



sim

FZD

FAHRZEUGTECHNIK
TU DARMSTADT

Technische Universität Darmstadt
Fachbereich 20 · Informatik
Fachgebiet Simulation und Systemoptimierung
Prof. Dr. rer. nat. Oskar von Stryk

Technische Universität Darmstadt
Fachbereich 16 · Maschinenbau
Fachgebiet Fahrzeugtechnik
Prof. Dr. rer. nat. Hermann Winner

Master Thesis
Nr. 414/06

Softwareentwicklung zur Messdatenerfassung und Echtzeit-Steuerung eines Reifenmessanhängers

Bearbeiter:
B. Sc. Marco Heimberger

Betreuer:
Dipl.-Ing. Marcus Reul

Darmstadt, den 23.04.2007

Ehrenwörtliche Erklärung

Hiermit versichere ich, die vorliegende Masterarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, April 2007

Marco Heimberger

Zusammenfassung

Der in den 90er Jahren erfolgreich eingesetzte Reifenmessanhänger PETRA I soll nach längerer Standzeit wieder in Betrieb genommen werden. PETRA I wird eingesetzt um Reibwert-Schlupf-Kurven unter realen Bedingungen zu ermitteln. Die Gründe für die Aktualisierungsmaßnahmen von PETRA I sind: Der Messanhänger soll autark und somit unabhängig vom Zugfahrzeug sein. PETRA I soll mit neuen Funktionen wie eine automatische Schwimmwinkelkompensation oder eine automatische Messungsdurchführung ausgestattet werden. Zudem soll PETRA I möglichst flexibel für spätere Projekte sein, daher wird eine möglichst hohe Abtastrate angestrebt. Da die zuvor verwendete Messtechnik, welche den kompletten Koffer- und Beifahrerraum benötigte, veraltet ist wurde zu Beginn dieser Arbeit PETRA I mit neuer Messtechnik und Erfassungselektronik ausgestattet. Hauptbestandteil dieser neuen Messtechnik ist das integrierte CompactRIO System von National Instruments. Dieses System, welches auf dem Messanhänger verbaut wird, enthält einen integrierten FPGA sowie ein Realtime-System. Im Zugfahrzeug des Anhängers wird sich ein Laptop befinden, der die Steuerung und Überwachung des Anhängers übernimmt. Für diese verteilte Architektur gilt es die Software zu entwickeln, welche die Messdatenerfassung und die Echtzeit-Steuerung von PETRA I ermöglicht. Dabei liegt die Herausforderung darin, die Software geschickt über die 3 Ebenen zu verteilen um eine hohe Performance des Gesamtsystems zu erreichen.

Bei der Softwareentwicklung für PETRA I wurde nach dem W-Modell vorgegangen. Für die Anforderungsanalyse wurde ein Brainstorming und ein Prototyp verwendet, beim Entwurf wurde die UML verwendet. Nach der Modulspezifikation wurden die bestbewerteten Module umgesetzt. Nach der Implementierung wurden die Module entsprechend der zuvor erstellten Modultests verifiziert. Die Integration erfolgte nach dem Bottom-Up Verfahren. Danach folgten die Integrationstests sowie der Systemtest, die jeweils anhand zuvor erstellter Testspezifikationen getestet wurden. Die Validierung erfolgte mittels Plausibilitätsprüfung von Reibwert-Schlupf-Kurven.

Zur Optimierung der Performance wurden die Messgrößen in zwei Kategorien aufgeteilt: „Echte“ Messgrößen die mit eingestellter Abtastrate erfasst werden und Kontrollgrößen, die zur Überwachung des Messbetriebs benötigt und mit geringerer Abtastrate erfasst werden. Diese Messgrößen werden mittels zweier separater DMA-Kanäle vom FPGA an das Realtime-System und von dort über TCP/IP an den Laptop übermittelt. Vom vorher verwendeten Messsystem zum jetzigen mit aktuell entwickelter Software wurde die Abtastrate von 100Hz auf 5000Hz verbessert bei 12 Kanälen. Außerdem erfolgt die Messungsdurchführung mit der neuen Software automatisch. Eine weitere Neuerung ist die automatische Schwimmwinkelkompensation die aus Performancegründen auf dem FPGA implementiert wurde und den sich bei der Messung einstellenden Schwimmwinkel ausregelt.

Inhaltsverzeichnis

Ehrenwörtliche Erklärung	II
Zusammenfassung	III
Inhaltsverzeichnis	IV
Formelzeichen und Abkürzungen	XI
1 Einleitung	1
1.1 Motivation	1
1.2 Konkretisierung der Aufgabenstellung	3
1.3 Klärung der Begriffe Verifikation und Validierung	3
1.4 Methodik des Vorgehens	4
2 Hardware des Messanhängers PETRA I	7
2.1 Aufbau	7
2.2 Aktorik	8
2.2.1 Analoge Aktorik	8
2.2.2 Digitale Aktorik	8
2.3 Sensorik	9
2.3.1 Analoge Sensorik	9
2.3.2 Digitale Sensorik	10
2.4 CRIO	11
2.4.1 Gehäuse CRIO-9104	11
2.4.2 Controller CRIO-9004	11
2.4.3 NI-Module	12
2.5 Laptop	13
3 LabVIEW	14
4 Anforderungsanalyse	17
4.1 Ziel der Anforderungsanalyse	17
4.2 Methodik der Anforderungsanalyse	17
4.3 Ablaufszenario	18
4.4 Darstellung der Anforderungen	20

5	Entwurf	23
5.1	Grobentwurf	23
5.1.1	Ziel des Grobentwurfs	23
5.1.2	Methodik des Grobentwurfs	23
5.1.3	Software- und Hardwarearchitektur	23
5.1.4	Schnittstellenbeschreibung	25
5.2	Feinentwurf	27
5.2.1	Ziel des Feinentwurfs	27
5.2.2	Methodik des Feinentwurfs	27
5.2.3	Feinentwurf des FPGA	27
5.2.4	Feinentwurf des Broker	29
5.2.5	Feinentwurf des Host	29
6	Modulspezifikation	32
6.1	Ziel der Modulspezifikation	32
6.2	Methodik der Modulspezifikation	32
6.3	Modulspezifikationen der FPGA – Module	33
6.3.1	Konfiguration	33
6.3.2	Schräglaufeinstellung	34
6.3.3	Schwimmwinkelkompensation	37
6.3.4	Drehzahlberechnung	40
6.3.5	Datenübertragung (HPL und NPL)	42
6.4	Modulspezifikationen der Broker – Module	44
6.4.1	Methodik und Vorgehen	44
6.4.2	Normal Priority Loop (NPL) – Modulspezifikation	45
6.4.3	Time Critical Loop (TCL) – Modulspezifikation	46
6.5	Modulspezifikationen der Host – Module	46
6.5.1	Hauptprogramm – Modulspezifikation	46
6.5.2	Einstellungen – Modulspezifikation	47
6.5.3	Kalibrierung – Modulspezifikation	48
6.5.4	R_{dyn} -Berechnung – Modulspezifikation	50
6.5.5	Messung – Modulspezifikation	51
6.5.6	Vorauswertung – Modulspezifikation	53
6.5.7	Statusanzeigen – Modulspezifikation	54

6.5.8	Warnung – Modulspezifikation	55
7	Programmierung	56
7.1	FPGA-Programmierung	56
7.1.1	Konfigurationsmodul	56
7.1.2	Mittelwertbildung	58
7.1.3	Schräglaufeinstellungsmodul	58
7.1.4	Schwimmwinkelkompensationsmodul	59
7.1.5	Drehzahlberechnungsmodul	60
7.1.6	LPL – Modul	61
7.1.7	HPL – Modul	64
7.2	Broker-Programmierung	64
7.2.1	Normal Priority Loop – Modul	64
7.2.2	Time Critical Loop – Modul	67
7.3	Host-Programmierung	75
7.3.1	Hauptprogramm	75
7.3.2	Konfiguration	77
7.3.3	Kalibrierung	77
7.3.4	Netzwerkverbindung mit CRIO herstellen	79
7.3.5	Netzwerkkommunikation	79
7.3.6	Frequenzabhängige Timeranpassung	79
7.3.7	R_{dyn} -Berechnung	80
7.3.8	Messung	81
7.3.9	Statusanzeigen	83
7.3.10	Warnung	83
8	Optimierungsmaßnahmen	85
8.1	DMA statt IRQ	85
8.2	Messgrößenaufteilung	85
8.3	LabVIEW 8.20 statt 8.01	85
8.4	Berechnungen auf dem RT-System minimiert	85
8.5	Timeroptimierung	85
8.6	Paketgrößenerhöhung	87

9	Verifikation	88
9.1	Testen	88
9.2	Modultests	89
9.2.1	Ziel des Modultests	89
9.2.2	Modultest: FPGA – Schräglaufeinstellung	89
9.2.3	Modultest: FPGA – Schwimmwinkelkompensation	90
9.2.4	Modultest: FPGA – Drehzahlmodul	90
9.2.5	Modultest: FPGA – Low Priority Loop	91
9.2.6	Modultest: FPGA – High Priority Loop	91
9.2.7	Modultest: Broker – NPL + TCL	92
9.2.8	Modultest: Host – Hauptprogramm	92
9.2.9	Modultest: Host – Einstellungen	92
9.2.10	Modultest: Host – Kalibrierung	92
9.2.11	Modultest: Host – R_{dyn} -Berechnung	93
9.2.12	Modultest: Host – Messung	93
9.2.13	Modultest: Host – Statusanzeigen	93
9.2.14	Modultest: Host – Warnung	94
9.3	Integrationstests	94
9.3.1	Integrationsmethodik	94
9.3.2	Ziel des Integrationstests	98
9.3.3	Integrationstest: FPGA - Priority Loops	98
9.3.4	Integrationstest: FPGA	99
9.3.5	Integrationstest: Broker	99
9.3.6	Integrationstest: Host	100
9.4	Systemtest	100
9.4.1	Ziel des Systemtests	100
9.4.2	Methodik des Systemtests	100
9.4.3	Fehlerbetrachtung der Messtechnik	101
9.4.4	Vorbedingungen des Systemtests „Stand“	101
9.4.5	Systemtest „Stand“	102
9.4.6	Vorbedingungen des Systemtests „Fahrt“	103
9.4.7	Systemtest „Fahrt“	107

10	Validierung	119
10.1	Ziel der Validierung	119
10.2	Validierungsvorschrift	119
10.3	Vorbedingungen der Validierung	120
10.3.1	Stufenanpassung	121
10.4	Durchführung der Validierung	121
10.5	Auswertung der Validierung	122
10.5.1	Methodik der Auswertung	122
10.5.2	Auswertung der Messdaten	124
11	Fazit und Ausblick	131
12	Literaturverzeichnis	134
13	Abbildungsverzeichnis	137
14	Tabellenverzeichnis	140
Anhang A	Anforderungsdefinition	141
A1.	Funktionale Anforderungen	141
A2.	Nichtfunktionale Anforderungen	153
Anhang B	Ein- und Ausgängeübersicht	156
B1.	Analogeingänge NI-9205	156
B2.	Analogausgänge NI-9263	157
B3.	Digitaleingänge NI-9401	157
B4.	Digitalausgänge NI-9472	158
Anhang C	Programmcodeübersicht	159
C1.	FPGA-Übersicht	159
C2.	Broker NPL – Übersicht	160
C3.	Broker TLC – Übersicht	161
C4.	Host – Übersicht	162
Anhang D	Programmcodefragmente	163
D1.	Host Programmcodefragmente	163
D1.1.	Konfiguration	163
D1.2.	Kalibrierung	164
D1.3.	CRIO suchen und reseten	165

D1.4.	Netzwerkkommunikation und Hauptfunktionen	166
Anhang E	Modultests	167
E1.	Modultests der FPGA-Module	167
E1.1.	Schräglaufeinstellung	167
E1.2.	Schwimmwinkelkompensation	170
E1.3.	Drehzahlberechnung	174
E1.4.	Low Priority Loop	176
E1.5.	High Priority Loop	177
E2.	Modultests der Host-Module	180
E2.1.	Einstellungen	180
E2.2.	R_{dyn} -Berechnung	187
E2.3.	Messung	189
E2.4.	Statusanzeigen	190
E2.5.	Warnung	195
Anhang F	Integrationstests	201
F1.	FPGA - Priority Loop	201
F1.1.	Integrationstestvorschrift: FPGA – Priority Loop	201
F1.2.	Integrationstestprotokoll: FPGA – Priority Loop	202
F2.	FPGA	204
F2.1.	Integrationstestvorschrift: FPGA	204
F2.2.	Integrationstestprotokoll: FPGA	206
F3.	Broker	209
F3.1.	Integrationstestvorschrift: Broker	209
F3.2.	Integrationstestprotokoll: Broker	210
F4.	Erweiterte Anzeige	212
F4.1.	Integrationstestvorschrift: erweiterte Anzeige	212
F4.2.	Integrationstestprotokoll: erweiterte Anzeige	213
F5.	Host - Send	214
F5.1.	Integrationstestvorschrift: Host – Send	214
F5.2.	Integrationstestprotokoll: Host – Send	216
F6.	Erweiterte R_{dyn}-Berechnung	218
F6.1.	Integrationstestvorschrift: erweiterte R_{dyn} -Berechnung	218
F6.2.	Integrationstestprotokoll: R_{dyn} -Berechnung	219

F7.	Erweiterte Messung	220
F7.1.	Integrationstestvorschrift: erweiterte Messung	220
F7.2.	Integrationstestprotokoll: erweiterte Messung	221
Anhang G	Systemtests	222
G1.	Systemtestvorschrift des Systemtests „Stand“	222
G2.	Systemtestprotokoll des Systemtests „Stand“	223
G3.	Correvitkalibrierung	225
G4.	Systemtestvorschrift des Systemtests „Fahrt“	228
G5.	Systemtestprotokoll des Systemtests „Fahrt“	233
Anhang H	Validierung	247
H1.	Versuchstabelle (Parametervariationsliste)	247
H2.	Validierungsprotokoll	248
H3.	Methodik der Auswertung	249
H3.1.	Brainstorming	249
H3.2.	Bewertungsmatrix	250
H3.3.	Bewertung der Auswertemöglichkeiten	251
H3.4.	Bewertung der Kriterien zur Detektion von stationären Zuständen	251
H4.	Auswertung	252
H4.1.	Rohdaten: μ -Schlupf-Kurven	252
H4.2.	Einzelne μ -Schlupf-Kurven	254
H5.	Alle μ-Schlupf-Kurven einer Geschwindigkeit	263
H6.	Positive μ-Schlupf-Kurven einer Geschwindigkeit	264
H7.	Vergleich der μ-Schlupf-Kurven von $\pm 2^\circ$	266
Anhang I	Bedienungsanleitung für PAMELA	267
Anhang J	Bedienungsanleitung für die Matlab-Auswertung	273

Formelzeichen und Abkürzungen

Lateinische Buchstaben

Zeichen	Einheit	Bezeichnung
d, D	mm	Durchmesser
f	Hz	Frequenz
F	N	Kraft
G	bit	Datengröße
K	./.	Kanalanzahl
M	Nm	Moment
n	1/s	Drehzahl
p	bar	Druck
r, R	mm	Radius, Reifenhalmmesser
s	%	Schlupf
T	°C	Temperatur
U	V	Spannung
v	m/s, km/h, Mbit/s	Geschwindigkeit

Griechische Buchstaben

Zeichen	Einheit	Bezeichnung
α	°, rad	Schräglaufwinkel
β	°, rad	Schwimmwinkel
δ	°, rad	Radlenkwinkel
σ	./.	Standardabweichung
μ	./.	Reibbeiwert, Kraftschlussbeiwert
φ	°, rad	Drehwinkel

Indizes

Zeichen	Bezeichnung
0	Nullstellung, Startposition
$Bord$	Bord
$CRIO$	CompactRIO
dyn	dynamisch
Fzg	Fahrzeug
$gleit$	Gleitreibung

<i>Hydraulik</i>	Hydraulik
<i>KMD</i>	Kraftmessdose
<i>längs</i>	längs
<i>max</i>	maximal
<i>MF</i>	Messfelge
<i>min</i>	minimal
<i>MR</i>	Messrad
<i>Prop</i>	Proportionalventil
<i>quer</i>	quer
<i>Rad</i>	Rad
<i>RR</i>	Referenzrad
<i>ü</i>	Übertragung
<i>x</i>	x-Koordinate
<i>y</i>	y-Koordinate
<i>z</i>	z-Koordinate

Abkürzungen

Abkürzung	Bezeichnung
AI	Analog Input
AO	Analog Output
CPU	Central Processing Unit
CRIO	Compact Reconfigurable Input / Output
DI	Digital Input
DIFF	Differentielles Messen
DIN	Deutsches Institut für Normung
DIO	Digital Input Output
DMA	Direct Memory Access
DO	Digital Output
DRAM	Dynamic Random Access Memory
FIFO	First In First Out
FPGA	Field Programmable Gate Array
G	grafische Programmiersprache in LabVIEW
GUI	Graphical User Interface
HPL	High Priority Loop auf dem FPGA
I/O	Input / Output
ID	Identifikationsnummer
IEEE	Institute of Electrical and Electronics Engineers
Ink.	Inkrement
IP	Internettransferprotokoll
IRQ	Interrupt Request
LabVIEW	Laboratory Virtual Instrumentation Engineering Work-
LED	Light Emitting Diode
LKW	Lastkraftwagen
LL	Linkslauf
LPL	Low Priority Loop auf dem FPGA

MR	Messrad
MR	Messrad
NI	National Instruments
NPL	Normal Priority Loop auf dem RTS
NRSE	Non.-Referenced Single-Ended
PAC	Programmable Automation Controller
PAMELA	PETRA's automatische Messdatenerfassungslogik und Auswertesoftware
PC	Personal Computer
PETRA	Personenwagen-Reifen-Traktions-Messanhänger
PKW	Personenkraftwagen
POTI	Potentiometer
RAM	Random Access Memory
RL	Rechtslauf
RLW	Radlenkwinkel
RR	Referenzrad
RSE	Referenced Single-Ended
RT	Realtime
RTS	Realtime System
RTS	Realtimesystem
SW	Schräglaufwinkel
SWK	Schwimmwinkelkompensation
SSW	Sollschräglaufwinkel
TCL	Time Critical Loop auf dem RTS
TCP	Transmission Control Protocol
TTL	Transistor Transistor Logik
UDP	User Datagram Protocol
UML	Unified Modeling Language
VI	Virtual Instrument
XOR	exklusiv Oder
XP	Extreme Programming

1 Einleitung

1.1 Motivation

Die einzige Schnittstelle, an der die für das Fahren eines Kraftfahrzeuges erforderlichen Kräfte übertragen werden, ist die Kontaktfläche zwischen Reifen und Fahrbahn. Deshalb kommt dem Reifen eine elementare Bedeutung für das Fahrzeugverhalten zu. Die Größe der maximal übertragbaren Kräfte in Längs- sowie in Querrichtung, die von den Eigenschaften des Reifens, der Fahrbahn, des Zwischenmediums und des Fahrzeugs abhängig ist, ist von entscheidender Bedeutung für die Sicherheit im Straßenverkehr. Zur Ermittlung der übertragbaren Kräfte zwischen Reifen und Fahrbahn dienen Kraftschlussbeiwertmessungen. Diese Messungen werden im Labor auf einem dafür entwickelten Prüfstand durchgeführt oder unter realen Bedingungen mit einem speziell dafür konstruierten System. Am Fachgebiet Fahrzeugtechnik der TU Darmstadt wurde der Messanhänger PETRA I (Personenwagen-Reifen-Traktions-Messanhänger) entwickelt um das Reibverhalten von PKW-Reifen unter realistischen Bedingungen zu analysieren. Die Analyse des Reibverhaltens ist dabei unter Variation von Reifentyp, Radlast, Schräglaufwinkel, Sturzwinkel, Zwischenmedium und Fahrbahntextur möglich.

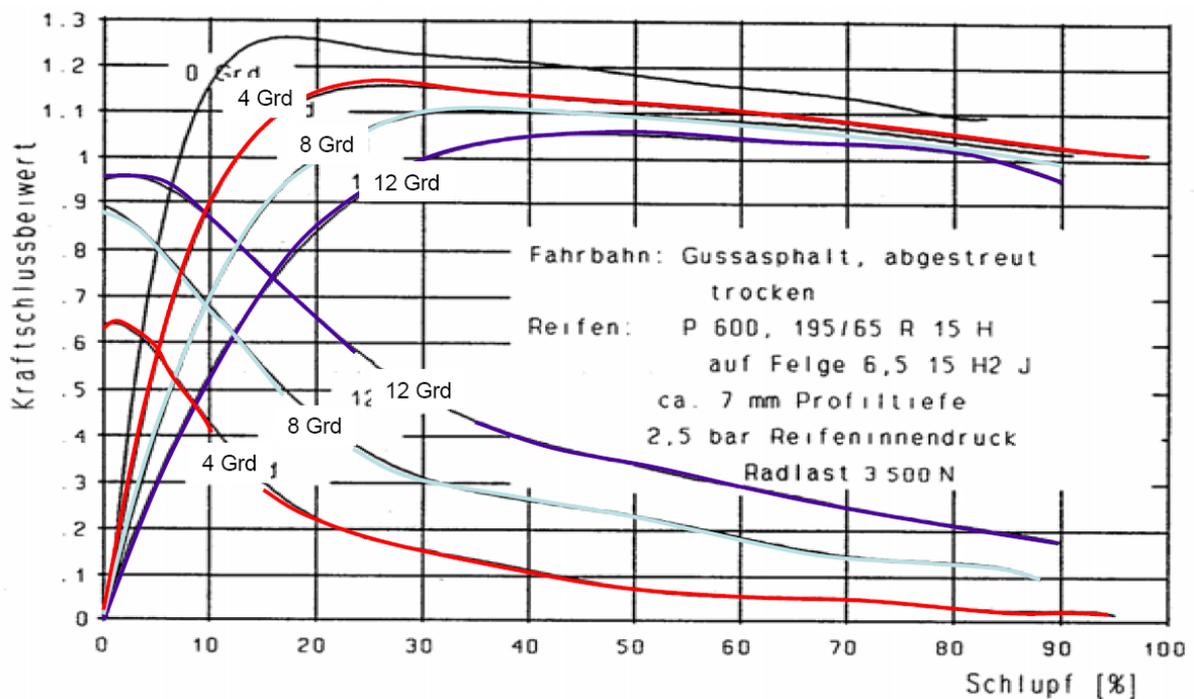


Abbildung 1.1: Reibwert-Schlupf-Kurven aus Winner¹

Die Abbildung 1.1 zeigt Reibwert-Schlupfkurven aus Winner ¹, die für das weitere Verständnis wichtig sind. Darin ist auf der Abszisse der Bremschlupf s aufgetragen und auf der Ordinate der Kraftschlussbeiwert μ (= Reibwert) in längs- und quer-Richtung. Dabei gelten folgende Definitionen:

$$s = \frac{|v_{MR} - v_{Fzg}|}{v_{Fzg}} \quad (1.1)$$

$$\mu_x = \frac{F_x}{F_z} \quad (1.2)$$

$$\mu_y = \frac{F_y}{F_z} \quad (1.3)$$

Die oberen Kurven sind dabei die Reibwert_{längs}-Schlupf-Kurven und die unteren die Reibwert_{quer}-Schlupf-Kurven.

PETRA wurde in den 90er Jahren erfolgreich im Rahmen mehrerer Dissertationen eingesetzt und schrittweise weiterentwickelt. Die in diesen Arbeiten mit PETRA ermittelten μ - s -Kurven lieferten wichtige Erkenntnisse über das Reibungsverhalten von PKW-Reifen. Nach längerer Standzeit wird PETRA für zukünftige Projekte aktualisiert und erneut in Betrieb genommen. Da in Zukunft mit dem Messanhänger auch die Möglichkeit bestehen soll Reifenschwingungen zu erfassen, die laut Michelin ² in einem Frequenzbereich zwischen ca. 30 und 2000Hz liegen, ist es erforderlich, dass die Messtechnik diesen Bereich erfassen kann.

Zu Beginn dieser Arbeit wurde PETRA I mit neuer Messtechnik ausgestattet. Es wird neben neuen Sensoren ein integriertes PAC (Programmable Automation Controller)-System, das so genannte CompactRIO von National Instruments (NI) verwendet. Dieses System, im Folgenden CRIO genannt, befindet sich wie die übrige Elektronik, auf dem Messanhänger PETRA I. An CRIO werden über die eingesteckten Analog- und Digitalmodule die Sensoren und Aktoren angeschlossen. Das CRIO übernimmt die Messdatenerfassung sowie die Sensoren- und Aktorenansteuerung. Ein Laptop der sich im Zugfahrzeug befindet steuert und überwacht über eine Netzwerkverbindung das CRIO.

¹ Winner: Skriptum Kraftfahrzeuge, 2006, S. 44

² Michelin: Der Reifen - Komfort, 2005, S. 54

1.2 Konkretisierung der Aufgabenstellung

Ziel der vorliegenden Arbeit ist es, eine Software zur Erfassung von Messgrößen und Steuerung sowie Regelung von Aktoren zu erstellen. Die Messdatenaufzeichnung umfasst alle notwendigen Messgrößen, die zur Bestimmung der Reibwert-Schlupf-Kurven in Längs- und Querrichtung benötigt werden. Außerdem soll der sich während des Versuchsbetriebs einstellende Schwimmwinkel des Messanhängers während des Messbetriebs automatisch kompensiert werden. Da es sich bei dem verwendeten System um eine verteilte Architektur handelt, liegt die Herausforderung darin, die Software geschickt auf die verschiedenen Ebenen zu verteilen um die Leistung des Gesamtsystems hinsichtlich der maximal möglichen Datenrate zu optimieren. Die Programmierung erfolgt in LabVIEW.

Die Aufgabenstellung umfasst die Anforderungsdefinition der zu erstellenden Software zu erarbeiten, einen Softwareentwurf zu erstellen, entsprechend des Entwurfs die jeweiligen Modulspezifikationen zu erarbeiten und zu bewerten, die eigentliche Programmierung, d.h. die Umsetzung der bestbewerteten Konzepte, sowie die Verifikation und die anschließende Validierung des Gesamtsystems durch Messfahrten.

Parallel zu dieser Masterthesis wurde die Diplomarbeit 413/06³ mit dem Titel „Entwicklung und Verifikation eines Sicherheitskonzepts für einen Messanhänger zur Ermittlung des Reibverhaltens von PKW-Reifen“ von Cand. Ing. Ingemar Wolf durchgeführt, auf die in dieser Arbeit verwiesen wird.

1.3 Klärung der Begriffe Verifikation und Validierung

Der Begriff Verifikation wird in der Informatik unterschiedlich verwendet. Daher wird hier definiert wie die Begriffe Verifikation und Validierung in dieser Arbeit verwendet werden. In der theoretischen Informatik wird unter der Verifikation der formale Beweis der Korrektheit eines Programms, während in der Softwareentwicklung das Prüfen auf Erfüllung der Spezifikation gemeint ist, verstanden. Die folgenden Definitionen sind die gebräuchlichen Definitionen der Softwareentwicklung wie sie in dieser Arbeit verwendet werden, vgl. Sommerville⁴ und Spillner⁵.

³ Wolf: Diplomarbeit 413/06

⁴ Sommerville: Software Engineering, 2001, S. 427

⁵ Spillner, Linz: Basiswissen Softwaretest, 2003, S. 41

- Verifikation

Prüfung, ob die Ergebnisse einer Entwicklungsphase die Vorgaben der Spezifikation (Phaseneingangsdokumente) erfüllen.

Wurde das System richtig realisiert?

- Validation (Validierung)

Prüfung, ob ein Entwicklungsergebnis die individuellen Anforderungen bezüglich einer speziellen beabsichtigten Nutzung erfüllt.

Wurde das richtige System realisiert?

1.4 Methodik des Vorgehens

Bei einem Vorgehensmodell in der Softwareentwicklung handelt es sich um eine abstrakte Darstellung eines Softwareprozesses. Das Vorgehensmodell legt fest, wie bei der Softwareentwicklung vorgegangen wird.

Aufzählung der in Betrachtung gezogenen Vorgehensmodelle:

- Wasserfallmodell
- Formale Systementwicklung
- Extrem Programming (XP)
- Evolutionäre Entwicklung
- Wiederverwendungsorientierte Entwicklung
- V-Modell
- W-Modell (erweitertes V-Modell)

Bewertung und Auswahl des zu verwendenden Vorgehensmodells:

Da die Software direkt in die Aktorik eingreift, ist sie maßgebend an der Sicherheit des Gesamtsystems beteiligt. Daher wird bei dem Vorgehen in der Softwareentwicklung ein besonderes Augenmerk auf ein formales Vorgehen, sowie auf die Verifikation und die Validierung gelegt. Im Folgenden werden die oben aufgezählten Vorgehensmodelle hinsichtlich ihrer Eignung zum Vorgehen in der Softwareentwicklung für das Projekt PETRA betrachtet:

Das „Wasserfallmodell“ und die „Formale Systementwicklung“ sind ungeeignet, da sie keine implizite Verifikation und Validierung verlangen.

Das XP ist für das Projekt ungeeignet, da kein formales Vorgehen genutzt wird. XP ist für Projekte geeignet, bei denen sich die Anforderungen schnell ändern, was bei PETRA nicht der Fall ist.

Die Evolutionäre Entwicklung hat den Nachteil, dass der Fortschritt im Projekt nicht gemessen werden kann.

Die Wiederverwendungsorientierte Entwicklung kann nicht angewendet werden, da kein Vorgänger- / Altsystem vorhanden ist, welches wiederverwendet werden kann.

Der Hauptvorteil des V-Modells ist, dass die Verifikation und die Validierung implizit vorgesehen sind. Ein weiterer Vorteil ist das formale Vorgehen im V-Modell, wodurch auch der Fortschritt sichtbar ist. Der Nachteil des V-Modells ist, dass die späte Betrachtung der Verifikation und Validierung häufig dazu führt, dass das System bzw. die Module (Komponenten) nur schwer zu verifizieren und zu validieren sind.

Das W-Modell ist dementsprechend erweitert, dass parallel zu jedem Schritt der Softwareentwicklung (linker Ast des V-Modells), vgl. nachstehende Abbildung 1.2, entsprechende Testaktivitäten betrachtet werden, welche später im Testprozess (rechter Ast des V-Modells) verwendet werden. Das heißt es wird parallel zur Entwicklung die entsprechende Validations- bzw. Verifikationsvorschrift erstellt. Beim W-Modell wird besonderen Wert auf den Testprozess gelegt.

Fazit:

Da das W-Modell nur Vorteile gegenüber den anderen betrachteten Vorgehensmodellen im Hinblick auf das Projekt hat, wird es als Vorgehensmodell für die Entwicklung und Validierung der Software für PETRA verwendet.

Die nachfolgende Abbildung zeigt das W-Modell welches als Vorgehensmodell für die Softwareentwicklung für das Projekt PETRA I gewählt wurde. Der linke, äußere Ast des W-Modells, welcher aus dem Wasserfall und V-Modell bekannt ist, enthält die Konstruktionsphasen in denen die Software zunehmend detaillierter beschrieben wird. Der rechte Ast, der ebenfalls aus dem V-Modell bekannt ist, enthält die Teststufen, bei denen jedem Spezifikations- bzw. Konstruktionsschritt eine korrespondierende Teststufe zugeordnet ist. Der entscheidende Mehrwert des W-Modells im Vergleich zum V-Modell ist der linke, innere Ast. Hier beginnen schon direkt nach der Spezifikationsphase die Testaktivitäten, die in Testspezifikationen enden, für den entsprechenden späteren Test. Der Vorteil ist, dass schon in dieser frühen Phase des Softwareprozesses festgestellt werden kann, ob die entsprechende Funktionalität verifiziert oder validiert werden kann.

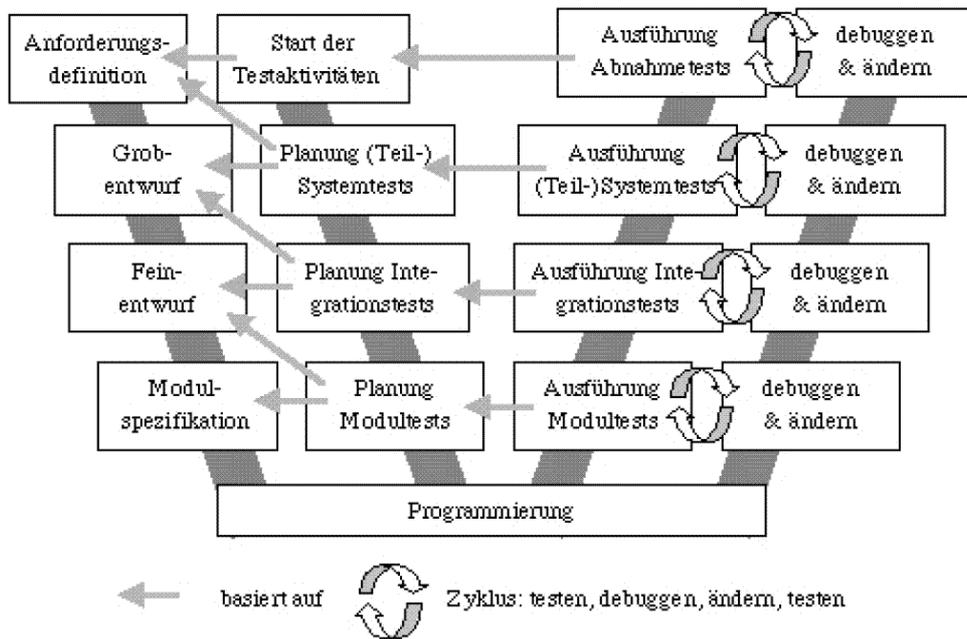


Abbildung 1.2: W-Modell (Erweitertes V-Modell)⁶

⁶ Spillner: W-Modell, 2006, S. 2

2 Hardware des Messanhängers PETRA I



Abbildung 2.1: PETRA I auf dem Flugplatz in Griesheim

2.1 Aufbau

PETRA I, welche in Abbildung 2.1 zu sehen ist, besteht aus einem Rahmen, der Messtechnik, dem Messrad auf der linken Seite, dem Referenzrad auf der rechten Seite und der Hydraulik die zum Abbremsen des Messrades benötigt wird.

Die Messtechnik befindet sich in der Schaltschrank hinter dem Messrad, darin befindet sich neben der Leistungselektronik, der Messfelgenauswerteeinheit von Kistler und der Correivertauswertebbox von Datron auch das CRIO. Die Elektronik wird über zwei LKW-Batterien, die sich im vorderen Teil des Anhängers vor dem Ölkühler befinden, mit Strom versorgt. PETRA verfügt somit über ein 12V Bordnetz.

PETRAS Rahmen ist asymmetrisch aufgebaut, damit das Messrad möglichst nahe auf der Verlängerung des Kupplungspunktes nach hinten liegt. Der möglichst kleine Hebelarm multipliziert mit der Bremskraft resultiert in einem Giermoment beim Aufbringen eines Bremsmoments am Messrad. Als Folge stellt sich ein Schwimmwinkel am Messanhänger ein, welcher durch die Asymmetrie des Rahmens klein ist.

Am Rahmen über dem Messrad befindet sich ein Gewindebolzen, auf den Zusatzgewichte gesteckt werden können um die Radlast zu erhöhen.

Hauptbestandteil für die Funktionalität von PETRA ist die Hydraulik. Diese besteht aus einem Ölkühler, einem Tank, einer Drossel, sowie einer Axialkolbenpumpe. Die einzige Aufgabe dieser Hydraulik ist es ein steuerbares Bremsmoment aufzubauen. Dazu fördert die Pumpe das Hydrauliköl aus dem Tank gegen die Drossel und erzeugt durch die entstehende Verzwängung ein Bremsmoment am Messrad. Die Axialkolbenpumpe wird über ein Getriebe, welches zwischen Messrad und Axialkolbenpumpe sitzt angetrieben. Das Getriebe ist notwendig, damit abhängig von der gefahrenen Geschwindigkeit ein anderer Gang verwendet werden kann und somit die Höchstdrehzahl der Pumpe nicht überschritten wird. Das Bremsmoment wird über ein Proportionalventil variiert, welches den Kolbenhub der Pumpe verstellt und über die Elektronik angesteuert wird. Zum Schalten dieses Ventils wird ein Mindestdruck von 40bar benötigt. Somit legt diese Grenze des Proportionalventils das kleinste einstellbare Bremsmoment fest.

Die Räder können über einen Doppelquerlenker, der jeweils über einen Motor verstellt werden kann, gelenkt werden. Somit ist ein Verstellen der Radlenkwinkel sowohl am Messrad als auch am Referenzrad während der Fahrt möglich.

Ein weiterer Motor erlaubt die Sturzverstellung des Messrades im Stand, welche über einen Schalter in der Messbox angesteuert wird. Eine Sturzwinkelsensorik ist zurzeit noch nicht vorhanden.

2.2 Aktorik

2.2.1 Analoge Aktorik

2.2.1.1 Analogverstärker des Proportionalventils

Über den Analogverstärker des elektromagnetischen Proportionalventils wird der Pumpenhub der Hydraulikpumpe und damit das Bremsmoment variiert.

2.2.2 Digitale Aktorik

2.2.2.1 Motoren für Radlenkwinkeleinstellung

Zur Verstellung der Radlenkwinkel werden Motoren verwendet, die über ein Getriebe die Lenkung der Räder übernehmen. Die Motoren werden von der Software digital angesteuert. Somit werden folgende 4 Signale benötigt:

- Motor Radlenkwinkelverstellung Messrad Rechtslauf
- Motor Radlenkwinkelverstellung Messrad Linkslauf
- Motor Radlenkwinkelverstellung Referenzrad Rechtslauf
- Motor Radlenkwinkelverstellung Referenzrad Linkslauf

2.3 Sensorik

2.3.1 Analoge Sensorik

2.3.1.1 Bordspannung PETRA

Die Bordspannung dient zur Überwachung des Ladezustandes der Batterien. Damit wird ein rechtzeitiger Versuchsabbruch bei Erreichen des kritischen Ladzustandes ermöglicht. Die Restenergie wird benötigt, um die Räder in Nullstellung zurückzustellen.

2.3.1.2 Hydraulikdruck

Der Hydraulikdruck wird über einen Drucksensor erfasst der in den Hochdruckbereich der Hydraulik eingebaut wurde. Durch diesen Sensor ist eine Überwachung des Druckes möglich und somit kann durch richtiges Reagieren der Bauteilschutz gewährleistet werden. Der verwendete Drucksensor hat einen Messbereich von 0-600bar welcher 0-10V entspricht.

2.3.1.3 Hydrauliktemperatur

Mit einem Thermoelement wird die Öl-Temperatur im Tank überwacht.

2.3.1.4 Elektroniktemperatur

Die Elektroniktemperatur wird durch den in CRIO eingebauten Temperatursensor überwacht. Damit kann das System bei Erreichen der Grenztemperatur automatisch heruntergefahren werden. Andere elektronische Bauteile werden nicht oder nur indirekt durch deren Wärmeabgabe überwacht.

2.3.1.5 Potentiometer der Radlenkwinkelverstellung

Für die Erfassung der Radlenkwinkelstellungen der beiden Räder wird jeweils ein Stabpotentiometer eingesetzt, welches den Weg misst um den die Lenkung verstellt wurde. Unter der Annahme, dass bei kleinen Winkeln der Weg proportional zum eingestellten Winkel ist, kann hiermit der eingestellte Radlenkwinkel am entsprechenden Rad ermittelt werden.

2.3.1.6 Messfelge (Kistler)

Mit der verwendeten Messfelge der Firma Kistler können alle Radkräfte und Radmomente gemessen werden. Die Messfelge wird mit der zugehörigen Auswertebox verbunden welche die Größen Bremskraft F_x , Seitenkraft F_y , Radkraft F_z , sowie die Momente M_x , M_y und M_z ausgibt und die für die Berechnung der Momente verwendeten $\cos \varphi$ und $\sin \varphi$ Signale.

2.3.1.7 Kraftmessdose

Die Kraftmessdose dient zur Ermittlung der Bremskraft am Messrad und ist die „Backuplösung“, wenn die Messfelge nicht montiert ist.

2.3.1.8 Correvit (Datron V2)

Der verwendete Correvit der Firma Datron, welcher direkt vor dem Messrad montiert ist, ermittelt zwei Geschwindigkeitsvektoren. Er wird mit der zugehörigen Auswerteeinheit verbunden, welche aus den beiden Vektoren die aktuelle Anhänger- bzw. Fahrzeuggeschwindigkeit und den Schräglaufwinkel des Messrades ermittelt.

2.3.2 Digitale Sensorik

2.3.2.1 Drehzahl Inkrementalgeber

Der verwendete Inkrementalgeber, der zur Erfassung der Drehzahl des Messrades verwendet wird, erfasst 1000 Inkremente (Striche) pro Umdrehung.

2.3.2.2 Endschalter der Radlenkwinkelmotoren

PETRA ist mit 4 Endschaltern ausgestattet die aktiv werden, sobald einer der Radlenkwinkelverstellmotoren in einer Richtung an seine Verfahrgrenze stößt. Diese Grenze ist durch ver-

schieben der Begrenzungselemente einstellbar, welche gegen die Endschalter drücken. Die Endschalter sind für die Bauteilsicherheit notwendig, da die enorme Kraft der Motor-Getriebe-Einheit zum Verbiegen von Bauteilen führen kann, falls die Motoren gegen das geometrische bzw. mechanische Ende stoßen. Daher erfolgt eine Abschaltung der Motoren am Endschalter sowohl elektronisch als auch über die Software.

2.4 CRIO

Das verwendete CRIO besteht aus dem Controller CRIO-9004 und dem Gehäuse CRIO-9104 in das die verwendeten NI-Module gesteckt werden. Die Abbildung 2.2⁷ zeigt ein CRIO-System.



Abbildung 2.2: CRIO

2.4.1 Gehäuse CRIO-9104

Das Gehäuse bietet Steckplätze für 8 Module von denen derzeit 4 Steckplätze verwendet werden. Das in Abbildung 2.2 verwendete CRIO-Gehäuse bietet nur 4 Steckplätze. In dem Gehäuse befindet sich der FPGA der eine 40MHz Clock besitzt und über 192kB RAM verfügt und maximal 3M Gatter verwenden kann⁸.

2.4.2 Controller CRIO-9004

Auf dem Controller befindet sich ein Realsystem (RT), welches mit einem 200MHz Prozessor getaktet ist und über 512MB nicht volatilen Speicher sowie 64MB DRAM verfügt⁹. In dem nicht volatilen Speicher befindet sich sowohl das Betriebssystem LabVIEW Realtime als auch die programmierte Software des Realsystems. Der Controller verfügt über eine 100Mbit/s Netzwerkschnittstelle die sowohl zur Kommunikation mit einem PC zur Programmierung als auch als Datenverbindung zu einem Hostsystem verwendet wird.

⁷ National Instruments: CRIO, 2007

⁸ National Instruments: Datenblatt CRIO-9104, 2007

⁹ National Instruments: Datenblatt CRIO-9004, 2007

2.4.3 NI-Module

Aus den verwendeten Sensoren und Aktoren ergeben sich die zu verwendenden Module. Es werden ein analoges Eingangsmodul (NI-9205), ein analoges Ausgangsmodul (NI-9263), ein digitales Eingangsmodul (NI-9401) und ein digitales Ausgangsmodul (NI-9472) verwendet. Beim Einstecken in die dafür vorgesehenen Steckplätze ist darauf zu achten, dass das Modul in den gleichen Steckplatz gesteckt wird welcher in der LabVIEW Entwicklungsumgebung angemeldet wurde, da sonst die Zuordnung nicht mehr funktioniert.

2.4.3.1 Analoges Eingangsmodul (NI-9205) ¹⁰

Dieses verwendete Modul verfügt über 32 analoge Eingangskanäle, die jeweils eine Auflösung von 16bit bieten. Jeder Kanal kann in den Bereichen $\pm 10V$, $\pm 5V$, $\pm 1V$ oder $\pm 0,2V$ betrieben werden. Zudem können drei verschiedene Anschlussarten verwendet werden:

- Differentielles Messen (DIFF): erlaubt genaueres Messen durch Reduzierung von Rauschen, belegt aber 2 Kanäle
- Referenced Single-Ended (RSE) Messen: massebezogenes Messen
- Non.-Referenced Single-Ended (NRSE) Messen: nicht massebezogenes Messen, hierbei wird ein weiterer Eingang mit Masse verbunden und dadurch das Rauschen im Vergleich zu RSE reduziert

Mit diesem Modul werden folgende Größen in Form einer Spannung erfasst: Bordspannung, Hydraulikdruck, Hydrauliktemperatur, Radlenkwinkel am Messrad, Radlenkwinkel am Referenzrad, die Radkräfte (F_x , F_y , F_z) und die Radmomente (M_x , M_y , M_z) von der Messfelge, $F_{x,KMD}$ von der Kraftmessdose, Fahrzeuggeschwindigkeit und der Schräglaufwinkel. Die Messgrößen der Messfelge und des Correvits werden differentiell erfasst, da dies die wichtigsten Größen sind. Die restlichen Kontrollgrößen werden mittels NRSE Messverfahren erfasst.

2.4.3.2 Analoges Ausgangsmodul (NI-9263) ¹¹

Dieses Modul besitzt 4 Kanäle von je 16bit Auflösung und erlaubt Ausgangsspannungen zwischen $-10V$ und $10V$. An dieses Modul ist zurzeit nur der Proportionalverstärker angeschlossen.

¹⁰ National Instruments: Bedienungsanleitung NI-9205, 2007

¹¹ National Instruments: Bedienungsanleitung NI-9263, 2007

2.4.3.3 Digitales Eingangsmodul (NI-9401) ¹²

Das digitale Modul NI-9401 wird verwendet um die Drehzahl des Messrades über den Inkrementalgeber zu erfassen. Das Modul erwartet 5V/TTL-Signale. Des Weiteren werden durch dieses Modul die Signale aller Endschalter erfasst.

2.4.3.4 Digitales Ausgangsmodul (NI-9472) ¹³

Dieses digitale Ausgangsmodul wird verwendet um die Radlenkwinkelmotoren anzusteuern, sowie eine Nullkalibrierung der Kistler Messfelge durchzuführen.

2.5 Laptop

Für die Tests und Versuchsfahrten wurde ein Laptop vom Type ThinkPad T60 von lenovo verwendet. Da die Systemtests und die Validierung mit diesem Gerät durchgeführt wurde, ist eine korrekte Funktion der programmierten Software nur für dieses Gerät bzw. Geräte die eine vergleichbare Systemanforderung erfüllen, gewährleistet. Die folgende Aufzählung zeigt die Systemspezifikation des verwendeten Notebooks.

- Type: IBM ThinkPad T60 (lenovo)
- Prozessor: Intel Core2 T7200 @ 2.00GHz
- Arbeitsspeicher: 1 GB RAM
- System: Ms Windows XP mit Service Pack 2

¹² National Instruments: Bedienungsanleitung NI-9401, 2007

¹³ National Instruments: Bedienungsanleitung NI-9472, 2007

3 LabVIEW¹⁴

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) ist eine graphische Softwareentwicklungsumgebung von National Instruments. In erster Linie wird LabVIEW in der Mess-, Steuerungs-, Regelungs-, und Automatisierungstechnik eingesetzt. Die Programmierung erfolgt in einer graphischen, datenflussorientierten Programmiersprache, die "G" genannt wird und auf sogenannten Blockdiagrammen „verdrahtet“ wird. Durch diese Besonderheit eignet sich LabVIEW zur Datenerfassung und -Verarbeitung. LabVIEW-Programme bestehen aus zwei Komponenten: Dem Frontpanel, es enthält die Benutzerschnittstelle, und dem Blockdiagramm, welches den graphischen Programmcode enthält. Der „G“-Code wird nicht von einem Interpreter abgearbeitet, sondern kompiliert wodurch die Performance nach Wikipedia¹⁵ vergleichbar mit der anderer höheren Programmiersprachen ist.

Die folgende Abbildung 3.1 zeigt das Frontpanel eines VIs. Dabei handelt es sich um ein Subpanel der grafischen Benutzerschnittstelle die in dieser Arbeit erstellt wurde. Auf dem Frontpanel sind die Ein- und Ausgabeelemente zu sehen, welche der Benutzer verwendet.



Abbildung 3.1: Frontpanel eines VIs

LabVIEW Programme werden als Virtuelle Instrumente oder einfach VIs bezeichnet, Unterprogramme entsprechend Sub-VIs. Die kleinsten nicht mehr weiter zerlegbaren Einheiten werden in LabVIEW Primitive genannt. Die Programmierung beginnt mit einem leeren VI, dem Blockschaltbild. In dieses VI können Funktionen, Elemente, Datentypen und Primitive gelegt werden, sowie weitere Sub-VIs eingefügt werden. Diese Symbole können mittels einer Verdrahtung verbunden werden. Des Weiteren können auf dem Frontpanel per „drag and drop“ Benutzerein- und ausgaben erstellt werden, welche als Symbol in dem Blockdiagramm erscheinen und einfach wie andere Elemente verdrahtet werden.

¹⁴ National Instruments: Labview Hilfe, 2006

¹⁵ Wikipedia: Labview, 2007

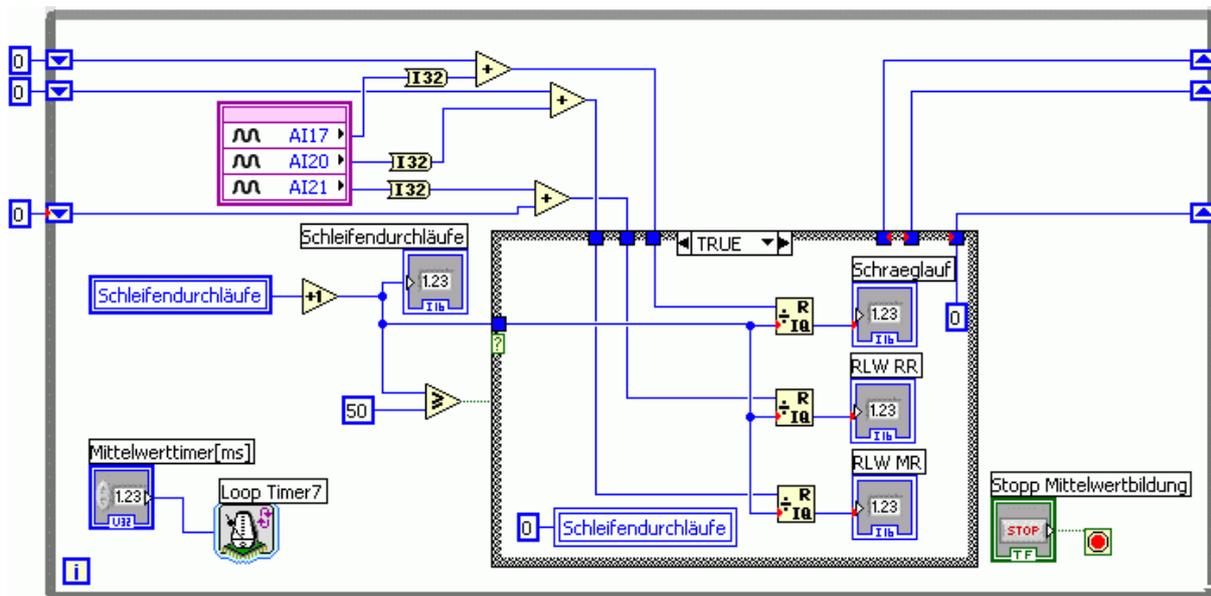


Abbildung 3.2: Blockdiagramm eines VIs mit der Funktionalität der Mittelwertbildung

Die obige Abbildung 3.2 zeigt beispielhaft das Blockdiagramm eines VIs. Dieses VI bietet die Funktion einer Mittelwertbildung auf einem FPGA.

Hat ein VI mehrere Eingänge, beginnt die Abarbeitung dieses VI erst dann wenn an allen Eingängen Daten vorhanden sind. Die Ergebnisse liegen erst an den Ausgängen an wenn das komplette VI abgearbeitet wurde. Somit wird eine Synchronisierung an jedem VI bzw. an jeder Funktion erreicht, die mehr als einen Eingang besitzt.

In „G“ gibt es keine definierte Abarbeitungsreihenfolge. Die Reihenfolge der Abarbeitung wird durch den Datenfluss sowie die Art der Anschlusselemente, Datenenke (Eingang) und Datenquelle (Ausgang), festgelegt. Soll gezielt eine bestimmte Abarbeitungsreihenfolge sichergestellt werden, werden sogenannte Sequenzstrukturen verwendet, welche hintereinander von links nach rechts abgearbeitet werden. Durch den Datenfluss sind keine rekursiven Aufrufe möglich, mittels sogenannter „Schieberegister“ oder „Rückkopplungsknoten“ lassen sich dennoch Rekursionen realisieren.

In LabVIEW sind viele Funktionen polymorph. Polymorphie heißt übersetzt "Vielgestaltigkeit" und ist als Begriff für die Vererbungsstruktur in objektorientierten Sprachen bekannt. An polymorphe Funktionen gibt können in LabVIEW verschiedene Eingangstypen gelegt werden und die Funktion erkennt selbst, ob z.B. zwei Double- oder zwei Integer- Zahlen addiert werden und lädt automatisch die entsprechende Funktion.

Vorteile von LabVIEW:

- Durch das Frontpanel lassen sich sehr einfach und schnell grafische Benutzerschnittstellen erstellen.
- Parallele Abarbeitung erfolgt automatisch wenn zwei Schleifen nebeneinander auf das Blockdiagramm gelegt werden, sofern diese unabhängig voneinander sind. Um die Threadzuweisung kümmert sich LabVIEW selbständig, sofern Threads aktiviert sind.
- Bei kleinen bis mittleren Programmen erhöht die grafische Darstellung des Programmablaufs die Lesbarkeit.
- Mit LabVIEW lassen sich neben Programmen für normale PC-Systeme auch Programme für FPGAs und Realtimesysteme erstellen.

Nachteile von LabVIEW:

- Kleine Änderungen am grafischen Code können große Umbauaktionen nach sich ziehen, wenn kein Platz für die Änderung an der entsprechenden Stelle vorhanden ist.
- In sehr großen, komplexen Programmen mit vielen Verdrahtungen wird schnell die Übersicht verloren.
- Übereinanderliegende Verdrahtungen können zu Verwirrungen führen.

LabVIEW 8.01 vs. 8.20

Die zu Beginn der Programmierung verwendete LabVIEWversion 8.01 lief sehr instabil und es kam häufig zu ungeklärten Systemabstürzen. 5-10 Fehlermeldungen am Tag gehörten zur Tagesordnung. Nicht selten stürzte die Entwicklungsumgebung ab und zerstörte dabei die zuletzt verwendete Version. Die Programmierung auf dem RT-System und dem FPGA, war mit der Version sehr mühselig, da Änderungen am Blockschaltbild viel Zeit benötigten. Zusammengefasst war ein produktives Programmieren nur schwer möglich. Von der Verwendung der LabVIEW 8.01 Version kann nur abgeraten werden.

Mit der LabVIEW Version 8.20, wurde dies besser. Fehlermeldungen gehören der Vergangenheit an und Systemabstürze sind nur direkt nach der Umkompilierung von 8.01er VIs nach 8.20er VIs aufgetreten. Mit der neuen Version ist auch das Programmieren auf dem FPGA und RT-System deutlich verbessert worden, denn Änderungen werden jetzt gleich übernommen.

4 Anforderungsanalyse

4.1 Ziel der Anforderungsanalyse

Ziel der Anforderungsanalyse ist die Klärung der Aufgabenstellung, die Bestimmung von Anforderungen und die Detaillierung dieser. Das Ergebnis der Anforderungsanalyse ist ein Anforderungsdefinitionsdokument oder ein Pflichtenheft das als Vertragsgrundlage dient.

4.2 Methodik der Anforderungsanalyse

Der Ablauf der Anforderungsanalyse beinhaltet laut Sommerville ¹⁶ folgende Aktivitäten: Verstehen des Anwendungsbereichs, Anforderungssammlung, Klassifizierung, Konfliktlösung, Setzen von Prioritäten und Anforderungsüberprüfung. Mögliche Methoden zur Anforderungsanalyse sind Viewpoint-orientierte Bestimmung, Szenarien, Ethnographie, Prototypen und die Strukturierte Analyse.

Für die Ermittlung der Anforderungen an die Software wurden mehrere Methoden kombiniert. Brainstorming auf Basis der Anforderungsliste für das Gesamtsystem und des erarbeiteten Verständnisses des Anwendungsbereichs. Finden und detaillieren von Anforderungen die aus dem von Dipl.-Ing. Marcus Reul entwickelten Prototyp, welcher um das neu entwickelte Drehzahlmodul erweitert wurde, abgeleitet werden konnte. Anschließend wurden mittels eines Ablaufszenarios weitere Anforderungen gefunden und bestehende detailliert. Der Vorteil des Ablaufszenarios ist, dass keine wichtige Anforderung übersehen wird.

Die ermittelten Anforderungen wurden wie in Sommerville ¹⁷ in funktionale und nichtfunktionale Anforderungen gegliedert. Funktionale Anforderungen sind Aussagen zu Diensten die das System leisten soll, zur Reaktion des Systems auf bestimmte Eingaben und zum Verhalten des Systems in bestimmten Situationen. Nichtfunktionale Anforderungen sind Beschränkungen der durch das System angebotenen Dienste oder Funktionen. Die Anforderungen wurden wie in der Softwareentwicklung üblich mit einer Priorität versehen.

Des Weiteren wurden die Anforderungen wie in der Produktentwicklung üblich in Festforderung, Bereichsforderung, Zielforderung und Wunsch gegliedert. Danach wurden die Anforde-

¹⁶ Sommerville: Software Engineering, 2001, S. 134 / 135

¹⁷ Sommerville: Software Engineering, 2001, S. 109 - 114

rungen wie in Sommerville ¹⁸ einer Konsistenzprüfung, einer Vollständigkeitsprüfung, einer Realisierbarkeitsprüfung und einer Verifizierbarkeitsprüfung unterzogen und damit validiert. Dies geschah durch Anforderungs-Reviews mit den Reviewern Dipl.-Ing. Marcus Reul und Cand.-Ing. Ingemar Wolf, und der Fertigstellung des Prototypen der alle Hauptanforderungen abdeckt, sowie durch Erarbeitung der Validierungsvorschriften. Das Ergebnis ist das Anforderungsdefinitionsdokument.

4.3 Ablaufszenario

Ein Ablaufszenario ist ein Szenario das den Ablauf des Programms zur Bewältigung einer bestimmten Aufgabe veranschaulicht. Für das Projekt PETRA wurde als Szenario „Kraftbewert-Schlupf-Kurven erfassen“ verwendet, da dieses Szenario alle benötigten Funktionalitäten abdeckt. Dieses Szenario ist in der nächsten Grafik zu erkennen, hierbei wurde die UML Notation verwendet.

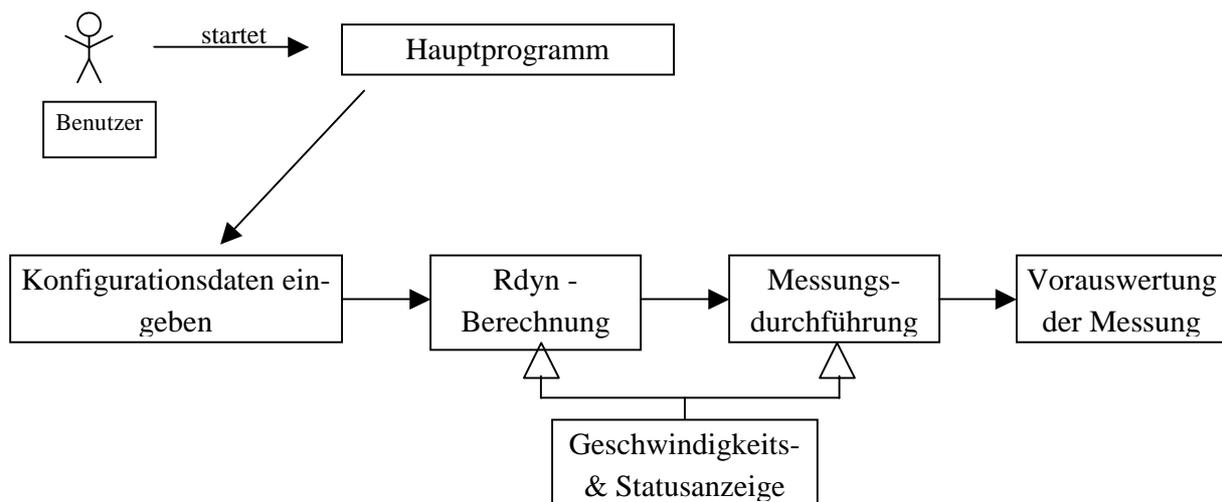


Abbildung 4.1: Ablaufszenario Reibwert-Schlupf-Kurven messen

Hauptprogramm:

Im Hauptprogramm wählt der Benutzer eine der Unterfunktionen: Konfigurationsdaten eingeben, R_{dyn} -Berechnung, Kalibrierung, Messungsdurchführung oder Vorauswertung der Messung.

¹⁸ Sommerville: Software Engineering, 2001, S. 147 / 148

Konfigurationsdaten eingeben:

In diesem Menü kann der Benutzer folgende Konfigurationsdaten zur Messdurchführung eingeben: Ordnerpfad, Messdateinamen, Messfelge montiert? ja/nein, Geschwindigkeitstoleranz, Schräglaufwinkeltoleranz, Schwimmwinkeltoleranz, Reifendruck, Reifentemperatur, Außentemperatur, Asphalttemperatur und weitere Bemerkungen.

Geschwindigkeits- & Statusanzeige:

In den Unterfunktionen „ R_{dyn} -Berechnung“ und „Messungsdurchführung“ werden die Informationen aktuelle Längsgeschwindigkeit, Messradgeschwindigkeit, Schlupf, Kraftschlussbeiwert, Sollschräglaufwinkel, Schräglaufwinkel, Radlenkwinkel Messrad, Radlenkwinkel Referenzrad, Öldruck, Öltemperatur, Bordspannung, CRIO-Gehäusetemperatur und ggf. weitere angezeigt.

R_{dyn} -Berechnung:

Der dynamische Reifenhalbmesser ist definiert als der Reifenhalbmesser der aus einem abrollenden Rad berechnet werden kann, dabei handelt es sich um keine geometrisch ermittelbare Größe.

Der Benutzer gibt an bei welcher Geschwindigkeit er die R_{dyn} -Berechnung durchführen möchte. Nach Bestätigung beschleunigt der Fahrer das Fahrzeug auf die entsprechende Geschwindigkeit und hält diese. Das System erkennt wenn die Geschwindigkeit erreicht wurde und beginnt die Messung. Wurden genug Werte erfasst wird die Berechnung gestartet und R_{dyn} ausgegeben.

Messungsdurchführung:

Der Benutzer gibt den gewünschten Schräglaufwinkel ein. Danach wählt er ob er eine dynamische Messung oder eine statische Messung durchführen möchte. Wählt er „dynamische Messung“ folgt daraus, dass alle Daten aufgezeichnet werden und somit dynamische Effekte erfasst werden. Wählt er „statische Messung“, gibt er an bei welcher Geschwindigkeit die μ -Schlupf-Kurve gemessen wird. Dabei werden die instationären Zustände nicht erfasst um die Messdaten zu reduzieren. Nach Bestätigung wird der Radlenkwinkel des Messrades eingestellt. Der Fahrer beschleunigt auf die entsprechende Geschwindigkeit und hält diese. Nach einer bestimmten, voreingestellten Zeitspanne innerhalb eines bestimmten Geschwindigkeitstoleranzbereichs und Schräglaufwinkeltoleranzbereichs werden die Messdaten gespeichert und danach die Radbremskraft erhöht. Diese Bremskraft kompensiert der Fahrer durch Gas geben. Sinkt die Geschwindigkeit unter den Toleranzbereich oder verlässt der Schräglaufwinkel den Toleranzbereich werden diese Messdaten nicht aufgezeichnet, um riesige Datenmengen zu vermeiden. Nach der Messung drückt der Benutzer den Knopf „Messung beendet“, daraufhin wird die Messdatei geschlossen und gespeichert.

Vorauswertung der Messung: (Wunschforderung)

Um dem Benutzer einen schnellen Überblick über die Messung zu geben wird eine $\mu_{\text{längs}}$ -Schlupf-Kurve und ein μ_{quer} -Schlupf-Kurve anhand weniger ausgewählter Punkte erstellt. Somit kann der Benutzer sehen ob die Messung plausibel ist. Dadurch können grobe Messfehler vor der eigentlichen Auswertung erkannt werden und ggf. die Messung wiederholt werden.

4.4 Darstellung der Anforderungen

Die Anforderungen wurden in Tabellenform in das Anforderungsdefinitionsdokument geschrieben vgl. Sommerville ¹⁹. Die Tabelleneinträge wurden dem Projekt angepasst.

Die verwendeten Einträge werden im Folgenden aufgelistet und beschrieben:

- Funktion: Benennung der Anforderung
- Anforderungsart: Gliederung in Festforderung, Bereichsforderung, Zielforderung und Wunsch
- ID: Eindeutige Kennzeichnung der Anforderung
- Priorität: Priorisierung der Anforderung
- Version: Die Versionsnummer wird bei Änderung der Anforderung angepasst
- Autor: Ersteller der Anforderung
- Reviewer: Reviewer die die Anforderung überprüfen
- Aufwandsschätzung: Schätzung des Aufwands zur Umsetzung der Anforderung
- Beschreibung: Beschreibung der Anforderung
- Eingaben: Eingabegrößen
- Quellen: Herkunft der Eingabegrößen
- Ausgaben: Ausgabegrößen
- Senke: Ziel der Ausgabegrößen
- Ziel: Ziel der Anforderung

¹⁹ Sommerville: Software Engineering, 2001, S. 121

Das Anforderungsdokument umfasst folgende Anforderungen:

Funktionale Anforderungen:

- Sensordatenerfassung
- Aktoransteuerung
- Schwimmwinkelkompensation
- Drehzahlberechnung
- Bremsmomentverstellung
- Anzeige von Kontrollparametern
- Anzeige von Statusparametern
- Anzeige weiterer Statusparameter
- Warnung beim Verlassen von Parametergrenzen
- Herunterfahren der Messtechnik
- Benutzerschnittstelle
- Konfigurationsmöglichkeit
- Zero-Kalibrierung
- Kalibrierung
- Rdyn-Berechnung
- Messungsdurchführung
- Messdatenaufzeichnung
- Messdatenpersistenzprüfung

Nichtfunktionale Anforderungen:

- Performance
- Benutzerfreundlichkeit
- Kompatibilität zu Matlab
- Zuverlässigkeit
- Speicherplatzanforderung

Die folgende Tabelle zeigt am Beispiel der Schwimmwinkelkompensation wie das Anforderungsdefinitionsdokument aufgebaut ist. Das vollständige Anforderungsdefinitionsdokument befindet sich im Anhang A.

Anforderung Schwimmwinkelkompensation aus dem Anforderungsdefinitionsdokument 4-1

Funktion	Schwimmwinkelkompensation
Anforderungsart	Zielforderung
ID	003
Priorität	hoch
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	mittel
Beschreibung	Ausregelung des Schwimmwinkels. Ein Schwimmwinkel kann sich einstellen durch die Asymmetrie des Messanhängers, durch Abbremsung des Messrades, durch Seitenwind und durch Einstellung eines Radlenkwinkels des Messrades zur Schräglaufwinkelerzeugung. Die Ausregelung erfolgt über den Radlenkwinkel des Referenzrades.
Eingabe	Schräglaufwinkel des Messrades Radlenkwinkel des Messrades Radlenkwinkel des Referenzrades
Quelle	Winkel aus Correxit Potentiometer der Radlenkwinkelverstellung am Messrad Potentiometer der Radlenkwinkelverstellung am Referenzrad
Ausgabe	Veränderung des Radlenkwinkels des Referenzrades
Senke	Motor zur Radlenkwinkelverstellung am Referenzrad
Ziel	Der Schwimmwinkel wird minimiert.

Die entsprechend dem W-Modell folgende Validierungsvorschrift wurde nach Beendigung der Anforderungsanalyse erstellt und befindet sich in diesem Dokument im Kapitel 10.

5 Entwurf

5.1 Grobentwurf

5.1.1 Ziel des Grobentwurfs

Ziel des Grobentwurfs, auch Architekturentwurf genannt, ist die Aufstellung eines grundlegenden strukturellen Rahmenwerkes der Software. Dieses beinhaltet die Bestimmung der Hauptkomponenten des Systems und zwischen ihnen stattfindende Kommunikation, gegebenenfalls ist die Aufteilung auf Prozessoren vorzunehmen. Ergebnis des Grobentwurfs ist eine Beschreibung der Softwarearchitektur.

5.1.2 Methodik des Grobentwurfs

Da es sich bei dem verwendeten CRIO um ein verteiltes System mit mehreren Prozessoren handelt baut der Grobentwurf auf einer verteilten Mehrprozessorarchitektur auf. Zur besseren Übersicht und des besseren Verständnisses wird eine grafische Darstellung gewählt, in der UML Notationen verwendet werden. Des Weiteren werden die Schnittstellen beschrieben und die Auswahl der Kommunikationsmethoden erläutert. Bei der Erstellung des Grobentwurfs werden die Hauptfunktionen so auf die vorhandenen Teilsysteme verteilt, dass möglichst wenig Datenkommunikation stattfindet und das Realsystem, welches die langsamste Komponente ist, entlastet wird.

5.1.3 Software- und Hardwarearchitektur

Das verteilte System besteht aus den folgenden Subsystemen:

- **Field Programmable Gate Array (FPGA):**

Ein FPGA ist ein frei programmierbarer Logikschaltkreis. Mit ihm können digitale Schaltungen beliebiger Funktionalität realisiert werden, die sonst aus einzelnen Logikgattern aufgebaut werden müssen. Dieses FPGA kann, wie die anderen Subsysteme, mit LabVIEW programmiert werden. Der Vorteil eines FPGAs ist, dass die programmierte Software in Hardware überführt wird und somit echte parallele Abarbeitung möglich ist. Der Nachteil ist, dass nur Grundfunktionen zur Programmierung bereitstehen und nur in Integer bzw. Boolean gerechnet werden kann. Daher werden die

verwendeten Größen erst in Integer umgerechnet. Der hier verwendete FPGA hat einen 40MHz Prozessor.

- **Realtime System (RTS):**

Das RTS wird im Folgenden auch Broker genannt da es die Aufgabe eines Vermittlers übernimmt. Das RTS verwendet ein eigenes Echtzeitbetriebssystem und kann somit deterministische Prozesse ausführen. Das RTS verwendet einen 200MHz Prozessor.

- **Host:**

Das Hostsystem ist das System das sich auf dem Laptop im Zugfahrzeug befindet und die Steuerung der Software sowie die Anzeige und Speicherung der Messdatenspeicherung übernimmt. Als Hostsystem sollen verschiedene Laptops verwendet werden können.

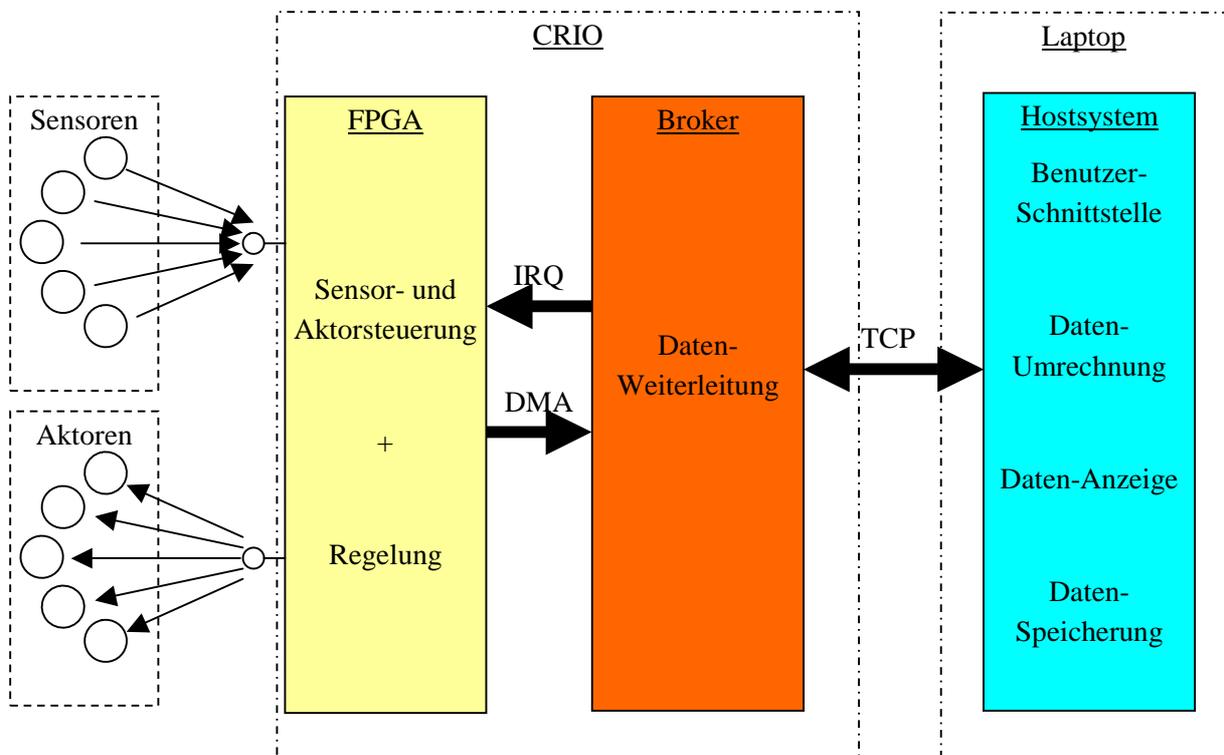


Abbildung 5.1: Softwaresystemarchitektur

5.1.3.1 Zuweisung der Hauptfunktionen auf die Teilsysteme

Bei der Zuweisung der Hauptfunktionen auf die Teilsysteme wurde so vorgegangen, dass zum einen die Datenkommunikation so gering wie möglich gehalten wird und zum anderen die Berechnungen auf dem Broker so gering wie möglich gehalten werden, da das Realtime System das langsamste Teilsystem ist.

5.1.4 Schnittstellenbeschreibung

5.1.4.1 Schnittstelle zwischen Sensoren/Aktoren und FPGA

Die Sensoren werden über digitale bzw. analoge Eingangsmodule und die Aktoren über digitale bzw. analoge Ausgangsmodule direkt verbunden. Diese Module werden in das CRIO-Gehäuse in die dafür vorgesehenen Slots gesteckt. Damit gibt es eine direkte Verbindung zwischen Sensoren/Aktoren und dem FPGA.

5.1.4.2 Schnittstelle zwischen FPGA und RTS

Es stehen 3 Möglichkeiten zur Datenübertragung zwischen FPGA und Broker zur Verfügung: Polling, Interruptkommunikation und Direktspeicherzugriff (DMA).

Funktionsweise und Bewertung:

- Polling: Beim Polling „fragt“ der CRIO-Host periodisch, ob der FPGA etwas „senden möchte“. Es ist sehr rechenzeitintensiv, da für einen einzigen Übertragungsvorgang mehrmals gepollt wird.
- Interrupt: Bei der Interruptkommunikation sendet der FPGA eine Unterbrechungsanfrage an die CRIO-Host-CPU, daraufhin unterbricht die CPU die aktuelle Abarbeitung und nimmt den Datenwert entgegen. Die Interruptkommunikation ist weniger rechenzeitintensiv als Polling, ist aber durch die Blockierung des Prozessors ebenfalls langsam, wenn häufig Daten übertragen werden.
- DMA: Bei DMA kann der FPGA ohne Umweg über die CRIO-Host-CPU direkt mit dem Arbeitsspeicher kommunizieren. Der Vorteil der DMA-Technik ist die schnellere Datenübertragung bei gleichzeitiger Entlastung des Prozessors.

Da die Messdaten mit hoher Frequenz (>1kHz) erfasst werden sollen und ein Hauptaugenmerk auf der Optimierung der Abtastrate liegt ist DMA-Kommunikation zu bevorzugen. Der entwickelte Prototyp der Interrupt zur Kommunikation nutzt, bestätigte, dass mit der Interruptkommunikation keine Abtastraten über 100Hz auf dem verwendeten CRIO erreicht werden können. Während der Programmierung hat sich herausgestellt, dass die DMA-Kommunikation des verwendeten CRIO nur in die eine Richtung vom FPGA zum RTS unterstützt wird aber nicht umgekehrt. Dies wurde auch vom technischen Support von National Instruments bestätigt. Somit wird für die Kommunikation vom RTS zum FPGA auf die Interruptkommunikation ausgewichen.

5.1.4.3 Schnittstelle zwischen RTS und Host

Durch Verwendung der 100Base-TX Ethernetschnittstelle, IEEE 802.3u, können 100Mbit/s übertragen werden. CRIO bietet neben TCP auch UDP zur Datenübertragung zwischen CRIO-Host und Host, beide Transportprotokolle nutzen IP als Vermittlungsschicht.

Funktionsweise und Bewertung:

- TCP: ist ein verbindungsorientiertes, zuverlässiges Transportprotokoll.
 - Vorteil: Garantie, dass gesendete Pakete in richtiger Reihenfolge ankommen
 - Nachteil: Höherer Overhead als UDP, d.h. längerer Verbindungsaufbau
- UDP: ist ein verbindungsloses, unzuverlässiges Transportprotokoll.
 - Vorteil: schnellerer Verbindungsaufbau, gut wenn nur kleine Datenmengen ausgetauscht werden
 - Nachteil: Keine Garantie dass Pakete ankommen, nicht Reihenfolge erhaltend

Da für die Messdatenaufzeichnung die Erhaltung der Reihenfolge der Messdaten notwendig ist und keine Daten verloren gehen dürfen, kann nur TCP verwendet werden.

Überschlagsrechnung ob 100Mbit/s ausreichend sind:

Die Überschlagsrechnung erfolgt nach folgender Formel:

$$v_{\ddot{u}} = f * K * G \quad (5.1)$$

minimale Abtastrate: $f = 1000\text{Hz} = 1000 \text{ Werte/s}$

Datengröße: $G = (\text{double}) = 32\text{bit}$

maximale Kanalanzahl: $K = 50$ (32 analog, 8 digital, 10 extra)

bei $f = 1000\text{Hz}$ $v_{\ddot{u}} = 1,6\text{Mbit/s}$

bei $f = 10\text{kHz}$ $v_{\ddot{u}} = 16\text{Mbit/s}$

Das heißt selbst bei einer Frequenz von 10kHz und 50 Kanälen sind 100Mbit/s ausreichend.

5.2 Feinentwurf

5.2.1 Ziel des Feinentwurfs

Ziel des Feinentwurfs, ist die Detaillierung der Hauptkomponenten auf Basis der zugewiesenen Funktionen durch Aufteilung in Teilmodule. Ergebnis des Feinentwurfs ist eine detaillierte Beschreibung der Hauptkomponenten.

5.2.2 Methodik des Feinentwurfs

Zur Entwicklung des Feinentwurfs stehen einige Methoden zur Verfügung. Diese lassen sich einer folgenden Kategorie zuordnen, die „natürliche Sprache“, Pseudo-Code und grafische Notationen. Da der Feinentwurf übersichtlich und einfach zu verstehen sein soll, sind der Pseudo-Code sowie ausschließlich „natürliche Sprache“ ungeeignet. Da LabVIEW eine strukturierte Programmiersprache ist werden auch Nassi-Shneiderman-Diagramme betrachtet, welche keiner der oben genannten Kategorien eindeutig zugewiesen werden können. Allerdings würde durch die Komplexität der verteilten Software ein Nassi-Shneiderman-Diagramm als Feinentwurf sehr schnell unübersichtlich. Aus gleichem Grund ist auch die grafische Notation eines Flussdiagramms ungeeignet. Daher werden die UML-Klassendiagramme zur Darstellung des Feinentwurfs verwendet, auch wenn LabVIEW nicht objektorientiert ist und nicht mit Klassen arbeitet. Die Darstellung des Feinentwurfs durch ein Klassendiagramm ist auch für nicht Informatiker verständlich und bietet dennoch die Möglichkeit der Detaillierung. Die Klassendiagramme können, sofern für das bessere Verständnis nötig, durch weitere UML-Diagramme und eine kurze Beschreibung in „natürlicher Sprache“ ergänzt werden.

5.2.3 Feinentwurf des FPGA

Das FPGA Modul wird in die Submodule Schräglaufeinstellung, Schwimmwinkelkompensation, Drehzahlberechnung, Konfiguration sowie High Priority Loop (HPL) und Low Priority Loop (LPL) aufgeteilt (siehe Kapitel 6.3.5).

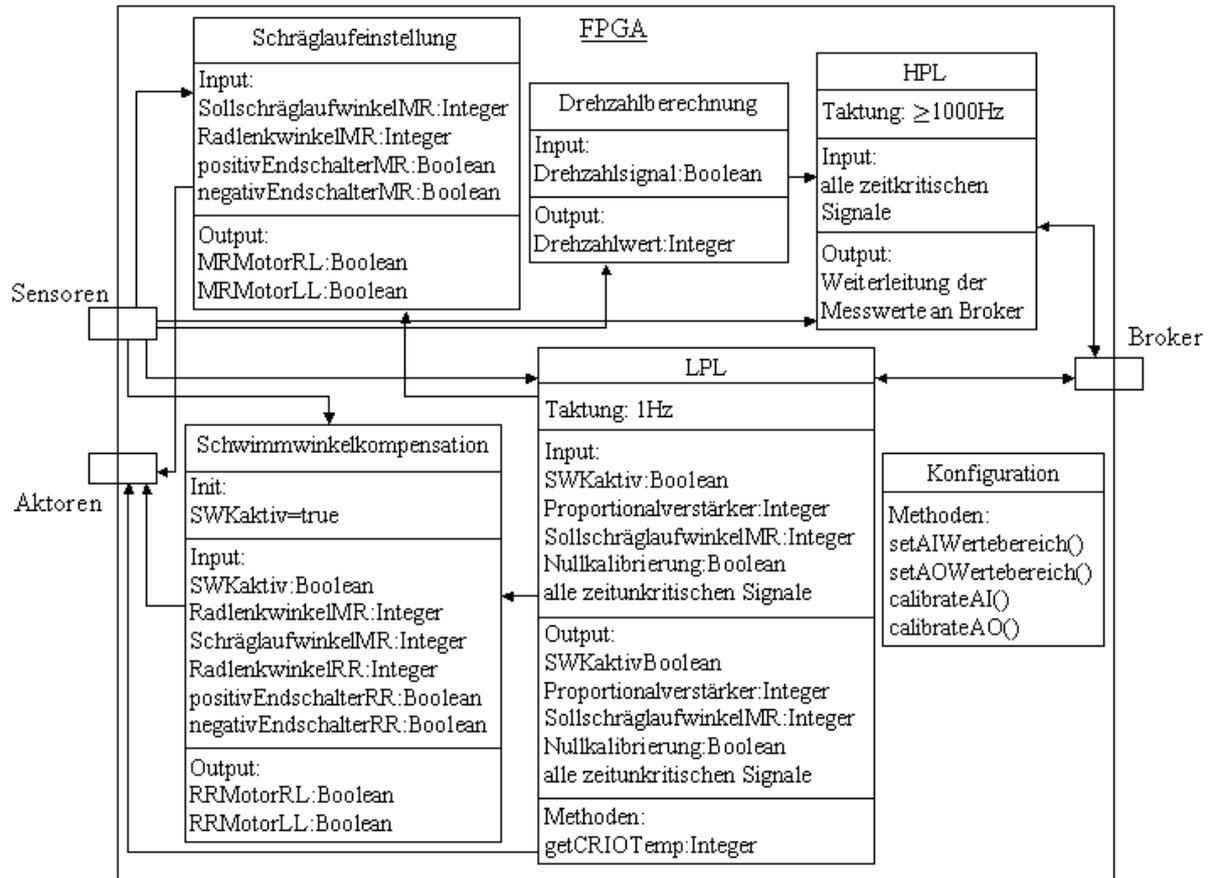


Abbildung 5.2: FPGA-Feinentwurf

Die Abkürzung MR steht für Messrad und die Abkürzung RR für Referenzrad. In dem Konfigurationsmodul werden die Einstellungen der analogen Hardwaremodule die im CRIO stecken vorgenommen und deren Kalibrierdaten ausgelesen. Das Schräglaufeinstellungsmodul vergleicht Soll- und Itschräglaufwinkel und steuert die Motoren an die den Radlenkwinkel des Messrades und damit den Schräglaufwinkel verstellen. Die Schwimmwinkelkompensation berechnet aus den Messgrößen Schräglaufwinkel und Radlenkwinkel des Messrades den Schwimmwinkel und kompensiert diesen indem es die Motoren der Radlenkwinkelverstellung des Referenzrades ansteuert.

Da die Messgrößen mit möglichst hoher Frequenz erfasst werden sollen werden im Hinblick auf die Optimierung Messgrößen in 2 Klassen unterschiedlicher Priorität aufgeteilt. Das HPL-Modul erfasst die Messgrößen und das LPL-Modul erfasst die Kontrollgrößen (= Messgrößen niedriger Priorität). Diese werden jeweils an den Broker weitergegeben. Außerdem nimmt das LPL-Modul die Steuerbefehle des Brokers entgegen, die mittels Interruptkommunikation übermittelt werden.

5.2.4 Feinentwurf des Broker

Der Broker wurde in ein zeitkritisches Modul (time critical Loop = TCL) und in ein zeitunkritisches Modul (normal priority loop = NPL) aufgeteilt um die Messdaten schnellstmöglich aus dem DMA-Puffer lesen zu können. Für die TCL besteht die Forderung, dass sie schnell genug ausgeführt wird, um die Messdaten aus dem DMA-Puffer zu lesen bevor dieser überläuft. Die Schleife mit normaler Priorität ist für die Netzwerkkommunikation zuständig. Sie baut die TCP/IP Kommunikation auf und erstellt die FIFO-Puffer die für die Kommunikation zwischen NPL und TCL verwendet werden. Da der Broker der „Flaschenhals“ des verteilten Systems hinsichtlich der Verarbeitungsgeschwindigkeit ist, werden die Daten auf dieser Ebene nur weitergeleitet und erst auf dem Hostsystem in die richtigen Größen umgerechnet.

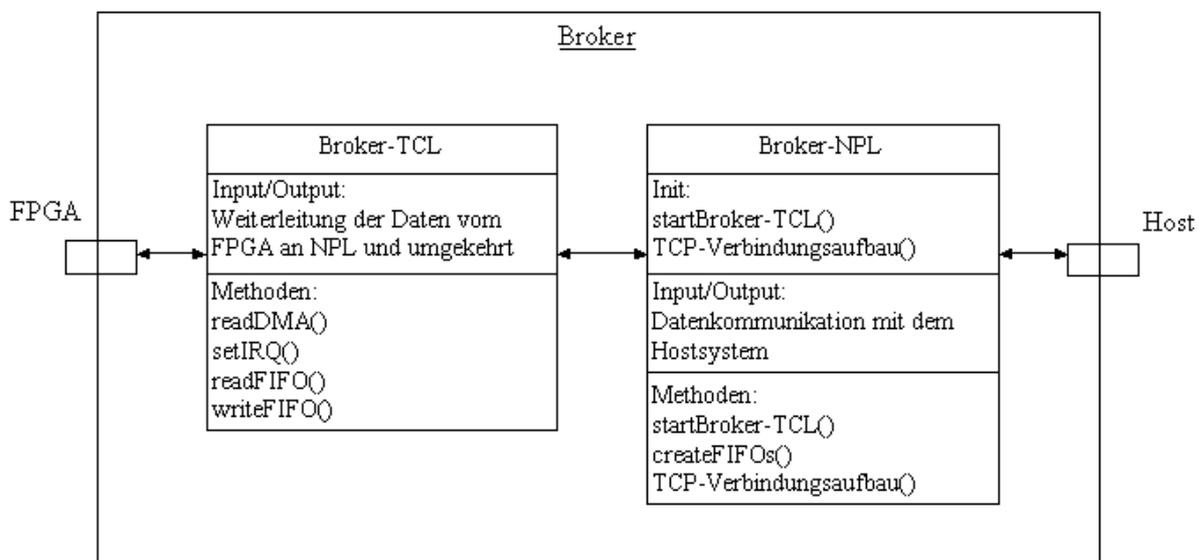


Abbildung 5.3: Broker-Feinentwurf

5.2.5 Feinentwurf des Host

Die Hostkomponente besteht aus den Submodulen Hauptprogramm, Einstellungen, Kalibrierung, Statusanzeige, Warnung, R_{dyn} -Berechnung, Messung und Vorauswertung.

Das Hauptprogramm enthält die grafische Benutzerschnittstelle (GUI) und dient zur Menüführung durch die Unterprogramme. Zudem werden Element vorgesehen um die Schwimmwinkelkorrektur abzuschalten und die Software zu beenden.

Das Untermenü „Einstellungen“ erlaubt dem Benutzer Einstellungen vorzunehmen. Hier wird unter anderem der Pfad angegeben wo die Konfigurationsdatei und die Messdateien gespeichert werden und der Messdateiname. Werden mehrere Messungen in Folge gemessen, werden diese durchnummeriert und unter Messdateiname und Nummerierung gespeichert.

In dem Untermenü „Kalibrierung“ werden alle wichtigen Kalibrierdaten angegeben die verwendet werden.

Das Submenü „ R_{dyn} -Berechnung“ erhält als Eingangsdaten das Messdatenarray. Als erstes wird die Sollgeschwindigkeit, die vom Benutzer eingegeben wird, eingelesen. Die R_{dyn} -Messung beginnt sobald der entsprechende Button betätigt wurde. Die R_{dyn} -Berechnung startet aber erst wenn ein Messdatenarray ankommt dessen Elemente alle innerhalb der Geschwindigkeits- und Schräglaufwinkeltoleranz liegen. Danach wird der berechnete dynamische Reifenhalmmesser dieser Geschwindigkeit ausgegeben.

Das Untermenü Messung erhält als Eingangsdaten das Messdatenarray, den berechneten dynamischen Reifenhalmmesser und die Information ob die Messfelge verwendet wird. Zu Beginn wird die Messgeschwindigkeit und der Sollschräglaufwinkel eingelesen. Wird die Messung durch den Benutzer gestartet wird zuerst der Schräglaufwinkel eingestellt. Die Messdatenaufzeichnung beginnt, sobald die Geschwindigkeit innerhalb des Geschwindigkeitstoleranzbereichs und der Schwimmwinkel (Sollschräglaufwinkel – Schräglaufwinkel) innerhalb des Schwimmwinkeltoleranzbereichs liegt. Ist eine einstellbare Zeit abgelaufen wird die nächste Bremsmomentstufe eingestellt und die Messaufzeichnung wird fortgesetzt, sobald die obigen Bedingungen wieder erfüllt sind. Wurden alle Stufen durchfahren wird die Messung beendet. Konnten nicht alle Stufen innerhalb der Toleranzen durchfahren werden, kann der Benutzer die Messung beenden.

