

Masterarbeit

Einbindung der elektrischen Force-Feedback Komponenten in einen Fahrsimulator

Matthias Pirstinger, BSc

Institut für Technische Informatik
Technische Universität Graz

Vorstand: Univ.-Prof. Dipl.-Inform. Dr.sc.ETH Kay Uwe Römer



Begutachter: Ao.Univ.-Doz. Dipl.-Ing. Dr. techn. Eugen Brenner
Graz, im Oktober 2014

Kurzfassung

Im Zuge des “*Muegen Driving* Projekts“ wurde am Institut für Fahrzeugtechnik Graz (FTG) ein Fahrsimulator konstruiert. Hierfür wurde am Institut eine State-of-the-Art Recherche durchgeführt, um den Fahrsimulator nach den neuesten technischen Standards auszuführen. Der Fahrsimulator bietet sich dahingehend als einzigartiges Forschungsinstrument an, als dass man die Fahrerin bzw. den Fahrer in eine möglichst wirklichkeitsnahe Umgebung bringt. Dadurch können auch gefährliche Szenarien wie Unfälle bzw. Beinahe-Unfälle simuliert werden, ohne die Fahrerin bzw. den Fahrer in Gefahr zu bringen. Um der Fahrerin bzw. dem Fahrer das Gefühl einer realitätsnahen Umgebung zu geben, ist es wichtig viele Aspekte der Wahrnehmung anzusprechen. Daher wurde als Plattform für den Simulator ein reales Fahrzeug (Mini Countryman) verwendet.

Außer dem visuellen Eindruck können auch andere Sinneseindrücke Informationen über den Fahrzeugzustand enthalten. Um diese Sinneseindrücke zu generieren wird im Fahrsimulator Force Feedback (FFB) verwendet. Unter diesem Zusammenhang versteht man im Fahrsimulator das Aufbringen von Kräften an Lenkrad, Ganghebel und Pedalerie um die Reaktion von realen Fahrzeugkomponenten zu simulieren. Um für verschiedene Fahrzeugkomponenten wie etwa der Bremshydraulik auch unterschiedliche Wirkungskennlinien simulieren zu können, werden in modernen Fahrsimulatoren elektronische Aktuatorssysteme verwendet. Diese Masterarbeit beschäftigt sich mit der State-of-the-Art Recherche sowie der Auswahl und Implementierung dieser FFB-Systeme.

Das Hauptaugenmerk liegt dabei auf der Bremshydraulik. Hier wird die Auswahl des Systems und die Umsetzung mittels Motor-Controllersystem, sowie die Anbindung des Systems mit Hilfe des CANopen Bussystems beschrieben. Zusätzlich werden die grundsätzlichen Funktionen von CANopen betrachtet.

Abstract

In the scope of the “*Muegen Driving Project*“ a driving simulator was developed at the Institute of Automotive Engineering Graz (FTG). A state-of-the-art research was performed to build the driving simulator according to the latest technical standards. The driving simulator lends is a unique research instrument, allowing the driver to operate in a realistic environment. Moreover the driver can be put in dangerous scenarios such as accidents or near accidents without putting him/her in danger. To give the driver the feeling of a realistic environment, it is important to feed the aspects of perception. Therefore, a real vehicle (Mini Countryman) was used as a platform for the simulator.

In addition to the visual impression, other sensations can contain information about the vehicle states. To generate these sensations in the driving simulator the approach of Force-Feedback is used. According to a driving simulator this covers the application of forces on the steering wheel, gear lever and pedals. This simulates the reaction of the real vehicle components. In order to simulate different characteristics for vehicle components such as brake hydraulic, electronic actuator systems are used in modern driving simulators. This thesis deals with a state-of-the-art research, selection and implementation of the Force-Feedback systems.

The main focus is on the hydraulic brake system. Here the selection of the system and the implementation with a motor-controller system, as well as the connection of the system via CANopen bus is described. In addition, the basic functions of CANopen are explained.

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.

Graz, am 10.Oktober 2014

.....
(Unterschrift)

STATUTORY DECLARATION

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 marked all material which has been quoted either literally or by content from the used sources.

Graz, October 10th 2014

.....
(signature)

Danksagung

Ohne die Mithilfe und Unterstützung einiger wichtiger Personen, wäre es nicht möglich gewesen, diese Masterarbeit zu verfassen. Deshalb möchte ich mich hier bei diesen Menschen besonders bedanken.

Als aller erstes möchte ich mich bei meiner Familie und meiner Freundin bedanken, besonders bei meinen Eltern, die mich seit meiner Kindheit in all meinen Bestrebungen unterstützt haben. Damit ist nicht nur die finanzielle Unterstützung gemeint, sondern auch der Rückhalt in allen Lebenslagen. Auch meine Freundin hat mich immer wieder in meinen Vorhaben bekräftigt und mir in lernintensiven Wochen genügend Zeit gegeben.

Weiters ist es mir wichtig, meinem Freundeskreis für seinen Rückhalt während meines gesamten Studiums zu danken. Vor allem die Gruppe von Schulfreunden, mit denen ich mein Studium begonnen habe, war eine große Hilfe während des gesamten Studiums.

Die vorliegende Masterarbeit ist am Institut für technische Informatik in Zusammenarbeit mit dem Institut für Fahrzeugtechnik Graz entstanden. Dabei möchte ich mich besonders bei meinen Betreuern Herrn Dipl.-Ing. Dr. Eugen Brenner und Herrn Dipl.-Ing. Dr. Arno Eichberger für die wertvolle Unterstützung während dieser Masterarbeit bedanken. Weiters möchte ich mich auch bei den weiteren Kolleginnen und Kollegen des Fahr Simulator Projektteams für die tolle Zusammenarbeit bedanken.

Graz, im Oktober 2014

Matthias Pirstinger

Inhaltsverzeichnis

1	Einleitung	1
1.1	Forschungsprojekt <i>Muegen Driving</i>	1
1.2	Fahrsimulator am Institut für Fahrzeugtechnik (FTG)	1
1.3	Force Feedback	2
1.4	Aufbau der Arbeit	3
2	Fahrsimulatoren - State of the Art	4
2.1	Einsatzgebiete	4
2.2	Ausführungsarten	6
2.2.1	PC System	6
2.2.2	Fixed Base Simulator	6
2.2.3	Motion Base Simulator	7
2.3	Motion Base Simulator des DLR	7
3	Fahrsimulator des FTG	9
3.1	Einleitung	9
3.2	Bremspedal	9
3.2.1	Anforderungen	9
3.2.2	Lösungsvarianten	11
3.2.3	Aktuatorwahl	13
3.2.4	Aktuatorsystem mit rotatorischem Motor	13
3.2.4.1	CANopen	14
3.2.4.2	Konfigurierung Controller CD1-k	18
3.2.4.3	CANopen Verbindung mittels Matlab Vehicle Network Tool- box VNT	20
3.2.4.4	Starten des Controllerbetriebs mittels CANopen	24
3.2.4.5	Aktivieren des Stromreglers mittels CANopen	27
3.2.4.6	Vorgabe des Motordrehmoments	30
3.2.4.7	Deaktivierung des Stromreglers mittels CANopen	31

3.2.5	Diskussion und Ausblick	32
3.3	Kupplungspedal	33
3.3.1	Einleitung	33
3.3.2	Positionsmessung	33
3.3.3	Konzept für Kupplungspositionsmessung	35
3.3.4	Diskussion und Ausblick	35
3.4	Gaspedal	37
3.4.1	Einleitung	37
3.4.2	Elektronisches Gaspedal (Mini Countryman)	37
3.4.3	Messung des Positionssignals	38
3.4.4	Einbindung in die Simulationsumgebung	39
3.4.5	Diskussion und Ausblick	40
3.5	Gangwahlhebel	42
3.5.1	Einleitung	42
3.5.2	Konzept zur Wahlhebeldetektion	42
3.5.3	Wahlhebelmodul des Mini Countryman	44
3.5.4	Einbindung in die Simulationsumgebung	46
3.5.5	Diskussion und Ausblick	46
3.6	Lenkrad	48
3.6.1	Einleitung	48
3.6.2	Lenkradaktuatorsystem SBW	48
3.6.3	Schaltungsaufbau mit Sicherheitseinrichtungen	48
3.6.4	Einbindung in die Simulationsumgebung	50
3.6.5	Vorgabe des Drehmoments	51
3.6.6	Diskussion und Ausblick	52
4	Zusammenfassung	54
5	Ausblick	57
A	Abkürzungsverzeichnis	59
B	Matlab Startup File (Startup.m)	61
	Bibliography	64

Kapitel 1

Einleitung

1.1 Forschungsprojekt *Muegen Driving*

Der Großteil aller Verkehrsunfälle mit Personenschaden lässt sich auf menschliches Fehlverhalten durch LenkerInnen von Personenkraftwagen zurückführen. Die Art des Fehlverhaltens weist hierbei auf eine deutliche Abhängigkeit von Alter und Geschlecht der Lenkerin bzw. des Lenkers. Diese Aspekte weisen sowohl auf eine generelle Notwendigkeit von Fahrerassistenzsystemen (FAS), als auch auf die verschiedenen Anforderungen an FAS hin. Weiters resultiert aus der Anpassung des FAS an die/den FahrerIn eine Senkung der Unfallschwere bis hin zur Vermeidung von Unfällen.

Im Forschungsprojekt *Muegen Driving* wird das Fahrerinnen und Fahrerverhalten in normalen und kritischen Fahrsituationen mit Hauptaugenmerk auf Geschlecht und Alter bei verschiedenen Straßenverhältnissen untersucht. Um die Tests reproduzierbar zu machen, wird der Fahrsimulator des Instituts für Fahrzeugtechnik verwendet. Hierzu muss den Probandinnen und Probanden die visuelle, akustische und kinästhetische Fahr- und Umgebungswahrnehmung ermöglicht werden. vgl. [1]

1.2 Fahrsimulator am Institut für Fahrzeugtechnik (FTG)

Im Zuge des *Muegen Driving* Projekts wurde am FTG ein Fahrsimulator mit einem Aufbau wie in Abbildung 1.1 ersichtlich konstruiert. Hierfür wurde am Institut eine State-of-the-Art Recherche durchgeführt, um den Fahrsimulator nach den neusten technischen Standards auszuführen. Wie in [2] bereits im Jahr 1990 gut ausgeführt, bietet sich der Fahrsimulator dahingehend als Forschungsinstrument an, als dass man die Fahrerin bzw. den Fahrer in eine möglichst wirklichkeitsnahe Umgebung bringt. Dadurch können auch gefährliche Szenarien wie Unfälle bzw. Beinahe-Unfälle simuliert werden, ohne die Fahrerin bzw. den Fahrer in Gefahr zu bringen. Ein weiterer Vorteil eines Fahrsimulators ist die

exakte Reproduzierbarkeit solcher Szenarien was wiederum zu einer exakten Auswertung der Versuche führt. Um für die Fahrerin bzw. den Fahrer das Gefühl einer realitätsnahen Umgebung zu geben ist es wichtig viele Aspekte der Wahrnehmung anzusprechen. Wichtig ist dabei auch den Fahrzeuginnenraum mit seiner Vielzahl an Instrumenten aufzubauen. Daher wurde als Plattform für den Simulator ein reales Fahrzeug (Mini Countryman) verwendet. Weiters ist es auch wichtig die Simulation der Umgebung mittels einer Visualisierung durchzuführen. Diese wird vom Projektpartner, dem Fraunhofer Institut, mit einem autostereokopen 3D-Visualisierungsverfahren (ohne Hilfsmittel wie einer 3D-Brille) erzeugt. Der nächste Aspekt ist die Fahrzeug- sowie auch die Umgebungsgeräuschkulisse, welche durch die Experten der Firma AVL implementiert wird. Als letzter Aspekt ist noch das Force Feedback (FFB) an den wichtigsten Berührungspunkten der Fahrerin bzw. des Fahrers mit dem Fahrzeug anzuführen. Dieses wird am FTG realisiert. Auf eine Bewegungsplattform wurde aufgrund der Anwendung im Bereich FAS, der hohen Kosten und des Integrationsaufwands vorerst verzichtet.

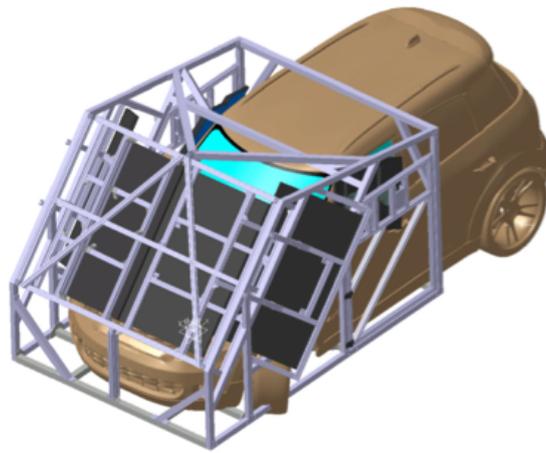


Abbildung 1.1: Aufbau Fahrersimulator des FTG

1.3 Force Feedback

In [3] wird beschrieben, dass außer dem visuellen Eindruck auch andere Sinneseindrücke Informationen über den Fahrzeugzustand enthalten können. Um diese Sinneseindrücke zu generieren wird im Fahrersimulator Force Feedback (FFB) verwendet. Unter diesem Zusammenhang versteht man im Fahrersimulator das Aufbringen von Kräften an Lenkrad, Ganghebel und Pedalerie um die Reaktion von realen Fahrzeugkomponenten zu simulieren. Um für verschiedene Fahrzeugkomponenten wie etwa der Bremshydraulik auch unterschiedliche Wirkungskennlinien simulieren zu können, werden in modernen Fahrersimulato-

ren elektronische Aktuatorsysteme verwendet. Diese Masterarbeit beschäftigt sich mit der State-of-the-Art Recherche sowie der Auswahl und Implementierung dieser FFB-Systeme.

1.4 Aufbau der Arbeit

Diese Masterarbeit teilt sich grob in drei Abschnitte. Im ersten Teil wird eine State-of-the-Art Recherche bezüglich Fahrsimulatoren durchgeführt. In diesem Abschnitt wird ein als State-of-the-Art geltender Fahrsimulator näher betrachtet. Im zweiten Teil wird der am FTG konstruierte Fahrsimulator betrachtet. Im Besonderen die Auswahl und Implementation der Force Feedback Komponenten. Das Hauptaugenmerk liegt im zweiten Teil auf dem FFB am Bremspedal. Im dritten und letzten Abschnitt wird dann die Arbeit zusammengefasst und ein Ausblick auf zukünftige Anwendungsgebiete gegeben.

Kapitel 2

Fahrsimulatoren - State of the Art

2.1 Einsatzgebiete

Fahrsimulatoren finden heute in sehr vielen Anwendungsgebieten ihren Einsatz. Vom einfachen Computerspiel bis hin zu hochkomplexen Entwicklungsaufgaben. Der Grund für den Einsatz von Fahrsimulatoren ist aber immer das Schaffen einer virtuellen Realität bzw. dem Gefühl ein reales Fahrzeug zu lenken. Im folgenden Abschnitt werden die verschiedenen Anwendungsgebiete betrachtet.

Unterhaltung

In der Unterhaltungsindustrie sind Fahrsimulationen weit verbreitet. Sie werden für Computer und Spielekonsolen verwendet. Sie haben einen fixen Monitor und meist nur einen bis keinen Freiheitsgrad. Aktiv findet man Force Feedback nur bei Lenkrädern. Passiv bei Pedalen. Ein Beispiel für ein State-of-the-Art Gerät aus diesem Segment findet man unter [4].

Training

In der Luft- und Raumfahrt werden Simulatoren bereits seit Jahrzehnten für die Ausbildung und das Training verwendet. So verwendet laut [5] die Lufthansa bereits seit dem Jahr 1957 Flugsimulatoren um Piloten für den Flug auszubilden. In den 80er Jahren begann man laut [6] Simulatoren für die Aus- bzw. Weiterbildung des Personals von Zügen zu nutzen. Im gleichen Jahrzehnt wurde damit begonnen Fahrsimulatoren für den Straßenverkehr zu bauen.

Mehrere Studien, wie etwa jene in [7], zeigen dass der Lerneffekt in Fahrsimulatoren vergleichbar mit jenen im realen Fahrzeug ist. Der Vorteil des Simulators hierbei ist, dass auch riskante Versuche wie etwa Unfall und Beinahe-Unfall-Szenarien ohne Gefahr für

die/den FahrerIn durchgeführt werden können. Weiters können die FahrerInnen während des Trainings beobachtet und angewiesen werden.

Forschung

Auch in der Forschung bieten sich Fahrsimulatoren an, um die Mensch-Maschinen-Schnittstelle zwischen Fahrzeug und FahrerIn zu erforschen. Hierbei ergeben sich Forschungsmöglichkeiten wie etwa:

- Studien zum Fahrverhalten
- Studien zu Mensch-Maschine Schnittstellen
- Untersuchung der Einflüsse von Müdigkeit und Drogen
- Reaktionen des Körpers auf bestimmte Einflüsse
- Studien zu FahrernInnen mit eingeschränkter Fahrtüchtigkeit
- Entwicklung von Assistenzsystemen
- Einfluss von Geräusch- und Vibrationsentwicklung auf das Fahrverhalten

Wie auch beim Training ist es ein großer Vorteil des Fahrsimulators, dass das FahrerInnenverhalten unter Extrembedingungen untersucht werden kann. Solche Extremsituationen in einem Realversuch durchzuführen wäre auch für sehr versierte Probanden und Probandinnen gefährlich und schwer zu verantworten. Weiters kann mittels Fahrsimulator der Einfluss des Verkehrs auf den/die FahrerIn getestet werden, ohne andere Verkehrsteilnehmer zu gefährden.

Ein weiterer Vorteil ist es die/den FahrerIn in verschiedenen Situationen gezielten Ablenkungen aussetzen zu können. Dies ist vor allem bei der Entwicklung von Fahrerassistenzsystemen ein wichtiger Aspekt. Da Fahrerassistenzsysteme zur Sicherheit beitragen, setzen Automobilhersteller auf die Entwicklung solcher Systeme. Diese können in Fahrsimulatoren nicht nur kostengünstig getestet, sondern auch parametrisiert werden. Der wohl wichtigste Vorteil für alle genannten Punkte ist jedoch, dass die Szenarien mit verschiedenen FahrerInnen exakt reproduzierbar durchgeführt werden und beliebig oft wiederholt werden können.

Entwicklung

Fahrsimulatoren werden auch bereits zur Entwicklung von Fahrzeugen verwendet. Wie in [8] beschrieben, werden Fahrsimulatoren zur Abstimmung von Fahrwerken benutzt um das subjektive Fahrverhalten bereits in einer frühen Entwicklungsphase (vor der Prototypenherstellung) zu bewerten.

2.2 Ausführungsarten

2.2.1 PC System

Als Beispiel für solche Systeme steht das Logitech G27 Racing Wheel (siehe auch [4]) wie in Abbildung 2.1. Es besteht aus einem Lenkrad mit einem Durchmesser von 28cm, einer Pedalerie mit drei Pedalen und einem Ganghebel. Das Lenkrad bietet aktives Force Feedback und eine Drehbewegung von 900°. Der Ganghebel bietet sechs verschiedene Gänge. Die Pedalerie mit ihren drei Pedalen für Kupplung, Bremse und Gas bietet kein aktives Force Feedback. Die Pedale sind aber mit verschiedenem Kraftaufwand zu betätigen. Solche Aufbauten werden für die Unterhaltungsindustrie verwendet und dienen dazu, SpielerInnen von Rennspielen ein realitätsnahes Fahrgefühl zu geben. Sie werden meist in Verbindung mit einem PC fix montiert. Für Forschungszwecke ist ein solcher Aufbau aufgrund von nicht exakt vorgebbarem Lenkmoment und dem Fehlen von weiteren Zubehör nicht verwendbar.



