



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Bachelorarbeit

Herry Sukandar Lau

Trajektorienplanung für autonome Fahrzeuge mit  
Hilfe von Bezierkurven, SVG-Grafiken und XML

Herry Sukandar Lau

Trajektorienplanung für autonome Fahrzeuge mit  
Hilfe von Bezierkurven, SVG-Grafiken und XML

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

Betreuender Prüfer : Prof. Dr. rer. nat. Reinhard, Baran  
Zweitgutachter : Prof. Dr.Ing. Meisel, Andreas

Abgegeben am 28. April 2011

**Herry Sukandar Lau**

**Thema der Bachelorarbeit**

Trajektorienplanung für autonome Fahrzeuge mit Hilfe von Bezierkurven, SVG-Grafiken und XML

**Stichworte**

Scalable Vector Graphics (SVG), autonome Systeme und Fahrzeug, Trajektorie, Bezierkurve, De Casteljau

**Kurzzusammenfassung**

In dieser Bachelorarbeit wurde eine Software zur Route- bzw. Trajektorieberechnung eines Fahrzeugs mit Hilfe von Algorithmen für Berechnung der Stütz- und Zielpunkten sowie Lenkung und Länge der Strecke entwickelt. Die Software ermöglicht ebenfalls die Steuerung des Fahrzeugs, dessen Position über einen Laserscanner erfasst und korrigiert wird.

**Herry Sukandar Lau**

**Title of the paper**

Trajectory for autonomous vehicles with Bezier curves, SVG graphics and XML

**Keywords**

Scalable Vector Graphics (SVG), autonomus System und Vehicle, Beziercurve, De Casteljau

**Abstract**

In this bachelor thesis a software for calculation of the vehicle route has been developed using algorithms for calculation of the basepoints, goal points, steering and length of the route. The software also enables the control of the vehicle, where the positions will be recorded by a laser scanner and corrected.

## **Danksagung**

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich bei Erstellung dieser Arbeit tatkräftig unterstützt haben.

Mein ganz besonderer Dank gilt meinen beiden Betreuern Herrn Prof. Dr. rer. nat. Reinhard Baran und Herrn Prof. Dr.Ing. Andreas Meisel. Sie ermöglichten mir meine Bachelorarbeit als Abschluss meines Studiums an der HAW-Hamburg anzufertigen und begleiteten mich in dieser letzten Phase mit vielen angeregten Diskussionen und Ideen. Ich hoffe, dass ich einen Beitrag zum Gelingen dieses Projektes beisteuern konnte.

Nicht zu vergessen ist meine Familie für Ihre Unterstützung und die daraus für mich resultierende Motivation.

# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>7</b>
<b>Abbildungsverzeichnis</b>	<b>8</b>
<b>1 Einleitung</b>	<b>9</b>
<b>2 Aufgabestellung</b>	<b>10</b>
<b>3 Hardware und Grundlage</b>	<b>12</b>
3.1 Scalable Vector Graphics (SVG)	12
3.2 Das Fahrzeug	14
3.2.1 Geometrie des Fahrzeugs	14
3.2.2 Steuerung des Fahrzeugs	15
3.3 Hokuyo URG-04LX-UG01 Laserscanner	16
3.4 Kommunikation über die API des Human Interface Device (USB)	17
<b>4 Bézier</b>	<b>19</b>
4.1 Grundlage	19
4.2 Lineare Bézierkurven	20
4.3 Quadratische Bézierkurven	20
4.4 Kubische Bézierkurven	21
4.5 De Casteljau	22
<b>5 Realisierung</b>	<b>24</b>
5.1 Aufbau der Strecke	24
5.2 Berechnung der Bezugspunkte	29
5.3 Länge der Gesamtstrecke	30
5.4 Abstand zwischen zwei Zielpunkten	32
5.5 Berechnung der Zielpunkte	32
5.6 Berechnung der Teilstrecke	33
5.7 Berechnung der Krümmung der Zielpunkte	35
5.8 Berechnung des Lenkwinkels	37
5.9 Simulation mit MATLAB	40

**6 Zusammenfassung**

**43**

**Literaturverzeichnis**

**44**

# Tabellenverzeichnis

3.1	Pfad-Kommandos ( <a href="#">Self SVG</a> ) . . . . .	13
5.1	Punkte der Stützstelle( $P_0, P_3$ ) sowie der Bézier-Stützpunkte( $P_1, P_2$ ) der Trajektorie ( Legende siehe Abbildung <a href="#">5.3</a> ) . . . . .	28
5.2	Die Genauigkeit der Streckenlänge(cm) von Anzahl der Punkte(n) . . . . .	31

# Abbildungsverzeichnis

2.1	Aufbau der Arbeit . . . . .	11
3.1	Die Kurve wird von <path>-element aus Listing 3.1 dargestellt . . . . .	12
3.2	Skizze und Abmessung Wedico LKW ( <a href="#">Arbaoui 2005</a> ) . . . . .	14
3.3	Die Kommunikation . . . . .	15
3.4	Hokuyo URG-04LX-UG01 Laserscanner . . . . .	16
4.1	Stützpunkte als Gewichte . . . . .	19
4.2	Lineare Bézierkurven . . . . .	20
4.3	Quadratische Bézierkurven . . . . .	21
4.4	Kubische Bézierkurven . . . . .	21
4.5	Schema von de Casteljau für $n = 3$ . . . . .	22
4.6	Kubische Bézierkurven Konstruktion mit de Casteljau für $t = 1/2$ . . . . .	23
4.7	Kubische Bézierkurven Konstruktion mit de Casteljau für $t = 1/4$ . . . . .	23
5.1	Teststrecke . . . . .	24
5.2	Abmessungen der Teststrecke . . . . .	25
5.3	Aufbau der Trajektorie mit Bézier . . . . .	26
5.4	Die Trajektorie . . . . .	26
5.5	Bezugspunkte mit Hilfe der Theorie von de Casteljau . . . . .	29
5.6	Berechnung des Abstands zwischen zwei Bezugspunkten mit $\Delta t=0.1$ $n = 10$ . . . . .	30
5.7	Bezugspunkte $\Delta t=0.1$ $n = 10$ . . . . .	31
5.8	Bezugspunkte $\Delta t=0.05$ $n = 20$ . . . . .	31
5.9	Bestimmung der Zielpunkte . . . . .	33
5.10	Anwendung vom de Casteljau . . . . .	34
5.11	Teilstrecke mit neuen Zielpunkten mit Hilfe der Bézier-Stützpunkte von de Casteljau . . . . .	34
5.12	Krümmungsverlauf der gefahrenen Strecke . . . . .	37
5.13	Ein-Spur Fahrzeugmodell . . . . .	38
5.14	Lenkungsverlauf der gefahrenen Strecke . . . . .	39
5.15	Abhängigkeit des Radius von dem Lenkwinkel . . . . .	39
5.16	Simulation der Trajektorie und Visualisierung der Daten . . . . .	41
5.17	Simulation Soll- und Ist-Zielpunkte sowie der Lenkung und Krümmung . . . . .	42

# 1 Einleitung

In vielen Bereichen der Industrie, des öffentlichen und privaten Bereichs finden autonome Systeme immer häufiger ihren Einsatz. Fahrerlose Transportsysteme flexibilisieren Produktion und Logistik. Assistenzroboter erleichtern Menschen mit körperlichen Einschränkungen den Alltag.

Ein Teilgebiet auf dem Sektor der autonomen Systeme ist der Bereich der autonomen Fahrzeuge. Dabei handelt es sich um fahrerlose, intelligente Fahrzeuge, die ihnen übertragene Aufgaben selbständig ohne menschliche Unterstützung durchführen können. Zur Wahrnehmung der Umwelt werden verschiedene Arten von Sensoren zum sicheren Navigieren eingesetzt.

An der Hochschule für Angewandte Wissenschaften Hamburg wird seit einigen Jahren an der Realisierung verschiedener autonomer Fahrzeugsysteme im Rahmen eines so genannten FAUST Projektes gearbeitet. Es werden Algorithmen zur Optimierung des Streckenverlaufs und Kurvenfahrt sowie eine Software zur Steuerung der Bewegung des Fahrzeugs entwickelt.

## 2 Aufgabestellung

Der Überblick über die Aufgabestellung dieser Arbeit ist in der Abbildung [2.1](#) zu finden. Zuerst werden die Stützpunkte für den Streckenverlauf mit Hilfe eines Bildbearbeitungsprogramms Inkscape bestimmt. Die Stützpunkte werden als Scalable Vector Graphics (SVG) gespeichert und ausgelesen, um die Zielpunkte mit Hilfe der Algorithmentheorie der Bézierkurve und von de Casteljau zu ermitteln. Die Zielpunkte stellen den Streckenverlauf bzw. Trajektorie des Fahrzeuges dar.

Die vom Laserscanner gemessene aktuelle Position des Fahrzeugs wird durch einen Algorithmus ausgewertet. Mit diesem kann das Fahrzeug gesteuert und seine Position ständig korrigiert werden, um dieses sicher ans Ziel führen zu können.