Das Submenu Statusanzeige wird von den Modulen R_{dyn} -Berechnung und Messung verwendet. Die Statusanzeige zeigt die aktuelle Geschwindigkeit an und informiert den Benutzer über den aktuellen Zustand des Systems, indem die Kontrollgrößen angezeigt werden.

Das Menü Warnung wird von dem Modul Statusanzeige verwendet und warnt den Benutzer beim Überschreiten von Grenzen. Wird eine kritische Grenze überschritten (Batterieladezustand zu gering bzw. CRIO-Temperatur zu hoch), wird das System kontrolliert heruntergefahren.

Das Untermenü Vorauswertung erlaubt die Vorauswertung der Messdaten. Zunächst wird geprüft ob die Messwerte in die Datei geschrieben wurden. Danach wird pro Bremsmomentstufe ein Punkt in ein 2-Dimensionales Diagramm (Reibwert über Schlupf) geplottet.

Die nachfolgende Abbildung zeigt den beschriebenen Feinentwurf des Hosts. Dabei steht die verwendete Nummerierung für die Reihenfolge der Schritte im Messbetrieb.

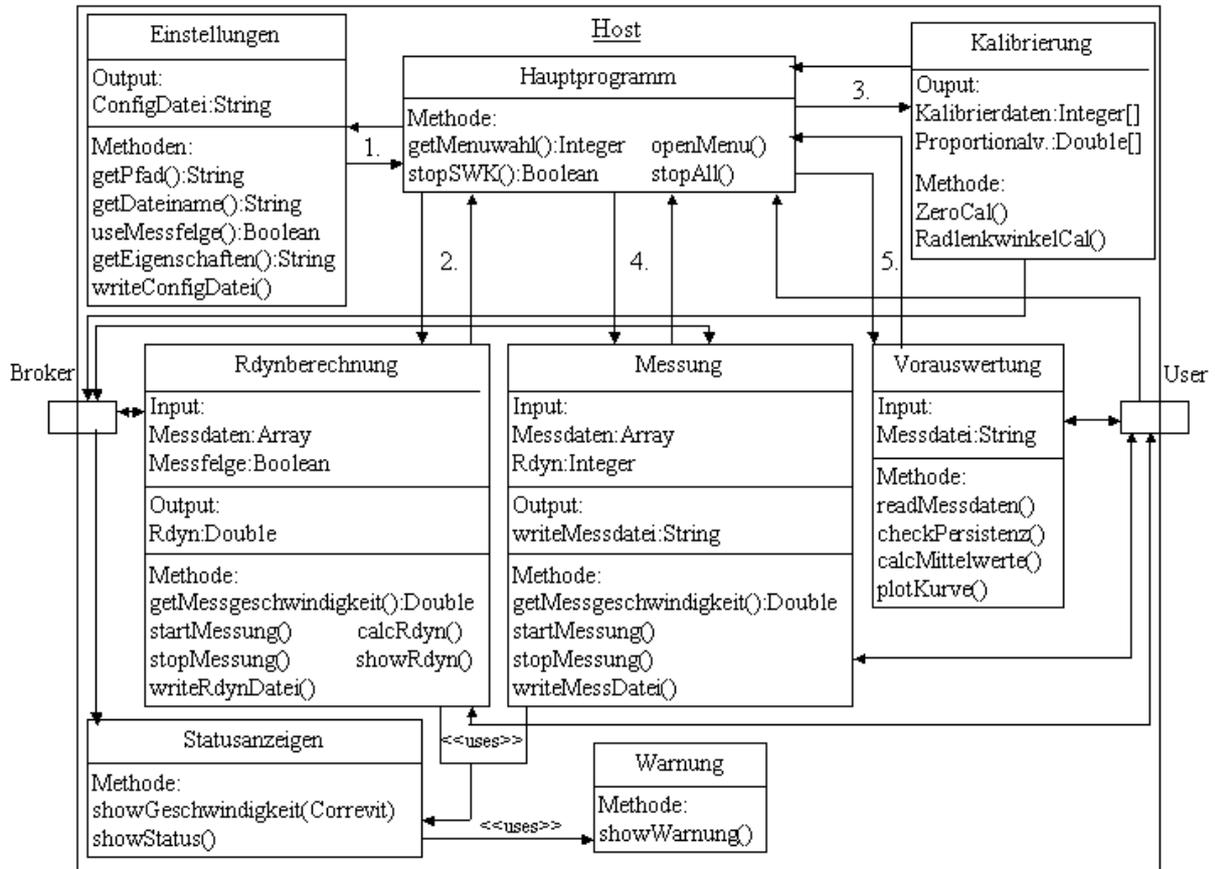


Abbildung 5.4: Host-Feinentwurf

6 Modulspezifikation

6.1 Ziel der Modulspezifikation

Ziel der Modulspezifikation ist eine genaue Beschreibung eines jeden Moduls zu erstellen.

Die Modulspezifikation besteht aus einer kurzen verbalen Beschreibung der Modulfunktionalität, sowie der Definition der notwendigen Algorithmen in einer entsprechenden Darstellungsform.

Ergebnis der Modulspezifikation ist eine genaue Beschreibung eines jeden Moduls, wodurch die Implementierung deutlich erleichtert werden soll. In dieser Arbeit wird Modulspezifikation und Modulentwurf synonym verwendet.

6.2 Methodik der Modulspezifikation

Die zur Programmierung verwendete Software LabVIEW ist eine grafische Programmiersprache. In LabVIEW werden Datenflüsse abgebildet und mit Operationen verknüpft. Daher werden die Modulspezifikationen in Flussdiagrammen oder in UML-Aktivitätsdiagrammen abgebildet, da diese die Datenflussprogrammierung am besten abbilden und somit in dieser Form am einfachsten umzusetzen sind. Flussdiagramme sowie UML-Aktivitätsdiagramme sind durch ihre grafische Darstellung sehr einfach zu verstehen. Da Flussdiagramme nach DIN 66001 nicht alle benötigten Strukturen bereitstellen, werden zur Modulspezifikation UML-Aktivitätsdiagramme verwendet. Die verwendete Notation weicht von der Standard UML-Aktivitätsdiagrammnotation ab. Daher werden in der nachfolgenden Abbildung die verwendeten Notationen erklärt. In der gewählten Darstellung befindet sich über jedem Aktivitätsdiagramm eine Kopfzeile, die die verwendeten Objekte aus dem Klassendiagramm enthält, welche in dem Aktivitätsdiagramm verwendet werden. Somit lassen die Aktivitätsdiagramme mit den UML-Klassendiagrammen der Feinentwürfe kombinieren. Die folgende Grafik zeigt eine Zeichenerklärung aller verwendeten Notationen. Wird innerhalb der Kopfzeile das Zeichen → verwendet, bedeutet dieses, dass die vor dem Pfeil stehende Variable im Aktivitätsdiagramm durch die entsprechende Abkürzung die hinter dem Pfeil steht verwendet wird.

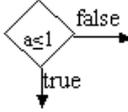
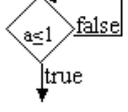
Modulname					
Init: Initialisierung einer Variablen mit einem Wert	Input: Eingabeschnittstelle des Moduls	Output: Ausgabeschnittstelle des Moduls	Methoden: Methoden die im Modul verwendet werden		
	Taktung		Beginn	Bezeichner:Typ	Variable vom Typ
	Anweisung		Ende	Bezeichner():Typ	Methode mit Rückgabewert vom Typ
	Bedingung		Verbindung	//Kommentar	Kommentar
	Schleife		unidirektionale Verbindung		
			bidirektionale Verbindung		
	Notiz		Verweis		

Abbildung 6.1: Zeichenerklärung der verwendeten Symbole

6.3 Modulspezifikationen der FPGA – Module

6.3.1 Konfiguration

Dieses Softwaremodul dient dazu die eingesteckten Hardwaremodule zu konfigurieren, damit diese entsprechend angeschlossen, verkabelt und verwendet werden können. Somit bildet dieses Modul die Grundlage für die Kommunikation des FPGAs mit der angeschlossenen Elektronik.

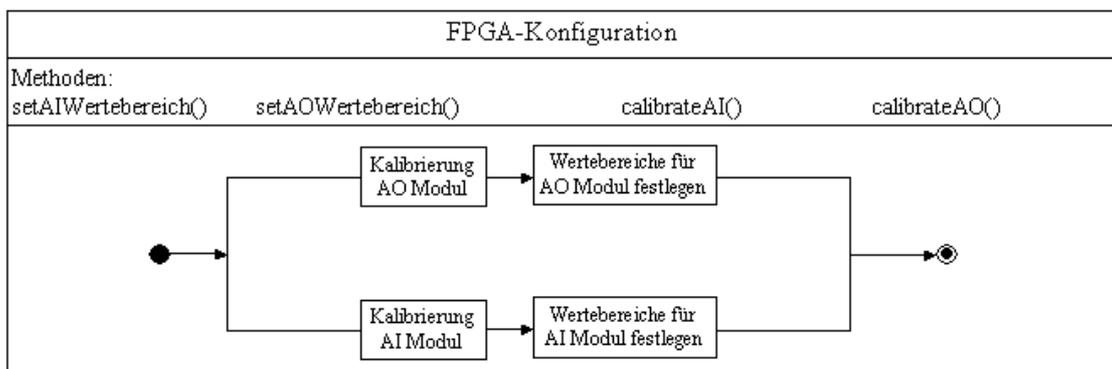


Abbildung 6.2: FPGA-Konfiguration – Modulspezifikation

In dem Modul Konfiguration, dessen Spezifikation in der vorherigen Abbildung zu sehen ist, werden die Kalibrierdaten der Analogmodule ausgelesen und die Wertebereiche für die verschiedenen Kanäle, sowie die Messart festgelegt. Die nachfolgende Grafik zeigt die Modulspezifikation des Moduls Konfiguration. Die Abkürzung AI steht hierbei für Analoginput und AO für Analogoutput.

6.3.2 Schräglafeinstellung

6.3.2.1 Methodik und Vorgehen

Am Messrad soll ein vom Benutzer gewünschter Sollschräglauf eingestellt werden können.

Prinzipiell gibt es zwei Möglichkeiten, den entsprechenden Schräglaufwinkel am Messrad einzustellen:

1. Möglichkeit (A1):

Am Messrad wird ein Radlenkwinkel eingestellt, der dem Sollschräglaufwinkel entspricht. Die parallel laufende Schwimmwinkelkompensation regelt den Schwimmwinkel aus. Ist der Schwimmwinkel kompensiert, entspricht der Radlenkwinkel am Messrad dem Schräglaufwinkel am Messrad.

Vorteil: Einfache Lösung.
 Keine Messfehler durch aktive Schräglaufregelung.
 Keine gegenseitige Beeinflussung parallel laufender Regelungen.

Nachteil: Vor der Messung wird gefordert, dass der Schwimmwinkel kompensiert ist.

2. Möglichkeit (A2):

Am Messrad wird der Schräglaufwinkel, welcher vorgegeben wurde, durch Verstellung des Radlenkwinkels am Messrad eingeregelt. Ist nun der Schwimmwinkel außerhalb eines Toleranzbereiches um $\beta = 0^\circ$, regelt die parallel laufende Schwimmwinkelkompensation den Schwimmwinkel durch eine Radlenkwinkelverstellung des Referenzrades aus. Diese Regelung ändert den Schräglauf des Messrades, deshalb wird der Radlenkwinkel des Messrades nachgeregelt werden.

Vorteil: Erlaubt Messbeginn bevor der Schwimmwinkel kompensiert wurde.

Nachteil: Es kann zu Messfehlern kommen, da auch während der Messung geregelt wird.
 Beide Regelungen beeinflussen sich gegenseitig.

Aufgrund der überwiegenden Vorteile der 1.Möglichkeit wird diese verwendet.

Für die Einstellung des Radlenkwinkels am Messrad gibt es 2 Varianten:

B1. Den Motor der Radlenkwinkelverstellung für eine des einzustellenden Radlenkwinkels entsprechenden Zeit aktivieren. Diese Variante ist ungenau, da, abhängig vom Moment das der Motor erzeugt, die Zeitdauer für die Verstellung des gleichen Winkels unterschiedlich sein kann.

B2. Einregelung des Radlenkwinkels über Vergleich von Soll- und Istradlenkwinkel und entsprechende Motorverstellung.

Für Einregelung des Radlenkwinkels gibt es verschiedene Regler zur Auswahl. Grundsätzlich werden Regler in stetige (C1) und unstetige Regler (C2) eingeteilt. In diesem Fall kommen stetige Regler, wie z.B. PID, nicht in Frage, da die Stellgröße unstetig ist, da die Verfahrensgeschwindigkeit bei maximaler Leistung noch sehr gering ist, und ein weiteres Absenken der Geschwindigkeit nicht erwünscht ist. Bei den unstetigen Reglern bleibt die Wahl zwischen Zweipunktregler (D1), Dreipunktregler (D2) und Fuzzyregler (D3). Da es nur eine Stellgröße mit 2 Werten gibt, nämlich Motor an und Motor aus und die Motoren nur eine geringe Dynamik aufweisen ist der Zweipunktregler die richtige Wahl.

Fazit:

Am Messrad wird ein Radlenkwinkel eingestellt, der dem Sollschräglaufwinkel entspricht. Die parallel laufende Schwimmwinkelkompensation regelt den Schwimmwinkel aus. Ist der Schwimmwinkel vollständig kompensiert, entspricht der Radlenkwinkel am Messrad dem Schräglaufwinkel am Messrad.

Bei der Einstellung des erforderlichen Radlenkwinkels wird ein Zweipunktregler verwendet. Sobald Soll- und Istradlenkwinkel eine Differenz von mehr als $\Delta\delta_{MR}$ aufweisen wird der entsprechende Motor aktiviert, bis ein Radlenkwinkel gemessen wird der innerhalb des Toleranzbereichs liegt. Die Abbildung 6.3 zeigt den Entscheidungsbaum der Lösungsfindung.

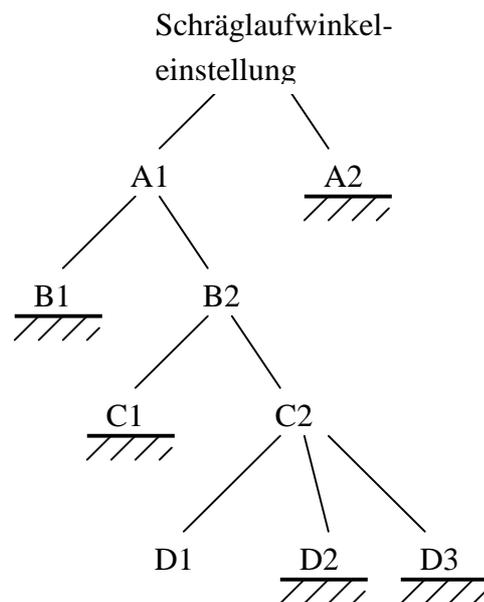


Abbildung 6.3: Entscheidungsbaum der Schräglaufwinkeleinstellung

6.3.2.2 Modulspezifikation

Die nachfolgende Abbildung zeigt den Modulentwurf des Schräglaufeinstellungsmoduls. Die angegebene Frequenz verdeutlicht nur, dass die Verstellung des Radlenkwinkels abhängig

von der verwendeten Frequenz ist. Dabei bedeutet eine kleine Frequenz eine Verstellung in großen Schritten während eine große Frequenz kleinere Schritte ermöglicht. In der Notiz ist überschlüssig berechnet wie groß die Winkelverstellung am Rad innerhalb eines Schleifendurchlaufs ist, bei entsprechend eingestellter Frequenz. Hierbei sei auch erwähnt, dass die Verstellung auch von der Stromstärke der Gleichstrommotoren abhängig ist welche einzeln in der Elektronik eingestellt werden kann. Die Wahl der Frequenz bleibt dem Benutzer überlassen.

Zunächst wird die Differenz zwischen einzustellendem und eingestelltem Radlenkwinkel am Messrad berechnet. Liegt diese innerhalb der angegebenen Schräglaufwinkeltoleranz, sind die Motoren aus. Befindet sich diese Differenz außerhalb dieser Toleranz, wird der entsprechende Motor aktiviert, sofern weder dessen Endschalter noch dessen in der Software abgelegten Grenzen erreicht sind. Die beiden Grenzen die zu Beginn der Softwareentwicklung mit $\pm 10^\circ$ angenommen wurden, wurden nach Durchführung der Radlenkwinkelkalibrierung siehe Kapitel 9.4.4 durch die entsprechenden Werte ersetzt. Hierbei sei nochmals darauf hingewiesen, dass der FPGA nur mit Integer und Boolean rechnen kann, daher werden alle Werte auf einer höheren Ebene, dem Host, in die entsprechenden Fließkommawerte umgerechnet.

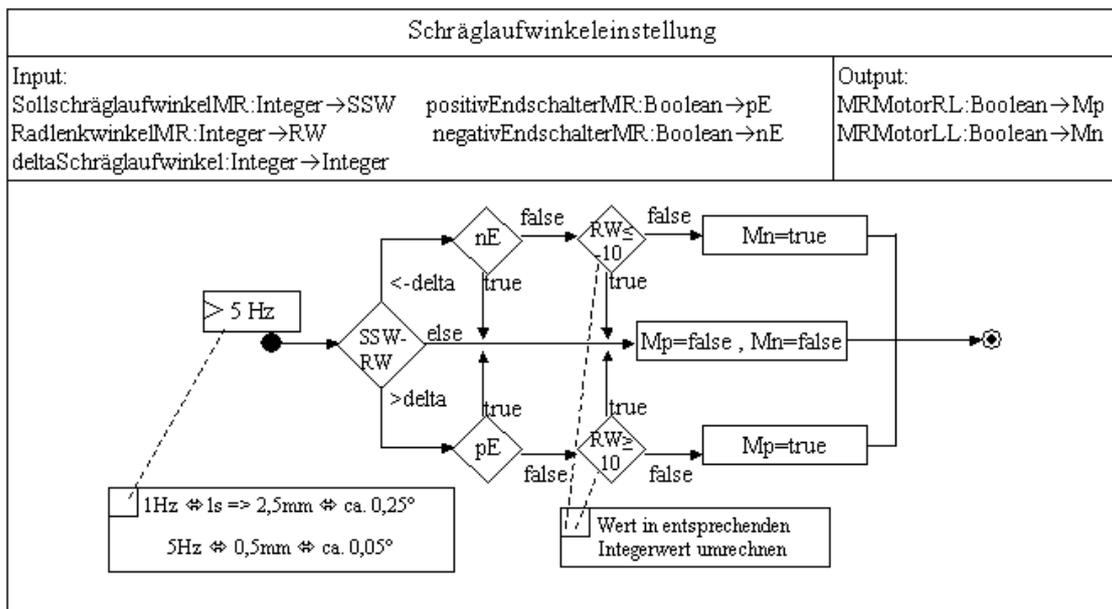


Abbildung 6.4: Schräglaufeinstellung – Modulspezifikation

6.3.3 Schwimmwinkelkompensation

6.3.3.1 Methodik und Vorgehen

Zunächst werden die Begriffe Schwimmwinkel, Schräglaufwinkel und Radlenkwinkel definiert:

Schräglaufwinkel (α): Ist der Winkel zwischen der Radlängsachse und der Bewegungsrichtung.

Schwimmwinkel (β): Ist der Winkel zwischen der Längsachse des Messanhängers und der Bewegungsrichtung des Zugfahrzeugs.

Radlenkwinkel (δ): Ist der Winkel zwischen der Radlängsachse und der Längsachse des Messanhängers.

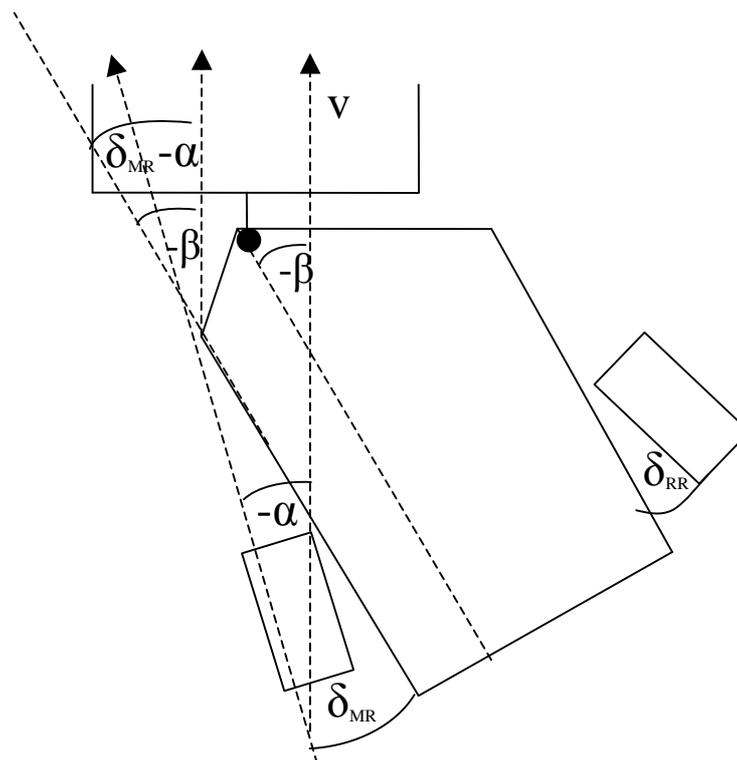


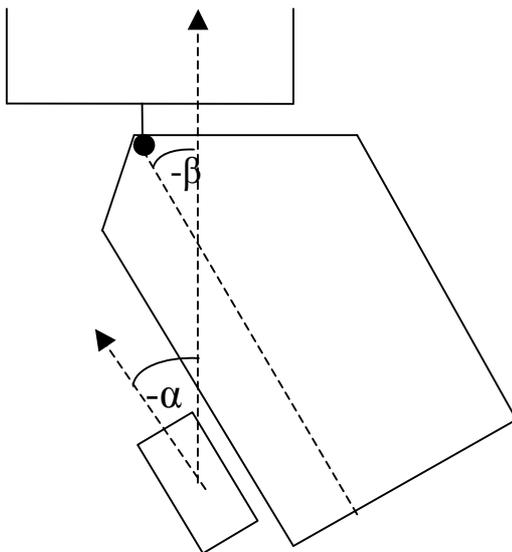
Abbildung 6.5: Beziehung der Winkel α , β , δ

Auftreten eines Schwimmwinkels:

Ein Schwimmwinkel kann sich einstellen durch Fahrwiderstände, Abbremsung des Messrades, Einstellung eines Schräglaufwinkels am Messrades und Seitenwind. Kompensiert wird der Schwimmwinkel durch Einstellung eines Schräglaufwinkels am Referenzrad.

Um eine automatische Schwimmwinkelkompensation zu entwerfen wird im Folgenden zum einfacheren Verständnis eine Fallunterscheidung gemacht. Es wird zunächst der einfachere Fall betrachtet bei dem sich ein Schwimmwinkel einstellt, ohne dass ein Radlenkwinkel am Messrad eingestellt ist. Als 2. Fall wird der Fall betrachtet bei dem sich ein Schwimmwinkel einstellt wenn ein Radlenkwinkel am Messrad eingestellt ist.

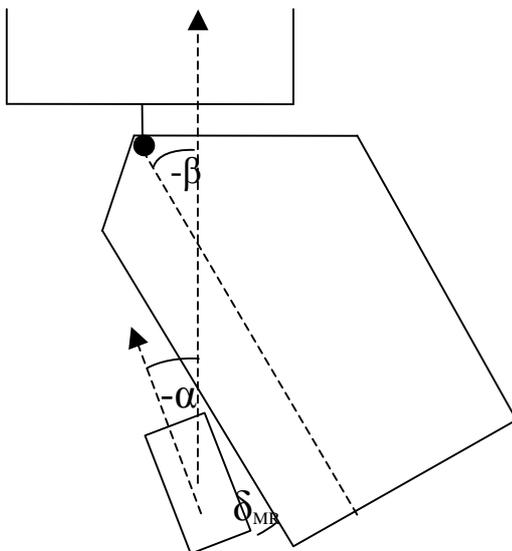
1. Fall (kein Radlenkwinkel am Messrad eingestellt):



$$\delta \quad \beta = \alpha$$

$$\delta \quad \delta_{RR} \text{ vergrößern bis } \beta = 0$$

2. Fall (Radlenkwinkel am Messrad eingestellt):



$$\delta \quad \beta = \alpha - \delta_{MR}$$

$$\delta \quad \delta_{RR} \text{ vergrößern bis } \beta = 0$$

Da beide Fälle die gleiche Beziehung beschreiben, der 1. Fall ist der 2. Fall mit $\delta_{MR} = 0$, lässt sich die Regelung einfach durch $\beta = \alpha - \delta_{MR} = 0$ erreichen.

Der Radlenkwinkel (δ_{MR}) am Messrad entspricht der Potentiometerstellung des Potentiometers, das die Auslenkung des Messrades misst.

Der Schräglaufwinkel (α) wird von dem Correvitsensor erfasst.

Eine weitere Möglichkeit den Schwimmwinkel auszuregeln ist einen zweiten Correvitsensor auf der Anhängerlängsachse zu verbauen, der den Schwimmwinkel direkt misst. Somit könnte der Schwimmwinkel genauer ausgeregelt werden, da im Vergleich zur verwendeten Variante keine Kalibrierungsfehler des Radlenkwinkels eingehen. Da es nicht der Anforderung entspricht den Schwimmwinkel genau auf 0° auszuregeln wurde dies als nicht nötig erachtet.

6.3.3.2 Modulspezifikation

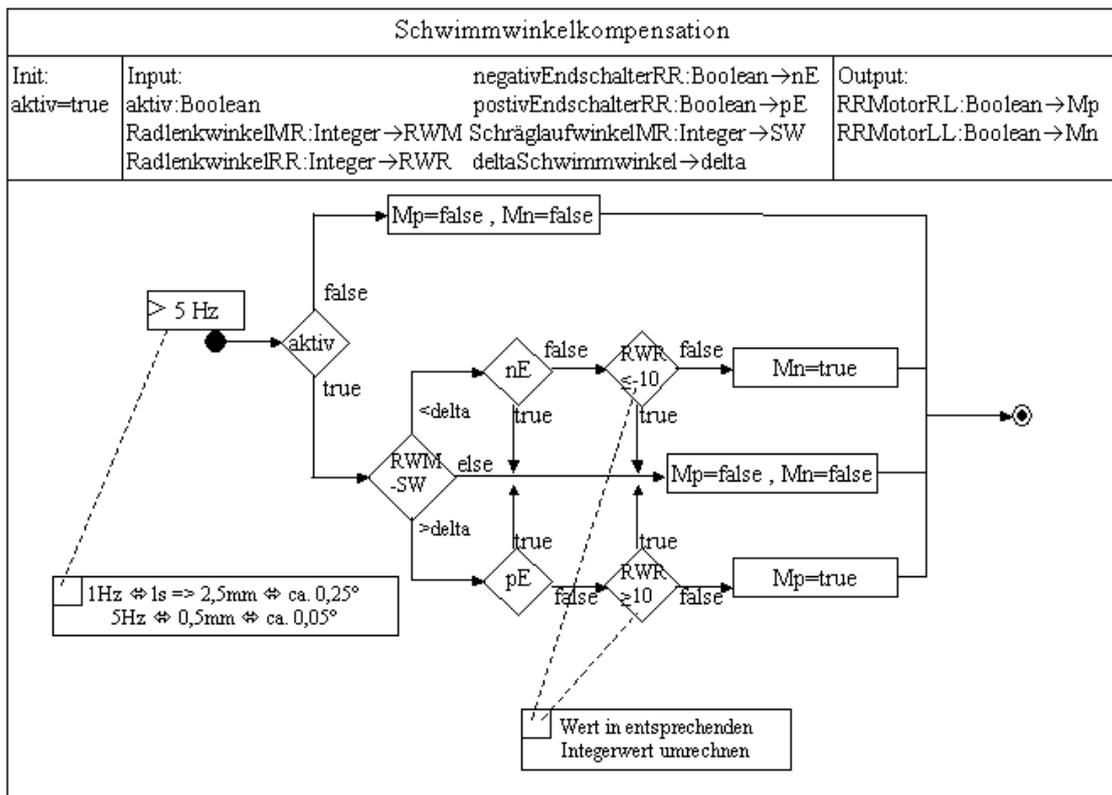


Abbildung 6.6: Schwimmwinkelkompensation – Modulspezifikation

Aus gleichem Grund wie bei der Wahl des Reglers bei der Schräglaufwinkeleinstellung wird auch für die Schwimmwinkelkompensation ein 2-Punktregler implementiert. Die vorherige Abbildung zeigt die Modulspezifikation der Schwimmwinkelkompensation. Wie zu erkennen ist, ist diese ähnlich der Modulspezifikation Schräglaufeinstellung aufgebaut. Da die Schwimmwinkelkompensation jederzeit abgeschaltet werden können soll, wird zu Beginn eine Abfrage durchgeführt ob der Status aktiv ist. Danach wird $\beta = \alpha - \delta_{MR}$ berechnet und

geprüft, ob der Schwimmwinkel innerhalb der Schwimmwinkeltoleranz liegt. Ist das der Fall, bleiben die Referenzradlenkwinkelmotoren aus. Liegt der berechnete Schwimmwinkel außerhalb des Toleranzbereichs wird der entsprechende Motor angesteuert, sofern weder die entsprechende Softwaregrenze noch der entsprechende Endschalter aktiv ist.

6.3.4 Drehzahlberechnung

6.3.4.1 Methodik und Vorgehen

Daten des verwendeten Drehzahlsensors:

Inkrementanzahl = 1000 Striche/Umdrehung

1 Strich = 1-mal '0' und 1-mal '1' ó eine Periode

Vorgehen und Methodik bei der Drehzahlberechnung:

Prinzipiell gibt es zwei Möglichkeiten die Drehzahl zu berechnen.

1. festes Zeitfenster wählen und die Striche zählen
2. feste Anzahl zu zählender Striche wählen und Zeit ermitteln

Jeweils wird die Anzahl der Striche durch die Zeit geteilt.

$$\varphi = \frac{1000}{360} \cdot \# \text{ Striche} \quad \dot{\varphi} = \frac{\Delta \varphi}{\Delta t} = n \quad (6.1)$$

Bei beiden Möglichkeiten gibt es den Zielkonflikt zwischen Genauigkeit der Drehzahl und Aktualität. Da die Zeit in kleineren Schritten ermittelt werden kann als die Drehzahl ist die 2. Variante zu bevorzugen.

Zum Zählen der Striche gibt es 2 Möglichkeiten.

1. Triggerfunktion, die einen Bitwechsel bzw. Spannungssprung erkennt
2. Speichern des Wertes in einer Variablen und vergleichen mit dem aktuellen Wert

In LabVIEW gibt es nur eine Triggerfunktion für analoge Signale, die für digitale Signale umprogrammiert werden müsste und langsamer wäre als die 2. Möglichkeit bei der der digitale Wert direkt verwendet werden kann.

Bestimmung der Drehzahlabtastrate:

Maximale Drehzahl:

bei minimalem Raddurchmesser: 145/65 R15 $D_{\min} = 2 \cdot 145 \cdot 0,65 + 15 \cdot 25,4 \approx 570 \text{mm}$

und maximaler Geschwindigkeit: $v_{\max} = 100 \text{km/h} \approx 28 \text{m/s}$

$$n_{\max} = \frac{v}{\pi i D_{\min}} = \frac{28}{\pi i 0,57} \approx \frac{15,6}{s} \Rightarrow \frac{16}{s} \text{ (aufgerundet)} \Rightarrow 16000 \text{ Hz} \quad (6.2)$$

Um jeden Wert zu erkennen wird nach dem Nyquist-Shannonschen Abtasttheorem mindestens 2-mal die höchste Frequenz benötigt. Das bedeutet mindestens 32000Hz für das Drehzahlmodul. Daher wird $f_n = 100\text{kHz}$, entspricht einem Timer von $10\mu\text{s}$, gewählt.

Wahl der festen Strichanzahl:

Minimale Drehzahl:

bei maximalem Raddurchmesser: $275/35 \text{ R20} \quad D_{\max} = 2 \cdot 275 \cdot 0,35 + 20 \cdot 25,4 \approx 700\text{mm}$

und minimaler Geschwindigkeit: $v_{\min} = 10\text{km/h} \approx 2,8\text{m/s}$

$$n_{\min} = \frac{v}{\pi i D_{\max}} = \frac{2,8}{\pi i 0,7} \approx \frac{1,3}{s} \Rightarrow \frac{1}{s} \text{ (abgerundet)} \Rightarrow f_{\min} = n_{\min} \cdot \text{Strichanzahl} = 1000 \frac{\text{Striche}}{s} \quad (6.3)$$

Das heißt, dass bei einer Geschwindigkeit von 10km/h, selbst wenn nach jedem erkannten Strich die Drehzahl neu berechnet wird, diese nur mit maximal 1300Hz erfasst werden kann. Soll die Drehzahl mit höherer Frequenz erfasst werden, ist ein Inkrementalgeber mit feinerer Auflösung notwendig. Um die aktuellste Geschwindigkeit zu ermitteln wird als feste Strichanzahl die kleinst mögliche gewählt. Damit das Modul anpassbar implementiert wird, wird in der ersten Version eine feste Strichanzahl > 1 gewählt. Diese kann jederzeit auf 1 gesetzt werden. In der Spezifikation wurde 10 gewählt.

6.3.4.2 Modulspezifikation

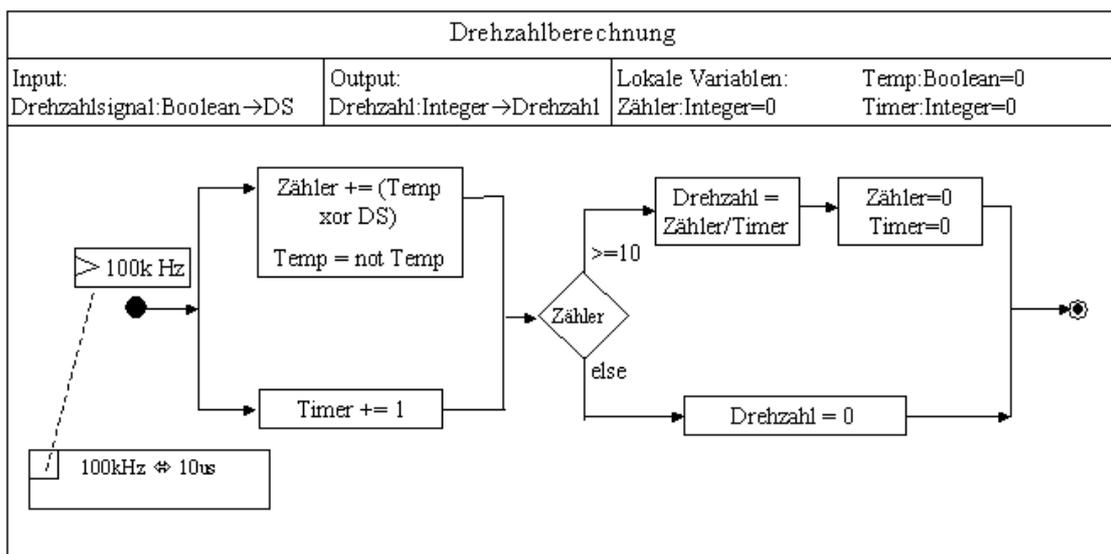


Abbildung 6.7: Drehzahlberechnung – Modulspezifikation

Die Abbildung 6.7 zeigt die erstellte Drehzahlmodulspezifikation. Hierbei wurde wie oben ermittelt die feste Abtastrate von 100kHz gewählt, um jeden Bitwechsel zu erkennen. Die Erkennung des Bitwechsels geschieht über eine temporäre Variable die den Wert des letzten erkannten Wertes speichert. Mittels einer XOR-Logik kann einfach ein Bitwechsel erkannt werden, denn alter Wert XOR neuer Wert gibt „true“ wenn ein Bitwechsel stattfindet. Wird ein Bitwechsel erkannt wird die temporäre Variable aktualisiert und der Bitwechsellähler um 1 erhöht. Der Timer der für die Berechnung der Zeit nötig ist wird in jedem Schleifendurchlauf um 1 erhöht. Ist der Zähler größer gleich der festen Strichanzahl wird die Drehzahl berechnet und sowohl der Zähler als auch der Timer zurückgesetzt. Ist der Wert des Zählers kleiner der festen Strichanzahl wird der vorhergehende Drehzahlwert beibehalten. Die Drehzahl wird in Umdrehungen pro ms ermittelt und später auf dem Hostsystem in Umdrehung pro Sekunde umgerechnet, da auf dem FPGA nur mit Ganzzahlen gerechnet werden kann.

6.3.5 Datenübertragung (HPL und NPL)

6.3.5.1 LPL-HPL-Methodik und Vorgehen:

Da die Software sowohl zeitkritische als auch weniger zeitkritische Daten verarbeitet und die Performance der Messdatenaufzeichnung das wichtigste Kriterium ist, werden die anfallenden Daten hinsichtlich ihrer Zeitkritik klassifiziert. Dazu wurden die zeitkritischen Messdaten in eine Klasse eingeteilt und weniger zeitkritische Messdaten in eine weitere Klasse. Wird im Weiteren von Messdaten gesprochen sind die zeitkritischen Messdaten gemeint. Eine Einteilung in mehr als 2 Klassen wird nicht vorgenommen, da es zeitkritische Messdaten, die alle zur gleichen Zeit erfasst werden sollen und zeitunkritische Messdaten gibt, aber keine Messdaten die klar einer Stufe dazwischen zugeordnet werden können. Außerdem würde ein weiterer DMA-Kanal zusätzlichen Puffer- und Speicherplatz benötigen. Die Daten werden parallel in zwei Prozessen verarbeitet. Die zeitkritischen Messdaten werden in einer Schleife mit hoher Priorität (HPL) und die zeitunkritischen Daten in einer Schleife geringerer Priorität (LPL) verarbeitet. Die TCL wird mit einer hohen Frequenz, die NPL mit einer geringeren Frequenz getaktet, da die entsprechenden Sensoren träge sind. Beide Prozesse erhalten separate DMA-Kanäle, damit es zu keiner Behinderung kommt. Die alternative Möglichkeit alle Daten über einen DMA Kanal weiterzuleiten ist ungünstiger, da zwischen den zeitkritischen Daten Zeitlücken für die zeitunkritischen Daten reserviert werden und der entstehende rechenzeitintensive Overhead für die Datenorganisation und –disorganisation die Datenübertragung bremsen würde.

6.3.5.2 Low Priority Loop (LPL) - Modulspezifikation

Die nachfolgende Abbildung zeigt die Modulspezifikation des Low Priority Loop Moduls. Da dieses Modul zwei verschiedene Eingangskanäle hat und die bisher verwendete Darstellung beide gleich darstellen würde, wurden die Eingangsgrößen mittels eines Verweises direkt an die Verbindung gebunden von der sie kommen. Die in dieser Modulspezifikation angegebene Abtastrate ist nur eine erste Annahme und wird so implementiert, dass sie vom Hostsystem zu Beginn der Übertragung übergeben wird. Die vom Realtime System zum FPGA zu übergebenden Variablen werden mittels Interrupt übergeben sobald sich diese auf dem RTS ändern. Diese werden bis auf die Variable „SWKaktiv“ direkt mittels einem Schreibzugriff auf FPGA I/O auf die entsprechenden Ausgangskanäle gelegt. Danach wird mittels Read FPGA I/O alle Eingangsgrößen die der geringen Priorität zugewiesen wurden von den Eingangskanälen gelesen. Diese werden zusammen mit der intern ermittelten CRIO-Temperatur über einen DMA FIFO Stream an das Realtime System übergeben. Hierbei sei darauf hingewiesen, dass aufgrund der Aufteilung der Messgrößen auch 2 verschiedene DMA-Kanäle verwendet werden. Der DMA-Kanal der geringen Priorität hat hierbei einen deutlich kleineren Puffer als der der hohen Priorität. Die eingestellte Größe kann im entsprechenden DMA-Kanal im Projektextplorer eingestellt und verändert werden.

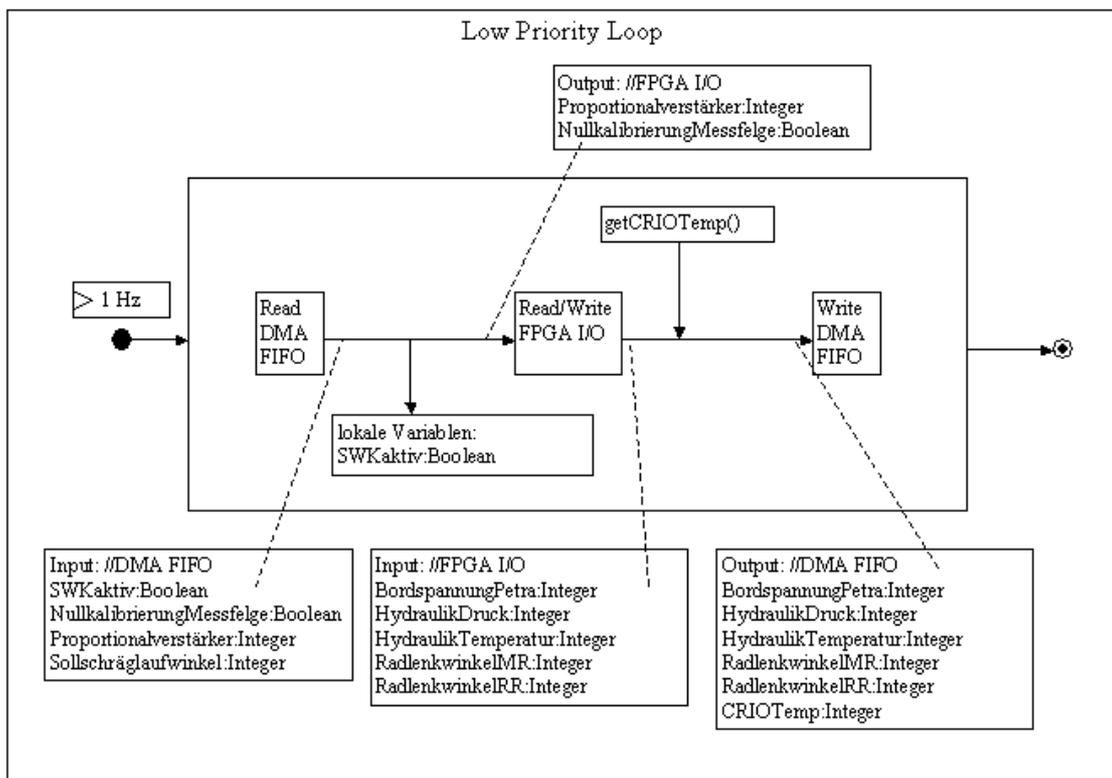


Abbildung 6.8: Low Priority Loop – Modulspezifikation

6.3.5.3 High Priority Loop (HPL) - Modulspezifikation

Wie bereits in dem Abschnitt Methodik und Vorgehen 6.3.5.1 erwähnt, werden die Messgrößen aufgeteilt. Die in Abbildung 6.9 dargestellte High Priority Loop Modulspezifikation, liest wie die LPL die Messgrößen mittels FPGA I/O von den Eingangskanälen, allerdings mit viel höherer Frequenz. Diese Frequenz wird genauso wie die der LPL vom RTS zu Beginn der Übertragung übermittelt.

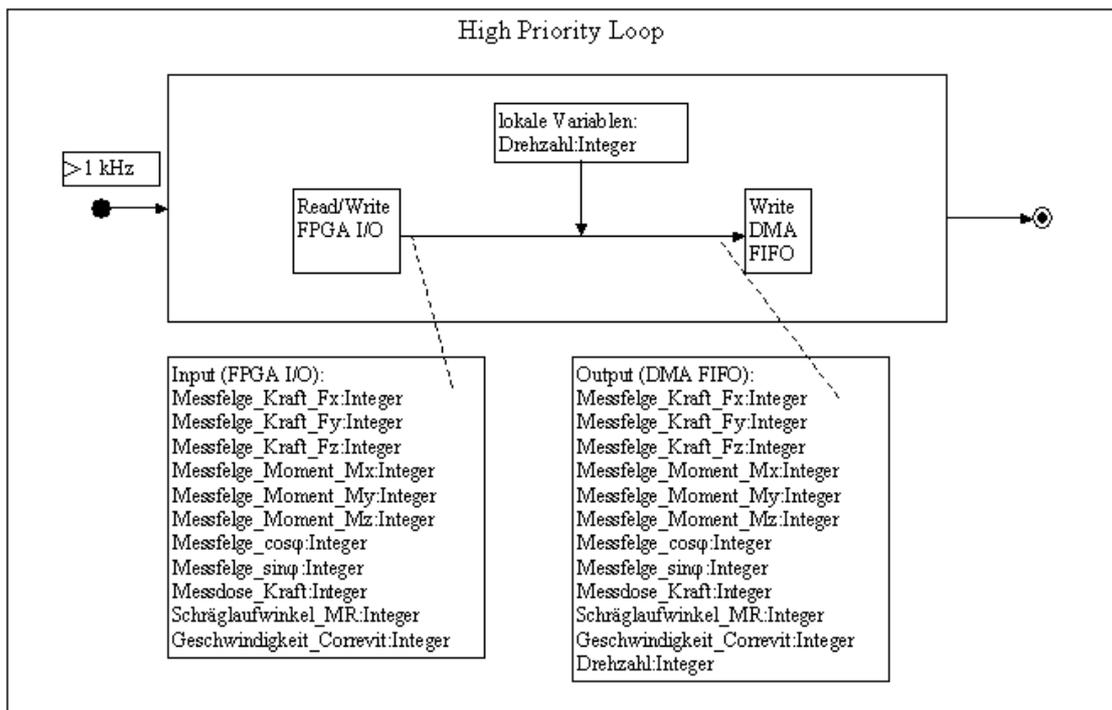


Abbildung 6.9: High Priority Loop – Modulspezifikation

6.4 Modulspezifikationen der Broker – Module

6.4.1 Methodik und Vorgehen

Da auf der FPGA-Ebene die Eingangsgrößen in Messgrößen und Kontrollgrößen aufgeteilt wurden und mittels separater DMA Kanäle an das Realtimesystem übertragen werden, werden diese beiden Kanäle in zwei verschiedenen Threads abgearbeitet. Würde das nicht geschehen ist der Vorteil der Trennung der Eingangsgrößen und verarbeiten dieser in parallelen Prozessen auf dem FPGA nicht gegeben, da sich die Abarbeitungsfrequenz der DMA-Kanäle innerhalb eines Threads an der langsamsten orientiert. Deswegen werden entsprechend der Trennung auf dem FPGA auch auf dem Realtimesystem zwei parallele Schleifen verwendet

die jeweils in einem eigenen Thread laufen und die Messgrößen weiterleiten. Des Weiteren wurde die Software des Brokers in zwei Module aufgeteilt. In Normal Priority Loop (NPL), welche die Netzwerkschnittstelle zum Host aufbaut und verwaltet und in die Time Critical Loop (TCL), welche die Mess- und die Kontrollgrößen parallel verarbeitet. Da eine TCP/IP Verbindung, welche, wie bereits in Kapitel 5.1.4.3 beschrieben, verwendet wird, eine Client/Server-Architektur ist, wird ein Netzwerkendpunkt zum Server und der andere zum Client. In diesem Projekt ist es von Vorteil wenn der Broker als Server agiert und die Verbindung aufbaut, während der Host als Client sich mit dem Broker verbindet, da verschiedene Laptops als Hostsystem verwendet werden können sollen.

6.4.2 Normal Priority Loop (NPL) – Modulspezifikation

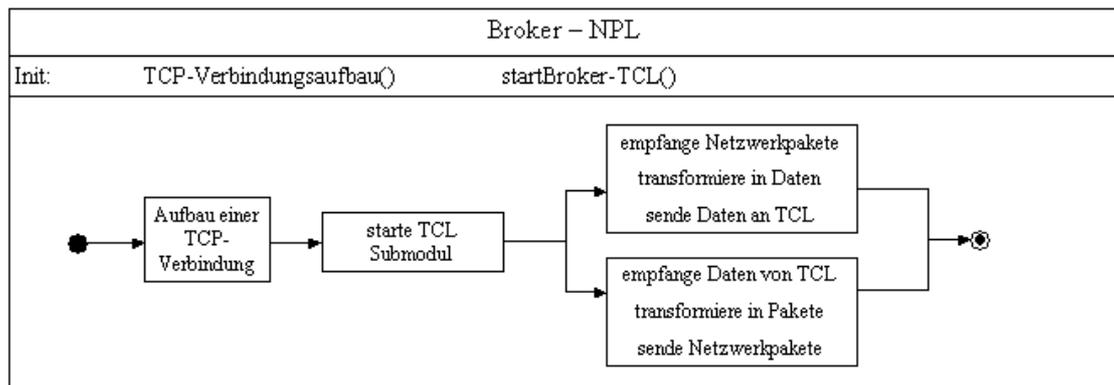


Abbildung 6.10: Normal Priority Loop – Modulspezifikation

Die obige Abbildung zeigt die Modulspezifikation des NPL-Moduls. Zunächst wird die TCP-Verbindung aufgebaut welche die IP-Schicht nutzt. Hierzu startet das NPL-Modul einen TCP-Server an einem bestimmten Port, der auf den eingehenden TCP-Verbindungsaufbau wartet. Mittels dem 3-Wege-TCP-Handshake, das wie folgt abläuft, wird die Verbindung hergestellt. Der Host, der als Client agiert, sendet eine TCP-Anfrage an den Server wofür er dessen IP und Port benötigt. Da der Port frei ist sendet der Server eine Bestätigung an den Client, welcher wiederum eine Bestätigung an den Server schickt und somit die Verbindung aufbaut. Nachdem die Verbindung mit dem Hostsystem, welches auf dem Laptop läuft, aufgebaut ist wird das Time Critical Loop Modul aufgerufen. Nach einmaligem Durchlauf dieses TCL-Moduls sind alle für die Kommunikation notwendigen Kontrollstrukturen aufgebaut. Der Datenaustausch zwischen den Modulen NPL und TCL erfolgt über Echtzeit-FIFO-Elemente (RT-FIFOs). Die über die RT-FIFOs erhaltenen Daten werden entsprechend der TCP-Konventionen in TCP-Pakete gefasst und über das Netzwerk geschickt. Die von der Netzwerkschnittstelle empfangenen TCP-Pakete werden ausgepackt und in die originale Form der

Daten zurücktransformiert. Diese Daten werden dann über die RT-FIFOs an das TCL- Modul weitergeleitet.

6.4.3 Time Critical Loop (TCL) – Modulspezifikation

Wie in dem folgenden Aktivitätsdiagramm zu sehen ist wird zunächst der FPGA des CRIO-Systems referenziert. Diese Referenzierung ist nötig um mit dem FPGA Daten austauschen zu können. Die Verarbeitung der Daten erfolgt in zwei Pfaden. Der obere Pfad liest die Daten mittels DMA und schreibt diese in die RT-FIFOs, aus welchen sie vom NPL-Modul gelesen werden. Der untere Pfad liest Daten aus den RT-FIFOs und sendet diese mittels Interrupt an den FPGA, sofern die Daten sich geändert haben. Der obere Pfad wird in zwei parallelen Schleifen mit separaten Threads implementiert, um die zwei DMA-Kanäle getrennt mit unterschiedlicher Häufigkeit abarbeiten zu können. Eine Umrechnung der Größen von Ganzzahlen nach Fließkommazahlen wird auf dieser Ebene vermieden um die Berechnungen auf dem RT-System zu Lasten des Hostsystems zu minimieren. Wie bereits erwähnt werden Daten, die vom Hostsystem kommen, nur dann übertragen, wenn sie sich geändert haben, um die Interruptkommunikation zu minimieren.

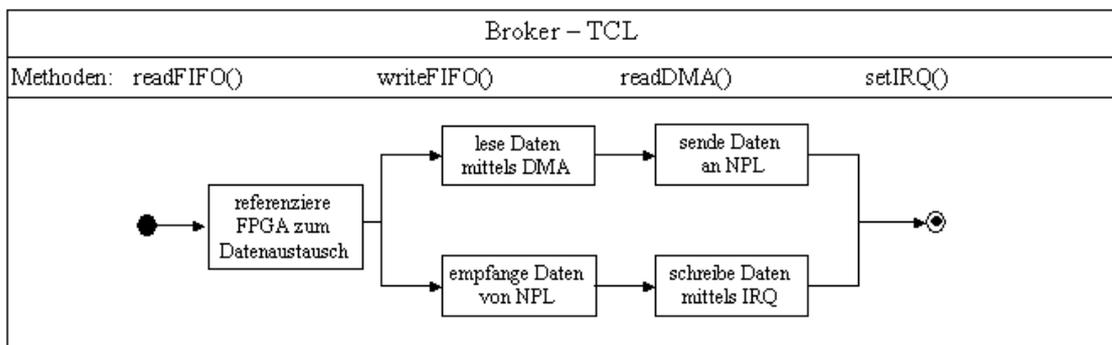


Abbildung 6.11: Time Critical Loop – Modulspezifikation

6.5 Modulspezifikationen der Host – Module

6.5.1 Hauptprogramm – Modulspezifikation

Die nachfolgende Abbildung zeigt den Aufbau des Hauptprogramms. Zunächst wird die GUI angezeigt und der Benutzer kann der Reihe nach die verschiedenen Untermenüs auswählen. Wird ein anderes als das angezeigte Menü ausgewählt, wird das neue Menü geöffnet und statt des alten angezeigt. Zu jeder Zeit während die Software läuft können durch betätigen des Be-

enden Buttons alle laufenden Prozesse abgebrochen und das Hauptprogramm beendet werden. Mit einem weiteren Button kann die Schwimmwinkelkompensation unterbrochen werden.

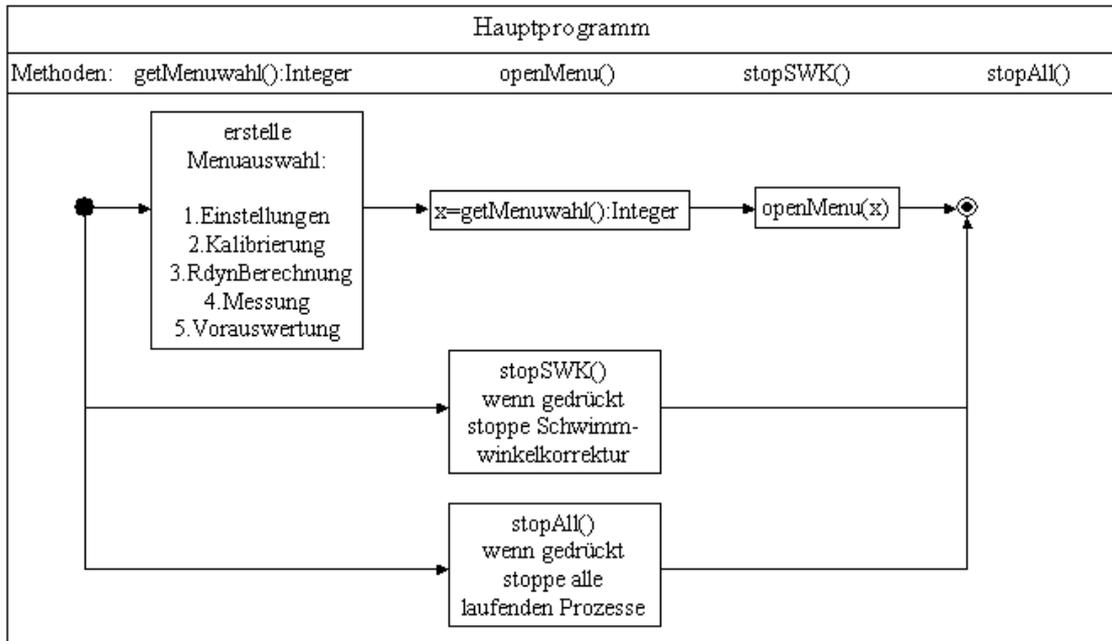


Abbildung 6.12: Hauptprogramm – Modulspezifikation

6.5.2 Einstellungen – Modulspezifikation

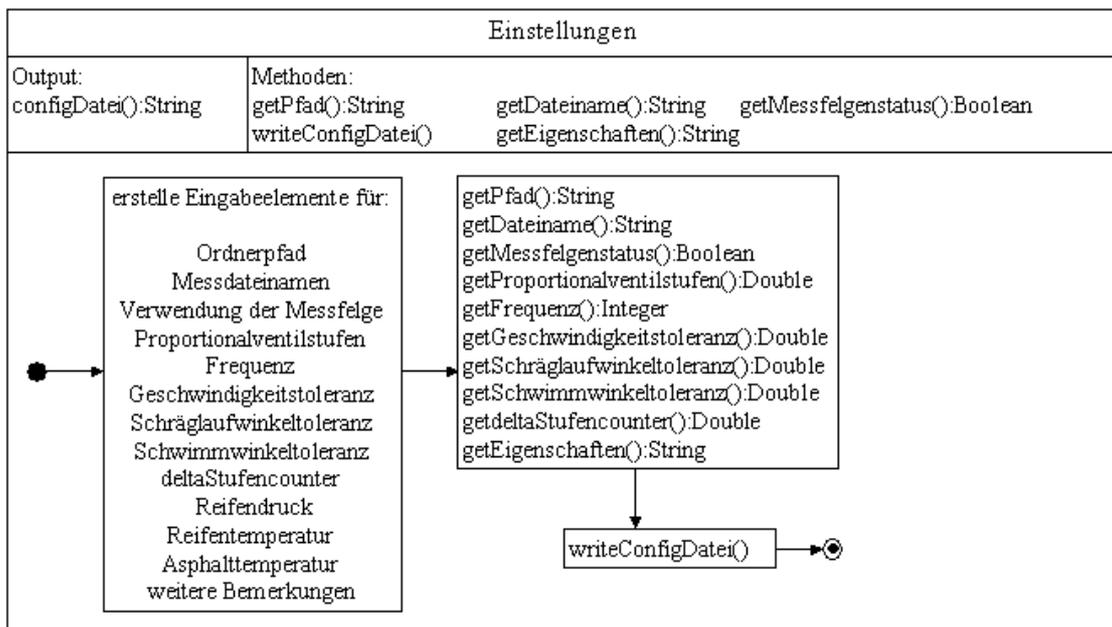


Abbildung 6.13: Einstellungen – Modulspezifikation

Die Modulspezifikation des Moduls Einstellungen sieht vor, dass für alle im vorhergehenden Aktivitätsdiagramm aufgelisteten Parametern ein Eingabeelement angelegt wird. Die eingegeben Parameter werden auf Gültigkeit überprüft und in entsprechenden internen Variablen gespeichert. Ist der eingegebene Wert außerhalb des vorgegebenen Bereichs wird der nächst gültige Wert gewählt. Ist der eingegebene Wert ungültig, wird der zuletzt gültige Wert verwendet. Sind zum Beispiel für die Frequenz Ganzzahlen im Eingabebereich zwischen 500Hz und 5000Hz zugelassen und es wird eine Frequenz von 5Hz eingegeben, wird automatisch 500Hz als kleinster gültiger Wert übernommen, wird danach „hundert“ als Frequenz eingegeben, ist dies keine gültige Ganzzahl und wird somit auf den zuletzt gültigen Wert zurückgesetzt. Damit nach der Messungsdurchführung die eingestellten Parameter dokumentiert werden können, werden diese als Text in eine Konfigurationsdatei geschrieben, die in dem aktuellen Verzeichnis, welches als Ordnerpfad ausgewählt wurde, abgelegt wird.

6.5.3 Kalibrierung – Modulspezifikation

In dem Modul Kalibrierung werden folgende Aufgaben erledigt: Erfassung und Kalibrierung aller Kalibrierdaten, Durchführung der Nullkalibrierung der Messfelge und Erfassung der Proportionalventilstufen. Die nachstehende Abbildung zeigt die Modulspezifikation der Kalibrierung. In den folgenden Zeilen werden die Funktionen beschrieben.

Erfassung und Kalibrierung aller Kalibrierdaten:

Hierzu werden zunächst die Eingabeelemente, für die in dem 1. Element aufgeführten Größen jeweils für den Offset und die Steigung (Faktor) erstellt. Die Messfelgengrößen F_x , F_y , F_z , M_x , M_y , M_z bilden hier eine Ausnahme, denn sie haben einen Umrechnungsfaktor aber keinen Offset. Nachdem der Benutzer diese Eingaben getätigt hat werden diese eingelesen. Eine besondere Stellung in dieser Funktion nimmt die Methode Radlenkwinkelkalibrierung ein. Diese enthält neben den normalen Kalibrierdaten, zur Umrechnung von Spannung in Winkel, weitere Informationen zur Berechnung von Winkeln auf dem FPGA in Integer. Die FPGA-Module Schräglaufeinstellung und Schwimmwinkelkompensation können nur mit Integerwerten rechnen. Deswegen werden zunächst die Kalibrierdaten des analogen Eingangsmoduls NI-9205 verwendet um damit die Umrechnungsfaktoren und Offsets von Spannung zu Integer zu berechnen. Damit und mit den eingelesenen Umrechengrößen (Faktor und Offset) der Umrechnung von Spannung in Winkel, werden in der Methode Radlenkwinkelkalibrierung die Umrechengrößen für die Größen Schräglaufwinkel (Winkel aus Correvit), Radlenkwinkel am Messrad und Radlenkwinkel am Referenzrad, zur Berechnung auf dem FPGA, berechnet. Des Weiteren werden die Maximal- und Minimalwinkel, die die Radlenkwinkelverstellungen zulassen, in einen Integerwert umgerechnet. Alle ermittelten Kalibrierdaten werden in einem Kalibrierdatenarray des Typs Integer gespeichert.

Erfassung der Proportionalventilstufen:

Es werden Eingabelemente erstellt in die die einzelnen Stufen des Proportionalventils eingetragen werden können. Diese Werte vom Typ Double die Spannungen am Proportionalventil entsprechen werden erfasst und in einem Array des Typs Double gespeichert. Die Verwendung von Fließkommazahlen ist möglich da die Proportionalventilstufenerhöhung während der Messung vom Host aus gesteuert wird.

Durchführung der Nullkalibrierung der Messfelge:

Diese Funktionalität kann erst durchgeführt werden, wenn eine Verbindung mit dem CRIO besteht. Auf Grund des funktionalen Zusammenhangs, wird diese Funktionalität dennoch dem Modul Kalibrierung zugeordnet. Wird eine Aktivierung des Nullkalibrierungsbuttons entdeckt, wird die Zerokalibrierungsmethode ausgeführt. Diese sorgt dafür, dass an dem entsprechenden Ausgang, mindestens so lange eine logische ‚1‘ anliegt wie für die Nullkalibrierung der Messfelge bei langsamer Fahrt benötigt wird.

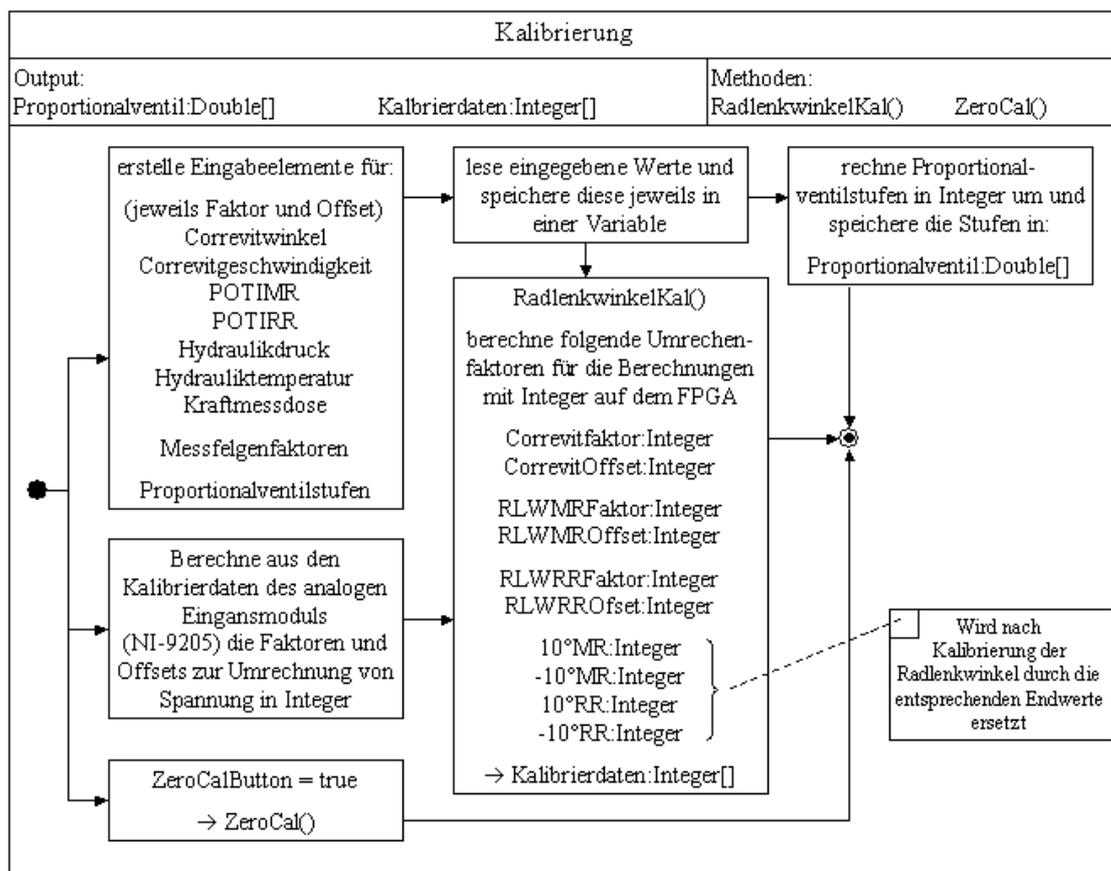


Abbildung 6.14: Kalibrierung – Modulspezifikation

6.5.4 R_{dyn}-Berechnung – Modulspezifikation

Methodik:

Der dynamische Reifenhalmmesser (R_{dyn}) kann auf 2 Arten während der Fahrt ermittelt werden.

1. R_{dyn} über Geschwindigkeit:

Unter der Annahme, dass bei niedrigen Geschwindigkeiten ein frei rollendes Rad schlupffrei ist, kann R_{dyn} durch Gleichsetzen der Rad- und der Fahrzeuggeschwindigkeit nach folgender Formel berechnet werden.

$$R_{dyn} = \frac{v_{Fzg}}{2 * \pi * n} \quad (6.4)$$

2. R_{dyn} aus der Kraftmessfelge:

Unter der gleichen Bedingung wie in 1. kann der dynamische Reifenhalmmesser aus der Kraftmessfelge ermittelt werden. Mittels dem Moment um die Radnarbe und der Bremskraft in der Aufstandsfläche des Reifens, nach folgender Formel für stationären Betrieb.

$$R_{dyn} = \frac{M_y}{F_x} \quad (6.5)$$

Während der Verifikation hat sich gezeigt, dass die Onlineberechnung von R_{dyn} nach (6.5) nicht zu gebrauchen ist. Der Vollständigkeit halber ist diese Art der Berechnung dennoch in der Modulspezifikation der R_{dyn}-Berechnung enthalten.

Berechnung der Mindestanzahl der R_{dyn}-Elemente:

Zur R_{dyn}-Berechnung wurde festgelegt, dass mindestens n = 10 Radumdrehungen zurückgelegt wurden um genügend Werte für die Mittelwertbildung zu erhalten. Daraus folgt bei einer Messgeschwindigkeit von v_{Fzg} = 2m/s mit R_{dyn} = 0,3m und einer Abtastrate von f = 500Hz mit:

$$s = 2 * \pi * n * R_{dyn} \approx 20m \Rightarrow t = \frac{s}{v_{Fzg}} \approx 10s \Rightarrow \#Werte_{min} = \frac{f}{t} = 5000 \quad (6.6)$$

Modulspezifikation:

Die nächste Abbildung zeigt die Modulspezifikation des Moduls R_{dyn}-Berechnung. Eingangsvariablen sind das Messdatenarray welches alle Messdaten hoher Priorität enthält, sowie die bereits im Modul Einstellung erfassten Eingaben Geschwindigkeitstoleranz, Schräglauftoleranz und Messfelgenstatus. Ausgabe ist der berechnete R_{dyn}. Als erstes wird die Messdatenübermittlung gestartet. Danach wird ein Eingabeelement erstellt in das der Benutzer die Messgeschwindigkeit eingibt. Diese Messgeschwindigkeit, bei der R_{dyn} bestimmt wird, wird eingelesen. Danach werden die benötigten Messgrößen aus dem Messdatenarray ausgepackt

und entsprechend der Kalibrierdaten erst in eine Spannung mit der Einheit Volt und dann in die entsprechende Einheit umgerechnet. In der folgenden Abfrage wird geprüft ob der Betrag der Differenz zwischen Fahrzeuggeschwindigkeit und Messgeschwindigkeit innerhalb des vorgegebenen Geschwindigkeitstoleranzbereichs liegt. Zusätzlich wird geprüft ob der Betrag des Schräglaufwinkels innerhalb des eingegebenen Schräglaufwinkeltoleranzbereichs liegt. Ist eine der Bedingungen nicht erfüllt werden die Daten nicht verwendet und das nächste Array wird entpackt. Sind beide Bedingungen erfüllt wird abhängig von der Bedingung ob die Messfelge verwendet wird R_{dyn} zusätzlich über die Messfelge berechnet. Im nächsten Schritt wird der dynamische Reifenhalbmesser berechnet nach Variante 1 bzw. 1. + 2. wenn die Messfelge montiert ist. Dieser wird dem R_{dyn} -Array angehängt. Danach wird geprüft ob das R_{dyn} -Array bereits 5000 Elemente, Ergebnis aus (6.6), enthält. Ist dies der Fall wird der Mittelwert der im Array enthaltenen dynamischen Reifenhalbmessern berechnet. Ansonsten werden weitere R_{dyn} s berechnet. Zum Schluss wird der mittels Mittelwert berechnete R_{dyn} angezeigt und danach die R_{dyn} -Messung gestoppt.

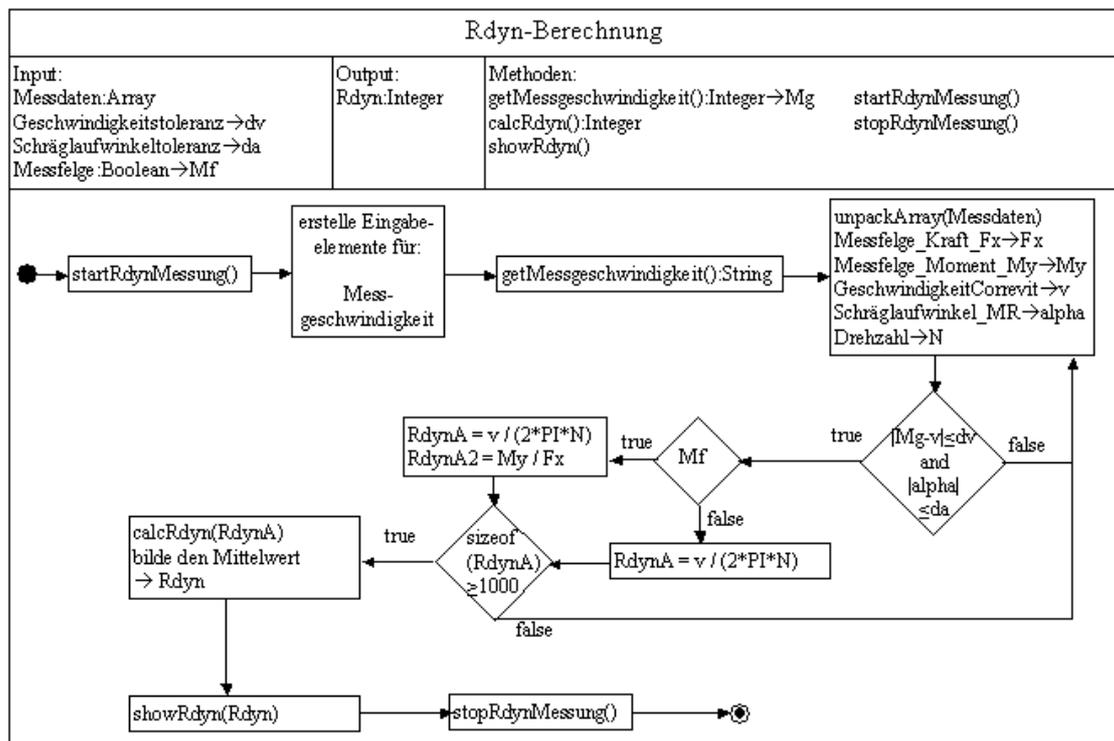


Abbildung 6.15: R_{dyn} -Berechnung – Modulspezifikation

6.5.5 Messung – Modulspezifikation

Die nächste Abbildung zeigt die Modulspezifikation des Moduls Messung. Eingangsvariablen sind das Messdatenarray welches alle Messdaten hoher Priorität enthält, sowie die bereits im

Modul Einstellung erfassten Eingaben Geschwindigkeitstoleranz und Schwimmwinkeltoleranz. Ausgabe ist die Messdatendatei die alle Messgrößen Zeilenweise in Textform enthält.

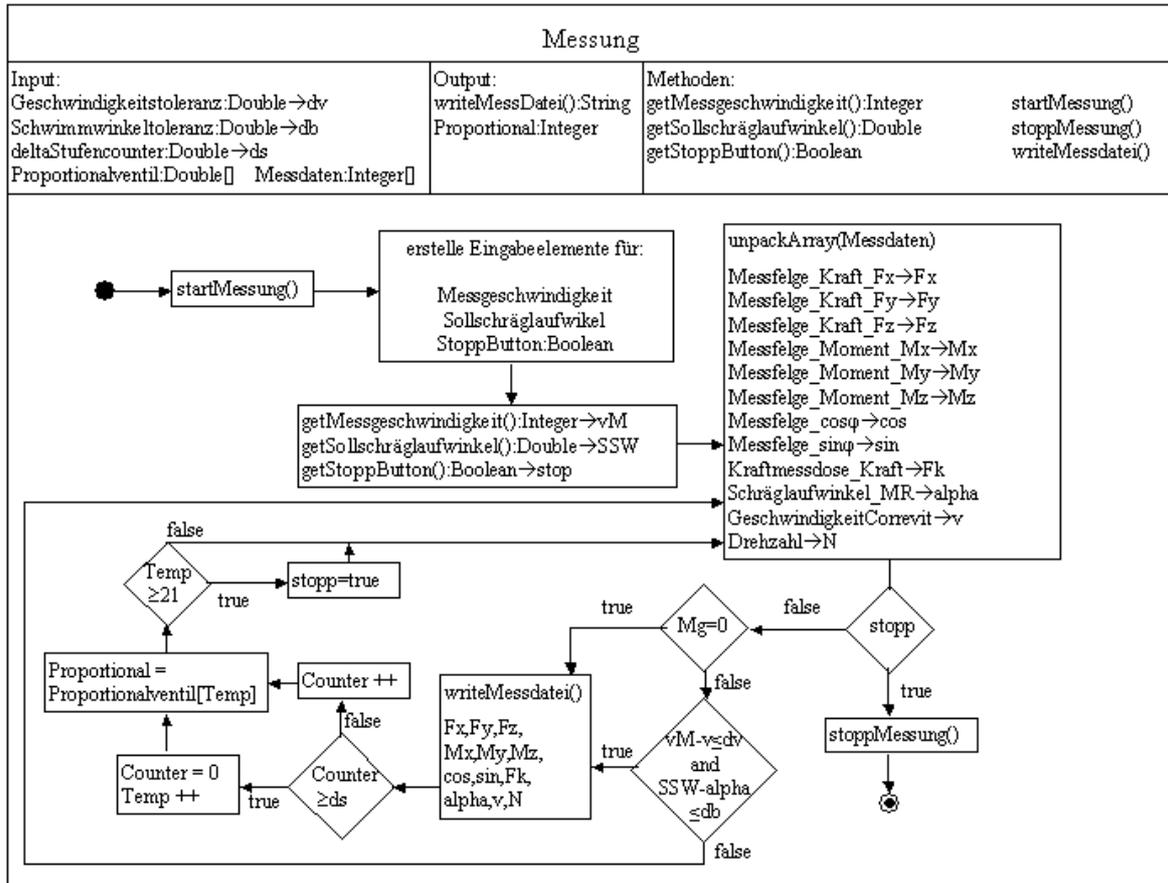


Abbildung 6.16: Messung – Modulspezifikation

Als Erstes wird die Messdatenübermittlung gestartet. Danach werden die Eingabelemente erstellt, in die der Benutzer die Eingaben Messgeschwindigkeit und Sollschräglaufwinkel tätigt und es wird ein Stoppbutton erstellt zum stoppen der Messungsdurchführung. Die Benutzereingaben werden eingelesen. Danach werden die benötigten Messgrößen aus dem Messdatenarray ausgepackt und entsprechend der Kalibrierdaten erst in eine Spannung mit der Einheit Volt und dann in die entsprechende Einheit umgerechnet. In der folgenden Abfrage wird geprüft ob die stopp-Variable „true“ ist, z.B. der Stoppbutton betätigt wurde. Ist dies der Fall wird die Messung gestoppt. Danach wird geprüft ob dynamische Messung ausgewählt wurde ($Mg = 0$), ist dies der Fall werden die Messdaten direkt aufgezeichnet. Ansonsten wird geprüft ob der Betrag der Differenz zwischen Fahrzeuggeschwindigkeit und Messgeschwindigkeit innerhalb des vorgegebenen Geschwindigkeitstoleranzbereichs liegt. Zusätzlich wird geprüft ob der Betrag der Differenz des Sollschräglaufwinkels und des Schräglaufwinkels, welcher dem Schwimmwinkel entspricht, innerhalb des eingegebenen Schwimmwinkeltoleranzbereichs liegt. Ist eine der Bedingungen nicht erfüllt werden die Daten nicht verwendet und das

nächste Array wird entpackt. Sind die Bedingungen erfüllt wird geprüft ob der Counter bereits dem „deltaStufencounter“ entspricht, wenn dies der Fall ist wird der Counter zurückgesetzt und die Variable „Temp“ erhöht. Ansonsten wird der Counter erhöht. Danach wird dem Proportionalventil die aktuelle Proportionalventilstufe übergeben, welche zuvor in Integer umgerechnet wird. Zum Schluss wird geprüft ob bereits alle Stufen durchlaufen wurden, hier $Temp \geq 21$, d.h. nach der 20. Stufe wird die Variable „stopp“ auf „true“ gesetzt und somit die Messung beendet.

6.5.6 Vorauswertung – Modulspezifikation

Die Abbildung 6.17 zeigt die Modulspezifikation der Vorauswertung, welche nur zur Vollständigkeit hier gezeigt wird. Da das Modul eine Wunschfunktionalität abbildet, wurde das Modul nur testweise implementiert, aber nicht in die Endversion der Software aufgenommen, da es keine Vorteile im Vergleich zur Offlineauswertung bietet.

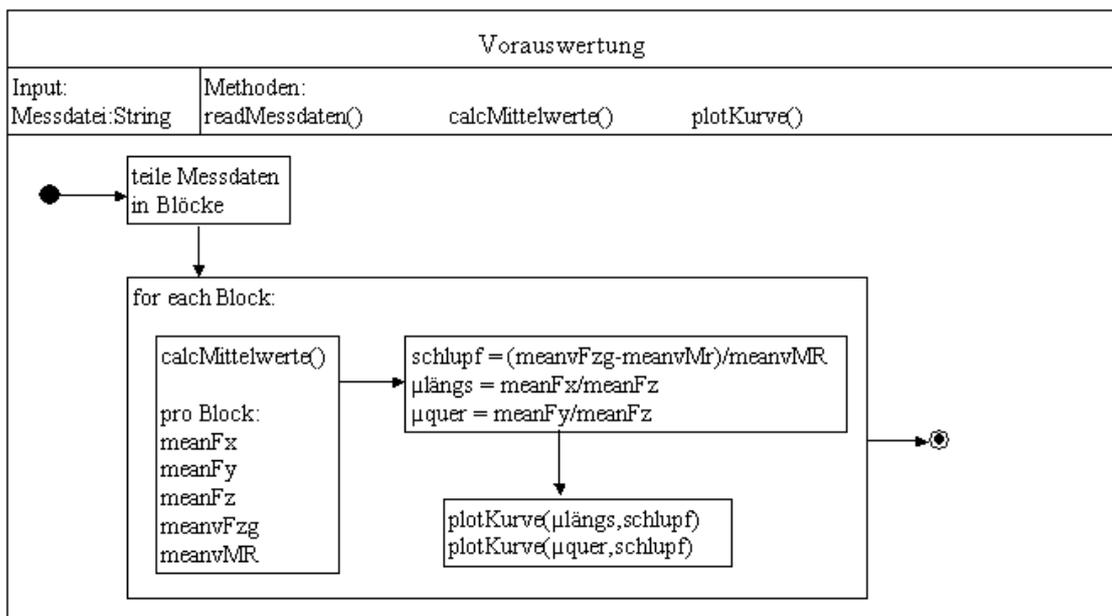


Abbildung 6.17: Vorauswertung – Modulspezifikation

Zur Durchführung der Vorauswertung gibt es 2 Möglichkeiten:

1. Vorauswertung nach Beendigung der Messung:

Hat den Vorteil, dass die Messungsaufzeichnung nicht beeinträchtigt wird. Der Nachteil ist, dass die komplette Messdatei erneut gelesen wird und somit keinen Zeitvorteil im Vergleich zur normalen Auswertung mit Matlab, welche ein genauere Beurteilung der Messung erlaubt, bietet.

2. Vorauswertung während der Messung:

Ein Zeitgewinn ist zu erzielen, wenn die Vorauswertungsrechnung online während der Messung mitläuft und nach Beendigung dieser geplottet wird. Diese Möglichkeit, die testweise implementiert wurde, wird aufgrund der Rechenzeit- und Speicherkapazität auf dem Hostsystem im Hinblick auf maximale Performance nicht verwendet.

6.5.7 Statusanzeigen – Modulspezifikation

Die Modulspezifikation der Statusanzeigen ist der nachstehenden Abbildung zu entnehmen. Die Eingangsgrößen des Moduls sind alles Statusgrößen, bzw. deren verrechnete Größen die angezeigt werden. Die Methode „showGeschwindigkeit“ erstellt ein besonders großes Anzeigeelement, damit der Fahrer die Längsgeschwindigkeit gut erkennen kann. Die Methode „calcSchwimmwinkel“ berechnet Schwimmwinkel durch die Differenz von Schräglaufwinkel und Radlenkwinkel am Messrad. In der Methode „showStatus“ werden alle Größen angezeigt die für die Überwachung und die Messdurchführung notwendig sind. Die Größen die angezeigt werden sind in der Methode, die in der nachstehenden Abbildung zu sehen ist aufgeführt.

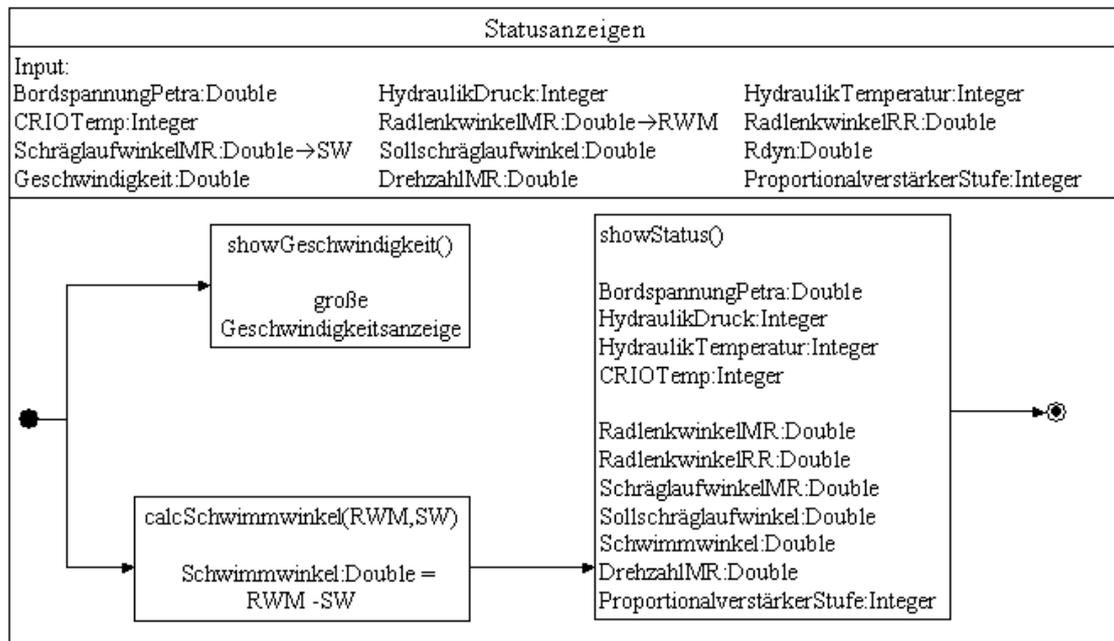


Abbildung 6.18: Statusanzeigen – Modulspezifikation

6.5.8 Warnung – Modulspezifikation

Das Modul Warnung warnt den Benutzer bei Überschreiten der Grenzwerte der kritischen Kontrollgrößen die zur Überwachung des Status des Messanhängers ermittelt werden. Die einzelnen Grenzwerte können dem Anforderungsdefinitionsdokument im Anhang entnommen werden. Das Warnen erfolgt, wie in der folgenden Modulspezifikation zu entnehmen ist in zwei Stufen. Überschreitet eine Größe die 1. Grenze, wird die Methode „showWarnung(x,'gelb')“ aufgerufen, hierbei steht x für die entsprechende Größe die ihre 1. Grenze überschritten hat. Die Methode „showWarnung“ mit dem Übergabeparameter ‚gelb‘, gibt einen einmaligen Warnton aus und zeigt das Ausgabeelement der Größe in Rot. Überschreitet eine Größe die 2. Grenze, bzw. wird bei den digitalen Größen ein „true“ detektiert, wird die Methode „showWarnung(x,'rot')“ aufgerufen. Die so aufgerufene Methode gibt einen sich dauernd wiederholenden Warnton aus und lässt das entsprechende Ausgabeelement rot blinken. Wird bei der Bordspannung die Grenze von 10,6V oder bei der CRIOTemperatur die Grenze von 67° überschritten, ist ein sofortiges einschreiten nötig und die Software sofort herunterzufahren. Hierzu wird die Methode „shutdown“ aufgerufen, die die aktuelle Messung sofort beendet und das CRIO kontrolliert herunterfährt.

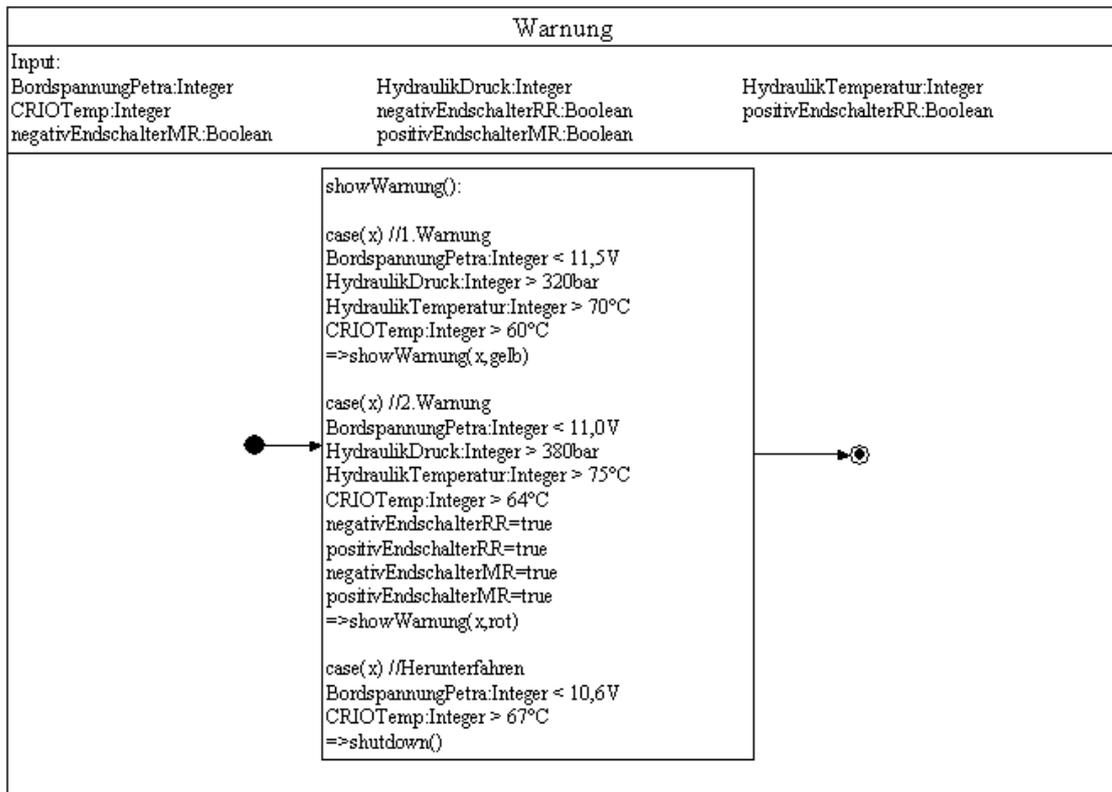


Abbildung 6.19: Warnung – Modulspezifikation

7 Programmierung

Programmiert wird in der LabVIEW-Entwicklungsumgebung, das Projekt wird mit Hilfe des LabVIEW-Projektexplorers verwaltet.

Die Aufteilung der Programmdateien ist wie folgt:

- FPGA.vi enthält alle FPGA-Module und ist im Projekt unter FPGA zu finden
- Broker-TCL.vi enthält das Broker-Modul TCL und befindet sich im Projekt unter CRIO
- Broker-NPL.vi enthält das Broker-Modul NPL und befindet sich im Projekt unter CRIO
- Host.vi enthält alle Host-Module und befindet sich im Projekt unter „My Computer“

7.1 FPGA-Programmierung

Die Programmierung auf dem FPGA ist nur mit einfachen Elementen möglich, es stehen nicht alle LabVIEWfunktionen zur Verfügung, sondern nur die, die auf dem FPGA umgesetzt werden können.

Die Interruptkommunikation ist auf dem FPGA sehr einfach zu implementieren. Um per Interrupt eine Variable auf dem FPGA zu empfangen, genügt es ein Eingabeelement zu erstellen. Umgekehrt genügt es ein Ausgabeelement zu erstellen, um eine Variable per Interrupt zu senden. Die eigentliche Interruptkommunikation erstellt LabVIEW selbst, sobald auf dem Realsystem ein Interruptzugriffselement, auf eines dieser erstellten Ein- und Ausgabeelemente, erstellt wird.

