

# Technische Universität Dresden

Fakultät Elektrotechnik und Informationstechnik

Institut für Regelungs- und Steuerungstheorie

## Studienarbeit

### Verbesserung der Spurerkennung und -verfolgung autonomer Modellfahrzeuge

vorgelegt von: James Vero Asghar  
geboren am: 3. Dezember 1997 in Austin, Texas, Vereinigte Staaten

Betreuer:	Dr.-Ing. Carsten Knoll M. Sc. Paul Auerbach (Barkhausen Institut)
Verantwortlicher Hochschullehrer:	Prof. Dr.-Ing. habil. Dipl.-Math. K. Röbenack
Tag der Einreichung:	12. Mai 2023

Bitte ersetzen Sie diese Seite vor dem Binden mit der Aufgabenstellung.

# Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die von mir am heutigen Tage an der Fakultät Elektrotechnik und Informationstechnik eingereichte Studienarbeit zum Thema

## **Verbesserung der Spurerkennung und -verfolgung autonomer Modellfahrzeuge**

selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Schriften entnommen sind, wurden als solche kenntlich gemacht.

Dresden, 12. Mai 2023

James Vero Asghar

## Kurzfassung

In der Forschungsgruppe Connected Robotics Lab (CoRoLa) am Barkhausen Institut wurde eine Demonstration mit ferngesteuerten zweiachsigen Fahrzeugen entwickelt. Die Demonstration ist derzeit im Einsatz, um die Kommunikation zwischen Fahrzeugen ähnlich einem Internet of Things (IoT) Netzwerk zu modellieren.

In dieser Arbeit wird ein nichtlinearer Regelalgorithmus eingeführt, um die Komplexität eines realen Systems besser zu modellieren. Der sogenannte Stanley-Regler [1] ist ein nichtlinearer Regler, der für den Einsatz bei der Querregelung von zweiachsigen Fahrzeugen entwickelt wurde. Eine Fisheye-Kamera wird verwendet, um Bilder der Strecke zu erstellen. Diese Bilder werden dann digital verarbeitet, um eine gelbe Fahrspur zu finden, der die Fahrzeuge folgen sollen. Sobald diese gefunden ist, werden die Querabweichung zwischen dem Fahrzeug und der Fahrspur sowie die Ausrichtung der Fahrspur berechnet.

## Abstract

At the Connected Robotics Lab (CoRoLa) research group at the Barkhausen Institut, a demonstration using remote controlled two axle vehicles was developed. The demonstration is currently in use in order to model communications between vehicles similar to an Internet of Things (IoT) network.

In this thesis, a nonlinear control algorithm is introduced in order to better model the complexities of a real world system. The so called Stanley controller [1] is a nonlinear controller developed for use with the lateral control of two axle vehicles. A fisheye camera is used to create images of the track. These images are then digitally processed to find a yellow lane for the vehicles to follow. Once found, the cross-track-error between the vehicle and the lane as well as the lane's heading are calculated.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Regelalgorithmus</b>	<b>5</b>
2.1	Stanley-Regler . . . . .	5
2.1.1	Kleinsignalverhalten . . . . .	6
2.1.2	Großsignalverhalten . . . . .	7
2.2	Simulation . . . . .	7
<b>3</b>	<b>Bildverarbeitung</b>	<b>12</b>
3.1	Verzeichnungs Korrektur . . . . .	12
3.1.1	Kalibrierung . . . . .	13
3.1.2	Korrektur . . . . .	14
3.2	„Color-Thresholding“ . . . . .	15
3.3	Perspektivtransformation . . . . .	17
3.4	Histogramm . . . . .	18
3.5	Gleitfenstermethode . . . . .	18
3.6	Berechnung von Ausrichtung und Querabweichung . . . . .	19
3.7	Ausreißer . . . . .	20
<b>4</b>	<b>Experimentelle Untersuchungen</b>	<b>22</b>
4.1	Hardware . . . . .	22
4.2	Experimentkriterien . . . . .	22
4.2.1	Höchstgeschwindigkeit . . . . .	22
4.2.2	Maximal möglicher Fehlerwinkel . . . . .	23
4.2.3	Bahnfolgeverhalten . . . . .	23
4.2.4	Reglerausgang . . . . .	23
4.3	Ergebnisse . . . . .	24
4.3.1	Höchstgeschwindigkeit . . . . .	24
4.3.2	Maximal möglicher Fehlerwinkel . . . . .	24
4.3.3	Bahnfolgeverhalten . . . . .	25
4.3.4	Reglerausgang . . . . .	26
<b>5</b>	<b>Fazit</b>	<b>29</b>
5.0.1	Künftige Arbeit . . . . .	29
	<b>Literatur</b>	<b>31</b>



# Kapitel 1

## Einleitung

Die Forschungsgruppe Connected Robotics Lab (CoRoLa) am Barkhausen Institut hat eine Demonstration entwickelt, die ferngesteuerte zweiachsige Fahrzeuge verwendet, um die Kommunikation zwischen ferngesteuerten Fahrzeugen in einem „Internet of Things“-Netzwerk zu modellieren.

Die bisherige autonome Regelung jedes ferngesteuerten Modellfahrzeugs auf der Strecke erfolgt über einen zentralen Computer. Die Strecke besteht, wie in Abb. 1 dargestellt, aus einem schwarzen Untergrund und einer gelben Fahrbahnlinie. Das Fahrzeug nutzt eine Fischaugenkamera zur Erkennung der Fahrspur. Eine Bildverarbeitungspipeline auf dem Computer verarbeitet mit einer Frequenz von 50 Hz jedes neue Bild der Fahrzeugkamera. Die Vorwärtsgeschwindigkeit des Fahrzeugs ist dabei konstant.



**Abbildung 1** – Darstellung der Strecke.

Die bisherige Pipeline extrahierte alle gelben Pixel aus dem Bild und betrachtete deren dichteste Gruppierung als die Fahrspurlinie. Der Abstand zwischen der  $x$ -Koordinate dieser Gruppierung und der Mittellinie des Bildes diente als Eingabe für einen PID-Regler, der das Fahrzeug auf die Fahrspur lenkte.

Aufgrund der Einschränkungen der bisherigen Fahrzeugregelung, wie z. B. der geringen maximal möglichen Vorwärtsgeschwindigkeit und der mangelnden Robustheit. Der bisher verwendete Regler hat bei höheren Vorwärtsgeschwindigkeiten Schwierigkeiten, das Modellfahrzeug auf die Strecke zu halten. Deswegen bestand im Vorfeld der Arbeit der Wunsch, eine neue Regelstrategie zu implementieren. Um dieses Ziel zu erreichen, umfasst

die hiererarbeitete Lösung die Entwicklung einer neuen Bildverarbeitungspipeline und die Ersetzung des PID-Reglers durch einen Stanley-Regler.



# Kapitel 2

## Regelalgorithmus

### 2.1 Stanley-Regler

In [1] wurde im Jahr 2005 ein Regelungsalgorithmus vorgeschlagen, der Stanley-Regler. Von den Autoren wird dieser Regler (unter Bezugnahme auf das Auto des Rennteams Stanford University „Stanley“) als „Stanley-Regler“ bezeichnet.

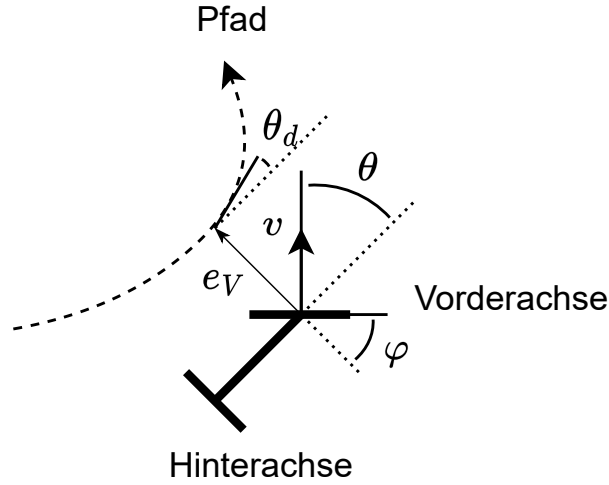
Der Stanley-Regler ist ein nichtlinearer Regelungsalgorithmus, um ein zweiachsiges Fahrzeug nur anhand der Ausrichtung des Fahrzeugs relativ zur Bahn und der Querabweichung zu der zu verfolgenden Bahn zu steuern. Der Algorithmus hat sich für das kinematische Modell eines zweiachsigen Fahrzeugs als asymptotisch global stabil erwiesen. [1]

Der Stanley-Regler ist ein Pfadfolgeregler anstelle eines Trajektorienfolgereglers. Im Wesentlichen ist ein Pfad eine rein geometrische Darstellung, die keinen Zeitbezug hat, während eine Trajektorie die Zeitverläufe der Zustandskomponenten in einem bestimmten Zeitintervall beschreibt. Als Querregler soll der Stanley-Regler das Fahrzeug auf der Bahn halten, hat aber keinen Einfluss auf die Vorwärtsgeschwindigkeit. Dieser Ansatz ermöglicht eine flexible Wahl der Fahrzeuggeschwindigkeit, die unter Berücksichtigung dynamischer Effekte nach Bedarf für eine bestimmte Anwendung gewählt werden kann. [1]

Der Stanley-Regler wird mathematisch durch die folgende Gleichung beschrieben:

$$u = \theta - \theta_d + \arctan\left(\frac{ke_V}{v}\right). \quad (2.1)$$

Dabei ist  $u$  der Reglerausgang, welcher im Wesentlichen die Änderung des Lenkwinkels  $\varphi$  bestimmt, siehe (2.4),  $\theta$  die aktuelle Ausrichtung des Fahrzeugs,  $\theta_d$  die Pfadausrichtung,  $k$  ein Skalierungsfaktor,  $v$  die Geschwindigkeit des Fahrzeugs und  $e_V$  die Querabweichung vom Mittelpunkt der Vorderachse des Fahrzeugs zum Pfad. Die geometrischen Beziehungen zwischen den Größen sind in Abb. 2 zu sehen.



**Abbildung 2** – Grafische Darstellung der geometrischen Beziehungen zwischen den Größen des Stanley-Reglers.

Der Stanley-Regler besteht aus zwei Komponenten: einer Komponente, die die Differenz zwischen der Ausrichtung des Fahrzeugs und der des Pfades behandelt, und einer Komponente, die die Querauslenkung des Fahrzeugs vom Pfad behandelt. Die erste Komponente, dargestellt durch  $\theta_d - \theta$ , beeinflusst den Lenkwinkel so, dass er parallel zum zu verfolgenden Pfad bleibt. Die zweite Komponente wird durch  $\arctan(\frac{ke_V}{v})$  dargestellt und beeinflusst den Lenkwinkel so, dass er sich auf den Pfad zubewegt, wenn sich das Fahrzeug weiter von ihm entfernt. Zusammen bewirken diese beiden Komponenten, dass das Fahrzeug auf den gewünschten Pfad gelenkt wird. [7]

### 2.1.1 Kleinsignalverhalten

Um das Verhalten des Stanley-Reglers (2.1) zu verstehen, wird der Regler um den Arbeitspunkt  $(x_V, \theta_d) = (0, 0)$  linearisiert:

$$\bar{u} \approx \theta + \frac{k}{v} e_V, \quad (2.2)$$

wobei die Geschwindigkeit  $v$  als konstant angenommen wird. Die sich daraus ergebende Gleichung (2.2) hat die Form eines PD-Reglers. Die P-Komponente wird durch  $\frac{k}{v} e_V$  gebildet. Da die Vorwärtsgeschwindigkeit des Fahrzeugs als konstant angenommen wird, besteht eine Proportionalität zwischen den Ableitungen von der Querauslenkung  $e_V$  nach Zeit und Ort. Die Ableitung von  $e_V$  nach dem Ort ist  $\theta$ . Daher entspricht im linearisierten Stanley-Regler die Fahrzeugausrichtung  $\theta$  der D-Komponente.

