

BACHELORTHESES
Philipp Badow

Sensorfusion in autonomen Fahrzeugen

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

Philipp Bandow

Sensorfusion in autonomen Fahrzeugen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Bachelor of Science Technische Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr.-Ing. Andreas Meisel
Zweitgutachter: Prof. Dr.-Ing. Thomas Netzel

Eingereicht am: 12. Dezember 2019

Philipp Bandow

Thema der Arbeit

Sensorfusion in autonomen Fahrzeugen

Stichworte

Sensorfusion, Zustandsschätzung, autonomes Fahren, GNSS, Inertialnavigation, LiDAR, Formula Student

Kurzzusammenfassung

Autonomes Fahren gewinnt in unserer Gesellschaft immer mehr an Bedeutung. Es ist ein großer, industrieübergreifender Forschungsschwerpunkt, der hier in der Anwendung auf ein autonomes Rennfahrzeug betrachtet wird. Diese Arbeit soll einen ganzheitlichen Systemansatz für die Realisierung eines autonomen Fahrzeugs auf Basis eines Formula Student Rennwagens aufstellen. Zuerst werden die für das autonome System notwendigen Sensoren und Aktoren thematisiert. Auf Basis dieser wird dann eine mögliche Systemarchitektur vorgeschlagen. Der Hauptfokus liegt hierbei auf der Entwicklung eines Konzepts für die Fusion der Daten der verschiedenen Sensoren, um eine Zustandsschätzung der Fahrzeugposition, Fahrzeuggeschwindigkeit sowie der Orientierung zu ermöglichen.

Philipp Bandow

Title of Thesis

Sensorfusion in autonomous vehicles

Keywords

sensor fusion, state estimation, autonomous driving, GNSS, inertial navigation, LiDAR, Formula Student

Abstract

Autonomous driving is becoming increasingly important in our society. It is a big, inter-industrial research focus, that will here be applied to an autonomous racecar. This thesis shall develop a holistic approach to realising an autonomous vehicle based on a Formula Student racecar. First the sensor and actuators necessary for the autonomous system will be presented. Subsequently a system architecture is proposed. The main focus is on the development of a concept for fusing the data of the different sensors, to enable the estimation of vehicle speed, heading and orientation.

Inhaltsverzeichnis

Abbildungsverzeichnis	viii
Tabellenverzeichnis	ix
Abkürzungen	x
1 Einleitung	1
1.1 Wettbewerbsgestaltung	2
1.1.1 Acceleration	2
1.1.2 Skidpad	3
1.1.3 Autocross & Trackdrive	4
1.2 Resultierende Aufgabenstellung	4
1.3 Aufbau der Arbeit	5
2 Sensoren & Aktoren	7
2.1 Basissystem	7
2.1.1 Motorsteuerung	8
2.1.2 Raddrehzahlsensoren	8
2.1.3 Drosselsteuerung	8
2.1.4 Schaltungssteuerung	9
2.1.5 Kupplungssteuerung	9
2.2 Sensoren des autonomen Systems	10
2.2.1 GNSS mit Trägheitsnavigation	10
2.2.2 Optischer Geschwindigkeitssensor	11
2.2.3 LiDAR	12
2.2.4 Kamera	13
2.3 Aktoren des autonomen Systems	14
2.3.1 Bremsaktor und EBS	14
2.3.2 Lenkungsaktor	14

3	Systemarchitektur	15
3.1	Fahrzeughssysteme	16
3.2	Zeitsynchronisation	18
3.2.1	Globale Zeitbasis	19
3.2.2	CAN-Bus und Echtzeitmesswerte	19
3.3	Koordinatensysteme	19
3.3.1	Globales Koordinatensystem	20
3.3.2	Fahrzeugkoordinatensystem	21
3.3.3	Konstruktionskoordinatensystem	22
3.3.4	Sensorkoordinatensysteme	22
3.4	Sensorreferenzierung	22
3.5	Softwarearchitektur	24
3.5.1	ROS	24
3.5.2	Entwicklungsprozess	24
3.5.3	Laufzeitumgebung im Fahrzeug	26
3.6	LiDAR Pipeline	26
3.6.1	Positionskorrektur	26
3.6.2	Bodentfernung	27
3.6.3	Pylonenerkennung	27
3.7	SLAM	27
4	Zustandsschätzung	29
4.1	Kalman Filter für lineare Systeme	29
4.1.1	Prädiktion	30
4.1.2	Korrektur	30
4.2	Extended Kalman Filter für nichtlineare Systeme	31
4.2.1	Prädiktion	31
4.2.2	Korrektur	32
4.3	Konkrete Anwendung als Fahrzeugzustandsschätzung	32
4.3.1	Zustandsmodell	33
4.3.2	Prozessmodell	33
4.3.3	Messmodell	35
5	Fazit	37
5.1	Zusammenfassung	37
5.2	Ausblick	38

Literaturverzeichnis	39
Glossar	43
Selbstständigkeitserklärung	44

Abbildungsverzeichnis

1.1	Streckenverlauf und Markierungen beim Acceleration	2
1.2	Streckenverlauf und Markierungen beim Skidpad	3
1.3	Streckenverlauf und Markierungen beim Autocross/Trackdrive	4
2.1	Basisfahrzeug HAWKS H14	7
2.2	Drosselklappen Steuergerät	9
2.3	Sensoren und deren Position am Testfahrzeug	10
2.4	Position des LiDAR auf dem Frontflügel des Fahrzeugs	12
2.5	Vergleich der verschiedenen Positionen des LiDAR am Fahrzeug	13
3.1	Architektur des autonomen Systems	15
3.2	Bussysteme im Fahrzeug	17
3.3	Überblick über die Zeitsynchronisation im Fahrzeug	18
3.4	Verschiedene Koordinatensysteme in einem Fahrzeug	20
3.5	Relation zwischen dem ECEF- und ENU-System	21
3.6	Messaufbau für die Sensorreferenzierung	23
3.7	CI-Pipeline des Hauptprojekts	25

Tabellenverzeichnis

1.2	Maximal zu erreichende Punkte bei der Formula Student Driverless	5
3.2	Sensoren und Messfrequenzen	18

Abkürzungen

ABS Antiblockiersystem.

CAD Computer Aided Design.

CAN Controller Area Network.

CI Continuous Integration.

DCU Driverless Compute Unit.

EBS Emergency Brake System.

ECEF Earth-centered, Earth-fixed.

EKF Extended Kalman Filter.

ENU Earth, North, Up.

FSD Formula Student Driverless.

GNSS Global Navigation Satellite System.

IMU Inertial Measurement Unit.

LiDAR Light Detection and Ranging.

PoE Power over Ethernet.

PPS Pulse-Per-Second.

PTP Precision Time Protocol.

Abkürzungen

RMS Root Mean Square.

ROS Robot Operating System.

SLAM Simultaneous Localization and Mapping.

SSD Solid State Disk.

u.i.v. unabhängig und identisch verteilt.

1 Einleitung

In der heutigen Zeit ist autonomes Fahren ein allgegenwärtiges Thema. Autonome Fahrzeuge versprechen unter anderem erhöhte Sicherheit im Straßenverkehr. Das autonome Fahren ist nicht nur ein großer, industrieübergreifender Forschungsschwerpunkt, sondern auch ein Feld mit diversen Lösungsansätzen für die Weiterentwicklung unserer Gesellschaft.

Das große Interesse der Industrie spiegelt sich auch in der Unterstützung einiger Projekte wieder, die in direktem Zusammenhang mit der Lehre und Forschung im Bereich der autonomen Fahrzeuge stehen, z.B. der Carolo Cup¹ oder die Formula Student².

Die Formula Student ist ein internationaler Konstruktionswettbewerb für Studenten unterschiedlichster Fachrichtungen, mit dem Ziel, innerhalb eines Jahres einen Monoposto-Rennwagen im Formelstil zu entwerfen, produzieren und damit gegen die internationale Konkurrenz anzutreten. Es gibt zum aktuellen Zeitpunkt weltweit insgesamt 17 Wettbewerbe, davon acht in Europa, vier in Amerika, vier in Asien sowie einen in Australien. Weltweit nehmen zur Zeit circa 950 verschiedene Teams aus 58 Ländern an den Wettbewerben teil [2].

Das Thema fahrerloser Fahrzeuge hat im November 2015 Einzug in die Formula Student gehalten [14]. Zuerst wurde ein Konzeptwettbewerb durchgeführt und mit den Erkenntnissen daraus wurde dann eine neue Klasse für Formula Student Rennwagen geschaffen. Seit Mitte 2017 ist diese ein fester Bestandteil des jährlich stattfindenden Formula Student Germany Wettbewerbs. Inzwischen wurde dieses Thema von vier weiteren Formula Student Wettbewerben in Europa aufgegriffen. Im November 2018 kündigte die Formula Student Germany an, ab 2021 die fahrerlose Technologie für alle teilnehmenden Fahrzeuge verpflichtend einzuführen und setzt damit ein klares Zeichen für die Zukunft [15]. Im

¹<https://wiki.ifr.ing.tu-bs.de/carolocup/carolo-cup>

²<https://www.formulastudent.de/>

im Oktober 2019 wurde das Ende der Teilnahmemöglichkeit für Fahrzeuge mit Verbrennungsmotor bei der Formula Student Germany ab 2022 angekündigt[16]. Damit wird die Formula Student Germany ab 2022 zu einem vollständig elektrischen Wettbewerb.

1.1 Wettbewerbsgestaltung

Als Basis für die Arbeit wird ein bereits existierender Formula Student Rennwagen umgebaut, um diesen zu befähigen, an der Formula Student Driverless (FSD) teilzunehmen. Die FSD beinhaltet mehrere Problemstellungen, die über unterschiedlich bewertete Disziplinen dargestellt werden. In den folgenden Unterabschnitten werden die dynamischen Disziplinen, also jene, in denen sich das Fahrzeug autonom bewegen soll, kurz erläutert.

Die Fahrbereiche der Disziplinen sind jeweils mit verschiedenfarbigen Pylonen (*engl. Cone*) gekennzeichnet. Links von der Fahrbahn stehen blaue Pylonen, rechts davon gelbe. Weiterhin gibt es kleine und große orangene Pylonen, die Start- und Stop-Positionen sowie Bremszonen markieren.

1.1.1 Acceleration

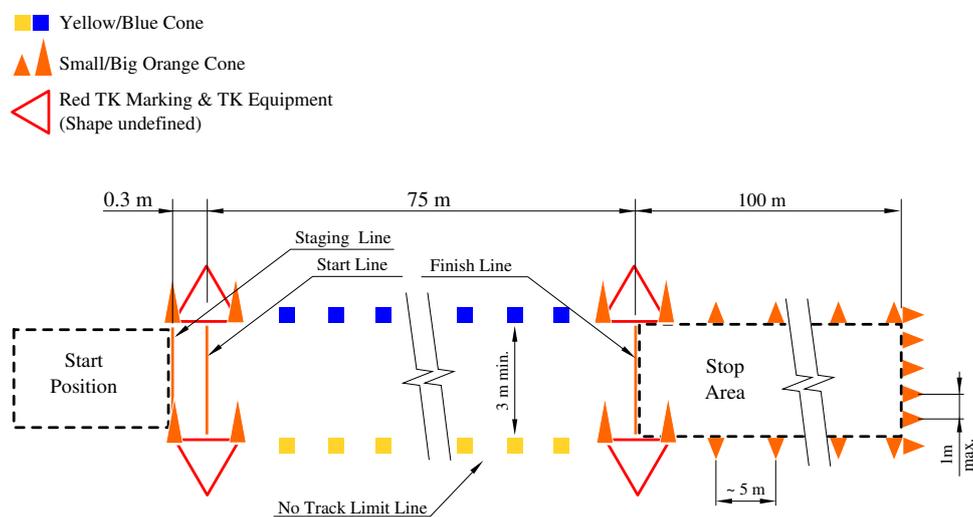


Abbildung 1.1: Streckenverlauf und Markierungen beim Acceleration Quelle: [13, S. 15]

Der **Acceleration** stellt ein klassisches Beschleunigungsrennen über 75 m dar. Der Aufbau der Strecke ist in Abbildung 1.1 dargestellt. Zusätzlich zur eigentlichen Beschleunigung muss das Fahrzeug auch in der Haltezone (*engl.* Stop Area) zum Stehen kommen. Die Farbgebung der Pylonen folgt dem in Abschnitt 1.1 beschriebenen Schema.