Abbildung 2.1: Logitech G27 Racing Wheel, Quelle [4]

2.2.2 Fixed Base Simulator

Fix Basis Fahrsimulatoren sind vom Aufbau her größer als PC Systeme. Die/Der FahrerIn sitzt in einem Fahrzeugsessel mit fix montiertem Lenkrad, Pedal und Ganghebel. Der nächste Unterschied zum PC System ist, dass bei Fix Basis Simulatoren die Fahrbahnsicht meist mit mehreren Bildschirmen simuliert wird. In Abbildung 2.2 ist ein Beispiel für einen Fix Basis Simulator zu sehen. Hier ist auch zu sehen, dass bereits Instrumente zur realitätsnahen FahrerInnenkabinendarstellung eingebaut sind. Aufgrund der Größe dieser Simulatoren und der simulierten FahrerInnenkabine werden solche Simulatoren für Messen und Großveranstaltungen verwendet. Die Ausführung von Force Feedback ist bei solchen Simulatoren ähnlich der von PC Systemen und somit gleichermaßen schlecht für Forschungsaufgaben geeignet.



Abbildung 2.2: Fix Base Simulator der Firma ECA Faros, Quelle [9]

2.2.3 Motion Base Simulator

Motion Base Simulatoren sind die aufwendigste und fortschrittlichste Art von Fahr simulatoren. Hier wird ein Fahrzeug Mockup verwendet um den ProbandInnen den Eindruck zu geben, wirklich in einem Fahrzeug zu sitzen. Bei dieser Art von Simulator ist der gesamte Simulatorenbau bewegbar auf einer Basis gelagert. Die bewegbare Basis dient dabei dazu, dass der Eindruck eines sich bewegenden Fahrzeugs erzeugt werden kann. In diesem Segment gilt ein Fahr simulator als State-of-the-Art wenn er zumindest sechs Freiheitsgrade in der Bewegung besitzt und der Simulator einen Blickwinkel von 220° abdeckt. Ein Beispiel für einen Motion Base Simulator ist der Fahr simulator des Instituts für Verkehrssystemtechnik (DLR) in Braunschweig [10]. Ersichtlich in Abbildung 2.3. Dieser wird im nächsten Abschnitt näher betrachtet.

2.3 Motion Base Simulator des DLR

Der Motion Base Simulator des DLR kann als State-of-the-Art Simulator bezeichnet werden. Der Simulator verfügt über ein hydraulisches Hexapod-System, dies bedeute ein hydraulisches Bewegungssystem mit sechs Freiheitsgraden. Die Kabine in der das Fahrzeug platziert wird, ist hierbei unterhalb der oberen Gelenke angebracht. Diese Konstruktion ermöglicht laut [10] eine lineare Bewegung von drei Metern trotz geringer Bauhöhe. Die sechs Freiheitsgrade des Hexapod Systems sind alle unabhängig voneinander steuerbar. In Tabelle 2.1 sind Bewegungsart, Weg, Geschwindigkeit und Beschleunigung des Systems ersichtlich.

Die Visualisierung der Verkehrs- und Umfeldbegebenheiten wird mittels eines 270° horizontalen bzw. 40° vertikalen Blickfelds dargestellt. Die Gesamtauflösung beträgt dabei 9200×1280 Pixel. Auch das Verkehrsgeschehen hinter dem Fahrzeug wird mittels LCD im Rückspiegel und den Seitenspiegeln simuliert.

Wie schon erwähnt wird auch bei diesem Simulator eine komplettes Fahrzeug verwendet

Bewegungsart	Weg	Geschwindigkeit	Beschleunigung
Längs	$\pm 1,5$ m	± 2 m/s	10 m/s ²
Quer	$\pm 1,4$ m	± 2 m/s	10 m/s ²
Vertikal	$\pm 1,4$ m	± 2 m/s	10 m/s ²
Rollen	$-20^\circ / +21^\circ$	$\pm 50^\circ$ /s	250° /s ²
Nicken	$\pm 21^\circ$	$\pm 50^\circ$ /s	250° /s ²
Gieren	$\pm 21^\circ$	$\pm 50^\circ$ /s	250° /s ²

Tabelle 2.1: Bewegungsdaten des Hexapod Systems, Quelle: [10]

um der/dem FahrerIn eine möglichst realitätsnahe Simulation zu bieten. Der Datenaustausch zwischen dem Fahrzeug und der Simulationssoftware findet via CAN-Bus statt. Der Simulationsrechner wiederum steuert die Instrumente des Fahrzeugs auch über den CAN-Bus. Zuletzt verwendet das DLR ein Surround-Soundsystem um die Umgebungs- und Fahrzeuggeräusche zu simulieren. Verwendung findet der Fahrsimulator in der Untersuchung des Fahrverhaltens in Extremsituationen mit und ohne Fahrerassistenzsysteme.



Abbildung 2.3: Motion Base Simulator des DLR, Quelle [10]

Kapitel 3

Fahrsimulator des FTG

3.1 Einleitung

Auf Basis der State-of-the-Art Recherche wurde auf dem FTG ein Fahrsimulator im Zuge des *Muegen Driving* Projekts gebaut. Hierbei wurde vorerst auf eine Bewegungsplattform verzichtet, jedoch beim Aufbau auf eine spätere Verwendung einer Bewegungsplattform Rücksicht genommen. Der folgende Abschnitt beschäftigt sich mit der Auswahl, Implementierung und Ansteuerung der FFB-Aktuatoren die im Fahrsimulator eingebaut wurden.

3.2 Bremspedal

3.2.1 Anforderungen

Um der/dem FahrerIn ein möglichst realitätsnahes FFB auf dem Bremspedal zu geben ist es unerlässlich zuvor Messungen an einer echten Bremshydraulik durchzuführen, um die Anforderungen an ein Aktuatorssystem genau beschreiben zu können. Aus diesem Grund wurden am FTG Messungen am Bremspedal eines Fahrzeugs durchgeführt. Hierzu wurde ein Kraftsensor am Pedal angebracht, um die Kraft die von der FahrerIn bzw. dem Fahrer aufgebracht werden muss, zu messen. Zusätzlich wurde ein Potentiometer zur Pedalwegmessung angebracht. Beides ersichtlich in Abbildung 3.1.

Im Zuge der Messungen wurden drei Szenarien durchgeführt. Zum Ersten eine Komfortbremsung, bei der von der FahrerIn bzw. dem Fahrer eine langsame Bewegung, wie etwa bei einer Bremsung zum Anhalten an einer Kreuzung durchgeführt wird. Zum Zweiten eine Normalbremsung, wie sie etwa bei einer Fahrt in der Stadt und einem bremsenden Vorderfahrzeug vorkommt und zum Dritten eine Notbremsung, die selbsterklärend ist. Exemplarisch ist in Abbildung 3.2 der Zusammenhang zwischen Weg und Kraft für eine Vollbremsung dargestellt.



Abbildung 3.1: Aufbau Pedalwegmessung am FTG

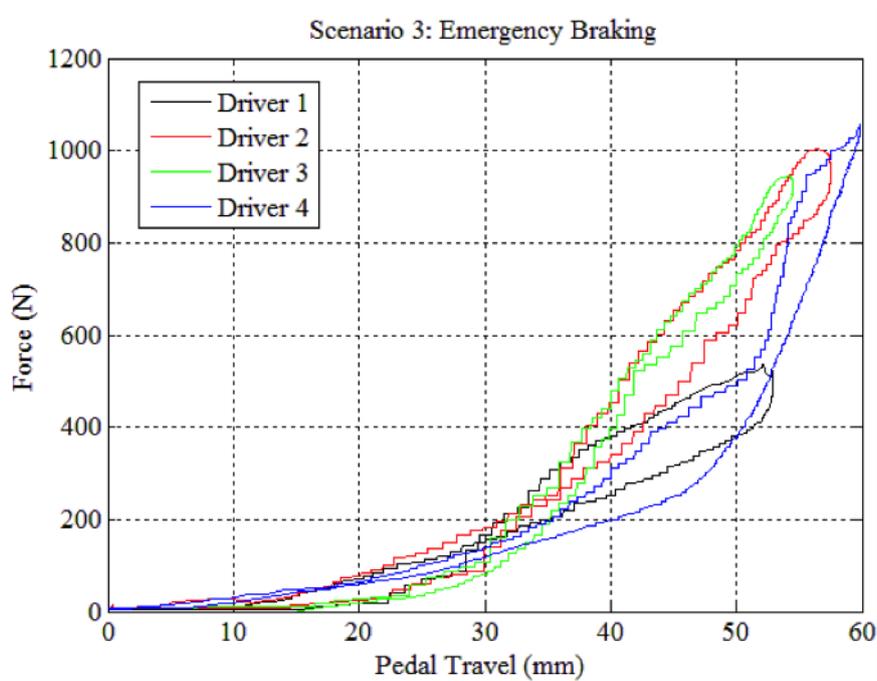


Abbildung 3.2: Zusammenhang von Pedalweg zu Kraft bei einer Vollbremsung

Aus diesen Messergebnissen wurden die Anforderungen an den Bremsaktuator, wie in Tabelle 3.1 ersichtlich, formuliert. Weiters war es wichtig, dass der Aktuator mit möglichst kleiner Zeitverzögerung aus der Simulationsumgebung ansprechbar ist. Die Zeitverzögerung ergibt sich aus der kinästhetischen Aufnahmefähigkeit des menschlichen Körpers, welche laut [11] im Bereich von 5-10Hz liegt. Die Simulationsoftware stammt hierbei vom Projektpartner AVL und läuft auf Basis des Programms Matlab Simulink.

Anforderung	Wert
max. Kraft (Normalbremsung)	500N
max. Kraft (Vollbremsung)	1000N
min. Weg	60mm
Geschwindigkeit	600 mm/s

Tabelle 3.1: Anforderungen für den Bremsaktuator aus den erhaltenen Messdaten

3.2.2 Lösungsvarianten

Nach der Definition der Anforderungen an den Aktuator, wurden verschiedene Systeme auf ihre Verwendbarkeit untersucht. Im Folgenden werden die verschiedenen Lösungsvarianten beschrieben und anschließend eine Nutzwertanalyse zur Entscheidungsfindung durchgeführt.

Linear Aktuator

Der lineare Aktuator ist ein häufig für Automatisierungsaufgaben genutztes Instrument. Als lineare Aktuatoren werden jene bezeichnet die eine translatorische Bewegung durchführen. Ein Beispiel für einen solchen Aktuator ist der in [12] ersichtliche Aktuator der Firma Linmot. Das Konzept für die Verwendung eines solchen Aktuators ist in Abbildung 3.3 zu sehen. Wie am Konzept ersichtlich, wird dieser hier direkt an das Pedal montiert. Die Anforderungen von Kraft, Geschwindigkeit und Weg sind direkt auf die Anforderungen für den Aktuator anwendbar.

Der Vorteil dieses Konzepts ist, dass der mechanische Aufbau sehr einfach gestaltet ist. Der Nachteil dieses Ansatzes besteht in der Steifigkeit des Systems bzw. von Linear Aktuatoren und der im Vergleich zum folgenden Konzept geringen Geschwindigkeit.

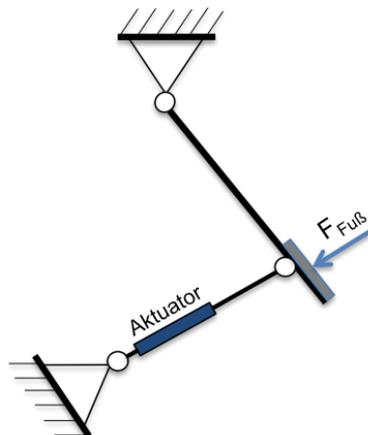


Abbildung 3.3: Aufbau Bremsaktuator mit linear Aktuator

Rotatorischer Aktuator (Motor)

Das zweite Konzept wurde mit einem rotatorischen Aktuator, also einem Standardmotor ausgearbeitet. Bei diesem Aufbau müssen die Anforderungen an den Motor auf Drehbewegungen umgerechnet werden. Der Motor ist hierbei über einen Verbindungsarm mit dem Pedal verbunden. Ein Beispiel für einen rotatorischen Motor, der die Anwendung dieses Konzepts erlaubt, ist der XtraforsPrime der Firma Infranor und findet sich in [13].

Der Vorteil dieses Konzepts liegt in der hohen Geschwindigkeit (durch hohe Drehzahl) und der hohen Kraft (durch hohes Drehmoment). Noch ein Vorteil ist, dass Motoren dieser Art weit verbreitet sind und mit dem selben Konzept bereits am Flugsimulator des Instituts für Mechanik [14], für den Steuerknüppel verwendet werden. Auch der mechanische Aufwand hält sich hier in Grenzen. Ein Nachteil dieses Aufbaus ist, im Hinblick auf eine Erweiterung des Simulators mit einer Bewegungsplattform, das zum ersten Konzept relativ hohe Gewicht des Aktuators.

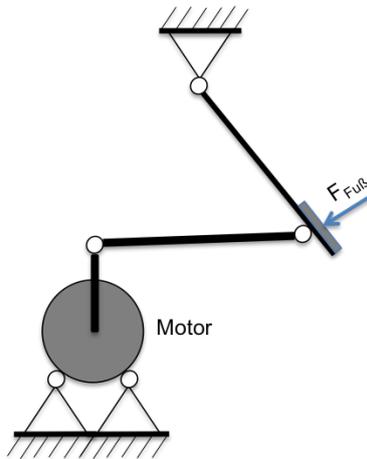


Abbildung 3.4: Aufbau Bremsaktuator mit rotatorischen Aktuator

Hydraulikzylinder

Das dritte Konzept verwendet den selben Aufbau wie das erste Konzept (linearer Aktuator). Der Unterschied besteht darin, dass hierbei ein Hydraulikzylinder anstelle eines elektrischen Aktuators verwendet wird.

Der Vorteil dieses Aufbaus sind die hohen Kräfte die von Hydraulikzylindern aufgebracht werden können. Die Nachteile sind der höhere Aufwand zur Ansteuerung des Systems und die relativ langsamen Geschwindigkeit im Vergleich mit den anderen Konzepten. Weiters sind Hydraulikzylinder auch sehr wartungsintensiv.

3.2.3 Aktuatorwahl

Zur Wahl des Aktuator systems wurde eine Nutzwertanalyse durchgeführt. Dazu werden verschiedene Kriterien ausgewählt und jedes System danach beurteilt, wie sehr das System ein Kriterium erfüllt. Dabei werden Punkte von 1-5 vergeben. Die Bewertung sieht wie folgt aus:

- 5 Punkte - Kriterium sehr gut erfüllt
- 4 Punkte - Kriterium gut erfüllt
- 3 Punkte - Kriterium ausreichend erfüllt
- 2 Punkte - Kriterium schlecht erfüllt
- 1 Punkt - Kriterium sehr schlecht erfüllt

Die Punkte werden mit dem Gewichtsprozentatz des zugehörigen Kriterium multipliziert und danach addiert. So ergibt sich eine maximale Punktzahl von 5 und eine minimale Punktzahl von 1. Die Gewichtung der Kriterien wurde im Projektteam festgelegt.

Kriterium	Linear	Rotatorisch	Hydraulisch	Gewichtung
Positionsgenauigkeit	4	4	3	17,5%
Geringer mechanischer Aufwand	5	4	5	15 %
Eignung für hohe Kräfte	3	5	5	20 %
Geschwindigkeit	4	5	4	20 %
Ansteuerungsaufwand	4	4	2	12 %
Kosten	3	4	1	15,5%
Summe	3,8	4,4	3,5	100 %

Tabelle 3.2: Nutzwertanalyse zur Bestimmung des Aktuator konzepts

Aus der Nutzwertanalyse in Tabelle 3.2 geht eindeutig hervor, dass die Lösung mit einem rotatorischen Aktuator den Anforderungen am besten entspricht. Im nächsten Abschnitt wird nun das Aktuator system mit einem rotatorischen Motor näher betrachtet.

3.2.4 Aktuator system mit rotatorischem Motor

Der nächste Schritt zur Ausführung des Aktuator systems ist die Wahl eines Motors bzw. Motor-Controller Systems. Bei der Suche nach einem geeigneten System entschied man sich für ein Motor-Controller System der Firma Infranor [13]. Der AC Servomotor XtraforsPrime wurde gewählt, da er laut [13] durch den nutenlosen Aufbau des Stators kein Cogging, das heißt kein Rastmoment, aufweist. Der ausgewählte Motor hat ein Nenndrehmoment 13,2Nm und ein maximales Drehmoment von 189Nm. Die Spannungsversorgung

des Motors kommt vom bereits angesprochenen Controller. Der Controller wird dazu mit zwei verschiedenen Spannungen versorgt, zum Einen mit einer 3-phasen 400V Wechselspannung zur Leistungsversorgung und zum Anderen mit einer 24V Gleichspannung für die Kontrolllogik. Der Aufbau des Motor-Controllersystem ist in Abbildung 3.5 ersichtlich. Wie dort ersichtlich ist wird auch ein Positionssignal vom an der Motorwelle angebrachten Absolutgeber an den Controller gesendet. Dies geschieht per separater Verbindung mit einer Auflösung von 4096 Punkten pro Umdrehung. Dieses System hat den Vorteil, dass es einen Controller mit CAN-Schnittstelle besitzt. Dieses Netzwerk ist eingeschränkt echtzeitfähig und wird auch von Fahrzeugherstellern als Bussystem im Fahrzeug verwendet. Das Motor-Controller System verwendet hierbei ein höherschichtiges Netzwerkprotokoll namens CANopen das im Folgenden näher erläutert wird.

Simulationsrechner

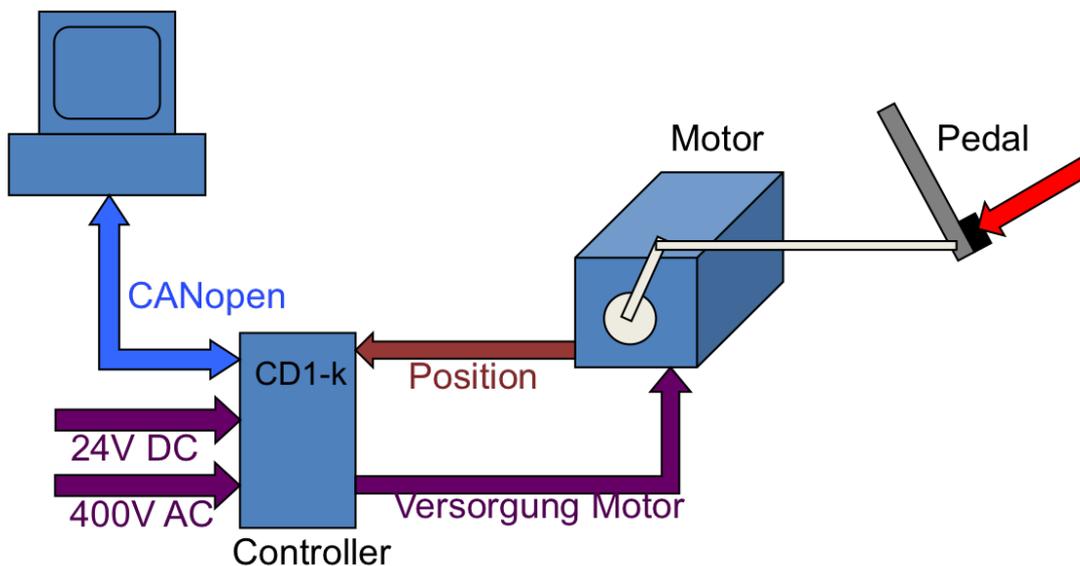


Abbildung 3.5: Motor-Controller System des Bremsaktuators mit rotatorischen Aktuator

3.2.4.1 CANopen

CANopen ist ein auf CAN basierendes höherschichtiges Netzwerkprotokoll. Es definiert Schicht 7 des OSI-ISO Schichtenmodells, die sogenannte Anwendungsschicht. Es benutzt CAN als Transportschicht, das heißt CANopen beschäftigt sich nicht mit physikalischen Signalen, wie Bit-timing und Synchronisation. Es wurde in einem Projekt unter der Leitung der Firma Bosch entwickelt. 1994 wurde eine internationale Vereinigung, die CAN in Automation (CiA) [15] gegründet, um die CANopen Spezifikation in eine internationale Norm einzubetten. CANopen wird hauptsächlich in der Automatisierungstechnik einge-

setzt. Es bietet standardisierte Kommunikationsobjekte: Prozessdatenobjekte zur Übertragung von Echtzeitdaten und Servicedatenobjekte zur Konfiguration der CANopen-Geräte. Weiters gibt es noch Kommunikationsobjekte für Netzwerkmanagement und andere Sonderfunktionen. Der CANopen Bus ist auf 127 Teilnehmer begrenzt und unterstützt eine Datenübertragungsrate von bis zu 1Mbit/s.

