



Gerald Singer, BSc MSc

Vollautomatisierung eines Motorprüfstands für konventionelle Verbrennungskraftmaschinen

MASTERARBEIT

zur Erlangung des akademischen Grades
Diplomingenieur
Masterstudium Maschinenbau

eingereicht an der

Technischen Universität Graz

Betreuer

Assoc. Prof. Dipl.-Ing. Dr. techn. Roland Kirchberger
Institut für Verbrennungskraftmaschinen und Thermodynamik

Zweitbetreuer/in

Dipl.-Ing. Dr.techn. Stephan Schmidt
Institut für Verbrennungskraftmaschinen und Thermodynamik

Graz, März 2019



Institut für Verbrennungskraftmaschinen und Thermodynamik
Vorstand: Univ.-Prof. DI Dr. Helmut Eichseder

Vorwort

Diese Diplomarbeit ist am Institut für Verbrennungskraftmaschinen und Thermodynamik, im Fachbereich Konstruktion und Auslegung, an der Technischen Universität Graz durchgeführt worden.

Ich möchte mich bei dem Institutsvorstand Herrn Univ.-Prof. Dipl.-Ing. Dr. techn. Helmut Eichseder und dem Leiter des Forschungsbereichs Konstruktion und Auslegung Herrn Assoc.-Prof. Dipl.-Ing. Dr. techn. Roland Kirchberger dafür bedanken, dass Sie es mir ermöglicht haben, diese Diplomarbeit zu verfassen.

Ein großer Dank gilt meinem Betreuer Herrn Dipl.-Ing. Dr. techn. Stephan Schmidt, der mir bei der Durchführung der Arbeit mit fachlicher Unterstützung zur Seite stand.

Hervorheben möchte ich, dass es mir ermöglicht wurde, ein Double Degree Programm durchzuführen. Herr Schmidt hat dabei die Organisation seitens der TU Graz sehr zielführend und kompetent übernommen. Ich schätze das Institut für seine Flexibilität und offenen Möglichkeiten, welche mir das Arbeiten sehr angenehm gestaltet haben.

Zusätzlich möchte ich noch meinen Arbeitskollegen für viele interessante Diskussionen danken, wobei ich Herrn Dipl.-Ing. Hans-Jürgen Schacht für viele Hinweise seitens der Elektronik und Regelungstechnik und die Herren Dipl.-Ing. Reinhard Stelzl und Dipl.-Ing. Gerd Neumann für die Unterstützung der Prüfstandsbedienung erwähnen möchte.

Ein weiterer Dank gebührt meinen Eltern, die mich jederzeit unterstützen. Bei meinen Freunden möchte ich mich für die schöne Zeit miteinander bedanken. Der größte Dank gebührt meiner ehemaligen Freundin Sophie. Ohne ihre Unterstützung hätte ich das Studium und die Auslandsaufenthalte nicht geschafft.

Inhaltsverzeichnis

FORMELZEICHEN, INDIZES UND ABKÜRZUNGEN	VI
EIDESSTATTLICHE ERKLÄRUNG	VII
ZUSAMMENFASSUNG	VIII
ABSTRACT.....	IX
1 EINLEITUNG	1
1.1 MOTIVATION	1
1.2 ZIELE.....	1
2 THEORETISCHE GRUNDLAGEN	2
2.1 SYSTEM TEST-LEVELS.....	2
2.2 AUTOMATISIERUNG MOTORPRÜFSTAND	2
2.3 REGELUNGSTECHNIK UND REGLERPARAMETRISIERUNG	3
2.3.1 <i>PID-Regler</i>	3
2.3.2 <i>Digitaler Regelkreis</i>	5
2.3.3 <i>Lösungsweg für Regelungsaufgaben</i>	6
2.3.4 <i>Anforderungen und Testfunktionen des Reglers</i>	7
2.3.5 <i>Reglerentwurf mit empirischen Einstellregeln</i>	8
3 SYSTEMARCHITEKTUR PRÜFSTAND UND ENTWICKLUNGSSCHRITTE	11
3.1 AUSGANGSBASIS MASTERARBEIT [1]	11
3.2 ENDSTATUS/SCHNITTSTELLE MASTERARBEIT [1]	13
3.3 ZIELARCHITEKTUR	15
4 BESTEHENDER AUTOMATISIERUNGSGRAD.....	17
4.1 GANG PARAMETER	18
4.2 SCHALTEN	19
4.3 AUTOMATIK SCHALTPUNKTE	20
4.4 BREMSEN.....	20
4.5 FAHRZEUG UND STRASSE	21
5 MODIFIZIERUNGEN UND IMPLEMENTIERUNGEN.....	23
5.1 KUPPLUNGSAKTUATORIK	23
5.1.1 <i>Verwendete Komponenten</i>	24
5.1.2 <i>Leitung- und Anschlussplan</i>	27
5.1.3 <i>Steuerung des Kupplungsaktuators</i>	28
5.2 GESCHWINDIGKEITSREGELUNG	29
5.3 EMULIERTES MECHANISCHES BREMSEN	31
6 VERIFIZIERUNG VOLLAUTOMATISierter PRÜFSTAND.....	32
6.1 GÜTEGRAD TEILAUTOMATISIERUNG UND VOLLAUTOMATISIERUNG	32
6.2 WIEDERHOLBARKEIT VOLLAUTOMATISierter MOTORPRÜFSTAND	34
6.3 VERIFIZIERUNG VOLLAUTOMATISierter MOTORPRÜFSTAND MIT ROLLENPRÜFSTAND UND THEORETISCHER LEISTUNG	36
7 ZUSAMMENFASSUNG.....	38
8 AUSSICHT.....	39
8.1 SYSTEMARCHITEKTUR HYBRIDREGELUNG	39
8.2 KENNFELD OPTIMIERUNG.....	39
8.3 VOLLAUTOMATISiertes REALES SCHALTEN	40
9 ANHANG	41
9.1 ERSTMALIGER AUFBAU UND INBETRIEBNAHME VOLLAUTOMATISIERUNG - SCHRITT-FÜR-SCHRITT ANLEITUNG.....	41

9.1.1	<i>Schritt 1: Vorbereitungen - Tornado Designer</i>	41
9.1.2	<i>Schritt 2: Vorbereitung – Kupplungsaktuator</i>	45
9.1.3	<i>Schritt 3: PID-Geschwindigkeitsregler Einstellung</i>	54
9.1.4	<i>Schritt 4: Emulierte mechanische Bremse Einstellung</i>	55
9.2	DURCHFÜHREN EINES VOLLAUTOMATISIERTEN PRÜFLAUF	55
9.3	GESAMTSYSTEM UND TEILSYSTEME INKLUSIVER VARIABLEN UND REGELGRÖßEN	55
9.3.1	<i>Variablen</i>	56
9.3.2	<i>Kupplung Öffnen/Schließen Befehlsprogramme</i>	57
9.3.3	<i>PID-Geschwindigkeitsregler</i>	58
9.3.4	<i>Emulierte mechanische Bremse Befehlsprogramm</i>	59
9.3.5	<i>Andere Funktionen</i>	60
9.4	ARDUINO UNO QUELLCODE	60
9.5	PRÜFLING BMW G 650 GS 2 SPARK.....	62
LITERATUR		63

Formelzeichen, Indizes und Abkürzungen

Lateinische Formelzeichen

a	m/s^2	Beschleunigung
d	m	Durchmesser
f	s^{-1}	Frequenz
F_0	N	Rollwiderstandskoeffizient
F_2	$\text{N}/(\text{km/h})^2$	Luftwiderstandskoeffizient
I	$\text{kg}\cdot\text{m}^2$	Massenträgheitsmoment
k_D	1	Differenzial-Anteil
k_I	1	Integral-Anteil
k_P	1	Proportional-Anteil
m	kg	Masse
\dot{m}	kg/s	Massenstrom
M	$\text{N}\cdot\text{m}$	Drehmoment
n	U/min	Drehzahl
P	$(\text{k})\text{W}$	Leistung
T_D	s	Vorhaltezeit
T_I	s	Nachstellzeit
T_t	s	Verzugszeit
v	m/s	Geschwindigkeit
v_{Ist}	m/s	Aktuelle Geschwindigkeit
v_{Soll}	m/s	Vorgegebene Geschwindigkeit

Konstanten

$g_n = 9,8067$	m/s^2	Normfallbeschleunigung
----------------	----------------	------------------------

Griechische Formelzeichen

α	$\%$	Drosselklappenstellung/-vorgabe
ρ	kg/m^3	Dichte
δ_v	1	Gütegrad Geschwindigkeitsabhängig
φ	$^\circ \text{KW}$	Kurbelwinkel

Operatoren und Bezeichnungen

Δ	Differenz zweier Größen
----------	-------------------------

Weitere Indices und Abkürzungen

ADAC	Advanced Data Acquisition and Control Unit
CAN	Controller Area Network
ECU	Engine Control Unit
MCU	Multi Control Unit
RLS	Road Load Simulation
VKM	Verbrennungskraftmaschine
WMTC	World Motorcycle Test Cycle

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.



Gerald Singer

Graz, 26.03.2019

Zusammenfassung

Ziel der vorliegenden Diplomarbeit ist es, auf einem Motorprüfstand Fahrzyklen konventioneller Verbrennungsmotoren automatisiert abfahren zu können. Ausgangspunkt dieser Arbeit ist ein Stationärprüfstand, der soweit modifiziert wurde, sodass vollautomatisierte Fahrzyklen abgefahren werden konnten. Zum vorgegebenen Prüfstand wird ein Kupplungsaktuator, ein Geschwindigkeitsregler sowie eine emulierte „mechanische“ Bremse implementiert. Die notwendigen Parametrisierungen und Einstellungen für das Gesamtsystem werden eingestellt, um automatisierte transiente Fahrzyklen abfahren zu können. Ein wesentlicher Bestandteil der Arbeit ist, dass die Dokumentation so ausgeführt ist, dass auch andere Verbrennungsmotoren eingestellt und vollautomatisiert betrieben werden können.

Die verwendeten sowie veränderbaren Variablen, Regelgrößen und Stellgrößen jedes einzelnen Teilsystems werden grafisch dargestellt und die Funktionen erklärt. Eine Schritt-für-Schritt Anleitung der notwendigen Einstellungen für das automatisierte Abfahren eines Fahrzyklus werden erklärt. Dabei wurde ein neuer Verbrennungsmotor auf dem Prüfstand montiert und die einzelnen Schritte im Detail erklärt. Hier ist noch zu erwähnen, dass bereits Vorarbeiten in Form einer anderen Masterarbeit durchgeführt wurden [1].

Abstract

Goal of this thesis is to develop a fully automated engine test bench for conventional IC engines to drive transient drive cycles. Starting point is a stationary engine test bench which was extended with a clutch actuation, a digital speed controller and an emulated mechanical brake. Additionally, the total control of the full engine test bench is implemented. An essential point of this work is that other IC engines can be set up to drive automatically as well.

All necessary variables and control parameters of each part system are described to reach a fully automated engine test bench. A step-by-step guidance of all the necessary settings and procedures are described to perform automated tests on the engine test bench.

1 Einleitung

1.1 Motivation

Durch die notwendige steigende Produktivität und zur Erhaltung der Konkurrenzfähigkeit ist eine Reduzierung der Entwicklungszeit sowie optimale Ausschöpfung der Ressourcen in jedem Entwicklungsprojekt essentiell. In der vorliegenden Arbeit betrifft dies einen stationären Motorprüfstand, wobei dieser bis zur Durchführung der Masterarbeit stationär bzw. teilautomatisiert betrieben wurde. Eine Vollautomatisierung des Motorprüfstands hat viele Vorteile hinsichtlich Produktivität, Prüfstandsauslastung und zur Vorhersage von Kennwerten eines Verbrennungsmotors.

Durch die Automatisierung können verschiedenste transiente Fahrzyklen abgefahren werden, wobei diese unter einer emulierten Fahrzeuglast durchfahren werden. Die Fahrwiderstände sowie der spezifische Fahrzyklus wird von der Gesetzgebung vorgegeben. Nicht nur gesetzgebungsrelevante Zyklen, sondern auch verschiedene Fahrzeugvarianten/-konfigurationen können bereits vorab auf dem Motorprüfstand getestet werden. Für die Durchführung der automatisierten Prüfläufe wird ein BMW G 650 GS Verbrennungsmotor mit den Spezifikationen im Anhang in Kapitel 9.5 herangezogen.

Die Kennwerte des Verbrennungsmotors am Motorprüfstand können zur Vorhersage von Leistung, Verbrauch und Emissionen für das Gesamtfahrzeug auf der Fahrzeugrolle verwendet werden. Dies bringt den Vorteil, dass in einem frühen Entwicklungsstadium bereits auf die Leistungs-Kriterien des Gesamtfahrzeugs geschlossen werden kann.

1.2 Ziele

Aufbauend auf Vorarbeiten zur Erarbeitung eines Teilautomatisierungskonzeptes für einen Motorprüfstand [1] wird ein hochdynamischer Motorprüfstand für vollautomatisierte Prüffahrten adaptiert, umgerüstet und im Probetrieb evaluiert.

Ziel ist, dass der Motorprüfstand transiente Fahrzyklen vollkommen automatisiert abfahren kann. Dabei liegt der Fokus auf der Verwendbarkeit des Systems für zukünftige Entwicklungsarbeiten und auf einer einfachen Handhabung des vollautomatisierten Motorprüfstands.

Zum Erreichen des Ziels sind folgende Schritte notwendig:

- Analyse des gesamter Motorprüfstands
- Durchführung von Modifikationen und Implementierungen
- Abfahren von automatisierten Fahrzyklen
- Auswertung und Verifizierung der Messungen am Motorprüfstand
- Erläuterung der Zusammenhänge der Variablen, Regelgrößen und Stellgrößen der Teilsysteme sowie des Gesamtsystems
- Schritt-für-Schritt Anleitung der notwendigen Einstellung für zukünftige automatisierte Fahrzyklen

Zur besseren Lesbarkeit wurde die Schritt-für-Schritt Anleitung für den Betrieb im vollautomatisierten Modus in den Anhang gegeben.

2 Theoretische Grundlagen

2.1 System Test-Levels

Der Entwicklungsprozess bis zum vollständigen Gesamtfahrzeug durchläuft viele Entwicklungsschritte. Diese können in verschiedene Teilschritte mit unterschiedlichem Hardwareanteil unterteilt werden. In Abbildung 2-1 sind die verschiedenen Stadien der Entwicklung dargestellt, wobei dies von reiner Simulation ohne Hardware, der „Desktop Simulation“, bis zum Testen des Gesamtfahrzeuges, dem „Vehicle Level Testing“, durchläuft.

Der Motorprüfstand ist ein Teil im Entwicklungsprozess zur Optimierung des Motors, der sich in der folgenden Abbildung im „Powertrain Dyno Testing“ einordnet.

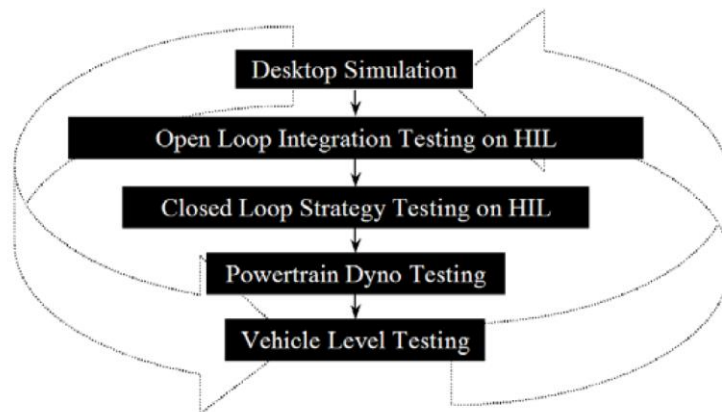


Abbildung 2-1: System, Zustand und Prozess [2]

Der Motorprüfstand wird verwendet, um die ECU für den Verbrennungsmotor optimal zu kalibrieren. Dabei ist neben dem stationären Applizieren der ECU auch das Überprüfen des transienten Verhaltens notwendig. Dies kann mit einem automatisierten Motorprüfstand durchgeführt werden, der auch transiente Fahrzyklen abfährt.

2.2 Automatisierung Motorprüfstand

Ein automatisierter Motorprüfstand bietet viele Vorteile gegenüber einem manuell, stationär betriebenen Motorprüfstand [3]:

- Reproduzierbare Versuchsbedingungen
- Hohe Prüfstand Sicherheit durch Vermeidung von manuellen Arbeiten
- Reduzierung der Auslastung des Prüfstandspersonals

Die Automatisierung birgt auch Risiken, da Überwachungsgrößen in der Prüfstandssteuerung eingebaut werden müssen. Diese sind notwendig, um auf etwaiges Fehlverhalten des Verbrennungsmotors bzw. der Prüfstandskomponenten reagieren zu können.

Die Kommunikation zwischen den verschiedenen Prüfstandssystemen kann ebenfalls zu Schwierigkeiten führen, da bei Fehlen bzw. Ausfall eines Teils, der Prüfstand die notwendigen Eingangsgrößen und Ausgangsgrößen für die Regelung nicht mehr zur Verfügung hat. Der Prüfstand hat mehrere (Teil-)Systeme, diese sind die Tornado Prüfstandssteuerung, das INCA bzw. ETAS ECU Kalibriertool und dem Dewetron Messsystem für Grad-Kurbelwinkel aufgelöstes Messen. Die Kommunikation der verschiedenen Systemen mit der Tornado Prüfstandssteuerung muss hergestellt werden und kann sich als herausfordernd darstellen, was ebenfalls in [4] festgestellt wurde.

2.3 Regelungstechnik und Reglerparametrisierung

Ein wesentlicher Punkt für die Automatisierung des Motorprüfstands ist, dass das Gesamtsystem innerhalb seiner physikalischen Grenzen geregelt werden muss und daher der Schwerpunkt der theoretischen Grundlagen auf die Regelungstechnik gelegt wird.

„Regeln ist das selbsttätige Beeinflussen eines technischen Prozesses in gezielter Weise“, wobei dies in einem geschlossenen Kontrollkreis, im Unterschied zum Steuern, abläuft. [5]

Abbildung 2-2 zeigt die Grundstruktur eines Regelkreises. Dieser besteht aus einer Führungsgröße bzw. dem Vergleichswert, der Regelabweichung, der Regeleinrichtung bzw. dem Regler, welcher eine Stellgröße ausgibt und die Regelstrecke bzw. den Prüfling ansteuert. Dieser wird durch verschiedene Umwelteinflüsse gestört und die Regelgröße wird wieder rückgeführt. Somit entsteht ein geschlossener Regelkreis.

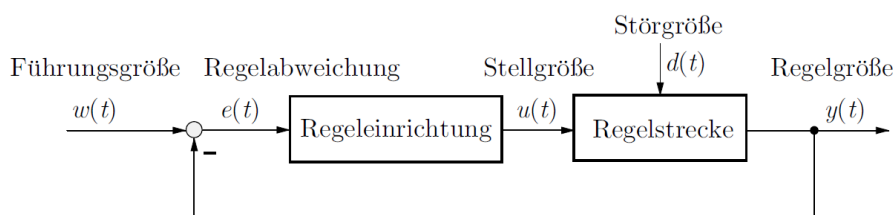


Abbildung 2-2: Grundstruktur Regelkreis [6]

Die prinzipielle Wirkungsweise des Reglers kann in drei Schritte eingeteilt werden, dem Messen der Regelgröße, dem Vergleichen mit der Führungsgröße und dem Stellen der Stellgröße.

Allgemein kann zwischen einer Regelung mit „schaltenden Reglern“ und stetigen Reglern unterschieden werden, wobei unter den schaltenden Reglern zum Beispiel ein Zweipunktregler und unter den stetigen Reglern ein PID-Regler fällt [7].

Die Tornado Prüfstandssteuerung kann einen bzw. mehrere PID-Regler digital ausführen. Daher wird näher auf diesen Regler eingegangen.

2.3.1 PID-Regler

Der PID-Regler besteht aus drei Teilen, dem Proportional-, dem Integral- und dem Differenzialglied. Das P-Glied ist am Ausgangssignal direkt proportional zum Eingangssignal, dabei erfährt die Ausgangsgröße eine Änderung ohne zeitliche Verzögerung. Jedoch kann eine Störung nicht vollständig ausgeregelt werden. Daher ist ein I-Glied notwendig, das Störungen ohne bleibende Regeldifferenz ausregelt. Der Nachteil des I-Glieds ist, dass eine Änderung der Eingangsgröße nur zu einer langsamen Reaktion der Ausgangsgröße führt. „Das D-Glied ist die Ausgangsgröße proportional der Änderung seiner Eingangsgröße“ [8]. Das D-Glied wird für eine schnelle Ausregelung von Störungen verwendet. [8] Ebenfalls kann der D-Anteil als Methode betrachtet werden, um ein unerwünschtes Tiefpassverhalten der Regelstrecke zu beseitigen [9].

Der PID-Regler setzt sich strukturell wie in Abbildung 2-3 dargestellt zusammen.

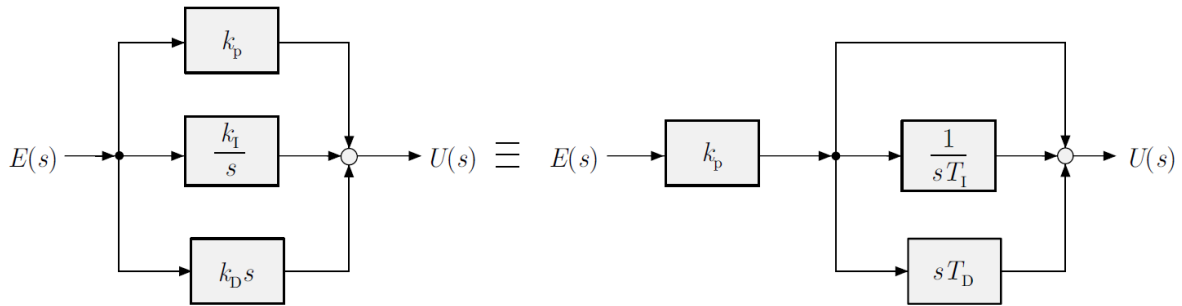


Abbildung 2-3: Struktur PID-Regler [6]

Wobei k_p , k_I und k_D die Übertragungsfunktionen der jeweiligen Glieder sind. $E(s)$ ist die Reglerabweichung und $U(s)$ die Reglerausgangsgröße in der Laplacedomain. Der P-Anteil wird öfters aus der Parallelschaltung herausgezogen und neue Variablen werden eingeführt (Gleichung (2-1) und (2-2)).

$$\text{Nachstellzeit } T_I = \frac{k_p}{k_I} \quad (2-1)$$

$$\text{Vorhaltezeit } T_D = \frac{k_p}{k_D} \quad (2-2)$$

Somit ergibt sich für den PID-Regler in der Zeitdomain die Gleichung (2-3):

$$u(t) = k_p * e(t) + \frac{k_p}{T_I} * \int e(t) * dt + k_p * T_D * \frac{de(t)}{dt} \quad (2-3)$$

Der PID-Regler besteht somit aus drei Gliedern die sich überlagern. Die Übergangsfunktion kann durch Abbildung 2-4 dargestellt werden.

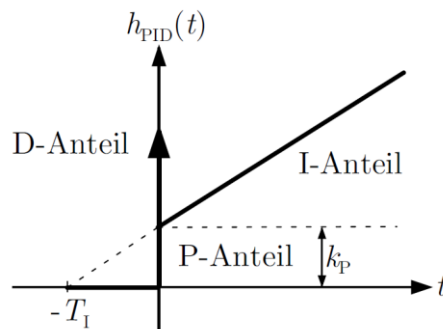


Abbildung 2-4: Übergangsfunktion eines PID-Reglers [6]

Zusammengefasst wirkt der P-Anteil direkt, der I-Anteil integrierend und der D-Anteil auf Veränderungen des Eingangssignals auf das Ausgangssignal.

Es ergeben sich Unterschiede im Regelkreisverhalten, wenn ein digitaler PID-Regler verwendet wird, da ein digitaler Regler zeitdiskret und nicht kontinuierlich realisiert ist. Durch die Implementierung eines digitalen Reglers in der Tornado Prüfstandssteuerung werden der digitale Regelkreis und deren Änderungen analysiert.

2.3.2 Digitaler Regelkreis

Der Regler wird mittels eines Algorithmus auf einem, im Kern digitalen, Rechner realisiert, daher ist auch die Stellgröße nicht kontinuierlich. Das heißt, dass aktuelle Messwerte für die Berechnung der Regelabweichung herangezogen werden und die Stellgröße zeitdiskret berechnet wird.

Abbildung 2-5 zeigt schematisch wie ein digitaler Regler funktioniert. Die Regelabweichung $e(t)$ wird abgetastet und der Analog-Digital-Wandler führt dieses Signal in ein digitales über, wobei die Abtastpunkte k mit der Zeit t durch die Abtastzeit T definiert sind. Nach erfolgreicher Berechnung der Stellgröße mit dem Regler wird diese mit dem Digital-Analog-Wandler in ein analoges Signal übergeführt. Durch ein Halteglied wird dieses Signal in eine Zeitfunktion gewandelt und der Regelstrecke zugeführt. [10]

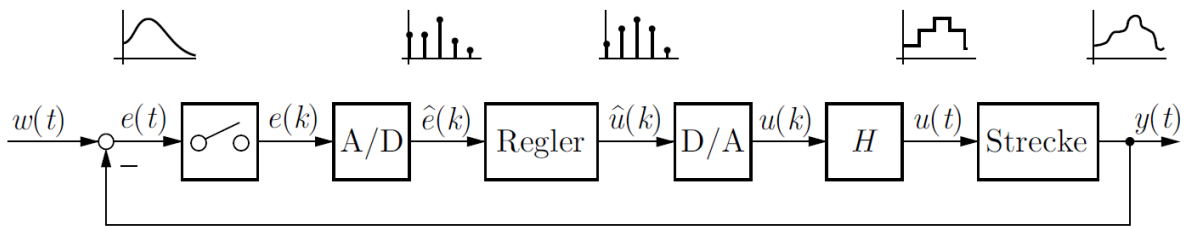


Abbildung 2-5: Digitaler Regelkreis [10]

Die Unterschiede zu einem kontinuierlichen Regler besteht darin, dass ein digitaler Regler zusätzlich noch A/D bzw. D/A-Wandler, die Abtastung des Signals sowie das Halteglied hat.

Durch die sehr hohe Auflösung können die Quantisierungsfehler der A/D bzw. D/A-Wandler vernachlässigt werden. [6]

Jedoch kann die Abtastfrequenz des Signals nicht vernachlässigt werden, da bei einer zu niedrigen Abtastfrequenz der Aliasing-Effekt auftreten kann.

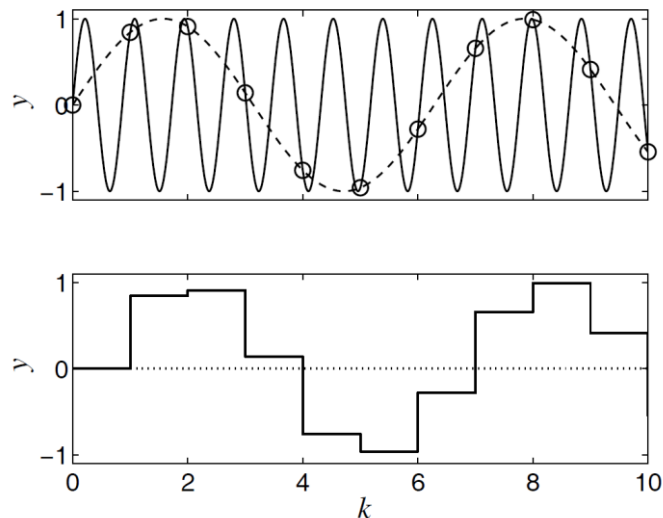


Abbildung 2-6: Aliasing-Effekt [6]

Der Aliasing-Effekt ist in Abbildung 2-6 dargestellt, bei dem die Sinusschwingung mit der hohen Frequenz durch die niedrige Abtastfrequenz nicht sichtbar ist. Durch die Sample-and-Hold Schaltung wird dadurch eine „falsche“ Sinuskurve dem Messsystem sichtbar. Daraus folgt das Abtasttheorem, das besagt, dass die Abtastfrequenz mindestens doppelt so hoch sein muss wie die höchste auftretende Frequenz. Ein Antialiasing-Filter kann vor dem Taster die hochfrequenten Schwingungen herausfiltern, sodass Aliasing nicht auftreten kann.

