DGK Deutsche Geodätische Kommission bei der Bayerischen Akademie der Wissenschaften

Reihe C

Dissertationen

Heft Nr. 664

Alexander Born

Algorithmen zur Positionsbestimmung sowie deren Genauigkeiten in drahtlosen Sensornetzwerken

München 2011

Verlag der Bayerischen Akademie der Wissenschaften in Kommission beim Verlag C. H. Beck

DEK Deutsche Geodätische Kommission bei der Bayerischen Akademie der Wissenschaften

Reihe C

Dissertationen

Heft Nr. 664

Algorithmen zur Positionsbestimmung sowie deren Genauigkeiten in drahtlosen Sensornetzwerken

Dissertation zur Erlangung des akademischen Grades Doktor der Ingenieurwissenschaften (Dr.-Ing.) an der Agrar- und Umweltwissenschaftlichen Fakultät der Universität Rostock

vorgelegt von

Dipl.-Ing. Alexander Born

aus Berlin, geb. 05.07.1975 in Templin

München 2011

Verlag der Bayerischen Akademie der Wissenschaften in Kommission beim Verlag C. H. Beck Adresse der Deutschen Geodätischen Kommission:

Ø. **Д р**ак

Deutsche Geodätische Kommission

Alfons-Goppel-Straße 11 • D – 80 539 München Telefon +49 – 89 – 23 031 1113 • Telefax +49 – 89 – 23 031 - 1283/ - 1100 e-mail hornik@dgfi.badw.de • http://www.dgk.badw.de

1. Gutachter: Prof. Dr.-Ing. Ralf Bill, Professur für Geodäsie und Geoinformatik, Universität Rostock

- 2. Gutachter: Prof. Dr.-Ing. Dirk Timmermann, Institut für Angewandte Mikroelektronik und Datentechnik, Universität Rostock
- 3. Gutachter: Prof. (a.D.) Dr.-Ing. Dr. h.c. Lothar Gründig, Institut für Geodäsie und Geoinformationstechnik, Technische Universität Berlin

Tag der Einreichung:04.02.2011Tag der wissenschaftlichen Aussprache:22.07.2011

© 2011 Deutsche Geodätische Kommission, München

Alle Rechte vorbehalten. Ohne Genehmigung der Herausgeber ist es auch nicht gestattet, die Veröffentlichung oder Teile daraus auf photomechanischem Wege (Photokopie, Mikrokopie) zu vervielfältigen

Kurzfassung

Große Mengen preiswerter und einfach auszubringender Sensorknoten, die sich selbst organisieren, drahtlos miteinander kommunizieren, Messungen durchführen und auswerten können, ermöglichen eine großflächige Beobachtung städtischer Gebiete und unzugänglicher Terrains. Eine der wichtigsten Aufgaben für eine raumbezogene Auswertung ist bei der gegebenen zufälligen Ausbringung der Sensorknoten die Bestimmung der Position jedes einzelnen. Ungenaue Eingangsgrößen wie z.B. Distanzmessungen und die sehr eingeschränkten und begrenzten Energie- und Rechenressourcen eines jeden Knotens sowie die hoch miniaturisierte Hardware und geringe Batteriekapazität erfordern die Entwicklung robuster, energieeffizienter und genauer Lokalisierungsalgorithmen. Diese Arbeit gibt einen Überblick über aktuelle Forschungen auf dem Gebiet der Lokalisierung in Geosensornetzwerken und beschreibt Methoden zur Standortbestimmung von Sensorknoten. Ein im Rahmen der DFG-Projekte GeoSens und GeoSens2 entwickelter Lokalisierungsalgorithmus auf Basis geodätischer Ausgleichungsansätze wird hergeleitet. Dieser reduziert deutlich den Berechnungsaufwand und den Energieverbrauch auf den Sensorknoten bei gleichzeitiger signifikanter Erhöhung der Positionsgenauigkeit. Er wird den bekanntesten Algorithmen zur Positionsbestimmung in Geosensornetzwerken gegenübergestellt. Weiterhin erfolgt eine detaillierte Untersuchung der in der Literatur am häufigsten zitierten Lokalisierungsalgorithmen. Hier werden die verschiedensten Einflussfaktoren funktional und stochastisch modelliert und mathematisch abgeleitet sowie in umfangreichen Simulationen analysiert, visualisiert und zu verallgemeinerbaren Aussagen geführt.

Da die Sensorknoten in der Regel mit Sensoren zum Messen unterschiedlichster physikalischer Parameter ausgestattet sind, bietet es sich an, durch Einbeziehen von Sensormesswerten die Positionsschätzung zu verbessern. Dazu wird in dieser Arbeit ein Verfahren hergeleitet. Zudem kann dieser Algorithmus zur Hindernis-, aber auch zur Ausreißerdetektion eingesetzt werden. Aufgrund der limitierten Hardware- und Energieressourcen können auf den Sensorknoten nur eingeschränkt Messsysteme installiert werden, die Beobachtungselemente für eine genaue Lokalisierung liefern. Deshalb werden Distanzen aus Signaleigenschaften abgeleitet, die bei der Kommunikation direkt bestimmt werden können. Diese elektromagnetischen Signale unterliegen jedoch Einflüssen, die die Positionsschätzungen nachhaltig beeinflussen. Daher werden speziell für den Indoor-Bereich Wellenausbreitungsmodelle zur Verbesserung der Positionsschätzung implementiert.

Weiterhin werden Vorschläge entwickelt, wie große Sensornetzfelder effizient partitioniert und als Cluster einer nachträglichen Gesamtnetzausgleichung zugeführt werden können. Abschließend wird auf dieser Grundlage der performanteste Lokalisierungsalgorithmus für drahtlose Sensornetzwerke hergeleitet.

Summary

Large numbers of cheap and easily deployable wireless sensors enable area-wide monitoring of both urban environments and inhospitable terrain. Due to the random deployment of these sensor nodes, one of the key issues is their position determination. Noisy distance measurements and the highly limited resources of each sensor node, due to highly miniaturized hardware and minimal battery capacity, demand the development of robust, energy aware and precise localization algorithms. This thesis presents an overview of current research in the field of localization in geosensor networks and describes methods for the positioning of sensor nodes before describing a fine-grained positioning algorithm developed as part of the DFG-funded GeoSens and GeoSens2 projects. This algorithm uses a least squares adjustment approach which significantly reduces the computational and energy requirements of the sensor nodes while simultaneously giving a considerable increase in accuracy, which is demonstrated in comparison with the most important algorithms existing in literature. Furthermore, the most frequently cited algorithms will be investigated in terms of accuracy as well as their computational and energy requirements. For this, various influencing factors are functionally and stochastically deduced, mathematically analyzed and visualized in various simulations in order to produce generalizable results.

Sensor nodes are usually equipped with additional hardware to monitor physical parameters. The measured information may therefore be used to improve the estimated position of the sensor nodes. A new approach to take this additional information into account has been developed as part of this work. Moreover, this approach can be utilized to detect obstacles in the sensor field and to detect outliers in the observation set. Due to the limited computational and energy resources, it is not appropriate to install additional hardware on the sensor nodes to determine observations for positioning only. Distances are therefore deduced from the characteristics of communication signals such as the received signal strength. These signals are subject to influences which significantly affect the position estimation. To overcome these effects, signal propagation models are implemented to improve the estimated position of the sensor nodes.

Furthermore, proposals are formulated as to how sensor networks may be more efficiently clustered. The accuracy of the estimated position may be improved by adding a network-wide adjustment phase based on these clusters. Based on this approach the most performant algorithm for wireless sensor networks has been developed.

Inhaltsverzeichnis

1	Einl	leitung	9
	1.1	Motivation	1
	1.2	Zielstellung	2
	1.3	Gliederung der Arbeit	3
2	Sen	nsornetzwerke 1	5
	2.1	Definition	5
	2.2	Aufbau der Sensorknoten	6
		2.2.1 Knotenverteilungsdichte	6
		2.2.2 Knotenplattformen und Ressourcen	17
		2.2.3 Energiespeicher und Energieerzeugung	22
	2.3	Energieverbräuche und Einsparpotenziale	23
		2.3.1 Mikrocontrollereinheit	24
		2.3.2 Sensoreinheit	25
		2.3.3 Kommunikationseinheit	25
		2.3.4 Technische Herausforderungen	25
	2.4	Anwendungen für Sensornetzwerke	27
3	Mat	thematische Grundlagen	31
	3.1	Neupunktbestimmung	31
		3.1.1 Methoden zur Neupunktbestimmung	31
		3.1.2 Fehlerhafte Beobachtungen	33
	3.2	Linearisierung nichtlinearer Gleichungssysteme	35
		3.2.1 Taylorreihenentwicklung 3	35
		3.2.2 Linearisierungswerkzeug	35
	3.3	Methode der kleinsten Quadrate	36
	3.4	Lineare Lösungsverfahren	37
		3.4.1 Normalgleichungen	37
		3.4.2 Orthogonalisierungsmethoden 3	38
	3.5	Varianzfortpflanzung	10
	3.6	Netzausgleichung	10
		3.6.1 Das stochastische Modell	11
		3.6.2 Gewichtung von Beobachtungen	12
		3.6.3 Gauß–Markov–Modell	14
		3.6.4 Bedingte Ausgleichung	15
		3.6.5 Apriori–Genauigkeitsanalyse bei der Netzplanung	16
		3.6.6 Eliminierung einzelner Unbekannter	17

4	Entv	wicklungsstand	49
	4.1	Klassifizierung der Algorithmen	49
	4.2	Entfernungsunabhängige Verfahren	51
	4.3	Entfernungsabhängige Verfahren	54
	4.4	Beobachtungselemente zur Lokalisierung	57
		4.4.1 Distanzbestimmung	58
		4.4.2 Winkelmessungen	63
		4.4.3 Sonstige Verfahren	63
	4.5	Fazit	64
5	Eint	beziehung von Sensormesswerten und Signalausbreitungsmodellen	67
	5.1	"Anomaly Correction in Localization"–Algorithmus	67
		5.1.1 Funktionsweise	68
		5.1.2 Das Modell	70
		5.1.3 Ablauf des Algorithmus	70
		5.1.4 Simulation und Ergebnisse	71
		5.1.5 Einschränkungen für ACL	75
	5.2	"extended Anomaly Correction in Localization"–Algorithmus	76
		5.2.1 Den Signalausbreitungskanal beeinflussende Effekte	76
		5.2.2 Dämpfungsmodelle	80
		5.2.3 Grundprinzip des <i>eACL</i>	83
		5.2.4 Simulation	85
		5.2.5 Simulationsergebnisse	86
		5.2.6 Einschränkungen für den <i>eACL</i> –Algorithmus	92
	5.3	Fazit	93
6	Entv	wicklung von Lokalisierungsalgorithmen	95
	6.1	"Resource Aware Localization"–Algorithmus	95
		6.1.1 Lösen der Lokalisierungsaufgabe	96
		6.1.2 Normalgleichungen	96
		6.1.3 Orthogonalisierungsverfahren	97
		6.1.4 Algorithmischer Ablauf von RAL	98
	6.2	Ressourcenverbrauchsanalyse	99
		6.2.1 Rechenaufwand	99
		6.2.2 Kommunikationsaufwand	102
	6.3	Simulation und Ergebnisse	103
		6.3.1 Simulationsaufbau	103
		6.3.2 Untersuchungen zum Energieverbrauch	104
		6.3.3 Vergleich mit etablierten Verfahren	104
	6.4	Anpassung für große Netzwerke	107
		6.4.1 Clusterisierung des Sensorfeldes	107
		6.4.2 Nachträgliche Netzausgleichung	113
		6.4.3 Simulationen	114
		6.4.4 Kommunikationsaufwand	116
	6.5	Fazit	118

7	Gen	nauigkeitsbetrachtungen	121
	7.1	Untersuchte Algorithmen	121
		7.1.1 "Weighted Centroid Localization"–Algorithmus	121
		7.1.2 Standardlokalisierung	122
		7.1.3 "Distributed Least Squares"-Algorithmus	122
		7.1.4 "Resource Aware Localization"–Algorithmus	123
	7.2	Simulationen und Ergebnisse	123
		7.2.1 "Weighted Centroid Localization"–Algorithmus	124
		7.2.2 "Distributed Least Squares"-Algorithmus	135
		7.2.3 "Resource Aware Localization"–Algorithmus	141
		7.2.4 Standardlokalisierung	145
	7.3	Vergleich der Algorithmen	149
	7.4	Performanz der untersuchten Algorithmen	156
	7.5	Fazit	157
8	Zus	ammenfassung und Ausblick	159
	8.1	Zusammenfassung	159
	8.2	Ergebnisse der Arbeit	160
	8.3	Ausblick	163
Lit	eratu	urverzeichnis	165
Ab	bildu	ungs- und Tabellenverzeichnis	179

1 Einleitung

"Es ist nicht das Wissen, sondern das Lernen, nicht das Besitzen, sondern das Erwerben, nicht das Dasein, sondern das Hinkommen, was den größten Genuß gewährt."

> (Carl Friedrich Gauß, dt. Mathematiker, Astronom, Geodät u. Physiker)

Der Gedanke, Sensoren großräumig zur Überwachung und zur Messung physikalischer Parameter einzusetzen, wird seit über 30 Jahren diskutiert. Die rasante Entwicklung auf den Gebieten der drahtlosen Kommunikation und der Mikroelektronik bringt dieses Wissenschaftsgebiet in den Fokus aktueller Forschungsvorhaben. Das führte zur Entwicklung miniaturisierter, kostengünstiger, energiesparender und multifunktionaler Sensoren, die in der Lage sind, ihre Umgebung auf verschiedenste physikalische Parameter hin abtasten zu können. Durch moderne Signalübertragungsverfahren können hunderte solcher Mikrosensoren über kurze Entfernungen miteinander kommunizieren und bilden somit ein leistungsfähiges, verteiltes, drahtloses Sensornetzwerk (DSN (engl. Wireless Sensor Network (WSN)). Eine spezielle Ausprägung eines Sensornetzwerkes ist das Geosensornetzwerk (GSN), welches ein drahtloses Sensornetzwerk mit der Notwendigkeit verknüpft, die Position eines oder mehrerer Knoten in einem übergeordneten Koordinatenreferenzsystem zu bestimmen [SN05, Bil10]. Dieses sowie die Einsatzmöglichkeiten in großen und unzugänglichen Gebieten, in dem sich das Sensornetzwerk selbstständig organisiert und langfristig autark Messungen vornimmt, führen zu zahlreichen denkbaren Anwendungsszenarien, von denen einige detaillierter in dieser Arbeit vorgestellt werden. Nach Heunecke und in Erweiterung durch Bill werden in Tabelle 1.1 Geosensornetzwerke nach ihren Parametern und Unterscheidungsmerkmalen charakterisiert [Heu08, Bil10].

Bei drahtlosen Sensornetzwerken handelt es sich um ein außerordentlich interdisziplinäres Thema. So werden Algorithmen zum Betrieb der Netze in der Informatik entwickelt, Entwicklung der Sensorknotenhardware und der Kommunikationsprotokolle in der Elektro- und Nachrichtentechnik und mit dem Problem der Lokalisierung, Datenauswertung und Visualisierung finden sich klassische Themen für die Geowissenschaften.

Bereits zum Ende des vergangenen Jahrhunderts listete "Business Week" Mikrosensoren zu den 21 wichtigsten Technologien des 21. Jahrhunderts [Bus99]. Das "Massachusetts Institute of Technology" (*MIT*) wählte sie im Jahr 2003 sogar als eine der zehn wichtigsten Technologien, die unsere Welt verändern könnten [Fai03]. Aus einer Studie der "ARC Advisory Group" wurde indes "... für den weltweiten Markt zur Herstellung drahtloser Technologien über die nächsten fünf Jahre ein jährlicher Anstieg von 26 % prognostiziert" [AAG].

Die Geschichte der Sensornetzwerke beginnt beim Militär. Als deren Vorläufer gilt das als "Sound Surveillance System" (*SOSUS*) benannte Geräuschüberwachungssystem, das während des kalten

Krieges vom "Committee for Undersea Warfare" der Vereinigten Staaten von Amerika ins Leben gerufen worden war. Es wurde in den 50er Jahren des vergangenen Jahrhunderts in den Ozeanen in seichten Gewässern installiert, um sowjetische U-Bootbewegungen überwachen zu können. Die *SOSUS*-Stationen bestehen aus mehreren auf dem Meeresgrund angebrachten Hydrophonen. Allerdings funktionierte dieses System nicht drahtlos, vielmehr sind die Sonden mit Unterwasserkabel über weite Strecken mit einer Station an Land verbunden. Die Sensoren sind extrem empfindlich und dadurch in der Lage, akustische Leistungen von weniger als einem Watt über mehrere hundert Kilometer Entfernung zu detektieren [Wikb]. Seine militärische Bedeutung verlor das System mit Ende des kalten Krieges, weshalb die meisten Unterwasserbojen abgeschaltet wurden. Einige noch aktive Sensoren werden heute benutzt, um Walwanderungen zu "belauschen" sowie zur Erdbebendetektion [NC94].

 Tabelle 1.1: Parameter und Unterscheidungsmerkmale von Geosensornetzwerken nach Heunecke / Bill [Heu08, Bil10].

Designparameter	Eigenschaften, Unterscheidungen		
Knotenanzahl k	beliebig; i. d. R. $k = 10$ bis 1.000		
Mobilität der Knoten	statisch, Knoten teilweise in Bewegung (aktiv, passiv)		
Autarkie der Knoten	Lebensdauer von einigen Stunden bis Tagen;		
	robust gegen äußere Einflüsse		
Ausbringung der Knoten	geplant/zufällig (z.B. Abwurf aus Flugzeug)		
	einmalig/ständige Erweiterung des Netzes		
Abdeckung der Daten	vereinzelt/räumlich verdichtet und redundant		
	homogen / heterogen (bezüglich der eingesetzten Sensorik)		
Registrierung	permanent/sporadisch/ereignisgesteuert		
Netztopologie	Infrastrukturell/ad hoc		
(Kommunikation)	sternbasiert / netzbasiert		
Datenkommunikation	unidirektional/bidirektional		
	permanenter Datenfluss/nur auf Anfrage/sporadisch		
Lokalisierung	nicht vorhanden/bei Ausbringung/aus		
	Kommunikationssignalen (knoten-, netzwerkbasiert oder		
	verteilt) / Positionssensor integriert		

Die neue Ära der Sensornetzwerke begann mit den Projekten "Distributed Sensor Networks" und "Sensor Information Technology" um 1980. Diese Projekte werden von der "Defense Advanced Research Projects Agency" (*DARPA*) finanziert, die zur Entwicklung militärischer und wirtschaftlicher Technologien mit militärischen und universitären Einrichtungen zusammenarbeitet. In erster Linie wurden in diesen Arbeiten erste Anforderungen an Sensornetzwerke spezifiziert. Einen sehr umfassenden Überblick über drahtlose Sensornetzwerke geben [SN05] und [MF09]. Bedingt durch die Fortschritte in der Computerhardwareentwicklung in den 1990er Jahren, erlebte die Sensornetzforschung einen Aufschwung. Mittlerweile sind wissenschaftliche Aktivitäten an vielen Einrichtungen in aller Welt anzutreffen. Auch zeigt die seit Jahren stetig ansteigende Anzahl von Konferenzen und Journalen zu Sensornetzwerken die hohe Relevanz dieses Forschungsgebiets. Unter anderem ist dies auf die fortschreitende Miniaturisierung im Hardwarebereich zurückzuführen (hohe Transistordichte auf geringer Schaltkreisfläche, gemeinsame Integration von Kommunikationseinheit und Recheneinheit auf einem einzigen Chip und fortschrittliche Sensormodule mit geringen geometrischen Abmessungen). In absehbarer Zeit ist aus diesem Grunde mit hoch miniaturisierten Sensorknoten ($\sim 1 \, mm^3$) zu rechnen. Als Beispiel dafür soll der "Smart Dust"–Knoten (Abbildung 1.1) herangezogen werden. Dieser wurde unter der Leitung von Kris Pister an der kalifornischen Universität Berkeley entwickelt [KKKP00]. Hier wurde auch das oft benutzte Synonym *Mote* für Sensorknoten geprägt. Die Knoten bestehen aus mehreren Einheiten, die eng miteinander verbunden sind. Die einzelnen Komponenten werden in Kapitel 2 detailliert erklärt. An dieser Stelle soll vorrangig auf den



Abbildung 1.1: "Smart Dust"-Knoten und seine Komponenten [Ber].

Energiespeicher eingegangen werden. Wie aus der Abbildung ersichtlich wird, nimmt die Batterie mit Abstand den größten Platz ein. Ausgehend von einer gewünschten Ausdehnung des Sensorknotens von nur ca. $1 mm^3$ und einer möglichen Kapazität der Batterie von nur 1 J pro mm^3 (bei Solarzellen sinkt dieser Wert auf 1 bis 10 mJ) müssen die verbaute Hardware und die eingesetzten Algorithmen extrem energiesparend ausgelegt bzw. entwickelt werden [Zel06].

1.1 Motivation

Nachdem das Geosensornetzwerk über dem Gebiet von Interesse ausgebracht wurde, verfügen die Sensorknoten noch über keinerlei Positionsinformationen. Allerdings sind aus Sensormessungen gewonnene Informationen zumeist nur sinnvoll, wenn ihre räumliche Position bekannt ist. Eine mögliche Technologie zur Positionsbestimmung sind "Global Navigation Satellite Systems" (*GNSS*) [BCKM04] oder das "Global System for Mobile Communications" (*GSM*) [Gib96]. In Hinblick auf die miniaturisierte Größe der Knoten ist die Anwendung dieser Techniken zur Positionsbestimmung allerdings wenig

sinnvoll und nur auf einigen wenigen denkbar. Auch deren zusätzlicher Energiebedarf sowie die zusätzlichen Kosten verhindern eine uneingeschränkte Installation auf allen Sensorknoten.

Die übliche Methodik besteht nun darin, einige wenige, energie- und ressourcenstärkere Knoten mit existierenden Systemen zur Positionsbestimmung auszurüsten. Diese Knoten werden im Folgenden als *Beacons* bezeichnet. Nachdem diese Beacons ihre Position ermittelt haben, bestimmen die restlichen Knoten ihre Position, z. B. durch Strecken- oder Winkelmessungen, eigenständig.

Eine Kenntnis über die geografische Position der Sensorknoten ist aus folgenden Gründen erforderlich:

- Ermittelte Daten ohne Positionsinformation sind generell unbrauchbar [RBT04b] oder z. B. auf das SOSUS–Projekt bezogen, ein U-Boot wurde vom Sensor detektiert, aber man weiß nicht wo.
- Bekannte Positionen ermöglichen ein energieeffizientes, geografisches Routing [BB05].
- Selbstkonfiguration und Selbstheilung sind Schlüsselmechanismen f
 ür die Robustheit und k
 önnen einfach mittels Positionsinformation realisiert werden.
- Die Administration eines Sensornetzwerkes, bestehend aus tausenden Sensorknoten, erfolgt durch gezieltes Ansteuern geografisch abhängiger Gruppen von Sensorknoten.
- Hindernisse in Sensornetzwerken können ohne Mehraufwand detektiert werden.
- In vielen Anwendungen ist die Position an sich von Interesse (Kapitel 2.4).

1.2 Zielstellung

Aus den genannten Gründen ist also ein Positionsbewusstsein der Sensorknoten nötig. Um mit den verfügbaren, ressourcenlimitierten Möglichkeiten eine Ortsbestimmung in drahtlosen Sensornetzwerken durchführen zu können, existieren in der Literatur zahlreiche Vorschläge zur Lokalisierung, die sich im Allgemeinen in der erreichbaren Genauigkeit und dafür benötigten Ressourcen unterscheiden. Da die bekannten genauen Positionsbestimmungsmethoden in der Regel einen hohen Aufwand an Berechnung und Kommunikation erfordern, wurden zahlreiche Ansätze vorgeschlagen, die eine approximative Positionsbestimmung zulassen. Diese Methoden verbrauchen in der Regel weniger Ressourcen, bedingen aber auch einen relativ großen Schätzfehler.

Ziel dieser Dissertationsschrift ist es, ausgewählte Lokalisierungsalgorithmen hinsichtlich ihrer Genauigkeit zu untersuchen. Dabei wird ein in dieser Arbeit neu formulierter Algorithmus mit einem auf dem selben mathematischen Modell beruhenden verglichen und dem "Fine-Grained Localization"– Algorithmus (Kapitel 4.3 stellt diesen detailliert vor) gegenübergestellt. Ein auf der Schwerpunktbestimmung beruhender Algorithmus wird ebenfalls Genauigkeitstests unterzogen. Weiterhin sollen exakte Algorithmen entwickelt werden, die in einem geringen Positionsfehler resultieren, Ausreißer detektieren und statistische Aussagen zu erreichten Genauigkeiten zulassen. Er soll robust, skalierbar, dabei ressourcensparend und je nach Anwendung flexibel einsetzbar sein.

Den in dieser Arbeit durchgeführten Simulationen wird eine Ausdehnung der Sensornetzwerke von $(100 \, m \times 100 \, m)$ zugrunde gelegt. Die angegebenen Fehlermaße der Positionen und Beobachtungsgrößen werden daraus folgend in Prozent [%] der Sensorfeldgröße angegeben. Ein Fehler von 1 % entspricht somit 1 m.

1.3 Gliederung der Arbeit

Kapitel 2 gibt einen Überblick über Sensorknotenhardware, benennt Anforderungen zur Realisierung von drahtlosen Sensornetzwerken und stellt Anwendungsbeispiele vor.

Kapitel 3 vermittelt die mathematischen Grundlagen zur Positionsbestimmung und diskutiert verschiedene Methoden hinsichtlich ihrer Einsetzbarkeit für energie- und ressourcenlimitierte Sensornetzwerke.

Einen Überblick über den aktuellen Forschungsstand zu Lokalisierungstechniken und möglichen Beobachtungselementen gibt **Kapitel 4**.

Die Knotenplattformen sind in der Regel mit Sensoren zum Messen unterschiedlichster physikalischer Parameter ausgestattet. In **Kapitel 5** wird ein Algorithmus vorgestellt, der durch Einbeziehung von Sensormessungen die Positionsschätzung verbessert und Ausreißer detektiert. Es werden Wellenausbreitungsmodelle implementiert, die eine Hindernisdetektion gewährleisten und der Algorithmus für den Einsatz in Indoorszenarien erweitert.

In **Kapitel 6** wird ein neu entwickelter, feinkörniger Lokalisierungsalgorithmus "Resource Aware Localization" (*RAL*) vorgestellt. Dieser beruht im Wesentlichen auf dem "Distributed Least Squares"–Algorithmus (*DLS*) [BRBT08], verwendet allerdings ein anderes mathematisches Modell. *RAL* reduziert nochmals den Berechnungsaufwand im Sensornetzwerk. Weiterhin werden Möglichkeiten vorgestellt, den Ansatz auf sehr große Netzwerke zu übertragen, indem eine Clusterisierung des Netzes unter verschiedenen Gesichtspunkten vorgenommen werden kann. Eine Nachausgleichung der ermittelten Sensorpositionen wird diskutiert und mit Blick auf die zusätzlichen energetischen Anforderungen analysiert.

Einige ausgewählte der in **Kapitel 4** vorgestellten Algorithmen werden **in Kapitel 7** eingehenden Genauigkeitsanalysen unterzogen. Ein im Zuge dieser Arbeit entwickeltes Werkzeug zur Genauigkeitsuntersuchung wird kurz vorgestellt.

Kapitel 8 fasst die Dissertation zusammen und gibt einen Ausblick auf weiterführende Arbeiten.

2 Sensornetzwerke–Definitionen, Restriktionen und Anwendungen

"The cheapest, fastest and most reliable components of a computer system are those that aren't there."

(Gordon Bell, am. Computeringenieur u. Unternehmer)

Dieses Kapitel gliedert sich in vier Teile. Prinzipiell werden Grundlagen zur Positionsbestimmung in drahtlosen Sensornetzwerken erklärt. Der erste Teil definiert den Begriff der Sensornetzwerke, bevor im zweiten Teil der Aufbau von Sensorknoten beschrieben wird. Im dritten Teil wird auf limitierende Faktoren – mit speziellem Blick auf die Lokalisierungsaufgabe – hingewiesen und die technischen Herausforderungen erläutert. Im vierten Teil werden Anwendungsbeispiele aufgeführt.

2.1 Definition

Als Sensornetzwerk wird üblicherweise ein Zusammenschluss hunderter oder tausender vernetzter Sensorknoten bezeichnet, die miteinander kommunizieren können. Diese Intrakommunikation ist wichtig, da so Netzwerkausdehnungen erreicht werden können, die größer sind als die Sendereichweite einzelner Sensoren.

Da die Sensorknoten mit Seitenlängen von nur wenigen Millimetern bezüglich Arbeitsspeicher, Rechenleistung und vor allem in der Energieversorgung stark eingeschränkt sind, wird nach Möglichkeit die benötigte Sendeleistung minimiert. Deckt ein Sensor mit seiner Kommunikationsreichweite nur seine direkte Nachbarschaft ab, werden weiter entfernte Sensoren weniger in ihrer Kommunikation gestört. Dadurch kann deren Stromverbrauch ebenfalls sinken.

Neben der Nutzung von Mikrocontrollereinheiten (*MCUs*) mit integrierten Stromsparmechanismen werden die Sensoren üblicherweise so programmiert, dass sie nur dann in Betrieb sind (aktiver oder Sende-/Empfangsmodus), wenn eine Datenakquisition oder ein Datenaustausch notwendig ist und den Rest der Zeit inaktiv verbringen (inaktiver oder Schlafmodus). Die Algorithmen, mittels derer ein minimaler Energieverbrauch sichergestellt wird, sind dabei alles andere als trivial [Rei07].

Wie bereits erwähnt, stellen drahtlose Sensornetzwerke sehr hohe Anforderungen an die eingesetzte Hard- und Software. Einige Einsatz- und Ausbringungsarten (z. B. Ausstreuen per Flugzeug) erfordern neben Robustheit eine sehr hohe Knotendichte und damit eine hohe Anzahl von Sensorknoten, da die Wahrscheinlichkeit von Knotenausfällen und zufälligen Lücken (z. B. durch Hindernisse wie Bewuchs, Abdriften durch Wind etc.) in der Netzabdeckung bei solchen Ausbringungsprozessen sehr hoch ist. Zudem sollte die Hardware die Umwelt möglichst nicht belasten, also ökologisch verträglich sein. Im Idealfall sind sie ubiquitär und nicht direkt wahrnehmbar. Solange keine biologisch abbaubaren Sensorkomponenten operationell verfügbar sind, können diese nach deren Einsatz mit Hilfe der bekannten Position wieder entfernt werden.

2.2 Aufbau der Sensorknoten

Dieser Abschnitt beschreibt einige aktuelle Sensorknotenentwicklungen. Da die Ausstattung der Knotenplattformen mit Sensoren stark anwendungsabhängig und flexibel ist, wird zwischen Knotenplattformen als Basis und den darauf einsetzbaren Sensoren unterschieden. Ein schematischer Aufbau eines



Abbildung 2.1: Schematischer Aufbau eines Sensorknotens (nach [Rei07]).

Sensorknotens ist in Abbildung 2.1 dargestellt. Abschnitt 2.2.2 gibt einen detaillierten Überblick über die einzelnen Knotenkomponenten. Wie bereits in Kapitel 1 erwähnt, ist in einem Sensornetzwerk die verfügbare Energie die knappste Ressource. Aus diesem Grunde werden im Besonderen unterschiedliche Energiespeicher und eine mögliche Energieerzeugung auf den Knoten diskutiert sowie die Energieverbräuche und -einsparpotenziale der einzelnen Knotenkomponenten analysiert.

2.2.1 Knotenverteilungsdichte

Die Art und Weise der Ausbringung von Sensorknoten und der Installation eines Sensornetzwerkes hängen stark vom jeweiligen Anwendungsszenario ab. So ist denkbar, dass im Anwendungsfall der "Präzisen Landwirtschaft" eine gleichmäßige Verteilung mit äquidistanten Abständen zwischen den Knoten, im Falle von Vulkanmonitoring hingegen eine Konzentration auf das Zentrum des Netzwerkes wünschenswert ist. Eine weitere Rolle spielt die Auflösung des Sensornetzwerkes, d. h., in welchem räumlichen Maße werden Messwerte benötigt, um das zu beobachtende kontinuierliche Phänomen hinreichend diskret zu repräsentieren. Hinsichtlich der Wirtschaftlichkeit des Netzwerkes müssen diese Belange gegeneinander abgewogen werden, z. B. unter Einbeziehung geostatistischer Methoden.

Allerdings muss auch die maximale Konnektivität des Netzes sichergestellt werden, d.h., es dürfen keine Funklöcher entstehen, wodurch Teile des Netzes vom Gesamtnetz abgeschnitten würden. Zur Beschreibung der Netzabdeckung wird in der Regel die Knotenverteilungsdichte μ_k herangezogen:

$$\mu_k = \frac{k}{\Phi} \tag{2.1}$$

Diese gibt die Anzahl der Knoten k über der Sensornetzwerkfläche Φ an. Hier wird allerdings ausschließlich die Anzahl der Sensorknoten über einem bestimmten Gebiet berücksichtigt. Unbeachtet bleibt die Sendereichweite der Knoten. Zur Sicherstellung einer maximalen Konnektivität ist es aber nötig, eine hohe Anzahl direkter Nachbarn in Kommunikationsreichweite sicherzustellen.

Es soll darauf hingewiesen werden, dass die Ausbringungsproblematik hier nicht weiter betrachtet wird. Die Umsetzung in die Praxis ist äußerst komplex und nicht Gegenstand dieser Arbeit. Um jedoch realistische Ergebnisse aus den durchgeführten Simulationen zu erhalten, wurde die in der Literatur anerkannte und oft verwendete Gleichverteilung der Knoten herangezogen.

2.2.2 Knotenplattformen und Ressourcen

Eine Sensorknotenplattform besteht im Wesentlichen aus folgenden Bauteilen:

- *Mikrocontroller* zum Ausführen diverser Aufgaben (Datenprozessierung, -aggregation, -kompression, Lokalisierung, Steuerung der anderen Knotenkomponenten etc.),
- Externer Speicher zum Zwischenspeichern von Messungen etc.,
- *Transceiver* zur Kommunikation mit anderen Knoten/Beacons, RSS-Messung (Signalempfangsstärkemessung),
- Sensoren zur Detektion und Überwachung physikalischer Umweltparameter und
- Energiespeicher zur Energieversorgung der einzelnen Komponenten.

Im Folgenden werden die einzelnen Bestandteile näher erläutert.

2.2.2.1 Mikrocontroller und Speicher

Der auf dem Knoten installierte Mikrocontroller führt einfache Berechnungsaufgaben durch. Darüber hinaus prozessiert er die gemessenen Daten, d. h., es erfolgen Datenaggregation, -kompression, - dekompression sowie Datenver- und entschlüsselung. Des Weiteren steuert und kontrolliert er die Funktionen der anderen Komponenten auf dem Knoten. Alternativ zum Mikrocontroller wären Mikroprozessoren, Digital-Signal-Prozessoren etc. denkbar. Mikrocontroller sind aber wegen ihres relativ geringen Energieverbrauchs, der einfachen Programmierbarkeit sowie der unterschiedlichsten Peripherie, z. B. Speicherbausteine, Analog–Digital–Wandler, Zeitgeber und externe Schnittstellen die beste Wahl für den Einsatz auf miniaturisierten Sensorknoten. Auch die Möglichkeit den Controller in den Schlafmodus versetzen zu können, während Teile im aktiven Zustand verbleiben, macht ihn attraktiv für die geforderten Anwendungen.

Bekannte Mikrocontroller unterscheiden sich im Allgemeinen in der Busbreite (4, 8 sowie 16 *bit*), der Rechenleistung sowie in Art und Größe des eingesetzten Speichers.

Aus energetischer Sicht ist der auf dem Mikrocontroller verbaute On-Chip-Speicher die relevanteste Speicherart. Da dieser mit nur wenigen Kilobyte in der Regel aber nur sehr knapp bemessen ist, kann eine Erweiterung durch Flash-Speicher erfolgen. Vom bekannten Off-Chip-Random-Access-Memory, kurz RAM, wird aufgrund des erhöhten Energie- und Platzbedarfs nur selten Gebrauch gemacht. Flash-Speicher werden hingegen wegen ihrer relativ geringen Kosten und der großen Speicherkapazität häufig verwendet. Im Unterschied zum On-Chip-Speicher kann allerdings nicht dieselbe Zugriffsgeschwindigkeit erreicht werden. Außerdem wird etwas mehr Energie benötigt. Aber auch an dieser Stelle muss erwähnt werden, dass die Speicheranforderung an den Sensorknoten sehr stark anwendungsabhängig ist.

2.2.2.2 Transceiver

Für den Einsatz in drahtlosen Sensornetzwerken sind vielfältige Kommunikationstechnologien denkbar. Zu nennen wären der Einsatz von Schall- oder elektromagnetischen Wellen und optischer Kommunikation (Laser oder Infrarot). Der Vorteil von Laser ist der relativ geringe Energiebedarf, allerdings wird hier eine direkte Sichtverbindung zwischen den Knoten vorausgesetzt, was in Sensornetzwerken schwierig zu realisieren ist. Ein Beispiel für deren Einsatz ist auf dem "SmartDust"–Knoten zu finden [KKP99]. Wie Laser benötigt Infrarot keine zusätzlichen Antennen, aber auch hier wird eine direkte Sichtverbindung notwendig. Des Weiteren können sich Störeinflüsse durch Umgebungslicht negativ auf die Kommunikation auswirken. Elektromagnetische Wellen, wie z. B. das s. g. "Industrial, Scientific and Medical"-Band (*ISM*) vor allem im Radiofrequenzbereich, sind für den Einsatz in drahtlosen Sensornetzwerken am praktikabelsten. Dabei handelt es sich um lizenzfreie Frequenzbereiche für Hochfrequenzsendegeräte. Der Vorteil liegt in der weltweiten Gültigkeit und des großen Frequenzspektrums. Für bestimmte Szenarien sind Schallwellen denkbar. Die Signaldämpfung z. B. unter Wasser ist so stark, dass der Einsatz elektromagnetischer Wellen unmöglich ist. Schallwellen bieten hier großes Potenzial, wie [OYM⁺04] zeigt.

Transceiver sind kombinierte Sende- und Empfangseinheiten, die in Sensornetzwerken die Betriebszustände Senden, Empfangen, Warten auf Nachrichten und Schlafmodus (engl. Send, Receive, Idle, Sleep) unterstützen. Der Schlafmodus sollte in drahtlosen Sensornetzwerken vorgezogen werden, da er sehr wenig Energie verbraucht. Der Verbrauch im Idle-Modus hingegen ist dem im Empfangsmodus gleichzusetzen. Ein ebenfalls signifikanter Energieverbrauch ist beim Wechsel vom Schlaf- in den Sendemodus zu verzeichnen [XHE01].

2.2.2.3 Sensornetz- und Medienzugriffsprotokolle

Wie im vorigen Abschnitt beschrieben, eignen sich elektromagnetische Wellen besonders gut für den Einsatz in drahtlosen Sensornetzwerken. Aus diesem Grunde wird an dieser Stelle als Teil der Kommunikationsplattform zusätzlich auf die Sensornetz- und Medienzugriffsprotokolle für die Kommunikation eingegangen.