Das Objektverzeichnis

CANopen verwendet ein Objektverzeichnis (Abbildung 3.6) um die Funktionalität eines Gerätes zu beschreiben. Im Objektverzeichnis sind alle für die Verwendung des Geräts notwendigen Daten gespeichert und können mittels eines klar definierten Kommunikationsprotokolls verändert bzw. gelesen werden. Es stellt also die Schnittstelle zwischen der/dem AnwenderIn und dem Gerät dar. Alle diese Kommunikationsobjekte sind standardisiert mit einem 16-Bit Index adressiert. Für Objekte die Arrays darstellen, gibt es einen zusätzlichen 8-Bit Sub-Index.

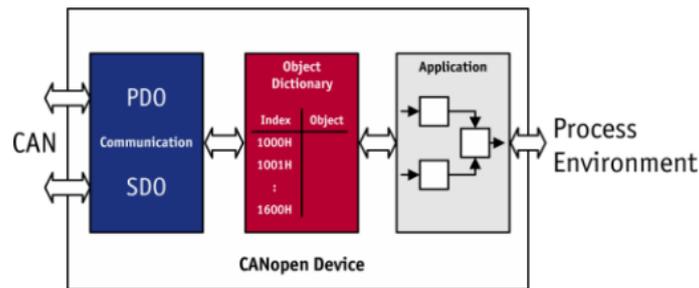


Abbildung 3.6: Schematische Darstellung eines Geräteprofils, Quelle [16]

Prozessdatenobjekte PDO

Prozessdatenobjekte werden in einer einzelnen CAN-Nachricht ohne weiteren Protokoll-Overhead übertragen. Jedes PDO hat einen eindeutigen CAN Identifier und darf nur von einem Gerät übertragen werden, kann aber von mehreren Geräten empfangen werden. Dies entspricht einer Producer-Consumer Verbindung. Aufgrund der Spezifikation von CAN können nur 8 Byte an Daten übertragen werden. Die Auslöser einer PDO Übertragung können verschiedene Ereignisse sein. Der wichtigste Auslöser für das verwendete Aktuator System ist die Sync-Nachricht. Wird diese Nachricht empfangen, löst dies das Senden von voreingestellten Prozessdaten aus. Die Sync Nachricht wird in der gegebständlichen Anwendung im Fahrsimulator vom Simulationsrechner gesendet und hat den CAN-Identifier 128. Die Zeit zwischen zwei Sync-Nachrichten ist in jedem Gerät durch die Kommunikationszykluszeit definiert. Die Identifier von PDO-Nachrichten sind hochprior gewählt, um

eine kurze Antwortzeit zu garantieren. Eine PDO Übertragung findet ohne Bestätigung statt.

Servicedatenobjekte SDO

Servicedatenobjekte können auf mehrere CAN-Nachrichten aufgeteilt gesendet werden. Sollte dies der Fall sein, so werden die Daten durch einen Index adressiert. Sie schreiben bzw. lesen Objekte aus dem Objektverzeichnis. Jedes SDO hat einen eindeutigen CAN Identifier. Die Übertragung von SDO erfolgt mit Bestätigung und entspricht einer Client-Server Kommunikation. Im Gegensatz zu PDO-Identifiern sind SDO-Identifizier niederprior, da sie vor der Verwendung des Geräts zur Konfigurierung gesendet werden und somit eine kurze Antwortzeit nicht wichtig ist.

Netzwerkmanagementobjekte

Die Netzwerkmanagementobjekte umfassen alle Objekte zur Netzwerksteuerung. Sie dienen vor allem zur Steuerung der CANopen Zustandsmaschine eines Geräts. Sie sind in einer einzelnen CAN Nachricht mit 2 Byte Datenlänge und einem Identifier Null, also höchster Priorität, definiert. Netzwerkmanagementobjekte werden nur vom NMT-Master versendet, welcher in unserer Anwendung der Simulationsrechner darstellt.

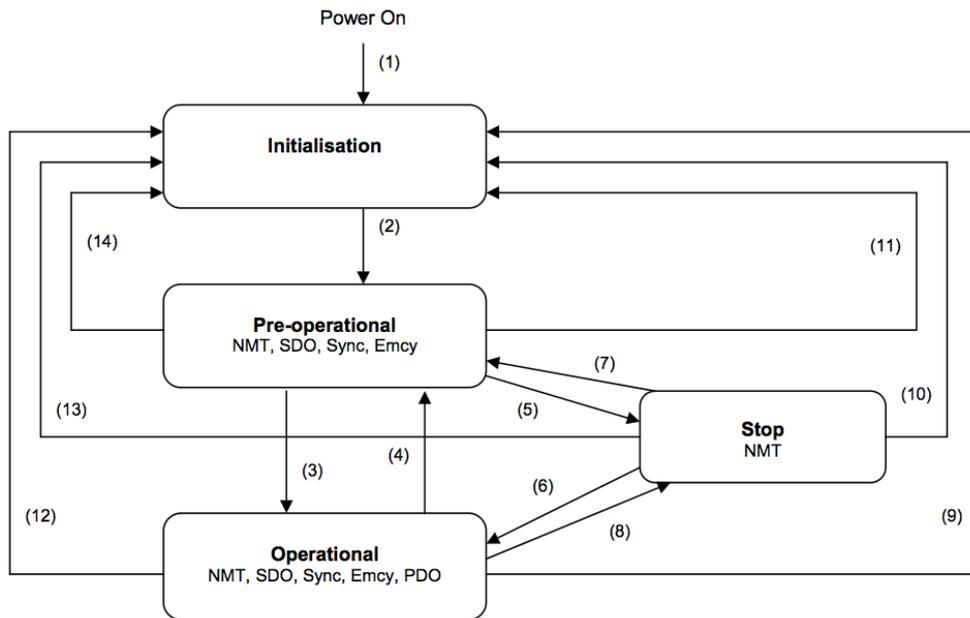


Abbildung 3.7: CANopen NMT-Zustandsmaschine, Quelle CANopen Communication Protocol [13]

Jedes CANopen Gerät hat eine interne Initialisierungsphase, die nach dem Einschalten gestartet wird. Danach wechselt das Gerät von selbst in den Pre-Operational Modus. Im Pre-Operational Modus werden die SDO übertragen und das Gerät wird damit konfiguriert. Ein Zustandswechsel von Pre-Operational in den Operational Modus kann nur durch NMT-Objekte durchgeführt werden. Sobald ein Gerät im Operational Modus ist, kann mit der Übertragung von PDO Nachrichten begonnen werden. Im Zustand Stopped ist keine Kommunikation, außer von NMT-Objekten, mehr möglich.

Kommunikationsobjekt-ID (COB-ID)

CANopen ist ein kommunikationsobjektorientiertes Netzwerk, daher ist die Zuordnung der CAN-ID ein zentrales Thema im Systemdesign. CANopen hat die Zuordnung von Identifiern standardisiert um der/dem AnwenderIn dieses Design abzunehmen. Die Zuordnung der COB-ID erfolgt bei CANopen in Abhängigkeit der Geräte-ID. Der Aufbau der COB-ID ist in Tabelle 3.3 ersichtlich.

10	9	8	7	6	5	4	3	2	1	0
Funktionscode				Geräte-ID						

Tabelle 3.3: Aufbau der 11-Bit COB-ID

Funktionscode

Der Funktionscode gibt an welche Funktion mit der Nachricht erfüllt werden soll. Es gibt sechs vordefinierte PDO die über spezielle Funktionscodes direkt angesprochen werden. Sie teilen sich in drei Sende-PDO (TPDO) und drei Empfangs-PDO (RPDO) mit denen gespeicherte Objektdaten direkt angesprochen werden können bzw. im Falle der Sende-PDO vom Gerät gesendet werden können. Die verschiedenen Funktionscodes sind in Tabelle 3.4 zu sehen.

Funktionscode	Objekt
0000	Netzwerkmanagement
0001	SYNC- / Notfall-Nachricht
0011	TPDO1
0100	RPDO1
0101	TPDO2
0110	RPDO2
0111	TPDO3
1000	RPDO3
1011	SDO Transmit
1100	SDO Receive

Tabelle 3.4: CANopen Funktionscodes

Kommandocode

Der Kommandocode wird im 1. Byte der CANopen Nachricht übertragen und gibt an, ob und wie viele Bytes gelesen oder geschrieben werden sollen. Die drei im nachstehenden Kapitel verwendeten Kommandocodes sind:

- 43 - Schreiben von 2 Datenbytes
- 47 - Schreiben von 1 Datenbyte
- 64 - Leseanforderung

3.2.4.2 Konfigurierung Controller CD1-k

Der erste Schritt zur Implementierung des Bremspedalsystems ist die Konfigurierung des Motor-Controller Systems als Untersystem. Wie in Abbildung 3.5 ersichtlich, bestehen zwei Verbindungen zwischen dem Controller und dem Motor. Die eine Verbindung stellt die Versorgungsspannung für den Motor dar. Dies ist ein Ausgangssignal des Controllers. Die zweite Verbindung stellt das Positionssignal des Motors und somit ein Eingangssignal des Controllers dar. Um dieses System verwenden zu können, muss zuerst der Controller mit den Motordaten konfiguriert werden. Zu diesem Zweck stellt Infranor eine Parametrierungssoftware namens Visual Drive Setup (VDS) zur Verfügung.

Visual Drive Setup (VDS)

Mit dem Visual Drive Setup werden die Kenndaten des am Controller angeschlossenen Motors und Positionssensors im Controller gespeichert. In Abbildung 3.8 ist das Startfenster des VDS zu sehen. Es ist nicht nur zur Konfigurierung des Controllers, sondern

auch zur Überwachung der Funktion des Controllers verwendbar. Für das verwendete System musste mittels VDS ein neuer Motor gespeichert werden. Die wichtigsten Kenndaten für das Erzeugen des Motors werden im folgenden betrachtet:

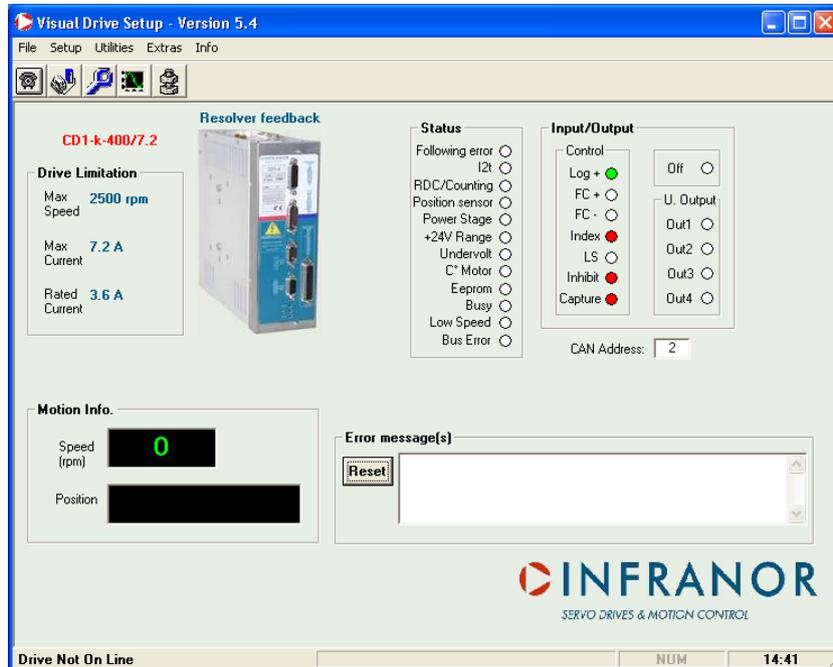


Abbildung 3.8: Startfenster des Visual Drive Setup [13]

Motorinduktivität Als erstes muss die im Datenblatt des Motors ersichtliche Induktivität eingegeben werden. VDS berechnet daraus automatisch die Parameterdaten der Steuerung.

Spannungsversorgung und Maximalstrom Im nächsten Schritt muss die Spannungsversorgung und der Maximalstrom des Motors angegeben werden.

Polpaarzahl Die Polpaarzahl gibt an wie viele Polpaare ein Motor hat. Auch die Polpaarzahl des Motors muss dem Controller zur Steuerung des Motors bekannt sein.

Encoder Type Der Encoder entspricht dem Positionssensor der beim Motor verbaut ist. Im Falle des gewählten Motors ist dies ein *Hiperface*[17] Absolutencoder mit einer Auflösung von 4096 Punkten/Umdrehung. Der Encoder ist vor allem für die richtige Steuerung des Motors im Positionsmodus wichtig.

Nach Eingabe der richtigen Werte wird eine Auto-Phasing Prozedur zur Überprüfung der eingestellten Werte durchgeführt. Wird diese erfolgreich abgeschlossen kann der Motor bereits mittels VDS gesteuert werden. Danach kann das System an den CANopen Bus geschlossen und angesprochen werden.

3.2.4.3 CANopen Verbindung mittels Matlab Vehicle Network Toolbox VNT

Da die Simulationsumgebung in Matlab eingebettet ist, war es notwendig nach einer geeigneten Schnittstelle zum CAN Netzwerk zu suchen. Diese wird durch die VNT ausgeführt. Die VNT bietet mehrere Funktionen um aus Matlab CAN-Nachrichten zu senden bzw. zu empfangen. Sie bietet zusätzlich die Funktion CAN-Nachrichten nach eigenen Definitionen zusammenzustellen und somit auch die Option CANopen PDO und SDO Nachrichten zu senden.

Die VNT wurde nun zuerst dazu benutzt, um den Controller zu konfigurieren und mittels NMT-Objekten in den Operational Modus zu bringen. Hierzu wurde ein Skript in Matlab geschrieben. Dieses Skript enthält die Initialisierung des CAN-Kanals, das Starten des Betriebs sowie das Beenden des Betriebs und ist in Anhang B ersichtlich. Anhand von Auszügen aus diesem Skript werden nun die verschiedenen Funktionen des VNT beschrieben. vgl. [18]

```
%% Creating Channel
commCh = canChannel( 'Vector', 'VN1640_1', 1)
%configBusSpeed(commCh,500000)
```

canCh = canChannel(vendor, device , devicechannelindex)

Dieser Befehl stellt den Konstruktor eines CAN-Kanals mit Verbindung zum angegebenen Hardwarekanal dar. Die Argumente des Befehls sind selbsterklärend: der Hersteller (vendor) des Geräts das benutzt werden soll, der Gerätenamen (device) und die Nummer des Kanals bei mehrkanaligen Geräten(devicechannelindex).

configBusSpeed(canCh, busspeed)

Mit diesem Befehl kann die Busübertragungsrate des erstellten Kanal konfiguriert werden. Der Befehl ist im Skript auskommentiert, da die Standardbusübertragungsrate, die vom Controller benutzt wird, 500kBit/s entspricht. Die Argumente sind der verwendet Kanal (canCh) und die gewünschte Busübertragungsrate (busspeed).

```
% Creating Startup Messages
operationMode = canMessage(1537, false, 8)
operationMode.Data = [47 96 96 0 4 0 0 0]
```

canNachricht = canMessage(id, extended, datalength)

Mit dieser Funktion wird eine CAN Nachricht laut den übergebenen Argumenten erstellt. Die Argumente geben folgende Daten an: Den Nachrichten-Identifizier (id), ob es sich um einen überlangen Nachrichten-Identifizier handelt (extended) und die Datenlänge der Nachricht (datalength).

canNachricht.Data

Mit dieser Code Zeile wird die Nachricht mit den gewünschten Daten befüllt. Hierbei ist zu beachten das die Werte im Dezimalformat sind.

```
% Start the Channel
start(commCh)
```

start(canch)

Mit dieser Funktion wird der Hardware CAN Kanal mit den voreingestellten Konfigurationen gestartet. Ist ein Kanal gestartet, so kann seine Übertragungsgeschwindigkeit nicht mehr verändert werden.

```
%% Send Operation Mode Message
transmit(commCh, operationMode)

% Receive Operation Mode Message
rxmsg = receive(commCh, 2)
```

transmit(canch, message)

Diese Funktion sendet eine vorher definierte CAN Nachricht (message) über den angegebenen Kanal (canch). Wichtig ist hierbei, dass der Kanal vorher gestartet und funktionsfähig sein muss.

canNachricht = receive(canch, messagesrequested)

Diese Funktion stellt das Komplement zur transmit-Funktion dar. Mit ihr werden eine angegebene Anzahl an Nachrichten (messagesrequested) vom Empfangsbuffer des angegebenen Kanals (canch) geladen. Die Anzahl an Nachrichten muss hierbei größer Null sein.

```
%% Stop the Channels  
stop(commCh)
```

stop(canch)

Diese Funktion stellt das Komplement zum start Befehl dar. Es beendet die Kommunikation des angegebenen Kanals (canch).

pack / unpack

Zusätzlich gibt es noch die Funktionen **pack** und **unpack** mit denen individuelle CAN-Nachrichten aufgebaut und mit Daten befüllt werden können. Diese Funktionen finden vor allem in Matlab Simulink ihre Anwendung.

Flussdiagramm

In Abbildung 3.9 ist das Flussdiagramm einer CAN-Verbindung dargestellt. Zuerst muss der Kanal erstellt, konfiguriert und gestartet werden. Danach teilt sich das Diagramm in Sende- und Empfangsast auf. Im Sendeast wird die Nachricht zuerst erstellt und dann mittels transmit gesendet. Im Empfangsast wird mittels receive der Nachrichtenpuffer ausgelesen und die Nachricht danach entpackt(unpack). Nach dem Senden bzw. Empfangen aller erforderlichen Daten, muss der Kanal gestoppt werden.

