

# Bachelorarbeit

Sven Geibert

Gyroskopische Lenkregelung für ein autonomes  
Modellfahrzeug

Sven Geibert

Gyroskopische Lenkregelung für ein autonomes  
Modellfahrzeug

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

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

Abgegeben am 4. Oktober 2007

**Sven Geibert**

**Thema der Bachelorarbeit**

Gyroskopische Lenkregelung für ein autonomes Modellfahrzeug

**Stichworte**

Gyroskop, Regelung, Modellfahrzeug, PID, Microcontroller, A/D-Wandlung, Lenkung, autonom

**Kurzzusammenfassung**

Damit Fahrzeuge sich autonom bewegen können, müssen diese dazu in der Lage sein ihre Richtung zu erkennen und zu korrigieren. Diese Arbeit beschäftigt sich mit der Implementierung einer Lenkregelung für ein Modellfahrzeug. Mit Hilfe eines Gyroskops werden Richtungsänderungen gemessen und diese mit der Regelungs-Software verarbeitet. Die Software korrigiert die Richtung des Fahrzeugs durch das kontinuierliche Einstellen des entsprechenden Lenkeinschlags.

**Sven Geibert**

**Title of the paper**

Gyrospherical steering control for an autonomous model vehicle.

**Keywords**

Gyroscope, steering, control, model vehicle, PID, microcontroller, a/d-conversion, autonomous

**Abstract**

In order for vehicles to move autonomously, they have to be able to sense and correct their direction of movement. This study deals with the implementation of a steering control for a model vehicle. Using a gyroscope, changes in direction are measured and then processed by a control software. The software corrects the vehicle's direction by continuously setting the correct steering angle.

## **Danksagung**

Die Person, die das alles erst ermöglicht hat, ist meine Mutter. Ihr möchte ich ganz besonders für die Unterstützung danken.

Des Weiteren möchte ich mich bei meiner Freundin Nicole bedanken. Sie hat mich während des Studiums begleitet und alle Höhen und Tiefen in dieser Zeit miterlebt.

Und natürlich gilt mein Dank dem intelliTruck-Team um Prof. Dr. rer. nat. Stephan Pareigis, mit dem ich viele angeregte Diskussionen geführt und Ideen ausgetauscht habe. Ich hoffe, dass ich einen Beitrag zum Gelingen dieses Projekts beisteuern konnte.

# Inhaltsverzeichnis

<b>1 Einleitung</b> .....	<b>10</b>
1.1 Kreiselstabilisierte Systeme.....	10
1.2 Aufgabenstellung.....	10
1.3 Serienentwicklung.....	11
<b>2 Systemanalyse</b> .....	<b>16</b>
2.1 Die Kurve allgemein.....	16
2.2 Das Fahrzeug allgemein.....	17
2.2.1 Die Lenkung.....	17
2.2.2 Steuermöglichkeiten.....	19
2.2.3 Signalverarbeitung.....	19
2.3 Das Gyroskop.....	20
2.3.1 Funktionsweise.....	20
2.3.2 Verarbeitung der Daten.....	20
2.3.3 Genauigkeit und Arbeitsbedingungen.....	23
<b>3 Die Regelung</b> .....	<b>25</b>
3.1 Das Grundprinzip der Regelung.....	25
3.1.1 Das Proportionalglied.....	26
3.1.2 Das Integralglied.....	26
3.1.3 Das Differenzialglied.....	27
3.1.4 Der PID-Regler.....	27
3.2 Gewünschtes Verhalten.....	28
3.3 Reglerauslegung.....	29
3.3.1 Festlegung des P-Anteils.....	30
3.3.2 Festlegung des I-Anteils.....	30
3.3.3 Festlegung des D-Anteils.....	31
3.4 Begrenzung der Ausgangsgröße und Sollwertvorgaben.....	32
3.4.1 Begrenzung der Ausgangsgröße.....	32
3.4.2 Begrenzung der Sollwertvorgaben.....	32
3.5 Geschwindigkeitsabhängiges Verhalten.....	33
3.5.1 Berechnung des Soll-Winkel.....	34

3.5.2 Berechnung des Ist-Radius.....	35
<b>4 Implementierung.....</b>	<b>36</b>
4.1 Der allgemeine Regler.....	36
4.1.1 Die Methode PIDController::processCurrentParameter().....	37
4.1.2 Die Berechnung des Proportionalanteils.....	39
4.1.3 Die Berechnung des Differentialanteils.....	39
4.1.4 Die Berechnung des Integralanteils.....	39
4.1.5 Problematik mit dem Zeitintervall.....	39
4.2 Konvertierung der Daten.....	40
4.3 Der konkrete Regler.....	42
4.3.1 Implementierung der radiusbasierten Kurvenfahrt.....	43
4.3.2 Messung der Lenkwirkung.....	44
4.3.3 Die Methode PIDSteering::processCurrentParameter().....	45
4.3.4 Klassendiagramm für PIDSteering.....	46
4.4 Das Programm GyroControl.....	47
4.5 Integration des Gyroskops auf dem Microcontroller.....	49
4.6 Integration in das Fahrzeug.....	50
<b>5 Test.....</b>	<b>51</b>
5.1 Labortest.....	51
5.1.1 Bestimmung der maximalen Abtastrate.....	51
5.1.2 Bestimmung des optimalen fixedDelay.....	55
5.1.3 Parameter für die Offset-Elimination.....	55
5.1.4 Filtergröße für gleitenden Mittelwert.....	56
5.1.5 Spannungsfestigkeit.....	57
5.1.6 Fehlerabschätzung und Aliaseffekt.....	59
5.1.7 Integrationstest mit der Hauptsteuerung.....	61
5.2 Fahrzeugtest.....	62
5.2.1 Störungen durch Vibrationen.....	62
5.2.2 Genauigkeit der radiusbasierten Kurvenfahrt.....	63
5.2.3 Verbesserung der radiusbasierten Kurvenfahrt.....	66
<b>6 Zusammenfassung und Ausblick.....</b>	<b>69</b>
6.1.1 Verbesserungen.....	69

## **Tabellenverzeichnis**

Tabelle 1: Darstellbare Parameter in GyroControl.....	48
Tabelle 2: Zusammenhang zwischen Driftverhalten und Offset-Elimination.....	56
Tabelle 3: Zusammenhang zwischen Driftverhalten und gleitendem Mittelwert.....	57
Tabelle 4: Tabellarische Darstellung eines Spannungsabfall.....	57
Tabelle 5: Fehlmessungen bei ruhendem Gyroskop.....	60
Tabelle 6: Durchschnittlicher Fehler.....	61
Tabelle 7: Messergebnis der radiusbasierten Kurvenfahrt - unkorrigiert.....	65
Tabelle 8: Messergebnis der radiusbasierten Kurvenfahrt - korrigiert.....	67

## Abbildungsverzeichnis

Abbildung 1.1: Das neue Seitenteil mit platzsparend angeordneten Akkus.....	12
Abbildung 1.2: Das Seitenteil des alten Wagens.....	12
Abbildung 1.3: Die neue Rechnerhalterung.....	12
Abbildung 1.4: Die alte Rechnerhalterung.....	12
Abbildung 1.5: Das Rotationszentrum; zwischen Tank und Fahrakku.....	13
Abbildung 1.6: Plattform für die Rechnerhalterung.....	13
Abbildung 1.7: Der ursprüngliche Felgenmitnehmer an der Hinterachse.....	14
Abbildung 1.8: Der breitere Felgenmitnehmer an der Hinterachse.....	14
Abbildung 1.9: Die neue Sensorhalterung.....	14
Abbildung 1.10: Die alte Sensorhalterung der Hinterachse.....	14
Abbildung 2.1: Definition der Kurve für die Lenkregelung.....	16
Abbildung 2.2: Das Modell für die radiusbasierte Kurvenfahrt.....	17
Abbildung 2.3: Das Lenktrapez.....	18
Abbildung 2.4: Der Aliaseffekt (Quelle: Wikipedia).....	21
Abbildung 2.5: Illustration der Integralrechnung.....	22
Abbildung 3.1: Der Regelkreis.....	25
Abbildung 3.2: Zusammenhang zwischen Regelabweichung und Stellgröße.....	30
Abbildung 3.3: Korrektur der Ausrichtung nach einer umweltbedingten Einwirkung. .	31
Abbildung 3.4: Darstellung zur Ermittlung des Soll-Winkel und Ist-Radius.....	34
Abbildung 4.1: Klassendiagramm des allgemeinen Reglers.....	36
Abbildung 4.2: Die notwendigen Verarbeitungsschritte der Regelung.....	40
Abbildung 4.3: Klassendiagramm DataProcessor.....	41
Abbildung 4.4: Klassendiagramm der konkreten Regelung.....	46
Abbildung 4.5: Beispielscreenshot aus dem Programm GyroControl.....	47
Abbildung 4.6: Platinenlayout für das Gyroskop und einen G-Sensor.....	49
Abbildung 4.7: Mit Sensoren ausgerüsteter Microcontroller.....	50
Abbildung 4.8: Der Microcontroller unterhalb der Plattform.....	50
Abbildung 4.9: Der Microcontroller auf der Rechnerplattform platziert .....	50
Abbildung 5.1: Das Rauschen des Gyroskops mit einem Oszilloskop gemessen.....	52
Abbildung 5.2: Abtastung mit 50 Hz.....	53
Abbildung 5.3: Abtastung mit 51 Hz.....	54
Abbildung 5.4: Verhalten des Systems bei einem Spannungsabfall.....	58



Abbildung 5.5: Verlauf der Schwankungen der Winkelanzeige.....	59
Abbildung 5.6: Darstellung des Einflusses der Vibrationen.....	63
Abbildung 5.7: Darstellung des Versuchs zur radiusbasierten Kurvenfahrt.....	64
Abbildung 5.8: Illustration des systematischen Fehlers.....	66
Abbildung 5.9: Illustration des systematischen Fehlers - nach der Korrektur.....	67

# 1 Einleitung

## 1.1 Kreiselstabilisierte Systeme

Unter einem Kreisel oder Gyroskop versteht man eine Vorrichtung, die Winkeländerungen misst und diese als Daten anderen Komponenten zur Verarbeitung bereitstellen kann. Die ersten Systeme waren mechanischer Art und beruhten auf dem Prinzip der Eigenstabilität einer rotierenden, möglichst perfekt ausgewuchteten, Masse - daher der Name Kreisel.

Gyroskope dienen somit in erster Linie zur Lagebestimmung und -Korrektur. Diese Eigenschaft ist sehr nützlich für Fluggeräte, aber auch die Rollbewegung von Schiffen kann so kompensiert werden. Im Prinzip kann jedes bewegliche Objekt damit um eine oder mehrere Achsen stabilisiert und geregelt werden.

Kombiniert man ein Gyroskop mit einer genauen Streckenmessung, hat man ein System mit dem man navigieren kann. Im Zeitalter von GPS scheint dies zwar überflüssig, aber diese Art der Navigation ist für einige Spezialfälle, für die kein GPS zur Verfügung steht, sehr wichtig. Zum Beispiel um unterirdisch einen Tunnelbohrer zu navigieren oder ein U-Boot oder ein Raumfahrzeug zu steuern. Für solche Anwendungen benutzt man so genannte Faseroptische-Gyroskope. Das System beruht auf einer Glasfaserspule, durch die zwei Laserstrahlen entgegengesetzt geschickt werden. Dreht sich nun die Spule, kann eine Phasenverschiebung gemessen werden. Ein solches System arbeitet masselos und ist dadurch sehr präzise.

Deutlich weniger präzise aber dafür preiswerter und kompakter sind Gyroskope in MEMS-Technik. MEMS steht für Micro-Electrical-Mechanical-System. Darunter versteht man die Kombination aus Halbleitern mit mechanischen Komponenten. Diese Gyroskope findet man in Alltagsgegenständen wie zum Beispiel Autos und sind somit bestens für die Aufgabenstellung dieser Arbeit geeignet.

## 1.2 Aufgabenstellung

Das an der HAW ins Leben gerufene Projekt „intelliTruck“ beschäftigt sich mit der

Modifikation eines Modellautos zu einem autonomen Fahrzeug. Damit ein Fahrzeug autonom Strecken bewältigen kann, muss es dazu in der Lage sein die Richtung seiner Fahrt zu bestimmen und zu korrigieren. Voraussetzung dafür ist eine Wahrnehmung der Winkeländerung und die Kontrolle über die Lenkung. In bereits bestehenden Projekten der HAW wurde das Gyroskop ADXRS300 von Analog Devices bereits erfolgreich zur Anwendung gebracht. Nun soll es ebenfalls im intelliTruck zum Einsatz kommen.

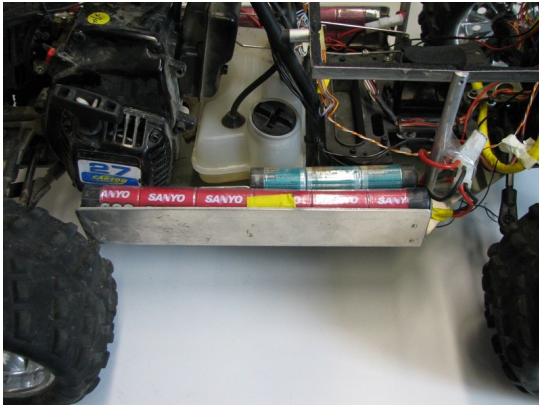
Ziel dieser Arbeit ist es ein elektronisches Kreiselssystem in den intelliTruck zu integrieren, das zusammen mit einer Regelung eine definierte Kurvenfahrt ermöglicht. Dieses System ist dann Teil einer Übergeordneten Steuerungssoftware, die darauf aufbauend weiterführende Manöver ausführen kann.

Als primäre Funktion für die Steuerungssoftware steht durch den Regler die relative Winkeländerung zur aktuellen Fahrtrichtung zur Verfügung, ebenso kann ein bestimmter Radius bis zu einem vorgegebenen Winkel gefahren werden. Als sekundäre Funktion bietet der Regler eine geschwindigkeitsabhängige Regelung der Kurvenfahrt, sodass das Fahrzeug in den Kurven bei hohen Geschwindigkeiten nicht übersteuert, das heißt, der Lenkeinschlag darf nicht zu stark sein. In diesem Zusammenhang soll es ebenfalls eine Funktion geben, die der Hauptsteuerung mitteilt, ob der Radius einer Kurve eingehalten wird.

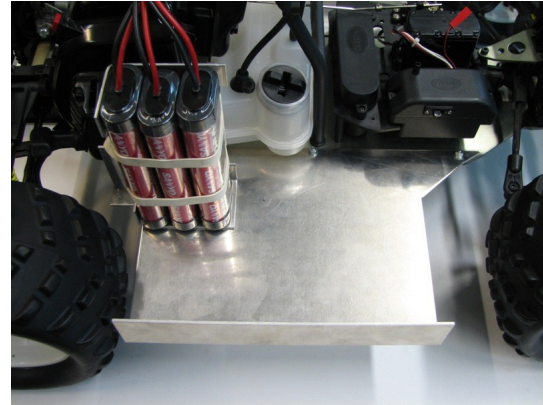
### **1.3 Serienentwicklung**

Zeitgleich mit der Einarbeitung in das Projekt wurde ein zweiter Wagen bereitgestellt. Die Modifikationen am ersten Fahrzeug sollten, wenn möglich, in verbesserter Form und genau beschrieben durchgeführt werden, damit eine Dokumentation für den Nachbau weiterer Fahrzeuge möglich wird. Weitestgehend bestehen die Modifikationen aus Halterungen für den Rechner, die Akkus und die Radsensoren sowie Verkabelungen für die Stromversorgung und den Datenaustausch. Ein großes Problem war, dass es keine Halterungen für die Microcontroller gab. Ein Umstand, der auf das geringe Platzangebot zurückzuführen ist. Da für die Weiterentwicklung ohnehin der Platzbedarf steigen wird und die Betriebssicherheit von stabil befestigten Komponenten abhängt, gab es hier die erste signifikante Änderung. Die Seitenteile, die als Akkuhalterung vorgesehen waren, wurden bis auf die Spurbreite verbreitert

und die langen 12-Zellen-Akkus wurden durch jeweils zwei 6-Zellen-Akkus ersetzt, die senkrecht gehalten werden.

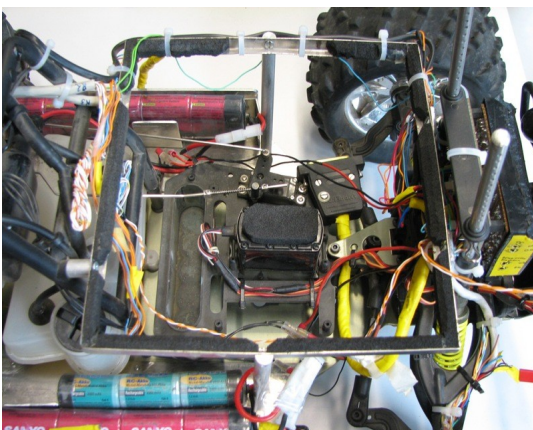


*Abbildung 1.2: Das Seitenteil des alten Wagens*

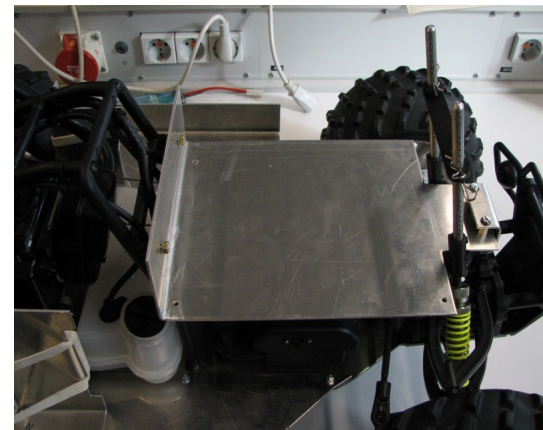


*Abbildung 1.1: Das neue Seitenteil mit platzsparend angeordneten Akkus*

Auf beiden Seiten besteht die Möglichkeit drei 6-Zellen-Akkus zu befestigen. Der dadurch gewonnene Platz ist offensichtlich. Die Halterung für den Rechner wurde ebenfalls überarbeitet. In der alten Version wird der Rechner in einen Korb gelegt. Als Untergrund dient der Lenkservo, der mit ein wenig Schaumstoff deswegen gepolstert ist. Damit das Gehäuse in den Korb passt, mussten die Winkel für die Halterung abgetrennt werden. Der Rechner wird nicht weiter gesichert, lediglich durch die Enge des Korbs wird der Rechner vom raus fallen bewahrt. In der neuen Version werden die Winkel des Gehäuses nicht abgetrennt. Diese dienen nun zur Fixierung auf einer Grundplatte. Auf einer 2 mm starken Aluminiumplatte wird der Rechner jetzt festgeschraubt. Diese Platte wird zum einen an dem Überrollkäfig montiert, zum anderen auf dem Chassis über der Lenkung befestigt.

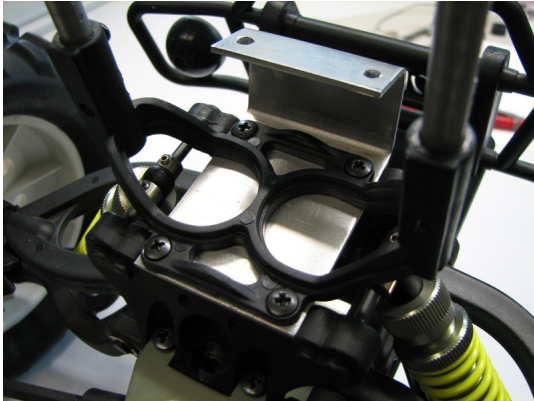


*Abbildung 1.4: Die alte Rechnerhalterung*



*Abbildung 1.3: Die neue Rechnerhalterung*

Dabei dient eine mehrfach gebogene Aluminiumplatte als Plattform zwischen Chassis und Rechnerhalterung. Die Platte wird zwischen der Karosseriehalterung und dem Chassis angebracht und die ursprüngliche Konstruktion muss nicht angepasst werden. Das Gewicht des Rechners wird durch diese Lösung direkt auf die Federung übertragen. Diese Teile sind sehr einfach herzustellen.



*Abbildung 1.6: Plattform für die Rechnerhalterung*



*Abbildung 1.5: Das Rotationszentrum; zwischen Tank und Fahrakku*

Unter der Platte entsteht dadurch ein weiterer Platz, der auf Grund seiner Lage wichtig ist, denn dort befindet sich ungefähr das Rotationszentrum des Wagens. Für weitere Experimente mit bewegungssensitiven Sensoren kann dieser Raum daher wichtig sein.

