



## Masterarbeit

# Lighthouse Keeper

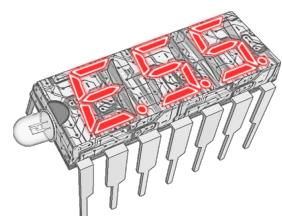
## Ein neues Verfahren zur Planung und Evaluation von Beacon-Konfigurationen

von

André Alexander Pieper  
Geb. 28.03.1988 in Berlin  
Matrikelnummer: 184960

04. Mai 2015

Erstprüfer: Jun.-Prof. Dr.-Ing. Sebastian Zug  
Otto-von-Guericke-Universität  
Fakultät für Informatik  
Institut für Verteilte Systeme  
Lehrstuhl Embedded Smart Systems  
Universitätsplatz 2, D-39106, Magdeburg



Zweitprüfer: Prof. Dr.-Ing. Abbas Omar  
Otto-von-Guericke-Universität  
Fakultät für Elektro- und Informationstechnik  
Institut für Informations- und Kommunikationstechnik  
Lehrstuhl für Hochfrequenz- und Kommunikationstechnik  
Universitätsplatz 2, D-39106, Magdeburg





# Kurzdarstellung

Satellitengestützte Navigationssysteme sind in der heutigen Zeit ein fester Bestandteil des alltäglichen Lebens. Sie ermöglichen zum Beispiel die Orientierung eines Autofahrers auf ihm unbekannten Straßen und sind mittlerweile serienmäßig in Autos, Smartphones und sogar Kameras integriert. Aber die Signale der satellitengestützten Lokalisierungssysteme wie dem „Global Positioning System“ (GPS), „Globalnaja nawigazionnaja sputnikowaja sistema“ (GLONASS) und „Galileo“ verlieren sich in Gebäuden, da sie von deren Materialien absorbiert bzw. reflektiert werden. So wie die „Outdoor“-Lokalisierungssysteme den Straßenverkehr revolutionierten, soll ein neues „Indoor“-System die Fortbewegung von Menschen in Gebäuden neu erfinden: die „Beacons“ oder zu Deutsch „Leuchtfeuer“. Obwohl bereits viele Unternehmen diese Technik herstellen und vermarkten, existiert im Moment noch kein zufriedenstellendes Konzept, das eine Möglichkeit bietet die nötige Infrastruktur für ein Lokalisierungssystem basierend auf Beacons effizient zu planen und zu testen. Um das Potential dieser aufstrebenden Technologie auszuschöpfen, liegt der Forschungsschwerpunkt der folgenden Ausführungen auf der Entwicklung eines dafür möglichen Verfahrens und dessen Automation mithilfe von Robotern. In Anspielung auf die Bedeutung der Beacons wird das Gesamtkonzept als „Lighthouse Keeper“ benannt – frei übersetzt „Herr der Leuchtfeuer“.



# Danksagung

Die vorliegende Arbeit entstand am Lehrstuhl für Embedded Smart Systems (ESS) der Otto-von-Guericke-Universität Magdeburg.

Einen besonderen Dank möchte ich an dieser Stelle meinem Betreuer, Jun.-Prof. Dr.-Ing. Sebastian Zug, für seine Unterstützung, Förderung und dem mir entgegengebrachten Vertrauen im Laufe meiner Arbeit aussprechen. Er stellte mir ein sehr gutes Arbeitsumfeld zur Verfügung, in dem ich ungehindert forschen und mich auf mein Thema konzentrieren konnte. Vielen Dank auch an meinen Zweitprüfer, Prof. Dr.-Ing. Abbas Omar, und seinem Doktoranden, Abdo Nasser Ali Gaber, für den Einblick in ihre Forschungen und den damit verbunden Inspirationen für mein Thema.

Ich möchte auch das Stipendium mit finanzieller Unterstützung der Stiftung Industrieforschung nicht vergessen, das mir mehr Unabhängigkeit und Freiheit in meiner Arbeit ermöglichte. Vielen Dank dafür.

Viele Menschen haben einen signifikanten Beitrag zu dieser Arbeit geleistet. Als erstes möchte ich Christoph Steup nennen, der sich mit mir den Arbeitsplatz teilte und mir in anregenden Diskussionen, ob professionell oder persönlich, neuen Antrieb schenkte. Speziellen Dank auch an Dirk Steindorf, der seine Expertise im Umgang mit dem Scitos G5 Roboter mit mir teilte und mich in der Evaluationsphase tatkräftig unterstützte.

Meinen langjährigen Kommilitonen und Freunden möchte ich für die Unterstützung im Studium danken. Allen voran Andreas Himmel und Hannes Heinemann, deren Zukunft ihnen hoffentlich nur Erfolg verspricht. Mit beiden habe ich viele Höhen und Tiefen erlebt und gemeinsam haben wir diese gemeistert. Ich möchte auch Julian Scholle nicht vergessen, der zwischenzeitlich neue Wege ging und mich in die Welt der Informatik mitnahm.

Zuletzt möchte ich mich herzlich bei meinen Eltern und meiner Partnerin Carolin Richter für ihre Unterstützung, ihre Geduld und ihren Verständnis bedanken.

André Pieper  
Magdeburg, Mai 2015



# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>VII</b>
<b>Tabellenverzeichnis</b>	<b>IX</b>
<b>Abkürzungsverzeichnis</b>	<b>XI</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Stand der Technik – Einführung in die Beacon-Technologie</b>	<b>5</b>
2.1 Entwicklungsgeschichte . . . . .	5
2.2 Aufbau und Funktionsweise von Beacons . . . . .	6
2.3 Ortungsmethoden . . . . .	8
2.3.1 Definitionen verschiedener Anwendungsfelder . . . . .	9
2.3.2 Mikro-Lokalisierung . . . . .	10
2.4 Anwendungsbereiche der Indoor-Lokalisierung . . . . .	12
2.4.1 Mögliche Einsatzszenarien . . . . .	13
2.4.2 Planungskonzepte für Beacon-Konfigurationen . . . . .	15
<b>3 Konzeptplanung</b>	<b>17</b>
3.1 Vorüberlegungen . . . . .	17
3.2 Zielsetzungen . . . . .	18
3.2.1 Hardware-Anforderungen . . . . .	19
3.2.2 Konzept der Software-Architektur . . . . .	19
<b>4 Software-Entwicklung</b>	<b>23</b>
4.1 Verwendete Hardware . . . . .	23
4.1.1 Motorola Moto G . . . . .	23
4.1.2 Youbot . . . . .	25
4.1.3 Scitos G5 . . . . .	25
4.2 Verwendete Software . . . . .	26
4.2.1 Robot Operating System – ROS . . . . .	27
4.2.2 Miracenter . . . . .	28
4.2.3 Android-Studio, Estimote SDK und ROSjava . . . . .	29
4.3 App-Entwicklung . . . . .	30
4.3.1 Beacon-Detektierung . . . . .	30

4.3.2 Lagemessung . . . . .	30
4.3.3 ROS-Anbindung . . . . .	32
4.3.4 Programmablauf . . . . .	33
<b>5 Experimente</b>	<b>37</b>
5.1 Messung der BLE-Signalausbreitung . . . . .	37
5.2 Evaluierung der SDK-eigenen Distanzschätzung . . . . .	42
5.3 Auswirkung der Smartphone-Orientierung . . . . .	42
5.4 Wechselwirkungen zwischen Funkquellen . . . . .	44
5.5 Energieverbrauch eines Beacons . . . . .	46
<b>6 Modellentwurf und Simulation</b>	<b>51</b>
6.1 Modellbildung . . . . .	51
6.1.1 Das WINNER II-Modell . . . . .	53
6.1.2 Parameterschätzung . . . . .	54
6.2 Vertrauenskala . . . . .	57
6.3 Entwicklung der Simulationsumgebung . . . . .	59
6.3.1 Grundrissanalyse . . . . .	60
6.3.2 Optimierung der Beacon-Infrastruktur . . . . .	62
6.3.3 Simulationsbeispiel . . . . .	67
<b>7 Evaluierung</b>	<b>69</b>
7.1 Dynamisches Experiment . . . . .	69
7.1.1 Durchführung . . . . .	69
7.1.2 Einsatz des Lighthouse Keepers . . . . .	70
7.1.3 Pfadplanung und Navigation . . . . .	71
7.2 Auswertung . . . . .	72
7.3 Diskussion . . . . .	76
<b>8 Zusammenfassung und Ausblick</b>	<b>77</b>
<b>Literaturverzeichnis</b>	<b>79</b>
<b>Anhang</b>	<b>85</b>
A Bilder . . . . .	85
<b>Selbstständigkeitserklärung</b>	<b>89</b>

# Abbildungsverzeichnis

2.1	Offizielles Bluetooth Smart Logo [10] . . . . .	6
2.2	Explosionszeichnung Beacon [13] . . . . .	6
2.3	Signalschwächung durch Objekte, z.B. Wände [7] . . . . .	8
2.4	Menschliche Körper blockieren zusätzlich die Signale [7] . . . . .	8
2.5	Beispiel einer Trilateration zur Positionsbestimmung . . . . .	11
2.6	Beacon-Nutzung in Geschäften [1] . . . . .	13
2.7	Beacon-Nutzung im Flughäfen [41] . . . . .	14
2.8	Demonstration der Estimote Indoor Localization App [14] . . . . .	15
3.1	Kreislauf einer Prozessplanung . . . . .	18
3.2	Überblick über die Software-Anforderungen . . . . .	21
3.3	Überblick über die Simulationsumgebung . . . . .	21
4.1	Vorderseite des Motorola Moto G [6] . . . . .	24
4.2	Rückseite vom Moto G mit Antennen-Gerüst [8] . . . . .	24
4.3	Youbot mit Halterung (gelber Aufsatz) . . . . .	25
4.4	Zeichnung eines Youbot-Arms und seiner fünf Rotationsachsen [24] . . . . .	25
4.5	Scitos G5 von der MetraLabs GmbH . . . . .	26
4.6	Erstellter Grundriss eines Flures in Gebäude 29 der OvGU, Stockwerk 3 .	28
4.7	Aufbau des Komplementärfilters, in Anlehnung an [33] . . . . .	32
4.8	Quellcode-Beispiel eines Message-Paketes für eine ROSjava-Implementierung . . . . .	33
4.9	Programmablaufplan der Lighthouse Keeper-Applikation für Android-Smartphones . . . . .	35
5.1	Verlauf der Signalstärke aus einer Entfernung von 1, 4 und 10 Metern .	38
5.2	Mittelwert der RSSI-Werte von allen Distanzen . . . . .	39
5.3	Mittelwert der 100 stärksten RSSI-Werte von allen Distanzen . . . . .	40
5.4	Top-View auf die Signalstärken-Verteilung zu den einzelnen Distanzen .	41
5.5	Seitliche Ansicht auf die Signalstärken-Verteilung zu den einzelnen Distanzen . . . . .	41
5.6	Vergleich der durchschnittlichen Distanz-Schätzung zur realen Entfernung	42
5.7	Skizzierung der Anordnung der Beacons und der Drehung des Roboterarms	43

5.8 Verlauf der gemessenen Signalstärke zu der Drehung des Smartphones in Z-Richtung . . . . .	44
5.9 Gleichzeitige Messung dreier naheliegender Beacon-Signale . . . . .	45
5.10 Verlauf der Signalstärke dreier nah nebeneinander liegender Beacons aus einer Entfernung von einem Meter . . . . .	46
5.11 Anteile eines Datenpakets in der Beacon-Kommunikation auf der MAC- Ebene . . . . .	47
5.12 Einfluss der einstellbaren Parameter auf die Lebensdauer einer CR2450- Batterie . . . . .	49
6.1 Qualitativer Verlauf des WINNER II-Modells . . . . .	54
6.2 Durchschnittlicher RSSI-Wert und berechneter RSSI-Wert im Vergleich . . . . .	56
6.3 Vergleich der Distanz-Schätzung mit dem WINNER II-Modell zur gemes- senen Entfernung . . . . .	57
6.4 Wahrscheinlichkeit des Empfangs gültiger Signale in einem Toleranzbe- reich von 0,5 Meter und dessen Approximation . . . . .	58
6.5 Wahrscheinlichkeit des Empfangs gültiger Signale in einem Toleranzbe- reich von 2 Metern und dessen Approximation . . . . .	59
6.6 Schematischer Ablauf der Grundriss-Analyse . . . . .	62
6.7 Beispiel einer Paretofront . . . . .	66
6.8 Beispiel einer Lighthouse Keeper-Simulation . . . . .	68
7.1 Grundriss des Testraumes . . . . .	70
7.2 Foto des Testraumes . . . . .	70
7.3 Bearbeiteter Grundriss . . . . .	70
7.4 Beacon-Konfiguration der dynamischen Experimente . . . . .	71
7.5 Drift des Scitos G5 . . . . .	72
7.6 Haltepunkte mit Drift . . . . .	73
7.7 Haltepunkte ohne Drift . . . . .	73
7.8 Beseitigung vom Drift . . . . .	73
7.9 Zeitlicher Verlauf der Signalstärke und des Abstandes für Beacon 1 . . . . .	74
7.10 Zeitlicher Verlauf der Signalstärke und des Abstandes für Beacon 2 . . . . .	75
7.11 Zeitlicher Verlauf der Signalstärke und des Abstandes für Beacon 3 . . . . .	75

# Tabellenverzeichnis

2.1	Beispiel der Informationsnutzung; in Anlehnung an: [7] . . . . .	9
2.2	Beispiel der Bereichsdefinition; in Anlehnung an: [7] . . . . .	10
5.1	Standardabweichung der Messungen aus Abb. 5.1 und aus Abb. 5.10 . .	46
6.1	Übersicht der Kanalmodelle im Indoor-Bereich, in Anlehnung an [4] . . .	52
6.2	Parameter des Modells . . . . .	56
6.3	Parameter der Vertrauensskala . . . . .	59



# Abkürzungsverzeichnis

<b>GPS</b>	Global Positioning System (globales Positionsbestimmungssystem)
<b>GLONASS</b>	Globalnaja nawigazionnaja sputnikowaja sistema (globales Satellitennavigationssystem)
<b>ULP</b>	Ultra Low Power Bluetooth (sehr leistungssparendes Bluetooth)
<b>BLE</b>	Bluetooth Low Energy (energiesparendes Bluetooth)
<b>WLAN</b>	Wireless Local Area Network (drahtloses lokales Netzwerk)
<b>RSSI</b>	Received Signal Strength Indication (Indikator für die Empfangsfeldstärke)
<b>UUID</b>	Universally Unique Identifier (Standard für Identifikatoren)
<b>ROS</b>	Robot Operating System (Software-Framework für Roboter)
<b>P2P</b>	Peer to Peer (Rechner-Rechner-Verbindung)
<b>PNG</b>	Portable Network Graphics (portable Netzwerkgrafik)
<b>XML</b>	Extensible Markup Language (erweiterbare Auszeichnungssprache)
<b>IDE</b>	Integrated Development Environment (integrierte Entwicklungsumgebung)
<b>SDK</b>	Software Development Kit (eine Sammlung von Werkzeugen und Anwendungen, um eine Software zu erstellen)
<b>IMU</b>	Inertial Measurement Unit (Inertialsensoren)
<b>IP</b>	Internetprotokoll
<b>dBm</b>	Dezibel Milliwatt
<b>WINNER</b>	Wireless-World-Initiative-New-Radio (ein Konsortium zur Verbesserung der Leistung von Mobilfunkkommunikationssystemen)
<b>LOS</b>	Line Of Sight (Sichtverbindung)
<b>NLOS</b>	Non Line Of Sight keine Sichtverbindung
<b>PSO</b>	Partikel-Swarm-Optimization (Partikel-Schwarm-Optimierung)
<b>GUI</b>	Graphical User Interface (grafische Benutzeroberfläche)



# 1 Einleitung

Herkömmliche Navigationssysteme für den „Outdoor“-Bereich haben viele Handlungen des alltäglichen Lebens weitgehend erleichtert. Die Tage gehören der Vergangenheit an, in denen Fahrtstrecken über unbekannte Straßen und durch fremde Länder mit analogen Karten weit im Voraus geplant werden mussten. Die Navigation bzw. Lokalisierung basiert dabei auf einer Positionsbestimmung mithilfe von Funksignalen, die von Satelliten in der Erdumlaufbahn kreisend ausgesendet werden. Aus diesen Signalen können die Navigationssysteme ihre Position auf der Erde berechnen und sie schließlich dem Nutzer zur Verfügung stellen. Die eigene Lokalisierung wäre nun auch für den Besucher in einem Gebäude hilfreich, damit er sich orientieren kann und beispielsweise in einem weitläufigen Kaufhaus schneller zu einem Geschäft seiner Wahl findet. Um die Vorteile einer Positionsbestimmung im „Indoor“-Bereich anzuwenden, können die bisherigen satellitengestützten Systeme (GPS, GLONASS, Galileo) jedoch noch nicht genutzt werden, da die massive Bauweise der Gebäude ihre Signale absorbiert bzw. reflektiert. Seit einigen Jahren arbeiten Forschungseinrichtungen und Unternehmen an einer Lösung für die Lokalisierung von Objekten im Inneren von Gebäuden. Die Forschungsprojekte verfolgen dabei viele verschiedene Ansätze und Technologien zur Erreichung dieses Ziels, wie beispielsweise die Projekte „EVARILOS“ [39], „Google Indoor Maps“ [18] und „Mobile Indoor Localization“ [28] zeigen. Jedoch existiert noch keine Lösung, die sich bereits durchgesetzt hat und einen kommerziellen Nutzen erzielt.

Eine neue und vielversprechende Variante zur innerräumlichen Lokalisierung stellt dabei die Entwicklung von Beacons (z. Dt. Leuchtfeuer) dar. Die Vorteile dieses Systems gegenüber bisherigen Lösungen sind zum einen die geringen Kosten sowie die hohe Flexibilität und Autonomie der einzelnen Elemente. Der größte Gewinn ist jedoch die plattformübergreifende Nutzbarkeit, denn mittlerweile unterstützen alle mobilen Betriebssysteme den verwendeten Bluetooth-Standard. Viele bisherigen Technologien zur Indoor-Lokalisierung benötigen hingegen spezielle Empfangsgeräte mit einem hohen Anschaffungspreis. Dies sind z.B. in den Boden eingelassene künstliche Magnetfelder [22], RFID-Transponder [2] oder funkbasierter Lösungen [42]. Diese Systeme sind zwar erprobt und erzielen eine hohe Lokalisierungsgenauigkeit, besitzen jedoch den Nachteil einer kostenintensiven und aufwendigen Infrastruktur, die meist auch eine bauliche Veränderung am Gebäude erfordert. Die Einfachheit der Anbringung der Beacons, die Kostenvorteile und die weitere Verbreitung des verwendeten Bluetooth-Protokolls auf

mobilen Geräten sorgen dabei für ein vielfältiges Einsatzspektrum. Der größte Vorteil der kleinen Sender gegenüber konkurrierender Technologien stellt die variable Verwendung der Beacons beispielsweise für einfache, rudimentäre Positionsbestimmungen bis hin zur hochpräzisen Ortung dar. Je nach Einsatzszenario können die Beacons problemlos angepasst und so für ihren Verwendungszweck effizienter gestaltet werden.

Nun wird für ein Lokalisierungssystem in Gebäuden ein Planungskonzept benötigt, das die Funkbaken optimal für ihre Aufgabe positioniert. Denn im Gegensatz zu den Satellitensystemen im Outdoor-Bereich, bewegen sich die Beacons nicht und somit können auch nicht deren Konzepte übernommen werden. Während eine Vielzahl von Methoden existieren, aus einer Signalstärke oder der Laufzeit eines Funksignals eine Distanz zu berechnen, wurde noch kein Verfahren zur Berechnung der dafür nötigen Infrastruktur vorgestellt. Somit mangelt es an Methoden, die Gebäude mit Beacon-Systemen zweifelsfrei auf die Qualität der Lokalisierung und der räumlichen Abdeckung zu validieren. Gegenstand dieser Arbeit soll es sein, ein geeignetes Konzept für die Erstellung von Beacon-Konfigurationen zu gestalten und die Anordnung anschließend experimentell zu überprüfen. Dabei wird auf ein automatisiertes und strukturiertes Verfahren mithilfe von Robotern zurückgegriffen, um die Evaluierung weniger fehleranfällig zu designen und somit zu standardisieren.

Aus diesen Anforderungen stellen sich drei zentrale Zielstellungen:

- Entwicklung eines Frameworks und einer systematischen Versuchsplanung für die Messung von Beacon-Signalen
- Auswahl und Parameterbestimmung eines Modells für die Signalausbreitung von Beacons mit anschließender Erstellung einer Simulationsumgebung sowie einer sich daraus konkludierenden optimalen Verteilung von Funkbaken in einem Raum
- Validierung des Modells und der Simulationsergebnisse in einem Testszenario mithilfe eines Roboters

Um die Problematik eines bisher fehlenden Planungskonzeptes darzulegen, wird zuerst der Stand der Technik von Beacons beschrieben und erläutert. Daraus schlussfolgern sich in Kapitel 3 erste Ansätze darüber wie ein mögliches Verfahren zur Beacon-Konfiguration aufgebaut sein muss. In Kapitel 4 werden für die erste Zielsetzung, der Modellierung von Beacons, alle Werkzeuge zur Messung der Funksignale vorgestellt und ausgewertet. Mit den Messwerten werden anschließend die Parameter für die Signalausbreitung der Beacons in einem freien Raum bestimmt und damit die Grundlage

für ein Simulationsprogramm gelegt. Da noch keine bekannten Veröffentlichungen über die Lokalisierungsgenauigkeit mittels Beacons existieren, wird dafür zusätzlich aus den Messwerten eine Vertrauensskala bestimmt und erklärt. Mithilfe der Simulation und eines Optimierungsalgorithmus werden anhand der selbstdefinierten Qualitätskriterien geeignete Beacon-Konfigurationen berechnet und diese in Kapitel 7 experimentell überprüft. Dahingehend wird im Anschluss besonders auf die Vorhersage-Genauigkeit aller getroffenen Annahmen eingegangen und die statischen Messungen werden mit den dynamischen verglichen. Am Ende der Arbeit werden die drei verfassten Zielstellungen mit dem Erreichten verglichen und die Ergebnisse daraus diskutiert. Abschließend wird das Konzept basierend auf den Forschungserkenntnissen entwickelt.



# 2 Stand der Technik – Einführung in die Beacon-Technologie

Der erste Teil dieses Kapitels beschäftigt sich mit der Entstehungsgeschichte von Beacons und nachfolgend mit deren Funktionsweise. Hier wird besonders auf die technischen Möglichkeiten der Technologie eingegangen. Im letzten Abschnitt wird erläutert, wo Beacon-Systeme zum Einsatz kommen und wie aktuelle Lösungsansätze für die Planung der dafür nötigen Infrastruktur aufgebaut sind. Anschließend wird aus den gegebenen Charakteristika der Beacons und der momentan genutzten Fähigkeiten differenziert und eine Aussicht auf ein zukünftiges Konzept zu einer effizienteren Nutzung gegeben.

## 2.1 Entwicklungsgeschichte

Den Grundstein für die Beacon-Technologie legte das finnische Unternehmen Nokia im Jahre 2006. Damals entwickelte die Firma den neuen Standard *Wibree* für die Funkübertragung, der den veralteten Bluetooth-Standard ersetzen sollte. Mit der Neuentwicklung versprach man sich im Gegensatz zu Bluetooth einen geringeren Stromverbrauch und geringere Produktionskosten bei gleichbleibenden Übertragungsraten. Ab dem Jahr 2009 wurde der Bluetooth-Standard um Wibree ergänzt und erst unter den Bezeichnungen *Ultra Low Power Bluetooth* (ULP) und dann später als *Bluetooth Low Energy* (BLE) darin aufgenommen [9] und anschließend als *Bluetooth Smart* vermarktet. Da viele Hersteller von mobilen Geräten in ihren Datenblättern die Unterstützung von BLE nicht explizit erwähnen, findet sich meist folgendes Logo 2.1 in den Produktbeschreibungen und das erkennen lässt, ob die Geräte den neuen Standard unterstützen.

Die Idee der Nutzung von Bluetooth Low Energy zur Indoor-Lokalisierung stammt dabei von der Firma Apple Inc. und wurde von ihr im Jahre 2013 auf der WWDC (Worldwide Developers Conference)[11] unter dem Namen *iBeacon* angekündigt. Obwohl zu dem Zeitpunkt noch kein fertiges Gerät zur Verfügung stand, wurde diese Technologie als Neuerung in Apples mobilem Betriebssystem iOS 7 vorgestellt. Jedoch verzichtet Apple seither auf die Produktion von iBeacons, was andere Unternehmen nutzen, um selbst in den Markt einzusteigen. Deren Produkte wurden darauf in Bea-

cons umbenannt und unterstützen zusätzlich die mobilen Betriebssysteme Android ab Version 4.3, Windows Phone 8 und die neueste Version von Blackberries OS [3]. Somit wäre die Beacon-Technologie mit der nötigen Hardware-Unterstützung in mittlerweile über 99,5% aller mobilen Geräte (Smartphones, Tablets, Smartwatches etc.) weltweit nutzbar [29].



Abb. 2.1: Offizielles Bluetooth Smart Logo [10]

## 2.2 Aufbau und Funktionsweise von Beacons

Die Grundbausteine der kleinen Leuchtfeuer sind in der rechten Abbildung 2.2 ersichtlich, in der ein *Estimote Beacon* der Firma Estimote Inc. in seine einzelnen Bestandteile untergliedert ist. Ein Beacon misst ungefähr  $5,5 \text{ cm} \times 3,5 \text{ cm} \times 1,5 \text{ cm}$  in Länge, Breite und Höhe und ist dabei 50 g schwer. Zur Anbringung der Beacons an Wänden, Decken usw. dient die auf der Rückseite befindliche Silikonplatte. Diese haftet an nahezu jeder glatten Oberfläche und kann mithilfe von Wasser sehr einfach gereinigt werden. Somit können die Beacons im Grunde unzählige Male angebracht und wieder abgenommen werden. Die äußere Schutzhülle besteht dabei ebenfalls aus einem Silikon und schützt die inneren Bauteile. Zu den inneren Komponenten gehören die Platine mit dem darauf verlötzten Nordic nRF51822 Chip (32-Bit ARM Cortex M0 CPU mit 256 kB Flash-Speicher und dem 2,4 GHz BLE-Sendemodul) [31], eine Antenne und eine Knopfbatterie zur autarken Stromversorgung. Die Beacons haben eine variable Sendeleistung von 4 dBm bis -30 dBm (entspricht einer Leistung von 2,512 Milliwatt bis 1 Mikrowatt) und übertragen ihre Daten in Intervallen von 50 bis 0,5 Hertz. Die Kommunikation verläuft dabei bidirektional, d.h. vom Beacon zum Empfangsgerät und zurück. Während die Kommunikation eines Beacons zu einem mobilen Gerät dazu dient, die Lokalisierung des Gerätes zu ermöglichen, eignet sich die Kommunikation von einem Smartphone, Tablet etc. zu einem Beacon zu dessen Programmierung und zur Überprüfung des Betriebszustandes,

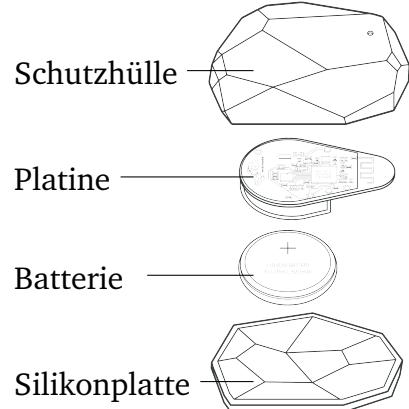


Abb. 2.2: Explosionszeichnung Beacon [13]

wie z.B. dem Akku-Ladezustand.

Die bei der Signalübertragung verwendete Protokoll-Architektur Bluetooth Low Energy sendet dabei im 2,4 Ghz Band, welches ebenfalls von den Protokollen 802.11 ac/a/b/g/n, älteren Bluetooth-Standards, ZigBee, NFC, etc. genutzt wird. Eine kompakte Übersicht zu den einzelnen Varianten und deren Eigenschaften findet sich im Anhang in Bild A.1. Die Vorteile des 2,4 Ghz Bandes gegenüber anderen Frequenzbändern ergeben sich aus einer großen Reichweite der Funksignale und einer geringen Größe der Antenne für deren Erzeugung. Jedoch besitzt dieses Band auch gewisse Nachteile, die bei der Auslegung von Beacon-Konfigurationen beachtet werden müssen. Die Verwendung des Protokolls im 2,4 Ghz Band ist dabei historisch bedingt und nahezu alternativlos, da es zu den ISM-Bändern [20] zählt und nur diese somit frei nutzbar sind. Bei vermehrter Nutzung des 2,4 Ghz Bandes für die Datenübertragungen in WLAN oder Bluetooth können geringfügige Störungen in den Übertragungen auftreten. Somit hängt die Qualität der Signale und schlussendlich auch die Lokalisierungsgenauigkeit von der lokalen Auslastung des 2,4 Ghz Bandes ab. Des Weiteren können auch sich zwischen Sender und Empfänger befindliche physische Objekte einen störenden Faktor auf die Empfangsqualität ausüben. In den Bildern 2.3 und 2.4 ist eine mögliche Beeinträchtigung der Empfangsqualität von BLE-Signalen durch eine im Sichtfeld befindliche Wand oder durch einen menschlichen Körper dargestellt. Denn durch die physikalischen Eigenschaften des 2,4 Ghz Bandes werden die Funksignale durch Materialien wie Wasser und Stahl besonders gut absorbiert. Die meisten Gebäude wurden aus Stahlbeton errichtet und der menschliche Körper besteht aus einem großen Anteil aus Wasser. Diese Umstände wirken sich besonders negativ auf die Qualität der BLE-Signale aus und eignen sich dadurch eigentlich nicht für die Anwendung einer Lokalisierung von Menschen in Gebäuden. Jedoch sei dies nur am Rande erwähnt und würde unter zusätzlicher Betrachtung dieser Aspekte den zeitlichen Rahmen dieser Arbeit in großem Ausmaß sprengen. Zusammenfassend werden hier noch einmal die Gründe für eine Qualitätsminderung der Lokalisierung festgehalten: die Schwächung der Signale an Objekten, die Phasenauslöschung durch andere Signalquellen und unberechenbare Reflexionen.

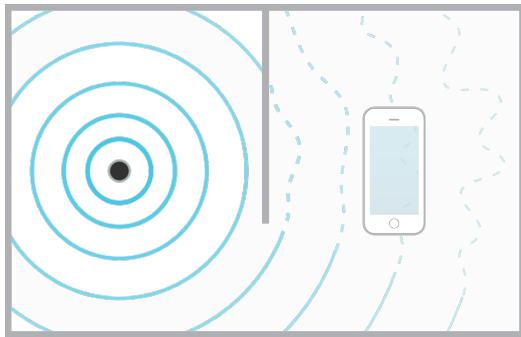


Abb. 2.3: Signalschwächung durch Objekte, z.B. Wände [7]

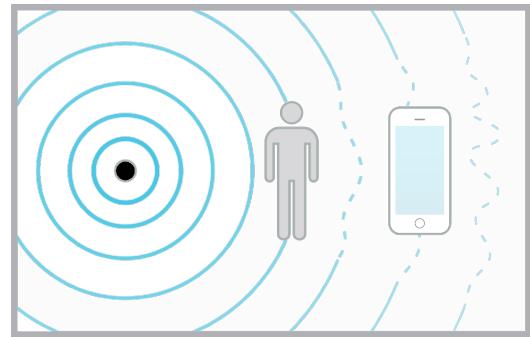


Abb. 2.4: Menschliche Körper blockieren zusätzlich die Signale [7]

Die Daten, die bei der Kommunikation übertragen werden, sind dabei essentiell für die Positionsbestimmung von Objekten und bestehen aus einer:

- Identifikationsnummer (Länge von 16 Bytes),
- eingestellten Sendeleistung (2 Bytes)
- und zusätzlichen Information, beschrieben als Major und Minor (jeweils 2 Bytes).

