



U N I V E R S I T Ä T
K O B L E N Z · L A N D A U

Fachbereich 4: Informatik

Haptische Rückfahrsistenz für Fahrzeuge mit 2-achsigem Anhängen

Bachelorarbeit

zur Erlangung des Grades eines Bachelor of Science
im Studiengang Informatik

vorgelegt von

Marco Brack

Erstgutachter: Prof. Dr. Dieter Zöbel
AG für Echtzeitsysteme

Zweitgutachter: Andreas Stahlhofen
AG für Echtzeitsysteme

Koblenz, im Juni 2015

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ja Nein

Mit der Einstellung der Arbeit in die Bibliothek bin ich ein-
verstanden.

Der Veröffentlichung dieser Arbeit im Internet stimme ich
zu.

.....
(Ort, Datum)

.....
(Unterschrift)

Zusammenfassung

Die Arbeitsgruppe Echtzeitsysteme an der Universität Koblenz beschäftigt sich seit mehreren Jahren mit der Thematik autonomes und assistiertes Fahren. Eine große Herausforderung stellen in diesem Zusammenhang mehrgliedrige Fahrzeuge dar, deren Steuerung für den Fahrer während der Rückwärtsfahrt sehr anspruchsvoll ist. Um präzise Manöver zu ermöglichen, können elektronische Fahrerassistenzsysteme zum Einsatz kommen. Im Rahmen vorhergehender Arbeiten sind bereits einige Prototypen entstanden, von denen jedoch keiner eine geeignete Lösung für moderne, zweiachsige Anhänger darstellt. Im Rahmen dieser Arbeit wurde ein prototypisches Fahrerassistenzsystem entwickelt, wobei es noch weiterer Forschungs- und Entwicklungsarbeit bedarf, um das System straßentauglich zu machen.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	3
1.2	Ziel der Arbeit	5
1.3	Aufbau der Arbeit	6
2	Grundlagen	8
2.1	Technische Grundlagen	8
2.1.1	Lenkanlagen	9
2.1.2	Gliederfahrzeuge	12
2.1.3	Überblick über Fahrerassistenzsysteme	14
2.2	EZAuto	14
2.2.1	Leitstand	15
2.2.2	Verwaltung	16
2.2.3	EreignisverteilerServer	17
2.2.4	Hardwarestub	17
2.2.5	Kinematikaufbereitung	18
2.2.6	Laser	18
2.2.7	Wichtige Details von EZAuto	18
2.3	Kinematische Grundlagen	20
2.3.1	Das Einspurmodell	20
2.3.2	Definition von Bezeichnern	22
2.3.3	Stabile Fahrt	25
3	Entwicklung des Assistenzsystems	30
3.1	Lenklogik	30
3.1.1	Einknickwinkel ändern	32

3.1.2	Einknickwinkel halten	37
3.2	Softwarearchitektur	38
3.2.1	Integration in bestehende Software	39
3.2.2	Aufbau der Zweiachs-Lenklogik	40
3.3	Anwendungsmodul im Leitstand	41
4	Schlussbemerkungen	43
4.1	Zusammenfassung	43
4.2	Bewertung	45
4.3	Ausblick	47
5	Anhang	49
5.1	Codefragmente	49
5.1.1	Übergangsbedingungen	49
5.1.2	Radwinkel setzen	52

Abbildungsverzeichnis

2.1	Aufbau einer Lenkanlage nach [Rat70] und [Rat92] aus [RB00] . . .	10
2.2	Schematischer Aufbau einer elektronischen Lenkanlage aus [Wey06]	12
2.3	GUI des Leitstands während der Ausführung der Zweiachs-Assistenz	16
2.4	Einspurmodell mit 2 Gliedern. Quelle: [Zöb01]	22
2.5	Stabile Fahrt eines General-3-Trailers	23
3.1	Verlauf der Einknickwinkel während eines Manövers	36
3.2	Verlauf der Einknickwinkel mit superpassing	37
3.3	Vereinfachtes Class diagram der neuen Architektur	39

Tabellenverzeichnis

3.1	Radeinschläge in Schritten	34
-----	--------------------------------------	----

Kapitel 1

Einleitung

In den vergangenen Jahren wurden immer mehr Fahrzeuge mit Fahrerassistenzsystemen ausgestattet. Die einfachsten dieser Systeme warnen den Fahrer bei der Rückwärtsfahrt vor sonst nur schwer bemerkbaren Hindernissen durch akustische Signale. Komplexere Systeme helfen, während der Fahrt die Spur zu halten, entweder ebenfalls passiv durch Warnsignale, oder aktiv durch Lenkunterstützung (siehe hierzu auch [Wik15d]); oder sie verringern Unfallgefahr und Schwere von Auffahrunfällen durch eine automatische Notbremsung bei einer drohenden Kollision (siehe hierzu auch [Wik15c]). Dies ist nur eine Auswahl bekannterer Systeme.

Die Arbeitsgruppe Echtzeitsysteme der Universität Koblenz beschäftigt sich daher unter anderem mit diversen Fahrerassistenzsystemen. Dabei liegt ein Fokus auf der Unterstützung des Fahrers bei rückwärts gerichteten Manövern. Neben der reinen Logik zum Lenken eines Fahrzeugs wurden auch diverse Projekte zur Entwicklung von so genannten Mensch-Maschine Schnittstellen durchgeführt.

Eine *Mensch-Maschine-Schnittstelle* (MMS) ist, abstrakt gesehen, ein klar definiertes System zur Interaktion zwischen Bediener und Maschine. Im Fall eines Fahrerassistenzsystems interagiert das Fahrzeug mit dem Fahrer. Diese Interaktion findet mit Hilfe einer Vielzahl von Komponenten statt. Das Fahrzeug, beziehungsweise das auf ihm laufende Assistenzsystem, teilt dem Fahrer über verschiedene *Kanäle* Informationen über zum Beispiel den Zustand des Fahrzeugs mit. Die üblichsten Kanäle sind der *akustische Kanal* (zum Beispiel Warntöne beim Rückwärtsfahren), der *optische Kanal* (zum Beispiel das Tachometer) und der *hap-*

tische Kanal (zum Beispiel Übertragung vom Unebenheiten in der Straße auf das Lenkrad). Beim Design einer MMS ist darauf zu achten, dass ein Mensch nur eine begrenzte Aufnahmefähigkeit hat, und manche Informationen nicht über jeden Kanal gleich gut vermittelt werden können. Für weitere Beispiele zu Mensch-Maschine-Schnittstellen siehe [Wik15b].

Ein zentrales Projekt der Arbeitsgruppe war die Entwicklung eines Frameworks zur Steuerung von Fahrzeugen. Dieses Projekt, EZAuto, beinhaltet heute zusätzlich Funktionen zur Simulation von Fahrzeugen, zum Anzeigen diverser optischer Informationen und Funktionen zum Steuern von Fahrzeugen durch realitätsnahe Eingabegeräte, die sogar eine Kraftrückkopplung von der Software auf das Lenkrad beinhaltet. Des Weiteren wurde im Rahmen des Projekts ein kinematisches Modell zur Beschreibung der Bewegung eines mehrgliedrigen Fahrzeugs entwickelt, mit dessen Hilfe auch schon eine Lenkassistentz für einachsige Anhänger und auch eine alternative Lenkassistentz für zweiachsige Anhänger umgesetzt wurden.

Die erwähnte Assistentz für einachsige Anhänger operiert ausschließlich auf elektronischen Lenkanlagen, das heißt, die Assistentz ist in der Lage, die Beziehung zwischen dem vom Fahrer am Lenkrad gelenkten Winkel und dem tatsächlich von den Rädern eingeschlagenen Winkel frei neu zu definieren (für eine ausführlichere Beschreibung siehe 2.1.1.2). Diese neue *Semantik* besagt, dass der Einknickwinkel des Anhängers während der Rückwärtsfahrt der gleiche sein soll, der sich auch einstellen würde, wenn der Fahrer mit dem gleichen Lenkradwinkel vorwärts fahren würde. Zu dieser Assistentz wurde [BZ05] publiziert.

Die Arbeitsgruppe Echtzeitsysteme hält ein Patent für eine Komponente des Fahrerassistenzsystems. Die Komponente ist eine Mensch-Maschine-Schnittstelle basierend auf einer elektronischen Lenkanlage, bei der der Fahrer durch Kraftrückkopplung auf das Lenkrad über den Zustand des Fahrzeugs erfährt (Siehe [DZ05] und [B⁺08]).

Da also bereits eine Assistentz für Fahrzeuge mit einachsigen Anhänger existiert inklusive einer Mensch-Maschine-Schnittstelle, war es Ziel dieser Bachelorarbeit, aufbauend auf EZAuto die Problematik einer Rückfahrassistentz für Fahrzeuge mit zwei Anhängergliedern¹ zu behandeln und einen Prototypen für ein Rückfahr-Lenkassistentzsystem mit der gleichen Semantik für den Lenkradwinkel wie in [BZ05] auf Basis einer elektronischen Lenkanlage zu entwickeln. Die größ-

¹Zum Beispiel Drehschemel-Anhänger

te Herausforderung ist dabei das komplizierte Zusammenspiel der Fahrzeugglieder während der Rückwärtsfahrt. Außerdem sollte die durch die Arbeitsgruppe patentierte Mensch-Maschine-Schnittstelle übertragen werden. Der im Rahmen dieser Arbeit entwickelte Prototyp ist nicht vollständig, bildet aber eine wichtige Grundlage, auf der weitere Arbeiten aufbauen können.

1.1 Motivation

Nach [Wey06] ist das Durchführen von rückwärts gerichteten Fahrmanövern für die meisten Fahrer von Kraftfahrzeugen selbst ohne Anhänger im Gegensatz zum Vorwärtsfahren schwieriger. Eine mögliche Ursache dafür ist die Tatsache, dass die Hinterräder gewöhnlich un gelenkt sind, und das Fahrzeug daher nicht wie gewohnt exakt der durch die Lenkung vorgegebenen Bahn folgt. Zusätzlich werden entsprechende Manöver erschwert, wenn das Fahrzeug einen Anhänger beim Rückwärtsfahren vor sich her schiebt. Dessen Verhalten ist für viele Fahrer unintuitiv und in manchen Situationen schwer vorherzusagen, geschweige denn zu kontrollieren. Ein zweiachsiger Anhänger oder zwei Anhänger erhöhen diese Schwierigkeit noch ein mal um eine weitere Stufe. Verhält sich das Gespann nicht so, wie der Fahrer beabsichtigt hat, so kann es zu kostspieligen Schäden an Zugfahrzeug, Anhängern und Umgebung kommen. In den meisten Fällen leiden Kupplungen, Unterfahrschutz oder die Deichsel des Anhängers. Fahrzeugführer benötigen für rückwärts gerichtete Manöver mit solchen Gespannen langjährige Erfahrung und müssen in schwierigen Situationen ein hohes Maß an Konzentration aufbringen.

Im Rahmen der Arbeit [BZ06] wurde eine Umfrage unter Berufskraftfahrern durchgeführt. Ziel war es die Ursachen der auftretenden Schwierigkeiten zu deduzieren. Die Umfrage ergab folgende Hauptursachen:

- Ungewohnte Fahrtrichtung
- Unintuitive Lenkmanöver
- Eingeschränkte Sicht

Die zentrale Ursache für die Unsicherheit beim Rückwärtsfahren ist der Umstand, dass rückwärts gerichtete Fahrmanöver nur einen kleinen Bruchteil al-

ler Manöver ausmachen. Daher versuchen sogar manche Fahrer diese eher unvertrauten Manöver nach Möglichkeit zu vermeiden, was die Situation nur verschlimmert.

Des Weiteren verhält sich das Gespann nach dem Gefühl der meisten Fahrer bei derartigen Manövern unintuitiv. Gewöhnlich wird erwartet, dass sich Manöver von einfachen Fahrzeugen ohne Anhänger oder mit einachsigen Anhänger übertragen lassen, was nicht der Fall ist. Beispielsweise schlägt beim Rückwärtsfahren mit einem einachsigen Anhänger der Fahrer das Lenkrad nach links ein, um ein Einknicken des Gespanns auf der rechten Seite zu bewirken. Dann wird er einen passenden Lenkwinkel einschlagen, mit dem dieser Einknickwinkel stabil bleibt. Der vordere Teil eines zweiachsigen Anhängers würde sich bei diesem Vorgehen gleich verhalten, aber der hintere Anhängerteil wird (in den meisten Fällen) in Relation zum vorderen Anhängerteil nach links einknicken und das Gespann wird verkanten. Um die gewünschte Fahrt auf einer rückwärts gerichteten, stabilen Kreisbahn herbeizuführen und beizubehalten wird ein mehrschrittiges Verfahren benötigt, das phasenweise große Lenkwinkel benötigt.

Neben der genannten Komplexität des benötigten Manövers sorgt auch die eingeschränkte Sicht für zusätzliche Erschwernis, da phasenweise Teile der Anhänger von anderen Teilen verdeckt sind. Zudem befindet sich das Ende des Gespanns weit weg vom Fahrer und ist unter Umständen schwer erkenntlich.

Laut [Wey06] können diese Probleme durch den Einsatz von Assistenzsystemen in vielerlei Hinsicht gelöst oder abgeschwächt werden. Die meisten modernen Assistenzsysteme, wie sie bereits in vielen Autos zu finden sind, bieten dabei noch vergleichsweise rudimentäre Unterstützung an: Die Angabe von Entfernungen durch akustische Signale. Da selbst bei Personenwagen die Sicht nach hinten eingeschränkt ist, und die beim Rückwärtsfahren verbleibende Distanz vor allem zu niedrigen Hindernissen schwer einzuschätzen ist, werden diese durch einen Sensor gemessen und dem Fahrer mitgeteilt. Zusätzlich gibt es Systeme, die diese Informationen graphisch darstellen, teilweise schematisch, teilweise mit einer realen Kamera am Heck des Wagens.

Fortgeschrittenere Fahrassistenz-Systeme nehmen dem Fahrer das Lenken in gewissen Situationen ab (zum Beispiel Einparkhilfen) oder korrigieren offensichtliche Fehler (zum Beispiel Spurhaltesysteme). Auch sind Systeme im Einsatz, die automatisch bremsen und so Auffahrunfälle vermeiden.

Es ist ersichtlich, dass die Funktion von Fahrassistenz-Systemen nicht bloß

auf Komfort beschränkt ist; durch sie können große Schäden vermieden werden.

Die primäre Einschränkung beim Einsatz dieser Systeme ist laut [Wey06] die dem Auto zugrunde liegende Technik. Gebräuchliche Lenksysteme sind meist nicht elektronisch, das heißt die Räder können durch die Lenkassistentz nicht unabhängig vom Lenkrad bewegt werden. Daher bewegt sich das Lenkrad bei automatischen Einparkmanövern mit. Auf dieser Grundlage sind bereits viele Systeme denkbar, wie die in [Wey06] entwickelte Lenkassistentz für zweiachsige Anhänger, doch mit unabhängigen Rädern (elektronische Lenkanlagen) eröffnen sich neue Möglichkeiten.

Diese Arbeit setzt eine ähnliche Assistentz für dieselbe Art von Gespannen um, beschränkt sich jedoch auf elektronische Lenkanlagen und verfolgt einen anderen Ansatz. Sie reduziert die Komplexität der vom Fahrer geforderten Eingaben noch weiter und automatisiert jeden Aspekt des Manövers.

1.2 Ziel der Arbeit

Das primäre Ziel dieser Bachelorarbeit ist es, die von der AG Zöbel im Rahmen des Softwareprojektes EZAuto entwickelte Rückfahr-Lenkassistentz für einachsige Anhänger auf Anhängerkonfigurationen mit zwei Drehachsen zu übertragen. Dabei soll prinzipiell nicht unterschieden werden zwischen Drehschemelanhängern oder zwei aneinanderhängenden Anhängern. Auch soll der Ansatz unabhängig sein von den Maßen der Anhänger(-teile), das heißt die Assistentz muss für relevante Größen parametrisierbar sein.

Die Lenkassistentz ist dabei, im Gegensatz zu [Wey06], vollkommen abhängig von einer elektronischen Lenkanlage und kann demnach nicht auf elektromechanische oder pur mechanische Lenkanlagen übertragen werden.

Die Arbeit [Wey06] führte eine komplette Überarbeitung des gesamten Grundgerüsts für Lenkassistentzen in EZAuto ein. In ihr wird von Teilen der Assistentzlogik abstrahiert und so vor allem der Implementationsaufwand für neue Assistentzen verringert. Außerdem wurden Generalisierungen für verschiedene Mensch-Maschine-Schnittstellen konzipiert.