Bis auf das Konfigurationsmodul, welches vor den anderen abgearbeitet wird, laufen alle Schleifen auf dem FPGA in der Hardware echt parallel. Jede WHILE-Schleife erhält einen Timer, der festlegt wann die Schleife das nächste Mal gestartet wird.

Da beim Kompilieren mit aktiver Fehlerbehandlung auf dem FPGA ein Fehler auftritt und dieser, vom NI Support bestätigt, nur durch Weglassen der Fehlerbehandlung auf dem FPGA zu lösen ist, wird auf jegliche Fehlerbehandlung auf dem FPGA verzichtet.

7.1.1 Konfigurationsmodul

Das Konfigurationsmodul auf dem FPGA hat folgende 2 Hauptaufgaben. Die 1. Aufgabe ist, die einzelnen Eingänge des Analogen Eingangsmoduls (NI-9205) so zu konfigurieren wie sie

verdrahtet wurden, diese Implementierung ist in der nächsten Abbildung auf der linken Seite des vertikalen Balkens zu sehen. Zur Konfiguration wird jedem Eingang im 1. Block (Set Voltage Range) einer der möglichen Spannungsbereiche ($\pm 10V$, $\pm 5V$, $\pm 1V$, $\pm 0,2V$) und im 2. Block das verwendete Messverfahren (RSE, NRSE, DIFF) zugewiesen. Die zu jedem Kanal verwendeten Spannungsbereiche und Anschlussvarianten, können der sich im Anhang B befindenden Tabelle mit einer Übersicht der Ein- und Ausgänge, entnommen werden. Um nicht in jedem Eingangskanal einzeln diese Konfigurationsparameter zu setzen, werden die Parameter mit 4 FOR-Schleifen gesetzt. Erläuterung des Ablaufs anhand der 2. Schleife, siehe Abbildung 7.1. Die Schleifendurchlaufanzahl im Beispiel 10 wird mit dem N links oben verbunden. D.h. die Schleife zählt von 0 bis 9 und der aktuelle Schleifenzähler kann dem Element i entnommen werden. Zu diesem wird die Konstante 8 addiert und mit dieser Summe die Eingänge adressiert, d.h. die AI-Kanäle 8 bis 17. Jedem dieser Kanäle wird im 1. Block $\pm 5V$ und im 2. Block NRSE zugewiesen.

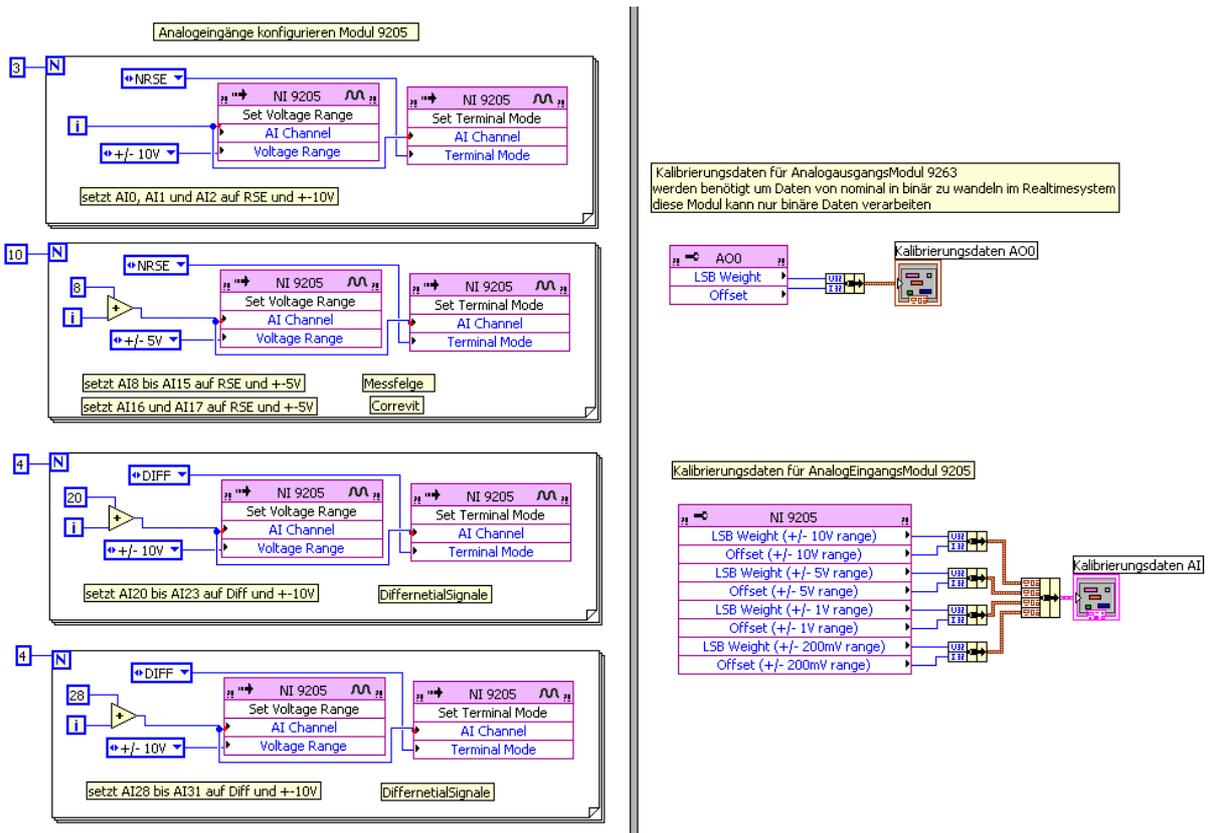


Abbildung 7.1: Konfigurationsmodul

Die 2. Hauptaufgabe ist das Auslesen der Kalibrierdaten der Analogmodule und bereitstellen dieser Information für den Broker. Zur Interruptkommunikation zwischen FPGA und Broker genügt es, wenn die entsprechenden Daten auf dem FPGA in einem Ausgabeelement gespeichert werden. Das Realtimesystem kann dann mittels Interrupt auf diese Ausgabeelemente zugreifen. Im rechten, oberen Teil werden die Kalibrierdaten des Analogausgangsmoduls NI-

9263 ausgelesen und diese in einem Cluster gespeichert. Im rechten, unteren Teil werden die Kalibrierdaten des Analogeingangsmoduls NI-9205 für die verschiedenen Spannungsbereiche ausgelesen und in einer Clustervariable gespeichert.

7.1.2 Mittelwertbildung

Da auf allen analogen Eingangskanälen ein Rauschen von ca. 10mV bei inaktiver Motoransteuerung, bzw. ca. 100mV bei aktiver Motoransteuerung liegt, werden die Module Schwimmwinkelkompensation bzw. Schräglaufeinstellung, bei denen abhängig von den Eingangsgrößen geregelt wird, gestört. Daher werden die Eingangsgrößen die für die Regelungen notwendig sind einer Mittelwertbildung unterzogen. Die Mittelwertbildung wird nur für das Schräglaufeinstellungs- und das Schwimmwinkelkompensationsmodul verwendet. Die Größen werden ohne Mittelwertbildung an den Broker weitergeleitet.

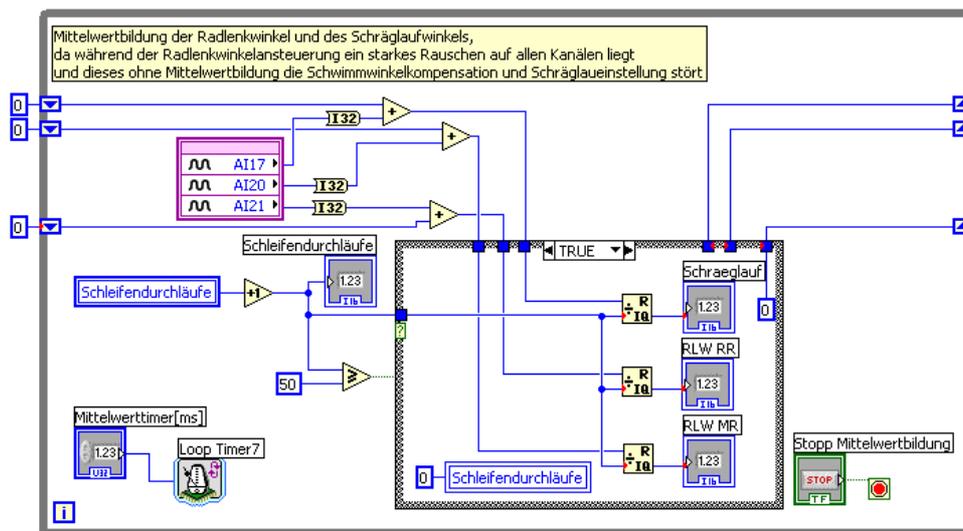


Abbildung 7.2: Mittelwertbildung

Die obige Abbildung 7.2 zeigt die Implementierung der Mittelwertbildung für die Größen Schräglaufwinkel (AI17), Radlenkwinkel am Messrad (AI20) und Radlenkwinkel am Referenzrad (AI21). Da es auf dem FPGA kein Element Mittelwertbildung gibt, wurde diese programmiert. Das Modul addiert jeweils 50 Elemente und teilt die Summe durch 50. Die Zwischenergebnisse werden dabei mittels Schieberegistern von einem Schleifendurchlauf in den nächsten übergeben.

7.1.3 Schräglaufeinstellungsmodul

Die nachfolgende Abbildung 7.3 zeigt die Implementierung des Moduls Schräglaufeinstellung. Mittels AND-Logik konnte hier komplett auf IF-Abfragen verzichtet werden. Zunächst

werden aus dem FPGA I/O Element die Variablen DIO2, entspricht negativer Endschalter am Messrad, und DIO3, entspricht positiver Endschalter am Messrad, gelesen. Die Größe Radlenkwinkel am Messrad (RLW MR) wird per Variable von der Mittelwertbildung übergeben. Die anderen Elemente werden per Interrupt vom Realsystem übergeben. Der übergebene Sollschräglaufwinkel entspricht dem einzustellenden Radlenkwinkel. Dieser und der Radlenkwinkel am Messrad werden zunächst um den Offset korrigiert und dann voneinander subtrahiert, das Ergebnis ist die Differenz zwischen Ist- und Sollradlenkwinkel. Diese wird mit dem deltaSchräglaufwinkel verglichen und somit ermittelt ob der Istradlenkwinkel außerhalb des Schräglauftoleranzbereichs liegt. Ist dies der Fall und es ist nicht der Grenzwinkel (-10°MR-RLW bzw. 10°MR-RLW) erreicht und nicht der Endschalter in dieser Richtung aktiv, kann die Messradmotorverstellung in diese Richtung aktiviert werden. Dies geschieht über das FPGA I/O Element, in welchem die Ausgänge DO0 (Motor Messrad negativ) und DO1 (Motor Messrad positiv) angeschlossen wurden. Mittels dem Eingabeelement „Schräglaufeinstellung aktiv“ lässt sich die Schräglaufeinstellung stoppen.

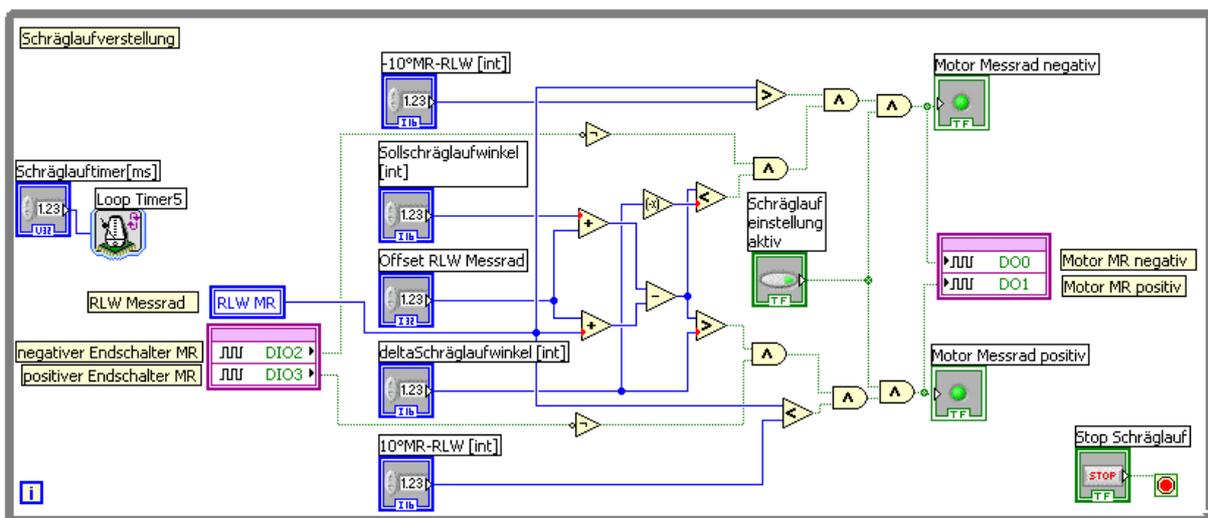


Abbildung 7.3: Schräglaufeinstellung

7.1.4 Schwimmwinkelkompensationsmodul

Die Implementierung der Schwimmwinkelkompensation ist der Schräglaufeinstellung sehr ähnlich, wie in Abbildung 7.4 zu erkennen ist. Die Größen Schräglaufwinkel, Radlenkwinkel am Messrad und Radlenkwinkel am Referenzrad werden von der Mittelwertbildung per Variablen übergeben. Nach einer Offsetkorrektur von Schräglaufwinkel und Radlenkwinkel Messrad werden diese jeweils mit dem Faktor des anderen multipliziert, um beide Größen miteinander vergleichen zu können. Die Differenz der beiden so berechneten Winkel ergibt den Schwimmwinkel, welcher laut Anforderung ausgeregelt werden soll. Diese wird mit dem deltaSchwimmwinkel verglichen und somit ermittelt ob der Schwimmwinkel außerhalb des

Schwimmwinkeltoleranzbereichs liegt. Ist dies der Fall und es ist nicht der Grenzwinkel ($-10^{\circ}RR-RLW$ bzw. $10^{\circ}RR-RLW$) erreicht und nicht der Endschalter in dieser Richtung aktiv, kann die Referenzradmotorverstellung in diese Richtung aktiviert werden. Dies geschieht über das FPGA I/O Element, in welchem die Ausgänge DO2 (Motor Referenzrad negativ) und DO3 (Motor Referenzrad positiv) angeschlossen wurden. Mittels dem Eingabeelement „Schwimmwinkelkompensation aktiv“ lässt sich die Schwimmwinkelenausregelung stoppen.

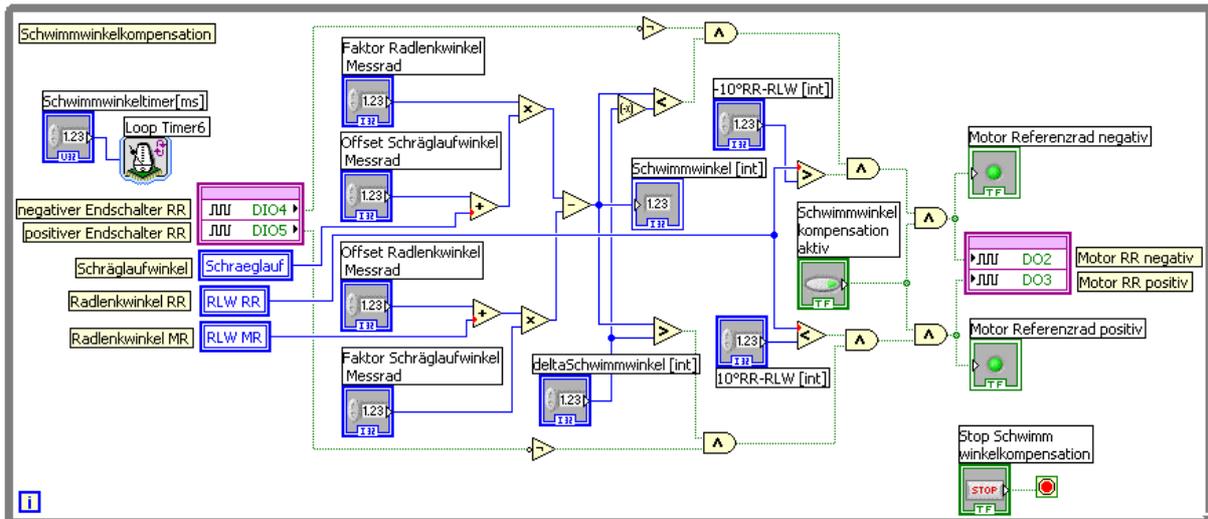


Abbildung 7.4: Schwimmwinkelkompensation

7.1.5 Drehzahlberechnungsmodul

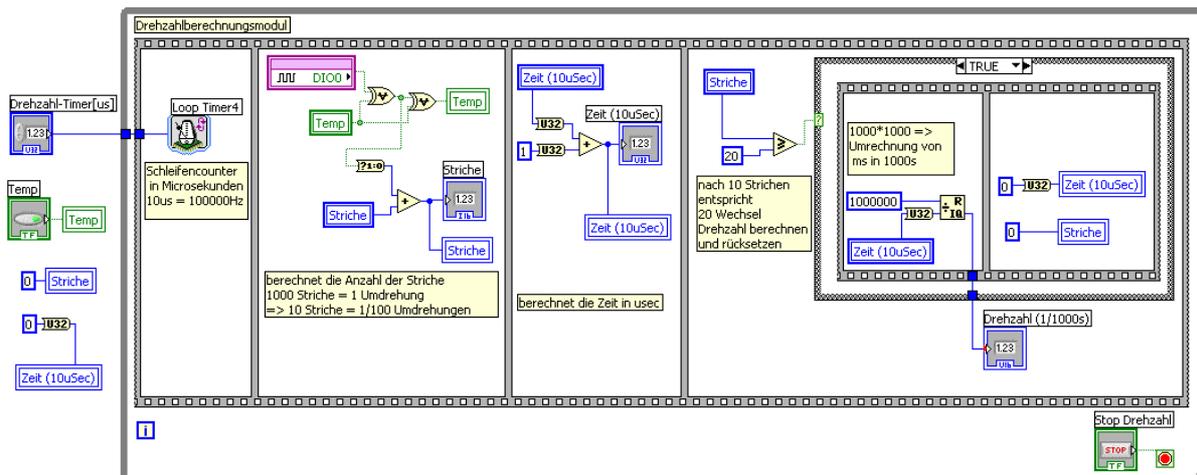


Abbildung 7.5: Drehzahlberechnungsmodul

Wie in Abbildung 7.5 zu erkennen ist wurden zunächst die Hilfsvariablen Temp, Striche und Zeit erstellt und auf 0 gesetzt. Wird diese Modulspezifikation mit der vorherigen verglichen fällt auf, dass der Schleifentimer außerhalb der Schleife gesetzt wird. Das hat den Vorteil,

dass der Wert fest ist und somit die Schleife schneller ausgeführt werden kann. Der Nachteil ist, dass der Wert zur Laufzeit nicht mehr geändert werden kann, was aber bei der Drehzahlberechnung nicht vorgesehen ist. Im inneren der Schleife ist eine sequentielle Struktur zu erkennen, d.h. die Blöcke innerhalb dieser Struktur werden hintereinander verarbeitet. Im ersten Block wird der Schleifentimer aktualisiert. Im 2. Block wird zunächst der aktuelle Wert des Inkrementalgebers gelesen und mittels einem XOR mit dem vorherigen Wert, welcher in der Variable temp gespeichert wurde verglichen. Hat sich der Wert im Vergleich zum vorherigen Wert geändert liefert die XOR-Operation eine ‚1‘ sonst eine ‚0‘. Dieser Wert XOR temp liefert wiederum den aktuellen Wert, der der temp-Variablen zugewiesen wird. Der mittels dem 1. XOR ermittelte Wert, welcher einen Bitwechsel erkennt, wird nun in der Variable Striche aufaddiert. Im 3. Block wird in der Variable Zeit in jedem Schleifendurchlauf inkrementiert. Da die Schleife alle 10µs durchlaufen wird, zählt die Variable die Zeit in 10µs. Im 4. Block wird geprüft, ob bereits 20 Bitwechsel, entsprechend 10 Strichen, entdeckt wurden. Im positiven Fall wird die neue Drehzahl berechnet und danach die Hilfsvariablen Zeit und Striche zurückgesetzt. Im negativen Fall wird geprüft ob die Zeitvariable bereits größer 32000 ist, wenn dies der Fall ist wird die Drehzahl auf 0 gesetzt, wenn nicht wird der vorherige Wert beibehalten. Die Konstante 32000, entspricht 320ms, wird benötigt um festzulegen ab welcher Zeit, bei Stillstand des Drehzahlsensors, die Drehzahl auf 0 gesetzt wird. Hierbei ist zu beachten, dass die Zeitvariable zurückgesetzt wird bevor sie überläuft.

7.1.6 LPL – Modul

Das Low Priority Loop Modul besteht aus 2 einzelnen Submodulen, die parallel zueinander ausgeführt werden. Aus Performance Gründen ist es notwendig die LPL-Down Schleife parallel zu der LPL-Up Schleife zu betreiben. Würden beide Teilmodule in einer Schleife laufen, würde die langsame Interruptkommunikation des LPL-Down-Submoduls die Ausführungsgeschwindigkeit des LPL-Up-Submoduls deutlich beschränken und somit die DMA-Kommunikation des LPL-Up-Submoduls ausbremsen.

7.1.6.1 LPL-Down-Modul

Die folgende Abbildung zeigt die Implementierung des LPL-Down-Moduls. Da die Interruptkommunikation auf der FPGA-Seite wie bereits erwähnt sehr einfach zu implementieren ist, besteht das Modul nur aus dem Schleifenzähler im vorderen Block, den Eingabeelementen des Proportionalventils und der Zerokalibrierung, welche mit den entsprechenden Ausgängen am FPGA I/O Element verbunden sind.

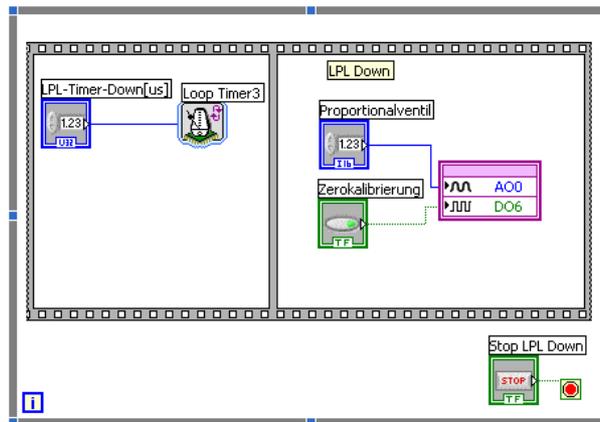


Abbildung 7.6: LPL-Down Modul

7.1.6.2 LPL-Up-Modul

Die nachfolgende Abbildung zeigt die Implementierung des Moduls LPL-Up. Die Daten werden mittels einem DMA-Kanal zum Realsystem übertragen. Der DMA-Kanal wurde zuvor im Projektextplorer angelegt. Bei der Programmierung trat das Problem auf, dass ein einmal angelegter DMA-Kanal zwar umbenannt werden konnte, aber danach nicht mehr funktionierte. Der NI-Support konnte dies in der LabVIEW Version 8.20 nicht bestätigen. Da die FPGA-Module mit der Version 8.01 erstellt und nachträglich in 8.20 konvertiert wurden, kann dieser Fehler aber nicht ausgeschlossen werden. Es wird daher empfohlen die DMA-Kanäle nicht umzubenennen.

Wichtig bei der Dimensionierung der DMA-Kanalgröße ist, dass die Größe der Summe aller DMA-Kanäle plus des verwendeten Speichers durch den FPGA-Code die RAM-Größe von 192KB nicht überschreiten darf. Ein überschreiten der RAM-Größe wird erst nach der Kompilierung des FPGA-Codes erkannt. Dies hat zur Folge, dass es nötig wird den FPGA-Code erneut mit geringerer DMA-Kanalgröße zu kompilieren.

Mittels DMA können in dem verwendeten CRIO nur Daten in Integer übertragen werden. Daher werden auch boolesche Variablen in Integer umgewandelt. Dazu gibt es 2 Möglichkeiten. Jede boolesche Variable wird direkt in ein Integer umgewandelt, d.h. eine 1 Bit Variable benötigt dann 32 Bit. Die 2. Möglichkeit ist mehrere boolesche Variablen in einen Integer zu packen, lassen sich bis zu 32 boolesche Werte in einer Integer Variablen transportieren. Allerdings bedeutet dies einen höheren Aufwand um die booleschen Werte in richtiger Reihenfolge wieder zu entpacken. In diesem Modul wird die 2. Variante verwendet, die wie folgt implementiert ist. Die 1. boolesche Variable wird in den Integer geschrieben, dieser hat nun den Wert „0“ oder „1“. Diese Bitfolge, welche den Integer repräsentiert, wird mit Hilfe des Shiftoperators um eine Stelle nach links verschoben. Damit steht die 1. eingefügte Variable nun an vorletzter Stelle und entspricht einem Integer von 2 falls der ursprüngliche Wert „true“ war

bzw. „0“ bei „false“. Nun wird die 2. boolsche Variable addiert und ebenfalls ein Linksshift angewendet. Dies wird wiederholt bis die letzte Variable addiert wurde.

Folgendes Beispiel erklärt dieses Vorgehen:

v1 = „true“, v2 = „false“, v3 = „true“

Die Variablen v1, v2 und v3 werden in dieser Reihenfolge in den Integer gepackt. Somit ergibt sich eine Bitfolge (führende Nullen weggelassen) des Integers von „101“ was dem Wert 5 entspricht.

Diese Funktion wird in LabVIEW durch das Element „boolschesArraynachInteger“ ermöglicht.

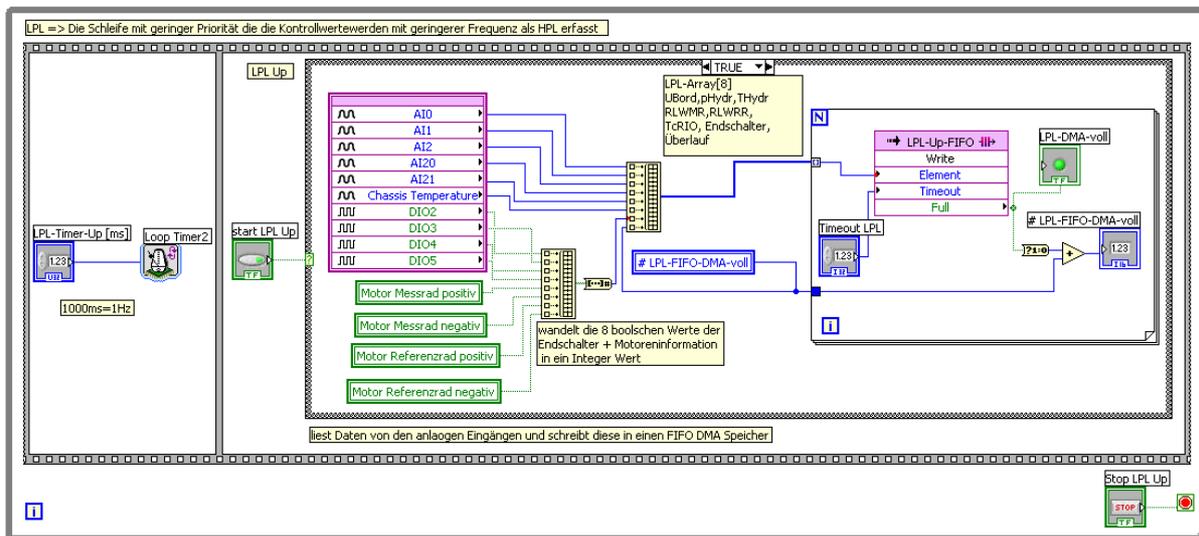


Abbildung 7.7: LPL-Up Modul

Die Abbildung 7.7 zeigt die Implementierung des LPL-Up-Moduls. Nachdem der Timer gesetzt wurde, wird zunächst das Eingabeelement „start LPL Up“ und eine CASE-Struktur verwendet um die Übertragung der Daten während der Ausführung stoppen zu können. Zunächst wird mittels einem FPGA I/O Element die Kontrollgrößen Bordspannung (AI0), der Hydraulikdruck (AI1), die Hydrauliktemperatur (AI2), der Radlenkwinkel am Messrad (AI20), der Radlenkwinkel am Referenzrad (AI21) und die Gehäusetemperatur des CRIO, sowie die digitalen Eingänge der Endschalter ausgelesen. Zur weiteren Kontrolle wurden Variablen übergeben welche anzeigen ob ein Motor gerade aktiv ist. Alle diese digitalen Größen werden wie oben beschrieben in einen Integer verwandelt. Die oben aufgezählten Kontrollgrößen werden in ein Integerarray, zusammen mit dem digitalen Integer und der Statusinformation #LPL-FIFO-DMA-voll, die falls ein Überlauf am DMA-Kanal auftrat größer 1 ist, gepackt. Dieses Array wird dann über eine FOR-Schleife an den Eingang des Elements des DMA-Kanals angeschlossen. Die FOR-Schleife geht der Reihe nach die Elemente des Arrays durch und sendet diese jeweils über den DMA-Kanal. Hierbei werden die reinen Integerwerte gesendet.

D.h. läuft der DMA-Kanal voll, wird die Reihenfolge der Variablen nicht mehr eingehalten und die Informationen können nicht mehr zugeordnet werden. Um diesen Überlauf zu detektieren wird die Variable #LPL-FIFO-DMA-voll verwendet, die bei jedem erkannten Überlauf inkrementiert wird und als letztes Element des Arrays mit übertragen wird.

7.1.7 HPL – Modul

Das HPL-Modul ist wie das LPL-Up-Modul aufgebaut. Der Hauptunterschied ist, dass es andere Größen erfasst, einen anderen DMA-Kanal nutzt und zur Laufzeit schneller getaktet wird. Der hier verwendete DMA-Kanal trägt die Bezeichnung AI-FIFO, da der Name wie bereits erwähnt nicht mehr geändert werden konnte. Wie in der Abbildung 7.8 zu sehen ist werden über diesen „schnelleren“ DMA-Kanal die Messgrößen, die berechnete Drehzahl sowie die Überlauferkennung übertragen.

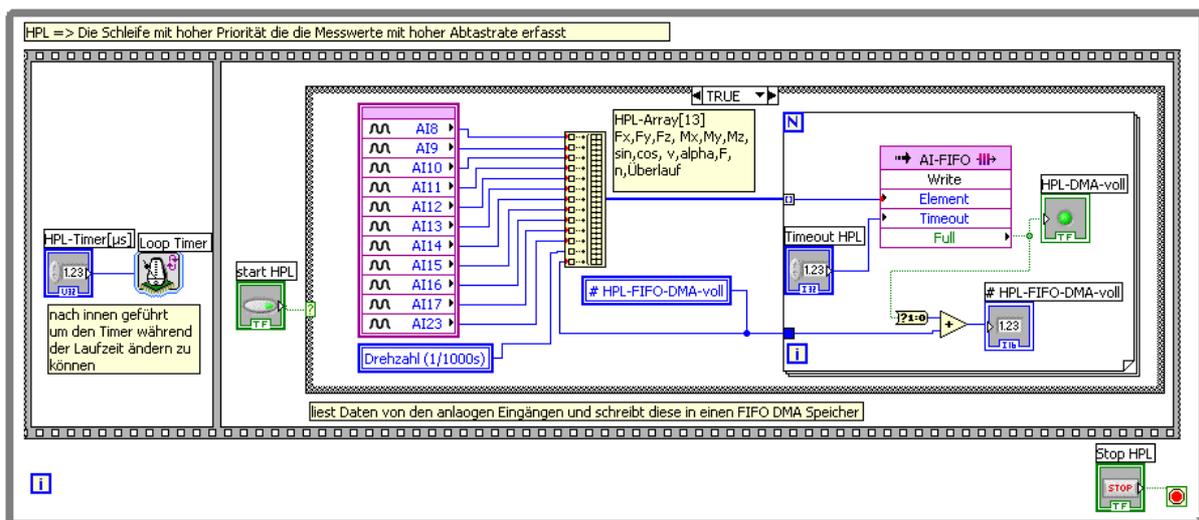


Abbildung 7.8: HPL-Modul

7.2 Broker-Programmierung

7.2.1 Normal Priority Loop – Modul

7.2.1.1 RT-FIFOs erstellen

Zunächst werden die Realtime-FIFOs erstellt, die die Datenkommunikation zwischen NPL-Modul und TCL-Modul in Echtzeit ermöglichen. Die Erstellung der RT-FIFOs wird vor dem

Kommunikationsaufbau durchgeführt, da das Anlegen der RT-FIFOs rechenzeitintensiv ist. Diese werden mittels dem Element „createRTFIFO“ erstellt. Die FOR-Schleife geht mit Hilfe der CASE-Struktur die einzelnen zu erstellenden FIFOs durch. Es werden folgende RT-FIFOs angelegt: „HPL-Daten“ für die Übertragung der Messdaten zum Host, „LPL-Daten“ für die Übertragung der Kontrolldaten zum Host, „Kalibrierdaten“ zum Übertragen der Kalibrierdaten der analogen NI-Module zum Host, „Startparameter“ zum Übertragen der Parameter die nur einmal zu Beginn der Übertragung vom Host übergeben werden, „Onlineeingabe“ Parameter die sich ändern können, wie z.B. Proportionalventilstufe und Sollschräglaufwinkel.

Die RT-FIFOs erhalten einen String als Bezeichner, welcher für die Unterscheidung und die Adressierung nötig ist, in Abbildung 7.9 der String „Startparameter“. Damit erhält diese RT-FIFO den Bezeichner „Startparameter“. Die Array-Konstante die in der Abbildung 7.9 über dem Bezeichner zu sehen ist, dient dazu der RT-FIFO den Typ Integer Array anzugeben. Die obere Konstante die im Beispiel den Wert 16 hat, gibt dabei die Arraygröße an. Die linke Konstante mit dem Wert 2 gibt an wie oft dieser Platz im RT-FIFO reserviert wird und gibt somit die FIFO-Länge an. In dem Beispiel wird daher ein 2-Dimensionales Integerarray der Größe 16*2 als FIFO mit dem Bezeichner „Startparameter“ angelegt. Ist ein RT-FIFO voll wird das zuletzt hineingeschriebene Element überschrieben. Da die FIFO-Länge dem Puffer entspricht und die Forderung besteht, dass es bei den Messgrößen, den HPL-Daten, zu keinem Überschreiben kommt, wird dessen FIFO-Länge so groß wie möglich (500 Elemente bei einer maximalen Arraygröße von 1000 und 13 Kanälen) gewählt.

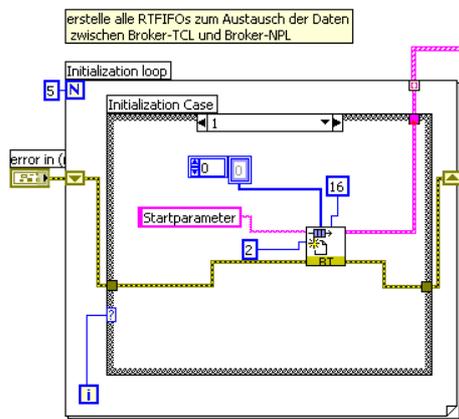


Abbildung 7.9: NPL: Erstellung der Realtime-FIFOs

7.2.1.2 Netzwerkkommunikation

Wie in der nächsten Abbildung zu sehen ist wird nach Erstellung der RT-FIFOs, das TCL-Modul aufgerufen (siehe Abbildung 7.10 unten) und nebenläufig dazu die Kommunikation zwischen den RT-FIFOs und der Netzwerkschnittstelle aufgebaut (siehe Abbildung 7.10 oben) und danach betrieben. In der äußeren Schleife wird zunächst die TCP-Verbindung auf-

gebaut, dazu wird dem in der Abbildung links oben zu sehenden Element der Port zugewiesen, an dem der TCP-Server auf eine eingehende Verbindung wartet. Erst wenn der TCP-Client, welches wie in der Modulspezifikation festgelegt der Host ist, eine Anfrage gestellt hat und die Verbindung mittels 3-Wege-Handshake hergestellt wurde, wird die innere Schleife, die den Datenaustausch regelt, gestartet. In dieser Schleife sind 2 Casestrukturen zu erkennen, die linke, der Writer Case, hat die Aufgabe die Daten die von der Time Critical Loop in die entsprechenden RT-FIFOs geschrieben wurden, in einen Bytestream zu packen und direkt an die Netzwerkschnittstelle über das TCP Send Recieve Element zu senden. Die rechte Casestruktur, der Reader Case, hat die Aufgabe die Daten von der Netzwerkschnittstelle, über das TCP Send Recieve Element zu lesen, den empfangenen Bytestream in die original Daten zurückzuwandeln und in die entsprechenden RT-FIFOs zu schreiben, damit das TCL-Modul diese lesen kann.

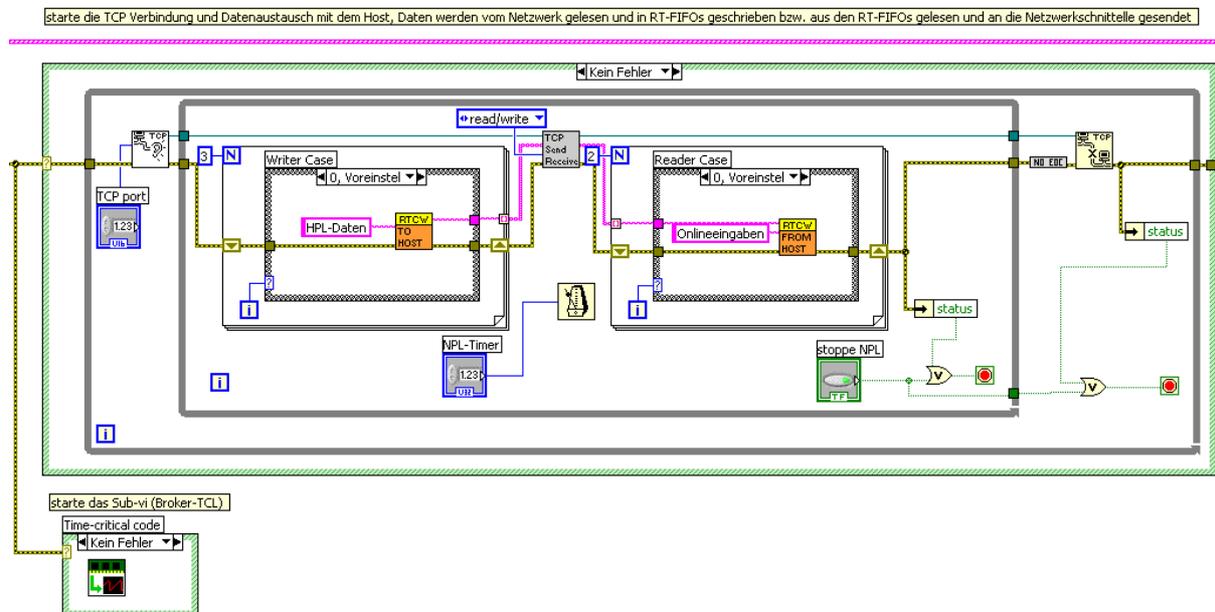


Abbildung 7.10: NPL: Sende/Empfange Daten

Der Writer Case arbeitet innerhalb der linken FOR-Schleife die 3 Fälle HPL-Daten, LPL-Daten und Kalibrierdaten ab. Das dort verwendete SubVI „RTCW TO HOST“ erhält als Eingabe die Bezeichnung des jeweiligen RT-FIFOs und gibt als Ausgabe den Bytestream zurück. Dieses SubVI entfernt aus dem entsprechenden RT-FIFO die gespeicherten Elemente und wandelt diese in einen TCP-Bytestream, der entsprechende Kontrollelemente enthält, um die Daten auf dem Host wieder in den richtigen Typ verwandeln zu können.

Der Reader Case arbeitet innerhalb der rechten FOR-Schleife die 2 Fälle Onlineeingaben und Startparameter ab. Das dort verwendete SubVI „RTCW FROM HOST“ bekommt als Eingabe den jeweiligen Bytestream, wandelt diesen in die Daten mit entsprechendem Typ um und speichert diese in das zugehörige RT-FIFO.

Die beiden SubVIs „RTCW TO HOST“ und „RTCW FROM HOST“ können mit Hilfe des LabVIEW-Kommunikationswizzard erstellt werden.

7.2.2 Time Critical Loop – Modul

7.2.2.1 FPGA-Verbindung herstellen

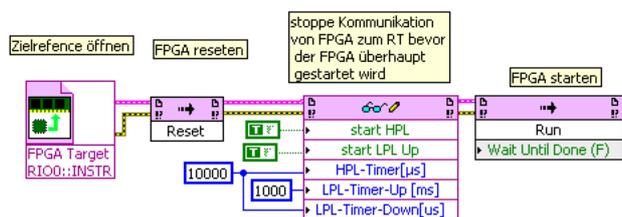


Abbildung 7.11: TCL: FPGA referenzieren und starten

Zunächst wird wie in Abbildung 7.11 zu sehen ist mittels FPGA-Target Element das Zielsystem referenziert. Nachdem das CRIO referenziert ist wird es mittels dem Reset-Element, resettet und in den Ursprungszustand gesetzt. Dies ist nötig, da sich in dem Speicher der RT-FIFOs noch alte Daten befinden können. Danach wird mittels Interrupt, die Variablen „start HPL“ und „start LPL Up“ auf „true“ gesetzt und die Timer der Schleifen auf die in der Abbildung zu sehenden Startwerte gesetzt. Dies ist nötig damit die Schleifen auf dem FPGA ausgeführt werden und dadurch Daten in die DMA-Kanäle geschrieben werden. Sind die DMA-Kanäle leer, können keine Daten gelesen werden und deswegen keine Kommunikation aufgebaut werden.

7.2.2.2 Konfigurationsdaten übertragen und DMA starten

In der folgenden Schleife wird gewartet bis die Variable „Kalibrierungsdaten sind verfügbar“, welche per Interrupt gelesen wird, „true“ wird. Danach werden erst die Kalibrierdaten des Moduls NI-9205 und dann die des Moduls NI-9263 mittels Interrupt gelesen. Da über die RT-FIFOs keine Cluster übertragen werden können, werden diese entweder nach String geflattet, oder wie in der Abbildung 7.12 oben zu erkennen ist in ein Integerarray gepackt und dieses in das RT-FIFO geschrieben. Diese Variante ist schneller, da nicht die kompletten Kontrolldaten (Typ, Name, etc.) in String verwandelt und übertragen wird.

Daran folgend wird die Puffergröße der DMA-Kanäle auf der RT-Seite festgelegt. Diese ist deutlich größer zu wählen als die Puffergröße der FPGA-Seite, da das RT-System im ver-

gleich zum FPGA nicht echt parallel arbeitet und sich somit die Daten im Speicher auf der RT-Seite stauen. Nachdem die Puffergröße festgelegt ist werden die beiden DMA-Kanäle gestartet. Als ausreichend große Puffergrößen wurden für die AI-FIFO (für HPL Daten) 500000 Elemente und für die LPL-Up-FIFO (für LPL Daten) 350 Elemente im Versuch festgelegt.

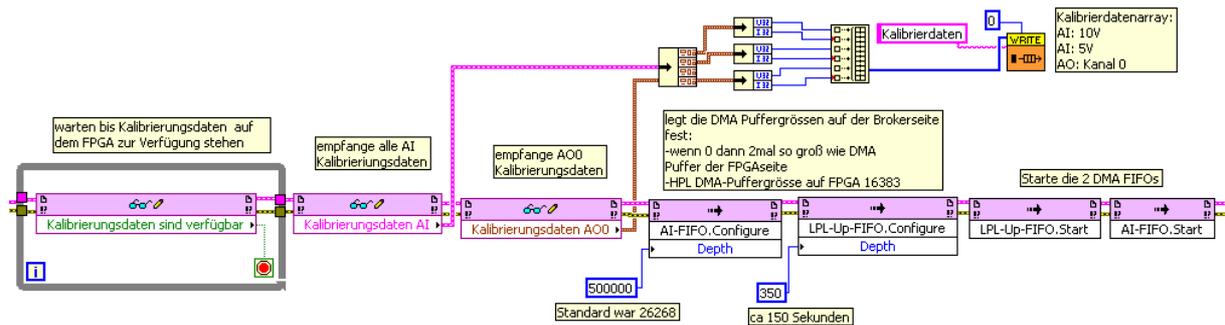


Abbildung 7.12: TCL: Kalibrierdaten übertragen und DMA konfigurieren

7.2.2.3 Datenaustausch zwischen FPGA und NPL

Es wurde davon ausgegangen, dass innerhalb der Hauptschleife, die direkt nach dem Starten der DMA-Kanäle betreten wird, alle enthaltenen Schleifen „parallel“ abgearbeitet werden. Dies ist allerdings nicht der Fall, wie sich während der Programmierung gezeigt hat. Der NI-Support bestätigen, dass die Schleifen nur sequentiell abgearbeitet werden, da auf dem CRIO standardmäßig nur ein Thread aktiviert ist, was aber in der Dokumentation nicht zu finden ist. Ein Aktivieren mehrerer Threads wird durch Setzen der Variablen `ESys.StdNParallel` auf 1 innerhalb der Datei `ni-rt.ini` auf dem RT-System erreicht. Dies führt allerdings dazu, dass keine Supportleistung durch den NI-Support nach Aktivierung des Multithreading erbracht wird. Daher wurde innerhalb dieser Arbeit mit Rücksprache des Betreuers darauf verzichtet das Multithreading des RT-Systems zu aktivieren.

Die folgenden Kapitel beschreiben die Implementierung der jeweiligen Schleife.

7.2.2.4 TCL – HPL – Schleife

Die folgende Abbildung zeigt die Durchsatzregelung oben rechts, die Überlauferkennung darunter und die Datenvermittlung vom DMA-Kanal zur NPL.

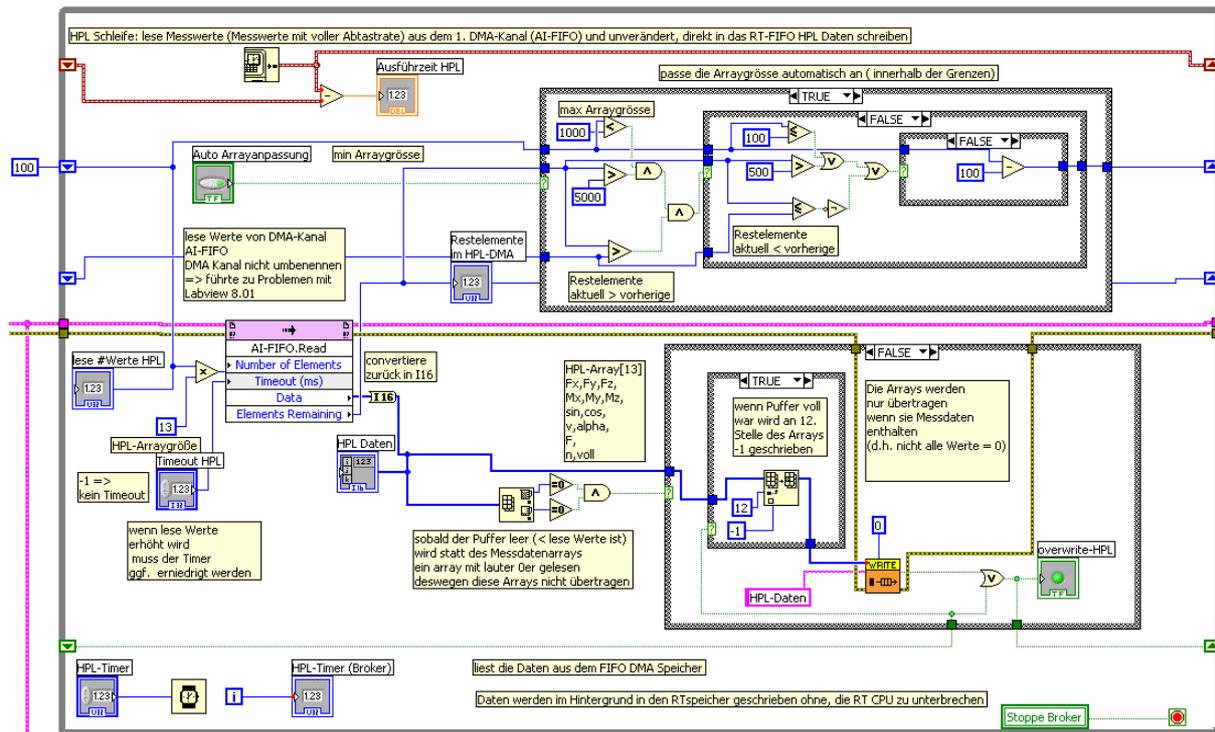


Abbildung 7.13: TCL: High Priority Loop

7.2.2.4.1 Datenvermittlung von DMA-Kanal zu NPL

Die Abbildung 7.13 zeigt die High Priority Loop, die die Messgrößen aus dem DMA-Kanal „AI-FIFO“ liest und in die RT-FIFO „HPL-Daten“ schreibt. Das Element „AI-FIFO.Read“ ermöglicht das Lesen der Elemente aus dem DMA-Speicher, die an dem Ausgang „Data“ als Integerarray entnommen werden. Der Eingang „Number of Elements“ beträgt ein Vielfaches der HPL-Arraygröße, damit die Daten Blockweise gelesen werden. Wichtig ist, dass, wie bereits in dem Kapitel FPGA-Programmierung erwähnt, der DMA-Puffer nicht überläuft, da sonst Daten verloren gehen. Wenn die Anzahl der verlorenen Daten nicht ein vielfaches der HPL-Arraygröße ist, können die einzelnen Werte den entsprechenden Messgrößen nicht mehr zugeordnet werden. Des Weiteren darf der DMA-Kanal nicht leer laufen, da sonst die Abarbeitung nicht weitergeführt werden kann und die Kommunikation zusammen bricht. Dies tritt nur im Ausführungsmodus als Anwendung, nicht aber im Debugmodus auf. Befinden sich Elemente im DMA-Speicher, aber weniger als gelesen werden sollen, werden die fehlenden Daten mit 0ern aufgefüllt. Deshalb wird in der unteren Case-Struktur der Abbildung 7.13, geprüft ob das Array nur aus 0ern besteht und in diesem Fall das Array nicht an die RT-FIFO weitergeleitet. Enthält das Array Daten, dann wird die in der Abbildung 7.13 zu sehende Aktion ausgeführt, das Array wird in die RT-FIFO „HPL-Daten“ gespeichert.

7.2.2.4.2 Durchsatzregelung

Wie bereits beschrieben darf der Puffer des DMA-Kanals auf der RT-Seite weder überlaufen, sonst gehen Daten verloren, noch darf er leerlaufen, denn sonst droht der Kommunikationszusammenbruch. Um dies zu verhindern gibt es 2 Einflussgrößen. Die Abtastrate der HPL-Schleife des FPGAs, da diese der Frequenz entspricht mit der der DMA-Kanal gefüllt wird und die Anzahl der Elemente die pro Schleifendurchlauf der „HPL-Schleife“ auf dem Broker die Elemente aus dem Puffer liest.

1.Variante: Bei drohendem Überlauf die Frequenz verringern und bei drohendem Leerlaufen die Frequenz erhöhen.

2.Variante: Bei drohendem Überlauf die Anzahl der zu lesenden Elemente erhöhen und bei drohendem Leerlaufen diese verringern.

3.Variante: Frequenz und Anzahl der zu lesenden Werte auf einen festen Wert stellen, der beide Pufferextreme verhindert.

Die 1. Variante wird nicht verwendet, da eine feste Abtastrate eingestellt wird und die Regelung langsamer ist, da der Frequenzwechsel erst mittels Interrupt an den FPGA übermittelt wird. Die 3.Variante wird ebenfalls nicht verwendet, da nur eine einzige Frequenz möglich ist und diese aufgrund des Einpendelvorgangs niedrig, maximal 500Hz (Erfahrungswert) sein wird.

Deswegen wurde entsprechend der 2.Variante eine Durchsatzregelung implementiert, die in der Abbildung 7.13 oben zu sehen ist. Die maximale übertragbare Arrayanzahl die während der Erstellung des RT-FIFOs, auf 1000 gesetzt wurde, was einer maximalen Elementanzahl von 13000, bei 13 Elementen die mit voller Abtastrate übertragen werden, entspricht. Somit ist die obere Grenze der „lese #Werte HPL“ Variablen 1000, die untere Grenze wurde auf 100 Elemente festgelegt, da die Anpassung der Regelung in 100er Schritten, erfolgt.

Es wird kein 2. Punktreger verwendet, da dieser in jedem Schleifendurchlauf die Variable „lese #Werte HPL“ verändern würde und damit zu einem dauernden Ändern der Abarbeitungszeit führt.

Deshalb wurde zunächst ein 3-Punktreger implementiert, der abhängig von der Anzahl der Restelemente regelt. Dieser Regler führte in der Praxis zu einem Überschwingen, wodurch sich die Anzahl der Restelemente immer wieder stark änderte, was zur Folge hatte, dass sich der Delay der Daten veränderte und sich als großer Jitter bemerkbar machte. Dies hätte eine Messungsdurchführung sehr beeinträchtigt, da z.B. auf Veränderung der Fahrzeuggeschwindigkeit zu spät reagiert werden könnte.

Es wurde die folgende 3-Punkt-Reglung mit Schwellenwerten umgesetzt, die neben der Betrachtung der absoluten Restelemente eine Betrachtung der Änderung, welche der Ableitung entspricht, vorsieht:

- Wenn die Anzahl der Restelemente im DMA-Puffer > 5000 ist und größer der Anzahl der Restelemente im Schleifendurchlauf zuvor, wird die Variable „lese #Werte HPL“ um 100 erhöht, falls diese nicht bereits das Maximum von 1000 erreicht hat. Damit werden im nächsten Durchlauf 1300 Elemente mehr gelesen.
- Wenn die Anzahl der Restelemente im DMA-Puffer < 500 ist und kleiner der Anzahl der Restelemente im Schleifendurchlauf zuvor, wird die Variable „lese #Werte HPL“ um 100 verringert, falls diese nicht bereits das Minimum von 100 erreicht hat. Damit werden im nächsten Durchlauf 1300 Elemente weniger gelesen.
- Ansonsten wird der vorherige Wert beibehalten.

7.2.2.4.3 Pufferüberlauferkennung

Ist dieses RT-FIFO voll, d.h. es wird das zuletzt eingefügte Element überschrieben, dann meldet das RT-FIFO-WRITE-Element am Ausgang „overwrite = true“. Dieses wird detektiert und mittels Shiftregister in den nächsten Schleifendurchlauf übertragen. Ist dieser Schieberegistereingang „true“, wird der innerste Case „true“ und es wird an die 13. Stelle (entspricht dem Index 12) des HPL-Datenarrays eine -1 geschrieben. Die 13. Stelle entspricht dem Integer der einen Überlauf bei der DMA-Übertragung detektiert. Durch einen ODER-Vergleich des Schieberegistereingangs mit dem nächsten „overwrite“ Wert, wird dauerhaft eine -1 für Netzwerkpufferüberlauf gesetzt.

Zusammenfassend für die Pufferüberlaufdetektion der HPL-Daten gilt:

Ist der Wert der Variablen „Überlauf“, der auf dem Host ankommt, 0 kam es zu keinem Pufferüberlauf. Ist der empfangene Wert -1 kam es zu einem Pufferüberlauf des RT-FIFO „HPL-Daten“, was einem Netzwerkpufferüberlauf entspricht. Ist der empfangene Wert weder 0 noch -1, kam es zu einem DMA-Pufferüberlauf bei dem die Reihenfolge der Kanäle nicht mehr gegeben ist und somit ein anderer Kanal als „Überlauf“ interpretiert wird.

7.2.2.5 TCL – NPL-Up – Schleife

Diese Schleife ist genauso wie die HPL-Schleife aufgebaut, da sie die gleiche Funktionalität hat. Der einzige Unterschied ist, dass die Kontrolldaten statt der Messdaten übertragen werden und daher ein anderer DMA-Kanal und ein anderes RT-FIFO genutzt wird. Da auf dem FPGA mit deutlich niedrigerer Frequenz (entsprechend der Tabelle 7-1) die Daten erfasst und in den „LPL-Up-FIFO“ DMA-Kanal geschrieben werden, werden auf der Broker Seite des DMA-Kanals deutlich weniger Werte (die genaue Anzahl wird im nächsten Abschnitt erläutert) als in der HPL-Schleife pro Schleifendurchlauf gelesen.

Da ein LPL-Array eine Größe von 8 hat ist der Multiplikator der „lese #Werte LPL“ Variable 8, wie in der Abbildung 7.14 auf der linken Seite zu erkennen. Somit wird gewährleistet, dass immer ganze Arrayblöcke am Stück gelesen werden.

Die Überlauferkennung funktioniert äquivalent zu der, der „HPL-Schleife“.

Es wird allerdings ein einfacherer Regler, als der der „HPL-Schleife“, implementiert, da die Anzahl der Restelemente sich deutlich seltener und langsamer ändert.

Es ist folgende Regelung implementiert:

Anzahl der Restelemente > 49 \Rightarrow „lese #Werte LPL“-Variable = 10

49 > Anzahl der Restelemente > 21 \Rightarrow „lese #Werte LPL“-Variable = 3

21 > Anzahl der Restelemente > 7 \Rightarrow „lese #Werte LPL“-Variable = 2

7 > Anzahl der Restelemente \Rightarrow „lese #Werte LPL“-Variable = 1

Bei dieser Regelung fällt auf, dass bei einem Anstieg der Restelemente auf über 49 die „lese #Werte LPL“-Variable mit 10, im Vergleich zu den anderen Größen sehr hoch gesetzt wird. Da die Kontrollgrößen nur ziemlich selten im Vergleich zu den Messgrößen erfasst werden und es im Normalfall zu keiner Datenhäufung kommt, pendelt die „lese #Werte LPL“-Variable zwischen 1 und 3. Kommt es auf dem RT-System, z.B. wegen Betriebssystemoperationen, zu einem oder mehreren Aussetzern, kann sich der Puffer doch stärker füllen. Um in diesem Fall den daraus resultierenden großen Delay schnellstmöglich zu reduzieren, werden 10 * 8 Elemente pro Schleifendurchlauf auf einmal gelesen.

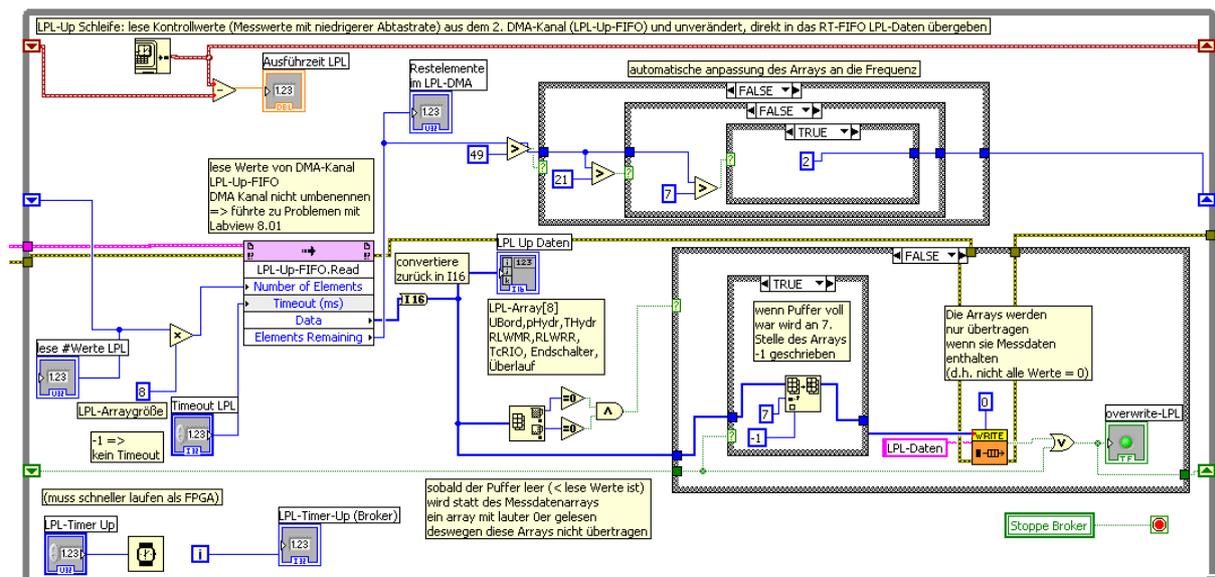


Abbildung 7.14: TCL: Low Priority Loop – Up

7.2.2.6 TCL – LPL-Down – Schleife

Die Hauptaufgabe der LPL-Downschleife ist die Datenvermittlung von der NPL zum FPGA. Eine weitere Aufgabe ist die Schleifentimer der Schleifen auf dem RT-System zu setzen und zu ändern.

7.2.2.6.1 Datenvermittlung von NPL zum FPGA

Da die Interruptkommunikation wie bereits in Kapitel 5.1.4.2 angesprochen, sehr langsam ist und die anderen Berechnungen des Systems kurzzeitig stoppt, ist das Setzen von IRQs zu minimieren. Daher wird nur dann ein Interrupt auf ein entsprechendes Element gesetzt, wenn dieses sich geändert hat.

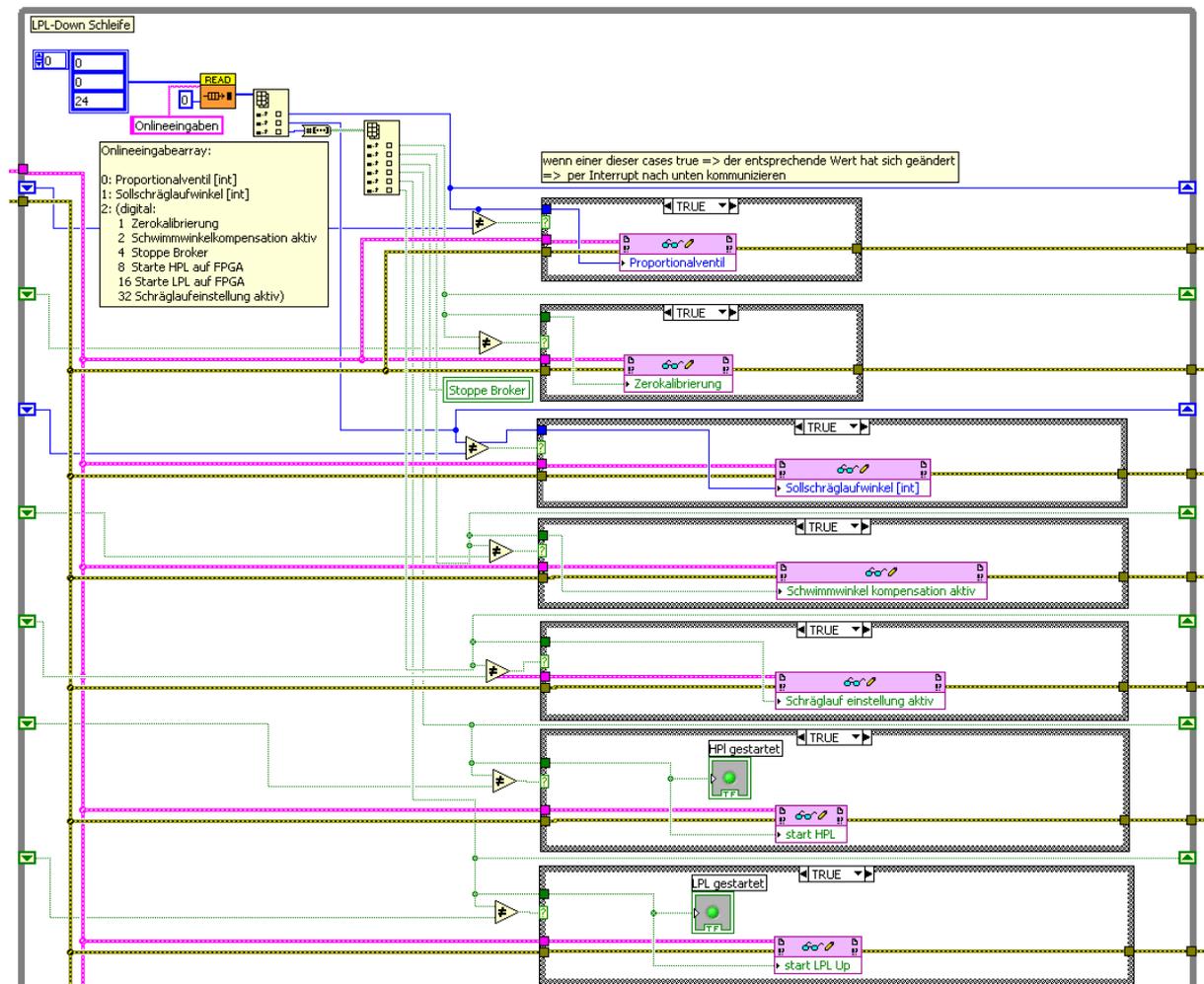


Abbildung 7.15: TCL: Low Priority Loop – Down, oberer Teil

Für die „Onlineeingaben“-Parameter, die sich während der Kommunikation ändern, wird daher für jedes einzelne Element nur dann ein Interrupt gesetzt wenn dieses Element sich geändert hat. Dies wurde so wie es in Abbildung 7.15 zu sehen ist umgesetzt. Zunächst wird in der

Abbildung wie oben links zu erkennen, das Onlineeingabearray aus dem RT-FIFO „Onlineeingabe“ gelesen. Dieses wird in die Integer Proportionalventil, Sollschräglauf und digital aufgespalten. Der Integer „digital“ enthält die folgenden boolschen Werte, die analog zu der unter Kapitel 7.1.6.2 verwendeten Logik, ebenfalls entpackt werden. Dies sind die Variablen: „Schwimmwinkelkompensation aktiv“, „Schräglaufeinstellung aktiv“, „start LPL“ und „start HPL“. Die Erkennung ob ein Wert neu ist erfolgt jeweils über ein Schieberegister und dem Vergleich mit dem neuen Wert. Ist dieser neue Wert ungleich dem alten, ist der folgende Case „wahr“ und der neue Wert wird mittels Interrupt auf den FPGA übertragen.

Die Abbildung 7.16 zeigt oben, entsprechend der „Onlineeingabe“, die Interruptübertragung der „Startparameter“. Da diese sich nicht, bzw. nur zu Beginn der Kommunikation ändern, wird das ganze Startparameterarray mit dem vorherigen verglichen. Bei Änderung mindestens eines der darin enthaltenen Elemente, werden in der in Abbildung 7.16 oberen CASE-Struktur alle Startelemente per Interrupt an den FPGA übermittelt.

7.2.2.6.2 Setzen und Ändern der Timerstufen

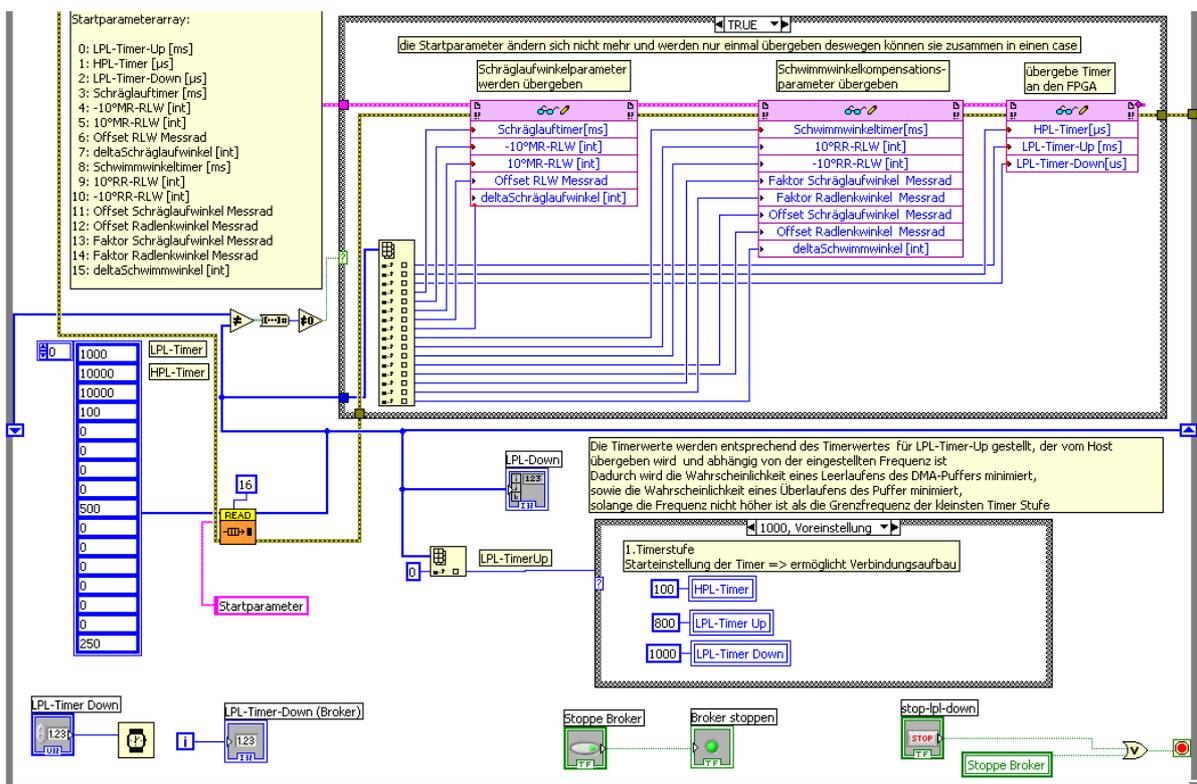


Abbildung 7.16: TCL: Low Priority Loop – Down unterer Teil

Die untere in Abbildung 7.16 zu sehende CASE-Struktur übernimmt die Aufgabe des Setzens und Ändern der Schleifentimer. Wie in Kapitel 7.2.2.3 beschrieben, werden alle diese Schleifen innerhalb der Hauptschleife nur sequentiell ausgeführt. Da die Schleifen hintereinander

ausgeführt werden, hat jede Schleifenausführung somit Einfluss auf die anderen. Wird hierbei nur eine der 3 Schleifen schneller ausgeführt, folgt daraus, dass alle Schleifen schneller ausgeführt werden. Deswegen werden die Kontrollgrößen deutlich höher, entsprechend der Tabelle 7-1, abgetastet als in der Spezifikation mit 1Hz angestrebt war. Somit bremst die am langsamsten getaktete Schleife (d.h. die mit dem kleinsten Timer), die anderen aus.

Da die ersten Schleifendurchläufe durch Anlegen von Variablen und weitere durch das Betriebssystem unbekannte Vorgänge, sehr langsam ist, kann der Broker nur mit recht kleinen Schleifentimern gestartet werden. Hier hat sich durch Versuche ergeben, dass als HPL-Timer = 100, LPL-Timer Up = 800 und LPL-Timer Down = 1000 eine passende Timerwahl zum Starten ist. Werden kleinere Timer, zum Starten gewählt, kann es passieren, dass das System abstürzt.

Hat sich das System mit den Startparametern „eingespielt“, können die Timer zur Laufzeit erniedrigt werden, wodurch sich die Ausführungszeit verringert. Jedoch ist es erforderlich gleichzeitig die Abtastraten des FPGAs zu erhöhen, um ein Leerlaufen der DMA-Kanäle und somit einen Absturz zu verhindern. Durch Verändern der Timer zur Laufzeit lässt sich die Abtastrate deutlich erhöhen.

Timer werden abhängig von der vom Host übergebenen Variable „LPL-Timer Up“, die den LPL-Timer Up auf dem FPGA und damit die Kontrollgrößenfrequenz setzt, gesetzt entsprechend der in Tabelle 7-1 in Kapitel 7.3.6. festgelegten Stufen.

7.3 Host-Programmierung

Bei der Host-Programmierung wird auf eine detaillierte Beschreibung der einzelnen Komponenten verzichtet, da diese den Umfang dieser Masterarbeit überschreiten würde.

7.3.1 Hauptprogramm

Das Hauptprogramm besteht in erster Linie aus dem grafischen Benutzerinterface (GUI) und dem Programmablauf. Die Bedienungsanleitung der Software kann im Anhang I eingesehen werden.

7.3.1.1 GUI

Die GUI besteht aus zwei Hauptbestandteilen, dem oberen Tabelement und dem unteren Statusanzeigenelement. Mit dem Tabelement ist es mittels Karteireiter möglich, durch die einzelnen SubGUIs zu navigieren. Es kann durch folgende SubGUIs navigiert werden: Konfiguration, Kalibrierdaten, R_{dyn} -Berechnung und Messungen. Das Statusanzeigenelement, das das

untere Viertel der Benutzeroberfläche einnimmt, wird die ganze Zeit angezeigt und zeigt die aktuellen Statusmeldungen.

- **Konfigurations-GUI:**
Dient dazu die Konfigurationsdaten eingeben zu können. Es sind folgende Eingabemöglichkeiten vorhanden: Messdateipfad, Messdateiname, Messfelgennutzung, Toleranzangaben (Schräglauf-, Schwimmwinkel und Geschwindigkeit), delta Stufencounter, Messfrequenz, sowie weitere Eingabefelder für Reifen und Asphalteeigenschaften.
- **Kalibrierdaten-GUI**
Hier sind bereits alle aktuellen Kalibrierdaten enthalten. Diese können bei Änderung einer Kalibrierung geändert werden. Des Weiteren werden die einzelnen Stufen der Bremsmomentverstellung, in dieser SubGUI festgelegt. Diese sind ebenfalls auf Standardwerte voreingestellt die für die Messungsdurchführung der Validierung verwendet wurden. Bei anderen Bedingungen ist es nötig die Stufen anzupassen.
- **R_{dyn}-Berechnung-GUI**
Zunächst wird die Sollgeschwindigkeit angegeben. Die in der Konfiguration eingestellte Schräglaufwinkel- und Geschwindigkeitstoleranz, wird zur Kontrolle angezeigt. Der Button „Start R_{dyn} Berechnung“ startet die R_{dyn}-Berechnung und die aktuelle Geschwindigkeit wird angezeigt. Befinden sich die Geschwindigkeit und der Schräglaufwinkel im Toleranzbereich, beginnt die entsprechende LED zu leuchten. Ist die R_{dyn}-Berechnung durchgeführt, erscheint das Ergebnis und der Button „Start R_{dyn} Berechnung“ wird automatisch deaktiviert.
- **Messung-GUI**
Dieses GUI ist so aufgebaut wie das R_{dyn}-Berechnungs-GUI enthält allerdings weitere Eingabelemente, wie Sollschräglaufwinkel, Messungsart und Bemerkungen. Letzteres wird direkt mit in die Messdatei geschrieben. Als Ausgabeelement gibt es zusätzlich die Anzeige der aktuellen Bremsmomentstufe und die Messradgeschwindigkeit.

7.3.1.2 Programmablauf

Der Ablauf ist wie folgt aufgebaut, dabei wird auf eine Detaillierung der Implementierung verzichtet:

- GUI erstellen und Ausgabelemente auf Standardanzeige setzen
- Konfigurationseingaben lesen, speichern und Konfigurationsdatei schreiben
- Kalibrierdaten lesen, umrechnen, speichern und Startarray erstellen
- CRIO suchen und reseten

- Netzwerkverbindung zu CRIO aufbauen
- Netzwerkkommunikation

7.3.2 Konfiguration

Erfasst die Eingabelemente der Konfigurations-GUI und speichert diese in Variablen, die später verwendet werden. Außerdem werden die eingegebenen Konfigurationsdaten in eine Textdatei geschrieben, die im angegebenen Pfad gespeichert wird. Diese Implementierung ist im Anhang D1.1 zu finden.

7.3.3 Kalibrierung

Zunächst werden die in der Kalibrierdaten-GUI eingegebenen Daten gelesen. Die Kalibrierdaten werden gespeichert, damit sie für die spätere Umrechnung der erfassten Größen von der Eingangsspannung in die eigentliche Einheit umgerechnet werden können.

Da auf dem FPGA in Integer gerechnet wird, wird für die Schwimmwinkelkompensation und die Schräglaufeinstellung, die auf dem FPGA laufen, die Umrechnungsfaktoren und Offsets der betroffenen Größen (Schräglaufwinkel und die beiden Radlenkwinkel) in Integer benötigt.

Diese werden in 2 Schritten umgerechnet.

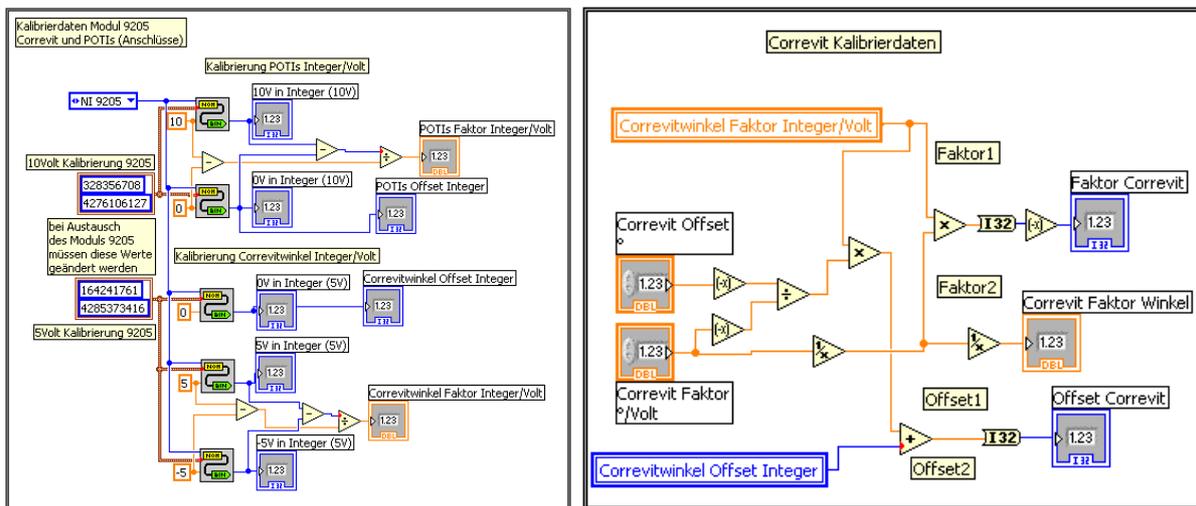


Abbildung 7.17: Umrechnung Volt-Integer Abbildung 7.18: Correvitkalibrierdaten in Integer

Zunächst wird aus den Kalibrierdaten des NI-9205 Moduls Umrechnungsfaktor und Offset für die Umrechnung von Volt nach Integer berechnet. Dies ist in der Abbildung 7.17 zu sehen. Der obere Teil berechnet aus den 10Volt-Kalibrierdaten des NI-9205 Moduls den Umrechnungsfaktor und Offset für die Radlenkwinkel, da diese im Messbereich bis 10Volt erfasst

werden. Der untere Teil berechnet aus den 5Volt-Kalibrierdaten des NI-9205 Moduls den Umrechnungsfaktor und Offset für den Schräglaufwinkel, da dieser im Messbereich bis 5Volt erfasst wird.

In einem 2. Schritt wird dann aus den eingegebenen Kalibrierdaten die auf dem FPGA zu verwendenden Kalibrierdaten berechnet.

Erläuterung am Bsp. des Schräglaufwinkels:

Mittels der 5Volt-Kalibrierdaten des Analogmoduls und dem SubVI „Nominal to Binary“, welches in der LabVIEW Bibliothek vorhanden ist, werden die Spannungen $-5V$, $0V$ und $5V$ in Integer umgerechnet. Dies ist in der Abbildung 7.17 implementiert. Der $0V$ Spannungswert in Integer entspricht dem Correvitoffset in Integer. Durch Division der Differenz der Integerwerte von $5V$ und $-5V$ und der Differenz von $5V$ und $-5V$, ergibt sich der Correvitwinkelfaktor in Integer/V.

Die weitere Berechnung der Correvitkalibrierdaten ist in Abbildung 7.18 zu sehen. Dieser Correvitwinkelfaktor in Integer/V wird mit dem Kehrwert des eingegeben Correvitwinkelfaktors in $^{\circ}/V$ multipliziert. Das Ergebnis ist der Correvitwinkelfaktor in Integer/ $^{\circ}$.

Der eingegebene Correvitwinkeloffset in $^{\circ}$ dividiert durch Correvitwinkelfaktor in $^{\circ}/V$ ergibt den Correvitwinkeloffset in V. Dieser wird multipliziert mit dem Correvitwinkelfaktor in Integer/V. Das Ergebnis wird zu dem Correvitwinkeloffset der Integerumrechnung addiert. Das Ergebnis ist der Correvitwinkeloffset in Integer.

Die Kalibrierdaten der Radlenkwinkel, sowohl die des Messrades, als auch des Referenzrades, werden analog zum aufgezeigten Beispiel berechnet.

Des Weiteren werden die Schwimmwinkeltoleranz und die Schräglaufwinkeltoleranz in Integer umgerechnet. Dabei werden die eingestellten Toleranzen halbiert, d.h. es wird auf dem FPGA doppelt so genau geregelt, wie auf dem Host eingestellt wird. Dadurch wird gewährleistet, dass bei der Nachregelung während einer Messung, die Werte nicht außerhalb der Toleranzbereiche kommen und daher die Messung unterbrochen wird.

Auch die Endwinkel der Radlenkwinkelverstellungen werden mittels den ermittelten Kalibrierdaten in Integer umgerechnet. Die Endwinkel beenden die Motoransteuerung bevor die Radlenkwinkelverstellung die Endschalter auslösen.

All diese Werte werden zusammen in einem „Startparameter“-Array gespeichert, welches als Erstes an den Broker gesendet wird.

Im Kalibrierungsmodul werden auch die Stufen der Bremsmomentverstellung erfasst und in ein Array geschrieben, aus dem während der Messung nacheinander die Stufen gelesen werden.

Die Implementierung der kompletten Kalibrierung kann im Anhang D1.2 bzw. im Host.vi eingesehen werden.

7.3.4 Netzwerkverbindung mit CRIO herstellen

Zunächst werden alle im Netzwerk sich befindenden Systeme gesucht. Dann wird geprüft ob das System mit dem Hostnamen „CRIO“ dabei ist. Wurde das CRIO-System gefunden, wird es zunächst resetet. Dies ist nötig damit ein Verbindungsaufbau garantiert stattfindet. Wird das CRIO nicht ordnungsgemäß heruntergefahren, oder ist noch Speicher vom vorherigen Betrieb alloziert, ist ein Verbindungsaufbau nicht möglich oder es kann zu Fehlfunktionen während dem Betrieb kommen. Das Hostsystem wartet, bis das CRIO-System erneut im Netzwerk gefunden wird. Während des ganzen Vorgangs wird in der Statusanzeige der entsprechende Status als String eingeblendet. Die zugehörige Implementierung ist im Anhang D1.3 abgebildet.

Danach kann die TCP-Verbindung zum CRIO hergestellt werden. Dazu wird der angegebene TCP-Port, sowie die zuvor ermittelte IP-Adresse, verwendet. Der Host agiert als Client und sendet die TCP-Anfrage an das CRIO. Nachdem der TCP-3-Wege-Handshake abgewickelt ist, ist die Verbindung hergestellt.

7.3.5 Netzwerkkommunikation

Nachdem die Netzwerkverbindung aufgebaut wurde, kann die Netzwerkkommunikation erfolgen. Dazu werden umgekehrt zu denen im Broker-NPL-Modul verwendeten die Reader und Writercases auf dem Host durchgegangen. Dabei wird die Umwandlung in Bytestream, bzw. aus Bytestream in die original Daten vorgenommen. Diese sind im Einzelnen:

- WriterCases:
 - Startarray an CRIO schicken
 - Onlineeingaben an CRIO schicken
- ReaderCases:
 - Kalibrierdaten empfangen und speichern
 - Kontrollgrößen empfangen, umrechnen, anzeigen und ggf. Benutzer warnen bzw. System herunterfahren
 - Messgrößen empfangen, umrechnen, daraus R_{dyn} berechnen oder in Messdatei schreiben

7.3.6 Frequenzabhängige Timeranpassung

Innerhalb des Startarrays werden die Timer des FPGAs für die Messgrößenfrequenz (HPL) und Kontrollgrößenfrequenz (LPL-Up) vom Host an den Broker übertragen. Hierbei wird

ausgehend von der vom Benutzer angegebenen Frequenz, der HPL-Timer und der LPL-Timer-Up ermittelt. Aus dem LPL-Timer-Up werden die Timer der Brokerschleifen abgeleitet, entsprechend der Tabelle 8-1. Die vom Benutzer eingegebene Frequenz entspricht der HPL-Frequenz in der folgenden Tabelle.

Tabelle 7-1: Steuerung der in Kapitel 7.2.2.6.2 genannten Timer vom Host aus

Stufe	HPL-Frequenz [Hz]	HPL-Timer (FPGA) [μ s]	LPL-Timer-Up (FPGA) [ms]	LPL-Frequenz [Hz]
1	<500	1M/HPL-Frequenz	1000	1
2	500<f<2000	1M/HPL-Frequenz	100	10
3	>2000	1M/HPL-Frequenz	50	20

7.3.7 R_{dyn}-Berechnung

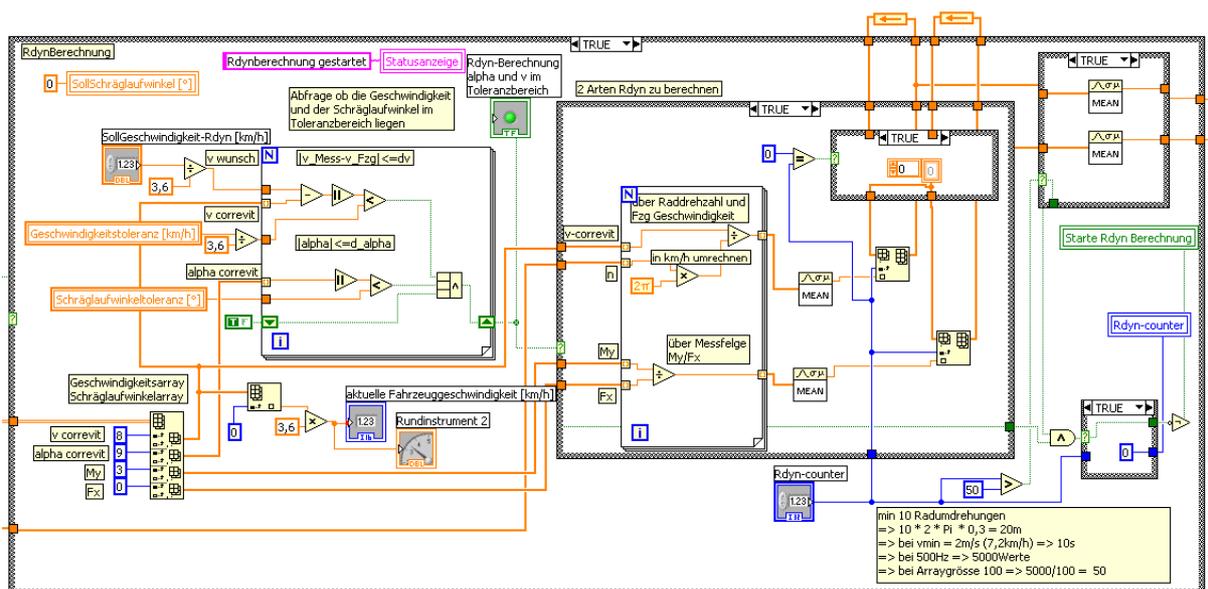


Abbildung 7.19: R_{dyn}-Berechnung

Die R_{dyn}-Berechnung wird innerhalb des CASE „Messdaten“ durchgeführt. Abbildung 7.19 zeigt die Umsetzung der R_{dyn}-Berechnung, die gestartet wird nachdem der Benutzer den Button „start R_{dyn}-Berechnung“ aktiviert hat. In der linken oberen Ecke ist zu erkennen, dass die Variable Sollschräglaufwinkel auf 0 gesetzt wird. In der FOR-Schleife auf der linken Seite in der Abbildung, wird geprüft ob die Differenz zwischen Sollgeschwindigkeit und Fahrzeuggeschwindigkeit innerhalb des Geschwindigkeitstoleranzbereichs liegt und ob der Schräglaufwinkel innerhalb des Schräglauftoleranzbereichs liegt. Die Abfrage wird in einer FOR-Schleife durchgeführt, da jedes einzelne Element der Arrays verglichen wird. Liegt nur ein

Element des Arrays außerhalb einer der beiden Toleranzen, wird dieses Array nicht für die R_{dyn} -Berechnung verwendet, da dieses für die R_{dyn} -Berechnung verfälschen würde. Die Arraygröße ist abhängig von der Regelung auf dem Broker und beträgt ein vielfaches von 100. Für die R_{dyn} -Berechnung wird die Frequenz zuvor auf konstant 500Hz gestellt, um immer die gleichen Bedingungen für die R_{dyn} -Berechnung zu schaffen.

Die R_{dyn} -Berechnung erfolgt wie in dem Methodikkapitel 6.5.4 bereits erläutert auf zwei Arten, wie in der Mitte der Abbildung 7.19 zu erkennen ist nach den Formeln (6.4) und (6.5). Danach wird jeweils aus dem so berechneten R_{dyn} -Array der Mittelwert gebildet. Diese Mittelwerte werden jeweils an ein bestehendes Array angehängt und per Rückgabeelemente (Pfeile rechts oben) in den nächsten Schleifendurchlauf übergeben. Nach 50 Schleifendurchläufen wird aus den gesammelten R_{dyn} der Mittelwert berechnet. Aus Speicherplatz und Rechenzeitgründen wird kein großes Array zusammengebaut und nur 1 Mittelwert gebildet. Diese beiden so ermittelten dynamischen Reifenhalmmesser werden in einem Ausgabelement dem Benutzer angezeigt. Der dynamische Reifenhalmmesser der vom Benutzer ausgewählt wird, wird gespeichert und in der Messung verwendet. Hier sei nochmals darauf hingewiesen, dass das R_{dyn} aus der Messfelge, wie sich in der Verifikation herausgestellt hat, in der verwendeten Implementierung nicht zu gebrauchen ist.

Berechnung der Schleifendurchläufe:

gegeben:

$n_{\text{min}} = 10$ Radumdrehungen; $v_{\text{min}} = 2\text{m/s}$; $f = 500\text{Hz}$; $\text{Arraygröße}_{\text{min}} = 100$ Werte; $R_{\text{dyn}} = 0,3\text{m}$

gesucht: Anzahl der Schleifendurchläufe (n):

$$s = 2 * \pi * n * R_{\text{dyn}}; t = \frac{v}{s}; \#Werte = f * t; n = \frac{\#Werte}{\text{Arraygröße}_{\text{min}}} \quad (7.1)$$

Durch einsetzen und ausrechnen, unter Verwendung der Formeln (7.1), ergibt sich die mindest Anzahl von Schleifendurchläufen $n = 50$.