Mithilfe geeigneter Software lassen sich diese Parameter ändern und den Anforderungen entsprechend anpassen. Zusätzlich lässt sich noch die Intervalllänge der Sendefrequenz variieren. Im Falle der Estimote Beacons geschieht dies mit der kostenlosen Software „Estimote“ und ist für die Plattformen iOS und Android auf mobilen Geräten verfügbar. Mithilfe dieses Werkzeuges kann die Ortungsgenauigkeit signifikant erhöht bzw. verringert werden. Da die genauen Einflüsse dieser Parameter auf die Batterielebensdauer und die Ortungsgenauigkeit noch nicht erforscht wurden oder zumindest nicht öffentlich zugänglich sind, werden diese in Kapitel 4 untersucht und erklärt. Denn gerade der adaptive Aspekt für die Indoor-Lokalisierung lässt diese Technologie so attraktiv erscheinen.

## 2.3 Ortungsmethoden

Mit der Identifikationsnummer kann zwischen den einzelnen Beacons unterschieden werden, wodurch erst die Möglichkeit einer Lokalisierung entsteht. Denn die Technik setzt voraus, dass die Identitäten der Beacons in einer Datenbank mit Positionsangabe in einem Gebäude hinterlegt sind. Beim Empfang eines gültigen Signals wird der Sender in der Datenbank gesucht und anschließend dessen Distanz zum Empfangsgerät geschätzt. Die Berechnung findet auf der Grundlage eines Ausbreitungsmodells von Signalen in Abhängigkeit zur eingestellten Sendeleistung des Beacons statt, in der

die empfangene Signalstärke als RSSI-Wert interpretiert und dadurch eine Distanz geschätzt wird. Die Modelle der Hersteller sind typischerweise nicht frei zugänglich, weswegen sie auch hier nicht vorgestellt werden. Die als zusätzliche Informationen gekennzeichneten Daten sind eine Erweiterung der Identifikationsnummer und bieten lediglich einen gesteigerten Komfort für die Entwicklung der Datenbanken. Im nächsten Abschnitt wird dies in Tabelle 2.1 noch einmal veranschaulicht.

Geschäftsstandort		Berlin	Magdeburg	München
<b>UUID</b>		U8T7V56I-4689-10U9-7G63B4GAR21M		
<b>Sendeleistung</b>		-12 dBm	-6 dBm	1 dBm
<b>Major</b>		1	2	3
<b>Minor</b>	<b>Kleidung</b>	10	10	10
	<b>Elektronik</b>	20	20	20
	<b>Küche</b>	30	30	30

Tabelle 2.1: Beispiel der Informationsnutzung; in Anlehnung an: [7]

Bei dem obigen Beispiel nutzt eine Warenhauskette mit mehreren Geschäftsstandorten die Beacon-Technologie zur Lokalisierung ihrer Besucher an ihren verschiedenen Standorten. Zur Vereinfachung besitzen die Beacons einen *Universally Unique Identifier* (UUID) und unterscheiden sich jeweils nur in ihren zusätzlichen Informationen und der Sendeleistung. Die gezeigten Informationen liegen auf den mobilen Geräten der Nutzer vor, genauso wie eine Applikation auf den Geräten vorhanden sein muss, die diese Daten verarbeiten kann. Die Information Major steht dabei für den jeweiligen Standort und Minor für die Abteilung, in der die Signale der Beacons empfangen werden können und der RSSI-Wert in einem definierten Bereich liegt. Abhängig vom Konzept der Ortung kann dieses Wissen unterschiedlich genutzt werden. Bei der Beacon-Technologie unterscheidet man daher zwischen zwei Anwendungskonzepten der Ortung.

### 2.3.1 Definitionen verschiedener Anwendungsfelder

Die einfachste Form einer Lokalisierung ist die Aufteilung eines Raumes in Bereiche, in der die Position eines Nutzers zu einem Beacon dadurch angegeben wird, ob er im Nah- oder Fernbereich zu ihm positioniert ist. Diese Art der Ortung ist dabei sehr ungenau, da die Distanzinformationen nicht explizit vorliegen, sondern nur ein homogener Distanz-Bereich um ein Beacon definiert ist. Die Distanz zu einer Funkbake wird dabei durch die empfangene Signalstärke von einem Beacon geschätzt. Diese Beurteilung beruht derzeit noch auf den Erfahrungen des Beacon-Programmierers bzw. des Installateurs der Beacon-Systeme. Als Beispiel für eine solche Definition wurden einmal vier

Zustände in der Tabelle 2.2 festgelegt und deren Bereiche in Abhängigkeit zu der empfangenen Sendeleistung eines Beacons eingegrenzt. In einem fiktiven Einsatzszenario könnte somit ein Kunde in einem Supermarkt oder Warenhaus gezielt auf ein Sonderangebot in seiner Nähe aufmerksam gemacht oder zusätzliche Informationen zu einem Produkt in einer entsprechenden Applikation angezeigt werden.

Lagebeschreibung	Definition
Sehr nah	Dieser Bereich besitzt eine sehr hohe Wahrscheinlichkeit, dass der Nutzer direkt vor einem Beacon steht. Dies gilt für einen Bereich, der in einer Distanz von unter 1 Meter zum Beacon liegt.
Nahbereich	Hier befindet sich der Bereich in einer Sichtlinie zu einem Beacon in einer Distanz von 1 bis 5 Metern. Da es aufgrund von Störungen, wie vorbeilaufender Menschen oder anderer Objekte zur Signalbeeinträchtigung kommen kann, könnte dieser Zustand nicht angezeigt werden, obwohl das Empfangsgerät in diesem Bereich liegt.
Fernbereich	Hier werden zwar die Signale von einem Beacon empfangen, jedoch kann aufgrund der Signalschwäche dem Empfänger kein eindeutiger Bereich zugeordnet werden. Dies impliziert jedoch keine große Entfernung zum Beacon, da wegen der genannten Störungen das Signal möglicherweise verfälscht wurde. Hier müssen weitere Verfahren und Strategien angewendet werden, um den Nutzer und sein Empfangsgerät genauer zu lokalisieren. Beispielsweise kann dem Nutzer empfohlen werden, sein Empfangsgerät höher zu halten oder er sollte ein wenig umherlaufen.
Kein Empfang	Hier wurde ein Beacon nicht erkannt und somit liegt keine physische Nähe zum Beacon vor.

Tabelle 2.2: Beispiel der Bereichsdefinition; in Anlehnung an: [7]

### 2.3.2 Mikro-Lokalisierung

Im Gegensatz zur relativen Lokalisierung mittels Lagebeschreibung existiert außerdem die Methode der Mikro-Lokalisierung. Diese berechnet eine genaue Distanz zu einem Sender und letztlich kann mithilfe von zusätzlichen Signalen mehrerer Beacons eine genaue Lokalisierung mit Koordinaten im Raum durchgeführt werden. Für die Bestimmung einer Position aus den empfangenen Beacon-Signalen existieren mehrere Verfahren. Eine gängige Methode ist die Tri- bzw. Multilateration, die aus der Entfernung eines Punktes zwischen drei oder mehreren Orientierungspunkten dessen Position in einem Koordinatensystem bestimmt. In diesem Fall ließen sich die Positionen der Beacons, die Orientierungspunkte und die Entferungen aus den Signalen der Beacons berechnen. Mit einem Modell der Signalausbreitung wird dabei die Signalstärke, die durch den

Weg vom Beacon zum Empfangsgerät geschwächt wurde, direkt in eine Distanz umgerechnet. Im übernächsten Kapitel wird ein solches Modell dafür beschrieben. An dieser Stelle soll noch einmal auf die Vorgehensweise zur Lokalisierung mittels Trilateration eingegangen werden. Abbildung 2.5 dient zur Veranschaulichung der Lösung. Im ersten Schritt werden dazu die Positionen der Beacons zur Vereinfachung in ein neues Koordinatensystem transformiert. Dabei wird ein Beacon in den Koordinatenursprung verlegt und ein zweiter durch Rotation um die Z-Achse auf die X-Achse gesetzt.

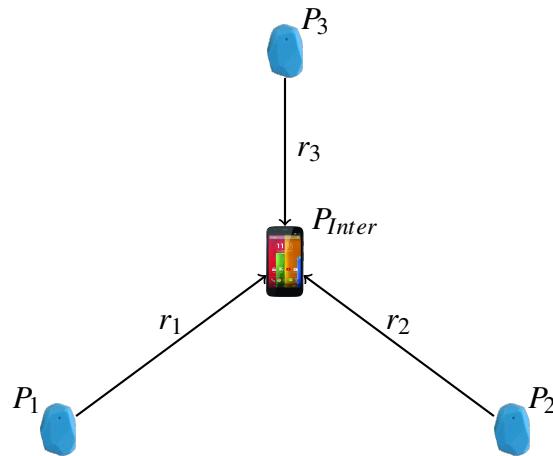


Abb. 2.5: Beispiel einer Trilateration zur Positionsbestimmung

Translation:

$$T = -P_1$$

$$P'_1 = P_1 + T = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$P'_2 = P_2 + T$$

$$P'_3 = P_3 + T$$

Rotation:

$$\alpha = -\arcsin \left( \frac{y_2}{\sqrt{x_2^2 + y_2^2}} \right)$$

$$R_z(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

$$\begin{aligned} P_1'' &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ P_2'' &= R_z(\alpha) \cdot P_2' \\ P_3'' &= R_z(\alpha) \cdot P_3' \end{aligned}$$

Anhand zweier Formeln lassen sich dann die Koordinaten des Empfangsgeräts aus den einzelnen Distanzen berechnen und schließlich ins ursprüngliche Koordinatensystem zurück transformieren [34]. Die Distanzen sind dabei gegeben als  $r_1$ ,  $r_2$  und  $r_3$  und die Koordinaten entsprechen der Nummerierung der Punkte nach der Transformation.

$$\begin{aligned} x_{Inter} &= \frac{r_1^2 - r_2^2 + x_2^2}{2x_2^2} \\ y_{Inter} &= \frac{r_1^2 - r_3^2 + x_3^2 + y_3^2}{2y_3^2} - \frac{x_3}{y_3} \cdot x_{Inter} \end{aligned}$$

Rücktransformation:

$$P_{Inter} = R_z(-\alpha) \cdot \begin{pmatrix} x_{Inter} \\ y_{Inter} \end{pmatrix} - T$$

Ein beträchtlicher Nachteil dieser Methode ist die ausgedehnte Fehleranfälligkeit der Signalübertragung. In der Anwendung in Gebäuden werden die Signale oftmals von Wänden reflektiert oder gedämpft, während die hohe Dichte von Signalquellen weitere Interferenzen verursacht. Um diese Nachteile auszugleichen, wird statt der Trilateration eine Multilateration angewendet. Durch die Nutzung mehrerer Beacons können Messrauschen und Störungen zum Teil ausgeglichen und die Genauigkeit der Lokalisierung somit erhöht werden. Jedoch kann in dieser Arbeit nicht auf dieses Verfahren eingegangen werden, da lediglich drei Beacons zur Verfügung standen.

## 2.4 Anwendungsbereiche der Indoor-Lokalisierung

Der Bedarf an einer innerräumlichen Ortung ist normalerweise dort vorhanden, wo es für den Besucher schwer ist, sich zu orientieren. Dies können weitläufige Gebäudekomplexe sein oder verwinkelte und unüberschaubare Gänge. Infolge der Digitalisie-

rung der Gesellschaft (oft als „Digitale Revolution“ oder in einem anderen Kontext als „Zweite Moderne“ [16] bezeichnet), entstehen durch den starken Grad der Verbreitung von mobilen „smartten“ Geräten riesige Netzwerke mit einer schier unendlichen Datenflut. Diese Entwicklung eröffnet folglich eine Möglichkeit zur Kommunikation zwischen Mensch und Gebäude, in der die Beacons als „Sinne“ des Gebäudes verstanden werden können.

### 2.4.1 Mögliche Einsatzszenarien

Viele Unternehmen nutzen die Möglichkeiten einer vernetzten Welt, um das Verhalten der Kundschaft besser verstehen zu können. Das Beispiel der Warenhauskette zeigt, dass das Wissen für das Unternehmen von Vorteil wäre, wie viele Kunden am Tag eine Filiale besuchen, wie ihr Bewegungsprofil aussieht und was sie am Ende einkaufen. Nützlich wären die Informationen für die Preisgestaltung, die Anordnung der Abteilungen im Warenhaus und die Erkenntnisse daraus könnten auch zu einer besseren „Just-In-Time“ Lieferkette führen und somit Lagerkapazitäten einsparen. Natürlich wäre dies auch mit herkömmlichen Methoden möglich, indem z.B. Umfragen stattfinden und die Mitarbeiter eines Geschäftes die Kunden genau beobachten würden. Dies wäre aufgrund hoher Personalkosten nicht nur unrentabel, sondern würde den Kunden außerdem ein Gefühl der Überwachung vermitteln. Mit der Beacon-Technologie ist dies kostengünstig und voll automatisiert möglich. Damit potentielle Kunden auch die dafür nötige Software auf ihrem mobilen Endgerät installieren und der Auswertung ihrer Daten zustimmen, können die Kunden mit einer entsprechenden Applikation an exklusiven Gewinnspielen, Rabattaktionen oder Punktesystemen teilnehmen, so wie es heute schon durch einige Unternehmen angeboten wird (z.B. DeutschlandCard, Payback etc.). Die gesellschaftliche Akzeptanz wäre sicherlich gegeben, da schon mit den modernen Stauwarnsystemen ein Vergleichsfall im Outdoor-Bereich existiert. Da die Position eines Fahrzeugs entweder durch das eingebaute Navigationssystem bekannt oder durch die Mobilfunkgeräte der Fahrzeuginsassen ermittelt werden kann, werden schon heute diese Systeme zur Erschaffung von Bewegungsprofilen und damit zur Stauvorhersage genutzt [19]. Für ein solches Szenario im Indoor-Bereich würde hauptsächlich die kontextbezogene Lokali-



Abb. 2.6: Beacon-Nutzung in Geschäften [1]

sierung in Betracht gezogen werden, da hier die meisten Geschäfte überschaubar sind und schon gute Strukturen zur Orientierung der Kundschaft genutzt werden. Extreme Ausmaße annehmende Einkaufszentren wie die „Golden Resources Mall“ in Peking mit 557,419 m<sup>2</sup> Ladenfläche, wo eine genauere Lokalisierung zur Navigation sicherlich sinnvoll wäre, bilden hier eher die Ausnahme.



Abb. 2.7: Beacon-Nutzung im Flughäfen [41]

Im Gegensatz dazu stellt die Mikro-Lokalisierung ein weiteres mögliches Szenario dar. Sie soll dazu dienen, Menschen oder Objekte genau zu lokalisieren, um unter Verwendung von speziellen Applikationen eine bessere Orientierung zu gewährleisten. Daran interessiert sind meist Branchen, die ihre Kundschaft gerne schnell und ohne Umwege zu dem führen möchten, weswegen sie in die Einrichtung des Unternehmens gekommen sind.

Als Beispiele können hier Messehallen- und Flughafenbetreiber genannt werden. In diesen Branchen sind die Gebäude meist sehr unübersichtlich und die Besucher häufig ortsfremd. Der Vorteil, den die Unternehmen aus dem System ziehen können, ist eine bessere Verteilung der Kunden durch eine intelligente Pfadplanung. So können die Platzkapazitäten der Einrichtungen besser genutzt und die Gäste somit schneller abgefertigt werden. Wenn zudem ein Unternehmen in seinen Gebäuden ein ähnliches System wie ein Navigationsgerät für die Straße bieten kann, bedeutet dies auch einen Wettbewerbsvorteil und fördert die Kundenzufriedenheit.

Grundsätzlich sind dies hier nur Annahmen für mögliche Szenarien. Da es noch keine praktikablen Anwendungen der Indoor-Lokalisierung gibt, können keine genauen Vorhersagen diesbezüglich getroffen werden. Viele Unternehmen schrecken noch davor zurück, weil kein einheitlicher Standard existiert und nur wenige Menschen mit der Technik vertraut sind.

## 2.4.2 Planungskonzepte für Beacon-Konfigurationen

Bisherige Ansätze um ein Lokalisierungssystem in einem Raum zu realisieren, beruhen auf fachmännischen Einschätzungen des Beacon-Installateurs. Das bedeutet, dass sich das Konzept lediglich auf Erfahrungswerte beruft. Die Beacons werden dabei an strategische Punkte gesetzt und manuell an jeder Position neu konfiguriert. Wie schon oben beschrieben, werden die Befestigungen der Beacons im Raum vermessen oder grob geschätzt und so zur Positionsbestimmung in eine Lokalisierungs-Applikation ebenfalls wieder manuell eingepflegt. Von der Firma Estimote gibt es seit Neuestem eine Anwendung namens „Estimote Indoor Localization“, die den Entwickler einer Navigations-App unterstützen soll. Der Entwickler installiert dazu an allen vier Wänden eines Raumes einen Beacon der Firma, startet ausgehend vom Eingang die App und läuft im Uhrzeigersinn den ganzen Raum ab. Dabei werden die Signale der Beacons an den Wänden aufgezeichnet und mithilfe der inertialen Sensoren des Smartphones zusätzlich die Bewegungsinformationen gespeichert. Daraus errechnet die App eine Karte vom Raum und liefert auch gleich eine Positionsbestimmung mithilfe der genannten Sensordaten in Echtzeit. Die Demonstration des Programms ist hierfür in Abbildung 2.8 einmal festgehalten. Der Vorteil dieser App ist es, dass die Karte auch in weiteren Programmen für das Smartphone verwendet werden kann, jedoch ist dafür eine Migration der Daten notwendig.

Da für die Experimente in dieser Arbeit nur drei Beacons zur Verfügung standen und diese App erst in der Endphase der Bearbeitungszeit veröffentlicht wurde, wird hier nicht mehr auf dieses Konzept eingegangen. Jedoch lassen die noch engen Begrenzungen der Anwendung (konstante Anzahl von vier Beacons, Einstellung der Parameter wieder manuell nach Erfahrungswert) keinen Handlungsspielraum für die Optimierung der Beacon-Konfigurationen zu. Zudem muss immer ein Entwickler die Räume ablaufen und später die Daten vom Smartphone in die eigene Entwicklung migrieren. Für kleine Räume ist diese Strategie sicherlich hilfreich, jedoch wird sie bei Großprojekten, wie z.B. der Schaffung eines Ortungssystems in einem Flughafen schlicht unbrauchbar.



Abb. 2.8: Demonstration der Estimote Indoor Localization App [14]



# 3 Konzeptplanung

Nach Betrachtung des Standes der Technik fällt auf, dass sowohl die Planung als auch die Inbetriebnahme der innerräumlichen Ortungssysteme stets ein experimentierhaltiger Prozess ist. Es existieren weder bekannte Theorien, noch computergestützte Hilfsmittel für deren Auslegung und Validierung. Zum einen liegt es daran, dass es dafür noch keine Notwendigkeit gab und zum anderen, weil aufgrund der Vielseitigkeit der Anwendungsgebiete und Situationen noch kein gemeinsamer Standard gefunden worden ist. Dabei ist das Problem der Erstellung einer guten Indoor-Lokalisierung nicht trivial und hoch komplex, wenn all die Faktoren hinzugerechnet werden, die das System beeinträchtigen können. Zudem verliert sich der Überblick über alle Signalquellen und deren Position in großen Gebäuden, weswegen auch noch keine Großprojekte diesbezüglich entstehen. Im Folgenden soll in diesem Abschnitt ein Konzept entwickelt werden, das die Theorie mit ergänzender Simulationstechnik im Zusammenspiel mit Experimenten unterstützt und somit ein bekanntes Schema in Wissenschaft und Industrie aufgreift: die Prozessplanung.

## 3.1 Vorüberlegungen

Der erste Schritt für eine systematische Prozessplanung ist die Schaffung einer Theorie bzw. eines Modells für die Signalausbreitung der BLE-Signale. Dieses Konzept muss erst durch Betrachtung der physikalischen Eigenschaften und mathematischen Beziehungen geschaffen oder anhand eines existierenden Modells aus vergleichbaren Technologien abgeleitet werden. Zusammen mit den Messungen werden die aufgestellten Annahmen schließlich auf das reale Verhalten durch eine geeignete Parametrisierung angepasst. Mithilfe der mathematischen Beschreibung des Ausbreitungsverlustes lässt sich anschließend eine Simulation erstellen. Dabei wird die Stärke der Beacon-Signale, abhängig von der Position der Funkbaken, an jedem Punkt im Raum prädiktiviert. Unter Erfüllung von definierten Optimalitätsbedingungen kann somit die Infrastruktur des Lokalisierungssystems effektiv geplant, im Anschluss die Anordnung der Beacons so je nach Einsatzszenario angepasst und vor der eigentlichen Installation simuliert und getestet werden. Jedoch muss die fertige Konfiguration erneut durch Messungen auf die gewünschten Funktionen und Qualitätsmerkmale der Indoor-Lokalisierung hin geprüft werden. Dieser Kreislauf aus Experimenten, Modellierungen und Simulationen wird dazu in Abbildung 3.1 grafisch veranschaulicht. Der Vorteil dieser Herangehensweise ist

die fundierte Lösung eines Problems, sodass auf die gewonnenen Informationen aufgebaut werden kann und sich mit ihrer Hilfe wissenschaftlich arbeiten lässt. Für die Planung eines Indoor-Lokalisierungsproblems mittels Beacons bedeutet das, dass die Konfigurationen vorab geplant werden können und diese einheitlich aufgebaut sind. Dadurch ergeben sich auch kommerzielle Vorteile durch eine schnellere Entwicklung von Ortungs- und Navigations-Applikationen und weiterer Softwareprojekte. Zudem wird die Leistungsfähigkeit der Beacon-Technologie messbar, die zu einer besseren Definition ihrer Einsatzmöglichkeiten führt.

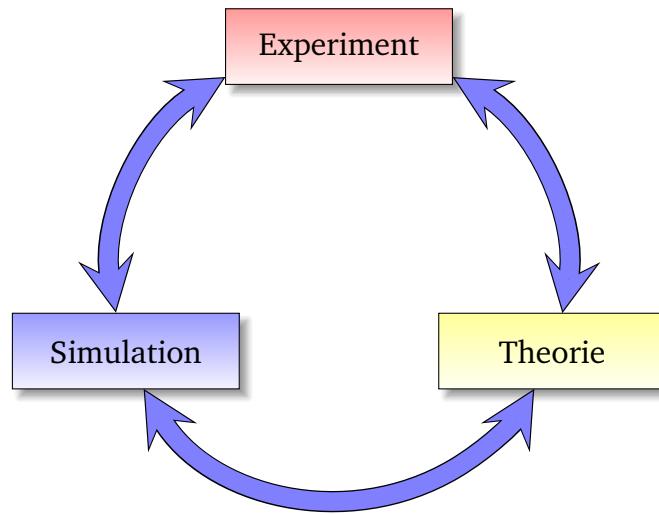


Abb. 3.1: Kreislauf einer Prozessplanung

## 3.2 Zielsetzungen

Aus den Vorüberlegungen ergibt sich eine klar definierte Struktur, die nun konkret mit einzelnen Abläufen in diesem Absatz beschrieben wird. Die Zielsetzungen sind dabei zum einen, ein Modell für die Ausbreitung von BLE-Signalen zu finden und zu parametrisieren. Für ein systematisches Vorgehen sind die Aufnahme der Messwerte und eine Plattform als Empfänger der BLE-Signale notwendig. Nachfolgend muss das Modell in eine Simulationsumgebung eingepflegt und ein Optimierungsalgorithmus für die Positionierung der Beacons implementiert werden. Im letzten Schritt soll als Abschluss der Arbeit ein Testfeld mit einer simulierten Konfiguration aufgebaut und die realen Signalstärken daraus mit den vorher berechneten verglichen werden. Dafür wird ein mobiles System benötigt, welches vorgegebene Punkte im Feld des Lokalisierungssystems ansteuern und seinen Standpunkt mit der errechneten Position aus der Trilateration vergleichen kann. Um all diese Punkte zu erfüllen, müssen Systeme und Werkzeuge gefunden werden, die die Konzeptplanung unterstützen. Im Anschluss werden diese näher

benannt und deren Zusammenspiel erläutert.

### **3.2.1 Hardware-Anforderungen**

An das Empfangsgerät wird lediglich die Anforderung gestellt, dass die Messung eines Beacon-Signals mit der einer Position synchronisierbar ist und das Gerät somit über Netzwerkfähigkeit oder Speicher verfügen muss. Dies ist gerade in der Validierungsphase von hoher Wichtigkeit, um standardisierte und reproduzierbare Ergebnisse zu gewährleisten. Da die eigentliche Nutzung des Positionierungssystems mithilfe von Smartphones stattfindet, ist die Verwendung dieser Endgeräte für die Messungen durchaus zu empfehlen. Denn Smartphones besitzen eine große Bandbreite an Schnittstellen und einen internen Speicher, sodass jeweils eine Strategie für die Messwertaufnahme verfolgt werden kann. Zudem existieren von den Herstellern der kleinen Sender bereits fertige Bibliotheken und Entwicklertools, um die Signale von ihren Beacons zu verarbeiten, was einen geringeren Aufwand für die Implementierung bedeutet. Zudem weisen Antennen der Empfangsgeräte Charakteristiken auf, die die Aufnahme der Messungen je nach Ausrichtung des Smartphones beeinflussen. Somit gilt es auch diese Einflüsse zu untersuchen und da die meisten Smartphones auch über Inertialsensoren verfügen, bieten sie gleich zusätzliche Sensordaten in einem Gerät. Die Problematik der Antennencharakteristiken findet sich dabei im nächsten Kapitel. Für die standardisierte Aufnahme der Messwerte bietet sich eine Kombination aus Maschine bzw. Roboter und Smartphone an. Für die Überprüfung der Beacon-Konfiguration bietet sich die Verwendung von Roboter-Systemen an. Diese sind mobil, können um eine Positionsbestimmung unabhängig von der Beacon-Ortung erweitert werden und besitzen ebenfalls durch entsprechende Middleware geeignete Kommunikationsfähigkeiten. Hierbei muss nur darauf geachtet werden, dass der Roboter eine entsprechende Halterung für das Smartphone besitzt oder eine an ihm angebracht werden kann. Darüber hinaus kann der gesamte Prozess durch die Verwendung von Robotern in den Experimenten automatisiert werden, was ebenfalls ein Ziel dieser Arbeit darstellt.

### **3.2.2 Konzept der Software-Architektur**

Im Mittelpunkt der Software-Architektur steht die Datenübertragung der Messwerte aus verteilten Systemen auf ein zentrales System, um diese später im Verarbeitungsschritt zu synchronisieren. Um dies zu gewährleisten, müssen alle Plattformen miteinander kommunizieren können und die Art der Übertragung darf dabei nicht zu starken Verzögerungen der Messwertaufnahme führen. So wird ein effektives und schnelles Framework benötigt, das die eigentlichen Messungen nicht beeinträchtigt. Dabei muss

zwischen den statischen und dynamischen Experimenten unterschieden werden. Denn für die Parameterbestimmung des Modells wird keine ständige Positionsmessung benötigt, da diese „per Hand“ ermittelt werden kann und sich im Laufe einer Messreihe für eine Distanz auch nicht ändert (statische Messungen). Dahingegen fließt bei der Überprüfung eines bestehenden Lokalisierungssystems die „externe“ Lokalisierung eines sich bewegenden Roboters in den Vektor aus Messwerten mit ein (dynamische Messungen), sodass über das Framework der Kommunikation verschiedene Datentypen von unterschiedlichen Plattformen auf die zentrale Verarbeitungsstelle geleitet werden müssen. Die Anforderungen an die einzelnen Anwendungen und unteren Ebenen der Software werden in Abbildung 3.2 einmal dargestellt. Die Aufgaben an die einzelnen Anwendungen sind beim Smartphone die Erfassung der einzelnen Beacon-Signale und die Messung der eigenen Ausrichtung. Im Gegensatz dazu muss der Roboter seine eigenen Positionsdaten übermitteln und zudem die auf einem externen Computer in der Simulation erstellten Raumelemente ansteuern und somit Befehle vom PC entgegennehmen. Die Simulationsumgebung speichert alle Sensordaten der verteilten Systeme und verarbeitet diese für die eigentliche Validierung der Beacon-Konfiguration.

Da die Simulation zur Erstellung eines Indoor-Lokalisierungssystems in Gebäuden bzw. Räumen angewendet wird, muss die Simulationsumgebung zuerst den Grundriss eines Gebäudes analysieren und die gewonnenen Daten für die spätere Verwendung verarbeiten. Es bietet sich dabei an, den Raum in einzelne Elemente zu unterteilen, sodass die Simulation vereinfacht wird und mit diskreten Werten gerechnet werden kann. Die Informationen über die Anordnung von Objekten und Wänden ist dabei essentiell für die manuelle oder automatische Verteilung der Sender im Raum und der Simulation der einzelnen Signalstärken in Abhängigkeit zu den Distanzen der Raumelemente zu den Beacons. Der grundlegende Ablauf ist in Abbildung 3.3 hierfür skizziert. Die Optimierung einer Anordnung von Beacons ist dabei stets subjektiv, da für jedes Szenario andere Anforderungen an das Ortungssystem gestellt werden. Es gilt, somit einen Optimierungsalgorithmus zu finden, der mit unterschiedlichen Ansprüchen je nach Situation umgehen und gegebenenfalls auf ein spezielles Optimierungsproblem angepasst werden kann. Auf die Optimierung wird speziell in Kapitel 5 eingegangen.

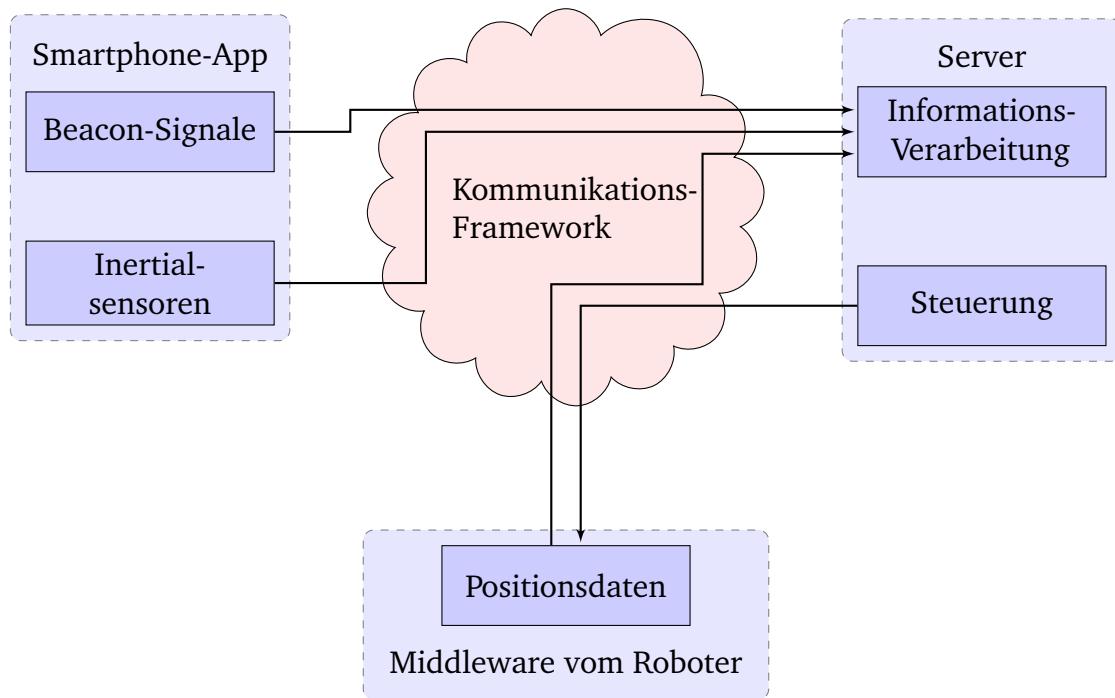


Abb. 3.2: Überblick über die Software-Anforderungen

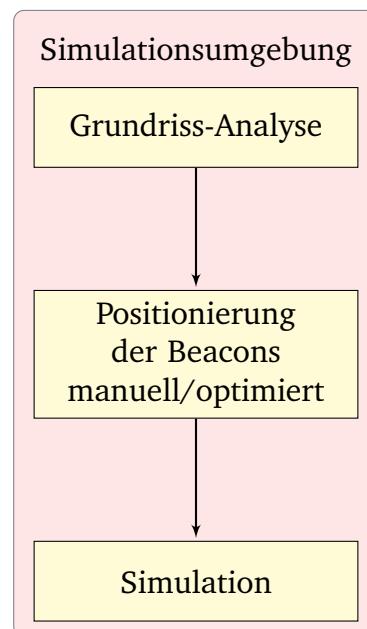


Abb. 3.3: Überblick über die Simulationsumgebung



# 4 Software-Entwicklung

Aus den anfänglichen Ideen für die Umsetzung des Lighthouse Keepers entstehen in diesem Kapitel die ersten Module für dessen Verwirklichung. Zum Anfang wird die verwendete Hardware vorgestellt und in Verbindung dazu werden die Aufgaben der genutzten Software-Komponenten näher erläutert. Durch die geeignete Wahl aller Bestandteile lassen sich viele Aufgaben auf vorgefertigte Module übertragen. Beispielsweise werden statt einer eigenen Entwicklung ein bestehendes Kommunikations-Framework und eine bereits existierende Middleware für den Roboter genutzt. Mit der Nutzung dieser fertigen Lösungen wird im großen Maße Arbeitsaufwand eingespart. Somit wird ein stärkerer Fokus auf das eigentliche Thema gelegt. Da dies im Hinblick auf die Smartphone-Applikation nicht immer möglich war, wird deren Implementierung in diesem Kapitel exemplarisch einmal dargestellt.