Der zweite größere Punkt sind die Radsensoren. Auf dem alten Wagen sind die Halterungen aus 2mm starken Aluminiumblech gefertigt. An den Vorderrädern besteht die Halterung aus einem gebogenen Blech. Die Hinterräder bestehen aus insgesamt drei Teilen, wobei das Messergebnis nicht optimal ist. Die Bewegung der Räder wird mit sog. Hallsensoren gemessen. Dabei wird mit einem Dauermagneten ein Feld induziert, das sich verändert, wenn ein magnetisches Objekt sich durch das Feld bewegt. Der Sensor reagiert auf eine solche Veränderung - vorausgesetzt die Geometrie des Objekts ändert sich (eine kreisrunde Stange wird nicht detektiert). An der Hinterachse messen die Sensoren die Rotation des Kardangelenks. Dieses ist aber bis auf zwei Schlitze rund. Es bot es sich an nachzubessern und eine ähnliche Lösung zu schaffen, wie an den Vorderrädern. An den Vorderrädern wird die Rotation des eckigen Felgenmitnehmers detektiert. Alle 90 Grad wird so ein Messimpuls generiert. Damit an der Hinterachse ähnlich zuverlässig gemessen werden kann, muss der gleiche Felgenmitnehmer verwendet werden. Dieser ist aber breiter und

benötigt deswegen eine längere Achse.



*Abbildung 1.7: Der ursprüngliche Felgenmitnehmer an der Hinterachse*



*Abbildung 1.8: Der breitere Felgenmitnehmer an der Hinterachse*

Die Sensorhalterung konnte ebenfalls verbessert werden. In der Prototypenversion werden der Sensor, die elektrische Bauteile und die Halterung, die mit dem Wagen verbunden ist, mit Klebstoff gesichert. Ein nachträgliches tauschen der Teile wird dadurch schwierig und das Erkennen von funktionellen Beeinträchtigungen nahezu unmöglich. Deshalb besteht die neue Halterung aus zwei Teilen, die mit Schrauben verbunden werden. Um es so einfach wie möglich zu machen, werden zwei Aluminium L-Träger (15 mm und 10 mm) verwendet. Der eine ist fest mit dem Wagen verbaut, während der andere mit dem Sensor verklebt wird.



*Abbildung 1.10: Die alte Sensorhalterung der Hinterachse*



*Abbildung 1.9: Die neue Sensorhalterung*

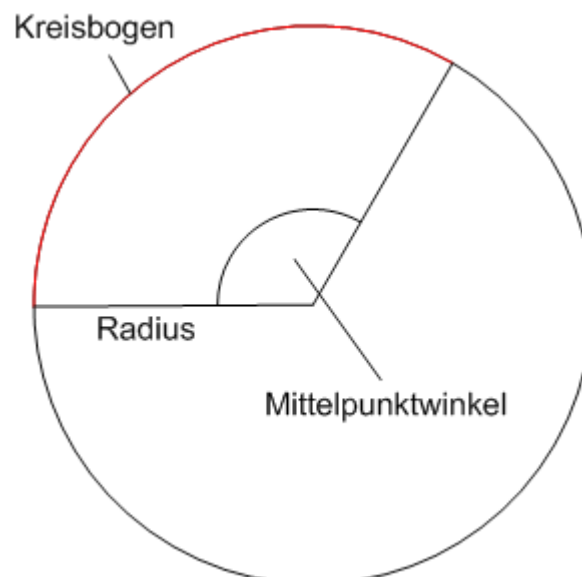
Da die Achsschenkel bereits mit Bohrungen versehen sind, eignen sich diese als Plattform für die Halterungen. Mit selbstschneidenden Schrauben wird dann der 15mm L-Träger mit dem Radlager dauerhaft verbunden. Der Abstand zwischen Sensor und Felgenmitnehmer kann nachträglich angepasst werden. Der Vollständigkeit halber

sollte noch die Karosseriehalterung erwähnt werden. Die Bolzen vorne wurden verlängert, die Karosserie kann so etwas höher gesetzt werden. Das ist eine Voraussetzung um den Rechner auf der Plattform unterbringen zu können. Auch die Verkabelung und Spannungsüberwachung wurde teilweise überarbeitet. In dem Prototypen wurden diese Schaltungen noch auf einer Versuchsplatine frei verdrahtet. Daraus wurde ein Layout erstellt um eine industrielle Platine fertigen zu können. Bei der Gelegenheit wurde die Funktionalität erweitert, damit der Hauptrechner einen Akkualarm für jede Spannungsquelle empfangen kann. Dazu wird das Signal der Anzeigediode abgegriffen und über einem Treiberbaustein bereitgestellt. Idealerweise sollte der Alarm „low-active“ sein, damit bei einer fehlenden Spannungsversorgung der Rechner nicht annimmt, es sei alles in Ordnung. Auch die Bauteile für die Radsensoren bekommen eine eigene Platine, ebenso der Spannungswandler für die Wandlung der PWM-Signale. Diese Platinen wurden nach den verwendeten Schaltplänen des Prototypen erstellt.

## 2 Systemanalyse

### 2.1 Die Kurve allgemein

Kurven können vielfältige Formen annehmen und selbstverständlich können diese auch eine Wegstrecke für ein Fahrzeug beschreiben. Bei der Definition der Kurve für die Lenkregelung sollen sämtliche Möglichkeiten abgedeckt werden. Häufig macht es aber Sinn Problemstellungen zu generalisieren und dann komplexere Szenarien aus kleinen Teillösungen zu bilden. Diese Vorgehensweise wird auch zur Modellbildung der Kurve für die Lenkregelung genutzt. Zur Beschreibung wird der Kreisbogen benutzt, der durch den Radius und Mittelpunktswinkel definiert wird.



*Abbildung 2.1: Definition der Kurve für die Lenkregelung*

Der Mittelpunktswinkel definiert die resultierende Winkeländerung nach der Kurve. Wenn es erforderlich ist komplexere Kurven, die in ihren Verlauf enger oder weiter werden, zu fahren, müssen diese in mehrere Kurven unterschiedlicher Radien unterteilt werden.



## 2.2 Das Fahrzeug allgemein

Bei dem nicht modifizierten Fahrzeug handelt es sich um einen 750mm x 490mm großen, ferngesteuerten Geländewagen. Der Wagen verfügt über einen Verbrennungsmotor mit 2,6 PS der seine Kraft, ebenso wie die Bremswirkung, nur auf die Hinterachse überträgt. Obwohl der Wagen eher den Geländefahrzeugen zuzuordnen ist, lässt sich das Differenzialgetriebe nicht sperren. Durch das permanente Differenzialgetriebe können wir also annehmen, dass die Hinterachse nicht mit zwei gleich schnell drehenden Rädern der Kurvenfahrt entgegenwirkt. Auch gibt es keine Gangschaltung und damit ebenso wenig die Möglichkeit rückwärts zu fahren. Der fehlende Rückwärtsgang macht daher weitere Überlegungen zum Rangieren überflüssig.

### 2.2.1 Die Lenkung

Die folgende Grafik stellt den Zusammenhang zwischen dem Lenkwinkel der Räder und dem Radius der Kurve dar und dient als Modell für die radiusbasierte Kurvenfahrt. Damit wird die Fähigkeit bezeichnet das Fahrzeug auf verschiedenen Kreisbahnen fahren zu lassen.

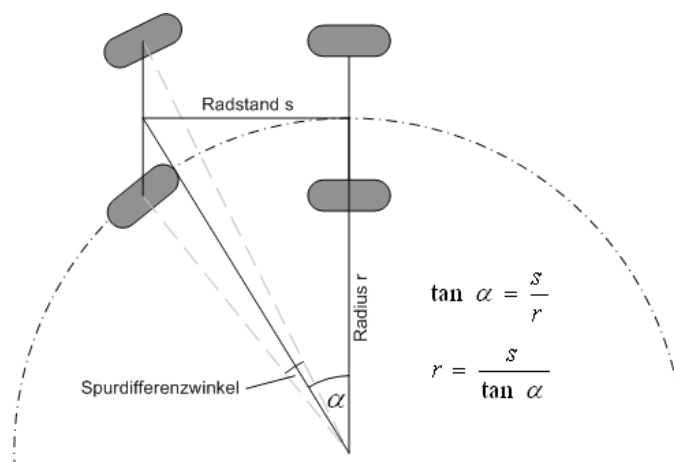


Abbildung 2.2: Das Modell für die radiusbasierte Kurvenfahrt

Um eine Kurve optimal zu durchfahren, muss die so genannte Ackermann-Bedingung erfüllt sein. Die Ackermann-Bedingung besagt, dass die Verlängerungen

aller Achsen bei einer Kurvenfahrt sich im Kreismittelpunkt treffen müssen. Die Hinterräder erfüllen die Bedingung stets, sofern sie starr sind. Die Vorderräder müssen abhängig von dem Radius ihrer Kreisbahn eingeschlagen werden. Die Lenkmechanik schlägt deswegen die Vorderräder nicht im gleichen Winkel ein, das innere Rad wird stärker eingeschlagen als das Äußere. Die Differenz dieser beiden Winkel nennt man „Spurdifferenzwinkel“ oder „Voreilwinkel“. Realisiert wird ein solches mechanisches Verhalten mit dem so genannten Lenktrapez. Das Lenktrapez kann nur den idealen Spurdifferenzwinkel annähern. So gibt es bei herkömmlichen Lenkungen nur zwei Einschlagwinkel, bei denen das Lenktrapez exakt ist, beziehungsweise die Ackermann-Bedingung erfüllt ist.

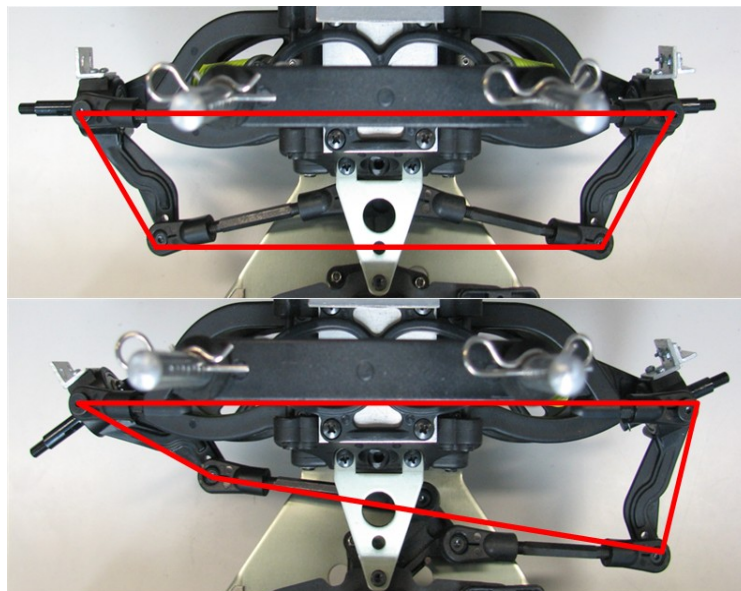


Abbildung 2.3: Das Lenktrapez

Die Formel zeigt also nur einen vereinfachten Zusammenhang. Wie man erkennen kann, teilt die Hypotenuse nicht genau den Spurdifferenzwinkel, somit wird die resultierende Lenkwirkung nicht korrekt beschrieben. Das stellt im ersten Schritt kein Problem dar, denn die Mechanik gewährleistet konstruktionsbedingt in mehrfacher Hinsicht keine ideale Kurvenfahrt. Zum einen wegen den Eigenschaften des Lenktrapezes, zum anderen besitzt die Lenkung des Modellfahrzeugs etwas Spiel. Zudem kommt noch, dass die Reifen bei einer Kurvenfahrt unterschiedlich belastet werden und so auf Grund ihrer weichen Beschaffenheit unterschiedliche Rollwiderstände und Führungskräfte entwickeln. Betrachtet man dazu noch unterschiedliche Bodenbeschaffenheiten, ist eine exakte Modellbildung für die

vielfältigen Umweltbedingungen sehr komplex. Deshalb bleiben diese Punkte in der Modellbildung unberücksichtigt. Für die Vorgabe einen bestimmten Radius zu fahren wird für die Regelung die einfache Formel zur Berechnung genutzt.

### **2.2.2 Steuermöglichkeiten**

Gesteuert wird das Fahrzeug mit üblichen Modellbauservos. Diese Servomotoren werden mit PWM-Signalen gesteuert und stellen einen Hebel auf einen bestimmten Winkel ein. Wirkt eine Kraft gegen das Servo, versucht es seinen Winkel zu halten und erzeugt eine Gegenkraft und der Strombedarf erhöht sich entsprechend. Insgesamt werden zwei Servos auf dem Fahrzeug verwendet. Zum einen für die Lenkung, zum anderen für die Motor- und Bremssteuerung. Das für die Lenkung relevante Servo hat eine Stellgeschwindigkeit von  $60^\circ / 0,15$  Sekunden ( $1^\circ / 0,0025$  Sekunden). Daraus ergibt sich eine Zeit von 112,5 ms für einen maximalen Lenkausschlag ( $45^\circ$ ) aus der Geradeausfahrt heraus. Bei höheren Geschwindigkeiten und engen Radien ähnelt so die gefahrene Kurve immer weniger einer Kreisbahn. Ein Aspekt, der bei der Bahnplanung zu berücksichtigen ist.

### **2.2.3 Signalverarbeitung**

Die Servomotoren werden über PWM-Signale angesteuert. Dabei wird ein periodischer Rechteck-Puls in der Länge verarbeitet, der einem TTL-Logikpegel entspricht. Die Generation dieser Signale übernimmt ein Microcontroller, der mit dem CAN-Bus an den Hauptrechner (ein PC) angeschlossen ist. Über eine CAN-Nachricht kann so die Pulsbreite und damit die Servostellung verändert werden. Um über Systemgrenzen hinweg elektrische Signale austauschen zu können, ist ein gemeinsamer Massepunkt nötig. Die Stromverteilerplatine des Fahrzeugs spielt in diesem Punkt die maßgebliche Rolle, da alle Spannungsquellen dort zusammen laufen, eignet sie sich dazu alle negativen Potentiale miteinander zu verbinden.

Es werden aber nicht nur Daten an die Microcontroller gesendet. Auch sie senden ihrerseits Nachrichten an den PC, die eine Entscheidungsgrundlage für die Hauptsteuerung darstellen. Aufgrund der Möglichkeit verschiedene Digital- und Analog-Signale zu verarbeiten werden an die Microcontroller sämtliche Sensoren,

einschließlich das Gyroskop, angeschlossen. Die so gewonnenen Informationen können über den CAN-Bus an den Hauptrechner geschickt werden.

In der momentanen Version des Fahrzeugs gibt es dafür keine spezielle Bustopologie. Da aber mit einer steigenden Anzahl von Teilnehmern und steigenden Datenraten Probleme entstehen, wenn die Buszugriffe nicht organisiert werden, geht die Betrachtung des Bussystems nicht über einen Hauptrechner und einen Microcontroller für das Gyroskop hinaus.

## **2.3 Das Gyroskop**

### **2.3.1 Funktionsweise**

Das Gyroskop von Analog Devices basiert auf einer schwingenden Mikromasse, die aufgrund ihrer Bewegungen unterschiedliche Radien auf der zu messenden Rotationsachse durchläuft [1]. Da auf unterschiedlichen Radien bei gleicher Winkelgeschwindigkeit unterschiedliche Strecken pro Zeiteinheit zurückgelegt werden, wird die Mikromasse durch ihre Schwingungen kontinuierlich in Drehrichtung beschleunigt und gebremst. Die Mikromasse ist in einem mechanisch gedämpften Rahmen eingelassen, sodass durch das Trägheitsmoment der Masse Lageänderungen gemäß der Winkelgeschwindigkeit des Gyroskops gemessen werden können. Die Messung der Lageänderung erfolgt durch elektrische Felder, genauer gesagt nach dem Prinzip eines Kondensators. Um die Genauigkeit zu verbessern, werden im Grunde zwei Gyroskope gleichzeitig betrieben.

### **2.3.2 Verarbeitung der Daten**

Als Ausgangsgröße liefert das Gyroskop ein Analogsignal, das die Winkeländerungen pro Zeiteinheit repräsentiert [2]. Dabei bedeutet eine Spannung von 2,5V, dass keine Winkeländerung stattfindet. Spannungen unter 2,5V stellen eine entsprechend schnelle Linksdrehung dar, Spannungen darüber eine Rechtsdrehung. Die Winkelgeschwindigkeit wird also in Volt / Grad / Sekunde dargestellt. Dabei bedeutet eine Änderung um 0,005V (unter Idealbedingungen) des Ausgangssignals

eine Änderung der Winkelgeschwindigkeit um 1 Grad / Sekunde. Dieses Analogsignal wird von dem Microcontroller mit einem A/D-Wandler digitalisiert. Der A/D-Wandler des verwendeten Atmel-Microcontroller besitzt eine Auflösung von 10 Bit, die Abtastung dauert 60 - 240µs [3].

Bei der A/D-Wandlung gelten die bekannten Gesetzmäßigkeiten und Probleme. Man muss also darauf achten, dass die Abtastrate mindestens dem Doppelten der maximal zu erwarteten Richtungsänderungen (das schließt Vibrationen mit ein) entspricht, da es sonst zum Aliaseffekt [4] kommt. Bei dem Aliaseffekt wird durch das Raster der Abtastung eine hohe Frequenz auf eine niedrigere Frequenz abgebildet:

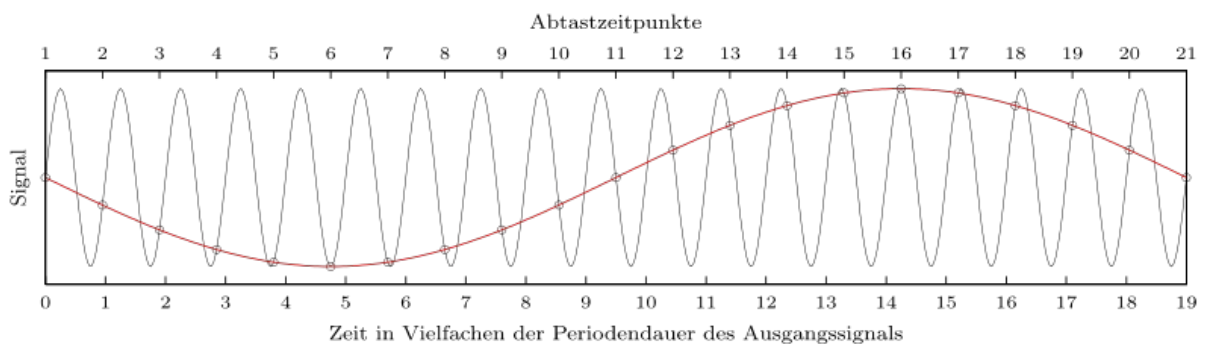


Abbildung 2.4: Der Aliaseffekt (Quelle: Wikipedia)

Durch die Digitalisierung können also nur Frequenzen der halben Abtastfrequenz (auch Nyquist-Frequenz genannt) korrekt aufgezeichnet werden. Die Bandbreite des Gyroskops liegt bei 40 Hz. Wir haben somit die Anforderung mindestens mit 80 Hz das Analogsignal abzutasten. Allerdings haben wir mit dieser Anforderung lediglich eine korrekte Abbildung der Frequenz erreicht. Von einer korrekten Darstellung der Form des Signals ist man mit dieser Bedingung noch weit entfernt, denn mit steigender Frequenz nimmt auch die „Treppenstufenbildung“ des Signals zu bis hin zu einem Rechteckimpuls bei der höchsten darstellbaren Frequenz. Das stört allerdings nicht die Berechnungen. Aus der Winkelgeschwindigkeit werden wir für die Regelung den Winkel berechnen, der das Integral aus der Winkelgeschwindigkeit ist. Durch die Digitalisierung erhalten wir stets eine Treppenstufenform des Signals, die sich besonders einfach mit der Rechteckregel integrieren lässt.

$$\int f(t) dt = \sum \text{Messwert} \cdot \text{Abtastintervall}$$

Alternativ könnte man die Trapezregel zum Integrieren benutzen. Ihre Lösung konvergiert quadratisch, während die Lösung der Rechteckregel sich linear annähert. Somit ist die Trapezregel genauer, aber sie basiert auf der Annahme, dass der Anstieg zwischen zwei Messwerten linear erfolgt. Es liegt zwar nahe, dass es so ist, aber streng genommen, kann darüber keine Aussage getroffen werden, denn zwischen den Abtastungen ist es einfach nicht definiert wie das Signal verläuft.

Die folgende Grafik zeigt die beiden Verfahren im Vergleich. Die rote Linie repräsentiert den Spannungsverlauf, die blaue Linie zeigt den angenäherten Verlauf durch die Trapezregel. Die Rechtecke stellen zum einen die Darstellung des Signals in digitalisierter Form dar, zum anderen repräsentieren sie grafisch die Anwendung der Rechteckregel.

