



Technische Universität Ilmenau

Fakultät für Informatik und Automatisierung

Fachgebiet Neuroinformatik und Kognitive Robotik

Benchmarking von Navigationsleistungen eines mobilen Roboters

Bachelorarbeit zur Erlangung des akademischen Grades Bachelor of Science

Thomas Golda

Betreuer: Dr. Christof Schröter

Verantwortlicher Hochschullehrer:

Prof. Dr. H.-M. Groß, FG Neuroinformatik und Kognitive Robotik

Die Bachelorarbeit wurde am 18.09.2014 bei der Fakultät für Informatik und Automatisierung der Technischen Universität Ilmenau eingereicht.

[urn:nbn:de:gbv:ilm1-2014200182](https://nbn-resolving.org/urn:nbn:de:gbv:ilm1-2014200182)

Erklärung: „Hiermit versichere ich, dass ich diese Bachelorarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe. Alle von mir aus anderen Veröffentlichungen übernommenen Passagen sind als solche gekennzeichnet.“

Ilmenau, 18.09.2014

.....
Thomas Golda

Inhaltsverzeichnis

1	Einleitung	1
1.1	Einschränkungen und Anpassungen	2
1.2	Struktur der Arbeit	3
2	Grundbegriffe und State of the Art	5
2.1	Grundbegriffe	5
2.1.1	Benchmark	5
2.1.2	Bewertungsmaß / Metrik	6
2.1.3	Navigation	6
2.2	State of the Art	7
2.2.1	Sicherheitsmetriken	8
2.2.2	Dimensionsmetriken	10
2.3	Zusammenfassung	12
3	Zielstellung	13
3.1	Fragestellungen	13
3.2	Voraussetzungen	14
3.2.1	Lokalisation	14
3.2.2	Navigation	15
3.3	Zusammenfassung	15
4	Implementierung	17
4.1	Aufbau	18
4.1.1	Das Benchmark-Tool	19

4.1.2	Die Evaluationskripte	22
4.2	Funktionsweise	25
4.3	Zusammenfassung	26
5	Experimentelle Untersuchungen	27
5.1	Das Wohnlabor	28
5.2	Hardwareaufbau	28
5.2.1	Die Lasertracker	29
5.2.2	Der Hutaufsatz	30
5.3	Ergebnisse	31
5.3.1	Abweichung der Roboterposen	32
5.3.2	Dauer der Pfadplanungsprozesse	34
5.3.3	Geschwindigkeitsanalysen	35
5.3.4	Evaluation der Zielanfahrten	37
5.3.5	Abstand zu Hindernissen	38
5.4	Probleme	39
5.4.1	Hutaufsatz	39
5.4.2	Lasertracker	40
5.4.3	Fazit	42
5.5	Zusammenfassung	43
6	Zusammenfassung und Ausblick	45
6.1	Zusammenfassung	45
6.2	Weiterführende Arbeiten	46
A	Ergänzende Unterlagen	47
A.1	Anleitung zur Nutzung der Software	47
B	Quellenangaben	51
	Literaturverzeichnis	54

Kapitel 1

Einleitung

Mobile Serviceroboter werden in zahlreichen Sektoren eingesetzt, in denen sie dem Menschen eine Dienstleistung erbringen sollen. Diese Dienstleistungen reichen von einfachen Aufgaben wie dem Staubsaugen einer Wohnung, dem Mähen eines Rasens oder der Reinigung eines Schwimmbads bis hin zu Aufgaben in der Gesundheitsassistenz, wie zum Beispiel das Durchführen und Überwachen von Bewegungsübungen bei älteren Menschen.

Zu Letzterem existiert das Projekt „Serroga“ am Fachgebiet Neuroinformatik und Kognitive Robotik, in welches sich diese Arbeit insbesondere einreicht.

Trotz der doch teilweise recht unterschiedlichen Einsatzgebiete haben all diese mobilen Serviceroboter insbesondere eines gemeinsam: Zur Durchführung ihrer Aufgaben müssen sie einerseits über eine Karte ihrer Umgebung verfügen, Informationen über ihre aktuelle Position aufweisen und die Fähigkeit besitzen, einen für die aktuelle Situation optimalen Weg zu einem bestimmten Ziel zu finden. Diese Aufgaben können durch spezielle Algorithmen umgesetzt werden. Eine Frage, die man sich stellt, sobald man eine Auswahl an Lösungsansätzen erhält: Für welche Variante sollte man sich entscheiden? Um eine Aussage über



Abbildung 1.1: Roboter „Tweety“
Einer der Roboter aus dem Projekt „Serroga“.

die Leistungsfähigkeit einzelner Lösungsansätze, d.h. einzelner Algorithmen, treffen zu können, benötigt man verschiedene Benchmarks und Bewertungsmaße. Diese erlauben es, Algorithmen hinsichtlich verschiedener Aspekte zu vergleichen und somit eine Aussage darüber zu treffen, welcher Algorithmus den eigenen Ansprüche am besten gerecht wird.

Das Wesen dieser Arbeit liegt in der Implementierung eines Tools, welches zur Durchführung solcher Benchmarks genutzt werden soll. Hierbei soll es den Nutzer dabei unterstützen, immer gleich ablaufende Mechanismen automatisiert durchzuführen und ihm somit zeitintensive Aufgaben abzunehmen. Benchmark-Experimente sollen weitestgehend autonom vonstattengehen und mit möglichst geringem Aufwand ausgewertet werden.

1.1 Einschränkungen und Anpassungen

In Absprache mit dem Betreuer wurde der Fokus dieser Arbeit auf die reine Implementierung des Tools gelegt. Die Recherche entsprechender Metriken und Bewertungsmaße wurde bereits in meinem vorangegangenen Hauptseminar [GOLDA, 2014] durchgeführt und bildet die theoretische Grundlage dieser Arbeit. Implementiert wurden hierbei eine Auswahl der recherchierten Metriken, die in Absprache mit dem Betreuer als nützlich und sinnvoll angesehen wurden und in ihrem Umfang der anzufertigenden Arbeit entsprachen. Weitere Informationen zu den Metriken finden sich in Kapitel 3, Kapitel 4 und Kapitel 5, in welchen sukzessive, angefangen bei den abstrakten Ideen der zu bewertenden Leistungen eines Roboters, über die effektive Umsetzung geeigneter Bewertungsmaße bis hin zur Vorstellung dieser anhand von realen Daten im Rahmen eines Testlaufs, die entsprechenden Bewertungsmaße vorgestellt und erläutert werden.

1.2 Struktur der Arbeit

Im Folgenden soll ein kurzer Überblick über den inhaltlichen Aufbau der Arbeit gegeben werden, um dem Leser ein Gefühl für den inhaltlichen Umfang zu geben.

Angefangen mit Kapitel 2, in welchem einige grundlegende Begriffe und Bewertungsmaße aus der Recherche für das vorangegangene Hauptseminar aufgegriffen und vorgestellt werden, beschäftigt sich Kapitel 3 mit den gesetzten Zielen und Voraussetzungen für die Arbeit.

Kapitel 4 befasst sich im Anschluss ausschließlich mit der Vorstellung des erstellten Tools und geht genauer auf die Vorgehensweise ein. Darauf aufbauend wird in Kapitel 5 das entstandene Tool hinsichtlich der Leistungsfähigkeit und die Ergebnisse, die es geliefert hat, hinsichtlich ihrer Aussagekraft untersucht. Dabei aufgetretene Probleme und Schwierigkeiten sowie die Ergebnisse eines beispielhaften Benchmark-Testlaufs runden das Kapitel inhaltlich ab.

Als Abschluss der Arbeit wird dann in Kapitel 6 eine Zusammenfassung der wichtigsten Erkenntnisse und eine Übersicht über prinzipielle Möglichkeiten für weiterführende Arbeiten gegeben.

Kapitel 2

Grundbegriffe und State of the Art

Für eine detailliertere Übersicht des State of the Art der Thematik des Benchmarkings von Navigationsleistungen eines mobilen Roboters befinden sich, wie bereits in der Einleitung erwähnt, weiterführende Informationen und eine umfangreichere Auswahl an Bewertungsmaßen in [GOLDA, 2014].

Auf den nun folgenden Seiten soll deshalb vor allem eine Übersicht von Begrifflichkeiten und Bewertungsmaßen gegeben werden, die im Rahmen der Arbeit auch Verwendung gefunden haben.

2.1 Grundbegriffe

In den einführenden Worten, aber auch auf den kommenden Seiten werden einige Begriffe besonders häufig verwendet, da sie den Kern dieser Arbeit ausmachen. Aus diesem Grund sollen im Folgenden diese Begriffe aufgegriffen und kurz erläutert werden.

2.1.1 Benchmark

In der Literatur zu Navigationsbenchmarks werden im Wortlaut unterschiedliche Definitionen für den Begriff des „Benchmarks“ vorgeschlagen. Der Kerngedanke ist jedoch bei allen der Gleiche: Bezeichnet wird mit dem Begriff die vergleichende Analyse von

Ergebnissen mit einem festgelegten Bezugswert. Diese Bezugswerte erhält man durch Bewertungsmaße.

2.1.2 Bewertungsmaß / Metrik

Bewertungsmaße dienen als Indikatoren für die - in diesem Falle vom Roboter - erbrachten Leistung. Mit dem Begriff des „Bewertungsmaßes“ stark verwandt ist der aus der Mathematik bekannte Begriff der „Metrik“. Der Begriff der Metrik ist hierbei wie folgt definiert.

Sei X eine beliebige Menge. Eine Abbildung $d : X \times X \rightarrow \mathbb{R}$ heißt Metrik auf X , wenn für Elemente x, y und z von X die folgenden Axiome gelten:

1. $d(x, y) \geq 0$ und $d(x, y) = 0 \Leftrightarrow x = y$
2. $d(x, y) = d(y, x)$
3. $d(x, y) \leq d(x, z) + d(z, y)$

Ein Bewertungsmaß muss diese Axiome nicht zwingend erfüllen. So kann zum Beispiel bereits der minimale Wert von Daten - welcher unter Umständen auch negativ sein kann - eine Aussage über die Leistung eines Algorithmus liefern.

2.1.3 Navigation

Der Begriff der Navigation setzt sich aus drei Komponenten zusammen:

- Lokalisation - Ausgehend von einer existierenden Karte versucht der Roboter seine Position zu bestimmen. Dies geschieht anhand von aktuell wahrgenommenen Sensoreindrücken, z.B. durch Laserscans, indem das aktuell wahrgenommene Umgebungsmuster mit einer entsprechenden Karte verglichen wird.
 - Pfadplanung - Von der aktuell bekannten Position versucht der Roboter einen Weg zu einem ihm vorgegebenen Zielpunkt zu bestimmen.
-

- Kartierung - Bei der Kartierung handelt es sich um die Fähigkeit, eine Karte der Umwelt zu erzeugen, durch die es möglich wird, Pfade zu planen oder die eigene Position zu bestimmen.

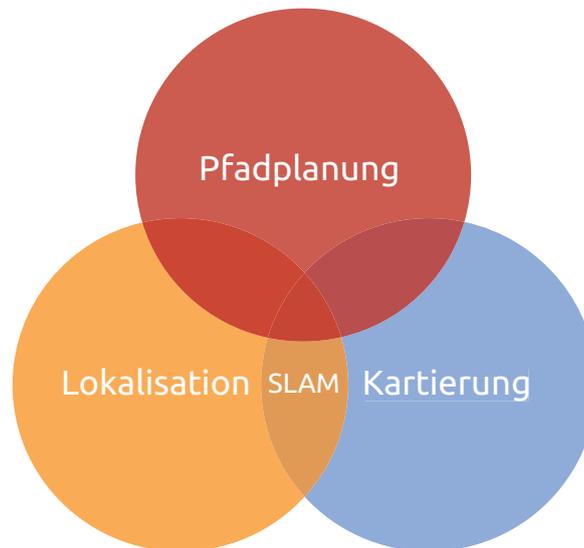


Abbildung 2.1: Bestandteile der Navigation

Neben den einzelnen Komponenten in der Ursprungsvariante ist oftmals auch der Schnitt aus Lokalisation und Kartierung, das sogenannte Simultaneous Localization and Mapping (kurz: SLAM) von Interesse.

2.2 State of the Art

Da es sich bei der Bestrebung, das Navigationsverhalten von Robotern zu vergleichen, um eine relativ junge Bewegung handelt, beschränkt sich die Auswahl an bereits vorhandenen Benchmarks auf eine geringe Anzahl. Abhängig davon, welche der in 2.1.3 vorgestellten Komponenten der Navigation untersucht werden soll, bietet die Literatur somit eine mehr oder weniger ausgeprägte Auswahl an Bewertungsmaßen. Als gute Übersicht von Bewertungsmaßen, wie sie an einigen Wettbewerben und Initiativen vorgeschlagen und umgesetzt werden, haben sich [Mob, 2010] und die EvAAL Competition (*Evaluating Ambient Assisted Living Systems through Competitive*

Benchmarking) erwiesen. Bei Letzterer handelt es sich um eine Initiative, die durch das Anbieten eines regelmäßigen Wettbewerbes Benchmarks und Bewertungsmaße für AAL-Systeme (kurz: **A**mbient **A**ssisted **L**iving Systeme, zu *Deutsch*: „Altersgerechte Assistenzsysteme für ein selbstbestimmtes Leben, umgebungsunterstütztes Leben, selbstbestimmtes Leben durch innovative Technik oder Assistenzsysteme fürs Alter“) zu etablieren versucht.

