



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

# Bachelorarbeit

Martin Arvidsson

ARM - gesteuerte Antriebsschlupfregelung eines  
autonomen Modellfahrzeuges:  
Analyse, Design, Implementierung

Martin Arvidsson

ARM - gesteuerte Antriebsschlupfregelung eines  
autonomen Modellfahrzeuges:  
Analyse, Design, Implementierung

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. -Ing. Franz Korf  
Zweitgutachter : Prof. Dr. rer. nat. Stephan Pareigis

Abgegeben am 19. August 2008

**Martin Arvidsson**

**Thema der Bachelorarbeit**

ARM - gesteuerte Antriebsschlupfregelung eines autonomen Modellfahrzeuges: Analyse, Design, Implementierung

**Stichworte**

Antriebsschlupfregelung, Schlupferkennung, Geschwindigkeit, Beschleunigung,  $\mu$ -Kennlinie, Hall-Sensor, ARM7 LPC2468 Mikrocontroller, Time Triggered embedded System, Signalverarbeitung, autonomes Modellfahrzeug

**Kurzzusammenfassung**

In dieser Bachelorarbeit werden Eigenschaften und physikalische Grundlagen für eine verbesserte Fahrzeugstabilität während der Beschleunigung untersucht. Mit diesem Wissen und einer anschließenden Analyse am Modellfahrzeug werden Anforderungen abgeleitet, mit denen ein Design für eine Antriebsschlupfregelung entwickelt wird. Die Umsetzung und die dafür benötigte Signalauswertung, Schlupfbestimmung und Ansteuerung erfolgt auf einem ARM7 Mikrocontroller, der im Modellfahrzeug integriert wird.

**Martin Arvidsson**

**Title of the paper**

ARM - driven traction control system for an autonomous model vehicle: analysis, design, implementation

**Keywords**

Traction control system, slip detection, speed, acceleration,  $\mu$ -characteristic line, hall-sensor, ARM7 LPC2468 microcontroller, time triggered embedded system, signal processing, autonomous model vehicle

**Abstract**

This bachelor thesis investigates characteristics and physical basics to improve vehicle stability during acceleration. With this knowledge and analyses on the model vehicle requirements are derived. Based on these requirements a traction control system is developed. The program with its signal processing, slip detection and control tasks runs on an ARM7 microcontroller, which will be integrated to the model vehicle.

# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>6</b>
<b>Abbildungsverzeichnis</b>	<b>7</b>
<b>1. Einführung</b>	<b>9</b>
<b>2. Grundlagen</b>	<b>11</b>
2.1. Physik des Reifens . . . . .	11
2.1.1. Zusammenhang Kraft und Moment . . . . .	11
2.1.2. Kraft, Moment und Geschwindigkeit . . . . .	12
2.1.3. Reibung . . . . .	14
2.1.4. Schlupf . . . . .	15
2.2. Hall-Sensor . . . . .	15
2.3. Servo . . . . .	16
<b>3. Antriebsschlupfregelung</b>	<b>17</b>
3.1. ASR . . . . .	18
3.1.1. Definition . . . . .	18
3.1.2. Einordnung der ASR . . . . .	18
3.1.3. Arbeitsweise . . . . .	21
3.1.4. Steuermöglichkeiten . . . . .	21
3.1.5. ASR-Typen . . . . .	23
3.2. weiterführende Systeme . . . . .	25
3.3. Vorteile und Grenzen der ASR . . . . .	26
<b>4. Analyse</b>	<b>27</b>
4.1. ASR . . . . .	27
4.1.1. $\mu$ - Schlupf - Kennlinie . . . . .	27
4.1.2. Eingriffspunkt und Aktivierung . . . . .	30
4.1.3. benötigte Informationen und Daten . . . . .	32
4.2. verfügbare Hardware im intelliTruck . . . . .	33
4.2.1. Mikrocontroller und I/O-Board . . . . .	34
4.2.2. Sensorik . . . . .	35

---

4.2.3. Aktorik . . . . .	36
4.3. Erfassung von Sensormesswerten und Ermittlung von Kenndaten . . . . .	37
4.3.1. Versuchsaufbau . . . . .	37
4.3.2. Aufbereitung und Auswertung der Messdaten . . . . .	39
4.3.3. Kennwerte . . . . .	41
4.4. Fehlerquellen . . . . .	42
4.5. Fahrsituationen und dafür benötigte Sensoren zur Bereichsermittlung . . . . .	42
4.6. Anforderungen und Beurteilungskriterien . . . . .	44
<b>5. Design</b>	<b>45</b>
5.1. Ablaufdiagramm und Sicherstellung der Bremsfunktionalität . . . . .	45
5.2. Regelung . . . . .	47
5.3. Hardware und Signalerfassung auf dem ARM . . . . .	50
5.4. Software: Systemarchitektur . . . . .	52
5.4.1. Kooperativer Scheduler . . . . .	52
5.4.2. Preemptiver Scheduler . . . . .	53
5.4.3. Hybrider Scheduler . . . . .	53
5.4.4. Entscheidung für den Cooperativen Ansatz mit einem Interrupt . . . . .	53
5.5. Fehlerhandhabung . . . . .	55
5.6. ASR-Zustandsdiagramm und daraus resultierende PWM-Steuersignale . . . . .	56
<b>6. Implementierung</b>	<b>58</b>
6.1. Entwicklungswerkzeug . . . . .	58
6.2. Scheduler . . . . .	59
6.3. Tasks des Schedulers . . . . .	60
6.4. verwendete Datenstrukturen . . . . .	61
6.5. Filterung der Sensormesswerte und Berechnung weiterer Größen . . . . .	62
6.6. Funktionstest der Software . . . . .	63
6.7. Auslastung und Reserven des ARM-Mikrocontrollers . . . . .	64
<b>7. Zusammenfassung</b>	<b>66</b>
<b>Literaturverzeichnis</b>	<b>68</b>
<b>A. Anhang</b>	<b>72</b>
A.1. PicoScope . . . . .	72
A.2. ARM - Einstellungen, PIN-Belegung und Anschlüsse auf dem I/O-Board . . . . .	73
A.3. Diagramme und Codeauszüge . . . . .	75
<b>Abkürzungsverzeichnis</b>	<b>78</b>

# Tabellenverzeichnis

2.1. Reibkoeffizienten . . . . .	14
3.1. einige Beispiele von Steuer- und Regelsystemen im Fahrzeug . . . . .	17
3.2. Sicherheitssysteme entsprechend der Unfallphasen . . . . .	20
3.3. ASR - Reaktionszeit . . . . .	23
4.1. Aktivierung der ASR-Steuerung . . . . .	31
4.2. Kenndaten des intelliTrucks . . . . .	33
4.3. Signale von den Hall-Sensoren . . . . .	35
4.4. Beschleunigungen bei unterschiedlichen Untergründen . . . . .	41
4.5. Kennwerte . . . . .	41
4.6. Fahrsituationen und benötigte Sensoren . . . . .	43
5.1. Zustände nach Linearisierung . . . . .	48
5.2. Einlesemöglichkeiten der eingehenden Signale auf dem ARM-Mikrocontroller . . . . .	51
5.3. Schedulingmethoden . . . . .	53
5.4. PWM-Steuersignal in Abhängigkeit vom Zustand der ASR und vom Fahrerbefehl . . . . .	57
6.1. exemplarische Einstellungen des TimerTicks . . . . .	59
6.2. Tasks des Schedulers . . . . .	60
A.1. wichtige Eigenschaften . . . . .	72
A.2. StartEasy Einstellungen für den ARM . . . . .	73
A.3. I/O-Board Anschlüsse . . . . .	74

# Abbildungsverzeichnis

1.1. intelliTruck . . . . .	10
2.1. Moment - Kraft - Hebelarm . . . . .	11
2.2. Kräfte und Momente . . . . .	12
2.3. Umfangsgeschwindigkeit . . . . .	12
2.4. Seitenführungskräfte . . . . .	13
2.5. Hall-Sensor mit Ausgangssignal . . . . .	16
2.6. PWM-Signal . . . . .	16
3.1. Sicherheitsansatz angelehnt an den VDI Bericht 1471 . . . . .	19
3.2. ASR-Übersicht . . . . .	24
4.1. Kraftschlussbeiwert als Funktion über dem Antriebsschlupf . . . . .	28
4.2. normaler Verlauf der Umfangsgeschwindigkeit (grün); Übergang von der Haft- in die Gleitreibung (rot) . . . . .	29
4.3. Kennlinien . . . . .	29
4.4. Kraftschluss- und Seitenkraftbeiwerte . . . . .	30
4.5. Use-Case-ASR . . . . .	32
4.6. verfügbare Hardwarekomponenten auf dem intelliTruck . . . . .	33
4.7. ARM7 mit I/O-Board . . . . .	34
4.8. Hall-Signal bei ansteigender Geschwindigkeit . . . . .	36
4.9. Versuchsaufbau für Kennzahlenermittlung . . . . .	38
4.10. 100 Prozent Schlupftest . . . . .	40
4.11. Ermittlung der Beschleunigung über Differenzialquotient . . . . .	40
4.12. verschiedene Reibkoeffizienten am Reifen nach Fahrbahnbeschaffenheit . . . . .	43
5.1. Ablauf ASR-System . . . . .	45
5.2. Vernetzung Hardwarekomponenten für die ASR auf dem intelliTruck . . . . .	46
5.3. Linearisierung einer $\mu$ - Schlupf - Kennlinie . . . . .	48
5.4. Zustandsdiagramm der ASR nach Linearisierung . . . . .	49
5.5. Zustandsdiagramm der ASR mit Fehlerzustand . . . . .	56
6.1. Entwicklungsumgebung HiTOP5 . . . . .	58
6.2. enum pwm_states und asr_states . . . . .	61

---

6.3. struct truck_status . . . . .	62
6.4. Codeauszug zur Integerarithmetik . . . . .	62
6.5. Testaufbau . . . . .	63
6.6. Auslastung des ARM-Mikrocontrollers . . . . .	64
A.1. Beschleunigung bei 100 Prozent Schlupf . . . . .	75
A.2. Beschleunigung maximal auf Asphalt . . . . .	75
A.3. Beschleunigung „normal“ auf Asphalt . . . . .	76
A.4. Beschleunigung auf Schotter . . . . .	76
A.5. Beschleunigung auf Schotter 2 . . . . .	76
A.6. Interruptserviceroutine: ISR_Timer0() . . . . .	77
A.7. Funktion: main() . . . . .	77
A.8. Auszug aus der Funktion: init() für SCH_init() . . . . .	77

# 1. Einführung

Ziel dieser Arbeit ist die Entwicklung einer für den intelliTruck (siehe Abbildung 1.1) speziell ausgerichtete Antriebsschlupfregelung (ASR) auf einer ARM-Mikrocontrollerarchitektur.

Neben der Einarbeitung in den ARM7-TDMI LPC2468 von NXP, der Physik des Reifens und der damit einhergehenden Fahreigenschaften eines Fahrzeuges, der Analyse der verfügbaren Hardware und der Funktionsweise einer ASR, beansprucht diese Arbeit ein hohes Maß an Echtzeit, das ein durchdachtes Design erfordert.

Speziell die Verbindung zwischen der realen Welt und der ablaufenden Software im Mikrocontroller macht diese Arbeit interessant. Durch eine Entscheidung der Software, die das Fahrzeug in seiner Bewegung beeinflusst, wird der Kreis zur realen Welt wieder geschlossen. Eine solche Entscheidung kann das Abbremsen eines Rades aber auch das Auslösen eines Airbags sein.

Dieser Kreislauf eröffnet neue Möglichkeiten die Sicherheit von Fahrzeugen und dessen Fahrstabilität zu erhöhen. In der heutigen Automobilindustrie liegt in diesen beiden Aspekten ein Hauptaugenmerk. Das macht sich unter anderem dadurch bemerkbar, dass bei guten Ergebnissen im herstellerunabhängigen Crashtest-Programm: Euro NCAP<sup>1</sup> um Kunden geworben wird. Auch die folgende Auflistung elektronischer Systeme im Fahrzeug zeigt die hohe Relevanz dieser Thematik (vgl. Oberhauser (2003)).

- Systeme für den Antriebsstrang:  
Zündung, Einspritzung, elektronisches Gaspedal, Getriebesteuerung
- Sicherheitssysteme:  
ABS, ASR, ESP, Brake-by-Wire, Steer-by-Wire, Airbag, Gurtstraffer, Überrollbügel, Alarmanlage, Abstandsregelung
- Komfortsysteme:  
Klimaregelung, Sitzverstellung, Einparkhilfe, Fahrwerkregelung, Nachtsichtsystem

---

<sup>1</sup>European New Car Assessment Program: <http://www.euroncap.com/>



Abbildung 1.1.: intelliTruck

Diese Arbeit findet im Rahmen des FAUST-Projektes<sup>2</sup> der Hochschule für Angewandte Wissenschaften Hamburg statt, in dem auf verschiedenen Fahrzeugplattformen autonome und Fahrerassistenzsysteme integriert werden. Mit den entwickelten Systemen wird auch an Wettbewerben wie dem Carolo-Cup<sup>3</sup> teilgenommen. Auf dem Carolo-Fahrzeug wird im Moment eine Umgebungskartierung von Andrej Rull (voraussichtlich 2008) entwickelt.

Autonom fahrende Fahrzeuge, sind Fahrzeuge, die selbständig und unabhängig fahren. In dieser Arbeit wird von einem Fahrer die Rede sein, der dem intelliTruck die Fahrbefehle gibt. Der Fahrer ist als ein Synonym zu betrachten, das für alle möglichen Systeme steht, die Steuerbefehle geben. Als ein System zählt z. B. eine Fahrspurerkennung, wie sie von Nils Kruse (voraussichtlich 2008) für den intelliTruck entwickelt wird. Die Fahrspurerkennung analysiert Aufnahmen einer Kamera und stellt den Lenkwinkel des intelliTrucks entsprechend der Auswertung ein.

---

<sup>2</sup>Fahrerassistenz- und Autonome Systeme: <http://www.informatik.haw-hamburg.de/faust.html>

<sup>3</sup><http://www.carolo-cup.de/>

## 2. Grundlagen

### 2.1. Physik des Reifens

In diesem Abschnitt werden die physikalischen Grundlagen für das Verständnis der ASR beschrieben. Der Reifen ist das Medium, das alle Kräfte zwischen Straße und Fahrzeug überträgt. Deshalb ist es sehr wichtig, dass sich der Reifen stets in einem optimalen Zustand zur Straße befindet.

#### 2.1.1. Zusammenhang Kraft und Moment

In dieser Arbeit wird abwechselnd je nach Zusammenhang von Kräften bzw. von Momenten gesprochen. Vom Motor wirkt z. B. ein Moment auf den Antriebsstrang und überträgt dieses auf die Antriebsräder. Auf der Fahrbahn wirkt dieses Moment als Kraft.

Das Moment und die Kraft stehen wie folgt in Zusammenhang:

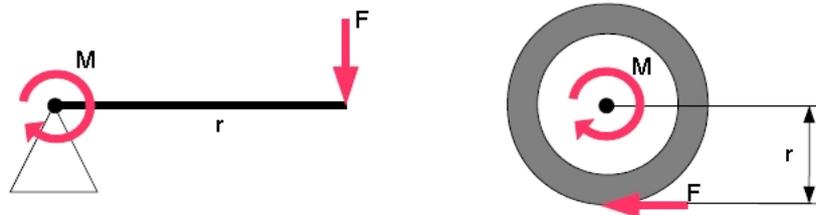


Abbildung 2.1.: Moment - Kraft - Hebelarm

$$M = F * r \quad (2.1)$$

Moment = Kraft \* Hebelarm

### 2.1.2. Kraft, Moment und Geschwindigkeit

In diesem Abschnitt werden die auftretenden Kräfte, Momente und Geschwindigkeiten am Rad beschrieben.

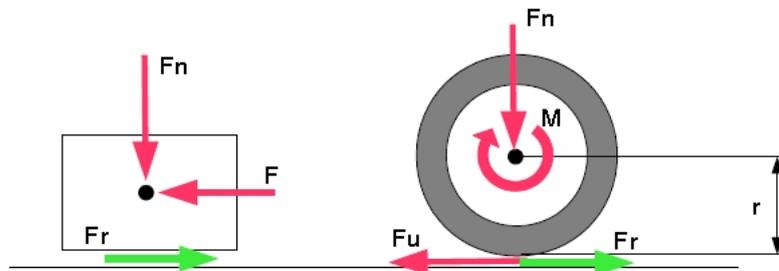


Abbildung 2.2.: Kräfte und Momente

- Die Normalkraft  $F_n$  bezeichnet den rechtwinkligen Anteil der Gewichtskraft zur Kontaktfläche.
- Das Antriebsmoment  $M$  wirkt im Mittelpunkt des Rades.
- Die Umfangskraft auch Tangentialkraft genannt  $F_u$  resultiert aus dem Antriebsmoment  $M$ . Sie wirkt parallel zur Kontaktfläche und entgegengesetzt zur Reibkraft  $F_r$ .
- Die Reibkraft  $F_r$  wirkt parallel zur Kontaktfläche und stets entgegengesetzt zur Umfangskraft  $F_u$  bzw. Kraft  $F$ .

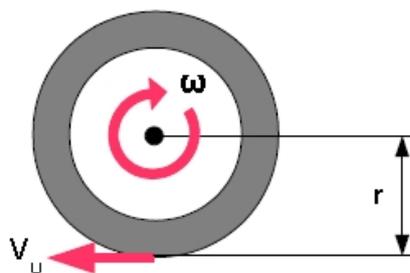


Abbildung 2.3.: Umfangsgeschwindigkeit

- Die Winkelgeschwindigkeit  $\omega$  wird in Umdrehungen pro Sekunde angegeben.
- Die Umfangsgeschwindigkeit  $V_u$  beschreibt die Geschwindigkeit eines Punktes der den Abstand  $r$  zum Mittelpunkt des Reifens hat.

Die Umfangsgeschwindigkeit  $V_U$  berechnet sich wie folgt:

$$V_U = (2 * \pi * r) * \omega \quad (2.2)$$

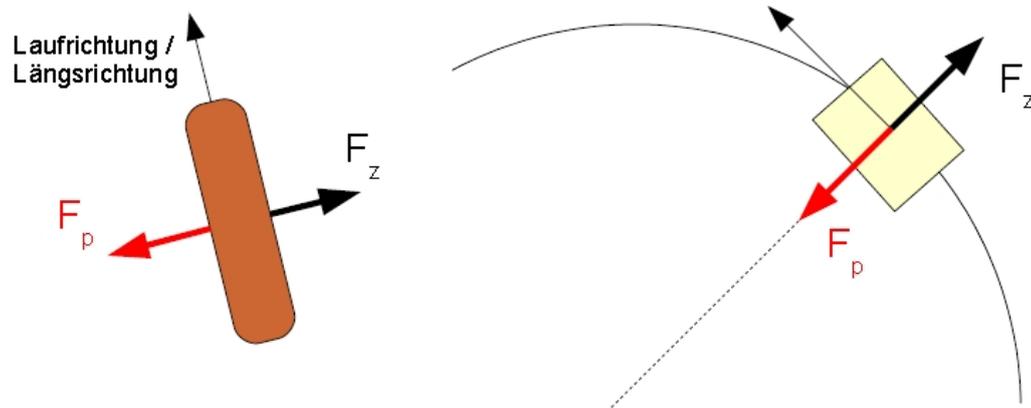


Abbildung 2.4.: Seitenführungskräfte

Die Zentripetalkraft  $F_p$  zeigt zum Kreismittelpunkt und wirkt der Zentrifugalkraft  $F_z$  entgegen. Die Ursache für die Zentrifugalkraft ist nicht immer eine Kurvenfahrt. Wind von der Seite bewirkt auch eine seitliche Kraft auf den Reifen. Wenn der Reifen die seitliche Kraft nicht kompensieren kann, weicht das Fahrzeug von der Fahrtrichtung ab. Deshalb wird die Zentripetalkraft auch Seitenführungskraft genannt.

Auf den Reifen wirken somit eine Kraft in Längsrichtung, die sich aus folgenden Kräften zusammensetzt:

- Antriebskraft
- Bremskraft
- Reibkraft

und eine Kraft in Querrichtung, die sich aus diesen Kräften zusammensetzt:

- Seitenführungskraft
- Zentrifugalkraft
- Reibkraft

### 2.1.3. Reibung

Für die Reibung gilt das Coulombsche Reibungsgesetz:

$$F_r = \mu * F_n \quad (2.3)$$

$$\mu = \frac{F_r}{F_n} \quad (2.4)$$

Das  $\mu$  wird in der Literatur als Reibkoeffizient oder Kraftschlussbeiwert bezeichnet. Es beschreibt das Verhältnis von der Reibkraft zur Normalkraft. Bei der Reibung wird zwischen zwei Arten unterschieden:

- **Haftreibung:  $\mu_0$**   
Reibung eines rollenden Reifens auf der Fahrbahn bzw. Reibung eines stehenden Reifens bei stehendem Fahrzeug
- **Gleitreibung:  $\mu$**   
Reibung bei blockiertem oder durchdrehenden Reifen auf der Fahrbahn

Je nach Reibungsart, Reifenzustand, Geschwindigkeit und Fahrbahnverhältnis kommen verschiedene Reibkoeffizienten zustande. In der folgenden Tabelle werden exemplarisch einige Werte bei neuen Reifen und einer im guten Zustand befindlichen Straße aufgezeigt (vgl. [Bauer, 2002](#), S. 383):

Luftreifen auf Beton- bzw. Teerfahrbahn				
Reibungsart	Geschwindigkeit [km/h]	trocken	nass	vereist
Haftreibung	50	0,85	< 0,65	< 0,1
	90	0,80	< 0,60	
	130	0,75	< 0,55	
Gleitreibung		0,30	0,15	

Tabelle 2.1.: Reibkoeffizienten

Aus der Tabelle ist folgendes abzulesen: Je schlechter die Fahrbahnverhältnisse sind, desto geringer ist der Reibkoeffizient. Ein Autoreifen, der auf Teer gefahren wird, hat einen höheren Reibkoeffizienten als ein Autoreifen auf vereister Strecke.

Bei einer Geschwindigkeit von 50 km/h und trockener Fahrbahn sinkt die Kraftübertragung schlagartig, wenn der Reifen von der Haft- zur Gleitreibung wechselt. Dabei sinkt der Reibkoeffizient von  $\mu_0 = 0,85$  auf  $\mu = 0,30$ , das entspricht einem Einbruch von 65%.

### 2.1.4. Schlupf

Der Schlupf beschreibt das Verhältnis von der theoretisch zurückgelegten Strecke zur tatsächlichen. Beim Anfahren setzt der Reifen einen Teil der Umfangskraft in Wärme und Verformung um, sodass nicht die ganze Energie in die Vorwärtsbewegung übergeht.

Es wird zwischen Antriebsschlupf  $\lambda_A$  und Bremschlupf  $\lambda_B$  unterschieden. Beide Vorgänge verhalten sich prinzipiell gleich und berechnen sich mittels der Fahrzeuggeschwindigkeit  $V_F$  und der Umfangsgeschwindigkeit Rades  $V_U$ .

$$\lambda_A = \frac{(V_U - V_F)}{V_U} \quad (2.5)$$

$$\lambda_B = \frac{(V_F - V_U)}{V_F} \quad (2.6)$$

Der Schlupf wird meist in Prozent angegeben. Ein frei rollendes Rad besitzt die gleiche Geschwindigkeit wie das Fahrzeug und somit ergibt sich ein Schlupf von 0%. Bei einem blockiertem Rad  $V_U = 0$  hingegen liegt der Schlupf bei 100%.