Sensornetzprotokolle

Als möglicher Standard im ISM-Band ist Bluetooth zu erwähnen. Hierbei handelt es sich um einen offenen Industriestandard gemäß *IEEE* 802.15.1 zur Drahtloskommunikation zum Datenaustausch

zwischen Geräten über kurze Distanzen [IEE]. Ursprünglich entwickelt wurde diese Technologie von der schwedischen Firma Ericsson mit dem Hauptzweck des Ersatzes von Kabelverbindungen. Mittlerweile existieren vier Klassen von Bluetooth-Protokollen, die in Tabelle 2.1 wiedergegeben sind. Bluetooth

Klasse	Maximale Leistung		ungefähre Reichweite	Datenrate	
	[<i>mW</i>]	[dBm]	[<i>m</i>]	[<i>Mb/s</i>]	
1	100	20	~ 100	1	
2	2,5	4	~ 10	1 - 3	
3	1	0	~ 1	< 24	
4	0,01 - 0,5	-203	~ 200	1	

Tabelle 2.1: Sendeleistung und -reichweiten für Bluetooth im Freifeld [Blu].

sendet auf dem 2,4 GHz Frequenzband. Um es gegen Störungen und Paketkollisionen zu sichern, wird ein Frequenzsprungverfahren (engl. Frequency Hopping) eingesetzt. Das Frequenzband wird in 79 Frequenzstufen im Abstand von 1 MHz eingeteilt, die bis zu 1.600 mal pro Sekunde gewechselt werden.

In einem Bluetooth-Netzwerk (s. g. Piconet) können bis zu 255 passive Teilnehmer zusammengefasst werden, von denen acht aktiv sind. Handelt es sich bei einem der aktiven Knoten um einen Master, so agieren die verbleibenden sieben als Slaves. Der Master übernimmt die Steuerung der Kommunikation sowie die Slotvergabe an die Slaves. Mehrere Piconetze können zu einem s. g. Scatternet zusammengefasst werden. Hierbei kann ein Master eines Piconets gleichzeitig Slave in einem anderen sein, allerdings ist es nicht möglich, einen Master für mehrere Piconets zu definieren. Somit lassen sich die Masterknoten in einem Bluetooth-Piconet mit den Clusterheads (vgl. Abschnitt 6.4) eines Sensornetzwerkes vergleichen. Eine Möglichkeit des Routings in Piconets gibt [Han06].

Für einen möglichen Einsatz in drahtlosen Sensornetzwerken ist es wichtig, im inaktiven Zustand Energie zu sparen. Bluetooth bietet hierfür drei unterschiedliche Energiesparmodi an:

- HOLD Einsatz zur asynchronen Abwesenheit, d. h., Slave teilt Master seine Abwesenheit für einen gewissen Zeitraum mit. Während dieser Zeit adressiert Master Slave nicht und Slave ignoriert alle ankommenden Masterpakete.
- SNIFF Einsatz zur reduzierten periodischen Aktivität, d. h. Slave hört nur in einem bestimmten Zeitintervall den Funkverkehr ab.
- PARK Einsatz zur Synchronhaltung eines Slaves, welcher nicht aktiv am Datenverkehr teilnimmt.

Trotz der möglichen Energiesparmodi ist der Energiebedarf bei Einsatz von Bluetooth relativ hoch (vgl. Tabelle 2.1). Eine viel versprechende Verbesserung scheint das neue Bluetooth der Klasse 4 ("Bluetooth Low Energy Wireless Technology" als Kernverbesserung der Spezifikation Bluetooth 4.0) zu sein. Aus der Tabelle ist ersichtlich, dass sich die benötigte Leistung und damit der Energiebedarf signifikant bei gleichzeitiger Steigerung der Reichweite zum "klassischen" Bluetooth verringert. Zwar ist auch eine Verringerung des Datendurchsatzes zu verzeichnen. Dieser spielt bei den hier behandelten drahtlosen Sensornetzwerken jedoch eine untergeordnete Rolle, da der Datendurchsatz i. d. R. nur ein paar Kilobyte beträgt und die Latenzzeiten zu vernachlässigen sind. Im Vordergrund steht die Lebensdauer des Netzwerkes und damit die Reduzierung des Energiebedarfs.

Als für drahtlose Sensornetzwerke besonders interessante Verbesserungen gegenüber dem klassischen Bluetooth werden von der "Bluetooth Special Interest Group" (*SIG*) folgende Eigenschaften genannt [SIG]:

- extrem niedriger Energiebedarf für die Betriebsmodi Peak, Average und Idle,
- Einsatz von Knopfzellenbatterien für einen jahrelangen Betrieb,
- geringer Herstellungs- bzw. Anschaffungspreis,
- herstellerunabhängig und
- verbesserte Reichweite.

Als weiterer Kandidat für den Einsatz in Drahtlosnetzwerken empfiehlt sich ZigBee. Wie Bluetooth handelt es sich hier um einen offenen Funknetzstandard. ZigBee ist eine Entwicklung der ZigBee-Allianz, ein Zusammenschluss von mehr als 230 Unternehmen [Zig], und basiert auf dem Industriestandard *IEEE* 802.15.4. Erste ZigBee-Produkte erschienen Anfang 2005 auf dem Markt.

ZigBee sendet auf dem ISM-Band, in Europa auf 868 *MHz* (Short Range Devices *SRD*–Band), 915 *MHz* in den USA und Australien und auf 2, 4 *GHz* in den meisten Ländern weltweit. Im 2, 4 *GHz*-Band ist ZigBee in 16 Kanäle unterteilt, jeder besitzt eine Frequenzbreite von 5 *Mhz*. Es ist in der Regel einfacher und preiswerter als Bluetooth und wird im Allgemeinen integriert in Mikrocontroller und Transmitter verkauft. Es werden ebenfalls die Modi Sleep und Active unterstützt. Der Wechsel zwischen diesen Modi geschieht sehr schnell (15 *ms*), womit die Latenzzeit sehr gering gehalten wird [Leg]. Da ZigBee-Module die meiste Zeit im Sleep-Modus verbringen können, sind sie sehr energiesparend. Das resultiert in einer verlängerten Batterielaufzeit. Mit einer Datenrate von 250 *kbps* im 2, 4 *GHz*-Band, 40 *kbps* für 915 *Mhz* sowie 20 *kbps* bei 868 *MHz* ist ZigBee bei weitem langsamer als Bluetooth, für den Einsatz in drahtlosen Sensornetzwerken ist dieser Durchsatz allerdings hinreichend.

Mit der Version ZigBee 2007 kam ebenfalls eine sogenannte Pro-Variante auf den Markt. Diese unterscheidet sich vor allem in der Sendereichweite. Bei ZigBee 2007 beträgt diese 10 bis 75 m, die Pro-Variante erlaubt sogar Übertragungsweiten von bis zu 1.500 m. Die maximale Leistung liegt generell bei 1 mW (0 dBm), was somit energiesparender als Bluetooth ist. Nur die low-power-Version von Bluetooth erlaubt zukünftig einen geringeren Energieverbrauch. Allerdings ist die Reichweite bei ZigBee um den Faktor sechs höher, wobei erwähnt werden muss, dass alle hier erwähnten Daten stark umgebungsabhängig sind.

Weitere Möglichkeiten, die ebenfalls auf dem *IEEE* 802.15.4 Industriestandard beruhen und hier nicht weiter verfolgt werden sollen, sind z. B. WirelessHART [HARa] und 6LoWPAN [HC08].

Ebenfalls wird mit Wireless-LAN (*WLAN*) und GSM experimentiert [WS]. Allerdings sind diese Technologien aufgrund ihres hohen Energieverbrauchs für den Einsatz in drahtlosen Sensornetzwerken eher wenig praktikabel. Als weitere Beispiele auf dem 2,4 *GHz*–Band operierender Funktechnologien wären das von ABB entwickelte "Wireless Interface for Sensors and Actuators" (*WISA*) sowie MeshScape zu nennen [Vik, Mil]. Große Aufmerksamkeit erfährt auch Ultrabreitband (engl. Ultra-Wideband (*UWB*)), auf das näher in Kapitel 4.4.3 eingegangen wird.

Medienzugriffsprotokolle

Ebenfalls existieren unterschiedliche Techniken, die eine zuverlässige Datenübertragung mit bestimmten, definierten Medienzugriffsprotokollen (engl. Media Access Control, (*MAC*)) ermöglichen. Eine primäre Rolle spielt dabei die Reduzierung des Energieverbrauchs im Kommunikationsmedium, was wiederum im Gegensatz zu den klassischen Drahtlostechnologien (WLAN, GSM) steht, bei denen es im Wesentlichen auf eine maximale Auslastung und faire Verteilung der zur Verfügung stehenden Bandbreite geht.

Wie bereits erwähnt, verfügen die Transceiver im Allgemeinen über die Modi Send, Receive, Idle und Sleep. Um Energie zu sparen, ist es vorzuziehen, die Transceiver vorrangig im Sleep–Modus zu belassen. Zur Aufgabe des MAC–Protokolls gehört zu entscheiden, wann Daten gesendet bzw. empfangen und das Funkmodul ein- bzw. ausgeschaltet werden soll. Dafür kommen in der Literatur zwei Verfahren zum Einsatz: Zufälliger Zugriff mit Trägerprüfung (engl. "Carrier Sense", kurz *CS*) und Zeitmultiplexverfahren (engl. "Time Division Multiple Access", (*TDMA*)) [DEA06].

Der zufällige Zugriff mit Trägerprüfung beruht auf dem s.g. "Low–Power–Listening" (*LPL*) [PHC04]. Hierbei wird der Funkverkehr regelmäßig für kurze Zeit abgehört und geprüft, ob das Medium besetzt ist. Ist dies nicht der Fall, wird das Funkmodul wieder in den Schlafmodus versetzt. Ist das Medium hingegen belegt, bleibt der Funk für den Datenaustausch aktiv. Der Nachteil dieses Verfahrens besteht darin, dass der Sender nicht weiß, wann der Empfänger zuhört. Beispiele für diese Technology sind "Berkeley Media Access Control" (*B–MAC*), bei dem eine Präambel gesendet wird, die länger ist als das Aufwachintervall des Empfängers, X–MAC oder SpeckMAC, bei denen ein langer Strom sich wiederholender Pakete übermittelt wird, oder WiseMAC, bei dem der Sender den Aufwachplan des Empfängers lernt.

Das Zeitmultiplexverfahren ist energetisch günstiger, da hier ein Zeitplan erstellt wird, wann welche Knoten senden und empfangen. Der Nachteil hierbei ist, dass ein größerer Aufwand hinsichtlich des Unterhalts des Zeitplans und der Synchronisation verursacht wird. Allerdings ist der Energiebedarf dieses Koordinationsaufwands geringer als des Verharrens im aktiven Modus und Warten auf Datenpakete [LH05]. Als Beispiele für dieses Verfahren sind "Sensor Media Access Control" (*S–MAC*), "Timeout MAC" (*T–MAC*), "Dynamic Sensor MAC" (*DSMAC*) und "Traffic–Adaptive MAC Protokoll" (*TRAMA*) zu nennen. Eine detaillierte Beschreibung und ein Vergleich der aufgeführten Protokolle findet sich in [DEA06].

2.2.2.4 Sensoren

Sensoren sind Geräte, die physikalische Umweltparameter in einem Gebiet, das überwacht werden soll, detektieren. Tabelle 2.2 listet eine Auswahl erhältlicher Sensortechnik für den Einsatz auf ressourcenlimitierten Knotenplattformen auf. Das detektierte analoge Signal wird mittels eines Analog– Digital–Wandlers digitalisiert und für die weitere Prozessierung an den Mikrocontroller gesendet.

Für den Einsatz in drahtlosen Sensornetzwerken müssen einige Anforderungen erfüllt werden. Aufgrund der geringen Größe der Knotenplattformen sollen die Sensoren extrem klein sein, wenig Energie verbrauchen, autonom und wartungsfrei in einer hohen Sensordichte fehlerfrei arbeiten. Sensoren werden je nach Funktion in drei Klassen eingeteilt:

- Passive, omnidirektionale Sensoren detektieren die Parameter ohne direktes Eingreifen in die Umwelt und ohne spezielle Richtung, z. B. Temperatur.
- Passive, gerichtete Sensoren sind wie die erste Klasse passiv, allerdings mit genau definierter Richtung, in der die Parameter gemessen werden (Druck, Kamera).

Sensor	Umweltparameter
Ambient Light Sensor TSL2550 (digital)	Lichtsensor
Light-to-voltage optical sensor TSL250R (analog)	Lichtsensor
Tiny Serial Digital Thermal Sensor TC74	Temperatursensor
SHT7x Series	Feuchtigkeitssensor
HIH4000 Series	Feuchtigkeitssensor
MS5534a	Drucksensor
CO ² -Aircheck	Gassensor
TGS4161	Gassensor
AMN1 Product Line	Passiver Infrarotsensor

Tabelle 2.2: Beispiele aktueller Sensormodule [ETH].

 Aktive Sensoren greifen zur Messung direkt in die Umwelt ein, indem sie beispielsweise Schalloder Radarwellen aussenden.

Vorzuziehen ist der Einsatz von passiven Sensoren, da sie energiesparender arbeiten. Energie wird vor allem für die Messung der physikalischen Parameter, die Vorkonditionierung und die Filterung der Signale und zur Analog–Digital–Wandlung benötigt.

Zu beobachten ist eine fortschreitende Miniaturisierung von Sensoren, was ihren Einsatz in drahtlosen Sensornetzwerken begünstigt.

2.2.3 Energiespeicher und Energieerzeugung

Bei der Herstellung wird ein besonderes Augenmerk auf die Entwicklung immer kleiner werdender, stromsparender und kostengünstiger Sensoren gelegt, was die Implementierung auf drahtlosen, miniaturisierten Sensorknoten erheblich begünstigt. Die Größe eines Sensorknotens ist so gering wie möglich zu halten. Diese Minimierungsforderung ist jedoch für Energiespeicher sehr schwer zu erfüllen. Das folgende Beispiel soll dieses Problem verdeutlichen: Die Sensorknotenplattform MICA wiegt ohne Batterien 18 g. Als Energieversorgung dienen zwei herkömmliche Mignonzellen mit einem Gesamtgewicht von 54 g (Abbildung 2.2). Damit verursachen die Batterien etwa 75 % des Gesamtgewichts des Knotens und machen zudem einen großen Anteil seiner Abmessungen aus [LWG05].

Auf den Sensorknoten wird Energie für die Messung von Umweltparametern, die Kommunikation und die Datenprozessierung benötigt. Tabelle 2.3 gibt einen Überblick über aktuelle Energiespeicher [Rie09]. Weiterhin besteht die Möglichkeit, Energie aus regenerativen Energiequellen direkt auf dem jeweiligen Sensorknoten zur erzeugen. So können die Sensorknoten mit Solarzellen ausgestattet werden [NHF07], wie es beispielsweise bei der Knotenplattform Golem Dust der Universität Berkeley eingesetzt wird. Die Firma Voltree hingegen entwickelte ein Verfahren zur direkten Energiegewinnung aus Bäumen [KF10]. Denkbare Lösungen wäre eine Energieerzeugung durch Vibration. Mit MEMS–Technologie (Micro-Electro-Mechanical Systems) lässt sich Vibrationsenergie aus der Umgebung gewinnen. Es können Leistungsdichtewerte im Bereich von $100 \,\mu W/cm^3$ bis $1mW/cm^3$ erzeugt werden [EPF06]. Die



Abbildung 2.2: MICA-Mote mit Batterien [Harb].

Energie-	Energie-	Ladewirkungs-	Selbstent-	Lebens-
speicher	dichte	grad	ladung/Monat	dauer
	[Wh/kg]	[%]	[%]	[Jahre/Ladezyklen]
Bleiakku	30-40	60 - 70	5 - 10	4-8/300-600
NiCd	40 - 50	70	10 - 15	15/800 - 1.500
NiMH	60 - 80	70	15 - 20	7 - 10/350 - 500
Lilon	120 - 180	90	1 - 2	10 - 15/500 - 800
LiPo	130 - 150	90	1-2	7 - 10/300 - 500

Tabelle 2.3: Energiespeicher und deren Energiedichte [Rie09].

Knoten sind nur dann aktiv, wenn die Kondensatoren ausreichend Energie für die jeweiligen Aufgaben bereitstellen können. Allerdings erfolgt die Energieerzeugung nur sehr langsam. Es besteht die Gefahr, dass der Sensor wichtige Vorgänge "übersieht". Außerdem muss befürchtet werden, dass die Energie für die kontinuierliche Aufrechterhaltung aller Grundfunktionen des Sensorknotens nicht ausreicht. Die Firma EnOcean erzielt gute Ergebnisse mit batterieloser Funksensorik [EnO].

2.3 Energieverbräuche und Einsparpotenziale

Die Energieverbräuche in Sensornetzwerken werden üblicherweise in zwei Komponenten unterteilt. Zum einen in den Teil, der für die Kommunikation gebraucht wird und zum anderen in den Teil, der für Berechnungen und Messungen aufgewendet werden muss. In Tabelle 2.4 sind beispielhaft ausgesuchte Aktivitäten gegenübergestellt.

Aktivität	Energie-
	verbrauch [mJ]
Temperaturmessung	0.001
(1 Messung)	0,001
Kommunikation (Bluetooth)	0.022
(64 Bit werden gesendet)	0,032
Rechenoperation	0.0018
(1.000 Operationen)	0,0018

Tabelle 2.4: Typische Sensorknotenaktivitäten und deren Energieverbrauch [Jac04].

Eine 0,05g Li-Ion-Polymer-Batterie mit einem Energievorrat von 27 *J* hätte damit bei einer Periode von 1s für die drei Aktivitäten Messen, Kommunizieren und Rechnen eine Laufzeit von nur neun Tagen [Jac04].

Für die Kommunikation wird mehr Energie verbraucht als für die Berechnung und Messung. Levis et al. zeigen, dass der Energieaufwand für das Senden eines 1 *bit*–Paketes über 20 *m* äquivalent mit dem Energiebedarf für 1.000 CPU-Instruktionen ist [LPSC04].

Soll der Energieverbrauch auf den Sensorknoten minimiert und optimiert werden, ist eine Kenntnis über die Einzelverbräuche der Komponenten nötig. Die Optimierung muss dabei nicht nur den einzelnen Sensorknoten bedenken, vielmehr ist es wichtig, den Gesamtenergiehaushalt des Sensornetzes zu berücksichtigen. So wäre es für den Einzelknoten sinnvoll, einzelne Berechnungsaufgaben selbst auszuführen, um Energie durch Kommunikationsvermeidung zu sparen. Aus Netzwerksicht hingegen wäre eine einmalige, zentrale Berechnung auf einem ressourcenstarken Knoten zur Redundanzvermeidung vorzuziehen.

2.3.1 Mikrocontrollereinheit

Der Energieverbrauch der Mikrocontrollereinheit hängt im Wesentlichen von zwei Faktoren ab. Einerseits ist die Leistungsaufnahme proportional zur Versorgungsspannung, die an dem Mikrocontroller anliegt. Laut Necchi et al. verringert sich bei einer gleichzeitigen Reduzierung der Spannung von 1, 2 auf 0, 54 *V* die Leistungsaufnahme von 14 auf 2, 7 *nJ* pro 1.000 Instruktionen [NLPV05]. Andererseits verringert sich der Energieverbrauch, wenn die Taktfrequenz niedriger gewählt wird. Jedoch entstehen bei niedrigeren Taktfrequenzen längere Wartezeiten für die Berechnung. Allerdings ist es durch Dynamic Voltage/Frequency Scaling möglich, die Taktfrequenz an die momentan geforderte Leistung anzupassen [HM09b].

Die Energieressourcen können weiterhin durch die unterschiedlichen Betriebsmodi (Kapitel 2.2.2.1) des Mikrocontrollers geschont werden. Zu beachten ist, dass der Wechsel von einem Modus in einen anderen einen erhöhten Energieverbrauch zur Folge hat. Es ist abzuwägen, ob ein Zustand länger aufrechterhalten wird, oder ob es sich lohnt, zwischendurch in einen vermeintlich energiesparenden Zustand umzuschalten. Mit neueren Technologien wie z. B. energymicro ist es aus energetischer Sicht

möglich, deutlich günstiger zwischen den Modi wechseln zu können, sodass dieser Energiebedarf fast vernachlässigbar ist [ene].

2.3.2 Sensoreinheit

Der Energiebedarf der Sensoreinheit ist stark abhängig von der Anzahl und der Art der eingesetzten Sensoren. Sensoren neuerer Art unterstützen ebenfalls wie Mikrocontroller und Transceiver die unterschiedlichen Betriebsmodi Aktive und Sleep und können auch hier zur Reduzierung des Energieverbrauchs beitragen.

2.3.3 Kommunikationseinheit

Die Kommunikationseinheit verbraucht im Sende- und Empfangsmodus die meiste Energie [RSPS02]. Aus diesem Grunde ist es sinnvoll, durch effiziente Medienzugriffsprotokolle den Energieverbrauch zu reduzieren (Abschnitt 2.2.2.3).

Wie bereits erwähnt, ist ebenfalls Energie nötig, um zwischen den unterschiedlichen Betriebsmodi zu wechseln. Speziell das Aktivieren der Kommunikationseinheit benötigt einen relativ hohen Energiebetrag. Deshalb ist es vorzuziehen, zu versendende Daten erst zu sammeln und sie in einem großen, statt in vielen kleinen Paketen zu übertragen.

2.3.4 Technische Herausforderungen

In Zukunft sollen extrem kleine, preiswerte Sensorknoten in großen Mengen über bestimmten Gebieten ausgebracht werden. Da die Art und Weise der Ausbringung der Sensornetzwerke stark anwendungsabhängig ist (z. B. per Hand, einem Fahr- oder Flugzeug) sowie die Einsatzgebiete stark variieren (z. B. indoor oder outdoor mit Hindernissen wie Bewuchs, Bebauung), sind die technischen Anforderungen sowohl an die Einzelbausteine als auch an das Sensornetzwerk im Ganzen sehr hoch.

Auch sind wirtschaftliche Aspekte nicht zu vernachlässigen. Da die Sensorknoten in hoher Stückzahl ausgebracht werden müssen, um eine feinkörnige Auflösung des zu beobachtenden Phänomens zu gewährleisten, dürfen die Kosten für Anschaffung und Unterhalt nicht zu sehr ins Gewicht fallen, um dieser Technologie Marktchancen zu eröffnen. Dafür steht die Entwicklung miniaturisierter Hardware (< 1 mm³) die bei gleichzeitiger Laufzeitmaximierung (> Jahre) zusätzlich eine hohe Fehlertoleranz aufweisen im Vordergrund. Dabei führt die geringe Größe naturgemäß zu knappen Energieressourcen. Um trotzdem Langlebigkeit zu garantieren, müssen einerseits Hardware (Sensorik, Prozessoren, Kommunikationstechnik) und Software (Betriebssystem, Medienzugriffsprotokolle) konsequent auf Energieeffizienz hin optimiert werden, andererseits müssen auf kleinstem Raum effiziente Techniken zur Energiespeicherung oder -gewinnung untergebracht werden.

Auf Sensornetzebene steht ebenfalls die Energieeffizienz im Vordergrund. So sollen Algorithmen und Protokolle nicht nur den Energiebedarf auf den Einzelknoten minimieren, sondern auch die Energiebelastung auf den Knoten verteilen, um eine maximale Lebenszeit des ganzen Sensornetzes zu garantieren. Die Netze unterliegen, wie bereits erwähnt, einer starken Dynamik. Knoten fallen häufig aus (z. B. wegen erschöpfter Batterien oder ungünstiger Umwelteinflüsse) und neue (auch unterschiedlicher Bauart) kommen hinzu, um ausgefallene zu ersetzen. Trotz dieser Dynamik muss Verlässlichkeit sichergestellt sein, d. h., auch bei großen Ausfällen müssen noch bestimmte Funktionen garantiert werden.

Die Netzwerke müssen auch aufgrund der anvisierten Einsatzgebiete ohne Infrastruktur auskommen, d. h., die in hoher Stückzahl vorliegenden Sensorknoten müssen sich spontan vernetzen. Dabei fällt auch der Skalierbarkeit eine wichtige Rolle zu. Die erwähnten Eigenschaften eines Sensornetzes (große Ausdehnung, geringe Größe der Sensorknoten, je nach Anwendung unzugängliches Terrain) kann menschliches Eingreifen und Konfigurieren verhindern. Aus diesem Grund müssen Sensornetze selbstkonfigurierend und selbstadministrierend sein. In sicherheitsrelevanten Anwendungsbereichen muss der Schutz des Netzwerkes vor störenden Einflüssen oder bösartigen Angriffen garantiert werden. Neben den traditionellen Sicherheitstechnologien kommen auch neue Aufgaben hinzu, wie z. B. das Verhindern des Einschleusens fremder Knoten oder das Fluten des Netzes mit physischen Stimuli.

Die Sensoren auf den Knoten messen in der Regel physikalische Parameter. Da es sich bei den Ergebnissen um reine Messwerte handelt, kommt der Signalverarbeitung eine große Bedeutung zu. Anstelle unter hohem Energieverbrauch die reinen Messwerte ins Netz zu verschicken, sollten die Daten lokal vorverarbeitet werden, um relevante Ereignisse zu extrahieren sowie geeignet zu aggregieren. Dabei wird aus einer großen Menge von Messwerten mit niedrigem Informationslevel (z. B. eine Menge von Sichtungen eines Objektes durch verschiedene Sensorknoten) eine kleine Datenmenge mit hohem Informationslevel (z. B. die Durchschnittsgeschwindigkeit und Bewegungsrichtung des beobachteten Objektes) generiert. Grob betrachtet handelt es sich hierbei schon um eine gewisse Form der Datenkompression und -fusion [NLF07].

Aus dem Vorangegangenen lassen sich für drahtlose Sensornetzwerke folgende technische Herausforderungen zusammenfassen:

- Selbstkonfiguration und -administration: automatische Anpassung an die Umgebung und Initialisierung (Ermittlung der optimalen Kommunikationsreichweite aller Nachbarknoten etc.),
- Selbstheilung: das Sensornetzwerk reagiert auf Knotenausfälle und garantiert eine möglichst lange Lebenszeit,
- Mobilität des Netzwerkes: Bewegungen des Netzwerkes werden registriert und ggf. darauf reagiert, z. B. mit Aufgabenneuverteilung,
- Aufgabenverteilung: sinnvolle Verteilung komplexerer Berechnungen und somit der Energiebelastung im Netz,
- Medienzugriffsmethoden: Steuerung des Zugriffs auf das gemeinsame Kommunikationsmedium (vgl. Abschnitt 2.2.2.3),
- Routingmethoden: sinnvolle Paketweiterleitungsmethoden im Netzwerk,
- Datenaggregation und -kompression: Vorverarbeiten, Filtern und Komprimieren der Messwerte, Reduzieren des hohen Datenverkehrs,
- Lokalisierungsverfahren: Um die an das Sensornetz gestellten Aufgaben erfüllen zu können, ist eine frühzeitige Positionsbestimmung und damit ein Lokalisationsbewusstsein eines jeden Sensorknotens hilfreich. Diese Lokalisierungsmethoden sollen möglichst genaue Positionen liefern sowie ressourcensparend sein [BRBT08].

2.4 Anwendungen für Sensornetzwerke

Drahtlose Sensornetzwerke finden in vielen Bereichen Anwendung. Typischerweise handelt es sich dabei um Überwachungs-, Verfolgungs- oder Kontrollaufgaben. In den Geowissenschaften wären Sensornetze vielseitig einsetzbar und könnten in der Ingenieurgeologie oder in der Seismik Anwendung finden, um z. B. Hangrutschungen und mikroseismische Signale zu detektieren. In der marinen Vermessung oder in der Hydrologie können Geosensornetzwerke zur Verfolgung von Sedimenttransporten oder in Flusssystemen zur Temperaturmessung genutzt werden. Auch ein Einsatz in der Bodenkunde ist denkbar und bereits in Anwendung, um Änderungen der Bodenfeuchte oder die elektrische Leitfähigkeit des Bodens zu messen. Darüber hinaus kommen Geosensornetzwerke überall dort zum Einsatz, wo es um dynamische Prozesse geht, die mit Hilfe von Sensoren erfasst und überwacht werden sollen, bzw. wo durch den Einsatz von mobilen Sensoren eine immer präzisere Abtastung und Erfassung der Umgebung erreicht werden kann. Wichtige weitere Anwendungsfelder liegen somit in der Verkehrsbeobachtung und Navigation sowie in der Bauwerksüberwachung.

Eine große Anzahl drahtloser Sensornetzwerke werden für den Einsatz für das Umweltmonitoring getestet. Durch die prototypische Natur sind viele von ihnen nur für eine kurze Lebenszeit konzipiert [HM06]. Beispiele mit längerer Einsatzdauer sind z. B. seit dem Jahr 2006 die Permafrostüberwachung in den Schweizer Alpen [The] oder das seit 2004 andauernde Projekt "Glacsweb" zur Gletscherüberwachung [Uni, MHO09].

Im Fall der Präzisionslandwirtschaft (engl. "Precision Farming") [NBB07] werden Sensorknoten über Agrarflächen ausgebracht, die in der Lage sind, den Boden auf unterschiedliche chemische und biologische Substanzen, pH-Werte und Bodenfeuchten zu untersuchen. Aus deren Daten ließen sich z. B. ein eventueller Pflanzenschutz-, Dünge- oder Bewässerungsbedarf ermitteln. Finanziell würde bei dieser Anwendung durch den sparsamen Einsatz von Dünge- und Pflanzenschutzmitteln der Landwirt profitieren sowie die Umwelt durch geringere Pflanzenschutzmittelbelastung. In dem Projekt "SoilNet" des Instituts für Chemie und Dynamik der Geosphäre des Forschungszentrum Jülich wurde ein Sensornetz zur Messung von Bodenfeuchten entwickelt, welches komplett unterirdisch arbeitet und auf ZigBee-Technologie (vgl. Abschnitt 2.2.2.2) beruht [BHM⁺09].

Aber auch das Erkunden von Tierverhalten in natürlichen oder auch künstlichen Umgebungen ist ein interessanter Anwendungsfall. Bei dem "Great-Duck-Island–Project" wurde das Brutverhalten von Schwalben durch auf der Insel installierte Sensoren untersucht [MPS⁺02]. Dies ist eines der bekanntesten frühen Beispiele von Geosensornetzwerken. Im "Skomer Island–Project" wird eine Kombination aus GPS und drahtlosen Sensornetzwerken verwendet, um das Fütterungs- und Futtersuchverhalten von Seevögeln zu überwachen [GMF⁺07]. In der hochgenauen Nutztierhaltung (engl. "Precision Livestock Farming") sind Geosensornetzwerke zur Beobachtung des Verhaltens von Nutztieren von Interesse. Neben den zurückgelegten Wegen oder Fresszeiten werden in [SCV⁺06] auch Reaktionen auf Aktoren (Vibration und Schall) einbezogen. [RT05], [RHT05], [DVA⁺04], [YOD⁺02] beschreiben das so genannte "Large Scale Ocean Monitoring", bei dem Geosensoren Ozeanschichten großflächig auf verschiedenste Parameter, wie z. B. Strömungsgeschwindigkeit, Salinität, Temperatur etc. hin untersuchen. Bisher ist das nur an ausgewählten Punkten möglich.

Die weltweite Zunahme von Naturkatastrophen und der damit verbundene Verlust an Menschenleben und materiellen Werten erfordert den Einsatz zuverlässiger Frühwarnsysteme. In dem BMBF-Verbundprojekt "Sensor based Landslide Early Warning System" (*SLEWS*) der Professur für Geodäsie

und Geoinformatik der Universität Rostock ist die Entwicklung von geeigneten Methoden und Technologien der Frühwarnung bei Massenbewegungen das Ziel [AAA+07]. Die zu entwickelnde Geodiensteinfrastruktur umfasst Sensoren, Geodatenbestände, Informations- und Kommunikationstechnologien sowie Methoden und Modelle zur Ableitung von stabilitätsrelevanten Parametern. Mit verschiedenen Sensoren sollen zuverlässige Echtzeitinformationen gewonnen werden, aus denen wiederum nach der Aufbereitung mit Hilfe geeigneter Verfahren und Algorithmen Warnungen oder Prognosen (Frühwarnungen) abzuleiten sind. Insofern stellt SLEWS ein ideales Anwendungsbeispiel für Geosensornetzwerke dar. Auch in Überflutungsgebieten sind Geosensoren vielseitig einsetzbar. Zukünftig sind Geosensornetzwerke denkbar, die in künstliche Hochwasserschutzdeiche eingebracht werden und somit eine flächendeckende, automatisierte Überwachung ermöglichen [Sch06]. Dazu können dem Füllmaterial für die Sandsäcke Sensoren beigemengt und später die Feuchte innerhalb des Sandsackes registriert werden. Steigt diese signifikant an, lösen die Sensoren frühzeitig Alarm aus und die Rettungsmannschaften sind in der Lage, zeitnah Verstärkungen an der undichten Stelle vorzunehmen. Katastrophale Folgen wie z. B. bei dem Elbehochwasser 2002 wären durch diese Technologie mit hoher Wahrscheinlichkeit in Grenzen gehalten worden. Ebenfalls ist es denkbar, chemisch oder biologisch kontaminierte Gebiete zu erkunden, ohne Rettungsmannschaften unnötigen Gefahren auszusetzen. Die Detektion besonders gefährlicher Sektoren können Geosensornetzwerke, die von Flugzeugen ausgebracht werden, übernehmen. Neben diesen Beispielen gibt es noch viele Anwendungsmöglichkeiten im Katastrophenschutz wie z. B. Frühwarnungen bei Waldbränden, aktiven Vulkanen [WALW⁺06] oder zur Überwachung der Gebäudestatik [CFP+06].

Im industriellen Sektor bietet sich ein breites Einsatzgebiet für Sensornetzwerke. In den USA werden Projekte für das Wasser- und Abwassermanagement als Teil des "American Recovery and Reinvestment Act" (*ARRA*) gefördert. Zum Monitoring des Füllstandes in Wasserwerken sind Sensornetzwerke sinnvoll. So können zu Spitzenverbrauchszeiten, wenn der Pegel und damit der Druck in den Reservoirs fällt sowie der Abfluss ansteigt, Pumpen anspringen, um die Wasserversorgung sicher zu stellen. Zur Detektion des Abflusses und des Pegelstandes können auch hier drahtlose Sensornetzwerke zum Einsatz kommen. Sensornetzwerke können ebenfalls auf Mülldeponien Anwendung finden. Zum einen, um die Wasserspiegel in allen Brunnen auf dem Deponiegelände zu messen und kontinuierlich zu beobachten. Zum anderen, um die Müllsickerwasserspeicherung und -entwässerung zu überwachen. Bislang werden Pumpen zur Sickerwasserentfernung manuell überwacht. Sensornetze könnten diese Daten sammeln, an ein zentrales Kontrollsystem übermitteln, welches diese Daten auswertet. Dieses System würde die Pumpleistungen überwachen und Kontrollzyklen für das Servicepersonal bereitstellen [Ban].

Weiterhin ist denkbar, in großen Werkshallen Objekte zu orten oder durch Produktions- und Lagerstraßen zu verfolgen. Ein Beispiel stellt das "Center for Life Science Automation" in Rostock-Warnemünde dar. Hier wurden ganze Räume mit Sensoren ausgestattet, welche zahlreiche Parameter messen. Durch diese Sensoren wären auch Personen oder Mischproben auf Transportwagen zu orten. In Laborräumen kann eine gleich bleibende Atmosphäre (z. B. konstante Temperatur, Giftdetektion in der Luft, etc.) sichergestellt werden [RBT04a], [GRB⁺06].

Auf dem Gebiet des so genannten "Health Monitoring" ist mit der Überwachung der Vitalfunktion älterer Menschen eine Erhöhung ihrer Lebensqualität denkbar. Es können Sensornetzwerke in Wohnungen derart platziert werden, dass der Hausarzt jederzeit über den jeweiligen aktuellen Aufenthaltsort, Vitalund Umgebungszustand informiert ist [JEZ⁺05, RB06, MLL⁺08, MZG⁺09]. Ändert sich die Position der betroffenen Person in einer ungewöhnlichen Art und Weise (z. B. der Oberkörper wird 30 *cm* über dem Boden detektiert) nicht innerhalb einer gewissen Zeit, ist eine automatische Alarmierung der jeweiligen Rettungsstelle möglich. Darüber hinaus können Messungen, z. B. eines Langzeit-EKGs dahingehend verbessert werden, dass es andere relevante Daten aufnimmt. Handelt es sich beim starken Herzklopfen um Anzeichen von Herzrythmusstörungen oder ist es auf Anstrengung zurückzuführen? So können mit verschiedenen drahtlosen Sensoren auch der Blutdruck, die Temperatur oder die Umgebungsbedingungen gemessen werden [Yan06].

Das Militär hat ebenfalls ein erhebliches Interesse an der Entwicklung von Sensornetzwerken. In den USA finanziert die DARPA zahlreiche solcher Projekte. So könnten eine große Anzahl Sensorknoten über einem Gebiet ausgestreut werden und feindliche Bewegungen (z. B. Hitzeentwicklung, Veränderungen des Drucks und des elektromagnetischen Feldes (EM-Feld), Geräusch-, Licht- und Vibrationsdetektion) entdecken und überwachen [Sam01].

3 Mathematische Grundlagen zur Positionsbestimmung

"Die Mathematik ist das Alphabet, mit dessen Hilfe Gott das Universum beschrieben hat."

(Galileo Galilei, ital. Mathematiker, Physiker u. Astronom)

Dieses Kapitel erläutert die mathematischen Grundlagen zur Bestimmung eines Neupunktes im \Re^2 (eine Erweiterung auf den \Re^3 ändert nichts an der prinzipiellen Vorgehensweise). Beginnend mit Abschnitt 3.1 wird das mathematische Modell für eine einfache Neupunktbestimmung hergeleitet.

In der Realität kommen keine fehlerfreien Messungen vor. Aus diesem Grund wird das Modell um fehlerbehaftete Eingangswerte erweitert. Wie diese fehlerbehafteten Eingangsdaten für die Simulationen erzeugt werden, wird detaillierter in Kapitel 7.2 beschrieben.

Da die gewählten Lokalisierungsmethoden auf linearen Gleichungen beruhen, die mathematischen Modelle jedoch auf nichtlineare Zusammenhänge zurückgeführt werden, behandelt Abschnitt 3.2 Möglichkeiten zur Linearisierung der nichtlinearen funktionalen Modelle. Diese linearen Gleichungssysteme sollen danach gelöst werden. Methoden dafür beschreibt Abschnitt 3.4. Verschiedene Beobachtungen und deren Fehler haben unterschiedliche Auswirkungen auf das Ergebnis der Positionsbestimmung. Grundlagen zu deren Berücksichtigung werden in Abschnitt 3.5 behandelt.

Wie in Kapitel 2 beschrieben, bestehen drahtlose Sensornetzwerke aus vielen Sensorknoten. Mit dieser großen Zahl an Knoten sind ebenso viele Distanz- oder Winkelmessungen zueinander möglich, die für die Lokalisierung herangezogen werden können (eine detaillierte Beschreibung wie Distanzen und Winkel in drahtlosen Sensornetzwerken gemessen werden, erfolgt in Kapitel 4.4). Das führt zu überbestimmten Gleichungssystemen, die die Genauigkeit der endgültigen Positionsschätzung des unbekannten Sensorknotens gegenüber einfachen Berechnungen aus Minimalkonfigurationen erhöhen können. Die darauf aufsetzende Ausgleichungsrechnung wird in Abschnitt 3.6 beschrieben. Darüber hinaus sei auf die zahlreich vorhandene Literatur zu überbestimmten Ausgleichungsverfahren verwiesen [Gro69, Bjö96, WG97, Nie08, JMS05].

