

Simon Parlow

**Evaluation verschiedener robuster
Lokalisierungsalgorithmen mit
einem virtuellen
Phasendifferenz-Distanzsensor**



FAKULTÄT FÜR
INFORMATIK

Institut für Intelligente Kooperierende Systeme

Bachelorarbeit

**Evaluation verschiedener robuster
Lokalisierungsalgorithmen mit einem virtuellen
Phasendifferenz-Distanzsensor**

Autor: Simon Parlow Sommersemester 2016

Professor: Prof. Dr. Mostaghim

Betreuer: Christoph Steup

Simon Parlow: *Evaluation verschiedener robuster Lokalisierungs-
algorithmen mit einem virtuellen Phasendifferenz-Distanzsensor*
Otto-von-Guericke-Universität
Magdeburg, 2016.

Inhaltsverzeichnis

Abbildungsverzeichnis	III
Tabellenverzeichnis	V
Abkürzungsverzeichnis	VIII
1 Einleitung	1
2 Grundlagen	3
2.0.1 ZIGPOS Positioning Kit	3
2.0.2 Gaußsches Mischmodell	5
2.0.3 Lokalisierungsalgorithmus	6
3 Verwandte Arbeiten	13
4 Konzept	17
4.1 Algorithmen sichten und sammeln	17
4.2 Algorithmen beschreiben und klassifizieren	18
4.3 Entwickeln der Evaluationssoftware	21
4.4 Erstellen des Phasendifferenzdistanzgenerators	22
4.5 Durchführung der Experimente	23
5 Implementierung	25
5.1 Entwicklung einer modularen Evaluationssoftware	25
5.1.1 Anforderungen an die Software	26
5.1.2 Aufbau der Software	26
5.1.3 Nutzung der Software	28
5.2 Erstellung eines virtuellen Phasendifferenzdistanzgenerator	28
5.2.1 Sammlung der Messdaten	29

5.2.2	Aufwertung der Messdaten	29
5.2.3	Finden eines gaußschen Mischmodells	32
5.3	Erstellung der Experimente	34
6	Evaluation	39
6.1	Fehlerhafte MDS-Verfeinerung	39
6.2	Variierende Ankeranzahl	40
6.2.1	Erwartungen	40
6.2.2	Ergebnis	41
6.2.3	Schlussfolgerung	42
6.3	Variierender Rauschfaktor	43
6.3.1	Erwartungen	43
6.3.2	Ergebniss	44
6.3.3	Schlussfolgerung	45
6.4	Diskussion	46
7	Zusammenfassung	49
	Literaturverzeichnis	59

Abbildungsverzeichnis

2.1	Remote-Ranging zwischen Knoten [18]	4
2.2	Ablauf des Ranging zwischen Knoten [6]	5
2.3	Lokalisierungsablauf [1]	8
5.1	Klassendiagramm der Evaluationssoftware	27
5.2	Gefilterte Datenmenge für einen Filter mit minimalem DQF-Wert von 90	31
5.3	Gefilterte Datenmenge	32
5.4	Zu hohe Komponentenanzahl	34
5.5	Zu geringe Komponentenanzahl	35
5.6	Passende Komponentenanzahl	36
6.1	Unterschied zwischen MDS und MDS mit Trilateration für verschiedene Iterationen	40
6.2	Mittlerer Positionierungsfehler bei variierender Ankeranzahl	42
6.3	Varianz der Distanzmessungen für verschiedene Rauschfaktoren	44
6.4	Distanzmessungen für einen Rauschfaktor von 5	45
6.5	Mittleres Positionierungsfehler bei variierendem Rauschfaktor	46

Tabellenverzeichnis

4.1	Klassifizierung aktueller Veröffentlichungen im Bereich der Sensornetzwerke	18
4.2	Algorithmenklassifizierungen	24
5.1	Messfilter	30
5.2	Unterschiedliche Komponentenanzahl für Mischmodelle	37
5.3	Verschiedene Grundparameter für Experimente	38
6.1	Mittlerer Positionierungsfehler bei variierender Ankeranzahl	42
6.2	Mittleres Positionierungsfehler bei variierendem Rauschfaktor	46

Abkürzungsverzeichnis

GPS	Global Positioning System
REB	Radio Extender Board
SMA	SubMiniature version A
DQF	Distance Quality Faktor
GMM	Gaußsches Mischmodell
ToA	Time of Arrival
TDoA	Time Difference of Arrival
RSSI	Received Signalstrength Indicator
MDS	Multi-dimensional Scaling
RLS	Robust Least-Squares
RLS+	Robust Least-Squares mit Optimierung
dwMDS	distributed weighted Multidimensional Scaling
MDS-MAP(P)	MDS-MAP using patches of relative maps
PDM	Proximity Distance Mapping
AHLoS	Ad-Hoc Localization System
MMSE	Minimum mean square error
MMDS	ModifiedMultidimensional Scaling

MSM Malguki SpringModel

WMDS Weighted Multidimensional Scaling

MLE Maximum Likelihood Estimation

MLE Comma-separated values

EM-Algorithmus Expectation-Maximization-Algorithmus

AIC Akaike information criterion

BIC Bayesian Information Criterion

CSV Comma-separated values

MDS+ Multi-Dimensional-Scaling mit Trilateration durch Anpassung

1 Einleitung

In den letzten Jahren haben Sensornetzwerke ein großes Forschungsinteresse auf sich gezogen. Diese Netzwerke bestehen aus verteilten, unabhängigen Sensorknoten welche physikalische Werte ihrer Umgebung messen. Ein Sensorknoten ist dabei grundsätzlich aus einem leistungsschwachen Prozessor, einer kleinen Menge an Speicher, einem Modul zur drahtlosen Kommunikation sowie einem Sensorboard aufgebaut [28]. Solche Knoten sind kostengünstig, klein und haben eine sehr begrenzte Rechenleistung und Energie. Mithilfe dieser Knoten ist es möglich Sensornetzwerke groß skalieren zu lassen und somit Netzwerke mit Tausenden von Knoten zu erstellen.

Durch Entwicklungen im Bereich der Elektronik und der drahtlosen Kommunikation wurden viele neue Anwendungsbereiche für Sensornetzwerke erschlossen. Sensornetzwerke können zur Überwachung von Gebieten eingesetzt werden um Regenmenge und Wasserstand zu messen oder Waldbrände, Fluten oder weitere natürliche Katastrophen frühzeitig zu erkennen [74, 26]. Auch der Einsatz im Gesundheitsbereich zur Kontrolle von Patienten oder die Nutzung von Sensorknoten zur Beobachtung von Wildtieren wird erforscht [45, 77]. Weitere mögliche Anwendungsbereiche sind das Bildungswesen [72] oder die Erkennung von Geräteschäden [38].

Das Aufgabenfeld eines Sensornetzwerks ist immer von dem jeweiligen Anwendungsbereich abhängig, jedoch ist der Bedarf an Positionsinformationen für die Sensorknoten durchgängig präsent. Drahtlose Geräte haben im Allgemeinen kein Wissen über ihren Standort, jedoch tragen die von den Knoten gesammelten Daten oftmals nur einen sehr geringen Informationswert oder sind gar ganz bedeutungslos ohne die Kenntnis wo diese Daten gesammelt worden sind. Auch ermöglicht das Wissen über die Position eines Knotens noch eine weitere große Anzahl von möglichen Einsatzszenarien für Sensornetzwerke. So beruhen Funktionen wie das geographische Routing oder das Detektieren und Verfolgen von Zielen zu diesen neu erschlossenen Anwendungsbereichen [24].

Die Fähigkeit zur Selbstlokalisierung, der eigenständigen Schätzung und Berechnung der Position, ist eine wünschenswerte Eigenschaft eines Sensornetzwerks und stellt eine grundsätzliche Problematik dar. Eine effiziente Lokalisierungsmethode ist eine der Hauptinteressen im Forschungsgebiet der drahtlosen Sensornetzwerke. Intuitive Ansätze zur Lösung des Lokalisierungsproblems bestehen in der Positionsbestimmung durch Global Positioning System (GPS) oder durch eine manuelle Konfiguration der Sensorknoten. Diese beiden Ansätze sind jedoch unpraktikabel in realistischen Anwendungen. GPS ist eine bekannte und weit verbreitete Lokalisierungsmethode, jedoch ist die dafür notwendige Hardware zu teuer und energiekonsumierend für den Einsatz mit Sensorknoten. Auch ist die Präzision der Positionsbestimmung zu gering und insbesondere abhängig vom Einsatzbereich. Innerhalb von Gebäuden, im Untergrund oder in dichten Wäldern kann GPS nicht eingesetzt werden. Auch die manuelle Konfiguration ist aufgrund der Größe von Sensornetzwerken sowie des Einsatzgebietes und der Art der Stationierung der Knoten nicht praktikabel.

Zur Lösung dieses Problems wurde eine Menge von Lokalisierungsalgorithmen entwickelt, welche mithilfe des Netzwerkes in welchem sich die Knoten befinden, den Standort der Sensorknoten berechnen sollen. Bei den meisten dieser Algorithmen wird dabei mithilfe der Position einiger Nachbarknoten und der zu ihnen gemessenen Entfernungen oder Winkel die Position des Knotens berechnet.

Besonders innerhalb von Gebäuden ist das Lokalisierungsproblem bedingt durch die Umgebung schwer lösbar. Viele Algorithmen beruhen bei der Distanzmessung auf der Signalstärke des Radiosignals, welche in Räumen allerdings stark schwanken kann und durch viele Faktoren beeinflusst wird. In dieser Arbeit sollen für die Indoor-Lokalisation geeignete, robuste Algorithmen gesichtet und anhand von realistischen Distanzmessungen evaluiert werden. Dabei wird eine von Atmel entwickelte, neuartige Phasendifferenzmethode zur Distanzbestimmung genutzt.

Die nachfolgende Arbeit ist so aufgebaut, dass in Kapitel 2 die für diese Arbeit erforderlichen Grundlagen erläutert und beschrieben werden. Daraufhin werden in Kapitel 3 ähnliche Arbeiten und Forschungen dargestellt. In Kapitel 4 und 5 wird zuerst das Konzept unserer Arbeit dargelegt und anschließend die Umsetzung und Implementierung beschrieben. Zum Schluss werden unsere Ergebnisse in Kapitel 6 evaluiert. Eine Zusammenfassung der Arbeit ist in Kapitel 7 zu finden.

2 Grundlagen

In diesem Kapitel werden einige grundlegende Begriffe und Konzepte vorgestellt welche zum Verständnis dieser Arbeit hilfreich sind. Die Beschreibungen sind sehr einfach gehalten. Für ein tieferes Verständnis empfiehlt es sich deinen Blick in die angegebenen Quellen zu werfen.

2.0.1 ZIGPOS Positioning Kit

Das ZIGPOS Positioning Kit ist eine Plattform für die Evaluation von Lokalisierungsalgorithmen und Tracking-Filtern innerhalb eines realen Sensornetzwerkes.

Das Kit enthält sechs batteriebetriebene, drahtlose Sensorknoten die jeweils aus einem Controller Base Board, einem Radio Extender Board sowie zwei Antennen bestehen. Die Funkmodule sind IEEE 802.15.4 kompatibel und besitzen eine Phasenmessenheit.

Das genutzte Radio Extender Board 233SMAD [4] verbindet sich direkt zu dem Controller Base Board oder kann in Kombination mit einem Atmel-Mikrokontroller genutzt werden. Zusammen mit einem Mikrokontroller formt das REB233SMAD einen vollfunktionsfähigen, drahtlosen Knoten. Das Radio Extender Board (REB) ist mit einem Atmel AT86RF233 [5] Radio und zwei SubMiniature version A (SMA) Konnektoren zur Unterstützung von Antennendiversität ausgestattet. Dadurch ist das Radio in der Lage zwischen den Antennenports den besten Empfangspfad zu wählen. Das Atmel AT86RF233 ist ein 2.4GHz Radio welches unter anderem für Anwendungen im ZigBee/IEEE 802.15.4 Bereich entwickelt wurde [10]. Insbesondere besitzt es eine Einheit zur Messung der Phase eines empfangenen Signals.

Die Netzwerkarchitektur der sechs Knoten ist aus einem Koordinator, Initiatoren und Reflektoren aufgebaut. Der Koordinator agiert als Verbindung zwischen dem Host-PC und den anderen sich im Netzwerk befindlichen Knoten. Er

fordert Distanzmessungen zwischen Initiatoren und Reflektoren an. Ein Knoten kann entweder Initiator oder Reflektor sein, je nachdem von wem die Distanzmessung ausgeht. Der Initiator ist dafür verantwortlich eine Distanzmessung zu starten, Signale vom Reflektor zu empfangen und die Distanz zu berechnen. Es existieren zwei verschiedene Ranging-Konfigurationen. Bei einer lokalen Distanzmessung ist einer der beiden Knoten direkt mit dem Host-PC verbunden. Bei der Remote-Distanzmessung initiiert der Koordinator das Ranging zwischen zwei anderen Knoten. Eine Darstellung der Topologie für Remote-Ranging ist in Abbildung 2.1 zu erkennen.

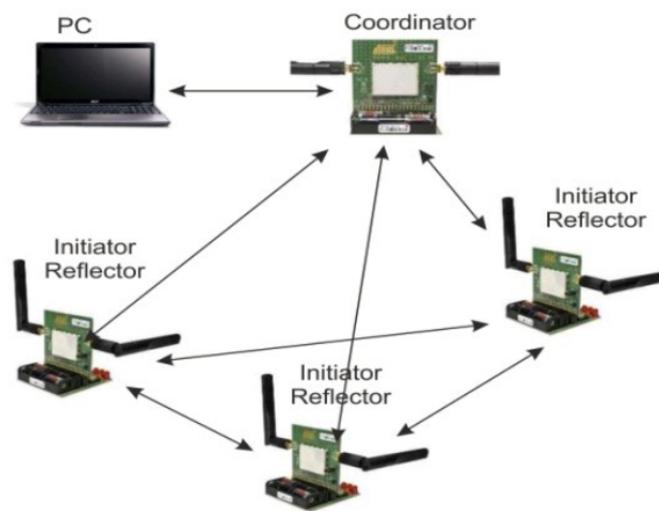


Abbildung 2.1: Remote-Ranging zwischen Knoten [18]

Beim Ranging durchläuft der Initiator eine Menge von Frequenzen. Der Reflektor misst dabei die Phase der eingehenden Frequenzen und überträgt dann selber Signale an den Initiator zur Phasenmessung. Initiator und Reflektor synchronisieren sich mithilfe von IEEE 802.15.4 Rahmen und teilen sich den zu nutzenden Radiokanal zu. Während der Distanzmessungen können keine normalen 802.15.4 Rahmen empfangen werden. Die Phasenmessungen werden in Hardware durchgeführt und die Berechnung der Distanzen in Software [76]. In Abbildung 2.2 ist der Ablauf einer Distanzmessung zwischen Initiator und Reflektor dargestellt.

Um die Qualität der Messungen verfeinern zu können, stellt die auf den Knoten laufende Firmware einige Parameter bereit, mit deren Hilfe man direkt auf

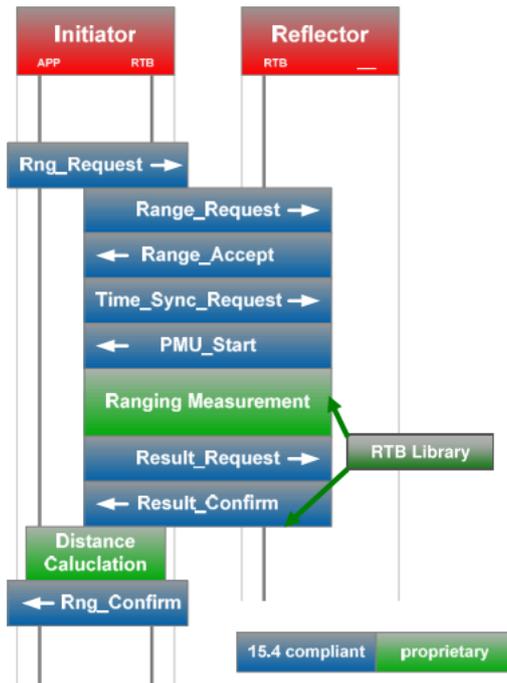


Abbildung 2.2: Ablauf des Ranging zwischen Knoten [6]

die Hardwareparameter der Radiomodule zugreifen kann. Dies hilft bei der Evaluation der Messungen und der Anpassung an die Umgebungsfaktoren [81].

