

Bachelorarbeit

Ivo Nikolov

Verfahren zur Fahrbahnverfolgung eines autonomen Fahrzeugs mittels „Pure Pursuit“ und „Follow-the-carrot“

Ivo Nikolov

Verfahren zur Fahrbahnverfolgung eines autonomen
Fahrzeugs mittels „Pure Pursuit“ und „Follow-the-carrot“

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

Betreuender Prüfer: Prof. Dr. rer. nat. Stepfan Pareigis
Zweitgutachter: Prof. Dr. Franz Korf

Abgegeben am

Ivo Nikolov

Thema der Bachelorarbeit

Verfahren zur Fahrbahnverfolgung eines autonomen Fahrzeugs mittels „Pure Pursuit“ und „Follow-the-carrot“

Stichworte

FAUST, Fahrbahnverfolgung, Carolo-Cup, autonomes Fahrzeug, Lenkwinkel, Steuerung, Follow-the-carrot, Pure Pursuit, Polaris

Kurzzusammenfassung

Zur Teilnahme am Carolo-Cup der Technischen Universität Braunschweig wird im Forschungsprojekt FAUST der Hochschule für Angewandte Wissenschaften Hamburg ein autonomes Modellfahrzeug ausgestattet. In dieser Arbeit wird ein Verfahren zur Fahrbahnverfolgung eines autonomen Fahrzeugs entwickelt. Das Verfahren ermittelt kontinuierlich ein Ziel anhand des Spurerkennungssystems, berechnet den Lenkwinkel in Bezug auf dieses Ziel mittels der Algorithmen „Follow-the-carrot“ oder „Pure Pursuit“ und schließlich korrigiert die Richtung des Fahrzeuges.

Ivo Nikolov

Title of the paper

Lane tracking method for an autonomous vehicle using "Pure Pursuit" and "Follow-the-carrot"

Keywords

FAUST, lane tracking, Carolo-Cup, autonomous vehicle, steering angle, Control, Follow-the-carrot, Pure Pursuit, Polaris

Abstract

An autonomous vehicle was equipped in the FAUST project of the University of Applied Sciences Hamburg in order to participate in the Carolo Cup at the Technical University of Braunschweig. The main purpose of this work is to develop a lane tracking method. This method continuously determines a goal point based on the lane detection system, calculates the steering angle using the algorithms "Follow-the-carrot" or "Pure Pursuit" and finally corrects the direction of the vehicle.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation dieser Arbeit.....	1
1.2	Problemstellung	2
1.3	Zielsetzung und Beitrag der Arbeit.....	3
1.4	Aufbau dieser Arbeit.....	4
2	Fahrzeugquerführung.....	5
2.1	Grundlage der Fahrzeugquerführung.....	5
2.2	Systemüberblick der Fahrzeugquerführung.....	7
3	Verfahren zur Fahrbahnverfolgung basiert auf den Spurerkennungsalgorithmus Polaris	10
3.1	Festlegung des Zieles.....	10
3.1.1	Zielentfernung zum Fahrzeug	10
3.1.2	Bestimmung des Zieles	11
3.2	Path Tracking Algorithmen	14
3.2.1	Follow-the-carrot.....	15
3.2.2	Pure Pursuit	15
3.3	Lenkwinkelsteuerung.....	17
4	Test.....	19
4.1	Teststrategie	19
4.2	Sensordatenanalyse.....	20
4.3	Testergebnisse.....	21
4.3.1	Bestimmung der Zielentfernung	21
4.3.2	Pure Pursuit versus Follow-the-carrot, Teststrecke 1	22
4.3.3	Pure Pursuit versus Follow-the-carrot, Teststrecke 2	25
4.3.4	Pure Pursuit versus Follow-the-carrot, Teststrecke 3	28
4.3.5	Auswertung der Algorithmen Pure Pursuit und Follow-the-carrot.....	31
5	Zusammenfassung	32
6	Ausblick.....	34

6.1	Tempomat	34
6.2	Dynamische Zielentfernung.....	34
6.3	Optimierung der Lenkwinkelsteuerung	34
Abbildungsverzeichnis		36
Tabellenverzeichnis		37
Literaturverzeichnis		38
A	Grundlagen der Regelung.....	40
B	PID-Regler Optimierung	42
C	Implementierung der Zielfestlegung.....	45
D	Implementierung der Lenkwinkelermittlung	46

1 Einleitung

Fahrerassistenzsysteme werden in Zukunft immer mehr für die Sicherheit und das Fahrkomfort des Fahrers beitragen. Diese Systeme greifen teilautonom oder autonom in Antrieb, Steuerung (z.B. Gas, Bremse) und Signalisierungseinrichtungen des Fahrzeuges ein. Einige Systeme warnen durch geeignete Mensch-Maschine-Schnittstellen den Fahrer kurz vor oder während kritischer Situationen.

Autonomes Fahren wurde auf Autobahnen und Landstraßen seit Anfang der neunziger Jahre in verschiedenen Projekten erforscht. Am Markt verfügbare Fahrerassistenzsysteme greifen bereits heute aktiv in die Fahrzeuglängs- und Querverführung ein. Für die Zukunft ist der Automatisierung in der Fahrzeugführung von der reinen Assistenz hin zu automatischen Fahrfunktionen denkbar [AAET, 2009]. Zudem werden aktuell kamerabasierte Systeme zur Fahrspurerkennung für die Navigation von autonomen Fahrzeugen entwickelt [(Alexander Kant, 2007), (Dennis Berger, 2008), (Eike Jenning, 2009)].

1.1 Motivation dieser Arbeit

Diese Bachelorarbeit entstand im Rahmen des „Autonomes Fahren“ Projektes, ein Entwicklungsprojekt aus dem Bereich Fahrerassistenz- und Autonome Systeme (FAUST) der HAW-Hamburg. Einer der Schwerpunkte ist das Projekt Carolo-Cup [Carolo-Cup], welches auf einem jährlich stattfindenden Wettbewerb der Technischen Universität Braunschweig basiert. Zu Beginn des Jahres werden die Hochschulen aufgefordert, selbstständig fahrende Modellautos zu entwickeln, welche einen Parcours bewältigen, Hindernissen ausweichen sowie automatisch Einparken können. Diese Wagen treten dann an zwei Wettbewerbstagen mit anschließendem Symposium gegeneinander an und stellen sich einer Jury aus Vertretern der Automobilindustrie.

Schwerpunkt dieser Arbeit ist die Entwicklung einer Steuerungssoftware, die bei der dynamischen Disziplin Rundstrecke des Carolo-Cup zum Einsatz kommt. Diese Software soll mittels einer Spurerkennung eine autonome Fahrbahnverfolgung ermöglichen.

1.2 Problemstellung

Bei der dynamischen Disziplin Rundstrecke des Carolo-Cup soll das Fahrzeug autonom drei Minuten lang auf einem unbekanntem Rundkurs so weit wie möglich fahren. Bei der Straße handelt es sich um die Nachbildung einer Landstraße, bestehend aus langen Geraden, schnellen Kurven, engen Serpentin und Kreuzungen. Die Straße ist konstant 820 mm breit und an den Rändern mit durchgezogenen Linien abgegrenzt. Die zwei Fahrspuren werden durch eine gestrichelte Mittellinie geteilt. Das Verlassen der eigenen Fahrspur mit mehr als einem Rad wird bestraft [Carolo-Cup-Regelwerk 2009].



Abbildung 1: Das Modellfahrzeug Nebula mit Kamera und Embedded-PC

Unser Team Nebula hat mit einem Fahrzeug gearbeitet, welches eine Leihgabe der TU Braunschweig zur Teilnahme am CaroloCup war [Lauk-Reineke, 2007]. Hierbei handelt es sich um ein Modell des VW Touareg im Maßstab 1:10. Das Modell hat ein wannenförmiges Chassis, in dem Motor, Getriebe, Fahrregler, Lenkservo und Akku untergebracht sind. Die Sensorik und Aktorik erfolgen über einen OSEK-Mikrokontroller, welcher über einen Nano-PC mit 1GHz gelesen sowie gesteuert wird. Die Sensordaten werden im 50 Millisekundentakt erfasst.

Zur Umfelderkennung stellt das Fahrzeug die folgenden Sensoren bereit:

- Inkrementalgeber an Vorderrädern
- je zwei Ultraschallsensoren nach vorne und hinten
- je ein Infrarotsensor nach rechts und links
- Kamera

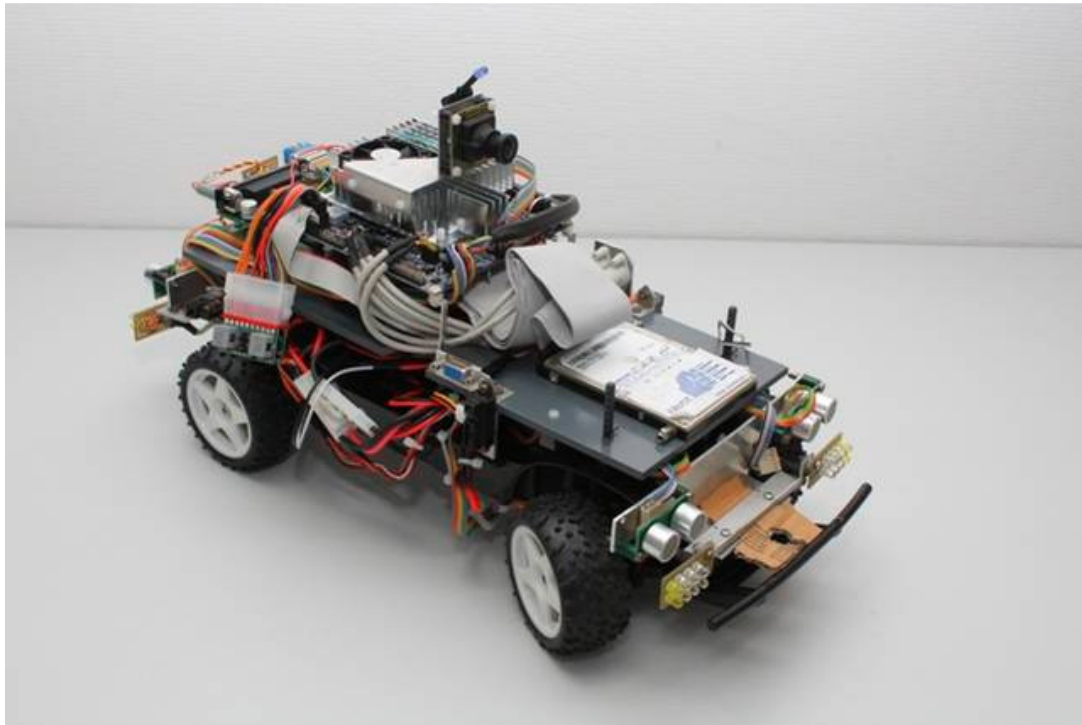


Abbildung 2: Das Modellfahrzeug ohne Karosserie

1.3 Zielsetzung und Beitrag der Arbeit

Folgende Zielvereinbarungen werden für diese Arbeit getroffen:

- Für das Fahrzeug Nebula soll eine autonome Fahrbahnverfolgung, basiert auf dem Fahrspurerkennungssystem Polaris, realisiert werden.
- Die Fahrbahnverfolgung muss auf einer Fahrbahn mit einer Breite von 40 cm und einem minimalen inneren Kurvenradius von 100 cm erfolgen.
- Eine möglichst hohe Geschwindigkeit ist bei der Fahrbahnverfolgung anzustreben, damit das Fahrzeug eine möglichst große Strecke innerhalb von drei Minuten zurücklegen kann.
- Die bestgeeignete Zielentfernung (vgl. Abschnitt 3.1 und Abschnitt 4.3.1) für die Fahrbahnverfolgung soll empirisch ermittelt werden.
- Stabilität der Fahrbahnverfolgung mittels der Algorithmen Pure Pursuit und Follow-the-carrot (vgl. Abschnitt 3.2) soll bei verschiedenen Geschwindigkeiten verglichen werden.

Der Beitrag dieser Arbeit ist die Entwicklung einer Steuerungssoftware, die kontinuierlich den Lenkungswinkel in Bezug auf die Lage und Orientierung des Fahrzeuges auf der Fahrspur ermittelt. Diese werden von dem Spurerkennungsalgorithmus Polaris (vgl. Abschnitt 2.2) bereitgestellt. Anschließend soll diese Software den Stellwert für die Lenkaktorik berechnen, um ein autonomes Fahren zu gewährleisten.

1.4 Aufbau dieser Arbeit

Nach dieser Einführung werden in **Kapitel 2** die Grundlagen der Fahrzeugquerführung definiert. Zusätzlich wird Überblick über das System verschafft.

In **Kapitel 3** wird das Verfahren zur Fahrbahnverfolgung beschrieben. Dieses ermittelt kontinuierlich ein Zielpunkt bezüglich des Fahrspurerkennungssystems, berechnet den passenden Lenkwinkel und korrigiert die Fahrtrichtung des Fahrzeuges.

In **Kapitel 4** folgt eine Auswertung des Verfahrens. Zunächst wird die bestgeeignete Zielentfernung für die Fahrbahnverfolgung empirisch ermittelt. Anschließend wird die Stabilität der Fahrbahnverfolgung mittels der Algorithmen Pure Pursuit und Follow-the-carrot verglichen.

Kapitel 5 fasst die Inhalte dieser Arbeit zusammen und in **Kapitel 6** sind weitere Entwicklungsschritte zur Optimierung des Verfahrens vorgestellt.

2 Fahrzeugquerführung

2.1 Grundlage der Fahrzeugquerführung

Die Fahrzeuglängs- und Querführung eines Fahrzeuges sind die wesentlichen Bestandteile von Systemen zur automatischen Spurführung. Die Längsführung regelt die Fahrzeuggeschwindigkeit in Bezug auf den Abstand zum vorausfahrenden Fahrzeug.

Die Querführung hat die Aufgabe das Fahrzeug durch geeignete Ansteuerung des Lenkaktuators stabil in der Fahrspur zu halten, d.h. mit dem Schwerpunkt die Mitte der Fahrspur zu verfolgen. Die Querführung kann durch eine in Fahrtrichtung ausgerichtete Kamera, ein Steuerungssystem zum Spurhalten und ein Lenkungssystem realisiert werden. Grundlage der Fahrzeugquerführung bildet eine kontinuierliche Bestimmung der Fahrspurgeometrie (Fahrspurbreite, horizontale Fahrspurkrümmung, horizontale Krümmungsänderung) sowie der eigenen Position und der eigenen Fahrzeugausrichtung bezüglich des Fahrstreifens über ein automatisches Kamerasystem [Rolf Isermann, 2006].

