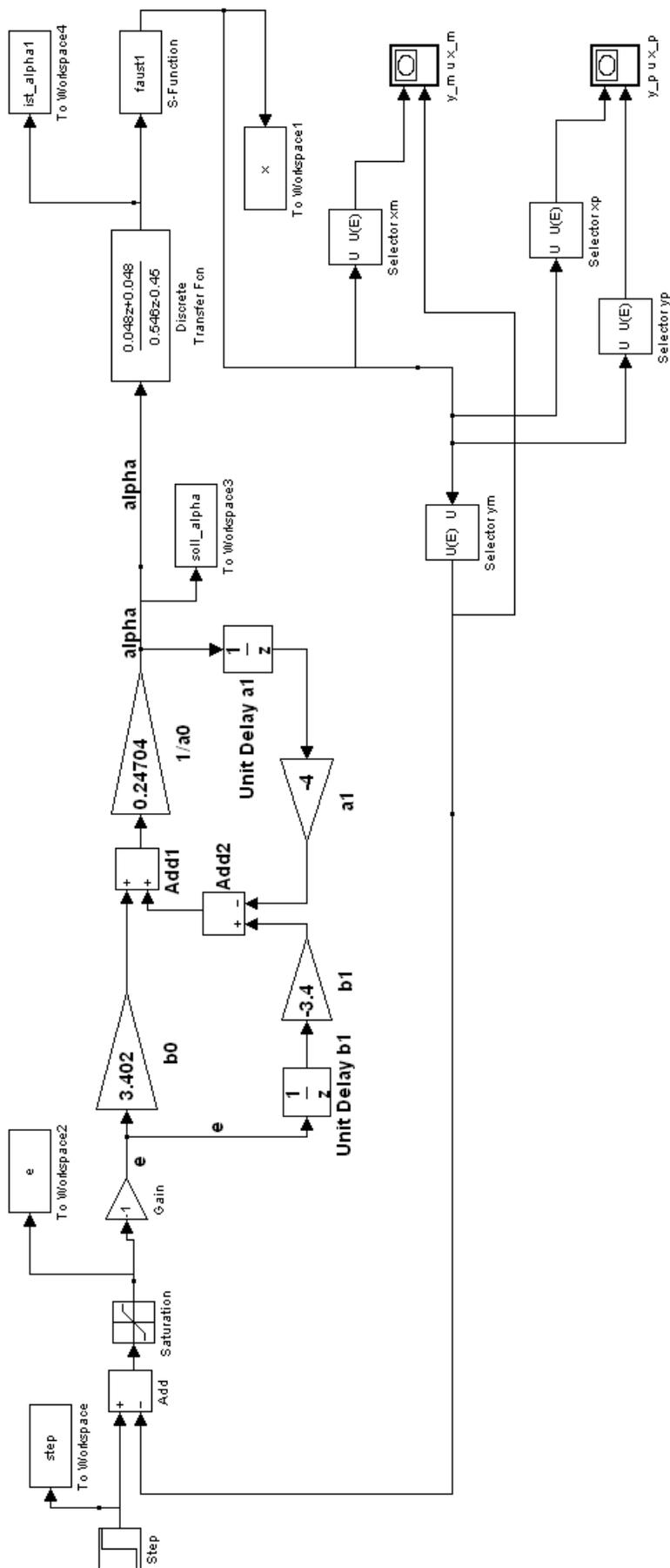


Abb. 70: Simulink Simulationsmodell für den diskreten Regelkreis

D Matlab Simulationscode

Listing 1: Matlab Simulationscode für die diskreten Systemgleichungen

```

L          = 0.8;           % in m
vm_vorgabe = 0.5;         % in m/s
alpha_vorgabe = -0.523599; % in rad (-30°)
h          = 0.05;        % in s
k          = 0;
n          = 500;         % Anzahl Berechnungen
pi       = 3.141593;

t          = [0];         % in s
alpha     = [0];         % in rad
theta     = [0];         % in rad
xp        = [0];         % in m
xm        = [0];         % in m
yp        = [0];         % in m
ym        = [0];         % in m
vm        = [0];         % in m/s
vp        = [0];         % in m/s

Lenkwinkel = [0];       % in Grad
Achswinkel = [0];       % in Grad

xm(1) = L;

alpha(1) = alpha_vorgabe;
vm(1)    = vm_vorgabe;
vp(1) = vm(1) * cos(alpha(1));
theta(1) = 0;

for k = 1:n
    t(k+1) = t(k) + h;

    alpha(k+1) = alpha_vorgabe;
    vm(k+1)    = vm_vorgabe;

    % Hinterradgeschwindigkeit
    vp(k+1) = vm(k) * cos(alpha(k));

    % Achswinkel
    theta(k+1) = theta(k) + (1/L) * ( 0.5*h*vm(k)*sin(-alpha(k)
        ) + 0.5*h*vm(k+1)*sin(-alpha(k+1)) );

    % Vorderrad
    xm(k+1) = xm(k) + 0.5*h*vm(k)*cos( theta(k)-alpha(k) ) +
        0.5*h*vm(k+1)*cos( theta(k+1)-alpha(k+1) );
    ym(k+1) = ym(k) + 0.5*h*vm(k)*sin( theta(k)-alpha(k) ) +
        0.5*h*vm(k+1)*sin( theta(k+1)-alpha(k+1) );

```

```

% Hinterrad
xp(k+1) = xp(k) + 0.5*h*vp(k)*cos(theta(k)) + 0.5*h*vp(k
+1)*cos(theta(k+1));
yp(k+1) = yp(k) + 0.5*h*vp(k)*sin(theta(k)) + 0.5*h*vp(k
+1)*sin(theta(k+1));

% Umrechnung der RAD Werte in Grad
Lenkwinkel(k) = alpha(k) * 180 / pi;
Achswinkel(k) = theta(k) * 180 / pi;
end

```

Listing 2: Linearisierung im AP und Symmetrisches Optimum

```

% Parameter
alpha      = 10                               % in Grad
alpha_rad  = (alpha / 180) * 3.14159;        % in Radian
theta      = 20;                               % in Grad
theta_rad  = (theta / 180) * 3.14159;        % in Radian
v_m        = 0.5;                               % m/s
L          = 0.8;                               % m

TL = 0.249                                     % s
Tv = 0.5                                       % s

% -----
% Linearisierung im Arbeitspunkt
% fuer Achswinkel und Vorderrad

% Linearisierungsfaktor thetaP
thetaP_fact_1 = cos(-alpha_rad);
% k theta alpha
kthetaalpha = (v_m/L) * thetaP_fact_1;

% Linearisierungsfaktoren xmP
xmP_fact_1 = -sin(theta_rad)*cos(alpha_rad)+cos(theta_rad)*sin(
alpha_rad);
% k x theta
kxtheta = v_m * xmP_fact_1;

xmP_fact_2 = cos(theta_rad)*(-sin(alpha_rad))+sin(theta_rad)*cos
(alpha_rad);
% k x alpha
kxalpha = v_m * xmP_fact_2;

% Linearisierungsfaktoren ymP
ymP_fact_1 = cos(theta_rad)*cos(alpha_rad)-(-sin(theta_rad))*sin
(alpha_rad);
% k y theta
kytheta = v_m * ymP_fact_1;

```

```

ymP_fact_2 =sin(theta_rad)*(-sin(alpha_rad))+cos(theta_rad)*cos
(alpha_rad);
% k y alpha
kyalpha = v_m * ymP_fact_2;

% -----
% Berechnung der Verhaeltnisse
% mit dem symmetrischen Optimum
Ts1 = kyalpha / (kthetaalpha * kytheta)
Ts2 = TL
ks1 = kthetaalpha * kytheta

Kp = 2 / ((Tv^2 + Ts1^2) * ks1)
Tpd = (0.0089746 * Kp * ks1 * (Tv + Ts1)^4) / Ts2

% -----
% Berechnung der Faktoren fuer die Uebertragungsfunktion
% des PD-Reglers
GrBs = Tv * Kp
GrB = Kp
GrAs = Tpd
GrA = 1

GrBs_KpVor = GrBs / GrB
GrB_KpVor = 1

Tvp1 = -(0.0089746 * Kp * ks1 * (Tv + Ts1)^4 / Ts2) + Tv
% -----
% Berechnung der Omega Koeffizienten
a = Tv * Ts1 + (1/(Kp*ks1))
b = (Tpd * Ts2) / (Kp * ks1)
c = Tv + Ts1
d = (Tpd + Ts2) / (Kp * ks1)

w2 = -2*a + c^2
w4 = a^2 + 2*b - 2*c*d
w6 = -2*(a*b) + d^2
% -----
% fuer Systemstabilität
stabi_a4 = 1
stabi_a3 = Tv + Ts1
stabi_a2 = (Tv * Ts1) + (1/(Kp*ks1))
stabi_a1 = (Tpd+Ts2) / (Kp*ks1)
stabi_a0 = (Tpd*Ts2) / (Kp*ks1)

% -----
% fuer Bode Diagramm Regler
num = [Tv 1]
den = [Tpd 1]