7.3.8 Messung

Die Messung wird wie auch die R_{dyn} -Berechnung innerhalb des CASE „Messdaten“ durchgeführt. Die Umsetzung der Messungsdurchführung ist in Abbildung 7.20 zu sehen. Zunächst wird geprüft, ob der Timer des aktuellen Bremsmoments abgelaufen ist und es somit Zeit ist ein höheres Bremsmoment zu erzeugen. Hierzu wird zur Variable „Stufencounter“ die Variable „Ausführungszeit“, in der die ermittelte Zeit, die zwischen dem Start des letzten Schleifendurchlaufs und dem Aktuellen vergangen ist, gespeichert ist, addiert. Ist die Variable „Stufencounter“ größer dem vom Benutzer eingestellten „Stufencounter_delta“, wird in der CASE-Struktur, die in Abbildung 7.20 ganz links zu sehen ist, aus dem Proportionalventilarray

das nächste Element gelesen und in der Variablen „Proportionventil“ gespeichert. Dadurch wird im nächsten Schleifendurchlauf die neue Proportionalventilspannung an den Broker und von dort an den FPGA übergeben. Die CASE-Struktur in der Mitte der Abbildung 7.20 prüft, wie in der R_{dyn} -Berechnung ob die Fahrzeuggeschwindigkeit innerhalb der Sollgeschwindigkeit \pm Geschwindigkeitstoleranzbereich liegt und ob die Differenz zwischen Schräglaufwinkel und Sollschräglaufwinkel innerhalb des Schwimmwinkeltoleranzbereichs liegt. Abhängig davon wird der Ausgang dieses CASE „true“, falls als Messungsart „stationär“ (bei der nicht kontinuierlich die Messdaten aufgezeichnet werden) gewählt wurde, andernfalls wurde „dynamisch“ gewählt und es wird immer „true“ ausgegeben und somit alle Messdaten gespeichert. Die FOR-Schleife die auf der rechten Seite der Abbildung zu sehen ist berechnet aus der Drehzahl und R_{dyn} die Messradgeschwindigkeit die ebenfalls in der Messdatei gespeichert wird. Der in der Abbildung 7.20 kleine CASE oben entfernt aus dem Messdatenarray die Größen die von der Messfelge ermittelt wurden, sofern diese nicht montiert ist.

Sind alle Bremsmomentstufen durchgelaufen, werden alle Variablen die für die Messdurchführung verwendet wurden zurückgesetzt.

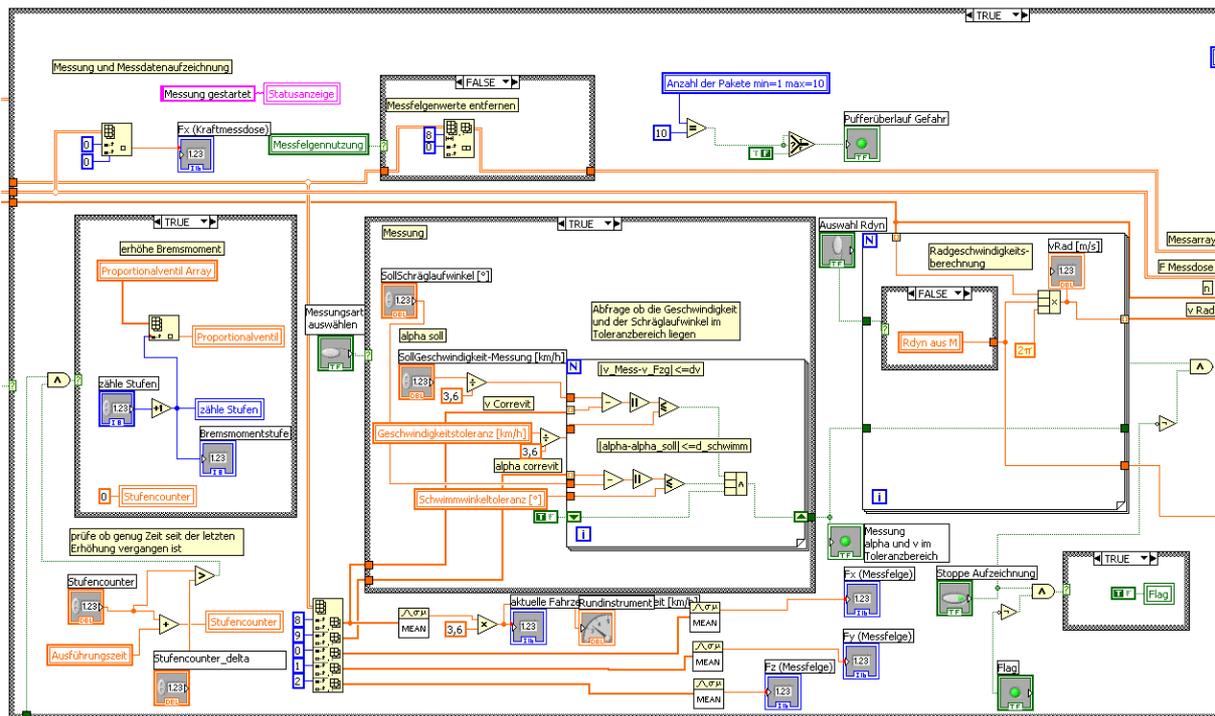


Abbildung 7.20: Messung – Messdurchführung

Die Messdatenspeicherung, welche direkt an die Messdurchführung folgt, wird nicht weiter erklärt, da es sich um eine Standardimplementierung handelt, die von LabVIEW zur Verfügung gestellt wird und nur mit den entsprechenden Eingängen belegt wird.

7.3.9 Statusanzeigen

Innerhalb des CASE „Kontrolldaten“, welcher im Blockschaltbild des Host.vi zu finden ist, werden die Kontrollgrößen in die physikalischen Einheiten umgerechnet und in den entsprechenden Anzeigen ausgegeben. Da es sich dabei um keine für das Projekt entscheidende Funktionalität handelt und eine detaillierte Beschreibung, wie bereits erwähnt, den Umfang dieser Arbeit übersteigen würde, wird darauf verzichtet.

7.3.10 Warnung

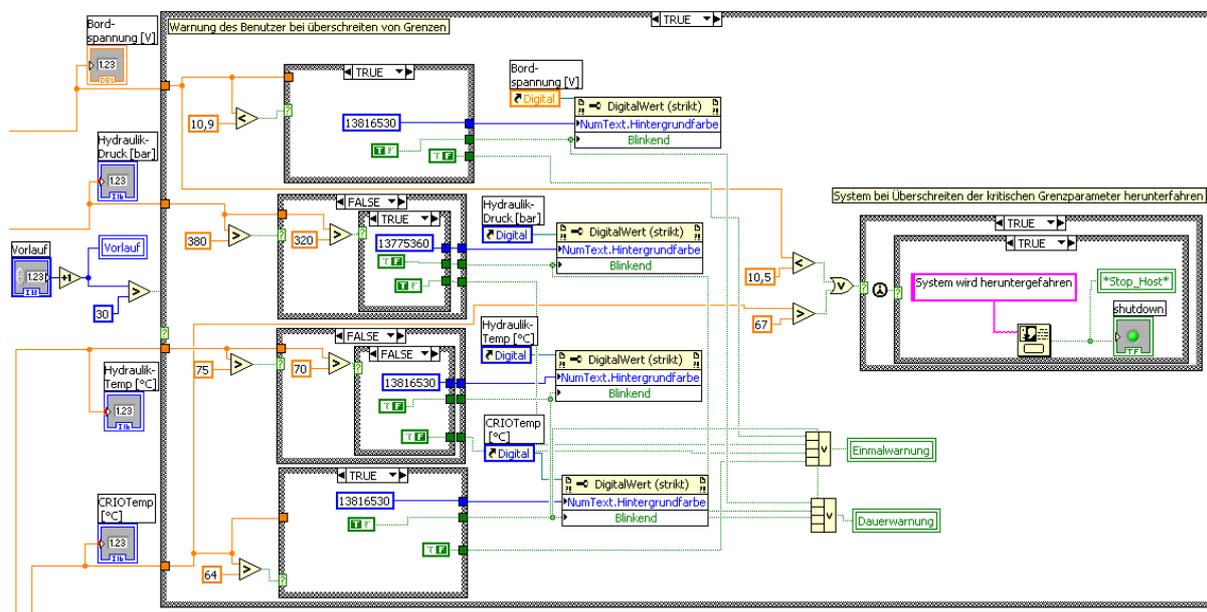


Abbildung 7.21: Warnung

Die Abbildung 7.21 zeigt die Implementierung des Moduls Warnung. Zunächst wird in den CASE-Strukturen auf der linken Seite geprüft in welchem Bereich sich die kritischen Kontrollgrößen Bordspannung, Hydraulikdruck, Hydrauliktemperatur und CRIOTemperatur befinden. Abhängig davon werden die entsprechenden Parameter, die im Folgenden erläutert werden, gesetzt. Tritt der Fall ein, dass keine Grenze überschritten wird, wie in dem 3. CASE von oben zu sehen ist, wird die Hintergrundfarbe auf die Standardfarbe gesetzt und „Blinkend“ sowie die Variable „Einmalwarnton“ auf „false“ gesetzt. Wird die 1. Grenze überschritten wie im 2. CASE (Hydraulikdruck) zu sehen ist, wird die Hintergrundfarbe auf Rot gesetzt, „Blinkend“ auf „false“ und „Einmalwarnton“ auf „true“. Wird die 2. Grenze überschritten, wie im 1. CASE, wird die Standardhintergrundfarbe übergeben, der „Einmalwarnton“ auf „false“ gesetzt und der Eingang „Blinkend“ auf „true“, diese Option lässt das Ausgabeelement rot blinken. Ist der Eingang „Blinkend“ „true“ wird auch der „Dauerwarnton“ auf „true“ gesetzt.

Auf der rechten Seite der Abbildung 7.21 wird geprüft ob die Bordspannung unter 10,5V gefallen ist oder die CRIO-Temperatur die 67° überschritten hat, diese Grenzwerte können dem Anforderungsdefinitionsdokument entnommen werden. Ist eine der Bedingungen erfüllt wird das System heruntergefahren. Dabei wird der Benutzer über ein Dialogelement das mitten im Bildschirm erscheint informiert.

8 Optimierungsmaßnahmen

In diesem Kapitel werden die Hauptoptimierungsmaßnahmen aufgezeigt.

8.1 DMA statt IRQ

Die Verwendung von direktem Speicherzugriff im Vergleich von Interrupts, bringt den höchsten Performancegewinn. Der Prototyp der die Interruptkommunikation verwendet, hatte gezeigt, dass mit ihm keine Frequenzen im Kilohertzbereich möglich sind.

8.2 Messgrößenaufteilung

Die Aufteilung der Messgrößen in Kontrollgrößen und Messgrößen hat einen erheblichen Einfluss auf die Performance und damit auf die maximale Abtastrate. Da allerdings die parallele Abarbeitung auf dem Broker nicht angewendet wurde, ist der Vorteil der Messgrößenaufteilung nicht ganz so hoch.

8.3 LabVIEW 8.20 statt 8.01

Der Umstieg von LabVIEW 8.01 auf 8.20 brachte hinsichtlich der Performance eine geringe Steigerung. Besonders die RT-FIFOs sind laut NI bei Frequenzen über 1kHz schneller geworden.

8.4 Berechnungen auf dem RT-System minimiert

Da das RT-System des CRIO der Flaschenhals des verteilten Systems ist, wurden die Berechnungen auf dem RT-System minimiert. So wurden z.B. alle Umrechnungen von Integer in Double auf den Host verlagert.

8.5 Timeroptimierung

Die für die einzelnen Schleifen des Brokers gewählten Timer wurden empirisch ermittelt indem kleinere Timer gewählt wurden bis es zu einem Absturz kam. Hierbei wurden alle 3 Timer gleichzeitig verkleinert und die Abtastrate auf dem FPGA erhöht. Die Timer sind so ge-

wählt, dass sie noch einen Sicherheitsabstand bei den verschiedenen Frequenzen aufweisen, bis es zu einem Absturz kommt.

Diese Optimierung erfolgte durch Timerverringerung eines einzelnen Timers bei einer festen Frequenz in kleinen Schritten bis es zu einem Absturz kam. Danach wurde die vorher funktionierende Timerkombination gewählt und einer der anderen Timer weiter verkleinert, bis es erneut zu einem Absturz kam. Dies wurde rekursiv durchgeführt, bis für die eingestellte Frequenz die minimale Timerstellung gefunden wurde. Nachdem die Timer für diese Frequenz minimiert sind können mit dieser Timerkombination auch höhere Abtastraten erreicht werden. Daher wird die Frequenz weiter erhöht bis es zu einem Absturz kommt und dann ebenfalls eine niedrigere funktionierende Frequenz gewählt. Diese Frequenz ist die neue Ausgangsfrequenz von der aus die aktuellen Timer entsprechend dem hier beschriebenen Vorgehen weiter verbessert werden können. Für die so ermittelten Timerstufen wird dann durch variieren der Frequenzen der entsprechende Frequenzbereich gewählt bei dem es mit einem Sicherheitsabstand in den folgenden Tests zu keinem Absturz kam.

Mit den so ermittelten Timern wurden mehrere Tests mit verschiedenen Frequenzen, durchgeführt und mittels simulierter Messdaten, welche über die kontinuierliche Verstellung der Eingangsspannungen an den verschiedenen Analogeingängen erzeugt wurde, ein Absturz provoziert. Hierbei wurde jeweils die Grenzfrequenz und mindestens eine weitere Frequenz getestet. Die Daten wurden zufällig mittels eines Spannungsgenerators an allen Eingängen erzeugt. Der Test dauerte 5 Minuten, da diese Zeit für eine Messung ausreichend ist. Diese Tests sind allerdings keine Garantie dafür, dass es im Messbetrieb zu keinem Absturz kommt.

Es wurden folgende Timerkombinationen ermittelt:

Tabelle 8-1: Timerstufen der Schleifen des Brokers

Stufe	HPL-Timer [ms]	LPL-Timer Up [ms]	LPL-Timer Down [ms]	Frequenzbereich [Hz]
1	100	800	1000	<500
2	10	100	100	500<f<2000
3	1	50	50	>2000

In der 3. Stufe wird der HPL-Timer bereits mit dem Minimum von 1ms getaktet, was dem maximalen Schleifentimer von 1000Hz entspricht.

8.6 Paketgrößenerhöhung

Die Paketgröße der Messgrößen in der HPL ist mitverantwortlich für den sich einstellenden Delay. Zu Beginn wurde eine maximale Paketgröße, welche der HPL-DatenArraygröße entspricht, von 500 gewählt, da die Verweilzeit eines einzelnen Arrays im RT-FIFO maßgebend den Delay beeinflusste. Nachdem die Timer optimiert wurden, stellte sich ein geringerer Delay durch die schnellere Abarbeitung der Schleifen des RT-Systems ein. Daher wurde die HPL-Daten-Arraygröße verdoppelt. Dadurch wird die Zeit die ein Array im RT-FIFO Puffer bleibt verdoppelt, sofern die Frequenz die gleiche bleibt und damit der Delay erhöht. Der Delay der sich nach der Paketgrößenerhöhung auf ca. 1 Sekunde (Schätzwert: kann nur geschätzt werden, da eine Synchronisation von Hostsystem und FPGA nicht möglich ist) beläuft und abhängig von der Frequenz und der aktuellen Arraygröße ist, ist klein genug wie sich in der Validierung gezeigt hat um die Messung durchführen zu können. Dabei ist die Fahrzeuggeschwindigkeit maßgebend, da diese schnell durch den Fahrer kompensiert wird.

Eine Verdopplung der Arraygröße korreliert mit einer Halbierung des Puffers und somit der Zeit bis zu einem Überlauf.

9 Verifikation

Die Begriffe Verifikation und Validierung sind bereits in Kapitel 1.3 eingeführt worden. In der Praxis beinhaltet ein Test beide Aspekte, wobei der Validierungsanteil mit steigender Teststufe zunimmt, vgl. Spillner²⁰. Dennoch werden in dieser Arbeit die Teststufen bis einschließlich des Systemtests der Verifikation und der Abnahmetest, der im Folgenden nicht als solcher bezeichnet wird da es keine Abnahme durch den Kunden gibt, der Validierung zugeordnet.

9.1 Testen

Fachbegriffe im Gebiet des Softwaretests werden häufig sehr unterschiedlich definiert und verwendet, was zu Missverständnissen führen kann. Die hier verwendeten Fachbegriffe des Softwaretests entsprechen der Verwendung und Definition von Spillner²¹.

Für den vollständigen Test der programmierten Software, welche den Namen PAMELA (PETRAs automatische Messdatenerfassungslogik und Auswertesoftware) erhält, wäre eine astronomische Anzahl von Testfällen erforderlich. Ein „vollständiges Austesten“, d.h. alle möglichen Parametervariationen zu testen, ist daher im Rahmen dieser Arbeit nicht möglich. Jeder Test ist deshalb nur eine Stichprobe, eine Fehlerfreiheit kann durch das Testen nicht nachgewiesen werden, vgl. Spillner²². Eine absolute Fehlerfreiheit für alle Anwendungsgebiete und –fälle kann nicht nachgewiesen werden, nur für die Testfälle der Verifikation kann unter den angegebenen Bedingungen die Fehlerfreiheit nachgewiesen werden

Bei jeder entwickelten Software ist selbst zu individuell zu entscheiden, wie intensiv und umfangreich sie zu testen ist. Die Entscheidung ist maßgebend vom Risiko bei fehlerhaftem Verhalten abhängig.

Ein Testfall besteht aus Eingabewert, Sollergebnis und der Angabe von Vor- und Nachbedingungen, unter denen der Test abläuft. Wird ein Testfall ausgeführt zeigt das Testobjekt ein Istverhalten. Die Sollwerte werden für jeden Testfall vor der Testdurchführung in der Testspezifikation bestimmt. Dabei wird der Sollwert von dem sogenannten Testorakel bestimmt, mögliche Testorakel sind Spezifikationen, Benutzerhandbuch, Diagramme und Modelle. Bei

²⁰ Spillner, Linz: Basiswissen Softwaretest, 2003, S.41

²¹ Spillner, Linz: Basiswissen Softwaretest, 2003

²² Spillner, Linz: Basiswissen Softwaretest, 2003, S.14 ff.

der Software PETRA werden in erster Linie die Spezifikationen als Testorakel herangezogen, aus denen die Sollwerte abgeleitet werden.

Die Tests lassen sich in folgende Tests unterteilen, Komponententest (= Modultest), Integrationstest, Systemtest und Abnahmetest, der im Projekt PETRA der Validierung entspricht.

9.2 Modultests

Die Modultestvorschriften, wurden direkt nach den Modultestspezifikationen geplant, wie es das Vorgehen des W-Modells vorsieht. Als Testorakel dienen die erarbeiteten Modulspezifikationen und Entwürfe. Die Modultestvorschriften wurden nach der Programmierung des entsprechenden Moduls detailliert. Direkt anschließend wurde das gerade fertiggestellte Testobjekt (Modul), entsprechend der aufgestellten Modultestvorschrift einem Modultest unterzogen. Die jeweiligen Testfälle sind in der Modultestvorschrift im Anhang E zu finden. Die Testfälle, geben an mit welchen Eingangswerten das Modul getestet wird und welche Sollergebnisse erwartet werden. Die entsprechenden Testprotokolle, die ebenfalls in Anhang E zu finden sind und die Testdurchführung dokumentieren, enthalten neben den Testfällen die tatsächlichen Ausgaben (= Istergebnisse).

9.2.1 Ziel des Modultests

Das Ziel des Modultests, auch Komponententest genannt, ist, zu verifizieren, dass das Modul sich so verhält wie es dessen Spezifikation vorsieht. Wichtigste Aufgabe ist nach Spillner²³ die Sicherstellung, dass das jeweilige Testobjekt die laut seiner Spezifikation geforderte Funktionalität korrekt und vollständig realisiert. Dabei werden die Fehlerwirkungen in der Programmierung der Module aufgedeckt und die Fehler können beseitigt werden. Ergebnis ist jeweils ein Modul das sich entsprechend der Spezifikation verhält, sobald die aufgedeckten Fehler beseitigt wurden.

9.2.2 Modultest: FPGA – Schräglaufeinstellung

Der Modultest Schräglaufeinstellung wurde entsprechend der Modultestvorschrift für die Schräglaufeinstellung, siehe Anhang E1.1.1., durchgeführt. Hierbei wurden die Eingaben „Endschalter positiv“, „Endschalter negativ“, „Toleranzbereich“, „Sollradlenkwinkel“ und „Radlenkwinkel“, entsprechend der Modultestvorschrift für die Schräglaufeinstellung, variiert und die erwarteten Ausgaben „Motor MR positiv“ und „Motor RR positiv“ mit den tatsächli-

²³ Spillner, Linz: Basiswissen Softwaretest, 2003, S.45

chen verglichen. Die Eingaben Sollschräglaufwinkel und der Schräglaufwinkeltoleranzbereich wurden vorher anhand der Kalibrierdaten in Integer umgerechnet und auf dem Frontpanel in die für den Test erzeugten Eingabelemente eingegeben. Der Radlenkwinkel am Messrad wurde über einen Spannungsgenerator am NI-9205 AI-Modul simuliert. Ist die Differenz zwischen Sollradlenkwinkel und Radlenkwinkel innerhalb des Toleranzbereichs, darf kein Motorausgang aktiv sein. Ist der Wert außerhalb des Toleranzbereichs wird der entsprechende Ausgang aktiv, es sei denn der Endschalter in dieser Richtung ist aktiv. Die Endschalter werden über Anlegen einer Spannung an das digitale Eingangsmodul simuliert. Die im Modultest getesteten Testfälle können im Anhang E1.1.2. eingesehen werden.

Bei der Durchführung dieses Tests wurde ein Fehler beim Vergleich mit den Grenzwinkeln ($\pm 10^\circ$), Testfälle 3 und 4 nicht bestanden, entdeckt und behoben, danach wurde der komplette Test erneut ausgeführt und bestanden. Somit ist das Schräglaufeinstellungsmodul verifiziert.

9.2.3 Modultest: FPGA – Schwimmwinkelkompensation

Der Modultest Schwimmwinkelkompensation wurde entsprechend der Modultestvorschrift für die Schwimmwinkelkompensation, siehe Anhang E1.2.1., durchgeführt. Die Eingaben vom Typ Boolean wurden direkt in der Software gesetzt, da zuvor die korrekten Erkennungen der digitalen Eingangsdaten überprüft wurden. Der Radlenkwinkel am Referenzrad und der Schwimmwinkeltoleranzbereich wurden anhand der Kalibrierdaten in Integer umgerechnet. Der Schräglaufwinkel und der Radlenkwinkel am Messrad wurden entsprechend der Kalibrierung über eine Spannungsquelle eingestellt und der Wert mit einem Voltmeter überprüft. Beide Werte wurden kontinuierlich verstellt. Ist die Differenz zwischen Schräglaufwinkel und Radlenkwinkel am Messrad innerhalb des Toleranzbereichs, darf kein Motorausgang aktiv sein. Ist der Wert außerhalb des Toleranzbereichs wird der entsprechende Ausgang aktiv, sofern nicht der entsprechende Endschalter in diese Richtung aktiv ist.

Der Modultest wurde mit den Testfällen die in Anhang E1.2.2. nachgelesen werden können durchgeführt und ohne Beanstandung bestanden. Somit ist das Schwimmwinkelkompensationsmodul verifiziert.

9.2.4 Modultest: FPGA – Drehzahlmodul

Der Modultest der Drehzahlberechnung wurde auf zwei Arten, entsprechend der Modultestvorschrift der Drehzahlberechnung, siehe Anhang E1.3.1., durchgeführt:

1. Am entsprechenden Eingabefeld des programmierten Rechtecksignalgenerators, wurde der Timer für den Schleifendurchlauf eingestellt und die berechnete Drehzahl im entsprechenden Ausgabeelement Drehzahl abgelesen .

2. Es wurde der Funktionsgenerator PM 5127 von Philips an den digitalen Eingang des Digitalmoduls NI-9401 angeschlossen. Bei diesem Test ist zu beachten, dass der Funktionsgenerator einen Fehler von 10% aufweisen kann. Daher sind die erwarteten Ausgaben mit entsprechendem Fehler als gültig zu sehen. Die Frequenzen wurden entsprechend der Testfälle eingestellt und die Ausgabe überprüft.

Da das Drehzahlmodul beide Modultests ohne Fehler bestanden hat, ist das Drehzahlmodul verifiziert.

9.2.5 Modultest: FPGA – Low Priority Loop

Das Modul Low Priority Loop wurde geprüft, indem die Messdaten auf dem Realsystem aus dem DMA-Speicher gelesen, angezeigt und mit den erwarteten Ausgabewerten verglichen wurden. Des Weiteren wurden Werte des Proportionalventils und der Nullkalibrierung auf dem RTS eingeben. Der eingestellte Wert des Proportionalventils wurde mit einem Voltmeter am entsprechenden Ausgang und der Wert der Nullkalibrierung wurde durch die entsprechende LED am digitalen Ausgangsmodul überprüft.

Der maximale Fehler des Analogausgangsmoduls NI-9263 beträgt 0,75% des maximalen Bereichs plus 0,35% Lesefehler, vgl. ²⁴. Die Messgenauigkeit des Analogmoduls NI-9205 wird mit 1% auf den eingestellten Bereich angegeben, siehe ²⁵. Das zur Verfügung stehende Voltmeter hat eine Messgenauigkeit von 2%, wie eine Vergleichsmessung mit mehreren kalibrierten Messgeräten ergab. Somit ist ein maximaler Gesamtfehler von 3% beim Eingangsmodul zu erwarten und 3,1% beim Ausgangsmodul. Der Modultest wurde entsprechend der Modultestspezifikation, Anhang E1.4.1., durchgeführt und die tatsächlichen Ausgaben können in Anhang E1.4.2. eingesehen werden.

Da alle Testfälle positiv abgearbeitet werden konnten, da alle Istwerte den Sollwerten entsprachen, ist das Modul Low Priority Loop verifiziert.

9.2.6 Modultest: FPGA – High Priority Loop

Das Modul High Priority Loop wurde geprüft, indem die Messdaten auf dem Realsystem aus dem DMA-Speicher gelesen und angezeigt wurden, wie im Modultest der Low Priority Loop. Die Messwerte wurden simuliert, indem der zu testende Kanal mit einer Spannung versehen wurde. Die Spannung wurde mit dem Voltmeter überprüft und mit dem auf dem Realsystem in Volt umgerechneten Messwert verglichen. Dabei ist ein Fehler von 3% zuläs-

²⁴ National Instruments: Bedienungsanleitung NI-9263, 2007, S. 13

²⁵ National Instruments: Bedienungsanleitung NI-9205, 2007, S. 22

sig, wie bereits im vorherigen Kapitel für das NI-9205 Modul angegeben wurde. Der Modultest wurde entsprechend der Modultestspezifikation, Anhang E1.5.1., durchgeführt und die tatsächlichen Ausgaben können in Anhang E1.5.2. eingesehen werden.

Das Modul Low Priority Loop ist verifiziert, da der Test ohne Beanstandung bestanden wurde.

9.2.7 Modultest: Broker – NPL + TCL

Das Modul NPL baut nur die TCP-Verbindung auf und diese ist nicht ohne Gegenstück testbar. Das Modul TCL hat die einzige Aufgabe die Daten zwischen dem DMA-Speicher und dem TCP (FIFO) Puffer hin und her zu kopieren. Aus diesem Grund werden die beiden Module nicht einzeln getestet sondern nur im Rahmen des Integrationstests mit den darunter liegenden, zuvor integrierten FPGA-Modulen.

9.2.8 Modultest: Host – Hauptprogramm

Das Modul Hauptprogramm hat die Hauptaufgabe zwischen den verschiedenen Submodulen wechseln zu können. Das Hauptprogramm wird erst getestet, wenn die darunter liegenden Submodule integriert sind, daher wird das Modul nicht einzeln getestet sondern im Rahmen des Integrationstests.

9.2.9 Modultest: Host – Einstellungen

Das Modul Einstellungen wurde entsprechend der Modultestvorschrift, siehe Anhang E2.1.1., getestet, indem in die Eingabefelder verschiedene Werte eingegeben und diese mit denen in der erstellten Konfigurationsdatei, bzw. in den entsprechenden Variablen, die für diesen Test ausgegeben wurden, verglichen wurden. Die Testfälle und die eingetretenen Ausgaben sind den Tabellen im Anhang E2.1.2. zu entnehmen.

Da bei allen Testfällen die erwarteten Ausgaben mit den tatsächlichen Ausgaben übereinstimmen ist das Modul verifiziert.

9.2.10 Modultest: Host – Kalibrierung

Das Modul Kalibrierung wird nicht in einem Modultest getestet, weil es einfacher ist das Modul mit einem komplett integrierten FPGA und Broker zu testen. Das Modul wird daher erst im Integrationstest bzw. im Systemtest überprüft.

9.2.11 Modultest: Host – R_{dyn} -Berechnung

Das Modul R_{dyn} -Berechnung hat die Eingänge Messfelgennutzung, Schräglaufwinkeltoleranz, Geschwindigkeitstoleranz und Messdatenarray. Die weiteren benötigten Werte werden aus dem Messdatenarray extrahiert. Zum Testen wird statt mehreren Arrays mit über 100 Elementen nur ein Array mit 3 Elementen verwendet, da es ein großer Aufwand ist Arrays der Größe 100 mit Testdaten zu erstellen. D.h. bei diesem Test führt die R_{dyn} -Berechnung nur die eine Mittelwertbildung aus 3 Elementen durch. Da die R_{dyn} -Berechnung automatisch die Arraygröße erkennt ist keine Änderung des Moduls nötig.

Der Modultest wurde entsprechend der Modulspezifikation, Anhang E1.3.1., durchgeführt. Die tatsächlichen Ausgabewerte und die Testfälle können dem Anhang E1.3.2. entnommen werden. Da ausführliche Tests im Systemtest folgen, wurden nicht alle möglichen Permutationen der Testdaten in diesem Modultest vorgenommen.

Der Modultest wurde ohne Beanstandung bestanden, somit ist das R_{dyn} -Berechnungsmodul verifiziert.

9.2.12 Modultest: Host – Messung

Das Modul Messung hat die Eingänge Sollschräglaufwinkel, Schräglaufwinkeltoleranz, Geschwindigkeitstoleranz und Messdatenarray. Es wird getestet, indem die Daten eingegeben werden, auch das Messdatenarray wird auf dem Testpanel eingegeben. Um den Testaufwand zu verringern wurde das Messdateneingangsarray auf 3 Elemente reduziert.

Der Modultest wurde entsprechend der Modulspezifikation, Anhang E1.4.1., durchgeführt. Die tatsächlichen Ausgabewerte und die Testfälle können dem Anhang E1.4.2. entnommen werden. Da ausführliche Tests im Systemtest folgen, wurden nicht alle möglichen Permutationen der Testdaten in diesem Modultest vorgenommen.

Der Modultest wurde ohne Beanstandung bestanden, somit ist das Modul Messung verifiziert.

9.2.13 Modultest: Host – Statusanzeigen

In dem Modul Statusanzeigen, geht es in erster Linie darum, dass die empfangenen Kontrollgrößen richtig umgerechnet und angezeigt werden. Daher werden zum Testen die Daten auf dem Testpanel in Volt eingegeben und diese vom Modul entsprechend der Kalibrierdaten umgerechnet und ausgegeben. Die Ausgaben werden mit den Werten die in der Modulspezifikation, vgl. Anhang E1.5.1., angegeben sind verglichen. Da zum Zeitpunkt des Modultests noch nicht alle Kalibrierdaten vorhanden waren, wurden die nicht vorhandenen Kalibrierdaten

durch Offset = 0 und Faktor = 1 ersetzt. Somit entsprechen diese Ausgabedaten den Eingabedaten in Volt.

Da im Modultest, siehe Anhang E1.5.2., die Soll- und Ista Ausgaben übereinstimmen ist dieser Test bestanden.

9.2.14 Modultest: Host – Warnung

Dieser Modultest hat die Aufgabe die Benutzerwarnungen und das Herunterfahren bei Über- bzw. Unterschreiten bestimmter Grenzen zu simulieren. Hierzu werden die entsprechenden Daten direkt auf dem Testpanel eingegeben und die Ausgabe, d.h. die Warnung bzw. die Aktion des Herunterfahrens der Software abgewartet. Entspricht diese der erwarteten Ausgabe die in der Modulspezifikation Warnung, Anhang E1.6.1., festgelegt wurde so ist der entsprechende Testfall bestanden.

In Anhang E1.6.2. können die Testfälle und eingetretenen Ausgaben eingesehen werden. Der Modultest wurde nach einer Nachbesserung, da ein Fehler beim Herunterfahren entdeckt wurde, bestanden. Somit ist das Modul Warnung verifiziert.

9.3 Integrationstests

9.3.1 Integrationsmethodik

9.3.1.1 Begriffserklärung und Aufzählung der Integrationsmethoden:

Definition Platzhalter: Ist ein Pseudomodul das die Funktionalität eines noch nicht entwickelten Moduls emuliert.

Definition Testtreiber: Ein Treiber ist ein Modul welches ein zu testendes Modul aufruft und mit Eingabedaten versorgt.

Nach Spillner²⁶ gibt es folgende Grundstrategien für das Vorgehen bei Integrationstests:

- **Top-Down-Integration**

Der Test beginnt mit der obersten Komponente des Systems, die weitere Komponenten aufruft. Die untergeordneten Komponenten sind dabei durch Platzhalter zu ersetzen. Sukzessive

²⁶ Spillner, Linz: Basiswissen Softwaretest, 2003, S.54 ff.

werden die Komponenten niedrigerer Systemschichten hinzu integriert. Die getestete höhere Schicht dient dabei jeweils als Testtreiber.

- **Bottom-Up-Integration**

Der Test beginnt mit den elementaren Komponenten des Systems, die keine weiteren Komponenten aufrufen. Größere Teilsysteme werden sukzessive aus getesteten Komponenten zusammengesetzt, mit anschließendem Test dieser Integration.

- **Ad-Hoc-Integration**

Die Bausteine werden in der Reihenfolge ihrer Fertigstellung integriert. Sobald eine Komponente ihren Komponententest absolviert hat, wird geprüft, ob sie zu einer anderen schon vorhandenen und bereits getesteten Komponente oder zu einem teilintegrierten Subsystem passt. Wenn ja, werden beide Teile integriert und der Integrationstest zwischen beiden wird durchgeführt.

- **Backbone-Integration**

Es wird ein Programmskelett erstellt in das bei der Integration schrittweise die einzelnen Module eingefügt werden.

- **Nicht inkrementelle Integration – Big-Bang-Integration**

Es wird hierbei mit der Integration gewartet, bis alle Softwarebauteile entwickelt sind und dann werden alle auf einmal integriert.

9.3.1.2 Bewertung der Integrationsverfahren:

Integrationsmethode	Vorteil	Nachteil
Top-Down	– Da bereits getestete, übergeordnete Komponenten den wesentlichen Teil der Ablaufumgebung bilden, werden keine bzw. nur einfache Testtreiber benötigt.	– Untergeordnete, noch nicht integrierte Komponenten werden durch Platzhalter ersetzt, was sehr aufwändig sein kann.
Bottom-Up	– Es werden keine Platzhalter benötigt.	– Übergeordnete Komponenten werden durch Testtreiber simuliert.
Ad-Hoc	– Zeitgewinn, da jeder Baustein frühestmöglich in seine passende Umgebung integriert wird.	– Es werden sowohl Platzhalter als auch Testtreiber benötigt.
Backbone	Module können in beliebiger Rei-	Dies erfordert, dass ein Programmske-

	henfolge integriert werden.	lett zuvor erstellt wird.
Big-Bang	<ul style="list-style-type: none"> – Zeitvorteil wenn der Test bestanden wird, d.h. es sind nur wenige einfach zu behebbende Fehlerzustände aufgetreten. – Nur bei sehr kleinen Softwareprojekten machbar. 	<ul style="list-style-type: none"> – Die Wartezeit bis zum Big-Bang ist verlorene Testdurchführungszeit. – Alle Fehlerwirkungen treten geballt auf. – Die Lokalisierung und Behebung von Fehlerzuständen gestaltet sich schwierig und zeitraubend.

	Bewertungskriterien		
Integrationsmethode	1. Zusatzaufwand zum Testen	2. Fehlerentdeckung	3. Aufwand zur Fehlerlokalisierung
Top-Down	– (nur Platzhalter)	+	++
Bottom-Up	O (nur Testtreiber)	+	++
Ad-Hoc	-- (Platzhalter + Testtreiber)	O	+
Backbone	--	O	–
Big-Bang	++	–	--

9.3.1.3 Auswahl des Integrationsverfahrens:

Die entscheidenden Bewertungskriterien sind:

- Bewertungskriterium 1: wenig Zusatzaufwand beim Integrationstest
- Bewertungskriterium 2: Wahrscheinlichkeit Fehlerwirkungen aufzudecken
- Bewertungskriterium 3: strukturiertes Vorgehen in kleinen Schritten um den Aufwand der Fehlerlokalisierung zu reduzieren

Für die Software des Projekts PETRA wird das Vorgehen Bottom-Up verwendet, da hierbei keine Platzhalter nötig sind und die übergeordneten Testtreiber einfach zu realisieren sind. Wohingegen das Erstellen von Platzhaltern für die untergeordneten Komponenten, wie es beim Top-Down nötig ist, sehr aufwändig wird. Das Erstellen eines Backbones wird durch die verteilte Architektur sehr aufwändig. Die Ad-Hoc Integration bietet nur einen Vorteil, wenn die Software im Team entwickelt wird und die Big-Bang Integration hat keine Vorteile, weswegen beide Varianten für dieses Projekt ungeeignet erscheinen.

Aus der Bottom-Up Integration ergibt sich, dass erst die FPGA-Module, danach die Broker-Module und zum Schluss die Module des Hosts integriert werden.

Innerhalb des FPGAs bietet sich folgende Mischintegration an. Als erstes werden HPL und LPL zu PL integriert, da diese beiden Module die einzige Schnittstelle zum Broker haben, ist die Kommunikation zum Broker als erstes herzustellen. Danach wird die Konfiguration dazu integriert, damit die Eingabewerte auf dem Broker zum Testen in Volt umgerechnet werden können. Anschließend werden die Module Drehzahlberechnung, Schräglaufeinstellung und Schwimmwinkelkompensation Schritt für Schritt integriert, dabei ist die Reihenfolge nicht entscheidend da die Module aus Sicht der Software unabhängig voneinander sind.

Die Module TCL und HPL werden zum Modul Broker integriert.

Bei der Integration der Host-Module bietet sich folgendes Vorgehen an. Da die Module Einstellungen und Kalibrierung unabhängig voneinander, aber beide Daten vom Host an den Broker senden werden diese zunächst zusammen mit der Netzwerkkommunikation, zu dem Host-Send-Modul, integriert. Danach wird das R_{dyn} -Berechnungsmodul hinzu integriert. In einem nächsten Schritt wird hierzu das Messungsmodul integriert. Die Module Warnung und Status werden separat integriert. Im letzten Schritt werden die integrierten Module und das Modul Hauptprogramm zum Hostmodul integriert.

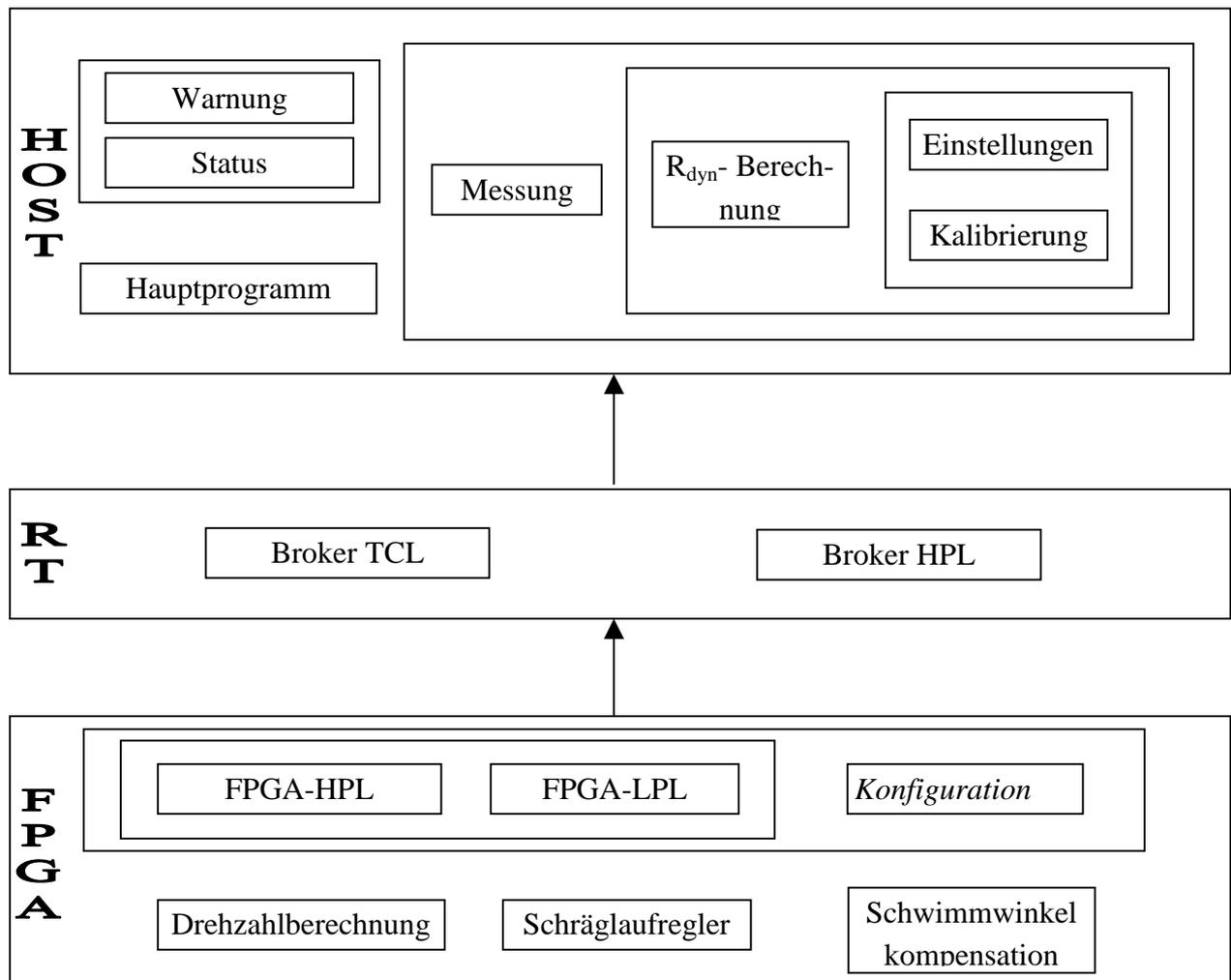


Abbildung 9.1: Integrationsvorgehen

9.3.2 Ziel des Integrationstests

Im Integrationstest wird geprüft ob die integrierten Module richtig kommunizieren und das entstandene integrierte Modul der Spezifikation entspricht. Hauptaugenmerk bei den Integrationstests liegt auf den Schnittstellen.

Ergebnis eines jeden Integrationstest ist ein verifiziertes, zusammengesetztes Modul.

9.3.3 Integrationstest: FPGA - Priority Loops

Zunächst wurden die FPGA-Module HPL und LPL integriert. Um die Funktionalität des integrierten Moduls entsprechend dem geplanten Einsatz zu testen, wurde gleich das Konfigurationsmodul hinzuintegriert. Das neu entstandene PL-Modul wurde entsprechend der PL-

Integrationstestvorschrift, siehe Anhang F1.1., getestet. Deren Testfälle bestehen aus der Konjunktion der Testfälle der Modultests LPL und HPL. Da LPL und HPL parallel zueinander laufen, war keine Wechselwirkung zu erwarten. Es trat bei dem durchgeführten PL-Integrationstest keine Fehlerwirkung auf.

9.3.4 Integrationstest: FPGA

Dieser Test dient dazu alle zusammengeführten FPGA-Module in einem integrierten FPGA-Modul zu testen. Das PL-Modul wurde zunächst um das Modul Drehzahlberechnung erweitert. Bei dem entstandenen Modul wurde nur geprüft ob ein erzeugtes Drehzahlsignal im HPL-Modul erkannt wird. Danach wurden Schräglaufeinstellung und Schwimmwinkelkompensation in dieser Reihenfolge integriert. Als Integrationstest für das FPGA-Modul wurden die Testfälle der Modultests LPL, HPL, Schräglaufwinkel und Schwimmwinkel erneut getestet, allerdings nur ein Teil der Testfälle wie im Anhang F2. zu erkennen ist. Während des Tests wurde sowohl sporadisch als auch kontinuierlich am angeschlossenen Inkrementalgeber gedreht um Wechselwirkungen ausschließen zu können. Da der Test ohne Fehlverhalten bestanden wurde gilt das integrierte FPGA-Modul als verifiziert.

9.3.5 Integrationstest: Broker

Das Broker Modul entsteht durch die Integration der Module NPL und TCL. Vorbedingung dieses Tests ist, dass das FPGA-Modul integriert und getestet ist. Es wird geprüft, indem zum Einen die Messdaten in einem Testtreiber auf dem Hostsystem aus der TCP/IP-Verbindung gelesen, umgerechnet und angezeigt werden, zum Anderen die benötigten Daten im Testtreiber eingegeben werden. Die Drehzahl wird geprüft, indem am Inkrementalgeber gedreht wird. Die restlichen Testfälle bestehen aus einer Untermenge der Testfälle aus dem Integrationstest des PL-Moduls, da keine weitere Funktionalität außer der Weiterleitung zum Hostsystem durch den Broker hinzugefügt wird. Dabei darf kein Pufferüberlauf, weder am DMA-Kanaleingang noch an den RT-FIFO-Puffern, auftreten.

Das Brokermodul wurde entsprechend der Modultestvorschrift, Anhang F3.1., des Broker-Moduls getestet. Hierbei wurde die HPL-Schleife des FPGAs mit 100Hz und die LPL-Schleife des FPGAs mit 1Hz getaktet. Da zu diesem Zeitpunkt noch keine Timeroptimierung vorgenommen wurde konnte das Modul nur mit der zu dieser Zeit erreichbaren Frequenz von 100Hz getestet werden. Bei der Durchführung des Tests wurden mehrere Fehlerwirkungen entdeckt, welche durch Beseitigung eines Kommunikationsfehlers im Fehlernachtest nicht mehr auftraten. Damit gilt das Brokermodul als verifiziert.

9.3.6 Integrationstest: Host

Die Module des Host wurden wie in Kapitel 9.3.1.3 beschrieben integriert. Die entsprechenden Integrationstests können im Anhang F4. – F7. eingesehen werden. Diese Integrationstests wurden erfolgreich beendet. Das integrierte Host-Modul welches im letzten Integrationsschritt entstand wurde ohne Testspezifikation auf augenscheinliche Funktionalität getestet, da durch diese Integration keine Probleme zu erwarten waren. Zudem wird im folgenden Systemtest das Gesamtsystem getestet.

9.4 Systemtest

Vor dem Systemtest werden die programmierte Software mit der Elektronik in dem Messanhänger integriert. Dazu wurde das CRIO in die Messbox integriert und verdrahtet. Die Verbindung zum Laptop erfolgt über ein Netzkabel, welches aus der Messbox heraus am Rahmen des Messanhängers entlang über den Kofferraum in das Zugfahrzeug hinein geführt wird. Bevor der Systemtest durchgeführt wurde, wurde die Elektronik getestet.

9.4.1 Ziel des Systemtests

Ziel des Systemtests ist die Verifikation des integrierten Systems, dabei wird geprüft ob das System sich entsprechend seiner Spezifikation verhält.

Ergebnis ist ein verifiziertes System. Das heißt allerdings nicht, dass das System seinen Einsatzzweck erfüllt.

9.4.2 Methodik des Systemtests

Die Verifikation durch den Systemtest wird in 2 Teile aufgeteilt, da aus Sicherheitsgründen erst gefahren wird wenn die Grundfunktionen im Stand nachgewiesen sind. Zudem wird im Stand die Schnittstelle zwischen Software und Elektronik getestet. D.h. es wird geprüft ob alle Verbindungen richtig angeschlossen sind und die Elektronik richtig agiert. Vor dem Systemtest während der Fahrt findet ein Systemtest im Stand statt. Die Systemverifikationsvorschriften wurden zeitlich gesehen, wie bei der Entwicklung nach dem W-Modell vorgesehen, direkt nach dem Grobentwurf erstellt und im Laufe des Entwicklungsprozesses detailliert.

9.4.3 Fehlerbetrachtung der Messtechnik

Durch in der Motoransteuerungselektronik verwendete Pulsweitenmodulation entsteht ein Rauschen, welches auf allen Kanälen gemessen werden kann. Dieses Rauschen wurde vom Elektroniker gemessen. Der maximale Fehler durch dieses Rauschen beträgt 10mV, wenn die Motoransteuerung nicht aktiv ist. Ist die Motorsteuerung aktiv beträgt der maximale Fehler durch dieses Rauschen 100mV. Da der Multiplexer des verwendeten analogen Eingangsmoduls NI-9205 die Messwerte nicht parallel sondern sequentiell erfasst, kann sich dieser maximale Fehler abhängig von der Abtastrate in den erfassten Messdaten wiederfinden. Dies betrifft besonders die „echten“ Messdaten, da diese alle im Bereich $\pm 5V$ erfasst werden.

Das heißt, alleine aus der Messtechnik kann sich ein maximaler Fehler von 0,2% bei inaktiver Motoransteuerung und 2% bei aktiver Motoransteuerung ergeben. Da während der Messung die Schwimmwinkelregelung nachregelt, ist der maximale Fehler von 2% anzunehmen.

9.4.4 Vorbedingungen des Systemtests „Stand“

Bevor der Systemtest im Stand durchgeführt wird, sind bestimmte Vorbedingungen zu erfüllen, welche durch Vorarbeiten zu erledigen sind. Neben den Vorbedingungen, dass die Integrationstests und die Elektroniktests abgeschlossen sind, sind folgende Aufgaben erledigt.

Sturzwinkel einstellen:

Der Sturzwinkel des Messrades ist im eingefederten Zustand auf 0 Grad einzustellen. Dies wird durch Verstellung des Sturzes „von Hand“ erreicht. Hierzu wird der entsprechende Motor über den Motorschalter in der Messbox angesteuert, die Überprüfung des Sturzwinkels erfolgt mit der Wasserwaage.

Radlenkwinkelkalibrierung:

Die Radlenkwinkel wurden mit einem Kalibrierverfahren unter Verwendung zweier Laser kalibriert. Die genaue Durchführung und das Ergebnis der Radlenkwinkelkalibrierung kann in der Diplomarbeit 413/06 nachgelesen werden. Die Kalibrierdaten wurden in der Software als Standardwerte abgelegt.

Messbereitschaft herstellen:

Die Messbereitschaft des Messanhängers PETRA wird hergestellt indem der Correvitsensor angebaut, die Messfelge montiert, alle Auswerteeinheiten eingebaut und angeschlossen werden.

9.4.5 Systemtest „Stand“

Der Systemtest im Stand wurde entsprechend der Systemverifikationsvorschrift des Systemtests „Stand“, die im Anhang G1. eingesehen werden kann durchgeführt. Das zugehörige Systemtestprotokoll befindet sich ebenfalls im Anhang unter G2..

1. Die Kontrollgrößen Bordspannung Öldruck und Öltemperatur wurden auf Plausibilität geprüft. Dieser Teilttest wurde bestanden.
2. Prüfung der Messfelgenwerte und die Zerokalibrierung der Messfelge. Zunächst wurde in unbelastetem Zustand nullkalibriert und die Messdaten wurden in aufgebocktem Zustand des Messrades erfasst. Die erfassten Messdatenwerte entsprachen den erwarteten Werten. Danach wurde die Messfelge auf den Boden abgelassen. Die erwarteten Werte $F_x = 0\text{N}$; $F_y \neq 0\text{N}$ (Verzwängung beim Ablassen); $F_z = \text{ca. } 5000\text{N}$ (statische Radlast) entsprachen den tatsächlichen Werten $F_x = \text{ca. } 40\text{N}$; $F_y = \text{ca. } 860\text{N}$; $F_z = \text{ca. } 4900\text{N}$. Danach wurde der Sturzwinkel des Messrades erhöht, dabei wurde wie erwartet F_y größer. Somit ist dieser Test bestanden
3. Die Kraftmessdose wurde getestet, indem das Getriebe an dem sich die Kraftmessdose abstützt, mit dem Körpergewicht belastet wurde. Das F_x der Messdose wurde kleiner, somit ist der Test bestanden.
4. Die Potentiometer der Radlenkwinkelverstellung und dessen Endschalter wurden wie folgt getestet. Radlenkwinkel wurden mittels Handschalter verfahren. Die Anzeigen der Software verhielten sich wie entsprechend der Bewegungsrichtung der Radlenkwinkelverstellungen angenommen wurde, d.h. die Schnittstelle zwischen Software und Hardware (der Motoransteuerung und der Radlenkwinkelmessung) wurde beiderseits richtig implementiert. Alle Endschalter wurden angefahren und lösten das Signal in der Software aus, da zuvor die Grenzwinkel der Software auf höhere Werte gestellt wurden. Danach wurden die Grenzwinkel wieder in der Software eingestellt und erneut die Grenzen angefahren. Die Software schaltete die Motoransteuerung wie erwartet vor Erreichen der Endschalter ab. Dieser Test wurde ohne Beanstandung bestanden.
5. Der Drehzahlsensor wurde wie folgt getestet. Das Messrad wurde in Fahrtrichtung gedreht, es wurde kein Drehzahlsignal vom System erkannt. Das Messrad wurde gegen die Fahrtrichtung gedreht, es wurde ein Drehzahlsignal angezeigt. Die Fehlersuche ergab, dass der Drehzahlsensor in der Elektronik falsch angeschlossen war. Nach Behebung dieses Fehlers und Durchführung des Fehlernachttests wurde dieser bestanden.
6. Die Schräglaufeinstellung und die Schwimmwinkelkompensation und damit die Schnittstelle zur Motoransteuerung wurden getestet, indem das Schräglaufwinkelsignal auf Masse gelegt wurde. Nacheinander wurden die Winkel 0° , 7° und -7° in der

Software eingestellt. Diese wurden angefahren und die Schwimmwinkelregelung verhielt sich auch wie angenommen. Dieser Test wurde ohne Beanstandung bestanden.

9.4.6 Vorbedingungen des Systemtests „Fahrt“

Bevor der Systemtest während der Fahrt durchgeführt wird ist es erforderlich, dass folgende Vorbedingungen erfüllt sind: Der Systemtest „Stand“ ist erfolgreich beendet und der Correvitensor kalibriert und die Messbereitschaft des Anhängers hergestellt.

9.4.6.1 Correvitkalibrierung

Für die Correvitkalibrierung wurde eine Lichtschranke von Sick verwendet, welche an die Correvit-V2-Auswerteeinheit angeschlossen wurde. Die Lichtschrankenkalibrierung erfolgte über die Software „CeCalWin Pro 1.2“. Für diese Kalibrierung wurde ein älterer Laptop mit serieller Schnittstelle verwendet, da die Auswertebox das Triggersignal der Lichtschranke auf PIN 8, der seriellen Schnittstelle zum Laptop durchschleust, dieses Kommunikationssignal aber nicht von Usb-to-Serial-Adaptern erkannt wird. Zu beachten ist, dass die Werkkalibrierung des Correvitsensors abgelaufen ist und daher der Angegeben Fehler von 0,5% in Frage gestellt werden kann.

Die Kalibrierung wurde auf einer 200m langen geraden Strecke mit gleichmäßiger Fahrbahntextur durchgeführt auf der am Anfang und am Ende eine Lichtschrankenreflektorbarke aufgestellt wurde, die den Lichtstrahl der Lichtschranke reflektiert. Die Kalibrierung erfolgte bei 30km/h und bei 60km/h, um eine mögliche Geschwindigkeitsabhängigkeit zu erkennen, dabei wird die Geschwindigkeit möglichst konstant gehalten. In der Software wird eingestellt, ob der Geschwindigkeitsfaktor ($Faktor_v$) oder der Winkeloffset ($Offset_\alpha$) ermittelt wird.

Berechnung des Geschwindigkeitsfaktors:

- Die Strecke $s = 200\text{m}$ wird vorgegeben
- Die Zeit t wird von der Software ermittelt
- Die Geschwindigkeit wird konstant gehalten
- Das Programm berechnet den $Faktor_v$ wie folgt:

$$v_{\text{gemessen}} = \frac{1}{n} \sum_{i=1}^n v_i; v_{\text{berechnet}} = \frac{s}{t}; Faktor_v = \frac{v_{\text{berechnet}}}{v_{\text{gemessen}}} \quad (9.1)$$

Fehlerbetrachtung des Geschwindigkeitsfaktors:

Streckenfehler = 0,10m (Schätzwert, ungenaue Messung der Strecke und ungenaue Positionierung der Reflektoren) bei $s = 200\text{m}$ 0,05 %

Zeitfehler = 100ms (Lichtschrankenfehler) bei $v_{Fzg} = 30\text{km/h}$ 0,83m 0,41%

bei $v_{Fzg} = 60\text{km/h}$ 1,66m 0,82%

Gesamtfehler(30) = 0,46% Gesamtfehler(60) = 0,87%

Auswertung und Bestimmung des Geschwindigkeitsfaktors:

Die folgenden 2 Abbildungen zeigen die ermittelten Geschwindigkeitsfaktoren bei $v = 30\text{km/h}$ und $v = 60\text{km/h}$. In die Abbildungen wurde jeweils der Mittelwert, die Standardabweichung und der berechnete Fehlerbereich eingezeichnet. Da beide Mittelwerte $\text{Faktor}_v(30) = 0,9957$ und $\text{Faktor}_v(60) = 0,9946$ sich nicht signifikant unterscheiden, wurden die Messpunkte zusammen betrachtet um das Kollektiv zu erhöhen. Mittels einem Wahrscheinlichkeitsnetz wurde eine Normalverteilung des Kollektivs bestätigt. Der zu verwendende Faktor_v wird durch die Mittelwertbildung über alle Punkte ermittelt und beträgt **Faktor_v = 0,9951**. Dieser Faktor_v wird in der Software als Correvitgeschwindigkeitsfaktor im Menü Kalibrierdaten als Standardwert hinterlegt.

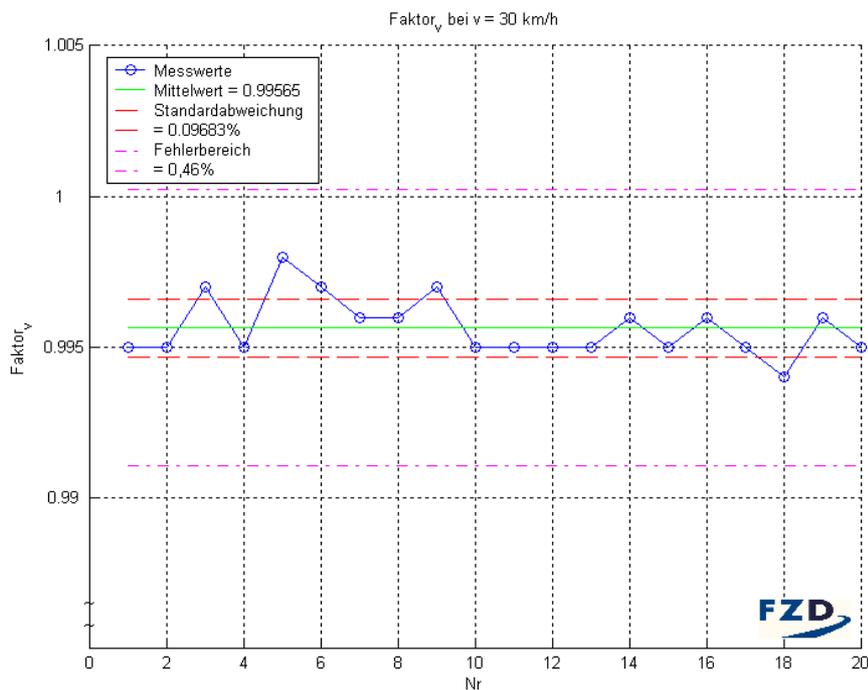


Abbildung 9.2: Faktor_v, 30km/h

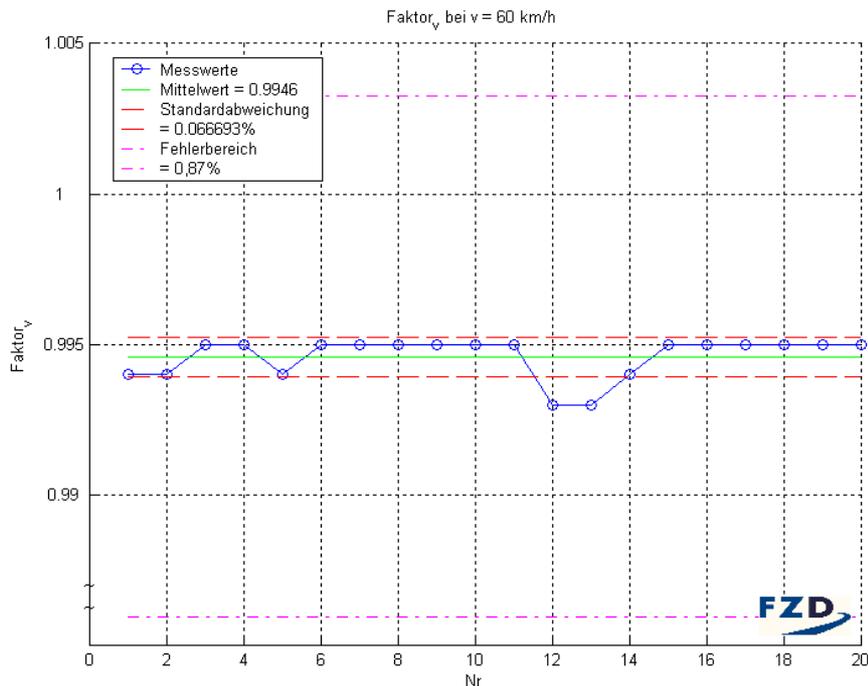


Abbildung 9.3: Faktor_v, 60km/h

Berechnung des Winkeloffsets:

Die Strecke $s = 200\text{m}$

Die Strecke wird möglichst gerade abgefahren

Das Programm berechnet den Offset_α wie folgt:

$$\alpha_{\text{gemessen}} = \frac{1}{n} \sum_{i=1}^n \alpha_i; \alpha_{\text{soll}} = 0; \text{Offset}_\alpha = \alpha_{\text{gemessen}} - \alpha_{\text{soll}} \quad (9.2)$$

Fehlerbetrachtung des Offset_α:

Versatz auf 200 m = 0,5 m $\arcsin 0,5/200 = 0,14^\circ$

Gesamtfehler = 0,14°

Auswertung und Bestimmung des Schräglaufwinkeloffsets:

Die folgenden 2 Abbildungen zeigen die ermittelten Schräglaufwinkeloffsets bei $v = 30\text{km/h}$ und $v = 60\text{km/h}$. In die Abbildungen wurde jeweils der Mittelwert, die Standardabweichung und der berechnete Fehlerbereich eingezeichnet. Da wie erwartet die Geschwindigkeit keinen Einfluss auf den Schräglaufwinkeloffset hat, wurden die Messpunkte zusammen betrachtet um das Kollektiv zu erhöhen. Mittels einem Wahrscheinlichkeitsnetz wurde eine Normalver-

teilung des Kollektivs bestätigt. Der zu verwendende Offset_α wird durch die Mittelwertbildung über alle Punkte ermittelt und beträgt **Offset_α = 0,697°**. Dieser Offset_α wird in der Software als Correvitschräglaufoffset im Menü Kalibrierdaten als Standardwert hinterlegt.

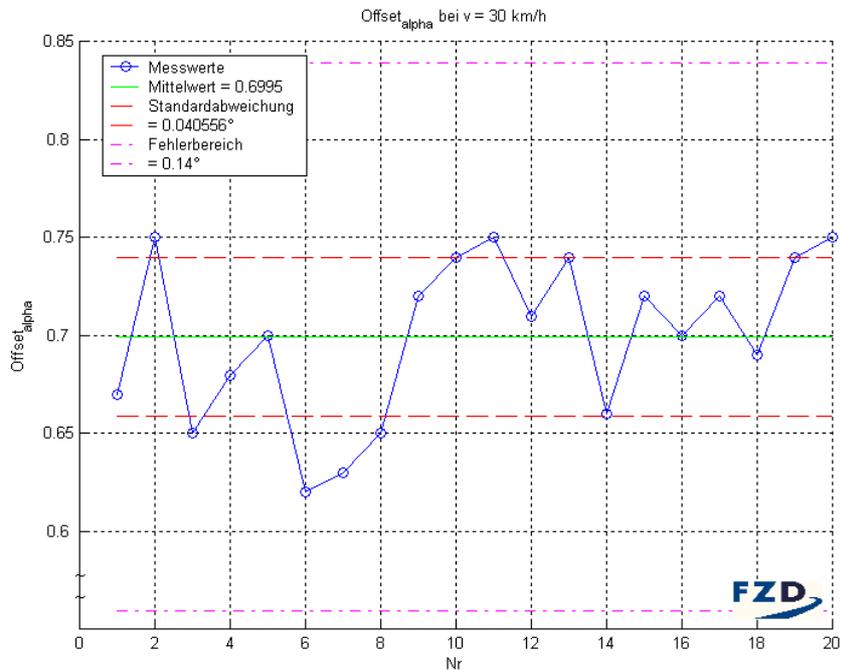


Abbildung 9.4: Offset_α 30 km/h

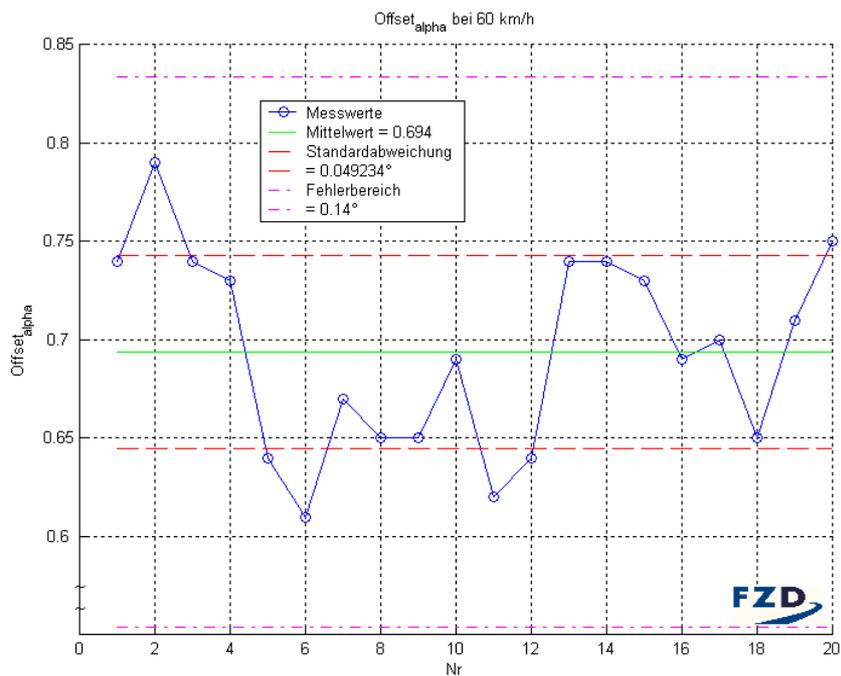


Abbildung 9.5: Offset_α 60 km/h

9.4.7 Systemtest „Fahrt“

Verifikation des Systems erfolgt durch den Systemtest während der Fahrt. Der Systemtest während der Fahrt wurde entsprechend der Systemverifikationsvorschrift des Systemtests „Fahrt“, die im Anhang G4. eingesehen werden kann durchgeführt. Das zugehörige Systemtestprotokoll befindet sich ebenfalls im Anhang unter G5..

1. Es wurde ein Smoke-Test, vgl. ²⁷, durchgeführt, indem wahllos Eingaben getätigt und alle Hauptfunktionen ausgelöst wurden. Beim Smoke-Test wird das System mit zufälligen Eingaben versehen um das System zum „Abrauchen“ zu bringen. Tritt kein offensichtliches Fehlverhalten auf, ist der Smoke-Test bestanden. Danach können intensivere Tests folgen. Der Smoke-Test ist ein Standardtest in der Softwarequalitätssicherung, der das Ziel hat festzustellen ob das Testobjekt für einen intensiveren Test geeignet ist. Der Nachteil des Smoke-Test ist, dass er kein festes Abbruchkriterium hat. Da es während diesem Test zu keinem offensichtlichem Fehlverhalten kam ist der Test bestanden.
2. Die Kontrollelementanzeigen wurden getestet indem der 2. Test der Systemtestvorschrift des Systemtests „Fahrt“ durchgeführt wurde. Die folgende Abbildung zeigt den Verlauf der Größen Bordspannung, CRIO-Temperatur und Hydrauliktemperatur.

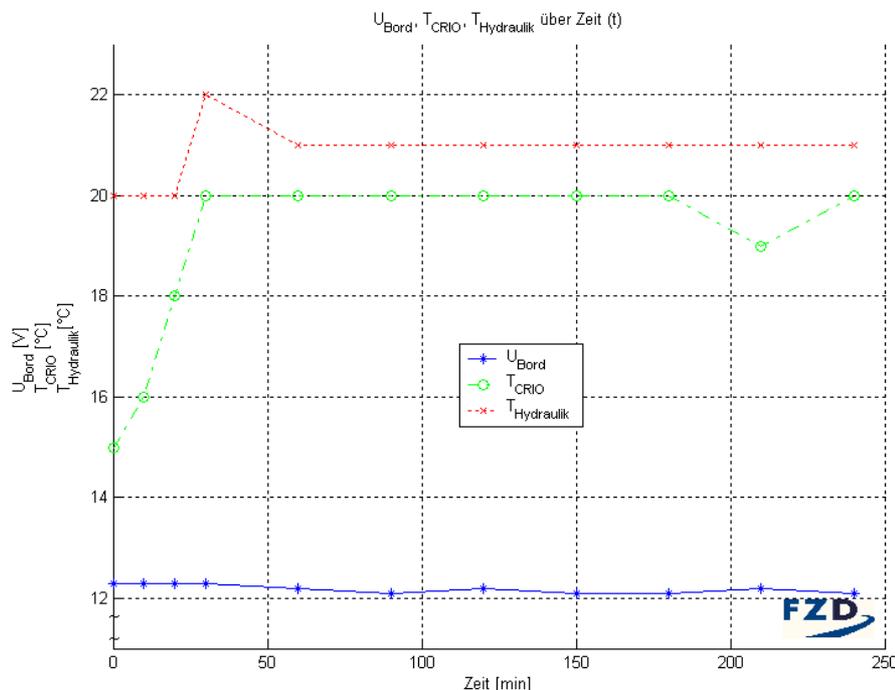


Abbildung 9.6: Veränderung der Kontrollgrößen U_{Bord} , T_{CRIO} , $T_{Hydraulik}$

²⁷ Spillner, Linz: Basiswissen Softwaretest, 2003, S.127

Wie erwartet fällt die Bordspannung mit der Dauer des Messbetriebs, während die CRIO-Temperatur steigt. Die Hydrauliktemperatur blieb über den kompletten Messbetrieb nahezu konstant, bei 21°C, obwohl die Außentemperatur deutlich geringer war. Daher wurde eine neue Kalibrierung durchgeführt, bei der festgestellt wurde, dass tiefere Temperaturen als 23°C in der aktuellen Konfiguration des Temperatursensors nicht gemessen werden können. Der Hydraulikdruck wurde während der Messung erfasst. Die nachfolgenden Abbildungen zeigen den Hydraulikdruckverlauf während der Messung bei 30km/h im 2. Gang und bei 40km/h im 4.Gang. Wie zu erwarten steigt der Druck bis μ_{\max} an und fällt danach wieder ab. Bei 40km/h im 4.Gang steigt der Druck höher als bei 30km/h im 2. Gang. Da die Anzeigen der Kontrollgrößen plausible Werte liefern ist dieser Test bestanden.

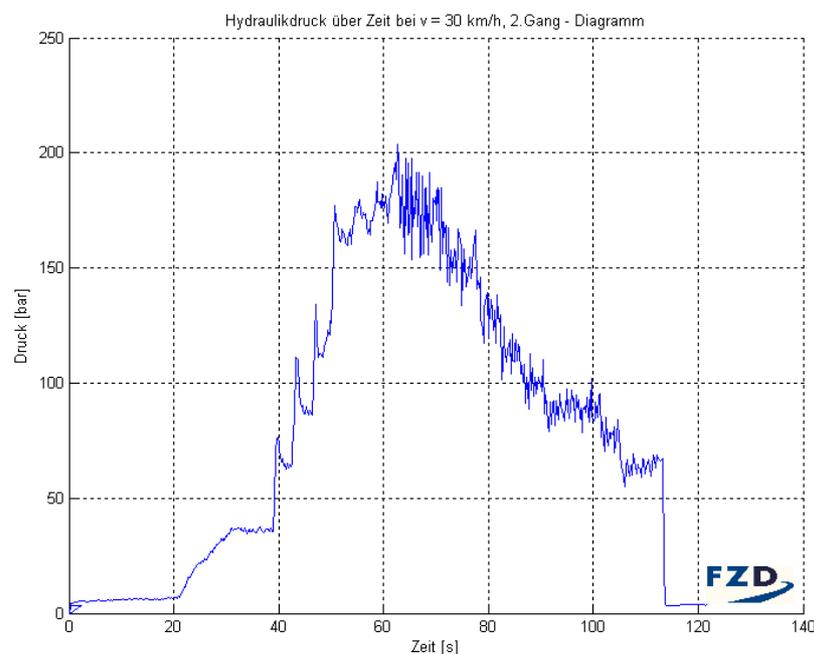


Abbildung 9.7: $p_{\text{Hydraulik}}$ bei 30km/h

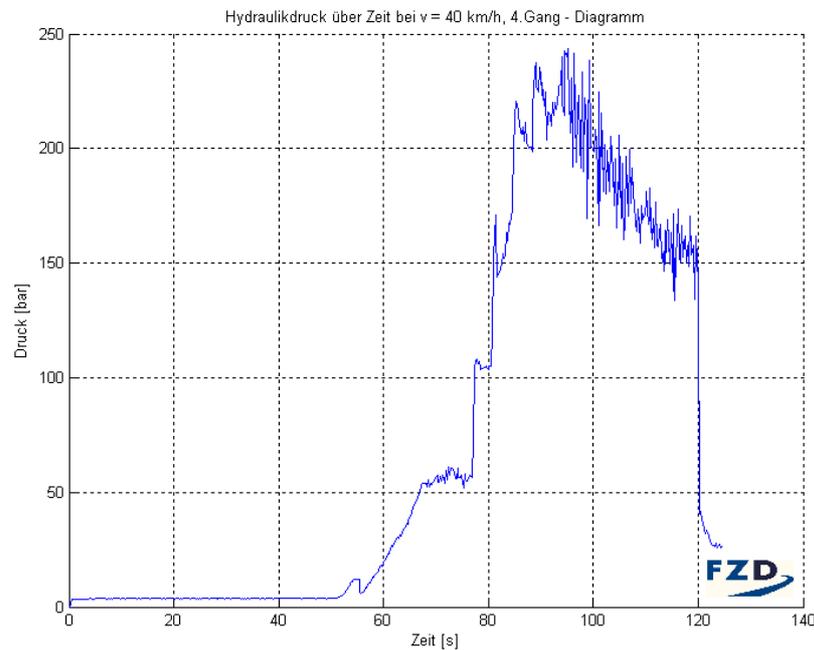


Abbildung 9.8: $p_{\text{Hydraulik}}$ bei 40 km/h

- Die Anzeige der Statusparameter wurde entsprechend der Systemtestvorschrift getestet. Zunächst wurde der Geschwindigkeitstest durchgeführt. Wie erwartet nimmt die Geschwindigkeit beim Beschleunigen bis v_{max} hier 15 m/s zu und beim Ausrollen ab. Die folgende Abbildung zeigt die Fahrzeuggeschwindigkeit über der Radgeschwindigkeit, bei schlupffreier Fahrt (es wird kein zusätzliches Bremsmoment aufgebracht). Wie erwartet haben beide den gleichen Verlauf. Die übernächste Abbildung zeigt die 2 Geschwindigkeiten übereinander. Hierbei ist schlupfbedingt eine Hysterese zu erkennen, der untere Teil der Kurve zeigt das Beschleunigen und der obere Teil das Ausrollen.

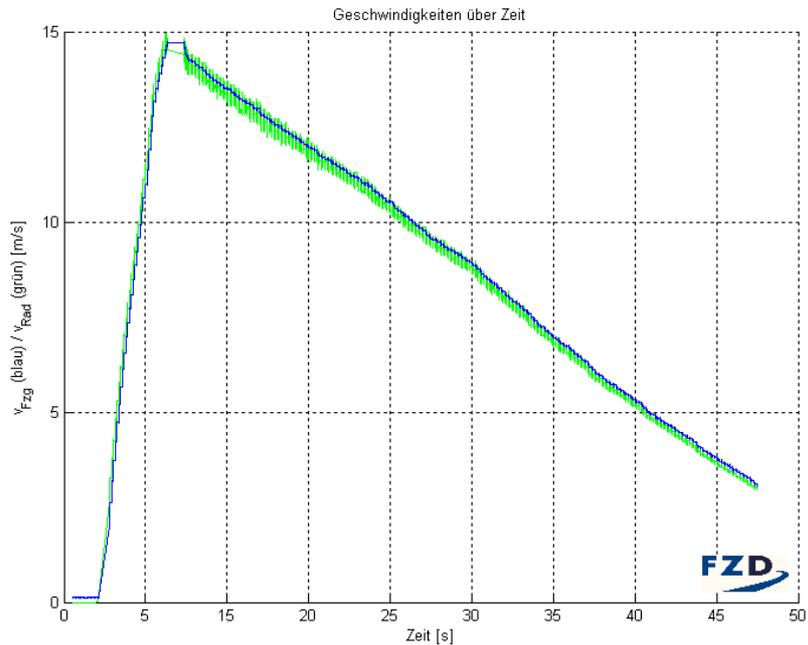


Abbildung 9.9: v_{FzG}, v_{Rad} über Zeit

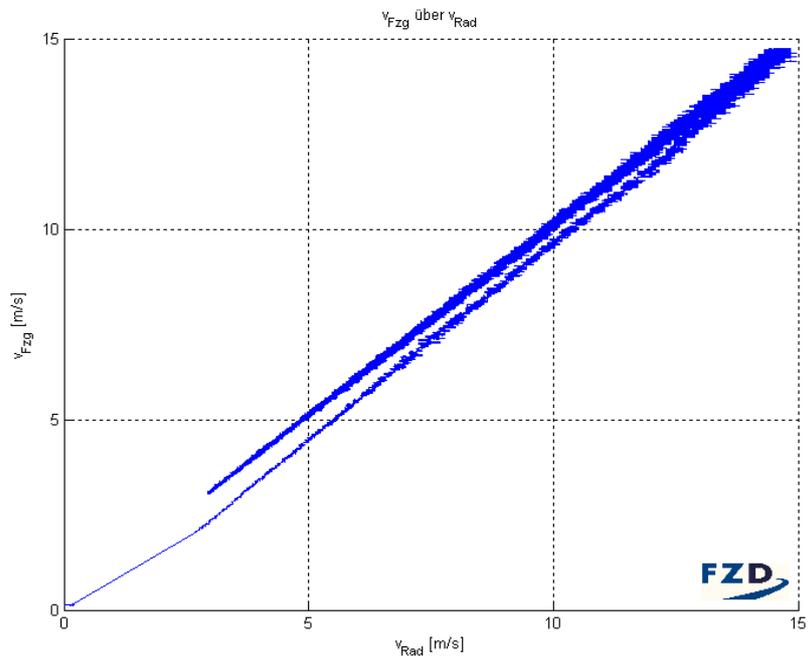


Abbildung 9.10: v_{FzG} über v_{Rad}

Im nächsten Schritt wurden verschiedene Schräglaufwinkel eingestellt und die Ausgaben Radlenkwinkel am Messrad, Schräglaufwinkel sowie Schwimmwinkel in Abhängigkeit der eingestellten Toleranz und Schwimmwinkelkompensation, mit den erwar-

teten Ausgaben verglichen. Da die Istwerte beim Test mit den Sollwerten übereinstimmen ist der Test, welcher auch die Schräglaufwinkeleinstellung und Schwimmwinkelkompensation prüft, bestanden.

4. Der Test der Warnungen beim Verlassen von Parametergrenzen wurde entsprechend der Systemtestvorschrift durchgeführt. Dieser Test wurde, zum Einen wegen Bauteilschutz und zum Anderen um die Grenzen überhaupt erreichen zu können, mit verringerten Grenzen durchgeführt. Dieser Test wurde ohne Beanstandung bestanden alle Warnungen funktionieren vorschriftsmäßig.
5. Der Test Herunterfahren der Messtechnik beim Überschreiten kritischer Parametergrenzen wurde wie der 4. Test wegen Bauteilschutz mit verringerten Werten durchgeführt. Es wurde jeweils die Messdatenaufzeichnung abgebrochen, die Messdaten gesichert, die Messhardware heruntergefahren und somit der Test bestanden.
6. Der Test zur Bremsmomentverstellung am Messrad wurde entsprechend der zugehörigen Vorschrift durchgeführt. Der Schwimmwinkel konnte dabei immer ausgeregelt werden, wie die Beobachtung der Größe Schwimmwinkel auf dem Hostsystem während der Messung zeigte. Die Messung erfolgte bei 30km/h im 2. Gang. Das Proportionalventil wurde in 20 Stufen von 1V bis 3,5V verstellt. Die Bremskraft die in der folgenden Abbildung zu sehen ist nimmt wie erwartet bis μ_{\max} zu und danach wieder ab. Zum Vergleich wurden Stufen eingezeichnet. Das Filtern des Bremskraftverlaufs führte zu keiner besseren Darstellung, daher wurde in der Abbildung darauf verzichtet. Da diese Verifikation mit der Kraftmessdose durchgeführt wurde, da die Messfelge einen Defekt hatte, ist ein sehr großes Schwingen des Systems zwischen $t = 30s$ und $t = 40s$ zu erkennen.

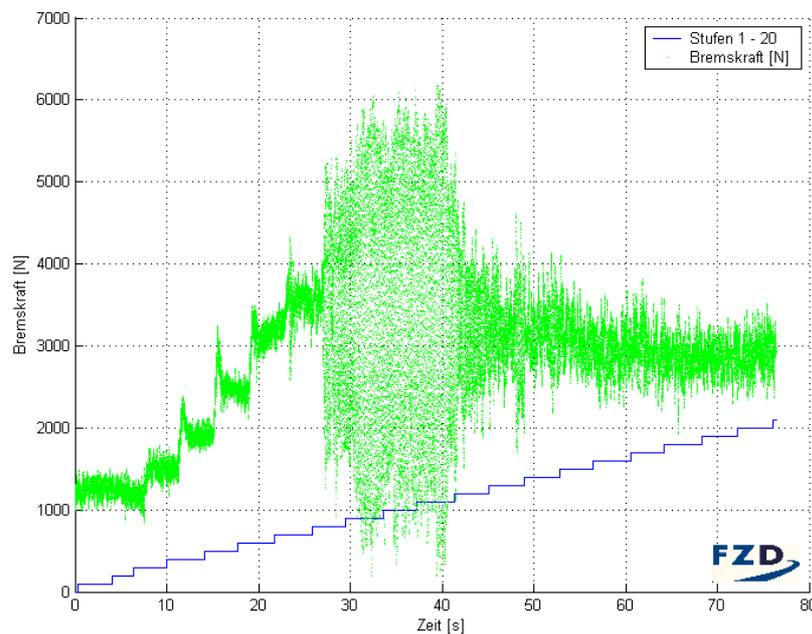


Abbildung 9.11: Bremskraftverlauf bei Proportionalventilerhöhung

7. Der Test zur Schräglaufverstellung am Messrad wurde wie in der Systemtestvorschrift beschrieben durchgeführt, hierbei wurde eine Schräglaufwinkeltoleranz von $0,5^\circ$ und eine Schwimmwinkeltoleranz von 1° eingestellt. Die Regelung erfolgte somit auf $0,25^\circ$ genau bei der Schräglaufeinstellung und auf $0,5^\circ$ genau bei der Schwimmwinkelregelung. Die abgelesenen Werte in der Software, welche bei den Testfällen eingeregelt wurden waren entsprechend den erwarteten Werten, somit ist der Test bestanden.

Bemerkungen:

Die Differenz zwischen Radlenkwinkel und Schräglaufwinkel entspricht nicht immer dem Schwimmwinkel, da aufgrund des Messtechnikrauschens die Werte verfälscht werden und durch Mittelwertbildung und Rundung auf eine Stelle nach dem Komma Rundungsfehler entstehen.

Beim Testfall 15, bei dem ein Schräglaufwinkel von -5° bei 50km/h eingestellt wurde, konnte der Schwimmwinkel erst nach längerem Überschwingen ausgeregelt werden, bei der eingestellten Motorstromstärke und Schwimmwinkeltoleranz von 1° , was einer Regelung von $0,5^\circ$ entspricht.

8. Die Schwimmwinkelkompensation wurde bereits durch die Tests 3, 6 und 7 verifiziert.

9. Zur Verifikation des Drehzahlsignals wurde R_{dyn} sowohl manuell als auch mittels Messung und anschließender Offlineauswertung ermittelt. Der verwendete Reifen ist ein neuer Conti Sport mit einer Profiltiefe von 7mm.

- manuelle R_{dyn} -Bestimmung:

Hierzu wurde mit einem Stück Kreide der Aufstandspunkt des Reifens, sowohl am Reifen als auch auf der Fahrbahn markiert. Es wurde der Anhänger exakt eine vorher festgelegte Anzahl Umdrehungen $U = 10$ des Rades gezogen und der Endpunkt auf der Fahrbahn markiert. Danach wurde die Strecke $s = 19,47\text{m}$ ermittelt und R_{dyn} wie folgt berechnet.

$$R_{\text{dyn,manuell}} = \frac{s}{2 * \pi * U} = 0,310 \quad (9.3)$$

- R_{dyn} -Bestimmung durch Messung und Offlineauswertung

Es wurde eine Messung bei $v = 30\text{km/h}$ ohne Bremsmoment gefahren und R_{dyn} mittels folgender Formel berechnet. Die Messung wurde bei 30km/h gefahren, da der Luftwiderstand von PETRA I bei geringer Geschwindigkeit zu vernachlässigen ist, es wurde nicht langsamer als 30km/h gefahren, da mit geringerer Geschwindigkeit die Drehzahl seltener aktualisiert wird.

$$R_{\text{dyn,gemessen}} = \frac{v_{\text{Fzg}}}{n_{\text{Rad}} * 2 * \pi} \quad (9.4)$$

Der Mittelwert von $R_{\text{dyn,gemessen}}$ beträgt $0,310$.

Die folgende Abbildung zeigt $R_{\text{dyn,gemessen}}$ und dessen Mittelwert sowie $R_{\text{dyn,manuell}}$. Beide ermittelten dynamischen Reifenhalmesser liegen innerhalb des Fehlerbereichs. Somit ist die Drehzahl verifiziert.

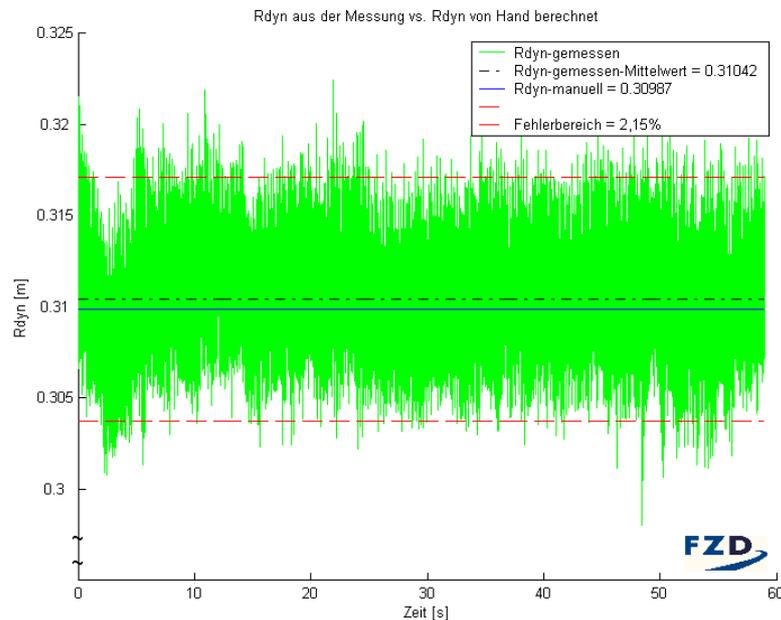


Abbildung 9.12: Vergleich $R_{dyn,gemessen}$ mit $R_{dyn,manuell}$

Fehlerbetrachtung der Drehzahlverifikation:

Fehler Streckenmessung = 10cm bei $s = 19,47m$ Fehler Streckenmessung = 0,51%

Fehler Rauschen Messtechnik = 10mV mit Correvit (Faktor = 20, $v_{max} = 100 \text{ km/h}$ bei $U_{max} = 5 \text{ V}$) $1V \triangleq 20\text{km/h}$ $0,01V \triangleq 0,2\text{km/h}$ Fehler Messtechnik bei 30km/h = 0,66%

Fehler R_{dyn} -Messung aus Differenzgeschwindigkeit

Correvitfehler = 0,5%

Correvitkalibrierfehler = 0,46% (siehe Correvitkalibrierfehler bei 30km/h)

Inkrementalgeberfehler = 0%

Drehzahlfehler durch Schräglauferfehler = 2° (aus Radlenkwinkelkalibrierung + eingestellter Toleranz)

$$\text{Fehler } \alpha = 1 - \cos(2^\circ) * 100 = 0,02\% \quad (9.5)$$

$$\text{Fehler } R_{dyn}\text{-Messung} = 0,5\% + 0,46\% + 0,02\% = 0,98\%$$

$$\text{Gesamtfehler} = 0,66 + 0,98 + 0,51 = 2,15\%$$

10. Der Test R_{dyn} -Berechnung wurde entsprechend der Testspezifikation durchgeführt. Die folgende Grafik zeigt R_{dyn} bei 30 km/h welches mit der Software 20 mal bestimmt wurde.