## 4.1 Verwendete Hardware

Da die Beacon-Technologie vorrangig zur Indoor-Lokalisierung von Personen eingesetzt wird und als Peripheriegerät meistens ein Smartphone Verwendung findet, wird auch ein solches für alle Experimente als Messgerät genutzt. Die Wahl fiel dabei auf ein Android-Smartphone mit dem Namen Motorola Moto G der gleichnamigen Firma Motorola Inc.. Es wurde ausgewählt, weil es alle Hardware- und Software-Anforderungen zum Empfang von BLE-Signalen erfüllt und als ein Standard-Smartphone gilt, sodass sich die mit ihm erzielten Ergebnisse auch auf andere Produkte übertragen lassen. Des Weiteren werden zwei Roboter in den Experimenten genutzt, um die Messungen reproduzierbar und standardisiert durchzuführen. Für die reinen Distanz-Signalstärke-Messungen wird der Roboter bzw. lediglich sein Arm namens „Youbot“ der Kuka AG und später für die Validierung einer Beacon-Konfiguration der Roboter „Scitos G5“ der MetraLabs GmbH verwendet.

### 4.1.1 Motorola Moto G

Das Moto G dient als Empfangsstation der BLE-Signale, dessen grundlegende Spezifikationen ein 1,2 GHz Snapdragon 400 Prozessor mit 1 GB RAM und ein WCN3620 BT/FM/WLAN RF Modul ausmachen [8]. Seine Abmaße betragen 129,9 mm × 65,9 mm × 11,6 mm bei einem Gewicht von 143 g. Weiterhin verwendet es standardmäßig

ein Android 4.3 als Betriebssystem, welches jedoch für die Experimente auf die Version 4.4 geupdated wurde. Neben der technischen Ausstattung und Software sind für die späteren Messungen die Antennen und deren Charakteristiken von hoher Bedeutung, denn deren Eigenschaften wirken sich direkt auf den Empfang der Signale aus. Um die Einflüsse besser zu verstehen, sind in den Abbildungen 4.1 und 4.2 die Anordnung der Antennen einmal skizziert. Dabei fällt auf, dass sich WLAN- und Bluetooth-Modul die selben Antennen teilen. Dies führt zu der Frage, ob es zu Konflikten in der Funktionsweise des Smartphones kommt, wenn gleichzeitig auf beide Module zugegriffen wird. Genauere Informationen hierzu finden sich im Abschnitt zur App-Entwicklung. Die zweite Frage die sich daraus stellt, ist die Veränderung der Empfangs- und Sendequalität des Moto G in verschiedenen Positionen. Eine Antenne hat je nach Bauform und Funktionsweise Bereiche, in der sie mit voller Leistung sendet und empfängt, aber auch Bereiche, in denen Funksignale sie weder verlassen noch erreichen können. Dieses Phänomen hat verschiedene physikalische Gründe. Jedoch sind diese komplexen nicht-linearen Eigenschaften der Antennen des Moto G nicht öffentlich bekannt und können auch nicht ohne Weiteres bestimmt werden. Es sei nur anzumerken, dass bei den Messungen auch darauf geachtet werden muss, wie und an welchen Haltepunkten das Moto G am besten befestigt wird, ohne deren Transceiver-Fähigkeiten negativ zu beeinflussen. Eine gute Annahme dabei ist es, das Smartphone so auszurichten als ob es flach in der Hand eines Menschen liegen würde. Der Hersteller ist schließlich darauf bedacht, sein Produkt für einen gängigen Betrieb auszulegen und somit auch die Konstruktion der Antennen dafür zu optimieren. Diese Annahme muss jedoch noch anhand von Messungen verifiziert werden.



Abb. 4.1: Vorderseite des Motorola Moto G [6]

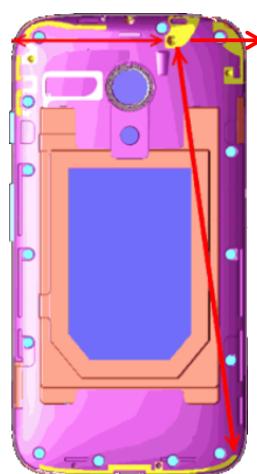


Abb. 4.2: Rückseite vom Moto G mit Antennen-Gerüst [8]

### 4.1.2 Youbot

Im vorigen Abschnitt wurde schon angesprochen, dass die Lageposition des Messinstrumentes zu seiner Empfangsleistung überprüft werden muss. Zudem sollte das Smartphone so gehalten werden als würde es sich in einer flachen Hand befinden, um dadurch die Nutzung durch einen Menschen zu simulieren. Für die Umsetzung der Anforderungen in einen automatisierten und reproduzierbaren Prozess, empfiehlt es sich, dabei einen Roboterarm als Messplattform zu benutzen. Aufgrund der Verfügbarkeit in der Universität wurde das Modell „Youbot“ der Firma „Kuka“ gewählt und für die Messungen mit einer Halterung aus einem 3D-Drucker ergänzt (siehe Abbildung 4.3). Die Vorteile des Systems sind zum einen der montierte hochpräzise Roboterarm mit Greifer auf dem Youbot und zum anderen, dass ein vollwertiger Rechner mit einem Linux Betriebssystem und eine drahtlose WLAN-Schnittstelle im System verbaut sind und so die Kommunikation zum Roboter sehr einfach aufgebaut werden kann. Der künstliche Arm kann sich dabei um seine fünf Achsen drehen (siehe Abbildung 4.4) und bietet somit genug Möglichkeiten, die Lageposition vom Moto G zu verändern.



Abb. 4.3: Youbot mit Halterung (gelber Aufsatz)

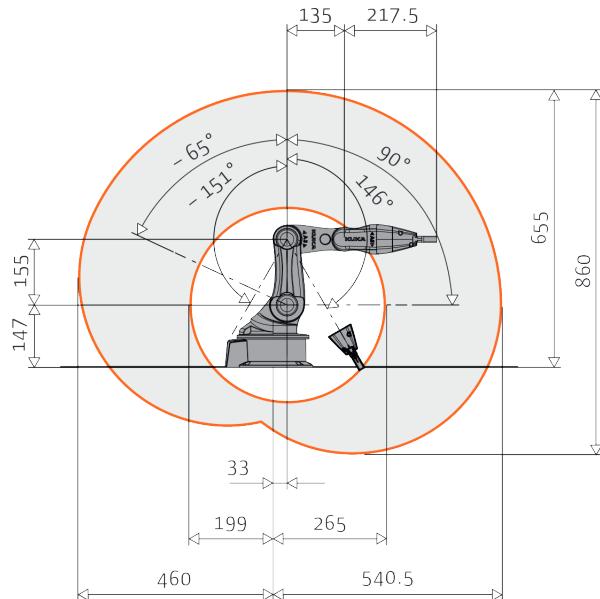


Abb. 4.4: Zeichnung eines Youbot-Arms und seiner fünf Rotationsachsen [24]

### 4.1.3 Scitos G5

Die Durchführung der Evaluation einer Beacon-Konfiguration wäre grundsätzlich auch mit dem modifizierten Youbot aus Abschnitt 4.1.2 möglich. Jedoch war in der Aufgabenstellung gefordert, den Roboter „Scitos G5“ von der MetraLabs GmbH zu verwenden.

Der Scitos G5 ist mit den Maßen 55 cm × 60 cm × 60 cm etwas größer als der Youbot und durch seine Dreirad-Lenkung weniger wendig (vgl. Abbildung 4.5). Die Vorteile gegenüber dem Youbot bestehen jedoch in seiner besseren Ausstattung und dem größeren Software-Umfang, weswegen es keiner zusätzlichen Entwicklung bedarf. Somit verringert sich der Arbeitsaufwand für den Entwurf des Lighthouse Keeper-Konzepts enorm. Der Scitos G5 verfügt über Schrittmotoren, einem Laserscanner und Inertialsensoren zur Navigation und er besitzt zudem einen leistungsfähigen Intel Core i7-Prozessor, WLAN und ein Linux-Betriebssystem, sodass auch hier genug Ressourcen für eine Kommunikation vorhanden sind. Die Machbarkeit des gesamten Kontroll-Prozesses von Beacon-Konfigurationen beruht dabei auf der Leistungsfähigkeit von diesem Roboter, denn er muss in der Lage sein, seine Position unabhängig von der Trilateration von Beacon-Signalen zu bestimmen und diese als Referenzquelle zur Verfügung zu stellen. Für seine Positionsbestimmung verwendet er die Software-Lösung „Miracenter“ (siehe 4.2.2 im nächsten Absatz), die die Sensordaten aus der Wahrnehmung (Motoren, IMU, Laserscanner) so verwertet, dass sie eine Karte der Umgebung erstellen kann. Mittels Abgleich von Sensorinformationen und der erstellten Karte kennt der Roboter somit stets seine aktuelle Position. Der Hersteller wirbt dabei mit einer Nutzlast von bis zu 50 kg und einer Maximalgeschwindigkeit von  $1,4 \frac{m}{s}$  bei einer batteriebetriebenen Laufzeit von ca. 20 Stunden [27], sodass ein künstlicher Roboterarm auf dem Scitos G5 ebenfalls denkbar wäre und somit auch die Aufgaben vom Youbot zukünftig übernehmen könnte.



Abb. 4.5: Scitos G5 von der MetraLabs GmbH

## 4.2 Verwendete Software

Um die Hardware zu nutzen und das geplante Konzept umzusetzen, benötigt es eine Kommunikation zwischen den Geräten und weiterer Werkzeuge zur Aufnahme von Messungen und deren Verarbeitung. Bei der Umsetzung wurde besonders auf Konformi-

tät der verschiedenen Systeme und deren reibungslosem Zusammenspiel geachtet. Um eine gemeinsame Basis zu schaffen, wurde das Software-Framework „Robots Operating System“(ROS) verwendet. Mit einem gemeinsamen Standard lassen sich die Messungen besser vergleichen, wodurch ihre Qualität und Aussagekraft zunimmt. Die Messungen müssen dabei auf der Smartphone- und der Roboter-Plattform aufgenommen und diese synchronisiert werden. Während ROS die übergeordnete Schnittstelle darstellt, müssen auf den einzelnen Hardware-Elementen die Messungen eigenständig durchgeführt werden. Die Messung auf dem Scitos-Roboter übernimmt dabei die Software „Miracenter“ ohne weiteres Zutun. Die Software für die Messungen auf dem Smartphone ist hingegen nicht vorgefertigt und muss mithilfe eines Editors für Android-Applikationen und einer speziellen Bibliothek für die Kommunikation Smartphone ↔ Estimote Beacon entwickelt werden.

#### **4.2.1 Robot Operating System – ROS**

Das „Robot Operating System“ oder kurz ROS ist ein Projekt zur Schaffung eines flexiblen Frameworks für die Entwicklung von Software für Roboter. Daraus entstand eine der mächtigsten Sammlungen aus Werkzeugen, Bibliotheken und Normen, um komplexe Verhalten zwischen Robotern über verschiedenste Robotik-Plattformen robust zu gestalten [36]. Der Anwendungsbereich für ROS ist demzufolge sehr umfangreich, jedoch werden für diese Arbeit nur folgende Eigenschaften [25] des Projektes benötigt und hier näher betrachtet:

- „Peer to Peer“ (P2P)-Verbindungen
- Modularer Aufbau
- Unterstützung mehrerer Programmiersprachen
- Freie Nutzung und Open-Source

Das Kommunikations-Framework des Lighthouse Keeper-Konzeptes, welches auf ROS aufbaut, besteht dabei aus mehreren miteinander verbundenen Rechnern (sog. „Hosts“) die zur Laufzeit über P2P miteinander kommunizieren. Bei der P2P-Verbindung können alle Teilnehmer ihre Dienste bzw. ihre Informationen gleichermaßen einander anbieten und nutzen, indem sie Daten im Netzwerk gleichzeitig empfangen und senden können. Dabei läuft auf einem zentralen Server der eigentliche Kern des Frameworks, über den sämtlicher Datenverkehr geleitet wird. Auf den Host-Systemen laufen dabei die eigentlichen modularen Anwendungen („Nodes“), die über Schnittstellen („Sockets“) des Betriebssystems ihre Daten auf das Netzwerk und schließlich an ROS verteilen. Dadurch

können die Nodes in verschiedenen Sprachen programmiert werden, die lediglich auf die entsprechende Schnittstelle zugreifen. Ausgehend von den Nodes werden den gesendeten Nachrichten gesonderte Bezeichnungen („Topics“) zugeordnet und mit einem Datentyp versehen. Diese Informationen sind allen Teilnehmern des P2P-Netzwerkes bekannt und können auch von ihnen angefordert werden. Der Kern organisiert dabei eine einheitliche Uhrzeit, die dadurch für alle Nachrichten bzw. deren Zeitstempel in den verteilten Systemen konsistent bleibt und so eine Synchronisierung der Messgeräte nicht mehr nötig wird. Somit ermöglicht die Verwendung von ROS den Aufbau der gesamten Kommunikation wie es in der Konzept-Planung in 3.2.2 gefordert wurde. Zudem erleichtert ROS durch seinen großen Funktionsumfang nachfolgende Erweiterungen und bietet somit Freiheiten für zukünftige Aufgaben.

#### 4.2.2 Miracenter

Mit dem Scitos G5 steht dem Anwender gleichzeitig auch die Software „Miracenter“ von der Firma MetraLabs GmbH zur Verfügung. Zusammen mit ihrer Erweiterung „CogniDrive“, dient sie als Navigator des Scitos G5 und ermöglicht es mit ihm den Grundriss eines Raumes zu erstellen (siehe Abbildung 4.6). Während beispielsweise der Roboter an den Wänden entlang fährt, messen seine Schrittmotoren den zurückgelegten Weg, die Inertialsensoiren verfeinern die Informationen und erweitern sie um die Ausrichtung des Roboters. Dabei misst zusätzlich der Laserscanner die Distanz des Roboters zu den Wänden und zu allen anderen festen Objekten in seiner Reichweite. Aus der Fusion aller Messwerte lässt sich so der Grundriss des zu vermessenden Raumes generieren. In obenstehendem Bild ist eine solche Karte als Beispiel aufgeführt. Die Daten die Miracenter dem Nutzer zur Verfügung stellt, bestehen dabei aus der Position des Roboters und seiner Ausrichtung in der Karte. Die Karte liegt dabei als „Portable Network Graphics“ (PNG)-Datei vor, wobei ein Pixel einer konstanten metrischen Länge entspricht, deren Verhältnis in einer „Extensible Markup Language“ (XML)-Datei definiert ist. Die

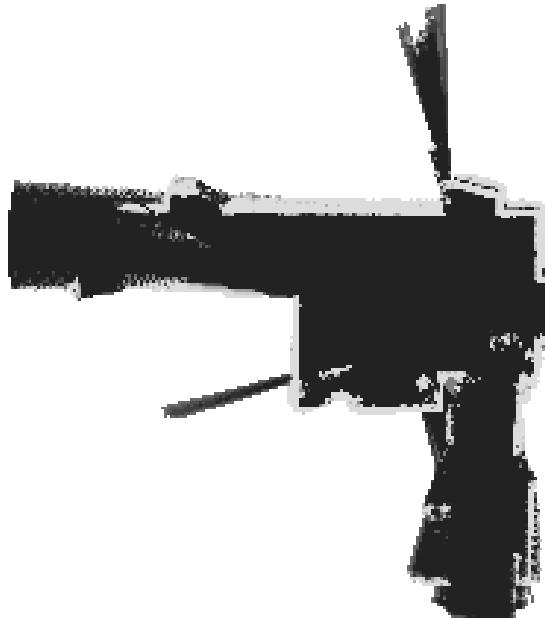


Abb. 4.6: Erstellter Grundriss eines Flures in Gebäude 29 der OvGU, Stockwerk 3

Farbwerte der Bildpunkte beschreiben zudem, ob an einer Stelle ein Hindernis oder ein frei befahrbarer Raum vorliegt. In der Abbildung ist zu erkennen, dass die erstellte Karte teilweise verrauscht ist, Wände nicht gerade verlaufen oder offene Türen und umherlaufende Personen die Messungen verfälschen. Die Karte kann dafür nach der Erstellung von jedem beliebigen Bildbearbeitungsprogramm geöffnet und bearbeitet werden. Dabei ist darauf zu achten, dass die Auflösung nicht verändert wird, da ansonsten die Dimensionen nicht mehr übereinstimmen oder gegebenenfalls die XML-Datei angepasst werden muss. Nach der Bearbeitung der Bilddatei kann sie anschließend im Miracenter geöffnet werden. Mittels Mausklick in die Karte wird die ungefähre Position des Roboters bestimmt. Danach orientiert sich der Scitos G5 automatisch und lokalisiert sich im weiteren Verlauf selbst anhand seiner Sensordaten. Nun können ihm Positionen und Ausrichtung vorgegeben werden und durch seine interne Pfadplanung steuert er sie autonom an. Durch ein Miracenter-ROS-Interface kann dabei die interne Lokalisierung und die Vorgabe von Position und Ausrichtung vom Scitos extern übermittelt werden. Die gesamte Software ist jedoch proprietär, d.h. nicht quelloffen, sodass ein tieferer Blick in die Funktionsweise der Software hier verwehrt wird.

#### 4.2.3 Android-Studio, Estimote SDK und ROSjava

Als letzter Baustein in der Software-Architektur fehlt die App für das Moto G, deren Entwicklung auf dem Zusammenwirken dreier Grundpfeiler aufgebaut sein wird:

1. Das Android-Studio [17] ist eine Entwicklungsumgebung für Programme speziell von Android-Smartphones. Es bietet die Möglichkeit, eine Applikation basierend auf der Programmiersprache Java auf ein Android-Gerät zu portieren und dort auch zu testen. Aufgrund der Implementierung der „Integrated Development Environment“ (IDE) in Java, lässt es sich auf beinahe jedem Betriebssystem verwenden und bietet dadurch eine hohe Flexibilität in der Anwendung.
2. Damit die Applikation die Beacon-Signale verarbeiten kann, müssen die gesendeten Nachrichtenpakete und das verwendete BLE-Protokoll in das Programm eingebettet werden. Vom Hersteller der Beacons wird dafür eigens eine Bibliothek unter dem Namen „Estimote Software Development Kit (SDK)“ [15] bereitgestellt, die sich problemlos integrieren lässt und einfach zu bedienen ist.
3. Um die empfangenen Signale an den Server via P2P zu senden, muss eigens eine Schnittstelle von der Java-basierten Android-App zum in C/C++ gehaltenen ROS implementiert werden. Dafür wird das Paket „Rosjava“ [35] benötigt, um die Unterstützung von ROS-Funktionen für Java-Programme zu realisieren. Dies

erlaubt es, ROS-Pakete in eine Android-App zu integrieren und dadurch die zu übermittelnden Nachrichten-Formate darin auch zu verwenden.

## 4.3 App-Entwicklung

Um das geplante Programm auf dem Smartphone zu verwirklichen, werden die aus 4.2.3 beschriebenen Hilfen und zudem ein eigener Programmcode benötigt. In diesem Abschnitt werden die einzelnen Elemente der App erläutert und anschließend ihr Zusammenspiel in der fertigen Applikation betrachtet.

### 4.3.1 Beacon-Detektierung

Der kritischste, aber auch der am essentiell wichtigste Programmteil ist die Aufnahme von Beacon-Signalen. Hierfür muss die SDK von Estimote in die App über das Android-Studio als Bibliothek importiert werden. Daraufhin stehen neue Klassen und Objekte der IDE zur Verfügung, die speziell auf die Verwendung von Beacons zugeschnitten sind. Zum einen muss dabei die App bei jedem Start für die Nutzung mit Beacons mit Abfragen neu initialisiert werden, das wären zum Beispiel die Überprüfung, ob Bluetooth angeschaltet ist oder ob BLE überhaupt auf dem Gerät unterstützt wird. Des Weiteren muss der sogenannte Beacon-Manager, welcher im Hintergrund der App läuft, mit Zeiten für die Pause zwischen zwei Abtastungen und der Länge eines Scans eingestellt werden. Die Problematik der Abstastraten wird hierbei später im Programmablaufplan noch erörtert. Zum anderen muss eine Empfangsroutine geschrieben werden, die bis zur Beendigung des Programms auf die Meldungen der Leuchtfeuer wartet und diese für die Verarbeitung im Programm aufbereitet. Die Anwendung von Filtern oder dergleichen entfällt hier, da schließlich das Verhalten der Signalausbreitung studiert werden soll und zusätzliche Einflüsse die Ergebnisse verfälschen könnten.

### 4.3.2 Lagemessung

Parallel zur Beacon-Detektierung soll gleichzeitig die Lage des Smartphones gemessen werden, um die Antennen-Charakteristiken besser zu verstehen und ihre Einflüsse auf die Empfangsqualität zu untersuchen. Für diese Aufgabe müssen die Inertialsensoren oder auch englisch „Inertial Measurement Unit“ (IMU) aus Gyroskop, Beschleunigungssensor und Magnetometer vom Smartphone ausgelesen werden, was jedoch einfach umzusetzen ist, da die IDE nativ dafür Bibliotheken bereitstellt.

Bei der Messung stellt sich hierbei die Frage nach der Häufigkeit der Sensordaten-abfragen, denn letztendlich senden die Beacons – zwar asynchron – ihre Signale alle 2 s bis 50 ms, letztlich können die Inertialsensoren in Intervallen von mehreren hundert Hertz ausgelesen werden. Aufgrund der höheren Verfügbarkeit der IMU-Daten, wäre es jedoch nicht zweckmäßig diese permanent über das Netzwerk zu senden und somit das Datenvolumen unnötigerweise künstlich aufzublähen. Es reicht daher völlig, zu jedem Zeitpunkt an dem eine Messung nach Beacon-Signalen definiert ist, eine Lageposition vom Smartphone vorrätig zu haben. Hier lässt sich dabei durch die schnelleren Messungen die Qualität der Informationen mit einem Filter erhöhen. Im Hinblick auf noch genügend freie Rechenkapazitäten auf dem Smartphone (bisher lediglich Abfragen der Sensoren), wurde ein Komplementärfilter auch im Hinblick auf die Sensordaten-fusion von den drei IMU-Sensoren gewählt. Der dazu nötige Quellcode wurde aus [33] entnommen, der im Grunde den Drift des Gyroskops und das Signalrauschen des Beschleunigungssensors und des Magnetometers entfernt. Der Aufbau ist diesbezüglich in Abbildung 4.7 einmal skizziert. Die gesamte Funktionsweise beruht auf dem Prinzip der Fusion mehrerer Daten, um die Qualität der eigentlichen Information zu erhöhen. Es würde für den Zweck der Orientierungserfassung des Smartphones ausreichen, die Information vom Schwerkraft-Vektor des Beschleunigungssensors mit der des Magnetometers zusammenzufassen und lediglich diese auszugeben. Jedoch sind besonders die Daten des Magnetometers sehr verrauscht, sodass ein Tiefpass oder anders ausgedrückt ein Mittelwert aus den Daten gebildet wird. Da das Moto G zusätzlich über ein Gyro-skop verfügt, welches die Bewegungen des Smartphones viel genauer messen kann, wird auch diese Informationsquelle benötigt. Aber auch hier existiert ein Nachteil in dessen Anwendung. Und zwar das Driften oder auch das Aufsummieren/Integrieren der Fehler oder des Messrauschens über die Zeit. Um auch diesem Problem zu be-gegnen, werden die Daten nur über kleine Zeitabschnitte mithilfe eines Hochpasses gesammelt. Dadurch werden anhand der Daten aus Accelerometer und Magnetometer als Stützinformationen und die Daten vom Gyroskop als Informationen über schnelle Änderungen verwendet.

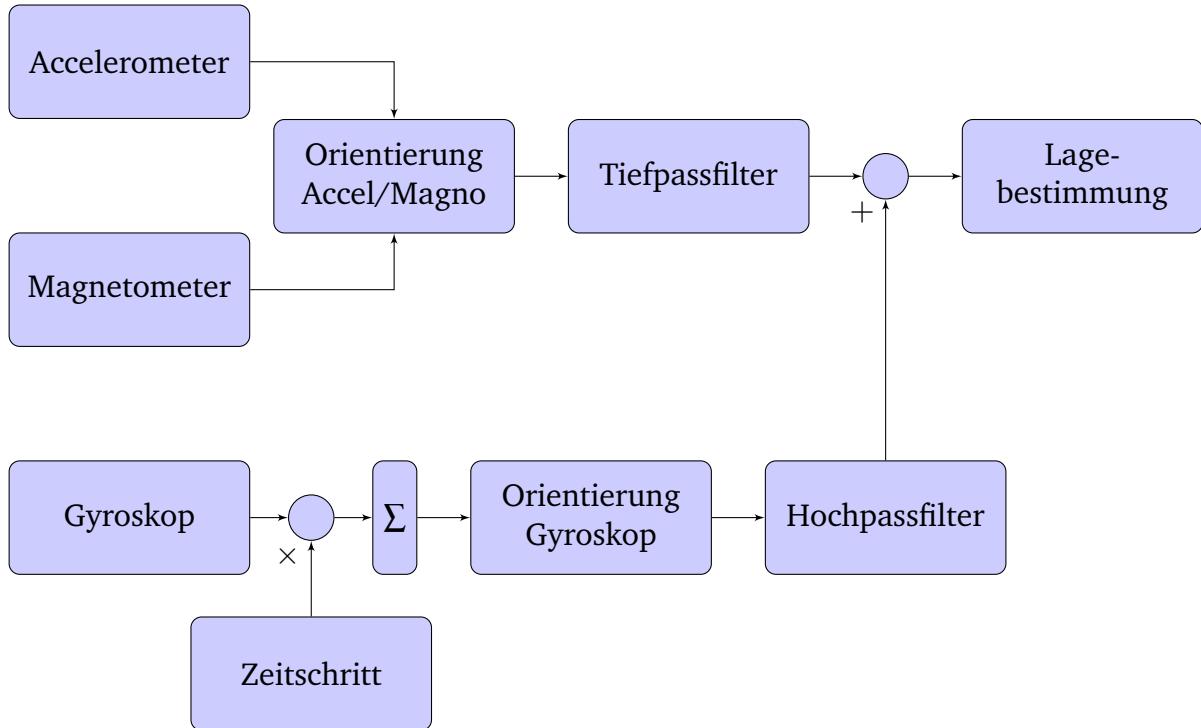


Abb. 4.7: Aufbau des Komplementärfilters, in Anlehnung an [33]

### 4.3.3 ROS-Anbindung

Die Realisierung der Datenübertragung über eine ROS-Schnittstelle in der App wurde so umgesetzt, dass nach dem Start der Applikation ein Feld zur manuellen Eingabe der Adresse des Servers bzw. des ROS-Masters (z.B. die Internetprotokoll [IP]-Adresse) eingegeben werden muss. Dies ist nötig, um einen Zielort für die Nachrichten zu bestimmen. Nach der manuellen Eingabe wartet der sogenannte „Publisher“ – also der Programmteil, der die Nachrichten aufbereitet und an das Netzwerk übermittelt – auf die Messdaten. Dabei muss für den Publisher eine Bibliothek bereitstehen, die die Anzahl der zu übertragenden Informationen und deren Datentypen definiert und in Java integrieren kann, um sie später in der App zu verwenden. Dazu muss wie anfänglich erwähnt die ROSjava-Erweiterung installiert sein und anschließend das Message-Format als Java-Paket damit erstellt werden. Da die versendeten Daten schon in vorigen Abschnitten erläutert wurden, wird in Abbildung 4.8 nur noch eine Zusammenfassung in Form der Implementierung präsentiert. Das dargestellte Message-Format enthält dabei die UUID empfangener Beacons als Zeichenkette oder String-Wert, sowie der empfangenen Signalstärke RSSI und der eingestellten Sendeleistung (bezeichnet als Power), als 32 Bit Integer und der daraus errechneten Distanz als 64 Bit Fließkommazahl (ein zusätzliches Feature der Estimote SDK). Des Weiteren beinhaltet sie die errechnete Orientierung vom Moto G und darüber hinaus den Zeitstempel der Zusammenstellung der

Nachricht. Natürlich existiert zwischen der Aufnahme von Messungen und dem Sendezeitpunkt eine zeitliche Differenz (kleiner als 1 ms Bearbeitungszeit), die jedoch in den statischen Experimenten zu vernachlässigen ist.

```
string UUID
int32[] RSSI
int32[] Power
float64[] Distance
float64[] Orientation
float64[] Time
```

Abb. 4.8: Quellcode-Beispiel eines Message-Paketes für eine ROSjava-Implementierung

#### 4.3.4 Programmablauf

Nachdem die einzelnen Module fertig gestellt wurden, musste aus ihnen ein gesamtes Programm zusammengefügt werden. Bei der Vereinigung der einzelnen Programmteile kristallisierten sich dabei verschiedene Probleme heraus, die unter anderem auch auf den verstärkten Einsatz von proprietärer Software zurückzuführen sind. Denn bei der notwendigen Verwendung von nicht quelloffener Software, können Fehlfunktionen der App, die eigentlich auf dem Versagen der genutzten Software beruhen, lediglich umgangen oder zumindest mit Einschränkungen im Ablauf der Applikation behoben werden. Im konkreten Fall verursachte die gleichzeitige Verwendung von WLAN und Bluetooth Programmabstürze, die nach unterschiedlichen Nutzungszeiten auftraten. Das Problem scheint hierbei primär beim Betriebssystem Android zu liegen, weil das Problem schon seit zwei Jahren in diversen Online-Foren bekannt ist und dort diskutiert wird. Der Fehler tritt jedoch auch weiterhin in den neuesten Versionen des Betriebssystems auf [37]. Die einfachste Umgehung des Konfliktes würde darin bestehen, wenn die Nutzung von Bluetooth durch die Estimote SDK und der Gebrauch der WLAN-Verbindung aufeinander abgestimmt würde. Jedoch bietet die SDK keine Einstellmöglichkeiten und keinerlei Dokumentation über die Verwendung des Bluetooth-Adapters, sodass auch nicht auf Seiten der WLAN-Übertragung die Anpassung stattfinden kann. Da die Zeit fehlte, sich weder mit dem Entwicklerteam der Firma Estimote auseinander zu setzen oder sich in das Android-Betriebssystem einzuarbeiten, wurde eine notdürftige Lösung gefunden.