Das Halteglied erzeugt aus einem diskretisierten Signal ein zeitkontinuierliches Signal. Das Halteglied ist näherungsweise ein Totzeitglied, das mit der Totzeit der halben Haltezeit wirkt. Das heißt, dass das Halteglied eine Phasenverschiebung erzeugt (Abbildung 2-7) und das Regelverhalten verschlechtert.

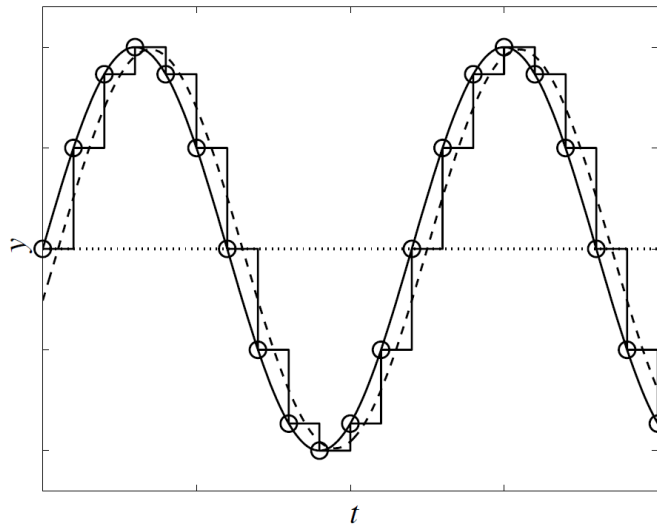


Abbildung 2-7: Halteglied Einfluss bei sinusförmigem Signal [6]

Eine Abtastfrequenz größer der zehnfachen maximal auftretenden Frequenz ist ausreichend. Eine Abtastfrequenz die 20-mal der auftretenden Frequenz ist, kann als kontinuierlich angesehen werden.

Zur Berechnung der Stellgröße tritt ebenfalls ein Berechnungsverzug auf, der bei diskreten und zeitkontinuierlichen Reglern auftritt. Diese sind jedoch bei PID-Reglern vernachlässigbar, da dies relativ „einfache“ Regler sind und rasch eine Stellgröße berechnen. [11]

Nach dem nun Regler im theoretischen Fall betrachtet wurden, werden die Lösungswege zur Erstellung eines passenden Reglers erläutert.

2.3.3 Lösungsweg für Regelungsaufgaben

Der Lösungsweg für Regelungsaufgaben kann in 7 Schritte eingeteilt werden [6]:

1. Formulierung der Regelungsaufgabe: Welche Eigenschaften soll der Regelkreis besitzen?
2. Auswahl der Regelgröße: Welche Größe soll als Regelgröße bei der Regelungsaufgabe verwendet werden? bzw. welche gut messbare Ersatzgröße kann verwendet werden?
3. Auswahl der Stellgröße: Welche Stellgröße hat einen ausreichend großen und wenig verzögerten Einfluss auf die Regelgröße unter Beachtung von physikalischen Beschränkungen?
4. Modellierung der Regelstrecke: Ein mathematisches Modell der Regelstrecke soll aufgestellt werden, welches die Grundlage für die folgenden Schritte bildet.
5. Reglerentwurf: In diesem Schritt wird der Regler entworfen und die Güteanforderungen des geschlossenen Regelkreises überprüft.
6. Analyse des Verhaltens des geschlossenen Regelkreises: Hier werden Untersuchungen des Regelkreises durchgeführt, indem auch Störungen, Messrauschen, nichtlineare Wirkungsabläufe und Modellunsicherheiten berücksichtigt werden.

7. Realisierung des Reglers: Der Regler wird in einem Rechner inklusive Reglerstruktur und Parametern realisiert.

2.3.4 Anforderungen und Testfunktionen des Reglers

Die wichtigsten Anforderungen an Regelungen können mit den folgenden vier Hauptkriterien zusammengefasst werden [12]:

1. Genauigkeit (stationäre Regelgüte): Die bleibende Regeldifferenz im stationären Fall soll gegen Null gehen.
2. Schnelligkeit (dynamische Regelgüte): Beim Auftreten von Regelabweichungen hat die Ausgangsgröße möglichst schnell und fortlaufend zu folgen, um somit jeder dynamischen Änderung folgen zu können. Dies wird mit Hilfe von Sprungsignalen getestet.
3. Stabilität: Der Regelkreis muss stabil sein und zu jeder Zeit kontrollierte Zustände annehmen. Durch die Rückkopplung der Regelgröße auf den Vergleich des Reglers könnten durch zeitliche Verzögerungen Schwingsysteme entstehen, die den Prüfstand zum Überschreiten von physikalischen Grenzen führen könnte.
4. Robustheit: Der Regelkreis muss robust sein. Durch Änderungen der Regelstrecke verändert sich auch das Verhalten des Regelkreises. Der Regler soll so ausgeführt sein, dass dieser ein zufriedenstellendes Regelverhalten hat und die ersten drei Anforderungen erfüllt sind, auch wenn sich die Parameter der Regelstrecke verändert haben.

Zur Analyse der dynamischen Regelgüte können unterschiedliche Testfunktionen erzeugt werden. Die einfachsten sowie wichtigsten Testfunktionen sind in Abbildung 2-8 gezeigt.



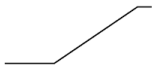

Testfunktion	Verlauf	Antwortfunktion
Impuls		Impulsantwort $g(t)$
Sprung		Sprungantwort $h(t)$
Anstieg		Anstiegsantwort $a(t)$
Schwingung		Frequenzgang $G(s)$

Abbildung 2-8: Testfunktionen [13]

Der Sprung ist für die Regelung das ungünstigste Signal, das praktisch auftreten kann.

Bei reinen I-Reglern besteht die Gefahr, dass kritische Grenzwerte übertreten werden, da diese Regler schnell zum Überschwingen der Ausgangsgröße führen. Daher wird ein Impuls als Eingangsgröße angelegt, da dieser nach einer kurzen Zeit wieder auf den Ausgangswert zurückgestellt wird. [13]

Bei Regelstrecken, die eine hohe Trägheitswirkung haben, verwendet man oft Anstiegsfunktionen. Schwingungen können ebenfalls als Testfunktion herangezogen werden. Bei einem Regler, bei dem die Soll-Geschwindigkeit mit der Ist-Geschwindigkeit verglichen wird und die Stellgröße die Drosselklappenstellung ist, sind die Anstiegstestfunktion und die Schwingungstestfunktion am sinnvollsten, da diese beiden Funktionen am ehesten in einem Fahrzyklus abgefahren werden.

2.3.5 Reglerentwurf mit empirischen Einstellregeln

Die folgenden Methoden können nur dann angewendet werden, wenn die Regelstrecke stabil ist und näherungsweise aperiodisches Übergangsverhalten hat [6]. Durch die Massenträgheiten des Verbrennungsmotors und des Prüfstands ist diese allgemeine Voraussetzung gegeben.

Unter den empirischen Einstellregeln gibt es viele verschiedene Verfahren, unter anderem der Reglerentwurf nach Ziegler-Nichols und nach Chien-Hrones-Reswick, T-Summen-Regel nach Kuhn, Verfahren nach Aström (Übergangsmethode und Frequenzantwortmethode), Verfahren nach Hallmann. Diese Methoden können übersichtlich in [14] eingesehen werden, wobei im folgenden auf die ersten beiden im Detail eingegangen wird.

Die folgenden Verfahren eruiieren die Parameterwerte k_s und T und die Verzugszeit T_t und werden im Weiteren in die Formeln (2-1) und (2-2) eingesetzt. Zusätzlich soll noch vermerkt werden, dass diese PID-Anteile „Faustformeln“ sind, das heißt, dass diese den Regler grob auslegen und die Werte noch angepasst werden können.

2.3.5.1 Verfahren nach Ziegler-Nichols

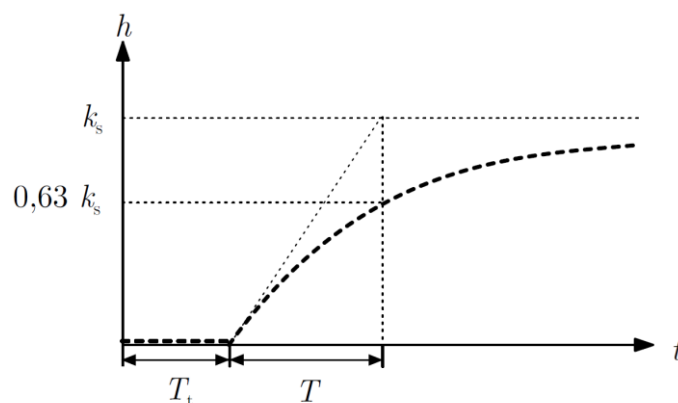
Ziegler und Nichols haben zwei Methoden entwickelt, um einen PID-Regler einzustellen. Die Tabelle 2-1 zeigt die Voraussetzungen und die Berechnungsvorschrift für die Reglerparameter.

Voraussetzung	Regler	Reglerparameter
Approximation der Regelstrecke durch ein PT_1T_t -Glied	P	$k_P = \frac{1}{k_s} \frac{T}{T_t}$
	PI	$k_P = \frac{0,9}{k_s} \frac{T}{T_t}, T_I = 3,33 T_t$
	PID	$k_P = \frac{1,2}{k_s} \frac{T}{T_t}, T_I = 2 T_t, T_D = 0,5 T_t$
Kritische Verstärkung und Periodendauer sind bekannt	P	$k_P = 0,5 k_{krit}$
	PI	$k_P = 0,45 k_{krit}, T_I = 0,85 T_{krit}$
	PID	$k_P = 0,6 k_{krit}, T_I = 0,5 T_{krit}, T_D = 0,12 T_{krit}$

Tabelle 2-1: Methode der Stabilitätsgrenze PID-Werte [6]

Methode der Übergangsfunktion (Tabelle 2-1 oben)

Regelstrecken, die eine Übergangsfunktion besitzen, kann man durch ein PT_1T_t -Modell approximieren. Dabei wird ein Sprung an das System angelegt und aus der Sprungantwort können die Parameterwerte k_s und T (bzw. die Steigung an der Wendetangente) und die Verzugszeit T_t abgelesen werden. Die somit erhaltenen Werte werden in die Tabelle 2-1 (oben) eingesetzt.

Abbildung 2-9: Übergangsfunktion eines PT_1T_t -Gliedes [6]

Vorteil dieser Methode ist, dass wenig Informationen über das System notwendig ist, jedoch werden Führungs- und Störgrößen gleichbehandelt. [6]

Methode der Stabilitätsgrenze (Tabelle 2-1 unten)

Der PID-Regler wird zuerst als P-Regler eingesetzt, wobei die Reglerverstärkung vergrößert wird bis sich der Regelkreis in Dauerschwingung befindet. Der dabei entstandene Verstärkungsfaktor k_{krit} wird als kritische Regelverstärkung abgelesen und die Periodendauer der Dauerschwingung T_{krit} ermittelt. [15] Die somit erhaltenen Werte werden in die Tabelle 2-1 (unten) eingesetzt.

2.3.5.2 Verfahren nach Chien, Hrones und Reswick [15]

Dieses Verfahren kann als Weiterentwicklung der „Methode der Übergangsfunktion“ von Ziegler und Nichols betrachtet werden. Ebenfalls wird ein Sprung angesetzt und die Sprungantwort analysiert.

Dabei wird das Führungsverhalten in zwei unterschiedliche Regelvorgänge unterteilt, dem aperiodischen und dem periodischen Regelvorgang mit ca. 20% Überschwingweite.

Aperiodischer Regelvorgang mit kürzester Dauer

Bei dieser Methode tritt kein Überschwingen auf. Die Reglerparameter werden mit den Formeln (2-4) berechnet.

$$PID - \text{Regler: } k_P = \frac{0,6}{k_S} * \frac{T}{T_t}, \quad T_I = 2 * T, \quad T_D = 0,5 * T_t \quad (2-4)$$

Periodischer Regelvorgang mit 20% Überschwingweite

Bei dieser Methode tritt ein Überschwingen von 20% des Sprungs auf. Die Reglerparameter werden mit den Formeln (2-5) berechnet.

$$PID - \text{Regler: } k_P = \frac{1,2}{k_S} * \frac{T}{T_t}, \quad T_I = 2 * T, \quad T_D = 0,42 * T_t \quad (2-5)$$

2.3.5.3 Empirisches Verfahren für Geschwindigkeit-Fahrzyklen-Regler

Im Falle einer Geschwindigkeitsregelung, bei der die vorgegebene Geschwindigkeit mit der aktuellen Geschwindigkeit verglichen wird, ist eine Testfunktion bei der ein Anstieg vorgegeben wird, am realistischsten. Daher wird auch der Regler mit einer Anstiegsfunktion ähnlich einem Fahrzyklus getestet und empirisch ausgelegt.

Die Anstiegsfunktionen werden aus dem geplanten Fahrzyklus entnommen. Dabei wird der langsamste und der schnellste Anstieg, sowie bei niedriger und hoher Geschwindigkeit ausgewählt. Diese vier Fahrprofile werden linearisiert und in einem Testzyklus öfters hintereinander angeordnet. Dieser Fahrzyklus wird abgefahren und die PID-Werte live angepasst.

Falls die eruierten „optimalen“ PID-Werte zu sehr variieren, können die PID-Werte über die Geschwindigkeit und/oder dem aktuellen Gang abhängig definiert werden.

3 Systemarchitektur Prüfstand und Entwicklungsschritte

Vorarbeiten zur Automatisierung für den Motorprüfstand wurden bereits in der Masterarbeit „Emulation of Transient Drive Cycles & Profiles on the Engine Test Bench for Conventional Motorcycles“ [1] durchgeführt. Um die genaue Schnittstelle dieser Masterarbeit „Vollautomatisierung eines Motorprüfstands für konventionelle Verbrennungsmotoren“ zu definieren, wird in diesem Kapitel die Ausgangsbasis von [1], der Endstand von [1] und die Zielarchitektur für die Vollautomatisierung beschrieben.

Für das automatisierte Ablaufen eines Fahrzyklus benötigt der Prüfstand einige Änderungen bzw. Modifikationen. Dabei ist zu beachten, dass das Durchfahren eines Fahrzyklus üblicherweise gewissen Gesetzgebungs-Standards unterliegt. In [1] und in dieser Masterarbeit wird der Fokus auf den Fahrzyklus World Motorcycle Test Cycle (WMTC) und dessen Bestimmungen gerichtet. Dabei sind die Sollgeschwindigkeit, die obere und untere Geschwindigkeitsgrenze und der Gangverlauf vorgegeben (Abbildung 3-4). Der Prüfling ist ein BMW G 650 GS Verbrennungsmotor, welcher in diesem Gesetztext in die Kategorie L3e fällt [16]. Bei einem WMTC Fahrzyklus sind ein Kaltstarttest und Messung der Emissionen notwendig, um die Typenzulassung des Motorrades zu erhalten. Der Ablauf des WMTC Fahrzyklus ist in der Gesetzgebung vorgegeben und bei der Automatisierung des Prüfstands zu beachten.

3.1 Ausgangsbasis Masterarbeit [1]

Der Prüfstand ist mit der Prüfstandssteuerung Tornado von Kristl und Seibt (KS Engineers) ausgestattet. Die allgemeine Systemkonfiguration ist in Abbildung 3-1 dargestellt. Der Prüfstand besteht aus der Verbrennungskraftmaschine, der Prüfstandsbremse bestehend aus einem Asynchronmotor und einer Wirbelstrombremse und dem ADAC ein Mess-, Regelungs- und Monitorsystem. Der ADAC kann ebenfalls mit der Multicontrolunit (MCU), dem Ethernet, dem CAN Bus und/oder Profibus verbunden sein. Die MCU ist die Schnittstelle zwischen dem Prüfstand und dem Prüfstandstechniker und besteht aus den verschiedenen Bedienelementen, die für das manuelle Regeln des Prüfstandes notwendig sind. Der Anschluss der zusätzlichen Teilsysteme ist abhängig von dem gewünschten Prüfstandslauf. Eingestellt und betrieben wird der Prüfstand mittels der Tornado Designer und Tornado RT (Tornado Runtime) Software, die als Interface für den Nutzer zur Verfügung stehen.

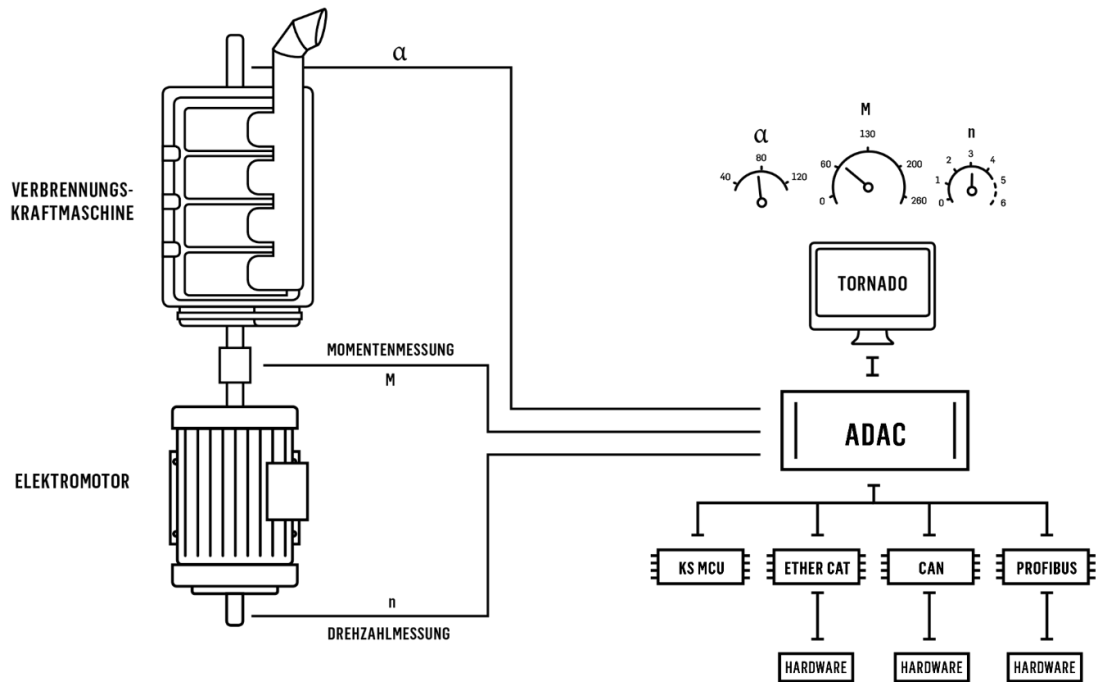


Abbildung 3-1: Systemkonfiguration Prüfstandssteuerung [17]

Vorteil dieses Prüfstandsaufbaues ist, dass das Gesamtsystem weitgehend frei konfiguriert werden kann. Dies ermöglicht, dass zusätzliche Komponenten flexibel hinzugefügt und erweitert werden können.

Der Prüfstand besitzt ebenfalls eine kontinuierliche Grenzwertüberwachung, die bei Über- bzw. Unterschreiten des Grenzwertes einen Nothalt bzw. ein anderes programmiertes Programm durchführt. Dies ist für das automatisierte Betreiben des Prüfstands wesentlich, da bei längeren Prüfzyklusläufen die Überwachungsfähigkeit des Prüfstandstechnikers nachlässt und eventuelle Reaktionen auf Grenzwertüberschreitungen nicht zeitnah ausgeführt werden können.

Die Tornado Prüfstandssteuerung weist zu diesem Zeitpunkt bereits einen geringen Automatisierungsgrad auf. Es ist ein Schaltautomat, der zwischen den Gängen softwaretechnisch schaltet, das heißt, dass die Drehzahl und das Bremsmoment auf den jeweiligen Gang anpassen kann, bereits implementiert. Ebenfalls kann der Fahrwiderstand für die Prüfstandsbremse in Abhängigkeit von der Geschwindigkeit, Beschleunigung und Steigung automatisiert gestellt werden. Anzumerken ist, dass diese Funktionen noch nicht verwendet wurden und diese im Zuge der Masterarbeit in [1] erarbeitet wurden.

Die Abbildung 3-2 zeigt die schematische Darstellung der Ausgangsbasis des Motorprüfstands.

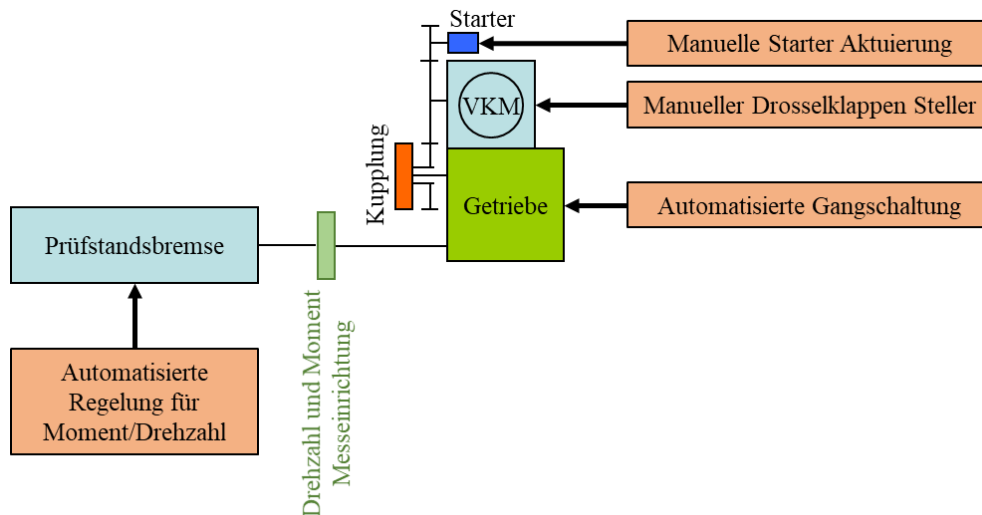


Abbildung 3-2: Ausgangsbasis Prüfstand

3.2 Endstatus/Schnittstelle Masterarbeit [1]

Der Verbrennungsmotor inklusive Getriebe, Starter und Kupplung ist aufgebaut und über eine Welle mit der Prüfstandsbremse verbunden. Die Kupplung, der Starter und die Drosselklappe müssen manuell bedient werden.

Ein Prüfprogramm startet den teilautomatisierten Prüfzyklus. Dabei wird der Fahrzyklus mit dem Geschwindigkeitsverlauf (Abbildung 3-3) und dem Gangverlauf (Abbildung 3-4) vorgegeben, geladen und die Variablen der Tornado Prüfstandssteuerung zur Verfügung gestellt.

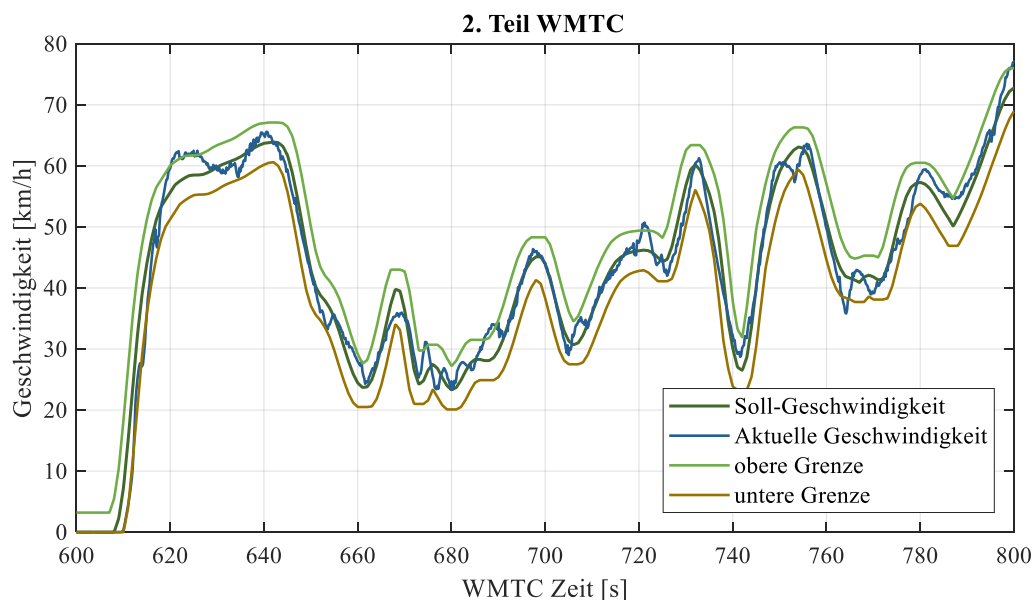


Abbildung 3-3: WMTC Geschwindigkeiten Teilausschnitt

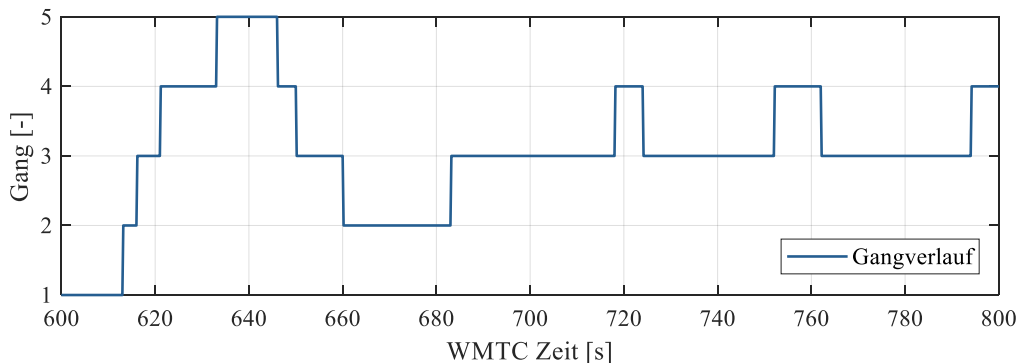


Abbildung 3-4: WMTC Gangverlauf Teilausschnitt

Beim Start des Fahrzyklus wird die Kupplung manuell gezogen und der Verbrennungsmotor manuell gestartet. Die aktuelle Geschwindigkeit wird mit der Drehzahl und den vorgegebenen Übersetzungen berechnet. Die vorgegebene Geschwindigkeit wird grafisch dargestellt und die aktuelle Geschwindigkeit wird manuell mit einem Motorradgasgriff via Gas-by-Wire geregelt, welche ebenfalls in Abbildung 3-3 dargestellt ist. Bei Gangwechsel wird die Drehzahl und das Moment für den neuen Gang gestellt. Über die gemessene Geschwindigkeit und die Übersetzungen wird der Fahrwiderstand berechnet und gestellt. Im Fall, dass der Schubtrieb des Verbrennungsmotor nicht für das Bremsen reicht, wird manuell mit dem n/α Regelbetrieb gebremst.