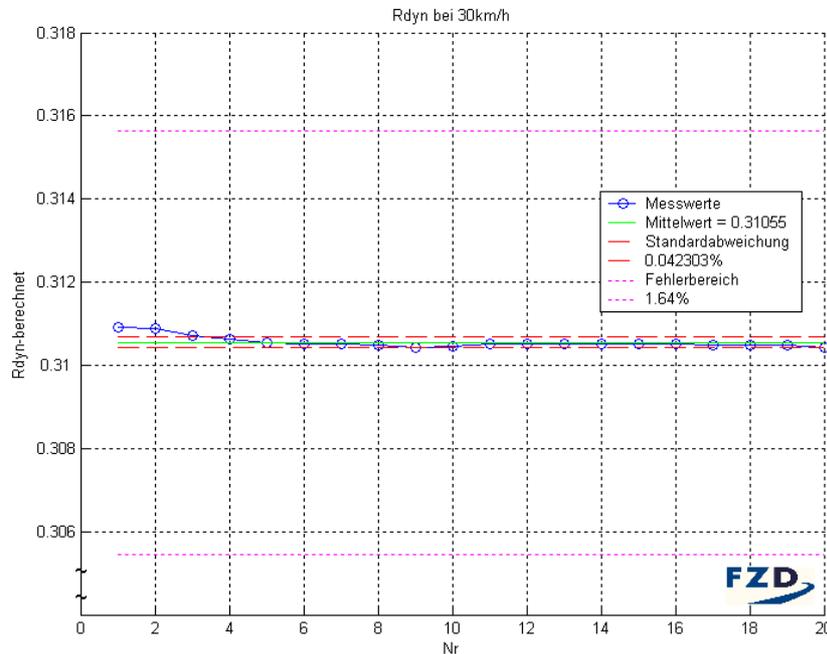


Abbildung 9.13: $R_{dyn,berechnet}$

Wird der Mittelwert der berechneten dynamischen Reifenhalmmesser mit dem manuell bestimmten Reifenhalmmesser und dem gemessenen Reifenhalmmesser verglichen, lässt sich erkennen, dass alle 3 innerhalb des Fehlerbereichs liegen.

$$R_{dyn,berechnet} = 0,31055$$

$$R_{dyn,manuell} = 0,30987$$

$$R_{dyn,gemessen} = 0,31042$$

Die Differenz der verschiedenen ermittelten dynamischen Reifenhalmmesser liegt bei 0,2%, damit ist die R_{dyn} -Berechnung verifiziert.

11. Test Messungsdurchführung und Messdatenaufzeichnung:

Der Test wurde entsprechend der Systemtestvorschrift durchgeführt. Da die Messfelge defekt war wurden diese Versuche ohne Messfelge gefahren. Die folgenden Abbildungen, welche im Anhang vergrößert zu finden sind, zeigen die Rohdaten der Reibwertschlupfkurven der gefahrenen Messungen. Es ist jeweils der Reibwertschlupfkurvenverlauf der Reibwert_{längs}-Schlupf-Kurve zu erkennen. Da zum Zeitpunkt der Verifikation die Messfelge einen Defekt hatte konnten mit der Kraftmessdose nur Reibwert_{längs}-Schlupf-Kurven ermittelt werden. Die Abbildungen 9.15 – 9.17 zeigen die stationär erfassten Kurven. Hierbei wurden die Messwerte nur erfasst, wenn diese in den erwähnten Toleranzbereichen lagen. Somit handelt es sich hierbei um einen quasistationären Zustand. Diese Kur-

ven zeigen den erwarteten Reibwert_{längs}-Schlupf-Verlauf. Bei der dynamischen Messung, Abbildung 9.14 wurden alle Messwerte aufgezeichnet, daher ergeben sich beim Beschleunigen negative Schlupfwerte. Außerdem sind die Punktwolken nicht voneinander zu unterscheiden, da hier alle dynamischen Effekte erfasst wurden. Auffällig ist, dass mit zunehmender Geschwindigkeit der stationären Messungen, das Streuen der Punktwolke und somit die Streuung der Bremskraft. Für diese Verifikation wurde die Kraftmessdose verwendet, da die Messfelge wie bereits erwähnt einen Defekt hatte. Wie zu erkennen ist, zeigt die Kraftmessdose eine starke Streuung der Messwerte entlang des Reibwerts. Spätere Messungen mit der Kraftmessfelge haben gezeigt, dass diese Streuung auch dort zu finden ist aber nicht so stark ausgeprägt. Aufgrund dieser starken Streuung, ist die Kraftmessdose für eine präzise Erfassung der μ -Schlupf-Kurven nicht zu empfehlen.

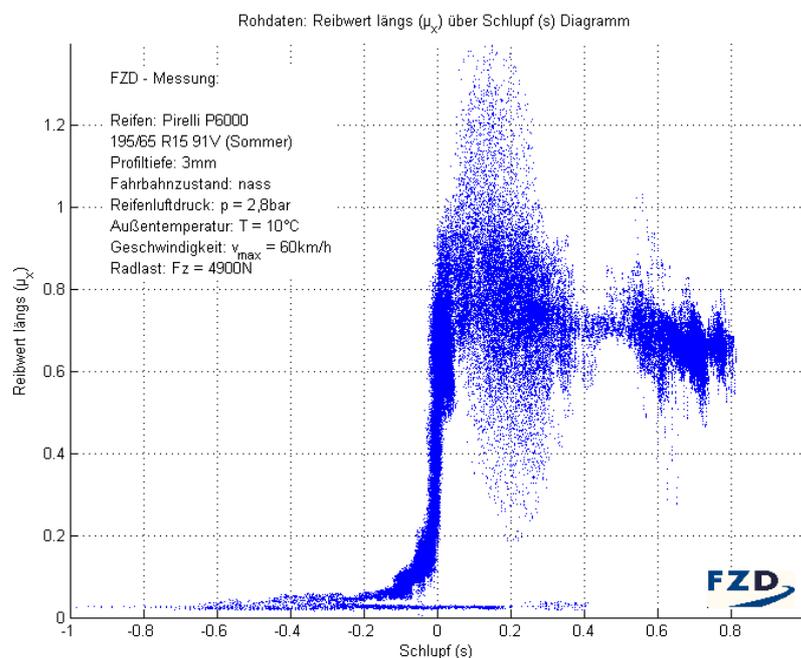


Abbildung 9.14: dynamische Messung

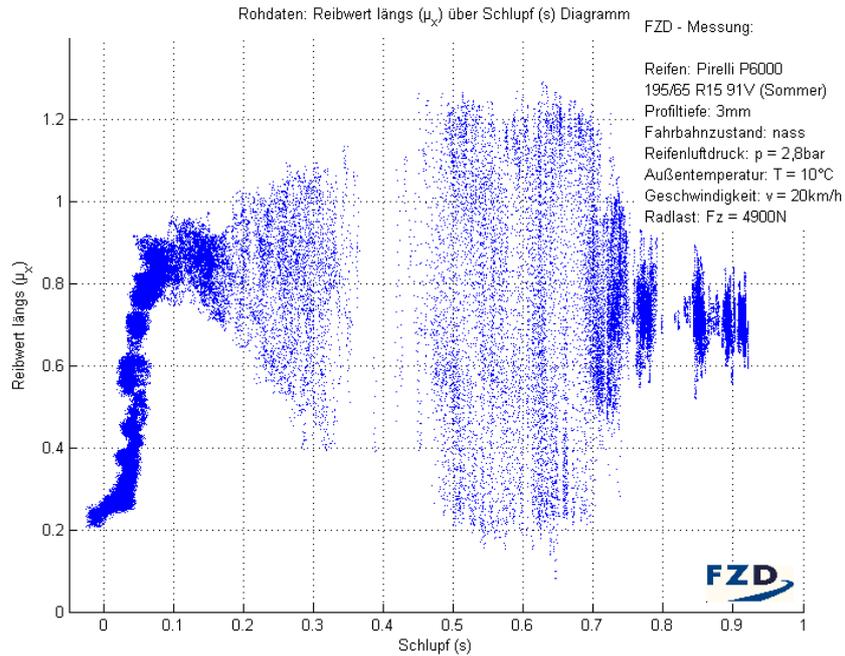


Abbildung 9.15: stationäre Messung 20

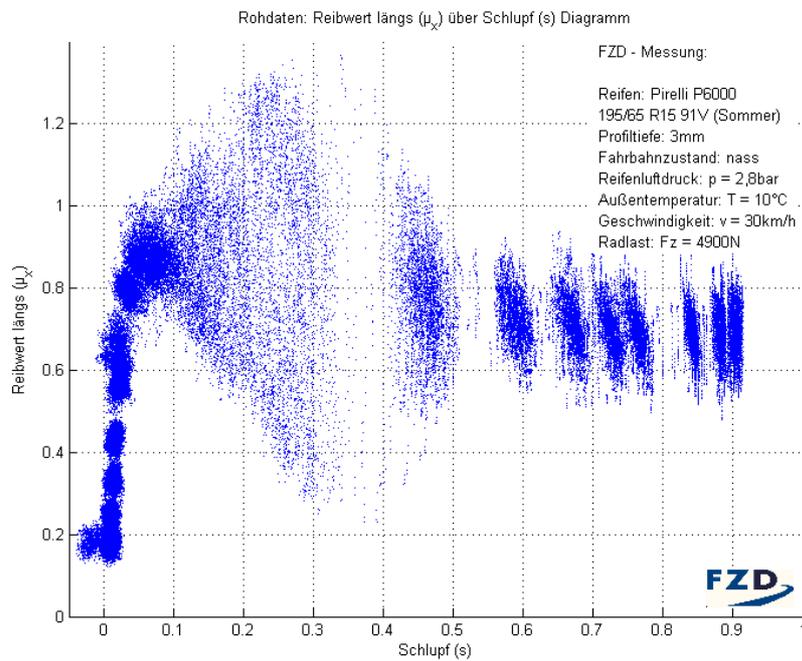


Abbildung 9.16: stationäre Messung 30

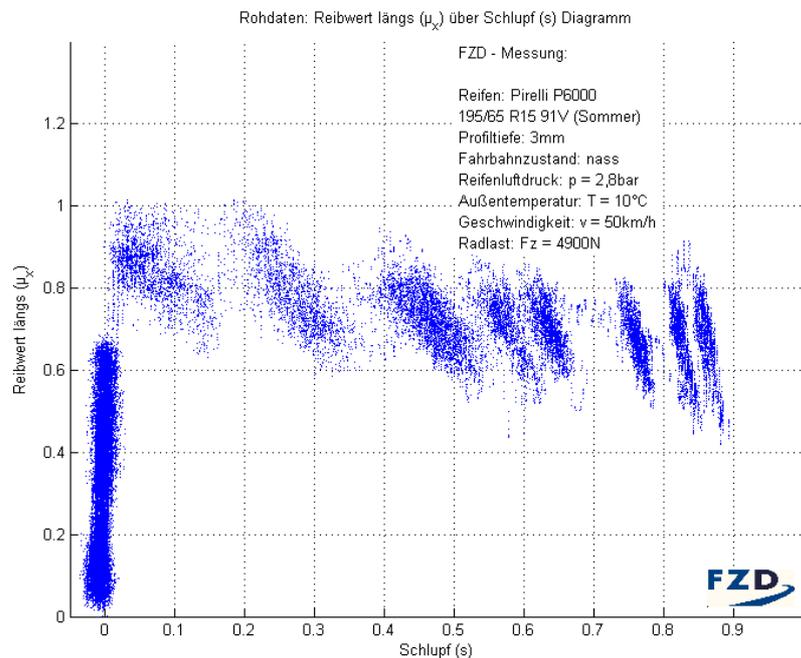


Abbildung 9.17: stationäre Messung 50

12. Der Performancetest wurde mit 4kHz begonnen und von Test zu Test um 1 kHz erhöht. Der erste Systemabsturz trat bei 8kHz auf. Während der Validierung die zunächst mit 7kHz begonnen wurde kam es damit öfter zum Systemabsturz. Bei 6 kHz kam es bisher zu keinem Systemabsturz. Daher wird, mit einem Sicherheitsabstand von 1kHz, empfohlen die maximale Abtastrate von 5kHz nicht zu überschreiten.
13. Der Kompatibilitätstest zu Matlab wurde durch einlesen der Messwerte in Matlab bestanden.
14. Als Zuverlässigkeitstest wurden während der Systemverifikation & Validierung die Systemabstürze gezählt. Es kam zu 3 Systemabstürzen, bei dem das CRIO-System nicht mehr reagierte, wenn die performancebedingten Systemabstürze nicht berücksichtigt werden. Da es sich bei der Software um eine Neuentwicklung handelt sind drei Systemabstürze bei der Anzahl der Tests akzeptabel.

10 Validierung

10.1 Ziel der Validierung

Ziel der Systemvalidierung ist zu prüfen, ob das System die festgelegten Aufgaben tatsächlich löst und daher für den Einsatzzweck tauglich ist, vgl. Spillner²⁸. Für dieses Projekt bedeutet dies, dass mit der Software PAMELA und dem Messanhänger PETRA Reibwert-Schlupf-Kurven gemessen werden.

Ergebnis der Validierung ist ein validiertes System, d.h. das System erfüllt seinen Einsatzzweck.

10.2 Validierungsvorschrift

Die Validierung erfolgt durch Messfahrt mehrerer μ -Schlupfkurven von Pkw-Reifen mit dem Messanhänger PETRA I und der Software PAMELA.

Es werden μ -Schlupfkurven bei 3 Geschwindigkeiten (20, 30 und 50km/h) mit je 5 Schräglaufwinkeln ($0, \pm 2$ und $\pm 4^\circ$) gemessen, d.h. 15 Parametervariationen. Vor jeder Parametervariation mit veränderter Geschwindigkeit wird R_{dyn} neu berechnet. Die Messdatenerfassung erfolgt mit dem aus der Verifikation als Maximum ermittelten Abtastrate von 7000Hz. Somit kann geprüft werden ob diese Abtastrate an der Grenze der maximal möglichen Abtastrate im Dauerbetrieb (Messbetrieb) gefahren werden kann.

Die Messwerte werden automatisch in die Messdatei geschrieben, welche offline mit z.B. Matlab eingelesen, verarbeitet und ausgewertet wird. Ergebnis ist zu jedem Versuch ein μ_{quer} -Schlupf- und ein $\mu_{\text{längs}}$ -Schlupf-Diagramm. Die Parameter die variiert werden sind Geschwindigkeit und Schräglaufwinkel. Außerdem haben Reifeneigenschaften (Gummimischung, Reifenart, Profiltiefe), Asphalteeigenschaften (Asphalttemperatur, Fahrbahnbelag, Makrotextur), sowie Eigenschaften des Zwischenmediums (Trockenheit, Nässe, Schnee, Eis) entscheidenden Einfluss auf den Verlauf der zu messenden μ_{quer} -Schlupf- und $\mu_{\text{längs}}$ -Schlupf-Kurven. Da die Versuche immer mit dem gleichen Reifen auf dem gleichen Streckenabschnitt bei gleichen Fahrbahnbedingungen gefahren werden, ist zusätzlich nur die Profiltiefe zu erfassen. Eine Lichtschranke, um zu gewährleisten, dass immer auf dem gleichen Streckenabschnitt gemessen wird, wird nicht verwendet, da die Streckenabschnitte auf dem die Messung aktiv

²⁸ Spillner, Linz: Basiswissen Softwaretest, 2003, S.41

ist, abhängig vom Fahrer sind, da dieser immer wieder durch Gas geben die Bremskraft kompensiert und die Messung erst dann startet wenn der Geschwindigkeitsbereich erreicht ist.

Die μ_{quer} -Schlupf- und $\mu_{\text{längs}}$ -Schlupf-Diagramme werden folgendermaßen auf Plausibilität geprüft. Der maximale Kraftschlussbeiwert (μ_{max}) auf dem Testgelände, dem Flugplatz in Griesheim, liegt (anhand von Erfahrungswerten) zwischen 1,1 und 1,2 bei trockenen Bedingungen. Der Schlupf bei dem der maximale Reibwert (μ_{max}) erreicht wird liegt nach Reimpell²⁹ zwischen 10% und 30%. Im Anfangsbereich der $\mu_{\text{längs}}$ -Schlupf-Kurven besteht ein lineares Verhalten und am Ende dieser Kurve bei Schlupf gleich 100% nähern sich alle Kurven einem Punkt, dem sogenannten Gleitreibwert (μ_{gleit}). Sind die μ_{quer} -Schlupf- und die $\mu_{\text{längs}}$ -Schlupf-Diagramme plausibel, ist die Software validiert. Es wird eine Plausibilitätsprüfung verwendet, da weder eine Simulation noch ein Vergleichssystem zur Validierung herangezogen werden kann.

Die Versuchstabelle (Parametervariationliste) ist dem Anhang H.1 zu entnehmen. Der Schräglaufwinkel wird nur bis $\pm 4^\circ$ eingestellt, um die Bauteilsicherheit zu gewährleisten, da bei einem Schräglaufwinkel von mehr als $\pm 4^\circ$ die Seitenkraft und das Reifenrückstellmoment sehr groß werden. Außerdem würde ein noch größerer Schräglaufwinkel, zu einem höheren Abrieb führen. Die maximale Geschwindigkeit beträgt 50 km/h im 1. Gang des Automatikgetriebes, da das Versuchsfahrzeug, ein Citroen XM (Zylinder: V6, ca. 140kW), keine höhere Geschwindigkeit im Messbetrieb erreicht ohne die Bauteilsicherheit (roter Bereich der Motordrehzahl, bzw. anormale Getriebegeräusche) zu überschreiten, wie sich bei den ersten Tests gezeigt hat. Auf eine Variation der Radlast, durch Zuladung wird verzichtet, da eine Erhöhung der Radlast mit einer Erhöhung des Bremsmoments gleichzusetzen ist und dies eine zusätzliche Belastung für das Getriebe darstellt, dieses aber bereits an seiner Leistungsgrenze ist, vgl. Diplomarbeit 413/06.

Die Messung wird bei 7000Hz mit einem Δ Stufencounter = 2,5s durchgeführt, daraus ergibt sich, bei einer, mittels Zeitmessung, geschätzten Bremsmomenteinpendelzeit von ca.1s, ein Kollektiv von ca. 10000 Punkten pro Stufe. Da dies ein ausreichend großes Kollektiv an Messpunkten ist und ein neuer Reifen nach ca. 30 Messungen verschlissen ist, wird jede Parametervariation nur einmal für die Validierung durchfahren.

10.3 Vorbedingungen der Validierung

Bevor die Validierung durchgeführt wird, ist der Systemtest erfolgreich zu bestehen. Die Messbereitschaft von PETRA I wird hergestellt, außerdem ist eine Stufenanpassung für den

²⁹ Reimpell, Fahrwerktechnik, 1995, S. 119

verwendeten Reifen und die Fahrbahnbedingungen durchzuführen. Ist diese beendet und dabei kein Fehler aufgetreten, kann die Validierung durchgeführt werden.

10.3.1 Stufenanpassung

Um die Reibwert-Schlupf-Kurve möglichst genau abzubilden, werden die Bremsmomentstufen an die Bedingungen angepasst. D.h. für die aktuelle Bedingung werden die Stufen so verteilt, dass besonders im Bereich um μ_{\max} mehrere Punktwolken liegen und somit μ_{\max} möglichst gut abgebildet wird. Dazu wird mit dem zu verwendenden Reifen eine Messung mit voreingestellten Stufen gefahren. Danach wird die Auswertung betrachtet und die Stufen entsprechend angepasst. Dieser Vorgang wird rekursiv wiederholt, bis die Reibwert-Schlupf-Kurve gut abgebildet wird. Da die Einstellung zusätzlich von der Fahrzeuggeschwindigkeit und dem gewählten Gang abhängig ist und da die Getriebedrehzahl mit der Pumpendrehzahl und somit mit dem Bremsmoment korreliert, erfordert auch jede Variation der Geschwindigkeit oder des Gangs eine Stufenanpassung.

10.4 Durchführung der Validierung

Die Validierung wurde am 8.3.2007 unter folgenden Bedingungen durchgeführt.

Tabelle 10-1: Validierungsbedingungen

Fahrbahn: trocken	Reifenbezeichnung: Continental Sport2
Außentemperatur: 13°	Reifengröße: 205/55 R16 91W
statische Radlast: $F_z = 4700\text{N}$	Reifentyp: Sommerreifen
Abtastrate: 7kHz	Reifenluftdruck: $p = 2,8\text{bar}$

Für den verwendeten Reifen und die Bedingungen, wurde die folgende Stufeneinstellung gefunden, entsprechend dem Verfahren welches in Kapitel 10.2 beschrieben wurde, die bei allen 3 zu fahrenden Geschwindigkeiten im entsprechenden Gang die Reibwert-Schlupf-Kurve für die Validierung zufriedenstellend abbildet.

Tabelle 10-2: Stufeneinstellung der Validierung

Stufe	1	2	3	4	5	6	7	8	9	10
Spannung [V]	1,2	1,225	1,25	1,275	1,3	1,35	1,375	1,4	1,425	1,45
Stufe	11	12	13	14	15	16	17	18	19	20
Spannung [V]	1,48	1,5	1,55	1,6	1,7	1,9	2	2,3	2,6	2,9

Die Validierungsversuche wurden entsprechend der Validierungsvorschrift durchgeführt. Da die Zerokalibrierung, die bei den Standversuchen funktioniert hat, nachdem die Messfelge von einer Reparatur bei der Firma Kistler zurückkam nicht mehr funktioniert, wurde F_y wie folgt ermittelt. Dabei handelt es sich nicht um ein Software- oder Schnittstellenproblem, da die Zerokalibrierung von Hand durch Betätigung des Zerokalibrierungshandschalters und gleichzeitigem Drehen des Messrades auch nicht F_y auf 0 zurücksetzt. Deswegen wurde der Offset F_y wie folgt bestimmt um in der Auswertung F_y korrigieren zu können. Der Anhänger wird ganz langsam (Kriechgeschwindigkeit der Automatik) gezogen, der Radlenkwinkel am Messrad wird auf 0 gestellt, Offset F_y wird bestimmt. Durch dieses Vorgehen soll erreicht werden, dass das Messrad nahezu Seitenkraft frei ist und somit das gemessene F_y in der Auswertung um diesen Offset korrigiert werden kann. Alternativ kann nach dem Anrollen, wodurch die Verzwängung des Rades gelöst wird, erneut angehalten und dann F_y ermittelt werden. Der jeweilige Offset, die Profiltiefe sowie der Name der Messdatei wurden im Validierungsprotokoll, Anhang H2., notiert. Da bei der Validierung die eingestellte Frequenz von 7000Hz zu einigen Softwareabstürzen führte, wurde der letzte Teil der Validierung mit 6000Hz gefahren. Mit 6000Hz kam es bisher bei keiner Messung zu einem Softwareabsturz, somit wird 6000Hz als Grenzfrequenz angenommen, vgl. Kapitel 9.4.7 Test 12.

10.5 Auswertung der Validierung

Die Auswertung wird mit Matlab offline durchgeführt.

10.5.1 Methodik der Auswertung

Zunächst wird nach Möglichkeiten der Auswertung gesucht und die Vorteile sowie Nachteile ermittelt. In der nachfolgenden Abbildung sind die gefundenen Auswertemöglichkeiten sowie deren Vor- und Nachteile aufgeführt. Zur Erkennung eines „quasistationären“ Zustandes wurde ein Brainstorming durchgeführt. Für die Möglichkeiten der Auswertung wurde anschließend eine Bewertung in 2 Schritten durchgeführt. Im 1. Schritt wurden die Auswertemethoden bewertet. Für die bestbewertete Methode „beliebig viele Stufen“ wurde eine Bewertung

der Kriterien zur Erkennung der „quasistationären Zustände“ durchgeführt. Das bestbewer- tetste Kriterium „Kombination von Schlupf und Längsreibwert“ wird in der Auswertung ver- wendet. Das Brainstorming, die beiden Bewertungen sowie die Bewertungsmatrix können im Anhang H3. nachgesehen werden.

Für die Auswertung wird somit die Methode „beliebig viele Stufen“ mit dem Kriterium „Kombination von Schlupf und Längsreibwert“ zur Erkennung der „quasistationären Zustän- de“ verwendet.

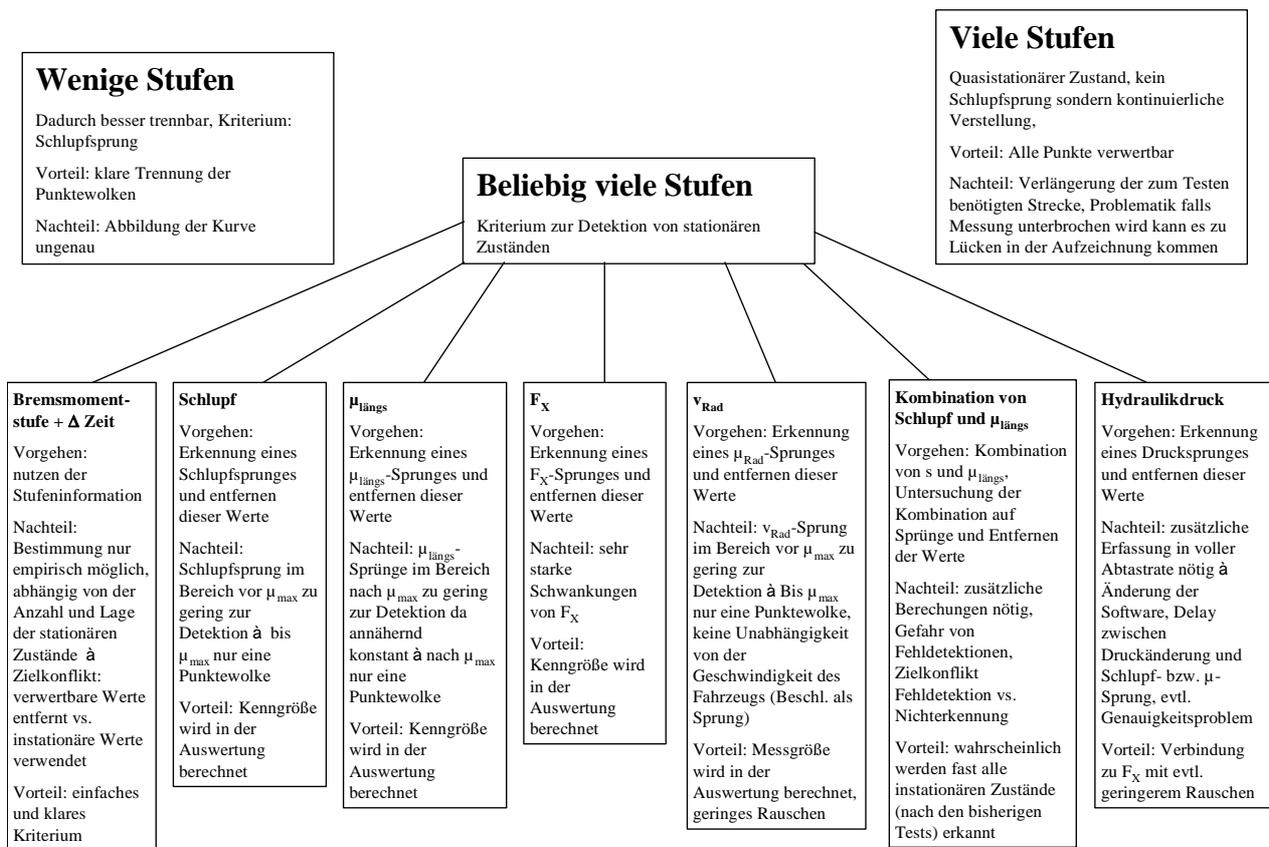


Abbildung 10.1: Möglichkeiten zur Auswertung der Messdaten

Nachdem das Kriterium „Kombination von Schlupf und Längsreibwert“ feststeht, wird eine Methode bzw. ein Algorithmus benötigt den Schlupf und Längsreibwert zu kombinieren. Hierzu wird jeweils die Ableitung des Schlupfs und des Längsreibwerts gebildet, da ein Schlupfsprung bzw. ein $\mu_{längs}$ -Sprung zu einem Extremwert der Ableitung führt. Je stärker die Änderung, d.h. der Sprung ist, desto höher ist der Peak der Ableitung an dieser Stelle. Wird ein $\mu_{längs}$ -Sprung oder ein Schlupfsprung detektiert, liegt an dieser Stelle ein instationäres Verhalten vor. Da die Rohsignale ein Rauschen aufzeigen, werden für die Erkennung der qua-

sistationären Punktwolken zunächst die Rohsignale gefiltert, daraus Schlupf und $\mu_{\text{längs}}$ berechnet, deren Ableitung gebildet und mittels Schwellenwerte die instationären Bereiche erkannt. Die Eckfrequenz des Filters sowie die Schwellenwerte wurden empirisch ermittelt. Zur Filterung wurde eine Butterworth Tiefpassfilter 1. Ordnung verwendet der mittels der Matlab-funktion „filtfilt“ gefiltert wurde. Die empirisch ermittelten Werte sind:

Eckfrequenz = 0,75Hz Schlupfschwelle = $\pm 0,00003$ Reibwertschwelle = $\pm 0,00009$

10.5.2 Auswertung der Messdaten

Alle Plots bzw. Abbildungen in diesem Kapitel, sind im Anhang H4. vergrößert zu finden. Des Weiteren sind im Anhang weitere Plots zu finden die in diesem Kapitel den Rahmen der Arbeit sprengen würden, wie z.B. alle gefahrenen Messungen einzeln. Daher werden im Folgenden die Messdaten anhand exemplarisch ausgewählter Kurven erläutert.

Werden die Rohdaten der μ -Schlupf-Kurven, mit dem Schräglaufwinkel 0 Grad, bei den verschiedenen Geschwindigkeiten, siehe die folgenden Abbildungen bzw. Anhang H4.1., verglichen, fällt auf, dass ein starkes Rauschen bei 20km/h im mittleren Bereich und bei 50km/h am Ende der Kurve zu finden ist, wohingegen das Rauschen bei 30km/h nicht so stark ist. Dieses Phänomen wird in der Diplomarbeit 413/06 betrachtet, konnte aber auch dort nicht geklärt werden.

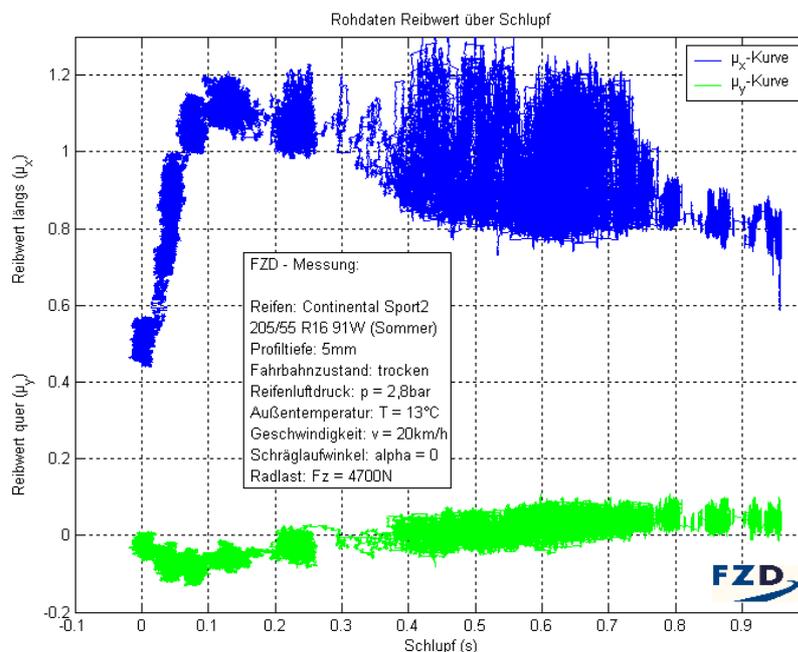


Abbildung 10.2: μ -Schlupf-Rohdaten bei $v = 20\text{km/h}$ und $\alpha = 0^\circ$

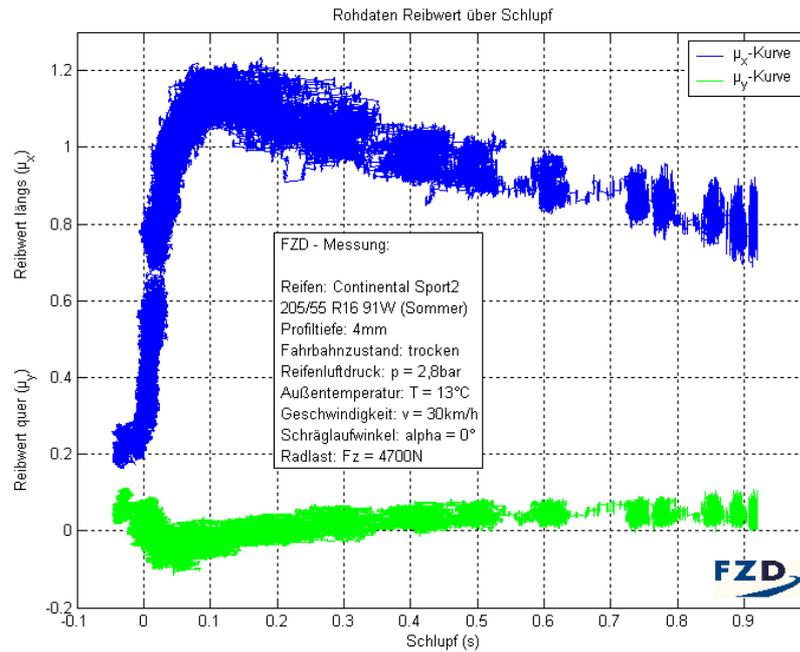


Abbildung 10.3: μ -Schlupf-Rohdaten bei $v = 30\text{km/h}$ und $\alpha = 0^\circ$

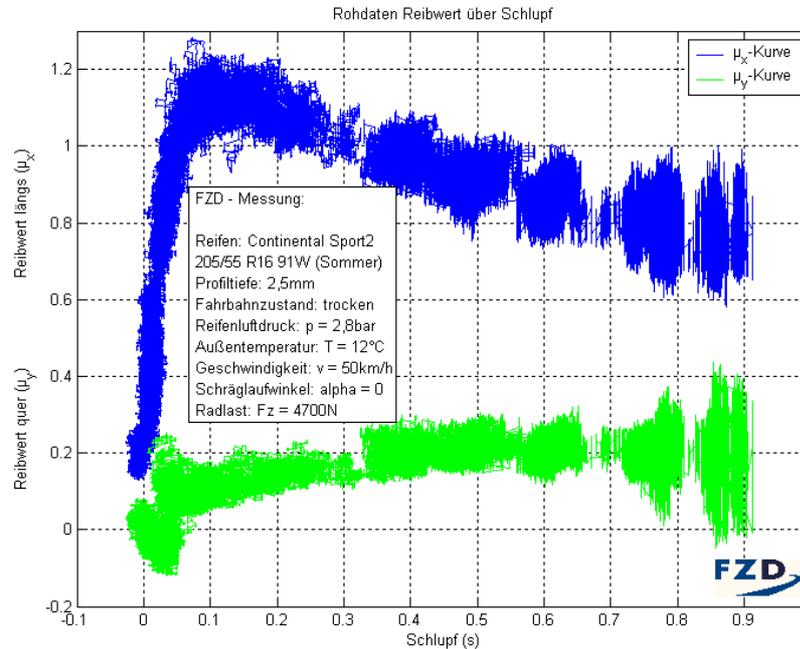


Abbildung 10.4: μ -Schlupf-Rohdaten bei $v = 50\text{km/h}$ und $\alpha = 0^\circ$

In der Auswertung wird zu jedem Punkt auf der μ -Schlupf-Kurve ein Fehlerbalken sowohl für den Reibwert als auch den Schlupf eingezeichnet. Dabei wurde ein Fehlerbereich mit der ein-

fachen Standardabweichung (1σ) angegeben. D.h., dass bei einer Normalenverteilung 68% der Messwerte in diesem Fehlerbereich liegen.

Bei den erzeugten $\mu_{\text{langs}}$ -Schlupf-Kurven wurde zusatzlich der Punkt (0,0) eingefugt, da bei 0% Schlupf keine Langskraft ubertragen wird, der Punkt aber nicht gemessen werden kann da bei dem Minimaldruck der Pumpe von 40bar, ab dem diese schalten kann sofort ein Bremsmoment anliegt.

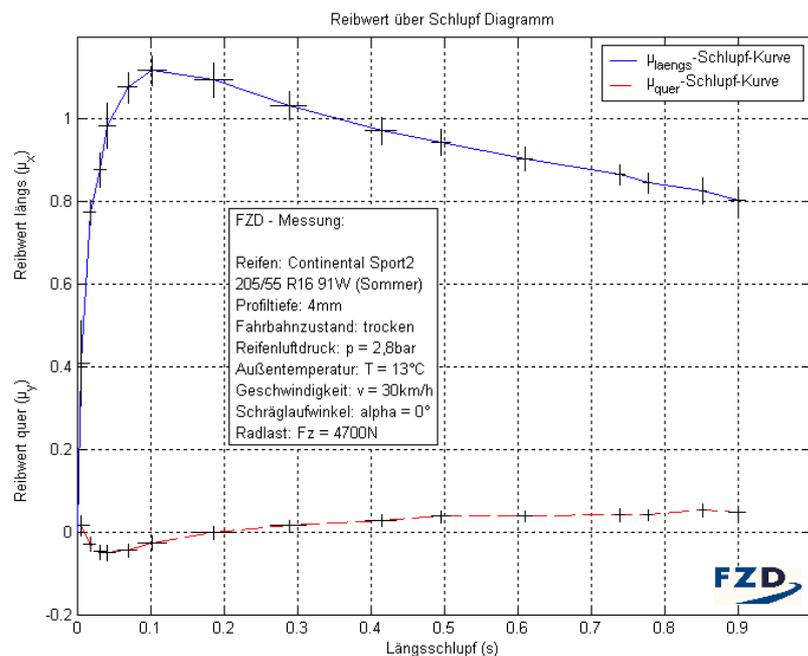


Abbildung 10.5: μ -s-Kurve ($v=30\text{km/h}$ $\alpha=0^\circ$)

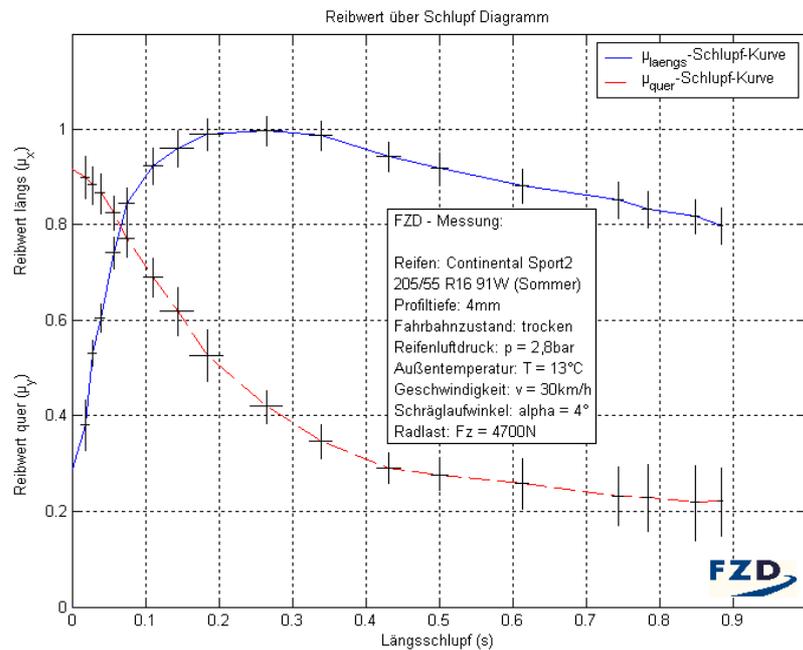


Abbildung 10.6: μ -s-Kurve ($v=30\text{km/h}$ $\alpha = 4^\circ$)

Die Abbildung 10.5 zeigt beispielhaft eine der gemessenen Reibwert-Schlupf-Kurven, die auf Plausibilität geprüft wird. Die $\mu_{\text{längs}}$ -Schlupf-Kurve hat den typischen Verlauf der erwartet wurde. μ_{max} liegt wie erwartet zwischen 1,1 und 1,2 und wird bei ca. 10% erreicht, was auch dem Vergleich mit der Literatur entspricht. Die μ_{quer} -Schlupf-Kurve der Abbildung 10.6 hat auch den erwarteten Verlauf, nämlich zu Beginn ihr Maximum und nimmt mit höherem Schlupf ab. Die zugehörige μ_{quer} -Schlupf-Kurve hat ebenfalls wie erwartet ihr Maximum zwischen 10 und 30% Schlupf.

Bei den folgenden Abbildungen wurden zur besseren Übersicht die Reibwerte, die mit einem negativen Schräglaufwinkel gemessen wurden, negativ dargestellt. Hierbei sei darauf hingewiesen, dass es keinen „negativen“ Reibwert gibt. Die Abbildung 10.7 zeigt die Kurven (0° , $\pm 2^\circ$, $\pm 4^\circ$) bei $v = 20\text{km/h}$ und die Abbildung 10.8 bei $v = 30\text{km/h}$. Es ist zu erkennen, dass sich die $\mu_{\text{längs}}$ -Schlupf-Kurven μ_{gleit} bei maximalem Schlupf annähern. Die μ_{quer} -Schlupf-Kurven nähern sich ebenfalls einem gemeinsamen Punkt an, wobei die Kurve bei 30km/h und 0° herausfällt. Somit ist das Softwaresystem PETRA validiert.

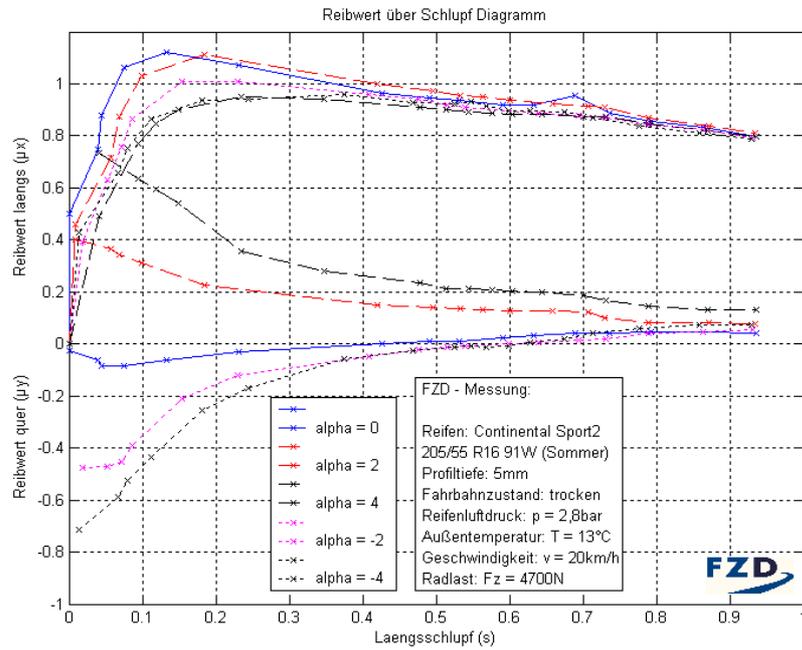


Abbildung 10.7: alle Kurven 20km/h

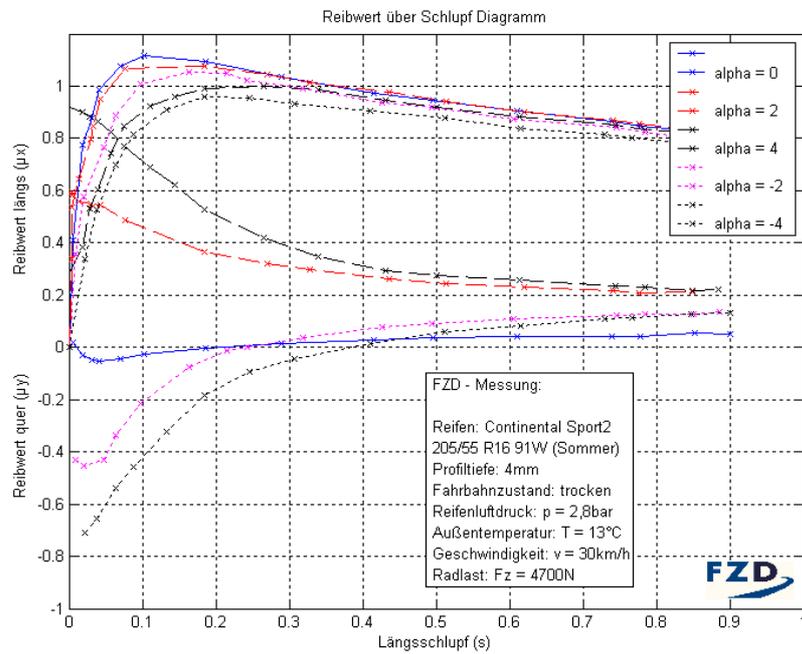


Abbildung 10.8: alle Kurven 30km/h

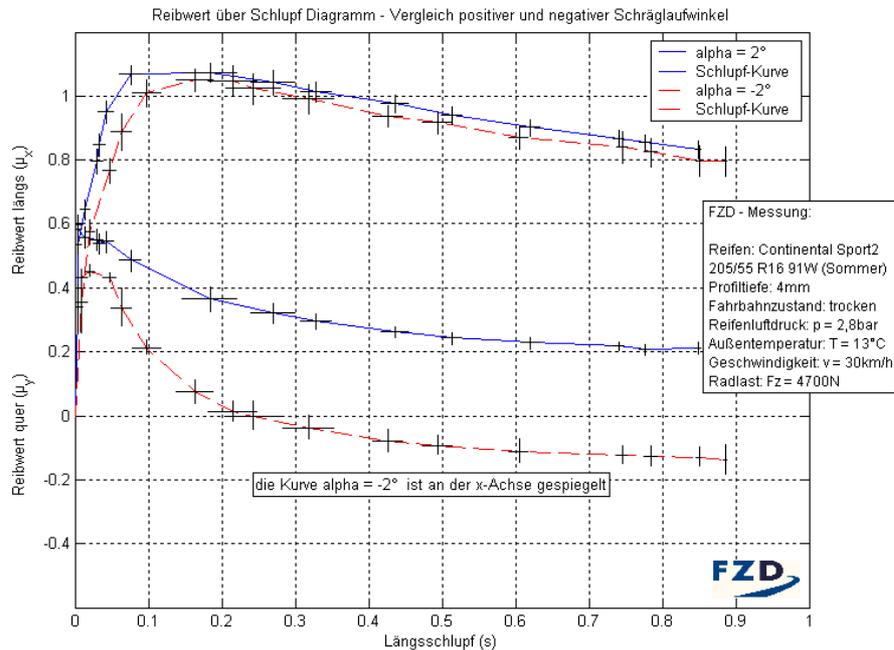


Abbildung 10.9: Vergleich der μ -Schlupf-Kurven von $\pm 2^\circ$ bei $v = 30\text{km/h}$

Die obige Abbildungen 10.9 zeigt den Vergleich zwischen Schräglauf -2° und 2° bei der Messgeschwindigkeit von 30km/h . Dies ist jedoch nicht der Fall. Allerdings kann, nach Overhoff³⁰, durch die statische Seitenkraft die sich aus Konus- und Winkeleffekt zusammensetzt ein Reifen eine Seitenkraft bei $\alpha = 0^\circ$ aufbauen. Konus- und Winkeleffekt weist normalerweise jeder Reifen auf und kann durch die Reifenfertigung bewusst beeinflusst werden, vgl. Overhoff³¹.

³⁰ Overhoff, Grundlagen Reifentechnologie, 2002/2003, Kapitel 6, S.4

³¹ Overhoff, Grundlagen Reifentechnologie, 2002/2003, Kapitel 4.10, S.8ff

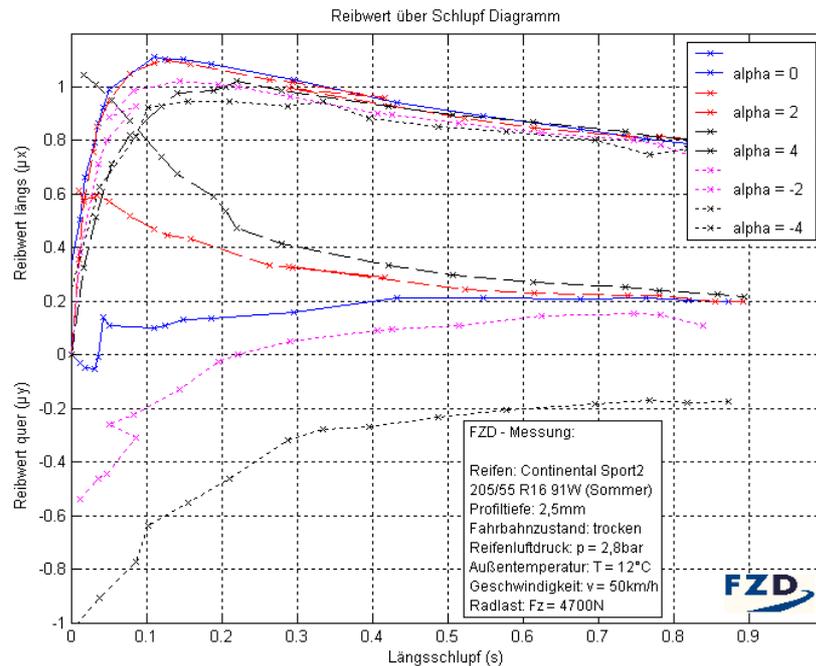


Abbildung 10.10: alle Kurven 50km/h

Die Messungen bei $v = 50\text{km/h}$ waren schwierig zu fahren, da die Geschwindigkeit nicht schnell genug kompensiert werden konnte nach Einstellung einer höheren Bremsmomentstufe. Ob sich allerdings allein dadurch die großen Schlupffehlerbereiche eines Punktes auf der μ -Schlupf-Kurve erklären lassen ist fraglich. Um dies genauer untersuchen zu können müssten Vergleichsmessungen mit gleichen Bedingungen und einem stärker motorisierten Fahrzeug gefahren werden. Die Ecken in den μ -Schlupf-Kurven 2° und -2° bei 50km/h lassen sich durch das Rausfallen aus der Messung begründen. Der Verlauf der μ -Schlupf-Kurve bei 0° lässt vermuten, dass auch hier ein „vom Gas gehen“ für den Knick verantwortlich ist. Bei der μ_{quer} -Schlupf-Kurve ($v = 50\text{km/h}$ und $\alpha = -4^\circ$) fällt auf, dass diese im Vergleich zu den anderen μ_{quer} -Schlupf-Kurven bei $v = 50\text{km/h}$ sich einem anderen μ_{gleit} nähert. Dies lässt vermuten, dass bei der vorherigen F_y -Offset-Bestimmung die für die letzte Messung erneut durchgeführt wurde, da die Radlenkwinkelverstellung am Messrad hängen blieb, ein Fehler gemacht wurde.

Auffallend ist außerdem, dass das μ der $\mu_{\text{längs}}$ -Schlupf-Kurve 2° annähernd so groß ist wie das der 0° Kurve und im höheren Schlupfbereich sogar darüber liegt. Hier bleibt zu klären, ob dies auf die großen Fehlerbereiche zurückzuführen ist oder ob vielleicht das Messrad bei Schräglaufwinkel 0° nicht Schräglaufwinkel frei ist. Aus diesem Grund kann die Validierung nicht als 100% erfüllt gelten, da der Grund für diese Abweichung noch zu klären ist.

11 Fazit und Ausblick

Die Software PAMELA ist umgesetzt, verifiziert und validiert. Dabei sind folgende Funktionen umgesetzt:

- automatische R_{dyn} -Bestimmung
- automatisierte Messungsdurchführung
 - alle Messgrößen werden mit eingestellter Abtastrate erfasst
 - alle Kontrollgrößen werden erfasst
- das Gesamtsystem ist hinsichtlich möglichst hoher Datenrate optimiert
 - $\text{Abtastrate}_{\text{max}} = 6000 \text{ Hz}$ bei 13 Kanäle
- Schräglaufwinkel wird eingestellt
- Bremsmoment wird automatisch in Stufen oder kontinuierlich erhöht
- automatische Schwimmwinkelkompensation
- übersichtliche, benutzer- und konfigurationsfreundliche, grafische Benutzerschnittstelle

Die Detailanalyse der Messdaten steht noch aus, hierzu sind ggf. noch weitere Versuche nötig. Außerdem bleibt zu klären warum die R_{dyn} -Berechnung über das Bremsmoment (F_y) und die Bremskraft (F_x) in der jetzigen Implementierung nicht verwendbar ist. Zudem bleiben weitere Optimierungsmaßnahmen die die Performance des Systems erhöhen können. Diese werden im Folgenden aufgezählt und erläutert:

Timer weiter verringern

Wenn die sequentielle Abarbeitung beibehalten wird, ist eine weitere Optimierung durch Verringerung der Schleifentimer auf dem Broker möglich. Da in der 3. Stufe, wie der Tabelle 8-1 aus Kapitel 8.1.5 zu entnehmen ist, der HPL-Timer bereits an dem Minimum von 1ms angeht, was dem maximalen Schleifentimer von 1000Hz entspricht, ist eine Abtastratenerhöhung nur durch weiteres Verringern der LPL-Timer möglich. Ein weiteres Verringern des LPL-Timer Up erfordert, dass die Abtastrate der Kontrollgrößen weiter erhöht wird. Dies führt zu Problemen wie z.B. zu kleine Puffergröße und längere Verarbeitungszeit, da mehr Daten verarbeitet werden. Daher ist nicht sicher ob ein weiteres Verringern dieses Timers zu einer Erhöhung der Messdatenabtastrate führt. Eine weitere Verringerung des LPL-Timer Down hat vermutlich einen deutlicheren Vorteil auf die Abtastratenerhöhung der Messdaten, da eine Verringerung des LPL-Timer Down dafür sorgt, dass bei der sequentiellen Abarbei-

tung die HPL-Schleife früher erneut ausgeführt wird. Das Verringern der beiden LPL-Timer widerspricht aber dem eigentlichen Vorhaben, die Schleifen selten auszuführen, um mehr Rechenzeit für die HPL-Schleife zur Verfügung zu stellen. Daher wird empfohlen eine weitere Optimierung durch Multithreading einzuleiten.

Multithreading aktivieren

Aktivierung des Multithreading auf dem CRIO, welches standardmäßig auf einen Thread eingestellt ist. Ist das Multithreading aktiviert, weist LabVIEW parallelen Schleifen automatisch separate Threads zu. Multithreading lässt sich aktivieren durch Setzen der Variablen E-Sys.StdNParallel auf 1 innerhalb der Datei ni-rt.ini auf dem RT-System. Dies führt allerdings dazu, dass keine Supportleistung durch den NI-Support nach Aktivierung des Multithreading erbracht wird. Da die komplette Programmierung von Anfang an auf Multithreading ausgelegt ist, bringt das Aktivieren des Multithreading zusätzliche Performance, da die 2 LPL Schleifen im TCL-Modul wieder langsamer getaktet werden können wie die HPL-Schleife. Dies erfordert auch ein Anpassen der Timerstufen. Allerdings werden beim Verringern der Timer die NPL-Timer nicht mitverringert, da diese immer mit der selben Frequenz betrieben werden können, da diese nun einen eigenen Thread haben. Somit wird beim Optimieren der Timer nur noch der HPL-Timer vermindert wenn die Abtastrate der Messgrößen erhöht wird. Dadurch wird das Gesamtsystem einfacher zu optimieren, da deutlich weniger Nebeneffekte zu beachten sind. Nach Aktivierung des Multithreading und neuer Optimierung der Timer, ist eine deutlich höhere maximale Frequenz zu erwarten.

Paketdurchsatz pro HPL Schleifendurchlauf erhöhen

Durch weiteres Erhöhen der HPL-Arraygröße lässt sich der Paketdurchsatz pro Schleifendurchlauf und damit auch die Frequenz erhöhen. Mit den Nachteilen eines höheren Delays und einer höheren Pufferüberlaufwahrscheinlichkeit, da die gleiche Puffergröße bei Erhöhung der Paketgröße weniger Elemente fassen kann.

Codeoptimierung

Durch Optimierung des Codes lässt sich die Performance weiter steigern. Allerdings stellt sich hier die Frage, welche Optimierungen der Compiler bereits vornimmt. Daher resultiert eine vermeintliche Codeoptimierung nicht unbedingt in einer schnelleren Ausführung.

Fehlerbehandlung abschalten

Eine weitere Möglichkeit die Abarbeitungszeit zu verkürzen ist auf Fehlerbehandlung zu verzichten. Besonders auf dem RT-System lässt sich durch Entfernen aller Fehlerbehandlungsverbindungen und -elemente die Ausführungszeit verringern, wenn diese durch andere Maßnahmen nicht weiter verbessert werden kann.

Messgrößen reduzieren

Eine Möglichkeit die deutliche Performanceerhöhung mit sich bringt ist die Verringerung der Messgrößen welche mit voller Abtastrate erfasst werden. Hierbei kann als erstes auf den Sinus und Kosinus der Kistler-Messfelge verzichtet werden, die nur übertragen werden um einen möglichen Temperaturdrift erkennen zu können, bzw. für den Nachweis der korrekten Zerokalibrierung übertragen werden, da sin- und cos-Verläufe bei nicht korrekter Zerokalibrierung auch in F_y zu erkennen sind. Des Weiteren ist es möglich, 2 verschiedene Softwareversionen zu entwickeln, abhängig davon ob die Messfelge zur Datenerfassung verwendet wird oder die Messdose. Bei Verwendung der Messfelge kann dabei zwar nur auf das Messdosensignal, bei Nichtverwendung allerdings auf alle Messfelgensignale verzichtet werden. Was in letzterem Fall zu einem deutlichen Performancegewinn führt.

12 Literaturverzeichnis

- Michelin Reifenwerke kGaA [Der Reifen - Komfort, 2005]:
Der Reifen – Komfort mechanisch und akustisch; 1. deutsche Auflage, Michelin Reifenwerke kGaA Öffentlichkeitsarbeit, Köln, 2005
- Sommerville, Ian [Software Engineering, 2001]:
Software Engineering; 6. Auflage, Pearson Education Deutschland GmbH, München, 2001
- Spillner, Andreas
Linz, Tilo [Basiswissen Softwaretest, 2003]:
Basiswissen Softwaretest, Aus- und Weiterbildung zum Certified Tester Foundation Level nach ASQF- und ISEB-Standard; 1. Auflage, dpunkt-Verlag, Heidelberg, 2003
- Reimpell, Jörnßen [Fahrwerktechnik Grundlagen, 1995]:
Fahrwerktechnik Grundlagen, Vogel Buchverlag, 3. überarbeitete Auflage, Würzburg, 1995
- Birkhofer, Herbert [Skriptum Produktentwicklung I, 2002/2003]:
Skriptum zur Vorlesung Produktentwicklung I; pmd, Darmstadt, Wintersemester 2002/2003
- Overhoff, Dietrich [Skriptum Grundlagen Reifentechnologie, 2002/2003]:
Skriptum Grundlagen Reifentechnologie; fzd, Darmstadt, Wintersemester 2002/2003
- Winner, Hermann [Skriptum Kraftfahrzeuge II, 2006]
Skriptum Kraftfahrzeuge II; fzd, Darmstadt, Sommersemester 2006
- National Instruments [LabVIEW Hilfe, 2006]
Hilfe von LabVIEW inklusive LabVIEW FPGA; Version 8.20, 2006
- Wolf, Ingemar [Diplomarbeit 413/06]
Diplomarbeit 413/06: Entwicklung und Verifikation eines Sicherheitskonzepts für einen Messanhänger zur Ermittlung des Reibverhaltens von PKW-Reifen; fzd, Darmstadt, 2007

Internetquellen:

- National Instruments [CRIO, 2007]:
Abbildung des CRIO;
http://sine.ni.com/images/products/us/040730_crioembedded_1.jpg, 2007, Zugriff 20.03.2007
- National Instruments [Datenblatt CRIO-9104, 2007]:
Datenblatt des Gehäuses CRIO 9104;
http://www.ni.com/pdf/products/us/cat_crio_9104.pdf, 2007, Zugriff 20.03.2007
- National Instruments [Datenblatt CRIO-9004, 2007]:
Datenblatt des CRIO-9004 Real-Time Embedded Controllers;
http://www.ni.com/pdf/products/us/6358_crio_rt_controllers.pdf, 2007, Zugriff 20.03.2007
- National Instruments [Bedienungsanleitung NI-9205, 2007]:
Operation Instruction NI-9205;
<http://www.ni.com/pdf/manuals/374188c.pdf>, 2007, Zugriff 20.03.2007
- National Instruments [Bedienungsanleitung NI-9263, 2007]:
Operation Instruction NI-9263;
<http://www.ni.com/pdf/manuals/373781c.pdf>, 2007, Zugriff 20.03.2007
- National Instruments [Bedienungsanleitung NI-9401, 2007]:
Operation Instruction NI-9401;
<http://www.ni.com/pdf/manuals/374068c.pdf>, 2007, Zugriff 20.03.2007
- National Instruments [Bedienungsanleitung NI-9472, 2007]:
Operation Instruction NI-9472/9474;
<http://www.ni.com/pdf/manuals/373509d.pdf>, 2007, Zugriff 20.03.2007
- Spillner, Andreas [W-Modell, 2005]:
Das W-Modell – Testen als paralleler Prozess zum Software-Entwicklungsprozess; <http://www.informatik.hs-bremen.de/spillner/WModell/WModell.pdf>, 2006, Zugriff 6.12.2006

Wikipedia

[LabVIEW, 2007]:

Eintrag zum Thema LabVIEW;

<http://de.wikipedia.org/wiki/LabVIEW>, 2007, Zugriff

21.03.2007

13 Abbildungsverzeichnis

Abbildung 1.1: Reibwert-Schlupf-Kurven aus Winner ¹	1
Abbildung 1.2: W-Modell (Erweitertes V-Modell)	6
Abbildung 2.1: PETRA I auf dem Flugplatz in Griesheim	7
Abbildung 2.2: CRIO	11
Abbildung 3.1: Frontpanel eines VIs	14
Abbildung 3.2: Blockdiagramm eines VIs mit der Funktionalität der Mittelwertbildung	15
Abbildung 4.1: Ablaufszenario Reibwert-Schlupf-Kurven messen	18
Abbildung 5.1: Softwaresystemarchitektur	24
Abbildung 5.2: FPGA-Feinentwurf	28
Abbildung 5.3: Broker-Feinentwurf	29
Abbildung 5.4: Host-Feinentwurf	31
Abbildung 6.1: Zeichenerklärung der verwendeten Symbole	33
Abbildung 6.2: FPGA-Konfiguration – Modulspezifikation	33
Abbildung 6.3: Entscheidungsbaum der Schräglaufwinkeleinstellung	35
Abbildung 6.4: Schräglaufeinstellung – Modulspezifikation	36
Abbildung 6.5: Beziehung der Winkel α , β , δ	37
Abbildung 6.6: Schwimmwinkelkompensation – Modulspezifikation	39
Abbildung 6.7: Drehzahlberechnung – Modulspezifikation	41
Abbildung 6.8: Low Priority Loop – Modulspezifikation	43
Abbildung 6.9: High Priority Loop – Modulspezifikation	44
Abbildung 6.10: Normal Priority Loop – Modulspezifikation	45
Abbildung 6.11: Time Critical Loop – Modulspezifikation	46
Abbildung 6.12: Hauptprogramm – Modulspezifikation	47
Abbildung 6.13: Einstellungen – Modulspezifikation	47
Abbildung 6.14: Kalibrierung – Modulspezifikation	49
Abbildung 6.15: R_{dyn} -Berechnung – Modulspezifikation	51
Abbildung 6.16: Messung – Modulspezifikation	52
Abbildung 6.17: Vorauswertung – Modulspezifikation	53
Abbildung 6.18: Statusanzeigen – Modulspezifikation	54
Abbildung 6.19: Warnung – Modulspezifikation	55
Abbildung 7.1: Konfigurationsmodul	57
Abbildung 7.2: Mittelwertbildung	58
Abbildung 7.3: Schräglaufeinstellung	59
Abbildung 7.4: Schwimmwinkelkompensation	60

Abbildung 7.5: Drehzahlberechnungsmodul	60
Abbildung 7.6: LPL-Down Modul	62
Abbildung 7.7: LPL-Up Modul	63
Abbildung 7.8: HPL-Modul	64
Abbildung 7.9: NPL: Erstellung der Realtime-FIFOs	65
Abbildung 7.10: NPL: Sende/Empfange Daten	66
Abbildung 7.11: TCL: FPGA referenzieren und starten	67
Abbildung 7.12: TCL: Kalibrierdaten übertragen und DMA konfigurieren	68
Abbildung 7.13: TCL: High Priority Loop	69
Abbildung 7.14: TCL: Low Priority Loop – Up	72
Abbildung 7.15: TCL: Low Priority Loop – Down, oberer Teil	73
Abbildung 7.16: TCL: Low Priority Loop – Down unterer Teil	74
Abbildung 7.17: Umrechnung Volt-Integer Abbildung 7.18: Correvitkalibrierdaten in Integer	77
Abbildung 7.19: R_{dyn} -Berechnung	80
Abbildung 7.20: Messung – Messdurchführung	82
Abbildung 7.21: Warnung	83
Abbildung 9.1: Integrationsvorgehen	98
Abbildung 9.2: Faktor $_{v, 30km/h}$	104
Abbildung 9.3: Faktor $_{v, 60km/h}$	105
Abbildung 9.4: Offset $_{\alpha 30 km/h}$	106
Abbildung 9.5: Offset $_{\alpha 60 km/h}$	106
Abbildung 9.6: Veränderung der Kontrollgrößen U_{Bord} , T_{CRIO} , $T_{Hydraulik}$	107
Abbildung 9.7: $p_{Hydraulik}$ bei 30km/h	108
Abbildung 9.8: $p_{Hydraulik}$ bei 40km/h	109
Abbildung 9.9: v_{Fzg}, v_{Rad} über Zeit	110
Abbildung 9.10: v_{Fzg} über v_{Rad}	110
Abbildung 9.11: Bremskraftverlauf bei Proportionalventilerhöhung	112
Abbildung 9.12: Vergleich $R_{dyn,gemessen}$ mit $R_{dyn,manuell}$	114
Abbildung 9.13: R_{dyn} berechnet	115
Abbildung 9.14: dynamische Messung	116
Abbildung 9.15: stationäre Messung 20	117
Abbildung 9.16: stationäre Messung 30	117
Abbildung 9.17: stationäre Messung 50	118
Abbildung 10.1: Möglichkeiten zur Auswertung der Messdaten	123
Abbildung 10.2: μ -Schlupf-Rohdaten bei $v = 20km/h$ und $\alpha = 0^\circ$	124
Abbildung 10.3: μ -Schlupf-Rohdaten bei $v = 30km/h$ und $\alpha = 0^\circ$	125
Abbildung 10.4: μ -Schlupf-Rohdaten bei $v = 50km/h$ und $\alpha = 0^\circ$	125
Abbildung 10.5: μ -s-Kurve ($v=30km/h \alpha=0^\circ$)	126
	138

Abbildung 10.6: μ-s-Kurve ($v=30\text{km/h}$ $\alpha =4^\circ$)	127
Abbildung 10.7: alle Kurven 20km/h	128
Abbildung 10.8: alle Kurven 30km/h	128
Abbildung 10.9: Vergleich der μ-Schlupf-Kurven von $\pm 2^\circ$ bei $v = 30\text{km/h}$	129
Abbildung 10.10: alle Kurven 50km/h	130

14 Tabellenverzeichnis

Tabelle 7-1: Steuerung der in Kapitel 7.2.2.6.2 genannten Timer vom Host aus	80
Tabelle 8-1: Timerstufen der Schleifen des Brokers	86
Tabelle 10-1: Validierungsbedingungen	121
Tabelle 10-2: Stufeneinstellung der Validierung	122

Anhang A Anforderungsdefinition

A1. Funktionale Anforderungen

Funktion	Sensordatenerfassung
Anforderungsart	Festforderung
ID	001
Priorität	hoch
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	gering
Beschreibung	Erfasst die an den Sensoren anliegenden, analogen und digitalen Signale und wandelt diese in weiterverarbeitbare Daten.
Eingaben	analoge Signale zwischen -10V und 10V digitale TTL-Signale (0V = '0' / 5V = '1')
Quelle	alle Sensoren
Ziel	Die Signale liegen in einem weiterverarbeitbaren Dateiformat vor.

Funktion	Aktoransteuerung
Anforderungsart	Festforderung
ID	002
Priorität	hoch
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	gering
Beschreibung	Ansteuerung der Aktoren, analog: Proportionalverstärker der Axialkol-

	benpumpe, digital: Motoren für Radlenkwinkel des Messrades und des Referenzrades je Rechts-/Linkslauf
Ausgabe	analoges Signal zwischen 12 – 17V digitale TTL-Signale (0V = '0' / 5V = '1')
Senke	alle Aktoren
Ziel	Die Aktoren führen die entsprechende Aktion aus.

Funktion	Schwimmwinkelkompensation
Anforderungsart	Zielforderung
ID	003
Priorität	hoch
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	mittel
Beschreibung	Ausregelung des Schwimmwinkels. Ein Schwimmwinkel kann sich einstellen durch die Asymmetrie des Messanhängers, durch Abbremsung des Messrades, durch Seitenwind und durch Einstellung eines Radlenkwinkels des Messrades zur Schräglaufwinkelerzeugung. Die Ausregelung erfolgt über den Radlenkwinkel des Referenzrades.
Eingabe	Schräglaufwinkel des Messrades Radlenkwinkel des Messrades Radlenkwinkel des Referenzrades
Quelle	Winkel aus Correvit Potentiometer der Radlenkwinkelverstellung am Messrad Potentiometer der Radlenkwinkelverstellung am Referenzrad
Ausgabe	Veränderung des Radlenkwinkels des Referenzrades
Senke	Motor zur Radlenkwinkelverstellung am Referenzrad
Ziel	Der Schwimmwinkel wird minimiert.

Funktion	Drehzahlberechnung
Anforderungsart	Festforderung
ID	004
Priorität	hoch
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	mittel
Beschreibung	Das vom Drehzahlsensor am Messrad erzeugte Signal wird in eine Drehzahl gewandelt.
Eingabe	Drehzahlsignal
Quelle	Inkrementalgeber
Ziel	Die Drehzahl wird berechnet.

Funktion	Bremsmomentverstellung
Anforderungsart	Festforderung
ID	005
Priorität	hoch
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	mittel
Beschreibung	Verstellmöglichkeit des Bremsmomentes am Referenzrad bereitstellen. Die Verstellung geschieht über das Proportionalventil. Das Bremsmoment kann nur verstellt werden wenn der Hydraulikdruck größer 40 bar ist. Dabei soll während einer Messung der Proportionalverstärker schrittweise verändert werden können. Je nach Verstellung bzw. Schrittweite ändert sich das Bremsmoment.
Ausgabe	Veränderung der Stellung des Proportionalventils
Senke	Proportionalverstärker

Ziel	Das Bremsmoment am Messrad wird verstellt.
------	--

Funktion	Anzeige von Kontrollparametern
Anforderungsart	Festforderung
ID	006
Priorität	hoch
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	mittel
Beschreibung	Zur Überwachung wichtiger Kontrollparameter wie Öldruck, Öltemperatur, CRIO-Gehäusetemperatur, Bordspannung und die 4 Endschalter der Radlenkwinkelverstellung (links/rechts).
Ausgabe	Anzeige auf Display
Senke	Benutzer
Ziel	Der Benutzer wird über Kontrollparameter informiert.

Funktion	Anzeige von Statusparametern
Anforderungsart	Zielforderung
ID	007
Priorität	mittel
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	mittel
Beschreibung	Informieren des Benutzers über aktuelle Statusinformationen des Messanhängers. Folgende Statusparameter sollen angezeigt werden: Geschwindigkeit des Anhängers, Geschwindigkeit des Messrades, Sollschräglaufwinkel des Messrades, Schräglaufwinkel des Messrades, Schwimmwinkel des Anhängers. Das System soll ein großes Ge-

	schwindigkeitsdisplay haben, damit der Fahrer bei der Messfahrt einfacher die Geschwindigkeit ablesen und konstant halten kann.
Ausgabe	Anzeige auf Display
Senke	Benutzer
Ziel	Der Benutzer wird über möglichst viele Statusparameter informiert.

Funktion	Anzeige weiterer Statusparameter
Anforderungsart	Wunsch
ID	008
Priorität	gering
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	mittel
Beschreibung	Informieren des Benutzers über weitere Statusinformationen des Anhängers. Folgende Statusparameter können angezeigt werden: Stufe des Proportionalverstärkers, Radlenkwinkel des Messrades, Radlenkwinkel des Referenzrades, Schlupf, μ_{quer} , $\mu_{\text{längs}}$.
Ausgabe	Anzeige auf Display
Senke	Benutzer
Ziel	Der Benutzer wird über weitere Statusparameter informiert.

Funktion	Warnung beim Verlassen von Parametergrenzen
Anforderungsart	Festforderung
ID	009
Priorität	hoch
Version	2
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf

Aufwandsschätzung	hoch
Beschreibung	<p>Warnen des Benutzers bei über- oder unterschreiten von festgelegten Parametergrenzen. Wird eine der festgelegten Parametergrenzen über- oder unterschritten, warnt das System den Benutzer mit einer entsprechenden Anzeige und ggf. Warnton.</p> <p>Folgende Parametergrenzen werden geprüft, dabei steht die erste Grenze für die 1. Warnung und die zweite Grenze für die 2. Warnung:</p> <p>Hydraulikdruck > 350 bar / > 390 bar</p> <p>Hydrauliktemperatur > 75 °C / > 80 °C</p> <p>Bordspannung < 11,5 V / < 11 V</p> <p>CRIO-Gehäusetemperatur > 60 °C / > 64 °C</p>
Anmerkung	<p>11,5V Bordspannung ó 6,4V am analog Eingang</p> <p>11V Bordspannung ó 6,1V am analog Eingang</p>
Ausgabe	Anzeige auf Display, ggf. über Lautsprecher
Senke	Benutzer
Ziel	Der Benutzer wird beim Verlassen von Parametergrenzen gewarnt.

Funktion	Herunterfahren der Messtechnik
Anforderungsart	Festforderung
ID	010
Priorität	hoch
Version	2
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	hoch
Beschreibung	<p>Bei Erreichen folgender, kritischer Parametergrenzen, Messdatenaufzeichnung abbrechen, Messdaten sichern und Messhardware herunterfahren:</p> <p>Bordspannung $\leq 10,5$ V</p> <p>CRIO-Gehäusetemperatur ≥ 68 °C</p>
Anmerkung	10,5V Bordspannung ó 5,9V am analog Eingang

Ausgabe	CRIO-Messtechnik abschalten
Senke	CRIO-Messtechnik
Ziel	Die Messtechnik wird rechtzeitig und kontrolliert heruntergefahren um Schädigung an der Messtechnik und Messdatenverlust zu vermeiden.

Funktion	Benutzerschnittstelle
Anforderungsart	Festforderung
ID	011
Priorität	gering
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	mittel
Beschreibung	Bereitstellung einer Schnittstelle zwischen Benutzer und Software. Eine intuitive, benutzerfreundliche Benutzerschnittstelle ist erwünscht.
Eingabe	Eingabe über Tastatur und Maus
Quelle	Benutzer
Ausgabe	Anzeige auf Display
Senke	Benutzer
Ziel	Eine Kommunikationsmöglichkeit zwischen dem Benutzer und der Software wird bereitgestellt.

Funktion	Konfigurationsmöglichkeit
Anforderungsart	Festforderung
ID	012
Priorität	mittel
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf

Aufwandsschätzung	mittel
Beschreibung	Möglichkeit zur Eingabe folgender Konfigurationsdaten: Messdateinamen, Verwendung der Messfelge, statische Radlast, Geschwindigkeitstoleranz, Schräglaufwinkeltoleranz, Schwimmwinkeltoleranz, Reifeneigenschaften, ermittelte Werte wie Reifendruck, Reifentemperatur und Asphalttemperatur. Diese Daten werden dann in einer Konfigurationsdatei gespeichert. Es ist erforderlich, dass die Datei um weitere Größen erweiterbar ist.
Eingabe	Eingabe über Tastatur und Maus
Quelle	Benutzer
Ausgabe	Speicherung in Datei
Senke	Dateisystem
Ziel	Die Konfigurationsdaten können direkt in der Software abgelegt werden.

Funktion	Zero-Kalibrierung
Anforderungsart	Festforderung
ID	013
Priorität	mittel
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	mittel
Beschreibung	Da die Verwendung der Kistler Messfelge mit der Zeit größere Driftfehler erzeugt, soll an der Messfelge nach jeder Messung eine Nullkalibrierung durchgeführt werden können. Es ist eine Möglichkeit vorzusehen die Kalibrierung per Knopfdruck durchzuführen.
Eingabe	Eingabe über Tastatur und Maus
Quelle	Benutzer
Ausgabe	Messfelgenkalibrierung aktiviert
Senke	Messfelge

Ziel	Die Messfelge führt eine Nullkalibrierung durch.
------	--

Funktion	Kalibrierung
Anforderungsart	Wunsch
ID	014
Priorität	gering
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	mittel
Beschreibung	Eingabemöglichkeit für die Kalibrierung der 2 Potentiometer für Radlenkwinkelverstellung am Messrad und am Referenzrad, sowie die Kalibrierung für die Kraftmessdose.
Eingabe	Eingabe über Tastatur und Maus
Quelle	Benutzer
Ziel	Die Kalibrierungsdaten können über eine Benutzereingabe geändert werden.

Funktion	R_{dyn} -Berechnung
Anforderungsart	Festforderung
ID	015
Priorität	hoch
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	hoch
Beschreibung	Es wird eine Möglichkeit benötigt, den dynamischen Reifenhalmmesser (R_{dyn}) dynamisch vor einer Messung, zu berechnen. Eine mögliche Mess- und Berechnungsdurchführung könnte wie folgt aussehen: Der Benutzer gibt an bei welcher Geschwindigkeit er die R_{dyn} Berech-

	<p>nung durchführen möchte. Nach Bestätigung beschleunigt der Fahrer das Fahrzeug auf die entsprechende Geschwindigkeit und hält diese möglichst genau. Das System erkennt, wenn die Geschwindigkeit im Toleranzbereich liegt und beginnt die Messung, sobald zudem der Schräglaufwinkeltoleranzbereich eingehalten wird. Wurden genug Werte (100) erfasst wird die Messung gestoppt und die Berechnung gestartet. Hierbei wird der Mittelwert aus den ermittelten R_{dyn} Werten gebildet. Danach wird R_{dyn} ausgegeben und in eine Datei gespeichert.</p>
Eingabe	Eingabe über Tastatur und Maus
Quelle	Benutzer
Ausgabe	Anzeige auf Display, Speicherung in Datei
Senke	Dateisystem
Ziel	Der dynamische Reifenhalmmesser wird ermittelt und in den darauffolgenden Messungen verwendet.

Funktion	Messungsdurchführung
Anforderungsart	Festforderung
ID	016
Priorität	hoch
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	sehr hoch
Beschreibung	<p>Es wird eine Möglichkeit zur Messungsdurchführung benötigt. Eine mögliche Messdurchführung könnte wie folgt aussehen:</p> <p>Als erstes gibt der Benutzer den Schräglaufwinkel des Messrades ein. Danach gibt er an, bei welcher Geschwindigkeit die μ-Schlupf-Kurven gemessen werden sollen (10 - 80 km/h). Der Benutzer kann zwischen dynamischer Messung (alle Messdaten aufzeichnen) und statischer Messung (nur Messdaten innerhalb des Toleranzbereichs aufzeichnen) wählen. Nach Bestätigung wird der Radlenkwinkel des Messrades eingestellt. Es ist ratsam den Anhänger langsam rollen zu lassen, um das benötigte Motorenmoment der Verstellung zu reduzieren. Danach kann</p>

	die Messung beginnen. Der Fahrer beschleunigt auf die entsprechende Geschwindigkeit und hält diese. Innerhalb eines fest vorgegebenen Zeitintervalls, eines bestimmten Geschwindigkeitstoleranzbereichs und Schräglaufwinkeltoleranzbereichs wird die Aufzeichnung durchgeführt. Nach Ablauf des Zeitintervalls wird die Bremskraft erhöht. Diese Bremskraft kompensiert der Fahrer durch Gas geben. Sinkt die Geschwindigkeit unter den Toleranzbereich werden diese Messdaten nicht aufgezeichnet um riesige Datenmengen zu vermeiden. Nach der Messung drückt der Benutzer den Knopf „Messung beendet“, daraufhin wird die Messdatei geschlossen.
Eingabe	Eingabe über Tastatur und Maus
Quelle	Benutzer
Ausgabe	Anzeige auf Display
Senke	Benutzer
Ziel	Die Messung wird durchgeführt und die Messdaten werden erfasst.

Funktion	Messdatenaufzeichnung
Anforderungsart	Festforderung
ID	017
Priorität	hoch
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	mittel
Beschreibung	Während der Messungsdurchführung sind die Messdaten zu erfassen und in eine Messdatei zu speichern.
Ausgabe	Speicherung in Datei
Senke	Dateisystem
Ziel	Die erfassten Messdaten werden gespeichert.

Funktion	Messdatenpersistenzprüfung
Anforderungsart	Festforderung
ID	018
Priorität	hoch
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	mittel
Beschreibung	Nach der Messung prüfen der Messdaten auf Persistenz. D.h. es wird geprüft ob die Messdaten in die Datei geschrieben wurden.
Eingabe	Messdatei
Quelle	Dateisystem
Ziel	Die Persistenz der Messdaten ist sichergestellt.

Funktion	Vorauswertung der Messung
Anforderungsart	Wunsch
ID	020
Priorität	gering
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	hoch
Beschreibung	Um direkt nach der Messung einen ersten Eindruck von den gemessenen Daten zu bekommen und diese auf Plausibilität prüfen zu können, ist es wünschenswert, wenn das System eine Vorauswertung der Messung macht. So können grobe Messfehler vor der eigentlichen Auswertung erkannt werden und ggf. die Messung wiederholt werden. Erreicht werden kann dies über eine $\mu_{\text{längs}}$ -Schlupf-Kurve und ein μ_{quer} -Schlupf-Kurve, welche anhand ausgewählter Punkte erstellt wird.
Eingabe	Messdatei

Quelle	Dateisystem
Ausgabe	$\mu_{\text{l\"angs}}$ -Schlupf-Kurve, μ_{quer} -Schlupf-Kurve
Senke	Benutzer
Ziel	Die Messung wird durch den Benutzer anhand μ -Schlupf-Kurven auf Plausibilität geprüft.

A2. Nichtfunktionale Anforderungen

Funktion	Performance
Anforderungsart	Zielforderung
ID	021
Priorität	sehr hoch
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	sehr hoch
Beschreibung	Aufzeichnen und speichern der analogen Messdaten mit mindestens 1000 Hz.
Ziel	Die Messdatenaufzeichnung wird mit mindestens 1000 Hz durchgeführt.

Funktion	Benutzerfreundlichkeit
Anforderungsart	Wunsch
ID	022
Priorität	gering
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	mittel
Beschreibung	Benutzerfreundliche und intuitive graphische Benutzeroberfläche ist erwünscht. Die Benutzung der Software soll für eine studentische

Anforderungsdefinition

	Hilfskraft innerhalb kurzer Zeit erlernbar sein.
Ziel	Bedienungsfehler werden minimiert, Bedienung ist intuitiv und die Schulungsdauer wird minimiert.

Funktion	Kompatibilität zu Matlab
Anforderungsart	Festforderung
ID	023
Priorität	hoch
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	sehr gering
Beschreibung	Es ist sicherzustellen, dass die gespeicherten Messdaten mit Matlab eingelesen werden können.
Ziel	Die Messdatenauswertung wird in Matlab ermöglicht.

Funktion	Zuverlässigkeit
Anforderungsart	Zielforderung
ID	024
Priorität	mittel
Version	1
Autor	Marco Heimberger
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Aufwandsschätzung	hoch
Beschreibung	Das System soll sich möglichst stabil verhalten.
Ziel	Die Nichtverfügbarkeit des Systems wird minimiert.

Funktion	Speicherplatzanforderung
Anforderungsart	Zielforderung

ID	025
Priorität	mittel
Version	1
Reviewer	Dipl.-Ing. Marcus Reul, Cand.-Ing. Ingemar Wolf
Autor	Marco Heimberger
Aufwandsschätzung	hoch
Beschreibung	Das System soll mit möglichst wenig Speicherplatz auskommen.
Ziel	Der Speicherplatzbedarf wird minimiert.