### 2.1.2 Großsignalverhalten

Bei großen Signalen wird der Stanley-Regler durch das Verhalten der arctan-Funktion dominiert. Die Funktion arctan ist auf einen Wert zwischen -90 und 90 Grad begrenzt und ist glatt. Das führt dazu, dass der P-Anteil des Reglers diesen Wert nicht überschreiten kann, was der physischen Realität entspricht.

## 2.2 Simulation

Um ein besseres Verständnis für das Verhalten des Stanley-Reglers in einem Modellfahrzeug zu gewinnen und auch um die Auslegung des Reglers für die experimentellen Untersuchungen vorzubereiten, wurden im Rahmen dieser Arbeit simulative Untersuchungen des Regelkreises durchgeführt. Die Simulation umfasst drei Hauptkomponenten: den Pfad, dem das Fahrzeug folgen muss, den Stanley-Regler und das Fahrzeugmodell. Zur Ermittlung der Zeitreihendaten des zweiachsigen Fahrzeugs in der Simulation wurde ein Differentialgleichungslöser eingesetzt.

Der Pfad, dem das Fahrzeug folgen muss, ist eine virtuelle Darstellung der realen Strecke, die das Fahrzeug durchfahren wird. Eine visuelle Darstellung der Fahrbahn findet sich in Abb. 3. In der Simulation wird der Pfad durch die folgende kontinuierliche statische Funktion definiert:

$$\underline{p}_P(s) = (x_p, y_p, \theta_d), \quad (2.3)$$

wobei  $s$  der Eingangsparameter Kurvenlänge ist und  $\underline{p}_P$  bzw.  $(x_p, y_p, \theta_d)$  die kartesischen Koordinaten des Pfadpunkts und die entsprechende Ausrichtung an diesem Punkt darstellen. Diese Funktion wird für eine beliebige Anzahl von  $s$ -Werten diskretisiert, um eine Liste von  $(x_p, y_p, \theta_d)$ -Werten zu erzeugen. Der Differentialgleichungslöser bezieht den Stanley-Controller und das Fahrzeugmodell ein, um die Zeitreihen des zweiachsigen Fahrzeugs unter Verwendung dieser Liste als Eingabe zu analysieren.

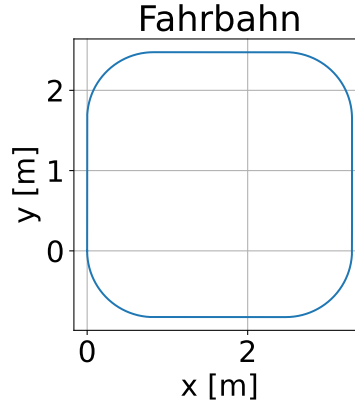
Das verwendete Fahrzeugmodell ist das Modell eines zweiachsigen Fahrzeugs mit gelenkter Vorderachse [13]. Dieses Modell wird mithilfe der folgenden nichtlinearen Zustandsraumdarstellung formuliert:

$$\dot{x} = v \cos(\theta) \quad (2.4a)$$

$$\dot{y} = v \sin(\theta) \quad (2.4b)$$

$$\dot{\theta} = \frac{v}{l} \tan(\varphi). \quad (2.4c)$$

Die Zustandskomponenten des Modells sind  $x$ ,  $y$ ,  $\varphi$  und  $\theta$ , wobei  $x$  und  $y$  die kartesischen Koordinaten des Mittelpunkts der Hinterachse sind und  $\theta$  die Ausrichtung des Fahrzeugs ist.  $\varphi$  ist der Lenkwinkel des Fahrzeugs. Die Eingangsgrößen des Modells sind  $v$  und  $\varphi$ , wobei  $v$  die Geschwindigkeit des Fahrzeugs und  $\varphi$  der Lenkwinkel ist. Der Parameter  $l$  bezeichnet den Abstand zwischen der Hinterradachse und der Vorderradachse.



**Abbildung 3** – Grafische Darstellung der Fahrbahn.

Der Lenkwinkel  $\varphi$  ist selbst ein dynamisches System. Experimentell zeigt sich, dass der Lenkwinkel  $\varphi$  sich wie ein PT1-Glied verhält. Das PT1-Glied wird durch die folgende Gleichung modelliert [8]:

$$\dot{\varphi} = \frac{(u - \varphi)}{T}, \quad (2.5)$$

wobei die Zeitkonstante  $T$  konfigurierbar ist. Das Verhalten des Stanley-Reglers bei dynamischen Einflüssen auf den Lenkwinkel ist von großer Bedeutung, da die globale asymptotische Stabilität des Reglers nur für das kinematische Zweiachsmodell nachgewiesen wurde. [1]

Da der Stanley-Regler für die Querabweichung den Mittelpunkt der Vorderachse benötigt, muss dieser aus der Hinterachse berechnet werden. Der Mittelpunkt der Vorderachse wird aus den Zustandskomponenten durch die folgende statische Funktion [13] berechnet:

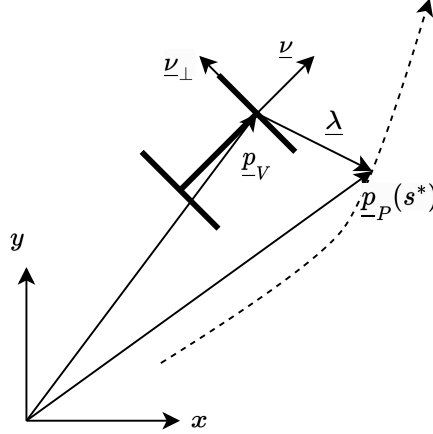
$$\underline{p}_H := \begin{bmatrix} x & y \end{bmatrix}^T \quad (2.6a)$$

$$\begin{bmatrix} x_V & y_V \end{bmatrix}^T = \underline{p}_H + l \begin{bmatrix} \cos(\theta + \pi/2) \\ \sin(\theta + \pi/2) \end{bmatrix}, \quad (2.6b)$$

wobei  $x_V$  und  $y_V$  die kartesischen Koordinaten dieses Vorderradachsemittelpunkts sind. Der Parameter  $\underline{p}_H$  bezeichnet den Positionsvektor des Hinterradachsemittelpunkts.

Wie bereits erwähnt, besteht der Stanley-Regler aus mehreren Teilen, die getrennt voneinander berechnet werden können. Die Ausrichtung  $\theta$  ist eine Zustandsvariable, daher ist sie unmittelbar verfügbar. Um die Querabweichung und die Pfadausrichtung zu berechnen, muss der richtige Bahnpunkt gewählt werden. Der Stanley-Regler verwendet den nächstliegenden Bahnpunkt vom Vorderachsmittelpunkt des Fahrzeugs, um die Ausrichtung des Pfades  $\theta_d$  und die aktuelle Querabweichung  $e_V$  zu bestimmen. Für die Simulation wird dieser bestimmt, indem der Pfadpunkt mit dem geringsten Abstand  $\underline{\lambda}$

zum Vorderachsmittelpunkt gefunden wird (entspricht Kurvenparameter  $s^*$ ). An diesem Punkt wird dann die Ausrichtung des Pfades ermittelt. Das Skalarprodukt zwischen dem Vektor senkrecht zur Fahrzeugausrichtung  $\underline{v}_\perp$  und dem Vektor vom Pfadpunkt zur Fahrzeugvorderachse  $\underline{\lambda}$  ist die Querabweichung  $e_V$ . Zur Verdeutlichung ist in Abb. 4 eine visuelle Darstellung dieser beiden Vektoren und ihres Skalarprodukts gezeigt. Die Berechnung der wichtigen Parameter ist in folgenden Gleichungen dargestellt:



**Abbildung 4** – Grafische Darstellung der Beziehungen zwischen den benötigten Parametern. Der Vektor  $\underline{p}_P(s^*)$  ist hier der Pfadpositionsvektor mit dem kleinsten Abstand zum Vorderradachsemittelpunkt.

$$\underline{v} := [\cos(\theta) \quad \sin(\theta)]^T, \quad (2.7a)$$

$$\underline{v}_\perp := [\cos(\theta + \pi/2) \quad \sin(\theta + \pi/2)]^T, \quad (2.7b)$$

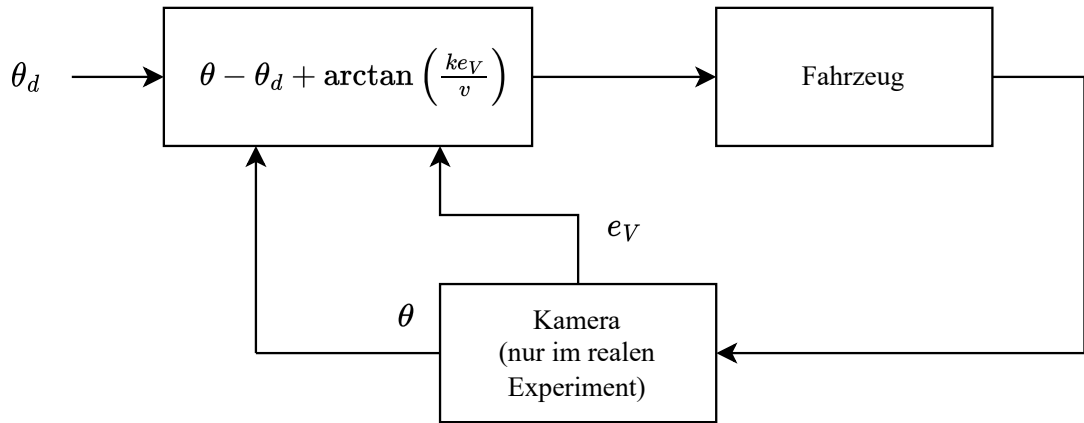
$$s^* := \arg \min_s \|\underline{p}_P(s) - \underline{p}_V\|_2, \quad (2.7c)$$

$$\underline{\lambda} := \underline{p}_P(s^*) - \underline{p}_V, \quad (2.7d)$$

$$e_V := \underline{\lambda} \cdot \underline{v}_\perp. \quad (2.7e)$$

Die Pfadausrichtung und die Querabweichung werden dann an den Stanley-Regler übergeben, und der Ausgang des Reglers wird in das Fahrzeugmodell eingespeist. Ein Übersichtsdiagramm des Regelkreises ist in Abb. 5 dargestellt.