Unglücklicherweise wurden diese Änderungen zu der Zeit nicht in die offizielle Version von EZAuto integriert und daher sind die beiden Versionen seitdem divergiert. Aus diesem Grund war es nicht möglich auf den Ergebnissen der benannten Arbeit aufzubauen, und als weiteres Ziel dieser Arbeit mussten

so erneut diese Generalisierungen konzeptioniert und umgesetzt werden. Da der Fokus dieser Bachelorarbeit allerdings nicht auf derartigen softwaretechnischen Herausforderungen lag, wurde davon abgesehen, diese in dem Umfang anzugehen, wie es in [Wey06] getan wurde.

Neben der Lenklogik an sich ist die Vermittlung von Zustandsinformationen von Assistenz zu Fahrer und von Instruktionen von Fahrer zu Assistenz ein wichtiger Bestandteil des entwickelten Systems. Eine solche Mensch-Maschine-Schnittstelle (im Folgenden auch MMS) kann prinzipiell über verschiedene Kanäle umgesetzt werden, bereits angesprochen waren der akustische und der visuelle Kanal. Durch die elektronische Lenkanlage steht noch ein weiterer Kanal zur Verfügung: Der haptische Kanal. Konkret bedeutet das, dass über das Lenkrad dem Fahrer durch Anlegen von Kräften, die dem Lenken des Fahrers entgegenwirken, diesem mitgeteilt wird, dass er im Begriff ist, den (stabilen) Zustand des Fahrzeugs zu ändern, oder dass er sich in einem stabilen Zustand befindet. Um mit der Assistenz zu interagieren muss der Fahrer der anliegenden Kraft entgegenwirken. Eine derartige MMS wurde bereits für einachsige Anhänger umgesetzt und als sinnvoll befunden, sodass sich die in dieser Arbeit entwickelte MMS ebenfalls auf den haptischen Kanal beschränken sollte. Diese Anforderung konnte, wie später zu sehen sein wird, nicht derart umgesetzt werden.

1.3 Aufbau der Arbeit

Die Arbeit ist im Wesentlichen in zwei Kapitel unterteilt: Nach dieser Einleitung werden im zweiten Kapitel die Grundlagen, auf denen die Arbeit aufgebaut hat, geschildert. Dabei wird auf die technischen Grundlagen von Fahrzeugen eingegangen, was die verschiedenen möglichen Lenkanlagen, eine generelle Übersicht über so genannte Gliederfahrzeuge und einen Überblick über Lenkassistenzsysteme beinhaltet. Außerdem werden die softwaretechnischen Grundlagen behandelt, also die für die Arbeit relevanten Komponenten des EZAuto Softwarepakets. Schließlich wird das kinematische Modell für Gliederfahrzeuge erläutert, inklusive der Definition später genutzter Bezeichner, und einer Beschreibung der stabilen Fahrt, welche ein besonderer und für die entwickelte Lenkassistenz zentraler Zustand ist.

Im dritten Kapitel wird dann der entwickelte Prototyp im Detail beschrieben. Dabei werden vor allem auf die Erkenntnisse zur Lenklogik eingegangen und

dokumentiert, wie sie umgesetzt wurden. Des Weiteren wird eine neue Softwarearchitektur vorgestellt, mit deren Hilfe das neue System umgesetzt wurde, und mit der es in Zukunft einfacher sein soll, weitere Assistenzsysteme zu entwickeln. Außerdem soll es, zusammen mit der Dokumentation im Code, zukünftigen Entwicklern möglich gemacht werden, den Prototypen weiter zu entwickeln. Zum Schluss des Kapitels wird noch die Integration in die Benutzeroberfläche von EZ-Auto thematisiert.

Zum Abschluss der Arbeit wird eine kurze Zusammenfassung gegeben und danach die Ergebnisse der Arbeit bewertet. Es werden in diesem Zuge vor allem Erfolge und Schwachstellen aufgezeigt. Dann wird noch ein kurzer Ausblick auf weiterführende Arbeiten gegeben, um die Weiterentwicklung auf den richtigen Weg zu bringen.

Kapitel 2

Grundlagen

In diesem Kapitel sollen alle wichtigen Grundlagen und Gegebenheiten für die vorliegende Arbeit vorgestellt und erläutert werden. Zunächst wird auf die technischen Grundlagen eingegangen, also die wichtigsten Vorgaben, die durch Fahrzeuge für die Assistenz bestehen. Dann wird die bestehende Software vorgestellt, in die das zu entwickelnde Lenkassistenzsystem eingebettet werden soll. Zum Schluss wird das mathematische, beziehungsweise kinematische Modell eines mehrgliedrigen Fahrzeugs, mit dem die Assistenz arbeiten soll, im Detail gezeigt.

2.1 Technische Grundlagen

Assistenzsysteme stellen eine gewisse technische Herausforderung für die Entwickler dar. Viele Fahrzeuge haben fundamental unterschiedliche Gegebenheiten und schränken so die Möglichkeiten mehr oder weniger stark ein. In dieser Sektion sollen die wichtigsten Aspekte vorgestellt, um einen Überblick über die Möglichkeiten und Einschränkungen verschiedener Fahrzeugtypen zu geben.

Der hier bedeutsamste technische Aspekt ist die Lenkanlage. Es werden die drei gebräuchlichen Lenkanlagen vorgestellt und deren Vor- und Nachteile, sowie deren Verbreitung, diskutiert. Außerdem werden auf einer allgemeinen Ebene Gliederfahrzeuge, also Fahrzeuge mit mehreren aneinander gekoppelten Teilen, thematisiert, und schlussendlich der entwickelte Prototyp eingeordnet.

Da sich seit 2006 an den hier vorgestellten Technologien nichts Grundlegendes geändert hat, ist die gesamte folgende Untersektion weitestgehend eine Zusammenfassung der Beschreibungen in [Wey06], Sektion 2.1.

2.1.1 Lenkanlagen

Der für ein Lenkassistenzsystem wichtigste Aspekt eines Fahrzeugs ist die Lenkanlage. Nach Richtlinie des Rates der Europäischen Union ([Rat70] und [Rat92]) lässt sich die Aufbau von Lenkanlagen in drei Hauptbestandteile gliedern:

- *Betätigungseinrichtung*
Bestandteil, über den der Fahrer üblicher Weise die Fahrtrichtung vorgibt. Genauer gesagt gibt er durch den Lenkradwinkel den Radlenkwinkel¹ vor. Die meisten Betätigungseinrichtungen erfordern eine gewisse Kraftereinwirkung durch den Fahrer, um Änderungen hervorzurufen. Die Eingabemethode ist allerdings nicht beschränkt auf das (gewöhnlicher Weise genutzte) Lenkrad, sondern kann prinzipiell auch durch zum Beispiel einen Joystick erfolgen.
- *Übertragungseinrichtung*
Bestandteil, der im Normalfall die Übertragung der Lenkvorgaben des Fahrers auf die Räder realisiert. Im Allgemeinen erfolgt diese Übertragung mechanisch.
- *Gelenkte Räder*
Die Räder des Fahrzeugs, die durch die Lenkanlage gesteuert werden. Der eingeschlagene Winkel soll im Folgenden Radlenkwinkel genannt werden.

Lenkanlagen in modernen Fahrzeugen lassen sich basierend auf der Übertragungseinrichtung in drei Klassen einteilen:

- Mechanische Lenkanlage
- Elektromechanische Lenkanlage
- Elektrische Lenkanlage

Die drei Kategorien sollen im Folgenden näher betrachtet werden.

¹Die beiden Begriffe *Radlenkwinkel* und *Lenkradwinkel* sind nicht mit einander zu verwechseln

tragen.

Nahezu alle modernen mechanischen Lenkanlagen unterstützen den Fahrer durch zusätzliche Komponenten beim Lenken. Solche *Hilfskraftanlagen* (auch: Servolenkung, im Gegensatz zur *manuellen Lenkanlagen*) werden weiterhin unterschieden, je nach Art der Kraftunterstützung, in *hydraulisch*, *pneumatisch* und *elektrisch*.

2.1.1.2 Elektromechanische Lenkanlage

Obwohl diese Klasse der Lenkanlagen nur in wenigen wissenschaftlichen Veröffentlichungen erwähnt wird (zum Beispiel in [BSWZ04]), ist ihre Erwähnung in dieser Arbeit unerlässlich, da der in [Wey06] entwickelte Prototyp eines Lenksistenzsystems für zweiachsige Anhänger, im Gegensatz zu dem in dieser Arbeit entwickelten Prototypen, auch auf elektromechanischen Lenkanlagen operieren kann. Aufgrund der Seltenheit dieser Lenkanlage wurde diese Anforderung nicht an den neuen Prototypen gestellt, was das gesamte System stark vereinfacht und neue Möglichkeiten eröffnet hat.

Elektromechanische Lenkanlagen haben wie rein mechanische Lenkanlagen eine direkte Kopplung zwischen Betätigungseinrichtung und gelenkten Rädern. Sie zeichnen sich jedoch dadurch aus, dass zusätzlich zu der vom Fahrer aufgewendeten Lenkkraft und den externen Kräften, die durch die Räder auf die Lenkanlage wirken, mit Hilfe eines Motors gezielt weitere Kräfte auf die Übertragungseinrichtung gewirkt werden können. Dies bewirkt auf der einen Seite eine Krafrückkopplung an der Betätigungseinrichtung, die der Fahrer wahrnimmt, und auf der anderen Seite eine aktive Änderung am Radlenkwinkel.

Es ist wichtig, elektromechanische Lenkanlagen klar von Hilfskraftanlagen zu unterscheiden. Eine Hilfskraftanlage unterstützt lediglich die Kraftaufwendung des Fahrers, während eine elektromechanische Lenkanlage eigenständig Änderungen an der Übertragungseinrichtung bewirken kann.

Moderne PKW mit Parkassistenten besitzen bereits eine elektromechanische Lenkanlage. Beim Einparken übernimmt ein Motor die Steuerung der Lenksäule und bewegt das Fahrzeug automatisch in die mit Hilfe von Sensoren erfasste Parklücke. Ein solches Fahrerassistenzsystem wäre mit einer mechanischen Lenkanlage nicht möglich, da bei ihr keine Möglichkeit zum Steuern der Lenksäule vorhanden ist.

2.1.1.3 Elektrische Lenkanlage

Anders als bei den beiden bisher vorgestellten Lenkanlagen, gibt es bei einer elektrischen Lenkanlage keine direkte Kopplung mehr zwischen Betätigungseinrichtung und gelenkten Rädern. Die Übertragungseinrichtung ist vollkommen elektronisch, weswegen diese Art zu Lenken auch *Steer-by-Wire* genannt wird. Abbildung 2.2 zeigt einen schematischen Aufbau einer elektronischen Lenkanlage.

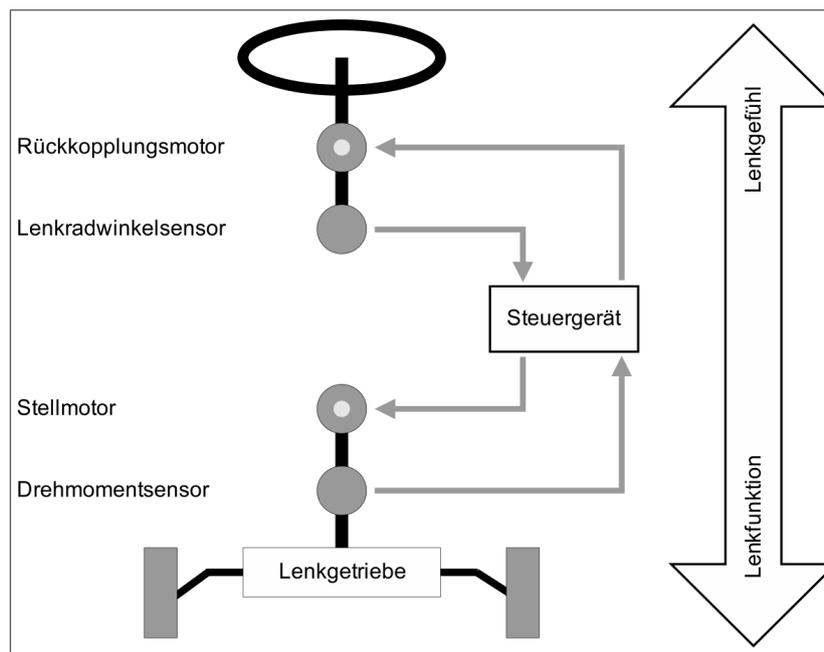


Abbildung 2.2: Schematischer Aufbau einer elektronischen Lenkanlage aus [Wey06]

Da keine mechanische Kopplung besteht, kann die Beziehung zwischen Lenkradwinkel und Radlenkwinkel neu definiert werden. Wie in [BZ05] soll für den hier entwickelten Prototypen während der Rückfahrt gelten, dass der Lenkradwinkel proportional ist zum Einknickwinkel des Anhängers³.

2.1.2 Gliederfahrzeuge

Gliederfahrzeuge (im Englischen auch *articulated vehicle*) sind im allgemeinen Fahrzeuge, die eine oder mehrere Einknickstellen haben. Das erleichtert vor al-

³Im Fall eines zweiteiligen Anhängers der Hintere Einknickwinkel

lem langen Fahrzeugen schärfere Kurven zu fahren. Die Einknickstellen können dabei permanent sein, also fest in die Konstruktion des Fahrzeugs integriert, oder semi-permanent, also in der Regel verbunden, aber prinzipiell abkoppelbar. Im Rahmen dieser Arbeit wurden nur mehrgliedrige Straßenfahrzeuge betrachtet, der Begriff kann jedoch auch für andere Fahrzeuge wie Züge gebraucht werden.

Für das Verhalten des Fahrzeugs macht es, wie bereits in der Einleitung erwähnt, einen sehr großen Unterschied, wie viele Einknickpunkte beziehungsweise Glieder das Fahrzeug (das Gespann) hat. In der Regel haben sowohl Lastkraftwagen sowie Personenkraftwagen nur einen an der Kupplung hängenden Anhänger. In den letzten Jahren kommen allerdings vermehrt Gespanne mit zwei anhängenden Gliedern zum Einsatz, unter anderem einige Typen des *EuroCombi*. [Wik15a] hat eine ausführliche Darstellung möglicher Variationen des EuroCombi. Setzt sich dieser Trend fort, wird der Bedarf für eine Lenkassistenten steigen, um auch potenziell weniger gut ausgebildete Fahrern das fahren mit einem solchen Gespann zu ermöglichen.

Die Relevanz einer Fahrerassistenz für mehrgliedrige Fahrzeuge, wenn man die gesetzliche Lage zu Schwertransportern betrachtet. Laut [Lar09] musste die Europäische Union die Höhen- und Breitenlimitierungen für Schwertransporter erhöhen, als Schweden und Finnland 1995 der europäischen Union beigetreten sind. Dort waren schon seit längerer Zeit größere Gespanne erlaubt, als im Rest der Union zulässig, sodass eine Lösung gefunden werden musste, damit ausländische Lastwagen in den beiden Ländern auch eine wirtschaftliche Chance haben würden. Daraufhin wurde in 1996 die EU Richtlinie 96/53/EG festgelegt (für den genauen Gesetzestext siehe [Rat09]). Sie erlaubt die Nutzung von Gespannen mit größeren Abmessungen auf festgelegten Strecken unter der Bedingung, dass bereits existierende, genormte Module genutzt werden.

Durch die Kombination der verschiedenen Module sind die oben genannten Varianten des EuroCombi entstanden, was somit nach Vermutung der Arbeitsgruppe Echtzeitsysteme einen Markt für Fahrerassistenzsysteme für diese Fahrzeuge eröffnen wird. Der Markt wird allerdings durch die Abhängigkeit von einer elektronischen Lenkanlage eingeschränkt. Diese kommen derzeit so gut wie nicht zum Einsatz.

2.1.3 Überblick über Fahrerassistenzsysteme

Unter Fahrerassistenzsystemen (kurz: Fahrassistenz) wird im Rahmen dieser Arbeit im allgemeinen ein elektronisches System zur Unterstützung des Fahrers beim Bedienen eines Fahrzeugs verstanden. Sie sind klar abzugrenzen von Systemen, die das Fahrzeug autonom steuern. Ein Lenkassistenzsystem (kurz: Lenkassistenz, oder nur Assistenz) unterstützt den Fahrer lediglich beim Lenken, sowohl aktiv durch Einflussnahme auf die Lenkung, als auch passiv durch Informieren des Fahrers.