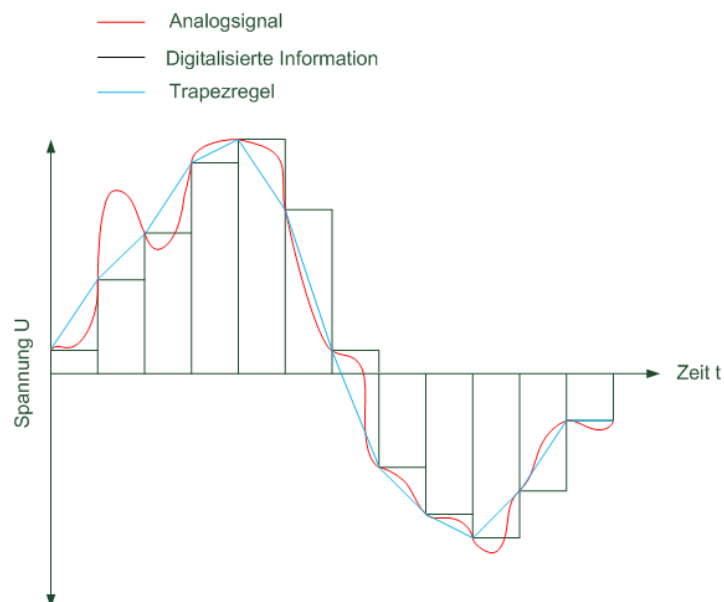


Abbildung 2.5: Illustration der Integralrechnung

Wichtig für die Integration, ist eine exakte Bestimmung der Nulllinie. Verschiebt sich die Nullstellung aus irgend einem Grund, wird eine Änderung des Winkels registriert und eine Lagebestimmung ist nicht mehr möglich. Ebenso wichtig ist die zeitliche Zuordnung der Messwerte. Aus der Analyse des Fahrzeugs ergab sich, dass die Messwerte über einen CAN-Bus versendet werden. Die Software des CAN-Busses ist dazu in der Lage Nachrichten zu puffern. Ebenso kann das Betriebssystem des Rechners (Windows XP) einen Prozess suspendieren oder die Ressourcen anders

verteilen, was ein vorhersagbares Verhalten nicht ermöglicht. Es muss also damit gerechnet werden, dass die Nachrichten, auch wenn die Messwerte immer im gleichen Intervall aufgenommen werden, nicht in den gleichen Zeitabständen eintreffen. Das Heranziehen der Nachrichtenintervalle zur Berechnung kann somit zu falschen Ergebnissen führen.

### **2.3.3 Genauigkeit und Arbeitsbedingungen**

Die Empfindlichkeit des Gyroskops liegt bei  $300^\circ / \text{Sekunde}$  [2]. Für die Messung der meisten Kurvenfahrten genügt diese Auflösung sicherlich. Vermutlich können aber Vibrationen durch den Motor Ungenauigkeiten hervorrufen, ebenso wie kurze, harte Stöße im unebenen Gelände. Die Resonanzfrequenz des Gyroskops liegt bei 14 kHz. Dabei ist zu bedenken das zum Beispiel das harmonische Vielfache und Halbierende einer Frequenz sich ebenfalls auswirken können. Da der Motor einen weiten Drehzahlbereich besitzt, kann es durchaus sein, dass je nach Drehzahl eine zur Resonanzfrequenz harmonische Schwingung auftritt und das Gyroskop stört. Damit das jedoch zu einem Problem werden kann, müssen die Schwingungen in Richtung der Bewegungen der Mikromasse wirken und die Schwingung muss innerhalb der Bandbreite (40 Hz) des Gyroskops liegen. Darüber gibt es noch keine konkreten Untersuchungen. Jedoch zeigte sich in vorangegangenen Arbeiten, dass der Rauschpegel des Gyroskops bei laufendem Motor erhöht ist [5].

Daher muss in den folgenden Praxistests verstärkt auf mechanische Einwirkungen geprüft werden, denn selbst von Hand konnten in den ersten Versuchen die  $300^\circ / \text{Sekunde}$  erreicht werden und anschließend eine deutliche Verschiebung des Winkels gemessen werden.

Ebenso spielt die Umgebungstemperatur eine Rolle. Wie bereits erwähnt, handelt es sich bei dem Ausgangssignal um ein Analogsignal, das wie alle anderen elektrischen Signale von temperaturabhängigen Eigenschaften der elektrotechnischen Bauteile beeinflusst wird. Aus diesem Grund liefert das Gyroskop zusätzlich eine Temperaturinformation. Damit können die thermischen Störfaktoren kompensiert werden. Außerdem dient die Temperaturanzeige zur Warnung, falls der spezifizierte Temperaturbereich verlassen wird, keine verlässlichen Daten zur Verfügung stehen und gegebenenfalls mit einer dauerhaften Beschädigung gerechnet werden muss.

Wenn der spezifizierte Temperaturbereich verlassen wurde, muss nach dem Zurückkehren in den zulässigen Bereich mit einer Ungenauigkeit von  $0,1^\circ / \text{Sekunde}$  gerechnet werden. Die Funktionalität der Temperaturkompensation wurde bislang in den bestehenden Projekten nicht berücksichtigt, vermutlich, weil bei Laborbedingungen die Schwankungen der Raumtemperatur recht gering sind. Auch hier wird es sich erst im Praxistest zeigen, ob im Außeneinsatz nicht doch eine Temperaturkompensation nötig ist. Zumindest zeigte sich bei den ersten Begutachtungen, dass der Nullpunkt des Gyroskops nicht immer den gleichen Wert hatte, was vermutlich auf den Temperatureinfluss zurückzuführen ist.

Nach dem Einschalten benötigt das Gyroskop 35 ms bis halbwegs verlässliche Daten zur Verfügung stehen. In dieser Zeit schwankt die Genauigkeit um  $1/2^\circ$  pro Sekunde. Anschließend gibt es eine drei minütige Phase, in der sich die Empfindlichkeit noch um  $0,03^\circ / \text{Sekunde}$  ändern kann. Danach ist die Empfindlichkeit nicht mehr vom Einschaltmoment abhängig, sondern von der Versorgungsspannung mit  $0,6^\circ / \text{Sekunde} / \text{V}$  oder  $0,7\% / \text{V}$  (beides bei 4,75 V bis 5,25 V), der Beschleunigung mit  $0,2^\circ / \text{Sekunde} / \text{g}$  und der Temperatur.

Die Empfindlichkeit kann noch angepasst werden. Es wird dabei empfohlen nicht mehr als das Vierfache der Empfindlichkeit einzustellen. Die Temperaturstabilität leidet allerdings darunter. Die Bandbreite wäre ebenfalls wählbar, jedoch ist sie in unserem Fall auf 40 Hz festgelegt, weil das Evaluationsboard von Analog Devices benutzt wird [6]. Ansonsten kann diese durch einen Kondensator angepasst werden.



## 3 Die Regelung

### 3.1 Das Grundprinzip der Regelung

Unter einer Regelung [7] versteht man die Überwachung einer Regelgröße (Istwert) mit einer Führungsgröße (Sollwert). Die Stärke der Abweichung beider Werte (Regeldifferenz) dient als Entscheidungsgrundlage für die Intensität des Gegensteuerns (Stellgröße), mit der der Istwert in den Bereich des Sollwertes geregelt werden soll. Störungen können die Regelaufgabe erschweren, sie fließen in das System indirekt über den Istwert mit ein. Das Gesamtsystem wird Regelkreis genannt und besteht aus dem eigentlichen Regler und dem Restsystem das auch Regelstrecke genannt wird.

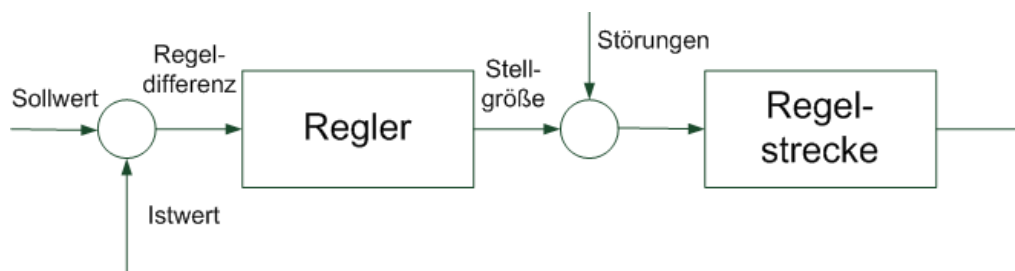


Abbildung 3.1: Der Regelkreis

Um einen Regelkreis zu realisieren benötigt man einen Sensor für die zu überwachende Messgröße, einen Aktor um diese Größe zu verändern zu können und den Regler der auf Grund der Sensordaten eine neue Steuergröße vorgibt. Das Gyroskop dient zusammen mit dem Mikrocontroller als Sensor, der Lenkservo, ebenfalls gesteuert durch den Microcontroller ist der Aktor und der PC der Regler.

Weil die Regeldifferenz nur aus Werten der gleichen physikalische Einheit gebildet werden kann, sind Umformungen der Sensordaten und der Stellgröße nötig, denn beide arbeiten mit unterschiedlichen Maßeinheiten. Im Fall der Lenkregelung muss aus einer Spannung in Volt von dem Gyroskop ein PWM-Signal für das Lenkservo in Timerwerten für den Microcontroller gewandelt werden. Auf die einzelnen Schritte der Wandlung wird im Kapitel der Implementierung genauer eingegangen. Ein Regler kann aus mehreren sog. Übertragungsgliedern bestehen, die jeweils ein bestimmtes Verhalten aufweisen.

### 3.1.1 Das Proportionalglied

Das einfachste Verhalten eines Reglers wird mit dem Proportionalglied beschrieben. Dabei wird die Regeldifferenz mit einem Faktor, dem Proportionalitätsbeiwert, skaliert. Die Stellgröße ändert sich so entsprechend der Regeldifferenz.

$$a = K_p \cdot e \quad \text{mit} \quad \left\{ \begin{array}{ll} a & : \quad \text{Ausgangsgröße} \\ e & : \quad \text{Eingangsgröße} \\ K_p & : \quad \text{Proportionalitätsbeiwert} \end{array} \right\}$$

Das Proportionalglied hat den Vorteil, dass es schnell reagiert, nachteilig ist die bleibende Regeldifferenz. Die kann allerdings verkleinert werden, indem der Proportionalitätsbeiwert vergrößert wird. Wenn aber ein zu großer Proportionalitätsbeiwert eingestellt wird, kann die Regelung anfangen zu schwingen.

### 3.1.2 Das Integralglied

Das Integralglied summiert die Regeldifferenzen über die Zeit, das heißt je nach dem wie stark und wie lange eine Regeldifferenz andauert, fällt die Reglerantwort mehr oder weniger stark aus. Ist die Eingangsgröße konstant und ungleich Null, wächst die Ausgangsgröße linear an.

$$a = \frac{e}{T_I} \cdot \Delta t + a_0 \quad \text{mit} \quad \left\{ \begin{array}{ll} a & : \quad \text{Ausgangsgröße} \\ e & : \quad \text{Eingangsgröße} \\ a_0 & : \quad \text{Summe der Regeldifferenzen} \\ T_I & : \quad \text{Integrationszeit} \end{array} \right\}$$

Die Integrationszeit  $T_I$  bestimmt die Stärke der Reglerantwort, ähnlich wie der Proportionalitätsbeiwert. Dabei ist  $T_I$  die Zeitspanne, die benötigt wird bis die Ausgangsgröße um den Wert der Eingangsgröße gewachsen ist. Alternativ kann die Wirkung des Integralgliedes mit dem so genannten Integrierbeiwert  $K_I$  eingestellt werden.

$$K_I = \frac{1}{T_I} \quad a = K_I \cdot e \cdot \Delta t + a_0 \quad \text{mit} \quad \left. \begin{array}{l} a : \quad \text{Ausgangsgröße} \\ e : \quad \text{Eingangsgröße} \\ a_0 : \quad \text{Summe der Regeldifferenzen} \\ K_I : \quad \text{Integrierbeiwert} \end{array} \right\}$$

Das Integralglied hat den Vorteil, dass es Regeldifferenzen vollständig beseitigen kann, jedoch arbeitet es sehr träge, da eine lang anhaltende Regeldifferenz entsprechend lange abgebaut werden muss.

### 3.1.3 Das Differenzialglied

Ein Übertragungsglied mit differentiellen Verhalten generiert eine Reglerantwort, die von der Änderungsgeschwindigkeit der Regeldifferenz abhängig ist. Eine konstante Regeldifferenz erzeugt keine Änderung des Ausgangs, egal wie stark die Regeldifferenz ist. Hingegen erzeugt ein sprunghafter Anstieg der Regelgröße eine sehr heftige Antwort des Übertragungsgliedes.

$$a = K_D \cdot \frac{\Delta e}{\Delta t} \quad \text{mit} \quad \left. \begin{array}{l} a : \quad \text{Ausgangsgröße} \\ e : \quad \text{Eingangsgröße} \\ \Delta e : \quad \text{Anstieg der Regeldifferenz} \\ K_D : \quad \text{Differentialbeiwert} \end{array} \right\}$$

Differenzialglieder reagieren daher sehr schnell auf Änderungen, jedoch kann eine konstante Regeldifferenz nicht behoben werden. Um den Einfluss eines Differenzialgliedes steuern zu können wird es mit dem so genannten Differentialbeiwert multipliziert.

### 3.1.4 Der PID-Regler

Welche der drei Übertragungsfunktionen für eine Regelaufgabe am geeignetsten, lässt sich anhand der genannten Vor- und Nachteile schlecht beurteilen. Jede Funktion hat seine wichtige positive Eigenschaft und leider auch ein Defizit, dass einen Einsatz einer Übertragungsfunktion allein wahrscheinlich unmöglich macht. Als logische Konsequenz bietet sich der PID-Regler an, der die drei

Übertragungsfunktionen parallel ausführt und deren Ausgangsgrößen zu einer summiert.

$$a = K_P \cdot e + K_I \cdot e \cdot \Delta t + a_0 + K_D \cdot \frac{\Delta e}{\Delta t}$$

mit

$$\left. \begin{array}{l} a : \text{Ausgangsgröße} \\ e : \text{Eingangsgröße} \\ \Delta e : \text{Anstieg der Eingangsgröße} \\ a_0 : \text{Integral} \\ K_P : \text{Proportionalitätsbeiwert} \\ K_I : \text{Integrierbeiwert} \\ K_D : \text{Differentialbeiwert} \end{array} \right\}$$

Mit den jeweiligen Beiwerten lässt sich der Einfluss jeder einzelnen Übertragungsfunktion einstellen und so auf die speziellen Regelanforderungen anpassen. Auch andere Kombinationen wie PI- oder PD-Regler sind möglich.

### 3.2 Gewünschtes Verhalten

Wenn ein Mensch eine Kurve in seinem Auto fährt, weiß er instinktiv, wie das Lenkrad eingeschlagen werden muss, damit die Kurve möglichst flüssig durchfahren wird. Das Lenken kann in drei Phasen unterteilt werden.

*Phase 1:* Die Lenkung wird bis zu dem gewünschten Winkel einschlagen

*Phase 2:* Der Lenkwinkel wird gehalten

*Phase 3:* Die Lenkung wird zurück gestellt

Ähnlich sollte die Regelung für das Modellfahrzeug agieren können. Dabei ist schon die *Phase 1* nicht ganz unproblematisch. Ein sanftes hineinfahren in die Kurve hat einen Komfort Hintergrund, der erstmal vernachlässigt wird. Bei höheren Geschwindigkeiten muss aber darauf geachtet werden, dass die Lenkung nicht zu stark eingeschlagen wird, damit das Fahrzeug nicht ins Schleudern gerät. Dieser Punkt wird im Absatz über geschwindigkeitsabhängiges Verhalten noch näher erörtert.

Die *Phase 2* ist naturgemäß einfach. Die Herausforderung ist das Herausfinden des Zeitpunktes zwischen der *Phase 2* und der *Phase 3*, sowie die Geschwindigkeit, mit der die Lenkung in *Phase 3* zurückgestellt wird. Denn diese Punkte sind für eine saubere Kurvenfahrt maßgeblich und dazu von der Geschwindigkeit des Fahrzeugs abhängig. Es sollte nicht so sein, dass die Räder erst nach einer Regelabweichung von "0" ruckartig zurückgestellt werden. Die Regelung sollte zusätzlich Geschwindigkeitsänderungen in der Kurve berücksichtigen, beziehungsweise am besten nicht davon beeinflusst werden.

### 3.3 Reglerauslegung

Die erste wichtige Festlegung für die Auslegung des Reglers ist die Führungsgröße. Aus der allgemeinen Betrachtung der Regelungstechnik ist bekannt, dass eine Regeldifferenz nur aus gleichen physikalischen Größen gebildet werden kann. Mittels des Gyroskops können entweder die Winkelgeschwindigkeit oder der Richtungswinkel des Fahrzeugs als Ist-Wert geliefert werden. Damit sind die möglichen physikalischen Größen bereits festgelegt.

Die Winkelgeschwindigkeit eignet sich weniger für die Lenkregelung, wenn es darum geht verständliche Sollwertvorgaben machen zu können, denn diese muss dann ebenfalls als Winkelgeschwindigkeit angegeben werden. Auch die anschließende Umformung der Ausgangsgröße in einen Stellwert ist nicht besonders intuitiv.

Betrachtet man die Definition der Kurve für die Lenkregelung, bietet sich die Richtung des Fahrzeugs als Führungsgröße an. Das Gyroskop liefert den Winkel in Grad und die Hauptsteuerung eine gewünschte Fahrtrichtung ebenfalls in Grad. Die Differenz aus beiden stellt die Regelabweichung dar. Die Stellgröße für die Lenkmechanik wird ebenfalls in Grad beschrieben, wie aus dem Modell für die radiusbasierte Kurvenfahrt hervor geht.

Werden diese Aspekte in einem Systemmodell vereinigt, lässt sich daraus der Regler in seiner Grundauslegung ableiten, obwohl die Geschwindigkeit, von der ja der zeitliche Verlauf abhängt, nicht unmittelbar einfließt. Wenn der Lenkeinschlag der Räder in ein proportionales Verhältnis von ungefähr 1:1 zur Regelabweichung gesetzt wird (P-Verhalten), ist die Lenkregelung bereits realisiert. Die Regelabweichung verändert sich proportional zur Geschwindigkeit des Fahrzeugs

wenn die Räder eingeschlagen sind und mit diesem Prinzip auch der Lenkeinschlag (Regelabweichung = Lenkeinschlag/Stellgröße).

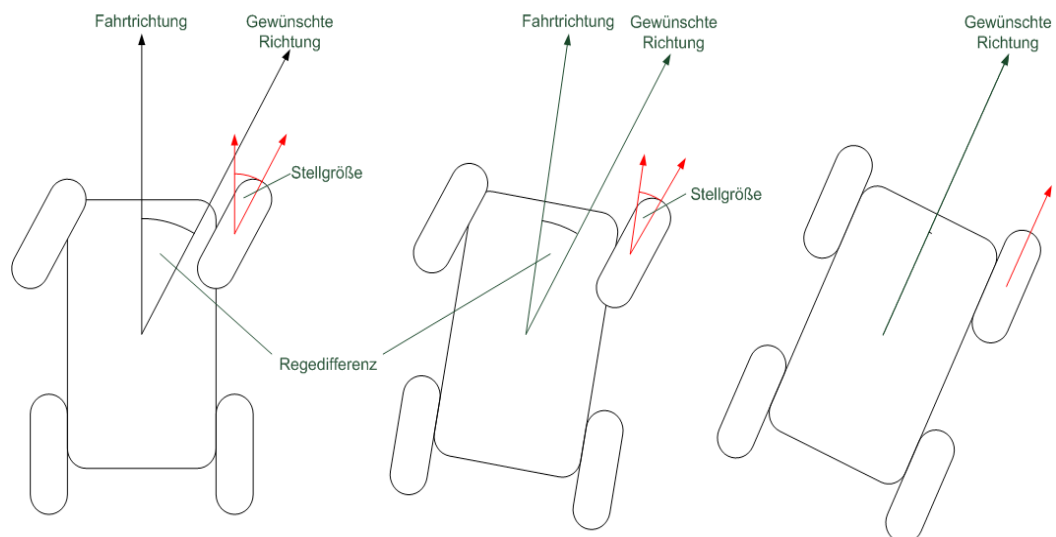


Abbildung 3.2: Zusammenhang zwischen Regelabweichung und Stellgröße

Natürlich kann die Mechanik nicht jeden Stellwert der sich aus der Regelabweichung ergibt, umsetzen, aber das Zurückstellen der Lenkung erfolgt ähnlich dem eines menschlichen Autofahrers.