In Abbildung 4.9 ist die gesamte App als Ablaufplan mit Blockschaltbildern dargestellt. Während die eigentliche Aufgabe der Applikation schon im Vorfeld erklärt wurde, fällt hier eine zusätzliche Kontrollfunktion auf. Diese ist im Grunde eine Sicherheitsabfrage

vor der Messung und der Sendung der Nachricht über den Publisher. Sie überprüft, ob eine Verbindung zum P2P-Netzwerk existiert und erst danach folgen die Abfragen und der Aufruf des Publishers. Falls keine Verbindung mehr besteht, versucht die Kontrollfunktion durch die Deaktivierung und der erneuten Aktivierung des WLAN-Adapters die Konnektivität zum Netzwerk wiederherzustellen. Solange keine Verbindung besteht, wechselt die App in einen Leerlaufmodus und nimmt dabei keine neuen Messwerte auf. Bei diesem Vorgang vergehen mehrere Sekunden, was in den statischen Versuchen keine Bedeutung hat und rein statistisch auch nur jede Stunde einmal vorkam. Im Hinblick auf die dynamischen Experimente ist dies problematischer, da dadurch Messungen an Standorten fehlen könnten und der Roboter gegebenenfalls erneut die Raumkoordinaten anfahren müsste. Das Problem könnte auch die kommerzielle Nutzung betreffen, wenn beispielsweise der Kunde aktuelle Informationen über seine Position über das Internet abrufen möchte und sich dafür mit einem WLAN-Netzwerk verbindet. Bei einem Verbindungsabbruch durch die gleichzeitige Nutzung von Bluetooth und WLAN ergäben sich für den Nutzer Wartezeiten, die er wahrscheinlich nicht tolerieren würde. Hier muss noch auf Seiten der Hersteller der Basis-Software (Estimote SDK und Android) nachgebessert werden.

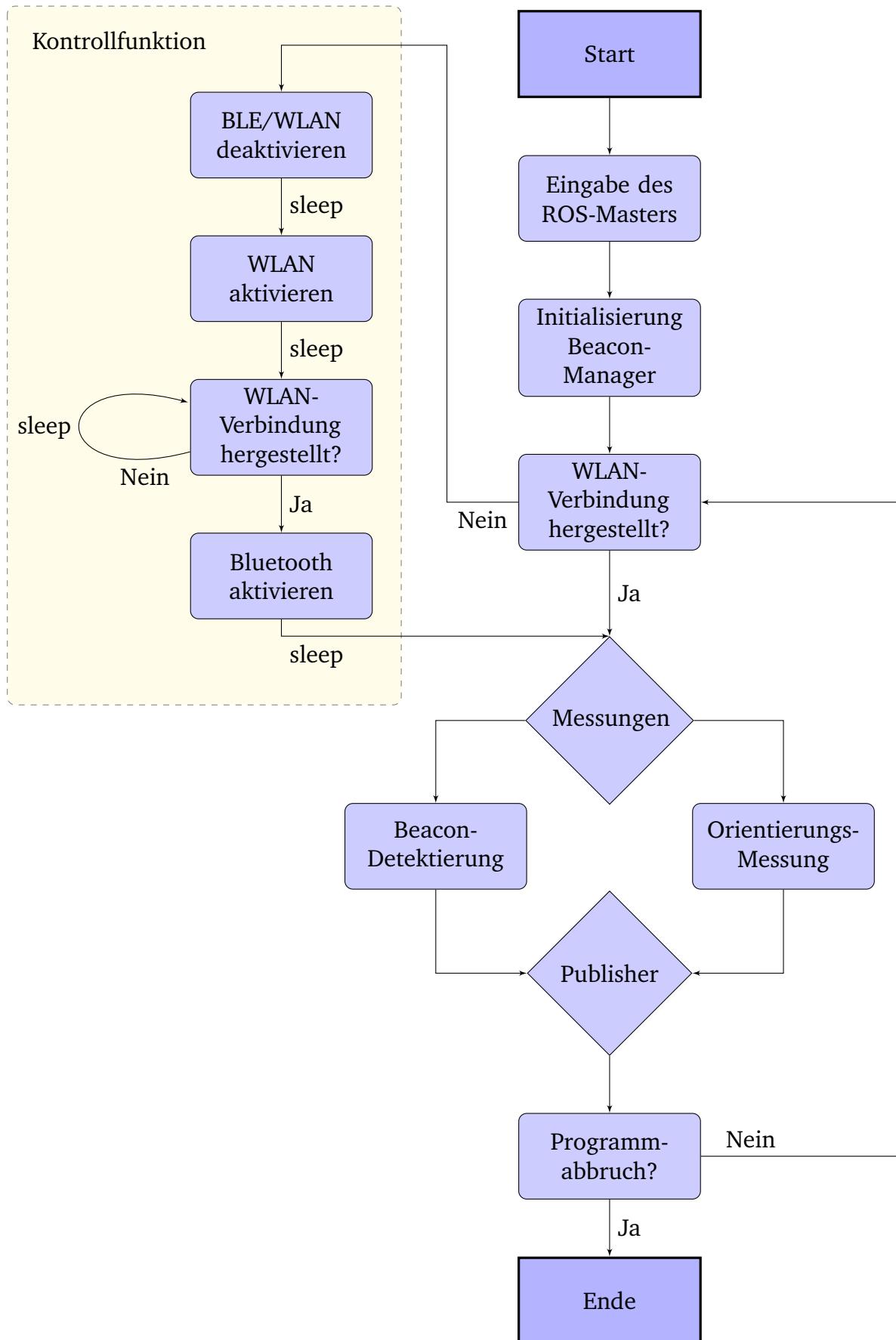


Abb. 4.9: Programmablaufplan der Lighthouse Keeper-Applikation für Android-Smartphones



# 5 Experimente

Die durchgeführten Experimente beziehen sich in diesem Teil der Arbeit auf die Messung der Signalstärke der Beacons in Abhängigkeit zu deren Entfernung. Diese Phase in der Prozessplanung legt den Grundstein für die Modellbildung, die anschließend im nächsten Kapitel vorgenommen wird. Ferner werden einzelne Einflüsse auf die Messungen näher betrachtet und eine Analyse zu der Batterielaufzeit eines Beacons in Abhängigkeit zu seinen Einstellungen aufgestellt. Die hier vorgenommenen Versuche betrachten speziell die Ausbreitung der Signale bei einer fest justierten, für die Arbeit vorher definierten Einstellung der Beacons. Des Weiteren werden auch Aspekte der Beacon-Technologie betrachtet, die für spätere Forschungen von Interesse wären, aber auch die Notwendigkeit dieser Arbeit unterstreichen sollen. Denn es macht gerade den Reiz der Beacons aus, dass sie relativ einfach an ihre Bestimmung angepasst werden können. Schließlich ergab sich nach der Aufnahme mehr als 30 Stunden Sensordaten-Material die Erkenntnis, dass das gesamte Thema weit umfangreicher ist und die möglichen Modi genauere Untersuchungen verdienen. Gerade deswegen soll dies nach der ursprünglichen Aufgabe der Aufnahme von Messwerten für die Modellbildung nachträglich mit einigen Abschnitten über das Verhalten der Beacons gewürdigt werden.

## 5.1 Messung der BLE-Signalausbreitung

Für die Versuche wurde zunächst ein großer Raum benötigt, damit der Abstand eines Beacons zum Smartphone um mehr als 15 Meter variiert werden konnte. Das Equipment für die statischen Experimente bestand aus dem Youbot, drei Beacons, dem Moto G und einem Laptop mit installiertem Linux-Betriebssystem. In den Bildern ?? bis ?? im Anhang sind einige Fotos hinterlegt, die den Aufbau porträtieren. Die Messungen wurden dabei für jeden der drei Beacons für eine Distanz für jeweils zehn Minuten durchgeführt. Aufgrund der Vielzahl an Daten werden hier nur drei Messreihen exemplarisch in der Abbildung 5.1 für die Signalstärke -12 dBm und der Sende-Frequenz von 200 ms gezeigt. Es fällt dabei auf, dass die Empfangsfeldstärke mit zunehmender Distanz abnimmt, da das Signal durch das Medium Luft einen Widerstand überwinden muss. Auf den ersten Blick nehmen auch die Abweichungen bzw. das Rauschverhalten zu, jedoch ist zu bedenken, dass der Leistungspegel Dezibel Milliwatt (dBm) logarithmischer Natur ist. D.h. in der ersten Messung in der Distanz von einem Meter variiert die gemessene Signalstärke um 2 dBm und im Abstand von 10 Metern um 6 dBm. Da die Differen-

zen in einem anderen Wertebereich liegen, beträgt die Leistungsschwankung der Stärke bei einem Meter umgerechnet rund  $0,5 \cdot 10^{-7}$  mW, während sie bei 10 Metern mit nur rund  $0,5 \cdot 10^{-8}$  mW in Erscheinung tritt. Wird dies wiederum in Relation gesetzt und die Abweichungen prozentual beschrieben, stimmen die anfänglichen Aussagen wieder überein und es kann von einem stärkeren Rauschen bei größeren Distanzen gesprochen werden.

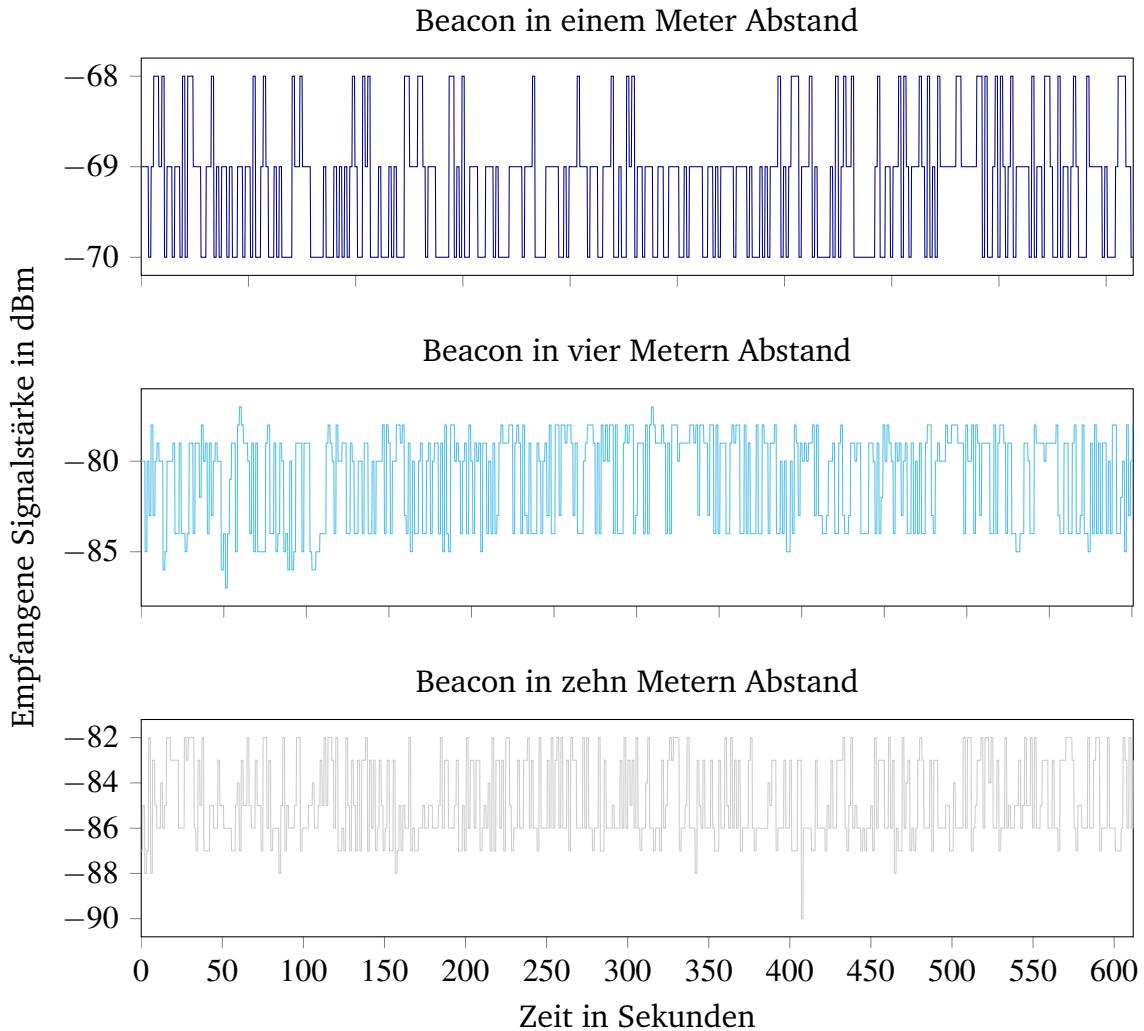


Abb. 5.1: Verlauf der Signalstärke aus einer Entfernung von 1, 4 und 10 Metern

Für die ersten vier Distanz-Meter ist diese Aussage über alle Messreihen die Regel, für darauffolgende größere Entfernungen ereignet es sich jedoch, dass sich die Signalstärke wieder erhöht und sich das Rauschen vermindert. Dies ist in Abbildung 5.2 einmal dargestellt. Besonders deutlich wird die Missachtung der Regel zwischen dem zehnten und dem fünfzehnten Meter, wo der Verlauf der Signalstärke eher einer Achterbahnfahrt gleicht. Der Grund dafür könnte die „Verschmutzung“ des Raumes durch andere Signa-

le im 2,4 GHz Band sein oder auch durch Reflektionen der Wände, der Decke oder des Fußbodens hervorgerufen werden. Aufgrund der schon erwähnten Asynchronität der Beacon-Signale, der nicht einsehbaren Messzyklen-Dauer der Estimote SDK und/oder auch der parallelen Nutzung von Bluetooth- und WLAN-Modul des Smartphones ergab sich anfänglich der Verdacht, dass die Ursache für die Signalstörungen in der Messwertaufnahme mithilfe der Lighthouse Keeper-App selber liegt. Somit wäre es denkbar, dass die stärksten Signale zufällig nicht aufgenommen und nur deren „Echo“ wahrgenommen wird, weil ihnen zu dem Zeitpunkt als die Signale die Antennen vom Smartphone passierten niemand „zuhörte“. Dieser Verdacht sieht sich im Nachhinein nicht bestätigt, wenn nur Abbildung 5.1 in dem Abstand von einem Meter betrachtet wird. Hier ist der Abstand zwischen Smartphone und Beacon sehr gering, nichtsdestotrotz werden keine reflektierten Signale aus größerer Entfernung aufgezeichnet, weder in dieser Messung noch in weiteren in kurzer Entfernung. Also muss die Verstärkung des Rauschens auf die physikalischen, statt den programmiertechnischen Erklärungen zurückzuführen sein. In dieser ersten Betrachtung des Systems kann schon von vornherein gesagt werden, dass das Rauschverhalten bzw. das Auftreten von Störungen raumabhängig und somit nur schwer vorhersagbar ist.

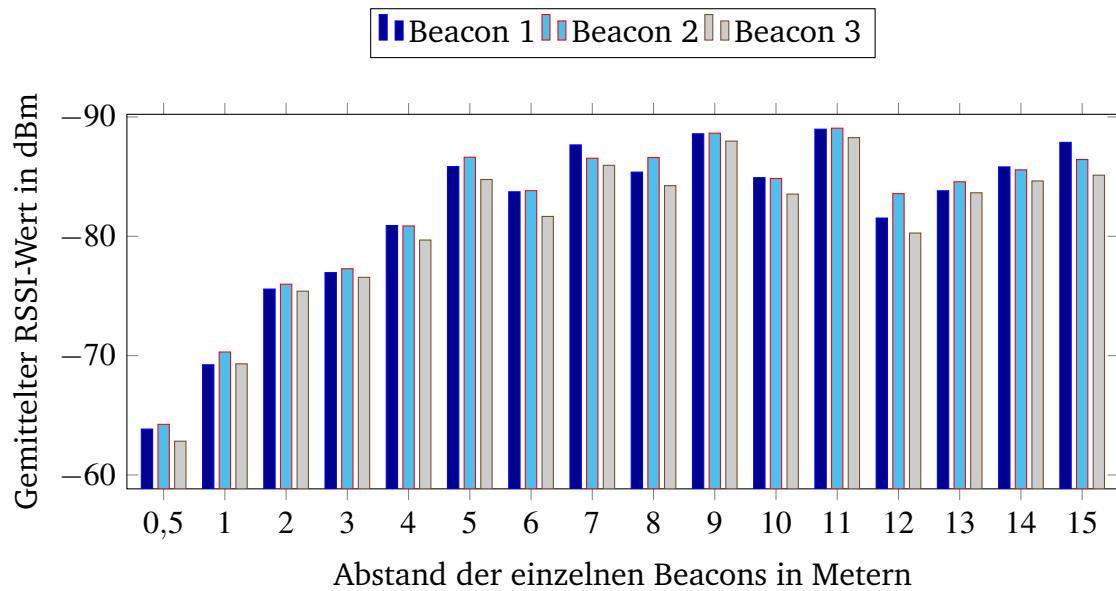


Abb. 5.2: Mittelwert der RSSI-Werte von allen Distanzen

Um nun doch wie erwünscht eine qualitativ hochwertige Schätzung über die Signalausbreitung von BLE-Signalen zu gewinnen, muss es zum Anfang ein Modell geben, das zuerst von einem völlig idealen Verhalten ausgeht. Es muss dabei die Realität nur im Groben abbilden können und ausschließlich den Grundstein für die Simulation und deren Erweiterungen legen. Um eine Theorie dafür aufstellen zu können, müssen die

gesammelten Daten genauer betrachtet und die Ausbreitung des BLE-Signals der Beacons ohne Störungen herausgefiltert werden. In Abbildung 5.2 wurden schon einmal alle Messreihen in einem Diagramm zusammengefasst. Dabei wurde von jeder Messung eines jeden Beacons der Mittelwert der empfangenen Empfangsstärke über der Distanz zwischen Smartphone und Leuchtfeuer aufgetragen. Auf den ersten Blick fällt auf, dass lediglich bis vier Meter eine genaue Aussage über die Entfernung getroffen werden kann. Die Unterschiede bzw. das Fehlen eines Unterschiedes macht es für größere Entferungen schwer anhand dessen eine qualitative Berechnung durchzuführen. Deswegen wird der Anteil an darstellbaren Informationen in Abbildung 5.4 und 5.5 erhöht, indem nicht der Mittelwert aller Messungen, sondern die Verteilung der empfangenen Signalstärke farblich über die Entfernen markiert wurde. Hier erscheint verstärkt das Phänomen, dass das Rauschen nicht mit der Distanz korreliert, sondern stark ortszabhängig ist. Zumindest erhöht sich mit einem größeren Luftweg die Wahrscheinlichkeit, dass dies auftritt. Für die Parameterbestimmung des Modells ist es jedoch wichtig, dass die allgemeine Theorie auch von den Messwerten erfüllt wird. Insbesondere bei der Signalausbreitung wird davon ausgegangen, dass die Signalstärke über den zurückgelegten Weg abnimmt. Ist dies nicht der Fall, muss eine neue Theorie (mit Reflexionen, Signalauslöschung etc.) gefunden oder die Daten müssen so aufbereitet werden, dass die Theorie wieder zutrifft. In Folge letzterer Aussage werden statt aller Messwerte einer Entfernung nur die 100 stärksten zur Mittelwertbildung herangezogen, wie es in Abbildung 5.3 gezeigt wird, sieht die Entwicklung der Abnahme der Signalstärke deutlich vorhersehbarer aus und wird dadurch verwertbar.

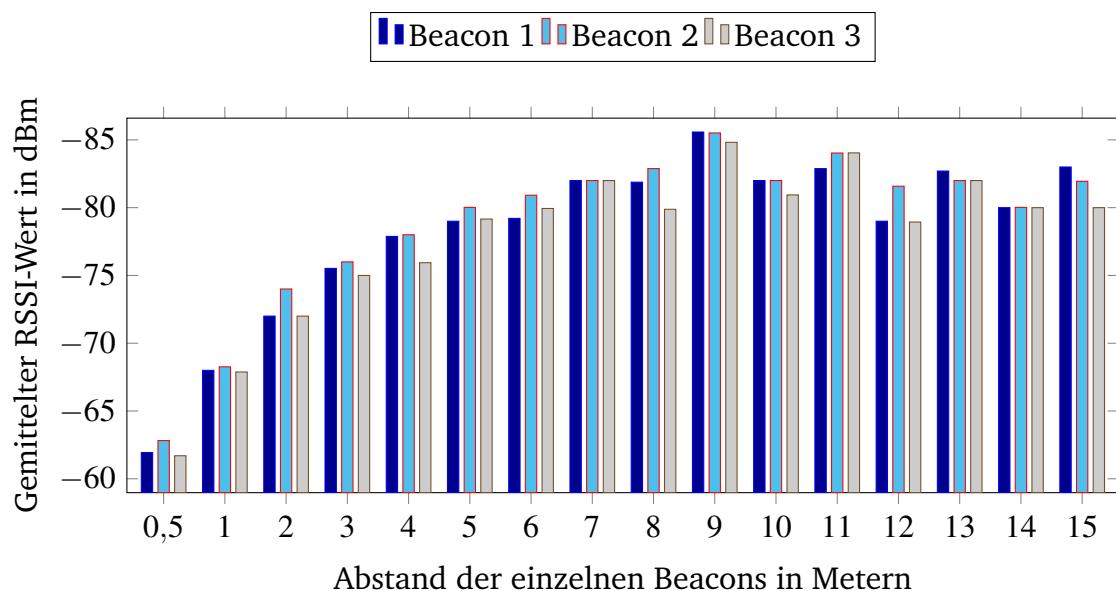


Abb. 5.3: Mittelwert der 100 stärksten RSSI-Werte von allen Distanzen

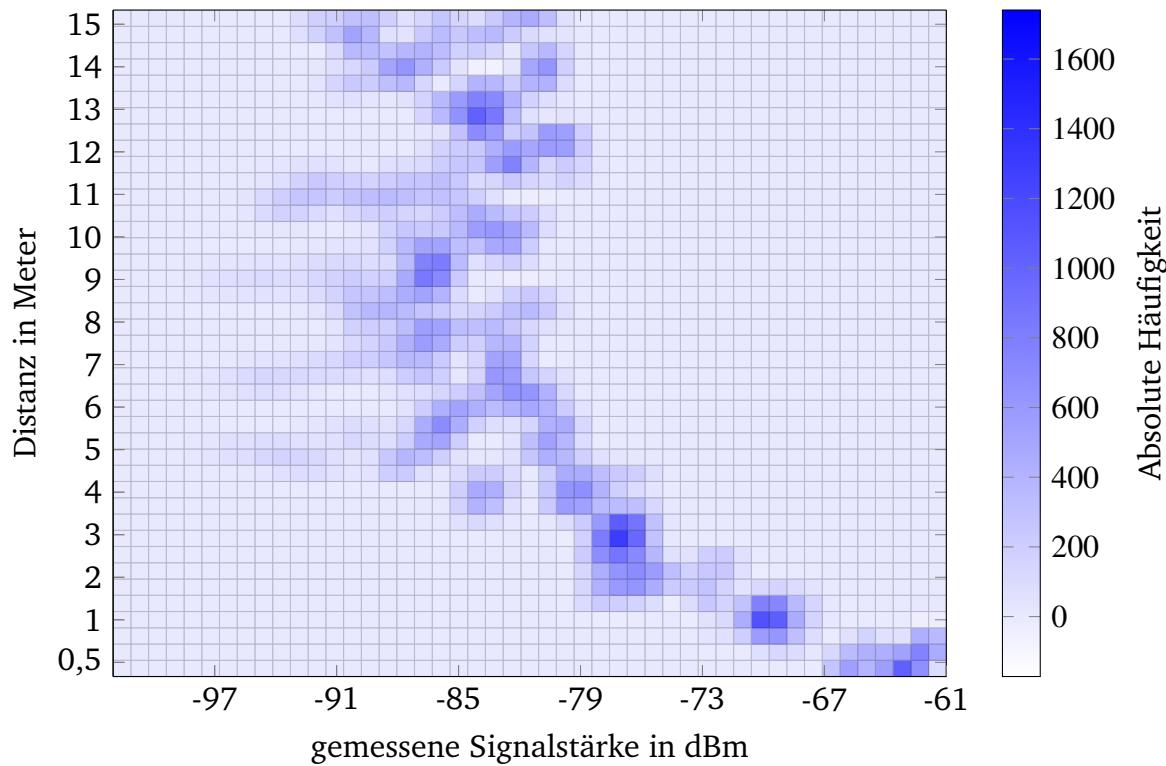


Abb. 5.4: Top-View auf die Signalstärken-Verteilung zu den einzelnen Distanzen

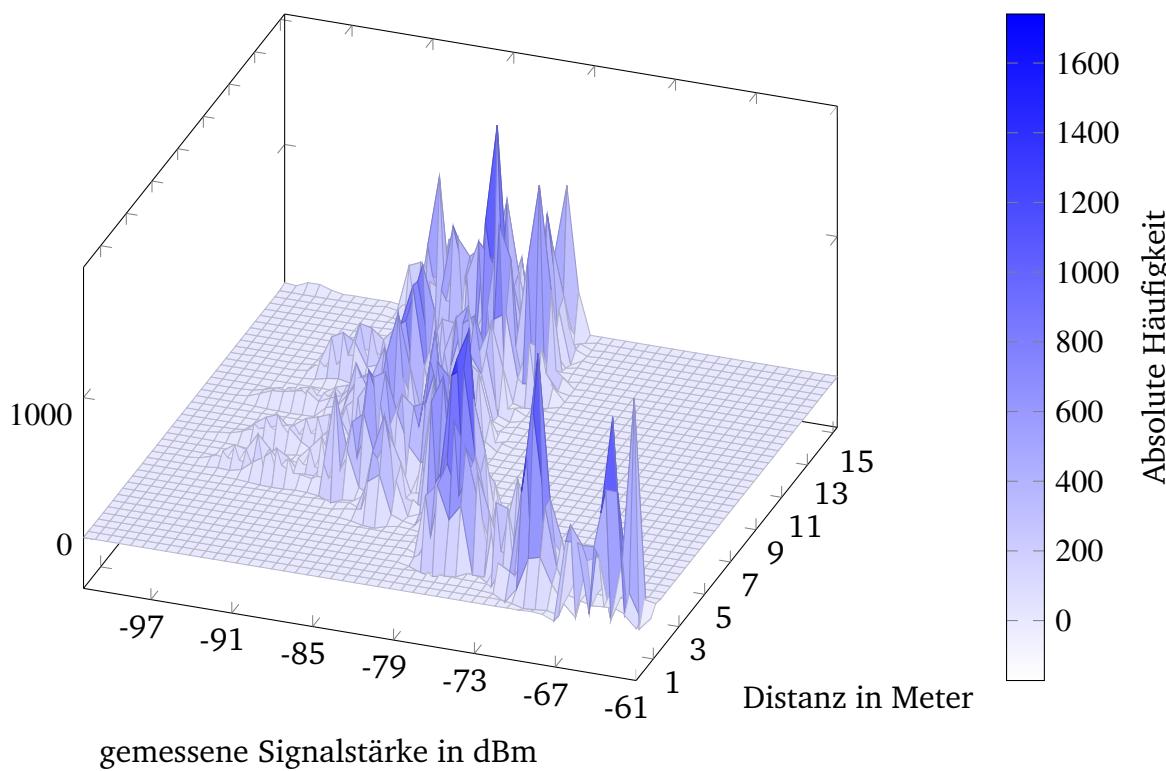


Abb. 5.5: Seitliche Ansicht auf die Signalstärken-Verteilung zu den einzelnen Distanzen

## 5.2 Evaluierung der SDK-eigenen Distanzschätzung

In dem Abschnitt über ROSjava wurde in Abb. 4.8 eine von der SDK berechneten Distanz erwähnt. Diese ist jedoch nur eingeschränkt nutzbar, wie es Abbildung 5.6 belegt.

Hier wurden der optimale Verlauf (geschätzter Wert gleich der tatsächlichen Distanz) mit denen aus den 100 stärksten Signalen geschätzten Entfernung der SDK gegenübergestellt und zueinander aufgetragen. Dazu wurde der optimale Verlauf eingetragen, bei dem die geschätzten Werte gleich den tatsächlichen entsprechen. Das Ergebnis der SDK-internen Berechnung spiegelt dabei in keiner Weise die Realität wider und verlangt die Entwicklung eines neuen Modells, da mit diesem Resultat kein Indoor-Lokalisierungssystem mittels Trilateration oder Landmarken betrieben werden kann.

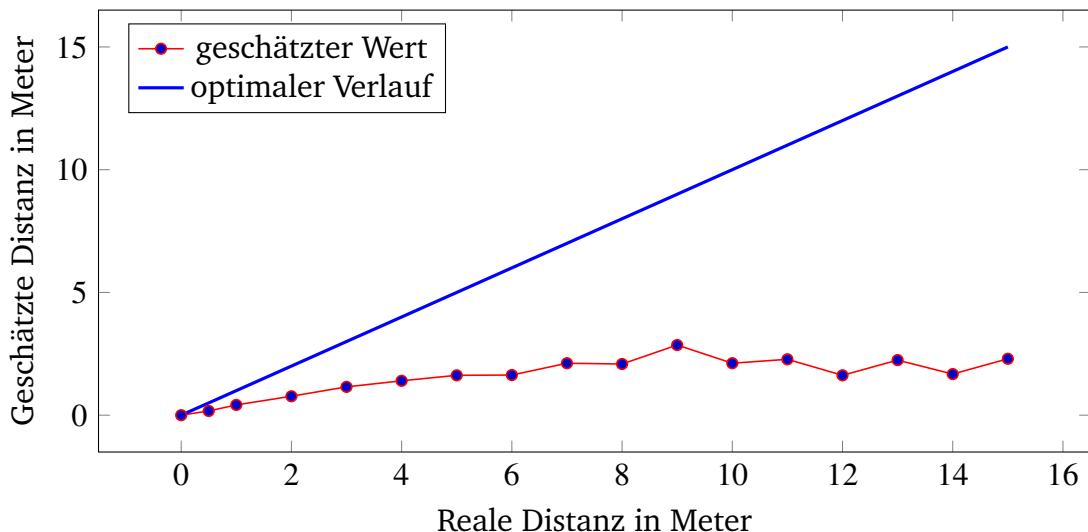


Abb. 5.6: Vergleich der durchschnittlichen Distanz-Schätzung zur realen Entfernung

## 5.3 Auswirkung der Smartphone-Orientierung

Im Zuge der vorangegangenen Untersuchungen und unter der Vermutung des Einflusses der Antennen-Charakteristiken vom Moto G stellt sich die Frage, wie sehr die Lage des Smartphones die Messungen beeinträchtigt. Schließlich ist der Empfänger – also das Moto G – ständig in Bewegung und befindet sich fortlaufend in neuen Positionen und Ausrichtungen zu den Standorten der Beacons. Dies gilt natürlich nicht nur für die statischen Versuche in diesem Kapitel, sondern soll im Hinblick auf die dynamische Validierung mit dem Scitos G5 und auch später für die reale Anwendung erforscht werden. Es soll hierbei die Aussagekraft der ausgeführten Experimente und zukünftigen Tests überprüft werden, die die Grundpfeiler der Modellbildung und später der Simulation bilden.

In diesem Sinne gilt es, die Abhängigkeit der gemessenen Empfangsstärke des Signals zu der möglichen Orientierung des Moto G zur Signalquelle zu ermitteln. Um den Aufwand der Messungen so gering wie möglich zu halten, wurden die drei verfügbaren Beacons so um das Smartphone herum positioniert, dass sie sich quasi auf den drei Raumachsen mit dem Moto G im Koordinatenursprung befinden (siehe Abbildung 5.7). Weil der Youbot seinen Roboterarm nur in Z-Richtung drehen konnte, ohne dabei die Abstände zu den Beacons zu ändern, wurden einige Konstellationen nicht betrachtet. Es soll hierbei jedoch nicht darum gehen, wie das Smartphone am besten zu halten sei, sondern ob die Intensität der Signale ausgehend von der Decke, von den Wänden, von vorn oder von der Seite anders wahrgenommen wird. Die Ergebnisse finden sich in der Abbildung 5.8 wieder. Dort sind zum einen die gemessene Empfangsstärke der Beacons in festen Abständen zum Moto G zu sehen und dazu gegenüber die Ausrichtung an der Drehachse in Z-Richtung. Es fällt auf, dass egal aus welcher Richtung die Signale auf das Smartphone treffen, der Empfang durch die Drehung gleichermaßen und zum gleichen Zeitpunkt beeinträchtigt wird. Das Ergebnis überrascht insofern, dass eigentlich die Änderungen der erhaltenen Signalstärken zu jedem Drehwinkel die gleichen waren. Es gab keine Unterschiede in der Intensität der Änderungen, obwohl die Beacons alle in unterschiedlichen Winkeln zu dem Smartphone und dadurch zu dessen Antennen standen. Somit erklärt sich dieses Verhalten nicht anhand der Argumentation, dass eine Antenne bestimmte Charakteristiken aufweist und je nach Winkel der die Stärke des Signals ab- oder zunimmt. Um den Einfluss dieses Phänomens auf die statischen Versuche genauer abzuschätzen, wurden nachfolgend die Messungen aus 5.1 noch einmal wiederholt und lediglich die Drehung um besagte Z-Achse hinzugefügt (siehe Bild ?? im Anhang). Die Ergebnisse in der Abbildung verwundern allerdings genauso, denn die fast proportionale Änderung von Ausrichtung zur Stärke der Signalaufnahme ist relativ leicht zu erfassen und sicherlich auch gut zu filtern, doch mit dem Verständnis wie Antennen funktionieren und der bekannten Theorien lässt sich dieses Verhalten nicht erklären. Die einzige Erkenntnis aus den hier durchgeführten Experimenten ist, dass die Vermutung der höchsten Empfangsgüte der Antennen bei normaler Halterung des Smartphones in der flachen Hand sich bewahrheitete und die Richtung, aus der ein Signal stammt, dabei keinen Einfluss hat.