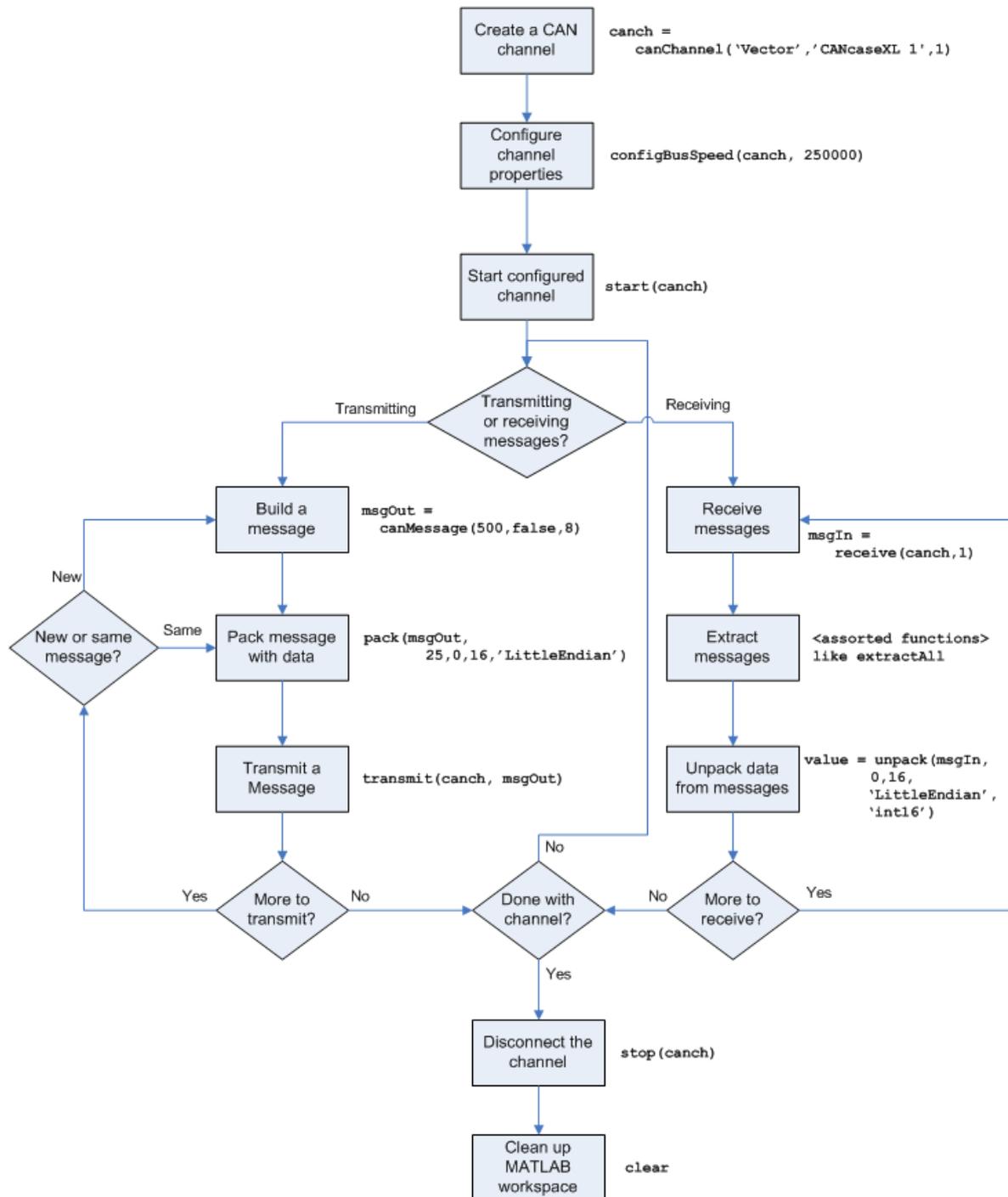


Abbildung 3.9: Flussdiagramm einer CAN Netzwerkverbindung mittels VNT, Quelle [18]

3.2.4.4 Starten des Controllerbetriebs mittels CANopen

Wie im vorigen Abschnitt erklärt, wurde ein Skript geschrieben um den Controller in die richtige Betriebsart zu bringen bzw. um den Betrieb zu starten. Dazu wurden die oben beschriebenen Funktionen, wie im Anhang B (Startup.m) beschrieben, verwendet. Desweiteren muss der vom Controller bereitgestellte Torque Modus eingestellt werden. In diesem Modus folgt der Controller einem vorgegebenen Drehmoment das über die Systemgeometrie in eine Pedalkraft umgeformt wird. Dazu muss zuerst eine Konfiguration der CANopen Objekte erfolgen. Das Flußdiagramm des Ablaufs ist in Abbildung 3.10 zu sehen. Zwischen den einzelnen Nachrichten wird immer eine Pause von 0,1s eingelegt, um dem Controller die Möglichkeit zu geben die übertragenen Werte zu speichern. Erst dann ist er bereit neue Nachrichten zu empfangen und sie zu verarbeiten. Nach jeder empfangenen Nachricht antwortet der Controller, wie im Abschnitt 3.2.4.1 beschrieben, mit einer Antwortnachricht um anzuzeigen ob die Nachricht erfolgreich übernommen werden konnte. Der folgende Abschnitt beschreibt nun den Aufbau der übertragenen Nachrichten:

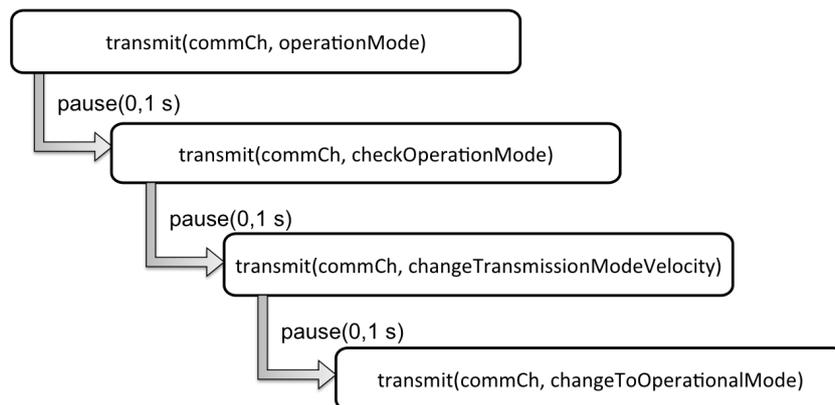


Abbildung 3.10: Flussdiagramm Initialisierung Controller CD1-k

operationMode

Diese Nachricht dient dazu den Controller in den richtigen Betriebsmodus (Torque-Modus) zu bringen. Die ID setzt sich hier aus dem Funktionscode der SDO Übertragung und der Geräteadresse zusammen. Die Objektdaten sind in Tabelle 3.5 aufgelistet. Die zugehörigen Datenwerte und -beschreibungen in Tabelle 3.6.

Mode of Operation			
Funktionscode (binär - 4 Bit)	Geräteadresse (binär - 7 Bit)	ID (binär)	ID(dezimal)
1100	0000001	1100 0000001	1537
Datentyp	integer8		
Objectadresse (hexadezimal)	6060		
Werte	1 Profile Position Mode 3 Profile Velocity Mode 4 Torque Profile Mode 6 Homing Mode 7 Interpolated Position Mode		

Tabelle 3.5: Objektdaten Mode of Operation, vgl. CANopen Communication Profile [13]

Daten - operationMode							
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte	8. Byte
Kommandocode	Objektadresse		Sub-Index	Wert			
47	96	96	0	4	0	0	0

Tabelle 3.6: Datensegment checkOperationMode

checkOperationMode

Diese Nachricht dient zur Abfrage des eingestellten Betriebsmodus und wurde in das Skript eingebaut um die vorher gesendete Einstellung zu überprüfen. Die Objektdaten sind in Tabelle 3.7 aufgelistet. Die zugehörigen Datenwerte und -Beschreibungen sind in Tabelle 3.8 ersichtlich.

Mode of Operation Display			
Funktionscode (binär - 4 Bit)	Geräteadresse (binär - 7 Bit)	ID (binär)	ID(dezimal)
1100	0000001	1100 0000001	1537
Datentyp	integer8		
Objectadresse (hexadezimal)	6061		
Werte	1 Profile Position Mode 3 Profile Velocity Mode 4 Torque Profile Mode 6 Homing Mode 7 Interpolated Position Mode (voreingestellt)		

Tabelle 3.7: Objektdaten Mode of Operation Display, vgl. CANopen Communication Profile [13]

Daten - operationMode							
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte	8. Byte
Kommandocode	Objektadresse		Sub-Index	Wert			
64	97	96	0	0	0	0	0

Tabelle 3.8: Datensegment checkOperationMode

changeTransmissionModeVelocity

Diese Nachricht dient dazu, den Übertragungsmodus des TPDO2 so zu ändern, dass sie wie TPDO3 zyklisch bei Empfang des SYNC Objekts übertragen wird. Es handelt sich hierbei um die Motorgeschwindigkeit. Die Objektdaten sind in Tabelle 3.9 aufgelistet. Die zugehörigen Datenwerte und -beschreibungen in Tabelle 3.10.

2nd Transmit PDO Parameter			
Funktionscode (binär - 4 Bit)	Geräteadresse (binär - 7 Bit)	ID (binär)	ID(dezimal)
1100	0000001	1100 0000001	1537
Datentyp	integer8		
Objectadresse (hexadezimal)	1801		
Werte	1 Zyklisch 253 asynchron (voreingestellt)		

Tabelle 3.9: Objektdaten TPDO2, vgl. CANopen Communication Profile [13]

Daten - TPDO2							
1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte	7. Byte	8. Byte
Kommandocode	Objektadresse		Sub-Index	Wert			
47	1	24	2	1	0	0	0

Tabelle 3.10: Datensegment changeTransmissionModeVelocity

changeToOperationalMode

Diese Nachricht beinhaltet das NMT-Objekt zur Steuerung der CANopen Zustandsmaschine. Es beinhaltet im Speziellen das Objekt das einen Zustandswechsel auf den Operational Modus zur Folge hat. Im Gegensatz zu den bisher besprochenen Objekten ist es nur 2 Byte lang. Die COB-ID eines NMT-Objekt ist immer Null, die Adressierung eines Geräts erfolgt im Datensegment. Die Objektdaten sind in Tabelle 3.11 aufgelistet. Die zugehörigen Datenwerte und -Beschreibungen sind in Tabelle 3.12 ersichtlich.

NMT Object			
Funktionscode (binär - 4 Bit)	Geräteadresse (binär - 7 Bit)	ID (binär)	ID(dezimal)
0000	-	00000000000	0
Datentyp	raw		
Werte	1 Start Remote Node 2 Stop Remote Node 128 Enter Pre-Operational Mode 129 Reset Node 130 Reset Communication		

Tabelle 3.11: Objektdaten NMT-Objekt, vgl. CANopen Communication Profile [13]

Daten - NMT Object	
1. Byte	2. Byte
Wert	Geräteadresse
1	1

Tabelle 3.12: Datensegment changeToOperationalMode

3.2.4.5 Aktivieren des Stromreglers mittels CANopen

Nach der Ausführung dieses Segments ist der Controller nun im Operational Mode und kann mittels CANopen Daten betrieben werden. Als nächste Ebene muss nun die Motorsteuerung des Controllers aktiviert werden. Um dies durchzuführen muss zunächst die Zustandsmaschine des Stromreglers betrachtet werden (Abbildung 3.11). Um den Stromregler zu aktivieren müssen die Transitionen 2, 3 und 4 durchgeführt werden. Diese Zustandsänderungen können mit dem Control Word Objekt erreicht werden. In diesem Segment wurde darauf verzichtet mehrere Nachrichten zu generieren, da sich nur das Datensegment ändert.

In Tabelle 3.13 sind die Objektdaten des Control Word Objekts aufgelistet. Der Aufbau unterscheidet sich von den bisher betrachteten darin, dass hier jedem Bit im Objekt eine

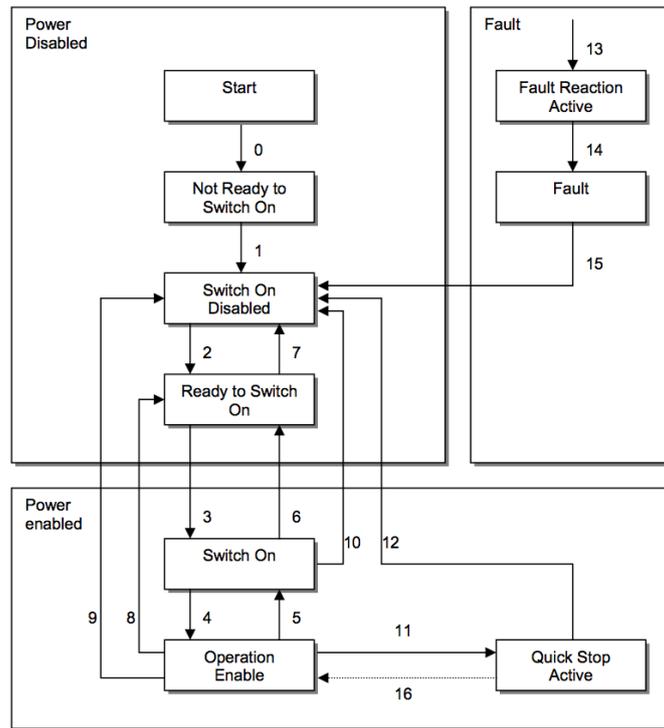


Abbildung 3.11: Zustandsmaschine Stromregler Controller CD1-k, Quelle [13]

eigene Funktion zugeordnet ist. Daraus ergibt sich auch, dass es sich hier um einen unsigned16 Datentyp handelt. Die Bits 4-6 sind operationsmodus-spezifisch, dies bedeutet das sie je nach Betriebsmodus eine andere Bedeutung haben.

Aus dem Zusammenhang zwischen den Objektdaten und der Zustandsmaschine des Stromreglers, ergibt sich nun Tabelle 3.14. Dort werden die Zusammenhänge zwischen Zustand, Bitwerten und möglichen Transitionen dargestellt.

Nach Ausführung des Abschnitts "Start Operation" der Datei "Startup.m" aus Anhang B ist der Stromregler des Motors aktiviert und betriebsbereit. Der Motor folgt nun dem vorgegebenen Drehmomentwerten. Der Aufbau der Verbindung aus Matlab wird im nächsten Abschnitt betrachtet.

Control Word Parameter			
Funktionscode (binär - 4 Bit)	Geräteadresse (binär - 7 Bit)	ID (binär)	ID(dezimal)
1100	0000001	1100 0000001	1537
Datentyp	unsigned16		
Objectadresse (hexadezimal)	6040		
Werte			
Bit 0	Switch On		
Bit 1	Disable Voltage		
Bit 2	Quick Stop		
Bit 3	Enable Operation		
Bit 4	Operation Mode Specific		
Bit 5	Operation Mode Specific		
Bit 6	Operation Mode Specific		
Bit 7	Reset Fault (rising edge)		
Bit 11	Out_of_Limit (manufacturer specific)		

Tabelle 3.13: Objektdaten Control Word Parameter, vgl. CANopen Communication Profile [13]

Command Bit of the control_word	bit 7 Fault Reset	bit 3 Enable Operation	bit 2 Quick Stop	bit 1 Disable Voltage	bit 0 Switch On	Transition
Shutdown	X	X	1	1	0	2, 6, 8
Switch On	X	X	1	1	1	3
Disable Voltage	X	X	X	0	X	7, 9, 10, 12
Quick Stop	X	0	1	1	1	7, 10, 11
Disable Operation	X	0	1	1	1	5
Enable Operation	X	1	1	1	1	4, 16
Fault Reset	↑	X	X	0	X	15

Tabelle 3.14: Zustände, Bitwerte und mögliche Transitionen, vgl. CANopen Communication Profile [13]

3.2.4.6 Vorgabe des Motordrehmoments

Da die Simulation in Matlab Simulink realisiert ist, muss auch die Vorgabe des Motordrehmoments aus Matlab Simulink funktionieren. Für die Vorgabe des Drehmoments wird das Prozessdatenobjekt "Target Torque" verwendet. Die Auflösung des übergebenen Wert entspricht einem tausendstel des Maximalstroms des Motors. Der Aufbau des Objekts ist in Tabelle 3.15 ersichtlich.

Target Torque			
Funktionscode (binär - 4 Bit)	Geräteadresse (binär - 7 Bit)	ID (binär)	ID(dezimal)
1100	0000001	1100 0000001	1537
Datentyp	integer16		
Objectadresse (hexadezimal)	6071		
Werte	per thousand of the drive max. current		

Tabelle 3.15: Objektdaten Target Torque, vgl. CANopen Communication Profile [13]

Die Berechnung des Motordrehmoments erfolgt über die Motorkonstante des ausgewählten Motors der Type FP-3314. Die Drehmomentkonstante K_t beträgt $2,27Nm/A$. Die Berechnung erfolgt laut Formel (3.1), wobei M_a dem Motordrehmoment und I_c dem Controllerstrom entspricht.

$$M_a = I_c * K_t, \quad (3.1)$$

Die Übertragung des Drehmomentwerts erfolgt in Matlab Simulink in Funktionsblöcken die sich analog zu den bereits beschriebenen Funktionen verhalten. Auch hier können CAN-Kanäle erstellt, konfiguriert und gestartet werden. Auch die zu übertragenden CAN-Nachrichten können frei zusammengestellt werden, da die Übertragung der Nachrichten in einer Grundform von CAN erfolgt. In Abbildung 3.12 ist die Übertragung des Drehmomentwerts sowie der Sync-Nachricht aus Matlab Simulink ersichtlich. Hierbei wurde als erster Test eine Kennlinie mit linearem Anstieg des Drehmoments im Bezug auf die Pedalposition gewählt. Die Pedalposition muss dazu zuerst aus den 4 Byte-Werten der in Integer32 übertragenen CAN-Nachricht, erzeugt werden. Die Nullposition entspricht einem Absolutwert von 2850, dieser wird dem aktuellen Positionswert hinzugefügt, da die Bewegung in vom Motor aus gesehener negativer Richtung stattfindet. Dann wird der Wert skaliert und durch ein Sättigungsglied begrenzt. Danach wird der Wert in eine CAN-Nachricht gepackt und übertragen. Dazu muss der Wert in 2 Byte aufgeteilt werden.

Die Umrechnung des am Motor erzeugten Drehmoments in die am Pedal wirkende Kraft wurde in einem anderen Projekt am FTG [19] erarbeitet. Sie ergibt sich aus der Berechnung eines Vierbeingelenks. Diese Berechnung wurde dann in einem anderen Projekt [20] in die Simulationsumgebung einbezogen, so dass wiederum die Werte für die Drehmomentvorgabe bekannt sind.