Um eine bessere Einsicht in die Prüfstandssteuerung zu erhalten, wird auf die Masterarbeit [1] / Kapitel 3 verwiesen. Bei dieser Masterarbeit wurde der Prüfstand bereits teilautomatisiert betrieben, wobei die Systemarchitektur schematisch in Abbildung 3-5 dargestellt ist.

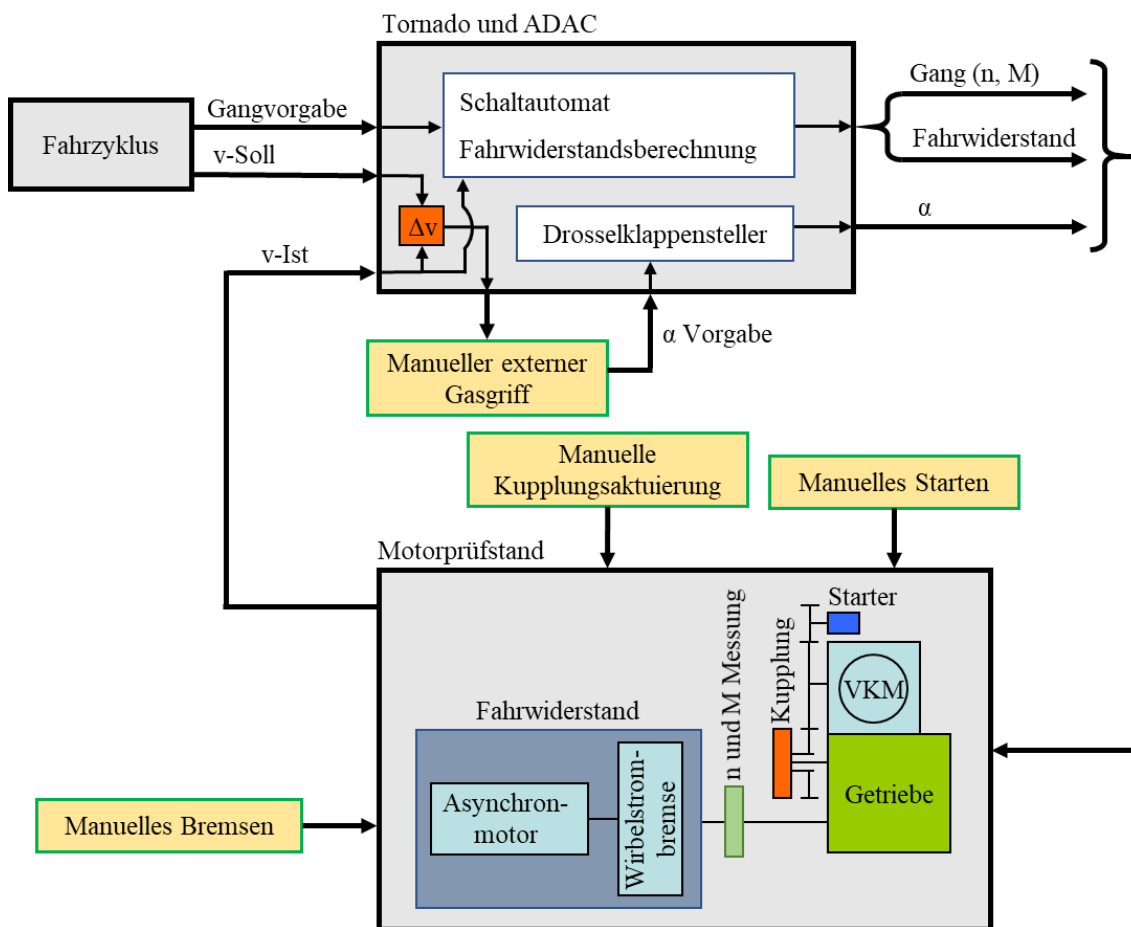


Abbildung 3-5: Teilautomatisierte Prüfstandsfahrten

3.3 Zielarchitektur

Um die Verbrennungskraftmaschine automatisiert betreiben zu können, sind die folgenden Teilsysteme so zu modifizieren, sodass diese von der Prüfstandssteuerung automatisiert bedient werden können:

Die VKM wurde entweder hochgeschleppt und die Zündung eingeschaltet oder mit dem Starter manuell gestartet. Das Starten mit dem Serienstarter, welcher am Motor angebracht ist, muss verwendet werden, um einen WMTC-Zyklus gesetzeskonform abfahren zu können. Das automatisierte Starten ist bei Start des Fahrzyklus durchzuführen.

Für die manuelle Kupplung ist ein Aktuator notwendig, der das Anfahren bzw. das Auskuppeln zum Stillstand ermöglicht. Dies ist für die meisten Fahrzyklen notwendig, da diese gewisse Stillstandszeiten bei Leerlaufdrehzahl haben.

Ein physikalisch fixierter Gang ist eingelegt. Um zwischen den Gängen schalten zu können, wurde bereits ein Schaltautomat in die Prüfstandssteuerung implementiert. Die einzelnen Getriebegänge mit deren Übersetzungsverhältnisse werden umgerechnet und die Drehzahl bzw. das Moment automatisiert berechnet und geregelt. Dieses System ist bereits vorhanden und funktioniert. Jedoch ist hier anzumerken, dass der Schaltvorgang realitätsnaher gestaltet werden könnte.

Die Fahrwiderstandskoeffizienten des Motorrades sind laut WMTC vorgegeben. Ebenfalls ist dieses System in der Prüfstandssteuerung vorhanden. Dies funktioniert insofern, dass die aktuelle Fahrgeschwindigkeit gemessen wird, daraus der Fahrwiderstand berechnet wird (4-4) und daraufhin die Prüfstandsbremse mit dem berechneten Moment geregelt wird.

Der WMTC gibt eine Soll-Geschwindigkeit mit einer Ober- und Untergrenze vor. Um dieser Geschwindigkeitsvorgabe folgen zu können, ist die Verbrennungsmotor-Drehzahl anzupassen.

Die Regelung der Drehzahl erfolgt indirekt über das Regelorgan für das Drehmoment der VKM und damit über die Drosselklappe. Dies wurde bei früheren Messungen manuell über einen Motorrad-Gasgriff eingestellt. Für die Automatisierung des Prüfstandes ist die Drosselklappenstellung in Abhängigkeit von der aktuellen und vorgegeben Geschwindigkeit zu regeln.

Die Zielarchitektur des Motorprüfstands für die Vollautomatisierung ist in Abbildung 3-6 dargestellt.

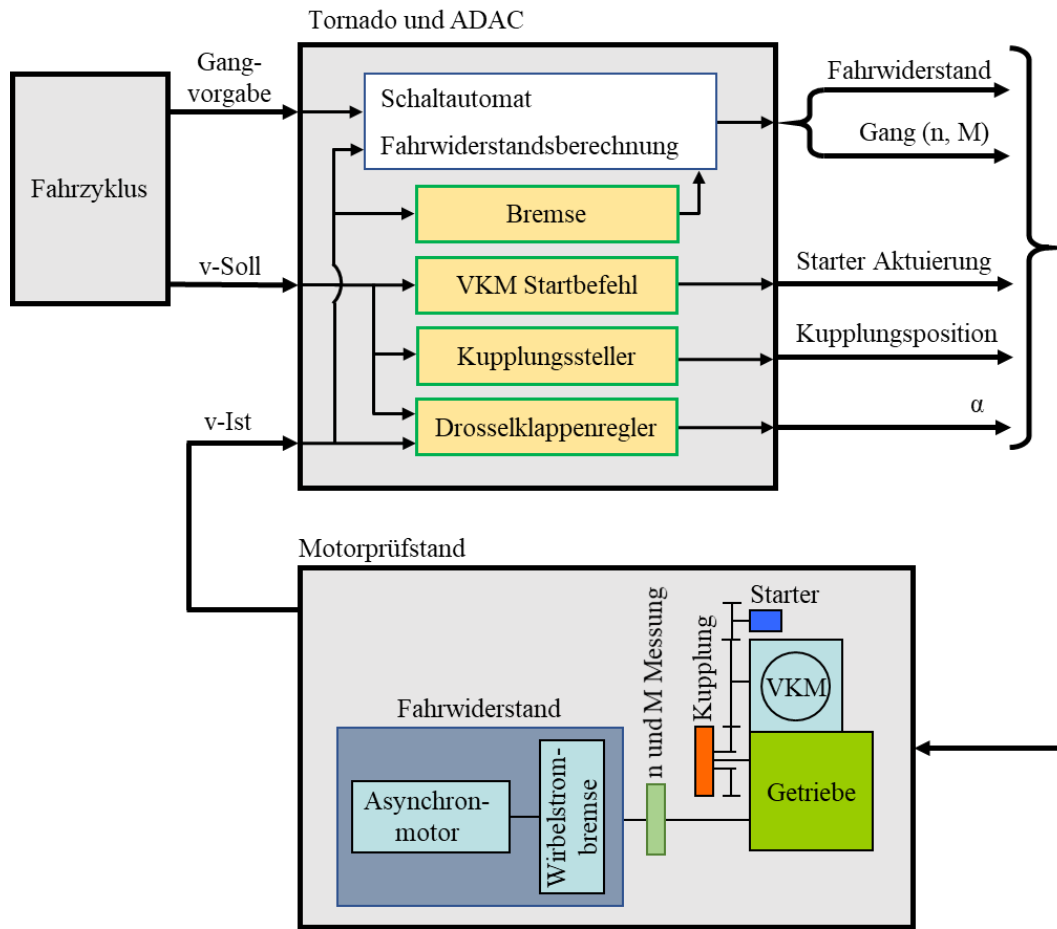


Abbildung 3-6: Prüfstand Systemarchitektur Ziel

Dabei ist für die Vollautomatisierung des Prüfstands zu achten, dass die Einstellungen für den automatisierten Prüfzyklus sich so einfach wie möglich gestalten, um die Funktion auch in zukünftigen Projekten so komfortabel wie mögliche verwenden zu können.

Durch transiente Fahrzyklen treten Zustände auf, bei der die Schubabschaltung bzw. das Motorbremsmoment zur Reduktion der Geschwindigkeit nicht ausreichen. Daher ist eine emulierte „mechanische“ Bremse notwendig. Diese wurde über eine Steigungsvorgabe realisiert, die damit den Fahrwiderstand erhöht. Die VKM wird über den VKM Startbefehl am Anfang des Prüfzyklus gestartet. Der Kupplungssteller ist dafür verantwortlich, dass die Kupplungsposition beim Ein-/Auskuppeln definiert abläuft. Der Drosselklappenregler ist dafür zuständig, dass die vorgegebene Geschwindigkeit eingehalten wird.

4 Bestehender Automatisierungsgrad

In diesem Kapitel wird der Startzustand des Motorprüfstands für die Automatisierung erläutert. Dabei wird diese Konfiguration von Kristl und Seibt unter dem Namen *Dynamischer Modus Parameter* in die Prüfstandssteuerung miteinbezogen. Die Erläuterung dieser Funktionen wurde in [1] noch nicht behandelt, deshalb erfolgt dies nun in diesem Kapitel.

Der Prüfstand am P8a besitzt eine Tornado Steuerung, die den Prüfstand regelt. Diese wird mit der „Road-Load-Simulation“ (RLS) aktiviert. Dabei gibt es die Regelung RLS α (alpha) und RLS v . Mit der RLS Regelungsstrategie emuliert der Tornado Prüfstand eine Fahrwiderstandskurve inklusive Beschleunigungswiderstand bezogen auf die Masse des simulierten Fahrzeuges. Zusätzlich kann der Prüfstand zwischen den Gängen schalten. Dies funktioniert, indem die Drehzahl und das Moment angepasst werden.

Wenn RLS α aktiviert ist, dann ist die Drosselklappe die veränderbare Größe, wobei bei RLS v die Geschwindigkeit die veränderbare Größe ist.

Die Parametrisierung erfolgt in der „Parameter Dynamische Regelungsarten“ – Benutzeroberfläche im Tornado RT.

Parameter Dynamische Regelungsarten			
Gang Parameter			
	Übersetzung	Wirkungsgrad [%]	Trägheitsmoment [kgm ²]
Gang 1	4.7820	100.00	0.0000
Gang 2	3.4000	100.00	0.0000
Gang 3	2.6830	100.00	0.0000
Gang 4	2.2810	100.00	0.0000
Gang 5	2.0240	100.00	0.0000
Gang 6	1.8650	100.00	0.0000
Gang 7	0.0000	100.00	0.0000
Differential	2.3529	100.00	0.0000
Physikalisch	1.8650	104.00	
Automatik Schaltpunkte			
Rauf (Beschl.)	Runter (Beschl.)	Rauf (Verz.)	Runter (Verz.)
3500		3500	
3500	1500	3500	1500
3500	1500	3500	1500
3500	1500	3500	1500
3500	1500	3500	1500
3500	1500	3500	1500
	1500		0
Fahrzeug und Strasse			
Fahrzeugmasse	289.0	kg	
Fahrwiderstandskoeffizient R0	25.500	N	
Fahrwiderstandskoeffizient R1	0.000	N/(m/s)	
Fahrwiderstandskoeffizient R2	0.316	N/(m/s) ²	
Dynamischer Reifendurchmesser	0.614	m	
Äquivalente Radmasse	1.0	kg	
Fahrbahnsteigung	0.0	%	
Zugkraft Fahrzeug	0.0	N	
Wegstrecke Fahrzeug	285.7	km	
Beschleunigung Fahrzeug	0.0	m/s ²	
Geschwindigkeit Fahrzeug	0.0	km/h	
Schalten		Bremsen	
Schaltzeit Einkuppeln	0,3 s	Modus	kein Bremsen
Schaltzeit Auskuppeln	0,3 s	Abweichung	4,0 (m/s)/(m/s ²)
Max. Kupplungsmoment	140,0 Nm	Faktor	500,0 N/(m/s)
Min. Kupplungsmoment	-140,0 Nm	Max. Bremskraft	2000,0 N
Anzahl Gänge	6 -		
Gang Soll	5	Loadline Parameter	
Gang Ist	5		
Schalstatus	0		
Schaltmodus	Manuell		

Abbildung 4-1: Parameter Dynamische Regelungsarten Übersicht

Die einzelnen Felder werden im Folgenden erklärt.

4.1 Gang Parameter

Die einzelnen Gänge inklusive Differential und der vorhandenen physikalischen Übersetzung sind einzutragen, inklusive dem Wirkungsgrad und dem Trägheitsmoment.

Übersetzung:

Gang 1-7: Die jeweilige Übersetzung der einzelnen Gänge multipliziert mit der Primärübersetzung ist einzufügen.

Z.B.: mit der Primärübersetzung = 68/35 und erster Gangübersetzung = 32/13 ergibt in (4-1) die Übersetzung für den ersten Gang.

$$\text{Gang 1} = i_{\text{primär}} * i_{\text{erster Gang}} = 4,782 \quad (4-1)$$

Differential: Die Übersetzung des Differentials des Fahrzeugs ist einzusetzen. Wenn das Fahrzeug ein Motorrad ist, dann ist die Übersetzung des Kettentriebs (bzw. vom Riemtrieb) einzusetzen.

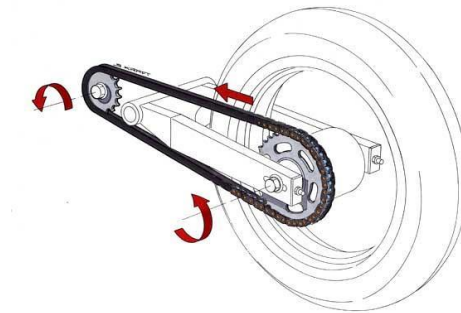


Abbildung 4-2: Sekundärübersetzung Motorrad [18]

Physikalisch: Die Gesamtübersetzung vom Verbrennungsmotor bis zur Bremse ist einzusetzen. Das heißt, dass der in „echt“ eingelegte Gang hier zu berücksichtigen ist, wobei das Differential nicht hineingerechnet wird.

Tornado verwendet die einzelnen Werte in der Gang Parameter Tabelle, um die Hinterraddrehzahl bzw. die Fahrzeuggeschwindigkeit zu berechnen. Die Fahrzeuggeschwindigkeit ist weiters für die Berechnung des Fahrwiderstands notwendig.

Wirkungsgrad: Jeder Getriebeübersetzung kann ein Wirkungsgrad zugeordnet werden, falls die einzelnen Übersetzungswirkungsgrade gegeben sind, damit bei jedem Gang auf das effektive Motormoment zurückgerechnet werden kann.

Der Wirkungsgrad wird >100% eingesetzt, damit das effektive Motormoment mit der Formel (4-2) berechnet werden kann, da das gemessene Moment der Bremse kleiner ist als das des Motors.

$$M_{\text{Motor}} = \text{Wirkungsgrad} * M_{\text{Bremse}} \quad (4-2)$$

Trägheitsmoment: Für jede Getriebeübersetzung kann ein Trägheitsmoment eingegeben werden, um die einzelnen Trägheiten für die Beschleunigungsvorgänge zu berücksichtigen. Die unterschiedlichen Trägheitsmomente der einzelnen Gänge werden mit der reduzierten Masse im Tornado berücksichtigt.

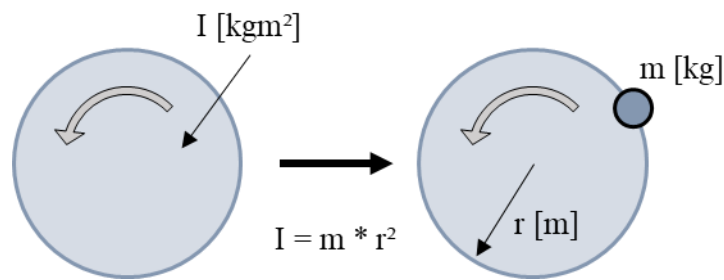


Abbildung 4-3: Berechnung Massenträgheitsmoment
(Adaptiert und übernommen von KS Engineers Quelle)

4.2 Schalten

Die Schaltzeiten für das Einkuppeln beziehungsweise für das Auskuppeln werden eingesetzt.

Dabei ist zu beachten, dass der Prüfstand bei dem Kupplungsmoment limitiert ist. Das maximale Kupplungsmoment kann das maximale Bremsmoment vom Prüfstand bei der gegebenen Drehzahl nicht überschreiten. Erfahrungswerte zeigten, dass eine kürzere Zeit für das Aus- und Einkuppeln als 0,3s bei einer Verbrennungskraftmaschine mit 800ccm nicht sinnvoll ist, da die Bremse das Schaltverhalten nicht schneller umsetzen kann.

In das „Anzahl Gänge“-Feld werden die Anzahl der Gänge eingestellt. Wenn dort, z.B.: 6 eingetragen wird, dann verwendet die Schaltautomatik nur die ersten 6 Gänge.

„Gang Soll“ zeigt die aktuelle notwendige Getriebeübersetzung an, wobei „Gang Ist“ den aktuellen eingelegten emulierten Gang anzeigt.

Für den Schaltmodus kann entweder Manuell oder Automatik gewählt werden. Bei Manuell wird eine Tabelle mit den vorgegebenen Gängen verwendet oder, wenn diese die Gänge nicht verwendet, kann manuell am Prüfstand auf der Multicontrolunit (MCU) geschaltet werden. Ebenfalls können die Schaltpunkte in den Automatik Modus gesetzt werden (siehe Punkt Automatik Schaltpunkte).

Der Vorgang für das manuelle Schalten kann im Tornado Designer unter Schaltverhalten => Standard definiert werden. Hier die aktuelle Konfiguration.

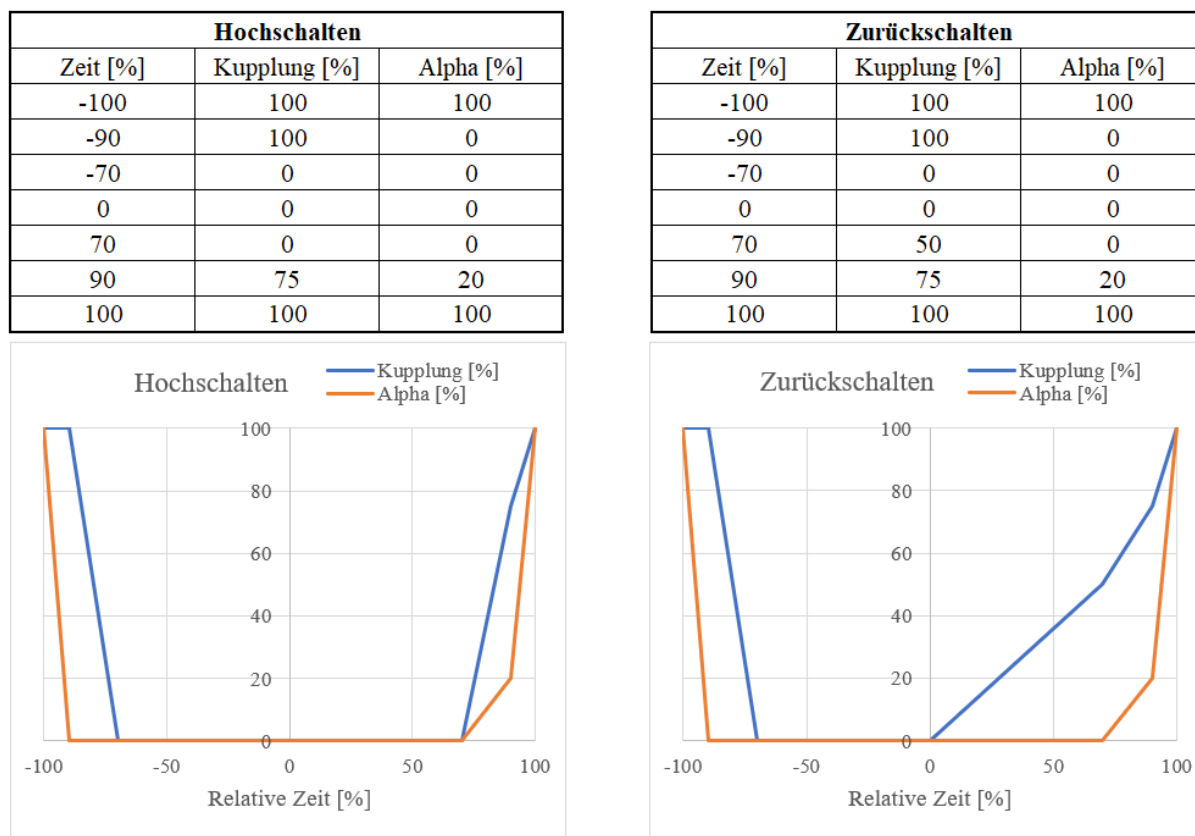


Abbildung 4-4: Tornado Schaltvorgang

Die „Schaltzeit Auskuppeln“ entspricht der relativen Zeit von -100 bis 0%. Die „Schaltzeit Einkuppeln“ entspricht der relativen Zeit von 0 bis 100%. Beim Schalten sind die Drosselklappe und das Kupplungsmoment vorgegeben, wobei diese beiden Regelungsgrößen vom Prüfstand umgesetzt werden.

4.3 Automatik Schaltpunkte

Für jeden einzelnen Gang kann hier die Drehzahl der einzelnen Schaltpunkte definiert werden. Hier wird unterschieden, ob das Fahrzeug bzw. der Motor aktuell beschleunigt (Beschl.) oder verzögert (Verz.). Das heißt, dass der Prüfstand hier zwischen den Schaltpunkten im Beschleunigungsvorgang bzw. im Verzögerungsvorgang unterscheidet.

Wenn, z.B.: der Motor beschleunigt und die Motordrehzahl die Schaltpunkt-Drehzahl in „Rauf (Beschl.)“ überschreitet, dann „schaltet“ die Prüfstandsbremse auf den nächsten höheren Gang.

4.4 Bremsen

Die folgende Beschreibung der einzelnen Modi wurde von KS Engineers zur Verfügung gestellt. Dabei ist anzumerken, dass die „fehlenden“ Modi 4, 5 und 6 nicht zur Verfügung gestellt wurden:

- Modus 0: kein Bremsen
- Modus 1: Differenz v_{Soll} zu $v_{Ist} >$ Abweichungswert. Der Stellwinkel Alpha wird auf 0% gesetzt.
- Modus 2: Differenz v_{Soll} zu $v_{Ist} >$ Abweichungswert, aktive Bremsung. Der Stellwinkel Alpha wird auf 0% gesetzt, zusätzlich erfolgt eine Bremsung (Bremskraft abhängig von Abweichung v_{Soll} zu v_{Ist} relativ durch den Bremsfaktor).

- Modus 3: Differenz v_{Soll} zu $v_{Ist} >$ Abweichungswert, aktive Bremsung + F_{max} . Der Stellwinkel Alpha wird auf 0% gesetzt, zusätzlich erfolgt eine Bremsung mit maximaler Bremskraft.
- Modus 7: Differenz v_{Soll} zu $v_{Ist} >$ Abweichungswert, aktive Bremsung. Der Stellwinkel Alpha wird *nicht* auf 0% gesetzt. Es erfolgt eine Bremsung (Bremskraft abhängig von Abweichung v_{Soll} zu v_{Ist} relativ durch den Bremsfaktor).
- Modus 8: Differenz v_{Soll} zu $v_{Ist} >$ Abweichungswert, aktive Bremsung + F_{max} . Der Stellwinkel Alpha wird *nicht* auf 0% gesetzt. Es erfolgt eine Bremsung mit maximaler Bremskraft.

Der Abweichungswert gibt an, ab welcher Grenzwertüberschreitung die Bremse aktiviert wird. Die „aktive Bremsung“ wird, wie folgt, berechnet, wenn die Bremse aktiviert wurde.

$$Bremskraft = (v_{Soll} - v_{Ist}) * Faktor \quad [N] \quad (4-3)$$

Hier muss angemerkt werden, dass diese Bremsenmodi nicht eingesetzt werden konnten. Daher wurde eine emulierte „mechanische Bremse“ als Befehlsprogramm implementiert.

4.5 Fahrzeug und Straße

Fahrzeug und Strasse		
Fahrzeugmasse	289,0	kg
Fahrwiderstandskoeffizient R0	25,500	N
Fahrwiderstandskoeffizient R1	0,000	N/(m/s)
Fahrwiderstandskoeffizient R2	0,316	N/(m/s) ²
Dynamischer Reifendurchmesser	0,614	m
Äquivalente Radmasse	1,0	kg
Fahrbahnsteigung	0,0	%
Zugkraft Fahrzeug	0,0	N
Wegstrecke Fahrzeug	285,7	km
Beschleunigung Fahrzeug	0,0	m/s ²
Geschwindigkeit Fahrzeug	0,0	km/h

Abbildung 4-5: Fahrzeug und Straße Parametrisierung

Für die Fahrzeugmasse wird die Masse des Fahrzeuges plus des Fahrers eingetragen (beim WMTC werden hier 75kg zum Fahrzeug addiert).

Der Fahrwiderstand wird mit der Formel (4-4) berechnet:

$$F_{Widerstand} = R_0 + R_1 * v + R_2 * v^2 + m * \frac{dv}{dt} + m * g * \sin \alpha \quad [N] \quad (4-4)$$

Für die Masse m wird von Tornado die „Fahrzeugmasse“ und die „Äquivalente Radmasse“ addiert (4-5).

$$m = m_{Fahrzeug} + m_{Äquivalente Radmasse} \quad (4-5)$$

Die äquivalente Radmasse wird mit der zur Verfügung gestellten Formel von KS berechnet. Das heißt die Äquivalente Radmasse = Massenträgheitsmoment beider Räder (Quelle: KS Engineers).

$$\text{Äquivalente Radmasse} = (m_{\text{Vorderrad}} + m_{\text{Hinterrad}}) * r^2 \quad (4-6)$$

Der dynamische Reifendurchmesser definiert das Rad, um die Geschwindigkeit des Fahrzeuges von der Raddrehzahl zu berechnen.