3.1 Mathematische Grundlagen zur Neupunktbestimmung

3.1.1 Methoden zur Neupunktbestimmung

Für die feinkörnige oder exakte Neupunktbestimmung (vgl. Kapitel 4.1) in drahtlosen Sensornetzwerken kommen im Wesentlichen drei Verfahren infrage. Das ist zum einen die auf Winkelmessungen beruhende Triangulation, zum anderen die auf Streckenmessungen beruhende Trilateration oder eine Kombinati-

on aus beiden. Da die Sensorknoten nur über limitierte Energie- und Hardwareressourcen verfügen, ist die dritte Herangehensweise nicht empfehlenswert, da zusätzliche Hardware für die unterschiedlichen Beobachtungsgrößen installiert werden müsste. Auch sind winkelbasierte Lokalisierungsmethoden aus demselben Grund nicht ohne weiteres möglich. Jedoch sollen diese Verfahren an dieser Stelle kurz erläutert werden. In Abbildung 3.1 sind zwei Lokalisierungsverfahren in Minimalkonfiguration auf Basis von Winkelmessungen illustriert. Der Vorwärtsschnitt ist in Abbildung 3.1 (a) dargestellt. Auf



Abbildung 3.1: (a) Vorwärtseinschneiden und (b) Rückwärtseinschneiden durch Messen der Winkel α_1 und α_2 .

den bekannten Punkten $P_1(x_1; y_1)$ und $P_2(x_2; y_2)$ werden die Winkel α_1 und α_2 zu dem Unbekannten $P_N(x_N; y_N)$ bestimmt. Die Koordinatenbestimmung erfolgt durch die Berechnung des Schnittpunktes der beiden Strahlen, die vom jeweiligen Festpunkt P_1 , P_2 zum Neupunkt P_N verlaufen. Aus den gemessenen Winkeln lässt sich über die Winkelsumme der Winkel $P_1P_NP_2 = \alpha_3 = 180^\circ - (\alpha_1 + \alpha_2)$ berechnen. Der Sinussatz liefert die Strecken $s_{P_1P_N} = \overline{P_1P_N}$ und $s_{P_2P_N} = \overline{P_2P_N}$:

$$s_{P_1P_N} = \frac{s_{P_1P_2}\sin(\alpha_2)}{\sin(\alpha_3)}, \ s_{P_2P_N} = \frac{s_{P_1P_2}\sin(\alpha_1)}{\sin(\alpha_3)}.$$
(3.1)

Die Richtungswinkel *t* zum Neupunkt berechnen sich zu $t_{P_1}^{P_N} = t_{P_1}^{P_2} - \alpha_1$ und $t_{P_2}^{P_N} = t_{P_2}^{P_1} + \alpha_2$. Mit diesen Größen ist die gesuchte Position $P_N(x_N; y_N)$ durch polares Anhängen zu berechnen [BR09]:

$$x_N = x_{P_1} + s_{P_1 P_N} \sin(t_{P_1}^{P_N}), \ y_N = y_{P_1} + s_{P_1 P_N} \cos(t_{P_1}^{P_N})$$
(3.2)

oder

$$x_N = x_{P_2} + s_{P_2 P_N} \sin(t_{P_2}^{P_N}), \ y_N = y_{P_2} + s_{P_2 P_N} \cos(t_{P_2}^{P_N}).$$
 (3.3)

Der Vorwärtsschnitt kann in Sensornetzwerken nur durchgeführt werden, nachdem die Sensorknoten gesendet haben. Weiterhin bestimmen die Sensorknoten ihre Position nicht selbst, d. h., die berechneten Koordinaten müssen in das Netzwerk zurück geflutet werden, was zusätzlichen Kommunikationsaufwand erfordert.

Der Vorteil des Rückwärtseinschneidens (Abbildung 3.1 (b)) dagegen ist, dass die Lokalisierung auf den Sensorknoten durchgeführt wird und ein Senden auf Seite der Sensorknoten nicht erforderlich ist. Der

ebene Rückwärtsschnitt (Abbildung 3.1 (b)) ist eine trigonometrische Methode zur Landvermessung. Zum Beispiel werden die Koordinaten eines Neupunktes P_N durch drei Punkte P_1 , P_2 und P_3 mit bekannten Koordinaten bestimmt, wenn die Horizontalwinkel $P_1P_NP_2 = \alpha_2$ und $P_1P_NP_3 = \alpha_1$ bekannt sind. Diese Winkel können aus einer Richtungsmessung im Neupunkt P_N zu den Punkten P_1 , P_2 und P_3 berechnet werden. Durch Schneiden der beiden Peripheriewinkelkreise mit den Winkeln α_2 (über der Sehne P_1P_2) und α_1 (über der Sehne P_1P_3) ergibt sich die Lösung. In der Literatur existieren allerdings zahlreiche weitere Methoden zur Lösung des Rückwärtsschnittes [FLB09].

Basieren die Beobachtungen hingegen auf Streckenmessungen, kann mittels Trilateration die Position des Unbekannten berechnet werden. Abbildung 3.2 verdeutlicht diesen Sachverhalt. Die Berechnung



Abbildung 3.2: Bogenschnittverfahren.

erfolgt durch Schnittpunktbestimmung der Kreise, die durch die Distanzmessungen vom Neupunkt zu den Festpunkten definiert werden. Liegen alle Punkte nicht auf einer Linie, schneiden sich die Kreise in zwei unterschiedlichen Punkten. Die Lokalisierung ist somit nicht eindeutig. Aus diesem Grund müssen im \Re^2 drei Streckenmessungen vorliegen, um ein eindeutiges Ergebnis zu erhalten (Abbildung 3.2). Aus den Streckenmessungen lässt sich ein Gleichungssystem unter Verwendung der euklidischen Distanz ableiten:

$$s_{NP_i} = \sqrt{(x_i - x_N)^2 + (y_i - y_N)^2}.$$
 (3.4)

Durch Auflösen des Gleichungssystems 3.4 ist der gesuchte Neupunkt $P_N(x_N; y_N)$ eindeutig bestimmbar. Eine detaillierte Betrachtung folgt im weiteren Verlauf dieses Kapitels.

3.1.2 Fehlerhafte Beobachtungen

Die o. g. Methoden liefern eine hoch genaue Lokalisierung, wenn fehlerfreie Beobachtungen vorausgesetzt werden. In der Realität sind diese Beobachtung jedoch fehlerbehaftet. Da die Messungen in einem zeitlich begrenzten Rahmen stattfinden, repräsentieren sie nur eine Stichprobe aus einer fiktiven Grundgesamtheit. In der Fehlertheorie werden drei Obergruppen möglicher Fehler unterschieden [Tay96]:

- Grobe Fehler stellen das Messverfahren selbst in Frage, wie z. B. ein falsches Ablesen der Anzeige. Sie sollten vermieden werden, können aber durch sogenannte Ausreißertests auch statistisch nachgewiesen werden.
- Zu den systematischen Fehlern gehören beispielsweise die Messinstrumentenfehler, die bei jeder Messung gleichermaßen auftreten. Sie werden in der Regel durch Justierung und Kalibrierung der Messinstrumente vorab eliminiert. Eine Variation der Messanordnung kann den systematischen Fehler ebenfalls ausgleichen.
- Zufällige Fehler entstehen durch unkontrollierbare Änderungen der äußeren Verhältnisse, wie Temperaturschwankungen oder ein endliches Auflösungsvermögen der Messeinrichtung. Sie können im Sinne einer Ausgleichung reduziert werden.

Mit allen empirisch gewonnenen Messgrößen ist es möglich, ein bestmögliches Ergebnis zu erhalten. Liegen dabei mehr Messungen als Unbekannte vor, wird von einer überbestimmten Lokalisierung gesprochen. Auf den Fall der Sensornetzwerke übertragen heißt das, es werden mehr als zwei Winkel oder mehr als drei Strecken zwischen dem zu positionierenden Sensorknoten und den Beacons und/oder den anderen Sensorknoten gemessen. Diese Überbestimmung kann genutzt werden, um eine statistisch optimale Schätzung in einer Ausgleichung durchzuführen. Für das Problem der Neupunktbestimmung auf Grundlage von Streckenbeobachtungen (Trilateration) lässt sich folgendes Modell formulieren: Eine Menge von Beobachtungen formt den Vektor b. Ein Vektor aus den gesuchten Neupunktkoordinaten bildet Vektor x, den es nach ausgesuchten Optimierungszielen zu bestimmen gilt. Besteht eine funktionale Beziehung zwischen den Unbekannten und den Beobachtungen, ist die Menge der Messungen auf die Menge der Unbekannten abzubilden. Den dabei auftretenden Fehler gilt es zu minimieren. Im weiteren Verlauf der Arbeit werden exakte/wahre Koordinaten $\tilde{P}_i(\tilde{x}_i; \tilde{y}_i)$, exakte/wahre Distanzen \tilde{d}_i sowie geschätzte Koordinaten $\hat{P}_i(\hat{x}_i; \hat{y}_i)$ und Beobachtungen \hat{d}_i unterschieden.

Bei dieser s.g. Mulitlateration, d.h. es liegen mehr als die drei zur Trilateration benötigten Streckenbeobachtungen vor, basiert das funktionale Modell auf Euklidischen Distanzen $\hat{d}_1 \dots \hat{d}_m$ (Gleichung 3.5), wobei es sich bei n > 3 Festpunkten mit den Beaconkoordinaten $\tilde{B}(\tilde{x}_1, \tilde{y}_1) \dots \tilde{B}(\tilde{x}_n, \tilde{y}_n)$ um ein überbestimmtes Gleichungssystem mit n Gleichungen und u = 2 Unbekannten \hat{x}_N und \hat{y}_N handelt [Bul00].

$$(\tilde{x}_{1} - \hat{x}_{N})^{2} + (\tilde{y}_{1} - \hat{y}_{N})^{2} = \hat{d}_{1}^{2}$$

$$\vdots$$

$$(\tilde{x}_{n} - \hat{x}_{N})^{2} + (\tilde{y}_{n} - \hat{y}_{N})^{2} = \hat{d}_{n}^{2}$$
(3.5)

In dieser Arbeit wird nur das zweidimensionale Lokalisierungsproblem betrachtet. Wie bereits erwähnt, ist eine Überführung in drei Dimensionen ohne Weiteres möglich. Der prinzipielle Ablauf bleibt derselbe.

Zur Lösung dieses überbestimmten Gleichungssystems kommen mehrere Ansätze in Betracht. Zum einen können nichtlineare Optimierungen, wie beispielsweise das Simplex–Verfahren [LRWW98], Anwendung finden. Oder das nichtlineare Gleichungssystem wird linearisiert und durch einen geeigneten linearen Ansatz gelöst. Da die nichtlineare Optimierung gute Näherungspositionen benötigt, anfangs keine weiteren Aussagen über den Berechnungsaufwand sowie die Konvergenz zulässt und nicht explizit das globale Minimum, sondern eventuell auch nur ein lokales Minimum als Ergebnis liefert, wird diese Methode hier nicht weiter verfolgt.
Zur Lösung linearer Gleichungssysteme hingegen existieren zuverlässige und effiziente Methoden, die in dem übernächsten Abschnitt diskutiert werden. Der folgende Abschnitt beschreibt Möglichkeiten zur Überführung des nichtlinearen Gleichungssystems 3.5 in eine lineare Form.

3.2 Linearisierung nichtlinearer Gleichungssysteme

3.2.1 Taylorreihenentwicklung

Die Methode der Fixpunktiteration und speziell der Nullstellensuche nach Newton ermöglicht eine äußerst genaue Linearisierung. Bei dieser Linearisierungsmethode wird das Ausgangsproblem im eindimensionalen Fall in eine Funktion der Form g(x) = 0 umgeformt und anschließend die gesuchte Nullstelle iterativ bestimmt [Nie08]. Abbildung 3.3 beschreibt den Sachverhalt. Ausgehend von einem



Abbildung 3.3: Linearisierung durch Nullstellensuche nach Newton (nach [Nie08]).

Näherungswert x_0 , an dem die Funktion den Wert $f(x_0)$ besitzt, wird eine bessere Näherung für die gesuchte Nullstelle bestimmt. Dies geschieht durch Bildung der Tangente f'(x) an der Näherungsstelle $(x_0, f(x_0))$. Die Tangente schneidet die *x*-Achse im Punkt x_1 , von dem aus die Methode iterativ wiederholt werden kann. Bei der Linearisierung beliebiger Funktionen kann demselben Prinzip gefolgt werden, da für jede stetig differenzierbare Funktion das Fortschreiten der Funktion in einem Nachbarschaftsbereich durch die Tangentengleichung approximiert werden kann. Für genügend gute Näherungswerte ist jede nichtlineare Funktion durch eine Reihenentwicklung darstellbar.

3.2.2 Linearisierungswerkzeug

Eine weitere Methode zur Linearisierung des Gleichungssystems 3.5 ergibt sich durch Subtraktion der *j*-sten Gleichung von allen anderen. Im Fall der Geosensornetzwerke bedeutet das, dass ein Beacon "festgehalten" und von den anderen Beobachtungsgleichungen abgezogen wird. Dabei wird

der festgehaltene Beacon als Linearisierungswerkzeug bezeichnet [MH95]. Der genaue Ablauf wird in Kapitel 7.2.2 sowie in [LR03, MH95, RBTB06, TBBC08] detailliert erläutert.

Nach der Linearisierung, die im Wesentlichen eine Umformung ist, kann das Gleichungssystem 3.5 in die Matrixschreibweise Ax = b überführt werden, wobei A die Koeffizientenmatrix, x den Lösungsvektor und b das Absolutglied bezeichnen. Gesucht ist die Lösung des Gleichungssystems $x = A^{-1}b$.

3.3 Methode der kleinsten Quadrate

Im Gegensatz zu eindeutigen linearen Gleichungssystemen ohne Überbestimmung kann bei überbestimmten Systemen nicht davon ausgegangen werden, dass eine eindeutige Lösung existiert. Es wird eine Lösungstrategie gewählt, die auf der Minimierung des Verbesserungsvektors $\mathbf{v} = \mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}$, mit $\tilde{\mathbf{x}}$ der wahren Lösung, beruht. Es besteht also die Aufgabe, das Ausgleichungsproblem $\mathbf{A}\mathbf{x} \approx \mathbf{b}$ zu lösen. Mit anderen Worten muss ein Vektor \mathbf{x} im Spaltenraum von \mathbf{A} gefunden werden, der, im Sinne der Euklidischen Norm, so nahe wie möglich an \mathbf{b} liegt. Diese Herangehensweise wird als L2 - Norm oder "Methode der kleinsten Quadrate" (*MdkQ*) bezeichnet. Mathematisch entspricht das der Minimierung der Summe der Verbesserungsquadrate:

$$\Omega = \sum_{i=1}^{n} v_i^2 \to \min$$
(3.6)

oder als Vektorschreibweise im Verbesserungsvektor \mathbf{v} :

$$\Omega = \mathbf{v}^T \mathbf{v} \to \min.$$
(3.7)

Wird eine Gewichtung der Beobachtungen angesetzt, ergibt sich als Erweiterung die gewichtete Verbesserungsquadratsumme Ω_{Σ} zu:

$$\Omega_{\Sigma} = \mathbf{v}^{T} \mathbf{P} \mathbf{v}, \tag{3.8}$$

mit der Gewichtsmatrix **P** (das stochastische Modell wird in Kapitel 3.6 beschrieben). Für die weiteren Betrachtungen zu den Lösungsverfahren soll ein lineares, gleichgenaues, reguläres Problem ohne Kovarianzinformation behandelt werden, d. h., die Gewichtsmatrix wird zur Einheitsmatrix und das Minimierungsproblem kann mit Gleichung 3.7 beschrieben werden.

Durch eine orthogonale Projektion von **b** auf den Spaltenraum von **A** kann diese Bedingung eingehalten werden, was gleichbedeutend mit einem Verbesserungsvektor ist, der orthogonal auf dem Spaltenraum von **A**, oder im Nullraum von \mathbf{A}^T , liegt. Die anschließende Ausgleichungsaufgabe kann so mit

$$0 = \mathbf{A}^T \mathbf{v} = \mathbf{A}^T (\mathbf{A}\mathbf{x} - \mathbf{b})$$
(3.9)

beschrieben werden.

Neben der Euklidischen Norm gibt es noch eine Reihe weiterer robuster und nichtrobuster Schätzer. Da in dieser Arbeit ausschließlich die L2 - Norm Anwendung findet, sei für detailliertere Informationen auf Jäger et al. verwiesen [JMS05].

3.4 Lineare Lösungsverfahren

In den nächsten Abschnitten wird die Berechnung von linearen und linearisierten Ausgleichungsproblemen für die Methode der kleinsten Quadrate mittels der Normalgleichungen beschrieben. Da die Lösung über Normalgleichungen jedoch numerisch instabil ist und bei schlecht konditionierten Matrizen versagt, können Orthogonalisierungsmethoden herangezogen werden, um dennoch eine Berechnung zu ermöglichen. Aus diesem Grund werden in diesem Abschnitt zwei bekannte Methoden vorgestellt, die auch im weiteren Verlauf dieser Arbeit zum Einsatz kommen.

3.4.1 Normalgleichungen

Da das Minimum einer Funktion bestimmt werden soll, muss deren 1. Ableitung zu null gesetzt werden. Wird dieses Prinzip auf das Problem 3.7 angewendet, gilt für die quadratische Form von $\mathbf{v}^T \mathbf{v}$:

$$\mathbf{v}^{T}\mathbf{v} = (\mathbf{x}^{T}\mathbf{A}^{T} - \mathbf{b}^{T})(\mathbf{A}\mathbf{x} - \mathbf{b})$$

= $\mathbf{x}^{T}\mathbf{A}^{T}\mathbf{A}\mathbf{x} - \mathbf{x}^{T}\mathbf{A}^{T}\mathbf{b} - \mathbf{b}^{T}\mathbf{A}\mathbf{x} + \mathbf{b}^{T}\mathbf{b}$ (3.10)
= $\mathbf{x}^{T}\mathbf{A}^{T}\mathbf{A}\mathbf{x} - 2\mathbf{x}^{T}\mathbf{A}^{T}\mathbf{b} + \mathbf{b}^{T}\mathbf{b}$.

Dann gilt für die 1. Ableitung nach x:

$$d(\mathbf{v}^{T}\mathbf{v}) = d\mathbf{x}^{T}\mathbf{A}^{T}\mathbf{A}\mathbf{x} + \mathbf{x}^{T}\mathbf{A}^{T}\mathbf{A}d\mathbf{x} - 2d\mathbf{x}^{T}\mathbf{A}^{T}\mathbf{b}$$

= $2d\mathbf{x}^{T}\mathbf{A}^{T}\mathbf{A}\mathbf{x} - 2d\mathbf{x}^{T}\mathbf{A}^{T}\mathbf{b}$ (3.11)
= $2d\mathbf{x}^{T}(\mathbf{A}^{T}\mathbf{A}\mathbf{x} - \mathbf{A}^{T}\mathbf{b}).$

Gleichung 3.11 wird null, wenn der Klammerausdruck zu null wird. Daraus folgt die Bestimmungsgleichung für x:

$$\begin{aligned} (\mathbf{A}^T \mathbf{A} \mathbf{x} - \mathbf{A}^T \mathbf{b}) &= 0 \\ (\mathbf{A}^T \mathbf{A}) \mathbf{x} &= \mathbf{A}^T \mathbf{b}. \end{aligned}$$
 (3.12)

Abschließend folgt durch Linksmultiplikation von $(\mathbf{A}^T \mathbf{A})^{-1}$ und Berücksichtigung der Identität

$$I = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{A}$$
(3.13)

die Berechnungsformel für den Lösungsvektor

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b},\tag{3.14}$$

vorausgesetzt die Inverse $(\mathbf{A}^T \mathbf{A})^{-1}$ existiert, was eine spaltenreguläre Matrix voraussetzt. Das System 3.12 wird als Normalgleichungssystem bezeichnet. In der Literatur finden sich für $\mathbf{N} = \mathbf{A}^T \mathbf{A}$ und $\mathbf{n} = \mathbf{A}^T \mathbf{b}$ die Begriffe der Normalgleichungsmatrix und der rechten Seite [Nie08]. Das verkürzte Normalgleichungssystem kann dann wie folgt geschrieben werden:

$$\mathbf{N}\mathbf{x} = \mathbf{n}.\tag{3.15}$$

Für die Inverse der Normalgleichungsmatrix N erhält man die Kofaktormatrix der Unbekannten $Q_{\hat{x}\hat{x}}$

$$\mathbf{N}^{-1} = (\mathbf{A}^T \mathbf{A})^{-1} = \mathbf{Q}_{\hat{\mathbf{x}}\hat{\mathbf{x}}},\tag{3.16}$$

auf die in den nachfolgenden Abschnitten gesondert eingegangen wird (Kapitel 3.6).

3.4.2 Orthogonalisierungsmethoden

Die Lösung großer überbestimmter Gleichungssysteme über die Normalgleichungen ist eine gute Wahl, da die aus der Matrixmultiplikation $\mathbf{N} = \mathbf{A}^T \mathbf{A}$ resultierende Normalgleichungsmatrix eine quadratische Matrix der Größe ($u \times u$) ist und somit eine einfache Berechnung gewährleistet. Numerische Instabilitäten, auch wenn sie kaum auftreten, können jedoch dazu führen, dass die Lösung über Normalgleichungen zu falschen Ergebnissen führt. In diesem Fall kann der Einsatz von Orthogonalisierungsmethoden zu zufriedenstellenden Ergebnissen führen.

Orthogonalisierungsmethoden transformieren Vektoren auf verschiedene Weise, wobei die Euklidische Norm beibehalten wird. Auch sind orthogonale Transformationen invariant gegenüber der L2 - Norm, d. h., Fehler werden nicht verstärkt.

In der Literatur existieren viele Orthogonalisierungsmethoden, die hier im Einzelnen nicht näher erläutert werden sollen. Die QR-Faktorisierung (*QRF*) und die Singulärwertzerlegung (engl. Singular–Value Decomposition (*SVD*)) sind die mit am besten dokumentierten und untersuchten, weswegen sie in dieser Arbeit Anwendung finden.

3.4.2.1 QR–Faktorisierung

Die QR-Faktorisierung transformiert ein überbestimmtes lineares Gleichungssystem der Form $Ax \approx b$ mit einer beliebigen A-Matrix der Größe ($n \times u$) mit n > u zu:

$$\mathbf{A} = \mathbf{Q} \begin{pmatrix} \mathbf{R}_1 \\ 0 \end{pmatrix}. \tag{3.17}$$

Besitzt A vollen Rang, ist $\mathbf{R_1}$ eine reguläre obere $(n \times n)$ –Dreiecksmatrix und \mathbf{Q} eine orthogonale Matrix. Mit der QR-Faktorisierung wird das Problem $\mathbf{Ax} \approx \mathbf{b}$ in das dreieckförmige System 3.18 transformiert:

$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_{2} = \left\| \mathbf{Q} \begin{pmatrix} \mathbf{R}_{1} \\ 0 \end{pmatrix} \mathbf{x} - \mathbf{b} \right\|_{2} = \left\| \begin{pmatrix} \mathbf{R}_{1} \\ 0 \end{pmatrix} \mathbf{x} - \mathbf{Q}^{T} \mathbf{b} \right\|_{2}.$$
 (3.18)

Das Umstellen der Terme führt zu dem neuen abgeänderten System:

$$\begin{pmatrix} \mathbf{R}_1 \\ 0 \end{pmatrix} \mathbf{x} \approx \mathbf{Q}^T \mathbf{b}.$$
(3.19)

Das System 3.19 wird mit $R_1 x = b$ nach x mit $\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = Q^T b$ aufgelöst und Matrix Q in Submatrizen

 Q_1 und Q_2 zweigeteilt. Dabei besteht Q_1 aus den ersten *u* Spalten und Q_2 aus den verbleibenden n - u Spalten von Q. Mit

$$\mathbf{A} = \begin{pmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{pmatrix} \begin{pmatrix} \mathbf{R}_1 \\ 0 \end{pmatrix} = \mathbf{Q}_1 \mathbf{R}_1$$
(3.20)

entsteht die reduzierte QR-Faktorisierung:

$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \mathbf{Q}_1 \mathbf{b} = \begin{pmatrix} \mathbf{Q}_1 \mathbf{b} \\ \mathbf{Q}_2 \mathbf{b} \end{pmatrix}.$$
(3.21)

Der generelle Ablauf für die Berechnung kann wie folgt zusammengefasst werden:

- 1. Berechnung der reduzierten QRF von A mit $A = Q_1 R_1$,
- 2. Berechnung von $\mathbf{z} = \mathbf{Q_1}^T \mathbf{b}$,
- 3. Rückwärtssubstitution von $R_1 x = z$.

Die Berechnung der QRF wird in Abschnitt 6.1.3 durch Spiegelung mit Householder–Matrizen erreicht, wäre allerdings auch durch Givens-Drehungen möglich.

3.4.2.2 Singulärwertzerlegung

Eine zweite Orthogonalisierungsmethode ist die Singulärwertzerlegung. Die SVD arbeitet wie die QRF mit der A-Matrix und nicht mit $\mathbf{A}^T \mathbf{A}$. Voraussetzung für einen erfolgreichen Einsatz der SVD ist die $(n \times u)$ große A-Matrix mit n > u und vollem Rang. Es werden drei Matrizen, eine orthogonale $(n \times u)$ Matrix U, eine $(u \times u)$ Diagonalmatrix \mathbf{S}_1 und eine orthogonale $(n \times n)$ Matrix V gebildet:

$$\mathbf{A} = \mathbf{U} \begin{pmatrix} \mathbf{S}_1 \\ 0 \end{pmatrix} \mathbf{V}^T.$$
(3.22)

Hat A vollen Rang, gilt $\forall \lambda_{ii} > 0$. Dabei sind alle Werte von S₁ Singulärwerte der Koeffizientenmatrix A und V die Adjungierte einer unitären $(n \times n)$ Matrix. Dafür müssen die Eigenwerte der Matrix A berechnet werden. Die SVD transformiert das Ausgleichungsproblem Ax \approx b in ein Diagonalsystem:

$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_{2} = \left\|\mathbf{U}\begin{pmatrix}\mathbf{S}_{1}\\0\end{pmatrix}\mathbf{V}^{T}\mathbf{x} - \mathbf{b}\right\|_{2} = \left\|\begin{pmatrix}\mathbf{S}_{1}\\0\end{pmatrix}\mathbf{V}^{T}\mathbf{x} - \mathbf{U}^{T}\mathbf{b}\right\|_{2} = \left\|\begin{pmatrix}\mathbf{S}_{1}\\0\end{pmatrix}\mathbf{y} - \mathbf{U}^{T}\mathbf{b}\right\|_{2}.$$
 (3.23)

Für die geänderte Aufgabe bedeutet dies die Lösung des Diagonalsystems nach y:

$$\begin{pmatrix} \mathbf{S}_1 \\ 0 \end{pmatrix} \mathbf{y} \approx \mathbf{U}^T \mathbf{b}. \tag{3.24}$$

Um die reduzierte SVD herzuleiten, muss U aufgeteilt werden. Es entsteht

$$\begin{pmatrix} \mathbf{b_1} \\ \mathbf{b_2} \end{pmatrix} = \mathbf{U}^T \mathbf{b}. \tag{3.25}$$

 U_1 beinhaltet hier die ersten u und U_2 die verbleibenden n - u Spalten. Es gilt weiterhin:

$$\mathbf{A} = \begin{pmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{pmatrix} \begin{pmatrix} \mathbf{S}_1 \\ 0 \end{pmatrix} \mathbf{V}^T = \mathbf{U}_1 \mathbf{S}_1 \mathbf{V}^T.$$
(3.26)

Abschließend erhält man nun für die reduzierte SVD:

$$\begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} = \mathbf{U}_1 \mathbf{b} = \begin{pmatrix} \mathbf{U}_1 \mathbf{b} \\ \mathbf{U}_2 \mathbf{b} \end{pmatrix}.$$
(3.27)

Analog zur reduzierten QRF ergibt sich für die reduzierte SVD folgender genereller Ablauf:

- 1. Aufstellen der reduzierten SVD von A mit $A = U_1 S_1 V^T$,
- 2. Auflösen des Diagonalsystems $S_1y = U_1b$ nach y,
- 3. Ermittlung von x durch Rückwärtssubstitution mit x = Vy.

Eine Übersicht effizienter Lösungsmethoden für die QRF und SVD können in Golub et al. und Stewart nachgelesen werden [GR71, GL96, Ste00].

3.5 Varianzfortpflanzung

Das allgemeine Varianzfortpflanzungsgesetz (*VFG*) kann aus den Rechenregeln für Erwartungswert und Varianz von Zufallsvariablen bei deren linearer Transformation abgeleitet werden. Gegeben sei die lineare Form

$$\mathbf{y} = \mathbf{F}\mathbf{x} + \mathbf{b},\tag{3.28}$$

mit dem $(u \times 1)$ Zufallsvektor x, dem $(n \times 1)$ Zufallsvektor y, der $(n \times u)$ Matrix F und dem konstanten $(n \times 1)$ Vektor b. x enthält dabei die Ausgangsvariablen und y die abgeleiteten Linearkombinationen. Für die Fortpflanzung der Erwartungswerte *E* gilt:

$$E(\mathbf{y}) = E(\mathbf{F}\mathbf{x} + \mathbf{b}) = \mathbf{F} \cdot E(\mathbf{x}) + \mathbf{b},$$
(3.29)

und für die Fortpflanzung der Varianzen und Kovarianzen

$$\mathbf{V}(\mathbf{y}) = \mathbf{C}_{\mathbf{F}\mathbf{F}} = \mathbf{V}(\mathbf{F}\mathbf{x} + \mathbf{b}) = \mathbf{F} \cdot \mathbf{V}(\mathbf{x}) \cdot \mathbf{F}^T = \mathbf{F} \cdot \mathbf{C}_{\mathbf{x}\mathbf{x}} \cdot \mathbf{F}^T,$$
(3.30)

mit C_{xx} der Kovarianzmatrix von x, und C_{FF} der Kovarianzmatrix von y. Für den Fall, dass der funktionale Zusammenhang y = F(x) nichtlinear ist, hat eine Linearisierung nach Abschnitt 3.2.1 zu erfolgen. Die Matrix F enthält dann die partiellen Ableitungen y_i nach den x_k [Grü96, Nie08]

$$\mathbf{F}_{ik} = \frac{\partial \mathbf{y}_i}{\partial \mathbf{x}_k}.$$
(3.31)

3.6 Netzausgleichung

Die Ausgleichungsrechnung nach der Methode der kleinsten Quadrate ist eines der zentralen mathematischen Werkzeuge der Geodäsie. Auch auf das Anwendungsbeispiel der drahtlosen Sensornetzwerke lässt sich diese Methode anwenden. Redundante Messungen führen zu Überbestimmung und damit zur Erhöhung der Genauigkeit und Zuverlässigkeit. Die Berechnung wahrscheinlichster Schätzwerte für die gesuchten Parameter, hier der Positionen der unbekannten Sensorknoten, mit minimalen Varianzen ist das Ziel dieser Auswertung. Dafür werden normalverteilte Messfehler vorausgesetzt.

Es existieren verschiedene funktionale Modelle für die Ausgleichungsrechnung [Nie08, Grü03, JMS05]:

 Gauss–Markov–Modell (*GMM*) oder vermittelnde Ausgleichung. Es werden Parameter geschätzt, die durch Beobachtungen vermittelt werden, da sie i. d. R., wie im hier behandelten Fall der drahtlosen Sensornetzwerke, nicht direkt messbar sind (Koordinaten aus Strecken- oder Winkelmessungen).

- Bedingte Ausgleichung. Es werden ausgeglichene Beobachtungen geschätzt, die funktionale Zusammenhänge zwischen den Beobachtungen konsistent machen. Diese Zusammenhänge werden im funktionalen Modell als Bedingungen zwischen den Beobachtungen formuliert.
- Gauss–Helmert–Modell (GHM) oder bedingte Ausgleichung mit Unbekannten. Die Formulierung von Bedingungen ist hier nur mit zusätzlichen Parametern möglich.
- GMM oder GHM mit Restriktionen oder Allgemeinfall der Ausgleichung. Hier werden zusätzliche Bedingungen zwischen den Parametern eingefügt.

Wie bereits in Abschnitt 3.3 erläutert wurde, besteht das allgemeine Prinzip der Methode der kleinsten Quadrate darin, dass die Beobachtungen Verbesserungen erhalten, die zu den wahrscheinlichsten Werten für die gesuchten Parametern, hier den Koordinaten der Sensorknoten führen.

3.6.1 Das stochastische Modell

Nachdem das funktionale Modell bereits in Abschnitt 3.4 eingeführt wurde, erfolgt an dieser Stelle die Beschreibung des stochastischen Modells [Nie08, Grü03, JMS05].

Gegeben seien zwei Distanzen \hat{d}_1 und \hat{d}_2 , für die jeweils *n* Beobachtungen vorliegen. Der $(n \times 2)$ große Beobachtungsvektor **b** kann somit beschrieben werden als:

$$\mathbf{b} = \begin{pmatrix} \mathbf{b_1}^T \\ \mathbf{b_2}^T \end{pmatrix} = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \end{pmatrix}.$$
(3.32)

Für die Erwartungswerte bedeutet das:

$$\mathbf{E}(\mathbf{B}) = \begin{pmatrix} \mathbf{E}(B_1) \\ \mathbf{E}(B_2) \end{pmatrix} = \begin{pmatrix} \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^n b_{i1} \\ \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^n b_{i2} \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} = \boldsymbol{\mu},$$
(3.33)

mit μ dem Erwartungswertvektor. Kann der Erwartungswert, sollte er vorliegen, dem wahren Wert gleichgesetzt werden, können auch die wahren Residuen berechnet werden:

$$\boldsymbol{\varepsilon}^{T} = \begin{pmatrix} \boldsymbol{\varepsilon}_{1}^{T} \\ \boldsymbol{\varepsilon}_{2}^{T} \end{pmatrix} = \begin{pmatrix} b_{11} - \mu_{1} & b_{12} - \mu_{1} & \dots & b_{1n} - \mu_{1} \\ b_{21} - \mu_{2} & b_{22} - \mu_{2} & \dots & b_{2n} - \mu_{2} \end{pmatrix} = \mathbf{b}^{T} - \mu \mathbf{e}^{T}.$$
(3.34)

Die Standardabweichung einer Zufallszahl ist nach [Nie08] definiert als:

$$\sigma_0^2 = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^n \varepsilon_i^2 = \mathbb{E}(\varepsilon^T \varepsilon) = \mathbb{E}\left((b_i - \mu_x)^2\right).$$
(3.35)

Erweitert man diesen Ansatz auf zweidimensionale Zufallsvariablen, erhält man eine (2×2) Matrix mit allen Angaben zu den Genauigkeiten des Zufallsvektors. Es folgt die Kovarianzmatrix C_{II} :

$$\mathbf{C}_{\mathbf{l}\mathbf{l}} = E\left([\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}}][\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}}]^{T}\right) = E\left([\mathbf{x} - E(\mathbf{x})][\mathbf{x} - E(\mathbf{x})]^{T}\right) = E(\boldsymbol{\varepsilon}^{T}\boldsymbol{\varepsilon}), \quad (3.36)$$

mit ε als $(n \times 2)$ Matrix aus Gleichung 3.34. Durch Ausmultiplizieren von Gleichung 3.36 folgt weiterhin:

$$\mathbf{C}_{\mathbf{l}\mathbf{l}} = E\left(\left[\begin{array}{c} \varepsilon_{\mathbf{1}}^{T} \\ \varepsilon_{\mathbf{2}}^{T} \end{array}\right] \left[\begin{array}{c} \varepsilon_{\mathbf{1}}^{T} & \varepsilon_{\mathbf{2}}^{T} \end{array}\right]\right) = E\left(\begin{array}{c} \varepsilon_{\mathbf{1}}^{T}\varepsilon_{\mathbf{1}} & \varepsilon_{\mathbf{1}}^{T}\varepsilon_{\mathbf{2}} \\ \varepsilon_{\mathbf{2}}^{T}\varepsilon_{\mathbf{1}} & \varepsilon_{\mathbf{2}}^{T}\varepsilon_{\mathbf{2}} \end{array}\right).$$
(3.37)

Dabei gilt:

$$E(\boldsymbol{\varepsilon}_{1}^{T}\boldsymbol{\varepsilon}_{1}) = E(\boldsymbol{\varepsilon}_{1}^{2}) = \sigma_{1}^{2}$$

$$E(\boldsymbol{\varepsilon}_{2}^{T}\boldsymbol{\varepsilon}_{2}) = E(\boldsymbol{\varepsilon}_{2}^{2}) = \sigma_{2}^{2}.$$
(3.38)

Somit stehen auf der Hauptdiagonalen die Varianzen der jeweiligen Zufallsvariablen. Das Nebendiagonalelement wird als Kovarianz bezeichnet und ist stets zwischen zwei Zufallsvariablen definiert. Es gilt:

$$\sigma_{12} = E(\boldsymbol{\varepsilon}_1^T \boldsymbol{\varepsilon}_2) = E(\boldsymbol{\varepsilon}_2^T \boldsymbol{\varepsilon}_1). \tag{3.39}$$

Mit dem Korrelationskoeffizienten $\rho_{12} = \frac{\sigma_{12}}{\sigma_1 \cdot \sigma_2}$ und der Cauchy–Schwarzschen Ungleichung folgt für die Erwartungswerte und die Kovarianz:

$$(E(\varepsilon_1^T \varepsilon_2))^2 = (E(\varepsilon_1 \varepsilon_2))^2 \leq E(\varepsilon_1^2) \cdot E(\varepsilon_2^2)$$

$$(\sigma_{12})^2 = (\rho_{12} \cdot \sigma_1 \cdot \sigma_2)^2 \leq \sigma_1^2 \cdot \sigma_2^2.$$

$$(3.40)$$

Damit folgt für den Korrelationskoeffizienten der Wertebereich $-1 \le \rho_{12} \le +1$. Der Korrelationskoeffizient gilt als Maß für die stochastische Abhängigkeit zwischen \hat{d}_1 und \hat{d}_2 . Strebt $\rho_{12} \to 0$ sind die Zufallsvariablen kaum oder nicht korreliert. Bei $\rho_{12} \to \pm 1$ liegt starke bis sehr starke positive oder negative Korrelation vor. Abschließend folgt mit Gleichungen 3.38 und 3.40 als allgemeingültige Formel für die Kovarianzmatrix einer zweidimensionalen Zufallsvariablen:

$$\mathbf{C}_{\mathbf{l}\mathbf{l}} = \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}.$$
(3.41)

3.6.2 Gewichtung von Beobachtungen

In der Kovarianzmatrix aus Gleichung 3.41 sind in der Hauptdiagonalen die Varianzen der Beobachtungen b_i aufgeführt. In der Nebendiagonalen befinden sich mögliche Kovarianzen oder Korrelationen zwischen den Beobachtungen. Diese Kovarianzmatrix ist für jeden Beobachtungsvektor vor einer Ausgleichung aufzustellen. Eine häufig genutzte und auch gerechtfertigte Annahme ist der Verzicht auf Korrelationen im stochastischen Modell. Die Kovarianzmatrix ergibt sich dann zu:

$$\mathbf{C}_{\mathbf{ll}} = \begin{pmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & 0 & \\ & & \sigma_3^2 & & \\ & 0 & \ddots & \\ & & & & \sigma_n^2 \end{pmatrix}.$$
 (3.42)

Werden weiterhin gleichgenaue Beobachtungen angenommen gilt:

$$\sigma_1^2 = \sigma_2^2 = \dots = \sigma_n^2 = \sigma^2,$$
(3.43)

und für die Kovarianzmatrix:

$$\mathbf{C}_{\mathbf{II}} = \begin{pmatrix} \sigma^{2} & & & \\ & \sigma^{2} & & \\ & & \sigma^{2} & & \\ & & \sigma^{2} & & \\ & & & \sigma^{2} \end{pmatrix} = \sigma^{2} \cdot I.$$
(3.44)

Oftmals sind keine Genauigkeitsinformationen für den Beobachtungsvektor verfügbar. In der Regel lassen sich nur Aussagen über Genauigkeitsrelationen zwischen den Beobachtungen treffen. Das absolute Genauigkeitniveau σ_0 ist hier also die unbekannte Größe. Diese wird vor der Ausgleichung beliebig willkürlich festgelegt und unter der gängigen Bezeichnung $\sigma_{0_apriori}$ eingeführt. Damit kann jede Kovarianzmatrix dergestalt zerlegt werden, dass:

$$\mathbf{C}_{\mathbf{l}\mathbf{l}} = \sigma_0^2 _{apriori} \cdot \mathbf{Q}_{\mathbf{l}\mathbf{l}}. \tag{3.45}$$

Die Kofaktormatrix der Beobachtungen Q_{II} enthält Aussagen über Genauigkeitsrelationen zwischen den Beobachtungen. Für die Schätzung der Parameter x nach der Methode der kleinsten Quadrate muss nur die Kofaktormatrix der Beobachtungen Q_{II} bekannt sein [Nie08]. Während des Ausgleichungsprozesses wird für den unbekannten Varianzfaktor $\sigma_{0_apriori}$ ein Schätzwert $\sigma_{0_aposteriori}$ bestimmt.

