

Protoypisierung der winkelbasierten Bluetooth Ortung eines intelligenten Fahrradhelms

Schwerpunkt Systemdesign

Studienarbeit

Studiengang Elektrotechnik

Studienrichtung Automation

Duale Hochschule Baden-Württemberg Ravensburg, Campus Friedrichshafen

von

Andy Dennis Gardy

Abgabedatum:	17.07.2023
Bearbeitungszeitraum:	09.04.2023 - 17.07.2023
Matrikelnummer:	1376352
Kurs:	TEA20
Gutachter der Dualen Hochschule:	Prof. Dr.-Ing. Tobias Frank

Erklärung

gemäß Ziffer 1.1.13 der Anlage 1 zu §§ 3, 4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2017 in der Fassung vom 25.07.2018.

Ich versichere hiermit, dass ich meine Studienarbeit mit dem Thema:

Protoypisierung der winkelbasierten Bluetooth Ortung eines intelligenten Fahrradhelms

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Potsdam, den 17. Juli 2023

A handwritten signature in black ink, appearing to read "D. Gardy", is written over a horizontal line.

Andy Dennis Gardy

Gender-Hinweis

Zur besseren Lesbarkeit wird in der vorliegenden Arbeit auf die gleichzeitige Verwendung männlicher und weiblicher Sprachformen verzichtet. Es wird das generische Maskulinum verwendet, wobei beide Geschlechter gleichermaßen gemeint sind.

Kurzfassung

Die bestimmungsgemäße Verwendung des Prototyps intelligenter Fahrradhelm Teilsystem Bluetooth Ortung sieht vor, die Position eines intelligenten Fahrradhelms in einem Abdeckungsbereich rechtsseitig eines Lkw zu bestimmen. Das Ortungssystem des Prototyps ist auf niedrige Relativgeschwindigkeiten zwischen Lkw und Fahrradfahrer ausgelegt. Für die Auslegung des Systems ist eine Relativgeschwindigkeit $v = 20 \frac{\text{km}}{\text{h}}$ angenommen. Der Prototyp ist in der Funktion eines Abbiege-Assistenten ausschließlich bei einer Annäherung des Fahrradfahrers von hinten in Fahrtrichtung des Lkw zu verwenden. Die Ortung erfolgt nach der Methode der Triangulation. Dafür bestimmen zwei, am Lkw verortete Empfänger jeweils den Einfallsinkel von Signalen, die ein Sender im Fahrradhelm initiiert. Das Systemdesign des Prototyps definiert im ersten Schritt seinen Mindestabdeckungsbereich. Dieser beträgt von der vorderen rechten Ecke des Lkw ausgehend mindestens 3 m in y -Richtung und 5 m in x -Richtung. Die Einschränkung des x -Bereichs auf 5 m erfolgt aus konstruktiven Gründen. Aufgrund dieser Einschränkung ist die Anforderung an den Mindestabdeckungsbereich, gemäß Verkehrsblatt 2022, Heft 8, Nr. 65 des Bundesministeriums für Digitales und Verkehr, das eine Abdeckung in Längsrichtung im Bereich $-2 \text{ m} < x < 9 \text{ m}$ vorschreibt, nicht erfüllt. Dafür sind im Wesentlichen der Abstand zwischen den Empfängern sowie die Genauigkeit der Hardware mit welcher der Prototyp realisiert ist verantwortlich. Der Prototyp besteht neben der Stromversorgung und Leitungen zur Datenübertragung aus zwei Empfängern und einem Sender der u-Blox Hardware XPLR-AoA1. Im Rahmen einer Vorstudie werden die Eigenschaften dieser Hardware identifiziert und bewertet. Hohe Relevanz hat dabei die Genauigkeit der Signal-Einfallsinkelbestimmung. Je geringer die Abweichungen der von den Empfängern bestimmten Einfallsinkel vom wahren Wert ist, desto genauer kann die Position des Senders bestimmt werden. Es wird aufgezeigt, dass in guter Näherung der Signal-Einfallsinkelbereich $\pm 75^\circ$ abgedeckt wird. Damit ergibt sich für jeden Empfänger ein Signal-Einfallsinkel-Erfassungsbereich mit einem Öffnungswinkel von 150° . Infolge der eingeschränkten Signal-Einfallsinkel-Erfassungsbereich werden neben der Länge d_1 die Stellgrößen $\theta_{2_{\text{offset}}}$ und $\theta_{3_{\text{offset}}}$ definiert. Über die Einstellung dieser Freiheitsgrade wird der Bereich in dem die Ortung mit akzeptabler Genauigkeit erfolgt,

variiert. Der Prototyp sieht eine statische Einstellung der Stellgrößen bei einer Montage am Lkw vor. Durch Implementierung eines Algorithmus wird die Stellgrößeneinstellung identifiziert, mit welcher die Genauigkeit der Ortung im Betrachtungsbereich optimal ist. Der Betrachtungsbereich ist größer gewählt als der o. g. Mindestabdeckungsbereich. Er beginnt in horizontaler Richtung mit der vorderen rechten Ecke des Lkw und endet bei $x = 15$ m. In vertikaler Richtung wird der Bereich im Abstand zwischen $0 \text{ m} < y < 10$ m rechtsseitig des Lkw betrachtet. Nach Festlegung der Grenzen wird der Betrachtungsbereich äquidistant in 0,5 m große Zellen unterteilt. Jede Zelle wird mit einem Faktor gewichtet, der die Unfallgefahr zwischen Fahrradfahrer und LKW sowie die Sichtbarkeit des Zellbereichs mithilfe der optischen Einrichtungen Fenster bzw. Spiegel berücksichtigt. Der Algorithmus berechnet die Schnittpunkte der wahren Signal-Einfallsinkel und der abweichenden Signaleinfallsinkel für eine bestimmte Position und gibt die absolute Differenz beider Werte zurück. Diese Differenz gibt vereinfacht die Ungenauigkeit der Ortung für eine bestimmte Position an. Des Weiteren erfolgt die Bestimmung der Ungenauigkeit der Ortung jeder einzelnen Zelle. Die Gesamtungenaugkeit entspricht dem Mittelwert der gewichteten Ungenauigkeiten aller Zellen. Diese Berechnung erfolgt iterativ für alle möglichen Stellgrößen-Kombinationen. Im Ergebnis werden mit $d_1 = 5$ m, $\theta_{2_{offset}} = 55^\circ$ und $\theta_{3_{offset}} = -10^\circ$ die Stellgrößen für die geringste Gesamtungenaugigkeit identifiziert. Nach der Untersuchung von Systemeigenschaften und Synthese der Einstellparameter folgt der Schritt Prototyp-Entwicklung inklusive Implementierung der Triangulation und Visualisierung der Ortungsergebnisse. Beide Funktionen sind über ein Python Script realisiert, dass über einen Host-PC aufgerufen wird. Inhaltlich umfasst das Python Script die Konfiguration der seriellen Schnittstellen, eine Verarbeitung der seriellen Eingangsdaten, welche von den Empfängern an den Host-PC übertragen werden, eine Berechnung der Position des Senders und den Aufruf der grafischen Oberfläche. Der Prototyp wird auf die Eigenschaften Genauigkeit der Ortung und räumliche Auflösung untersucht. Dabei werden verschiedene Positionen des Senders bei Einstellung der identifizierten Stellgrößen getestet. Die Genauigkeit erreicht in den getesteten Fällen akzeptable Werte. Um den vom Gesetzgeber geforderten Mindestabdeckungsbereich zu erfüllen sind die Stellgrößen auf $d_1 = 9$ m, $\theta_{2_{offset}} = 40^\circ$ und $\theta_{3_{offset}} = -60^\circ$ anzupassen. Neben der Erfüllung des Mindestabdeckungsbereichs wird die Genauigkeit der Ortung in nahezu jedem Bereich verbessert. Die räumliche Auflösung der Position des bewegten Fahrradhelms bei konstanter Relativgeschwindigkeit zwischen Lkw und Fahrradfahrer wird als unzureichend eingestuft. Ursache dafür ist die Programmlaufzeit

des implementierten Python Scripts. Mit der Implementierung eines gleitenden Mittelwertfilters wird hierfür eine Verbesserungsmaßnahme vorgeschlagen. Abschließend wird die Forschungsfrage der Studienarbeit beantwortet.

Inhaltsverzeichnis

1 Einleitung	1
1.1 Motivation	1
1.2 Problemstellung	1
1.3 Ausgangslage	2
1.4 Zielsetzung der Studienarbeit	2
1.5 Geplantes Vorgehen	2
2 Stand der Technik bei Abbiegeassistentensystemen	3
3 Vorstudienergebnisse der Untersuchung von Eigenschaften der Hardware XPLR-AOA-1	7
3.1 Genauigkeit der Signal-Einfallswinkelbestimmung	7
3.2 Robustheit bei Störeinflüssen	9
3.3 Räumliche Auflösung der Messwerterfassung	10
4 Systemdesign intelligenter Fahrradhelm Teilsystem Bluetooth Ortung	11
4.1 Systembeschreibung	11
4.2 Freiheitsgrade zur Beeinflussung der Genauigkeit der Ortung	14
4.3 Algorithmus zur Identifizierung der optimalen Stellgrößen	18
4.4 Hardware	26
4.4.1 Empfänger	26
4.4.2 Sender	27
4.4.3 Halterung	28
4.5 Software	28
4.5.1 Konfiguration der seriellen Schnittstelle	28
4.5.2 Eingangsdatenverarbeitung	29
4.5.3 Positionsberechnung	34
4.5.4 Grafische Ausgabe	36
4.5.5 Programmlaufzeit des Python Scripts zum Aufruf der Visualisierung	39
4.5.6 Bediener-Schnittstelle	41

Inhaltsverzeichnis

5 Verifizierung und Bewertung des Prototyps	43
5.1 Genauigkeit der Ortung	43
5.1.1 Versuchsbeschreibung	43
5.1.2 Durchführung	44
5.1.3 Auswertung	44
5.2 Zeitverhalten des Python Scripts zum Aufruf der Visualisierung	54
6 Zusammenfassung	57
Literatur	61
Verzeichnis verwendeter Formelzeichen und Abkürzungen	64
Abbildungsverzeichnis	67
Tabellenverzeichnis	69

1 Einleitung

1.1 Motivation

Circa 50 % aller Verkehrsteilnehmer, die schwere Verletzungen infolge von Unfällen im deutschen Straßenverkehr erleiden, sind der Gruppe der Radfahrer zuzuordnen [Tho22]. Ein Großteil der Unfälle erfolgt im Zusammenhang mit schweren Lastkraftwagen (Lkw) [SS15]. Besonders für Lkw-Fahrer, welche konstruktiv bedingt über ein eingeschränktes Sichtfeld verfügen, ist das Erkennen von Fahrradfahrern erschwert. Im Fall einer Kollision schützt den Fahrradfahrer ausschließlich sein Fahrradhelm. Bei Betrachtung eines Lkw fällt auf, dass ein besonders hohes Risiko für die schwächeren Verkehrsteilnehmer resultiert, wenn diese unter den LKW geraten. Das Bestreben der Studienarbeit liegt darin, einen Beitrag zur Verbesserung der Sicherheit von Fahrradfahrern im Straßenverkehr zu leisten.

1.2 Problemstellung

Entsprechend der EU-Verordnung 2019/2144 sind Abbiegeassistenten ab dem 6. Juli 2022 für neue Fahrzeugtypen und ab dem 7. Juli 2024 für alle neu zugelassenen Fahrzeuge Pflicht. [Gui22] Eine Problemstellung die der Arbeit zugrunde liegt, ist, dass derzeit am Markt verfügbare Abbiegeassistenzsysteme in der Nachrüstung, der eignen Einschätzung zur Folge, hohe Kosten erzeugen. Die Bedeutung der vorliegenden Studienarbeit liegt in der Entwicklung eines kostengünstigen ersten Prototypen mit dem Funktionsumfang eines Abbiegeassistenzsystems.

1.3 Ausgangslage

Die vorliegende Arbeit schließt inhaltlich an die Studienarbeit [Den23] an. In einer Vorstudie wurden Eigenschaften der Hardware XPLR-AOA-1 bei der Erfassung des Signal-Einfallswinkels (Angle of Arrival (AoA)) untersucht. Auf Grundlage der gewonnenen Erkenntnisse skizziert [Den23] einen Prototyp. Eine Realisierung des Prototyps liegt nicht vor.

1.4 Zielsetzung der Studienarbeit

Die vorliegende Arbeit verfolgt das Ziel, diese Lücke zu schließen. Gegenstand der Studienarbeit ist die Entwicklung und Erprobung des Prototyps inklusive Beantwortung der Forschungsfrage ob ein Prototyp auf Basis der u-Blox Hardware XPLR-AoA1, als Abbiegeassenzsystem geeignet ist.

1.5 Geplantes Vorgehen

Zunächst wird der Stand der Technik für Abbiegeassenzsysteme herausgearbeitet. Darüber hinaus werden die Vorstudienergebnisse zusammengefasst und für die Entwicklung des Prototyps aufbereitet. Der Entwicklungsprozess sieht ein Systemdesign inklusive Eingrenzung der bestimmungsgemäßen Verwendung, Untersuchung von Anforderungen und Synthese von Stellgrößen vor. Darauf aufbauend wird die Realisierung des Prototyps beschrieben. Es folgt abschließend eine Verifizierung und Bewertung der Anforderungen. Auf Grundlage der gewonnenen Erkenntnisse wird die Forschungsfrage beantwortet. Das Systemdesign umfasst methodisch numerische Berechnungen zur Identifizierung der Stellgrößen, messtechnische Untersuchungen des Prototyps und Software-Entwicklung zur Implementierung von Funktionen zur Ortung und Visualisierung des Fahrradhelms.

2 Stand der Technik bei Abbiegeassenzsystemen

Abbiegeassenzsysteme sind Sicherheitseinrichtungen, die den Umgebungsbereich eines Lastkraftwagens (Lkw) bzw. Busses überwachen und bei Erkennen einer Gefährdung den Fahrer auf die potenziell gefährliche Situation hinweisen. Mögliche Gefährdungen sind Fußgänger oder Fahrradfahrer im toten Winkel des Sichtbereichs des Lkw-Fahrers. Gegenwärtig am Markt verfügbare Abbiegeassistenten verfügen über eine zweistufige Warneinrichtung. Befindet sich der gefährdete, schwächere Verkehrsteilnehmer im toten Winkel erfolgt ein visueller Hinweis. Erkennt der Abbiegeassistent eine bevorstehende Kollision z. B. indem der Lkw-Fahrer den Abbiegevorgang einleitet, ertönt ein akustisches Signal [o V23b]. Analysen der Unfallforschung der Versicherer zufolge sterben jährlich circa 28 Fahrradfahrer im Zusammenhang mit schweren Lkw bei Abbiegeunfällen im deutschen Straßenverkehr. Daraüber hinaus sind 160 Schwerverletzte zu verzeichnen [Dr-18]. Die Integration von Abbiegeassenzsystemen weist demnach ein hohes Potenzial auf, Zusammenstöße zwischen abbiegenden Lkw und Fahrradfahrern zu verringern. Abbiegeassenzsysteme, welche über eine von Kraftfahrtbundesamt (KBA) erteilte Allgemeine Betriebserlaubnis (ABE) verfügen, weisen Stand 07.2023 zwei technische Ansätze zur Erfüllung ihrer Funktion auf. Kamerabasierte Systeme erfassen durch Verwendung einer Weitwinkelkamera vollständig den toten Winkel rechtsseitig des Lkw. In Kombination verbaute Monitore erweitern durch eine Darstellung des Kamerabilds den Sichtbereich des Lkw-Fahrers. Die Bilddaten können über geeignete Algorithmen analysiert und erfasste Objekte klassifiziert werden. Bei einer Zuordnung der Bilddaten zur Klasse dynamischer Objekte z. B. Fahrradfahrern wird der Lkw-Fahrer gewarnt. Statische Objekte z. B. Ampeln oder parkende Autos werden ignoriert. Einen zweiten Ansatz bilden Radio detection and ranging (Radar) basierte Systeme. Sie überwachen mithilfe von elektromagnetischen Wellen im Radiofrequenzbereich den Bereich rechts neben dem Lkw und detektieren bewegte Objekte über implementierte Ortungsalgorithmen. Bis zum 04.04.2023 erhielten sechs Abbiegeassenzsysteme zur Freiwilligen Aus- und Nachrüstung von Lastkraftwagen und Bussen eine ABE des KBA. [o V23a]

2 Stand der Technik bei Abbiegeassistsystemen

Typbezeichnung	Genehmigungsgeber	Technischer Ansatz
ProFleet Assist+ Gen 2	Knorr-Bremse NFZ GmbH	Kamerasytem
ICA Turn-Assist XR45	AXION AG	Radarbasiertes System
ICA Turn-Assist AI40	AXION AG	Kamerasytem
ABE 100 ID	Autel Europe GmbH	Radarbasiertes System
LUIS TURN DETECT 4.0	LUIS Technology GmbH	Kamerasytem
TurnCAM-KI	ROSHO AS GmbH	Kamerasytem

Tabelle 2.1: Abbiegeassistenten, die über eine ABE des KBA verfügen Stand (04.04.2023)

Diese werden in der Tabelle 2.1 inklusive Typbezeichnung, Genehmigungsgeber und technischem Ansatz aufgeführt. Darüber hinaus veröffentlichte das KBA Publikationen von neun Abbiegeassistsystemen die keine ABE aufweisen, jedoch die Anforderungen an die Funktion von Abbiegeassistsystemen Nummern 2.1 bis 2.9 im Verkehrsblatt 2022, Heft 8, Nr. 65 des Bundesministeriums für Digitales und Verkehr erfüllen. Davor sind sechs Systeme kamerabasiert, drei radarbasiert und ein Assistent verfügt über die Kombination beider Ansätze. In Summe weisen 10 Abbiegeassistenten eine Kamera als zentrales Bauelement und fünf ein Radarortungssystem auf. Damit ist eine Tendenz zum optischen Prinzip erkennbar. Eine Bewertung, dass sich ein Ansatz gegenüber dem anderen durchgesetzt hat, kann nicht getroffen werden.

„Am 15. November 2019 ist die UN-Regelung Nr. 151 über einheitliche Bedingungen für die Genehmigung von Kraftfahrzeugen hinsichtlich des Totwinkel-Assistenten zur Erkennung von Fahrrädern völkerrechtlich in Kraft getreten. Die UN-Regelung Nr. 151 legt Anforderungen für Abbiegeassistsysteme fest, die aufgrund der EU Typ- Genehmigungsvorschriften für Nutzfahrzeuge mit einer zulässigen Gesamtmasse > 3,5 Tonnen und Kraftomnibusse ab 6. Juli 2022 für neue Fahrzeugtypen und ab 7. Juli 2024 für alle neuen Fahrzeuge verpflichtend vorgeschrieben sind“[Gui22]. Der aktuelle Stand der Technik ist in der UN-Regelung Nr. 151 niedergeschrieben. Eine Zusammenfassung ist im Verkehrsblatt 2022, Heft 8, Nr. 65 des Bundesministeriums für Digitales und Verkehr aufgeführt. Nachfolgend werden die technischen Anforderungen beschrieben. Die Nummerierung ist dem Originaldokument entsprechend übernommen.

2.1: Das System muss einen Abdeckungsbereich im Abstands von $0,9 \text{ m} < y < 3,5 \text{ m}$ seitlich des Lkw überwachen. In Längsrichtung ist die Abdeckung im Bereich $-2 \text{ m} < x$

< 9 m von der vorderen rechten Ecke des Kraftfahrzeugs ausgehend, zu gewährleisten.
[Gui22]

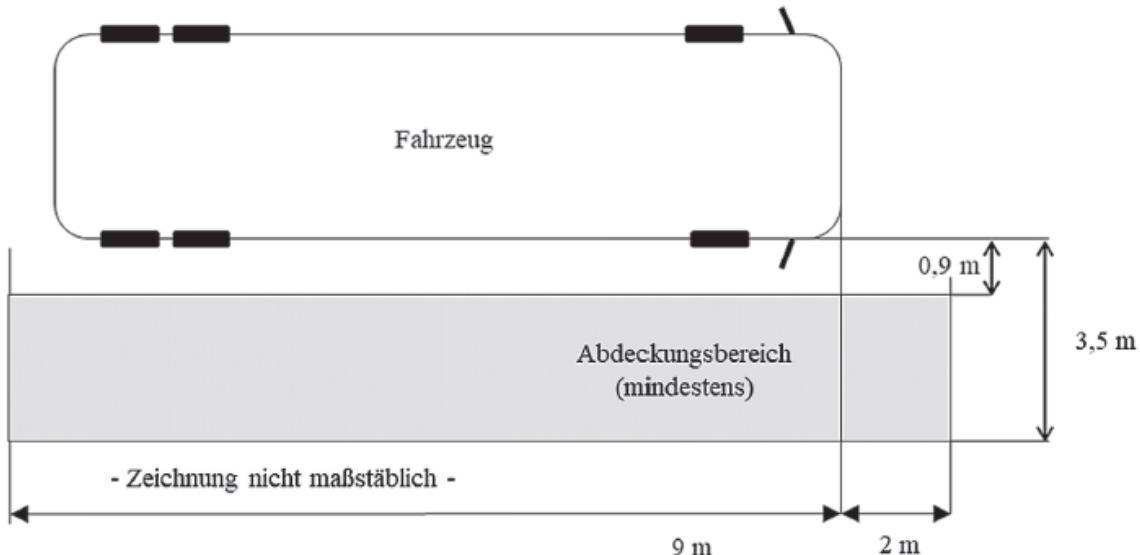


Abbildung 2.1: Anforderung an den Abdeckungsbereich von Abbiegeassistenten
[Gui22]

2.2: „Eine optische Signalisierung von gefährdeten Radfahrenden, die sich in gleicher Richtung bewegen, muss mindestens im Abdeckungsbereich erfolgen, sobald sich ein Teil des Fahrrades oder des Radfahrenden (was zuerst in den Bereich eintritt) im Abdeckungsbereich befindet (unabhängig von der Eintrittsrichtung). Die Signalisierung muss bei Fahrzeuggeschwindigkeiten mindestens im Bereich vom Stillstand bis $30 \frac{\text{km}}{\text{h}}$ und bei Fahrradgeschwindigkeiten mindestens im Bereich von $5 \frac{\text{km}}{\text{h}}$ bis $20 \frac{\text{km}}{\text{h}}$ im Innenraum des Kraftfahrzeuges erfolgen.“ [Gui22]

2.3: „Bei erhöhtem Kollisionsrisiko muss zusätzlich zur Signalisierung eine optische, akustische oder haptische Warnung im Innenraum erfolgen, sofern kein Stillstand des Fahrzeugs detektiert wird und eine potenzielle Kollision erkannt wird.“ [Gui22]

2.4: „Die Signalisierung darf nicht abhängig von der Aktivierung der Fahrtrichtungsanzeiger sein. Einstellbare Signalisierung oder Warnung sind unzulässig.“ [Gui22]

2.5: Das System ist gegen Manipulation z. B. Abschaltung durch den Lkw-Fahrer zu

schützen. [Gui22]

2.6: Eine Warnung ist ausschließlich bei bewegten Objekten auszulösen, wenn diese als schwächere Verkehrsteilnehmer identifiziert wurden.[Gui22]

2.7: „Eine Signalisierung oder Warnung auf statische Objekte, mit denen eine bevorstehende Kollision erkannt wurde, ist erlaubt.“[Gui22]

2.8: Der Monitor zur Visualisierung des Kamerabilds ist bei Linksslenkern mindestens 30° rechts von der Blickachse des Lkw-Fahrers anzubringen.[Gui22]

2.9: „Es muss für den Lkw-Fahrer jederzeit klar erkennbar sein, ob das Abbiegeassistsystem betriebsbereit ist und fehlerfrei funktioniert. Eine eingeschränkte Funktionalität, eine Fehlfunktion, z. B. Sensorsausfall oder -verdeckung, defekte Signalisierung oder Warnung, Teil- oder Komplettausfall des Systems, muss angezeigt werden. Falls eine dauerhafte Anzeige im Fehlerfall nicht möglich ist, kann alternativ eine Fehlermeldung angezeigt werden, die vom Fahrenden bestätigt werden muss. Die Funktion der Anzeige des Betriebsstatus ist in der Bedienungsanleitung zu beschreiben.“[Gui22]

3 Vorstudienergebnisse der Untersuchung von Eigenschaften der Hardware XPLR-AOA-1

[Tim22] schlägt vor, einen intelligenten Fahrradhelm zur Unfallprävention auf Grundlage der u-Blox Hardware XPLR-AOA-1 aufzubauen. [Den23] untersucht die Performance des Systems in einer Vorstudie und validiert auf Grundlage der Untersuchungen die Eignung der Hardware zur Einfallswinkelbestimmung eines Bluetooth Low Energy (BLE) Signals. Neben der Signal-Einfallswinkelgenauigkeit werden die Robustheit bei Störeinflüssen und die räumliche Auflösung der Messwerterfassung bei bewegtem Sender bzw. Empfänger betrachtet.