```

```

% Plot Regler
subplot(2,1,1);
bode(num, den);
title('Bode_Diagramm_Regler', 'FontSize',16, 'LineWidth',3)
% -----
% fuer Bode Diagramm offener Regelkreis
b0 = 1
b1 = (Tv + Ts1)
b2 = (Tv * Ts1)

a0 = 0
a1 = 0
a2 = 1
a3 = Tpd + Ts1
a4 = Tpd * Ts2

num = [b2 b1 b0]
den = [a4 a3 a2 a1 a0]

% Plot offener Regelkreis
subplot(2,1,2);
bode(num, den)
title('Bode_Diagramm_offener_Regelkreis', 'FontSize',16,'
      LineWidth',3)

```

Listing 3: Bode-Diagramme für Frequenzkennlinienverfahren

```

% -----
% Simulationsparameter fuer das
% Frequenzkennlinienverfahren
Tv = 85
Tpd = 2.5
Ts1 = 1.4287
Ts2 = 0.249

% -----
% fuer Bode Diagramm Regler
num = [Tv 1]
den = [Tpd 1]

% Plot Regler
subplot(2,1,1);
bode(num, den);
title('Bode_Diagramm_Regler', 'FontSize',16, 'LineWidth',3)
% -----
% fuer Bode Diagramm offener Regelkreis
b0 = 1
b1 = (Tv + Ts1)
b2 = (Tv * Ts1)

```

```

a0 = 0
a1 = 0
a2 = 1
a3 = Tpd + Ts1
a4 = Tpd * Ts2

num = [b2 b1 b0]
den = [a4 a3 a2 a1 a0]

% Plot offener Regelkreis
subplot(2,1,2);
bode(num, den)
title('Bode_Diagramm_offener_Regelkreis', 'FontSize',16,'
      LineWidth',3)

```

Listing 4: Matlab Simulationscode für das Fahrzeugmodell im digitalen Regelkreis

```

function [sys,x0,str,ts] = faust1(t,x,u,flag)
% Dispatch the flag. The switch function controls the calls to
% S-function routines at each simulation stage.

switch flag,

    case 0
        [sys,x0,str,ts] = mdlInitializeSizes; % Initialization

    case 2
        sys = mdlUpdate(t,x,u); % Update discrete states

    case 3
        sys = mdlOutputs(t,x,u); % Calculate outputs

    case 9
        sys = mdlTerminate(t,x,u); % Plot Diagrams by Terminate

    case { 1, 4 }
        sys = []; % Unused flags

    otherwise
        error(['Unhandled_flag_',num2str(flag)]); % Error
            handling
end;

%=====
% Function mdlInitializeSizes initializes the states, sample
% times, state ordering strings (str), and sizes structure.
%=====
function [sys,x0,str,ts] = mdlInitializeSizes

```

```

L          = 0.8;          % in m
vm_vorgabe = 0.5;        % in m/s

% Call function simsizes to create the sizes structure.
sizes = simsizes;
% Load the sizes structure with the initialization information.
sizes.NumContStates= 0;
sizes.NumDiscStates= 9;
sizes.NumOutputs= 9;
sizes.NumInputs= 1;
sizes.DirFeedthrough=1;
sizes.NumSampleTimes=1;
% Load the sys vector with the sizes information.
sys = simsizes(sizes);
%

theta = 0;          % in rad
xp     = 0;          % in m
xm     = L;          % in m
yp     = 0;          % in m
ym     = 0;          % in m
vm     = vm_vorgabe; % in m/s
vp     = 0;          % in m/s
last_theta = 0;
last_alpha = 0;

%      1      2 3  4 5  6 7      8      9
x0 = [theta , xm,ym, xp,yp, vm,vp, last_theta , last_alpha]; %
      continuous states
%
str = []; % No state ordering
%
ts = [-1 0]; % Inherited sample time
% End of mdlInitializeSizes.
%=====
% Update the discrete states
%=====
function sys = mdlUpdate(t,x,u)

pi          = 3.141593;
L            = 0.8;          % in m

H = 0.048;

alpha = u;

disp ([ 'alpha_=_', num2str(u) ] );
disp ([ 'theta_=_', num2str(x(1)) ] );
disp ([ 'xm_=_', num2str(x(2)) ] );

```

```

vm      = x(6);
last_theta = x(8)
last_alpha = x(9);

vp = vm * cos(alpha);

theta = x(1) + (1/L) * ( 0.5*H*vm*sin(-last_alpha) + 0.5*H*
    vm*sin(-alpha) );

xm = x(2) + 0.5*H*vm*cos( last_theta-last_alpha ) + 0.5*H
    *vm*cos( theta-alpha );

ym = x(3) + 0.5*H*vm*sin( last_theta-last_alpha ) + 0.5*H
    *vm*sin( theta-alpha );

xp = x(4) + 0.5*H*vp*cos(last_theta) + 0.5*H*vp*cos(theta
    );

yp = x(5) + 0.5*H*vp*sin(last_theta) + 0.5*H*vp*sin(theta
    );

last_alpha = alpha;
last_theta = theta;

x = [theta , xm,ym, xp,yp, vm,vp, last_theta , last_alpha]; %
    continuous states

sys = x;
% End of mdlUpdate.
%=====
% Function mdlOutputs performs the calculations.
%=====
function sys = mdlOutputs(t,x,u)

sys = [x(1),x(2),x(3),x(4),x(5),x(6),x(7),x(8),x(9)];

% End of mdlOutputs.
%=====
% Function mdlTerminate performs the plots.
%=====
function sys = mdlTerminate(t,x,u)

sys = [x(1),x(2),x(3),x(4),x(5),x(6),x(7),x(8),x(9)];

% End of mdlTerminate.

```

E Gleichungen zur Dreiecksberechnung

Im Folgenden sind die wesentlichen mathematischen Grundlagen für die geometrischen Berechnungen aufgelistet.

Kosinussatz

$$a^2 = b^2 + c^2 - 2bc \cos \alpha \quad (155)$$

$$c^2 = a^2 + b^2 - 2ab \cos \gamma \quad (156)$$

Sinussatz

$$\frac{a}{\sin \alpha} = \frac{b}{\sin \beta} = \frac{c}{\sin \gamma} \quad (157)$$

Satz des Pythagoras am rechtwinkligen Dreieck

$$a^2 + b^2 = c^2 \quad (158)$$

F Berechnung eines exemplarischen Arbeitspunktes aus der Linearisierung

Ein konkretes Beispiel:

$$\begin{aligned}\alpha &= -30^\circ = -0,523\text{rad} & v_m &= 0,5\frac{\text{m}}{\text{s}} \\ \theta &= 0^\circ = 0\text{rad} & L &= 0,8\text{m}\end{aligned}$$

$$\begin{aligned}\Delta\dot{\theta} &\approx \frac{v_m}{L} \cdot \cos(-\alpha)|_{AP} \cdot \Delta\alpha \\ &\approx \underbrace{0,541\frac{1}{\text{s}}}_{k_{\theta\alpha}} \cdot \Delta\alpha\end{aligned}\quad (159)$$

$$\begin{aligned}\Delta\dot{x}_m &\approx v_m \cdot \{ [-\sin(\theta) \cdot \cos(\alpha) + \cos(\theta) \cdot \sin(\alpha)]|_{AP} \cdot \Delta\theta \\ &\quad + [\cos(\theta) \cdot (-\sin(\alpha)) + \sin(\theta) \cdot \cos(\alpha)]|_{AP} \cdot \Delta\alpha \} \\ &\approx 0,5\frac{\text{m}}{\text{s}} \cdot (-0,5 \cdot \Delta\theta + 0,5 \cdot \Delta\alpha) \\ &\approx \underbrace{-0,25\frac{\text{m}}{\text{s}}}_{k_{x\theta}} \cdot \Delta\theta + \underbrace{0,25\frac{\text{m}}{\text{s}}}_{k_{x\alpha}} \cdot \Delta\alpha\end{aligned}\quad (160)$$

$$\begin{aligned}\Delta\dot{y}_m &\approx v_m \cdot \{ [\cos(\theta) \cdot \cos(\alpha) - (-\sin(\theta)) \cdot \sin(\alpha)]|_{AP} \cdot \Delta\theta \\ &\quad + [\sin(\theta) \cdot (-\sin(\alpha)) + \cos(\theta) \cdot \cos(\alpha)]|_{AP} \cdot \Delta\alpha \} \\ &\approx 0,5\frac{\text{m}}{\text{s}} \cdot (8,661 \cdot \Delta\theta + 8,661 \cdot \Delta\alpha) \\ &\approx \underbrace{0,433\frac{\text{m}}{\text{s}}}_{k_{y\theta}} \cdot \Delta\theta + \underbrace{0,433\frac{\text{m}}{\text{s}}}_{k_{y\alpha}} \cdot \Delta\alpha\end{aligned}\quad (161)$$

G Messreihen des räumlichen Mittelwertfilters

In Abb. 71 sind die Polarkoordinaten des Scans aus dem Laborraum für die Winkel 0° bis 109° dargestellt. Das Diagramm verdeutlicht die Nachteile der abgeschliffenen Ecken und die Objektübergänge. Das Messrauschen und die Glättung des Filters ist in Abb. 72 veranschaulicht, dort ist nur der Bereich der linken Laborwand aufgetragen.