Um eine Querführung zu leisten, richtet die Lenkung das Fahrzeug in der gewünschten Fahrtrichtung aus. Die drei wichtigsten Anforderungen dabei sind:

1. eine präzise und schnelle Umsetzung des Lenkwinkels
2. gute Rückstellung in einen stabilen Geradeauslauf
3. Antriebs-, Brems- und Beschleunigungskräfte dürfen keinen Einfluss haben

Die Querführung des Fahrzeuges erfolgt durch eine Vorderachslenkung. Um eine optimale Kurvenfahrt zu erreichen, muss das Fahrzeug die Ackermann-Bedingung erfüllen. Die Ackermann-Bedingung besagt:

$$\cot\delta_o - \cot\delta_i = \frac{w}{l}$$

wobei δ_i und δ_o die Lenkwinkel des inneren bzw. äußeren Vorderrad entsprechen (vgl. Abbildung 3) [Reza N. Jazar, 2005]. Der Unterschied dieser Lenkwinkel wird als Winkeldifferenz oder als Voreilwinkel bezeichnet. Die Länge der Spurstange l heißt auch Radstand und der Abstand zwischen den Lenkungsachsen w wird Abstand der Drehpunkte der gelenkten Räder genannt. Aus der Ackermann-Bedingung folgt, dass die Verlängerungen aller Achsen bei einer Kurvenfahrt sich im Kreismittelpunkt treffen müssen. Die Hinterräder erfüllen die Bedingung stets, weil deren Lenkwinkel konstant gleich null ist (Vorderachslenkung). Die Vorder-

räder werden in Bezug auf die Fahrspurgeometrie eingeschlagen. Damit die Ackermann-Bedingung weiter erfüllt bleibt, wird das innere Rad stärker als das äußere eingeschlagen.

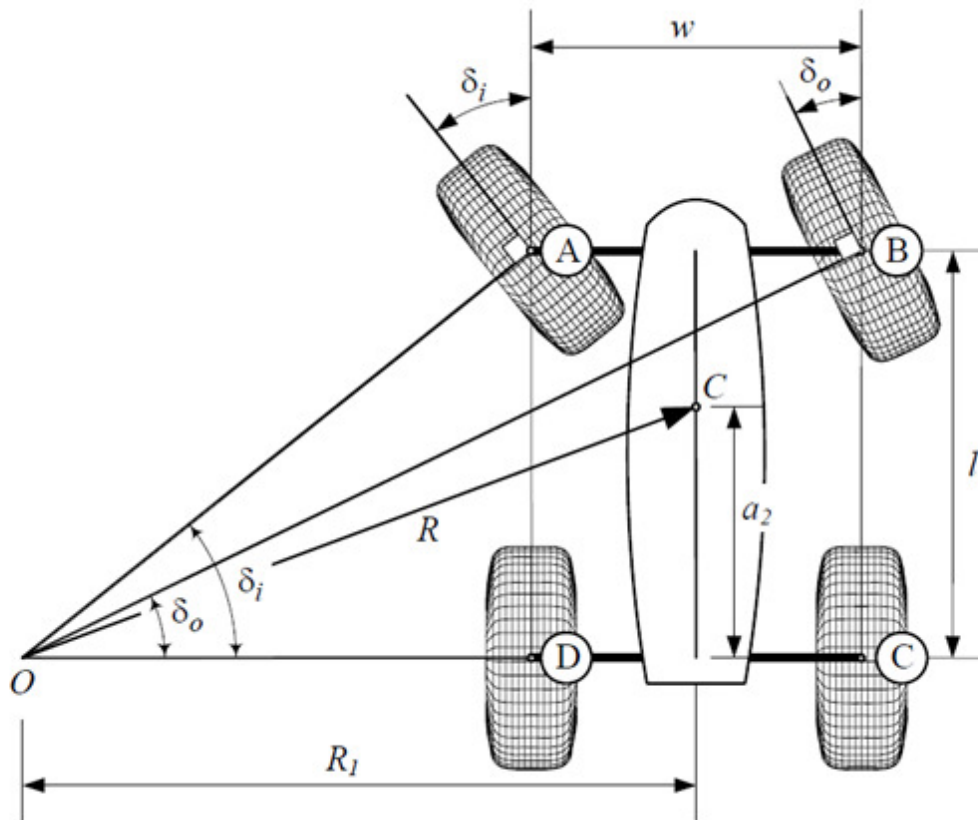


Abbildung 3: Fahrzeug mit Vorderachslenkung und Lenkwinkel der inneren und äußeren Vorderräder [Reza Jazar, 2005]

Das Massenzentrum eines gelenkten Fahrzeugs wird dann eine Kurvenfahrt mit dem Radius R leisten:

$$R = \sqrt{a_2^2 + l^2 \cot^2 \delta}$$

Wobei δ ist der Kotangens-Durchschnitt der inneren und äußeren Lenkwinkel.

$$\cot \delta = \frac{\cot \delta_o + \cot \delta_i}{2}$$

Nach einer Vereinfachung der Formel kann die Relation zwischen dem Lenkwinkel und der Radius der Kurve ermittelt werden:

$$\delta = \operatorname{atan}\left(\sqrt{\frac{l^2}{R^2 - a^2}}\right)$$

Der Abstand a ist relativ klein und kann somit vernachlässigt werden. Nach Ersetzung des Radius R mit seinem Kehrwert (die Krümmung γ), ergibt sich der folgende Zusammenhang:

$$\delta = \operatorname{atan}(l\gamma)$$

Daraus folgt, dass der Lenkwinkel δ eingeschlagen werden muss, damit ein Fahrzeug mit dem Radstand l eine Kurvenfahrt der Krümmung γ realisiert.

2.2 Systemüberblick der Fahrzeugquerführung

Der funktionale Systemaufbau der Fahrzeugquerführung ist in Abbildung 3 dargestellt. Hier ist die Signalverarbeitungskette der Fahrzeugquerführung zu verfolgen. Hardware und Software Komponenten sind in grau bzw. weiß abgebildet.

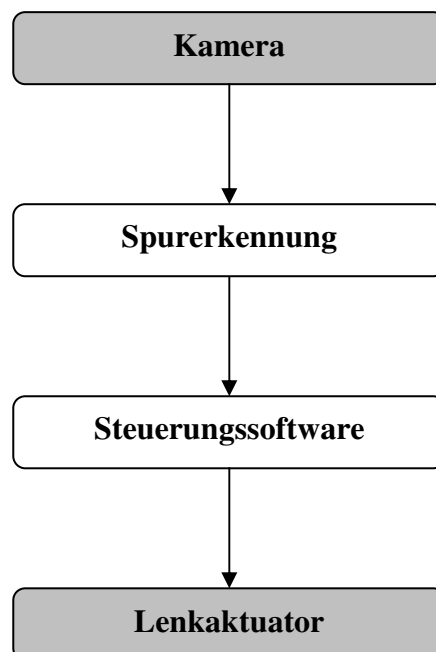


Abbildung 4: Systemkomponenten der Fahrzeugquerführung

Die Fahrzeugquerführung ist durch folgende Systemkomponenten realisiert:

- Kamera
- Spurerkennungsalgorithmus
- Steuerungssoftware
- Lenkaktuator

Das System zur Fahrspurerkennung soll in der Lage sein, mit Hilfe einer Kamera das Modellauto zügig durch einen unbekanntem Parcours fahren zu lassen gemäß dem Regelwerk des Carolo-Cup [Carolo-Cup-Regelwerk 2009]. Hier wurde eine Monochrom-USB-Kamera UI-1226LE-M aus der Ueye-Serie der Firma IDS-Imaging verwendet. Mit einer Größe von 44mm x 44mm x 20mm und einem Gewicht von 12g ist sie äußerst handlich und leicht zu montieren. Sie liefert bis zu 87 Bilder pro Sekunde und die Belichtungszeit kann bis auf 80 μ s reduziert werden, so dass auch für bewegliche Objekte ausreichend scharfe Fotos entstehen. Der Chip basiert auf der CMOS-Technologie.

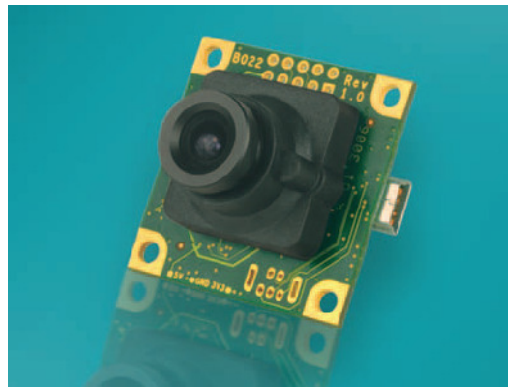


Abbildung 5: Monochrom-USB-Kamera UI-1226LE-M

Der Spurerkennungsalgorithmus Polaris hat die Aufgabe die entstandenen Bilder zu bearbeiten und die Fahrspurgeometrie zu bestimmen [Eike Jennings, 2009]. Um eine zentimetergenaue Positionsbestimmung der Fahrspurmarkierung zu realisieren wird ein Polynom 2. oder 3. Grades eingesetzt. Im Bild werden die Fahrspurmarkierungen durch einzeilige Betrachtungsbereiche diskretisiert. In jedem Betrachtungsbereich wird das Bild mit dem Sobel-Operator gefiltert und anschließend maximal ein Fahrspurmarkierungspunkt extrahiert. Die Koordinaten der extrahierten Bildpunkte werden durch eine Linsenverzeichnungskorrektur bearbeitet, sodass der Abbildungsfehler des Weitwinkelobjektivs reduziert wird. Die Korrektur erfordert die Kalibrierung der internen Kameraparameter. Durch eine projektive Transformation werden die verzeichnungsfreien Bildpunkte in das Fahrzeugkoordinatensystem übertragen. Um die Parameter des Polynoms zu approximieren wird die Methode der kleinsten Quadrate angewendet. Auf diese Weise liefert der Algorithmus Polaris zwei Polynome, die die zwei zur Fahrspur gehörigen Fahrbahnmarkierungen im Fahrzeugkoordinatensystem beschreiben.



Abbildung 6: Fahrspurerkennung durch Polaris

Ziel der Steuerungssoftware ist den Lenkwinkel in Bezug auf diese Information festzulegen und die Fahrtrichtung des Fahrzeuges zu korrigieren. Diese Systemkomponente wird in dieser Arbeit ausführlich beschrieben. Die Anforderungen der Steuerung werden von einem Lenkaktuator umgesetzt. Die Lenkung des Fahrzeuges Nebula erfolgt durch ein Standart-Servo mittlerer Baugröße.

3 Verfahren zur Fahrbahnverfolgung basiert auf den Spurerkennungsalgorithmus Polaris

Um einen unbekanntem Parcours zu bewältigen, muss das Fahrzeug zuerst die Fahrspur erkennen. Der Algorithmus Polaris liefert zwei Polynome 2. Grades, die die zwei Fahrspurmarkierungen im Fahrzeugkoordinatensystem approximieren (vgl. Abschnitt 2.2). Um eine Fahrbahnverfolgung zu realisieren, muss als nächstes mit Hilfe der zwei Polynome den Lenkwinkel ermittelt werden.

Das Verfahren zur Fahrbahnverfolgung beruht teilweise auf dem Verhaltensmuster des Menschen beim Fahren. Der Fahrer fokussiert sein Blickfeld in einer gewissen Entfernung vor dem Fahrzeug und dreht das Lenkrad relativ zu der Fahrspur. Analog dazu lokalisiert das Verfahren kontinuierlich ein Ziel, das das Fahrzeug verfolgen soll. Danach wird der Lenkwinkel in Bezug auf das Ziel mittels der Algorithmen „Follow-the-carrot“ oder „Pure Pursuit“ berechnet [R. Craig Coulter, 1992], [Martin Lundgren, 2003]. Schließlich wird die Richtung des Fahrzeuges korrigiert.

Das Verfahren lässt sich in drei Phasen aufteilen:

- Festlegung des Zieles
- Bestimmung des gewünschten Lenkwinkels in Bezug auf das Ziel
- Lenkwinkelsteuerung

3.1 Festlegung des Zieles

Das Ziel ist ein Punkt im Fahrzeugkoordinatensystem und liegt idealerweise in der Mitte der Fahrspur. Diese vorteilhafte Lateralposition für das Fahrzeug schafft Robustheit für die Querführung. Andernfalls würde das Fahrzeug schneller die Fahrspur verlassen bei ungünstigen äußeren Umständen z.B. fehlende oder verschmutzte Spurmarkierungen, eingeschränkte Sichtverhältnisse, ungünstige Lichtverhältnisse.

3.1.1 Zielentfernung zum Fahrzeug

Die Entfernung zwischen Fahrzeug und Ziel spielt eine signifikante Rolle im autonomen Fahren. Kleine Änderungen der Zielentfernung können die Stabilität der Fahrbahnverfolgung stark beeinflussen. Die Wahl der bestmöglichen Entfernung ist bedingungsabhängig. Die Schwierigkeit besteht darin, dass höhere Präzision der Fahrt nur auf Kosten der Stabilität er-

reicht werden kann. Wenn das Fahrzeug möglichst genau in der Mitte der Fahrbahn fahren soll, muss die Entfernung relativ klein sein, damit die Fahrspurgeometrie optimaler berücksichtigt wird. Das wiederum beeinträchtigt die Stabilität so erheblich, dass höhere Geschwindigkeiten nicht erreichbar sind, weil die Verzögerung des Lenkungssystems ausschlaggebende Schwingungen beim Fahren verursacht. Um die Bedeutung dieser Verzögerung zu verringern, kann die Entfernung erhöht werden. Auf diese Weise werden die Anzahl der Schwingungen reduziert und somit wird bessere Stabilität erreicht, die die Fahrbahnverfolgung bei höheren Geschwindigkeiten ermöglicht. Das verschlechtert gleichzeitig die Präzision der Fahrt und ab einer bestimmten Entfernung kann der Algorithmus nicht mehr gewährleisten, dass das Fahrzeug dauerhaft auf der zu verfolgenden Fahrbahn bleibt. Der ausschlaggebende Faktor für die größte mögliche Entfernung ist der minimale Kurvenradius, weil die Fahrt in den engsten Kurven der Strecke am meisten Präzision fordert. Da sowohl Präzision, als auch Stabilität sehr wichtig sind, muss eine passende Entfernung hinsichtlich der gestellten Anforderungen bestimmt werden. Die bestgeeignete Zielentfernung ist bezüglich des minimalen Kurvenradius bei Carolo-Cup empirisch ermittelt worden (vgl. Abschnitt 4.3.1).