3.1 Genauigkeit der Signal-Einfallswinkelbestimmung

Die Genauigkeit der Signal-Einfallswinkelbestimmung bei Freiraumausbreitung wird systematisch in der Messkampagne Freiraumverhalten untersucht. Der Messaufbau sieht einen ortsfesten Empfänger, sowie einen, für den Zeitraum der Messwertaufnahme fest verorteten Sender vor. Der Sender wird systematisch in seiner x - und y -Position gegenüber dem Empfänger verschoben. Damit wird eine Aussage bezüglich des abgedeckten Winkelbereichs und Winkelfehlers in Abhängigkeit vom y -Abstand und dem wahren Signal-Einfallswinkel θ angestrebt. Tabelle 3.1 zeigt die Ergebnisse der Messkampagne auf. Die Zeilen sind dabei in den y -Messbereich $2 \text{ m} < y < 8 \text{ m}$ unterteilt. Die Spalten bilden den wahren Signal-Einfallswinkel im Bereich $-75^\circ < \theta < 75^\circ$ ab. Die Zelleinträge entsprechen jeweils der Winkelabweichung $\Delta\theta$ in Abhängigkeit des Abstands y . Sie werden mit der Berechnungsvorschrift des mittleren absoluten Fehlers (mean absolute error (MAE)), siehe Gleichung 3.1 über jeweils $n = 100$ Messwerte bestimmt, wobei θ_{wahr} dem wahren Wert des Signal-Einfallswinkels entspricht und θ_{mess} dem gemessenen.

$$MAE = \frac{1}{n} \sum_{i=1}^n |\theta_{wahr} - \theta_{mess}| \quad (3.1)$$

θ_{wahr}	-75°	-60°	-45°	-30°	-15°	0°	15°	30°	45°	60°	75°
$\Delta\theta$ bei $y = 2$ m	2,5°	0,3°	3,8°	7,2°	4,9°	0°	7,3°	8,0°	1,3°	9,5°	13°
$\Delta\theta$ bei $y = 4$ m	1,4°	1°	4,8°	5,3°	7°	0°	6°	5,3°	4,3°	6,9°	15,2°
$\Delta\theta$ bei $y = 6$ m	0,8°	1,1°	2,8°	8,6°	7,1°	0,6°	5,2°	3,9°	2,1°	7,8°	2,1°
$\Delta\theta$ bei $y = 8$ m	4,6°	2,2°	6°	8,1°	4,9°	0,8°	2,9°	8°	2,7°	9,1°	19,8°
$\Delta\theta$ über alle y	2,3°	1,1°	4,3°	7,3°	6°	0,4°	5,4°	6,3°	2,6°	8,3°	12,5°

Tabelle 3.1: Winkelabweichungen vom wahren Signaleinfallsinkel in der Messkampagne Freiraumverhalten

Die arithmetischen Mittelwerte des wahren Signal-Einfallsinkels θ werden in Abbildung 3.1 für alle y -Abstände dargestellt. Der Durchschnitt aller Mittelwerte beträgt 5,1° und entspricht damit näherungsweise den Herstellerangaben von 5° [o V22a]. In der Messkampagne Freiraumverhalten wurden auffällig hohe Ungenauigkeiten im Bereich kleiner bzw. großer Signal-Einfallsinkel $75^\circ < |\theta|$ festgestellt. Damit wird die Erfassung des Signal-Einfallsinkels auf den Bereich $-75^\circ < \theta < 75^\circ$ begrenzt. Der qualitative Verlauf der Winkelabweichung wird durch das, in Gleichung 3.2 beschriebene Polynom sechster Ordnung, siehe Abbildung 3.1 ausgedrückt. Die Diskreten Werte entsprechen den Winkelabweichungen $\overline{\Delta\theta}$ aus der Tabelle 3.1. Die blau eingezeichnete Kurve entspricht dem, mithilfe von MATLAB erzeugten Polynom sechster Ordnung Gleichung 3.2 zur Approximation der diskreten Winkelabweichungen in Abhängigkeit vom wahren Signal-Einfallsinkel. Die orange eingezeichnete Kurve ist eine einfachere Näherung ebenfalls auf Grundlage eines Polynoms sechster Ordnung.

Winkelabweichung $\Delta\theta =$

$$\pm 2,21 \cdot 10^{-10}\theta^6 - 3,65 \cdot 10^{-9}\theta^5 - 1,73 \cdot 10^{-6}\theta^4 + 4,51 \cdot 10^{-5}\theta^3 + 0,0034\theta^2 - 0,068\theta + 3,6 \quad (3.2)$$

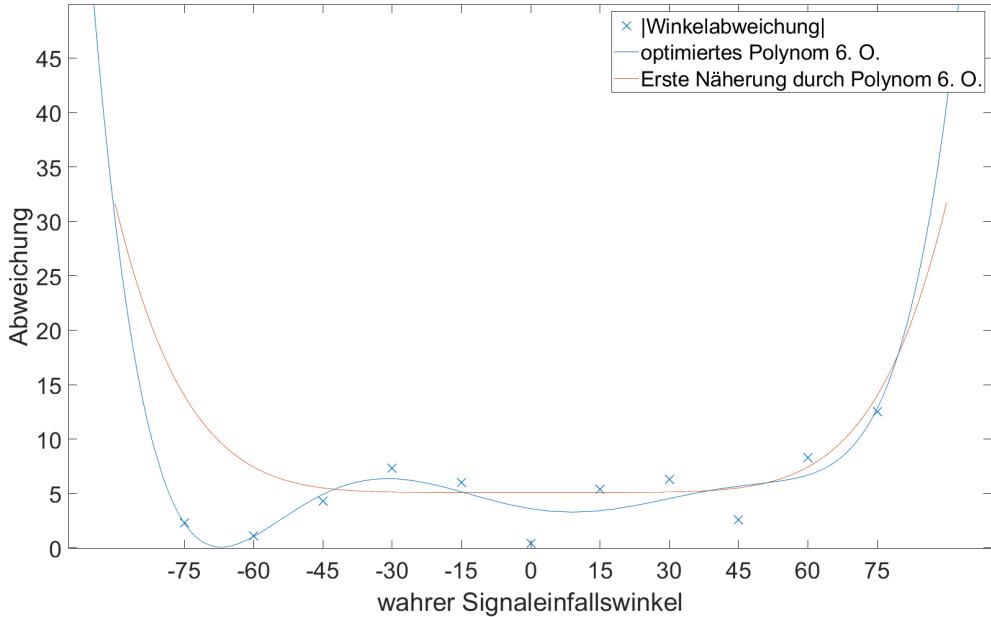


Abbildung 3.1: Approximation der Abweichungen vom wahren Signal-Einfallsinkel

3.2 Robustheit bei Störeinflüssen

Aus der Messkampagne Empfangsverhalten bei Hindernissen resultieren Erkenntnisse bezüglich der Robustheit. Der Einfluss kleiner Hindernisse z. B. Mülltonnen liegt im Mittel, in der vom Hersteller spezifizierten Hardware-Ungenaugkeit $\pm 5^\circ$. Große Hindernisse z. B. PKW (Personenkraftwagen) unterbinden eine zuverlässige Bestimmung des Signal-Einfallsinkels. Im urbanen Gebiet werden die BLE Signale häufig an Hindernissen z. B. Gebäuden oder Fahrzeugen reflektiert. Damit wird eine direkte Sichtverbindung (line-of-sight (LOS)) zwischen Sender und Empfänger unterbunden. Zudem resultiert ein Signal, welches den Empfänger über einen indirekten Pfad erreicht. In der Literatur wird dieser Pfad als non-line-of-sight (NLOS) bezeichnet [Car11]. Die Empfänger sind auf der maximal zulässigen Höhe am Lkw zu montieren, sodass eine LOS zwischen Sender und Empfänger nahezu durchgehend gewährleistet werden kann [Den23].

3.3 Räumliche Auflösung der Messwerterfassung

Die größte Gefahr einer Kollision, durch zu spätes bzw. Nicht-Erkennen des Fahrradfahrers durch den Lkw-Fahrer resultiert aus einer Annäherung des Fahrradfahrers von hinten rechts in Fahrtrichtung des Lkw. Zum Zeitpunkt des Lkw Abbiegevorgangs wird die maximale Relativgeschwindigkeit zwischen beiden Verkehrsteilnehmern mit $20 \frac{\text{km}}{\text{h}}$ angenommen. Diese resultiert aus einem stehenden Lkw und der Annäherung des Fahrradfahrers von hinten mit $20 \frac{\text{km}}{\text{h}}$. Die Aktualisierungsrate der Signal-Einfallswinkelbestimmung der Hardware XPLR-AOA-1 beträgt 40 Messwerte pro Sekunde [o V22a]. Damit folgt bei konstanter Relativgeschwindigkeit und linearer Annäherung eine räumliche Auflösung von circa $0,139 \frac{\text{m}}{\text{Messwert}}$. Der im Stand der Technik 2.1 beschriebene Fall, eines mit $5 \frac{\text{km}}{\text{h}}$ vorauselgenden Fahrradfahrers bei einer Annäherung des Lkw mit $30 \frac{\text{km}}{\text{h}}$ und somit der Fall der maximalen Relativgeschwindigkeit wird in dieser Arbeit nicht betrachtet, da der Fahrradfahrer in diesem Fall als sicher durch den Lkw-Fahrer erkennbar, angenommen wird.

$$20 \frac{\text{km}}{\text{h}} \cdot \frac{1}{3,6} \frac{\text{m} \cdot \text{h}}{\text{s} \cdot \text{km}} \cdot \frac{1}{40} \frac{\text{s}}{\text{Messwert}} = 0,139 \frac{\text{m}}{\text{Messwert}} \quad (3.3)$$

4 Systemdesign intelligenter Fahrradhelm Teilsystem Bluetooth Ortung

4.1 Systembeschreibung

Aufgabe des Prototyps intelligenter Fahrradhelm Teilsystem Bluetooth Ortung ist die Positionsbestimmung des intelligenten Fahrradhelms in einem, am Lkw verorteten Bezugskoordinatensystem. Auf Grundlage der Positionskoordinaten x und y wird die Gefahr einer möglichen Kollision zwischen den Verkehrsteilnehmern Fahrradfahrer und Lkw bestimmt. Die Lage des Fahrradhelms wird dem Lkw-Fahrer in Echtzeit angezeigt, so dass dieser Maßnahmen zur Vermeidung eines Zusammenstoßes einleiten kann. Die Visualisierung verfolgt den Anspruch eine intuitive Bewertung der Situation durch den Lkw-Fahrer zu ermöglichen.

Das kartesische Bezugskoordinatensystem in Abbildung 4.1 dient dem Zweck, die Position des intelligenten Fahrradhelms relativ zum Lkw zu definieren. Dieses wurde in [Den23] erstmalig definiert. Sein Ursprung ist in dem, an der vorderen, rechten Ecke des Lkw montierten, ersten Empfänger verortet. Die y -Achse entspricht dem neutralen Signaleinfallsinkel $\theta = 0^\circ$ und ist orthogonal zur langen Seite (Beifahrerseite) des Lkw ausgerichtet. Der Richtungsvektor der positiven x -Achse zeigt von der vorderen rechten Ecke zum Heck des Lkw. Anwendung findet das Bezugskoordinatensystem in der Visualisierung der Fahrradhelm-Position.

Nachfolgend werden die Grenzen beschrieben in, denen der Prototyp betrachtet wird. Zunächst werden die räumlichen Konstellationen der Relativbewegung von Lkw und Fahrradfahrer zueinander eingeschränkt. Das Ortungssystem erfasst den intelligenten Fahrradhelm ausschließlich rechtsseitig des Lkw. Die Gefahr einer Kollision wird aufgrund

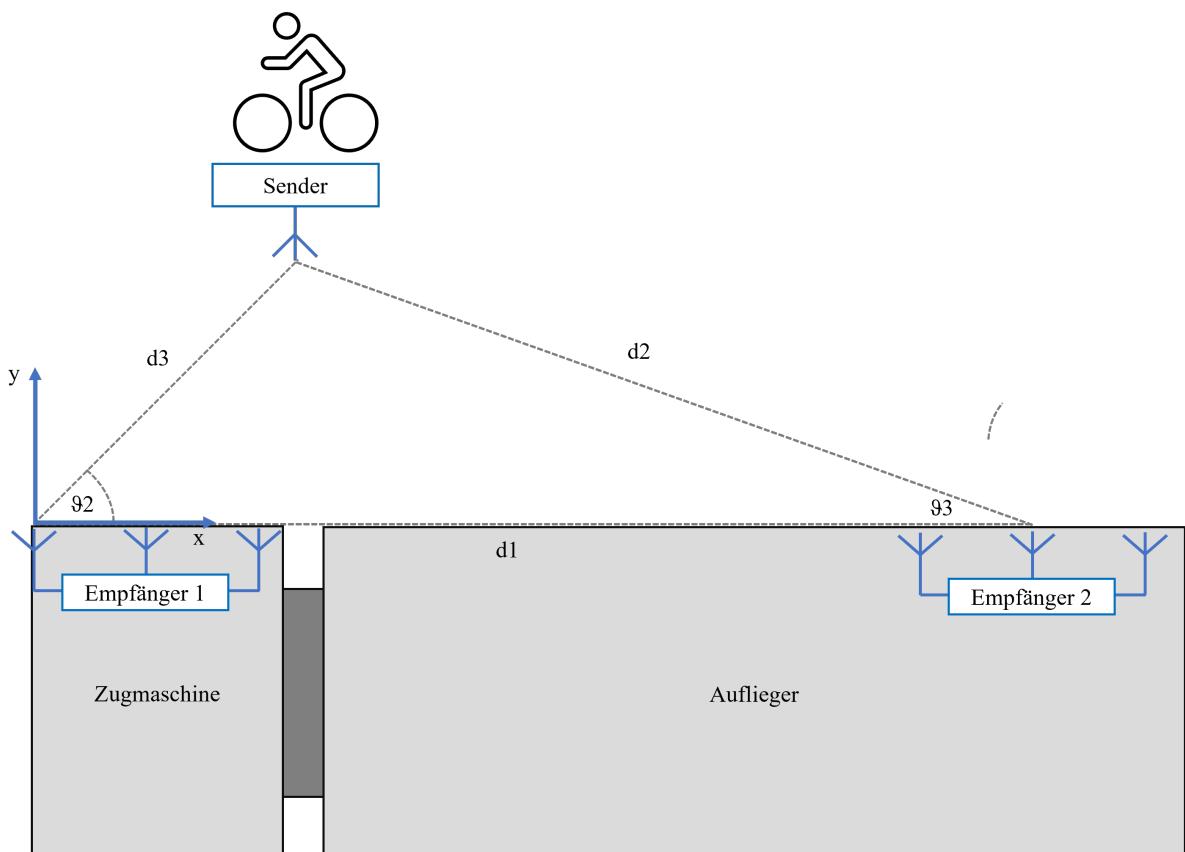


Abbildung 4.1: Bezugskoordinatensystem

des toten Winkels, siehe Abbildung 4.2 bei einem Rechtsabbiegevorgang des Lkw höher eingestuft. Es wird angenommen, dass der rot schraffierte tote Winkel linksseitig des Lkw durch Einsatz eines Schulterblicks vollständig eingesehen werden kann. Diese Annahme trifft für die rechte Seite nicht zu. Rechtsseitig gibt es je nach Konstruktion des Lkws, Spiegeleinstellung und Sitzposition des Fahrers einen nicht vernachlässigbaren toten Winkel. Des Weiteren wird eingeschränkt, dass die größte Gefahr aus einer Annäherung des Fahrradfahrers von hinten rechts in Fahrtrichtung des Lkw resultiert. Fahrradfahrer, welche sich von vorn auf den Lkw zubewegen, werden als sicher durch den Lkw Fahrer detektierbar angenommen und nicht weiter betrachtet. Nachfolgend wird diese Annahme belegt. Der Unfalltypenkatalog (UNKA) der Unfallforschung der Versicherer unterscheidet drei Arten von Rechtsabbiegeunfällen zwischen Lkw und Fahrradfahrern. Unfalltyp 243 beschreibt die Bewegung des Fahrradfahrers auf einem abgetrennten Radweg parallel und in gleicher Richtung wie der Lkw, bevor dieser nach rechts abbiegt. Der Unfalltyp 232 unterscheidet sich vom zuvor genannten dadurch, dass der Fahrradfahrer die Fahrbahn des Lkw mitbenutzt hat. Unfalltyp 244 liegt vor, wenn der Fahrradfahrer auf einem abgetrennten Radweg in entgegengesetzter Richtung zum Lkw fährt, bevor der Lkw nach rechts abbiegt. [DD16] Die Bundesanstalt für Straßenwesen (BASt) beschreibt in ihrem Heft F 104, dass gemäß German In-Depth Accident Study (GIDAS) 65 % aller Abbiegeunfälle dem Unfalltyp 243 zuzuordnen sind. Für diese Angabe wurden 122 Abbiegeunfälle zwischen Lkw und Fahrradfahrern im Zeitraum 1999 bis 2014 durch die Unfallforschung Dresden und Hannover in den jeweiligen Stadtgebieten sowie der umliegenden Kreise ausgewertet. Der Anteil des Typs 244 beträgt sechs Prozent. Unfalltyp 232 liegt in einem Prozent der Fälle vor. [SS15] Folglich ist der Typ 243 und damit der, in Fahrtrichtung des Lkw fahrende Fahrradfahrer von größter Relevanz. Eine zuverlässige Ortung wird unabhängig vom Unfalltyp mindestens für Abstände $< 3 \text{ m}$ gefordert. Dieser Nahbereich weist ein hohes Gefährdungspotenzial eines möglichen Zusammenstoßes auf. Eine Ortung des intelligenten Fahrradhelms über diesen Bereich hinaus ist eingeschränkt möglich. Die Genauigkeit variiert dabei signifikant mit der Position. Die verbaute Hardware ist prinzipiell in der Lage zeitgleich die Signale von bis zu 20 intelligenten Fahrradhelmen auszuwerten [o V22a]. Der Prototyp verfolgt vorerst ausschließlich das Ziel der Konzeptüberprüfung (Proof of Concept (PoC)) in der Funktion eines Abbiegeassistenten und ist ausschließlich zur Auswertung eines Senders ausgelegt.

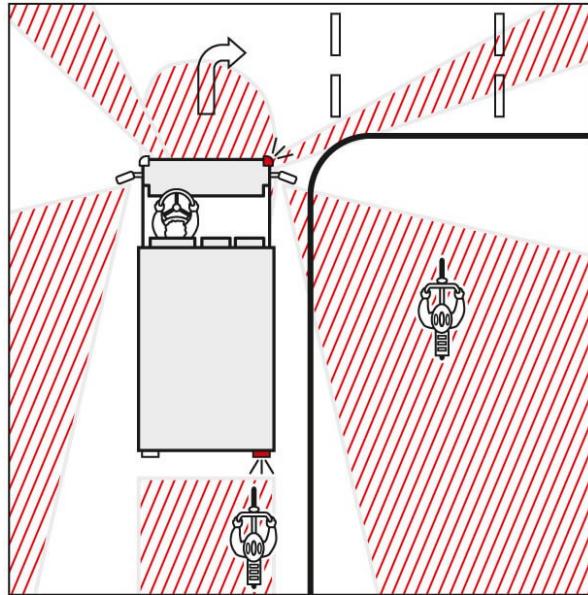


Abbildung 4.2: Darstellung des **toten Winkels** (rot schraffiert) [Pro19]

4.2 Freiheitsgrade zur Beeinflussung der Genauigkeit der Ortung

Die Ortung basiert auf der Methode der Triangulation, siehe Abbildung 4.3. Die Größen θ_2 und θ_3 entsprechen den Signal-Einfallswinkeln des ersten und zweiten Empfängers, bezogen auf das empfangene Signal des Senders. Der Abstand d_1 zwischen den Empfängern wird als bekannt vorausgesetzt. Aus den Signal-Einfallswinkeln θ_2 und θ_3 und der Innenwinkelsumme 180° lässt sich θ_1 nach der Vorschrift $\theta_1 = 180^\circ - (\theta_2 + \theta_3)$ berechnen. Je nach Bezugskoordinatensystem werden der Abstand d_2 gemäß Gleichung 4.1 oder d_3 gemäß Gleichung 4.2 berechnet. Im vorliegenden Fall wird der Sender im Koordinaten- system des ersten Empfängers und damit die Länge d_3 bestimmt. Schlussendlich werden die Koordinaten x und y des Senders im Bezugskoordinatensystem mit den Gleichungen 4.3 und 4.4 berechnet.

$$d_2 = \frac{d_1 \cdot \sin(\theta_2)}{\sin(\theta_1)} \quad (4.1)$$

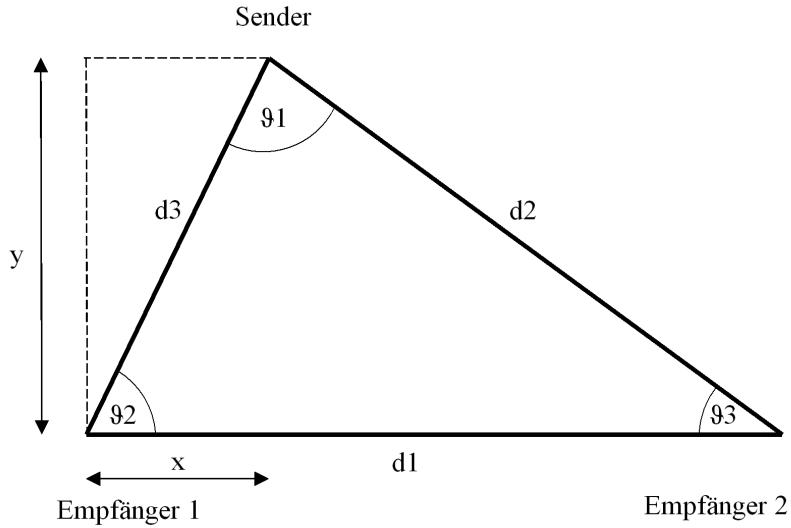


Abbildung 4.3: Schematischer Aufbau der Triangulation

$$d_3 = \frac{d_1 \cdot \sin(\theta_3)}{\sin(\theta_1)} \quad (4.2)$$

$$x = d_3 \cdot \cos(\theta_2) \quad (4.3)$$

$$y = d_3 \cdot \sin(\theta_2) \quad (4.4)$$

Kapitel 3.1 erklärt, dass die Signal-Einfallsinkel θ_2 und θ_3 im Empfangsbereich $-75^\circ < \theta < 75^\circ$ Winkelabweichungen aufweisen. Diese werden über ein Polynom, siehe Gleichung 3.2 approximiert. Aus den Winkelabweichungen resultiert eine Ungenauigkeit bei der Ortung des Senders, siehe Abbildung 4.4. Die wahre Position des Senders liegt im Schnittpunkt der blau dargestellten Geraden. Die rot eingefärbten Geraden stellen jeweils die untere und obere Abweichung $\theta_{wahr} \pm \Delta\theta$ dar. In Abbildung 4.4 betragen $\theta_2 = 45^\circ$, $\theta_3 = -45^\circ$ und $d_1 = 3$ m. Die Bezugsachse $\theta = 0^\circ$ ist für jeden Empfänger eine Senkrechte parallel zur y -Achse und wird vom Hersteller der Hardware XPLR-AOA-1, u-Blox definiert. Eine Gefährdung infolge der Ungenauigkeit resultiert, wenn der Fahrradfahrer weiter entfernt vermutet wird, als seine wahre Position. In vertikaler Richtung wird die Ungenauigkeit über den Abstand $distance_y$ approximiert, horizontal

über $distance_x$. Die Ungenauigkeit der Ortung beträgt in der Beispielkonfiguration in beiden Dimensionen horizontal und vertikal circa. 0,23 m.