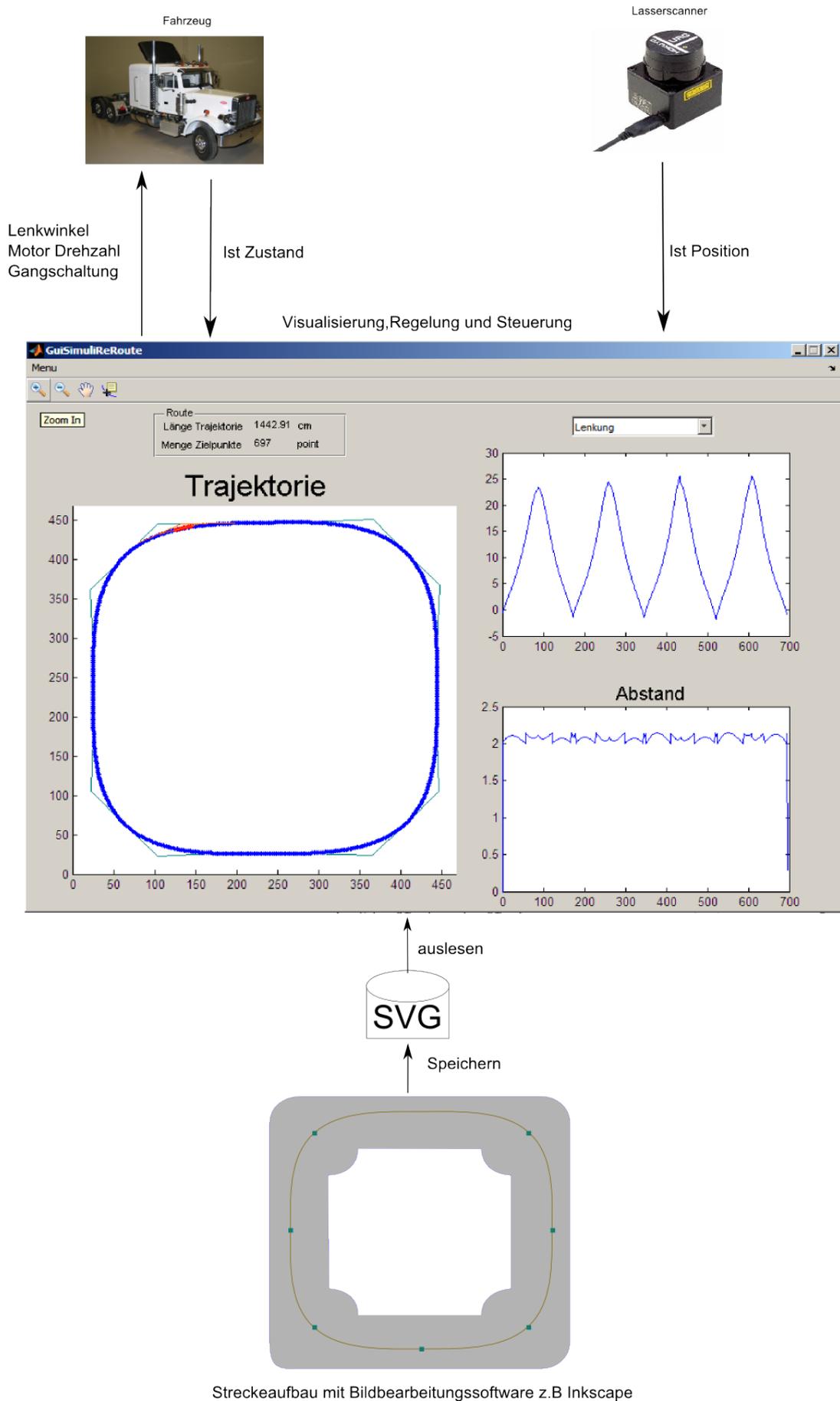


Abbildung 2.1: Aufbau der Arbeit

# 3 Hardware und Grundlage

## 3.1 Scalable Vector Graphics (SVG)

Scalable Vector Graphics (SVG) ist eine Sprache, um zweidimensionale Grafiken in XML zu beschreiben. Einige der gebräuchteste Webbrowser können ohne nachträgliche Installation von Erweiterung einen Großteil des Sprachumfangs darstellen. Da SVG ein XML-basiertes Dateiformat ist, können SVG-Dateien mit Hilfe eines Texteditors bearbeitet werden. Texte, die in SVG-Dateien verwendet werden, sind für gegebenenfalls erforderliche computerunterstützte Übersetzung leicht zugänglich. Es gibt jedoch auch spezielle Programme zur Bearbeitung, zum Beispiel das freie Vektorgrafik-Programm Inkscape.

```
<path d="M20 200 C20 150 250 200 200 200" style="fill:none;stroke-width:3;stroke:black;" />
```

Listing 3.1: Beispiel eines Pfades

Das SVG-path-Element beschreibt einen Linienzug, dem der Weg eines Zeichenstiftes zugrunde liegt. Die jeweilige Position des Zeichenstiftes wird als aktuelle Position oder Cursor-Koordinate im Hintergrund mitgeführt. Das Attribut d enthält die eigentlichen Pfaddaten, das sind Kommandos, meist gefolgt von lokalen Koordinaten oder Parametern. Kommandos werden in großen oder kleinen Buchstaben angegeben. Große Buchstaben entsprechen absoluten Koordinatenangaben. Kleine Buchstaben entsprechen relativen Angaben, relativ zum letzten Punkt des Pfades.

Tabelle 3.1 bietet eine Kurzübersicht über verfügbare Kommandos



Abbildung 3.1: Die Kurve wird von <path>-element aus Listing 3.1 dargestellt

Kommando	Beschreibung
M,m (MoveTo)	Setzt den aktuellen Punkt fest, von dem aus der Pfad starten soll
Z,z (ClosePath)	Erstellt eine geschlossene Form
H,h (Horizontal LineTo)	Zeichnet eine Linie vom aktuellen zum angegebenen Punkt
V,v (Vertical LineTo)	Zeichnet vom aktuellen Punkt aus eine vertikale Linie
C,c (CurveTo)	Zeichnet vom aktuellen Punkt aus eine Bézier-Kurve
S,s (Shorthand/Smooth CurveTo)	Ähnlich dem Kommando C, jedoch wird hier als erster Kontrollpunkt der letzte aus der direkt zuvor festgelegten Bézier-Kurve verwendet
Q,q (Quadratic Bézier CurveTo)	Zeichnet eine quadratische Bézier-Kurve
T,t (Shorthand/Smooth Quadratic Bézier CurveTo)	Ähnlich dem Kommando Q, jedoch wird hier als erster Kontrollpunkt der letzte aus der direkt zuvor festgelegten quadratischen Bézier-Kurve verwendet

Tabelle 3.1: Pfad-Kommandos ([Self SVG](#))

## 3.2 Das Fahrzeug

### 3.2.1 Geometrie des Fahrzeugs

Als Fahrzeug wurde ein Modellbau-LKW im Maßstab 1:16 aus der Firma Wedico (Abbildung 3.2) verwendet und besteht aus einer Sattelzugmaschine und einem Pritschen-Auflieger. Die Zugmaschine hat 3 Achsen, wobei die vordere Achse lenkbar ist und der Pritschen-Auflieger 2 Achsen hat. Die Räder an der Lenkachse lassen sich um  $26^\circ$  jeweils nach links und nach rechts drehen.

Die beiden Hinterachsen werden von dem Motor über ein 3-Gang- Schaltgetriebe angetrieben und haben je ein Differentialgetriebe. Der erste Gang hat eine Übersetzung von 11:1, der zweite Gang 5,5:1 und der dritte Gang 2,75:1. Durch das Differential erhält man noch eine weitere Übersetzung von 2:1. Am Motor kann die Drehzahl abgenommen werden und an einen Geräuschgenerator weitergeleitet werden. Weitere später benötigte Daten sind der Umfang der Räder mit 210mm und der Abstand zwischen den idealisierten Mittelpunkten der Vorder- und Hinterachse von 440mm. Weitere technische Details wie auch die Hintergründe zu den Fahrstufen können in einer früheren Studienarbeit gelesen werden. (siehe (Hübener 2004))

Die Abmessungen sind aus der Abbildung 3.2 zu entnehmen.

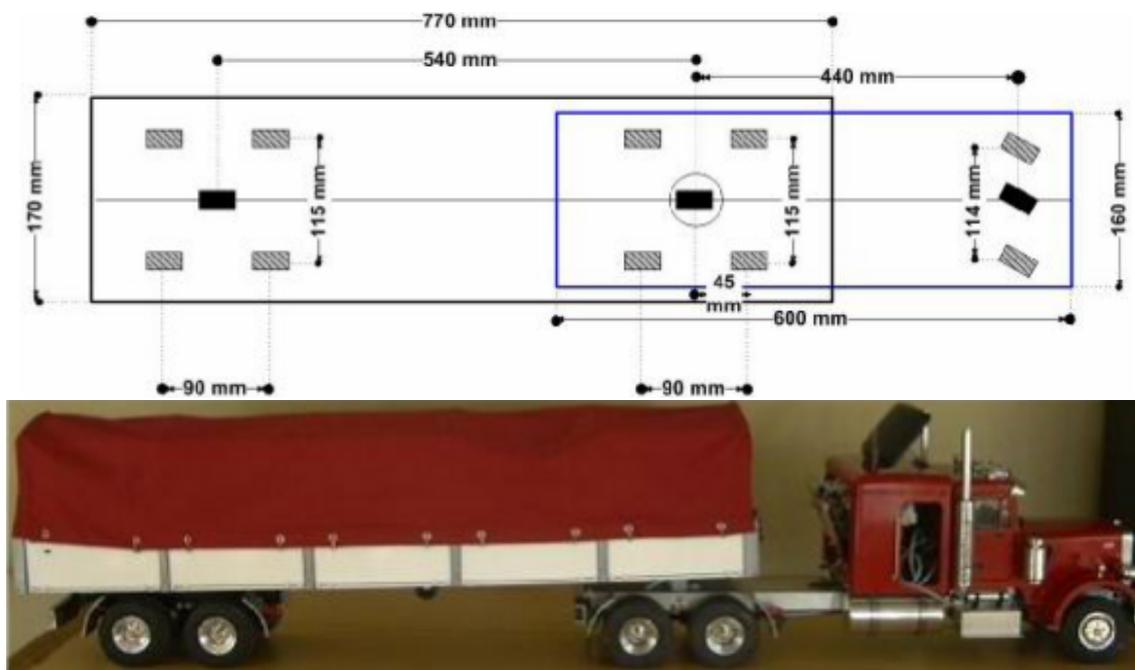


Abbildung 3.2: Skizze und Abmessung Wedico LKW (Arbaoui 2005)

### 3.2.2 Steuerung des Fahrzeugs

Ein LPC-Stick (Mikrocontroller auf ARM7-Basis) dient als Schnittstelle zur Sensorik des Fahrzeugs. Die eigentliche Steuerung des Fahrzeugs erfolgt auf einem mitgeführten Netbook, welches über USB mit dem Mikrocontroller kommuniziert. Hierzu wird auf das HID<sup>1</sup>-Protokoll (siehe Kapitel 3.4) zurückgegriffen. (siehe Bachelorarbeit Erik Böhme <sup>2</sup>)

Abbildung 3.3 zeigt den Kommunikationsaustausch zwischen dem Netbook und Lastwagen. Der Lastwagen wurde mit verschiedenen Sensoren zur Messung und Überwachung der entsprechenden Parameter (Drehzahl, Lenkung, usw.) ausgestattet. Die Sensordaten (Ist Zustand des Fahrzeugs) werden über den Mikrocontroller (LPC Stick) an das Netbook zur Weiterverarbeitung und Steuerung des Fahrzeuges gesendet. Das Netbook berechnet die neuen Parameter und sendet die entsprechenden Befehle über USB an den LPC Stick. Die Informationen werden anschließend vom LPC-Stick in Steuersignale (wie z.B. über PWM<sup>3</sup> für die Aktorik) umgewandelt und an den Lastwagen übermittelt

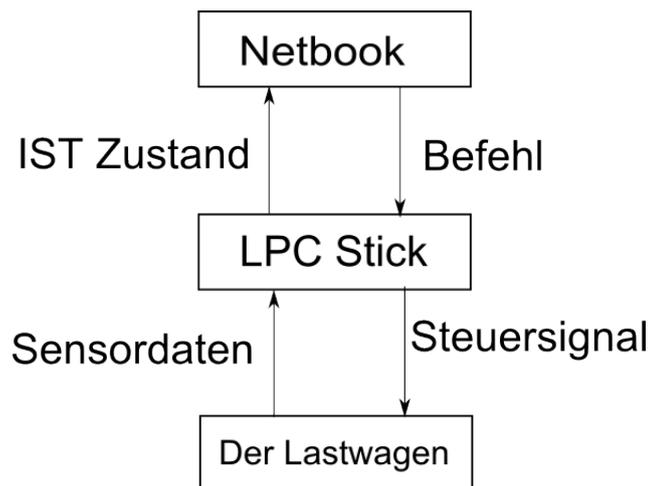


Abbildung 3.3: Die Kommunikation

<sup>1</sup>Human Interface Device

<sup>2</sup>Die Arbeit zum Zeitpunkt der Veröffentlichung ist noch nicht abgeschlossen

<sup>3</sup>Pulse-width modulation / Pulsweitenmodulation

### 3.3 Hokuyo URG-04LX-UG01 Laserscanner

Zur Lokalisierung der Position vom Fahrzeug kommt ein 2D-Laserscanner der Firma Hokuyo zum Einsatz (siehe Abbildung 3.4). Der Laserscanner liefert ein zweidimensionales Entfernungsbild über ein Sichtfeld von  $240^\circ$  und eine Reichweite von 5,6 meter. Seine Scanzeit beträgt von 100 ms pro scan. Die Informationen des Laserscanners werden mit Hilfe einer Software (siehe Bachelorarbeit Lief Böhme <sup>1</sup>) ausgewertet und in die aktuelle X/Y-Raumkoordinate sowie die Richtung( $\delta$ ) des Fahrzeugs umgewandelt. Der Laserscanner wird auf der Sattelzugmaschine montiert. Er wird über den USB Port betrieben und durch diesen mit Strom versorgt.