1.1.2 Skidpad

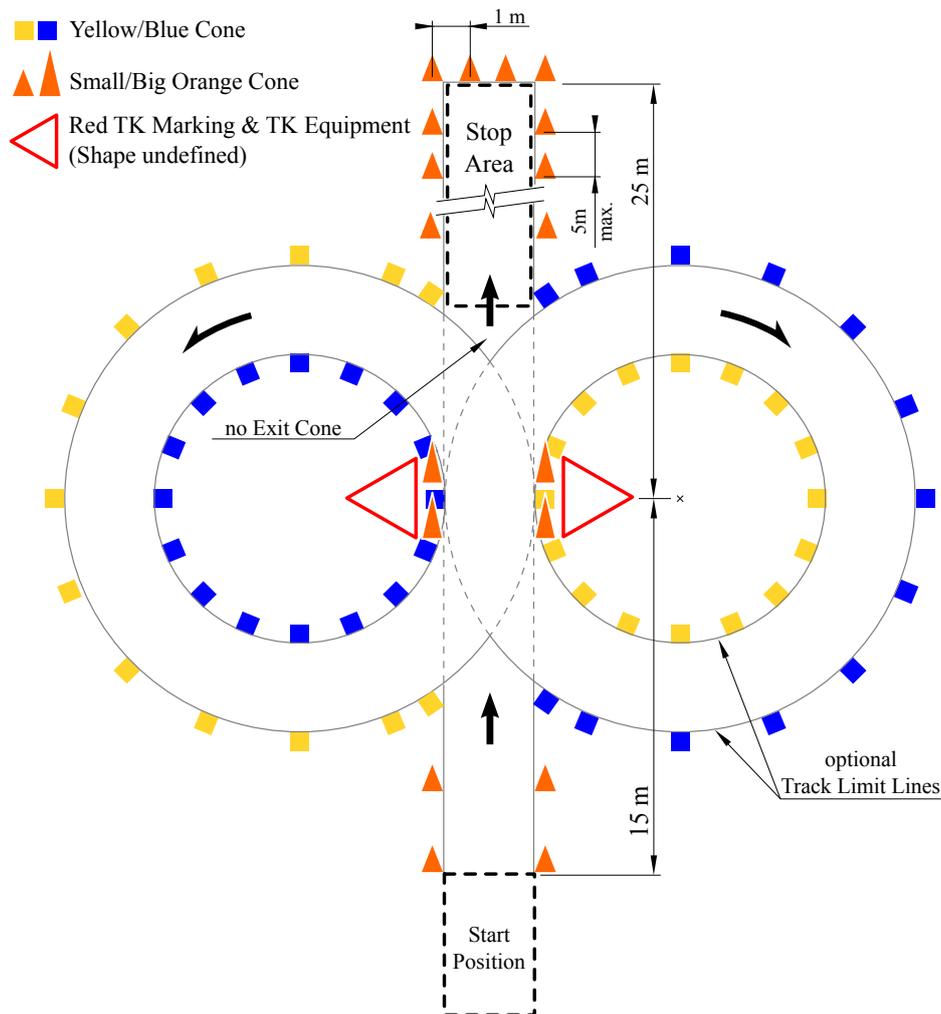


Abbildung 1.2: Streckenverlauf und Markierungen beim Skidpad Quelle: [13, S. 16]

Der **Skidpad** bildet eine liegende Acht, bestehend aus zwei Kreisen. Der Streckenaufbau ist in Abbildung 1.2 visualisiert. Es müssen zwei volle Kreisbahnen in linker Richtung, gefolgt von zwei vollen Kreisbahnen in rechter Richtung durchfahren werden. Auch hier muss das Fahrzeug anschließend in der Haltezone zum Stehen kommen.

1.1.3 Autocross & Trackdrive

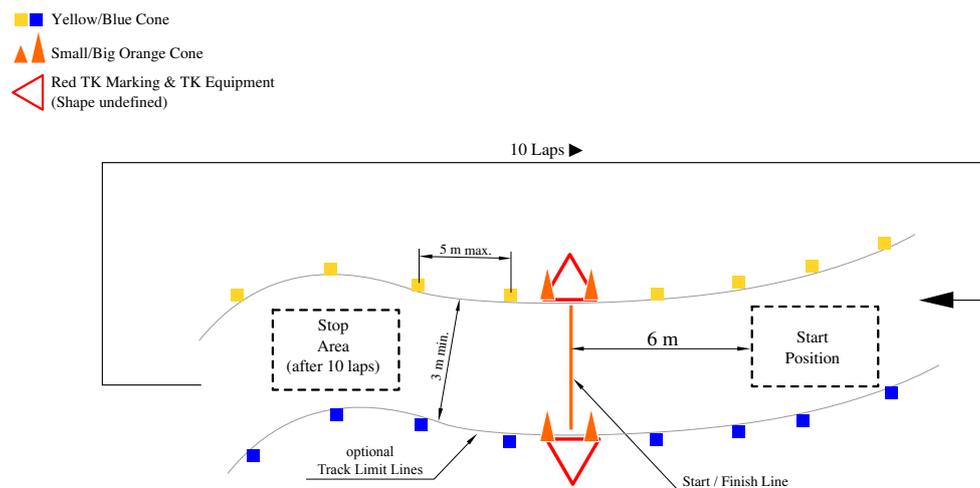


Abbildung 1.3: Streckenverlauf und Markierungen beim Autocross/Trackdrive Quelle: [13, S. 15]

Der **Autocross** und der **Trackdrive** werden auf einer Rundstrecke mit im Reglement[3, S. 129] aufgeführten Randbedingungen absolviert. Während der **Autocross** die Leistungsfähigkeit des Fahrzeugs bei weitestgehend unbekannter Strecke auf einer einzigen Runde bewertet, wird beim **Trackdrive** die Fähigkeit des Fahrzeugs, auch längere Strecken ohne Fehler fahren zu können, über insgesamt 10 Runden geprüft. Im Ziel muss das Fahrzeug in der Haltezone, wie in Abbildung 1.3 dargestellt, zum Stehen kommen.

1.2 Resultierende Aufgabenstellung

Um das übergeordnete Ziel, die erfolgreiche Teilnahme mit einem autonomen Fahrzeug an der FSD, zu erfüllen, ist die Entwicklung eines autonomen Systems notwendig. Das autonome System ist als ein solches zu definieren, dem es möglich ist, die Daten der Sensoren

	Maximale Punkte
Statische Events:	
Business Plan Presentation	75 Punkte
Cost and Manufacturing	100 Punkte
Engineering Design	300 Punkte
Dynamische Events:	
Skid Pad	75 Punkte
Acceleration	75 Punkte
Autocross	100 Punkte
Efficiency	75 Punkte
Trackdrive	200 Punkte
Gesamt	1000 Punkte

Tabelle 1.2: Maximal zu erreichende Punkte bei der Formula Student Driverless Quelle: [3, S. 11]

zu verarbeiten und daraus eine Schätzung der Position und Orientierung des Fahrzeugs sowie eine Karte seiner Umgebung zu erstellen. Weiterhin muss es aus diesen Daten die Fortbewegung des Fahrzeugs planen und entsprechend die Aktoren ansteuern.

In Tabelle 1.2 ist ersichtlich, dass die dynamischen Disziplinen mit 525 zu erreichenden Punkten mehr als die Hälfte der erzielbaren Punkte des Wettbewerbs ausmachen. Die Sensorfusion bildet die entscheidende Basis für die erfolgreiche Teilnahme an diesen Disziplinen. Die Aufgabe dieser Arbeit ist es daher, die für die Sensorfusion notwendigen Komponenten zu beschreiben und dann ein Konzept für die Sensorfusion zu entwickeln. Dabei wird darauf geachtet, das Gesamtsystem des Fahrzeugs zur Verdeutlichung der Komplexität der Fragestellung genau zu betrachten.

1.3 Aufbau der Arbeit

In dieser Arbeit werden zunächst in Kapitel 2 die bereits im Fahrzeug vorhandenen Sensoren und Aktoren beschrieben. Dann wird aufgezeigt, welche Sensoren sowie Aktoren zusätzlich in das Fahrzeug integriert werden müssen, um die Funktion des autonomen Systems zu ermöglichen. In Kapitel 3 wird die allgemeine Systemarchitektur des autonomen Systems erläutert. Es werden die Einzelkomponenten des autonomen Systems beschrieben. Weiterhin werden die physikalischen und logischen Zusammenhänge dieser

Systeme erläutert. Es wird auch kurz auf die grundlegende Softwarearchitektur eingegangen. In Kapitel 4 wird dann der Ansatz für die Sensorfusion beschrieben. Hierzu werden erst allgemein die verwendeten mathematischen Verfahren erläutert. Darauf aufbauend werden die Modelle für die Fusion der Sensordaten des Fahrzeugs gebildet. Abschließend wird in Kapitel 5 das vorgestellte Konzept zusammengefasst und ein Ausblick auf mögliche Erweiterungen und Verbesserungen gegeben.

2 Sensoren & Aktoren

Im Fahrzeug werden eine Vielzahl von Sensoren und Aktoren in verschiedenen Subsystemen verbaut. Diese lassen sich grob in zwei Kategorien aufteilen: Zum einen gibt es Low-Level Sensoren und Aktoren, die für die Grundfahrfunktionen des Fahrzeugs notwendig sind, also nicht erst durch das autonome System notwendig werden. Diese werden in Abschnitt 2.1 beschrieben. Im Gegensatz dazu gibt es die Sensoren und Aktoren, die für die Funktion des autonomen Systems notwendig werden. Die Sensoren des autonomen Systems werden in Abschnitt 2.2 beschrieben. Die dazugehörigen Aktorsysteme werden in Abschnitt 2.3 nur kurz erläutert, da sie für den Kern der Arbeit nicht notwendig sind.

2.1 Basissystem



Abbildung 2.1: Basisfahrzeug HAWKS H14 Quelle: [9]

In diesem Abschnitt werden die Sensoren und Aktoren beschrieben, die bereits in dem in Abbildung 2.1 dargestellten Basisfahrzeug vorhanden sind. Das Basisfahrzeug hat zum jetzigen Zeitpunkt einen Verbrennungsmotor. Daher wird hier auch auf die für ein Fahrzeug mit Verbrennungsmotor zusätzlich notwendigen Sensoren und Aktoren eingegangen. Hierbei wird nur auf die Systeme eingegangen, die auch für die Funktion des autonomen Systems notwendig sind.

2.1.1 Motorsteuerung

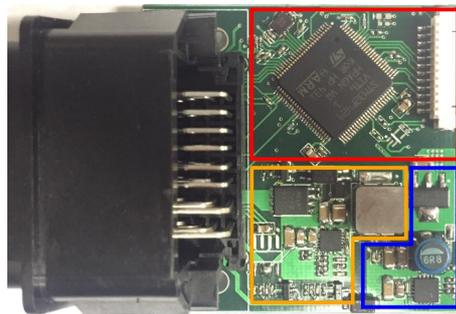
Eine **MS6**[8] der *Bosch Engineering GmbH* wird zur Steuerung des Verbrennungsmotors verwendet. Dazu sind eine Vielzahl von Sensoren und Aktoren notwendig, die hier nicht näher thematisiert werden, da sie nicht direkt relevant für die Entwicklung des autonomen Systems sind.

2.1.2 Raddrehzahlsensoren

Zur Erfassung der Raddrehzahlen werden **DF11** Doppelhallsensoren der Firma *Bosch* verwendet, die über ein an der Radnabe montiertes, ferromagnetisches Inkrementenrad die Raddrehung aufnehmen. Diese werden von der **MS6** erfasst, verarbeitet und auf den Controller Area Network (CAN)-Bus ausgegeben.

2.1.3 Drosselsteuerung

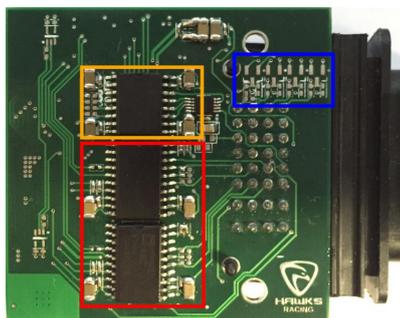
Die Drosselsteuerung besteht aus einem Steuergerät (siehe Abbildung 2.2), das den an der Drosselklappe verbauten Servomotor ansteuert. Dieses Steuergerät nimmt direkt die digitalen Signale der redundanten Drosselklappenpositionssensoren auf. Die Sollwertvorgabe für die Drossel erfolgt entweder (im manuellen Fahrmodus) über einen ebenfalls redundanten Fahrpedalsensor oder vom autonomen System, jeweils über CAN-Bus. Intern nimmt das Steuergerät eine Positionskorrektur mit Hilfe des Positionssensors an der Drossel sowie dem Stromverbrauch des Servomotors vor. Für den manuellen Fahrbetrieb ist die Einstellung verschiedener Übersetzungskennlinien zwischen dem Fahrpedal und der Drossel möglich, um die Fahrbarkeit für den Fahrer zu erhöhen.



STM32F429VI Mikrocontroller

Stromversorgung - Servo

Stromversorgung - Peripherie



Galv. isolierte CAN Transceiver

Isolierter DAC für Drosselklappensignal zur MS6

I/O Filter und Eingangsbeschaltung

Abbildung 2.2: Drosselklappen Steuergerät

2.1.4 Schaltungssteuerung

Für die Ansteuerung der Schaltung wird ebenfalls ein separates Steuergerät verwendet. Dieses steuert direkt einen dreiphasigen, bürstenlosen Motor an, der über ein mehrstufiges Getriebe direkt die Schaltwalze vom Getriebe des Verbrennungsmotors dreht. Ein Positionssensor an der Walze sowie Hall-Encoder an dem bürstenlosen Motor geben eine Rückmeldung über die Ist-Positionen. Die Sollwertvorgabe kommt über den CAN-Bus, entweder über Schaltwippen vom Lenkrad (im manuellen Fahrmodus) oder vom autonomen System.

2.1.5 Kupplungssteuerung

Für die Ansteuerung der Kupplung wird ein Steuergerät verwendet, das elektromechanisch identisch zu dem der Schaltungssteuerung ist. Hier wird ebenfalls ein dreiphasiger, bürstenloser Motor angesteuert, der als Hydraulikpumpe für das Kupplungssystem fungiert. Die Sollwertvorgabe ist analog zur Schaltungssteuerung.