3.1.2 Bestimmung des Zieles

Der Spurerkennungsalgorithmus Polaris liefert zwei Polynome 2. Grades, die die zwei zur Fahrspur gehörigen Fahrbahnmarkierungen im Fahrzeugkoordinatensystem beschreiben (vgl. Abschnitt 2.2). Der Ursprung des Fahrzeugkoordinatensystems wird von Polaris direkt mittig vor dem Fahrzeug platziert [Eike Jenning, 2009]. Dennoch ist für die Steuerungssoftware ein Fahrzeugkoordinatensystem mit Ursprung im Massenzentrum des Fahrzeuges besser geeignet. Deswegen verschiebt die Software den Ursprung des Fahrzeugkoordinatensystems, indem die Hälfte der Fahrzeuglänge zur x-Koordinate des Zieles addiert wird.

Die x-Werte von allen Fahrspurmarkierungspunkten sind immer positiv, weil die Kamera immer die Fahrspurmarkierungen vor dem Fahrzeug wahrnimmt. Außerdem ist es zu beachten, dass Polaris allen extrahierten Fahrspurmarkierungspunkte, die sich rechts von der Fahrzeuglängsachse befinden, einem positiven y-Wert zuordnet. Analog dazu haben die linksstehenden Punkte einen negativen y-Wert. Somit approximiert das Polynom p1 auf Abbildung 7 die rechte Fahrspurmarkierung, p2- die linke.

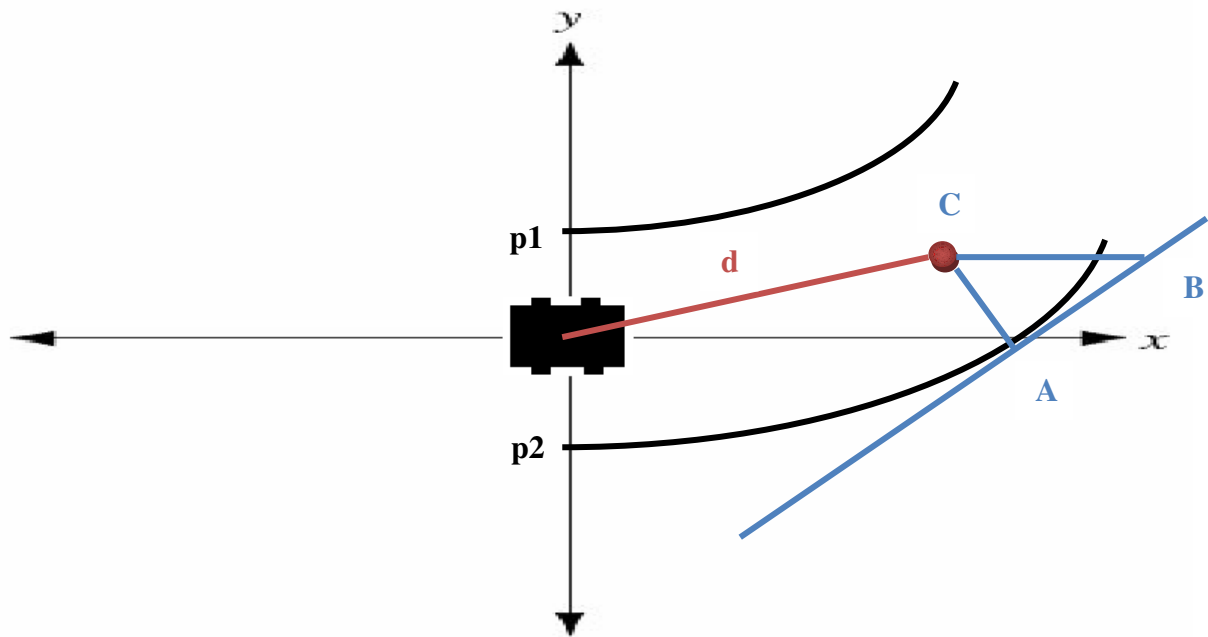


Abbildung 7: Bestimmung des Zieles

Bei Carolo-Cup ist die Fahrbahn konstant 820 mm breit und an den Rändern mit durchgezogenen Linien abgegrenzt. Die zwei Fahrspuren werden durch eine 20 mm breite, gestrichelte Mittellinie geteilt [Carolo-Cup-Regelwerk 2009]. Es hat sich bei den Testfahrten ergeben, dass bei der Spurerkennung die durchgezogene Linie präziser als die gestrichelte Mittellinie approximiert wird. Deswegen war es vorteilhafter die Querführung in Bezug auf die durchgezogene Fahrspurmarkierung zu realisieren, d.h. wenn das Fahrzeug sich auf der rechten Fahrspur befindet, wird das Ziel in Abhängigkeit von der rechten Fahrspurmarkierung ermittelt. Analog dazu ist die linke Fahrspurmarkierung bei Fahrten in der linken Fahrspur bestimmend.

Das zu verfolgende Ziel soll in der Mitte der zwei zur Fahrspur gehörigen Fahrbahnmarkierungen liegen. Um die Koordinaten der Fahrspurmitte zu bestimmen, wird die Tangente des entsprechenden Polynoms in einem Punkt A in gewisser Entfernung von dem Fahrzeug gebildet. Die x-Koordinate dieses Punktes wird mit dem Bisektionsverfahren festgelegt, das später in diesem Abschnitt genauer beschrieben wird. Da der Punkt A auf dem Polynom liegt, kann der Wert der x-Koordinate in der Gleichung des Polynoms eingesetzt werden, um den y-Wert des Punktes zu bestimmen.

$$f(x) = a_2 * x^2 + a_1 * x + a_0$$

Mittels der Tangentengleichung wird der Punkt B, der an der Tangente liegt, festgelegt. Die Fahrspurmitte entspricht Punkt C (der Durchschnitt von AC und BC), wobei AC orthogonal zur Tangente AB ist und die Hälfte der Fahrbahnbreite beträgt (Spurbreite bei Carolo-Cup ist 40 cm). Hier wird die folgende Form für die Geradengleichungen eingesetzt:

$$y = m * x + b$$

Durch die Geradengleichungen von AC und BC kann deren Durchschnitt ermittelt werden:

1. $C_y = m_{AC} * C_x + b_{AC}$
2. $C_y = m_{BC} * C_x + b_{BC}$

Aus Gleichungen 1 und 2 sind die Koordinate der Fahrspurmitte wie folgt zu bestimmen:

$$C_x = \frac{b_{AC} - b_{BC}}{m_{BC} - m_{AC}}$$
$$C_y = m_{AC} * C_x + b_{AC}$$

Wenn die Steigungen der Geraden bekannt sind, können die y-Achsenabschnitte berechnet werden:

$$b_{AC} = A_y - m_{AC} * A_x$$
$$b_{BC} = B_y - m_{BC} * B_x$$

Die Steigung von AB und AC sind:

$$m_{AB} = \frac{B_y - A_y}{B_x - A_x}$$
$$m_{AC} = -\frac{1}{m_{AB}}$$

(da AC orthogonal zu AB ist)

Um die Steigung von BC zu berechnen kommt die Gleichung für Winkel α zwischen die Geraden AB und BC zum Einsatz:

$$\tan\alpha = \frac{m_2 - m_1}{1 + m_1 m_2}$$

(wobei $\tan\alpha = AC/AB$ und $m_2 > m_1$)

An dieser Stelle werden zwei Fälle für die Steigung von BC unterschieden:

Fall 1: Die Querführung wird in Bezug auf die linke Fahrspurmarkierung realisiert (vgl. Abbildung 7). In diesem Fall ist m_{AB} stets größer als m_{BC} . Daraus folgt:

$$m_{BC} = \frac{m_{AB} - \tan\alpha}{1 + m_{AB} * \tan\alpha}$$

Fall 2: Die Querführung wird in Bezug auf die rechte Fahrspurmarkierung realisiert. In diesem Fall ist m_{AB} stets kleiner als m_{BC} . Daraus folgt:

$$m_{BC} = \frac{m_{AB} + \tan\alpha}{1 - m_{AB} * \tan\alpha}$$

Somit sind die Steigungen und die y-Achsenabschnitte der Geraden AC und BC ermittelt und die Fahrspurmitte kann festgelegt werden.

Weiter soll das Ziel in einer bestimmten Entfernung vor dem Fahrzeug liegen. Um die gewünschte Entfernung d zu erzielen, muss ein geeigneter Punkt A bestimmt werden, in dem die Tangente gebildet wird. Hier kommt das Bisektionsverfahren zum Einsatz. Durch Intervallhalbierung der x-Koordinate vom Punkt A wird ein passendes Ziel in einigen Bisektionsschritten ermittelt (vgl. Anhang C).

Vor Ablaufbeschreibung der Zielbestimmung, ist es wichtig zwei Anmerkungen zu machen. Die erste ist, dass ein Bisektionsschritt als ein Durchlauf von den Schritten 2 bis 6 definiert ist. Weiterhin soll die x-Koordinate von A im Intervall $[0 - \text{MAX_LOOK_AHEAD_DISTANCE}]$ liegen, wobei $\text{MAX_LOOK_AHEAD_DISTANCE}$ dem maximalen Wert der x-Koordinate vom Punkt A entspricht.

Der Prozess der Zielbestimmung ist durch die folgende 6 Schritte realisiert:

1. Anzahl der erforderlichen Bisektionsschritten wird ermittelt.
2. Die Mitte des Intervalls wird der x-Koordinate vom A zugewiesen.
3. Die Tangente des entsprechenden Polynoms wird am Punkt A gebildet und das Ziel wird festgelegt.
4. Die Entfernung d wird berechnet.
5. Falls d kleiner als gewünscht ist, wird die Mitte des Intervalls als linke Grenze für das Intervall weiter verwendet.
Falls d größer als gewünscht ist, wird die Mitte des Intervalls als rechte Grenze für das Intervall weiter verwendet.
6. Falls Anzahl der erforderlichen Bisektionsschritten noch nicht erreicht ist, weiter zum Schritt 2.
Sonst ist der Prozess der Zielbestimmung abgeschlossen.

3.2 Path Tracking Algorithmen

Die nächste Herausforderung ist den Lenkwinkel in Bezug auf das Ziel zu bestimmen. An dieser Stelle wurden zwei Path Tracking Algorithmen eingesetzt. Follow-the-carrot und Pure Pursuit sind traditionelle, geometrische Algorithmen, mit der Aufgabe einen gewünschten Lenkwinkel zu berechnen damit ein Fahrzeug eine gewünschte Position auf der Fahrbahn erreicht [R. Craig Coulter, 1992], [Martin Lundgren, 2003].

3.2.1 Follow-the-carrot

Das Konzept von Follow-the-carrot ist ganz einfach. Um den gewünschten Lenkwinkel δ für das Lenkungssystem zu berechnen, wird der Winkel zwischen Fahrzeugausrichtung und Ziel eingesetzt (vgl. Abbildung 8).

$$\delta = \text{atan}\left(\frac{y}{x}\right)$$

Follow-the-carrot ermittelt kontinuierlich den Lenkwinkel relativ zu dem aktuellsten Ziel. Dadurch wird immer wieder die Richtung des Fahrzeuges korrigiert und somit wird eine Fahrbahnverfolgung realisiert. Ein Proportionalglied kann eingesetzt werden, um die Stabilität des Algorithmus zu verbessern (vgl. Anhang B).

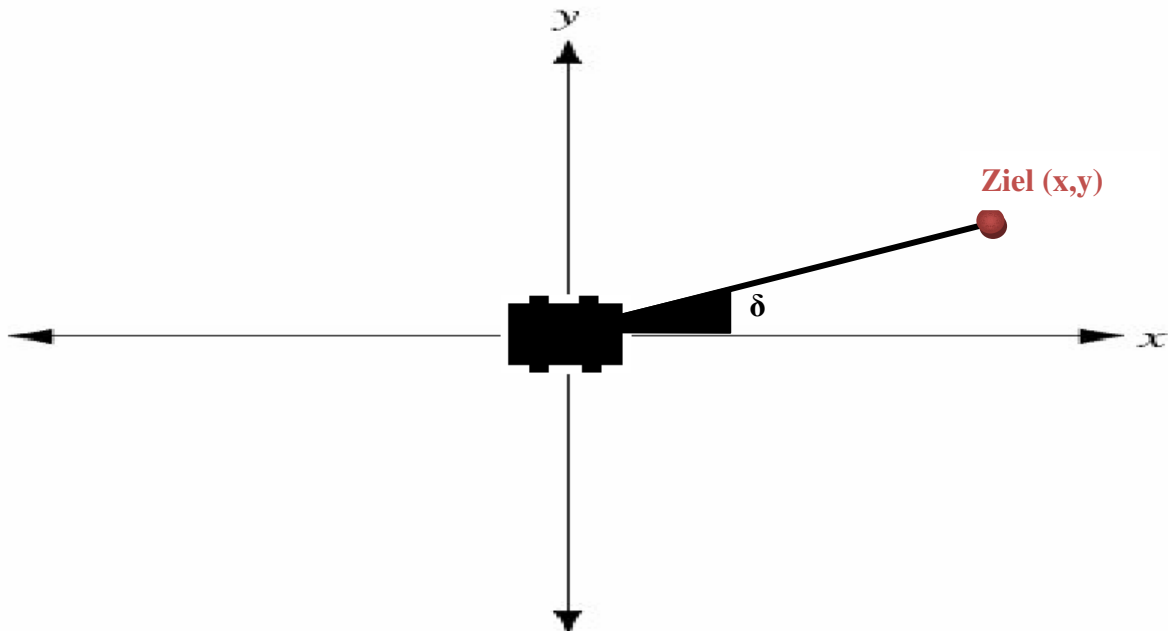


Abbildung 8: Follow-the-carrot

3.2.2 Pure Pursuit

Pure Pursuit ist ein Path Tracking Algorithmus [R. Craig Coulter, 1992]. Der Algorithmus basiert auf der Krümmung des Kreisbogens, der Fahrzeug und Ziel verbindet. Das Fahrzeug reagiert kontinuierlich mit einer Kurvenfahrt dieser Krümmung, um das Ziel zu erreichen.

Ursprünglich wird der Bogen, der Fahrzeug und Ziel verbindet, gebildet. Pure Pursuit berechnet die Krümmung des Bogens. Diese Krümmung dient als Führungsgröße für die Steuerung der Lenkaktorik.