Technische Daten:

- Abmessung 50 x 50 x 70 mm
- Gewicht 160g
- Versorgungsspannung 5V DC
- Reichweite bis 5,6 m
- Öffnungswinkel  $240^\circ$
- Scanfrequenz 10 Hz
- Schnittstellen USB 2.0



Abbildung 3.4: Hokuyo URG-04LX-UG01 Laserscanner

---

<sup>1</sup>Die Arbeit zum Zeitpunkt der Veröffentlichung ist noch nicht abgeschlossen

## 3.4 Kommunikation über die API des Human Interface Device (USB)

Das Human Interface Device (HID) ermöglicht die Kommunikation zwischen Anwender und dem Fahrzeug. Da HID-Gerätetreiber in den gängigen Betriebssystemen enthalten sind, können die Entwickler sich auf die in Windows bereits vorhandenen HID-Treiber stützen. Sobald ein HID-Gerät (während des Betriebs) angeschlossen wird, wird es direkt als Gerätetyp „Eingabegeräte“ (Human Interface Devices) erkannt und dann z.B. im Gerätemanager von Microsoft Windows angezeigt.

Die Windows-API des Human Interface Device umfasst Funktionen, die von Anwendern verwendet werden können, um sich Informationen über die Berichte eines HIDs zu beschaffen und Berichtsdaten zu senden und zu empfangen. Hierzu ist das Einbinden der Standard-Windows Dynamic Link Library's (DLL) des HID notwendig, die in einer Programmiersprache wie C/C++ und Visual Basic (siehe ([USB 2.0 Handbuch für Entwickler](#))) über die Headerfiles (hid.h, hidspi.h) einzubinden sind.

Ein USB-Gerät lässt sich durch die Kombination einer Vendor ID (VID) und der Produkt ID (PID) eindeutig identifizieren. Die Vendor ID klassifiziert einen Hersteller, der Produkt IDs eindeutig vergibt. Die Produkt IDs (PID) können allerdings unter verschiedenen Vendor IDs mehrfach auftreten.

### Definition der USB-Schnittstelle zum LPC-Stick:

Zu sendenden Daten zum Fahrzeug:

- Byte 0 → Die Lenkung (0 (linker Anschlag -26°), 127 (Mitte 0°), 255 (rechter Anschlag 26°))
- Byte 1 → Soll (Drehzahl Wert mal 20 = Solldrehzahl, also 0-5100 RPM)
- Byte 2 → Gang (1,2 oder 3 Werte, 0 oder größer 3 werden ignoriert)
- Byte 4 → Licht und Ton (Aufteilung noch nicht festgelegt)
- Byte 5 → Richtung (1 = Vorwärts 2 = Rückwärts, Der Rest wird ignoriert)
- byte 6 → Steuerung Timeout (Wert mal 20 ms also 0-5100 ms)
- Byte 7 → Mode (0 = Truck gibt Lenkung, Motorschub in %, Gang und Licht 1 = Truck gibt Lenkung, Sensor Links/Front/Rechts aus)

Gelesene Daten vom Fahrzeug:

- Byte 0 → Lenkung IST Wert (Aufteilung noch nicht festgelegt)
- Byte 1 → Motorschub oder Sensor links in cm (Aufteilung noch nicht festgelegt)
- Byte 2 → Gang oder Sensor Vorne in cm (Aufteilung noch nicht festgelegt)
- byte 3 → Lichtzustand oder Sensor rechts in cm (Aufteilung noch nicht festgelegt)
- Byte 4 → IST Drehzahl HighByte
- Byte 5 → IST Drehzahl LowByte
- Byte 6 → Impulszähler HighByte
- Byte 7 → Impulszähler LowByte

Aufgrund des noch nicht abgeschlossenen Interfaceprogramms des Mikrocontrollers konnten nur die zu sendenden Bytes 0, 1, 2, 5 und 7 ausgiebig getestet werden. Die Funktionsweise aller anderen Bytes jedoch sind noch nicht erfolgreich getestet.

## 4 Bézier

Bei dem EDV Begriff Bézier-Kurve handelt es sich um ein mathematisches Verfahren zur Darstellung von Kurven. Die Bézier-Kurve wurde vom französischen Mathematiker und Ingenieur Pierre Bézier im Jahr 1962 bei der Automobil Firma Renault entwickelt. Pierre Bézier fand ein einfaches Verfahren zur Darstellung beliebiger Kurven, welches als Bézierkurve bezeichnet wird. Bézierkurven sind nach mathematischen Regeln formbare Linien.

### 4.1 Grundlage

Bézier Kurven sind Kurvenverläufe, die durch zwei Endpunkte einer Kurve und die Parameter der an ihnen angelegten Tangenten definiert sind. Durch Unterteilung von Kurvenzügen lassen sich alle Kurven als Bézier Kurven darstellen. Wenn man einen beliebigen Stützpunkt, zwischen Anfangs- und Endpunkt bewegt, wird die Kurve in der Nähe des entsprechenden Punktes gestaucht bzw. gestreckt. Der restliche Teil wird von der Veränderung umso weniger beeinflusst, desto weiter der Stützpunkt entfernt liegt. Folglich lassen sich die Stützpunkte also auch als eine Art Gewicht verstehen, welches die Kurve verbiegt. Dabei wäre ein Stützpunkt, der weit von der Kurve entfernt liegt, ein schweres Gewicht als ein naher Stützpunkt und würde den Verlauf auch stärker beeinflussen.

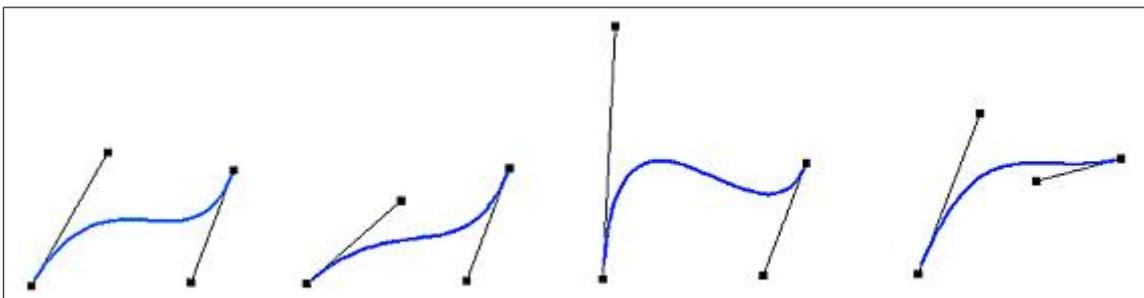


Abbildung 4.1: Stützpunkte als Gewichte

Die Kurve wird durch  $n + 1$  Stützpunkte  $P_0, P_1, \dots, P_n$  bestimmt. Die Kurve läuft nicht durch

alle Stützpunkte, sondern wird von ihnen beeinflusst.

$$P(t) = \sum_{i=0}^n B_{i,n}(t) \cdot P_i, \quad 0 \leq t \leq 1 \quad (4.1)$$

Der Punkt  $P_i$  wird gewichtet mit Hilfe eines Bernstein-Polynoms

$$B_{i,n}(t) = \binom{n}{i} \cdot t^i \cdot (1-t)^{n-i}, \quad i = 0, \dots, n \quad (4.2)$$

## 4.2 Lineare Bézierkurven

Gegebene Punkte  $P_0$  und  $P_1$ , ist eine lineare Bézier Kurve einfach eine Gerade zwischen jenen zwei Punkten. Die Kurve wird vorbei gegeben und ist mit gleichwertig lineare Interpolation.

$$P(t) = \sum_{i=0}^n B_{i,n}(t) \cdot P_i = (1-t)P_0 + tP_1, \quad 0 \leq t \leq 1 \quad (4.3)$$

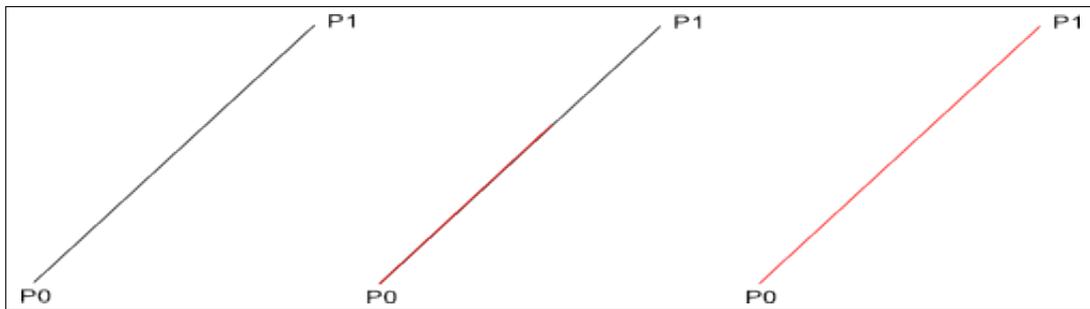


Abbildung 4.2: Lineare Bézierkurven

## 4.3 Quadratische Bézierkurven

Eine quadratische Gleichung Bézier Kurve ist der Weg, der durch die Funktion verfolgt wird.  $P(t)$ , Punkte gegeben  $P_0$ ,  $P_1$  und  $P_2$ , eine quadratische Gleichung Bézier Kurve ist auch ein Parabolisches Segment.

$$P(t) = \sum_{i=0}^n B_{i,n}(t) \cdot P_i = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2, \quad 0 \leq t \leq 1 \quad (4.4)$$

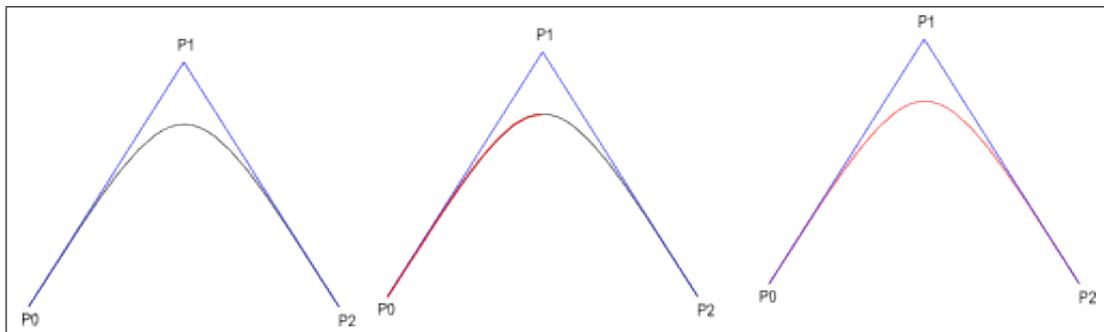


Abbildung 4.3: Quadratische Bézierkurven

## 4.4 Kubische Bézierkurven

Vier Punkte  $P_0, P_1, P_2$  und  $P_3$  bestimmen eine kubische Bézierkurve. Die Kurve Anfänge an  $P_0$  Gehen in Richtung zu  $P_1$  und kommt in an  $P_3$  Kommen von der Richtung von  $P_2$ . Normalerweise überschreitet es nicht durch  $P_1$  oder  $P_2$ , diese Punkte sind nur dort, Richtungsinformationen zur Verfügung zu stellen. Der Abstand zwischen  $P_0$  und  $P_1$  stellt „fest, wie lang“ die Kurve in Richtung bewegt  $P_2$  vor dem Drehen in Richtung zu  $P_3$ .

$$P(t) = \sum_{i=0}^n B_{i,n}(t) \cdot P_i = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t) P_2 + t^3 P_3, \quad 0 \leq t \leq 1 \quad (4.5)$$

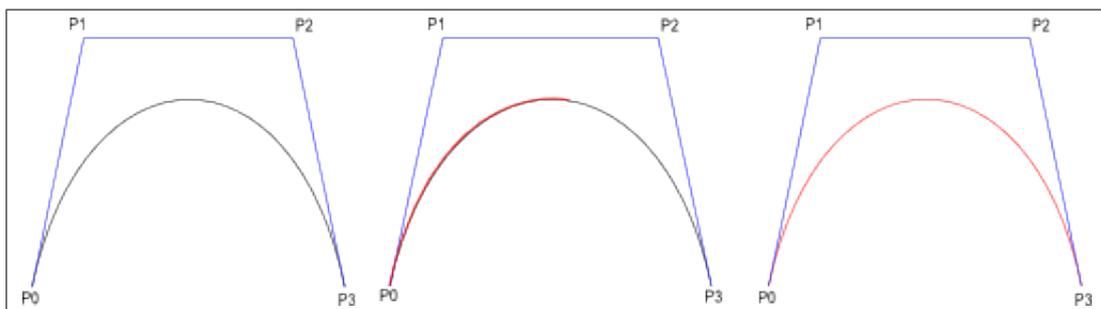


Abbildung 4.4: Kubische Bézierkurven

## 4.5 De Casteljau

Der Algorithmus von de Casteljau ermöglicht die effiziente Berechnung einer beliebig genauen Näherungsdarstellung von Bézierkurven durch einen Polygonzug. Der Algorithmus wurde Anfang der 1960er Jahre von Paul de Faget de Casteljau bei Citroën entwickelt.

