

Aufbau eines virtuellen Umweltmodelles für die Entwicklungen von Algorithmen für die sichere Personenerkennung in Automated Guided Vehicle Applikationen

PROJEKTARBEIT

für die Prüfung zum

Bachelor of Science

des Studienganges Informationstechnik

an der

Dualen Hochschule Baden-Württemberg Karlsruhe

von

Lorenz Scherrer

Abgabedatum 24. September 2023

Bearbeitungszeitraum

12 Wochen

Matrikelnummer

8809469

Kurs

tinf21b3

Ausbildungsfirma

SICK AG

Waldkirch

Betreuer der Ausbildungsfirma

Manfred Haberer

Gutachter der Studienakademie

Prof. Dr. Jürgen Vollmer

Erklärung

Ich versichere hiermit, dass ich meine Projektarbeit mit dem Thema: »Aufbau eines virtuellen Umweltmodelles für die Entwicklungen von Algorithmen für die sichere Personenerkennung in Automated Guided Vehicle Applikationen« selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt. _____

Ort Datum

Unterschrift

Sofern vom Dualen Partner ein Sperrvermerk gewünscht wird, ist folgende Formulierung zu verwenden:

Sperrvermerk

Der Inhalt dieser Arbeit darf weder als Ganzes noch in Auszügen Personen außerhalb des Prüfungsprozesses und des Evaluationsverfahrens zugänglich gemacht werden, sofern keine anderslautende Genehmigung vom Dualen Partner vorliegt.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Projektumfeld und Kontext	1
1.2	Problemstellung	1
1.3	Zielsetzung	1
1.4	Fragestellung	1
2	Planung und Kontrolle	2
2.1	Zeitmanagament	2
2.1.1	Meilensteinen und Etappen	2
2.1.2	Zeitplan	3
2.2	Risikoanalyse	6
3	Grundlagen	7
3.1	gRPC	7
3.2	Thales	7
3.3	Senoren in der VirtuellenUmgebung	7
4	Methodik	8
4.1	Erstellung der Sensoern in der Virtuellen Umgebung	8
4.2	AGVs in der Virtuellen Umgebung	8
5	Implementierung	9
5.1	Implementierung der Schnittstelle	9
5.2	AGVs in der virtuellen Umgebung	9
6	Ergebnis	10
6.1	Diskussion	10
7	Notizen	11
7.0.1	Aufgaben	11
7.0.2	Zeitplan	11
7.0.3	Notizen 03.07	13
7.0.4	12.07	13
7.0.5	13.07	13
7.0.6	14.07	13
7.0.7	17.07	13
7.0.8	18.07	14

INHALTSVERZEICHNIS

II

7.0.9	19.07	14
7.0.10	20.07	15
7.0.11	21.08	15
7.0.12	22.08	16

Anhang	VIII
---------------	-------------

Index	VIII
--------------	-------------

Literaturverzeichnis	VIII
-----------------------------	-------------

Liste der ToDo's	IX
-------------------------	-----------

Abbildungsverzeichnis

Tabellenverzeichnis

Liste der Algorithmen

Formelverzeichnis

Abkürzungsverzeichnis

Kapitel 1

Einleitung

1.1 Projektumfeld und Kontext

GICTICISM Global Industry Centers Technical Industry Competence & Innovation

1.2 Problemstellung

DSM Project was ist dieses Projekt? Wo finde ich informationen zum DSM Projekt? Boris morgen fragen Was trage ich zu diesem Projekt bei?

Es gib keine Simulation in der man Gefahrensituationen testen kann. Ohne Sie im echten Hallen zu testen.

Im weiteren verlauf kann der Algorithmus der die Sicherheitsfelder der AGVs berechnet an ein Institut geschickt werden der diesen Validiert. Oder ein in der Simulation getesteten Algorithmus in echten Szenarien getestet werden. Hier können die Sicherheitsalgorithmen erstmals getestet werden.

1.3 Zielsetzung

Daten übermitteln und ein AGV in Unity erstellen.

1.4 Fragestellung

Wie kann ein AGV in Unity erstellt werden und dessen Mechanische Daten über gRPC / ThalesApi mit der Außenwelt geteilt werden und auch Daten erhalten?