Anhang B Ein- und Ausgängeübersicht

B1. Analogeingänge NI-9205

Kanal	Signal	Bemerkung	zeitkritisch
AI 0	Bordspannung Petra	±10V NRSE	nein
AI 1	Hydraulik Druck	±10V NRSE	nein
AI 2	Hydraulik Temperatur	±10V NRSE	nein
AI 3			
AI 4			
AI 5			
AI 6			
AI 7			
AI 8	Messfelge Kraft Fx	±5V NRSE / Kistler	ja
AI 9	Messfelge Kraft Fy	±5V NRSE / Kistler	ja
AI 10	Messfelge Kraft Fz	±5V NRSE / Kistler	ja
AI 11	Messfelge Moment Mx	±5V NRSE / Kistler	ja
AI 12	Messfelge Moment My	±5V NRSE / Kistler	ja
AI 13	Messfelge Moment Mz	±5V NRSE / Kistler	ja
AI 14	Messfelge sin	±5V NRSE / Kistler	ja
AI 15	Messfelge cos	±5V NRSE / Kistler	ja
AI 16	Fahrzeuggeschwindigkeit Correvit	±5V NRSE / Datron	ja
AI 17	Schräglaufwinkel Correvit	±5V NRSE / Datron	ja
AI 18			
AI 19			
AI 20	Radlenkwinkelverstellung MR	±10V DIFF +	nein
AI 21	Radlenkwinkelverstellung RR	±10V DIFF +	nein

AI 22	(Sturzverstellung MR)	$\pm 10V$ DIFF +	nein
AI 23	Kraftmessdose	$\pm 10V$ DIFF +	nein
AI 24			
AI 25			
AI 26			
AI 27			
AI 28	Radlenwinkelverstellung MR	$\pm 10V$ DIFF -	nein
AI 29	Radlenwinkelverstellung RR	$\pm 10V$ DIFF -	nein
AI 30	(Sturzverstellung MR)	$\pm 10V$ DIFF -	nein
AI 31	Kraftmessdose	$\pm 10V$ DIFF -	nein

B2. Analogausgänge NI-9263

Kanal	Signal	Bemerkung	zeitkritisch
AO 0	Proportionalverstärker Axialkolbenmaschine		nein
AO 1			
AO 2			
AO 3			

B3. Digitaleingänge NI-9401

Kanal	Signal	Bemerkung	zeitkritisch
DI 0	Drehzahl Inkrementalgeber MR	TTL / 1000Ink/U.	ja
DI 1			
DI 2	Endschalter Rechtslauf, Radlenwinkelverstellung MR	TTL	nein
DI 3	Endschalter Linkslauf,	TTL	nein

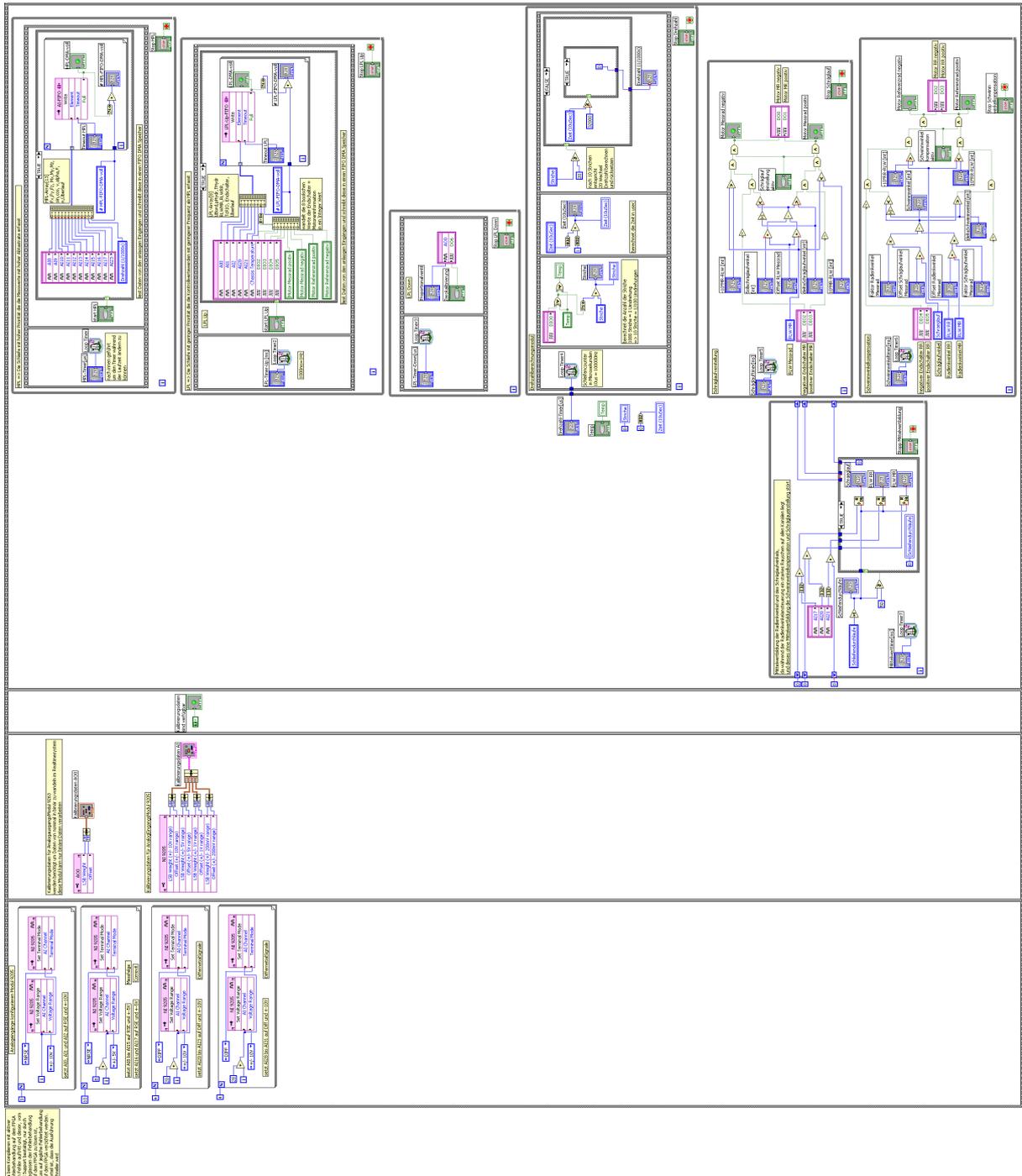
	Radlenkwinkelverstellung MR		
DI 4	Endschalter Rechtslauf, Radlenkwinkelverstellung RR	TTL	nein
DI 5	Endschalter Linkslauf Radlenkwinkelverstellung RR	TTL	nein
DI 6	(Endschalter Rechtslauf, Sturzverstellung MR)	TTL	nein
DI 7	(Endschalter Linkslauf, Sturzverstellung MR)	TTL	nein

B4. Digitalausgänge NI-9472

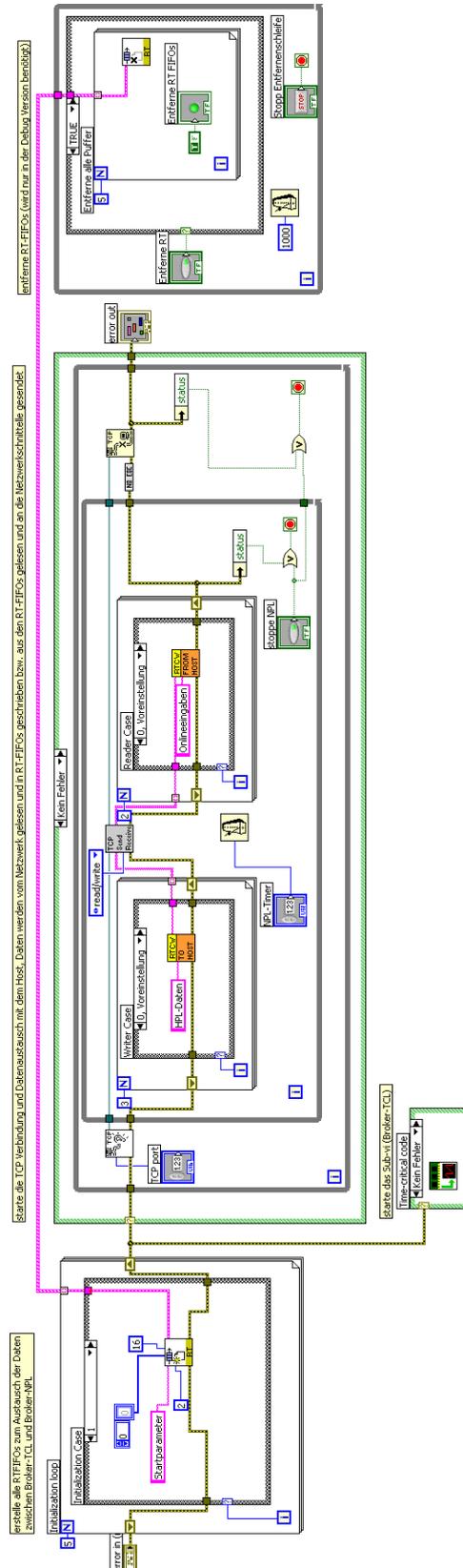
Kanal	Signal	Bemerkung	zeitkritisch
DO 0	Motor Rechtslauf, Radlenkwinkelverstellung MR		nein
DO 1	Motor Linkslauf, Radlenkwinkelverstellung MR		nein
DO 2	Motor Rechtslauf, Radlenkwinkelverstellung RR		nein
DO 3	Motor Linkslauf Radlenkwinkelverstellung RR		nein
DO 4	(Motor Rechtslauf, Sturzverstellung MR)		nein
DO 5	(Motor Linkslauf, Sturzverstellung MR)		nein
DO 6	Messfelge Zero Calibration	Kistler	nein
DO 7			

Anhang C Programmcodeübersicht

C1. FPGA-Übersicht



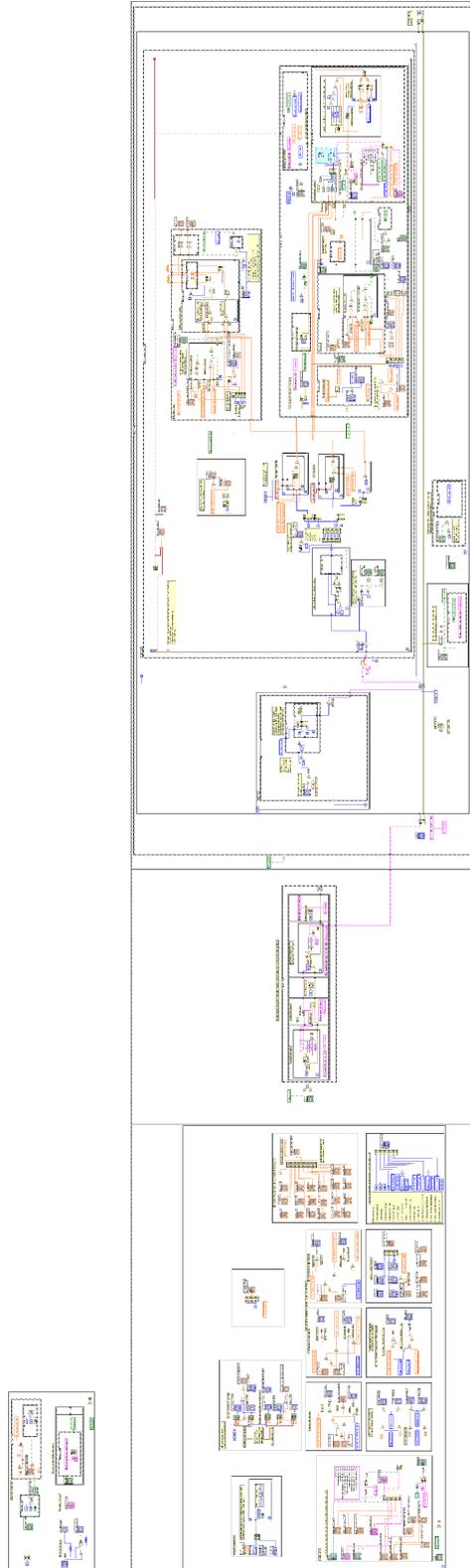
C2. Broker NPL – Übersicht



C3. Broker TLC – Übersicht



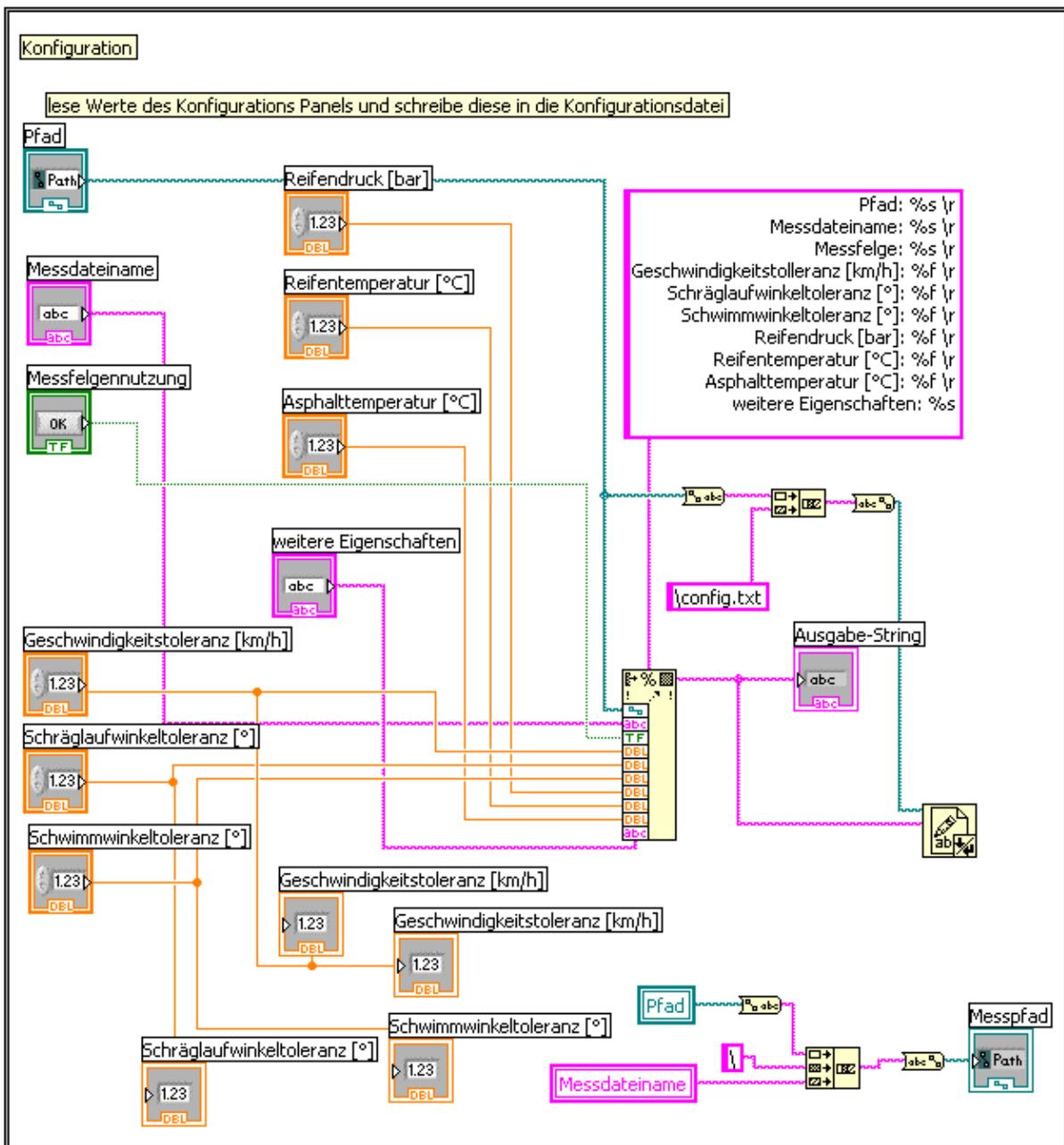
C4. Host – Übersicht



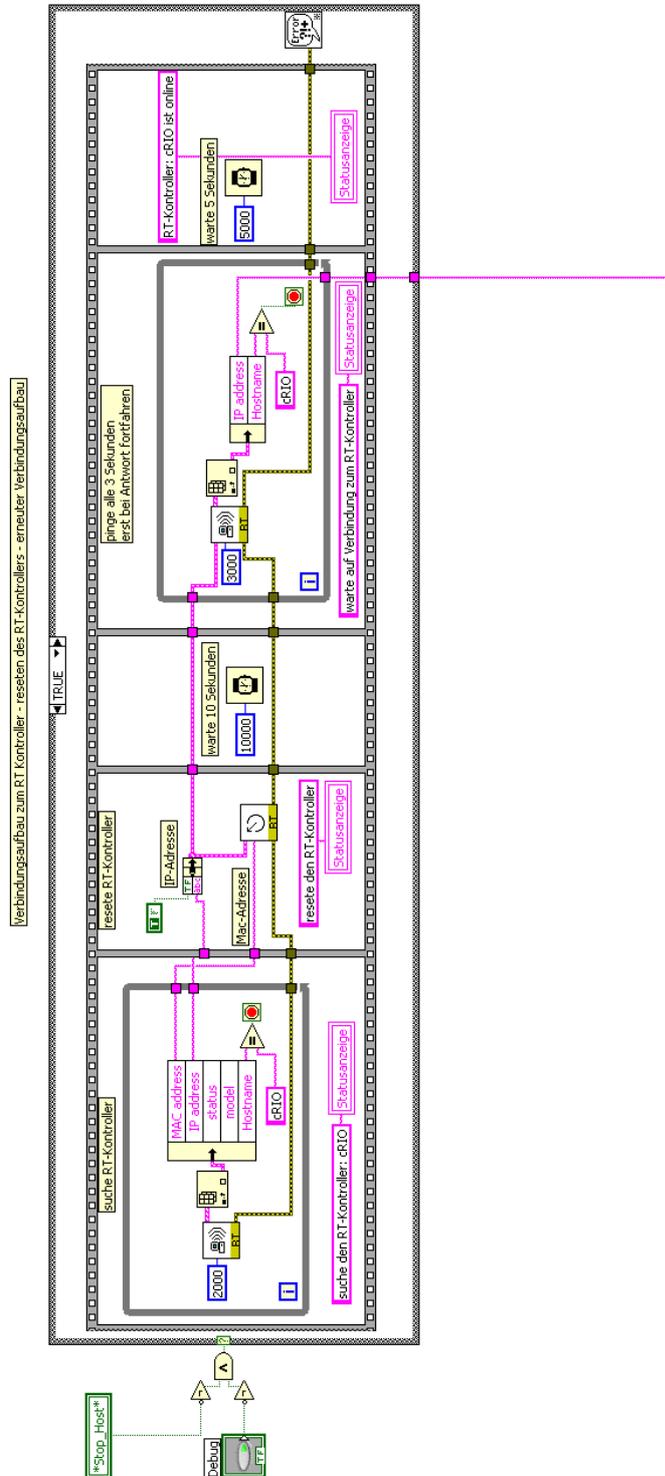
Anhang D Programmcodefragmente

D1. Host Programmcodefragmente

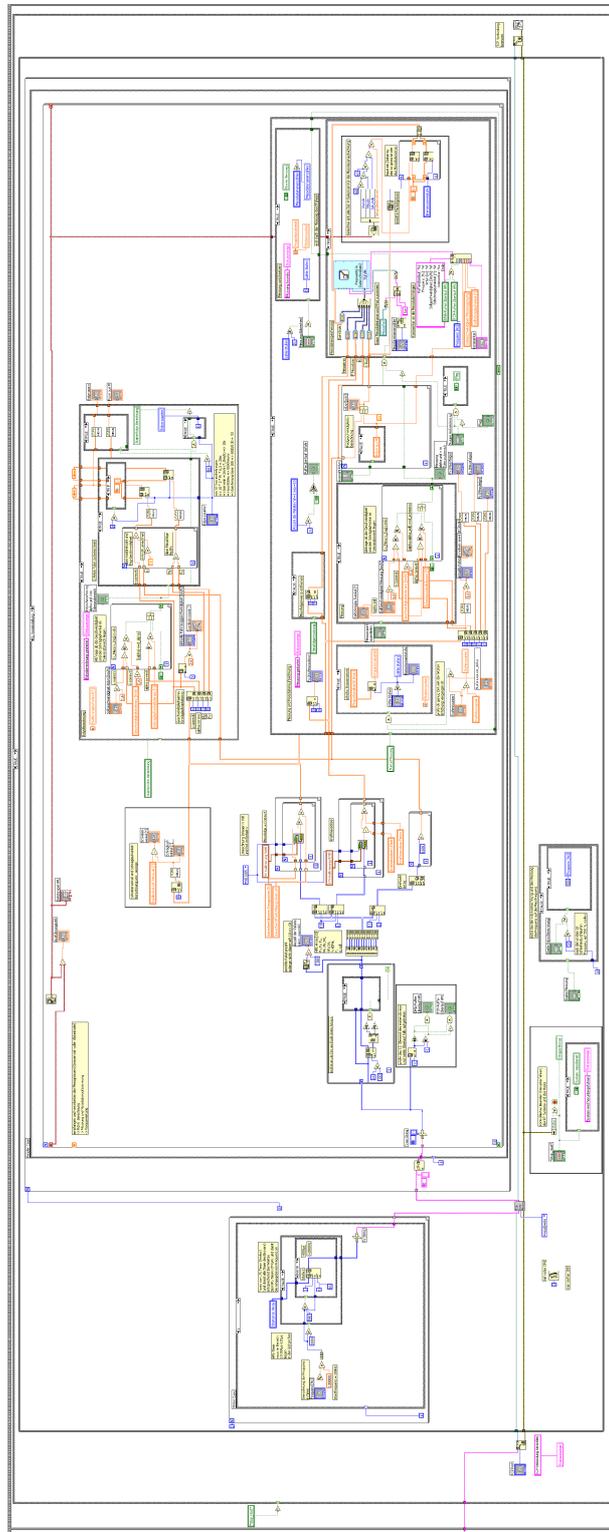
D1.1. Konfiguration



D1.3. CRIO suchen und reseten



D1.4. Netzwerkkommunikation und Hauptfunktionen



Anhang E Modultests

E1. Modultests der FPGA-Module

E1.1. Schräglaufeinstellung

E1.1.1. Modultestvorschrift: Schräglaufeinstellung

Das Modul Schräglaufeinstellung hat die Eingänge Radlenkwinkel am Messrad, Sollradlenkwinkel und Schräglaufwinkeltoleranzbereich je vom Typ Integer, sowie positiver Endschalter am Messrad, negativer Endschalter am Messrad und die Ausgänge Motor (Rechts- / Linkslauf) für Messradschräglaufwinkelverstellung vom Typ Boolean. Die Eingaben Sollschräglaufwinkel und der Schräglaufwinkeltoleranzbereich werden vorher anhand der Kalibrierdaten in Integer umgerechnet. Der Radlenkwinkel wird auf dem Frontpanel eingegeben. Ist die Differenz zwischen Sollradlenkwinkel und Radlenkwinkel innerhalb des Toleranzbereichs, darf kein Motorausgang aktiv sein. Ist der Wert außerhalb des Toleranzbereichs, wird der entsprechende Ausgang aktiv.

Die folgenden Testfälle sind zu überprüfen, der Eingangswert ist entsprechend der Kalibrierdaten in Integer umzurechnen, x steht dabei für eine beliebige Eingabe:

Nr.	Eingaben					erwartete Ausgaben	
	Endschalter positiv	Endschalter negativ	Toleranzbereich [°]	Sollradlenkwinkel [°]	Radlenkwinkel [°]	MotorMR positiv	MotorMR negativ
1	0	0	0	0	0	0	0
2	0	0	0,1	0	0±0,1	0	0
3	0	0	x	x	<-10	x	0
4	0	0	x	x	>10	0	x
5	0	1	x	x	x	x	0
6	1	0	x	x	x	0	x
7	0	0	0,1	5	5±0,1	0	0
8	0	0	0,1	5	-10 bis 4,89	1	0

9	0	0	0,1	5	5,11 bis 10	0	1
---	---	---	-----	---	-------------	---	---

Da die Umrechnung der eingestellten Kalibrierungswerte auf dem Host erfolgt, erfolgt ein intensiveres Testen erst wenn Host und FPGA integriert sind.

E1.1.2. Modultestprotokoll: Schräglaufeinstellung

Das Modul Schräglaufeinstellung wird entsprechend der Modultestvorschrift für dieses Modul getestet. Da zum Zeitpunkt des Tests noch keine Kalibrierdaten vorhanden waren, werden folgende Beispieldaten verwendet.

Maximaler Bereich in Grad:

$$\delta_{MR}: -11^\circ \text{ bis } 12^\circ \quad \text{Eingang: } -10\text{V bis } 10\text{V} \Rightarrow \text{nutzbar: } 0,5\text{V bis } 9,5\text{V}$$

Beziehung Winkel zu Spannung:

$$\delta_{MR}: -11^\circ \triangleq 0,5\text{V} \quad 0^\circ \triangleq 4,9\text{V} \quad 12^\circ \triangleq 9,5\text{V}$$

Beziehung Spannung zu Integer:

$$\delta_{MR}: -10\text{V} \triangleq -31900 \quad 0\text{V} \triangleq 100 \quad 10\text{V} \triangleq 32100$$

Offset in Integer:

$$\delta_{MR}: 1\text{V} = (32100-100)/10 = 3200 \quad \Rightarrow \quad \text{Offset } \delta_{MR} = -(100+4,9*3200) = -15780$$

Deltaberechnung:

$$\Rightarrow \text{delta } x^\circ = x * 3200$$

Beispieleingaben:

$$(\delta_{MR} = -10^\circ) \Rightarrow \delta_{MR} = 4,9+(-10*(9,5-4,9)/12) = 1,07\text{V} \Rightarrow \delta_{MR} = (1,07*3200)+100 = 3513$$

$$(\delta_{MR} = -5^\circ) \Rightarrow \delta_{MR} = 4,9+(-5*(9,5-4,9)/12) = 2,98\text{V} \Rightarrow \delta_{MR} = (2,98*3200)+100 = 9646$$

$$(\delta_{MR} = 0^\circ) \Rightarrow \delta_{MR} = 4,9+(0*(9,5-4,9)/12) = 4,9\text{V} \Rightarrow \delta_{MR} = (4,9*3200)+100 = 15780$$

$$(\delta_{MR} = 5^\circ) \Rightarrow \delta_{MR} = 4,9+(5*(9,5-4,9)/12) = 6,82\text{V} \Rightarrow \delta_{MR} = (6,82*3200)+100 = 21913$$

$$(\delta_{MR} = 10^\circ) \Rightarrow \delta_{MR} = 4,9+(10*(9,5-4,9)/12) = 8,73\text{V} \Rightarrow \delta_{MR} = (8,73*3200)+100 = 28046$$

$$(\delta = 0,1^\circ) \Rightarrow \delta = 0,1 * (9,5-4,9)/12 \Rightarrow \delta_{MR} = (0,038*3200) = 123$$

Die folgenden Testfälle wurden überprüft, der Eingangswert wurde entsprechend der Kalibrierdaten in Integer umgerechnet, x steht hierbei für einen beliebigen Wert:

Für die Grenzwinkel des Radlenkwinkels des Messrades wurden folgende Grenzwerte genommen $\delta_{MR}(-10) = 3513$ und $\delta_{MR}(10) = 28046$.

Nr.	Eingaben					erwartete Ausgaben	
	Endschalter positiv	Endschalter negativ	Toleranzbereich [°] [int]	Sollradlenk-winkel [°] [int]	Radlenkwinkel [°] [int]	MotorMR positiv	MotorMR negativ
1	0	0	0 0	0 15780	0 15780	0 0	0 0
2	0	0	0,1 123	0 15780	0±0,1 15657 bis 15903	0 0	0 0
3	0	0	x	x	<-10 <3513	x x	0 0
4	0	0	x	x	>10 >28046	0 0	x x
5	0	1	x	x	x	x x	0 0
6	1	0	x	x	x	0 0	x x
7	0	0	0,1 123	5 21913	5±0,1 21790 bis 22036	0 0	0 0
8	0	0	0,1 123	5 21913	-10 bis 4,89 3513 bis 21790	1 1	0 0
9	0	0	0,1 123	5 21913	5,11 bis 10 22035 bis 28046	0 0	1 1

Bei dieser Testdurchführung wurde ein Fehler beim Vergleich mit den Grenzwinkeln ($\pm 10^\circ$) entdeckt und behoben, danach wurde der Test erneut ausgeführt und bestanden.

E1.2. Schwimmwinkelkompensation

E1.2.1. Modultestvorschrift: Schwimmwinkelkompensation

Das Modul Schwimmwinkelkompensation hat die Eingänge aktiv, positiver Endschalter am Referenzrad, negativer Endschalter am Referenzrad vom Typ Boolean, sowie Radlenkwinkel am Messrad, Radlenkwinkel am Referenzrad, Schräglaufwinkel, Schräglaufwinkeltoleranzbereich je vom Typ Integer und die Ausgänge Motor (Rechts- / Linkslauf) für Referenzrad-schräglaufwinkelverstellung vom Typ Boolean. Die Eingaben vom Typ Boolean werden direkt in der Software gesetzt, ein vorheriges Überprüfen der korrekten Erkennung der digitalen Eingangsdaten vorausgesetzt. Der Radlenkwinkel am Referenzrad und der Schwimmwinkel-toleranzbereich werden vorher anhand der Kalibrierdaten in Integer umgerechnet. Der Schräglaufwinkel und der Radlenkwinkel am Messrad werden entsprechend der Kalibrierung über eine Spannungsquelle eingestellt und der Wert mit einem Voltmeter überprüft. Beide Werte werden kontinuierlich verstellt. Ist die Differenz zwischen Schräglaufwinkel und Radlenkwinkel am Messrad innerhalb des Toleranzbereichs, darf kein Motorausgang aktiv sein. Ist der Wert außerhalb des Toleranzbereichs wird der entsprechende Ausgang aktiv.

Die folgenden Testfälle sind zu überprüfen, der Eingangswert ist entsprechend der Kalibrierdaten in Integer umzurechnen:

Nr.	Eingaben							erwartete Ausgaben	
	aktiv	End-schalter positiv	End-schalter negativ	Toleranzbereich [°]	RadlenkwinkelRR [°]	RadlenkwinkelMR [°]	Schräglaufwinkel [°]	MotorRR positiv	MotorRR negativ
1	0	0	0	x	x	x	x	0	0
2	1	0	0	0,2	x	2	2	0	0
3	1	0	0	0,2	x	2	2,2	1	0
4	1	0	0	0,2	10	2	2,2	0	0
5	1	1	0	0,2	x	2	2,2	0	0
6	1	0	0	0,2	x	2	1,8-2,2	0	0
7	1	0	0	0,2	x	2	2,2	0	1

8	1	0	0	0,2	-10	2	2,2	0	0
9	1	0	1	0,2	x	2	2,2	0	0
10	1	0	0	0,2	x	>2,2	2	0	1
11	1	0	0	0,2	x	1,8-2,2	2	0	0
12	1	0	0	0,2	x	<1,8	2	1	0

Da die Umrechnung der eingestellten Kalibrierungswerte auf dem Host erfolgt, erfolgt ein intensiveres Testen erst wenn Host und FPGA integriert sind.

E1.2.2. Modultestprotokoll: Schwimmwinkelkompensation

Da zum Testzeitpunkt noch keine Kalibrierdaten vorlagen, wurden folgenden Kalibrierwerte beispielhaft angenommen.

Winkelbereich:

α : -40° bis 40° Eingang: -5V bis 5V

δ_{MR} : -11° bis 12° Eingang: -10V bis 10V \Rightarrow nutzbar: 0,5V bis 9,5V

Beziehung Winkel zu Spannung:

α : -40° \triangleq -5V 0° \triangleq 0V 40° \triangleq 5V

δ_{MR} : -11° \triangleq 0,5V 0° \triangleq 4,9V 12° \triangleq 9,5V

Offset $\alpha = 0V$

Offset $\delta_{MR} = 11/(23/9) = -4,9V$

Beziehung Spannung zu Integer:

α : -5V = -31900 0V = 100 5V = 32100

δ_{MR} : -10V = -32000 0V = 0 10V = 32000

Offset in Integer:

α : $1V = (32100-100)/5 = 1600 \Rightarrow$ Offset $\alpha = -(100+0*1600) = -100$

δ_{MR} : $1V = (32000-0)/10 = 3200 \Rightarrow$ Offset $\delta_{MR} = -(0+4,9*3200) = -15680$

Faktorisierung:

$$\alpha: \quad \text{maxWinkelbereich} = 40+40=80 \quad \text{maxIntbereich} = 31900+32100 = 64000$$

$$\Rightarrow \quad \text{Faktor } \alpha = 1^\circ = 64000/80^\circ = 800$$

$$\delta_{MR}: \quad \text{maxWinkelbereich} = 11+12=23 \quad \text{maxIntbereich} = 3200-(2*1600) = 28800$$

$$\Rightarrow \quad \text{Faktor } \delta_{MR} = 1^\circ = 28800/23^\circ = 1252$$

Deltaberechnung:

$$\Rightarrow \text{delta } x^\circ = \text{Faktor } \alpha * \text{Faktor } \delta_{MR} * x$$

$$(\alpha = 2^\circ) \Rightarrow \alpha = 0+(2*(5-0)/40) = 0,25V \Rightarrow \alpha = (0,25*6400)+100 = 1700$$

$$(\delta_{MR} = 2^\circ) \Rightarrow \delta_{MR} = 4,9+(2*(9,5-4,9)/12) = 5,66V \Rightarrow \delta_{MR} = (5,66*3200) = 18133$$

Kontrollrechnung:

Bsp: (2°)

$$\text{Int1} \alpha = 1700 \Rightarrow 1700+\text{Offset} \quad \alpha = 1700-100 = 1600$$

$$1600 * \text{Faktor } \delta_{MR} = 1600 * 1252 = 2003200$$

$$\text{Int2} \quad \delta_{MR} = 18184 \Rightarrow 18184+\text{Offset} \quad \delta_{MR} = 18184-15680 = 2504$$

$$2504 * \text{Faktor } \alpha = 2504 * 800 = 2003200$$

Bsp: (delta 0,2°)

$$\text{delta } 0,2 \text{ in Int} = 1252 * 800 * 0,2 = 200320$$

Die folgenden Testfälle sind zu überprüfen, der Eingangswert ist entsprechend der Kalibrierdaten in Integer umzurechnen, x steht dabei für eine beliebige Eingabe:
Für die Grenzwinkel des Referenzrades werden folgende Grenzwerte angenommen $\delta_{MR}(-10) = 2826$ und $\delta_{MR}(10) = 27946$.

Nr.	Eingaben							Ausgaben erwartet tatsächlich	
	aktiv	End- schalter positiv	End- schalter negativ	Toleranz- bereich [°] [int]	Radlenk- winkelRR [°] [int]	Radlenk- winkelMR [°] [int]	Schräg- laufwinkel [°] [int]	Mo- torRR positiv	Mo- torRR negativ
1	0	0	0	x	x	x	x	0 0	0 0
2	1	0	0	0,2 200320	x	2 18184	2 1700	0 0	0 0
3	1	0	0	0,2 200320	x	2 18184	>2,2 >1860	1 1	0 0
4	1	0	0	0,2 200320	10 27946	2 18184	2,2 >1860	0 0	0 0
5	1	1	0	0,2 200320	x	2 18184	2,2 >1860	0 0	0 0
6	1	0	0	0,2 200320	x	2 18184	1,8-2,2 1540-1860	0 0	0 0
7	1	0	0	0,2 200320	x	2 18184	1,8 <1540	0 0	1 1
8	1	0	0	0,2 200320	-10 2826	2 18184	1,8 <1540	0 0	0 0
9	1	0	1	0,2 200320	x	2 18184	1,8 <1540	0 0	0 0
10	1	0	0	0,2 200320	x	>2,2 >18434	2 1700	0 0	1 1
11	1	0	0	0,2 200320	x	1,8-2,2 17934-18434	2 1700	0 0	0 0
12	1	0	0	0,2 200320	x	<1,8 <17934	2 1700	1 1	0 0

Bei diesem Modultest wurden die Testfälle der Tabelle durchgetestet. Die tatsächlichen Ausgaben entsprachen den erwarteten Ausgaben somit ist der Modultest bestanden.

E1.3. Drehzahlberechnung

E1.3.1. Modultestvorschrift: Drehzahlberechnung

Der Modultest der Drehzahlberechnung wird auf zwei Arten ausgeführt, da bei der 1.Variante die Prüffrequenz von der gleichen Clock abhängig ist wie die Drehzahlberechnung, die 1.Variante aber genauer ist. Die Eingabe ist eine Rechteckfrequenz, welche das Signal des Inkrementalgebers simuliert. Die Ausgabe ist ein Ausgabeelement des Drehzahlberechnungsmoduls, das die Drehzahl in [1/1000s] ausgibt.

1. Es wird in einem zur Drehzahlberechnung parallelen Prozess auf dem FPGA ein Signalgenerator simuliert. Dieser wurde programmiert. Am entsprechenden Eingabefeld wird der Timer für den Schleifendurchlauf eingestellt, der eine entsprechende Frequenz erzeugt. Die folgenden Testfälle sind zu überprüfen:

Nr.	Eingabefrequenz [Hz = Striche/s]	erwartete Ausgabe [1/1000s]	Bemerkung
1	1000	1000	untere Grenze ca. 10km/h
2	16000	16000	obere Grenze ca. 100km/h
3	5555	5555	
4	12112	12112	
5	200	200	außerhalb des Messbereichs
6	30000	30000	

2. Es wird ein Signalgenerator an den digitalen Eingang des Digitalmoduls NI-9401 angeschlossen. Am Signalgenerator wird die Frequenz eingestellt und die Signalart, ein Rechteckimpuls, ausgewählt. Das berechnete Drehzahlsignal wird auf der Bildschirmanzeige abgelesen. Da die Einstellung analog über ein Drehrad erfolgt und der Frequenzgenerator einen Fehler von 5-10% hat, wird hier ein Bereich von $\pm 10\%$ als richtiger Wert betrachtet. Die folgenden Testfälle sind zu überprüfen:

Nr.	Eingabefrequenz [Hz = Striche/s]	erwartete Ausgabe [1/1000s]	Bemerkung
1	1000	900-1100	untere Grenze ca. 10km/h

2	16000	14400-17600	obere Grenze ca. 100km/h
3	5000	4500-5500	
4	12000	10800-13200	
5	200	180-220	außerhalb des Messbereichs
6	30000	27000-33000	

E1.3.2. Modultestprotokoll: Drehzahlberechnung

Der Modultest der Drehzahlberechnung wurde auf zwei Arten, entsprechend der Modultestvorschrift der Drehzahlberechnung, durchgeführt:

1. Die folgenden Testfälle wurden, entsprechend der Modultestvorschrift für das Drehzahlberechnungsmodul durchgeführt:

Nr.	Eingabe Timer [µs]	Frequenz = $10^6 / (\text{Eingabe} * 2)$ [Hz]	erwartete Ausgabe [1/1000s]	Ausgabe	Status
1	500	1000	1000	1000	OK
2	31	16129	16129	16129	OK
3	100	5000	5000	5000	OK
4	42	11904	11904	11904	OK
6	2500	200	200	200	OK
7	30000	29411	29411	29411	OK

2. Es wurde der Funktionsgenerator PM 5127 von Philips an den digitalen Eingang des Digitalmoduls NI-9401 angeschlossen. Die Frequenzen wurden entsprechend des folgenden Testplans eingestellt und die Ausgabe überprüft:

Nr.	Eingabefrequenz [Hz = Striche/s]	erwartete Ausgabe [1/1000s]	Ausgabe [1/1000s]	Status
1	1000	900-1100	1052	OK
2	16000	14400-17600	16129	OK
3	5000	4500-5500	5076	OK
4	12000	10800-13200	12048	OK

6	200	180-220	202	OK
7	30000	27000-33000	30303	OK

Da das Drehzahlmodul beide Modultests ohne Fehler bestanden hat, ist das Drehzahlmodul verifiziert.

E1.4. Low Priority Loop

E1.4.1. Modultestvorschrift: Low Priority Loop

Das Modul Low Priority Loop wird geprüft, indem die Messdaten auf dem Realsystem aus dem DMA-Speicher gelesen und angezeigt werden. Des Weiteren wird auf dem Realsystem der Wert des Proportionalventils und der der Nullkalibrierung der Messfelge eingegeben und über den DMA-Speicher an das Low Priority Loop Modul gesendet. Der eingestellte Wert des Proportionalventils wird mit einem Voltmeter am entsprechenden Ausgang überprüft. Der Wert der Nullkalibrierung wird durch die entsprechende LED am digitalen Ausgangsmodul überprüft.

Die Messdaten werden erzeugt, indem der zu testende Kanal mit einer Spannung versehen wird. Um Wechselwirkungen auszuschließen werden alle weiteren zu prüfenden Kanäle an einen zweiten Spannungsgenerator angeschlossen und die Spannung während jedes Testwertes verändert. Die Spannung wird mit einem Voltmeter überprüft. Um die erfassten Messdaten auf Korrektheit zu prüfen werden die Messwerte auf dem Realsystem in Volt umgerechnet. Die Ausgabegenauigkeit des Analogausgangsmoduls NI-9263 beträgt 0,75% des maximalen Bereichs plus 0,35% Lesefehler. Die Messgenauigkeit des Analogmoduls NI-9205 wird mit 1% auf den eingestellten Bereich angegeben. Das zur Verfügung stehende Voltmeter hat eine Messgenauigkeit von 2%, wie eine Vergleichsmessung mit mehreren kalibrierten Messgeräten ergab. Somit ist ein maximaler Gesamtfehler von 3% beim Eingangsmodul zu erwarten und 3,1% beim Ausgangsmodul.

Die folgenden Testfälle sind zu überprüfen, der entsprechende Kanal ist mit der entsprechenden Spannung zu versehen:

Nr.	Messgröße	Eingabe [V]					erwartete Ausgabe [V] $\pm 3\%$					
		0	1				0	1				
1	Nullkalib.	0	1				0	1				
2	U_{Prop}	0	2,3	5	7,8	10	$0 \pm 0,31$	$2,3 \pm 0,31$	$5 \pm 0,31$	$7,8 \pm 0,31$	$10 \pm 0,31$	
3	U_{Bord}	0	2,3	5	7,8	10	$0 \pm 0,3$	$2,3 \pm 0,3$	$5 \pm 0,3$	$7,8 \pm 0,3$	$10 \pm 0,3$	

4	$p_{\text{Hydraulik}}$	0	2,3	5	7,8	10	$0\pm 0,3$	$2,3\pm 0,3$	$5\pm 0,3$	$7,8\pm 0,3$	$10\pm 0,3$
5	$T_{\text{Hydraulik}}$	0	2,3	5	7,8	10	$0\pm 0,3$	$2,3\pm 0,3$	$5\pm 0,3$	$7,8\pm 0,3$	$10\pm 0,3$
7	δ_{MR}	0	2,3	5	7,8	10	$0\pm 0,3$	$2,3\pm 0,3$	$5\pm 0,3$	$7,8\pm 0,3$	$10\pm 0,3$
8	δ_{RR}	0	2,3	5	7,8	10	$0\pm 0,3$	$2,3\pm 0,3$	$5\pm 0,3$	$7,8\pm 0,3$	$10\pm 0,3$

E1.4.2. Modultestprotokoll: Low Priority Loop

Das Modul Low Priority Loop wurde entsprechend der Modultestvorschrift für dieses Modul getestet.

Die folgenden Testfälle wurden überprüft, der entsprechende Kanal wurde mit der in den Testfällen angegebenen Spannungen versehen:

Nr.	Messgröße	Eingabe [V]					erwartete Ausgabe [V] $\pm 0,3$ beobachtete Ausgabe				
		0	1				0	1			
1	Nullkalib.	0	1				0	1			
		0	1				0	1			
2	U_{Prop}	0	2,3	5	7,8	10	$0\pm 0,31$ 0,000	$2,3\pm 0,31$ 2,300	$5\pm 0,31$ 5,002	$7,8\pm 0,31$ 7,802	$10\pm 0,31$ 10,003
3	U_{Bord}	0	2,3	5	7,8	10	$0\pm 0,3$ -0,013	$2,3\pm 0,3$ 2,258	$5\pm 0,3$ 4,990	$7,8\pm 0,3$ 7,770	$10\pm 0,3$ 9,975
4	$p_{\text{Hydraulik}}$	0	2,3	5	7,8	10	$0\pm 0,3$ -0,013	$2,3\pm 0,3$ 2,260	$5\pm 0,3$ 4,992	$7,8\pm 0,3$ 7,770	$10\pm 0,3$ 9,976
5	$T_{\text{Hydraulik}}$	0	2,3	5	7,8	10	$0\pm 0,3$ -0,013	$2,3\pm 0,3$ 2,260	$5\pm 0,3$ 4,990	$7,8\pm 0,3$ 7,770	$10\pm 0,3$ 9,975
7	δ_{MR}	0	2,3	5	7,8	10	$0\pm 0,3$ -0,014	$2,3\pm 0,3$ 2,263	$5\pm 0,3$ 4,995	$7,8\pm 0,3$ 7,781	$10\pm 0,3$ 9,976
8	δ_{RR}	0	2,3	5	7,8	10	$0\pm 0,3$ -0,014	$2,3\pm 0,3$ 2,263	$5\pm 0,3$ 4,995	$7,8\pm 0,3$ 7,781	$10\pm 0,3$ 9,976

Alle Ausgaben liegen innerhalb des Toleranzbereichs der erwarteten Ausgaben und es fand kein Pufferüberlauf statt, damit ist der Modultest des Moduls FPGA-LPL bestanden.

E1.5. High Priority Loop

E1.5.1. Modultestvorschrift: High Priority Loop

Das Modul High Priority Loop wird geprüft, indem die Messdaten auf dem Realsystem aus dem DMA-Speicher gelesen und angezeigt werden. Des Weiteren wird mittels einer Ab-

frage geprüft, ob der DMA-Puffer voll war und deshalb keine Daten hineingeschrieben werden konnten. Die Messwerte werden erzeugt, indem der zu testende Kanal mit einer Spannung versehen wird. Um Wechselwirkungen auszuschließen werden alle weiteren zu prüfenden Kanäle an einen zweiten Spannungsgenerator angeschlossen und die Spannung während jedes Testwertes verändert. Die Spannung wird mit einem Voltmeter überprüft. Um die erfassten Messdaten auf Korrektheit zu prüfen werden die Messwerte auf dem Realsystem in Volt umgerechnet. Die Messgenauigkeit des Analogmoduls NI-9205 wird mit 1% auf den eingestellten Bereich angegeben und die zur Verfügung stehenden Voltmeter haben eine Messgenauigkeit von 2%, somit ist ein maximaler Gesamtfehler von 3% zu erwarten.

Die folgenden Testfälle sind zu überprüfen, der entsprechende Kanal ist mit der entsprechenden Spannung zu versehen:

Nr.	Messgröße	Eingabe [V]					erwartete Ausgabe [V] ± 0,15				
		-5	-2,3	0	2,3	5	-5±0,15	-2,3±0,15	0±0,15	2,3±0,15	5±0,15
1	F _{X, KMF}	-5	-2,3	0	2,3	5	-5±0,15	-2,3±0,15	0±0,15	2,3±0,15	5±0,15
2	F _{Y, KMF}	-5	-2,3	0	2,3	5	-5±0,15	-2,3±0,15	0±0,15	2,3±0,15	5±0,15
3	F _{Z, KMF}	-5	-2,3	0	2,3	5	-5±0,15	-2,3±0,15	0±0,15	2,3±0,15	5±0,15
4	M _{X, KMF}	-5	-2,3	0	2,3	5	-5±0,15	-2,3±0,15	0±0,15	2,3±0,15	5±0,15
5	M _{Y, KMF}	-5	-2,3	0	2,3	5	-5±0,15	-2,3±0,15	0±0,15	2,3±0,15	5±0,15
6	M _{Z, KMF}	-5	-2,3	0	2,3	5	-5±0,15	-2,3±0,15	0±0,15	2,3±0,15	5±0,15
7	cosφ	-5	-2,3	0	2,3	5	-5±0,15	-2,3±0,15	0±0,15	2,3±0,15	5±0,15
8	sinφ	-5	-2,3	0	2,3	5	-5±0,15	-2,3±0,15	0±0,15	2,3±0,15	5±0,15
9	F _{X, KMD}	0	2,3	5	7,8	10	0±0,3	2,3±0,3	5±0,3	7,8±0,3	10±0,3
10	α	-5	-2,3	0	2,3	5	-5±0,15	-2,3±0,15	0±0,15	2,3±0,15	5±0,15
11	v _{Anhänger}	-5	-2,3	0	2,3	5	-5±0,15	-2,3±0,15	0±0,15	2,3±0,15	5±0,15

E1.5.2. Modultestprotokoll: High Priority Loop

Das Modul High Priority Loop wurde entsprechend der Modultestvorschrift getestet.

Die folgenden Testfälle wurden überprüft, der entsprechende Kanal wurde mit der entsprechenden Spannung versehen:

Nr.	Messgröße	Eingabe [V]					erwartete Ausgabe [V] ±0,15				
		-5	-2,3	0	2,3	5	beobachtete Ausgabe				
1	F _{X, KMF}	-5	-2,3	0	2,3	5	-5±0,15	-2,3±0,15	0±0,15	2,3±0,15	5±0,15
							-5,016	-2,336	-0,054	2,295	4,979

2	$F_{Y, KMF}$	-5	-2,3	0	2,3	5	$-5\pm 0,15$ -5,016	$-2,3\pm 0,15$ -2,342	$0\pm 0,15$ -0,032	$2,3\pm 0,15$ 2,295	$5\pm 0,15$ 4,977
3	$F_{Z, KMF}$	-5	-2,3	0	2,3	5	$-5\pm 0,15$ -5,016	$-2,3\pm 0,15$ -2,336	$0\pm 0,15$ -0,055	$2,3\pm 0,15$ 2,295	$5\pm 0,15$ 4,976
4	$M_{X, KMF}$	-5	-2,3	0	2,3	5	$-5\pm 0,15$ -5,015	$-2,3\pm 0,15$ -2,335	$0\pm 0,15$ -0,043	$2,3\pm 0,15$ 2,296	$5\pm 0,15$ 4,975
5	$M_{Y, KMF}$	-5	-2,3	0	2,3	5	$-5\pm 0,15$ -5,016	$-2,3\pm 0,15$ -2,336	$0\pm 0,15$ -0,054	$2,3\pm 0,15$ 2,295	$5\pm 0,15$ 4,974
6	$M_{Z, KMF}$	-5	-2,3	0	2,3	5	$-5\pm 0,15$ -5,014	$-2,3\pm 0,15$ -2,336	$0\pm 0,15$ -0,033	$2,3\pm 0,15$ 2,296	$5\pm 0,15$ 4,974
7	$\cos\varphi$	-5	-2,3	0	2,3	5	$-5\pm 0,15$ -5,017	$-2,3\pm 0,15$ -2,336	$0\pm 0,15$ -0,054	$2,3\pm 0,15$ 2,293	$5\pm 0,15$ 4,974
8	$\sin\varphi$	-5	-2,3	0	2,3	5	$-5\pm 0,15$ -5,017	$-2,3\pm 0,15$ -2,335	$0\pm 0,15$ -0,055	$2,3\pm 0,15$ 2,294	$5\pm 0,15$ 4,974
9	$F_{X, KMD}$	0	2,3	5	7,8	10	$0\pm 0,3$ -0,057	$2,3\pm 0,3$ 2,272	$5\pm 0,3$ 4,976	$7,8\pm 0,3$ 7,76	$10\pm 0,3$ 9,975
10	α	-5	-2,3	0	2,3	5	$-5\pm 0,15$ -5,016	$-2,3\pm 0,15$ -2,336	$0\pm 0,15$ -0,051	$2,3\pm 0,15$ 2,295	$5\pm 0,15$ 4,974
11	$v_{\text{Anhänger}}$	-5	-2,3	0	2,3	5	$-5\pm 0,15$ -5,016	$-2,3\pm 0,15$ -2,333	$0\pm 0,15$ -0,045	$2,3\pm 0,15$ 2,295	$5\pm 0,15$ 4,974

Alle Ausgaben liegen innerhalb des Toleranzbereichs der erwarteten Ausgaben und es fand kein Pufferüberlauf statt, damit ist der Modultest des Moduls FPGA-HPL bestanden.

E2. Modultests der Host-Module

E2.1. Einstellungen

E2.1.1. Modultestvorschrift: Einstellungen

Das Modul Einstellungen wird getestet, indem die Eingabefelder mit verschiedenen Werten belegt werden und diese mit den Werten in der erstellten Konfigurationsdatei, bzw. in den entsprechenden Variablen die für diesen Test ausgegeben werden, verglichen werden. Die zu testenden Eingabewerte und die erwarteten Ausgaben sind den folgenden Tabellen zu entnehmen:

Eingabefeld: Ordnerpfad		
Nr.	Eingabe	erwartete Ausgabe
1	U:\LabVIEW\Messungen	U:\LabVIEW\Messungen
2		Fehler

Eingabefeld: Messdateiname		
Nr.	Eingabe	erwartete Ausgabe
1	Reifen1	Reifen1
2	10Fahrten	10Fahrten

Eingabefeld: Messfelge montiert?		
Nr.	Eingabe	erwartete Ausgabe
1	Schalterstellung: 1	true
2	Schalterstellung: 0	false

Eingabefeld: Proportionalventilstufen [V] (für jede Stufe testen)		
Nr.	Eingabe	erwartete Ausgabe
1	0	0
2	5,3	5
3	10	10

4	n456	Fehler: keine Zahl
5	-3	Fehler: negativ \Rightarrow 0
6	abc	Fehler: keine Zahl

Eingabefeld: Geschwindigkeitstoleranz [0,1 – 10 km/h]		
Nr.	Eingabe	erwartete Ausgabe
1	0,1	0,1
2	0,3	0,3
3	2,7	2,7
4	0,02	Fehler: zu klein \Rightarrow 0,1
5	-0,6	Fehler: negativ \Rightarrow 0,1
6	10,3	Fehler: zu groß \Rightarrow 10
7	yx	Fehler: keine Zahl
8	t567	Fehler: keine Zahl

Eingabefeld: Schräglaufwinkeltoleranz [0,1 – 5°]		
Nr.	Eingabe	erwartete Ausgabe
1	0,1	0,1
2	0,3	0,3
3	1,5	1,5
4	0,005	Fehler: zu klein \Rightarrow 0,1
5	-0,6	Fehler: negativ \Rightarrow 0,1
6	5,3	Fehler: zu groß \Rightarrow 5
7	yx	Fehler: keine Zahl
8	t567	Fehler: keine Zahl

Eingabefeld: Schwimmwinkeltoleranz [0,1 – 5°]		
Nr.	Eingabe	erwartete Ausgabe
1	0,1	0,1
2	0,3	0,3
3	1,5	1,5
4	0,005	Fehler: zu klein \Rightarrow 0,1
5	-0,6	Fehler: negativ \Rightarrow 0,1
6	5,3	Fehler: zu groß \Rightarrow 5
7	yxa	Fehler: keine Zahl
8	t567	Fehler: keine Zahl

Eingabefeld: Reifentemperatur [-10 – 200°C]		
Nr.	Eingabe	erwartete Ausgabe
1	-10	-10
2	32,4	32,4
3	100	100
4	-50,2	Fehler: zu klein \Rightarrow -10
5	200,3	Fehler: zu groß \Rightarrow 200
6	yxa	Fehler: keine Zahl
7	t567	Fehler: keine Zahl

Eingabefeld: Reifendruck [1 – 10 bar]		
Nr.	Eingabe	erwartete Ausgabe
1	1	1
2	2,3	2,3
3	10	10
4	0,9	Fehler: zu klein \Rightarrow 1
5	-0,6	Fehler: negativ \Rightarrow 1

6	10,1	Fehler: zu groß \Rightarrow 10
7	yxa	Fehler: keine Zahl
8	t567	Fehler: keine Zahl

Eingabefeld: Asphalttemperatur [-30 – 100°C]		
Nr.	Eingabe	erwartete Ausgabe
1	-30	-30
2	32,4	32,4
3	100	100
4	-50,2	Fehler: zu klein \Rightarrow -30
5	100,3	Fehler: zu groß \Rightarrow 100
6	yxa	Fehler: keine Zahl
7	t567	Fehler: keine Zahl

Eingabefeld: weitere Bemerkungen		
Nr.	Eingabe	erwartete Ausgabe
1	-50	-50
2	350/70	350/70
3	Reifen 250	Reifen 250
4	412 km	412 km

E2.1.2. Modultestprotokoll: Einstellungen

Das Modul Einstellungen wurde entsprechend der Modultestvorschrift getestet.

Die Testfälle und die eingetretenen Ausgaben sind den folgenden Tabellen zu entnehmen:

Eingabefeld: Ordnerpfad			
Nr.	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
1	U:\LabVIEW\Messungen	U:\LabVIEW\Messungen	U:\LabVIEW\Messungen

2		Fehler	Fehler
---	--	--------	--------

Eingabefeld: Messdateiname			
Nr.	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
1	Reifen1	Reifen1	Reifen1
2	10Fahrten	10Fahrten	10Fahrten

Eingabefeld: Messfelge montiert?			
Nr.	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
1	Schalterstellung: 1	true	true
2	Schalterstellung: 0	false	false

Eingabefeld: Proportionalventilstufen [V] (für jede Stufe getestet)			
Nr.	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
1	0	0	0
2	5,3	5	5
3	10	10	10
4	n456	Fehler: keine Zahl	10 \triangleq letztem gültigen Wert
5	-3	Fehler: negativ \Rightarrow 0	0
6	abc	Fehler: keine Zahl	0 \triangleq letztem gültigen Wert

Eingabefeld: Geschwindigkeitstoleranz [0,1 – 10 km/h]			
Nr.	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
1	0,1	0,1	0,1
2	0,3	0,3	0,3
3	2,7	2,7	2,7
4	0,02	Fehler: zu klein \Rightarrow 0,1	0,1
5	-0,6	Fehler: negativ \Rightarrow 0,1	0,1

6	10,3	Fehler: zu groß \Rightarrow 10	10
7	yxá	Fehler: keine Zahl	10 \triangleq letztem gültigen Wert
8	t567	Fehler: keine Zahl	10 \triangleq letztem gültigen Wert

Eingabefeld: Schräglaufwinkeltoleranz [0,1 – 5°]			
Nr.	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
1	0,1	0,1	0,1
2	0,3	0,3	0,3
3	1,5	1,5	1,5
4	0,005	Fehler: zu klein \Rightarrow 0,1	0,1
5	-0,6	Fehler: negativ \Rightarrow 0,1	0,1
6	5,3	Fehler: zu groß \Rightarrow 5	5
7	yxá	Fehler: keine Zahl	5 \triangleq letztem gültigen Wert
8	t567	Fehler: keine Zahl	5 \triangleq letztem gültigen Wert

Eingabefeld: Schwimmwinkeltoleranz [0,1 – 5°]			
Nr.	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
1	0,1	0,1	0,1
2	0,3	0,3	0,3
3	1,5	1,5	1,5
4	0,005	Fehler: zu klein \Rightarrow 0,1	0,1
5	-0,6	Fehler: negativ \Rightarrow 0,1	0,1
6	5,3	Fehler: zu groß \Rightarrow 5	5
7	yxá	Fehler: keine Zahl	5 \triangleq letztem gültigen Wert
8	t567	Fehler: keine Zahl	5 \triangleq letztem gültigen Wert

Eingabefeld: Reifentemperatur [-10 – 200°C]			
Nr.	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
1	-10	-10	-10
2	32,4	32,4	32,4
3	100	100	100
4	-50,2	Fehler: zu klein \Rightarrow -10	-10
5	200,3	Fehler: zu groß \Rightarrow 200	200
6	yx	Fehler: keine Zahl	200 \triangleq letztem gültigen Wert
7	t567	Fehler: keine Zahl	200 \triangleq letztem gültigen Wert

Eingabefeld: Reifendruck [1 – 10 bar]			
Nr.	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
1	1	1	1
2	2,3	2,3	2,3
3	10	10	10
4	0,9	Fehler: zu klein \Rightarrow 1	1
5	-0,6	Fehler: negativ \Rightarrow 1	1
6	10,1	Fehler: zu groß \Rightarrow 10	10
7	yx	Fehler: keine Zahl	10 \triangleq letztem gültigen Wert
8	t567	Fehler: keine Zahl	10 \triangleq letztem gültigen Wert

Eingabefeld: Asphalttemperatur [-30 – 100°C]			
Nr.	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
1	-30	-30	-30
2	32,4	32,4	32,4
3	100	100	100
4	-50,2	Fehler: zu klein \Rightarrow -30	-30

5	100,3	Fehler: zu groß \Rightarrow 100	100
6	yxα	Fehler: keine Zahl	100 \triangleq letztem gültigen Wert
7	t567	Fehler: keine Zahl	100 \triangleq letztem gültigen Wert

Eingabefeld: weitere Bemerkungen			
Nr.	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
1	-50	-50	-50
2	350/70	350/70	350/70
3	Reifen 250	Reifen 250	Reifen 250
4	412 km	412 km	412 km

E2.2. R_{dyn} -Berechnung

E2.2.1. Modultestvorschrift: R_{dyn} -Berechnung

Das Modul R_{dyn} -Berechnung hat die Eingänge Messfelgennutzung, Schräglaufwinkeltoleranz, Geschwindigkeitstoleranz, und Messdatenarray. Es wird getestet, indem die folgenden Testparameter verwendet werden und der entsprechende dynamische Reifenhalbmesser berechnet und mit dem aus der Tabelle verglichen wird. Um den Testaufwand zu vereinfachen wird die Mittelwertbildung auf 3 reduziert.

Die folgenden Testfälle sind zu überprüfen:

Nr.	Eingaben									erwartete Ausgabe
	Messfelge	$\Delta\alpha$ [°]	Δv [km/h]	v_{Mess} [km/h]	F_x [N]	M_x [Nm]	v [km/h]	α [°]	N [1/s]	R_{dyn}
1	0	0,2	2	50	1000	300	51	0	26	0,299
					999	300	51	0,1	25	
					998	299	51	0,1	25	
2	1	0,2	2	50	1000	300	51	0	26	0,320
					999	300	51	0,1	25	
					998	299	51	0,1	25	
3	0	0,2	2	50	1000	300	51	0	26	Abbruch $\Delta\alpha$
					999	300	51	0,3	25	

					998	299	51	0,1	25	
4	1	0,2	2	50	1000	300	51	0	26	Abbruch
					999	300	51	0,3	25	$\Delta\alpha$
					998	299	51	0,1	25	
5	0	0,2	2	50	1000	300	51	0	26	Abbruch
					999	300	51	0	25	Δv
					998	299	53	0,1	25	
6	1	0,2	2	50	1000	300	51	0	26	Abbruch
					999	300	51	0	25	Δv
					998	299	53	0,1	25	

E2.2.2. Modultestprotokoll: R_{dyn} -Berechnung

Das Modul R_{dyn} -Berechnung wurde entsprechend dessen Modultestvorschrift getestet.

Die folgenden Testfälle wurden dabei verwendet:

Eingaben										erwartete tatsächliche Ausgabe
Nr.	Mess- felge	$\Delta\alpha$ [°]	Δv [km/h]	v_{Mess} [km/h]	F_x [N]	M_x [Nm]	v [km/h]	α [°]	N [1/s]	R_{dyn}
1	0	0,2	2	50	1000	300	51	0	26	0,299
					999	300	51	0,1	25	0,299
					998	299	51	0,1	25	
2	1	0,2	2	50	1000	300	51	0	26	0,320
					999	300	51	0,1	25	0,320
					998	299	51	0,1	25	
3	0	0,2	2	50	1000	300	51	0	26	Abbruch
					999	300	51	0,3	25	Abbruch
					998	299	51	0,1	25	
4	1	0,2	2	50	1000	300	51	0	26	Abbruch
					999	300	51	0,3	25	Abbruch
					998	299	51	0,1	25	
5	0	0,2	2	50	1000	300	51	0	26	Abbruch
					999	300	51	0	25	Abbruch
					998	299	53	0,1	25	

6	1	0,2	2	50	1000	300	51	0	26	Abbruch
					999	300	51	0	25	Abbruch
					998	299	53	0,1	25	

E2.3. Messung

E2.3.1. Modultestvorschrift: Messung

Das Modul Messung hat die Eingänge Sollschräglaufwinkel, Schräglaufwinkeltoleranz Geschwindigkeitstoleranz und Messdatenarray. Es wird getestet, indem die folgenden Testparameter verwendet werden und die erstellte Messdatei mit den Eingangswerten verglichen wird. Um den Testaufwand zu verringern, wird das Messdateneingangsarray auf 3 Elemente reduziert.

Die folgenden Testfälle sind zu überprüfen:

Nr.	Eingabe						erwartete Ausgabe
	α_{Soll} [°]	$\Delta\alpha$ [°]	Δv [km/h]	v_{Mess} [km/h]	v [km/h]	α [°]	Messdatenaufzeichnung
1	0	0,2	2	50	51 51 51	0 0,1 0,1	ja
2	2	0,2	2	50	51 51 51	2 1,9 2,1	ja
3	0	0,2	2	50	51 51 51	0 0,3 0,1	nein $\Delta\alpha$
4	0	0,2	2	50	51 51 51	0 0,3 0,1	nein $\Delta\alpha$
5	0	0,2	2	50	51 51 53	0 0 0,1	nein Δv
6	0	0,2	2	50	51 51	0 0	nein

					53	0,1	Δv
--	--	--	--	--	----	-----	------------

E2.3.2. Modultestprotokoll: Messung

Der Modultest des Moduls Messung wurde entsprechend der Modultestspezifikation durchgeführt.

Die folgenden Testfälle wurden überprüft:

Nr.	Eingabe						erwartete Ausgabe	tatsächliche Ausgabe
	α_{Soll} [°]	$\Delta\alpha$ [°]	Δv [km/h]	v_{Mess} [km/h]	v [km/h]	α [°]	Messdatenauf- zeichnung	Messdatenauf- zeichnung
1	0	0,2	2	50	51 51 51	0 0,1 0,1	ja	ja
2	2	0,2	2	50	51 51 51	2 1,9 2,1	ja	ja
3	0	0,2	2	50	51 51 51	0 0,3 0,1	nein $\Delta\alpha$	nein $\Delta\alpha$
4	0	0,2	2	50	51 51 51	0 0,3 0,1	nein $\Delta\alpha$	nein $\Delta\alpha$
5	0	0,2	2	50	51 51 53	0 0 0,1	nein Δv	nein Δv
6	0	0,2	2	50	51 51 53	0 0 0,1	nein Δv	nein Δv

E2.4. Statusanzeigen

E2.4.1. Modultestvorschrift: Statusanzeigen

Das Modul Statusanzeigen hat die Eingänge Bordspannung, Hydraulikdruck, Hydrauliktemperatur, CRIOTemperatur, Geschwindigkeit, Radlenkwinkel am Messrad, Radlenkwinkel am

Referenzrad, Schräglaufwinkel, Sollschräglaufwinkel, Drehzahl des Messrades und Proportionalverstärkerstufe sowie dieselben Ausgänge plus Schwimmwinkel.

Bei den Größen, bei denen die Kalibrierdaten fehlen, wird der Offset = 0 und der Faktor = 1 verwendet.

Die folgenden Testfälle sind zu überprüfen:

	Eingabe	erwartete Ausgabe
Nr.	Bordspannung [V]	Anzeige Bordspannung [V]
1	6,6	12
2	6,4	11,5
3	6,1	11

	Eingabe	erwartete Ausgabe
Nr.	Hydraulikdruck [V]	Anzeige Hydraulikdruck [bar]
1	0	0
2	4,3	860
3	10	2000

Die Kalibrierung fehlt, daher erfolgt die Ausgabe als Spannung:

	Eingabe	erwartete Ausgabe
Nr.	Hydrauliktemperatur[V]	Anzeige Hydrauliktemperatur [V]
1	0	0
2	4,8	4,8
3	10	10

	Eingabe	erwartete Ausgabe
Nr.	CRIOTemperatur [°C]	Anzeige CRIOTemperatur [°C]
1	20	20
2	23	23

	Eingabe	erwartete Ausgabe
Nr.	Geschwindigkeit [V]	Anzeige Geschwindigkeit [km/h]
1	-5	0,25
2	0	125
3	5	250

	Eingabe	erwartete Ausgabe
Nr.	Drehzahl [N]	Anzeige Drehzahl [N]
1	45	45
2	60	60
3	61	61

	Eingabe	erwartete Ausgabe
Nr.	Verstärkerstufe	Anzeige Verstärkerstufe
1	1	1
2	2	2
3	5	5

Die Kalibrierung fehlt, daher erfolgt die Ausgabe als Spannung:

	Eingabe	erwartete Ausgabe
Nr.	RadlenkwinkelRR [V]	Anzeige RadlenkwinkelRR [V]
1	-10	-10
2	-4,8	-4,8
3	0	0
4	4,8	4,8
5	10	10

Die Kalibrierung fehlt, daher erfolgt die Ausgabe als Spannung:

	Eingabe			erwartete Ausgabe		
Nr.	δ_{MR} [V]	α [V]	α_{Soll} [V]	Anzeige δ_{MR} [°]	Anzeige α [°]	Anzeige α_{Soll} [°]
1	-10	-10	-10	-10	-10	-10
2	-4,8	-4,8	-4,8	-4,8	-4,8	-4,8
3	0	0	0	0	0	0
4	4,8	4,8	4,8	4,8	4,8	4,8
5	10	10	10	10	10	10

E2.4.2. Modultestprotokoll: Statusanzeigen

Das Modul Statusanzeigen wurde entsprechend dessen Modulspezifikation getestet.

Die folgenden Testfälle wurden dabei verwendet:

	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
Nr.	Bordspannung [V]	Bordspannung [V]	Bordspannung [V]
1	6,6	12	12
2	6,4	11,5	11,5
3	6,1	11	11

	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
Nr.	Hydraulikdruck [V]	Hydraulikdruck [bar]	Hydraulikdruck [bar]
1	0	0	0
2	4,3	860	860
3	10	2000	2000

fehlt Kalibrierung daher Ausgabe als Spannung:

	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
Nr.	Hydrauliktemperatur[V]	Hydrauliktemperatur [V]	Hydrauliktemperatur [V]

Modultests

1	0	0	0
2	4,8	4,8	4,8
3	10	10	10

	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
Nr.	CRIOTemperatur [°C]	CRIOTemperatur [°C]	CRIOTemperatur [°C]
1	20	20	20
2	23	23	23

	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
Nr.	Geschwindigkeit [V]	Geschwindigkeit [km/h]	Geschwindigkeit [km/h]
1	-5	0,25	0,25
2	0	125	125
3	5	250	250

	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
Nr.	Drehzahl [N]	Drehzahl [N]	Drehzahl [N]
1	45	45	45
2	60	60	60
3	61	61	61

	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
Nr.	Verstärkerstufe	Verstärkerstufe	Verstärkerstufe
1	1	1	1
2	2	2	2
3	5	5	5

Die Kalibrierung fehlt, daher erfolgt die Ausgabe als Spannung:

	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
Nr.	RadlenkwinkelRR [V]	RadlenkwinkelRR [V]	RadlenkwinkelRR [V]
1	-10	-10	-10
2	-4,8	-4,8	-4,8
3	0	0	0
4	4,8	4,8	4,8
5	10	10	10

Die Kalibrierung fehlt, daher erfolgt die Ausgabe als Spannung:

Nr.	δ_{MR} [V]	α [V]	α_{Soll} [V]	δ_{MR} [°]	α [°]	α_{Soll} [°]	δ_{MR} [°]	α [°]	α_{Soll} [°]
1	-10	-10	-10	-10	-10	-10	-10	-10	-10
2	-4,8	-4,8	-4,8	-4,8	-4,8	-4,8	-4,8	-4,8	-4,8
3	0	0	0	0	0	0	0	0	0
4	4,8	4,8	4,8	4,8	4,8	4,8	4,8	4,8	4,8
5	10	10	10	10	10	10	10	10	10

E2.5. Warnung

E2.5.1. Modultestvorschrift: Warnung

Das Modul Warnung hat die Eingänge Bordspannung, Hydraulikdruck, Hydrauliktemperatur und CRIOTemperatur je vom Typ Integer, sowie die Eingänge negativerEndschalterRR, positiverEndschalterRR, negativerEndschalterMR und positiverEndschalterMR je vom Typ Boolean.

Die folgenden Testfälle sind zu überprüfen:

	Eingabe	erwartete Ausgabe
Nr.	Bordspannung	Anzeige Bordspannung
1	6,52	grün
2	6,40	grün

3	6,39	gelb
4	6,10	gelb
5	6,09	rot
6	5,90	rot
7	5,89	herunterfahren

	Eingabe	erwartete Ausgabe
Nr.	Hydraulikdruck	Anzeige Hydraulikdruck
1	300	grün
2	320	grün
3	321	gelb
4	380	gelb
5	381	rot
6	400	rot

	Eingabe	erwartete Ausgabe
Nr.	Hydrauliktemperatur	Anzeige Hydrauliktemperatur
1	45	grün
2	70	grün
3	71	gelb
4	75	gelb
5	76	rot
6	80	rot

	Eingabe	erwartete Ausgabe
Nr.	CRIOTemperatur	Anzeige CRIOTemperatur
1	45	grün
2	60	grün

3	61	gelb
4	64	gelb
5	65	rot
6	67	rot
7	68	herunterfahren

	Eingabe	erwartete Ausgabe
Nr.	positiverEndschalterRR	Anzeige positiverEndschalterRR
1	0	grün
2	1	rot

	Eingabe	erwartete Ausgabe
Nr.	negativerEndschalterRR	Anzeige negativerEndschalterRR
1	0	grün
2	1	rot

	Eingabe	erwartete Ausgabe
Nr.	positiverEndschalterMR	Anzeige positiverEndschalterMR
1	0	grün
2	1	rot

	Eingabe	erwartete Ausgabe
Nr.	negativerEndschalterMR	Anzeige negativerEndschalterMR
1	0	grün
2	1	rot

E2.5.2. Modultestprotokoll: Warnung

Die folgenden Testfälle wurden im Rahmen des Modultest entsprechend der Modultestspezifikation überprüft:

	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
Nr.	Bordspannung	Anzeige Bordspannung	Anzeige Bordspannung
1	6,52	grün	grün
2	6,40	grün	grün
3	6,39	gelb	gelb
4	6,10	gelb	gelb
5	6,09	rot	rot
6	5,90	rot	rot
7	5,89	herunterfahren	herunterfahren

	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
Nr.	Hydraulikdruck	Anzeige Hydraulikdruck	Anzeige Hydraulikdruck
1	300	grün	grün
2	320	grün	grün
3	321	gelb	gelb
4	380	gelb	gelb
5	381	rot	rot
6	400	rot	rot

	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
Nr.	Hydrauliktemperatur	Anzeige Hydrauliktemperatur	Anzeige Hydrauliktemperatur
1	45	grün	grün
2	70	grün	grün
3	71	gelb	gelb
4	75	gelb	gelb
5	76	rot	rot
6	80	rot	rot

	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
Nr.	CRIOTemperatur	Anzeige CRIOTemperatur	Anzeige CRIOTemperatur
1	45	grün	grün
2	60	grün	grün
3	61	gelb	gelb
4	64	gelb	gelb
5	65	rot	rot
6	67	rot	rot
7	68	herunterfahren	herunterfahren

	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
Nr.	positiverEndschalter RR	Anzeige positiverEndschalter RR	Anzeige positiverEndschalter RR
1	0	grün	grün
2	1	rot	rot

	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
Nr.	negativerEndschalter RR	Anzeige negativerEndschalter RR	Anzeige negativerEndschalter RR
1	0	grün	grün
2	1	rot	rot

	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
Nr.	positiverEndschalter MR	Anzeige positiverEndschalter MR	Anzeige positiverEndschalter MR
1	0	grün	grün
2	1	rot	rot

	Eingabe	erwartete Ausgabe	tatsächliche Ausgabe
Nr.	negativerEndschalterMR	Anzeige negativerEndschalterMR	Anzeige negativerEndschalterMR
1	0	grün	grün
2	1	rot	rot

Anhang F Integrationstests

F1. FPGA - Priority Loop

F1.1. Integrationstestvorschrift: FPGA – Priority Loop

Das Modul erweiterte Priority Loop entsteht aus den Modulen HPL, LPL und Konfiguration. Es wird geprüft, indem die Messdaten auf dem Realsystem aus dem DMA-Speicher gelesen und angezeigt werden. Des Weiteren wird mittels einer Abfrage geprüft, ob der DMA-Puffer voll war und deshalb keine Daten hineingeschrieben werden konnten.

Außerdem wird auf dem Realsystem der Wert des Proportionalventils und der Nullkalibrierung der Messfelge eingeben und über den DMA-Speicher an das Low Priority Loop Modul gesendet. Der eingestellte Wert des Proportionalventils wird mit einem Voltmeter am entsprechenden Ausgang überprüft. Der Wert der Nullkalibrierung wird durch die entsprechende LED am digitalen Ausgangsmodul überprüft.

Die Messwerte werden erzeugt, indem der zu testende Kanal mit einer Spannung versehen wird. Um Wechselwirkungen auszuschließen werden alle weiteren zu prüfenden Kanäle an einen zweiten Spannungsgenerator angeschlossen und die Spannung während jedes Testwertes verändert. Die Spannung wird mit einem Voltmeter überprüft. Um die erfassten Messdaten auf Korrektheit zu prüfen werden die Messwerte auf dem Realsystem in Volt umgerechnet. Die Messgenauigkeit des Analogmoduls NI-9205 wird mit 1% auf den eingestellten Bereich angegeben und die zur Verfügung stehenden Voltmeter haben eine Messgenauigkeit von 2%, somit ist ein maximaler Gesamtfehler von 3% zu erwarten.

Die folgenden Testfälle sind zu überprüfen, der entsprechende Kanal ist mit der entsprechenden Spannung zu versehen:

HPL-Testfälle:

Nr.	Messgröße	Eingabe [V]					erwartete Ausgabe [V] $\pm 0,15$				
1	$F_{x, KMF}$	-5	-2,3	0	2,3	5	$-5\pm 0,15$	$-2,3\pm 0,15$	$0\pm 0,15$	$2,3\pm 0,15$	$5\pm 0,15$
2	$F_{y, KMF}$	-5	-2,3	0	2,3	5	$-5\pm 0,15$	$-2,3\pm 0,15$	$0\pm 0,15$	$2,3\pm 0,15$	$5\pm 0,15$
3	$F_{z, KMF}$	-5	-2,3	0	2,3	5	$-5\pm 0,15$	$-2,3\pm 0,15$	$0\pm 0,15$	$2,3\pm 0,15$	$5\pm 0,15$
4	$M_{x, KMF}$	-5	-2,3	0	2,3	5	$-5\pm 0,15$	$-2,3\pm 0,15$	$0\pm 0,15$	$2,3\pm 0,15$	$5\pm 0,15$
5	$M_{y, KMF}$	-5	-2,3	0	2,3	5	$-5\pm 0,15$	$-2,3\pm 0,15$	$0\pm 0,15$	$2,3\pm 0,15$	$5\pm 0,15$

6	$M_{z, KMF}$	-5	-2,3	0	2,3	5	$-5\pm 0,15$	$-2,3\pm 0,15$	$0\pm 0,15$	$2,3\pm 0,15$	$5\pm 0,15$
7	$\cos\varphi$	-5	-2,3	0	2,3	5	$-5\pm 0,15$	$-2,3\pm 0,15$	$0\pm 0,15$	$2,3\pm 0,15$	$5\pm 0,15$
8	$\sin\varphi$	-5	-2,3	0	2,3	5	$-5\pm 0,15$	$-2,3\pm 0,15$	$0\pm 0,15$	$2,3\pm 0,15$	$5\pm 0,15$
9	$F_{x, KMD}$	0	2,3	5	7,8	10	$0\pm 0,3$	$2,3\pm 0,3$	$5\pm 0,3$	$7,8\pm 0,3$	$10\pm 0,3$
10	α	-5	-2,3	0	2,3	5	$-5\pm 0,15$	$-2,3\pm 0,15$	$0\pm 0,15$	$2,3\pm 0,15$	$5\pm 0,15$
11	v_{Fzg}	-5	-2,3	0	2,3	5	$-5\pm 0,15$	$-2,3\pm 0,15$	$0\pm 0,15$	$2,3\pm 0,15$	$5\pm 0,15$

LPL-Testfälle:

Nr.	Messgröße	Eingabe [V]					erwartete Ausgabe [V] $\pm 3\%$				
		0	1				0	1			
1	Nullkalib.	0	1				0	1			
2	U_{Prop}	0	2,3	5	7,8	10	$0\pm 0,31$	$2,3\pm 0,31$	$5\pm 0,31$	$7,8\pm 0,31$	$10\pm 0,31$
3	U_{Bord}	0	2,3	5	7,8	10	$0\pm 0,3$	$2,3\pm 0,3$	$5\pm 0,3$	$7,8\pm 0,3$	$10\pm 0,3$
4	$p_{Hydraulik}$	0	2,3	5	7,8	10	$0\pm 0,3$	$2,3\pm 0,3$	$5\pm 0,3$	$7,8\pm 0,3$	$10\pm 0,3$
5	$T_{Hydraulik}$	0	2,3	5	7,8	10	$0\pm 0,3$	$2,3\pm 0,3$	$5\pm 0,3$	$7,8\pm 0,3$	$10\pm 0,3$
7	δ_{MR}	0	2,3	5	7,8	10	$0\pm 0,3$	$2,3\pm 0,3$	$5\pm 0,3$	$7,8\pm 0,3$	$10\pm 0,3$
8	δ_{RR}	0	2,3	5	7,8	10	$0\pm 0,3$	$2,3\pm 0,3$	$5\pm 0,3$	$7,8\pm 0,3$	$10\pm 0,3$

F1.2. Integrationstestprotokoll: FPGA – Priority Loop

Das Modul erweiterte Priority Loop wurde entsprechend der Integrationstestvorschrift getestet.

Die folgenden Testfälle wurden durchgeführt:

Nr.	Messgröße	Eingabe [V]					erwartete Ausgabe [V] $\pm 0,15$ beobachtete Ausgabe				
		-5	-2,3	0	2,3	5	$-5\pm 0,15$	$-2,3\pm 0,15$	$0\pm 0,15$	$2,3\pm 0,15$	$5\pm 0,15$
1	$F_{x, KMF}$	-5	-2,3	0	2,3	5	$-5,016$	$-2,336$	$-0,054$	$2,295$	$4,979$
2	$F_{y, KMF}$	-5	-2,3	0	2,3	5	$-5,016$	$-2,342$	$-0,032$	$2,295$	$4,977$
3	$F_{z, KMF}$	-5	-2,3	0	2,3	5	$-5,016$	$-2,336$	$-0,055$	$2,295$	$4,976$

4	$M_{x, \text{KMF}}$	-5	-2,3	0	2,3	5	$-5 \pm 0,15$ -5,015	$-2,3 \pm 0,15$ -2,335	$0 \pm 0,15$ -0,043	$2,3 \pm 0,15$ 2,296	$5 \pm 0,15$ 4,975
5	$M_{y, \text{KMF}}$	-5	-2,3	0	2,3	5	$-5 \pm 0,15$ -5,016	$-2,3 \pm 0,15$ -2,336	$0 \pm 0,15$ -0,054	$2,3 \pm 0,15$ 2,295	$5 \pm 0,15$ 4,974
6	$M_{z, \text{KMF}}$	-5	-2,3	0	2,3	5	$-5 \pm 0,15$ -5,014	$-2,3 \pm 0,15$ -2,336	$0 \pm 0,15$ -0,033	$2,3 \pm 0,15$ 2,296	$5 \pm 0,15$ 4,974
7	$\cos\varphi$	-5	-2,3	0	2,3	5	$-5 \pm 0,15$ -5,017	$-2,3 \pm 0,15$ -2,336	$0 \pm 0,15$ -0,054	$2,3 \pm 0,15$ 2,293	$5 \pm 0,15$ 4,974
8	$\sin\varphi$	-5	-2,3	0	2,3	5	$-5 \pm 0,15$ -5,017	$-2,3 \pm 0,15$ -2,335	$0 \pm 0,15$ -0,055	$2,3 \pm 0,15$ 2,294	$5 \pm 0,15$ 4,974
9	$F_{x, \text{KMD}}$	0	2,3	5	7,8	10	$0 \pm 0,3$ -0,057	$2,3 \pm 0,3$ 2,272	$5 \pm 0,3$ 4,976	$7,8 \pm 0,3$ 7,76	$10 \pm 0,3$ 9,975
10	α	-5	-2,3	0	2,3	5	$-5 \pm 0,15$ -5,016	$-2,3 \pm 0,15$ -2,336	$0 \pm 0,15$ -0,051	$2,3 \pm 0,15$ 2,295	$5 \pm 0,15$ 4,974
11	v_{fzg}	-5	-2,3	0	2,3	5	$-5 \pm 0,15$ -5,016	$-2,3 \pm 0,15$ -2,333	$0 \pm 0,15$ -0,045	$2,3 \pm 0,15$ 2,295	$5 \pm 0,15$ 4,974

Nr.	Messgröße	Eingabe [V]					erwartete Ausgabe [V] $\pm 0,3$ beobachtete Ausgabe				
		0	1				0	1			
1	Nullkalib.	0	1				0	1			
		0	1				0	1			
2	U_{Prop}	0	2,3	5	7,8	10	$0 \pm 0,31$ 0,000	$2,3 \pm 0,31$ 2,300	$5 \pm 0,31$ 5,002	$7,8 \pm 0,31$ 7,802	$10 \pm 0,31$ 10,003
3	U_{Bord}	0	2,3	5	7,8	10	$0 \pm 0,3$ -0,013	$2,3 \pm 0,3$ 2,258	$5 \pm 0,3$ 4,990	$7,8 \pm 0,3$ 7,770	$10 \pm 0,3$ 9,975
4	$p_{\text{Hydraulik}}$	0	2,3	5	7,8	10	$0 \pm 0,3$ -0,013	$2,3 \pm 0,3$ 2,260	$5 \pm 0,3$ 4,992	$7,8 \pm 0,3$ 7,770	$10 \pm 0,3$ 9,976
5	$T_{\text{Hydraulik}}$	0	2,3	5	7,8	10	$0 \pm 0,3$ -0,013	$2,3 \pm 0,3$ 2,260	$5 \pm 0,3$ 4,990	$7,8 \pm 0,3$ 7,770	$10 \pm 0,3$ 9,975
7	δ_{MR}	0	2,3	5	7,8	10	$0 \pm 0,3$ -0,014	$2,3 \pm 0,3$ 2,263	$5 \pm 0,3$ 4,995	$7,8 \pm 0,3$ 7,781	$10 \pm 0,3$ 9,976
8	δ_{RR}	0	2,3	5	7,8	10	$0 \pm 0,3$ -0,014	$2,3 \pm 0,3$ 2,263	$5 \pm 0,3$ 4,995	$7,8 \pm 0,3$ 7,781	$10 \pm 0,3$ 9,976

Alle Ausgaben liegen innerhalb des Toleranzbereichs der erwarteten Ausgaben und es fand kein Pufferüberlauf statt, damit ist der Integrationstest der erweiterten PL bestanden.

F2. FPGA

F2.1. Integrationstestvorschrift: FPGA

Das integrierte FPGA Modul entsteht durch die Integration der Module PL, Drehzahlberechnung, Schwimmwinkel und Schräglaufwinkel. Es wird geprüft, indem zum Einen die Messdaten in einem Testtreiber auf dem Realtimesystem aus dem DMA-Speicher gelesen und angezeigt werden, zum Anderen die benötigten Daten im Testtreiber eingegeben werden.

Die Drehzahl wird geprüft, indem am Inkrementalgeber gedreht wird. Die restlichen Testfälle bestehen aus einer Untermenge der Testfälle aus den entsprechenden Modul- und Integrationstests.

Die folgenden HPL Testfälle sind zu überprüfen:

Nr.	Messgröße	Eingabe		erwartete Ausgabe	
		[V]	[V]	[V]	[V]
1	$F_{x, KMF}$	-2,3	5	$-2,3 \pm 3\%$	$5 \pm 3\%$
2	$F_{y, KMF}$	-2,3	5	$-2,3 \pm 3\%$	$5 \pm 3\%$
3	$F_{z, KMF}$	-2,3	5	$-2,3 \pm 3\%$	$5 \pm 3\%$
4	$M_{x, KMF}$	-2,3	5	$-2,3 \pm 3\%$	$5 \pm 3\%$
5	$M_{y, KMF}$	-2,3	5	$-2,3 \pm 3\%$	$5 \pm 3\%$
6	$M_{z, KMF}$	-2,3	5	$-2,3 \pm 3\%$	$5 \pm 3\%$
7	$\cos\varphi$	-2,3	5	$-2,3 \pm 3\%$	$5 \pm 3\%$
8	$\sin\varphi$	-2,3	5	$-2,3 \pm 3\%$	$5 \pm 3\%$
9	$F_{x, KMD}$	2,3	10	$2,3 \pm 3\%$	$10 \pm 3\%$
10	α	-2,3	5	$-2,3 \pm 3\%$	$5 \pm 3\%$
11	v_{Fzg}	-2,3	5	$-2,3 \pm 3\%$	$5 \pm 3\%$

Die folgenden LPL Testfälle sind zu überprüfen, der entsprechende Kanal ist mit der entsprechenden Spannung zu versehen:

Nr.	Messgröße	Eingabe [V]		erwartete Ausgabe [V]	
		0	1	0	1
1	Nullkalibrierung	0	1	0	1
2	U_{Prop}	2,3	7,8	$2,3 \pm 3\%$	$7,8 \pm 3\%$
3	U_{Bord}	2,3	7,8	$2,3 \pm 3\%$	$7,8 \pm 3\%$
4	$p_{Hydraulik}$	2,3	7,8	$2,3 \pm 3\%$	$7,8 \pm 3\%$
5	$T_{Hydraulik}$	2,3	7,8	$2,3 \pm 3\%$	$7,8 \pm 3\%$
7	δ_{MR}	2,3	7,8	$2,3 \pm 3\%$	$7,8 \pm 3\%$
8	δ_{RR}	2,3	7,8	$2,3 \pm 3\%$	$7,8 \pm 3\%$

Die folgenden Schwimmwinkelkompensationstestfälle sind zu überprüfen, der Eingangswert ist entsprechend der Kalibrierdaten in Integer umzurechnen:

Nr.	Eingaben							erwartete Ausgaben	
	aktiv	Endschalter positiv	Endschalter negativ	Toleranzbereich [°]	RadlenkwinkelRR [°]	RadlenkwinkelMR [°]	Schräglaufwinkel [°]	MotorRR positiv	MotorRR negativ
1	0	0	0	x	x	x	x	0	0
2	1	0	0	0,2	x	2	2	0	0
3	1	0	0	0,2	x	2	2,2	1	0
4	1	0	0	0,2	10	2	2,2	0	0
5	1	1	0	0,2	x	2	2,2	0	0
6	1	0	0	0,2	x	2	1,8-2,2	0	0
7	1	0	0	0,2	x	2	2,2	0	1
8	1	0	0	0,2	-10	2	2,2	0	0

Die folgenden Schräglaufwinkeltestfälle sind zu überprüfen, der Eingangswert ist entsprechend der Kalibrierdaten in Integer umzurechnen:

Nr.	Eingaben					erwartete Ausgaben	
	End-	End-	Toleranz-	Sollradlenk	Radlenkwinkel	MotorMR	MotorMR

	schalter positiv	schalter negativ	bereich [°]	-winkel [°]	kelMR [°]	positiv	negativ
1	0	0	0	0	0	0	0
2	0	0	0,1	0	0±0,1	0	0
3	0	0	x	x	<-10	x	0
4	0	0	x	x	>10	0	x
5	0	0	0,1	5	5±0,1	0	0
6	0	0	0,1	5	-10 bis 4,89	1	0
7	0	0	0,1	5	5,11 bis 10	0	1

F2.2. Integrationstestprotokoll: FPGA

Die folgenden Testfälle wurden entsprechend der FPGA- Integrationstestvorschrift durchgeführt.

Die folgenden HPL Testfälle wurden geprüft:

Nr.	Messgröße	Eingaben [V]		Ausgaben [V] erwartet tatsächlich	
		-2,3	5	-2,3±3%	5±3%
1	F _{x, KMF}	-2,3	5	-2,336	4,979
2	F _{x, KMF}	-2,3	5	-2,340	4,977
3	F _{z, KMF}	-2,3	5	-2,336	4,978
4	M _{x, KMF}	-2,3	5	-2,335	4,974
5	M _{y, KMF}	-2,3	5	-2,336	4,974
6	M _{z, KMF}	-2,3	5	-2,335	4,974

7	$\cos\varphi$	-2,3	5	-2,3±3% -2,335	5±3% 4,975
8	$\sin\varphi$	-2,3	5	-2,3±3% -2,336	5±3% 4,974
9	$F_{X, KMD}$	2,3	10	2,3±3% 2,270	10±3% 9,975
10	α	-2,3	5	-2,3±3% -2,336	5±3% 4,974
11	$v_{Anhänger}$	-2,3	5	-2,3±3% -2,335	5±3% 4,976

Die folgenden LPL Testfälle wurden überprüft:

Nr.	Messgröße	Eingaben [V]		Ausgaben [V] erwartet tatsächlich	
		0	1	0	1
1	Null- kalibrierung	0	1	0 0	1 1
2	U_{Prop}	2,3	7,8	2,3±3% 2,258	7,8±3% 7,770
3	U_{Bord}	2,3	7,8	2,3±3% 2,258	7,8±3% 7,770
4	$p_{Hydraulik}$	2,3	7,8	2,3±3% 2,261	7,8±3% 7,778
5	$T_{Hydraulik}$	2,3	7,8	2,3±3% 2,263	7,8±3% 7,778
7	δ_{MR}	2,3	7,8	2,3±3% 2,263	7,8±3% 7,783
8	δ_{RR}	2,3	7,8	2,3±3% 2,263	7,8±3% 7,783

Die folgenden Schwimmwinkelkompensationstestfälle wurden überprüft:

Nr.	Eingaben							Ausgaben erwartet tatsächlich	
	aktiv	End- schalter positiv	End- schalter negativ	Toleranz- bereich [°] [int]	Radlenk- winkelRR [°] [int]	Radlenk- winkelMR [°] [int]	Schräg- laufwinkel [°] [int]	Mo- torRR positiv	Mo- torRR negativ
1	0	0	0	x	x	x	x	0 0	0 0
2	1	0	0	0,2 200320	x	2 18184	2 1700	0 0	0 0
3	1	0	0	0,2 200320	x	2 18184	>2,2 >1860	1 1	0 0
4	1	0	0	0,2 200320	10 27946	2 18184	2,2 >1860	0 0	0 0
5	1	1	0	0,2 200320	x	2 18184	2,2 >1860	0 0	0 0
6	1	0	0	0,2 200320	x	2 18184	1,8-2,2 1540-1860	0 0	0 0
7	1	0	0	0,2 200320	x	2 18184	1,8 <1540	0 0	1 1
8	1	0	0	0,2 200320	-10 2826	2 18184	1,8 <1540	0 0	0 0

Die folgenden Schräglaufwinkeltestfälle wurden überprüft:

Nr.	Eingaben					erwartete Ausgaben tatsächliche Ausgaben	
	End- schalter positiv	End- schalter negativ	Toleranz- bereich [°] [int]	Sollradlenk -winkel [°] [int]	Radlenkwinkel [°] [int]	MotorMR positiv	MotorMR negativ
1	0	0	0	0	0	0	0

			0	15780	15780	0	0
2	0	0	0,1 123	0 15780	0±0,1 15657 bis 15903	0 0	0 0
3	0	0	x	x	<-10 <3513	x x	0 0
4	0	0	x	x	>10 >28046	0 0	x x
5	0	0	0,1 123	5 21913	5±0,1 21790 bis 22036	0 0	0 0
6	0	0	0,1 123	5 21913	-10 bis 4,89 3513 bis 21790	1 1	0 0
7	0	0	0,1 123	5 21913	5,11 bis 10 22035 bis 28046	0 0	1 1

F3. Broker

F3.1. Integrationstestvorschrift: Broker

Das Broker Modul entsteht durch die Integration der Module NPL und TCL. Vor diesem Test ist es erforderlich, dass das FPGA-Modul integriert und getestet ist. Es wird geprüft, indem zum Einen die Messdaten in einem Testtreiber auf dem Hostsystem aus der TCP/IP-Verbindung gelesen und angezeigt werden, zum Anderen die benötigten Daten im Testtreiber eingegeben werden.