### 3.3.1 Festlegung des P-Anteils

Der P-Anteil sollte nach dem erörterten Prinzip die Regelabweichung möglichst unverändert lassen, damit sie als Stellgröße verarbeitet werden kann. Der ideale Multiplikator für den P-Anteil wird deshalb wahrscheinlich zwischen 0,7 und 1,3 liegen. Den am besten geeignete Wert muss im Betrieb des Fahrzeugs ermittelt werden.

$$K_p = 0,7..1,3$$

### 3.3.2 Festlegung des I-Anteils

Wie bei der Vorstellung der einzelnen Übertragungsglieder erwähnt, besitzt das I-Glied die Möglichkeit Regelabweichungen vollständig zu beheben. Auch wenn ein

solches Verhalten wünschenswert ist, muss darauf verzichtet werden. Der I-Anteil eines Reglers ist einfach zu langsam und entspricht auch nicht den Anforderungen einer Kurvenfahrt. Insbesondere die unterschiedlichen Geschwindigkeiten des Fahrzeugs lassen den I-Regler ungeeignet erscheinen. Betrachtet man den Extremfall, bei dem das Auto in einer Kurve darauf wartet, bis ein Hindernis aus dem Weg gegangen ist, steigt die Regeldifferenz kontinuierlich an, da der I-Regler die Regeldifferenz über die Zeit integriert. Unsere Regelabweichung bleibt aber einfach bestehen, wenn das Fahrzeug sich nicht bewegt. Somit bildet ein I-Anteil unsere Regelanforderung falsch ab.

$$K_I = 0$$

### 3.3.3 Festlegung des D-Anteils

Durch die Regelung würde nicht nur die Kurvenfahrt realisiert werden, sondern auch die Geradeausfahrt. Sollte durch äußere Einflüsse das Fahrzeug seine Richtung ändern, könnte die Regelung auch das erkennen und korrigieren. Betrachtet man den Fall, in dem das Fahrzeug durch eine starke Unebenheit auf nur einer Seite des Weges und bei hoher Geschwindigkeit, die Richtung schlagartig ändert, würde eine Korrektur durch eine sanft gefahrene Kurve eine Verschiebung auf der Wegstrecke zu Folge haben.

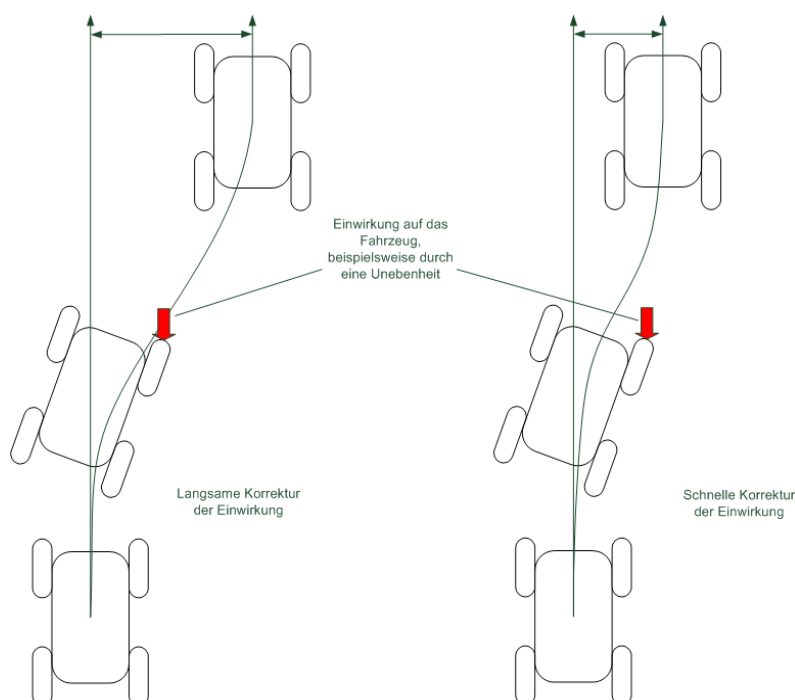


Abbildung 3.3: Korrektur der Ausrichtung nach einer umweltbedingten Einwirkung

Um das zu vermindern, müsste schnell und ebenso stark gegengesteuert werden. Mit dem Differentialanteil kann eine entsprechende Reglerantwort generiert werden, die sich nach der Geschwindigkeit der Richtungsänderung und damit nach der Stärke der Einwirkung richtet. Dieser Effekt würde sich jedoch nie ganz durch die Lenkregelung beheben lassen, aber durch einen differenziellen Anteil in der Regelung könnte man ihm entgegenwirken. Ein geeigneter Wert muss im direkten Test mit dem Fahrzeug ermittelt werden und bleibt deshalb erstmal offen.

$$K_D = x \{ \text{muss experimentel ermittelt werden} \}$$

Der D-Anteil ist für die grundsätzliche Funktion der Lenkregelung nicht wichtig, er stellt lediglich eine Erweiterung dar. Deswegen ist möglich diesen mit "0" einzustellen.

## **3.4 Begrenzung der Ausgangsgröße und Sollwertvorgaben**

### **3.4.1 Begrenzung der Ausgangsgröße**

Die Mechanik der Lenkung kann keine beliebigen Stellgrößen umsetzen. Es ist sinnvoll diese softwareseitig auf das mechanisch Mögliche zu begrenzen, denn das Servo kann, wenn es den Anschlag erreicht, zerstört werden. Das geschieht nicht im Regler, sondern sinngemäß in der Regelstrecke, die ihrerseits auch in Software abgebildet werden kann. Ferner wird, wenn ein bestimmter Radius eingehalten werden soll, ebenfalls eine Begrenzung des Lenkausschlags notwendig. Die exakte Begrenzung hängt also von dem konkreten Anwendungsfall ab und sollte variabel sein.

### **3.4.2 Begrenzung der Sollwertvorgaben**

Um die gewünschte Richtung zu beschreiben, ist es zweckmäßig, eine Einteilung von  $-180^\circ$  bis  $+180^\circ$  zu benutzen, denn so kann eine entsprechende Links- oder Rechtskurve leicht nachvollziehbar initiiert werden. Bei Regeldifferenzen über  $\pm 180^\circ$



gibt es Mehrdeutigkeiten. Beispielsweise kann bei einer Regeldifferenz von  $200^\circ$ , eine Kurve mit nur  $-160^\circ$  gefahren werden und das Fahrzeug fährt trotzdem in die gleiche Richtung. Auch wenn man den Eindruck gewinnen könnte, dass es sich hier um eine Optimierung handelt, darf die Regelung nicht selbst entscheiden und die Kurven auf diese Weise abkürzen, schließlich ist der Unterschied zwischen Links- und Rechtsabbiegen zu massiv.

Wenn die Regeldifferenz ein Vielfaches von  $\pm 360^\circ$  beträgt, lässt sich an dieser Stelle etwas verbessern. Im Grunde bedeutet das Vielfache, wie oft der Wagen im Kreis fahren würde. In einem Fehlerfall, wenn eine zufällige Zahl als Sollwertvorgabe übergeben wird, kann das zum Problem werden. Vielfache von  $360^\circ$  werden deshalb beim setzen der Sollwertvorgabe nicht beachtet. Wenn die Hauptsteuerung mehrere  $360^\circ$ -Kurven nach einander fahren möchte, müssen diese sequentiell ausgeführt werden.

### **3.5 Geschwindigkeitsabhängiges Verhalten**

Weil das Fahrzeug Geschwindigkeiten von bis zu 65 Km/h erreichen kann, sollte die Lenkung mit einer Dämpfung versehen werden, die den Radius entsprechend der Geschwindigkeit vergrößert. Im einfachsten Fall kann dazu der P-Anteil verringert werden oder es kann ein weiterer Radius vorgeben werden.

Denkbar ist es auch, der Lenkregelung eine Information über die aktuelle Geschwindigkeit bereitzustellen und im Gegenzug den maximalen Radius für eine Kurve zu erhalten. Die Lenkregelung müsste also über Wissen verfügen, mit dem sie das Verhalten des Fahrzeugs vorhersagen kann. Betrachtet man die Modellbildung der Lenkung aus dem vorangegangenen Kapitel, stellt man fest, dass die Geschwindigkeit alleine dazu nicht ausreicht. Viele andere Parameter wie Gewicht des Fahrzeugs, Beschaffenheit des Untergrunds und eine exakte Vermessung der Mechanik sind nur einige Punkte. Einfacher aber nicht weniger nützlich ist eine Überwachung der Bodenhaftung der Vorderräder. Erhält die Lenkregelung eine Geschwindigkeitsinformation, kann in einer Kurve überprüft werden, ob die Vorderräder haften, indem nach einer bestimmten Zeit nach dem Lenkausschlag die Regelabweichung geprüft wird. Ändert sich die Regelabweichung für die angegebene Geschwindigkeit nicht oder zu langsam, übersteuert die Lenkung und das Fahrzeug

sollte sofort abgebremst werden.

### 3.5.1 Berechnung des Soll-Winkel

Durch den Radius kennt man den Umfang des gewünschten Kreises. Errechnet man aus der gefahrenen Strecke und den Umfang das Verhältnis, kann man auf den Soll-Winkel schließen, wenn man die  $360^\circ$ , die ein Kreis hat, damit multipliziert. Mit dieser simplen Berechnung ist es möglich, nicht nur einmalig nach einem festen Zeitpunkt die Kurve zu kontrollieren, sondern kontinuierlich die Lenkwirkung zu überwachen. Damit wäre die Hauptsteuerung in der Lage Grenzbereiche zu erkennen und sich darin zu bewegen.

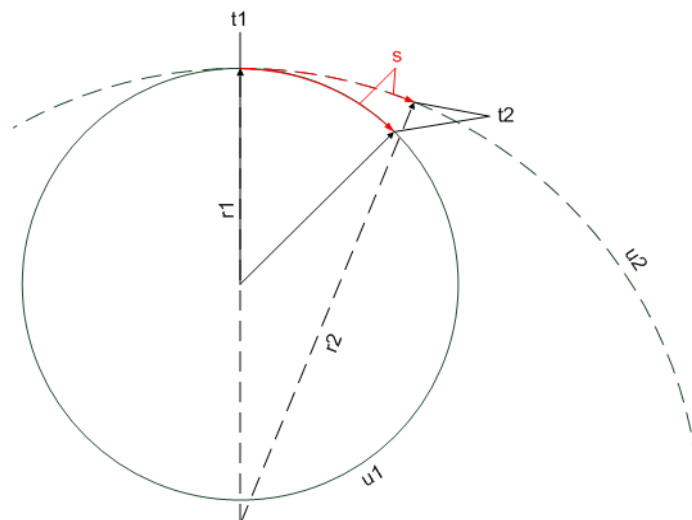


Abbildung 3.4: Darstellung zur Ermittlung des Soll-Winkel und Ist-Radius

$t_1$  und  $t_2$  stellen in dieser Grafik die beiden Zeitpunkte dar, nach denen die Lenkwirkung überprüft wird. Der durchgängig gezeichnete Kreis mit dem Radius  $r_1$  ist die gewünschte Kurve, der gestrichelte Kreis die tatsächliche Kurve mit dem Radius  $r_2$ . Die rot eingezeichneten Sektoren kennzeichnen die Länge der gefahrenen Strecke innerhalb der Zeit  $t_2 - t_1$  und sind gleich lang. Um den Winkel zu ermitteln, den das Fahrzeug nach der Zeitspanne  $t_2 - t_1$  erreicht haben müsste, genügen Standardformeln.

$$\text{Strecke } s = v \cdot (t_2 - t_1)$$

$$\text{Umfang } u_1 = 2 \cdot \text{Soll-Radius} \cdot \pi$$

$$\text{Anteil der gefahrenen Strecke auf der Kreisbahn } a = \frac{s}{u_1}$$

$$\text{Soll-Winkel}_{(t_2-t_1)} = a \cdot 360^\circ$$

$$\text{Soll-Winkel}_{(t_2-t_1)} = \frac{(v \cdot (t_2 - t_1))}{(2 \cdot \text{Soll-Radius} \cdot \pi)} \cdot 360^\circ$$

### 3.5.2 Berechnung des Ist-Radius

Es ist sinnvoll die Berechnung auch um zu kehren und aus dem erreichten Ist-Winkel nach einer bestimmten Zeit, den tatsächlichen Radius der Hauptsteuerung zu liefern, damit diese Information direkt in die Bahnplanung einfließen kann. Die Differenz aus den beiden Radien, repräsentiert ebenfalls die Wirksamkeit des Lenkeinschlags. Ist die Differenz nahe bei „0m“ besitzt das Fahrzeug optimale Traktion und der gewünschte Radius wird eingehalten. Der erreichte  $\text{Winkel}_{(t_2-t_1)}$  wird direkt von dem Gyroskop abgelesen.

$$\text{Strecke } s = v \cdot (t_2 - t_1)$$

$$\text{Ist-Radius} = \frac{u_2}{2 \cdot \pi}$$

$$\text{Multiplikator für die gefahrene Strecke } a = \frac{360^\circ}{\text{Winkel}_{(t_2-t_1)}}$$

$$u_2 = a \cdot s$$

$$\text{Ist-Radius} = \frac{\left( \left( \frac{360^\circ}{\text{Winkel}_{(t_2-t_1)}} \right) \cdot (v \cdot (t_2 - t_1)) \right)}{(2 \cdot \pi)}$$

## 4 Implementierung

### 4.1 Der allgemeine Regler

Im ersten Schritt wurde ein allgemeiner Softwareregler entworfen, der die vorangegangenen Anforderungen aus der Regelungstechnik erfüllt. Es macht Sinn die Übertragungsglieder in einzelne Methoden zu kapseln, um die Übersicht zu gewährleisten und leicht individuelle Algorithmen zusammenstellen zu können. Der Regler kann während das Programm läuft, konfiguriert werden und ermöglicht ein ergonomisches entwickeln. Auch eine spätere adaptive Regelung kann damit auf das Regelverhalten Einfluss nehmen.

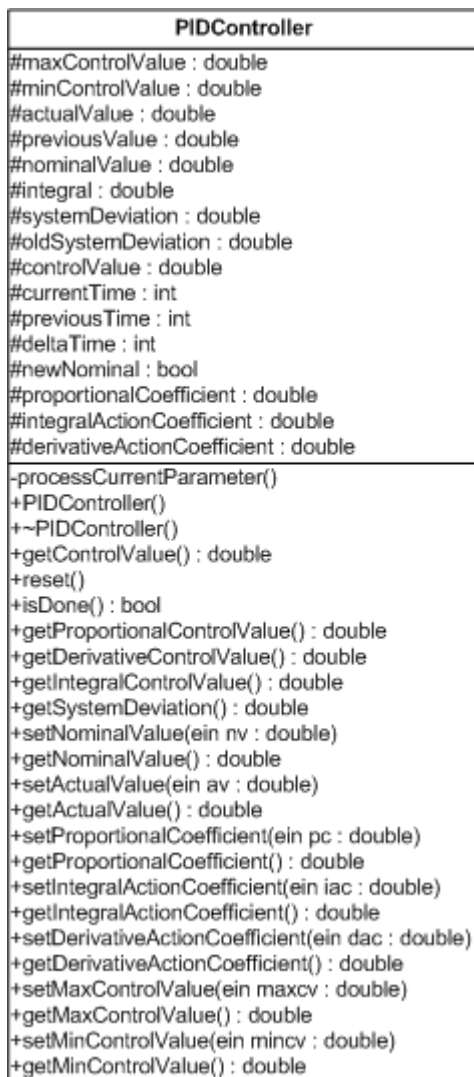


Abbildung 4.1: Klassendiagramm des allgemeinen Reglers

Die Berechnung eines neuen Stellwertes wird mit jedem neuen Istwert ausgeführt, damit wird die Berechnung der Regelabweichung initiiert und die Übertragungsglieder liefern ihre Ausgangsgrößen. Diese werden zu einer Ausgangsgröße summiert und die aufrufende Komponente erhält mit einer *Get*-Methode die Reglerantwort.

Die Hauptsteuerung [8], die mit dem PID-Regler, eine Kurvenfahrt durchführen soll, arbeitet verhaltensbasiert. Das bedeutet, dass die Lenkregelung nur für die Dauer der Kurvenfahrt die Kontrolle über die Lenkung hat. Danach wird die Kontrolle der Lenkregelung entzogen und ein anderes Verhalten, wie zum Beispiel „Zielmarkenfahrt“, tritt in den Vordergrund. Das erfordert einen Mechanismus, der den Anfang und das Ende einer Kurvenfahrt erkennen und der Hauptsteuerung melden kann. Der Anfang einer Kurvenfahrt wird mit dem Einstellen einer

neuen Sollwertvorgabe eingeleitet. Mit jedem weiteren Wert des Gyroskops wird die bestehende Regelabweichung berechnet. Vollzieht die Regelabweichung einen Vorzeichenwechsel, also ist die Regelabweichung durch den Nullpunkt gewandert, ist formal die Kurve beendet. Das Klassendiagramm zeigt den Entwurf des allgemeinen Softwarereglers. Um die Anforderungen, wie die Begrenzung des Stellwertes für die Lenkung oder die Rückmeldung über das Ende der Regelaufgabe, zu realisieren, sind weitere Parameter und Methoden nötig, die nicht im Zusammenhang der Regelungstechnik vorgestellt wurden.

#### 4.1.1 Die Methode *PIDController::processCurrentParameter()*

Die Berechnung des neuen Stellwertes erfolgt in der Methode *processCurrentParameter()*. Von dort aus werden die Methoden der Übertragungsglieder aufgerufen und deren Rückgabewerte summiert. Ebenso erfolgt dort die Beschränkung des Stellwertes auf die mechanischen Grenzen und die Berechnung zur Feststellung des Durchgangs der Regelabweichung durch den Nullpunkt. Der Nulldurchgang sollte nicht mit einem direkten Vergleich mit "0" bestimmt werden. Es ist sehr unwahrscheinlich, dass der Gleitkommawert der Regelabweichung damit exakt getroffen wird. Besser ist es einen Vorzeichenwechsel zu registrieren.

Dazu wird die neue Regelabweichung nach dem setzen einer neuen Sollwertvorgabe gespeichert und anschliessend mit der aktuellen Regelabweichung multipliziert. Betrachtet man eine negative Regelabweichung, die gespeichert wurde, sind alle darauf folgenden ebenfalls negativ, wenn auch größer. Die Multiplikation liefert immer ein positives Ergebnis. Erst ab dem Nullpunkt wird die Regelabweichung positiv und das Ergebnis wird negativ. Ebenso verhält es sich bei positiven Regelabweichungen, die immer kleiner werden und ab dem unterschreiten des Nullpunkts ein negatives Ergebnis liefern.

$$\begin{aligned} \text{Regelabweichung}_{\text{start}} \cdot \text{Regelabweichung}_{\text{aktuell}} \leq 0 &\Rightarrow \text{Nulldurchgang} \\ \text{Regelabweichung}_{\text{start}} \cdot \text{Regelabweichung}_{\text{aktuell}} > 0 &\Rightarrow \text{kein Nulldurchgang} \end{aligned}$$

Die Begrenzung der Stellgröße wurde so implementiert, dass auch eine

asymmetrische Lenkmechanik im Regler konfiguriert werden kann. Es ist zwar unwahrscheinlich und auch nicht zweckmäßig mit einer asymmetrischen Lenkung zu fahren, jedoch soll die Implementierung des PID-Reglers so allgemein wie möglich sein, damit der Code im Projekt wiederverwendet werden kann. Deshalb kann die Grenze für den positiven und negativen Ausschlag einzeln festgelegt werden.

Um über die Zeit integrieren zu können, benötigt man das Zeit-Intervall, zwischen den Messwerten. Diese Berechnung erfolgt vor dem Aufruf der Übertragungsglieder, zusammen mit der Berechnung der Regelabweichung, mit dem Systemaufruf *timeGetTime()*. Der Zeitwert wird als Integer in Millisekunden ab Betriebssystemstart geliefert und muss noch auf Sekunden skaliert werden.

```
void PIDController::processCurrentParameter(void)
{
    systemDeviation = nominalValue - actualValue;
    currentTime = timeGetTime() / 1000.0;

    deltaTime = currentTime - previousTime;
    previousTime = currentTime;
    controlValue =
    (
        getProportionalControlValue() +
        getDerivativeControlValue() +
        getIntegralControlValue()
    );

    if(controlValue > maxControlValue)
        controlValue = maxControlValue;

    if(controlValue < minControlValue)
        controlValue = minControlValue;

    if(systemDeviation * oldSystemDeviation < 0 && newNominal == true)
    {
        newNominal = false;
    }
}
```

Die Berechnung der Übertragungsglieder ist mit ihren mathematischen Formeln aus dem vorangegangenen Kapitel beschrieben. Dennoch müssen ein paar Feinheiten beachtet werden.