2.2 Sensoren des autonomen Systems

In diesem Abschnitt werden die Sensoren des autonomen Systems beschrieben. Diese dienen der Erfassung der Umgebung des Fahrzeugs sowie der Bestimmung seiner Position. Die Position der Sensoren im Fahrzeug lässt sich aus Abbildung 2.3 entnehmen. Ebenso ist dort die Position der zwei GNSS Antennen dargestellt.

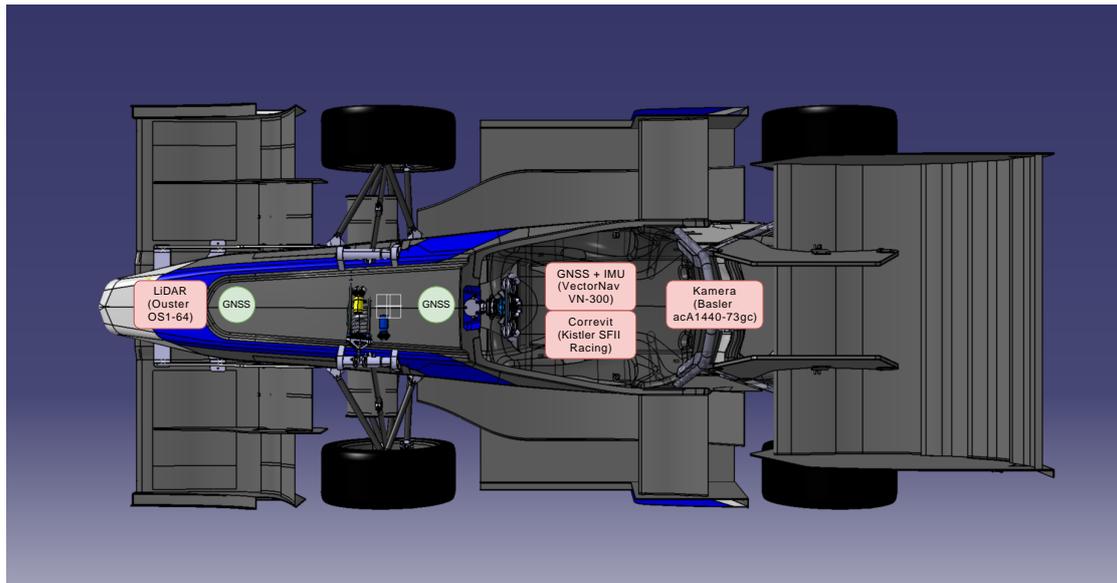


Abbildung 2.3: Sensoren und deren Position am Testfahrzeug

2.2.1 GNSS mit Trägheitsnavigation

Zur Positionsschätzung des Fahrzeugs wird ein Global Navigation Satellite System (GNSS) System mit zwei externen Antennen und integrierter Inertial Measurement Unit (IMU) genutzt. Für das Fahrzeug wurde ein **VN-300** Dual Antenna GNSS/INS[29] von der Firma *VectorNav Technologies, LLC* ausgewählt. Dieses System empfängt an zwei Positionen im Fahrzeug jeweils die Signale der verfügbaren Satelliten der verschiedenen weltweiten Satellitensysteme, derzeit GPS, GLONASS, Galileo und Beidou. Es gibt die durch die beiden integrierten GNSS-Module errechneten Positionen separat aus. Durch die Verwendung von zwei Antennen ist die Errechnung eines Kurses ohne Abhängigkeit von dem magnetischen Nordpol möglich.

Im **VN-300** ist eine 9-achsige IMU integriert. Diese beinhaltet einen Beschleunigungssensor, der in drei Achsen die Beschleunigung misst. Weiterhin ist ein Drehratensensor verbaut, der die Rotation um diese drei Achsen ausgibt. Ebenfalls integriert ist ein dreiachsiges Magnetometer, das den magnetischen Kurs des Fahrzeugs messen kann und als Lagesensor dient. Diese Messwerte werden als Rohdaten für die in Kapitel 4 beschriebene Zustandsschätzung ausgegeben und aufgezeichnet.

Weiterhin führt das **VN-300** intern eine Schätzung der Position, Geschwindigkeit und des Kurses mit einer Frequenz von bis zu 400 Hz durch. Es liefert so korrigierte Positionsdaten mit einer Genauigkeit von bis zu 2,5 m Root Mean Square (RMS) (horizontal) bzw. 2 m RMS (vertikal). Ebenso wird die Geschwindigkeit auf bis zu $\pm 0,05$ m/s und der Kompasskurs mit einer Genauigkeit von bis zu $0,3^\circ$ RMS genau ermittelt.

2.2.2 Optischer Geschwindigkeitssensor

Zur Erhöhung der Genauigkeit der Positionsschätzung wird ein optischer Geschwindigkeitssensor (auch bekannt als *Correvit*) verwendet. Dieser liefert eine absolute Fahrzeuggeschwindigkeit in zwei Achsen (Längs- und Quergeschwindigkeit), eine daraus zusammengeführte Betragsgeschwindigkeit sowie den Schräglaufwinkel des Fahrzeugs. Im Gegensatz zu Raddrehzahlsensoren ermöglicht dieser Sensor eine schlupffreie Messung der absoluten Fahrzeuggeschwindigkeit. Schlupf ist das (meist ungewollte) Durchdrehen oder Blockieren eines oder mehrerer Reifen. Bei Schlupf ist die Fahrzeuggeschwindigkeit nicht mehr korrekt über die in Unterabschnitt 2.1.2 beschriebenen Raddrehzahlsensoren ermittelbar. Diese Form der Messung ist eine der wenigen Möglichkeiten, eine absolute Fahrzeuggeschwindigkeit festzustellen, sowie Schlupf mit einer hohen Konfidenz zu messen.

Für das Fahrzeug wird ein **SFII**[19] der Firma *Kistler Instrumente GmbH* genutzt. Dieser Sensor liefert die Geschwindigkeiten mit einer Genauigkeit von unter 0,5% vom Gesamtmessbereich von $0,3$ km/h bis 250 km/h bei einer Frequenz von bis zu 250 Hz. Der Schräglaufwinkel wird mit einer Genauigkeit von unter $0,5^\circ$ bei einer Auflösung von $0,1^\circ$ gemessen.

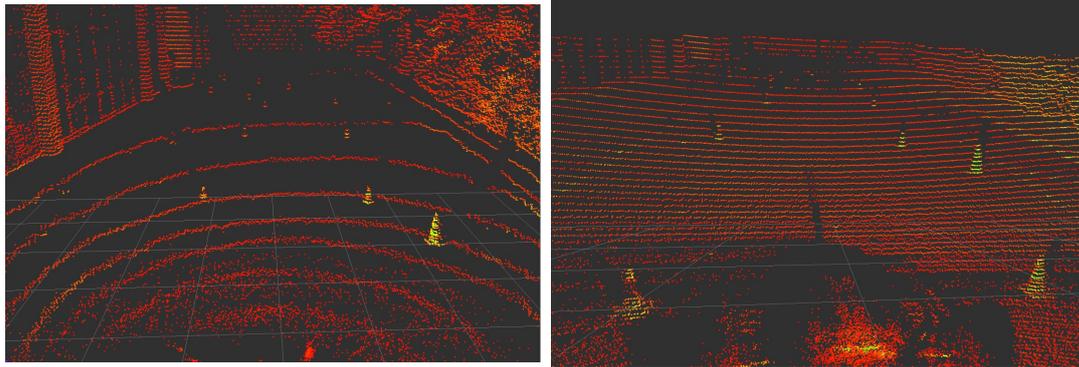
2.2.3 LiDAR

Ein Light Detection and Ranging (LiDAR) ist ein Sensor, der die Entfernung zu Objekten durch Aussendung von Laserimpulsen und Auffangen von deren Reflektion misst (*time-of-flight*-Prinzip). Dabei wird grundsätzlich zwischen 2D- und 3D- LiDAR unterschieden. 2D-LiDAR messen nur in einer horizontalen Schicht, während 3D-LiDAR zeitgleich mehrere horizontale Schichten abtasten. Zum Zeitpunkt dieser Arbeit sind größtenteils LiDAR mit einem rotierenden Sensorkopf verfügbar. Solid-state LiDAR, die ohne rotierende Komponenten auskommen und damit die Fehleranfälligkeit und den Preis signifikant reduzieren sollen, wurden schon von zahlreichen Firmen angekündigt [30, 24].

Als LiDAR für das Fahrzeug wurde ein OS1-64 [23] der Firma *Ouster, Inc.* ausgewählt. Mit einer Auflösung von $0,53^\circ$ in 64 Schichten in der Horizontalen, und bis zu $0,18^\circ$ in 2048 Schritten um die z-Achse, ist dieser gut geeignet zur Erkennung von kleinen, weit entfernten Objekten. Er hat ein Sichtfeld von $-16,6^\circ$ bis $16,6^\circ$ in der Horizontalen und ein Sichtfeld von 360° um die z-Achse. Die Rotationsgeschwindigkeit ist auswählbar zwischen 10 Hz und 20 Hz (bei geringerer Auflösung um die z-Achse). Ebenfalls enthält der OS1-64 eine 9-achsige IMU zur Korrektur der erfassten Punktwolken mittels Trägheitsnavigation.



Abbildung 2.4: Position des LiDAR auf dem Frontflügel des Fahrzeugs



(a) Sensorposition auf dem Frontflügel

(b) Sensorposition auf dem Überrollbügel

Abbildung 2.5: Vergleich der verschiedenen Positionen des LiDAR am Fahrzeug

Der LiDAR wurde, wie in Abbildung 2.4 ersichtlich, auf dem Frontflügel des Fahrzeugs angebracht. Diese Position eignet sich gut, weil so sichergestellt wird, dass zumindest eine horizontale Schicht des LiDAR parallel zum Boden ist. Somit kann die Erfassungsrate auch auf größere Distanzen maximiert werden. Weiterhin wird im näheren Distanzbereich so auch bei größeren Nickbewegungen eine gleichmäßige Verteilung der horizontalen Schichten über die Pylonen erreicht.

In Abbildung 2.5a lässt sich bei Positionierung des LiDAR auf dem Frontflügel die durch Pylonen gekennzeichnete Kurve noch gut erkennen. Dahingegen ist die Kurve bei Positionierung des LiDAR auf dem Überrollbügel, wie in Abbildung 2.5b dargestellt, nicht mehr gut zu erkennen.

2.2.4 Kamera

Für das Fahrzeug wurde eine **acA1440-73gc**[6] der Firma *Basler AG* als Kamera ausgewählt. Diese erreicht eine Bildwiederholrate von bis zu 73 Bildern pro Sekunde bei einer Auflösung von 1440x1080 Pixel. Sie unterstützt sowohl die Aufzeichnung von monochromen als auch die von farbigen Bildern. Die Kamera wird über Gigabit Ethernet angebunden und über Power over Ethernet (PoE) nach IEEE 802.3af[5] mit Strom versorgt. In dieser Arbeit wird auf die Kamera und das dazugehörige Aufzeichnungssystem nicht näher eingegangen.

2.3 Aktoren des autonomen Systems

Im Folgenden werden die notwendigen Aktoren und deren wichtigste Eigenschaften erläutert. Die Aktoren dienen der Umsetzung der vom autonomen System kommenden Steuerbefehle auf die Bewegung des Fahrzeugs. Eine besondere Anforderung an die Entwicklung der Aktoren ist, dass es weiterhin möglich sein muss, das Fahrzeug manuell zu fahren, ohne dafür Umbauten vorzunehmen.

2.3.1 Bremsaktor und EBS

Im Fahrzeug wird ein auf das normale Bremssystem wirkender Aktor verbaut. Dieser muss die Funktionen des vom Reglement geforderten Notbremssystems - *Emergency Brake System (EBS)* - erfüllen. Außerdem muss er im autonomen Fahrbetrieb Möglichkeiten bieten, das Fahrzeug mit variablen Verzögerungswerten zu bremsen. Ein mögliches Konzept für das Bremssystem eines autonomen Formula Student Rennwagens wird in [11] beschrieben.

2.3.2 Lenkungsaktor

Zur Lenkung des Fahrzeugs wird ein in [10] entwickelter Aktor auf Höhe der Vorderachse des Fahrzeugs unter dem Rahmen verbaut. Der Lenkungsaktor dreht über die Spurstange die Vorderräder des Fahrzeugs und ermöglicht so die Steuerung des autonomen Fahrzeugs.

3 Systemarchitektur

In diesem Kapitel werden die Grundlagen für den Aufbau eines autonomen Systems beschrieben und damit die Basis für die in Kapitel 4 beschriebene Sensorfusion geschaffen.