Haben die Beobachtungen eine unterschiedlich starke Bedeutung für die Berechnung der unbekannten Parameter, ist die Einführung s.g. Gewichte in das stochastische Modell möglich. Formal ist die Gewichtsmatrix **P** die Inverse der Kofaktormatrix und wird ausgedrückt als:

$$\mathbf{P} = \mathbf{Q}_{\mathbf{l}\mathbf{l}}^{-1}.$$

Mit der Varianz σ^2 gilt für die Gewichtsmatrix:

$$\mathbf{P} = \mathbf{Q}_{\mathbf{II}}^{-1} = \begin{pmatrix} \frac{\sigma_{0_apriori}^2}{\sigma_1^2} & & & \\ & \frac{\sigma_{0_apriori}^2}{\sigma_2^2} & & & \\ & & \frac{\sigma_{0_apriori}^2}{\sigma_3^2} & & \\ & & & \frac{\sigma_{0_apriori}^2}{\sigma_n^2} \end{pmatrix} = \begin{pmatrix} p_{11} & & & \\ & p_{22} & 0 & \\ & & p_{33} & & \\ & & 0 & & \ddots & \\ & & & & p_{nn} \end{pmatrix}.$$
(3.47)

Die Gewichte der einzelnen Beobachtung sind damit durch $p_{ii} = \frac{\sigma_{0_apriori}^2}{\sigma_i^2}$ gegeben. Es ist ersichtlich, dass Beobachtungen mit einer großen Standardabweichung ein kleines Gewicht für die Berechnung der unbekannten Parameter erhalten.

Mit diesen Grundlagen lässt sich die in Gleichung 3.7 formulierte Minimierungsaufgabe für die gewichtete Berechnung zu:

$$\Omega = \mathbf{v}^T \mathbf{P} \mathbf{v} \to \min \tag{3.48}$$

erweitern.

Da in dieser Arbeit ausschließlich die vermittelnde und die bedingte Ausgleichung angewendet werden, sollen auch nur diese im Folgenden vorgestellt werden. Für weiterführende Informationen und detaillierte Herleitungen zu den anfangs aufgezählten Ausgleichungsmethoden sei auf [Nie08, JMS05, Grü03] verwiesen.

3.6.3 Gauß-Markov-Modell

Das Gauss–Markov–Modell, oder auch vermittelnde Ausgleichung genannt, basiert auf dem funktionalen Zusammenhang zwischen den Beobachtungen und den zu schätzenden Parametern:

$$E(\mathbf{b}) = \mathbf{F}(\mathbf{x}) \rightarrow \mathbf{b} + \mathbf{v} = \mathbf{F}(\mathbf{x}). \tag{3.49}$$

Das Ziel ist eine erwartungstreue Schätzung $E(\hat{x}) = \tilde{x}$. Die Differenz zwischen der Anzahl der Beobachtungen *n* und der Anzahl der unbekannten Parameter *u* ist die Überbestimmung (Redundanz *r*) des Ausgleichungsproblems. Für den Fall, dass für das Ausgleichungsproblem Nichtlinearität vorliegt, ist das Gleichungssystem nach Abschnitt 3.2 zu linearisieren.

Die weitere Berechnung kann mit den Methoden nach Abschnitt 3.4 erfolgen. An dieser Stelle wird die Herleitung des GMMs anhand der Normalgleichungen durchgeführt.

Für den Fall, dass die Linearisierung mittels Taylorreihenentwicklung durchgeführt wurde, ergibt sich für den Ausgleichungsprozess das folgende funktionale Modell oder die s.g. Verbesserungsgleichungen [Grü03]:

$$\mathbf{v} = \mathbf{A}\mathbf{x} - \mathbf{b},\tag{3.50}$$

mit der $(n \times u)$ Koeffizientenmatrix **A**. Im nichtlinearen Fall enthält **A** die partiellen Ableitungen der Beobachtungen nach den Parametern.

Das Einsetzen von Gleichung 3.50 in die Zielfunktion 3.48 sowie Bildung der Ableitungen nach den Parametern liefert als Minimum die Normalgleichungen

$$\mathbf{N}\mathbf{x} = \mathbf{n} \text{ mit } \mathbf{N} = \mathbf{A}^T \mathbf{P} \mathbf{A} \text{ und } \mathbf{n} = \mathbf{A}^T \mathbf{P} \mathbf{b}.$$
(3.51)

Die Lösung ergibt sich zu

$$\mathbf{x} = \mathbf{N}^{-1}\mathbf{n}.\tag{3.52}$$

Wird im Weiteren das VFG auf Gleichung 3.52 angewendet, ergibt sich letztendlich die Kofaktormatrix der geschätzten Parameter mit

$$\mathbf{Q}_{\mathbf{x}\mathbf{x}} = \mathbf{N}^{-1} = (\mathbf{A}^T \mathbf{P} \mathbf{A})^{-1}$$
(3.53)

als Inverse der Normalgleichungsmatrix. Abschließend lässt sich der aposteriori Varianzfaktor $\sigma_{0_aposteriori}^2$ nach

$$\sigma_{0_aposteriori}^{2} = \frac{\mathbf{v}^{T} \mathbf{P} \mathbf{v}}{n-u} = \frac{\Omega}{r}$$
(3.54)

berechnen.

Im GMM beinhaltet die Kofaktormatrix $Q_{\hat{x}\hat{x}}$ die komplette Genauigkeitsinformation bezüglich der zu schätzenden Parameter sowie über das VFG auch Funktionen der Parameter:

$$\mathbf{Q}_{\hat{\mathbf{x}}\hat{\mathbf{x}}} = \begin{pmatrix} q_{\hat{\mathbf{x}}\hat{\mathbf{x}}} & q_{\hat{\mathbf{x}}\hat{\mathbf{y}}} \\ q_{\hat{\mathbf{y}}\hat{\mathbf{x}}} & q_{\hat{\mathbf{y}}\hat{\mathbf{y}}} \end{pmatrix}.$$
(3.55)

Aus einer Spektralzerlegung dieser Kovarianzmatrix lassen sich globale Genauigkeitsmaße für ein ausgeglichenes Netz bestimmen. Die Standardabweichungen der gesuchten Parameter ergeben sich zu

$$\sigma_x = \sigma_{0_aposteriori} \sqrt{q_{\hat{\mathbf{x}}\hat{\mathbf{x}}}} \quad \text{und} \quad \sigma_y = \sigma_{0_aposteriori} \sqrt{q_{\hat{\mathbf{y}}\hat{\mathbf{y}}}}.$$
(3.56)

Ebenfalls lassen sich die Parameter der Punktfehlerellipsen aus $Q_{\hat{x}\hat{x}}$ ableiten. Im Fall von Koordinaten als gesuchte Parameter stellen die Fehlerellipsen Vertrauensbereiche für die Punktlage dar. Die große und die kleine Halbachse *a* und *b* der Fehlerellipse errechnen sich:

$$a = \sigma_{0_aposteriori} \sqrt{\lambda_1},$$

$$b = \sigma_{0_aposteriori} \sqrt{\lambda_2},$$
(3.57)

und für den Richtungswinkel φ der großen Halbachse:

$$\tan 2\varphi = \frac{2q_{\hat{\mathbf{x}}\hat{\mathbf{y}}}}{q_{\hat{\mathbf{x}}\hat{\mathbf{x}}} - q_{\hat{\mathbf{y}}\hat{\mathbf{y}}}},\tag{3.58}$$

mit den Eigenwerten $\lambda_{1,2}$:

$$\lambda_{1,2} = \frac{1}{2} (q_{\hat{\mathbf{x}}\hat{\mathbf{x}}} - q_{\hat{\mathbf{y}}\hat{\mathbf{y}}}) \pm \frac{1}{2} \sqrt{(q_{\hat{\mathbf{x}}\hat{\mathbf{x}}} - q_{\hat{\mathbf{y}}\hat{\mathbf{y}}})^2 + 4q_{\hat{\mathbf{x}}\hat{\mathbf{y}}}^2}.$$
(3.59)

3.6.4 Bedingte Ausgleichung

Die Besonderheit bei der bedingten Ausgleichung ist dadurch charakterisiert, dass in dem gesamten Ausgleichungsvorgang keine Unbekannten vorkommen. Vielmehr bewegt man sich ausschließlich im Bereich der Beobachtungen. Die Berechnung ausgeglichener Koordinaten wird in zwei Schritten erreicht, wobei im ersten lediglich die ausgeglichenen Beobachtungen $\hat{\mathbf{l}} = \mathbf{l} + \mathbf{v}$ berechnet werden. Im zweiten Schritt folgt daraus die Berechnung der Unbekannten x.

Bei Ausgleichungsaufgaben geringer Redundanz ergibt sich der Vorteil der bedingten Ausgleichung, da die Dimension des zu invertierenden Normalgleichungssystems auf die Redundanz r des Problems beschränkt ist [JMS05]. Das heißt, die Anzahl der Bedingungsgleichungen r ergibt sich als Differenz aus der Anzahl aller Beobachtungen n und der Anzahl der notwendigen Beobachtungen m:

$$n-m=r. (3.60)$$

Das nichtlineare funktionale Model der bedingten Ausgleichung ergibt sich nach [Grü03] zu

$$\mathbf{f}(\hat{\mathbf{l}}) = \mathbf{l} + \mathbf{v} = \mathbf{s}. \tag{3.61}$$

Diese nichtlinearen Bedingungsgleichungen 3.61 werden mittels Taylorreihenentwicklung (siehe Abschnitt 3.2.1) am Entwicklungspunkt der durchgeführten Beobachtungen 1 linearisiert

$$\mathbf{f}(\mathbf{l}+dl) = \mathbf{f}(\mathbf{l}) + \frac{\delta \mathbf{f}}{\delta \mathbf{l}} dl \to \mathbf{s} = \mathbf{f}(\mathbf{l}+v) = \mathbf{f}(\mathbf{l}) + \frac{\delta \mathbf{f}}{\delta \mathbf{l}} \mathbf{v}.$$
(3.62)

Mit den partiellen Ableitungen wird die Jakobimatrix B aufgestellt

$$\mathbf{B}^{T} = \begin{bmatrix} \frac{\delta f_{j}}{\delta l_{i}} \end{bmatrix}, \quad \text{mit} \quad i = 1, ..., n; \quad j = 1, ..., r.$$
(3.63)

In Matrizendarstellung ergibt sich

$$\mathbf{B}^T \mathbf{v} = \mathbf{s} - \mathbf{f}(\mathbf{l}) = \mathbf{w} \tag{3.64}$$

mit dem Widerspruchsvektor w. Das Aufstellen der Gewichtsmatrix erfolgt identisch zu Gleichung 3.47. Daraus ergeben sich die Normalgleichnungsmatrix N

$$\mathbf{N} = \mathbf{B}^T \mathbf{P}^{-1} \mathbf{B} \tag{3.65}$$

und der Korrelatenvektor k zu

$$\mathbf{k} = \mathbf{N}^{-1}\mathbf{w}.\tag{3.66}$$

Die Verbesserungen der Beobachtungen berechnen sich dann zu

$$\mathbf{v} = \mathbf{P}^{-1}\mathbf{B}\mathbf{k} \tag{3.67}$$

und abschließend die ausgeglichenen Beobachtungen zu

$$\hat{\mathbf{l}} = \mathbf{l} + \mathbf{v}. \tag{3.68}$$

3.6.5 Apriori–Genauigkeitsanalyse bei der Netzplanung

Für das beschriebene Ausgleichungsverfahren werden als Eingangsgrößen der Beobachtungsvektor **b**, die Kovarianzmatrix der Beobachtungen C_{II} und die Koeffizientenmatrix **A** benötigt. Eventuell können auch Bedingungen zwischen den unbekannten Parametern oder Vorinformationen ins Ausgleichungsmodell einfließen. Stehen diese Informationen nicht oder nur partiell zur Verfügung, wie z. B. bei einer Vorplanung des zu erwartenden Netzes, können dennoch Aussagen über **A** und C_{II} getroffen werden. Um die zu erwartende Genauigkeit der Messungen abzuschätzen, können z. B. Herstellerangaben für das jeweilige Messsystem herangezogen oder alternativ in Testmessungen vorab empirisch ermittelt werden, um nachfolgend in die Kovarianzmatrix einzufließen. Die Informationen für die Koeffizientenmatrix können gewonnen werden, indem die Lage der Beacons festgelegt werden. Für die Unbekannten sind dann Näherungswerte zu definieren, die die mögliche Lage der Sensorknoten relativ gut repräsentieren. Mit diesen Informationen können die linearisierten Beobachtungsgleichungen aufgestellt werden.

Die zu erwartende Genauigkeit des Netzes wird durch die Kovarianzmatrix der geschätzten Unbekannten $C_{\hat{x}\hat{x}}$ beschrieben. Diese lässt sich durch Anwendung des VFGs aus dem Normalgleichungssystem 3.51 gewinnen. Unter ausschließlicher Verwendung von A und C_{II} ergibt sich für die geschätzten Unbekannten \hat{x} :

$$\hat{\mathbf{x}} = \left(\mathbf{A}^T \mathbf{C}_{\mathbf{l}\mathbf{l}}^{-1} \mathbf{A}\right)^{-1} \mathbf{A}^T \mathbf{C}_{\mathbf{l}\mathbf{l}}^{-1} \mathbf{b}.$$
(3.69)

Die Kovarianzmatrix der geschätzten Unbekannten errechnet sich zu:

$$\mathbf{C}_{\hat{\mathbf{x}}\hat{\mathbf{x}}} = \left(\mathbf{A}^T \mathbf{C}_{\mathbf{l}\mathbf{l}}^{-1} \mathbf{A}\right)^{-1}$$
(3.70)

und enthält alle relevanten Informationen über die Bestimmungsgenauigkeit der unbekannten Sensorknoten im geplanten Netzwerk. Damit können insbesondere die zu erwartende Standardabweichung, die zu erwartenden mittleren Punktfehler und die sich daraus ergebenen Punktfehlerellipsen abgeschätzt werden.

3.6.6 Eliminierung einzelner Unbekannter

Für manche Berechnungen ist es von Vorteil, wenn einzelne Unbekannte aus den Fehlergleichungen eliminiert werden. Dazu gibt es einige Methoden, von denen an dieser Stelle eine näher erörtert werden soll, da sie für die Berechnungen in Kapitel 7.2.3 benutzt wird.

Eliminierung einzelner Unbekannter durch Blockzerlegung

Der allgemeingültige Ansatz für eine Eliminierung einzelner Unbekannter ist eine blockweise Reduktion des funktionalen Modells [Nie08]. Ausgehend von gleichgenauen Beobachtungen ergeben sich die Verbesserungsgleichungen (Gleichung 3.50). Der Parametervektor und die Beobachtungsgleichungen werden unterteilt in die Subvektoren x_1 und x_2 sowie in die Subkoeffizientenmatrizen A_1 und A_2 . Man erhält:

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}, \tag{3.71}$$

$$\mathbf{A} = \left(\begin{array}{cc} \mathbf{A}_1 & \mathbf{A}_2 \end{array}\right). \tag{3.72}$$

Damit ergeben sich die Verbesserungsgleichungen zu:

$$\mathbf{v} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} - \mathbf{b}, \tag{3.73}$$

woraus die Normalgleichungen und deren rechte Seiten folgen:

$$\mathbf{N} = \mathbf{A}^{T} \mathbf{P} \mathbf{A} = \begin{bmatrix} \mathbf{A}_{1}^{T} \\ \mathbf{A}_{2}^{T} \end{bmatrix} \mathbf{P} \begin{bmatrix} \mathbf{A}_{1} & \mathbf{A}_{2} \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{A}_{1}^{T} \mathbf{P} \mathbf{A}_{1} & \mathbf{A}_{1}^{T} \mathbf{P} \mathbf{A}_{2} \\ \mathbf{A}_{2}^{T} \mathbf{P} \mathbf{A}_{1} & \mathbf{A}_{2}^{T} \mathbf{P} \mathbf{A}_{2} \end{bmatrix} = \begin{bmatrix} \mathbf{N}_{11} & \mathbf{N}_{12} \\ \mathbf{N}_{21} & \mathbf{N}_{22} \end{bmatrix},$$
(3.74)

$$\mathbf{n} = \mathbf{A}^{T} \mathbf{P} \mathbf{b} = \begin{bmatrix} \mathbf{A}_{1}^{T} \\ \mathbf{A}_{2}^{T} \end{bmatrix} \mathbf{P} \mathbf{b} = \begin{bmatrix} \mathbf{A}_{1}^{T} \mathbf{P} \mathbf{b} \\ \mathbf{A}_{2}^{T} \mathbf{P} \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{n}_{1} \\ \mathbf{n}_{2} \end{bmatrix}.$$
(3.75)

Für das Gesamtsystem ergibt sich die Blockstruktur:

$$\begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}.$$
 (3.76)

Weiterhin folgt aus der zweiten Gleichung von 3.77:

$$N_{21}x_1 + N_{22}x_2 = n_2$$

$$N_{22}x_2 = n_2 - N_{21}x_1.$$
(3.77)

Falls die Inverse N_{22}^{-1} existiert, erhält man formal die Bestimmungsgleichung für x_2 :

$$\mathbf{x}_2 = \mathbf{N}_{22}^{-1} \mathbf{n}_2 - \mathbf{N}_{22}^{-1} \mathbf{N}_{21} \mathbf{x}_1. \tag{3.78}$$

Eingesetzt in 3.77 folgt

$$\begin{split} N_{11}x_1 + N_{12}x_2 &= n_1 \\ N_{11}x_1 + N_{12} \left(N_{22}^{-1}n_2 - N_{22}^{-1}N_{21}x_1 \right) &= n_1 \\ \left(N_{11} - N_{12}N_{22}^{-1}N_{21} \right) x_1 &= n_1 - N_{12}N_{22}^{-1}n_2 \end{split} \tag{3.79}$$

oder mit der reduzierten Normalgleichungsmatrix $\check{N}_{11} = N_{11} - N_{12}N_{22}^{-1}N_{21}$ und der reduzierten rechten Seite $\check{n}_1 = n_1 - N_{12}N_{22}^{-1}n_2$

$$\tilde{\mathbf{N}}_{11}\mathbf{x}_1 = \check{\mathbf{n}}_1. \tag{3.80}$$

Mit der Inversen der reduzierten Normalgleichungsmatrix \check{N}_{11} lässt sich der reduzierte Unbekanntenvektor direkt bestimmen:

$$\mathbf{x}_1 = \check{\mathbf{N}}_{ll}^{-1}\check{\mathbf{n}}_1. \tag{3.81}$$

Für die Verbesserungen im reduzierten Modell ergeben sich durch Einsetzen von Gleichung 3.78 in Gleichung 3.73:

$$\mathbf{v} = \mathbf{A}_{1}\mathbf{x}_{1} + \mathbf{A}_{2}\mathbf{N}_{22}^{-1}\mathbf{n}_{2} - \mathbf{A}_{2}\mathbf{N}_{22}^{-1}\mathbf{N}_{21}\mathbf{x}_{1} - \mathbf{b}$$

$$= \left(\mathbf{A}_{1} - \mathbf{A}_{2}\mathbf{N}_{22}^{-1}\right)\mathbf{x}_{1} - \left(\mathbf{b} - \mathbf{A}_{2}\mathbf{N}_{22}^{-1}\mathbf{n}\right).$$

$$(3.82)$$

An der Berechnung des Schätzwertes $\sigma_{0_aposteriori}$ für die Gewichtseinheit nach der Ausgleichung ändert sich nichts. Die Berechnung erfolgt über:

$$\sigma_{0_aposteriori}^2 = \frac{\mathbf{v}^T \mathbf{P} \mathbf{v}}{n-u}.$$
(3.83)

Wie bereits beschrieben, werden im weiteren Verlauf der Arbeit alle stochastischen Größen mit dem Symbol (^) und die wahren Größen mit (~) gekennzeichnet.

4 Entwicklungsstand zur Lokalisierung in Sensornetzwerken

"It [the computer] is approachable only through complex jargon that has nothing to do with the tasks for which which people actually use computers. The state of the art is perhaps analogous to the period when scribes had to know as much about making ink or baking clay as they did about writing."

(Mark Weiser, am. Informatiker)

4.1 Klassifizierung der Algorithmen

Der Lokalisierungsprozess in einem drahtlosen Geosensornetzwerk basiert auf unterschiedlichen Netzwerktopologien. Wie in Kapitel 2 beschrieben, setzen die meisten Lokalisierungsalgorithmen einige Knoten voraus, denen ihre Position inertial bekannt ist – die Beacons. In dem ersten Fall



Abbildung 4.1: Ad hoc vernetzte Sensorknoten: (a) mit Beacons, (b) mit Beacons und einer festen Senke (freies Netz), (c) ohne Beacons und vier Senken (Infrastrukturfall).

(Abbildung 4.1 (a)) besteht das Netz aus ad hoc vernetzten Sensorknoten und den Beacons. Hier wird der Lokalisierungsalgorithmus entweder auf allen Sensorknoten und/oder zusätzlich auf den Beacons durchgeführt. Das bringt den Vorteil mit sich, dass jeder Knoten seine Position mit dem geringsten Kommunikationsaufwand im Netz selbst bestimmt.

Im Gegensatz dazu (Abbildung 4.1 (c)) senden alle Sensorknoten ihre Daten zur Senke (auch häufig als Basis bezeichnet) im Netzwerk, bei der es sich z. B. um einem Server, PC oder Laptop handeln

kann. Die Senke übernimmt dabei alle Berechnungsschritte der komplexen Lokalisierungsalgorithmen. Die Nachteile dieser Vorgehensweise sind der extrem hohe Kommunikationsaufwand im Netzwerk sowie die Anfälligkeit bei Unerreichbarkeit der Senke durch deren Ausfall oder Blockade der Route zwischen den Knoten und der Senke. Von Vorteil ist allerdings, dass die Senke Teil einer Infrastruktur ist, die die Bestimmung eines geodätischen Datums (Ursprung, Rotation, Translation und Maßstab eines Referenzkoordinatensystems) ermöglicht. Damit kann die Position eines einzelnen Sensorknotens in einem weltweiten Referenzsystem (z. B. das World Geodetic System 1984 (WGS 84)) wiedergegeben werden.

Die Techniken haben den Nachteil, dass sie entweder einen hohen Kommunikations- oder Berechnungsaufwand nach sich ziehen. Im günstigsten Fall ergibt sich eine Mischtopologie (Abbildung 4.1 (b)), in der der weniger komplexe Teil der Lokalisierung auf den Sensorknoten und der energieaufwändige Teil, im Hinblick auf möglichst geringe Kommunikationskosten, auf die Senke ausgelagert wird. Dieser hybride Algorithmus muss flexibel genug sein, um alle möglichen Knotenausfälle zu kompensieren und damit sowohl eine dauerhafte, verteilte als auch eine zentralisierte Berechnung zuzulassen, was im Endeffekt zu einer längeren Lebensdauer des Netzwerkes führen kann.

In Abhängigkeit von der verwendeten Messgröße und der Netzkonstellation lassen sich alle Lokalisierungsmethoden in Geosensornetzwerken, abhängig von der erreichten Genauigkeit und der Komplexität des jeweiligen Algorithmus, in exakte und approximative Methoden unterteilen (Abbildung 4.2). In der



Abbildung 4.2: Klassifizierung üblicher Lokalisierungstechniken in drahtlosen Sensornetzwerken.

Literatur werden diese auch als "Fine-grained Localization" (*FGL*) – die Bezeichnung geht auf den gleichnamigen, von Savvides et al. entwickelten Lokalisierungsalgorithmus zurück [SHS01] – oder "Coarse-grained Localization" (*CGL*) bezeichnet. Daneben gibt es weitere Methoden, die i. d. R. auf komplett anderen Annahmen beruhen. In dieser Arbeit wird, wie auch in der Literatur üblich, der Begriff exakt (feinkörnig oder "fine-grained") als Abgrenzung zu approximativ (grobkörnig oder "coarse-grained") in Verbindung mit den einzelnen Lokalisierungsmethoden verwendet. Dabei sollte beachtet werden,

dass alle Messungen fehlerbehaftet sind und diese Klassifizierung ausschließlich der Unterscheidung zu den approximativen Methoden hinsichtlich der Genauigkeit, Robustheit und Zuverlässigkeit der Positionsbestimmung dient.

Eine Auflistung aktueller Lokalisierungssysteme gibt Abbildung 4.3, auf die in der weiteren Arbeit nicht näher eingegangen werden soll, da im Wesentlichen algorithmische Fragestellungen im Vordergrund stehen.



Abbildung 4.3: Aktuelle Lokalisierungssysteme [Mau10].

Im Folgenden wird eine Auswahl aktueller Lokalisierungsalgorithmen vorgestellt und der jeweiligen Methode (approximativ oder exakt) zugeteilt.

4.2 Entfernungsunabhängige Verfahren

In der Literatur existieren viele approximative Lokalisierungsmethoden, von denen an dieser Stelle einige wenige skizziert werden sollen. Diese Methoden sind ressourceneffizient, resultieren aber in größeren Positionsfehlern, wobei die meisten approximativen Techniken auf die Benutzung von Distanzmessungen verzichten [SN05, MF09].

"Centroid Localization"-Algorithmus

In [Bul00] wurde der "Centroid Localization"–Algorithmus (*CL*) vorgeschlagen (Abbildung 4.4 (a)). In der ersten Phase senden alle Beacons ihre Positionen $\tilde{B}(\tilde{x}, \tilde{y})$ und deren dazugehörige Netzwerkadresse zu allen Sensorknoten in Sendereichweite. In der zweiten Phase berechnen alle Sensorknoten ihre

eigene Position durch Schwerpunktbestimmung aus den *n*-Positionen aller empfangenen Signale der Beacons:

$$\hat{P}_N(\hat{x}_N; \hat{y}_N) = \frac{1}{n} \sum_{j=1}^n \tilde{B}_j(\tilde{x}_j, \tilde{y}_j) \qquad j = 1, 2, 3, ..., n.$$
(4.1)

Dieser Algorithmus zeichnet sich durch den relativ geringen Kommunikationsaufwand – es müssen lediglich die Beaconpositionen und die Netzwerkadressen von den Knoten empfangen werden – und den geringen Berechnungsaufwand aus, resultiert aber auch in relativ großen Positionsfehlern von ca. 18 % [Bul00].



Abbildung 4.4: (a) "Centroid Localization"–Algorithmus nach [Bul00]. (b) "Weighted Centroid Localization"–Algorithmus (nach [BRT05a]).

"Weighted Centroid Localization"-Algorithmus

Der *CL*–Algorithmus hat den Nachteil, dass er Entfernungsabhängigkeiten außer Acht lässt. In Blumenthal et al. wurde der *CL* zum "Weighted Centroid Localization"–Algorithmus (*WCL*) (Abbildung 4.4 (b)) weiterentwickelt und um distanzabhängige Gewichte erweitert [BRT05a]. Aufgrund der entfernungsabhängigen Fehler werden hier die näheren Beacons höher gewichtet als die weiter entfernten. Diese distanzabhängige Gewichtung verschiebt die ermittelte Position aus der Sensorfeldmitte heraus hin zu den näheren Beacons.

Die Schwerpunktgleichung 4.1 wird durch das Gewicht w_i als Funktion der Distanz wie folgt erweitert:

$$\hat{P}_{N}(\hat{x}_{N};\hat{y}_{N}) = \frac{\sum(w_{j} \cdot \tilde{B}_{j}(\tilde{x}_{j},\tilde{y}_{j}))}{\sum w_{j}} \qquad j = 1, 2, 3, ..., n$$
(4.2)

mit $w_j = \frac{1}{d_{Nj}^8}$ und der Potenz *g*, auf die im Zuge dieser Arbeit detaillierter eingegangen werden wird und die den Pfadverlustexponenten β , den Kapitel 5.2.2 näher beschreibt, wiedergibt. Der Algorithmus läuft ähnlich dem *CL*–Algorithmus ab. In der ersten Phase empfangen die Knoten die Positionen der Beacons in Empfangsreichweite sowie deren Netzwerkadressen. Zudem wird die Entfernung nach einem geeigneten Verfahren nach Kapitel 4.4 bestimmt. In Phase zwei ermittelt jeder Sensorknoten seine eigene Position nach der gewichteten Schwerpunktmethode selbstständig. *WCL* erreicht im Vergleich zu *CL* eine gesteigerte Genauigkeit von ca. 13 %. Eine detaillierte Genauigkeitsbetrachtung erfolgt in Kapitel 7. Da beim *WCL* gerade in den Randbereichen des Sensorfeldes der Lokalisierungsfehler maximal wird, erweitern Behnke et al. den *WCL* um s. g. virtuelle Beacons. Der Positionsfehler des *WCL* konnte damit um bis zu 27 % verbessert werden [BSG⁺09].

Beide Algorithmen, CL und WCL, sind den approximativen Algorithmen zuzuordnen.

"Approximate Point In Triangulation"-Algorithmus

Der "Approximate Point In Triangulation"–Algorithmus (*APIT*) verzichtet komplett auf Distanzen [HHB⁺03]. Abbildung 4.5 zeigt hierzu den schematischen Aufbau. Zu Beginn werden aus allen Beaconpositio-



Abbildung 4.5: APIT–Algorithmus nach [HHB⁺03]. (a) Flächenüberlagerung bei APIT; (b) theoretisch erfolgreicher PIT–Test; (c) praktisch erfolgreicher PIT–Test.

nen Dreiecke gebildet. Dann bestimmt jeder Knoten durch einen "Point in Triangulation"–Test (*PIT*– Test), innerhalb welchen Dreiecks er sich befindet. Eine nachfolgende Schnittmengenbildung mit allen betreffenden Dreiecken führt zu einem Flächenstück, dessen Schwerpunkt im letzten Schritt als wahrscheinlichste Position des unbekannten Sensorknotens ermittelt wird.

Abbildung 4.5 (b) zeigt einen theoretisch erfolgreichen PIT–Test. Wenn sich der Sensorknoten in irgendeine Richtung um einen geringen Wert dx bewegt, muss eine Richtung existieren, um von allen Beaconpositionen gleichzeitig weiter weg oder an alle näher heran zu kommen.

In Abbildung 4.5 (c) ist ein praktisch erfolgreicher *PIT*–Test wiedergegeben. Dabei wird Signalempfangsstärke zu vier Nachbarn bestimmt und getestet, ob sich einer der Nachbarn gleichzeitig weiter weg oder näher zu allen drei Dreieckspunkten befindet.

Da diesem Algorithmus ausschließlich Nachbarschaftsbeziehungen zugrundeliegen, wird der *APIT* den approximativen Algorithmen zugeordnet.

Szenenanalyse

Ein weiterer approximativer Algorithmus ist die "Szenenanalyse" (*SA*). Da ein in dieser Arbeit vorgestellter Algorithmus im weiteren Sinne ebenfalls auf der Szenenanalyse beruht, soll diese hier kurz skizziert werden. Der Signalausbreitungscharakter von Funksignalen mit stationären Beacons besitzt in statischen Umgebungen eine bestimmte Signatur. Diese statische Umgebung wird als streng vorausgesetzt und umfasst auch mögliche mobile Objekte wie z. B. Personen. Sind diese Voraussetzungen erfüllt, kann diese Signatur bei der Szenenanalyse eingesetzt werden. Vor der Lokalisierung werden die Signaleigenschaften an mehreren Messpunkten aufgenommen und in einer s. g. Signalkarte abgespeichert. Diese Signalkarte besteht damit aus einer Vielzahl von Messungen (z. B. Signalempfangsstärken) an den unterschiedlichsten Positionen, welche auf den Sensorknoten abgespeichert wird. Dieser vergleicht die Signaturen der empfangenen Signale mit der Signalkarte. Die Datensätze, die am besten zu den gemessenen Werten passen, sind die wahrscheinlichsten Positionen. Diese Art der Lokalisierung wird in der Literatur auch als "Pattern Matching Technique" (*PMT*), "Pattern Recognition" (*PR*) oder "Fingerprinting" (*FP*) bezeichnet [BP00].

"Bounding Box"–Verfahren

Die Lokalisierung in [SS02] erfolgt durch die Bildung von Schnittflächen, die sich durch Überlappung zweier Sendereichweiten bilden. Damit wird die Region eingegrenzt, in der sich die zu lokalisierenden Sensorknoten befinden. Eine Erweiterung dazu ist der oben aufgeführte *APIT*–Algorithmus.

"Nearest Beacon"–Algorithmus

Ein weiterer simpler Ansatz ist der von Born et al. eingeführte "Nearest Beacon"–Ansatz (*NB*), der die nächsten Beaconkoordinaten als Sensorposition verwendet [BRBT06]. Da es sehr fehlerhafte Positionen liefert, wurde dieses Verfahren allerdings ausschließlich zur Startwertbestimmung für konvergente, feinkörnige und vor allem iterative Lokalisierungsalgorithmen eingeführt.

4.3 Entfernungsabhängige Verfahren

Die mathematischen Grundlagen der Neupunktbestimmung aufgrund von überbestimmten Beobachtungen (Multilateration) wurde im letzten Kapitel ausgiebig diskutiert (siehe Kapitel 3.1). Die gängigste Methode zur Lösung der dabei entstehenden nichtlinearen Gleichungssysteme ist die Anwendung der "Methode der kleinsten Quadrate" (MdkQ). Die MdkQ liefert die besten, linearen, unverzerrten Schätzer (best linear unbiased estimators - *BLUE*), ist in ihrer Berechnung aber sehr komplex, ressourcenintensiv und daher eigentlich unbrauchbar für den Einsatz auf den sehr ressourcenarmen Sensorknoten. Zudem müssen die nichtlinearen Gleichungssysteme für die Lösung mit der MdkQ in eine lineare Form überführt werden. Geschieht diese Linearisierung nach der Newton–Methode, ist die Kenntnis von Näherungskoordinaten Voraussetzung. Einen Ansatz zur Verwendung approximativer Lokalisierungsmethoden zur Näherungskoordinatenbestimmung im Hinblick auf deren Genauigkeit geben Born et al. [BRBT06].

"Fine Grained Localization"–Algorithmus

Der "Fine-grained Localization"–Algorithmus (*FGL*) wurde von Savvides et al. formuliert und ist einer der meist zitierten exakten Lokalisierungsalgorithmen [SHS01]. Dieser benötigt mindestens drei Beacons und läuft auf jedem Sensorknoten ab. *FGL* basiert auf der Methode der kleinsten Quadrate und einem überbestimmten Gleichungssystem bei mehr als drei Beacons. Jeder Sensorknoten $\hat{P}_{N_i}(\hat{x}_{N_i}, \hat{y}_{N_i})$ bestimmt die Distanz zu allen Beacons $\tilde{B}_j(\tilde{x}_j; \tilde{y}_j)$ in Empfangsreichweite. Der Distanzfehler *f* errechnet sich zu:

$$f_i(\hat{x}_{N_i}, \hat{y}_{N_i}, v) = v t_{ij} - \sqrt{(\tilde{x}_j - \hat{x}_{N_i})^2 + (\tilde{y}_j - \hat{y}_{N_j})^2},$$
(4.3)

mit v der Ausbreitungsgeschwindigkeit eines Ultraschallsignals in Luft und t_{ij} der Zeit, die das Signal von Sensorknoten i zu Beacon j benötigt. Mit mindestens drei Beacons wird ein nichtlineares Gleichungssystem formuliert, das nach Linearisieren, Quadrieren und Umformen der Terme zu:

$$-\tilde{x}_j^2 - \tilde{y}_j^2 = \hat{x}_{N_i}^2 + \hat{y}_{N_i}^2 + \hat{x}_{N_i} - 2\tilde{x}_j + \hat{y}_{N_i} - 2\tilde{y}_j - v^2 t_{ij}^2$$
(4.4)

führt. Subtrahieren der letzten Gleichung i = h von den verbleibenden ergibt:

$$-\tilde{x}_{j}^{2} - \tilde{y}_{j}^{2} + \hat{x}_{N_{h}}^{2} + \hat{y}_{N_{h}}^{2} = 2\tilde{x}_{j}(\hat{x}_{h} - \tilde{x}_{j}) + 2\tilde{y}_{j}(\hat{y}_{h} - \tilde{y}_{j}) + v^{2}(t_{jh}^{2} - t_{ij}^{2}).$$
(4.5)

Diese Gleichung kann in die Form Ax = b überführt und mit der Methode der kleinsten Quadrate nach $x = (A^T A)^{-1} A^T b$ gelöst werden. Dieser Grundalgorithmus wird in der Literatur als atomare Multilateration bezeichnet und dient in Kapitel 6 als Vergleichsalgorithmus.

In weiteren Schritten wurde die atomare Multilateration zur "iterativen" oder "kollaborativen Multilateration" von Savvides et al. weiterentwickelt. Bei der iterativen Multilateration (*iMUL*) geht die atomare Multilateration von einem Sensorknoten aus und breitet sich dann über die Nachbarknoten aus. Dabei wird jeder Knoten, der seine Position ermittelt hat, selbst zu einem Beacon und sendet seine Koordinaten zu seinen Nachbarn. Die Genauigkeit der iterativen Multilateration hängt stark von der Genauigkeit der geschätzten Position der Sensorknoten ab, da sich der Lokalisierungsfehler über die Sensorknoten fortpflanzt.