Die Drehzahl wird geprüft, indem am Inkrementalgeber gedreht wird. Die restlichen Testfälle bestehen aus einer Untermenge der Testfälle aus dem Integrationstest des FPGA-Moduls. Dabei darf kein Pufferüberlauf, weder am DMA-Kanaleingang noch an den RT-FIFO-Puffern, auftreten.

Die folgenden Testfälle sind zu überprüfen, der entsprechende Kanal ist mit der entsprechenden Spannung zu versehen:

Nr.	Messgröße	Eingabe [V]		erwartete Ausgabe [V]	
		-2,3	5	-2,3±3%	5±3%
1	F _{x, KMF}	-2,3	5	-2,3±3%	5±3%
2	F _{y, KMF}	-2,3	5	-2,3±3%	5±3%
3	F _{z, KMF}	-2,3	5	-2,3±3%	5±3%
4	M _{x, KMF}	-2,3	5	-2,3±3%	5±3%

5	$M_{y, \text{KMF}}$	-2,3	5	$-2,3 \pm 3\%$	$5 \pm 3\%$
6	$M_{z, \text{KMF}}$	-2,3	5	$-2,3 \pm 3\%$	$5 \pm 3\%$
7	$\cos\varphi$	-2,3	5	$-2,3 \pm 3\%$	$5 \pm 3\%$
8	$\sin\varphi$	-2,3	5	$-2,3 \pm 3\%$	$5 \pm 3\%$
9	F_x, KMD	2,3	10	$2,3 \pm 3\%$	$10 \pm 3\%$
10	α	-2,3	5	$-2,3 \pm 3\%$	$5 \pm 3\%$
11	v_{Fzg}	-2,3	5	$-2,3 \pm 3\%$	$5 \pm 3\%$
12	U_{Prop}	2,3	7,8	$2,3 \pm 3\%$	$7,8 \pm 3\%$
13	U_{Bord}	2,3	7,8	$2,3 \pm 3\%$	$7,8 \pm 3\%$
14	$p_{\text{Hydraulik}}$	2,3	7,8	$2,3 \pm 3\%$	$7,8 \pm 3\%$
15	$T_{\text{Hydraulik}}$	2,3	7,8	$2,3 \pm 3\%$	$7,8 \pm 3\%$
16	δ_{MR}	2,3	7,8	$2,3 \pm 3\%$	$7,8 \pm 3\%$
17	δ_{RR}	2,3	7,8	$2,3 \pm 3\%$	$7,8 \pm 3\%$
18	Nullkali- brierung	0	1	0	1
19	SWK aktiv	0	1	0	1
20	Proportio- nalventil	2,1	5,6	$2,1 \pm 3,1\%$	$5,6 \pm 3,1\%$

F3.2. Integrationstestprotokoll: Broker

Die folgenden Testfälle wurden entsprechend der Integrationstestvorschrift des Brokermoduls überprüft, der entsprechende Kanal wurde mit der entsprechenden Spannung versehen:

Nr.	Messgröße	Eingabe [V]		Ausgabe [V]	
				erwartet	tatsächlich
1	F_x, KMF	-2,3	5	$-2,3 \pm 3\%$ -2,335	$5 \pm 3\%$ 4,974
2	F_y, KMF	-2,3	5	$-2,3 \pm 3\%$ -2,335	$5 \pm 3\%$ 4,972
3	F_z, KMF	-2,3	5	$-2,3 \pm 3\%$	$5 \pm 3\%$

				-2,335	4,972
4	$M_{x, \text{KMF}}$	-2,3	5	-2,3±3% -2,338	5±3% 4,972
5	$M_{y, \text{KMF}}$	-2,3	5	-2,3±3% -2,338	5±3% 4,972
6	$M_{z, \text{KMF}}$	-2,3	5	-2,3±3% -2,335	5±3% 4,972
7	$\cos\varphi$	-2,3	5	-2,3±3% -2,335	5±3% 4,972
8	$\sin\varphi$	-2,3	5	-2,3±3% -2,337	5±3% 4,974
9	$F_{x, \text{KMD}}$	2,3	10	2,3±3% 2,277	10±3% 7,76
10	α	-2,3	5	-2,3±3% -2,335	5±3% 4,974
11	v_{Fzg}	-2,3	5	-2,3±3% -2,336	5±3% 4,974
12	U_{Prop}	2,3	7,8	2,3±3% 2,278	7,8±3% 7,79
13	U_{Bord}	2,3	7,8	2,3±3% 2,279	7,8±3% 7,76
14	$p_{\text{Hydraulik}}$	2,3	7,8	2,3±3% 2,279	7,8±3% 7,76
15	$T_{\text{Hydraulik}}$	2,3	7,8	2,3±3% 2,276	7,8±3% 7,78
16	δ_{MR}	2,3	7,8	2,3±3% 2,272	7,8±3% 7,76
17	δ_{RR}	2,3	7,8	2,3±3%	7,8±3%

				2,272	7,76
18	Nullkali- brierung	0	1	0 0	1 1
19	SWK aktiv	0	1	0 0	1 1
20	Proportio- nalventil	2,1	5,6	2,1±3,1% 2,122	5,6±3,1% 5,632

F4. Erweiterte Anzeige

F4.1. Integrationstestvorschrift: erweiterte Anzeige

Das erweiterte Statusanzeigenmodul besteht aus den Modulen Statusanzeigen und Warnungen. Als Testfälle dienen eine Untermenge der Testfälle aus den entsprechenden Modultests. Da zum Zeitpunkt der Testdurchführung die Kalibrierdaten des Hydrauliktemperatursensors und der Radlenkwinkel fehlten, wurde zum Testen der Hydrauliktemperaturfaktor mit 10 angenommen und die Radlenkwinkel in Volt ausgegeben.

Die folgenden Statusanzeigen Testfälle sind zu überprüfen:

Nr.	Eingabefeld	Eingabe	erwartete Ausgabe	
1	Bordspannung	6,4 [V]	grün	11,5 [V]
2	Bordspannung	5,89 [V]	herunterfahren	
3	Hydraulikdruck	4,3 [V]	rot	860 [bar]
4	Hydraulikdruck	1,1 [V]	grün	210 [bar]
5	Hydrauliktemperatur	7 [V]	grün	70 [°C] *
6	Hydrauliktemperatur	7,6 [V]	rot	76 [°C] *
7	CRIOTemperatur	T_{CRIO} [°C]	grün	20-23 [°C]
8	CRIOTemperatur	68 [°C]	herunterfahren	68 [°C]
9	Geschwindigkeit	2 [V]		40 [km/h]
10	Drehzahl	> 0 [N]		> 0 [N]
11	Verstärkerstufe	2		2
12	RadlenkwinkelMR	-3,2 [V]		-3,2 [V] *
13	RadlenkwinkelRR	-4,8 [V]		-4,8 [V] *

14	positiverEndschalterRR	0	grün	
15	positiverEndschalterRR	1	rot	
16	negativerEndschalterRR	0	grün	
17	negativerEndschalterRR	1	rot	
18	positiverEndschalterMR	0	grün	
19	positiverEndschalterMR	1	rot	
20	negativerEndschalterMR	0	grün	
21	negativerEndschalterMR	1	rot	

* = noch nicht kalibriert

F4.2. Integrationstestprotokoll: erweiterte Anzeige

Das erweiterte Statusanzeigenmodul wurde entsprechend seiner Integrationstestvorschrift getestet.

Die folgenden Testfälle wurden dabei geprüft:

Nr.	Eingabefeld	Eingabe	erwartete Ausgabe tatsächliche Ausgabe	
1	Bordspannung	6,4 [V]	grün grün	11,5 [V] 11,5 [V]
2	Bordspannung	5,89 [V]	herunterfahren herunterfahren	
3	Hydraulikdruck	4,3 [V]	rot rot	860 [bar] 861 [bar]
4	Hydraulikdruck	1,1 [V]	grün grün	210 [bar] 210 [bar]
5	Hydrauliktemperatur	7 [V]	grün grün	70 [°C] 70 [°C]
6	Hydrauliktemperatur	7,6 [V]	rot rot	76 [°C] 76 [°C]
7	CRIOTemperatur	T _{CRIO} [°C]	grün	20-23 [°C]

			grün	22 [°C]
8	CRIOTemperatur	68 [°C]	herunterfahren	68 [°C]
9	Geschwindigkeit	2 [V]		40 [km/h]
10	Drehzahl	> 0 [N]		> 0 [N]
11	Verstärkerstufe	2		2
12	RadlenkwinkelMR	-3,2 [V]		-3,2 [V]
13	RadlenkwinkelRR	-4,8 [V]		-4,8 [V]
14	positiverEndschalterRR	0	grün grün	
15	positiverEndschalterRR	1	rot rot	
16	negativerEndschalterRR	0	grün grün	
17	negativerEndschalterRR	1	rot rot	
18	positiverEndschalterMR	0	grün grün	
19	positiverEndschalterMR	1	rot rot	
20	negativerEndschalterMR	0	grün grün	
21	negativerEndschalterMR	1	rot rot	

F5. Host - Send

F5.1. Integrationstestvorschrift: Host – Send

Das Modul Host-Send entsteht aus den Modulen Einstellungen und Kalibrierung zusammen mit der Netzwerkkommunikation. Vorbedingung ist, dass die Netzwerkkommunikation zuvor getestet wurde. Da in der Brokerintegration bereits die Verbindung vom Broker zum FPGA

getestet wurde, genügt es bei der Hostintegration die Verbindung zum Broker zu testen. Hierzu werden die übermittelten Daten auf dem Broker gelesen und ausgegeben um sie mit den Sollwerten zu vergleichen.

Testfall des Startarrays:

Eingaben		erwartete Ausgaben	
Eingaben:	Werte	Startparameterarray	Werte
LPL-Timer-Up [ms]	1000	0: LPL-Timer-Up [ms]	1000
HPL-Timer [µs]	10000	1: HPL-Timer [µs]	10000
LPL-Timer-Down [µs]	10000	2: LPL-Timer-Down [µs]	10000
Schräglauftimer [ms]	100	3: Schräglauftimer [ms]	100
Correvitoffset [°]	0	4: -10°MR-RLW [int]	114149
Correvit Faktor [°/V]	4	5: 10°MR-RLW [int]	1239368
POTI Messrad Offset [°]	-10	6: Offset RLW Messrad	-121759
POTI Messrad Faktor [°/Volt]	4	7: deltaSchräglaufwinkel [int]	380
POTI Referenzrad Offset [°]	-10	8: Schwimmwinkeltimer [ms]	100
POTI Referenzrad Faktor [°/Volt]	4	9: 10°RR-RLW [int]	15166
Schräglaufwinkeltoleranz [°]	1	10: -10°RR-RLW [int]	-54
Schwimmwinkeltoleranz [°]	1,5	11: Offset Schräglaufwinkel Messrad	-58
POTI Messrad Endwinkel +	10	12: Offset Radlenkwinkel Messrad	-121759
POTI Messrad Endwinkel -	-10	13: Faktor Schräglaufwinkel Messrad	-1522
POTI Referenzrad Endwinkel +	10	14: Faktor Radlenkwinkel Messrad	761
POTI Referenzrad Endwinkel -	-10	15: deltaSchwimmwinkel [int]	868682

Testfälle des Onlineeingabearrays:

	Eingaben			erwartete Ausgaben		
	TF 1	TF 2	TF 3	TF 1	TF 2	TF 3
Onlineeingabearray						
1: Propotionalventil [Integer]	100	300	500	100	300	500
2: Sollschräglaufwinkel [Integer]	200	500	-300	200	500	-300
3: Zerokalibrierung	1	0	0	1	0	0

3: Schwimmwinkelkompensation	0	1	1	0	1	1
3: stoppe Broker	1	0	1	1	0	1
3: starte Messung	0	1	0	0	1	0
3: starte Kontrollgrößen	1	0	1	1	0	1
3: Schräglaufwinkel aktiv	0	1	1	0	1	1

F5.2. Integrationstestprotokoll: Host – Send

Das Modul Host-Send wurde entsprechend der Integrationstestvorschrift Host-Send getestet.

Testfall des Startarrays:

Eingaben		Ausgaben	
Eingaben:	Werte	Startparameterarray	erwartet tataächlich
LPL-Timer-Up [ms]	1000	0: LPL-Timer-Up [ms]	1000 1000
HPL-Timer [µs]	10000	1: HPL-Timer [µs]	10000 10000
LPL-Timer-Down [µs]	10000	2: LPL-Timer-Down [µs]	10000 10000
Schräglauftimer [ms]	100	3: Schräglauftimer [ms]	100 100
Correvitoffset [°]	0	4: -10°MR-RLW [int]	114149 114149
Correvit Faktor [°/V]	4	5: 10°MR-RLW [int]	1239368 1239368
POTI Messrad Offset [°]	-10	6: Offset RLW Messrad	-121759 -121759
POTI Messrad Faktor [°/Volt]	4	7: deltaSchräglaufwinkel [int]	380 380
POTI Referenzrad Offset [°]	-10	8: Schwimmwinkeltimer [ms]	100

			100
POTI Referenzrad Faktor [°/Volt]	4	9: 10°RR-RLW [int]	15166 15166
Schräglaufwinkeltoleranz [°]	1	10: -10°RR-RLW [int]	-54 -54
Schwimmwinkeltoleranz [°]	1,5	11: Offset Schräglaufwinkel Messrad	-58 -58
POTI Messrad Endwinkel +	10	12: Offset Radlenkwinkel Messrad	-121759 -121759
POTI Messrad Endwinkel -	-10	13: Faktor Schräglaufwinkel Messrad	-1522 -1522
POTI Referenzrad Endwinkel +	10	14: Faktor Radlenkwinkel Messrad	761 761
POTI Referenzrad Endwinkel -	-10	15: deltaSchwimmwinkel [int]	868682 868682

Testfälle des Onlineeingabearrays:

Onlineeingabearray	Eingaben			erwartete Ausgaben tatsächliche Ausgaben		
	TF 1	TF 2	TF 3	TF 1	TF 2	TF 3
1: Proportionalventil [Integer]	100	300	500	100 100	300 300	500 500
2: Sollschräglaufwinkel [Integer]	200	500	-300	200 200	500 500	-300 -300
3: Zerokalibrierung	1	0	0	1 1	0 0	0 0
3: Schwimmwinkelkompensation	0	1	1	0 0	1 1	1 1

3: stoppe Broker	1	0	1	1	0	1
				1	0	1
3: starte Messung	0	1	0	0	1	0
				0	1	0
3: starte Kontrollgrößen	1	0	1	1	0	1
				1	0	1
3: Schräglaufwinkel aktiv	0	1	1	0	1	1
				0	1	1

F6. Erweiterte R_{dyn} -Berechnung

F6.1. Integrationstestvorschrift: erweiterte R_{dyn} -Berechnung

Das erweiterte R_{dyn} -Berechnungsmodul wird aus den Modulen Host-Send und R_{dyn} -Berechnung zusammengesetzt. Das erweiterte Messungsmodul wird wie folgt getestet.

Da zum Testen die Drehzahl am Inkrementalgeber nicht konstant genug gehalten wird, wird diese als Variable vom Broker übergeben. Der Eingang Schräglaufwinkel sowie der Eingang Correvitgeschwindigkeit wird an den Spannungsgenerator angeschlossen. Der Schräglaufwinkel und die Fahrzeuggeschwindigkeit wird auf dem Host angezeigt. Durch Verändern der beiden Spannungen lassen sich Daten erzeugen die entweder in oder außerhalb der Toleranzen liegen. Für die Berechnung über das Moment werden zusätzlich die Eingänge F_x und M_y mit einer Spannung versorgt, diese werden allerdings konstant gehalten.

Die in der Tabelle aufgezeigten 3 Elemente stehen dabei nur exemplarisch für alle Elemente der Arrays.

		Eingaben								erwartete Ausgabe
Nr.	Mess-felge	$\Delta\alpha$ [°]	Δv [km/h]	v_{Mess} [km/h]	F_x [N]	M_y [Nm]	v [km/h]	α [°]	N [1/s]	R_{dyn}
1	0	0,2	2	50	1000	300	50	0	25	0,277
					1000	300	50	0,1	25	
					1000	300	50	0,1	25	
2	1	0,2	2	50	1000	300	50	0	25	0,333
					1000	300	50	0,1	25	

					1000	300	50	0,1	25	
3	0	0,2	2	50	1000	300	50	0	25	Abbruch
					1000	300	50	0,3	25	
					1000	300	50	0,1	25	
4	1	0,2	2	50	1000	300	50	0	25	Abbruch
					1000	300	50	0,3	25	
					1000	300	50	0,1	25	
5	0	0,2	2	50	1000	300	52	0	25	Abbruch
					1000	300	53	0	25	
					1000	300	53	0,1	25	
6	1	0,2	2	50	1000	300	52	0	25	Abbruch
					1000	300	53	0	25	
					1000	300	53	0,1	25	

F6.2. Integrationstestprotokoll: R_{dyn} -Berechnung

Das erweiterte R_{dyn} -Berechnungsmodul wurde entsprechend der Integrationsvorschrift getestet.

Eingaben										erwartete tatsächliche Ausgabe
Nr.	Mess- felge	$\Delta\alpha$ [°]	Δv [km/h]	v_{Mess} [km/h]	F_x [N]	M_y [Nm]	v [km/h]	α [°]	N [1/s]	R_{dyn}
1	0	0,2	2	50	1000	300	50	0	25	0,277
					1000	300	50	0,1	25	0,277
					1000	300	50	0,1	25	
2	1	0,2	2	50	1000	300	50	0	25	0,333
					1000	300	50	0,1	25	0,329
					1000	300	50	0,1	25	
3	0	0,2	2	50	1000	300	50	0	25	Abbruch
					1000	300	50	0,3	25	Abbruch
					1000	300	50	0,1	25	
4	1	0,2	2	50	1000	300	50	0	25	Abbruch
					1000	300	50	0,3	25	

					1000	300	50	0,1	25	Abbruch
5	0	0,2	2	50	1000	300	52	0	25	Abbruch
					1000	300	53	0	25	Abbruch
					1000	300	53	0,1	25	
6	1	0,2	2	50	1000	300	52	0	25	Abbruch
					1000	300	53	0	25	Abbruch
					1000	300	53	0,1	25	

Bei Testfall 2 wurde nicht genau die erwartete Ausgabe ausgegeben. Der Test ist dennoch bestanden, da sich F_x und M_y nicht ganz genau auf die vorgegebenen Werte einstellen ließ.

F7. Erweiterte Messung

F7.1. Integrationstestvorschrift: erweiterte Messung

Das erweiterte Messungsmodul wird aus den Modulen R_{dyn} -Berechnung und Messung zusammengesetzt. Das erweiterte Messungsmodul wird wie folgt getestet.

Der Eingang Schräglaufwinkel sowie der Eingang Correvitgeschwindigkeit wird an den Spannungsgenerator angeschlossen. Der Schräglaufwinkel und die Fahrzeuggeschwindigkeit wird auf dem Host angezeigt. Durch Verändern der beiden Spannungen lassen sich Daten erzeugen die entweder in oder außerhalb der Toleranzen liegen. Die anderen Eingaben werden auf dem Host eingegeben.

Nr.	Eingabe						erwartete Ausgabe
	α_{Soll} [°]	$\Delta\alpha$ [°]	Δv [km/h]	v_{Mess} [km/h]	v [km/h]	α [°]	Messdatenaufzeichnung
1	0	0,2	2	50	51	0	ja
2	2	0,2	2	50	49	2,1	ja
3	0	0,2	2	50	49	0,3	nein, $\Delta\alpha$
4	2	0,2	2	50	51	1,7	nein, $\Delta\alpha$
5	0	0,2	2	50	53	0	nein, Δv
6	2	0,2	2	50	53	2,1	nein, Δv

F7.2. Integrationstestprotokoll: erweiterte Messung

Das erweiterte Messungsmodul wurde entsprechend der Integrationsvorschrift getestet.

	Eingabe						erwartete tatsächliche Ausgabe
Nr.	α_{Soll} [°]	$\Delta\alpha$ [°]	Δv [km/h]	v_{Mess} [km/h]	v [km/h]	α [°]	Messdatenauf- zeichnung
1	0	0,2	2	50	51	0	ja ja
2	2	0,2	2	50	49	2,1	ja ja
3	0	0,2	2	50	49	0,3	nein, $\Delta\alpha$ nein
4	2	0,2	2	50	51	1,7	nein, $\Delta\alpha$ nein
5	0	0,2	2	50	53	0	nein, Δv nein
6	2	0,2	2	50	53	2,1	nein, Δv nein

Anhang G Systemtests

G1. Systemtestvorschrift des Systemtests „Stand“

1. Kontrollgrößen:

Nr.	Test	Erwartete Ausgabe
1	Bordspannung	Gleich der Bordspannungsanzeige in der Messbox
2	Öldruck	Umgebungsdruck: 1bar \pm 0,2
3	Öltemperatur	Umgebungstemperatur

2. Messfelge + Zerokalibrierung:

Die Messfelge wird in unbelastetem Zustand nullkalibriert und die Messdaten werden unbelastet erfasst.

$$\Rightarrow F_x, F_y, F_z = 0 \text{ N} \Rightarrow M_x, M_y, M_z = 0 \text{ Nm}$$

Danach wird die Messfelge auf den Boden abgelassen.

$$\Rightarrow F_x = 0 \text{ N}; F_y \neq 0 \text{ N (Verzwängung beim Ablassen)}; F_z = \text{ca. } 5000 \text{ N (statische Radlast)}$$

Sturzwinkel des Messrades erhöhen.

$$\Rightarrow F_y \text{ wird größer}$$

3. Kraftmessdose:

Mit dem Körpergewicht auf das Getriebe stellen.

$$\Rightarrow F_x \text{ der Messdose wird sich ändern}$$

4. Potentiometer (Radlenkwinkel) + Endschalter:

Radlenkwinkel werden mittels Handschalter verfahren:

$$\Rightarrow \text{die entsprechende Radlenkwinkelanzeige steigt bzw. fällt}$$

Radlenkwinkel werden bis zu den Endschaltern gefahren

$$\Rightarrow \text{der Wert der Anzeige entspricht dem Maximal-/ Minimalwert aus der Kalibrierung und der entsprechende Endschalter leuchtet auf}$$

5. Drehzahlsensor:

Es wird das Messrad in Fahrtrichtung gedreht
 ⇒ die Drehzahlanzeige steigt

6. Motoren:

Das Schräglaufwinkelsignal wird auf Masse gelegt, damit wird 0 Grad Schräglaufwinkel vom Correvitsensor simuliert.

- Sollschräglaufwinkel von 0° wird in der Software eingestellt:

⇒ Radlenkwinkel am Messrad fährt auf 0°

- Sollschräglaufwinkel von 7° wird in der Software eingestellt:

⇒ Radlenkwinkel am Messrad fährt auf 7°, Referenzrad bewegt sich in die entgegengesetzte Richtung bis zum Ende

- Sollschräglaufwinkel von -7° wird in der Software eingestellt:

⇒ Radlenkwinkel am Messrad fährt auf -7°, Referenzrad bewegt sich in die entgegengesetzte Richtung bis zum Ende

G2. Systemtestprotokoll des Systemtests „Stand“

1. Kontrollgrößen:

Nr.	Test	Erwartete Ausgabe	tatsächliche Ausgabe
1	Bordspannung	Gleich der Bordspannungsanzeige in der Messbox	$U_{\text{Messbox}} = 12,2 \text{ V}$ $U_{\text{Software}} = 12,2 \text{ V}$
2	Öldruck	Umgebungsdruck: 1bar ± 0,2	$p = 1,1 \text{ bar}$
3	Öltemperatur	Umgebungstemperatur	$T_{\text{Hydraulik}} = 23 \text{ °C}$

⇒ Test ist bestanden

2. Messfelge + Zerokalibrierung:

Die Messfelge wurde in unbelasteten Zustand nullkalibriert und die Messdaten wurden in aufgebocktem Zustand des Messrades erfasst.

Nr.	Größe	erwarteter Wert	tatsächlicher Wert
1	F_x	0 N	ca. 40 N

2	F_y	0 N	ca. -50 N
3	F_z	0 N	ca. 65 N
4	M_x	0 Nm	ca. 95 Nm
5	M_y	0 Nm	ca. -115 Nm
6	M_z	0 Nm	ca. 135 Nm

Die Werte der Messgrößen rauschen stark.

Danach wurde die Messfelge auf den Boden abgelassen.

erwartete Werte:

$F_x = 0 \text{ N}$; $F_y \neq 0 \text{ N}$ (Verzwängung beim Ablassen); $F_z = \text{ca. } 5000 \text{ N}$ (statische Radlast)

tatsächliche Werte:

$F_x = \text{ca. } 40 \text{ N}$; $F_y = \text{ca. } 860 \text{ N}$; $F_z = \text{ca. } 4900 \text{ N}$

Sturzwinkel des Messrades wurde erhöht:

F_y wurde größer

⇒ Test ist bestanden

3. Kraftmessdose:

Mit dem Körpergewicht wurde das Getriebe belastet.

F_x der Messdose wurde kleiner

⇒ Test ist bestanden

4. Potentiometer (Radlenkwinkel) + Endschalter:

Radlenkwinkel wurden mittels Handschalter verfahren. Die Anzeigen der Software verhielten sich entsprechend der Bewegungsrichtung der Radlenkwinkelverstellungen. Alle Endschalter wurden angefahren und lösten das Signal in der Software aus, da zuvor die Grenzwinkel der Software auf höhere Werte gestellt wurden. Danach wurden die Grenzwinkel wieder in der Software eingestellt und erneut die Grenzen angefahren. Die Software schaltete die Motoransteuerung wie erwartet vor Erreichen der Endschalter ab.

⇒ Test ist bestanden

5. Drehzahlsensor:

Das Messrad wurde in Fahrtrichtung gedreht \Rightarrow kein Drehzahlsignal

Das Messrad wurde gegen die Fahrtrichtung gedreht \Rightarrow Drehzahlsignal wird erfasst

\Rightarrow Drehzahlsensor in der Elektronik falsch angeschlossen

6. Motoren:

Schräglaufwinkelsignal wurde auf Masse gelegt.

-Sollschräglaufwinkel von 0° wurde in der Software eingestellt \Rightarrow Radlenkwinkel am Messrad fuhr auf 0°

-Sollschräglaufwinkel von 7° wurde in der Software eingestellt \Rightarrow Radlenkwinkel am Messrad fuhr auf 7° , Referenzrad bewegte sich in negative Richtung bis zum Ende

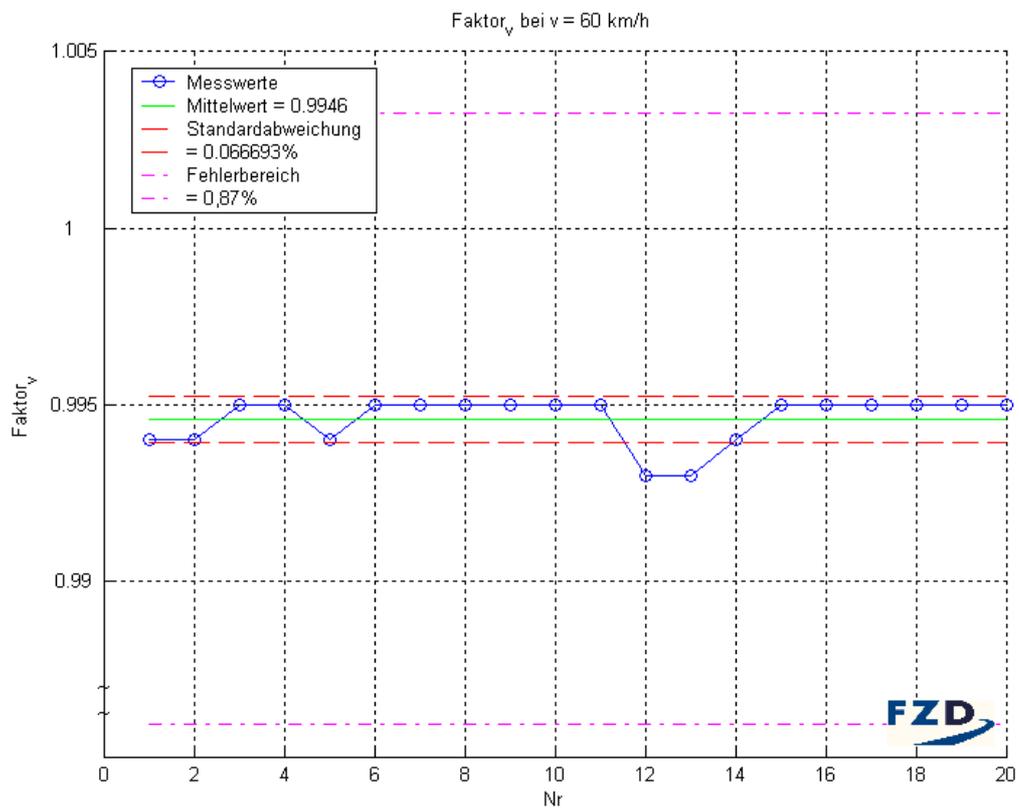
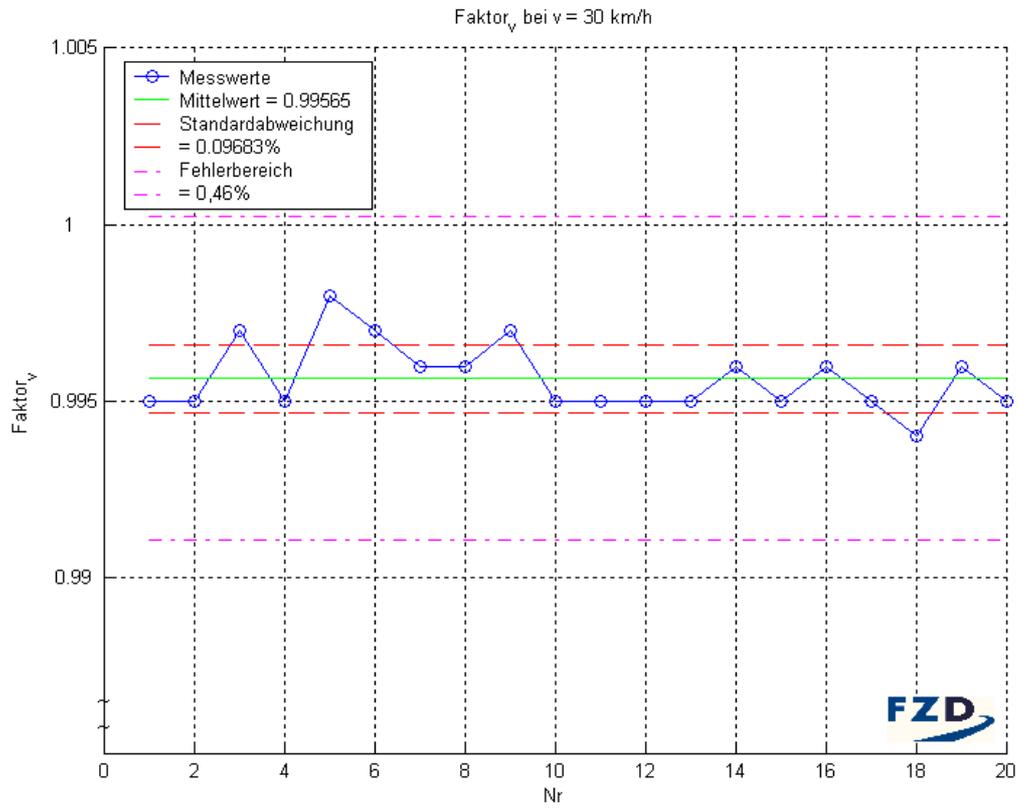
-Sollschräglaufwinkel von -7° wurde in der Software eingestellt \Rightarrow Radlenkwinkel am Messrad fuhr auf -7° , Referenzrad bewegte sich in positive Richtung bis zum Ende

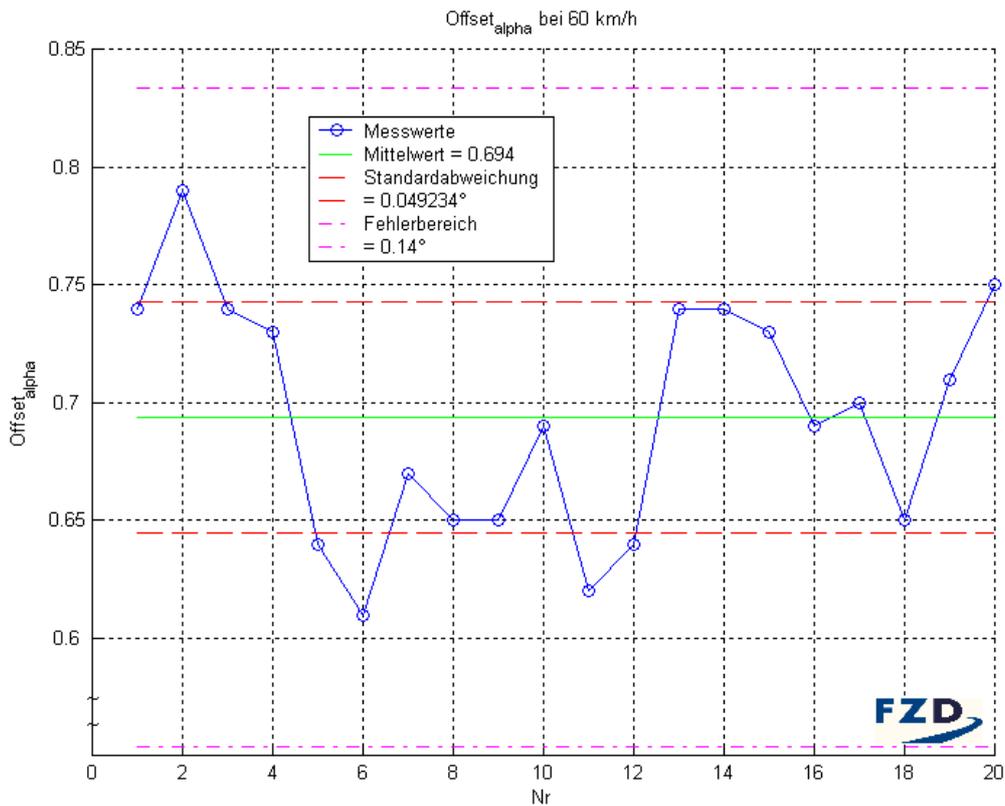
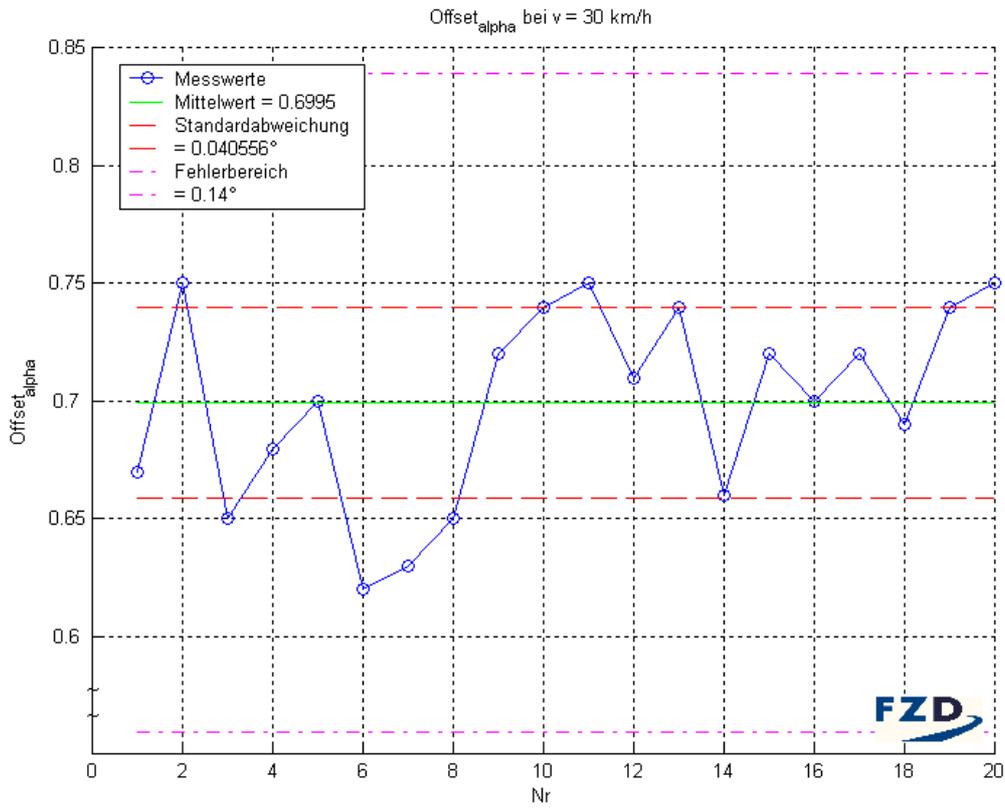
\Rightarrow Test ist bestanden

G3. Correvitkalibrierung

Die nachfolgenden Abbildungen zeigen:

- Faktor_v bei $v = 30$ km/h
- Faktor_v bei $v = 60$ km/h
- Offset _{α} bei $v = 30$ km/h
- Offset _{α} bei $v = 60$ km/h





G4. Systemtestvorschrift des Systemtests „Fahrt“

Verifikation des Systems während der Fahrt durch Systemtests. Die Eingabe erfolgt entsprechend des Systemtestablaufplans. Bei dem Test handelt es sich um einen Black-Box-Test, denn es werden Daten eingegeben und eine Ausgabe erwartet, nicht aber das innere der Software zur Fehlersuche herangezogen.

1. Smoke-Test:

Beim Smoke-Test werden wahllos Eingaben getätigt um das System zum „Abrauchen“ zu bringen. Tritt kein offensichtliches Fehlverhalten auf, ist der Smoke-Test bestanden. Danach können intensivere Tests folgen. Der Smoke-Test ist ein Standardtest in der Softwarequalitätssicherung, der das Ziel hat festzustellen ob das Testobjekt für einen intensiveren Test geeignet ist. Der Nachteil des Smoke-Test ist, dass er kein festes Abbruchkriterium hat.

2. Test der Anzeige von Kontrollparametern:

Der Test erfordert, dass die Kontrollelementanzeigen im entsprechenden Anforderungsbereich liegen. Danach werden die Eingaben, wie in der Tabelle zu sehen ist, vorgenommen und die Anzeigen beobachtet.

Nr.	Test	Messintervall	Eingabe	Erwartete Ausgabe
1	Bordspannung	30 min (manuell)	keine	Bordspannung fällt mit Dauer des Tests
2	Öldruck	1 Messung	Proportionalverstärker erhöhen	Hydrauliköldruck steigt bis μ_{\max}
3	Öltemperatur	30 min (manuell)	Messungen durchführen	Temperaturen steigen während des Messbetriebs
4	CRIO-Temperatur	30 min (manuell)		

3. Test der Anzeige von Statusparametern:

Die Anzeige der Statusparameter werden während der Fahrt getestet. Die Schwimmwinkelkorrektur ist abzuschalten und es wird kein Bremsmoment aufgebracht, d.h. Getriebestellung N. Der Test erfordert, dass alle Anzeigen während der Fahrt einen Wert im zulässigen Bereich aufweisen. Danach werden die Parameteranzeigen entsprechend der folgenden Tabelle getestet. Bei der Schräglaufwinkelverstellung sicherheitshalber in Gradschritten an die $\pm 5^\circ$ herantasten. Beim gemessenen Schräglaufwinkel und beim berechneten Schwimmwinkel ist die eingestellte Toleranz (Δ Schräglaufwinkel, Δ Schwimmwinkel) zulässig. Das Δ ist dabei erst $0,5^\circ$ zu wählen und danach 2° . Aufzeichnung der Werte erfolgt durch Messung.

Nr.	Test	Eingabe	Erwartete Ausgabe
1	Geschwindigkeit des Anhängers und des Messrades	Beschleunigen auf 50 km/h und dann ausrollen lassen	Geschwindigkeitsanzeigen zeigen entsprechend eine steigende / fallende Geschwindigkeit an
2	Radlenkwinkel des Messrades	Sollschräglaufwinkel = 0 Sollschräglaufwinkel = 5 Sollschräglaufwinkel = -5	SchräglaufwinkelMR = $0 \pm \Delta$ SchräglaufwinkelMR = $5 \pm \Delta$ SchräglaufwinkelMR = $-5 \pm \Delta$
3	Sollschräglaufwinkel	Sollschräglaufwinkel = 0 Sollschräglaufwinkel = 5 Sollschräglaufwinkel = -5	Sollschräglaufwinkel = $0 \pm \Delta$ Sollschräglaufwinkel = $5 \pm \Delta$ Sollschräglaufwinkel = $-5 \pm \Delta$
4	Schwimmwinkel des Anhängers	Sollschräglaufwinkel = 0 Sollschräglaufwinkel = 5 Sollschräglaufwinkel = -5	Schwimmwinkel = $0 \pm \Delta$ Schwimmwinkel < 0 Schwimmwinkel > 0

4. Test der Warnungen beim Verlassen von Parametergrenzen:

Um die Bauteilsicherheit zu gewährleisten werden hierzu tiefere Parametergrenzen als eigentlich vorgesehen eingestellt. Dann wird eine Messung gestartet und das Verhalten (Warnungen) beim Erreichen der eingestellten Parametergrenzen geprüft. Die Aufzeichnung erfolgt manuell (es gibt nur bestanden oder nicht bestanden).

Nr.	Test	Eingabe	Erwartete Ausgabe
1	Warnung bei zu hohem Hydraulikdruck	Parametergrenze1 = 250bar Parametergrenze2 = 300bar Messfahrt	Die Anzeige wird „gelb“ beim Erreichen der 1.Parametergrenze und „rot“ bei der 2.
2	Warnung bei zu hoher Hydrauliktemperatur	Parametergrenze1 = 50°C Parametergrenze2 = 60°C Messfahrt	Die Anzeige wird „gelb“ beim Erreichen der 1.Parametergrenze und „rot“ bei der 2.
3	Warnung bei zu niedriger Bordspannung	Parametergrenze1 = 11,8V Parametergrenze2 = 11,7V Messfahrt	Die Anzeige wird „gelb“ beim Erreichen der 1.Parametergrenze und „rot“ bei der 2.
4	Warnung bei zu hoher CRIO-Gehäusetemperatur	Parametergrenze1 = 35°C Parametergrenze2 = 40°C Messfahrt	Die Anzeige wird „gelb“ beim Erreichen der 1.Parametergrenze und „rot“ bei der 2.

5. Test des Herunterfahrens der Messtechnik beim Überschreiten kritischer Grenzen:

Um die Bauteilsicherheit zu gewährleisten werden hierzu tiefere Parametergrenzen als eigentlich vorgesehen eingestellt. Dann wird eine Messung gestartet und das Verhalten (Messdatenaufzeichnung abbrechen, Messdaten sichern und Messhardware herunterfahren) beim Erreichen der eingestellten Parametergrenzen geprüft. Die Aufzeichnung erfolgt manuell (es gibt nur bestanden oder nicht bestanden).

Nr.	Test	Eingabe	Erwartete Ausgabe
1	Herunterfahren bei zu niedriger Bordspannung	Parametergrenze = 11,6V Messfahrt	Messdatenaufzeichnung abbrechen, Messdaten sichern und Messhardware herunterfahren
2	Herunterfahren bei zu hoher CRIO-Temperatur	Parametergrenze = 45°C Messfahrt	Messdatenaufzeichnung abbrechen, Messdaten sichern und Messhardware herunterfahren

6. Test zur Bremsmomentverstellung am Messrad:

Das Fahrzeug befindet sich im bewegten Zustand und das Messrad wird schräglaufwinkel-frei gehalten. Es wird der Proportionalverstärker in festen Stufen vom Minimalwert bis zum Maximalwert verstellt. In Folge dessen wird sich die Bremskraft erhöhen. Dabei ist der Schwimmwinkel zu beobachten der ausgegletzt wird und somit immer wieder gegen Null geht. Die Aufzeichnung erfolgt durch Messung.

Nr.	Eingabe		Erwartete Ausgabe	
	Geschwindigkeit [km/h]	Proportionalverstärker	Bremskraft F_x [N]	Schwimmwinkel [°]
1	30	Stufe 0	erst zuneh- mend bis μ_{max} dann abneh- mend	$0 \pm \Delta$
2	30	Stufe 1		$0 \pm \Delta$
3	30	Stufe 2		$0 \pm \Delta$
4	30	Stufe 3		$0 \pm \Delta$
5	30	Stufe n		$0 \pm \Delta$

7. Test zur Schräglaufverstellung am Messrad:

Das Fahrzeug befindet sich im bewegten Zustand und am Messrad wird kein Bremsmoment eingestellt. Danach wird schrittweise der Radlenkwinkel am Messrad verstellt und mittels Anzeige kontrolliert. Es darf kein Endschalter detektiert werden. Zudem stimmt

der Schräglaufwinkel am Messrad, den der Correvit misst, mit dem Radlenkwinkel am Messrad überein, wenn die Schwimmwinkelkompensation funktioniert. Der Testablauf erfolgt nach folgender Tabelle. Die Aufzeichnung der Messwerte erfolgt manuell. Es besteht die Forderung, dass der Correvisensor für diesen Test kalibriert ist.

Nr.	Eingabe		Erwartete Ausgabe		
	Geschwindigkeit [km/h]	Sollschräglaufwinkel am Messrad [°]	Radlenkwinkel am Messrad [°]	Schräglaufwinkel am Messrad [°]	Schwimmwinkel [°]
1	10	0	0±Δ	0±Δ	0±Δ
2	10	2	2±Δ	2±Δ	0±Δ
3	10	5	5±Δ	5±Δ	0±Δ
4	10	-2	-2±Δ	-2±Δ	0±Δ
5	10	-5	-5±Δ	-5±Δ	0±Δ
6	30	0	0±Δ	0±Δ	0±Δ
7	30	2	2±Δ	2±Δ	0±Δ
8	30	5	5±Δ	5±Δ	0±Δ
9	30	-2	-2±Δ	-2±Δ	0±Δ
10	30	-5	-5±Δ	-5±Δ	0±Δ
11	50	0	0±Δ	0±Δ	0±Δ
12	50	2	2±Δ	2±Δ	0±Δ
13	50	5	5±Δ	5±Δ	0±Δ
14	50	-2	-2±Δ	-2±Δ	0±Δ
15	50	-5	-5±Δ	-5±Δ	0±Δ

8. Test zur Schwimmwinkelkompensation:

Bereits mit Punkt 6 und 7 abgedeckt.

9. Test Drehzahlsignal:

R_{dyn}-Ermittlung von Hand vs. Software.

10. Test R_{dyn} -Berechnung:

Die R_{dyn} -Berechnung ist 20-mal mit 30km/h durchzuführen und die über die Messfelge, sowie die über die Geschwindigkeit berechneten $R_{\text{dyn}s}$ sind zu vergleichen. Die so unterschiedlich ermittelten dynamischen Reifenhalmmesser sind zu vergleichen. Die Aufzeichnung erfolgt manuell.

11. Test Messungsdurchführung und Messdatenaufzeichnung:

Vorbedingung ist, dass die Stufen grob eingestellt werden. Es ist „dynamische Messung“ einzustellen, die Messfahrt zu beginnen. Danach wird auf 30 km/h beschleunigt und die Messung durchfahren. Danach ist eine „statische“ Messung durchzuführen mit $v = 30\text{km/h}$. Nach der Messung ist zu prüfen ob Messdaten in die Messdatei geschrieben wurden. Diese statische Messung ist mit den Grenzgeschwindigkeiten 10km/h und 50km/h zu wiederholen. Alle 3 ermittelten Kurven sind auf Plausibilität zu prüfen. Hierbei darf kein Pufferüberlauf stattfinden, da sonst Daten verloren gegangen sein könnten. Die Aufzeichnung erfolgt automatisch.

12. Performancetest:

Es wird eine Abtastrate von 1000Hz eingestellt und eine Messung durchgeführt. Es ist zu prüfen, ob die mit 1000Hz getaktete Schleife auf dem FPGA, welche die Messdaten in den DMA-Puffer schreibt, auch wirklich mit 1000Hz läuft oder ob es durch Berechnungen nicht möglich ist die 1000Hz zu erreichen. Ein Vergleich mit der Systemclock (40Mhz \Leftrightarrow 25ns) wäre die genaueste Variante, diese ist aber nicht möglich, da ein Schleifenzähler der mit der Clock getaktet ist jeden zur Verfügung stehenden Puffer sofort überlaufen lässt. Deshalb wird eine leere Schleife parallel mitlaufen, welche mit gleicher Frequenz getaktet ist und beide Schleifenzähler werden verglichen. Ein direkter Vergleich mittels einer Division ist nicht möglich, da dies eine Synchronisation der beiden Schleifen erfordert. Hierbei ist darauf zu achten, dass kein Pufferüberlauf eintritt. Dieser Test wird wiederholt und die Abtastrate erhöht bis ein Pufferüberlauf eintritt. Somit kann die maximale Abtastrate ermittelt werden. Die Aufzeichnung erfolgt automatisch.

13. Kompatibilitätstest zu Matlab:

Die Messdatei wird mit Matlab eingelesen.

14. Zuverlässigkeitstest:

Während der Systemverifikation & Validierung ist die Häufigkeit der Nichtverfügbarkeit des Softwaresystems zu erfassen.

G5. Systemtestprotokoll des Systemtests „Fahrt“

Verifikation des Systems während der Fahrt durch Systemtests. Die Eingabe erfolgte entsprechend der Systemtestvorschrift des Systemtests während der Fahrt.

1. Smoke-Test:

Es wurde der Smoke-Test durchgeführt, indem wahllos Eingaben getätigt wurden. Da es während diesem Test zu keinem offensichtlichem Fehlverhalten kam ist der Test bestanden.

2. Test der Anzeige von Kontrollparametern:

Die Kontrollelementanzeigen wurden entsprechend des 2.Tests der Systemtestvorschrift des Systemtests „Fahrt“ durchgeführt.

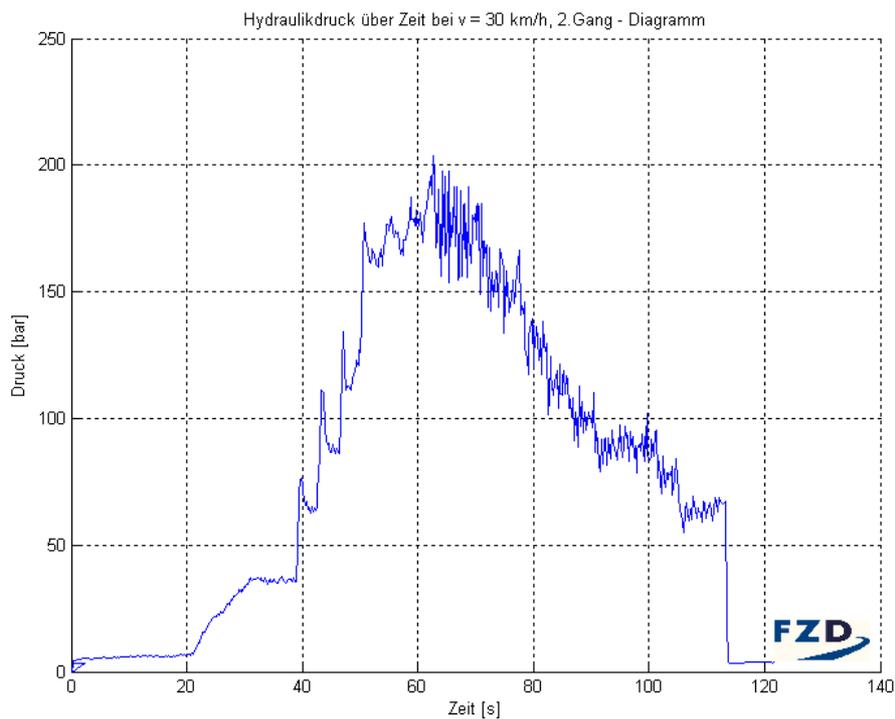
Zeit [min]	Bordspannung [V]	CRIOTemp [°C]	Hydrauliktemp.[°c]
000	12.3	15	20
010	12.3	16	20
020	12.3	18	20
030	12.3	20	21
060	12.2	20	21
090	12.1	20	21
120	12.2	20	21
150	12.1	20	21
180	12.1	20	21
210	12.2	19	21
240	12.1	20	21

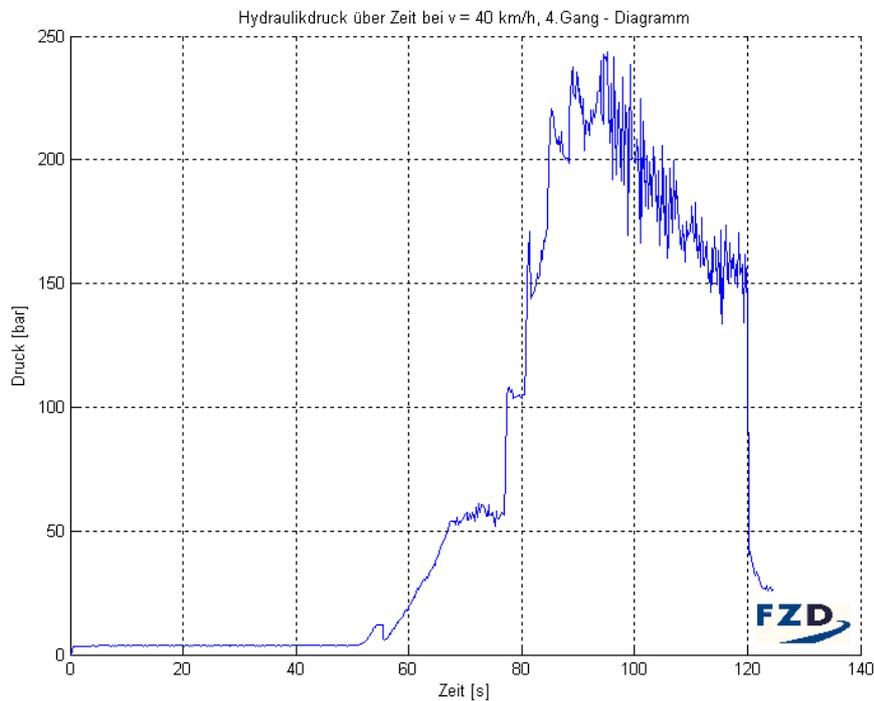
Die in dieser Tabelle protokollierten Werte wurden manuell erfasst.

Wie erwartet fällt die Bordspannung mit der Dauer des Messbetriebs, während die CRIO-Temperatur steigt. Die Hydrauliktemperatur blieb über den kompletten Messbetrieb na-

hezu konstant, bei 21°C, obwohl die Außentemperatur deutlich geringer war. Daher wurde eine neue Kalibrierung durchgeführt, bei der festgestellt wurde, dass tiefere Temperaturen als 23°C in der aktuellen Konfiguration des Temperatursensors nicht gemessen werden können.

Der Hydraulikdruck wurde während der Messung erfasst. Die nachfolgenden Abbildungen zeigen den Hydraulikdruckverlauf während der Messung bei 30km/h und bei 40km/h. Wie zu erwarten steigt der Druck bis μ_{\max} an und fällt danach wieder ab.

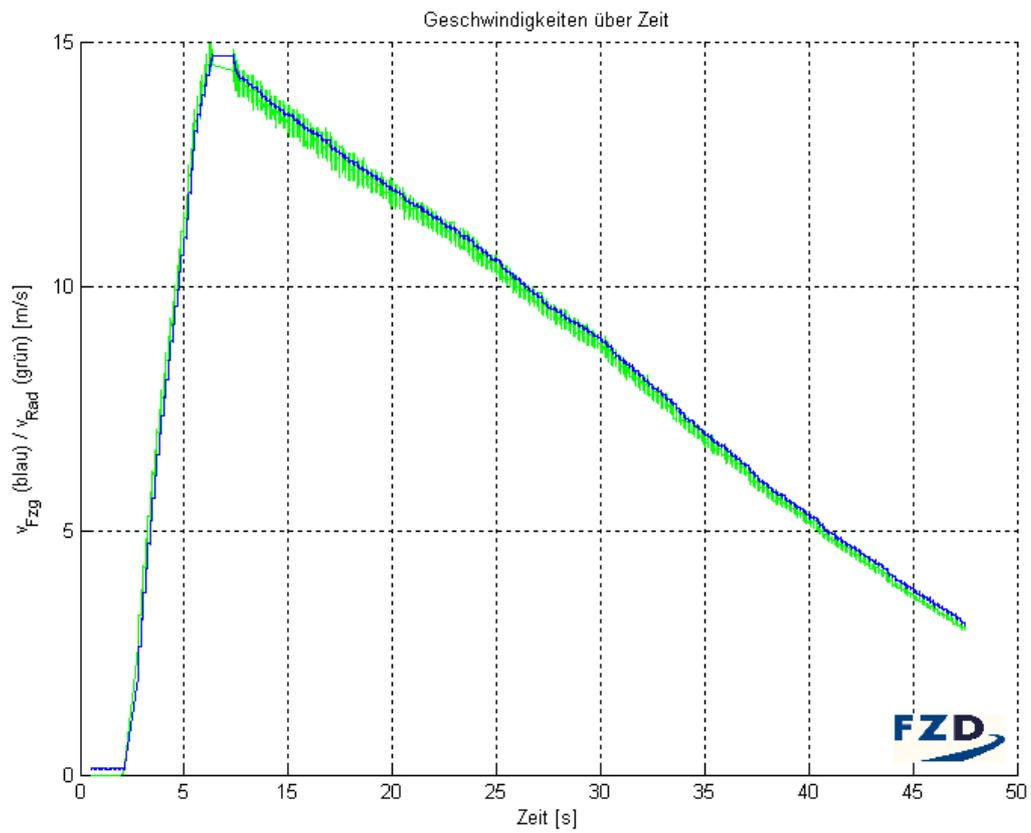


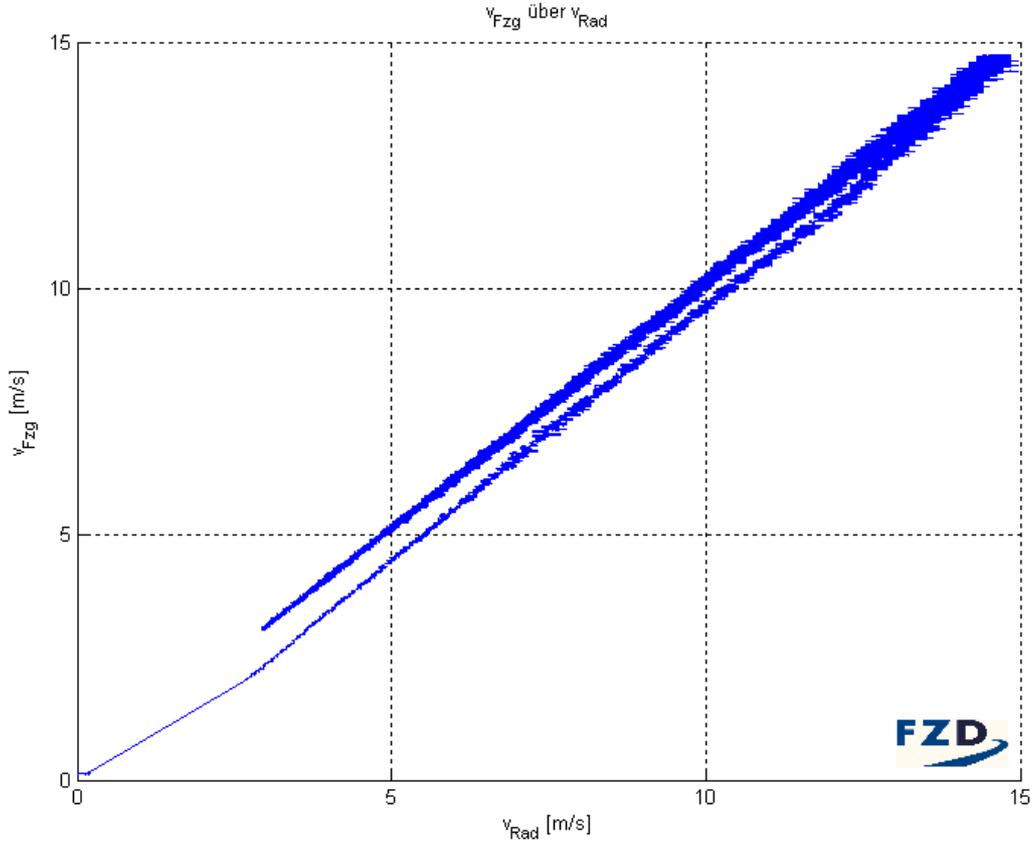


3. Test der Anzeige von Statusparametern:

Die Anzeige der Statusparameter wurde entsprechend der Systemtestvorschrift getestet.

Geschwindigkeitstest: Wie erwartet nimmt die Geschwindigkeit beim Beschleunigen zu bis v_{\max} hier 15 m/s und bei Ausrollen ab. Die folgende Abbildung zeigt die Fahrzeuggeschwindigkeit über der Radgeschwindigkeit. Wie erwartet haben beide den gleichen Verlauf. Die übernächste Abbildung zeigt die 2 Geschwindigkeiten übereinander. Hierbei ist schlupfbedingt eine Hysterese zu erkennen.





Die hier angegebene Toleranz wurde als Schwimmwinkeltoleranz und Schräglaufwinkeltoleranz eingestellt

Nr.	Eingabe			erwartete Ausgabe			
	Soll-schräglaufwinkel	Toleranzen	Kompensation	Radlenkwinkel MR	Schräglaufwinkel	Schwimmwinkel	
1	0	0,5	aktiv	0±0,5 0,0	0±0,5 0,1	0±0,5 -0,1	Radlenkwinkel RR 0,0
2	5	0,5	aktiv	5±0,5 5,0	5±0,5 5,2	0±0,5 0,0	5,2
3	-5	0,5	aktiv	-5±0,5 -5,0	-5±0,5 -5,1	0±0,5 0,1	-7,2

4	0	0,5	inaktiv	0±0,5 0,0	0±0,5 0,2	0±0,5 -0,2	-0,3
5	5	0,5	inaktiv	5±0,5 4,9	>0 1,6	>0 3,5	-0,4
6	-5	0,5	inaktiv	-5±0,5 -4,9	<0 -1,6	<0 -3,3	-0,3
7	0	2,0	aktiv	0±2,0 -0,9	0±2,0 -0,2	0±2,0 0,0	-2,5
8	5	2,0	aktiv	5±2,0 4,1	5±2,0 3,7	0±2,0 0,3	4,5
9	-5	2,0	aktiv	-5±2,0 -4,1	-5±2,0 -3,9	0±2,0 -0,4	-6,3
10	0	2,0	inaktiv	0±2,0 -0,9	0±2,0 -0,2	0±2,0 -0,8	-0,2
11	5	2,0	inaktiv	5±2,0 4,2	>0 1,3	>0 2,8	-0,3
12	-5	2,0	inaktiv	-5±2,0 4,3	<0 1,1	<0 -3,2	-0,2

Die eingestellten Winkel und die Schwimmwinkelkompensation lieferte die Werte die erwartet wurden. Somit ist der Test bestanden.

4. Test der Warnungen beim Verlassen von Parametergrenzen:

Dieser Test wurde, wegen Bauteilschutz, mit den Werten die in der Spalte Eingabe stehen durchgeführt.

Nr.	Test	Eingabe	erwartete Ausgabe
1	Warnung bei zu hohem Hydraulikdruck	Parametergrenze1 = 80bar Parametergrenze2 = 100bar Messfahrt	Die Anzeige wird „gelb“ beim Erreichen der 1.Parametergrenze und „rot“ bei der 2.

2	Warnung bei zu hoher Hydrauliktemperatur	Parametergrenze1 = 20°C Parametergrenze2 = 21°C Messfahrt	Die Anzeige wird „gelb“ beim Erreichen der 1.Parametergrenze und „rot“ bei der 2.
3	Warnung bei zu niedriger Bordspannung	Parametergrenze1 = 12,0V Parametergrenze2 = 11,9V Messfahrt	Die Anzeige wird „gelb“ beim Erreichen der 1.Parametergrenze und „rot“ bei der 2.
4	Warnung bei zu hoher CRIO-Gehäusetemperatur	Parametergrenze1 = 20°C Parametergrenze2 = 21°C Messfahrt	Die Anzeige wird „gelb“ beim Erreichen der 1.Parametergrenze und „rot“ bei der 2.

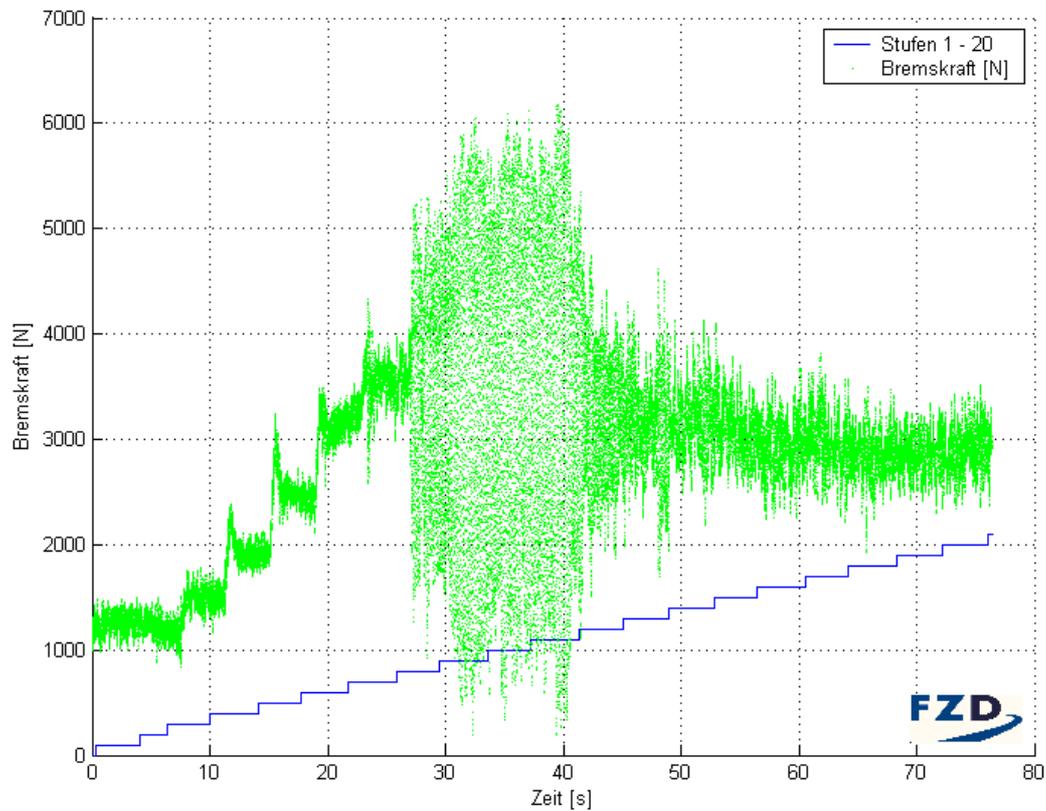
5. Test Herunterfahren der Messtechnik beim Überschreiten kritischer Parametergrenzen:
Dieser Test wurde, wegen Bauteilschutz, mit den Werten die in der Spalte Eingabe stehen durchgeführt.

Nr.	Test	Eingabe	erwartete Ausgabe
1	Herunterfahren bei zu niedriger Bordspannung	Parametergrenze = 11,8V Messfahrt	Messdatenaufzeichnung abbrechen, Messdaten sichern und Messhardware herunterfahren
2	Herunterfahren bei zu hoher CRIO-Temperatur	Parametergrenze1 = 22°C Messfahrt	Messdatenaufzeichnung abbrechen, Messdaten sichern und Messhardware herunterfahren

Test 4 und 5 wurden ohne Beanstandung absolviert.

6. Test zur Bremsmomentverstellung am Messrad:

Der Test wurde wie in der zugehörigen Vorschrift vorgesehen durchgeführt. Der Schwimmwinkel konnte immer ausgegelt werden. Die Messung erfolgte bei 30 km/h im 2. Gang. Das Proportionalventil wurde in 15 Stufen von 1V bis 3,5V verstellt. Die Bremskraft die in der folgenden Abbildung zu sehen ist nimmt wie erwartet bis μ_{\max} zu und danach wieder ab.



7. Test zur Schräglaufverstellung am Messrad:

Dieser Test wurde wie in der Systemtestvorschrift beschrieben durchgeführt, hierbei wurde ein Schräglaufwinkeltoleranz von $0,5^\circ$ und eine Schwimmwinkeltoleranz von 1° eingestellt. Die Regelung erfolgt somit auf $0,25^\circ$ genau bei der Schräglaufeinstellung und $0,5^\circ$ genau bei der Schwimmwinkelausregelung.

Nr.	Eingabe		erwartete Ausgabe tatsächliche Ausgabe		
	Geschwindigkeit [km/h]	Sollschräglaufwinkel am Messrad [$^\circ$]	Radlenkwinkel am Messrad [$^\circ$]	Schräglaufwinkel am Messrad [$^\circ$]	Schwimmwinkel [$^\circ$]
1	10	0	$0 \pm \Delta$	$0 \pm \Delta$	$0 \pm \Delta$
			-0,1	0,1	-0,2
2	10	2	$2 \pm \Delta$	$2 \pm \Delta$	$0 \pm \Delta$
			2,1	2,1	0,1

3	10	5	5±Δ 5,0	5±Δ 5,2	0±Δ 0,1
4	10	-2	-2±Δ -2,0	-2±Δ -1,9	0±Δ 0,1
5	10	-5	-5±Δ -5,0	-5±Δ -5,1	0±Δ -0,3
6	30	0	0±Δ 0,0	0±Δ 0,2	0±Δ 0,2
7	30	2	2±Δ 2,1	2±Δ 1,8	0±Δ -0,2
8	30	5	5±Δ 5,0	5±Δ 4,9	0±Δ 0,2
9	30	-2	-2±Δ -2,1	-2±Δ -1,8	0±Δ 0,1
10	30	-5	-5±Δ -4,9	-5±Δ -4,9	0±Δ -0,2
11	50	0	0±Δ -0,1	0±Δ -0,2	0±Δ -0,3
12	50	2	2±Δ 1,9	2±Δ 1,8	0±Δ 0,2
13	50	5	5±Δ 4,9	5±Δ 5,0	0±Δ 0,0
14	50	-2	-2±Δ -2,0	-2±Δ -1,8	0±Δ -0,1
15	50	-5	-5±Δ -4,9	-5±Δ -5,0	0±Δ -0,3

Die abgelesenen Werte in der Software, welche bei diesen Testfällen eingeregelt wurden entsprechen den erwarteten Werten.

8. Test zur Schwimmwinkelkompensation:

Bereits mit Punkt 6 und 7 abgedeckt.

9. Test Drehzahlsignal:

Zur Verifikation der Drehzahl wurde R_{dyn} sowohl manuell als auch mittels Messung und Offlineauswertung ermittelt.

- manuelle R_{dyn} -Bestimmung:

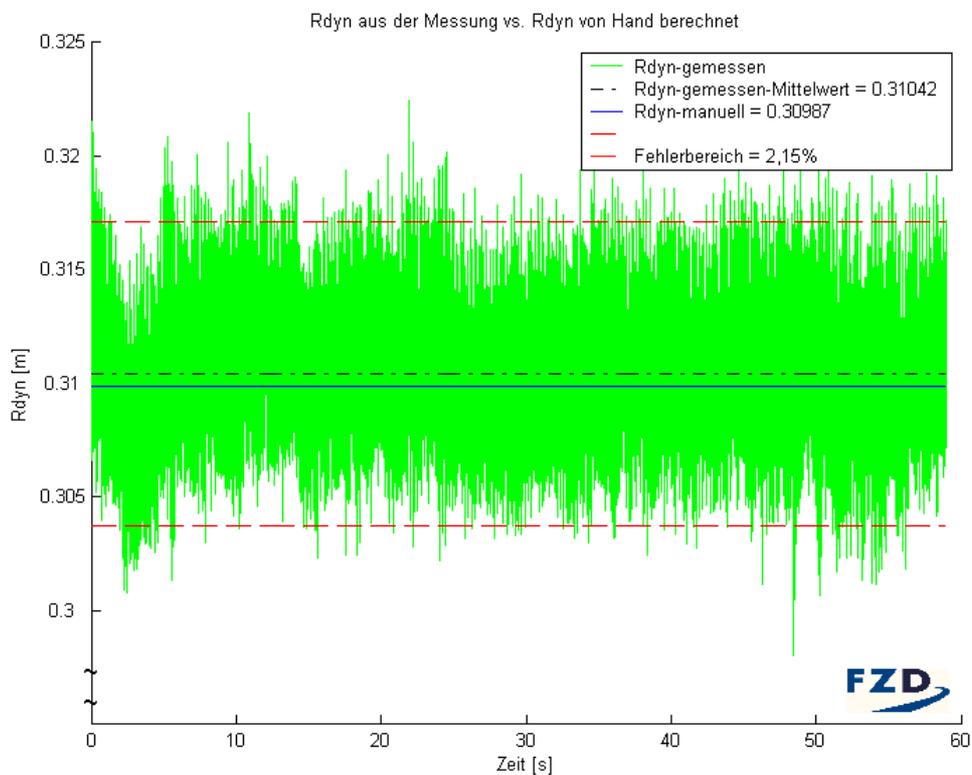
$$U = 10, s = 19,47 \text{ [m]}$$

$$R_{dyn,manuell} = \frac{s}{2 * \pi * U} = 0,30987$$

- 2. R_{dyn} -Bestimmung durch Messung und Offlineauswertung

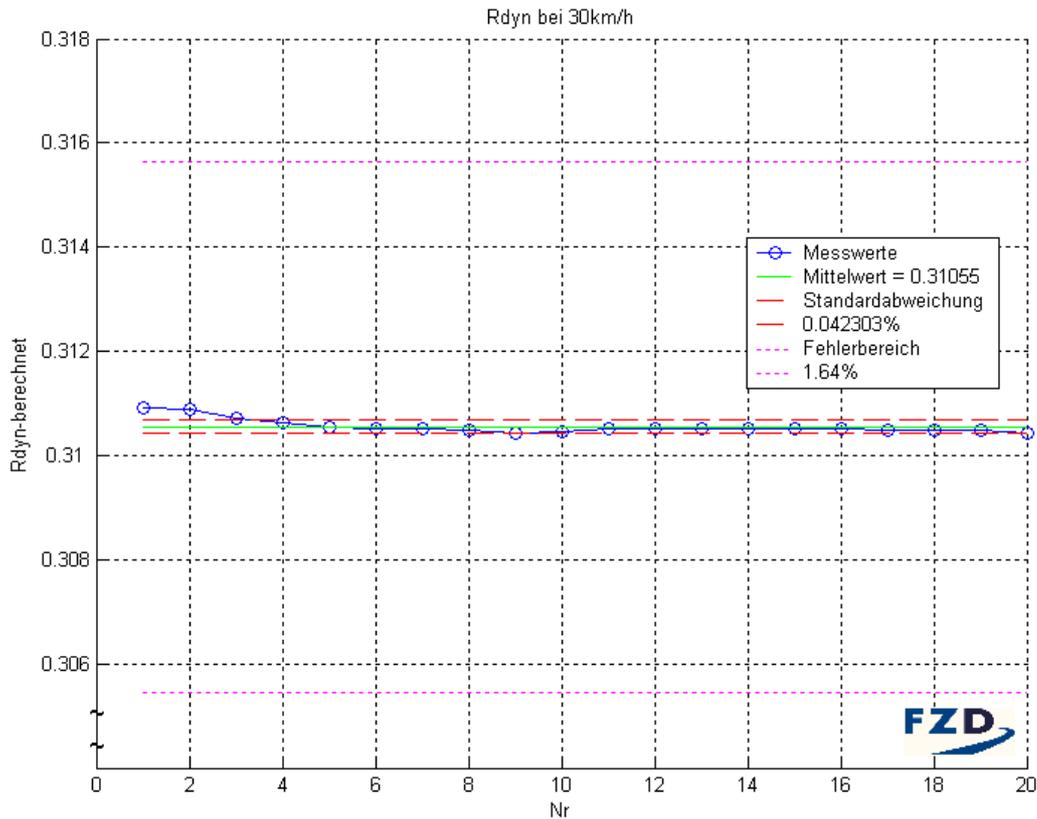
$$R_{dyn,gemessen} = \frac{V_{Fzg}}{n_{Rad} * 2 * \pi}$$

Der Mittelwert von $R_{dyn,gemessen}$ beträgt 0,31042.



10. Test R_{dyn} -Berechnung:

Dieser Test wurde entsprechend der Testspezifikation durchgeführt. Die folgende Grafik zeigt R_{dyn} bei 30 km/h welches mit der Software 20 mal hintereinander bestimmt wurde.



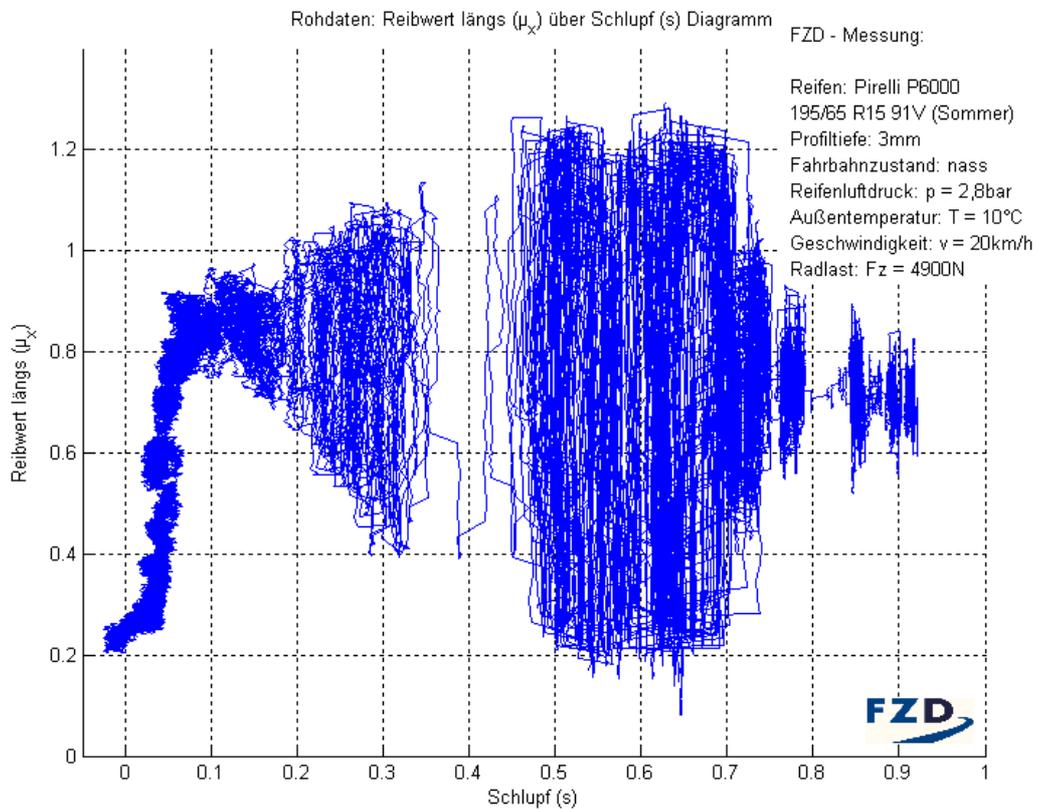
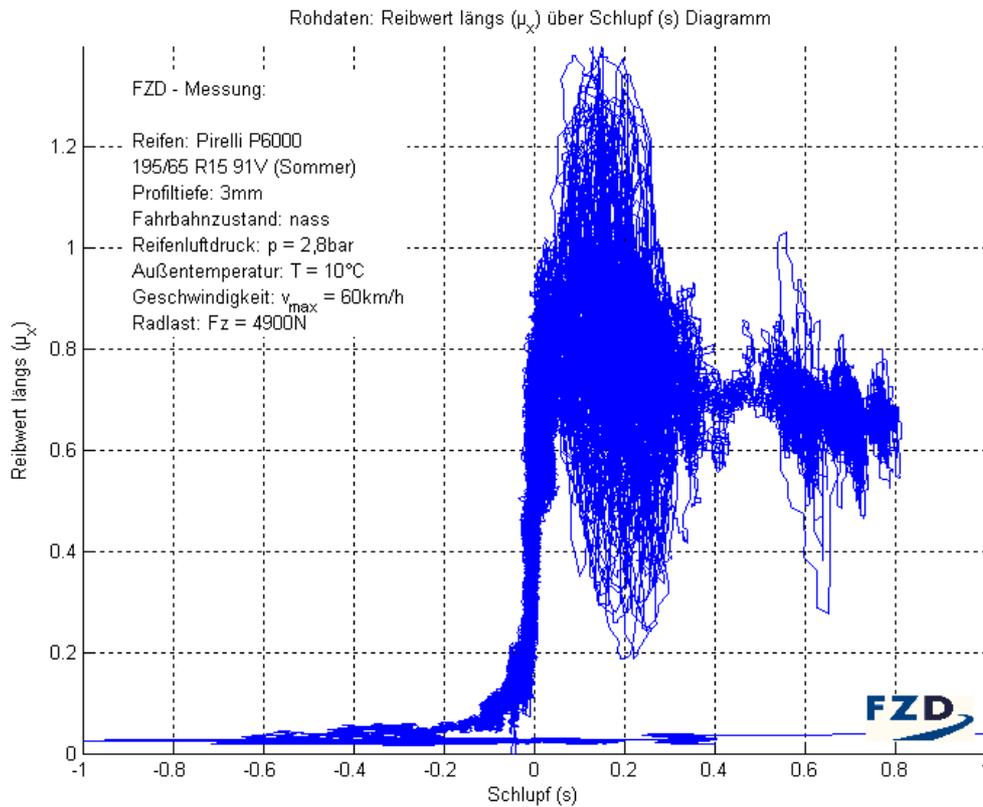
$$R_{\text{dyn,berechnet}} = 0,31055; \quad R_{\text{dyn,manuell}} = 0,30987; \quad R_{\text{dyn,gemessen}} = 0,31042$$

Die Differenz der verschieden ermittelten dynamischen Reifenhalbmesser liegt bei 0,02%.

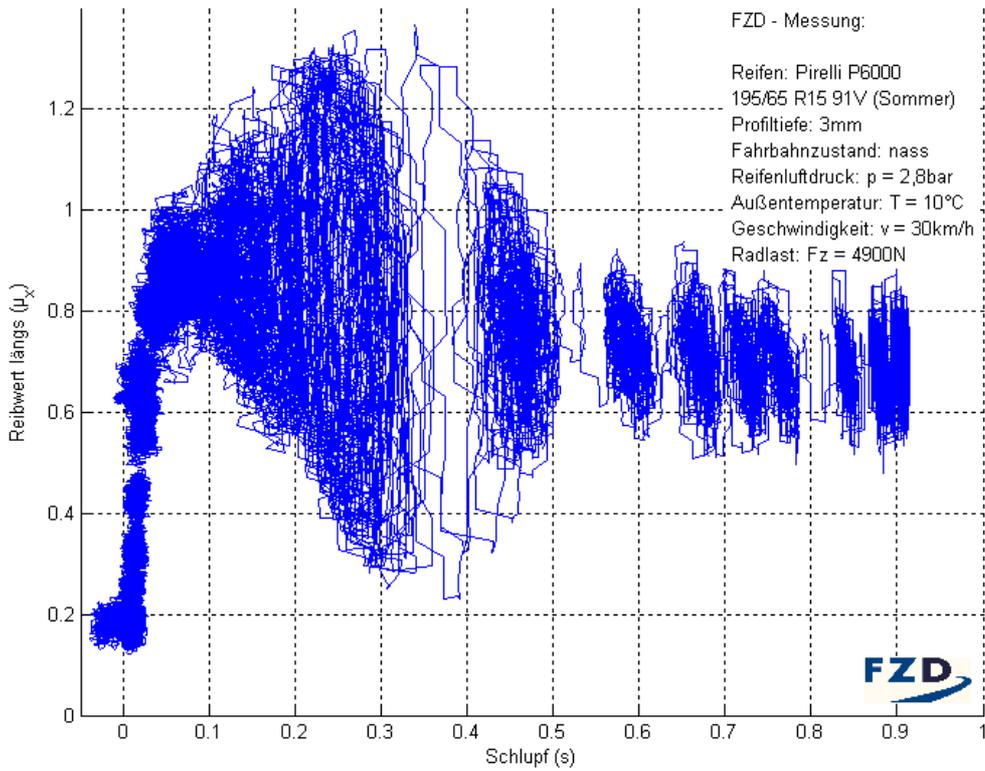
Damit ist die R_{dyn} -Berechnung verifiziert.

11. Test Messungsdurchführung und Messdatenaufzeichnung:

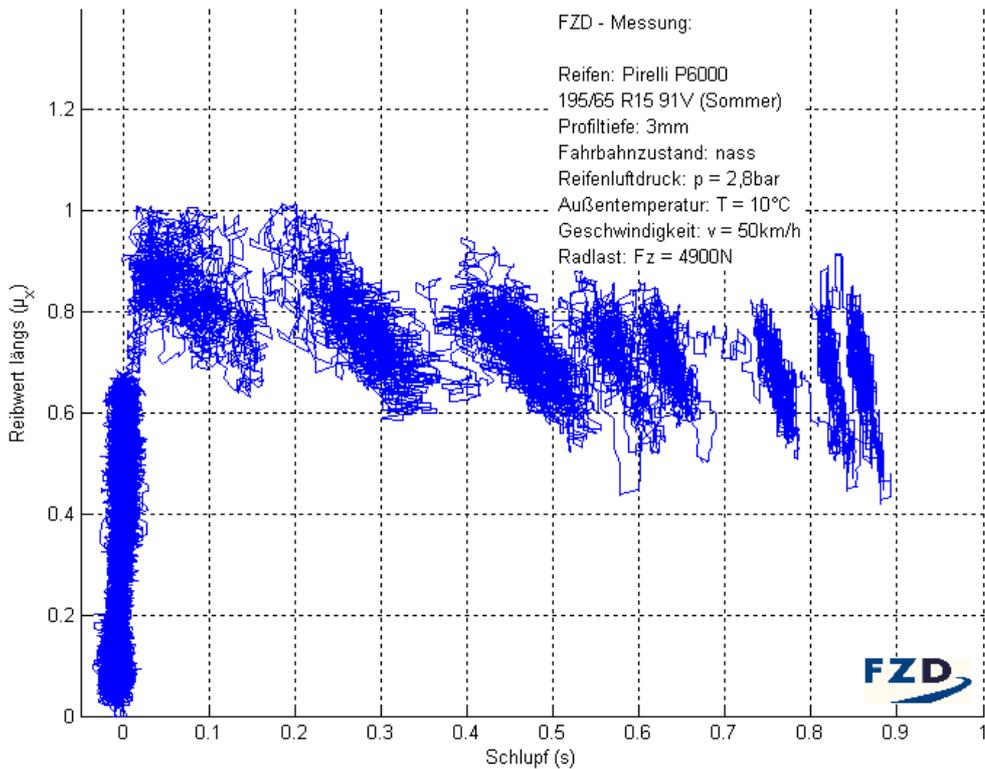
Der Test wurde entsprechend der Systemtestvorschrift durchgeführt. Da die Messfelge defekt war wurden diese Versuche ohne Messfelge gefahren.



Rohdaten: Reibwert längs (μ_x) über Schlupf (s) Diagramm



Rohdaten: Reibwert längs (μ_x) über Schlupf (s) Diagramm



12. Performancetest:

Der Performancetest wurde mit 4kHz begonnen und von Test zu Test um 1kHz gesteigert. Der erste Systemabsturz trat bei 8kHz auf. Während der Validierung die zunächst mit 6kHz begonnen wurden kam es hin und wieder zum Systemabsturz. Bei 5kHz kam es bisher zu keinem Systemabsturz. Daher wird empfohlen die maximale Abtastrate von 5kHz nicht zu überschreiten.

13. Kompatibilitätstest zu Matlab:

Die Messdateien wurden mit Matlab eingelesen und ausgewertet.

14. Zuverlässigkeitstest:

Während der Systemverifikation & Validierung kam es zu 3 Systemabstürzen, wenn die performancebedingten Systemabstürze nicht berücksichtigt werden.