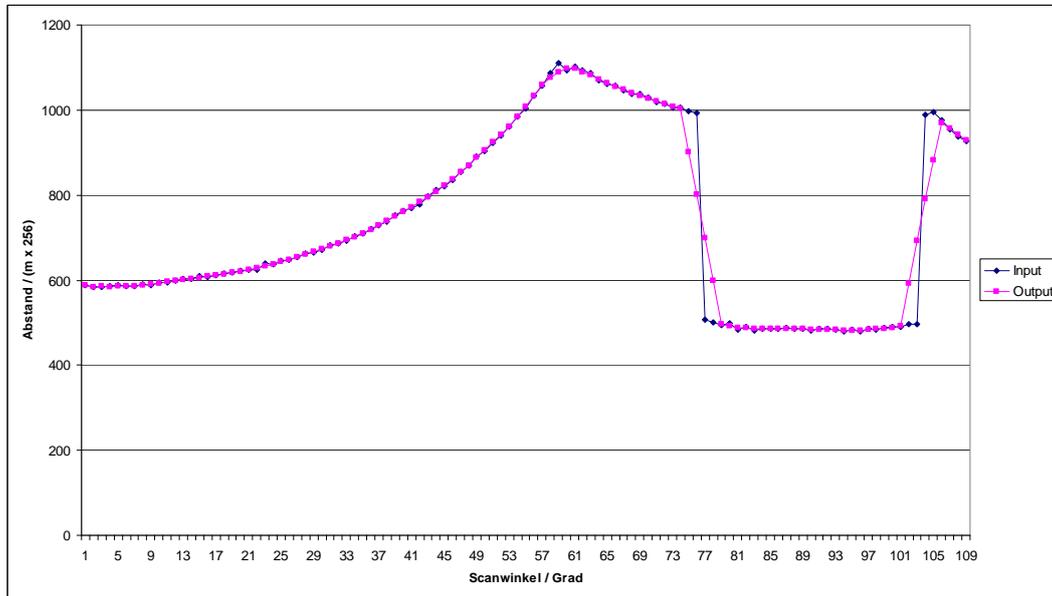


Abb. 71: MA-Filter, ein Scan in Polarkoordinaten, Winkel von 0° bis 109°

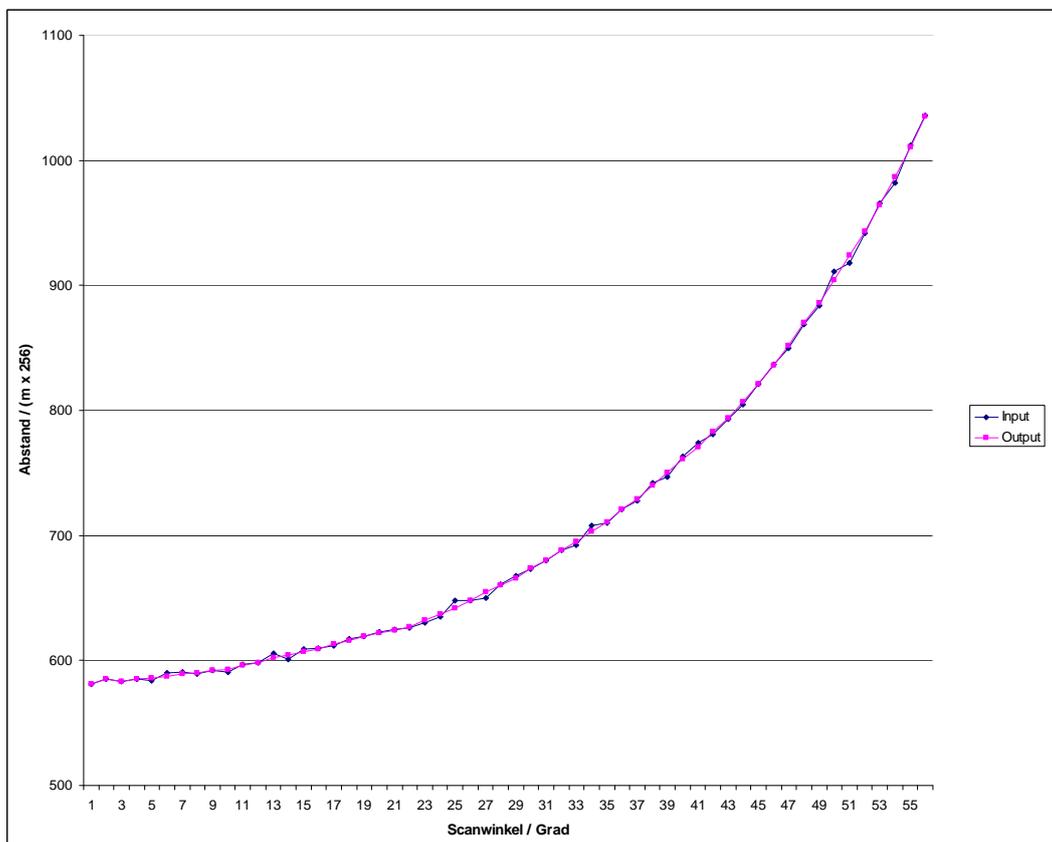


Abb. 72: MA-Filter, ein Scan in Polarkoordinaten, Winkel von 0° bis 56°

n	81	82	83	84	85	86
$x_n \left[\frac{m}{256} \right]$	982	976	978	513	514	506
$y_n \left[\frac{m}{256} \right]$	980	978	822	668	511	509

Tabelle 16: Räumliches Mittelwertfilter $N = 3$

n	79	80	81	82	83	84	85	86	87	88
$x_n \left[\frac{m}{256} \right]$	985	983	985	980	977	515	507	514	509	512
$y_n \left[\frac{m}{256} \right]$	986	985	982	888	792	698	604	511	510	510

Tabelle 17: Räumliches Mittelwertfilter $N = 5$

n	76	77	78	79	80	81	82	83	84	85	86	87	88	89
$x_n \left[\frac{m}{256} \right]$	997	987	984	985	983	979	975	512	505	509	511	506	508	507
$y_n \left[\frac{m}{256} \right]$	994	990	987	984	915	846	778	710	642	575	508	507	508	507

Tabelle 18: Räumliches Mittelwertfilter $N = 7$

H Transformation der Abstandswerte, Beispielrechnung und Rechts-Ausweichen

Beispielrechnung für das Links-Ausweichen:

Parameter: $\theta = 46^\circ$

Rechnung:

$$\beta = \theta + \alpha = 46^\circ + 80,8^\circ = 126,8^\circ$$

$$\beta' = \beta - 90^\circ = 126,8^\circ - 90^\circ = 36,8^\circ$$

$$\alpha' = 90^\circ - \beta' = 90^\circ - 36,8^\circ = 53,2^\circ$$

$$a'_x = \cos(\alpha') \cdot a = \cos(53,2^\circ) \cdot 31,4\text{cm} = 18,81\text{cm}$$

$$a'_y = \sin(\alpha') \cdot a = \sin(53,2^\circ) \cdot 31,4\text{cm} = 25,14\text{cm}$$

$$M_{ax} = S_{ax} - a'_x$$

$$M_{ay} = S_{ay} - a'_y$$

Transformation während des Rechts-Ausweichens:

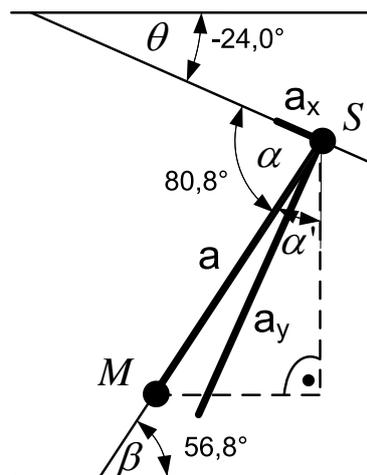


Abb. 73: Transformation der Abstandswerte bei einem Achswinkel von 0° bis -90° (Rechts-Ausweichen)