Kapitel 2

Planung und Kontrolle

2.1 Zeitmanagement

2.1.1 Meilensteinen und Etappen

Innerhalb ersten Meilensteins liegt der Fokus auf der Umsetzung einer funktionsfähigen Schnittstelle zwischen dem gRPC Server und Client, um Daten in beiden Richtungen zu übertragen. Ein zentrales Ziel ist es, Thales Sensordaten von insgesamt 4 Sensoren über den gRPC Server vom Client abrufen zu können. Die Datenabfrage sollte in der Lage sein, mit einer Aufnahmefrequenz von mindestens alle 40 Millisekunden durchgeführt zu werden.

Zusätzlich zu den Sensordaten sollen auch andere Parameter über den gRPC Server übertragen werden, wie beispielsweise der Lenkwinkel oder die Geschwindigkeit eines zukünftig erstellten AGVs. Dieser Meilenstein bildet einen wesentlichen Schritt, um die grundlegende Funktionalität des Projektes zu etablieren und legt den Grundstein für die weitere Entwicklung im Projektverlauf.

Der Meilenstein umfasst die Erstellung der Fahrphysik für die AGVs. Dies beinhaltet die Realisierung von realistischen Bewegungsabläufen und Steuerungsmechanismen, die den Charakter der jeweiligen AGV-Typen authentisch wiedergeben.

Innerhalb dieses Meilensteins ist die Entwicklung mehrerer Arten von AGVs geplant, wobei mindestens eine Variante umgesetzt wird. In der Anfangsphase werden verschiedene AGVs entworfen, darunter ein AGV, das auf einem herkömmlichen Auto basiert, gefolgt von einem dreirädrigen Gabelstapler und schließlich einer Panzerrolle.

Die Steuerung und Fähranweisung der AGVs wird über ein Script in der Unity-Umgebung gesteuert. Dieses Script gibt die Geschwindigkeit, Beschleunigung und den Lenkeinschlag der Steuerachse vor.

Die erfolgreiche Umsetzung dieses Meilensteins wird die Basis für die funktionale und realistische Integration der AGVs in das Forschungsprojekt legen. Durch die Entwicklungsarbeit im zweiten Meilenstein wird die Grundlage für weiterführende Tests und Analysen der AGVs geschaffen, die zur Erreichung der gesamten Projekts beitragen.

Innerhalb dieses Meilensteins ist das Hauptziel die Erstellung eines Polygonzugs, der den Bremsweg des AGVs abbildet. Dieser Polygonzug soll den Verlauf des Bremsvorgangs in einer geometrischen Darstellung wiedergeben.

Nach der Erstellung des Polygonzugs muss überprüft werden, ob dieser direkt auf das AGV-Modell übertragen werden kann. Es ist von Bedeutung zu ermitteln, ob die Daten des Polygonzugs in das Simulationssystem eingeführt werden können, um eine korrekte Darstellung des Bremsverhaltens zu ermöglichen.

Falls eine direkte Übertragung des Polygonzugs nicht möglich ist, wird eine alternative Methode in Betracht gezogen. Hierbei wird die Länge des Bremsschlauchs in Verbindung mit dem Lenkeinschlag genutzt, um den Bremsweg auf eine plausible Weise darzustellen.

Die erfolgreiche Erfassung des AGV-Bremswegs als Polygonzug oder über alternative Darstellungsformen wird eine wichtige Grundlage für die Simulation des Bremsverhaltens in späteren Forschungsphasen bilden. Dieser Meilenstein trägt dazu bei, das Verhalten des AGVs unter verschiedenen Bedingungen besser zu verstehen und die Gesamtfunktionalität des Modells zu verbessern.

Im Rahmen dieses Meilensteins wird das Hauptziel verfolgt, die Signalübertragung vom gRPC-Client zum AGV zu implementieren. Das Signal soll die Abbremsung des AGVs auslösen, wenn es empfangen wird.

Es ist von entscheidender Bedeutung, die Steuerungsfunktion im AGV zu konfigurieren, damit sie auf das empfangene Signal reagiert und die notwendigen Maßnahmen zur Abbremsung einleitet. Die erfolgreiche Umsetzung dieses Meilensteins wird die Fähigkeit des AGVs zur externen Steuerung und zur Initiierung einer Abbremsung über das gRPC-Client-Signal demonstrieren. Diese Funktionalität kann im weiteren Verlauf der Forschungsarbeit dazu verwendet werden, verschiedene Szenarien und Anwendungen zu testen, bei denen eine externe Steuerung des AGVs erforderlich ist.