Die nun folgenden Bewertungsmaße wurden weitestgehend aus den oben genannten Quellen übernommen. Der Fokus der Untersuchung der Navigationsleistung liegt dabei auf den beiden Teilgebieten der Lokalisation und Pfadplanung. Für die Kartierung wurden weder bei EvAAL noch bei den anderen recherchierten Quellen Verfahren und Bewertungsmaße vorgestellt. In [Mob, 2010] unterscheidet man hinsichtlich der Bewertungsmaßen nicht direkt zwischen Lokalisation und Pfadplanung, sondern teilt diese in die Kategorien „Sicherheits-“, „Dimensions-“ und „Smoothnessmetriken“ ein. Auf Letztere wird in dieser Arbeit nicht weiter eingegangen.

2.2.1 Sicherheitsmetriken

Vorgestellt werden dabei drei Sicherheitsmetriken. Alle betrachten die vom Roboter gemessenen Abstände zu Hindernissen in seiner Umgebung.

	Messpunkt 1	Messpunkt 2	Messpunkt 3
t = 1	10 cm	10 cm	15 cm
t = 2	15 cm	15 cm	20 cm
t = 3	15 cm	20 cm	19 cm
t = 4	30 cm	28 cm	22 cm

Abbildung 2.2: Beispieldaten für Sicherheitsmetrikbestimmung

Die Tabelle stellt beispielhaft von einem Lasersensor an drei Messpunkten bestimmte Werte für die Zeitschritte 1 bis 4 dar.

Die vorgestellten Sicherheitsmetriken werden dabei als Security Metric 1 (SM1), Security Metric 2 (SM2) und Security Metric 3 (SM3) bezeichnet. Anhand der in Abbildung 2.2 aufgezeigten Beispieldaten soll nun die Bestimmung der Werte gezeigt werden. Für ein besseres Verständnis kann man sich vorstellen, dass es sich bei den Messpunkten 1 bis 3 um einen Messpunkt zur Linken, einen nach vorne und einen zur Rechten des Roboters handelt. Für die allgemeinen Fälle sei n die Anzahl an Messpunkten und m die Anzahl an Zeitschritte.

- **SM1:** Die Werte aller Messpunkte zu jedem Zeitpunkt werden addiert und anschließend durch die Anzahl geteilt. Abbildung 2.2 stellt abstrahiert dar, wie eine solche Übersicht an Messwerten aussehen kann.

$$\text{SM1} = \frac{\sum_{i=1}^m \sum_{j=1}^n x_{i,j}}{m \cdot n} = \frac{219 \text{ cm}}{12} = 18.25 \text{ cm}$$

- **SM2:** Von jedem Messpunkt wird der minimal gemessene Abstand über alle Zeitschritte bestimmt. Anschließend wird die Summe über diese gebildet. Diese Metrik gibt Aufschluss darüber, welches Risiko der Roboter bei der Fahrt durch den Raum eingegangen ist, da sämtliche Abstände aus dem Sichtfeld des Roboters berücksichtigt werden.

$$\text{SM2} = \frac{\sum_{j=1}^n \min_{i \in \{1,2,\dots,m\}} x_{i,j}}{n} = \frac{35 \text{ cm}}{3} \approx 11.7 \text{ cm}$$

- **SM3:** Von allen aufgezeichneten Messwerten wird das Minimum gewählt. Es macht eine Aussage darüber, wie knapp der Roboter während des Benchmarks an ein Hindernis herangefahren ist.

$$\text{SM3} = \min_{\substack{i \in \{1,2,\dots,m\} \\ j \in \{1,2,\dots,n\}}} x_{i,j} = 10 \text{ cm}$$

Neben den drei eben genannten Sicherheitsmetriken wird insbesondere bei EvAAL betrachtet, zu wie vielen Kollisionen es während des Benchmarks gekommen ist. Diese Zahl gibt ebenfalls Aufschluss darüber, wie gut der Roboter Hindernissen ausweichen kann. Damit lässt sich auch direkt eine Aussage über die potentielle Akzeptanz eines

Roboters bei den Bewohnern des Haushalts angeben. Viele Kollisionen mindern die Akzeptanz sehr stark, weshalb das Bestimmen dieser Anzahl während des Benchmarks sich als sehr wichtige und nützliche Aussage erweist.

2.2.2 Dimensionsmetriken

Neben der Betrachtung der Sicherheit ist für die Bewertung der Navigation vor allem ausschlaggebend, wie sich die Qualität des Fahrverhaltens des Roboters zeigt. Die in [Mob, 2010] vorgeschlagenen Dimensionsmetriken liefern hierbei eine Auswahl entsprechender Bewertungsmaße, wie sie in der Literatur oftmals angewandt werden.

Pfadlänge

Die Berechnung der Pfadlänge folgt einer ganz simplen und sehr intuitiven Formel. Es handelt sich dabei um die Summe der Distanzen zwischen zwei aufeinanderfolgenden Positionen des Roboters, wie sie in der folgenden Formel zu sehen ist.

$$p = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

Abweichung zur Zielposition

Interessanter als die Pfadlänge ist die Abweichung zur Zielposition. Dabei kann die Zielposition einem anzufahrenden Ziel entsprechen oder aber der aktuellen Position auf der gefahrenen Trajektorie des Roboters. Eine Trajektorie stellt in diesem Kontext einen Pfad dar, wie er beispielsweise von der Pfadplanung des Roboters zu einem Zielpunkt bestimmt wurde oder die Spur, die effektiv vom Roboter gefahren wurde. In [Mob, 2010] wird dabei insbesondere der Mittelwert dieser Abweichungen bestimmt und als Metrik gewählt.

$$mgd = \frac{\int_{i=0}^l l_n^2 ds}{n}$$

wobei n die Anzahl an Punkten auf der Trajektorie ist und

$$l_i = \min(\forall n(\sqrt{(x_i - x_n)^2 + (y_i - y_n)^2}))$$

der Distanz zum nächsten Punkt auf der extern ermittelten Trajektorie zur aktuellen Position i entspricht.

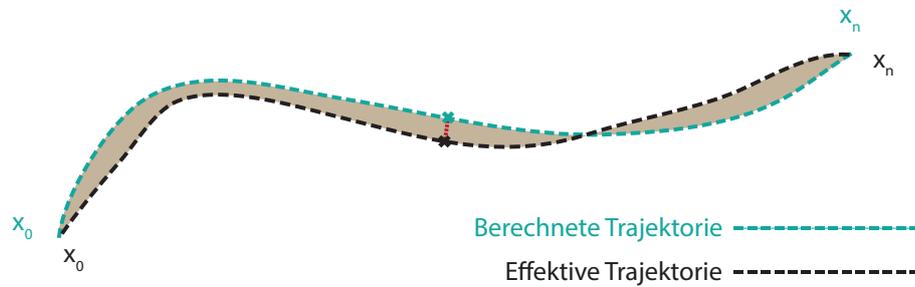


Abbildung 2.3: Abweichung der gefahrenen zur geschätzten Trajektorie des Roboters

Unterschiede zwischen der vom Roboter geschätzten Position und der wirklichen Position treten in unterschiedlichem Maße auf. Der Verlauf der beiden daraus resultierenden Trajektorien kann somit mehr oder weniger stark voneinander abweichen.

Aus dieser Metrik lassen sich dann Abwandlungen herleiten, die sich ebenfalls zur Bewertung des Roboterverhaltens nutzen lassen. Auch muss man sich bei der Bestimmung nicht allein auf den Mittelwert beschränken. Sehr interessant können auch die minimale, mehr aber noch die maximale Abweichung zwischen der geschätzten und der gefahrenen Trajektorie sein.

2.3 Zusammenfassung

Mit der Erklärung aller nötigen Begriffe und Grundlagen soll dem Leser nun ein Grundverständnis für die weitere Arbeit geschaffen worden sein. Auf den folgenden Seiten sollen diese Grundlagen nun weitergeführt werden und in der vorgestellten oder auch in angepasster Form für diese Arbeit weiterverwendet werden.

Dabei sollen insbesondere die Metriken, wie sie in der Literatur zu finden sind, für diese Arbeit als Grundlage dienen. Neben den in Kapitel 2.2 vorgestellten Bewertungsmaßen sollen auch weitere Metriken umgesetzt werden, die unter anderem auch die Aspekte Geschwindigkeit und Pfadplanungszeit des Roboters berücksichtigen. Hierzu finden sich in EvAAL weitere Vorschläge. Von den bereits vorgestellten Bewertungsmaßen finden sich in dieser Arbeit größtenteils abgewandelte Formen wieder, sowie einzelne Erweiterungen für gewisse Spezialfälle. Mehr Informationen dazu finden sich in Kapitel 4.1.2, in dem es um die gewählten und umgesetzten Bewertungsmaße geht.

Kapitel 3

Zielstellung

Nachdem in Kapitel 2 bereits ein Blick auf die Grundbegriffe und den State of the Art geworfen wurde, soll dieses Kapitel in erster Linie einen kurzen Überblick über das gewünschte Ergebnis der Arbeit liefern. In 3.1 soll dabei insbesondere eine Übersicht der zu evaluierenden Leistungen gegeben werden. Etwaige Voraussetzungen für die Durchführung der Benchmarks sollen dabei in 3.2 genannt und anschließend in Kapitel 4 und Kapitel 5 konkretisiert werden.

3.1 Fragestellungen

Bevor ein Tool zur Durchführung von Benchmarks entwickelt werden kann, stellt sich die Frage, was es an Funktionsumfang mitbringen soll. Insbesondere von Interesse ist dabei die Frage, welche Leistungen des Roboters genauer untersucht werden sollen. Hierbei wurden in Absprache mit dem Betreuer im Wesentlichen festgelegt, dass die Software in der Lage sein soll, Fragen hinsichtlich der Lokalisations- und der Navigationsleistung eines Roboters zu beantworten.

Die Auswertung der Lokalisation, das heißt sowohl die Position, als auch die Orientierung des Roboters zu jedem beliebigen Zeitpunkt, schließt insbesondere auch die vom Roboter angefahrenen Zielpunkte ein.

Neben der Lokalisation soll dabei auch die Frage beantwortet werden, wie gut die Na-

vigation des Roboters in der Wohnumgebung ist. Dies umfasst insbesondere Aussagen über die gefahrene Geschwindigkeit während des Benchmarks, die angefahrenen Zielpunkte, die gescheiterten Anfahrten von Zielen mit Angabe eines Grundes und die vom robotereigenen Laser Range Scanner gemessenen Entfernungsdaten zu Hindernissen.

3.2 Voraussetzungen

Sowohl die Untersuchung der Lokalisation, als auch die der Navigation haben eine Gemeinsamkeit, was die Voraussetzungen betrifft: Die Umgebung des Roboters ist in beiden Fällen eine künstlich nachgebaute Wohnumgebung. Weitere funktionspezifische Voraussetzungen werden in den folgenden Unterpunkten vorgestellt.

3.2.1 Lokalisation

Neben den allgemeinen Voraussetzungen hinsichtlich der Wohnumgebung, wurden für die Betrachtung der Lokalisation, weitere Einschränkungen und Prämissen festgelegt. Damit sich der Roboter lokalisieren kann, dürfen keine technischen Hilfsmittel, wie beispielsweise im Boden verbaute RFID-Tags oder Marker, verwendet werden. Es wird jedoch ausdrücklich hingewiesen, dass auch mobile Wohnungsgegenstände, wie zum Beispiel bewegliche Sitzmöbel oder Türen vorhanden sein können.

Um bestimmen zu können, an welcher Stelle sich der Roboter zu einem bestimmten Zeitpunkt befindet, muss sichergestellt sein, dass das System auf „Ground Truth“ Daten für die Position des Roboters zurückgreifen kann. Diese „Ground Truth“ Daten können beispielsweise Positionsdaten eines Laser Range Scanner Systems sein, wie sie bei den Experimenten in dieser Arbeit (siehe 5.2.1) Verwendung gefunden haben.

3.2.2 Navigation

Nachdem nun bereits die für die Lokalisation benötigte Umgebungsbedingungen vorgestellt wurden, folgen nun für die Navigation weitere Ergänzungen. Eine wichtige Erweiterung für die Benchmarks ist die Definition der Untergründe, auf welchen der Roboter agieren soll. Dabei wurde explizit festgelegt, dass es sich bei der Untergrundbeschaffenheit um eine heterogene Landschaft aus glatten, rauen und rutschigen Untergründen handeln soll. Diese umfassen sowohl glatten Untergründe wie zum Beispiel PVC-Boden, als auch Teppiche oder Teppichläufer mit Kanten, welche vom Roboter ebenfalls überfahren werden sollen. Die Läufer sollen dabei insbesondere nicht am Boden befestigt - d. h. festgeklebt oder -getackert - sein und somit ruhig verrutschen können. Diese ergänzenden Voraussetzungen ergeben eine weitere Schwierigkeit für den Roboter und erlaubt es, ein authentischeres Bild einer Wohnung wiederzugeben.