#### 4.1.2 Die Berechnung des Proportionalanteils

```
double PIDController::getProportionalControlValue(void) {
    return proportionalCoefficient * systemDeviation;
}
```

#### 4.1.3 Die Berechnung des Differentialanteils

```
double PIDController::getDerivativeControlValue(void) {
    double diff = 0.0;
    if(deltaTime != 0)
        diff = derivativeActionCoefficient * (actualValue - previousValue) / deltaTime;
    previousValue = actualValue;
    return diff;
}
```

#### 4.1.4 Die Berechnung des Integralanteils

```
double PIDController::getIntegralControlValue(void) {
    integral += integralActionCoefficient * systemDeviation * deltaTime;
    return integral;
}
```

#### 4.1.5 Problematik mit dem Zeitintervall

Dadurch, dass die Aufrufe der *Set*-Methode von den eingehenden Nachrichten abhängt und diese wiederum gepuffert werden können, gibt es Nachrichtenintervalle, die kleiner als eine Millisekunde sind (also 0). Wird die Differenz aus dem vorangegangenen und aktuellen Wert durch die Zeit 0 dividiert, erhält man einen klassischen Programmabbruch. Aus dem Quellcodeausschnitt erkennt man, dass die Berechnung des Differentialanteils deshalb mit einer Abfrage versehen ist. Das bedeutet leider zwangsläufig, dass einige Wertepaare nicht berechnet werden können. Auch der I-Anteil wird dadurch verfälscht. Zwar kann es nicht zu einem Programmabbruch kommen, aber das betreffende Intervall wird mit „0“ multipliziert, bevor es auf die Fläche addiert wird. Um diese Problematik zu lösen wird an dieser Stelle das Attribut *fixedDelay* eingeführt. Damit wird die Zeitberechnung im Prinzip

überflüssig. Der Hintergrund für diese Lösung liegt in den festen Zeiten, in denen der Microcontroller das Analogsignal digitalisiert. Dieses Intervall ist eigentlich maßgeblich für die Berechnung und sollte sich in den Nachrichten-Intervallen widerspiegeln. Da es nicht der Fall ist, wird ein Wert fest eingestellt.

## 4.2 Konvertierung der Daten

Durch die unterschiedlichen Datenformate, von einer Analogspannung bis hin zum PWM-Signal des Servos, entsteht eine Verarbeitungskette, die über den Regler hinausgeht.

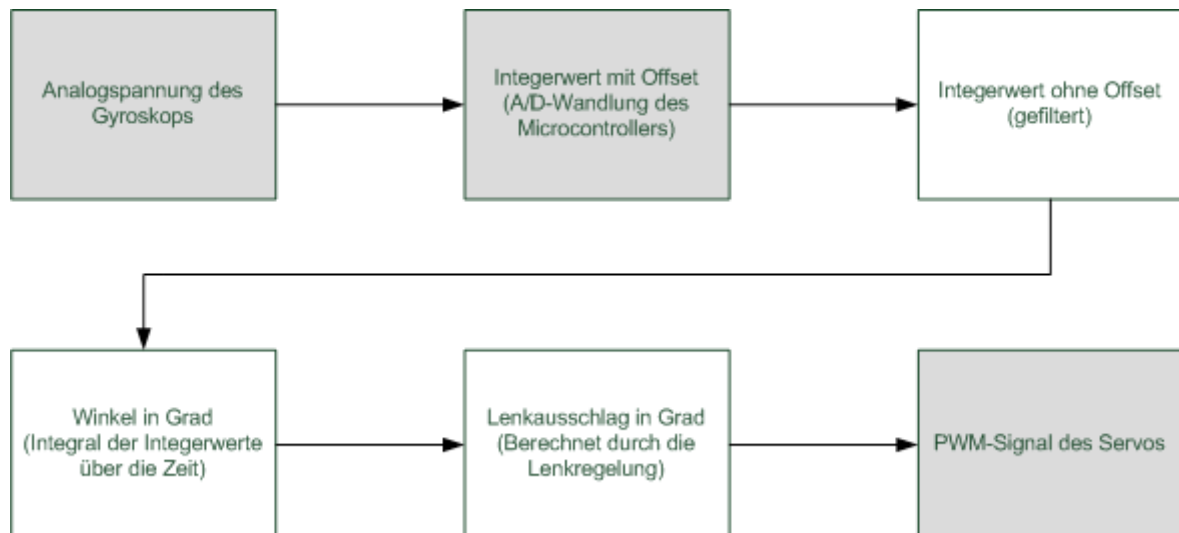


Abbildung 4.2: Die notwendigen Verarbeitungsschritte der Regelung

Jeder Pfeil in dem Bild repräsentiert eine Konvertierung der Daten. Die grau unterlegten Kästen sind Stellen im System, die außerhalb der Hauptsteuerung liegen. Die Konvertierung von dem Winkel zum Lenkausschlag wird durch die Lenkregelung bereits realisiert. Für die anderen Konvertierungen sind eigene Implementierungen nötig.



## DataProcessor

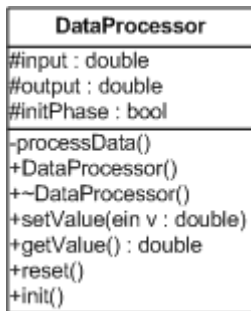


Abbildung 4.3:  
Klassendiagramm  
DataProcessor

Um einen wiederverwendbaren Code zu erhalten, wird für die Konvertierung die Klasse *DataProcessor* eingeführt. Sie stellt das Grundgerüst für die Datenkonvertierungen dar. Für die Initialisierungen gibt es eine Algorithmik, um einen eventuellen Start-Code ausführen zu können. Solange die Initialisierungsmethode nicht ausgeführt ist, werden keine Daten weitergereicht. Von dieser Klasse werden alle weiteren Software-Prozessoren abgeleitet.

## DataProcessorAnalogToRaw

Als erster Schritt ist es nötig, aus den Daten der A/D-Wandlung den Nullpunkt zu ermitteln. Das Gyroskop liefert stets positive Spannungswerte. Um die Winkeländerungen über die Zeit zu integrieren, muss der Offset von den Werten abgezogen werden. Deshalb sollte eine Initialisierung durchgeführt und dabei der Mittelwert des Analogsignals ermittelt werden. Während der Initialisierungsphase darf das Gyroskop nicht bewegt werden. Implementiert ist die Mittelwertbildung in der Initialisierungsmethode. Da in dieser Klasse die Rohdaten eintreffen wird hier auch das Logfile geschrieben, mit dem Testläufe später wieder ausgewertet werden können. Ferner besitzt die Klasse einen gleitenden Mittelwertfilter [9], mit dem das Rauschen geglättet werden kann, indem über eine bestimmte Anzahl der letzten Messwerte, der Mittelwert gebildet wird.

## DataProcessorRawToAngle

Liegen die Daten in einem integrierbaren Format vor, kann der Winkel berechnet werden. Die Formel für die Berechnung des Integrals wurde bereits unter dem Punkt "*Verarbeitung der Daten*" im Kapitel 2 vorgestellt. Innerhalb dieser Klasse werden auch weitere Berechnungen durchgeführt, wie zum Beispiel die Ermittlung der

Winkelgeschwindigkeit. Diese Berechnungen sind nicht wichtig für die Regelaufgabe, sondern dienen zur Beurteilung der Messwerte.

### *DataProcessorControlToPWM*

Nachdem für eine Regelabweichung eine Stellgröße ermittelt wurde, muss ein äquivalentes PWM-Signal generiert werden. Der Microcontroller, der die PWM-Signale erzeugt, erwartet Integerwerte als Eingabe. Der Bereich, liegt für das Lenkservo je nach Einstellung der Mechanik zwischen 1700 und 3200 für die jeweiligen Vollausschläge. Der Stellwert muss auf diese Werte skaliert werden. Die Klasse ist für diese Aufgabe gedacht und kann auch auf asymmetrische Skalen abbilden. Für jede Lenkmechanik muss empirisch die Werte für die Mittelstellung und den linken und rechten Vollausschlag ermittelt werden. Zusammen mit den maximalen Stellwerten in Grad, mit denen auch der Regler konfiguriert wurde, wird eine Skala gebildet.

$$PWM = \begin{cases} \text{Stellgröße} \geq 0: & PWM_{\text{Mitte}} + \text{Stellgröße} \cdot ((PMM_{(\text{Max})} - PWM_{(\text{Mitte})}) / \text{Stellgröße}_{(\text{Max})}) \\ \text{Stellgröße} < 0: & PWM_{\text{Mitte}} - \text{Stellgröße} \cdot ((PMM_{(\text{Mitte})} - PWM_{(\text{Min})}) / \text{Stellgröße}_{(\text{Min})}) \end{cases}$$

## 4.3 Der konkrete Regler

Ergänzend zu dem allgemeinen PID-Regler sind für die Lenkregelung noch weitere Komponenten notwendig. Um einen konkreten Regler zu implementieren und die Ergänzungen einfließen zu lassen, wird die Klasse *PIDControl* abgeleitet zu *PIDSteering*. *PIDSteering* benutzt die erforderlichen *DataProcessor*-Derivate und überschreibt die Funktionen *getDervativeControlValue()*, *getIntegralControlValue()* und *processCurrentParameter()*. Beide *Get*-Methoden in der Basisklasse greifen auf die Zeitberechnung zurück, die auf den Messungen der Nachrichtenpausen beruht und daher unzuverlässig ist. Statt Zeitmessungen werden, wie bereits vorgestellt, fest eingestellte Werte verwendet. *processCurrentParameter()* wurde um die Aufrufe

für die *DataProcessor*-Klassen erweitert.

### 4.3.1 Implementierung der radiusbasierten Kurvenfahrt

Um das Modell aus der Systemanalyse für die Radiusfahrt umzusetzen, sind ebenfalls weitere Methoden und Attribute nötig. Das Einstellen eines Radius bedeutet prinzipiell lediglich die Begrenzung des Lenkwinkels. Eine Begrenzung, die höchstens auf den maximalen Lenkwinkel einstellt, ist implementiert. Allerdings ist sie noch nicht variabel, wie es für beliebige Radien erforderlich ist. Deshalb werden weitere Begrenzungen benötigt, die nach jedem Einstellen eines neuen Radius neu berechnet werden.

$$\tan \alpha = \frac{s}{r}$$

$$\alpha = \arctan\left(\frac{s}{r}\right)$$

$$\text{mit } \left\{ \begin{array}{l} s : \text{Radstand} \\ r : \text{Radius} \\ \alpha : \text{Lenkeinschlag} \end{array} \right\}$$

Aus diesem Zusammenhang ergibt sich folgende Implementierung.

```
void PIDSteering::setRadius(double r)
{
    radius = r;
    maxControlRadius = floor(((180 * atan(radStand/radius)) / M_PI) * 100) / 100;
    minControlRadius = -maxControlRadius;

    if(maxControlRadius > maxControlValue)
        maxControlRadius = maxControlValue;

    if(minControlRadius < minControlValue)
        minControlRadius = minControlValue;
}
```

Der Arcustangens liefert ein Ergebnis im Bogenmaß, deshalb muss in das Gradmaß gewandelt werden. Die Grenzen *maxControlRadius* und *minControlRadius* werden als Grenzen für die Reglerantwort benutzt. Sie werden ihrerseits auch auf die

mechanischen Grenzen (*maxControlValue*, *minControlValue*) beschränkt. Mit  $\text{floor}(x * 100) / 100$  wird lediglich das Ergebnis auf zwei Nachkommastellen abgeschnitten.

### 4.3.2 Messung der Lenkwirkung

Die Überprüfung der Lenkwirkung wird gemäß der vorgestellten Methodik innerhalb eines Zeitintervalls ausgeführt. Die Hauptsteuerung kann mit den Methoden *startCheck(v)* und *endCheck(v)* eine Kontrollmessung veranlassen und muss beiden Methoden die aktuelle Geschwindigkeit in Km/h übergeben. Die Geschwindigkeit wird dann innerhalb der Zeitspanne gemittelt. Mit *startCheck(v)* wird am Anfang der Zeitpunkt, Winkel und Geschwindigkeit festgehalten. Mit *endCheck(v)* werden am Ende die gleichen Parameter zum Vergleich nochmal gespeichert. Wenn die Vergleichswerte vorliegen, können die Abweichung des Radius und des Winkels errechnet werden.

$$Ist - Radius = \left( \frac{\left( \frac{360^\circ}{(Ist - Winkel_{(t2-t1)})} \right) \cdot (v \cdot (t2 - t1))}{(2 \cdot \pi)} \right)$$

$$Soll - Winkel_{(t2-t1)} = \frac{(v \cdot (t2 - t1))}{(2 Soll - Radius \cdot \pi)} \cdot 360^\circ$$

```

void PIDSteering::startCheck(double v)
{
    v1 = v;
    a1 = actualValue;
    t1 = timeGetTime() / 1000;
}

void PIDSteering::endCheck(double v)
{
    v2 = v;
    a2 = actualValue;
    t2 = timeGetTime() / 1000;

    dt = t2 - t1; // Zeit in Sec
    va = (v1 + v2) / 2; // Durchschnittsgeschwindigkeit in Km / h
    da = a2 - a1; // Erreichter Winkel in Grad

    s = va * (3600 / dt); // Distanz

```

```

    checkRadius = (s * (360 / da)) / (2 * M_PI) - radius;
    checkAngle = s / (2 * radius * M_PI) * 360 - actualValue;
}

```

Um stets korrekte und vor allem aktuelle Daten zu erhalten, muss das aufrufende Programm die Aufrufreihenfolge einhalten und unmittelbar die Daten weiterverarbeiten. Ansonsten können alleine auf Grund einer zu langen Pause, zwischen Messung und Verarbeitung Fehler entstehen.

#### 4.3.3 Die Methode *PIDSteering::processCurrentParameter()*

Zum Vergleich zur allgemeinen Lösung werden folgende überschriebene Methoden (Änderungen rot eingezeichnet) für die Lenkregelung verwendet.

```

void PIDSteering::processCurrentParameter(void)
{
    dPAnalogToRaw.setValue(actualValue);
    dPRawToAngle.setValue(dPAnalogToRaw.getValue());
    actualValue = dPRawToAngle.getValue();

    systemDeviation = nominalValue - actualValue;

    controlValue =
    (
        getProportionalControlValue() +
        getDerivativeControlValue() +
        getIntegralControlValue()
    );

    if(controlValue > maxControlRadius)
        controlValue = maxControlRadius;

    if(controlValue < minControlRadius)
        controlValue = minControlRadius;

    dPControlToPWM.setValue(controlValue);

    if(systemDeviation * oldSystemDeviation < 0 && newNominal == 1)
    {
        newNominal = false;
    }
}

double PIDSteering::getDerivativeControlValue(void)
{
    double diff = 0.0;

    diff = derivativeActionCoefficient * (actualValue - previousValue) / fixedDelay;
}

```

```

        previousValue = actualValue;

        return -diff;
    }

    double PIDSteering::getIntegralControlValue(void)
    {
        integral += integralActionCoefficient * systemDeviation * fixedDelay;
        return integral;
    }

```

#### 4.3.4 Klassendiagramm für *PIDSteering*

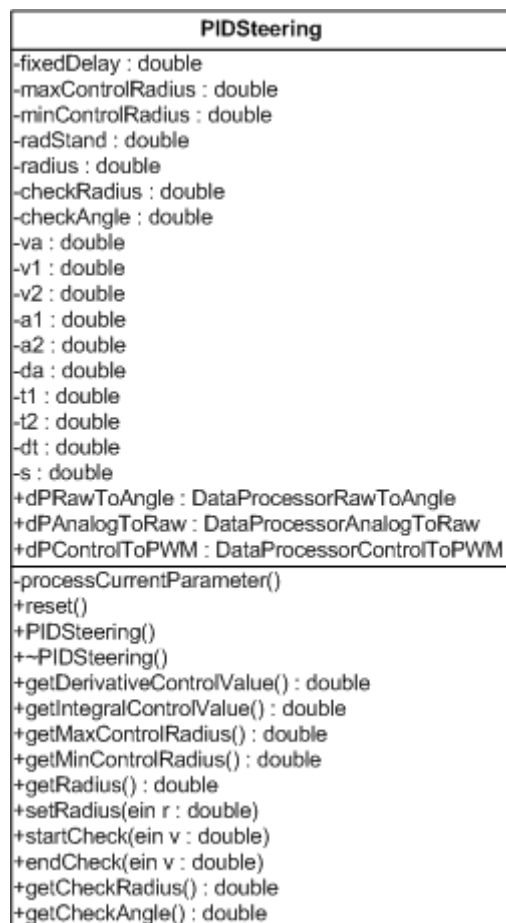


Abbildung 4.4: Klassendiagramm der konkreten Regelung

## 4.4 Das Programm GyroControl

Im Laufe der Einarbeitung in das Projekt zeigte sich, dass ein Werkzeug zur Darstellung der Funktion des Gyroskops und des Reglers sehr nützlich sein kann. Es ist schwierig anhand von Zahlenkolonnen das korrekte Verhalten des Reglers zu bestimmen. Deshalb wurde eine Visualisierungssoftware entwickelt, mit der die Daten des Microcontrollers, die Ergebnisse der einzelnen Verarbeitungsschritte und die Reglerantwort in Echtzeit parallel angezeigt werden kann. Ferner kann der Regler konfiguriert, Sollwertvorgaben erstellt und das System sowie einzelne Komponenten daraus zurückgesetzt werden. Auch die Möglichkeit Logfiles einzulesen und das Verhalten des Gyroskops und der Software nach einem Testlauf zu reproduzieren ist möglich.

Die Diagramme sind frei konfigurierbar und können bei entsprechender Einstellung das Signal auch vergrößern. Die Darstellung erfolgt ereignisbasiert, das heißt, wenn ein neuer Messwert eintrifft wird der Cursor um eine Position weiter gestellt. Es handelt sich somit bei der X-Achse nicht direkt um eine Zeitachse, sondern sie stellt nur die Reihenfolge der Messwerte dar. Durch den kontinuierlichen Datenstrom des Microcontrollers entsteht aber eine korrekte, zeitliche Darstellung. Bei einer Abtastrate von 50Hz entsprechen 50 Werte einer Sekunde.

Die parallele Visualisierung der Berechnungsschritte macht das Verhalten des Systems besonders anschaulich. Deswegen werden in den Testläufen die Szenarien mit Screenshots festgehalten und erörtert.

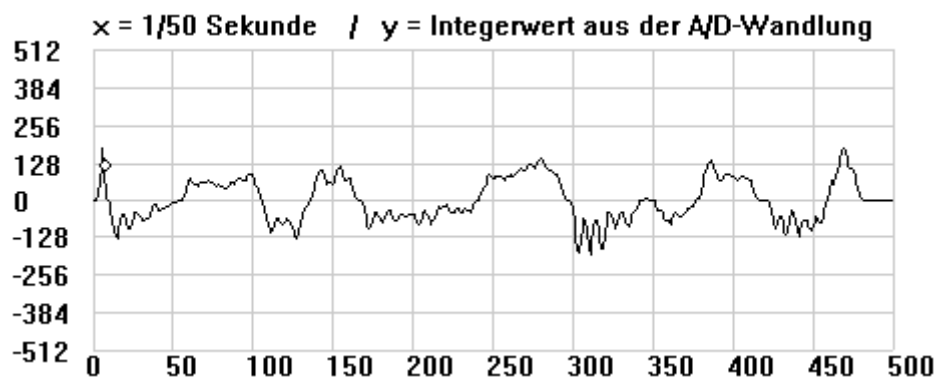


Abbildung 4.5: Beispielscreenshot aus dem Programm GyroControl

Wenn das Ende des Messwertspeichers eines Diagramms erreicht ist, fängt das

Beschreiben am Anfang wieder an. Dadurch erhält das Programm den Charakter eines Oszilloskops. Im Laufe der Aufgabenbearbeitung hat sich die Gegenüberstellung der Rohdaten, des errechneten Winkels und der Reglerantwort als sinnvoll herausgestellt. Zusätzlich gibt es weitere Parameter, die als Zahlenwert ausgegeben werden. Auch das erwies sich als sehr nützlich, denn so wurde für bestimmte Testläufe die einfach Oberfläche angepasst. In den Screenshots werden diese Werte jedoch nicht gezeigt um die Übersicht zu gewährleisten.