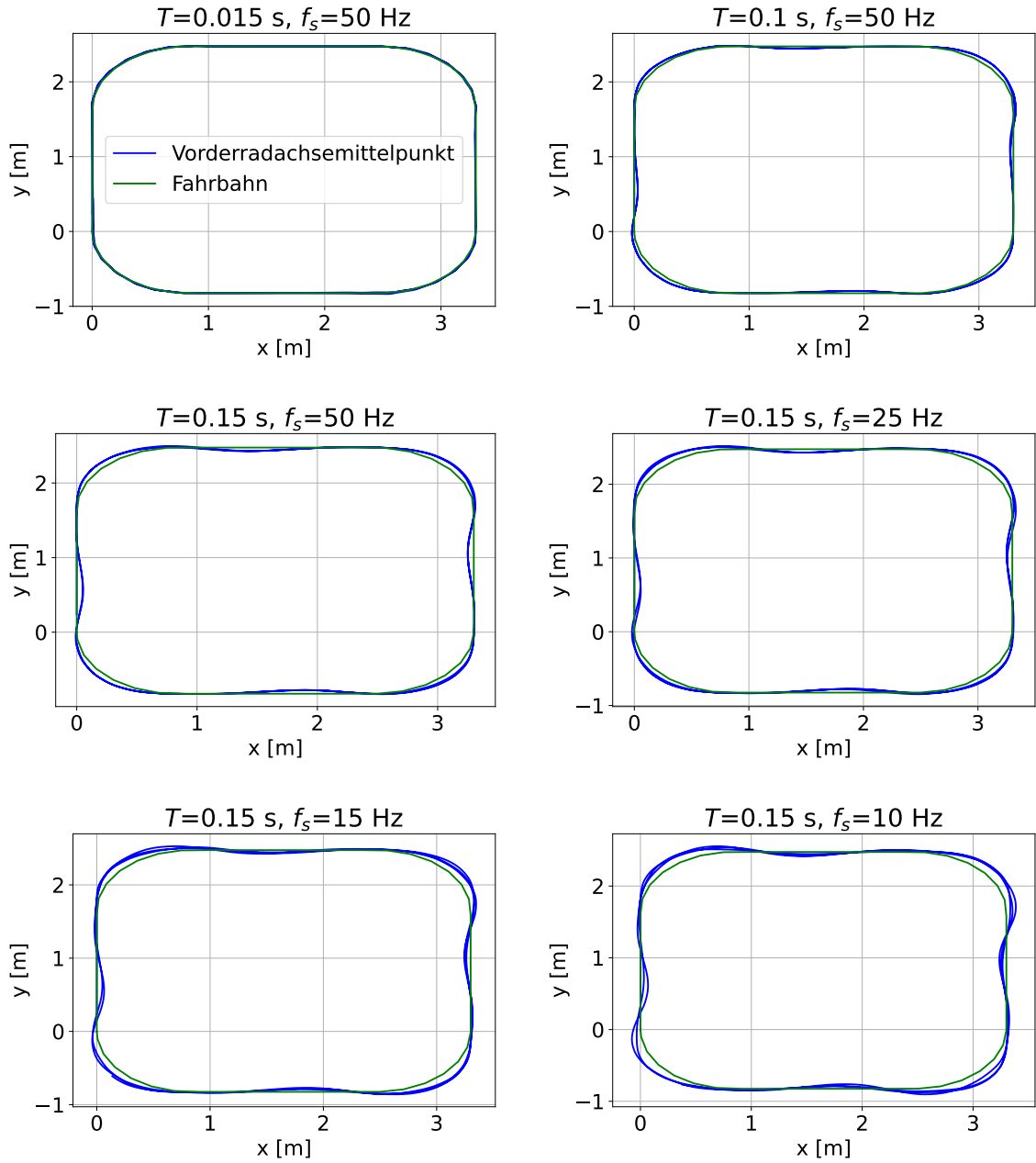
Der nominelle Stanley-Regler (2.1) arbeitet mit zeitkontinuierlichen Signalen, die sich von den vom „echten“ Fahrzeug verwendeten zeitdiskreten Signalen unterscheiden. Das Fahrzeug nimmt Bilder mit einer festen Abtastfrequenz auf, die von der Pipeline verarbeitet werden, bevor sie in den Regler eingespeist werden (näheres dazu in den Kapiteln 3 und 4). Um dieses Verhalten zu simulieren, wird ein Speicher in den Simulator eingebaut, um die Ausrichtung und die Querabweichung zu speichern. Der Stanley-Regler



**Abbildung 5** – Darstellung des Regelkreises.

verwendet dann diese zwischengespeicherten Werte für eine bestimmte Dauer. Dieser Speicher-Ansatz ahmt das Verhalten eines Halteglieds nullter Ordnung nach, und die Ausgabe dieses Haltegliedes wird anschließend in den Stanley-Regler eingespeist. Dieser Simulationsansatz ermöglicht die Untersuchung des Fahrzeugverhaltens für verschiedene Abtastfrequenzen. Von besonderer Bedeutung ist die Stabilität des Regelkreises mit verschiedenen Abtastfrequenzen. Der Vergleich der Simulationsergebnisse bei verschiedenen Abtastfrequenzen ist in Abb. 6 dargestellt.

Wie in Abb. 6 ersichtlich, wird der Stanley-Regler durch die Abtastfrequenz beeinflusst, wobei bei niedrigeren Frequenzen ein deutlicher Effekt zu beobachten ist. Oberhalb eines Schwellenwerts von 15 Hz scheint das Verhalten des Fahrzeugs jedoch nicht wesentlich durch die Abtastfrequenz beeinflusst zu werden.



**Abbildung 6** – Verhalten des Stanley-Reglers bei verschiedenen Abtastfrequenzen und Zeitkonstanten. Das Verhalten bleibt ähnlich bei Frequenzen unter 15 Hz. Der Stanley-Regler folgt den Pfad besser wenn die Zeitkonstante des PT1-Glieds niedrig ist, aber die Lenkwinkelstrecke im Auto hat eine Zeitkonstante zwischen 100 und 200 ms.

# Kapitel 3

## Bildverarbeitung

Im Vergleich zu dem vorher verwendeten PID-Regler benötigt der Stanley-Regler zusätzliche Informationen, nämlich die Ausrichtung des Pfades. Die neue Bildverarbeitungspipeline ist darauf ausgelegt, diese notwendigen Informationen zu extrahieren. Der beschriebene Prozess basiert auf [6]. Wie in Abb. 7 zu sehen, besteht die Pipeline aus mehreren Schritten zur Berechnung der Ausrichtung und der Querabweichung. Im Folgenden werden die einzelnen Schritte der Bildverarbeitung genauer beschrieben.

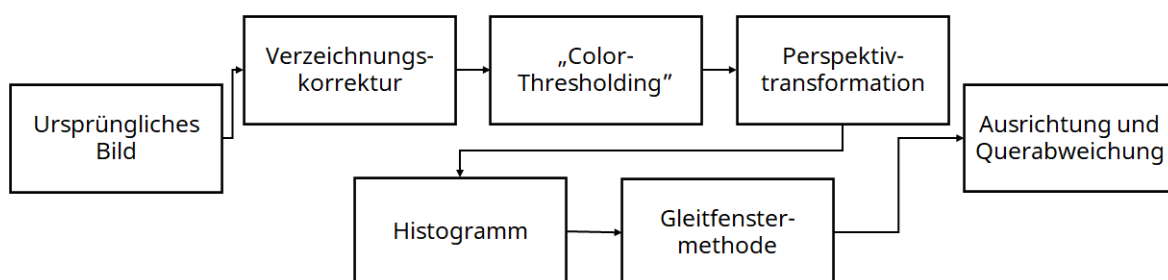
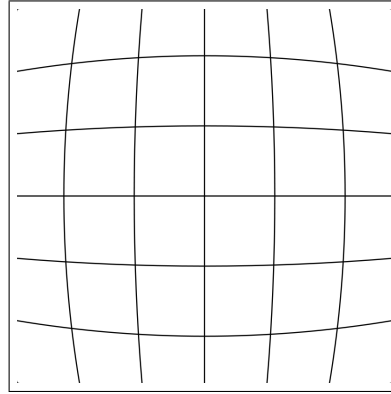


Abbildung 7 – Grafische Darstellung des Bildverarbeitungspipelines.

### 3.1 Verzeichnungskorrektur

Für die CoRoLa Car Platform wurde eine Fischaugenkamera gewählt. Der Vorteil einer Fischaugenkamera ergibt sich aus ihrem größeren Blickwinkel im Vergleich zu einer normalen Kamera. Eine Fischaugenkamera hat eine sog. „tonnenförmige Verzeichnung“ [10], wodurch gerade Linien gekrümmt werden. Die Krümmung dieser Linien hängt von ihrem radialen Abstand vom Bildmittelpunkt ab. Ein Beispiel ist in Abb. 8 zu sehen. Diese Verzeichnung kann mit Hilfe der Software [4] auf Grundlage einer Kamerakalibrierung kompensiert werden.





**Abbildung 8** – Ein Beispiel einer tonnenförmigen Verzeichnung.

Diese ist notwendig, um die extrinsischen und intrinsischen Parameter der Kamera näherungsweise zu bestimmen.

### 3.1.1 Kalibrierung

Die intrinsischen Parameter einer Kamera werden häufig durch die 3x3-Matrix

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.1)$$

„Kameramatrix“ dargestellt, wobei  $f_x$  und  $f_y$  die Brennweiten der Kamera in Pixeln in den Richtungen  $x$  und  $y$  sind und  $c_x$  und  $c_y$  die Koordinaten des Bildmittelpunkts bezeichnen. Die extrinsischen Parameter werden häufig durch die folgende Matrix dargestellt:

$$\begin{bmatrix} R & T \end{bmatrix}. \quad (3.2)$$

Dabei ist  $R$  die Rotationsmatrix der Kamera in Bezug auf das dem Experiment zugrunde liegenden Bezugskordinatensystem und  $T$  der Positionsspaltenvektor des Ursprungs des Bezugssystem, ausgedrückt in den Koordinaten des Kamerabildes. Die „Kameratransformationsmatrix“  $M$  ist die Abbildung von den Weltkoordinaten auf Pixelkoordinaten, die durch die Matrix-Matrix-Multiplikation dargestellt wird,

$$M = K \begin{bmatrix} R & T \end{bmatrix}. \quad (3.3)$$

Durch das Rekonstruieren der  $M$ -Matrix mit Hilfe von Software [4] kann das verzerrte Bild wieder in das Bezugssystem des Experiments überführt werden, wodurch gekrümmte Linien gerade werden.

Im Einzelnen läuft der Prozess folgendermaßen ab: eine Kamera wird anhand einer Sammlung von Fotos mit bekannten geraden Linien kalibriert. In der Regel wird eine

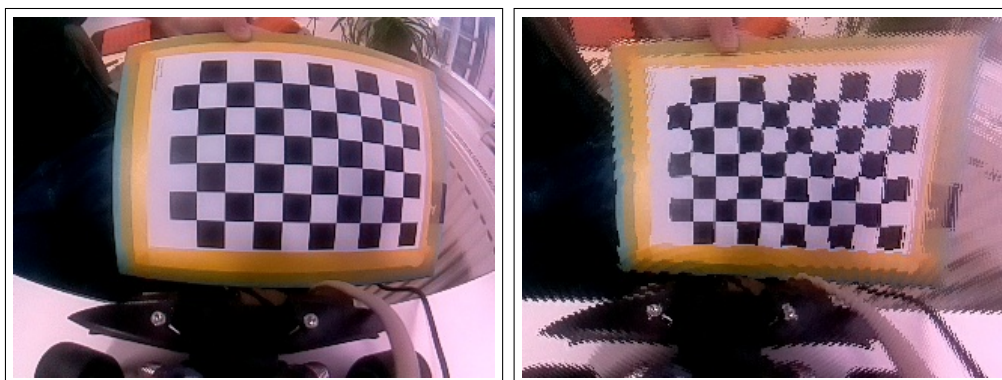
Reihe von Schachbrettbildern mit bekannten Dimensionen verwendet. [5] Danach werden Fotos des Schachbretts in verschiedenen Winkeln und in verschiedenen Positionen relativ zur Kamera aufgenommen. Diese Bilderserie wird dann in den Algorithmus zur Kamerakalibrierung eingespeist, der zunächst die Positionen der Schachbrettfelder und der sie verbindenden Linien bestimmt. Die Anzahl der benötigten Bilder hängt von der jeweiligen Kamera ab, eine große Sammlung von Fotos führt jedoch zu einer genaueren Annäherung an die wirklichen Parameter. Die Verwendung von Bildern mit einer höheren Auflösung führt ebenfalls zu genaueren Parametern. [5] Die angenäherten Parameter sind jedoch nur gültig für die Verarbeitung von Bildern, die mit der gleichen Auflösung aufgenommen wurden wie die für die Kalibrierung verwendeten Bilder. Um die Kameramatrix für Bilder mit einer niedrigeren Auflösung zu verwenden, muss die intrinsische Kameramatrix  $K$  mit der folgenden Formel skaliert werden:

$$K_n = kK, \quad (3.4)$$

wobei  $k$  ein positiver skalarer Wert ist, der den Skalierungsfaktor darstellt [2]. Da es sich bei der intrinsischen Kameramatrix um eine affine Abbildung handelt, muss der Wert bei  $K_{n,3,3}$  auf 1 gesetzt werden. Die resultierende intrinsische Kameramatrix funktioniert bei kleineren Auflösungen, die dem Seitenverhältnis der für die Kalibrierung verwendeten Bilder gleich sind.