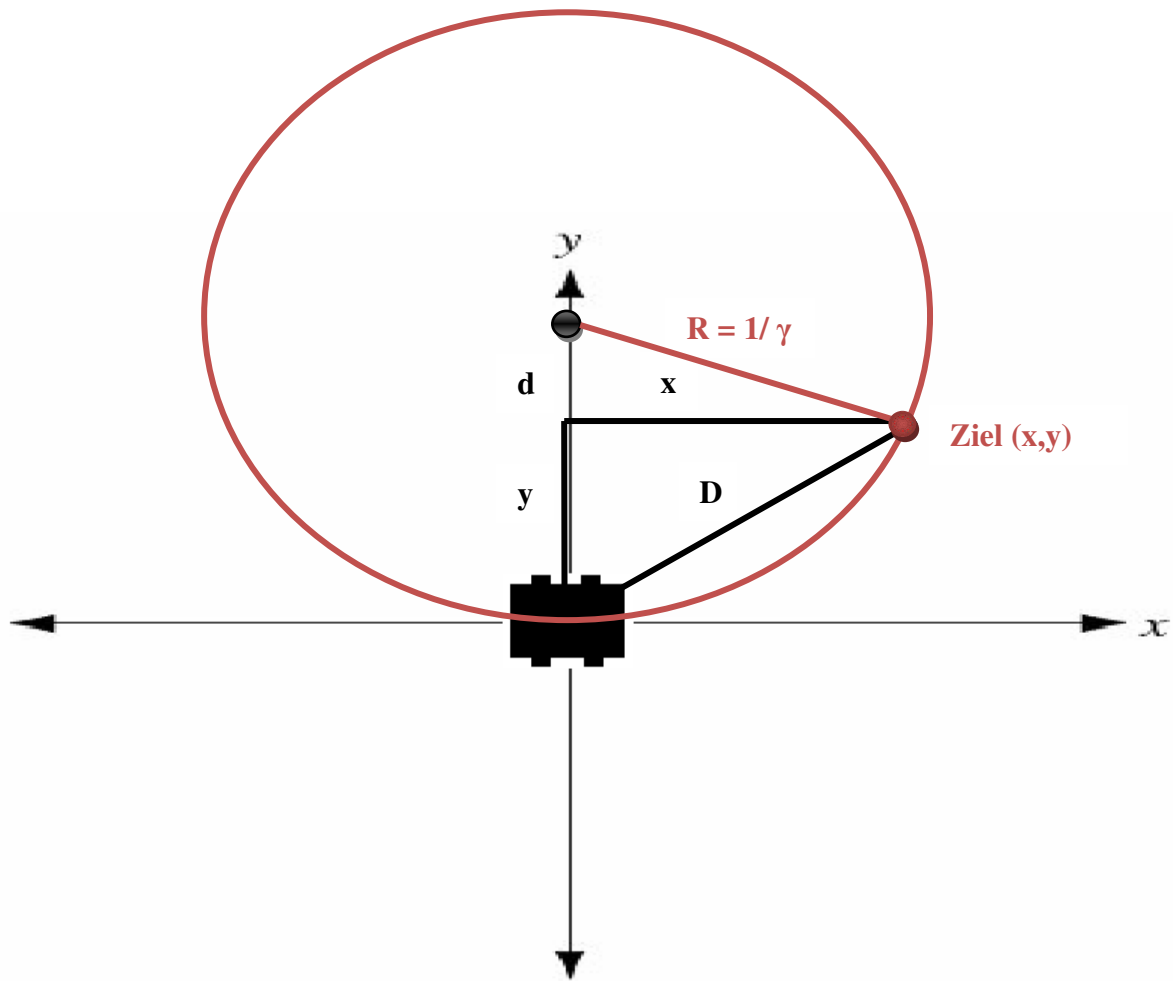


Abbildung 9: Pure Pursuit

Nachdem das Ziel im Fahrzeugkoordinatensystem bestimmt worden ist, kann die Krümmung ermittelt werden. Es erfolgt anhand der zwei Gleichungen:

1. $x^2 + y^2 = D^2$

2. $y + d = r$

- Durch die erste Gleichung wird die Zielentfernung gemäß Satz des Pythagoras berechnet.
- Die zweite Gleichung beschreibt die Relation zwischen Radius und y-Offset des Zieles

Daraus folgt:

$$d = r - y$$

$$(r - y)^2 + x^2 = r^2$$

$$\begin{aligned}
r^2 - 2ry + y^2 + x^2 &= r^2 \\
x^2 + y^2 &= 2ry \\
2ry &= D^2 \\
r &= D^2/2y
\end{aligned}$$

Der Kehrwert der Krümmung ist der Radius des Kreises. Daraus ergibt sich die folgende Gleichung für die Krümmung:

$$\gamma = 2y/D^2$$

Um eine optimale Kurvenfahrt dieser Krümmung zu erreichen wird der entsprechende Lenkwinkel ($\delta = \text{atan}(l\gamma)$) eingeschlagen (vgl. Abschnitt 2.1).

3.3 Lenkwinkelsteuerung

Eine Steuerung wird meistens eingesetzt in gut definierten Systemen, wo ein mathematisches Verhältnis zwischen Eingang- und Ausganggröße existiert. Nachteil der Steuerung ist, dass die Störungen der Aktoren nicht korrigierbar sind. Um die Anforderungen der Steuerung präziser umzusetzen ist häufig eine fortlaufende Rückkopplung der Ausgangsgröße vorhanden. In diesem Fall spricht man von einer Regelung (vgl. Anhang A). Um die Anforderungen an dem Lenkungssystem sehr präzise umzusetzen, ist ein PID Regler vorteilhaft. Die Eigenschaften und die Optimierung eines PID Reglers sind im Anhang dieser Arbeit beschrieben (vgl. Anhang B). Zu diesem Zweck sind Sensoren, die den Lenkwinkel der Vorderräder messen erforderlich. Da das Fahrzeug Nebula über keine solchen Sensoren verfügt, ist bei diesem System eine Regelung nicht umsetzbar. Aus diesem Grund wurde die automatische Querführung mittels einer Steuerung des Lenkaktuators realisiert.

Nach der Festlegung des Lenkwinkels, soll die Lenkaktorik gesteuert werden. Der OSEK-Mikrokontroller stellt eine Schnittstelle zwischen dem Lenkservo und dem Nano-PC bereit. Dadurch lässt sich das Lenkservo im Wertebereich von -100 (voller Lenkeinschlag nach links) bis 100 (voller Lenkeinschlag nach rechts) steuern. Um eine Lenkwinkelsteuerung zu realisieren ist es erforderlich ein mathematisches Verhältnis zwischen einer lenkungsrelevanten Größe (z.B. Radius, Krümmung der Kurve, Lenkwinkel) und dem Stellwert e zu bestimmen. Der Lenkwinkel in Grad war in diesem Fall als Messgröße am besten geeignet aufgrund des annähernd linearen Verhältnisses zum Stellwert. Die Messergebnisse der Vorderräderlenkwinkel bei den entsprechenden Stellwerten sind in Tabelle 1 eingetragen. Daraus ergibt sich das folgende mathematische Verhältnis für die Steuerung:

$$e = 5\delta - 20$$

Stellwert e	Linkes Rad Lenkwinkel ($^{\circ}$)	Rechtes Rad Lenkwinkel ($^{\circ}$)	Resultierender Lenkwinkel $\delta(^{\circ})$
100	20	24	22
80	18	20	19
60	16	16	16
40	11	11	11
20	9	10	9
0	4	6	5
-20	0	0	0
-40	4	3	3
-60	7	7	7
-80	11	11	11
-100	18	15	16

Tabelle 1: Die Messergebnisse der Vorderräder-Lenkwinkel bei den entsprechenden Stellwerten

Die Antwort auf die Frage wie hoch die Genauigkeit dieser Approximation sein soll ist systemabhängig. Da ein großes Spiel des Lenkungssystems vorhanden ist, muss das mathematische Verhältnis nicht sehr präzise sein. Es führt aber natürlich zu Störungen des autonomen Fahrens, besonders bei höheren Geschwindigkeiten.

4 Test

4.1 Teststrategie

Um eine optimale Querführung zu erreichen, wurden mehrere Testfahrten auf drei Strecken bei verschiedenen Geschwindigkeiten ausgeführt. Die erste Strecke ist eine Kombination von zwei Kurven mit einem Radius von 100 cm (der minimale Innenradius einer Kurve beträgt 1000 mm – gemäß dem Regelwerk des Carolo-Cup). Eine weitere Herausforderung für das Fahrzeug ist eine stabile Fahrt im Kreis zu leisten. Die dritte Strecke ist 8 förmig und beinhaltet aufeinander folgende linke und rechte Kurven. Bei der Testdurchführung wurden 50 Bilder pro Sekunde von der Spurerkennung ausgewertet und die Belichtungszeit der Kamera wurde auf 17ms reduziert, so dass während der Fahrt ausreichend scharfe Fotos entstehen.

Wie im Kapitel 3 bereits erläutert wurde, die Entfernung vom Fahrzeug zum Ziel spielt eine signifikante Rolle in der autonomen Fahrbahnverfolgung. Deswegen wurden zuerst die am besten geeigneten Zielentfernungen für die Algorithmen Follow-the-carrot und Pure Pursuit bestimmt.

Das nächste Ziel war die Stabilität der Fahrbahnverfolgung-Algorithmen zu vergleichen. Um eine Stabilität der Fahrbahnverfolgung zu erzielen, soll ein gewünschtes Verhalten möglichst genau reproduziert werden. Bei geraden Strecken sollte die Lenkung in einen stabilen Geradeauslauf eingestellt werden. Die größte Herausforderung der Fahrbahnverfolgung sind die Kurven. Um eine stabile Kurvenfahrt zu leisten, sollte den gewünschten Lenkwinkel eingeschlagen und dann gehalten werden. Am Ende der Kurve sollte die Lenkung zurückgestellt werden. Folglich während einer optimalen Fahrt muss der Lenkwinkel nur bei Betreten oder beim Verlassen einer Kurve korrigiert werden. Somit je kleiner die Lenkwinkeländerung, desto stabiler die Fahrt. Aus diesem Grund wird die Stabilität der Algorithmen hinsichtlich der durchschnittlichen Lenkwinkeländerung pro Zeiteinheit in einem Durchlauf der Teststrecken gemessen. Die bessere Stabilität der Algorithmen ermöglicht eine autonome Fahrt bei höherer Geschwindigkeit.

Anderer wichtiger Faktor der Fahrbahnverfolgung ist die Präzision der Querführung. Um diese zu ermitteln, muss die Lateralposition des Fahrzeuges auf der Fahrbahn analysiert werden. Dennoch verfügt das Fahrzeug über keine geeigneten Sensoren zu diesem Zweck.

4.2 Sensordatenanalyse

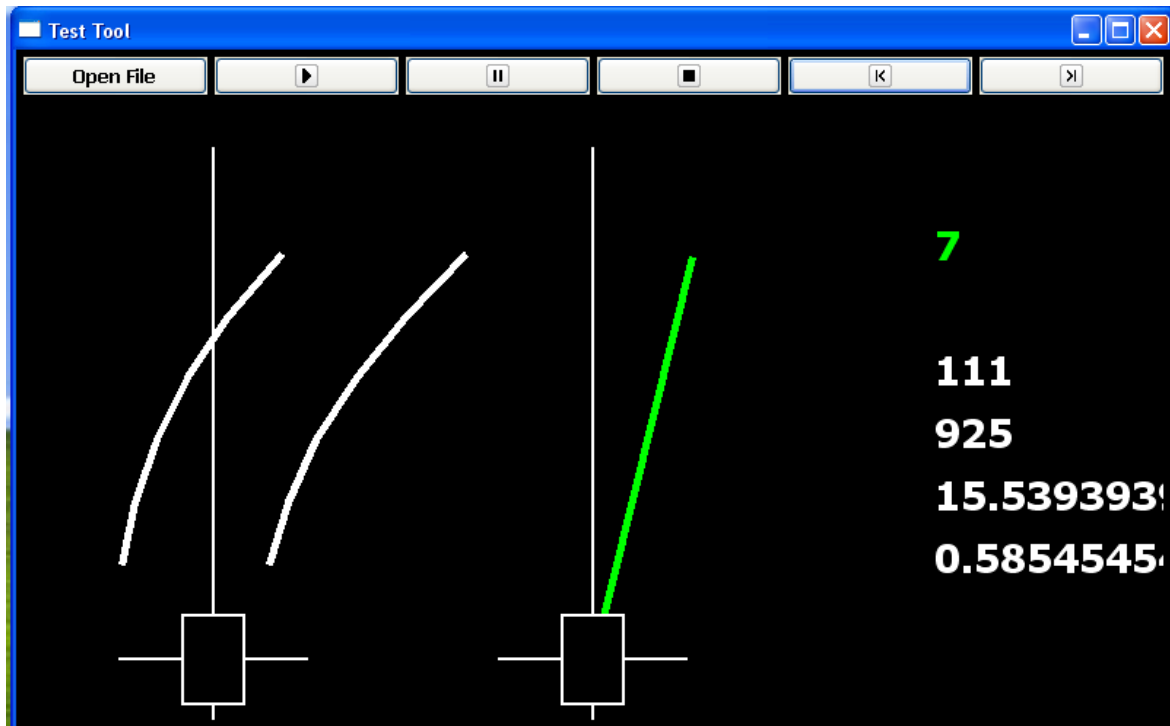


Abbildung 10: Test Tool zur Sensordatenanalyse

Die Architektur des FAUSTcore Softwaresystems ermöglicht während einer Fahrt das Speichern von Sensordaten in einer Logdatei. Um diese Information zu analysieren wurde eine Anwendung mit dem Standard Widget Toolkit (SWT) entwickelt. SWT ist eine grafische Benutzeroberfläche (GUI) Toolkit für die Programmiersprache Java [Steve Northover, Mike Wilson, 2004].

Das Test Tool liest die entsprechenden Daten (z.B. Lenkwinkel, Geschwindigkeit, Polynome der Spurerkennung usw.) aus der Logdatei, speichert sie in einem Datenkontainer und stellt sie graphisch dar. Somit entsteht die Möglichkeit Abläufe mehrere Testfahrten nachträglich zu verfolgen und auszuwerten. Bei der Stabilitätsanalyse der Fahrbahnverfolgung z.B. wurde mittels dieser Anwendung die durchschnittliche Lenkwinkeländerung in einer Runde der Teststrecken ermittelt.

4.3 Testergebnisse

4.3.1 Bestimmung der Zielentfernung

Bei Carolo-Cup in der dynamischen Disziplin Rundstrecke soll das Fahrzeug autonom drei Minuten lang auf einem unbekanntem Rundkurs so weit wie möglich fahren. Um eine stabile und somit schnelle Fahrbahnverfolgung zu erreichen, wurden möglichst große Zielentfernungen für die Algorithmen ausgewählt. Um den minimalen Kurvenradius von 100 cm zu berücksichtigen wurde die Zielentfernung bei einer Teststrecke getestet, die zwei Kurven mit diesem Radius beinhaltet (vgl. Abbildung 11).