Parameter: $\theta = -24^\circ$

Rechnung:

$$\beta = \theta + \alpha = -24^\circ + 80,8^\circ = 56,8^\circ$$

$$\beta' = \beta$$

$$\alpha' = 90^\circ - \beta' = 90^\circ - 56,8^\circ = 33,2^\circ$$

$$a'_x = \cos(\alpha') \cdot a = \cos(33,2^\circ) \cdot 31,4\text{cm} = 26,27\text{cm}$$

$$a'_y = \sin(\alpha') \cdot a = \sin(33,2^\circ) \cdot 31,4\text{cm} = 17,19\text{cm}$$

$$M_{ax} = S_{ax} + a'_x$$

$$M_{ay} = S_{ay} - a'_y$$

I Bode-Diagramme zu Regler 3

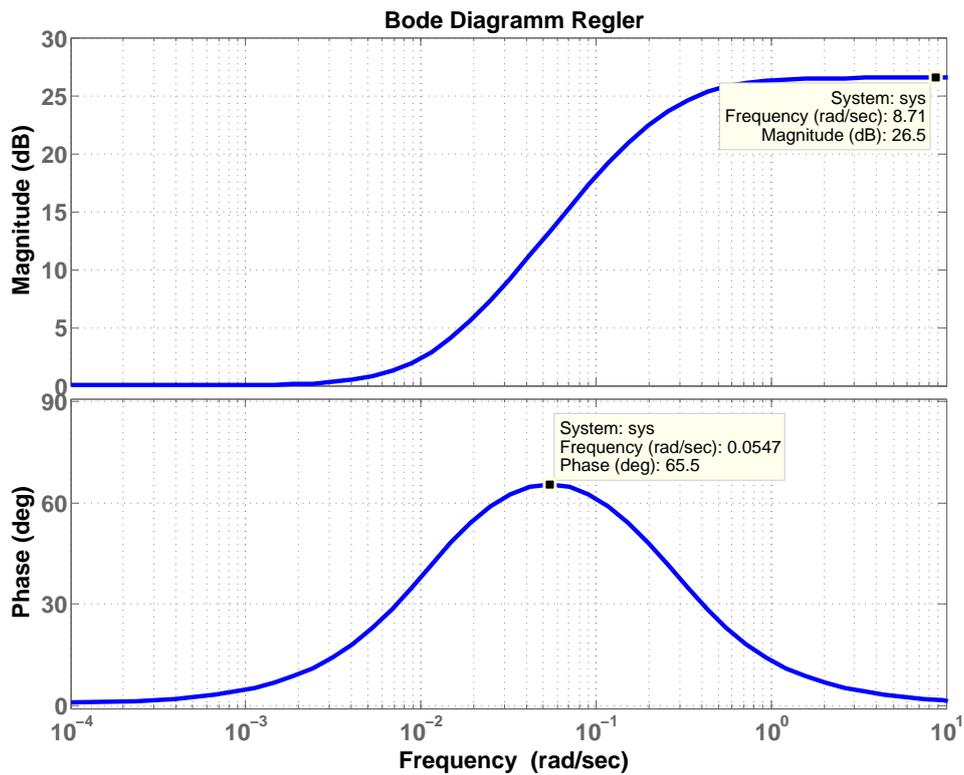


Abb. 74: Bode-Diagramm des Reglers 3

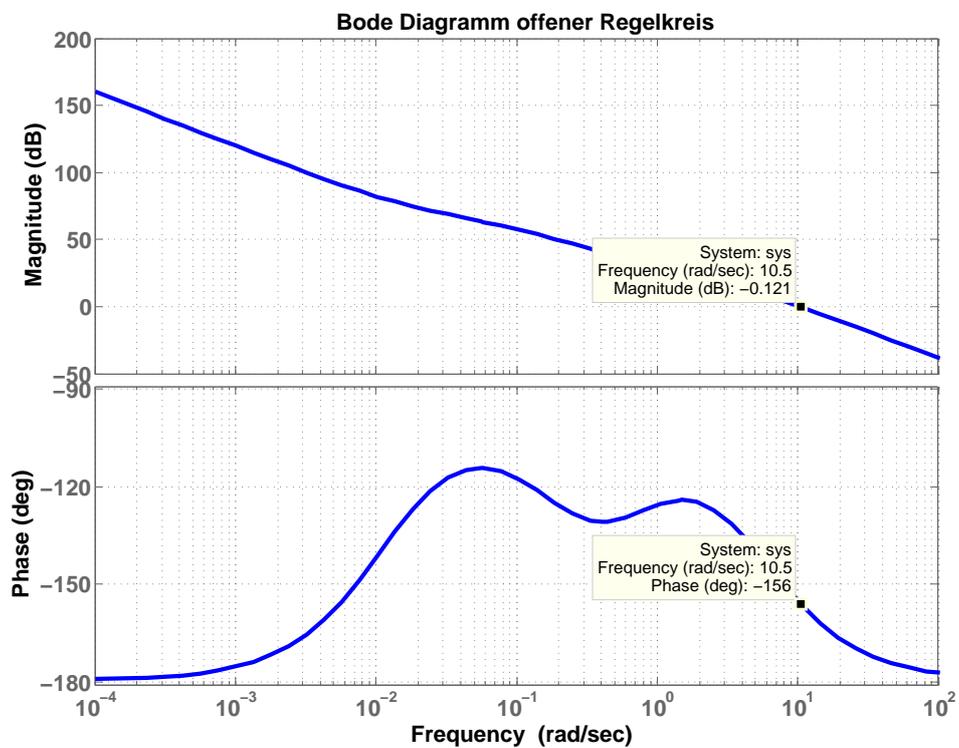


Abb. 75: Bode-Diagramm des offenen Regelkreises zu dem Regler 3

J Dymola Simulationsmodell zum Reglervergleich

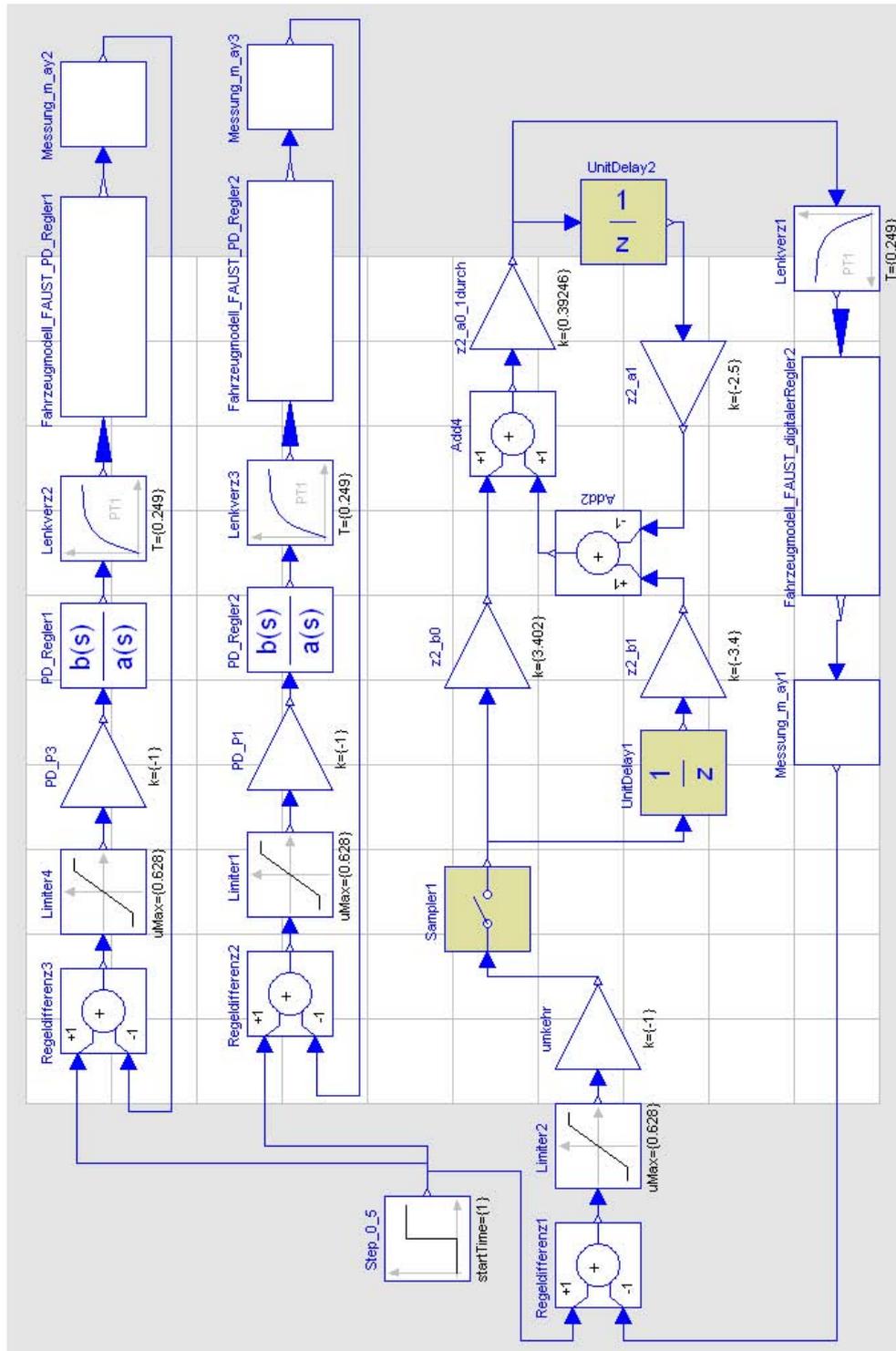


Abb. 76: Kontinuierlicher und diskreter Regelkreis zum Reglervergleich

K Ausgewählter Sourcecode des Ausweichassistenten

Listing 5: Räumliches Mittelwertfilter; *LDComm.c*

```

/*-----
 * auswertung :
 * Diese Funktion wertet die vom Scanner gesendeten Daten aus
 * und schreibt sie in ein Array
 *-----*/
int auswertung(T_LD_Msg** canMsg)
{
    T_Nachricht *response_msg;
    short msgs=0, config=0, laysec=0, profil=0;
    short i=0, j=0 ,h=0;

    short scan_daten [2][ANZAHL_SCANWERTE];

    void* p;
    short intData [4];
    p = (*canMsg)->DATA+2;
    memcpy((void*)&msgs ,p, 2);

    /* Einlesen der Nachrichten */
    do
    {
        response_msg = (T_Nachricht*)msg_dequeue(mq_LD_Recv);

        *canMsg = (T_LD_Msg*)&response_msg->nutzdaten;

        memcpy((void*)&intData [0] ,(*canMsg)->DATA, 2);
        memcpy((void*)&intData [1] ,(*canMsg)->DATA+2, 2);
        memcpy((void*)&intData [2] ,(*canMsg)->DATA+4, 2);
        memcpy((void*)&intData [3] ,(*canMsg)->DATA+6, 2);

        /* Zweite Nachricht ? */
        if (intData [0] == msgs-1)
        {
            /* ja, dann sichere ... */
            config = intData [1];           // die configuration
            laysec = intData [2];          // layer und sectoren
                zahl
            profil = intData [3];          // profilnummer
        }
        else
        {
            for (h=1 ;h<=3 && i < ANZAHL_SCANWERTE;h++)
            {
                // sichere die daten
                scan_daten [0][i] = intData [h];
            }
        }
    }
}