"Ad Hoc Positioning System"–Algorithmus

In einem weiteren Algorithmus "Ad Hoc Positioning System" (*APS*) fluten die Beacons ihre Koordinaten in das Netzwerk. Die Distanzbestimmung erfolgt über das "Distance Vector-Hop"–Verfahren (*DV-Hop*) (Abschnitt 4.4). Dazu werden die minimalen Hop–Zähler abgespeichert und anschließend einer Trilateration zugeführt [NN01, NN03].

Savarese et al. erweitern den *APS*–Algorithmus um einen weiteren Schritt, einer so genannten Refinementphase [SRL02]. Die Positionen aus der ersten Phase, der s.g. "Hop–Terrainphase", werden an alle Nachbarn weitergesendet, welche sich anhand dieser eine neue Position berechnen. So genannte "Confidence–Werte" sollen die Fehlerfortpflanzung verringern, indem sie mehr oder weniger stark in die Berechnung einfließen.

Goldenberg et al. beschreiben einen iterativen Algorithmus, der auf Positionsbestimmung durch Trilateration beruht [GBY⁺06]. Dabei werden bei mehr als drei Beacons mehr als drei mögliche Positionen berechnet, miteinander kombiniert und eine endgültige Position geschätzt. Anfänglich werden Knoten dreier Kategorien unterschieden. Das sind:

- 1. Knoten mit genau einer Position,
- 2. Knoten mit einer Reihe von Schätzpositionen und
- 3. Knoten ohne Position.

In dem ersten Schritt besitzen ausschließlich die Beacons eine oder mehrere Positionen. Hat ein Sensorknoten Kontakt zu mindestens zwei Beacons, berechnet er daraus eine Menge von Schätzpositionen. Im nächsten Schritt bestimmen die Sensorknoten ihre Position durch Funkkontakt mit dem Beacon oder Sensorknoten in Reichweite, die über Positionsinformationen aus dem ersten Schritt verfügen und ermitteln so ihre Koordinaten. Dieser iterative Prozess wird fortgeführt, bis alle Sensorknoten ihre Position, soweit möglich, bestimmt haben.

"Assumption Based Coordinates"-Algorithmus

Der "Assumption Based Coordinates"–Algorithmus (*ABC*) beginnt die Positionsbestimmung mit einem initialen Sensorknoten S_0 , dessen Position in den Ursprung eines lokalen Koordinatensystems $S_0(0;0)$ gelegt wird [SRB01, SRL02]. Die nächsten Sensorknoten S_1 und S_2 bestimmen in einem weiteren Schritt die Distanzen \hat{d}_{01} und \hat{d}_{02} zum ersten Knoten und die Distanz \hat{d}_{12} zueinander. Die Position des zweiten Sensorknotens wird dann dadurch berechnet, dass eine Koordinatenachse durch den ersten und den zweiten Knoten gelegt wird. Es ergibt sich die Position $\hat{S}_1(\hat{d}_{01};0)$, wodurch ein lokales Koordinatensystem seine Koordinatenachse durch den ersten und den zweiten Knoten gelegt wird. Es ergibt sich die Position $\hat{S}_1(\hat{d}_{01};0)$, wodurch ein lokales Koordinatensystem definiert ist. Die Position des dritten Knotens ergibt sich damit zu $\hat{S}_2\left(\frac{\hat{d}_{01}^2 + \hat{d}_{02}^2 + \hat{d}_{12}^2}{2\hat{d}_{01}^2}; \sqrt{\hat{d}_{02}^2 - \hat{S}_2(\hat{x}_2^2)}\right)$. Alle weiteren Sensorknoten positionieren sich danach durch Trilateration mit den ersten drei Punkten als Beacons. Dieser Vorgang wird für das ganze Netzwerk iterativ über die neu hinzukommenden Sensorknoten berechnet. Soll das ganze Sensornetzwerk in einem übergeordneten Koordinatensystem vorliegen, ist eine abschließende Transformation durchzuführen, wobei dafür identische Punkte in beiden Systemen vorhanden sein müssen. Ähnliche Algorithmen werden in [PBDT03, CHH02] verwendet. Es ist anzumerken, dass sich bei dieser Technik große Positionsfehler mit ansteigender Netzwerkgröße ergeben.

Robuste Quadrilaterale

Bei [MLRT04] werden robuste Quadrilaterale gebildet, wobei es sich um Subgraphen handelt. Jeder von vier Sensorknoten hat eine Verbindung zu den verbleibenden drei Nachbarn. Dies ermöglicht immer eine Trilateration zur Positionsbestimmung. Barbeau et al. diskutieren weitere Möglichkeiten, um bei der Bildung robuster Quadrilaterale unverbundene Sensorknoten bei der Trilateration auszuschließen [BKK⁺04]. Mautz et al. verbessern Moore's Algorithmus um eine erweiterte Multilateration. Weiterhin erfolgt eine Erweiterung auf eine 3D–Lokalisierung [MOBK07].

"Multidimensionale Skalierung"

In Ji, Yi und Ahmed basieren die Lokalisierung auf der "Multidimensionalen Skalierung" (*MDS*), um mit stark fehlerhaften Distanzmessungen und in inhomogenen Umgebungen eine Positionsschätzung durchführen zu können [JZ04, YR04, ASS05]. Dabei beginnt die *MDS* mit einer Distanzanzahl zwischen unbekannten Sensorknoten und findet die optimale Konfiguration der Punkte für die nachfolgende Trilateration.

"Distributed Least Squares"-Algorithmus

In dem von Reichenbach et al. entwickelten feinkörnigen "Distributed Least Squares"–Algorithmus *DLS* werden die Euklidischen Distanzen benutzt, um eine Position nach dem Bogenschnittverfahren (vgl. Abschnitt 3.1) zu ermitteln [RBTB06, Rei07, BRBT08]. Der Vorteil bei diesem Algorithmus ist die verteilte Berechnung. Der komplexe Teil der Berechnung wird dabei von der Senke übernommen. Auf den Sensorknoten muss lediglich eine simple Nachberechnung durchgeführt werden. Dieser *DLS*–

Grundalgorithmus dient in Kapitel 6 als Vergleichsalgorithmus und wird in Kapitel 7 näher erläutert sowie hinsichtlich seiner Genauigkeitseigenschaften untersucht.

In Behnke et al. wird der *DLS*–Algorithmus auf große Netzwerke ausgedehnt und als "scalable Distributed Least Squares"–Algorithmus (*sDLS*) formuliert. [BSLT09, BST10b, BST10a]. Die Vorberechnung wird dabei nur auf Beacons in Empfangsreichweite der Sensorknoten begrenzt.

"iterative Distributed Least Squares"-Algorithmus

Reichenbach entwickelt den ursprünglichen *DLS*–Algorithmus zum "iterative Distributed Least Squares"– Algorithmus *iDLS* weiter. Ziel hierbei ist es, wie bei dem *iMUL*–Algorithmus, die berechneten Positionen der Nachbarknoten zu benutzen, um die Positionsschätzung des einzelnen Sensorknotens zu verbessern. Dabei werden die Nachbarknoten ebenfalls zu Beacons und deren Position durch Matrixupgrade der QR-Faktorisierung der Berechnung hinzugefügt. Dabei kann die Genauigkeit der Positionsschätzung um Faktor 1,5 verbessert werden [RT07].

"mobile Distributed Least Squares"-Algorithmus

In [Rei07, RLT08] wurde der *DLS*–Algorithmus für den Einsatz in mobilen Sensornetzwerken zum "mobile Distributed Least Squares"–Algorithmus (*mDLS*) erweitert. Dabei werden die Positionen der Beacons in der QR-Faktorisierung durch Matrixupdating aktualisiert und neue Distanzmessungen durchgeführt. Damit ist es möglich, in eingeschränktem Maße Sensorknoten in dynamischen Umgebungen zu positionieren und die Positionen zu aktualisieren. Da die Positionsschätzung jedoch relativ schnell zu sehr ungenauen Ergebnissen führt, ist eine komplette Neuberechnung des Sensornetzes unabdingbar.

Weitere distanzbasierte Lokalisierungsverfahren

Kwon et al. präsentieren eine verteilte Lösung, die ebenfalls auf der Methode der kleinsten Quadrate beruht und Fehler in der Distanzmessung mittels Ultraschall reduziert [KMS⁺05]. Ahmed et al. schlugen einen neuen Ansatz vor, welcher die Vorteile von absoluter und relativer Lokalisierung vereint [ASS05]. Weiterhin entwickelten Kabarar et al. ein energiearmes System zur Positionsbestimmung mit Hilfe der Methode der kleinsten Quadrate zur Integration auf einzelnen Sensorknoten [Kab05]. Lieckfeldt verfolgt in [LYT08, Lie10] einen Ansatz zur optimalen Beaconauswahl für die Trilateration auf Basis der Cramer-Rao Untergrenze (engl. Cramer-Rao Lower Bound (*CRLB*)).

Einen generellen Überblick über verteilte Systeme zur Positionsbestimmung geben Langendoen und Reijersin sowie Mao et al. [LR03, MF09].

4.4 Beobachtungselemente zur Lokalisierung

In diesem Abschnitt sollen mögliche Beobachtungselemente betrachtet werden, aufgrund derer eine Lokalisierung in Sensornetzwerken erfolgen kann. Bei der Distanz- oder Winkelbestimmung zwischen Sensorknoten und Beacons kommen eine Reihe möglicher Techniken infrage. Im Prinzip beruhen alle Verfahren auf der Analyse der empfangenen Signale. Abbildung 4.6 gibt einen Überblick über mögliche Beobachtungselemente in drahtlosen Sensornetzwerken.



Abbildung 4.6: Mögliche Beobachtungselemente zur Positionsbestimmung (nach [Rei07]).

4.4.1 Distanzbestimmung

Fast alle feinkörnigen Lokalisierungsalgorithmen (Abbildung 4.2) benutzen Distanzschätzungen zur Positionsbestimmung. Aus diesem Grund kommt ihnen auch eine große Bedeutung für den Einsatz in drahtlosen Sensornetzwerken zu.

Signalempfangsstärkemessung

Alle Sensorknoten in einem Sensornetzwerk besitzen eine Kommunikationseinheit, um die gesammelten Daten senden oder um mit den anderen Knoten kommunizieren zu können (vgl. Kapitel 2.3.3). Diese Einheit kann benutzt werden, um über den Signalcharakter einer Funkwelle auf eine Distanz zwischen Sender und Empfänger zu schließen. Die Messung der Sendeempfangsstärke ist das am meisten verbreitete Verfahren. Der große Vorteil dieses Verfahrens ist in der Tatsache begründet, dass keine zusätzliche Hardware auf den Knoten verbaut werden muss (die Empfangsstärke wird einfach mitgemessen). Als nachteilig haben sich die relativ hohen Ungenauigkeiten bei den Distanzschätzungen ergeben.

In der Kommunikationseinheit wird die analoge Empfangsleistung in einen digitalen Signalempfangsstärkewert ("Received Signal Strength Indicator" (*RSSI*)) umgewandelt. Über diesen RSS–Wert kann im Idealfall eine Distanz errechnet werden (Abbildung 4.7 (a)).

Allerdings ist in der Realität ein abweichender Verlauf der RSSI-Kurve zu verzeichnen (Abbildung 4.7 (b)). Diese Unterschiede sind vor allem auf unterschiedliche Hardwareeigenschaften, die Ausrichtung der Module und Reflexionen an Objekten bzw. Wänden zurückzuführen. Hinsichtlich der Zuordnung einer Distanz zu einem RSS–Wert führen Ausreißer zu ungewollten Mehrdeutigkeiten. Die Ergebnisse werden detailliert in [RKT06] diskutiert. Auf den Ausbreitungskanal beeinflussende Faktoren und



Abbildung 4.7: RSSI-Verfahren: Distanzbestimmung mittels des RSS-Indikators (a) im Idealfall [AKT09] und (b) eines Chipcon CC1010–Moduls bei ansteigender Distanz zwischen einem Sender und einem Empfänger in einem Außenbereich ohne Hindernisse in der Realität [RKT06].

Ausbreitungmodelle wird detailliert in Kapitel 5.2.1 eingegangen.

Nachbarschaftsbeziehungen

Nachbarschaftsbeziehungen basieren auf folgender Annahme. Empfängt Knoten S_1 das Signal eines anderen Knotens S_2 , so ist S_1 in Sendereichweite von S_2 . Damit ist weiterhin S_2 der Nachbar von S_1 . Unter der Annahme, dass alle Sendereichweiten ideal konzentrisch sind, liegt S_1 innerhalb des Senderadius von S_2 . Nachteil ist hier, dass die Wellenausbreitung nie ideal konzentrisch erfolgt [ZHKS04], was zu einem erheblichen Einfluss auf die Genauigkeit näherungsbasierter Verfahren führt.

"Minimal Transmission Power"–Verfahren

Blumenthal et. al entwickelten das "Minimal Transmission Power"–Verfahren (*MTP*) auf Basis von Nachbarschaftsbeziehungen [BTB⁺06]. Auf der Kommunikationseinheit ist die Sendeleistung einstellbar, was sich *MTP* zunutze macht. Dabei stellt der sendende Knoten die geringste Sendeleistung ein, welche zusammen mit seiner Netzwerkadresse an die Nachbarknoten gesendet wird. Im nächsten Schritt erfolgt eine Erhöhung der Sendeleistung auf den nächst höheren Wert, welcher wiederum mit der Netzwerkadresse versendet wird. Dieser Prozess wird solange wiederholt, bis die höchst mögliche Sendeleistung erreicht ist. Die empfangenden Knoten speichern dabei jeweils nur das erste sie erreichende Paket ab und ignorieren die folgenden. Eine Distanz zu dem sendenden Knoten kann der Empfänger über den Zusammenhang Sendeleistung–Sendereichweite abschätzen.

Limitierende Faktoren sind im Wesentlichen die Auflösung der Sendeleistung in dem Kommunikationsmodul, d. h., in welcher Schrittweite das ankommende Signal aufgelöst werden kann. In einer weiterführenden Arbeit wurde gezeigt, dass das Verfahren Vorteile gegenüber Signalempfangsstärkemessungen hat [BRT06].

4.4.1.1 Laufzeitmessungen

Time of Arrival

Weitere Verfahren zur Distanzbestimmung messen die Laufzeit der Signale von Sender zum Empfänger. Dieses Verfahren wird als "Time of Arrival" (*ToA*) bezeichnet und ist bereits erfolgreich beim "Global Positioning System" (*GPS*) im Einsatz. Hierbei bestimmt der Sensorknoten die Zeitspanne oder Laufzeit Δt , in der das Signal die gesuchte Strecke durchläuft. Mit dieser Zeitdifferenz und der Signalausbreitungsgeschwindigkeit *c* ergibt sich für die Distanz:

$$d = v \cdot \Delta t \quad \text{mit} \quad v = c \tag{4.6}$$

Die Ausbreitungsgeschwindigkeit für elektromagnetische Wellen ist mit der Lichtgeschwindigkeit *c* jedoch sehr hoch (~ $300.000 \frac{km}{s}$ im Vakuum). Somit haben kleine Fehler in der Zeitmessung große Auswirkungen auf die Genauigkeit der Streckenberechnung. So würde sich bei einer Abweichung von 1 *ns* ein Distanzfehler von 0,3 *m* ergeben. Im Falle von *GPS* sind die Satelliten mit sehr teuren Atomuhren ausgerüstet, die eine sehr hohe Genauigkeit besitzen (Abweichung von einer Milliardstel Sekunde pro Monat). Außerdem ist die Entfernung Satellit–Empfänger so groß, dass die Signale $\frac{1}{15}s$ zur Erde benötigen. Die Knoten in einem Geosensornetzwerk haben dem gegenüber allerdings einen sehr geringen Abstand von nur einigen Zentimetern bis Metern. Außerdem widerspricht der Einsatz dieser extrem teuren und schweren Uhren dem Prinzip der Kostenersparnis und der Miniaturisierung in drahtlosen Sensornetzwerken.

Time Difference of Arrival

Bei dem so genannten "Time Difference of Arrival"–Verfahren mit unterschiedlichen Signalen (*TDoA*) werden zwei Signale mit unterschiedlichen Geschwindigkeiten verwendet (z. B. Radio- und Schallwellen). Die Messung erfolgt, indem beide Signale zur selben Zeit ausgesendet werden. Aufgrund der großen Laufzeitdifferenz der beiden Signale verwendet der Empfänger den Empfangszeitpunkt der Radiowelle als Startzeit t_1 und den Empfangszeitpunkt der Schallwelle als Stoppzeit t_2 . Durch Anwenden der Gleichung 4.6 mit $\Delta t = t_2 - t_1$ und v als Schallgeschwindigkeit erhält man eine absolute Distanz. Ein Problem bei der Benutzung von Schallwellen ist die hohe Empfindlichkeit gegenüber äußeren Störeinflüssen. Ein weiterer Nachteil ist der notwendige Einsatz zusätzlicher Hardware für die unterschiedlichen Messsignale.

Round Trip Time

Ein in der Netzwerktechnik häufig benutztes Verfahren ist das "Round Trip Time"–Verfahren (*RTT*). Hierbei wird die Zeit gemessen, die das Signal zum Empfänger und wieder zurück benötigt. Der Sender misst die Ankunftszeit und ermittelt die Distanz nach Gleichung 4.6 mit $\Delta t = \frac{t_{gem}}{2}$, wobei t_{gem} die gemessene Zeit ist, die das Signal zum Empfänger und wieder zurück benötigt. Dieses Verfahren der Messung hat gegenüber der (nur in eine Richtung gemessenen) Latenzzeit den Vorteil, dass keine aufwändige Zeitsynchronisation der beiden beteiligten Endgeräte benötigt wird. Nachteilig ist bei der *RTT*-Messung, dass bei vielen Netzen unsymmetrische Verzögerungszeiten auftreten können, und die halbe Laufzeit nicht unbedingt eine gute Näherung für die Verzögerungszeit in eine Richtung liefert. Außerdem müssen die Sensorknoten die Datenpakete zu den Beacons zurücksenden, was einen zusätzlichen Energieverbrauch nach sich ziehen würde. Unter Umständen müssten diese auch mit zusätzlicher Hardware ("Time-to-Digital-Converter" (*TDC*)) ausgestattet werden. Das Ziel in energielimitierten, drahtlosen Geosensornetzwerken soll aber darin bestehen, den Hardware- und Energieaufwand für die Lokalisierung so gering wie möglich zu halten, weswegen dieses Verfahren für diese Anwendungen eher ungeeignet ist.

Ein Problem aller Laufzeitmessungen ist jedoch die Abhängigkeit der Ausbreitungsgeschwindigkeit vom Medium, welches das Signal durchläuft. Bei unterschiedlichen Medien entlang des Ausbreitungspfades ergeben sich unterschiedliche Geschwindigkeiten, die nur schwierig zu modellieren sind.

Phasenvergleich

Das Phasenvergleichsverfahren arbeitet nach folgendem Prinzip, das u. a. in [Kah05] detailliert erläutert wird (Abbildung 4.8). Der Sender erzeugt ein sinusförmiges Messsignal, welches in Abhängigkeit der Länge des durchlaufenen Weges eine Phasenverschiebung erfährt. Diese Phasendifferenz kann mit einem Phasenmesser gemessen und eine eindeutige Distanz zwischen Sender und Empfänger ermittelt werden, wenn sie nicht größer als die halbe Wellenlänge des Signals ist. Phasenmesser besitzen nur eine begrenzte Genauigkeit von $\frac{\lambda}{4000}$ bis $\frac{\lambda}{8.000}$, die jedoch höher ist als die der vorigen Verfahren [Kah05]. Bei einer Wellenlänge von $\lambda = 20 m$ und einer Genauigkeit des Phasenmessers von $\frac{\lambda}{4.000}$ ergibt sich eine Auflösung für die Genauigkeit der Strecke von 5 mm. Als Vorteile des Phasenvergleichsverfahrens wären folgende Eigenschaften zu benennen:

- 1. es ist sehr ausgereift und
- 2. unempfindlich gegen kurzzeitige Messunterbrechungen, z. B. durch vorbeifahrende Fahrzeuge, sich im Zielstrahl bewegende Blätter oder dergleichen:

Dem stehen folgende Nachteile gegenüber:

- das Verfahren ist ausschlie
 ßlich anwendbar, wenn eine direkte Sichtverbindung (engl. "Line-of-Sight" (LoS)) zwischen Sender und Empfänger besteht,
- 2. die Notwendigkeit mehrerer Maßstäbe zur Bestimmung einer Strecke und dadurch bedingte längere Messzeiten sowie
- 3. aufwendige Optik und hoher Energieverbrauch.



Abbildung 4.8: Phasenvergleichsverfahren: Distanzbestimmung über Phasendifferenz $\Delta\lambda$ mit λ –Wellenlänge, D–Distanz [Kah05].

4.4.1.2 Hop-basierte Verfahren

Wie bereits in Kapitel 2 erwähnt wurde, bestehen Geosensornetzwerke aus einer großen Anzahl von Knoten. Diese hohe Knotendichte kann nun benutzt werden, um Rückschlüsse auf absolute Distanzen zu ziehen (Abbildung 4.9). Über Multihopping soll ein Paket auf dem kürzesten Weg von dem Knoten K_1



Abbildung 4.9: Hop-basiertes Verfahren: Distanzbestimmung über Zählung der benötigten "Hops" zwischen Startund Zielknoten (nach [Rei07]).

über K_2 , K_3 und K_4 nach K_5 gesendet werden. Ein integrierter Hop-Zähler registriert dabei die Anzahl der ausgeführten "Sprünge". Dieser Zähler kann auf dem Zielknoten ausgelesen und in eine Distanz umgewandelt werden, wenn folgende Voraussetzungen gegeben sind:

- die Zwischenknoten auf dem k
 ürzesten Weg zwischen K₁ und K₅ sollten m
 öglichst nah an und am besten auf der direkten Strecke liegen,
- alle Knoten sollten möglichst gleichverteilt sein,
- die Knotendichte sollte möglichst hoch sein, und
- es dürfen keine Hindernisse zwischen Sender und Empfänger existieren, welche weiträumig umgangen und somit den Hop-Zähler unnötig erhöhen würden.

Die Güte der Distanzmessung wird durch die Einhaltung der genannten Bedingungen beeinflusst. Per Routingprotokoll kann der kürzeste Weg, als Summe des durchschnittlichen Knotenabstandes multipliziert mit der Hop-Anzahl, ermittelt werden. Für die Schätzung des durchschnittlichen Knotenabstandes stehen unterschiedliche Verfahren zur Verfügung:

"Distance-Vector-Hop" (DV-Hop)

Dieses Verfahren beruht auf einem sehr einfachen Prinzip. Die Beacons sammeln die kleinsten Hop-Zähler zu allen anderen Beacons im Netzwerk. Da die Beacons über ein Lokalisationsbewusstsein verfügen, können die Hop-Zähler mit absoluten Distanzen in Zusammenhang gebracht und eine mittlere Distanz pro Sprung berechnet werden. Diese durchschnittliche Distanz wird an die Sensorknoten übermittelt, welche durch Multiplikation mit den gezählten Hops zu anderen Knoten eine absolute Strecke berechnen können. Diese Verfahren findet Anwendung in [ZXL⁺09].

Geometrische Beziehungen

Bei gleichverteilten Knoten, bekannter Knotendichte im Netzwerk und bekannter Sensorfeldfläche kann ein durchschnittlicher Knotenabstand nach [KS78] berechnet werden.

"Average distance"

Im Prinzip ist dieses Verfahren eine Abwandlung der DV-Hop–Methode. Ein zentraler Knoten sammelt alle Distanz-Hops zu den anderen Beacons und mittelt diese. Der weitere Verlauf folgt dem des DV-Hop–Verfahrens.

4.4.2 Winkelmessungen

Zur Lokalisierung können auch Winkelbeobachtungen genutzt werden. Hierbei versucht der Sensor, den Einfallswinkel der höchsten Signalempfangsstärke eines ankommenden Signals zu ermitteln "Angle of Arrival" (*AoA*) [NL02]. Ein interessanter Prototyp hierfür ist der Medusa-Knoten [Net10].

Allerdings ist ein erhöhter Hardwareaufwand nicht zu umgehen. Im Falle des Medusa-Knotens sind mehrere Antennen nötig, um 360° abzudecken. Ein weiterer Nachteil ist bei der Ausbringung der Sensorknoten unvermeidbar: bei einer zufälligen Verteilung (z. B. Ausstreuung per Flugzeug) kann nicht garantiert werden, dass der Sensorknoten richtig steht und nicht "umkippt". Des Weiteren sind die *AoA*–Verfahren durch Rauschen oder Mehrwegeausbreitung beeinflusst. Aus diesen Gründen spielen sie eine noch immer untergeordnete Rolle als Beobachtungselement für die Lokalisierung in drahtlosen Sensornetzwerken.

4.4.3 Sonstige Verfahren

"Inertial Measurement Unit"

Ein Inertialmesssystem (engl. "Inertial Measurement Unit" (*IMU*)) besteht aus Beschleunigungssensoren (Accelerometer) und Drehratensensoren (Gyroskop) und ist ein wichtiger Baustein in vielen integrierten Positionierungssystemen. Auch Sensorknoten könnten mit IMUs zur Positionsbestimmung in dynamischen Umgebungen ausgerüstet werden, indem aus gemessenen Beschleunigungen Geschwindigkeiten und aus gemessenen Winkeln Richtungen berechnet werden. Die relative Positionsschätzungen auf Grundlage von IMU-Messungen sind grundsätzlich stabil, die doppelte mathematische Integration der Beschleunigungswerte zur Ermittlung des zurückgelegten Weges und die einfache mathematische Integration der Drehraten zur Bestimmung der Richtung führt aber in kurzer Zeit zur Fortpflanzung des Fehlers, was eine sehr schlechte langfristige Stabilität zur Folge hat. Ein weiterer Nachteil dieser Systeme ist der zusätzliche Hardwareaufwand.

Werden die Messungen einer *IMU* mit *GNSS*–Messungen fusioniert, so ergibt sich ein integriertes Navigationssystem, bei dem sich die beiden Systeme aufgrund ihrer unterschiedlichen Fehlereigenschaften sehr gut ergänzen. Die Langzeitstabilität des GNSS und die kurzzeitgenaue Positionsschätzungen auf Grundlage der IMU-Messungen gehen über eine Sensordatenfusion als Eigenschaften des integrierten Navigationssystems ein [FBVS10].

Da *IMUs* allerdings ausschließlich Beschleunigungen und Drehraten detektieren, ist deren Einsatz auf mobile, d. h., sich bewegende Sensornetzwerke beschränkt. Ein Einsatz in statischen Netzen würde keinen Einfluss auf die Positionsbestimmung haben.

Ultra-Wideband

Ultrabreitband ("Ultra-Wideband" (*UWB*)) ist durch seine große Bandbreite von bis zu einigen *GHz* gekennzeichnet. Verglichen mit den bisherigen erwähnten Funktechniken mit nur wenigen Ausbreitungspfaden stellt das dichte UWB–Mehrwegeszenario (gerade in Innenräumen) eine ehrgeizige Herausforderung dar, wenn die direkte Sichtverbindung zwischen Sender und Empfänger gefunden werden soll und eine noch größere, wenn keine direkte Sichtverbindung (engl. "Non–Line–of–Sight" (*NLoS*)) besteht [Nor09].

Durch die hohe Bandbreite hat *UWB* die Fähigkeit, zuverlässig durch Hindernisse, Wände etc., hindurch messen zu können. Damit stellt es eine vielversprechende Technologie zur drahtlosen Kommunikation mit höchsten Datenraten und zur Lokalisierung dar. Allerdings ist die Erforschung der Ortung mit *UWB*–Systemen noch in den Anfängen und somit sehr teuer, sodass momentan ein massenhafter Einsatz auf vielen Sensorknoten eines Netzwerkes nicht praktikabel wäre [Nor09, BNW09].

4.5 Fazit

Dieses Kapitel gab einen Überblick über aktuelle Algorithmen zur Lokalisierung miniaturisierter Sensorknoten, die in Tabelle 4.1 vergleichend gegenüber gestellt sind.

Es erfolgte eine Einteilung in approximative und exakte Algorithmen, die sich nicht nur in der erreichbaren Genauigkeit unterscheiden, sondern auch in der Art der verwendeten Beobachtungselemente. So benutzen approximative Algorithmen i. d. R. geometrische Beziehung zwischen den Sensorknoten oder zwischen den Sensorknoten und den Beacons. Die meisten feinkörnigen Lokalisierungsalgorithmen hingegen verwenden Distanz- oder Winkelmessungen, auf deren Grundlage Positionen für die unbekannten Sensorknoten bestimmt werden. Dadurch vergrößert sich bei den letztgenannten Algorithmen der rechentechnische sowie der kommunikative Aufwand. Das hat zufolge, dass sich die Energiereserven auf den ressourcenlimitierten Sensorknoten schneller erschöpfen als bei den approximativen Algorithmen. Dafür bieten die feinkörnigen Algorithmen jedoch eine hohe Positionsgenauigkeit. In dieser Arbeit wird ein Algorithmus vorgestellt, der die Vorteile beider Algorithmenkategorien (hohe Positionsgenauigkeit bei geringem Berechnungsaufwand) in sich vereint.

In diesem Kapitel erfolgte weiterhin eine Beschreibung möglicher Beobachtungselemente zur Lokalisierung in drahtlosen Geosensornetzwerken.

Algorithmus	Distanz-	Verfahren	Kom-	Speicher-	Genauig-
			plexität	bedarf	keit
CL	unabhängig	Schwerpunkt- bestimmung	gering	gering	gering
WCL	(unabhängig)*	gewichtete Schwer- punktbestimmung	gering	gering	gering
APIT	unabhängig	Schnittmengen- bildung	gering/ mittel	gering	gering
Szenen- analyse	unabhängig	Fingerprint- verfahren	gering	gering	gering
Bounding- Box	unabhängig	Schnittmengen- bildung	gering	gering	gering
Nearest Beacon	unabhängig	Koordinaten- nächster Beacon	sehr gering	sehr gering	sehr gering
FGL	abhängig	eukl. Distanzen Multilateration	sehr hoch	sehr hoch	sehr hoch
APS	abhängig	Distanzen aus Hop-Zählung Trilateration	mittel	mittel	mittel
ABC	abhängig	eukl. Distanzen Trilateration Transformation	mittel	gering	mittel
Robuste Quadrilaterale	abhängig	eukl. Distanzen Tri-/ Multilateration	hoch	hoch	mittel/ hoch
MDS	abhängig	eukl. Distanzen Trilateration	hoch	mittel	mittel
DLS	abhängig	eukl. Distanzen Multilateration	gering/ mittel	gering/ mittel	hoch
iDLS	abhängig	eukl. Distanzen Multilateration Matrixupgrades	mittel	gering/ mittel	mittel/ hoch
mDLS	abhängig	eukl. Distanzen Multilateration Matrixupdates	mittel	gering/ mittel	mittel/ hoch

Tabelle 4.1: Tabellarischer Vergleich gängiger Lokalisierungsalgorithmen (*Distanzen indirekt als Gewichte)

5 Einbeziehung von Sensormesswerten und Signalausbreitungsmodellen

"... geosensor network technologies are revolutionizing the way that geospatial information is collected, analyzed and integrated with the geospatial content of the information being of fundamental importance."

(Anthony Stefanidis, griech. Geodät)

Kapitel 4 beschrieb den Entwicklungsstand der Algorithmen zur Positionsbestimmung in drahtlosen Sensornetzwerken. Bisher wurden intensive Anstrengungen unternommen, um Positionsfehler durch bessere Lokalisierungsalgorithmen zu verringern, welche zumeist auf den "klassischen" Beobachtungsgrößen (Beaconpositionen, Distanz- und Winkelmessungen sowie geometrische Abhängigkeiten) beruhen. In diesem Kapitel soll ein gänzlich anderer Weg beschritten und der eigentliche Zweck von Geosensornetzwerken, nämlich der Messung physikalischer Parameter, benutzt werden, um Ausreißer im Beobachtungsmaterial zu detektieren und damit die Positionsgenauigkeit zu verbessern.

5.1 "Anomaly Correction in Localization"– Ausreißerdetektion mit geringem Aufwand

Zahlreiche Fehlereinflüsse in o.g. Beobachtungsdaten können auch mit den besten Lokalisierungsalgorithmen nicht vollständig beseitigt werden. Aus diesem Grunde können weitere, zur Verfügung stehende Informationen mit einbezogen werden. Von jedem Sensorknoten wird im Laufe der Zeit eine große Anzahl von Daten gesammelt, die Aussagen über die aktuelle Umgebung ermöglichen, in der sich die einzelnen Sensorknoten befinden. In der Regel besteht ein direkter Zusammenhang zwischen Sensordaten und dem Ort ihrer Aufnahme. Wenn es gelingen würde, diesen Zusammenhang mathematisch mithilfe von klar definierten Regeln apriori zu formulieren, können diese Informationen genutzt werden, um die Genauigkeit der Lokalisierung nachhaltig verbessern. Um das zu ermöglichen, werden dazu in diesem Kapitel zwei Algorithmen vorgeschlagen.

Zur Verdeutlichung dieser Idee soll an dieser Stelle ein einfaches Beispiel dienen. Bei dem in Kapitel 2.4 genannten "Precision Farming" sollen Sensorknoten zukünftig in großer Anzahl z. B. mit einem Flugzeug über Ackerflächen abgeworfen werden. Die Sensorknoten messen die chemische und biologische Zusammensetzung des Bodens als auch typische physikalische Parameter wie die Temperatur, die Bodenfeuchte, die Lichtintensität oder den Luftdruck. Zur Beschreibung dieses Sachverhalts soll an dieser Stelle der Sensor zur Messung der Bodenfeuchte herausgegriffen werden. Knoten, die beispielsweise in einer Senke zum Liegen kommen, würden dauerhaft höhere Bodenfeuchtewerte

messen, da sich das Wasser in diesen Senken sammelt. Demgegenüber würden Sensorknoten auf einer Anhöhe geringere Feuchtewerte beobachten. Um einen Rückschluss auf die Lokalisierung zu schaffen, könnten digitale Geländemodelle (engl. "Digital Terrain Model" (*DGM*)) kombiniert mit den Schätzpositionen der Knoten und den Sensordaten erstellt werden. Dafür ist eine Skalierung oder Transformation Voraussetzung, d. h., die Messwerte der Sensoren (die als *z*-Werte interpretiert werden können) müssen in den Wertebereich der in der Ausbreitung des Geosensornetzwerkes vorkommenden Höhenwerte skaliert oder transformiert werden. Zum Beispiel hat das DGM Höhenwerte von 30 *m* bis 60 *m* ($\Delta h = 30m$), die Sensoren messen in einem anderen Wertebereich die Bodenfeuchte. Die auftretenden Bodenfeuchteunterschiede müssen auf Δh transformiert werden, was einer Interpolation gleichkommt. Danach sind die Oberflächenmodelle zu verschneiden (Subtraktion der *z*-Werte) und zu prüfen, ob diese beiden parallel verlaufen. Je nachdem, ob außer der Werteskalierung auch eine Translation vorgenommen wurde, ist für den Fall, dass beide Oberflächenmodelle parallel verlaufen, die Differenz in allen Punkten gleich oder null. Kriterium für eine falsche Lage, und somit einem Ausreißer, ist dann eine deutliche Abweichung von dieser mittleren Differenz.

Diese Idee soll in diesem Ansatz wie folgt genutzt werden: In Sensornetzwerken sind oft mehr Beacons verfügbar, als es für eine einfache Trilateration notwendig wäre. Daraus resultiert die Möglichkeit, mehrere verschiedene Schätzpositionen zu ermitteln, um anschließend Fehler zu eliminieren. In diesem Abschnitt soll ein Algorithmus formuliert werden, der mit Hilfe der Messdaten der einzelnen Sensoren stark fehlerhafte Positionen aussortiert und damit die Lokalisierung verbessert. Dies basiert auf der Idee, dass unter Umständen für bestimmte Messgrößen im voraus verschiedene ortsabhängige Wertebereiche bestimmte Gebiete für die eigene Position ausschließen. Schätzpositionen in diesen ausgeschlossenen Bereichen werden dann nicht mehr zur Bestimmung der endgültigen Position verwendet, was in den meisten Fällen den Lokalisierungsfehler verringert.

Zusammenfassend soll es das Ziel sein, die sehr ressourcensparende Trilateration mit nur geringem Zusatzaufwand auf den Knoten ausführbar zu machen und trotzdem eine relativ genaue und ausreißerbefreite Position ermitteln zu können. Besonders einzelne Ausreißer bei den Distanzmessungen können die Ergebnisse der Trilateration derart verfälschen, dass eine Positionsbestimmung nicht oder nur mit sehr großem Lokalisierungsfehler möglich ist. Diesen Effekt soll der nun folgende "Anomaly Correction in Localization"–Algorithmus (*ACL*) verringern bzw. sogar vollständig vermeiden [RBN⁺08].

5.1.1 Funktionsweise

ACL verbessert das Lokalisierungsergebnis, indem ungenaue Positionsschätzungen eliminiert werden. Dabei werden vorliegende Sensormesswerte mit Vorwissen über die zu messende Umgebung kombiniert. Abbildung 5.1 illustriert die Problemstellung anhand von Bodenfeuchtesensoren. Es werden Positionsschätzungen verworfen, wenn eine Diskrepanz zwischen erwarteter Sensor- und tatsächlicher Messung festzustellen ist. In diesem Fall werden nur Positionen als gültig markiert, die mit dem erwarteten Sensormesswert korrespondieren.

Für einen sinnvollen Einsatz von ACL sind einige Voraussetzungen zu formulieren:

• es müssen ausreichend viele Sensoren mit räumlich zuzuordnenden Abhängigkeiten auf den Sensorknoten installiert sein, und



Abbildung 5.1: Verschiedene Positionen durch Trilateration und Sensormesswerte.

 es muss möglich sein, die räumlichen Eigenschaften mit definierten Regeln mathematisch zu erfassen, z. B. ein Sensorknoten befindet sich auf einer Anhöhe, da seine Bodenfeuchtemessung in einem speziellen Intervall liegt.

Es müssen also Umweltparameter einem konkreten Raumausschnitt zuzuordnen sein. Diese wichtige Zuordnung eines bestimmten Raumausschnitts zu einem festgelegten Sensorwertebereich wird im Folgenden vereinfachend als Sensorintervall bezeichnet. Mögliche Sensormessgrößen sind dabei Temperatur, Lichtstärke, Feuchtigkeit, pH-Wert, Druck, Schall aber auch alle anderen skalar messbaren und mathematisch interpretierbaren physikalischen Größen. Ein Sensorintervall gilt dann als einem Raumausschnitt zugehörig, wenn der zu erwartende Messwert mit hoher Wahrscheinlichkeit innerhalb eines bestimmten Intervalls liegt.

Eine weitere wichtige Voraussetzung ist eine Inhomogenität der jeweiligen Umweltparameter. Wenn das gesamte Messfeld einem einzigen Intervall angehört oder die Messgröße nicht klassifizierbar ist, ist diese Größe für die Lokalisierung nicht nutzbar. Deshalb sind nur Größen anzuwenden, die sich signifikant in zwei oder mehr Intervalle unterteilen lassen.

