

FH Aachen

**Fachbereich
Maschinenbau und Mechatronik
Studiengang Mechatronik**

Bachelorarbeit

**Validierung einer Simulationsumgebung für Umfeldsensorik von
Schienenfahrzeugen**

**Luigi Ferraioli
Matr.-Nr.: 3232755**

Referent: Prof. Dr. Raphael Pfaff

Korreferent: Patrick Cönen, M.Sc.

4. Oktober 2023

Erklärung

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Aachen, 4. Oktober 2023

A handwritten signature in black ink, appearing to read 'Gerrard', written in a cursive style.

Danksagung

Hiermit möchte ich meine Dankbarkeit gegenüber all jenen ausdrücken, die dazu beigetragen haben, dass diese Arbeit erfolgreich abgeschlossen werden konnte.

An erster Stelle danke ich meinem Professor und Betreuer Raphael Pfaff für seine wertvolle Unterstützung, seine Bereitschaft, immer zur Stelle zu sein, wenn es nötig war und seine mitreißende Begeisterung für Züge, die diese Arbeit erst möglich gemacht haben.

Ebenso möchte ich meinem Zweitprüfer Patrick Cönen meinen aufrichtigen Dank aussprechen. Er stand jederzeit bereit, um mir mit wertvollen Ratschlägen und neuen Perspektiven zur Seite zu stehen. Seine Unterstützung war für mich von unschätzbarem Wert und hat maßgeblich dazu beigetragen, dass diese Arbeit vollendet werden konnte.

Ein besonderer Dank geht auch an Lukas Hildebrand, der mich bereits während meiner ersten Bachelorarbeit erfolgreich unterstützt hat und mir wertvolle Einblicke in das wissenschaftliche Arbeiten vermittelt hat.

Mein Dank gilt auch David, der nicht nur gemeinsam mit mir am Projekt des Gazebo-Trains gearbeitet hat, sondern auch immer bereit war, mich mit Rat und Tat zu unterstützen.

Schließlich gebührt mein Dank meinen Eltern und meinem Bruder. Ihr unermüdlicher Glaube an mich und ihre bedingungslose Unterstützung (besonders finanziell) haben mir die Kraft gegeben, meine akademischen Ziele zu verfolgen.

Vor allem möchte ich mich bei Caro bedanken, die in Zeiten von Stress und Anspannung immer an meiner Seite war und stets aufmunternde Worte für mich hatte. Die Unterstützung hat mir in schwierigen Momenten geholfen und meine Studienzeit noch bedeutungsvoller gemacht.

Kurzfassung

Diese Bachelorarbeit befasst sich mit der digitalen Nachbildung eines Testgeländes sowie eines Schienenfahrzeugs in der Simulationsumgebung Gazebo. Der Schwerpunkt liegt auf der präzisen Abbildung der Umfeldsensorik anhand eines realen Schienenfahrzeuges. Ziel ist die Erzeugung äquivalenter Messdaten der Simulationsumgebung und des realen Schienenfahrzeuges unter gleichen Einsatzbedingungen. Dazu werden unterschiedliche Verfahren eingesetzt, um die Parameter der einzelnen Sensorik so zu konfigurieren, dass die Messergebnisse mit den Messdaten der realen Sensorik konvergieren.

Die Ergebnisse der Messdaten zeigen, dass obwohl die Simulationsumgebung einige physikalische Materialeigenschaften nicht berücksichtigt, eine präzise Abbildung der Sensorik und geometrischen Strukturen des Testgeländes möglich ist. Darüber hinaus ermöglicht die Kombination von Gazebo und ROS2 Integrationstests und die Entwicklung von Softwareanwendungen sowohl in der Simulation als auch auf dem realen Schienenfahrzeug. Eine realitätsnahe und reproduzierbare Auswertung der Sensormessdaten der Simulationsumgebung für Schienenfahrzeuge ist somit realisierbar.

Inhalt

1 Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung und Simulationsanforderungen	2
1.3 Stand der Technik	3
2 Theoretische Grundlagen	5
2.1 Simulation	5
2.1.1 Erstellung eines Modells	5
2.1.2 Modellerweiterung Sensor	7
3 Hardware- und Softwarekomponenten	9
3.1 Hardwarekomponenten	9
3.1.1 EMMA	9
3.1.2 Messaufbau	10
3.1.3 Eingesetzte Sensorik	11
3.2 Softwarekomponenten	13
3.2.1 Robot Operating System	13
3.2.2 Gazebo	14
3.2.3 Gazebo-Train	14
4 Methodik	16
4.1 Simulationsmodell	16
4.1.1 Kalibrierung des Simulationsmodells	16
4.1.2 Kalibrierung der IMU	17
4.1.3 Kalibrierung des Lidars	21
4.1.4 Kalibrierung der Stereokamera	25
5 Evaluation	27
5.1 Beurteilung der Ergebnisse	27
5.1.1 Beurteilung des IMU-Sensors	27
5.1.2 Beurteilung des Lidars	29
5.1.3 Beurteilung der Stereokamera	32
5.1.4 Beurteilung der Simulationsumgebung	33
5.2 Limitation	34
5.3 Ausblick für die Forschung	35
6 Fazit	36
Quellenverzeichnis	39

Inhalt

Abbildungsverzeichnis	41
Tabellenverzeichnis	42
7 Anhang	43

1 Einleitung

1.1 Motivation

Eine fortlaufende Verbesserung und Erweiterung verschiedenster Systeme ist für eine ökonomische und ökologische Anwendung zwingend erforderlich. Hierbei kommen gezielt Softwaresimulationen zum Einsatz, die in nahezu allen Bereichen wiederzufinden sind und sich stetig weiterentwickeln. Dabei sind Effizienz, Sicherheit und Zeitersparnis von essenzieller Bedeutung. Medizinische Simulationen können für die Produktentwicklung, Diagnose und Therapie für den Arzt und Patient einige Vorteile bringen, wie beispielsweise das Testen neuer Behandlungsmethoden oder Operationen, ohne das Leben eines Patienten zu gefährden. So werden bereits heute, in speziellen Disziplinen wie der Arthroskopie oder Wirbelsäuleneingriffe, computergestützte Simulation verwendet [1].

Besonders in technischen Zusammenhängen spielen Softwaresimulationen eine entscheidende Rolle. In der Architektur und im Bauwesen werden Softwaresimulationen verwendet, um Strukturen und Gebäudeentwürfe visualisieren und optimieren zu können. Des Weiteren ist es möglich, das physikalische Verhalten der Gebäude zu analysieren und ebenfalls zu optimieren [2].

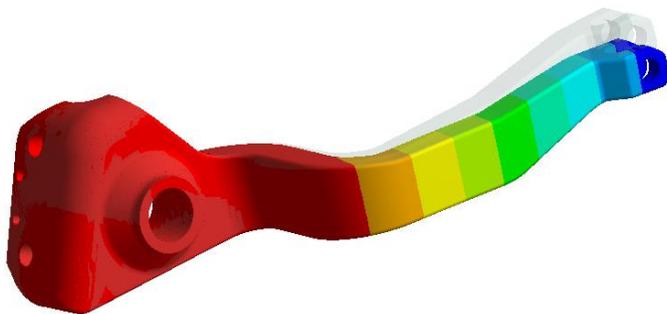


Abbildung 1.1: Simulation einer statischen Festigkeitsanalyse mit 3D-Solids [3]

Das bedeutendste Näherungsverfahren für strukturmechanische Berechnungen stellt die Finite Elemente Methode dar. Mit dieser Simulationsmethode ist es möglich, umfassende Eigenschaften, wie beispielsweise die Ermittlung von Spannungen, Verformungen, Temperaturen, Eigenfrequenzen usw. der Bauteile und Baugruppen zu ermitteln [3]. Die Simulation einer strukturmechanischen Berechnung mit 3D-Solids ist in Abbildung [1.1]

dargestellt. Mithilfe der Finite Elemente Methode ist es möglich, Bauprozesse zu verbessern und Material effizienter einzusetzen.

Durch den immer größer werdenden Fortschritt in der Robotik- und der Automatisierungsbranche gewinnen Simulationen auch hier an Bedeutung. Roboter und automatisierte Systeme können vor ihrer physischen Umsetzung virtuell getestet und optimiert werden. Bereits heute können digitale Nachbildungen erstellt und an den realen Sensor angepasst werden [4]. Eine fließende Übertragung der Software auf die Hardware reduziert nicht nur Entwick-

1 Einleitung

lungszeiten und -kosten, sondern erhöht auch die Sicherheit und Zuverlässigkeit der Systeme. Auch im Schienenverkehrswesen wird diese Methode genutzt, um das Fahrverhalten beispielsweise beim Beschleunigen, Ausrollen oder Bremsen des Zuges zu beurteilen. Diese Daten können genutzt werden, um das Design oder die Auslegung des Schienenfahrzeugsystems zu überprüfen und zu verbessern [5].

1.2 Zielsetzung und Simulationsanforderungen

Diese Bachelorarbeit untersucht die Simulationsgüte für Umfeldsensorik von Schienenfahrzeugen in der Simulationsumgebung von Gazebo. Ausgehend von einem physischen Schienenfahrzeug sollen die Simulationsumgebung und die Sensoren für das Simulationsmodell optimiert werden. Die Simulation soll unter dem Robot Operating System (ROS) bzw. ROS2 und der dazugehörigen Simulationssoftware Gazebo simulierbar sein. Im Zuge der Entwicklung werden die Sensoren so konfiguriert, dass diese dem physikalischen Modell entsprechen. Die Messabweichungen zwischen realen und simulierten Sensorsignalen dürfen die jeweiligen Grenzwerte der Hersteller nicht überschreiten. Die relative Messabweichung repräsentiert die Standardabweichung des Sensorsignals, während die absolute Messabweichung die Differenz zwischen den realen und simulierten Sensorsignalen darstellt. Die jeweiligen Schwellenwerte der Hersteller sind der Tabelle 1.1 zu entnehmen.

Tabelle 1.1: Systemanforderungen [6], [7], [8]

	Relative Messabweichung	Absolute Messabweichung
Messabweichung des IMU-Sensors		
Beschleunigung in m/s^2	0,031	0,031
Winkelgeschwindigkeit in $^\circ/s$	0,16	0,16
Messabweichung des Lidars		
Messabweichung in mm	30	—
Übereinstimmung der 2D-Scans in mm	± 550	± 550
Stereokamera Abstandsmessung		
Messabweichung	$> 1\%$ bis 3 m $> 5\%$ bis 15 m	$> 1\%$ bis 3 m $> 5\%$ bis 15 m
Simulation		
Messabweichung in mm	± 550	—
Physikalische Eigenschaften	—	z. B. Materialeigenschaften

1 Einleitung

Durch die Validierung der Simulationsumgebung für Umfeldsensorik von Schienenfahrzeugen sollen künftig Tests durchgeführt werden können, die reproduzierbare und realitätsnahe Ergebnisse liefern. Das Einbetten und Testen neuer Sensorik sowie Aktuatorik soll, mithilfe der modularen Struktur von ROS2, erleichtert werden. Zusätzlich soll ROS2 eine nahtlose Integration von Hardware und Software ermöglichen, wodurch die Entwicklungszeit beschleunigt werden kann.

1.3 Stand der Technik

Der heutige Schienenverkehr befindet sich inmitten einer Transformation. Schienenfahrzeuge werden zunehmend mit Sensorik und Kommunikationstechnologien ausgestattet. Diese Fortschritte revolutionieren die Branche und ermöglichen eine effizientere und sicherere Nutzung des Schienennetzes durch teilautonomes und autonomes Fahren. Heute gibt es weltweit mehr als 60 Metrosysteme, die vollautomatisch über eine zentrale Leitstelle gesteuert werden [9]. Die Aufgaben, die sonst von Menschen ausgeführt werden, wie das Anfahren, das Anhalten an Haltestellen und das Öffnen sowie Schließen der Türen, werden von Computersystemen übernommen. Möglich ist dies durch die Vernetzung von Fahrzeugen, Gleisen, Stellwerken und der Ausstattung von Sensorik am Schienenfahrzeugsystem und entlang der Strecke. Abbildung 1.2 zeigt die Verwendung des Positive Train Controls, eines Ortungssystems, um die Sicherheit und Effizienz im Bahnverkehr zu verbessern. Dieses System wurde dafür konzipiert, Unfälle zu verhindern, die durch das Missachten von Signalen oder das Überschreiten von Geschwindigkeitsbegrenzungen verursacht werden können [10].

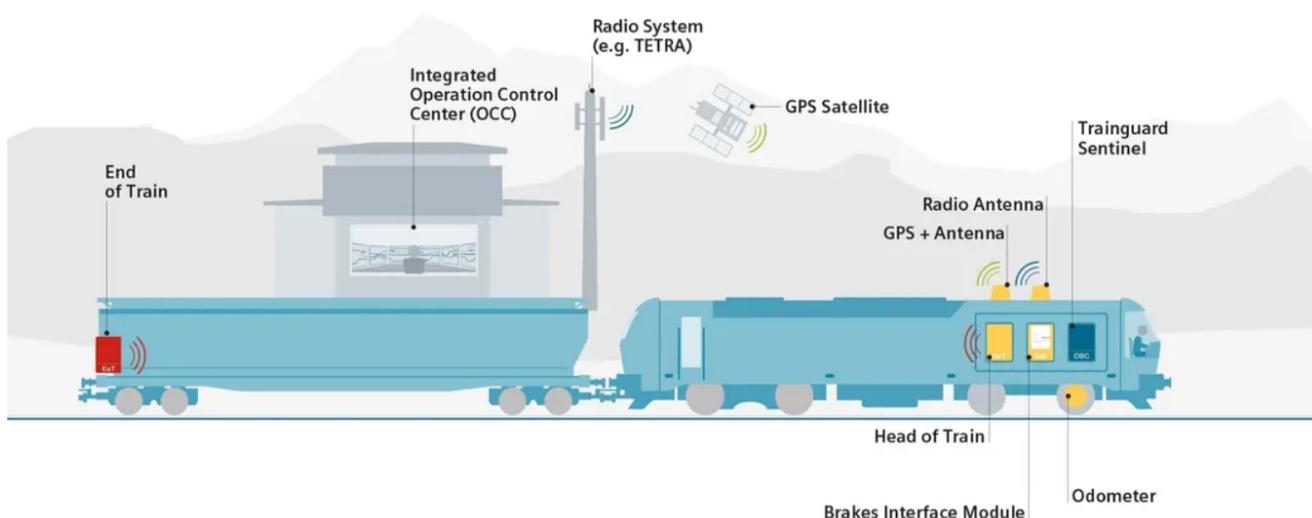


Abbildung 1.2: Positive Train Control [10]

1 Einleitung

Die computergestützte Simulation von Sensorik und Prozessen ist für viele moderne Anwendungen von zentraler Bedeutung. Digitale Zwillinge, also Nachbildungen in der Simulation, werden genutzt, um die Sicherheit, Energieeinsparung, Emissionsreduzierung und den effizienten Betrieb des realen Schienenfahrzeugsystems zu gewährleisten [11]. Neben der manuellen Erstellung digitaler Modelle können mithilfe von effizienten und intelligenten Technologien, wie die Verwendung von Light-Detection-and-Ranging (Lidar)-Daten und Algorithmen für maschinelles Lernen, automatisch Modelle generiert werden [12]. Die Erstellung von mehreren Kilometern Oberleitungsanlagen kann durch den Einsatz von luftgestützten Lidar-Daten realisiert werden, was den Entwicklungsprozess erheblich beschleunigt. Die digitalen Zwillinge ermöglichen es, Bahnanlagen, Bahnhöfe, Bahnsysteme und einzelne Anlagen wie Züge, Weichen, Sensoren usw. zu generieren [13].

Ein wichtiger Punkt in der Simulation von intelligenten Systemen ist die Wahrnehmung der eigenen Bewegungen sowie deren Umfeld. Diese ist ein fundamentaler Aspekt für die Lokalisierung und die Navigation. Die Wahrnehmung erfolgt bei der Simulation und der Hardware über Sensoren. Hierbei kann zwischen zwei Arten von Sensoren unterschieden werden: exterozeptive und interozeptive Sensoren. Exterozeptiven Sensoren, wie beispielsweise ein Lidar oder eine Kamera, sind in der Lage, Umgebungsinformationen zu erfassen. Interozeptive Sensoren hingegen konzentrieren sich auf die Erfassung der Daten über den Zustand und die Bewegungen des eigenen Systems. Die Fusion von exterozeptiven und interozeptiven Sensoren ist elementar, um ein umfangreiches Verständnis der Umgebung und des Systemzustands zu gewährleisten. Moderne Schienenfahrzeuge sind mit einer Vielzahl von Sensoren ausgestattet, die zur Sicherheit, Effizienz und Kontrolle des Zugbetriebs beitragen. So sorgen bei autonomen Schienenfahrzeuge die Kamera- und Radarsysteme für Sicherheit auf dem Bahnsteig [9]. Eine Geschwindigkeitsüberwachung kann durch die Verwendung eines Global Navigation Satellite System (GNSS) und einer Inertial Measurement Unit (IMU) realisiert werden, wobei es sich bei dem GNSS um ein Satellitennavigationssystem und bei der IMU um eine Kombination aus Beschleunigungs- und Drehratensensor handelt [14].

2 Theoretische Grundlagen

In diesem Kapitel werden die theoretischen Grundlagen erläutert, die für eine Erstellung eines Simulationsmodells unter ROS benötigt werden. Zunächst wird die Erstellung eines visuellen Modells behandelt, gefolgt von einer Erläuterung, wie durch Erweiterungen sensorische Eigenschaften hinzugefügt werden können.

2.1 Simulation

2.1.1 Erstellung eines Modells

Es gibt mehrere Möglichkeiten, ein Modell oder eine Simulationsumgebung unter ROS zu erstellen. Die zwei Hauptformate zum Erstellen eines Modells sind das Unified Robot Description Format (**URDF**) und Simulation Description Format (**SDF**). Sie basieren beide auf der Extensible Markup Language (**XML**). Der wesentliche Unterschied der beiden Formate besteht darin, dass mit einem URDF der Roboter beschrieben wird, während mit einem SDF sowohl der Roboter als auch die Simulationsumgebung beschrieben werden kann.