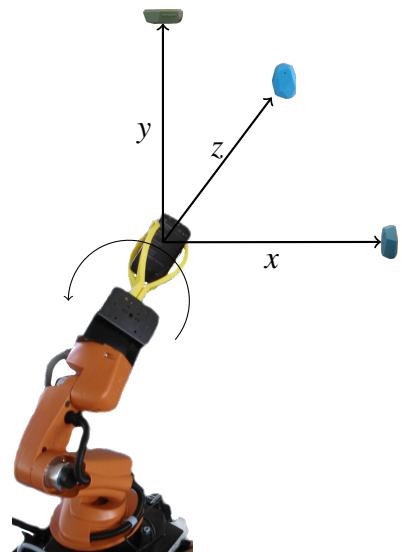


Abb. 5.7: Skizzierung der Anordnung der Beacons und der Drehung des Roboterarms

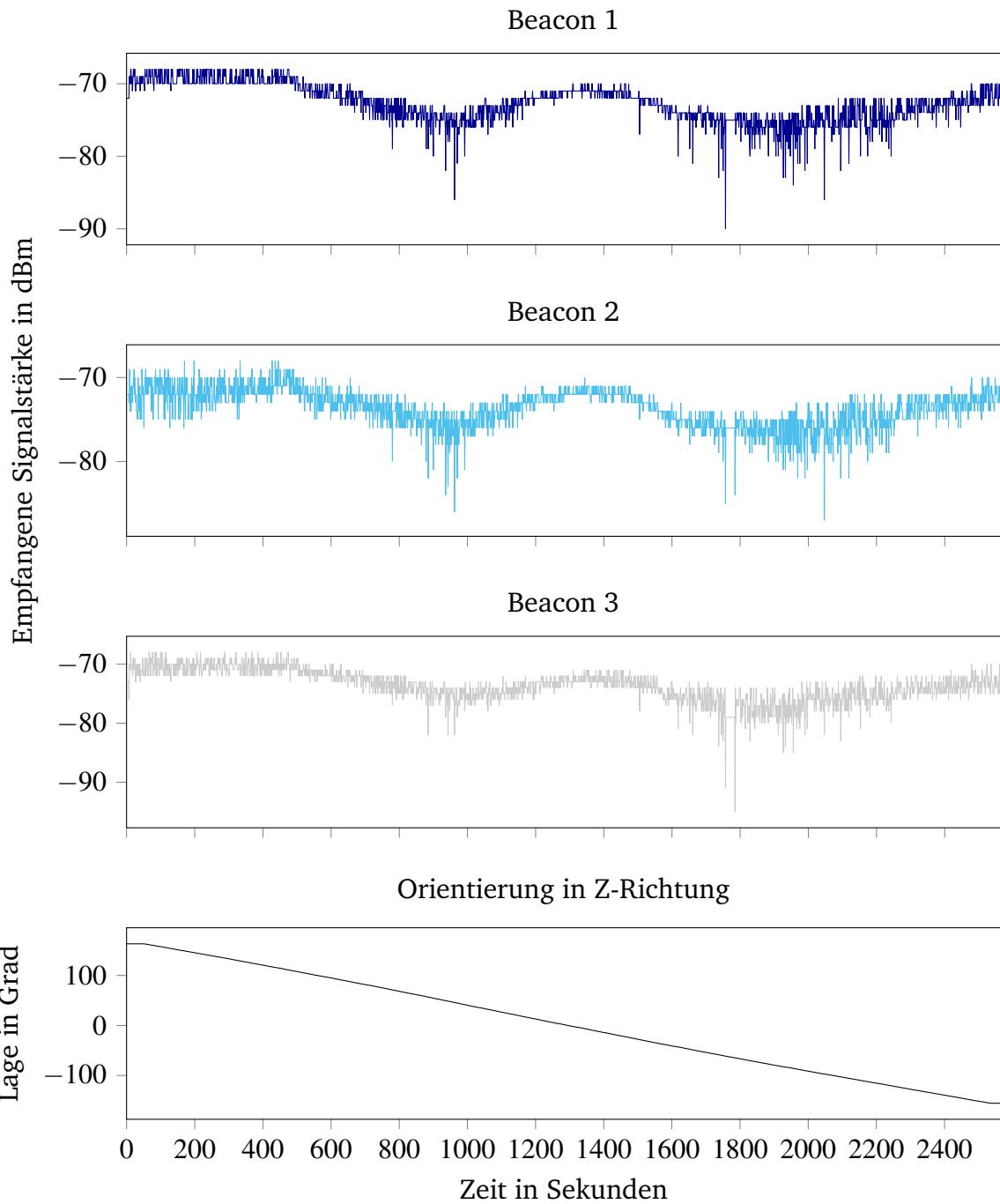


Abb. 5.8: Verlauf der gemessenen Signalstärke zu der Drehung des Smartphones in Z-Richtung

## 5.4 Wechselwirkungen zwischen Funkquellen

Unter dem Aspekt der angesprochenen „Verschmutzung“ eines Raumes durch andere Signalquellen, welche auch im 2,4 GHz Band senden, blieb ein Beweis dieser Behauptungen bisher aus. Der Beweis oder generell die Frage nach Wechselwirkungen mit

anderen Transmittern ist insoweit auch wichtig, wenn es sich um einen Sicherheitsabstand zwischen zwei Beacons handelt, bevor sie anfangen sich gegenseitig zu stören. In Bezug auf die generelle Störung von Bluetooth-Signalen durch beispielsweise WLAN-Geräte sind die Auswirkungen weitgehend bekannt und auch hinreichend erforscht (z.B. [30]). Jedoch bleibt die Frage, ob sich Beacons untereinander beeinflussen und wenn ja, wie groß der Mindestabstand gewählt werden muss, sodass die Auswirkungen vernachlässigbar sind. Zur Beantwortung wurden die drei verfügbaren Leuchtfeuer relativ nah zueinander befestigt (siehe Abbildung 5.9) und zu den gleichen Distanzen wie in Abb. 5.1 die Signalstärken aufgenommen. In Abbildung 5.10 sind die Ergebnisse aus einem Abstand von 100 cm der Beacons zueinander dargestellt. Im Vergleich zu den Messungen aus Abb. 5.1 werden Unterschiede in der empfangenen Signalstärke schnell sichtbar. Das Signal aus der Beacon-Ansammlung ist bei gleicher Distanz eines Beacons zum Smartphone mitunter 5 bis 10 dBm schwächer und deutlich verrauschter (siehe Tabelle 5.1) als das Signal von einem alleinstehenden Beacon. Dies beweist somit die grundsätzliche negative Beeinflussung von nah beieinander liegenden Beacons auf deren Sendequalität. In Folge weiterer Untersuchungen ergab sich ein empfohlener Mindestabstand von 50 cm, durch den keine messbaren Verschlechterungen in der Signalstärke mehr vorkamen. An dieser Stelle ist anzumerken, dass hier generell mehr nicht gleich besser bedeutet. Im Hinblick auf die Planung eines Indoor-Lokalisierungssystems muss diese Betrachtung in den Gestaltungsprozess mithilfe einer Simulation einfließen, sodass der Sicherheitsabstand gewahrt bleibt.



Abb. 5.9: Gleichzeitige Messung dreier naheliegender Beacon-Signale

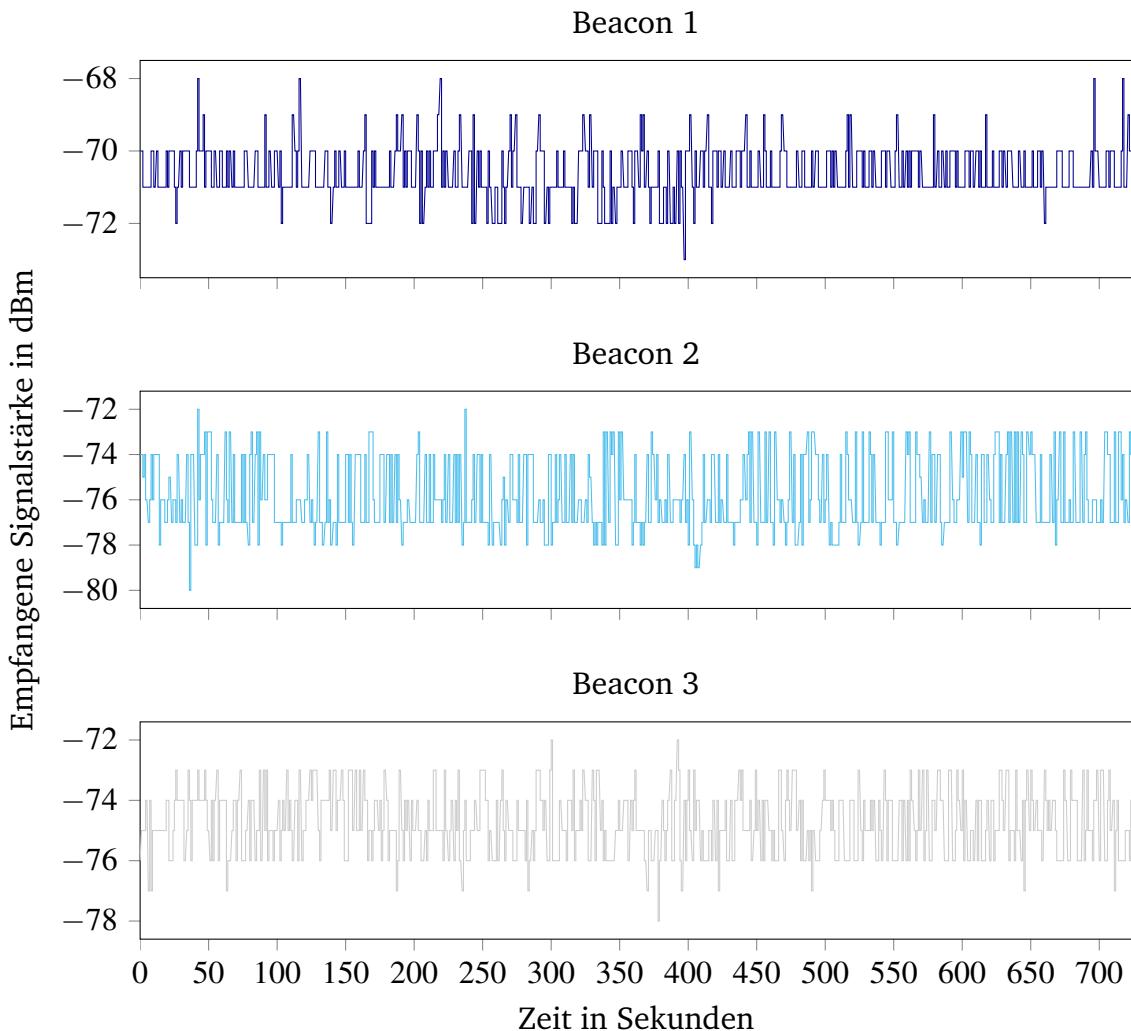


Abb. 5.10: Verlauf der Signalstärke dreier nah nebeneinander liegender Beacons aus einer Entfernung von einem Meter

		Standardabweichung		
		Beacon 1	Beacon 2	Beacon 3
Messung	aus 5.1	0,6946	0,8950	0,7822
	aus 5.10	0,7316	1,6508	1,0937

Tabelle 5.1: Standardabweichung der Messungen aus Abb. 5.1 und aus Abb. 5.10

## 5.5 Energieverbrauch eines Beacons

Der Wartungsaufwand für ein Sender hängt maßgeblich an der Laufzeit der in jedem Beacon eingebauten Batterie ab. Die in Kapitel 1 und 2 erwähnten Einstellmöglichkeiten von Sendeleistung und der Häufigkeit von Sendeintervallen beeinflussen dabei den Stromverbrauch der Beacons und somit auch die Betriebsdauer der Batterien. Dieser

Abschnitt soll sich damit beschäftigen, inwieweit sich die Parameter auf die Wartungszyklen auswirken. Dies soll später dabei helfen, zwischen der Auslegung der beiden Parameter und des gewünschten Wartungsaufwandes abwägen zu können. Denn in großen Projekten mit mehreren hundert Beacons steigt der Wartungsaufwand proportional und wenn einzelne Beacons andere Einstellungen als die Mehrheit aufweisen (um beispielsweise wichtige Gebiete besser abzudecken) wird auch der Überblick über die nötigen Wartungszyklen verloren gehen. Um die Abhängigkeiten zu veranschaulichen, wird anhand vom Datenblatt des verwendeten Hardware-Chips nRF51822 [31] in den genutzten Beacons eine Simulation entworfen, die die Einstellmöglichkeiten als Variablen betrachtet. Als Ausgangsbasis für den Energiespeicher wird eine CR2450-Batterie mit 1,8 Wh [38] angenommen. Hierbei wurde nicht vom Idealzustand ausgegangen, sondern ein üblicher Faktor von 0,7 hinzu multipliziert, um der Alterung der Zelle und weiterer Effekte Rechnung zu tragen. Durch die gering fließenden Ströme und der minimalen Bauweise der Beacons können auch keine direkten Messungen vorgenommen werden. Zudem kann auch nicht verifiziert werden, wie lange ein Sendeaufruf oder ein Zustand vom Prozessor dauert. Um trotzdem ein gutes Simulationsergebnis zu erreichen, muss tiefer in die Funktionsweise der Beacons geschaut werden. Als größte Unbekannte tritt dabei die Sendedauer auf. Bei einer angenommenen Datenrate von 250Kbit/s und bei einer Größe eines Datenpaketes von 38 Bytes [5] (siehe 5.11), dauert eine Sendesequenz rund 1 ms. Die benötigte elektrische Leistung für eine vorher definierte Sendeleistung beträgt dabei ein Vielfaches dessen und ist davon im hohen Maße nichtlinear abhängig. Die dazu nötigen Beziehungen können anhand des Datenblattes des nRF51822-Chips hergeleitet werden. In diesem Fall wurde ein Polynom 4. Grades als Funktion der einstellbaren Sendeleistung aufgestellt, welches annähernd diese Abhängigkeiten beschreibt. Da die Kommunikation bidirektional ist, existiert auch ein Empfangsmodus, der ebenfalls mit 1 ms Empfangsdauer angenommen wird, aber dessen Modus hingegen einen konstanten Stromverbrauch aufweist.

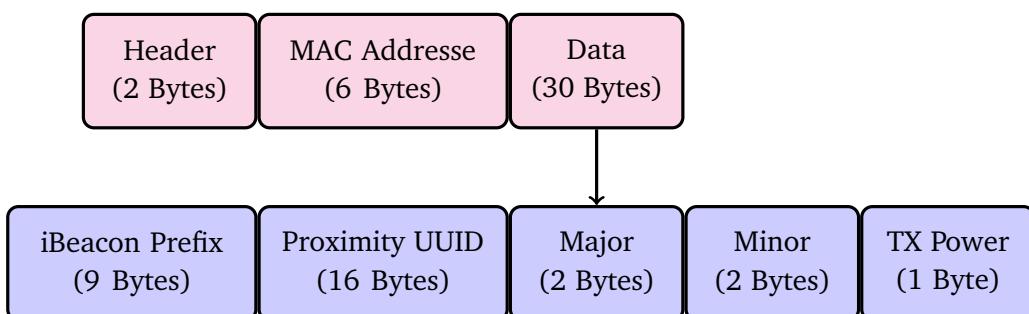


Abb. 5.11: Anteile eines Datenpaketes in der Beacon-Kommunikation auf der MAC-Ebene

Bei den tatsächlichen Laufzeiten des Sende- und Empfangsmodus muss jedoch ein Un-

sicherheitsfaktor eingerechnet werden, sodass nicht der berechnete Wert für eine Übertragung genommen wurde, sondern das Zweifache dessen, um ein weniger idealisiertes Bild für die einzelnen Vorgänge zu gestalten.

$$t_{TX} = 2 \text{ ms} \cdot \text{Intervalle [Hz]}$$

$$t_{RX} = 2 \text{ ms} \cdot 1 \text{ Hz}$$

$$t_{CPU} = t_{TX} + t_{RX}$$

$$t_{IDLE} = 1 - t_{CPU}$$

Weil für den Sendevorgang noch zusätzlich der Cortex M0 Prozessor aufgeweckt werden muss und dieser das Paket vorbereitet und bis zum Ende der Sendung abwartet, muss hierfür zusätzlich der Energieverbrauch berechnet werden. Weil wie erwähnt eine direkte Leistungsmessung an den Beacons durch die gering fließenden Ströme nicht möglich ist, wird diese aus dem Datenblatt des Prozessors rechnerisch ermittelt. Die aktive Phase des Prozessors für alle Operationen vor und nach einer Sendung wird dabei mit 2 ms Sekunden angenommen, was exakt der Sende- und Empfangsdauer entsprechen. Der Energiebedarf der Recheneinheit wurde im Datenblatt mit 4,1 mA bei einem Takt von 16 MHz angegeben. So ergibt sich für den Prozessor eine Leistung von 12,3 mW. Zusätzlich wird der Stromverbrauch im IDLE-Modus (untätige Phase des Prozessors) mit 2,6  $\mu$ A angegeben. Aus den genannten Gegebenheiten lassen sich nachfolgende Gleichungen erstellen und damit eine Simulation eines Lebenszyklus von einer Batterie näherungsweise bestimmen:

$E_{Bat} =$	$1,8 \text{ Wh} \cdot 0,7 =$	$1,26 \text{ Wh}$
$L_{CPU} =$	$4,1 \text{ mA} \cdot 3V =$	$12,3 \text{ mW}$
$L_{TX} =$ Polynom 4. Grades (von 4,7 mA $\sim$ 11,8 mA) $\cdot 3V =$	$14,1 \text{ mW} \sim 35,4 \text{ mW}$	
$L_{RX} =$	$6,1 \text{ mA} \cdot 3V =$	$18,3 \text{ mW}$
$L_{IDLE} =$	$2,6 \mu\text{A} \cdot 3V =$	$7,8 \mu\text{W}$

ergeben den Zusammenhang

$$t_{Laufzeit} = \frac{E_{Bat}}{L_{CPU} \cdot t_{CPU} + L_{TX} \cdot t_{TX} + L_{RX} \cdot t_{RX} + L_{IDLE} \cdot t_{IDLE}}$$

und das ergibt in Abbildung 5.12 folgendes Simulationsergebniss:

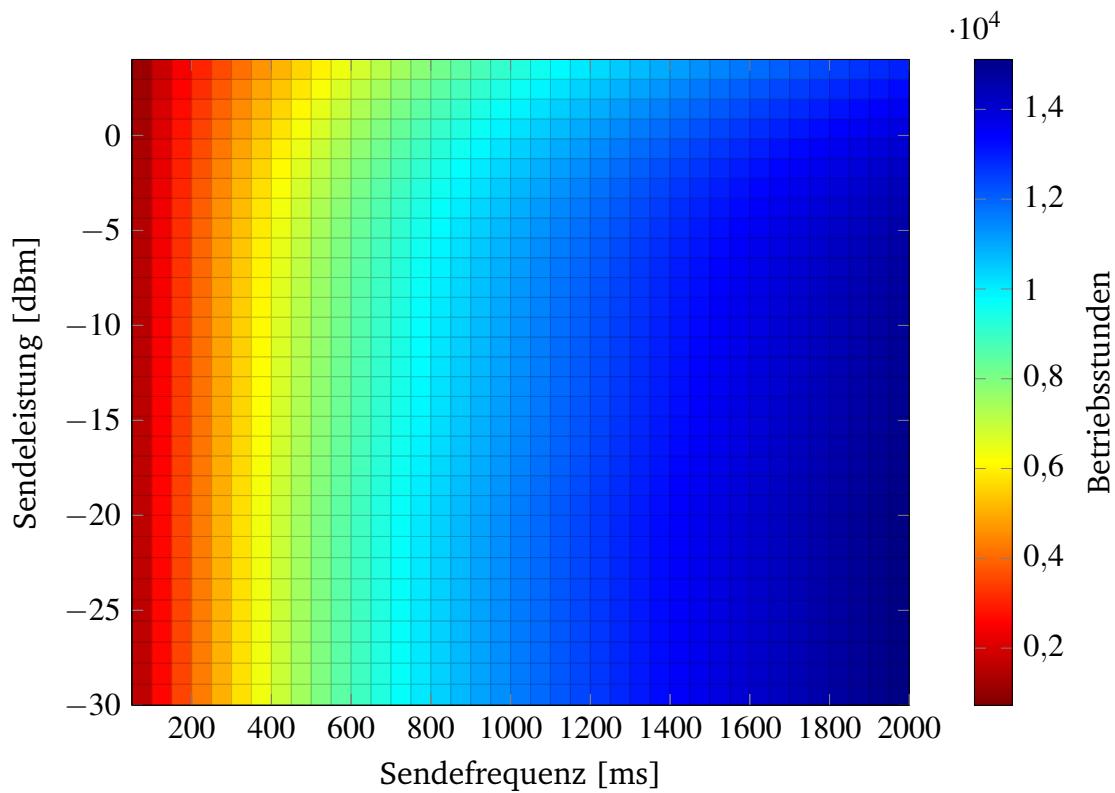


Abb. 5.12: Einfluss der einstellbaren Parameter auf die Lebensdauer einer CR2450-Batterie

Aufgrund der getroffenen Annahmen würde eine Batterie in den niedrigsten Einstellungen für rund zwei Jahre den Betrieb eines Beacons ermöglichen. Im realen Betrieb hatte sich gezeigt, dass eine Batterie schon nach vier bis fünf Monaten gewechselt werden musste. Während dieser Zeit wurden die Einstellungen nicht geändert und bei einer Intervalllänge von 5 Hz und Sendeleistung von -12 dBm belassen. Beides stimmt mit der Simulation ungefähr überein, was noch kein Beweis für die Gültigkeit der Annahmen bedeutet. Jedoch wird der qualitative Verlauf der wechselseitigen Beziehungen von Einstellparameter und Batterilaufzeit skizziert, der als Orientierung zur Auslegung der Parameter benutzt werden kann.



# 6 Modellentwurf und Simulation

Nach der sehr detaillreichen Analyse der Messwerte wird in diesem Kapitel ein Modell aus den aufbereiteten Daten gewonnen und dieses anschließend in eine Simulationsumgebung implementiert. Aus den Messungen muss zunächst qualitativ eine Aussage über das Verhalten der BLE-Signale getroffen werden, um nachfolgend ein geeignetes mathematisches Modell zu finden. Dieses gewählte Modell wird darauf an die Gegebenheiten mit einer Parameterschätzung angepasst. Für die Anpassung wird zunächst ein Minimierungsproblem anhand der Messwerte aus dem vorigen Kapitel erstellt, das mit einem Partikel-Schwarm-Optimierer gelöst wird. Die daraus gewonnenen Parameter lassen sich in die Modellgleichung einsetzen, um mit dessen Hilfe die Signalausbreitung der Beacons in einem Gebäude vorherzusagen.

Die Prädiktion der Ausbreitung von BLE-Funkwellen in einer Simulation kann dazu genutzt werden, mithilfe eines Optimierungsverfahrens die Leuchtfeuer in einem Raum so anzubringen, dass ein Lokalisierungssystem aus dieser Technologie mit maximaler Genauigkeit konfiguriert werden kann. Um zuerst die Abmessungen des betreffenden Raumes zu ermitteln, wird der Grundriss durch die Verwendung des Scitos G5 und der Software Miracenter als Bilddatei erstellt. Somit gelten die Wände als Begrenzung und der verbleibende Raum als Platz für die Signalausbreitung. Die maximale Genauigkeit der Positionsbestimmung würde logischerweise erreicht werden, wenn der komplette Raum mit Beacons unter Einhaltung des Mindestabstandes übersäht wäre, was jedoch nicht ökonomisch ist. Deswegen wird zusätzlich die Anzahl der Beacons in die Problemformulierung eingearbeitet, sodass für jede mögliche Beacon-Anzahl eine optimale Konfiguration existiert.

## 6.1 Modellbildung

Zunächst wird in diesem Abschnitt das zugrunde liegende Modell für das Verhalten der Signalausbreitung der Beacons bestimmt. Dazu wird aus Tabelle 6.1 ein geeignetes Kanalmodell ausgesucht und dessen ungefährer Verlauf bestimmt. An diesem Verlauf wird es an den realen Messwerten validiert und anschließend in dessen Qualität durch die Parameterschätzung verbessert. Die genannten Modelle in der Tabelle können mit ihren Spezifikationen im Grunde alle als Ausbreitungsmodell für die Leuchtfeuer dienen. Jedoch fällt die erste Wahl auf das sogenannte WINNER II-Modell, weil in dessen

Beschreibung die Anwendung in großen Räume mit hohem Aufkommen von Datenverkehr steht. Und weil diese Gegebenheiten in den Messungen aus dem vorigen Kapitel vorhanden sind und für den späteren Einsatzort (z.B. Kaufhäuser, Flugplätze etc.) der Beacon-Systeme gegeben sein müssen, ist dieses Modell vorzuziehen. Zudem beträgt die Reichweite eines Beacons nur 70 Meter, wohingegen die Geltungsbereiche der anderen Modelle dafür überdimensioniert erscheinen.

Modell	Antennen-höhen	Frequenz	Bereich	In-/Outdoor	Bemerkungen
Free Space Loss	nicht be-grenzt	$f \geq 30$ MHz	LOS be-grenzt	Indoor-Indoor- bzw. Outdoor-Outdoor- Übertragungen bei direkter Sichtverbindung (LOS)	Systeme, bei denen direkte Sichtverbindung angenommen werden kann
Extended Hata	$h_1 = [1 \text{ m} ; 10 \text{ m}]$ $h_2 = [30 \text{ m} ; 100 \text{ m}]$	30 MHz – 3GHz	$d \leq 40$ km	automatisches Hinzuaddieren von Verlusten an Wänden bei Indoor-Outdoor- bzw. Indoor-Indoor Übertragungen (mehrere Räume/Gebäude)	mobile und andere Dienste, ohne direkte Sichtverbindung, nur im Bereich 2 – 3 GHz implementiert
Extended Hata-SRD	$h = [1,5 \text{ m} ; 3 \text{ m}]$	30 MHz – 3GHz	$d \leq 0,3$ km	automatisches Hinzuaddieren von Verlusten an Wänden bei Indoor-Outdoor- bzw. Indoor-Indoor Übertragungen (mehrere Räume/Gebäude)	Kurzstrecken-verbindungen, bei denen wenigstens LOS angenommen werden kann
WINNER II	$h \leq 6 \text{ m}$	2 GHz – 6GHz	$d = [5 \text{ m} ; 100 \text{ m}]$	Indoor-Indoor- Übertragungen bei LOS oder NLOS	Anwendung in großen offenen Räumen bei hohem Datenverkehr (z.B. Konferenzhalle, Fabrikgebäude)

Tabelle 6.1: Übersicht der Kanalmodelle im Indoor-Bereich, in Anlehung an [4]

### 6.1.1 Das WINNER II-Modell

Das WINNER II-Modell [32] benannt nach dem „Wireless-World-Initiative-New-Radio“ (WINNER)-Konsortium, das aus dem Zusammenschluss mehrerer Unternehmen besteht und die Verbesserung der Leistung von Mobilfunkkommunikationssystemen anstrebt, beinhaltet verschiedene Aspekte in der Berechnung von Ausbreitungsverlusten. Die Formel, die dem Modell zu Grunde liegt, beschreibt den Verlust einer Signalstärke in Abhängigkeit zu einem zurückgelegten Weg folgendermaßen:

$$L_{\text{WINNER}} [\text{dBm}] = A \cdot \log_{10} (d [\text{m}]) + B + C \cdot \log_{10} \left( \frac{f_c [\text{GHz}]}{5} \right)$$

In der Formel steht  $L_{\text{WINNER}}$  für den Leistungspegel in Dezibel Milliwatt (dBm), der das logarithmische Verhältnis aus einer Leistung in Form der Empfangsstärke im Vergleich zur anfänglichen Sendeleistung des Beacons angibt. So kann die obige Gleichung umgeschrieben werden zu:

$$10 \cdot \log_{10} \left( \frac{P_{\text{Sendung}} [\text{mW}]}{P_{\text{Empfang}} [\text{mW}]} \right) = A \cdot \log_{10} (d [\text{m}]) + B + C \cdot \log_{10} \left( \frac{f_c [\text{GHz}]}{5} \right)$$

oder auch direkt als dBm-Einheit formuliert werden:

$$L_{\text{Sendung}} [\text{dBm}] - L_{\text{Empfang}} [\text{dBm}] = A \cdot \log_{10} (d [\text{m}]) + B + C \cdot \log_{10} \left( \frac{f_c [\text{GHz}]}{5} \right) \quad (6.1)$$

Die Parameter der Gleichung bestehen primär aus einem Koeffizienten  $A$ , der die in der Luft zurückgelegte Strecke des Signals proportional gewichtet. Des Weiteren fließen zum einen noch der additive Einfluss von dem Parameter  $B$  für sonstige Störfaktoren und zum anderen der ebenfalls multiplikative Faktor  $C$  zur Gewichtung des Ausbreitungsverlustes durch die Frequenz  $f_c$  mit ein. In Abbildung 6.1 ist der qualitative Verlauf der Empfangsstärke, die nach oben hin abnimmt und der Distanz, die nach rechts zunimmt, skizziert. Als Parameter wurden Werte aus der Fachliteratur [4] entnommen und in die Gleichung eingesetzt. Im Vergleich zu Abbildung 5.3 sieht der Funktionsverlauf weitgehend identisch aus, sodass nun mit der Parameterschätzung begonnen werden kann.