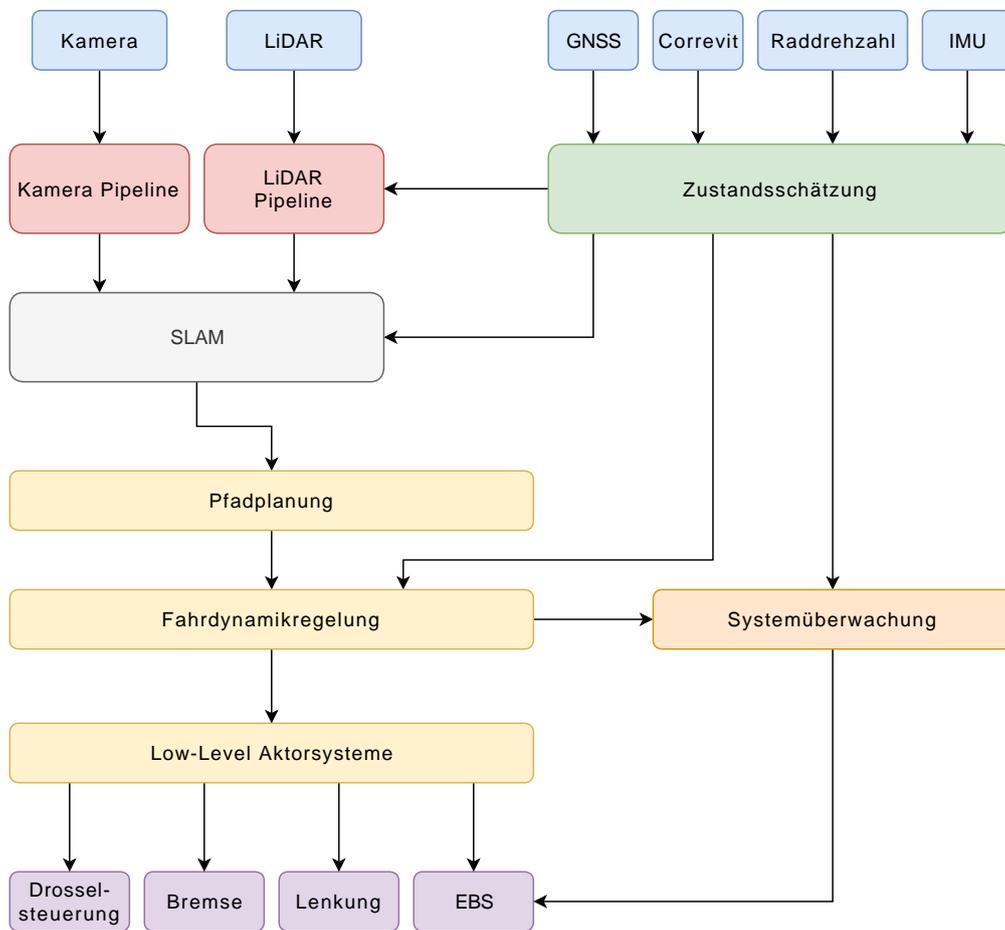


Abbildung 3.1: Architektur des autonomen Systems

In Abbildung 3.1 sind die verschiedenen Komponenten des autonomen Systems dargestellt. Die Abbildung kann als Datenfluss von den oben dargestellten Sensoren bis zu den unten dargestellten Aktoren verstanden werden.

In Abschnitt 3.1 wird zuerst die Vernetzung der Sensoren und Aktoren mittels verschiedener Bussysteme erläutert. Danach wird in Abschnitt 3.2 die Synchronisation der Zeitbasen der Komponenten erläutert. In Abschnitt 3.3 werden die verschiedenen, bei der Fusion der Sensordaten zu erwartenden Koordinatensysteme verdeutlicht. Abschnitt 3.4 beschreibt die Referenzierung der Sensoren aufeinander. Als nächstes wird in Abschnitt 3.5 die Basis für die Entwicklung der Software erörtert. Schließlich wird in Abschnitt 3.6 ein Ansatz für die Pipeline zur Verarbeitung der LiDAR-Punktwolken gebildet.

3.1 Fahrzeugbussysteme

In Abbildung 3.2 ist eine Übersicht der Kommunikationsverbindungen der verschiedenen Sensoren und Aktoren dargestellt. Im Fahrzeug sind insgesamt zwei CAN 2.0B-Busse mit einer Baudrate von jeweils 1 Mbit/s verbaut. Der CAN-Bus ist ein überwiegend im Automobilbereich genutztes differenzielles Feldbussystem. Dieses zeichnet sich unter anderem durch seine hohe Toleranz gegenüber externen elektronischen Störeinflüssen (EMI) aus. Er wird dank seiner großen Ausfallsicherheit oftmals zur Anbindung von sicherheitskritischen Systemen wie zum Beispiel einem Antiblockiersystem (ABS) eingesetzt. Über diese CAN-Busse ist der Großteil der Fahrzeugsensorik untereinander verbunden. Weiterhin wird im Fahrzeug für die Daten der Sensoren des autonomen Systems Gigabit-Ethernet genutzt. Zwischen den CAN-Bussen und dem Ethernet-Switch ist ein CAN-Ethernet-Gateway angebunden. Dieses kapselt die auf den CAN-Bussen eintreffenden Nachrichten in Ethernet-Frames und spielt in Ethernet Frames gekapselte CAN-Nachrichten auf die CAN-Busse ein.

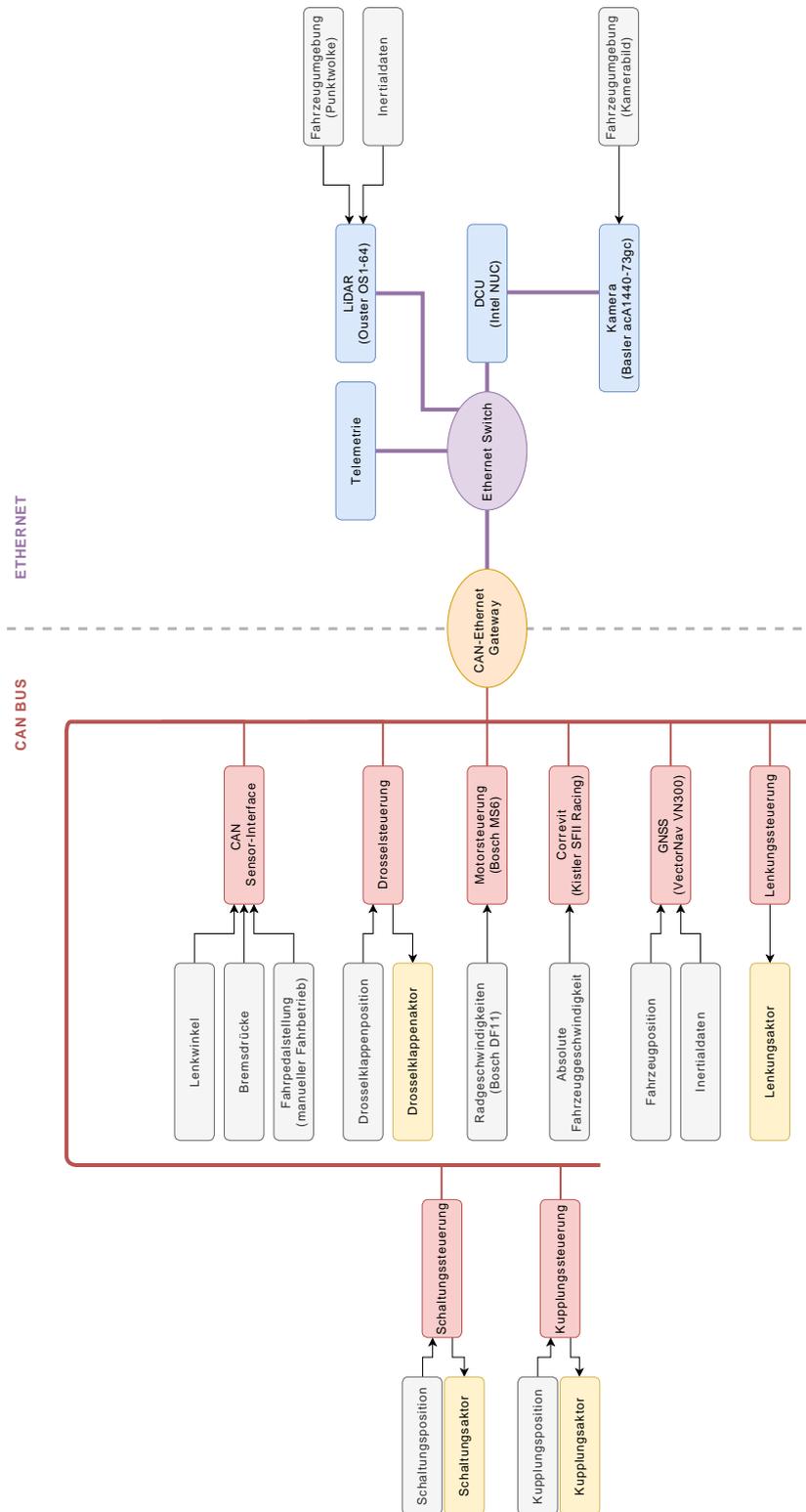


Abbildung 3.2: Bussysteme im Fahrzeug

Messdaten (Sensor)	Messfrequenz
Absolute Fahrzeuggeschwindigkeit (Correvit)	250 Hz
Positionsdaten (GNSS)	400 Hz
Inertialdaten (GNSS)	800 Hz
Punktwolke (LiDAR)	10 Hz
Inertialdaten (LiDAR)	100 Hz
Radgeschwindigkeiten	100 Hz
Lenkwinkel	100 Hz
Bremsdrücke	100 Hz

Tabelle 3.2: Sensoren und Messfrequenzen

In Tabelle 3.2 ist ersichtlich, dass die Messwerte aller Sensoren in den Messfrequenzen stark voneinander abweichen. Eine Reduktion der einzelnen Sensorfrequenzen auf eine gemeinsame Frequenz ist nicht als sinnvoll anzusehen, da dies zu einer Verringerung der Aktualisierungsfrequenz der Sensorfusion und somit zu einer Reduktion der Reaktionsgeschwindigkeit des autonomen Systems führen würde. Somit stellt dies für die Systemarchitektur die Anforderung, diese Sensorwerte robust zu einer gemeinsamen Schätzung zu fusionieren. Dies wird in Kapitel 4 betrachtet.

3.2 Zeitsynchronisation

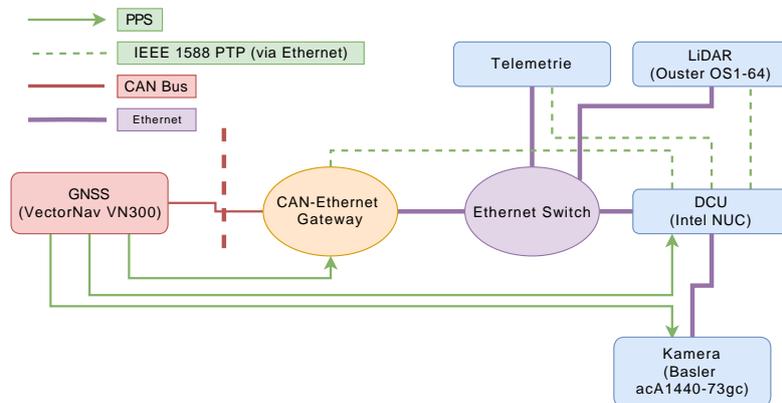


Abbildung 3.3: Überblick über die Zeitsynchronisation im Fahrzeug

Aufgrund der Komplexität des Gesamtsystems mit vielen verschiedenen Sensoren, Aktoren und Steuereinheiten, die über unterschiedliche Kommunikationsschnittstellen mitein-

ander verbunden sind, ist eine präzise Zeitsynchronisation notwendig. Für die Synchronisation der Messdaten wird eine Kombination aus verschiedenen Konzepten eingesetzt, welche im Folgenden beschrieben wird und in Abbildung 3.3 dargestellt ist.

3.2.1 Globale Zeitbasis

Die globale Zeitbasis kommt vom in Unterabschnitt 2.2.1 beschriebenen GNSS. Die globale Zeit wird einmal pro Sekunde auf dem CAN-Bus verteilt und kommt so auch über das CAN-Ethernet-Interface zur Driverless Compute Unit (DCU). Weiterhin kommt vom GNSS ein über dedizierte Signalleitungen verteiltes Pulse-Per-Second (PPS) Signal, das an das CAN-Ethernet-Gateway, die DCU und die Kamera angeschlossen wird. Mithilfe des über CAN und daher auch Ethernet verfügbaren Zeitstempels vom GNSS und dem PPS kann die DCU nun eine hochgenaue Systemzeit feststellen. Diese Zeit wird dann mit dem IEEE 1588 Precision Time Protocol (PTP) [4] im Netzwerk synchronisiert, wobei die DCU als Clock Master fungiert.

3.2.2 CAN-Bus und Echtzeitmesswerte

Ein Großteil der Messwerte wird auf einem der CAN-Busse des Fahrzeugs übertragen. Der CAN-Bus ist generell nur mit Einschränkungen als echtzeitfähig anzusehen. Es kann jedoch für Gruppen von wichtigen Nachrichten durch Priorisierung eine weiche Echtzeitfähigkeit erreicht werden, die für den hier gegebenen Anwendungszweck ausreichend ist [27]. Generell setzt dieses Verfahren eine Buslast im Bereich von weniger als zwei Dritteln der maximal möglichen Last voraus, sonst kommt es vermehrt zu Kollisionen. Für das konkrete System können wir annehmen, dass alle relevanten Daten mit einer fixen Frequenz gemessen und auf den Bus gesendet werden. Auf dem CAN-Ethernet Interface werden die eintreffenden Nachrichten mit globalen Zeitstempeln versehen, gruppiert und in Ethernet Frames verpackt.