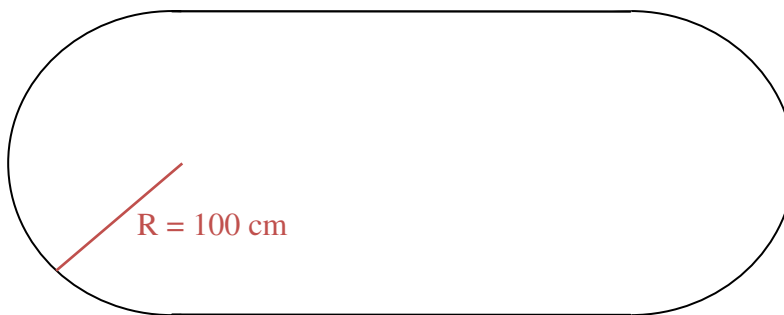


Abbildung 11: Spurgeometrie der Teststrecke 1

Die Stabilität wurde hinsichtlich der durchschnittlichen Lenkwinkeländerung pro Zeiteinheit in einem Durchlauf der Teststrecken ermittelt. Um die durchschnittliche Änderung zu berechnen, wird die Lenkwinkeländerung im 20 Millisekundentakt ermittelt und die Summe dieser Änderungen wird geteilt durch die Anzahl der Takte, die das Fahrzeug für einen Durchlauf der Teststrecke benötigt. In den folgenden Tabellen sind die Ergebnisse bei drei unterschiedlichen Geschwindigkeiten dargestellt. Die stabilste Fahrt auf dieser Teststrecke wurde mit 100% gekennzeichnet. Die Stabilität aller anderen Fahrten ist prozentual dazu dargestellt.

Zielentfernung[cm]	50	60	70	80	90	100
Follow-the-carrot	31	49	51	—	—	—
Pure Pursuit	41	69	83	94	96	100

Tabelle 2: Stabilität[%] der Fahrten, relativ zu der stabilsten Fahrt bei Geschwindigkeit von 80 cm/s

Entfernung[cm]	50	60	70	80	90	100
Follow-the-carrot	22	31	35	—	—	—
Pure Pursuit	21	44	53	54	59	67

Tabelle 3: Stabilität[%] der Fahrten, relativ zu der stabilsten Fahrt bei Geschwindigkeit von 120 cm/s

Entfernung[cm]	50	60	70	80	90	100
Follow-the-carrot	—	—	—	—	—	—
Pure Pursuit	—	36	40	42	42	43

Tabelle 4: Stabilität[%] der Fahrten, relativ zu der stabilsten Fahrt bei Geschwindigkeit von 160 cm/s

Die Ergebnisse zeigen, dass Änderungen der Zielentfernung viel mehr die Stabilität von Follow-the-carrot beeinflussen. Mit Pure Pursuit war eine Fahrbahnverfolgung mit Zielentfernung von 100 cm und Geschwindigkeit bis zu 160 cm/s möglich. Jedoch war die Fahrt nicht ausreichend präzise und an mehreren Stellen ist das Fahrzeug über die Fahrbahnmarkierung gefahren. Damit Präzision und gleichzeitig Stabilität gewährleistet werden, wird Pure Pursuit kontinuierlich den Lenkwinkel berechnen in Bezug auf ein Ziel, das 80 cm vom Fahrzeug entfernt ist. Analog dazu ist für Follow-the-carrot eine Zielentfernung von 60 cm am besten geeignet.

4.3.2 Pure Pursuit versus Follow-the-carrot, Teststrecke 1

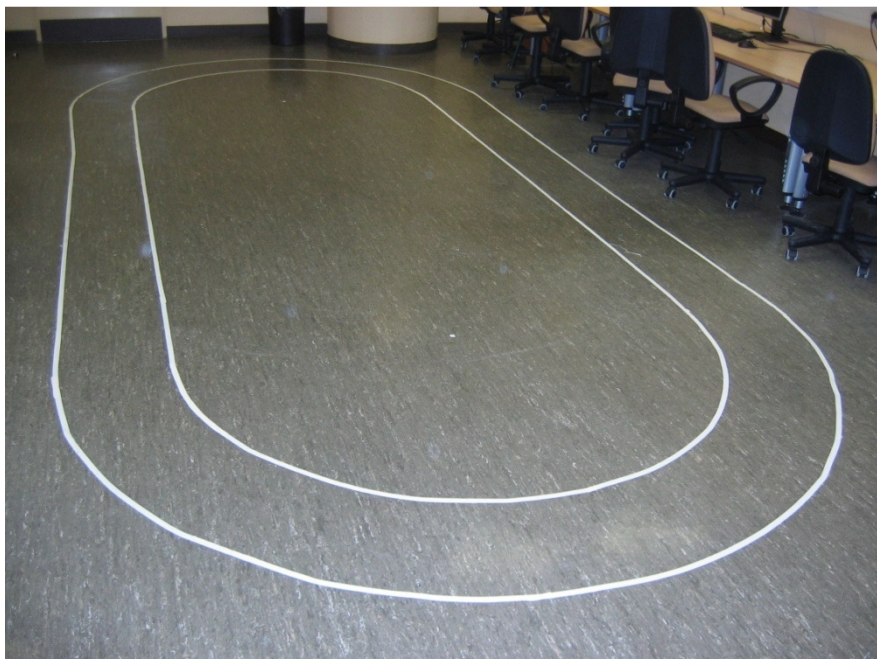


Abbildung 12: Teststrecke 1

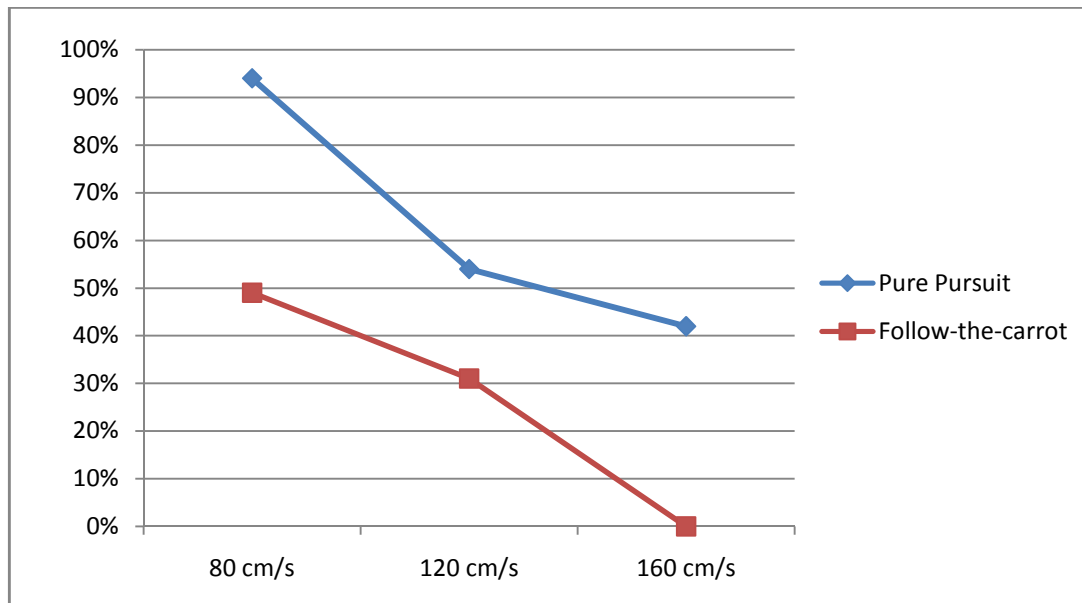


Abbildung 13: Stabilität[%] der Fahrten (Teststrecke 1) bei 80, 120 und 160 cm/s

Bei der ersten Teststrecke wurde die Stabilität der Fahrbahnverfolgung mit den Algorithmen Pure Pursuit und Follow-the-carrot bei drei unterschiedlichen Geschwindigkeiten gemessen. Von Abbildung 13 lässt sich leicht erkennen, dass Pure Pursuit deutlich bessere Stabilität als Follow-the-carrot leistet. Folglich ist auch eine schnellere Fahrbahnverfolgung mit diesem Algorithmus möglich. Außerdem zeigen die oben erfassten Ergebnisse, dass die Erhöhung der Geschwindigkeit immer mit einer Verschlechterung der Stabilität verbunden ist.

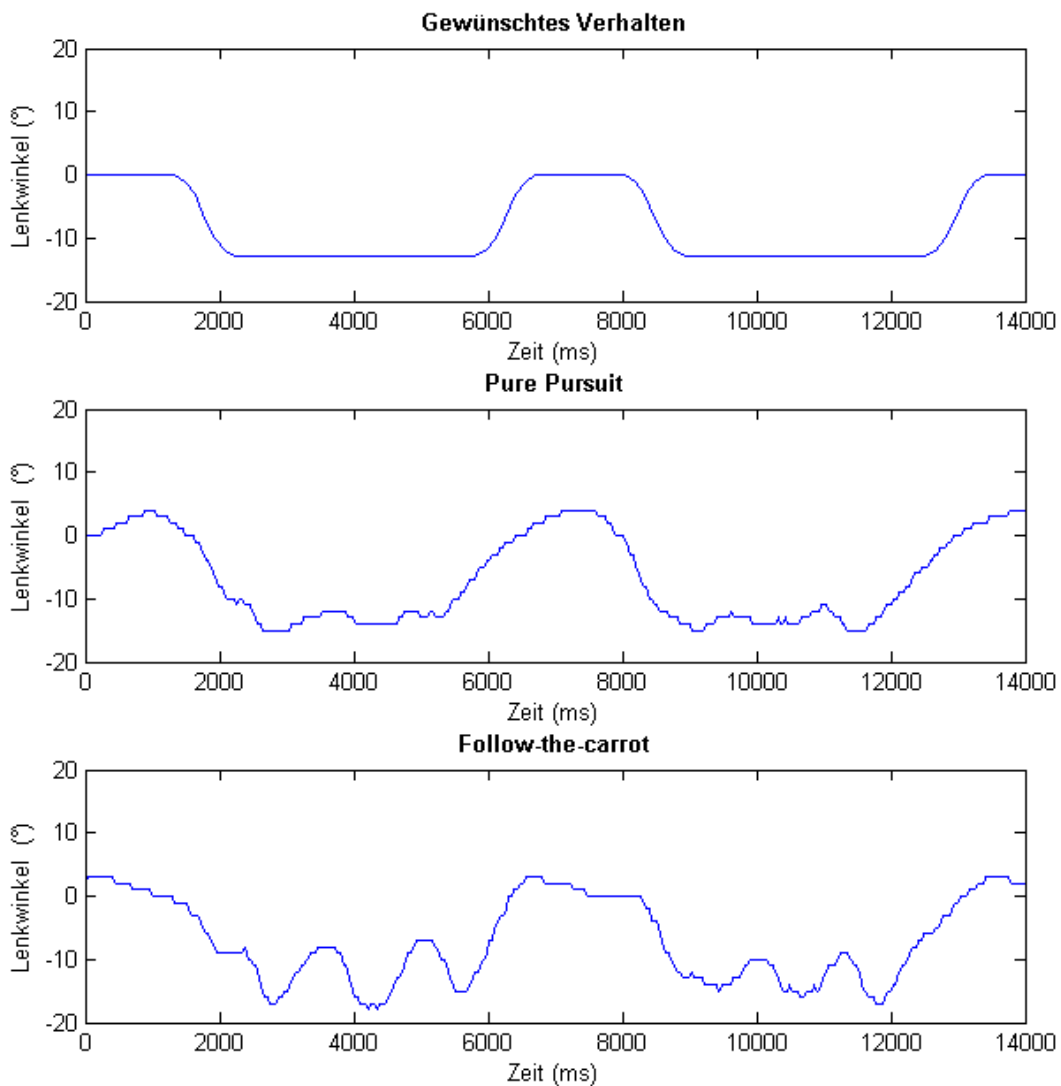


Abbildung 14: Vergleich der Lenkungswinkelverläufe von Pure Pursuit und Follow-the-carrot mit dem gewünschten Verhalten(Teststrecke 1)

Die Lenkungswinkelverläufe von gewünschtem Verhalten, Pure Pursuit und Follow-the-carrot in einer Runde der Teststrecke sind in Abbildung 14 veranschaulicht. Somit lässt sich die Stabilität der beiden Algorithmen graphisch vergleichen. Auf der y-Achse ist der Lenkwinkel in Grad zum entsprechenden Zeitpunkt auf der x-Achse eingetragen. Die negativen Lenkwinkelwerte entsprechen einen linken Ausschlag, die positive einen rechten. Der Lenkungswinkelverlauf kann bei der Teststrecke unter vier zeitlichen Phasen unterteilt werden:

1. (0 – 1500ms) Die Analyse startet immer in der geraden Strecke, folglich ist der Lenkwinkel im Idealfall gleich null.
2. (1500 – 2000ms) Bei Zufahrt in einer Kurve sollte der passende Lenkwinkel möglichst schnell eingeschlagen werden.

3. (2000 – 6000ms) Während der Kurvenfahrt sollte dieser Winkel gehalten werden.
4. (6000 – 6500ms) Schließlich sollte der Lenkwinkel in einem stabilen Geradeauslauf zurückgestellt werden.

Da die zwei Kurven der Strecke identisch sind, wird dieser Vorgang nochmals wiederholt. Auf diese Weise wird eine komplette Runde der Strecke durchgefahren.

Von der Abbildung ist es zu erkennen, dass Pure Pursuit eine deutlich stabilere Kurvenfahrt erzielt. Das ist auf der Tatsache zurückzuführen, dass Follow-the-carrot größere Schwingungsamplituden in dem Lenkungssystem verursacht.

4.3.3 Pure Pursuit versus Follow-the-carrot, Teststrecke 2



Abbildung 15: Teststrecke 2

Auf der zweiten Strecke wurde getestet wie gut die Algorithmen den Lenkwinkel in einer Kurve halten. Da musste das Fahrzeug im Kreis fahren auf einer Fahrbahn mit einem konstanten Kurvenradius von 100 cm. Dabei hat Pure Pursuit wieder bessere Ergebnisse erzielt (vgl. Abbildung 16).