Die Distanzberechnungen stellen zu jedem ermittelten Wert auch eine Qualitätseinschätzung bereit. Dieser Distance Quality Factor (DQF)-Wert reicht von 0 bis 100. Dabei stellt ein Wert von 0 eine fehlerhafte Messung dar, die Verworfen werden kann. Je höher der DQF-Wert, desto genauer sollte die gemessene Distanz sein.

2.0.2 Gaußsches Mischmodell

Allgemein besteht ein Mischmodell aus der Konvexkombination von Wahrscheinlichkeitsdichtefunktionen [49] und ist wie folgt definiert:

$$p(x) = \sum_{k=1}^K \pi_k p(x|k) \quad (2.1)$$

Dabei gibt K die Anzahl an Mischkomponenten an, π_k sind Gewichtungen und $p(x|k)$ gibt die Dichte für die jeweilige Komponente an.

Das Gaußsche Mischmodell (GMM) ist eine Spezialisierung dieser allgemeinen Form. Gaußsche Mischmodelle werden beispielsweise beim Datenclustering oder der Mustererkennung eingesetzt [63, 79] Die Dichtefunktion wird dargestellt als die gewichtete Summe von K gaußschen Komponentendichten:

$$p(x|\theta) = \sum_{k=1}^K \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \quad (2.2)$$

Dabei sind $\theta = \{\pi_k, \mu_k, \Sigma_k\}$ die zur Beschreibung des Modells nötigen Parameter. π_k kennzeichnen die Mischgewichte welche die Bedingungen $0 \leq \pi_k \leq 1$ und $\sum_{k=1}^K \pi_k = 1$ erfüllen. μ_k und Σ_k beschreiben den Mittelwert und die Kovarianz der Komponente k .

Das komplette gaußsche Mischmodell ist definiert durch die Mittelwertsvektoren, Kovarianzmatrizen und Mischgewichte aller Komponentendichten. Die Kovarianzmatrizen Σ_k können vollen Rang haben oder Diagonal sein. Zusätzlich können die Parameter geteilt oder gebunden sein zwischen den gaußschen Komponenten, wie zum Beispiel eine gemeinsame Kovarianzmatrix für alle Komponenten. Die Wahl der Konfiguration (Anzahl der Komponenten, Volle oder Diagonale Kovarianzmatrix, etc.) wird oft bestimmt durch die Anzahl der verfügbaren Daten zur Schätzung der Parameter und wie das GMM in der Anwendung eingesetzt werden soll. Mithilfe einer Lerndatenmenge und einer GMM Parameterkonfiguration kann ein GMM geschätzt werden, welches am Besten mit der Verteilung der Lerndatenmerkmale zusammenpasst. Das Erlernen von gaußschen Mischmodellen findet mithilfe des Expectation-Maximization-Algorithmus (EM-Algorithmus) statt.

2.0.3 Lokalisierungsalgorithmus

Im Bereich der Sensornetzwerke beschreibt Lokalisierung den Vorgang zur Bestimmung des Standortes von Sensorknoten. Algorithmen welche diese Berechnungen durchführen werden Lokalisierungsalgorithmen genannt.

Lokalisierungsalgorithmen werden meist in drei Schritte unterteilt [55, 37, 36, 48]. Im ersten Schritt werden Informationen über Zusammenhänge von Knoten

im Netzwerk gesammelt. Dies können Konnektivitätsinformationen, Distanzmessungen oder Ähnliches sein. Auf diesen Informationen aufbauend wird anschließend eine Position bestimmt. Dies geschieht abhängig von den vorhandenen Informationen und der gewünschten Genauigkeit mithilfe unterschiedlicher Methoden. In einer optionalen dritten Phase kann dann die erste geschätzte Position unter anderem mithilfe von Filtern oder Optimierungsverfahren noch weiter verfeinert werden.

Viele Lokalisierungsalgorithmen schätzen den Standort eines Knotens mithilfe der Positionen einiger umliegender Knoten und den zu ihnen gemessenen Distanzen. Knoten deren Standort schon a priori bekannt ist und nicht berechnet werden muss, werden Ankerknoten genannt. Ihre Position wird oftmals durch Methoden wie GPS oder durch manuelle Konfiguration erlangt. Mithilfe von Ankerknoten können absolute Positionen berechnet werden, ansonsten befindet sich die geschätzte Position relativ im Netzwerk [1]. In Abbildung 2.3 ist solch ein Ablauf eines Lokalisierungsalgorithmus dargestellt. Es ist zu erkennen, dass unterschiedliche Phasen existieren. Mithilfe der Messtechniken werden Distanzen aus den Signalmodalitäten bestimmt. Der Lokalisierungsalgorithmus beschreibt die Bestimmung der Position anhand der Distanzen und bekannten Knotenpositionen.

Für diese Arbeit definieren wir einige wichtige, wiederkehrende Begriffe:

- **Ranging** ist der Prozess der Distanzschätzung zwischen zwei Knoten [55].
- **Positioning** ist der Prozess der Positionsschätzung innerhalb eines Referenzsystems [55].
- **Refinement** ist ein optionaler Prozess zur Verbesserung einer initial geschätzten Position oder Distanzmessung [48].
- **Ankerknoten** sind Knoten deren Position a priori bekannt ist [8].

Ranging

Distanzen sind immer notwendig um eine Position zu bestimmen. Manchmal werden diese Distanzen geschätzt, in anderen Situationen werden Distanzen gemessen und berechnet. Während des Ranging kann mithilfe verschiedener Signale (Radio, Infrarot, Ultraschall, etc.) die Entfernung zu anderen Knoten

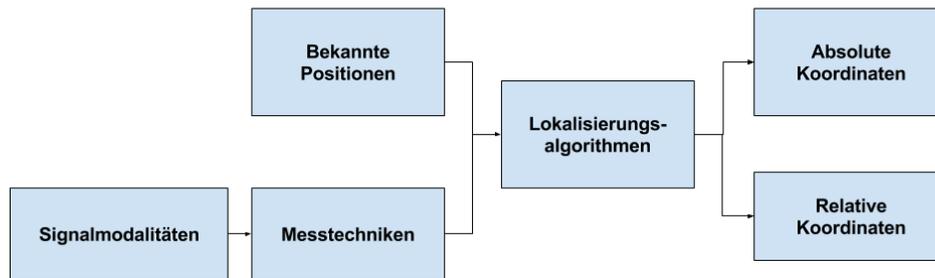


Abbildung 2.3: Lokalisierungsablauf [1]

bestimmt werden [36]. Es existieren unterschiedliche Methoden um Entfernungen zu bestimmen.

Ein einfacher Ansatz ist anzunehmen, dass wenn zwei Knoten miteinander kommunizieren können, sie sich in Radioreichweite befinden. Sie können also nicht weiter als die maximal mögliche Radioreichweite voneinander entfernt sein. Andere Methoden messen eine Eigenschaft des Radiosignals, wie beispielsweise die empfangene Signalstärke, um daraus eine Entfernung abzuleiten. Weitere Ansätze benötigen spezielle Hardware um Messungen durchzuführen und Distanzen zu bestimmen. Alle Methoden welche genutzt werden um Distanzen zwischen Knoten zu bestimmen werden als Rangingmethoden bezeichnet, [48, 8, 55]. Die existierenden Methoden können dabei je nach Art des Vorgehens in verschiedene Klassen eingeteilt werden [46].

Im Folgenden sind einige bekannte Methoden aufgelistet und kurz erläutert. Gründliche Erklärungen dieser Methoden finden sich in [55, 8] wieder.

- **Received Signalstrength Indicator (RSSI):** Auf der Basis der empfangenen Signalstärke kann mithilfe eines Pfadverlustmodells und der

Sendestärke die Distanz zwischen Sender und Empfänger geschätzt werden. Die genutzten Pfadverlustmodelle können empirisch oder theoretisch erstellt werden. Vorteilhaft an dieser Methode ist, dass Signalstärkemessungen einfach in Hardware umzusetzen sind und die Messungen während der normalen Kommunikation durchgeführt werden können ohne eine zusätzliche Bandbreite oder zusätzlichen Energieverbrauch zu benötigen. Die Messungen sind allerdings oftmals durch eine falsche Kalibrierung der Radiokomponenten oder fehlerhafte Pfadverlustmodelle sehr ungenau [58].

- **Hop-Count/Proximity:** Die Basis dieser Methode bildet die Beobachtung, dass wenn zwei Knoten in der Lage sind miteinander zu kommunizieren, die Distanz zwischen ihnen maximal die Radioreichweite sein kann. Dadurch können, bei bekannter Radioreichweite, einfache Konnektivitätsinformationen zur Lokalisierung genutzt werden. Mithilfe dieses Ansatzes können auch Distanzen zwischen entfernten Knoten innerhalb des Netzes berechnet werden. Dabei wird das Netzwerk als ein ungewichteter Graph betrachtet, indem die Sensorknoten die Knoten innerhalb des Graph darstellen und die Radioverbindungen die Kanten. Der sogenannte Hop-Count zwischen zwei Knoten ist dann die Länge des kürzesten Pfades zwischen Ihnen. Die Distanz beträgt dann den Wert des Produktes zwischen Hop-Count und maximaler Radioreichweite.
- **Time of Arrival (ToA):** Mit dieser Technik wird die Distanz zwischen zwei Knoten berechnet, indem ein Signal von einem Knoten zu einer bestimmten Zeit ausgesendet wird und diese Sendezeit mit der Ankunftszeit am Zielknoten verglichen wird. Aus der Übertragungszeit und dem Wissen über die Ausbreitungsgeschwindigkeit des Signals lässt sich eine Distanz ableiten. Dabei ist es dringend notwendig, dass die Uhren beider Knoten synchronisiert sind. Die Zeitsynchronisation ist sehr komplex und schwierig zu gestalten. Die Messungen können durch additives weißes Rauschen sowie Mehrwegempfang negativ beeinflusst werden [58].
- **Time Difference of Arrival (TDoA):** TDoA ist eine Erweiterung der ToA-Technik und misst den Unterschied zwischen den Empfangszeiten zweier eingehender Signale um eine Distanz zu bestimmen. Die Signale werden von verschiedenen Nachbarn gesendet. Es ist dabei notwendig, dass die Uhren der sendenden Nachbarn synchronisiert sind. Auch ist es möglich, dass ein einzelner Nachbar zwei Signale unterschiedlichen

Mediums (Radio und Ultraschall) sendet, damit sie verschiedene Ausbreitungszeiten haben. Diese Methodik erfordert spezielle Hardware und ist auch aufgrund der Zeitsynchronisation schwierig umzusetzen.

Positioning

Das Ziel der Lokalisation ist die Bestimmung des Standortes der Sensorknoten. Während des Positioning wird basierend auf den innerhalb des Ranging gesammelten Daten die Position des Knotens berechnet.

Es gibt viele unterschiedliche Wege dies zu implementieren. Manche Methoden bestimmen direkt aus den Informationen eine Position während andere Ansätze sich iterativ an eine optimale Lösung annähern. Mithilfe der Multilateration kann durch Distanzmessungen und zugehöriger Positionen direkt ein möglicher Standort berechnet werden. Dabei können Messfehler allerdings großen Einfluss auf die geschätzte Position haben. Andere Methoden, wie beispielsweise der IQL-Algorithmus, verfeinern ausgehend von einem groben Schätzwert die Position immer weiter bis eine präzisere Schätzung gefunden wurde [36].

Existierende Positionierungsalgorithmen können in unterschiedliche Klassen eingeteilt werden. Die bekanntesten Klassen prüfen dabei ob die Algorithmen absolute Distanzen zwischen Knoten benötigen, Ankerknoten im Netzwerk voraussetzen oder ihre Berechnungen verteilt oder zentral im Netzwerk erledigen [52]. Viele weitere Klassifikationen sind möglich [1, 8, 50].

Im folgenden werden zwei Positionierungsstrategien vorgestellt.

MDS mit Trilateration In [57] wird ein aus zwei Phasen bestehender, hybrider Ansatz zur Lokalisation vorgestellt. Während der ersten Phase wird mithilfe von Multi-dimensional Scaling (MDS) eine Schätzung der Position berechnet. In der zweiten Phase wird diese Schätzung mithilfe der Trilateration durch Anpassung iterativ optimiert.

Im ersten Schritt des Algorithmus werden Distanzen zu den benachbarten Knoten bestimmt. Anschließend wird mithilfe dieser Distanzen und MDS eine erste Schätzung der Sensorknotenposition berechnet. MDS ist eine Datenanalysetechnik welche Distanzdaten als geometrische Struktur darstellt. Die Ausgabe ist eine relative Karte, welche mithilfe der Positionen von Ankerknoten in eine absolute Karte gewandelt werden kann. Die Berechnung der Position mithilfe

von MDS ist energie-effizient, da Kommunikation zwischen Knoten nur zu Beginn notwendig ist, um die Distanzen zu den Nachbarknoten zu ermitteln. Es existieren unterschiedliche Typen von MDS-Techniken. In diesem Algorithmus wird ein klassischer MDS-Algorithmus genutzt. Die Daten sind quantitativ und Objektähnlichkeiten werden als Distanzen im euklidischen Raum beschrieben.

Nach der Berechnung einer ersten Position wird diese durch die Optimierungsmethode Trilateration durch Anpassung verfeinert. Diese Technik verbessert die Positionsschätzung durch schrittweise Anpassung der Distanzen und Positionen. Es wird dabei jeder Distanzmessung ein Gewicht anhand ihrer Qualität zugeteilt und während jedes Iterationsschrittes die geschätzte Position sowie die Distanzen angepasst. Die Iterationen werden solange ausgeführt bis eine maximale Anzahl an Iterationen erreicht oder die Größe der Änderungen einen Schwellwert unterschreiten. Die Änderung der Position innerhalb eines Iterationsschrittes wird beschrieben durch:

$$\Delta = (B^T W B)^{-1} (B^T W f) \quad (2.3)$$

Die Matrix W kodiert dabei die Gewichte der Distanzen. f ist ein Vektor von Fehlern, der Unterschiede zwischen Distanzmessungen und euklidischem Abstand zwischen dem bekannten und unbekanntem Knoten. B ist eine Jakobimatrix der Fehlerfunktionen.

Der beschriebene Algorithmus wird in dieser Arbeit Multi-Dimensional-Scaling mit Trilateration durch Anpassung (MDS+) genannt.

ILS mit Fehlermanagement In [41] wird eine iterative Methode zur Knotenlokalisierung vorgestellt. Dabei wird von einer geringen Ankeranzahl in einem großen Sensornetzwerk ausgegangen. Während jeder Iteration werden Knoten mit einem robusten Iterative-Least-Squares Algorithmus (ILS) lokalisiert. Bereits lokalisierte Knoten werden anschließend als Pseudo-Ankerknoten für noch nicht lokalisierte Knoten genutzt.

Während jedes Iterationsschrittes wird eine Positionsschätzung mithilfe einer robusten Formulierung des ILS Algorithmus durchgeführt. Verglichen zu der traditionellen Formulierung ist diese Methode stabiler gegenüber Messfehlern in der Distanzbestimmung. Zu Beginn der Positionsschätzung sammelt ein freier

Knoten Distanzmessungen zu allen umgebenden Knoten mit bekannten Positionen. Mithilfe der Positionen und den dazugehörigen Distanzmessungen wird ein Gleichungssystem aufgestellt und eine Positionsschätzung gesucht welche den quadratischen Fehler minimiert. Die traditionelle Formulierung lautet:

$$\hat{x}_t = \arg \min_x \|Ax - b\|^2 \quad (2.4)$$

Dabei beschreibt die Matrix A die Positionen der umliegenden Nachbarknoten und Vektor b die zu ihnen gemessenen Distanzen. Diese Formulierung geht von fehlerfreien Konditionen aus. In der Praxis existieren Fehler in den Positionen sowie in den gemessenen Distanzen. Die in A und b entstehenden Abweichungen werden durch ΔA und Δb beschrieben. Die robuste Formulierung sieht dementsprechend wie folgt aus:

$$\hat{x}_t = \arg \min_x E\|(A + \Delta A)x - (b + \Delta b)\|^2 \quad (2.5)$$

In iterativen Lokalisierungsmethoden ist Fehlerpropagation ein Problem. Dabei beeinflussen Knoten mit einer ungenauen Positionsbestimmung negativ die Lokalisierung umliegender Knoten. Um eine Akkumulation und Propagation von Lokalisationsfehlern zu verhindern, nutzt dieser Algorithmus einen Fehlerkontrollmechanismus. Dieser besteht grundlegend daraus, dass zu jeder Positionsschätzung auch eine dazugehörige Qualitätsschätzung berechnet wird. Die zur Lokalisation genutzten Knoten werden anhand dieser Qualitätsschätzung selektiert.