### 3.1.2 Korrektur

Nach der erfolgten Kalibrierung gleicht der Algorithmus anhand des Modells der Fischaugenkamera die Verzeichnung aus. Wie in Abb. 9 zu sehen, sind beispielsweise die gekrümmt abgebildeten Linien des Schachbretts nach der Kalibrierung wieder begradigt.

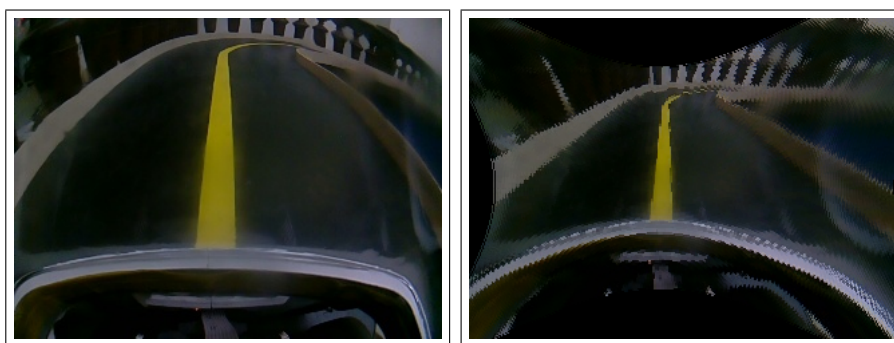


**Abbildung 9** – Die Kalibrierung des linken Bilds liefert das rechte Bild. Die gekrümmte Linie des Schachbretts sind gerade nach dem Verzeichnungskorrektur. Der Grund für die wellenförmigen Verzerrungen ist die nicht perfekte Kalibrierung.

Ein Nachteil der Verzeichnungskorrektur ist, dass jedes korrigierte Bild eine geringere Auflösung hat als das Originalbild. Dies ist eine Folge des Korrektursprozess da dieser

einen Teil des Bildes verzerrt, insbesondere die Pixel nahe den Ecken des Bildes. Daher werden diese Pixel beim Korrektursprozess aus dem resultierenden Bild entfernt. Um das Bild in seiner ursprünglichen Auflösung zu rekonstruieren, wird eine Interpolation verwendet. Die Interpolation liefert jedoch ein unschärferes Bild als das Original. Um dies zu kompensieren, empfiehlt es sich grundsätzlich, eine Kamera mit hochauflösenden Bildern zu kalibrieren und dann Bilder mit dieser Auflösung aufzunehmen. Anstatt den Interpolator zu verwenden, verkleinert man die Bilder dann auf eine Auflösung, die für die jeweilige Anwendung ausreichend ist. Das Ergebnis ist ein genaueres Bild ohne Unschärfe. Leider ist dieser Prozess sehr rechenintensiv, und es wurde beschlossen, nur die Bilder aus dem Interpolator zu verwenden, um die Rechenleistung der Pipeline zu erhöhen.

Ein weiterer Nachteil der Verzeichnungskorrektur ist, dass sich der Mittelpunkt der Kamera verschieben kann. Ein Beispiel dafür ist in Abb. 10 zu sehen. Durch manuelles Ändern von  $c_x$  kann der Bildmittelpunkt wieder an seine ursprüngliche Position verschoben werden.



**Abbildung 10** – Ein Beispiel von dem Korrekturschritt. Die Mittellinie im rechten Bilds ist leicht nach links verschoben.

## 3.2 „Color-Thresholding“

Als Nächstes folgt der Schritt des sogenannten „Color-Thresholding“. In diesem Schritt werden alle Pixel aus dem Bild entfernt, die nicht zur Bahnlinie gehören. In dieser Implementierung sind alle Farbe außer Gelb unerwünscht.

Zuerst wird das Bild von dem RGB in den HSV-Farbraum (Hue, Saturation, Value engl. für Farbwert, Sättigung und Helligkeitswert) umgewandelt.

Im RGB-Farbraum ist reines Gelb so definiert, dass sowohl der rote als auch der grüne Kanal gleich sind und der blaue Kanal auf Null gesetzt ist. Dies lässt nur einen Freiheitsgrad für die Justierung der Pipeline zu. Ein einziger Freiheitsgrad führt zu Problemen bei der Abstimmung, da die Pipeline dadurch weniger in der Lage ist,

Schwankungen zu berücksichtigen. Unterschiedliche Lichtverhältnisse oder reflektierende Oberflächen sind Beispiele für solche Störungen.

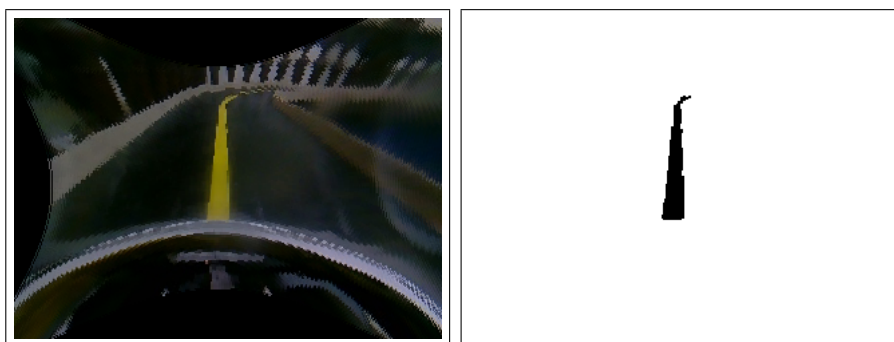
Im HSV-Farbraum bestimmt der Farbwertkanal die Farbe, der Sättigungskanal die Reinheit des Farbtons und der Wertekanal die Helligkeit des Farbtons. [3] Nach der Auswahl des Intervalls von Farbwerten, in diesem Fall innerhalb des gelben Bereichs, werden der Helligkeits- und der Sättigungskanal verwendet, um den spezifischen Gelbton auszuwählen. Die Verwendung dieser beiden Kanäle ermöglicht eine zuverlässigere Erkennung der Farbe.

Um die gelbe Spur zu erkennen, filtert die Pipeline Farben außerhalb des gelben Farbtonbereichs heraus und schneidet dann den unteren Teil des Helligkeits- und Sättigungskanals ab. Das Ergebnis ist, dass nur reines Gelb im Bild übrig bleibt. Die in der Pipeline verwendeten Werte für Gelb werden in Tab. 1 dargestellt.

	H	S	V
Minimaler Wert	15	90	90
Maximaler Wert	40	255	255

**Tabelle 1** – Minimalen und Maximalen Werte für die Bestimmung des gelben Farbtonbereiches. Der Bereich der Werte sind zwischen 0 und 255 begrenzt.

Das Ergebnis dieses Schritts der Pipeline ist für ein Beispielbild in Abb. 11 dargestellt. Um die Darstellung zu vereinfachen, werden die gelben Pixel hier und im Folgenden als schwarze Pixel auf weißem Grund dargestellt. Als Eingabe bekommt die Pipeline ein kalibriertes Bild und gibt dann nur Pixel aus, die zur Bahnlinie gehören.



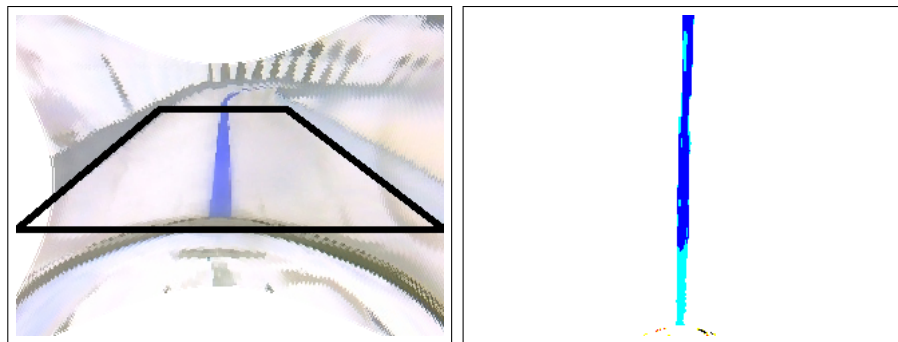
**Abbildung 11** – Ein Beispiel von der Ausgabe des Color-Thresholdings. Links: Bild nach dem Kalibrierungsschritt, Eingang zum „Color-Thresholding“. Rechts: binäres Bild mit nur Pixeln, die als gelb erkannt wurden.

### 3.3 Perspektivtransformation

Der dritte Teil der Bildverarbeitungspipeline ist die Perspektivtransformation. Die Kamera steht nicht senkrecht zur Bahn, wie es bei einer idealen Draufsicht der Fall wäre, sondern die Blickrichtung bildet einen Winkel von ca.  $30^\circ$  zur Bahnebene.

Wenn ein Bild mit einer am Fahrzeug montierten Kamera aufgenommen wird, ist die Fahrbahn nicht rechteckig, sondern trapezförmig. Diese Perspektive erfordert, dass alle Berechnungen bezüglich der Fahrspurlinie deren scheinbar abnehmende Breite kompensieren müssen. Um dies zu vermeiden, wird eine Perspektivtransformation durchgeführt.

Bei der Perspektivtransformation wird eine Teilmenge des Bildes ausgeschnitten und so angepasst, dass sie das gesamte Bild füllt. Im allgemeinen Fall muss diese Scherung nicht das gesamte Bild umfassen, sondern sie kann in einen anderen Bildausschnitt übertragen werden. Ein Beispiel ist in Abb. 12 dargestellt. Mithilfe dieser Perspektivtransformation wird die trapezförmige Form der Fahrbahnlinie zu einer rechteckigen Form korrigiert.

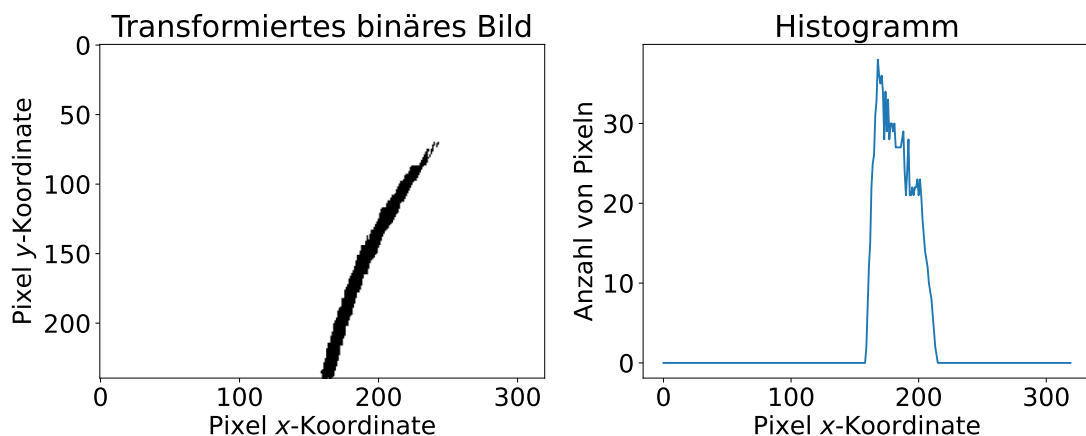


**Abbildung 12** – Perspektivtransformation für ein Beispielbild. Die Trapez im linken Bild wird ausgeschnitten und gestreckt, sodass sich das rechte Bild ergibt. Zu diesem Zeitpunkt hat noch kein „Color-Thresholding“ stattgefunden.