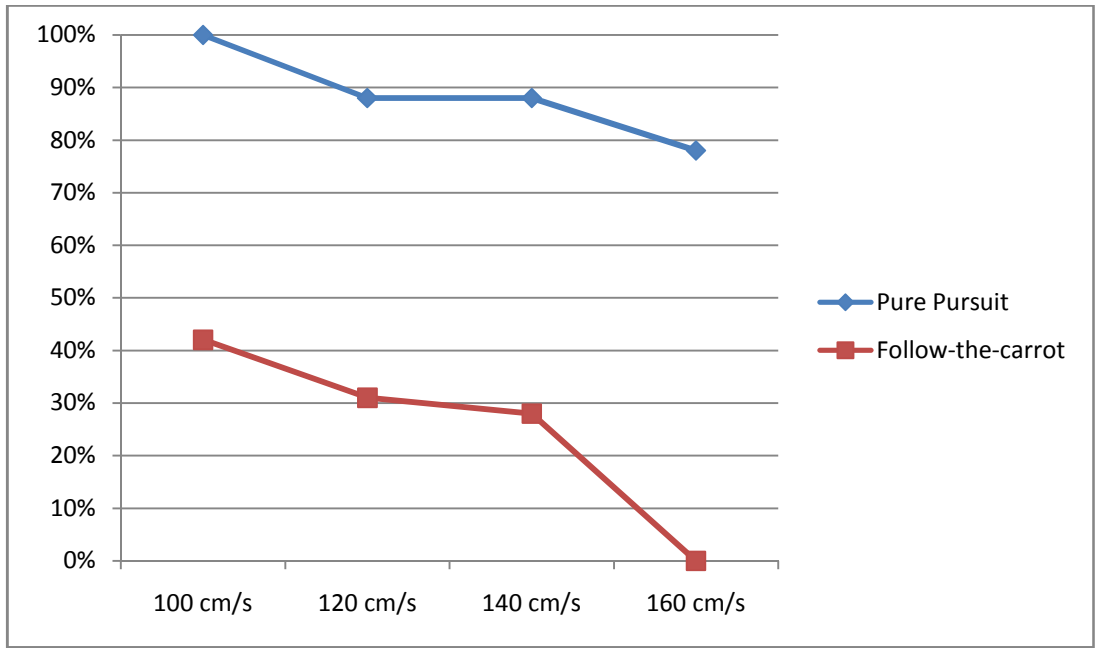


Abbildung 16: Stabilität[%] der Fahrten (Teststrecke 2) bei 100, 120,140 und 160 cm/s

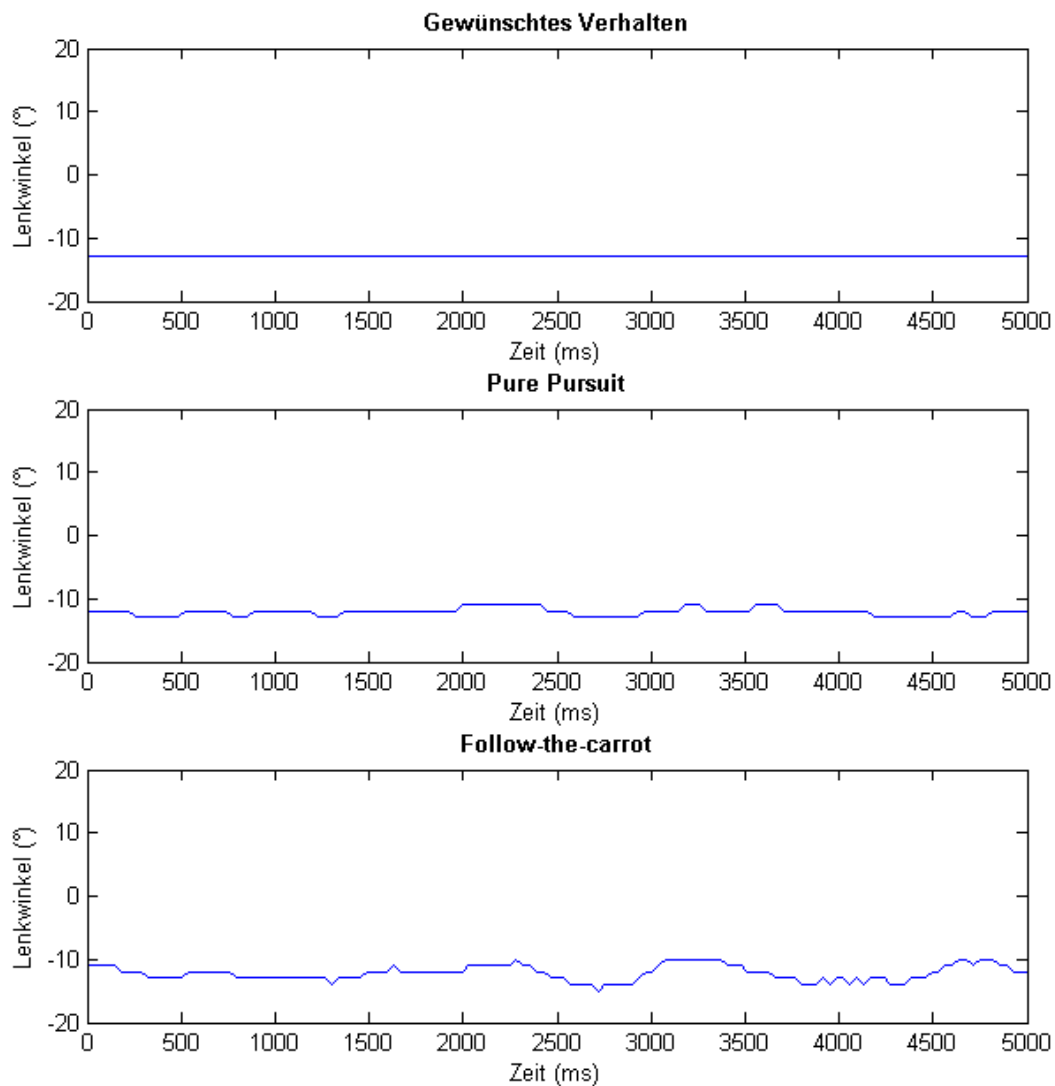


Abbildung 17: Vergleich der Lenkungswinkelverläufe von Pure Pursuit und Follow-the-carrot mit dem gewünschten Verhalten(Teststrecke 2)

Die Lenkungswinkelverläufe von gewünschtem Verhalten, Pure Pursuit und Follow-the-carrot in einer Runde der Teststrecke sind in Abbildung 17 veranschaulicht. Somit lässt sich die Stabilität der beiden Algorithmen graphisch vergleichen. Auf der y-Achse ist der Lenkwinkel in Grad zum entsprechenden Zeitpunkt auf der x-Achse eingetragen. Die negativen Lenkwinkelwerte entsprechen einen linken Ausschlag, die positive einen rechten.

Um diese Teststrecke schwingungsfrei zu bewältigen muss der Lenkeinschlag konstant sein. Der Innenradius der Strecke beträgt 100 cm. Da die Fahrspur 40 cm breit ist, ergibt sich, dass der Außenradius 140 cm ist. Damit das Fahrzeug die Mitte der Fahrbahn verfolgt, muss sein

Massenzentrum eine Kurvenfahrt mit dem Radius von 120 cm leisten bzw. mit der Krümmung von 0.008 cm^{-1} .

(vgl. Abschnitt 2.1)

$$\delta = \text{atan}(l\gamma)$$

$$\delta = \text{atan}(27 * 0.008)$$

Daraus folgt, dass ein Lenkwinkel von 12.7° eingeschlagen werden soll, damit eine optimale Kurvenfahrt erreicht wird.

4.3.4 Pure Pursuit versus Follow-the-carrot, Teststrecke 3

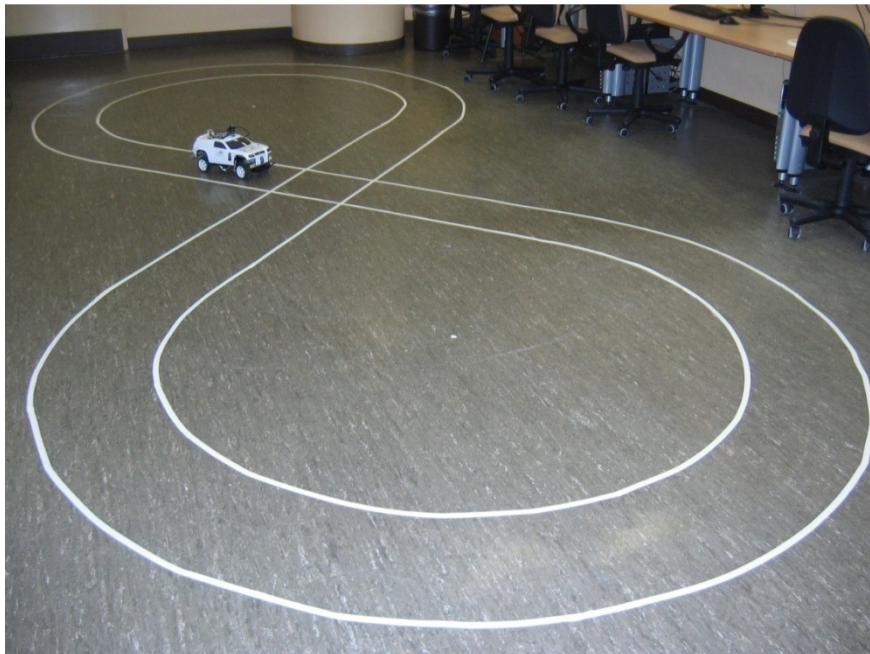


Abbildung 18: Teststrecke 3

Die dritte Strecke ist 8 förmig. So wird das Verhalten der Fahrbahnverfolgung bei sowohl rechten, als auch linken Kurven innerhalb einer Strecken getestet. Im Vergleich zu Follow-the-carrot, hat Pure Pursuit bessere Stabilität geleistet und somit wurde ein autonomes Fahren bei deutlich höherer Geschwindigkeit realisiert (vgl. Abbildung 19).

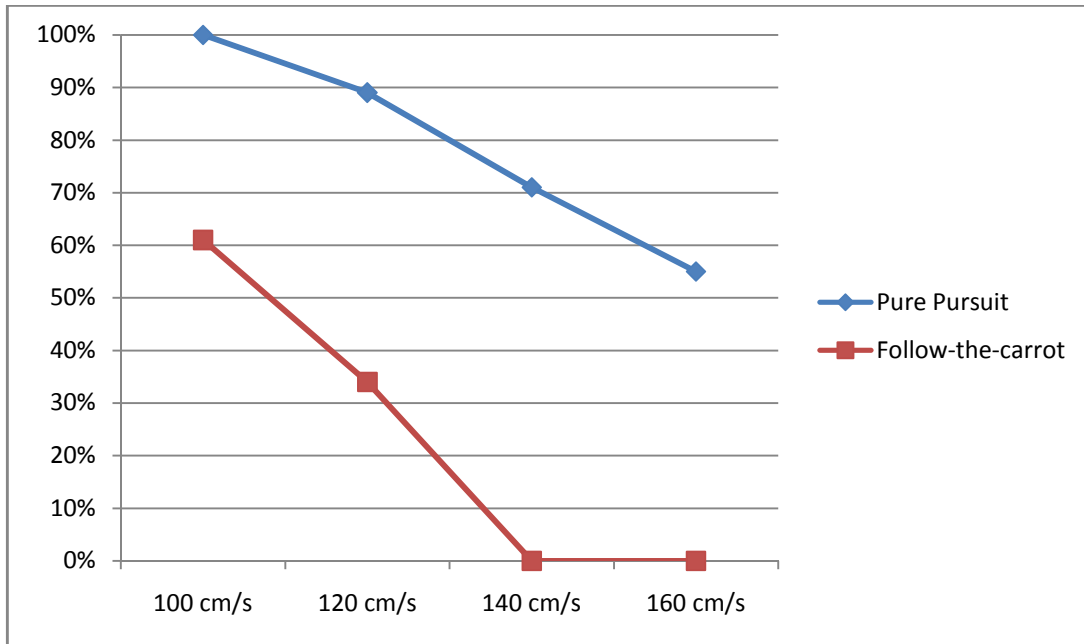


Abbildung 19: Stabilität[%] der Fahrten (Teststrecke 3) bei 100, 120,140 und 160 cm/s

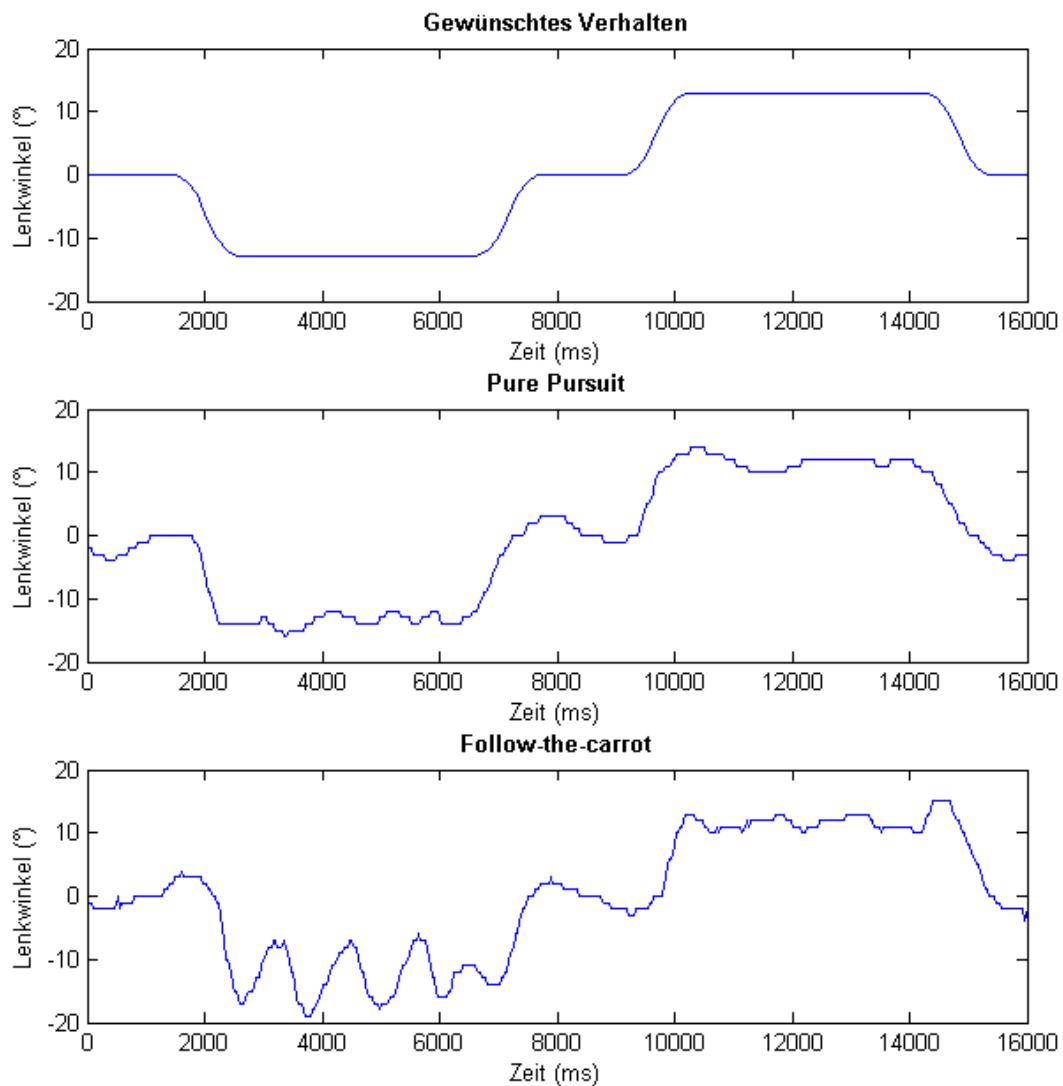


Abbildung 20: Vergleich der Lenkungswinkelverläufe von Pure Pursuit und Follow-the-carrot mit dem gewünschten Verhalten(Teststrecke 3)

Die Lenkungswinkelverläufe von gewünschtem Verhalten, Pure Pursuit und Follow-the-carrot in einer Runde der Teststrecke sind in Abbildung 20 veranschaulicht. Somit lässt sich die Stabilität der beiden Algorithmen graphisch vergleichen. Auf der y-Achse ist der Lenkwinkel in Grad zum entsprechenden Zeitpunkt auf der x-Achse eingetragen. Die negativen Lenkwinkelwerte entsprechen einen linken Ausschlag, die positive einen rechten.