Während der Beschleunigungsphase gilt:

$$V_U > V_F \quad (2.7)$$

und analog für die Verzögerungsphase:

$$V_U < V_F \quad (2.8)$$

## 2.2. Hall-Sensor

Dieser Sensor nutzt den Hall-Effekt und liefert bei jedem metallenen Gegenstand, der registriert wird ein High-Signal. Mit einem Treiberbaustein, der die Pegel für den ARM-Mikrocontroller anpasst, liegt der LOW-Pegel bei 0V und der HIGH-Pegel bei 3,3V. Für weitere Informationen wird an dieser Stelle auf die Arbeiten von Bachir [Blal \(2007\)](#) und Umut [Nar \(voraussichtlich 2008\)](#) verwiesen.

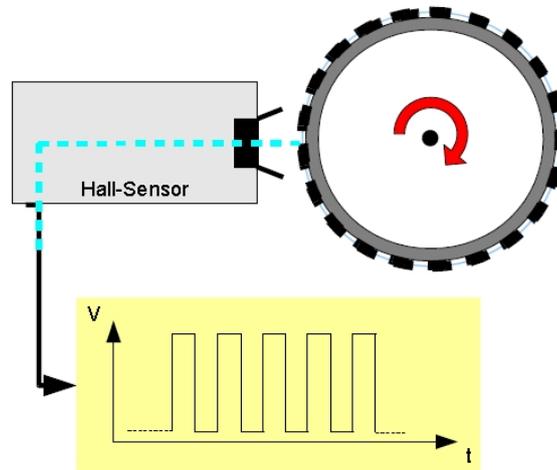


Abbildung 2.5.: Hall-Sensor mit Ausgangssignal

### 2.3. Servo

Ein Servo ist ein Stellmotor, der viel im Modellbau eingesetzt wird. Dieser kann mittels eines so genannten PWM-Signals<sup>1</sup> auf einen gewünschten Stellwinkel gefahren werden.

Ein PWM-Signal sieht wie folgt aus:

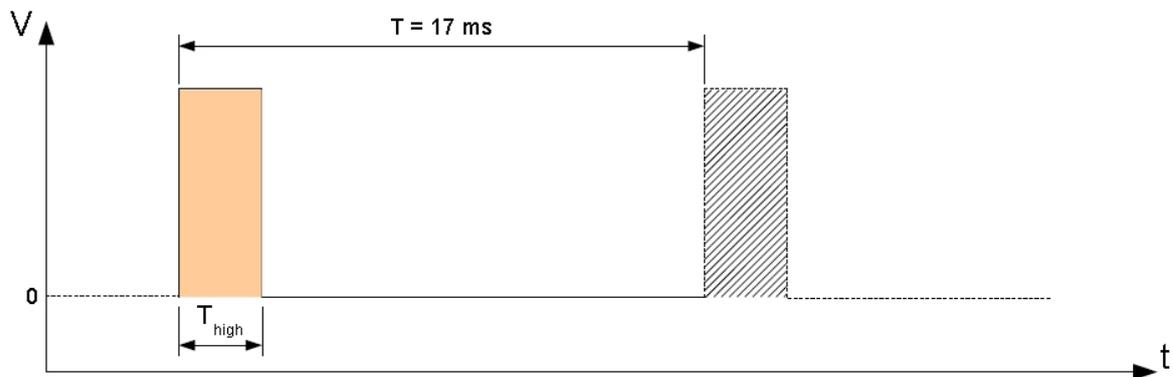


Abbildung 2.6.: PWM-Signal

Die Breite des HIGH-Pulses  $T_{high}$  liegt zwischen einer und zwei Millisekunden. Der kleinste einstellbare Winkel liegt bei  $T_{high} = 1 \text{ ms}$  und der größte bei  $T_{high} = 2 \text{ ms}$ . Weitere Informationen können in der Bachelorarbeit von Enrico [Hensel \(2007\)](#) nachgelesen werden.

<sup>1</sup>Puls Weiten Moduliertes Signal

### 3. Antriebsschlupfregelung

Nach der Einführung, in der das FAUST - Projekt, der intelliTruck und das Ziel der Arbeit beschrieben wird, beschäftigt sich das folgende Kapitel mit elektronischen Systemen (vgl. Übersicht 1) in Fahrzeugen, mit einem Schwerpunkt auf die ASR. Dabei wird ein Bogen von den Anfängen, der Einordnung der ASR über die Theorie bis hin zur Umsetzung und den Regelmöglichkeiten am Fahrzeug gespannt.

Die Entwicklung von Steuer- und Regelsystemen im Fahrzeug nimmt seit den 60er Jahren bis heute mit zunehmender Geschwindigkeit zu. Zwischen zwei bedeutenden Entwicklungen liegen heute nur noch wenige Jahre. Die Tabelle (siehe 3.1) lehnt sich an die Abbildung: „Bedeutende Meilensteine im Automobilbau“ (in Braess und Seiffert, 2003, S. 15)

Jahr ca.	Entwicklung
1959	Transistorzündung
1967	elektronische Benzineinspritzung
1978	Antiblockiersystem
1984	Leerlaufdrehzahlregelung
1988	elektronisches Gaspedal
1990	<b>Antriebsschlupfregelung</b>
1995	elektronisches Stabilitätsprogramm
1997	Bremsassistent und Adaptive Cruise Control
2000	elektro - hydraulische Bremse
ab 2002	PRE-SAFE

Tabelle 3.1.: einige Beispiele von Steuer- und Regelsystemen im Fahrzeug

Die Idee den Schlupf des Fahrzeuges zu regeln, ist schon sehr früh geboren. Das Grundkonzept des heutigen ABS - Systems beschreibt Fritz Oswald (1940) schon in seiner Diplomarbeit. Zu dieser Zeit ist die Technik jedoch für die Erfassung der Raddrehzahl und für schnelle hydraulische Schaltventile nicht weit genug ausgereift, so dass erst 1978 die ersten serienreifen, elektronisch geregelten Bremssysteme vertrieben werden konnten.

Anfang der 90er Jahre gelingt die Regelung des Antriebsschlupfes und wird seitdem unter der Bezeichnung ASR auf den Markt gebracht.

## 3.1. ASR

Im Volksmund wird die ASR meist als „Antischlupfregelung“ bezeichnet und bei spezielleren Veranstaltungen wie der Formel1 wird auch von „Traktionskontrolle“ gesprochen.

Im Englischen wird sie „TCS“<sup>1</sup> genannt.

Alle meinen das Gleiche: „keine durchdrehenden Räder während der Beschleunigungsphase“. Die Abkürzung ASR steht für Antriebsschlupfregelung.

### 3.1.1. Definition

Die Definition nach BOSCH (vgl. [Bauer, 2002](#)):

*„Während das Antiblockiersystem (ABS) das Blockieren der Räder im Bremsfall durch Absenken der Radbremsdrücke verhindert, verhindert ASR im Antriebsfall das Durchdrehen der Räder durch Reduktion des wirksamen Antriebsmoments an jedem einzelnen Antriebsrad.“*

Lehrlingsbuch (vgl. [Gerigk und Bruhn, 1991](#), S.442):

*„Die Antriebs-Schlupf-Regelung (ASR) hat die Aufgabe, das Durchdrehen der Räder zu verhindern und die günstigsten Eingriffsverhältnisse zwischen Rad und Fahrbahn zu erzeugen.“*

Elektronische Bremssysteme (vgl. [Mencher u. a., 2007](#)):

*„Aufgabe einer Traktionsregelung ist es, die angetriebenen Räder am Durchdrehen zu hindern und damit die Fahrstabilität zu verbessern.“*

### 3.1.2. Einordnung der ASR

In der Automobilindustrie wird zwischen aktiven und passiven Systemen im Fahrzeug unterschieden. Die Zuordnung hängt dabei vom thematischen Zusammenhang, wie der Sicherheit und des Antriebes, ab.

---

<sup>1</sup>Traction Control System

### 3.1.2.1. Sicherheitssysteme

Aus der Sicht der Sicherheitssysteme ist die Abgrenzung eher fließend zwischen aktiven und passiven Systemen: Laut dem VDI - Bericht 1471 werden alle „Maßnahmen zur Unfallvermeidung“ in die aktive Sicherheit und „Maßnahmen, die Unfallfolgen für Insassen und andere Verkehrsteilnehmer mildern“ in die passive Sicherheit eingestuft. „Beispielsweise kann die Bremsanlage eines Fahrzeuges, ein klassisches Merkmal der aktiven Sicherheit, auch als Element der passiven Sicherheit verstanden werden, wenn sie durch Precrasherkenkung kurz vor einem Unfall automatisch aktiviert wird und damit die Schwere des folgenden Unfalls und die Folgen für die Insassen mildert“ (vgl. [Justen und Baumann, 1999](#), S.375).

Wie in dem VDI - Bericht 1471 kann die Autofahrt in verschiedene Phasen eingeteilt werden.

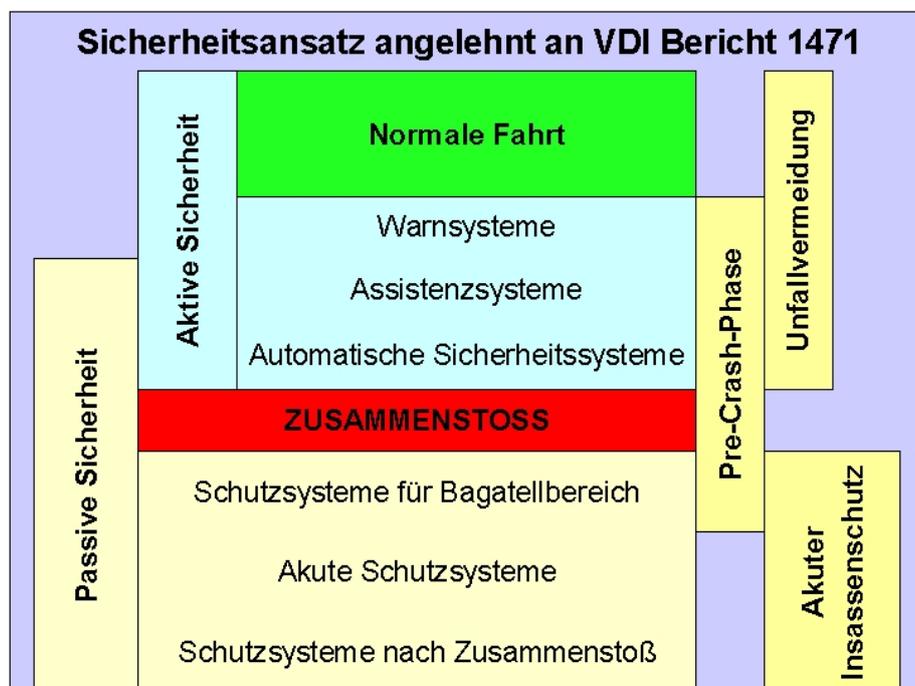


Abbildung 3.1.: Sicherheitsansatz angelehnt an den VDI Bericht 1471

Das Fahrzeug befindet sich solange in der „Normales Fahren“ - Phase, bis das Sicherheitssystem eine Gefahr erkennt.

Ein Unfall zweier Fahrzeuge kann in verschiedene Phasen unterteilt werden. Jede Phase beinhaltet ein bestimmtes oder mehrere Sicherheitssysteme (siehe Tabelle [3.2](#)).

Phase	Sicherheitssystem	Beispiel
Gefahrenphase	Warnsysteme	Spurhaltesystem
Risikophase	Assistenzsysteme	Stabilisierungs- und Bremssystem
Precrashphase	Automatische Sicherheitssysteme	automatische Notbremsung
Eindringphase	Schutzsysteme	Airbag
Postcrashphase	Schutzsysteme	Entriegelung der Türen
Rettungsphase	Rettungsunterstützende Systeme	automatischer Rettungsruf

Tabelle 3.2.: Sicherheitssysteme entsprechend der Unfallphasen

Wie aus der Tabelle 3.2, die sich an die Übersicht (vgl. Reif, 2006, S.240) anlehnt, zu erkennen ist, gehört die ASR zu den aktiven Sicherheitssystemen in einem Kraftfahrzeug. Sie greift aktiv in das Fahrzeug ein, wenn registriert wird, dass die Räder durchdrehen.

### 3.1.2.2. Antriebssysteme

Aus der Sicht der Antriebstechnik wird die ASR zu den passiven Systemen eingeordnet. Sie baut auf die Grundfunktionen des ABS auf und wertet grundsätzlich mit wenigen Ausnahmen die Radumdrehungssignale aus. Mit diesen Werten analysiert das System die Zustände der Räder des Fahrzeuges und greift anschließend gegebenenfalls ein: Beispielsweise durch einen gezielten Bremseneingriff.

Aktive Systeme hingegen beziehen weitere Messgrößen wie Querbewegung, Gierrate und Lenkwinkel mit ein. Mit diesen Eingangsgrößen vergleicht das System den gewünschten Kurs des Fahrers mit dem tatsächlichen des Fahrzeuges. Bei zu großer Differenz der beiden Kurse greift das System gezielt ins Fahrzeug ein, ohne dass der Fahrer aktiv wird:

Das ESP<sup>2</sup> gehört zu den aktiven Systemen und leitet automatisch eine gezielte Bremsung und/oder einen Eingriff ins Motormanagement ein, ohne dass der Fahrer z. B. auf die Bremse treten muss (vgl. Braess und Seiffert, 2003, S.306-307), (vgl. Breuer und Bill, 2004, S. 9).

In dieser Arbeit wird ein Schwerpunkt auf die Physik und die einhergehenden Fahreigenschaften bezüglich des Antriebes gelegt.

<sup>2</sup>elektronisches Stabilitätsprogramm

### 3.1.3. Arbeitsweise

Nach der Einordnung der ASR wird nun die Arbeitsweise erläutert. Die Definitionen aus Abschnitt 3.1.1 kommen der Arbeitsweise schon sehr nahe.

Beim Beschleunigen des Fahrzeuges erhöht sich das Motordrehmoment und damit auch das Antriebsmoment an den Antriebsrädern (siehe Kapitel 2.1.1). Solange die resultierende Umfangskraft aus dem Antriebsmoment kleiner gleich der Reibkraft (siehe Kapitel 2.1.2) ist, kann die Umfangskraft auf die Strasse übertragen werden. Wenn die Umfangskraft die Reibkraft überschreitet, fängt mindestens ein Antriebsrad an durchzudrehen. Dadurch nimmt die Seitenführungskraft ab und das Fahrzeug wird instabil. Dies ist der Punkt an dem die ASR eingreift. Beim Durchdrehen eines oder mehrerer Antriebsräder reduziert sich die übertragbare Umfangskraft deutlich. Bei zu großem Schlupf wird das Fahrzeug instabil. In Sekundenbruchteilen regelt die ASR mit gezielten Eingriffen in die Steuerung den Schlupf der Antriebsräder, um das Fahrzeug zu stabilisieren.

Die ASR - Steuerung teilt sich in zwei Teile auf:

- Analyse und Erfassung der Radumdrehungen
- Steuereingriffe

### 3.1.4. Steuermöglichkeiten

Der optimale Zustand der Reifen zur Fahrbahn hängt von dem Antriebsmoment (siehe Kapitel 2.1.1) an den Antriebsrädern ab. Dies ist aus der Arbeitsweise der ASR zu erkennen. Für jedes Antriebsrad lässt sich eine Momentenbilanz aufstellen, die durch ASR - Eingriffe verändert werden kann, um einen optimalen Schlupf zu gewährleisten.

Momentenbilanz eines Antriebsrades:

$$M_{Antr} = \frac{M_{Mot}}{2} + M_{Br} + M_{Str} \quad (3.1)$$

- **Motormoment**  $M_{Mot}$  (geht von Motor und Getriebe aus)
- **Bremsmoment**  $M_{Br}$  (geht von der Bremse aus)
- **Strassenmoment**  $M_{Str}$  (auf die Strasse übertragbares Moment)
- **Antriebsmoment**  $M_{Antr}$

Dabei gehen  $M_{Br}$  und  $M_{Str}$  negativ in die Bilanz ein.

Aus dieser Bilanz lassen sich die beiden Stellgrößen  $M_{Br}$  und  $M_{Mot}$  für das Antriebsmoment  $M_{Antr}$  ableiten:

Das Bremsmoment  $M_{Br}$  lässt sich an jedem Rad einzeln über den Bremsdruck verändern. Wichtig zu erwähnen ist, dass dafür eine Erweiterung der ursprünglichen ABS - Hydraulik (ASR-Druckversorgung und ABS-Hydroaggregat) notwendig ist. Darauf wird in dieser Arbeit nicht weiter eingegangen. Weiterführende Informationen zu diesem Thema liefern (Breuer und Bill, 2004, S. 77), (Bauer, 1998, S.105ff und 120) und (vgl. Abbildung Bauer, 1998, S.72).

Das Motormoment  $M_{Mot}$  kann je nach verwendetem Motortypen unterschiedlich verändert werden (vgl. Mencher u. a., 2007, S.51):

- Ottomotor
  - Drosselklappe (Drosselklappenverstellung)
  - Zündanlage (Zündwinkelverstellung)
  - Einspritzanlage (Ausblendung einzelner Einspritzimpulse)
- Dieselmotor
  - elektronische Dieselregelung EDC (Reduzierung der Einspritzmenge)

Für die Drosselklappenverstellung wird ein zusätzliches System benötigt:

Um die Motorleistung von der ASR regeln lassen zu können, muss das Gaspedal vom Fahrer zum Motor physisch getrennt werden. Der Bowdenzug wird durch das EMS<sup>3</sup> ersetzt. Sie wird auch „elektronisches Gaspedal“ oder „E-Gas“ genannt. Diese stellt am Motor das gewünschte Signal ein, wobei die ASR priorisiert vor dem Fahrer angelegt wird (vgl. Bauer, 1998, S.72-73).

Die Motormomentabnahme bei der Zündwinkelverstellung ist nur begrenzt einsetzbar, denn der Winkel kann nicht beliebig verstellt werden. Diese Motormomentabnahme reicht bei einem Wechsel von einer griffigen auf einen glatten Fahrbahnuntergrund nicht aus.

Beim Ausblenden einzelner Einspritzimpulse nimmt das Motormoment deutlich ab und bietet eine effektive und schnelle Regelmöglichkeit. „Die Einspritzmenge ist ungefähr proportional dem Motormoment“ (vgl. Reichel, 2001, S.116-117). Es ist auch möglich einzelne Zylinder auszuschalten, um das Motormoment zu reduzieren, während die anderen normal weiter gezündet werden.

---

<sup>3</sup>elektronische Motorleistungssteuerung

Vergleich der ASR-Eingriffsmöglichkeiten bezogen auf die Reaktionszeit, [Reichel \(2001\)](#):

ASR - Reaktionszeit	ASR mit
kurz	Drosselklappen- und Radbremseneingriff
kurz	Drosselklappen- und Zündungseingriff
lang	Drosselklappeneingriff

Tabelle 3.3.: ASR - Reaktionszeit

„Die Darstellung läßt erkennen, daß eine ausschließliche Steuerung des Antriebsmoments mit der Drosselklappe bei Fahrzeugen mit zwei Antriebsrädern wegen der relativ langsamen Reaktionszeit nicht zu einem befriedigenden Ergebnis führt“ (vgl. [Bauer, 1998](#), S.74). Die Verzögerung kommt dadurch zustande, dass das Kraftstoffgemisch zwischen Drosselklappe und Zylinder erst aufgebraucht werden muss, bevor das Drehmoment abnimmt (vgl. [Reichel, 2001](#), S.116).

### 3.1.5. ASR-Typen

Wie aus der Tabelle 3.3 abzulesen ist, werden die unterschiedlichen Eingriffsmöglichkeiten in das Fahrzeug auch kombiniert eingesetzt. Je nach Geschwindigkeit des Fahrzeuges und der Art des Eingriffes in die Steuerung sind die resultierenden Wirkungen auf das Verhalten des Fahrzeuges unterschiedlich.

#### 3.1.5.1. B-ASR

B-ASR steht für Brems-Antriebsschlupfregelung. Für nicht konstant griffige Fahrbahnen auch „ $\mu$  – Split“ genannt erfährt das Rad, das auf dem weniger griffigen Teil der Fahrbahn steht, wegen des Differenzials ein erhöhtes Antriebsmoment. Das B-ASR erkennt diesen Fall und bremst dieses Rad ab, um das Antriebsmoment auf der gegenüberliegenden Seite zu erhöhen. Diese Vorgehensweise ist für den Anfahrtsbereich, also niedrige Geschwindigkeiten, gedacht. Solche Bremsengriffe können nicht ständig genutzt werden, weil die Bremsanlage sonst überhitzt. Zum Schutz der Bremse wird eine Elektronik eingesetzt, die eine solche Überhitzung verhindern soll und die maximale Bremsdauer kontrolliert (vgl. [Braess und Seifert, 2003](#), S.513).

Die Wirkung ist mit der Differenzialsperre vergleichbar und funktioniert nur bei niedrigen Geschwindigkeiten (bis etwa 40 km/h). Heute wird diese Lösung als elektronische Differenzialsperre (EDS) bezeichnet.

### 3.1.5.2. M-ASR

Um die Belastung der Bremse zu verringern, regelt die M-ASR das Motordrehmoment, so dass kaum noch das Eingreifen der Bremse nötig wird. So kann sichergestellt werden, dass die Bremse und der Motor nicht gegeneinander arbeiten. Ab einer Geschwindigkeit von 40 km/h greift die B-ASR kaum noch ein und die M-ASR übernimmt die Regelung, da sie für den ganzen Geschwindigkeitsbereich des Fahrzeuges geeignet ist, um das Fahrverhalten zu stabilisieren.

### 3.1.5.3. MSR

Die ASR - Steuerung wird mit der MSR<sup>4</sup> ergänzt, um die Fahrstabilität zu erhöhen. Sie greift für den Fall, dass zu schnell runtergeschaltet oder das Gas abrupt weggenommen wird, ein. Die Antriebsräder weisen zu diesem Zeitpunkt einen zu hohen Bremschlupf durch den Motor auf. Es kann dazu führen, dass bei heckgetriebenen Fahrzeugen das Heck ausbricht. Durch kurzes Gasgeben nimmt die Geschwindigkeit der Antriebsräder langsamer ab und die Fahrstabilität bleibt stets erhalten (vgl. [Bauer, 1998, S.72](#)).

Die Abbildung 3.2 zeigt eine Übersicht der ASR.

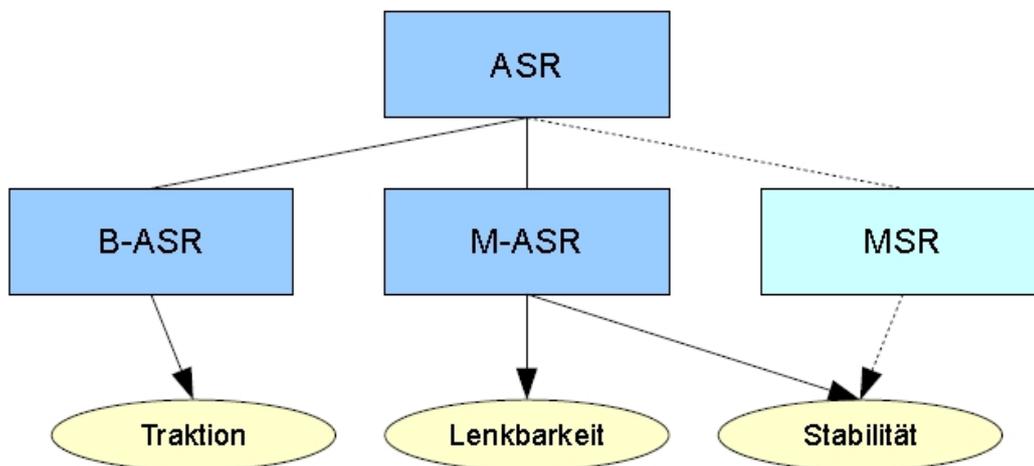


Abbildung 3.2.: ASR-Übersicht

<sup>4</sup>Motorschleppmomentregelung

## 3.2. weiterführende Systeme

1995 erhält das ESP <sup>5</sup> als große Neuerung Einzug in den Sicherheitsbereich. Dabei werden vier Systeme kombiniert. Sie setzt sich aus der ABS, ASR, EBV <sup>6</sup> und GMR <sup>7</sup> zusammen. Das EBV ist ein zusätzlicher ABS-Software-Algorithmus, der die Bremskraftverteilung optimiert, um die Fahrstabilität und einen guten Kraftschlusswert beizubehalten. Die GMR stabilisiert das Fahrzeug in Quer- und Längsrichtung.