3.3 Zusammenfassung

Wie bereits in der Einleitung angesprochen, ist das Ziel der Arbeit ein Tool zu entwickeln, welches dem Nutzer erlaubt, Benchmarks möglichst einfach und weitreichend automatisiert durchzuführen und auszuwerten. Dabei ist das Ziel der Software, einen geringen Aufwand für die den Benchmark durchführende Person und die Sicherstellung der Reproduzierbarkeit aller nötigen Bedingungen zu gewährleisten.

Am Ende sollen die in diesem Kapitel genannten Dinge konkret umgesetzt werden und die Leistung eines Roboters nach selbst festgelegten und den Ansprüchen angepassten Erfolgskriterien eingestuft werden können.

Kapitel 4

Implementierung

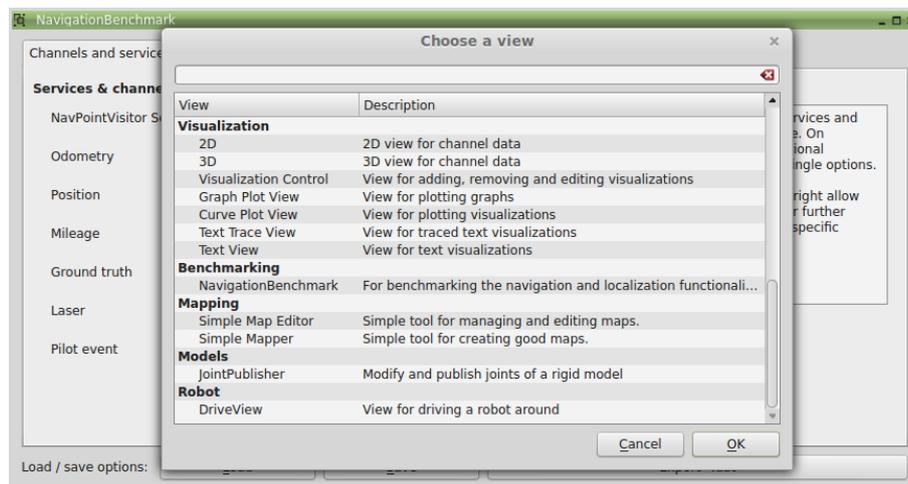


Abbildung 4.1: Öffnen des Benchmarkdialogs

Über die View-Auswahl des miracenters lässt sich mit wenigen Klicks auf das Benchmarktool zugreifen. Dieses befindet sich in einer eigenen Kategorie, in der sich zukünftige weitere Benchmarktools ebenfalls befinden werden.

Damit es möglichst einfach wird, die gewünschten Benchmarks umzusetzen, bietet es sich an, den Nutzer durch diesen Prozess zu führen. Dies kann am besten durch ein grafisches Tool, welches alle nötigen Einstellungen vom Nutzer entgegennimmt, erreicht werden. Da sämtliche im Fachgebiet eingesetzten Roboter das Roboter-Softwareframework „mira“ verwenden, bietet es sich an, die entsprechende Implemen-

tierung des Tools in eben diesem umzusetzen. Weiterhin erlaubt es auf relativ einfache Art und Weise, alle für einen Roboter notwendigen Funktionalitäten zu implementieren und liefert eine Vielzahl an nützlichen Tools zur Arbeit mit diesen. Aufgrund der Konstruktion des Softwareframeworks bietet „mira“ die Möglichkeit, verteilte Frameworks auf einfache Art und Weise zu verbinden und somit auf die Daten eines Roboters zugreifen zu können. Das Benchmark-Tool benötigt somit keine Eingriffe in die eigentliche Robotersoftware und kann einfach von einem anderen Rechner aus bedient und überwacht werden. Diese Fähigkeit, verschiedene Frameworks miteinander zu verbinden, umfasst auch die Möglichkeit, RPC-Aufrufe (kurz: Remote Procedure Call) auf einfache Art und Weise, über die Grenzen eines Rechners hinweg, auszuführen. RPC-Funktionen werden im Kontext von „mira“ als „Services“ bezeichnet. Ein weiteres wichtiges Konzept, welches in diesem Softwareframework Verwendung findet, sind „Channels“. Dabei handelt es sich um ein Mittel zur Kommunikation und zum Datenaustausch zwischen einzelnen Komponenten eines oder mehrerer Frameworks. Die Option, zwei Frameworks miteinander zu verbinden, schließt dabei auch das Darstellen und Einbinden der Channels der jeweiligen Instanz des Frameworks ein. Somit wird auch der Austausch und das Zurverfügungstellen aller möglichen Daten, unabhängig vom Typ und der Komplexität dieser, mittels einfach zu verwendenden Methoden und einem intuitiven Konzept, einfach nutzbar. Die im Rahmen des Benchmarks nötige Aufzeichnung der Daten auf den Channels kann ebenso durch „mira“ geschehen. Hierfür liefert es ein weiteres fundamentales Konzept, nämlich die Aufzeichnung von Daten in sogenannten „Tapes“. Diese Tapes beinhalten die Inhalte sämtlicher spezifizierter Channels und können nach der Aufzeichnung entweder abgespielt werden oder mittels eigener Algorithmen ausgewertet werden.

4.1 Aufbau

Die entwickelte Software setzt sich aus zwei Komponenten zusammen. Der erste und gleichzeitig größte Teil der Software wird in Form einer Toolbox für das Roboterframework „mira“ implementiert und stellt als Hauptkomponente einen View für das graphische Frontend „miracenter“ bereit, welcher der Steuerung und Überwachung der

Benchmarks dient. Im Anschluss an die Durchführung des Benchmarks folgt die Evaluation der gesammelten Daten. Dies geschieht durch Matlab-Skripte, deren Aufgabe es ist, dem Nutzer die Daten und wesentliche extrahierte Informationen grafisch und übersichtlich darzustellen. Im Folgenden werden die einzelnen Komponenten für sich genauer vorgestellt.

4.1.1 Das Benchmark-Tool

Das Roboter-Softwareraster „mira“ bildet wie bereits erwähnt die Grundlage der in dieser Arbeit entwickelten Software.

Die grafische Oberfläche des Tools setzt sich aus drei inhaltlich getrennten Tabs zusammen. Neben den funktionstechnisch gekapselten Tabs selbst, verfügt das Tool über globale Buttons für den Import und Export aller vorgenommenen Einstellungen als XML-Datei.

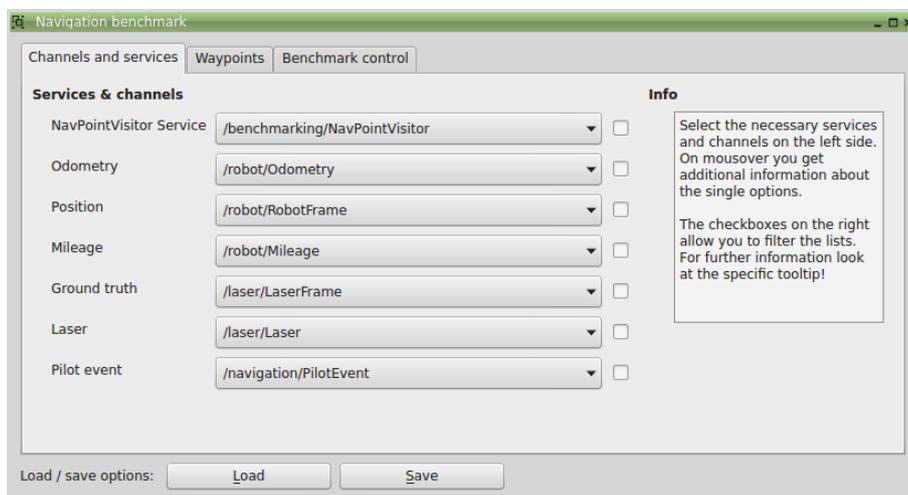


Abbildung 4.2: Channels and Services

Bietet eine Auswahl verschiedener Channeltypen und Services, aus denen der Nutzer die richtig auszuwählen hat.

Channels & Services

Dieser Teil des Tools besteht aus einer Liste von Dropdownboxen, wie in Abbildung 4.2 zu sehen ist, über die der Nutzer die notwendigen Channels für die Aufzeichnung der Daten festlegen kann. Jede Dropdownbox beinhaltet dabei die Namen von Channels oder Services eines speziellen Typs. Für den Fall, dass die Übersicht aufgrund zu vieler Einträge nicht gewahrt werden kann, existiert zur Rechten jeder einzelnen Box eine Checkbox zum Filtern der Einträge anhand des Channelnamens (zusätzlich zum Typ). Beim Überfahren der Checkbox mit der Maus erscheint indes ein Tooltip, welcher dem Nutzer verrät, nach welchem Kriterium - in Form eines regulären Ausdrucks - die Liste nach Auswahl der Checkbox gefiltert wird.

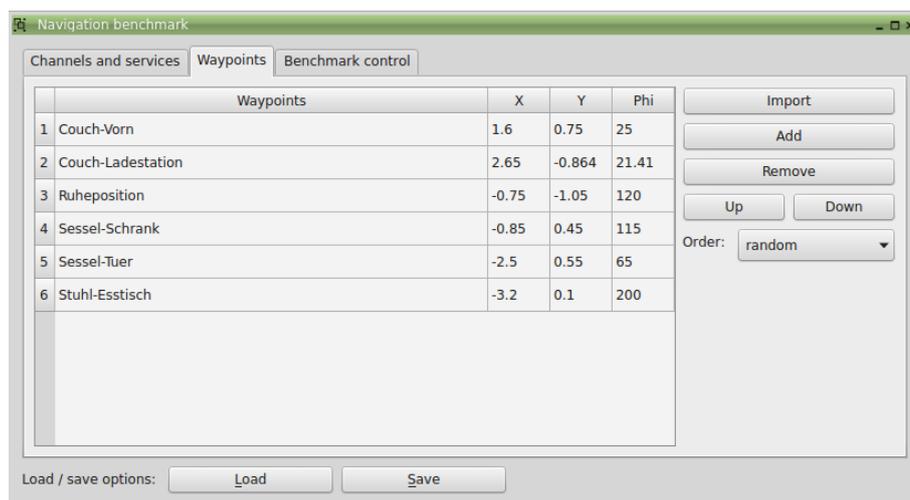


Abbildung 4.3: Waypoints

Auswahl der vom Roboter anzufahrenden Zielpunkte, spezifiziert durch einen Namen, der Position in x - und y -Koordinate und der Orientierung in Grad.

Waypoints

Um dem Roboter Zielpunkte zum Anfahren vorzugeben, liefert das Tool die Möglichkeit manuell Punkte festzulegen. Abbildung 4.3 zeigt auf, wie dies, durch Angabe eines (eindeutigen) Namen für den Punkt, sowie der Koordinaten und der Orientierung

in Grad an dieser Position, aussehen kann. Darüber hinaus besteht die Möglichkeit, Einträge zu löschen oder in ihrer Reihenfolge zu verändern. Eine zusätzliche Auswahlmöglichkeit auf der rechten Seite erlaubt es, dem Roboter vorzugeben, in welcher Reihenfolge die eingetragenen Punkte angefahren werden sollen. Hierbei existiert die Möglichkeit, die Ziele zufällig oder in der Reihenfolge, in der sie in der Liste dargestellt werden, anzufahren. Neben der globalen Option Einstellungen zu importieren, liefert der Tab die Möglichkeit, aus einer XML-Datei eine Liste von Zielpunkten zu extrahieren.

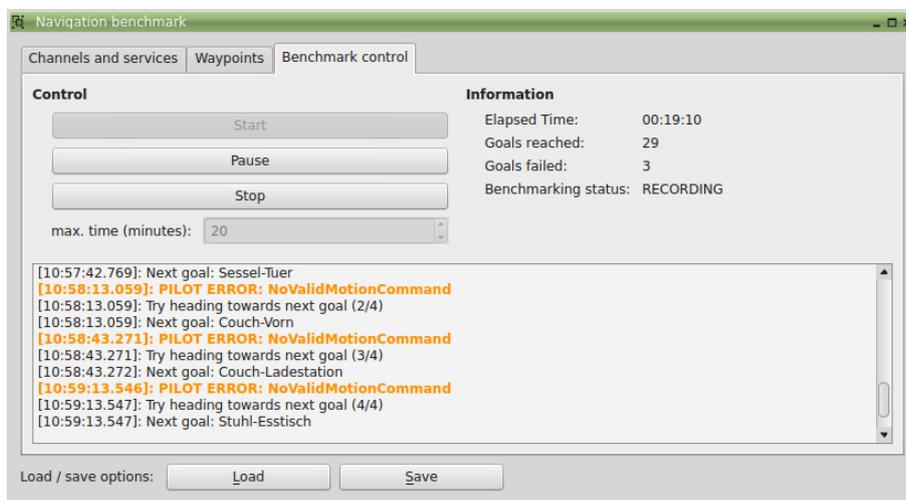


Abbildung 4.4: Benchmark control

Die Abbildung stellt einen bereits fortgeschrittenen Benchmarkdurchlauf dar. Neben der Anzahl an bereits erreichten und nicht erreichten Ziele, sind in der Textbox auch Hinweise und Fehlermeldungen zu erkennen.