Ebenfalls kann eine „Fahrbahnsteigung“ als zusätzliche oder reduzierte Last während des Prüfzyklus definiert werden.

Die Felder „Wegstrecke Fahrzeug“, „Beschleunigung Fahrzeug“ sowie die „Geschwindigkeit Fahrzeug“ sind Ist-Werte. Mit der Umrechnung auf die Fahrzeuggeschwindigkeit wird der Fahrwiderstand in (4-4) berechnet und der Prüfstandsbremse vorgegeben.

Für einen vollautomatisierten Motorprüfstand sind einige Modifikationen notwendig, die auf der Ausgangsbasis aufbauen.

5 Modifizierungen und Implementierungen

Für den vollautomatisierten Betrieb des Motorprüfstandes sind einige Erweiterungen und Modifikationen am Prüfstand notwendig. Diese umfassen Hardware- und Softwareanpassungen, wobei die Anpassungen im Wesentlichen die folgenden Teilsysteme beinhalten: die Kupplungsaktuatorik für den ausgekuppelten Leerlaufzustand und das Anfahren, der Geschwindigkeitsregler und das emulierte mechanische Bremsen.

5.1 Kupplungsaktuatorik

Die meisten transienten Fahrzyklen haben gewisse Fahrzeug-Stillstandszeiten, bei welchen die Geschwindigkeit Null ist, der Motor jedoch im Leerlauf verweilt. Für diesen Zustand ist ein Kupplungsaktuator notwendig, der die mechanische Kupplung zieht und damit die Verbrennungskraftmaschine vom Antriebsstrang trennt. Ebenfalls ist ein Ablauf notwendig, der ein ruckfreies Einkuppeln bei Ansteigen der vorgegebenen Geschwindigkeit einleitet.

Für diesen Kupplungsaktuator wurde ein System entwickelt, das flexibel für jede mechanische Kupplung eines Motorrades aufgebaut werden kann (Abbildung 5-1).

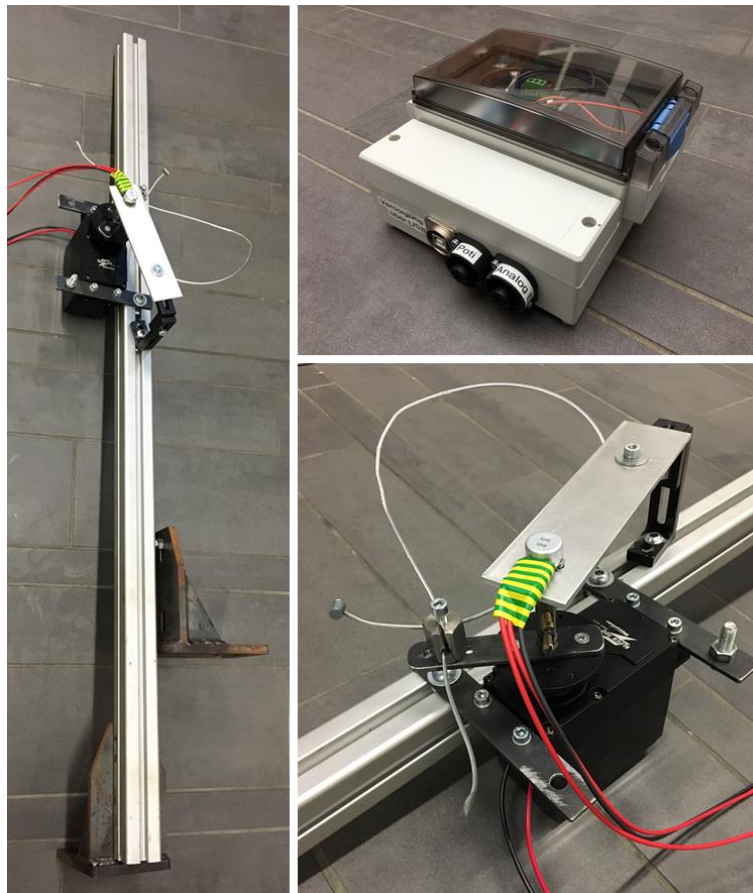


Abbildung 5-1: Kupplungsaktuator: Profil und Montagewinkel (links), Elektronikbox (rechts oben), Elektromotor mit Potentiometer (rechts unten)

Das Profil wird direkt am Motorprüfstand bei passenden Befestigungsbohrungen angeschraubt. Der Elektromotor ist auf ein Profil fixiert und eine Halterung positioniert den Drehpotentiometer in der Drehachse des Elektromotors. Am Hebelarm des Elektromotors ist eine geschlitzte Mutter angebracht, mit der ein Kupplungsseil geklemmt werden kann.

5.1.1 Verwendete Komponenten

Für das Kupplungsaktuatorsystems wird ein Mikrocontroller, ein solid-state Motorcontroller und ein DC Elektromotor mit Positionsrückmeldung verwendet. Zusätzlich wird noch eine Spannungsversorgung mit einstellbarem Maximalstrom von 3,5A bei 12V benötigt.

Mikrocontroller - Arduino Uno

Der Arduino Uno ist ein kostengünstiges Entwickler-Board, das für Aufgaben von Mess-, Steuer- und Regelungsaufgaben geeignet ist.

Dieses Board besitzt einen Microcontroller, sowie digitale und analoge Ein- und Ausgänge. Von den 14 digitalen Ein- und Ausgängen sind sechs als PWM Ausgang verwendbar. Dieses Board hat zusätzlich sechs analoge Eingänge mit einer 10 bit Auflösung. [19]



Abbildung 5-2: Arduino Uno [19]

Das Microcontroller Board eignet sich für das Ansteuern des Motorcontrollers und zum Einlesen der Position des Elektromotors mittels einem Drehpotentiometers.

Für das Betreiben des Arduinos als externes System wird ebenfalls ein Transformator mit 12V und mindestens 1,5A benötigt.

Motorcontroller

Der Motorcontroller fungiert als Schnittstelle zwischen der Steuer- und der Leistungselektronik. Allgemein erhält der Motorcontroller ein Steuersignal von dem Arduino Uno und schaltet darauf die Leistung abhängig von dem PWM-Steuersignal für den Elektromotor frei.

Ausgewählt wurde ein kostengünstiger, jedoch leistungsstarker solid-state Motorcontroller mit der Typenbezeichnung IBT-2 (Abbildung 5-3 und Tabelle 5-1).

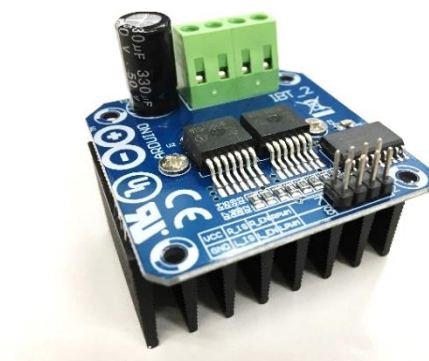


Abbildung 5-3: Motorcontroller IBT-2

Halbbrücken	BTS7960
Eingangsspannung	6 – 27V
Maximalstrom	43A
Kontrollmodus	PWM-Signal oder Analogsignal
Abmessungen	5x5x4cm

Tabelle 5-1: Motorcontroller Spezifikationen

Hier ist auch festzuhalten, dass der Motorcontroller einen wesentlich stärkeren Elektromotor ansteuern kann, da dieser „nur“ bei ca. 1/10 der Nennleistung betrieben wird. Daher ist anzunehmen, dass dieses System auch für das Betätigen einer PKW-Kupplung geeignet ist.

Allgemein ist dies ein voll integrierter Motor-Treiber. Die Hauptbestandteile dieses Motorcontrollers sind zwei Halbbrücken mit jeweils einem p-Kanal und einem n-Kanal MOSFET und einem integrierten Controller (Abbildung 5-4).

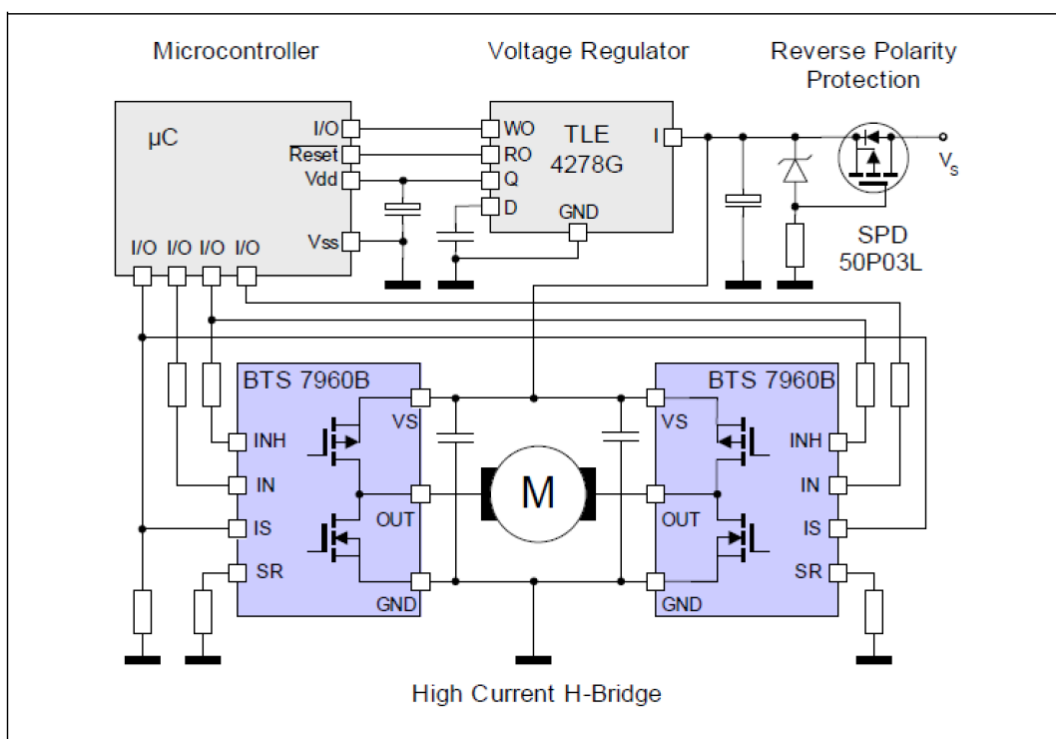


Abbildung 5-4: Schematische Darstellung Motorcontroller [20]

Der Elektromotor wird zwischen den beiden Halbbrücken angeschlossen. Die beiden MOSFETs beider Halbbrücken werden vom Microcontroller so angesteuert, sodass entweder der MOSFET links oben und rechts unten durchschaltet (Vorwärtsdrehung) oder rechts oben und links unten durchschaltet (Rückwärtsdrehung). Dabei besitzt der Motorcontroller ebenfalls einen Polschutz bzw. einen Freilauf.

Elektromotor

Der Elektromotor „Super300 High Torque Metal Servo“ (Abbildung 5-5) wurde zerlegt und die integrierte Schaltung überbrückt, da diese der genauen Regelung nicht gerecht wurde. Dieser Servomotor besitzt eine Positionierbarkeit und Wiederholbarkeit in der Größenordnung von $\pm 20^\circ$. Jedoch ist für die präzise Stellung des Elektromotors eine Positioniergenauigkeit von mindestens $\pm 5^\circ$ notwendig. Daher wurde dieser Servomotor als reiner DC-Elektromotor ohne integrierter Schaltung betrieben, wobei die Position direkt von der Rotationsachse mit einem Drehpotentiometer abgegriffen wird. Die Genauigkeit der Positionsrückmeldung des Drehpotentiometers ist ausreichend genau für diesen Anwendungsfall.

Der Elektromotor hat ein maximales Moment von 300kgcm und kann mit einer Spannung von 12 bis 24V betrieben werden [21].

Die in 5.1.1 beschriebene Spannungsversorgung mit 3,5A bei 12V ist ebenfalls eine Sicherung, da bei einem Fehler der Elektromotor keine Beschädigungen anrichten kann. Der Elektromotor kann bei dieser Maximalleistung die mechanische Kupplung ausreichen schnell und genau betätigen, jedoch kann bei Fehlverhalten dieses Teilsystems der Elektromotor nicht höher bestromt werden.



Abbildung 5-5: Elektromotor [21]

5.1.2 Leitung- und Anschlussplan

Das Fritzing Diagramm (Abbildung 5-6) zeigt die Anschlüsse der Steuer- und Leistungselektronik schematisch.

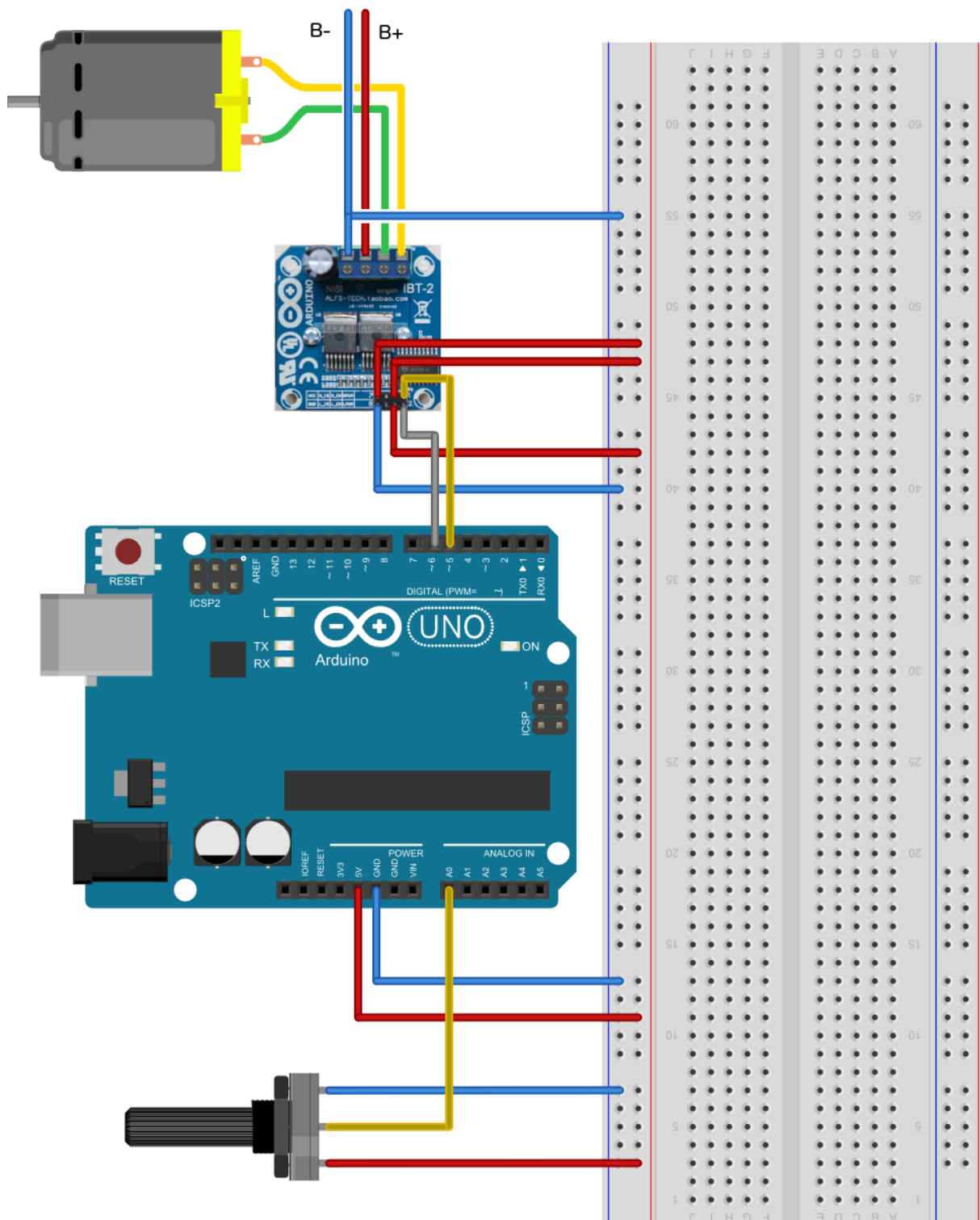


Abbildung 5-6: Schematischer Anschlussplan [22]

Der schematische Anschlussplan wird exakt so verwendet, mit der Ausnahme, dass das Drehpotentiometer direkt an der Drehachse des Elektromotors verbunden ist.

Der Motorcontroller hat vier Leistungsanschlüsse (B+ und B- für die Versorgungsspannung, M+ und M- für den Elektromotoranschluss) und acht Steueranschlüsse.



Abbildung 5-7: Steueranschlüsse Motorcontroller

Diese Steueranschlüsse 1 bis 8 (Abbildung 5-7) werden wie folgt an den Mikrocontroller angeschlossen:

VCC an eine 5V Spannungsversorgung. GND an den Groundanschluss. R_EN (right enabled) und L_EN (left enabled) werden kurzgeschlossen und mit 5V dauerhaft versorgt. L_PWM und R_PWM werden jeweils an einen digitalen PWM-Ausgang am Mikrocontroller angeschlossen. Die Anschlüsse R_IS und L_IS können für einen Stromalarmausgang verwendet werden. [22]

Die Position des Elektromotors wird über einen analogen Ausgang an der Sensorbox über Tornado vorgegeben. Der Arduino Uno greift dieses über einen analogen Eingang ab und verwendet dieses für die Positionsregelung des Elektromotors.

5.1.3 Steuerung des Kupplungsaktuators

Die Vorgabe der Position wird von Tornado vorgegeben. Diese Sollposition wird als analoges Eingangssignal mit der Variable „Soll-Poti“ mit dem „Ist-Poti“-Wert vom Drehpotentiometer verglichen. Wenn dieser Vergleich positiv ist und einen Grenzwert überschreitet, dann wird die Kupplung schnellstmöglich geöffnet. Falls der Vergleich negativ ist und einen Grenzwert unterschreitet, dann wird ein PWM-Signal abhängig von der Größe der Differenz an den Motorcontroller übertragen. Dieses PWM-Signal führt dazu, dass ein ruckfreies Einkuppeln ermöglicht wird. Der Quellcode für den Arduino Uno kann in Anhang 9.4 eingesehen bzw. direkt kopiert und verwendet werden.

Der Einkuppelvorgang ist in der Tornado Prüfstandssteuerung vorgegeben. Beim Auskuppeln wird direkt der Auskuppelendwert vorgegeben. Beim Einkuppelvorgang durchläuft die Vorgabe der Position ein vorgegebenes Einkuppelprofil (Abbildung 5-8) in einem Befehlsprogramm. Während des Einkuppelvorgangs wird ebenfalls die Drosselklappenstellung vorgegeben und am Ende des Einkuppelvorgangs wird die Drosselklappenstellung wieder an die Geschwindigkeitsregelung übergeben.

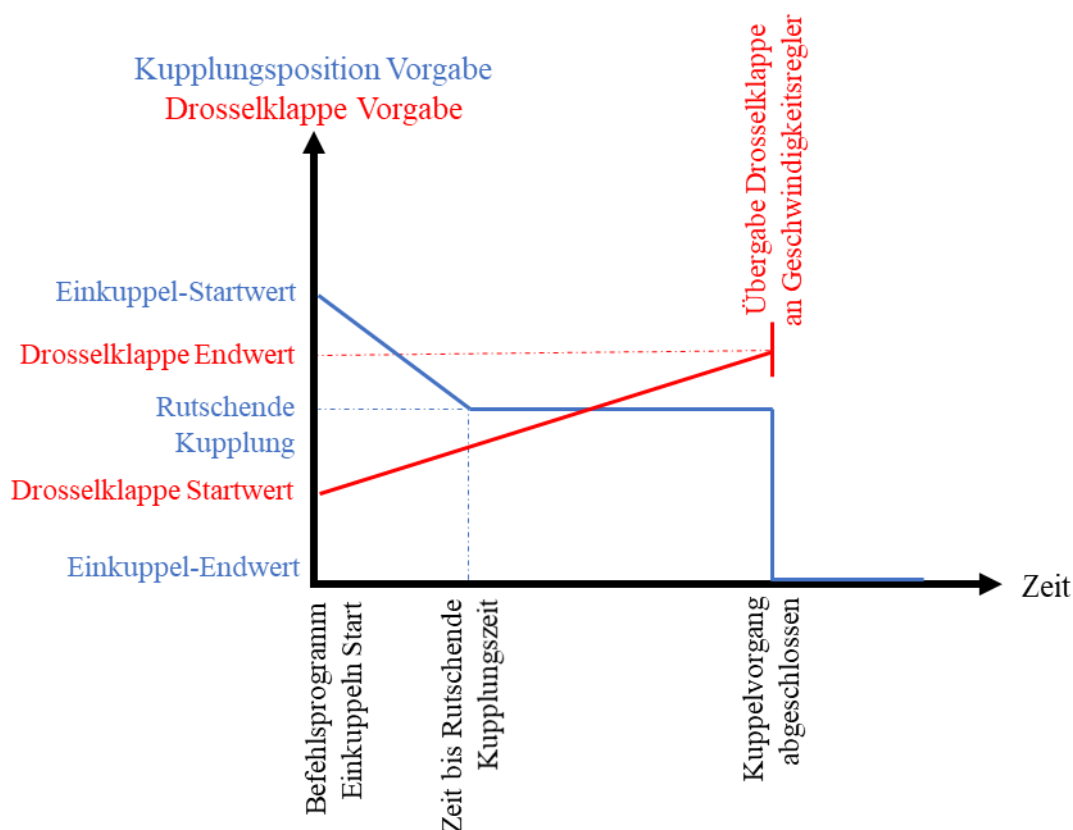


Abbildung 5-8: Vorgabe Einkuppelvorgang

Beim Abschluss des Befehlsprogramms für das Einkuppeln wird die Fahrgeschwindigkeitsregelung einem PID-Regler übergeben.

5.2 Geschwindigkeitsregelung

Allgemein vergleicht der Geschwindigkeitsregelkreis die vorgegebene Geschwindigkeit v_{soll} mit der aktuellen Fahrzeuggeschwindigkeit v_{lst} und berechnet daraus eine Stellgröße für die Drosselklappenposition, um die aktuelle Geschwindigkeit des Fahrzeuges der Vorgabe anzupassen.

Der Regelkreis besteht aus der Regeleinrichtung, der Stelleinrichtung und der Regelstrecke und ist schematisch in Abbildung 5-9 dargestellt.

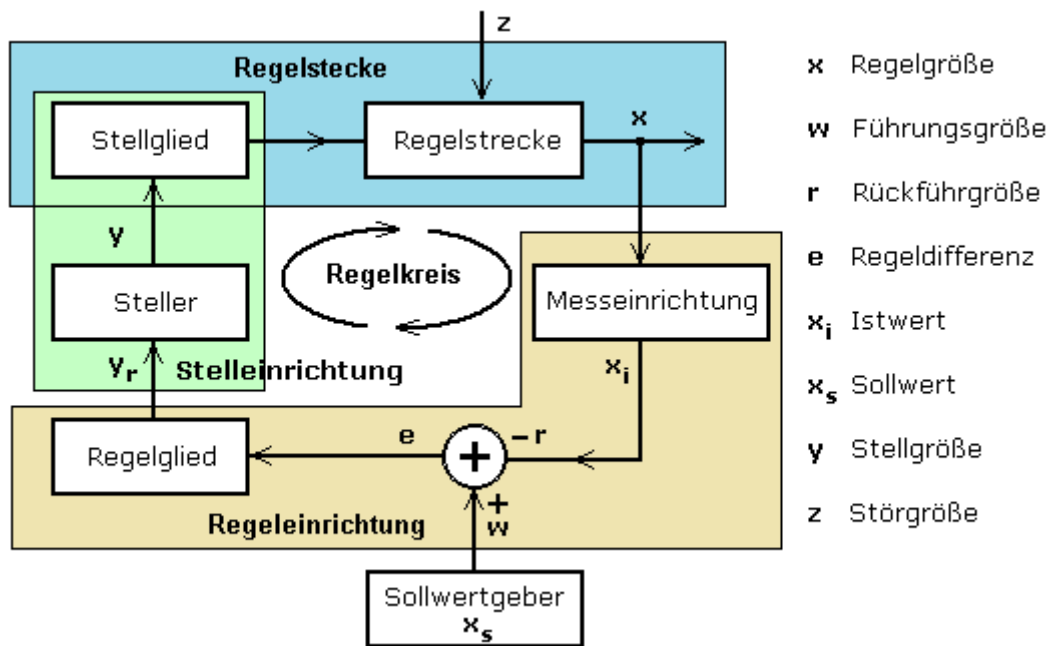


Abbildung 5-9: Schematische Darstellung Regelkreis nach DIN 19226 [23]

Die vorgegebene Geschwindigkeit v_{Soll} wird mit einer Parametertabelle als zeitlich diskreter Wert vorgegeben und der Regeleinrichtung zugeführt. Die Messeinrichtung misst die aktuelle Drehzahl an der Bremse. Danach wird die Fahrzeuggeschwindigkeit v_{Ist} mit der gemessenen Drehzahl, dem Raddurchmesser und dem aktuell eingelegten Gang berechnet. Die Regeldifferenz e bildet sich somit aus der Differenz von $v_{Soll} - v_{Ist}$ und wird dem Regelglied, einem PID-Regler zugeführt. Der PID-Regler berechnet dann die Stellgröße in zeitdiskreten Schritten.

Anhand des folgenden Beispiels soll die Berechnung der Stellgröße erklärt werden. Die PID-Werte und die Regeldifferenz e werden in (5-1) beispielhaft angegeben. Die Formel (5-2) wird von der Formel (2-3) an dieses Beispiel angepasst und es wird angenommen, dass die Regelabweichung zwei Sekunden lang auftritt. Nach dem Einsetzen der Werte, Integrieren und Differenzieren ergibt sich eine Stellgröße y_r von 8% (5-3).

$$k_p = 2, \quad k_I = 1, \quad k_D = 0,1, \quad e = 2 \frac{km}{h} = \text{konstant} \quad (5-1)$$

$$y_r = k_p * e + k_I * \int_0^2 e * dt + k_D * \frac{de}{dt} \quad (5-2)$$

$$y_r = 2 * 2 + 1 * \int_0^2 2 * dt + 0,1 * 0 = 4 + 4 + 0 = 8\% \quad (5-3)$$

Diese Stellgröße y_r , welcher der Drosselklappenstellung-Vorgabe entspricht, wird an die Stelleinrichtung weitergegeben. Diese Stelleinrichtung besteht aus einem Servocontroller „Servomatic“ und einem Servomotor (Abbildung 5-10). Dieser Controller vergleicht ebenfalls die aktuelle Drosselklappenstellung mit der Vorgabe und zieht an dem Gasseil, das mit der Drosselklappe verbunden ist, um die Regelabweichung kleinstmöglich zu halten.