1996 erscheint das BAS <sup>8</sup> auf dem Markt, das bei einer Vollbremsung die maximale Bremskraft aufbaut, wenn der Fahrer zu langsam oder zu sanft auf das Bremspedal tritt. Die Weiterentwicklung dazu ist das ACC <sup>9</sup>, das eine konstante Wunschgeschwindigkeit des Fahrers hält. Wenn das Fahrzeug ein vorausfahrendes erreicht, drosselt es die Geschwindigkeit, hält einen konstanten Sicherheitsabstand und fordert gleichzeitig über Signale den Fahrer dazu auf einzugreifen. Ein Eingreifen des Fahrers ist nicht zwingend notwendig. Zum Beispiel nach einem Wechsel auf eine freie Spur gibt das ACC automatisch Gas bis die Wunschgeschwindigkeit wieder erreicht ist.

Seit 2002 widmet sich die Autoindustrie dem / der sogenannten PRE-SAFE / PRE-CRASH Detection <sup>10</sup>. Die Idee dahinter ist die frühzeitige Erkennung von Gefahren und drohenden Unfällen (siehe Kapitel 3.1.2.1). So sollen in Zukunft Unfälle vermieden und die Schwere von Unfällen gesenkt werden. Als ein aktuelles Beispiel sei der „Bremsassistent PRO“ von VOLVO erwähnt. Bei einem drohenden Auffahrunfall wird der Fahrer durch optische und akustische Signale gewarnt. Wenn eine Reaktion des Fahrers ausbleibt und ein Unfall unvermeidbar ist, wird das Fahrzeug ohne Eingriff des Fahrers abgebremst (Heise (2008)).

---

<sup>5</sup>elektronisches Stabilitätsprogramm

<sup>6</sup>elektronische Bremskraftverteilung

<sup>7</sup>Giermomentenregelung

<sup>8</sup>Bremsassistent

<sup>9</sup>Adaptive Cruise Control

<sup>10</sup>Oberbegriff für vorzeitiges Erkennen einer Gefahrensituation

### 3.3. Vorteile und Grenzen der ASR

Zusammenfassend bietet die ASR folgende Vorteile:

- verbessertes Gripverhalten / Erhöhung der Vortriebskräfte
- geringerer Reifenverschleiß
- Sicherung der Fahrstabilität bei heckgetriebenen Fahrzeugen bzw. Lenkbarkeit bei frontgetriebenen Fahrzeugen
- Sperrdifferenzialfunktion
- Warnung vor Erreichen der physikalischen Stabilitätsgrenze bzw. Ausfall der Steuerung durch eine Informationsleuchte z. B. bei Glätte

Wichtig dabei ist, dass die ASR ausschließlich den Längsschlupf des Fahrzeuges regeln kann. Um Querschleunigungen z. B. im physikalisch optimalen Bereich zu halten, sind andere Systeme notwendig, die zusätzliche Sensormesswerte zu den Radumdrehungssignalen berücksichtigen (siehe Kapitel [3.1.2.2](#)).

## 4. Analyse

Das ASR-System wird in den intelliTruck (siehe Kapitel 1.1) eingebaut. Im folgenden Kapitel wird das Fahrzeug unter verschiedenen Gesichtspunkten untersucht. Dazu gehören z. B. die Fahreigenschaften auf unterschiedlichen Fahrbahnbeschaffenheiten, der Aufbau des Fahrzeuges sowie die vorhandene Sensorik und Aktorik. Die ermittelten Erkenntnisse aus diesem Kapitel bieten anschließend eine Grundlage für die folgende Designentscheidung.

### 4.1. ASR

In diesem Abschnitt werden die Voraussetzungen ermittelt, um die aufgeführten Vorteile (siehe Kapitel 3.3) der ASR zu erzielen bzw. zu ermöglichen.

#### 4.1.1. $\mu$ - Schlupf - Kennlinie

Im Grundlagenkapitel wird der Schlupf (siehe Kapitel 2.1.4) und die Reibung (siehe Kapitel 2.1.3) als wichtige Kenngrößen für den Reifen eingeführt. Um das Beschleunigungs- bzw. Bremsverhalten des Fahrzeuges beschreiben zu können, müssen diese beiden Größen in Zusammenhang gebracht werden. Dazu wird eine Graphik benutzt, die den Kraftschlussbeiwert  $\mu$  (siehe Kapitel 2.1.3) über den Antriebsschlupf  $\lambda$  aufträgt. Der daraus entstehende Graph wird  $\mu$  - Schlupf - Kennlinie  $\mu$  - Schlupf - Kennlinie genannt. Jede dieser Kennlinien ist vom eingesetzten Reifen, der Radlast, des Reifendrucks und der Fahrbahnoberfläche abhängig. Laut Hans-Rolf (Reichel, 2001, S.32) können aus dem Graphen allgemeine Informationen abgeleitet werden. Weitere Informationen wie z. B. die Bestimmung einer solchen Kennlinie, können im selben Buch auf Seite 32 nachgelesen werden.

In Abbildung 4.1 ist ein solcher Graph dargestellt. Damit ist die Reibkraft abhängig von der Funktion des Kraftschlussbeiwertes über dem Schlupf:

$$F_r = \mu(\lambda) * F_n \quad (4.1)$$

Anfangs steigt die Kennlinie steil bis zum Maximum (Punkt A) an. Dieser Bereich wird als „stabil“ bezeichnet, da die Umfangskraft (siehe Kapitel 2.1.2) stets gleich der Reibkraft ist:

$$F_u = F_r = \mu(\lambda) * F_n \quad (4.2)$$

Am Maximum, auch kritischer Bereich genannt, der zwischen 8 und 20% liegt, ist die Kraftübertragung des Antriebsmomentes auf die Strasse maximal. Hier ist die Umfangskraft gleich der maximalen Reibkraft:

$$F_u = F_{r_{max\ haft}} = \mu_0 * F_n \quad (4.3)$$

Ab dem Maximum beginnt der „instabile“ Bereich. Dabei wird die Umfangskraft größer als die Reibkraft, was dazu führt, dass der Reifen von der Haftreibung in die Gleitreibung übergeht. Dieser Übergang verläuft nahtlos. Die übertragbare Umfangskraft auf die Fahrbahn nimmt mit steigendem Schlupf ab (Punkt B). Für die maximale Reibkraft bei Gleitreibung gilt:

$$F_{r_{max\ gleit}} = \mu * F_n \quad (4.4)$$

Für die Reibkoeffizienten gilt:

$$\mu_0 > \mu \quad (4.5)$$

Daraus folgt:

$$F_{r_{haft}} > F_{r_{reib}} \quad (4.6)$$

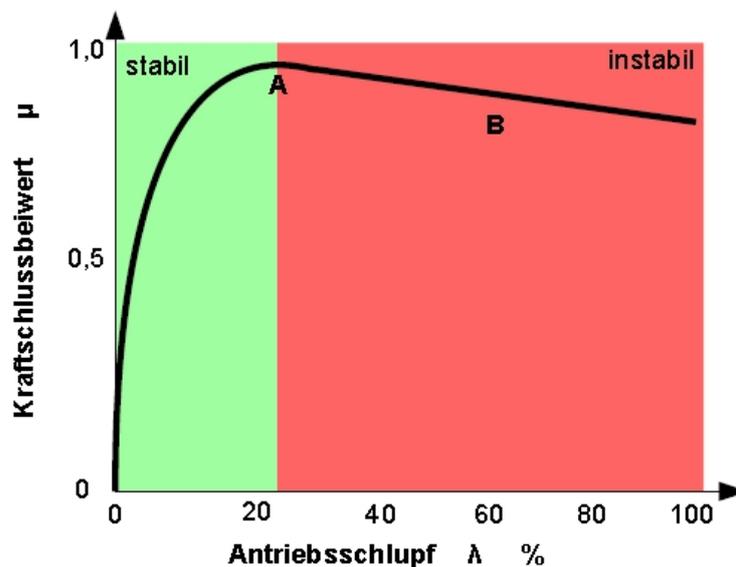


Abbildung 4.1.: Kraftschlussbeiwert als Funktion über dem Antriebsschlupf

Die Umfangskraft wird im „stabilen“ Bereich von der Reibkraft kompensiert. Nach dem Über-

gang in den „instabilen“ Bereich sinkt die Reibkraft ab, so dass die überschüssige Umfangskraft, die nicht auf die Fahrbahn übertragen werden kann, die Umfangsgeschwindigkeit schlagartig erhöht (vgl. Abbildung 4.2). Dies wird im folgenden Graphen am Verlauf der Umfangsgeschwindigkeit  $V$  eines Reifens über die Zeit  $t$  deutlich. Im Zeitpunkt  $t_0$  liegt das Maximum der  $\mu$  - Schlupf - Kennlinie (Übergang von der Haft- zur Gleitreibung).

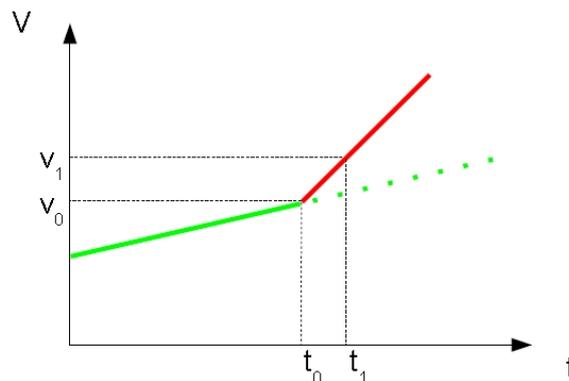


Abbildung 4.2.: normaler Verlauf der Umfangsgeschwindigkeit (grün); Übergang von der Haft- in die Gleitreibung (rot)

Je nach Fahrbahnbeschaffenheit kommen charakteristische Kennlinien (siehe Abbildung 4.3) zustande (**a** Idealfall **b** Asphalt **c** Neuschnee **d** Eis).

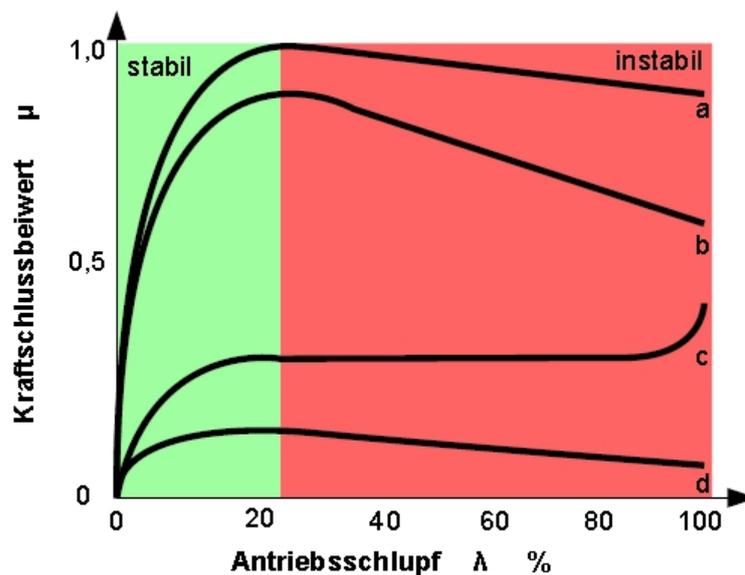


Abbildung 4.3.: Kennlinien

Aus dieser Graphik kann abgelesen werden, dass bei allen vier Graphen mit Ausnahme

des Neuschnees (siehe Abbildung 4.3 Graph c) der Verlauf in Steigung und Maximum sich ähnlich verhält. Bei schlechter werdenden Fahrbahnverhältnissen fällt die Kraftübertragung geringer aus. Dies ist an der Lage der Maxima zu erkennen, die mit dem Graphen eins bis vier abnimmt.

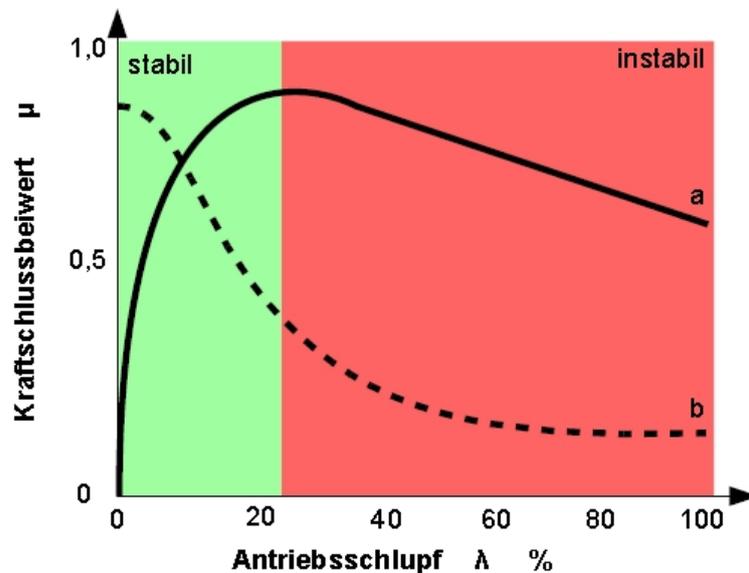


Abbildung 4.4.: Kraftschluss- und Seitenkraftbeiwerte

Der „stabile“ Bereich zeichnet sich weiter dadurch aus, dass die Seitenführungskraft hoch ist (siehe Abbildung 4.4). Diese nimmt mit steigendem Schlupf stark ab, was z. B. bei heckgetriebenen Fahrzeugen zum Ausbrechen des Hecks führen kann.

#### 4.1.2. Eingriffspunkt und Aktivierung

Nach der Beschreibung der Graphen werden nun die Ziele und Vorteile der ASR aufgeführt. Diese werden dahin untersucht, in welchem Bereich sie auftreten und wann die ASR aktiviert werden muss.

1. Die Fahrstabilität bei heckgetriebenen Fahrzeugen und die Lenkbarkeit bei frontgetriebenen Fahrzeugen hängen hauptsächlich von den Seitenführungskräften auf den angetriebenen Rädern ab. Hohe Seitenführungskräfte bewirken, dass das Fahrzeug bestmöglich in der Spur gehalten wird. Wie aus dem Graphen (siehe Abbildung 4.4) zu erkennen ist, nehmen die Seitenführungskräfte mit zunehmenden Schlupf ab. Folglich sind sie bei geringem Schlupf am größten und diese wirken im „stabilen“ Bereich der  $\mu$  - Schlupf - Kennlinie.

2. Für die Erhöhung der Vortriebskräfte muss ermittelt werden, wann die Kraftübertragung am höchsten ist. Dieser Punkt ist in den Graphen leicht zu erkennen. Die Maxima (vgl. Abbildung 4.1 Punkt A und Abbildung 4.3) in den  $\mu$  - Schlupf - Kennlinien sind die Punkte, an denen die Kraftübertragung am größten sind (siehe Kapitel 4.3). Wenn sich das Fahrzeug im entsprechenden Schlupfbereich für das Maximum befindet, wird auch das Gripverhalten verbessert.
3. Im „stabilen“ Bereich der  $\mu$  - Schlupf - Kennlinie ist der Verschleiß der Reifen am geringsten.
4. Für die Sperrdifferenzialfunktion und für das Verhindern von durchdrehenden Reifen ist der Übergang vom Maximum in den „instabilen“ Bereich von Bedeutung. Bei solch einem Übergang nimmt der Kraftschlussbeiwert ab, da der Reifen in die Gleitreibung übergeht, während der Schlupf zum Punkt B (siehe Abbildung 4.1 und Abbildung 4.3) hin zunimmt, so dass es zu durchdrehenden Rädern kommt.

Die Tabelle 4.1 zeigt, welche Bereiche durch die ASR-Steuerung angestrebt ( $\odot$ ) werden und wann diese aktiviert ( $\oplus$ ) wird.

angestrebter Bereich ( $\odot$ ), Bereich aktiviert ( $\oplus$ ) Regelung			
Ziel	„stabiler“ Bereich	Maximum	„instabiler“ Bereich
Fahrstabilität und Lenkbarkeit erhöhen	$\odot$	$\odot$	$\oplus$
Erhöhung der Vortriebskräfte und Gripverhalten	$\oplus$	$\odot$	$\oplus$
Reifenverschleiß	$\odot$	$\odot$	$\oplus$
Sperrdifferenzialfunktion Verhindern durchdrehenden Räder	$\odot$	$\odot$	$\oplus$

Tabelle 4.1.: Aktivierung der ASR-Steuerung

Hieraus ist abzulesen, dass die ASR immer dann aktiv wird, wenn das Fahrzeug in den „instabilen“ Bereich übergeht bzw. sich dort befindet. Die Regelung zielt in erster Linie darauf ab das Fahrzeug stets im „stabilen“ Bereich zu halten. Optimal wäre es das Fahrzeug am Maximum zu „halten“, denn hier sind zusätzlich zu den „stabilen“ Eigenschaften des Fahrzeuges die Vortriebskräfte am größten.

### 4.1.3. benötigte Informationen und Daten

Zur Aktivierung der ASR muss, wie oben beschrieben, der Bereich bestimmt werden, in dem sich das Fahrzeug aktuell befindet. Welche Informationen und Daten dafür nötig sind, werden jetzt ermittelt:

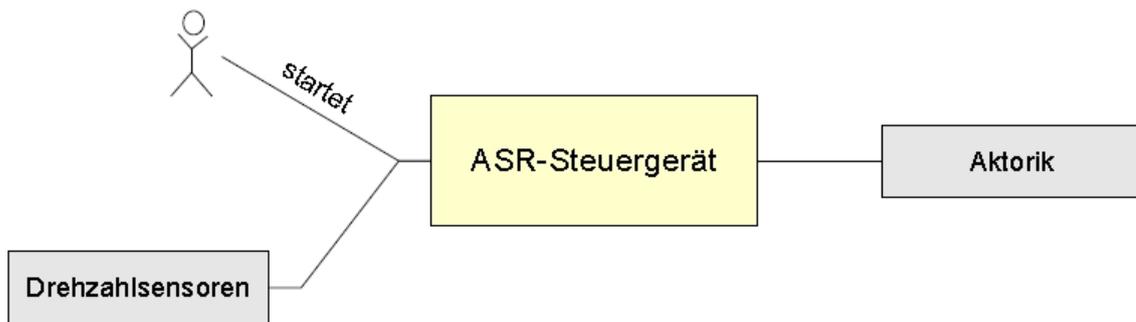


Abbildung 4.5.: Use-Case-ASR

Die Eingangsgrößen für die ASR-Steuerung sind je nach ASR-Ausführung unterschiedlich. Das gleiche gilt für die Ausgangsgrößen. Für weitere Informationen zu den ASR-Ausführungen bietet das Buch von (Bauer, 1998, S.72ff) eine gute Übersicht. Um den Bereich der  $\mu$  - Schlupf - Kennlinie, in dem sich das Fahrzeug befindet, zu bestimmen, reichen die Umfangsgeschwindigkeiten und deren zeitliche Änderung aus. Bei aufwändigeren Systemen besteht ein bidirektionaler Austausch zwischen der ASR-Steuerung und anderen Systemen (vgl. Bauer, 1998). Dadurch kann die ASR-Steuerung nicht nur Stellbefehle geben, sondern auch weitere Daten wie z.B. die Motordrehzahl in ihren Berechnungen mit einbeziehen.

## 4.2. verfügbare Hardware im intelliTruck

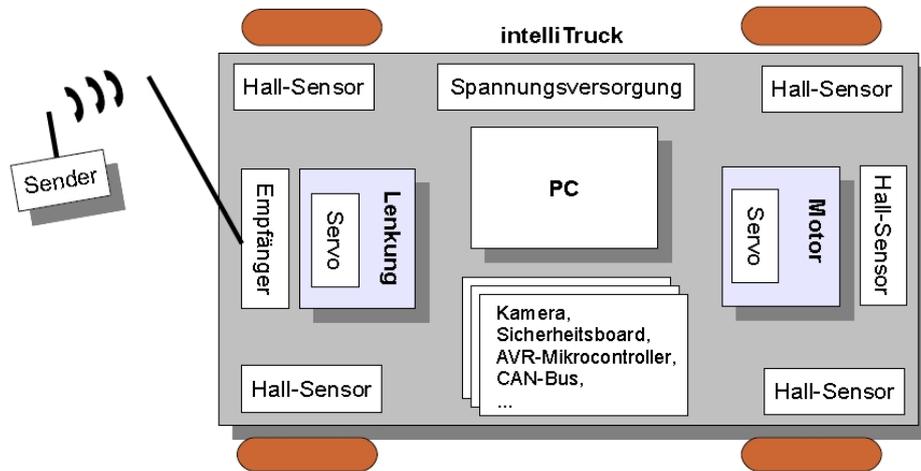


Abbildung 4.6.: verfügbare Hardwarekomponenten auf dem intelliTruck

In diesem Bild (siehe Abbildung 4.6) sind die einzelnen Hardwarekomponenten schematisch aufgeführt. Die entwickelten Komponenten aus früheren Projekten und Arbeiten sind in einem Block zusammengefasst.

Im Folgenden werden Kenndaten des intelliTrucks, die für die ASR relevant sind, aufgeführt:

Höchstgeschwindigkeit $V_{max}$	65 km/h
Antriebsart	Heckantrieb mit Differenzial
Reifenradius	90 mm
Inkrementgeber (Räder)	24 Zähne
Inkrementgeber (Getrieberad)	48 Zähne

Tabelle 4.2.: Kenndaten des intelliTrucks

### 4.2.1. Mikrocontroller und I/O-Board

In dieser Arbeit wird mit einem 32 Bit ARM7-TDMI LPC2468 Mikrocontroller gearbeitet. Dieser Mikrocontroller ist von NXP <sup>1</sup>, ehemals Philips, entwickelt worden. Er zeichnet sich durch folgende Eckdaten aus (vgl. [NXP, 2007](#)):

- 32 Bit
- bis zu 72 MHz Taktfrequenz
- einfaches Bespielen und Debuggen über die USB-Schnittstelle
- kleine kompakte Bauweise
- Temperatureinsatzbereich: -40 bis +85 °C
- über die Schleifkontaktleiste kann ein I/O-Board angeschlossen werden

Der ARM-Mikrocontroller stellt den Zugriff ausgewählter PINs über die Schleifkontaktleiste zur Verfügung, die über das I/O-Board angesprochen bzw. verbunden werden können. Durch diese Auswahl lassen sich nicht alle Möglichkeiten des ARM-Mikrocontrollers ausnutzen, was beim späteren Design berücksichtigt werden muss. Als Beispiel sei erwähnt: Jeder der vier Timer besitzt je vier Captureregister. Der ARM-Mikrocontroller besitzt folglich 16 Captureregister, wovon nur *vier* über das I/O-Board ansprechbar sind.