Aufbauend auf den technischen Gegebenheiten eines Fahrzeugs gibt es verschiedene Möglichkeiten, Fahrerassistenzsysteme zu integrieren. Eine genauere Betrachtung verschiedener Klassen von Fahrerassistenzsystemen und Lenkassistenzsystemen findet in [Wey06] 2.2 statt und soll an dieser Stelle nicht wiederholt werden. Dort findet sich ebenfalls eine genaue Beschreibung des Aufbaus und der Funktionalität von Lenkanlagen, sowie eine Beschreibung der Möglichkeiten zur Realisierung von Lenkassistenzsystemen. Dort wird auch noch mal auf den Unterschied zwischen einer elektronischen und einer elektromechanischen Lenkanlage eingegangen. Der wichtigste Unterschied des hier entwickelten Prototypen und dem aus [Wey06] ist die vorausgesetzte Lenkanlage. Dieser Prototyp beschränkt sich auf elektronische Lenkanlagen, was die Menge der Möglichkeiten für die Assistenz erhöht.

Der Teil der Lenkassistenz, der die neue Semantik zwischen Betätigungseinrichtung und gelenkten Rädern bei einer nicht-mechanischen Lenkanlage neu festlegt, soll im Folgenden als *Lenklogik* bezeichnet werden.

2.2 EZAuto

Zu Beginn der Arbeit stand das über die Jahre sehr umfangreich gewordene Softwarepaket EZAuto (Siehe zu diesem Abschnitt [Woj05] und [Wey06]). Dessen Softwarearchitektur soll es ermöglichen, Anwendungen zu assistiertem oder autonomem Fahren unter flexiblen Fahrzeug- und Umgebungsbedingungen zu entwickeln. Dabei sind die jeweiligen Anwendungen soweit von der spezifischen Logik zur Interaktion mit dem Fahrzeug getrennt, dass es keinen Unterschied macht, ob es sich um ein Modellfahrzeug, eine Simulation oder ein reales Fahrzeug handelt. Die Architektur besteht aus vielen, mehr oder weniger lose gekop-

pelten Komponenten, die flexibel genutzt werden können, um neue Anwendungen zu entwickeln. Da es wie bei jeder größeren Software verschiedene Sichten auf das System gibt, soll an dieser Stelle nur aus Sicht des Entwicklers einer Fahrerassistenz eine kurze Beschreibung der relevanten Komponenten erfolgen. Für eine thematisch gegliederte, schichtweise Sicht siehe [Wey06].

Zur Steuerung eines Fahrzeugs werden 6 Komponenten benötigt, die als parallel laufende Prozesse mit einander interagieren:

1. Leitstand
2. Verwaltung
3. EreignisverteilerServer
4. Hardwarestub
5. Kinematikaufbereitung
6. Laser

Bei der Ausführung auf einem physischen Fahrzeug werden, bis auf den Leitstand, alle Prozesse auf einem eingebetteten Rechner auf dem Fahrzeug ausgeführt, und der Leitstand auf einem externen Rechner. In einer Simulation werden alle Komponenten auf einem Computer ausgeführt. Im Folgenden sollen die Funktionen der einzelnen Komponenten, wie sie in [Wey06] beschrieben werden, zusammengefasst werden. Dabei liegt ein Fokus auf den ersten beiden der genannten Punkten, da die anderen Prozesse für die hier entwickelte Fahrerassistenz eher irrelevant sind. Sie sollen dennoch nicht unerwähnt bleiben, um dem Leser ein vollständiges Bild der an der Assistenz beteiligten Komponenten zu geben.

2.2.1 Leitstand

Der *Leitstand* stellt dem Benutzer ein Bedienfenster (siehe Abbildung 2.3 zur Interaktion mit der Steuerungssoftware zur Verfügung. Der Leitstand muss zur Ausführung des Programms mit Hilfe eines Dropdown-Menüs mit einem Verwaltungsprozess, beziehungsweise mit einem Fahrzeug, verbunden werden.

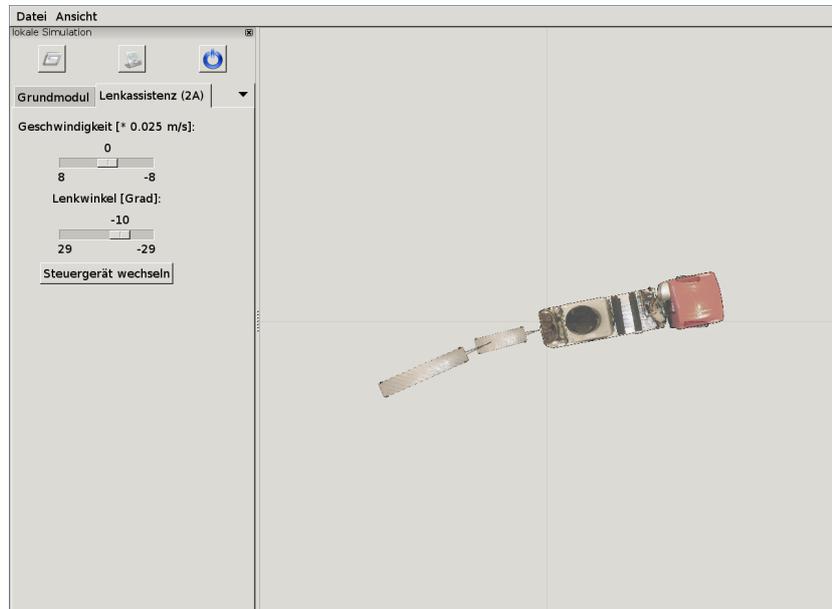


Abbildung 2.3: GUI des Leitstands während der Ausführung der Zweiachs-Assistentz

Da der Leitstand für viele unterschiedliche Anwendungen genutzt werden soll, wurde für ihn eine stark modulare Architektur gewählt: Es wird unterschieden zwischen Anwendungsmodulen und Funktionsmodulen. Anwendungsmodule stellen, wie der Name schon sagt, ganze Anwendungsfälle dar. Es kann gleichzeitig immer nur ein Anwendungsmodul aktiv sein. Zum Beispiel gibt es ein Anwendungsmodul für die einachsige Lenkassistentz. Funktionsmodule hingegen stellen nur kleinere, ergänzende Funktionen zur Verfügung, wie dem Anzeigen von Fahrzeugdaten. Von ihnen können mehrere aktiv sein.

Im Rahmen von Lenkassistentzsystemen übernimmt der Leitstand einige oder alle Teile der Mensch-Maschine Schnittstelle. Er verfügt über Schieberegler zum Bedienen des Fahrzeugs und stellt das Fahrzeug grafisch dar.

2.2.2 Verwaltung

Die *Verwaltung* koordiniert und überwacht die Prozesse auf dem Fahrzeug. Sie beinhaltet einen Zustandsautomaten, der ihren aktuellen Zustand überwacht. Mögliche Zustände sind unter anderem *direkt* zur direkten Steuerung des Fahrzeugs, *Assistentz* zum Fahren mit einer Lenkassistentz und *Fehler*. Der Zustand *Fehler* wird

eingenommen, wenn eine der überwachten Komponenten nicht reagiert oder einen Fehler meldet.

Eine weitere Aufgabe der Verwaltung ist das Instantiieren und Konfigurieren einiger Komponenten zu Beginn der Ausführung. Außerdem *pingt* die Verwaltung diese Komponenten periodisch an, um sicher zu stellen, dass sie noch funktionsfähig sind. So wird verhindert, dass Komponenten unbemerkt inaktiv werden und Fehler verursachen.

Im Rahmen dieser Arbeit ist die Lenkassistentz-Komponente der Verwaltung von großer Wichtigkeit. Sie wurde, wie in 3.2 beschrieben wird, generalisiert. Zu Beginn der Ausführung des Verwaltungsprozesses muss ein Parameter übergeben werden, um welchen Typen von Fahrzeug es sich handelt (derzeit mögliche Typen sind *einachs*, *sattelzug* und *zweiachs*). Diesem Parameter entsprechend wird eine geeignete Assistenz gewählt. Sollten in Zukunft noch weitere Fahrzeugtypen oder Assistenztypen hinzugefügt werden, muss gegebenenfalls in der Verwaltung die Instanziierung und Konfiguration der Assistenzen angepasst werden.

2.2.3 EreignisverteilerServer

Der *EreignisverteilerServer* vermittelt zwischen den an der Steuerung des Fahrzeugs beteiligte Prozessen (für Details zur Kommunikation zwischen Prozessen siehe 2.2.7.1). Er tauscht zwischen den Prozessen Ereignisse aus, wobei die Behandlung empfangener Ereignisse in vielen Fällen an den Verwaltungsprozess delegiert wird.

Dieser Prozess ist für die Lenkassistentz nur indirekt von Bedeutung. Sie muss nur mit der Verwaltung kommunizieren, um Ereignisse zu versenden und zu empfangen, und hat daher keinen direkten Kontakt mit dem Ereignisverteiler-Server.

2.2.4 Hardwarestub

Der *Hardwarestub* bildet die Schnittstelle zur Hardware des Fahrzeugs. Er greift periodisch auf die Potentiometer zu, die die Einknickwinkel messen, und stellt diese Informationen für Interessenten über Ereignisse zur Verfügung. Außerdem setzt er Steuerungs-Ereignisse um, indem er die Motoren des Fahrzeugs bedient, ohne dass die den Hardwarstub benutzenden Prozesse die technischen Details

des Fahrzeugs kennen müssen. Der Hardwarestub kann daher auch als *Treiber* für die Fahrzeuge angesehen werden.

Im Falle einer Simulation übernimmt der Hardwarestub die Simulation der Fahrzeugbewegungen.

Auch der Hardwarestub ist für die Entwicklung einer Lenkassistentz nicht direkt interessant. Die benötigte Logik zum Steuern und Überwachen des Fahrzeugs ist vollständig entkoppelt von der Assistentz, so dass diese nur die relevanten Ereignisse behandeln muss.

2.2.5 Kinematikaufbereitung

Die *Kinematikaufbereitung* ist eine optionale Komponente, die durch Interpolation die gemessenen Fahrzeugdaten aufbereitet, um Messfehler und kurzzeitige Ausfälle der Sensoren zu kompensieren. Sie stellt die errechneten Daten über Ereignisse zur Verfügung, sodass andere Komponenten die Wahl zwischen normalen und aufbereiteten Daten haben.

Die Komponente ist nicht relevant für die Lenkassistentz.

2.2.6 Laser

Der *Laser*-Prozess ist eine weitere optionale Komponente, die mit dem Hardware-Laser auf dem Fahrzeug interagiert (wenn vorhanden), um Positions- und Ausrichtungsdaten zu sammeln. In einer Simulation wird dieser Prozess nicht ausgeführt.

Die Komponente ist nicht relevant für die Lenkassistentz.

2.2.7 Wichtige Details von EZAuto

Nachdem nun die an der Ausführung der Assistentz beteiligten Programmteile erläutert wurden, soll noch kurz auf einige für den Entwickler wichtige Details eingegangen werden, um einen leichteren Einstieg in das recht umfangreiche EZAuto-Projekt zu erleichtern.

2.2.7.1 Verteilte Programmausführung

Leitstand und der Rest der Komponenten müssen nicht auf derselben Maschine ausgeführt werden. Die Kommunikation⁴ erfolgt über eine in EZAuto integrierte Bibliothek zum Austausch von Ereignissen. Das zugrundeliegende Protokoll zum Übertragen der Ereignisse ist TCP/IP. Aus diesem Grund beinhaltet die in 2.2.1 erwähnte Konfigurationsdatei vor allem IP-Adresse und dem Port des EreignisverteilerServer-Prozesses.

Ereignisse (auch: *Events*) werden durch ihren beliebig wählbaren Typen voneinander unterschieden. Dieser kann durch ein spezielles string-Attribut des Event-Objekts festgelegt werden. Weiterhin können einem Event unter einem beliebigen Bezeichner Daten von den Typen *bool*, *int*, *double* und *string* zugeordnet werden. Außerdem ist es möglich, komplexe Objekte zu speichern, allerdings müssen diese in Form eines Schnittstellenzeiger⁵ auf das Objekt übergeben werden.

Die Verteilung der Ereignisse wird durch dem oben beschriebenen *EreignisverteilerServer* realisiert.

2.2.7.2 Smart Pointer

EZAuto bringt seine eigene Implementation einer Smart Pointer Bibliothek mit sich. Nach der Boost Dokumentation [Boo15]⁶ sind Smart Pointer:

Smart pointers are objects which store pointers to dynamically allocated (heap) objects. They behave much like built-in C++ pointers except that they automatically delete the object pointed to at the appropriate time. Smart pointers are particularly useful in the face of exceptions as they ensure proper destruction of dynamically allocated objects. They can also be used to keep track of dynamically allocated objects shared by multiple owners.

Die Implementation der Smart Pointer liegt in `src/lib/ezbasis/EZSchnittstelle`. Jede Klasse, die auf deren Instanzen als Smart Pointer zugegriffen werden soll, muss eine *Schnittstelle*

⁴Inter-process communication

⁵Siehe 2.2.7.2

⁶Boost ist eine bekannte Sammlung von C++ Bibliotheken, die unter Anderem eine Implementierung zu Smart Pointern bereitstellt

implementieren. Daher werden in **EZSchnittstelle** neben der eigentlichen Implementation der Smart Pointer auch Makros zum Implementieren der Schnittstellen bereitgestellt. In einem späteren Teil der Arbeit wird eine neue Softwarearchitektur für Lenkassistenzen vorgestellt, die das Entwickeln weiterer Assistenzen vereinfacht. Da die Klassen einer Assistenz eine solche Schnittstelle implementieren müssen, ist es wichtig, dass der Entwickler richtig mit der Schnittstellen-Bibliothek umgeht. Die korrekte Nutzung kann exemplarisch im Code der **Lenkassistentenzweiachs** eingesehen werden.

2.3 Kinematische Grundlagen

In dieser Sektion sollen alle wichtigen mathematischen, beziehungsweise kinematischen Grundlagen beschrieben werden. Auf ihnen basiert die Logik des Lenkassistentensystems. Zunächst wird das in [Zöb01] ursprünglich eingeführte Einspurmodell beschrieben. Es liefert ein ausreichend abstraktes Modell, um relativ einfach mit Gliederfahrzeugen rechnen zu können. Dann wird dieses abstrakte Modell angewandt auf die von der Assistenz zu unterstützende Art von Gespann und die wichtigsten Bezeichner festgelegt, da diese in der Literatur nicht einheitlich sind. Zuletzt werden mit Hilfe des spezialisierten Modells und dessen Bezeichnern einige für die Assistenz zentrale Funktionen hergeleitet und kurz erklärt.

2.3.1 Das Einspurmodell

Das Einspurmodell (nach [Zöb01] Fahrradmodell) ist eine vereinfachte Darstellung eines mehrgliedrigen Fahrzeuggespanns. Das Modell beschreibt auf kinematischer Ebene, wie das Gespann aus mehreren eindimensionalen, mit einander an Kupplungspunkten verbundenen Teilen aufgebaut ist, sowie deren Verhalten. Da das Modell die Grundlage aller Berechnungen der Lenkassistentenz bilden, soll im Folgenden eine kurze Beschreibung nach [Zöb01], Seite 4 erfolgen.

Das Einspurmodell wird auch als *Fahrradmodell* bezeichnet, da jedes Fahrzeugglied durch die gemachten Abstraktionen auf ein fahrradartiges Gerüst reduziert wird. Diese Abstraktionen sind folgende:

1. Jedes Fahrzeugglied hat ein Paar *Lenkräder*⁷ und ein *statisches Rad*. Glieder, die diesem Schema nicht entsprechen, müssen auf dieses Schema vereinfacht werden.
2. Jedes Paar von Rädern wird auf ein einzelnes Rad genau in der Mitte der beiden Räder vereinfacht
3. Jedes Rad berührt im Mittelpunkt des Rades den Boden⁸

Weiterhin wird angenommen, dass das Fahrzeug keine Masse hat, damit keine Bewegungsträgheit erfährt und so keine Reibung mit der Straße erfolgt. Diese Annahme kann bei starken brems- oder beschleunigungs-Manövern zu signifikanten Fehlern führen, was aber für die Rückfahrsistenz nicht relevant ist, da diese gewöhnlich bei sehr niedrigen Geschwindigkeiten operiert.