Der *ACL*–Algorithmus baut auf eine vorangegangene Trilateration auf. Da meist mehr Beacons als die drei benötigten in der Empfangsreichweite eines Sensorknotens zur Verfügung stehen, bietet es sich an, mehrere Trilaterationen zu verschiedenen Beacons durchzuführen. Durch Permutation können $\binom{n}{3}$ verschiedene Kombinationen berechnet werden. Die Frage ist nun, welche Trilateration zum besten und damit genauesten Resultat führen würde, um den Rechenaufwand minimal zu halten. Dieses Optimum wird in einer Ersatzfragestellung formuliert. Welche Trilateration führt zu einem Resultat, das den klar definierten Regeln des *ACL*–Algorithmus nicht widerspricht. Exemplarisch auf das erwähnte Beispiel angewendet bedeutet das, es liegt ein Widerspruch vor, wenn eine Schätzposition auf einer Anhöhe liegt, der Sensor aber eine sehr hohe Bodenfeuchte gemessen hat. Dieser Widerspruch macht das Ergebnis der Trilateration unglaubwürdig. Die in *ACL* angewendeten mathematischen Grundlagen beschreibt der folgende Abschnitt.

5.1.2 Das Modell

Es wurde ein generisches Modell entwickelt, welches real gegebene Umweltparameter abstrahiert und eine variable Intervallzuordnung gestattet. Dazu wird zunächst ein rechteckiges Messfeld festgelegt, auf dem die Sensorknoten platziert werden. Das gesamte Areal ist in ein quadratisches, beliebig fein skalierbares Raster aufgeteilt, das auf eine rechteckige Matrix abgebildet wird. Damit ist bei sehr flexibler Konfiguration eine schnelle Berechnung gewährleistet.

Sensorintervalle haben keine festgelegten Werte, sondern werden durch jene Flächen symbolisiert, in denen die Sensorwerte innerhalb bestimmter Intervalle liegen. Diese Flächen werden aus einer Menge zusammenhängender Quadrate gebildet, wobei jedes Quadrat durch eine logische 1 an der entsprechenden Position in der Matrix repräsentiert wird. Das vereinfacht die notwendigen Berechnungen sehr, da keine geometrischen Tests durchgeführt werden müssen, ob sich ein Punkt innerhalb einer Fläche befindet, sondern lediglich eine Abbildung der Flächenkoordinaten auf die Dimension der Matrix und ein logischer Vergleich notwendig sind. Die Größe, Anzahl und Position dieser Sensorintervalle sind generisch einstellbar.

Die Messung der Sensorwerte wird durch die Zuordnung der Knoten zu den Sensorintervallen modelliert, in denen die tatsächliche Position des Sensors liegt. Dabei wird ein möglicher Messfehler zunächst ausgeschlossen, d. h. jeder Sensor liefert Werte aus dem exakten, zur Position gehörenden Intervall. Die gemessenen Sensorinformationen werden nachfolgend abstrakter Sensorprofil genannt und gelten als definierte, räumlich zugeordnete Eigenschaft.

In Abbildung 5.2 wird zur Erklärung ein Messfeld beispielhaft veranschaulicht. Jeder Knoten besitzt einen binären Vektor, der die Zugehörigkeiten zu den einzelnen Sensorintervallen beinhaltet. Knoten 1 (N_1) besitzt in diesem Fall den Vektor $V(N_1) = (1, 0, 0)$, was bedeutet, dass sich N_1 innerhalb des Sensorintervalls 1 befindet, aber außerhalb der Intervalle 2 und 3.

5.1.3 Ablauf des Algorithmus

An dieser Stelle folgt eine Algorithmusbeschreibung. Es gibt zwei mögliche Abläufe, die später untersucht werden sollen.

5.1.3.1 ACL mit Mittelung

Der folgende Ablauf ist rechenaufwendiger, aber auch genauer, da alle möglichen Trilaterationen mit einbezogen werden:

- 1. Die Beacons senden ihre Position an die in Sendereichweite befindlichen Sensorknoten.
- 2. Die Sensorknoten speichern die empfangenen Positionen und ermitteln die Distanzen zu den Beacons über ein geeignetes Messverfahren, wie z. B. Signalempfangsstärkemessung.
- 3. Nachdem alle Daten abgespeichert wurden, berechnen die Sensorknoten alle $q = \binom{n}{3}$ möglichen Positionen durch wiederholte Trilateration.
- 4. Alle resultierenden Positionen werden durch den ACL-Algorithmus überprüft. Dazu wird verglichen, ob sich die tatsächlich gemessenen Sensorwerte des Knotens innerhalb der vorher


Abbildung 5.2: Schematische Darstellung der Lokalisierung und Ausreißerdetektion mit ACL.

definierten Sensorintervalle befinden. Ist dies nicht der Fall, wird die Position als ungültig eingestuft, andernfalls als gültig.

5. Nachdem alle Positionen den *ACL*-Test durchlaufen haben, wird mit den gültigen Positionen eine Mittelung durchgeführt. Diese Mittelung führt dann zur endgültigen Schätzposition.

5.1.3.2 ACL mit erster gültiger Position

Da der vorher beschriebene Algorithmus nachteilig alle Positionen berechnen muss, soll eine praktisch interessantere Version eingeführt werden. Hierbei bricht die Berechnung bei der ersten gültigen Trilateration sofort ab. Der Ablauf ist der gleiche wie zuvor beschrieben, wobei ab dem vierten Schritt solange Trilaterationen durchgeführt und *ACL* unterzogen werden, bis die erste durch *ACL* gültig bewertete Position ermittelt wurde. Diese wird dann als Schätzposition des Knotens abgespeichert. Einschränkend ist anzumerken, dass im ungünstigsten Fall ebenso viele Trilaterationen wie im letzten Abschnitt beschrieben durchgeführt werden müssen. Das ist dann der Fall, wenn nur eine, die letzte Trilateration einen gültigen Wert liefert. Bei dem *ACL* mit erster gültiger Position ist ein höherer Lokalisierungsfehler zu erwarten, was im folgenden Abschnitt erläutert werden soll.

5.1.4 Simulation und Ergebnisse

Zur Simulation wurde der Paketsimulator J-Sim verwendet [jsi07]. An dieser Stelle sollen die Simulationsbedingungen und die Ergebnisse im Vordergrund der Betrachtungen stehen. Bei der Lokalisierung wird von einer Normalverteilung aller Knoten ausgegangen, sodass günstige und ungünstige Anordnungen bei der Trilateration gleichermaßen vorkommen. Deshalb sind schon bei diesem idealisierten Gauß'schen Fehlermodell starke Ausreißer in der Lokalisierung zu erwarten.

In den Simulationen werden der Lokalisierungsfehler der Erstposition E_f und der Lokalisierungsfehler der ersten korrekten Position $E_{f,ACL}$ bestimmt. Es wird also dargestellt, um wie viel sich der Lokalisierungsfehler im Mittel verringern lässt, wenn der *ACL*–Algorithmus mit Variante 1 bzw. 2 angewendet wird. Die Verbesserung ist in den Diagrammen in der Einheit Meter dargestellt, da aber der Bezug die Länge des Sensorfeldes mit 100 *m* ist, kann die Einheit ebenfalls in Prozent angegeben werden. Gleiches gilt für die Standardabweichung. Die Größe des Sensorfeldes wurde auf $(100 \, m \times 100 \, m)$ mit einer Rastereinteilung in $(5 \, m \times 5 \, m)$ Quadrate festgelegt. Dies gilt für alle folgenden Simulationen. Abbildung 5.3 zeigt eine typische Darstellung unter Anwendung von J-Sim.

	0				_					Node 7
guiGSI guiGSI guiGSI	0 1 2			√N 24				• N	19	-14006 /
				• N	11	•1	N 20			
		ΔN	ode 1			▶N 30	·	N 13	~~~~~	b5 le 5
		-	• N 18	14	Node 1	0			• N 28	
						ode 2				•N 22
	-517	5			⊷N 21			• N 27		-14 22
	14-2									
							- I	N 12		
		•N	26							
								N 14		
						/N 16			•117	
				∆Ni ∆Node	ode 4 3		Node 8			
		_			ode 6					
						•₩23	+N:	29		
			∆Node	y						

Abbildung 5.3: Simulationsergebnis in J-Sim; mit Dreiecken als Beacons; Punkte als Sensorknoten mit ihren Fehlervektoren; Sensorintervalle als dunkle Flächen.

5.1.4.1 ACL mit Mittelung

In dieser ersten Simulation wurden auf dem Feld zehn Beacons und 20 Sensorknoten verteilt. Weiterhin ist mit drei Sensorintervallen der Größe $49 \ m \times 49 \ m$ und einem Distanzfehler mit $\sigma_d = 1 \ m$ gerechnet worden. Für den Fall, dass sich alle drei Sensorintervalle nicht überschneiden, entspricht deren Fläche etwa 36 % der Gesamtfläche. In einer Simulation mit zehn Beacons ergeben sich $\binom{10}{3} = 120$ mögliche

Trilaterationen. Die Daten sind nach dem Anteil der Ausreißer sortiert, d. h., es wurde berechnet, wie viele der 120 möglichen Positionen außerhalb des korrekten Sensorintervalls liegen.

Für jeden der verwendeten 20 Knoten ergibt sich ein bestimmter Prozentsatz an Ausreißern und damit eine Linie im Diagramm. Jede Linie kennzeichnet den absoluten Wert der mittleren Verbesserung, die einer der Knoten bei 120 Trilaterationen mit dem entsprechenden Anteil an Ausreißern erzielt hat. Bei mehreren Knoten mit gleicher Anzahl an Ausreißern wird die Verbesserung gemittelt.

Da die reinen Daten sehr hohe Streuungen aufweisen, wird eine lineare Regressionsgerade errechnet, die den Trend verdeutlicht. Mit zunehmender Anzahl von Ausreißern steigt die Verbesserung. Dieser Trend ist nicht nachweislich linear, jedoch wurde bei vielen Simulationen ein ähnliches Verhalten beobachtet, sodass die Vermutung nahe liegt, dass die Verbesserung mindestens linear oder sogar potentiell ansteigt. Gleichfalls zeigt sich bei geringem Anteil von Ausreißern, bis etwa 15 %, dass die Verbesserung negativ ist. Das zeigt, dass eine bestimmte untere Grenze für die Ausreißer existiert, unterhalb der die Anwendung des *ACL*–Algorithmus nicht empfehlenswert ist. Diese Grenze ist von der jeweiligen Konfiguration abhängig und kann deshalb nicht allgemein angegeben werden.

5.1.4.2 Abhängigkeit der Anzahl der Ausreißer vom Distanzfehler

Der Anteil der Ausreißer ist wiederum vom Fehler der Distanzmessung, hier also von der Standardabweichung σ_d abhängig. In den nächsten Simulationen wurde die gleiche Konfiguration wie im letzten Abschnitt bei veränderlichen Distanzfehlern untersucht. Der Parameter σ_d wurde von 1m bis 10min Einserschritten eingestellt. Als Funktion davon wurde der Anteil der Ausreißer aufgenommen. In



Abbildung 5.4: (a) Durchschnittliche Verbesserung über Ausreißeranzahl mit einer Standardabweichung von $\sigma_d = 1,0 m$ und (b) Ausreißeranzahl in Abhängigkeit der Standardabweichung σ_d mit 10,0 m Beacons, 20 Sensorknoten.

Abbildung 5.4(a) ist der über mehrere Simulationen gemittelte Anteil der Ausreißer in Abhängigkeit von σ_d dargestellt. Es zeigt sich, dass die Anzahl der Ausreißer mit zunehmender Standardabweichung steigt. Auch hier ist die Regressionsgerade berechnet worden. Diese soll verdeutlichen, dass die Messgrößen stark zufallsbehaftet sind und in jeder Simulation andere Daten liefern. Es lässt sich jedoch erkennen, dass der Wert von durchschnittlich 15% Ausreißern schnell überschritten wird. Das heißt, der *ACL*-Algorithmus wird nur bei sehr kleinen Werten von σ_d unwirksam oder wirkt sich, wie in Abbildung 5.4 (b) gezeigt, negativ aus.

5.1.4.3 Einfluss von Größe und Anzahl der Sensorintervalle

Einen weiteren wichtigen Einfluss auf die Effektivität von *ACL* haben Größe, Anzahl und Lage der Sensorintervalle. In den Simulationen wurden die Intervalle jeweils in Größe und Anzahl parametrisiert, aber zufällig angeordnet. Dies ist neben der zufälligen Knotenverteilung einer der Hauptgründe für die starken Schwankungen der Ergebnisse. Es werden daher keine expliziten Ergebnisse in der Form erzielt, dass eine Formel für die Genauigkeit des *ACL* in Abhängigkeit dieser Parameter hergeleitet werden kann. Eine Untersuchung des Einflusses von Größe und Anzahl der Sensorintervalle hat aber im Wesentlichen die erwarteten Ergebnisse ergeben. Je mehr Sensorintervalle existieren, umso feiner und vielfältiger ist die Unterteilung des Messfeldes in kleinere Sektionen mit verschiedenen Sensorprofilen. Dies trifft vor allem auch dann zu, wenn die Sensorintervalle größer sind und sich damit mehr überschneiden. Daher wächst die Anzahl der Ausreißer mit steigender Größe und Anzahl der Sensorintervalle, was zu einer erhöhten Verbesserung führt. Es ist auch zu beachten, dass Sensoren, die in der Nähe einer Intervallgrenze liegen, am stärksten von der Ausreißerdetektion profitieren, da dort die Wahrscheinlichkeit für Ausreißer höher ist als in einem homogenen Bereich. In



Abbildung 5.5: Ausreißer und durchschnittliche Verbesserung in Abhängigkeit von der Anzahl der Sensorintervalle mit zehn Beacons und 20 Sensorknoten. Vergleich von ACL mit Mittelung und ACL mit erster gültiger Position in Abhängigkeit der Standardabweichung σ_d .

Abbildung 5.5 (a) ist diese Abhängigkeit dargestellt. Mit dem Anteil der Ausreißer steigt auch die mittlere Verbesserung \overline{V} bei zunehmender Anzahl von Sensorintervallen. Es wurde dazu ein Sensorintervall mit drei Teilflächen der jeweiligen Größe $49 m \times 49 m$ erzeugt. Im zweiten Schritt wurde ein weiteres gleichartiges Sensorintervall hinzugenommen und dann ein drittes, während die vorherigen jeweils in konstanter Anordnung gehalten wurden. Im Feld sind zehn Beacons, 20 Sensorknoten platziert und die Standardabweichung der Strecken $\sigma_d = 1 m$ gesetzt.

5.1.4.4 Vergleich von ACL mit Mittelung und mit erster gültiger Position

Aufgrund der verschiedenen Möglichkeiten eine endgültige Schätzposition zu bestimmen, wurden Versuche durchgeführt, um die eine oder andere Möglichkeit zu begünstigen. Dazu wird auf einem Feld mit drei Sensorintervallen der Größe $49 \ m \times 49 \ m$ mit zehn Beacons und 20 Knoten ein Vergleich von *ACL* mit Mittelung \overline{V} und *ACL* mit erster gültiger Position V_f durchgeführt. Die Werte sind dabei jeweils über die 20 Knoten gemittelt, um die Streuung nicht zu stark ins Gewicht fallen zu lassen.

Die Standardabweichung der Distanzen wird dazu wieder von 1,0 *m* bis 10,0 *m* durchlaufen. Es zeigt sich in Abbildung 5.5 (b) deutlich, dass *ACL* mit erster gültiger Position weit weniger stark wächst als *ACL* mit Mittelung. Zum einen liegt es daran, dass bei *ACL* mit erster gültiger Position weit weniger Trilaterationen ausgewertet werden. Das lässt sich dadurch erklären, dass sie nicht in der Nähe des Randes eines Intervalls liegen und damit keine oder nur wenige Ausreißer besitzen. Auch diese Trends sind nicht unbedingt linear und sind durch Regressionsgeraden verdeutlicht. *ACL* mit Mittelung erzielt hier erheblich bessere Ergebnisse und ist damit die Methode der Wahl, jedoch muss auch die erhöhte Rechenleistung berücksichtigt werden.

5.1.5 Einschränkungen für ACL

Die Datenauswertung der Simulation gestaltete sich wegen des hohen Zufalleinflusses sehr schwierig und daher sind die Ergebnisse auch nicht sehr explizit, doch es wäre zu unrealistisch, alle Parameter inklusive der Knotenverteilung konstant zu halten. So konnte sichergestellt werden, dass der Algorithmus nicht nur für eine ganz bestimmte Anordnung funktioniert, sondern auch robust ist gegenüber ungünstigen Knotenkonstellationen. Gleichzeitig dienen die Schwankungen als Beleg dafür, wie unsicher die einfache Trilateration mit fehlerhaften Distanzen ist.

Einschränkend muss die Möglichkeit erwähnt werden, dass ein Sensorintervall mit demselben Sensorprofil auf mehrere räumlich getrennte Flächen fällt. *ACL* unterscheidet in diesem Falle jedoch nicht, was dazu führt, dass Positionen, die in einer Teilfläche liegen, jedoch in der anderen mit selbem Sensorintervall lokalisiert werden, nicht als Ausreißer erkannt werden.

Die bisherige Idee ist noch nicht optimal umgesetzt. So gilt es ein besseres mathematisches Modell zu finden, auf dessen Grundlage der *ACL*–Algorithmus läuft. Derzeit ist es ein reiner Vergleich von Messwerten mit den entsprechenden Sensorintervallen. Dies ist zwar sehr rechensparend durchzuführen, aber die räumlichen Informationen können nur sehr grob ausgenutzt werden. Als aussichtsreiches Modell wäre z. B. die konvexe Optimierung vorstellbar.

Des Weiteren ist die Definition der Sensorintervalle noch recht statisch und wenig flexibel. Dies kann auch während der Laufzeit zu Problemen führen, da sich Sensorintervalle zeitlich ändern können. Gerade beim Beispiel der Bodenfeuchte sind starke Änderungen der Intervalle denkbar. Diesbezüglich müsste eine automatische Intervallgenerierung erfolgen, was mit einem evolutionären Algorithmus möglich erscheint.

Letztendlich wurden die zusätzlichen Informationen als reine Ausreißerdetektion bei der Trilateration genutzt. Ziel muss es sein, weitere Lokalisierungsalgorithmen mit diesen Informationen zu verbinden, sodass eine höhere Genauigkeit erreicht werden kann. Dies hängt aber vor allem von der Leistungsfähigkeit neuer mathematischer Modelle ab.

5.2 "extended Anomaly Correction in Localization"– Berücksichtigung von Hindernissen im Signalausbreitungskanal

Im vorangegangen Abschnitt wurde ein Algorithmus vorgestellt, der unter Berücksichtigung von Sensormesswerten Ausreißer detektiert, aus dem Lokalisierungsprozess eliminiert und dadurch die Positionsschätzung verbessert. Dazu wurden im Wesentlichen erwartete Sensormesswerte auf eine Footprintkarte übertragen, die auf den Sensorknoten gespeichert ist. Durch einfache Trilateration zu jeweils drei Beacons bestimmt der Sensorknoten seine Position und validiert diese durch einen Vergleich der erwarteten Sensormesswerte mit den tatsächlich gemessenen. Die Genauigkeit der Positionsbestimmung wird wiederum stark durch die Qualität der jeweiligen Entfernungsmessung beeinflusst. Gerade die oftmals eingesetzte Distanzschätzung auf Basis von Signalempfangsstärkemessungen ("Received Signal Strength", *RSS*) ist sehr fehleranfällig (vgl. Kapitel 4.4.1). In diesem Abschnitt werden *RSS*-beeinflussende Faktoren detailliert erklärt und ein Modell vorgestellt, das auf Basis des *ACL*-Algorithmus diese Faktoren berücksichtigen soll, um die Distanzmessungen zu verbessern, bevor der "extended Anomaly Correction in Localization"-Algorithmus (*eACL*) formuliert wird.

5.2.1 Den Signalausbreitungskanal beeinflussende Effekte

Bevor der neue Algorithmus formuliert wird, soll eine Betrachtung möglicher Effekte erfolgen, die das Signal entlang des Ausbreitungsweges beeinflussen. Diese Effekte beeinträchtigen die Analyse des Signals und damit die daraus resultierende Distanzschätzung. Es wird dabei unterschieden, über welche Entfernungen sich diese Effekte äußern. "Large Scale Fading" (*LSF*) charakterisiert die Signalstärke über größere Distanzen zwischen Sender und Empfänger und wird im Wesentlichen durch Signalabschwächung (Dämpfung) und Reflexion beschrieben. Das "Small Scale Fading" (*SSF*) hingegen beschreibt Signalfluktuationen, die auftreten, wenn sich die Distanz zwischen Sender und Empfänger um kleine Beträge ändert. Abbildung 5.6 illustriert SSF und die mehr graduellen LSF-Variationen für ein Indoor-Kommunikationssystem. In diesem Beispiel entfernt sich der Empfänger von der Sendeeinheit. Gut zu erkennen ist das SSF, welches durch eine starke Fluktuation der Signalstärke zum Ausdruck kommt.

5.2.1.1 Large Scale Fading

LSF beschreibt die Ausbreitung eines Signal entlang des Kommunikationspfades. Es bezeichnet dabei die Veränderung der mittleren Amplitude bzw. des mittleren Pegels des empfangenen Signals bei Verwendung eines raumabhängigen Mehrwegekanals. Ursache von LSF ist die geänderte "Sichtbarkeit" des Sendesignals. Mit anderen Worten kann ein Übertragungskanal in zwei Kategorien eingeteilt werden. Ist die Sichtverbindung zwischen Sender und Empfänger frei, spricht man von "Line–of–sight"– Bedingung (LoS). Das Signal unterliegt dabei lediglich der Freiraumdämpfung. Befinden sich allerdings Hindernisse zwischen Sender und Empfänger, liegt eine "Non–Line–of–sight"–Bedingung (NLoS) vor (vgl. Kapitel 4.4.3). Befindet sich ein Sender in einem Bereich, der LoS aufgrund einer Verdeckung verhindert, so ist ein starker Abfall des Empfangssignals detektierbar.



Abbildung 5.6: Large Scale und Small Scale Fading [Rap02].

5.2.1.2 Small Scale Fading

SSF beschreibt kurzfristige Fluktuationen der Amplituden, Phasen oder Verzögerung durch Mehrwegeausbreitung eines Funksignals über eine sehr kurze Zeitspanne oder Distanz. LSF–Effekte werden hierbei außer Acht gelassen. SSF wird hervorgerufen durch Interferenz zwischen zwei oder mehr Versionen des ausgesendeten Signals, welche den Empfänger zu leicht unterschiedlichen Zeiten erreichen.

Viele physikalische Effekte in der Signalausbreitung können das SSF beeinflussen. Im Wesentlichen sind das folgende:

- Mehrwegeausbreitung: Reflektierende Objekte und Streuungen im Signalpfad kreieren eine ständig wechselnde Umgebung, was die Signalenergie über die Zeit verbraucht und auch Phase und Amplitude verändert. Dieser Effekt resultiert in mehreren Versionen des ausgesendeten Signals, die an der Empfangsantenne detektiert werden. Die zufälligen Phasen und Amplituden der unterschiedlichen Mehrwegekomponenten induzieren SSF-Effekte und rufen somit Fluktuationen in der Signalempfangsstärke hervor.
- Im Bewegungsfall des Empfängers: Die relative Bewegung zwischen Sender und mobilem Empfänger resultiert in zufälligen Frequenzmodulationen, hervorgerufen durch Dopplerverschiebung einer jeden einzelnen Mehrwegekomponente. Abhängig von der Bewegung (kleiner/größer werdende Entfernung zum Sender) nimmt die Dopplerverschiebung positive/negative Werte an.
- Bewegung von Objekten im Signalpfad: Bewegen sich Objekte im Ausbreitungskanal, werden zeitvariierende Dopplerverschiebungen induziert. Ist die Bewegungsrate der Objekte höher als die des Empfängers, dann dominiert dieser Effekt das SSF. Im umgekehrten Fall kann die Bewegung umgebender Objekte vernachlässigt werden.
- Bandbreite des ausgesendeten Signals: Ist die Bandbreite des ausgesendeten Signals größer als die des Mehrwegekanals, so wird das Empfangssignal verzerrt. Die Signalstärke wird dabei

aber nur in sehr geringem Maße gedämpft und tritt nur lokal auf. Das heißt, das SSF ist nicht signifikant. Ist die Bandbreite des ausgesendeten Signals kleiner, ändert sich die Amplitude des Signals. Es wird aber nicht verzerrt.

Das "Small Scale Fading" wurde an dieser Stelle vorgestellt, im weiteren Verlauf des Kapitels wird allerdings nicht weiter darauf eingegangen. Die Modellierung von SSF–Effekten ist eine viel diskutierte Problematik in der Literatur. Aufgrund ihrer komplexen Natur ist eine realistische Modellierung schwierig umsetzbar und auch nicht Gegenstand dieser Arbeit. In dem erweiterten *ACL*–Algorithmus werden ausschließlich "Large Scale Fading"–Effekte berücksichtigt.

5.2.1.3 Grundlegende Mechanismen bei der Signalausbreitung

Reflexion, Diffraktion und Diffusion sind neben der Signaldämpfung die drei grundlegenden Mechanismen, die die Signalausbreitung in einem Funksystem beeinflussen.

Reflexion tritt auf, wenn eine sich ausbreitende elektromagnetische Welle auf ein Objekt großer Ausdehnung, d. h. größer als die Wellenlänge λ des Signals, trifft. Wenn eine sich durch ein Medium Aausbreitende Welle auf ein anderes Medium B mit unterschiedlichen elektrischen Eigenschaften trifft, wird ein Teil der Welle reflektiert und ein Teil in das Medium B übertragen. Handelt es sich bei B um einen Nichtleiter wird ein Teil dieser Energie in das Medium übertragen und teilweise wird die Energie in das Medium A reflektiert. Bei einem Nichtleiter tritt ebenso ein Energieverlust durch Absorption auf. Ist das zweite Medium B hingegen ein perfekter elektrischer Leiter, wird alle Energie ohne Verlust in das erste Medium A reflektiert.

Die elektrische Feldintensität der reflektierten und der übertragenden Welle kann mit der einfallenden aus dem ursprünglichen Medium durch den *Fresnel'schen Reflexionskoeffizienten* Γ in Beziehung gesetzt werden. Der Reflexionskoeffizient hängt von den Materialeigenschaften der Medien sowie von der Polarisation der Welle, des Einfallswinkels und der Frequenz der Einfallswelle ab. Abbildung 5.7 (a) zeigt die Abhängigkeit am Beispiel des Einfallswinkels. Ein oft verwendetes Modell zur Berücksichtigung von Reflexionen ist das "Ground Reflection Model", welches später detaillierter betrachtet werden soll.

Diffraktion tritt unter anderem dann auf, wenn der Ausbreitungspfad zwischen Sender und Empfänger durch ein Objekt oder eine Oberfläche mit scharfen Unregelmäßigkeiten (Kanten, etc.) blockiert ist. Auch an einem Objekt, das sich außerhalb des Signalpfades befindet, kann Diffraktion auftreten. Das Prinzip der Diffraktion erlaubt es elektromagnetischen Wellen, sich hinter Hindernissen weiter auszubreiten. Zwar nimmt die Feldstärke rapide ab, je tiefer ein Empfänger in den abgeschatteten Bereich vordringt. Allerdings können in den meisten Fällen auch weiterhin brauchbare Signale erzeugt werden.

Das Phänomen der Diffraktion beruht auf dem Huygens'schen Prinzip, das besagt, dass alle Punkte einer Wellenfront als Ausgangspunkte einer neuen Welle betrachtet werden können. Diese neuen Wellen werden als Elementarwellen bezeichnet [Rap02]. Abbildung 5.7 (b) verdeutlicht das Prinzip. Diese neu erzeugten Elementarwellen formen eine neue Wellenfront in Ausbreitungsrichtung. Diffraktion wird durch die Ausbreitung der Elementarwellen in den abgeschatteten Bereichen deutlich. Die Feldstärke der gebeugten Welle in der abgeschatteten Region ist die Vektorsumme aller Elementarwellen im



Abbildung 5.7: (a) Reflexionskoeffizient in Abhängigkeit des Einfallswinkels auf die Grenzfläche zweier idealer Leiter und (b) Brechung einer ebenen Wellenfront nach dem Huygens'schen Prinzip [Wika].

Umfeld des Hindernisses. Bei hohen Frequenzen hängt die Diffraktion wie auch Reflexion stark von der Geometrie des Objektes ab.

Diffusion kann beobachtet werden, wenn an einem Empfänger eine höhere Signalempfangsstärke gemessen wird, als es durch Reflexions- und Diffraktionsmodelle beschrieben werden kann. Dieses Phänomen rührt daher, dass eine elektromagnetische Welle auf eine raue Oberfläche trifft. Die reflektierte Energie wird in alle Richtungen gestreut. Objekte wie z. B. Straßenlaternen und Baumbelaubung streuen das Signal in alle Richtungen und bewirken somit eine zusätzliche detektierbare Energie am Empfänger.

Glatte Oberflächen mit größerer Ausdehnung als die Wellenlänge des Signals können als reflektierende Flächen modelliert werden. Die Rauheit der Oberflächen ruft allerdings andere Ausbreitungseffekte hervor, als sie durch die spiegelnde Reflexion beschrieben wird. Die Oberflächenrauheit kann berücksichtigt werden, indem der Fresnel'sche Reflexionskoeffizient modifiziert wird [Rap02]. Dazu kann das "Rayleigh-Kriterium" herangezogen werden, welches eine kritische Höhe h_c für die Oberflächenstruktur und einen bestimmten Einfallswinkel θ angibt:

$$h_c = \frac{\lambda}{8\sin\theta_i}.\tag{5.1}$$

Oberflächen, deren minimale und maximale Strukturen kleiner als h_c sind, gelten als glatt und als rau, wenn die Grenzhöhe überschritten wird. Für diese rauen Oberflächen muss der Reflexionskoeffizient mit einem Streuungsverlustfaktor ρ_s multipliziert werden. Der Streuungsverlustfaktor berechnet sich nach:

$$\rho_s = exp\left[-8\left(\frac{\pi\sigma_h\sin\theta_i}{\lambda}\right)^2\right]I_0\left[8\left(\frac{\pi\sigma_h\sin\theta_i}{\lambda}\right)^2\right]$$
(5.2)

mit I_0 als Besselfunktion erster Art und nullter Ordnung. Ament geht davon aus, dass die Höhe h (Oberflächenrauheit) einer Gaußverteilung mit der Standardabweichung σ_h unterliegt [Ame53]. Das re-

flektierte E-Feld für $h > h_c$ kann nun für raue Oberflächen mit dem modifizierten Reflexionskoeffizienten Γ_{rough} berechnet werden:

$$\Gamma_{rough} = \rho_s \Gamma \tag{5.3}$$

5.2.2 Dämpfungsmodelle

An dieser Stelle sollen Möglichkeiten vorgestellt werden, um LSF–Effekte zu modellieren und für die Distanzschätzung zu nutzen. Da Reflexionen, Diffraktion und Diffusion besonders für SSF bedeutend sind und sich nur unzureichend modellieren lassen, wird ausschließlich die Signalabschwächung durch Dämpfung benutzt. In der Literatur sind viele Dämpfungsmodelle zu finden [Rap02]. Die wichtigsten Kategorien sollen im Folgenden kurz erläutert werden.

5.2.2.1 Free Space Propagation–Modell

Das "Free Space"–Modell betrachtet die Wellenausbreitung im leeren Raum. Der Wert, der sich aus dem "Free Space"–Modell als Signalabschwächung ergibt, wird als Freiraumdämpfung bezeichnet. Nach dem Energieerhaltungssatz kann Energie nicht verloren gehen und da sich im leeren Raum keine Objekte befinden, welche die Wellen beeinflussen können, verteilt sich die ausgesendete Energie gleichmäßig auf die Oberfläche einer imaginären Kugel mit der Antenne als Ursprung. Je größer der Radius der Kugel, desto weniger Energie entfällt auf jeden einzelnen Abschnitt der Kugeloberfläche. Die Oberfläche einer Kugel *A* kann mit Hilfe des Radius *r* und der bekannten Formel $A = 4\pi r^2$ beschrieben werden. Das heißt, die Energie im leeren Raum nimmt mit dem Quadrat der Entfernung ab. Der Pfadverlust im Free Space–Modell errechnet sich nach:

$$PL_{dB} = 20 \cdot log_{10}(\frac{4 \cdot \pi \cdot d}{\lambda})[dB].$$
(5.4)

Das "Free Space"–Modell berechnet die Wellenausbreitung unter Idealbedingungen, welche praktisch nie gegeben sind. Selbst wenn zwischen Sender und Empfänger Sichtverbindung besteht, beeinflussen umstehende Objekte die tatsächliche Ausbreitung des Signals. Für kurze Funkstrecken mit Sichtverbindung liefert das "Free Space"–Modell dennoch halbwegs präzise Distanzschätzungen.

Abbildung 5.8 (a) zeigt den Pfadverlust in Abhängigkeit der Distanz. Im bereits sehr kurzen Abstand zur Sendeantenne beträgt der Pfadverlust ca. 50 *dB*. Dieser Mindestverlust wird als "Minimum Coupling Loss" (*MCL*) bezeichnet. An den verschiedenen Funktionsgraphen ist zu erkennen, dass sich die Freiraumdämpfung bei unterschiedlichen Frequenzen um einen konstanten Faktor unterscheidet. Mit Sender und Empfänger, die einen maximalen Pfadverlust von 90 *dB* tolerieren, kann bei einer Trägerfrequenz von 500 *MHz* eine Distanz von über 1.000 m im Freiraum überbrückt werden. Bei 3.500 MHz sind es gerade 200 m. Das heißt, je niedriger die Frequenz, desto höher ist die mögliche Reichweite. Mit anderen Worten muss bei höher werdender Frequenz die Sendeleistung erhöht werden, um die gleiche Entfernung zu überbrücken.

5.2.2.2 Ground Reflection-Modell

Das "Two Ray Ground Reflection"–Modell, kurz: "Two Ray"–Modell, beschreibt die Wellenausbreitung über einer freien Fläche ohne Hindernisse. Es wird deshalb auch als Ideal Ground Modell bezeichnet. Die Wellen können sich ungehindert ausbreiten und werden lediglich vom Erdboden reflektiert.

Die Leistung des Empfangssignals wird beim "Two Ray"-Modell wie folgt berechnet:

$$P_{RX}(d) = \frac{P_{TX}Gh_{TX}^2h_{RX}^2}{d^4}$$
(5.5)

mit h_{TX}^2 als Höhe des Senders und h_{RX}^2 des Empfängers.

Allerdings kann in diesem Modell der Pfadverlust nicht geringer sein als die entsprechende Freiraumdämpfung, da Verstärkungen des Signals aufgrund von Interferenzen durch Mehrwegeausbreitung, etc. vom Modell nicht berücksichtigt werden. Bei geringen Entfernungen liefert das "Two Ray"–Modell aufgrund der Wellenüberlagerung keine realistischen Ergebnisse; daher wird hier einfach der entsprechende Wert der Freiraumdämpfung angenommen [Rap02]. Der entstehende Grenzwert wird als Cross–over Distanz bezeichnet. Bei dieser Distanz entspricht der berechnete Pfadverlust des "Two Ray"–Modells genau dem des "Free Space"–Modells. Die Cross–over Distanz kann berechnet werden, indem die Free Space Gleichung mit der des "Two Ray"–Modells gleichgesetzt:

$$\frac{P_{TX}Gh_{TX}^2h_{RX}^2}{d^4} = \frac{\lambda}{(4\cdot\pi\cdot d)^2}$$
(5.6)

und nach d_0 aufgelöst wird:

$$d_0 = \frac{4 \cdot h_{TX} \cdot h_{RX}}{\lambda} \tag{5.7}$$

Je nachdem, wie stark der Boden reflektiert, kommen beim Empfänger zwei Wellen an, die sich konstruktiv oder destruktiv überlagern können. Beim "Two Ray"–Modell findet sich die Frequenz in Form der Wellenlänge wieder. Der entscheidende Faktor bei diesem Modell ist aber die Höhe der Antennen (Abbildung 5.8 (b)). Befinden sich Sender und Empfänger in einer Höhe von 50 cm, beträgt der Pfadverlust nach 200 m bereits 112 dB. In Höhe von 2 m verringert sich der Pfadverlust bei gleicher Entfernung auf nur 80 dB. Es ist bei der Einheit Dezibel zu beachten, dass es sich um ein logarithmisches Maß handelt. Eine Differenz von 30 dB entspricht dabei einem linearen Faktor von 1.000. Mit anderen Worten bewirkt die alleinige Höhenänderung der Antenne um 1, 5 m, dass das Signal beim Empfänger mit 1.000-facher Leistung eintrifft. Das bedeutet: Je höher die Sende- und Empfangsantenne, desto höher die Sendereichweite.

Weiterhin sollte für möglichst präzise Simulationen die Beschaffenheit des Erdbodens berücksichtigt werden, da dadurch die Reflexionen maßgeblich beeinflusst werden (vgl. Abschnitt 5.2.1).

5.2.2.3 Log Normal Shadowing–Modell

In der Realität wird man selten Szenarien vorfinden, bei denen sich keine Hindernisse zwischen Sender und Empfänger befinden. Gerade in Indoorumgebungen ist deren Anzahl sehr hoch und unterschiedlicher Struktur. All diese Hindernisse beeinflussen das Signal. Das Shadowing–Modell berücksichtigt diese Hindernisse auf zwei Arten. Erstens in Form eines empirischen Pfadverlust–



Abbildung 5.8: (a) Free Space–Modell für verschiedene Frequenzen (b) Ground Reflection Modell für verschiedene Sender- und Empfängerhöhen [Dei06].

Exponenten und zweitens durch Addition eines normalverteilten Shadowing–Wertes. Der Pfadverlust in Dezibel beträgt:

$$PL[dB] = 10 \cdot \beta \cdot \log_{10}(d) \tag{5.8}$$

 P_{R_0} entspricht dem Pfadverlust aus dem "Free Space"–Modell bei einer Entfernung von einem Meter. β ist der so genannte Pfadverlustexponent. Je nach Umgebungstyp liegt dieser Wert zwischen 1,5 und 6 (siehe Tabelle 5.1). Dabei entspricht ein Pfadverlustexponent von 4 dem "Two–Ray"–Modell. Die Höhe wird nicht berücksichtigt. Ein Pfadverlustexponent von 2 entspricht der Freiraumdämpfung. Ein geringerer Verlustexponent als 2 wäre nur Indoor bei direkter Sichtverbindung denkbar. Das Signal wird

 Tabelle 5.1: Pfadverlustexponenten nach Umgebungstyp.

	Indoor LoS	Free Space	Outdoor	Indoor NLoS
Pfadverlustexponent β	1,5-2	2	2,5-4	2-6

an den Wänden reflektiert und die verschiedenen Signale addieren sich im Idealfall konstruktiv. Besteht keine Sichtverbindung, ist die Dämpfung innerhalb von Gebäuden dafür umso stärker. Jede Wand führt zu einem zusätzlichen Signalverlust von 5 bis 45 *dB*. Laut [Rap02] beträgt der Unterschied zwischen LoS und NLoS durchschnittlich zwischen 25 und 30 *dB*, wobei dieser Wert stark variiert, je nachdem wie viele Wände, Mauern etc. sich zwischen Sender und Empfänger befinden. Dieser Pfadverlustexponent wurde für diverse Szenarien gemessen und ist abhängig von der Beschaffenheit der Wände (Indoor), der Anzahl der Wände, der sonstigen Hindernisse, der Frequenz, der Besiedlungsdichte (Outdoor), der Höhe der Antennen und vielen weiteren Faktoren. Die Vielzahl der Faktoren macht es äußerst schwierig, ein gegebenes Szenario eindeutig in eines der bekannten einzuordnen.