Abbildung 4.7.: ARM7 mit I/O-Board

<sup>1</sup>[www.nxp.com](http://www.nxp.com)

Der ARM-Mikrocontroller besitzt gegenüber dem AVR von Atmel<sup>2</sup> (AT90CAN128), mit dem die Mikrocontrollerprogrammierung während des Studiums gelehrt wurde, eine andere Architektur und andere Spezifikationen. Er bietet mehr Funktionen und Betriebsmodi an. Um diese nutzen zu können, ist ein Teil der Bearbeitungszeit dieser Arbeit zur Einarbeitung genutzt worden. Für spätere Arbeiten wird ein anderes I/O-Board genutzt, das zur Zeit in eigener Entwicklung ist. Als besondere Neuerung ist die Integration von drei LPC-Sticks auf dem neuen I/O-Board.

### 4.2.2. Sensorik

Für die Ermittlung der notwendigen Umfangsgeschwindigkeit werden Sensoren an den Rädern benötigt. Die Sensorwahl und die Umsetzung auf dem intelliTruck sind in der Diplomarbeit von Bachir [Blal \(2007\)](#) beschrieben. Die Umsetzung gibt vier Signale pro Umdrehung des Rades wieder. Diese Auflösung ist relativ grob, da das Fahrzeug zwischen zwei Signalen 14 cm weit rollt. Aufbauend auf diese Lösung verbessert Umut [Nar \(voraussichtlich 2008\)](#) in seiner Diplomarbeit die Auflösung von vier auf 24 Signale pro Umdrehung, indem ein Felgenreifen mit integriertem Zahnkranz (Inkrementgeber) montiert wird. Nun bewegt sich der intelliTruck 2,4 cm pro Signal und bietet eine bessere Auflösung, die für die ASR genutzt wird.

Insgesamt sind fünf Hall-Sensoren (siehe Kapitel [2.2](#)) auf dem intelliTruck (vgl. [Abbildung 4.6](#)) verbaut. Die folgende [Tabelle 4.3](#) und die [Abbildung 4.8](#) geben eine Übersicht über das eingehende Signal der Hall-Sensoren bezogen auf die Eigenschaften des intelliTrucks (vgl. [Tabelle 4.2](#)).

Hall-Sensor	Signale pro Umdrehung	Frequenz bei $V_{max}$	$\Delta t$ zwischen zwei Signalen bei $V_{max}$
Räder	24	766 Hz	1,31 ms
Getrieberad	48	1533 Hz	0,65 ms

Tabelle 4.3.: Signale von den Hall-Sensoren

Anhand der Hall-Signale ermittelt das ASR-System die Geschwindigkeiten der einzelnen Räder und entscheidet anhand dessen, ob ein Eingreifen in das Fahrzeug nötig ist. Zur Schlupfbestimmung (siehe Kapitel [2.1.4](#)) wird die tatsächliche Geschwindigkeit des Fahrzeuges  $V_F$  und die der Antriebsräder  $V_U$  benötigt: Für die Bestimmung von  $V_U$  können sowohl die Hinterräder als auch das Getrieberad des intelliTrucks genutzt werden. Die Vorderräder werden

<sup>2</sup>[www.atmel.com](http://www.atmel.com)

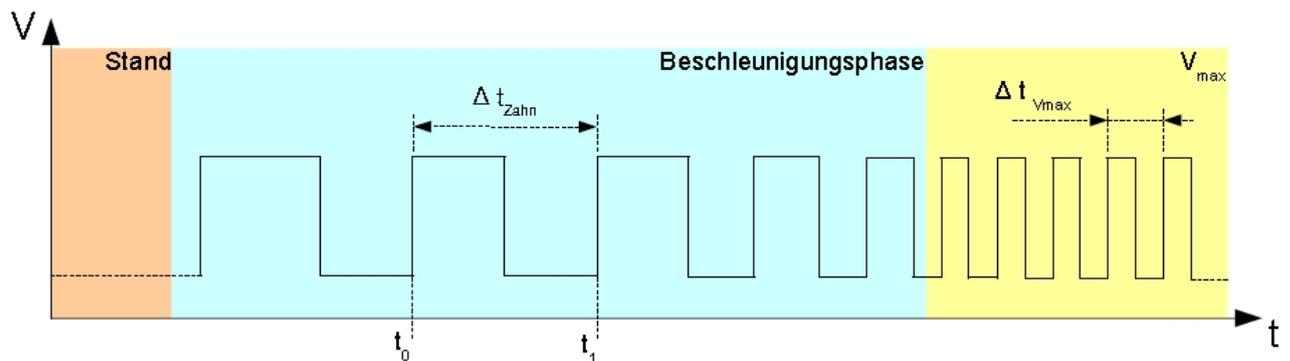


Abbildung 4.8.: Hall-Signal bei ansteigender Geschwindigkeit

nicht angetrieben und bewegen sich folglich in der tatsächlichen Geschwindigkeit  $V_F$  des intelliTrucks. Damit sind beide Geschwindigkeiten bestimmt und der Schlupf kann ermittelt werden.

### 4.2.3. Aktorik

Die Sensorik liefert alle nötigen Daten zur Schlupfbestimmung. Anschließend greift das ASR-System gegebenenfalls ein, dies geschieht über die Aktorik.

Zur Geschwindigkeitsregelung wird im intelliTruck ein Servo genutzt, der gleichzeitig mit der Drosselklappe des Motors und mit der Bremse verbunden ist. In der Grundstellung steht das Fahrzeug in einem Ruhezustand, indem die Bremse nicht betätigt und kein Gas gegeben wird. Zur einen Drehrichtung des Servos wird die Drosselklappe verstellt und der intelliTruck beschleunigt. Zur anderen setzt die Beschleunigung aus und die Bremse am Getriebe bremst gleichzeitig beide Antriebsräder ab. Wie in Kapitel 3.1.4 beschrieben wird, hängt das Antriebsmoment von der Momentenbilanz (siehe Gleichung 3.1) ab, das durch das Bremsmoment und Motormoment verändert werden kann.

Am intelliTruck ist ein kurzzeitiges und gezieltes Bremsen *eines* Antriebsrades wegen seiner Bauart nicht möglich. *Beide* Antriebsräder werden gleichzeitig durch die Getriebeklemmung, die vor dem Differenzial sitzt, abgebremst. Daraus lässt sich ableiten, dass das ASR-System nur über die Drosselklappenverstellung die Momentenbilanz im Sinne der ASR-Funktion beeinflussen kann (vgl. Kapitel 3.1.5.1). Die Drosselklappe ist somit die einzige Stellgröße.

### 4.3. Erfassung von Sensormesswerten und Ermittlung von Kenndaten

Für die Designentscheidung und den späteren Plausibilitätstest, ob die eingehenden Sensormesswerte richtig sind, werden einige Untersuchungen gemacht.

Die benutzte Beschreibung der Geschwindigkeit der Räder ist bis jetzt nicht präzise genug, denn damit kann sowohl die Umfangs- als auch die Winkelgeschwindigkeit gemeint sein. Im Folgenden wird von der Umfangsgeschwindigkeit (siehe Abbildung 2.3) eines Rades gesprochen.

Aus der Abbildung (vgl. Abbildung 4.2 im Analysekapitel) ist zu erkennen, dass die Umfangsgeschwindigkeit beim Durchdrehen der Räder schlagartig ansteigt und für die ASR-Steuerung eine wichtige Größe bildet. Um die Höhe des schlagartigen Anstieges herauszufinden, wird die maximale Beschleunigung des intelliTrucks durch einen 100% - Schlupf - Test bestimmt.

Zusätzlich werden die maximal möglichen Beschleunigungen auf unterschiedlichen Fahrbahnobergründen (z. B. Asphalt und Schotter) empirisch ermittelt. Mit der Information, auf welcher Fahrbahnoberfläche gerade gefahren wird, können besser abgestimmte Stellgrößen von der ASR-Steuerung ausgegeben werden.

#### 4.3.1. Versuchsaufbau

Für die Versuche wird ein PicoScope (siehe Anhang A.1) (PC-Oszilloskop) verwendet, das auf dem intelliTruck untergebracht und an den dort verfügbaren PC angeschlossen wird (siehe Abbildung 1.1). Über ein Remote Desktop (VNC<sup>3</sup>) wird der PC von einem Notebook per W-LAN „ferngesteuert“. Ein Bildschirm wird auf dem intelliTruck somit nicht benötigt. Zusätzlich besitzt es die Möglichkeit während der Fahrt Testdaten aufzunehmen. Der Hall-Sensor am Getriebe wird mit dem Messfühler des PicoScopes verbunden. Die anderen Hall-Sensoren sind aufgrund von Umbauarbeiten am intelliTruck für die Tests nicht verfügbar.

Bei ausgeschaltetem Motor im Labor liefern die Hall-Sensoren bei Drehen der Räder gute Hall-Signale auf dem PicoScope. Erste Tests bei laufendem Motor weisen hingegen durch die Zündkerze starke Störungen auf. Diese können mittels eines abgeschirmten Kabels unterbunden werden.

Zur Bestimmung der maximalen Beschleunigung wird das Hall-Signal während der Beschleunigungsphase des intelliTrucks aufgenommen. Dazu bietet das PicoScope einen sogenannten Log-Modus an, der in festgelegten Zeitabständen den Pegel des Hall-Signals

---

<sup>3</sup>[www.realvnc.com](http://www.realvnc.com)

ausliest und diesen in einer Datei abspeichert. Die Log-Datei wird für die anschließende Analyse ausgewertet. Durch diesen „Abtast-Aufnahmemodus“, wird nicht das ganze Signal kontinuierlich aufgenommen sondern zeitdiskret (in festen Zeitabständen) abgetastet. Durch eine zu grobe Abtastung kann es passieren, dass ein Flankenwechsel auf dem Hall-Signal nicht erkannt wird und es somit zu einer fehlerhaften Messung kommt, die die Analyse verfälschen würde. Aus diesem Grund muss sichergestellt sein, dass die Abtastfrequenz am PicoScope gewissen Anforderungen entspricht. Diese Anforderungen werden unter das Abtasttheorem zusammengefasst, das im Buch (Kiencke und Jäkel, 2005, S.213ff) nachgelesen werden.

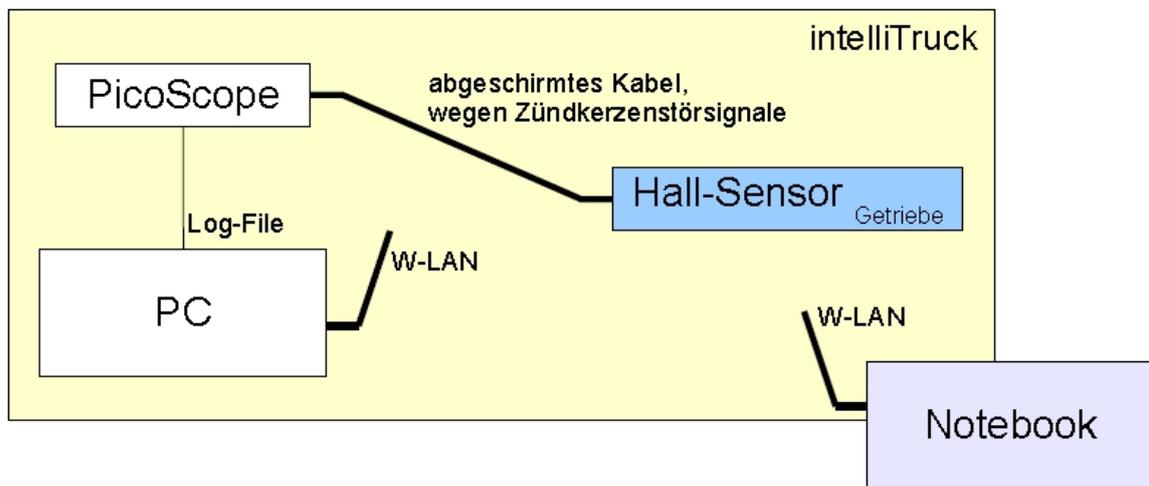


Abbildung 4.9.: Versuchsaufbau für Kennzahlenermittlung

Für eine fehlerfreie Aufnahme ergibt sich folgende Abtastfrequenz  $F'$ , die sich aus den Eigenschaften des Hall-Signals am Getriebe des intelliTrucks (siehe Tabelle 4.3) bestimmen lässt:

$$t_{min} = 0,65ms \rightarrow F_{max} = 1538Hz \quad (4.7)$$

Für eine Überabtastung, auch Oversampling genannt, muss gelten:

$$t' < t_{min} \rightarrow F' > F_{max} \quad (4.8)$$

Aus der vorangegangenen Ausführung muss die Abtastfrequenz über  $F_{max}$  und damit auch die Abtastzeit unter  $t_{min}$  liegen. Dafür eignet sich der „schnelle Block“-Modus bei einer Abtastzeit von  $384\mu s$ . Um den Zeitpunkt der steigenden und fallenden Flanken des Hall-Signals sehr genau detektieren und bestimmen zu können, wird eine Abtastzeit von  $20\mu s$  genutzt. Mit diesen Einstellungen kann eine Fahrt über etwa 20 Sekunden aufgezeichnet werden (vgl. Anhang A.1).

### 4.3.2. Aufbereitung und Auswertung der Messdaten

Das erzeugte Log-File mit einer Größe von 16 MB beinhaltet 1 Million Messpunkte. Da eine Auswertung per Hand zeitlich unrentabel wäre, wurde ein Auswertungstool speziell für die Auswertung der Messdaten geschrieben. Das in C geschriebene Auswertungstool liest das Log-File ein, detektiert Fehler, analysiert die Messpunkte und schreibt die Ergebnisse in eine CSV-Datei <sup>4</sup> [Shafranovich \(2005\)](#). Das Auswertungstool ermittelt die Zeitpunkte der auftretenden Flanken und berechnet damit die Zeitdifferenz  $\Delta t_{Zahn}$  zweier Zähne am Getrieberad in  $\mu s$  (vgl.  $\Delta t_{V_{max}}$  in Abbildung 4.8). Anhand dieser Zeitdifferenz  $\Delta t_{Zahn}$  und der Eigenschaft, dass das Getrieberad eine eins zu eins Übersetzung zu den Antriebsrädern aufweist, berechnet sich die Umfangsgeschwindigkeit wie folgt:

$$V_U = MS2KMH * \left[ Radumfang * \left( \frac{Zaehne}{US2SEC} * \Delta t_{Zahn} \right)^{-1} \right] \quad (4.9)$$

$$V_U = 3.6 * \left[ 0.565 * \left( \frac{48}{1000000} * \Delta t_{Zahn} \right)^{-1} \right] \quad (4.10)$$

$\Delta t_{Zahn}$  bildet eine Periode: High-Pegel und Low-Pegel (vgl. Abbildung 4.8). In der CSV-Datei wird der Zeitpunkt  $t$  und die dazu gerade herrschende Umfangsgeschwindigkeit  $V_U$  abgelegt. Dazu werden noch Informationen, wie z. B. die Anzahl fehlerhafter Pegelniveaus während einer Periode, die Rückschlüsse über die Richtigkeit der Berechnung geben können, in die CSV-Datei geschrieben. Dies kann bei nicht realistischen Messdaten oder unerwarteten Datenfolgen aufschlussreich sein.

Diese CSV-Datei wird anschließend mit einem Tabellenkalkulationsprogramm, wie OpenOffice<sup>5</sup> oder Microsoft Excel<sup>6</sup>, geöffnet und mit den angegebenen Punkten wird ein Graph (Umfangsgeschwindigkeit über die Zeit) gezeichnet.

<sup>4</sup> „Comma-Separated Values“

<sup>5</sup> [www.OpenOffice.org](http://www.OpenOffice.org)

<sup>6</sup> <http://office.microsoft.com/>

Zu den Messdaten aus dem Log-File ist die aufgeführte Abbildung 4.10 entstanden (weitere Diagramme sind im Anhang A.3 zu finden). Daraus werden die interessanten Beschleunigungsbereiche ausgewählt und weiter analysiert.

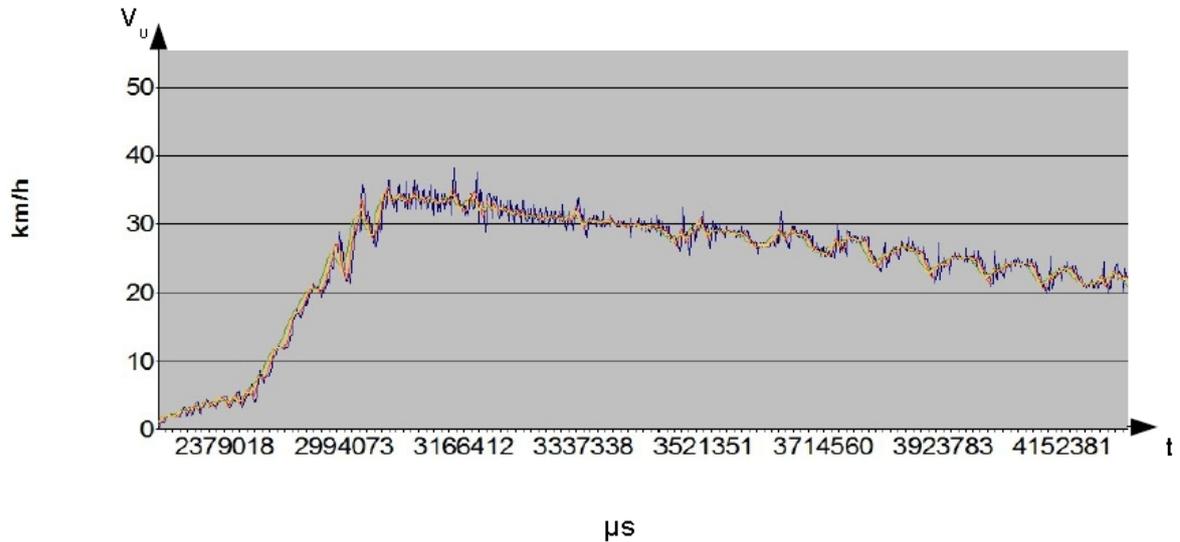


Abbildung 4.10.: 100 Prozent Schlupftest

Anhand der Zeiten  $t_0$  und  $t_1$  und der umgerechneten Umfangsgeschwindigkeit über den Zeitraum zweier Zähne  $V_0$  und  $V_1$  kann die Beschleunigung als Tangente (Differenzialquotient) angelegt und berechnet werden (vgl. Abbildung 4.11).

$$a = \frac{(V_1 - V_0)}{(t_1 - t_0)} \quad (4.11)$$

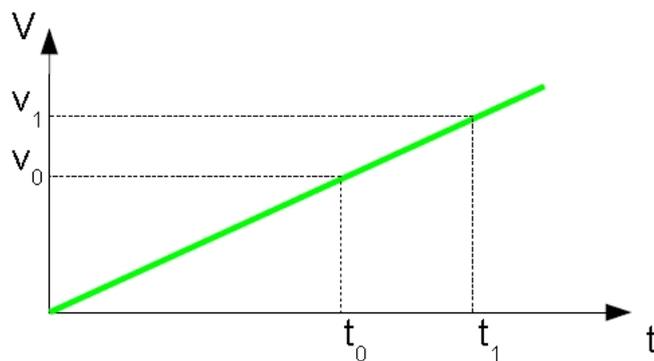


Abbildung 4.11.: Ermittlung der Beschleunigung über Differenzialquotient

Die Auswertung der Versuchsreihen ergeben folgende Beschleunigungen:

Fahrversuch	$t_0$ [ $\mu s$ ]	$t_1$ [ $\mu s$ ]	$\Delta t$ [s]	$V_0$ [ $\frac{km}{h}$ ]	$V_1$ [ $\frac{km}{h}$ ]	$a$ [ $\frac{m}{s^2}$ ]	$g$
100% - Schlupf (Antriebsachse ohne Bodenkontakt)	2638848	2990837	0,352	5,16	27,22	17,41	1,78
	7527219	8062320	0,535	4,54	33,37	14,97	1,53
	14667264	15111741	0,444	2,79	27,23	15,27	1,56
	19986512	20456734	0,470	5,07	33,37	16,72	1,70
Asphalt (Vollgas)	2035732	4796231	2,760	0,63	35,05	3,46	0,35
	11297361	13198970	1,902	0,81	16,04	2,22	0,23
	15580487	16633077	1,053	2,14	10,09	2,10	0,21
Schotter (Vollgas)	569733	2519388	1,950	0,44	20,28	2,83	0,29
	12580331	15303700	2,723	4,01	29,55	2,61	0,27
Asphalt (normale Fahrt)	2327121	6379827	4,053	0,88	20,09	1,32	0,13
	12867031	16767180	3,900	3,69	20,89	1,23	0,13

Tabelle 4.4.: Beschleunigungen bei unterschiedlichen Untergründen

### 4.3.3. Kennwerte

Zusammengefasst ergeben sich aus der Analyse folgende relevante Kennwerte:

PWM- Periode	17 $ms$
100% Schlupf (Vollgas)	16 $m/s^2$
Asphalt (Vollgas)	2,6 $m/s^2$
Schotter (Vollgas)	2,7 $m/s^2$

Tabelle 4.5.: Kennwerte

Daraus ergibt sich, dass der intelliTruck nie eine Beschleunigung über  $16m/s^2$  aufweisen kann. Wenn die obere Beschleunigungsgrenze überschritten werden sollte, liegt ein Messfehler vor.

## 4.4. Fehlerquellen

Die Fehlerquellen und deren Ursache bezogen auf die Hall-Sensoren, der Datenübertragung und des Mikrocontrollers werden in der Bachelorarbeit von Enrico [Hensel \(2007\)](#) behandelt. Hier werden nun die für die ASR-Steuerung relevanten Fehler(quellen) aufgeführt:

- Hall-Sensor
  - kein Signal
  - fehlerhaftes Signal: zu viele bzw. zu wenige Impulse
- ARM-Mikrocontroller
  - Softwaredefekt: Anwendung arbeitet fehlerhaft bzw. läuft nicht weiter
  - Hardwaredefekt
- Empfänger
  - kein Signal
  - fehlerhaftes Signal: zu viele bzw. zu wenige Impulse

## 4.5. Fahrsituationen und dafür benötigte Sensoren zur Bereichsermittlung

Die folgende Ausführung bezieht sich auf ein heckgetriebenes Fahrzeug mit Differenzial, wie es dem intelliTruck entspricht. Wie im Abschnitt Sensorik [4.2.2](#) beschrieben, werden für die Momentenbilanz die beiden Umfangsgeschwindigkeiten  $v_U$  und  $v_F$  bestimmt. Die Anzahl benötigter Sensoren kann variieren. Bei einer Geradeausfahrt z. B. liefern die beiden Vorderräder die gleiche Umfangsgeschwindigkeit, sodass ein Sensor an den Vorderrädern eingespart werden könnte. Während einer Kurvenfahrt legt das äußere Rad allerdings einen längeren Weg zurück, als das innere. Deshalb werden beide Vorderradsensoren benötigt, um  $v_U$  ermitteln zu können. Anhand dieses Beispielles soll verdeutlicht werden, dass die Anzahl der benötigten Sensoren von den Fahrmanövern abhängt. Zusätzlich zu den Fahrmanövern kann sich die Fahrbahnbeschaffenheit (Reibkoeffizient) im Laufe der Fahrt ändern und beeinflusst somit auch die Anzahl benötigter Sensoren.