Eine Folge dieser neuen Perspektive ist, dass das resultierende Bild einer Draufsicht ähnelt. Die Perspektivtransformation vereinfacht auch die weitere Bildverarbeitung, da alle Objekte außerhalb des Trapezes abgeschnitten werden und nur die Spur übrig bleibt. Wie in Abb. 12 zu sehen ist, führt die Perspektivtransformation jedoch zu farbliehen Artefakten im Bild. Daher ist es erforderlich, dass dieser Schritt nach dem „Color-Thresholding“-Schritt erfolgt.

### 3.4 Histogramm

Nach der Perspektivtransformation wird ein Histogramm der schwarzen Pixel erstellt, um den Startpunkt für die im nächsten Schritt folgende Gleitfenstermethode zu bestimmen. Die Anzahl der schwarzen Pixel wird für jede Spalte des Bildes gezählt, und diese Zahlen werden in einer Liste gespeichert. Es wird davon ausgegangen, dass die Spalten mit den höchsten Zahlen Fahrspurinformationen enthalten, und der Index der Spalte mit der höchsten Zahl wird als Startpunkt für die Gleitfenstermethode gewählt. Dieser Schritt trägt dazu bei, die Berechnungszeit zu verkürzen, indem Teile des Bildes herausgefiltert werden, die keine Fahrspurinformationen enthalten. Eine visuelle Darstellung des Histogramms ist in Abb. 13 zu sehen.



**Abbildung 13** – Histogramm der schwarzen Pixel. Die Spitze des Histogramms liegt bei dem  $x$ -Schnittpunkts.

### 3.5 Gleitfenstermethode

Im fünften Schritt der Pipeline wird die Form der Fahrspur bestimmt. Dazu soll in einem vorbereitenden Schritt das Bild bereinigt werden, indem Pixel gelöscht werden, die klar nicht zu der gesuchten Fahrspur gehören. Dies wird durch die Verwendung der Gleitfenstermethode erreicht, die in [6] beschrieben ist.

Dabei wird eine ausgewählte Anzahl von Rechtecken (Fenstern) erstellt. Die Höhe wird so gewählt, dass die Summe der Höhe aller Rechteck gleich der Bildhöhe ist, während ihre Breite entsprechend proportional zur Bildbreite gewählt wird. Die Anzahl der Fenster wird so gewählt, dass der aus dem Histogramm bekannte Bereich der schwarzen Pixel umfasst wird, wie in Abb. 14 zu sehen ist.

Als zweiter Schritt wird die  $x$ -Koordinate des Mittelpunkts des ersten Fensters auf

den Spitzenwert des Histogramms aus dem vorherigen Schritt gesetzt. Dann wird die Anzahl der schwarzen Pixel innerhalb des Fensters gezählt. Wenn die Anzahl über einem gewählten Schwellenwert liegt, werden die Pixel zu einem Array hinzugefügt und der Mittelwert der  $x$ -Koordinaten dieser Pixel berechnet. Die  $x$ -Koordinate des nächsten Fensters wird auf diesen berechneten Mittelwert gesetzt und um die Breite der Fenster in  $y$ -Richtung verschoben. Dieser Vorgang wird dann für alle verbleibenden Fenster wiederholt. Wenn in einem Fenster nicht genügend schwarze Pixel enthalten sind, wird das Fenster ignoriert und das folgende Fenster darüber gelegt.

Die Pixel, die nicht innerhalb der einzelnen Fenster liegen, werden dann aus dem Bild herausgefiltert. Das Bild vor und nach dem Filterungsprozess ist in Abb. 14 zu sehen. Mit Hilfe der Methode der kleinsten Quadrate wird ein Polynom 2. Ordnung durch die verbleibenden Pixel approximiert.

Um das Polynom, das die Fahrspur repräsentiert, aus den  $n$  verbleibenden Pixeln zu bestimmen, wird die folgende Berechnung [11] verwendet:

$$\underline{x} := \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}^T \quad (3.5a)$$

$$\mathbf{Y} := \begin{bmatrix} 1 & y_1 & y_1^2 \\ 1 & y_2 & y_2^2 \\ \vdots & \vdots & \vdots \\ 1 & y_n & y_n^2 \end{bmatrix} \quad (3.5b)$$

$$\underline{\hat{\beta}} := \begin{bmatrix} \hat{\beta}_0 & \hat{\beta}_1 & \hat{\beta}_2 \end{bmatrix}^T \quad (3.5c)$$

$$\underline{\hat{\beta}} = (\mathbf{Y}^T \mathbf{Y})^{-1} \mathbf{Y}^T \underline{x}, \quad (3.5d)$$

wobei  $\underline{x}$  und  $\underline{y}$  die Vektoren sind, die die  $x$ - und  $y$ -Koordinaten der übrigen Pixel enthalten.  $\hat{\beta}_0$ ,  $\hat{\beta}_1$  und  $\hat{\beta}_2$  sind die Koeffizienten des approximierten Polynoms.

Unter Verwendung der approximierten Koeffizienten erhält man die folgende Gleichung:

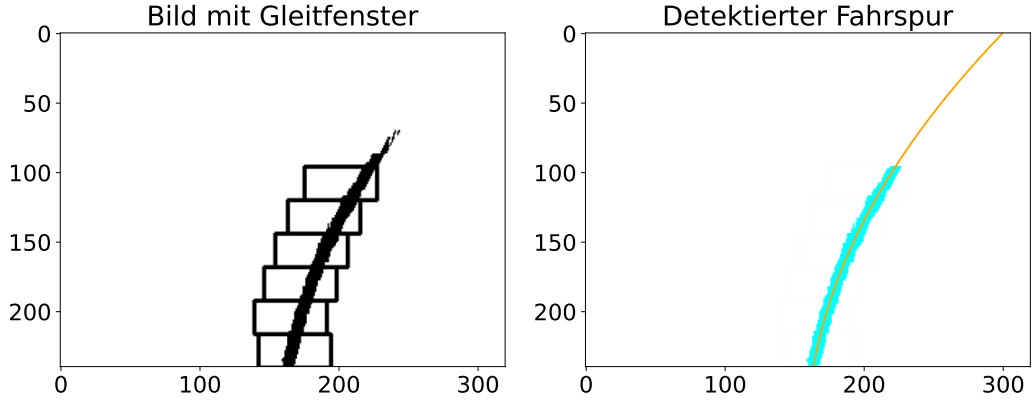
$$\hat{x} = \hat{\beta}_0 + \hat{\beta}_1 y + \hat{\beta}_2 y^2, \quad (3.6)$$

wobei  $y$  die  $y$ -Koordinate und  $\hat{x}$  die angenäherte  $x$ -Koordinate der Fahrspur ist. Die Gleichung liefert eine analytische Darstellung der Fahrspur.

## 3.6 Berechnung von Ausrichtung und Querabweichung

In dem letzten Schritt der Pipeline werden die Ausrichtung des Fahrzeugs und die Querabweichung von der Fahrspurlinie berechnet.





**Abbildung 14** – Das linke Bild ist ein Beispiel von der Gleitfenstermethode. Das rechte Bild zeigt das Polynom, das von den verbleibenden Pixel approximiert wurde.

Aus dem in dem vorangegangenen Schritt berechneten Polynom (3.6) wird eine analytische Ausrichtung an einem ausgewählten Punkt entlang der Bahn mit der folgenden Gleichung [12] ermittelt:

$$\hat{x} = g(y), \quad (3.7a)$$

$$\theta = \arctan(g'(y)). \quad (3.7b)$$

Der Arkustangens der Ableitung des Polynoms liefert die Ausrichtung der Spur an einem bestimmten Punkt. Für den Stanley-Regler wird nur die Differenz zwischen der Bahnausrichtung und der Ausrichtung des Fahrzeugs benötigt (2.1). Da kein globales Koordinatensystem für das Fahrzeug benötigt wird, kann die Ausrichtung des Fahrzeugs zu jedem Zeitpunkt als  $0^\circ$  definiert werden. Die aus dem Bild erkannte Ausrichtung  $\theta$  der Bahn bildet somit direkt die für den Stanley-Regler relevante Winkel-Eingangsgröße.

Die Abweichung von der Fahrspurlinie wird aus dem angepassten Polynom mit folgender Gleichung [6] berechnet:

$$e_V = \left( \frac{w}{2} - g(h) \right) \cdot \sigma_{\text{mpp}}. \quad (3.8)$$

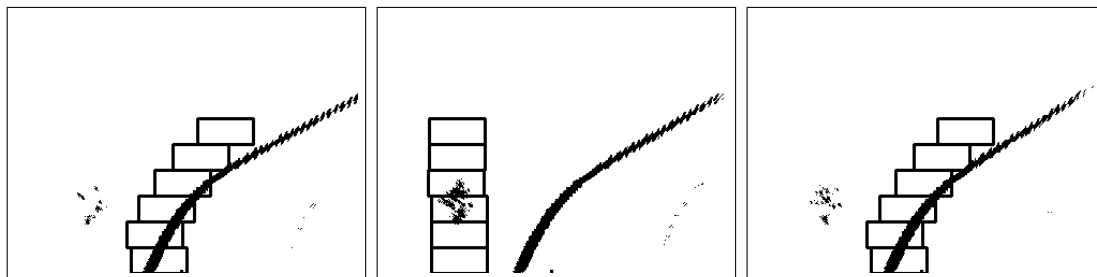
Dabei ist  $w$  die Breite des Bildes,  $e_V$  der Abstand des Fahrzeugs zur Fahrspur,  $\sigma_{\text{mpp}}$  eine Skalierungskonstante zur Umrechnung von Pixeln in Meter und  $g(h)$  die Ausgabe des berechneten Polynoms am unteren Rand des Bildes, d.h. bei  $y = h$ . Zusammenfassend wird die Differenz aus der  $x$ -Komponente des Mittelpunkts und der  $x$ -Komponente des Polynompunkts am unteren Bildrand von Pixeln in Meter umgerechnet.

### 3.7 Ausreißer

Fahrspurerkennungsalgorithmen können zu einem Phänomen führen, das als „Ausreißer“ bekannt ist. Diese treten auf, wenn der Algorithmus die Fahrspurlinie in einem



bestimmten Bild falsch erkennt. Im Folgenden wird dieses am Beispiel dieser Pipeline diskutiert. Abb. 15 zeigt eine Sequenz von drei Einzelbildern, wobei das erste Einzelbild (links) den Ausgangspunkt bildet. Im ersten Bild wird die Fahrspurlinie richtig erkannt, aber im zweiten Bild tritt ein Ausreißer auf, bei dem die Fahrspurlinie falsch bestimmt wird. Im letzten Bild wird die Fahrspurlinie wieder korrekt erkannt.



**Abbildung 15** – Ein Beispiel von einem Ausreißer. Die drei Bilder wurden nacheinander aufgenommen. Im rechten und linken Bilder wurde die Fahrspur richtig erkannt. Im mittleren Bild wurde die Fahrspur falsch erkannt.

Eine der Hauptursachen für Ausreißer ist die falsche Auswahl der HSV-Werte im Color-Thresholding-Schritt. Wenn die im Algorithmus verwendeten HSV-Werte nicht restriktiv genug sind, werden Farben außer Gelb möglicherweise nicht herausgefiltert, was zu einer ungenauen Bestimmung der Fahrspurlinie führt.

Ausreißer sind ein unvermeidbares Resultat des Bildverarbeitungsprozesses. Die Empfindlichkeit mit der ein Regelungsalgorithmus auf sie reagiert ist ein wichtiges Kriterium bei der Wahl eines Algorithmus. Wie in Kapitel 4 genauer beschrieben, erschien in den hier durchgeführten Tests der Stanley-Regler in dieser Hinsicht weniger empfindlich als der PID-Regler.