Dieser Algorithmus ohne Fehlerkontrolle wird im weiteren RLS und mit genutzter Fehlerkontrolle RLS+ genannt.

3 Verwandte Arbeiten

Die Wahl eines passenden Lokalisierungsalgorithmus ist ein wichtiges Kriterium für den erfolgreichen Einsatz von drahtlosen Sensornetzwerken. Die Vielfältigkeit an Sensornetzwerken, bedingt durch das Ziel des Netzwerkes, den Einsatzort und die Art der Sensorknoten erfordert den Einsatz eines für das jeweilige Sensornetzwerk angepassten Lokalisierungsalgorithmus. Die große Anzahl an existierenden Algorithmen, mit jeweils unterschiedlichen Voraussetzungen und Verhaltensweisen, erschwert die Wahl eines Algorithmus [52].

In [70] wird ein auf Konnektivitätsinformationen basierender, zentralisierter Algorithmus namens MDS-MAP vorgestellt. Der Algorithmus kann zusätzlich Distanzinformationen und bekannte Positionen mit in die Positionsbestimmung einbeziehen. In der Arbeit wird mithilfe einer Simulation gezeigt, dass der Algorithmus robuster gegenüber Messfehlern ist als ähnliche Methoden, besonders wenn die Knoten gleichmäßig verteilt sind. Die Vorteile dieses Algorithmus liegen darin, dass er auch bei keinen oder nur wenigen Ankerknoten angewendet werden kann. Erweiterungen dieses Algorithmus, dwMDS und MDS-MAP(P), werden in [19, 69] dargestellt. dwMDS gewichtet Distanzmessungen anhand ihrer Qualität und MDS-MAP(P) stellt eine verteilte Implementierung des Original-Algorithmus dar. In [16] wird ein aus MDS und Proximity Distance Mapping (PDM) bestehender, verteilter Lokalisierungsalgorithmus beschrieben. Der Algorithmus liefert bei nur wenigen Ankerknoten akkurate Positionsbestimmungen und arbeitet Phasenweise. In der ersten Phase wird eine Menge an Sensorknoten mithilfe des MDS-Algorithmus lokalisiert und in Ankerknoten gewandelt. Die restlichen Knoten im Netz lokalisieren sich anschließend mithilfe des PDM-Algorithmus, wodurch sich die benötigte Rechenleistung und Komplexität der Lokalisierung verringert. Im Vergleich mit MDS-MAP erzielt der Algorithmus ähnlich gute Ergebnisse. Auch wurde der Effekt von verschiedenen Ankerplatzierungsstrategien untersucht und der vorgeschlagene Algorithmus ist weniger anfällig für schlechte Platzierun-

gen. Vorteilhaft ist die geringe Komplexität des Algorithmus sowie die verteilte Implementation.

Algorithmen wie MDS berechnen die Position in einem Schritt. Im Gegensatz dazu existieren Algorithmen welche durch schrittweise Optimierung der Lösung eines Gleichungssystems die bestmögliche Position bestimmen. Solche Verfahren basieren oft auf Multilateration. Multilateration ist ein einfacher, verteilter, geometrie-basierter Lokalisierungsalgorithmus basierend auf dem Prinzip, dass die Position auf einer zweidimensionalen Ebene durch die Distanz zu mindestens drei nicht-kollinearen Referenzpunkten beschrieben werden kann.

In [68] wird AHLoS, ein verteilter Algorithmus basierend auf einem iterativen Multilaterationsansatz vorgestellt. Knoten innerhalb des Netzwerkes können ihre Position abhängig von der Zahl ihrer Ankernachbarn mit unterschiedlichen Multilaterationsmethoden (Atomic Multilateration, Collaborative Multilateration) bestimmen. Sobald ein Knoten seine Position bestimmt hat, kann er selber zum Ankerknoten werden und so weiteren Knoten die Lokalisation ermöglichen. In diesem Ansatz wird eine Maximum Likelihood Schätzung der Knotenposition mithilfe des Minimum mean square error (MMSE) eines Gleichungssystem berechnet.

Bachrach et al. stellen einen robusten und verteilten Lokalisierungsalgorithmus vor [7]. Die Basis bildet Multilateration indem mithilfe der Umgebungsknotenpositionen und der zu ihnen geschätzten Distanzen eine Position bestimmt wird. Die Methode nutzt iterativ den Gradientenabstieg um Koordinaten zu finden welche den quadratischen Fehler des durch Positionen und Distanzen aufgestellten Gleichungssystems minimieren. Den Startpunkt der Optimierung gibt dabei entweder die letzte Positionsschätzung oder die Position des Knotens mit der geringsten geschätzten Entfernung an. Der Algorithmus wurde mithilfe einer Simulation und in einer realen Umgebung getestet und liefert gute Ergebnisse.

Um das Verhalten von Lokalisierungsalgorithmen besser einzuschätzen sowie die Qualität der Positionsbestimmungen untereinander zu vergleichen, existieren Studien in denen mehrere Algorithmen innerhalb verschiedener Test-szenarien evaluiert werden [15, 21, 32, 37, 39, 44, 48, 89]. In [37] wird gezeigt, dass Lokalisierungsalgorithmen abhängig von der Qualität der Distanzmessungen sind. Dabei werden Lokalisierungsalgorithmen in drei Phasen unterteilt (Distanzmessung, Positionierung, Verfeinerung) und die Art der Distanzmessungen geändert. Die Autoren geben an, dass es keinen besten Algorithmus

gibt und noch ein großer Bedarf an Verbesserungen vorhanden ist. In [21] werden vier Lokalisierungsalgorithmen (MLE, MMDS, MSM, WMDS) welche Distanzen zu Nachbarknoten aus der empfangenen Signalstärke ableiten innerhalb einer Simulation evaluiert. Genutzt wird dabei ein einfaches Ausbreitungsmodell für das Radiosignal. Die dabei empfangene Signalstärke ist gauß-verteilt. Die Algorithmen wurden in verschiedenen Szenarien getestet in denen die Ankeranzahl sowie die Größe des Distanzmessfehlers variiert wurden. Um den Positionierungsfehler zu minimieren empfehlen die Autoren WMDS, zur Minimierung der Komplexität sollte MLE genutzt werden. Mit sinkendem Messfehler und genügend Ankerknoten verbessert sich die Qualität des MDS-Algorithmus. Die Autoren empfehlen den Einsatz mehrerer Ankerknoten bei steigendem Rauschen des Radiosignals.

In vielen dieser Arbeiten werden sogenannte RSSI-basierte Algorithmen evaluiert [15, 21]. Diese Algorithmen nutzen die empfangene Signalstärke des Radiosignals um die Entfernung zum Sender zu schätzen. Da diese Methode sehr einfach ist und keine spezielle Hardware erfordert, erfreut sie sich großer Beliebtheit in Sensornetzwerken. Solche Distanzbestimmungen sind allerdings oft ungenau, weshalb auch sogenannte distanz-freie Algorithmen, welche nicht auf absoluten Distanzmessungen beruhen, an Interesse gewinnen [39].

Das Problem dieser Studien, welche die Algorithmen meistens innerhalb eines Simulationsmodells evaluieren, besteht darin, dass für die Modellierung der Signalstärke ein sehr einfaches Modell genutzt wird [15, 21]. Die Irregularitäten realer RSSI-Messungen werden nicht korrekt abgebildet und die Evaluation der Algorithmen fällt damit sehr optimistisch aus. Eine gute Modellierung der bestimmten Distanzen ist sehr komplex und ein inherentes Problem bei der Evaluation von Lokalisierungsalgorithmen mithilfe einer Simulation. Ein interessanter Ansatz dieses Problem zu beheben lässt sich in [89] wiederfinden. Dort wird eine Menge von realen RSSI-Messungen gesammelt um mithilfe dieser Messungen die Parameter für das Simulationsmodell zu schätzen. Die Autoren kommen zum Schluss, dass RSSI-basierte Lokalisation innerhalb von Indoor-Umgebungen nicht geeignet ist um akkurate und präzise Positionsbestimmungen zu gewährleisten.

Eine neue und bisher noch nicht sehr viel erforschte Methode ist die Distanzbestimmung mithilfe der Phase des Radiosignals [60]. In dieser Arbeit wird eine neue, von Atmel entwickelte Technologie zur Bestimmung der Distanz mit-

hilfe der Phase genutzt um robuste Lokalisierungsalgorithmen innerhalb von Indoor-Szenarien zu vergleichen [18].

4 Konzept

Im Folgenden wird das Konzept zur Evaluation der Lokalisierungsalgorithmen vorgestellt. Ziel ist es robuste Lokalisierungsalgorithmen zu finden und diese mithilfe realistischer Distanzmessungen zu evaluieren.

4.1 Algorithmen sichten und sammeln

Eine Literaturstudie der wissenschaftlichen Veröffentlichungen zum Thema Lokalisation in Sensornetzwerken ermöglicht einen Überblick der aktuell existierenden Lokalisierungsalgorithmen. Ziel der Literaturstudie ist eine Sammlung der veröffentlichten Studien um später Algorithmen zu klassifizieren und auszuwählen. Die gesichteten Arbeiten wurden dabei grob in die folgenden Kategorien eingeteilt.

- Vergleichsstudien
- Lokalisierungsalgorithmen
- Distanzbestimmung
- Allgemein
- Sensornetzwerke

In die Kategorie der Vergleichsstudien gehören Arbeiten welche verschiedene Lokalisierungsalgorithmen evaluieren und die Ergebnisse vergleichen. In der Kategorie der Lokalisierungsalgorithmen werden direkt Algorithmen zur Positionsbestimmung beschrieben. Veröffentlichungen welche Ansätze zur Berechnung oder Schätzung der Distanz zwischen Knoten beschreiben werden in der Kategorie Distanzbestimmung aufgeführt. Die Kategorie Allgemein beschreibt Forschungsarbeiten die sich mit Klassifikation oder der allgemeinen Arbeitsweise von Lokalisierungsalgorithmen beschäftigen. Innerhalb der

Sensornetzwerke-Kategorie werden Arbeiten aufgeführt welche sich mit dem Thema der Sensornetzwerke ansich beschäftigen.

Die von uns gesichteten Publikationen und deren Einteilung sind in Tabelle 4.1 zusammengefasst. Viele Arbeiten könnten in mehrere Kategorien eingeteilt werden. In der Tabelle gehört jede Arbeit genau einer Kategorie an, um eine Übersichtlichkeit zu gewährleisten. Die Kategorisierung ermöglicht eine Übersicht der aktuellen Veröffentlichungen und erleichtert die Erstellung einer Klassifikation sowie die Auswahl und Klassifizierung existierender Algorithmen.

Vergleichsstudien	Lokalisierungsalgorithmen	Distanzbestimmung	Allgemein	Sensornetzwerke
[15]	[3],[7],[84]	[88]	[1]	[2]
[21]	[9],[11],[82]	[18]	[8]	[17]
[23]	[12],[14],[75]	[87]	[10]	[25]
[32]	[16],[19],[71]	[85]	[13]	[67]
[37]	[20],[22],[69]	[60]	[56]	[35]
[39]	[27],[83],[66]		[78]	
[44]	[29],[30],[64]		[36]	
[44]	[31],[33],[80]		[46]	
[48]	[34],[59],[73]		[58]	
[89]	[86],[40],[70]		[50]	
	[41],[42],[68]		[52]	
	[43],[47],[65]		[55]	
	[51],[53],[62]			
	[54],[57],[61]			

Tabelle 4.1: Klassifizierung aktueller Veröffentlichungen im Bereich der Sensornetzwerke

4.2 Algorithmen beschreiben und klassifizieren

Mithilfe der in der Literaturstudie gefundenen Arbeiten werden nun Algorithmen anhand einer erstellten Klassifikation beschrieben. Dies ist für eine begründete Auswahl für Algorithmen zur Evaluation hilfreich. Es wird eine Tabelle mit den gefundenen Algorithmen und ihrer Eigenschaften aufgestellt. Als Grundlage für diese Tabelle nutzen wir die in [52] vorgestellte Klassifikation und erweitern diese um einige Eigenschaften. Unsere Klassifikation enthält die folgenden Aspekte:

- **Distanz-basiert/Distanz-frei** Distanz-basierte Algorithmen nutzen zur Berechnung der Positionen absolute Distanzen zu Nachbarknoten.

Distanz-freie Algorithmen benötigen diese Werte nicht und arbeiten entweder mithilfe von Konnektivitätsinformationen oder mit dem relativen Vergleich von Funksignalstärken.

- **Anker-basiert/Anker-frei** Anker-basierte Algorithmen gehen davon aus, dass sich Knoten im Netzwerk befinden welche ihre Position schon a priori kennen. Anker-freie Algorithmen benötigen diese speziellen Sensorknoten nicht.
- **Verteilt/Zentralisiert** Bei den verteilten Algorithmen finden alle Berechnungen zur Positionsbestimmung direkt auf dem zu lokalisierenden Knoten statt. Zentralisierte Algorithmen hingegen leiten Daten an eine zentrale Stelle im Netzwerk weiter, an dem die Berechnung mithilfe globaler Informationen stattfindet.
- **Approximativ/Exakt** Exakte Lokalisation basiert auf präzisen Messungen zwischen Sensoren und berechnen eine Position mit hoher Genauigkeit. Nachteilig sind oftmals die komplexe Berechnung und ein hoher Netzwerkverkehr [12]. Da nicht immer eine exakte Position benötigt wird, reicht es in manchen Szenarien aus, eine approximative Positionsbestimmung durchzuführen. Approximative Lokalisierungsalgorithmen bestimmen eine Fläche, in welcher sich der Knoten aufhalten könnte.
- **Induktiv/Deduktiv** Deduktive Methoden leiten aus physischen Eigenschaften eines Signals direkt Informationen zur Distanzbestimmung ab. Induktive Methoden benötigen eine Trainingsphase, in der sie vorher die Signalstärke an verschiedenen Positionen erlernen. Während der Lokalisation wird mithilfe einer Tabelle die Position aus der empfangenen Signalstärke abgeleitet [42].
- **Distanzmethode** Die Distanzmethode beschreibt die genutzte Technik zur Berechnung von Entfernungen zwischen Sensorknoten.
- **Positionierungsmethode** Die Positionierungsmethode beschreibt die genutzte Technik zur Berechnung einer Position aus den gesammelten Grunddaten.
- **2D/3D** Diese Eigenschaft beschreibt ob ein Algorithmus für zweidimensionale oder dreidimensionale Szenarien anwendbar ist.
- **Mobil/Statisch** Ein mobiler Algorithmus kann genaue Positionen auch für sich im Sensornetzwerk bewegende Sensorknoten berechnen. Andernfalls ist der Algorithmus statisch.

- **One-Hop/Multi-Hop** One-Hop Lokalisierungsalgorithmen sind Methoden in denen der zu lokalisierende Sensorknoten der direkte Nachbar einer genügenden Anzahl von Ankerknoten ist [46]. Wenn Knoten, welche nicht in direkter Reichweite genügender Ankerknoten sind, sich trotzdem mithilfe entfernter Ankerknoten lokalisieren können, so werden diese Algorithmen Multi-Hop Algorithmen genannt [15].
- **Iterativ/Gleichzeitig** Bei der iterativen Lokalisation werden Sensorknoten, welche bereits ihre Position bestimmt haben, als Ankerknoten für unlokalisierte Knoten genutzt. Dadurch werden Lokalisationsinformationen iterativ durch das Netzwerk getragen. Bei gleichzeitigen Algorithmen geschieht dies nicht, und Sensorknoten brauchen nur einmal versuchen sich zu lokalisieren. Wenn sie dabei nicht genügend Ankerknoten als Nachbarn haben, bleiben sie unlokalisiert.

Mithilfe dieser Klassifikation werden die in der Literaturstudie gesichteten Lokalisierungsalgorithmen beschrieben. Nicht alle Arbeiten geben dabei über die gesuchten Eigenschaften Auskunft oder deuten gewisse Funktionalitäten nur an. Es wurde versucht sich so nah an den Algorithmenbeschreibungen wie möglich zu halten. Wenn in einer Arbeit beispielsweise nicht explizit auf eine mögliche Nutzung im 3D-Raum hingewiesen wurde und der Algorithmus nur in 2D beschrieben und getestet wurde, so klassifizieren wir ihn unter 2D. Falls eine Eigenschaft überhaupt nicht beschrieben worden ist, so wird sie nicht mit aufgeführt.