2.1.2 Zeitplan

Das Projekt startet am 3. Juli und die Abgabe der schriftlichen Projektarbeit ist am 21. September. Nach der Berücksichtigung der Abwesenheiten umfasst das Projekt 35 Arbeitstage.

In den ersten fünf Tagen des Projekts steht die Vorbereitung und Planung im Vordergrund. Hierbei werden die genauen Anforderungen und Ziele des Projekts geklärt. Es erfolgt eine detaillierte Aufschlüsselung der Arbeitsschritte sowie die Definition der benötigten Ressourcen. Vom sechsten bis zum fünfzehnten Tag wird der erste Meilenstein erreicht - die Implementierung der gRPC-Server-Client-Schnittstelle. Ziel ist es, die bidirektionale Datenübertragung zwischen dem Server und Client zu ermöglichen.

Vom sechzehnten bis zum einundzwanzigsten Tag liegt der Fokus auf dem zweiten Meilenstein, der die Umsetzung der AGVs beinhaltet. Es wird die Fahrphysik für verschiedene AGV-Typen erstellt, beginnend mit einem AGV, das sich wie ein herkömmliches Auto verhält. Die Bewegungsabläufe und Steuerung werden getestet und optimiert.

Vom zweiundzwanzigsten bis zum sechsundzwanzigsten Tag wird die Umsetzung der AGVs fortgesetzt. Ein Gabelstapler-AGV wird entwickelt, um dreirädrige Bewegungen zu simulieren. Parallel dazu wird die Vorbereitung für die Umsetzung des dritten AGV-Typs, einer Panzerrolle, getroffen.

Vom siebenundzwanzigsten bis zum neunundzwanzigsten Tag wird der dritte Meilenstein erreicht. Ein Polygonzug wird erstellt, um den Bremsweg des AGVs darzustellen. Die Übertragbarkeit dieser Daten auf das Modell wird geprüft, alternativ wird die Berechnung des Bremswegs anhand des Bremsschlauchs in Betracht gezogen.

Die letzten beiden Tage, der dreißigste und der zweidreißigste Tag, widmen sich dem vierten Meilenstein. Hierbei steht die Implementierung der externen Abbremsung des AGVs durch ein Signal vom gRPC-Client im Fokus. Die Steuerungsfunktion im AGV wird entsprechend konfiguriert und getestet, um sicherzustellen, dass das AGV zuverlässig auf das Signal reagiert.

Aufgabe	Anfang	Ende	28	29	30	31	32	33	34	35	36	37	38	39
Planung+ Aufgabe Prio 1	12.07	14.07												
Aufgabe Prio 2	17.07	21.07												
Abwesenheit	24.07	01.09												
Aufgabe Prio 2	04.09	08.09												
Aufgabe Prio 3	11.09	15.09												
Aufgabe Prio 4+Doku	18.09	29.09												
Praxisbericht + Kollogium Vorbereitung	18.09	22.09												
Kollogium	25.09	29.09												

2.2 Risikoanalyse

Im Rahmen des Projekts wurden potenzielle Risiken identifiziert, die während der Umsetzung der Meilensteine und des 35-tägigen Zeitplans auftreten könnten. Diese Risiken wurden analysiert und bewertet, um angemessene Maßnahmen zur Risikobewältigung zu identifizieren.

Ein hoch bewertetes Risiko besteht in technischen Herausforderungen bei der Implementierung der gRPC-Server-Client-Schnittstelle (Meilenstein 1). Die Eintrittswahrscheinlichkeit wird als mittel eingestuft, aber die Auswirkungen könnten hoch sein, da dies die Grundlage für die Kommunikation im Projekt bildet. Ähnlich verhält es sich mit dem Risiko unvorhergesehener Schwierigkeiten bei der Signalübertragung für die externe Abbremsung (Meilenstein 4). Hierbei könnte die Eintrittswahrscheinlichkeit mittel sein, aber die Auswirkungen wären hoch, da die externe Steuerung des AGVs beeinträchtigt werden könnte.