Bewegt sich ein solches Fahrzeugglied nun auf einer Kurve, befahren die beiden Räder konzentrische Kreise⁹. Wenn l der Abstand zwischen Lenkrad und statischem Rad ist, und r_0 und r_1 die Radien der konzentrischen Kreise sind, ist der Winkel zwischen den beiden Radien gegeben als:

$$\alpha = \cos\left(\frac{l}{r_0}\right)$$

$$\alpha = \arctan\left(\frac{l}{r_1}\right)$$

Wird nun der Fahrzeugzug um ein Glied erweitert, muss dieses den oben genannten Abstraktionen entsprechen. Das weitere Glied wird mit einem Abstand m hinter das statische Rad angehängen. Abbildung 2.4 stellt die resultierende Konfiguration grafisch dar¹⁰.

Der Vorgang kann prinzipiell beliebig oft wiederholt werden, das resultierende Gespann stellt dann einen *General-n-Trailer* dar. Im Folgenden soll aber lediglich ein spezifischer General-3-Trailer betrachtet werden, da die Lenkassistentz ausschließlich für diesen entwickelt werden soll. Trotzdem werden die behandel-

⁷Gemeint sind hier lenkende Räder auf der Straße, nicht die Betätigungseinrichtung

⁸In [Zöb01] sind die Aufzählungspunkte 2 und 3 getauscht

⁹Kreise mit dem selben Mittelpunkt, aber nicht unbedingt gleichen Radien

¹⁰Die Bezeichner der Abbildung entsprechen nicht den in dieser Arbeit genutzten Bezeichnern, die in der nächsten Untersektion ausführlich erklärt werden

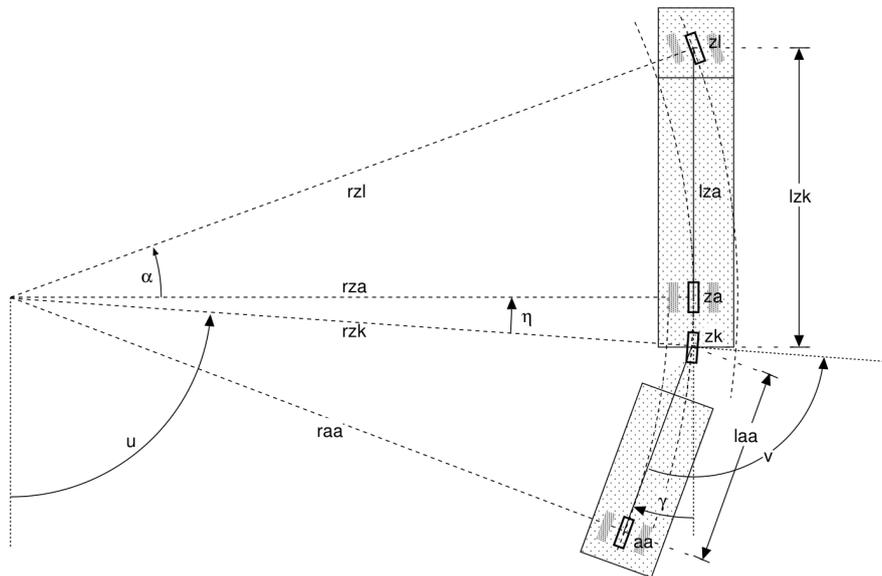


Abbildung 2.4: Einspurmodell mit 2 Gliedern. Quelle: [Zöb01]

ten Sachverhalte möglichst generell beschrieben, um die Entwicklung alternativer Assistenzsysteme auf Basis dieser Arbeit so einfach wie möglich zu machen.

2.3.2 Definition von Bezeichnern

Im Folgenden wird eine Vielzahl von mathematischen Bezeichnern benutzt, um die Eigenschaften und den Zustand des behandelten General-3-Trailer zu beschreiben. Sie sollen an dieser Stelle kurz vorgestellt und erklärt werden. Abbildung 2.5 zeigt ein solches Gespann mit zweiachsigem Anhänger in einer schematischen Darstellung nach dem Einspurmodell, inklusive aller wichtigen Bezeichner.

Die Bezeichner l_1 , m_1 , l_2 und l_3 sind Längen-Eigenschaften des Fahrzeugs:

- l_1 ist der Abstand zwischen Vorder- und Hinterrädern des Zugfahrzeugs
- m_1 ist der Abstand der Hinterräder zur Kupplung des Zugfahrzeugs
- l_2 ist die Länge des ersten Anhängerteils
- l_3 ist die Länge des zweiten Anhängerteils

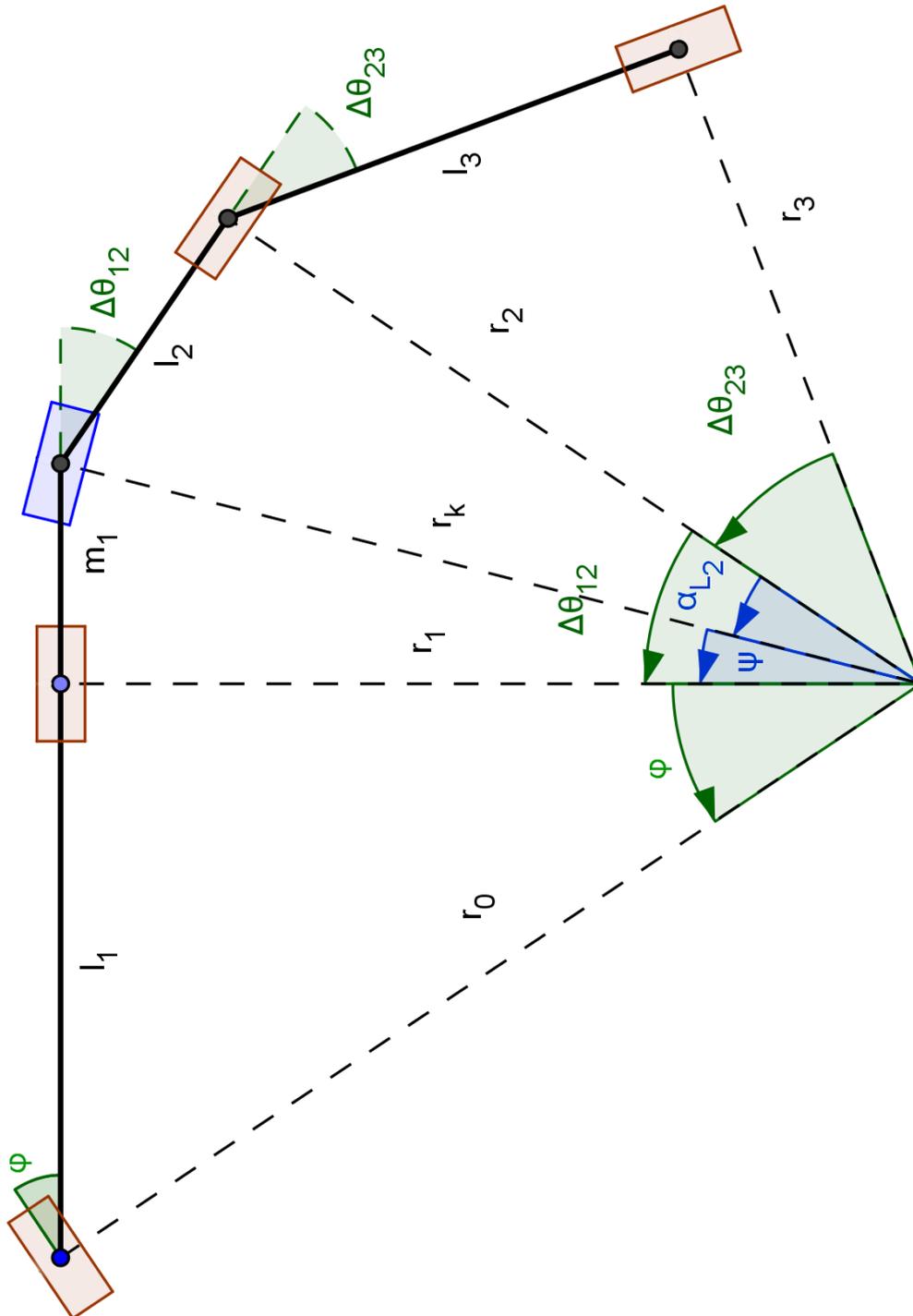


Abbildung 2.5: Stabile Fahrt eines General-3-Trailers

Im Falle eines Drehschemelanhängers ist l_2 gleichzeitig der Abstand von Kupplung des Zugfahrzeugs zu den vorderen Rädern des Anhängers. l_3 ist der Abstand dieser Räder zu den hinteren Rädern des Anhängers. Die Indizes entsprechen der jeweiligen Nummer des Fahrzeuggliedes, wobei 1 das Zugfahrzeug bezeichnet. Der Wertebereich aller Maße des Fahrzeugs sind alle reellen positiven Zahlen.

Alle Räder des Einspurmodells, inklusive des virtuellen Rads an der Kupplung, würden eine Kreisbahn befahren, wenn sie nicht durch den Rest des Fahrzeugs beeinflusst würden. Die Radien dieser Kreisbahnen werden bezeichnet als r_0, r_1, r_k, r_2 und r_3 , wobei die Radien theoretisch auch unendlich sein können. Die Indizes geben die Nummer des Kupplungspunktes an, von dem der Radius ausgeht. Kupplungspunkte sind benannt nach dem Fahrzeugglied, an das sie anschließen, wobei der erste Kupplungspunkt 0 tatsächlich an nichts anschließt. r_k ist ein Sonderfall, da m_1 nur eine Verlängerung des ersten Fahrzeuggliedes ist. Der Index k soll an *Kupplung* erinnern. Ist eine stabile Fahrt (Siehe 2.3.3) gegeben, helfen diese Radien bei einigen Berechnungen, zum Beispiel Berechnung für eine stabile Fahrt benötigten Radlenkwinkel. Der Wertebereich aller Radien sind alle reellen, positiven Zahlen. Üblicherweise wird nicht mit negativen Radien gerechnet, sondern gegebenenfalls eine Fallunterscheidung gemacht.

Der Radlenkwinkel wird im Allgemeinen durch φ bezeichnet. Er bezeichnet den Winkel, den das vordere Rad des Zugfahrzeugs (im Einspurmodell) im Bezug auf die Grundstellung eingeschlagen hat. Grundstellung meint in diesem Fall den Radlenkwinkel, den das Fahrzeug einschlagen müsste, um geradeaus zu fahren. Es ist wichtig, den Radlenkwinkel klar vom Lenkradwinkel abzugrenzen. Wie in 2.1.1 beschrieben wurde, sind bei elektronischen Lenkanlagen, auf welche sich diese Arbeit beschränkt, der vom Fahrer am Lenkrad gelenkte Winkel (der *Lenkradwinkel*) und der von den Vorderrädern eingeschlagene Winkel (*Radlenkwinkel*) physisch von einander getrennt. Winkel werden in EZAUTO immer im Bogenmaß benutzt, nur zur Anzeige oder manuellen Eingabe werden sie gelegentlich in Grad umgewandelt. Der Wertebereich von φ ist daher potenziell $[-\pi, \pi]$, allerdings liegen bei herkömmlichen Fahrzeugen die Werte eher im Bereich $[-\frac{\pi}{4}, \frac{\pi}{4}]$, also etwa $[-30^\circ, 30^\circ]$. Diese maximalen Radlenkwinkel werden im Folgenden auch $\varphi_{min,links}$ und $\varphi_{max,rechts}$ genannt. Da sie üblicher Weise absolut betrachtet gleich sind, werden sie auch als φ_{max} zusammengefasst. Das Vorzeichen des Winkels gibt die Lenkrichtung an, also ob nach links oder rechts gelenkt

wird. Welches Vorzeichen welche Richtung angibt, wird allein durch die Hardware des Fahrzeugs vorgegeben.

$\Delta\theta_{12}$ und $\Delta\theta_{23}$ bezeichnen die Einknickwinkel der Anhängerteile in Bezug auf ihr Vorgängerglied. Die Indizes erinnern daran, zwischen welchen Gliedern der Winkel gemessen wird. Üblicher Weise liegen diese Winkel im gleichen oder in einem in beide Richtungen etwas größeren Wertebereich wie φ . Die Messung der beiden Winkel erfolgt durch Potentiometern an den Kupplungspunkten des Fahrzeugs. Mathematisch setzt sich $\Delta\theta_{12}$ zusammen aus dem Radlenkwinkel des auf der Kupplung sitzenden Rades ψ und α_{L_2} . In vielen Fällen, zum Beispiel bei einem Drehschemel-Anhänger, ist das auf der Kupplung sitzende Rad *virtuell*, das heißt, es existiert nicht wirklich, kann aber zur Veranschaulichung der Berechnungen "dazugedacht" werden, um der ersten Abstraktion des Einspurmodells zu entsprechen. Der Wertebereich dieser Winkel wird vorgegeben durch die Fahrzeugphysik. Nach [Wey06] ist die Nomenklatur für die Grenzen des Wertebereichs *kritisch*, da sie gegebenenfalls nie erreicht werden dürfen, um Schäden am Gespann zu vermeiden. Die Wertebereiche sind somit, im Bogenmaß, $[\Delta\theta_{12,kritisch,links}, \Delta\theta_{12,kritisch,rechts}]$ beziehungsweise $[\Delta\theta_{23,kritisch,links}, \Delta\theta_{23,kritisch,rechts}]$ oder auch nur $\Delta\theta_{12,kritisch}$ und $\Delta\theta_{23,kritisch}$, da auch hier wie beim Radlenkwinkel die Grenzen in den meisten Fällen absolut gesehen gleich sind.

2.3.3 Stabile Fahrt

Unter gewissen Umständen können sich die einzelnen Fahrzeugteile eines mehrgliedrigen Gespanns auf einer *stabilen Bahn* befinden. Dieser Umstand tritt ein, wenn das Fahrzeug mit einem beliebigen, aber festen Radlenkwinkel eine ausreichend lange Zeit geradeaus fährt. Dann konvergieren alle Einknickwinkel der nachgezogenen Fahrzeugteile zu den sogenannten *stabilen Einknickwinkeln*. Wird der Radlenkwinkel weiter gehalten, bewegen sich dann alle Räder beziehungsweise Kupplungspunkte auf konzentrischen Kreisen, also Kreisen mit potenziell unterschiedlichen Radien, aber demselben Mittelpunkt. Dieser Zustand wird im Folgenden auch *stabile Fahrt* genannt. Eine besondere Eigenschaft der stabilen Fahrt ist, dass, ist sie einmal erreicht, sich die jetzt konstanten Einknickwinkel auch bei einer idealen Rückwärtsfahrt nicht ändern. Laut [Wey06] existiert für jedes beliebige n -gliedrige Gespann und ein φ ein n -langes Tupel

$(\varphi, \Delta\theta_{1,2}, \dots, \Delta\theta_{n-1,n})$, das die Einknickwinkel für eine stabile Bahn des Gespanns beschreibt, allerdings sind für diese Arbeit nur Gespanne mit $n = 2$ Gliedern relevant.

Im Folgenden soll beschrieben werden, wie aus einem gegebenen Radlenkwinkel auf einer stabilen Bahn die beiden Einknickwinkel berechnet werden können, und wie anders herum der Radlenkwinkel aus den beiden Einknickwinkeln berechnet werden kann.

2.3.3.1 Einknickwinkel aus Radlenkwinkel

Durch die stabile Bahn (für eine schematische Darstellung siehe Abbildung 2.5) sind folgende trigonometrischen Abhängigkeiten gegeben:

$$\begin{aligned} r_1 &= \frac{l_1}{\tan(\varphi)} \\ r_k &= \sqrt{r_1^2 + m_1^2} \\ \psi &= \arctan\left(\frac{m_1}{r_1}\right) \\ \alpha_{L_2} &= \arcsin\left(\frac{l_2}{r_k}\right) \end{aligned}$$

Wie bereits erwähnt werden die absoluten Werte der Radien benutzt. Basierend auf dem Lenkwinkel würde nun eine Fallunterscheidung gemacht werden, um das Vorzeichen des Ergebnisses zu erhalten. Dieses Vorgehen ist aus zwei Gründen nötig: erstens geht durch das Quadrieren in der Rechnung das Vorzeichen verloren, und zweitens ist die Bedeutung der Vorzeichen bei den Fahrzeugen nicht notwendiger Weise konsistent. Die Simulation benutzt zum Beispiel negative Werte um Lenkwinkel nach links darzustellen, aber positive Werte für Einknickwinkel nach links. Daher soll bei den folgenden Formeln der Einfachheit halber angenommen werden, dass alle Winkel positiv sind.