# Kapitel 4

## Experimentelle Untersuchungen

### 4.1 Hardware

Die CoRoLa-Car-Plattform besteht aus einem busförmigen Modellfahrzeug, in [16](#) dargestellt, mit einem Raspberry Pi, einer Fisheye-Kamera, einem bürstenlosen Gleichstrommotor und einem Motorregler zur Steuerung der Fahrzeuggeschwindigkeit und einem Servomotor um den Radeinschlag des Fahrzeugs zu steuern. Der Raspberry Pi sendet Power-Werte-Signale an den Motorregler und den Servoregler, um das Fahrzeug zu steuern.

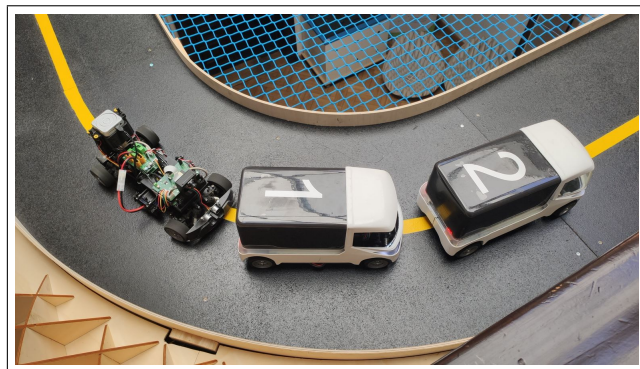


Abbildung 16 – Modellfahrzeuge.

### 4.2 Experimentkriterien

#### 4.2.1 Höchstgeschwindigkeit

Um die Höchstgeschwindigkeit zu messen, fährt das Fahrzeug mit einer bestimmten Vorwärtsgeschwindigkeit auf der Strecke. Wenn das Verhalten des Fahrzeugs als „akzeptabel“ angesehen wird, wird das Fahrzeug angehalten und die Geschwindigkeit erhöht.

Dieser Vorgang wird so lange wiederholt, bis das Verhalten des Fahrzeugs nicht mehr „akzeptabel“ ist. Das „akzeptable“ Verhalten des Fahrzeugs wird empirisch anhand der folgenden beiden Kriterien definiert: Fähigkeit, der Fahrspur grundsätzlich zu folgen, d.h. nicht mit der Randbegrenzung zu kollidieren, und Ausbleiben von beobachteten Schwingungen.

### **4.2.2 Maximal möglicher Fehlerwinkel**

Die maximalen Fehlerwinkel sind definiert als die maximalen Winkel, bei denen das Fahrzeug in der Lage ist, den Weg links bzw. rechts der Fahrspurlinie wiederzufinden. Um dies zu messen, wird das Fahrzeug bei deaktiviertem Motorregler parallel zur Fahrspurlinie platziert. Dann wird das Fahrzeug im Uhrzeigersinn gedreht, wobei der Ausgang des jeweiligen Reglers in Echtzeit angezeigt wird. Das Fahrzeug wird so lange gedreht, bis der Ausgang des Reglers entweder den plausiblen Gültigkeitsbereich verlässt oder konstant Null ist. In beiden Fällen ist das Fahrzeug nicht mehr in der Lage die Spur zu wiederzufinden.

### **4.2.3 Bahnfolgeverhalten**

Die Messung des Fahrzeugversatzes erfolgt durch die folgenden zwei Versuche. Zunächst fährt das Fahrzeug mit einer bestimmten Vorwärtsgeschwindigkeit auf der Strecke. Dabei wird mit einer Überwachungskamera ein Video von der Kurvenfahrt des Fahrzeugs aus der Vogelperspektive aufgenommen. Das Fahrzeug wird bei unterschiedlichen Vorwärtsgeschwindigkeiten aufgezeichnet. Bei der ersten Aufnahmerunde wird das Fahrzeug mit dem Stanley-Regler gesteuert, bei der zweiten mit dem PID-Regler. Dieses Experiment wird für die gewählten Geschwindigkeiten von 1,0, 1,25, 1,5, 1,75 und 2,0 Metern pro Sekunde durchgeführt.

### **4.2.4 Reglerausgang**

Um den Reglerausgang der beiden Regler zu messen, wird das Fahrzeug dreimal die vollständige Strecke abfahren gelassen. Während der Fahrt wird der Ausgang des Reglers aufgenommen. Nach der Fahrt werden die Daten im Zeitverlauf grafisch dargestellt und der zeitliche Mittelwert der Daten angegeben. Dies ist sinnvoll, da für das Fahren von Kurven immer ein bestimmtes Stellsignal für den Lenkwinkel anliegen muss. Zu erwarten wäre ein periodischer Signalverlauf, da es sich um eine geschlossene Bahn handelt.

## 4.3 Ergebnisse

### 4.3.1 Höchstgeschwindigkeit

Wenn der PID-Regler für das Modellfahrzeug verwendet wird, führen Geschwindigkeiten von mehr als  $1,5 \frac{\text{m}}{\text{s}}$  zu einer geringeren Robustheit. Im Gegensatz zum Stanley-Regler bringt der PID-Regler das Fahrzeug zum Schwingen, wenn ein Ausreißer auftritt, und benötigt mehr Zeit, um auf die Spur zurückzukehren. Das Schwingungsverhalten nimmt mit der Vorwärtsgeschwindigkeit des Fahrzeugs zu. Wenn die Geschwindigkeit  $2,0 \frac{\text{m}}{\text{s}}$  übersteigt, steuert der PID-Regler das Fahrzeug gegen die Wand des Streckenaufbaus. Selbst Geschwindigkeiten zwischen  $1,5$  und  $1,75 \frac{\text{m}}{\text{s}}$  erfordern eine Überwachung durch einen Menschen, um Kollisionen zu vermeiden. Dieser muss das Fahrzeug vor der Spurwand manuell mit einem Notaus stoppen.

Im Vergleich dazu benötigt der Stanley-Regler keine Überwachung, bis die Vorwärtsgeschwindigkeit des Fahrzeugs  $2,3 \frac{\text{m}}{\text{s}}$  erreicht. Bei höheren Geschwindigkeiten weicht das Fahrzeug jedoch immer weiter von der gewünschten Bahn ab. Dabei oszilliert es um die gelbe Fahrspur ohne je wieder für einen längeren Zeitraum genau auf die Spur zu gelangen.

### 4.3.2 Maximal möglicher Fehlerwinkel

Die Ergebnisse des Experiments sind in Tab. 2 zu sehen. Der Stanley-Regler ist im Vergleich zum PID-Regler weniger tolerant gegenüber einer Abweichung von der Fahrspur. Dies liegt an dem Kalibrierungsschritt der Pipeline und der Anforderung eines approximierten Polynoms, was es für den Stanley-Regler schwieriger macht, die Fahrspur zu erkennen. Im Gegensatz dazu benötigt der PID-Regler nur eine Gruppe von gelben Pixeln im Bild.

	PID (°)	Stanley (°)
Uhrzeigersinn	$75 \pm 1$	$50 \pm 1$
Gegenuhrzeigersinn	$54 \pm 1$	$44 \pm 1$

**Tabelle 2** – Maximal möglicher Fehlerwinkel beider Regler, wenn das Fahrzeug im oder gegen den Uhrzeigersinn gedreht wird.

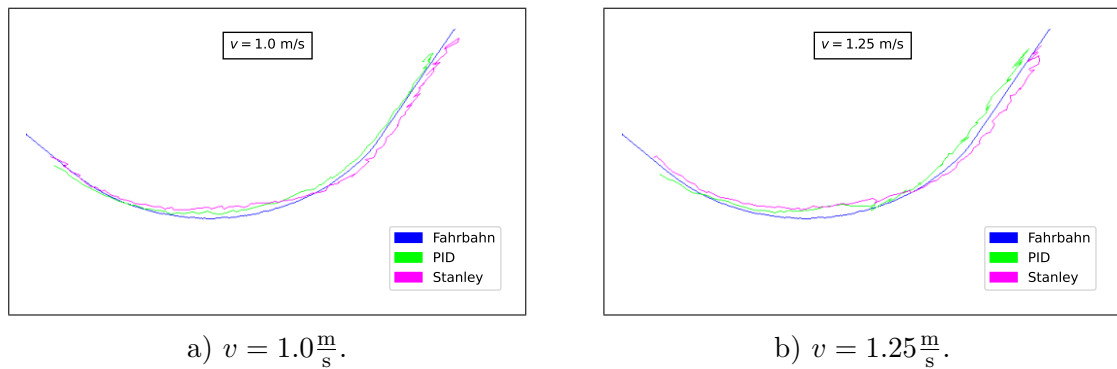
Der Stanley-Controller benötigt für die Gruppierung der gelben Pixel eine bekannte Geometrie, typischerweise in Form einer Linie oder eines Kurvenabschnitts, während der PID-Controller keine solchen Voraussetzungen hat.

### 4.3.3 Bahnfolgeverhalten

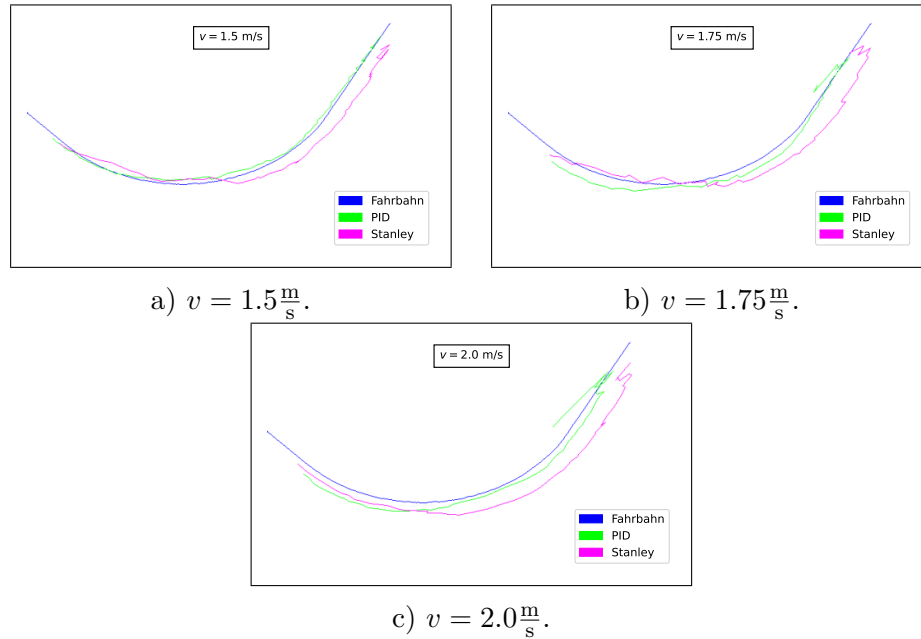
Abb. 17a und 17b zeigen, dass die beiden Regler sich bei niedrigeren Vorwärtsgeschwindigkeiten ähnlich verhalten. Aber Abb. 18a, 18b und 18c zeigen, dass der PID-Regler bei höheren Vorwärtsgeschwindigkeiten die Bahn genauer verfolgt als der Stanley-Regler. Selbst unter Berücksichtigung des anfänglichen Versatzes bringt der Stanley-Regler das Fahrzeug erst am Ende der Kurve auf die Fahrspur, während der PID-Regler das Fahrzeug mit einer leichten nach links driftenden Bewegung, die mit der Vorwärtsgeschwindigkeit skaliert, auf der Bahn hält.