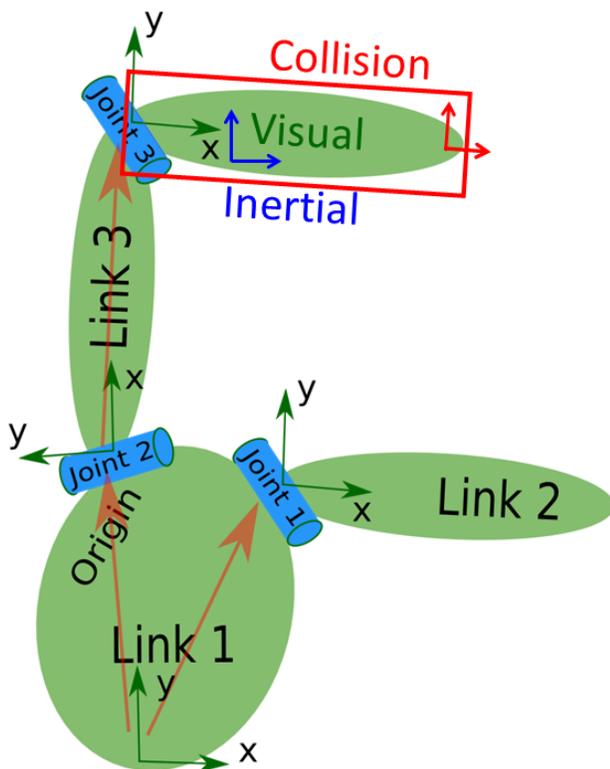


Abbildung 2.1: Verbindung einer Simulationstruktur nach [15]

Jedes Modell besteht grundlegend aus Gelenken (engl.: Joints) und Verbindungen (engl.: Links), die zu einem System verknüpft werden. Jeder Joint und jeder Link besteht wiederum aus mehreren Elementen und Attributen. Da die Joints und Links unterschiedliche Eigenschaften besitzen können, müssen sich diese durch ein spezifisches Namensattribut voneinander differenzieren. Mithilfe von Links können sichtbare Elemente eines Modells dargestellt werden. Statische Objekte bzw. deren Links müssen mindestens aus den zwei Elementen Visual und Collision bestehen (siehe Abbildung 2.1). Die Eigenschaft Visual ist für das optische Erscheinungsbild nötig und kann entweder durch ein 3D-Mesh, also einem 3D-Objekt oder durch eine mathematische Funktion beschrieben werden. Die Collision ist entscheidend für die Interaktion des Objekts mit seiner Umgebung. In den meisten Fällen wird

eine vereinfachte mathematische Funktion oder 3D-Mesh verwendet und über das visuelle Modell gelegt. Dies bietet den Vorteil, dass die Komplexität des Simulationsmodells und so-

2 Theoretische Grundlagen

mit auch die Rechenleistung verringert wird. Der Link kann durch zusätzliche Elemente wie Reibung und Dämpfung erweitert werden, um ein realistischeres Modell zu erhalten.

Für dynamische Objekte bzw. deren Links kommt ein drittes Element, das Inertial, hinzu. Dieses Element beschreibt die Massenträgheit, den Schwerpunkt und die Masse des Links. Um realitätsnahe dynamische Bewegungsabläufe des realen Objektes zu realisieren, ist eine exakte Bestimmung dieser Elemente von hoher Priorität. Die Abbildung [2.2](#) visualisiert beispielhaft die drei genannten Elemente, die zur Beschreibung eines Modells benötigt werden. Ausnahmen der Darstellung stellen Links wie beispielsweise der `camera_lens_links` dar. Dieser erhält lediglich das Namensattribut und repräsentiert einen virtuellen Link. Der Link wird benötigt, um die Kameralinse nicht am Montagepunkt der Kamera zu fixieren und dient somit als Hilfslink.

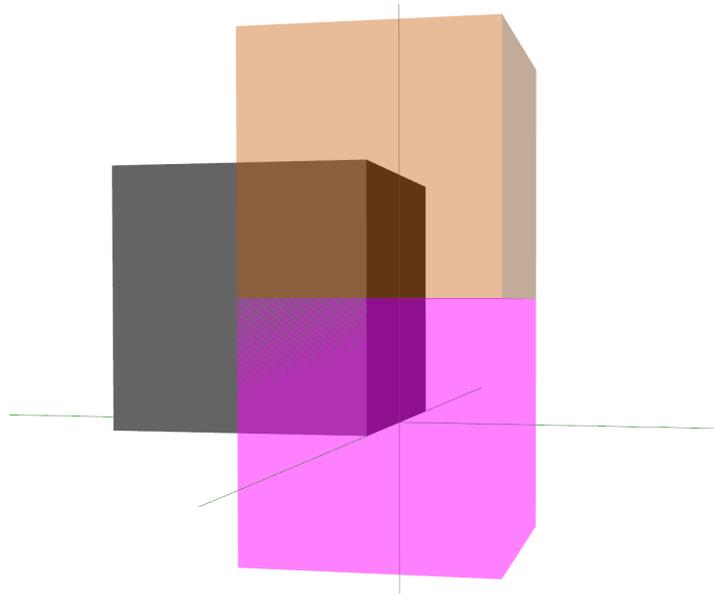


Abbildung 2.2: Darstellung von Visual (grau), Collision (orange) und Inertial (pink)

Um die Links miteinander zu verknüpfen, müssen diese jeweils mit dem selben Joint verbunden werden. Die Joints werden dann mit dem Element Pose versehen [\[16\]](#). Eine Pose ist hierbei die Kombination aus Position und Orientierung in Bezug zu dem Ursprungskoordinatensystem. Die kinematische Verbindung der beiden Elemente wird dann über eine sogenannte Parent-Child-Beziehung realisiert. Diese Beziehung spiegelt die kinematische Hierarchie des Robotermodells wider. In dem Kontext bezeichnet ‚Parent‘ das übergeordnete Glied, während ‚Child‘ das untergeordnete Glied darstellt. Eine Bewegung oder Transformation des übergeordneten Gliedes wirkt sich unmittelbar auf die Kinematik des untergeordneten Gliedes aus. Die Art und Weise wie das untergeordnete Glied beeinflusst wird, hängt wiederum vom Typ des Joints ab. Diese können als unterschiedliche Gelenktypen, wie beispielsweise als Drehgelenk, Getriebe oder als starre Verbindung ausgelegt werden. Die Abbildung [2.3](#) zeigt beispielhaft die Verbindungen der einzelnen Links und Joints und deren Beziehung zueinander. Durch gezielte Ansteuerung dieser Gelenke, mittels eines ROS-Gazebo-Plugins, können Antriebssteuerung realisiert werden [\[15\]](#), [\[16\]](#), [\[17\]](#).

2 Theoretische Grundlagen

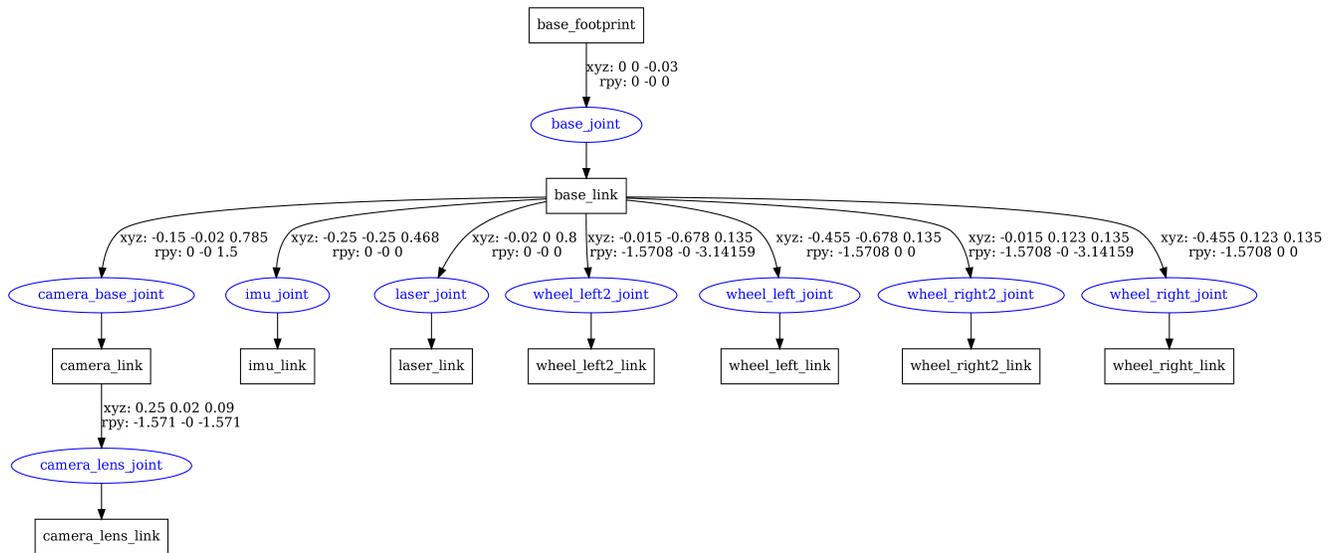


Abbildung 2.3: Darstellung der Transformation des Simulationsmodells

2.1.2 Modellerweiterung Sensor

Innerhalb der Simulationsumgebung von Gazebo stehen zahlreiche Sensorerweiterungen zur Verfügung, die in einem Simulationsmodell integriert werden können. Unter Nutzung von ROS2 besteht die Möglichkeit, auf die Sensorschnittstelle zuzugreifen, um Sensorinformationen zu empfangen und die Aktoren zu steuern. Unabhängig von den individuellen Sensorcharakteristika besitzen die Sensoren einen identischen Aufbau. Für eine realitätsnahe Darstellung besitzt jeder Sensor die Fähigkeit, dem Signal ein Rauschelement hinzuzufügen. Dies beinhaltet die Festlegung eines Mittelwertes sowie die Standardabweichung. Sensoren wie beispielsweise die IMU oder ein GNSS-Sensor besitzen zusätzlich noch die Möglichkeit, zu der systematischen und relativen Messabweichung weitere dynamische Eigenschaften dem Sensor hinzuzufügen. So ist es möglich, mit dem Attribut `dynamic_bias_correlation_time` langsame Schwankungen in einer Sensorverzerrung zu modellieren. Das Messrauschen kann individuell für jede Achse eingestellt werden [16]. Damit ist es möglich ungewollte Messausschläge einer Achse zu reduzieren, ohne die anderen Achsen dabei zu beeinflussen. Es kann zwischen drei Typen von Messrauschen ausgewählt werden. Der Typ `None` beschreibt kein Messrauschen, wodurch die Messwerte konstant und unverändert bleiben. Die Sensorwerte für den Parameter `gaussian` werden so modelliert, dass jede Messung zufällige Werte aus einer Gauss'schen Verteilung generiert. Mit den Attributen `mean` und `stddev` kann dann die Statistik der Verteilung bestimmt werden. Unter Berücksichtigung eines realen Sensors wird der Parameter `gaussian_quantized` genutzt. Hierbei erhalten die Ausgangswerte zusätzlich eine Quantisierung und werden auf festgelegte Werte gerundet. Dies ergibt sich aus der Tatsache, dass viele Sensoren in der Regel nur diskrete Werte ausgeben [18]. Im Quellcode [2.1] wird die Integration eines IMU-Sensors aus der Auswahl der frei verfügbaren Sensoren dargestellt.

Listing 2.1: IMU-Sensor in SDF-XML

```
<!-- IMU-Sensor -->
<sensor name="imu_sensor" type="imu">
  <always_on>true</always_on> <!-- Immer eingeschaltet -->
  <update_rate>255</update_rate> <!-- Aktualisierungsrate in Hz -->
  <visualize>true</visualize> <!-- In der Simulation visualisieren -->
  <imu>
    <angular_velocity>
      <x>
        <noise type="gaussian">
          <mean>2.17e-3</mean> <!-- Mittelwert Messrauschen -->
          <stddev>1.38e-3</stddev> <!-- Standardabweichung Messrauschen -->
        </noise>
      </x>
    </angular_velocity>
  </imu>
</sensor>
<plugin name="imu_plugin" filename="libgazebo_ros_imu_sensor.so">
  <ros>
    <remapping>~/out:=imu</remapping> <!-- ROS Remapping-Konfiguration -->
  </ros>
</plugin>
```

3 Hardware- und Softwarekomponenten

3 Hardware- und Softwarekomponenten

3.1 Hardwarekomponenten

3.1.1 EMMA

Im Rahmen eines Forschungsprojekts wurde das Schienenfahrzeug EMMA entwickelt (siehe Abbildung 3.1). Diese wird von zwei in Reihe geschalteten 12-V-Batterien betrieben und besitzt somit eine Gesamtspannung von 24V. Zusätzlich verfügt EMMA über eine pneumatische Bremsanlage, einen Ouster OS1 Lidar und eine Stereolabs ZED2 Kamera. Darüber hinaus ist ein leistungsstarker Einplatinencomputer, der NVIDIA Jetson Nano, auf dem Fahrgestell integriert. Dieser Computer betreibt ein GNU-Betriebssystem mit der Distribution Ubuntu 18.04 in Kombination mit ROS Melodic.

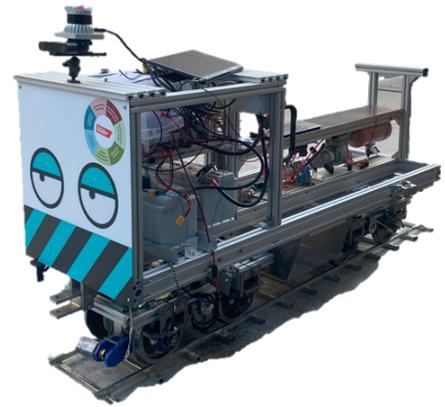


Abbildung 3.1: Schienenfahrzeug des Forschungsprojektes

Aufgrund von unterschiedlichen Versionen des Betriebssystems sowie der ROS Distribution wird für die Aufnahme der Daten eine ROS-Bag-Datei angelegt. ROS Bags ermöglichen die Erfassung von Daten während der Ausführung, die zu einem späteren Zeitpunkt und auf einem anderen Computer ausgewertet werden können. Abbildung 3.2 zeigt das Auswertungsverfahren der Sensordaten sowohl für die Hardware als auch für die Simulation.

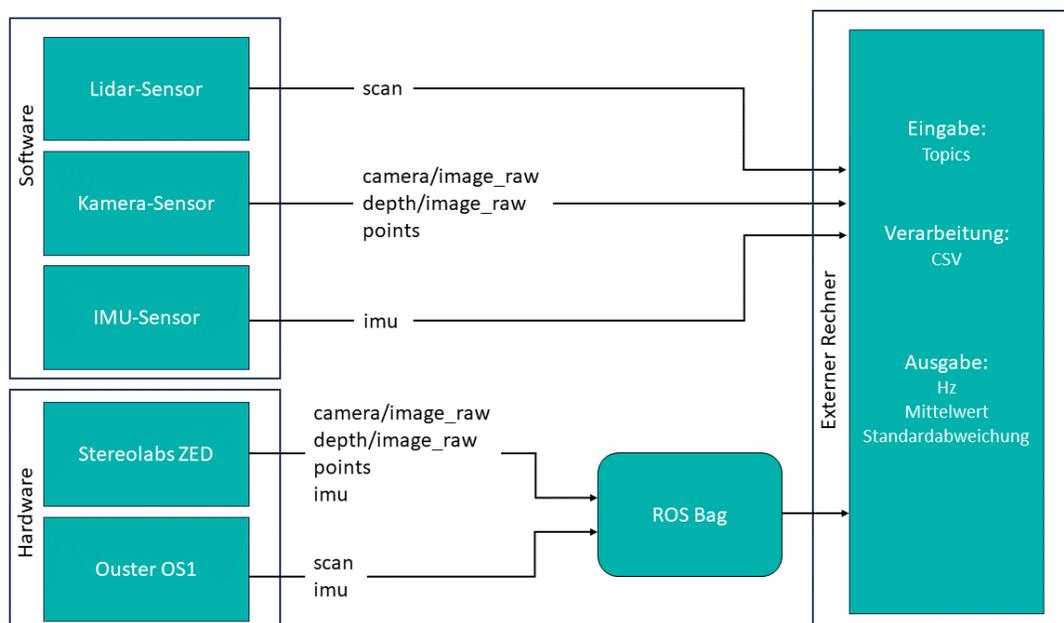


Abbildung 3.2: Auswertungsverfahren der Sensordaten

3 Hardware- und Softwarekomponenten

3.1.2 Messaufbau

Das Schienenfahrzeug EMMA befindet sich im Schienenfahrzeuglabor im Untergeschoss des Kompetenzzentrum Mobilität FH Aachen (KMAC). Außerhalb des Schienenfahrzeuglabors befindet sich der Messplatz für die Datenerfassung, der aus einem Vorbau des Gebäudes mit freiem Blick zum Himmel besteht (siehe Abbildung 3.3). Aufgrund der tiefergelegten Fläche und der Abschirmung des nebenstehenden Gebäudes befindet sich ein annähernd gleiches Lichtverhältnis am Messplatz. Das Schienenfahrzeug wird für die Erfassung in die Mitte des Messplatzes auf die dafür vorgesehenen Gleise rausgefahren. Um die Zuverlässigkeit und Qualität der Messdaten zu erhöhen, wurden alle Messungen während trockener Witterungsphasen durchgeführt. Diese Vorgehensweise ermöglicht es, mögliche Einflüsse auf die Messdaten durch sich ändernde Wetterverhältnisse zu minimieren und das Schienenfahrzeug vor Schäden durch Niederschlag zu schützen.