Die Lenkungswinkelverläufe dieser Strecke weisen darauf hin, dass die Fahrbahnverfolgung unterschiedliches Verhalten bei linken und rechten Kurven aufweist. Das ist auf Störungen des Lenkungssystems zurückzuführen.

4.3.5 Auswertung der Algorithmen Follow-the-carrot und Pure Pursuit

Bei der Stabilitätsanalyse der Algorithmen hat Follow-the-carrot im Vergleich zu Pure Pursuit schlechtere Ergebnisse erzielt. Trotz der leichten Implementierung und Verständlichkeit, hat Follow-the-carrot einen großen Nachteil. Das Fahrzeug orientiert sich ständig nur an einem Ziel, das in gewisse Entfernung vom Fahrzeug liegt. Somit wird die entsprechende Fahrspur-geometrie zwischen Fahrzeug und Ziel nur teilweise berücksichtigt. Zur Folge schneidet das Fahrzeug die Kurven bei der autonomen Fahrbahnverfolgung. Dennoch ist Follow-the-carrot als Vergleichsbasis für andere Path Tracking Algorithmen sehr nützlich.

Ähnlich wie bei Follow-the-carrot, berechnet Pure Pursuit den gewünschten Lenkwinkel in Bezug auf ein Ziel auf der Fahrbahn. Dennoch basiert dieser Algorithmus auf die Krümmung des Bogens, der Fahrzeug und Ziel verbindet. Auf diese Weise entsteht eine stabilere Fahrt, weil die Krümmung im Verlauf einer Strecke zum großen Teil konstant ist. Folglich werden die Änderungen des Lenkwinkels stark reduziert. Daraus resultiert die kleinere Amplitude der Schwingungen in dem Lenkungssystem. Somit leistet Pure Pursuit auch bei höheren Geschwindigkeiten eine stabile Fahrbahnverfolgung.

5 Zusammenfassung

Die hier vorgestellte Arbeit ist im Forschungsprojekt FAUST (Fahrerassistenz- und Autonome Systeme) zur Teilnahme am Carolo-Cup 2009 entstanden. Die Aufgabe war eine autonome Fahrbahnverfolgung, basiert auf dem Fahrspurerkennungsalgorithmus Polaris zu realisieren. Polaris liefert zwei Polynome, die die zwei zur Fahrspur gehörigen Fahrbahnmarkierungen im Fahrzeugkoordinatensystem beschreiben [Eike Jenning, 2009].

Das Verfahren zur Fahrbahnverfolgung beruht teilweise auf dem Verhaltensmuster des Menschen beim Fahren. Anhand der Fahrspurerkennung wird kontinuierlich ein Ziel lokalisiert. Danach wird der Lenkwinkel in Bezug auf das Ziel mittels der geometrischen Algorithmen „Follow-the-carrot“ oder „Pure Pursuit“ berechnet. Schließlich wird die Richtung des Fahrzeuges korrigiert.

Das Ziel ist ein Punkt im Fahrzeugkoordinatensystem und liegt idealerweise in der Mitte der Fahrspur. Um die Koordinaten der Fahrspurmitte zu bestimmen, wird die Tangente des Polynoms gebildet, die eine der beiden Spurlinien beschreibt. Die Entfernung zwischen Fahrzeug und Ziel spielt eine signifikante Rolle im autonomen Fahren. Kleine Änderungen der Zielentfernung können die Stabilität der Fahrbahnverfolgung stark beeinflussen. Die bestgeeignete Zielentfernung ist empirisch ermittelt worden hinsichtlich des minimalen Kurvenradius bei Carolo-Cup.

Die nächste Herausforderung war den Lenkwinkel in Bezug auf das Ziel zu bestimmen. An dieser Stelle wurden zwei Path Tracking Algorithmen eingesetzt. Follow-the-carrot und Pure Pursuit sind traditionelle, geometrische Algorithmen, mit der Aufgabe einen gewünschten Lenkwinkel zu berechnen damit das Fahrzeug eine gewünschte Position auf der Fahrbahn erreicht. Zur Ermittlung des Lenkwinkels berechnet Follow-the-carrot den Winkel zwischen Fahrzeugausrichtung und Ziel. Pure Pursuit basiert auf die Krümmung des Kreisbogens, der Fahrzeug und Ziel verbindet. Das Fahrzeug reagiert kontinuierlich mit einer Kurvenfahrt dieser Krümmung, um das Ziel zu erreichen. Beide Path Tracking Algorithmen haben einen großen Nachteil. Sowohl Follow-the-carrot als auch Pure Pursuit schneiden die Kurven, weil bei einer Zielverfolgung die Fahrspurgeometrie nur teilweise berücksichtigt wird.

Nach der Festlegung des Lenkwinkels, soll ein Stellwert für die Lenkaktorik ermittelt werden. Um eine Lenkwinkelsteuerung zu realisieren wurde ein mathematisches Verhältnis zwischen Lenkwinkel in Grad und dem Stellwert festgelegt.

Die Stabilität der Fahrbahnverfolgung mit den Algorithmen Pure Pursuit und Follow-the-carrot wurde auf drei Teststrecken bei unterschiedlichen Geschwindigkeiten verglichen. Pure Pursuit hat deutlich bessere Stabilität gewährleistet, weil dieser Algorithmus weniger Schwingungen in dem Lenkungssystem verursacht. Infolgedessen wurde eine Fahrbahnverfolgung mittels Pure Pursuit auch bei höherer Geschwindigkeit möglich.

6 Ausblick

In diesem Kapitel werden weitere Entwicklungsschritte zur Optimierung der autonomen Fahrbahnverfolgung vorgestellt.

6.1 Tempomat

Mit der Veränderung der Fahrspurgeometrie ändern sich auch die Präzisions- und Stabilitätsanforderungen an der Fahrbahnverfolgung. Ein Tempomat kann die Geschwindigkeit bezüglich dieser Anforderungen regeln. So wird eine niedrigere Geschwindigkeit während Kurvenfahrt eingestellt. Gerade Strecken werden dagegen maximal schnell gefahren um eine möglichst schnelle Fahrbahnverfolgung zu erreichen. Dafür sind Reduktion der Fahrzeugvibrationen und sehr genaue Kamerakalibrierung nötig, damit eine sehr präzise Fahrspurerkennung fern vor dem Fahrzeug erfolgen kann.

6.2 Dynamische Zielentfernung

Um Schwingungen in dem Lenkungssystem zu vermeiden und somit höhere Geschwindigkeiten zu erreichen, soll die Entfernung zum Ziel möglichst groß sein(vgl. Abschnitt 3.1.1). Für das entwickelte Verfahren ist die Zielentfernung hinsichtlich des minimalen Kurvenradius bei Carolo-Cup empirisch ermittelt worden. Dennoch ist es vorteilhafter die Zielentfernung dynamisch zu berechnen, weil die bestgeeignete Zielentfernung mit dem aktuellen Kurvenradius der Fahrspur gebunden ist.

6.3 Optimierung der Lenkwinkelsteuerung

6.3.1 Lernendes System zur Verbesserung der Lenkwinkelsteuerung

Um eine Lenkwinkelsteuerung zu realisieren, soll ein mathematisches Verhältnis zwischen einer lenkungsrelevanten Größe (z.B. Radius, Krümmung der Kurve, Lenkwinkel usw.) und der Stellwert ermittelt werden(vgl. Abschnitt 3.3). Hierzu kann ein lernendes System entwickelt werden. Dieses System sollte ein bestgeeignetes, mathematisches Verhältnis ermitteln indem es die Präzision und die Stabilität der Fahrbahnverfolgung analysiert.

6.3.2 Lenkwinkelregelung

Anhand einer Regelung des Lenkungssystems kann der gewünschte Lenkwinkel deutlich präziser umgesetzt werden (vgl. Anhang A). Dadurch wird der Einfluss der Reibungskräfte und die Störungen in dem Lenkungssystem minimiert. Zu diesem Zweck sind geeignete Sensoren zur Messung vom Lenkwinkel der Vorderräder erforderlich.

Abbildungsverzeichnis

Abbildung 1: Das Modellfahrzeug Nebula mit Kamera und Embedded-PC	2
Abbildung 2: Das Modellfahrzeug ohne Karosserie	3
Abbildung 3: Fahrzeug mit Vorderachslenkung und Lenkwinkel der inneren und äußeren Vorderräder [Reza Jazar, 2005].....	6
Abbildung 4: Systemkomponenten der Fahrzeugquerführung	7
Abbildung 5: Monochrom-USB-Kamera UI-1226LE-M.....	8
Abbildung 6: Fahrspurerkennung durch Polaris	9
Abbildung 7: Bestimmung des Zieles	12
Abbildung 8: Follow-the-carrot.....	15
Abbildung 9: Pure Pursuit	16
Abbildung 10: Test Tool zur Sensordatenanalyse.....	20
Abbildung 11: Spurgeometrie der Teststrecke 1	21
Abbildung 12: Teststrecke 1.....	22
Abbildung 13: Stabilität[%] der Fahrten (Teststrecke 1) bei 80, 120 und 160 cm/s.....	23
Abbildung 14: Vergleich der Lenkungswinkelverläufe von Pure Pursuit und Follow-the-carrot mit dem gewünschten Verhalten(Teststrecke 1)	24
Abbildung 15: Teststrecke 2.....	25
Abbildung 16: Stabilität[%] der Fahrten (Teststrecke 2) bei 100, 120,140 und 160 cm/s.....	26
Abbildung 17: Vergleich der Lenkungswinkelverläufe von Pure Pursuit und Follow-the-carrot mit dem gewünschten Verhalten(Teststrecke 2)	27
Abbildung 18: Teststrecke 3.....	28
Abbildung 19: Stabilität[%] der Fahrten (Teststrecke 3) bei 100, 120,140 und 160 cm/s.....	29
Abbildung 20: Vergleich der Lenkungswinkelverläufe von Pure Pursuit und Follow-the-carrot mit dem gewünschten Verhalten(Teststrecke 3)	30
Abbildung 21: Informationsfluss eines zu steuernde Systems	40
Abbildung 22: Standartregelkreis (Quelle: Wikipedia).....	42

Tabellenverzeichnis

1 Die Messergebnisse der Vorderräder-Lenkwinkel bei den entsprechenden Stellwerten.....	18
2 Stabilität[%] der Fahrten, relativ zu der stabilste Fahrt bei Geschwindigkeit von 80 cm/s.....	21
3 Stabilität[%] der Fahrten, relativ zu der stabilste Fahrt bei Geschwindigkeit von 120 cm/s.....	21
4 Stabilität[%] der Fahrten, relativ zu der stabilste Fahrt bei Geschwindigkeit von 160 cm/s.....	22

Literaturverzeichnis

- [AAET, 2009] *Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme für Transportmittel*, Gesamtzentrum für Verkehr Braunschweig. 2009
- [Alexander Kant 2007] Alexander Kant: *Bildverarbeitungsmodul zur Fahrspurerkennung für ein autonomes Fahrzeug*. Bachelorarbeit, Hochschule für Angewandte Wissenschaften. 2007
- [Carolo Cup] *Homepage des Carolo-Cup Wettbewerbs*. - [http:// www.carolo-cup.de/](http://www.carolo-cup.de/)
- [Carolo-Cup-Regelwerk 2009] *Carolo-Cup Regelwerk 2009*. http://www.carolo-cup.de/uploads/media/20080507_Carolo-Cup_Regelwerk.pdf
- [Dennis Berger, 2008] Dennis Berger: *Fahrspurerkennung mit Three Feature Based Lane Detection Algorithm (TFALDA)*. Studienarbeit, Hochschule für Angewandte Wissenschaften Hamburg. 2008
- [Eike Jenning, 2009] Eike Jenning: *Systemidentifikation eines autonomen Fahrzeugs mit einer robusten, kamerabasierten Fahrspurerkennung in Echtzeit*. Masterarbeit, Hochschule für Angewandte Wissenschaften Hamburg. 2009
- [Jim Ledin, 2004] Jim Ledin: *Embedded Control Systems in C/C++: An Introduction for Software Developers Using Matlab*
- [Martin Lundgren, 2003] Martin Lundgren: *Path Tracking and Obstacle Avoidance for a Miniature Robot*, Master Thesis 2003 Department of Computer Science, Umea University, Sweden
- [Marko Natzke, 2007] Marko Natzke: *System und Kommunikationsarchitektur zum autonomen Betrieb eines Modellfahrzeuges*. Diplomarbeit, Hochschule für Angewandte Wissenschaften Hamburg. 2008
- [R. Craig Coulter, 1992] R. Craig Coulter: *Implementation of the Pure Pursuit Tracking Algorithm*, Robotics Institute, Carnegie Mellon University, 1992
- [Reza N. Jazar, 2005] Reza N. Jazar: *Vehicle Dynamics: Theory and Application*
- [Rolf Isermann, 2006] Rolf Isermann: *Fahrdynamik-Regelung : Modellbildung, Fahrerassistenzsysteme, Mechatronik*
- [Steve Northover, Mike Wilson, 2004] Steve Northover, Mike Wilson: *SWT: The Standard Widget Toolkit, Volume 1*
- [Sven Geibert, 2007] Sven Geibert: *Gyroskopische Lenkregelung für ein autonomes Fahrzeug*. Bachelorarbeit, Hochschule für Angewandte Wissenschaften Hamburg. 2007

[Thies Lauk-Reineke, 2007] Thies Lauk-Reineke: *Aufbau einer Versuchsplattform für die Implementierung und Erprobung von Fahrassistenzsysteme auf Basis einer TriCore-CPU mit einem OSEK-Betriebssystem*. Diplomarbeit, Institut für Regelungstechnik, Technische Universität Braunschweig. 2007

[Younguk Yim; Se-Young Oh,1999] Younguk Yim; Se-Young Oh: *Three-feature based automatic lane detection algorithm (TFALDA) for autonomous driving*, 1999

A Grundlagen der Regelung

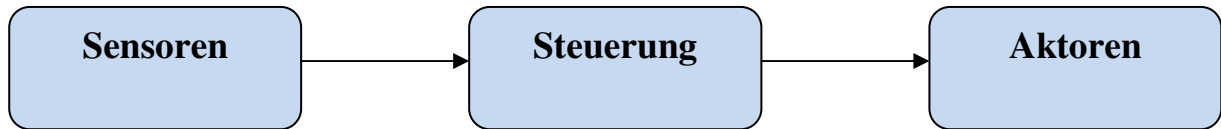


Abbildung 21: Informationsfluss eines zu steuernde Systems

In Abbildung 21 ist der Informationsfluss eines zu steuernde Systems zu verfolgen. Die Sensoren haben die Aufgabe das Umfeld eines Systems zu erfassen. Als nächstes werden die erfassten Daten von einem Steuerung-Algorithmus verarbeitet und schließlich beeinflusst die Steuerung den Arbeitsablauf der Aktoren relativ zu diesen Daten.