```

```

        if ( i >= (MA_BREITE-1) && i < ANZAHL_SCANWERTE
            )
        {
            // RM-Filter
            scan_daten [1][ i - ((MA_BREITE-1)/2) ] = (
                scan_daten [0][ i -4]
                    + scan_daten [0][ i -3]
                    + scan_daten [0][ i -2]
                    + scan_daten [0][ i -1]
                    + scan_daten [0][ i ] ) /
                MA_BREITE;
        }
        if ( i < (MA_BREITE-1) || i > ANZAHL_SCANWERTE - (((
            MA_BREITE-1)/2)+1) )
        {
            // Anfang und Ende, Grenzbereiche
            ungefiltert
            scan_daten [1][ i ] = scan_daten [0][ i ];
        }
        i++;
    }
}

/* Letzte Profilmnachricht – beenden der Auswertung */
if (intData [0] == 1)
{
    /* zurücksetzen des Index der Entfernungsdaten */
    i = 0;
}
free (response_msg);
} while (intData [0] != 1);

pthread_mutex_lock (&scanwerte . sema);

memcpy (scanwerte . werte ,
        scan_daten [1],
        ANZAHL_SCANWERTE * sizeof (scan_daten [0][0]));

pthread_cond_signal (&scanwerte . condvar);
pthread_mutex_unlock (&scanwerte . sema);

return      EXIT_SUCCESS;
}

```

Listing 6: Ausweichstatus Links; *AusweichstatusAbstRegl.c*

```

unsigned char get_ausweichStatusLinksAbstRegl (double
    kurzObjGrenzeX ,
                                double kurzObjGrenzeY , double
                                linkeObjGrenze)

```

```

{
    unsigned char Status = FALSE;
    T_Polar_Koord* polar;

    polar = (T_Polar_Koord*) malloc (sizeof (T_Polar_Koord));
    kartesToPolar( kurzObjGrenzeX, kurzObjGrenzeY, polar);

    if( ( (((kurzObjGrenzeX-SCANNER_HINTERRAD_X) * M_FAKTOR) <
        1.5) || // kleiner 1,5m oder
        ( ((kurzObjGrenzeX-SCANNER_HINTERRAD_X) *
          M_FAKTOR) < // wenn x < y
          ((linkeObjGrenze+ 2*SICHERHEITS_SEITENABSTAND
            + GRENZE_LINKS + SCANNER_HINTERRAD_Y)*
            M_FAKTOR)
          ) )
        && (fabs(polar->phi) < 0.17) && (get_speed
          () !=0) ) // kleiner 10°, muss
          immer gegeben sein
    {
        Status = TRUE;
    }

    return Status;
}

```

Listing 7: Entscheidungsautomat Z_ENT_L externe Reaktion; Z_ENT_L.c

```

/*-----
 * Z_ENT_L_ext_reaktion :
 * In dieser Funktion wird auf Ereignisse reagiert,
 * die einen Zustandswechsel zur Folge haben.
 *-----*/
T_zst_reaktion Z_ENT_L_ext_reaktion (T_aktivitaet*
arg_aktivitaet ,REF_T_Nachricht arg){
    T_zst_reaktion reaktion = ZST_Hat_Reagiert;
    int signal = (int)((T_Nachricht*)arg)->nutzdaten.ca_signal;

    switch(signal){
        case L_off:
            zustand_annehmen( arg_aktivitaet ,ID_Z_ENT);
            break;
        case R_on:
            zustand_annehmen( arg_aktivitaet ,ID_Z_ENT_LR);
            manoeverAbstRegl = MANOEVER_AUSWEICHENRECHTS;
            break;
        default:
            reaktion = ZST_Hat_Nicht_Reagiert;
            break;
    }
    return reaktion;
}

```

}

Listing 8: Funktionen zur relativen Positionsbestimmung des SCV; *AusweichenAbstRegl.c*

```

/*+++++++ berechne_vor_hindernis ++++++*/
T_Pos_Daten_a* berechne_vor_hindernis(T_Polar_Koord links ,
    T_Polar_Koord rechts , T_Polar_Koord kurz)
{
    // Variablen , die berechnet werden
    T_Pos_Daten_a* daten;

    // Hilfsvariablen
    double a , a1 , b , b1 , c;
    double alpha , w_b1 , w_c;    // w fuer Winkel

    daten = (T_Pos_Daten_a*) malloc (sizeof (T_Pos_Daten_a));

    // Hilfsvariablenzuweisung
    if (manoeverAbstRegl == MANOEVER_AUSWEICHENLINKS)
    {
        b    = rechts.r;
        c    = links.r;
        w_c  = links.phi;
    }
    else
    {
        b    = links.r;
        c    = rechts.r;
        w_c  = rechts.phi;
    }
    b1 = kurz.r;
    alpha = links.phi - rechts.phi;
    w_b1 = kurz.phi;

    // Berechnungen nach Gleichungen
    a = sqrt( b*b + c*c - 2*b*c*cos(alpha) );
    if ( c > b1)
    {
        a1 = sqrt( c*c - b1*b1 );
    }
    else {
        a1 = 0;
    }
    daten->s_ax = b1;
    if (manoeverAbstRegl == MANOEVER_AUSWEICHENLINKS)
    {
        daten->s_ay = (a/2) - a1;
        // theta = Winkel vom kürzesten Punkt
        daten->theta = -kurz.phi;
    }
}

```

```

    else
    {
        daten->s_ay = - ((a/2) - a1);
        // theta = Winkel vom kürzesten Punkt
        daten->theta = kurz.phi;
    }

    daten->a = a;
    // Daten in Polarkoordinaten , s_ax und s_ay in m, theta in
    rad
    return daten;
}
/*+++++++ berechne_schraeg_vor_hindernis ++++++*/
T_Pos_Daten* berechne_schraeg_vor_hindernis(T_Polar_Koord links
, T_Polar_Koord rechts , T_Polar_Koord kurz)
{
    // Variablen , die berechnet werden
    T_Pos_Daten* daten;

    // Hilfsvariablen
    double a1 , a2 , b1 , b2 , c;
    double alpha1 , alpha2 , beta1 , beta2 , delta , gamma;

    daten = (T_Pos_Daten*) malloc (sizeof (T_Pos_Daten));

    if( (manoeverAbstRegl==MANOEVER_AUSWEICHENLINKS && rechts.
        phi!=kurz.phi)
        || (manoeverAbstRegl==MANOEVER_AUSWEICHENRECHTS &&
            links.phi!=kurz.phi)
        )
    {
        // Hilfsvariablenzuweisung
        if (manoeverAbstRegl == MANOEVER_AUSWEICHENLINKS)
        {
            alpha1 = -(rechts.phi - kurz.phi);
            b1      = rechts.r;
        }
        else
        {
            alpha1 = links.phi - kurz.phi;
            b1      = links.r;
        }
        c          = kurz.r;
        delta      = -kurz.phi;

        // Berechnungen nach Gleichungen
        a1         = sqrt( b1*b1 + c*c - 2*b1*c*cos(alpha1) );
        if (manoeverAbstRegl == MANOEVER_AUSWEICHENLINKS)

```

```

    {
        gamma = acos( (a1*a1 + b1*b1 - c*c)/(2*a1*b1) );
        beta1 = M_PI - gamma - alpha1;
    }
    else
    {
        gamma = acos( (a1*a1 + b1*b1 - c*c)/(2*a1*b1) );
        beta1 = M_PI - gamma - alpha1;
    }

    beta2 = M_PI - beta1;
    alpha2 = M_PI/2 - beta2;
    b2 = cos(alpha2)*c;
    a2 = sqrt(c*c - b2*b2);

    daten->s_ax = b2;

    if (manoeverAbstRegl == MANOEVER_AUSWEICHENLINKS)
    {
        daten->theta = (delta - alpha2);
        daten->s_ay = (a1/2) + a2;
    }
    else
    {
        daten->theta = (delta + alpha2);
        daten->s_ay = -((a1/2) + a2);
    }
}
else
{
    daten->s_ax = 0;
    daten->s_ay = 0;
    daten->theta = 0;
}

return daten;
}

/*+++++++ berechne_neben_hindernis ++++++*/
T_Pos_Daten* berechne_neben_hindernis(T_Polar_Koord links ,
    T_Polar_Koord rechts , T_Polar_Koord kurz)
{
    // Variablen , die berechnet werden
    T_Pos_Daten* daten;

    // Hilfsvariablen
    double a,b,c;
    double alpha , beta , gamma;