Die Klassifikation der gefundenen Lokalisierungsalgorithmen ist in Tabelle 4.2 wiedergegeben. Aus Formatierungsgründen wurden die Eigenschaften abgekürzt.

Zur begründeten Auswahl von Algorithmen stellen wir einen Anforderungskatalog auf. Dieser Katalog enthält für eine Indoor-Lokalisierung günstige Eigenschaften.

Ein wichtiges Kriterium für jeden Lokalisierungsalgorithmus ist eine präzise Positionsbestimmung. Solch eine präzise Lokalisation lässt sich nur mit Algorithmen erreichen, welche distanz-basiert sind. Auch aus dem Aspekt der präzisen Positionsbestimmung heraus, fordern wir dass der Algorithmus Ankerbasiert ist. Durch den Einsatz von Ankerknoten kann die Positionsbestimmung von Sensorknoten verbessert werden, desweiteren ist es möglich absolute Positionen im Raum zu bestimmen. Die dritte Eigenschaft ist die Berechnung der

Position auf den Knoten selber. Dies hat den Vorteil, dass wir einen geringeren Kommunikationsaufwand haben, sowie eine schnellere Berechnung der Position stattfindet. Dies ist insbesondere für mobile Szenarien notwendig. Da sonst eine berechnete Position schon veraltet und falsch sein könnte. Die Anforderungen die wir an zu evaluierende Algorithmen stellen, lauten wie folgt:

- Distanz-basiert
- Anker-basiert
- Verteilt

Aus der Menge der Algorithmen, die diese Anforderungen erfüllen, werden zwei Ansätze, MDS+ und Robust Least-Squares mit Optimierung (RLS+), ausgewählt [57, 41]. Diese Algorithmen versprechen eine robuste Positionierung auch bei ungenauen Distanzmessungen sowie eine einfache Implementierung.

4.3 Entwickeln der Evaluationssoftware

Zur Evaluation der Lokalisierungsalgorithmen existieren verschiedene Ansätze. Zum einen können die Algorithmen auf echten Sensorknoten innerhalb einer realen Umgebung ausgeführt werden. Dafür wird eine bekannte Testumgebung geschaffen in welcher wir ein Sensornetzwerk aufbauen. Diese Methode hat den Vorteil, dass die entstehenden Ergebnisse sehr aussagekräftig sind und man einen realistischen Eindruck vom Verhalten der Algorithmen bekommt. Die Umsetzung dieses Ansatzes ist allerdings mit einem sehr hohen Aufwand verbunden. Nachteilig sind zum Einen die hohen Anschaffungskosten für die Sensorknoten und eine damit einhergehende geringe Skalierbarkeit des Sensornetzes. Auch können während der Evaluation Fehler auftreten, die unabhängig von unseren Experimenten sind und die Qualitätsbestimmung beeinflussen.

Aus diesen Gründen evaluieren wir die Algorithmen innerhalb einer Simulationssoftware. Vorteil dieser Methode ist die komplette Kontrolle der Evaluationsparameter. Wir können Größe und Aufbau des Netzes beliebig skalieren und auch weitere Parameter wie die Größe des Distanzmessfehlers direkt beeinflussen. Auch entstehen keine Kosten durch Extra-Hardware und wir sind flexibler in der Erstellung der Experimente. Nachteilig an dieser Methode sind durch Vereinfachung und Abstrahierung von Konzepten entstehende Abweichungen von einem realen Verhalten. Aus diesem Grund sollte darauf geachtet

werden, dass insbesondere die für uns wichtigen Konzepte, wie beispielsweise die Sensordistanzmesswerte, sehr realitätsgetreu modelliert werden. Die Simulationssoftware sollte modular aufgebaut sein, damit wir die Art der Positionsbestimmung sowie die Art der Distanzbestimmung problemlos austauschen und kombinieren können. Somit ist die Simulationssoftware nicht auf bestimmte Algorithmen und Distanzbestimmungsmethoden zugeschnitten, sondern kann leicht erweitert und verschiedene Kombinationen getestet werden.

4.4 Erstellen des Phasendifferenzdistanzgenerators

Zur Generierung realistischer Distanzmesswerte für die simulierten Sensorknoten existieren zwei grundsätzliche Vorgehensweisen. Eine Möglichkeit besteht darin, das Funksignal in der Simulation so zu modellieren, dass wir die Phase herauslesen können und mithilfe mathematischer Modelle und Formeln eine Distanz daraus ableiten. Dies setzt eine vernünftige Modellierung des Funksignals voraus, welche aufgrund der vielen Faktoren die Einfluss auf die Qualität des Signals haben, sehr komplex und schwer umzusetzen ist. Insbesondere in Indoor-Szenarien ist das Radiosignal durch Störungen wie Reflexionen und Interferenzen so gut wie unvorhersehbar. Aus einem solchen Signal abgeleitete Distanzmessungen können stark schwanken und einer nicht-linearen Verteilung folgen.

Wir entscheiden uns für einen statistischen Ansatz zur Erstellung eines virtuellen Phasendifferenz-Distanzgenerators. Dafür wird mithilfe von realen Sensorknoten eine große Menge an Distanzdaten gesammelt welche mithilfe der Phasendifferenz des Radiosignals berechnet wurden. Diese Daten werden für verschieden große Abstände zwischen den Knoten gesammelt. Mithilfe der gesammelten Distanzdaten wird eine Wahrscheinlichkeitsverteilung bestimmt, welche die Verteilung der Daten am Besten widerspiegelt. Diese Wahrscheinlichkeitsverteilung stellt den Kern des virtuellen Distanzgenerators dar. Um in der Simulation einen Distanzwert zu generieren, wird eine bedingte Zufallsstichprobe für die ausgewählte Entfernung gezogen.

4.5 Durchführung der Experimente

Zur Evaluation der Algorithmen wird die Qualität der Positionsbestimmungen innerhalb verschiedener Szenarien verglichen. Dabei können verschiedene Metriken zur Einschätzung der Qualität genutzt werden. Interessante Eigenschaften sind der maximale sowie der minimale Lokalisierungsfehler, die Anzahl der lokalisierten Knoten, der mittlere Positionierungsfehler sowie die dazugehörige Varianz und Standardabweichung. Als Metrik nutzen wir den mittleren Positionierungsfehler der lokalisierten Knoten, da insbesondere die Lokalisationsgenauigkeit in Indoor-Szenarien eine sehr wichtige Rolle spielt. Diese Metrik wird in unterschiedlichen Experimenten berechnet. Ein Experiment besteht aus mehreren Szenarien und beschreibt dabei vollständig alle wichtigen Parameter. Dabei wird zwischen festen und variablen Parametern unterschieden. Die festen Parameter beschreiben ein Grundscenario in welchem die Lokalisierungsalgorithmen bei unterschiedlichen Größen der variablen Parameter analysiert werden. Auf diese Weise wird der Einfluss der variablen Parameter auf das Lokalisierungsverfahren bestimmt. Parameter können beispielweise die Größe des Spielfeldes, Anzahl und Platzierung der Sensorknoten oder die Größe der Distanzmessungen sein. Wir entscheiden uns den Einfluss der Anzahl der Ankerknoten im Spielfeld sowie den Einfluss der Genauigkeit der Distanzmessungen auf die Algorithmen zu evaluieren.

Quelle	DB/DF	AB/AF	V/Z	A/E	I/D	Distanzmethode	Positionierung	Dimension	M/S	E/M	S/G
[57]	DB	AB	V	E	D	RSSI	MDS	2D	S	E	G
[41]	DB	AB	V	E	D	/	ILS	2D	S	M	S
[62]	DB	AB	V	E	/	/	Hop-Terrain+	2D	S	M	G
[68]	DB	AB	V	E	/	/	AHLoS	2D	S	M	S
[66]	DB	AB	V/Z	E	/	/	Coll. Multilateration	2D	S	M	S
[19]	DB	AB	V	E	D	RSSI/TOA	dwMDS	2D	S	E	G
[31]	DF	AF	V	A	/	RSSI	Region Partition	2D	M	E	G
[84]	DB	AB	V	E	I	RSSI	Multilateration	2D	S	E	G
[86]	DB	AB	V	A	/	/	MD-Min-Max	2D	S	E	G
[11]	DB	AB	V	A	D	RSSI	WCL	2D	S	E	G
[12]	DF	AB	V	A	/	Hop-Count	WCL	2D	S	M	G
[53]	DB	AB	V	E	D	AoA	APS	2D	M	M	S
[47]	DB	AF	Z	E	D	Radio Interferometrie	RIPS	3D	S	E	G
[30]	DB	AB	V	E	D	Radio Interferometrie	ILuIR	3D	S	E	S
[9]	DF	AB	Z	E	I	RSSI	Triangulation	2D	M	E	G
[43]	DB	AB	V	E	I	RSSI	DTN	2D	M	E	G
[51]	DB	AF	V	E	/	/	Robust Quadrilaterals	2D	M	E	G
[83]	DB	AF	V	E	/	/	MSPA	3D	S	E	G
[59]	DB	AB	V	E	D	Radio Interferometrie	dwMDS	2D	S	E	S
[61]	DB	AF	V	E	/	/	AFL	3D	S	M	G
[22]	DB	AF	V	E	/	/	Spatial Maps	2D	S	E	S
[34]	DB	AB	Z	E	/	/	SAL	2D	S	E	G
[42]	DF	AB	V	A	I	RSSI	Stochastic Approach	3D	S	E	G
[3]	DB	AB	Z	E	I	RSSI	MLS	2D	S	M	G
[34]	DB	AB	Z	E	/	/	SAL	2D	S	E	G
[75]	DF	AB	V	E	/	/	dMCL,mMCL	2D	M	E	G
[27]	DF	AB	V	A	/	/	APIT	2D	S	E	G
[71]	DF	AB	V	A	/	/	/	2D	S	E	/
[16]	DB	AB	V	E	/	/	MDS, PDM	2D	S	M	S
[20]	DB	AB	V	A	/	RSSI	ROCRSSI+	2D	S	E	G
[29]	DF	AB	V	E	/	Proximity	sMCL	3D	M	M	G
[40]	DF	AB	V	A	/	RSSI	ROCRSSI	2D	S	E	G
[73]	DF	AB	V	E	/	/	Geometry conjecture	2D	M	E	G
[80]	DF	AB	V	A	/	RSSI/	FRORF	2D	S	E	G
[82]	DF	AB	V	A	/	Proximity	CAB	2D	S	E	G
[64]	DF	AB	V	E	/	/	MSL,MSL*	2D	M	M	G
[69]	DF	AF	V	E	/	/	MDS	2D	S	M	S
[54]	DB	AB	V	E	/	/	APS	2D	M	M	S
[7]	DF	AF	V	E	/	Hop-Count	Multilateration	2D	S	M	G
[33]	DF	AF	Z	E	/	Konnektivität	MDS-MAP	2D	S	E	G
[70]	DF	AF	Z	E	/	Konnektivität	MDS-MAP	2D	S	M	G

Tabelle 4.2: Algorithmenklassifizierungen

5 Implementierung

In diesem Kapitel wird die Umsetzung und Implementierung unseres Konzeptes vorgestellt. Zu Beginn beschreiben wir die Entwicklung der Software zur Evaluation der Algorithmen. Danach wird die Vorgehensweise zur Erstellung des Phasendifferenz-Distanzgenerators beschrieben. Zum Schluss werden die Parameter der Experimente in denen wir die Algorithmen testen vorgestellt.

5.1 Entwicklung einer modularen Evaluationssoftware

Die Software zur Evaluierung der Lokalisierungsalgorithmen wurde mithilfe der Programmiersprache Python entwickelt. Python ist eine objektorientierte, aspektorientierte und funktionale Programmiersprache. Python bietet eine umfangreiche Standardbibliothek und eine Menge weiterer zusätzlicher Pakete im Python Package Index an. Dies ermöglicht eine schnelle Programmierung von Programmen. Mithilfe der einfachen Syntax entsteht ein kurzer und gut lesbarer Quelltext. Da Pythoncode interpretiert wird, kann es jedoch bei rechenlastigen Programmen zu Performanceproblemen kommen. Auch Parallelität ist mit Python schwer zu implementieren. Wir wählen Python insbesondere wegen der `sk-learn`-, `Numpy`- und `Pyplot`-Bibliotheken welche für uns hilfreiche Funktionalitäten bereitstellen.

Nachfolgend wird die Entwicklung der Simulationssoftware beschrieben. Dabei gehen wir zu Beginn auf Anforderungen ein, welche die Software zu erfüllen hat, um anschließend den Aufbau und die Nutzung der Software zu erläutern.

5.1.1 Anforderungen an die Software

Eine umfangreiche Simulationssoftware für Sensornetzwerke ist sehr komplex und nicht im Rahmen dieser Arbeit. Um die Programmierung der Software so minimal und einfach wie möglich zu gestalten, definieren wir vor Beginn der Programmierung Anforderungen welche die Software erfüllen sollte. Dies hilft uns dabei, zielstrebig zu Programmieren und uns nicht in dem Erstellen eines großen Simulationframeworks zu verfangen.

Die genutzten Modelle und Konzepte innerhalb der Software werden einfach gehalten, damit wir uns auf die eigentliche Evaluation der Algorithmen konzentrieren können. Bei späterer Betrachtung der Evaluationsergebnisse sollte dies mit einbezogen werden. Die minimalen Anforderungen die wir an die Software stellen, ist zum einen die Modellierung eines zweidimensionalen Spielfeldes. Zusätzlich sollten Sensorknoten innerhalb dieses Spielfeldes statisch verteilt werden. Die Knoten sollten miteinander kommunizieren und Distanzen bestimmen können wenn sie sich in einer bestimmten Reichweite zueinander befinden. Die Lokalisierung der Sensorknoten, bestehend aus Distanzberechnung und Positionsbestimmung, sollte modular aufgebaut sein und ein einfaches Austauschen der jeweiligen Methoden erlauben. Weiterhin ist es wichtig, dass neue Algorithmen einfach in die Software implementiert werden können. Zur Evaluierung der implementierten Algorithmen sollten die dafür vorgesehenen Experimente einfach definierbar und die Bedienung der Software verständlich sein.

Die wichtigsten Aspekte der zu erstellenden Software sind folgend noch einmal aufgelistet:

- Modularer Aufbau
- Leicht bedienbar
- Minimal und einfach gehalten

5.1.2 Aufbau der Software

Für die Entwicklung der Software wurde ein objektorientierter Ansatz gewählt. Die in unseren Anforderungen gewünschten Konzepte wurden mithilfe von erstellten Klassen abgebildet.

Die dabei wichtigsten von uns modellierten Objekte sind:

- Knoten
- Spielfeld
- Experiment
- Szenario
- Distanzgenerator
- Lokalisierungsalgorithmus

Die verschiedenen Objekte interagieren auf unterschiedlichen Wegen miteinander. Manche Objekte beinhalten andere oder kommunizieren mit ihnen. Ein Überblick über den Aufbau und das Zusammenspiel der Klassen ist in Abbildung 5.1 gegeben.

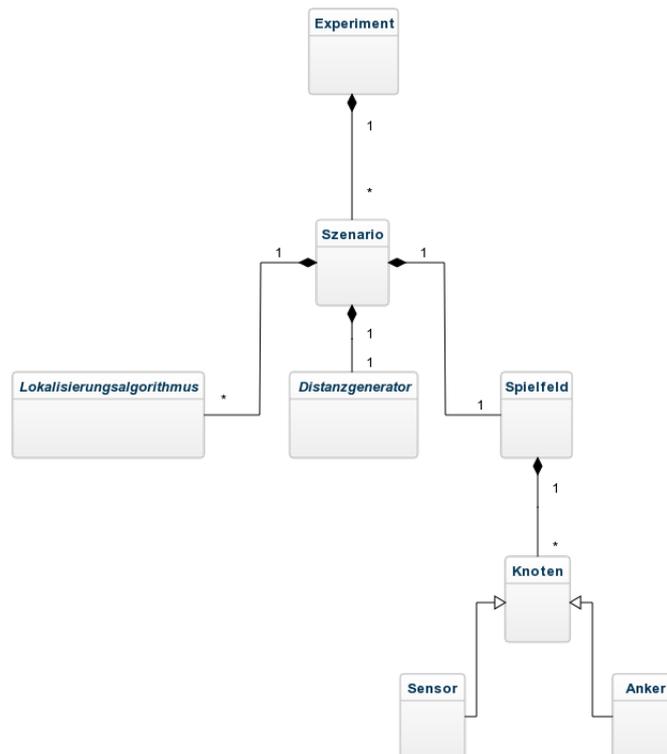


Abbildung 5.1: Klassendiagramm der Evaluationssoftware

Ein Experiment beschreibt die Evaluation der Algorithmen. Es ist aus mehreren Szenarien aufgebaut in denen die Algorithmen analysiert werden. Ein

Szenario beschreibt vollständig ein Sensornetzwerk. Die Szenarien innerhalb eines Experimentes unterscheiden sich nur in den zu variierenden Parametern. Die wichtigsten Bestandteile eines Szenarios sind die zu evaluierenden Lokalisierungsalgorithmen, der zu nutzende Distanzgenerator sowie das Spielfeld mit den Sensorknoten. Ein Spielfeld besteht aus einer gewissen Anzahl von Sensorknoten, welche entweder Ankerknoten sind oder normale Sensorknoten welche sich lokalisieren müssen.