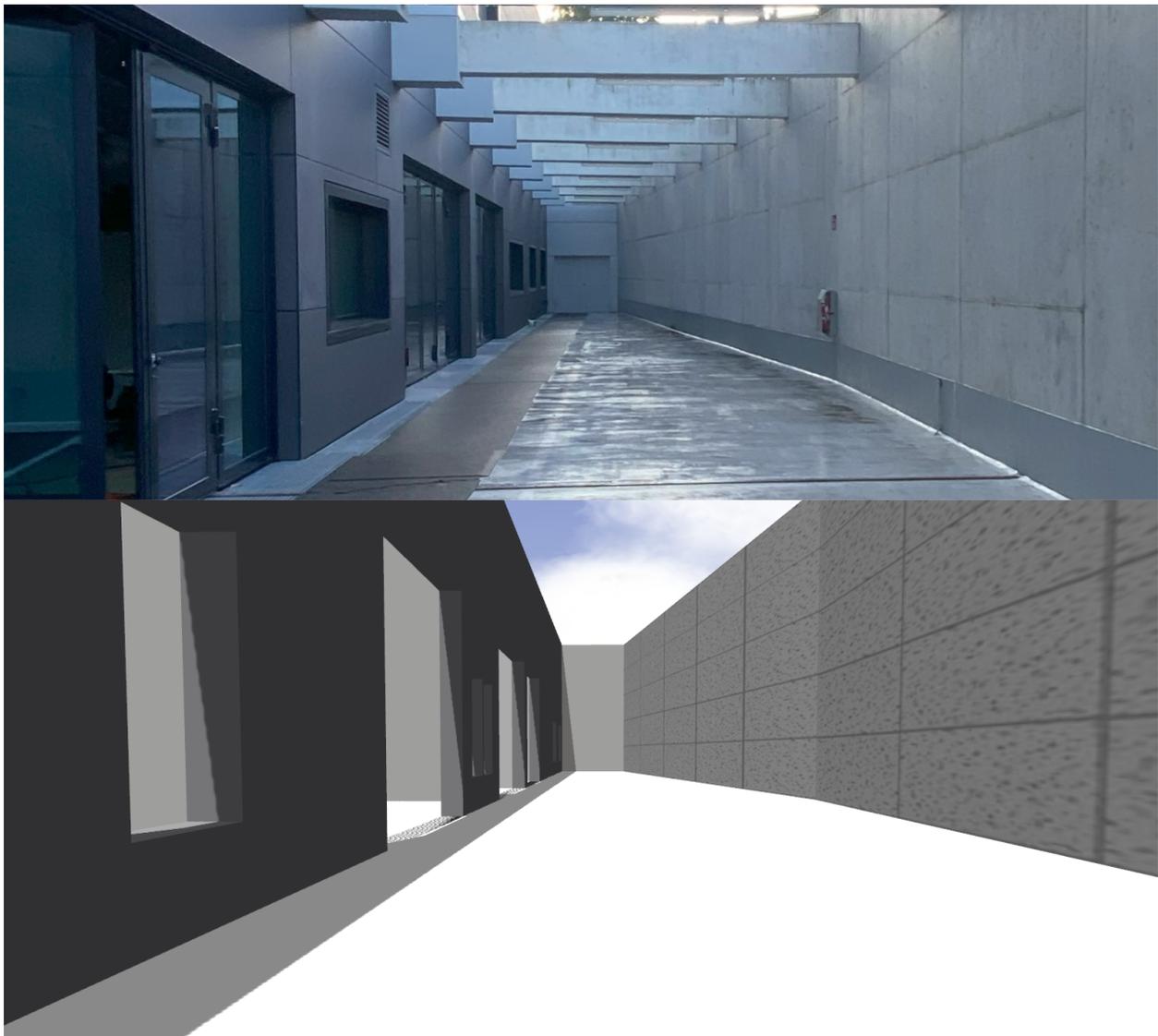


Abbildung 3.3: Messplatz am KMAC

3 Hardware- und Softwarekomponenten

3.1.3 Eingesetzte Sensorik

Bei den verwendeten Sensormodulen handelt es sich um die Stereokamera ZED2 von Stereolabs und den Lidar OS1 von Ouster (siehe Abbildung 3.4). Beide Geräte verfügen über eine integrierte IMU, welche die Messung von Beschleunigungen und Winkelgeschwindigkeiten ermöglicht.



Abbildung 3.4: Stereolabs ZED2 und Ouster OS1 [7], [19]

Die ZED2 verfügt über zwei Kameras und bietet die Möglichkeit, über verschiedene Videomodi zu wechseln. Durch eine Änderung der Auflösung der Bildwiederholungsrate (engl.: Frames per second (fps)) oder des Sichtbereichs der Kamera ist es möglich, die Verwendung der ZED2 individuell anzupassen [19]. Durch die Fusion der linken und rechten Kamera kann die ZED2 die Funktionsweise des menschlichen Sehens nachbilden. Dies ermöglicht nicht nur eine Tiefenwahrnehmung der Umgebung, sondern gibt auch einen Rückschluss über 3D-Bewegungen im Raum. Mithilfe dieser Technik ist eine Tiefenwahrnehmung von bis zu 40 Metern mit einer Abstrakte von 100 fps sowohl in Innenräumen als auch im Freien erzielbar. Dabei nimmt die Tiefengenauigkeit mit zunehmender Entfernung im Nahbereich bis 1% und im Fernbereich bis 9% ab [19]. Die Fusion der

IMU-Daten und der visuellen Tracking-Informationen der Stereokamera führen zu einer zuverlässigeren Schätzung der Kamerabewegung [19]. Die ZED2 verfügt zusätzlich über ein Magnetometer und ein Barometer. Durch die Ausrichtung nach dem Erdmagnetfeld, was eine absolute Orientierung darstellt, und der relativen Höhenmessung des Barometers kann die Tiefenkamera ihre Position und Orientierung mit höherer Zuverlässigkeit bestimmen. Das zugehörige ZED Software Development Kit (SDK) ist mit dem Jetson Nano vollständig kompatibel und kann daher problemlos direkt auf EMMA verwendet werden.

Der OS1 von Ouster ist ein Lidar mit einer mittleren Reichweite und einem 360 Grad Sichtfeld. Dieser verfügt über eine konfigurierbare horizontale Auflösung von 512, 1024 oder 2048 und eine statische vertikale Auflösung von 32 für einen Sichtwinkel von $\pm 22,5^\circ$. Dabei besitzt der OS1 eine Winkelgenauigkeit von $\pm 0,01^\circ$ sowohl in der vertikalen als auch horizontalen Abtastung. Je nach Lichtreflexion der Objekte besitzt der OS1 eine maximale Reichweite von 90 m bis 170 m und bietet dabei eine Wahrscheinlichkeit von mehr als 90% Objekte zuverlässig zu erkennen. Wahlweise kann der Sensor mit einer Abtastfrequenz von 10 Hz oder

3 Hardware- und Softwarekomponenten

20 Hz betrieben werden. Neben der Erzeugung von Punktwolken stehen auch Bilder von Informationen wie Reichweite, Intensität, Rauschen und Reflexionswerten zur Verfügung (siehe Abbildung [3.5](#)). Darüber hinaus ist die Transformation des 0-Grad-Rings der Punktwolke verfügbar, um daraus einen 2D-Laserscan zu generieren [\[7\]](#), [\[20\]](#).

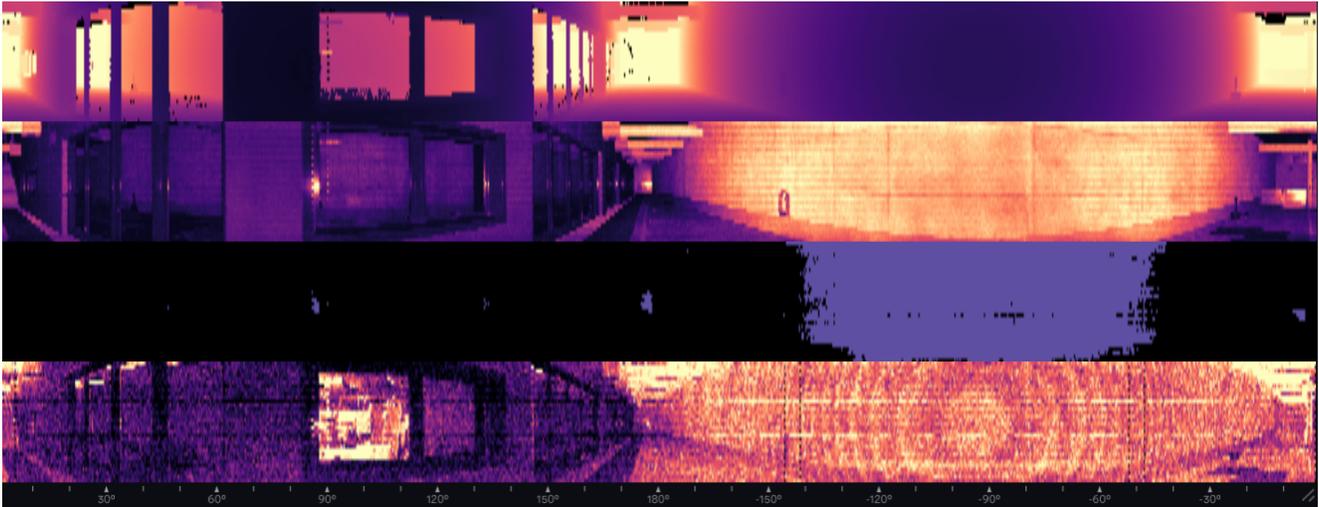


Abbildung 3.5: Sensorbilder des Ouster OS1 der Entfernungswerte, Intensitätswerte, Rauschwerte und Reflektivitätswerte am Messstandort

Die Hauptunterschiede zwischen den integrierten IMU-Modulen sind in Tabelle [3.1](#) dargestellt. Beim Vergleich zeigt sich, dass die IMU der ZED2 eine bessere Auflösung sowohl für das Accelerometer als auch für das Gyroskop bietet. Dagegen hat die IMU des Ouster OS1 eine geringere Rauschdichte, was auf eine geringere zufällige Messabweichung hindeutet. Die tatsächlichen Messabweichungen der IMUs werden im weiteren Verlauf genauer untersucht, um zum Beispiel Einflüsse auf die Messabweichungen durch die Rotation des Lidars auszuschließen.

3 Hardware- und Softwarekomponenten

Tabelle 3.1: IMU-Daten der Stereolabs ZED2 und des Ouster OS1 [6], [8], [19]

	Stereolabs ZED2	Ouster OS1
Abtastrate in Hz	400	100
Auflösung in bit	16	16
Messbereich Accelerometer in g	±8	±16
Sensitivität Accelerometer in $\frac{\text{LSB}}{g}$	4096	2048
Rauschdichte Accelerometer in $\frac{\mu g}{\sqrt{\text{Hz}}}$	320	135
Messbereich Gyroskop in °/s	±1000	±2000
Sensitivität Gyroskop in $\frac{\text{LSB}}{(\text{°/s})}$	32,76	16,38
Rauschdichte Gyroskop in $\frac{(\text{°/s})}{\sqrt{\text{Hz}}}$	0,16	0,005
Magnetometers	✓	✗
Barometer	✓	✗

3.2 Softwarekomponenten

3.2.1 Robot Operating System

ROS bzw. ROS2 ist ein weitverbreitetes Programmierwerkzeug für die Entwicklung und Steuerung von Robotern. Dabei handelt es sich um eine sogenannte Middleware. Die Middleware besteht aus einer Softwareschicht und dient als Schnittstelle zwischen dem Betriebssystem und der Benutzeroberfläche. ROS besteht aus einer Vielzahl von Bibliotheken, Hardwaretreibern, Entwicklungsmethodiken und verfügt über Entwickler- und Darstellungswerkzeuge. Zusätzlich bietet ROS eine flexible Wahl der Programmiersprachen C++, Python und weiteren gängigen Sprachen. ROS2 ist die Weiterentwicklung von ROS und bietet eine Kompatibilität mit verschiedenen Betriebssystemen sowie eine bessere Skalierbarkeit, Sicherheit und Echtzeitanforderungen. Aufgrund dessen, dass ROS auf der aktuellsten GNU Distribution Ubuntu LTS 22.04 nicht mehr unterstützt wird, wurde sich für die Erstellung der Simulation für ROS2 entschieden [21].

3 Hardware- und Softwarekomponenten

3.2.2 Gazebo

Gazebo ist eine Open-Source-Plattform, die speziell für die Nutzung der Robotik- und Automatisierungsbranche entwickelt wurde. Die 3D-Simulationsumgebung verfügt über die Möglichkeit Robotermodelle, Sensoren, Aktoren und deren physikalische Eigenschaften zu simulieren. Gazebo stellt einige Bibliotheken zur Verfügung, um den Datenaustausch zwischen Aktoren und Sensoren sowie die Wahrnehmung der Umwelt zu gewährleisten. Des Weiteren beinhaltet es ein Entwicklertool, um einfache Modelle und Simulationsumgebungen innerhalb der Anwendung zu entwickeln. Gazebo dient ROS als Standardplattform für die virtuelle Darstellung von Robotern und Umgebungen und spielt somit eine wichtige Rolle bei der Entwicklung von autonomen Systemen und Robotern. Die Integration von Gazebo mit ROS2 lässt eine nahtlose Kommunikation mit ROS2-Komponenten und Anwendungen zu. Jede ROS-Distribution besitzt seine eigene offizielle Gazeboversion, die während der gesamten Lebensdauer vollständig unterstützt wird [17]. Die verwendete Version in diesem Projekt ist für ROS2 Humble Gazebo Fortress Version. Gazebo unterstützt die Formate URDF und SDF.

3.2.3 Gazebo-Train

Der Gazebo-Train entstand im Rahmen eines Studentenprojektes und dient als Grundlage für diese Bachelorarbeit. Hierbei handelt es sich um eine Schienenfahrzeug-Simulation, die auf Gazebo basiert und ROS2 integriert (siehe Abbildung 3.6). Der Zug verfügt über mehrere Möglichkeiten der Steuerung, wie zum Beispiel mittels eines Gaming-Controllers, einer Tastatureingabe oder einem Graphical User Interface (GUI).

Die GUI erscheint beim Starten der Simulationsumgebung automatisch in der oberen rechten Bildschirmecke und enthält Schaltflächen zum Starten, Stoppen, Richtungswechseln sowie verschiedene Geschwindigkeitsstufen. Zusätzlich verfügt das Simulationsmodell einen Lidar, eine Stereokamera und eine IMU.

Mithilfe des Gazebo-Buildereditors konnten zu den bereits existierten Simulationsumgebungen weitere erstellt werden. Mittels technischer Zeichnung können Maßstäbe der Pixelanzahl angepasst werden und somit detailgetreue Nachbildungen erstellt werden. Die Erstellung der Simulationsumgebung ist in Abbildung 3.7 dargestellt.

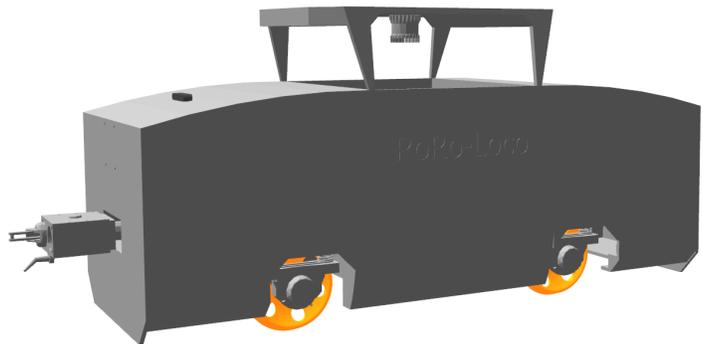


Abbildung 3.6: Simulationsmodell Gazebo-Train

3 Hardware- und Softwarekomponenten

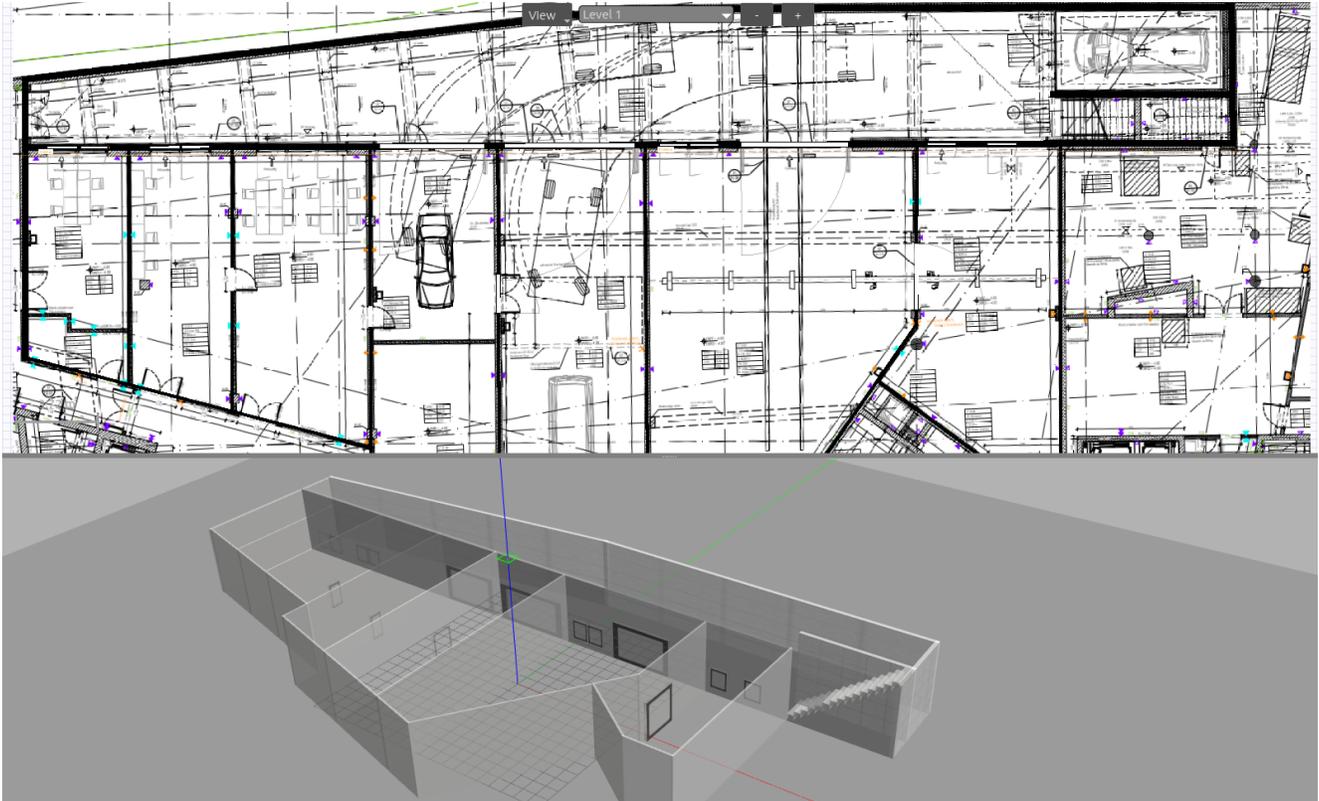


Abbildung 3.7: Erstellung einer Simulationsumgebung mit dem Gazebo-Buildeditor

4 Methodik

4.1 Simulationsmodell

4.1.1 Kalibrierung des Simulationsmodells

Die Kalibrierung der Sensoren und Aktoren des Simulationsmodells erfolgt durch die aufgenommenen Messdaten des Schienenfahrzeugs. Hierfür werden jeweils die Daten des Lidars und der Stereokamera sowie deren integrierte IMU über einen Zeitraum von 300 Sekunden aufgenommen. Während dieser Aufnahme befindet sich der Zug im Stillstand. Anschließend wird der Zug verschoben und erneut eine Aufnahme der Daten über den oben genannten Zeitraum durchgeführt. Zur Auswertung der Messdaten wird Python verwendet.