„**Steuerung** ist ein Vorgang in einem System, bei dem eine oder mehrere Größen als Eingangsgrößen andere Größen als Ausgangsgrößen aufgrund der dem System eigentümlichen Gesetzmäßigkeiten beeinflussen.“

Definition der Steuerung nach DIN 19226

Nachteil der Steuerung ist, dass keine fortlaufende Rückkopplung der Ausgangsgröße vorhanden ist. Somit lassen sich Störungen in der Aktorik nicht korrigieren. Im Falle, dass eine solche Rückkopplung angebracht worden ist, spricht man von einer Regelung.

„Das *Regeln*, die **Regelung**, ist ein Vorgang, bei dem *fortlaufend* eine Größe, die Regelgröße (zu regelnde Größe), erfasst, mit einer anderen Größe, der Führungsgröße, verglichen und im Sinne einer Angleichung an die Führungsgröße beeinflusst wird. Kennzeichen für das Regeln ist der geschlossene Wirkungsablauf, bei dem die Regelgröße im Wirkungsweg des Regelkreises fortlaufend sich selbst beeinflusst.“

Definition der Regelung nach DIN 19226

Das folgende Beispiel demonstriert den Unterschied zwischen Steuerung und Regelung. Zu betrachten ist ein Heizungssystem, bestehend aus einem Ofen und einem Controller, der den Ofen in bestimmte Zeitintervalle aus- und einschaltet, um die gewünschte Raumtemperatur zu erreichen. Zuerst wird eine Steuerung und danach eine Regelung für diese System beschrieben.

Steuerung: Das Heizungssystem wird relativ zu der Außentemperatur gesteuert. Um die gewünschte Innentemperatur zu erreichen, ist es möglich, experimentell ein Zusammenhang zwischen der Außentemperatur und dem Arbeitsablauf der Ofen zu bestimmen. Angenom-

men, erzielt das System die gewünschte Innentemperatur bei einer Außentemperatur durch einen wiederholten Arbeitsablauf-Zyklus (der Ofen wird für 5 Minuten eingeschaltet und dann für 10 Minuten ausgeschaltet). In diesen Fall wird die Steuerung wird nur so lange richtig funktionieren bis das System unverändert bleibt. Das Öffnen des Fensters oder die Ersetzung des Luft-Filters zum Beispiel führen zu Änderungen. Dann wird die gewünschte Innentemperatur nicht mehr aufrechterhalten. Dies ist eindeutig ein unbefriedigendes Design.

Regelung: Das System misst zu regelnde Größe (Innentemperatur) und schaltet den Ofen ein und aus bezüglich diese Messung. Dieses System wird sich automatisch bei Änderungen des Systems anpassen.

B PID-Regler Optimierung

Regler beeinflussen selbsttätig in einem meist technischen Prozess eine oder mehrere physikalische Größen auf ein vorgegebenes Niveau unter Reduzierung von Störeinflüssen. Ihre Behandlung ist Kern der Regelungstechnik. Dazu vergleichen Regler innerhalb eines Regelkreises (Abb.) laufend das Signal des Sollwertes w mit dem gemessenen und zurückgeführten Istwert y der Regelgröße und ermitteln aus dem Unterschied der beiden Größen – der Regelabweichung e (Regeldifferenz) – eine Stellgröße u , welche die Regelstrecke so beeinflusst, dass die Regelabweichung spätestens im eingeschwungenem Zustand zu einem Minimum wird.

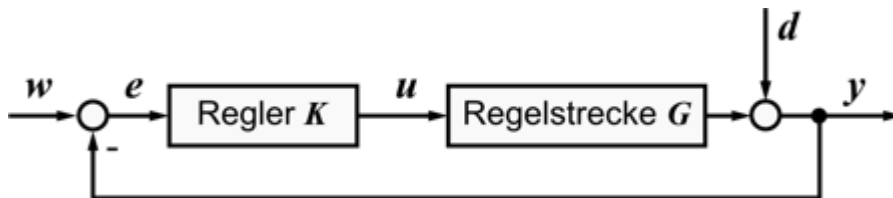


Abbildung 22: Standardregelkreis (Quelle: Wikipedia)

Die Standard-Regler für lineare Systeme sind:

- P-Regler (P-Glied) mit proportionalem Verhalten
- PD-Regler (P-Glied und D-Glied) mit proportionalem und differentialem Verhalten
- PI-Regler (P-Glied und I-Glied) mit proportionalem und integralem Verhalten
- PID-Regler (P-Glied, I-Glied und D-Glied) mit proportionalem, integralem und differentialem Verhalten

B.1 P Regler (Proportionalglied)

Der P Regler entspricht dem Proportionalglied.

$$u = K_p e$$

Dabei wird die Regelabweichung mit einer konstanten Gewichtung K_p multipliziert, um die Stellgröße zu berechnen. Wenn K_p das richtige Vorzeichen hat, versucht dieser Regler immer die Regeldifferenz e zu minimieren. Um das beste Ergebnisse zu erzielen, sollte eine geeignete K_p ausgewählt werden. Wird K_p ein kleiner Wert zugewiesen, reagiert der Regler zu langsam auf Änderungen in e . Um die Reaktionszeit des Reglers zu verbessern, sollte K_p vergrößert werden.

bert werden. Dennoch, wenn dieser Koeffizient zu groß wird, können Schwingungen in der Regelung auftreten. Um diese zu vermeiden, kommt das Differenzialglied zum Einsatz.

B.2 PD Regler (Proportional- und Differenzialglied)

Der PD Regler besteht aus der Kombination eines P-Gliedes mit einem D-Glied.

$$u = K_p \left(e + K_d \frac{de}{dt} \right)$$

Das D-Glied ist ein Differenzierer, der nur in Verbindung zu Reglern mit P- und/oder I-Verhalten als Regler eingesetzt wird. Er reagiert nicht auf die Regelabweichung, sondern nur auf deren Änderungsgeschwindigkeit. Der D-Anteil ermöglicht eine schnelle Reaktion des Reglers, weil Sie eine große K_p erlaubt. So kann die Bandbreite erhöht werden, bis die Grenze des Systems erreicht sind.

Das Differenzialglied führt aber zu Verstärkung der Systemstörungen. Aus diesem Grund sollte das D-Glied bei störungsempfindlichen Systemen vermieden werden.

B.3 PI Regler(Proportional und Integralglied)

Der PI Regler besteht aus der Kombination eines P-Gliedes mit einem I-Glied.

$$u = K_p \left(e + K_i \int_0^t e \, d\tau \right)$$

Das Integralglied summiert die Regeldifferenzen über die Zeit. Bei konstanter Regelabweichung, wächst die Ausgangsgröße linear an. Der PI Regler ist ein genauer aber langsamer Regler.

B.4 PID Regler

$$u = K_p \left(e + K_i \int_0^t e \, d\tau + K_d \frac{de}{dt} \right)$$

Der PID Regler ermittelt die Stellgröße u mit Hilfe des Integrals und Differenzials der Regelabweichung e . K_p steht für die Gewichtung des Proportionalgliedes, K_i für das Integralglied und K_d fürs Differenzialglied

B.5 PID-Regler Optimierung

1. K_p wird einen kleinen Wert zugewiesen um Schwingungen in der Regelung zu vermeiden. $K_d = K_i = 0$.
2. Test mit einem geeigneten Eingangssignal wird durchgeführt. Das Ergebnis sollte ein zu langsamer Regler sein.
3. Der K_p Wert sollte erhöht werden, um die Reaktionszeit des Reglers zu kürzen. Bei einem zu hohen K_p Wert können Schwingungen auftreten.
4. K_d sollte erhöht werden um die Schwingungen zu minimieren. Falls der Regler zu langsam wird sollte K_d verkleinert werden.
5. Falls eine konstante Regelabweichung auftritt, sollte das Integralglied angepasst werden.

C Implementierung der Zielfestlegung

```
/* This function searches a goal point, which lies at goalPointDistance cm from the car and
 * edgeDistance cm from the orientation line of the lane.
 */
CvPoint2D64f ControlTaskPolaris::computeGoalPoint(PolarisLanePtr polarisLane){

    CvPoint2D64f A;           // a point on the edge of the lane
    CvPoint2D64f B;           // a point on a tangent which touches the right lane at A
    CvPoint2D64f goalPoint;   // C - the goalpoint

    // line equation:  $y = m*x + b$ 
    double lineABLen = 0.0;
    double mAB = 0.0; double mAC = 0.0; double mBC = 0.0; // the slopes
    double bAC = 0.0; double bBC = 0.0; // the y-intercepts

    // Bisection Method
    // compute the number of iterations that the bisection method would need to converge to
    // a road within the tolerance of 1cm
    int numberOfSteps = (int)(log2(MAX_LOOK_AHEAD_DISTANCE)) + 1;
    double tempDistance = 0; // distance from the car center to the current goal point
    A.x = 0;

    for(int steps = 1; steps <= numberOfSteps; steps++){
        // Bisection Method
        if(tempDistance < goalPointDistance)
            A.x += MAX_LOOK_AHEAD_DISTANCE/pow(2.0 , steps);
        else
            A.x -= MAX_LOOK_AHEAD_DISTANCE/pow(2.0 , steps);

        // compute the y-coordinate of A and find B
        if(leftLineOrientation == 1){
            A.y = polarisLane->getLeftEdge()->getValue(A.x);
            B.x = A.x + edgeDistance; // so that  $B.x - A.x \geq edgeDistance$ 
            B.y = A.y + (B.x - A.x)*(polarisLane->getLeftEdge()->getGradient(A.x));
        }
        else{
            A.y = polarisLane->getRightEdge()->getValue(A.x);
            B.x = A.x + 15;
            B.y = A.y + (B.x - A.x)*(polarisLane->getRightEdge()->getGradient(A.x));
        }

        // compute the length of AB
        lineABLen = sqrt(pow(B.x - A.x,2) + pow(B.y - A.y,2));
        mAB = (B.y - A.y)/(B.x - A.x); // the slope of AB
        mAC = -1.0/mAB; // AC and AB are perpendicular

        // compute the slope of BC
        // tan (angle ABC) = edgeDistance/lineABLen
        if(leftLineOrientation == 1)
            mBC = (mAB - edgeDistance/lineABLen)/(1 + mAB*edgeDistance/lineABLen);
        else
            mBC = (mAB + edgeDistance/lineABLen)/(1 - mAB*edgeDistance/lineABLen);

        // compute the y-intercepts
        bBC = B.y - mBC*B.x;
        bAC = A.y - mAC*A.x;

        // the goalPoint is the intersection of the lines AC and BC
        goalPoint.x = (bAC - bBC)/(mBC - mAC);
        goalPoint.y = mAC*goalPoint.x + bAC;

        // The center of the coordinate system lies right in front of the car.
        // In order to shift it to the car center, CAR_LENGTH/2 will be added to the
        // x-coordinate
        // move the coordinate system center to the center of the car
        goalPoint.x += CAR_LENGTH/2;

        // compute the distance from the car center to the current goal point
        tempDistance = sqrt(pow(goalPoint.x, 2) + pow(goalPoint.y, 2));
    }

    return goalPoint;
}
```

D Implementierung der Lenkwinkelermittlung

```
/* This function computes a suitable steering angle in relation to a goal point in a vehicle
 * coordinate system.
 */
double ControlTaskPolaris::computeSteeringAngle(CvPoint2D64f goalPoint, int algorithmChoice){

    double steeringAngle;

    // Path Tracking Algorithms
    if(algorithmChoice == FOLLOW_THE_CARROT){

        //----- 1. Follow-The-Carrot
        // Follow-The-Carrot is a path tracking algorithm.
        // Obtain a goal point, then aim the vehicle towards that point.

        steeringAngle = atan(goalPoint.y/(goalPoint.x))* RAD_TO_GRAD;
        steeringAngle *= frictionCoefficient; // [1.0, 1.4]
        // convert the steering angle to control value
        steeringAngle *= 5.0; // multiply with 5 to match protocol(1° -> 5)
        steeringAngle -= 20.0; // offset
    }
    else {

        //----- 2. Pure Pursuit
        // Pure Pursuit is a tracking algorithm that works by calculating the curvature
        // that will move a vehicle from its current position to some goal position.

        double goalPointDistancePow2 = pow((goalPoint.x), 2) + pow(goalPoint.y, 2);
        double curvature = (2.0*goalPoint.y)/goalPointDistancePow2;
        steeringAngle = atan(curvature*WHEELBASE)* RAD_TO_GRAD;
        steeringAngle *= frictionCoefficient; // [1.0, 1.4]
        // convert the steering angle to control value
        steeringAngle *= 5; // multiply with 5 to match protocol(1° -> 5)
        steeringAngle -= 20.0; // add steering offset
    }

    return steeringAngle;
}
```

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.

Ort, Datum

Unterschrift