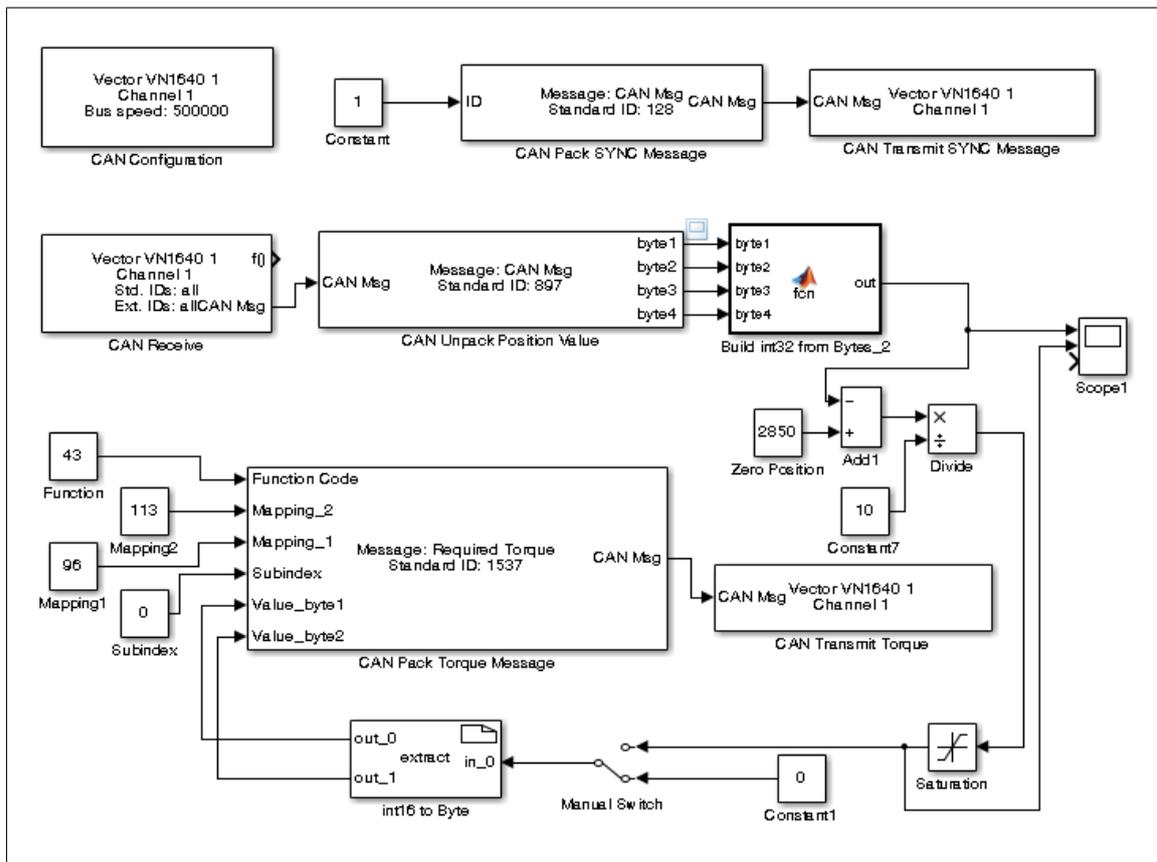


Abbildung 3.12: Matlab Simulink Model der Controlleransteuerung

3.2.4.7 Deaktivierung des Stromreglers mittels CANopen

Die Deaktivierung des Stromreglers erfolgt mittels des Stop Operation Segments der Datei "Startup.m" aus Anhang B. In diesem Segment werden die Transitionen 5, 6 und 7 aus Abbildung 3.11 durchgeführt. Dies entspricht einer ordnungsgemäßen Abschaltung des Stromreglers.

3.2.5 Diskussion und Ausblick

Die Implementation mittels eines rotatorischen Motors ergab sich durch die in der Industrie ausgiebig erprobte Technik, als erfolgreiche Lösung der Anforderungen. Sowohl die hohe Kraft als auch die Schnelligkeit von rotatorischen Motoren ist für die gestellten Anforderungen mehr als ausreichend. Durch die Anwendung des eingeschränkt echtzeitfähigen CAN-Netzwerks ermöglicht auch eine schnelle Ansprechzeit des Systems. Mittels Vehicle Network Toolbox war die Anbindung des CAN-Netzwerks bzw. des höherprioren CANopen Netzwerks in Matlab durch die selbstkonfigurierbaren Nachrichten möglich. CANopen bietet hierbei durch vordefinierte Prozessdaten- und Servicedatenobjekte eingebettet in ein definiertes Geräteprofil eine Verbindung mit weniger Overhead als jene mittels CAN.

Die gewählte Ansteuerung des Stromreglers entspricht dem eines "open Loop " Ansatzes, also ohne Rückführung des Ausgangssignals. Um die Genauigkeit der Ansteuerung zu steigern wurde bereits an einer "closed Loop " Variante gearbeitet. Diese Variante sieht einen Kraftsensor im Kraftübertragungsarm zum Pedal laut Abbildung 3.13 vor.

Simulationsrechner

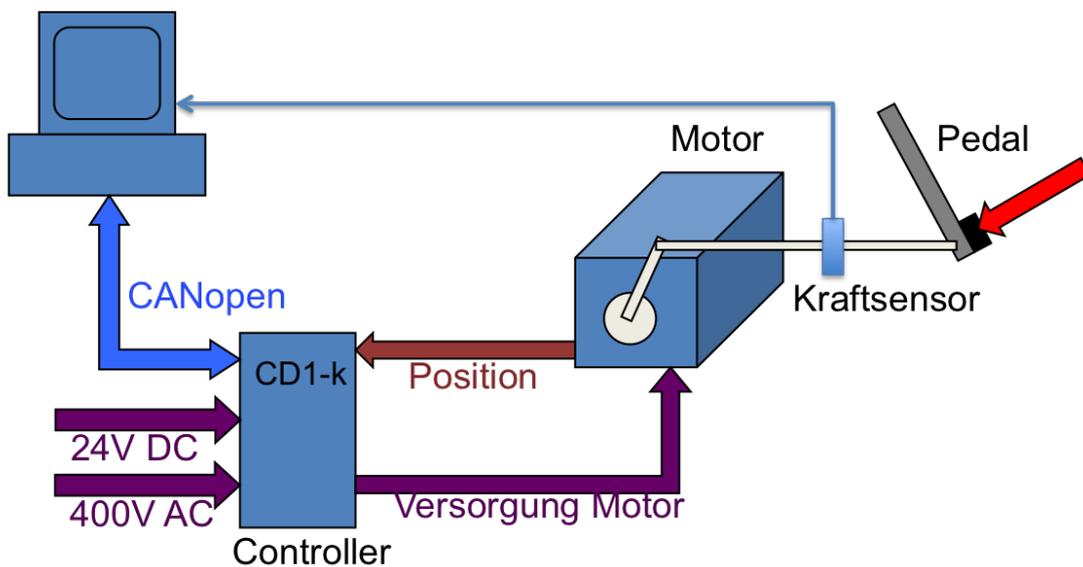


Abbildung 3.13: Motor-Controller System des Bremsaktuators mit rotatorischen Aktuator und Rückführung der Bremskraft

3.3 Kupplungspedal

3.3.1 Einleitung

Die Anforderungen an das Kupplungspedal sind ähnlich denen, die an das Bremspedal gestellt wurden. Auch hier sollte es möglich sein verschiedene Kennlinienverläufe aus der Simulation heraus zu produzieren. Einer der Unterschiede zum Bremspedal ist es, dass die Kräfte die beim Kupplungspedal auftreten kleiner sind als jene die bei einem Bremspedal auftreten. Sie liegen wie in [7] ersichtlich im Bereich von max. 500N. Da der Lösungsansatz mittels eines Aktuatorsystems wie es bei dem Bremspedal verwendet wird, jedoch kostenaufwendig ist, wurde hier im Zuge einer Bachelorarbeit an einer kostengünstigeren, mechanischen und passiven Lösungsvariante gearbeitet [21]. Da die Lösungsvariante mittels rotatorsichem Aktuator bereits ausreichend im Kapitel zur Bremse diskutiert wurde, setzt sich dieses Kapitel mit der Pedalpositionsmessung einer mechanischen Kupplungsvariante auseinander.

3.3.2 Positionsmessung

Die Kupplungspositionsmessung lässt sich im Grunde auf eine einfache Wegmessung zurückführen. Zur Wegmessung gibt es grundsätzlich zwei verschiedene Ansätze, die tastenden und die berührungslosen Wegaufnehmer. Da das Kupplungssystem sehr robust ist, werden im Folgenden die tastenden Wegaufnehmer verglichen und ein Lösungsansatz gewählt.vgl. [22]

Wegmessung mittels Widerstandsänderung

Die Wegmessung mittels Widerstandsänderung funktioniert nach einem relativ einfachen Prinzip. Sie besteht im Großen und Ganzen aus einem Seil und einem Drehpotentiometer. Die elektrische Nutzlänge von Drehpotentiometern bewegt sich laut [23] im Bereich von 5 bis 4000mm bzw. kann durch spezielle Ausführungsformen (Konstruktionen) einen Weg von bis zu 50 Metern messen. Das Prinzip einer solchen Wegmessung ist in Abbildung 3.14 zu sehen. Sie beruht auf der Messung eines Spannungsteilers. Der Weg ist hierbei proportional zum Spannungsverhältnis zwischen Versorgungsspannung und Messspannung. Der große Vorteil dieses Wegaufnehmers ist dabei die einfache Auswertung der Wegmessung durch eine Spannungsmessung. Der Nachteil des Systems ist, dass es zu Problemen mit dem Abrieb bei Regelkreisen mit geringer Regelschwingung kommen kann.

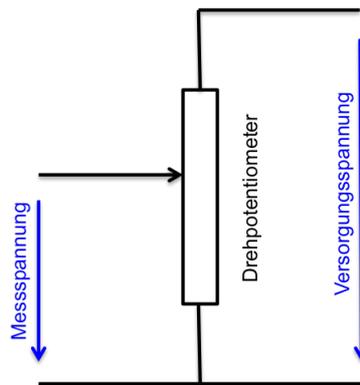


Abbildung 3.14: Messschaltung Wegmessung Drehpotentiometer

Induktive Wegmessung

Bei der induktiven Wegmessung wird der Effekt ausgenutzt, dass sich die Impedanz einer Spule ändert wenn ein ferromagnetischer Kern in die Spule eintaucht. Der häufigste Aufbau einer induktiven Wegmessung erfolgt durch den Aufbau einer Differentialdrossel. Eine Differentialdrossel besteht aus zwei symmetrisch gewickelten Spulen die mechanisch fest miteinander verbunden sind. Das Messobjekt (ein ferromagnetischer Kern) bewegt sich im Inneren der beiden Spulen. Er befindet sich in seiner Nullposition wenn er sich genau in der Mitte der beiden Spulen befindet. Wird der Kern aus der Nullposition gebracht, so taucht er tiefer in eine der beiden Spulen ein, während er aus der anderen austritt. Dies führt zu einer gegengleichen Änderung der Impedanzen. Durch diesen Aufbau ergibt sich eine sehr gute Temperaturstabilität. Die Messung der Impedanzänderung gestaltet sich für Differentialdrosseln im Gegensatz zur Messung mittels Widerstandsänderung komplexer. Der Aufbau einer Differentialdrossel ist in Abbildung 3.15 zu sehen. Der Vorteil der induktiven Wegmessung ist, dass sie auch unter widrigen Umgebungsbedingungen (Öl, Wasser oder ähnliches) eingesetzt werden können und eine hohe Genauigkeit besitzen. Der Nachteil ergibt sich aus der aufwändigeren Messung der Wegänderung.

Digital kodierte Längengeber

Mit kodierten Längengebern können Wegänderungen mit Hilfe von Rastern von einem festen Bezugspunkt aus gemessen werden. Das Raster kann mechanisch oder optisch abgetastet werden. Die mechanische Abtastung erfolgt durch Schleifkontakte und spannungsführende Leiterbahnen auf einem isolierten Untergrund. Hierbei sind die spannungsführenden Leiterbahnen das Raster.

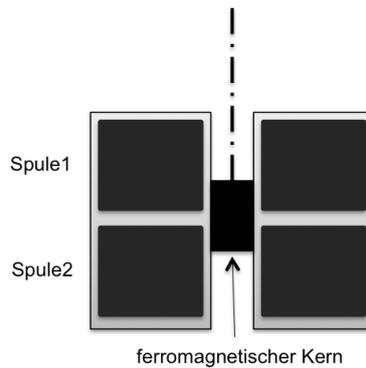


Abbildung 3.15: Aufbau einer Differentialdrossel im Schnitt

Bei optischer Abtastung bestimmt ein von Licht hinterleuchteter Lochraster die Position, welche von lichtempfindlichen Aufnehmern aufgenommen wird. Ein Beispiel für ein Raster ist in Abbildung 3.16 zu sehen. Bei beiden Lösungen ist die Genauigkeit der Messung von der Genauigkeit des Rasters abhängig. Die Auswertung der Position ist hierbei digital, was zu einem Quantisierungsfehler führt.

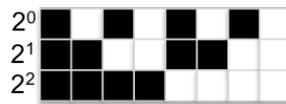


Abbildung 3.16: Raster für kodierte Längengeber im Binärcode

3.3.3 Konzept für Kupplungspositionsmessung

Nach der Recherche von möglichen Messmethoden wurde ein Konzept für den Aufbau der Messung im Fahrzeug erstellt. Der geometrische Aufbau der Messung ist in Abbildung 3.17 ersichtlich. Hierbei wird das Messinstrument an einem fixen Punkt montiert. Die Bewegung des Kupplungspedals wird parallel zum Hydraulikzylinder des Kupplungssystems mittels eines Seiles gemessen. Das Messinstrument kann für dieses Konzept beliebig gewählt werden.

3.3.4 Diskussion und Ausblick

Die Ausführung des Force Feedback der Kupplung kann grundsätzlich mit dem selben System ausgeführt werden, dass bereits bei der Bremse verwendet wird. Aufgrund des hohen Kostenaufwands einer solchen aktiven FFB Lösung wurde am FTG eine Bachelorarbeit zur Konstruktion eines passiven FFB Systems durchgeführt. Die Kupplungspositionsmessung lässt sich hierbei auf eine Wegmessung zurückführen. Es wurden mehrere Möglichkeiten betrachtet und ein Konzept zur Ausführung erstellt.

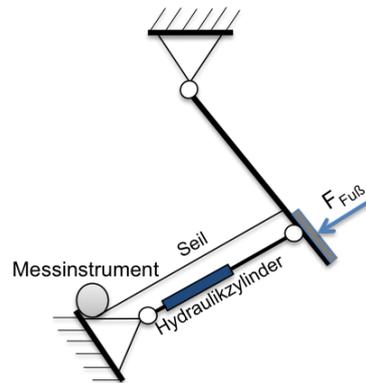


Abbildung 3.17: Konzept der Kupplungspedalpositionsmessung

Aufgrund zweier Faktoren wurde auf den Einbau des Kupplungspedals in den Fahrsimulator verzichtet. Der erste Grund liegt darin, dass Fahrzeughersteller vermehrt auf Fahrzeuge mit Automatikgangschaltung setzen und somit auch die Entscheidung gefällt wurde, den Fahrsimulator mit einer Automatikschaltung auszustatten. Weiters konnte mit dem passiven FFB System leider vorerst kein akkurater Kraftverlauf nachgestellt werden. Daher ist dieses Projekt noch in Bearbeitung. Sollte in Zukunft die Entscheidung fallen doch auf eine manuelle Gangschaltung umzustellen, gibt es zwei Konzepte zum Einbau des Kupplungspedals.

3.4 Gaspedal

3.4.1 Einleitung

FFB für Gaspedale ist nicht nur eine Thematik die es für Fahrsimulatoren zu betrachten gilt. Denn auch in der Fahrzeugindustrie werden immer seltener mechanischen Verbindungen zwischen Gaspedal und Motor wie etwa mittels eines Bowdenzugs mehr verwendet. Die Gründe hierfür liegen an der gestiegenen Komplexität des Gesamtsystems Fahrzeug. Die Aufgaben für das Motormanagement steigen mit den höheren Anforderungen an Abgasemission, Verbrauch und Sicherheit. Sicherheitssysteme wie ESP oder ASR greifen ebenso in die Motorsteuerung ein wie bei steigender Leistungsanforderung bei Einschalten eines Komfortsystems wie etwa einer Klimaanlage. Aus diesem Grund wird in der Fahrzeugindustrie schon seit ca. einem Jahrzehnt auf elektronische Gaspedale zurückgegriffen. Das Gaspedal dient hier als Eingangsgröße für den Drehmomentwunsch der Fahrerin bzw. des Fahrers. Dieser kann jedoch zusätzlich durch andere Systeme manipuliert werden. Wie in [24] beschrieben führt vor allem im Zusammenhang mit Hybridsystemen kein Weg mehr am elektronischen Gaspedal vorbei. Der FahrerInnenwunsch wird dann von einem der Antriebssysteme erfüllt. Im folgenden wird nun der Aufbau eines elektronischen Gaspedals betrachtet.

3.4.2 Elektronisches Gaspedal (Mini Countryman)

Der mechanische Aufbau des Gaspedals ist simpel gehalten. Das Force Feedback des Pedals ist meist nur eine lineare Feder, da dies dem Verhalten des früher verwendeten Bowdenzug zur Drosselklappe und der dort angebrachten Feder entspricht. Die Anbindung an das Motormanagement erfolgt nun, wie bereits erwähnt über ein elektrisches Signal. Dieses kann entweder durch ein rein analoges Signal oder eine Busanbindung übertragen werden. Beim betrachteten Modell des Mini Countryman, wird auf eine Busanbindung verzichtet. Das Messsignal der Pedalposition wird mittels analogem Signal weitergeleitet. Da es sich beim Gaspedal aber um ein sicherheitsrelevantes Bauteil handelt, wird das Signal redundant übertragen. Zu diesem Zweck sind im Gaspedalmodul zwei Pedalpositionsmessung mittels Widerstandsänderung verbaut. Der schematische Aufbau des gesamten Gaspedals ist in Abbildung 3.18 ersichtlich.

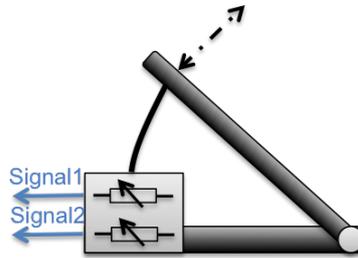


Abbildung 3.18: Schematische Darstellung des Gaspedalmoduls des Mini Countryman

3.4.3 Messung des Positionssignals

Da der grundsätzliche Aufbau und die Signalübertragung des Pedals bekannt sind, ist es nun die nächste Aufgabe die Signale richtig auszulesen und in die Simulationsumgebung einzubinden. Dazu musste ein schematisches Schaltbild der BMW AG mit Hilfe der in der DIN Norm 72552 definierten Klemmenbezeichnungen betrachtet werden. Das Schaltbild ist in Abbildung 3.19 ersichtlich.

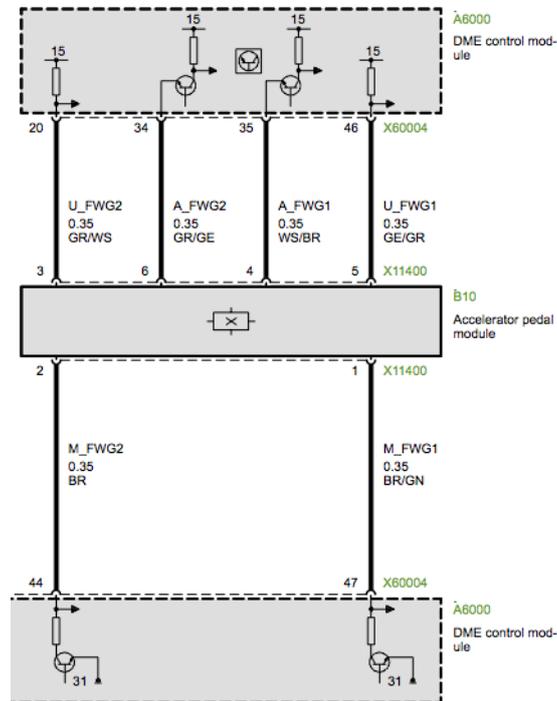


Abbildung 3.19: Schaltplan Gaspedalmodul; Quelle: BMW AG

Betrachtet man diesen Schaltplan nun mit der Information, dass die Klemmbezeichnung 15 den geschalteten Pluspol und die Klemmbezeichnung 31 der Rückleitung zum Minuspol bzw. der Fahrzeugmasse entspricht, so ergeben sich folgende Annahmen im Bezug auf das Gaspedalmodul (Accelerator Pedal Modul):

- Die erste Messschaltung ergibt sich aus den Pins 2, 3 und 6
- Die zweite Messschaltung ergibt sich aus den Pins 1, 4 und 5
- Die Pins 1 und 2 entsprechen der Rückleitung zum Minuspol bzw. der Fahrzeugmasse
- Die Pins 3 und 5 entsprechen der positiven Spannungsversorgung der Messschaltungen
- Die beiden Pedalpositionssignale liegen an den Pins 4 und 6

Nach Aufstellung dieser Annahmen wurden dieselben in einer Labormessung geprüft. Das Resultat bestätigte die Annahmen, dass die Messsignale an den Pins 4 und 6 des Gaspedalmoduls nach anlegen einer Versorgungsspannung von +5V Gleichspannung gemessen werden können.