Benchmark control

Der in Abbildung 4.4 dargestellte dritte und letzte Tab beinhaltet die Steuerelemente. Er erlaubt es, einen Benchmark zu starten, zu pausieren oder komplett zu beenden. Außerdem liefert er während des Benchmarks einige Informationen, die von Interesse sein können. Beispiele hierfür sind die Anzahl an erfolgreichen und fehlgeschlagenen

Zielanfahrten oder Hinweise in Form eines Textlogs. Bei diesen Hinweisen kann es sich entweder um Warnungen oder Fehlermeldungen handeln oder aber um kurze Infomeldungen, wie zum Beispiel das als nächste angefahrne Ziel. Versehen mit einem Zeitstempel erlaubt es somit jederzeit nachzuvollziehen, wie viel Zeit zwischen zwei Ereignissen vergangen ist.

4.1.2 Die Evaluationskripte

Eine Auswahl an Matlab-Scripten setzt die in Kapitel 2.2 vorgestellten Bewertungsmaße um. Dabei handelt es sich in erster Linie um Erweiterungen dieser oder um die Anwendung der Grundideen für die Erstellung eigener Benchmarks. Konkret ausgedrückt handelt es sich um fünf umgesetzte Auswertungen der Daten, die dem Nutzer nach Ausführen des Start-Skripts auf dem Bildschirm angezeigt werden. Eine Kopie der Grafiken wird zeitgleich als PDF im Arbeitsverzeichnis von Matlab angelegt.

Abweichung der Roboterposen

Aus den vom Roboter geschätzten Posen und denen, die von den externen Trackereinheiten bestimmt wurden, werden die Abweichungen berechnet und visuell aufbereitet. Als Output erscheinen zwei Fenster mit jeweils zwei Diagrammen, wie in Abbildung 4.5 (a) und (c) dargestellt. Bei den beiden Fenstern wird einmal die Abweichung der Position und einmal die der Rotation visualisiert. Dabei zeigt das erste Diagramm jeweils sämtliche Positionen des Roboters, die von den Trackern wahrgenommen wurden. Die Abweichungen werden dabei farblich anhand der Positionspunkte dargestellt. Eine grüne Darstellung steht dabei für eine geringe bis sehr geringe Abweichung, während eine rote Darstellung eine starke Abweichung darstellt. Der Schwellwert für den Übergang zwischen „guter“ und „schlechter“ Leistung kann dabei vom Benutzer durch Anpassen der entsprechenden Parameter im Kopfteil des Skriptes verändert werden.

Dauer von Pfadplanungsprozessen

Abbildung 4.5 (b) stellt für alle durchgeführten Pfadplanungsprozesse die Zeit dar, die vom Start der Pfadplanung bis zur Bewegung des Roboters verstrichen ist. Dabei

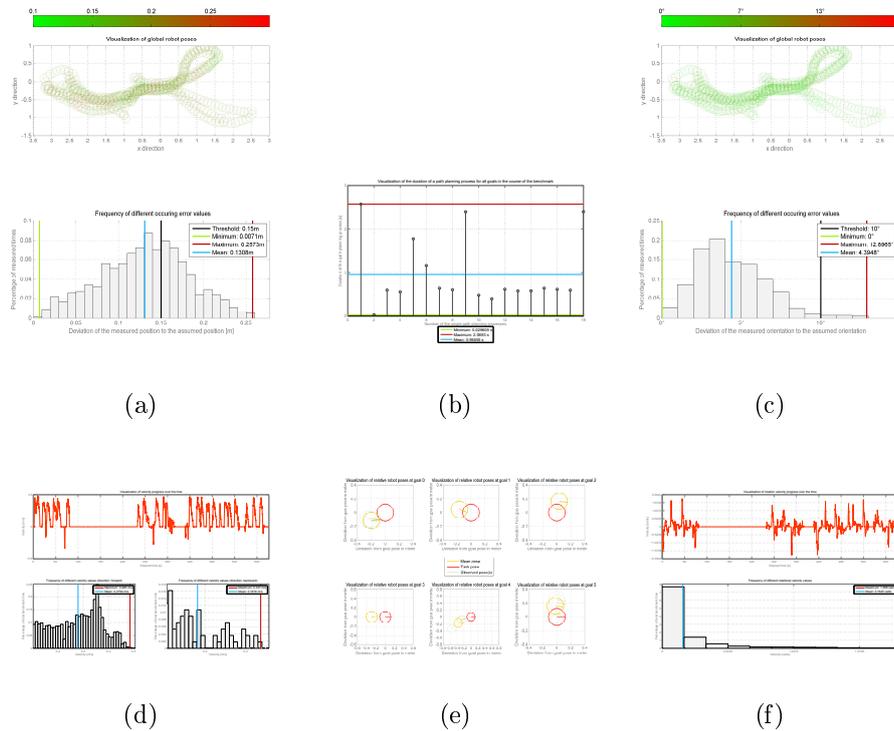


Abbildung 4.5: Übersicht des Outputs der Evaluationskripte

Nach dem Durchlauf des Benchmarks und der anschließend durchgeführten Evaluation, liefern die Matlab-Skripte grafische Auswertungen der Daten. Eine größere Darstellung der Einzelausgaben findet sich in Kapitel 5.3.

werden neben den Zeiten selbst, durch farbige Linien, die maximale und im Mittel verstrichene Zeit bis zum Losfahren des Roboters dargestellt. Die einzelnen Zeiten sind dabei chronologisch sortiert, angefangen vom ersten Navigationstask bis hin zum letzten.

Geschwindigkeitsanalysen

Wie bei der Untersuchung der Abweichung der Roboterposen wird auch bei der Geschwindigkeitsanalyse zwischen der Translation und der Rotation unterschieden. Abbildung 4.5 (d) und (f) stellen dabei die translationale respektive rotationale Geschwindigkeitsanalyse dar. Die in beiden Figuren ausgegebene obere Teilgrafik stellt den Ge-

schwindigkeitsverlauf während des Benchmarks dar. Durch die untere Teilgrafik wird dabei in Form eines Histogramms die Häufigkeit der Geschwindigkeiten, die der Roboter während des Benchmarks gefahren ist, visualisiert. Dabei wird bei der Translation zwischen dem Vorwärts- und dem Rückwärtsfahren unterschieden. Bei der Rotation hingegen wird lediglich der Betrag der Rotationsgeschwindigkeit betrachtet, das heißt also, wie schnell sich der Roboter gedreht hat. Die Drehrichtung wird hierbei jedoch nicht weiter betrachtet.

Evaluation der Zielfahrt

Bei dieser Variante der Darstellung von Daten handelt es sich in erster Linie um eine qualitative Übersicht. Bei der Anfahrt von Zielpunkten besitzt der Roboter einen Spielraum, in welchem er ein Ziel als „erreicht“ ansieht. Um zu sehen, ob und wie weit der Roboter auch wirklich in diesem Bereich landet, wird eine Übersicht für jeden einzelnen Zielpunkt und die relative erreichten Posen gegeben. Ein Beispiel ist in Abbildung 4.5 (e) zu sehen. Dabei stehen die grünen Objekte für die effektiv angefahrenen Posen, das gelbe Objekte für die aus diesen gemittelte Pose und das rote Objekt als Referenzpose.

Abstand zu Hindernissen

Das letzte, im Rahmen der Arbeit, umgesetzte Bewertungsmaß ist in Anlehnung an die in Kapitel 2.2.1 vorgestellten Sicherheitsmetriken implementiert worden. Dabei handelt es sich um keine Visualisierung wie die bisher vorgestellten Evaluationskripte. Dem Nutzer werden nach der Ausführung dieses Skriptes vier numerische Werte dargestellt, wobei es sich um den minimalen und maximalen Wert, sowie den Mittelwert und die Standardabweichung der Daten handelt. Bei den in der Auswertung zugrundeliegenden Daten handelt es sich nicht um sämtliche gemessene Werte, sondern bereits vom Tool vorverarbeitete Daten. Dabei exportiert dieses zu jedem Zeitpunkt den minimal gemessenen Abstand zu einem Hindernis in der Umgebung des Roboters. Am Ende enthält die exportierte Textdatei somit eine Liste von jeweils kleinsten Abständen zu jeglichen Zeitpunkten während des gesamten Benchmarks.

4.2 Funktionsweise

Nachdem nun bereits die Komponenten des in dieser Arbeit entwickelten Tools näher vorgestellt wurden, soll nun eine Übersicht der Funktionsweise dessen gegeben werden. Das Tool lässt sich starten, indem im miracenter die View-Übersicht ausgewählt wird und anschließend aus der Kategorie „Benchmarking“ der Eintrag „NavigationBenchmark“ ausgewählt wird. Diese ist in Abbildung 4.1 zu sehen. Anschließend öffnet sich eine Instanz des Benchmark-Tools, welches in Abschnitt 4.1.1 in seinem Aufbau bereits näher vorgestellt wurde. Der Nutzer hat nun alle für das Benchmarken der Navigation benötigten Channels auszuwählen und eine Liste von Zielpunkten anzugeben, die der Roboter anfahren soll. Dabei gilt sicherzustellen, dass alle benötigten Datenquellen vorhanden sind. Dies schließt insbesondere auch die ordnungsgemäße Inbetriebnahme der Lasertracker und des Roboters ein. Bevor der Benchmark gestartet werden kann, ist der Nutzer jedoch noch darauf angewiesen eine Dauer für den Benchmark anzugeben, nach welcher dieser gestoppt werden soll. Nachdem alle Einstellungen getroffen wurden, lässt sich der Benchmark starten.

Mit dem Beginn des Benchmarks fährt der Roboter nun wiederholt alle festgelegten Ziele an. Dabei werden sämtliche Daten, die zuvor im Tool spezifiziert wurden, in einem oder mehreren Tapes aufgezeichnet. Um die Menge der aufgezeichneten Daten nicht künstlich aufzublähen, wird beim Pausieren - egal ob durch den Nutzer initiiert oder der automatischen Pause bei Fehlern im Ablauf - ein Tape abgeschlossen. Sobald der Benchmark wieder aufgenommen wird, beginnt die Aufnahme mit einem weiteren Tape.

Sobald das gesetzte Zeitlimit erreicht oder der Benchmark manuell vom Nutzer gestoppt wurde, beginnt das Tool zunächst die aufgezeichneten Tapes zu analysieren und die Daten in Textdateien zu exportieren. Dabei werden je nach Art der Daten nur ein Teil der auf den Channels befindlichen Daten auch extrahiert. So wird hinsichtlich der Lokalisation beispielsweise die Position alle zehn Zentimeter abgespeichert. Anschließend erhält der Nutzer die Aufforderung, die aus den Aufzeichnungen extrahierten Daten zu speichern. Er wird angewiesen, ein Ziel-Verzeichnis auszuwählen, in welches die extrahierten Daten gespeichert werden sollen.

Der Nutzer hat anschließend lediglich das Skript „start_evaluation.m“ in Matlab auszuführen. Nach einer Weile erscheint eine Reihe von Fenstern mit den Visualisierungen der in Abschnitt 4.1.2 vorgestellten Bewertungsmaße.

4.3 Zusammenfassung

Das Softwareframework „mira“ wurde in dieser Arbeit um ein Tool zum Benchmarken der Navigationsleistung erweitert. Es wurde darauf geachtet, eine möglichst einfache und intuitive Nutzung zu ermöglichen. Eine ausführliche Beschreibung für die Vorgehensweise und Nutzung des Tools findet sich im Anhang (siehe Anhang A.1). Unterstützt wird der Nutzer dabei durch zahlreiche Tooltips. Jegliche einstellbare Optionen wurden mit solchen versehen, um eventuelle Unklarheiten direkt vor Ort zu beseitigen.

Für die Visualisierung der extrahierten Daten wurde Matlab als Tool gewählt, da es sehr gut zum Arbeiten mit großen Mengen an Daten geeignet ist und eine einfach zu nutzende und anpassbare Form der Auswertung der Daten liefert.

Mit der Entwicklung dieses Tools soll eine Grundlage für zukünftige weitere Benchmarktools geschaffen werden, deren Ziel es ebenfalls sein soll, dem Nutzer Arbeit abzunehmen und die Experimente zu automatisieren.

Im nun folgenden Kapitel soll der allgemeine experimentelle Aufbau für die Durchführung der Benchmarks näher beleuchtet werden. Außerdem sollen beispielhaft die Daten eines durchgeführten Benchmarks betrachtet werden und ein wenig auf die Ergebnisse eingegangen werden.

Kapitel 5

Experimentelle Untersuchungen

Wie im bisherigen Verlauf der Arbeit bereits häufiger erwähnt wurde, sind neben der Implementierung eines Tools auch folglich Tests und experimentelle Untersuchungen der Software Teil der Arbeit.



Abbildung 5.1: Wohnlabor im Fachgebiet Neuroinformatik und Kognitive Robotik
Die Testumgebung für die Benchmarks. Eingerichtet nach dem Vorbild einer richtigen Wohnung.

Dabei wird auf den folgenden Seiten zu Beginn auf den benötigten Hardwareaufbau und die Testumgebung eingegangen. Dabei werden sowohl das Wohnlabor als auch die benötigten Hardwarekomponenten vorgestellt. Im Anschluss wird ein beispielhaft durchgeführter Testlauf der Software als Demonstration herangezogen. Die Ergebnisse des Benchmarkdurchlaufs liefern reale Werte, die dazu dienen sollen, die Funktionalität der Software und die verwendeten Bewertungsmaße zu stützen und vorzustellen. Weiterhin soll auf Probleme bei der Durchführung der Benchmarkdurchläufe eingegangen werden. Welche Probleme - insbesondere Probleme der realen Umwelt und der Hardware - sind aufgetreten und wie wurden diese gelöst? Am Ende des Kapitels folgt dann eine Zusammenfassung und Bewertung des gesamten Benchmarksystems und der durchgeführten Versuche.