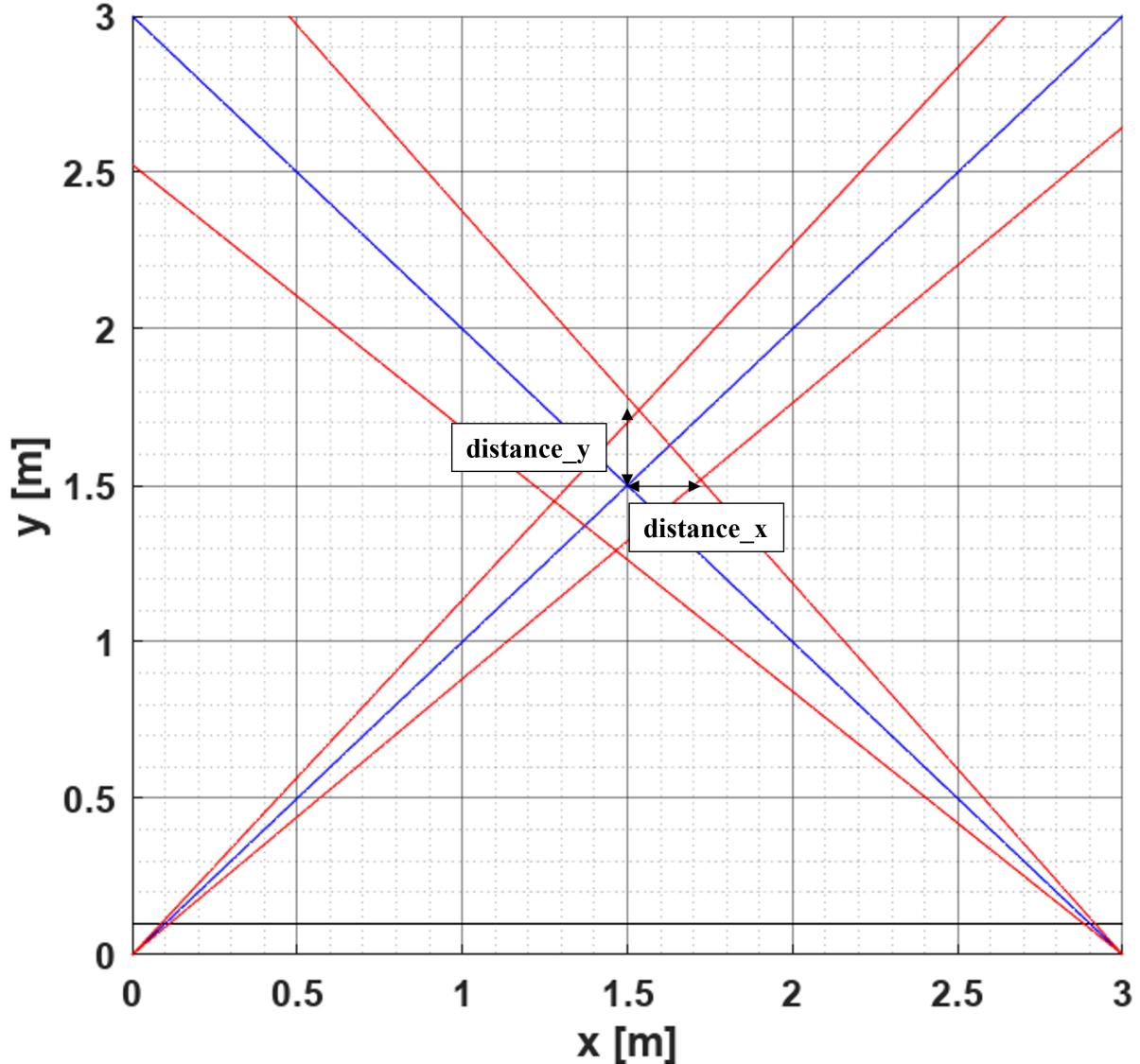


Abbildung 4.4: Visualisierung der Ungenauigkeit der Ortung

Abbildung 4.5 zeigt die Ungenauigkeit der Ortung für geänderte Stellgrößen. Die Länge d_1 beträgt hierbei 1,5 m, $\theta_2 = 32^\circ$ und $\theta_3 = -5^\circ$. Das Ausmaß der Ungenauigkeit liegt bei circa 0,15 m in horizontaler Richtung und circa 0,76 m in vertikaler Richtung. Damit nimmt die Ungenauigkeit der Ortung für diese Konstellation in vertikaler Richtung signifikant zu.

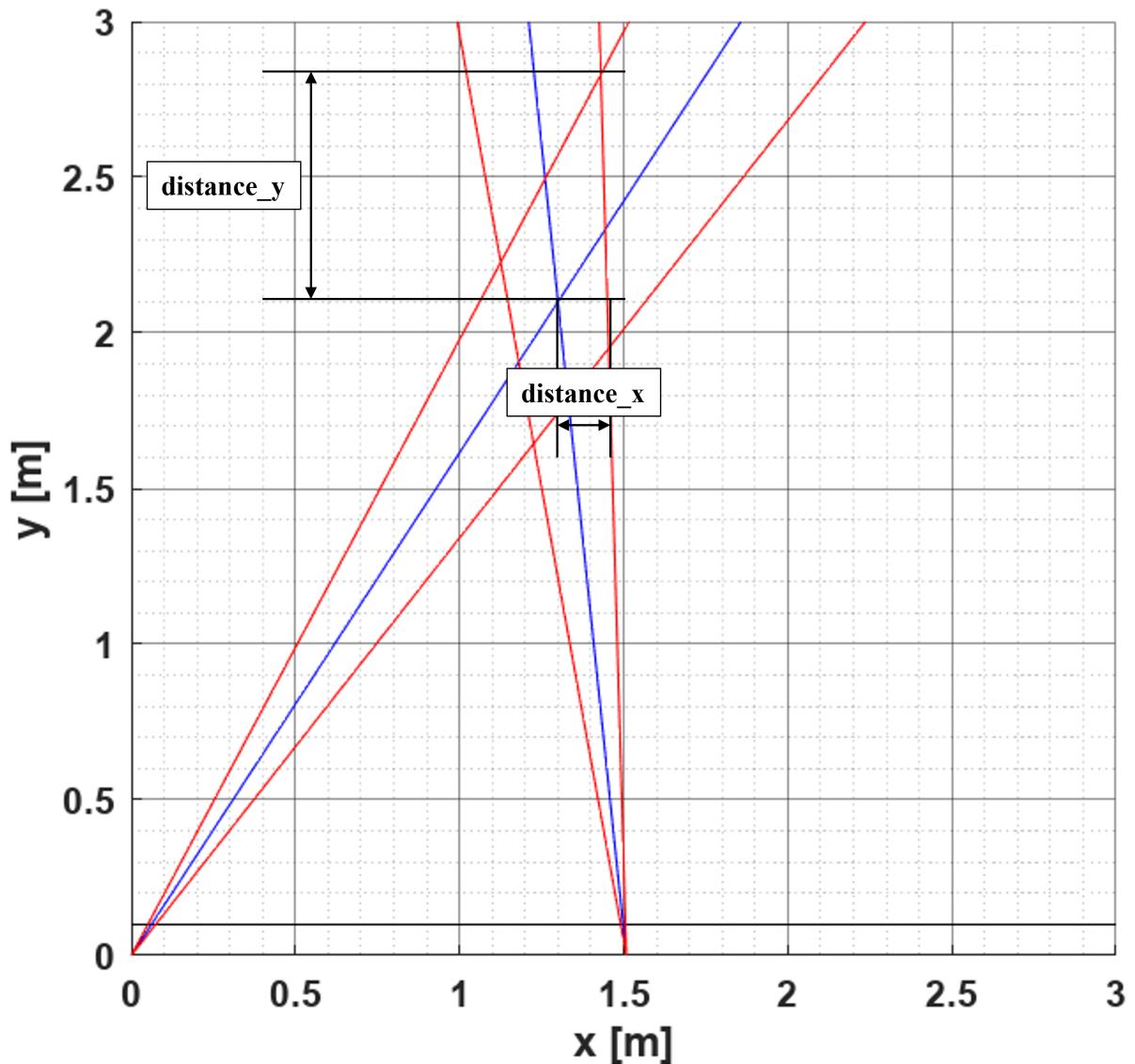


Abbildung 4.5: Visualisierung der Ungenauigkeit der Ortung für eine geänderte Lage des Senders und eine Variation des Abstands d_1

Die Größe der Winkelabweichung hängt gemäß Gleichung 3.2 vom Signal-Einfallswinkel θ und damit von der Position des Senders relativ zu den Empfängern ab. Das Ausmaß der Ungenauigkeit der Ortung resultiert aus den Winkelabweichungen und aus der Position des Senders relativ zu den Empfängern. Die Länge d_1 und die Orientierung (Rotation) der Empfänger-Koordinatensysteme $\theta_{2_{offset}}$, $\theta_{3_{offset}}$ werden als Stellgrößen bzw. Freiheitsgrade zur Beeinflussung der Ungenauigkeit der Ortung definiert.

4.3 Algorithmus zur Identifizierung der optimalen Stellgrößen

Für eine bestimmte Position des Senders in einem festen Bezugskoordinatensystem kann der Unsicherheitsbereich der Ortung durch Variation der Stellgrößen d_1 , $\theta_{2_{offset}}$ und $\theta_{3_{offset}}$ minimiert werden. Ziel des in MATLAB implementierten numerischen Berechnungsalgorithmus ist die Identifizierung von Stellgrößen, bei denen die Unsicherheit der Ortung im gesamten Erfassungsbereich auf ein Gesamtminimum ausbalanciert wird. Zunächst wird das Betrachtungsgebiet um den Lkw herum eingeschränkt. Der Betrachtungsbereich beginnt in horizontaler Richtung mit der vorderen rechten Ecke des Lkw und endet bei $x = 15$ m. In vertikaler Richtung wird der Bereich im Abstand zwischen $0 \text{ m} < y < 10 \text{ m}$ rechtsseitig des Lkw betrachtet. Diese Einschränkungen begründen sich in der Unfallgefahr, welche in der Nähe des Lkw für den Fahrradfahrer vorherrscht. Der Bereich vor dem Lkw erhält keine Betrachtung, da ausschließlich von hinten kommende Fahrradfahrer berücksichtigt werden. Größere Abstände in den Dimensionen x und y bieten keine Gefahr für eine Kollision und werden deshalb ebenfalls vernachlässigt. Der Bereich wird äquidistant in 0,5 m große Zellen unterteilt. Diese Zellgröße ist ein Abwägen (Trade-off) der Auflösung gegenüber dem Berechnungsaufwand des Algorithmus. Für jede Zelle werden die Stellgrößen d_1 , $\theta_{2_{offset}}$ und $\theta_{3_{offset}}$ iteriert und die dazugehörige Ungenauigkeit der Ortung in x - und y -Richtung berechnet. Die Ungenauigkeiten aller Zellen werden einzeln gewichtet und anschließend gemittelt. Für das Minimum aller gemittelten Ungenauigkeiten werden die dazugehörigen Stellgrößen d_1 , $\theta_{2_{offset}}$ und $\theta_{3_{offset}}$ anschließend ausgegeben. Je höher die Gewichtung einer Zelle ausfällt, desto stärker werden die Stellgrößen auf diese Zelle angepasst. Der Gewich-

tungsfaktor setzt sich aus zwei Teilstufen zusammen. Der erste Teilstufen repräsentiert die Unfallgefahr, welche in einem Zellbereich vorliegt. Die Unfallgefahr wird höher bewertet, je geringer der Abstand zwischen Lkw und Fahrradfahrer in x - bzw. y -Richtung ist. Der zweite Teilstufen bewertet die Sichtbarkeit eines Zellbereichs für den Lkw-Fahrer über die optischen Einrichtungen Fenster und Spiegel, siehe Abbildung 4.6. Abbildung 4.7 zeigt die Sichtbereiche in der Umgebung des Lkw. Grün umrandet sind die Bereiche, welche über die Fenster bzw. den Hauptspiegel eingesehen werden können. Die Verwendung der Spiegel mit Weitwinkel Optik deckt zusätzlich den gelb umrandeten Bereich ab. Rot dargestellt ist der, als totter Winkel bezeichnete, nicht einsehbare Bereich. Die Abgrenzung der Gebiete erfolgte am Beispiel eines Lkw vom Fabrikat Mercedes-Benz Arocs mit einem Eigengewicht von 12 t, einer Länge von 7 m und einer Breite von 2,5 m. Die Spiegel sind auf einen 1,80 m großen Fahrer eingestellt. Das sicher zu detektierende Versuchsstück soll exemplarisch ein Fahrrad-fahrendes Kind darstellen und weist eine Höhe von 1 m auf. Die Bemessung der Bereiche erfolgte bei Eintreten des Versuchsstücks in den jeweiligen Sichtbereich des Fahrers in den Spiegeln bzw. Fenstern. Diese Größe der Sichtbereiche ist der Abbildung 4.8 zu entnehmen. Die Kombination beider Teilstufen ergibt Gewichtungsfaktoren, welche in der Abbildung 4.9 aufgeführt sind. Je größer die Unfallgefahr eingeschätzt wird, bzw. je schlechter der Bereich einsehbar ist, desto höher fällt der Gewichtungsfaktor aus. Eine hohe Gewichtung wird insbesondere im Bereich der Schleppkurve des Lkw wirksam.