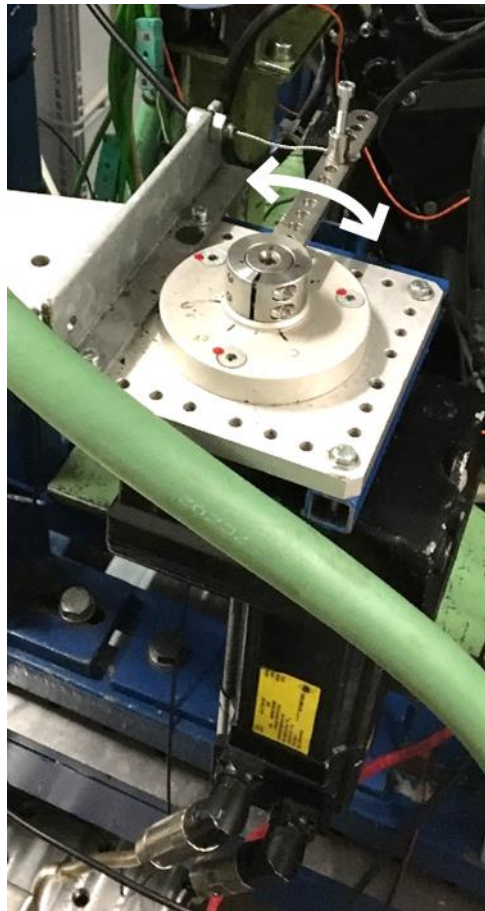


Abbildung 5-10: Servomotor

Somit wird die Regelstrecke, der Verbrennungsmotor, mit der Drosselklappe beschleunigt, konstant gehalten oder verlangsamt und somit die aktuelle Geschwindigkeit beeinflusst. Somit ist der Regelkreis geschlossen. Ein wesentlicher Punkt dieses Regelkreises ist die Parametrisierung der PID-Werte. Diese wird in der Schritt-für-Schritt Anleitung in Anhang 9.1 im Detail erklärt.

5.3 Emuliertes mechanisches Bremsen

Bei manchen Fahrzyklen tritt eine rasche Verzögerung auf, bei der ein Stellen der Drosselklappe auf 0% nicht ausreicht, um das Fahrzeug im Schubbetrieb zu bremsen. Daher ist eine Implementierung einer Bremsfunktion notwendig, die das Fahrzeug bei Überschreiten eines Grenzwertes mitbremst.

Die emulierte mechanische Bremse wurde mittels eines Befehlsprogramms implementiert. Das Befehlsprogramm aktiviert sich, wenn die aktuelle Geschwindigkeit einen Grenzwert überschreitet und die Drosselklappe beinahe Null ist. Nach Überschreitung des Grenzwertes wird eine Steigung vorgegeben, die damit in die Berechnung des Fahrwiderstands (4-4) einfließt und das Fahrzeug somit zusätzlich abbremst. Nach dem Unterschreiten des Grenzwertes wird die Steigung wieder auf den Ausgangswert (aktuell 0%) zurückgesetzt.

6 Verifizierung vollautomatisierter Prüfstand

Im Folgenden wird ein Test im teilautomatisierten Zustand (manueller Gasgriff, Kupplung und Bremsen) mit einem vollautomatisierten Prüfzyklus verglichen, danach die Wiederholbarkeit untersucht und mit einer Motorrad-Rollenmessung verglichen.

Bevor die beiden Fahrzyklen im Detail verglichen werden, soll angemerkt werden, dass die Fahrzyklen auf einem Verbrennungsmotor im Prototypen-Zustand durchgeführt wurden. Das heißt, dass die Applizierung der Kennfelder noch nicht optimiert sind bzw. die ECU in einem anderen Setup arbeitet wie die Serien-ECU und dadurch der Verbrennungsprozess zum Teil nicht gleich abläuft. Aufgrund der Verwendung eines konstant parametrisierten Geschwindigkeitsregler ergeben sich hierbei größere Regelabweichungen, die bei einem „optimal“ applizierten Motor nicht auftreten würden. Daher wird angenommen, dass die Geschwindigkeitsabweichungen für einen serienapplizierten Motor niedriger sind. Diese Abweichungen sind ebenfalls der Grund wieso eine Vollautomatisierung sinnvoll ist, da der Prüfstandstechniker Abweichungen bzw. fehlerhafte Systemzustände identifizieren kann, die noch optimiert werden müssen. Zusätzlich hat der Applikateur während eines Prüfzyklusses mehr Zeit, den Verbrennungsmotor zu beobachten und die Kennfelder zu applizieren.

6.1 Gütegrad Teilautomatisierung und Vollautomatisierung

Zuerst wird die Geschwindigkeitsabweichung zwischen v_{Soll} und v_{Ist} betrachtet. Der WMTC gibt eine obere und untere Geschwindigkeitsgrenze vor. Diese Limits sind bei dem WMTC für L3e Fahrzeuge $\pm 3,2 km/h$ bei jeweils $\pm 1s$ ausgehend von der Soll-Geschwindigkeit (Abbildung 6-1). Der Zyklus ist in drei gleich langen Teilen mit je 600 Sekunden aufgeteilt und der Gütegrad wird berechnet.

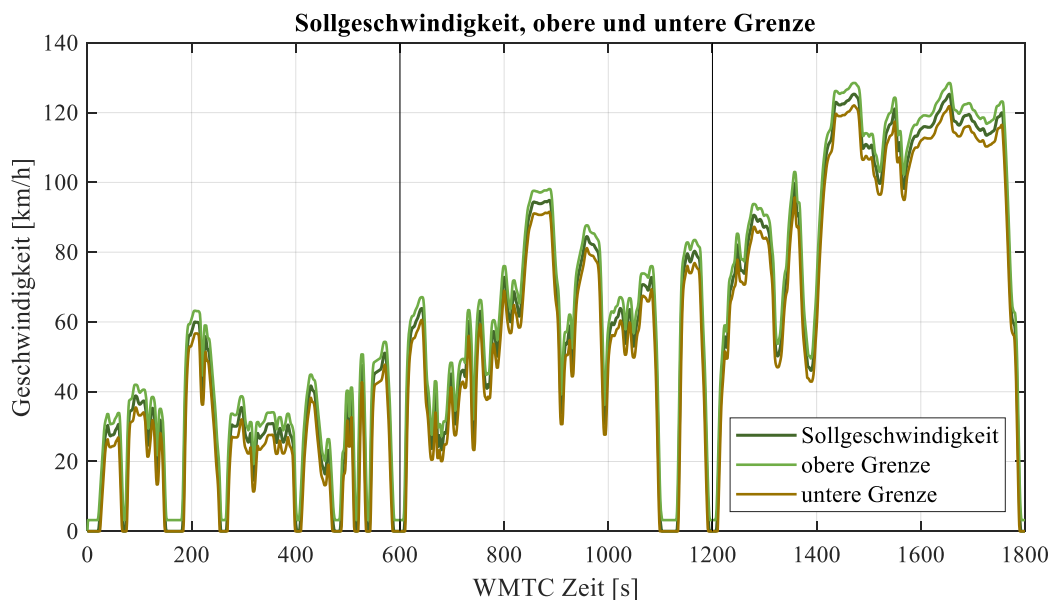


Abbildung 6-1: WMTC Geschwindigkeiten

Es wird ein Gütegrad δ_v eingeführt, der prozentuell darstellt, wie „lange“ die aktuelle Geschwindigkeit zwischen der oberen und der unteren Grenze ist. Dies wird für jeden Messpunkt durchgeführt und falls v_{Ist} innerhalb dieser liegt, dann entspricht dieser Messpunkt 100% und wenn v_{Ist} außerhalb ist, dann entspricht dies 0%. Allgemein ist ein Wert von 100% anzustreben.

In Tabelle 6-1 ist ersichtlich, dass im vollautomatisierten Betrieb die Geschwindigkeit vor allem in höheren Lastpunkten leichter eingehalten werden kann und dass der Geschwindigkeitsregler

vor allem im ersten Teil Schwierigkeiten hat, der Geschwindigkeit zu folgen. Auch im teilautomatisierten Betrieb ist es schwieriger in den niedrigen Lastpunkten innerhalb der Toleranz zu bleiben.

Test	Teil 1 [%]	Teil 2 [%]	Teil 3 [%]	Gesamt [%]
Teilautomatisiert	88,6	91,5	98,1	92,7
Vollautomatisiert 1	81,8	92,8	98,0	90,9
Vollautomatisiert 2	83,9	91,5	98,4	91,3

Tabelle 6-1: Gütegrad Abschnitte WMTC

Abbildung 6-2 zeigt einen repräsentativen Ausschnitt aus dem ersten Fahrzyklusteil. Dies ist ein relativ niedriger Lastpunkt. Beim Schalten von einem Gang zum nächsten durchläuft der Prüfstand ein vorgegebenes Programm und es entsteht Instabilität. Dadurch treten vor allem in diesen Bereichen, sowohl für den voll- als auch für den teilautomatisierten Fahrverlauf, Überschreitungen der Grenzen auf.

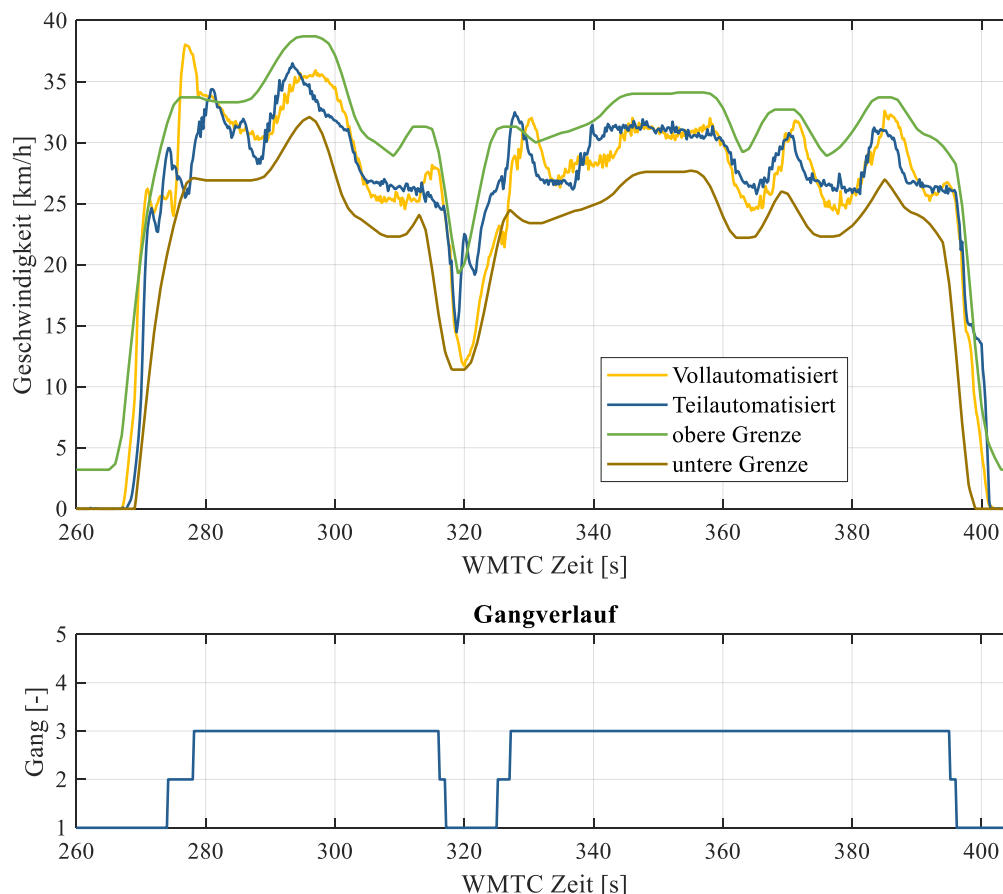


Abbildung 6-2: Vergleich Teil- mit Vollautomatisierung

In höheren Lastpunkten, ohne das Schalten von einem Gang zum nächsten, zeigt sich, dass sich die gefahrene Geschwindigkeit innerhalb der Grenzen befindet (Abbildung 6-3). Dies führt dazu, dass der Gütegrad δ_v im dritten Teil des WMTCs in allen drei Tests über 98% liegt (Tabelle 6-1).

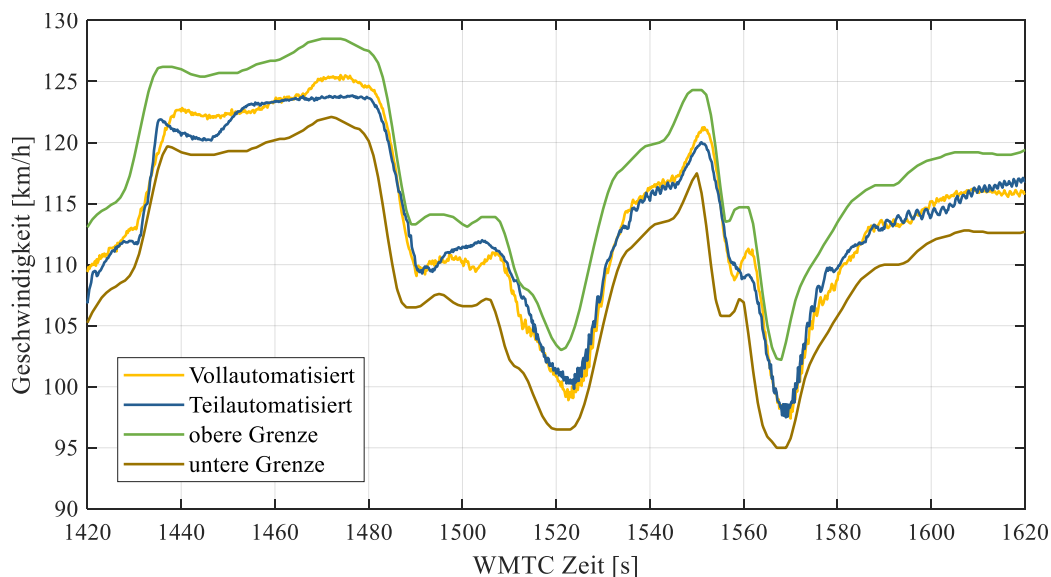


Abbildung 6-3: Teil 3 WMTC - Ausschnitt

Durch eine Optimierung des Schaltvorgangs könnte der Gütegrad noch erhöht werden, um auch im ersten und zweiten Teil des WMTCs einen Gütegrad über 98% zu erreichen.

Der Gesetzestext jedes Fahrzyklus gibt vor, wie lange das Fahrzeug innerhalb der vorgegebenen Grenzwerte mindestens sein muss, um diesen als gültig zu werten. Um Vorhersagen der Kennwerte des Verbrennungsmotors für die Legislation machen zu können, ist ein hoher Gütegrad notwendig.

Zusätzlich ist ein hoher Gütegrad notwendig, um den Verbrennungsmotor im aktuellen Setup zu einem Motorrad auf einem Rollenprüfstand vergleichen zu können, da auf der Rolle die Geschwindigkeitsgrenzen gut eingehalten werden können.

6.2 Wiederholbarkeit vollautomatisierter Motorprüfstand

Im Folgenden werden die beiden vollautomatisierten Prüfzyklen aus Tabelle 6-1 verglichen, wobei das gesamte Setup für beide gleich war.

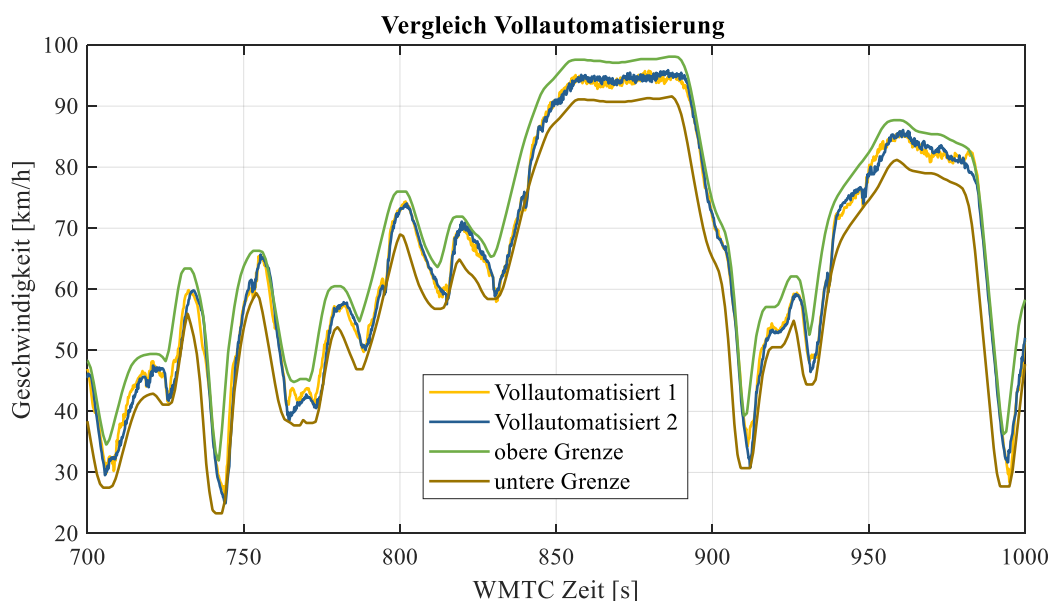


Abbildung 6-4: Vergleich Vollautomatisierung

Abbildung 6-4 stellt einen hohen transienten, repräsentativen Bereich des Fahrzykluses dar. Dabei ist ersichtlich, dass beide Messfahrten beinahe identisch sind. Daraus lässt sich schließen, dass bei gleichem Prüfstandsautomatisierungsgrad auch ein beinahe gleiches Verhalten des Prüflings zu erwarten ist. Dies ist durchaus vorteilhaft, da bei Änderungen des Prüflings bzw. der ECU direkte Vergleiche gemacht werden können.

In Abbildung 6-5 ist ein Detailausschnitt des Zykluses dargestellt. Zusätzlich zur oben genannten Wiederholbarkeit ist hier zu vermerken, dass sich größere Abweichungen der aktuellen Geschwindigkeit zur vorgegebenen Geschwindigkeit bei beiden Prüfläufen gleich verhält. Hervorzuheben dabei sind die Verläufe bei Sekunde 1250, bei der die aktuelle Geschwindigkeit oberhalb des oberen Grenzwertes liegt, jedoch tritt dieses Verhalten bei beiden Läufen identisch auf.

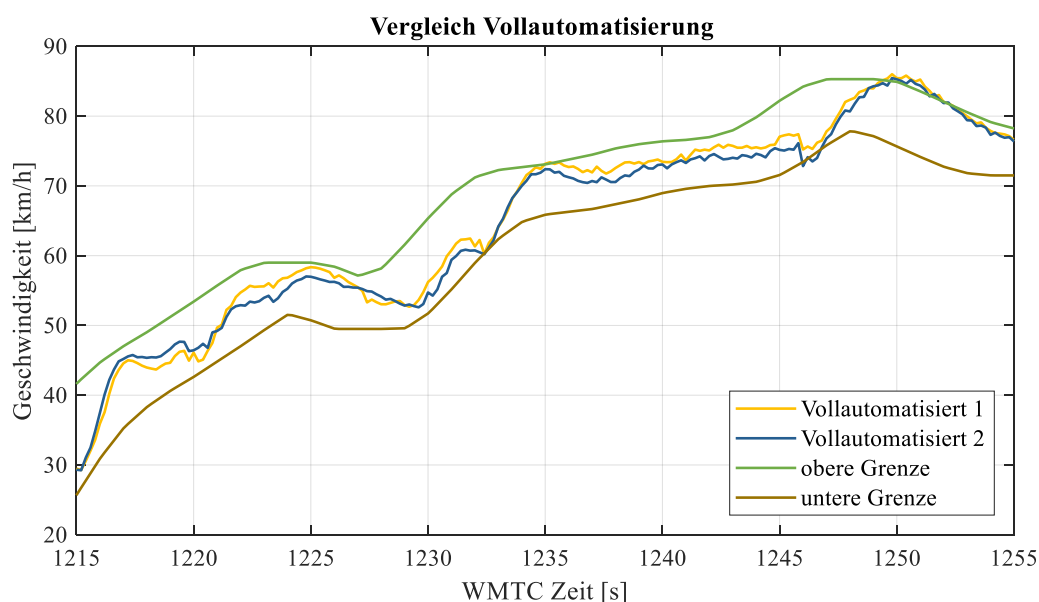


Abbildung 6-5: Unterschiede zweier Zyklen im Detail

Durch diese hohe Wiederholbarkeit des Gesamtprüfstandes ist eine Veränderung des Prüflings direkt mit der vorherigen Messung bzw. Referenzmessung vergleichbar. Daher kann auch auf Veränderung des Prüflings zurückgeschlossen werden, ob sich diese positiv bzw. negativ ausgewirkt hat. Vor allem bei Optimierungsaufgaben der ECU ist dies nützlich, da schnell positive Auswirkungen identifiziert werden können.

Diese Wiederholbarkeit ist nicht nur im Vergleich der Geschwindigkeiten zu beobachten, sondern auch in der gestellten Drosselklappenstellung, der abgegebenen Leistung und dem gemessenen Kraftstoffverbrauch. Abbildung 6-6 zeigt den gleichen Zeitbereich wie in Abbildung 6-5. Dabei ist zu erkennen, dass die Drosselklappenstellung, die Leistung und der Kraftstoffverbrauch einen beinahe identen Verlauf haben. Kleinere Abweichungen sind durch Umwelteinflüsse, wie unterschiedliche Öl-, Kühlwasser-, Umgebungstemperatur und Umgebungsdruck gegeben. Ebenfalls wird hier vermerkt, dass die Verläufe von der Sekunde 1245 bis 1253 ebenfalls in allen drei Diagrammen beinahe ident sind.

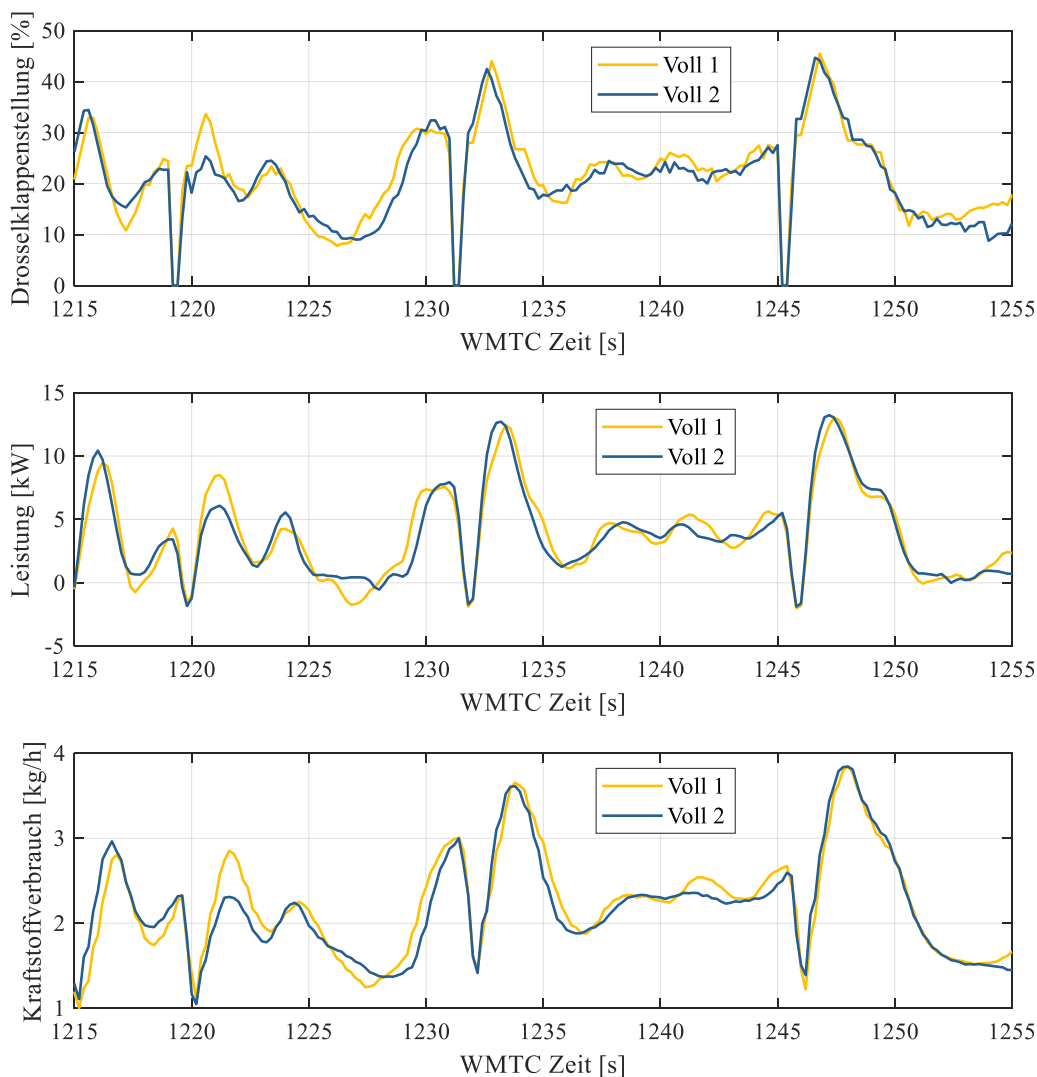


Abbildung 6-6: Vergleich Drosselklappenstellung, Kraftstoffverbrauch, Leistung

Im Folgenden kann angenommen werden, dass der Prüfstand und der Prüfling bei gleichem Setup eine sehr hohe Wiederholbarkeit aufweisen. Dabei ist kein hoher Gütegrad notwendig, wie in Kapitel 6.1 beschrieben, da z.B.: ein nicht „optimal“ eingestellter Geschwindigkeitsregler sich für beide Prüfläufe gleich verhält und somit die beiden Prüfzyklen relativ zueinander wieder sehr gut vergleichbar sind.

6.3 Verifizierung vollautomatisierter Motorprüfstand mit Rollenprüfstand und theoretischer Leistung

Eine Verifizierung eines vollautomatisierten WMTC Ablaufs wurde mit einer Rollenmessung und der theoretisch berechneten Leistung durchgeführt, um einen Richtwert zu schaffen und um zu untersuchen, ob sich die Leistung in einem verhältnismäßigen Bereich befindet. Das heißt, dass der Absolutwert der Leistungen im stationären Fall $< 2\%$ ist, wobei dies bereits Messungenauigkeiten sein können. In transienten Fall treten größere Unterschiede auf, jedoch sollten die Leistungsabweichungen zur theoretischen Leistung $< 2 \text{ kW}$ sein.

Abbildung 6-7 zeigt einen weniger transienten Hochlast-Bereich, welcher aufgrund der Geschwindigkeit gut vergleichbar ist. Es ist ersichtlich, dass die Geschwindigkeiten der theoretischen bzw. der Soll-Geschwindigkeit sehr gut folgen. Die theoretische Leistung wurde aus dem WMTC Gesetzestext berechnet und dargestellt. Ebenfalls liegt die Leistung in einem

nachvollziehbaren Bereich, wobei Leistungsspitzen abweichend von der theoretischen Leistung auftreten. Diese stammen aus kurzzeitig transienten Beschleunigungs- bzw. Verzögerungsvorgängen.