3.4.4 Einbindung in die Simulationsumgebung

Für die Einbindung der Gaspedalstellung in die Simulationsumgebung wurde ein Datenerfassungssystem der Firma Dewetron verwendet. Hierbei handelt es sich um ein modulares Datenerfassungssystem, welches über USB mit der Simulationsumgebung kommuniziert. Das gewählte Model bietet 16 Steckplätze für verschiedene Messverstärkereinschubkarten. Die Einbindung in Matlab wurde hierbei zunächst mittels der Data Acquisition Toolbox von Matlab durchgeführt. Es stellte sich jedoch heraus, dass diese nicht mit der Echtzeitumgebung von Matlab kompatibel ist. Daher wurden im Zuge eines weiteren Projekts [20], der Implementierung der Simulationsumgebung, eigene Funktionen zur Anbindung der Messsignale ausgearbeitet.

Analoges Eingangsmodul DAQN-AIN

Das verwendete Modul ist ein analoges Eingangsmodul mit einem Messspannungsbereich von $\pm 10V$. Die Auflösung des nachgeschalteten A/D Wandlers beträgt 16 Bit. Zum Anschluss des Gaspedalmoduls an das Eingangsmodul wird ein Standard 9-pin Sub-D Stecker verwendet. Die Pinfunktionen sind in Tabelle 3.16 ersichtlich.

Daraus ergibt sich, dass das Modul nicht die benötigte +5V Spannungsversorgung liefert, die für den Betrieb des Gaspedalmoduls notwendig ist. Daher wurde die Spannungsversorgung von +9V (Pin 5) mittels eines Festspannungsreglers auf 5V begrenzt. Da es sich bei der Gaspedalpositionsmessung im Fahrsimulator nicht um eine sicherheitsrelevante Information wie im Fahrbetrieb im Fahrzeug handelt, wurde zunächst auf eine redundante Messung des Signals verzichtet. Der verwendete Messaufbau ist in Abbildung 3.20 zu sehen. Das Messsignal des Gaspedalmoduls bewegt sich im Bereich von 0,7 bis 4 Volt

Pin	Funktion
1	Input (+)
2	RS-485 A (wenn verfügbar)
3	RS-485 B (wenn verfügbar)
4	GND
5	+9V Spannungsversorgung
6	+12V Spannungsversorgung
7	Reserviert für Spezialfunktionen
8	Reserviert für zusätzliche Sensorspannungsversorgungen
9	-9V Spannungsversorgung

Tabelle 3.16: Pinbelegung 9-Pin Sub-D Stecker für DAQ-AIN; Quelle: Datenblatt DAQ-AIN

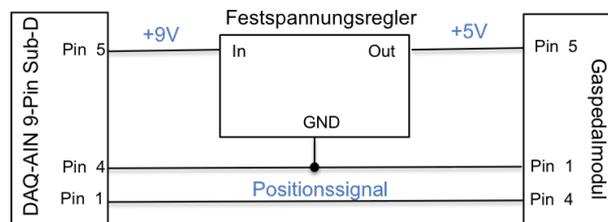


Abbildung 3.20: Schematischer Schaltplan Gaspedalmessung

3.4.5 Diskussion und Ausblick

Der technische Stand bei Fahrzeuggaspedalen ist bereits durch elektronische Gaspedalpositionsmessung gegeben. Daher wird im Fahrsimulator auch das Originalgaspedal des Mini Countryman eingesetzt. Mit Hilfe des Schaltplans des Gaspedalmoduls und den normierten Klemmbezeichnungen, wurden Messungen durchgeführt um die Gaspedalposition mittels Widerstandsänderung zu messen. Zur Anbindung an die Simulationsumgebung wurde das analoge Ausgangssignal des Gaspedalmoduls mit Hilfe eines Datenerfassungssystems der Firma Dewetron durchgeführt.

Wie im Tagungsband der Tagung zur Fahrerassistenz der TU München [25] zu sehen ist, wird bereits an zusätzlichen Möglichkeiten der Verwendung von FFB am elektronischen Gaspedal gearbeitet. Wie aus diesem Bericht hervorgeht liegt in der Unterstützung der Fahrerin oder des Fahrers durch haptisches Feedback am Gaspedal ein großes Potential. Die/Der FahrerIn reagiert auf haptisches Feedback besser als auf optisches Feedback. Daher wird auch schon von Automobilteileherstellern wie Continental an Gaspedalen mit FFB gearbeitet. Ein Beispiel hierfür ist das von Continental entwickelte elektronische Gaspedal AFFP [26]. Die Implementierung eines haptischen Gaspedals ist für die Zukunft vorgesehen.

3.5 Gangwahlhebel

3.5.1 Einleitung

Das FFB bei Gangwahlhebeln im Fahrzeug entsteht durch den mechanischen Aufbau der gesamten Gangschaltung. Man unterscheidet hier grundsätzlich zwischen manuellem Schaltgetriebe, in der theoretisch von jedem Gang in einen beliebigen anderen Gang geschaltet werden kann und dem Automatikgetriebe, bei dem die Gangwahl durch eine automatisches System durchgeführt wird. Bei beiden Systemen besteht nach technischen Stand eine mechanische Verbindung zwischen Wahlhebel und Getriebe. Die Entwicklung im Bereich Shift-by-Wire also der Gangwahl durch eine elektrisches Signal, steht erst am Beginn. Auch beim Gangwahlhebel wurde, wie auch bei der Kupplung, am FTG eine eigene Bachelorarbeit zur Nachbildung des Schaltsystems mittels passivem FFB durchgeführt [27]. Deshalb wird im nächsten Abschnitt ein Konzept zur Wahlhebelstellungsdetektion betrachtet.

3.5.2 Konzept zur Wahlhebeldetektion

Anders als bei den Positionsmessungen der Pedalerie ist die Detektion der Wahlhebelposition diskret. Die Gangwahl trifft nur bei einer bestimmten Position zu. Der Ganghebel kann sich nur nach vorgegebenen Bahnformen, wie in Abbildung 3.21 zu sehen, bewegen. Die Messung muss daher nur in den definierten Positionen stattfinden. Um diese Position zu messen, wurden folgende Möglichkeiten betrachtet.

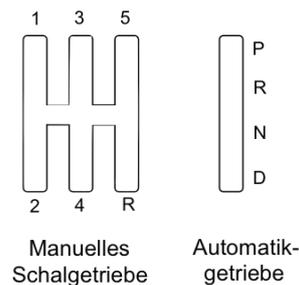


Abbildung 3.21: Bahnformen für Schalthebel von manuellen und automatischen Schaltgetriebe

Drucktaster

Eine Möglichkeit die Position des Ganghebels auszulesen, ist die Anbringung von Drucktastern an den Haltepositionen des Hebels. Diese werden so angebracht, dass die Schalter in den Haltepositionen aktiviert werden. Hier ergibt sich ein Nachteil dieses Konzepts, denn die Schalter müssen exakt positioniert werden damit die Schalter auch wirklich aktiviert werden. Ein weiterer nachteiliger Aspekt ist, dass die Positionierung der Taster beim Automatikgetriebe noch schwieriger als beim manuellen Getriebe ist.

Berührungsloser Schalter

Eine weitere Möglichkeit sind berührungslose Schalter. Hierbei gibt es zwei mögliche Ausführungsformen: induktive bzw. kapazitive Sensoren. Induktive Sensoren arbeiten ähnlich wie Relais, wo eine von Strom durchflossene Spule ein Magnetfeld erzeugt um einen ferromagnetischen Schalter zu schließen. Der induktive Sensor arbeitet aber mit einem Permanentmagnet, der den ferromagnetischen Schalter schließt, sobald er in die Nähe dessen gebracht wird. Beim kapazitiven Sensor wird die Kapazität einer Elektrode gegenüber dem Erdpotential gemessen. Durch die Annäherung eines Gegenstands wird diese Kapazität verändert und durch eine Messelektronik ausgewertet. vgl. [22].

Der Aufbau des Konzepts ergibt sich hier ähnlich wie beim Drucktaster. Die Sensoren werden auch hier an den Haltepositionen des Wahlhebels angebracht. Der einzige Unterschied ist hierbei, dass bei der Verwendung von induktiven Sensoren ein Magnet am Wahlhebel angebracht werden muss. Der Vorteil der Verwendung von berührungslosen Sensoren ist, dass die Positionierung der Sensoren nicht so exakt sein muss.

Digitales Eingangsmodul PAD-DI8

Das Einlesen der verschiedenen Wahlhebelpositionen funktioniert mit Drucktastern und berührungslosen Schaltern gleich. Verwendet wird hierzu das digitale Eingangsmodul PAD-DI8, welches ein Steckmodul des bereits beim Gaspedal verwendeten Datenerfassungssystems ist. Das Modul hat 8 Digitale Eingänge, wobei 2 davon eine gemeinsame Basis (GND) besitzen. Das Modul bietet auch eine Spannungsversorgung von 12V Gleichspannung. Als Steckverbindung wird ein 25-Pin Sub-D Stecker verwendet. Die Pinanordnung des Moduls ist in Tabelle 3.17 ersichtlich. Die einzelnen Schalter werden dann an die verschiedenen digitalen Eingänge angeschlossen um einen geschlossenen Schalter zu detektieren. Somit werden die Eingänge den Wahlhebelpositionen zugewiesen. Die Beschaltung eines Eingangs wird exemplarisch in Abbildung 3.22 dargestellt.

Da an den digitalen Eingängen Optokoppler zur galvanischen Trennung verwendet werden, entspricht ein offener Schalter einer logischen Null.

Pin	Funktion	Pin	Funktion
1	IN0 (+)	13	IN6 (+)
2	IN0 (-)	14	IN6/IN7 GND
3	IN1 (+)	15	IN7 (+)
4	IN1 (-)	16	Nicht verwendet
5	IN2 (+)	17	Nicht verwendet
6	IN2 (-)	18	Nicht verwendet
7	IN3 (+)	19	Nicht verwendet
8	IN3 (-)	20	+12V Spannungsversorgung
9	IN4 (+)	21	Init (Kontakt zur Modulation der Baudrate)
10	IN4 (-)	22	GND / Init
11	IN5 (+)	23	Nicht verwendet
12	IN5 (-)	24	Nicht verwendet
		25	Nicht verwendet

Tabelle 3.17: Pinbelegung 25-Pin Sub-D Stecker für PAD-DI8; Quelle: Datenblatt PAD-DI8

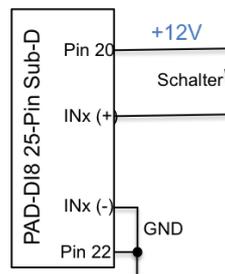


Abbildung 3.22: Exemplarische Eingangsbeschaltung PAD-DI8

Da es beim Projekt zur Anfertigung des FFB-Systems zu Problemen mit der Haptik des Systems kam und bereits ein Zeitdruck zur Einbindung des Wahlhebels bestand, wurde auch hier entschieden den Originalhebel des Mini Countryman zu verwenden.

3.5.3 Wahlhebelmodul des Mini Countryman

Auch beim Wahlhebelmodul wird zuerst die Schaltung des Wahlhebelmoduls unter Rücksichtnahme auf die normierten Klemmbezeichnungen betrachtet. Diese ist in Abbildung 3.23 ersichtlich. Da die Verbindung zwischen Wahlhebel und Getriebe hier mittels Bowdenkabel ausgeführt war, musste ein anderer Weg zur Detektierung der Wahlhebelposition gesucht werden. Hierbei liegt das Augenmerk auf dem Modul S227 (dem Wahlhebelmodul)

und dem Modul E82, welches die aktuelle Schaltposition via LED anzeigt. Beim Betrachten der Schaltung wurden folgende Annahmen aufgestellt:

- Die Spannungsversorgung von +12V liegt an den Pins 13 und 14 des Moduls S227
- Die Fahrzeugmasse bzw. Rückleitung zum Minuspol liegt an Pin 6 des Moduls S227
- Die Fahrzeugmasse bzw. Rückleitung zum Minuspol liegt an Pin 2 des Moduls E82
- Die Spannungsversorgung für das Modul E82 liegt an Pin 3 des Moduls und kommt vom Modul S227 (Pin 12)

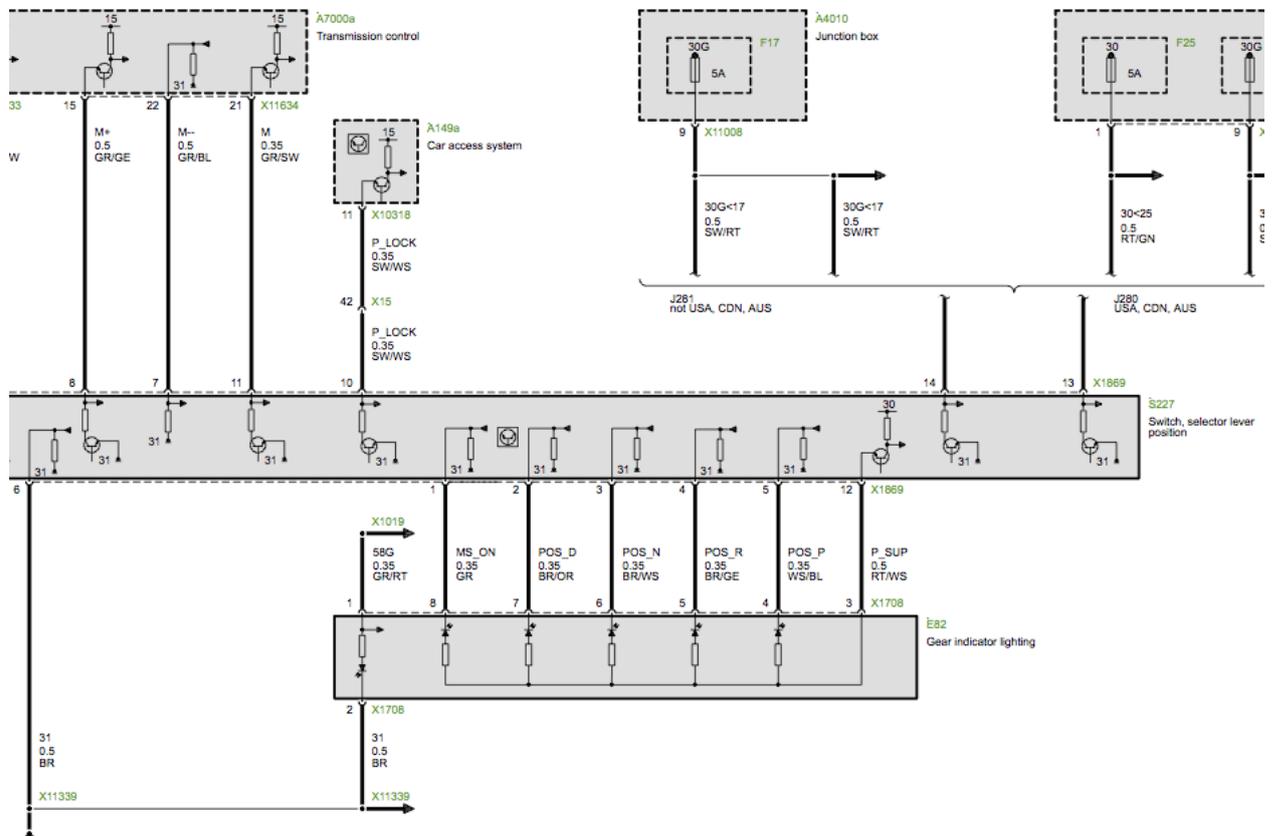


Abbildung 3.23: Auszug aus dem Schaltplan des Wahlhebelmoduls; Quelle: BMW AG

Um oben genannte Annahmen zu überprüfen, wurden die beiden Module laut Schaltplan verbunden und mit Spannung versorgt. Es stellte sich heraus, dass die Annahmen zutrafen. Danach wurden die Spannungsabfälle an den Dioden mittels Voltmeter gemessen. Da sich der Spannungsabfall bei aktiviertem LED klar vom inaktiven Wert unterscheidet, wurde entschieden die Detektion der Wahlhebelposition über eine Spannungsmessung an den LED's durchzuführen.

3.5.4 Einbindung in die Simulationsumgebung

Die Spannungsabfälle der LED's variierten dabei bei aktiven LED zwischen 1,8 und 3,1 Volt. Die Spannungsabfälle bei deaktiviertem LED zwischen 6 und 10 Volt. Zur Messung der Spannungsabfälle wurde wieder das Modul DAQN-AIN, das bereits im Abschnitt 3.4.4 erklärt wurde, verwendet. Die Spannungsversorgung wird mittels einer Spannungsversorgung von 12V, die auch für die fahrzeuginternen Systeme benutzt wird, hergestellt. In der Simulationsumgebung wird der gemessene Spannungsabfall mit dem gewählten Entscheidungswert von 4V verglichen, um die Wahlhebelposition zu detektieren. Die schematische Darstellung der Messschaltung ist in Abbildung 3.24 zu sehen.

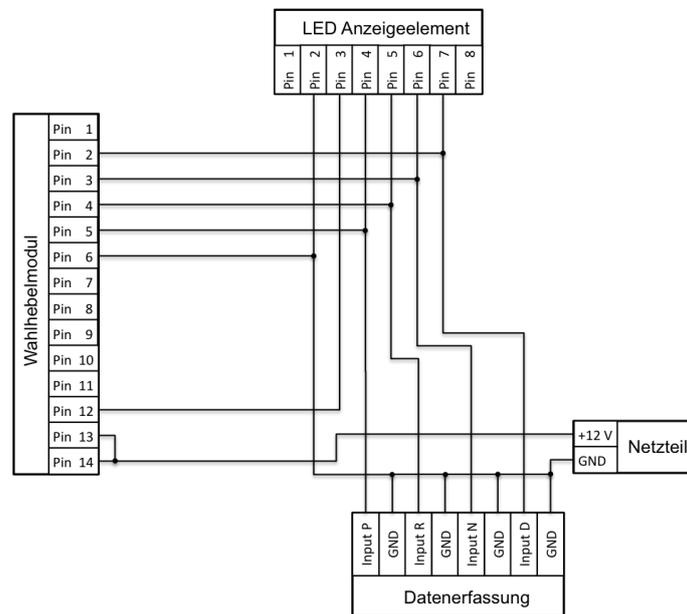


Abbildung 3.24: Schematischer Schaltplan Wahlhebelmodulmessung

3.5.5 Diskussion und Ausblick

Zur Simulation des FFB des Wahlhebels wurde am FTG eine Bachelorarbeit bezüglich eines mechanischen Konzepts durchgeführt. Deshalb wurde zuerst ein Konzept zur Detektion der Wahlhebelposition einer mechanischen FFB Variante erstellt. Dieses enthält entweder Drucktaster oder berührungslose Schalter und wird mit digitalen Eingängen des Moduls PAD-DI8 eingelesen. Da das vorläufige mechanische Konzept eine unzureichende Haptik aufwies um den im Projekt gestellten Anforderungen genüge zu tun und bereits Zeitdruck bestand, wurde auch hier das originale Wahlhebelmodul des Mini Countryman der mechanischen Lösung vorgezogen.