Die Summe der beiden Winkel ψ und α_{L_2} ergibt den Einknickwinkel $\Delta\theta_{12}$ zwischen Zugmaschine und erstem Anhängerteil:

$$\Delta\theta_{12} = \psi + \alpha_{L_2} \quad (2.1)$$

Fasst man diese Formeln zusammen, kann man die Funktion zum berechnen des ersten Einknickwinkels $\Delta\theta_{12}$ auf einer stabilen Bahn mit gegebenem Radlenkwinkel φ in Abhängigkeit von der Fahrzeuggeometrie wie folgt ausdrücken:

$$\Delta\theta_{12}(\varphi) = \arctan\left(\frac{m_1}{\frac{l_1}{\tan(\varphi)}}\right) + \arcsin\left(\frac{l_2}{\sqrt{\frac{l_1}{\tan(\varphi)}^2 + m_1^2}}\right) \quad (2.2)$$

Um den Einknickwinkel des hinteren Anhängerteils zu Berechnen, wird noch eine weitere Abhängigkeit benötigt:

$$r_2 = \sqrt{r_k^2 - l_2^2}$$

Dann kann der hintere Einknickwinkel berechnet werden mit:

$$\Delta\theta_{23} = \arcsin\left(\frac{l_3}{r_2}\right) \quad (2.3)$$

Zusammengefasst kann jetzt ähnlich wie für $\Delta\theta_{12}$ eine Funktion für $\Delta\theta_{23}$ angegeben werden:

$$\Delta\theta_{23}(\varphi) = \arcsin\left(\frac{l_3}{\sqrt{m_1^2 + \frac{l_1}{\tan(\varphi)}^2 - l_2^2}}\right) \quad (2.4)$$

Diese Berechnung hängt natürlich weiterhin von den in der Formel benutzten geometrischen Fahrzeuggrößen ab.

2.3.3.2 Radlenkwinkel aus Einknickwinkeln

Ähnlich wie aus dem Radlenkwinkel die Einknickwinkel berechnet werden, kann auf umgekehrtem Wege auch der Radlenkwinkel aus jeweils einem der (absoluten) Einknickwinkel wie folgt hergeleitet werden:

$$\begin{aligned} r_2 &= \frac{l_3}{\sin(\Delta\theta_{23})} \\ r_k &= \sqrt{l_2^2 + r_2^2} \\ r_1 &= \sqrt{r_k^2 - m_1^2} \end{aligned}$$

An dieser Stelle sei noch einmal darauf hingewiesen, dass diese Berechnung

nur auf einer stabilen Bahn gültig ist.

Die erste Formel bricht für $\Delta\theta_{23} = 0^\circ$, also wenn das Fahrzeug gerade steht. Daher sei für diesen Fall festgelegt, dass $\varphi = 0^\circ$ sein muss. Andernfalls ist der Lenkwinkel φ dann:

$$\varphi = \arctan\left(\frac{l_1}{r_1}\right) \quad (2.5)$$

Oder zusammengefasst und als Funktion ausgedrückt:

$$\varphi(\Delta\theta_{23}) = \arctan\left(\frac{l_1}{\sqrt{l_2^2 + \frac{l_3^2}{\sin^2(\Delta\theta_{23})} - m_1^2}}\right) \quad (2.6)$$

Auch hier sind natürlich die geometrischen Fahrzeuggrößen zu beachten. Die Berechnung von $\Delta\theta_{12}$ bei einem gegebenen φ erfolgt analog, soll an dieser Stelle allerdings nicht noch einmal ausgeführt werden, da sie für die Assistenz nicht benutzt wurde.

2.3.3.3 Maximale Winkel

Da der Radlenkwinkel des Fahrzeugs aus technischen Gründen beschränkt ist auf das Intervall $[\varphi_{max,links}, \varphi_{max,rechts}]$, ist auch die stabile Fahrt beschränkt. Führt das Fahrzeug mit maximalem Radlenkwinkel φ_{max} auf einer stabilen Bahn, stellen sich die beiden Einknickwinkel auf ihre stabilen Maxima ein. Diese werden auch als $\Delta\theta_{12,limit}$ und $\Delta\theta_{23,limit}$ bezeichnet. Praktisch können diese Winkel auch noch größer werden, ohne dass Schäden am Gespann entstehen, aber im Rahmen der in 3 vorgestellten Assistenz ist es wichtig, nie die Maxima auf der stabilen Bahn zu verlassen, um zu gewährleisten, dass das Fahrzeug jederzeit noch steuerbar ist. Steuerbar bedeutet, dass die Assistenz (während der Rückwärtsfahrt) noch in der Lage ist, das Fahrzeug in jeden anderen steuerbaren Zustand zu überführen.

Die maximalen Einknickwinkel bei stabiler Fahrt können durch Einsetzen des maximalen Radlenkwinkels in die Funktionen 2.2 und 2.4 errechnet werden. Es gilt also:

$$\Delta\theta_{12,max} = \Delta\theta_{12}(\varphi_{max}) \quad (2.7)$$

$$\Delta\theta_{23,max} = \Delta\theta_{23}(\varphi_{max}) \quad (2.8)$$

Die Maximalwinkel geben direkt den Wirkungsgrad der Lenkassistenz vor. Gemäß der Intuition benötigt ein agileres Fahrzeug mit größerem Radlenkintervall (und somit größeren Einknickwinkel-Intervallen) in den meisten Fällen weniger Züge, um ein Manöver auszuführen, als ein Gespann mit den gleichen Maßen, aber kleineren Operationsintervallen.

Kapitel 3

Entwicklung des Assistenzsystems

In diesem Kapitel wird das entwickelte Assistenzsystem im Detail erläutert. Das Kapitel soll alleinstehend dem Leser ein umfassendes Verständnis des Systems vermitteln, und zusammen mit der Dokumentation im Quelltext der Assistenz anderen Entwicklern ermöglichen, auf der hier getanen Arbeit aufzubauen und das System weiterzuentwickeln. Dazu wird zunächst die genaue Funktionsweise der Lenklogik erklärt. Dann wird die gewählte Softwarearchitektur dokumentiert, und zum Schluss die auf die Anbindung zum Leitstand und die Thematik der Mensch-Maschine-Schnittstelle eingegangen.

3.1 Lenklogik

Die Lenklogik ist der Teil der Fahrassistenz, der die Eingaben des Fahrers in tatsächliche Lenkmanöver übersetzt. Dem Lenkradwinkel wird bei der Rückwärtsfahrt eine komplett neue Bedeutung zugeordnet: Der Fahrer bestimmt mit dem Lenkrad nur noch den Einknickwinkel des hinteren Fahrzeuggliedes. Somit verhält sich das Fahrzeug aus Sicht des Fahrers wie ein Auto ohne Anhänger und ist daher wesentlich einfacher zu steuern. Dies ist möglich, da bei elektronischen Lenkanlagen Lenkradwinkel und Radlenkwinkel technisch voneinander entkoppelt sind, und so eine Fahrassistenz zwischengeschaltet werden kann, die diese Beziehung neu definiert.

Beim Übersetzen des Lenkradwinkels L auf einen Soll-Einknickwinkel $\Delta\theta_{23,soll}$ werden 0° Lenkradwinkel gemäß der Intuition auf 0° Einknickwinkel übertragen. Die beiden maximalen Lenkradwinkel $L_{max,links}$ und $L_{max,rechts}$ werden auf die jeweiligen Limits des Einknickwinkels $\Delta\theta_{23,limit,links}$ und $\Delta\theta_{23,limit,rechts}$ übersetzt, und die Werte zwischen den Extremen linear interpoliert. Es ist sehr wichtig, zu beachten, dass die maximalen Lenkradwinkel nicht auf die maximalen Einknickwinkel übertragen werden, sondern nur auf die absolut gesehen kleineren Limits (die zum Beispiel 90% der Maxima betragen). Der maximale Einknickwinkel ist definiert als der Einknickwinkel bei einer stabilen Fahrt mit maximalem Radlenkwinkel und errechnet sich für eine gegebene Gespannkonfiguration über die stabile Fahrt wie in 2.3.3.3 ausführlich beschrieben, indem der die jeweiligen Lenkradwinkel-Maxima als Radlenkwinkel eingesetzt werden. Würde man dem Fahrzeug erlauben, diese maximalen Einknickwinkel einzunehmen, könnte man diese Zustände nicht mehr ohne Vorwärtsfahren verlassen, da die dafür benötigten Manöver immer größere Radlenkwinkel benötigen, als für die stabile Fahrt gehalten werden müssen, was nicht mehr möglich ist, wenn der Radlenkwinkel φ_{stabil} schon maximal φ_{max} ist. Das heißt, es muss jeder Zeit gelten, dass $\varphi_{stabil} < \varphi_{max}$ (mit einem realistischem Abstand).

Im Gegensatz zum Fahren ohne Anhänger sind Zustandsänderungen, das heißt Änderungen an den Einknickwinkeln, nicht sofort möglich. Es müssen vorgegebene Limits eingehalten werden, und das Manöver besteht aus vielen Schritten, die selbst jeweils eine gewisse Zeit in Anspruch nehmen.

Die Lenklogik kennt folgende Manöver:

- Die Einknickwinkel ändern
- Die Einknickwinkel halten

Die Lenklogik geht davon aus, dass sich das Fahrzeug zu Beginn der Rückwärtsfahrt auf einer stabilen Bahn befindet. Tut es das nicht, wird eine Warnung ausgegeben und das Verhalten der Assistenz ist indeterminiert; es ist möglich, dass das Manöver trotzdem erfolgreich durchgeführt werden kann, aber nicht garantiert.

Das Ändern der Einknickwinkel wird durch zwei Ereignisse aktiviert: Der Fahrer ändert den gewünschten Winkel, oder das Fahrzeug verlässt von selbst den gewünschten Winkel. Letzterer Fall kann auftreten durch externe Einflüsse

wie durch die Beschaffenheit der befahrenen Straße, oder durch das Verhalten des Fahrzeugs über Zeit. Theoretisch ist es möglich, eine perfekte stabile Bahn zu befahren, die nie verlassen wird, allerdings existieren in der Praxis selbst gewisse Ungenauigkeiten, die mit der Zeit zu signifikanten Abweichungen führen.

Gehalten werden die Einknickwinkel, wenn sich das Fahrzeug mit einer gewissen Präzision auf einer stabilen Bahn befindet. Es wird der für die stabile Fahrt benötigte Lenkwinkel berechnet und eingestellt. Gleichzeitig werden die Einknickwinkel überwacht. Weicht der hintere Einknickwinkel von dem vorgegebenem Winkel zu weit ab, wird der Änderungs-Modus der Assistenz aktiv, korrigiert den Winkel und bringt das Fahrzeug auf eine stabile Bahn.

Im Folgenden bezeichnet *Manöver* die beiden beschriebenen "Modi" der Lenklogik: Winkel ändern und Winkel halten. Das Änderungs-Manöver ist, wie in der nächsten Untersektion im Detail erklärt wird, aufgeteilt in zwei *Schritte*, die mehrere Male wiederholt werden. Eine solche Wiederholung wird als ein *Zug* bezeichnet.

3.1.1 Einknickwinkel ändern

Ziel des Manövers ist es, den hinteren Ist-Einknickwinkel des Fahrzeugs $\Delta\theta_{23,ist}$ auf den durch den Fahrer vorgegebenen Soll-Einknickwinkel $\Delta\theta_{23,soll}$ zu bringen, und das Fahrzeug gleichzeitig zum Ende des Manövers in einer stabilen Fahrt zu hinterlassen.

Es gibt zwei analoge Fälle, die unterschieden werden müssen: In Fahrtrichtung gesehen, den die Kurve, die das Fahrzeug fährt, nach links oder nach rechts enger zu machen. Dies beinhaltet auch, gegebenenfalls die Kurve zum Beispiel aus einer links-Kurve in eine rechts-Kurve zu überführen. Die jeweils benötigten Schritte sind grundsätzlich die gleichen, nur sind die Lenkeinschläge invers. Da die Messdaten vom Fahrzeug Werte im Bogenmaß mit negativen Werten für Einknick nach links und positiven Werten für Einknick nach rechts liefern (oder umgekehrt), bietet es sich an, die beiden Fälle als *UP* ($\Delta\theta_{23,ist} < \Delta\theta_{23,soll}$) und *DOWN* ($\Delta\theta_{23,ist} > \Delta\theta_{23,soll}$) zu bezeichnen. Da diese Unterscheidung trivial ist, und die in beiden Fällen benötigten Schritte analog sind, werden im folgenden die Schritte allgemein beschrieben, ohne die beiden Fälle zu unterscheiden.

Um den Einknickwinkel zu ändern sind zwei Schritte nötig:

- Den hinteren Einknickwinkel $\Delta\theta_{23,ist}$ verschieben in Richtung $\Delta\theta_{23,soll}$

- Den vorderen Einknickwinkel $\Delta\theta_{12,ist}$ zurück auf die stabile Bahn nachziehen

Die beiden Schritte zusammen bilden einen Zug. Es werden mehrere Züge (mindestens einer) nacheinander ausgeführt, bis der gewünschte Winkel erreicht ist, und die stabile Fahrt besteht. Die Assistenz wechselt also zwischen diesen beiden Sub-Zuständen hin und her, wobei immer mit dem ersten begonnen, und mit dem zweiten aufgehört wird. Wann zwischen den beiden Schritten gewechselt wird, und wann das Manöver beendet wird, wird im Folgenden beschrieben.

Der erste Schritt verschiebt den fraglichen hinteren Einknickwinkel $\Delta\theta_{23,ist}$ in Richtung seines Ziels. Dabei vergrößert sich der vordere Einknickwinkel $\Delta\theta_{12,ist}$. Die technische Bezeichnung für diesen Schritt ist *SECOND*, da der *zweite* Einknickwinkel geändert wird. Um dessen Limit einzuhalten muss also der erste Schritt rechtzeitig aufhören, bevor das Limit überschritten wird. Wann genau aufgehört wird, ist ein noch nicht vollständig gelöstes Problem. Später wird ein funktionierender, aber nicht perfekter Ansatz vorgestellt.

Der zweite Schritt zieht den vorderen Einknickwinkel so zu sagen nach, um erstens die stabile Fahrt als Ausgangszustand für den ersten Schritt wieder herzustellen, und zweitens um den ersten Einknickwinkel wieder von seinem Limit zu entfernen, damit er im ersten Schritt wieder vergrößert werden kann. Bei diesem Schritt vergrößert sich der hintere Einknickwinkel wieder, allerdings nicht so viel, wie er im ersten Schritt verkleinert wurde. Die technische Bezeichnung für diesen Schritt ist *FIRST*, da der *erste* Einknickwinkel geändert wird.

Es ist für das Manöver unerheblich, ob die Fahrtrichtung gewechselt wird, also die Winkel 0° passieren. Dieser Spezialfall hatte zu Beginn der Entwicklung mehrere Male zu Problemen geführt, da die Assistenz unterschied, ob der Einknickwinkel absolut vergrößert oder verkleinert werden soll, und dieser Zustand sich um 0° herum schnell mehrere Male änderte.

3.1.1.1 Submanöver

Während dem ersten Schritt (*SECOND*) muss der Radlenkwinkel φ in die Richtung eingeschlagen werden, in die der Einknickwinkel geändert werden soll. Das heißt, wenn der Soll-Einknickwinkel rechts von Ist-Einknickwinkel liegt, müssen die Vorderräder nach rechts eingeschlagen werden. Für Soll-Einknickwinkel auf der linken Seite, muss nach links eingeschlagen werden.

Für den zweiten Schritt *FIRST* muss der Lenkwinkel im Vergleich zum vorherigen Schritt umgekehrt werden, also entgegen der Richtung, in die der hintere Einknickwinkel geändert werden soll. Wenn der Soll-Einknickwinkel also rechts vom Ist-Einknickwinkel liegt, muss jetzt nach links eingeschlagen werden, und wenn er links liegt, muss nach rechts eingeschlagen werden.