<b>Liste der dargestellten Parameter</b>
<b>A/D-Wandlung</b>
Offsetfreier Zahlenwert aus der A/D-Wandlung
Mittelwert des Rauschen (Offset)
Momentaner Wert aus der A/D-Wandlung
<b>Winkelberechnung</b>
Momentaner Winkel
Pause zwischen den Nachrichten in ms
Winkelgeschwindigkeit innerhalb einer Periode
Winkelgeschwindigkeit innerhalb einer Sekunde (Mittelwert der letzten 50 Werte)
Maximale Winkelgeschwindigkeit/sec.
Minimale Winkelgeschwindigkeit/sec.
Durchschnittliche Winkelgeschwindigkeit
<b>PID-Regler</b>
Sollwert
Istwert
Stellvorgabe
Regelabweichung
Sollwertvorgabe erfüllt
Maximaler Stellwert für einen bestimmten Radius
Minimaler Stellwert für einen bestimmten Radius

*Tabelle 1: Darstellbare Parameter in GyroControl*

Die Anzeige der grafischen Daten übernimmt die Klasse *Diagramm*. Sie verfügt über Konvertierungsroutinen, die Bildpunkte von einem mathematischen Koordinatensystem in Bildkoordinaten umwandeln und über einen Messwertspeicher einstellbarer Größe. Für die Umwandlung der Koordinaten benötigt man die Position des Diagramms auf dem Bildschirm, sowie die Breite und Höhe in Bildpunkten. Das Zeichnen der Gitternetzlinien, der bisherigen Messwerte und des neuen Messwertes mit einem Cursor ist in einzelnen Methoden gekapselt. Die Unterteilung der Gitternetzlinien ist im Programm fest eingestellt.



## 4.5 Integration des Gyroskops auf dem Microcontroller

Das Gyroskop kann entweder über Kabel oder über eine weitere Platine mit entsprechenden Steckkontakten mit dem Microcontroller verbunden werden. Die Kabelvariante wurde aufgrund der möglichen elektromagnetischen Störungen verworfen, hätte aber den Vorteil gehabt, dass das Gyroskop auch an unzugänglichen Stellen verbaut werden kann. Die Platinenvariante hat den Vorteil der kurzen Signalwege und einer einmaligen Lösung des vibrationsfesten Integrieren der Elektronik. Damit eine feste Verbindung mit dem Microcontroller hergestellt werden kann, musste ein unkonventionelles Layout entwickelt werden.

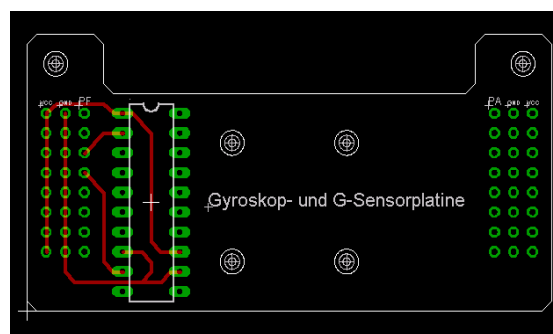
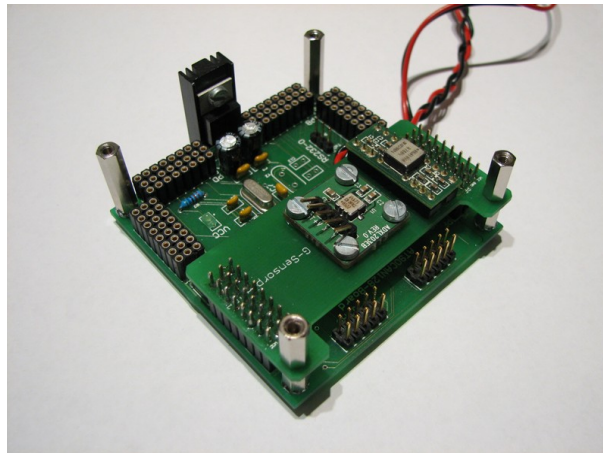


Abbildung 4.6: Platinenlayout für das Gyroskop und einen G-Sensor

Die Platine kann nicht nur das Gyroskop, sondern auch einen Beschleunigungssensor aufnehmen. Dazu sind die vier Bohrungen in der Mitte vorgesehen. Der Beschleunigungssensor wurde deshalb mittig angeordnet, damit dieser, wenn die Elektronik in einem passenden Gehäuse verbaut wird, intuitiv ausgerichtet werden kann. Wenn dieser nicht mittig zur Fahrtrichtung ausgerichtet ist, werden in Kurven verfälschte Werte geliefert. Für das Gyroskop ist der 20-Pin Sockel vorgesehen.

Die dreireihigen Blöcke (links und rechts) stellen den Kontakt zum Microcontroller her und werden in die Anschlussleisten von einem Port eingesteckt. Die Platine stellt so eine Brücke zwischen den Ports her und wird dadurch schwingungsärmer. Die Leisten für die Stromversorgung des linken Ports sind zudem durchkontaktiert um Wackelkontakte in der Stromversorgung des Gyroskops zu vermeiden. Die Pins des Gyroskops für die Winkelgeschwindigkeit und Temperatur sind direkt mit den A/D-Wandlern verbunden. Die Bohrungen an der Stirnseite der Platine können zusätzlich

zur mechanischen Verbindung genutzt werden. Die Aussparung zwischen den Bohrungen verhindert das die darunter liegenden Anschlüsse verdeckt werden.

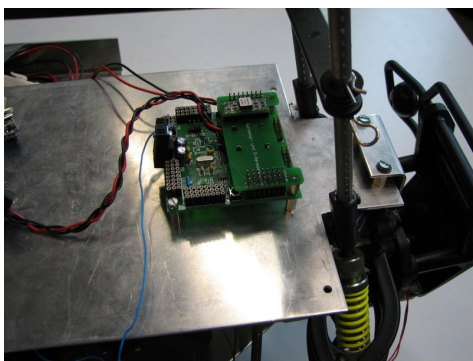


*Abbildung 4.7: Mit Sensoren ausgerüsteter Microcontroller*

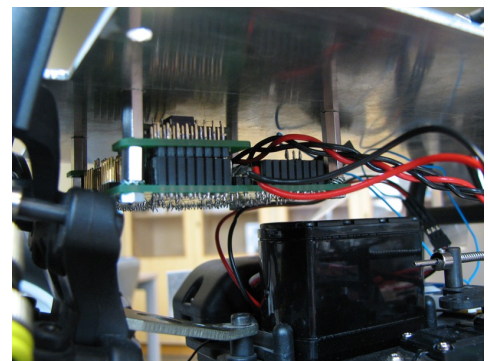
Um den Microcontroller ebenfalls sicher mit Strom zu versorgen und eine ebenso sichere Kommunikation über den CAN-Bus zu ermöglichen, sind diese Verbindungen fest verlötet.

#### 4.6 Integration in das Fahrzeug

Die am besten geeignete Unterbringung für den Microcontroller ist direkt unter der Rechnerplattform, denn so kann der Beschleunigungssensor mittig zur Fahrtrichtung angebracht werden. Zudem ist diese Variante platzsparend. Für die Tests mit dem Fahrzeug war kein Rechner auf dem Fahrzeug nötig, so konnte der Microcontroller leicht zugänglich auf der Plattform montiert werden.



*Abbildung 4.9: Der Microcontroller auf der Rechnerplattform platziert*



*Abbildung 4.8: Der Microcontroller unterhalb der Plattform*

# 5 Test

## 5.1 Labortest

Das Gesamtsystem muss nun eingestellt und beurteilt werden. Natürlich haben schon parallel zur Entwicklung verschiedene Tests stattgefunden. Diese waren nicht systematischer Natur, aber es zeigte sich, dass es zwei Parameter gibt, die auf die Regelung Einfluss nehmen.

- Die Driftgeschwindigkeit des Winkels in  $^{\circ}$  / Zeiteinheit
- Der angezeigte Winkel in  $^{\circ}$

Mit der Driftgeschwindigkeit wird die Geschwindigkeit bezeichnet, mit der die Anzeige des Gyroskops in die eine oder andere Richtung abdriftet, ohne dass das Gyroskop bewegt wird. Idealerweise sollte es keine Abweichung geben. Damit optimale Messwerte zur Verfügung stehen, wird der komplette Weg von dem Analogsignal bis zur Integralrechnung untersucht. Es können folgende Parameter konfiguriert werden.

- Die Abtastrate des A/D-Wandlers
- Das *fixedDelay* (Intervallbreite für Integralrechnung)
- Der Parameter für die Offset-Elimination
- Die Filtergröße für den gleitenden Mittelwert

Die Tests wurden nach einer Zeit von mindestens drei Minuten nach dem Einschalten des Gyroskops gemacht. Nach dieser Zeit sollte das Gyroskop die optimale Genauigkeit erreicht haben.

### 5.1.1 Bestimmung der maximalen Abtastrate

Bei der Bestimmung der optimalen Abtastrate gilt, dass die Genauigkeit der Abbildung des Analogsignals mit der Frequenz steigt. Idealerweise sollte die A/D-Wandlung so oft wie möglich durchgeführt werden. Prinzipiell ist es möglich die höchste Frequenz aus den technischen Daten des Microcontroller herzuleiten, aber

auf ihm laufen eine Reihe von Prozessen deren Zeitbedarf nicht spezifiziert ist. Deshalb werden in diesem Test verschiedene Abtastraten probiert und anschließend die grafische Auswertung beurteilt um Messfehler zu erkennen. Um Messfehler erkennen zu können, wurde vorher das Analogsignal analysiert.



Abbildung 5.1: Das Rauschen des Gyroskops mit einem Oszilloskop gemessen

Das Rauschen des Gyroskops bewegt sich in einem Bereich von + 15 mV bis -15mV. Betrachtet man die Auflösung des A/D-Wandlers mit  $5V/1024 = 4,9 \text{ mV} / \text{Integerwert}$ , darf das Signal um ungefähr  $15\text{mV} / 4,9\text{mV} = 3$  Integerwerte um seinen Mittelwert schwanken. Die Frequenz des Rauschen beträgt  $32 \text{ Schwingungen} / 20 \text{ } \mu\text{s} = 1,6 \text{ MHz}$ . Einige Schwingungen liegen unter dem Messbereich des A/D-Wandlers. Auffällig sind die Amplituden im 15mV-Bereich. Sie treten in einer Frequenz von  $3 \text{ Amplituden} / 26 \text{ } \mu\text{s} = 115,38 \text{ kHz}$  auf und ähneln ein wenig dem Verlauf einer Entlade-Kurve eines Kondensators.

In diesem Test wurde die Abtastrate kontinuierlich erhöht, bis Verfälschungen der eintreffenden Daten erkennbar wurden. Dabei wurde das Gyroskop nicht bewegt. Der Gleitende Mittelwertfilter wurde ausgeschaltet.

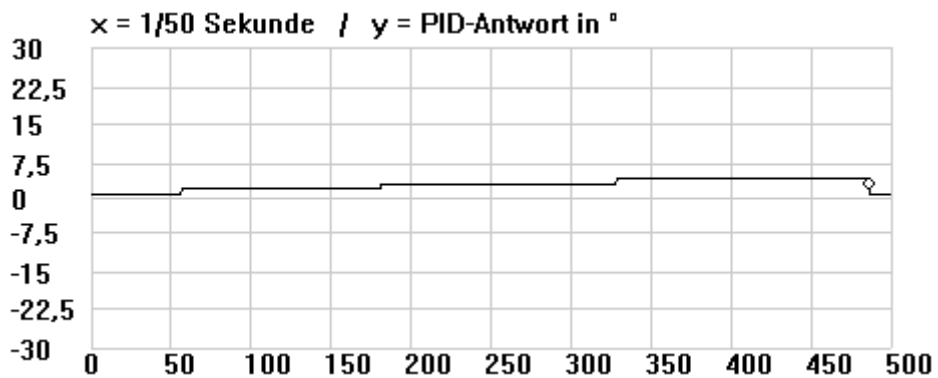
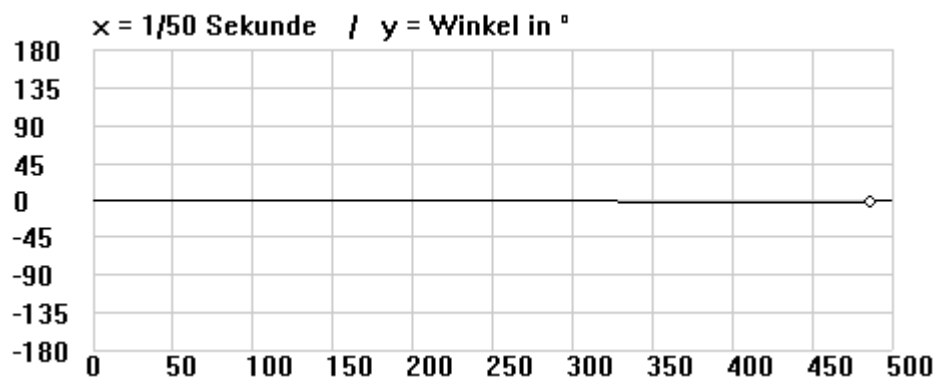
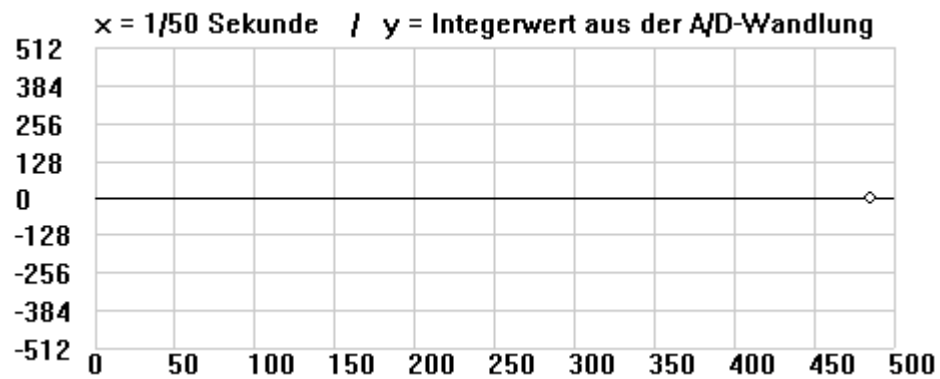


Abbildung 5.2: Abtastung mit 50 Hz

Das Diagramm mit den Integerwerten zeigt noch bei 50 Hz eine glatte Linie und der abgelesene Winkel beträgt nach fast 10 Sekunden  $-3,65^{\circ}$ . Der Drift ist nicht ideal, aber noch im Rahmen. Die Reglerantwort auf die Ungenauigkeit fällt moderat aus.

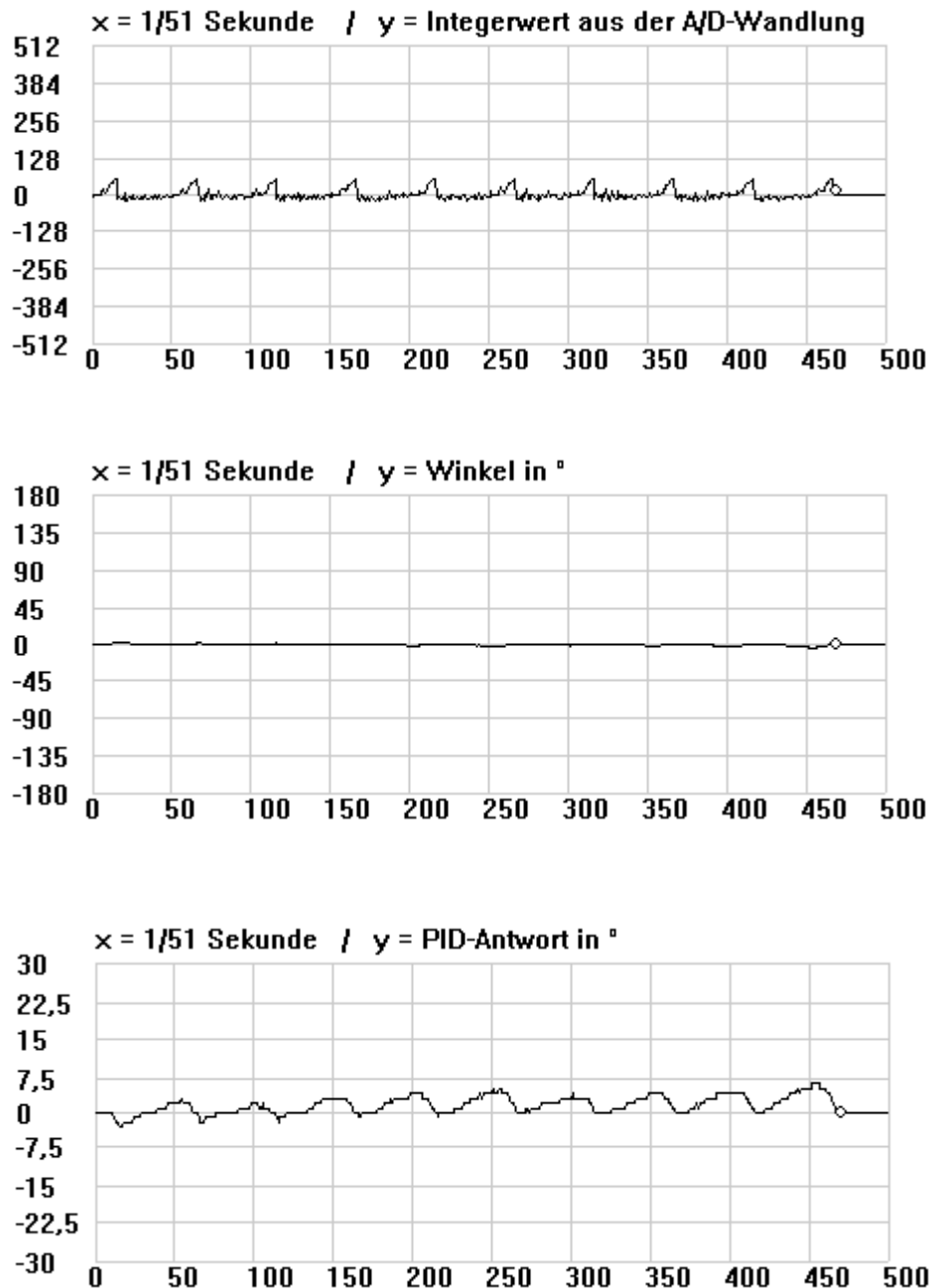


Abbildung 5.3: Abtastung mit 51 Hz

Ab 51Hz arbeitet das System unter den gleichen Bedingungen nicht mehr zuverlässig. Kontinuierlich sind Ausschläge zu erkennen. Die Reglerantwort zeigt, dass die Daten unbrauchbar sind. Auffällig ist auch, dass der Mittelwert, der als *Noiseaverage* bezeichnet wird, ungefähr 53 Werte unter dem bei 50Hz liegt (abgelesen von der Visualisierungssoftware).

Die ermittelte Grenze für die Abtastung liegt also bei 50Hz. Aus der Spezifikation des Gyroskops leitete sich die Anforderung ab, mit mindestens 80Hz abzutasten um die

gesamte Bandbreite des Gyroskops nutzen zu können. Diese kann nicht erfüllt werden.

### 5.1.2 Bestimmung des optimalen *fixedDelay*

Der Parameter *fixedDelay* beschreibt die Intervallbreite für die Integration, um von der Winkelgeschwindigkeit auf den Winkel zu schließen und entspricht dem Zeitwert zwischen den Messwerten. Um den Parameter genau zu bestimmen wurde nicht auf die eingestellte Frequenz der Abtastung vertraut, sondern ein 90° Referenzwinkel gemessen. Den Referenzwinkel bilden dabei die Kanten des Microcontroller.

Der Microcontroller wurde mit dem Gyroskop auf eine Seite gestellt, der Winkel anschließend in der Software zurückgesetzt. Dann wurde der Microcontroller zügig auf die benachbarte Kante gedreht und der Winkel abgelesen. Dies musste schnell passieren, damit der vorhandene Drift das Messergebnis so wenig wie möglich verfälschte. Das Prozedere wurde für die entgegengesetzte Richtung wiederholt, anschließend der Parameter angepasst. Als bestes Ergebnis für die Intervallbreite wurde der Wert 0,014 ermittelt. Die Toleranz für den angezeigten Winkel beträgt dabei weniger als  $\pm 3^\circ$ .

### 5.1.3 Parameter für die Offset-Elimination

Aus der A/D-Wandlung werden nur positive Werte generiert, deren Mittelwert möglichst exakt bestimmt werden muss, damit dieser als Offset von den Werten abgezogen werden kann. Die gelieferten Werte bewegen sich im Ruhezustand des Gyroskops ungefähr um  $\pm 3$  A/D-Wandler-Schritte, dabei sind die Werte nicht gleichmäßig verteilt. Natürlich könnte die Berechnung großzügig dimensioniert werden, indem besonders viele Werte einfließen. Aber hier muss ein Kompromiss gefunden werden zwischen dem Zeitaufwand für die Mittelwertberechnung und der Genauigkeit, mit der gearbeitet werden soll. Insbesondere sollte man dabei bedenken das der Mittelwert nur zu einem bestimmten Zeitpunkt gültig ist. Wenn das Gyroskop auf Grund von Umweltbedingungen wie der Temperatur seinen Arbeitspunkt ändert, muss ein neuer Mittelwert berechnet werden. Das heißt, die Zeitspanne zwischen Berechnung und Einsatz darf nicht zu lang sein.