Berücksichtigt man (zufällige) Hindernisse zwischen Sender und Empfänger, so steigt der Pfadverlust über die Entfernung nicht mehr gleichmäßig. Hinter einem Objekt ist der Pfadverlust meist deutlich höher als vor dem Objekt. Aus diesem Grund wird auf den oben berechneten Pfadverlust zusätzlich ein Shadowing Wert X mit Erwartungswert $\mu = 0$ und Standardabweichung σ addiert. Die Shadowing Werte folgen einer linear normierten Gauss–Verteilung. Die Standardabweichung σ wird dabei als Shadowing Deviation bezeichnet. Hat man genaues Wissen über die Topologie, so könnte man rein theoretisch eine Shadowing Deviation von 0 erreichen [Lin96]. Modelliert man die Hindernisse zufällig und ohne Bezug zum tatsächlichen Terrain, dann sollte die Shadowing Deviation sehr hoch gewählt werden (ca. 10 dB bis 12 dB). Bei genauer Kenntnis der Umgebung kann das Sigma kleiner gewählt



Abbildung 5.9: Shadowing–Modell für verschiedene Pfadverluste (a) ohne Shadowing Deviation und (b) mit Shadowing Deviation $\sigma = 8 dB$ [Dei06].

werden, d. h., je höher das Sigma, desto ungenauer ist das Modell. Typische Werte für Sigma liegen zwischen 3 dB und 12 dB.

$$\left[\frac{P_r(d)}{P_r(d_0)}\right]_{dB} = -10\beta \log\left(\frac{d}{d_0}\right) + X_{dB}$$
(5.9)

Wie aus Abbildung 5.9 (a) ersichtlich, entspricht ein Pfadverlustexponent von $\beta = 2$ genau dem Free Space Loss. Je höher der Pfadverlustexponent, desto stärker die Dämpfung. Ohne das zufällige, normalverteilte Sigma verläuft der Pfadverlust gleichmäßig. Wird es hingegen berücksichtigt, ergibt sich Abbildung 5.9 (b), d. h., das Signal fluktuiert über die Entfernung. Möchte man eine robuste Funkverbindung haben, so sollte die Empfindlichkeit des Empfängers oberhalb des Funktionsgraphen liegen.

5.2.3 Grundprinzip des eACL

Die in dem vorangegangenen Abschnitt diskutierten Faktoren beeinflussen die Distanzschätzung mittels RSSI in hohem Maße. Gerade in Indoorszenarien wirken sich Reflexionen und Dämpfungen stark auf die Signalempfangsstärke aus. Aus diesem Grunde wird der vorliegende *ACL*–Algorithmus für Indooranwendungen angepasst. Das Signal beeinflussende Hindernisse sollen bei der Distanzschätzung berücksichtigt werden. Befindet sich ein Hindernis im Ausbreitungspfad, wird das Signal abgeschwächt. Wenn dieser Einfluss nicht in die Streckenberechnung aus den RSSI–Werten mit einfließt, wird die Distanz zu groß bestimmt. Abbildung 5.10 illustriert diesen Sachverhalt. Daraus folgt



Abbildung 5.10: Lokalisierungsfehler aufgrund eines Hindernisses im Signalpfad.

weiterhin, dass die berechnete Position grob fehlerhaft wird. Im ungünstigsten Fall kann es passieren, dass kein Bogenschnitt gerechnet werden kann und die Lokalisierung fehlschlägt. Dahingestellt sei, ob der ungünstige Fall nicht doch günstiger ist, bevor sich auf eine grob fehlerhafte Position verlassen wird. Im optimalen Fall erkennt der Sensorknoten grob fehlerhafte RSS–Messwerte. Diese können dann entweder verworfen oder korrigiert werden, um an der endgültigen Positionsbestimmung teilzunehmen. In Abschnitt 5.1.2 wurde eine Footprintkarte definiert, auf der mögliche Sensorintervalle verzeichnet sind. Diese Karte kann nun mit möglichen Hindernissen (Wände, Türen, Einrichtungsgegenstände,...) erweitert werden, die gegebenenfalls vorhandenen Gebäudeplänen oder ähnlichem zu entnehmen sind. Bei der Distanzschätzung kann dieses Hindernis mit in die Berechnung einfließen und der fehlerhafte RSS–Wert korrigiert werden.

Der erweiterte Algorithmus kann wie folgt formuliert werden [BSB10, BNSB10]. Lokalisiert sich der Sensorknoten wie in Abschnitt 5.1.2 beschrieben, wird die Position durch den *ACL* getestet. Fällt die jeweilige Position durch den Test, wird untersucht, ob sich ein auf der Footprintkarte verzeichnetes Hindernis im Signalpfad befindet. Kann keines identifiziert werden, wird die Position als Ausreißer verworfen und nach Abschnitt 5.1.2 verfahren. Befindet sich jedoch ein Hindernis im Signalpfad, werden die korrespondierenden Distanzschätzungen mit dem Hindernis verbessert. Danach wird der *ACL*–Test wiederholt.

Da als Dämpfungsmodell das Shadowing-Modell zum Einsatz kommt, ist für die erfolgreiche Anwendung des *eACL* erforderlich, dem jeweiligen Hindernis ein bestimmtes, charakteristisches Dämpfungsvermögen zuordnen zu können. Wie in Abschnitt 5.2.1 beschrieben, ist dieses Dämpfungsvermögen jedoch materialabhängig. Für die Footprintkarte heißt das im Detail, dass der Dämpfungswert mit gespeichert werden muss. Eine andere Möglichkeit wäre, die Dämpfung vom Sensornetzwerk bestimmen zu lassen. Befindet sich ein Hindernis zwischen zwei Beacons, kann über die gemessene und die aus Koordinaten bekannte Distanz auf einen Dämpfungswert geschlossen werden. Allerdings wäre dann ein Fehler zu erwarten, der sich durch Interferenz der Mehrwegekomponenten ergibt. In den folgenden Simulationen wird eine andere Herangehensweise benutzt. In Rappaport [Rap02] werden detaillierte und umfangreiche Messszenarien zu diesem Thema beschrieben. Aus intensiven Messungen wurden für vielfältige Materialien Dämpfungswerte bestimmt, auf die in diesen Simulationen zurückgegriffen werden soll.

Material	PAF	η	σ	Frequenz
	[dB]		[dB]	[MHz]
Betonwand	15	3,0	7,0	1.300
Betonwand mit Stahlbewehrung 1	20	2,4	9,6	1.300
Betonwand mit Stahlbewehrung 2	20	3,0	7,0	1.300
Holzwand	6	3,0	5,0	1.300
Ziegelwand	10	3,0	7,0	1.300

 Tabelle 5.2: Für die Simulation genutzte Parameter nach [Rap02].

Tabelle 5.2 listet die für die Simulationen verwendeten Materialen und deren Dämpfungseigenschaften auf. An dieser Stelle ist zu erwähnen, dass der *eACL* auch mit der ersten beschriebenen Methode funktioniert.

Als Ausbreitungsmodell wurde das "Path Attenuation Factor"–Modell (*PAF*) [Rap02] benutzt, welches auf dem "Path Loss"–Modell basiert. Bezogen auf Gleichung 5.9 heißt das, es wird der Term *X* durch einen materialspezifischen Wert, dem "Path Attenuation Factor" *PAF*, ersetzt. Befinden sich zwischen Sender und Empfänger mehrere Hindernisse, so werden die charakteristischen PAF–Werte aufsummiert. Daraus folgt sich Gleichung 5.10 für das Path Attenuation Model:

$$\left[\frac{P_r(d)}{P_r(d_0)}\right]_{dB} = -10\beta \log\left(\frac{d}{d_0}\right) + \sum PAF$$
(5.10)

Diese PAF-Werte müssen auf der Footprintkarte den jeweiligen Hindernissen zugeordnet sein.

5.2.4 Simulation

Es wurden verschiedene Simulationen durchgeführt. An dieser Stelle soll sich auf ein Szenario beschränkt werden. Der Simulationsaufbau ist wie folgt: Es wird ein Teil eines Gebäudestockwerks, bestehend aus zwei Räumen und einem Flur simuliert (Abbildung 5.11). Die Dimensionen für Büroraum 1 betragen $4m \times 6m$, für Laborraum $2.6m \times 6m$ und der Korridor besitzt die Ausmaße $10m \times 4m$. Des Weiteren werden Intervalle für die Sensormessungen definiert. Gemessen wird die Temperatur in den Räumen. Für den Laborraum 2 wird eine exakte Temperatur von 23°C gemessen. Büroraum 1 besitzt eine Temperatur zwischen $19 - 22^{\circ}$ C und der Korridor von $17 - 19^{\circ}$ C. Wie bereits im Kapitel 5.1.2 beschrieben, ist Voraussetzung für die erfolgreiche Anwendung von ACL, dass die Intervalle eindeutig zuweisbar sein müssen. Aus diesem Grund werden die Intervalle als konstant angenommen. Weiterhin werden fünf Beacons ausgebracht. Davon befinden sich drei im Korridor und jeweils einer in den verbleibenden Räumen. Das führt zu $\binom{n}{3} = 10$ möglichen Trilaterationen. Jede von diesen wird 10.000 Mal wiederholt, um empirische Einflüsse auszuschließen. Es werden die Simulationen für unterschiedliche Materialien durchgeführt (Tabelle 5.2). Die Strecken werden mit den typischen Dämpfungswerten für die unterschiedlichen Materialien verfälscht. Darüber hinaus ist jede Distanz mit einem zufälligen Streckenfehler überlagert, der Reflexionen an den Hindernissen beinhaltet. Den Beacons wird eine Sendereichweite von maximal 100 m zugewiesen. Wird ein Signal demzufolge um einen Betrag gedämpft, der einer Entfernung von mehr als 100 m entspricht, gilt diese Distanz als nicht



Abbildung 5.11: Simulationsaufbau für eACL.

messbar. Da der *eACL* im jetzigen Stand ausschließlich Dämpfungseinflüsse berücksichtigt, ist in den Simulationen ein relativ großer Anteil ungültiger Positionen, d. h., Ergebnisse, die nach Anwendung von *eACL* nicht im richtigen Sensorintervall lokalisiert werden, zu erwarten.

In weiteren Simulationen wird der Einfluss der Beaconanzahl untersucht. Bei einer höheren Anzahl verfügbarer Beacons werden auch mehr Trilaterationen möglich. Dabei erhöht sich die Beaconanzahl schrittweise um einen Festpunkt, wobei die vorigen in ihrer Position festgehalten werden.

5.2.5 Simulationsergebnisse

Die folgenden Simulationen verlaufen in unterschiedlichen Schritten. In den ersten Untersuchungen werden die unterschiedlichen Materialien betrachtet. In einem zweiten Schritt wird zusätzlich die Beaconanzahl schrittweise erhöht und analysiert, wie sich diese auf die Genauigkeit von *ACL* mit Mittelung und *eACL* auswirkt. Als abschließende Simulation erfolgt eine schrittweise Verkleinerung der Sensorintervalle, was gleichbedeutend mit einer Zunahme von Hindernissen (Wänden, etc.) ist.

5.2.5.1 Einfluss unterschiedlicher Materialeigenschaften der Hindernisse

Die Simulationsergebnisse sind in der Tabelle 5.3 für Laborraum 2 und in der Tabelle 5.4 für Büroraum 1 wiedergegeben. In Laborraum 2 ist bei allen Trilaterationen pro Distanz maximal eine Wand zu durchlaufen. Abbildung 5.12 zeigt exemplarisch die Positionen aller Knoten für (a) Holzwände und (b) Betonwände mit Stahlbewehrung der Kategorie zwei. Es ist zu erkennen, dass im Anwendungsfall der Holzwand, und damit mit dem geringsten Dämpfungspotenzial, die Positionen nach Anwendung von *ACL* und *eACL* ziemlich gut übereinstimmen. Von den 10.000 Wiederholungen lieferte *ACL* 6,7% oder 670 gültige Positionen für die abschließende Trilateration. Die verbleibenden 93,3% der Positionen liegen in dem falschen Sensorintervall und werden als Ausreißer ausgeschlossen. Diese Ausreißer sollen nun dem *eACL* unterworfen werden. Mit den verbesserten Distanzen werden erneut Trilaterationen gerechnet und erneut dem *ACL* unterzogen. Im Fall der Holzwand ergeben sich nun inklusive der vorherigen Positionen 14, 98% oder 1.498 gültige, für den Fall der Betonwand, für die hier angeführten



Abbildung 5.12: Simulationsergebnis eine Wand: (a) für Holzwand (b) für Betonwand mit Stahlbewehrung 2.

Beispielmaterialien das mit der höchsten Dämpfung, errechnet *ACL* 1, 14 % oder 114 gültige Positionen. Nach Anwendung von *eACL* steigt deren Zahl auf 10, 12 % oder 1.012. Obwohl die Zahl der gültigen

Material	Laborraum 2							
	Gültig nach <i>ACL</i> [%]	Gültig nach <i>eACL</i> [%]	Positions- fehler nach ACL [%]	Positions- fehler nach eACL [%]				
Betonwand	2,19	10,42	0,73	0,20				
Betonwand mit Stahlbewehrung 1	1,04	8,55	0,66	0,32				
Betonwand mit Stahlbewehrung 2	1,14	10,12	0, 89	0,21				
Holzwand	6,76	14,98	0,75	0,71				
Ziegelwand	3,96	10,97	0,63	0, 39				

abelle 5.3: Simulationsergebnisse zu eACL in Laborraum 2.

Positionen nahezu verzehnfacht werden kann, scheint ein wesentlicher Anteil der Fehlereinflüsse auf die hier nicht beachteten RSSI–Effekte (vgl. Abschnitt 5.2.1) zu entfallen.

Die unterschiedlich starken Einflüsse der Dämpfung finden sich auch hier wieder. Bei der schwächer dämpfenden Holzwand sind die Positionsfehler im selben Bereich erkennbar, wohingegen die Position bei der Betonwand signifikant verbessert werden kann.

In Büroraum 1 (Abbildung 5.13 (a) und (b)) hingegen müssen zu den zwei Beacons im östlichen Teil des Korridors zwei Wände passiert werden. Die Ergebnisse sind in der Tabelle 5.4 aufgeführt. Die gültigen Positionen nach *ACL* nehmen mit zunehmender Dämpfung teilweise dramatisch ab. Bei den



Abbildung 5.13: Simulationsergebnis zwei Wände: (a) für Holzwand (b) für Betonwand mit Stahlbewehrung 2.

Material	Büroraum 1							
	Gültig nach <i>ACL</i> [%]	Gültig nach <i>eACL</i> [%]	Positions- fehler nach ACL [%]	Positions- fehler nach eACL [%]				
Betonwand	0,80	6,23	1,03	0,69				
Betonwand mit Stahlbewehrung 1	0,28	4,73	1,19	0,72				
Betonwand mit Stahlbewehrung 2	0,36	5,97	1,05	0,67				
Holzwand	3,44	9,42	0,89	0,74				
Ziegelwand	1,82	6,92	1,03	0,87				

Tabelle 5.4: Simulationsergebnisse zu eACL in Büroraum 1.

Betonwänden mit Stahlbewehrung 1 und 2 sind lediglich 0, 36 % bzw. 0, 28 % der ermittelten Positionen nach *ACL* gültig. Für schwach dämpfende Materialien hingegen ist eine geringere Abnahme der gültigen Positionen festzustellen. Nach der Anwendung von *eACL* ist weiterhin eine Abnahme der gültigen Positionen im Vergleich zur Konfiguration mit nur einer Wand zu verzeichnen. Jedoch fällt diese weitaus gutmütiger aus. Verglichen mit der vorigen Simulation stehen noch immer mehr als 50 % der Positionen für eine endgültige Koordinatenberechnung zur Verfügung.

Die Abbildungen 5.14 (a) und (b) zeigen den Lokalisierungsfehler in Abhängigkeit vom PAF. Zusätzlich zu den Positionsfehlern ist der Fehler aufgeführt, der sich ohne Anwendung von *ACL* und *eACL* ergibt. In Abbildung 5.14 (a) befindet sich der zu lokalisierende Sensorknoten in Laborraum 2, d. h., es befindet sich jeweils ein Hindernis zwischen Beacon und Knoten. In Abbildung 5.14 (b) hingegen ist der



Abbildung 5.14: Lokalisierungsfehler über die Dämpfung der einzelnen Materialien.

Sensorknoten in Büroraum 1 lokalisiert. Damit werden zwei Distanzen von zwei Wänden blockiert. Es ist zu erkennen, dass der Lokalisierungsfehler deutlich mit der Dämpfung korreliert. Durch die Anwendung von *eACL* kann der Lokalisationsfehler gegenüber *ACL* bis zu einem Drittel gesenkt werden. Gerade bei stark dämpfenden Materialen lässt sich eine signifikante Verbesserung feststellen.

5.2.5.2 Einfluss steigender Beacons

Weiterhin soll der Einfluss einer steigenden Beaconanzahl untersucht werden. Zusätzliche Beacons sind in allen Raumecken und -mitten platziert. Bei jedem Durchlauf wird ein weiterer Beacon in die Rechnung übernommen. Abbildung 5.15 zeigt die endgültige Beaconanordnung. Die Nummerierung gibt die Reihenfolge wieder, in der die Festpunkte mit in die Berechnung einfließen. Die Ergebnisse der Lokalisierung sind in Abbildung 5.16 (a) für Laborraum 2 und (b) für Büroraum 1 nach Abbildung 5.15 dargestellt. In Laborraum 2 kommt nun mit Beacon 7 ein Festpunkt hinzu, bei dem das Signal für die Streckenmessung ebenfalls zwei Wände durchläuft. Ist der Sensorknoten dagegen in Büroraum 1 platziert, müssen zu den Beacons 4, 5 und 6 mehrere Wände durchmessen werden. Die Positionsfehler für ACL und eACL verlaufen wie erwartet. Die steigende Anzahl der Beacons hat einen relativ kleinen Einfluss auf die Positionsgenauigkeit, da ausschließlich Positionen innerhalb des Sensorintervalls an der endgültigen Koordinatenberechnung teilnehmen. In Büroraum 1 (Abbildung 5.16 (b)) ergeben sich bei der höchsten Anzahl von Beacons Positionsfehler von 1, 12 m und 0, 67 m für ACL und eACL. Für die Positionsbestimmung mit allen verfügbaren Trilaterationen ergibt sich hingegen ein anderes Bild. Der Positionsfehler steigt von 4,3 m bei der Minimalkonfiguration mit drei Beacons auf einen Positionsfehler von 12,7 m bei sechs Beacons, bevor der Fehler danach sukzessive auf 1,9 m sinkt. Dieser Effekt lässt sich auf die hohe Anzahl möglicher Trilaterationen zurückführen. Bei 12 Beacons sind bereits $\binom{n}{3} = 220$ Kombinationen möglich, von denen nur noch 27 von zwei Wänden gedämpft werden. Darüber hinaus fließen mit den Beacons 4 bis 6 diejenigen in die Simulation ein, die durch zwei Wände vom Sensorknoten getrennt sind. Die korrespondierenden RSS-Werte werden hier am meisten gedämpft. Die folgenden Beacons sind hingegen nur noch durch eine Wand getrennt oder aber sie



Abbildung 5.15: Steigende Anzahl Beacons für eACL.



Abbildung 5.16: Lokalisierungsfehler über steigender Beaconanzahl.

liegen im selben Raum wie der Unbekannte. Mit der Hinzunahme des Beacons 8 ergeben sich zumeist ungünstige Schnittbedingungen, weswegen der Fehler wieder ansteigt. Für die nachfolgenden Beacons 9 bis 12 werden auch diese besser, weswegen der Positionsfehler wieder kleiner wird. Die endgültigen, durch *ACL* und *eACL* verbliebenen Trilaterationen werden von den Schnittgeometrien nur in geringem Maße beeinflusst, da diese in anderen Sensorintervallen lokalisiert und somit herausgefiltert werden.

Für Laborraum 2 liefern die Positionsberechnungen Fehler für *ACL* von 0,96 m und 0,73 m für *eACL*. Es ergibt sich ein vergleichbares Bild (Abbildung 5.16 (a)). Auch hier hat die Schnittgeometrie einen großen Einfluss auf die Positionsgenauigkeit, wenn die Trilaterationen ohne *ACL* und *eACL* ablaufen. Mit zunehmender Beaconanzahl verbessert sich die Lokalisierung. Die Anwendung beider Algorithmen kann die endgültige Position um Faktor 4 verbessern. Bei ungünstiger Schnittgeometrie kann sogar

eine Verbesserung von bis zu Faktor 30 erreicht werden, da die fehlerhaften Positionen verlässlich herausgefiltert werden und nicht in die endgültige Berechnung einfließen.

5.2.5.3 Einfluss bei Erhöhung der Anzahl der Sensorintervalle

In einer weiteren Simulation wird der Einfluss zunehmender Sensorintervalle untersucht. Die Abbildung 5.17 (a) verdeutlicht den Sachverhalt. Dazu wird nach jedem Simulationsdurchlauf dem gegebenen



Abbildung 5.17: Steigende Anzahl Sensorintervalle für eACL durch Hinzunahme von zusätzlichen Hindernissen (a) und daraus resultierende Positionsfehler (b).

Szenario ein neues Hindernis, in diesem Fall eine neue Wand, hinzugefügt. Das heißt weiterhin, dass die Gebiete desselben Intervalls auch kleiner werden. Als Material für die Hindernisse dient die Betonwand mit Stahlbewehrung 2 aus Tabelle 5.2. Es ist zu erwarten, dass die Zunahme der Sensorintervalle bei deren gleichzeitiger Verkleinerung einen Einfluss auf die Position hat. Durch die Zunahme von Hindernissen wird sich die Qualität der Positionsschätzung ohne *ACL* und *eACL* verschlechtern, da die Streckenmessungen in noch höherem Maße verfälscht werden. Im ungünstigsten Fall wird eine Lokalisierung unmöglich, da die resultierenden drei Strecken nicht in einem Punkt schneiden, das System divergiert. Zum anderen können mehr Ausreißer durch *ACL* und *eACL* detektiert werden. Bei *ACL* besteht die Möglichkeit, dass auch hier keine Position ermittelt werden kann. Dieser Effekt kann auftreten, wenn durch die starke Streckenverfälschung keine Position im gültigen Sensorintervall berechnet wird. Durch die Anwendung von *eACL* sollte es jedoch möglich sein, gültige Positionsschätzungen zu erhalten. Abbildung 5.17 (b) gibt einige Simulationsergebnisse wieder.

Die Simulation bei der Verkleinerung der Sensorintervalle bei gleichzeitiger Zunahme dieser zeigt die erwarteten Ergebnisse. Der Fehler der Positionsschätzung ohne *ACL* und *eACL* wird mit zunehmender Anzahl von Sensorintervallen größer. Zwischen den Beacons und dem Unbekannten kommen nach jedem Simulationsdurchlauf neue Hindernisse hinzu. Es ist jedoch möglich, in jedem Szenario Bogenschnitte zu berechnen. *ACL* zeigt, dass zudem auch Positionen in dem gültigen Sensorintervall berechnet werden (Tabelle 5.5). Die Anwendung von *eACL* hat zur Folge, dass mehr Streckenmessungen gültige Schätzpositionen liefern, die in die endgültige Positionsberechnung einfließen. In dem

Algorithmus	Anzahl gültiger Positionen [%]						
	1	2	3	4	5		
ACL	1,71	1,17	0,98	0,65	0,21		
eACL	9,78	8,41	7,50	4,69	2,47		

Tabelle 5.5: Anzahl gültiger Positionen über ansteigende Sensorintervalle.

Szenario mit den meisten Hindernissen kann die Anzahl im Gegensatz zu *ACL* sogar verzehnfacht werden, was sich auch in der erreichbaren Positionsgenauigkeit wiederspiegelt (Abbildung 5.17). Durch *eACL* erhöhte sich die daraus resultierende Genauigkeit auf 0, 29 m. Die alleinige Anwendung von *ACL* liefert hingegen nur eine Genauigkeit von 0, 72 m. Es konnte demzufolge eine Steigerung um Faktor 2,5 erreicht werden. Eine Lokalisierung über alle verfügbaren Positionen lieferte hingegen einen Positionsfehler von 6, 28 m. Bei einer Raumgröße von $3 m \times 4 m$, und damit der endgültigen Größe des aktuellen Sensorintervalls, ist der Positionsfehler sogar größer als das Sensorintervall. Es kann davon ausgegangen werden, das eine Mittelung über alle verfügbaren Trilaterationen keine Positionen im gültigen Sensorintervall liefert.

5.2.6 Einschränkungen für den eACL-Algorithmus

Die Materialbeschaffenheit von Hindernissen hat einen großen Einfluss auf die Entfernungsschätzung mittels RSS–Messungen. Die Anwendung des hier vorgestellten "extended Anomaly Correction in Localization"–Algorithmus (*eACL*) kann diese Effekte für die Positionsbestimmung verringern. Allerdings beruht diese Technik ausschließlich auf der Berücksichtigung von Dämpfungen durch Hindernisse. In der Realität beeinflussen jedoch andere Effekte die Signalausbreitung und sollten mit in die Kalkulation einfließen. Die Positionsgenauigkeit könnte damit weiter erhöht werden.

Eine Erweiterung um Streuungseffekte, Reflexionen und Beugung gestaltet sich jedoch schwierig. Jede Schätzposition, die mittels Trilateration falsch geschätzt wird, weist auch ein anderes Signalempfangsbild auf. Durch Interferenz von Wellen können sich die RSS–Werte an den unterschiedlichen Positionen erhöhen oder auslöschen. Eine Simulation dieser Effekte würde sich als schwierig herausstellen. Eine Möglichkeit, diese Effekte zu berücksichtigen, wäre eine detailliertere Footprintkarte, die zusätzlich zu den Sensorintervallen und Hindernissen auch repräsentative RSS–Werte für bestimmte Regionen enthält. Diese Technik ist allerdings nur theoretisch denkbar, da in realen Szenarien Bewegungen von Objekten und Personen etc. die RSS–Werte verändern und diese an den repräsentativen Orten nicht mit denen auf der Footprintkarte übereinstimmen würden.

Als weitere Einschränkung von *eACL* muss erwähnt werden, dass er nicht auf dynamische Veränderung der Umgebung reagiert. So kann die Hinzu- oder Wegnahme von Hindernissen dazu führen, dass mögliche, das Signal dämpfende Körper nicht oder fälschlicherweise in die Berechnung mit einfließen und das Ergebnis der Positionsbestimmung verfälschen. Jedoch ist davon auszugehen, dass die Positionsschätzung unter Anwendung von *eACL* und *ACL* genauer ausfällt als die Berechnung ohne Berücksichtigung der Sensorintervalle, da grob fehlerhafte Trilaterationen ausgeschlossen werden können. Eine Positionsbestimmung mittels *ACL* und *eACL* wird immer Ergebnisse liefern, die innerhalb des Sensorintervalls liegen, sofern es richtig bestimmt ist und eindeutig zugewiesen werden kann.

Da der *eACL* auf dem *ACL* beruht, treffen die in 5.1.5 formulierten Einschränkungen ebenfalls auf den *eACL* zu.

5.3 Fazit

In diesem Kapitel wurden zwei Algorithmen vorgestellt, die unter Verwendung von Sensormesswerten eine Ausreißerdetektion bei durchgeführten Trilaterationen erlaubt. Dabei sind s. g. erwartete Sensorintervalle definiert und mit tatsächlich gemessenen Sensormesswerten verglichen worden. Dieser Ansatz wurde in einem zweiten Schritt auf Indoorszenarien ausgeweitet. Bei diesem Algorithmus werden Hindernisse und deren Einfluss auf die Ausbreitungseigenschaften elektromagnetischer Signale in das Modell integriert. Nachfolgend werden diese fehlerbehafteten Distanzmessungen auf dieser Grundlage korrigiert und einer erneuten Positionsbestimmung zugeführt. Mit beiden Ansätzen konnte die Genauigkeit der Positionsschätzung signifikant gesteigert werden.

Die in diesem Kapitel eingeführten Algorithmen besitzen ein hohes Erweiterungspotenzial und sollten in weiteren Arbeiten fortgesetzt werden. So wäre denkbar, die Algorithmen als Nebenbedingung in eine Ausgleichung einzuführen. Ebenfalls können Map-Matching-Verfahren genutzt werden, um die Sensorintervalle optimaler auszunutzen. Aufgrund des zu erwartenden hohen Umfangs werden beide Ansätze in dieser Arbeit nicht weiter diskutiert. Vielmehr wird im nächsten Kapitel ein feinkörniger Lokalisierungalgorithmus vorgestellt, der bei geringem Berechnungs- und Kommunikationsaufwand eine hohe Positionsgenauigkeit zulässt.

6 Entwicklung von Lokalisierungsalgorithmen

"Ich glaube, daß es, im strengsten Verstand, für den Menschen nur eine einzige Wissenschaft gibt, und diese ist reine Mathematik. Hierzu bedürfen wir nichts weiter als unseren Geist."

> (Georg Christoph Lichtenberg, dt. Mathematiker u. Physiker)

6.1 "Resource Aware Localization"–Algorithmus

Wie bereits in Kapitel 2 beschrieben, ist der Energievorrat die limitierende Größe in drahtlosen Sensornetzwerken. Um eine möglichst lange Lebensdauer der Sensornetze zu garantieren, steht die Erforschung energieeffizienter Hard- und Software im Fokus der Entwicklung. Speziell für die Lokalisierung soll die Komplexität exakter Lokalisierungsalgorithmen dahingehend angepasst werden, bei gleichbleibender Genauigkeit den Rechen- und Kommunikationsaufwand auf den Sensorknoten zu minimieren. Aus diesem Grunde wurde der "Resource Aware Localization"–Algorithmus (*RAL*) entwickelt, der im Prinzip an den im Kapitel 4 beschriebenen und im weiteren Verlauf im Kapitel 7.2.2 ausführlich untersuchten "Distributed Least Squares"–Algorithmus (*DLS*) angelehnt ist. Aufgrund des veränderten mathematischen Modells ergeben sich jedoch andere Eigenschaften, die in diesem und im nächsten Kapitel diskutiert werden sollen.

RAL basiert auf den in Kapitel 3 beschriebenen Grundlagen der geodätischen Ausgleichungsrechnung. Dabei werden die verfügbaren Beacons in dem Sensornetzwerk genutzt und mittels der Euklidischen Distanzen vom zu lokalisierenden Sensorknoten zu den Beacons ein überbestimmtes, nichtlineares Gleichungssystem aufgebaut. Bei einer hohen Anzahl von Beacons ergibt sich somit ein großes Gleichungssystem, was bei einer vollständigen Berechnung auf jedem einzelnen Sensorknoten zu einem hohen Rechenaufwand und hoher Redundanz führen würde. Darüber hinaus werden bei der bekannten Linearisierung mittels Taylorreihenentwicklung und Abbruch nach dem ersten Glied (vgl. Kapitel 3.2.1) Näherungskoordinaten für den unbekannten Sensorstandort verwendet, die apriori nicht verfügbar sind. Eine denkbare Möglichkeit wäre der Einsatz approximativer Lösungsmethoden als Startwertbestimmung für die nachfolgende Lösung des überbestimmten Gleichungssystems (Geichung 6.1). Diese Möglichkeit wurde in Born et al. [BRBT06] für einige approximative Verfahren untersucht. Diese steigert jedoch den Berechnungs- und Kommunikationsaufwand im Sensornetz. An dieser Stelle wird dazu ein anderer Weg bestritten.

6.1.1 Lösen der Lokalisierungsaufgabe

Der mathematische Ausgangspunkt von *RAL* ist die Formulierung der Beobachtungsgleichungen auf Basis der Euklidischen Distanzen (Gleichung 6.1).

$$(\hat{x}_N - \tilde{x}_i)^2 + (\hat{y}_N - \tilde{y}_i)^2 = \hat{r}_i^2 \quad (i = 1, 2, \dots, n).$$
(6.1)

Das Auflösen der binomischen Formeln führt zu:

$$\left(\tilde{x}_{i}^{2}+\tilde{y}_{i}^{2}\right)-2\tilde{x}_{i}\hat{x}_{N}-2\tilde{y}_{i}\hat{y}_{N}+\left(\hat{x}_{N}^{2}+\hat{y}_{N}^{2}\right)=\tilde{r}_{i}^{2}.$$
(6.2)

Durch Umstellen erhält man das weiterhin nicht lineare Gleichungssystem:

$$-2\tilde{x}_{i}\hat{x}_{N} - 2\tilde{y}_{i}\hat{y}_{N} + \left(\hat{x}_{N}^{2} + \hat{y}_{N}^{2}\right) = \hat{r}_{i}^{2} - \left(\tilde{x}_{i}^{2} + \tilde{y}_{i}^{2}\right)$$
(6.3)

und durch Einführen einer Hilfsunbekannten $\hat{w} = (\hat{x}_N^2 + \hat{y}_N^2)$ sowie durch Substitution von $(\tilde{x}_i^2 + \tilde{y}_i^2) = K_i$ folgt das lineare Gleichungssystem [MS77]:

$$\begin{array}{rclcrcl}
-2\tilde{x}_{1}\hat{x}_{N}-2\tilde{y}_{1}\hat{y}_{N}+\hat{w}&=&r_{1}^{2}-K_{1}&=&b_{1}\\
-2\tilde{x}_{2}\hat{x}_{N}-2\tilde{y}_{2}\hat{y}_{N}+\hat{w}&=&r_{2}^{2}-K_{2}&=&b_{2}\\
&\vdots\\
-2\tilde{x}_{n}\hat{x}_{N}-2\tilde{y}_{n}\hat{y}_{N}+\hat{w}&=&r_{n}^{2}-K_{n}&=&b_{n}.
\end{array}$$
(6.4)

Dieses Gleichungssystem kann nunmehr in die Matrizenschreibweise Ax = b überführt werden:

$$\mathbf{A} = \begin{pmatrix} -2\tilde{x}_1 & -2\tilde{y}_1 & 1\\ -2\tilde{x}_2 & -2\tilde{y}_2 & 1\\ \vdots & \vdots & \vdots\\ -2\tilde{x}_n & -2\tilde{y}_n & 1 \end{pmatrix}, \mathbf{x} = \begin{pmatrix} \hat{x}_N\\ \hat{y}_N\\ \hat{w} \end{pmatrix}, \mathbf{b} = \begin{pmatrix} \hat{b}_1\\ \hat{b}_2\\ \vdots\\ \hat{b}_n \end{pmatrix}.$$
(6.5)

6.1.2 Normalgleichungen

Dieses Gleichungssystem kann unter Anwendung der Methode der kleinsten Quadrate gelöst werden. Wie in Kapitel 3.3 dargelegt, kommen dazu unterschiedliche Verfahren zum Einsatz. Vorerst wird hier die Lösung durch die Normalgleichungen (vgl. Kapitel 3.4.1) beschrieben. Im Folgenden wird die gesamte Berechnung des Gleichungssystems (6.5) vom Aufstellen der Matrizen bis hin zur Lösung des Unbekanntenvektors als Standardlokalisierung bezeichnet. Durch die gewählte Umformung (Gleichung 6.3) des nichtlinearen Gleichungssystem ergeben sich folgende neue Eigenschaften, die für die spätere Formulierung des neuen Algorithmus benutzt werden können:

- 1. Die Elemente der Koeffizientenmatrix **A** berechnen sich ausschließlich aus bekannten Beaconpositionen $\tilde{B}_i(\tilde{x}_i, \tilde{y}_i)$.
- 2. Vektor **b** beinhaltet mit **K** ebenfalls bekannte Beaconpositionen $\tilde{B}_i(\tilde{x}_i, \tilde{y}_i)$, die vor der Berechnung zur Verfügung stehen.
- 3. Da die Hilfsunbekannte \hat{w} für die Bestimmung der endgültigen Position (\hat{x}_N, \hat{y}_N) nicht gebraucht wird, kann diese aus der Berechnung entfernt werden. Dazu gibt es mehrere Möglichkeiten, auf

die in Kapitel 7.2.3 detailliert eingegangen wird. Zunächst soll eine Unabhängigkeit von \hat{x} , \hat{y} und \hat{w} angenommen und die dritte Zeile in $\mathbf{A}_{\mathbf{p}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ nicht mit übermittelt werden, um den Kommunikationsaufwand so gering wie möglich zu halten.

Diese Eigenschaften erlauben es nun, die Berechnung des Lösungsvektors $\hat{\mathbf{x}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$ in Analogie zum *DLS*–Ansatz aufzuspalten, denn es ergibt sich mit den bekannten Beaconkoordinaten $\tilde{B}_i(\tilde{x}_i, \tilde{y}_i)$ eine Möglichkeit zur Vorberechnung des komplexen Teils $\mathbf{A}_p = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ und zur weniger komplexen Nachberechnung $\mathbf{A}_p \mathbf{b}$. Wie in Kapitel 4.1 bereits erwähnt, besteht ein Sensornetzwerk aus extrem energielimitierten Sensorknoten, einigen ressourcenstärkeren Beacons und der Senke, die keinen Limitationen hinsichtlich Energie-, Rechen- und Speicherleistung unterliegt. Ziel ist es, so wenig Energie wie möglich auf den Sensorknoten zu verbrauchen. Mit der vorgenommenen Aufspaltung der Berechnung in Vor- und Nachberechnung ist es möglich, die komplexe Vorberechnung auf die Senke auszulagern, ohne dabei die Sensorknoten einzubeziehen. Da der Vektor **b** neben Teilen der Vorberechnung auch aus den Distanzen des Sensorknotens zu den Beacons \hat{r}_i besteht, ist es nicht möglich, diesen Teil der Berechnung auszulagern. Auf den Sensorknoten bleibt damit nur die weniger komplexe Nachberechnung $\mathbf{x} = \mathbf{A}_p(\mathbf{\hat{r}}_i^2 - \mathbf{K}_i)$. Unter der Voraussetzung, dass jeder Sensorknoten in der Lage ist, eine Distanz zu jedem Beacon zu bestimmen, lassen sich zusammenfassend zwei Vorteile für dieses Vorgehen benennen:

- 1. Da die Vorberechnung nur aus Beaconkoordinaten $\tilde{B}_i(\tilde{x}_i, \tilde{y}_i)$ besteht, ist für das gesamte Netzwerk die einmalige Berechnung von A_p und K ausreichend.
- 2. Auf jedem Sensorknoten muss ausschließlich die weniger komplexe Nachberechnung $\hat{x} = A_p(\hat{r}_i^2 K_i)$ ausgeführt werden, was zu einer geringeren energetischen Belastung eines jeden einzelnen Knotens führt.

Damit erfüllt dieser Schritt zwei der in Kapitel 1.2 formulierten Ziele für Lokalisierungsalgorithmen in drahtlosen Sensornetzwerken. Er läuft zum einen verteilt ab und schont dabei die Ressourcen der energielimitierten Sensorknoten.

Bevor der Ablauf des Algorithmus skizziert wird, soll auf die Berechnung mittels Orthogonalisierungsmethoden eingegangen werden, was mehrere Vorteile mit sich bringt.