In der Simulation hat es sich als effektiv erwiesen, immer voll bis zum Maximum einzuschlagen, was die Anzahl der benötigten Züge, und damit die Dauer des Manövers, minimiert. Dieses Vorgehen ist allerdings in der Praxis nicht sehr tauglich, da dort im Gegensatz zur Simulation das Wechseln zwischen den Lenkeinschlägen nicht sofort geschieht. Beim Wechseln zwischen den beiden Schritten müsste der Winkel jeweils vom einen zum anderen Maximum geändert werden und das Rad so einen weiten Weg zurücklegen. Das würde zu starkem Reifenabrieb führen und könnte für den Fahrer wie eine Fehlfunktion wirken.

Tabelle 3.1: Radeinschläge in Schritten

Schritt	Pos. $\Delta\theta_{23,soll}$	Einschlag
FIRST	links	rechts
FIRST	rechts	links
SECOND	links	links
SECOND	rechts	rechts

3.1.1.2 Übergänge von Schritten

Der erste Schritt zum Anpassen des hinteren Einknickwinkels wird initial betreten, sobald der $\Delta\theta_{23,soll}$ außerhalb des Toleranzbereichs um $\Delta\theta_{23,ist}$ liegt. Es wird basierend auf der Differenz ein Zwischenziel $\Delta\theta_{23,target}$ für den Einknickwinkel gesetzt, mit $\Delta\theta_{23,soll} \leq \Delta\theta_{23,target} < \Delta\theta_{23,ist}$ (UP) oder $\Delta\theta_{23,ist} > \Delta\theta_{23,target} \geq \Delta\theta_{23,soll}$ (DOWN). Ist $\Delta\theta_{23,ist}$ nah genug an $\Delta\theta_{23,soll}$, wird $\Delta\theta_{23,target}$ auf $\Delta\theta_{23,soll}$ gesetzt. Der erste Schritt wird beendet und es wird der zweite Schritt zum Anpassen des vorderen Lenkwinkels übergegangen, sobald $\Delta\theta_{23,target}$ erreicht ist.

Wie genau $\Delta\theta_{23,target}$ bestimmt wird, kann durch eine Abbildungsfunktion festgelegt werden. Es sind noch weitere Überlegungen nötig, wie genau diese Funktion aussehen muss oder ob ein derartiges Vorgehen überhaupt sinnvoll ist.

Es ist in jedem Fall wichtig, dass der Wert nicht zu weit entfernt vom Ist-Wert gesetzt wird, da sonst während dem ersten Schritt $\Delta\theta_{12,limit}$ überschritten wird, was das Gespann unsteuerbar machen würde, aber auch nicht zu klein, damit nicht zu viele Züge notwendig sind. Die Differenz von $\Delta\theta_{23,ist}$ und $\Delta\theta_{23,target}$ sollte also *so groß wie möglich* sein.

Dieses Vorgehen mit den Zwischenzielen ist nur ein Lösungsansatz. Es fehlen die mathematischen Grundlagen um vorherzusagen, wie sich die Einknickwinkel bei einer gegebenen Fahrzeugkonfiguration bei einer gegebenen Fahrt verhalten würden. Optimal wären die Schritte zwischen den Zwischenschritten so groß wie möglich, ohne die Maxima der Winkel zu erreichen. Allerdings müsste auch bei einer perfekten Vorhersage ein gewisser Spielraum eingeplant werden, um Messungenauigkeiten und Unregelmäßigkeiten des Fahrzeugs vorzubeugen.

Der zweite Schritt zum Anpassen des vorderen Lenkwinkels läuft so lange, bis die stabile Fahrt wieder hergestellt ist, beziehungsweise bis eine stabile Fahrt mit einem entsprechenden Lenkwinkel wieder möglich wäre. Dafür wird kontinuierlich durch einen Algorithmus geprüft, ob dieses Kriterium erfüllt ist. Danach wird geprüft, wie weit der hintere Einknickwinkel noch von seinem Soll-Wert entfernt ist. Befindet sich diese Differenz innerhalb eines akzeptablen Rahmens, geht die Assistenz in den *Einknickwinkel-halten*-Zustand über. Wenn nicht, wird $\Delta\theta_{23,target}$ neu berechnet und mit dem ersten Schritt zum Ändern des hinteren Einknickwinkels erneut begonnen.

In Abbildung 3.1 wird der Verlauf der beiden Einknickwinkel ($\Delta\theta_{23,ist}$ (rot) und $\Delta\theta_{12,ist}$ (violett)). Als der Fahrer $\Delta\theta_{23,soll}$ auf ca -0.13 setzt und rückwärts fährt, wird die Assistenz aktiv und verringert im *SECOND*-Schritt den hinteren Einknickwinkel, bis das Zwischenziel $\Delta\theta_{23,target}$ erreicht ist. Deutlich ist zu sehen, wie dabei der vordere Einknickwinkel in die entgegengesetzte Richtung steigt. Danach wechselt sie zum *FIRST*-Schritt und holt den vorderen Einknickwinkel wieder ein, bis die beiden Einknickwinkel zusammen wieder eine stabile Bahn befahren könnten. Auch hier ist zu sehen, wie der im ersten Schritt mit dem hinteren Einknickwinkel gemacht Fortschritt teilweise wieder verloren geht, allerdings nicht vollständig, sodass netto dem gewünschten Einknickwinkel dennoch näher gekommen wird. Es werden hier insgesamt 4 Züge benötigt, bis $\Delta\theta_{23,ist} \approx \Delta\theta_{23,soll}$, mit einer gewissen Toleranz (in diesem Fall etwa 0.03).

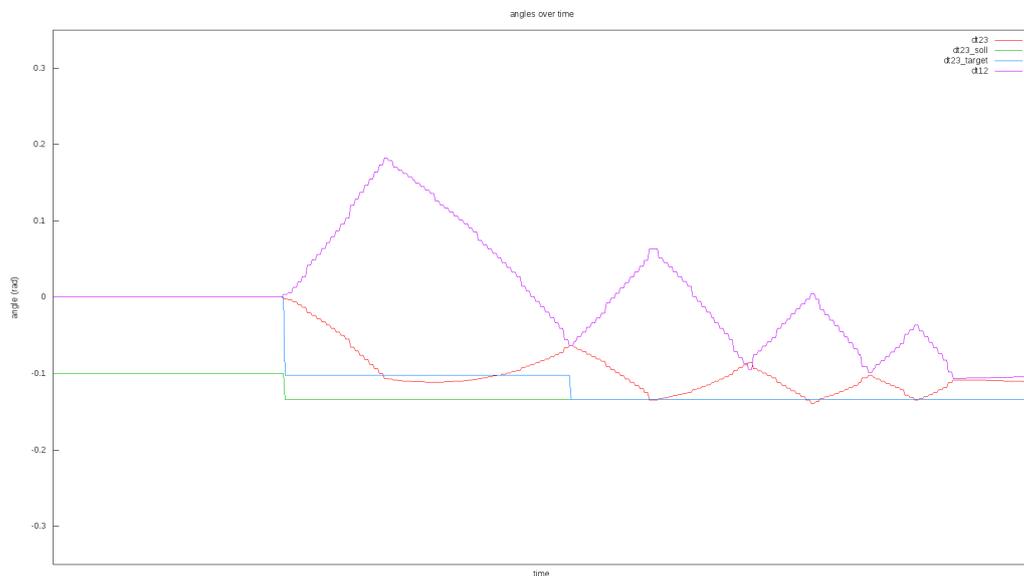


Abbildung 3.1: Verlauf der Einknickwinkel während eines Manövers

3.1.1.3 Superpassing

Um den oben beschriebenen Fortschritts-Verlust im *FIRST*-Schritt auszugleichen, ist es eine Möglichkeit, im *SECOND*-Schritt $\Delta\theta_{23,ist}$ das Zwischenziel $\Delta\theta_{23,target}$ ein kleines Stück überschreiten zu lassen und damit zu spekulieren, dass das überschrittene Stück wieder verloren geht. Somit ist es möglich, die Anzahl der benötigten Züge zu verringern. Bei einer Fahrt ohne Lenkassistenten würde der Fahrer ähnlich vorgehen und selbst schätzen, um wie viel das Zwischenziel überschritten werden darf.

Abbildung 3.2 stellt das gleiche Manöver wie Abbildung 3.1 dar, mit *superpass*-Funktion. Deutlich sieht man, vor allem im zweiten und dritten Zug, wie das blaue Zwischenziel von der roten Ist-Linie überschritten wird, bevor der nächste Schritt aktiviert wird, und die Ist-Linie wieder zurück geht. Vergleicht man die beiden Abbildungen, sieht man, dass mit *superpass* nur 3 Züge, also ein Zug weniger, benötigt wurden, für das gleiche Manöver.

Wie weit genau der Zwischenziel-Wert überschritten werden darf, kann zu diesem Zeitpunkt allerdings nur geschätzt werden. Vermutet wichtige Einschränkungen sind der Abstand des ersten Einknick-Winkels zu seinem Maximum, das durch *superpassing* nicht überschritten werden darf, und der erwartete Fortschritts-Verlust im nächsten Schritt, welcher ausreichen muss, um das *super-*

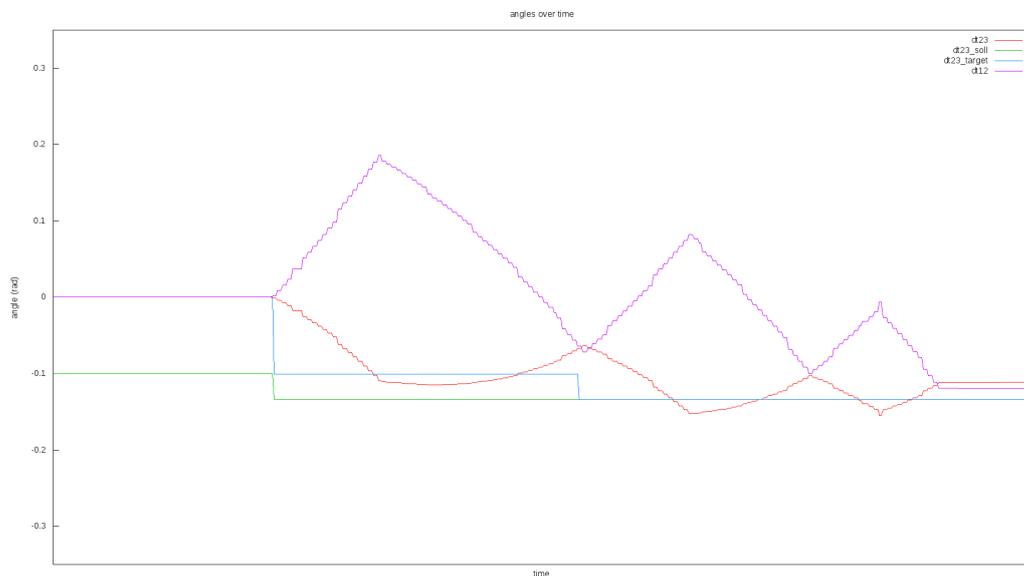


Abbildung 3.2: Verlauf der Einknickwinkel mit superpassing

passing wieder auszugleichen.

Diese Funktion stellt eine Möglichkeit zur Optimierung dar, um die Anzahl der benötigten Züge zu verringern, ist aber völlig optional und wird nicht benötigt für die Funktion der Assistenz.

3.1.1.4 Änderungen am Soll-Einknickwinkel

Während einer Rückwärtsfahrt ist es wahrscheinlich, dass der Fahrer den Soll-Einknickwinkel ändert, bevor der zuvor eingestellte Einknickwinkel erreicht wurde. Die Assistenz muss auf diese Änderung möglichst sofort reagieren, damit der Fahrer weiß, dass seine Änderungen einen Effekt haben werden. Es ist allerdings nicht möglich, während eines Zuges sofort zu reagieren, da sich das Fahrzeug in dieser Zeit in einem unbestimmten Zustand befindet und nicht klar ist, wie auf die Änderung zu reagieren wäre. Aus diesem Grund werden Änderungen erst am Ende eines Zuges umgesetzt.

3.1.2 Einknickwinkel halten

Die zweite wichtige Funktion der Lenkassistenz ist, dafür zu sorgen, dass das Fahrzeug bei der Rückwärtsfahrt seinen Einknickwinkel hält, wenn er dem vom

Fahrer gewünschten Einknickwinkel entspricht. Diese Funktion besteht wiederum aus zwei Aufgaben: Kontinuierlich den Radlenkwinkel für eine stabile Fahrt mit der gegebenen Konfiguration berechnen und einschlagen, und Ausbrüche der Fahrzeugglieder abfangen und korrigieren. Für diese Korrekturen kann die erste Funktion der Assistenz benutzt werden. Der für die stabile Fahrt zu benutzende Radlenkwinkel kann mit Hilfe der Formeln 2.6 berechnet werden.

Eine Schwachstelle dieses Manövers ist seine Unfähigkeit, manche Ausbrüche zu abzufangen. Es geht davon aus, dass sich das Fahrzeug zu Beginn des Manövers annähernd auf einer stabilen Bahn befindet. Erfahrungen in der Simulation haben gezeigt, dass je weiter das Gespann zu Beginn des Manövers von dieser stabilen Bahn entfernt ist, desto wahrscheinlicher ist es, dass es nicht gelingt, das Fahrzeug wieder unter Kontrolle zu bringen. Für Abweichungen von der stabilen Fahrt kann es zwei Gründe geben:

- Kleine Ungenauigkeiten der Winkel verstärken sich über Zeit
- Externe Einflüsse

Ersterer Fall tritt vor allem deswegen auf, weil die Assistenz nie genau die stabile Fahrt anpeilt, sondern nur einen epsilon-Bereich um den benötigten Winkel, um Messungenauigkeiten auszugleichen. Nur auf einer perfekten stabilen Bahn ändern sich beim rückwärts fahren die Winkel nicht.

3.2 Softwarearchitektur

Die programmiertechnische Gestaltung der Assistenz erwies sich im Verlauf der Entwicklung als große Herausforderung. Neben den funktionalen Anforderungen an das System standen zwei nicht-funktionale Anforderungen im Vordergrund: Erhalt der bestehenden Struktur und Assistenzen, und Kozeptionierung eines wartbaren Systems innerhalb einer erweiterbaren und wiederverwendbaren Umgebung.

Die für die Lenklogik relevanten Dateien befinden sich alle in `src/app/ezmodellfahrzeug/ezkomponenten`. Die für Darstellung im Leitstand und generell die Mensch-Maschine Schnittstelle relevanten Klassen befinden sich in `src/app/ezleitstand/module`.

3.2.1 Integration in bestehende Software

Um zu gewährleisten, dass zukünftig einfacher weitere Lenkassistenzen entwickelt werden können, wurde die existierende Lenkassistentz für einachsige Anhänger abstrahiert. Von der abstrakten Klasse **Lenkassistentz** erben jetzt die Klassen **LenkassistentzEinachs** und **LenkassistentzZweiachs**, wobei einige Member und Methoden, die für beide Klassen gleich sind, in die Oberklasse verschoben wurden. Allerdings müssen trotzdem an einigen Stellen in der Verwaltung einige Initialisierungen und dergleichen explizit vorgenommen werden.