Um die Wahlhebelpositionsdetektion durchzuführen, musste der Schaltplan des Wahlhebelmoduls betrachtet werden. Danach wurde entschieden, dass mittels Messung des Spannungsabfalls an den Positions-LED's, die die Wahlhebelposition im Fahrzeuginnenraum anzeigen, die Wahlhebelposition detektiert werden kann. Der Spannungsabfall der LED's wird hier mit dem Modul DAQN-AIN, welches auch beim Gaspedal verwendet wird, gemessen.

Das Konzept mittels Originalteil und Spannungsmessung ergibt eine realitätsnahe Lösung des Gangwahlhebels, da hier mit wenig mechanischen Aufwand das originale Fahrzeug nachgebildet werden kann. Da zum Zeitpunkt der Erstellung des Systems, genügend analoge Eingangsmodule zur Verfügung standen, wurde für jedes LED eine Spannungsmessung ausgeführt. In Zukunft könnte eine elektronische Komparatorschaltung für jedes LED erstellt werden, um auch für die Lösung mittels originalen Wahlhebelmoduls das Einlesen mittels digitalen Eingängen zu ermöglichen. So könnten Steckplätze im Datenerfassungssystem eingespart werden.

3.6 Lenkrad

3.6.1 Einleitung

Durch das FFB am Lenkrad sollen die Kräfte, die durch die Lenkungsmechanik in Form von Rückstellmoment am Lenkrad ankommen, simuliert werden. Natürlich sind hier auch andere Kräfte die etwa durch Bordsteinberührungen aufkommen mitinbegriffen. Das Aktuatorsystem des Lenkrads wird durch den Projektpartner Steer-by-Wire (SBW), einem langjährigen Hersteller von Steer-by-Wire Systemen im Schiffsbau, bereitgestellt. Dieses System wird nun betrachtet.

3.6.2 Lenkradaktuatorsystem SBW

Das von SBW verwendete Lenkradaktuatorsystem ist dem der Bremse ähnlich. Auch hier wird ein Motor-Controller System verwendet. Als Controller dient hier ein Gleichspannungs-Antriebsregler mit CAN-Anbindung. Auf diesem Controller läuft die von SBW erstellte und proprietäre Regelungssoftware. Der verwendete Motor ist ein BLDC Motor. Dieser bringt mit Hilfe eines Getriebes das geforderte Drehmoment auf die Ausgangswelle. Der schematische Aufbau des Systems ist in Abbildung 3.25 ersichtlich. Um die Sicherheit der Positionsermittlung zu erhöhen, wird die Motorposition in diesem System redundant ermittelt. Zum einen mittels Hall-Sensorik und zum anderen mittels eines absolut Sensors mit CAN-Anbindung. Der Controller liest das Positionssignal des absolut Sensors über den erwähnten CAN-Bus ein. Das ganze System wird mit einer Versorgungsspannung von 24V Gleichspannung versorgt, wobei hier die gesamte Energie über den Controller geliefert wird. Das maximale Drehmoment des Gesamtsystems beträgt bei einer Stromaufnahme von 7A, welcher max 3 Sekunden fließen darf, 12,5 Nm. Das geforderte Drehmoment wird durch ein berechnetes analoges Spannungssignal an das System angelegt.

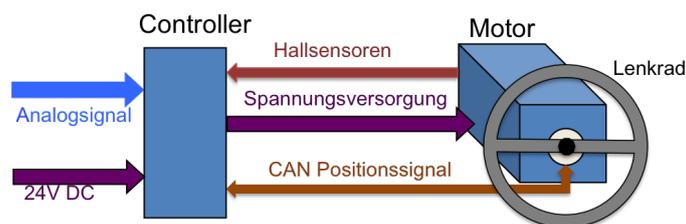


Abbildung 3.25: Schema Lenkradsystem SBW

3.6.3 Schaltungsaufbau mit Sicherheitseinrichtungen

Um das Lenkradaktuatorsystem bzw. die/den BenutzerIn des Lenkrads zu schützen, wurden mehrere Maßnahmen durch einen geeigneten Schaltungsaufbau realisiert.

Verpolungsschutz

Um den Motor gegen Verpolung der Versorgungsspannung zu schützen, wurde ein Verpolungsschutz mittels Diode in Sperrrichtung in Verbindung mit einer Sicherung verwendet. Diese Schaltung hat den Vorteil, dass sie auch einen Schutz gegen Überlast beinhaltet. Die Schaltung ist in Abbildung 3.26 dargestellt. Als Sicherung wird eine KFZ-Sicherung mit Auslösestrom von 7,5A gewählt. Die Funktion ist hierbei wie folgt vgl. [28]:

- Spannungsversorgung richtig gepolt: Die Diode ist in Sperrrichtung und die gesamte Versorgungsspannung liegt an. Der Strom fließt durch die Sicherung.
- Spannungsversorgung ist falsch gepolt: Die Diode ist in Durchlassrichtung und hat einen Spannungsabfall je nach Diode von maximal etwa 0,7V. Dadurch fließt ein Strom durch die Sicherung der diese zerstört und die Spannung trennt.

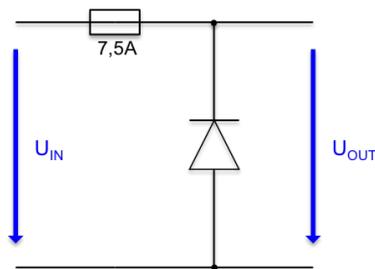


Abbildung 3.26: Verpolungsschutz mittels Diode in Sperrrichtung; vgl. [28]

Pufferbatterie

Da im Leistungsdatenblatt der Firma SBW eindeutig darauf hingewiesen wird, dass es durch den Betrieb des Lenkradsystems zu Spannungsspitzen kommen kann, wird empfohlen eine Batterie als Puffer vorzusehen. Für das verwendete System wurden zwei 12V Batterieblöcke in Serie verwendet.

Stop-Taster

Durch die genannten Pufferbatterien kam es zu einem weiteren Problem mit der Spannungsversorgung. Denn sollte die Spannungsversorgung abgeschaltet sein, wird die Schaltung nun von den beiden Pufferbatterien gespeist. Daher war es notwendig mittels eines

Relais und Stop-Taster (Taster mit Einrastfunktion) eine Trennung der Batterien vom System auszuführen. Das Relais wird benötigt um den maximalen Strom des Systems von 7A schalten zu können. Hierzu wurde der Taster direkt neben der Fahrerin bzw. dem Fahrer platziert um ihn im Falle einer Fehlfunktion betätigen zu können. Der Schaltungsaufbau ist in Abbildung 3.27 zu sehen.

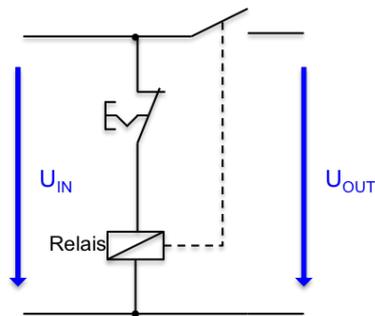


Abbildung 3.27: Relaisschaltung zur Trennung des Systems von der Spannungsquelle

Gesamtschaltung

Im Gesamtsystem wurde zusätzlich noch ein weiterer Schalter mit zwei Wechselkontakten eingebaut. Dies hatte zwei Gründe: Zum Einen sollte damit das Lenkradsystem von der Versorgungsspannung getrennt werden. Zum Anderen werden damit die Pufferbatterien vom Netzteil getrennt, damit andere Geräte die an das selbe Netzteil angeschlossen werden, nicht von den Pufferbatterien gespeist werden. Die Gesamtschaltung ist in Abbildung 3.28 ersichtlich.

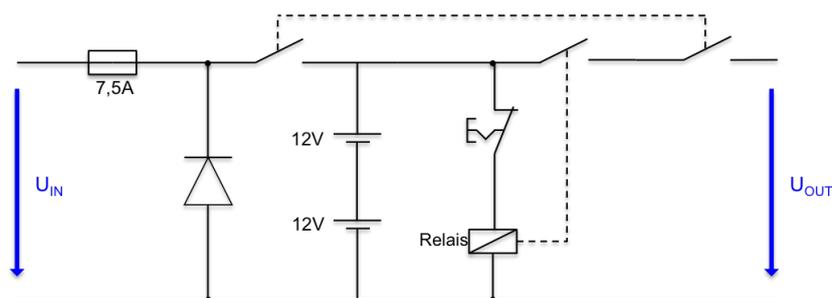


Abbildung 3.28: Gesamtschaltung der Spannungsversorgung des Lenkradsystems

3.6.4 Einbindung in die Simulationsumgebung

In diesem Abschnitt wird die Einbindung des Lenkradaktuatorsystems in die Simulationsumgebung betrachtet. Zunächst musste ein Weg gefunden werden um die Lenkradposition

einzu lesen. Die Lösung dieses Problems war die Anbindung des Simulationsrechners an den CAN-Bus des Lenkradsystems, da ja bekannt war, dass der zusätzliche Positionssensor den Positionswert zyklisch an den Motorcontroller sendet. Dazu musste zunächst der Aufbau der gesendeten Nachricht betrachtet werden. Der Aufbau der Nachricht ist in Tabelle 3.19 ersichtlich. Ein Unterschied zu den bisher betrachteten Nachrichten ist, dass der CAN-Extended Identifier verwendet wird. Die Datenlänge beträgt jedoch ebenfalls 8 Byte. Die Auflösung des Absolutensors beträgt 1024 Punkte pro Umdrehung und hat das Datenformat integer16.

ID (hex)	ID (dez)	1.Byte	2.Byte	3.Byte - 8.Byte
18F01DE4	418389476	unteres Byte (int16)	oberes Byte (int16)	nicht betrachtet

Tabelle 3.18: Aufbau CAN-Nachricht der Lenkradposition

Die Einbindung in Matlab Simulink funktioniert analog zum Einlesen der Position des Bremspedals. Es wird wieder mittels der Funktionen der VNT die CAN-Verbindung gestartet. Hier muss darauf geachtet werden, dass die Busgeschwindigkeit, die vom Lenkradsystem verwendet wird, 250kBit/s beträgt. Danach wird die Nachricht mittels CAN-Unpack ausgelesen. In dem Funktionsblock CAN-Unpack wird für das 2.Byte ein Multiplikationsfaktor von 256 eingestellt, damit das aktuelle (immer positive) Positionssignal durch einfache Addition ausgelesen werden kann. Der Simulink Empfangsblock ist in Abbildung 3.29 zu sehen. Danach kann in der Simulationsumgebung mit dem Positionssignal das erforderliche Drehmoment berechnet werden.

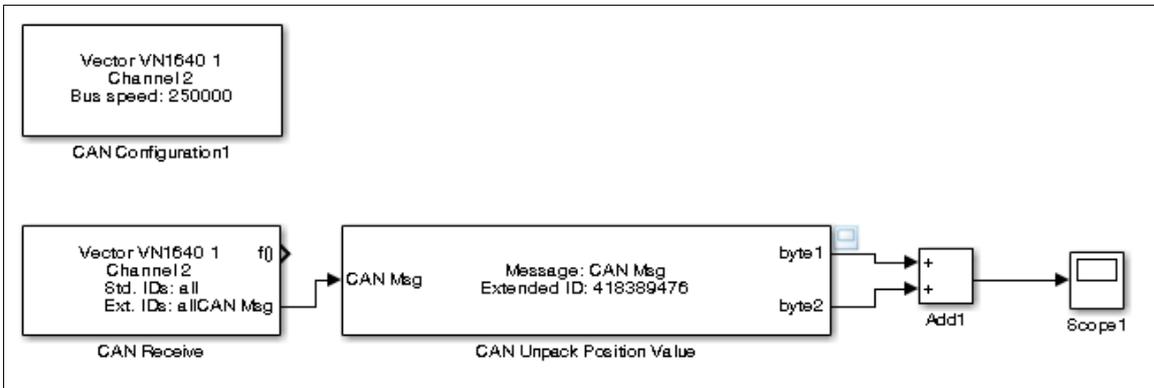


Abbildung 3.29: Simulink Modell der Positionsabfrage des Lenkrads

3.6.5 Vorgabe des Drehmoments

Die Vorgabe des Drehmoments erfolgt wie in Abbildung 3.25 dargestellt über ein analoges Eingangssignal. Das Drehmoment folgt hier einer analogen Eingangsspannung und kann in

Richtung der Lenkrad Null-Position aufgebracht werden. Das erforderliche Drehmoment wird in der Simulationssoftware aus den Fahrzeugdaten errechnet und anschließend mittels linearer Interpolation laut Tabelle 3.19 in das erforderliche Spannungssignal umgerechnet.

Drehrichtung links ←			Drehrichtung rechts →		
Spannung Analogeingang	Strom	Lenk- moment	Spannung Analogeingang	Strom	Lenk- textbfmoment
4 V	1390 mA	2,5 Nm	6 V	1390 mA	2,5 Nm
3,5 V	2085 mA	3,25 Nm	6,5 V	2085 mA	3,25 Nm
3 V	2780 mA	5 Nm	7 V	2780 mA	5 Nm
2,5 V	3475 mA	6,25 Nm	7,5 V	3475 mA	6,25 Nm
2 V	4170 mA	7,5 Nm	8 V	4170 mA	7,5 Nm
1,5 V	4865 mA	8,75 Nm	8,5 V	4865 mA	8,75 Nm
1 V	5560 mA	10 Nm	9 V	5560 mA	10 Nm
0,5 V	6255 mA	11,25 Nm	9,5 V	6255 mA	11,25 Nm
0 V	6950 mA	12,5 Nm	10 V	6950 mA	12,5 Nm

Tabelle 3.19: Zusammenhang zwischen analogem Eingangssignal und resultierendem Drehmoment; Quelle Datenblatt Lenkradsystem SBW

DAQN-V-OUT

Für die Vorgabe des gewünschten Drehmoments in Form eines analogen Spannungssignals wird ebenfalls das bereits mehrfach erwähnte Datenerfassungssystem der Firma Dewetron verwendet. Das verwendete Modul heißt DAQN-V-OUT und generiert ein Ausgangssignal im Bereich von -10V bis +10V. Auch hier wurde im Zuge eines anderen Projekts [20] die Anbindung in die Echtzeitumgebung durchgeführt. Das Modul benutzt wiederum einen 15-Pin Sub-D Stecker mit Pinbelegung laut Tabelle 3.20 um das analoge Spannungssignal bereit zu stellen. Das analoge Spannungssignal zur Drehmomentanforderung liegt zwischen den Pins 7 (OUT+) und 8 (OUT-) an.

3.6.6 Diskussion und Ausblick

Für die Simulation der Kräfte, die von der Lenkungsmechanik auf das Lenkrad übertragen werden, wird FFB benötigt. Das System das hier für die Ausführung des FFB verwendet wird, wurde von dem Projektpartner SBW zur Verfügung gestellt. Es handelt sich um ein Motor-Controller System mit analogem Eingangssignal zur Drehmomentsteuerung.

Für die Berechnung des erforderlichen Drehmoments wird die Position des Lenkrads benötigt. Diese wird mittels der am CAN-Bus liegenden Positionsnachricht des zusätzlichen Absolutensors ausgelesen. Daraufhin wird mit der Umrechnungstabelle der aus der Fahrzeugsimulation vorgegebene Drehmomentwert, in den zugehörige Spannungswert

Pin	Funktion
1	Nicht verwendet
2	Nicht verwendet
3	Nicht verwendet
4	GND (nicht isoliert)
5	+9V Spannungsversorgung (nicht isoliert)
6	Nicht verwendet
7	OUT+ (-10V bis +10V, isoliert)
8	OUT- (-10V bis +10V, isoliert)
9	-9V Spannungsversorgung (nicht isoliert)

Tabelle 3.20: Pinbelegung 9-Pin Sub-D Stecker für DAQ-V-OUT; Quelle: Datenblatt DAQ-V-OUT

umgewandelt. Dieser wird dann mit dem Modul DAQN-V-OUT an das Lenkradsystem angelegt. In der Schaltung für die Spannungsversorgung des Systems wurden ein Verpolungsschutz, Pufferbatterien und ein Taster zur Abschaltung des Systems miteinbezogen um das System und die/den FahrerIn zu schützen.

Das FFB am Lenkrad kann, wie auch beim Gaspedal, dazu verwendet um der/den FahrerIn ein weiteres haptisches Feedback zu geben. Hierzu wird laut [29] haptisches Feedback am Lenkrad bereits von Spurhalteassistenten benutzt, um die/den FahrerIn auf das Fahren am Fahrbahnstreifen aufmerksam zu machen. Weiters wird schon an Steer-by-Wire Lösungen für Fahrzeuge gearbeitet, bei denen keine mechanische Verbindung zwischen Lenkrad und Reifen besteht. Hier gibt es noch viele Möglichkeiten das haptische Feedback am Lenkrad zu benutzen.

Kapitel 4

Zusammenfassung

Fahrsimulatoren finden heute in sehr vielen Anwendungsgebieten ihren Einsatz. Der Grund dafür ist immer das Schaffen einer virtuellen Realität. In der Forschung bieten sich Fahrsimulatoren zur Untersuchung der Mensch-Maschine-Schnittstelle an. Weiters wird dadurch die Entwicklung von Fahrerassistenz- bzw. Sicherheitssystemen unterstützt, da kritische Situationen ohne Gefahr für die Probandin bzw. den Probanden durchgeführt werden können. Die Ausführungsarten von Fahrsimulatoren reichen vom einfachen PC System bis hin zum Motion Base Simulator. Die modernste und aufwendigste Art von Fahrsimulatoren sind Motion Base Simulatoren, bei denen meist ein vollständiges Fahrzeugmockup auf einer bewegbaren Basis aufgebaut wird. Diese stellen den derzeitigen Stand der Technik dar.

Auf Basis einer State-of-the-Art Recherche wurde auf dem FTG ein Fahrsimulator im Zuge des *Muegen Driving* Projekts gebaut. Um der/dem FahrerIn ein möglichst realitätsnahes FFB auf dem Bremspedal zu geben, ist es unerlässlich zuvor Messungen an einer echten Bremshydraulik durchzuführen. Aus diesem Grund wurden am FTG Messungen am Bremspedal eines Fahrzeugs durchgeführt und damit die Anforderungen an das FFB-System formuliert. Aufgrund von diesen Anforderungen wurden drei verschiedene Systeme betrachtet. Mittels Nutzwertanalyse ist dann ein rotatorisches Motor Controller System mit CANopen Netzwerkanbindung als beste Lösung ausgewählt worden.

CANopen ist ein auf CAN basierendes höherschichtiges Netzwerkprotokoll und wird hauptsächlich in der Automatisierungstechnik eingesetzt. Es bietet standardisierte Kommunikationsobjekte: Prozessdatenobjekte zur Übertragung von Echtzeitdaten und Servicedatenobjekte zur Konfiguration der CANopen-Geräte. Der CANopen Bus ist auf 127 Teilnehmer begrenzt und unterstützt eine Datenübertragungsrate von bis zu 1Mbit/s und verwendet ein Objektverzeichnis um die Funktionalität eines Gerätes zu beschreiben. Die Implementation ergab sich als erfolgreiche Lösung der gestellten Anforderungen. Sowohl die hohe

Kraft als auch die Schnelligkeit von rotatorischen Motoren ist für die gestellten Anforderungen mehr als ausreichend. Mittels Vehicle Network Toolbox war die Anbindung des CANopen-Netzwerks in Matlab durch die selbstkonfigurierbaren Nachrichten möglich.