Anhang H Validierung

H1. Versuchstabelle (Parametervariationsliste)

Nr.	Geschwindigkeit [km/h]	Schräglaufwinkel [°]
1	20 (1. Gang)	0
2	20 (1. Gang)	2
3	20 (1. Gang)	5
4	20 (1. Gang)	-2
5	20 (1. Gang)	-5
6	30 (2. Gang)	0
7	30 (2. Gang)	2
8	30 (2. Gang)	5
9	30 (2. Gang)	-2
10	30 (2. Gang)	-5
11	50 (5. Gang)	0
12	50 (5. Gang)	2
13	50 (5. Gang)	5
14	50 (5. Gang)	-2
15	50 (5. Gang)	-5

H2. Validierungsprotokoll

Aufbau des Messdateinamen: (Datei)Parametersatz(Nr).lvm

Datei	Nr	v _{Fzg} [km/h]	Schräg- lauf- winkel	Offset F _y [N]	Fre- quenz [Hz]	Profiltiefe		Bemerkung
						(außen) [mm]	(in- nen) [mm]	
0	1	20	0	425	7000	6	5,5	ok
0	2	20	0	425	7000	6	5,5	ok
0	3	20	2	425	7000	6	5,5	ok
0	4	20	2	425	7000	6	5	Abbruch aus der Messung gefallen
0	5	20	2	425	7000	6	5	ok
0	6	20	4	425	7000	6	5	ok
0	7	20	4	425	7000	5,5	4,5	ok
0	8	20	-2	425	7000	5,5	4,5	ok
0	9	20	-2	425	7000	5,5	4,5	ok
0	10	20	-4	425	7000	5,5	4,5	Softwareabsturz ⇒ gelöscht
1	0	20	-4	425	7000	5,5	4	ok
1	1	20	-4	425	7000	5,5	4	ok
1	2	30	0	425	7000	5,5	4	Abbruch Anzeige Host hing
1	3	30	0	425	7000	5	3,5	ok
2	0	30	2	80	7000	5	3,5	ok
2	1	30	4	80	7000	5	3,5	Abbruch aus der Messung gefallen
2	2	30	4	80	7000	5	3,5	ok
2	3	30	-2	80	7000	5	3	ok
2	4	30	-4	80	7000	4,5	3	Abbruch aus der Messung gefallen
2	5	30	-4	80	7000	4,5	3	ok
2	6	50	0	80	7000	4,5	2,5	Softwareabsturz ⇒ gelöscht
3	0	50	0	80	7000	4,5	2,5	ok

3	1	50	2	80	7000	4	2	Softwareabsturz \Rightarrow gelöscht
4	0	50	2	80	7000	4	2	Softwareabsturz \Rightarrow gelöscht
5	0	50	2	80	6000	4	2	ok (vom Gas gegangen, Verkehr)
5	1	50	4	80	6000	3,5	1,5	ok
5	2	50	-2	80	6000	3,5	1,5	ok (vom Gas gegangen, Verkehr)
5	3	50	-4	80	6000	3	1	Abbruch RLW Motor MR hing
6	0	50	-4	130	6000	3	1	ok

H3. Methodik der Auswertung

H3.1. Brainstorming

Mess/Kontroll/Kenngrößen zur Ermittlung eines quasistationären Schlupfzustandes:

	Drehzahl	
v_{Rad}	v_{Fzg}	Radlenkwinkel
Hydraulikdruck	Hydrauliktemperatur	Schräglaufwinkel Correvit
F_x	F_y	F_z
Schlupf	$\mu_{\text{längs}}$	Bremsmoment- stufe

H3.2. Bewertungsmatrix

	1	2	3	4	5
Anzahl der verwertbaren Messpunkte	Sehr wenig Punkte verwertbar	Wenig Punkte verwertbar	Viele Punkte verwertbar	Sehr viele Punkte verwertbar	Alle Punkte verwertbar
Aufwand in der Umsetzung und Messung	Sehr hoher Aufwand	Hoher Aufwand	Mittlerer Aufwand	Sehr geringer Aufwand	Kein Aufwand
Klare Zuordnung der Zustände zu den Stufen	Keine Zuordnung möglich	Wenig Punkte zuordenbar	Die meisten Punkte zuordenbar	Fast immer alle Punkte zuordenbar	Immer alle Punkte zuordenbar
Anzahl der verwertbaren Messpunkte	<10%	10% - 50%	50% - 80%	80% - 100%	100%
Aufwand in der Umsetzung	Starker Softwareeingriff	Geringer Softwareeingriff	Komplizierte Auswertung	Einfache Auswertung	Sehr einfache Auswertung
Erkennung von instationären Bereichen	Sehr Geringe Erkennungsrate	Geringe Erkennungsrate	Mittlere Erkennungsrate	Hohe Erkennungsrate	Sehr hohe Erkennungsrate
Auswirkungen von Fehlerkennungen (Anzahl mal Gewichtung)	Sehr große Auswirkung	Große Auswirkung	Mittlere Auswirkung	Sehr geringe Auswirkung	Keine Auswirkung

H3.3. Bewertung der Auswertemöglichkeiten

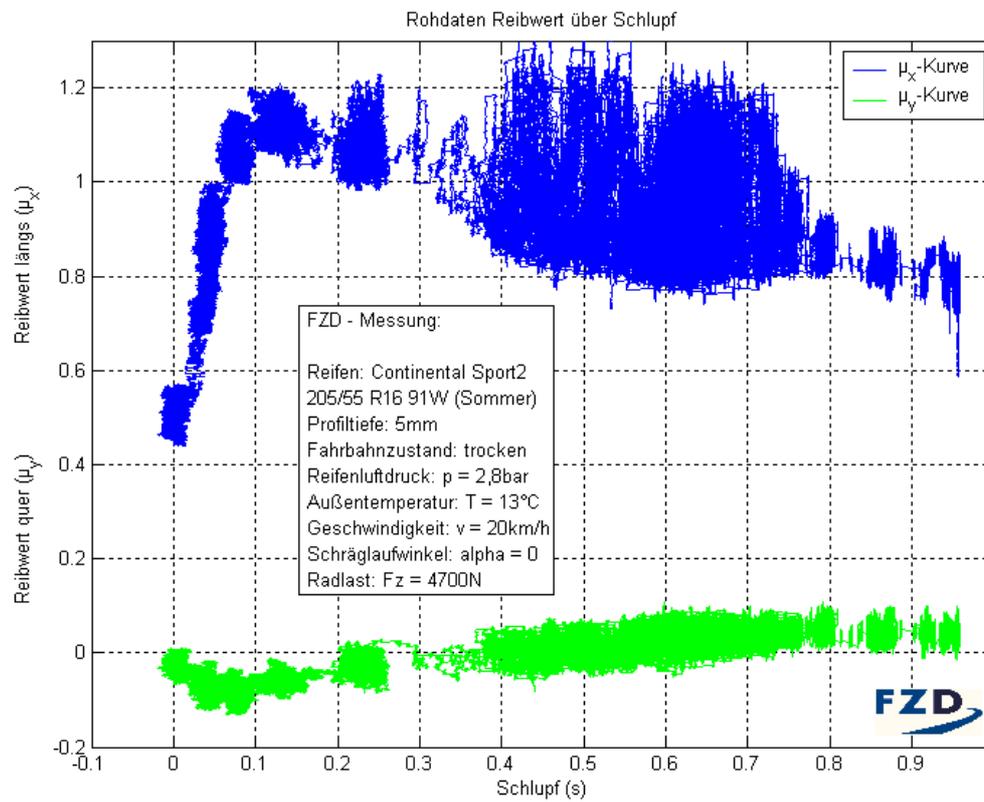
	Anzahl der verwertbaren Messpunkte	Aufwand in der Umsetzung und Messung	Klare Zuordnung der Zustände zu den Stufen	Gesamt
Gewichtung	0,4	0,4	0,2	1
Wenige Stufen	1	4	4	2,8
Beliebige Stufen und Kriterium	3	4	3	3,4
Viele Stufen	5	2	1	3
Maximale Punktzahl	5	5	5	5

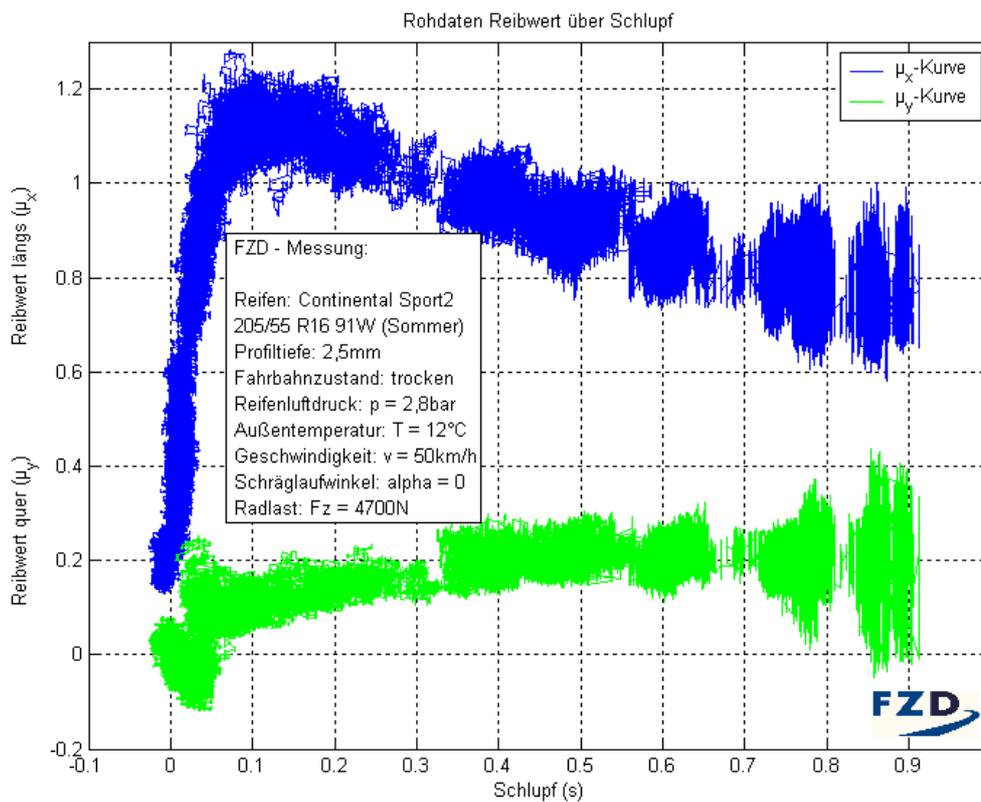
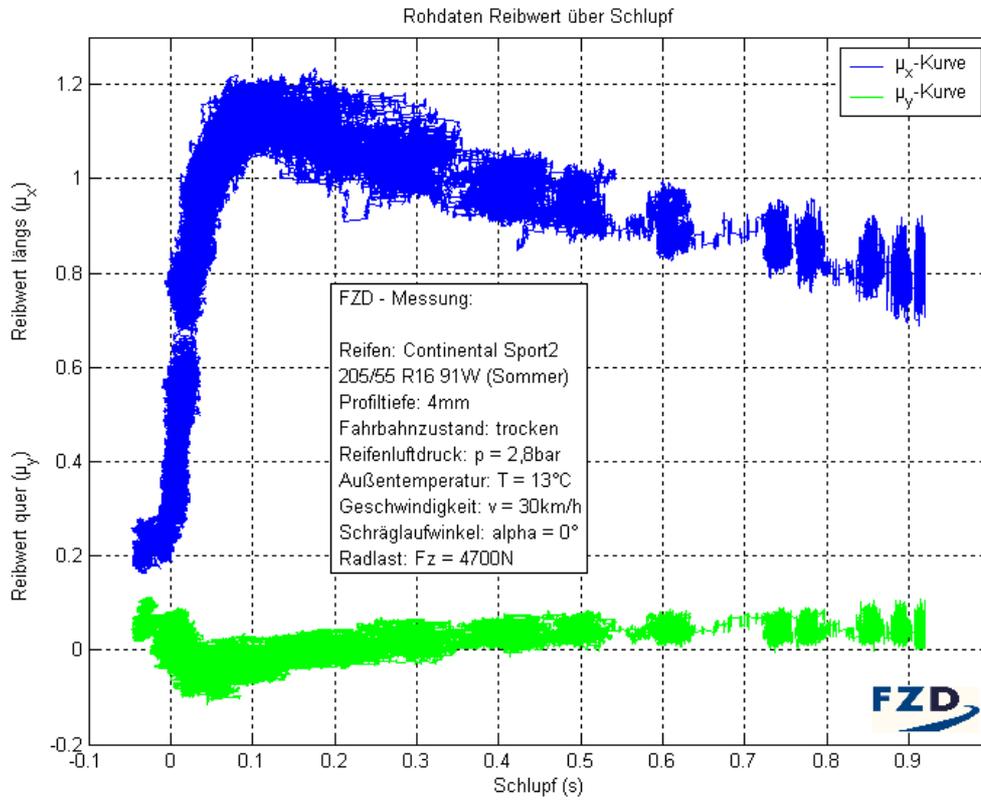
H3.4. Bewertung der Kriterien zur Detektion von stationären Zuständen

	Anzahl der verwertbaren Messpunkte	Aufwand in der Umsetzung	Erkennung von instationären Bereichen	Auswirkungen von Fehlerkennungen (Anzahl x Gewichtung)	Gesamt
Gewichtung	0,3	0,2	0,3	0,2	1
Bremsmomentstufe + Δ Zeit	2	4	4	3	2,9
Schlupf	4	4	3	3	3,5
$\mu_{\text{längs}}$	4	4	2	2	3
F_x	4	4	1	1	2,5
v_{Rad}	4	4	2	2	3,0
Hydraulikdruck	3	1	3 ?	3 ?	2,6
Kombination von Schlupf und $\mu_{\text{längs}}$	4	3	5	2	3,7
Maximale Punktzahl	5	5	5	5	5

H4. Auswertung

H4.1. Rohdaten: μ -Schlupf-Kurven



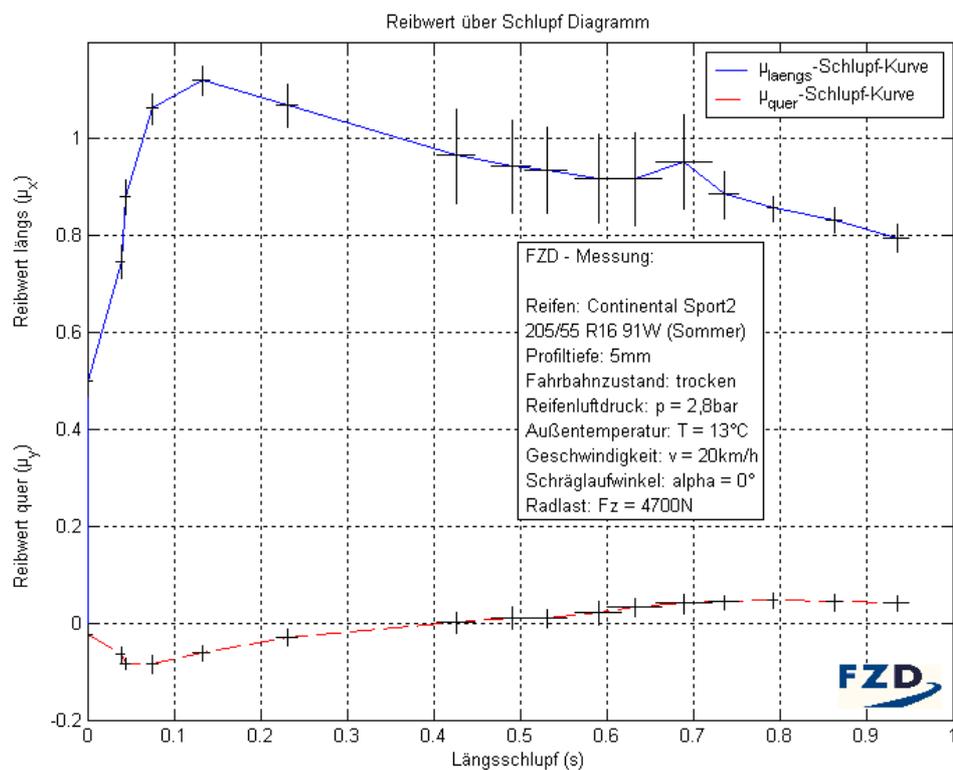


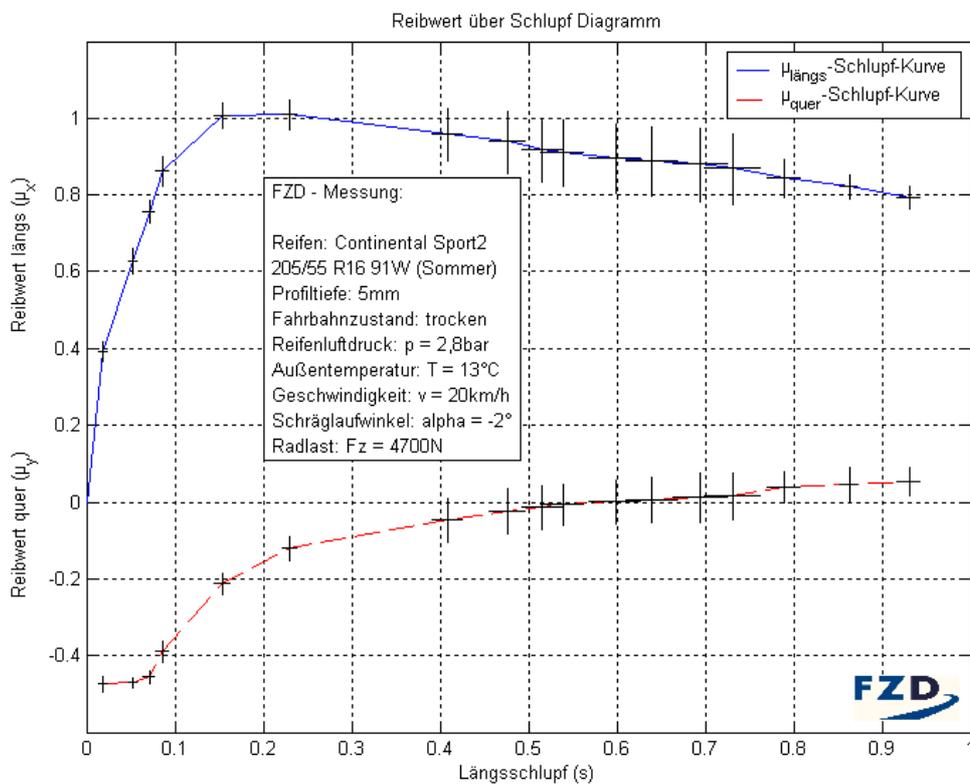
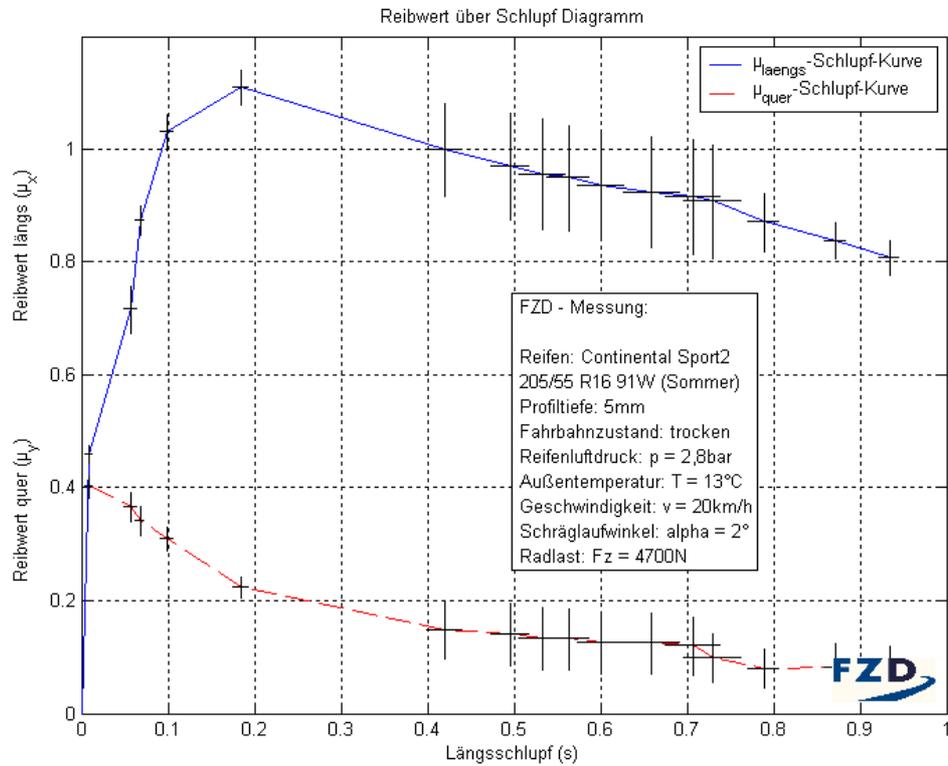
H4.2. Einzelne μ -Schlupf-Kurven

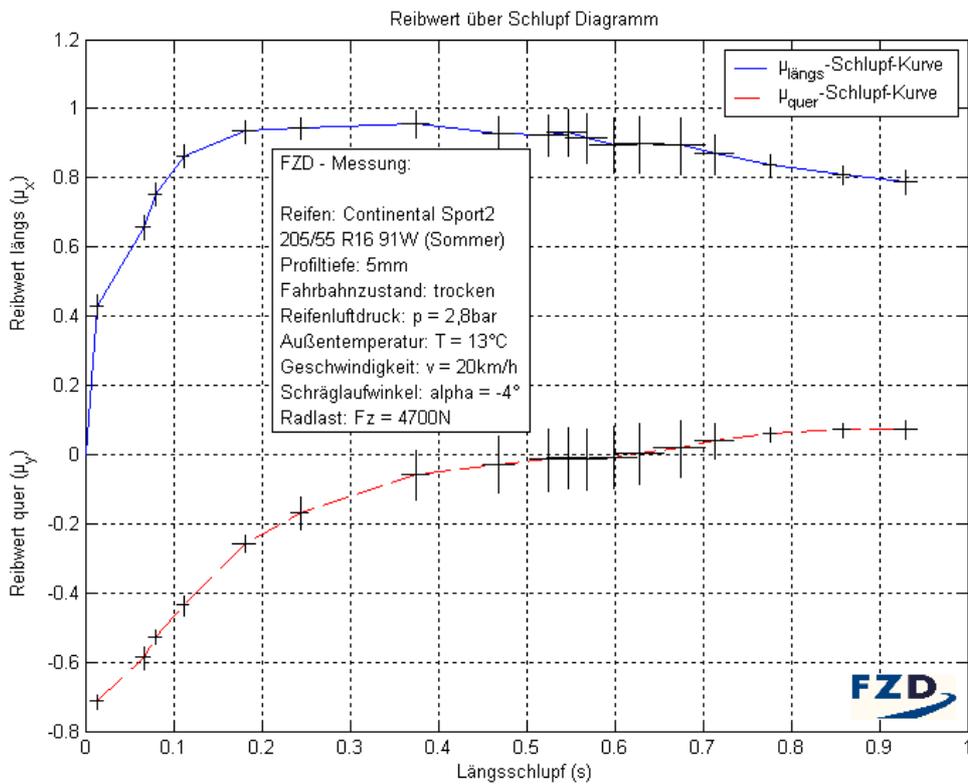
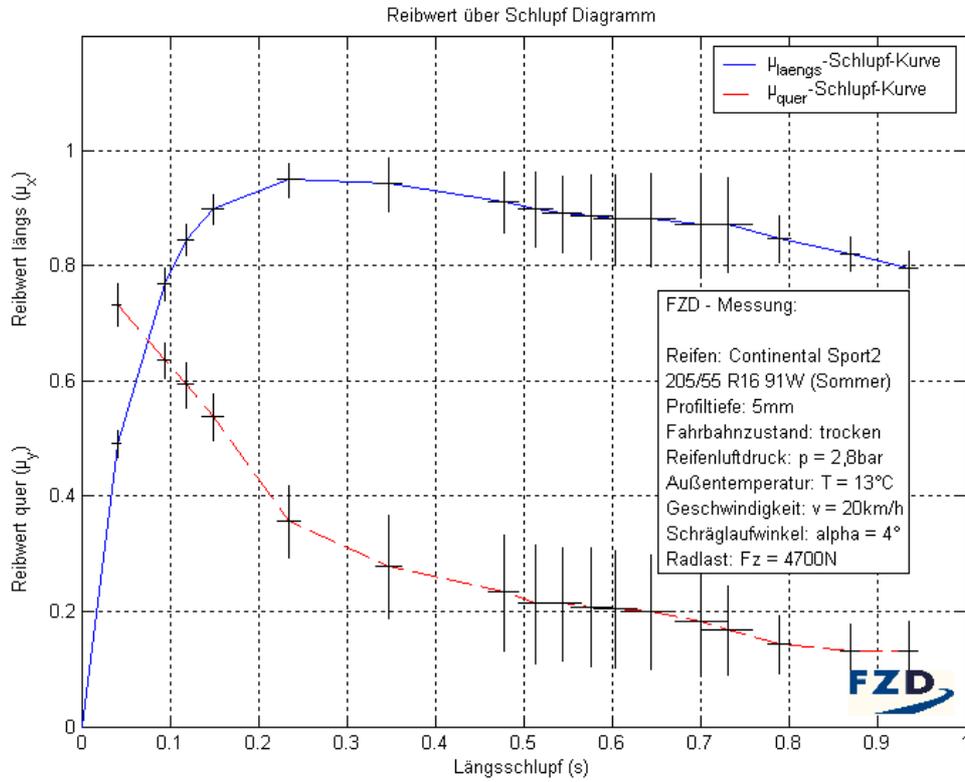
H4.2.1. Einzelne Plots bei $v = 20 \text{ km/h}$

Reihenfolge:

- $v = 20 \text{ km/h}, \alpha = 0^\circ$
- $v = 20 \text{ km/h}, \alpha = 2^\circ$
- $v = 20 \text{ km/h}, \alpha = -2^\circ$
- $v = 20 \text{ km/h}, \alpha = 4^\circ$
- $v = 20 \text{ km/h}, \alpha = -4^\circ$



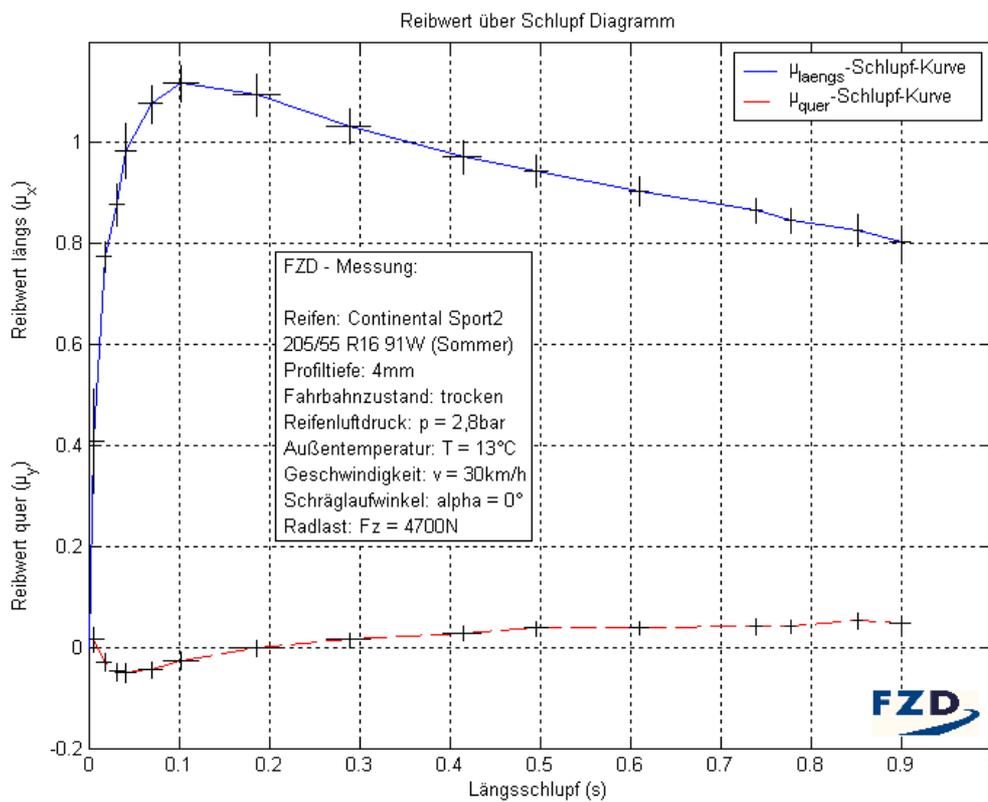


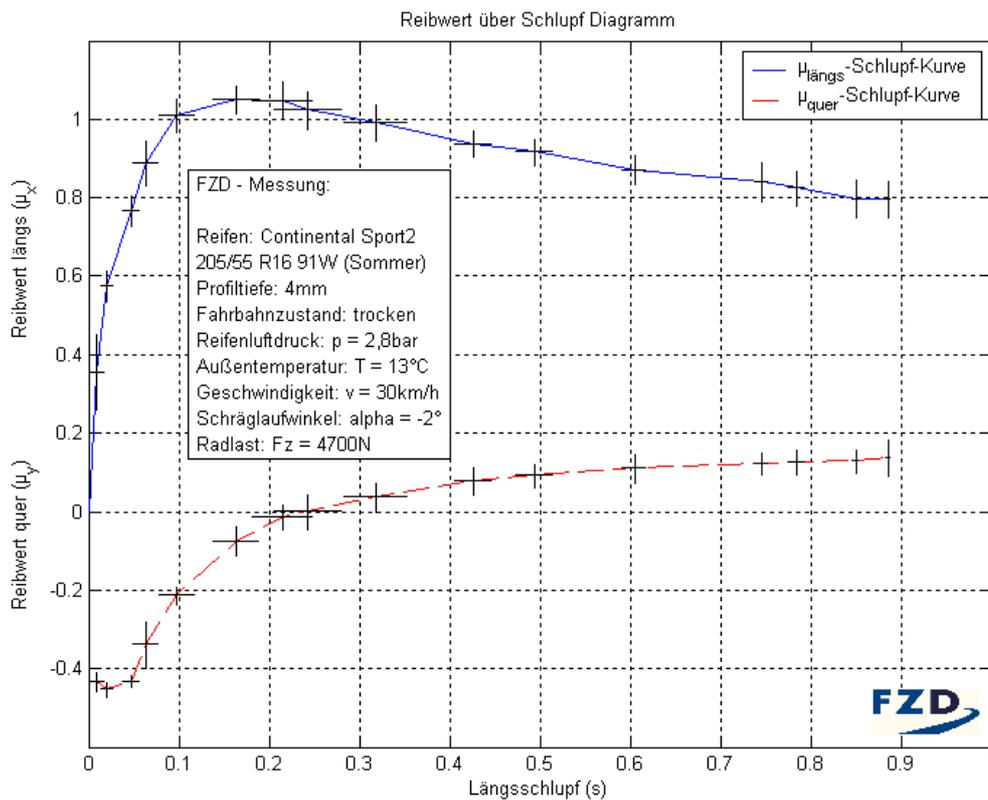
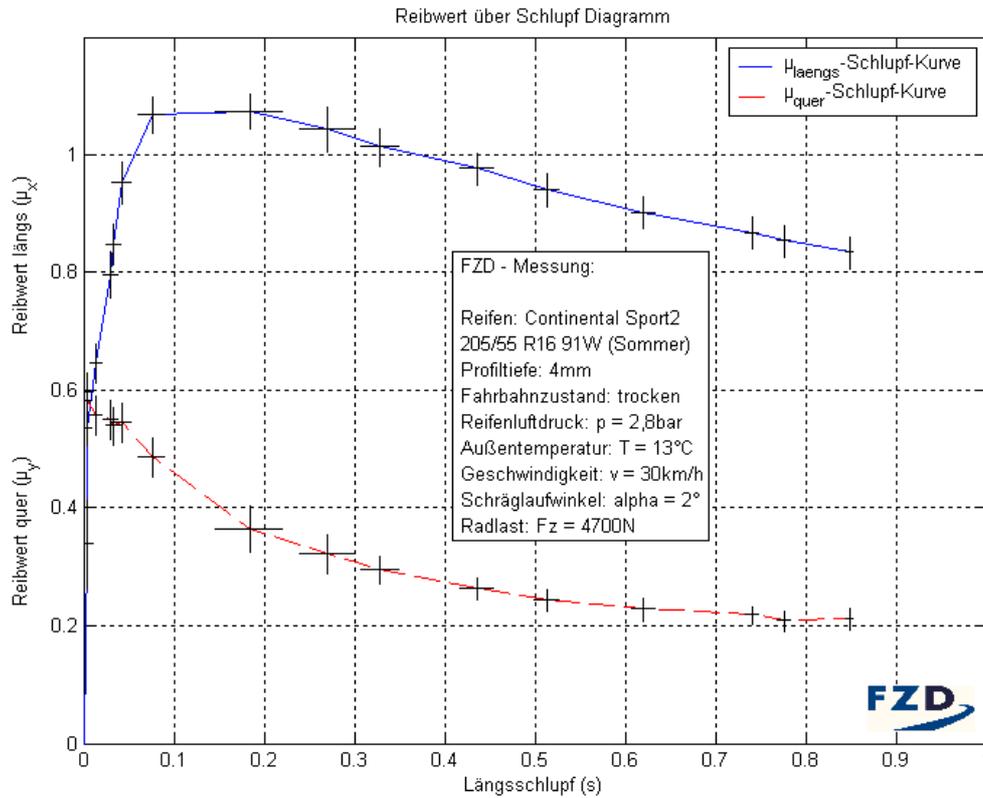


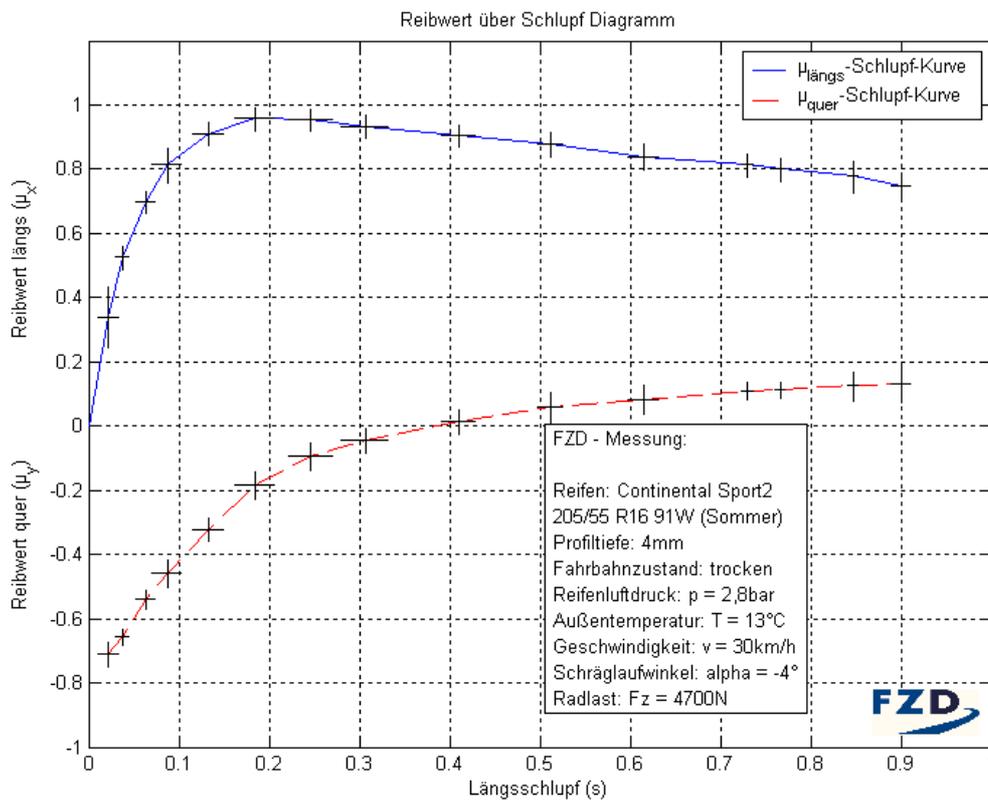
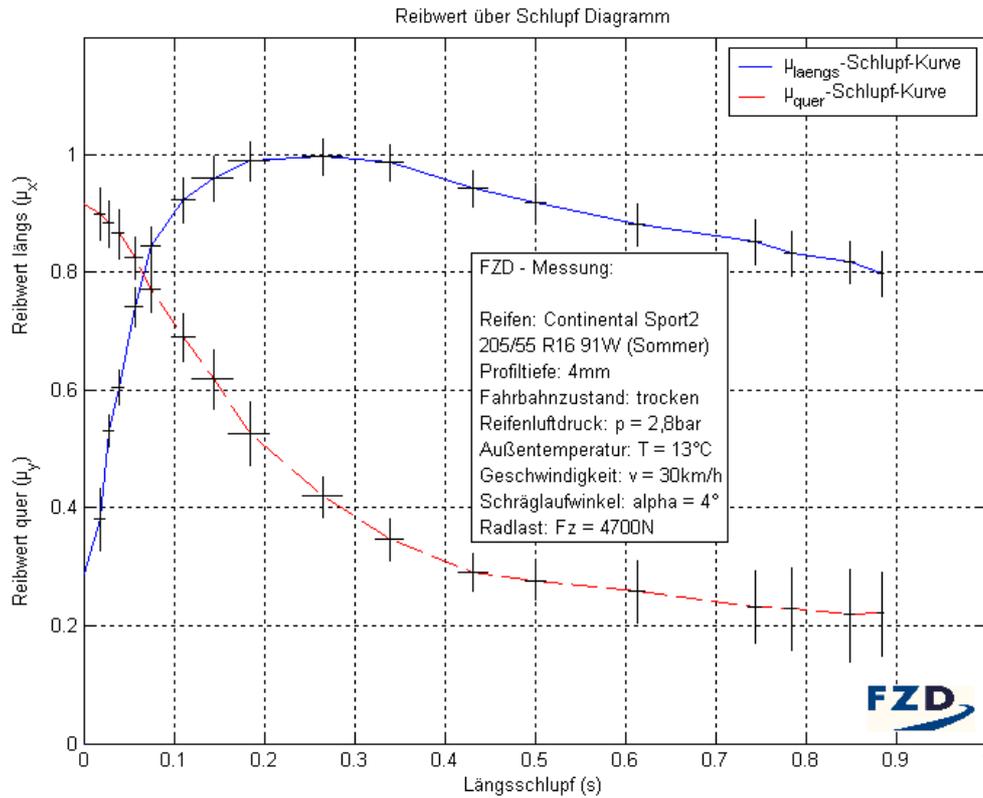
H4.2.2. Einzelne Plots bei $v = 30 \text{ km/h}$

Reihenfolge:

- $v = 30 \text{ km/h}, \alpha = 0^\circ$
- $v = 30 \text{ km/h}, \alpha = 2^\circ$
- $v = 30 \text{ km/h}, \alpha = -2^\circ$
- $v = 30 \text{ km/h}, \alpha = 4^\circ$
- $v = 30 \text{ km/h}, \alpha = -4^\circ$



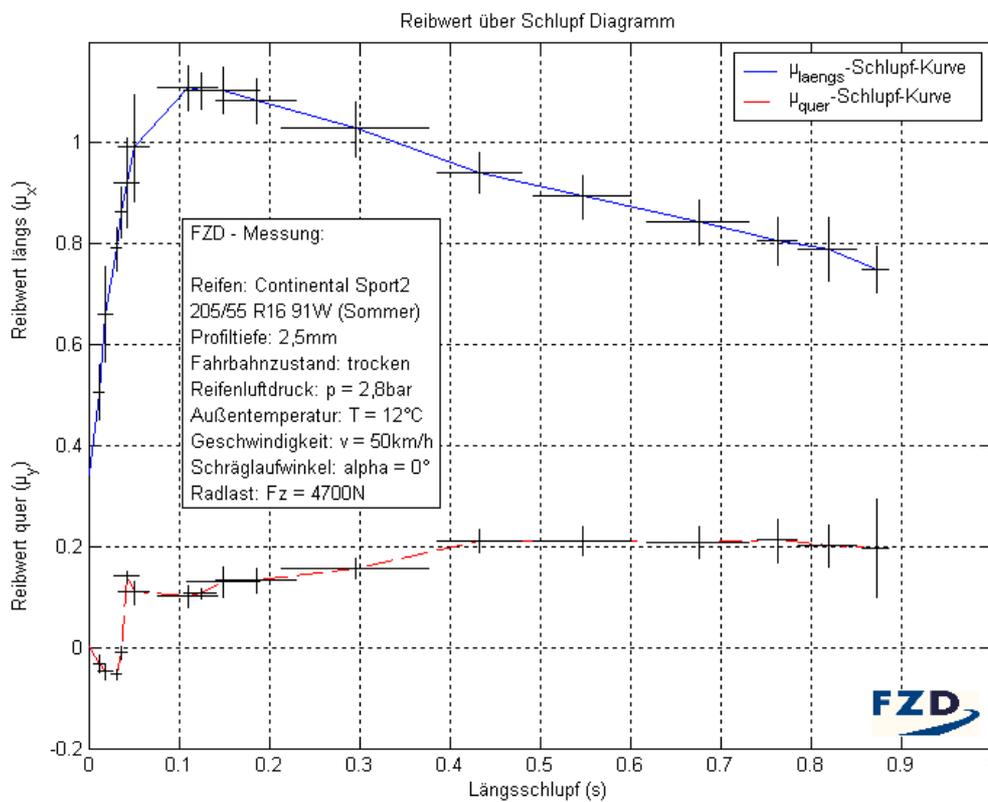


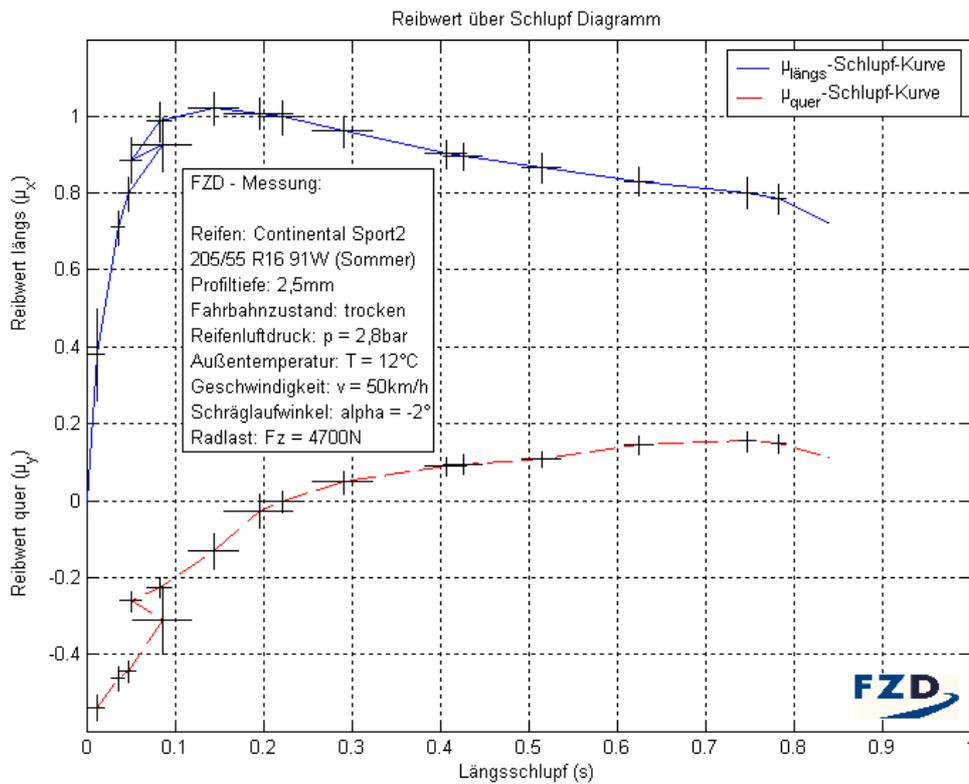
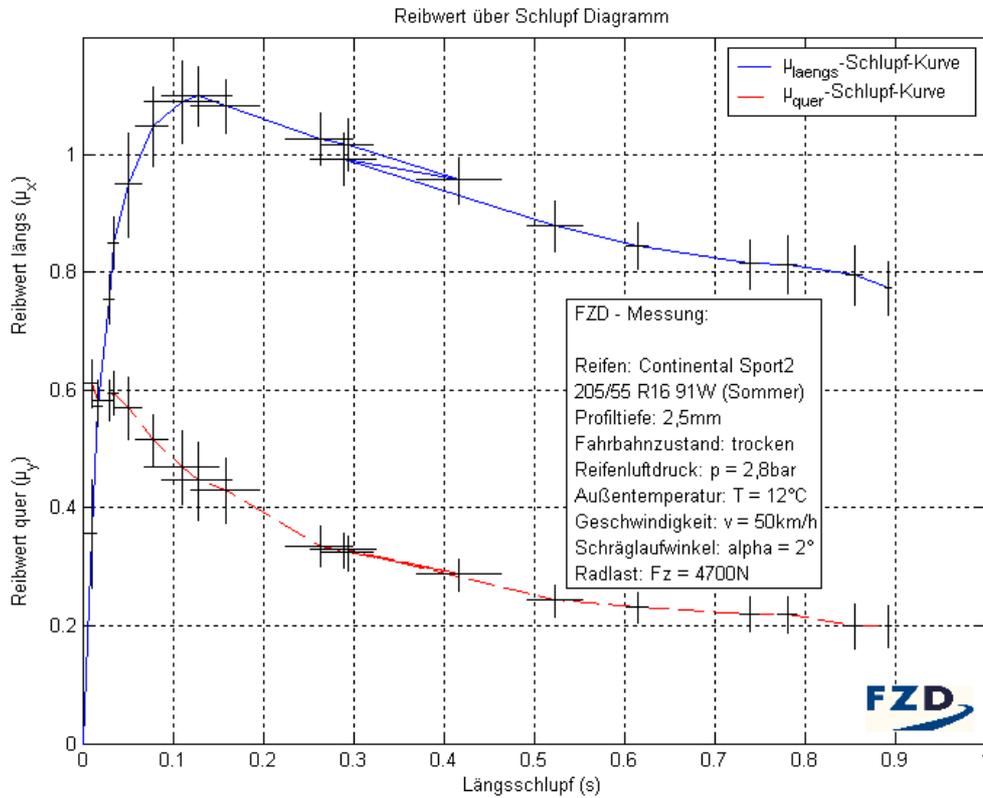


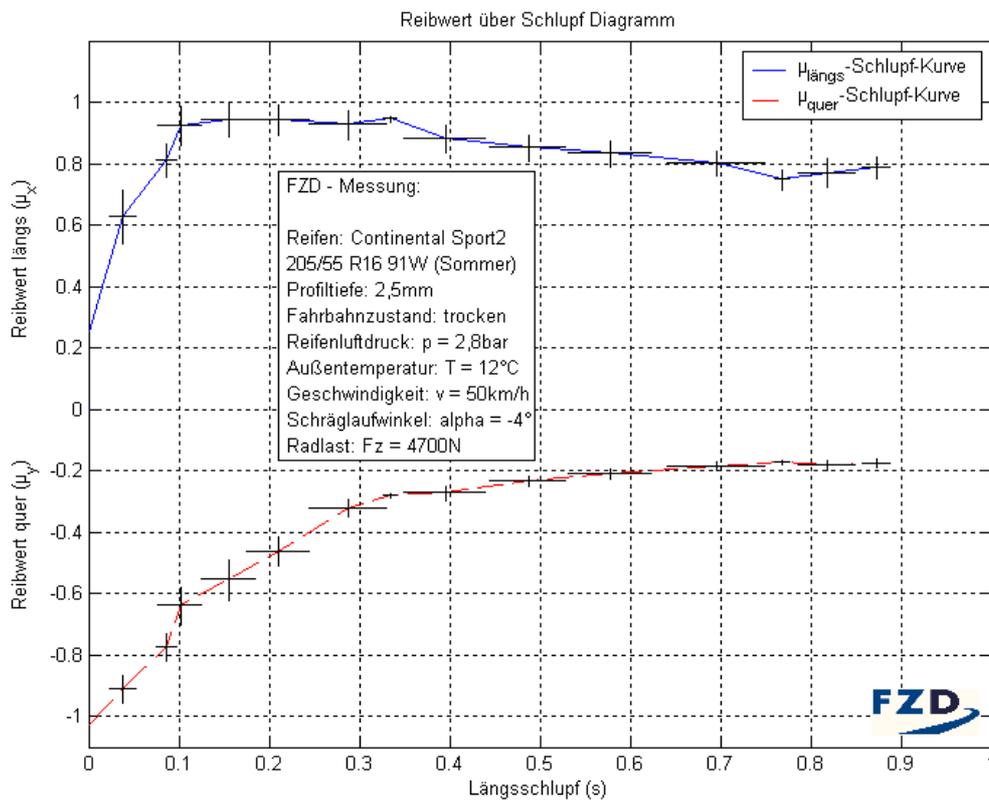
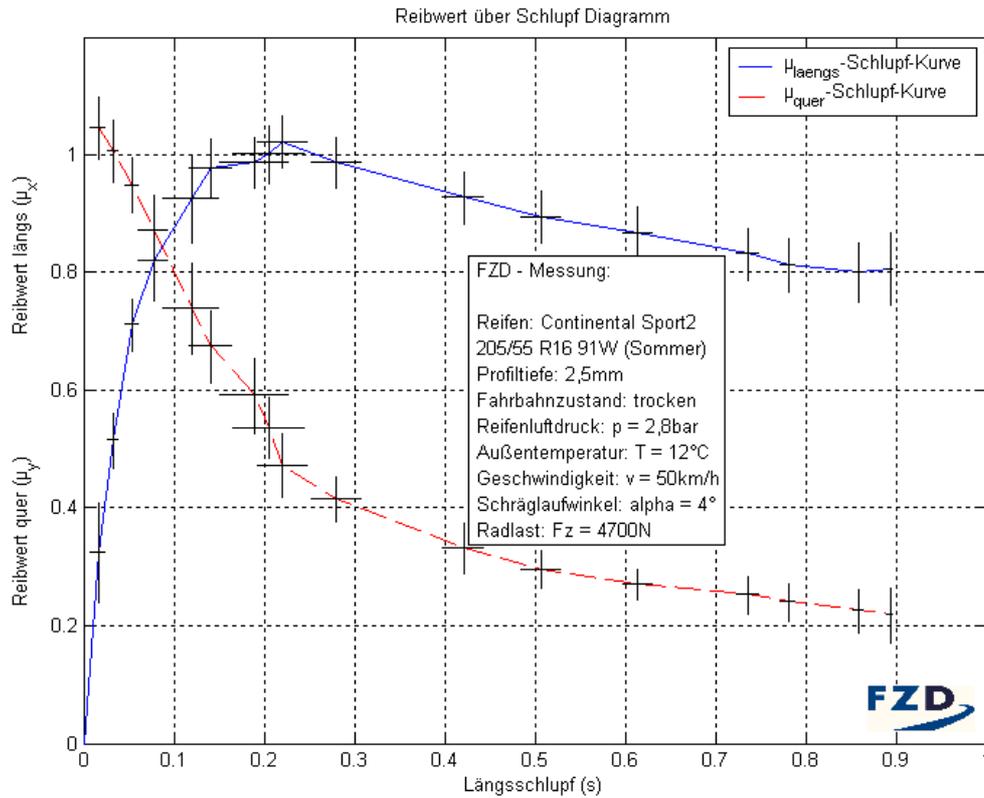
H4.2.3. Einzelne Plots bei $v = 50 \text{ km/h}$

Reihenfolge:

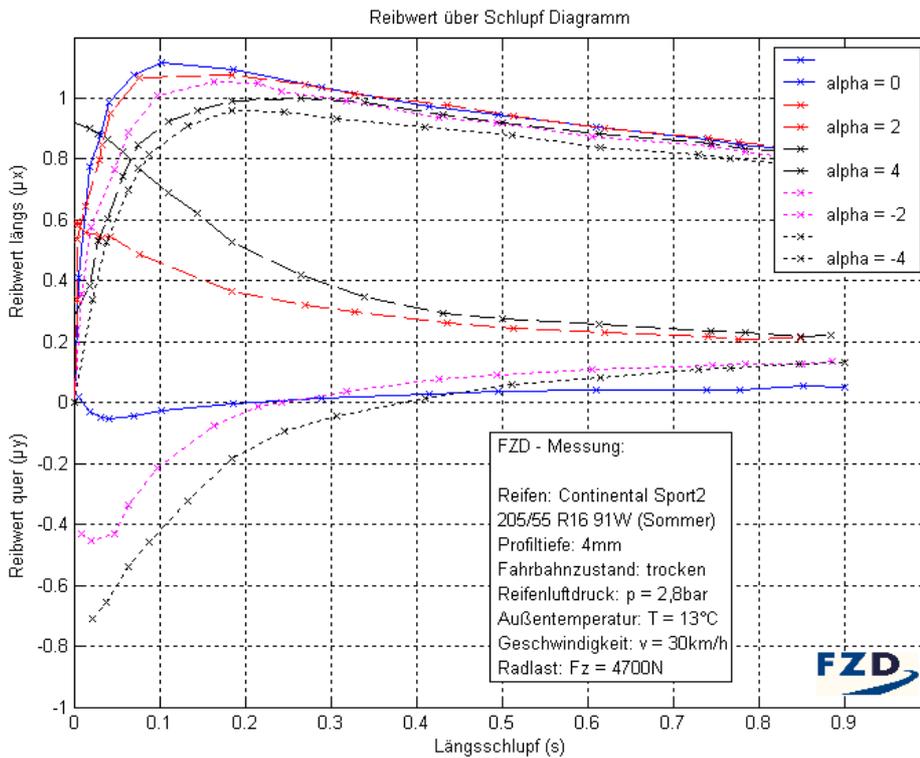
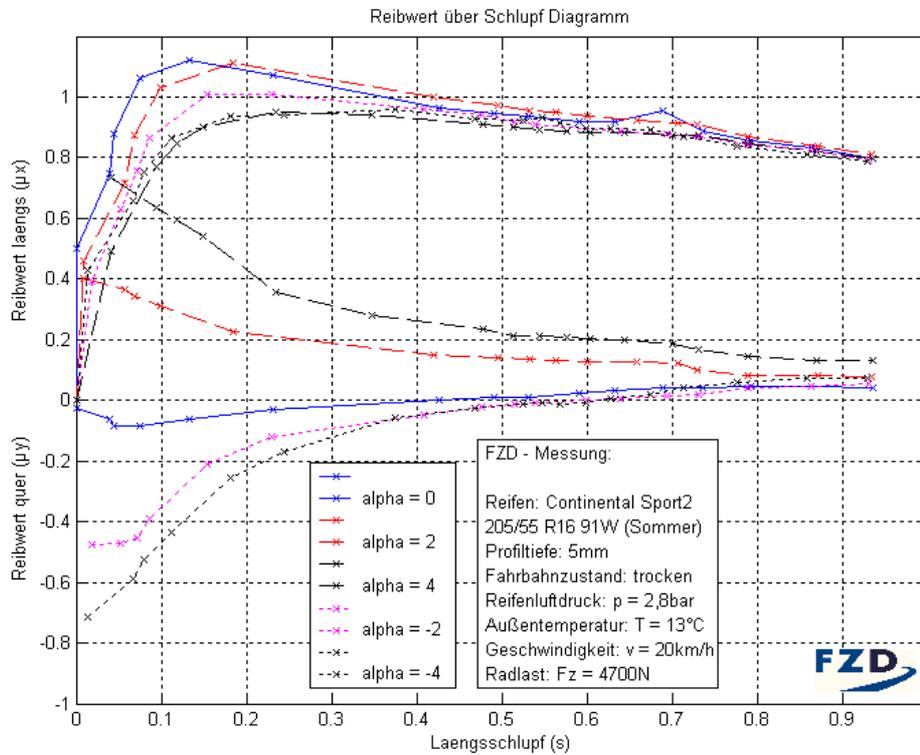
- $v = 50 \text{ km/h}$, $\alpha = 0^\circ$
- $v = 50 \text{ km/h}$, $\alpha = 2^\circ$
- $v = 50 \text{ km/h}$, $\alpha = -2^\circ$
- $v = 50 \text{ km/h}$, $\alpha = 4^\circ$
- $v = 50 \text{ km/h}$, $\alpha = -4^\circ$

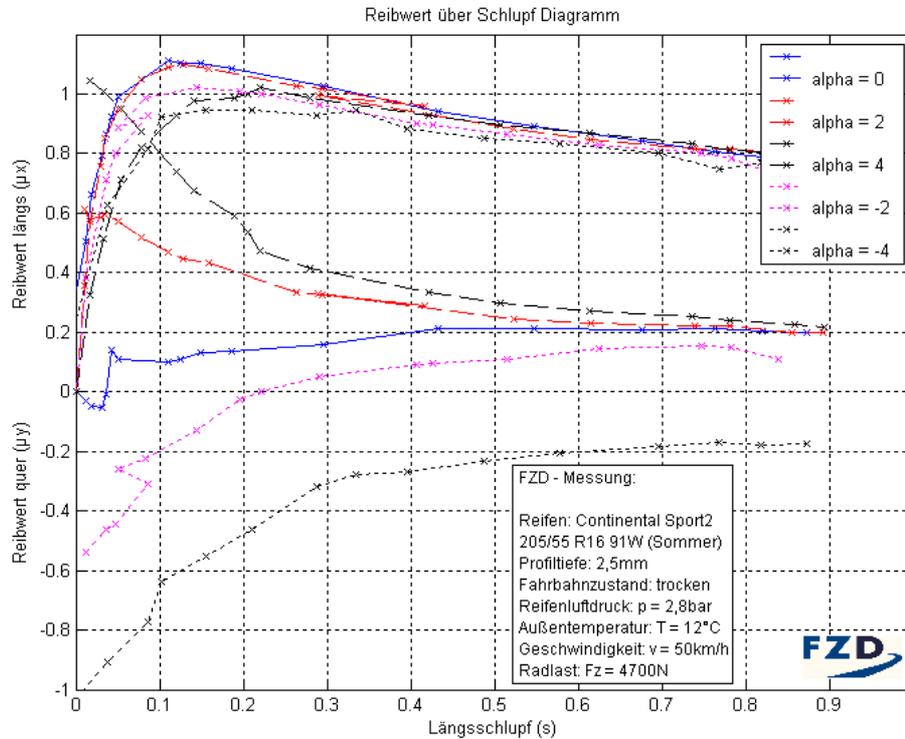




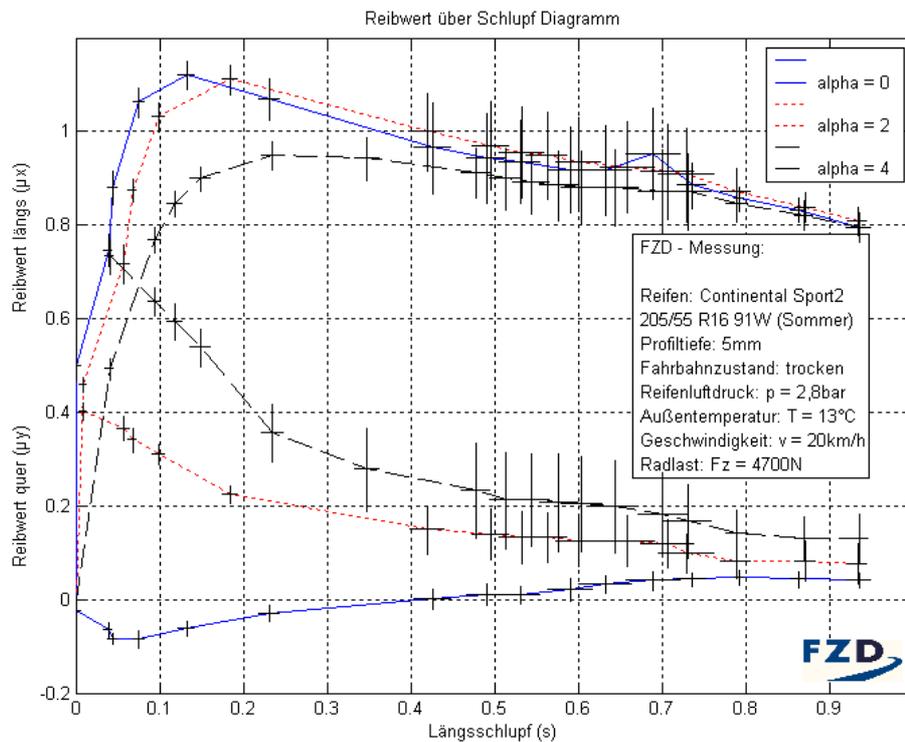


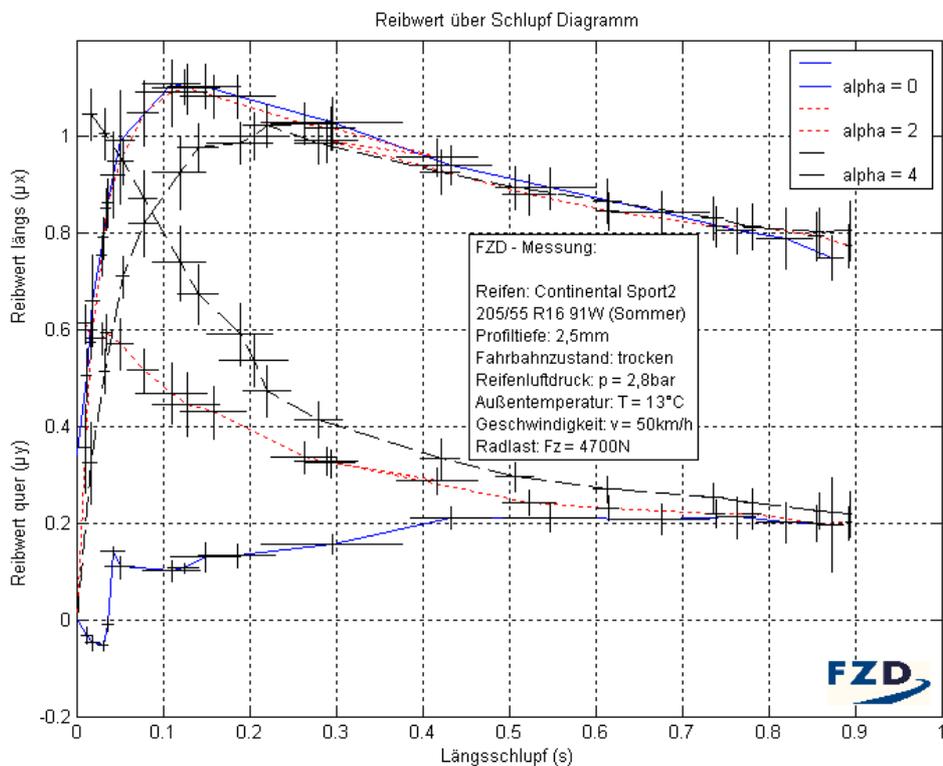
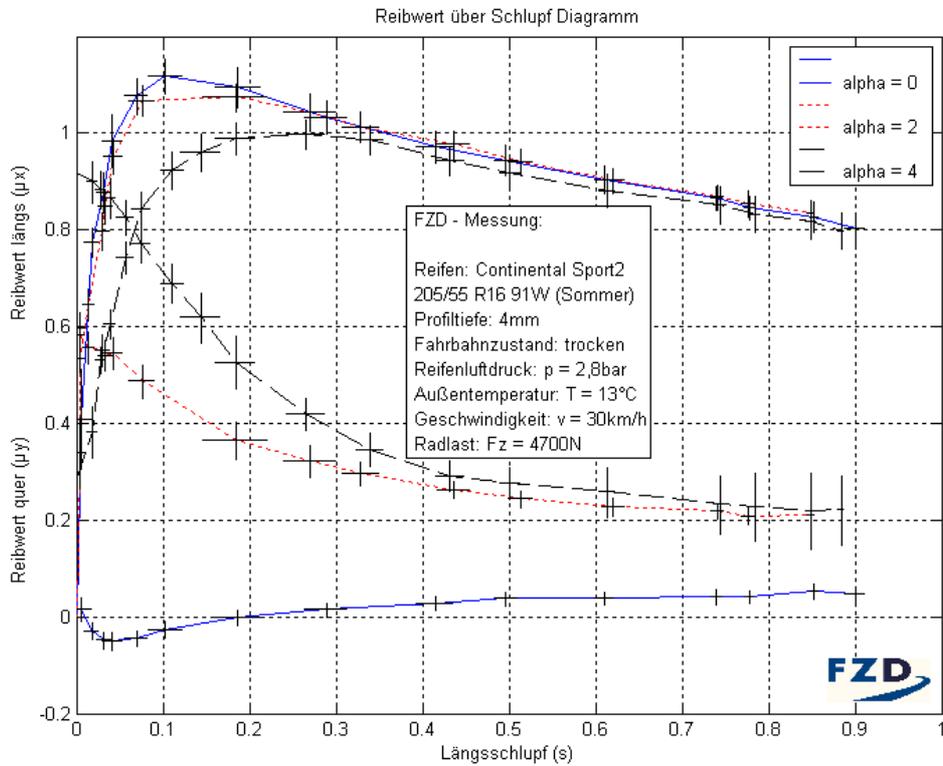
H5. Alle μ -Schlupf-Kurven einer Geschwindigkeit



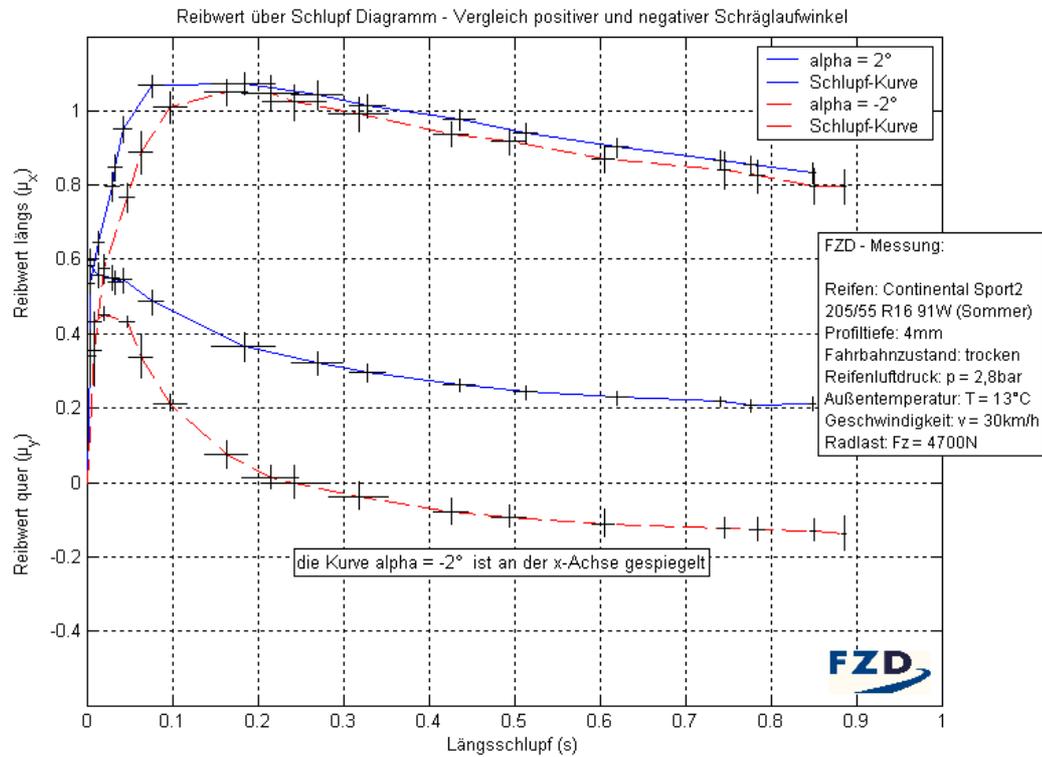


H6. Positive μ -Schlupf-Kurven einer Geschwindigkeit





H7. Vergleich der μ -Schlupf-Kurven von $\pm 2^\circ$



Anhang I Bedienungsanleitung für PAMELA

Nach der Installation von PAMELA (= PETRAs automatische Messdatenerfassungslogik und Auswertesoftware), wird die Datei PAMELA-Host.exe gestartet. Es öffnet sich ein neues Fenster, in dem die grafische Benutzeroberfläche von PAMELA zu sehen ist. Die Konfigurationsbenutzeroberfläche, die automatisch nach dem Start geladen wird ist in der nächsten Abbildung zu sehen. Die PAMELA-Benutzeroberfläche ist zweigeteilt. Während der untere Teil, die Kontroll-GUI, fest ist kann durch Klicken auf die Reiter des Tabelements, welches den oberen Teil einnimmt zwischen den einzelnen Sub-GUIs gewechselt werden.

Kontrollanzeigen und -elemente:

Das untere Panel der GUI, welche in der nächsten Abbildung unten zu sehen, dient zur Überwachung und Kontrolle von PETRA. Die Anzeigen in diesem Panel werden erst aktiv, wenn die Datenübertragung läuft.

Auf der linken Seite sind die Kontroll-LEDs der Endschalter zu sehen. Leuchtet einer dieser Endschalterleuchtelemente auf, ist der entsprechende Endschalter aktiv. Dies darf im Normalbetrieb nicht auftreten, da sonst ein Fehler vorliegt. Im Handbetrieb können die Motoren bis zur Endschalteraktivierung verfahren werden.

In der Mitte dieses Panels werden die Kontrollgrößen angezeigt. Überschreitet eine der ersten 4 Größen die erste Grenze, wechselt das jeweilige Ausgabeelement die Farbe nach Rot. Wird die zweite Grenze überschritten beginnt das Element rot zu blinken. Fällt die Bordspannung unter die 3. Grenze bzw. steigt die CRIO-Temperatur über die 3. Grenze, fährt die Software das System automatisch herunter.

Die LED „System messbereit“ leuchtet auf sobald die Verbindung mit dem CRIO hergestellt wurde.

Der Button „Zerokalibrierung“ aktiviert die Nullkalibrierung der Messfelge. Ist diese abgeschlossen geht der Button automatisch in seine Grundstellung zurück.

Der Button „Schräglaufwinkel aktiv“ wird vom Benutzer eingeschalten, wenn der Radlenkwinkel am Messrad und damit der Schräglauf verändert werden soll, nachdem die Verbindung mit dem CRIO hergestellt wurde.

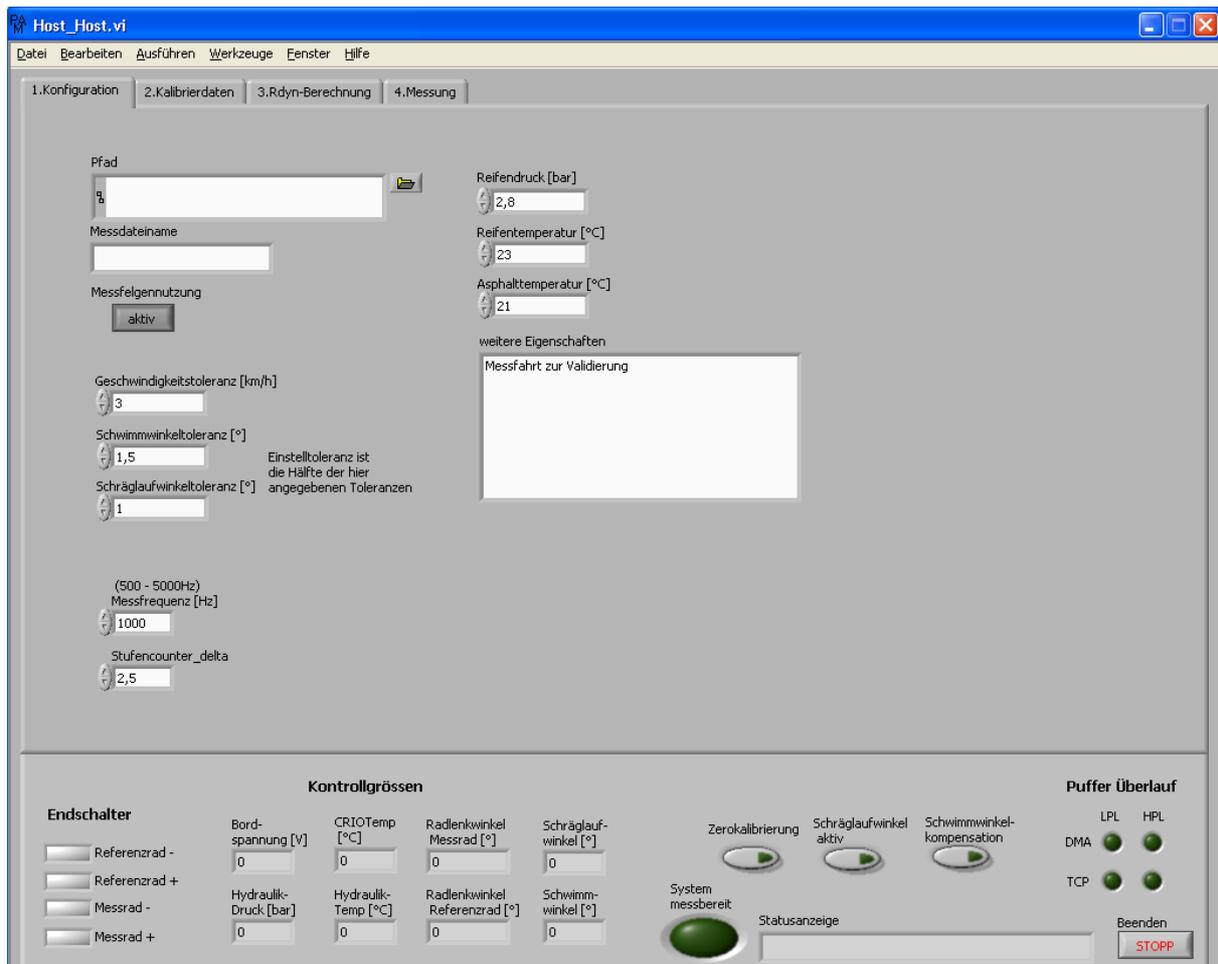
Der Button „Schwimmwinkelkompensation aktiv“ wird vom Benutzer eingeschalten, wenn der Radlenkwinkel am Referenzrad verstellt und damit der Schwimmwinkel automatisch ausgeregelt werden soll.

Durch diese beiden Schalter lässt sich die Radlenkwinkelverstellung jederzeit stoppen, sofern Verbindung zum CRIO besteht.

Die Statusanzeige unter diesen Kontrollbuttons zeigt den aktuellen Status in Form einer Statusmeldung an.

Die 4 LEDs auf der rechten Seite melden sobald einer der Puffer übergelaufen ist.

Der Button „STOPP“ rechts unten ermöglicht das „saubere“ Beenden der Software.

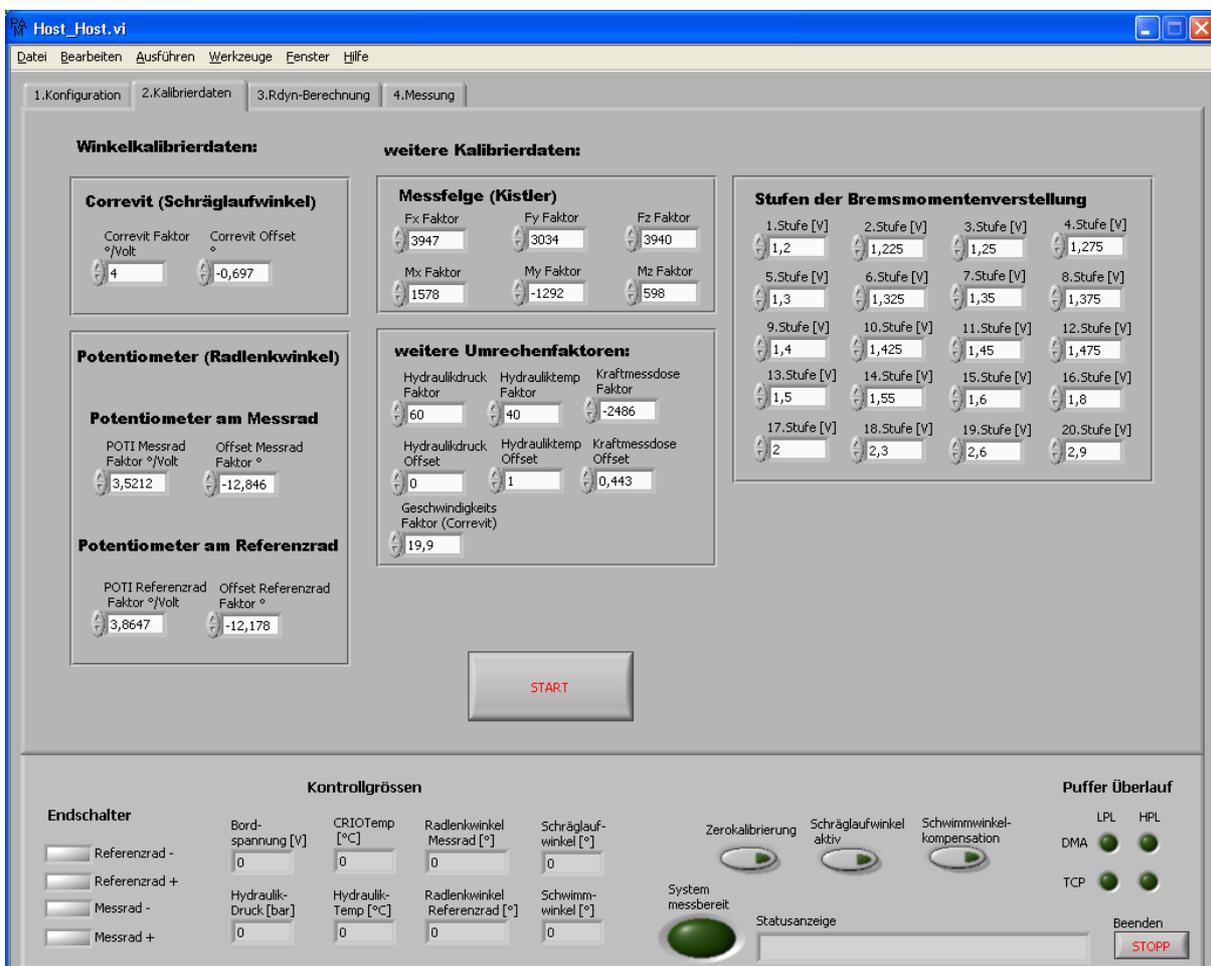


Konfiguration:

Die in der vorherigen Abbildung zu sehende Grafik zeigt die Bedieneroberfläche der Konfiguration. Als Erstes wählt der Benutzer den Pfad, an dem die Konfigurationsdatei und Messdatei/en gespeichert werden, indem er rechts von dem Pfadeingabelement auf den abgebildeten Ordner klickt und dann im erscheinenden Pop-Up in den zu verwendenden Ordner navigiert. Alternativ kann der Pfad auch direkt in das Eingabelement eingegeben werden. Im Eingabefeld „Messdateiname“ gibt der Benutzer als nächstes den zu verwendenden Messdateinamen an. Werden mehrere Messungen hintereinander mit der gleichen Konfigurations- und Kalibriereinstellung gefahren, werden die Messdateien automatisch durchnummeriert. Den Button Messfelgennutzung lässt der Benutzer auf „aktiv“, wenn die Messfelge montiert ist oder stellt ihn auf „inaktiv“, wenn die Messfelge nicht montiert ist. In den nächsten drei Eingabefeldern werden die folgenden Toleranzen eingegeben: Geschwindigkeitstoleranz,

Schwimmwinkeltoleranz, Schräglaufwinkeltoleranz. Die voreingestellten Werte sind Werte die sich während der Verifikation und Validierung bewährt haben. Im nächsten Eingabeelement kann der Benutzer die Messfrequenz verändern, die Standardmäßig auf 1000Hz eingestellt ist. Im untersten Eingabeelement auf der linken Seite gibt der Benutzer das Stufencounter_delta ein, welches die Zeit angibt, wie lange das aktuelle Bremsmoment gehalten wird bevor das nächst höhere eingestellt wird. Hier hat sich als geeigneter Wert 2,5 Sekunden ergeben, daher ist dieser Wert als Standard vorgegeben. In den Eingabeelementen auf der rechten Seite können Eigenschaften wie Reifendruck, Reifentemperatur, Asphalttemperatur und weitere Eigenschaften eingegeben werden.

Kalibrierdaten:

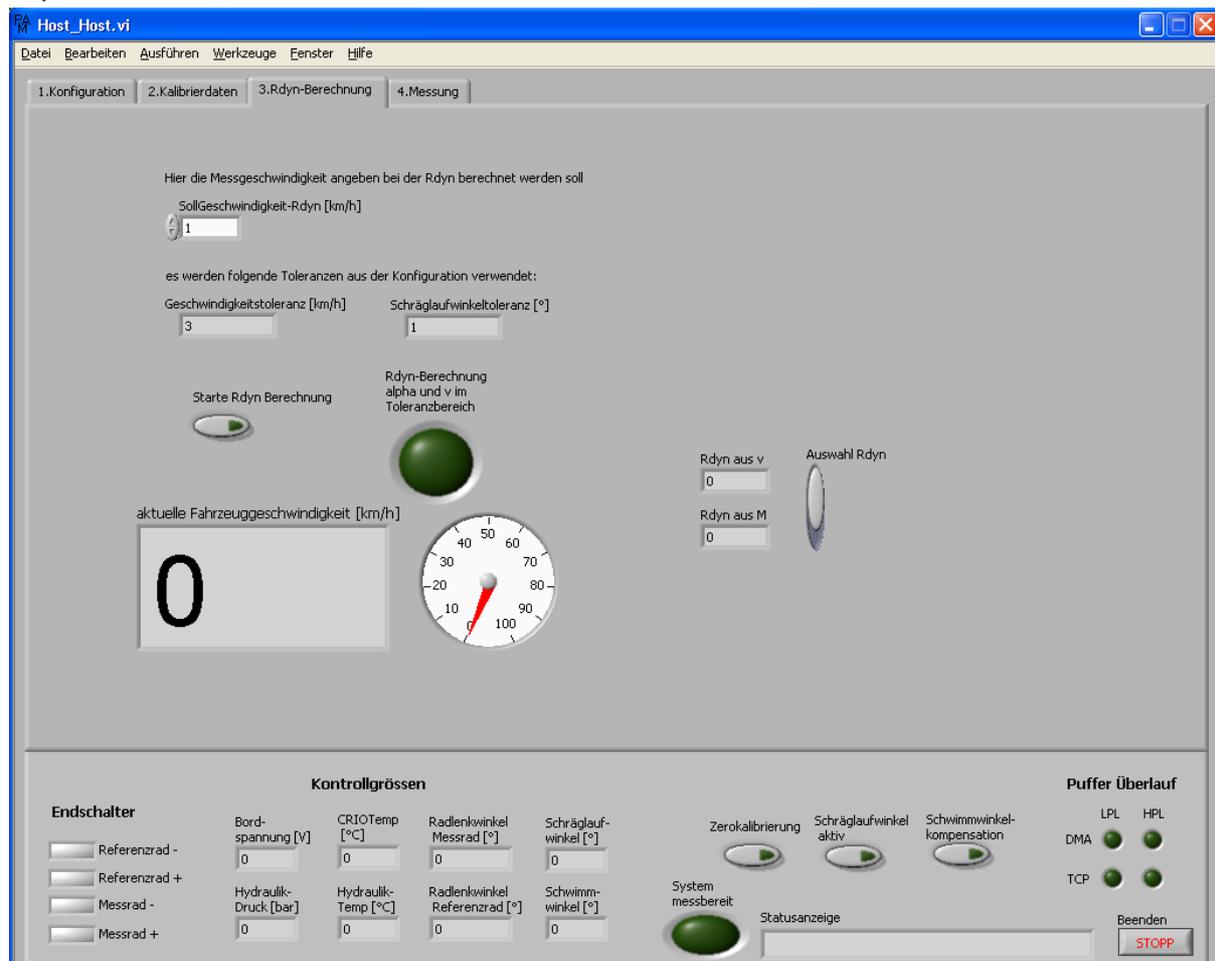


In der Kalibrierdatenbenutzereingabeoberfläche, ist es nicht notwendig etwas zu ändern, da alle Standardkalibrierdaten voreingestellt sind. Auf der linken Seite können die Winkelkalibrierdaten geändert werden, in der Mitte oben stehen die Messfelgenkalibrierdaten und in der Mitte unten die restlichen Kalibrierdaten.

Auf der rechten Seite kann der Benutzer die Stufen der Bremsmomentverstellung verändern. Damit lassen sich die Punkte auf der μ -Schlupf-Kurve verschieben.

Sind alle Kalibrierdaten so wie sie der Benutzer haben möchte, drückt er auf den Button „START“. Mit diesem Button wird der Verbindungsaufbau zu CRIO gestartet. Wurde der Verbindungsaufbau gestartet, sind weitere Veränderungen im Konfigurationsmenü sowie im Kalibrierdatenmenü ohne Nutzen, da diese Änderungen nicht mehr an das CRIO übertragen werden. Möchte der Benutzer nach betätigen des START-Buttons eine Einstellung in der Konfiguration oder in den Kalibrierdaten ändern, beendet er erst die Verbindung durch drücken des Buttons „STOPP“ im Kontrollmenü unten rechts. Danach kann über das Menü „Bearbeiten“ → „Standardwerte wiederherstellen“ die grafische Benutzerschnittstelle wieder zurück in den Urzustand gesetzt und erneut begonnen werden.

R_{dyn}-Berechnung:



Die vorherige Abbildung zeigt die GUI, der R_{dyn}-Berechnung. Im oberen Eingabefeld „Sollgeschwindigkeit-R_{dyn}“ gibt der Benutzer die Geschwindigkeit an bei der R_{dyn} gemessen werden soll. Die zwei Ausgabeelemente darunter erinnern den Benutzer daran, welche Ge-

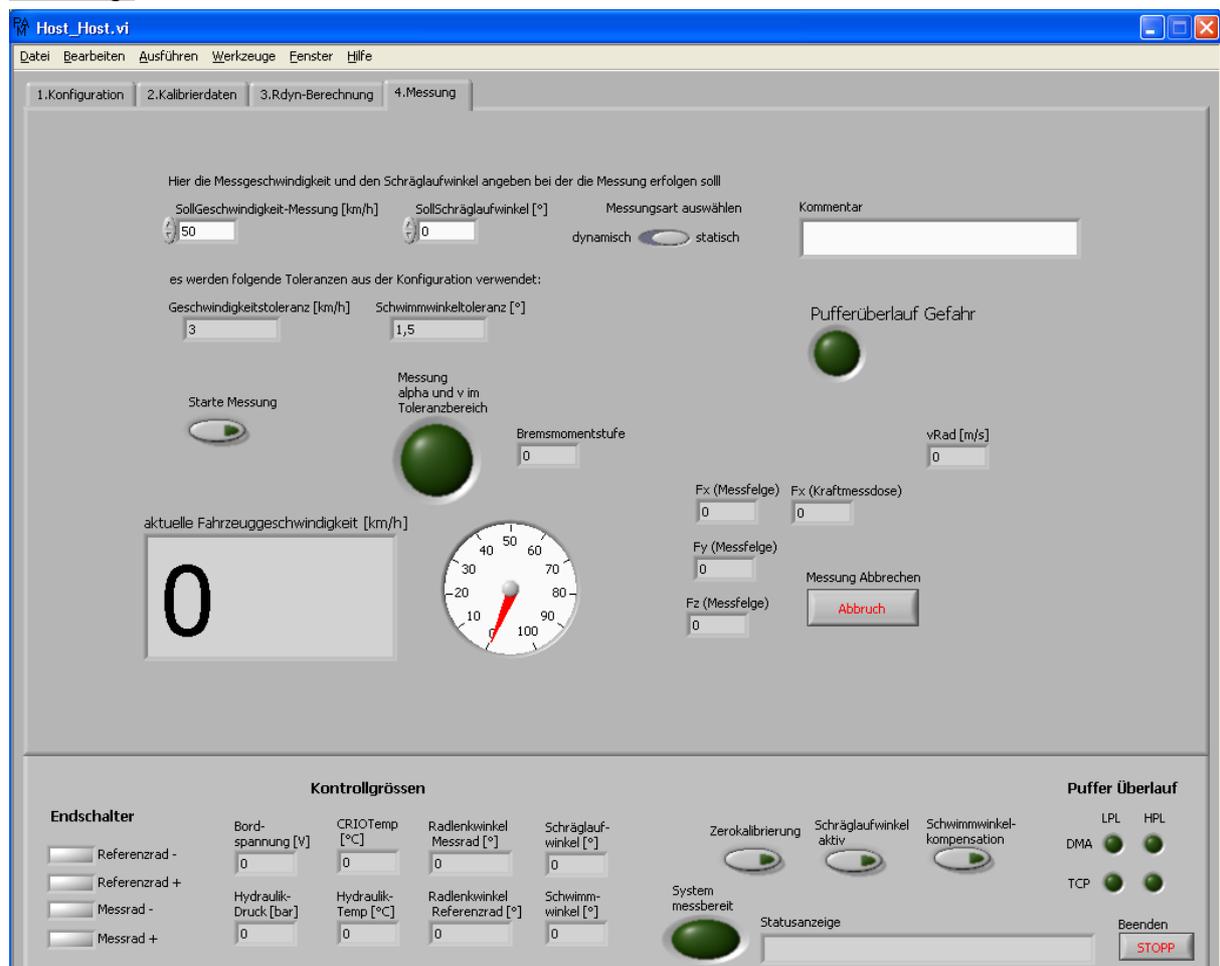
schwindigkeitstoleranz und welche Schräglaufwinkeltoleranz er im Konfigurationsmenü eingegeben hat.

Durch klicken auf den Button „Starte R_{dyn} -Berechnung“ wird die R_{dyn} -Berechnung gestartet.

Die aktuelle Fahrzeuggeschwindigkeit wird in dem großen Ausgabeelement angezeigt sowie zusätzlich dahinter als analoge Anzeige. Befindet sich die aktuelle Fahrzeuggeschwindigkeit im Geschwindigkeitstoleranzbereich, leuchtet die LED „ R_{dyn} -Berechnung alpha und v im Toleranzbereich“ auf, wenn auch die Schräglauftoleranz eingehalten wird.

Ist die R_{dyn} -Berechnung beendet erlischt die LED, der Button „Starte R_{dyn} -Berechnung“ wird inaktiv und der berechnete dynamische Reifenhalmeser wird in den beiden Ausgabeelementen auf der rechten Seite ausgegeben. Mittels dem Schalter „Auswahl R_{dyn} “ kann zwischen „ R_{dyn} aus v“ und „ R_{dyn} aus M“ gewählt werden. Die Standardstellung wählt das R_{dyn} welches über die Geschwindigkeit berechnet wurde. Das andere R_{dyn} ist in der aktuellen Implementierung nicht zu gebrauchen.

Messung:



Die vorherige Abbildung zeigt die grafische Benutzerschnittstelle zur Messungsdurchführung. In dem Feld „SollGeschwindigkeit-Messung“ gibt der Benutzer die Geschwindigkeit an bei der die Messung gefahren wird. Im Feld „SollSchräglauf“ wird der Schräglauf eingestellt. Durch verschieben des Buttons „Messungsart auswählen“ kann der Benutzer von „statisch“ auf „dynamisch“ wechseln. Bei „dynamisch“ werden alle Messdaten aufgezeichnet, während bei „statisch“ nur dann die Messdaten aufgezeichnet werden wenn die Geschwindigkeit und der Schwimmwinkel innerhalb des Toleranzbereichs liegen, was durch das aufleuchten der LED „Messung alpha und v im Toleranzbereich“ signalisiert wird. In das Eingabefeld Kommentar kann der Benutzer Kommentare eingeben, die direkt mit den Messdaten in die Messdatei geschrieben werden. Die Ausgaben Geschwindigkeitstoleranz und Schwimmwinkeltoleranz dienen dazu den Benutzer an die eingestellten Werte zu erinnern.

Durch drücken des Buttons „Starte Messung“ wird die Messungsdurchführung gestartet. Das große Geschwindigkeitsdisplay dient wie bei der R_{dyn} -Berechnung dazu den Fahrer über die aktuelle Geschwindigkeit zu informieren, damit dieser diese bei Bedarf ändert. Ist die LED „Messung alpha und v im Toleranzbereich“ für die eingestellte Zeit „Stufencounter_delta“ aktiv gewesen, wird die nächst höhere Bremsmomentstufe eingestellt. Das Ausgabeelement Bremsmomentstufe informiert den Benutzer über die aktuelle Bremsmomentstufe. Die Ausgabeelemente F_x , F_y , F_z Messfelge und F_x Kraftmessdose, sowie v_{Rad} dienen dazu den Benutzer über den aktuellen Verlauf der Messung zu informieren.

Durch betätigen des Buttons „Messung Abbrechen“ kann die Messungsdurchführung beendet werden. Dabei wird die Bremsmomentstufe zurück auf 0 gesetzt, damit ein Fahren ohne Bremskraft möglich ist.

Anhang J Bedienungsanleitung für die Matlab-Auswertung

Das m-File „auswertung.m“ wird geöffnet. In der Variable `fid = fopen('filename.lvm','r');` ist der String ‚filename‘, durch den Pfad der Messdatei die ausgewertet werden soll, zu ersetzen. Danach wird Matlab gestartet und das m-File ausgeführt. Die Auswertung erfolgt automatisch und erstellt die Reibwert-Schlupf-Kurven.