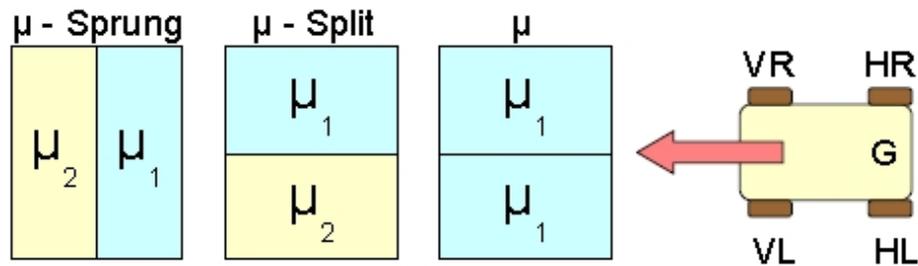


Abbildung 4.12.: verschiedene Reibkoeffizienten am Reifen nach Fahrbahnbeschaffenheit

Die Tabelle 4.6 gibt eine Übersicht über die Anzahl der benötigten Hall-Sensoren an den einzelnen Rädern unter Berücksichtigung verschiedener Fahrmanöver und Reibkoeffizienten (**V**orne, **H**inten, **L**inks, **R**echts, **G**etriebe z. B. VL: vorne links).

Reibkoeffizient	Fahrmanöver	Radsensoren
$\mu$	geradeaus Kurve	(VL   VR) & (HL   HR   G) VL, VR, HL, HR
$\mu - Split$	geradeaus Kurve	VL, VR, HL, HR VL, VR, HL, HR
$\mu - Sprung$	geradeaus Kurve	(VL   VR) & (HL   HR) VL,VR,HL,HR

Tabelle 4.6.: Fahrsituationen und benötigte Sensoren

Aus der Tabelle lässt sich ableiten, dass während einer Geradeausfahrt bei konstantem Reibkoeffizient  $\mu$  zwei Sensoren (z. B. VL und G) zur Ermittlung des Schlupfes ausreichen. Bei Kurvenfahrt werden beide Vorderradsensoren benötigt, da die Räder einen unterschiedlich langen Weg zurücklegen. Die Hinterräder drehen sich wegen der Kurvenfahrt und des eingebauten Differenzials unterschiedlich schnell, so dass die Antriebsräder auch einzeln betrachtet werden müssen.

Beim  $\mu - Split$  und  $\mu - Sprung$  kann der Sensor am Getrieberad gar nicht genutzt werden, da er wegen des Differenzials keine Aussage über die Raddrehzahlen an den beiden Antriebsrädern liefern kann.

## 4.6. Anforderungen und Beurteilungskriterien

Die funktionalen Anforderungen sind in den Kapiteln 3.3 und 4.1.2 beschrieben. Übersichtlichkeitshalber werden sie hier nochmals kurz aufgeführt:

- verbessertes Gripverhalten und Erhöhung der Vortriebskräfte
- Sicherung der Fahrstabilität bei heckgetriebenen Fahrzeugen bzw. Lenkbarkeit bei frontgetriebenen Fahrzeugen
- Sperrdifferenzialfunktion
- geringerer Reifenverschleiß
- Warnung vor Erreichen der physikalischen Stabilitätsgrenze bzw. Ausfall der Steuerung durch eine Informationsleuchte z. B. bei Glätte

Zu den nicht-funktionalen Anforderungen gehören Anforderungen (vgl. Reif, 2006):

- Zeitanforderungen: harte Echtzeit und damit vorhersagbares Verhalten
- Zuverlässigkeit und Verfügbarkeit: bei Nutzung des Fahrzeuges soll die ASR stets verfügbar sein. MTTR (mean time to repair) bei Fehler/Reset, muss das System in weniger als 17ms (eine PWM-Periode) wieder einsatzfähig sein
- Sicherheit (Safety): bei Absturz oder Totalausfall muss das Fahrzeug in einen sicheren Zustand gebracht werden (normal steuerbar ohne ASR-Funktionalität oder Einleitung einer Bremsung)

Laut dem Bremsenhandbuch (vgl. Breuer und Bill, 2004, S.78) wird ein ASR-System unter den folgenden Kriterien beurteilt:

- Beschleunigung auf unterschiedlichen Fahrbahnreibungswerten ( $\mu$ -Split)
- Beschleunigung auf Fahrbahnen mit niedrigem Reibwert
- Beschleunigung auf plötzlich wechselnden Fahrbahnuntergründen ( $\mu$ -Sprung)
- Steigungsvermögen auf Fahrbahnen mit unterschiedlichen Reibwerten ( $\mu$ -Split am Hügel)
- Regelkomfort / -geräusch
- Zurückrollen am Berg

# 5. Design

In diesem Kapitel wird das Software- und Hardwaredesign für die ASR-Steuerung beschrieben. Als Grundlage für die Entscheidungen dient die vorangegangene Analyse (siehe Kapitel 4).

## 5.1. Ablaufdiagramm und Sicherstellung der Bremsfunktionalität

In jedem ASR-System werden grundsätzlich folgende Aufgaben bearbeitet (vgl. Reichel, 2001, S.82):

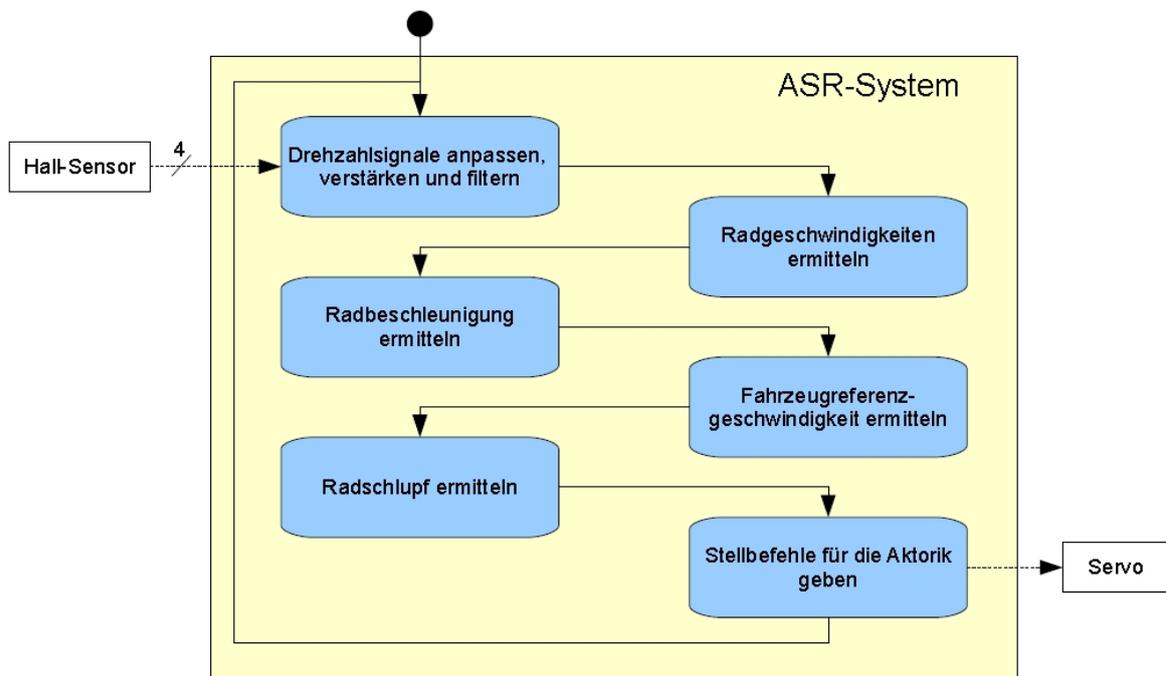


Abbildung 5.1.: Ablauf ASR-System

Die Signale, die von den Hall-Sensoren kommen, werden auf dem intelliTruck mittels Hardware aufbereitet. Diese Vorverarbeitung ist auf den ARM-Mikrocontroller abgestimmt und ermöglicht den direkten Anschluss der Hall-Sensoren an das I/O-Board. Nähere Informationen sind in den Arbeiten von [Nar \(voraussichtlich 2008\)](#) und [Schuckert \(2007\)](#) zu finden. nachgelesen werden. Die Bremsfunktionalität wird an einem Fahrzeug mit einem ASR-System dadurch gewährleistet, dass sie auf dem ABS-System aufsetzt (vgl. Kapitel 3.1.2.2 und 3.1.4). Damit kann sowohl die ASR-Steuerung als auch der Fahrer einen Bremseneingriff unabhängig einleiten. Am intelliTruck existiert kein ABS-System und die Bremse wird über den gleichen Servo wie für die Drosselklappe angesteuert (vgl. Kapitel 4.2.3).

Für den Fall, dass die ASR während einer Beschleunigungsphase eingreift, übernimmt sie die volle Kontrolle am Servo. Während der Regelungsphase der ASR werden folglich alle Steuerbefehle des Fahrers ignoriert. Problematisch wird es in einer Gefahrensituation, wenn der Bremsbefehl des Fahrers dadurch nicht beachtet wird. Die Bremsung verzögert sich bis der Fahrer die Kontrolle nach der Regelung zurückerhält. Dieser Bremsbefehl muss aus Sicherheitsgründen detektiert und schnellstmöglich umgesetzt werden, indem der Servo in die Bremsstellung gefahren wird.

Damit ist der Bremsbefehl des Fahrers gegenüber den ASR-Steuerbefehlen von höherer Priorität.

Mit dieser zusätzlichen Anforderung und der abzuarbeitenden Aufgaben in der ASR (vgl. Abbildung 5.1) ergibt sich folgende(r) Aufbau und Verbindungen auf dem intelliTruck:

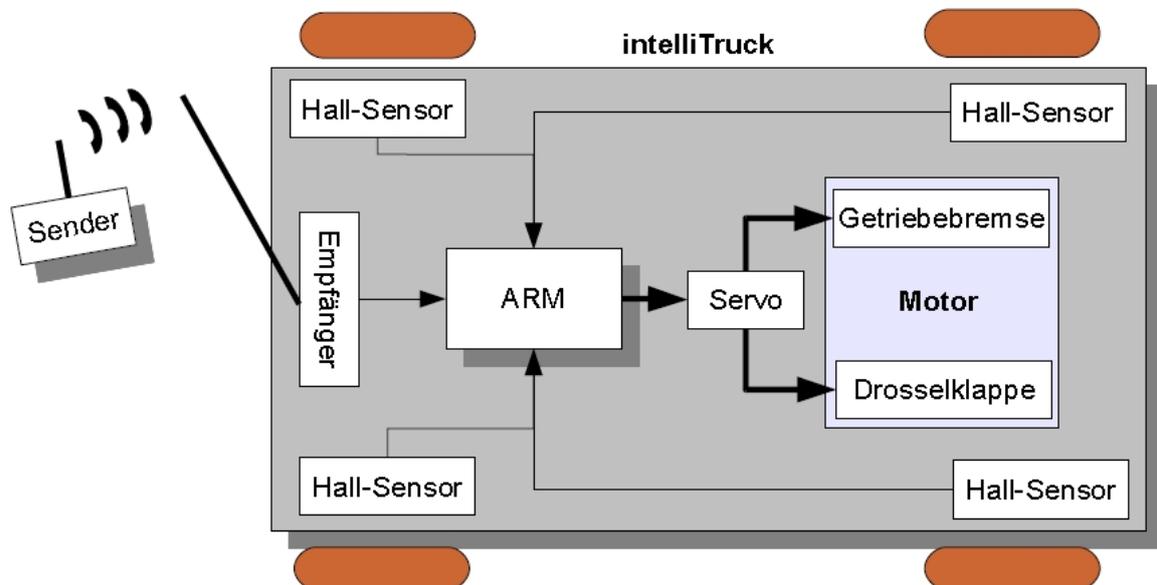


Abbildung 5.2.: Vernetzung Hardwarekomponenten für die ASR auf dem intelliTruck

## 5.2. Regelung

In der ASR werden die eingehenden Sensordaten ausgewertet und mit den Sollvorgaben abgeglichen. Daraus leiten sich anschließend Stellgrößen ab, die auf die Aktorik übertragen werden. Die Interpretation der Sensordaten und dessen Auswertung geschieht auf Grundlage eines mathematischen Modells des zu regelnden Systems. Dieses Modell spiegelt die physikalischen Eigenschaften und Zusammenhänge je nach Komplexität unterschiedlich gut wieder. Die Schwierigkeit liegt darin ein passendes Modell zu finden. Ein zu einfaches Modell, das nur wenige Größen mit einbezieht, beschreibt das System für die Regelung möglicherweise unzureichend. Umgekehrt wird bei zu vielen Größen das Modell sehr komplex und damit schwer umzusetzen.

Die Herausforderung in der Umsetzung liegt in der Nichtlinearität und Unbestimmtheit des Modells. Je mehr Größen mit einbezogen werden, desto mehr Freiheitsgrade liegen vor, die berücksichtigt werden müssen. Für solche Problemstellungen werden z. B. schaltende- und Fuzzy- Regler genutzt. Die aufgeführten Regler werden in dieser Arbeit nicht weiter beschrieben. Für weitere Informationen wird auf die folgende Literatur verwiesen: [Gevatter \(2000\)](#). Die Vorteile dieser Regler liegt in ihrer Stabilität. Sie erfordern aber durch die Komplexität ihrer Algorithmen eine erhöhte Rechenleistung, die im Embeddedbereich nicht immer gewährleistet ist. Somit werden einfachere und schnellere Lösungen bzw. Algorithmen für den Real-Time-Einsatz im Embeddedbereich gesucht (vgl. [Choi und Hong, 2002](#), S.1675).

Eine Möglichkeit diese Herausforderung zu lösen ist der Ansatz über eine hybride Regelung (vgl. [Unbehauen \(2000\)](#) und [Antsaklis \(2000\)](#)). Dazu gibt es ein Framework Namens: HYSDEL<sup>1</sup>. Die Beschreibung des Frameworks kann in diesem Dokument: [Torrì und Bemporad \(2004\)](#) und auf deren Homepage: [The Automatic Control Laboratory \(2008\)](#) nachgelesen werden. Es unterteilt das nichtlineare Problem in lineare Teilprobleme zwischen denen die hybride Regelung wechselt. Da die Teilprobleme linear sind, können sie leicht mit wenig Rechenaufwand erfasst und analysiert werden. Somit kann eine ASR sicher und stabil in einem Real-Time Embeddedmilieu ein- und umgesetzt werden.

Das oben erwähnte HYSDEL - Framework kommt in dieser Arbeit nicht zum Einsatz. Die Grundidee der Linearisierung wird im folgenden aufgegriffen und für die Regelung umgesetzt.

---

<sup>1</sup>Hybrid Systems Description Language

Nach der Linearisierung sieht die  $\mu$  - Schlupf - Kennlinie wie folgt aus:

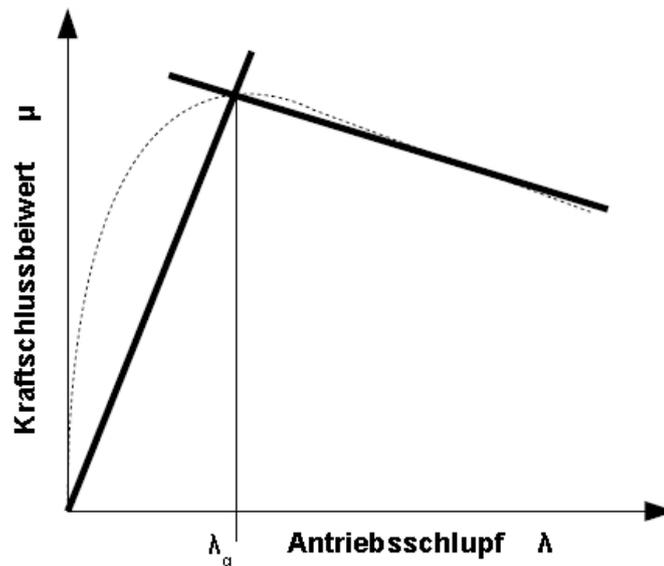


Abbildung 5.3.: Linearisierung einer  $\mu$  - Schlupf - Kennlinie

Daraus folgen zwei lineare Graphen, die zwei Bereiche beschreiben. Diese beiden Bereiche sind schon aus der Analyse (siehe Kapitel 4) als „stabilen“ und „instabilen“ Bereich bekannt. Daraus ergeben sich also zwei Zustände, zwischen denen der Regler wechselt:

Zustand	Bereich aus Analysekapitel	Schlupfintervall [%]
NORMAL	„stabil“	$[0 : \lambda_g]$
EMERGENCY	„instabil“	$] \lambda_g : 100]$

Tabelle 5.1.: Zustände nach Linearisierung

Mit diesen beiden Zuständen kann der Schlupf des intelliTrucks geregelt werden. Die  $\mu$  - Schlupf - Kennlinien sind von der Fahrbahnbeschaffenheit abhängig (siehe Abbildung 4.3) und der kritische Bereich, an dem das Fahrzeug von der Haft- zur Gleitreibung übergeht, liegt zwischen 8 und 20% Schlupf (vgl. Reichel, 2001, S.32).

Eine Regelung auf acht Prozent Schlupf stellt eine Lösung dar, die bei sehr schlechten Fahrbahnverhältnissen ausreichend ist. Bei besseren bis sehr guten Fahrbahnverhältnissen kann diese Lösung der Anforderung nach Erhöhung der Vortriebskräfte nicht gerecht werden (siehe Anforderungen 4.6). Somit ist es notwendig die Fahrbahnbeschaffenheit bestimmen zu können, auch wenn diese sich während der Fahrt ändert.

Der kritische Punkt, das Maximum der  $\mu$  - Schlupf - Kennlinie, zeichnet sich dadurch aus, dass die Umfangsgeschwindigkeit schlagartig ansteigt (vgl. Abbildung 4.2). Um diesen Anstieg erkennen zu können, werden in der Analyse empirische Beschleunigungstest mit dem intelliTruck gemacht. Die ermittelten, maximal möglichen Beschleunigungen auf unterschiedlichen Fahrbahuntergründen sind in der Tabelle 4.4 aufgelistet und dienen im weiteren Verlauf als Referenzwerte für den Regler.

Beim Auftreten schlagartiger Anstiege der Umfangsgeschwindigkeit wird der aktuelle Schlupf abgespeichert. Dieser dient als obere Grenze des zulässigen Schlupfes  $\lambda_g$  (vgl. Abbildung 5.3). Beim Überschreiten des  $\lambda_g$  greift die Regelung ein, bis der Schlupf sich wieder unterhalb des Grenzwertes befindet. Laut Choi und Hong (2002) kann es bei dieser Regelung zu einem so genannten „chattering“ kommen. Das bedeutet, dass der Regler im schnellen Wechsel eingreift (Überschreiten des  $\lambda_g$ ) und die Kontrolle wieder dem Fahrer überlässt (Unterschreiten des  $\lambda_g$ ). Um dieses Schwingen zu dämpfen wird eine Konstante  $\alpha$  eingeführt. Diese wird vom  $\lambda_g$  abgezogen und dient als neuer Grenzwert, der unterschritten werden muss, bevor der Regler die Kontrolle abgibt. Das  $\alpha$  sollte nicht zu groß gewählt werden, da der Regler sonst nicht effizient arbeiten kann.

Aus der Analyse für die Beschleunigung auf unterschiedlichen Untergründen ergibt sich ein Grenzwert  $a_g$ , der bei Überschreiten auf durchdrehende Räder hindeutet. Dieser wird auf:  $a_g = 5 \frac{m}{s^2}$  festgelegt. Er liegt oberhalb der maximalen Beschleunigung auf Schotter  $2,7 \frac{m}{s^2}$  und unterhalb vom 100%-Schlupf-Test  $16 \frac{m}{s^2}$  (vgl. Tabelle 4.5). Die Grenze für den Schlupf wird auf:  $\lambda_g = 20\%$  festgelegt, da dies etwa die obere Grenze für den kritischen Bereich darstellt (vgl. Reichel, 2001, S.32).

Damit ergeben sich folgende Übergänge zwischen den beiden Zuständen des ASR-Reglers:

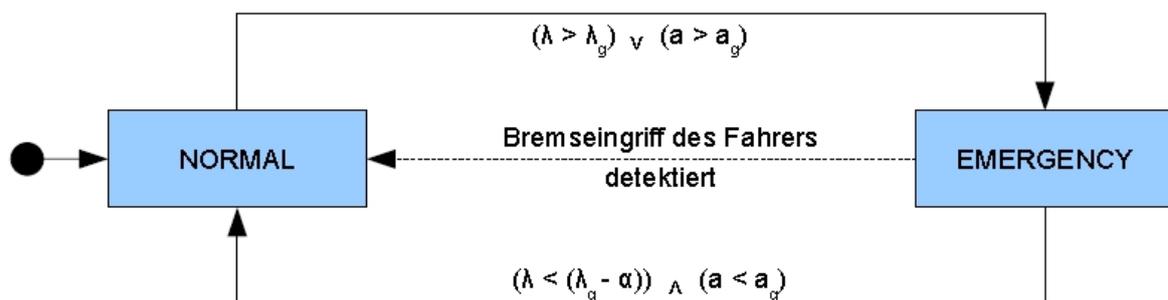


Abbildung 5.4.: Zustandsdiagramm der ASR nach Linearisierung

Der Zustandsübergang (siehe Abbildung 5.4: gestrichelte Linie), „Bremseneingriff des Fahrers detektiert“, dient der Veranschaulichung, dass die ASR in solch einem Fall keine weiteren Eingriffe vollzieht. Die Kontrolle über die Fahrfunktionen des intelliTrucks liegt somit

auf seiten des Fahrers. Der Übergang vom EMERGENCY- in den NORMAL-Zustand geschieht dabei automatisch über den unteren Zustandsübergang. Beim Abbremsen geht der Antriebsschlupf gegen null ( $< a_g$ ) und die Beschleunigung in Fahrtrichtung wird negativ ( $< (\lambda_g - \alpha)$ ).

Wie im Kapitel 4.2.3 beschrieben wird, ist die Drosselklappe die einzige Stellgröße am intelliTruck und diese wird beim Eingriff der ASR in die Grundstellung gefahren. Laut Bosch (vgl. Kapitel 3.1.4) sei die Drosselklappenverstellung wegen der relativ langsamen Reaktionszeit nicht in der Lage eine befriedigende Lösung für die ASR bereitzustellen. In wie weit diese Aussage auf den intelliTruck zutrifft, muss in einem Test ermittelt werden.

### 5.3. Hardware und Signalerfassung auf dem ARM

In der Abbildung 5.2 ist der Aufbau der Komponenten in Bezug auf die ASR des intelliTrucks aufgezeigt. Das PWM-Signal des Empfängers, das für die Bremsfunktionalität ausgelesen werden muss und die Signale der Hall-Sensoren werden an den ARM-Mikrocontroller angeschlossen. In diesem Abschnitt wird analysiert welche Möglichkeiten es gibt die eingehenden Signale für die ASR zu digitalisieren.

Das Einlesen der Signale kann über verschiedene Arten geschehen. Wie in Kapitel 4.2.1 angedeutet, werden die bereitgestellten Möglichkeiten des ARM-Mikrocontrollers durch das I/O-Board eingeschränkt, da nicht alle PINs des ARM-Mikrocontrollers auf das I/O-Board herausgeführt werden. Aus diesem Grund muss ein Kompromiss eingegangen werden.

Die beiden Signaltypen, Hall- (siehe Abbildung 4.8) und PWM-Signal (siehe Abbildung 2.6), sind sehr ähnlich, denn ihre Pegelhöhen sind gleich und liegen zwischen 0 und 3,3V. Sie unterscheiden sich nur in der Frequenz der auftretenden Flanken. Das PWM-Signal besitzt eine Periodendauer von 17ms und beim Hall-Signal ändert sich diese je nach Umfangsgeschwindigkeit der Räder von 1,32ms, bei Höchstgeschwindigkeit, bis  $\rightarrow \infty$ , bei Stillstand des intelliTrucks.