Die Umsetzung der AGV-Typen (Meilenstein 2) birgt das Risiko von Schwierigkeiten bei der Erstellung der Fahrphysik, da die Eintrittswahrscheinlichkeit als hoch eingeschätzt wird und die Auswirkungen mittel sein könnten. Zudem besteht die Gefahr von zeitlichen Verzögerungen bei der Umsetzung der AGV-Typen, da eine mittlere Eintrittswahrscheinlichkeit und mittlere Auswirkungen gegeben sind.

In Bezug auf den dritten Meilenstein, die Erfassung des Bremsweg-Polygonzugs, besteht das Risiko technischer Inkompatibilität bei der Übertragung, obwohl die Eintrittswahrscheinlichkeit niedrig ist, könnten die Auswirkungen mittel sein. Zusätzlich könnte das Risiko bestehen, dass sich die Anforderungen an den Bremsweg ändern, was jedoch als gering eingeschätzt wird.

Die mögliche Fehlkommunikation zwischen dem gRPC-Server und dem Client (Meilenstein 1) sowie unerwartete Schwierigkeiten bei der Konfiguration der Steuerungsfunktion für die externe Abbremsung (Meilenstein 4) könnten sich als hohe Risiken mit mittleren Auswirkungen erweisen.

Ein weiteres bedeutendes Risiko betrifft die mangelnde Verfügbarkeit von Ressourcen wie Hardware oder Software während des gesamten Projekts. Hierbei wird eine hohe Eintrittswahrscheinlichkeit angenommen, verbunden mit hohen Auswirkungen auf den Projektfortschritt.

Die Risikoanalyse zeigt auf, dass eine gründliche Planung und regelmäßige Überwachung der Projektschritte unerlässlich ist, um auf mögliche Risiken rechtzeitig reagieren zu können. Es empfiehlt sich, für jedes identifizierte Risiko geeignete Maßnahmen zur Risikominderung und -bewältigung zu entwickeln.

Kapitel 3

Grundlagen

3.1 gRPC

3.2 Thales

3.3 Sensoren in der Virtuellen Umgebung

Kapitel 4

Methodik

4.1 Erstellung der Sensoren in der Virtuellen Umgebung

4.2 AGVs in der Virtuellen Umgebung

Fahr physik in Unity Wie wird der Wendekreis berechnet

Kapitel 5

Implementierung

5.1 Implementierung der Schnittstelle

5.2 AGVs in der virtuellen Umgebung

Kapitel 6

Ergebnis

6.1 Diskussion

Kapitel 7

Notizen

7.0.1 Aufgaben

Die Aufgabe ist es sich in Unity ein zuarbeiten. Dort eine Scenerie aufzubauen und die Daten die die vier Sensoren liefern mit den Bewegungs vektoren übertragen. Die Übertragung soll über einen gRPC Client und einen gRPC Server laufen. Dannach sollen mit den erzeugten Daten das Feld des DSM Host in Unity angezeigt werden(Best Case für Demo gedacht).

- Zeitplan
- Riskomanagment
-

7.0.2 Zeitplan

- 13. Juli - 21. Juli = 7 Tage
- 4. September - 30. September = 20 Tage
- 27 Arbeitstage a 7 Stunden gleich 189 Stunden

Aufgabe	Anfang	Ende	28	29	30	31	32	33	34	35	36	37	38	39
Planung+ Aufgabe Prio 1	12.07	14.07												
Aufgabe Prio 2	17.07	21.07												
Abwesenheit	24.07	01.09												
Aufgabe Prio 2	04.09	08.09												
Aufgabe Prio 3	11.09	15.09												
Aufgabe Prio 4+Doku	18.09	29.09												
Praxisbericht + Kollogium Vorbereitung	18.09	22.09												
Kollogium	25.09	29.09												

Prios der arbeit Prio 1 Senordaten über gRPC übertragen Prio 2 Wenderadius und Geschwindigkeit übertragen mit der Thales (Einauto selbst Desginen) es gibt mehrer auto: gabelstapler, Panzerrolle, normale Auto Prio 3 Polghonzug erhalten und in Unity darstellen Prio 4 NotSignal erhalten und das AGV anhalten

7.0.3 Notizen 03.07

Unity zum laufen bekommen auf dem anderen PC.