```

```

daten = (T_Pos_Daten*) malloc (sizeof (T_Pos_Daten));

// Hilfsvariablenzuweisung
if (manoeverAbstRegl == MANOEVER_AUSWEICHENLINKS)
{
    alpha = links.phi - rechts.phi;
    b      = links.r;
    c      = rechts.r;
}
else
{
    alpha = links.phi - rechts.phi;
    b      = rechts.r;
    c      = links.r;
}

// Berechnungen nach Gleichungen
a      = sqrt( b*b + c*c - 2*b*c*cos(alpha) );

if (manoeverAbstRegl == MANOEVER_AUSWEICHENLINKS)
{
    gamma = acos( (a*a + b*b - c*c)/(2*a*b) );
    beta  = M_PI - gamma - alpha;
}
else
{
    gamma = acos( (a*a + b*b - c*c)/(2*a*b) );
    beta  = M_PI - gamma - alpha;
}

daten->s_ax = 0;

if (manoeverAbstRegl == MANOEVER_AUSWEICHENLINKS)
{
    daten->theta = beta - M_PI/2;    // Naeherungsweise
    daten->s_ay = rechts.r;
}
else
{
    daten->theta = -(beta - M_PI/2); // Naeherungsweise
    daten->s_ay = -links.r;
}

return daten;
}

```

Listing 9: Schleife zur Aktionsausführung des Ausweichmanövers; *AusweichenAbstRegl.c*

```

while( (manoever_is_running_abst_regl) && obj != NULL){

```

```

// Positionsdaten berechnen
if( position == VOR_HINDERNIS )
{
    pos_daten_a = berechne_vor_hindernis(*links , *rechts , *
        kurz);
    if( !isInitOdometrie )
    {
        resetOdometrieDaten();
        setOdometrieAchswinkel( pos_daten_a->theta );
        isInitOdometrie = TRUE;
    }
    pos_daten = (T_Pos_Daten*) malloc ( sizeof (T_Pos_Daten)
        );
    pos_daten->s_ax = pos_daten_a->s_ax;
    pos_daten->s_ay = pos_daten_a->s_ay;
    pos_daten->theta = pos_daten_a->theta;
    pos_daten = trans_S_to_M( pos_daten );
    if( ( fabs( pos_daten_a->a - last_a ) > 0.15 ) && ( started
        == TRUE ) )
    {
        position = SCHRAEG_VOR_HINDERNIS; // wenn SCV
            schraeg vor Hindernis
        printf("!!!_SCHRAEG_VOR_HINDERNIS_!!!\n");
        printOdometrieDaten();
    }
    last_a = pos_daten_a->a;
    free( pos_daten_a );
}
if( position == SCHRAEG_VOR_HINDERNIS )
// kein else , wenn in 'vor Hindernis' festgestellt wird ,
// das ungueltige Berechnung ,
// denn dann direkt 'schraeg vor Hindernis'
{
    pos_daten = berechne_schraeg_vor_hindernis(*links , *
        rechts , *kurz);
    if( pos_daten->s_ax != 0 || pos_daten->theta != 0 )
    {
        pos_daten = trans_S_to_M( pos_daten );
    }
    else
    {
        position = NEBEN_HINDERNIS;
        printf("!!!_NEBEN_HINDERNIS_!!!\n");
        printOdometrieDaten();
    }
}
if( position == NEBEN_HINDERNIS )
{

```

```

    pos_daten = berechne_neben_hindernis(*links , *rechts , *
        kurz);
    pos_daten = trans_S_to_M( pos_daten );

    // idealisierte Ausweichkurve ist abgefahren
    anzahl_kurve = 1;
    // Fuehrungsgroesse Ys1 auf SSA setzen
    kurve[1] = ( 1.5*SICHERHEITS_SEITENABSTAND +
        GRENZE_LINKS)*M_FAKTOR;

    if(pos_daten->theta < 0 ) // Abbruchkriterium
    {
        remote_speed = 0;
        manoever_is_running_abst_regl = FALSE;
        printf(" !!!_FERTIG_!!!\n");
        if( !SIMULIERE_SCANDATEN ){
            daten_odo = getOdometrieDaten();
            printOdometrieDaten();
        }
    }
}

ist_lenkwinkel = get_angle() / 10.0;
achswinkel = pos_daten->theta;

#ifdef AUSGABE
    set_ausgabe_parameter((int)(pos_daten->s_ax * ((double)1/(
        double)H_KURVE)),
        (int)((x_max-pos_daten->s_ax)*((double)1/(double)
            H_KURVE)), achswinkel);
#endif

if( achswinkel * (180/M_PI) > 2.0)
{
    started = TRUE;
}
// Ende der kurve ist Fuehrungsgroesse
// soll_lenkwinkel ist Reglerausgangsgroesse
soll_lenkwinkel = pd_regler(kurve[anzahl_kurve], *pos_daten
    );

soll_msg = (T_Nachricht*)malloc(sizeof(T_Nachricht));
soll_msg->n_art = CAS_Soll_Recv;
soll_msg->nutzdaten.soll_recv_msg.abs_Speed = v_soll;
soll_msg->nutzdaten.soll_recv_msg.rel_Direction = (short)((
    double)(soll_lenkwinkel * 10.0));
msg_enqueue(msgRecvQueue, soll_msg);
if(remote_control == TRUE)
{

```

```

        remote_angle = (short)((double)(soll_lenkwinkel * 10.0)
            );
    }

    // delay a time
    nanosleep(&itime.it_value ,NULL);

    // neue Objektdaten uebernehmen
    obj = get_objekt(objektID);
    if(obj != NULL){
        // Punkte in Polarkoordinate für Algo. speichern
        kart_links->x = obj->Links.X * M_FAKTOR;
        kart_links->y = obj->Links.Y * M_FAKTOR;
        kart_links = trans_P_to_S(kart_links ,achswinkel);

        kart_rechts->x = obj->Rechts.X * M_FAKTOR;
        kart_rechts->y = obj->Rechts.Y * M_FAKTOR;
        kart_rechts = trans_P_to_S(kart_rechts ,achswinkel);

        kart_kurz->x = obj->Kurz.X * M_FAKTOR;
        kart_kurz->y = obj->Kurz.Y * M_FAKTOR;
        kart_kurz = trans_P_to_S(kart_kurz ,achswinkel);

        kartesToPolar(kart_links->x, kart_links->y, links);
        kartesToPolar(kart_rechts->x, kart_rechts->y, rechts);
        kartesToPolar(kart_kurz->x, kart_kurz->y, kurz);
    }
}

```

Listing 10: Implementierung des PD-Reglers; *Regler.c*

```

#include <faust_globals.h>
#include <version.h>
#include <AusweichenAbstRegl.h>
#include <Math.h>

/*+++++++ Exportierte Header
+++++++*/
#include <Regler.h>

/*+++++++ Interne Daten
+++++++*/
double last_regl_diff = 0;           // 1/z Glied
double last_z_a0_d    = 0;           // 1/z Glied

/*+++++++ reset_regler
+++++++*/
void reset_regler(void)
{
    last_regl_diff = 0;           // 1/z Glied

```

```
    last_z_a0_d    = 0;           // 1/z Glied
}

/*+++++++ pd_regler
+++++++*/
double pd_regler( double fuehrung_ay , T_Pos_Daten pos_daten)
{
    int    kurve_zugriff;
    double soll_lenkwinkel = 0.0;
    double s_ax;
    double s_ay;
    double regl_diff;           // Regeldifferenz
    double sum1, sum2;

    // Reglerparameter nach Matlab Berechnung
    // sample period ist nach timer: 0,048 sec
    // nach Regler 2
//    double z_b0    = 3.402;
//    double z_b1    = -3.4;
//    double z_a0_d  = 0.39246;
//    double z_a1    = -2.5;

    // nach nach Regler 3
    double z_b0    = 3.402;
    double z_b1    = -3.4;
    double z_a0_d  = 0.24704;
    double z_a1    = -4;

    // aktuelle Messwerte
    s_ax = pos_daten.s_ax;
    s_ay = pos_daten.s_ay;

    // Bestimmung der Regeldifferenz
    regl_diff = fuehrung_ay - s_ay;

    // Limiter zur Vermeidung von Lenkwinkeluebersteuerung und
    // Vorzeichenumkehr
    if( regl_diff < -0.628)
    {
        regl_diff = 0.628;
    }
    else if( regl_diff > 0.628)
    {
        regl_diff = - 0.628;
    }
    else
    {
        regl_diff = - regl_diff;
    }
}
```

```

    // digitaler PD-Regler
    sum1 = last_regl_diff * z_b1 - last_z_a0_d * z_a1;
    sum2 = regl_diff * z_b0 + sum1;
    soll_lenkwinkel = sum2 * z_a0_d;

    // 1/z Glieder
    last_regl_diff = regl_diff;
    last_z_a0_d     = soll_lenkwinkel;

    return soll_lenkwinkel * (180/M_PI);
}