Ziel ist ein stabiles und schnelles Verfahren zur

- Berechnung der Werte  $p(t)$  einer Bezier-Kurve durch fortgesetzte Konvexkombination und
- Zerlegung einer Bézierkurve in Teilkurven vom gleichen Grad.
- Sowie Berechnung neuer Stützpunkte.

$$P_i^j(t) = P_i^{(j-1)}(1-t) + tP_{(i+1)}^{(j-1)} \quad \left\{ \begin{array}{l} j = 1, 2, \dots, n \\ i = 0, 1, \dots, n-j \end{array} \right. \quad (4.6)$$

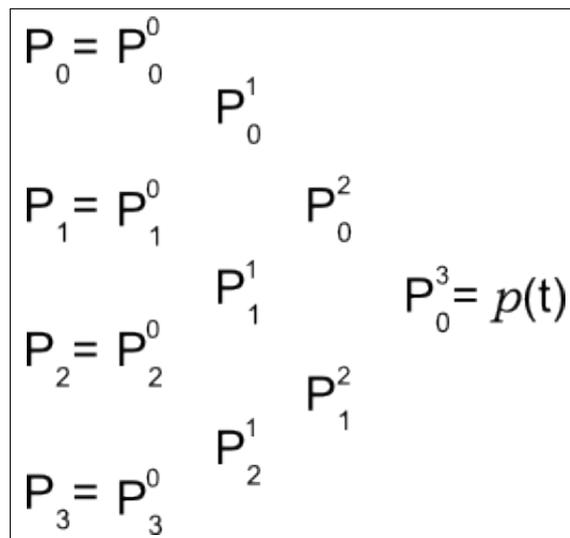
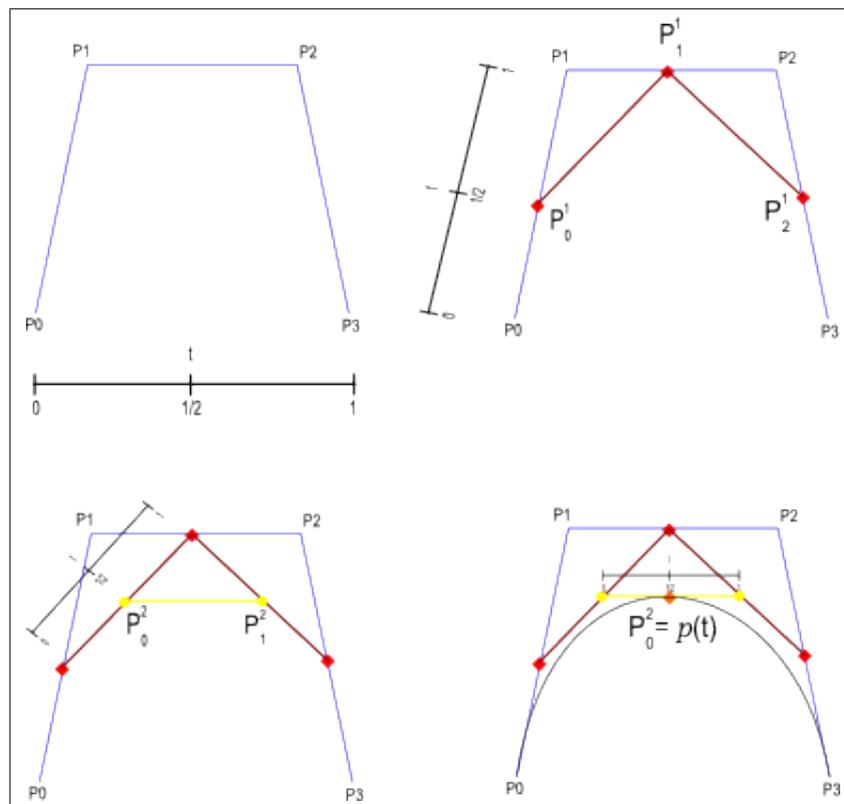
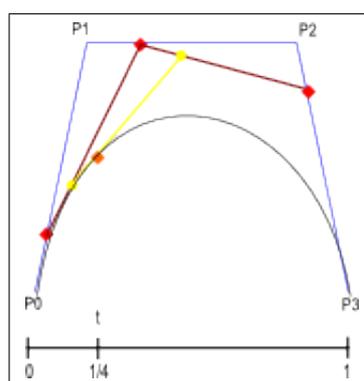


Abbildung 4.5: Schema von de Casteljau für  $n = 3$

Abbildung 4.6: Kubische Bézierkurven Konstruktion mit de Casteljau für  $t = 1/2$ Abbildung 4.7: Kubische Bézierkurven Konstruktion mit de Casteljau für  $t = 1/4$

# 5 Realisierung

## 5.1 Aufbau der Strecke

Abbildung 5.1 zeigt das Photo der Teststrecke, dessen Abmessungen in der Abbildung 5.2 dargestellt wird. Diese wird als Vorlage für die Simulation der Strecke zur Steuerung des Fahrzeuges verwendet.



Abbildung 5.1: Teststrecke

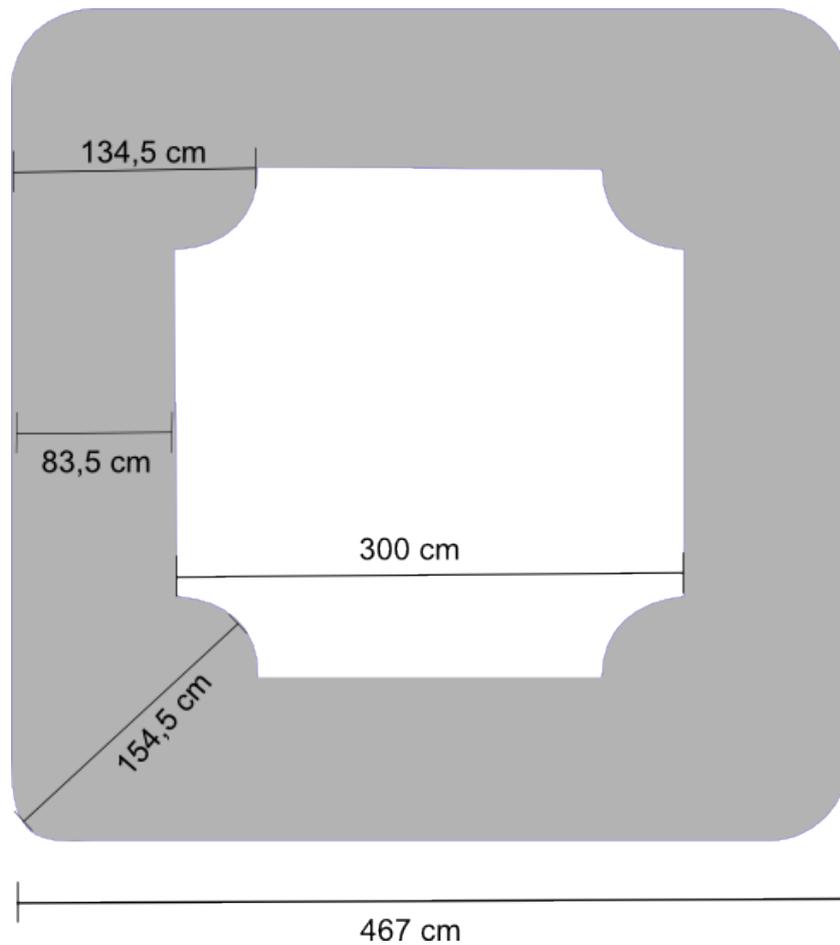


Abbildung 5.2: Abmessungen der Teststrecke

Mit Hilfe eines Bildbearbeitungsprogramms Inkscape (siehe Kapitel 3.1) werden die Stützpunkte zuerst festgelegt. Der in dem Inkscape integrierte Bézier Algorithmus ermöglicht die Glättung der Kurvenfahrt. Die Abbildung 5.3 zeigt die Schritte zum Aufbau der Kurvenfahrt mit Hilfe von Bézier, bis sich eine vollständige Trajektorie (Abbildung 5.4) ergibt.