5.1 Das Wohnlabor

Das Wohnlabor ist ein eigens für die Simulation und Durchführung von Versuchen eingerichteter Raum. Es stellt den Wohnbereich nach, wie er in vielen Haushalten älterer Menschen zum Standard gehört. Dabei existieren in der Umgebung sowohl engere Passagen wie Durchgänge, aber auch Bücherregale, Sessel, Tische, Lampen und Sofas.

5.2 Hardwareaufbau

Damit es möglich wird, mit der Software Benchmarks durchzuführen, müssen von Seiten der Hardware einige Voraussetzungen erfüllt sein. Neben dem eigentlichen Roboter, mit dem der Benchmark durchgeführt werden muss, werden vor allem die folgenden Komponenten benötigt und sind fundamentaler Bestandteil des Hardwaresetups.



Abbildung 5.2: Foto eines Lasertrackers

Sie kommunizieren über WLAN mit der Empfangsstation, die am Rechner angeschlossen ist.

- Einen **Rechner** auf dem die nötige Software installiert ist. Die schließt das „mira“ Softwareframework und das in dieser Arbeit entwickelte Benchmarktool ein.
- Mindestens zwei **Lasertracker** zum stetigen Verfolgen der aktuellen Roboterposition.
- Einen **Hutaufsatz** für den Roboter, welcher dem Tracken des Roboters dient.

5.2.1 Die Lasertracker

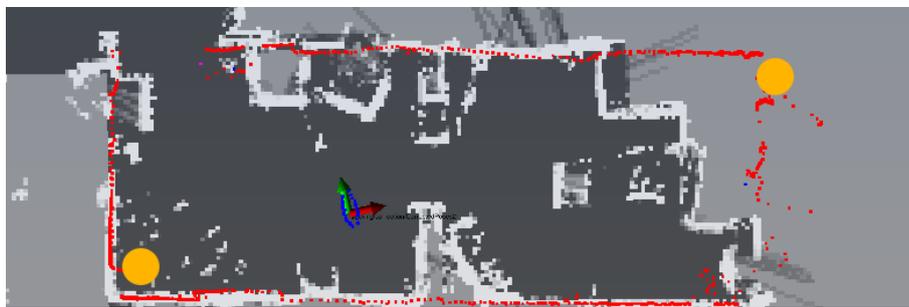


Abbildung 5.3: Übersicht des Trackeraufbaus

Zwei Tracker sind an gegenüberliegenden Positionen im Raum aufgestellt, hier gekennzeichnet durch die orangenen Kreise.

Die Aufgabe der Lasertracker ist das Tracken des Roboters im Wohnlabor. Sie werden somit genutzt, um die „Ground Truth“ Daten zur Position des Roboters zu erhalten. Hierfür müssen diese Tracker so im Raum verteilt sein, dass eine möglichst weitläufige Abdeckung mit ihnen möglich ist. Dabei muss die Abdeckung alle vom Roboter erreichbaren und für den Benchmark relevanten Zonen einschließen. Abbildung 5.3 zeigt beispielhaft eine mögliche Aufbaukonstellation von zwei Trackereinheiten, wie sie auch in den Tests in Abschnitt 5.3 Anwendung gefunden hat. Ein Tracker deckt für sich alleine einen Bereich von 270° ab und erlaubt somit, einen rechteckigen Raum von jeder beliebigen Position komplett zu erfassen. Ein Problem das mit zunehmender Entfernung zwischen Roboter und Tracker entsteht, ist, dass die relative Größe des Roboters zum Tracker immer geringer wird. Dies hat zur Folge, dass die Lasermesspunkte immer weiter auseinander liegen und somit immer weniger den Roboter

erfassen. Dadurch wird das Bestimmen der Position und insbesondere auch der Orientierung sehr erschwert. Im schlimmsten Fall kann es passieren, dass der Roboter sich in einer ungünstigen Orientierung befindet und somit genau zwischen zwei Messpunkten liegt. Die Konsequenz ist, dass der Roboter somit von den Lasertrackern nicht mehr gesehen wird.

Mit jedem weiteren Tracker, der in den Aufbau integriert wird, steigt jedoch wiederum die Anzahl an Messpunkten, wodurch es möglich wird, auch in größeren oder in verwinkelten Räumen respektive Räumen mit verdeckten Bereichen den Roboter zu erfassen. Auch für die Verfolgung mehrerer Ziele im Raum ist es nötig, mehrere Tracker einzusetzen.

5.2.2 Der Hutaufsatz



(a) Erste Variante des Hutes. Hergestellt aus einem festen Schaumstoff sowie einer metallischen Halterung zur Fixierung des Hutes am Roboter. (b) Überarbeitete und gewichtsoptimierte Version des Hutaufsatzes aus Papier und Karton, angebracht an der Metallhalterung.

Abbildung 5.4: Aufbauten für den Roboter

Damit der Roboter auch von den externen Trackereinheiten verfolgt werden kann, benötigt er einen speziellen Aufbau. Damit ist er groß genug, um im Erfassungsbereich der Lasertracker zu landen und somit von diesen erkannt zu werden.

Wie in den einleitenden Worten für dieses Kapitel bereits erwähnt, benötigt der Roboter, mit dem die Benchmarks durchgeführt werden sollen, einen speziellen Aufsatz.

Dieser Aufsatz wird ihm wie ein Hut aufgesetzt und dient dazu, die Höhe des Roboters zu vergrößern. Der Roboter verfügt bereits über Druckknöpfe, die am „Kopf“ angebracht sind. Diese Druckknöpfe werden genutzt, um die Halterung für den Hut am Roboter zu befestigen. Erst durch diesen Aufsatz kann der Roboter auch von den externen Trackern erkannt werden.

Insgesamt wurden in der Arbeit zwei verschiedene Huttypen gebaut und getestet. Diese Hüte sind in Abbildung 5.4 zu sehen. Wie zu erkennen ist, sind beide Hüte durch eine breite Form gekennzeichnet. Diese war nötig, um den externen Trackern und der darauf basierenden Software erst das Bestimmen der Orientierung des Roboters zu ermöglichen. Ausschlaggebend für die Wahl eines entsprechenden Hutes waren neben Gewicht und Größe weiterhin auch die Stabilität der Konstruktion. Der im ersten Versuch entworfene Hut-Prototyp erwies sich, trotz seiner relativ leichten Bauweise und stabilen Konstruktion, als weniger gut geeignet. Durch die Größe der Konstruktion schwankte diese sehr stark beim Beschleunigen und Abbremsen des Roboters, wobei es oftmals zu Problemen und Verfälschungen der Messungen kam. Auch die Form des Aufsatzes wies einige Schwächen auf. Insbesondere die Tatsache, dass der Aufsatz relativ dünn war, führte in einigen Situationen dazu, dass der Roboter entweder gar nicht oder lediglich als einziger Messpunkt von den Trackern erkannt wurde. Weitere Informationen zu aufgetretenen Problemen finden sich in Abschnitt 5.4.

5.3 Ergebnisse

Während der Bearbeitung dieses Themas wurden eine Vielzahl an Testläufen zur Überprüfung der Funktionalität und Stabilität der Software vorgenommen. Insbesondere die Tests am Ende der Implementierungsphase dienten auch als erste komplette Benchmarkdurchläufe. Um dem Leser ein besseres Verständnis dafür zu geben, wie die Ergebnisse eines solchen Durchlaufs aussehen können, wird in diesem Abschnitt ein Blick auf einen konkreten Benchmark geworfen.

Hierbei handelte es sich um einen kurzen Durchlauf mit einer Gesamtdauer von zehn Minuten. Festgelegt wurden insgesamt sechs Zielpunkte in der Wohnumgebung, welche vom Roboter in zufälliger Reihenfolge angefahren wurden. Dabei führte er im

gesamten Durchlauf insgesamt 18 Zielfahrten durch. Die Ergebnisse werden dabei in erster Linie in Form der Ausgabe der Evaluationskripte vorgestellt und durch einige erklärende Worte ergänzt. Der Roboter nutzte für die Lokalisation und Navigation dabei lediglich den intern verbauten Hauptlaserscanner, welcher in Blickrichtung des Roboters orientiert ist. Der zweite Laserscanner, der in die entgegengesetzte Richtung orientiert ist, war während des Testdurchlaufs nicht aktiv und hatte keinerlei Einflüsse auf die Lokalisation und Navigation des Roboters.

5.3.1 Abweichung der Roboterposen

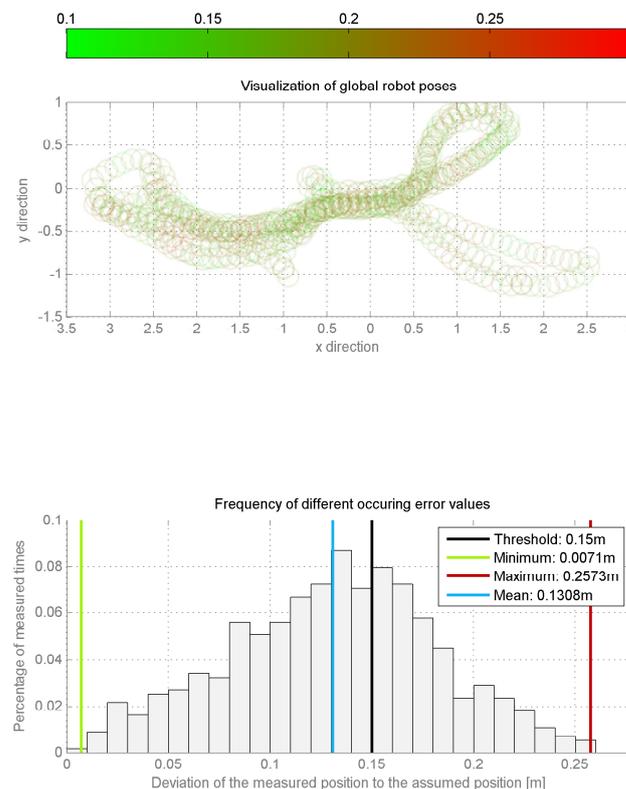


Abbildung 5.5: Visualisierung der Roboterposen und Positionsfehler
Darstellung der gefahrenen Trajektorie im Verlauf des zehnminütigen Testlaufs und den gemessenen Abweichungen zu den vom Roboter vermuteten Positionen.

Sowohl Abbildung 5.5 als auch Abbildung 5.6 stellen in der oberen Hälfte der jeweiligen Abbildung die vom Roboter gefahrene Trajektorie dar. Dabei handelt es sich bei den Kreisen um sämtlichen Positionen, an denen sich der Roboter während der Zeit befunden hat. Der Unterschied in beiden Grafiken ist die farbliche Darstellung der Kreise. Während in Abbildung 5.5 die Positionsabweichung in Meter zwischen vermutterter und gemessener Position dargestellt wird, zeigt der obere Teil in Abbildung 5.6 die Abweichung des Rotationswinkels an den entsprechenden Messpunkten an. Diese

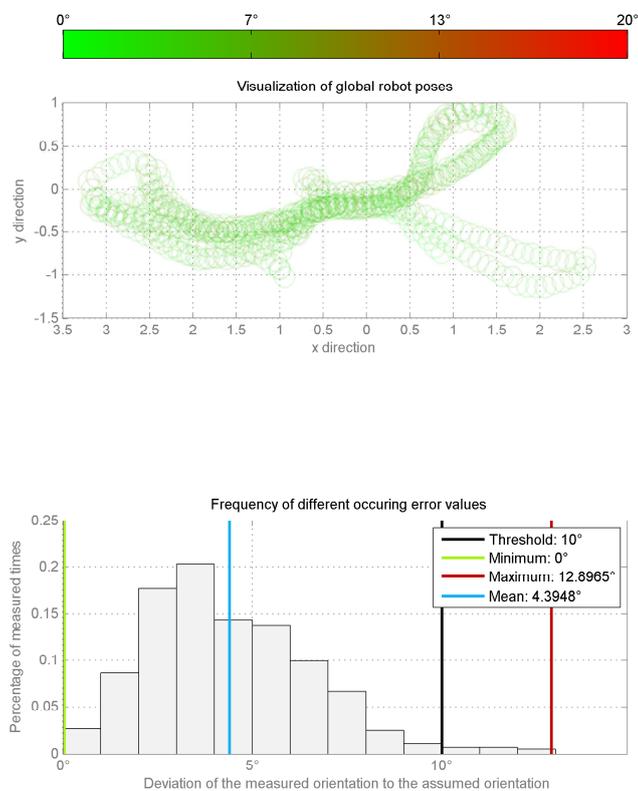


Abbildung 5.6: Visualisierung der Roboterposen und Orientierungsfehler
*Darstellung der gefahrenen Trajektorie im Verlauf des zehnminütigen Testlaufs und
den gemessenen Abweichungen zu den vom Roboter vermuteten Orientierungen.*

Darstellung erlaubt es eventuelle Zusammenhänge zwischen Fehler und Position zu erkennen. So ist beispielsweise bei genauerem Hinsehen zu erkennen, dass im oberen

linken und unteren rechten Bereich die Fehler im Mittel größer werden. Eine Ursache für diese Beobachtung kann in diesem Zusammenhang der Minimalaufbau mit zwei Trackereinheiten sein. Dieser weist starke Messschwächen auf, sobald sich der Roboter in den genannten Eckbereichen bewegt. Die Fehler sind also nicht eindeutig als Lokalisationsfehler des Robotersystems zu bestimmen, sondern mit hoher Wahrscheinlichkeit zumindest teilweise der Unsicherheit der „Ground Truth“ zuzuschreiben.