Die Klassen des Distanzgenerators sowie des Lokalisierungsalgorithmus sind abstrakt und definieren ein von der Simulation genutztes Interface. Ausgeprägte Implementierungen erben von diesen Klassen um innerhalb der Simulation genutzt werden zu können. Für die Lokalisierungsalgorithmen implementieren wir die von uns gewählten Algorithmen. Auch wurden unterschiedliche Generatoren implementiert, welche die Distanz zwischen zwei Knoten auf verschiedene Weise berechnen.

5.1.3 Nutzung der Software

Das Programm liest die zur Erstellung von Experimenten benötigten Parameter aus einer CSV-Datei. Jede Zeile innerhalb dieser Datei beschreibt vollständig ein Experiment. Die zu variierenden Parameterwerte eines Experiments werden mithilfe von Semikolons getrennt. Es können bis zu drei Parameter innerhalb eines Experimentes variiert werden. Nach Ausführung der Experimente wird für jedes Experiment ein Ordner mit den Ergebnissen erstellt.

5.2 Erstellung eines virtuellen Phasendifferenzdistanzgenerator

Der virtuelle Phasendifferenz-Distanzgenerator stellt realistische Distanzmesswerte für unsere Simulationsexperimente bereit. Der Generator wird mithilfe einer Lerndatenmenge und einer darauf basierenden Wahrscheinlichkeitsverteilung erstellt. In diesem Kapitel wird die Sammlung der Distanzmessungen für die Lerndaten beschrieben sowie die Erstellung und Auswahl einer passenden Wahrscheinlichkeitsverteilung.

5.2.1 Sammlung der Messdaten

Die gemessenen Distanzwerte wurden mithilfe des ZIGPOS Positioning Kit gesammelt. Zwei der in dem Kit enthaltenen Sensorknoten wurden innerhalb eines Raumes der Universität platziert. Es wurde darauf geachtet, die Umgebung möglichst rauschfrei zu gestalten. Um den Einfluss von Interferenzen so minimal wie möglich zu halten, wurden die Messwerte nachts gesammelt wenn nur wenige Personen sich im Gebäude befanden. Um Reflektionen gering zu halten wurden die beiden Sensorknoten auf der Kante von Kartons platziert, so dass sich zwischen ihnen nicht direkt der Boden befindet.

Zur Erstellung des Lerndatensets wurden die beiden Sensorknoten in einem Abstand von 16cm voneinander aufgestellt. Innerhalb dieses Abstandes wurden Distanzmessungen durchgeführt und abgespeichert. Anschließend wurde die Knoten 2cm weiter voneinander entfernt aufgestellt und die Messungen wiederholt. Dies wurde bis zu einem Abstand von 250cm durchgeführt.

Die gesammelten Messdaten wurden in eine Comma-separated values (CSV)-Datei geschrieben welche wie folgt aufgebaut ist:

Von | Zu | DQF | Messdistanz | Status | Reichweite

Die erste Spalte bezeichnet die ID des Initiators und die zweite Spalte die ID des Reflektors zwischen denen die Distanzmessung stattgefunden hat. Die dritte Spalte enthält den zu der Distanzmessung gehörenden DQF-Wert. In der vierten Spalte wird die gemessene Distanz wiedergegeben. Der Status beschreibt den Status des Knotens während der Messung. Die letzte Spalte beschreibt die wahre Distanz zwischen den Knoten.

Die für unsere Arbeit interessanten Werte sind die gemessene Distanz, der DQF-Wert sowie die wahre Entfernung der Knoten.

5.2.2 Aufwertung der Messdaten

Um die Qualität der gesammelten Distanzmesswerte zu verbessern, werden die Daten anhand ihres DQF-Wertes gefiltert. Dieser Filter besteht aus einem minimalen und maximalen DQF-Wert und entfernt jede Messung die nicht innerhalb dieser Grenzen liegt. Zur Bestimmung eines guten Lerndatensets,

wurden verschiedene Filter genutzt und die resultierende Datenmenge bewertet. Wir bewerten die gefilterte Datenmenge anhand ihrer restlichen Anzahl an Messpunkten und dem durchschnittlichen Messfehler zur wahren Entfernung. Messwerte mit einem DQF-Wert von 0 werden automatisch ausgefiltert, da dieser Wert eine ungültige Messung kennzeichnet.

Die Ergebnisse sind in Tabelle 5.1 aufgelistet.

Minimaler DQF	Maximaler DQF	Fehler	Restpunkte
1	100	16.0873728814	23600
10	100	16.0873728814	23600
20	100	16.0873728814	23600
30	100	16.0873728814	23600
40	100	16.0873728814	23600
50	100	16.0873728814	23600
60	100	16.1504315133	23406
70	100	16.2706566621	22523
80	100	15.7498469246	17965
90	100	14.1722579762	7303

Tabelle 5.1: Messfilter

In der ersten Spalte wird der für eine Messung minimal benötigte DQF-Wert beschrieben und in der zweiten Spalte der maximale DQF-Wert. Die dritte Spalte beschreibt den Fehler der resultierenden Datenmenge. Dieser Fehler ergibt sich aus dem durchschnittlichen Abstand zwischen gemessender Distanz und wahren Entfernung der Messungen. Die vierte Spalte beschreibt die Anzahl der in der gefilterten Datenmenge vorhandenen Messpunkte.

Der maximale DQF-Wert ist für die verschiedenen Filter gleich. Der minimale DQF-Wert wird schrittweise erhöht. Wir erwarten, dass bei steigendem minimalem DQF-Wert der Fehler sinkt, da wir mögliche fehlerhafte Messwerte, welche einen niedrigen DQF-Wert haben, aussortieren. Dieses Verhalten ist in der Tabelle auch grundsätzlich erkennbar, obwohl der Fehler kurzzeitig etwas ansteigt. Der Unterschied zwischen einem Filter mit minimalem DQF-Wert von 50 und einem minimalen DQF-Wert von 90 beträgt nur knapp um die 2cm. Gleichzeitig gehen aber um die 16000 Messpunkte verloren. Die Relation zwischen Fehlerverbesserung und Messpunktanzahlverlust ist unverhältnismäßig. Der Verlust der Messpunkte ist dabei nicht gleichmäßig über die Entfernungen

verteilt, sondern es werden ganze Messreihen ausgefiltert werden. In Abbildung 5.2 ist dies erkennbar. Dargestellt ist die entstehende Lerndatenmenge bei einem Filter mit einem minimalen DQF-Wert von 90. Rot gekennzeichnet sind die Punkte die bei diesem Filter wegfallen. Schwarz dargestellt ist die entstehende Lerndatenmenge. Man kann erkennen, dass insbesondere ab einer Entfernung von 200cm die meisten Distanzmessungen ausgefiltert werden. Für die Erstellung einer Wahrscheinlichkeitsverteilung ist dies sehr nachteilhaft.

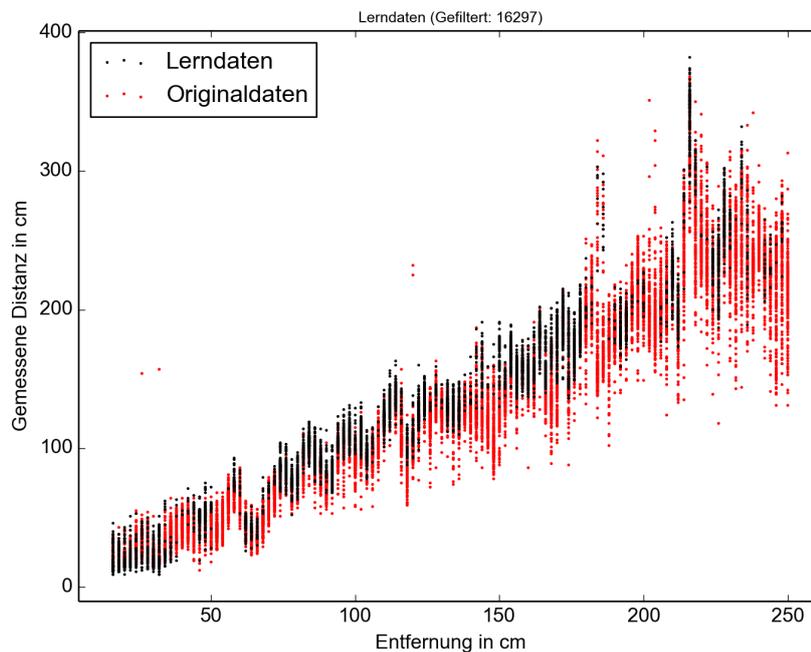


Abbildung 5.2: Gefilterte Datenmenge für einen Filter mit minimalem DQF-Wert von 90

Anhand der oben genannten Beobachtungen haben wir uns dazu entschlossen einen Filter mit einem minimalen DQF-Wert von 50 und einem maximalen DQF-Wert von 100 zu nutzen. Die resultierende Messdatenmenge ist in Abbildung 5.3 zu sehen. Es ist gut zu erkennen, dass die Messdistanzen nur annähernd linear mit der Entfernung steigen. An verschiedenen Stellen gibt es sogar Einbrüche in den Distanzmessungen wo die gemessene Distanz trotz steigender Entfernung sinkt. Auch ist zu erkennen, dass die gemessenen Distanzen stark streuen. Insbesondere bei steigender Entfernung verstärkt sich die Größe der Streuung.

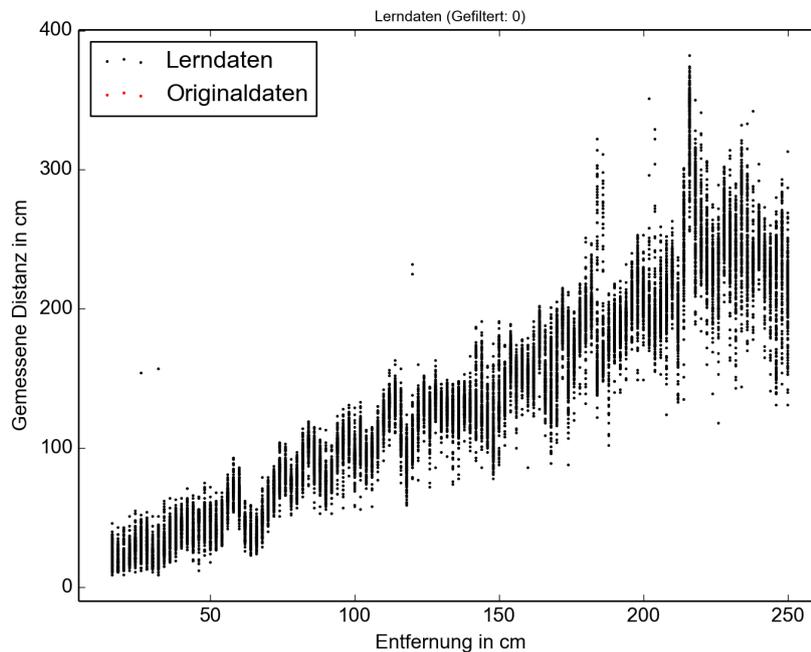


Abbildung 5.3: Gefilterte Datenmenge

Die durch den Filter erstellten Lerndatenmengen werden in einer CSV-Datei mit folgendem Aufbau abgespeichert:

Messdistanz | Reichweite

5.2.3 Finden eines gaußschen Mischmodells

Mithilfe der gefilterten Distanzmessungen erstellen wir ein gaußsches Mischmodell welches die Verteilung der Lerndaten widerspiegelt.. Die Erstellung sowie das Training eines Mischmodells wird mithilfe des Python-Pakets scikit-learn durchgeführt. Für die Erstellung eines gaußschen Mischmodells mithilfe dieser Pythonbibliothek sind grundsätzlich zwei Parameter entscheidend. Zum einen die zu nutzende Komponentenanzahl sowie die Art der Kovarianzmatrix. Wir wählen einen diagonalen Kovarianztyp. Eine passende Komponentenanzahl bestimmen wir indem mehrere gaußsche Mischmodelle mit unterschiedlichen Komponentenanzahlen erstellt, bewertet und verglichen werden.

Die Ergebnisse der verschiedenen Modelle sind in Tabelle 5.2 beschrieben.

Die Tabelle zeigt in der ersten Spalte die Komponentenzahl des Modells. Das erlernte Mischmodell stellt einen Akaike information criterion (AIC)-Wert und einen Bayesian Information Criterion (BIC)-Wert bereitete welche in Spalte Zwei und Drei beschrieben sind. Die vierte Spalte enthält den relativen Fehler des Modells. Dieser Fehler wird so berechnet, dass für jeden Wert aus der Grunddatenmenge ein Wert für die jeweilige Distanz aus dem Modell gezogen wird. Der Mittelwert der Differenzen dieser Tupel ergibt den relativen Fehler des Modells.

In der Tabelle kann man erkennen, dass bei steigender Komponentenanzahl alle drei Referenzwerte sinken. Es lässt sich daraus schließen, dass das Modell genauer wird je höher die Komponentenanzahl ist.

Bei der grafischen Betrachtung der Distanzdaten eines Modells mit 240 Komponenten ist erkennbar, dass diese Schlussfolgerung falsch ist. In Abbildung 5.4 sind von einem Mischmodell mit 240 Komponenten erstellte Distanzmessungen abgebildet. In einem Intervall von 1 bis 250cm wurden alle 1cm 50 Messungen mit dem Modell berechnet und dargestellt. Die von dem Modell erstellten Messungen sind rot markiert. Der Lerndatensatz des Modells ist zum Vergleich in Schwarz abgebildet. Es ist klar zu erkennen, dass die virtuellen Distanzdaten nur teilweise den Lerndaten entsprechen. Insbesondere in den Bereichen 1-10cm, 45-110cm und 150-180cm lassen sich typische Strukturen eines Overfitting erkennen. Für die Entfernungen, für die es keine Originaldaten gibt, da die realen Messungen nur alle 2cm ausgeführt wurden, sind die Distanzwerte des Modells stark fehlerhaft.

Eine Auswahl nur basierend auf den Tabellenwerten ist dementsprechend nicht möglich. Die weitere Auswahl eines geeigneten Mischmodells wurde anschließend mithilfe dieser grafischen Darstellungen der virtuellen Distanzdaten durchgeführt. Dafür wurde mit einem Modell mit einer geringen Komponentenanzahl angefangen und die Komponentenanzahl schrittweise erhöht. In Abbildung 5.5 ist die Evaluation eines Modells mit zu wenigen Komponenten dargestellt. Es sind keine Strukturen eines Overfittings zu erkennen, jedoch weichen die erstellten Distanzdaten stark von den Originaldaten ab.

Wir erhöhen die Komponentenanzahl solange bis die erstellten Distanzdaten zu den Originaldaten visuell passen. Falls ein Overfitting erkennbar wird, verringern wir die Komponentenanzahl wieder. Dies führen wir solange durch, bis ein passendes Modell gefunden wurde.