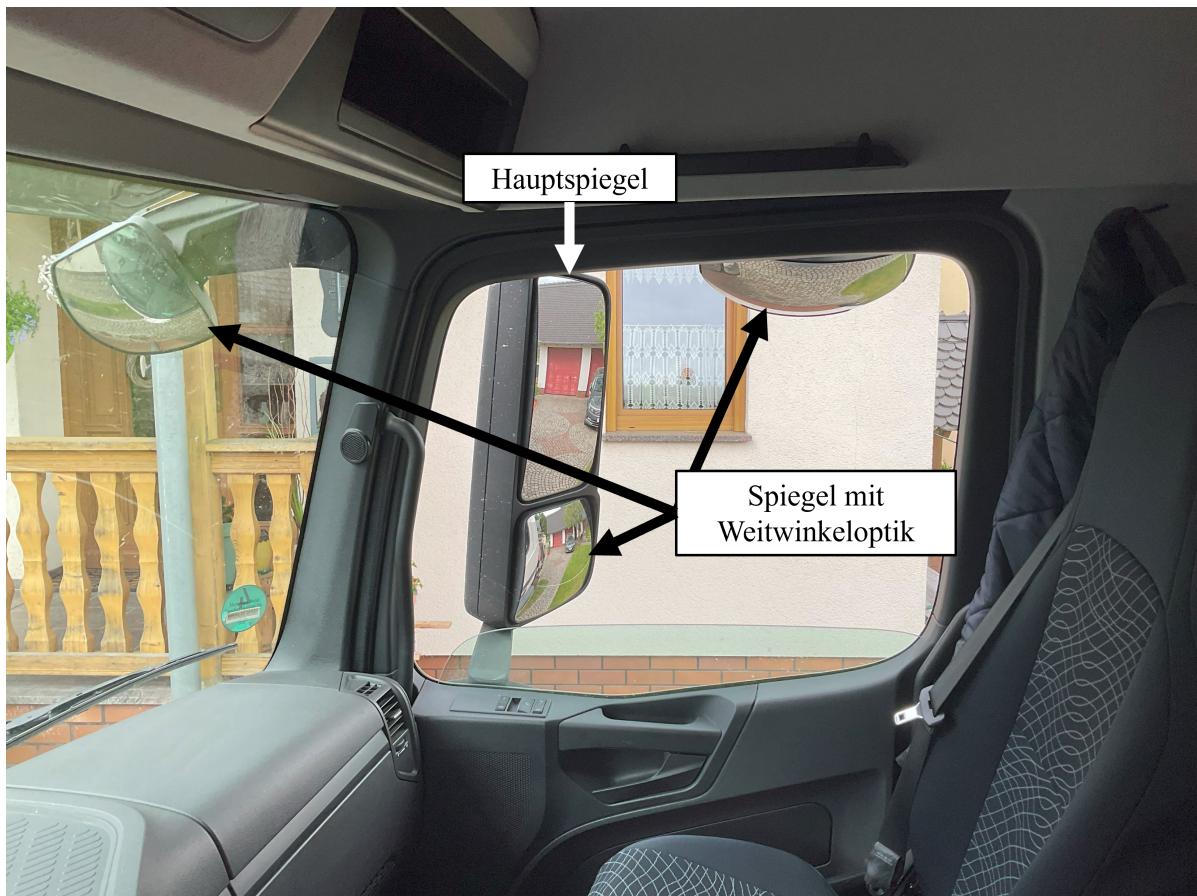


Abbildung 4.6: Spiegel am Lkw

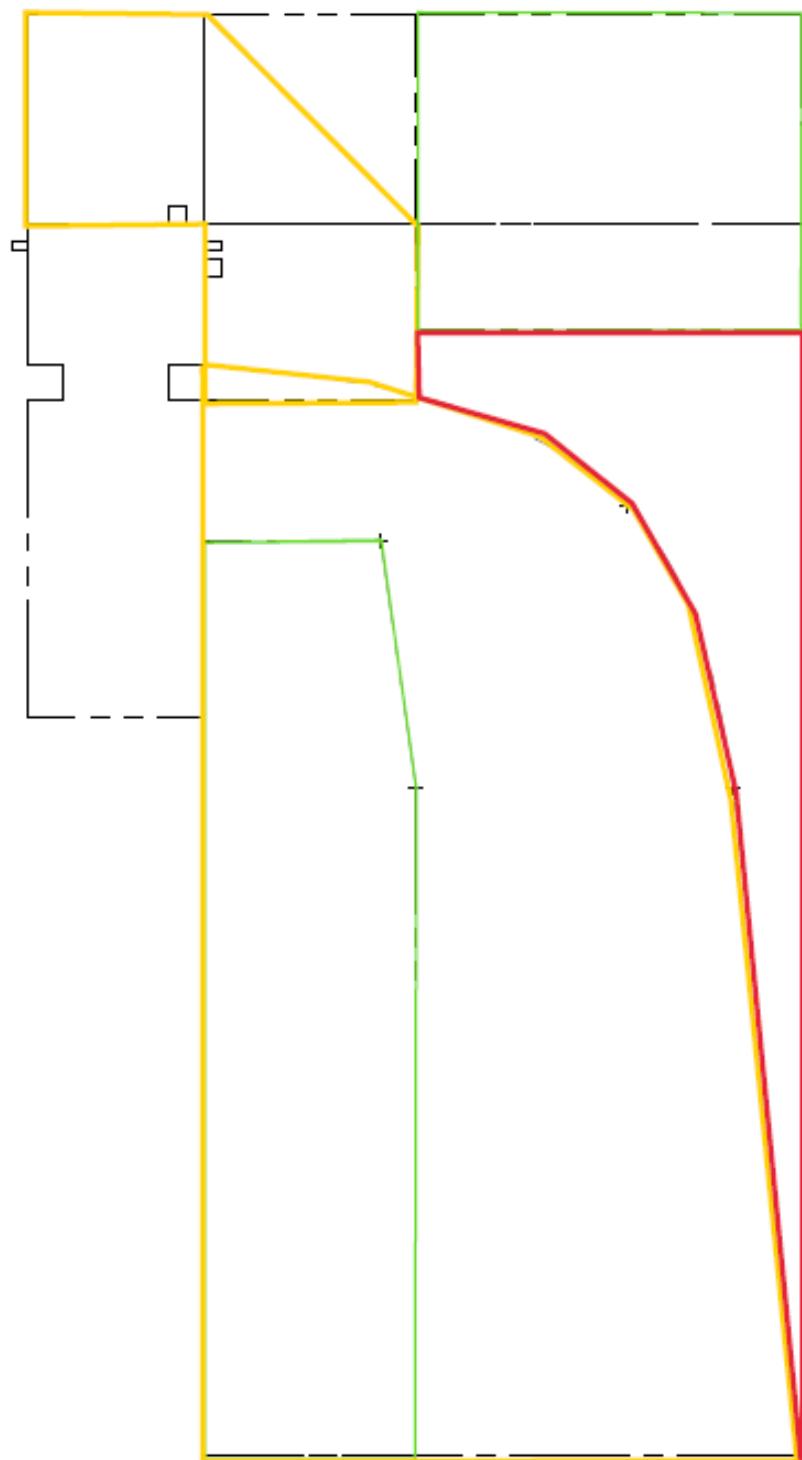


Abbildung 4.7: Unterscheidung der Sichtbereiche in der Lkw Umgebung

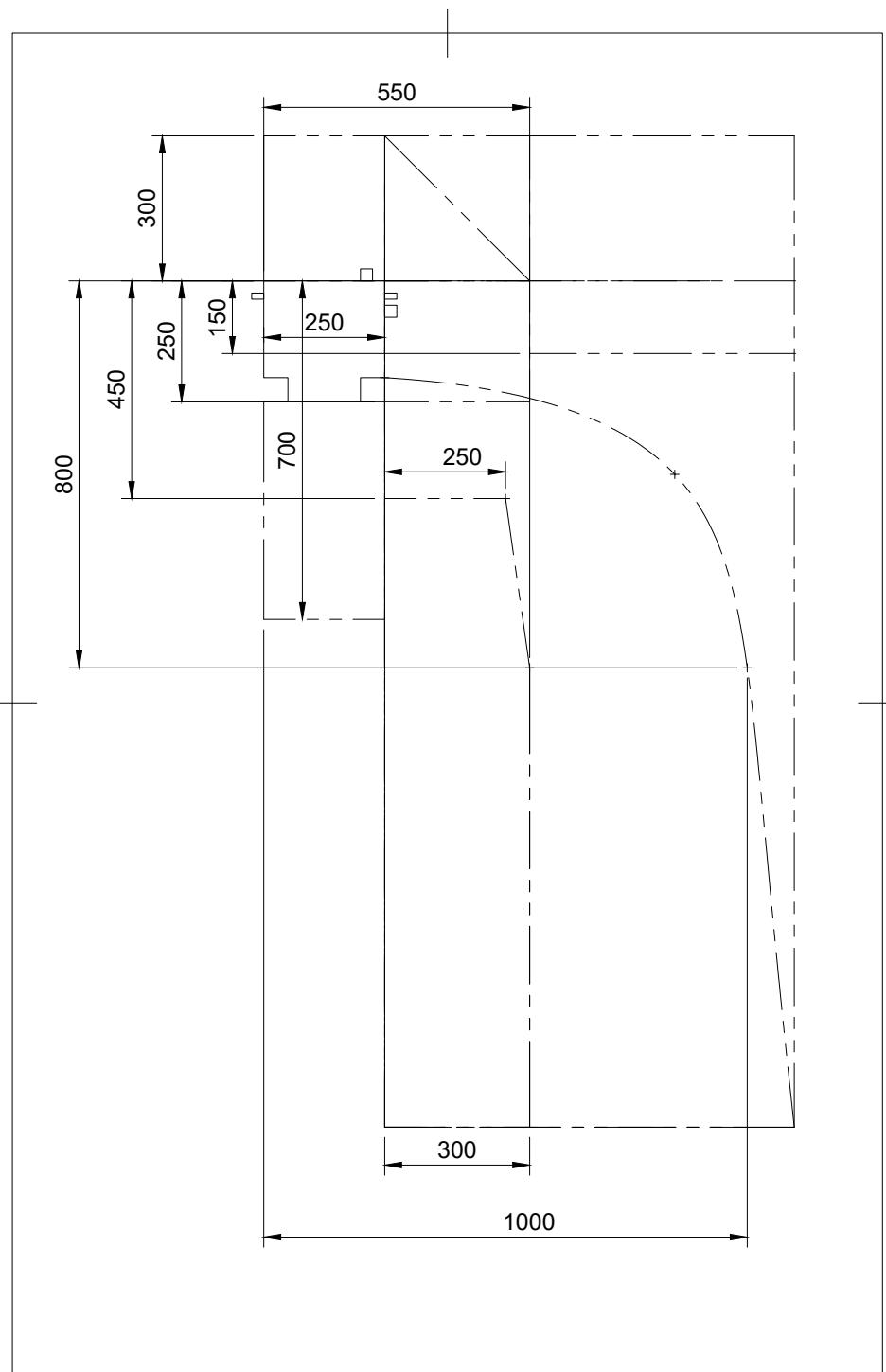


Abbildung 4.8: Bemaßung der Sichtbereiche in der Lkw Umgebung

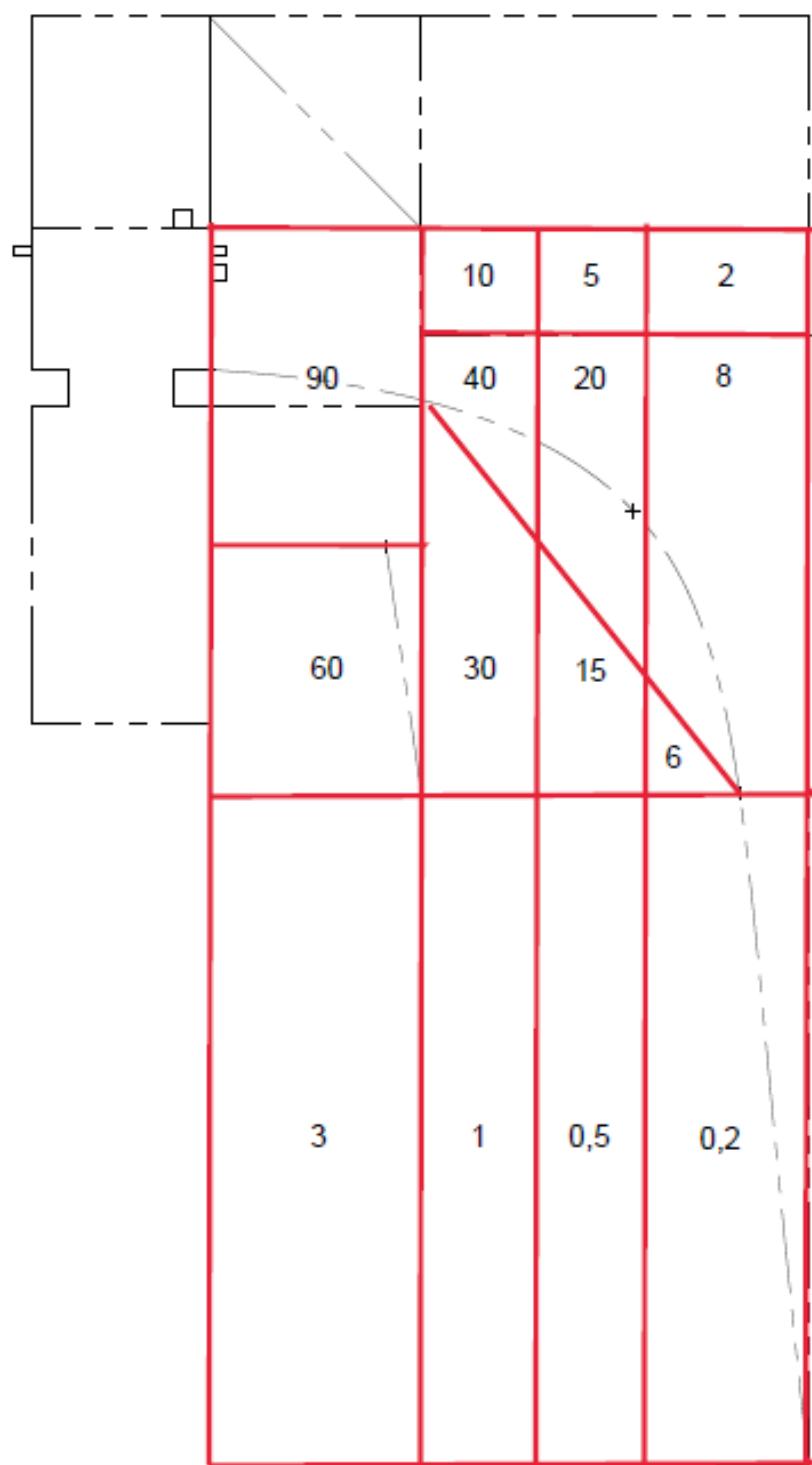


Abbildung 4.9: Gewichtungsfaktor kombiniert aus Sichtbereich und Unfallgefahr

Mithilfe des Algorithmus werden die Stellgrößen $d_1 = 5 \text{ m}$, $\theta_{2_{offset}} = 55^\circ$ und $\theta_{3_{offset}} = -10^\circ$ identifiziert. Abbildung 4.10 zeigt für diese Einstellung die zu erwartende, systematische Ungenauigkeit je Zelle auf. Die Farbdarstellung ist logarithmisch gewählt. Kleine Ungenauigkeiten werden hell dargestellt. Mit zunehmender Ungenauigkeit wird die Darstellung dunkler. Für Zellen, die schwarz eingefärbt sind, kann keine Ungenauigkeit berechnet werden, da kein Schnittpunkt der Geraden existiert, welche die Winkelabweichung repräsentieren. Im Nahbereich $y < 0,5 \text{ m}$ wird das System aufgrund der Ungenauigkeit von bis zu $2,57 \text{ m}$ keine zuverlässigen Messwerte liefern. Im Bereich $0,5 \text{ m} < y < 3 \text{ m}$ und $0 \text{ m} < x < 6 \text{ m}$ beträgt die Ungenauigkeit $< 1 \text{ m}$. Hinsichtlich der Unfallgefahr ist dieser Bereich besonders kritisch. Für einen Prototyp wird die Genauigkeit der Ortung in diesem Bereich als ausreichend bewertet. In den Bereichen $y > 5 \text{ m}$ und $x > 6 \text{ m}$ nimmt die systematische Ungenauigkeit erkennbar zu. Hier sind keine zuverlässigen Ortungsergebnisse mit Hilfe der Hardware für die gegebenen Stellgrößen zu erwarten.

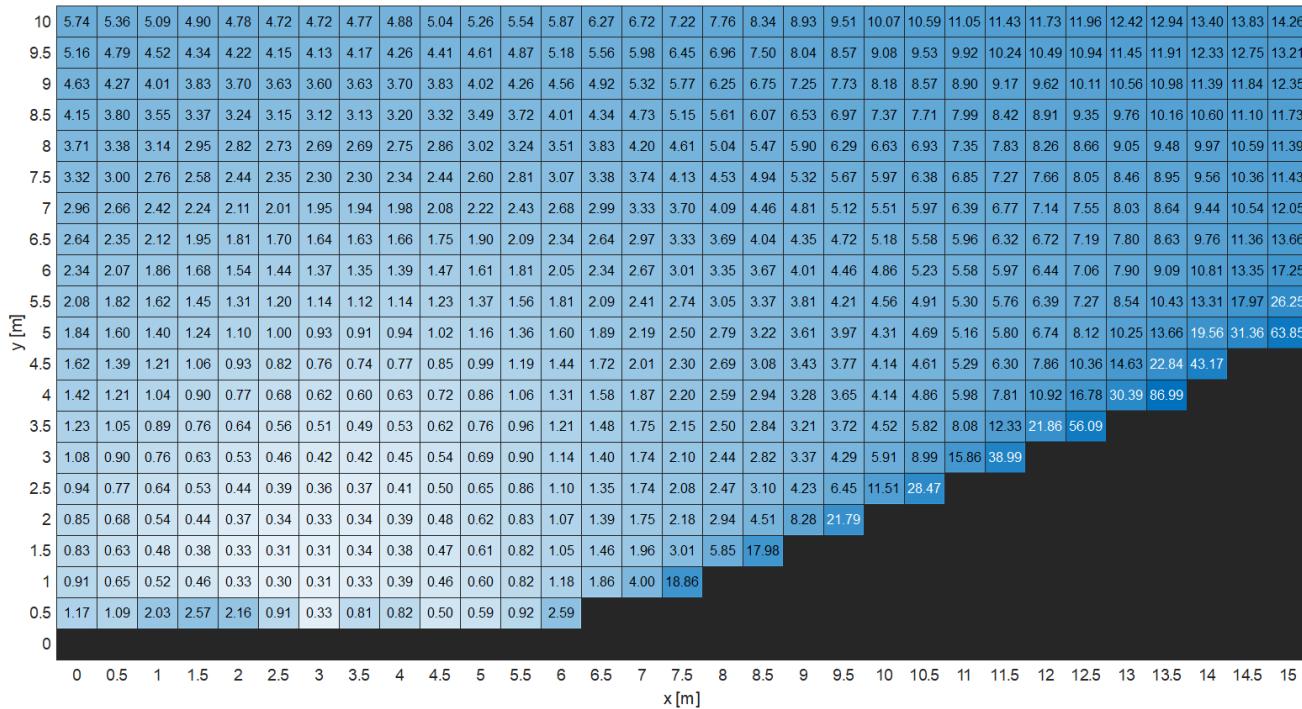


Abbildung 4.10: Ungenauigkeit der Ortung je Zelle bei Einstellung der Stellgrößen $d_1 = 5$ m, $\theta_{2_{offset}} = 55^\circ$ und $\theta_{3_{offset}} = -10^\circ$

4.4 Hardware

Das Ortungssystem des Prototyps basiert auf dem uBlox XPLR-AOA-1 explorer kit. Diese Hardware nutzt den Standard Bluetooth 5.1 mit der Funktion direction finding (Peilung) zur Bestimmung eines Signal-Einfalls winkels (Angle of Arrival (AoA)). Die Ortung erfolgt über die Methode der Triangulation. Dafür werden ein Sender und zwei Bluetooth 5.1 Low Energy Empfänger verwendet.

4.4.1 Empfänger

Der in Fahrtrichtung vorn verbaute Empfänger ist vom Typ XPLR-AOA-1-00 C211 und verfügt Leuchtdioden (light-emitting diode (LED)) zum Anzeigen der Signaleinfallsrichtung. Der im hinteren Bereich montierte Empfänger ist vom Typ XPLR-AOA-1-01 C211. Dieser hat keine eingebaute LED Anzeige. Die Empfänger setzen sich jeweils aus einem NINA-B411 Bluetooth 5.1 Low Energy Modul und einem Antennenarray zusammen. Jedes Antennenarray besteht aus fünf Antennen, über die der horizontale Signal-Einfalls winkel (Azimut) und der vertikale Signal-Einfalls winkel (Elevation) geschätzt werden. Der Prototyp verarbeitet ausschließlich den horizontalen Signal-Einfalls winkel, der im weiteren mit θ bezeichnet wird. Dieser wird auf Grundlage der, vom Sender übertragenden Nachrichten mit Constant Tone Extension (CTE) in den Empfängern berechnet. Die vom Hersteller implementierte u-connectLocate Software zur Auswertung der empfangenen Nachrichten und anschließender Berechnung läuft auf einer embedded micro controller unit (MCU) im NINA-B411 BLE Modul. Die Software führt eine Median Filterung¹ der empfangenen Daten durch. Damit werden Signale als Folge einer Mehrwegeausbreitung² unterdrückt. Die Auswertung der Daten beider Empfänger und Durchführung der Triangulation erfolgt auf einem Notebook (Host). Die Kommunikation der NINA-B411 Bluetooth 5.1 Low Energy Module mit dem Host erfolgt über

¹Verwendung des Medians einer definierten Anzahl empfangener Daten

²Signale werden mittels elektromagnetischer Wellen übertragen. Eigenschaften elektromagnetischer Wellen sind Reflexion und Beugung. Erreichen vom Sender ausgesendete Wellen die Empfänger auf indirektem Weg liegt eine Mehrwegeausbreitung vor [Dip06]

eine Universal Serial Bus (USB)³ Schnittstelle. Die Energieversorgung der NINA-B411 Bluetooth 5.1 Low Energy Module stellt der Host über die USB-Leitungen bereit. Bei Verwendung eines Host mit Windows Betriebssystem wird über eine Component Object Model- (COM) Schnittstelle auf die seriellen Daten des NINA-B411 Bluetooth 5.1 Low Energy Moduls zugegriffen. Die Daten des Empfängers werden in Form von Events mit der vorher eingestellten Baudrate an die COM-Schnittstelle übertragen. In der Standardeinstellung ist die Übertragung auf eine Baudrate von 115200 kbps und folgenden Protokollaufbau definiert: 8 data bits, no parity, 1 stop bit (8N1), Flow control enabled using Request to Send and Clear to Send (RTS/CTS). Die Empfänger sind durch Verbindung der Anschlusspins drei und fünf, bzw. vier und sechs mithilfe von Jumpern, siehe Abbildung 4.11 auf eine USB Verbindung eingestellt. [o V22b]

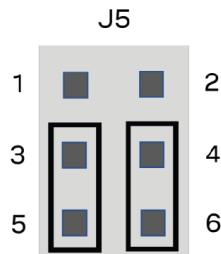


Abbildung 4.11: Jumper Einstellung für eine Kommunikation über USB [o V22b, S. 12]

4.4.2 Sender

Der Sender, siehe Abbildung 4.12 basiert auf einem NINA-B406 Prozessor (central processing unit (CPU)). Die CPU wird über eine Micro-USB-Verbindung oder eine CR2032⁴ Batterie mit Energie versorgt. In der Standardeinstellung sendet er Daten mit einer Frequenz von 50 Hz an die Empfänger. Die Intervall-Länge kann über den Schalter SW2 von 20 ms auf 100 ms bzw. 1000 ms verlängert werden. [o V22b]

³Schnittstelle zur bitseriellen Datenübertragung [o V21]

⁴Lithium-Batterie als Rundzelle

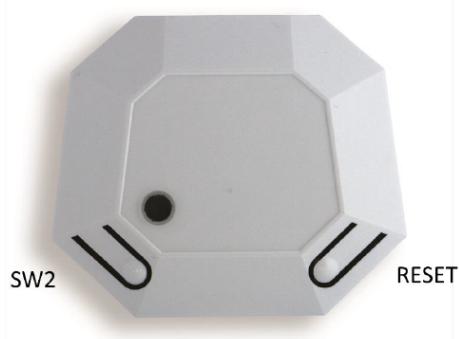


Abbildung 4.12: Sender C209 [o V22b, S. 21]

4.4.3 Halterung

Es ist geplant die Befestigung der Empfänger am Lkw über Halterungen auszuführen. An diese Halterungen wird die Anforderung gestellt, die Empfänger drehbar aufzuhängen, um eine Rotation um $\theta_{2_{offset}}$ bzw. $\theta_{3_{offset}}$ durchzuführen. Anforderungen an den Schutz gegen mechanische Einwirkung und Feuchtigkeit sind für den Prototyp nicht näher spezifiziert. Die Empfänger sind am Versuchsfahrzeug in einem Abstand von $d_1 = 5$ m zueinander anzubringen.

4.5 Software

Die Auswertung der Signal-Einfallswinkel θ zur Ortung des Senders sowie die Visualisierung der Position erfolgt mithilfe eines Python Scripts auf dem Host. Nachfolgend aufgeführte Code Listings beziehen sich exemplarisch auf den ersten Empfänger.

4.5.1 Konfiguration der seriellen Schnittstelle

Die Daten der Empfänger werden seriell über virtuelle COM-Schnittstellen auf dem Host empfangen. Diese werden zunächst initialisiert. Dafür wird das Betriebssystem übergreifende Modul pySerial genutzt. [Chr20] Die Baudrate beider COM-Schnittstellen ist auf

115200 Symbole pro Sekunde eingestellt. Die Variable timeout=0.05 gibt an, dass die Abfrage einer Schnittstelle nach 50 ms abgebrochen wird, wenn kein vollständiger Datensatz eingelesen wird. Dies stellt eine Fehlerbehandlung dar, die das zeitliche Verhalten des Aufrufs der Visualisierung begrenzt. Die Länge des Datensatzes wird über die PySerial Funktion Serial.read(<>, <size>) definiert. Der Platzhalter <size> gibt die Anzahl der einzulesenden Bytes an. Die Variable rtscts=1 aktiviert eine Datenflussteuerung. Hierzu wird auf den Standard für serielle Schnittstellen (Recommended Standard 232 (RS-232)) verwiesen. [Cam84]

```
1 serialdata_anchor1 = serial.Serial('COM3', 115200, timeout=0.05, rtscts  
=1)
```

Zur Vermeidung unerwünschter Programmabbrüche bei einem Aufruf bereits geöffneter Schnittstellen wird eine Ausnahmen Behandlung (exception handling) durchgeführt.

```
1 try:  
2     serialdata_anchor1.isOpen()  
3     print("port is opened!")  
4  
5 except IOError:  
6     serialdata_anchor1.close()  
7     serialdata_anchor1.open()  
8     print("port was already open, was closed and opened again!")
```

4.5.2 Eingangsdatenverarbeitung

Gemäß Datenblatt werden 40 AoA-Schätzungen pro Sekunde durchgeführt. Die entsprechenden Daten werden als sogenannte Angle calculation Events, gekennzeichnet durch den String +UUDF an die COM-Schnittstelle übertragen. Die Syntax der Events, siehe Tabelle 4.1 ist nachfolgend dargestellt.

Index	Parameter	Description
1	ed_instance_id	6-byte Eddystone instance id
2	rssi_pol1	RSSI of polarization 1
3	angle_azimuth	Azimuth angle in range -90° to 90°
4	angle_elevation	Elevation angle in range -90° to 90°
5	<not used>	
6	channel	Channel from which the packet angle was calculated
7	anchor_id	default value = MAC address
8	user_defined_str	User defined string that can be set to any value
9	timestamp_ms	Time since boot in milliseconds

Tabelle 4.1: Syntax der vom Empfänger übertragenen Events [o V22b, S. 28]

+UUID:<ed_instance_id>, <rssi_pol1>, <angle_azimuth>, <angle_elevation>, <rssi_pol2>, <channel>, <anchor_id>, <user_defined_str>, <timestamp_ms>

Das Einlesen einzelner Events +UUID an der virtuellen COM-Schnittstelle verzögert die Programmlaufzeit des Python Scripts signifikant. Stattdessen werden die seriellen Eingangsdaten in einem Eingangsspeicher gesammelt und über einen zyklischen Schleifenaufruf abgefragt. Die Zykluszeit entspricht der Programmlaufzeit des gesamten Python Scripts. Die entsprechende Abfrage laute:

```
1 if serialdata_anchor1.in_waiting() > 0:
```

Die PySerial Funktion `in_waiting()` liefert die Anzahl der Bytes, welche im Eingangsspeicher (input buffer) anliegen. Diese Form des Einlesens benötigt eine Sonderbehandlung, um die seriell eingelesenen Daten in ihre ursprünglichen Events zu zerlegen, sodass die Signal-Einfallsinkel θ des bewegten Senders in Echtzeit⁵ gelesen werden können. Die nachfolgende Funktion wandelt 80 Bytes der seriellen Eingangsdaten in einen String um. In den Standardeinstellungen entspricht die Länge des +UUID Events 60 Zeichen. Hinzu kommen Leerzeichen und Escape-Sequenzen⁶ zwischen den Events. 80 Byte haben sich in der Erprobung als praxistauglich erwiesen.