3.3 Koordinatensysteme

Für die Systemarchitektur muss auf die verschiedenen Koordinatensysteme Rücksicht genommen werden. Diese ergeben sich aus der Konstruktion des Fahrzeugs, der Eigenschaften der Sensoren oder als globale Referenz. So ist zum Beispiel die Punktwolke

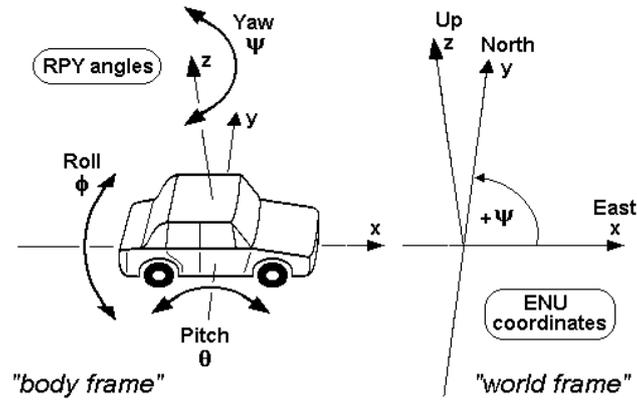


Abbildung 3.4: Verschiedene Koordinatensysteme in einem Fahrzeug Quelle: [25]

des LiDAR initial in einem lokalen Koordinatensystem desselben orientiert und muss im Laufe der Verarbeitung in das globale Koordinatensystem transformiert werden. In Abbildung 3.4 sind das in Unterabschnitt 3.3.1 beschriebene globale Koordinatensystem (*engl.* world frame) und das in Unterabschnitt 3.3.2 beschriebene Fahrzeugkoordinatensystem (*engl.* body frame) dargestellt.

3.3.1 Globales Koordinatensystem

Das globale Koordinatensystem dient als Referenz für Position und Orientierung des Fahrzeugs. Es wird als rechtshändiges Koordinatensystem definiert. Zur Vereinfachung der Berechnung wird hier ein Earth, North, Up (ENU) Koordinatensystem gewählt, das eine Ebene als Tangente der Erdoberfläche in einem Referenzpunkt \mathbf{p}_{Ref} bildet. Die Ebene ist definiert als Schnittpunkt der Tangente der in Ost-West-Richtung verlaufenden Kreislinien (Breitenkreise) und der Tangente der in Nord-Süd-Richtung verlaufenden Kreislinien (Längenkreise). Die x-Achse zeigt nach Osten, die y-Achse nach Norden und die z-Achse zeigt nach oben.

$$\mathbf{p}_{ENU} = \mathbf{R}(\mathbf{p}_{ECEF} - \mathbf{p}_{Ref}) \quad (3.1)$$

$$\mathbf{p}_{ECEF} = \mathbf{R}^T \mathbf{p}_{ENU} + \mathbf{p}_{Ref} \quad (3.2)$$

$$\mathbf{R} = \begin{bmatrix} -\sin(\lambda) & \cos(\lambda) & 0 \\ -\sin(\phi) \cos(\lambda) & -\sin(\phi) \sin(\lambda) & \cos(\phi) \\ \cos(\phi) \cos(\lambda) & \cos(\phi) \sin(\lambda) & \sin(\phi) \end{bmatrix} \quad (3.3)$$

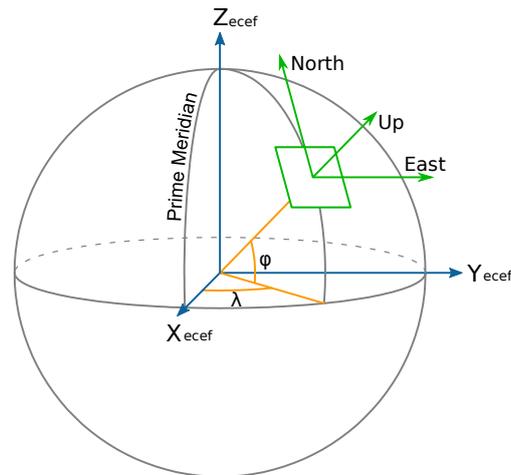


Abbildung 3.5: Darstellung der Relation zwischen dem ECEF und ENU System Quelle: [21]

Die vom GNSS gelieferten Positionsdaten müssen erst aus dem Earth-centered, Earth-fixed (ECEF)-System, das auf Längengraden (λ) und Breitengraden (ϕ) basiert, in das ENU-System transformiert werden. Die Relation zwischen dem ENU-System und dem ECEF-System wird in Abbildung 3.5 dargestellt. Dafür wird die Gleichung 3.1 mit der in Gleichung 3.3 definierten Rotationsmatrix \mathbf{R} angewendet. Zur Rücktransformation kann die Gleichung 3.2 mit der transponierten Rotationsmatrix \mathbf{R}^T verwendet werden.

Beim Start des autonomen Systems wird das ENU-System auf die Position $[0, 0, 0]^T$ initialisiert. Hierin spiegelt sich wider, dass das ENU-System lokal zur Fahrzeugposition ist. Die Referenz für dieses lokale System zum ECEF-System, in dem die Positionsangaben des GNSS vorliegen, wird als \mathbf{p}_{Ref} initialisiert.

Durch die Rahmenbedingungen des Wettbewerbs lässt sich annehmen, dass der Boden weitestgehend eben ist und keine Schiefelage der Bodenebene zu erwarten ist. Mit diesen Annahmen lässt sich durch Reduktion des Koordinatensystems auf zwei Dimensionen die Komplexität der Berechnungen erheblich verringern.

3.3.2 Fahrzeugkoordinatensystem

Das Fahrzeugkoordinatensystem ist ein körperfestes Koordinatensystem, das als Basis für das autonome System genutzt wird. Es ist ein rechtshändiges Koordinatensystem und definiert wie in Abbildung 3.4 dargestellt die x-Achse entlang der Fahrzeuglängsachse, die

y-Achse in Fahrtrichtung links und die z-Achse nach oben. Weiterhin werden in Abbildung 3.4 die drei Euler-Winkel dargestellt. Dies sind der Gierwinkel (*engl.* Yaw angle), der Nickwinkel (*engl.* Pitch angle) und der Rollwinkel (*engl.* Roll angle). Zur Vereinfachung der Berechnung im Zustandsmodell wird es so positioniert, dass sein Ursprung und seine Rotation mit der der IMU des **VN-300** übereinstimmt.

3.3.3 Konstruktionskoordinatensystem

Das Fahrzeug hat ein in der Konstruktion definiertes, lokales rechtshändiges Koordinatensystem. Der Ursprung dieses Systems liegt in der Mitte der Vorderachse, mit der x-Achse nach hinten und z-Achse nach oben. Dieses System wird zur Definition der Position der Sensoren verwendet. Es wird hierfür ein Computer Aided Design (CAD) System zur Hilfe genommen.

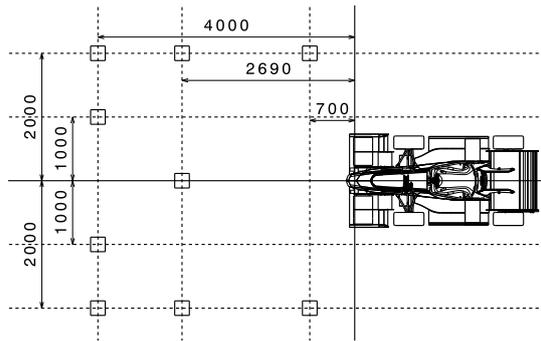
3.3.4 Sensorkoordinatensysteme

Weiterhin haben die Umgebungssensoren wie LiDAR oder Kamera eigene Koordinatensysteme, welche relativ zum Ursprung der jeweiligen Sensoren sind. Diese sind durch Rotation und Translation in Relation zum Fahrzeugkoordinatensystem positioniert.

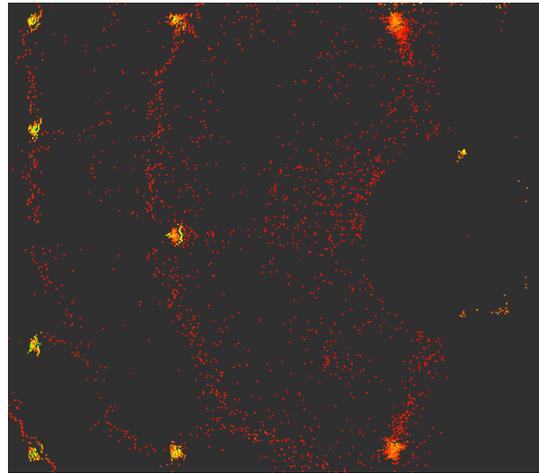
3.4 Sensorreferenzierung

Für eine saubere Fusion der Sensordaten ist eine Referenzierung der Sensoren aufeinander notwendig. Dies ist insbesondere für den LiDAR und die Kamera notwendig, da diese einen weiten Bereich der Umgebung des Fahrzeugs erfassen. Somit können schon geringe Abweichungen hohe Fehler in der Navigation zur Folge haben. Als Basis für die Sensorreferenzierung wird die aus dem CAD bekannte Position der Sensoren angenommen.

In Abbildung 3.6a wird die für die Sensorreferenzierung geplante Position der Pylonen im Sensorerfassungsbereich des Fahrzeugs für die Referenzierung dargestellt. Die Position der Pylonen wird vom Ursprung des LiDAR aus festgelegt, da von diesem aus einfach gemessen werden kann. Es wird ein Kreuzlinienlaser genutzt, um eine Linie durch den Ursprung des LiDAR und die Mittelachse des Fahrzeugs zu projizieren. Danach werden die einzelnen Referenzlinien mit einem Laser-Entfernungsmesser ermittelt und auf dem



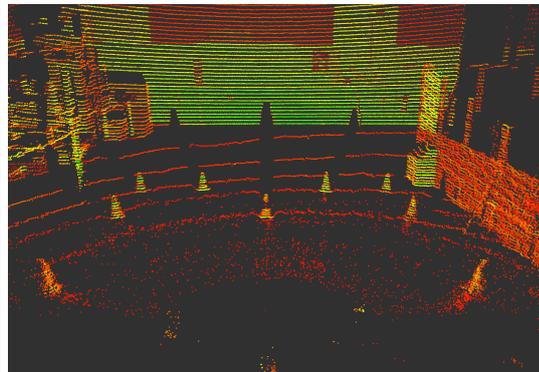
(a) Zeichnung der Pylonenposition in Relation zum LiDAR



(b) Draufsicht auf die LiDAR Punktwolke



(c) Kamerabild



(d) LiDAR Punktwolke in Kameraperspektive

Abbildung 3.6: Messaufbau für die Sensorreferenzierung

Boden angezeichnet. Schließlich werden mit Hilfe dieser Linien die Pylonen positioniert. In Abbildung 3.6b ist die LiDAR Punktwolke des Messaufbaus in einer Draufsicht vergleichbar mit Abbildung 3.6a dargestellt. In Abbildung 3.6c wird das zum Messaufbau gehörige Bild der Kamera gezeigt. Abbildung 3.6d zeigt die LiDAR Punktwolke in der Perspektive der Kamera.

Mit diesem Messaufbau lässt sich die Orientierung des Kamerabildes relativ zur LiDAR Position ermitteln. Weiterhin lässt sich für den LiDAR durch die mit dem Kreuzlinienlaser geschaffene Referenzachse ein Winkelfehler in der z-Achse feststellen.

3.5 Softwarearchitektur

Ausgehend von der in Abbildung 3.1 dargestellten Systemarchitektur bietet sich für die Softwarearchitektur eine Aufteilung der Komponenten in lose, eigenständige Module an. Diese Module können dann von verschiedenen Entwicklern parallel umgesetzt werden. Für diese Architektur wird ein Grundgerüst gewählt, das Möglichkeiten zur asynchronen Verteilung von Nachrichten bereitstellt. Weiterhin sollte das Grundgerüst die Möglichkeit bieten, den Status und die Messwerte des Systems in geordneter Form aufzuzeichnen.

3.5.1 ROS

Das Robot Operating System (ROS)[26] wird als Basis für die Entwicklung genommen, da es die oben genannten Anforderungen erfüllt. ROS wurde ursprünglich für Anwendungszwecke in der Robotik entwickelt, aus der sich viele Aufgabenstellungen auch auf das autonome Fahren übertragen lassen. ROS bietet nicht nur Schnittstellen für eine Vielzahl an Sensoren und Aktoren, es stellt auch Bibliotheken für die Verarbeitung von Sensordaten und die Bewegungsplanung bereit. Es ist open-source und wird von einer aktiven Community kontinuierlich weiterentwickelt.

3.5.2 Entwicklungsprozess

Für die Verwaltung des Source Codes des Projekts wird Gitlab¹ benutzt. Weiterhin werden die vielfältigen Möglichkeiten der Gitlab Continuous Integration (CI) genutzt,

¹<https://about.gitlab.com/>

um unter anderem Tests automatisiert auszuführen und Docker Images zu bauen.

Um das komplexe Softwaresystem mit guten automatisierten Testmöglichkeiten auszustatten und identische Softwarestände in der Test- und Fahrzeugumgebung zu gewährleisten, wird ein starker Fokus auf die Verwendung von Docker² Containern gesetzt. Die in Abbildung 3.7 dargestellte CI-Pipeline des Hauptprojekts spiegelt die Umsetzung dieser Ziele wider: Der Code wird zuerst gebaut und statischen Tests (Formatierung, Zeichensatz) unterzogen. Im nächsten Schritt wird die Software im Testcontainer installiert. Es folgen statische Analyse des kompilierten Codes sowie Unit-Tests mit Coverage Reports. Als letzter Schritt wird der getestete Code in ein Docker Image eingefroren, das später auch direkt im Fahrzeug verwendet werden kann.