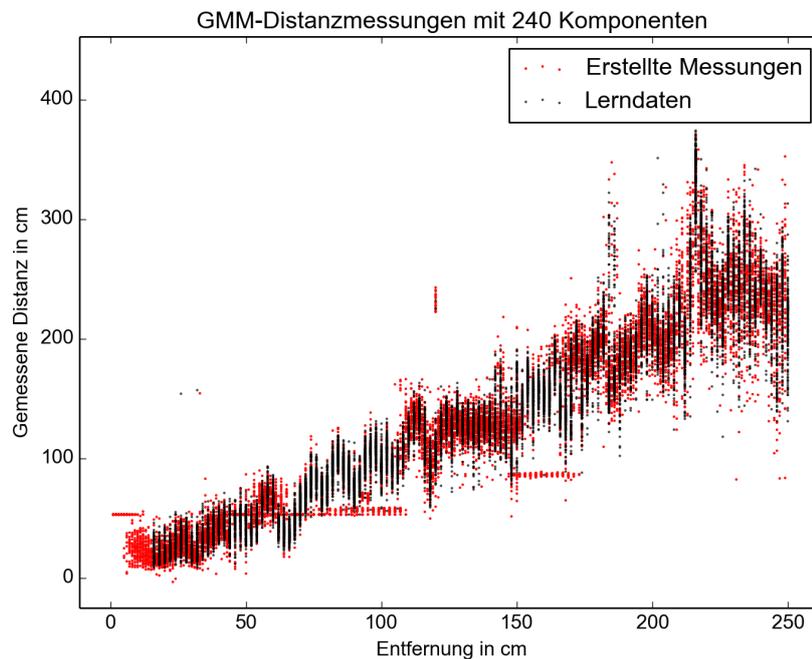


Abbildung 5.4: Zu hohe Komponentenanzahl

Für den Phasendifferenz-Distanzgenerator wird ein gaußsches Mischmodell mit einer Komponentenanzahl von 100 genutzt. In Abbildung 5.6 sind Distanzmessungen aus diesem Modell abgebildet. Es ist zu erkennen, dass die erstellten Werte sich sehr gut den Originaldaten annähern. Auch in Bereichen in denen keine Originaldaten vorhanden sind, beispielsweise in dem Bereich von 0 bis 16cm, sind die Werte des Generators plausibel.

5.3 Erstellung der Experimente

Für die Erstellung der Experimente wird ein Grundszenario entworfen, in welchem die Lokalisierungsalgorithmen evaluiert werden. Die Parameter des Grundszenarios sind in Tabelle 5.3 aufgelistet.

Die Radioreichweite gibt an, bis zu welchem maximalen Abstand Knoten miteinander kommunizieren und die Distanz zwischen sich messen können. Der Rauschfaktor gibt den Faktor an mit dem wir die Kovarianzmatrix des Mischmodells multiplizieren, um die gemessenen Distanzwerte ungenauer zu gestalten.

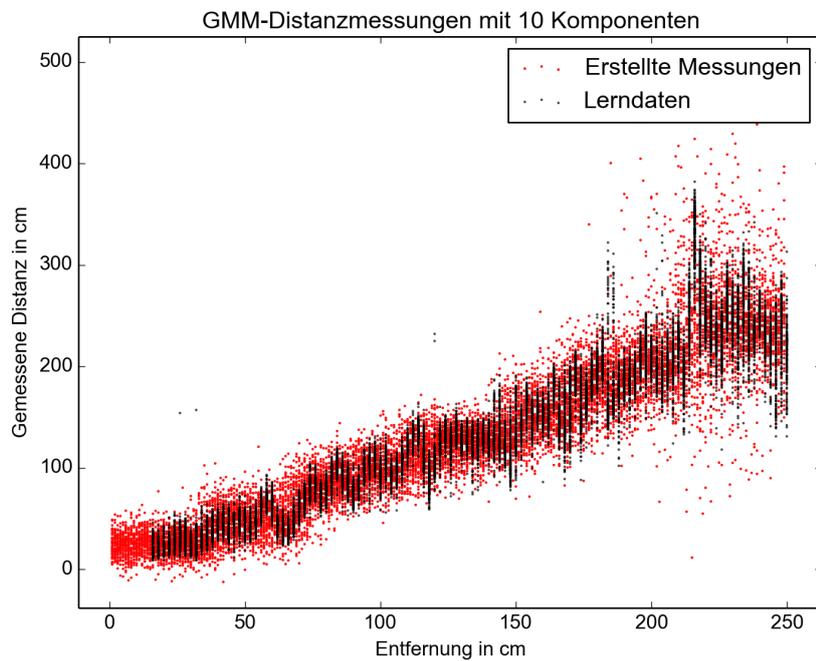


Abbildung 5.5: Zu geringe Komponentenanzahl

ten. Die Messanzahl gibt an wie oft eine Distanzmessung durchgeführt wird. Aus den Distanzmessungen wird dann mithilfe der genutzten Messmetrik ein einzelner Distanzwert bestimmt. Der boolsche Parameter des Pseudoankers gibt an, ob Sensorknoten, welche sich schon lokalisiert haben, als Pseudo-Anker für noch nicht-lokalisierte Sensorknoten genutzt werden. Die Ankeranzahl bestimmt die im Spielfeld platzierten Ankerknoten und die Sensoranzahl die im Spielfeld platzierten Sensorknoten. Die Parameter der Anker- und Sensorverteilung geben an, auf welche Weise die jeweiligen Knoten im Spielfeld platziert werden. Die Anzahl der Wiederholungen gibt an wie oft das gleiche Szenario mit einer jeweils unterschiedlichen Saat für den Zufallsgenerator durchgeführt wurde. Wir erstellen zwei Experimente in denen wir jeweils einen Parameter variieren.

In dem ersten Experiment lassen wir die Anzahl der Anker von 7 bis 20 schrittweise variieren. In dem zweiten Experiment wird der Rauschfaktor schrittweise um 2 bis auf 10 erhöht.

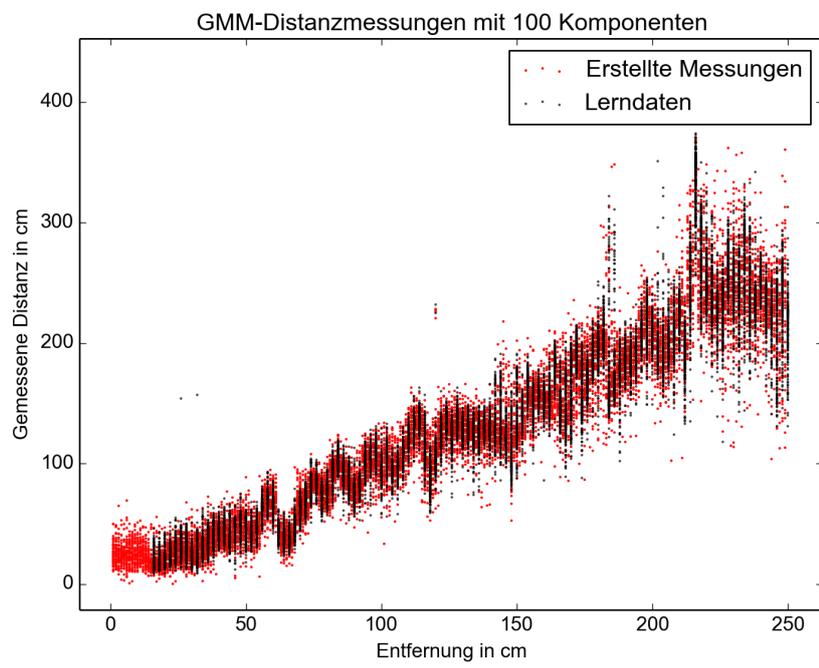


Abbildung 5.6: Passende Komponentenanzahl

Komponenten	AIC	BIC	Fehler
10	460802.792272	461198.173369	24.9898045359
20	457683.545788	458482.376986	23.4069130866
30	454544.317216	455746.598513	21.6687559585
40	454889.342101	456495.073498	21.7491055475
50	446332.872912	448342.054408	19.6736889808
60	449628.107596	452040.739191	20.2699707047
70	448527.944389	451344.026084	20.3802663266
80	442202.335454	445421.867248	19.0859628544
90	438305.255307	441928.237201	18.4917911616
100	434624.229222	438650.661216	18.5267341437
110	429885.559405	434315.441498	17.9672838838
120	434451.704615	439285.036807	18.0534499829
130	424593.070558	429829.85285	17.5528484143
140	421546.147945	427186.380337	17.8334895299
150	413099.40681	419143.089301	17.6999850208
160	417715.11003	424162.242621	17.7190020519
170	405673.463628	412524.046318	17.5832318446
180	414141.182985	421395.215775	17.5164088093
190	395426.794118	403084.277007	17.3852303169
200	389512.312635	397573.245624	17.6260945387
210	379768.72362	388233.106709	17.7654780638
220	373751.189604	382619.022792	17.5423821821
230	365732.919913	375004.203201	17.4325131562
240	363209.856579	372884.589966	17.5852318669

Tabelle 5.2: Unterschiedliche Komponentenanzahl für Mischmodelle

Parameter	Wert
Radioreichweite	250cm
Rauschfaktor	0
Messanzahl	10
Messmetrik	Median
Distanzgenerator	Phasendifferenz
Pseudoanker	Wahr
Ankeranzahl	15
Sensoranzahl	100
Ankerverteilung	Zufällig
Sensorverteilung	Zufällig
Spielfeldgröße	5m x 5m
Wiederholungen	10

Tabelle 5.3: Verschiedene Grundparameter für Experimente

6 Evaluation

In diesem Kapitel werden die Ergebnisse der durchgeführten Experimente vorgestellt. Wir evaluieren die verschiedenen Algorithmen in zwei unterschiedlichen Experimenten. Während des ersten Experiments wird die Anzahl der platzierten Anker variiert. Im zweiten Experiment wird der Rauschfaktor verändert. Für die Bewertung der Qualität der Algorithmen nutzen wir den mittleren Positionierungsfehler. Dieser ergibt sich aus der gemittelten Summe der Differenzen zwischen den wahren Sensorknotenpositionen und den durch die Algorithmen geschätzten Positionen. Am Ende dieses Kapitels findet eine Diskussion der ermittelten Ergebnisse statt.

6.1 Fehlerhafte MDS-Verfeinerung

Die Evaluation der Algorithmen hat deutlich gemacht, dass der Optimierungsschritt des MDS+-Algorithmus fehlerhaft ist. In Abbildung 6.1 sind der normale MDS-Algorithmus ohne Optimierungsschritt sowie drei weitere Algorithmen mit jeweils einer unterschiedlichen Anzahl an Optimierungsschritten abgebildet. Die Grafik beschreibt den mittleren Positionierungsfehler der unterschiedlichen Algorithmen für eine steigende Anzahl von Ankerknoten. Es ist deutlich zu sehen, dass die Algorithmen mit Optimierung schlechtere Positionen bestimmen als der Grundalgorithmus. Je mehr Iterationen die Optimierung durchläuft, desto größer wird der Fehler. Es ist anzunehmen, dass der Fehler quadratisch mit der Anzahl der Iterationen steigt.

Es ist nicht gelungen, den Grund für diese fehlerhafte Optimierung zu finden. Anzunehmen ist, dass der Optimierungsalgorithmus falsch beschrieben worden ist. In den weiteren Experimenten wurde der Optimierungsschritt entfernt und nur noch der Grundalgorithmus MDS benutzt.

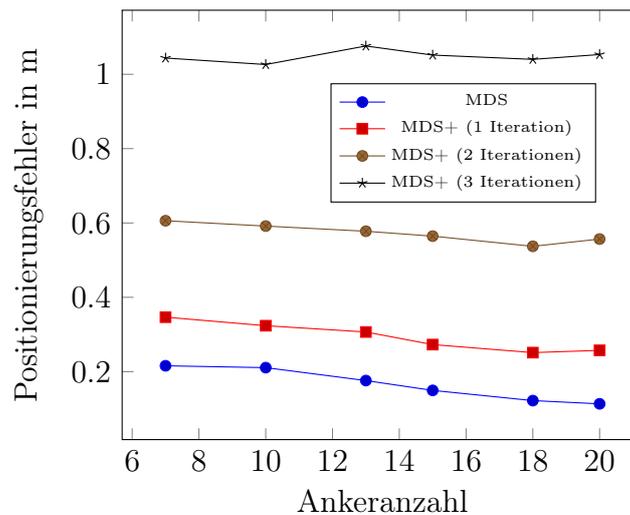


Abbildung 6.1: Unterschied zwischen MDS und MDS mit Trilateration für verschiedene Iterationen

6.2 Variierende Ankeranzahl

In diesem Experiment wurde die in dem Spielfeld platzierte Ankeranzahl variiert. Wir beginnen mit einer geringen Anzahl an platzierten Ankerknoten und erhöhen diese Schrittweise. Durch die zufällige Platzierung ist darauf zu achten, dass wir eine gewisse Mindestmenge an Ankern platzieren, da sonst durch eine ungünstige Platzierung kein Sensorknoten die zur Lokalisation nötige Mindestmenge von 3 Ankerknoten als Nachbarn hat.

6.2.1 Erwartungen

Für die Algorithmen erwarten wir, dass der Positionsfehler kleiner wird, wenn wir die Anzahl der Ankerknoten erhöhen. Durch eine höhere Anzahl an Ankerknoten existieren für einen Sensorknoten mehr Nachbarn mit einer korrekten Position die einen positiven Einfluss auf die Lokalisation hat. Dadurch können direkt zu Beginn, Sensorknoten eine qualitativ hochwertige Position bestimmen. Diese Knoten dienen anschließend als Anker mit einem geringen Positionsfehler. Dies hat Auswirkungen auf die nächste Iteration von lokalisierenden Sensorknoten und pflanzt sich so durch das Netzwerk fort. Desweiteren vermuten wir, dass der RLS+-Algorithmus besser abschneidet als der Robust

Least-Squares (RLS)-Algorithmus, besonders bei einer geringen Anzahl an Ankerknoten.

6.2.2 Ergebnis

Die Ergebnisse des Experiments sind in Abbildung 6.2 zu sehen. Dort ist auf der X-Achse die Anzahl der in dem Szenario platzierten Ankerknoten aufgetragen und auf der Y-Achse ist der mittlere Positionierungsfehler in Metern abgebildet. Der Positionierungsfehler befindet sich bei beiden Algorithmen in jedem Szenario im zweistelligen Zentimeterbereich. Der MDS-Algorithmus schneidet mit Abstand am Besten ab und liefert schon mit einer Ankeranzahl von 7 einen Positionierungsfehler den RLS+ erst mit einer Anzahl von 20 Ankerknoten erreicht. Auffallend ist, dass RLS+ während des ersten Szenarios mit 7 Ankern deutlich schlechter abschneidet als RLS, anschließend aber kontinuierlich besser ist.

MDS

MDS erzielt bei dem Anstieg von 7 auf 10 Ankerknoten keine deutliche Verbesserung. Anschließend sinkt der Fehler jedoch annähernd linear ab. Bei Betrachtung der Tabelle 6.1 kann man erkennen, dass die Positionierung in jedem Schritt um ca. 3cm genauer wird. Im Schritt von 18 auf 20 Ankerknoten wird nur eine Verbesserung von um die 1cm erreicht.

RLS+

Dieser Algorithmus weist bei einer Ankeranzahl von 7 einen großen Positionierungsfehler von nahezu einem halben Meter auf. Der Fehler sinkt bei steigender Ankeranzahl stark ab. Insbesondere im Schritt von 7 auf 10 Ankerknoten verbessert sich dieser Algorithmus um fast 11cm. Anschließend sinkt der Positionierungsfehler mit steigender Ankeranzahl annähernd linear ab. Den besten Wert erreicht dieser Algorithmus bei einer Ankeranzahl von 20 mit einem Positionierungsfehler von um die 20cm.

RLS

RLS hat bei einer Ankeranzahl von 7 einen Positionierungsfehler von um die 43cm. Bei der Erhöhung auf 10 Ankerknoten sinkt der Fehler nur gering. Erst ab einer Ankeranzahl von 13 erreicht dieser Algorithmus einen Fehler von um die 33cm und sinkt bis auf einen Fehler von 25cm bei 20 Ankerknoten ab. Auffallend ist, dass zwischen 18 und 20 Ankerknoten so gut wie keine Verbesserung zu erkennen ist.