⁵Unter Echtzeitfähigkeit versteht man, dass auf ein Ereignis innerhalb einer genau definierten maximalen Zeitspanne reagiert wird - diese Zeitspanne wird Latenzzeit genannt. [WB05] Im vorliegenden Fall ist die Latenzzeit auf 25 ms definiert. Damit entspricht die Latenzzeit, der vom Hersteller angegebenen mittleren Zeitspanne zwischen zwei Events [o V22a]

⁶Zeichenkombination, die eine Sonderfunktion ausführen z. B. Zeilenvorschub

```
1 dataStream_anchor1 = str(serialdata_anchor1.read(80))
```

Mithilfe regulärer Ausdrücke (Regular Expression (regex))⁷ werden die seriellen Eingangsdaten in Events unterteilt. Über die regex Funktion re.split wird ein Array erzeugt, dass bei jedem Auftreten des Strings UUDF einen neuen Eintrag (listing) erhält. Über eine for-Schleife werden die Einträge durchlaufen und ihre Gültigkeit überprüft. Die Gültigkeitsprüfung umfasst die Abfrage der Media Access Code- (MAC) Adresse des Empfängers, damit die Signal-Einfallsinkel θ dem richtigen Empfänger zugeordnet werden. Über die Funktion split des Python Enhancement Proposal (PEP) 484 Standard Moduls typing werden gültige Events entsprechend der Syntax in ihre Bestandteile (parts) zerlegt. Anschließend wird die Vollständigkeit der Syntax überprüft und der am dritten Index hinterlegte Signal-Einfallsinkel θ inklusive der Rotation des Empfänger-Koordinatensystems abgespeichert.

```

1  regex_anchor1 = re.split("UUDF:", dataStream_anchor1)
2  for listing in regex_anchor1:
3      if "6C1DEBA0982F" in listing:
4          parts = listing.split(",")
5          if len(parts) == 9:
6              fifo_anchor1.put(int(parts[2])+theta2_offset)

```

Die Signal-Einfallsinkel θ werden in einem First In First Out (FIFO) Speicher verwaltet. Dafür wird das Python Modul Queue verwendet. Diese Art der Speicherung sichert ab, dass die Winkel in der richtigen Reihenfolge ausgelesen werden. Zudem wird die Möglichkeit vorgehalten die Daten sicher zwischen Prozessen (Threads) auszutauschen. Der nachfolgende Code initialisiert den FIFO Speicher.

```

1  fifo_anchor1: Queue[Any] = queue.Queue()

```

Die FIFO-Speicher der Empfänger werden zu unterschiedlichen Zeitpunkten befüllt und zum selben Programmzeitpunkt ausgelesen. Damit erfolgt eine Synchronisation der beide Empfänger. Folglich wird eine Triangulation auf Basis zusammengehöriger Signal-Einfallsinkel gewährleistet. Die Triangulation erfolgt auf Grundlage gemittelter Winkel θ_2 bzw. θ_3 mit dem Ziel die Visualisierung des bewegten Senders stabilisiert und

⁷Zeichenfolge, die über Syntaxregeln beschrieben und ausgewertet werden kann [San21]

ohne Ausreißer darzustellen. Je Empfänger wird eine Mittelwertbildung der letzten vier Signal-Einfallswinkel mit anschließender Ausreißer-Behandlung durchgeführt.

Damit eine Schätzung der Signal-Einfallswinkel zuverlässig durchgeführt werden kann ist eine LOS zwischen Sender und Empfänger notwendig. Diese kann im urbanen Bebauungsgebiet nicht durchgehend gewährleistet werden. Ein Vorteil der bewegten Systeme Sender und Empfänger ist jedoch der dynamische Wechsel zwischen LOS und NLOS. Winkel, die während eines NLOS Signalempfangs berechnet werden, weichen auffällig stark von den LOS Signal-Einfallswinkeln ab [Den23]. Folglich ist eine Bewertung der Winkel θ nach LOS bzw. NLOS Charakteristik durchzuführen. NLOS-Winkel werden als Ausreißer erkannt. Nachdem der Wert mit der größten Abweichung vom Mittelwert⁸ entfernt wird, erfolgt eine Berechnung des Mittelwerts der verbliebenen drei Einträge. Die Größe der FIFO-Einträge wird über die Queue Funktion qsize() abgefragt. Für die Berechnung des arithmetischen Mittelwerts mithilfe mathematischer Funktionen des Python Moduls math werden die Signal-Einfallswinkel θ aus dem FIFO Speicher in ein Array transferiert. Dies erfolgt über die Programmzeile records_anchor1.append(fifo_anchor1.get()). Die Queue Funktion get() entfernt die Einträge nach erfolgreichem Transfer aus dem FIFO Speicher. Über queue.Queue() wird der FIFO Speicher anschließend reinitialisiert.

```

1 if fifo_anchor1.qsize() >= 4 and fifo_anchor2.qsize() >= 4:
2     records_anchor1 = []
3     for i in range(4):
4         records_anchor1.append(fifo_anchor1.get())
5     arithmetic_mean_fifo_anchor1 = sum(records_anchor1) / len(
6         records_anchor1)
7     max_deviation_listing_anchor1 = max(records_anchor1, key=lambda x: abs(
8         x - arithmetic_mean_fifo_anchor1))
9     records_anchor1.remove(max_deviation_listing_anchor1)
10    arithmetic_mean_fifo_anchor1 = sum(records_anchor1) / len(
11        records_anchor1)
12    fifo_anchor1 = queue.Queue()

```

⁸Der Winkel mit der größten Abweichung vom Mittelwert entspricht dem Ausreißer.

4.5.3 Positions berechnung

Zunächst wird die Definition der Signal-Einfalls winkel θ_2 und θ_3 mit entsprechendem Vorzeichen in Abbildung 4.13 aufgezeigt. Der grüne Kreisbogen repräsentiert den Signal-Einfalls winkelbereich mit positivem Vorzeichen, der rote Kreisbogen stellt den Bereich eines Signal-Einfalls mit negativem Vorzeichen dar. Die Festlegung der Vorzeichen ist von der Hardware XPLR-AOA-1 vorgegeben. Für die Berechnung werden die Koordinatensysteme gemäß Abbildung 4.14 neu definiert. Die Positions berechnung über beide Signal-Einfalls winkel erfolgt ausschließlich über das, im linken Bereich dargestellte Koordinatensystem. Die Invertierung des Vorzeichens und Umrechnung der Winkelgrößen in das Bogenmaß wird durch den nachfolgenden Programm befehl realisiert.

```

1 theta2 = -arithmetic_mean_fifo_anchor1 * pi / 180
2 theta3 = -arithmetic_mean_fifo_anchor2_listing * pi / 180

```

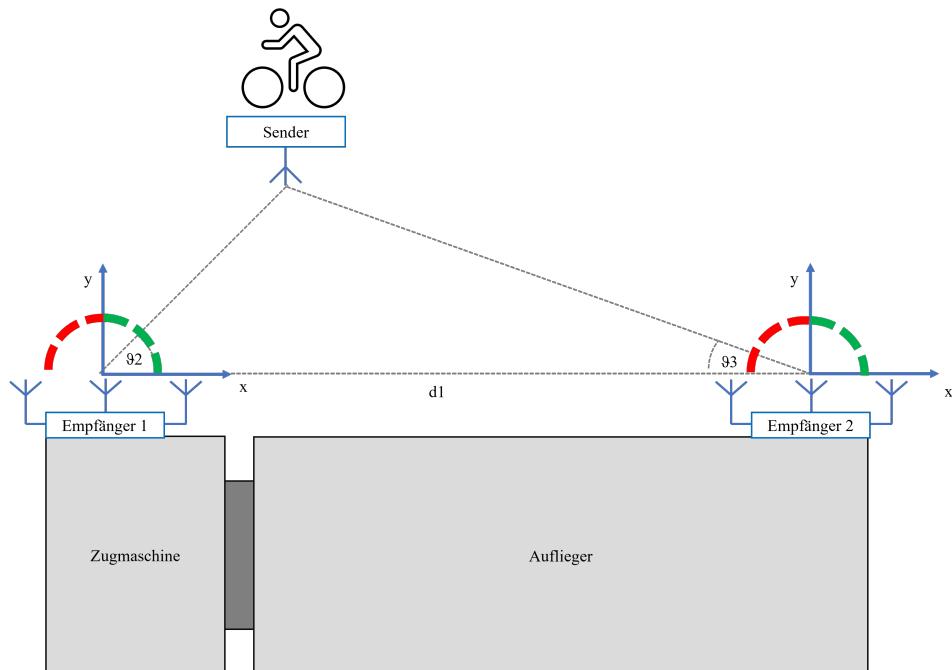


Abbildung 4.13: Definition der Empfänger-Koordinatensysteme bei Signaleinfall

Für eine Berechnung des Schnittpunkts der Geraden d_2 und d_3 werden zwei trigonometrischen Beziehungen in den Gleichung 4.5 und Gleichung 4.6 aufgestellt. Anschließend

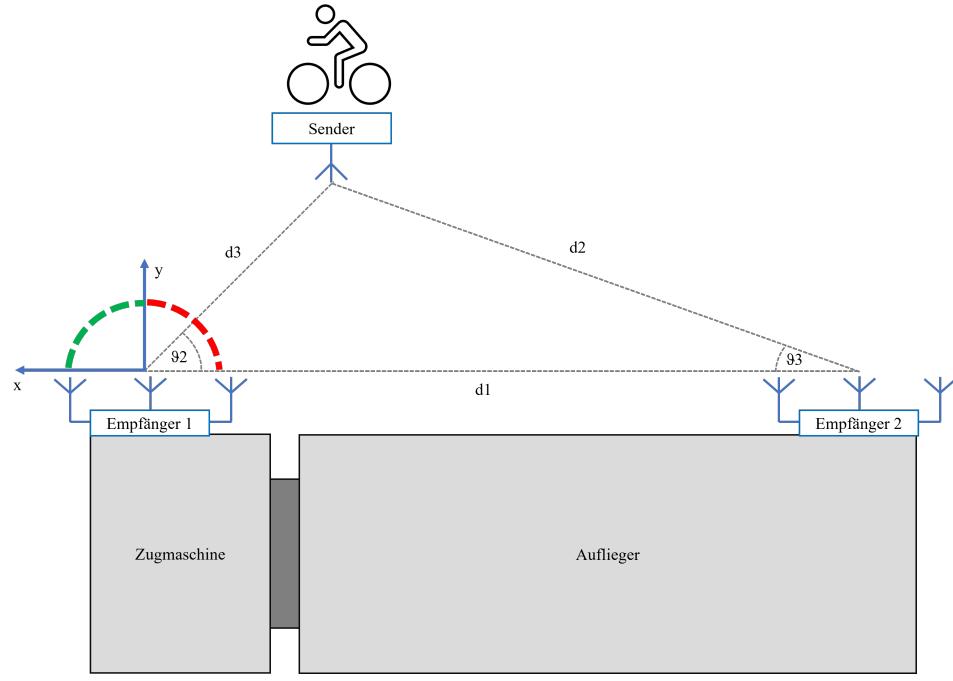


Abbildung 4.14: Definition der Empfänger-Koordinatensysteme für die Positionsrechnung des Senders

wird x für beide Signal-Einfallsinkel gemäß 4.7 gleichgesetzt. Gleichung 4.7 wird nach y umgestellt. Es resultieren 4.8 und 4.9, wobei das Vorzeichen von x invertiert wird, um die Position des Senders bzw. Fahrradhelms im Bezugskoordinatensystem des Lkw darzustellen.

$$x = \tan(\theta_2) \cdot y \quad (4.5)$$

$$x = \tan(\theta_3) \cdot y - d_1 \quad (4.6)$$

$$\tan(\theta_2) \cdot y = \tan(\theta_3) \cdot y - d_1 \quad (4.7)$$

$$y = \frac{d_1}{\tan(\theta_3) - \tan(\theta_2)} \quad (4.8)$$

$$x = -y \cdot \tan(\theta_2) \quad (4.9)$$

Des Weiteren werden die Sonderfälle berücksichtigt, in denen θ_2 bzw. θ_3 senkrecht über den Empfängern stehen. Die Berechnung der Position erfolgt über den nachfolgenden Programmauszug, wobei x und y auf drei signifikante Stellen gerundet werden.

```

1  try:
2      if theta2 == 0: # exception
3          y = round(tan(pi / 2 - theta3) * distance ,3)
4          x = 0
5      elif theta3 == 0: # exception
6          y = round(tan(pi / 2 + theta2) * distance ,3)
7          x = distance
8      else: # normal case
9          y = round(distance / (tan(theta3) - tan(theta2)) ,3)
10         x = round(-y * tan(theta2) ,3)
11 except ZeroDivisionError:
12     pass

```

4.5.4 Grafische Ausgabe

Die grafische Ausgabe visualisiert die Position des Fahrradfahrers im Bezugskoordinatensystem des Lkw. Es wird an die Visualisierung die Anforderung gestellt, dem Lkw Fahrer eine intuitive Einschätzung der Unfallgefahr durch angemessene Darstellung der Position zu ermöglichen. Abbildung 4.15 zeigt die grafische Ausgabe zur Visualisierung der Fahrradhelm-Position. Die blau dargestellten Halbkreise repräsentieren die Empfänger. Der schwarze Kreis bildet die Postion des Senders (Fahrradhelm) ab. Die grauen Linien visualisieren die Grenzen des Signal-Einfallswinkel-Erfassungsbereichs. Eine Ortung des

Senders außerhalb der Grenzen wird nicht unterstützt.

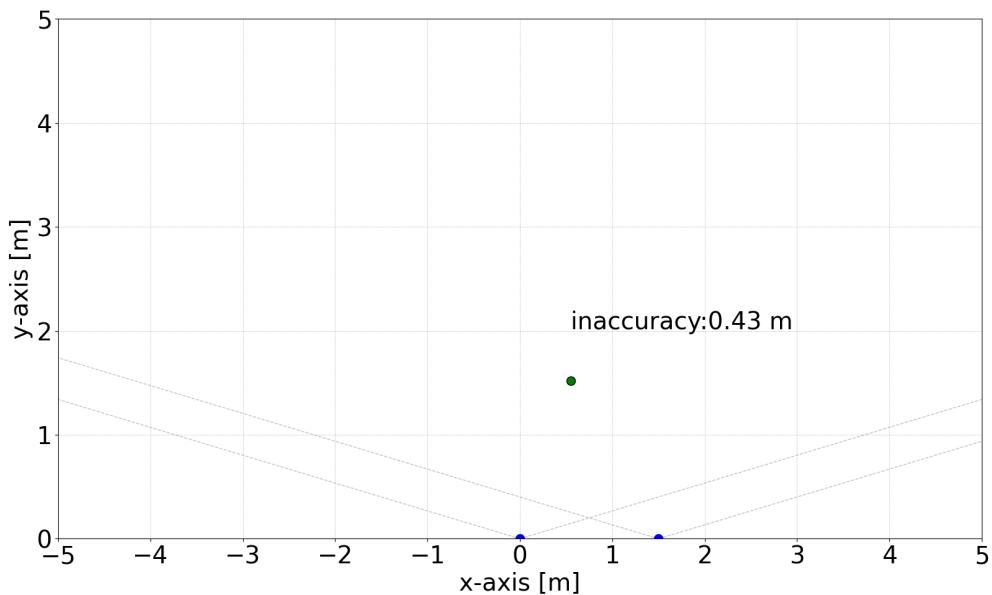


Abbildung 4.15: Visualisierung der Fahrradhelm-Position

Abbildung 4.16 zeigt die grafische Oberfläche bei Einstellung der identifizierten Stellgrößen. Die rechte Grenze des vorderen Signal-Einfallsinkel-Erfassungsbereichs liegt aufgrund der Rotation des ersten Empfängers unterhalb der x -Achse. Der Öffnungswinkel beider Erfassungsbereiche beträgt jeweils 150° . Gegenwärtig erfolgt die Ortung des intelligenten Fahrradhelms dauerhaft nach einmaligem Start des Python Scripts. Ein selbstständiges Ein- bzw. Ausschalten der Ortung und Visualisierung durch das System ist nicht implementiert. Problematisch ist dies, da dem Lkw-Fahrer auch bei dem einfachen Überholen von Fahrradfahrern im Stadtgebiet Warnmeldungen ausgegeben werden. Spätere Ausbaustufen sind mit einem selbstständigen Ein- bzw. Ausschalten der Visualisierung bei Erreichen von Geschwindigkeiten oder Lenkwinkeln, die einen Abbiegevorgang indizieren umzusetzen.

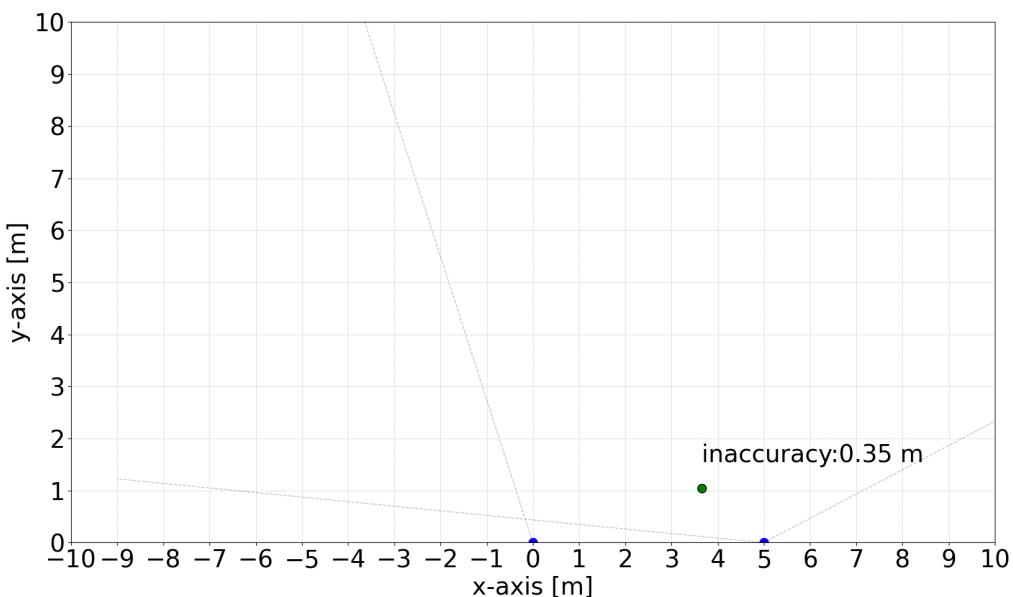


Abbildung 4.16: Visualisierung der Fahrradhelm-Position bei rotierten Empfänger-Koordinatensystemen

4.5.5 Programmlaufzeit des Python Scripts zum Aufruf der Visualisierung

Der Fahrradhelm inklusive Sender sowie der Lkw inklusive Empfänger sind bewegte Systeme. Damit unterliegt die Schätzung der Signal-Einfallswinkel einer Dynamik. Das zeitliche Verhalten ist insbesondere für den Wahrheitsgehalt der Visualisierung von Relevanz. Maßgeblich für das zeitliche Verhalten ist die Programmlaufzeit des Aufrufs der Visualisierung. Wenn die gespeicherten Signal-Einfallswinkel nicht innerhalb definierter Zeitgrenzen verarbeitet werden, repräsentiert die Visualisierung der Fahrradhelm-Koordinaten Zustände aus der Vergangenheit, siehe Abbildung 4.17. Die roten Indikatoren signalisieren einen Programmdurchlauf. Die grüne Linie umrahmt die Anzahl der zu verarbeitenden Events einer Mittelwertbildung. Die blauen und schwarzen Indikatoren entsprechen den Signal-Einfallswinkeln, welche vom ersten und zweiten Empfänger über die seriellen Schnittstellen an den Host-PC übertragen werden. Der schwarze Rahmen bildet einen Aufruf der Visualisierung ab. Eine unzulässig lange Programmlaufzeit verursacht, dass die dargestellte Fahrradhelm-Position vergangenen Zuständen entspricht. Die Zeitdifferenz zwischen dem dargestellten und dem wahren Zustand nimmt mit der Laufzeit des Scripts seit Beginn des Programmaufrufs zu. Die Darstellung der Events durch äquidistante blaue und schwarze Balken repräsentiert nur näherungsweise die Realität. Gemäß Hersteller werden die Daten je COM-Schnittstelle mit einer Frequenz von 40 Hz aktualisiert. In der praktischen Anwendung stellt sich bei der Verwendung von zwei COM-Schnittstellen heraus, dass die Daten über das Python Script mit einer Frequenz von 20 Hz abgerufen werden. Es wird vermutet, dass die seriellen Daten ke Schnittstelle mit 40 Hz anliegen, das Python Script, jedoch eine serielle Verarbeitung beider Schnittstellen durchführt und folglich nur 50 % der Events auswertet werden, wohingegen die anderen 50 % verworfen werden. Vereinfacht wird nachfolgend von einer Vollständigkeit der Daten ausgegangen, wenn 50 %, der Eingangsdaten über das Script mit 20 Hz eingelesen und verarbeitet werden. Die blauen und schwarzen Indikatoren sind äquidistant eingezeichnet. In der Realität ist die Anzahl der blauen und schwarzen Indikatoren identisch. Ihre Reihenfolge kann mitunter abweichen. Es sind beispielsweise zwei aufeinanderfolgende schwarze und danach zwei aufeinanderfolgende blaue Indikatoren möglich. Der Abstand ist nur im Mittel konstant. Eine Äquidistanz dient ausschließlich der besseren Darstellung.

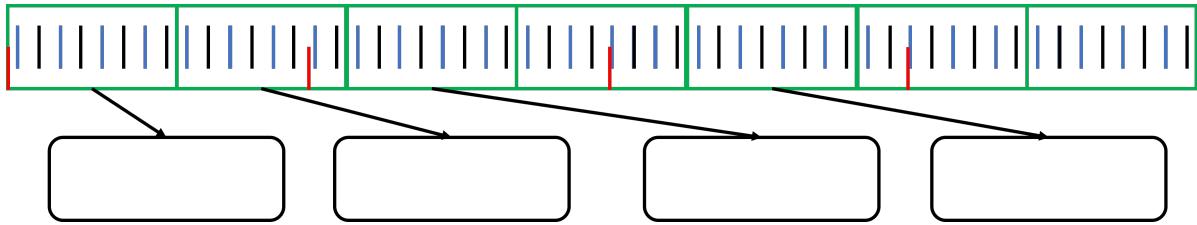


Abbildung 4.17: Keine Gewährleistung der Echtzeitdarstellung aufgrund unzulässig langer Programmlaufzeit des Python Scripts zum Aufruf der Visualisierung

An die Visualisierung wird die Forderung gestellt, die Fahrradhelm-Position in Echtzeit darzustellen. Notwendige Voraussetzung dafür ist, dass ausschließlich aktuelle Signal-Einfallswinkel verarbeitet werden. Die Bezeichnung aktuell bezieht sich auf die Signal-Einfallswinkel $\theta(k - n)$, mit $n = 0, 1, \dots, i$, wobei i für die Anzahl, der zur arithmetischen Mittelwertbildung genutzten Signal-Einfallswinkel steht. Eine Verarbeitung von Werten aus der Vergangenheit $\theta(k - n)$, mit $n > i$ wird programmtechnisch unterbunden. Der entsprechende Programmbeispiel ist nachfolgend abgebildet. Hierbei werden Werte, die nicht innerhalb eines Programmdurchlaufs aus dem Eingangsspeicher gelesen werden, verworfen. D. h wenn die Verarbeitung von i Events einer Laufzeit des Python Scripts entspricht, in der mehr als i Events empfangen werden, sind diese zusätzlich Empfangenen Daten im nächsten Aufruf nicht weiterzuverwenden. Stattdessen werden sie überschrieben. Der entsprechende Programmbeispiel ist nachfolgend aufgeführt.

```
1 serialdata_anchor1.reset_input_buffer()
```

Das Ergebnis dieser Optimierung wird in Abbildung 4.18 dargestellt. Werte außerhalb der grünen Rahmen werden verworfen.