```

Listing 11: Implementierung der Odometrie; *odometrie.c*

```

/*+++++++ Importierte Header ++++++*/

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
#include <time.h>
#include <faust_globals.h>
#include <sys/neutrino.h>
#include <Utils.h>
#include <Math.h>
#include <syslog.h>

#ifdef THREAD_TIMINGS
#include <sys/neutrino.h>
#include <inttypes.h>
#include <hw/inout.h>
#endif

/*+++++++ Exportierte Header ++*/
#include <Odometrie.h>

/*+++++++ Interne Konstanten ++*/

#define DELTA_T 0.05 // 20Hz --> 0,05 s
#define RADSTAND 0.8

/*+++++++ Interne Daten ++++++*/

Odometrie* daten_odo;

double t = 0.0;

double xm = RADSTAND; // in m
double ym = 0.0;

```

```

double xp      = 0.0;
double yp      = 0.0;

double vp      = 0.0;

double theta_odo = 0;

double last_vm = 0;
double last_vp = 0;
double last_alpha = 0;
double last_theta = 0;

/*+++++
 *      Interne Funktionen
 *+++++*/

/*-----
 * Funktionsname : berechne_Odometrie
 * Funktionsbeschreibung :
 * berechnet die aktuelle Position in
 * x- und y- Richtung und den Achswinkel
 * mit Vorderradgeschwindigkeit, Lenkwinkel
 * und Radstand. Das heisst, die Berechnungen
 * sind nur fuer MODE_FRONT gueltig.
 *-----*/

void berechne_Odometrie(double vm, double alpha, double h)
{
    // Berechnung der Odometrie mit der Trapezintegration

    t = t + h;

    vp = vm * cos(alpha);

    theta_odo = last_theta + (1/RADSTAND) * ( 0.5*h*last_vm*sin(-
        last_alpha) + 0.5*h*vm*sin(-alpha) );

    xm = xm + 0.5*h*last_vm*cos( last_theta-last_alpha ) + 0.5*h
        *vm*cos( theta_odo-alpha );

    ym = ym + 0.5*h*last_vm*sin( last_theta-last_alpha ) + 0.5*h
        *vm*sin( theta_odo-alpha );

    xp = xp + 0.5*h*last_vp*cos( last_theta ) + 0.5*h*vp*cos(
        theta_odo );

    yp = yp + 0.5*h*last_vp*sin( last_theta ) + 0.5*h*vp*sin(
        theta_odo );

```

```

    // speichern fuer naechsten Berechnungsschritt
    last_vm = vm;
    last_vp = vp;
    last_alpha = alpha;
    last_theta = theta_odo;

    // Position und Achswinkel speichern
    daten_odo->xm = xm;
    daten_odo->ym = ym;
    daten_odo->theta = theta_odo;
}

/*+++++++
 *      Externe Funktionen
 *+++++++*/

/*-----
 * Funktionsname : *odometrie_thread
 * Funktionsbeschreibung :
 * Repraesentiert den Odometrie Thread
 * empfaengt Ist Daten von den Arm Knoten
 * und ruft damit berechne_Odometrie auf
 *-----*/

void *odometrie_thread(void) {

    T_Nachricht      *nachricht;

    daten_odo = (Odometrie*) malloc (sizeof (Odometrie));
    resetOdometrieDaten();

    printf("Odometrie_Thread_started_\n");

    do {

        /* auf nachricht warten */
        nachricht = (T_Nachricht*)msg_dequeue(MQ_Odometrie);

        if(nachricht != NULL){
            /* nachricht verarbeiten */
            double angle_front = 0;
            double angle_rear  = 0;
            double speed_front = 0;
            double speed_rear  = 0;

            switch( nachricht->n_art){

                case Ist_Recv:

```

```

        angle_front = ((double) nachricht->nutzdaten .
            ist_recv_msg.angle_front) / 10.0;
        speed_front = ((double) nachricht->nutzdaten .
            ist_recv_msg.speed_front) /1000.0;
        angle_rear = ((double) nachricht->nutzdaten .
            ist_recv_msg.angle_rear) / 10.0;
        speed_rear = ((double) nachricht->nutzdaten .
            ist_recv_msg.speed_rear) /1000.0;
        break;
    default:
        printf("unhandled_msg_in_Odometrie");
    }
    if( speed_front > 0 )
    {
        angle_front = angle_front * ((double)M_PI/((double)
            180);
        berechne_Odometrie( speed_front , angle_front ,
            DELTA_T);
    }
    free(nachricht);
}
} while ( 1 );

exit(0);
}

/*-----
 * Funktionsname : getOdometrieDaten
 * Funktionsbeschreibung :
 * gibt die aktuellen Odometriedaten zurueck
 *-----*/

Odometrie* getOdometrieDaten(void)
{
    return daten_odo;
}

/*-----
 * Funktionsname : setOdometrieAchswinkel
 * Funktionsbeschreibung :
 * setzt den Achswinkel in den Odometriedaten
 *-----*/

void setOdometrieAchswinkel(double in_theta)
{
    daten_odo->theta = in_theta;
}

/*-----

```

```
* Funktionsname : resetOdometrieDaten  
* Funktionsbeschreibung : setzt alle Odometriedaten auf 0  
-----*/
```

```
void resetOdometrieDaten(void)
```

```
{  
    daten_odo->xm    = 0;  
    daten_odo->ym    = 0;  
    daten_odo->theta = 0;  
  
    t    = 0.0;  
  
    xm    = RADSTAND;    // in m  
    ym    = 0.0;  
  
    xp    = 0.0;  
    yp    = 0.0;  
  
    vp    = 0.0;  
  
    theta_odo = 0;  
  
    last_vm = 0;  
    last_vp = 0;  
    last_alpha = 0;  
    last_theta = 0;  
}
```

```
/*-----*/  
* Funktionsname : printOdometrieDaten  
* Funktionsbeschreibung :  
* gibt alle Odometriedaten aus  
-----*/
```

```
void printOdometrieDaten(void)
```

```
{  
    printf("odo->xm: %f\n", daten_odo->xm );  
    printf("odo->ym: %f\n", daten_odo->ym );  
    printf("odo->theta: %f\n", daten_odo->theta );  
}
```

L Kalman-Filter zur Sensordatenfusion

Es wird ein Beispiel zur Sensordatenfusion mit der Wurfparabel aus [16] präsentiert. Die Methode zur Datenfusion ist [23] entnommen.

Die Gleichungen für das Kalman Filter mit der Wurfparabel wurden in der Datei *Kalman.m* in eine Funktion gekapselt. In der Datei *Wurfparabel.m* wird eine Wurfparabel simuliert und mit zwei simulierten Sensoren beobachtet, dabei werden die Messwerte zufällig gestört. Die Messwerte je eines Sensors werden an einen Kalman Filter übergeben und dort die Messsignale rekonstruiert. Somit stehen zwei unterschiedliche Rekonstruktionen zur Verfügung. Die Messfehlervarianzen R_k für die beiden Sensoren sind in den Gln. 162 und 163 genannt.

Abhängig von der Varianz des Messfehlers, sprich dem jeweiligen Sensor, gehen die Messwerte zu einem gewichteten Anteil in den fusionierten Messwert ein. Der Faktor für den ersten Sensor wird in Gl. 164 berechnet, der für den zweiten Sensor in Gl. 165. Der fusionierte Messwert $\hat{y}_{fusion}(k)$ wird aus den beiden Schätzwerten $\hat{y}_{kal1}(k)$ und $\hat{y}_{kal2}(k)$ und den beiden bestimmten Faktoren mit Gl. 166 bestimmt. Ein Simulationsergebnis ist in Abb. 77 präsentiert. Der Matlab-Code ist in Listing 12 und 13 dokumentiert.

$$R_{k1} = 0,01 \quad (162) \quad \sigma_1 = \frac{R_{k2}}{R_{k1} + R_{k2}} \quad (164)$$

$$R_{k2} = 0,0225 \quad (163) \quad \sigma_2 = \frac{R_{k1}}{R_{k1} + R_{k2}} \quad (165)$$

$$\hat{y}_{fusion}(k) = (\sigma_1 \cdot \hat{y}_{kal1}(k)) + (\sigma_2 \cdot \hat{y}_{kal2}(k)) \quad (166)$$

- Wurfparabel ideal: Blaue gestrichelte Linie
- Messung Sensor 1: Kreuze in grün
- Messung Sensor 2: Punkte in magenta
- Kalman-Filter 1: Linie in rot
- Kalman-Filter 2: Linie in gelb
- Messwerte fusioniert: Linie in cyan