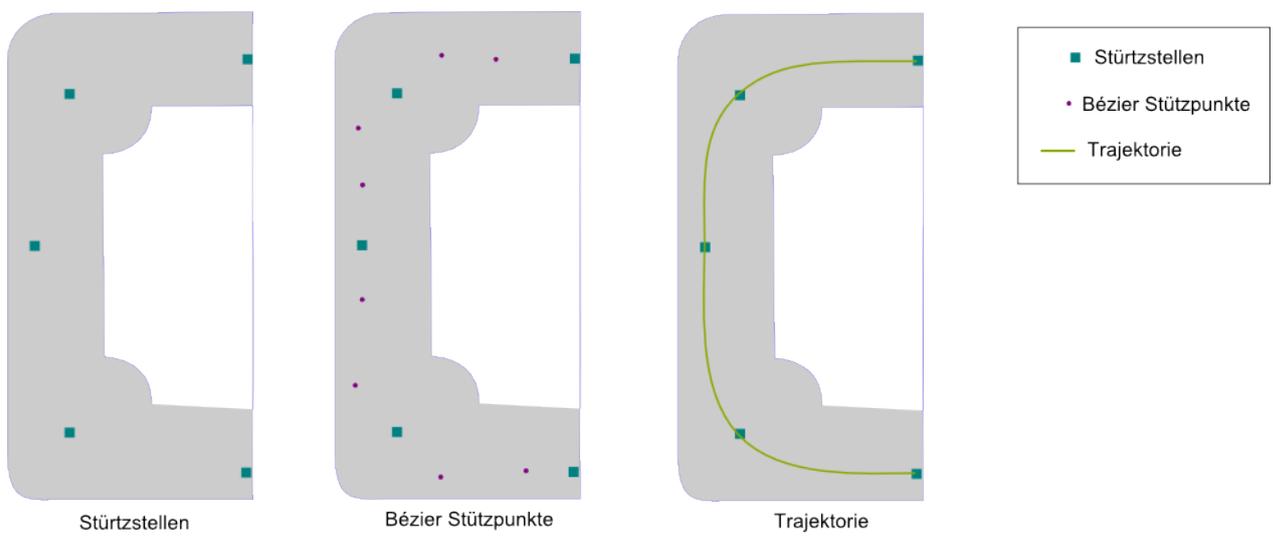


Abbildung 5.3: Aufbau der Trajektorie mit Bézier

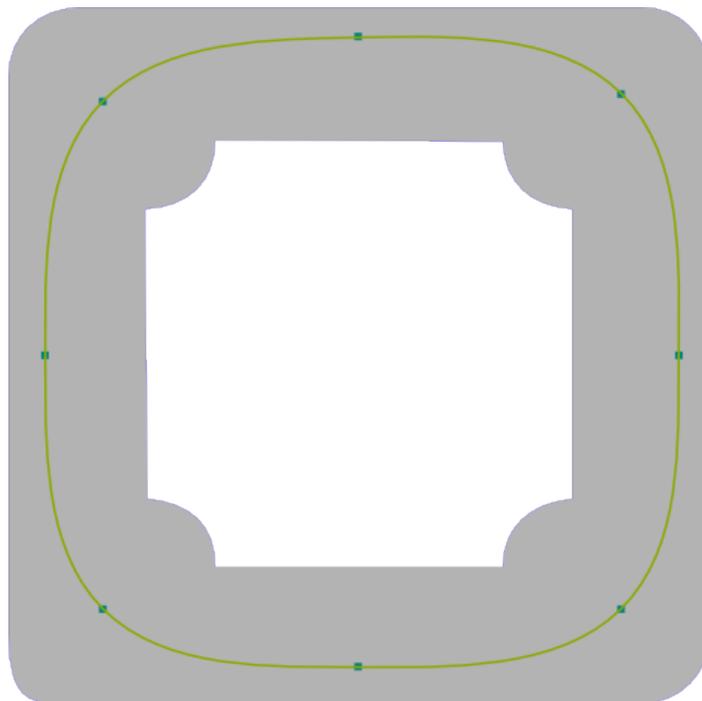


Abbildung 5.4: Die Trajektorie

Da Inkscape die SVG-Sprache (siehe Kapitel 3.1) unterstützt, können die Koordinaten der Punkte ausgelesen werden. (siehe Listing 5.1 der Punkte).

```
<path
d="M 237.32491,26.388819
  C 180.38893,26.509202
  110.27606,23.004332
  69.968299,63.264004
  29.660542,103.52368
  33.093114,172.85918
  33.093114,229.67509
  c 0,56.81591
  -3.299731,126.23617
  36.875185,166.41109
  40.174911,40.17492
  109.714631,37.06353
  166.411091,36.87519
  56.69646,-0.18834
  125.37527,2.31913
  165.46557,-37.8207
  40.0903,-40.13983
  37.51299,-109.59518
  37.8207,-166.41109
  C 439.97337,171.91367
  443.86329,102.44578
  403.736,62.318486
  363.60871,22.191193
  294.26089,26.268436
  237.32491,26.388819 z"
id="route "
/>
```

Listing 5.1: SVG Darstellung ( Legende siehe Abbildung 5.3)

Mit der Methode aus dem Listing 5.2 können nun die Anfangs- (P0) und Endpunkte (P1) sowie die Bézier Stützpunkte (P1 und P2) ausgelesen werden. Hierzu werden folgende transformierenden Funktionen erkannt und herausgefiltert:

- Translation: Verschiebung des Ursprungs
- Skalierung: (z.B. Anpassung von Einheiten)
- Rotation: (Drehung des Koordinatensystems selbst)

```
int ReadSvgFile::getControlPoint(vector<Control_Point>&
    out_control_point)
```

Listing 5.2: Funktion Auslesen der Stützstelle und der Bézier Stützpunkte

P0		P1		P2		P3	
x	y	x	y	x	y	x	y
237	441	180	441	110	444	69	404
69	404	29	364	33	295	33	238
33	238	33	182	30	112	69	72
69	72	109	32	178	35	235	36
235	36	291	36	360	34	400	73
400	73	440	113	437	182	437	239
437	239	439	296	443	365	403	405
403	405	363	445	294	441	237	441

Tabelle 5.1: Punkte der Stützstelle( $P_0, P_3$ ) sowie der Bézier-Stützpunkte( $P_1, P_2$ ) der Trajektorie ( Legende siehe Abbildung 5.3)

## 5.2 Berechnung der Bezugspunkte

Die Bestimmung der Bezugspunkte ist entscheidend für die spätere Berechnung der Zielpunkte, die ihrerseits zur Fahrorientierung des Fahrzeug dienen. Die Formel aus der Theorie von de Casteljaou (siehe Formel 4.6) kann in den Gleichungen aus Abbildung 5.1 ausgedrückt werden, woraus zu erkennen ist, dass die Bezugspunkte  $P_3^0$  aus den Bezier-Stützpunkten  $P_0, P_1, P_2, P_3$  bestimmt werden können.

$$P_0^0 = P_0$$

$$P_0^1 = P_1$$

$$P_0^2 = P_2$$

$$P_0^3 = P_3$$

$$P_1^0(t) = (1 - t) \cdot P_0^0 + t \cdot P_0^1$$

$$P_1^1(t) = (1 - t) \cdot P_0^1 + t \cdot P_0^2$$

$$P_1^2(t) = (1 - t) \cdot P_0^2 + t \cdot P_0^3$$

$$P_2^0(t) = (1 - t) \cdot P_1^0 + t \cdot P_1^1$$

$$P_2^1(t) = (1 - t) \cdot P_1^1 + t \cdot P_1^2$$

$$P_3^0(t) = (1 - t) \cdot P_2^0 + t \cdot P_2^1 \quad (5.1)$$

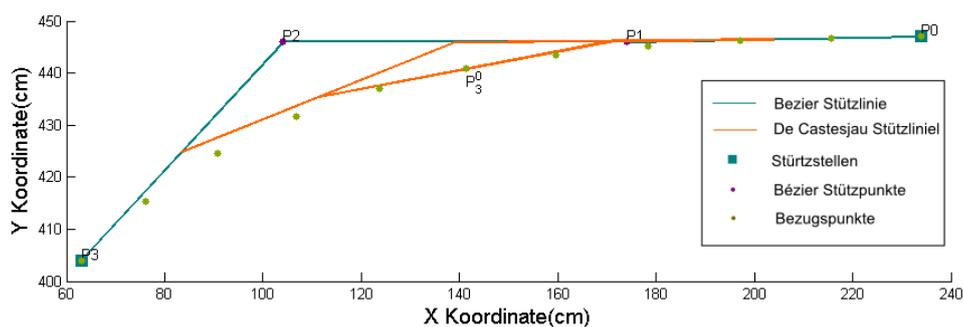


Abbildung 5.5: Bezugspunkte mit Hilfe der Theorie von de Casteljaou

### 5.3 Länge der Gesamtstrecke

Zur genauen Berechnung eines Kurvenstücks ist das Auflösen eines Integrals (Formel 5.2) vonnöten. Wie jedoch in der Bachelorarbeit von Biglari (Biglari 2009) bewiesen, lässt sich dieser Ansatz nicht umsetzen. Daher wird ein Annäherungsverfahren mit Hilfe des Satzes des Pythagoras (Formel 5.3) eingesetzt. Je kleiner der Abstand zwischen zwei Punkten gewählt wird, desto genauer kann die Länge bestimmt werden.

$$L = \int_a^b \sqrt{\left(\frac{dx(t)}{dt}\right)^2 + \left(\frac{dy(t)}{dt}\right)^2} dt \quad (5.2)$$

$$l_k = \sqrt{(\Delta x)^2 + (\Delta y)^2} \quad (5.3)$$

Abbildung 5.6 veranschaulicht die Berechnung des Abstands mit Pythagoras:

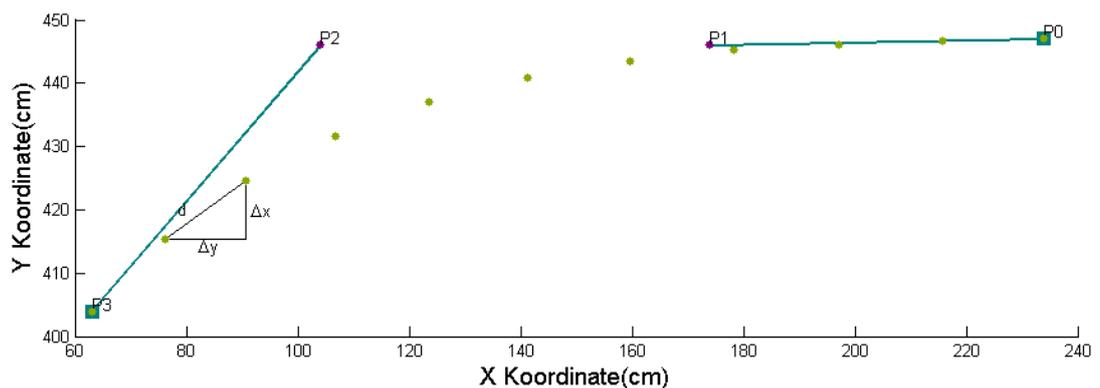


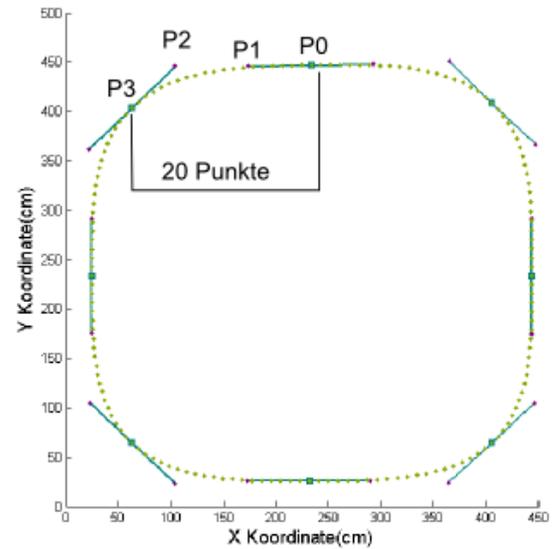
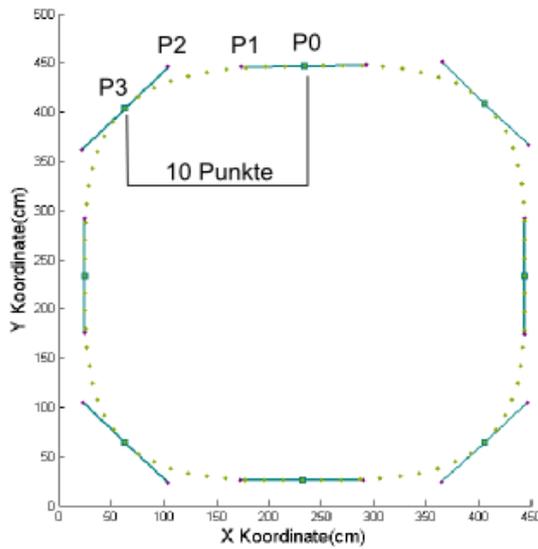
Abbildung 5.6: Berechnung des Abstands zwischen zwei Bezugspunkten mit  $\Delta t=0.1$   $n = 10$

Die Berechnung der Position der Bezugspunkte auf der X,Y-Koordinate in der Abbildung 5.6 wurde in dem vorherigen Kapitel 5.2 beschrieben.

Die Gesamtstrecke kann aus der Summierung der Längen ermittelt werden.

$$l_{(\text{Gesamtstrecke})} = \sum_{i=1}^k l_i, 1 < k < n \quad (5.4)$$

Die Genauigkeit der Berechnung der Gesamtstrecke nimmt mit der Verringerung des Abstands zwischen den Bezugspunkten zu (siehe Tabelle 5.2). Die hier erreichte Genauigkeit von 1mm wird als gegebene Toleranz angenommen.