Beim PWM-Signal gibt die Breite des High-Pegels den Inhalt des Signals wieder (vgl. Kapitel 2.3). Die Bestimmung der Breite des High-Pegels kann über einen Interrupt geschehen: Bei der steigenden Flanke wird ein Interrupt ausgelöst, der in der entsprechenden ISR einen Timer auf dem ARM-Mikrocontroller startet und bei der fallenden Flanke stoppt. Der Inhalt des Timers entspricht der Breite des High-Pegels. Die gleiche Vorgehensweise kann für das Hall-Signal genutzt werden, indem die Zeit zwischen zwei steigenden Flanken gemessen wird. Dies entspricht der Zeit einer vierundzwanzigstel Umdrehung des Rades (vgl. Tabelle 4.2) und damit kann z. B. die Umfangsgeschwindigkeit berechnet werden.

Anstelle eines Interrupts, der den Timer ein- und ausschaltet, kann das Capture-Register des Timers genutzt werden. Dazu wird der Timer einmal eingeschaltet, der kontinuierlich weiterläuft. Beim Überlauf startet er von null. Wenn z. B. eine positive Flanke vom (Hall- bzw. PWM-) Signal eintrifft, wird der aktuelle Zählerstand des Timers im Capture-Register abgelegt. Dieser wird von der Software ausgelesen und zwischengespeichert. Beim Eintreffen der nächsten Flanke (je nach Einstellung: steigende oder fallende) wird wieder der aktuelle Zählerstand des Timers im Capture-Register abgelegt und anschließend von der Software ausgelesen. Die Differenz der beiden Zählerstände ergibt die Zeit zwischen den beiden Flanken, die für weitere Berechnungen genutzt werden können.

Eine weitere Möglichkeit ist das Abtasten der Signale. Dieses Verfahren wird im Kapitel 4.3.1 zur Kenndatenermittlung genutzt und wird dort beschrieben. Diese Möglichkeit wird zur Vollständigkeit mit aufgeführt, aber nicht weiter verfolgt.

Zur Ermittlung der Umfangsgeschwindigkeit der Reifen ist es möglich die positiven Flanken des Hall-Signals über einen festgelegten Zeitraum zu zählen. Dafür kann der Counter des ARM-Mikrocontrollers genutzt werden. Für eine ASR, auf Grundlage dieses Verfahrens, wären dafür eine Zahnanzahl des Inkrementgebers um 2500 pro Radumdrehung laut Hans-Rolf Reichel nötig (vgl. [Reichel, 2001](#), S.69). Der intelliTruck besitzt 24 Zähne pro Umdrehung.

Ein Ansatz zur Digitalisierung des PWM-Signals ist den Analog-Digital-Wandler des ARM-Mikrocontrollers zu nutzen. Dafür wird das PWM-Signal über ein RC-Glied gelegt, das anschließend über den AD-Wandler eingelesen wird. Auf die Umsetzung und den Test wird hier nicht weiter eingegangen. Im verwendeten Versuchsaufbau sind die sich ändernden High-Pegelbreiten so gut wie nicht zu unterscheiden, da der Kondensator mit der Zeit wieder Spannung verliert.

Zusammenfassend ergibt sich die Übersicht der Einlesearten der beiden Signaltypen (vgl. Tabelle 5.2):

Signaltyp	Einleseart	Umsetzbarkeit
PWM-Signal	Timer mit Interrupt	+
	Timer mit Capture-Register	+
	RC-Glied + AD-Wandler	-
	Polling (Sampeln)	+/-
Hall-Signal	Timer mit Interrupt	+
	Timer mit Capture-Register	+
	Counter	-
	Polling (Sampeln)	+/-

Tabelle 5.2.: Einlesemöglichkeiten der eingehenden Signale auf dem ARM-Mikrocontroller

Der ARM-Mikrocontroller besitzt vier Timer mit je vier Capture-Register. Von diesen insgesamt 16 Capture-Registern bietet das I/O-Board nur Zugriff auf vier. Es werden für die ASR vier Hall- und ein PWM-Signal(e) benötigt. Somit können alle fünf Signale, aber mindestens eins muss, mittels Interrupt digitalisiert werden.

## 5.4. Software: Systemarchitektur

Die Systemarchitektur ist eines der grundlegenden Aspekte dieser Arbeit, die als Grundlage für die weitere Umsetzung und Implementierung dient. Dabei soll die Anforderung nach harter Echtzeit und vorhersagbares Verhalten der ASR (siehe Kapitel 4.6) erfüllt werden. Der für diese Arbeit verwendete ARM-Mikrocontroller besitzt kein mitgeliefertes Betriebssystem, das einen Scheduler bereitstellt. So ergeben sich zwei Möglichkeiten:

- ein Real Time Operating System (RTOS)<sup>2</sup> auf dem ARM aufzusetzen
- einen eigenen Scheduler zu schreiben

In dem Buch von Pont (2001) werden für diese Problemstellung der harten Echtzeit und vorhersagbares Verhalten auf einem „single-processor system“ drei Lösungsmustern vorgestellt, die schnell auf dem ARM-Mikrocontroller umsetzbar sind. Damit wird eine Suche nach einem geeigneten RTOS und dessen Analyse, ob es die Zeitanforderungen einhalten kann, eingespart.

Als Grundgedanke wird die Zeitachse in äquidistante Intervalle, auch TimerTicks (TT) genannt, mittels eines Timers unterteilt. Innerhalb dieser TT werden die Aufgaben (Tasks) des Programmes ausgeführt.

Die drei Lösungsmustern von Pont werden im folgenden beschrieben:

### 5.4.1. Kooperativer Scheduler

Der kooperative Scheduler ist eine so genannte Single-Tasking System Architektur. Dabei wird die laufende Task bis zum Ende ausgeführt und anschließend geht die Kontrolle zum Scheduler zurück, der die nächste wartende Task startet bzw. auf den nächsten TT wartet, wenn keine Task zur Ausführung bereitsteht. Die laufende Task kann nicht unterbrochen werden (non-preemptive). Eine schnelle Reaktion auf externe Ereignisse ist nur unter einem guten Design gegeben. Die Reaktionszeit hängt unter anderem davon ab, wie oft die dafür zuständige Task ausgeführt wird.

---

<sup>2</sup>z. B.: [www.freertos.org](http://www.freertos.org)

Ein Vorteil dieses Schedulers liegt darin, dass der Zugriff auf gemeinsam genutzte Ressourcen sicher ist (thread-safe). Jede Aktion des Schedulers ist vorhersagbar.

### 5.4.2. Preemptiver Scheduler

Der preemptive Scheduler ist eine so genannte Multi-Tasking System Architektur. Dabei können Tasks, die gerade ausgeführt werden, unterbrochen werden (preemptive). Dieser Scheduler ist im Vergleich zum kooperativen dadurch komplexer und weniger überschaubar. Die Aktionen des Schedulers sind damit nicht gut vorhersagbar. Es muss darauf geachtet werden, dass gemeinsam genutzte Ressourcen bei Wechsel der Tasks gegebenenfalls geschützt werden müssen, um Dateninkonsistenzen zu vermeiden. Auf externe Ereignisse kann bei diesem Scheduler schnell reagiert werden.

### 5.4.3. Hybrider Scheduler

Der hybride Scheduler erlaubt eingeschränkte Multi-Tasking Funktionalität und arbeitet wie ein kooperativer. Der Unterschied liegt darin, dass es eine Task gibt (preemptive Task), die die kooperativ ablaufenden unterbrechen kann. Eine preemptive Task ist z. B. nötig, wenn eine Schnittstelle während des Betriebes regelmäßig versorgt werden muss. Dieser Scheduler kann wie der preemptive schnell auf externe Ereignisse reagieren. Bei einem guten Design sind die Aktionen des Schedulers vorhersagbar.

### 5.4.4. Entscheidung für den Cooperativen Ansatz mit einem Interrupt

Kategorie	Scheduler		
	kooperativer	preemptiver	hybrider
vorhersagbares Verhalten	+	-	+
einfach implementierbar	+	-	+
thread-safe	+	-	+/-
preemptive Task(s)	-	+	+ (eine)

Tabelle 5.3.: Schedulingmethoden

Wie aus der Tabelle (vgl. Tabelle 5.3) abzulesen ist, kommt der kooperative und er hybride Scheduler als Lösung in Frage: Beide weisen das geforderte vorhersagbare Verhalten (siehe Kapitel 4.6) auf, dazu sind sie einfach implementierbar und thread-safe.

Da z. B. keine Schnittstelle in regelmäßigen Abständen zu versorgen ist, wird in dieser Arbeit ein kooperativer Scheduler umgesetzt.

Laut Pont darf auf einem Mikrocontroller nur eine Interruptquelle und zwar die des Timers für den TT aktiviert sein, alle anderen müssen deaktiviert sein. Andernfalls kann die Funktionsfähigkeit des Schedulers nicht garantiert werden (vgl. Pont, 2001, S.263). Durch die Einschränkungen, die mit dem I/O-Board einhergehen, muss ein Sensor-Signal über einen Interrupt digitalisiert werden (vgl. Kapitel 5.3).

Der Interrupt des Sensor-Signals wird nun in den Scheduler integriert. Dazu ergeben sich weitere Bedingungen, die einzuhalten sind, um die Funktionsfähigkeit des Schedulers garantieren zu können. Je nach Wahl des TT können währenddessen kein bis mehrere PWM-Interrupts auftreten. Diese müssen zusätzlich zu den Tasks innerhalb des TT bearbeitet werden, damit keine Überschneidungen mit den Tasks entstehen. So muss stets gelten:

$$TT > (T_{scheduler} + T_{task} + N * T_{interrupt}) \quad (5.1)$$

- $N$  entspricht der maximalen Anzahl auftretender PWM-Interrupts pro TT
- $T_{interrupt}$  entspricht der Bearbeitungszeit des PWM-Interrupts
- $T_{scheduler}$  entspricht der Bearbeitungszeit des Schedulers
- $T_{task}$  entspricht der Bearbeitungszeit einer Task

Zur Digitalisierung des Sensor-Signals über einen Interrupt, kommt das PWM-Signal des Empfängers in Frage. Das PWM-Signal erzeugt kontinuierlich zwei Interrupts (steigende und fallende Flanke) pro PWM-Periode (17 ms), was einer Frequenz von 118 Hz entspricht. Ein Hall-Signal weist bei Höchstgeschwindigkeit eine Frequenz von 766 Hz (vgl. Tabelle 4.3) auf. Somit besitzt das  $N$  aus der Gleichung 5.1 einen konstanten Wert und ist nicht von der Geschwindigkeit des Fahrzeuges abhängig.

Der ARM-Mikrocontroller bietet die Möglichkeit Interruptquellen zu priorisieren. Bei gleichzeitigem Auftreten des Timer- und PWM-Interrupts erhält der Timer-Interrupt immer den Vorzug, um keine Verschiebung des TT entstehen zu lassen. Eine Verschiebung des TT würde die Vorhersagbarkeit des Schedulers nichtig machen.

## 5.5. Fehlerhandhabung

Die Fehler(quellen), die bei den Hall-Sensoren und dem Empfänger auftreten können, sind ähnlich (siehe Kapitel 4.4) und somit wird sich die Fehlerhandhabung gleichen.

Wenn kein Signal detektiert wird, ist der Empfänger bzw. Hall-Sensor bei rollendem Fahrzeug ausgefallen und der Signalpegel liegt bei null. In diesem Fall wird untersucht, welche Auswirkungen der Ausfall hat und ob der Ausfall mit den anderen funktionierenden Sensoren kompensiert werden kann. Beispielsweise kann der Ausfall eines Hall-Sensors an den Vorderrädern in Kauf genommen werden, denn der zweite liefert genug Daten, um die Umfangsgeschwindigkeit des Reifens und damit die Fahrzeuggeschwindigkeit zu ermitteln. Die Genauigkeit leidet jedoch darunter. Anders verhält es sich mit den Antriebsrädern. Wenn einer dieser ausfällt, kann z. B. kein  $\mu - split$  (siehe Abbildung 4.12) detektiert werden und im schlimmsten Fall wird das langsam bis stehende Rad auf der griffigeren Fahrbahnseite für die Schlupfbestimmung herangezogen. Das kann zur Konsequenz haben, dass der Eingriff der Regelung ausbleibt. Aus diesem Grund wird die ASR bei Ausfall eines Hall-Sensors an den Antriebsrädern in einen sicheren Zustand (SAFE) überführt.

Beim Ausfall des Empfängers entsteht anders als bei den Hall-Sensoren eine kritische Situation, denn damit ist die Verbindung zum Fahrer unterbrochen. In diesem Fall wird die ASR in einen Fehlerzustand (ERROR/STOP) gebracht, der das Fahrzeug zum Stehen bringt, um keine Gefahrensituationen entstehen zu lassen.

Fehlerhafte Signale machen sich dadurch bemerkbar, dass zu viele bzw. zu wenig Impulse registriert werden. Es gibt verschiedene Methoden mit solchen Schwankungen umzugehen. In dieser Arbeit werden zwei verwendet, die im folgenden beschrieben werden:

Zum einen kann ein MA-Filter<sup>3</sup> genutzt werden, der den Durchschnittswert mehrerer Messungen bildet. Damit fallen auftretende Abweichungen weniger ins Gewicht.

Average-Filter:

$$avg[n] = \sum_{i=0}^N b_i * y[n - i] \quad | \quad b \hat{=} Gewichtungsfaktor \quad (5.2)$$

Moving-Average-Filter mit einer Filterlänge von  $N = 2$ :

$$avg_{mov}[n] = \frac{1}{3} * y[n] + \frac{1}{N+1} * y[n-1] + \frac{1}{N+1} * y[n-2] \quad | \quad b = \frac{1}{N+1} \quad (5.3)$$

Zum anderen ist es möglich einen Plausibilitätstest durchzuführen, der überprüft, ob der ermittelte Messwert des Sensors mit dem physikalisch möglichen übereinstimmt. Es ist z. B. nicht möglich, dass die Geschwindigkeit des Fahrzeuges schlagartig von 20 auf 40 km/h

---

<sup>3</sup>Moving Average Filter

steigt (vgl. Reichel, 2001). Mit den Referenzwerten des intelliTrucks (siehe Tabelle 4.4 und 4.2) kann die Umfangsgeschwindigkeiten der Räder je nach Fahrbahnbeschaffenheit vorausgerechnet werden. Diese Ergebnisse dienen anschließend als Vergleichswerte zu den neu eingelesenen Sensormesswerten und geben Hinweise auf die Richtigkeit der Sensormesswerte.

Auf dem ARM-Mikrocontroller besteht die Gefahr, dass sich die Software aufhängt bzw. der ARM-Mikrocontroller abstürzt. Um dies erkennen und Gegenmaßnahmen einleiten zu können, wird ein Watchdog benutzt. Weitere Informationen dazu können in Pont (2001) und dem Datenblatt NXP (2007) nachgelesen werden.

## 5.6. ASR-Zustandsdiagramm und daraus resultierende PWM-Steuersignale

Im vorherigen Abschnitt wird die Handhabung von Fehlern beschrieben, wodurch weitere Zustände zum Zustandsdiagramm (siehe Abbildung 5.4) hinzukommen. Die Erweiterung ist in der Abbildung 5.5 abgebildet.

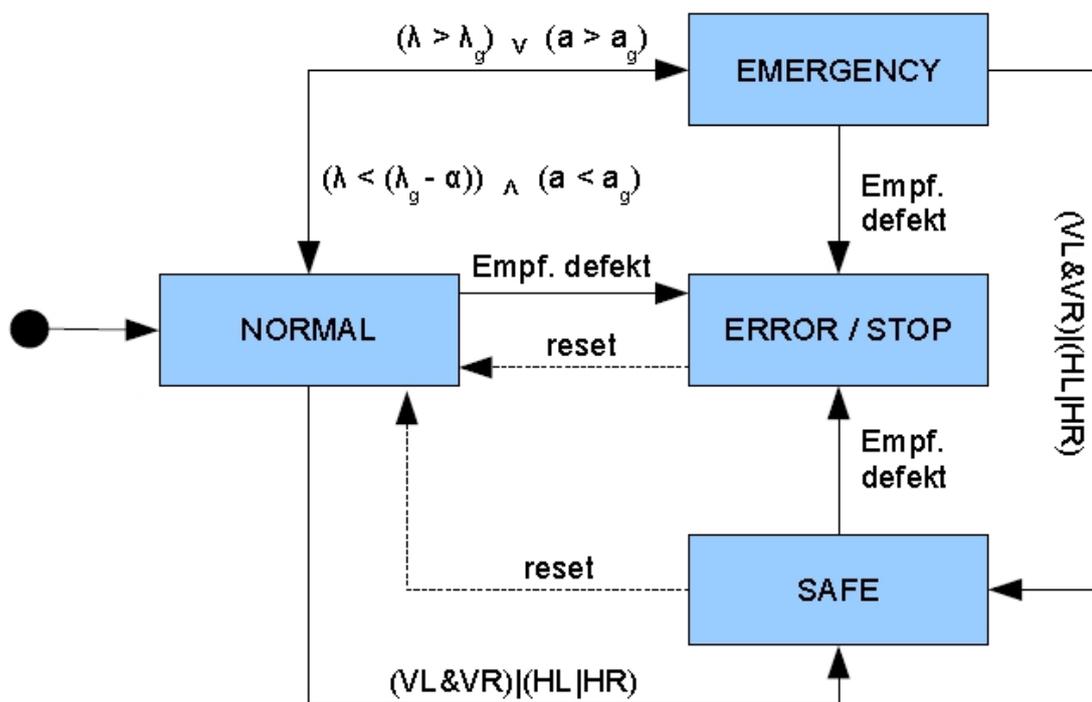


Abbildung 5.5.: Zustandsdiagramm der ASR mit Fehlerzustand

- Empf. defekt  $\hat{=}$  Empfänger defekt
- (VL&VR)|(HL|HR)  $\hat{=}$  Zustandsübergang erfolgt, wenn ein Hall-Sensor der Hinterräder oder beide Hall-Sensoren der Vorderräder defekt sind

Je nach Zustand der ASR wird das PWM-Steuersignal verschieden beeinflusst. Unter Idealbedingungen wird die PWM des Empfängers direkt an den Servo weitergeleitet. Bei einem Eingriff der ASR wird sie verändert.

PWM-Steuersignal		Fahrerbefehl (PWM des Empfängers)		
		bremsen	neutral	beschleunigen
Zustand der ASR	NORMAL	<b>bremsen</b>	<b>neutral</b>	<b>beschleunigen</b>
	EMERGENCY	<b>bremsen</b>	<b>neutral</b>	<b>neutral</b>
	SAFE	<b>bremsen</b>	<b>neutral</b>	<b>langsam beschleunigen</b>
	ERROR / STOP	<b>bremsen</b>	<b>bremsen</b>	<b>bremsen</b>

Tabelle 5.4.: PWM-Steuersignal in Abhängigkeit vom Zustand der ASR und vom Fahrerbefehl

Der Bereich „bremsen“ erstreckt sich vom kleinsten Winkel, der mit dem Servo angesteuert werden kann, bis hin zur neutralen Mittelstellung (vgl. Kapitel 2.3). Ähnliches gilt für den Bereich „beschleunigen“, der sich von der Mittelstellung bis zum größtmöglichen Winkel beläuft.

Aus dieser Tabelle ist zu erkennen, dass das Bremsen von hoher Priorität ist, denn der Bremsbefehl des Fahrers (PWM des Empfängers) wird zustandsunabhängig auf den Servo direkt weitergeleitet.

## 6. Implementierung

In diesem Kapitel wird das entwickelte Design (siehe Kapitel 5) auf dem ARM-Mikrocontroller umgesetzt. Dazu wird das Entwicklungswerkzeug, die Datenstrukturen und der Code beschrieben bzw. erläutert.

### 6.1. Entwicklungswerkzeug

Als Entwicklungswerkzeug wird das zum ARM-Mikrocontroller mitgelieferte Programm HiTOP5 von der Firma Hitex in der Version 5.20<sup>1</sup> eingesetzt.

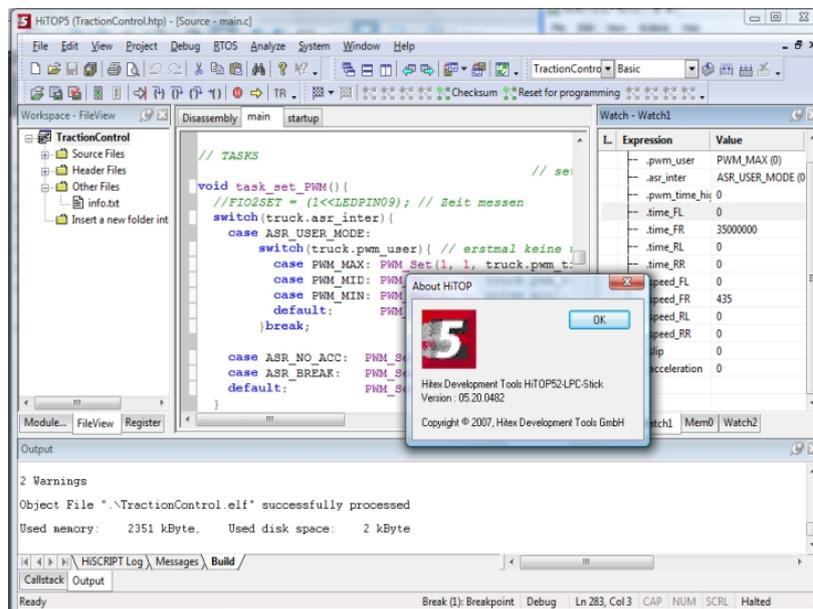


Abbildung 6.1.: Entwicklungsumgebung HiTOP5

Das StartEasy-Tool (siehe Anhang A.2) ermöglicht die Erstellung von Startupcode und Einstellungen wie z.B. die Pinbelegung. Der erzeugte Projektcode enthält neben den eigenen

<sup>1</sup>[http://www.hitex.com/products.html?con\\_hitop.html~content](http://www.hitex.com/products.html?con_hitop.html~content)

Einstellungen zusätzlich alle Standardeinstellungen. Weitere Informationen zum StartEasy-Tool können im Anhang [A.2](#) nachgelesen werden.

## 6.2. Scheduler

Der kooperative Scheduler, der in Kapitel [5.4](#) beschrieben wird, wird nach dem Buch [Pont \(2001\)](#) umgesetzt. Der Timer, hier *TIMER0*, löst nach Ablauf der vordefinierten Zeit (*TIMER\_CPM*) einen Interrupt aus. Exemplarische Einstellungen für *TIMER\_CPM* sind in der Tabelle [6.1](#) dargestellt. In der ISR (*ISR\_Timer0()*) wird die Aufgabenliste, die alle zu bearbeitenden Aufgaben (Tasks) des Systems enthält, mit der Funktion *SCH\_Update()* aktualisiert (vgl. Abbildung [A.6](#)). In der *while(1)*, die sich in der *main()* befindet, wird mit der Funktion *SCH\_Dispatch\_Tasks()* die Aufgabenliste durchlaufen, die als nächstes auszuführende Task ermittelt und diese anschließend gestartet (vgl. Abbildung [A.7](#)). Initialisiert wird der Scheduler mit der Funktion *SCH\_init()* (vgl. Abbildung [A.8](#)), die unter anderem die Aufgabenliste leert. Gestartet wird der Scheduler mit der Funktion *SCH\_Start()* (vgl. Abbildung [A.7](#)). Grundsätzlich wird pro TimerTick (TT) eine Task ausgeführt, dabei muss sichergestellt sein, dass ihre Bearbeitungszeit  $T_{task}$  die Vorgabe aus der Gleichung [5.1](#) einhält, da sie sonst mit der neu startenden Task kollidiert.