Der untere Teil der beiden Abbildungen stellt in Form eines Histogramms sämtliche bestimmte Abweichungen dar, jedoch ohne Bezug zur Position. Neben der dargestellten Häufigkeit der einzelnen Abweichungswerte, werden durch farbige Linien auch die maximale und die mittlere Abweichung angezeigt. Eine weitere Linie stellt einen im Skript festlegbaren Thresholdwert dar, um einen eigenen Grenzwert festzulegen und somit eine Orientierung zu erhalten.

Die effektiven Werte werden nochmals in einer im Histogramm befindlichen Box als Werte dargestellt.

5.3.2 Dauer der Pfadplanungsprozesse

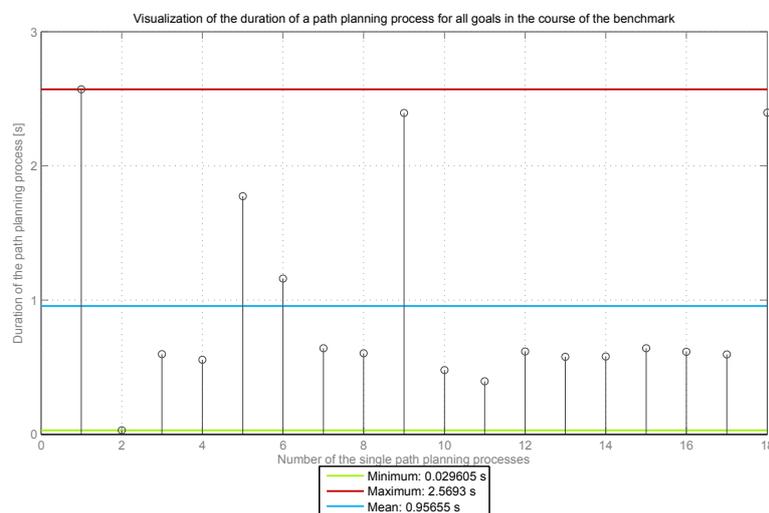


Abbildung 5.7: Visualisierung der Pfadplanungsdauern

Chronologische Darstellung der Laufzeiten für den Pfadplanungsprozess.

Die in Abbildung 5.7 dargestellten Punkte geben Auskunft über die Pfadplanungszeiten, die für sämtliche angefahrene Punkte benötigt wurde. Sie bestimmen sich aus der Zeit, die vom Starten der Pfadplanung bis hin zum Losfahren des Roboters verstreicht. Weiterhin werden wie in Abbildung 5.5 und 5.6 auch zusätzliche Linien angezeigt, die den minimalen, mittleren und maximalen Wert darstellen. Man sieht bereits mit dem ersten Blick, dass in den zehn Minuten des Durchlaufs meist weniger als eine Sekunde zwischen Planung der Strecke und Losfahren des Roboters verstrichen sind. Der maximale Wert lag dabei bei etwa 2.6 Sekunden.

5.3.3 Geschwindigkeitsanalysen

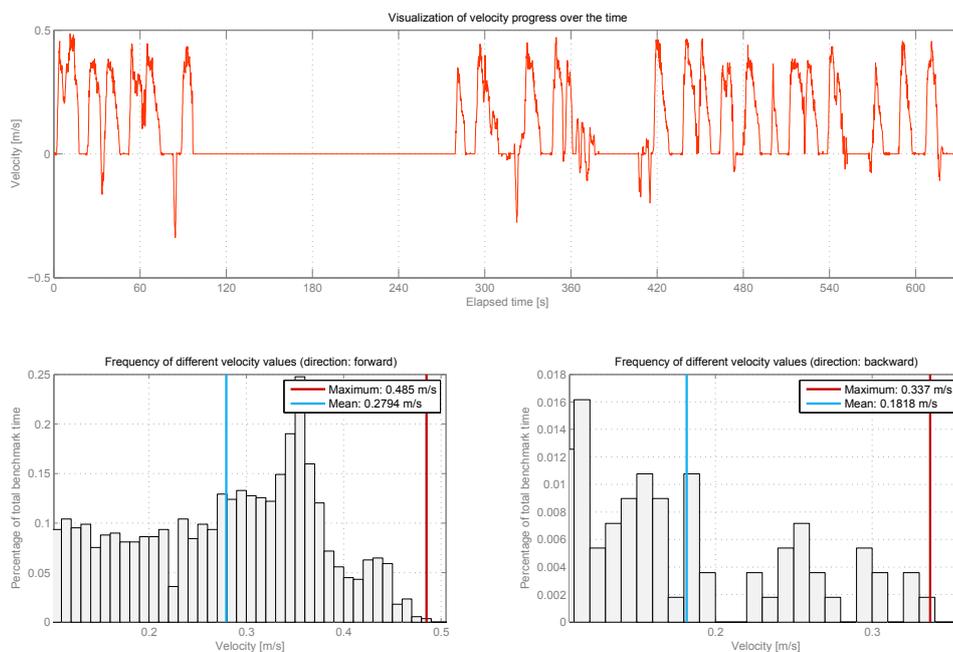


Abbildung 5.8: Visualisierung der Bewegungsgeschwindigkeit

Darstellung der gefahrenen Trajektorie im Verlauf des zehnminütigen Testlaufs und den gemessenen Abweichungen zu den vom Roboter vermuteten Posen.

Abbildungen 5.8 und 5.9 stellen beide Geschwindigkeitsanalysen dar, wobei die eine die Fortbewegungsgeschwindigkeit (Abbildung 5.8) und die andere die Rotationsge-

schwindigkeit (Abbildung 5.9) ausgibt. Erneut wird zwischen einer oberen und einer unteren Teilgrafik unterschieden. Die obere stellt den gesamten Geschwindigkeitsverlauf während des Benchmarks dar.

Man sieht dabei in Abbildung 5.8 sehr gut, dass es einige Situationen gab, in denen der Roboter kurzzeitig rückwärts fahren musste (negative Geschwindigkeit). Darüber hinaus stechen auch sofort eine fast dreiminütige sowie eine etwa halbminütige Stillstandsphase (keine Geschwindigkeit) ins Auge. In der langen Stillstandsphase kam es zu Timeouts. Der Roboter hat sich festgefahren und konnte sich nicht von selbst wieder „befreien“ und musste somit nach einigen Versuchen manuell in eine zur Fortführung des Benchmarks geeignete Position gebracht werden.

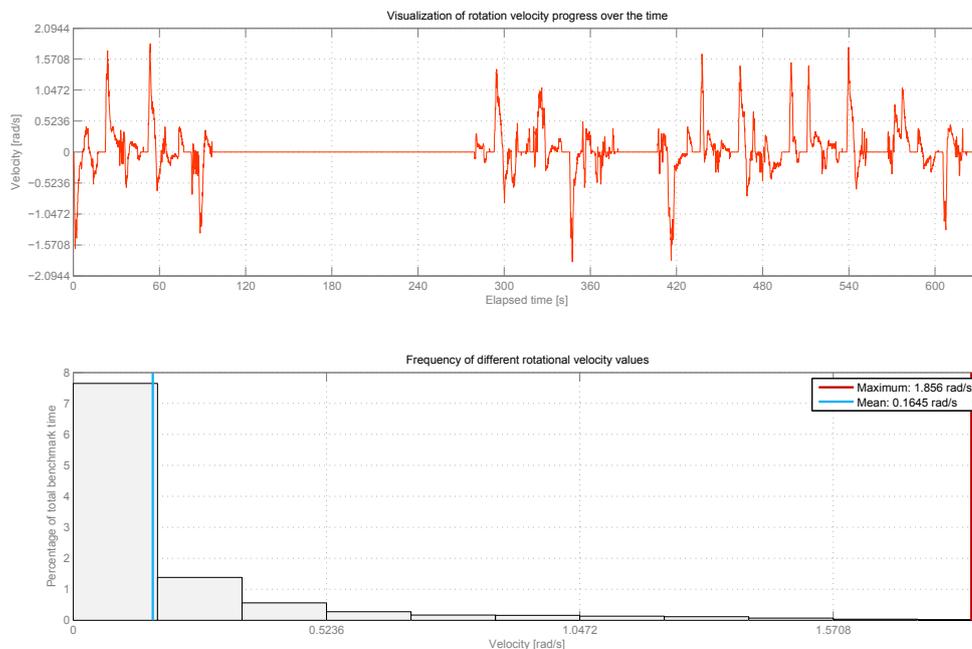


Abbildung 5.9: Visualisierung der Rotationsgeschwindigkeit

Darstellung der gefahrenen Trajektorie im Verlauf des zehnminütigen Testlaufs und den gemessenen Abweichungen zu den vom Roboter vermuteten Posen.

Im unteren Teil der Abbildungen finden sich erneut Histogramme. Das linke Histogramm stellt dabei die normierten Vorwärtsgeschwindigkeitsanteile dar, während das

rechte Histogramm sämtliche gemessene Rückwärtsgeschwindigkeiten darstellt. Logischerweise und wie auch schon im oberen Teil der Abbildung zu sehen ist, sind die Anteile der Rückwärtsgeschwindigkeiten wesentlich geringer. Ebenfalls dargestellt in beiden Histogrammen sind wieder maximale und mittlere Geschwindigkeit.

Abbildung 5.9 hingegen besitzt nur ein Histogramm. Es wird hierbei nicht zwischen einer Links- und Rechtsdrehung unterschieden sondern der absolute Rotationsgeschwindigkeitswert in Relation zu deren Häufigkeit dargestellt. Zwei Linien dienen wieder als Indikator für die maximale und mittlere Rotationsgeschwindigkeit. Man sieht, dass der Roboter sich hierbei meist mit weniger als $10^\circ/s$ nach links oder rechts gedreht hat.

5.3.4 Evaluation der Zielanfahrten

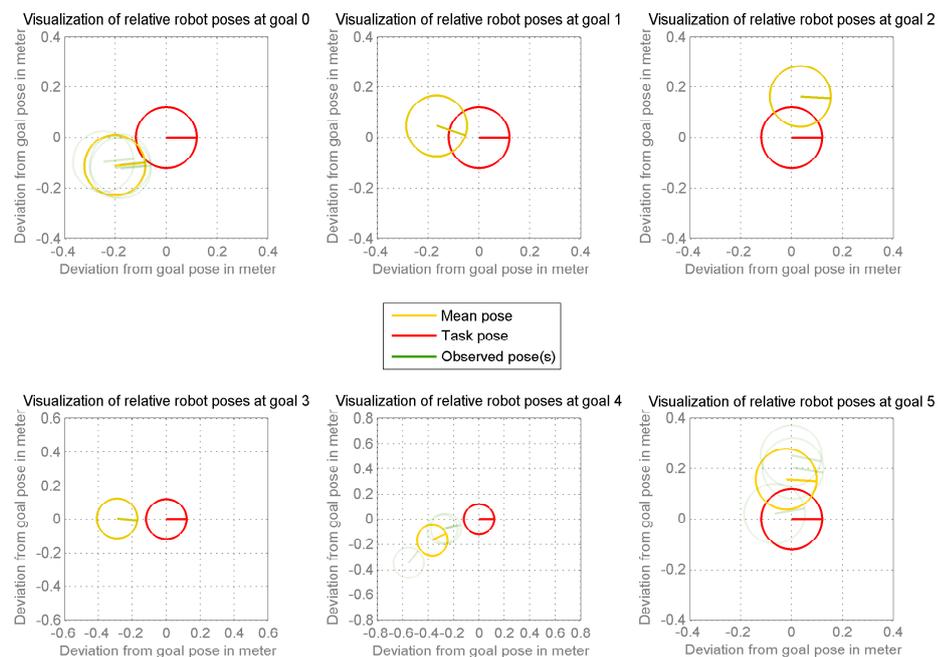


Abbildung 5.10: Visualisierung der Zielpositionen

Darstellung der angefahrenen Zielpositionen und der unterschiedlichen effektiven Ruhepositionen.

Da der Roboter während des Benchmarks stetig neue Aufgaben erhält, bestimmte Zielpunkte anzufahren, bietet es sich an, auch zu betrachten, ob er diese am Ende erreicht. Hierbei unterteilt sich Abbildung 5.10 in sechs Teilabbildungen, eine für jeden beim Benchmark angegebenen Zielpunkt. Dabei steht der rote Kreis beispielhaft für die festgelegte Zielposition, während die grünen Kreise für die effektiv erreichten Endpositionen steht. Diese sind relativ zur wirklichen Zielposition eingezeichnet. Darüber hinaus existiert auch jeweils ein weiterer gelber Kreis für die mittlere Pose an jeder der sechs Zielpositionen. Da der Roboter eine gewisse Toleranz beim Anfahren der Ziele besitzt, lässt sich hiermit gut visualisieren, wie stark er diese auch ausnutzt. Die in der Abbildung verwendeten Nummern entsprechen dabei den im Tool angegebenen Zielpunkten, wobei an dieser Stelle jedoch bei Null, und nicht wie üblich bei Eins, angefangen wird zu zählen.