Die Kalibrierung und Einstellung jedes Sensors erfolgt gemäß dem in Abbildung 4.1 dargestellten Verfahren. Zunächst wird die Datenübertragungsrate des realen Sensors ermittelt, um diese dann auf das Modell zu übertragen. Anschließend werden an dem simulierten Sensor die Mittelwerte und Standardabweichungen eingestellt. Da innerhalb der Messzeit genügend viele Einzelmessungen durchgeführt worden sind, kann von einer Normalverteilung der Messwerte ausgegangen werden. Aufgrund dessen wird in das Attribut des Sensorrauschens der Typ `gaussian` eingestellt, welcher die Gauß-Verteilung darstellt [22]. Danach erfolgt eine erste Messung des simulierten Sensors ohne, dass sich der Zug auf den Gleisen befindet. Damit können erste Signalstörungen durch die Kontakte zwischen Radsatz und Gleis eliminiert werden. Die nach der ersten Messung ermittelten Werte werden mit den realen Sensordaten verglichen. Anschließend werden die Parameter des simulierten Sensors so angepasst, dass dessen Werte mit denen des realen Sensors übereinstimmen. Dieses Verfahren wird so lange wiederholt bis die gewünschte Genauigkeit erzielt wurde. Im nächsten Schritt wird derselbe Test durchgeführt, jedoch diesmal während sich der Zug auf den Gleisen befindet. Die weiteren Mes-

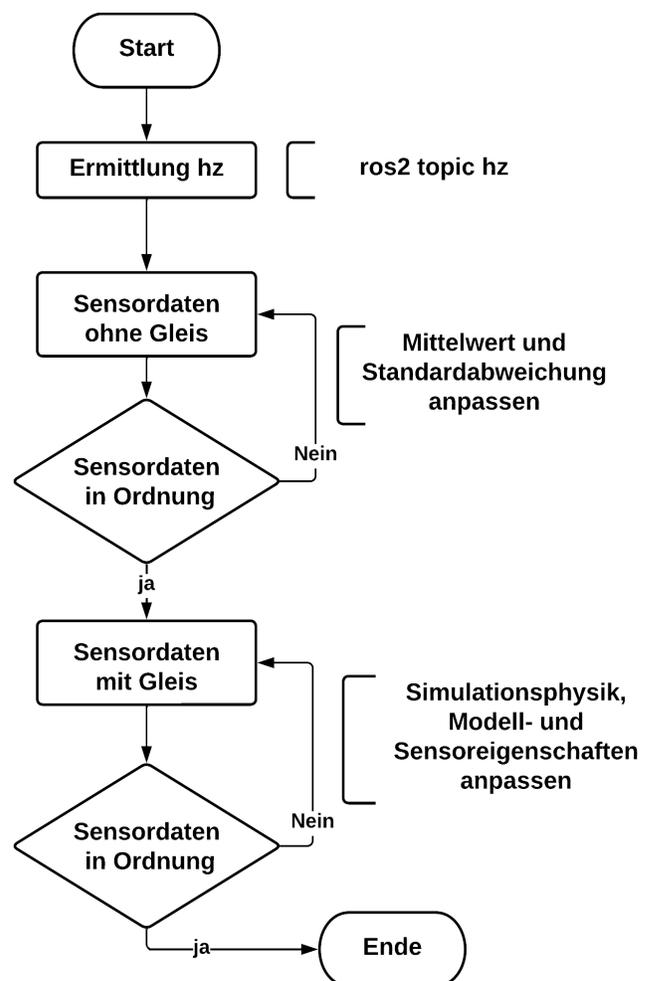


Abbildung 4.1: Vorgehensweise für die Sensoreinstellung des Modells

4 Methodik

sungen werden erneut mit den Messwerten des realen Sensors verglichen. Dabei werden die Simulationsphysik, Modell- und die Sensoreigenschaften so angepasst, dass eine Übereinstimmung zwischen den Sensordaten des simulierten Sensors und den Messwerten des realen Sensors erreicht ist. Dieser Anpassungsprozess wird iterativ wiederholt, bis die gewünschte Genauigkeit erreicht ist.

4.1.2 Kalibrierung der IMU

Um die Sensoren der Simulation kalibrieren zu können, ist es von entscheidender Bedeutung, dass das Schienenfahrzeug sich in einem stabilen Zustand befindet. Es muss sichergestellt werden, dass die Simulation keine unerwünschten vertikalen Sprünge oder diskontinuierliche Bewegungen aufweist. Dies kann durch die Auswertung der IMU-Daten erfolgen. Hierfür werden zunächst die IMU-Daten der ZED2 ohne Lidar und anschließend mit eingeschaltetem Lidar ausgewertet. Das Ergebnis der Auswertung ist in Abbildung 4.2 dargestellt. Es ist deutlich zu erkennen, dass durch die Rotation des Lidars die Messabweichungen höher sind als die der Stereokamera. Aufgrund dessen wird mit der Auswertung der IMU-Daten der Stereolabs ZED2 fortgefahren. Des Weiteren wird der Aufbau so geändert, dass die Rotation des Lidars keinen Einfluss auf die Messabweichung der ZED2-IMU-Daten hat.

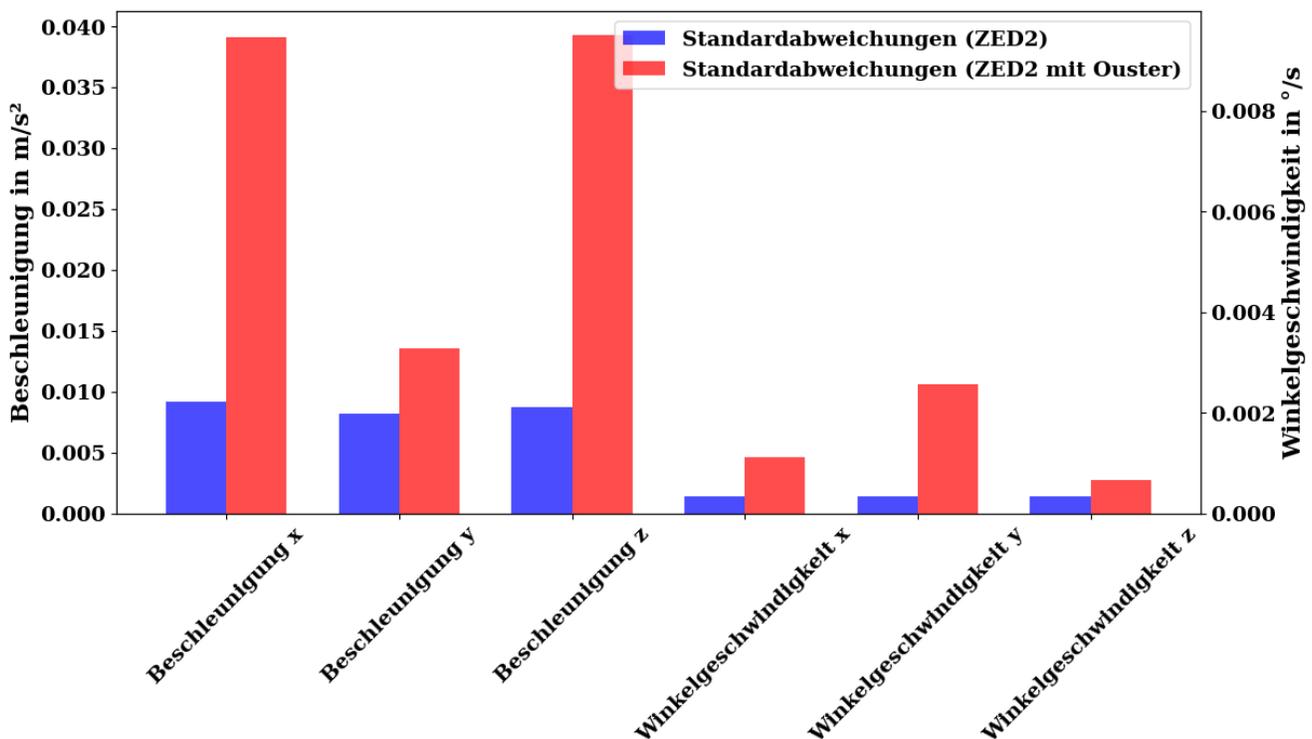


Abbildung 4.2: Vergleich der IMU-Daten der ZED2 und ZED2 mit Ouster

Die Frequenz der eintreffenden IMU-Daten kann auf verschiedene Weisen ermittelt werden. Eine Möglichkeit besteht darin, den ROS2-Befehl `ros2 topic hz /imu` zu verwenden,

4 Methodik

um die Datenrate des IMU-Topics zu überprüfen. Alternativ dazu kann die Stichprobenanzahl durch die benötigte Zeit dividiert werden, um die Frequenz der ankommenden Daten abzuschätzen. Die Frequenz wird anschließend in das Simulationsmodell übernommen. In Tabelle 4.1 sind die Mittelwerte und Standardabweichungen sowie die minimalen und maximalen Werte dargestellt. Die Werte werden verwendet, um das Attribut des Sensorrauschens in dem Simulationsmodell einzustellen.

Tabelle 4.1: Zusammenfassung der ZED-Daten

	Beschleunigung			Winkelgeschwindigkeit		
	x in m/s ²	y in m/s ²	z in m/s ²	x in 1/s	y in 1/s	z in 1/s
count	75000	75000	75000	75000	75000	75000
mean	0,095	0,165	9,789	-0,000	0,000	0,000
std	0,009	0,009	0,009	0,001	0,001	0,001
min	0,055	0,126	9,753	-0,006	-0,006	-0,006
max	0,137	0,219	9,824	0,006	0,006	0,006

In Abbildung 4.3 ist die erste Messung ohne Verwendung der Gleise dargestellt. Die Stichprobengröße von 75000 ergibt sich hierbei aus einer Datenübertragungsrate von ca. 250 Hz. Aufgrund von vertikalen Ausschlägen und einer Verschiebung des Mittelwertes wird der Anpassungsprozess iterativ wiederholt. Stellt sich nach der Iteration der gewünschte Wert ein, wird dieser überprüft. Hierzu wird wie in dem realen Messversuch das Schienenfahrzeug um eine gewisse Länge verschoben und die Messdaten erneut aufgenommen. Werden auch hier die gewünschten Werte erzielt, so kann mit dem nächsten Verfahren fortgefahren werden.

4 Methodik

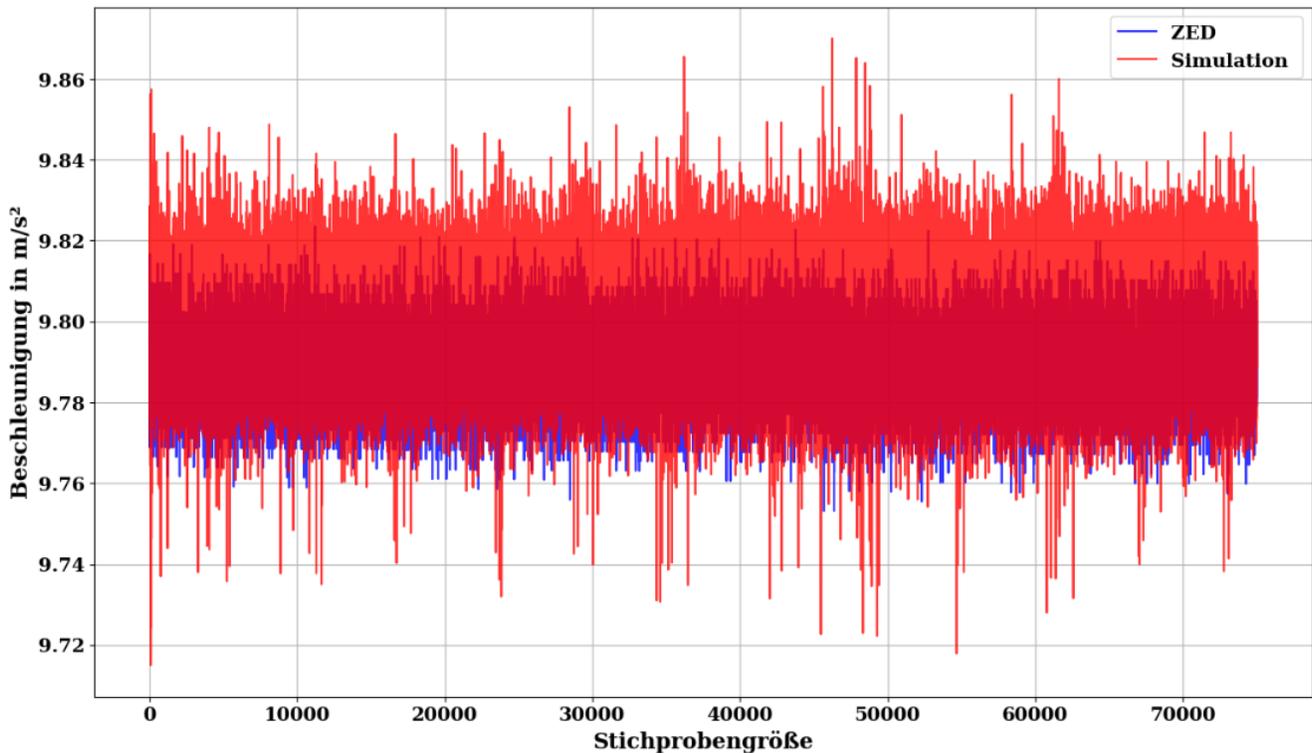


Abbildung 4.3: Lineare Beschleunigung in Z-Richtung nach erster Messung ohne Gleise

Nachdem die Konfiguration des Sensors in der Simulationsumgebung ohne Gleise durchgeführt wurde, werden die ermittelten Parameter in der Simulationsumgebung mit Gleisen getestet. Das Ergebnis für die lineare Beschleunigung in z-Richtung der ersten Messung mit Gleisen ist in [Abbildung 4.4](#) dargestellt. Deutlich zu erkennen sind die vertikalen Sprünge, die das Messsignal verfälschen. Die Gründe hierfür könnten die Einstellungen der Simulationsparameter, die Modellparameter oder die Kontakteinstellungen zwischen Schienenfahrzeug und Gleis sein. Da der Sensor in der vorherigen Messung korrekt eingestellt wurde, müssen nun die Parameter der Simulationsumgebung und des Modells überprüft werden. Aufgrund dessen, dass das Schienenfahrzeug in der Simulation auf Gleisen steht, werden alle Parameter, die Einfluss auf die Kontaktstellen haben, zuerst geprüft.

4 Methodik

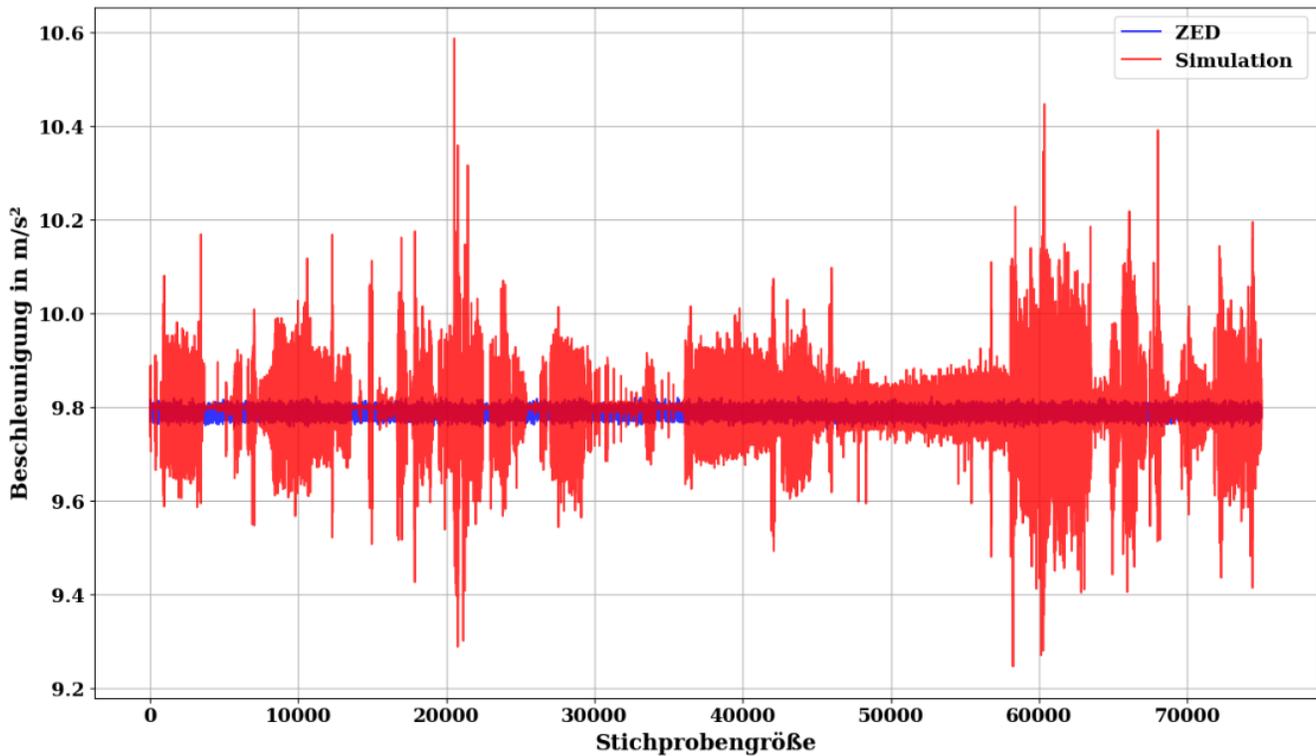


Abbildung 4.4: Lineare Beschleunigung in Z-Richtung nach erster Messung auf den Gleisen

Um die Richtung der Zugbewegung und die linearen Beschleunigungen sowie die Winkelgeschwindigkeiten zu verdeutlichen, werden diese separat betrachtet. In [Abbildung 4.5](#) sind die Beschleunigungen des simulierten Schienenfahrzeuges im Vergleich zur ZED2 dargestellt. Dabei sind gerichtete Bewegungen in den x- und y-Richtungen und einige Ausreißer in z-Richtung zu erkennen. Ebenfalls signifikant sind die Streuungen der Simulationsmodell-IMU im Vergleich zur ZED2-IMU.