Tasks werden mit der Funktion *SCH\_Add\_Task(task, DELAY, PERIOD)* (vgl. Abbildung [A.7](#)) in die Aufgabenliste eingefügt, dabei beschreibt die Periode (*PERIOD*) die Anzahl an TT's bis die Task (*task*) wieder ausgeführt wird und das Delay (*DELAY*) beschreibt die Verzögerung in TT's bis die Task erstmals gestartet wird. Mit *SCH\_Delete\_Task(task\_index)* wird die Task (*task\_index*) entsprechend entfernt.

Die Tabelle [6.1](#) gilt unter der Voraussetzung, dass die CPU mit 72 MHz getaktet wird und damit die periferale clock bei 18 MHz liegt. In der Tabelle [6.1](#) bezeichnet *N* die maximal auftretenden PWM-Interrupts pro TT (vgl. Gleichung [5.1](#)).

TT [ $\mu s$ ]	TIMER_CPM	N
2000	36000	2
1000	18000	2
500	9000	1
250	4500	1
125	2250	1
63	1134	1
31	558	1

Tabelle 6.1.: exemplarische Einstellungen des TimerTicks

Bei einem TT von  $2\text{ms}$  bzw.  $1\text{ms}$  treten maximal zwei PWM-Interrupts innerhalb eines TTs auf. Das wiederholt sich alle  $\sim 9$  bzw.  $17$  TTs. Zwischen der steigenden und fallenden Flanke liegen minimal  $1\text{ms}$  (vgl. Kapitel 2.3). Ab einem TT unter  $1\text{ms}$  treten maximal ein PWM-Interrupt pro TT auf.

### 6.3. Tasks des Schedulers

In der Tabelle 6.2 werden alle Tasks der ASR mit Periode und Ausführungszeit aufgelistet. Die Ausführungszeiten der Tasks können durch die Einstellung einer Compileroptimierungsstufe verkürzt werden. Für die Messungen liegt der Code in der normalen Form ohne Optimierung vor.

Task	Periode [ms]	$T_{task}$ [ $\mu\text{s}$ ]
task_set_PWM	17,0	2,9
task_analyse_PWM	17,0	1,5
task_get_Sensordata	1,3	16,4
task_analyse_Sensordata	1,3	28,1
task_calculate_Slip_Acc	1,3	14,7
task_ASR_intervention	1,3	1,0

Tabelle 6.2.: Tasks des Schedulers

Der TT wird auf  $125\mu\text{s}$  (siehe Tabelle 6.1) eingestellt. Es wird pro TT eine Task ausgeführt und sie werden nacheinander zyklisch abgearbeitet. Damit ergibt sich für jede Task eine Periode von:  $6 * 125\mu\text{s} = 750\mu\text{s}$ . Damit sind die Vorgaben aus der Tabelle 6.2 erfüllt und bilden eine Lösung für das Scheduling.

Laut der Tabelle 6.1 kann während eines TT mit  $125\mu\text{s}$  ein Interrupt ( $N = 1$ ) auftreten. Für eine einwandfreie Funktion des Schedulers, muss die Gleichung 5.1 erfüllt werden (vgl. Kapitel 5.3).

Die Bearbeitungsdauer für den Scheduler beträgt  $T_{scheduler} = 11,9\mu\text{s}$  und für die ISR des PWM-Empfängersignals  $T_{interrupt} = 3,6\mu\text{s}$ . In die Gleichung wird die Task eingesetzt, die die längste Ausführungszeit aufweist. Wenn diese Task (*task\_analyse\_Sensordata*) die Gleichung erfüllt, wird die Gleichung von den anderen Tasks automatisch auch erfüllt.

$$TT > (T_{scheduler} + T_{task} + N * T_{interrupt})$$

$$125\mu s > (11,9\mu s + 28,1\mu s + 1 * 3,6\mu s)$$

$$(125\mu s > 43,6\mu s) \Rightarrow true$$

Eine andere Einstellung des TT oder auch die Optimierung des Codes durch eine andere Compileroptimierungsstufe ist im Moment nicht erforderlich. Eine Optimierung des Codes kann am Ende geschehen, wenn die Implementierung und die Tests abgeschlossen sind. Andernfalls ist das zeilenweise Debuggen im Code nicht möglich.

## 6.4. verwendete Datenstrukturen

Alle ermittelten Sensormesswerte werden in einem **struct** *truck\_status* gesammelt (siehe Abbildung 6.3). Für jeden Hall-Sensor wird die Zeit zwischen zwei Zähnen des Inkrementgebers über die Captureregister des Timers zwei und drei berechnet (vgl. Kapitel 5.3) und in *truck\_status.time\_XX* abgelegt. In der ISR (*ISR\_External\_PWM*) wird der Timer eins benutzt, um den High-Pegel des PWM-Signals zu messen (vgl. Kapitel 5.3). Die ermittelte Zeit wird in *truck\_status.pwm\_time\_high* abgelegt. Die beiden **enums** geben die Stellung des Servos anhand des PWM-Signals und den Zustand der ASR wieder (siehe Abbildung 6.2).

<pre>enum pwm_states{   PWM_MAX,   PWM_MID,   PWM_MIN,   PWM_UNDEF };</pre>	<pre>enum asr_states{   ASR_NORMAL,   ASR_EMERGENCY,   ASR_SAFE,   ASR_ERROR_STOP,   ASR_UNDEF };</pre>
---	---

Abbildung 6.2.: enum *pwm\_states* und *asr\_states*

Weitere Größen werden anhand der vorangegangenen Werte berechnet und in *truck\_status.[slip, acceleration, speed\_XX]* abgelegt.

```

struct truck_status {
    enum pwm_states  pwm_state;    uint32_t  time_FL;
    enum asr_states  asr_state;    uint32_t  time_FR;
                                         uint32_t  time_RL;
    uint8_t  slip;                uint32_t  time_RR;
    uint8_t  acceleration;
    uint32_t  pwm_time_high;      uint32_t  speed_FL;
                                         uint32_t  speed_FR;
                                         uint32_t  speed_RL;
                                         uint32_t  speed_RR;
};

```

Abbildung 6.3.: struct truck\_status

## 6.5. Filterung der Sensormesswerte und Berechnung weiterer Größen

Um fehlerhafte Sensormesswerte zu behandeln, wird, wie in Kapitel 5.5 beschrieben, ein Moving-Average-Filter eingesetzt. Dieses ist in der Task (*task\_get\_Sensordata*) umgesetzt. Die Filterlänge wird anfangs auf  $N = 3$  festgelegt.

Der Plausibilitätstest der Sensormesswerte wird in der Task (*task\_analyse\_Sensordata*) durchgeführt. Dieser berechnet mit den Kenndaten des intelliTrucks (vgl. Tabelle 4.5) die physikalisch maximal mögliche Geschwindigkeit zur letzten Messung und vergleicht sie mit der neu ermittelten Geschwindigkeit der aktuellen Sensormesswerte (vgl. Kapitel 5.5).

Anhand der ermittelten Sensormesswerte wird die Umfangsgeschwindigkeit von jedem Rad, die Beschleunigung des Fahrzeuges und der Schlupf bestimmt. Der ARM-Mikrocontroller besitzt keine Floating-Point-Unit (FPU), sodass keine Fließkommazahloperationen mit hoher Genauigkeit durchgeführt werden können. Um dennoch Berechnungen durchführen zu können, die brauchbare Ergebnisse liefern, wird die Formel 4.9, wie in Abbildung 6.4 gezeigt, mittels Integerarithmetik umgesetzt.

```

else
    truck.speed_FR = (MS2KMH * 10) * ((ROUND*100000000) / (
        (truck.time_FR/1000) * (((TEETH * 1000) / (pclk/1000000)))));

```

Abbildung 6.4.: Codeauszug zur Integerarithmetik

## 6.6. Funktionstest der Software

Die Funktion der Software wird auf einem Versuchsaufbau (vgl. Abbildung 6.5) in begrenztem Umfang und unter Idealbedingungen getestet, da die Hardware auf dem intelliTruck aufgrund aktueller Umbauarbeiten nicht einsatzbereit ist. Die eigentlichen Hall-Signale der Antriebs-

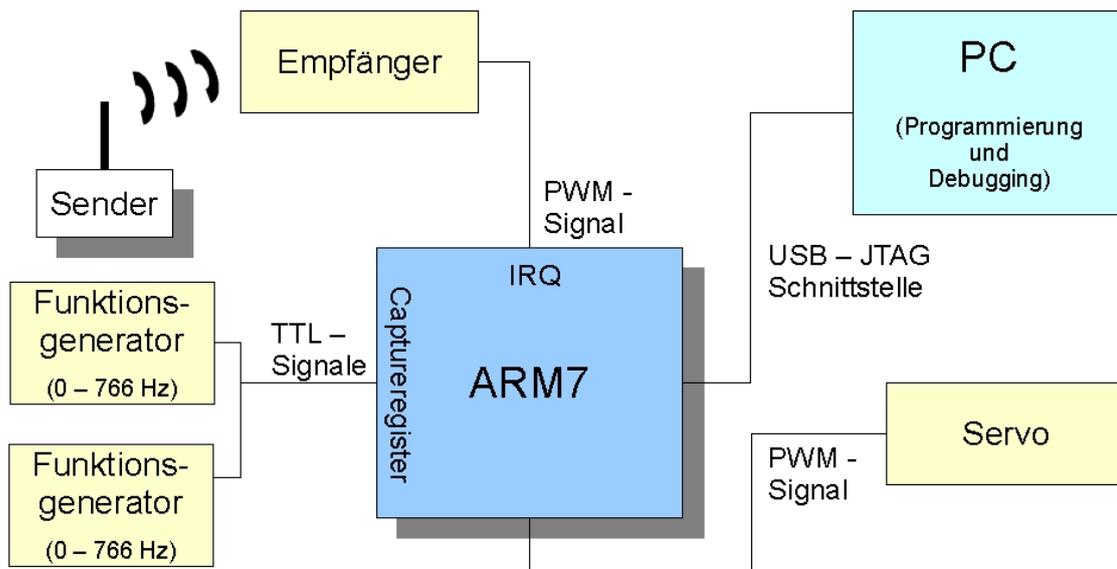


Abbildung 6.5.: Testaufbau

sowie der Vorderräder werden in Form von TTL-Signalen<sup>2</sup> zweier Funktionsgeneratoren erzeugt und an die entsprechenden Captureeingänge des ARM-Mikrocontrollers angeschlossen. Die beiden Vorderräder erhalten das selbe Signal des einen Funktionsgenerators und die Antriebsräder die des anderen. Damit ist die Umfangsgeschwindigkeit der Vorderräder identisch und die der Antriebsräder auch. Über die Frequenzeinstellungen an den Funktionsgeneratoren wird der Schlupf beeinflusst und damit zwischen den beiden Zuständen (NORMAL und EMERGENCY) der ASR (vgl. Abbildung 5.5) gewechselt. Anschließend werden die drei unterschiedlichen Fahrbefehle (bremsen, neutral und beschleunigen) über die Fernbedienung an den Empfänger gesendet und die ausgehenden PWM-Steuersignale auf dem Servo nach der Tabelle 5.4 verglichen und kontrolliert. Dieser Test verläuft erfolgreich. Somit ist die Funktion der Antriebsschlupfregelung mit einer festgelegten Schlupfgrenze ( $\lambda_g$ ), mit Gewährleistung der Bremsfunktionalität, in seiner Funktion verifiziert.

<sup>2</sup>Transistor-Transistor-Logik

## 6.7. Auslastung und Reserven des ARM-Mikrocontrollers

In der Abbildung 6.6 wird der Ablauf des Schedulers bezogen auf die Ausführung der Tasks (vgl. Tabelle 6.2) auf der Zeitachse aufgetragen. Dabei symbolisieren die blauhinterlegten Balken die Rechenzeit in jedem TT. Die Rechenzeit setzt sich wie folgt zusammen:

$$T_{\text{rechenzeit}} = T_{\text{scheduler}} + T_{\text{interrupt}} + T_{\text{task}} \quad (6.1)$$

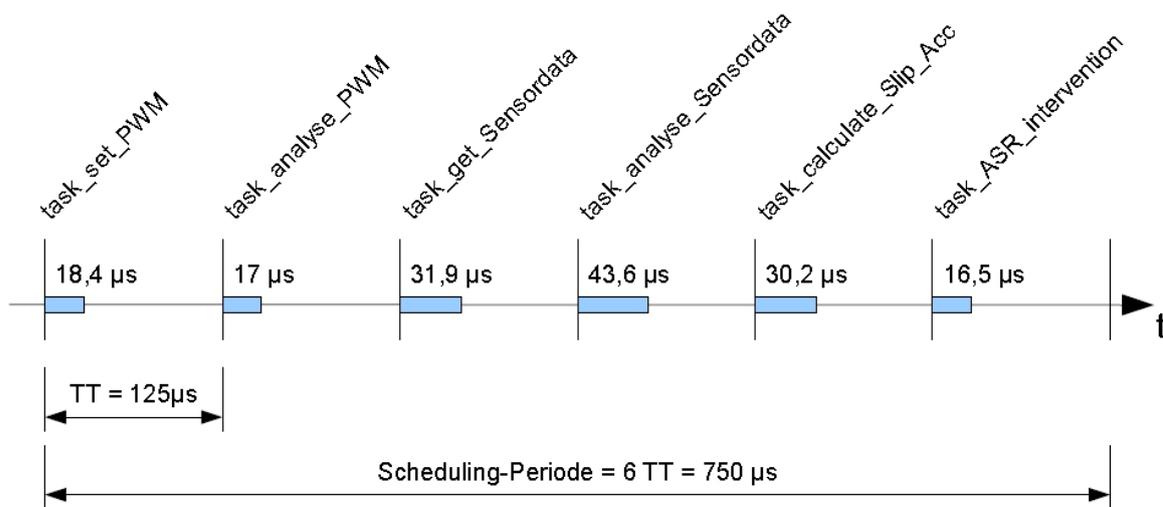


Abbildung 6.6.: Auslastung des ARM-Mikrocontrollers

$$\begin{aligned}
 \text{Auslastung}[\%] &= 100 * \frac{\text{Bearbeitungszeit}}{\text{Scheduling-Periode}} \quad (6.2) \\
 &= 100 * \frac{157,6 \mu s}{750 \mu s} = \mathbf{21\%}
 \end{aligned}$$

Die Auslastung des ARM-Mikrocontrollers liegt bei 21% und somit bietet er Reserven für Erweiterungen des Projektes bzw. Integration von weiteren Modulen wie z. B. eine Kommunikationsschnittstelle (USB, CAN).

Dabei ist es wichtig zu wissen, dass diese Analyse und Berechnung auf dem aktuellen Stand des Codes durchgeführt wird. Die Auslastung des ARM-Mikrocontrollers lässt sich weiter optimieren, indem eine Codeoptimierungsstufe im Compiler genutzt wird. Auch durch anders

---

gewählte TTs und Scheduling kann die Auslastung beeinflusst werden. Im jetzigen Scheduling werden z. B. die *task\_set\_PWM* und *task\_analyse\_PWM* je 22 mal in der Zeit ausgeführt, in der je eine Ausführung ausreichend wäre. Die Mehrfachausführungen bewirken keine Veränderungen am PWM-Steuersignal, da die beiden Tasks innerhalb der  $17\text{ms}$  immer identische Ergebnisse liefern.

## 7. Zusammenfassung

Ziel dieser Bachelorarbeit ist die Entwicklung einer für den intelliTruck speziell ausgerichteten ASR auf einem ARM-Mikrocontroller. Dabei werden Eigenschaften und physikalische Grundlagen für eine verbesserte Fahrzeugstabilität während der Beschleunigung untersucht. Die Herausforderung liegt in der Erfassung des zu regelnden Systems, der Gewährleistung der harten Echtzeit und dem vorhersagbaren Verhalten der ASR auf einer ARM7-Mikrocontrollersystemarchitektur.

Nach dem Grundlagenkapitel wird die ASR als solches in seiner Funktion im Fahrzeug unter den Gesichtspunkten der Sicherheit und des Antriebes beschrieben. Im Weiteren zeigt ein Überblick der verschiedenen ASR-Typen das Zusammenspiel und die Abhängigkeiten zu aufbauenden (z. B. ESP) und „untergeordneten“ Systemen wie dem ABS auf.

Die Analyse beinhaltet die Betrachtung auftretender Kräfte und Geschwindigkeiten am Reifen und am Fahrzeug. Diese werden in einen für die ASR relevanten Zusammenhang gebracht. Daraus ergeben sich in Bezug auf die Fahrbahnbeschaffenheit, der Radzustände (Haft- und Gleitreibung), der Beschleunigung (Umfangsgeschwindigkeit der Antriebsräder) und der abgeleiteten  $\mu$ -Schlupf-Kennlinie Regelgrößen und Eingriffspunkte der ASR. Zur Analyse gehört zudem die Ermittlung von Kenngrößen des intelliTrucks und der charakteristischen Eigenschaften der verfügbaren Hardware (Sensorik und Aktorik). Die Ermittlung der Kenngrößen des intelliTrucks erfordert einen umfangreichen Test. Dabei werden ein Testaufbau, Testszenarien, sowie eine Auswertung und ein Auswertungstool, das eigens zur Aufbereitung der erfassten Testdaten geschrieben wurde, entwickelt. Das Kapitel zeigt zudem Lösungsstrategien auf, um auftretende Fehler(quellen) behandeln und handhaben zu können.

Die Ergebnisse der Analyse dienen als Grundlage für das Design. Dabei werden die physikalischen Grundlagen und Zusammenhänge aufgegriffen und ein Regler, der einen Zustandsautomaten enthält, konzipiert. Anhand der ermittelten Kenngrößen, der Signalverläufe (z. B. Frequenz) der Sensoren und der aufgeführten Anforderungen ist eine Softwaresystemarchitektur für den ARM-Mikrocontroller entworfen worden. Diese erfüllt die harten Echtzeitanforderungen sowie die Vorhersagbarkeit.

Im Implementierungsteil wird im Detail auf das entworfene Design eingegangen. Der umgesetzte Scheduler, ein Kooperativer mit einem zusätzlichen Interrupt, wird mit seinen Funktio-

nen und Parametern sowie den verwendeten Tasks in ihren Aufgaben und Ausführungszeiten beschrieben und aufgeführt. Wichtige Enums und Datenstrukturen werden zusammen mit der umgesetzten Fehlerhandhabung erläutert. Zusätzlich wird die Auslastung des ARM-Mikrocontrollers betrachtet und beurteilt. Nach der Erstellung des ASR-Prototypen wird ein Labortestaufbau entworfen, der alle Sensorsignale und Aktoren nachbildet bzw. dem Testaufbau physisch zur Verfügung stellt. Zur Bestätigung der Anforderungen verifiziert ein Test die Funktionen des ASR-Prototypen unter Idealbedingungen.

Eine Integration der ASR auf dem intelliTruck war nicht möglich, da Umbauarbeiten an den Achsen und den Hall-Sensoren erst kurz vor Ende dieser Arbeit fertiggestellt wurden. Der intelliTruck ist nun mit neuen, gekapselten Hall-Sensoren mit eigens dafür entwickelten Platinen und mit neu gestalteten Inkrementgebern (Felgenmitnehmer mit 24 Zähnen) ausgestattet. Dadurch sind die Hall-Sensoren gut geschützt, leichter austauschbar und die Verkabelung einheitlich und gebündelt. Die Kapselung bietet noch den großen Vorteil, dass weniger Störeinflüsse von außen auf den Sensor wirken. Auch der Abstand zwischen Hall-Sensor und Inkrementgeber bleibt durch die Bauweise stets konstant. Erste Messungen der neu gestalteten Hall-Sensoren boten ein gutes Bild und zeigten keine Störungen bzw. Auffälligkeiten. Eine Integration der ASR auf dem intelliTruck sollte nun durchführbar sein.

Der Sicherheitsaspekt kann in diesem Projekt weiterentwickelt werden. Z.B. kann der Ausfall von Sensoren, ARM-Mikrocontroller etc. bemerkt und Gegenmaßnahmen eingeleitet werden. Weiter wird es möglich sein auf der ASR aufzusetzen und weitere Regelungen und Systeme wie z. B. einen Tempomaten zu entwickeln.