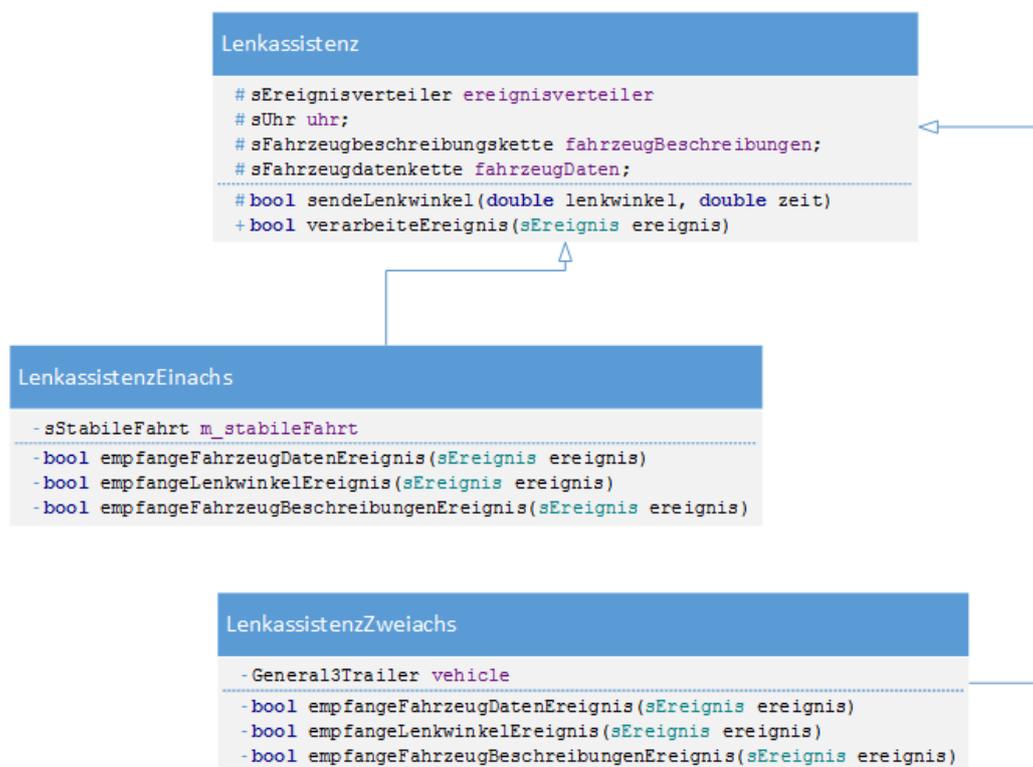


Abbildung 3.3: Vereinfachtes Class diagram der neuen Architektur

Durch die Wahl einer abstrakten Oberklasse sind die erbenden Assistenzen nur noch für die tatsächlich für sie relevanten Programmteile zuständig. Zu diesen gehören vor allem, wie auch in Abbildung 3.3 zu sehen:

- Empfangen von Fahrzeugbeschreibungen
- Empfangen von Fahrzeugdaten

- Empfangen von geänderten Lenkradwinkeln

Dabei wird die tatsächliche Arbeit zum Empfangen von Events über die Interprozesskommunikation von der Oberklasse übernommen, es wird lediglich die Verarbeitung des Ereignis an die Unterklassen delegiert.

Fahrzeugbeschreibungen sind statische Daten, die sich gewöhnlich nicht während der Ausführung ändern. Daher wird diese Funktion in der Regel nur ein Mal, zu Beginn der Ausführung, aufgerufen. Aus der Beschreibung eines Fahrzeugs lassen sich die Parameter für das Einspurmodell errechnen, vor allem l_1 , m_1 , l_2 und l_3 , sowie die maximalen Radlenkwinkel.

Fahrzeugdaten sind dynamische Daten. Sie werden dargestellt durch eine Kette von Fahrzeuggliedern. Von jedem Glied kann der Einknickwinkel zu seinem Vorgänger abgefragt werden, sowie seine Position, seine Ausrichtung und seine Geschwindigkeit¹. Für die Lenkassistentz ist vor allem der Einknickwinkel von Bedeutung.

Ändert der Fahrer den Einschlag des Lenkradwinkels wird ein Ereignis versendet, das alle interessierten Prozesse und damit alle interessierten Komponenten über die Änderung in Kenntnis setzt. Die entsprechende Funktion einer Lenkassistentz erhält dieses Event und kann den neuen Lenkwinkel und den Time-stamp der Änderung aus ihm erhalten.

3.2.2 Aufbau der Zweiachs-Lenklogik

Mit der neuen Architektur steht es jedem Entwickler frei, innerhalb des vorgegeben Rahmens die tatsächliche Logik zum Lenken des Fahrzeugs ohne weitere Restriktionen eigenständig umzusetzen. Da die Rückfahrassistentz für Fahrzeuge mit zweiachsigen Anhängern zumindest beim Rückwärtsfahren eine dauerhafte Überwachung der Parameter und Anpassung des Radlenkwinkels voraussetzt, wird bei Aufruf der beiden letzten oben genannten Funktionen der Zustand des Fahrzeugs bzw. des Lenkradwinkels in der Assistentz-Klasse gespeichert und eine Funktion `update()` aufgerufen. Dadurch, dass dauerhaft Fahrzeugdaten empfangen werden, wird die `update()`-Funktion im Endeffekt kontinuierlich aufgerufen, aber mindestens oft genug um auf Änderungen reagieren zu können. Die genaue Funktionsweise der `update()`-Funktion ist im Code selbst dokumen-

¹Bei einem mehrgliedrigen Gespann haben alle Glieder die gleiche Geschwindigkeit

tiert, aber der Rest dieses Unterabschnitts soll den grundsätzlichen Ablauf grob erklären.

Fährt das Fahrzeug vorwärts oder steht es, wird der vom Fahrer eingestellte Lenkradwinkel direkt als Radlenkwinkel weitergeleitet. Befand sich das Fahrzeug im vorherigen Update in Rückwärtsfahrt, wird die Lenklogik deaktiviert. Außerdem werden gegebenenfalls bisher aufgezeichnete Winkeldaten in eine Datei geschrieben, zu Diagnosezwecken.

Fährt das Fahrzeug rückwärts, wird zunächst geprüft, ob die Einknickwinkel ihre Limits überschreiten und gegebenenfalls eine Warnung ausgegeben. Dann wird unterschieden, in welchem Manöver und gegebenenfalls in welchem Schritt sich die Assistenz befindet, und die entsprechenden Übergangsbedingungen geprüft. Wird eine solche Bedingung erreicht, wird das aktuelle Manöver entsprechend umgeschaltet. Danach werden je nach Manöver der Radlenkwinkel gesetzt.

Am Ende der eines Updates wird der neu gesetzte Radlenkwinkel mit Hilfe einer Funktion der abstrakten Oberklasse als Ereignis an dessen Interessenten gesendet, was vor allem bedeutet, dass der Winkel an das Fahrzeug gesendet wird, wo er dann eingeschlagen wird.

3.3 Anwendungsmodul im Leitstand

Im Rahmen der Entwicklung wurde der Leitstand um ein Anwendungsmodul für die Rückfahrassistenz mit einem zweiachsigen Anhänger ergänzt. Dabei wurde größtenteils das bereits existierende Modul für einachsige Anhänger übernommen. Ursprünglich war es Anforderung, eine zur einachs-Assistenz äquivalente haptische Mensch-Maschine-Schnittstelle umzusetzen. Während der Entwicklung der neuen Assistenz stellte sich jedoch heraus, dass dies nicht möglich ist.

Nach der alten Assistenz entsprach bei der Rückwärtsfahrt der hintere Einknickwinkel dem Winkel, den er bei gleichem Lenkradwinkel beim Vorwärtsfahren einnehmen würde. Da der Fahrer das Lenkrad schneller betätigen kann, als die Assistenz den Einknickwinkel ändern kann, musste der Fahrer darüber informiert werden, dass der von ihm gewünschte Zustand vom Fahrzeug noch nicht erreicht wurde. Dies geschah über eine Krafrückkopplung auf das Lenkrad: Je weiter der Fahrer das Lenkrad und damit den Soll-Einknickwinkel vom

Ist-Einknickwinkel entfernt hatte, desto mehr Kraft musste er der Rückkopplung entgegen setzen. Änderte sich dann durch die Assistenz der Ist-Einknickwinkel in Richtung Soll-Einknickwinkel, ließ die Krafrückkopplung nach.

Die Problematik, dieses Vorgehen auf die oben beschriebene Lenklogik zu übertragen, ergibt sich daraus, dass der Ist-Einknickwinkel während einem Manöver schwankt. Er bewegt sich auf den Soll-Winkel zu, und auch wieder weg, wie in den Abbildungen 3.1 und 3.2 zu sehen ist. Würde man entsprechend der Differenz zwischen Ist- und Soll-Einknickwinkel wie bei der alten Assistenz eine Krafrückkopplung auf das Lenkrad geben, würde auch diese Schwanken und dem Fahrer für ihn unverständliche Informationen liefern. Im schlimmsten Fall merkt er, wie die Kraft zunimmt, obwohl er nichts am Lenkrad ändert, und bekommt den Eindruck, dass die Assistenz falsch, also entgegen seiner Eingabe, arbeitet.

Aufgrund dieser Problematik und der Ermanglung einer bessern Lösung wurde die Anforderung fallen gelassen. Ein Ansatzpunkt für eine alternative Krafrückkopplung könnte aber sein, nicht *kontinuierlich* den Abstand zwischen Ist- und Soll-Einknickwinkel darzustellen, sondern nur den Abstand zu Beginn jedes Zuges.

Kapitel 4

Schlussbemerkungen

Abschließend sollen die im Rahmen dieser Bachelorarbeit entwickelten Modelle und vor allem das entwickelte Lenkassistentensystem mit Hinblick auf die zu Beginn gesetzten Ziele zusammengefasst werden. Nachfolgend sollen noch offene Probleme der Lösung aufgezeigt und die Ergebnisse der Arbeit kritisch bewertet werden. Zum Schluss sollen Ansatzpunkte für weiterführende wissenschaftliche Arbeiten zur Weiterentwicklung des Prototyps thematisiert werden.

4.1 Zusammenfassung

Während der Entwicklung wurde das Einspurmodell benutzt, um die mathematischen Grundlagen für die Lenkassistentenz zu erarbeiten. Im Labor wurde geeignete Manöver entwickelt, um die Einknickwinkel eines zweiachsigen Anhängers im Rahmen einer Lenkassistentenz zu steuern. Im EZAuto-Projekt wurde eine abstrakte Datenstruktur für Lenkassistentensysteme geschaffen und diese genutzt, um auf Basis der erarbeiteten Manöver ein Lenkassistentenz für zweiachsige Anhänger zu implementieren. Die dabei aufgetretenen Schwierigkeiten wurden weitestgehend separiert, so dass sie leicht durch weitere Arbeiten verbessert werden können. Des Weiteren wurde die bestehende Lenkassistentenz für einachsige Anhänger refactored (und in diesem Zuge etwas vereinfacht), um der neuen Architektur zu entsprechen. Eher beiläufig entstanden dabei ein Setup zum automatischen Aufsetzen einer virtuellen Maschine als compilier- und Ausführungsumgebung für

EZAuto¹ und ein kleines Hilfsmittel zum speichern und plotten² der Winkeldaten. Außerdem wurden einige minimale Reparaturen am roten Modellfahrzeug der AG Echtzeitsysteme vorgenommen, um es wieder einsatzfähig zu machen.

Diese Ausarbeitung hat zunächst die technischen Gegebenheiten für die Assistenz ausführlich beschrieben. Dann wurden EZAuto als Software-Rahmen eingeführt, worauf später vor allem die Beschreibung der neuen Lenkassistenten-Architektur aufbaute. Dann wurden die kinematischen Grundlagen erläutert und weiterführend wichtige mathematische Beziehungen für die Assistenz hergeleitet. Diese wurden dann benutzt, um die entwickelte Lenklogik des Prototypen zu beschreiben.

Ziel der Arbeit war es, eine Rückfahr-Lenkassistenten zu entwickeln, bei der der Einknickwinkel des hinteren Anhängerteils während der Rückwärtsfahrt dem Winkel entspricht, wenn das Fahrzeug mit gleichem Lenkradwinkel vorwärts fahren würde. Bei der Entwicklung stellte sich heraus, dass dieses Ziel aus zwei Gründen nur teilweise erreicht werden kann:

1. Um einen vorgegebenen Einknickwinkel zu erreichen *und* das Fahrzeug in einem kontrollierbaren Zustand zu haben, muss der Einknickwinkel sich teilweise vom Zielwinkel entfernen. Also kann die Anforderung nur im Zielzustand erfüllt werden, nicht während dem Manöver.
2. Das Intervall der Einknickwinkel, die während der Vorwärtsfahrt entstehen können, kann nicht auf das Intervall der Einknickwinkel während dem Einsatz der Assistenz projiziert werden, da diese sonst an den Intervallgrenzen nicht mehr operieren könnte.

Aus Grund 2 wurde die Anforderung abgeschwächt. Stattdessen wird jetzt das Intervall der Einknickwinkel beim Vorwärtsfahren auf das Intervall der Limits der Assistenz projiziert. Werden die Limits nahe genug an den Maxima gewählt, ist die ursprüngliche Anforderung beinahe erfüllt (mit der Einschränkung aus Grund 1).

¹Es handelt sich dabei um ein Vagrant-Setup (<https://www.vagrantup.com/>)

²Mit Gnuplot (<http://www.gnuplot.info/>)

4.2 Bewertung

Im Folgenden soll der entwickelte Prototyp kurz bewertet werden und offene Probleme aufgezeigt werden. Dabei liegt ein Fokus auf der von der Lenkassistentz benutzten Methodik, da das kinematische Modell, so wie es in Kapitel 2 beschrieben wurde, als für die Zwecke dieser Arbeit als ausgereift angesehen wird.

Tests in der Simulation haben ergeben, dass sich das Lenkassistentzsystem in den meisten Fällen richtig verhält und das gewünschte Manöver erfolgreich ausführt. Tests mit dem optionalen superpass-Feature lieferten Ergebnisse mit weniger Zügen, was positiv zu bewerten ist, führten aber auch oft zu ungewollter Überschreitung der Einknickwinkel-Limits, zum Erreichen der kritischen Einknickwinkel und somit zum Notstopp des Fahrzeugs.

Tests mit dem roten Modell-LKW lieferten vielversprechende Ergebnisse. Es war möglich, eine 90°-Kurve auf einer auf dem Boden ausgelegten Modellstraße rückwärts zu befahren. Dieses Manöver ist ohne Assistenz schon relativ anspruchsvoll. Positiv zu vermerken ist hierbei auch, dass das Fahrzeug auch in der Realität präzise genug für die Assistenz war. Jedoch war auch hier nicht jeder Versuch erfolgreich, und das Gespann verkeilte sich einige Male.

Dass manche Manöver nicht erfolgreich durchgeführt werden können, liegt vermutlich an einer oder mehreren der folgenden Fehlerquellen:

- Messungenauigkeit beim Messen der Einknickwinkel
- Zu geringe Update-Frequenz der Assistenz-Logik
- Latenz der Interprozesskommunikation
- Ungenaueres Verfahren zum bestimmen der Zwischenziele
- Festlegung von Grenzwerten und Genauigkeiten rein empirisch
- Lenkassistentz geht von stabiler Fahrt aus, die nie genau vorliegt

Die ersten drei genannten Punkte sind besonders kritisch, wenn Glieder des Gespanns stark unterschiedliche Längen haben, wie es bei einem Drehschemelanhänger der Fall ist. Je ungleicher das Verhältnis ist, desto höher ist die benötigte Präzision. Es wird vermutet, dass selbst die Präzision der Simulation nicht in jedem Fall ausreichend ist. Diese Problematik wird deutlich, wenn die Lenkassistentz eine sehr große Änderung am Winkel vornehmen soll. Dann rufen kleinste

Änderung während des *SECOND*-Schritts sehr viel längere *FIRST*-Schritte hervor.

Das Verfahren zur Ermittlung der Zwischenziele basiert nur auf einigen sehr grob gesetzten Rahmenbedingungen und Ausprobieren in der Simulation. Sie ist daher nicht mathematisch perfekt und kann zu Fehlern im Verhalten der Assistenz führen. Außerdem ist die Umsetzung nicht unabhängig vom Gespann, auf dem die Assistenz ausgeführt wird, das heißt, um die Assistenz auf einem anderen Fahrzeug nutzen zu können, muss die Funktion zum Bestimmen von $\Delta\theta_{23,target}$ erneut durch Ausprobieren kalibriert werden. Grundsätzlich ist Kalibrieren, also das Anpassen von Softwareparametern an gegebene Hardware, nicht problematisch, aber hier fehlt ein aufwandsfreier Weg zur Umsetzung. Allerdings wurde die betroffene Logik im Programmcode soweit isoliert, dass das Umstellen auf eine andere Methode keinerlei Probleme verursachen sollte.

Eine weitere Problemquelle sind die beiden zuvor erwähnten Genauigkeiten für die stabile Fahrt und den hinteren Einknickwinkel, die ebenfalls nur auf Beobachtungen basieren. Ersterer darf einerseits nicht zu gering gewählt werden, da es sonst aufgrund der oben erwähnten geringen Präzision passieren kann, dass das Zielintervall "übersprungen" wird, aber auch nicht zu groß, da sonst die Assistenz nicht rechtzeitig erkennt, wenn das Gespann auszubrechen droht, um noch reagieren zu können. Außerdem ist Voraussetzung für ein Manöver der Lenkassistenz, dass sich das Fahrzeug auf einer stabilen Bahn befindet. Je weiter dem Gespann durch einen zu großen Epsilon-Wert erlaubt wird, diesen Zustand tatsächlich zu verlassen, desto höher ist die Chance, dass das Manöver fehlschlägt. Die Präzision zum Erreichen des Zieleinknickwinkels muss ebenfalls mit Bedacht gewählt werden. Wählt man sie zu groß, erreicht das Fahrzeug nie genau genug das vom Fahrer vorgegebene Ziel. Wählt man sie zu klein, werden mehr Züge pro Manöver benötigt. Es ist möglich, dass diesem Problem durch Wahl geeigneter Werte bei superpassing entgegengewirkt werden kann und dass so dieses Epsilon sehr klein gewählt werden könnte.