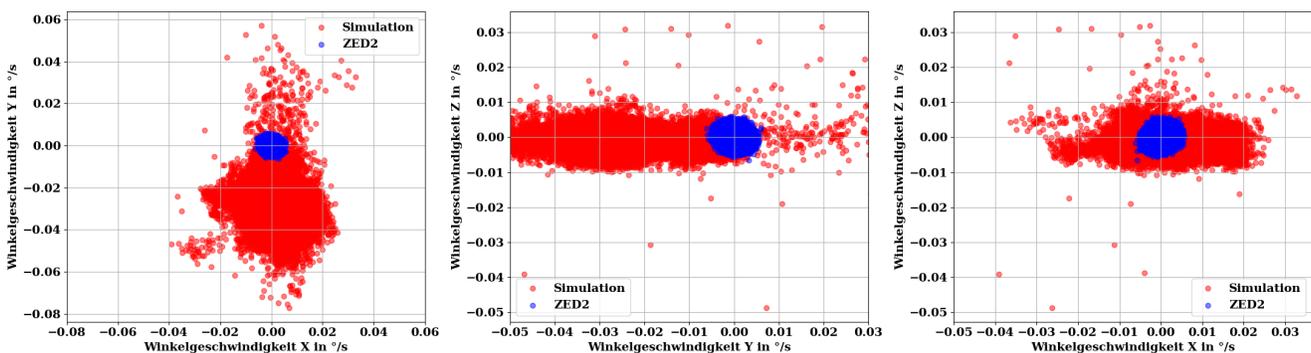


Abbildung 4.5: Linearen Beschleunigungen nach erster Messung auf den Gleisen

Eine Möglichkeit, dieses Problem zu minimieren, besteht darin, die `max_step_size` in der Konfiguration für die physikalische Simulation zu verändern. Die Bedeutung dieses Parameters liegt darin, dass er die Balance zwischen Genauigkeit und Geschwindigkeit in der

4 Methodik

Simulation beeinflusst. Ein kleinerer Wert führt zu kleineren Zeitschritten und erhöht die Genauigkeit der Simulation, führt aber auch zu einem höheren Rechenaufwand.

Eine weitere Möglichkeit, die Berechnung der Simulationsphysik anzupassen, besteht darin, die Parameter `contact_surface_layer` und `contact_max_correcting_vel` zu optimieren. Die Erhöhung dieser Werte kann dazu beitragen, Jittering-Probleme zu vermeiden, die durch das wiederholte Herstellen und Unterbrechen von Kontakten entstehen [16]. Sollte eine Änderung der genannten Attribute nicht zum gewünschten Ziel führen, muss mit der Einstellung der Modellparameter fortgefahren werden.

Anschließend erfolgt eine neue Datenmessung und Auswertung. Die Werte der Simulationsphysik werden iterativ angepasst, bis das gewünschte Ergebnis erreicht wird. Das Ergebnis nach acht Iterationsschritten ist der Abbildung 4.6 zu entnehmen und weist eine deutliche Verbesserung der Streuung auf. Es ist zu erkennen, dass die Abweichung um den Mittelwert sowie die Standardabweichung sich kaum von den Messdaten der ZED2 unterscheiden. Auf Basis der bestandenen Übereinstimmung zwischen den Mittelwerten und Standardabweichungen kann von einem funktionierendem Modellsensor ausgegangen werden. Darauffolgend wird das Modell in Kapitel 5 getestet und die Güte des Sensors bestimmt.

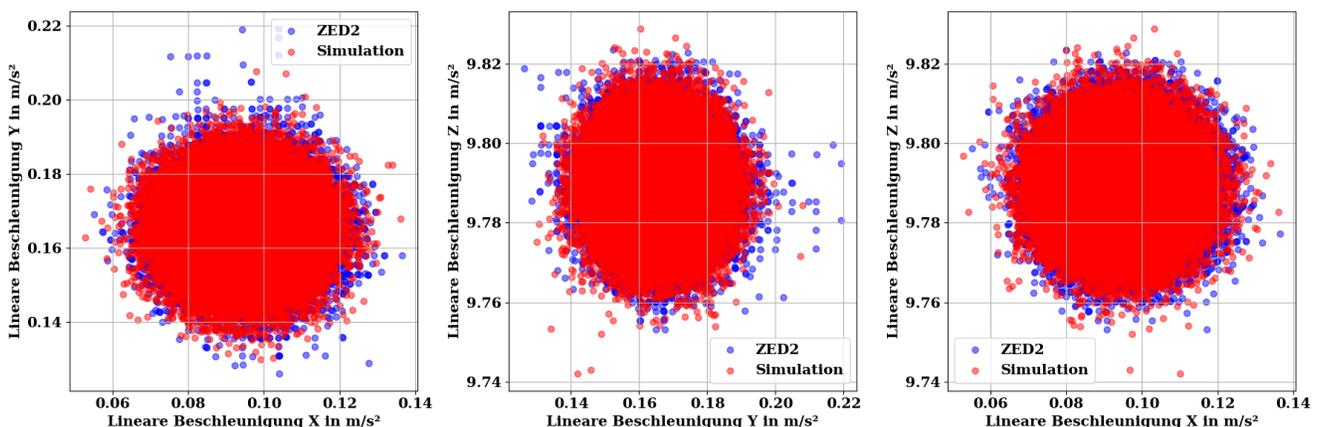


Abbildung 4.6: Abweichung der linearen Beschleunigungen nach erster Messung

4.1.3 Kalibrierung des Lidars

Die Kalibrierung des Lidars erfolgt auf ähnliche Weise wie die der IMU. Nach der Aufnahme der Messdaten werden zunächst die Eigenschaften des Sensors ermittelt und für den simulierten Sensor übernommen. Dadurch, dass in der Simulation nur die Datenerfassung eines 2D-Laserscans verfügbar ist, wird im folgenden Verlauf nur dieser betrachtet. Für die Scanfrequenz wurde eine Datenübertragungsrate von 10 Hz ermittelt und für die Anzahl der Punkte pro Umdrehung ein Wert von 1024 Punkten. Des Weiteren kann durch die Summation der Einzelstandardabweichungen jedes Strahls und anschließende Division der Anzahl

4 Methodik

der Punkte pro Umdrehung die durchschnittliche Standardabweichung bestimmt werden.

Die ermittelten Werte werden in das Simulationsmodell übernommen und durch sich wiederholende Messungen iterativ angepasst. Die Ergebnisse des ersten Vergleichs der Standardabweichungen sind in Abbildung 4.7 dargestellt. In Hinblick auf die durchschnittlichen Standardabweichungen liegen die Ergebnisse, in beiden Fällen, über dem maximalen Wert der Herstellervorgabe von 30 mm. Während die relativen Messabweichungen des Simulationsmodells homogen verteilt sind, weisen die relativen Messabweichungen des Ousters große Schwankungen auf. Durch die Betrachtung ausgewählter Messflächen konnten diese Schwankungen auf die verglasten Türen und Fenster zurückgeführt werden. Hierbei stellt sich ebenfalls eine homogenere Verteilung mit einer geringeren relativen Messabweichung ein. Aufgrund dessen, dass die Einflüsse der verglasten Oberflächen die Messergebnisse des Ousters verfälschen, werden die Messdaten auf einen ausgewählten Bereich reduziert. Dadurch ist eine Identifizierung der realen Messabweichungen möglich.

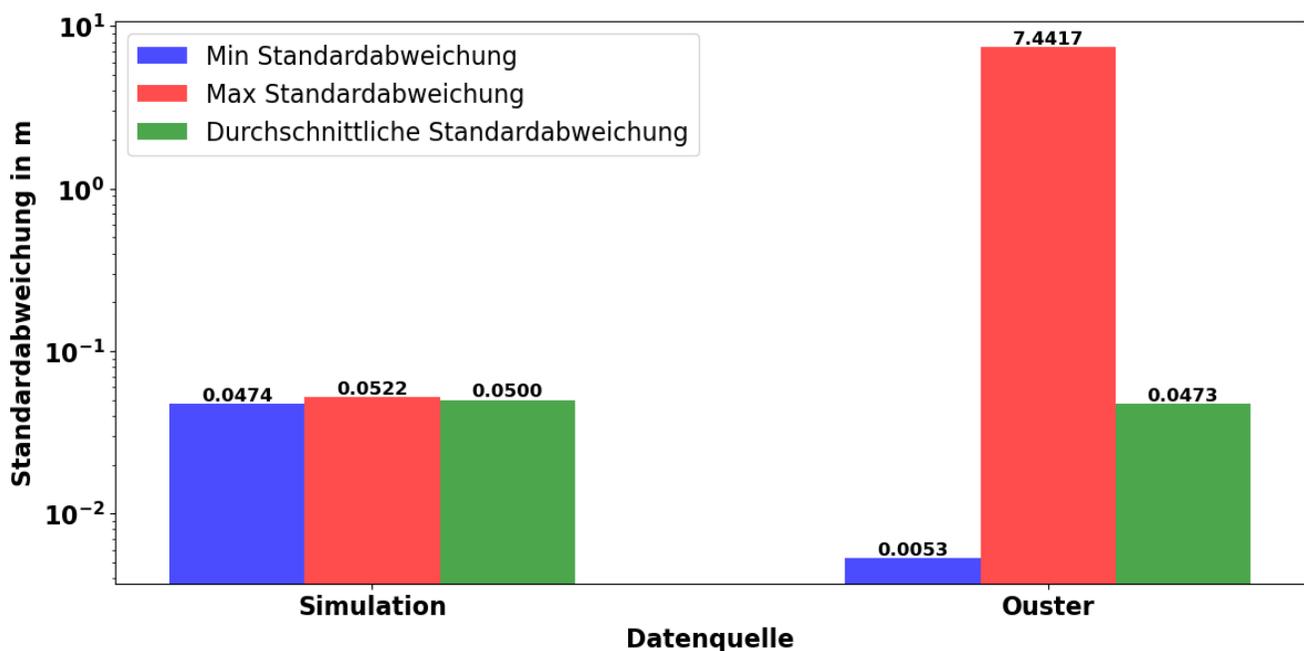


Abbildung 4.7: Standardabweichungen der Lidar-Daten der Simulation und des Ousters

Da der OS1 eine vertikale Auflösung von 32 Ringen besitzt, kann zwischen den Transformationen des 2D-Scans zwischen 32 Ringen gewählt werden. Dabei erfolgt die Ringnummerierung von oben nach unten. Damit der Scan einen möglichst großen Bereich abdeckt und dem Simulationsmodell entspricht, werden die horizontalen Ringe näher untersucht. In dieser Hinsicht sind die mittleren Ringe 14 bis 18 relevant. In Abbildung 4.8 ist der voreingestellte Scanning und der ausgewählte Scanning dargestellt. Auffällig ist der deutlich größere Scanbereich des ausgewählten Rings. In Hinblick auf die Genauigkeit des Scans fällt auf,

4 Methodik

dass die 3D-Punktwolke im größeren Scan deutlich mehr Anomalien aufweist. Ein Grund hierfür können die verglasten Türen und Fenster sein, welche sich auf der Höhe des mittleren Rings befinden. Eine Möglichkeit diesen Einfluss zu untersuchen besteht darin, die verschiedenen Bilder des Lidars, wie beispielsweise Reflektivität oder die Intensität der Bereiche genauer zu betrachten. Alternativ wäre die Datenerfassung an einem anderen Messplatz denkbar.

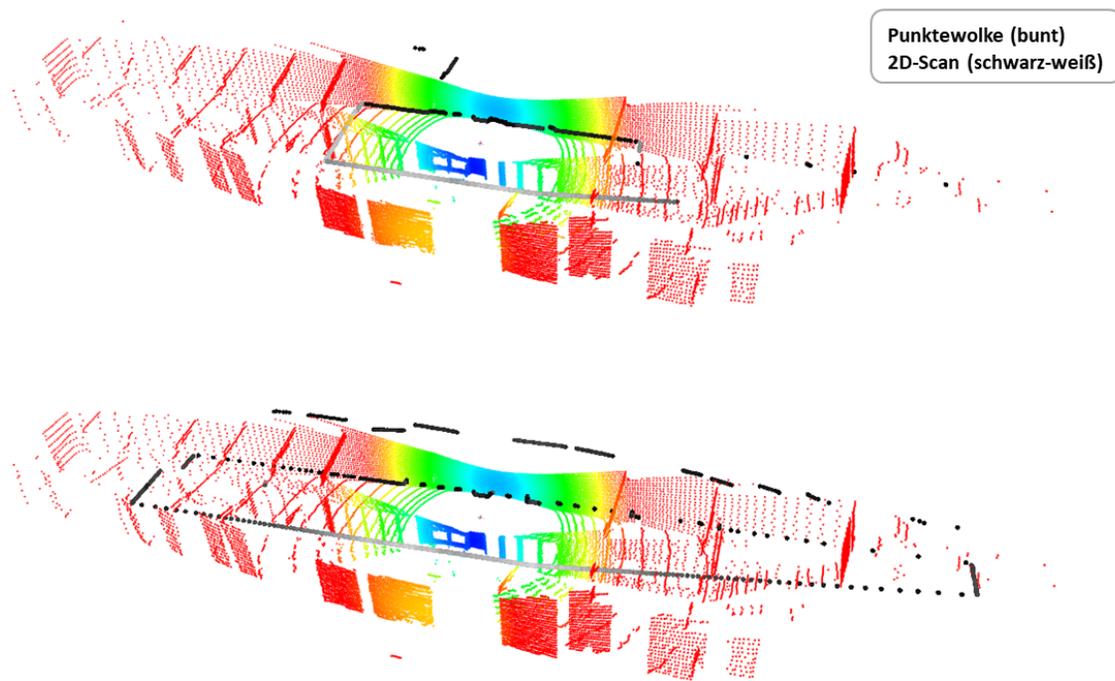


Abbildung 4.8: Punktwolken des Messplatzes mit dem Scan des 0-Rings (oben) und des 16-Rings (unten)

Die Güte des Scans in der Simulation kann durch einen Iterative-Closest-Point (ICP)-Registrierungsalgorithmus analysiert werden. Dieses Verfahren wurde bereits für die Evaluation von Punktwolken durch den Abgleich konventioneller 3D-Modelle verwendet [23]. Dabei handelt es sich um ein iteratives Annäherungsverfahren, das mithilfe von Rotation und Translation die Quell-Punktwolke auf die Ziel-Punktwolke transformiert [24]. Um die Fehleranfälligkeit der verwendeten Algorithmen zu evaluieren, werden jeweils verschiedene Methoden von zwei Softwareanwendungen durchgeführt. Die Python Bibliothek Open3D stellt die zwei Methoden Point-to-Point und Point-to-Plane zur Verfügung [25]. Beide Methoden verwenden die Berechnungen der Residuen und Jacobimatrizen zum Transformieren der Punkte. Der Point-to-Point-Algorithmus durchläuft im Allgemeinen zwei Schritte. Zunächst wird die Korrespondenzmenge aus der Zielpunktwolke und der transformierten Quellpunktwolke ermittelt. Im zweiten Schritt wird die Transformation durch Minimierung der Residuen optimiert. Der Point-to-Plane-Algorithmus hingegen minimiert den Abstand zwischen der Punktwolke und

4 Methodik

einer Ebene, wobei die Ebene eine angepasste Struktur der Ziel-Punktwolke ist [24]. Dabei zeichnet sich der Point-to-Plane-Algorithmus durch eine schnellere Konvergenzgeschwindigkeit als der Point-to-Point-Algorithmus aus.

CloudCompare stellt eine Anwendung zur Verarbeitung von Punktwolken und dreidimensionaler Geometrien dar. Hier besteht die Möglichkeit, dem Algorithmus die maximale Anzahl von Iterationen oder den Schwellenwert der Fehlerdifferenz zu übergeben. Anschließend kann der Abstand der Punkte beider Punktwolken berechnet werden. Der Berechnung des Abstandes ermöglicht die Ableitung des Mittelwerts sowie der Standardabweichung der Punktwolken [26].

Die Voraussetzung beider Softwareanwendungen ist die Verwendung der gleichen Objekte und eine grobe Ausrichtung der Punktwolken. In Abbildung 4.9 ist die Transformation der Quell-Punkt wolke zur Zielpunkt wolke des Point-to-Point-Algorithmus dargestellt. Dabei ist die Zielpunkt wolke der 2D-Scan der Simulation und die Quell-Punkt wolke der 2D-Scan des Ouster OS1. Zusätzlich wird eine Auswertung der Hälfte der 2D-Scans durchgeführt, um Fehlerquellen durch Ausreißer aufgrund von Refraktion und Reflexion zu untersuchen. Das ausgewählte Segment stammt von der festen Betonwand des Messplatzes.

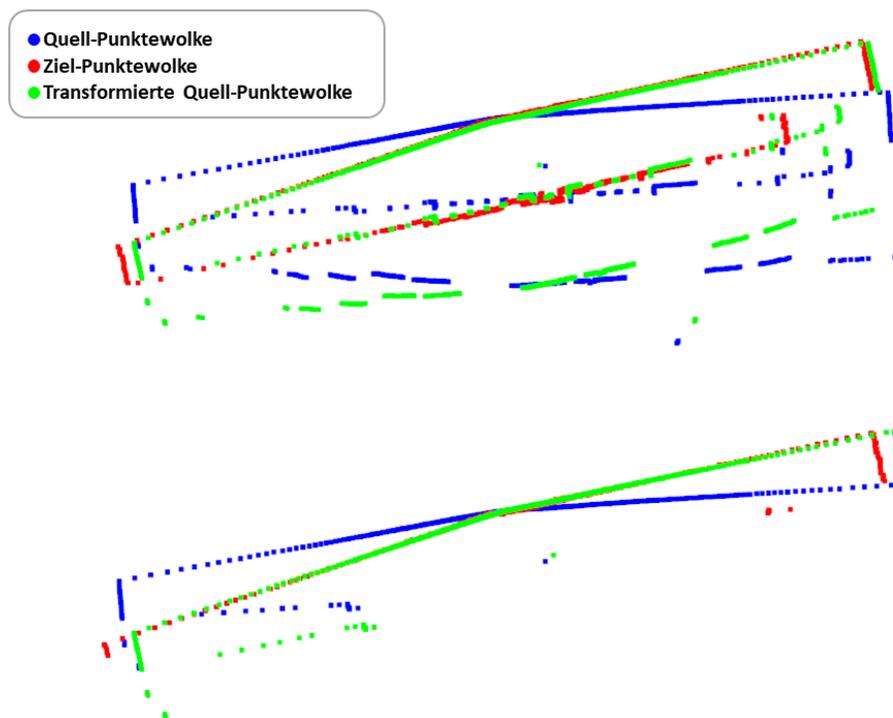


Abbildung 4.9: ICP-Registrierungsalgorithmus des ganzen Scans (oben) und halben Scans (unten)

4 Methodik

4.1.4 Kalibrierung der Stereokamera

Für die Kalibrierung des Kameramodells der Simulation müssen die Kameraparameter der ZED2 bestimmt werden. Hierfür stellt ROS eine Kamerakalibrierung für Stereo-Kamerasysteme auf der Basis von OpenCV, einer Python-Bibliothek für Bildverarbeitung und Computer-Vision-Aufgaben, zur Verfügung. Mithilfe der Kamerakalibrierung können die Parameter der intrinsischen Kameramatrix, die Verzerrung, die Rektifikationsmatrix und Projektionsmatrix bestimmt werden. Diese Informationen sind notwendig, um Verzerrungen und Unregelmäßigkeiten in den Bildern zu korrigieren und die Beziehungen zwischen den 3D-Weltpunkten und der 2D-Bildebene abbilden zu können [27]. In Abbildung 4.10 ist die Kamerakalibrierung mithilfe eines sogenannten Checkerboards dargestellt.