Aufgrund der verhältnismäßig kurzen Benchmarkdauer und der zufälligen Anfahrt der einzelnen Zielpunkte, wurden nicht alle Ziele gleich oft angefahren.

5.3.5 Abstand zu Hindernissen

Das letzte im Testlauf betrachtete Bewertungsmaß ist der Abstand vom Roboter zu Hindernissen. Wie bereits in 4.1.2 erläutert, handelt es sich bei den vier Werten in Tabelle 5.11 um die Auswertung der zu jedem Zeitpunkt während des Benchmarks kleinsten gemessenen Entfernung zu einem Hindernis. Dies geschieht durch Verwendung der vom Roboter aufgezeichneten Range Scans, welche durch den intern verbauten Laserentfernungsmesser bestimmt werden und sich aus zahlreichen Entfernungswerten im Messbereich des Lasers zusammensetzen.

Minimum	Maximum	Mittelwert	Standardabweichung
0.0096m	0.8443m	0.1377m	0.1575m

Abbildung 5.11: Ausgabe der Range Scan Auswertung

Das bedeutet, dass der Roboter beispielsweise im Mittel etwa 13,8cm Abstand zu Objekten in seiner Umgebung eingehalten hat. Interessant ist an dieser Stelle der Wert

für das Minimum, denn dieser liegt bei unter einem Zentimeter. Dies sorgte dafür, dass sich der Roboter während des Benchmark festfuhr und sich aus dieser Situation nicht mehr befreien konnte.

Wie in Kapitel 2.2.1 beschrieben, kann man diese Werte als Aussage hinsichtlich der Sicherheit der vom Roboter gefahrenen Trajektorie ansehen. So gab es zumindest laut der Entfernungsrechnung keine Kollision mit umliegenden Gegenständen, dennoch hat er einige Gegenstände mit einem sehr geringen Abstand passiert. Da keine Menschen im Wohnumfeld anwesend waren und Menschen an sich auch ein Verhalten aufweisen, bei dem sie dem Roboter wohl ausgewichen wären, kann man keine Aussage über die Akzeptanz des Verhaltens des Roboters treffen. Prinzipiell ist anzunehmen, dass eine Person sich nicht sehr wohl fühlen würde, wenn ein Roboter sie so knapp passiert, allerdings lässt sich aus diesem Testlauf, wie bereits erwähnt, keine wirkliche Aussage dazu treffen. Bei festen Objekten der Umwelt ist dies natürlich nicht so kritisch zu beurteilen.

5.4 Probleme

Was in der Theorie einfach klingt, erweist sich in der Praxis oftmals als komplizierter als es der anfängliche Anschein erwarten lässt.

So ergaben sich auch eine Vielzahl an Problemen und Schwierigkeiten bei der Umsetzung der Benchmarktests, wie sie in den letzten Wochen des praktischen Teils dieser Arbeit durchgeführt wurden. Auf den folgenden Seiten sollen diese Probleme kurz vorgestellt und anschließend zusammengefasst werden.

5.4.1 Hutaufsatz

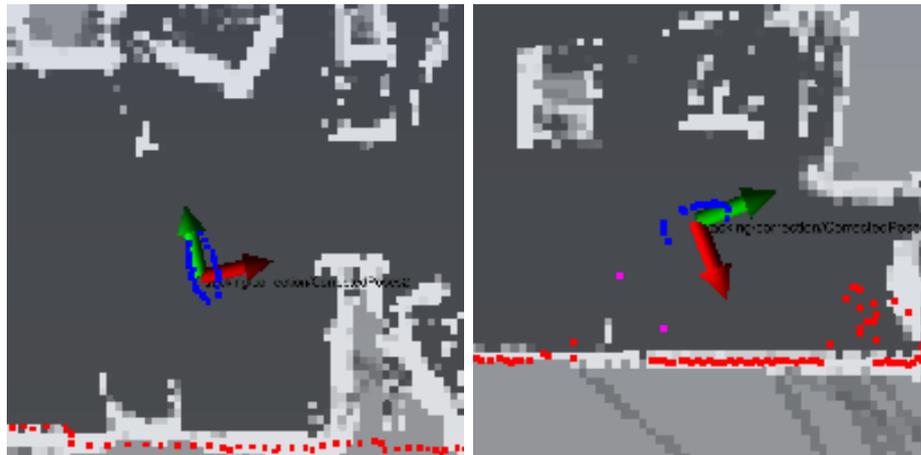
Wie bereits in Kapitel 5.2.2 angesprochen, gab es bereits die ersten Schwierigkeiten mit den Roboterhüten. Der erste entwickelte Prototyp aus einem Stück Schaumstoff erwies sich in seiner Konstruktion, trotz des verhältnismäßig leichten Materials, als zu schwer für den Einsatz auf dem Roboter. Es gab zwar keine Beeinflussung des Roboters durch diesen Aufbau, doch aufgrund seiner Größe neigte das Konstrukt

zu starken Schwingungen. Dies zeigte sich insbesondere in Situationen wie dem starken Beschleunigen, zum Beispiel beim Losfahren, und Abbremsen, sowie dem Fahren über Unebenheiten im Boden, wie sie beispielsweise durch Teppichläufer entstehen. Das schwingende Verhalten sorgte dabei für Messungenauigkeiten bei der Bestimmung der Position und irritierte auch in Einzelfällen die Software, welche aus den Messungen der Position eine Orientierung bestimmte. Dabei kam es zu einer falschen Orientierungsannahme, die eine um 180° verdrehte Orientierung des Roboters zur Folge hatte.

Aufgrund der genannten Schwierigkeiten und Probleme wurde anschließend ein weiterer Prototyp entwickelt. Dieser zeichnete sich durch kleinere Gesamtmaße und leichtere Materialien, wie Papier und Karton, aus. Des Weiteren wurde auch die Form des Hutes überarbeitet, um so eine bessere und stabilere Verfolgbarkeit des Roboters durch die Tracker zu erreichen.

5.4.2 Lasertracker

Eine Grundvoraussetzung für diese Arbeit war das Vorhandensein eines funktionstüchtigen und einsatzbereiten Trackersystems, welches als „Ground Truth“-Quelle fungieren sollte. Bis auf das Ausstatten des Roboters mit einem geeigneten Aufsatz für das Tracking, sollte keinerlei weitere Arbeit in das Trackersystem investiert werden. In Wirklichkeit jedoch lag zu Beginn der Arbeit eine nicht ausgereifte Software vor, sowie Probleme mit der Hardware. Obwohl in Kapitel 5.2.1 der Aufbau von zwei Trackereinheiten vorgestellt wurde, fanden die ersten Versuche mit drei solcher Einheiten statt. Während den ersten richtigen Tests stellte sich jedoch heraus, dass eine Einheit defekt zu sein schien, da die Entfernungsdaten an einigen Stellen aus unerklärlichen Gründen nicht den wirklichen Entfernungen entsprachen. Diese Tatsache war der Grund dafür, die Durchführung der Benchmarktests lediglich mit den zwei übrigen Einheiten auszuführen und den Aufbau auf ihre Umsetzbarkeit hin zu untersuchen. Es stellte sich heraus, dass zwei Tracker gerade so ausreichend für das Tracken der Roboterposition waren. Allerdings entstanden durch das Entfallen eines Trackers neue Probleme, wie in



(a) Gut zu erkennen ist die ausgeprägte Ellipsenform der blauen Punkte. Von allen Seiten ist der Aufsatz von Messpunkten umgeben.
 (b) Die Ellipse ist nur teilweise erkennbar. Auf der unteren Seite befinden sich fast keine Messpunkte.

Abbildung 5.12: Problematik der Abdeckungen durch Messpunkte

Blaue Punkte stellen die Silhouette des Roboters dar.

Rote Punkte identifizieren den Hintergrund, also nicht bewegliche Objekte im Raum.

Die Blickrichtung des Roboters wird dargestellt durch den roten Pfeil.

Abbildung 5.12 zu sehen ist. Aufgrund des weggefallenen Trackers entstanden nun neue Zonen, in denen der Roboter nicht komplett von den Trackereinheiten erfasst werden konnte. Es konnte zwar detektiert werden, dass sich der Roboter an einer bestimmten Position befand, jedoch war aufgrund der teilweise fehlenden Messpunkte, wie in Abbildung 5.12 (b) dargestellt, in vielen Fällen keine stabile Aussage über die Orientierung des Roboters möglich. An solchen Stellen tendierte die Orientierungsschätzung der Software zu starken Schwankungen, sodass die zur Visualisierung angezeigten Orientierungspfeile starke Sprünge vollzogen. Dies deutete auf eine instabile Schätzung der Orientierung hin, die aufgrund einer unzureichenden Menge an Messpunkten, die den Roboter erfassten, keine exakte Orientierungsschätzung erlaubte. In einigen Fällen passierte es jedoch, dass der Roboter beim Losfahren respektive Weiterfahren einer um etwa 180° verdrehten Orientierungsschätzung durch die Trackersoftware unterlag. Als

Konsequenz daraus folgte, dass bei der weiteren Fahrt durch die Wohnung die Ausrichtung des Roboters falsch angenommen wurde und somit die Ergebnisse verfälscht wurden. Diese eben genannten Zonen befanden sich dabei in den Eckbereichen des Wohnlabors, in denen sich keine Tracker befanden. Hier war es für die Messungen nur möglich, den Roboter von der zum Raum zugewandten Seite zu detektieren.

Zusammenfassend ist an dieser Stelle zu sagen, dass durch die nicht eingehaltenen Voraussetzungen für diese Arbeit ein großer Mehraufwand für Fehlersuchen, Tests und Verbesserungen entstanden ist, der im Vorfeld nicht eingeplant wurde und viel Zeit gekostet hat.

5.4.3 Fazit

Hutaufsatz

Bezüglich des Hutaufsatzes für den Roboter haben sich bereits einige Eigenschaften herauskristallisiert, die für einen optimalen Roboteraufsatz als Hut von Nöten sind. Dabei bleibt festzuhalten, dass eine möglichst leichte aber stabile Konstruktion benötigt wird, die im Idealfall möglichst wenig anfällig auf Schwankungen sein sollte. Diese sollte weiterhin eine für den Tracker gut wahrnehmbare Form besitzen, beispielsweise wie die ellipsenförmige Querschnittsform des zweiten Prototypen.

Lasertracker

Für die Umsetzung der Benchmarks mit zwei Trackern kann Folgendes festgestellt werden: Prinzipiell ist eine Umsetzung mit zwei entsprechenden Einheiten möglich. Um jedoch eine möglichst stabile Posenschätzung des Roboters in allen Bereichen der Räumlichkeit zu erhalten, wird für jede Ecke des Raumes eine Trackereinheit benötigt. Erst dann wird es möglich, den Roboter in allen Bereichen des Raumes zu detektieren und eine stabile Schätzung der Position und Orientierung zu erhalten.

Weiterhin kann es sich als schwierig erweisen, wenn die Tracker nicht ausreichend genau mit der im Roboter befindlichen Karte abgeglichen wurde. Dabei würde sich ein

konstanter Offsetwert zwischen der wirklichen und der gemessenen Pose ergeben. Ein einfaches „Herausrechnen“ ist jedoch nicht ohne Weiteres möglich. Falls der Roboter beispielsweise selbst einen konstanten Fehler bei der Positionsbestimmung besitzt, so würde dieser dann nicht auffallen, da er sich mit dem der Tracker überlagern würde.

5.5 Zusammenfassung

Nachdem in diesem Kapitel sowohl die Umgebung vorgestellt, sowie ein wenig auf einen beispielhaften Testlauf eines Benchmarks eingegangen und erläutert wurde, folgte auch ein Darstellung der Schwierigkeiten und Probleme, die sich im Zusammenhang mit der Arbeit ergaben. Einige Schwierigkeiten und Probleme bleiben zwingend bestehen. Es bleibt lediglich die Aufgabe, die Auswirkungen dieser auf ein möglichst niedriges Niveau zu bringen und somit den Einfluss dieser fast völlig auszumerzen.

Kapitel 6

Zusammenfassung und Ausblick

6.1 Zusammenfassung

Abschließend soll ein Blick auf die Ziele der Arbeit und deren Umsetzung geworfen werden. Das erste Ziel dieser Arbeit, welches der Entwicklung eines Softwaretools zum Automatisieren von Navigationsbenchmarks entspricht, wurde erreicht. Die Wiederverwendbarkeit der entwickelten Software wurde von Anfang an bei der Entwicklung der Software so weit wie möglich verfolgt. Aufgrund der Heterogenität der Fähigkeiten, die ein Roboter besitzen kann, gestaltet es sich jedoch als sehr schwer, eine gemeinsame Grundlage für weitere Benchmarktools zu schaffen. Das erstellte Tool lässt sich jedoch hinsichtlich dem Benchmarking der Navigation einfach durch weitere aufzuzeichnende Daten und deren Evaluationsmöglichkeit erweitern. Möchte man das Tool verwenden, um von der Navigation unabhängige Benchmarks umzusetzen, so kann die Software lediglich hinsichtlich der grafischen Oberfläche und der Aufzeichnung entsprechender Daten als Vorlage verwendet werden.