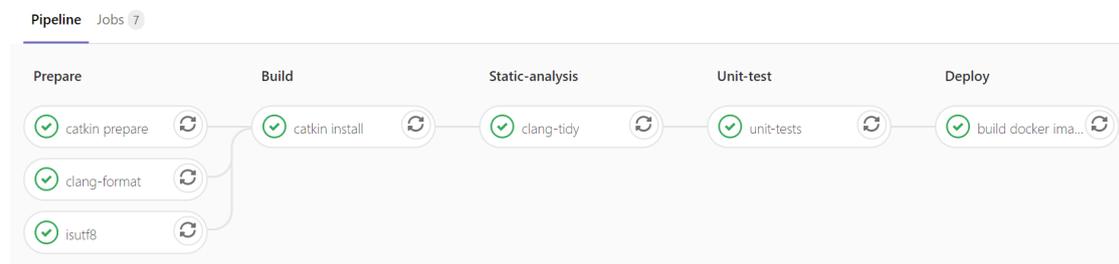


Abbildung 3.7: CI-Pipeline des Hauptprojekts

Durch diese Herangehensweise wird sichergestellt, dass nur getesteter Code im Fahrzeug eingesetzt wird. Dies ist wichtig, um die ohnehin knappe Testzeit am Fahrzeug effektiv zu nutzen. Weiterhin wird durch den Build-Prozess eine kontinuierliche Versionierung der gesamten Laufzeitumgebung gewährleistet. Dies ist insbesondere wichtig, um Probleme schnell zu diagnostizieren.

Zur Koordination der Aufgaben der Entwickler und zur Überwachung des Entwicklungsfortschritts wird Atlassian Jira³ eingesetzt. Dies ermöglicht den Einsatz von agilen Methoden für die Softwareentwicklung, was gerade in einem kleinen Team mit stets wechselnden Mitgliedern wichtig ist.

Zur Dokumentation des Entwicklungsprozesses wird Atlassian Confluence⁴ eingesetzt.

²<https://docker.com>

³<https://www.atlassian.com/de/software/jira>

⁴<https://www.atlassian.com/de/software/confluence>

3.5.3 Laufzeitumgebung im Fahrzeug

Die Laufzeitumgebung für Softwarekomponenten des autonomen Systems wird durch die Driverless Compute Unit (DCU), das zentrale Rechensystem des Fahrzeugs, bereitgestellt. Die DCU dient auch der dauerhaften Aufzeichnung aller Sensordaten und des Systemstatus, um die nachträgliche Analyse des Verhaltens des autonomen Systems möglich zu machen. Hierzu wird ein Intel **NUC8I7BEH**⁵ Mini-PC eingesetzt. Dieser ist ausgestattet mit einem Vierkernprozessor und 32 GB Arbeitsspeicher. Als Speichermedium wird eine 1 TB große Solid State Disk (SSD) benutzt. Es wird die Linux-Distribution Ubuntu in der Version 16.04 als Betriebssystem verwendet. Auf dem Betriebssystem wird eine Docker Laufzeitumgebung verwendet, um die durch den Build-Prozess erzeugten Container auszuführen. Die DCU wird über die in Abschnitt 3.1 beschriebenen Bussysteme mit den Sensoren und Aktoren des Fahrzeugs vernetzt.

3.6 LiDAR Pipeline

Die LiDAR-Pipeline beschreibt die Verarbeitung der Rohdaten (Punktwolken im LiDAR Sensorkoordinatensystem) zu verwertbaren Informationen für den SLAM-Algorithmus. In diesem Abschnitt wird ein mögliches Konzept für die LiDAR Pipeline skizziert. Hierzu ist zuerst eine Korrektur der Rohdaten mit der zum Zeitpunkt der Erfassung der Punkte gemessenen Fahrzeugposition und Orientierung notwendig. Danach werden jene Punkte aus der Punktwolke entfernt, die auf der Fahrbahn liegen und damit nicht von Interesse für die weitere Verarbeitung sind. Schließlich werden in der reduzierten Punktwolke die Pylonen und deren jeweilige Position erkannt.

3.6.1 Positionskorrektur

Die dauerhafte Bewegung des Fahrzeugs während der Messdatenerfassung in Kombination mit dem in Unterabschnitt 2.2.3 beschriebenen Funktionsprinzip des LiDAR macht eine Korrektur der Rohdaten mit der während der Erfassung der jeweiligen Punkte erfolgten Rotation und Translation des Fahrzeugs notwendig. Angenommen, das Fahrzeug bewegt sich mit einer Geschwindigkeit von 20 m/s, so bewegt sich der LiDAR mit dem Fahrzeug bei seiner normalen Rotationsfrequenz von 10 Hz pro Umdrehung um 2 m. Dies

⁵<https://www.intel.de/content/www/de/de/products/boards-kits/nuc/kits/nuc8i7beh.html>

würde unkorrigiert eine erhebliche Störung der Daten hervorrufen. Daher werden die Rohdaten im ersten Schritt über die in Kapitel 4 beschriebene Zustandsschätzung korrigiert.

3.6.2 Bodenentfernung

Im nächsten Schritt muss der Boden aus der Punktwolke entfernt werden. Dieser enthält keine notwendigen Informationen für das System und würde somit nur den Aufwand für die Verarbeitung der Sensordaten erhöhen. Hierzu wird in Anlehnung an [18, 28] die Punktwolke in radiale Sektionen unterteilt, die wiederum in Segmente mit definiertem Abstand zum Sensor unterteilt werden. Innerhalb dieser Segmente wird dann nach dem tiefsten Punkt gesucht. Alle Punkte, die tiefer als dieser Punkt plus ein Offset liegen werden als Bodenpunkt markiert.

3.6.3 Pylonenerkennung

Im nächsten Schritt müssen in der bearbeiteten Punktwolke die Pylonen erkannt werden. Hierzu kann der Ansatz aus [18] verwendet. Darin wird geschildert wie zuerst aus einer LiDAR Punktwolke mit Hilfe eines euklidischen Clustering- Algorithmus die grobe Position von möglichen Pylonen erkannt wird. Im Anschluss daran werden die im vorherigen Schritt fälschlich entfernten Bodenpunkte in der Umgebung dieser Clusters wiederhergestellt. Schließlich werden die einzelnen Cluster mit einem mathematischen Modell auf Ähnlichkeit zu dem Modell einer Pylone untersucht. Weiterhin ist es auf eine begrenzte Entfernung möglich, die Farbe der Pylonen durch Unterscheidung der Intensität der Punkte auf den verschiedenen Ringen zu erkennen.

3.7 SLAM

Das aus der Robotik bekannte Simultaneous Localization and Mapping (SLAM) Problem beschreibt für ein autonomes System die Notwendigkeit, während seiner kontinuierlichen Fortbewegung eine Karte seiner Umgebung zu erstellen und sich innerhalb dieser zu lokalisieren. Hier kann ein SLAM-Algorithmus, wie zum Beispiel in [22, 1] beschrieben, für die Kombination der Daten aus der in Abschnitt 3.6 skizzierten LiDAR Pipeline

mit den Zustandsschätzungen aus der in Kapitel 4 geschilderten Sensorfusion genutzt werden.

4 Zustandsschätzung

Die Basis für ein solides autonomes System ist eine präzise Zustandsschätzung für das Fahrzeug. Für die Erfassung und Verarbeitung der Sensordaten, die Lokalisation des Fahrzeugs und für die Erstellung der Umgebungskarte ist eine zuverlässige Schätzung der Fahrzeuggeschwindigkeit, Fahrzeugposition und des Fahrzeugkurses notwendig.

Es wird zunächst in Abschnitt 4.1 das Kalman Filter für lineare Systeme als Basis erläutert. In Abschnitt 4.2 wird dann das Extended Kalman Filter (EKF) als Erweiterung auf nichtlineare Systeme beschrieben. Schließlich wird in Abschnitt 4.3 die konkrete Anwendung des EKF auf die Zustandsschätzung des Fahrzeugs beschrieben. Hierzu werden die Daten der verschiedenen Sensoren analog zu den in [20, 17, 7, 28, 12] beschriebenen Modellen fusioniert.

4.1 Kalman Filter für lineare Systeme

Das Kalman Filter ist ein mathematisches Verfahren mit dem Ziel, Fehler in Messwerten zu reduzieren und Schätzungen für nicht direkt beobachtbare Messgrößen zu liefern. Hierzu werden potenziell verrauschte, in begrenztem Maße extern beeinflusste oder generell ungenaue Sensordaten in einem rekursiven Prozess in zwei Schritten zu einer Schätzung des realen Zustands verarbeitet. Es wird ein mathematisches Modell der Dynamik des Systems zugrunde gelegt, über das die Daten miteinander verknüpft werden. Weitere Eingangsgrößen sind externe Einflüsse auf das System. Es ist zu beachten, dass das Kalman Filter nur für die Modellierung linearer Zustandsräume eingesetzt werden kann.

4.1.1 Prädiktion

Im ersten Schritt, der Prädiktion, wird eine Voraussage des aktuellen Zustands auf Basis des vorherigen Resultats und der physikalischen Zusammenhänge getroffen.

$$\mathbf{x}_{t_p} = \mathbf{F}_{t-1}\mathbf{x}_{t-1} + \mathbf{B}_{t-1}\mathbf{u}_{t-1} \quad (4.1)$$

\mathbf{x} ist Zustandsmatrix des Systems, \mathbf{x}_{t_p} beschreibt hier die Zustandsmatrix für den aktuellen Zeitpunkt t_p (Prädiktion), \mathbf{x}_{t-1} die Zustandsmatrix des vorherigen Schrittes. \mathbf{F}_{t-1} ist die Übergangsmatrix, die den Zustand des Systems vom Zeitpunkt $t - 1$ zum jetzigen Zeitpunkt propagiert. \mathbf{B}_{t-1} ist die Steuermatrix, die die externen Einflüsse auf das System \mathbf{u}_{t-1} auf den jetzigen Zeitpunkt propagiert.

$$\mathbf{P}_{t_p} = \mathbf{F}_{t-1}\mathbf{P}_{t-1}\mathbf{F}^T + \mathbf{Q}_t \quad (4.2)$$

\mathbf{P}_{t_p} ist die Prädiktion der Kovarianzmatrix. Die Kovarianzmatrix bildet die bei der Prädiktion von \mathbf{x}_{t_p} entstandenen Fehler ab. \mathbf{P}_{t-1} ist die Kovarianzmatrix aus dem vorherigen Schritt. \mathbf{Q}_t ist die Prozessrauschmatrix, die weitere Unsicherheiten in der Prädiktion, zum Beispiel Modellierungsfehler oder externe (unbekannte) Störungen beschreibt. Das hochgestellte \mathbf{T} bezeichnet die transponierte Matrix, also eine Matrix die durch Vertauschung von Zeilen und Spalten der originalen Matrix entsteht.

4.1.2 Korrektur

Im zweiten Schritt, der Korrektur, wird die Prädiktion aus dem ersten Schritt mit der Einpflegung neuer Messwerte korrigiert.

$$\mathbf{K}_t = \frac{\mathbf{P}_{t_p}\mathbf{H}_t^T}{\mathbf{H}_t\mathbf{P}_{t_p}\mathbf{H}_t^T + \mathbf{R}_t} \quad (4.3)$$

Zuerst wird die Kalman-Gain-Matrix \mathbf{K}_t berechnet. Hierzu werden die im ersten Schritt gebildete Prädiktion der Kovarianzmatrix \mathbf{P}_{t_p} und die Beobachtungsmatrix \mathbf{H}_t benötigt. \mathbf{H}_t bildet die Relation des Systemzustands zu den gemessenen Größen ab. Auch hier denotiert das hochgestellte \mathbf{T} wieder die transponierte Form der jeweiligen Matrix. Weiterhin wird hier \mathbf{R}_t als Kovarianzmatrix des Messrauschens eingeführt. Sie wird in

Annahme der Messungenauigkeit der Sensoren und der Korrelation des Messrauschens gebildet.

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t_p} \quad (4.4)$$

Nun wird die im ersten Schritten gebildete Prädiktion der Kovarianzmatrix \mathbf{P}_{t_p} korrigiert und als neue Kovarianzmatrix \mathbf{P}_t aufgestellt. Hier wird zusätzlich noch die Identitätsmatrix \mathbf{I} benötigt.

$$\mathbf{x}_t = \mathbf{x}_{t_p} + \mathbf{K}_t (\mathbf{z}_t - \mathbf{H}_t \mathbf{x}_{t_p}) \quad (4.5)$$

Jetzt kann die korrigierte Vorhersage \mathbf{x}_t berechnet werden. Die Kalman-Gain-Matrix spiegelt hier das zentrale Element des Verfahrens wider. Sie gewichtet Zustandsschätzung \mathbf{x}_{t_p} gegen die gemessenen Werte \mathbf{z}_t . Somit symbolisiert sie in gewisser Weise das *Vertrauen* des Prozesses in die eine oder andere Variable.

4.2 Extended Kalman Filter für nichtlineare Systeme

Das EKF ist eine Variante des Kalman Filters, die auch die Modellierung nichtlinearer Zustandsräume ermöglicht. Hierzu wird eine Linearisierung um den aktuellen Mittelwert und die aktuelle Kovarianz vorgenommen.