Die Anforderungen an das Kupplungspedal sind ähnlich denen, die an das Bremspedal gestellt wurden. Der größte Unterschied zum Bremspedal ist, dass die auftretenden Kräfte kleiner sind. Sie liegen im Bereich von max. 500N. Hier wurde im Zuge einer Bachelorarbeit an einer mechanischen und passiven Lösungsvariante gearbeitet [21]. Daher ist es nur notwendig die Position der Kupplung in die Simulationsumgebung einzulesen. Es wurden mehrere Möglichkeiten betrachtet und ein Konzept zur Ausführung erstellt.

FFB für Gaspedale ist nicht nur eine Thematik die es für Fahrsimulatoren zu betrachten gilt. Denn auch in der Fahrzeugindustrie wird immer öfter auf mechanische Verbindungen zwischen Gaspedal und Motor verzichtet. Die Gründe hierfür liegen an der gestiegenen Komplexität des Gesamtsystems Fahrzeug. Zur Anbindung an die Simulationsumgebung wurde das analoge Ausgangssignal des originalen elektronischen Gaspedalmoduls des Mini Countryman mit Hilfe eines Datenerfassungssystems der Firma Dewetron durchgeführt.

Das FFB bei Gangwahlhebeln im Fahrzeug entsteht durch den mechanischen Aufbau der gesamten Gangschaltung. Man unterscheidet hier grundsätzlich zwischen manuellem Schaltgetriebe und dem Automatikgetriebe. Bei beiden Systemen besteht nach technischem Stand eine mechanische Verbindung zwischen Wahlhebel und Getriebe. Auch beim Gangwahlhebel wurde am FTG eine eigene Bachelorarbeit zur Nachbildung des Schaltsystems mittels passivem FFB durchgeführt [27]. Es wurde ein Konzept zur Detektion der Wahlhebelposition der mechanischen FFB Variante erstellt. Dieses enthält entweder Drucktaster oder berührungslose Schalter und wird mit digitalen Eingängen eingelesen. Da das vorläufige mechanische Konzept Probleme mit der Haptik aufwies, wurde auch hier das originale Wahlhebelmodul des Mini Countryman der mechanischen Lösung vorgezogen. Um die Wahlhebelpositionsdetektion durchzuführen, wurde eine Messung des Spannungsabfalls an den Positions-LED's, die die Wahlhebelposition im Fahrzeuginnenraum anzeigen, gewählt.

Durch das FFB am Lenkrad sollen die Kräfte, die durch die Lenkungsmechanik in Form von Rückstellmoment am Lenkrad ankommen, simuliert werden. Das Aktuatorssystem des Lenkrads wird durch den Projektpartner *Steer-by-Wire (SBW)*, einem langjährigen Hersteller von *Steer-by-Wire* Systemen im Schiffsbau, bereitgestellt. Das von SBW verwendete Lenkradaktuatorsystem ist ein Motor-Controller System. Als Controller dient hier ein Gleichspannungs-Antriebsregler mit CAN-Anbindung auf dem die proprietäre Rege-

lungssoftware läuft. Der verwendete Motor ist ein BLDC Motor. Dieser bringt mit Hilfe eines Getriebes das geforderte Drehmoment auf die Ausgangswelle. Das geforderte Drehmoment, wird durch ein berechnetes analoges Spannungssignal an das System angelegt.

Für die Berechnung des erforderlichen Drehmoments wird die Position des Lenkrads benötigt. Diese wird mittels der am CAN-Bus liegenden Positionsnachricht eines zusätzlichen Absolutensors ausgelesen. Daraufhin wird mit einer linear interpolierten Umrechnungstabelle der aus der Fahrzeugsimulation vorgegebene Drehmomentwert, in den zugehörige Spannungswert umgewandelt. Dieser wird dann mit dem Modul DAQN-V-OUT an das Lenkradsystem angelegt. In der Schaltung für die Spannungsversorgung des Systems wurden ein Verpolungsschutz, Pufferbatterien und ein Taster zur Abschaltung des Systems miteinbezogen um das System und die Fahrerin bzw. den Fahrer zu schützen.

Kapitel 5

Ausblick

Das Potential von Fahrsimulatoren ist sehr hoch. Sie werden zukünftig noch häufiger in der Entwicklung von Assistenzsystemen zur Anwendung kommen als bereits bisher. In Zukunft wird auch die Entwicklung von sicherheitsrelevanten Systemen vermehrt an Fahrsimulatoren durchgeführt. Da Fahrsimulatoren als Werkzeuge zur Untersuchung der Mensch-Maschinen-Kommunikation benutzt werden können, können auch Entwicklungen das Gesamtfahrzeug betreffend durch Fahrsimulatoren getestet werden. So wird der Fahrsimulator zu einem der wichtigsten Werkzeuge in der modernen Fahrzeugentwicklung.

Für zukünftige Aufgaben wird geplant, den bereits bestehenden Fahrsimulator auf eine Bewegungsplattform zu platzieren. Im Bremssystem gibt es bezüglich der Steuerung des Systems einen Ansatz um den bisherigen open-loop Ansatz mittels einer Kräftemessung am Pedal zu einem closed-loop Ansatz zu verbessern.

Beim Gaspedal gibt es bereits Ansätze auch hier ein haptisches Feedback an die/den FahrerIn weiter zu geben. Dazu wurde vom Automobilteilehersteller Continental bereits ein aktives FFB Pedal entwickelt, welches auch im Fahrsimulator des FTG zum Einsatz kommen könnte.

Bei der Detektion der Wahlhebelstellung werden derzeit die Spannungspegel der Positionsanzeige-LED's im Fahrzeuginnenraum gemessen. Hier könnten durch eine Komparator-schaltung und anschließenden Einlesen durch digitale Eingänge, Steckplätze am Datenerfassungssystem eingespart werden.

Auch das FFB am Lenkrad kann, wie auch beim Gaspedal, dazu verwendet der/den FahrerIn ein weiteres haptisches Feedback zu geben. Hierzu wird laut [29] haptisches Feedback am Lenkrad bereits von Spurhalteassistenten benutzt, um die/den FahrerIn auf das Fah-

ren am Fahrbahnstreifen aufmerksam zu machen. Weiters wird schon an Steer-by-Wire Lösungen für Fahrzeuge gearbeitet, bei denen keine mechanische Verbindung zwischen Lenkrad und Reifen besteht. Hier gibt es noch viele Möglichkeiten das haptische Feedback am Lenkrad zu nutzen.

Anhang A

Abkürzungsverzeichnis

ASR	Antriebs-Schlupf-Regelung
AVL	Anstalt für Verbrennungskraftmaschinen List
BLDC	Bürstenloser Gleichstrom (Motor)
CAN	Controller Area Network
CiA	CAN in Automation
COB-ID	Communication Object ID
DLR	Deutsches Zentrum für Luft- und Raumfahrt
ESP	Elektronisches Stabilitätsprogramm
FAS	Fahrerassistenzsystem
ESP	Elektronisches Stabilitätsprogramm
FFB	IForce Feedback
FTG	Institut für Fahrzeugtechnik Graz
LCD	Liquid Crystal Display
LED	Light Emittig Diode
NMT	Netzwerkmanagement
PDO	Prozessdatenobjekt
SBW	Steer-by-Wire

SDO	Servicedatenobjekt
USB	Universal Serial Bus
VDS	Visual Drive Setup
VNT	Vehicle Network Toolbox

Anhang B

Matlab Startup File (Startup.m)

```
1 %% Creating Channel
2 commCh = canChannel('Vector', 'VN1640_1', 1)
3 %configBusSpeed(commCh,500000)
4
5 % Creating Startup Messages
6 operationMode = canMessage(1537,false,8)
7 operationMode.Data = [47 96 96 0 4 0 0 0]
8
9 checkOperationMode = canMessage(1537,false,8)
10 checkOperationMode.Data = [64 97 96 0 0 0 0 0]
11
12 changeTransmissionModeVelocity = canMessage(1537,false,8)
13 changeTransmissionModeVelocity.Data = [47 1 24 2 1 0 0 0]
14
15 changeToOperationalMode = canMessage(0,false,2)
16 changeToOperationalMode.Data = [1 1]
17
18
19 % Start the Channel
20 start(commCh)
21
22 % Proof of Channel properties
23 commCh
24
25 %% Send Operation Mode Message
26 transmit(commCh,operationMode)
```

```
27
28 % Receive Operation Mode Message
29 rxmsg = receive(commCh,2)
30 rxmsg(1)
31
32 % Send Operation Mode Check Message
33 transmit(commCh,checkOperationMode)
34
35 % Receive Operation Mode Proof Message
36 rxmsg = receive(commCh,2)
37 rxmsg(1)
38
39 % Wait for the Controller to store Parameters
40 pause(0.1)
41
42 % Send Transmission Mode Velocity Message
43 transmit(commCh,changeTransmissionModeVelocity)
44
45 % Receive Transmission Mode Velocity Proof Message
46 rxmsg = receive(commCh,2)
47 rxmsg(1)
48
49 % Wait for the Controller to store Parameters
50 pause(0.1)
51
52 % Send Operational Mode Message
53 transmit(commCh,changeToOperationalMode)
54
55 % Receive Operational Mode Message
56 rxmsg = receive(commCh,2);
57 rxmsg(1)
58
59 %% Start Operation
60 operationState = canMessage(1537,false,8);
61
62 % Ready to Switch On
63 operationState.Data = [43 64 96 0 6 0 0 0];
64 transmit(commCh,operationState)
```

```
65
66 % Switch On
67 operationState.Data = [43 64 96 0 7 0 0 0];
68 transmit(commCh,operationState)
69
70 % Operation Enable
71 operationState.Data = [43 64 96 0 15 0 0 0];
72 transmit(commCh,operationState)
73
74 %% Stop Operation
75
76 % Switch On
77 operationState.Data = [43 64 96 0 7 0 0 0];
78 transmit(commCh,operationState)
79
80 % Ready to Switch On
81 operationState.Data = [43 64 96 0 6 0 0 0];
82 transmit(commCh,operationState)
83
84 % Off
85 operationState.Data = [43 64 96 0 0 0 0 0];
86 transmit(commCh,operationState)
87
88
89 %% Stop the Channels
90 stop(commCh)
```

Literaturverzeichnis

- [1] Fraunhofer Austria. <http://www.fraunhofer.at/de/vc/Projekte/muegen.html>, 2014. Zugriff am 25.8.2014.
- [2] Peter A Philipp and Stuttgart Daimler-Benz-Aktiengesellschaft. *Forschungsinstrument Fahrsimulator*. VDI-Verlag, 1990.
- [3] Andrew Liu and Stacey Chang. Force feedback in a stationary driving simulator. In *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on*, volume 2, pages 1711–1716. IEEE, 1995.
- [4] Logitech G27 Racing Wheel. <http://gaming.logitech.com/de-at/product/g27-racing-wheel>, 2014. Zugriff am 26.8.2014.
- [5] Niels Klußmann and Arnim Malik. *Lexikon der Luftfahrt*, volume 2. Springer, 2007.
- [6] G Helmchen and W Predl. Moderne Simulation fuer das Training von Fahrpersonal von Schienen- und Radfahrzeugen. *VDI-Berichte*, (1745), 2003.
- [7] R Dietrich and W Tomaske. Technische und lernpsychologische Zusammenhaenge bei Trainingssimulatoren fuer die Fahrausbildung. *VDI-Berichte*, (1745), 2003.
- [8] Michael Kleczka, Matthias Baumann, Helmut Reichelt, and Martin Schetter. Fahrwerk und Antriebsstrang der neuen C-Klasse. *ATZ-Automobiltechnische Zeitschrift*, 102(7-8):516–523, 2000.
- [9] ECA Faros. <http://www.ecafaros.com/>, 2014. Zugriff am 28.8.2014.
- [10] Deutsches Zentrum für Luft- und Raumfahrt DLR. <http://www.dlr.de>, 2014. Zugriff am 28.8.2014.
- [11] T.L. Brooks. Telerobotic response requirements. In *Systems, Man and Cybernetics, 1990. Conference Proceedings., IEEE International Conference on*, pages 113–120, Nov 1990.

- [12] LinMot. <http://www.linmot.com/products/linear-motors/p01-48x360f/>, 2014. Zugriff am 31.8.2014.
- [13] Infranor XtraforsPrime. <http://www.infranor.com/de/>, 2014. Zugriff am 1.9.2014.
- [14] Institut für Mechanik. www.mechanik.tugraz.at/, 2014. Zugriff am 1.9.2014.
- [15] CAN in Automation. <http://www.can-cia.org/>, 2014. Zugriff am 2.9.2014.
- [16] Vector - CANopen solutions. <http://canopen-solutions.com/>, 2014. Zugriff am 2.9.2014.
- [17] *HIPERFACE®-Beschreibung Description of the HIPERFACE® Interface*. Sick-Stegman, <http://www.sick.com/>, 2014.
- [18] Matlab - Vehicle Network Toolbox. <http://www.mathworks.de/de/help/vnt/index.html>, 2014. Zugriff am 4.9.2014.
- [19] Sajjad Samiee. Development of a Force-Feedback mechanism for a brake pedal. Technical Report, Institut für Fahrzeugtechnik, 2013.
- [20] Daniel Hammer. Echtzeitumgebung für einen Fahr Simulator. Diplomarbeit in Bearbeitung, Institut für Technische Informatik, 2014.
- [21] Markus Stromberger. Entwicklung eines Pedalkraftsimulators für die Anfahrkupplung eines PKWs. Bachelorarbeit, Institut für Fahrzeugtechnik, 2014.
- [22] Wolfgang Schmusch. *Elektronische Meßtechnik: Prinzipien, Verfahren, Schaltungen*. Vogel, 1993.
- [23] Untersweg Wolfgang. Prüfstand zur Messung mechanischer Größen. Diplomarbeit, 1999.
- [24] Roger Bauer, Thomas Raste, and Peter E Rieth. Systemvernetzung von Hybridantrieben. *ATZelektronik*, 2(4):6–10, 2007.
- [25] Christian Lange, Heiner Bubb, Marcus Tönnis, and Gudrun Klinker. Sicherheitspotential und Verbrauchsreduzierung durch ein intelligent geregeltes aktives Gaspedal. *Tagungsband Tagung Fahrerassistenz TU München*, 3:7–8, 2008.
- [26] Continental - Accelerator Force Feedback Pedal. <http://www.conti-online.com/>, 2014. Zugriff am 19.9.2014.
- [27] Aron Kiraly. Konstruktion einer mechanischen Lösung für Force-Feedback eines Getriebes im Zuge der Konstruktion eines Fahrzeugsimulators. Bachelorarbeit, Institut für Fahrzeugtechnik, 2014.

- [28] Patrick Schnabel. Elektronik-fibel. *Elektronik, Bauelemente, Schaltungstechnik, Digitaltechnik*, 4, 2007.
- [29] Anna Schieben, Daniel Damböck, Johann Kelsch, Herbert Rausch, and Frank Flemisch. Haptisches Feedback im Spektrum von Fahrerassistenz und Automation. In *Proceedings of*, volume 3, pages 7–8, 2008.
- [30] Holger Zeltwanger. *CANopen: das standardisierte, eingebettete Netzwerk*. VDE Verlag, 2008.

Abbildungsverzeichnis

1.1	Aufbau Fahrsimulator des FTG	2
2.1	Logitech G27 Racing Wheel, Quelle [4]	6
2.2	Fix Base Simulator der Firma ECA Faros, Quelle [9]	7
2.3	Motion Base Simulator des DLR, Quelle [10]	8
3.1	Aufbau Pedalwegmessung am FTG	10
3.2	Zusammenhang von Pedalweg zu Kraft bei einer Vollbremsung	10
3.3	Aufbau Bremsaktuator mit linear Aktuator	11
3.4	Aufbau Bremsaktuator mit rotatorischen Aktuator	12
3.5	Motor-Controller System des Bremsaktuators mit rotatorischen Aktuator	14
3.6	Schematische Darstellung eines Geräteprofils, Quelle [16]	15
3.7	CANopen NMT-Zustandsmaschine, Quelle CANopen Communication Protocol [13]	16
3.8	Startfenster des Visual Drive Setup [13]	19
3.9	Flussdiagramm einer CAN Netzwerkverbindung mittels VNT, Quelle [18]	23
3.10	Flussdiagramm Initialisierung Controller CD1-k	24
3.11	Zustandsmaschine Stromregler Controller CD1-k, Quelle [13]	28
3.12	Matlab Simulink Model der Controlleransteuerung	31
3.13	Motor-Controller System des Bremsaktuators mit rotatorischen Aktuator und Rückführung der Bremskraft	32
3.14	Messschaltung Wegmessung Drehpotentiometer	34
3.15	Aufbau einer Differentialdrossel im Schnitt	35
3.16	Raster für kodierte Längengeber im Binärkode	35
3.17	Konzept der Kupplungspedalpositionsmessung	36
3.18	Schematische Darstellung des Gaspedalmoduls des Mini Countryman	38
3.19	Schaltplan Gaspedalmodul; Quelle: BMW AG	38
3.20	Schematischer Schaltplan Gaspedalmessung	40
3.21	Bahnformen für Schalthebel von manuellen und automatischen Schaltgetriebe	42

3.22 Exemplarische Eingangsbeschaltung PAD-DI8	44
3.23 Auszug aus dem Schaltplan des Wahlhebelmoduls; Quelle: BMW AG	45
3.24 Schematischer Schaltplan Wahlhebelmodulmessung	46
3.25 Schema Lenkradsystem SBW	48
3.26 Verpolungsschutz mittels Diode in Sperrichtung; vgl. [28]	49
3.27 Relaisschaltung zur Trennung des Systems von der Spannungsquelle	50
3.28 Gesamtschaltung der Spannungsversorgung des Lenkradsystems	50
3.29 Simulink Modell der Positionsabfrage des Lenkrads	51

Tabellenverzeichnis

2.1	Bewegungsdaten des Hexapod Systems, Quelle: [10]	8
3.1	Anforderungen für den Bremsaktuator aus den erhaltenen Messdaten	11
3.2	Nutzwertanalyse zur Bestimmung des Aktuatorkonzepts	13
3.3	Aufbau der 11-Bit COB-ID	17
3.4	CANopen Funktionscodes	18
3.5	Objektdaten Mode of Operation, vgl. CANopen Communication Profile [13]	25
3.6	Datensegment checkOperationMode	25
3.7	Objektdaten Mode of Operation Display, vgl. CANopen Communication Profile [13]	25
3.8	Datensegment checkOperationMode	26
3.9	Objektdaten TPDO2, vgl. CANopen Communication Profile [13]	26
3.10	Datensegment changeTransmissionModeVelocity	26
3.11	Objektdaten NMT-Objekt, vgl. CANopen Communication Profile [13] . . .	27
3.12	Datensegment changeToOperationalMode	27
3.13	Objektdaten Control Word Parameter, vgl. CANopen Communication Pro- file [13]	29
3.14	Zustände, Bitwerte und mögliche Transitionen, vgl. CANopen Communica- tion Profile [13]	29
3.15	Objektdaten Target Torque, vgl. CANopen Communication Profile [13] . .	30
3.16	Pinbelegung 9-Pin Sub-D Stecker für DAQ-AIN; Quelle: Datenblatt DAQ- AIN	40
3.17	Pinbelegung 25-Pin Sub-D Stecker für PAD-DI8; Quelle: Datenblatt PAD-DI8	44
3.18	Aufbau CAN-Nachricht der Lenkradposition	51
3.19	Zusammenhang zwischen analogem Eingangssignal und resultierendem Dreh- moment; Quelle Datenblatt Lenkradsystem SBW	52
3.20	Pinbelegung 9-Pin Sub-D Stecker für DAQ-V-OUT; Quelle: Datenblatt DAQ-V-OUT	53