Abbildung 4.10: Kamerakalibrierung mit ROS

Um die Genauigkeit der simulierten Kamera zu beurteilen, kann eine Überprüfung der Abstandmessung mithilfe der Tiefenkameras durchgeführt werden. Diese Verfahren wurde bereits eingesetzt, um die Genauigkeit von Tiefenkameras zu bestimmen [28]. Neben der Kamerakalibrierung können die Fehler in den Entfernungsdaten aus den Einflüssen von Lichtquellen und glatter oder heller Oberflächen stammen [28]. Hierfür werden auf dem Messplatz jeweils drei Markierungen in unterschiedlichen Abständen zur Kamera aufgestellt. Für eine bessere Identifizierung der Abstandsmarkierungen im Bild wurden die farbigen Gestelle mit viereckigen Markern, sogenannten Apriltags, versehen. Anschließend daran werden die Bild- und Tiefeninformationen über einen kurzen Zeitraum aufgenommen. Mithilfe eines Pythoprogramms ist es möglich, durch die Selektion bestimmter Pixel im Bild, die Tiefeninformationen ausgewählter Pixel zu bestimmen. In der Simulationsumgebung wird das gleiche Verfahren zur Abstandsmessung durchgeführt. Für jede Abstandsmessung werden jeweils drei verschiedene Pixel, mit einer Stichprobengröße von 100, untersucht. Dadurch können Einflüsse wie das Sensorrauschen oder Eingabefehler entdeckt und minimiert werden. Um einen möglichst großen Messbereich abzudecken, werden im ersten Versuch Messungen mit einem Abstand von 3 Metern zueinander durchgeführt. Im zweiten Versuch hingegen werden Messungen im Nahbereich mit einem Abstand von 75 cm zueinander vollzogen. In Abbildung 4.11 ist der Messaufbau sowohl im Schienenfahrzeuglabor des KMAC als auch der Simulationsumgebung dargestellt.

4 Methodik

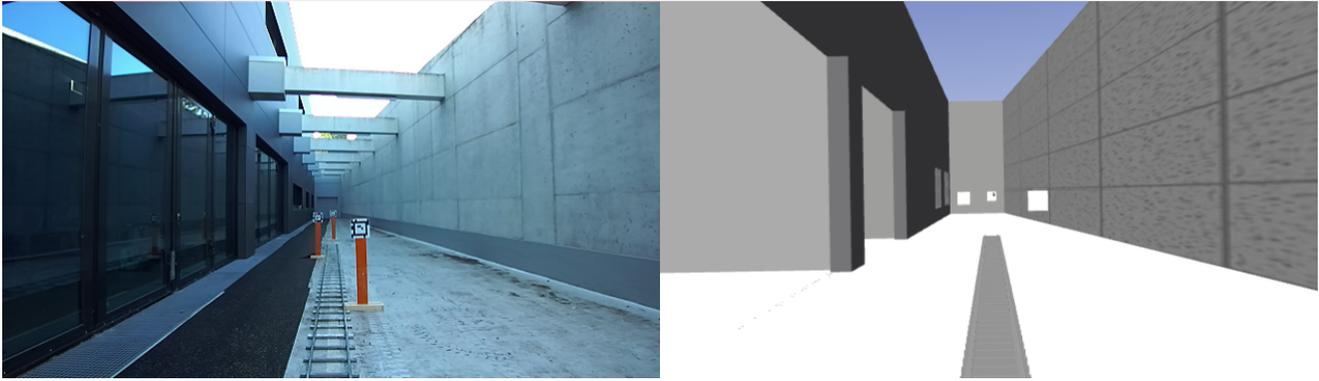


Abbildung 4.11: Aufbau des Messplatzes Schienenfahrzeuglabor (links) und Simulation (rechts)

Um die Grenzen der Abstandsmessung des realen Schienenfahrzeuges zu visualisieren, wurde RViz2, eine grafische Oberflächenanwendung von ROS, verwendet. [Abbildung 4.12](#) zeigt die Abstandsmessung mit jeweils 3 m Abstand zwischen den Markierungen und die mit jeweils 75 cm Abstand. Unter Betrachtung der erzeugten Punktwolke der Stereokamera fällt auf, dass der Informationsgehalt mit zunehmender Entfernung abnimmt. Dies ist durch die geringere Punktdichte und teilweise fehlenden Punkte im hinteren Bereich zu erkennen. Es ist deutlich sichtbar, dass die Punktwolke in der Messung mit geringerem Abstand einen höheren Informationsgehalt aufweist. Im Gegensatz dazu sind die hinteren Markierungen in der Messung mit größerem Abstand kaum bis gar nicht sichtbar. Eine Überprüfung der Abstandsmessung für diesen Messplatz ist somit nur bis maximal 10 m sinnvoll. Die Messung eines größeren Abstandes sollte daher mit dem Ouster OS1 durchgeführt werden, da dieser sowohl eine größere Reichweite als auch eine geringere Messabweichung aufweist. Des Weiteren sind Unstimmigkeiten zwischen dem Messumfeld und der Punktwolke im Bereich der verglasten Flächen zu erkennen. Dies deutet ebenfalls darauf hin, dass durch die Effekte, wie beispielsweise Spiegelung, eine korrekte Abstandsbestimmung nicht durchgeführt werden kann.

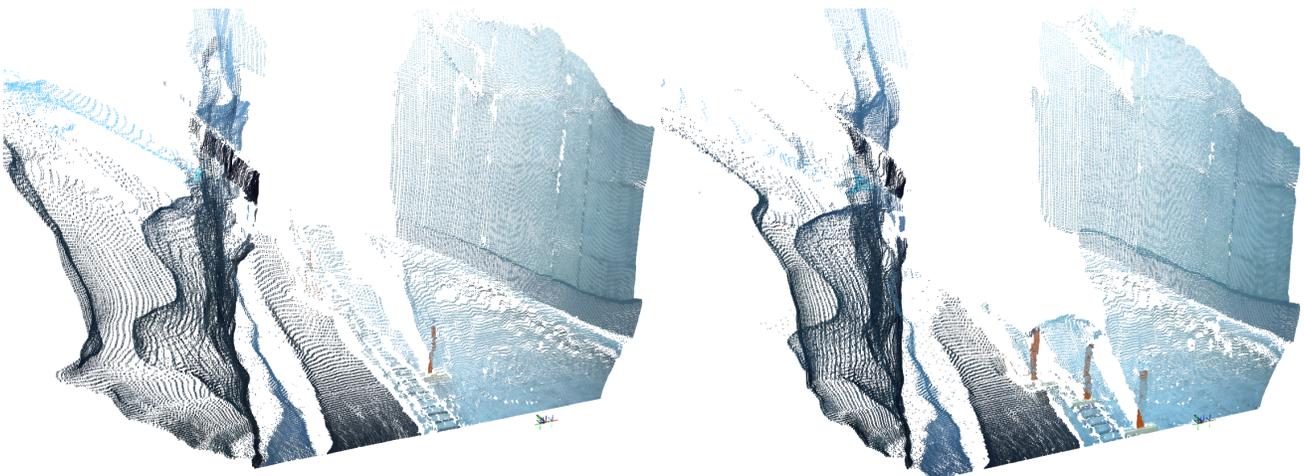


Abbildung 4.12: Punktwolke der Distanzerkennung am Messplatzes

5 Evaluation

In dieser Arbeit wurden Methoden für die Kalibrierung der Simulationsumgebung erläutert und mithilfe der ermittelten Parameter ein Simulationsmodell erstellt. Damit das Modell in der Realität zuverlässig einsetzbar ist, wird in diesem Kapitel das Simulationsmodell und die Simulationsumgebung validiert und die Grenzen des erstellten Modells erläutert.

5.1 Beurteilung der Ergebnisse

5.1.1 Beurteilung des IMU-Sensors

Um eine qualitative Aussage über die Güte des simulierten Sensors zu erhalten, werden die relativen und absoluten Messabweichungen miteinander verglichen. Eine Gegenüberstellung der Mittelwerte der simulierten und realen IMU ist in Abbildung 5.1 dargestellt. Dabei besitzen die Messabweichungen der Mittelwerte der Beschleunigungen eine maximale Differenz von $7,5 \times 10^{-04} \text{ m/s}$. Somit liegen die absoluten Messabweichungen der Beschleunigungswerte innerhalb der Standardabweichung von $0,031 \text{ m/s}$ und erfüllen damit die Anforderungen an das Simulationsmodell. Unter Betrachtung der Mittelwerte der Winkelgeschwindigkeiten stellt sich eine maximale absolute Messabweichung von $5,15 \times 10^{-04} \text{ }^\circ/\text{s}$ ein. Auch dieser Wert ist innerhalb des zulässigen Schwellenwertes von $0,16 \text{ }^\circ/\text{s}$ und kann somit als zufriedenstellend betrachtet werden. Die niedrigste Differenz der Mittelwerte weist dabei die Winkelgeschwindigkeit in x-Richtung auf. Mit einer absoluten Messabweichung von $2,19 \times 10^{-06} \text{ }^\circ/\text{s}$ stellt diese Komponente der IMU die deutlich genaueste Anpassung an den realen Sensor dar.

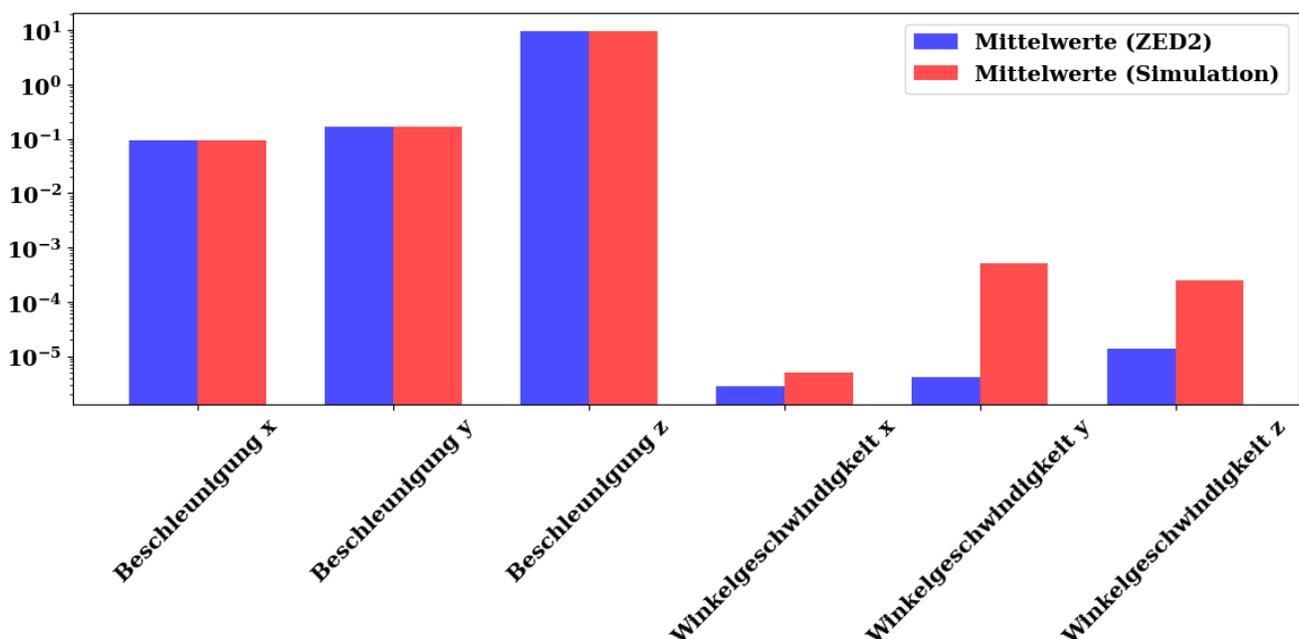


Abbildung 5.1: Absolute Messabweichung der IMU-Daten

5 Evaluation

Ein ähnliches Verhalten ist bei den relativen Messabweichungen der Standardabweichungen zu beobachten. Für die Beschleunigungen stellt sich in y-Richtung die größte Differenz mit $3,35 \times 10^{-04} \text{ m/s}$ ein. Dies ist annähernd ein hundertstel kleiner als der geforderte Schwellenwert und dadurch auch ein Indikator für eine gute Sensorkalibrierung. Zuletzt werden die relativen Messabweichungen der Winkelbeschleunigung betrachtet. Hierfür wurde für eine maximale Differenz der Winkelgeschwindigkeit in z-Richtung von $6,00 \times 10^{-06} \text{ }^\circ/\text{s}$ ermittelt. In Hinblick auf die signifikante Unterschreitung des Schwellenwertes von $0,16 \text{ }^\circ/\text{s}$ kann ebenfalls von einer erfolgreichen Sensorkalibrierung ausgegangen werden.

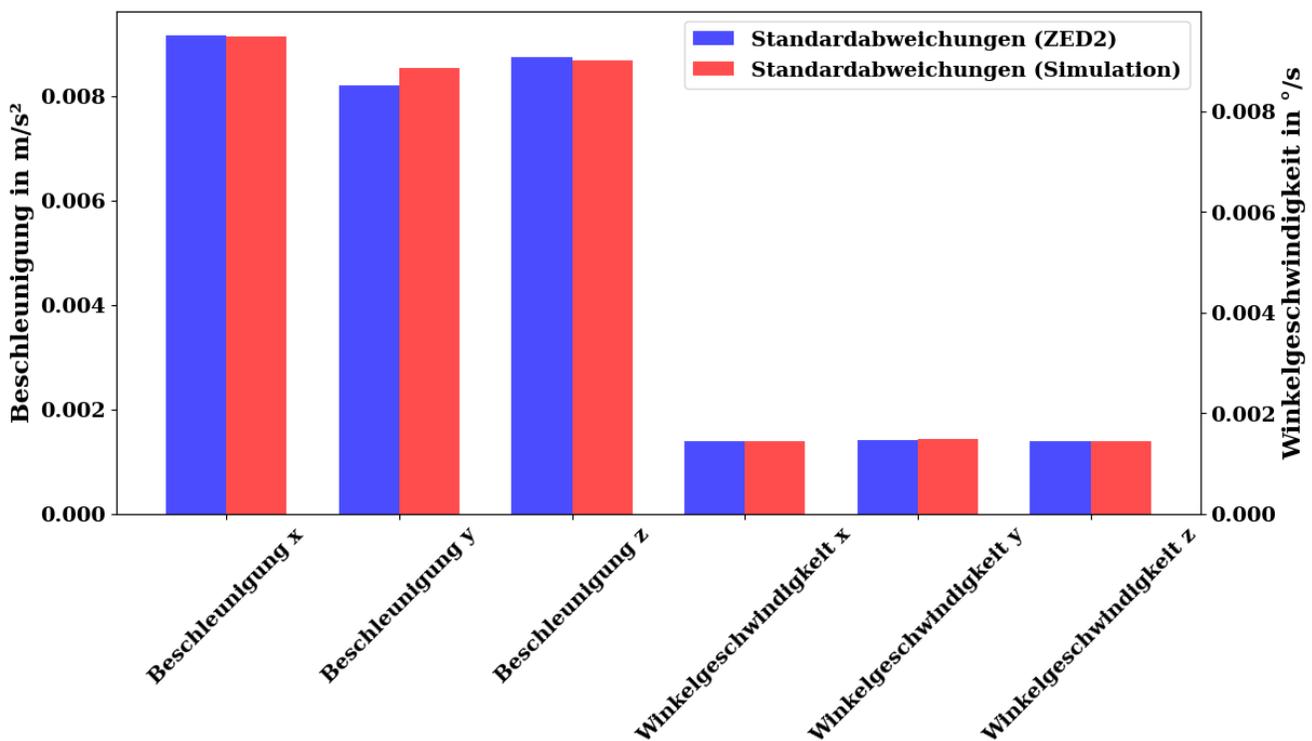


Abbildung 5.2: Relative Messabweichung der IMU-Daten

Sowohl die absoluten als auch die relativen Messabweichungen liegen deutlich unter dem Schwellenwert. Eine Auflistung aller Komponenten und deren Messabweichungen sind in Tabelle 5.1 dargestellt. Daraus lässt sich schlussfolgern, dass die Simulationen der Beschleunigung sowie der Winkelgeschwindigkeiten eine gute Annäherung an den realen Sensor darstellen. Die Ergebnisse unterstreichen die Qualität des simulierten Sensors und dessen Fähigkeit, die Bewegungen des Schienenfahrzeugs abbilden zu können.