4.2.1 Prädiktion

$$\mathbf{x}_{t_p} = f(\mathbf{x}_{t-1}, \mathbf{u}_t) + \mathbf{w}_t \quad (4.6)$$

Die Schritte der Prädiktion ähneln der des Kalman Filters, mit dem Unterschied, dass hier für die Zustandsüberführung eine differenzierbare Funktion $f(\mathbf{x}_{t-1}, \mathbf{u}_t)$ verwendet wird. Diese Funktion hat als ersten Parameter die vorherige Schätzung \mathbf{x}_{t-1} und als zweiten Parameter die externen Einflüsse auf das System \mathbf{u}_t . \mathbf{w}_t sei multi-dimensionales, unabhängig und identisch verteilt (u.i.v.) gaußsches Rauschen mit der Verteilung $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}_t)$.

$$\mathbf{P}_{t_p} = \mathbf{F}_{t-1} \mathbf{P}_{t-1} \mathbf{F}_{t-1}^T + \mathbf{Q}_t \quad (4.7)$$

Die Prädiktion der Kovarianzmatrix wird hier analog zum Kalman Filter mit der Übergangsmatrix \mathbf{F}_{t-1} gebildet, die in Gleichung 4.8 definiert wird. \mathbf{Q}_t ist wie bisher die Prozessrauschensmatrix.

$$\begin{aligned}\mathbf{F}_t &= J_f(t) \\ &= \left. \frac{\partial f(x, \mathbf{u}_t)}{\partial \mathbf{x}} \right|_{x=\hat{x}_t}\end{aligned}\quad (4.8)$$

$f(\mathbf{x}_{t-1}, \mathbf{u}_t)$ wird hierzu in \mathbf{F}_t als Jacobi-Matrix $J_f(t)$ partiell abgeleitet. Dadurch wird eine Linearisierung des Zustandsübergangs erreicht.

4.2.2 Korrektur

$$\mathbf{K}_t = \frac{\mathbf{P}_{t_p} \mathbf{H}_t^T}{\mathbf{H}_t \mathbf{P}_{t_p} \mathbf{H}_t^T + \mathbf{R}_t}\quad (4.9)$$

Die Kalman-Gain-Matrix wird analog zum Kalman Filter gebildet, auch hier wird $h(t)$ als \mathbf{H}_t linearisiert:

$$\begin{aligned}\mathbf{H}_t &= J_h(t) \\ &= \left. \frac{\partial h(x)}{\partial \mathbf{x}} \right|_{x=\hat{x}_t}\end{aligned}\quad (4.10)$$

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t_p}\quad (4.11)$$

$$\mathbf{x}_t = \mathbf{x}_{t_p} + \mathbf{K}_t (\mathbf{z}_t - h(\mathbf{x}_{t_p}))\quad (4.12)$$

Nun wird noch analog zum Kalman Filter die korrigierte Kovarianzmatrix \mathbf{P}_t und die neue Vorhersage \mathbf{x}_t gebildet.

4.3 Konkrete Anwendung als Fahrzeugzustandsschätzung

In den folgenden Abschnitten wird die konkrete Anwendung des in Abschnitt 4.2 vorgestellten EKF für die Zustandsschätzung des Fahrzeugs beschrieben.

4.3.1 Zustandsmodell

In diesem Unterabschnitt wird das Zustandsmodell in Form des Zustandsvektor \mathbf{x} erläutert.

$$\begin{aligned}\mathbf{x} &= \begin{bmatrix} \mathbf{p} \\ \theta \\ \mathbf{v} \\ r \end{bmatrix} \\ \mathbf{p} &= \begin{bmatrix} x \\ y \end{bmatrix} \\ \mathbf{v} &= \begin{bmatrix} v_x \\ v_y \end{bmatrix}\end{aligned}\tag{4.13}$$

\mathbf{p} ist die Position des Fahrzeugs im globalen Koordinatensystem, mit Bezug auf das globale Koordinatensystem. θ ist der Kurs des Fahrzeugs, ebenfalls mit Bezug auf das globale Koordinatensystem. \mathbf{v} ist die Geschwindigkeit des Fahrzeugs, r die Gierrate, beides mit Bezug auf das lokale Koordinatensystem der IMU.

4.3.2 Prozessmodell

In diesem Unterabschnitt wird das Prozessmodell erläutert.

$$\begin{aligned}\dot{\mathbf{p}} &= \mathbf{R}(\theta)^T \mathbf{v} \\ &= \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}^T \begin{bmatrix} v_x \\ v_y \end{bmatrix} \\ &= \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} \\ &= \begin{bmatrix} \cos(\theta)v_x - \sin(\theta)v_y \\ \sin(\theta)v_x + \cos(\theta)v_y \end{bmatrix}\end{aligned}\tag{4.14}$$

$\mathbf{R}(\theta)$ sei die Rotationsmatrix der Ebene \mathbb{R}^2 für eine Drehung um θ . Die Ableitung der Position $\dot{\mathbf{p}}$ wird als die ins globale Koordinatensystem transformierte Geschwindigkeit definiert.

$$\dot{\theta} = r \quad (4.15)$$

Die Ableitung des Kurses $\dot{\theta}$ ist die Gierrate r .

$$\begin{aligned} \dot{\mathbf{v}} &= \mathbf{a} + \begin{bmatrix} v_y \\ -v_x \end{bmatrix} r + \mathbf{n}_v \\ &= \begin{bmatrix} a_x \\ a_y \end{bmatrix} + \begin{bmatrix} v_y r \\ -v_x r \end{bmatrix} + \mathbf{n}_v \\ &= \begin{bmatrix} a_x + v_y r + \mathbf{n}_v \\ a_y - v_x r + \mathbf{n}_v \end{bmatrix} \end{aligned} \quad (4.16)$$

Die Ableitung der Geschwindigkeit $\dot{\mathbf{v}}$ ist die Summe aus der Beschleunigung \mathbf{a} , der durch die Gierrate r verursachten Änderung der Geschwindigkeit sowie dem Messrauschen \mathbf{n}_v

$$\dot{r} = \mathbf{n}_r \quad (4.17)$$

Die Ableitung der Gierrate \dot{r} wird als Wert ohne Messmöglichkeit nur durch das Messrauschen \mathbf{n}_r modelliert.

Die in Gleichung 4.16 und Gleichung 4.17 verwendeten Messrauschen $\mathbf{n}_{\{.\}}$ werden jeweils als u.i.v. weißes Rauschen mit der Verteilung $\mathbf{n}_x \sim \mathcal{N}(0, \Sigma_x)$ modelliert.

$$\begin{aligned} f &= \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\theta} \\ \dot{\mathbf{v}} \\ \dot{r} \end{bmatrix} \\ &= \begin{bmatrix} \cos(\theta)v_x - \sin(\theta)v_y \\ \sin(\theta)v_x + \cos(\theta)v_y \\ r \\ a_x + v_y r + \mathbf{n}_v \\ a_y - v_x r + \mathbf{n}_v \\ \mathbf{n}_r \end{bmatrix} \end{aligned} \quad (4.18)$$

Aus den zuvor dargestellten Teilen wird die Funktion f gebildet. Diese wird durch die Gleichung 4.8 zu F_t linearisiert und schließlich in Gleichung 4.6 eingesetzt. Daraus ergibt sich x_{t_p} als Prädiktion des Zustands für den Zeitpunkt t .

$$\mathbf{P}_{\text{initial}} = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_\theta^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{v_x}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{v_y}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_r^2 \end{bmatrix} \quad (4.19)$$

Die Kovarianzmatrix \mathbf{P} muss für die erste Ausführung des Filters initialisiert werden. Hier wird vereinfacht angenommen, dass die gemessenen Größen sich nicht gegenseitig beeinflussen. So wird nur die Hauptdiagonale mit Werten gefüllt. Für die einzelnen Varianzen wird jeweils geschätzt, wie weit die initialen Messwerte von der Realität abweichen.

4.3.3 Messmodell

In diesem Unterabschnitt wird die Einführung der gemessenen Werte $\mathbf{z}_{\{\cdot\}}$ in das System beschrieben. Das Rauschen $\mathbf{n}_{\{\cdot\}}$ wird jeweils als u.i.v. gaußsches Rauschen modelliert.

$$\begin{aligned} \mathbf{z}_{\mathbf{p}} &= h_{\mathbf{p}}(x) = \mathbf{p} + \mathbf{R}(\theta)^T \mathbf{p}_s + \mathbf{n}_{\mathbf{z}_{\mathbf{p}}} \\ &= \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_y \end{bmatrix} + \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}^T \begin{bmatrix} \mathbf{p}_{s,x} \\ \mathbf{p}_{s,y} \end{bmatrix} + \mathbf{n}_{\mathbf{z}_{\mathbf{p}}} \\ &= \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_y \end{bmatrix} + \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \mathbf{p}_{s,x} \\ \mathbf{p}_{s,y} \end{bmatrix} + \mathbf{n}_{\mathbf{z}_{\mathbf{p}}} \\ &= \begin{bmatrix} \mathbf{p}_x + \cos(\theta)\mathbf{p}_{s,x} - \sin(\theta)\mathbf{p}_{s,y} + \mathbf{n}_{\mathbf{z}_{\mathbf{p}}} \\ \mathbf{p}_y + \sin(\theta)\mathbf{p}_{s,x} + \cos(\theta)\mathbf{p}_{s,y} + \mathbf{n}_{\mathbf{z}_{\mathbf{p}}} \end{bmatrix} \end{aligned} \quad (4.20)$$

Die Messdaten der Position $\mathbf{z}_{\mathbf{p}}$ werden eingeführt als Summe aus den gemessenen Daten \mathbf{p} , der mit der inversen Rotationsmatrix $\mathbf{R}(\theta)^T$ in das globale Koordinatensystem transformierten Sensorposition \mathbf{p}_s und dem Rauschen $\mathbf{n}_{\mathbf{z}_{\mathbf{p}}}$.

$$\mathbf{z}_\theta = h_\theta(x) = \theta + \theta_s + \mathbf{n}_{\mathbf{z}_\theta} \quad (4.21)$$

Die Messdaten des Kurses \mathbf{z}_θ werden eingeführt als Summe aus dem gemessenen Kurs θ , der Drehung des Sensors im Fahrzeugkoordinatensystem θ_s , sowie dem Rauschen $\mathbf{n}_{\mathbf{z}_\theta}$.

$$\begin{aligned}
 \mathbf{z}_v = h_v(x) &= \mathbf{R}(\theta_s)(\mathbf{v} + r \begin{bmatrix} -\mathbf{p}_{s,y} \\ \mathbf{p}_{s,x} \end{bmatrix}) + \mathbf{n}_{\mathbf{z}_v} \\
 &= \begin{bmatrix} \cos(\theta_s) & \sin(\theta_s) \\ -\sin(\theta_s) & \cos(\theta_s) \end{bmatrix} \begin{bmatrix} v_x - r\mathbf{p}_{s,y} \\ v_y + r\mathbf{p}_{s,x} \end{bmatrix} + \mathbf{n}_{\mathbf{z}_v} \\
 &= \begin{bmatrix} \cos(\theta_s)(v_x - r\mathbf{p}_{s,y}) + \sin(\theta_s)(v_y + r\mathbf{p}_{s,x}) + \mathbf{n}_{\mathbf{z}_v} \\ -\sin(\theta_s)(v_x - r\mathbf{p}_{s,y}) + \cos(\theta_s)(v_y + r\mathbf{p}_{s,x}) + \mathbf{n}_{\mathbf{z}_v} \end{bmatrix}
 \end{aligned} \tag{4.22}$$

Die Messdaten der Geschwindigkeit \mathbf{z}_v werden eingeführt als Summe der jeweils mit der Rotationsmatrix $\mathbf{R}(\theta_s)$ ins Fahrzeugkoordinatensystem transformierten Geschwindigkeit und aus der Gierrate und Sensorposition gebildeten Geschwindigkeit, sowie dem Rauschen $\mathbf{n}_{\mathbf{z}_v}$.

$$\mathbf{z}_r = h_r(x) = r + \mathbf{n}_{\mathbf{z}_r} \tag{4.23}$$

Die Messdaten der Gierrate \mathbf{z}_r werden eingeführt als Summe aus der gemessenen Gierrate r und dem Rauschen $\mathbf{n}_{\mathbf{z}_r}$.

$$\begin{aligned}
 h &= \begin{bmatrix} h_p(x) \\ h_\theta(x) \\ h_v(x) \\ h_r(x) \end{bmatrix} \\
 &= \begin{bmatrix} \mathbf{p}_x + \cos(\theta)\mathbf{p}_{s,x} - \sin(\theta)\mathbf{p}_{s,y} + \mathbf{n}_{\mathbf{z}_p} \\ \mathbf{p}_y + \sin(\theta)\mathbf{p}_{s,x} + \cos(\theta)\mathbf{p}_{s,y} + \mathbf{n}_{\mathbf{z}_p} \\ \theta + \theta_s + \mathbf{n}_{\mathbf{z}_\theta} \\ \cos(\theta_s)(v_x - r\mathbf{p}_{s,y}) + \sin(\theta_s)(v_y + r\mathbf{p}_{s,x}) + \mathbf{n}_{\mathbf{z}_v} \\ -\sin(\theta_s)(v_x - r\mathbf{p}_{s,y}) + \cos(\theta_s)(v_y + r\mathbf{p}_{s,x}) + \mathbf{n}_{\mathbf{z}_v} \\ r + \mathbf{n}_{\mathbf{z}_r} \end{bmatrix}
 \end{aligned} \tag{4.24}$$

Aus den zuvor dargestellten Teilen wird die Funktion h gebildet. Diese wird durch die Gleichung 4.10 zu H_t linearisiert und schließlich in den Gleichungen 4.9, 4.11, 4.12 eingesetzt. Aus Gleichung 4.12 ergibt sich damit die neue Vorhersage \mathbf{x}_t .