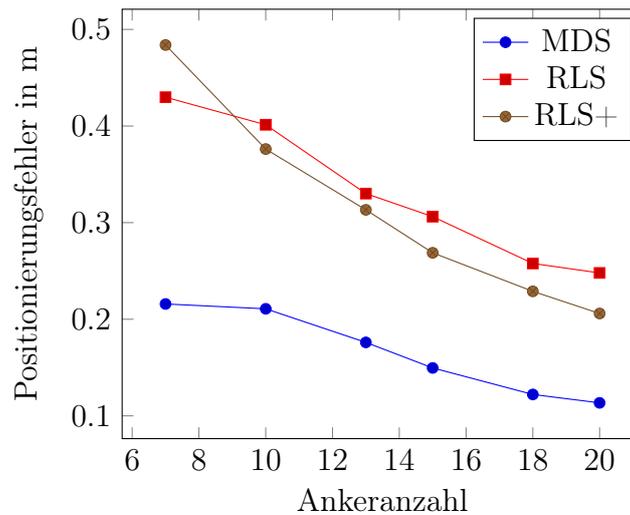


Abbildung 6.2: Mittlerer Positionierungsfehler bei variierender Ankeranzahl

Ankeranzahl	MDS	Multilateration	MultilaterationRefinement
7	0.21580778	0.4298087606	0.4838497691
10	0.2108045216	0.4012843734	0.3761619588
13	0.1760387066	0.330082421	0.3132315357
15	0.1496074359	0.3062133775	0.2687816215
18	0.122158097	0.2577387069	0.2288780455
20	0.1134652566	0.2479482558	0.2059535238

Tabelle 6.1: Mittlerer Positionierungsfehler bei variierender Ankeranzahl

6.2.3 Schlussfolgerung

Die Ergebnisse lassen erkennen, dass der MDS Algorithmus konstant die besten Positionen berechnet. Je höher die Ankeranzahl ist, desto genauer sind auch die

berechneten Positionen. Dies stimmt mit unseren Erwartungen überein. Eine für den MDS Algorithmus gute Ankeranzahl ist 18, da darüber hinaus die Verbesserung nur noch gering ist. Mit unseren Erwartungen stimmt auch überein, dass der RLS+ genauere Positionen berechnet als RLS. Unsere Erwartung dass dies besonders bei einer geringen Anzahl von Ankerknoten auffällig ist, hat sich nicht bestätigt. Dass bei einer Ankerknotenanzahl von 7 RLS+ schlechter ist als RLS lässt sich darauf zurückführen, dass weniger Knoten durch RLS+ aufgrund der Nachbarselektion in der Fehlerkontrolle lokalisiert worden sind und diesen Knoten eine zufällige Position gegeben wurde.

6.3 Variierender Rauschfaktor

Während dieses Experiments wird der Rauschfaktor, welcher die Genauigkeit der Distanzmessungen beeinflusst, erhöht. Die Kovarianzmatrix des für den Phasendifferenz-Distanzgenerator genutzten Mischmodells wird mit dem Rauschfaktor multipliziert. Durch die dadurch entstehenden größeren Werte in der Kovarianzmatrix streuen die erstellten Distanzen stärker. Die Varianz der Distanzmessungen steigt dabei linear mit dem Rauschfaktor an. In 6.3 sind Varianzen für verschiedene Rauschfaktoren dargestellt. Die Varianz ist dabei der Mittelwert der für die Distanzen von 1cm bis 250cm berechneten Varianzen von je 1000 Messungen.

Wie sich der Rauschfaktor auf die Distanzmessungen auswirkt ist in Abbildung 6.4 dargestellt. In der Grafik sind virtuelle Distanzmessungen für einen Rauschfaktor von 0 und einen Rauschfaktor von 5 abgebildet. Es ist zu erkennen, dass die verrauschten Messungen zwar dem Grundverlauf der Kurve folgen, aber stärker streuen.

6.3.1 Erwartungen

Bei diesem Experiment erwarten wir, dass die Qualität der Algorithmen bei steigendem Rauschfaktor abnimmt. RLS+ sollte dabei weit besser abschneiden als RLS, insbesondere je größer der Rauschfaktor wird. Desweiteren nehmen wir an, dass MDS gut mit den ungenauer werdenden Distanzmessungen zurechtkommt, da sich die Fehler innerhalb der Berechnungen gegenseitig ausgleichen.

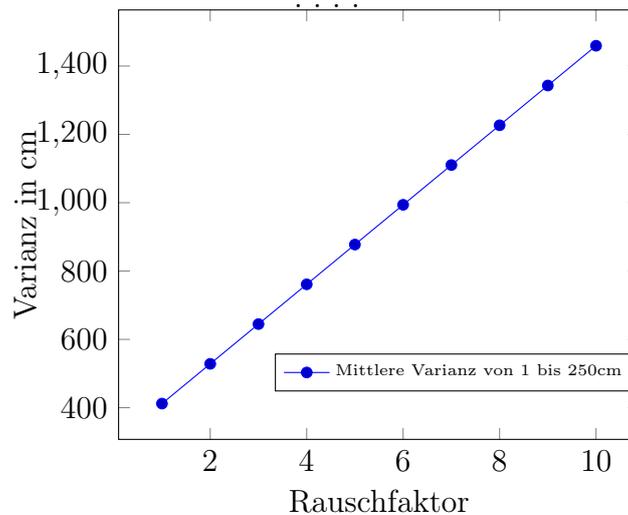


Abbildung 6.3: Varianz der Distanzmessungen für verschiedene Rauschfaktoren

6.3.2 Ergebniss

Die Ergebnisse des Experiments sind in Abbildung 6.5 zu erkennen. Auf der X-Achse ist die Größe des Rauschfaktors angegeben. Auf der Y-Achse ist der mittlere Positionierungsfehler in Metern aufgetragen. Man kann sehen, dass der Positionierungsfehler der Algorithmen größer wird bei wachsendem Rauschfaktor. Der Fehler bleibt konstant im zweistelligen Zentimeterbereich. MDS ist durchgängig der Algorithmus mit dem kleinsten Fehler. RLS+ wird bei steigendem Rauschfaktor relativ zu RLS immer besser.

MDS

Zu Beginn beträgt der Fehler von MDS um die 15cm. Mit der Erhöhung des Rauschfaktors steigt dieser Fehler annähernd linear mit ca. 1cm pro Schritt an. Bei einem Rauschfaktor von 10 beträgt der Fehler 20cm.

RLS+

RLS+ weist bei einem Rauschfaktor von 0 einen Positionierungsfehler von ca. 27cm auf. Bei wachsendem Rauschfaktor wird der Fehler größer. Der Anstieg

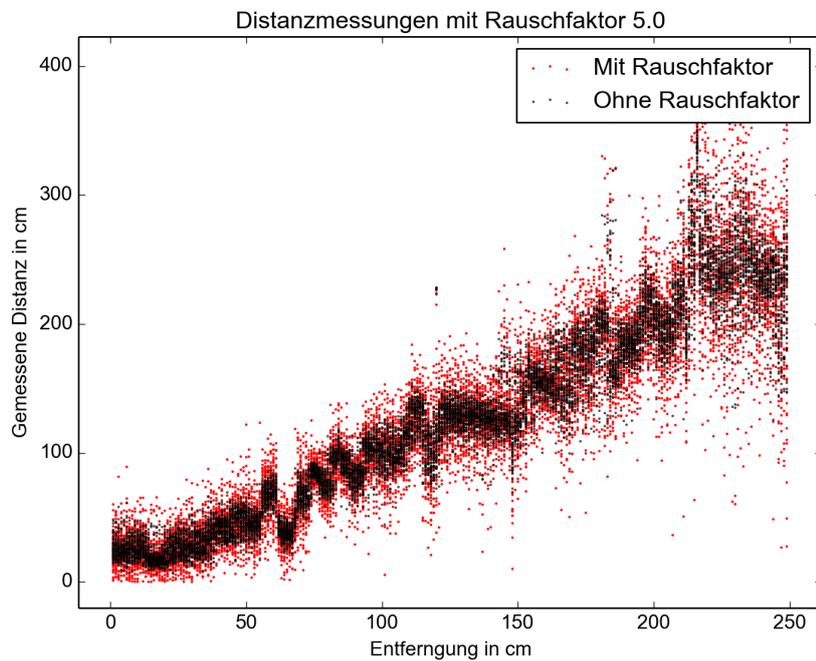


Abbildung 6.4: Distanzmessungen für einen Rauschfaktor von 5

beträgt dabei knapp 1cm pro Schritt und ist auch annähernd linear. Mit einem Rauschfaktor von 10 erreicht dieser Algorithmus einen Fehlerwert von um die 40cm.

RLS

Der Positionierungsfehler von RLS beträgt zu Beginn 30cm. Mit Steigerung des Rauschfaktors erhöht sich dieser Fehler zu Beginn erst leicht mit ca. 1cm und ab einem Rauschfaktor von 4 mit ca. 2cm pro Schritt. Mit einem Rauschfaktor von 10 hat dieser Algorithmus einen Positionierungsfehler von 39cm.

6.3.3 Schlussfolgerung

Aus den Ergebnissen lässt sich erkennen, dass MDS auch bei steigendem Rauschfaktor der qualitativ beste Algorithmus ist. MDS kommt gut mit den ungenauer werdenden Distanzmessungen zurecht, so wie wir es erwartet haben. Auch RLS+ schafft es bei steigendem Rauschfaktor nur einen linearen Anstieg

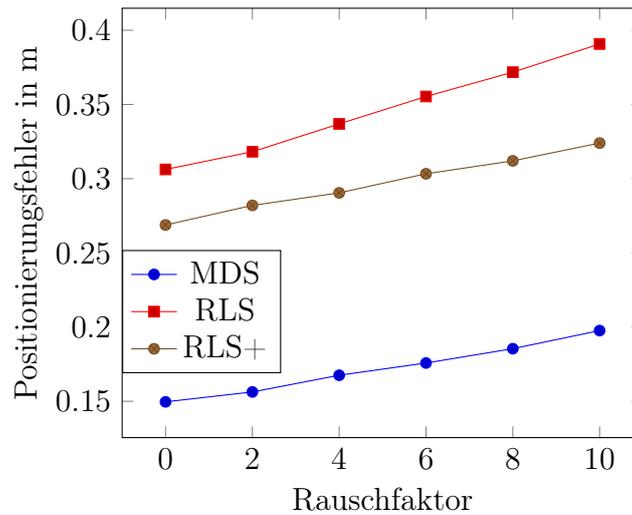


Abbildung 6.5: Mittleres Positionierungsfehler bei variierendem Rauschfaktor

Rauschfaktor	MDS	Multilateration	MultilaterationRefinement
0	0.1496074359	0.3062133775	0.2687816215
2	0.1562727126	0.3181441516	0.2819742558
4	0.167525959	0.3368855775	0.2904238683
6	0.1757680563	0.3553578633	0.3032840197
8	0.1854219156	0.3717401242	0.3119804806
10	0.1976184415	0.3907203884	0.3239677046

Tabelle 6.2: Mittleres Positionierungsfehler bei variierendem Rauschfaktor

des Positionierungsfehlers zu haben. Mit unseren Erwartungen trifft auch überein, dass RLS+ besser mit den ungenauen Distanzmessungen zurechtkommt als RLS+. Dies ist insbesondere bei hohen Rauschfaktoren erkennbar. Dies lässt den Schluss zu, dass die Fehlerkontrolle des RLS+ hilfreich ist.

6.4 Diskussion

Die Evaluation der Algorithmen ergibt, dass MDS die genauesten Positionen berechnet. Trotz fehlendem Optimierungsschritt kann MDS bei jeder Ankeranzahl sowie auch bei jedem Rauschfaktor einen geringeren Fehler aufweisen als die anderen Algorithmen. Der Fehler ist jedoch stets über 10cm und daher

nur bedingt für eine präzise Indoorlokalisierung geeignet. Als Basis für Optimierungsmethoden bietet MDS jedoch einen sehr guten Startwert.

Im Gegensatz dazu kann RLS+, trotz Fehlerkontrolle und robuster Formulierung, keine für eine Indoorlokalisierung geeigneten Positionen bestimmen. Die Nutzung der Fehlerkontrolle hilft bei der Eingrenzung des Positionierungsfehlers und kann besonders bei sehr schlechten Distanzmessungen nützlich sein. Wie stark sich diese Nutzung lohnt ist jedoch fraglich, da die Einstellung der für die Fehlerkontrolle erforderlichen Parameter, besonders in realen Szenarien sehr aufwendig ist. Auch ist problematisch, dass sich manche Knoten, besonders bei einer geringen Ankerzahl, gar nicht lokalisieren können. Dieses Problem wäre aber wohl auch durch eine Anpassung der Parameter eingrenzbar. Der normale RLS-Algorithmus kann keine guten Positionsschätzungen berechnen und ist für eine Indoor-Lokalisierung nicht geeignet. Dieser Algorithmus ist besonders anfällig für Fehler in den Distanzmessungen.

7 Zusammenfassung

In dieser Arbeit wurden verschiedene robuste Lokalisierungsalgorithmen (MDS, RLS+, RLS) innerhalb einer Simulation mit realistischen Distanzwerten in unterschiedlichen Szenarien analysiert. Dabei wurden wissenschaftliche Veröffentlichungen zum Thema Funklokalisierung gesichtet und Lokalisationsansätze klassifiziert und ausgewählt. Zur Erstellung der realistischen Distanzwerte innerhalb der Simulation wurde mithilfe eines gaußschen Mischmodells und real gemessenen Sensordaten ein virtueller Phasendifferenz-Distanzgenerator entwickelt.

Unsere Ergebnisse zeigen, dass MDS sehr viel genauere Positionen berechnet als RLS+ und RLS und sich somit für eine Indoor-Positionierung besser eignet. Die Fehlerkontrolle des RLS+-Algorithmus verbessert die Genauigkeit der bestimmten Positionen, insbesondere je ungenauer die gemessenen Distanzen werden. Eine Steigerung der Ankeranzahl wirkt sich auf alle Algorithmen positiv aus. RLS+ und MDS gehen mit schlechter werdenden Distanzmessungen ähnlich gut um. Erstaunlich war, dass diese beiden Algorithmen auch bei stark verrauschten Messungen noch annehmbare Positionsschätzungen berechnen konnten. Während der Erstellung des Phasendifferenz-Distanzgenerators war erkennbar, wie sehr reale Distanzmessungen schwanken und sich nur gering linear gegenüber der Entfernung verhalten. Trotz dieser Ungenauigkeiten konnten die gesammelten Daten mit einem passenden gaußschen Mischmodell gut modelliert und somit ein realistischer Distanzgenerator entwickelt werden.

Während dieser Arbeit ist besonders beim Sammeln der realen Messwerte mithilfe der Sensorknoten aufgefallen, wie schwer es ist, in realen Einsatzszenarien Distanzen zwischen Knoten zu berechnen und wie ungenau diese ausfallen. Auch bei einer angepassten Umgebung schwankten die berechneten Distanzen noch sehr. Während der Auswahl der zu evaluierenden Algorithmen war die große Anzahl existierender Lokalisationsansätze auffällig. Es gibt viele verschiedene Methoden von denen die Wenigsten ausreichend getestet wurden. Die

Implementation der Algorithmen ergab desweiteren, dass die Beschreibung der Ansätze kritisch betrachtet werden sollte, da diese auch fehlerhaft sein können.

Die Ergebnisse dieser Arbeit lassen viel Raum für weitere Forschung. MDS konnte trotz fehlender Optimierungsphase sehr gute Ergebnisse liefern. Eine Evaluation von MDS-Algorithmen mit verschiedenen Optimierungsansätzen bietet daher einen interessanten Ansatz zur Indoor-Lokalisation. Auch eine Einbindung der Fehlerkontrolle des RLS+-Algorithmus ist eine interessante Möglichkeit. Eine Verbesserung des entwickelten Distanzgenerators ist wünschenswert. Dies kann durch die Erstellung mehrerer und besserer Lernmengen geschehen. Auch die Entwicklung einer besseren Evaluierung der zu den Lerndaten passenden Mischmodellen und ihrer Komponentenanzahl bietet einen interessanten Forschungsansatz.

Diese Arbeit hat gezeigt, dass an einer präzisen Indoor-Lokalisation weiterhin geforscht werden muss. Insbesondere eine sinnvolle und realistische Evaluation existierender Lokalisierungsalgorithmen ist erstrebenswert.

Literaturverzeichnis

- [1] Samira Afzal. A Review of Localization Techniques for Wireless Sensor Networks. *Journal of Basic and Applied Scientific Research*, 2(8):7795–7801, 2012.
- [2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.
- [3] Cesare Alippi and Giovanni Vanini. A RSSI-based and calibrated centralized localization technique for Wireless Sensor Networks. *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)*, pages 1–5, 2006.
- [4] Atmel Corporation. Atmel AVR2162: REB233SMAD - Hardware User Manual. (April):1–15, 2011.
- [5] Atmel Corporation. Low Power, 2.6GHz Transceiver for ZigBee, RF4CE, IEEE 802.15.4, 6LoWPAN, and ISM Applications. (MCU Wireless AT86RF233):225, 2011.
- [6] Atmel Corporation. Atmel Wireless Solutions 1 © 2013. 2013.
- [7] Jonathan Bachrach, Radhika Nagpal, Michael Salib, and Howard Shrobe. Experimental results for and theoretical analysis of a self-organizing global coordinate system for ad hoc sensor networks. *Telecommunication Systems*, 26(2-4):213–233, 2004.
- [8] Jonathan Bachrach and Christopher Taylor. Localization in Sensor Networks. *Handbook of Sensor Networks: Algorithms and Architectures*, pages 277–310, 2005.
- [9] P. Bahl and V.N. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, 2, 2000.