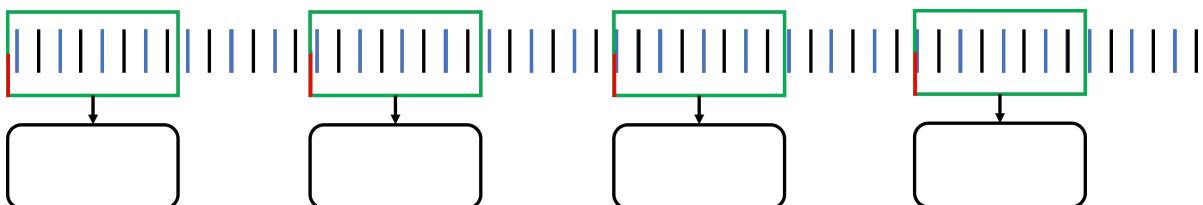


Abbildung 4.18: Programmtechnisch verbessertes Zeitverhalten zur Gewährleistung der Echtzeitdarstellung

Weiterhin problematisch ist jedoch, dass eine Programmalaufzeit $t_{Programm} \geq (2i) \cdot T_0$ zur Folge hat, dass einzelne Events und damit einzelne Signal-Einfallswinkel θ verworfen werden. Die Größe i entspricht vier Werten, die für eine arithmetische Mittelwertbildung benötigt werden. Der Faktor zwei resultiert aus dem seriellen Einlesen der beiden Schnittstellen mit einer Frequenz von 20 Hz. T_0 entspricht der Dauer zwischen zwei empfangenen Events je Schnittstelle, diese liegt bei ca. 25 ms [o V22a]. Wird diese Bedingung nicht eingehalten, erfolgt die Mittelwertbildung über einen längeren Betrachtungszeitraum, siehe Abbildung 4.18 und somit auch über längere Distanzen, welche das Fahrrad zwischen zwei Aktualisierungsaufrufen der Visualisierung zurücklegt. Für den Betrachter der grafischen Ausgabe äußert sich dies in sprunghaften Bewegungen des Fahrradhelms. Unter der Annahme, einer konstanten Relativgeschwindigkeit zwischen Fahrradhelm und Lkw folgt aus einer längeren Programmalaufzeit eine geringere Aktualisierungsrate und damit eine, ungenauere Auflösung der Position. Ferner besteht die Forderung, dass innerhalb eines bestimmten Zeitintervalls $t_{Programm} < (2i) \cdot T_0$ und damit $t_{Programm} < 200$ ms alle empfangenen Events $\theta(k)$, mit $k = 0, 1, \dots, i$ verarbeitet werden. Wenn diese Forderung eingehalten wird, ergibt sich eine Verarbeitung der Events gemäß Abbildung 4.19.

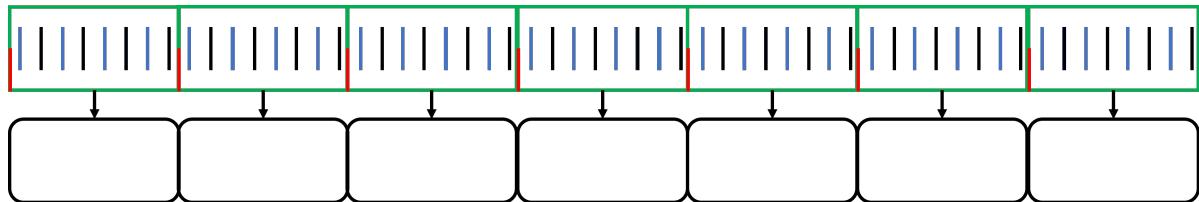


Abbildung 4.19: Programmtechnisch verbesserte Event-Verarbeitung, welche eine höhere Genauigkeit der Fahrradhelm-Position zur Folge hat

4.5.6 Bediener-Schnittstelle

Der Prototyp wird über das Python Script, das auf einem Host ausgeführt wird, gestartet. Bei Aufruf des Python Scripts wird der Benutzer aufgefordert Terminal-Eingaben durchzuführen, über welche dem Python Script die Stellgrößen des Messaufbaus übergeben werden. Nachdem der Benutzer darauf hingewiesen wurde, dass die COM-Schnittstellen erfolgreich geöffnet wurden (COM3 (first anchor) is opened!) bzw. (COM4 (second anchor) is opened!), wird er dazu aufgefordert die Länge d_1 zwischen dem ersten und zweiten Empfänger einzugeben. Dafür erscheint der Aufruf (please enter the distan-

ce d1 between anchor 1 and anchor 2). Anschließend ist der Gesamtbetrachtungsbe-
reich des Messaufbaus einzugeben (please enter the size of the overall measurement
area in m). Gemeint sind damit die Grenzen des Bereichs, der visualisiert werden soll.
Der Gesamtbetrachtungsbereich wird, quadratisches formatiert, dargestellt. Die unteren
Grenzen von x und y entsprechen dem negativen Wert der ganzzahligen Eingabe. Die
oberen Grenzen entsprechen dem positiven Eingabewert. Die Rotation der Empfänger-
Koordinatensysteme wird über die Abfragen (please enter the rotational angle of anchor
1 -60°...60°) und (please enter the rotational angle of anchor 2 -60°...60°) angefordert.
Alle Eingaben sind mit ganzzahligen Werten zu beantworten und im Anschluss mit der
Eingabe (Enter) Taste zu quittieren. Die Ausgabe des Terminals gibt die Koordinaten
 x und y eines Aufrufs sowie den aktuellen Zeitstempel im Portable Operating System
Interface (POSIX) Format zurück und wird für Analysezwecke zyklisch mit dem Aufruf
der Visualisierung des Fahrradhelms aktualisiert.

5 Verifizierung und Bewertung des Prototyps

5.1 Genauigkeit der Ortung

Kapitel 4.3 untersucht die Genauigkeit der Ortung für alle möglichen Stellgrößeneinstellungen innerhalb festgelegter Grenzen. Als Stellgrößen sind der Abstand zwischen den Empfängern d_1 im Bereich $1 \leq d_1 \leq 5$ mit einer Schrittweite von 1 m und die Rotation der Empfänger $\theta_{2_{offset}}$, $\theta_{3_{offset}}$ in den Bereichen $-60^\circ \leq \theta_{offset} \leq 60^\circ$ mit einer Schrittweite von 5° definiert. Über einen iterativen Ansatz wird eine Einstellung für die geringste Gesamtgenauigkeit der Ortung identifiziert. Die geringste Gesamtgenauigkeit entspricht dem kleinsten Mittelwert der Ortungsgenauigkeiten für alle möglichen Sender-Positionen (Fahrradhelm-Positionen) in einem definiert abgegrenzten Betrachtungsgebiet rechtsseitig des Lkw. Die geringste Gesamtgenauigkeit wird für die Stellgrößeneinstellung $d_1 = 5$ m, $\theta_{2_{offset}} = 55^\circ$ und $\theta_{3_{offset}} = -10^\circ$ identifiziert. Gegenstand der nachfolgenden Untersuchungen ist die Verifizierung und Bewertung der erreichten Genauigkeit des realisierten Prototyps.

5.1.1 Versuchsbeschreibung

Es wird die Prämisse gesetzt, ein innerstädtisches Ortungsszenario inklusive der damit verbundenen Einflüsse nachzubilden. Die Messung wird aus Sicherheitsgründen auf einem Innenhof durchgeführt. Der Abstand der Empfänger zur nächsten Hauswand beträgt circa 20 m in Richtung des Betrachtungsgebiets. Das Betrachtungsgebiet ist der Raum rechtsseitig des Lkw. Die Montagehöhe der Empfänger am Lkw bzw. des Senders am Fahrradhelm werden nicht nachgebildet. Sender und Empfänger werden auf High-Density-Polyethylen Kisten in einer Höhe von 260 mm über dem Boden platziert. Damit

wird eine LOS zwischen Sender und Empfänger gewährleistet. Sender und Empfänger sind freistehend, d. h. nicht am Lkw bzw. Fahrradhelm befestigt. Eventuelle Einflüsse z. B. eine Dämpfung aufgrund der Lkw Karosserie bzw. des Anhänger-Seitenwand-Materials werden nicht berücksichtigt. Der Versuchsumfang sieht die Messung verschiedener Sender-Positionen bei Einstellung der identifizierten Stellgrößen vor. Zudem wird eine Vergleichsmessreihe ohne Rotation der Empfänger für dieselben Positionen des Senders durchgeführt. Die erste Position liegt im vorderen Betrachtungsbereich des Gefahrengebiets bei $x = 0 \text{ m}$ und $y = 1 \text{ m}$ und bildet einen linken Grenzfall innerhalb des Betrachtungsbereichs ab, da die Position in der Nähe der linken Signal-Einfallsgrenze des ersten Empfängers liegt. Die zweite Position ist zentral verortet bei $x = 3 \text{ m}$ und $y = 2 \text{ m}$. Die Koordinaten im hinteren Bereich sind mit $x = 5 \text{ m}$ und $y = 1 \text{ m}$ festgelegt. Darüber hinaus wird mit $x = 6 \text{ m}$ und $y = 1 \text{ m}$ ein rechter Grenzfall ausschließlich für die Einstellung rotierter Empfänger-Koordinatensysteme getestet.

5.1.2 Durchführung

Der Sender wird im vorderen Randbereich der Kiste platziert und mit einer externen Stromversorgung verbunden. Die Empfänger werden über USB Leitungen¹ mit den seriellen Schnittstellen des Host-PC gekoppelt. Nach Aufruf des Python Scripts ist eine manuelle Eingabe der eingestellten Stellgrößen durchzuführen. Die Auswertung der Messungen erfolgt auf Grundlage von Bildschirmaufnahmen der grafischen Oberfläche. Hierbei wird die Abweichung der vom Prototyp georteten Sender-Position von der wahren Position abgelesen.

5.1.3 Auswertung

Abbildung 5.1 zeigt die grafische Ausgabe der Ortung des Senders bei $x \approx 0 \text{ m}$ und $y \approx 8,5 \text{ m}$ für den wahren Positions値 $x = 0 \text{ m}$ und $y = 1 \text{ m}$ ohne Rotation der Empfänger. Zunächst ist erkennbar, dass die wahre Position außerhalb der grau mar-

¹der Spezifikation 2.0

kiertem Signaleinfallsgrenzen und damit nicht innerhalb des unterstützten Messbereichs liegt. Eine fehlerhaft Ortung ist anzunehmen. Insbesondere in y -Richtung bestätigt sich diese Annahme. Die Abweichung von der wahren Position beträgt hierbei circa 7.5 m. Die Abweichung lässt sich systematisch auf die Einschränkung des abgedeckten Winkelmessbereichs durch den zweiten Empfänger entsprechend der hohen Winkelabweichungen im Bereich $\theta_3 < -75^\circ$ zurückführen.

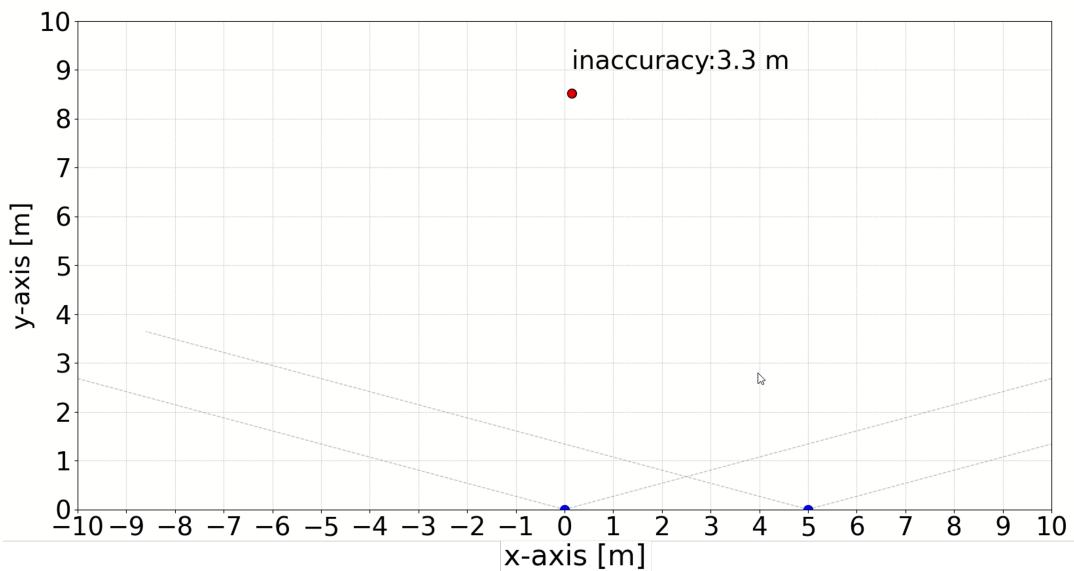


Abbildung 5.1: Vergleichsmessung ohne Rotation der Empfänger für den wahren Positions値 $x = 0 \text{ m}$ und $y = 1 \text{ m}$

Abbildung 5.2 zeigt die grafische Ausgabe für den wahren Positions Wert des Senders bei $x = 3 \text{ m}$ und $y = 2 \text{ m}$ ohne Rotation der Empfänger. Die wahre Position des Senders liegt innerhalb der grau markierten Signaleinfallsgrenzen. Eine Ortung der Position wird damit theoretisch vom Prototyp unterstützt. Die Lage des georteten Senders weist in y -Richtung eine positive Abweichung von der wahren Position auf. Damit wird der Sender hier weiter entfernt vermutet als er in der Realität ist. Die Tendenz ist kritisch zu bewerten, die Größenordnung ist jedoch $< 0,5 \text{ m}$ und wird als akzeptabel eingestuft. In x -Richtung weicht die geortete Lage negativ von der wahren Position ab. Die Größenordnung ist ebenfalls $< 0,5 \text{ m}$ und damit akzeptabel. In der praktischen Anwendung des Ortungssystems ist die wahre Position des Fahrradhelms nicht bekannt. Die Information *inaccuracy* gibt die, zu erwartende Positions-Ungenauigkeit an. Diese ist kreisförmig um die geortete Position zu ziehen. Die Berechnung erfolgt über das Python Script äquivalent zu der berechneten Ungenauigkeit je Position, in Kapitel 4.3 (Algorithmus) auf Grundlage der approximierten Winkelabweichungen. Die Größenordnung der berechneten Positions-Abweichung $\text{inaccuracy} = 0,42 \text{ m}$ entspricht in etwa der abgelesenen Abweichung von der wahren Position.

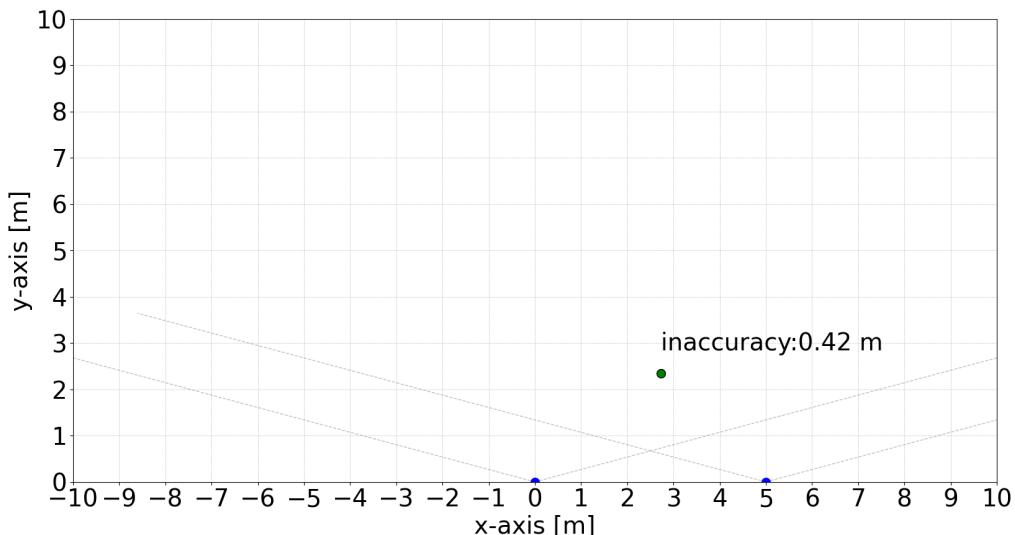


Abbildung 5.2: Ortung ohne Rotation der Empfänger für den wahren Positions Wert $x = 3 \text{ m}$ und $y = 2 \text{ m}$

Abbildung 5.3 zeigt die grafische Ausgabe der Ortung des Senders bei $x \approx 5$ m und $y \approx 6$ m für den wahren Positions値 $x = 5$ m und $y = 1$ m ohne Rotation der Empfänger. Hierbei ist ein identisches Verhalten zur ersten Messung bei $x = 0$ m und $y = 1$ m zu erkennen. Die Abweichung von der wahren Position beträgt in y -Richtung circa 5 m.

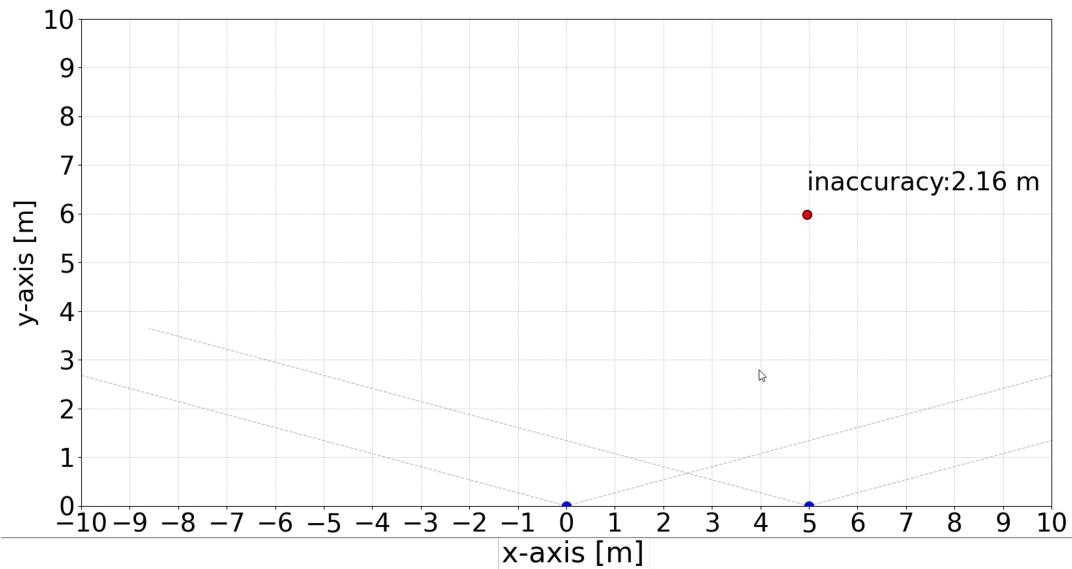


Abbildung 5.3: Ortung ohne Rotation der Empfänger für den wahren Positions値 $x = 5$ m und $y = 1$ m

Es ist gefordert, alle exemplarisch ausgewählten Positionen über den Abbiegeassistenten mit hoher Genauigkeit abzudecken, da diese im Gefährdungsbereich verortet sind. Ohne eine Rotation der Empfänger ist eine zuverlässige Ortung des Fahrradhelms nicht in allen drei Fällen möglich. Die folgenden Abbildungen 5.6, 5.4 und 5.5 zeigen die Ergebnisse bei Einstellung der Stellgrößen gemäß der identifizierten Werte $d_1 = 5 \text{ m}$, $\theta_{2_{offset}} = 55^\circ$ und $\theta_{3_{offset}} = -10^\circ$. Im zweiten und dritten Fall entspricht die mithilfe des Prototyps bestimmte Position der wahren Position des Senders. Die Anforderung eine Genauigkeit unter einem Meter zu erreichen wird in diesem Bereich erfüllt.