Die Ausführung eines Manövers der Lenkassistenz beginnt immer mit dem *SECOND*-Schritt. Dieser wird durch entsprechendes Lenken in Richtung des Zielwerts geändert. Dieses Lenken funktioniert allerdings nur garantiert, wenn sich das Fahrzeug auf einer stabilen Bahn befindet. Ist dies nicht der Fall, ist das Verhalten des Fahrzeugs theoretisch unbestimmt. Tests haben aber in der Praxis gezeigt, dass die Assistenz auch in den meisten Fällen noch funktioniert, wenn die

Einknickwinkel leicht von der stabilen Bahn abweichen. Da ein Modell zur genauen Vorhersage des Verhaltens des Fahrzeugs fehlt, kann auch an dieser Stelle nicht garantiert werden, dass das Manöver gelingt.

Weiterhin ist es nicht gelungen, eine haptische Mensch-Maschine-Schnittstelle zu implementieren. Dies ist vor allem darauf zurück zu führen, dass sich das Konzept, das eigentlich von der in [BZ05] beschriebenen einach-Assistenz übernommen werden sollte, während der Entwicklung als ungeeignet herausgestellt hat. Aus Ermangelung einer sinnvollen Alternative wurde die Anforderung daher fallen gelassen.

Fährt der Fahrer mit voll eingeschlagenem Lenkrad geradeaus (egal in welche Richtung), befährt das Fahrzeug, wie in 2.3.3 beschrieben, eine stabile Bahn mit maximalen Einknickwinkeln. Wenn der Fahrer jetzt rückwärts fährt, ist die Assistenz nicht mehr in der Lage, etwas anderes zu tun, als den Einknickwinkel stabil zu halten. Er kann nicht geändert werden, da der *SECOND*-Schritt sofort dazu führen würde, dass $\Delta\theta_{12}$ sein Maximum überschreitet. In diesem Fall kann dem Fahrer eine Warnung ausgegeben werden, dass die Assistenz nicht verfügbar ist, allerdings schränkt auch diese technische Limitierung den Prototypen weiter ein.

Insgesamt ist das Ergebnis dieser Arbeit eher gemischt. Es wurden einerseits wichtige Erkenntnisse zum Verhalten des Fahrzeugs gewonnen und ein begrenzt funktionsfähiger Prototyp entwickelt, der die Ideen der in [BZ05] Lenkassistentz für einachsige Anhänger so weit wie möglich auf zweiachsige Anhänger überträgt. Andererseits ist das Verhalten des Fahrzeugs bei aktiver Assistenz nicht immer vorhersagbar, die Assistenz ist nur unter Einschränkungen nutzbar, und die Assistenz verhält sich in manchen Fällen fehlerhaft. Es wurden mit dem Prototypen wichtige Grundlagen geschaffen, auf denen weitere Projekte oder Arbeiten aufbauen können, aber in ihrem momentanen Zustand ist die Assistenz nicht für den Einsatz unter realen Bedingungen brauchbar.

4.3 Ausblick

Aus den oben genannten Schwächen des Prototypen gehen diverse Ansatzpunkte für weiterführende Projekte hervor. Die Wichtigsten seien hier zum Abschluss noch einmal zusammengefasst:

- Es muss ein Verfahren gefunden werden, dass weniger anfällig ist für die

teilweise geringe Präzision. Aktuell werden relativ große Epsilons toleriert, aber das führt nicht nur zu unvollständigen Ergebnissen, sondern auch zu Fehlern an sich (vergl. vorherige Sektion). Es wäre außerdem nützlich, eine Möglichkeit zu finden, die benutzten Epsilons nicht empirisch bestimmen zu müssen

- Das kinematische Modell muss erweitert werden, um das Verhalten des Fahrzeugs unter allen Umständen präzise vorhersagen zu können
- Basierend auf einem besseren kinematischen Modell müssen die Zwischenziel-Schritte so groß wie möglich gesetzt werden, um die Anzahl der benötigten Züge zu verringern
- Die Abhängigkeit der Assistenz von einer stabilen Bahn zu Beginn der Manöver sollte verringert werden
- Es sollte eine sinnvolle, haptische Mensch-Maschine-Schnittstelle entworfen und entwickelt werden
- Kombination von Anwendungsmodulen des Leitstands für Lenkassistenten
- Die Assistenz sollte nicht immer mit den maximalen Radlenkwinkeln arbeiten

Die Mensch-Maschine-Schnittstelle könnte ferner neben der haptischen Komponente auch um eine optische oder akustische erweitert werden um dem Fahrer gegebenenfalls Informationen über den Zustand des Fahrzeugs zukommen zu lassen, über den er aktuell bei einer Fahrt mit einem echten Fahrzeug überhaupt nicht informiert werden würde.

Es könnte nützlich sein, die Manöver zum Ändern der Einknickwinkel auf General-n-Trailer, also beliebig viele Anhängerteile, zu generalisieren. Nicht nur könnte dies zu einem besseren Verständnis des Verhaltens von General-3-Trailer führen, es würde auch Assistenzen für Gespanne mit mehr als 3 Gliedern möglich machen.

Kapitel 5

Anhang

5.1 Codefragmente

Um die Weiterentwicklung des Prototypen zu erleichtern, sollen an dieser Stelle noch einige wichtige Code-Fragmente in vereinfachter Form gezeigt und erläutert werden. Es wurden einige Bezeichner geändert, um die allein stehenden Fragmente kürzer oder verständlicher zu machen. Vom Anwendungsmodul für den Leistand abgesehen befindet sich der für die zweiachs-Assistenz relevante Code im Paket `src/app/ezmodellfahrzeug/ezkomponenten/` in folgenden Dateien:

- `EZLenkassistentzZweiachs.h`
- `EZLenkassistentzZweiachs.cpp`
- `General3Trailer.h`
- `General3Trailer.cpp`

Es ist dennoch ratsam, für ein besseres Verständnis in den Code zu schauen, um die Fragmente im richtigen Kontext betrachten zu können. Außerdem wurde der Code ausführlich mit Kommentaren versehen.

5.1.1 Übergangsbedingungen

In 3.1.1.2 wurden bereits die Übergänge zwischen den einzelnen Schritten genauer erklärt. Diese werden im Code *subMode* genannt. Außerdem gibt es noch den

übergeordneten *mode*, der entweder *NONE*, *UP* oder *DOWN* ist und angibt, in welche Richtung der Ist-Einknickwinkel korrigiert werden soll. *NONE* bezeichnet das zweite Manöver, den Einknickwinkel stabil halten.

5.1.1.1 Übergang von NONE nach UP/SECOND oder DOWN/SECOND

Aus der `update()`-Funktion in `EZLenkassistentenzweiachs.h`. Erst wird der verbleibende Abstand zum Ziel berechnet. Ist dieser zu groß, wird die Assistenz aktiv und korrigiert in die richtige Richtung. Dabei wird mit dem *SECOND*-Schritt angefangen. Für diesen muss auch immer ein Zwischenziel berechnet werden, was hier in Zeile 11 geschieht.

```
1 double delta = dt23_soll - dt23_ist;
2
3 if (mode == NONE) {
4     if (!deltaAcceptable) {
5         if (delta <= 0) {
6             mode = DOWN;
7         } else {
8             mode = UP;
9         }
10        subMode = SECOND;
11        dt23_target = calculateStepTarget(dt23_ist, dt23_soll);
12    }
13 }
```

5.1.1.2 Übergang von UP/SECOND oder DOWN/SECOND nach UP/FIRST oder DOWN/FIRST

Aus der `update()`-Funktion in `EZLenkassistentenzweiachs.h`. Zunächst wird ein Wert für das Superpassing berechnet, wie genau ist an dieser Stelle nicht relevant. Dann wird entsprechend der Richtung, in die die Assistenz gerade korrigiert geprüft, ob das das Zwischenziel weit genug überschritten, beziehungsweise unterschritten wurde.

```
1 // mode is UP/SECOND or DOWN/SECOND
2
3 // Superpassing
4 double superpass = /*...*/ 0;
5
6 // The actual check
7 if (mode == UP && (dt23_ist - superpass) > dt23_target || mode ←
    == DOWN && (dt23_ist + superpass) < dt23_target) {
8     subMode = FIRST;
9 }
```

5.1.1.3 Übergang von UP/SECOND oder DOWN/SECOND nach UP/FIRST oder DOWN/FIRST oder NONE

Aus der `update()`-Funktion in `EZLenkassistentzweiachs.h`. Falls durch den vorhergegangenen Zug der Einknickwinkel noch nicht nahe genug am Soll-Einknickwinkel liegt, wird ein weiterer Zug initiiert. Andernfalls wird die Assistenz in den Zustand-halten Modus versetzt.

```
1 // mode is UP/FIRST or DOWN/FIRST
2
3 if (vehicleOnStableCourse) {
4     if (!deltaAcceptable) {
5         // We need at least one more turn
6         subMode = SECOND;
7         dt23_target = calculateStepTarget(dt23_ist, dt23_soll);
8     } else {
9         // We have achieved our goal (by some margin)
10        mode = NONE;
11    }
12 }
```

5.1.1.4 Berechnung der Zwischenziele

Aus `EZLenkassistentzweiachs.h`. Die Werte beim Berechnen der Zwischenziele wurden durch Ausprobieren in der Simulation herausgefunden und können für andere Fahrzeuge anders sein. Dies ist genau die Stelle, wo ein Ver-

fahren gefunden werden muss, um die Zwischenziele zu berechnen. Im Grunde genommen wird hier nur abhängig von der Distanz zum Limit des hinteren Einknickwinkels eine Schrittweite berechnet und in die richtige Richtung auf den ist-Wert draufgerechnet. Zusätzlich wird sichergestellt, dass die Zwischenziele nicht oberhalb, beziehungsweise unterhalb des eigentlichen Ziels liegen.

```
1 double calculateDT23Target(double dt23_ist, double dt23_soll) {
2     double step =
3         0.02 +
4         0.2 * (dt23_limit - fabs(dt23_ist));
5
6     if (dt23_ist < dt23_soll) {
7         return fmin(dt23_ist + step, dt23_soll);
8     } else {
9         return fmax(dt23_ist - step, dt23_soll);
10    }
11 }
```

5.1.2 Radwinkel setzen

Aus der update()-Funktion in EZLenkassistentenzZweiachs.h.

Das folgende Code-Fragment zeigt, wie der Radlenkwinkel entsprechend des Zustands der Assistenz gesetzt wird. Soll die Assistenz den Einknickwinkel stabil halten, wird ein entsprechender Lenkwinkel berechnet. Andernfalls schlägt die Assistenz entsprechend des Modus entweder voll nach links oder voll nach rechts ein.

```
1 if (mode == NONE) {
2   lenkwinkel = calculateStableSteeringAngle(dt23_ist);
3 } else {
4   if (mode == UP && subMode == SECOND || mode == DOWN && ↵
        subMode == FIRST) {
5     lenkwinkel = vehicle->HART_RECHTS;
6   } else {
7     // mode == UP && subMode == FIRST || mode == DOWN && ↵
        subMode == SECOND
8     lenkwinkel = vehicle->HART_LINKS;
9   }
10 }
```

Literaturverzeichnis

- [B⁺08] Uwe Berg et al. Steuerungssystem zur regelung der rückwärtsfahrt eines gespans und verfahren zum betreiben eines solchen steuerungs-systems, 2008.
- [Boo15] Boost Documentation. Smart pointers. http://www.boost.org/doc/libs/1_58_0/libs/smart_ptr/smart_ptr.htm, 2015. Stand: 2015-06-21.
- [BSWZ04] Elisabeth Balcerak, Jacek Schikora, Philipp Wojke, and Dieter Zöbel. Maneuver-based assistance for backing up articulated vehicles. In *2004 IEEE Conference on Robotics, Automation and Mechatronics, RAM 2004, December 1-3, 2004, Singapore*, pages 1066–1071, 2004.
- [BZ05] Uwe Berg and Dieter Zöbel. Haptische Lenkassistentz zur Unterstützung der Rückwärtsfahrt von Fahrzeugen mit einachsigen Anhängern. In *6. Berliner Werkstatt Mensch-Maschine-Systeme: Zustandserkennung und Systemgestaltung*, Zentrum für Mensch-Maschine Systeme (ZMMS), Berlin, October 2005.
- [BZ06] Uwe Berg and Dieter Zöbel. Visual steering assistance for backing-up vehicles with one-axle trailer. In Alastair Gale, editor, *Vision in Vehicles 11*, Dublin, Ireland, July 2006.
- [DZ05] Uwe Berg Dieter Zöbel, Philipp Wojke. Lenkkraftprofil für das assistierte rückwärtsfahren mit einachsanhänger. 2005.
- [Lar09] Stefan Larsson. Weight and dimensions of heavy commercial vehicles as established by directive 96/53/ec and the european modular system (ems), 2009.

- [Rat70] Rat der Europäischen Union. Richtlinie 70/311/ewg des rates vom 8. juni 1970 zur angleichung der rechtsvorschriften der mitgliedstaaten über die lenkanlagen von kraftfahrzeugen und kraftfahrzeuganhängern. <http://eur-lex.europa.eu/legal-content/DE/ALL/?uri=CELEX:31970L0311>, 1970. Stand: 2015-06-21.
- [Rat92] Rat der Europäischen Union. Commission directive 92/62/eec of 2 july 1992 adapting to technical progress council directive 70/311/eec relating to steering equipment for motor vehicles and their trailers. <http://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1434891708359&uri=CELEX:31992L0062>, 1992. Stand: 2015-06-21.
- [Rat09] Rat der Europäischen Union. Richtlinie 96/53/eg des rates vom 25. juli 1996 zur festlegung der höchstzulässigen abmessungen für bestimmte straßenfahrzeuge im innerstaatlichen und grenzüberschreitenden verkehr in der gemeinschaft sowie zur festlegung der höchstzulässigen gewichte im grenzüberschreitenden verkehr. <http://www.euro-combi.de/dwl/EU-Richtlinie%2096-53.pdf>, 2009. Stand: 2015-06-22.
- [RB00] J. Reimpell and J.W. Betzler. *Fahrwerktechnik: Fahrwerktechnik. Grundlagen : Fahrwerk und Gesamtfahrzeug, Radaufhängungen und Antriebsarten, Achskinematik und Elastokinematik, Lenkanlage - Federung - Reifen, Konstruktions- und Berechnungshinweise*. Vogel Fachbuch. Vogel, 2000.
- [Wey06] Christian Weyand. Entwicklung eines fahrerassistenzsystems zur unterstützung der rückwärtsfahrt von gespannen mit zweiachsigen anhängern. Master's thesis, 2006.
- [Wik15a] Wikipedia. Eurocombi variationen. https://de.wikipedia.org/wiki/EuroCombi#LKW-Variationen_und_Nutzfahrzeugkombinationen, 2015. Stand: 2015-06-22.
- [Wik15b] Wikipedia. Mensch-maschine-schnittstelle. <https://de.wikipedia.org/wiki/Benutzerschnittstelle#Mensch-Maschine-Schnittstelle>, 2015. Stand: 2015-06-21.

-
- [Wik15c] Wikipedia. Notbremsassistent. <https://de.wikipedia.org/wiki/Notbremsassistent>, 2015. Stand: 2015-06-21.
- [Wik15d] Wikipedia. Spurhalteassistent. <https://de.wikipedia.org/wiki/Spurhalteassistent>, 2015. Stand: 2015-06-21.
- [Woj05] Phillip Wojke. EZauto - Softwarearchitektur für Anwendungen des assistierten und autonomen Fahrens. *Informatik - In Forschung und Technik, Springer Verlag*, 10, (submitted) 2005.
- [Zöb01] Dieter Zöbel. Mathematical modeling of the kinematics of vehicles. In Kamil Hrubina, editor, *Mathematical Modeling of Technical Processes*, pages 178–200. Sokrates/Erasmus Summer School, Prešov, Slovak Republic, July, 2-13 2001.