# Literaturverzeichnis

- [Antsaklis 2000] ANTSAKLIS, P.J.: A brief introduction to the theory and applications of hybrid systems. In: *879-887 IEEE* 88 (2000), Juli, Nr. 7, S. 879–887
- [Bauer 1998] BAUER, Horst: *Fahrsicherheitssysteme 2. Auflage*. Vieweg, 1998. – ISBN 3528038756
- [Bauer 2002] BAUER, Horst: *Kraftfahrtechnisches Taschenbuch 24. Auflage*. Vieweg, 2002. – ISBN 3528138769
- [Blal 2007] BLAL, Bachir: *Design und Implementation eines Konzeptes zur Geschwindigkeitsbestimmung eines autonomen Fahrzeugs unter Verwendung eines AVR Mikrocontrollers in Kombination mit Hall-Sensorik*, Hochschule für Angewandte Wissenschaften Hamburg, Diplomarbeit, 2007. – URL [http://www.informatik.haw-hamburg.de/fileadmin/faust\\_upload/Arbeiten/2007Ba\\_-\\_Bachir\\_Blal\\_-\\_...\\_Geschwindigkeitsregelung\\_eines\\_autonomen\\_Fahrzeugs\\_unter\\_Verwendung\\_eines\\_AVR-Mikrocontrollers\\_in\\_Kombination\\_mit\\_Hall-Sensorik.pdf](http://www.informatik.haw-hamburg.de/fileadmin/faust_upload/Arbeiten/2007Ba_-_Bachir_Blal_-_..._Geschwindigkeitsregelung_eines_autonomen_Fahrzeugs_unter_Verwendung_eines_AVR-Mikrocontrollers_in_Kombination_mit_Hall-Sensorik.pdf)
- [Borrelli u. a. 2006] BORRELLI, Francesco ; BEMPORAD, Alberto ; FODOR, Michael ; HROVAT, Davor: An MPC/Hybrid System Approach to Traction Control. In: *1063-6536 IEEE* 14 (2006), Mai, Nr. 3, S. 541–552
- [Braess und Seiffert 2003] BRAESS, Hans-Hermann ; SEIFFERT, Ulrich: *Handbuch Kraftfahrzeugtechnik*. Vieweg, 2003. – ISBN 3528231149
- [Breuer und Bill 2004] BREUER, Bert ; BILL, Karlheinz H.: *Bremsenhandbuch (Grundlagen, Komponenten, Systeme, Fahrdynamik)*. Vieweg, 2004. – ISBN 3528139528
- [Choi und Hong 2002] CHOI, Hyoun-Chul ; HONG, Suk-Kyo: Hybrid Control for Longitudinal Speed and Traction of Vehicles. In: *0-7803-7474-6 IEEE* (2002), Februar, S. 1675–1680
- [Drakunov u. a. 2000] DRAKUNOV, Sergey V. ; ASHRAFI, Behrouz ; ROSIGLIONI, Alessandro: Yaw Control Algorithm via Sliding Mode Control. In: *0-7803-5519-9 AACC* (2000), Juni, S. 580–583
- [Gerigk und Bruhn 1991] GERIGK ; BRUHN: *Kraftfahrzeugtechnik*. 1991. – ISBN 314221500

- [Gevatter 2000] GEVATTER, Hans-Jürgen: *Automatisierungstechnik 2*. Springer, Berlin, 2000. – ISBN 3540670858
- [Glatz 2008] GLATZ, Nicolai: *Möglichkeiten eines Radartransceivers zur Hinderniserkennung an einem autonomen Fahrzeug*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2008. – URL [http://www.informatik.haw-hamburg.de/fileadmin/faust\\_upload/Arbeiten/2008Ba\\_-\\_Nicolai\\_Glatz\\_Moeglichkeiten\\_eines\\_Radartransceivers\\_zur\\_Hinderniserkennung\\_an\\_einem\\_autonomen\\_Fahrzeug.pdf](http://www.informatik.haw-hamburg.de/fileadmin/faust_upload/Arbeiten/2008Ba_-_Nicolai_Glatz_Moeglichkeiten_eines_Radartransceivers_zur_Hinderniserkennung_an_einem_autonomen_Fahrzeug.pdf)
- [Heise 2008] HEISE: Volvo-Sicherheitssysteme zur Unfallprävention Hauptziel: Kollisionen vermeiden und Unfallfolgen minimieren. In: *heise Autos* (2008), Mai. – URL <http://www.heise.de/autos/artikel/s/5710>
- [Hensel 2007] HENSEL, Enrico: *Design und Implementation eines Sicherheitskonzepts für den Betrieb eines autonomen Fahrzeuges*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2007. – URL [http://www.informatik.haw-hamburg.de/fileadmin/faust\\_upload/Arbeiten/2007Ba\\_-\\_Enrico\\_Hensel\\_-\\_Design\\_und\\_Implementation\\_eines\\_Sicherheitskonzepts\\_fuer\\_den\\_Betrieb\\_eines\\_autonomen\\_Fahrzeuges.pdf](http://www.informatik.haw-hamburg.de/fileadmin/faust_upload/Arbeiten/2007Ba_-_Enrico_Hensel_-_Design_und_Implementation_eines_Sicherheitskonzepts_fuer_den_Betrieb_eines_autonomen_Fahrzeuges.pdf)
- [Hruschka und Rupp 2002] HRUSCHKA, Peter ; RUPP, Chris: *Agile Softwareentwicklung (für Embedded Real-Time Systems mit der UML)*. Hanser, 2002. – ISBN 3446219978
- [Isermann 2006] ISERMANN, Rolf: *Fahrdynamik-Regelung*. Vieweg, 2006. – ISBN 3834801097
- [Justen und Baumann 1999] JUSTEN, R. ; BAUMANN, K.-H.: Pre-crash-Erkennung, ein neuer Weg in der PKW-Sicherheit. In: *VDI Berichte NR. 1471* (1999), S. 361–379. ISBN 3-18-091471-8
- [Kiencke und Jäkel 2005] KIENCKE, Uwe ; JÄKEL, Holger: *Signale und Systeme*. Oldenburg, 2005. – ISBN 3486578111
- [Kost und Post 2004] KOST, Friedrich ; POST, Wulf: *Sicherheits- und Komfortsysteme*. Vieweg, 2004. – ISBN 3528138750
- [Kruse voraussichtlich 2008] KRUSE, Nils: *Kameragestützte Fahrspurerkennung für autonome Modellfahrzeuge*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, voraussichtlich 2008
- [Kuchling 2001] KUCHLING, Horst: *Taschenbuch der Physik*. 2001. – ISBN 3446217606
- [Mencher u. a. 2007] MENCHER, Bernhard ; REITER, Ferdinand ; ...: *Autoelektrik Autoelektronik*. Vieweg, 2007. – ISBN 3528238720

- [Nar voraussichtlich 2008] NAR, Umut: *Entwicklung eines Sensors zur Geschwindigkeitsmessung eines Modellfahrzeugs*, Hochschule für Angewandte Wissenschaften Hamburg, Diplomarbeit, voraussichtlich 2008
- [NXP 2007] NXP, Semiconductors: *UM10237 LPC24xx User manual*. URL <http://www.standardics.nxp.com/support/documents/microcontrollers/pdf/user.manual.lpc24xx.pdf>, 2007
- [Oberhauser 2003] OBERHAUSER, Mathias: *Mechatronische Getriebesysteme : Mechatronik und Design moderner Kfz-Getriebe*. Renningen-Malmsheim : expert-Verl, 2003. – ISBN 3816922392
- [Oswald 1940] OSWALD, Fritz: *Die Diplomarbeit konnte leider nicht ermittelt werden*, Diplomarbeit, 1940
- [Pein 2007] PEIN, Collin: *Geschwindigkeitsregelung mittels eines Beschleunigungssensors im autonomen Modellfahrzeug*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2007. – URL [http://www.informatik.haw-hamburg.de/fileadmin/faust\\_upload/Arbeiten/2007Ba\\_-\\_Collin\\_Pein\\_-\\_Geschwindigkeitsregelung\\_mittels\\_eines\\_Beschleunigungssensors\\_im\\_autonomen\\_Modellfahrzeug.pdf](http://www.informatik.haw-hamburg.de/fileadmin/faust_upload/Arbeiten/2007Ba_-_Collin_Pein_-_Geschwindigkeitsregelung_mittels_eines_Beschleunigungssensors_im_autonomen_Modellfahrzeug.pdf)
- [Pont 2001] PONT, Michael J.: *Pattern for time-triggered embedded systems : Building reliable applications with the 8051 family of microcontrollers*. ACM Press, 2001. – ISBN 0201331381
- [Prinz und Crawford 2006] PRINZ, Peter ; CRAWFORD, Tony: *C in a Nutshell*. O'Reilly, 2006. – ISBN 0596006977
- [Reichel 2001] REICHEL, Hans-Rolf: *Elektronische Bremssysteme, Vom ABS zum Brake-by-Wire*. Expert, Renningen, 2001. – ISBN 3816920101
- [Reif 2006] REIF, Konrad: *Automobilelektronik*. Vieweg, 2006. – ISBN 352803985X
- [Rull voraussichtlich 2008] RULL, Andrej: *Sensorbasierte Umgebungskartierung mit lokaler Positionskorrektur für autonome Fahrzeuge*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, voraussichtlich 2008
- [Schuckert 2007] SCHUCKERT, Simon M.: *Mikrocontrollerbasierte Telemetrie und Echtzeitauswertung von Sensordaten im Formula Student Rennwagen*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2007. – URL [http://www.informatik.haw-hamburg.de/fileadmin/faust\\_upload/Arbeiten/2007Ba\\_-\\_Simon\\_Schuckert\\_Mikrocontrollerbasierte\\_Telemetrie\\_und\\_Echtzeitauswertung\\_von\\_Sensordaten\\_im\\_Formula\\_Student\\_Rennwagen.pdf](http://www.informatik.haw-hamburg.de/fileadmin/faust_upload/Arbeiten/2007Ba_-_Simon_Schuckert_Mikrocontrollerbasierte_Telemetrie_und_Echtzeitauswertung_von_Sensordaten_im_Formula_Student_Rennwagen.pdf)

- [Shafranovich 2005] SHAFRANOVICH, Y.: Common Format and MIME Type for Comma-Separated Values (CSV) Files. In: *RFC 4180* (2005), S. 1–8
- [Siemers und Sikora 2003] SIEMERS, Christian ; SIKORA, Axel: *Taschenbuch Digitaltechnik*. 2003. – ISBN 3446218629
- [The Automatic Control Laboratory 2008] THE AUTOMATIC CONTROL LABORATORY, Schweiz: *HYSDEL Hybrid Systems Description Language*. August 2008. – URL <http://control.ee.ethz.ch/~hybrid/hysdel/>
- [Torrìsi und Bemporad 2004] TORRISI, Fabio D. ; BEMPORAD, Alberto: HYSDEL A Tool for Generating Computational Hybrid Models for Analysis and Synthesis Problems. In: *1063-6536 IEEE* 12 (2004), März, Nr. 2, S. 235–249
- [Unbehauen 2000] UNBEHAUEN, Hein: *Regelungstechnik 3: Identifikation, Adaption, Optimierung*. Braunschweig [u.a.] : Vieweg, 2000. – ISBN 3528533498

# A. Anhang

## A.1. PicoScope

Das PicoScope der 3000 Serie bietet als Datenlogger verschiedene Einstellmöglichkeiten. Dazu wird die mitgelieferte Software „PicoLog Recorder“ genutzt. Damit können je nach Einstellung Abtastzeiten von 5 ns mehreren Stunden genutzt werden.

Log-File Logging	als *.txt oder *.plw (internes Format) auch als Graph darstellbar
„kontinuierlich Echtzeit“	1 ms - mehrere Stunden Abtastzeit
„schneller Block“ Maximale Anzahl Messpunkte Größe des Log-Files	5 ns - 384 $\mu$ s Abtastzeit 1 Million (bei 20 $\mu$ s Abtastzeit) etwa 16 MB als *.txt (bei 20 $\mu$ s Abtastzeit)

Tabelle A.1.: wichtige Eigenschaften

Neben der Funktion des Datenloggens kann das PicoScope auch als ein PC-Oszilloskop genutzt werden, das zwei Kanäle als Eingänge bietet. Die abgetasteten Signale werden mit der mitgelieferten Software „PicoScope“ visualisiert. Durch die Triggermöglichkeiten und Standbilder können leicht Messungen in den Signalen Mittels einer eingebauten Messfunktion durchgeführt werden.

Das PicoScope kann über den externen Ausgang (EXT) auch als Funktionsgenerator genutzt werden. Die Besonderheit liegt in dem integrierten „Sweep Modus“, damit lassen sich Änderungen auf das Ausgangssignal vornehmen. In diesem Modus kann die Frequenz zu bestimmten Zeitintervallen erhöht werden und somit kontinuierliche Frequenzhübe erzeugt werden. Damit lassen sich Hall-Signale erzeugen, mit denen eine Beschleunigung des intelligenten Trucks simuliert werden kann.

Für weitere Informationen wird auf das Handbuch und auf die Internetseite <sup>1</sup> des Herstellers verwiesen.

---

<sup>1</sup>[www.picotech.com](http://www.picotech.com)

## A.2. ARM - Einstellungen, PIN-Belegung und Anschlüsse auf dem I/O-Board

Die Grundeinstellungen, wie z.B. Taktfrequenz und GPIO-Modus, werden mit dem StartEasy-Tool von Hitex <sup>2</sup> vorgenommen. Das Tool erstellt nach den vorgenommenen Einstellungen das Projekt, das die wichtigsten Dateien enthält, wie z.B. dem Startupcode. Beim Linken muss darauf geachtet werden, dass neue Headerdateien nicht automatisch zum Linker hinzugefügt werden. Dies kann durch Bearbeiten der ([Projektname].ld) Datei im Projektordner (./Settings/) behoben werden. Eine andere Möglichkeit ist das erneute Starten des StartEasy-Tool, in dem die neuen Dateien angegeben werden. Beim Erstellen des Codes sollte darauf geachtet werden, dass keine Dateien durch das Tool ungewollt überschrieben werden.

Vorgenommene Einstellungen im StartEasy-Tool:

Kategorie	Einstellung
Clock Source	Main Oscillator (4 MHz)
PLL state	Enabled
Divider (N)	2
Multiplier (M)	72
CPU clock divider	4
USB clock divider	6
MAM Module	Partly enabled
GPIO-Modus für Port 0 und 1	Fast GPIO

Tabelle A.2.: StartEasy Einstellungen für den ARM

Diese Einstellungen ermöglichen den Betrieb des ARMs mit  $72MHz$  und die USB-Schnittstelle mit  $48MHz$ .

---

<sup>2</sup>[www.hitex.com/con-starteasy.html](http://www.hitex.com/con-starteasy.html)

Pin	Port	Pin	Port
1	GND	2	CAN BM
3	GND	4	CAN BP
5	+9V	6	P2.11
7	+9V	8	P2.12
9	+5V	10	P2.13
11	+5V	12	P2.9
13	+3.3V	14	P2.8
15	+3.3V	16	P2.5
17	GND	18	P2.4
19	GND	20	P2.3
21	P0.26	22	P2.2
23	P0.25	24	P2.1
25	P0.24 ( <b>CAP3.1</b> )	26	P2.0 ( <b>PWM1.1</b> )
27	P0.23 ( <b>CAP3.0</b> )	28	P0.5 ( <b>CAP2.1</b> )
29	P0.20	30	P0.4 ( <b>CAP2.0</b> )
31	P0.19	32	P0.22 ( <b>Interrupt<sup>3</sup></b> )
33	P0.18	34	P0.21
35	P0.17	36	P1.12
37	P0.16	38	P1.11
39	P0.15	40	P1.7
41	P1.20	42	P1.21
43	GND	44	Power_enable
45	GND	46	Reset
46	n.c	48	n.c

Tabelle A.3.: I/O-Board Anschlüsse

### A.3. Diagramme und Codeauszüge

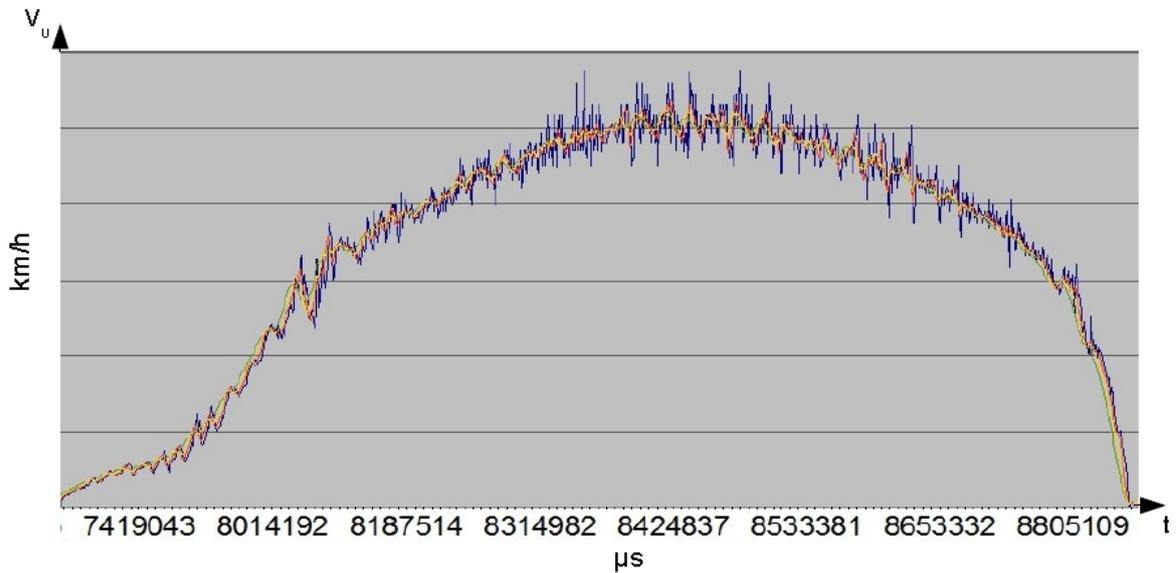


Abbildung A.1.: Beschleunigung bei 100 Prozent Schlupf

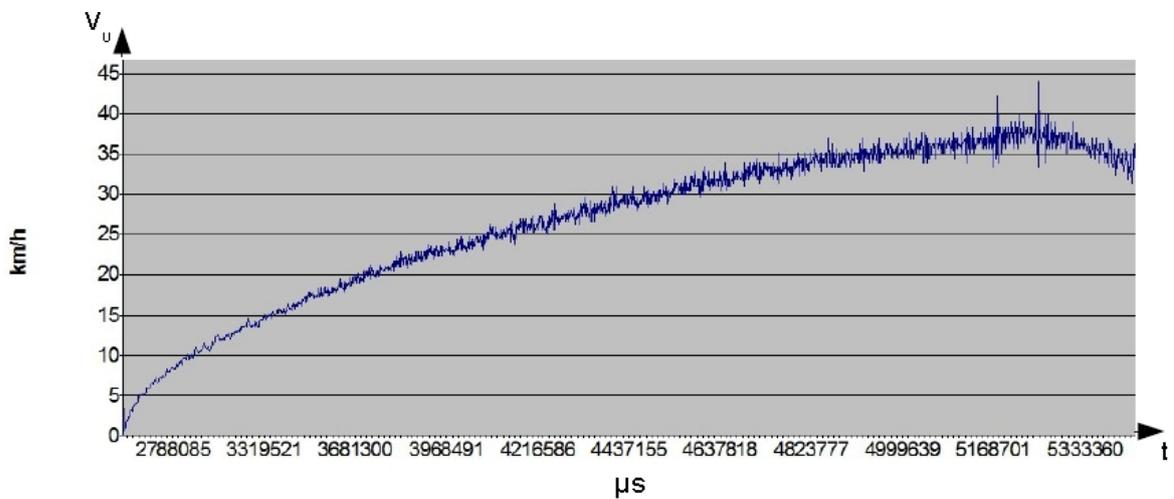


Abbildung A.2.: Beschleunigung maximal auf Asphalt

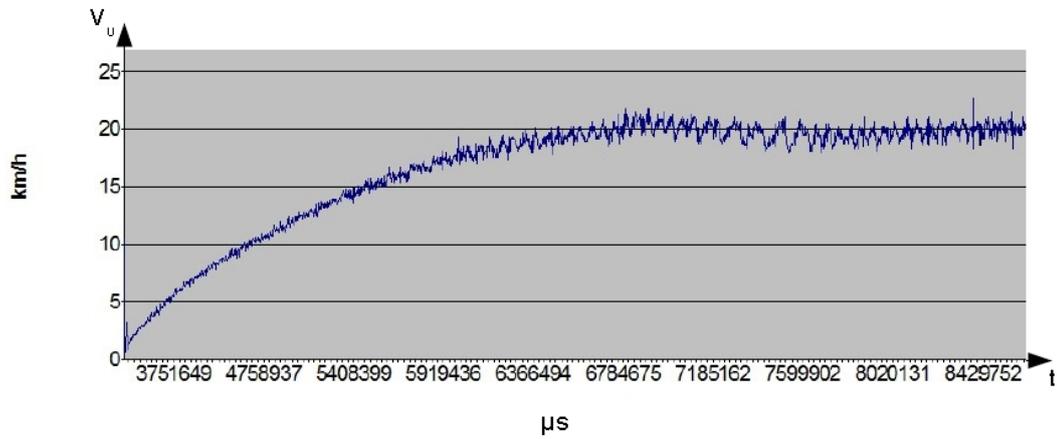


Abbildung A.3.: Beschleunigung „normal“ auf Asphalt

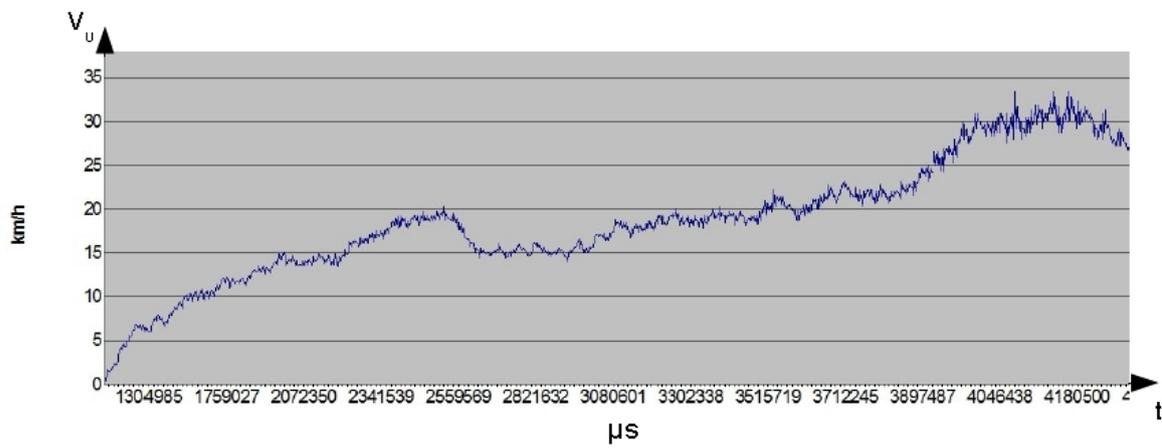


Abbildung A.4.: Beschleunigung auf Schotter

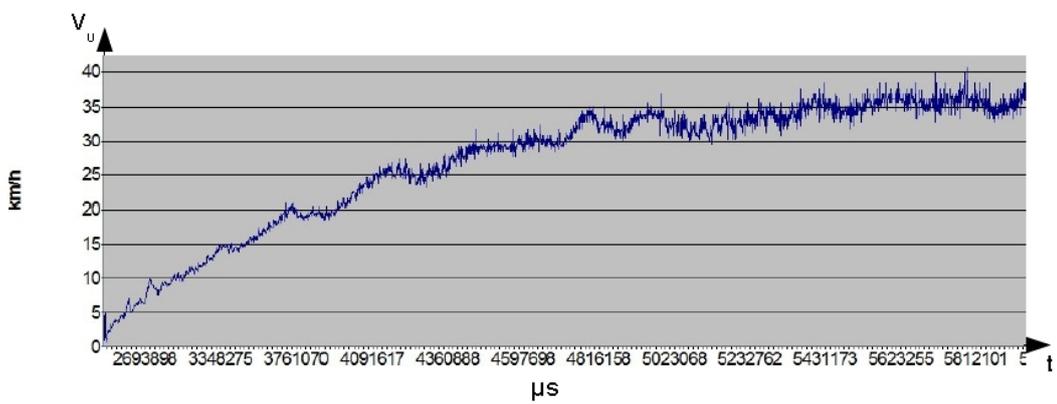


Abbildung A.5.: Beschleunigung auf Schotter 2

```

void ISR_Timer0 (void){
    SCH_Update();
    TIMER0_IR = 0x01;           // clear interrupt
    VICVectAddr = 0x00;
}

```

Abbildung A.6.: Interruptserviceroutine: ISR\_Timer0()

```

int main(void){                // Initialisieren
    init();
    SCH_Add_Task(task_set_PWM, 0, 6); // Einfuegen der Tasks mit Delay und Periode
    SCH_Add_Task(task_analyse_PWM, 1, 6);
    SCH_Add_Task(task_get_Sensordata, 2, 6);
    SCH_Add_Task(task_analyse_Sensordata, 3, 6);
    SCH_Add_Task(task_calculate_Slip_Acc, 4, 6);
    SCH_Add_Task(task_ASR_intervention, 5, 6);

    PWM_Start(1);              // PWM-Steuersignal und Scheduler starten
    SCH_Start();

    while(1){                  // Ausfuehren des Dispatchers
        SCH_Dispatch_Tasks();
    }
}

```

Abbildung A.7.: Funktion: main()

```

TIMER2_TCR = 0x01;           // Enable timer
SCH_Init();                  // SCHEDULER
TIMER0_MR0 = TIMER_CPM;     // Timer0 gibt Interrupt bei CompareMatch und startet neu
                             // Interrupt entspricht TimerTick fuer den Scheduler

```

Abbildung A.8.: Auszug aus der Funktion: init() für SCH\_init()

# Abkürzungsverzeichnis

- ASR** Antriebsschlupfregelung
- EDS** elektronische Differenzialsperre
- MA-Filter** Moving Average Filter
- ACC** Adaptive Cruise Control
- B-ASR** Brems-Antriebsschlupfregelung
- BAS** Bremsassistent
- CSV** Comma-Separated Values
- EBV** elektronische Bremskraftverteilung
- EMS** elektronische Motorleistungssteuerung
- ESP** elektronisches Stabilitätsprogramm
- GMR** Giermomentenregelung
- HYSDEL** Hybrid Systems Description Language
- M-ASR** Motor-Antriebsschlupfregelung
- MSR** Motorschleppmomentregelung
- TCS** Traction Control System
- TT** Timer Tick

# 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.

Hamburg, 19. August 2008

Ort, Datum

Unterschrift