Die Ergebnisse der Simulation des Stanley-Reglers in Abb. 6 mit einer Vorwärtsgeschwindigkeit von  $1,75 \frac{\text{m}}{\text{s}}$  und einer Abtastfrequenz von 50 Hz stimmen mit dem Verhalten des Stanley-Reglers auf der tatsächlichen Strecke weitgehend überein. Allerdings driftet der Stanley-Regler, wie in der Abb. 18c gezeigt, vor der Kurve vom geraden Streckenabschnitt weg. Dieses Phänomen wird als „Drift“ bezeichnet.

Im Rahmen der Arbeit wurde versucht, diesen Drift zu kompensieren, aber trotz der Anpassung des Bezugspunkts für die Querabweichung blieb der Drift bestehen. Eine manuelle Änderung des Referenzpunktes führte dazu, dass das Fahrzeug bei der Fahrt im oder gegen den Uhrzeigersinn um die Strecke unterschiedliche Parameter benötigte, was nicht praktikabel war. Daher wurde der Drift vorerst so belassen, wie er ist. In Kapitel 5 werden mögliche Anpassungen des Stanley-Reglers erörtert, die seine Leistung in Zukunft verbessern könnten.



**Abbildung 17** – Gezeigt ist der vom Fahrzeug gewählte Pfad (Fahrtrichtung von rechts nach links) für die Vorwärtsgeschwindigkeiten  $v = 1.0 \frac{\text{m}}{\text{s}}$  und  $v = 1.25 \frac{\text{m}}{\text{s}}$ . Dabei wurde der Standort des Fahrzeugs nicht immer ganz korrekt erfasst, was zu den sichtbaren Zacken führt. Es ist sichtbar, dass der PID- und der Stanley-Regler ungefähr das gleiche Verhalten haben.



**Abbildung 18** – Das Modellfahrzeug mit dem Stanley-Regler hat eine kleine Abweichung am Anfang der Kurve (rechts). Dabei handelt es sich um den in Kapitel 4 beschriebenen Drift. Am Ende der Kurve ist es aber auf der Fahrspur.

#### 4.3.4 Reglerausgang

Die Ergebnisse des Experiments sind in Tab. 3 aufgeführt. Die durchschnittliche Reglerausgabe beider Regler ist ähnlich, unabhängig von der Geschwindigkeit, außer für den PID-Regler bei einer Geschwindigkeit von 1,5 m/s. Mit zunehmender Vorwärtsgeschwindigkeit des Fahrzeugs steigt der durchschnittliche Reglerausgang für beide Regler an. Der Unterschied zwischen den beiden Reglern liegt jedoch im zeitlichen Verlauf der beiden Signale. Bei Vorwärtsgeschwindigkeiten zwischen 1,0 und 1,5  $\frac{\text{m}}{\text{s}}$  ist der Signalverlauf der beiden Regler ähnlich, wie in Abb. 19 und 21 zu sehen sind. Bei 1,75 und 2,0  $\frac{\text{m}}{\text{s}}$  verschlechtert sich jedoch der periodische Verlauf des PID-Reglers, während die Periodizität des Stanley-Reglers zunimmt.

Geschwindigkeit (m/s)	PID (°)	Stanley (°)
1,0	5,33	4,79
1,25	11,97	11,82
1,5	9,61	13,19
1,75	12,35	13,30
2,0	13,84	13,51

**Tabelle 3** – Zeitliche Mittelwerte der Reglerausgänge bei den verschiedenen Vorwärtsgeschwindigkeiten.

Bei  $1,75 \frac{\text{m}}{\text{s}}$  kann der PID-Regler das Fahrzeug immer noch auf die Fahrspur lenken. Wie in Abb. 22 gezeigt, führen Ausreißer allerdings zu starken Schwingungen im Reglerausgang. Wenn der PID-Regler die Fahrspur nicht richtig erkennt, hat er Schwierigkeiten, das Fahrzeug auf die Fahrspur zurückzubringen. Stattdessen kann es passieren, dass das Fahrzeug gegen die Wand fährt, wie am rechten Ende des Diagramms zu sehen ist. Im Vergleich dazu hat der Stanley-Regler keine Probleme, dem Weg mit der gleichen Geschwindigkeit zu folgen.

Bei  $2,0 \frac{\text{m}}{\text{s}}$  ist der PID-Regler nicht in der Lage, eine Kollision mit der Wand zu vermeiden. Wie in Abb. 23 dargestellt, schwankt der Ausgang zwischen -30 und 30 Grad, und es gibt keine Periodizität. Im Vergleich dazu weist der Stanley-Regler eine stabile Schwingung zwischen -30 und 0 Grad auf und ist periodisch.

Ausreißer führen nicht zu starken Schwingungen, wenn das Fahrzeug mit dem Stanley-Regler gesteuert wird. Es wurde beobachtet, dass sich das Fahrzeug viel schneller als der PID-Regler auf der Bahn neu orientiert.

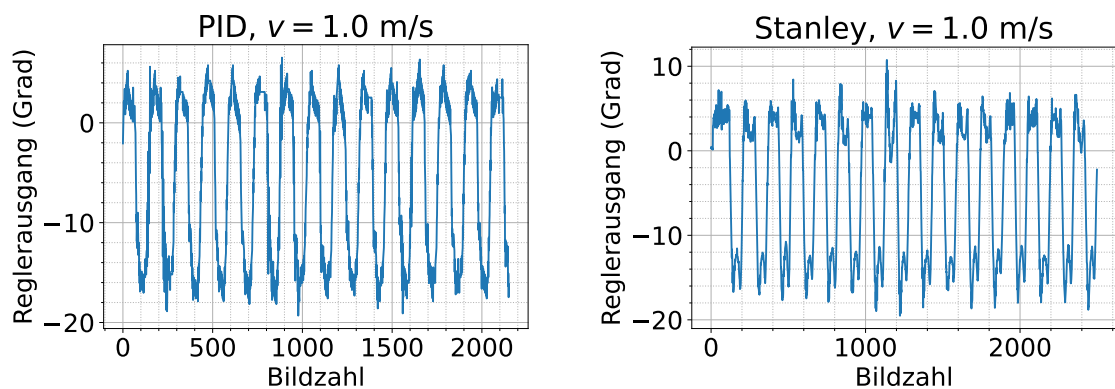


Abbildung 19 – Ausgang der Regler mit  $v = 1.0 \frac{\text{m}}{\text{s}}$ .

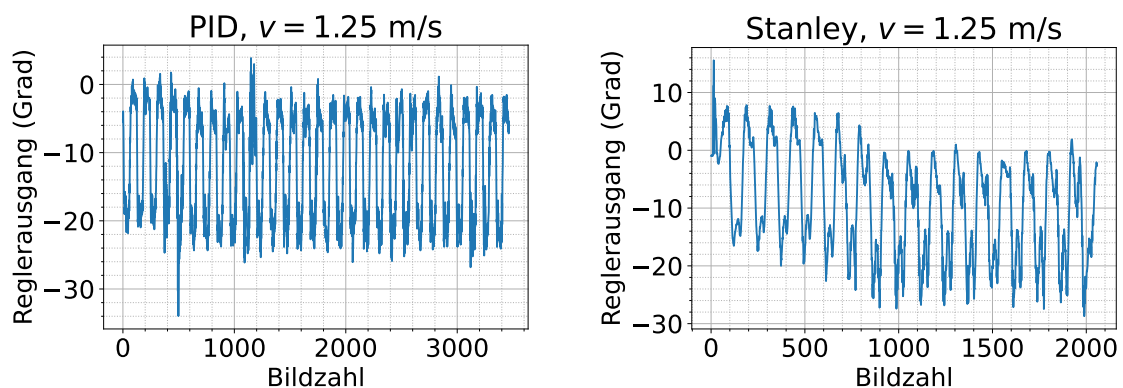
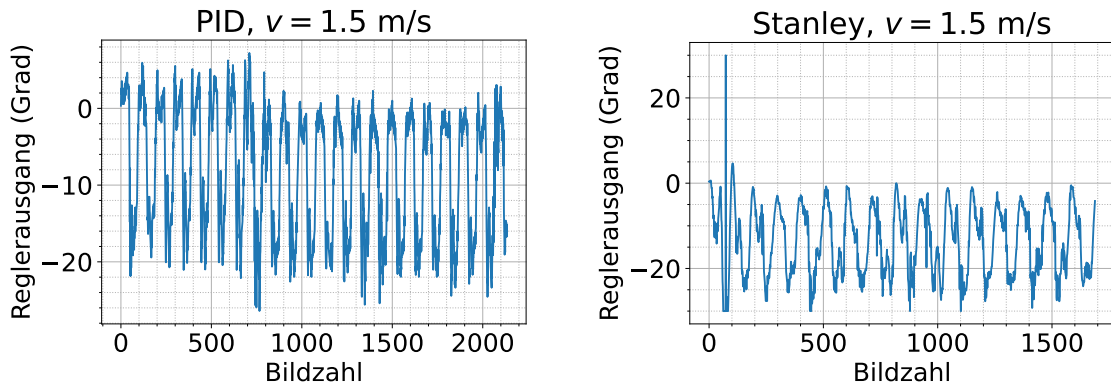
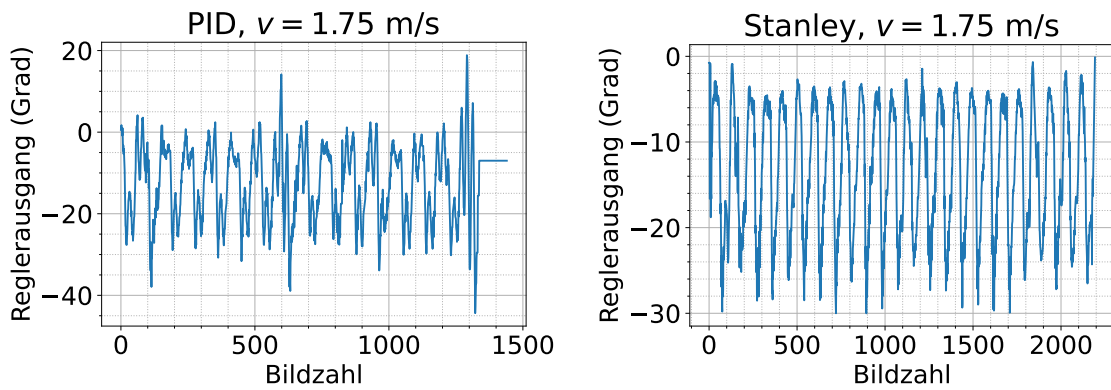


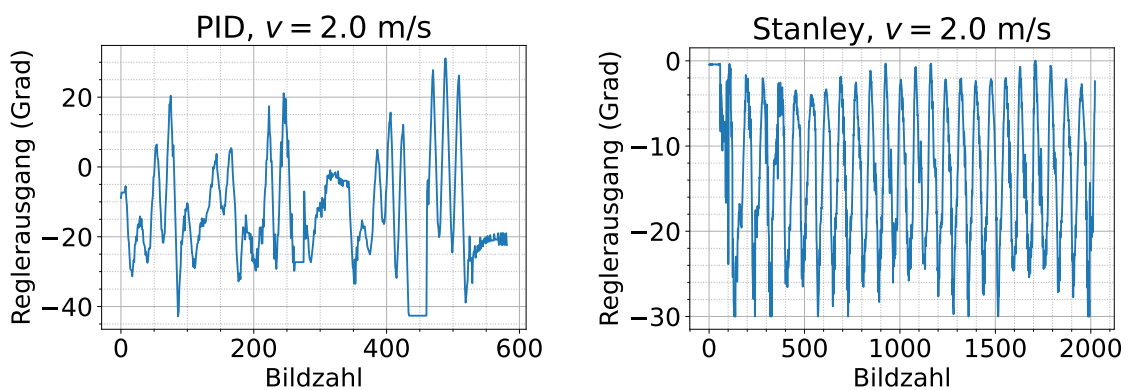
Abbildung 20 – Ausgang der Regler mit  $v = 1.25 \frac{\text{m}}{\text{s}}$ .