5 Fazit

In diesem Kapitel werden die Ergebnisse dieser Arbeit zusammengefasst und es wird ein Ausblick auf Weiterentwicklungsmöglichkeiten gegeben.

5.1 Zusammenfassung

In dieser Arbeit wurden die grundlegenden Aspekte eines autonomen Systems am Beispiel eines Formula Student Rennwagens thematisiert. Es wurde ein Überblick über die vom Reglement des Wettbewerbs vorgegebenen Voraussetzungen gegeben. Auf Basis dieser wurde die Aufgabenstellung erklärt.

Die für die Entwicklung eines autonomen Systems notwendigen Sensoren und Aktoren wurden betrachtet. Dabei wurde unterschieden zwischen den bereits vorhandenen Systemen des Fahrzeugs und den neu hinzukommenden Systemen. Die für die Zustandsschätzung nötigen Sensoren wurden detailliert beschrieben.

Darauf aufbauend wurde eine Systemarchitektur für das autonome System beschrieben. Hierbei wurde auf die Vernetzung der Komponenten, die Zeitsynchronisation des Systems und die zur Verwendung kommenden Koordinatensysteme eingegangen. Weiterhin wurde ein Konzept für die Referenzierung der Sensoren zueinander vorgeschlagen sowie die Softwarearchitektur beschrieben. Es wurde ein kurzer Ausblick auf Ansätze zur Verarbeitung der Punktwolken des LiDAR sowie für die Erstellung einer Karte der Fahrzeugumgebung und der Lokalisation darin gegeben.

Es wurde ein Verfahren für die Schätzung von Fahrzeugposition, Fahrzeugorientierung und Fahrzeuggeschwindigkeit beschrieben. Dazu wurde zunächst das Kalman Filter betrachtet. Vertiefend wurde das Extended Kalman Filter erläutert. Auf Basis des Extended Kalman Filter wurde ein Fahrzeugmodell gebildet, das die Fusion der verschiedenen Sensordaten ermöglicht.

5.2 Ausblick

Die in dieser Arbeit ausgeführte Basis für ein autonomes System lässt noch viel Raum für die Erweiterung durch weiterführende Arbeiten. In diesem Abschnitt soll Ausblick auf die möglichen Vertiefungen gegeben werden.

Wie in Kapitel 1 aufgezeigt, hat sich die Formula Student Germany als erster Wettbewerb der Elektrifizierung der Formula Student verschrieben. Es ist anzunehmen, dass sich dies als Trend auf weitere Wettbewerbe ausdehnen könnte. Daher sollte für die weitere Entwicklung des autonomen Systems Rücksicht auf die Kompatibilität zu verschiedenen Antriebsstrangkonzepthen genommen werden.

Im Laufe der Tests des autonomen Systems kann es sinnvoll werden, weitere Sensoren in das Fahrzeug zu integrieren, um die Präzision weiter zu steigern. Ebenso könnte damit eine Erweiterung des Zustandsmodells auf neue Mess- oder Prozessgrößen vorgenommen werden.

Ein weiterer wichtiger Aspekt für die Entwicklung eines robusten autonomen Systems ist die Redundanz in der Sensorbasis. Auch wenn ein Fahrzeugkonzept mit einem LiDAR als einzigem Umgebungssensor möglich ist, so lässt die fehlende Redundanz dies nicht als empfehlenswert erscheinen. Daher ist die Entwicklung einer Bildverarbeitung für die bereits in Unterabschnitt 2.2.4 angedeutete Kamera sinnvoll.

Ebenso ergibt eine tiefere Untersuchung der in Abschnitt 2.3 angerissenen Aktoren des autonomen Systems Sinn.

Die in Abschnitt 3.6 beschriebene LiDAR Pipeline kann ebenfalls noch als Anlass für weitere Vertiefungen verwendet werden. Hier könnte zum Beispiel die Verwendung eines neuronalen Netzes zur Erkennung der Pylonen untersucht werden.

Bei dem in Abschnitt 3.7 angesprochenen SLAM Algorithmus bedarf es der Recherche und Entwicklung eines ganzheitlichen Ansatzes. Dabei sollte auch untersucht werden, ob für den in Unterabschnitt 1.1.3 erklärten Autocross aufgrund der Unkenntnis der Strecke ein gänzlich anderer Ansatz gewählt werden sollte.

Abschließend ist noch die dem SLAM-Algorithmus im Datenfluss nachgelagerte Pfadplanung sowie Fahrdynamikregelung zu erwähnen. Diese Themen sind ebenfalls eine kritische Komponente der Funktion eines autonomen Fahrzeugs, für die es noch eines Konzeptes bedarf.

Literaturverzeichnis

- [1] FastSLAM 2.0. In: MONTEMERLO, Michael (Hrsg.) ; THRUN, Sebastian (Hrsg.): *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*. Springer Berlin Heidelberg, S. 63–90. – URL https://doi.org/10.1007/978-3-540-46402-0_4. – ISBN 978-3-540-46402-0
- [2] : *Formula Student - World Ranking*. – URL <https://mazur-events.de/fs-world/>. – Zugriffsdatum: 2019-11-11
- [3] Formula Student Rules 2019. . – URL https://www.formulastudent.de/fileadmin/user_upload/all/2019/rules/FS-Rules_2019_V1.1.pdf. – Zugriffsdatum: 2019-10-10
- [4] IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.
- [5] IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications - Data Terminal Equipment (DTE) Power Via Media Dependent Interface (MDI). . – ISSN null
- [6] BASLER AG: *Basler ace aca1440-73gc - Flächenkamera*. – URL </de/produkte/kameras/flaechenkameras/ace/aca1440-73gc/>. – Zugriffsdatum: 2019-09-08
- [7] BERSANI, M. ; VIGNATI, M. ; MENTASTI, S. ; ARRIGONI, S. ; CHELI, F.: Vehicle state estimation based on Kalman filters. In: *2019 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE)*, S. 1–6

- [8] BOSCH ENGINEERING GMBH: *Engine Control Unit MS 6.1*. – URL <http://www.bosch-motorsport.de/content/downloads/Raceparts/en-GB/48337035196150923.html>. – Zugriffsdatum: 2019-11-27
- [9] DE, Shidhartha: *HAWKS H14*. – URL <https://media.formulastudent.de/2018/Hockenheim/20180811-Saturday/i-MtHVnRp>. – Zugriffsdatum: 2019-09-11
- [10] DITTMER-SCHAFF, Marco: *Automatisierung einer Lenkung eines HAWKS Driverless Car*
- [11] DOSEN, Mario: *Konzeption einer Bremsanlage für einen autonomfahrenden Rennwagen*
- [12] FANG, Huazhen ; HAILE, Mulugeta A. ; WANG, Yebin: Robust Extended Kalman Filtering for Systems with Measurement Outliers. . – URL <http://arxiv.org/abs/1904.00335>. – Zugriffsdatum: 2019-10-06
- [13] FORMULA STUDENT GERMANY: FSG Competition Handbook 2019. . – URL https://www.formulastudent.de/fileadmin/user_upload/all/2019/rules/FSG19_Competition_Handbook_v1.1.pdf. – Zugriffsdatum: 2019-10-11
- [14] FORMULA STUDENT GERMANY: *New special award for driverless Formula Student Concepts*. – URL <https://www.formulastudent.de/pr/news/details/article/new-special-award-for-driverless-formula-student-concepts/>. – Zugriffsdatum: 2019-11-11
- [15] FORMULA STUDENT GERMANY: *An outlook on FSG 2021 and the following seasons*. – URL <https://www.formulastudent.de/pr/news/details/article/an-outlook-on-fsg-2021-and-the-following-seasons/>. – Zugriffsdatum: 2019-11-11
- [16] FORMULA STUDENT GERMANY: *Strategic Announcement*. – URL <https://www.formulastudent.de/pr/news/details/article/fsg-strategic-announcement/>. – Zugriffsdatum: 2019-11-11
- [17] HAUSMAN, K. ; WEISS, S. ; BROCKERS, R. ; MATTHIES, L. ; SUKHATME, G. S.: Self-calibrating multi-sensor fusion with probabilistic measurement validation for seamless sensor switching on a UAV. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*, S. 4289–4296

- [18] KABZAN, Juraj ; VALLS, Miguel I. ; REIJGWART, Victor J F. ; HENDRIKX, Hubertus F C. ; EHMKE, Claas ; PRAJAPAT, Manish ; BUHLER, Andreas ; GOSALA, Nikhil ; GUPTA, Mehak ; SIVANESAN, Ramya ; DHALL, Ankit ; CHISARI, Eugenio ; KARNCHANACHARI, Napat ; BRITS, Sonja ; DANGEL, Manuel ; SA, Inkyu ; DUBE, Renaud ; GAWEL, Abel ; PFEIFFER, Mark ; LINIGER, Alexander ; LYGEROS, John ; SIEGWART, Roland: AMZ Driverless: The Full Autonomous Racing System.
- [19] KISTLER INSTRUMENTE GMBH: *Correvit SFII: 2-achsige optische Sensoren.* – URL <https://www.kistler.com/de/produkt/type-csf2a/>. – Zugriffsdatum: 2019-10-31
- [20] LYNEN, S. ; ACHELNIK, M. W. ; WEISS, S. ; CHLI, M. ; SIEGWART, R.: A robust and modular multi-sensor fusion approach applied to MAV navigation. In: *Proc. of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, S. 3923–3929
- [21] MIKE1024: *A diagram showing ECEF, ENU, Longitude and Latitude coordinates and the relationship between them..* – URL https://commons.wikimedia.org/wiki/File:ECEF_ENU_Longitude_Latitude_relationships.svg. – Zugriffsdatum: 2019-10-10
- [22] MONTEMERLO, Michael ; THRUN, Sebastian ; KOLLER, Daphne ; WEGBREIT, Ben: FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In: *In Proceedings of the AAAI National Conference on Artificial Intelligence*, AAAI, S. 593–598
- [23] OUSTER INC.: *OS-1 Lidar Sensor.* – URL <https://www.ouster.io/product-os1>. – Zugriffsdatum: 2019-06-25
- [24] PRAWITZ, Sven: *Solid-State-Lidar: AMS kooperiert mit Ibeo und ZF.* – URL <https://www.automobil-industrie.vogel.de/solid-state-lidar-ams-kooperiert-mit-ibeo-und-zf-a-831322/>. – Zugriffsdatum: 2019-06-25
- [25] QNIEMIEC: *Roll-pitch-yaw angles of cars and other land based vehicles.* – URL https://commons.wikimedia.org/wiki/File:RPY_angles_of_cars.png?uselang=de. – Zugriffsdatum: 2019-10-10
- [26] QUIGLEY, Morgan ; GERKEY, Brian ; CONLEY, Ken ; FAUST, Josh ; FOOTE, Tully ; LEIBS, Jeremy ; BERGER, Eric ; WHEELER, Rob ; NG, Andrew: ROS: an open-source Robot Operating System. In: *ICRA Workshop on Open Source*

- Software*, URL <http://www.willowgarage.com/sites/default/files/icraoss09-ROS.pdf>. – Zugriffsdatum: 2019-08-01
- [27] TÖPPNER, Ralf: Echtzeitplanung bei CAN-basierter Kommunikation. . – URL <https://userpages.uni-koblenz.de/~zoebel/ws2004/CAN.pdf>. – Zugriffsdatum: 2019-09-03
- [28] VALLS, Miguel de la I. ; HENDRIKX, Hubertus Franciscus C. ; REIJGWART, Victor ; MEIER, Fabio V. ; SA, Inkyu ; DUBÉ, Renaud ; GAWEL, Abel R. ; BÜRKI, Mathias ; SIEGWART, Roland: Design of an Autonomous Racecar: Perception, State Estimation and System Integration. . – URL <http://arxiv.org/abs/1804.03252>. – Zugriffsdatum: 2018-10-13
- [29] VECTORNAV TECHNOLOGIES: *VN-300 - VectorNav Technologies*. – URL <https://www.vectornav.com/products/vn-300>. – Zugriffsdatum: 2019-10-31
- [30] VOLLMER, Alfred: *Was tut sich bei Solid-State-Lidar-Sensoren?*. – URL <https://www.all-electronics.de/lidar-sensoren-automotive/>. – Zugriffsdatum: 2019-06-25

Glossar

GNSS *engl.* **G**lobal **N**avigation **S**atellite **S**ystem - Sammelbegriff für verschiedene globale Satellitennavigationssysteme wie z.B. NAVSTAR GPS, GLONASS, Galileo oder Beidou.

LiDAR *engl.* **L**ight **D**etection **A**nd **R**anging - ein Sensor, der Entfernung (und optional Reflektivität sowie Intensität) von Objekten in diskreten Schritten der vertikalen und horizontalen Achse misst und diese i.d.R. als Punktwolke ausgibt.

SLAM *engl.* **S**imultaneous **L**ocalization **A**nd **M**apping - Problemstellung auf der Robotik, hier übertragen auf ein Fahrzeug: Dieses muss während es sich fortbewegt eine Karte seiner Umgebung erstellen und sich innerhalb dieser Karte lokalisieren.

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „– bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] – ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen.“

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: Bandow

Vorname: Philipp

dass ich die vorliegende Bachelorarbeit – bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Sensorfusion in autonomen Fahrzeugen

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Hamburg

Ort

12.12.2019

Datum

Unterschrift im Original