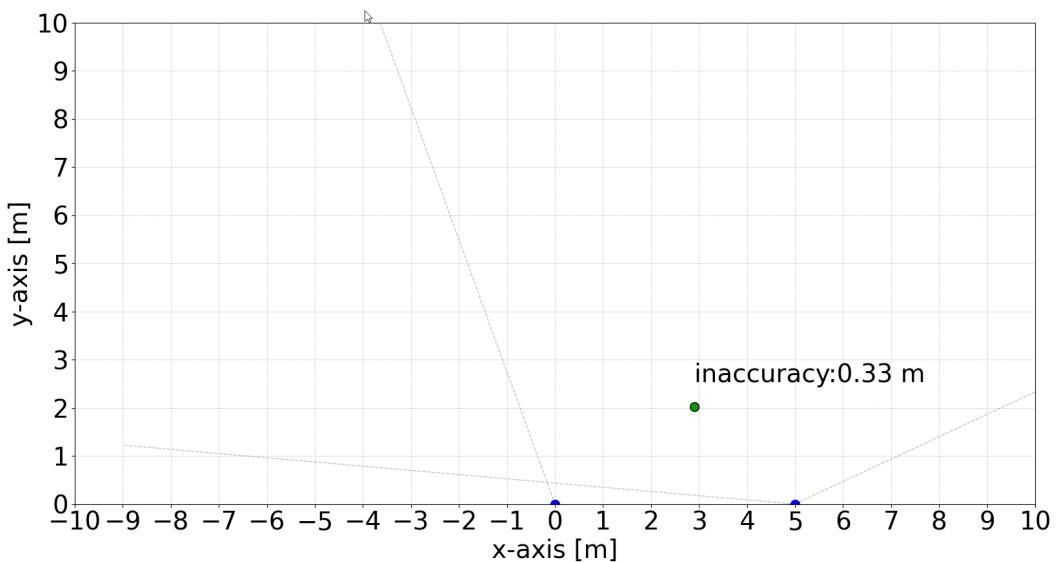


Abbildung 5.4: Messung 2 Ortung bei eingestellten Stellgrößen $d_1 = 5 \text{ m}$, $\theta_{2_{offset}} = 55^\circ$ und $\theta_{3_{offset}} = -10^\circ$ für den wahren Positions Wert $x = 3 \text{ m}$ und $y = 2 \text{ m}$

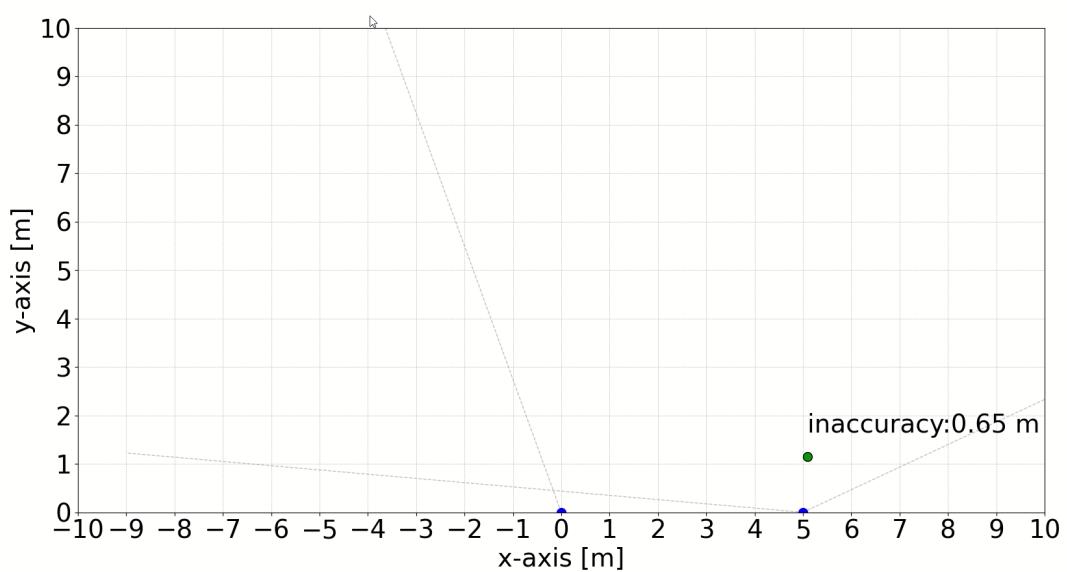


Abbildung 5.5: Messung 3 Ortung bei eingestellten Stellgrößen $d_1 = 5$ m, $\theta_{2_{offset}} = 55^\circ$ und $\theta_{3_{offset}} = -10^\circ$ für den wahren Positions Wert $x = 5$ m und $y = 1$ m

5 Verifizierung und Bewertung des Prototyps

In der ersten Messung, siehe Abbildung 5.6 weicht die geortete Position um circa 0,8 m von der wahren Position ab. Diese Abweichung entspricht der berechneten Positions-Abweichung $inaccuracy = 0,82$ m. Der Fahrradhelm wird hierbei jedoch weiter entfernt vermutet, als er in der Realität ist. In diesem Fall ist das Kollisionsrisiko aufgrund der Nähe zum Lkw stark ausgeprägt. Hierbei wird deutlich, dass die Berechnung der Gesamtungenaugkeit zur Identifikation der Stellgrößen in Kapitel 4.3 eine Abwägung tätigt, bei der Ungenauigkeiten in einem Teil des Abdeckungsbereich zu Gunsten anderer Teile in Kauf genommen werden müssen. Wenn die Koordinaten an den Grenzen des Betrachtungsbereichs liegen weist die Ortung auch bei Einstellung der Stellgrößen auffällige Abweichungen auf. Das Ausmaß der Abweichung gegenüber den nicht rotierten Empfängern konnte minimiert werden. Die Anforderung an den Prototyp eine Genauigkeit kleiner 1 m zu erreichen, wird auch für diese Messung verifiziert.

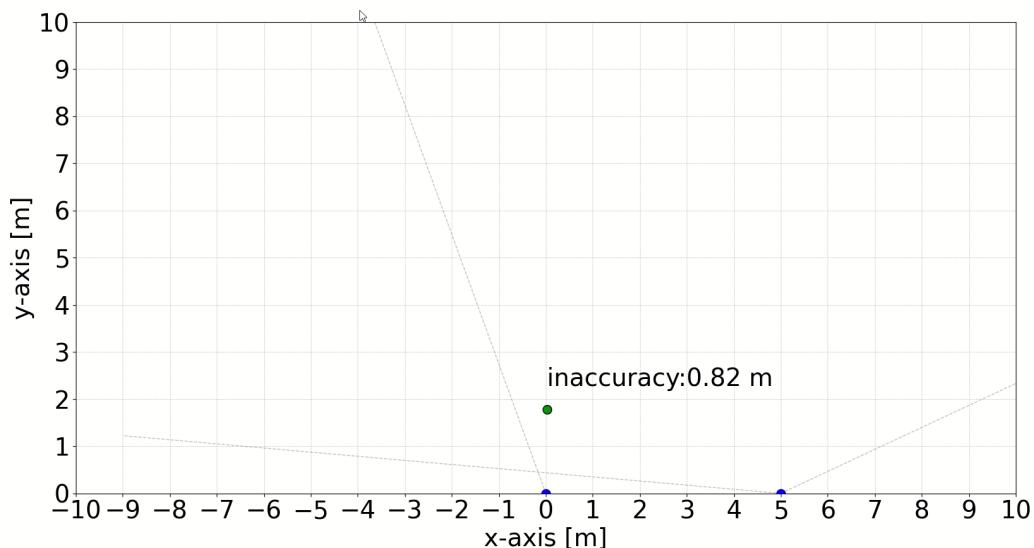


Abbildung 5.6: Messung 1 Ortung bei eingestellten Stellgrößen $d_1 = 5$ m, $theta_{2_{offset}} = 55^\circ$ und $theta_{3_{offset}} = -10^\circ$ für den wahren Positions-
wert $x = 0$ m und $y = 1$ m

Abbildung 5.7 stellt den Fall dar, dass der Sender in der Nähe der rechten Signaleinfalls-grenze des Betrachtungsbereichs liegt. Ähnlich zum linken Grenzfall ist eine kritische Abweichung von der wahren Position erkennbar. Nachfolgend wird die Ursache dieser Abweichung erklärt.

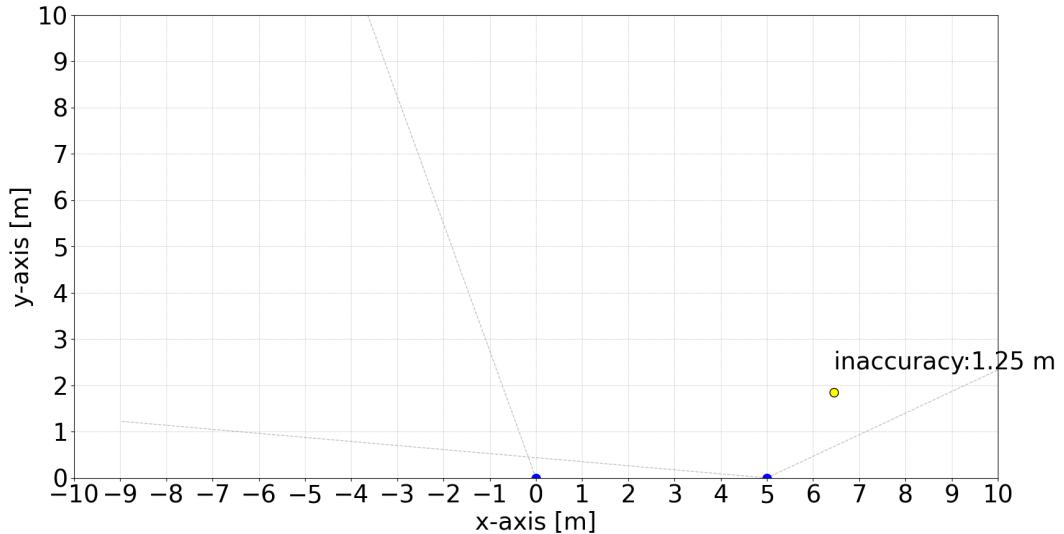


Abbildung 5.7: Messung 4 Ortung bei eingestellten Stellgrößen $d_1 = 5$ m, $\theta_{2_{offset}} = 55^\circ$ und $\theta_{3_{offset}} = -10^\circ$ für den wahren Positions-wert $x = 6$ m und $y = 1$ m

Bei Betrachtung des orange eingezeichneten Polynoms zur Approximation der Winkelabweichung in Abbildung 3.1 fällt auf, dass diese bereits ab $\pm 60^\circ$ auffällig stark ansteigt. Das Winkelerfassungsgebiet wird exemplarisch mit den Signaleinfallsgrenzen $\pm 60^\circ$ weiter eingeschränkt. Bei gleichbleibender Einstellung der Stellgrößen resultiert eine Abdeckung gemäß Abbildung 5.8. Hierbei wird das Gebiet in der Schleppkurve des Lkw bei einem Rechtsabbiegevorgang nicht abgedeckt. Dies ist nicht tolerierbar. Infolgedessen ist eine zusätzliche Rotation des hinteren Empfängers, siehe Abbildung 5.9 notwendig.

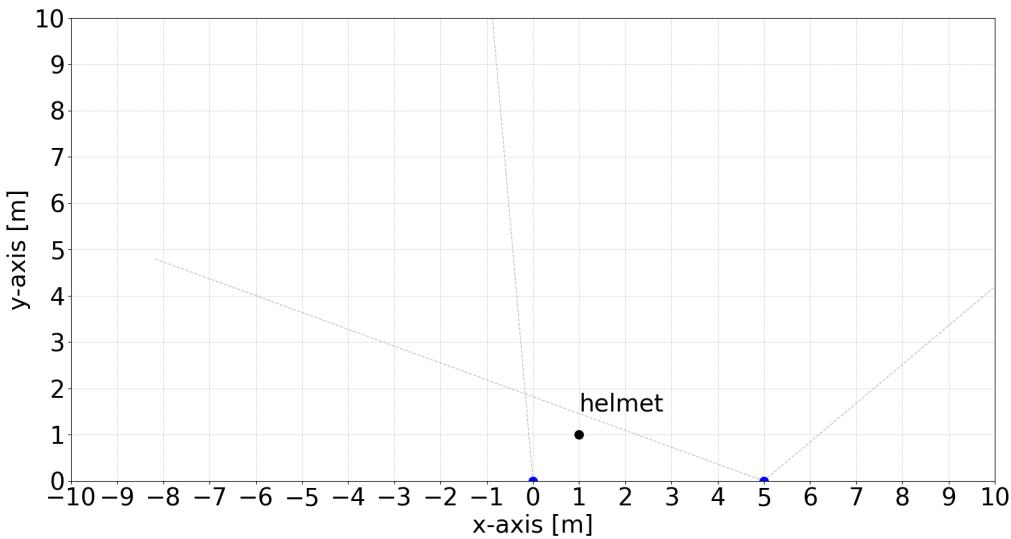


Abbildung 5.8: Beispieldarstellung für eine Einschränkung des zuverlässigen Signal-Einfallswinkelbereichs auf $\pm 60^\circ$ ohne Rotation der Empfänger

Zur Bestimmung der Stellgrößen wurde in Kapitel 4.3 ein Polynom implementiert, dass die Abweichungen des Signaleinfalls in Abhängigkeit vom wahren Wert approximiert. Dieses Polynom ist in der Abbildung 3.1 blau eingezeichnet. Insbesondere im Bereich -75° verfälscht es die Genauigkeit des Signaleinfalls ins Positive. Dieser Fehler wird durch Verwendung des Polynoms 6. Ordnung Gleichung 5.1 korrigiert. In der Abbildung 3.1 ist dieses orange eingezeichnet. Aufgrund des geänderten Polynoms und der notwendigen Rotation des zweiten Empfängers ergeben sich mithilfe des Algorithmus zur Stellgrößenidentifikation die neuen Stellgrößen $\theta_{2_{\text{offset}}} = 45^\circ$ und $\theta_{3_{\text{offset}}} = -30^\circ$. Der Abdeckungsbereich des Prototyps zur Ortung eines Fahrradhelms wird dadurch entsprechend der Abbildung 5.9 weiter eingeschränkt. Eine Vergrößerung des Abstands d_1 weit das Potenzial auf, den Abdeckungsbereich zu vergröern. Im Rahmen dieser Arbeit

wurde d_1 aus konstruktiven Gründen auf 5 m begrenzt. Auf Basis eines Vergleichs, des hier gezeigten Winkelabdeckungsbereichs mit dem ursprünglichen Winkelabdeckungsbereich bei einer Signal-Einfallsgrenze von $\pm 75^\circ$ wird die Schlussfolgerung gezogen, dass minimale Abweichungen in den Randbereichen zu Gunsten des Gesamtdeckungsbereichs toleriert werden. Damit werden die ursprünglichen Stellgrößen beibehalten. Die Entscheidung basiert auf zwei Thesen. Zum Einen ist es relevant den, im Kontext der Gewichtung beschriebenen toten Winkel abzudecken. Des Weiteren sind von hinten kommenden Fahrradfahrer zeitnah zu detektieren. Auf Grundlage dieser Detektion ist für nachfolgende Ausbaustufen des Prototyps eine Vorhersage ihrer Bewegung und damit ihrer Position umzusetzen. Dies setzt jedoch voraus, dass der Fahrradfahrer nicht erst kurz vor der Kollision, sondern über längere Distanzen erkannt wird. Die zweite These sagt aus, dass die Genauigkeit einer einzelnen Ortung bei einem bewegten Fahrradhelm durch Einsatz einer Mittelwertbildung verbessert werden kann.

$$\text{Winkelabweichung } \Delta\theta = \pm 5 \cdot 10^{-11} \theta^6 + 5,1 \quad (5.1)$$

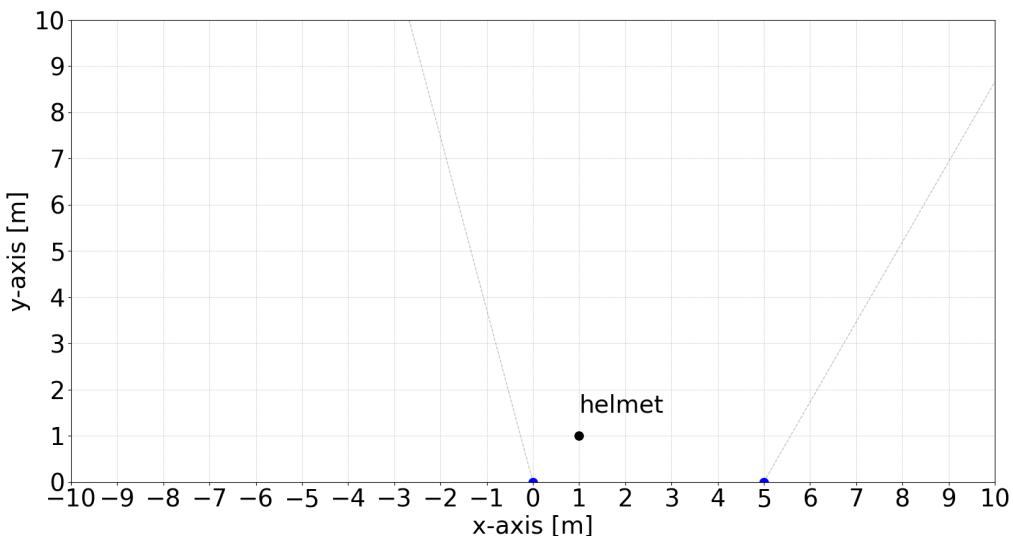


Abbildung 5.9: Beispieldarstellung für einen eingeschränkten Bereich der Signal-Einfallswinkelerfassung inkl. Rotation der Empfänger

5.2 Zeitverhalten des Python Scripts zum Aufruf der Visualisierung

Die Anforderung an die Programmalaufzeit lautet $t_{Programm} < (2i) \cdot T_0$. Die Zeitdauer zwischen zwei Events T_0 wird von der u-Blox Hardware mit 25 ms vorgegeben. Die Anzahl der, zur Mittelwertbildung genutzten Events i beträgt vier. Damit folgt die Anforderung $t_{Programm} < 200$ ms. Über diese Anforderung soll erreicht werden, dass alle, während eines Aktualisierungsdurchlaufs der Visualisierung empfangenen Events $\theta(k)$, mit $k = 0, 1, \dots, i$ verarbeitet werden. D. h., dass keine Events θ empfangen und anschließend verworfen werden sollen, weil das Programm nur vier Winkel pro Empfänger verarbeitet, die Laufzeit aber dem Empfang von z. B. sechs Winkeln entspricht. Eine Überprüfung des zeitlichen Verhaltens erfolgt durch das Protokollieren von Zeitstempeln bei jedem Aktualisierungsauftrag der grafischen Oberfläche. Abbildung 5.10 zeigt auf der y-Achse die Programmalaufzeit von 1864 Aktualisierungsdurchläufen der grafischen Oberfläche. Im Mittel beträgt die Programmalaufzeit 211 ms. In 7 % der Programmdurchläufe liegt die Laufzeit über der rot markierten Schwelle von 250 ms. Bis zu dieser Schwelle wird ein Wert pro Empfänger verworfen. Die maximale Laufzeit ist immer kleiner als 300 ms, sodass maximal 2 Werte je Empfänger verworfen werden. Die Anforderung an das Zeitverhalten des Aufrufs der Visualisierung kann nicht eingehalten werden. Die Ursache der Differenz von 11 ms ist dem implementierten Python Script zuzuordnen. Der Programmauftrag ist aus Stabilitätsgründen mit einer zusätzlichen Wartezeit von 10 ms versehen, in der alle Threads, die über das Python Script ausgeführt werden, gestoppt werden. Es besteht das Potential, die Wartezeit von 10 ms auf 0 ms zu reduzieren. Gegenwärtig führt diese Wartezeitreduktion jedoch zu einem Absturz der grafischen Oberfläche. Im Rahmen dieser Arbeit konnte keine Ursache identifiziert werden.

Unter der Annahme, dass sich der Fahrradfahrer mit einer Relativgeschwindigkeit von 20 $\frac{\text{km}}{\text{h}}$ fortbewegt, legt er in dem Zeitraum zwischen zwei Aktualisierungsaufträgen der Visualisierung, gemäß Gleichung 5.2 eine Strecke von 1,172 m zurück. Diese Strecke wird als räumliche Auflösung bezeichnet. Für die bestimmungsgemäße Verwendung des Prototyps als Abbiegeassistsystem wird die räumliche Auflösung als unzureichend hoch eingestuft. Ursache für das Ausmaß der räumlichen Auflösung bei konstanter Relativ-

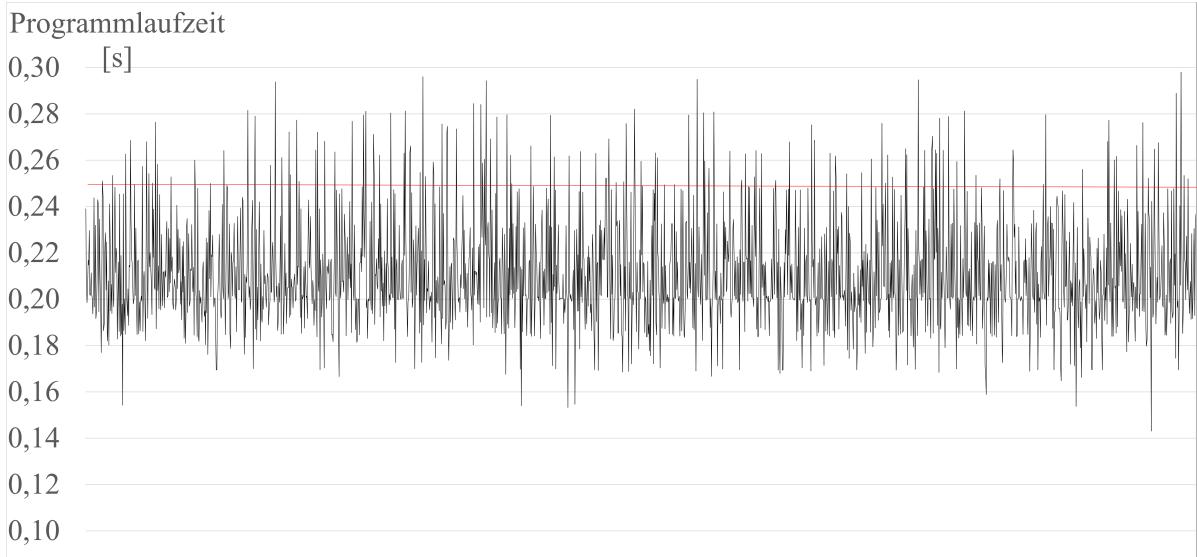


Abbildung 5.10: Programmlaufzeit des Python Scripts, betrachtet über 1864 Aufrufe

geschwindigkeit ist die Programmlaufzeit des Python Scripts. Wichtigste Einflussgröße der Programmlaufzeit ist die Anzahl der, zur Mittelwertbildung genutzten Events. Ein Ansatz zur Verbesserung der räumlichen Auflösung um 100 % auf 0,586 m ist die Mittelwertbildung über zwei an Stelle von vier Messwerten. Gänzlich ohne Mittelwertbildung wird eine Auflösung von ca. 0,293 m angenommen. Um das prognostizierte Vorstudienergebnis und damit dem bestmöglichen Wert von ca. 0,134 zu erreichen ist eine Optimierung der Programmlaufzeit des Python Scripts zur parallelen Verarbeitung der seriellen Daten des ersten und zweiten Empfängers notwendig.

$$20 \frac{\text{km}}{\text{h}} \cdot \frac{1}{3,6} \frac{\text{m} \cdot \text{h}}{\text{s} \cdot \text{km}} \cdot 0.211 \frac{\text{s}}{\text{Aktualisierung}} = 1,172 \frac{\text{m}}{\text{Messwert}} \quad (5.2)$$

Die implementierte Mittelwertbildung über mehrere Events inklusive Ausreißer-Unterdrückung verursacht einen Zielkonflikt zwischen der räumlichen Auflösung und der Genauigkeit einer einzelnen Ortung dar. Der Zielkonflikt wurde bei diesem Prototyp zugunsten der Genauigkeit einer einzelnen Ortung abgewogen. Ein Ansatz zur Verbesserung der Programmlaufzeit und damit zur Verbesserung der räumlichen Auflösung wird in der Anwendung angepasster Methoden zur Mittelwertbildung vermutet. Gegenwärtig erfolgt

diese über vier Werte, die allesamt nur einmal verwendet werden. Die Mittelwertbildung des nächsten Programmaufrufs erfolgt über vier neue Werte, siehe Abbildung 4.18. Bei der Implementierung eines gleitenden Mittelwertfilters wird ausschließlich der älteste Wert entfernt. Die übrigen drei Werte werden im nächsten Aufruf wiederverwendet, siehe Abbildung 5.11. Die roten Indikatoren signalisieren einen Programmduurchlauf. Der grüne Rahmen grenzt die Anzahl der zu verarbeitenden Daten für eine Mittelwertbildung ein. Die blauen und schwarzen Indikatoren entsprechen den Events inklusive der Signal-Einfallswinkel, welche vom ersten und zweiten Empfänger über die seriellen Schnittstellen an den Host-PC übertragen werden. Der schwarze Rahmen bildet einen Aufruf der Visualisierung ab. Es wird vermutet, dass sich die Programmlaufzeit durch Implementierung eines gleitenden Mittelwertfilters auf ca. 50 ms zwischen zwei Aktualisierungsaufrufen reduzieren lässt. Damit ist zum einen eine höhere Ortsgenauigkeit und zum anderen eine bessere räumliche Auflösung ohne Sprünge in der Visualisierung der Fahrradhelm-Position erreichbar.