Daher ist eine Untersuchung des Driftverhaltens im Bezug auf die Mittelwertberechnung wichtig. In dem Test wird bei unterschiedliche Anzahlen von Werten ( $N$ ), die Winkeldifferenz nach 30 Sekunden, bei ruhendem Gyroskop, gemessen. Um repräsentativere Werte zu erhalten, wird mit jeder Konfiguration fünf mal gemessen.

<b>Offset-Elimination</b>					
<b>Winkel nach 30 Sekunden in °</b>					
<b>N</b>	<b>Messung 1</b>	<b>Messung 2</b>	<b>Messung 3</b>	<b>Messung 4</b>	<b>Messung 5</b>
50	-3	4	-3	-2	0
100	-5	-6	5	-5	4
250	1	3	1	-2	1
500	-3	0	0	2	4
1000	1	1	1	0	3
2000	2	-1	-3	0	-3

*Tabelle 2: Zusammenhang zwischen Driftverhalten und Offset-Elimination*

Die Nachkommastellen sind bei den Messungen abgeschnitten, da diese sich zu schnell und ohne erkennbare Systematik ändern. Auch geht aus dieser Tabelle nicht hervor, dass die Werte in dem Zeitraum in einem Bereich von +/- dem Endwert um den Nullpunkt bewegen können.

Wirklich starke Schwankungen lassen sich aus den Werten nicht ablesen. Ab 250 Werten scheint fast keine Verbesserung mehr möglich, wenn man die Extremwerte einer Messreihe betrachtet. Für die Initialisierung des Mittelwertes könnten 1000 Werte eine gute Wahl sein. Bei dieser Anzahl dauert die Initialisierung aber 20 Sekunden, deshalb werden als Kompromiss 500 Werte genommen.

#### **5.1.4 Filtergröße für gleitenden Mittelwert**

Der Filter für den Gleitenden Mittelwert liefert bei einem neuen Messwert, den Mittelwert der letzten  $N$  Werte. Bei diesem Filter wirken sich Schwankungen weniger Stark aus, was vorteilhaft für das Driftverhalten sein kann. Bei zu großen  $N$  werden Änderungen im Signal gedämpft, beziehungsweise verzögert. In diesem Test wird nun die Eignung eines solchen Filters und dessen ideale Größe untersucht. Der Versuchsablauf ist der gleiche, wie für die Offset-Elimination. Geprüft werden Werte



von  $N=1$  bis  $N=32$ , ab diesem Wert werden die negativen Eigenschaften des Filters erkennbar.

<b>Gleitender Mittelwertfilter</b>					
<b>Filtergröße</b>	<b>Winkel nach 30 Sekunden in °</b>				
<b>N</b>	<b>Messung 1</b>	<b>Messung 2</b>	<b>Messung 3</b>	<b>Messung 4</b>	<b>Messung 5</b>
1	-2	-2	1	4	-3
2	0	-1	4	4	3
4	0	0	0	0	1
8	-1	0	-1	4	-5
16	-3	0	3	-13	-7
32	-4	-2	1	-5	-6

*Tabelle 3: Zusammenhang zwischen Driftverhalten und gleitendem Mittelwert*

Das beste Ergebnis liefert ein Filter der Größe 4. Allerdings gibt es, trotz des hier gemessenen Ergebnisses, Abweichungen. Es muss vereinzelt damit gerechnet werden, dass der Drift bei  $\pm 4^\circ$  liegen kann. Auch sei hier erwähnt, dass die Anzeige innerhalb der 30 Sekunden im Bereich  $\pm$  dem Endwert schwanken kann.

### 5.1.5 Spannungsfestigkeit

Ein wichtiger Punkt bei einem autonomen Fahrzeug, ist die Unempfindlichkeit der Komponenten von Spannungsschwankungen, da diese mit Akkus versorgt werden. Auch wenn eine Unempfindlichkeit nicht gewährleistet werden kann, ist zumindest ein vorhersagbares Verhalten wichtig. Aus diesem Grund wurde eine Tabelle erstellt, die die Spannung des Ausgangs des Gyroskops und die Versorgungsspannung des Microcontrollers darstellt. Das Gyroskop wird von dem Microcontroller versorgt.

<b>Versorgungsspannung Microcontroller (V)</b>	<b>Ausgangsspannung Gyroskop (V)</b>
6,5	2,49
6	2,49
5,5	2,55
5	2,34
4,5	1,7

*Tabelle 4: Tabellarische Darstellung eines Spannungsabfall der Versorgungsspannung*

Anhand dessen lässt sich erkennen, dass die Ausgangsspannung erst ein bisschen

ansteigt und danach abfällt. Das konkrete Verhalten des Systems bei einem Spannungsabfall wird mit einer Grafik verdeutlicht.

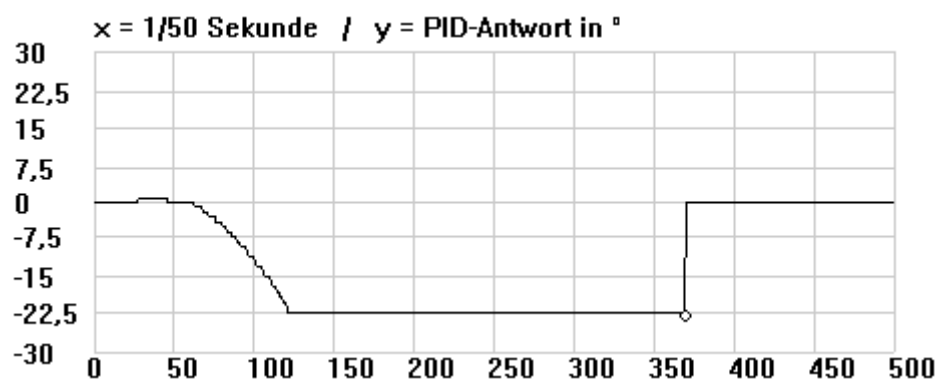
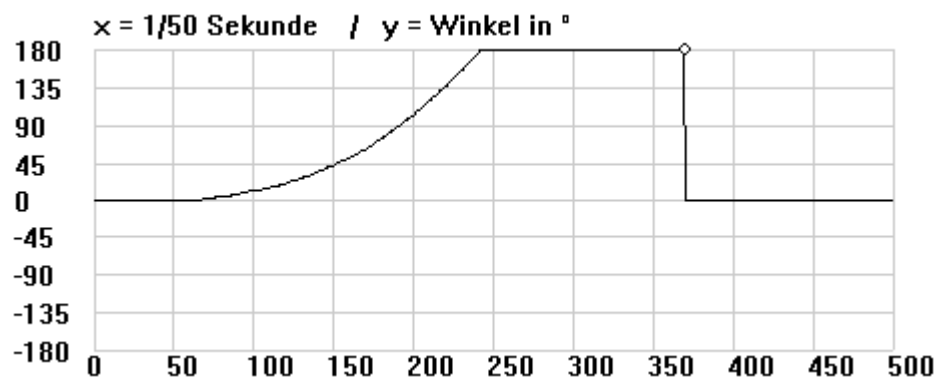
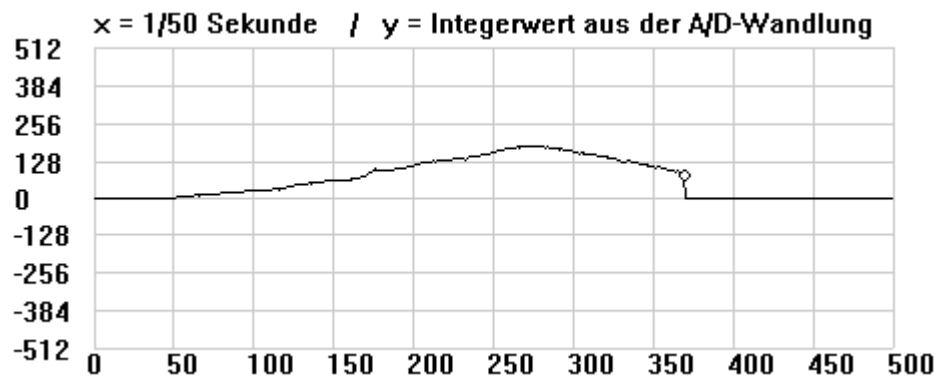


Abbildung 5.4: Verhalten des Systems bei einem Spannungsabfall

Die Grafik zeigt den Verlauf von 6,5V (Anfang Diagramm) bis 4,5V (Cursorposition). Bei 4,6V werden noch Messwerte aufgenommen und über den CAN-Bus verschickt. Das passiert bei dieser Spannung nicht mehr flüssig, sondern ruckartig. Unter 4,6V bleibt das System einfach stehen. Deutlich ist die Verschiebung des Nullpunktes zu erkennen und darunter die dadurch entstandene Verfälschung des Winkels.

### 5.1.6 Fehlerabschätzung und Aliaseffekt

In den vorangegangenen Tests wurden die gewählten Parameter anhand der Winkeldifferenz nach 30 Sekunden beurteilt. Dabei sind teilweise recht hohe Schwankungen in der Anzeige des Winkels beobachtet worden. Eine Kurvenfahrt wird bei kleineren Radien wesentlich kürzer dauern, als 30 Sekunden. Wenn innerhalb dieser Zeit Schwankungen entstehen, wird die Regelung dadurch wesentlich beeinträchtigt. Folgende Grafik visualisiert diese Schwankungen, die dem Aliaseffekt zu zuordnen sind. Der Screenshot zeigt den Verlauf der Null-Linie, sowie den angezeigten Winkel innerhalb 50 Sekunden. Die Werte aus der A/D-Wandlung sind mit einem Mittelwertfilter der Größe 64 gefiltert, um das Signal zu glätten und den Verlauf deutlicher zu machen.

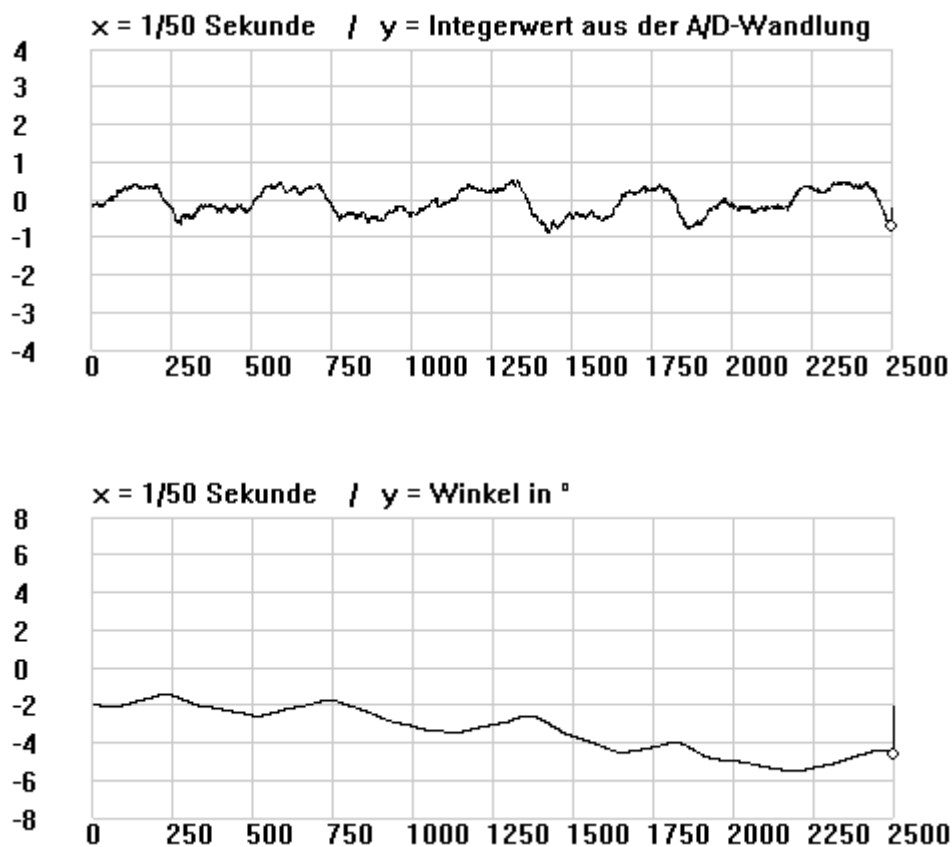


Abbildung 5.5: Verlauf der Schwankungen der Winkelanzeige

Betrachtet man das Bild 5.1 und die Schlussfolgerungen daraus, stellt man fest, dass das Abtasttheorem mehrfach verletzt wird. Zum einen gibt es die Frequenz von 1,6 MHz im Rauschen mit einer sehr geringen Amplitude, so dass der A/D-Wandler nicht

immer die Werte in seinem Raster von 4,9 mV eindeutig einordnen kann. Die Werte können im Grenzbereich zwischen zwei A/D-Werten liegen und so zu weiteren Effekten führen, die sich mit einer unzureichenden Abtastrate alleine nicht erklären lassen. Zum anderen gibt es eine weitere Frequenz von 115 kHz, deren Amplitude von dem A/D-Wandler registriert werden kann und zu einem Aliaseffekt führen müsste und dies offensichtlich auch im Bild 5.5 tut.

Eine Vorhersage über den zu erwartenden Fehler ist für die Beurteilung des Gesamtsystems wichtig. Als Parameter für die Beurteilung eignen sich die folgenden Parameter, die bei ruhendem Gyroskop gemessen werden.

- Die maximale und minimale Winkeldifferenz zwischen zwei Messwerten
- Die maximale und minimale Winkeldifferenz innerhalb einer Sekunde
- Die mittlere Winkeldifferenz über den gesamten Zeitraum

Um die Wahrscheinlichkeit zu erhöhen, damit jeweils der minimale und maximale Wert gemessen werden kann, wird der Test über 10 Minuten durchgeführt. Die Winkeldifferenz innerhalb einer Sekunde wird aus dem Mittelwert der letzten 50 Werte gebildet. Alle angegebenen Werte ergeben sich nur aus dem Rauschen des Gyroskops.

<b>Minimale und maximale Winkeländerungen innerhalb 10 Minuten</b>		
	<b>Minimum</b>	<b>Maximum</b>
<b>Winkeldifferenz zwischen zwei Messwerten</b>	0,084	0,966
<b>Winkel Differenz pro Sekunde</b>	0,002	0,641
<b>Durchschnittliche Winkeldifferenz</b>	-0,001	

*Tabelle 5: Fehlmessungen bei ruhendem Gyroskop*

Mit dieser Tabelle ist es nun möglich eine Fehlerabschätzung in Bezug auf die Regeldauer, also die Zeit, die benötigt wird um die Kurve zu beenden, durchzuführen. Die Winkeldifferenz zwischen zwei Messwerten soll an dieser Stelle verdeutlichen, wie stark die Abweichung selbst im Ruhezustand sind. Die Werte sind auf eine Änderung innerhalb einer Sekunde skaliert ( $^{\circ}$ \*Hz), die Maßeinheit beträgt eigentlich  $^{\circ}$ /Hz (Winkeländerung innerhalb einer Periode). Diese Werte ändern sich

aber so schnell, dass sie für eine Fehlerabschätzung ungeeignet sind. Der Mittelwert über 50 Messwerte eignet sich dazu eher.

Als durchschnittlicher Fehler wird der Mittelwert aus dem Minimum und Maximum berechnet. Bei der Benutzung des Systems sollte mit folgenden Abweichungen pro Sekunde gerechnet werden:

<b>Fehler in °/ Sekunde</b>		
<b>Minimum</b>	<b>Durchschnitt</b>	<b>Maximum</b>
0,002	0,322	0,641

*Tabelle 6: Durchschnittlicher Fehler*

Die Durchschnittliche Winkeldifferenz beträgt über den gesamten Zeitraum von 10 Minuten  $-0,001^\circ$ . Als Grundlage für diesen Wert dienen nicht die A/D-Werte direkt, sondern der bereits berechnete Winkel mit seinen Änderungen. Das heißt, sämtliche Berechnungs- / Rundungsfehler fließen darin ein. Damit ist dieser Wert als Kontrolle geeignet um von dem Winkel wieder auf die Winkelgeschwindigkeit zu schließen und diese mit der angezeigten Winkelgeschwindigkeit des Gyroskops zu vergleichen. Die angezeigte Winkelgeschwindigkeit des Gyroskops beträgt im Ruhezustand  $0^\circ / \text{Sekunde}$ . Verglichen mit den  $-0,001^\circ$  im Mittel über 10 Minuten scheinen durch die Software fast vernachlässigbare Fehler zu entstehen.

Man muss dazu die oben genannten Abweichungen zwischen den Messwerten betrachten. Daraus lässt sich herleiten, dass innerhalb einer langen Zeitspanne (innerhalb von Minuten) betrachtet, die Anzeige des Gyroskops sehr gut ist, betrachtet man eine sehr kurze Zeitspanne (innerhalb eines Messintervalls) sind die Schwankungen sehr stark.

### **5.1.7 Integrationstest mit der Hauptsteuerung**

Die Regler-Software wird von der Hauptsteuerung für die Fahrt einer definierten Kurve benutzt. Im Labortests konnte das einwandfreie Zusammenspiel beider festgestellt werden. In dem Test wurde von der Hauptsteuerung eine Kurve initiiert, indem der Regelung ein neuer Sollwert mitgeteilt wurde. Die Regelung bekam

anschließend die Kontrolle über die Lenkung und hat diese gemäß der Vorgabe eingestellt. Um die Bewegung des Fahrzeugs zu simulieren, wurde der Tisch, auf dem das Fahrzeug steht, langsam gedreht. Nach Erreichen des vorgegebenen Winkels erfolgte die Meldung an die Hauptsteuerung, dass die Aufgabe erfüllt worden ist. Die Kontrolle über die Lenkung wurde der Regelung entzogen und der Programmablauf wurde fortgeführt.

Ein vorgezogener Test (die vorangegangenen Testergebnisse sind nicht eingeflossen) auf dem Fahrzeug in freier Fahrt ist nicht erfolgreich gewesen. Anhand der Logfile-Auswertung wurde ein sehr starkes Driften festgestellt, was auf eine Fehlkonfiguration zurückgeführt wurde.

## **5.2 Fahrzeugtest**

Nachdem im Labor die Eigenschaften des Gyroskops und der Software überprüft wurde, muss nun das Zusammenspiel mit dem Fahrzeug getestet werden. Die folgenden Tests überprüfen die Einwirkung des Motors auf die Messgenauigkeit und die Genauigkeit der Lenkmechanik.

### **5.2.1 Störungen durch Vibrationen**

Während der Motor mit seinem Kolben den gesamten Wagen in Schwingung versetzt, misst das Gyroskop mit seiner ebenfalls schwingenden Mikromasse die Richtungsänderungen. Es liegt nahe, dass durch den Motor Störungen auftreten können. In diesem Test, werden die Hinterräder des Fahrzeugs abgebaut, im Stand der Motor über den gesamten Drehzahlbereich laufen gelassen und dabei die Anzeige des Gyroskops auf Anomalien überwacht.