- [10] Paolo Baronti, Prashant Pillai, Vince W C Chook, Stefano Chessa, Alberto Gotta, and Y. Fun Hu. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards. *Computer Communications*, 30(7):1655–1695, 2007.
- [11] Jan Blumenthal, Ralf Grossmann, Frank Golatowski, and Dirk Timmermann. Weighted centroid localization in Zigbee-based sensor networks. *2007 IEEE International Symposium on Intelligent Signal Processing, WISP*, 2007.
- [12] Jan Blumenthal, Frank Reichenbach, and Dirk Timmermann. Position Estimation in Ad hoc Wireless Sensor Networks with Low Complexity.
- [13] Nirupama Bulusu, Deborah Estrin, and Lewis Girod. Scalable coordination for wireless sensor networks: self-configuring localization systems. *Proc. of the 6th International Symposium on Communication Theory and Applications (ISCTA '01), Ambleside, UK, (July):1–6*, 2001.
- [14] Michael W. Carter, Holly H. Jin, Michael a. Saunders, and Yinyu Ye. SparseLoc: An Adaptive Subproblem Algorithm for Scalable Wireless Sensor Network Localization. *SIAM Journal on Optimization*, 17(4):1102–1128, 2007.
- [15] Peter De Cauwer, Tim Van Overtveldt, Jeroen Doggen, Jerry Bracke, and Maarten Weyn. WSN localization with the Senseless framework. *Telos*, pages 1–8, 2009.
- [16] King Yip Cheng, King Shan Lui, and Vincent Tam. Localization in sensor networks with limited number of anchors and clustered placement. *IEEE Wireless Communications and Networking Conference, WCNC*, pages 4428–4432, 2007.
- [17] Chee Yee Chong and Srikanta P. Kumar. Sensor networks: Evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, 2003.
- [18] International Conference. The use of RF communication devices in ranging applications. pages 1–6, 2014.
- [19] Jose a. Costa, Neal Patwari, and Alfred O. Hero. Distributed Weighted-Multidimensional Scaling for Node Localization in Sensor Networks. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, III(June):1–26, 2005.

- [20] Riccardo Crepaldi, Paolo Casari, Andrea Zanella, and Michele Zorzi. Test-bed implementation and refinement of a range-based localization algorithm for wireless sensor networks. *Proceedings of the 3rd international conference on Mobile technology applications systems Mobility 06*, page 61, 2006.
- [21] J. Desai and U. Tureli. Evaluating Performance of Various Localization Algorithms in Wireless and Sensor Networks. *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2007.
- [22] Inter-node Distances, Lambert Meertens, and Stephen Fitzpatrick. The Distributed Construction of a Global Coordinate System in a Network of Static Computational Nodes from. *Defense*, pages 1–17, 2004.
- [23] F. Franceschini, M. Galetto, D. Maisano, and L. Mastrogiacomo. A review of localization algorithms for distributed wireless sensor networks in manufacturing. *International Journal of Computer Integrated Manufacturing*, 22(7):698–716, 2009.
- [24] Hannes Frey and Daniel Görgen. Geographical Cluster Based Routing in Sensing-Covered Networks. *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW Workshop WWAN)*, pages 885–891, 2005.
- [25] Frank Golatowski, Jan Blumenthal, Matthias Handy, Marc Haase, Hagen Burchardt, and Dirk Timmermann. Softwarearchitektur für Sensornetze. pages 1–9.
- [26] Peter Han and Joo Chong. A Grid-Based Localization Technique for Forest Fire Surveillance in Wireless Sensor Networks :. pages 562–577.
- [27] Tian He, Chengdu Huang, Brian M Blum, John a Stankovic, and Tarek Abdelzaher. Range-free localization schemes for large scale sensor networks. *Proceedings of the 9th annual international conference on Mobile computing and networking MobiCom 03*, page 81, 2003.
- [28] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System architecture directions for networked sensors. *ACM SIGOPS Operating Systems Review*, 34(5):93–104, 2000.
- [29] Lingxuan Hu and David Evans. Localization for mobile sensor networks. *MobiCom*, page 45, 2004.

- [30] Rui Huang, Gergely V. Záruba, and Manfred Huber. Complexity and error propagation of localization using interferometric ranging. *IEEE International Conference on Communications*, pages 3063–3069, 2007.
- [31] Tao Huang, Zhikui Chen, Feng Xia, Cheng Jin, and Liang Li. A Practical Localization Algorithm Based on Wireless Sensor Networks. *2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, pages 50–54, 2010.
- [32] Nadeem Javaid, Ayesha Bibi, and Za Khan. Performance Study of Localization Techniques in Wireless Body Area Sensor Networks. *Trust, Security and Privacy in . . .*, (Id), 2012.
- [33] Miao Jin, Su Xia, Hongyi Wu, and Xianfeng Gu. Scalable and fully distributed localization with mere connectivity. *Proceedings - IEEE INFOCOM*, pages 3164–3172, 2011.
- [34] Anushiya a Kannan, Guoqiang Mao, and Branka Vucetic. Simulated Annealing based Wireless Sensor Network Localization with Flip Ambiguity Mitigation. *Most*, 00(c), 2006.
- [35] Ville a Kaseva, Mikko Kohvakka, Mauri Kuorilehto, Marko Hännikäinen, and Timo D Hämäläinen. A Wireless Sensor Network for RF-Based Indoor Localization. *EURASIP Journal on Advances in Signal Processing*, 2008(1):731835, 2008.
- [36] Ander De Keijzer. Localization in Ad Hoc Sensor Networks. EYES Project. (April), 2003.
- [37] Koen Langendoen and Niels Reijers. Distributed localization in wireless sensor networks: A quantitative comparison. *Computer Networks*, 43(4):499–518, 2003.
- [38] Mark Lin, Amrita Kumar, Xinlin Qing, Shawn J. Beard, Samuel S. Russell, James L. Walker, and Thomas K. Delay. Monitoring the integrity of filament wound structures using built-in sensor networks. pages 222–229, 2003.
- [39] C. Liu and K. Wu. Performance evaluation of range-free localization methods for wireless sensor networks. *PCCC 2005. 24th IEEE International Performance, Computing, and Communications Conference, 2005.*, 2005.

- [40] Chong Liu Chong Liu, Kui Wu Kui Wu, and Tian He Tian He. Sensor localization with Ring Overlapping based on Comparison of Received Signal Strength Indicator. *2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE Cat. No.04EX975)*, pages 516–518, 2004.
- [41] Juan Liu, Ying Zhang, and Feng Zhao. Robust distributed node localization with error management. *Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing - MobiHoc '06*, page 250, 2006.
- [42] Jaime Lloret, Jesus Tomas, Miguel Garcia, and Alejandro Canovas. A hybrid stochastic approach for self-location of wireless sensors in indoor environments. *Sensors*, 9(5):3695–3712, 2009.
- [43] R.C. Luo and S.H. Pan. Mobile user localization in wireless sensor network using grey prediction method. *31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005.*, page 6 pp., 2005.
- [44] Xiaowei Luo, William J. O'Brien, and Christine L. Julien. Comparative evaluation of Received Signal-Strength Index (RSSI) based indoor localization techniques for construction jobsites. *Advanced Engineering Informatics*, 25(2):355–363, 2011.
- [45] David Malan, Thaddeus Fulford-Jones, Matt Welsh, and Steve Moulton. Codeblue: An ad hoc sensor network infrastructure for emergency medical care. . . . *Implantable Body Sensor . . .*, pages 12–14, 2004.
- [46] Guoqiang Mao, Barış Fidan, and Brian D.O. Anderson. Wireless sensor network localization techniques. *Computer Networks*, 51(10):2529–2553, 2007.
- [47] Miklós Maróti, Péter Völgyesi, Sebestyén Dóra, Branislav Kusý, András Nádas, Ákos Lédeczi, György Balogh, and Károly Molnár. Radio interferometric geolocation. *Proceedings of the 3rd international conference on Embedded networked sensor systems - SenSys '05*, page 1, 2005.
- [48] R Masiero and Michele Rossi. Rssi Based Tracking Algorithms For Wireless Sensor Networks: Theoretical Aspects And Performance Evaluation. *Corso di Laurea Specialistica in Ingegneria delle . . .*, 2007.
- [49] Geoffrey J. McLachlan and Kaye E. Basford. Mixture Models. (3):272, 1987.

- [50] Asma Mesmoudi, Mohammed Feham1, and Nabila Labraoui. Wireless Sensor Networks Localization Algorithms: a Comprehensive Survey. *International Journal of Computer Networks & Communications*, 5, 6(6):45–64, 2013.
- [51] D Moore, J Leonard, D Rus, and S Teller. Robust distributed network localization with noisy range measurements. *Proceedings of the 2nd international conference on Embedded networked sensor systems. ACM*, pages 50–61, 2004.
- [52] U. Nazir, N. Shahid, M. a. Arshad, and S. H. Raza. Classification of localization algorithms for wireless sensor network: A survey. *ICOSST 2012 - 2012 International Conference on Open Source Systems and Technologies, Proceedings*, pages 60–64, 2012.
- [53] D. Niculescu and B. Nath. Ad hoc positioning system (APS). *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No.01CH37270)*, 5(C), 2001.
- [54] Dragoş Niculescu and Badri Nath. DV based positioning in ad-hoc networks. *Springer Journal on Telecommunication Systems*, 22(1):267–280, 2003.
- [55] Hakon K. Olafsen. Wireless Sensor network Localisation Strategies. *Signal Processing*, 9120(May):121, 2003.
- [56] Amitangshu Pal. Localization Algorithms in Wireless Sensor Networks: Current Approaches and Future Challenges. *Network Protocols and Algorithms*, 2(1):45–73, 2010.
- [57] Shailaja Patil. MDS and Trilateration Based Localization in Wireless Sensor Network. *Wireless Sensor Network*, 03(06):198–208, 2011.
- [58] N. Patwari, J.N. Ash, S. Kyperountas, a.O. Hero, R.L. Moses, and N.S. Correal. Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal Processing Magazine*, 22(4):54–69, 2005.
- [59] Neal Patwari. Indirect radio interferometric localization via pairwise distances. *Third IEEE Conf on Embedded Sensor*, pages 2–6, 2006.
- [60] Mathias Pelka, Christian Bollmeyer, and Horst Hellbruck. Accurate radio distance estimation by phase measurements with multiple frequencies. In *IPIN 2014 - 2014 International Conference on Indoor Positioning and Indoor Navigation*, number October, pages 142–151, 2015.

- [61] Nissanka B Priyantha, Hari Balakrishnan, Erik Demaine, and Seth Teller. Anchor-Free Distributed Localization in Sensor Networks. *Science*, 19(SenSys):340–341, 2003.
- [62] Csj Rabaey and K Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. *USENIX technical annual . . .*, 2:317–327, 2002.
- [63] Y Raja, S J McKenna, and S Gong. Segmentation and tracking using colour mixture models. *Computer Vision—ACCV’98*, 1997.
- [64] M Rudafshani and S Datta. Localization in wireless sensor networks. *International Symposium on Information Processing in Sensor Networks, 2007. IPSN 2007.*, 00(c):924–928, 2007.
- [65] C Savarese, Jm Rabaey, and J Beutel. Locationing in distributed ad-hoc wireless sensor networks. . . . , 2001. *Proceedings.(ICASSP’01). 2001 IEEE . . .*, 2001.
- [66] a Savvides, A Savvides, H Park, H Park, M B Srivastava, and M B Srivastava. The Bits and Flops of the N-hop Multilateration Primitive For Node Localization Problems. *1st ACM Int.\ Workshop on Wireless Sensor Networks and Application (WSNA 2002)*, pages 112–121, 2002.
- [67] Andreas Savvides. LOCALIZATION IN SENSOR NETWORKS. *Electrical Engineering*.
- [68] Andreas Savvides, Chih-Chieh Han, and Mani Strivastava. Dynamic fine-grained localization in Ad-Hoc networks of sensors. *Proceeding MobiCom ’01 Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179, 2001.
- [69] Yi Shang and Wheeler Ruml. Improved MDS-based localization. In *Proceedings - IEEE INFOCOM*, volume 4, pages 2640–2651, 2004.
- [70] Yi Shang, Wheeler Ruml, Ying Zhang, and Markus P. J. Fromherz. Localization from mere connectivity. *ACM International Symposium on Mobile ad hoc networking & computing*, page 201, 2003.
- [71] Sn Simic and Shankar Sastry. Distributed localization in wireless ad hoc networks. *University of California at Berkeley, Tech. Rep*, pages 1–13, 2002.

- [72] Mani Srivastava, Richard Muntz, and Miodrag Potkonjak. Smart Kindergarten: Sensor-based Wireless Networks for Smart Developmental Problem-solving Environments. *Proceedings of the 7th annual international conference on Mobile computing and networking - MobiCom '01*, pages 132–138, 2001.
- [73] Kuo Feng Ssu, Chia Ho Ou, and Hewijin Christine Jiau. Localization with mobile anchor points in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 54(3):1187–1197, 2005.
- [74] David C. Steere, Antonio Baptista, Dylan McNamee, Calton Pu, and Jonathan Walpole. Research challenges in environmental observation and forecasting systems. *Proceedings of the 6th annual international conference on Mobile computing and networking - MobiCom '00*, pages 292–299, 2000.
- [75] Enrique Stevens-Navarro, Vijayanth Vivekanandan, and Vincent W. S. Wong. Dual and Mixture Monte Carlo Localization Algorithms for Mobile Wireless Sensor Networks. *IEEE Wireless Communications and Networking Conference*, (Mcl):4024–4028, 2007.
- [76] The Stm, The Usart, and Doc Id. Atmel AVR2150: RTB Evaluation Application – User’s Guide. (April):1–15, 2011.
- [77] Robert Szewczyk, Alan Mainwaring, Joseph Polastre, John Anderson, and David Culler. An analysis of a large scale habitat monitoring application. *Proceedings of the 2nd international conference on Embedded networked sensor systems SenSys 04*, 2:214–226, 2004.
- [78] Stefano Tennina, Marco Di Renzo, Fabio Graziosi, and Fortunato Santucci. Statistical analysis of positioning error distribution in wireless sensor networks. *3rd International Conference on Communications and Networking in China, ChinaCom 2008*, pages 619–624, 2008.
- [79] L. Trailovic and L.Y. Pao. Variance estimation and ranking of Gaussian mixture distributions in target tracking applications. *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, 2(5), 2002.
- [80] Andrija S Velimirovic, Maja M Velimirovic, and Milica D Jovanovic. Fuzzy Ring-Overlapping Range-Free (FRORF) Localization Method for Wireless Sensor Networks.

- [81] Document Version. deRFEvaluationKit Positioning 2.4 GHz. pages 1–14, 2013.
- [82] Vijayanth Vivekanandan and Vincent W S Wong. Concentric Anchor-Beacons (CAB) Localization for Wireless Sensor Networks.
- [83] Lei Wang and Qingzheng Xu. GPS-free localization algorithm for wireless sensor networks. *Sensors*, 10(6):5899–5926, 2010.
- [84] Xinwei Wang, Ole Bischoff, Rainer Laur, and Steffen Paul. Localization in Wireless Ad-hoc Sensor Networks using Multilateration with RSSI for Logistic Applications. *Procedia Chemistry*, 1(1):461–464, 2009.
- [85] Kamin Whitehouse, Chris Karlof, and David Culler. A practical evaluation of radio signal strength for ranging-based localization. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(1):41, 2007.
- [86] Heiko Will, Thomas Hillebrandt, Yang Yuan, Zhao Yubin, and Marcel Kyas. The Membership Degree Min-Max localization algorithm. *2012 Ubiquitous Positioning, Indoor Navigation, and Location Based Service, UPINLBS 2012*, 2012.
- [87] F Winkler, E Fischer, E Grass, and P Langendörfer. An Indoor Localization System Based on DTDOA for Different Wireless LAN Systems. *Wpnc*, pages 117–122, 2006.
- [88] Frank Winkler, Erik Fischer, E. Grass, and Gunter Fischer. A 60 GHz OFDM Indoor Localization System Based on DTDOA. *14th IST Mobile & Wireless Communication Summit Dresden*, pages 1–5, 2005.
- [89] Giovanni Zanca, Francesco Zorzi, Andrea Zanella, and Michele Zorzi. Experimental comparison of RSSI-based localization algorithms for indoor wireless sensor networks. *Proceedings of the workshop on real-world wireless sensor networks*, pages 1–5, 2008.