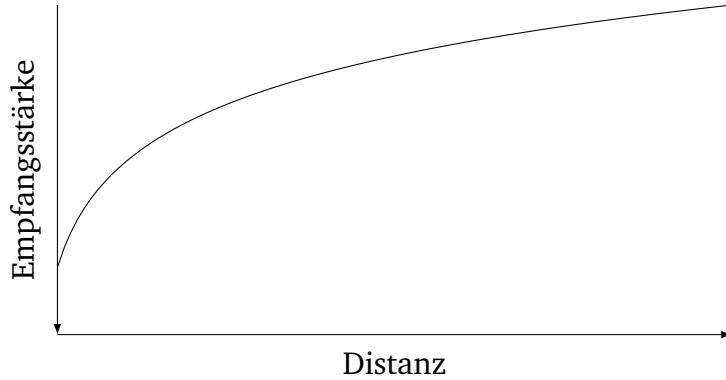


Abb. 6.1: Qualitativer Verlauf des WINNER II-Modells

### 6.1.2 Parameterschätzung

Um die qualitative Repräsentation des WINNER II-Modells in eine qualitative für die Beschreibung des Ausbreitungsverlustes von BLE-Signalen zu überführen, müssen die Parameter  $A$ ,  $B$  und  $C$  dementsprechend angepasst werden. Die Gleichung aus 6.1 vereinfacht sich dabei nochmals, wenn  $f_c$  als konstant angenommen wird und die Ausdrücke aus  $B$  und  $C$  miteinander zu dem neuen Parameter  $B^*$  zusammengefasst werden. Die Anpassung wird dabei in Form einer Optimierungsaufgabe vorgenommen, wobei es gilt, die richtige Distanz anhand der RSSI-Werte zu schätzen. Im vorangegangenen Kapitel ergaben die Messungen, dass unter Betrachtung aller Daten es sehr schwierig ist eine genaue Vorhersage zu treffen. Deswegen wurden nur die stärksten 100 Signale einer Distanz gemittelt und diese Ergebnisse unter Abb. 5.3 vorgestellt. Die Durchschnittswerte werden anschließend zusammen mit den realen Distanzen in gleicher Reihenfolge der Länge  $k$  in die zwei Vektoren  $d_{real}(k)$  und  $RSSI(k)$  gespeichert. Aus der Umstellung der Gleichung 6.1 nach  $d$  lässt sich aus dem empfangenen RSSI-Wert die geschätzte Entfernung bestimmen. Mit all diesen Informationen lässt sich ein Optimierungsproblem so formulieren, dass eine zu minimierende Kostenfunktion  $J$  als Differenz aus der direkt gemessenen Distanz eines Beacons zum Smartphone zu der berechneten Distanz aufgefasst und durch deren folgende Quadrierung so eine parabolische Funktion erstellt wird. Die Quadrierung bewirkt, dass der Wert der Kostenfunktion von allen Seiten in Richtung des globalen Minimums abnimmt und somit das Optimierungsproblem einfacher zu lösen ist.

$$\min_{A, B^*} J = \sum_{k=1}^{k_{end}} \left( d_{real}(k) - 10 \frac{L_{Sendung} - RSSI(k) - B^*}{A} \right)^2 \quad (6.2)$$

Als Lösungsalgorithmus für die Optimierung wird ein Partikel-Schwarm-Algorithmus (PSO) gewählt. Der Partikel-Schwarm-Optimierer ist ein Algorithmus mit einer künst-

lichen Intelligenz, die es zur Aufgabe hat, eine Funktion zu minimieren bzw. zu maximieren. Es ist dabei einfacher, sich die Partikel als einen Vogelschwarm vorzustellen. Der Grundgedanke dabei ist, dass mehreren Partikeln ein zufälliger Startwert  $p(t)$  im Suchraum unter Berücksichtigung der Gleichungsnebenbedingungen zugeordnet wird. D.h. im übertragenen Sinn, dass ein Vogelschwarm auf einer Wiese ausgesetzt wird. Ausgehend von diesem Startwert werden die einzelnen Vögel/Partikel im ersten Schritt in verschiedene Richtungen mit unterschiedlichen Geschwindigkeiten geschickt. Anschließend werden an den Zielpunkten die Kostenfunktionen mit den Positionen (oder auch Parametersätzen) berechnet und so der Funktionswert bestimmt. Die Schwarmintelligenz wird hierbei so realisiert, dass jedes Individuum seine aktuelle Position mit den zugehörigen Kosten kennt sowie über die Position des Partikels mit den geringsten Kosten im ganzen Schwarm informiert wird. In Folge des Wissens um die beste Position, fliegen alle Vögel/Partikel in Richtung des lokalen Minimum und decken somit den gesamten Raum ab.

Als Gleichung aufgefasst wird die Geschwindigkeit  $v$  im nächsten Zeitschritt  $t + 1$  eines Partikels aus der alten Geschwindigkeit  $v(t)$ , aus den drei konstanten Gewichtungen  $w$ ,  $r_1$  und  $r_2$ , zwei Zufallszahlen  $c_1$  und  $c_2$  und verschiedenen Positionen berechnet. Die Punkte sind zum einen die aktuelle Position  $x(t)$ , die mit den besten Kosten des einzelnen Individuums  $p$  und zum anderen die Generation der geringsten Kosten im ganzen Schwarm  $g$ . Im Ganzen lauten die Gleichungen zur Bestimmung des nächsten Punktes [23]:

$$v(t+1) = w \cdot v(t) + c_1 \cdot r_1 \cdot (p - x(t)) + c_2 \cdot r_2 \cdot (g - x(t)) \quad (6.3)$$

$$x(t+1) = x(t) + v(t+1) \quad (6.4)$$

Das Abbruchkriterium wird hierbei mit einer maximalen Anzahl an Generationen/Durchläufen der Hauptschleife angegeben. Der Vorteil dieses Verfahrens ist es, dass der komplette Suchraum abgetastet wird und durch die Intelligenz des Schwarms die Parametersätze mit den geringsten Kosten identifiziert werden. Der Nachteil ist der relativ hohe Rechenaufwand im Gegensatz zu anderen Verfahren (z.B. Newton, Trust-Region, Simplex etc.), zumal kein besseres Ergebnis mit dem PSO für ein solches Problem zu erwarten wäre. Jedoch baut der erweiterte Partikel-Schwarm-Optimierer in einem der nächsten Abschnitte auf diesem Verfahren auf, sodass in diesem Teil der Arbeit die Vorstellung der einfachen Variante zur Einführung vorteilig ist.

Als Ergebnis der Parameterschätzung wurden in Abbildung 6.2 zunächst die be-

rechneten RSSI-Werte für die jeweiligen Distanzen im Vergleich zu den gemessenen Mittelwerten aller drei Beacons gegenübergestellt. Wie schon in der Analyse der Messwerte kritisiert, entspricht das Verhalten des Ausbreitungsverlustes ab dem zehnten Meter nicht mehr der theoretischen Annahme. Während der Verlauf vom ersten Abschnitt sehr gut geschätzt werden kann, versagt aber das einfache WINNER II-Modell bei größeren Distanzen. Ähnlich verhält es sich bei der Schätzung der Entfernung, wie Abbildung 6.3 zeigt. Hier wurde das Modell lediglich nach  $d$  umgestellt und als Eingabe dienten die Mittelwerte aus 6.2. In Tabelle 6.2 sind die Parameter aus der Optimierung aufgeführt, welche in das Modell zur Berechnung der letzten Diagramme eingesetzt wurden.

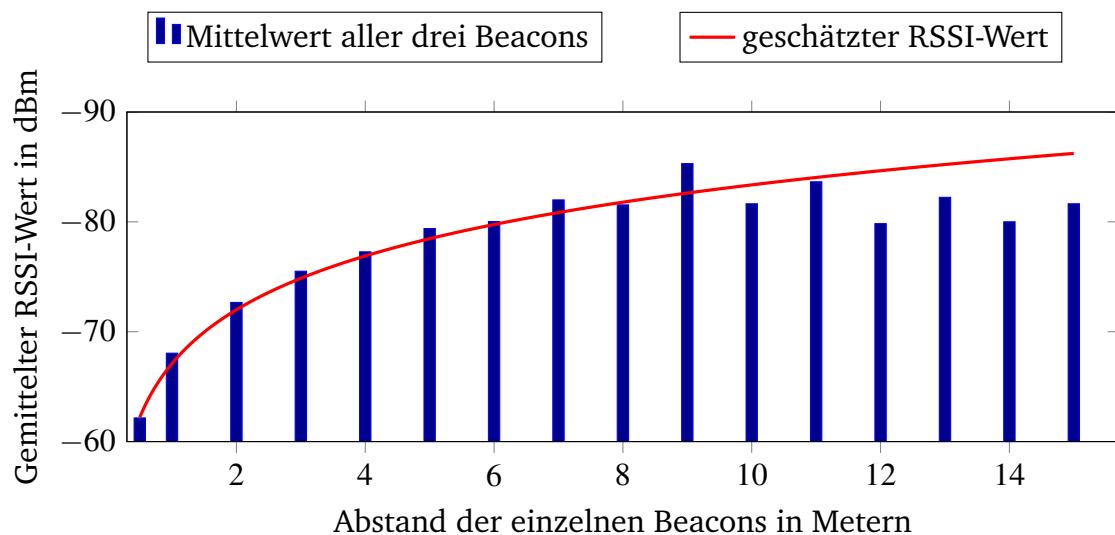


Abb. 6.2: Durchschnittlicher RSSI-Wert und berechneter RSSI-Wert im Vergleich

	Parameter	
	A	B*
Wert	16,2623	55,1046

Tabelle 6.2: Parameter des Modells

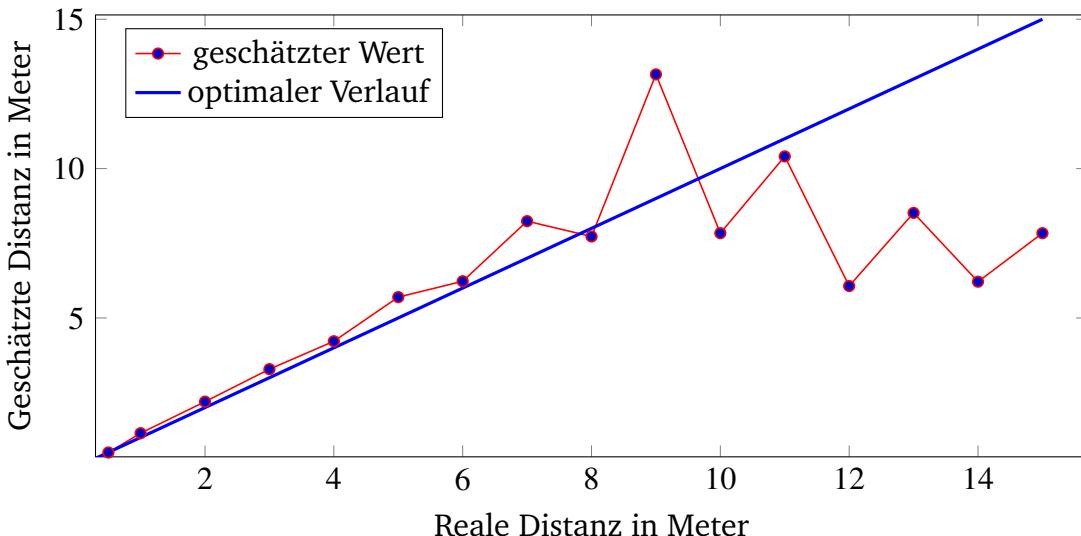


Abb. 6.3: Vergleich der Distanz-Schätzung mit dem WINNER II-Modell zur gemessenen Entfernung

## 6.2 Vertrauensskala

Im Hinblick auf die verschiedenen Betriebsmodi der Beacons, den unterschiedlichen Arten der Positionsbestimmung (Landmarkensystem und Mikro-Lokalisierung) und vielen noch nicht vorgestellten Techniken mit denen sich die Vorhersage der Position bestimmen und verbessern lassen, fällt es sehr schwer die Güte der Leuchtfeuer-Signale einzuschätzen. Mit Güte ist gemeint, inwieweit dem Modell 6.1 vertraut wird die aktuelle Position zu schätzen und wie genau diese Schätzung in Metern oder Zentimetern erfasst werden kann. Um unabhängig von der Art der Lokalisierung, dem Anwendungsbereich und den eingesetzten Techniken die Güte angeben zu können, wird lediglich die Wahrscheinlichkeit mit dem WINNER II-Modell die tatsächliche Distanz zu ermitteln, als Wert dafür angenommen. D.h. aus den Messungen der RSSI-Werte vom vorigen Kapitel werden dieses Mal alle Daten für die Distanzberechnung herangezogen und mit einer Toleranz versehen. Die berechneten Entfernungen werden daraufhin mit realen Distanzen verglichen und im Falle, dass sie sich nur in einem definierten Spielraum voneinander unterscheiden, wird dies als eine erfolgreiche Positionsbestimmung angesehen. Der prozentuale Anteil aller erfolgreichen Berechnungen wird so als Gütemaß gewertet und somit ist die Qualität der Beacon-Signale als eine statistische Wahrscheinlichkeit, des Empfanges einer für das Modell geeigneten Signalsstärke pro Messung, aufzufassen. Die Ergebnisse aus diesen Überlegungen finden sich in Abbildung 6.4 und 6.5 wieder. Sie zeigen einmal die Wahrscheinlichkeit  $W_{\text{berech}}$  ein gültiges Signal zu empfangen bei einer Toleranz von 0,5 und bei 2 Metern, wobei die dazugehörigen festen Distanzen

die jeweiligen gemittelten drei Messreihen  $d_{mitt}$  darstellen. Zudem wurde versucht, die Güte als Funktion zu charakterisieren. Dazu wurde die Funktion 6.5 aufgrund ihrer Art ausgewählt und um weitere Parameter erweitert (siehe Gleichung 6.6), die schließlich durch die Lösung des Minimierungsproblems unter 6.7 und unter der Verwendung vom PSO bestimmt wurden (Ergebnis siehe Tabelle 6.3).

$$W_{berech} = \frac{1}{d_{mitt} + 1} \quad (6.5)$$

$$W_{berech} = \frac{1}{p_1 \cdot d_{mitt} + p_2} + p_3 \quad (6.6)$$

$$\min_{p_1, p_2, p_3} J = \sum_{k=1}^{k=15} (W_{real}(k) - W_{berech})^2 \quad (6.7)$$

Der Vorteil dieser Betrachtungsweise liegt in den Funktionen der Beacons begründet, da hier die Frequenz bzw. Häufigkeit der Sendevorgänge eingestellt werden kann. Durch die Erhöhung der Übertragungen pro Sekunde lässt sich somit die Zeit verringern, bis ein gültiges Signal das Messgerät erreicht. Anhand dieser Einstellmöglichkeiten lässt sich zusammen mit der hier vorgestellten Skala die Dynamik der Messungen beeinflussen.

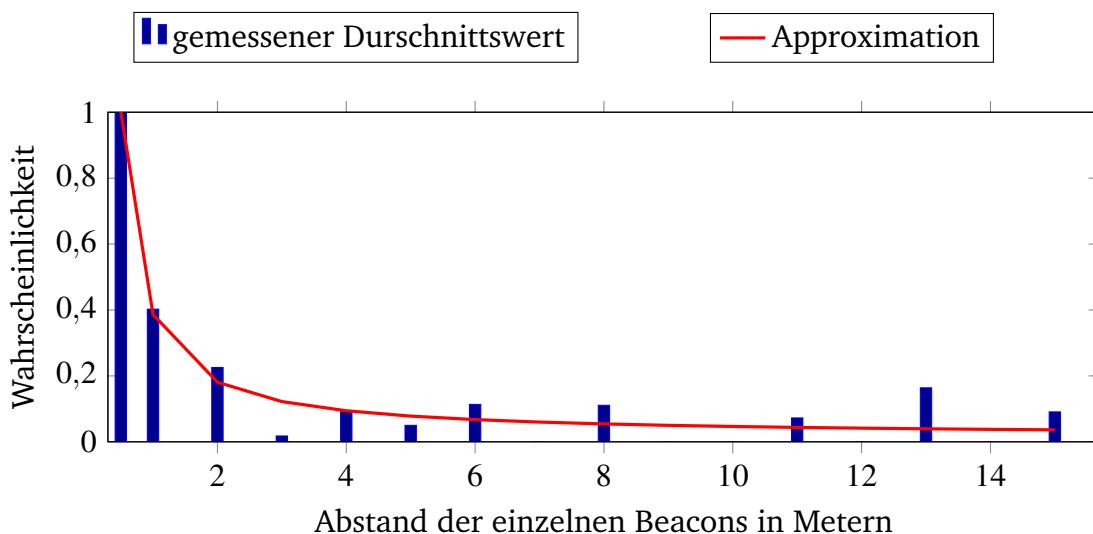


Abb. 6.4: Wahrscheinlichkeit des Empfangs gültiger Signale in einem Toleranzbereich von 0,5 Meter und dessen Approximation

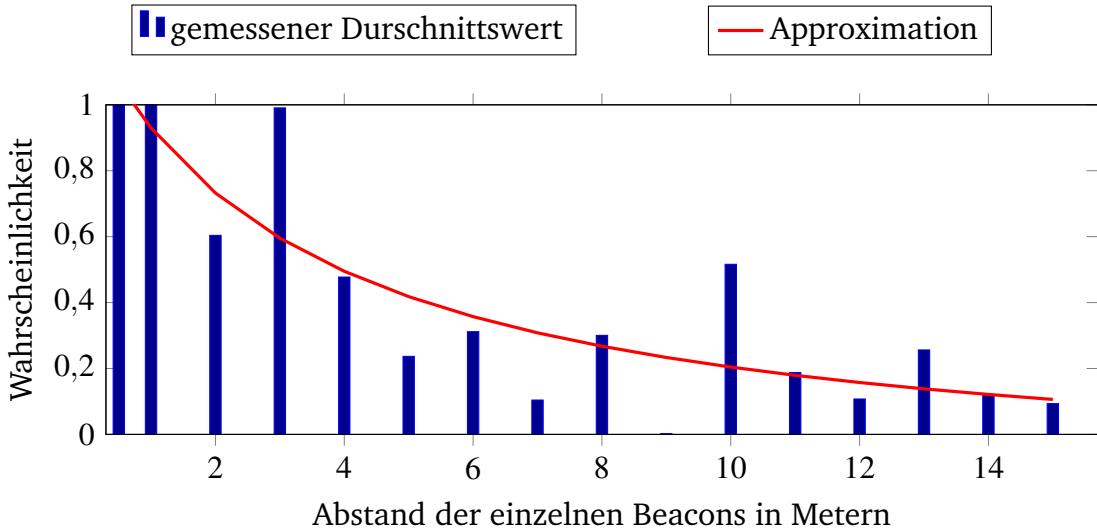


Abb. 6.5: Wahrscheinlichkeit des Empfangs gültiger Signale in einem Toleranzbereich von 2 Metern und dessen Approximation

	Parameter			
	$p_1$	$p_2$	$p_3$	
Toleranz   0,5 m	3,3777	-0,6746	0,0163	
	2 m	0,2035	0,7145	0,1594

Tabelle 6.3: Parameter der Vertrauensskala

## 6.3 Entwicklung der Simulationsumgebung

Mit den gewonnenen Erkenntnissen aus der Modellbildung können nun Vorhersagen über die Signalausbreitung und der zu erwartenden Genauigkeit der Entfernungsschätzungen getroffen werden, die schließlich zu einer optimalen Planung eines Indoor-Lokalisierungssystems genutzt werden können. Dazu wird zunächst eine Entwicklungsumgebung zur Herstellung der Simulation benötigt. Als Basis der Simulation wird eine grafische Oberfläche (GUI) mit dem Editor GUIDE erstellt, der Teil des Programmes MATLAB [26] ist. In dieser Sektion werden die einzelnen Funktionen der GUI besprochen und deren Verarbeitungskette vorgestellt. Hierfür wird als erstes die Konvertierung der Zeichnung eines Grundrisses (als PNG-Bilddatei) in verarbeitbare Informationen erläutert, die von substantieller Bedeutung für den gesamten Simulationsprozess ist. Anschließend wird in der weiteren Implementierung speziell der Aspekt einer optimalen Verteilung, unter besonderer Betrachtung der Vertrauensskala, aufgegriffen. Eigens dafür wurde der vorher vorgestellte Partikel-Schwarm-Algorithmus aus der Parameterschätzung erweitert, sodass er auch für den abstrakten Fall einer Anordnung von Funk-

baken eingesetzt werden kann. Abschließend wird mit einem fiktiven Grundriss das komplette Lighthouse Keeper-Prinzip exemplarisch vorgeführt.

### 6.3.1 Grundrissanalyse

Für eine weitgehende Automation der gesamten Planung müssen die Grundrisse bzw. Rauminformationen, die mit der Software Miracenter (siehe Abschnitt 4.2.2) generiert wurden, vor der Simulation ausgewertet werden. In der Auswertung oder Analyse dieser Daten muss das Programm selbstständig in der Lage sein, eine Wand oder generell einen Raum zu erkennen. Zudem müssen auch Befestigungspunkte definiert sein, an denen die Beacons angebracht werden können. Die Umsetzung dieser Funktionen basiert hierbei auf einer Eigenentwicklung, da bisher keine vergleichbaren Arbeiten existieren. Für die Erkennung von Wänden respektiven Räumen könnten zwar Algorithmen eingesetzt werden, die beispielsweise Formen erkennen oder Cluster von Farbgruppierungen bilden. Im Hinblick jedoch auf mögliche Erweiterungen der gesamten Problematik, z.B. mit Hindernissen oder mit Bereichen, in denen keine Beacons befestigt werden können, bietet sich das hier selbstentwickelte Verfahren an. Für ein besseres Verständnis wird der allgemeine Vorgang diesbezüglich in Abbildung 6.6 illustriert. Das Beispiel zeigt einen quadratischen Raum, wobei die Wände in schwarz und die freien Räume in weiß gehalten sind. Die einzelnen Unterteilungen sollen die Pixel darstellen, sodass eine Auflösung von  $7 \times 7$  Pixeln gegeben ist. In einer ersten Phase werden in vertikaler Reihenfolge die einzelnen Pixel von „links nach rechts“ sowie deren Vorgängerpixel und deren Nachfolgerpixel betrachtet. Im vorliegenden Fall werden dafür zwei Matrizen mit  $7 \times 7$  Einträgen initialisiert, in der jeder Eintrag zunächst auf Null gesetzt ist. Wechselt in der abgefragten Reihe ein weißer Pixel auf einen schwarzen, so wird die Position des schwarzen Pixels in der Befestigungs-Matrix um eins erhöht und in der Raum-Matrix wird die Position des weißen Pixels ebenfalls mit eins addiert. Die folgenden weißen Pixel werden anschließend auch als Raumkoordinaten markiert, solange kein weiterer schwarzer Punkt erscheint. Falls eine weitere Wand im Verlauf der Reihe detektiert wird, setzt die Raummarkierung aus und lediglich der schwarze Bildpunkt wird als Befestigung eingeschrieben.

Diese Vorgehensweise allein ist jedoch nicht robust genug, was auch in der Abbildung skizziert wurde. In diesem Fall wurden zwei Befestigungen und zwei Raumpunkte falsch interpretiert, weil der Algorithmus nur den unmittelbaren Nachbarn eines Bildpunktes betrachtet und somit nicht feststellen konnte, ob der Raum mit Wänden komplett umschlossen ist oder nicht. In Folge dieses bekannten Fehlers wurde für alle

weißen Pixelkoordinaten an den Bild-Rändern eine rückwärtsgerichtete Suche implementiert, die bis zur nächsten Wand alle Einträge beider Matrizen um eins reduziert, wie auch rechts neben der ersten Phase zu sehen ist. Somit wird sichergestellt, dass die Räume stets abgeschlossen sind und keine Befestigungen an den Ecken existieren. Die gleiche Vorgehensweise wird dreimal in jede Richtung („von rechts nach links“, „von unten nach oben“ und „von oben nach unten“) wiederholt. Dabei entstehen Vertrauenswerte in den Matrizen, anhand derer über die endgültige Definition nach Befestigungen und Räumen entschieden wird. In den verschiedenen Versuchen und Testszenarien mit offenen Türen und verwinkelten Gängen erwies sich der vorgestellte Algorithmus als sehr zuverlässig. Zusätzlich wurde die Analyse auf den Rot-, Grün- und Blau-Farbraum (RGB) erweitert, sodass mit weiteren Farben spezielle Objekte erfasst werden. Zum Beispiel können nun mit roten Farben Bereiche gekennzeichnet werden, an denen keine Beacons befestigt werden können.

Damit die Bildinformationen im metrischen Raum abgebildet werden können, muss zunächst deren Verhältnis von Pixel zu Meter bekannt sein. In der Simulation kann dieses Verhältnis direkt angegeben werden, indem die Abmessungen des gesamten Raumes in die GUI eingetragen werden. Aus diesen Informationen berechnet die Software eigenständig die Verhältnisse und überträgt die Pixel in metrische Koordinaten. Anschließend werden die Wände bzw. Befestigungen durch einfache Polygone ersetzt, wobei die Ecken von Wänden als Referenzpunkte genommen werden. Dies hat den Vorteil, dass für eine Aufteilung der Beacons entlang dieser Linien nur Abstandswerte kalkuliert werden müssen. Zudem kann die Kollision von Signalen und Wänden schneller berechnet und somit entschieden werden, ob sich ein Beacon in der LOS oder NLOS zu einem Punkt im Raum befindet. Im letzten Schritt der Analyse wird der Raum in einzelne Elemente untergliedert. Die Größe der Quadrate wird dazu in die Oberfläche der Simulation eingetragen. Die Aufteilung in finite Elemente ergibt sich aus der Nutzung der Vertrauensskala, die abhängig von der gewünschten Toleranz bzw. Lokalisierungsgenauigkeit ist. Diese Toleranz bildet die Größe eines Raumelementes, in der die modellbasierte Entfernungsschätzung ( $\pm$  der Toleranz) mit einer bestimmten Wahrscheinlichkeit den tatsächlichen Abstand eines Raumelementes zu einem Beacon angibt.

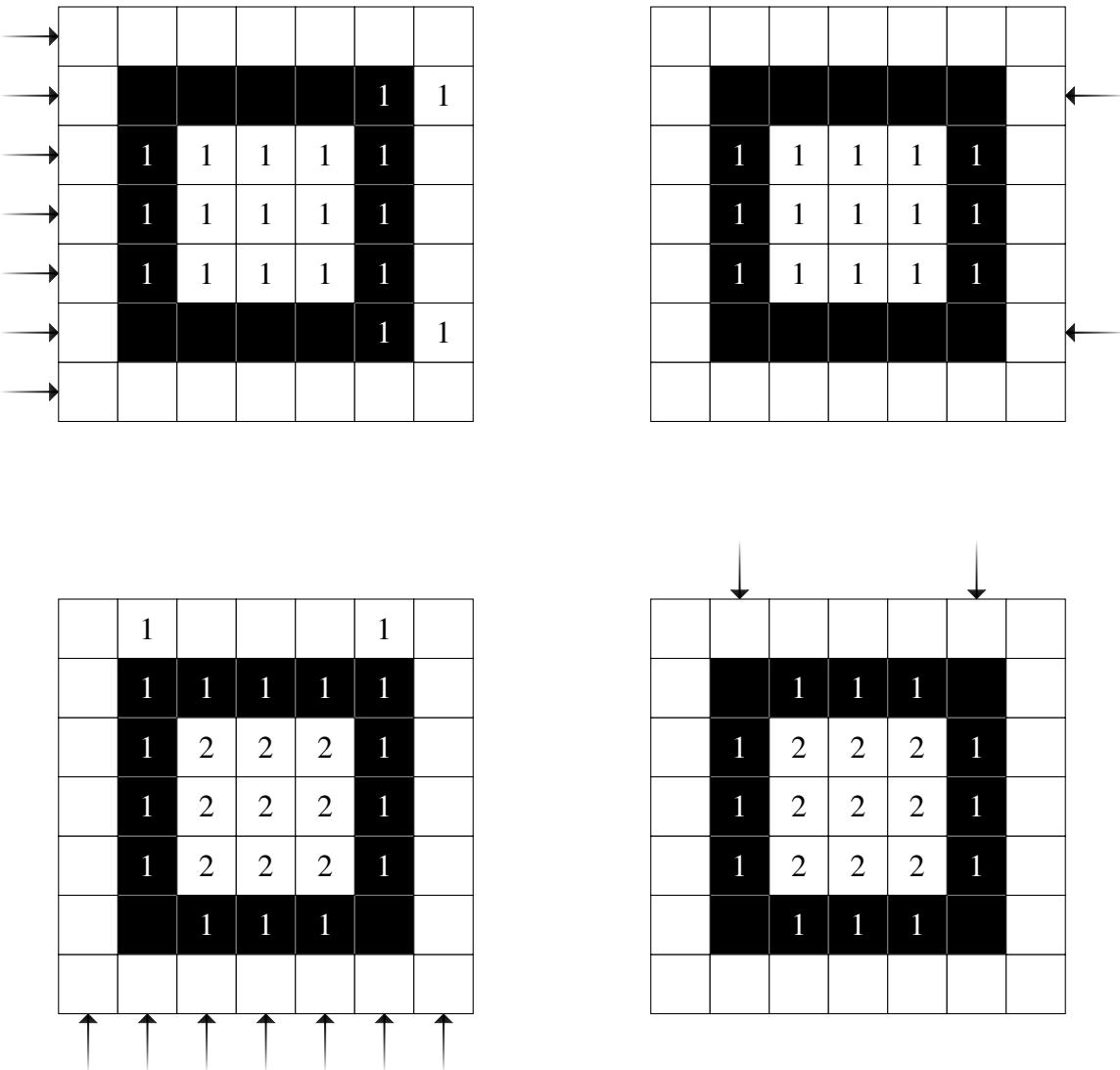


Abb. 6.6: Schematischer Ablauf der Grundriss-Analyse

### 6.3.2 Optimierung der Beacon-Infrastruktur

Das erklärte Ziel des Lighthouse Keeper-Konzeptes ist es, eine größtmögliche Genauigkeit in der Lokalisierung von Objekten im Indoor-Bereich, bei einer minimalen Anzahl von Beacons zu erreichen. Der Unterschied zu den bisherigen Optimierungsverfahren in dieser Arbeit besteht zum einen in den zu beeinflussenden Parametern und zum anderen in der mehrkriteriellen Optimierungsaufgabe. Während bisherige Parameter zu den rationalen Zahlen gehörten und sie somit kontinuierlich änderbar waren, existieren in dieser Optimierungsaufgabe lediglich Parameter als Boolesche Variablen (Beacon platziert/nicht platziert). Zudem muss die Kostenfunktion nach zwei Kriterien (Genauigkeit und Anzahl) optimiert werden, weswegen ein Ansatz in Richtung Pareto-Optimierung führt.

Im folgenden Abschnitt wird der Algorithmus zur Optimierung der Beacon-Infrastruktur näher erläutert. Die Optimierung findet dabei unter Zuhilfenahme der Vertrauensskala als Genauigkeitsmaßstab und der Verwendung des Modells zu den Ausbreitungsverlusten von BLE-Signalen statt. Im zweiten Teil wird auf die Bewertung der beiden unterschiedlichen Kriterien eingegangen und dazu wird der Begriff Pareto-Front näher erklärt. Ein Beispiel für die Optimierung findet sich im Anschluss an diese Sektion in dem Beispiel für den Lighthouse Keeper.

### **Erweiterter Partikel-Schwarm-Algorithmus**

Derweil genetische Algorithmen in den Bereichen der mathematischen Problemformulierungen nur eine geringe Rolle spielen und nur in wenigen Ausnahmen bessere Ergebnisse als herkömmliche Verfahren liefern, stellen sie bei Problemen mit Entscheidungszweigen die einzige praktikable Lösung dar. Die Arbeit hinter der Erweiterung des PSO wurde dabei maßgeblich noch von zwei weiteren genetischen Algorithmen beeinflusst. Es finden sich Teile des „Strength Pareto Evolutionary Algorithm 2“ (SPEA2) [12] sowie des „Simulated Annealing“-Verfahrens [40] in dieser neuen Methode wieder. Denn erst die Kombination aller genannten Ansätze lieferte die gewünschten Ergebnisse. Für die eigentliche Optimierung werden zunächst die Kostenfunktionen beider Merkmale benötigt. Erscheint indes die Funktion der Anzahl der Beacons mit  $\min J_{Anzahl}$  trivial, unterliegt die Lokalisierungsgenauigkeit mehreren Faktoren, die je nach Anwendungsfall unterschiedlich gewichtet werden können. Statt unterschiedlichen Formulierungen des Problems wird stellvertretend für die Qualität von Signalen die Vertrauensskala als allgemeine Referenz gewählt. Es wird davon ausgegangen, dass die Ortungsgenauigkeit beispielsweise mithilfe der Trilateration überall gleich ist, solange Beacon-Signale empfangen werden. Es hängt jedoch von den Wahrscheinlichkeiten ab, wie lange auf ein gültiges Signal gewartet werden muss, bis eine Positionsbestimmung die gewünschten Ergebnisse liefert.