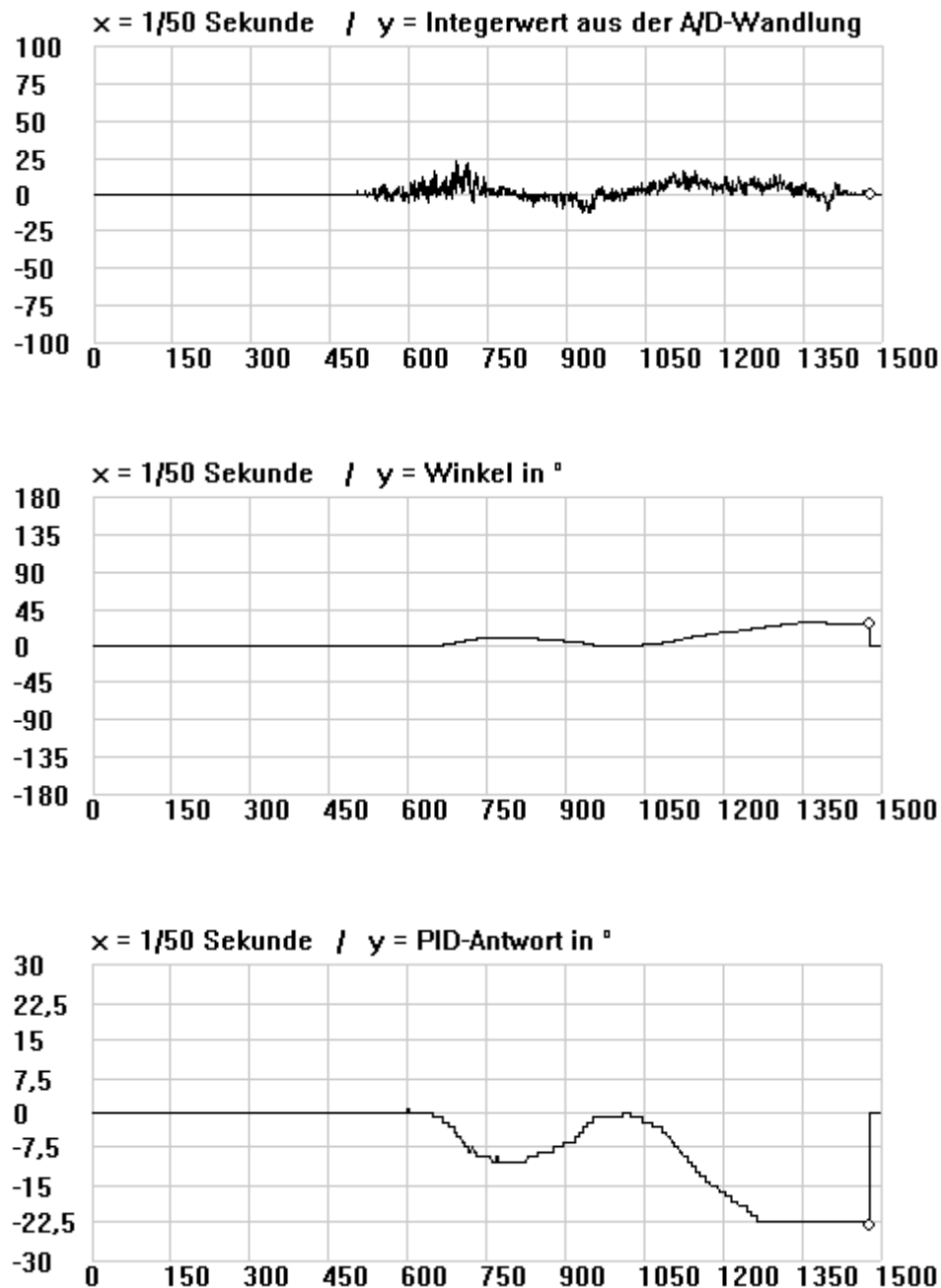


Abbildung 5.6: Darstellung des Einflusses der Vibrationen

Der Bereich ab Anfang des Diagramms bis zum Wert 1250 stellt den kompletten Drehzahlbereich dar. Ab ungefähr der Hälfte werden Verfälschungen sichtbar. In den unteren Drehzahlen funktioniert das Gyroskop ohne erkennbare Probleme.

### 5.2.2 Genauigkeit der radiusbasierten Kurvenfahrt

Bei der radiusbasierten Kurvenfahrt spielen alle Berechnungen und mechanischen Gegebenheiten ein. Der Test umfasst deshalb nicht nur das Abfahren eines Radius,

sondern auch, die Erkennung des Systems, wann eine Kurve beendet ist, sowie die Abweichung von dem gewünschten Winkel.

Für den Versuch wurde eine 180° Kurve gewählt. Zum einen weil diese Kurve relativ lange dauert und damit der Fehler über die Zeit deutlich wird, zum anderen weil der Radius leicht mit einem Maßband abgelesen werden kann. Dazu wird das Fahrzeug um 90° versetzt auf das Maßband gesetzt und anschließend durch die 180° Kurve geschoben. Nach den 180° müsste das Fahrzeug wieder im 90° Winkel auf dem Maßband zum stehen kommen, sofern das System ideal arbeitet. Mit der gemessenen Distanz auf dem Maßband hat man den Durchmesser der Kurve und damit den Radius.

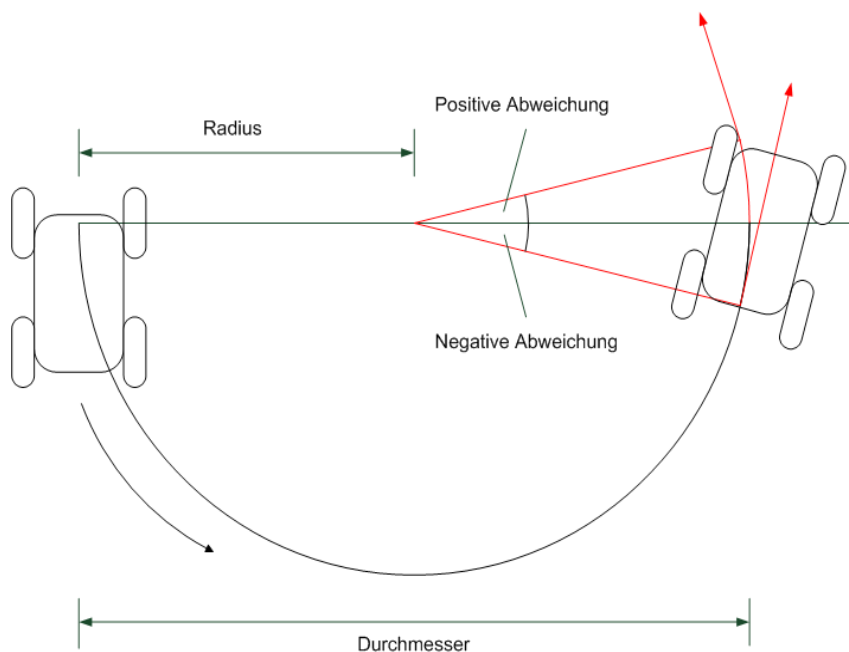


Abbildung 5.7: Darstellung des Versuchs zur radiusbasierten Kurvenfahrt

Bei den ersten Probeläufen hat sich heraus gestellt, dass mit dem Proportionalitätsbeiwert  $K_p=1$  das Fahrzeug für diesen Test zu sanft aus der Kurve heraus fährt. Selbst wenn man den Eindruck hatte, dass das die Regelung fertig sein müsste, weil das Fahrzeug fast unmerklich die Richtung änderte, fehlte noch etwas um die Kurve zu beenden. Der Grund für dieses Verhalten liegt in dem Verhältnis von 1:1 zwischen Regelabweichung und Stellwert. Fehlen  $0,5^\circ$ , wird die Lenkung ebenfalls um  $0,5^\circ$  eingeschlagen, was eine minimale Lenkwirkung zur Folge hatte. Die Regelabweichung wurde entsprechend langsam abgebaut. Prinzipiell ist das



Verhalten absolut wünschenswert, denn genau so fährt man auch im täglichen Leben. Für diesen Test wird jedoch eine gewisse Präzision verlangt. Der Proportionalitätsbeiwert wurde dazu auf  $K_p=5$  gestellt.

	Linkskurve					
	1		2		3	
Soll-Radius	Ist-Radius	Winkeldiff.	Ist-Radius	Winkeldiff.	Ist-Radius	Winkeldiff.
1,0m	1m	5°	1m	5°	1m	5°
2,0m	2m	5°	1,9m	5°	1,9m	5°
3,0m	2,8m	5°	3m	5°	3,5m	5°
4,0m	4m	20°	4,0m	20°	4,1m	20°

	Rechtskurve					
	1		2		3	
Soll-Radius	Ist-Radius	Winkeldiff.	Ist-Radius	Winkeldiff.	Ist-Radius	Winkeldiff.
1,0m	1,1m	-5°	1,1m	-5°	1,1m	-5°
2,0m	2,2m	-10°	2,1m	-10°	2,2m	-10°
3,0m	3m	-10°	3,4	-10°	3,2m	-15°
4,0m	4,5m	-10°	4,6m	-10°	4,6m	-10°

*Tabelle 7: Messergebnis der radiusbasierten Kurvenfahrt - unkorrigiert*

Die verbliebene Winkeldifferenz wurde geschätzt, da es kein taugliches Messinstrument für diesen Zweck gab. Mit den Werten soll eine Tendenz gezeigt werden, die sich während der Tests ergeben hat. Auch der gemessene Ist-Radius unterliegt Ungenauigkeiten und wurde auf volle 10cm gerundet.

Aus der Tabelle kann abgelesen werden, dass Linkskurven tendenziell um 5° zu weit gefahren werden und Rechtskurven 10° zu kurz. Die Radien der Rechtskurve sind etwas größer, als die der Linkskurve, die erstaunlich gut sind. Es deutet sich ein systematischer Fehler an, der durch eine Illustration des Ergebnisses veranschaulicht werden soll.

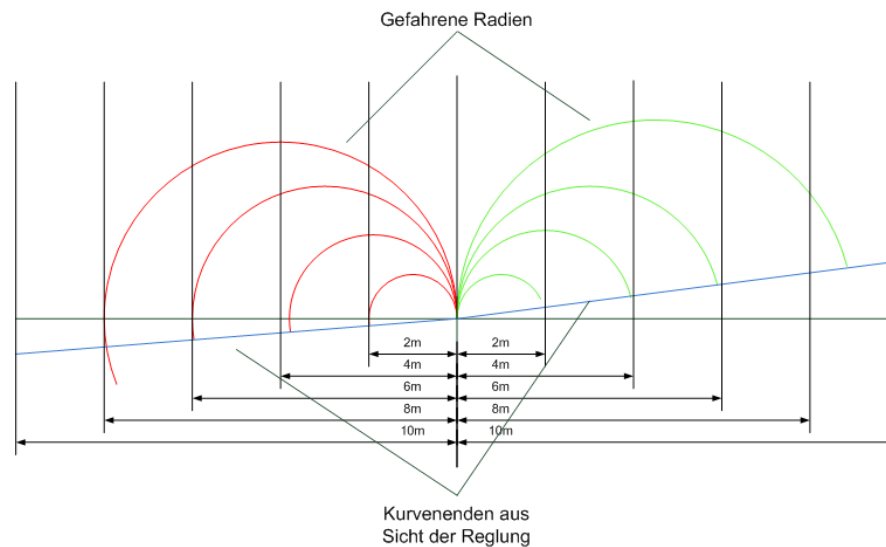


Abbildung 5.8: Illustration des systematischen Fehlers

Es zeigt sich, dass die beiden Richtungen des Gyroskops nicht gleich angezeigt werden. Über die Ursachen kann nur spekuliert werden, aber dass die mechanische Komponente des Gyroskops damit zu tun haben könnte, ist sehr wahrscheinlich. Vermutlich spielen Fertigungstoleranzen ein, die von Gyroskop zu Gyroskop neu ermittelt werden müssen. Bei einer direkten Betrachtung und Verarbeitung des Analogsignals fällt dieser Unterschied nicht auf, durch die Integration summieren sich jedoch die Abweichungen und es entsteht eine erkennbare Asymmetrie.

### 5.2.3 Verbesserung der radiusbasierten Kurvenfahrt

Als erste Strategie wurde die Anpassung des Sollwertes gewählt, dabei wird entsprechend der Richtung der Sollwert entweder erhöht oder verringert. Die Werte wurden nach mehreren Durchläufen angepasst, bis der Fehler nicht mehr erkennbar war. Die erhöhten Radien bei der Rechtskurve konnten etwas korrigiert werden, indem die in der Software vorgegebene mechanische Grenze für den Rechtsauschlag von  $24^\circ$  auf  $23,5^\circ$  verringert wurde. Somit wurden die PWM-Signale auf einen kleineren Wertebereich skaliert und der Lenkausschlag vergrößerte sich bei gleicher Stellvorgabe.

$$\text{Sollwert} = \begin{cases} \text{Sollwert} \geq 0 & : \text{Sollwert} + \text{Sollwert} \cdot 0.0125 \\ \text{Sollwert} < 0 & : \text{Sollwert} - \text{Sollwert} \cdot 0.02 \end{cases}$$

Mit den Parametern für die Sollwertkorrektur konnten die folgenden Ergebnisse erzielt werden.

	Linkskurve					
	1		2		3	
Soll-Radius	Ist-Radius	Winkeldiff.	Ist-Radius	Winkeldiff.	Ist-Radius	Winkeldiff.
1,0m	1m	0°	1m	1°	1m	1°
2,0m	2m	0°	2m	0°	2m	1°
3,0m	3,1m	1°	3,2m	1°	3,2m	1°
4,0m	4,2m	1°	4m	0°	4,1m	0°

	Rechtskurve					
	1		2		3	
Soll-Radius	Ist-Radius	Winkeldiff.	Ist-Radius	Winkeldiff.	Ist-Radius	Winkeldiff.
1,0m	1m	0°	1m	0°	1m	-1°
2,0m	2m	-1°	2m	-1°	2m	0°
3,0m	3,1m	1°	3,1m	0°	3,1m	0°
4,0m	4,2m	0°	4,2m	0°	4,3m	0°

Tabelle 8: Messergebnis der radiusbasierten Kurvenfahrt - korrigiert

Auffällig ist die Verbesserung der Radiengenauigkeit bei der Linkskurve, eigentlich hat keine bewusste Optimierung stattgefunden. Die Verbesserung ist durch die erhöhte Ablesegenauigkeit zu erklären, da die Regelung stets in unmittelbarer Nähe zum Maßband die Kurven beendet hat und damit viel genauer der Zusammenhang zwischen dem Radius und der Position des Fahrzeugs hergestellt werden konnte.

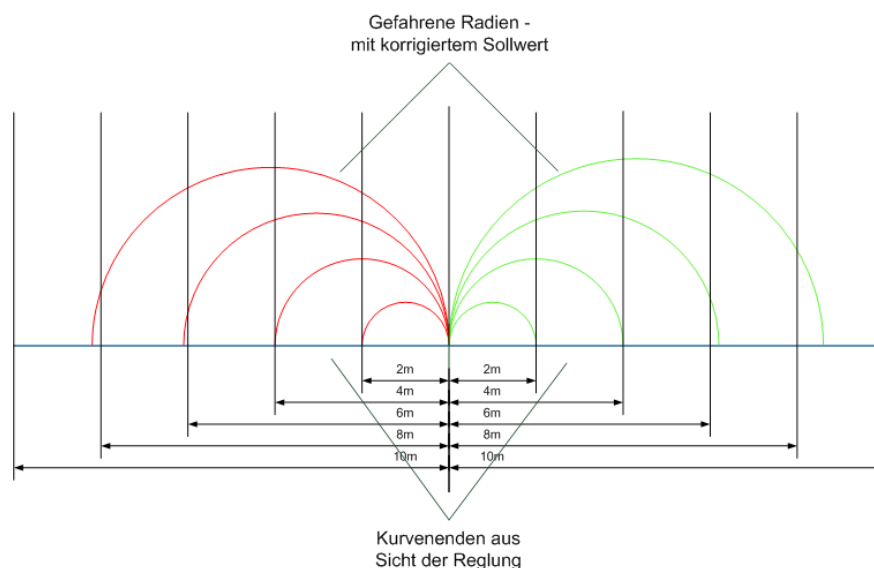


Abbildung 5.9: Illustration des systematischen Fehlers - nach der Korrektur

Die Korrektur der Sollwertvorgabe, bietet zwar jetzt ein ausgezeichnetes Ergebnis, aber der eigentliche angezeigte Winkel leidet noch immer an der Asymmetrie. Damit die Regelung den Sollwert nicht nachträglich ändern muss und auch andere Anwendungen von einer Korrektur profitieren können, wurde der Fehlerausgleich in die Berechnung des Winkels verlagert. Die eintreffenden Messwerte (mit eliminierten Offset und gleitenden Mittelwertfilter) mit positiven Vorzeichen werden dazu etwas verringert, die mit negativen Vorzeichen im Betrag vergrößert.

$$\text{Messwert} = \begin{cases} \text{Messwert} \geq 0 & : \text{Messwert} - (\text{Messwert} \cdot 0.019235) \\ \text{Messwert} < 0 & : \text{Messwert} + (\text{Messwert} \cdot 0.0052) \end{cases}$$

Sinngemäß bedeutet diese Vorgehensweise, dass die positiven Winkeländerungen minimal weniger berücksichtigt werden und die Negativen minimal mehr. Die Zahlen wurden empirisch festgestellt und funktionieren fast genauso gut wie die Korrektur der Sollwertvorgabe. Dabei wurde die Testreihe nicht komplett wiederholt, sondern nur jeder Radius einmal abgefahren. Es stellte sich eine Abweichung von ungefähr  $\pm 1^\circ$  heraus.

## 6 Zusammenfassung und Ausblick

Die Anforderungen an die Regelung konnten in Labortests nachweislich erfüllt werden. Auch die Einstellung der Radian funktioniert gut, obwohl das Modell stark vereinfacht wurde. Es ist anzunehmen, dass andere Fahrzeuge, die mit einer solchen Regelung ausgestattet werden, ähnliche Ergebnisse erzielen. Eine genauere Lenkmechanik würde auch eine weitere Auseinandersetzung mit der Abweichung der Radian lohnend erscheinen lassen. Diese ließ sich wahrscheinlich ebenfalls recht einfach korrigieren.

Die Vibrationen stören das Gyroskop ab einem bestimmten Drehzahlbereich. Für die weitere Verwendung des Systems sollte deswegen auf den oberen Geschwindigkeitsbereich des Fahrzeugs verzichtet werden. Dennoch sollte das Problem mit den Vibrationen unbedingt angegangen werden. In den Versuchsreihen mit dem Motor ist sogar deswegen eine kleinere Beschädigung aufgetreten - ein Spannungsstabilisator ist abgebrochen.

Aber nicht sämtliche Funktionen, die implementiert sind, wurden getestet. Die Kontrollen des Ist-Radius und Soll-Winkel konnten mangels einer Geschwindigkeitserfassung nicht getestet werden. Ebenso wenig konnte eine Aussage über die Wirksamkeit und Dimensionierung des D-Anteils getroffen werden. Eine abgeschlossene Teststrecke ist für einen solchen Test erforderlich, damit das Fahrzeug risikolos in eine entsprechende Situation gebracht und auf einer weitläufigen Fläche beobachtet werden kann.

Sollte ein weiteres System angewendet werden, sind bei dem neuen Gyroskop eine Reihe von Tests nötig, um die entsprechenden Korrekturparameter herauszufinden. Auf keinen Fall darf die Software nur kopiert werden.

### 6.1.1 Verbesserungen

Als großer Nachteil erwies sich die momentane A/D-Wandlung. Durch die Regler-Software muss der Offset herausgerechnet werden und es sind nur 50 Messungen pro Sekunde möglich. Auch das Rauschen bei 1,6 MHz sollte idealerweise gefiltert werden. In diesem Frequenzbereich befinden sich keine Informationen des Gyroskops.

Man könnte zum Beispiel die Referenzspannung (2,5 V) des Gyroskops ebenfalls digitalisieren, um den Offset zu filtern, indem dieser Wert von dem eigentlichen Messwert abgezogen wird (differentielle A/D-Wandlung).

Die ermittelten 50Hz der Abtastrate liegen deutlich unter dem Niveau des Microcontrollers. Normalerweise kann dieser mit bis zu 15 kHz digitalisieren. Durch die Software bleiben davon leider nur 50 Hz übrig. Es bietet sich an, nochmal genau zu prüfen, wie das gesamte Potenzial des Microcontrollers erschlossen werden kann.

Das Rauschen könnte man mit einem Analogfilter praktisch komplett eliminieren. Der Abstand zwischen maximaler Frequenz des Nutzsignals und dem Rauschen ist so hoch, dass keine Abstriche in der Signalqualität absehbar sind.

## Literaturverzeichnis

[1] John Geen, David Krakauer, *New iMEMS® Angular-Rate-Sensing Gyroscope*, <http://www.analog.com/library/analogDialogue/archives/37-03/gyro.html>

[2] ANALOG DEVICES, *Datasheet ADXR300*, 2004

[3] ATMEL, *Datasheet AT90CAN128*, 2007

[4] Wikipedia, *Alias-Effekt*, <http://de.wikipedia.org/wiki/Alias-Effekt>

[5] Collin Pein, *Geschwindigkeitsregelung mittels eines Beschleunigungssensors im autonomen Modellfahrzeug*, Bachelorarbeit HAW-Hamburg 2007

[6] ANALOG DEVICES, *Datasheet ADXRS300EB*, 2003

[7] Peter Busch, *Elementare Regelungstechnik*, 2002, ISBN: 3-8023-1918-4

[8] Kordian Kubat, *Eine hierarchische Steuerungsarchitektur mit Fuzzy-Regelung zur positionsmarkenbasierten Navigation eines autonomen Modellfahrzeugs*, Bachelorarbeit HAW-Hamburg 2007

[9] Sergej Meier, *Erfassung und Verarbeitung von Sensormesswerten für die Stabilisierung in Längs- und Querrichtung eines holonomen Fahrzeugs bei Kurvenfahrt*, Bachelorarbeit HAW-Hamburg 2006

# Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

---

Ort, Datum

---

Unterschrift