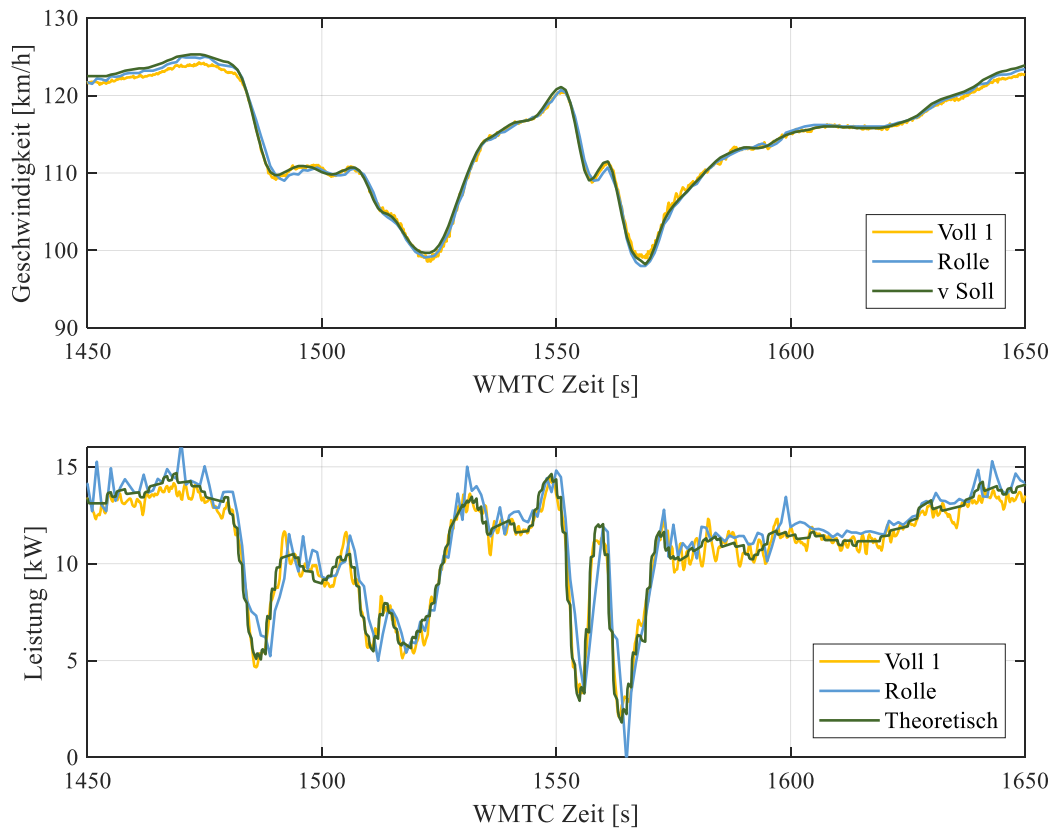


Abbildung 6-7: Vergleich vollautomatisierter Motorprüfstandsmessung mit Rollenmessung und theoretischer Leistung

Die theoretische Leistung wurde mit den folgenden Werten in (6-1) aus der WMTC Gesetzgebung für die BMW G 650 GS berechnet. Daraus berechnet sich die theoretische Leistung in Gleichung (6-2).

$$F_1 = 23,8N, \quad F_2 = 0,0241\left(\frac{km}{h}\right)^2, \quad m = 270kg \quad (6-1)$$

$$P_{theoretisch} = v_{Soll} * \left(F_1 + F_2 * v_{Soll}^2 + m * \frac{\Delta v}{\Delta t} \right) \quad (6-2)$$

Zum einen führen diese Leistungsunterschiede bzw. -spitzen durch das unterschiedliche Schaltverhalten und zum anderen durch die unterschiedlichen Fahrermodelle. Das Fahrermodell am Motorprüfstand ist ein PID-Regler und am Rollenprüfstand ein „echter“ Prüfstandstechniker.

Zusammenfassend liegt die Leistung im Hochlastbereich mit wenigen Ausnahmen innerhalb der oben genannten Bereiche sowohl für den vollautomatisierten Motorprüfstandsbetrieb als auch für die Rollenmessung im Vergleich zur theoretischen Leistung.

7 Zusammenfassung

Ziel der vorliegenden Diplomarbeit war es, auf einen bestehenden Stand einer Masterarbeit [1] einem Motorprüfstand Fahrzyklen konventioneller Verbrennungsmotoren vollautomatisiert abfahren zu können. Zum gegebenen Motorprüfstand wurden Erweiterungen und Modifikationen durchgeführt, um den Prüfstand automatisiert betreiben zu können. Dabei sind neben den einzelnen Teilsystemen gesamtübergreifende Einstellungen vorgenommen worden, um die jeweilige Fahrzyklussituation abfahren zu können.

Bei der Ausarbeitung der Vollautomatisierung war es notwendig, die „optimale“ Schnittstelle für zusätzliche Teilsysteme zu finden, um diese effizient nutzen zu können. Daher wird für zukünftige Arbeiten und eventuellen Erweiterungen zusätzlich zu dieser Masterarbeit aber auch auf eine vorangegangene Masterarbeit, bei der auf das Gesamtsystem und deren Teilsysteme im Detail eingegangen wird, verwiesen [24]. Daraufhin können die Architektur und die Veränderungen des Prüfstandes leichter nachvollzogen werden.

Ein adaptives Kupplungsaktuierungssystem wurde entwickelt, das zeiteffizient eingestellt und für einen anderen Verbrennungsmotor bzw. auf einem anderen Prüfstand modular eingesetzt werden kann. Dieses System wurde möglichst einfach aufgebaut, um zukünftige Entwicklungsarbeiten möglichst effizient gestalten zu können.

Zusammenfassend kann darauf geschlossen werden, dass eine Vollautomatisierung des Motorprüfstands eine wesentliche Erleichterung für den Prüfstandstechniker ergibt, um andere Tätigkeiten, wie z.B. das Applizieren diverser Kennfelder der ECU, live durchführen zu können.

Aufgrund der hohen Wiederholbarkeit bietet die Vollautomatisierung die Möglichkeit, dass Änderungen am Prüfling direkt mit den vorherigen Messfahrten verglichen werden können und dadurch schnell auf positive bzw. negative Veränderungen reagiert werden kann.

Ebenfalls ist ein Vergleich des Motorprüfstands mit einem Motorrad bei einer Rollenprüfstandsmessung mit dem gleichen Fahrzyklus möglich, um Vorhersagen bezüglich Performance und Kraftstoffverbrauch treffen zu können.

Hier wird angemerkt, dass in Zukunft hauptsächlich Fahrzyklen vollautomatisiert abgefahren werden sollten, da dies eine effiziente Steigerung der Entwicklungsschritte darstellt. Dabei muss dies nicht unbedingt ein Fahrzyklus einer Gesetzgebung sein, sondern es kann jede mögliche Zusammensetzung von verschiedensten Geschwindigkeit- und Gangvorgaben sein. Diese sehr hohe Reproduzierbarkeit und Optimierungsmöglichkeit sollten in Zukunft vermehrt genutzt werden.

Aus Gründen der Lesbarkeit ist die Beschreibung der Versuchsplanung bzw. Versuchsdurchführung für eine vollautomatisierte Prüfung im Anhang zu finden.

8 Aussicht

In diesem Kapitel sind noch weitere Möglichkeiten dargestellt, um den Prüfstand weiter auszubauen und für zukünftige Projekte nutzen zu können.

8.1 Systemarchitektur Hybridregelung

Der Motorprüfstand kann vollautomatisiert für reine Verbrennungsmotoren verwendet werden. Jedoch kann mit einigen Schritten eine Hybridsystemarchitektur geschaffen werden, um Themen wie Reduktion des Kraftstoffverbrauchs, Rekuperation und Lastpunktverschiebungen einzupflegen und den Verbrennungsmotor in einem Hybridsystem zu emulieren.

Eine relativ „einfache“ Möglichkeit wäre es, das Elektromotor-Moment als Fahrbahnsteigung in das Prüfstandssystem zu implementieren.

Eine weitere Möglichkeit bestünde darin, eine Hybrid Control Unit zu implementieren, die eine oder mehrere verschiedene Betriebsstrategien (dynamisches Fahren, minimaler Kraftstoffaufwand bzw. Minimierung von Emissionen) beinhaltet.

Angemerkt wird hier, dass durch die Implementierung eines Elektromotors die emulierte mechanische Bremse ersetzt werden kann. Daher kann Energie rückgewonnen werden und zur Minimierung des Energieverbrauchs beigetragen werden.

8.2 Kennfeld Optimierung

Durch die vorliegende Automatisierung des Motorprüfstands könnte eine automatisierte Optimierung der ECU Kennfelder während des Abfahrens von verschiedenen Zyklen durchgeführt werden. Dies hätte den Vorteil, dass ein Zyklus mehrmals abgefahren wird und die verschiedenen Kennfeld-Variablen und Kennwerte automatisiert verändert werden um, zum Beispiel, den „optimalen“ Kraftstoffverbrauch zu eruieren.

Dabei könnte folgendermaßen vorgegangen werden:

- Spezifizierung der verschiedenen Fahrzyklen und deren Bestimmungen
- Ablaufen dieser Fahrzyklen am Verbrennungsmotor inklusive Abgasmessung
- Analyse der Kennfeldbereiche in dem die Zyklen abgefahren wurden
- Vollautomatisches Abfahren dieser Zyklen und automatisierte Anpassung der Kennfelder für minimalen Kraftstoffverbrauch und Emissionsausstoß, um alle Gesetzschriften einzuhalten
- Applizierung der Kennfeldbereiche die außerhalb der Fahrzyklen liegen, um ein hochdynamisches Ansprechverhalten bzw. maximalen Fahrspaß zu erzielen

Vorteil dieser Methode wäre, dass zum einen die Gesetzgebungen und auch die kundenspezifischen Anforderungen, wie dynamisches Fahren, erfüllt sind.

8.3 Vollautomatisiertes reales Schalten

Mit der Kupplungsaktuierung ist es nun möglich ein weiteres Teilsystem zu entwickeln. Dies wäre eine Schaltvorrichtung, die direkt auf den Schalthebel am Getriebe zwischen den Gängen schaltet.

Eine mögliche Ausführung wäre ein Zwei-Wege Pneumatikzylinder, der über die Tornado Prüfstandssteuerung bzw. über den Microcontroller von der Kupplung aktuiert wird.

Dies hätte den Vorteil, dass das Schalten zwischen den Gängen realitätsnah abgebildet werden kann. Ebenfalls würden die physikalischen Wirkungsgrade und der Massenträgheitsmomente der Gänge vorliegen.

Zum Teil haben Messungen gezeigt, dass im Schaltvorgang im ungünstigsten Fall kurze jedoch hohe Geschwindigkeitssprünge auftreten und diese durch eine Erweiterung des Prüfstandes ausgemerzt werden können.

9 Anhang

9.1 Erstmaliger Aufbau und Inbetriebnahme Vollautomatisierung - Schritt-für-Schritt Anleitung

Hier werden die notwendigen Einstellungen des Motorprüfstandes für die vollautomatisierte transiente Fahrt aufgelistet. Eine Dokumentation und Schritt-für-Schritt Anleitung für zukünftige Fahrten. Dies ist ein Anhaltspunkt um einen Test erfolgreich durchzuführen. Dabei sind die Files in dem Ordner „Vollautomatisierung_Versuchsdurchführung“ zusammengefasst, welche als Referenz verwendet können. Es sind zwei sehr nützliche Dokumentationen „Tornado Designer – Benutzerhandbuch“ und „Tornado RT – Benutzerhandbuch“ vorhanden, welche für das Verständnis des Prüfstands zusätzlich notwendig sind.

9.1.1 Schritt 1: Vorbereitungen - Tornado Designer

9.1.1.1 Gewünschtes Fahrprofil

Hier wird eine Tabelle mit Geschwindigkeit- und Gangverlauf erstellt, die im Folgenden als Parametertabelle in den Tornado Designer geladen wird. Dabei kann ein bestehendes Excel File „EMISSION_EVALUATION_REGULATION_EU_134“ verwendet werden, das einen Fahrzyklus erstellt. Dabei ist darauf zu achten, dass die „global input parameters“ (Abbildung 9-1) korrekt für das Motorrad eingegeben werden. Danach kann unter dem Excel-Blatt „output“ mit „export speed table“ und „export gear table“ der notwendige WMTC Fahrzyklus erstellt werden.

manufacturer:	BMW
model:	650
frame number:	XXX
engine number:	XXX
milage [km]:	
displacement [cm ³]:	650,0
working principle:	4-stroke
fuel type:	Gasoline (E5)
GDI Engine:	NO
fuel density [kg/l]:	0,740
engine type:	PI engine
transmission:	manual
number of gears:	5
rated speed [rpm]:	6600
idling speed [rpm]:	1800
max. vehicle speed [km/h]:	200,0
Power [kW]:	37,0
tyre pressure [bar]:	3,0
vehicle mass [kg]:	195,0
driver mass [kg]:	75
reference mass [kg]:	270
equivalent inertia mass [kg]:	270

Abbildung 9-1: Globale Eingabewerte

Der erstellte Speedtable besteht aus den folgenden Spalten von A bis D: Zeit [s], vorgegebene Geschwindigkeit [km/h], untere und obere Geschwindigkeitsgrenze [km/h]. Beim Erstellen des Geartables wird die Zeit und der Gang in den Spalten A und B erstellt. Nur der Gangverlauf wird kopiert und in den Speedtable in Spalte E eingefügt. In Zelle E1 wird folgende Funktion eingesetzt und über die gesamte Länge des Prüflaufs kopiert:

$$= WENN(E1 < 0,5; 1; E1) \quad (9-1)$$

Diese Funktion erstellt den Gangverlauf von Gang 1 bis zum letzten Gang ohne den Nullern und minus Einsern. Als nächstes wird die Spalte F für die Kupplungsaktuierung für das Einkuppeln erstellt. Wenn der Gang von -1 auf 1 springt wird eine Eins mit der folgenden Funktion (9-2) gesetzt:

$$= WENN(UND(E2 = 1; E1 = -1); 1; 0) \quad (9-2)$$

Wenn Tornado RT diese Eins einließt, wird das Befehlsprogramm „Kupplung_Schließen“ durchgeführt. Hier ist anzumerken, dass eine Abfrage in Tornado erstellt werden kann, bei dem der Einkuppelvorgang automatisch eingeleitet wird. Aus zeitlichen Gründen wurde dies nicht mehr implementiert.

Somit wurde ein Fahrzyklus mit Zeit, vorgegebener Geschwindigkeit, untere und obere Geschwindigkeit, Gangverlauf, Gangverlauf für Tornado und Einkuppelaktivierung erstellt (Abbildung 9-2). Diese Tabelle muss mit dem Dateityp „.csv“ abgespeichert werden und es muss darauf geachtet werden, dass der Tornado Prüfstandsrechner darauf Zugriff hat.

	A	B	C	D	E	F	G
1	0	0,0	0,0	0,0	0	1	0
2	1	0,0	0,0	3,2	0	1	0
3	2	0,0	0,0	3,2	0	1	0
4	3	0,0	0,0	3,2	0	1	0

Abbildung 9-2: Fahrzyklus Teilausschnitt

Im nächsten Schritt wird die Tabelle in den Tornado Designer als Parametertabelle angelegt. Dabei im Tornado Designer unter Parametertabelle > WMTC_GangKupplung_Automatisiert mit Rechtsklick ein „neues Element einfügen“. Mit „Import/Export“ das vorhin erzeugte Fahrprofil öffnen und importieren auswählen. Darauf sollte nun der Fahrzyklus grafisch dargestellt sein (Abbildung 9-3). Die einzelnen Kurven können übersichtshalber unter „Dargestellte Kurve“ ausgewählt werden und auf Korrektheit geprüft werden.

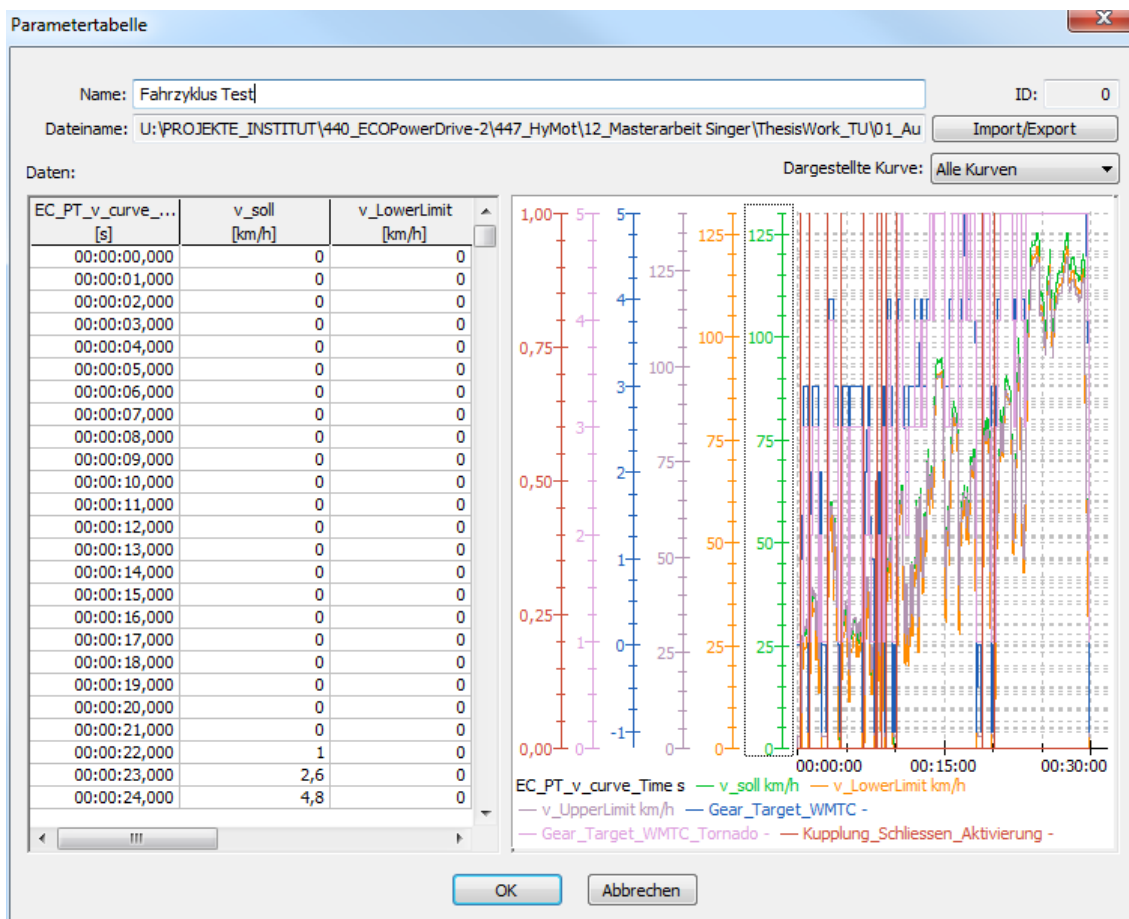


Abbildung 9-3: Geladene Parametertabelle

Nun wird noch der gewünschte Name vergeben und mit OK bestätigt. Daraufhin ist der Fahrzyklus als Parametertabelle angelegt.

9.1.1.2 Prüfprogramm Erstellen

Nun wird das Prüfprogramm erstellt. Hierbei kann auf das vorhandene Prüfprogramm im Tornado Designer „Transient_Fahrerleitsystem_WMTC“ (Abbildung 9-4) referenziert werden.

Zeile	Befehl	Parameter
1	Zuweisung	Alpha_Auswahl = 2 -
2	Zuweisung	EC_GearShift_Mode = "Manuell"
3	Parametertabelle laden	WMTC_GangKupplung_Automatisiert
4	Warten	2
5	Zuweisung	MCU_Handbetrieb_Out = 1 -
6	Recorderanzeige Start	Transient_RLS_alpha
7	Recorder laden	Transient_RLS_alpha
8	Recorderlogging Start	Transient_RLS_alpha
9	Zuweisung	v_soll = 0 km/h
10	Zuweisung	Geschwindigkeitsregelung_Aktivierung = 1 -
11	Zuweisung	Alpha_Geschwindigkeitsregelung = 0 -
12	Fahrerleitsystem	Initialisierung
13	Fahrerleitsystem	Start
14	Zuweisung	EC_PT_v_Aktivierung = 1 -
15	Warten	30:00
16	Zuweisung	EC_PT_v_Aktivierung = 0 -
17	Fahrerleitsystem	Stopp
18	Recorderlogging Stopp	Transient_RLS_alpha
19		

Abbildung 9-4: Prüfprogramm

Beim Start des Prüfprogramms wird jede Zeile der Abbildung 9-4 von 1 bis 18 abgearbeitet. Die Prüfprogramm Zeilen werden kurz erläutert:

1. Hier wird die Drosselklappenvariable aktiviert, auf die der Geschwindigkeitsregler und beim Ein-/Auskuppeln gegriffen wird.
2. Der manuelle Gangmodus wird ausgewählt, da der Gangverlauf in dem Fahrzyklus (wie in Kapitel 9.1.1.1) vorgegeben wird. Bei Stellung auf Automatik werden die „Automatik Schaltpunkte“ von Abbildung 4-1 ausgewählt.
3. Hier wird der Fahrzyklus für den Prüflauf geladen. Das heißt, hier muss der gewünschte Fahrzyklus ausgewählt werden, welcher beispielhaft in Kapitel 9.1.1.1 gezeigt wurde.
4. Das Laden des Fahrzyklus benötigt etwas Rechenzeit, daher wird hier eine Wartezeit eingefügt. Bei sehr großer Datenmenge des Fahrzyklus kann hier der Wartebefehl verlängert werden.
5. Die Zuweisung ist notwendig um die MCU zu aktivieren und manuell bedienen zu können und dadurch Eingriffe in die Steuerung tätigen zu können.
6. Hier wird das Recorderfenster gestartet.
7. Hier wird der spezifische Recorder, welcher im Tornado Designer erstellt wird, geladen.
8. Dieser Befehl startet das Speichern der Daten.
9. Die vorgegebene Geschwindigkeit wird hier Null gesetzt, da bei einem Abbruch eines automatisierten Fahrzyklus die letzte Geschwindigkeitsvorgabe nicht Null sein muss und daher der Geschwindigkeitsregler die Drosselklappe öffnen könnte.
10. Der Regler für die Geschwindigkeit wird aktiviert.
11. Die Drosselklappenstellung wird genullt.
12. Das Fahrerleitsystem wird initialisiert.
13. Das Fahrerleitsystem wird gestartet.
14. Mit der Zuweisung „EC_PT_v_Aktivierung“ wird das Durchlaufen des Fahrzyklusses bzw. der Parametertabelle gestartet.
15. Die Länge des Fahrzyklusses soll hier vorgegeben werden. Bei einem WMTC sind das 30 Minuten.
16. Stopp des Durchlaufens der Parametertabelle.
17. Das Fahrerleitsystem wird gestoppt.
18. Der Recorder wird gestoppt. Die Recorderdaten werden automatisch in C:\Tornado\data gespeichert.

Hier sei noch anzumerken, dass der Befehl „Motor Start“ noch eingefügt werden muss, wenn der Starter der VKM vorhanden ist. Dieser Befehl sollte zwischen dem Befehl 14 und 15 eingefügt werden. Dies wurde soweit vorbereitet, jedoch nicht durchgeführt, da der Starter der vorhandenen VKM nicht angeschlossen war. Daher wurde die VKM hochgeschleppt und die Zündung eingeschaltet und daraufhin das Prüfprogramm gestartet.

9.1.2 Schritt 2: Vorbereitung – Kupplungsaktuator

9.1.2.1 Aufbau

Der Kupplungsaktuator besteht im Wesentlichen aus dem Elektromotor und der Halterung, der Elektronikbox und der Leistungsversorgung.

Der Elektromotor wurde auf einem Profil montiert. Dabei wird ein 10k Ohm Potentiometer in der Drehachse befestigt. Das Profil wird mit zwei Montagewinkel auf der Prüfstandsmontageplatte so montiert (Abbildung 9-5, links), sodass die „Flucht“ des Kupplungsseils auf den Kupplungshebel ohne Streifen anderer Bauteile möglich ist (Abbildung 9-5, rechts). Dabei kann die Höhe des Elektromotors noch mit den eingezeichneten Schrauben eingestellt werden (Abbildung 9-6). Ebenfalls muss das Potentiometer nachgerückt werden. Das Kupplungsseil wird danach wieder ausgehängt.

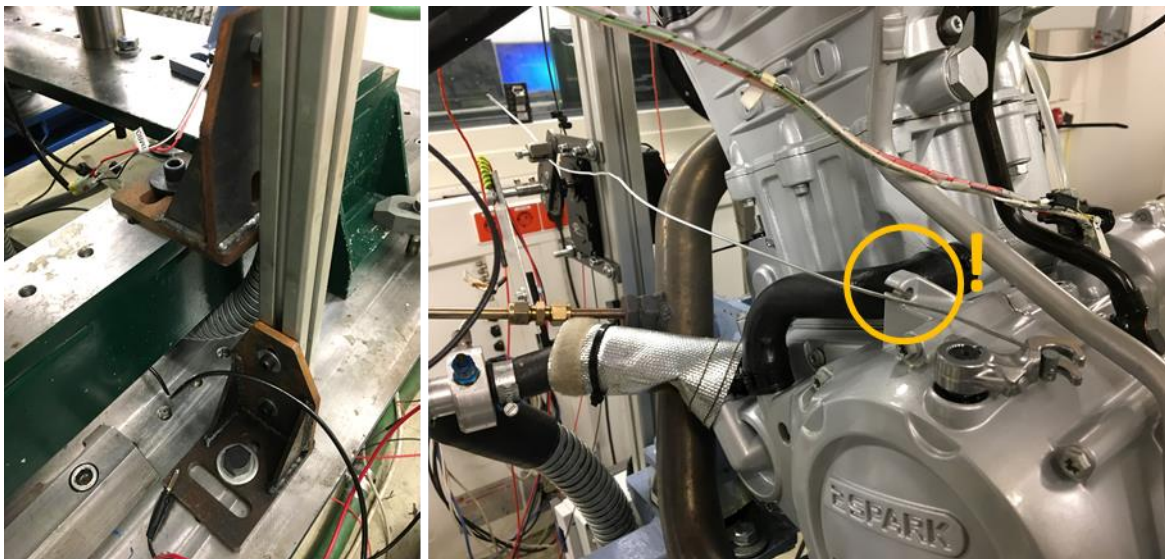


Abbildung 9-5: Montagewinkel (links), Kupplungsseilflucht (rechts)

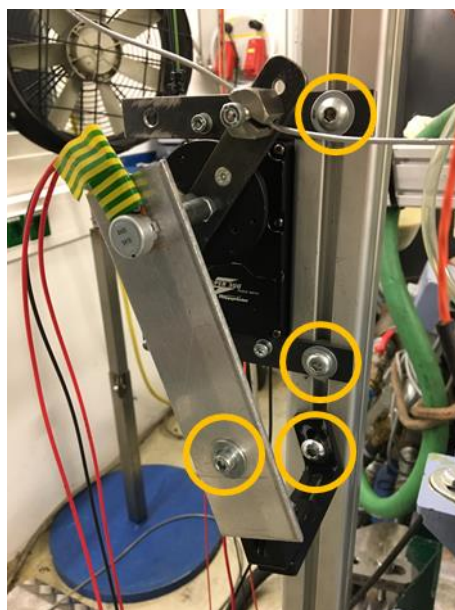


Abbildung 9-6: Elektromotor und Potentiometer Positionierung

Die Elektronikbox ist in Abbildung 9-7 in mehreren Ansichten gezeigt. Die Box kann mit dem blauen Hebel und mit den beiden Schrauben geöffnet werden (Abbildung 9-7, links oben). Die Box erfüllt den Standard IP68 und ist daher staubdicht und wasserfest, daher ist das Öffnen und

Schließen etwas „kräftiger“ durchzuführen.

Abbildung 9-7 links unten zeigt die Steueranschlüsse. Die Spannungsversorgung bzw. die Kommunikation erfolgt mit dem USB Typ B Anschluss. Das Potentiometerkabel und das analoge Eingang-Kabel werden über jeweils einen Industriestecker angeschlossen. Die Kabelanschlüsse des Potentiometers sind wie folgt im Industriestecker verlötet: auf 1 ist die 5V Spannungsversorgung, auf 2 ist ein analoger Eingang und der GND-Anschluss ist mit dem Ground vom Arduino verbunden. Der analoge Eingang Stecker ist wie folgt angeschlossen: auf 1 ist ein analoger Eingang und auf 2 der Ground angeschlossen.

Abbildung 9-7 rechts unten zeigt jeweils einen positiven und negativen Anschluss für die Spannungsversorgung (V) und die Motoranschlüsse (M). Dabei ist jeweils das rote Kabel (+Kabel) und das schwarze Kabel (-Kabel) an den dafür vorgesehenen Buchsen anzuschließen. Somit ist die Steuerbox vollständig angeschlossen und die Elektronikbox wird am Profil eingehakt (Abbildung 9-7, rechts oben).