Tabelle 5.1: Differenzen der IMU-Daten zwischen der ZED2 und der Simulation

	Mittelwert-Differenz	Standardabweichung-Differenz
Beschleunigung in x-Richtung in m/s²	$7,58 \times 10^{-04}$	$3,00 \times 10^{-05}$
Beschleunigung in y-Richtung in m/s²	$3,03 \times 10^{-04}$	$3,35 \times 10^{-04}$
Beschleunigung in z-Richtung in m/s²	$3,64 \times 10^{-04}$	$4,70 \times 10^{-05}$
Winkelgeschwindigkeit in x-Richtung in %s	$2,19 \times 10^{-06}$	$3,00 \times 10^{-06}$
Winkelgeschwindigkeit in y-Richtung in %s	$5,15 \times 10^{-04}$	$4,00 \times 10^{-06}$
Winkelgeschwindigkeit in z-Richtung in %s	$2,37 \times 10^{-04}$	$6,00 \times 10^{-06}$

5.1.2 Beurteilung des Lidars

Die Beurteilung des simulierten Lidars erfolgt auf zwei verschiedene Arten. Zum einen kann die relative Messabweichung der 2D-Scans analysiert werden. Zum anderen kann die Kompatibilität der Scandaten der Simulation und des Ousters untersucht werden. Die Ergebnisse der Auswertung der Standardabweichungen sind in Abbildung [5.3](#) dargestellt. In Hinblick auf die durchschnittliche Standardabweichung fällt auf, dass eine Anpassung an den realen Lidar mit einer Messabweichung von unter 1 mm erzielt wurde. Damit befindet sich die Messabweichung zwischen den Sensoren unterhalb des Schwellenwertes von 30 mm. Die jeweilige durchschnittliche relative Messabweichung von ca. 10 mm sind in beiden Fällen unter dem maximalen Wert des Herstellers von 30 mm. Während die relativen Messabweichungen des Simulationsmodells homogen verteilt sind, weisen die relativen Messabweichungen des Ousters größere Schwankungen auf. Die Überschreitung des maximalen Wertes der relativen Messabweichung des Ousters lässt sich auf nicht vollständig bereinigte Daten zurückführen. Aufgrund der geringen Messabweichung zueinander und dadurch, dass sich die durchschnittliche relative Messabweichung unterhalb des vorgeschriebenen Schwellenwertes von 30 mm befindet, kann von einer guten Anpassung des simulierten Sensors ausgegangen werden.

5 Evaluation

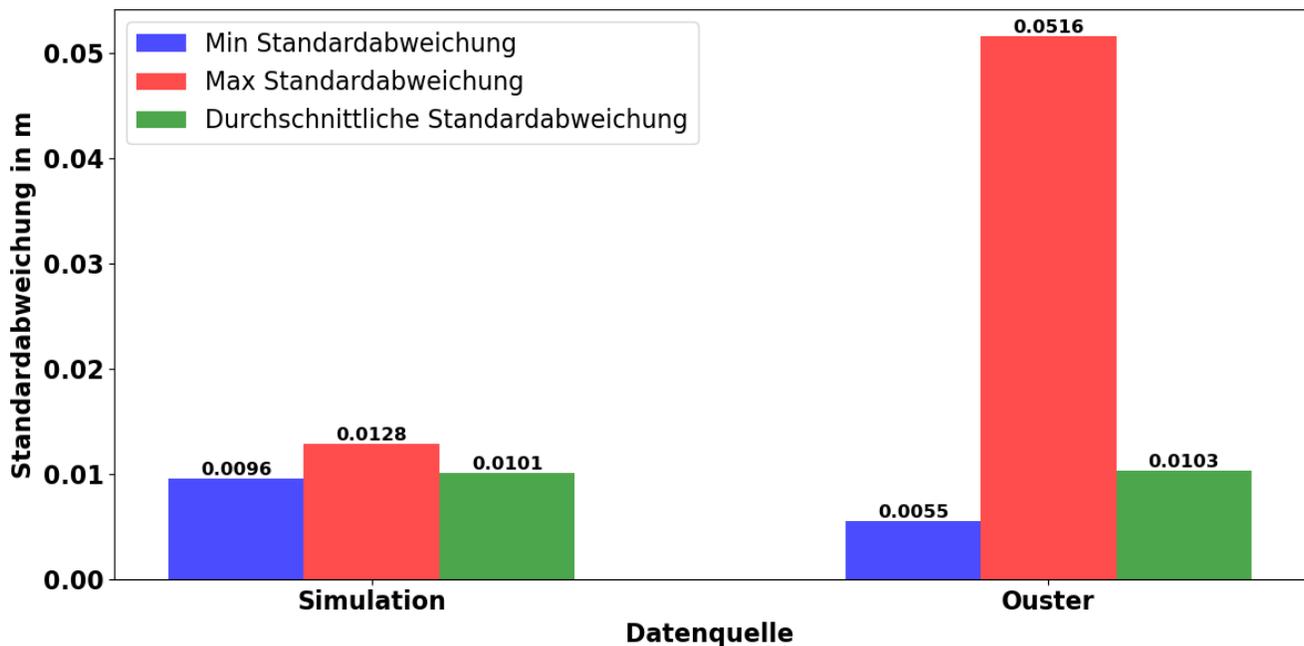


Abbildung 5.3: Standardabweichungen der Lidar-Daten der Simulation und des Ousters

Um die Güte der 2D-Scans und die Simulationsumgebung zu beurteilen, erfolgt die Datenauswertung der ICP-Algorithmen. In Tabelle 5.2 sind die Ergebnisse der ausgewählten Algorithmen dargestellt. Je kleiner die Wurzel des mittleren quadratischen Fehlers (engl.: Root Mean Squared Error (RMSE)), desto genauer ist die Korrespondenz der gesamten Quell-Punktwolke mit der Ziel-Punktwolke. Durch das Quadrieren der Abweichungen der Open3D-Algorithmen werden größere Abweichungen stärker gewichtet. Dies führt wiederum zu einem größeren RMSE. Die Werte der Point-to-Point-Methode zeigen, dass eine gute Anpassung der Quell-Punktwolken mit diesem Verfahren erzielt wird. Hinsichtlich des geringen RMSE und einer großen Übereinstimmung (engl.: Fitness) der Punktwolken kann für die Point-to-Point Methode der Ring 16 als gute Näherung betrachtet werden. Dabei kann der Wert für die Übereinstimmung zwischen 0 (keiner Übereinstimmung) und 1 (vollständiger Übereinstimmung) variieren. Mit einem RMSE von 0,26 m weist dieser die geringste Abweichung zur Ziel-Punktwolke auf und befindet sich unterhalb des Grenzwertes von 55 cm Messabweichung. Trotz eines größeren RMSE in der Point-to-Plane Methode liegt auch hier die größte Übereinstimmung der Punktwolken des 16. Scanrings vor. Bei Betrachtung der übrigen Ergebnisse fällt auf, dass diese ebenfalls ein RMSE unterhalb des Grenzwertes besitzen und dabei eine Übereinstimmung der Punktwolken von mindestens 95 % aufweisen. Diese gilt sowohl für die Ergebnisse der Point-to-Point als auch für die der Point-to-Plane-Methode. Eine mögliche Erklärung hierfür könnte die geringe Neigung der Scanringe von $0,71^\circ$ sein, welche die Fläche und Form der 2D-Scans bei dem verwendeten Messplatz nur minimal verändern [7].

5 Evaluation

Tabelle 5.2: Vergleich der 2D-Scans des Ouster OS1 und dem Simulationsmodell

	Open3D Point-to-Point		Open3D Point-to-Plane		CloudCompare	
	RMSE in m	Fitness	RMSE in m	Fitness	Mittelwert in m	Std in m
Ring 14	0,293	0,982	0,321	0,982	0,184	0,217
Ring 15	0,291	0,983	0,497	0,953	0,182	0,210
Ring 16	0,260	0,991	0,396	0,988	0,189	0,204
Ring 17	0,307	0,982	0,320	0,986	0,189	0,217
Ring 18	0,289	0,984	0,515	0,951	0,175	0,180

Mithilfe der Ergebnisse von CloudCompare können sowohl die relativen als auch die absoluten Messabweichungen miteinander verglichen werden. Unter Berücksichtigung aller Werte fällt auf, dass sich die Mittelwerte zwischen 17 cm und 19 cm befinden und somit unterhalb des Schwellenwertes von 55 cm liegen. In Bezug auf die relative Messabweichung stellt sich eine Standardabweichung zwischen 18 cm und 22 cm ein. Aus den ermittelten Ergebnissen kann eine maximale Gesamtabweichung von ca. 41 cm abgeleitet werden. Unter Berücksichtigung der zweifachen Standardabweichung, bei der sich 95% der ermittelten Werte innerhalb dieses Bereichs befinden, ergibt sich eine Gesamtabweichung von etwa 54 cm.

Der ICP-Algorithmus von Open3D stellt zusätzlich die Auswertung des RSME unter Berücksichtigung der verwendeten Punkte zur Verfügung. Damit können mögliche Ausreißer, welche beispielsweise durch die Reflexion der Verglasung entstanden sind, minimiert werden. Das Ergebnis der sogenannten Inlier RMSE ist in Abbildung 5.4 dargestellt. Es fällt auf, dass für beide Verfahren eine Verbesserung der Werte erzielt wird. Besonders für das Point-to-Plane Verfahren erfolgt eine Verbesserung hinsichtlich der Streuungen zwischen den 2D-Scanringen. Die maximale Differenz des RMSE zwischen den Ringen beträgt für das Point-to-Plane Verfahren ca. 4 cm, während es für das Point-to-Point Verfahren lediglich ca. 1 cm beträgt.

5 Evaluation

Angesichts der zulässigen RMSE der ICP-Algorithmen von Open3D und die Höhe der Korrespondenz zwischen der Quell- und der Ziel-Punktwolken kann insgesamt von einem guten Simulationsmodell des Ouster OS1 ausgegangen werden. Darauf stützend zeigen die Ergebnisse von CloudCompare eine ebenfalls gute Konfiguration des Simulationsmodelles. Eine realitätsnahe Datenanalyse mithilfe des Simulationsmodells ist somit möglich.

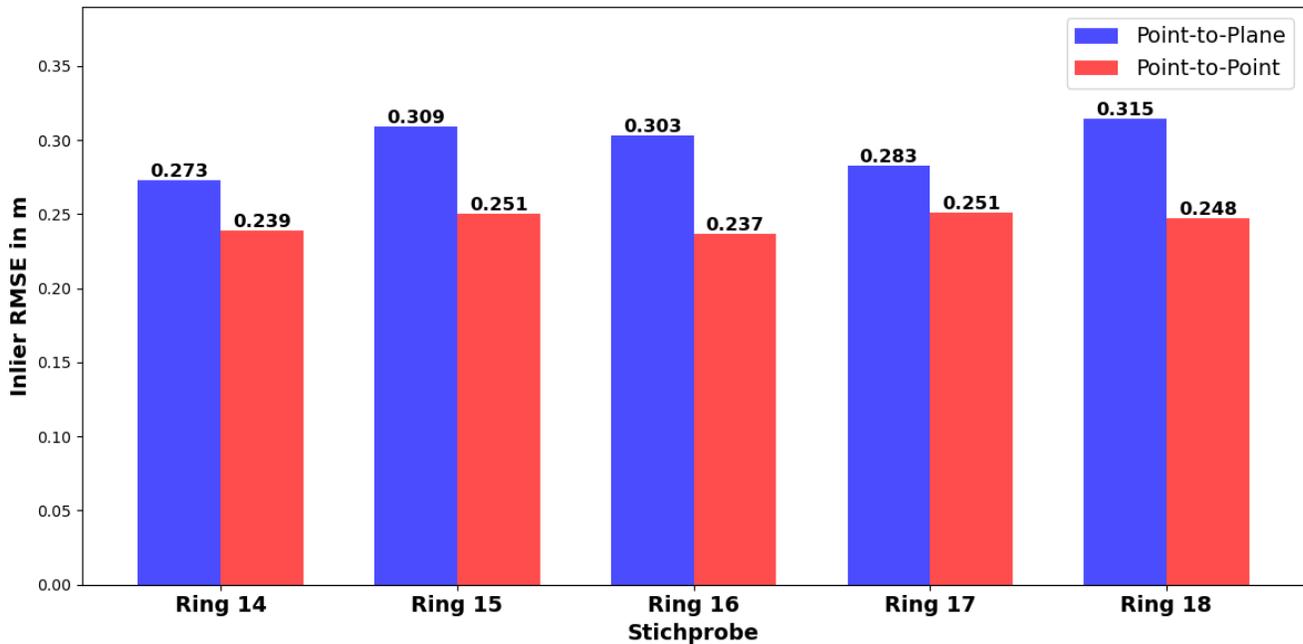


Abbildung 5.4: RSME der verwendeten Punkte der Punktwolken

5.1.3 Beurteilung der Stereokamera

Die Beurteilung des Simulationsmodells der Stereokamera erfolgt anhand der Abstandsmessungen. In [Abbildung 5.5](#) sind die durchschnittlichen Distanzen zwischen ZED und Markierung sowie Simulationsmodell und Markierung dargestellt. Sowohl die Messungen des realen Sensors als auch die des Simulationsmodells sind innerhalb der zulässigen Messabweichungen. Ausnahme hierbei stellt die Messung des 2,50 m Abstandes der ZED2 dar. Diese unterschreitet die zulässige Messabweichung von $2,50 \text{ m} \pm 0,025 \text{ m}$ um 2,5 cm. Ein Grund hierfür könnten Positionierungsfehler der Abstandsmarkierungen sein, die sich im Bereich von $\pm 5 \text{ cm}$ auswirken. Eine weitere Möglichkeit für diese Abweichung könnte ein übermäßig hohes Messrauschen, ausgelöst von den verglasten Flächen in dem untersuchten Bereich, sein [\[29\]](#). Verursacht durch sich ändernde Umwelteinflüsse, wie beispielsweise die Lichtverhältnisse, weist die reale Stereokamera größere Messdifferenzen auf. So befinden sich die Differenzen des simulierten Kameramodells für alle Werte unter 1 %, während sich die Messabweichung der ZED2 mit zunehmender Entfernung vergrößert. Befindet sich

5 Evaluation

die Gesamtdifferenz zwischen dem minimalen und maximalen Messwert im Nahbereich bei ca. 3%, steigt diese bis zur letzten Messung auf eine Gesamtabweichung von bis zu 16 % an. Trotz der Fehlmessung bildet das Simulationsmodell die Herstellerspezifikationen und die zulässige Messabweichung zwischen realem und simuliertem Sensor präzise ab. Aufgrund der geringen Differenzen der Abstandsmessungen kann auf eine zulässige Einstellung der Kameraparameter geschlossen werden, da diese für eine zuverlässige Abstandsbestimmung nötig sind. Durch die korrekten Kameraparameter kann eine zuverlässig 3D-Rekonstruktion durchgeführt werden und ermöglicht dadurch eine präzise Wahrnehmung der Umgebung [30].

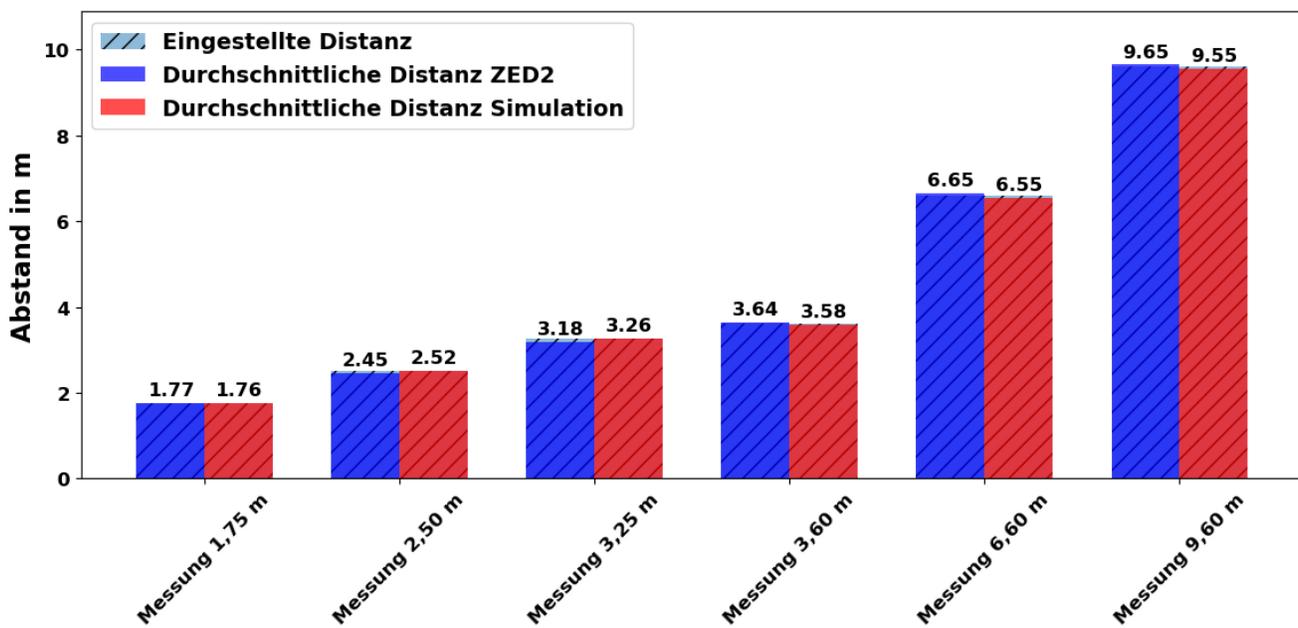


Abbildung 5.5: Abstandsmessung mit der Tiefenkamera

5.1.4 Beurteilung der Simulationsumgebung

Die Güte der Simulationsumgebung kann ebenfalls mithilfe der Übereinstimmung der 2D-Scans beurteilt werden. Mit diesem Verfahren werden die geometrischen Abmessungen untersucht und bewertet. Aus CloudCompare können die nötigen Informationen bezüglich Verteilung der Punkte ermittelt werden. In Abbildung 5.6 ist eine statistische Analyse der korrespondierten Punktwolken dargestellt. Dabei konnte für den 2D-Scan des 18. Rings eine absolute Messabweichung von 17,5 cm und eine relative Standardabweichung von 18,1 cm ermittelt werden. Es fällt auf, dass einige Punkte eine absolute Distanz zum jeweiligen korrespondierenden Punkt von mehr als einem Meter haben. Um den Einfluss der verglasten Flächen zu bestimmen, muss eine Betrachtung der Flächen ohne Verglasung durchgeführt werden.