**Abbildung 21** – Ausgang der Regler mit  $v = 1.5 \frac{\text{m}}{\text{s}}$ .



**Abbildung 22** – Ausgang der Regler mit  $v = 1.75 \frac{\text{m}}{\text{s}}$ .



**Abbildung 23** – Ausgang der Regler mit  $v = 2.0 \frac{\text{m}}{\text{s}}$ . Das PID-Regler-Signal hat keine Periodizität und schwankt von -40 zu 30 Grad. Er ist bei dieser Geschwindigkeit nicht mehr nutzbar.



# Kapitel 5

## Fazit

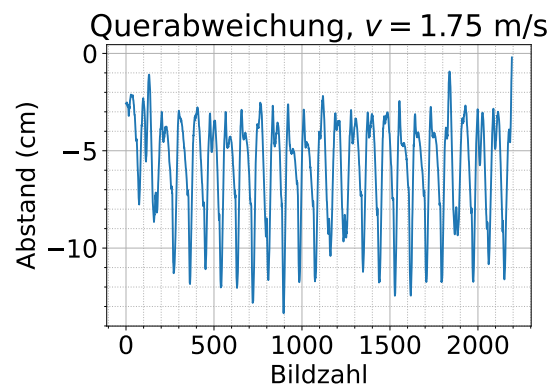
Das Bahnfolgeverhalten beider Regler wurde bewertet. Während der PID-Regler unter diesem Aspekt besser abschneidet, ist dies nicht das wichtigste Kriterium bei der Auswahl des Regelungsalgorithmus. Größere Bedeutung ist der Robustheit zuzumessen. In diesem Kontext hat sich der Stanley-Regler als überlegen erwiesen. Deshalb ist der Stanley-Regler geeigneter für die Anwendung des autonomen Fahrens, da er in der Lage ist, höhere Geschwindigkeiten ohne Überwachung zu bewältigen. Unter Berücksichtigung aller Aspekte der Problemstellung ist der Stanley-Regler daher die bessere Wahl für diese Anwendung.

### 5.0.1 Künftige Arbeit

Am Ende dieser Arbeit wurde festgestellt, dass das Fahrzeug vor jeder Kurve immer noch von der Fahrbahn abdriftet. Um dieses Problem zu lösen, wird vorgeschlagen, den Stanley-Regler in Anlehnung an [9] durch Hinzufügen eines Integrators zu verändern, um diese Abweichung zu kompensieren. Der adaptierte Stanley-Regler wird durch die folgende Gleichung beschrieben:

$$u = \theta - \theta_d + \arctan\left(\frac{ke_V}{v}\right) + k_i \int_0^t e_V d\tau, \quad (5.1)$$

In der linearen Regelungstheorie wird ein Integrator verwendet, um bleibende Regelabweichungen aus einer Anwendung zu entfernen. Da es sich bei der Kleinsignalversion des Stanley-Reglers um einen PD-Regler handelt, ist zu vermuten, dass das Hinzufügen eines Integrators den beobachteten Drift korrigieren wird. Wie in der Abb. 24 zu sehen ist, ist der Drift im Eingang des Reglers sichtbar. Durch die Integration der Querabweichung soll der Drift mit der Zeit kompensiert werden und das Fahrzeug auf der Fahrspur bleiben. Weitere Untersuchungen sind erforderlich, um diese Hypothese zu testen und die Wirksamkeit der vorgeschlagenen Anpassung zu überprüfen.



**Abbildung 24** – Das gemessene Querabweichungssignal mit  $v = 1.75 \frac{\text{m}}{\text{s}}$ . Der Mittelwert des Signals liegt bei -5.89 cm. Deswegen gibt es eine bleibende Abweichung, die mit dem ursprünglichen Stanley-Regler nicht kompensiert wird.

# Literatur

- [1] Gabriel M Hoffmann u. a. „Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing“. In: *2007 American control conference*. IEEE. 2007, S. 2296–2301.
- [2] Kenneth Jiang. *Calibrate fisheye lens using OpenCV — part 2*. [Online; accessed 11-May-2023]. 2017. URL: <https://medium.com/@kennethjiang/calibrate-fisheye-lens-using-opencv-part-2-13990f1b157f>.
- [3] John Kennedy, Alma Jenks und Steven White. *HSV Color Spaces*. [Online; Stand 11. Mai 2023]. 2021. URL: <https://learn.microsoft.com/en-us/windows/win32/wcs/hsv-color-spaces>.
- [4] OpenCV. *Camera Calibration*. [Online; Stand 11. Mai 2023]. URL: [https://docs.opencv.org/4.x/dc/dbb/tutorial\\_py\\_calibration.html](https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html).
- [5] Addison Sears-Collins. *How to Perform Camera Calibration Using OpenCV*. [Online; Stand 11. Mai 2023]. 2020. URL: <https://automaticaddison.com/how-to-perform-camera-calibration-using-opencv/>.
- [6] Addison Sears-Collins. *The Ultimate Guide to Real-Time Lane Detection Using OpenCV*. [Online; Stand 11. Mai 2023]. 2021. URL: <https://automaticaddison.com/the-ultimate-guide-to-real-time-lane-detection-using-opencv/>.
- [7] Jarrod M Snider u. a. „Automatic steering methods for autonomous automobile path tracking“. In: *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RITR-09-08* (2009).
- [8] Wikipedia. *PT1-Glied* — *Wikipedia, die freie Enzyklopädie*. [Online; Stand 11. Mai 2023]. 2023. URL: <https://de.wikipedia.org/w/index.php?title=PT1-Glied&oldid=232014372>.
- [9] Wikipedia. *Regler* — *Wikipedia, die freie Enzyklopädie*. [Online; Stand 11. Mai 2023]. 2023. URL: <https://de.wikipedia.org/w/index.php?title=Regler&oldid=230849162>.
- [10] Wikipedia. *Verzeichnung* — *Wikipedia, die freie Enzyklopädie*. [Online; Stand 11. Mai 2023]. 2023. URL: <https://de.wikipedia.org/w/index.php?title=Verzeichnung&oldid=230628646>.
- [11] Wikipedia contributors. *Polynomial regression* — *Wikipedia, The Free Encyclopedia*. [Online; accessed 11-May-2023]. 2022. URL: [https://en.wikipedia.org/w/index.php?title=Polynomial\\_regression&oldid=1113806162](https://en.wikipedia.org/w/index.php?title=Polynomial_regression&oldid=1113806162).

- [12] Wikipedia contributors. *Tangential angle* — *Wikipedia, The Free Encyclopedia*. [Online; accessed 11-May-2023]. 2022. URL: [https://en.wikipedia.org/w/index.php?title=Tangential\\_angle&oldid=1121139652](https://en.wikipedia.org/w/index.php?title=Tangential_angle&oldid=1121139652).
- [13] Jan Winkler. *Vorlesung Steuerung und Regelung flacher und verteiltparametrischer Systeme*. 2022.

# Abbildungsverzeichnis

1	Darstellung der Strecke. . . . .	3
2	Grafische Darstellung der geometrischen Beziehungen zwischen den Größen des Stanley-Reglers. . . . .	6
3	Grafische Darstellung der Fahrbahn. . . . .	8
4	Grafische Darstellung der Beziehungen zwischen den benötigten Parametern. Der Vektor $\underline{p}_P(s^*)$ ist hier der Pfadpositionsvektor mit dem kleinsten Abstand zum Vorderradachsemittelpunkt. . . . .	9
5	Darstellung des Regelkreises. . . . .	10
6	Verhalten des Stanley-Reglers bei verschiedenen Abtastfrequenzen und Zeitkonstanten. Das Verhalten bleibt ähnlich bei Frequenzen unter 15 Hz. Der Stanley-Regler folgt den Pfad besser wenn die Zeitkonstante des PT1-Glieds niedrig ist, aber die Lenkwinkelstrecke im Auto hat eine Zeitkonstante zwischen 100 und 200 ms. . . . .	11
7	Grafische Darstellung des Bildverarbeitungspipelines. . . . .	12
8	Ein Beispiel einer tonnenförmigen Verzeichnung. . . . .	13
9	Die Kalibrierung des linken Bilds liefert das rechte Bild. Die gekrümmte Linie des Schachbretts sind gerade nach dem Verzeichnungs Korrektur. Der Grund für die wellenförmigen Verzerrungen ist die nicht perfekte Kalibrierung. . . . .	14
10	Ein Beispiel von dem Korrekturschritt. Die Mittellinie im rechten Bilds ist leicht nach links verschoben. . . . .	15
11	Ein Beispiel von der Ausgabe des Color-Thresholdings. Links: Bild nach dem Kalibrierungsschritt, Eingang zum „Color-Thresholding“. Rechts: binäres Bild mit nur Pixeln, die als gelb erkannt wurden. . . . .	16
12	Perspektivtransformation für ein Beispielbild. Die Trapez im linken Bild wird ausgeschnitten und gestreckt, sodass sich das rechte Bild ergibt. Zu diesem Zeitpunkt hat noch kein „Color-Thresholding“ stattgefunden. . . . .	17
13	Histogramm der schwarzen Pixel. Die Spitze des Histogramms liegt bei dem $x$ -Schnittpunkts. . . . .	18
14	Das linke Bild ist ein Beispiel von der Gleitfenstermethode. Das rechte Bild zeigt das Polynom, das von den verbleibenden Pixel approximiert wurde. . . . .	20

15	Ein Beispiel von einem Ausreißer. Die drei Bilder wurden nacheinander aufgenommen. Im rechten und linken Bilder wurde die Fahrspur richtig erkannt. Im mittleren Bild wurde die Fahrspur falsch erkannt. . . . .	21
16	Modellfahrzeuge. . . . .	22
17	Gezeigt ist der vom Fahrzeug gewählte Pfad (Fahrtrichtung von rechts nach links) für die Vorwärtsgeschwindigkeiten $v = 1.0 \frac{\text{m}}{\text{s}}$ und $v = 1.25 \frac{\text{m}}{\text{s}}$ . Dabei wurde der Standort des Fahrzeugs nicht immer ganz korrekt erfasst, was zu den sichtbaren Zacken führt. Es ist sichtbar, dass der PID- und der Stanley-Regler ungefähr das gleiche Verhalten haben. . . . .	25
18	Das Modellfahrzeug mit dem Stanley-Regler hat eine kleine Abweichung am Anfang der Kurve (rechts). Dabei handelt es sich um den in Kapitel 4 beschriebenen Drift. Am Ende der Kurve ist es aber auf der Fahrspur. . . . .	26
19	Ausgang der Regler mit $v = 1.0 \frac{\text{m}}{\text{s}}$ . . . . .	27
20	Ausgang der Regler mit $v = 1.25 \frac{\text{m}}{\text{s}}$ . . . . .	27
21	Ausgang der Regler mit $v = 1.5 \frac{\text{m}}{\text{s}}$ . . . . .	28
22	Ausgang der Regler mit $v = 1.75 \frac{\text{m}}{\text{s}}$ . . . . .	28
23	Ausgang der Regler mit $v = 2.0 \frac{\text{m}}{\text{s}}$ . Das PID-Regler-Signal hat keine Periodizität und schwankt von -40 zu 30 Grad. Er ist bei dieser Geschwindigkeit nicht mehr nutzbar. . . . .	28
24	Das gemessene Querabweichungssignal mit $v = 1.75 \frac{\text{m}}{\text{s}}$ . Der Mittelwert des Signals liegt bei -5.89 cm. Deswegen gibt es eine bleibende Abweichung, die mit dem ursprünglichen Stanley-Regler nicht kompensiert wird. . .	30