Abbildung 5.7: Bezugspunkte  $\Delta t=0.1$   $n = 10$ Abbildung 5.8: Bezugspunkte  $\Delta t=0.05$   $n = 20$ 

$\Delta t$ Intervall	Anzahl Bezugspunkte	Durchschnittsabstand zwischen zwei Bezugspunkten(cm)	Länge der Gesamtstrecke(cm)
0.1	80	18.0942	1447.53
0.02	400	3.61123	1444.49
0.004	2000	0.723686	1447.37
0.0008	10000	0.144794	1447.94
0.00016	50000	0.0289611	1448.05
0.000032	250000	0.00579231	1448.08

Tabelle 5.2: Die Genauigkeit der Streckenlänge(cm) von Anzahl der Punkte(n)

## 5.4 Abstand zwischen zwei Zielpunkten

Die Bestimmung des Abstands zwischen zwei Zielpunkten ist entscheidend für die Lenkungssteuerung des Fahrzeugs. Der Abstand hängt von der Fahrzeugsgeschwindigkeit und der Taktrate, zu welchem Zeitpunkt das Lenkungsbehl an das Fahrzeug gesendet wird.

Die Formel lässt sich wie folgt beschreiben:

$$\text{Abstand} = \text{Geschwindigkeit} \cdot \text{Taktrate} = 2\text{m/s} \cdot 10\text{ms} = 2\text{cm} \quad (5.5)$$

- Abstand = Abstand zwischen zwei Zielpunkten
- Taktrate = in 10 ms Takt wird jedesmal neues Lenkungsbehl an das Fahrzeug gesendet
- Geschwindigkeit = konstante Fahrzeugsgeschwindigkeit 2 m/s

## 5.5 Berechnung der Zielpunkte

Wie in Kapitel 5.2 bereits erwähnt, dienen die Bezugspunkte zur Berechnung der Zielpunkte. Aus der Tabelle 5.2 ist zu ersehen, dass je kleiner das Intervall  $\Delta t$  und somit auch der Abstand zwischen zwei Bezugspunkten sind, desto genauer kann die Position der Zielpunkte auf der X,Y-Koordinate bestimmt werden.

Da der Abstand zwischen zwei Zielpunkten 2cm betragen soll (siehe Kapitel 5.4), kann die Position der Zielpunkte aus Summierung des Abstandes zwischen zwei Bezugspunkten (siehe Tabelle 5.2) bestimmt werden.

Die Position der Zielpunkte kann nun wie in Abbildung 5.9 dargestellt werden.

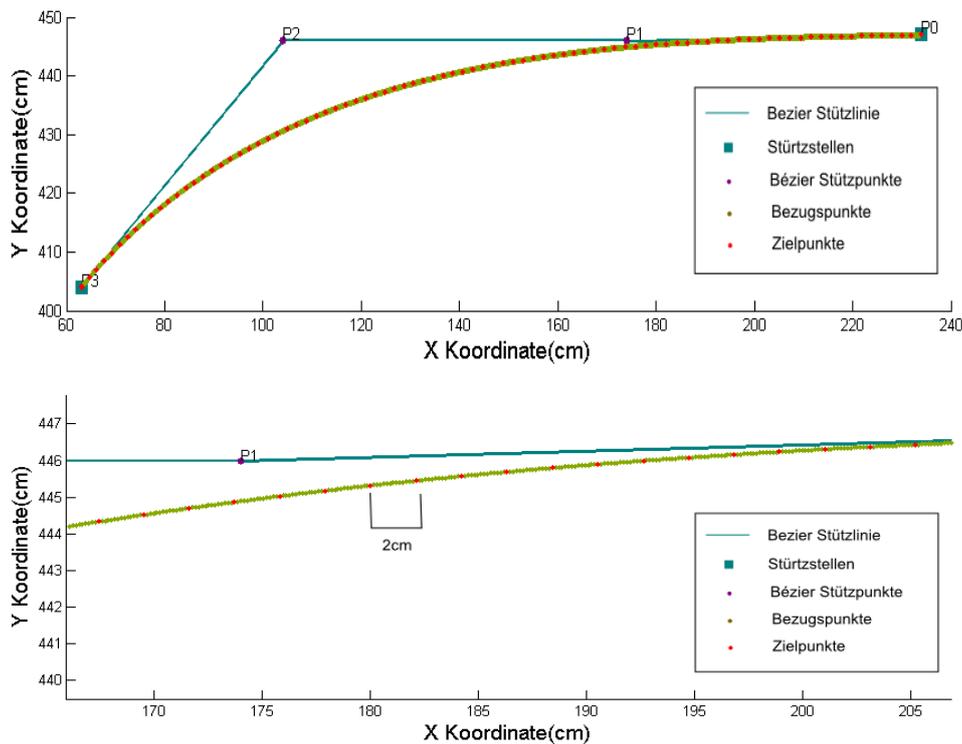


Abbildung 5.9: Bestimmung der Zielpunkte

## 5.6 Berechnung der Teilstrecke

Die Teilstrecke besteht aus neuen Zielpunkten, die kontinuierlich während der Fahrt neu berechnet werden. Mit Hilfe dieser Punkte kann das Fahrzeug auf seiner Spur gehalten oder bei Abweichung wieder auf seine Spur gelenkt werden. (siehe Abbildung 5.10)

Die aktuelle Position des Fahrzeugs ( $x, y, \delta$ ) wird vom Laserscanner erfasst. Da der Laserscanner nur alle 100 ms Rückmeldung über die Fahrzeugposition aussendet (Taktrate 100 ms) und das Fahrzeug das Steuersignal für die Lenkung alle 10 ms (Taktrate 10 ms) erhält, hat das Fahrzeug in 100 ms 10 Zielpunkte zurückgelegt. Das würde bedeuten, dass nach jeder Rückmeldung vom Laserscanner mindestens 10 neue Zielpunkte berechnet werden, um das Fahrzeug auf seiner Fahrspur zu halten und zu lenken. Dieses wird in Abbildung 5.11 dargestellt. Die Bézier-Stützpunkte von de Casteljau gibt die Richtungsinformation für die Bestimmung der neuen Zielpunkte vor.

Die benötigte Zeit des in dieser Arbeit vorgestellten Algorithmus zur Berechnung der nächsten 10 Lenkbefehle beträgt maximal 5ms und liegt somit in der Anforderung.

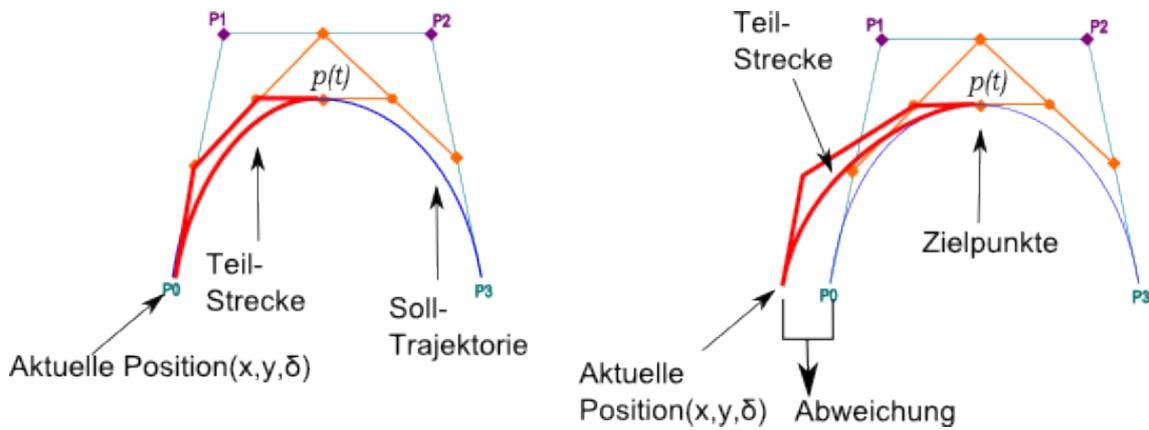


Abbildung 5.10: Anwendung vom de Casteljau

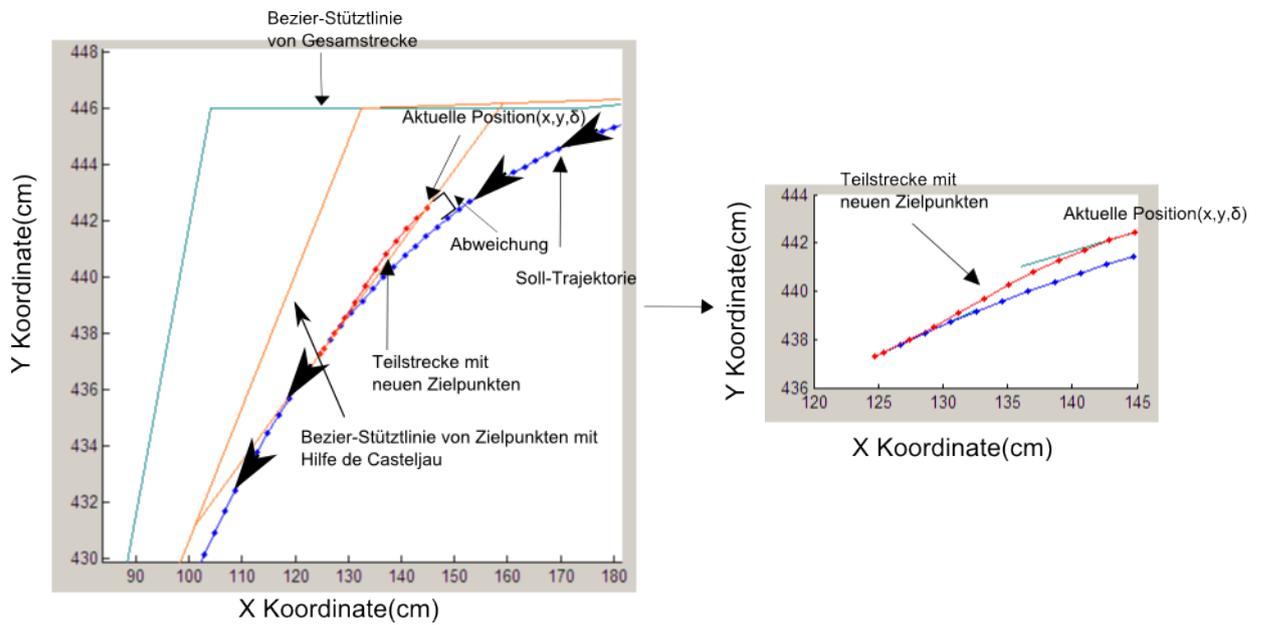


Abbildung 5.11: Teilstrecke mit neuen Zielpunkten mit Hilfe der Bézier-Stützpunkte von de Casteljau

## 5.7 Berechnung der Krümmung der Zielpunkte

Im Spezialfall einer ebenen Kurve können folgende Formeln verwendet werden. Bei ebenen Kurven kann außerdem die Krümmung mit einem Vorzeichen definiert werden. Eine Linkskurve hat eine positive Krümmung, eine Rechtskurve negative.

Nach ([Bronstein 1995 p197](#)) kann die Krümmung einer Kurve wie folgt berechnet werden.

$$\kappa = \frac{x' \cdot y'' - x'' \cdot y'}{\sqrt{((x')^2 + (y')^2)^3}} \quad (5.6)$$

$$x(t) = (1 - t)^3 \cdot x_0 + 3t(1 - t)^2 \cdot x_1 + 3t^2(1 - t) \cdot x_2 + t^3 \cdot x_3 \quad (5.7)$$

$$y(t) = (1 - t)^3 \cdot y_0 + 3t(1 - t)^2 \cdot y_1 + 3t^2(1 - t) \cdot y_2 + t^3 \cdot y_3 \quad (5.8)$$

Die Gleichung für x ist aus der Formel 5.7 und die für y aus der Formel 5.8, der Bezier-Terme zu entnehmen.