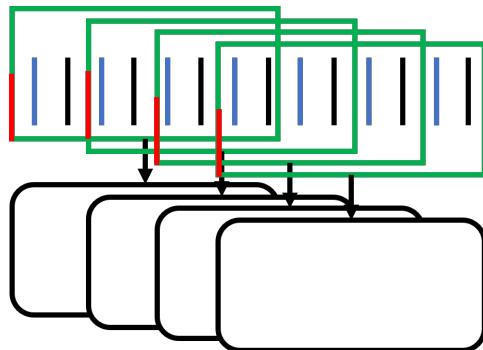


Abbildung 5.11: Zeitverhalten bei Implementierung eines gleitenden Mittelwertfilters

6 Zusammenfassung

Der Prototyps intelligenter Fahrradhelm Teilsystem Bluetooth Ortung wird eingesetzt, um die Position eines intelligenten Fahrradhelms in einem Abdeckungsbereich rechtsseitig eines Lkw zu bestimmen. Die Forschungsfrage der Arbeit lautet, ist ein Prototyp auf Basis der u-Blox Hardware XPLR-AoA1, als Abbiegeassistsystem geeignet?

Der Prototyp wird auf die Eigenschaften Genauigkeit der Ortung und räumliche Auflösung untersucht. Zur Bestimmung werden verschiedene Positionen des Senders bei Einstellung der identifizierten Stellgrößen $d_1 = 5 \text{ m}$, $\theta_{2_{offset}} = 55^\circ$ und $\theta_{3_{offset}} = -10^\circ$ getestet. Zudem wird eine Ortung bei Lage des Senders auf denselben Positionen ohne Rotation der Empfänger-Koordinatensysteme durchgeführt. Damit soll der Einfluss der Stellgrößen überprüft werden. Im Fall nicht rotierter Empfänger variiert die Genauigkeit der Ortung je nach Position des Senders im Abdeckungsbereich signifikant. In den Randbereichen der Signal-Einfallswinkelerfassung ist keine zuverlässige Ortung möglich. Dies ist problematisch, da die betrachteten Koordinaten in einem Bereich liegen, der vom Ortungssystem zwangsläufig abzudecken ist. Eine Anpassung des Aufbaus entsprechend der Stellgrößen erzielt hierbei die gewünschten Verbesserungen. Die Genauigkeit kann in allen Fällen auf akzeptable Werte gesteigert werden. Der im Stand der Technik 2.1 beschriebene Mindestabdeckungsbereich ist mit Ausnahme der horizontalen Bereiche $-2 \text{ m} < x < 0 \text{ m}$ und $6 \text{ m} < x < 9 \text{ m}$ erfüllt. Um den Anforderungen des Mindestabdeckungsbereichs gemäß Stand der Technik Nummer 2.1 gerecht zu werden, ist eine Ausbaustufe des Prototyps notwendig. Eine Möglichkeit den Abdeckungsbereich weiter zu steigern ist die Vergrößerung des Abstands d_1 zwischen den Empfängern. Bei Anpassung des Stellgrößenbereichs $0 \text{ m} \leq d_1 \leq 10 \text{ m}$ ergeben sich über den Algorithmus zur Identifizierung der Stellgrößen die Werte $d_1 = 9 \text{ m}$, $\theta_{2_{offset}} = 40^\circ$ und $\theta_{3_{offset}} = -60^\circ$. Neben der Erfüllung des Mindestabdeckungsbereichs wird die Genauigkeit der Ortung über diese Stellgrößenanpassung in nahezu jedem Bereich verbessert. Die Ungenauigkeiten der Ortung sind der Abbildung 6.1 zu entnehmen.

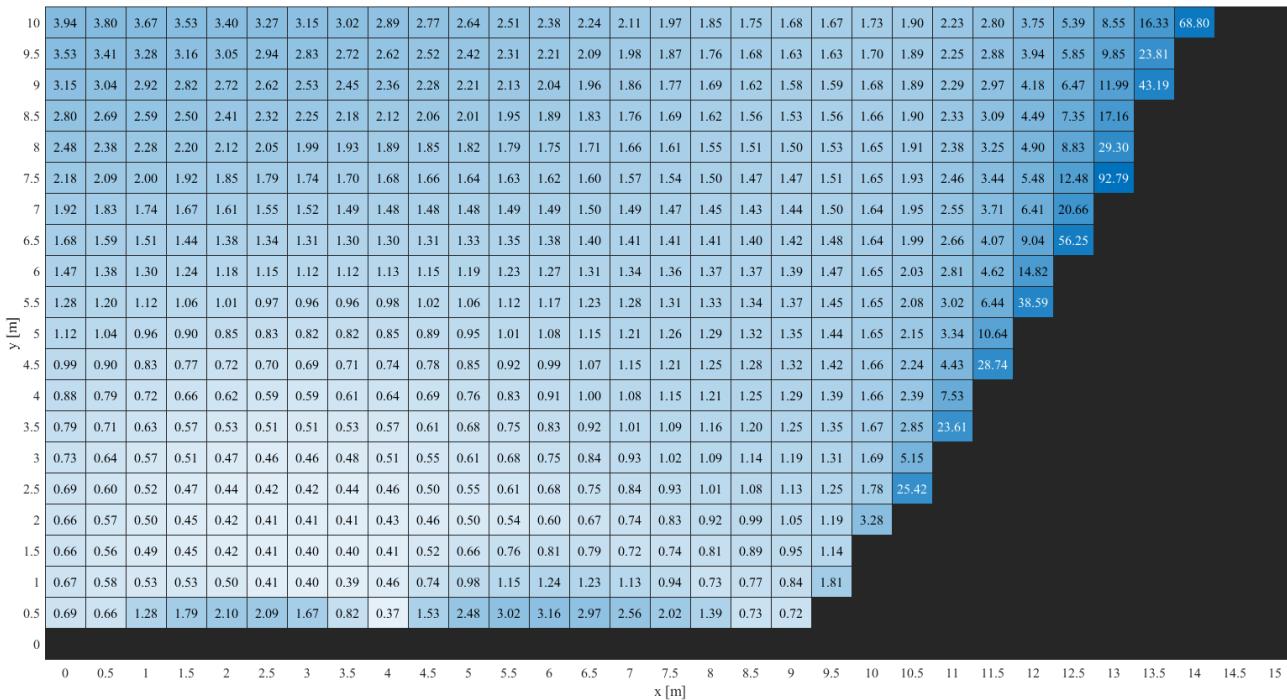


Abbildung 6.1: Ungenauigkeit der Ortung je Zelle bei Einstellung der Stellgrößen $d_1 = 9$ m, $\text{theta}_{2_{offset}} = 40^\circ$ und $\text{theta}_{3_{offset}} = -60^\circ$

6 Zusammenfassung

Die räumliche Auflösung der Position des bewegten Fahrradhelms bei konstanter Relativgeschwindigkeit zwischen Lkw und Fahrradfahrer ist im Wesentlichen von der Programmlaufzeit des Python Scripts abhängig. Bei einer Relativgeschwindigkeit von $20 \frac{\text{km}}{\text{h}}$ legt der Fahrradfahrer in dem Zeitraum zwischen zwei Aktualisierungsaufrufen der Visualisierung eine Strecke von 1,172 m zurück. Diese Auflösung wird als unzureichend eingestuft. Haupteinflussgröße ist die Programmlaufzeit des Python Scripts zum Aufruf der Visualisierung. Die Programmlaufzeit wird maßgeblich von der Anzahl der Events beeinflusst, deren Signaleinfallsinkel zur Mittelwertbildung angesetzt werden. Mit jedem verarbeiteten Event pro Empfänger nimmt die Programmlaufzeit um 25 ms zu. Die Laufzeit des gegenwärtig implementierten Python Scripts beträgt zuzüglich einer funktionsrelevanten Wartezeit circa 211 ms. Zur Verbesserung der Programmlaufzeit und damit zur Optimierung der räumlichen Auflösung wird ein gleitender Mittelwertfilter vorgeschlagen. Der Durchsatz, den dieser Filter erzeugt, ist viermal höher, sodass die Programmlaufzeit auf circa 50 ms und die räumliche Auflösung auf 0,293 m verbessert werden kann.

Zusammenfassend wird der Prototyp des winkelbasierten Bluetooth Ortungssystems auf Basis der U-Blox Hardware XPLR-AOA-1 bei Umsetzung der genannten Anpassungen als geeignet bewertet, um ein Abbiegeassistenzsystem zu realisieren. In der gegenwärtigen Ausbaustufe erfüllt er nicht vollständig die Anforderungen des Gesetzgebers an ein Abbiegeassistenzsystem gemäß Verkehrsblatt 2022, Heft 8, Nr. 65 des Bundesministeriums für Digitales und Verkehr. Verbesserungen sind hinsichtlich der Mittelwertberechnung der Einfallsinkel des Abdeckungsbereich zu treffen. Neben einer Variation der Stellgrößen stellt die Verwendung zusätzlicher Empfänger einen möglichen Ansatz zur Optimierung des Abdeckungsbereichs dar. Des Weiteren besteht die Möglichkeit, neben einem einzelnen Fahrradhelm bis zu 20 Sender über die Hardware XPLR-AOA-1 zu orten. Hier ist eine Bestimmung des Fahrradfahrers, welcher der größten Gefährdung ausgesetzt ist, umzusetzen. Neben dem Ortungssystem ist die grafische Ausgabe an die Anforderungen des Gesetzgebers, siehe Stand der Technik Kapitel 2 anzupassen. Die Anzeige der Position ist um eine zweistufige Warnung, erstens bei Eindringen des Fahrradfahrers in den Abdeckungsbereich und zweitens bei einem prognostizierten Zusammenstoß auf Basis des Lenkwinkels des Lkw zu erweitern. Es wird abschließend empfohlen, spätere Ausbaustufen des Abbiegeassistenzsystems nach dem Stand der Technik entsprechend der UN-Regelung Nr. 151 oder alternativ nach dem Verkehrsblatt 2022, Heft 8, Nr. 65

6 Zusammenfassung

des Bundesministeriums für Digitales und Verkehr zu verifizieren. Abschließend resultiert die Fragestellung, ob weitere Ausbaustufen den gesetzlichen Anforderungen gerecht werden.

Literatur

- [Cam84] Joe Campbell. *V24/RS-232-Kommunikation*. 1. Aufl. Berkeley: SYBEX-Verl., 1984. ISBN: 3-88745-075-2.
- [Car11] Carsten Fritzsche. „Statistical Data Fusion for Hybrid Localization of Mobile Terminals“. Diss. Darmstadt: Technische Universität, 2011. URL: <http://tuprints.ulb.tu-darmstadt.de/2499/>.
- [Chr20] Chris Liechti. *pySerial*. 2020. URL: <https://pyserial.readthedocs.io/en/latest/>.
- [DD16] Dipl.-Ing. Jörg Ortlepp und Dipl.-Ing. Petra Butterwegge, Hrsg. *Unfalltypenkatalog „UNKA“: Leitfaden zur Bestimmung des Unfalltyps*. Berlin, 2016. URL: <https://www.udv.de/udv/themen/unfalltypenkatalog-unka--81916>.
- [Den23] Dennis Gardy. *Prototypisierung der winkelbasierten Bluetooth Ortung eines intelligenten Fahrradhelms: mit dem Schwerpunkt empirischer Untersuchungen der Hardware Plattform XPLR-AOA-1*. Potsdam, 2023.
- [Dip06] Dipl.-Ing. Steffen Wild. „Mehrwegeausbreitung bei GNSS-gestützter Positionsbestimmung“. Diss. Dresden: Technische Universität, 2006. URL: <http://webdoc.sub.gwdg.de/ebook/disssts/Dresden/Wildt2007.pdf>.
- [Dr-18] Dr.-Ing. Axel Malczyk, Hrsg. *In-depth Analyse schwerer Unfälle mit schweren Lkw*. Berlin, 2018. URL: <https://www.udv.de/resource/blob/78928/9e50de657432b5427e21c7da1ae265b2/80-schwere-unfaelle-mit-schweren-lkw-data.pdf>.
- [Gui22] Guido Zielke. *Verkehrsblatt Amtlicher Teil: Empfehlungen zu technischen Anforderungen an Abbiegeassistsysteme für die Aus- und Nachrüstung an Nutzfahrzeugen mit einer zulässigen Gesamtmasse > 3,5 Tonnen und Kraftomnibussen mit mehr als 9 Sitzplätzen einschließlich Fahrerplatz zur Erteilung einer Allgemeinen Betriebserlaubnis für Abbiegeassistsysteme*. Bonn, 2022.

Literatur

- URL: https://www.bmdv.bund.de/SharedDocs/DE/Anlage/StV/aktion-abbiegeassistent-verkehrsblatt.pdf?__blob=publicationFile.
- [o V21] o. V. *USB 2.0 Specification / USB-IF*. 2021. URL: <https://www.usb.org/document-library/usb-20-specification>.
- [o V22a] o. V. *Product summary XPLR-AOA-1*. Thalwil, Switzerland, 2022. URL: <https://www.u-blox.com/en/product/xplr-aoa-1-kit>.
- [o V22b] o. V. *XPLR-AOA-1 and XPLRAOA-2 explorer kits: Bluetooth direction finding: User guide*. Thalwil, Switzerland, 2022. URL: https://content.u-blox.com/sites/default/files/XPLR-AOA-Explorer-kits_UserGuide_UBX-21004616.pdf.
- [o V23a] o. V. *ABE - Abbiegeassistent*. 2023. URL: https://www.kba.de/DE/Themen/Typgenehmigung/Informationen_TGV/ABE_Abbiegeassistent/abbiegeassistent_node.html.
- [o V23b] o. V. *Luis turn detect*. 2023. URL: <https://www.luis.de/turn-detect-7-zoll-pro-left-420500/>.
- [Pro19] Prof. Jürgen Brauckmann, Hrsg. *Aktion Toter Winkel: Gefahr für Radfahrer*. Düsseldorf, 2019. URL: <https://www.landesverkehrswacht-nrw.de/service/aktion-toter-winkel/>.
- [San21] Sandro Eichhübl. *Regex: Zeichenfolgen schnell und effizient überprüfen*. Dornbirn, Austria, 2021. URL: <https://www.massiveart.com/blog/regex-zeichenfolgen-die-das-entwickler-leben-erleichtern>.
- [SS15] Benjamin Schreck und Patrick Seiniger. *Abbiege-Assistenzsystem für Lkw: Grundlagen eines Testverfahrens*. Bd. Heft 104. Berichte der Bundesanstalt für Straßenwesen. Bremen: Fachverlag NW in der Carl Schünemann Verlag GmbH, 2015. ISBN: 9783956061714. URL: https://bast.opus.hbz-nrw.de/opus45-bast/frontdoor/deliver/index/docId/1345/file/F_104_barrierefreies_Internet_PDF.pdf.
- [Tho22] Thomas Magenheim-Hörmann. *Verkehrsunfälle: Jeder zweite Schwerverletzte ist Radfahrer: Studie zur Sicherheit*. Hannover, 2022. URL: <https://rnd.de/wirtschaft/verkehrsunfaelle-in-deutschland-jeder-zweite-schwerverletzte-ist-radfahrer-Z20G5EJFQVDKFEFB4MVEFAFP24.html>.

Literatur

- [Tim22] Tim Abert. *Möglichkeiten der Entwicklung eines intelligenten Fahrradhelms zur Unfallprävention*. Erfurt, 2022.
- [WB05] Heinz Wörn und Uwe Brinkschulte. *Echtzeitsysteme: Grundlagen, Funktionsweisen, Anwendungen : mit 440 Abbildungen und 32 Tabellen*. eXamen.press. Berlin, Heidelberg und New York: Springer, 2005. ISBN: 3540205888.

Verzeichnis verwendeter Formelzeichen und Abkürzungen

T_0	Dauer zwischen zwei empfangenen Events (25 ms)
$\Delta\theta$	Winkelabweichung
$\overline{\Delta\theta}$	arithmetischer Mittelwert aller Winkelabweichungen
θ	Signal-Einfallswinkel
$\theta(k)$	empfangenen Events während eines Aufrufs der Visualisierung
θ_1	Innenwinkel gegenüber der Seite d_1
θ_2	AoA am ersten Empfänger, Innenwinkel gegenüber der Seite d_1
θ_3	AoA am zweiten Empfänger, Innenwinkel gegenüber der Seite d_1
$\theta_{2_{offset}}$	Gradmaß der Rotation des ersten Empfängers
$\theta_{3_{offset}}$	Gradmaß der Rotation des zweiten Empfängers
θ_{mess}	gemessener Wert des Signal-Einfallswinkels
θ_{wahr}	wahrer Wert des Signal-Einfallswinkels
d_1	Länge zwischen dem ersten und zweiten Empfänger am LkW
d_2	Länge zwischen dem zweiten Empfänger und dem Sender
d_3	Länge zwischen dem ersten Empfänger und dem Sender
i	Anzahl der zur Mittelwertbildung genutzten Signal-Einfallswinkel

Verzeichnis verwendeter Formelzeichen und Abkürzungen

ABE Allgemeine Betriebserlaubnis

Akku Akkumulator

AoA Angle of Arrival

Azimut horizontale Signal-Einfallsinkel

*BAS*t** Bundesanstalt für Straßenwesen

BLE Bluetooth Low Energy

COM Component Object Model

CPU central processing unit

CTE Constant Tone Extension

CTS Clear to Send

directionfinding Peilung

Elevation vertikale Signal-Einfallsinkel

FIFO First In First Out

GIDAS German In-Depth Accident Study

Host Notebook, auf dem Python Script aufgerufen wird

inaccuracy Ungenauigkeit

KBA Kraftfahrtbundesamt

LED light-emitting diode

Lkw Lastkraftwagen

LOS line-of-sight

Verzeichnis verwendeter Formelzeichen und Abkürzungen

MAC Media Access Code

MAE mean absolute error

MCU micro controller unit

NLOS non-line-of-sight

PEP Python Enhancement Proposal

PKW Personenkraftwagen

PoC Proof of Concept

POSIX Portable Operating System Interface

Radar Radio detection and ranging

Regex Regular Expression

RS232 Recommended Standard 232

RSSI Received Signal Strength Indication

RTS Request to Send

Threads Prozesse

UNKA Unfalltypenkatalog

USB Universal Serial Bus

x Koordinate auf der Achse in Längsrichtung des Lkw

y Kordinate auf der senkrecht auf dem Lkw stehenden Koordinatenachse

Abbildungsverzeichnis

2.1	Anforderung an den Abdeckungsbereich von Abbiegeassistenten [Gui22]	5
4.1	Bezugskoordinatensystem	12
4.2	Darstellung des toten Winkels (rot schraffiert) [Pro19]	14
4.3	Schematischer Aufbau der Triangulation	15
4.4	Visualisierung der Ungenauigkeit der Ortung	16
4.5	Visualisierung der Ungenauigkeit der Ortung für eine geänderte Lage des Senders und eine Variation des Abstands d_1	17
4.6	Spiegel am Lkw	20
4.7	Unterscheidung der Sichtbereiche in der Lkw Umgebung	21
4.8	Bemaßung der Sichtbereiche in der Lkw Umgebung	22
4.9	Gewichtungsfaktor kombiniert aus Sichtbereich und Unfallgefahr	23
4.10	Ungenauigkeit der Ortung je Zelle bei Einstellung der Stellgrößen $d_1 =$ 5 m , $\theta_{2_{offset}} = 55^\circ$ und $\theta_{3_{offset}} = -10^\circ$	25
4.11	Jumper Einstellung für eine Kommunikation über USB [o V22b, S. 12]	27
4.12	Sender C209 [o V22b, S. 21]	28
4.13	Definition der Empfänger-Koordinatensysteme bei Signaleinfall	34
4.14	Definition der Empfänger-Koordinatensysteme für die Positionsberch- nung des Senders	35
4.15	Visualisierung der Fahrradhelm-Position	37
4.16	Visualisierung der Fahrradhelm-Position bei rotierten Empfänger- Koor- dinatensystemen	38
4.17	Keine Gewährleistung der Echtzeitdarstellung aufgrund unzulässig langer Programmlaufzeit des Python Scripts zum Aufruf der Visualisierung	40
4.18	Programmtechnisch verbessertes Zeitverhalten zur Gewährleistung der Echtzeitdarstellung	40
4.19	Programmtechnisch verbesserte Event-Verarbeitung, welche eine höhere Genauigkeit der Fahrradhelm-Position zur Folge hat	41

Abbildungsverzeichnis

5.1	Vergleichsmessung ohne Rotation der Empfänger für den wahren Positions Wert $x = 0 \text{ m}$ und $y = 1 \text{ m}$	45
5.2	Ortung ohne Rotation der Empfänger für den wahren Positions Wert $x = 3 \text{ m}$ und $y = 2 \text{ m}$	46
5.3	Ortung ohne Rotation der Empfänger für den wahren Positions Wert $x = 5 \text{ m}$ und $y = 1 \text{ m}$	47
5.4	Messung 2 Ortung bei eingestellten Stellgrößen $d_1 = 5 \text{ m}$, $\theta_{2_{offset}} = 55^\circ$ und $\theta_{3_{offset}} = -10^\circ$ für den wahren Positions Wert $x = 3 \text{ m}$ und $y = 2 \text{ m}$	48
5.5	Messung 3 Ortung bei eingestellten Stellgrößen $d_1 = 5 \text{ m}$, $\theta_{2_{offset}} = 55^\circ$ und $\theta_{3_{offset}} = -10^\circ$ für den wahren Positions Wert $x = 5 \text{ m}$ und $y = 1 \text{ m}$	49
5.6	Messung 1 Ortung bei eingestellten Stellgrößen $d_1 = 5 \text{ m}$, $\theta_{2_{offset}} = 55^\circ$ und $\theta_{3_{offset}} = -10^\circ$ für den wahren Positions Wert $x = 0 \text{ m}$ und $y = 1 \text{ m}$	50
5.7	Messung 4 Ortung bei eingestellten Stellgrößen $d_1 = 5 \text{ m}$, $\theta_{2_{offset}} = 55^\circ$ und $\theta_{3_{offset}} = -10^\circ$ für den wahren Positions Wert $x = 6 \text{ m}$ und $y = 1 \text{ m}$	51
5.8	Beispieldarstellung für eine Einschränkung des zuverlässigen Signal- Einfallswinkelbereichs auf $\pm 60^\circ$ ohne Rotation der Empfänger	52
5.9	Beispieldarstellung für einen eingeschränkten Bereich der Signal- Einfalls- winkelerfassung inkl. Rotation der Empfänger	53
5.10	Programmlaufzeit des Python Scripts, betrachtet über 1864 Aufrufe	55
5.11	Zeitverhalten bei Implementierung eines gleitenden Mittelwertfilters	56
6.1	Ungenauigkeit der Ortung je Zelle bei Einstellung der Stellgrößen $d_1 = 9 \text{ m}$, $\theta_{2_{offset}} = 40^\circ$ und $\theta_{3_{offset}} = -60^\circ$	58

Tabellenverzeichnis

2.1	Abbiegeassistenten, die über eine ABE des KBA verfügen Stand (04.04.2023)	4
3.1	Winkelabweichungen vom wahren Signaleinfallswinkel in der Messkampagne Freiraumverhalten	8
4.1	Syntax der vom Empfänger übertragenen Events [o V22b, S. 28]	30