Ausgehend von der Forderung eine bestmögliche Abdeckung des Raumes mit Beacon-Signalen zu erreichen, stellt sich die Frage nach der Relation der Vertrauenswerte in jedem Abschnitt des Raumes. Würden angenommen zwei Beacons relativ nah nebeneinander liegen, würde in ihrem unmittelbaren Umkreis der Vertrauenswert besonders hoch liegen, falls von beiden Beacons die Werte addiert werden. Jedoch sind zwei Beacons in einem konzentrierten Bereich wenig effektiv, wenn in einer entfernten Ecke eines Raumes dafür keine Beacons plaziert wurden. Unter Betrachtung dieser Überle-

gung wurde so eine Gütekriterium festgelegt, die sich aus dem höchsten Vertrauenswert eines Beacons in jedem Raumelemente zusammensetzt. Mathematisch formuliert werden über die Anzahl aller Beacons  $n$  und über die Menge der Raumelemente  $m$  zuerst die einzelnen Entferungen  $d_{n,m}$  bestimmt. Darauf folgend werden mit der Gleichung 6.6 die einzelnen Vertrauneswerte  $W_{n,m}$  geschätzt und daraus das Gütemaß mithilfe der Summe der besten Signale in einem Raumelement  $\text{best} \{ W_{1\dots n_{end},m} \}$  gebildet. Somit ergibt sich folgendes Maximierungsproblem:

$$\begin{aligned} d_{n,m} &= \sqrt{(Pos_{n,x}^{\text{Beacon}} - Pos_{m,x}^{\text{Element}})^2 + (Pos_{n,y}^{\text{Beacon}} - Pos_{m,y}^{\text{Element}})^2} \\ W_{n,m} &= \frac{1}{p_1 \cdot d_{n,m} + p_2} + p_3 \\ \max_{Pos^{\text{Beacon}}} J_{\text{Qualität}} &= \sum_{m=1}^{m_{end}} \text{best} \{ W_{1\dots n_{end},m} \} \end{aligned}$$

Der erweiterter Partikel-Schwarm-Algorithmus muss nun nach der Anzahl von kleinen Sendern optimieren. Dabei bedient sich das Verfahren an dem Belohnungs-/Bestrafungssystem (oder auch Selektion genannt) aus SPEA2 und an den Heuristiken aus dem Simulated Annealing. In der Initialisierungsphase werden die einzelnen Populationen des Schwarms mit Beacons voll besetzt, sodass diesesmal beim PSO jeder Partikel den gleichen Startwert besitzt. Lediglich im Belohnungs-/Bestrafungssystem werden die Gewichtungen anfangs für eine Zunahme bzw. Abnahme von Beacons zufällig zwischen den Werten 0 und 1 vergeben. Im nächsten Schritt wird für jede Konfiguration die jeweilige Güte  $J$  berechnet und anschließend im Bewertungssystem mit der vorigen Generation verglichen. Als Maßstab für eine bessere oder schlechtere Generation kommen daher die vorher definierten Kriterien zum Einsatz. Dabei werden im Endeffekt die Konfigurationen mit der geringen Anzahl an Beacons faktisch besser bewertet. Dieser Schritt ist notwendig, da mit weniger Beacons die räumliche Abdeckung naturgemäß abnimmt und somit  $J_{\text{Qualität}}$  nie ansteigen wird. Jedoch lassen sich die Partikel mit gleicher Beacon-Anzahl in ihrer Qualität vergleichen und hier wird das Individuum belohnt, welches die höchste Güte in der Vertrauenskala aufweist und das mit dem niedrigerem Wert wird bestraft. Dementsprechend werden die jeweiligen Gewichtungen in den Grenzen 0 und 1 prozentual erhöht bzw. gesenkt. Im Folgenden kommt der Charakter des PSO zum Einsatz. Denn bei der Neuanordnung der Konfigurationen werden alle Gewichtungen eines Partikels mit speziellen Individuen verrechnet. Diese Partikel bzw. deren Gewichtungen bestehen aus denen mit der besten Fitness im Schwarm und aus der eigenen besten Fitness, die jemals erreicht wurde. Im Grunde werden die einzelnen Gewichtungen zur Wegnahme oder Anbringung eines Beacons einander gegenüber gestellt, wobei sich der größere Wert durchsetzt. Tritt dabei nach

einer unbestimmten Iterationszeit ein totes Ende auf, d.h. das alle Gewichtungen genau 0 oder 1 sind, behilft sich der Algorithmus mit den Heuristiken des Simulated Annealing-Verfahrens. In diesem speziellen Fall wird ein totes Ende detektiert und das betroffende Individuum „mischt“ seine Gewichtungen mit anderen zufällig gewählten Partikeln und findet meist so aus einem lokalen Optimum heraus, solange es sich nicht im globalen Optimum befindet.

Zur Veranschaulichung wird hier noch einmal die Hauptschleife im ersten Aufruf durchlaufen:

Eingabe: **N** (Größe des Schwarms)  
**D** (Dimension des Problems – Anzahl an Befestigungspunkten)

**I** (Maximale Anzahl an Generationen)

**KR** (Mittelpunkt-Koordinaten der Raumelemente)

**KB** (Koordinaten der Befestigungen)

Ausgabe: **F** (Fitnesswerte einer jeden Generation)

**E** (Konfigurationen mit den besten pareto-optimierten Kriterien)

Schritt 1: **Initialisierung:** Generierung der Konfigurationen  $\mathbf{B}_0$  und aller Gewichtungen  $\mathbf{G}_{Zu,n}$  und  $\mathbf{G}_{Weg,n}$  der Beacons für alle Partikel.

Schritt 2: **Fitnessbestimmung:** Berechnung der Fitnesswerte für jedes Individuum im Schwarm.

Schritt 3: **Bewertung:** Vergleich aller Kopien der vorherigen Generation mit der neuen. Falls die neuen Parametersätze optimalere Ergebnisse liefern, werden alle Zustände auf Unterschiede untersucht und lediglich die Veränderungen belohnt. Bei schlechteren Fitnesswerten werden sie hingegen bestraft.

Schritt 4: **Schwarmintelligenz:** Mit den Gewichtungen der besten Konfigurationen und der aktuellen Generation werden die Beacons neu verteilt.

Schritt 5: **Totes Ende-Detektierung:** Falls keine Änderungen mehr eintreten, werden die Eigenschaften der Partikel zufällig mit denen anderer Individuen auf der Pareto-Front kombiniert.

## Paretofront

Mehrkriterielle Probleme werden häufig als Pareto-Optimierung, benannt nach dem italienischen Ingenieur, Ökonom und Soziologe *Vilfredo Federico Pareto*, bezeichnet. Die Optimierung verläuft dabei unter Betrachtung mehrerer Kostenfunktionen, wobei sich die Veränderungen der Kosten eines Problems auf die der anderen auswirkt. Jedoch bedingt die Verbesserung eines Kriteriums in der Beacon-Problematik die qualitative Verschlechterung des anderen. Wie weit diese Wechselwirkungen ausgeprägt sind, zeichnet sich beim Übertragen der Fitnesswerte in den gleichen mathematischen Raum ab. Dabei bildet sich eine Front, in dessen Richtung des negativen Gradienten keine weiteren Lösungen zu finden sind. Der Vorteil der in Abbildung 6.7 dargestellten Pareto-Front ist, die Möglichkeit zwischen den einzelnen Kriterien abzuwägen. Falls der Kunde eine qualitativ hochwertige und flächendeckende Lokalisierung benötigt, kann er sich aus der Menge an Lösungen eine für ihn ökonomische optimale Konfiguration auswählen. Für den Kunden mit geringeren Ansprüchen würden Konfigurationen mit weniger Beacons ausreichen, welche er sich unter den gleichen Optimalitätsbedingungen auswählen kann. Somit ist die Anforderungen an eine universelle optimale Planung erfüllt.

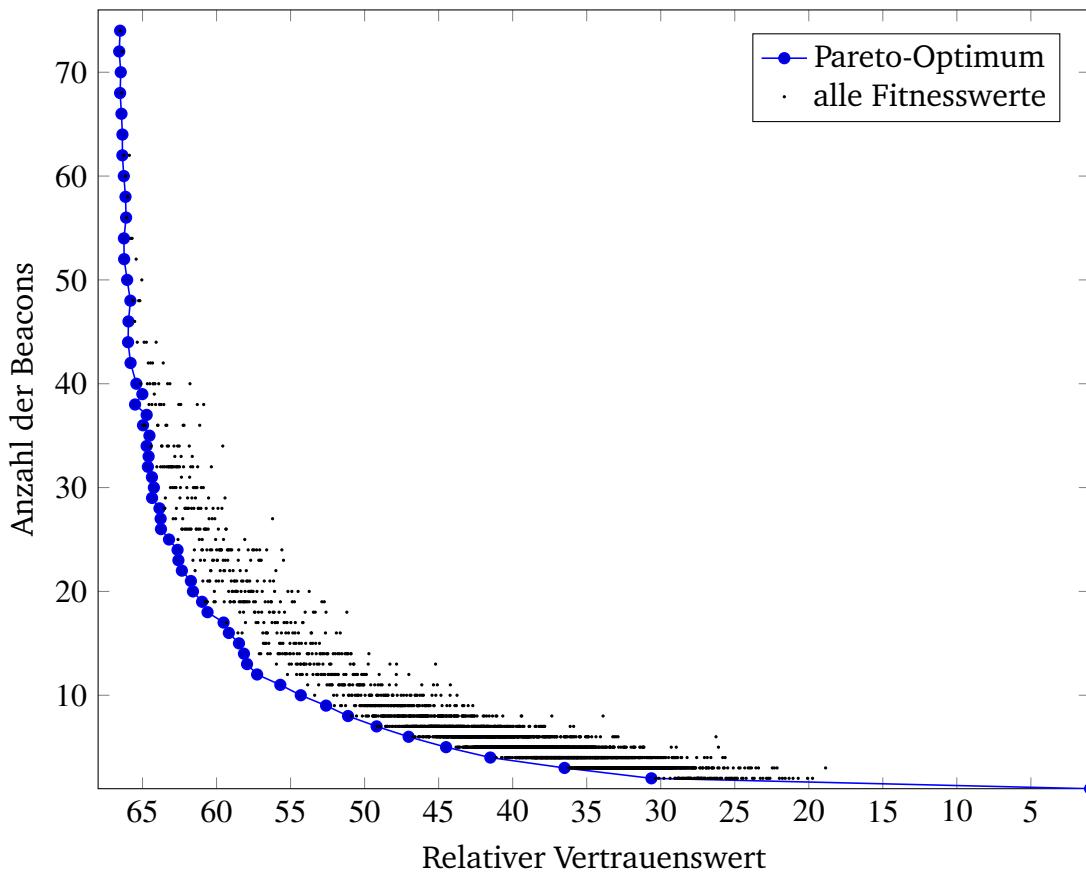


Abb. 6.7: Beispiel einer Paretofront

### 6.3.3 Simulationsbeispiel

Zum Abschluss der Thematik wurde in Abbildung 6.8 einmal eine komplette Simulation durchgeführt. Als Testobjekt dient dazu ein fiktiver Raum in quadratischer Form mit den Abmessungen von  $11,9 \times 7$  Metern. Zur Vereinfachung wurde die gleiche Höhe für die Sender wie für die Empfänger gewählt, welche denen in den statischen Experimenten gleichen. Als Toleranz für die Vertrauensskala wurde der Wert von einem Meter gewählt. Diese Wert wurde gleichsam für die Größe der Raumelemente übernommen, die schachbrettähnlich über die gesamte Grundfläche verteilt sind. In der echten Simulationsumgebung sind die Seiten der Unterteilungen gleich lang. Jedoch wurde das Bild für eine bessere Darstellung in der Darstellung gestreckt.

Die Optimierung wurde hier für die Zahl von 5000 Generationen durchlaufen und die Ergebnisse sind in der Paretofront im rechten unteren Teil des Bildes zu erkennen. Der Nutzer der Lighthouse Keeper-Simulation kann hier auf die einzelnen Konfigurationen der Pareto-Optimierung per Maus klicken. Dadurch entsteht im linken Teil die optimalen Beacon-Konfigurationen mit der zu erwartenden Qualität der Lokalisierung. Diese können als Textdatei mit den Positionsstandorten der Beacons und den einzelnen Vertrauenswerten für jedes Raumelement abgespeichert werden. Zur Veranschaulichung werden die Werte der berechneten Wahrscheinlichkeiten für gültige Signale farblich in den jeweiligen Raumteilungen markiert. Die Farbskala reicht dabei von blau für einen 100 prozentigen gültigen Empfang, über gelb für 50 Prozent und bis tiefrot für 0 Prozent gültige Signale.

Der Gedanke hinter dieser Aufführung ist es, die Signalqualität sichtbar zu machen und ein Gefühl für die Ausbreitungsverluste zu bekommen. Unabhängig von der Reichweite bzw. Signalleistung und der Frequenz der Sendeintervalle, will der Anwender eines Lokalisierungssystems schließlich die Sicherheit und Genauigkeit der Ortung betrachten können. Mit dieser Simulationsumgebung wird somit nicht nur die Planung realisiert, sondern auch die Machbarkeit eines auf Beacons basierendem Ortungssystems für eine erwünschte Genauigkeit und Dynamik untersucht. Somit kann schon alleine mit einer Grundrisszeichnungen eine vorläufige Planung durchgeführt werden und sich der Kunde ohne großen Aufwand einen ersten Eindruck über die Leistungsfähigkeit des Systems machen.

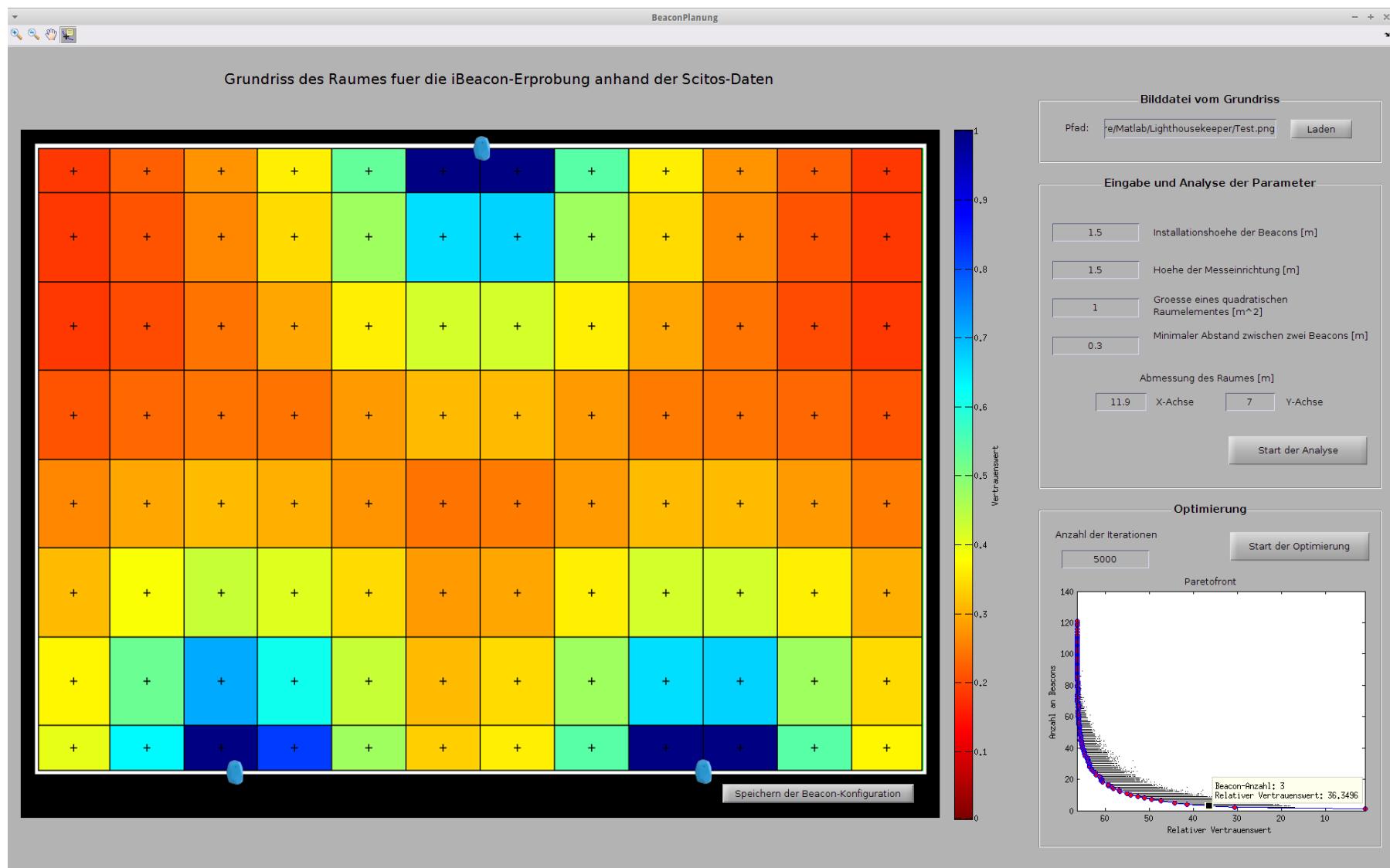


Abb. 6.8: Beispiel einer Lighthouse Keeper-Simulation

# 7 Evaluierung

Im letzten Teil dieser Arbeit wird mit den dynamischen Experimenten die letzte Brücke zwischen Simulation und Realität geschlagen. Dazu wird ein Raum mit einem Scitos G5 vermessen und diese Daten in die Simulationsumgebung eingepflegt. Aus den gewonnenen Informationen wird eine Konfiguration für drei Beacons im Raum optimiert, geplant und installiert. Anschließend werden einzelne Koordinaten im Raum mit dem Roboter autonom angefahren und die jeweilige Signalstärke eines Beacons gemessen. Dabei werden die vorher getroffenen Annahmen aus den statischen Experimenten mit den neuen Erkenntnissen aus der dynamischen Disziplin miteinander verglichen.

## 7.1 Dynamisches Experiment

In diesem letzten Experiment wird eine erstellte Beacon-Konfiguration auf ihre Ortungsgenauigkeit überprüft. Hierfür wird weniger auf die Vertrauensskala eingegangen, sondern die Signalstärken für einzelne Punkte im Raum aufgenommen, gespeichert und anschließend ausgewertet. In diesen Versuchen wird ausschließlich der Scitos G5 verwendet, weil er über eine eigene Lokalisierungsmöglichkeit verfügt und eine selbstständige Pfadplanung durchführen kann. Somit stellt seine Positionsschätzung die Referenz zu den beaconbasierten dar. Die Koordinaten die er ansteuert, kommen dabei aus der Simulationsumgebung als Mittelpunkte der Raumelemente. In diesem Test wird das ROS-Framework für die Kommunikation aller Geräte genutzt und wie in der Konzeptplanung in Abschnitt 3.2 gefordert wurde hier umgesetzt.

### 7.1.1 Durchführung

Im Experiment wird ein Adapter, der Daten zwischen dem Miracenter und ROS übertragen kann, benötigt. Dazu wird eine vorgefertige Software namens „Scitos Metralabs“ [43] verwendet, die die zweidimensionalen Raumkoordinaten vom Scitos an den ROS-Sever sendet und von diesem auch Zielkoordinaten entgegennehmen kann. Zur weiteren Vorbereitung zählt der Aufbau eines Raumes. Aufgrund des Mangels an großen Räumen ohne Interieur in der Universität, wurde kurzerhand ein Hörsaal umfunktionierte und die „Wände“ mit auf der Seite liegenden Tischen aufgebaut. Der dadurch entstandene Grundriss besitzt die Abmessungen von  $11,9 \times 7$  Meter. Dieser wird mit Hilfe des Roboters manuell kartografiert und als Bilddatei gespeichert (siehe Abbildung

7.1). Die Höhe der Tischkanten reicht dabei aus, um als Hindernis bzw. Wand vom dem Scitos wahrgenommen zu werden. Im Vergleich zur Zeichnung des Grundrisses ist in Abbildung 7.2 der Aufbau als Fotografie festgehalten. Dabei ist auch in der Mitte des Fotos der Scitos G5 zu erkennen.



Abb. 7.1: Grundriss des Testraumes



Abb. 7.2: Foto des Testraumes

Es wird dabei sehr gut ersichtlich, dass die künstlichen Wände nicht gerade verlaufen. Zudem ist der Grundriss mit kleinen Fragmenten und Fehlern behaftet, weswegen die Zeichnung in dieser Form nicht von der Simulationsumgebung verarbeitet werden kann. Deswegen muss jede Karte nachträglich geändert werden, indem die Wände begradigt und die kleinen Fragmente entfernt werden. Hierfür wurde ein normales Bildbearbeitungsprogramm verwendet und das Ergebnis aus der Änderung ist in Abbildung 7.3 zu sehen.

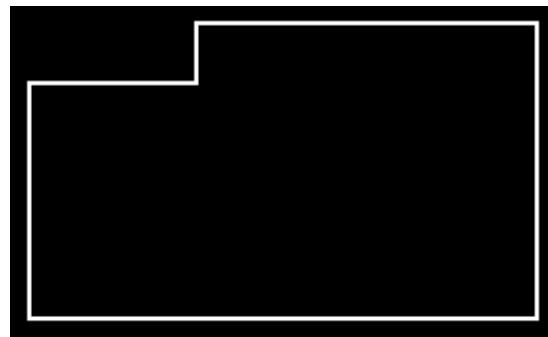


Abb. 7.3: Bearbeiteter Grundriss

### 7.1.2 Einsatz des Lighthouse Keepers

Aus dem erstellten Grundriss werden optimale Beacon-Konfigurationen mithilfe des Lighthouse Keepers erstellt. Die Wahl der Parameter orientiert sich hierbei an den Einstellungen des Testbeispiels aus Abschnitt 6.3.3. Lediglich die Toleranz wird auf 1,5 Meter erhöht, damit der Roboter sicher wenden und die vorgegebenen Positionen an-

steuern kann. Das Ziel der Planung in diesem Experiment ist es, alle verfügbaren Beacons anzubringen, damit in einem Durchlauf so viele Daten wie möglich gesammelt werden. Das Ergebnis der Simulation und Optimierung für drei Beacons findet sich in der unten stehenden Abbildung 7.4.

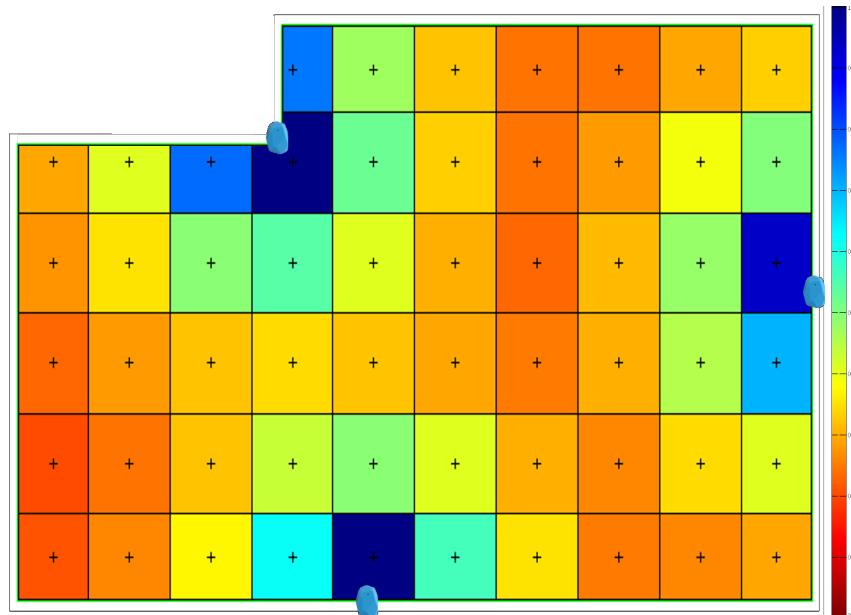


Abb. 7.4: Beacon-Konfiguration der dynamischen Experimente

### 7.1.3 Pfadplanung und Navigation

Die Positionen der Raumelemente und Befestigungspunkten werden in der Simulation als Textdokument gespeichert, sodass der Mira-ROS-Adapter ständig auf diese während des Experimentes zugreifen kann. Die Mittelpunkte der Raumelemente dienen als Kette von Punkten, die der Adapter einzeln an die Navigation des Scitos Roboters sendet. Speziell dafür wird eine Abfrage in den Adapter implementiert, die den nächsten Punkt auf der Trajektorie mit der aktuellen Position des Roboters vergleicht. Befindet sich der Roboter an den Koordinaten des angesteuerten Raumelementes, wird eine Wartezeit von zehn Sekunden vorgegeben. Nach Ablauf der Zeit überträgt der Adapter den nachfolgenden Punkt in der Textdatei und der Roboter fährt die nächsten Koordinaten an usw..

Die komplette Durchführung wurde in einem Zeitraffer-Video festgehalten und findet sich auf der beigelegten CD-ROM oder im Internet unter: [https://www.youtube.com/watch?v=\\_IeiKjpdh18](https://www.youtube.com/watch?v=_IeiKjpdh18).

## 7.2 Auswertung

Wie zum Anfang erwähnt wurde, setzt das Lighthouse Keeper-Prinzip auf den Scitos G5 und seiner internen Positionsbestimmung. Bei der Betrachtung der Positionsdaten vom Roboter fiel ein starker Drift der Daten bei fortschreitender Dauer des Experimentes auf. In Abbildung 7.5 ist dies einmal dargestellt. Die blaue Linie stellt die Bewegung des Roboters dar und der Startpunkt liegt mittig links. Im Video wird sehr gut ersichtlich, dass der Roboter alle Punkte entlang der Trajektorie zielsicher anfährt. In dem Experiment fuhr er dabei von oben links nach unten rechts. Die Daten, die der Roboter hingegen über seine Middleware an ROS weitergibt, spiegeln in keinster Weise das Verhalten aus dem Video wieder.

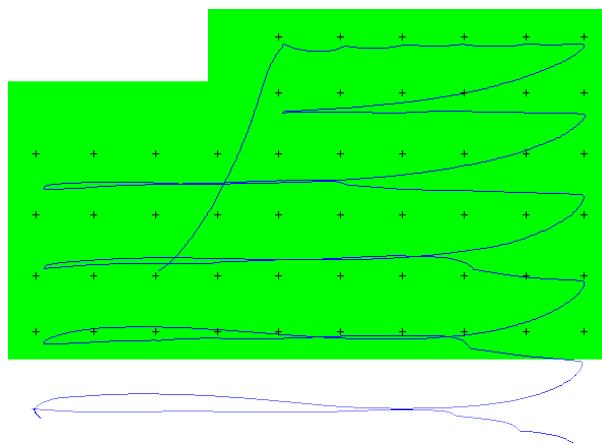


Abb. 7.5: Drift des Scitos G5

In der Abbildung ist zu erkennen, wie er Roboter (rein virtuell) den Raum sogar verlässt. Eine wahrscheinliche Begründung für die schlechte Ortung könnte das Interesse der Entwickler vom Miracenter sein, ihre Software und Ideen schützen zu wollen, indem sie die internen Vorgänge in der Navigations-Toolbox nicht preisgeben. Die gesendete Positionsdaten stammen mit hoher Wahrscheinlichkeit nur von den Odometrie-Sensoren. Die Verarbeitung aller Bewegungs- und Distanzsensoren geschieht (wahrscheinlich) nur für die eigene Navigation. Trotzdem müssen diese Daten für den Lighthouse Keeper aufbereitet werden, damit eine gute Referenzortung gegeben ist. Deswegen wird für die Beseitigung des Drifts ein weiteres Verfahren benötigt, das gerade diesen schätzt. Das Problem wird hier wieder als Optimierungsaufgabe formuliert. Die Kostenfunktion leitet sich dabei aus der Differenz eines Stopppunktes  $P$  des Roboters zu den Koordinaten des angestrebten Zielpunktes  $M$  (der Mittelpunkt von einem Raumelement) ab. Wiederum wird diese Funktion aus den bekannten Vorteilen quadriert. Die zu optimierenden Parameter  $a$  und  $b$  werden zur Berechnung des Drifts in eine Wachstumsfunktion eingesetzt.

Die Annahme dahinter ist das lineare Wachstum des Drifts mit zurückgelegtem Weg  $d$  in den x- und y-Richtung.

$$d_i = d_{i-1} + \sqrt{(P_i - P_{i-1})^2} \quad \text{mit} \quad d_0, P_0 = 0$$

$$P_{neu,i} = \begin{pmatrix} a \cdot d_i + P_{i,x} \\ b \cdot d_i + P_{i,y} \end{pmatrix}$$

$$\min_{a,b} J = \sum_{i=1}^n (M_i - P_{neu,i})^2$$

Als Optimierungsverfahren wird wieder der bewährte PSO genutzt. In den Abbildungen 7.6 bis 7.8 ist das Ergebnis der Bemühungen dargestellt. Es ist sehr gut ersichtlich, dass der lineare Ansatz funktioniert und im Vergleich zum Video das Verhalten sehr gut abgebildet ist. Weil der Zugriff auf die internen Positionsdaten vom Scitos verwährt bleibt, ist keine 100 prozentige Garantie gegeben, dass die Daten übereinstimmen. Für die Referenzgebung im Falle der Indoor-Lokalisierung reicht die gewonnene Qualität der Positionsbestimmung weitestgehend aus.

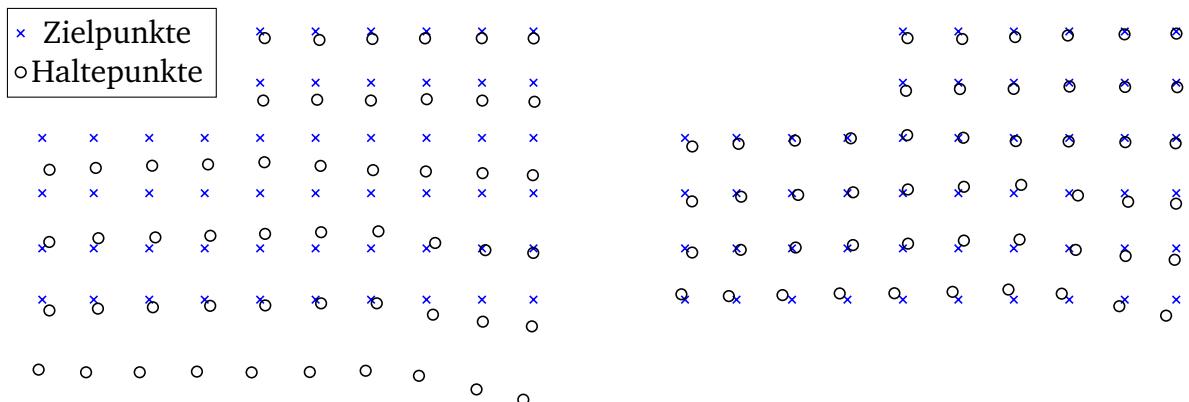


Abb. 7.6: Haltepunkte mit Drift

Abb. 7.7: Haltepunkte ohne Drift

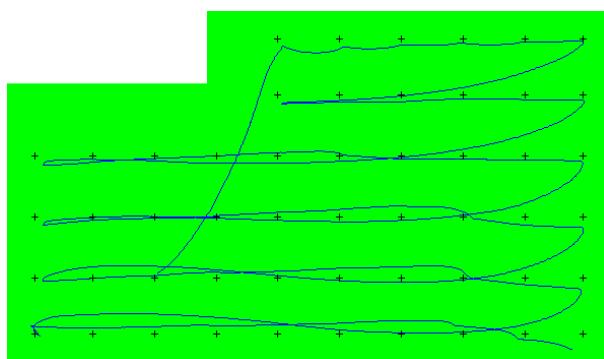


Abb. 7.8: Beseitigung vom Drift

Nachdem die driftlosen Positionsdaten zur Verfügung stehen, wird im Folgenden mit der Auswertung der Beacon-Signale begonnen. Als erstes wird dazu in den Abbildungen 7.9, 7.10 und 7.11 die Distanz von Roboter bzw. Smartphone zu einem Beacon über die Zeit aufgetragen und dazu die empfangene Signalstärke gegenüber gestellt. Ferner ist neben der gemessenen Signalstärke auch die geschätzte aus dem WIINER II-Modell in das Diagramm eingetragen. Die Messungen aus den dynamischen Experimenten verhalten sich dabei ähnlich wie die der statischen Versuchen. Signifikant ist dafür die Übereinstimmung von prädiktioniertem Ausbreitungsverlust zu dem realen. Lediglich nach zehn Minuten driften beide Verläufe voneinander weg. Dies wird daran liegen, dass der Drift aus der Odometrie nicht gänzlich verschwunden ist und dadurch der Referenzabstand ungenauer wird.