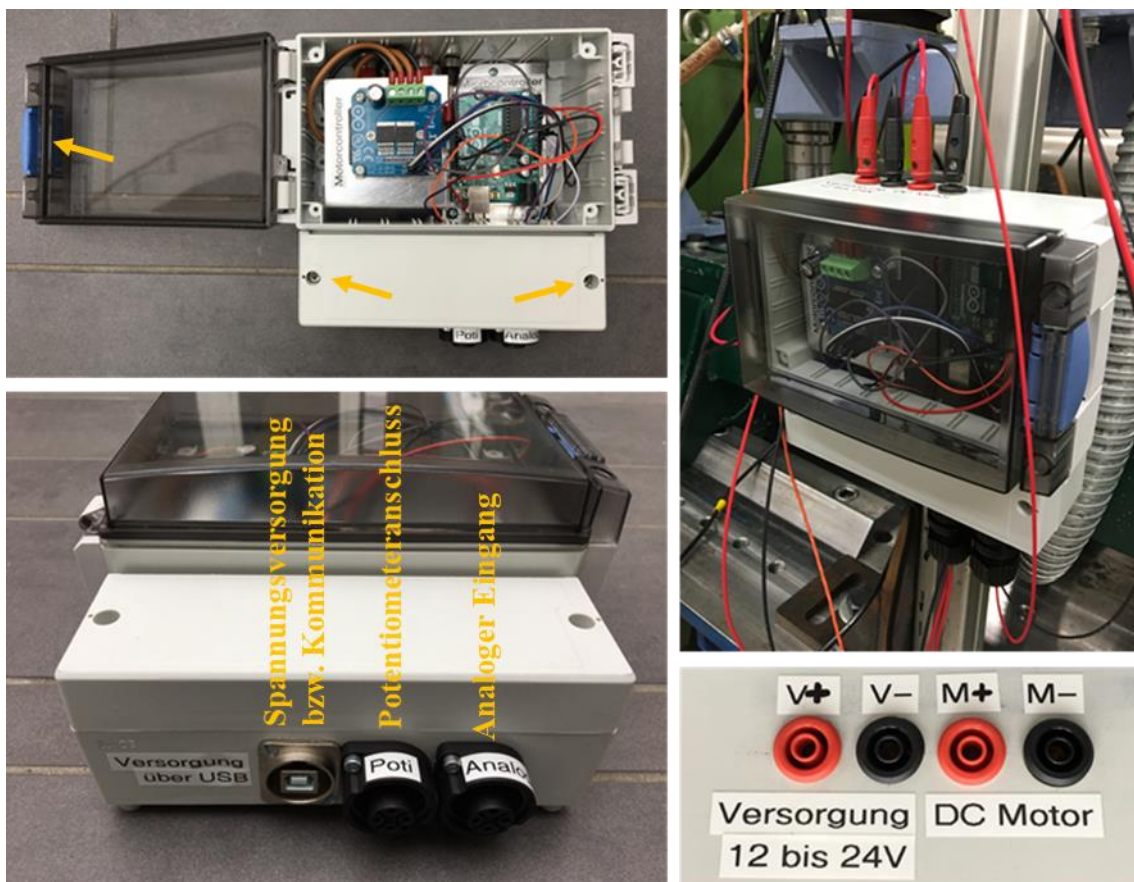


Abbildung 9-7: Elektronikbox und Anschlüsse

Die Spannungsversorgung (Abbildung 9-8) wird auf 12V und 3,5A Maximalstrom (bei einer Motorradkupplung) eingestellt. Der eingestellte Wert kann mit dem Drücken des „display voltage and current setting“ überprüft werden. Dies ist auch eine Sicherung gegen das Überziehen und einer mechanischen Beschädigung der Kupplung. Das plus und minus Versorgungskabel (V+ und V-) wird an die Spannungsversorgung an den vorgesehenen Bananenstecker-Buchsen (+ und -) angeschlossen.

Beim Auskuppelvorgang wird die Kupplung „schnellstmöglich“ geöffnet, wobei die limitierende Komponente der Maximalstrom der Spannungsversorgung ist. Daher ist darauf zu achten, dass der Maximalstrom auch bei 3,5A liegt.

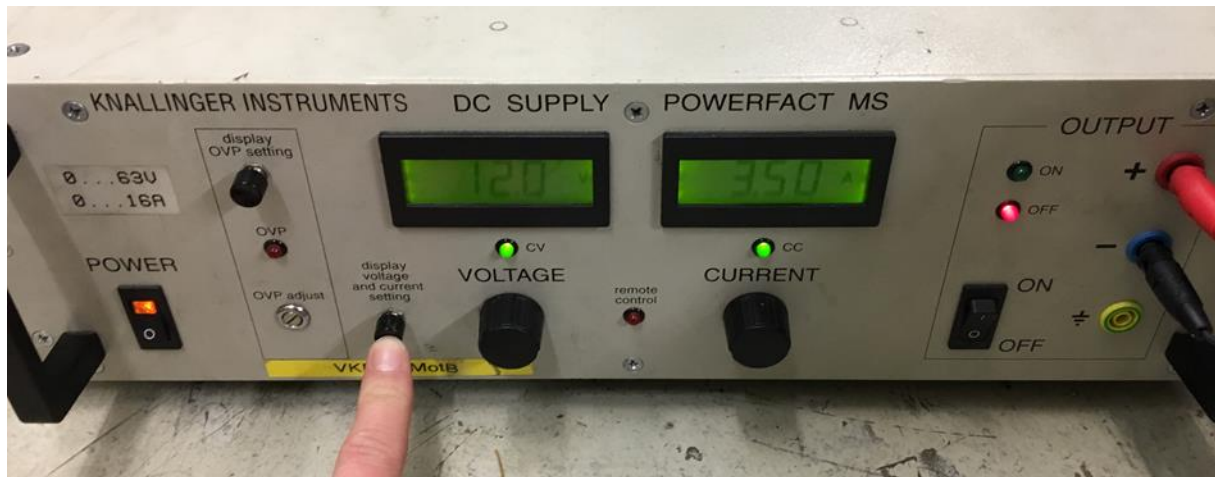


Abbildung 9-8: Spannungsversorgung für Leistungselektronik

Nun ist der Quellcode von Kapitel 9.4 für den Arduino UNO hochzuladen. Nachdem dies geschehen ist, kann mit dem angeschlossenen PC an der Elektronikbox der Serial Monitor geöffnet und der Poti_Soll und Poti_Ist Wert dargestellt werden. Die Poti_Soll Spannung liegt zwischen 0 und 5V. Der gemessene Poti_Ist-Wert liegt ebenfalls zwischen 0 und 5V. Dabei werden beide Werte im Serial Monitor in mV angezeigt und zyklisch aktualisiert.

Abbildung 9-9 zeigt die Zusammensetzung des Kupplungsaktuator-Teilsystem. Der Laptop wird hier temporär verwendet, um die Position des Potentiometers einzustellen. Der Laptop kann später, wenn der Prüfstand für den vollautomatisierten Betrieb bereit ist, abgeschlossen werden und mit einem Trafo direkt angeschlossen werden. Beim Abschließen des Laptops und Anschließen des Trafos geht der Quellcode im Arduino UNO nicht verloren. Der Quellcode wird nach Anschließen der Spannungsversorgung erneut automatisch berechnet.

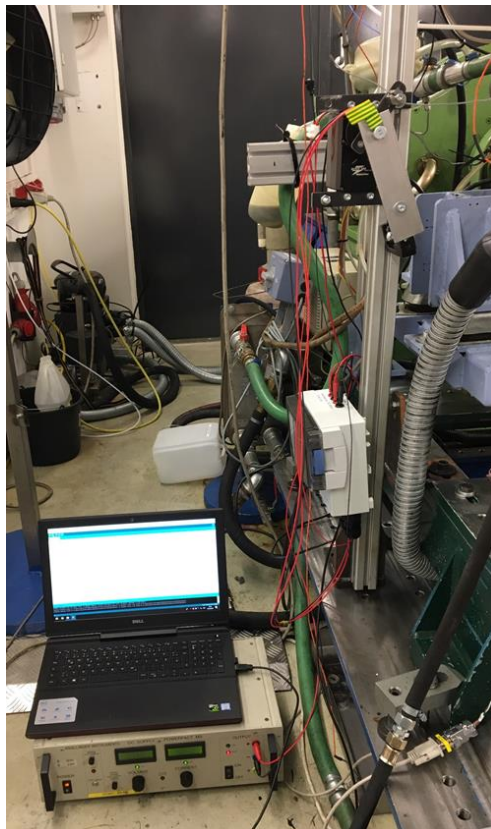


Abbildung 9-9: Kupplungsaktuator System

1;Heading 1 to the text that you want to appear here.

Nun ist noch darauf zu achten, dass keine Kabel im Drehbereich des Elektromotors sind. Nach dem Aufbau des Kupplungsaktuators kann nun mit der Positionierung bzw. der Bestimmung der Grenzen begonnen werden.

9.1.2.2 Positionsbegrenzung Arduino UNO

Die Bewegungsfreiheit des Elektromotors wird begrenzt, da zum einen das Kupplungsseil zu weit gezogen wird oder der Drehpotentiometer überdreht wird. Daher wird im Arduino ein Code implementiert, in dem zwei Variablen `Min_Poti_Position` und `Max_Poti_Position` eingestellt werden.

Bevor die Begrenzung eingestellt wird, muss noch die Umrechnung von der Tornado Variable auf den analogen Ausgang an der Sensorbox bzw. dem analogen Eingang an der Elektronikbox geklärt werden. Der `Kupplung_Status` Wert liegt zwischen 0 und 1, wobei 0 der geschlossenen Kupplung und 1 der voll geöffneten Kupplung entspricht. Dies wird im Alias linear auf den analogen Ausgang (0 bis 5V) umgerechnet und an der Sensorbox ausgegeben (Abbildung 9-10).

The screenshot shows the 'Alias' configuration window. The 'Quelle' (Source) section has 'Teilprüfstand' set to 'Kupplung_Status'. The 'Ziel' (Target) section has 'Teilprüfstand' set to 'SB_AOUT_X47'. Below these are checkboxes for 'Rechte in Tornado RT' (permissions). A table shows the mapping: 'Kupplung_Status' values 0 and 1 correspond to 'SB_AOUT_X47' values 0 and 5 respectively. A graph shows a linear relationship from (0,0) to (1,5). The 'Darstellung' (display) is set to 'Linearisierter Wert'. Buttons for 'k/d Wert', 'Importieren', 'OK', and 'Abbrechen' are visible.

Kupplung_Status	SB_AOUT_X47
0	0
1	5

Abbildung 9-10: Umrechnung `Kupplung_Status` auf analogen Ausgang `SB_AOUT_X47`

Nun muss in Tornado RT unter Diagnose > Tornado RT > Variablen die Variable Kupplung_Status gesucht werden. Danach kann am Laptop in der Programmiersoftware „Arduino IDE“ der Positionierbereich gestellt werden. Der Positionierbereich gibt die maximale und minimale Drehwinkelposition vor, die vom Elektromotor gestellt werden kann. Dabei soll die Kupplung zum einen voll geöffnet sein (bzw. die VKM vom Antriebsstrang getrennt) und zum anderen das Kupplungsseil leicht locker durchhängen.

Für die beiden Variablen Min_Poti_Position und Max_Poti_Position kann als Richtwert eine Differenz von ca. 1000mV geschätzt werden (vorausgesetzt einer Spannungsversorgung des Potentiometers von 5V wie sie in diesem Fall vorliegt). Das heißt zum Beispiel, dass die Min_Poti_Position auf 2000mV und Max_Poti_Position auf 3000mV gesetzt wird. Max_Poti_Position ist die Sicherung gegen das Überziehen der Kupplung. Falls ein Wert von Tornado vorgegeben wird, der die Position weiter vorgibt als dieser Wert, dann bleibt der Elektromotor in dieser Stellung.

Die Positionierung wird wie folgt eingestellt:

1. Potentiometer Fixierschraube öffnen (Abbildung 9-11)

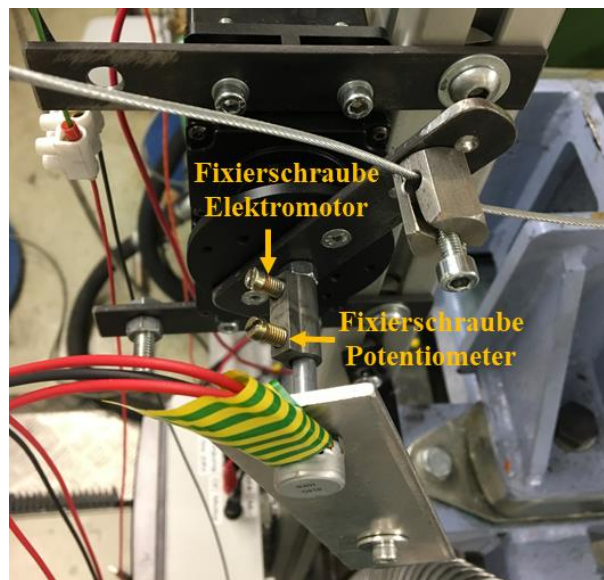


Abbildung 9-11: Kopplung Potentiometer mit Elektromotor

2. Elektromotor auf ca. +30° drehen (Abbildung 9-12)

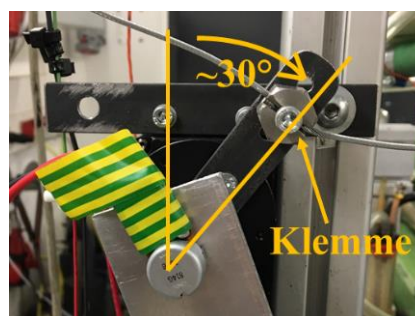


Abbildung 9-12: Elektromotorhebel Position

3. Das Kupplungsseil bei der Kupplung an der VKM einhängen, mit der Hand spannen (ca. 5kg Zugkraft, dabei hängt das Seil noch leicht durch) und das Kupplungsseil klemmen (Abbildung 9-12).
4. Die Position des Potentiometers sollte bei etwa 40% des Messbereichs (bei der anliegenden 5V Versorgung ist dies bei 2V) fixiert werden, um noch genügend

Drehweg für die Kupplungsaktuierung zu haben. Potentiometer auf den Messwert von 2000mV drehen (Ablesen in Serial Monitor in Arduino IDE) und Fixierschraube festziehen.

5. Diese Potentiometer Position für Min_Poti_Position im Arduino Quellcode ändern.
6. Nun den geänderten Code der Arduino Software hochladen.
7. Nun die Tornado Variable Kupplung_Status auf 0 setzen.
8. Die Ausgangsspannung der Spannungsversorgung der Leistungselektronik freigeben (Abbildung 9-8, rechts bei Output auf ON stellen).
9. Nun die Tornado Variable Kupplung_Status von 0 in 0,05 Schritten erhöhen bis die Kupplung vollständig geöffnet ist und sich die Bremse frei drehen lässt. Zwischen den Schritten prüfen ob die Kupplung nicht überzogen wird. Dabei sind die Variablen Poti_Soll und Poti_Ist im Serial Monitor mehrmals zu überprüfen, ob die Werte übereinstimmen.
10. Die gemessene Potentiometer Position plus 200mV für Max_Poti_Position im Arduino Quellcode ändern.
11. Nun den geänderten Code der Arduino Software hochladen.
 - a) Falls der Antriebsstrang noch nicht getrennt ist und die voreingestellte Positionsbegrenzung bereits aktiv ist, dann muss die Variable Kupplung_Status = 0 gesetzt werden, die Variable Max_Poti_Position erhöht werden und hochgeladen werden. Dann den vorherigen Schritt wiederholen.
 - b) Im Arduino UNO Serial Monitor wird die Vorgabe von Tornado mit Poti_Soll und die gemessene aktuelle Position mit Poti_Ist dargestellt. Die Begrenzung kann überprüft werden, in dem die Position mit Kupplung_Status = 0 oder 1 vorgegeben wird, jedoch der Elektromotor nur bis zur jeweiligen Begrenzung fährt.
12. Ab hier ist mit der Variable Kupplung_Status der Elektromotor an beide Positionsbegrenzungen und alle Zwischenpositionen anfahrbar.

Wenn die Begrenzungen funktionieren, kann fortgesetzt werden. Ansonsten muss die Positionsbegrenzung erneut durchgeführt werden.

Die Kupplungsposition-Vorgabe für das Aktuieren und der Feineinstellung der Kupplung erfolgt in Tornado RT.

9.1.2.3 Einstellung Einkuppelvorgang in Tornado RT

Die Feineinstellung für den Einkuppelvorgang erfolgt im Tornado RT. Dabei ist das Fenster in Tornado RT > Fenster > Kupplung zu öffnen (Abbildung 9-13).

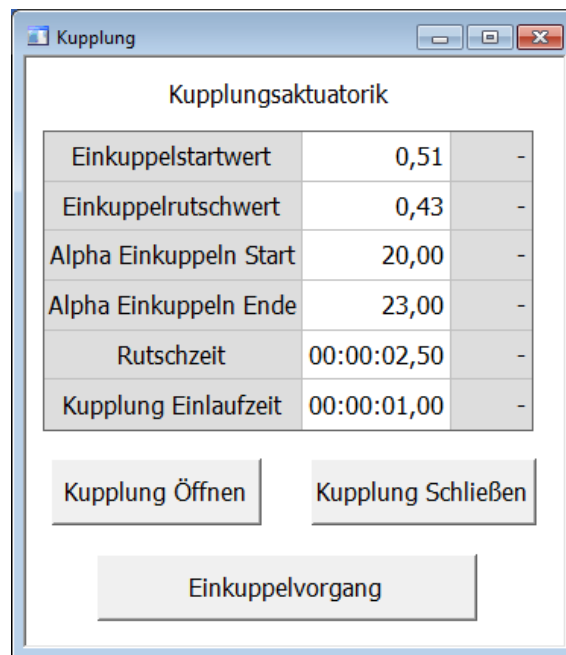


Abbildung 9-13: Kupplungseinstellungen

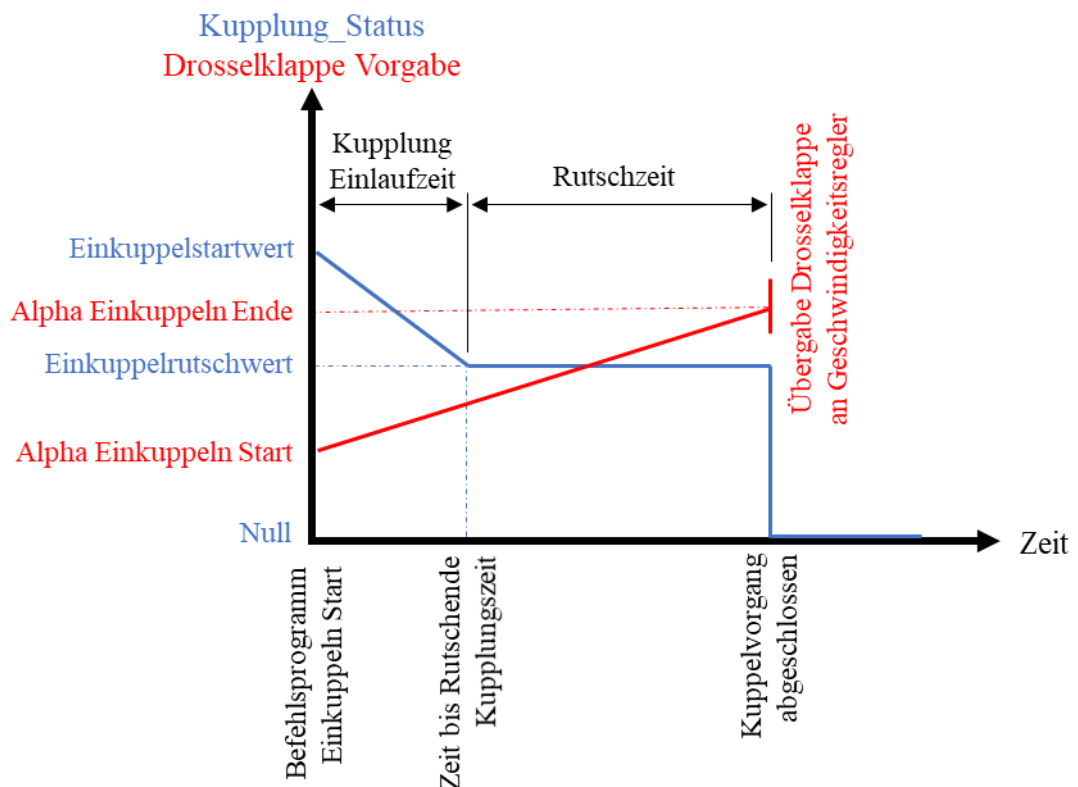


Abbildung 9-14: Einkuppelvorgang Variablenzuordnung

In Abbildung 9-13 wird der Einkuppelvorgang mit sechs Variablen definiert, welcher grafisch in Abbildung 9-14 dargestellt ist. Dabei gibt der Einkuppelstartwert die Position unmittelbar vor der Berührung der Kupplungsscheiben an. Der Einkuppelrutschwert ist der Wert bei dem die Kupplung einen „gewissen“ Schlupf hat (die Definition erfolgt später). Alpha Einkuppeln

Start gibt die Drosselklappenstellung vor, bei der das Befehlsprogramm gestartet wird und Alpha Einkuppeln Ende den Endwert zum Schluss des Befehlsprogramms. Die Drosselklappe wird vom Startwert bis zum Endwert linear über die Zeit berechnet. Die Kupplung Einlaufzeit ist die Zeit vom Starten des Befehlsprogramms (= Ausgekuppelten Zustand) bis zum Einkuppelrutschwert. Diese ist notwendig, um ein „sanftes“ Einkuppeln einzuleiten. Die Rutschzeit ist die Zeit anschließend der Einlaufzeit in der die Kupplung auf dem Einkuppelrutschwert verweilt. Der Einkuppelendwert ist von Tornado RT aus Null, jedoch von dem Arduino UNO begrenzt. Der Button Kupplung Öffnen und Schließen aktuieren den Kupplungsaktuator direkt auf die Endpositionen definiert im Arduino Quellcode. Der Button Einkuppelvorgang schließt die Kupplung wie bei in Abbildung 9-14 zu sehen.

Die Feineinstellung für dein Einkuppelvorgang erfolgt in den folgenden Schritten:

1. Voreinstellung:
 - a) Kupplung öffnen mit Kupplung Öffnen Button.
 - b) RLS alpha aktivieren.
 - c) Die VKM manuell mit „VKM Start“ starten. Da die Kupplung bei der Positionsbegrenzung geöffnet ist, muss die Bremsendrehzahl gleich Null bleiben und die VKM im Leerlauf verweilen. Falls dies nicht der Fall ist, dann muss die Positionsbegrenzung im Arduino Quellcode erhöht werden.
2. Der Einkuppelstartwert wird wie folgt bestimmt:
 - a) Bei geöffneter Kupplung, Kupplung_Status in 0,05 bzw. in 0,02 Schritten reduzieren und auf die Bremsendrehzahl achten.
 - b) Wenn diese sich mitdrehen beginnt, den Wert der Kupplung_Status notieren.
 - c) Kupplung wieder öffnen, die Bremsendrehzahl mit der Regelstrategie n/α wieder auf Null setzen.
 - d) Diesen Vorgang wiederholen bis der genaue Grenzwert gefunden ist.
 - e) Diesen Grenzwert plus 0,02 in das Feld Einkuppelstartwert in Abbildung 9-13 eintragen.
3. Der Einkuppelrutschwert wird wie folgt bestimmt:
 - a) Kupplung öffnen.
 - b) Prüfstandsregelung auf RLS alpha stellen.
 - c) Die VKM im Leerlauf lassen und die Variable Kupplung_Status verkleinern bis sich die Motordrehzahl soweit senkt, sodass die VKM beinahe abstirbt. Die Reduzierung der Variable Kupplung_Status sollte zügig stattfinden. Den Wert der Kupplung_Status notieren.
 - Beim Einkuppelvorgang wird auch die Drosselklappe geöffnet, damit der Motor in einem angemessenen Drehzahlbereich den Einkuppelvorgang durchführen kann. Daher ist auch sichergestellt, dass die VKM nicht absterben kann.
 - d) Dieser Vorgang muss des Öfteren wiederholt werden, um ein Gefühl für den Wert der Kupplung_Status für die Rutschposition zu erhalten. Ziel ist, dass die Kupplung bei voll geöffneter Position und direkten Ansteuern des Elektromotors auf die Rutschposition die VKM nicht abstirbt. Daher kann der Motor während eines vollautomatisierten Fahrzyklus nicht absterben.
4. Alpha Einkuppeln Start und Ende sowie die Rutschzeit und Einlaufzeit werden

empirisch bestimmt. Alpha Einkuppeln Start und Ende können mit 5% und 10% zu Beginn angenommen werden und die Rutschzeit kann mit 3 Sekunden und die Einlaufzeit mit 1 Sekunde angenommen werden. Diese Werte müssen angepasst werden. Dabei können folgende Richtwerte angenommen werden:

- a) Die Motordrehzahl soll während des gesamten Einkuppelvorgangs zwischen 2000 und 4000 U/min liegen.
- b) Die Rutschzeit soll möglichst kurz sein. Beim mehrmaligen Testen sollten 2 Sekunden möglich sein.
- c) Die Einlaufzeit soll kurz sein, jedoch soll der Einkuppelvorgang ruckfrei sein. Erfahrungswerte zeigten, dass eine kürzere Einlaufzeit als 0,6 Sekunden zu einem ruckartigen Einkuppeln führt.

5. Nun wird der Einkuppelvorgang getestet

- a) Kupplung öffnen.
- b) Motor starten.
- c) Regelstrategie auf RLS alpha stellen.
- d) Nun über den Button Einkuppelvorgang im Kupplungsfenster den Einkuppelvorgang starten.
- e) Den Einkuppelvorgang beobachten und die sechs Variablen für das Einkuppeln variieren bis der Einkuppelvorgang „flüssig“ ist.
- f) Die Punkte a bis e werden wiederholt, bis sich die Motordrehzahl beim Einkuppeln in einem angemessenen Bereich befindet. Das kann für einen Harley-Davidson Motor eine andere sein als bei einem Ducati Panigale Motor. Ein angenehmer Einkuppelvorgang soll akustisch abgeschätzt werden.

Nach erfolgter Einstellung der Kupplungswerte können diese ebenfalls beim Abfahren während der Fahrzyklen live in Abbildung 9-13 verändert werden.

Angemerkt wird hier, dass mehrere Versionen für das Einkuppeln getestet wurden. Zum Beispiel wurde ein PID-Kupplungsregler für den Einkuppelvorgang getestet, jedoch zeigte dieser aufgrund der sehr feinen Positionieranforderung nicht ausreichend zufriedenstellende Ergebnisse. Hierfür wäre eine hochgenaue Positionsbestimmung des Elektromotors notwendig.

Zusätzlich zeigte sich, dass die getesteten Versionen wesentlich komplizierter einzustellen waren und dennoch kaum zufriedenstellende Ergebnisse lieferten. Daher wurde das relativ einfache Konzept mit den sechs wichtigsten Variablen, dargestellt in Abbildung 9-13, ausgewählt, wobei dieses ebenfalls die besten Ergebnisse lieferte.

9.1.3 Schritt 3: PID-Geschwindigkeitsregler Einstellung

Die PID-Anteile für die Geschwindigkeitsregelung können im Tornado RT live unter Fenster > Geschwindigkeitsregelung verändert werden (Abbildung 9-15). Der Regler kann ebenfalls manuell in diesem Fenster aktiviert (=1) bzw. deaktiviert (=0) werden.