Wirft man einen Blick auf die in dieser Arbeit implementierten Bewertungsmaße, wie sie in Abschnitt 5.3 vorgestellt wurden, so kann der Umfang noch um weiterer Metriken vergrößert werden. Dies kann auf relativ einfache Art und Weise geschehen und wird, im Vergleich zur bereits geleisteten Vorarbeit, mit geringerem Aufwand umsetzbar sein. Hierfür ist es eventuell lediglich nötig die Aufzeichnung der Daten um

entsprechende Channels zu erweitern. Außerdem müssen eine entsprechende Export-Funktion geschrieben und passende Matlab-Skripte für die Darstellung erstellt werden. Weitere Beispiele für Bewertungsmaße finden sich in [Mob, 2010] respektive [GOLDA, 2014]. Aufgrund des zeitlichen Rahmens für diese Arbeit wurde jedoch auf die Implementierung weitere Bewertungsmaße verzichtet und der Fokus in erster Linie auf die Bewertung der Lokalisation und teilweise auch auf die Navigation beschränkt.

6.2 Weiterführende Arbeiten

Mit dieser Arbeit wurde ein erster Versuch gestartet, das Navigationsverhalten eines Roboters zu untersuchen und dieses somit vergleichbar zu machen. So gut eine künstliche Wohnumgebung auch eine Reale simuliert, beschränkt sich diese lediglich auf die Einrichtung einer realen Wohnung. Mehr Realitätsnähe erreicht man erst durch Hinzunahme von Menschen (und eventuell anderen Lebewesen) in die Wohnumgebung. Dies fordert den Roboter mehr und macht Benchmarks auch entsprechend anspruchsvoller, sowohl für den Roboter, als auch für den Menschen, der sie umsetzt und plant. Menschen in das Wohnfeld mit aufzunehmen, bedeutet jedoch auch einen erhöhten Aufwand für die Benchmarktechnik. An dieser Stelle kann man ansetzen und versuchen, die Idee des Benchmarkings um den Faktor „Mensch“ zu erweitern. Einige Fragen, die dazu untersucht werden müssten, könnten unter anderem sein:

- Was ist typisches Verhalten eines Menschen, auf welches der Roboter reagieren muss?
- Wie lässt sich „Ground Truth“ Material für den Roboter bestimmen, ohne dass es zu Verwechslungen mit dem Roboter kommt?

Diese Fragen können als Ausgangspunkt hergenommen und durch weitere Fragen ergänzt werden.

Wie immer mehr aufkommende Robotik-Wettbewerbe wie das bereits seit einigen Jahren etablierte RoboCup@Home oder der Newcomer RoCKIn@Home beweisen, ist auch der Wunsch danach, Roboter in ihrer Leistungsfähigkeit zu vergleichen, vorhanden und wird wohl auch in den kommenden Jahren weiter vorangetrieben.

Anhang A

Ergänzende Unterlagen

A.1 Anleitung zur Nutzung der Software

Das Package „Benchmarking“ stellt eine Konfiguration und einen View bereit, um einen Benchmark für Navigations- und Lokalisationssoftware durchzuführen, welche auf der Roboter-Middleware „mira“ basiert. Der Ablauf des Benchmarks besteht darin, dass der Roboter für eine vorab festgelegte Zeitspanne eine Menge von Zielpunkten in fester oder zufälliger Reihenfolge wiederholt anfährt, und währenddessen Daten aufzeichnet. Diese können anschließend mit ebenfalls bereitgestellten Matlab-Scripten ausgewertet und die Ergebnisse präsentiert werden (Details wurden in dieser Arbeit beschrieben). Die Konfiguration und Bedienung des Benchmark-Ablaufs über die GUI kann auf dem Roboter direkt oder auf einem (mittels des verteilten mira-Frameworks) damit verbundenen Rechner (im folgenden Benchmark-Rechner genannt) erfolgen. Da die Nutzung eines separaten Rechners in vielen Fällen die Handhabung vereinfachen sollte, wird diese Variante hier beschrieben.

Voraussetzungen

Um das externe Tracking des Roboters mittels Laserscanner-Netzwerk (Ground Truth Position) zu ermöglichen, muss dieses Netzwerk eingerichtet, mit dem Rechner verbunden und gestartet werden. Es bietet sich an, die Tracking-Software (Units) im selben lokalen mira-Framework laufen zu lassen, in dem das Benchmarking (auf dem

Benchmark-Rechner) gesteuert wird. Je nach Roboter ist möglicherweise ein entsprechender Tracking-Target-Aufsatz („Hut“) nötig, der vom externen Scanner-Netzwerk erfasst und getrackt werden kann. Für die Konfiguration und Initialisierung des Trackers siehe Dokumentation zum Paket „NewLRSTrackerUnit“.

Zunächst sollte das Robotersystem gestartet werden, das im Benchmark evaluiert werden soll. Dieses muss die entsprechenden Navigations- und Lokalisationsmodule enthalten. Außerdem muss ein Port eingerichtet werden, welcher eine Verbindung von einem externen mira-Framework ermöglicht. Dies kann beispielsweise als Startparameter in der Kommandozeile angegeben werden:

```
miracenter robotconfig.xml -p 1234
```

(Unter Umständen ist eine entsprechende Konfiguration des Betriebssystems oder anderer Komponenten notwendig, damit dieser Port tatsächlich von außen erreichbar ist).

Praktischer Hinweis: Achten Sie drauf, dass kein Netzstecker am Roboter angeschlossen ist!

Nun kann auf dem Benchmarking-Rechner ebenfalls ein mira-Framework gestartet und mit dem Roboter verbunden werden. Hierbei ist die Benchmarking-Konfiguration (sowie eventuell die Konfiguration für das Tracker-Netzwerk) anzugeben. Die Verbindung zum Roboter-Framework kann ebenfalls direkt als Startparameter angegeben werden oder alternativ nach dem Start über die GUI erfolgen. Ein möglicher Aufruf könnte hierbei in Etwa wie folgt aussehen:

```
miracenter Benchmarking.xml LRSTrackerUnit.xml -k 141.24.24.140:1234
```

Hinweis: Für die Initialisierung des externen Trackers ist es normalerweise notwendig, dass der Roboter zunächst einige Meter vorwärts fährt, dies sollte möglichst vor dem eigentlichen Benchmark-Start erfolgen, um diesen mit einer korrekten Ground-Truth-Position zu beginnen.

Benchmark-Einrichtung und -Durchführung

Schritt 1: Starten Sie, falls noch nicht geschehen, auf dem externen Rechner eine Instanz des „miracenter“. Öffnen Sie anschließend die View-Auswahl (Strg + D) und wählen Sie in der Kategorie „Benchmarking“ den Punkt „Navigation Benchmark“ aus.

Schritt 2a: Wählen Sie im Tab „Channels & Services“ die nötigen Channels und Services aus. Sollte ein entsprechender Channel oder Service nicht verfügbar sein, so überprüfen Sie bitte, ob alle Voraussetzungen erfüllt sind.

Schritt 2b: Alternativ können Sie auch den „Load“-Button (*Shortcut*: Alt + L) nutzen, um bereits früher exportierte Einstellungen erneut zu laden. Überprüfen Sie nach dem Import lediglich ob alle Einstellungen in den jeweiligen Tabs Ihren Wünschen entsprechen. Fahren Sie anschließend mit *Schritt 5* fort.

Schritt 3: Wechseln Sie zum Tab „Waypoints“ und fügen Sie entweder manuell einzelne Waypoints hinzu oder importieren Sie eine bereits vorhandene Liste aus einer XML-Datei. Achten Sie beim Einfügen von Waypoints darauf, eindeutige Namen zu vergeben. Diese sind nicht zwingend notwendig, vereinfachen jedoch das Verständnis der eventuell auftretenden Fehlermeldungen während des Benchmarks. Des Weiteren können Sie dann - ausgehend von der im Roboter vorhandenen Karte - die x- und y-Koordinaten der Zielpunkte angeben. Der Winkel an dieser Position muss hierbei in Grad angegeben werden! Legen Sie abschließend noch fest, ob die Waypoints in der Reihenfolge angefahren werden sollen, wie sie in der Liste dargestellt werden oder in zufälliger Reihenfolge.

Schritt 4: Rufen Sie den dritten und letzten Tab auf und stellen die gewünschte Dauer für den Benchmark ein.

Schritt 5: Bevor Sie den Vorgang nun starten, stellen Sie sicher, dass sich niemand in der Wohnumgebung befindet. Starten Sie den Benchmark über den „Start“-Button. Während der Laufzeit des Benchmarks erhalten Sie im unten befindlichen Textfeld

stetig Informationen, wie beispielsweise das nächste vom Roboter angefahrne Ziel, Hinweise oder Fehlermeldungen vom Pilot. Sollte es zu Fehlern kommen, so versucht der Roboter diese zuerst selbst zu lösen. Nach einer festgelegten Anzahl an Versuchen wird der Vorgang pausiert. Sie haben nun die Möglichkeit, die Wohnumgebung zu betreten und das Problem zu beseitigen. Anschließend setzen Sie den Benchmark-Vorgang fort, indem Sie erneut auf den „Start“-Button klicken.

Hinweise:

- Über den „Pause“-Button können Sie den Benchmark für einen Moment pausieren. Der „Stop“-Button bricht den Benchmark vorzeitig ab und leitet die finale Auswertung der bisher aufgezeichneten Daten ein.
- Das Betreten der Wohnumgebung während des Benchmarks ist nicht vorgesehen. Aus diesem Grund sollte dieser auch nicht betreten werden, es sei denn der Benchmark ist pausiert oder angehalten.

Schritt 6: Nach Ablauf der Zeit oder wenn Sie den Benchmark manuell abbrechen, werden die aufgezeichneten Daten vom Tool verarbeitet. Dies dauert, je nach Performanz des Rechners und Länge des Benchmarks, mehrere Minuten. Sobald diese Auswertung beendet wurde, öffnet sich ein Benutzerdialog, in dem Sie nach einem Verzeichnis gefragt werden, in welches die ausgewerteten Daten - und das aufgezeichnete Tape - gespeichert werden sollen. Wenn Sie dies nicht tun möchten, so verbleiben alle Daten im Temp-Ordner Ihres Betriebssystems.

Schritt 7: Kopieren Sie aus dem Benchmarking-Package-Verzeichnis den Ordner „matlab“ und fügen ihn an einer beliebigen geeigneten Stelle ein. Kopieren Sie aus dem im vorherigen Schritt gespeicherten (oder ggf. aus dem temporären) Verzeichnis alle txt-Dateien in den Unterordner „data“.

Schritt 8: Führen Sie anschließend das Matlab-Skript „start_evaluation.m“ aus. Dieser Vorgang dauert je nach Leistung des verwendeten Rechners einige Sekunden. Ihnen werden nun einige Fenster mit der Auswertung der Daten und dazugehörigen Legenden dargestellt. Der Vorgang ist damit beendet.

Anhang B

Quellenangaben

WEBRESSOURCEN				
EvAAL	*	Evaluating AAL Systems through Competitive Benchmarking <i>http://evaal.aaloo.org/</i>	*	05.08.2014
Wikipedia	*	Artikel: Ambient Assisted Living <i>http://de.wikipedia.org/wiki/Ambient_Assisted_Living</i>	*	28.08.2014
Wikipedia	*	Artikel: Benchmark <i>http://de.wikipedia.org/wiki/Benchmark</i>	*	05.08.2014
Wikipedia	*	Artikel: Metrischer Raum <i>http://de.wikipedia.org/wiki/Metrischer_Raum</i>	*	05.08.2014

Abbildungsverzeichnis

1.1	Roboter „Tweety“	1
2.1	Bestandteile der Navigation	7
2.2	Beispieldaten für Sicherheitsmetrikbestimmung	8
2.3	Abweichung der gefahrenen zur geschätzten Trajektorie des Roboters	11
4.1	Öffnen des Benchmarkdialogs	17
4.2	Channels and Services	19
4.3	Waypoints	20
4.4	Benchmark control	21
4.5	Übersicht des Outputs der Evaluationskripte	23
5.1	Wohnlabor im Fachgebiet Neuroinformatik und Kognitive Robotik	27
5.2	Foto eines Lasertrackers	28
5.3	Übersicht des Trackeraufbaus	29
5.4	Aufbauten für den Roboter	30
5.5	Visualisierung der Roboterposen und Positionsfehler	32
5.6	Visualisierung der Roboterposen und Orientierungsfehler	33
5.7	Visualisierung der Pfadplanungsdauern	34
5.8	Visualisierung der Bewegungsgeschwindigkeit	35
5.9	Visualisierung der Rotationsgeschwindigkeit	36
5.10	Visualisierung der Zielpositionen	37
5.11	Ausgabe der Range Scan Auswertung	38
5.12	Problematik der Abdeckungen durch Messpunkte	41

Literaturverzeichnis

[Mob, 2010] (2010). *Mobile Robots Navigation*, Kap. Quantitative Performance Metrics for Mobile Robots Navigation. InTech.

[GOLDA, 2014] GOLDA, THOMAS (2014). *Bewertungsmaße und Benchmarks zur Evaluation von Navigationsleistungen eines mobilen Roboters*. Hauptseminar im Fachgebiet Neuroinformatik und Kognitive Robotik.