Nach Umformung der Bezier-Formel nach t folgt:

$$\begin{aligned} x(t) &= x_0 + (-3x_0 + 3x_1) \cdot t + (3x_0 - 6x_1 + 3x_2) \cdot t^2 + (-x_0 + 3x_1 - 3x_2 + x_3) \cdot t^3 \\ y(t) &= y_0 + (-3y_0 + 3y_1) \cdot t + (3y_0 - 6y_1 + 3y_2) \cdot t^2 + (-y_0 + 3y_1 - 3y_2 + y_3) \cdot t^3 \end{aligned} \quad (5.9)$$

Mit den Definitionen

$$\begin{aligned} X_0 &= x_0 \\ X_1 &= (-3x_0 + 3x_1) \\ X_2 &= (3x_0 - 6x_1 + 3x_2) \\ X_3 &= (-x_0 + 3x_1 - 3x_2 + x_3) \end{aligned}$$

$$\begin{aligned} Y_0 &= y_0 \\ Y_1 &= (-3y_0 + 3y_1) \\ Y_2 &= (3y_0 - 6y_1 + 3y_2) \end{aligned}$$

$$Y_3 = (-y_0 + 3y_1 - 3y_2 + y_3) \quad (5.10)$$

in Gleichung 5.9 eingesetzt ergibt sich:

$$\begin{aligned} x(t) &= X_0 + X_1 \cdot t + X_2 \cdot t^2 + X_3 \cdot t^3 \\ y(t) &= Y_0 + Y_1 \cdot t + Y_2 \cdot t^2 + Y_3 \cdot t^3 \end{aligned} \quad (5.11)$$

Die Differenz aus Gleichung 5.6 (Zähler) ist hierdurch:

$$x' \cdot y'' - x'' \cdot y' = 6(X_2 \cdot Y_3 + X_3 \cdot Y_2) \cdot t^2 + 6(X_1 \cdot Y_3 - X_3 \cdot Y_1) \cdot t + 2(X_1 \cdot Y_2 - X_2 \cdot Y_1) \quad (5.12)$$

Die Summe unter der Wurzel in der Formel 5.6(Nenner) mit der Definition 5.10 lässt sich nun umformen zu:

$$\begin{aligned} & (x')^2 + (y')^2 \\ &= 9 \cdot (X_3^2 + Y_3^2)t^4 + 12 \cdot (X_3 \cdot X_2 + Y_3 \cdot Y_2)t^3 \\ &+ (6 \cdot (X_3 \cdot X_1 + Y_3 \cdot Y_1)) + (4 \cdot ((X_2 \cdot X_2) + (Y_2 \cdot Y_2)))t^2 \\ &+ 4 \cdot ((X_2 \cdot X_1) + (Y_2 \cdot Y_1))t + (X_1^2 + Y_1^2) \end{aligned} \quad (5.13)$$

Mit den Definitionen

$$\begin{aligned} Z_4 &= 9 \cdot (X_3^2 + Y_3^2) \\ Z_3 &= 12 \cdot (X_3 \cdot X_2 + Y_3 \cdot Y_2) \\ Z_2 &= 6 \cdot (X_3 \cdot X_1 + Y_3 \cdot Y_1) + 4 \cdot ((X_2^2) + (Y_2^2)) \\ Z_1 &= 4 \cdot (X_2 \cdot X_1) + (Y_2 \cdot Y_1) \\ Z_0 &= X_1^2 + Y_1^2 \end{aligned} \quad (5.14)$$

lässt sich der Nenner in Formel 5.6 ausdrücken als:

$$\sqrt{(x')^2 + (y')^2} = \sqrt{Z_4 \cdot t^4 + Z_3 \cdot t^3 + Z_2 \cdot t^2 + Z_1 \cdot t + Z_0} \quad (5.15)$$

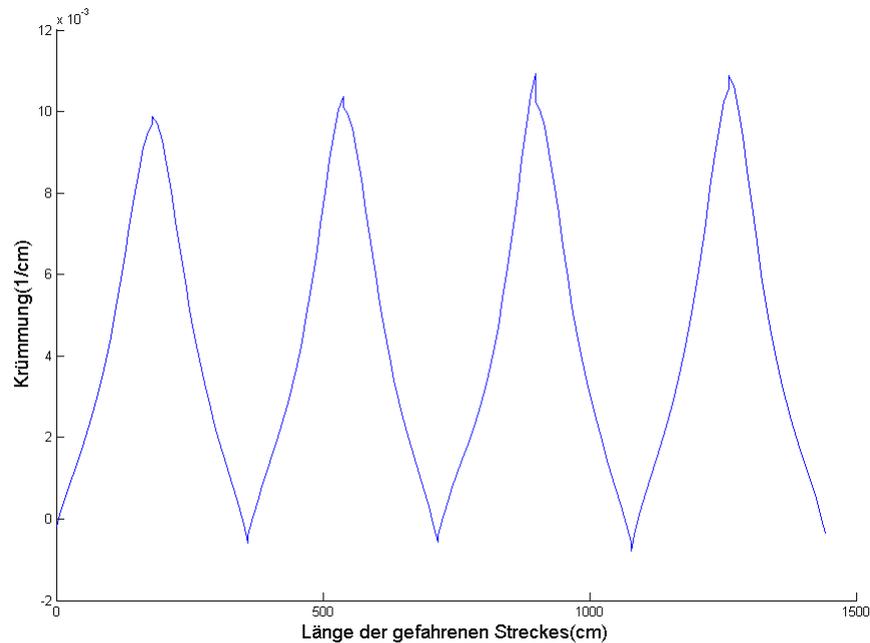


Abbildung 5.12: Krümmungsverlauf der gefahrenen Strecke

## 5.8 Berechnung des Lenkwinkels

Mit Hilfe der Krümmung (siehe Kapitel 5.7) kann nun der Lenkwinkel des Fahrzeugs errechnet werden.

Für die Lenkung wurde das „Ein-Spur Modell“ vom Ackermann-Lenkwinkel (siehe [Grundlagen der Kraftfahrzeugtechnik S.248](#)) bei sehr langsamer Kreisfahrt mit vielen weiteren Vereinfachungen wie Vernachlässigung der Umfangskräfte am Rad und der Fahrwiderstände verwendet. Mit dieser Vernachlässigung wird die Fahrzeugsgeschwindigkeit konstant gehalten.

$$\tan \delta = \frac{l}{R_h} \quad (5.16)$$

$\delta$  = Lenkwinkel

$l$  = Abstand zwischen zwei Rädern des Fahrzeugs (440mm siehe Kapitel 3.2.1)

$R_h$  = Radius der Fahrstrecke

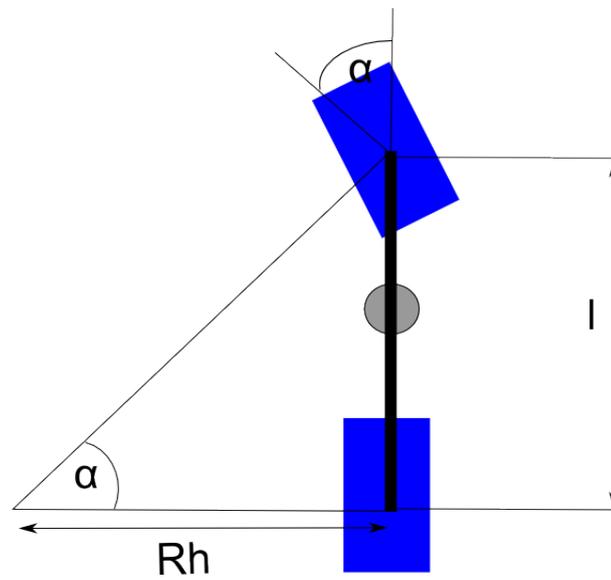


Abbildung 5.13: Ein-Spur Fahrzeugmodell

Wobei der Radius der Kehrwert der Krümmung  $\kappa$  ist.

$$\kappa = \frac{1}{R_h} \quad (5.17)$$

Daraus folgt:

$$\delta = \arctan(l \cdot \kappa) \quad (5.18)$$

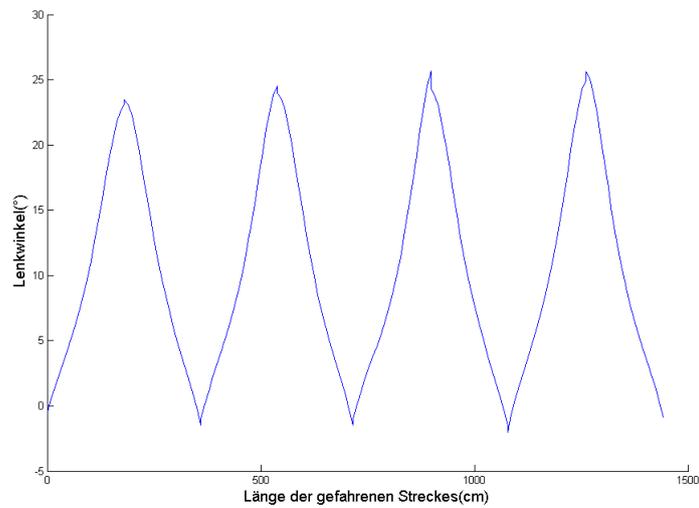


Abbildung 5.14: Lenkungsverlauf der gefahrenen Strecke

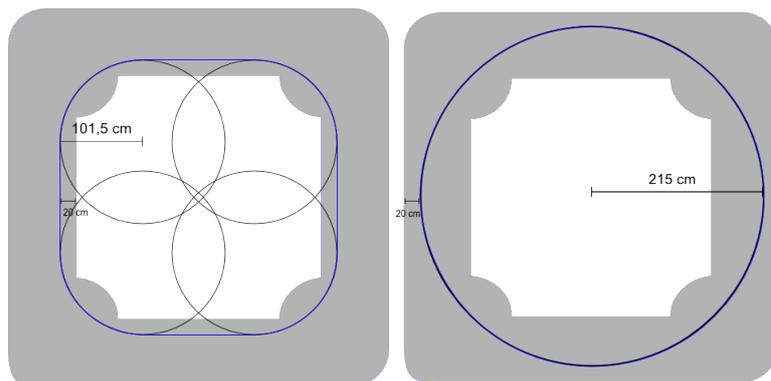


Abbildung 5.15: Abhängigkeit des Radius von dem Lenkwinkel

Da das Fahrzeug einen maximalen Lenkwinkel von  $26^\circ$  aufweist (siehe Kapitel 3.2.1), beträgt somit der dazugehörige Radius laut Formel 5.16 101,5 cm (siehe linkes Bild in Abbildung 5.15).

## 5.9 Simulation mit MATLAB

Alle Daten, Messwerte sowie die Telemetrie lassen sich mit MATLAB simulieren und visualisieren.

Das Hauptfenster (Abbildung 5.16) zeigt alle wichtigen Daten der Spüführung, wie Trajektorie, Lenkung sowie Krümmung und Abstand zwischen den Zielpunkten.

In Trajektorie sind sämtliche Zielpunkte als blaue Punkte dargestellt. Die von dem Anwender vorgegebenen Anfang-Stützstelle sind zu einem grünen Kontrollpolygon verbunden, aus dem die Bezierkurve generiert wurde. Die rote Teilstrecke wird durch das Casteljau - Kontrollpolygon (Farbe Orange) bestimmt.

Die rechte Hälfte stellt die Lenkung sowie Krümmung dar. Weiterhin wird hier der Abstand zwischen den aufeinander folgenden Zielpunkten gezeigt, wodurch sich die Abweichung ablesen lässt.