Geschwindigkeitsregelung	
Geschwindigkeitsregelung Aktivierung: Ein=1, Aus=0	
	0
GeschwRegelung_P	3,000
GeschwRegelung_I	0,500
GeschwRegelung_D	0,000

Abbildung 9-15: Aktivierung Geschwindigkeitsregelung und Einstellung PID-Anteile

Die PID-Werte für die Geschwindigkeitsregelung liegen in der folgenden Größenordnung:

$$1 \leq \text{GeschwRegelung } P = k_p \leq 5 \quad (9-3)$$

$$0,1 \leq \text{GeschwRegelung } I = k_i \leq 2 \quad (9-4)$$

$$0 \leq \text{GeschwRegelung } D = k_d \leq 0,5 \quad (9-5)$$

Diese Werte wurden empirisch ermittelt.

Aufgrund von den zeitlichen Rahmenbedingungen konnten die empirischen Reglerentwurfsmodelle von Kapitel (2.3.5) nicht mehr getestet werden. Es besteht auch die Möglichkeit, für niedrige, mittlere und hohe Geschwindigkeit unterschiedliche PID-Anteile zu verwenden. Folgende Werte könnten dabei verwendet werden:

$$v \leq 40 \frac{\text{km}}{\text{h}}: k_p = 3, k_i = 0,5, k_d = 0 \quad (9-6)$$

$$40 \leq v \leq 80 \frac{\text{km}}{\text{h}}: k_p = 4, k_i = 1, k_d = 0 \quad (9-7)$$

$$80 \leq v \leq 150 \frac{\text{km}}{\text{h}}: k_p = 5, k_i = 1,5, k_d = 0 \quad (9-8)$$

Für konstante PID-Werte beim Abfahren eines WMTCs wurden folgende Werte empirisch als optimal eruiert:

$$k_p = 4, \quad k_i = 0,5, \quad k_d = 0 \quad (9-9)$$

Die PID-Anteile in (9-9) wurden ebenfalls für Kapitel 6 Verifizierung vollautomatisierter Prüfstand verwendet.

9.1.4 Schritt 4: Emulierte mechanische Bremse Einstellung

Die emulierte mechanische Bremse wird aktiviert, wenn die Variable `v_Bremse_Limit` von der aktuellen Geschwindigkeit überschritten wird. Daraufhin wird eine Steigung für das Motorrad vorgegeben um dieses zu Bremsen. Falls die Bremskraft für den Fahrzyklus nicht ausreichend ist, kann die statische Steigungsvorgabe beim Übertreten des Limits erhöht werden. Diese kann in Befehlsprogramme > Automatisierung > Bremse in Zeile 3 und 8 in Abbildung 9-27 verändert werden.

9.2 Durchführen eines vollautomatisierten Prüflauf

Diese Anleitung dient dazu, dass nach vollständigem Aufbau der Vollautomatisierung der Prüfstand vollautomatisiert transiente Fahrzyklen abfährt.

Dabei ist wie folgt vorzugehen:

1. Den Prüfstand soweit in Betrieb setzen wie dies bei üblichem stationärem Betrieb notwendig ist. Das heißt, alle Kühlkreisläufe aktivieren, Kraftstoff aufdrehen, etc.
2. Die Spannungsversorgung für den Kupplungsaktuator einschalten bzw. einstecken.
3. Das gewünschte Prüfprogramm öffnen und starten. Nun etwa zwei Sekunden warten, dann wird im Fahrerleitsystem der Fahrzyklus dargestellt und der Prüfstand beginnt diesen vollautomatisiert abzufahren.
4. Es ist zu empfehlen, dass während den ersten Fahrzyklustest besonders das Verhalten des Gesamtprüfstandes beobachtet wird. Die Kupplungsvariablen und die Geschwindigkeitsregler Parameter können live verändert werden. Sobald der Fahrzyklus erfolgreich einen Zyklus vollautomatisiert absolviert hat, kann der Prüfling optimiert werden.

9.3 Gesamtsystem und Teilsysteme inklusiver Variablen und Regelgrößen

Grundsätzlich hat der Prüfstand zwei Hauptrechner. Diese sind der ADAC und der Tornado Rechner. Der Unterschied liegt darin, dass zeitkritische Funktionen im ADAC in Echtzeit ausgeführt werden und im Tornado Rechner alle „langsamen“ Vorgänge berechnet werden. In Abbildung 9-16 sind die Teilfunktionen dargestellt, wobei rote „Kästchen“ Funktionen die im ADAC ausgeführt werden und blaue „Kästchen“ im Tornado Rechner ausgeführt werden.

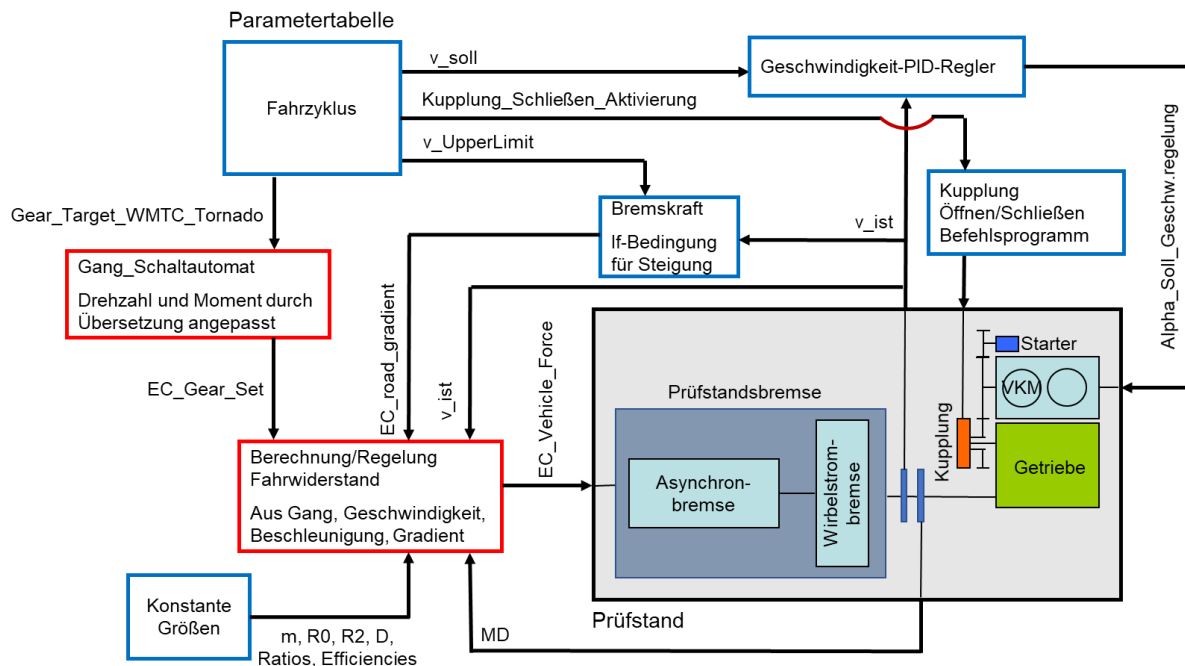


Abbildung 9-16: Schematische Funktionsdarstellung Gesamtprüfstand

Variablen, Ein-/Ausgänge, Regelungsgrößen und Stellgrößen sind anhand der folgenden Teilsysteme im Detail aufgelistet und dargestellt.

9.3.1 Variablen

Die Variablen für die Vollautomatisierung wurden unter Variablen > Automatisierung angelegt. Auf andere Variablen, die bereits vorhanden sind, wurde zugegriffen.

Name	Gruppe	Beschreibung
Kupplung_Zeit	Automatisierung	Einkuppelzeit
Kupplung_Schliessen_Aktivierung	Automatisierung	Aktiviert Befehlsprogramm für das Einkuppeln
GeschwRegelung_P	Automatisierung	P Anteil Regler
Geschwindigkeitsregelung_Aktivierung	Automatisierung	Geschwindigkeitsregelung_Aktivierung
v_Bremse_Limit	Automatisierung	Geschwindigkeitslimit für die Bremsenaktivierung
Kupplung_Befehlsprogramm_Zeit	Automatisierung	Gesamtzeit für den Einkuppelvorgang
Kupplung_Startwert	Automatisierung	Einkuppelstartwert
Alpha_Einkuppeln_Ende	Automatisierung	Alpha bei Ende des Einkuppelvorgangs
Alpha_Einkuppeln_Start	Automatisierung	Alpha bei Beginn des Einkuppelvorgangs
RLSv_Alias_aktivieren	Automatisierung	RLSv_Alias_aktivieren
Alpha_Soll_Automatisierung	Automatisierung	Alpha_Soll_Geschwindigkeitsregelung
GeschwRegelung_D	Automatisierung	D Anteil Regler
GeschwRegelung_I	Automatisierung	I Anteil Regler
Kupplung_Endwert	Automatisierung	Einkuppelendwert
Kupplung_Status	Automatisierung	0 = Kupplung geschlossen, bei 1 = offen
Kupplung_Einlaufzeit	Automatisierung	Einlaufzeit
Kupplung_Rutschzeit	Automatisierung	Zeit in Kupplungs-Rutschposition
Kupplung_Rutschwert	Automatisierung	Wert bei dem die Kupplung rutscht
Alpha_Geschwindigkeitsregelung	Automatisierung	Alpha_Geschwindigkeitsregelung

Abbildung 9-17: Variablen Automatisierung

Unter den Befehlsprogrammen wurde ein Ordner mit dem Namen Automatisierung angelegt, dieser beinhaltet die Befehlsprogramme Bremsen, Kupplung_Oeffnen, Kupplung_Schließen und Kupplung_Schließen_Einfach.

9.3.2 Kupplung Öffnen/Schließen Befehlsprogramme

Befehlsprogramm Kupplung_Oeffnen

Das Befehlsprogramm Kupplung_Oeffnen hat eine Aktivierungsbedingung und eine Abbruchbedingung.

Aktiviert wird das Befehlsprogramm, wenn die Gangvorgabe aus der Parametertabelle einen kleineren Wert wie 0,5 annimmt (Abbildung 9-18).

Eigenschaften				
Abbruchbedingungen				
Aktivierungsbedingungen				
Aktivierungsvariable	Operator	Vergleichswert	Dauer	
Gear_Target_WMTC	<	0,5	0	

Abbildung 9-18: Aktivierungsbedingung Kupplung_Oeffnen

Abgebrochen wird das Befehlsprogramm, wenn die Variable Kupplung_Schließen_Aktivierung größer 0,5 ist (Abbildung 9-19), da in diesem Fall die Kupplung geschlossen werden soll.

Eigenschaften				
Abbruchbedingungen				
Aktivierungsbedingungen				
Aktivierungsvariable	Operator	Vergleichswert	Dauer	
Kupplung_Schliessen_Aktivierung	>	0,5	0	

Abbildung 9-19: Abbruchbedingung Kupplung_Oeffnen

Das Befehlsprogramm weist der Variable Kupplung_Status den Wert 1 zu. Dieser entspricht der voll geöffneten Kupplung. Weiters wird die Geschwindigkeitsregelung deaktiviert und die Drosselklappe auf 0% gestellt (Abbildung 9-20).

Zeile	Befehl	Parameter
1	Zuweisung	Kupplung_Status = 1 -
2	Zuweisung	Geschwindigkeitsregelung_Aktivierung = 0 -
3	Zuweisung	Alpha_Soll_Automatisierung = 0 -
4		

Abbildung 9-20: Befehlsprogramm Kupplung_Oeffnen

Befehlsprogramm Kupplung_Schließen_Einfach

Bei diesem Befehlsprogramm wird die VKM mit dem Antriebsstrang schnellstmöglich geschlossen.

Befehlsprogramm Kupplung_Schließen, der definierte Einkuppelvorgang

Dieses Befehlsprogramm dient für den vorgegebenen definierten Einkuppelvorgang, welcher über die Kupplungsvariablen in Abbildung 9-13 eingestellt wird.

Aktiviert wird dieses Befehlsprogramm über die Kupplung_Schließen_Aktivierung Variable (Abbildung 9-21).

Eigenschaften				Abbruchbedingungen				Aktivierungsbedingungen			
Aktivierungsvariable		Operator		Vergleichswert		Dauer					
Kupplung_Schliessen_Akt		>		0,5		0					

Abbildung 9-21: Aktivierungsbedingung Kupplung_Schließen

Das Befehlsprogramm deaktiviert die Geschwindigkeitsregelung, durchläuft den Betriebspunkt, setzt die Kupplung auf die Positionsbegrenzung und aktiviert die Geschwindigkeitsregelung nach dem Einkuppeln wieder (Abbildung 9-22).

Zeile	Befehl	Parameter
1	Zuweisung	Geschwindigkeitsregelung_Aktivierung = 0 -
2	Betriebspunkt	Kupplung_Zeit; keine Messung; Kupplung_Status=Kupplung_Startwert;Kupplung_Ruts...
3	Zuweisung	Kupplung_Status = 0 -
4	Zuweisung	Geschwindigkeitsregelung_Aktivierung = 1 -

Abbildung 9-22: Befehlsprogramm Kupplung_Schließen

Der Betriebspunkt ist eine Funktion in der Tornado Prüfstandssteuerung, bei der man zeitabhängig bzw. bedingungsabhängig Werte zuweisen kann. Der Betriebspunkt ist so eingestellt, dass in 0,05s Schritten der Variable Kupplung_Status ein neuer Wert zugeordnet wird. Zeitgleich wird die Drosselklappenstellung vorgegeben (Abbildung 9-23).

Die Variable Kupplung_Zeit ist die Summe aus der Kupplung_Einlaufzeit und der Kupplung_Rutschzeit. Der zeitliche Ablauf ist in Abbildung 9-14 dargestellt.

Betriebspunkt					
Eigenschaften					
Einschwingverhalten					
Sollwerte					
Bedingte Sollwerte					
Ze...	Startzeit	Sollwert-Variable	Startwert	Endwert	Rampenzeit
BP...	[hh:mm:ss]				[hh:mm:ss]
<input type="checkbox"/>	0	Kupplung_Status	Kupplung_Startwert	Kupplung_Rutschwert	Kupplung_Einlaufzeit
<input type="checkbox"/>	Kupplung_Einlaufzeit	Kupplung_Status	Kupplung_Rutschwert	Kupplung_Rutschwert	Kupplung_Rutschzeit
<input type="checkbox"/>	0	Alpha_Soll_Automatisierung	Alpha_Einkuppeln_Start	Alpha_Einkuppeln_Ende	Kupplung_Zeit

Abbildung 9-23: Betriebspunkt Kupplung_Schließen

9.3.3 PID-Geschwindigkeitsregler

Der Geschwindigkeitsregler wird über die Variable Geschwindigkeitsregelung_Aktivierung aktiviert (=1) bzw. deaktiviert (=0). Der Stellausgang ist der Variable Alpha_Geschwindigkeitsregelung zugeordnet und diese Variable ist die Vorgabe für den Drosselklappenregler. Der Regler könnte in Echtzeit (im ADAC) ausgeführt werden, jedoch haben Erfahrungswerte gezeigt, dass es zwischen dem ADAC und dem Tornado Rechner zu keinen Performanceunterschieden kommt.

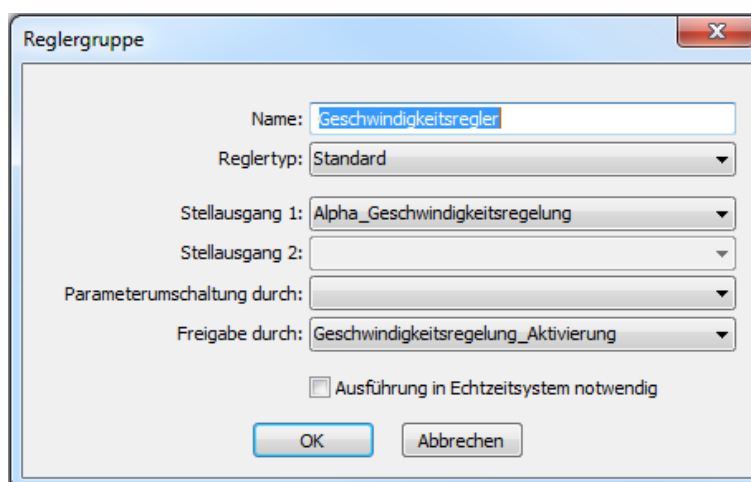


Abbildung 9-24: Geschwindigkeitsregler Stellausgang und Freigabe

In der Reglergruppe (Abbildung 9-24) wird ein spezifischer Regler erstellt (Abbildung 9-25).

Name	ID	Sollwert	Istwert	kP	kl	kD
v Regler	1	v_soll	EC_velocity	GeschwRegelung_P	GeschwRegelung_I	GeschwRegelung_D

Abbildung 9-25: v Regler Variablen Zuordnung

Dabei ist die Soll-Geschwindigkeit mit v_soll aus der Parametertabelle vorgegeben. Der Istwert wird aus der gemessenen Drehzahl, Übersetzungen und Reifendurchmesser berechnet. Den PID-Anteilen wird jeweils eine Variable zugeordnet, die in Abbildung 9-15 eingestellt werden.

9.3.4 Emulierte mechanische Bremse Befehlsprogramm

Die emulierte mechanische Bremse wird aktiviert, wenn EC_PT_v_Aktivierung größer 1 ist (Abbildung 9-26). Diese Variable wird auf 1 gesetzt, wenn das Prüfprogramm gestartet wird (Abbildung 9-4, Zeile 14).

Eigenschaften				Abbruchbedingungen				Aktivierungsbedingungen				
Aktivierungsvariable		Operator	Vergleichswert		Dauer							
EC_PT_v_Aktivierung		>	0,5		0							

Abbildung 9-26: Aktivierung emulierte mechanische Bremse

In Abbildung 9-27 sind die if-Abfragen für das emulierte mechanische Bremsen dargestellt. In Zeile 1 wird die aktuelle Geschwindigkeit mit der Variable v_Bremse_Limit verglichen. v_Bremse_Limit ist die Summe aus der oberen Geschwindigkeitsgrenze v_UpperLimit aus der Parametertabelle und der Konstante -1. Die Konstante kann in der Berechnung auf einen anderen gewünschten Wert verändert werden. Der Gradient wird auf 15% gestellt, wenn die Geschwindigkeit die obere Geschwindigkeitsgrenze überschreitet und wenn die Drosselklappenstellung kleiner 1% ist. Wenn die Kupplung geöffnet wird bzw. wenn die Variable Kupplung_Status auf 1 gesetzt wird und das Motorrad nicht im Stillstand ist, dann wird ein Gradient von 18% vorgegeben.

Zurückgesetzt wird der Gradient wenn die umgekehrten Bedingungen, zu den vorhin genannten, erfüllt sind.

Zeile	Befehl	Parameter
1	If	EC_velocity >= v_Bremse_Limit
2	If	Alpha_Soll_Automatisierung <= 1 -
3	Zuweisung	EC_road_gradient = 15 %
4	Endif	
5	Endif	
6	If	Kupplung_Status > 0,95 -
7	If	EC_velocity >= 0,5 km/h
8	Zuweisung	EC_road_gradient = 18 %
9	Endif	
10	Endif	
11	If	EC_velocity < v_Bremse_Limit
12	If	Alpha_Soll_Automatisierung > 1 -
13	Zuweisung	EC_road_gradient = 0 %
14	Endif	
15	Endif	
16	If	EC_velocity < 0,5 km/h
17	Zuweisung	EC_road_gradient = 0 %
18	Endif	
19		

Abbildung 9-27: Befehlsprogramm emulierte mechanische Bremse

Vorteilhaft bei dieser Methode ist, dass die Verzögerung konstant ist und unabhängig der Motorradmasse ist, da beispielsweise bei einem WMTC Fahrzyklus die Geschwindigkeitsvorgabe für unterschiedliche Motorradmassen dieselbe ist. Daher muss der Gradient nur einmal für einen Zyklus angepasst werden und unterschiedliche Motorradmassen verhalten sich ähnlich. Ebenfalls kann ein Gradient in Abhängigkeit einer Funktion vorgegeben werden, jedoch haben Erfahrungswerte gezeigt, dass eine konstante Vorgabe ausreichend ist.

9.3.5 Andere Funktionen

Alle anderen Funktionen von Abbildung 9-16 werden in Kapitel 4 Bestehender Automatisierungsgrad, in den beiden Dokumentationen „Tornado Designer – Benutzerhandbuch“ und/oder „Tornado RT – Benutzerhandbuch“ erklärt.

9.4 Arduino Uno Quellcode

Das vorliegende Arduino Skript dient zur Ansteuerung des Kupplungsaktuators mittels des Elektromotors über den Motorcontroller. Angesteuert wird der Arduino über ein analoges Spannungsausgangssignal von der Sensorbox mit Tornado.

// Dieses Skript dient zur Ansteuerung des Kupplungsaktuators über einen Motorcontroller

```
int analogPin0 = A0;
int analogPin1 = A1;
int Poti_Ist = 0; // 16 bit value
int Poti_Soll = 0;
int Min_Poti_Position = 2000; // Dieser Wert muss eingestellt werden
int Max_Poti_Position = 2700; // Dieser Wert muss eingestellt werden

int PWM_out_pin_Schliessen = 5;
int PWM_out_pin_Oeffnen = 10;
```

```
void setup() {
Serial.begin(9600);
pinMode(PWM_out_pin_Oeffnen, OUTPUT);
pinMode(PWM_out_pin_Schliessen, OUTPUT);
}

void loop() {
byte PWM_out_level_Schliessen;

Poti_Soll = analogRead(analogPin0)*5/1.023;
Poti_Ist = analogRead(analogPin1)*5/1.023;

// Die aktuellen Werte des Potentiometers sowie der Vorgabe werden im "Serial Monitor"
angezeigt wenn ein PC an den Arduino angeschlossen ist
Serial.println("Poti_Ist: ");
Serial.println(Poti_Ist);
Serial.println("Poti_Soll: ");
Serial.println(Poti_Soll);

// Stillstand Elektrozyylinder und Weglimitierung vermeiden
if ((abs(Poti_Soll - Poti_Ist) < 10) || (Poti_Ist < Min_Poti_Position && Poti_Soll <
Min_Poti_Position) || (Poti_Ist > Max_Poti_Position && Poti_Soll > Max_Poti_Position))
{
analogWrite(PWM_out_pin_Oeffnen, 0);
analogWrite(PWM_out_pin_Schliessen, 0);
}

else
{
if (Poti_Soll - Poti_Ist > 10) // Öffnen
{
analogWrite(PWM_out_pin_Schliessen, 0);
analogWrite(PWM_out_pin_Oeffnen, 255);
}
if (Poti_Soll - Poti_Ist < -10) // Schließen
{
analogWrite(PWM_out_pin_Oeffnen, 0);
PWM_out_level_Schliessen = abs(Poti_Soll - Poti_Ist)/10;
analogWrite(PWM_out_pin_Schliessen, PWM_out_level_Schliessen);
}
}
}
```

9.5 Prüfling BMW G 650 GS 2 Spark

Der Verbrennungsmotor, der auf dem Motorprüfstand montiert ist, ist ein Serienmotor von einer BMW G 650 GS mit teilweisen veränderten Teilen sowie einer „offenen“ ECU.

BMW G 650 GS	
Motorleistung	44 kW (60 PS) bei 7.500 U/min
Hubraum	652 ccm
Kupplung	Mehrscheibenkupplung im Ölbad, mechanisch betätigt
Getriebe	Klauengeschaltetes Fünfgang-Getriebe im Motorgehäuse integriert
Höchstgeschwindigkeit	ca. 170 km/h
Fahrzeugmasse	192 kg (fahrfertig, vollgetankt)

Tabelle 9-1: Technische Daten BMW G 650 GS
(Quelle: Homepage BMW Motorrad, 14.10.2015)

Literatur

- [1] G. Singer, “Emulation of Transient Drive Cycles & Profiles on the Engine Test Bench for Conventional Motorcycles,” 2018.
- [2] R. McGee *et al.*, “A Case Study in Hardware-In-the-Loop Testing: Development of an ECU for a Hybrid Electric Vehicle,” *SAE Tech. Pap. Ser.*, vol. 1, no. 724, 2004.
- [3] R. Kreß, “Robuste Fehlerdiagnoseverfahren zur Wartung und Serienabnahme elektrohydraulischer Aktuatoren,” Technische Universität Darmstadt, 2002.
- [4] S. Djinic, N. Miljic, and S. J. Popovic, “Software and Hardware Challenges of Engine Test Bed Automation-Example Software and Hardware Challenges of Engine Test Bed Automation,” no. December 2017, 2015.
- [5] W. Haager, *Regelungstechnik*, 2. Auflage. öbv&hpt, 2003.
- [6] J. Lunze, *Regelungstechnik 1*, 7. Auflage. Springer, 2008.
- [7] P. Böttle, G. Boy, and H. Clausing, *Elektrische Mess- und Regelungstechnik*, 11. überar. Vogel Fachbuch, 2011.
- [8] P. Busch, *Elementare Regelungstechnik*, 8. Auflage. Vogel Fachbuch, 2012.
- [9] U. Tietze, C. Schenk, and E. Gamm, *Halbleiter-Schaltungstechnik*, 15. Auflag. Springer Vieweg, 2016.
- [10] J. Lunze, *Regelungstechnik 2*, 4. Auflage. Springer, 2006.
- [11] G. Ellis, *Control System - Design Guideline*, 4th Editio. Elsevier, 2012.
- [12] F. Tröster, *Regelungs- und Steuerungstechnik für Ingenieure*, 4. Auflage. 2015.
- [13] W. Schneider, *Praktische Regelungstechnik*, 3. Auflage. Vieweg+Teubner, 2008.
- [14] B. Steffenhagen, *Kleine Formelsammlung Regelungstechnik*. Fachbuchverlag Leipzig, 2011.
- [15] W. Schumacher and M. Maurer, *Grundlagen der Regelungstechnik*. 2014.
- [16] E. Parliament, “Commission Delegated Regulation EU,” *No 134*, pp. 1–327, 2014.
- [17] K. Engineers, “Prüfstandsautomatisierung.” [Online]. Available: <https://www.ksengineers.at/de/Automotive-Testing/Management-und-Automation-Tools/Tornado-Test-Bed>. [Accessed: 05-Mar-2019].
- [18] “Motorrad Sekundärtrieb.” [Online]. Available: <https://www.motorradonline.de/schraubertipps/sekundaerantrieb-allgemeines.41260.html>. [Accessed: 11-Mar-2019].
- [19] “Arduino Uno.” [Online]. Available: <https://store.arduino.cc/arduino-uno-rev3>. [Accessed: 03-Mar-2019].
- [20] Infineon, “BTS 7960 - High Current PN Half Bridge NovalithIC,” 2004.
- [21] “Super300 Servomotor.” [Online]. Available: <https://www.amazon.co.uk/Happymodel-SUPER300-Torque-300kg-Cm-Degree/dp/B01HM0A668>. [Accessed: 03-Mar-2019].
- [22] R. Hessmer, “IBT-2 H-Bridge with Arduino.” [Online]. Available: <http://www.hessmer.org/blog/2013/12/28/ibt-2-h-bridge-with-arduino/>. [Accessed: 03-Mar-2019].
- [23] D. Mietke, “Steuerung und Regelung.” [Online]. Available: <https://elektroniktutor.de/regelungstechnik/regelung.html>. [Accessed: 04-Mar-2019].

- [24] P. Rieger, “Emulation von transienten Fahrzyklen am Motorprüfstand Masterarbeit Emulation von hybriden Antriebssträngen am Motorprüfstand,” vol. 43, no. 316, pp. 1–2.