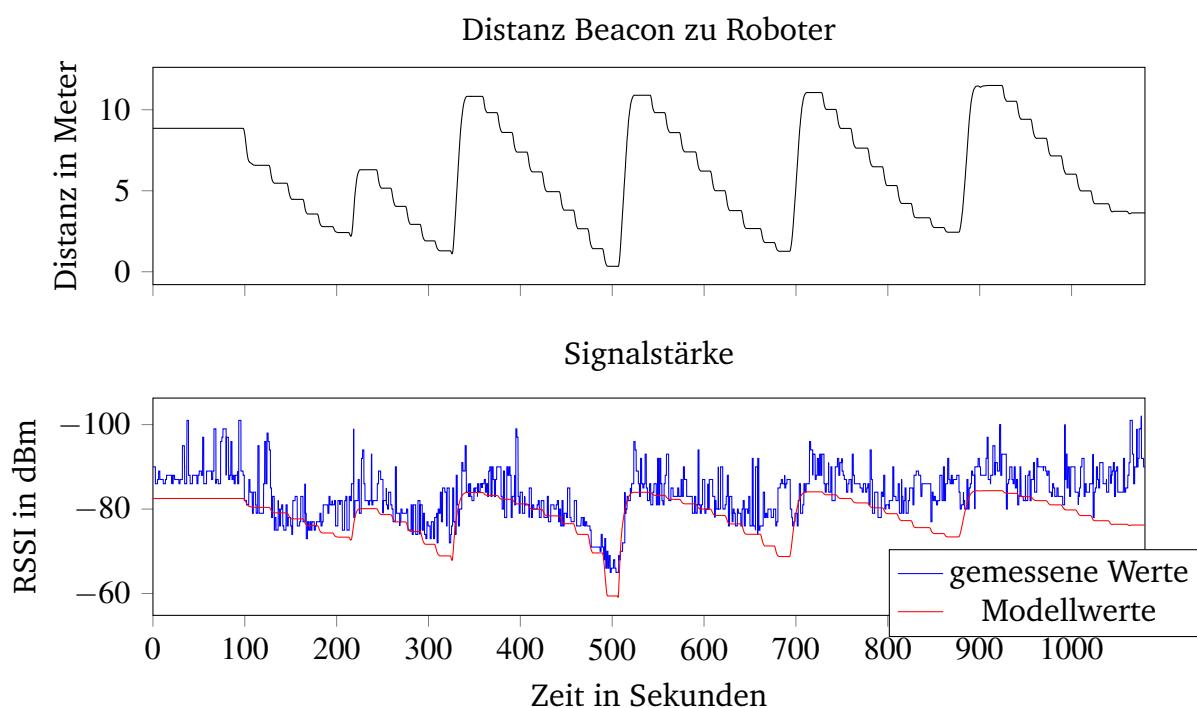


Abb. 7.9: Zeitlicher Verlauf der Signalstärke und des Abstandes für Beacon 1

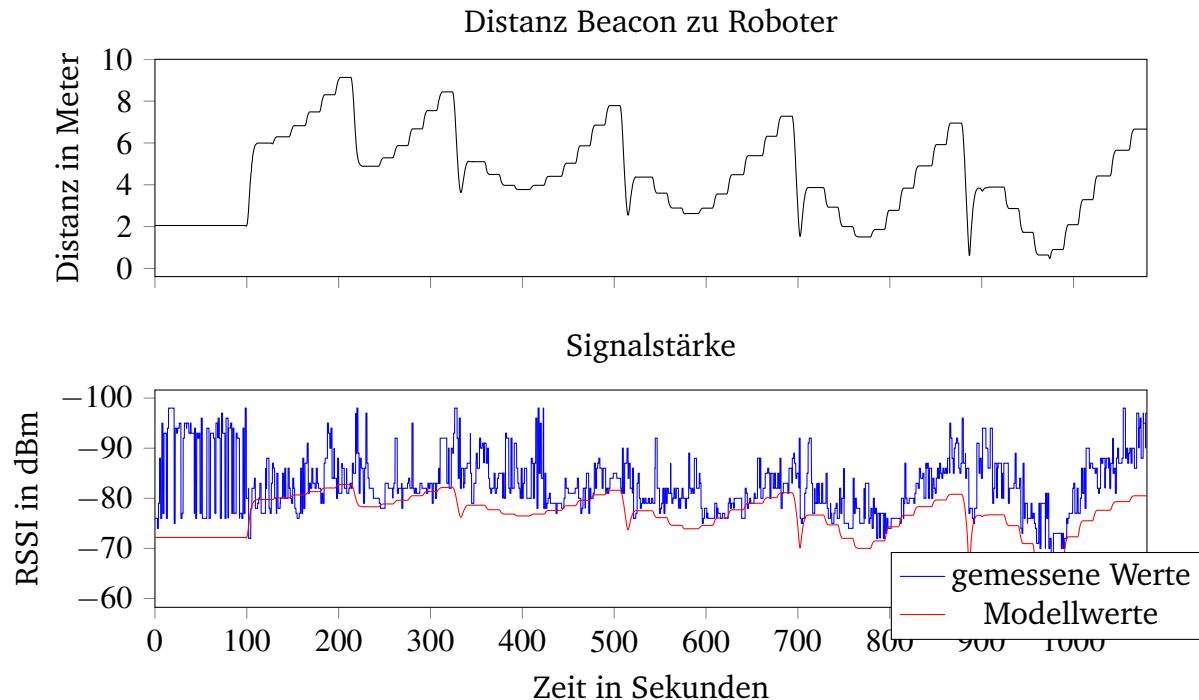


Abb. 7.10: Zeitlicher Verlauf der Signalstärke und des Abstandes für Beacon 2

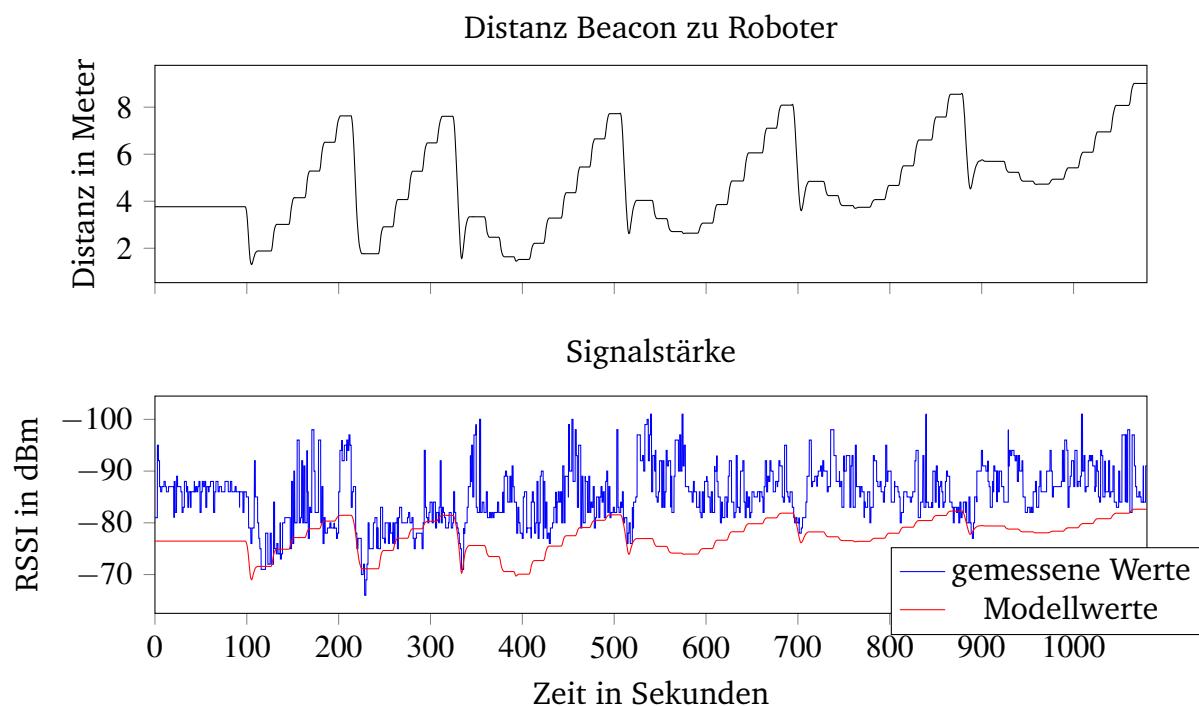


Abb. 7.11: Zeitlicher Verlauf der Signalstärke und des Abstandes für Beacon 3

## 7.3 Diskussion

Die Ergebnisse haben gezeigt, dass das Modell für die Ausbreitungsverluste sehr gut ausgelegt ist. Die Modellwerte bilden eine Grenze für die Maximalwerte der empfangbaren Signalleistung für alle Distanzen. Somit muss der Nutzer des Lokalisierungssystems lediglich auf das stärkste Signal warten. Die Simulationsumgebung positioniert dabei die Beacons so, dass die Wartezeiten auf ein gültiges Signal so kurz wie möglich sind. Jedoch konnte im Verlauf des dynamischen Experiments nicht auf die Vertrauensskala eingegangen werden. Hierbei wird der größte konzeptionelle Unterschied zwischen beiden Versuchsarten deutlich. Während mit den statischen Untersuchungen das Langzeitverhalten von Beacons untersucht wird, kommt es bei Evaluierung der Konfiguration auf die Geschwindigkeit an. Schließlich sollen ganze Gebäude auf die vollständige Abdeckung mit Beacons überprüft werden und der Testraum mit seinen 80 m<sup>2</sup> hat über 20 Minuten in Anspruch genommen. Die Zeit die ein Roboter für die Evaluierung benötigt, könnte sicherlich noch verkürzt werden, indem die Raumelemente größer ausgelegt werden und die Wartezeit an einem Haltepunkt minimiert wird. Es hat sich bei der Auswertung der Daten ergeben, dass keine Unterschiede zwischen dem ruhenden Roboter und dem fahrenden Roboter bezüglich des Signalempfanges bestehen. Die Streuung der RSSI-Werte hat in der Fahrt weder zugenommen, noch hat sich die Empfangsqualität verschlechtert.

# 8 Zusammenfassung und Ausblick

In Zukunft werden Indoor-Lokalisierungssysteme eine immer wichtigere Rolle einnehmen. Im Hinblick auf die Vernetzung der Gesellschaft gehört sie in die heutige Zeit. Derzeit sind jedoch noch viele Herausforderungen zu meistern, speziell in Anbetracht einer effektiven Planung und eines ökonomischen Umgangs mit dieser Technologie. Dafür müssen mehrere Schwerpunkte verfolgt werden:

- Erstellung geeigneter Werkzeuge für Experimente
- Standardisierbare Verfahren für Messungen
- Definition von Gütekriterien für Lokalisierungssysteme

Die vorliegende Arbeit befasst sich mit genau dieser Problematik. Das Ziel ist es, die Infrastruktur für ein Lokalisierungssystem anhand von fest definierten Kriterien optimal auszulegen. Zudem wird die Möglichkeit und Methodik einer vollautomatisierten Validierung eines Indoor-Ortungssystems beschrieben.

Bisherige Ansätze betrachten immer nur einen Teil der Herausforderungen, aber es fehlt ein komplettes Konzept dafür, das jeden Aspekt berücksichtigt und beherrscht. Deshalb erforderte es, ein eigenes Konzept zu entwickeln und einen tieferen Einblick in die Thematik zu wagen. Dieses Vorhaben wird Lighthouse Keeper genannt und umfasst die Planung von benötigter Hardware und Software. Um die Daten so nah wie möglich am Endverbraucher auszuwerten, wurde ein Smartphone als Empfangsgerät genutzt und eigens dafür eine Applikation geschrieben. Als Helfer für die Versuche wurden zwei Roboter, Youbot und Scitos G5 akquiriert, durch den der Prozess standardisierbar wurde. Die Kommunikation zwischen allen Plattformen wird mithilfe des Frameworks ROS verwirklicht. Mit der nötigen Hard- und Software konnten die ersten Messwerte aufgenommen werden und aus ihnen ein Modell des Ausbreitungsverlustes von BLE-Signalen erschlossen werden. Mit diesem Modell konnten aus den empfangenen RSSI-Werten die Distanz zwischen Sende- und Empfangsgerät geschätzt werden. Die softwaretechnische Auswertung fand dabei stets offline am PC statt. Deswegen lag der Fokus zunächst auf statischen Entfernungen. Die Langzeituntersuchungen von Beacon-Signalen häuften dabei Datensätze von mehr als 30 Stunden an, aus denen eine Vertrauensskala für die einzelnen Distanzen erstellt wurde. Die definierte Skala gibt an, wie lange an einer Distanz zu einem Beacon auf ein für das Modell gültiges Signal gewartet werden muss. Zudem wurden auch Untersuchungen für störende Faktoren eingeleitet, um Unsicher-

heitsfaktoren bei den Messungen zu minimieren. Im Anschluss wurde mithilfe der Vertrauensskala ein Optimalitätskriterium aufgestellt, anhand dessen die Beacons in einem Raum optimal verteilt werden. Der Prozess aus Experimenten, Modellierung und Simulation wird mit einer Evaluierung einer optimalen Beacon-Konfiguration gekrönt. Die Ergebnisse aus dem automatisierten dynamischen Experiment beweisen die Zuverlässigkeit des Modells und der Annahmen. Dadurch wurde die Funktionsfähigkeit und Korrektheit des Konzeptes bewiesen.

Die Ausführungen sind den Aufgaben in Bezug auf die Planung von Indoor-Infrastrukturen und der Durchführung von dynamischen Experimenten gerecht geworden. Es wurden alle Anforderungen bedient und es sind damit sehr gute Ergebnisse erreicht worden. Im Falle einer Weiterführung dieser Arbeit bieten sich folgende Schwerpunkte an:

- Validierung des Konzeptes unter Veränderungen der Beacon-Einstellungen (Sendeleistung und Sendefrequenz). Unter Betrachtung dieser Änderungen müssen ferner ökonomische Gutachten erstellt werden, die in die Optimierung mit einfließen.
- Erweiterung der Simulationsumgebung um Objekte im Raum. Zudem müssen die Befestigungen auch manuell gewählt werden können und die Beacon-Einstellungen zusätzlich auswählbar sein.
- Verzicht auf proprietärer Software durch Entwicklung eigener Programme. Denn die Probleme mit der App und der mangelnde Zugriff auf viele Funktionen auf den verschiedenen Plattformen haben im Endeffekt die Arbeit erschwert. Der Effekt der Zeitersparnis durch die Verwendung von fertiger Software, wurde nachträglich durch die verursachten Probleme der fremden Software wieder aufgezerrt.
- Erweiterung der Optimierung um Gewichtungen für die einzelnen Raumelemente. Dadurch kann der Fokus auf kritische Bereiche (z.B. Eingänge und Besonderheiten) gelegt werden.
- Weitere Untersuchungen des Ausbreitungsverlustes für größere Distanzen durchführen. Für größere Räume ist es ungewiss, ob das Modell noch dafür ausgelegt ist. Hierbei müssen die statischen Experimente erneut durchgeführt werden.

Der bisherige Mangel an einem umfassenden Planungskonzept wird hoffentlich mit dieser Arbeit zum Teil beseitigt. Es wäre wünschenswert, wenn diese Arbeit die Beacon-Technologie weiter voranbringt und andere Menschen für die weiterführende Forschung in diesem Bereich inspiriert.

# Literaturverzeichnis

- [1] 5280MOBILE.NET: What is iBeacon. 2011. – <http://5280mobile.net/ibeacon-proximity/>, Aufgerufen am 26.03.2015.
- [2] A. ATHALYE ; V. SAVIC ; M. BOLIC ; P. M. DJURIC: Novel Semi-Passive RFID System for Indoor Localization,. In: IEEE Sensors Journal 13 (2013), Nr. 2, S. 528–537. – <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6310003>, Aufgerufen am 19.03.2015.
- [3] A. CAVALLINI: iBeacon Bible 2.0. 2015. – <https://meetingofideas.files.wordpress.com/2014/06/ibeacon-bible-2-0.pdf>, Aufgerufen am 17.03.2015.
- [4] A. SCHIMSCHAR: Modellierung des Zeit- und Fehlerverhaltens industrieller Funklösungen zur Bewertung der Koexistenz, Otto-von-Guericke-Universität Magdeburg, Diss., 2013
- [5] A. WARSKI: How do iBeacons work? 2014. – <http://www.warski.org/blog/2014/01/how-ibeacons-work/>, Aufgerufen am 27.03.2015.
- [6] AMAZON.COM, INC.: Motorola Moto G. 2013. – [http://www.amazon.de/Motorola-Smartphone-HD-Display-Megapixel-Quad-Core-Prozessor-Schwarz/dp/B00GJG0Q0I/ref=sr\\_1\\_1?ie=UTF8&qid=1428844430&sr=8-1&keywords=Motorola+Xt1032](http://www.amazon.de/Motorola-Smartphone-HD-Display-Megapixel-Quad-Core-Prozessor-Schwarz/dp/B00GJG0Q0I/ref=sr_1_1?ie=UTF8&qid=1428844430&sr=8-1&keywords=Motorola+Xt1032), Aufgerufen am 12.04.2015.
- [7] APPLE, INC.: Getting Started with iBeacon. 2014. – <https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf>, Aufgerufen am 25.03.2015.
- [8] B. KLUG: Motorola Moto G Review. In: www.anandtech.com (2013). – <http://www.anandtech.com/show/7586/motorola-moto-g-review/7>, Aufgerufen am 26.03.2015.
- [9] BLUETOOTH SIG, INC.: SIG INTRODUCES BLUETOOTH LOW ENERGY WIRELESS TECHNOLOGY, THE NEXT GENERATION OF BLUETOOTH WIRELESS TECHNOLOGY. In: bluetooth.com (2009). – <http://www.bluetooth.com/Pages/Press-Releases-Detail.aspx?ItemID=4>, Aufgerufen am 16.03.2015.

- [10] BLUETOOTH SIG, INC.: Bluetooth Smart Logo. 2014. – <http://www.bluetooth.com/Pages/Bluetooth-Brand.aspx>, Aufgerufen am 15.03.2015.
- [11] D. E. DILGER: Inside iOS 7: iBeacons enhance apps' location awareness via Bluetooth LE. In: appleinsider.com (2013). – <http://appleinsider.com/articles/13/06/19/inside-ios-7-ibeacons-enhance-apps-location-awareness-via-bluetooth-le>, Aufgerufen am 15.03.2015.
- [12] E. ZITZLER ; M. LAUMANNS ; L. THIELE: SPEA2: Improving the Strength Pareto Evolutionary Algorithm / Swiss Federal Institute of Technology (ETH) Zurich. 2001. – Forschungsbericht
- [13] ESTIMOTE, INC.: Estimote Beacon Illustration 2. 2015. – <http://estimote.com/assets/gfx/press/press-beacon-illustration-2.cd71f93a.png>, Aufgerufen am 17.03.2015.
- [14] ESTIMOTE, INC.: Estimote Indoor Localization. 2015. – <http://estimote.com/indoor/>, Aufgerufen am 26.03.2015.
- [15] ESTIMOTE, INC.: Estimote SDK for Android. 2015. – <https://github.com/Estimote/Android-SDK>, Aufgerufen am 14.04.2015.
- [16] G. BRAUNBERGER: Macht der Maschinen. 2014. – <http://www.faz.net/aktuell/wirtschaft/menschen-wirtschaft/digitale-revolution-macht-der-maschinen-12910372.html>, Aufgerufen am 26.03.2015.
- [17] GOOGLE, INC.: Android Studio. 2014. – <http://developer.android.com/tools/studio/index.html>, Aufgerufen am 14.04.2015.
- [18] GOOGLE, INC.: Google Indoor Maps. 2015. – <https://www.google.com/maps/about/partners/indoormaps/>, Aufgerufen am 22.03.2015.
- [19] H.-C. DIRSCHERL: Stau-Warnung - Diese Verkehrslage-Dienste gibt es für Autofahrer. In: http://www.pcwelt.de (2015). – <http://www.pcwelt.de/ratgeber/Stau-Warnung-Google-Maps-Tomtom-Verkehrslage-Echtzeitverkehrsinformationen-373385.html>, Aufgerufen am 26.03.2015.
- [20] HTTP://WWW.BUNDESNETZAGENTUR.DE: Nutzungsbestimmungen für die Nutzung durch die Allgemeinheit für ISM-Anwendungen. 2003. – [http://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Telekommunikation/Unternehmen\\_Institutionen/F](http://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/F)

requenzen/Allgemeinzuteilungen/2003\_76\_ISM\_pdf.pdf?\_\_blob=publicationFile&v=5, Aufgerufen am 25.03.2015.

- [21] HTTP://WWW.ITWISSEN.INFO: Übersicht über die verschiedenen Funktechnologien. o.J.. – <http://www.itwissen.info/definition/lexikon/ultra-low-power-Bluetooth-ULP-ULP-Bluetooth.html>, Aufgerufen am 24.03.2015.
- [22] J. BLANKENBACH ; A. NORRDINE: Indoor-Positionierung mit künstlichen Magnetfeldern. In: zfv 1 (2013). – [geodesie.info/sites/default/files/privat/zfv\\_2013\\_1\\_Blanckenbach\\_Norrdine.pdf](http://geodesie.info/sites/default/files/privat/zfv_2013_1_Blanckenbach_Norrdine.pdf), Aufgerufen am 14.03.2015.
- [23] J. MCCAFFREY: Particle Swarm Optimization. In: MSDN Magazine (2011). – <https://msdn.microsoft.com/en-us/magazine/hh335067.aspx>, Aufgerufen am 23.04.2015.
- [24] KUKA AG: Datenblatt Youbot. 2014. – [http://www.kuka-labs.com/NR/rdonlyres/4833867A-D24F-410A-9AE4-300FBF671DFD/0/youBot\\_datenblatt\\_web\\_0514.pdf](http://www.kuka-labs.com/NR/rdonlyres/4833867A-D24F-410A-9AE4-300FBF671DFD/0/youBot_datenblatt_web_0514.pdf), Aufgerufen am 26.03.2015.
- [25] M. QUIGLEY ; B. GERKEY ; K. CONLEY ; J. FAUST ; T. FOOTE ; J. LEIBS ; E. BERGER ; R. WHEELER ; A. NG: ROS: an open-source Robot Operating System / Stanford University. 2009. – Forschungsbericht. – <https://www.willowgarage.com/sites/default/files/icraoss09-ROS.pdf>, Aufgerufen am 14.04.2015.
- [26] MATHWORKS, INC.: MATLAB GUI. 2015. – <http://de.mathworks.com/discovery/matlab-gui.html>, Aufgerufen am 27.04.2015.
- [27] METRALABS GMBH: Scitos G5. 2012. – [http://metralabs.com/index.php?option=com\\_content&view=article&id=70&Itemid=64](http://metralabs.com/index.php?option=com_content&view=article&id=70&Itemid=64), Aufgerufen am 13.04.2015.
- [28] MICROSOFT RESEARCH: Mobile Indoor Localization. 2015. – <http://research.microsoft.com/en-us/projects/indoorloc/>, Aufgerufen am 22.03.2015.
- [29] MOBILE-STUDIEN.DE: Marktanteile mobiler Betriebssysteme Q1 2014. 2014. – <http://mobile-studien.de/marktanteile-betriebssysteme/marktanteile-mobiler-betriebssysteme-q1-2014/>, Aufgerufen am 16.03.2015.

- [30] N. GOLMIE ; R.E. VAN DYCK ; A. SOLTANIAN ; A. TONNERRE ; O. REBALA: Interference Evaluation of Bluetooth and IEEE 802.11b Systems. In: Wireless Networks 9 (2003)
- [31] NORDIC SEMICONDUCTOR: nRF51822 Datasheet. o.J.. – <http://www.nordicsemi.com/eng/Products/Bluetooth-Smart-Bluetooth-low-energy/nRF51822>, Aufgerufen am 27.03.2015.
- [32] P. KYÖSTI ET AL: WINNER II Channel Models / WINNER. 2007. – Forschungsbericht. – <http://www.ist-winner.org/WINNER2-Deliverables/D1.1.2v1.1.pdf>, Aufgerufen am 23.04.2015.
- [33] P. LAWITZKI: Android Sensor Fusion Tutorial. 2014. – <http://www.codeproject.com/Articles/729759/Android-Sensor-Fusion-Tutorial>, Aufgerufen am 16.04.2015.
- [34] R. MARDENI ; S. N. OTHMAN: Node Positioning in ZigBee Network Using Trilateration MethodBased on the Received Signal Strength Indicator (RSSI). In: European Journal of Scientific Research (2010). – [http://godieboy.com/wp-content/uploads/2012/05/ejsr\\_46\\_1\\_05.pdf](http://godieboy.com/wp-content/uploads/2012/05/ejsr_46_1_05.pdf), Aufgerufen am 31.03.2015.
- [35] ROSJAVA ORGANISATION: Rosjava. 2015. – <http://wiki.ros.org/rosjava>, Aufgerufen am 15.04.2015.
- [36] ROS.ORG: About ROS. 2015. – <http://www.ros.org/about-ros/>, Aufgerufen am 16.04.2015.
- [37] R...@THEMMTEAM.COM: Nexus 4 loses wifi connection when bluetooth is in use. 2012. – <https://code.google.com/p/android/issues/detail?id=41631>, Aufgerufen am 18.04.2015.
- [38] SONY CORPORATION: Datenblatt CR2450B. o.J.. – <http://www.sony.net/Products/MicroBattery/cr/pdf/cr2450b.pdf>, Aufgerufen am 25.03.2015.
- [39] TECHNISCHE UNIVERSITÄT BERLIN ; ADVANTIC SISTEMAS Y SERVICIOS S.L. ; iMINDS ; SICS SWEDISH ICT AB ; TELEVIC HEALTHCARE: Evaluation Of RF-Based Indoor Localization Solutions For The Future Internet (EVARILOS). 2015. – <http://www.evarilos.eu>, Aufgerufen am 22.03.2015.
- [40] W. L.GOFFE ; G. D. FERRIER ; J. ROGERS: Global optimization of statistical functions with simulated annealing. In: Journal of Econometrics 60 (1994).

- <http://www.cpdee.ufmg.br/~joao/OtimMultiobjetivo/SimulatedAnnealing/Golfe1999.pdf>, Aufgerufen am 25.04.2015.
- [41] **WWW.CONTECHLAB.COM:** iBeacon Airport. 2014. – <http://www.contechlab.com/ibeacon-application-in-airport/>, Aufgerufen am 26.03.2015.
- [42] **Y. CHEN ; D. LYMBEROPoulos ; J. LIU ; B. PRIYANTHA:** FM-based Indoor Localization. 2012. – <http://research.microsoft.com/pubs/163038/sys029fp-Chen.pdf>, Aufgerufen am 14.03.2015.
- [43] **Y. GATSOULIS ; C. BURBRIDGE ; L. RIANO ; F. KOLBE:** Scitos Metralabs. 2013. – [http://wiki.ros.org/scitos\\_metalabs](http://wiki.ros.org/scitos_metalabs), Aufgerufen am 16.04.2015.



# Anhang

## A Bilder

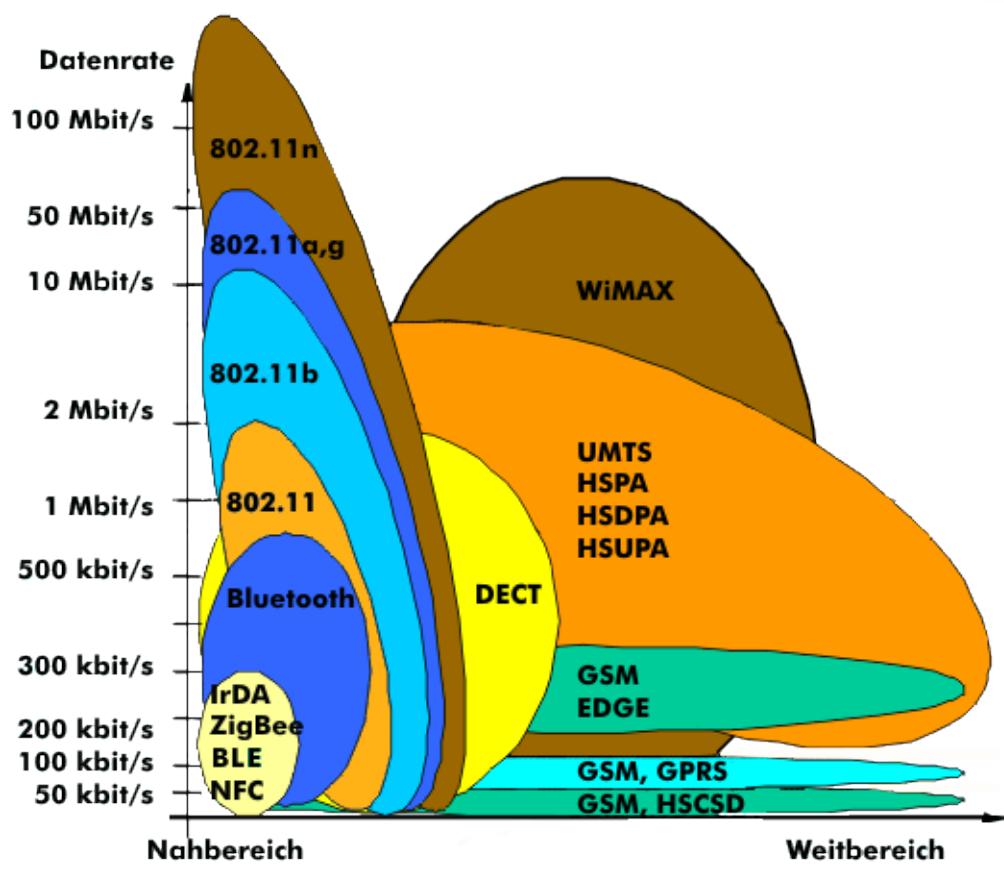


Abb. A.1: Überblick zu heutigen Funktechnologien [21]



Abb. A.2: Distanz-Signalstärke-Messung Bild 1

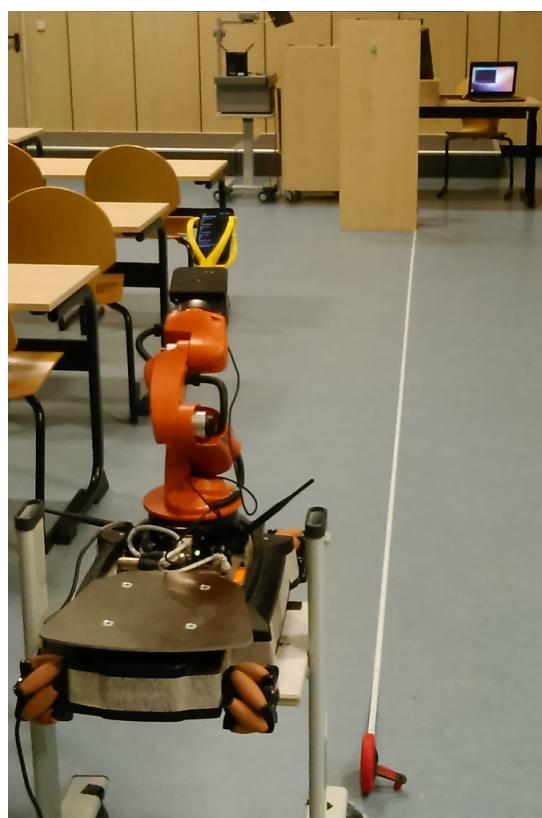


Abb. A.3: Distanz-Signalstärke-Messung Bild 2



Abb. A.4: Distanz-Signalstärke-Messung Bild 3



# **Selbstständigkeitserklärung**

Hiermit versichere ich,

.....  
(Name, Vorname) .....  
(Matrikel-Nr.)

dass ich die vorliegende Masterarbeit mit dem Thema:

.....  
.....  
.....

selbstständig und ohne unerlaubte Hilfe angefertigt habe. Alle verwendeten Hilfsmittel und Quellen sind im Literaturverzeichnis vollständig aufgeführt und die aus den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht.

---

Ort, Datum

Unterschrift