Das 2.Fenster (Abbildung 5.17) zeigt das Verhältnis von Soll-Werten (blau) zu den Ist-Werten (rot), bezogen auf den Lenkwinkel, die aktuell zu fahrende Teilstrecke sowie den Abstand zwischen den aufeinander folgenden Zielpunkten.

Die Tabellen veranschaulichen das Verhältnis von Ist- zu Soll-Werten der Teilstrecke sowie des Lenkwinkels in Zahlendarstellung.

Im rechten Bereich des Fensters wird dargestellt, wie stark die Ist-Länge von der Soll-Länge der Strecke abweicht. Weiterhin ist hier die Menge der zur Berechnung verwendeten Zielpunkte angegeben.

Mit Hilfe der Eingabemaske, ist es möglich alle Parameter (wie z.B. die Richtung oder die Abweichung) der Simulation einzustellen, um beliebige Testszenarien erzeugen zu können. Mit diesen kann schnell geprüft werden, ob die eingegebene Strecke mit den gewählten Parametern fahrbar ist.

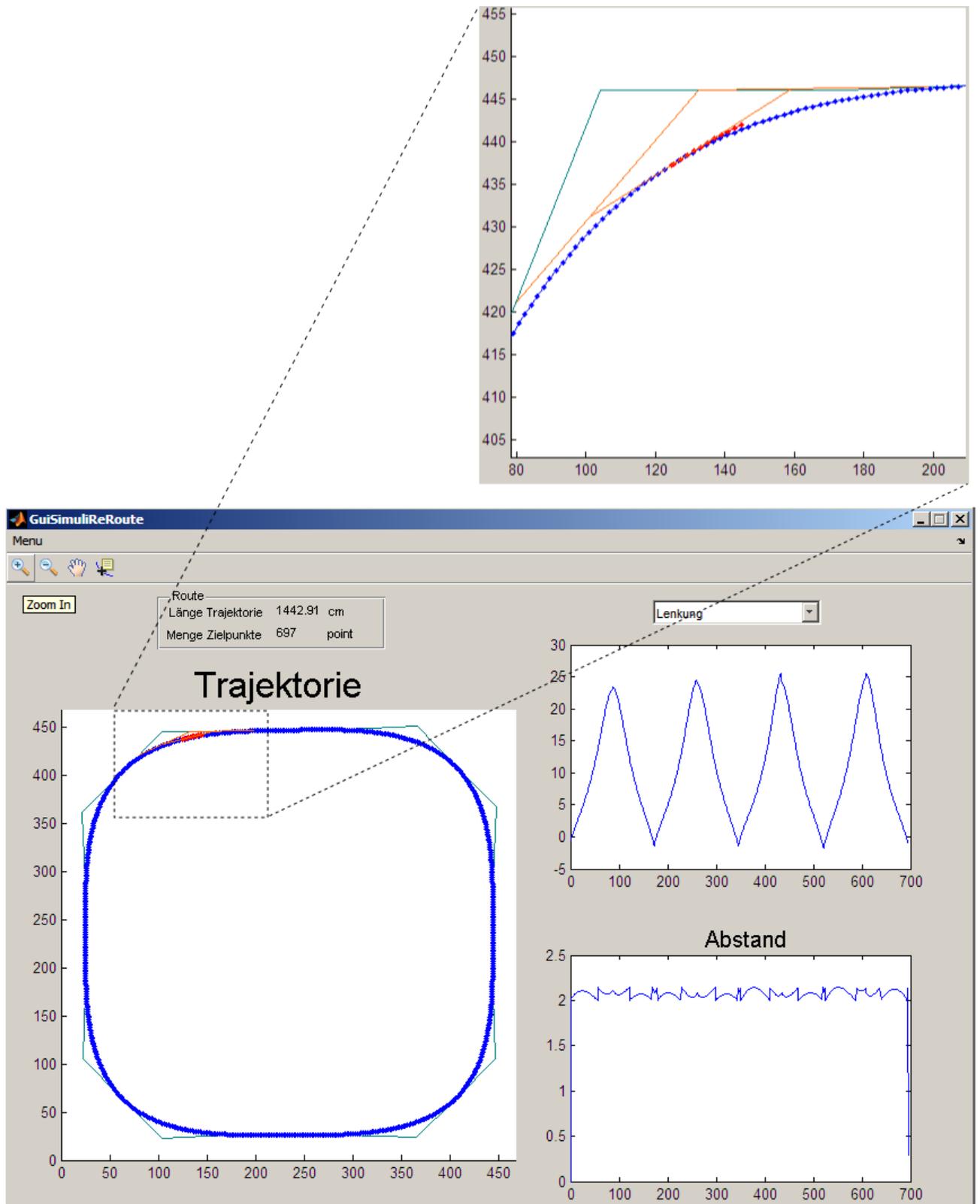


Abbildung 5.16: Simulation der Trajektorie und Visualisierung der Daten

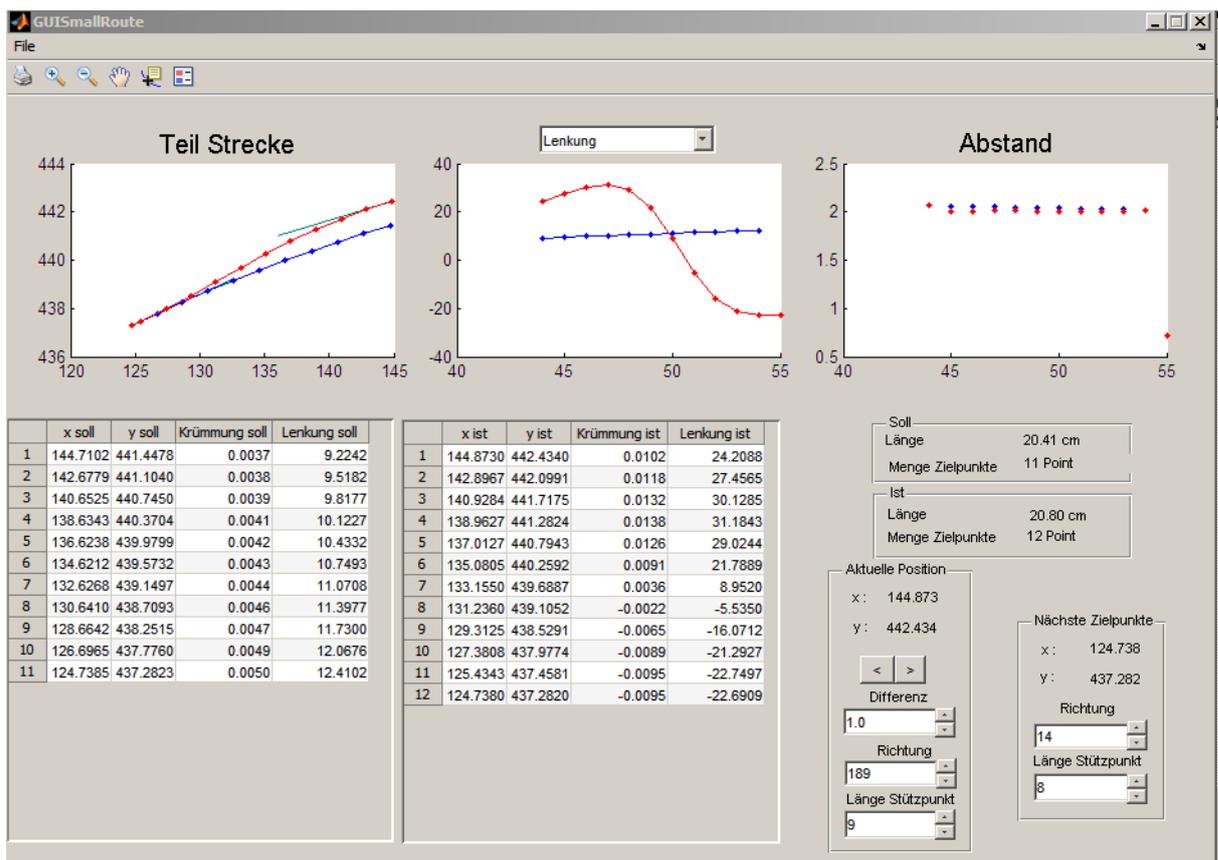


Abbildung 5.17: Simulation Soll- und Ist-Zielpunkte sowie der Lenkung und Krümmung

## 6 Zusammenfassung

In dieser Arbeit wurde eine Software zur Route- bzw. Trajektorieberechnung eines Fahrzeugs entwickelt mit Hilfe von Algorithmen für Berechnung der Stütz- und Zielpunkten sowie Krümmungen und Länge der Strecke. Ein Steuersystem für eine Überwachung und Prozessdatenerfassung vom Fahrzeug wurde realisiert. Dieses ermöglicht eine Steuerung des Fahrzeugs, Auswertung, Speicherung sowie Visualisierung der Telemetriedaten vom Leitstand.

Bei dem Absatz Hardware und Grundlagen wurden die verwendeten Hardwares bzw. Geräte wie z.B. Laserscanner, Fahrzeug, usw. beschrieben. Die Grundlagen über Bézier und De Casteljau als Basis für die Berechnung der Trajektorie sowie der Zielpunkten wurden erläutert. Es wurde auch auf die Spezifikation SVG und die Kommunikation über USB als Datenaustausch zwischen dem Fahrzeug und Anwender eingegangen.

Bei der Realisierung dieser Arbeit wurden verschiedene Aufgaben für die Steuerung gelöst. Die Fahrstrecke des Fahrzeugs wurden mit Hilfe von einem Bildbearbeitungssystem Inskape und den Algorithmen simuliert. Die Stützstellen und die Bezierstützpunkte lassen sich über eine SVG-Spezifikation ermitteln. Mit Berechnung des Abstandes zwischen beliebigen Zielpunkten, Länge der Gesamtstrecke, der Krümmung und des Lenkwinkels sowie Berechnung der Zielpunkten kann das Fahrzeug so gesteuert werden, so daß es autonom die Spur hält.

Abschließend werden alle relevanten Daten und Messwerte mit MATLAB überwacht und visualisiert.

# Literaturverzeichnis

- [Carsten Schreiber 2007] Telemetrie in mobilen Systemen mit TCP/IP,WLAN und Feldbus Technologien Hochschule für Angewandte Wissenschaften Hamburg, Studienarbeit, 2007.
- [Arbaoui 2005] ARBAOUI, Sadiq:Software in-the-Loop Simulation für Mobile Systeme,Hochschule für Angewandte Wissenschaften Hamburg, Diplomarbeit, 2005
- [Hübener 2004] HÜBENER, Burkhard:Remotesteuerung eines Modellbaulastwagens - Schaffung einer Erprobungsplattform für Experimente mit Fahrerassistenzsystemen, Hochschule für Angewandte Wissenschaften Hamburg, Studienarbeit, 2004.
- [Biglari 2009] Biglari,Ahmad: Modellierung der Trajektorie eines autonomen Fahrzeuges mit Bezierkurven Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2009.
- [Bronstein 1995 p197] BRONSTEIN, Ilja N.: Taschenbuch der Mathematik. Harry Deutsch Verlag Thun, 1995 p197
- [Wikipedia] <http://de.wikipedia.org/wiki/De-Casteljau-Algorithmus>
- [Wikipedia] <http://de.wikipedia.org/wiki/Bézierkurve>
- [Self SVG] <http://www.selfsvg.info/?section=3.5>
- [USB 2.0 Handbuch für Entwickler] USB 2.0 Handbuch für Entwickler von Jan Axelson
- [Grundlagen der Kraftfahrzeugtechnik S.248] Grundlagen der Kraftfahrzeugtechnik von Karl-Ludwig Haken ISBN-10:3-446-22812-8

# Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-TI-BM ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Hamburg, 28. April 2011

Ort, Datum

Unterschrift