5 Evaluation

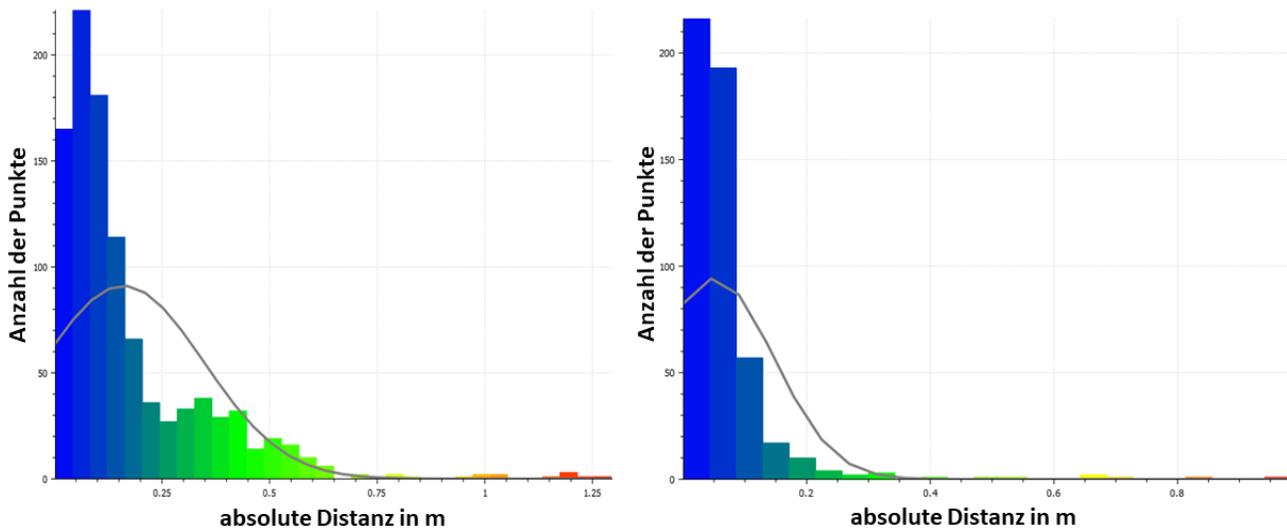


Abbildung 5.6: Statistische Verteilung mit CloudCompare des 18. Rings (links) und dem halb 18. Ring (rechts)

Hinsichtlich der absoluten Messabweichung von 7,0 cm und einer relativen Messabweichung von 9,1 cm fällt auf, dass durch eine Betrachtung der ausgewählten Fläche eine geringere Messabweichung ermittelt werden kann. Daraus lässt sich die Schlussfolgerung ableiten, dass eine Abbildung der Materialeigenschaften nicht erfolgreich war. So ist es zwar möglich dem Material Eigenschaften wie beispielsweise einen Emissionsgrad, Spiegelglanz oder Diffus zu geben, allerdings üben diese keinen Einfluss auf die Umgebung aus. Trotz der fehlenden Materialeigenschaften konnte eine Gesamtabweichung von 34,3 cm bei einer dreifachen Standardabweichung ermittelt werden. Demnach wurde eine erfolgreiche Erstellung der Simulationsumgebung hinsichtlich der Abmaße erzielt.

5.2 Limitation

Eine vermutlich ausschlaggebende Grenze des Modells ist die Nachbildung der realen Umgebung mit ihren materiellen Eigenschaften, wie beispielsweise Reflexion und Refraktion. Dabei weisen Glasfassaden einen deutlich geringeren Brechungsindex im Vergleich zu einer Putzfassade oder Natursteinfassade auf. Sensoren, die auf diese Umwelteinflüsse empfindlich reagieren, wie beispielsweise der Lidar, können dadurch erheblich größere Messabweichungen zwischen der Simulation und dem realen Sensor aufweisen.

Durch die Anpassung der Simulationsphysik wurde erreicht, dass das Schienenfahrzeug auf den Gleisen in einem stabilen Zustand bleibt. Allerdings hat dies zu einer deutlichen Verschlechterung der dynamischen Fahreigenschaften geführt. Auf der anderen Seite würde

5 Evaluation

eine Anpassung der dynamischen Fahreigenschaften zu einem unruhigen Verhalten im Stillstand führen, was wiederum die Messabweichungen aller Sensoren vergrößern würde. Ein Kompromiss zwischen dem statischen und dynamischen Verhalten des Simulationsmodells ist nicht realisierbar. Dies ist darauf zurückzuführen, dass der Kontakt zwischen den Gleisen und dem Schienenfahrzeug als Kollision der beiden Modelle erfasst wird und die Open Dynamics Engine kontinuierliche Korrekturen an den Objekten berechnet.

Ein weiterer Aspekt, der eine Messabweichung zwischen Simulationsmodell und realem Schienenfahrzeug darstellt, sind die Sensoraufnahmen. Durch eine sich ändernde Pose bei der Montage der Sensorik kann es, trotz Kalibrierung, zu geringfügigen Abweichungen kommen. Folglich muss nach jeder Inbetriebnahme der Anlage entweder eine Ausrichtung der Sensorik erfolgen oder eine erneute Kalibrierung der Simulationssensorik.

5.3 Ausblick für die Forschung

In Bezug auf die Limitation wäre eine Nachbildung der Gleise als Schienenfahrzeugkomponente denkbar. Hierfür könnte ein zusammenhängendes Streckenmodell als Joint des Schienenfahrzeuges erstellt werden. Eine Lösung des statischen und dynamischen Verhaltens durch die fehlende kontinuierliche Korrektur der Simulationsphysik wäre möglich. Somit könnten dynamische Tests des Simulationsmodells durchgeführt werden, um potenzielle Kalibrierungsanforderungen aufzudecken und zu adressieren. Mithilfe dynamischer Untersuchungen könnten weitere Parameter der Simulationsumgebung, wie beispielsweise Reibung zwischen den Gleisen und Schienenfahrzeug oder Beschleunigung und Bremsverhalten genauer ermittelt werden. Dies würde zu einer noch realistischeren Modellierung führen.

Eine weitere Möglichkeit, das Simulationsmodell und dessen Sensorik zu verbessern, besteht darin, eine Variation der Simulationsumgebungen und weitere Simulationsumgebungen zu erstellen. Eine Untersuchung unterschiedlicher Umgebungen mit variablen Lichtverhältnissen und Hindernissen könnten auch hier Kalibrierungsanforderungen aufdecken. Zusätzlich könnte eine fotorealistische Abbildung der Umgebung genauere Messdaten liefern. Eine zusätzliche Möglichkeit wäre die Analyse der simulierten Materialeigenschaften, insbesondere in Hinblick auf die Darstellung von Lichtreflexion, Brechung und Streuung. Dieser Ansatz könnte dazu beitragen, die sensorische Wahrnehmung in der Simulationsumgebung insgesamt zu verbessern.

6 Fazit

In dieser Bachelorarbeit wurde die Erstellung einer digitalen Nachbildung in der Simulationsumgebung mit Gazebo durchgeführt. Im Fokus standen die korrekte Abbildung des Testgeländes des Schienenfahrzeuges und dessen Sensorik nachzubilden. Hierfür wurden die Parameter der Simulationssensorik so an das reale Schienenfahrzeug angepasst, dass die Messdaten äquivalente Ergebnisse liefern.

Die in der Zielsetzung definierten Systemanforderungen konnten erfolgreich durch die in der Methodik beschriebenen Verfahren umgesetzt werden. Daraus folgend können der Tabelle 6.1 die gestellten Anforderungen und deren Umsetzung entnommen werden.

Tabelle 6.1: Systemanforderungen und Umsetzung

	Vorgabewert	Simulationswert	Erfüllt?
Absolute Messabweichung des IMU-Sensors			
Beschleunigung in m/s^2	0,031	$7,58 \times 10^{-04}$	✓
Winkelgeschwindigkeit in $^\circ/s$	0,16	$5,15 \times 10^{-04}$	✓
Relative Messabweichung des IMU-Sensors			
Beschleunigung in m/s^2	0,031	$3,53 \times 10^{-04}$	✓
Winkelgeschwindigkeit in $^\circ/s$	0,16	$6,00 \times 10^{-06}$	✓
Messabweichung des Lidars			
Relative Messabweichung in mm	30	1	✓
Übereinstimmung der 2D-Scans in cm	± 55	26	✓
Stereokamera Abstandsmessung			
Messabweichung	> 1% bis 3 m > 5% bis 15 m	> 1% bis 3 m > 2,5% bis 15m	✓
Simulation			
Messabweichung in cm	± 55	26	✓
physikalische Eigenschaften	z. B. Materialeigenschaften	—	✗

Trotz fehlenden physikalischen Materialeigenschaften kann mithilfe von Gazebo, in Kombination mit ROS2, die Simulationsumgebung mit ihrer Sensorik präzise abgebildet werden. Die geometrischen Strukturen des nachgebildeten Testgeländes konnten dabei erfolgreich an das reelle Testgelände angepasst werden. Durch die Verwendung von ROS konnten Integrationstests und Softwareanwendungen sowohl in der Simulation als auch auf dem realen Schienenfahrzeug ausgeführt werden. Eine künftig schnellere Entwicklungszeit und einfachere Integration neuer Hardwarekomponenten und Softwareanwendungen ist damit möglich.

Quellenverzeichnis

- [1] BLÄTZINGER, Markus: Simulation in der Chirurgie. (2023)
- [2] *BIM Simulation Software im Bauwesen*. <https://biblus.accasoftware.com/de/bim-simulation-software-im-bauwesen/>. – abgerufen am: 27.08.2023
- [3] RAATSCHEN, Hans J.: Finite Elemente Methode Eine problemorientierte Einführung. (2020), S. 0–2
- [4] NAKATH, David ; SHE, Mengkun ; SONG, Yifan ; KÖSER, Kevin: An Optical Digital Twin for Underwater Photogrammetry. (2022)
- [5] YILDIZ, Ahmet ; ARIKAN, Oktay ; KESKIN, Kemal: Validation of The Single Train Simulation Model With a Real-World Case Study. In: *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)* IEEE, 2022, S. 1–6
- [6] TDK INVENSENSE (Hrsg.): *IAM-20680HT High Performance Automotive 6-Axis MotionTracking Device*. : TDK InvenSense, 2021
- [7] OUSTER (Hrsg.): *Mid-Range High-Resolution Imaging Lidar*. : Ouster, 2023
- [8] STEREO LABS (Hrsg.): *ZED2 Camera and SDK Overview*. : Stereolabs, 2018
- [9] *Driverless rail vehicles: Autonomous travel across the country and around the city*. <https://safe-intelligence.fraunhofer.de/en/articles/driverless-rail-vehicles>. – abgerufen am: 06.09.2023
- [10] *Positive Train Control (PTC)*. <https://www.mobility.siemens.com/global/de/portfolio/schiene/bahnautomatisierung/zugbeeinflussung/positive-train-control.html>. – abgerufen am: 06.09.2023
- [11] GAO, Yan ; QIAN, Shuyue ; LI, Zihan ; WANG, Ping ; WANG, Feiyue ; HE, Qing: Digital twin and its application in transportation infrastructure. In: *2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI)* IEEE, 2021, S. 298–301
- [12] ARIYACHANDRA, MRMF ; BRILAKIS, Ioannis: Digital twinning of railway overhead line equipment from airborne lidar data. In: *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction* Bd. 37 IAARC Publications, 2020, S. 1270–1277
- [13] Bi, Zhuming ; DA XU, Li ; OUYANG, Puren: *Smart Manufacturing*. MDPI-Multidisciplinary Digital Publishing Institute, 2022

Quellenverzeichnis

- [14] SENGUPTA, Monish: Choice of sensor fusion framework for train positioning system. In: *WIT Transactions on The Built Environment. Computers in Railways XVII* 199 (2020), S. 53–64
- [15] *XML Robot Description Format (URDF)*. <http://wiki.ros.org/urdf/XML/model>. – abgerufen am: 27.08.2023
- [16] *SDFFormat*. <http://sdformat.org/spec>. – abgerufen am: 27.08.2023
- [17] *Gazebo*. <https://gazebo.org/about>. – abgerufen am: 27.08.2023
- [18] REUTER, Manfred ; ZACHE, Serge: *Regelungstechnik für Ingenieure*. Wiesbaden : Vieweg+Teubner, 2008
- [19] *Stereolabs*. <https://www.stereolabs.com/docs/get-started-with-zed/>. – abgerufen am: 04.09.2023
- [20] OUSTER (Hrsg.): *ROS2 Ouster Drivers*. : Ouster, 2023
- [21] RICO, Francisco M.: *A Concise Introduction to Robot Programming with ROS2*. CRC Press, 2023
- [22] SCHRÜFER, Elmar: *Elektrische Messtechnik*. Muenchen : Wichmann, 2007
- [23] SHANOER, Mohsin M. ; ABED, Fanar M.: Evaluate 3D laser point clouds registration for cultural heritage documentation. In: *The Egyptian Journal of Remote Sensing and Space Science* 21 (2018), Nr. 3, 295-304. <http://dx.doi.org/https://doi.org/10.1016/j.ejrs.2017.11.007>. – DOI <https://doi.org/10.1016/j.ejrs.2017.11.007>. – ISSN 1110–9823
- [24] RUSINKIEWICZ, Szymon ; LEVOY, Marc: Efficient Variants of the ICP Algorithm. In: *Stanford University* (2001)
- [25] *Open3D ICP Registration*. http://www.open3d.org/docs/latest/tutorial/Basic/icp_registration.html. – abgerufen am: 15.09.2023
- [26] *CloudCompare ICP*. <https://www.cloudcompare.org/doc/wiki/index.php/ICP>. – abgerufen am: 15.09.2023
- [27] *How to Calibrate a Stereo Camera*. http://wiki.ros.org/camera_calibration/Tutorials/StereoCalibration. – abgerufen am: 16.09.2023
- [28] CONDOTTA, Isabella C. ; BROWN-BRANDL, Tami M. ; PITLA, Santosh K. ; STINN, John P. ; SILVA-MIRANDA, Késia O.: Evaluation of low-cost depth cameras for agricultural applications. In: *Computers and Electronics in Agriculture* 173 (2020), 105394. <http://>

Quellenverzeichnis

<https://dx.doi.org/https://doi.org/10.1016/j.compag.2020.105394>. – DOI
<https://doi.org/10.1016/j.compag.2020.105394>. – ISSN 0168–1699

[29] *Can the ZED stereocameras detect glass doors or glass walls.*

[https://support.stereolabs.com/hc/en-us/articles/
4402086483735-Can-the-ZED-stereocameras-detect-glass-doors-or-glass-wal](https://support.stereolabs.com/hc/en-us/articles/4402086483735-Can-the-ZED-stereocameras-detect-glass-doors-or-glass-wal)

– abgerufen am: 23.09.2023

[30] BESCHI, Riccardo ; FENG, Xiao ; MELILLO, Stefania ; PARISI, Leonardo ; POSTIGLIONE, Lorena: Stereo camera system calibration: the need of two sets of parameters. In: *CoRR* abs/2101.05725 (2021). <https://arxiv.org/abs/2101.05725>

Abkürzungsverzeichnis

fps	Frames per second
GNSS	Global Navigation Satellite System
GUI	Graphical User Interface
ICP	Iterative-Closest-Point
IMU	Inertial Measurement Unit
KMAC	Kompetenzzentrum Mobilität FH Aachen
Lidar	Light-Detection-and-Ranging
ROS	Robot Operating System
RMSE	Root Mean Squared Error
SDF	Simulation Description Format
SDK	Software Development Kit
URDF	Unified Robot Description Format
XML	Extensible Markup Language

Abbildungsverzeichnis

1.1	Simulation einer statischen Festigkeitsanalyse mit 3D-Solids [3]	1
1.2	Positive Train Control [10]	3
2.1	Verbindung einer Simulationsstruktur nach [15]	5
2.2	Darstellung von Visual (grau), Collision (orange) und Inertial (pink)	6
2.3	Darstellung der Transformation des Simulationsmodells	7
3.1	Schienenfahrzeug des Forschungsprojektes	9
3.2	Auswertungsverfahren der Sensordaten	9
3.3	Messplatz am KMAC	10
3.4	Stereolabs ZED2 und Ouster OS1 [7], [19]	11
3.5	Sensorbilder des Ouster OS1 der Entfernungswerte, Intensitätswerte, Rauschwerte und Reflektivitätswerte am Messstandort	12
3.6	Simulationsmodell Gazebo-Train	14
3.7	Erstellung einer Simulationsumgebung mit dem Gazebo-Buildeditor	15
4.1	Vorgehensweise für die Sensoreinstellung des Modell	16
4.2	Vergleich der IMU-Daten der ZED2 und ZED2 mit Ouster	17
4.3	Lineare Beschleunigung in Z-Richtung nach erster Messung ohne Gleise	19
4.4	Lineare Beschleunigung in Z-Richtung nach erster Messung auf den Gleisen	20
4.5	Linearen Beschleunigungen nach erster Messung auf den Gleisen	20
4.6	Abweichung der linearen Beschleunigungen nach erster Messung	21
4.7	Standardabweichungen der Lidar-Daten der Simulation und des Ousters	22
4.8	Punktwolken des Messplatzes mit dem Scan des 0-Rings (oben) und des 16-Rings (unten)	23
4.9	ICP-Registrierungsalgorithmus des ganzen Scans (oben) und halben Scans (unten)	24
4.10	Kamerakalibrierung mit ROS	25
4.11	Aufbau des Messplatzes Schienenfahrzeuglabor (links) und Simulation (rechts)	26
4.12	Punktwolke der Distanzerkennung am Messplatzes	26
5.1	Absolute Messabweichung der IMU-Daten	27
5.2	Relative Messabweichung der IMU-Daten	28
5.3	Standardabweichungen der Lidar-Daten der Simulation und des Ousters	30
5.4	RSME der verwendeten Punkte der Punktwolken	32
5.5	Abstandsmessung mit der Tiefenkamera	33
5.6	Statistische Verteilung mit CloudCompare des 18. Rings (links) und dem halb 18. Ring (rechts)	34

Tabellenverzeichnis

1.1 Systemanforderungen [6], [7], [8]	2
3.1 IMU-Daten der Stereolabs ZED2 und des Ouster OS1 [6], [8], [19]	13
4.1 Zusammenfassung der ZED-Daten	18
5.1 Differenzen der IMU-Daten zwischen der ZED2 und der Simulation	29
5.2 Vergleich der 2D-Scans des Ouster OS1 und dem Simulationsmodell	31
6.1 Systemanforderungen und Umsetzung	36

7 Anhang

Git-Repository

1. Installations- und Bedienungsanleitung als README.md
2. ROS2 Schienenfahrzeug-Simulation
3. ROS2-Nodes zum Auslesen der Sensorik
4. Python-Quellcode zur Auswertung der Sensorik
5. Python-Quellcode zur Erstellung der Diagramme