7.0.4 12.07

Plan für heute dem AGV Geschwindigkeit geben. Und weiter geben. Das Problem die AGV/Service wird nicht erkannt. Obwohl der SplineWalker mit dem dem Agv Service Script verbunden ist. Die

7.0.5 13.07

- Projekt Einsicht in VS Studio
- Martin schreiben wegen dem AGV übertragung
- Fragen formulieren zur Thales Schnittstelle
-

Fragen zur Thales Schnittstelle

- Jan hat eine Agvservice script erstellt welches IAgv verwendet aber es wird nicht bei den Avalibele Services angezeigt
- Hallo Martin wir haben noch ein paar Fragen zur Thales Api. Hast du heute Nachmittag Zeit für ein Meeting mit Artem und mir?

7.0.6 14.07

- Sprint Meeting vorbereiten
- die vier Scanner abfragen und übertragen. Wie funktioiert das mit dem Stepper?
- Welche Scanner sollen übertragen werden?
- Wie werden AGVs in Unity umgesetzt?
- Aufgaben in Jira

7.0.7 17.07

- Doku erledigt
 - Nächste Aufgabe erstellen es AGV
- Nächste Aufgabe erstellen es AGV
- Scene aufbauen
 - Object für des AGV erstellen

7.0.8 18.07

Aufgaben für heute

- Zeitplan überarbeiten welche Aufgaben wurden genau bearbeitet orientieren an den Jira Tickets
- Risikoanalyse stellen
- wie könnte ein Polygonzug übergeben werden? es können nur Parameter geschickt werden die in den Protobuff passen
- Inhaltsverzeichnis für Bericht erstellen
- Git zum laufen bringen
- Arbeitsparkte für AGVs erstellen

Projektplan

Anforderungen von Manfred

- AutoStepper verstehen muss alle 10 ms senden
- Abstand zwischen Rad und lenkachse muss freibestimmbar sein
- Es muss ein Dreirädriges Fahrzeug sein
- es muss mit angaben eins Scripts laufen
- Aufgaben pakete müssen in Jira stehen

7.0.9 19.07

- gRPC Schnittstelle bearbeiten
- Inhaltsverzeichnis erstellen

Inhaltsverzeichnis

- Einleitung
 - Projektumgebung
- Aufgabenstellung
 - Interesse der Firma
 - Generelle Aufgabe
- Grundlagen
 - gRPC
 - Thales
 - Senoren in der Viruellen Umgebung

- Konzept
 - Design der Sensoren in der Virtuellen Umgebungen
 - AGVs in der Virtuellen Umgebung
- Implementierung
 - Implementierung der Schnittstelle
 - Echtzeitübertragung der Sensoren
- Ergebnis
- Ausblick

Fragen an Manfred

- Welche Scanner genau
-

7.0.10 20.07

Aufgaben für heute

- Sensoren Design -> fertig
- GitPush -> fertig
- Dokumentieren was gemacht wurde -> fertig
- namen für Directory heraus suchen -> fertig
- Urlaub buchen -> fertig
- Abwesenheit Email eintragen -> fertig

7.0.11 21.08

- Einen richtigen Zeitplan erstellen am besten mit Jira
- Verstehen was gemacht wurde
- was sind nochmal meine Aufgaben
- Zeitplan für den t2000 erstellen
- Kollogium vorbereiten
- Aktuelle wichtige aufgabe Version umstellen
- darstellung für den Zeitplan finden
- Struktur für den t2000 festlegen
- Wendekreis berechnen

7.0.12 22.08

- <https://www.youtube.com/watchv=xOLUiE0dxP0&list=PL6sL8yhCAsqVEWS5lzeWu3yA1LIb9rxx5&index=1>
- Fahrzeug erstellen
-

Änderungen

- 2020/03/13** Tippfehler korrigiert
aktuelle Formulierungen aus der Prüfungsordnung Technik übernommen
Formatdatei erklärt
- 2017/10/06** Anpassung an neuer Versionen diverse Pakete.
- 2016/03/16** Auf UTF-8 umgestellt, Indices.
- 2010/04/12** ToDo-Markierungen mit dem `\todo`-Kommando.
- 2010/01/27** Anhang (`appendix`), Selbständigkeits-Erklärung, `framed`-Paket.
- 2010/01/21** Abkürzungen (`acronym`), `table` und `tabular` benutzt, unübliche Pakete beigelegt.
- 2010/01/18** Code-Listings (`listings`), Literaturreferenzen `biblatex`)
- 2010/01/11** Initiale Version.

Liste der ToDo's