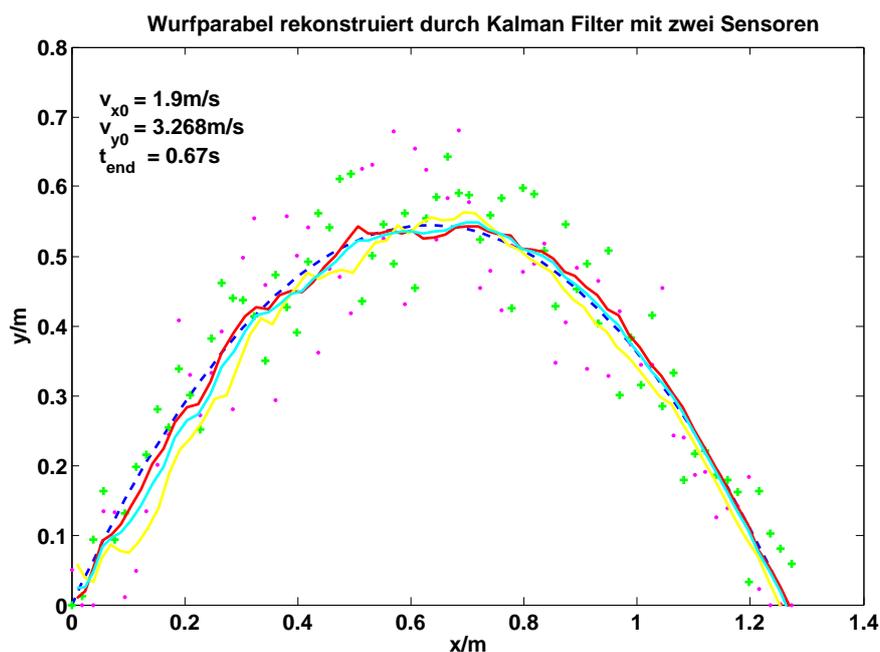


Abb. 77: Kalman-Filter mit zwei Sensoren

Listing 12: Wurfparabel rekonstruiert durch das Kalman Filter mit zwei Sensoren

```

g = 9.80665
H = 0.01
k = 0
n = 68

rk1 = 0.01 % (10/100)^2
random1 = randint(n,1,[-10,10])
rk2 = 0.0225 % (15/100)^2
random2 = randint(n,1,[-15,15])
% -----
% Fuer ideales System
t = [0]

x_ideal = [0];
y_ideal = [0];

v_0 = 3.8
vx_0 = 0.5 * v_0
vy_0 = 0.86 * v_0
vx_ideal = [vx_0];
vy_ideal = [vy_0];

x_fusion = [0];
y_fusion = [0];

for k = 1:n
    % -----
    % ideales System
    t(k+1) = t(k) + H

    x_ideal(k+1) = x_ideal(k) + H * vx_ideal(k)
    y_ideal(k+1) = y_ideal(k) + H * vy_ideal(k) - 0.5 * g * H *
        H

    vx_ideal(k+1) = vx_ideal(k)
    vy_ideal(k+1) = vy_ideal(k) - g * H
end

% Aufruf des jeweiligen Kalman-Filters
[x_kal1, y_kal1, x_mes1, y_mes1] = Kalman(rk1, random1, x_ideal
    , y_ideal);
[x_kal2, y_kal2, x_mes2, y_mes2] = Kalman(rk2, random2, x_ideal
    , y_ideal);

% Fusion
sigmaFaktor_1 = rk2 / (rk1 + rk2)
sigmaFaktor_2 = rk1 / (rk1 + rk2)

```

```

for k = 1:n
    x_fusion(k) = x_kal1(k)
    y_fusion(k) = (sigmaFaktor_1 * y_kal1(k)) + (sigmaFaktor_2 *
        y_kal2(k))
end

% Plot: blau ideal, gruen Messung 1, magenta Messung 2, rot
% Kalman-Filter 1, yellow Kalman-Filter 2, cyan Fusion
plot( x_ideal, y_ideal, '--', x_mes1, y_mes1, 'g+', x_mes2, y_mes2
    , 'm.', x_kal1, y_kal1, 'r', x_kal2, y_kal2, 'y', x_fusion, y_fusion
    , 'c', 'LineWidth', 2);
axis([0,1.4, 0,0.8])
xlabel('x/m', 'FontSize', 14)
ylabel('y/m', 'FontSize', 14)
title('Wurfparabel_rekonstruiert_durch_das_Kalman_Filter_mit_
    zwei_Sensoren', 'FontSize', 14)
str1 = ['v_{x0}_= ' num2str(vx_0) 'm/s'];
str2 = ['v_{y0}_= ' num2str(vy_0) 'm/s'];
str3 = ['t_{end}_= ' num2str(t(k)) 's'];
text(0.03,0.7, str1, 'FontSize', 12)
text(0.03,0.6, str2, 'FontSize', 12)
text(0.03,0.5, str3, 'FontSize', 12)

```

Listing 13: Matlab Funktion für das Kalman-Filter am Beispiel einer Wurfparabel

```

function [x_kal, y_kal, x_mes, y_mes] = Kalman(rk, random, x_ideal,
    y_ideal)

g = 9.80665
H = 0.01
k = 0
n = 68

% -----
% Fuer ideales System
t = [0]

% -----
% Fuer Kalman Filter
Mat_K = zeros(4,4);
Mat_Pm = [1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1] %
    Startschaetzung

Mat_R = [rk 0; 0 rk]
Mat_H = [1 0 0 0; 0 1 0 0]

Vec_Xm = [0 0 1 1].' %
    Startschaetzung
Array_Xm = zeros(4, n+1);
Array_Xm(1) = Vec_Xm(1);

```

```

Array_Xm(2) = Vec_Xm(2);
Array_Xm(3) = Vec_Xm(3);
Array_Xm(4) = Vec_Xm(4);
x_kal = [0];
y_kal = [0];
x_mes = [0];
y_mes  = [0];

Vec_Xp = [0 0 0 0].';

Vec_Y   = [0 0].';

Mat_Pp = zeros(4,4);
Mat_I   = [1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1]

Mat_A   = [1 0 H 0; 0 1 0 H; 0 0 1 0; 0 0 0 1]
Mat_B   = [0 0; 0 ( 0.5 * H*H); 0 0; 0 H]

Vec_u   = [0 -g].';

Mat_Q   = [0.00001 0 0 0; 0 0.00001 0 0; 0 0 0.00001 0; 0 0 0
          0.00001]

for k = 1:n
    % -----
    % Messung

    Vec_Y(1) = x_ideal(k)
    rand      = (random(k)/100)
    Vec_Y(2) = y_ideal(k) + rand

    if (Vec_Y(2) < 0)
        Vec_Y(2) = 0
    end

    x_mes(k) = Vec_Y(1);
    y_mes(k) = Vec_Y(2);

    % -----
    % Korrektur mit der Messung

    Inverse = inv(Mat_R + Mat_H * Mat_Pm * Mat_H')
    Mat_K    = Mat_Pm * Mat_H' * Inverse
    Vec_Xp   = Vec_Xm + Mat_K * (Vec_Y - Mat_H * Vec_Xm)

    Mat_Pp   = (Mat_I - Mat_K * Mat_H) * Mat_Pm

    % -----
    % Prädiktion

```

$$\text{Vec_Xm} = \text{Mat_A} * \text{Vec_Xp} + \text{Mat_B} * \text{Vec_u}$$

$$\text{Array_Xm}((k*4)+1) = \text{Vec_Xm}(1);$$

$$\text{Array_Xm}((k*4)+2) = \text{Vec_Xm}(2);$$

$$\text{Array_Xm}((k*4)+3) = \text{Vec_Xm}(3);$$

$$\text{Array_Xm}((k*4)+4) = \text{Vec_Xm}(4);$$

$$x_kal(k) = \text{Vec_Xm}(1);$$

$$y_kal(k) = \text{Vec_Xm}(2);$$

$$\text{Mat_Pm} = \text{Mat_A} * \text{Mat_Pp} * \text{Mat_A}' + \text{Mat_Q}$$

end

M Inhalt der beigelegten CD

Dieser Masterarbeit ist eine CD beigelegt, die folgende Daten enthält:

- Masterarbeit als PDF-Dokument
- Dymola
 - Linearisiertes Fahrzeugmodell als Blockschaltbild
 - Blockschaltbild mit drei parallelen Regelkreisen zum Vergleichen der Regler
- Grafiken
- Matlab
 - Lenkverzögerung
 - Fahrzeugmodell mit Trapezintegration
 - Kontinuierliches Fahrzeugmodell als Blockschaltbild
 - Linearisierung der Systemgleichungen und *Symmetrisches Optimum*
 - *Frequenzkennlinienverfahren*
 - Kontinuierlicher Ausweichregelkreis
 - Diskreter Ausweichregelkreis
 - Beispiel einer Sensordatenfusion am Beispiel einer Wurfparabel
- SourceCode
 - vollständiger Source Code
- Videos
 - Links-Ausweichen Regler 2
 - Links-Ausweichen Regler 3
 - Rechts-Ausweichen Regler 2
 - Rechts-Ausweichen Regler 3

Versicherung über Selbstständigkeit

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

Hamburg, 12. November 2007

Ort, Datum

Unterschrift