6.1.3 Orthogonalisierungsverfahren

Wie in Abschnitt 3.4 beschrieben, kann die Anwendung der Normalgleichungen zu numerischen Instabilitäten führen und die Lokalisierung unmöglich machen. Obwohl diese Instabilitäten nur sehr selten auftreten, werden Orthogonalisierungsmethoden untersucht, da diese eine hohe Stabilität gegenüber schwach konditionierten Matrizen aufweisen. Eine Einführung zu den hier verwendeten Verfahren gibt Abschnitt 3.4.2.

An dem grundsätzlichen Prinzip des Algorithmus ändert sich dabei nichts. Für den erfolgreichen Ablauf des Algorithmus werden ausschließlich andere mathematische Beziehungen herangezogen. Untersucht werden an dieser Stelle zwei Orthogonalisierungsmethoden. Zum einen ist das die in Abschnitt 3.4.2.1 beschriebene QR–Faktorisierung (*QRF*) und zum anderen die Singulärwertzerlegung (*SVD*), die in Abschnitt 3.4.2.2 hergeleitet wurde. Bei der QRF erfolgt die Anwendung der reduzierten QR–Faktorisierung. Für die Vorberechnung bedeutet das die Berechnung der Matrizen Q₁ und R₁ und für die Singulärwertzerlegung das Aufstellen der Matrizen U₁, S₁ und V^T. Durch die Anwendung der Orthogonalisierungsmethoden steigt zwar die Komplexität der Vorberechnung, jedoch werden die energielimitierten Sensorknoten nicht weiter belastet. In der nachfolgenden Tabelle 6.1 sind die Vorund Nachberechnungen für alle drei untersuchten Methoden zur Lösung der Lokalisierungsaufgabe gegenübergestellt.

 Tabelle 6.1: Übersicht der Vor- und Nachberechnung der benutzten Lösungsverfahren für die Methode der kleinsten Quadrate.

Methode	Vorberechnung	Nachberechnung		
Normalgleichungen	$\mathbf{A}_{\mathbf{p}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T, \mathbf{K} = \tilde{x}_i^2 + \tilde{y}_i^2$	$\mathbf{b} = \hat{r}_i^2 - \mathbf{K}, \mathbf{x} = \mathbf{A}_p \mathbf{b}$		
QR-Faktorisierung	$\mathbf{Q_1}, \mathbf{R_1}, \mathbf{K} = \tilde{x}_i^2 + \tilde{y}_i^2$	$\mathbf{b} = \hat{r}_i^2 - \mathbf{K}, \mathbf{R}_1 \mathbf{x} = \mathbf{Q}_1 \mathbf{b}$		
Singulärwertzerlegung	$\mathbf{U_1}, \mathbf{S_1}, \mathbf{V}^T, \mathbf{K} = \tilde{x}_i^2 + \tilde{y}_i^2$	$\mathbf{b} = \hat{r}_i^2 - \mathbf{K}, \mathbf{x} = \mathbf{V}\mathbf{y} \text{ mit } \mathbf{S_1}\mathbf{y} = \mathbf{U_1}^T \mathbf{b}$		

6.1.4 Algorithmischer Ablauf von RAL

Nach Herleitung des mathematischen Modells kann der Ansatz in einem Algorithmus, dem "Resource Aware Localization"–Algorithmus (*RAL*) zusammengefasst werden. Dieser läuft hauptsächlich in drei Phasen ab [BB10, BR10]. Dabei wird in unkritisch und kritisch ablaufenden Phasen unterschieden. Es wird von unkritisch gesprochen, wenn die Kommunikation ausschließlich zwischen den Beacons und der Senke stattfindet, da diese geringeren oder keinen Energielimitationen unterliegen. Kritisch hingegen verläuft die Kommunikation zwischen Sensorknoten und Beacons, da die Knoten ihren energiesparenden Schlafmodus verlassen müssen und Energie bei dem Empfang der Vorberechnungspakete und bei der Nachberechnung verbrauchen.

Phase 1 Initialisierung (unkritisch):

- Alle Beacons bestimmen ihre Position mittels eines geeigneten Lokalisierungssystems und
- senden ihre Position $\tilde{B}(\tilde{x}_i, \tilde{y}_i)$ zur Senke.

Phase 2 Vorberechnung (unkritisch):

– Nach dem Empfang aller Beaconpositionen berechnet die Senke Matrix A_p und Vektor K. Kommunikation (unkritisch):

- Die Senke sendet die vorberechnete Matrix Ap und Vektor K an die Beacons.

Phase 3 Nachberechnung (kritisch):

- Die Beacons senden die Vorberechnungspakete an die Sensorknoten.
- Die Sensorknoten verlassen den energiesparenden Schlafzustand und wechseln in den aktiven Modus und bestimmen ihre Entfernung zu jedem Beacon $\hat{r}_1...\hat{r}_n$ beim Erhalt der Vorberechnung A_p und K.

– Die Sensorknoten bauen Vektor **b** auf, berechnen ihre Position \bar{P}_{est} und wechseln in den energiesparenden Schlafmodus.

Der prinzipielle Ablauf des hier vorgestellten *RAL*–Algorithmus unterscheidet sich nicht gegenüber dem des *DLS*–Algorithmus. Wie bereits erwähnt, sind die Vorteile im mathematischen Ansatz begründet. Für umfassende Genauigkeitsuntersuchungen kann der Ablauf erweitert werden. Darauf wird detailliert in Abschnitt 6.4 eingegangen.

Nachfolgend soll der benötigte Berechnungs-, Speicher- und Kommunikationsaufwand betrachtet werden. In einem weiteren Schritt erfolgt ein Aufwandsvergleich mit den beiden Verfahren zur exakten Lokalisierung, den in Kapitel 4.3 beschriebenen "Fine-Grained-Localization"–Algorithmus (*FGL*) und dem *DLS*.

6.2 Ressourcenverbrauchsanalyse

6.2.1 Rechenaufwand

An dieser Stelle erfolgt eine Aufwandsabschätzung des neu eingeführten *RAL*–Algorithmus. Dabei ist die Analyse in zwei Bereiche unterteilt. Der erste Teil untersucht die Komplexitäten der im vorigen Abschnitt aufgeführten Berechnungsmethoden Normalgleichungen, QR–Faktorisierung und Singulärwertzerlegung. Im zweiten Teil folgen Überlegungen zur Kommunikationslast.

Zur Abschätzung der Komplexität der mathematischen Methoden wurde ein in der Literatur gängiges Verfahren gewählt. Dafür sind die für die Berechnung notwendigen Gleitkommaoperationen (engl. "Floating Point Operation", *Flops*) gezählt worden. Allerdings hängt die tatsächliche Anzahl *Flops* von der eingesetzten Hardwarearchitektur ab. Um dennoch ein aussagekräftiges Ergebnis zu erhalten, wurde jede Operation als eine *Flop* gezählt. Unabhängig davon ob es sich um eine Addition, Subtraktion, Multiplikation oder Division handelt. Es muss berücksichtigt werden, dass normalerweise eine Division oder Multiplikation komplexer ist als eine Addition oder Subtraktion. Darüber hinaus werden auch keine Kopieroperationen berücksichtigt, da deren Implementierung einen sehr starken Einfluss auf die Effizienz ausübt.

6.2.1.1 Gesamtkomplexität der Methoden

Komplexität der Normalgleichungen

Es gilt, das Gleichungssystem aus Abschnitt 6.1.1 zu lösen:

$$\mathbf{x} = \left(\mathbf{A}^T \mathbf{A}\right)^{-1} \mathbf{A}^T \left(\mathbf{r}^2 - \mathbf{K}\right),\tag{6.6}$$

was zu folgenden Teilkomplexitäten führt. Mit dem Lösungsvektor $\mathbf{x} = [\hat{x}_N, \hat{y}_N, \hat{w}]^T$ ergibt das für die Unbekannten u = 3 mit der Beaconanzahl n:

- 1. Multiplikation der $(3 \times n)$ -Matrix \mathbf{A}^T mit der $(n \times 3)$ -Matrix \mathbf{A} ergibt 8n 5 *Flops*. Einige Operationen können bei der Multiplikation einer transponierten Matrix mit sich selbst eingespart werden.
- 2. Die (3×3) -Matrix A wird invertiert, was zu einer $\mathcal{O}(u^3)$ -Komplexität führt [GL96].
- 3. Die (3×3) -Matrix aus 2. wird mit der $(3 \times n)$ -Matrix \mathbf{A}^T multipliziert, was 12n Flops beansprucht. Das führt zur vorberechneten Matrix $\mathbf{A}_{\mathbf{p}}$.
- 4. Matrix A_p wird mit dem *n*-Vektor **b** multipliziert. Dieser Schritt hat eine Komplexität von 4n 2 *Flops*.
- 5. Die Berechnung von b benötigt 5n Flops.

Die Schritte 1 bis 5 ergeben zusammen eine Gesamtkomplexität von 29n + 20 *Flops* für die Normalgleichungen.

Komplexität der QR–Faktorisierung

Auch hier muss Gleichung (6.6) gelöst werden, was durch eine QR-Faktorisierung erfolgt. Die Berechnung der Matrix Q beschränkt sich für u = 3 Unbekannte auf die Bestimmung von Q_1, Q_2 und Q_3 .

- 1. Berechnung von Q_1 benötigt $2n^2 + 5n + 1$ *Flops*.
- 2. Berechnung von Q_2 benötigt $2n^2 + n + 3$ *Flops*.
- 3. Berechnung von Q_3 benötigt $2n^2 3n 10$ Flops (die letzte Zeile wird nicht benötigt).
- 4. Multiplikation von $\mathbf{R} = \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \mathbf{A}$ inklusive der Berechnung aus 1 bis 3 benötigt $10n^2 8n 25$ *Flops*.
- 5. Berechnung von b benötigt 5n Flops.
- 6. Berechnung von $\mathbf{Q_1}^T$ benötigt $4n^3 20n^2 4n 23$ *Flops*
- 7. Berechnung von $\mathbf{Q_1}^T \mathbf{b}$ benötigt 6n 3 *Flops*.
- 8. Berechnung von $\mathbf{x} \approx \mathbf{Q}^T \mathbf{b}$ durch Rücksubstitution benötigt 6 *Flops*.

Damit ergibt sich eine Gesamtkomplexität für die Berechnung mittels QR–Faktorisierung von $4n^3 - 10n^2 - n - 45$ *Flops*.

Komplexität der Singulärwertzerlegung

Da für die Singulärwertzerlegung erst die Eigenwerte berechnet werden müssen, ist sie sehr komplex. Das kann durch etliche Methoden gelöst werden. Prinzipiell werden für die Berechnung von U_1 , S_1 und V für u = 3 Unbekannte $12n^2 + 32n + 243$ *Flops* benötigt [GL96]. Weiterhin wird x durch $S_1y = U^Tb$ und x = Vy berechnet, was zu einer Gesamtkomplexität von $12n^2 + 43n + 253$ *Flops* führt.

6.2.1.2 Reduzierte Komplexität

Nachdem die Gesamtkomplexitäten für alle Berechnungsmethoden bekannt sind, sollen diese den Komplexitäten der Nachberechnung gegenüber gestellt werden, um die Einsparmöglichkeiten zu verdeutlichen.

Normalgleichungen

Es verbleibt die bereits erwähnte Nachberechnung $\mathbf{x} = \mathbf{A}_{\mathbf{p}}\mathbf{b}$ und Vektor $\mathbf{b} = (\mathbf{r}^2 + \mathbf{K})$, was zu einer reduzierten Komplexität von 6n - 2 *Flops* führt. Wichtig zu erwähnen ist hierbei, dass bei der nach der Vorberechnung entstehenden $(3 \times n)$ - Matrix die letzte Zeile gestrichen wurde, da diese ausschließlich für die Berechnung der Hilfsvariablen notwendig ist. Es verbleibt bei der Nachberechnung nur die Multiplikation der reduzierten $(2 \times n)$ -Matrix mit dem Vektor b. Ergebnis ist der ebenfalls reduzierte Lösungsvektor $\mathbf{x} = [\hat{x}_N, \hat{y}_N]^T$. Dieses beschriebene Vorgehen ist mathematisch streng genommen nicht korrekt, allerdings steht an dieser Stelle das maximale Einsparpotential im Vordergrund. In Kapitel 3.6.6 wurde eine Möglichkeit beschrieben, Unbekannte bereits im Vorfeld aus der Berechnung zu streichen. Der Rechenaufwand dafür wird in Kapitel 7.2.3 gesondert diskutiert. Bei der Anwendung von *RAL*

ensteht ebenfalls das Problem, dass die Unbekannten über die Hilfsunbekannte miteinander korreliert sind. Wie diese Abhängigkeit aufgelöst werden kann, wird ebenfalls in Kapitel 7.2.3 besprochen.

QR–Faktorisierung

Im Fall der QR–Faktorisierung sendet die Senke die vorberechneten Matrizen Q_1 , R_1 und den Vektor K. Damit ergibt sich die Nachberechnung:

- 1. Aufstellen des Vektors b benötigt 2n Flops.
- 2. Multiplikation von $\mathbf{y} = \mathbf{Q_1}^T \mathbf{b}$, mit $\mathbf{Q_1}^T$ als $(3 \times n)$ -Matrix und \mathbf{b} als *n*-Vektor benötigt 6n 3 *Flops*.
- 3. Lösen von $\mathbf{R}\mathbf{x} \approx \mathbf{y}$ durch Rücksubstitution benötigt 6 *Flops*.

Zusammen ergibt das eine reduzierte Komplexität von 8n + 3 Flops für die QR–Faktorisierung.

Singulärwertzerlegung

Für die dritte Methode, die Singulärwertzerlegung, empfängt der Sensorknoten von der Senke die Matrizen U_1 , S_1 , V und K. Es verbleibt auf den Sensorknoten die Nachberechnung:

- 1. Aufstellen des Vektors b benötigt 2n Flops.
- 2. Lösen von $\mathbf{U_1}^T \mathbf{b}$, mit $\mathbf{U_1}^T$ als $(3 \times n)$ -Matrix und \mathbf{b} als *n*-Vektor, resultiert im Vektor \mathbf{z} und benötigt 6n 3 *Flops*.
- 3. y wird durch Rücksubstitution von $S_1y = z$ berechnet. Durch die drei Null-Elemente in S_1 werden nur 3 *Flops* benötigt.
- 4. Der letzte Teil der Berechnung benötigt 10 Flops, wobei x durch x = Vy ermittelt wird.

Alles zusammen ergibt sich eine reduzierte Komplexität von 8n + 10 *Flops*.

6.2.1.3 Die Komplexitäten im Vergleich

Ein Vergleich aller Komplexitäten ist in Tabelle 6.2 und grafisch in Abbildung 6.1 (a) wiedergegeben. Um aussagekräftige Werte zu erhalten, wurde beispielhaft mit n = 100 Beacons gerechnet. Die Gesamt-

Tabelle 6.2:	Vergleich	der Gesamt- ι	und reduzi	erten Kon	plexitäten	für die	Methode	der kle	einsten (Quadrate	ı für
n = 100 E	Beacons.										

Methode	Gesamtberechnung	Reduzierte	Einsparung
	Komplexität [Flops]	[Flops]	[%]
Normalgleichungen	2.920	598	79,52
QR-Faktorisierung	3.899.859	803	99,98
Singulärwertzerlegung	124.553	810	99,35

berechnung der Methode der kleinsten Quadrate erfordert viel mehr Operationen als die reduzierten Berechnungen. Speziell die QR-Faktorisierung würde die Ressourcen der Sensorknoten extrem belasten. Wie in den Betrachtungen gezeigt wurde, können die reduzierten Methoden die Komplexitäten der Berechnungen signifikant verringern und diese Berechnungen auf ressourcenlimitierten Sensorknoten ermöglichen. Es ist aber weiterhin ein gewisser Anteil an Kommunikation erforderlich, was im nächsten Abschnitt genauer analysiert werden soll.

6.2.2 Kommunikationsaufwand

In Kapitel 2 wurde beschrieben, dass die Kommunikation im Sensornetzwerk minimiert werden muss, da sie für den Großteil des Energieverbrauchs verantwortlich ist. Vor allem verbraucht das Versenden über große Distanzen viel Energie. Dabei ist die Kommunikation zwischen der Senke und den Beacons vorzuziehen, da deren Ressourcen keinen oder wenigen Limitationen unterliegen. Aus diesem Grund wurde bei der Algorithmusbeschreibung die Kommunikation in zwei Phasen unterteilt. In der kritischen Phase müssen die Sensorknoten theoretisch die Pakete nur empfangen. Praktisch treten jedoch bei jeder Transmission Paketverluste aufgrund von Unterbrechungen im Kommunikationskanal auf. Es kann auch erforderlich werden, auf Protokollebene den erfolgreichen Empfang eines Paketes zu signalisieren. In dieser Untersuchung liegt der Fokus ausschließlich auf theoretischen Betrachtungen des Algorithmus. Das heißt, sie erfolgt unabhängig von Protokolldefinitionen und Medienzugriffsprotokollen (vgl. Kapitel 2.2). Jeder Sensorknoten muss die vorberechnete Matrix A_p und den Vektor K mit $[3n + n = 4 \cdot n]$



Abbildung 6.1: (a) Komplexitätsvergleich und (b) Kommunikationsaufwand der Gesamt- und reduzierten Berechnung über eine steigende Beaconanzahl.

Elementen empfangen. Bei 100 Beacons beläuft sich der Aufwand auf 1.600 *Bytes*, wenn jedes Element als 32 *bit* Gleitkommazahl gerechnet wird. Auf herkömmlichen Mikrocontrollereinheiten, die für den Einsatz auf Sensorknoten eingesetzt werden, wird jedes Element als Gleitkommaelement als 4 *Byte*– Nummer abgespeichert. Für die Normalgleichungen wird die Vorberechnung vereinfacht, indem die dritte Zeile aus der vorberechneten Matrix A_p weggelassen wird (vgl. Abschnitt 6.2.1). Das heißt für die Kommunikation, es müssen *n* Elemente weniger verschickt werden. Somit reduziert sich der Kommunikationsaufwand auf $3 \cdot n$ Elemente. Im Beispiel der 100 Beacons ergeben sich damit 1.200 Elemente.

Für die Gesamtberechnung würden hingegen $2 \cdot n$ Elemente an der Kommunikation teilnehmen. Das wären im einzelnen die Koordinaten der Beacons $\tilde{B}_i(\tilde{x}_i, \tilde{y}_i)$, was in diesem Beispiel zu einem Kommunikationsaufwand von 800 Elementen führen würde. Das bedeutet, dass die Gesamtberechnung den Kommunikationsaufwand um 33 % reduzieren würde. Jedoch wird im nächsten Abschnitt gezeigt, dass aufgrund der Einsparungen in der Berechnung dieser Mehraufwand nicht weiter ins Gewicht fällt.

Abbildung 6.1 (b) gibt den Kommunikationsaufwand für alle drei Lösungsmethoden wieder. Die Grafik bildet den Verbrauch bis zu einer Anzahl von 50 Beacons ab. Der Kommunikationsaufwand steigt mit wachsender Beaconanzahl linear an. Tabelle 6.3 zeigt die einzelnen Elemente für die Lösungsmethoden, die an der Kommunikation teilnehmen.

Tabelle 6.3: Kommunikationsaufwand ohne Protokolloverhead der benutzten Lösungsverfahren für die reduzierteBerechnung der Methode der kleinsten Quadrate.

Methode	Elemente	Dimensionen	Anzahl	Byte
			Elemente	
Normalgleichungen	A _p , K	$[2 \cdot n + n]$	300	1.200
QR-Faktorisierung	Q ₁ , R ₁ , K	$[3 \cdot n + 3 \cdot n + n]$	406	1.624
Singulärwertzerlegung	$\mathbf{U_1}, \mathbf{S_1}, \mathbf{V}^T, \mathbf{K}$	$[3 \cdot n + 3 + 9 + n]$	412	1.648

6.3 Simulation und Ergebnisse

Bevor der neu eingeführte *RAL*–Algorithmus in Kapitel 6.3.3 anderen etablierten Lokalisierungsverfahren gegenübergestellt wird, erfolgt in diesem Abschnitt die Präsentation einiger Simulationsergebnisse.

6.3.1 Simulationsaufbau

Die Simulationen erfolgten im Paketsimulator J-Sim. In [Rei07] beschreibt Reichenbach ein erweitertes Energiemodell für diesen Simulator, welcher für die Simulation von Lokalisierungsalgorithmen um fehlende Distanzabhängigkeiten erweitert wurde. Damit können Sendereichweiten entsprechenden Sendeleistungen zugeordnet werden. Für diese Simulationen wurde kein spezielles Routingprotokoll zur Paketweiterleitung verwendet. Ausgehend von den unterschiedlichen Phasen von *RAL* wurde für jedes

Parameter	Wert	Parameter	Wert
Sensorfeldgröße [m]	100×100	Sendereichweite Beacons [m]	60
Sensorknotenanzahl	300	Varianz der Distanz [%]	10
Beaconanzahl	15	Simulationszeit [s]	130

Tabelle 6.4: Parameter für die Simulation von RAL.

Positionspaket, welches die Vorberechnung der Senke enthält, in Phase 1 pro Beacon genau eines weitergeleitet. Danach folgt ein Wechsel in den energiesparenden Sleep–Modus (vgl. Kapitel 2.2.2.1) bis zum Empfang der Vorberechnung aus Phase 2. Nun erfolgt ein Wechsel in den Kommunikationsmodus, wobei genau ein Vorberechnungspaket weitergeleitet wird, worauf hin wieder der Wechsel in den Schlafmodus erfolgt. Die Sensorknoten wechseln in Phase 3 in den Kommunikationsmodus und empfangen genau ein Vorberechnungspaket. Danach berechnen sie ihre Position und wechseln wiederum in den energiesparenden Sleep–Modus.

In den Simulationen wurde ein Sensorfeld der Größe $(100 \, m \times 100 \, m)$ verwendet. Die Anzahl der Sensorknoten betrug 300 mit 5% Beacons. Die genauen Simulationsparameter sind in Tabelle 6.4 wiedergegeben.

6.3.2 Untersuchungen zum Energieverbrauch

In Abbildung 6.2 (a) sind die Energieverbräuche dargestellt, die bei der Ausführung von *RAL* anfallen. Wie zu ersehen ist, startet *RAL* nach 69,3s mit der Positionsbestimmung der Beacons. Dieser Prozess dauert 4,6s. Die Phase 2, das Versenden der Positionspakete startet nach 100,1s für 9,3s.

Da die Vorberechnung auf der Senke abläuft, fällt dieser Energieverbrauch nicht ins Gewicht. Phase 3 schließt sich unmittelbar an, die Vorberechnung wird über die Beacons an die Sensorknoten versendet. 113 s nach Simulationsbeginn haben die Sensorknoten ihre Position bestimmt und wechseln danach in den Sleep–Modus. Dabei wurden 240 Pakete gesendet und 2.320 empfangen, davon 13 % auf den Sensorknoten.

In Abbildung 6.2 (b) ist der Energieverbrauch auf jedem Knoten dargestellt. Die Verbräuche liegen sehr nah beieinander. Das ist dadurch zu erklären, dass jeder Knoten das selbe Positionspaket erhält, unabhängig von seiner Lage im Sensorfeld. Der energiesparende Sleep–Modus wird nur für eine sehr kurze Zeit unterbrochen, in der die Sensorknoten die Vorberechnung empfangen und die Nachberechnung ausführen. Pro Sensorknoten liegt der Verbrauch bei gerade einmal 0,01 *J*.



Abbildung 6.2: Energieverbräuche (a) des gesamten Sensornetzes und (b) der einzelnen Sensorknoten bei Anwendung von RAL.

6.3.3 Vergleich mit etablierten Verfahren

Um die Performance des *RAL*–Algorithmus einschätzen zu können, soll er hinsichtlich seiner Komplexität, seines Energieverbrauchs und Kommunikationsaufkommens den Algorithmen *FGL* und *DLS* (vgl. Kapitel 4.3) vergleichend gegenüber gestellt werden. Für sämtliche Simulationen wurde die Positionsbestimmung mittels Normalgleichungen vorgenommen.

Rechenaufwand

Die Bestimmung der Anzahl der Rechenoperationen wurde wie in Tabelle 6.2 beispielhaft für 100 Beacons durchgeführt. Die Ergebnisse gibt Tabelle 6.5 wieder. Da für den *FGL*–Algorithmus auf

Tabelle 6.5: Vergleich der Komplexitäten für die Untersuchten Lokalisierungsverfahren für n = 100 Beacons.

Methode	FGL	DLS	RAL	Einsparung	Einsparung
	[Flops]	[Flops]	[Flops]	RAL–FGL [%]	RAL-DLS [%]
Normalgleichungen	3.050	791	598	80,40	24,40
QR-Faktorisierung	3.951.298	1.191	803	99,98	32,58
Singulärwertzerlegung	82.556	803	810	99,01	-0,87

jedem einzelnen Knoten die Gesamtberechnung durchgeführt wird, erfordert er auch die meisten Rechenschritte. Besonders fällt das bei den Orthogonalisierungsmethoden auf. *RAL* spart hier mehr als 99 %. Auch bei den Normalgleichungen ist ein großes Einsparpotenzial erkennbar. Fast 80 % der Berechnungen können auf die Senke ausgelagert werden.

Im Gegensatz dazu fallen die Einsparmöglichkeiten im Vergleich zu *DLS* geringer aus. Dennoch ergeben sich für die Normalgleichungen und die QR–Faktorisierung fast 25 % und 33 % weniger Rechenschritte. Diese Einsparungen lassen sich dadurch erklären, dass der b–Vektor bei *RAL* einfacher zu berechnen ist. Es werden pro Element nur eine Multiplikation und eine Addition nötig. Bei *DLS* hingegen kommen nochmals zwei Rechenschritte pro Element hinzu. Allerdings benötigt *DLS* bei der SVD weniger Rechenschritte. Der Mehraufwand bei *RAL* von 0,87 % ist jedoch vertretbar.

Energieverbrauch

Für die Vergleichssimulationen diente die Simulationskonfiguration aus Tabelle 6.4. Die Abbildung 6.3 (a) zeigt die Energieverbräuche des ganzen Sensornetzes, Abbildung 6.3 (b) hingegen der einzelnen Sensorknoten im Vergleich zu *FGL* und Abbildung 6.4 (a) und 6.4 (b) entsprechend zu *DLS*. Die



Abbildung 6.3: Energieverbrauch (a) des gesamten Sensornetzes und (b) der einzelnen Sensoren bei Anwendung von RAL und FGL.

Gegenüberstellung von *RAL* und *FGL* zeigt die Vorteile eines verteilten Algorithmus. Die Berechnungen für den "Fine Grained Localization"–Algorithmus müssen auf jedem Sensorknoten ausgeführt werden. Das heißt, die Berechnungen der Normalgleichungen erfolgen hoch redundant, was zu einem erhöhten Energieverbrauch führt. Die Sensorknoten müssen sehr viel länger aktiv bleiben und auch mehr Pakete empfangen. In diesen Simulationen erhielten die Sensorknoten die ersten Positionspakete nach 70, 2 s. Bei Zeitpunkt 105, 7 s hatte jeder Knoten ein Paket von den Beacons empfangen. Alles in allem wurden 236 Positionspakete versendet und 53.527 empfangen, von denen 93 % auf den Sensorknoten anfielen. Dabei waren die Sensorknoten 35, 5 s im aktiven Zustand.

Für den einzelnen Sensorknoten bedeutet das im Detail, dass der Sleep–Modus für lange Zeit verlassen wird. Die großen Sprünge in den Kurven der Abbildung 6.3 (b) sind darauf zurückzuführen, dass jeder Sensorknoten unterschiedliche Positionspakete erhält. Die Knoten in der Mitte des Sensorfeldes lokalisieren sich häufiger als die Knoten an den Rändern.

Wie im vorigen Abschnitt 6.3.2 gezeigt und in Abbildung 6.3 ersichtlich, liegt *RAL* deutlich unter dem Energiebedarf von *FGL*. Im Einzelnen macht das eine Energieersparnis von 89 % aus. Der "Distributed



Abbildung 6.4: Energieverbrauch (a) des gesamten Sensornetzes und (b) der einzelnen Sensoren bei Anwendung von RAL und DLS.

Least Squares"–Algorithmus durchläuft die Simulationsphasen in der selben Zeit wie der *RAL*–Algorithmus. Ebenso viele Pakete werden während der Simulation ausgetauscht. Der wesentliche Unterschied liegt in der Nachberechnung. Dadurch, dass *DLS* eine aufwendigere Nachberechnung durchzuführen hat, wird auch mehr Energie verbraucht. In Abbildung 6.4 ist der energetische Mehraufwand zu erkennen. In den Simulationen verbrauchte *RAL* 11,8% weniger Energie als *DLS*.

Für beide Algorithmen liegen die Verbräuche der einzelnen Sensorknoten ebenfalls dicht beieinander. Das ist darin begründet, dass die Vorberechnung und das Versenden der Positionspakete bei beiden Verfahren gleich abläuft. In beiden Fällen müssen die Sensorknoten den Sleep–Modus für nur sehr kurze Zeit verlassen, um die Pakete zu empfangen und die Nachberechnung durchzuführen.
6.4 Clusterisierung und nachträgliche Netzausgleichung – Anpassung für große Netzwerke

6.4.1 Clusterisierung des Sensorfeldes

In den bisherigen Genauigkeitsbetrachtungen wurde von einer Einzelpunkteinschaltung ausgegangen, d. h., es wurde jeweils nur ein Sensorknoten mit den aufgebauten Gleichungssystemen berechnet. Jede zusätzliche Beobachtung führte zu einer zusätzlichen Gleichung im Gleichungssystem. Bei einer Gesamtnetzausgleichung fließen jedoch alle Unbekannten und alle Beobachtungen in die Berechnungen ein, was zu sehr großen Gleichungssystemen führt. Bei z. B. 500 Sensorknoten im Netz ergibt jeder zusätzliche Beacon 500 weitere Beobachtungsgleichungen, wenn sich der Beacon in Empfangsreichweite eines jeden Sensorknotens befindet. Des Weiteren hätte die sich ergebende Normalgleichungsmatrix als symmetrische Matrix 1.000.000 verschiedene Elemente, wenn nur die halbe Matrix abgespeichert wird. Das iterative Invertieren solcher Matrizen ist mit einem sehr großen Aufwand verbunden und benötigt somit viel Rechenzeit.

Da die Beacons nicht über eine unendliche Sendereichweite verfügen, ist es notwendig, dass mehrere Vorberechnungen für einzelne sogenannte Netzwerkcluster aufgestellt werden. Die Wahl der Cluster ist dabei so vorzunehmen, dass mit wenigen das gesamte Sensornetzwerk möglichst gut abgedeckt wird, um den Kommunikations- und Rechenaufwand gering zu halten. Dabei erhält jeder Cluster eine eigene Vorberechnung, die auf einem zentralen Beacon, der den Mittelpunkt des Clusters definiert, durchgeführt wird. Darauf folgend bestimmen die Sensorknoten in dessen Reichweite ihre Position. Abbildung 6.5 verdeutlicht den Sachverhalt. Die Cluster überlappen sich in einem gewissen Bereich. Berechnen dort positionierte Sensorknoten mehrere Positionen, kann diese Mehrfachbestimmung ausgenutzt werden, die Cluster untereinander auszugleichen. Der dabei entstehende Mehraufwand ist ebenfalls zu analysieren und es ist abzuwägen, ob sich dieser aus energetischer Sicht lohnt oder ob darauf verzichtet werden sollte.

Als Vorteile der Clusterisierung wären zu nennen:

- Durch Cluster sinkt der Rechenaufwand, da weniger Beacons in die Multilateration eingehen.
- Die Robustheit steigt, da Beaconausfälle nur Teile des Sensornetzes betreffen, nicht aber das ganze Netzwerk ausfällt.
- Es entsteht eine höhere Realitätstreue, da für einige Szenarien mit großen oder ausgedehnten Sensornetzwerken, d. h. größer als die Sendereichweite der Beacons, gerechnet werden muss.

Demgegenüber stehen folgende Nachteile:

- Durch die geringere Anzahl von Beacons für die Multilateration sinkt die Genauigkeit der Positionsbestimmung.
- Es wird zusätzlicher Aufwand für die Clusterbildung nötig.
- Für die abschließende Nachausgleichung der Cluster ist ein erhöhter Kommunikationsaufwand nötig, da alle Positionsdaten und Distanzmessungen zur Senke gesendet werden müssen.

Im nachfolgenden Teil wird eine Strategie vorgestellt, die einerseits die dynamische Einteilung eines großen Sensornetzwerkes in kleinere Cluster beschreibt und darüber hinaus eine nachträgliche Netzausgleichung auf Basis der Methode der kleinsten Quadrate benutzt (vgl. Kapitel 3.6), um eine



Abbildung 6.5: Clusterisierung eines Sensornetzwerkes.

Genauigkeitssteigerung für die Sensorknoten zu erzielen. Da diese Strategie in einer zusätzlichen Kommunikationsphase resultiert, wird der Mehraufwand erneut berechnet und abgewogen, ob und ab wann dieses Vorgehen aus energetischer Sicht sinnvoll ist.

Für eine Clusterisierung eines Sensornetzwerkes sind mehrere Möglichkeiten denkbar und in der Literatur diskutiert [HM09a, LGF10, CHWJ10]. In den folgenden Untersuchungen werden zwei verschiedene Ansätze betrachtet, die sich in ihren Anwendungsfällen unterscheiden:

- Berechnungsaufwand: In sehr großen Sensornetzwerken mit einer großen Anzahl Sensorknoten steigt der Berechnungsaufwand einer nachträglichen Netzausgleichung (Ausgleichung in einem Guss) exponentiell an. Das Sensorfeld ist so zu clustern, dass eine bestimmte Anzahl an Sensorknoten (vgl. 2.2.1) pro Cluster nicht überstiegen wird.
- **2. Sendereichweite:** In einem zweiten Ansatz wird die Clusterisierung auf Basis der Sendereichweite (Abbildung 6.5) der Beacons durchgeführt.

Abbildung 6.7 (a) verdeutlicht beide Formen der Clusterisierung. In den nachfolgenden Untersuchungen werden Clusterisierungen analysiert, die die angeführten Faktoren *Berechnungszeit* und *Sendereichweite* zur Grundlage haben.

Berechnungsaufwand

Der nötige Berechnungsaufwand und damit die Berechnungszeit für eine nachträgliche Netzausgleichung nach Kapitel 3.6 soll an einem Beispiel illustriert werden. Es wird von einem Sensornetzwerk der Größe $(100 \ m \times 100 \ m)$ ausgegangen, in dem n = 500 Sensorknoten und b = 25 Beacons verteilt sind. Nach Gleichung 2.1 bedeutet das eine durchschnittliche Knotenverteilungsdichte von 0,05 Knoten pro Quadratmeter. Auf den Sensorknoten werden die Distanzen zu den Beacons bestimmt und der *RAL*-Lokalisierung zugeführt. Die berechneten Positionen werden im nächsten Schritt über die Beacons an die Senke übermittelt. Zeitgleich messen die Sensorknoten die Distanzen zu den in Reichweite befindlichen Sensorknoten und senden diese zu den Beacons, welche ebenfalls die Distanzen zu den Knoten, in deren Reichweite sie sich befinden, bestimmen. Wie aus Abbildung 6.6 zu entnehmen ist, ergeben sich für die Berechnungen variierende mögliche Beobachtungen je nach Wahl der Distanzmessungen. Die Anzahl der möglichen Beobachtungen beeinflusst maßgeblich die Berechnungsgeschwindigkeit, mit



Abbildung 6.6: Anzahl maximal möglicher Beobachtungen im Sensornetzwerk bei b = 25 Beacons über einer steigenden Anzahl Sensorknoten.

der das Sensornetz ausgeglichen werden kann. Werden alle Strecken mit in die Berechnung einbezogen (Beacon–Sensorknoten, Sensorknoten–Sensorknoten, Sensorknoten–Beacon und Beacon–Beacon) ergeben sich n(n-1) + 2nb + b(b-1) = 275.100 mögliche Beobachtungen (rote Linie). Werden die Distanzmessungen zwischen den Beacons ausgeschlossen, fließen noch immer 262.000 Beobachtungen in die Berechnung ein (blaue Linie). Sollen ausschließlich die Einfachmessungen Sensorknoten–Beacon benutzt werden, ergibt sich die geringste Zahl möglicher Beobachtungen. Den interessantesten Fall ergibt die Berechnung mittels Mehrfachbeobachtungen. Doppelbeobachtungen werden dabei gemittelt, was die Anzahl Beobachtungen halbiert. Die Überbestimmung bleibt hierbei erhalten, sodass kein Genauigkeitsverlust auftritt [BW03].

Jedoch ist der Aufwand relativ groß, der eine Berechnung von 262.000 Beobachtungen bei 1.000 Unbekannten mit sich bringt. Um diesen hohen Aufwand zu reduzieren soll das Sensornetzwerk nach



Abbildung 6.7 (a) aufgeteilt werden. Bei einer maximalen Anzahl von 30 Sensorknoten pro Cluster und der ermittelten durchschnittlichen Verteilungsdichte ergibt sich eine Rastergröße von 24,5m bei 16 Cluster. In diesem Beispiel ist zu erkennen, dass gerade an den Rändern nur kleine Bereiche neuen

Abbildung 6.7: Clusterisierung eines Sensornetzwerkes mit Punkten als Sensorknoten und Dreiecken als Beacons (a) durch feste Clusterisierung abhängig von der Sensorknotenanzahl und (b) mittels eines Clusterheads (grünes Dreieck) abhängig von der Sendereichweite.

Clustern zugeordnet sind. Da die Abbildung lediglich das Prinzip darstellt, sind die Restcluster in der Grafik ebenfalls eingezeichnet. Bei der Berechnung wird das Raster derart über das Sensorfeld verschoben, dass eine möglichst gute Abdeckung des Gebietes erreicht wird, dabei jedoch die Sensorknotenanzahl im Cluster nicht einen zu hohen Wert annimmt. Ist das Raster endgültig positioniert, werden die Restcluster an die nächstgrößeren angeschlossen und mitberechnet. Da die Berechnung eines jeden Clusters getrennt erfolgt, treten in den Clustern untereinander Inkonsistenzen auf. Aus diesem Grund werden zusätzlich Überlappungsbereiche zwischen den Clustern definiert (gestrichelte Linien in Abbildung 6.7 (a)). Knoten, die sich in diesen Überlappungsbereichen befinden, berechnen sich den jeweiligen Clustern zugehörige Positionen. Mit diesen Verknüpfungspunkten zwischen den Gebieten ist es dann möglich, über eine Gesamtausgleichung die Cluster anzupassen. Im nächsten Abschnitt wird dazu eine andere Möglichkeit verwendet, die es erlaubt, mit geringem Aufwand große Blöcke auszugleichen.

Nach Golub et al. [GR71, GL96] besitzt die Methode der kleinsten Quadrate eine Rechenkomplexität von $O(n^3)$. Abbildung 6.8 (a) stellt den Rechenaufwand über die steigende Knotenanzahl dar. Für das gewählte Szenario ergibt sich für m = n + b = 525 ein Aufwand von $16m^3 + 6m^2 + 2m = 2, 3$ *GFlops* für die Gesamtberechnung. Wird das Sensornetz geclustert, ergibt sich ein Berechnungsaufwand von ca. 8, 2 *MFlops* pro Cluster für die Ausgleichung. Das Sensorfeld ist durch u = 16 Cluster abgedeckt, was wiederum zu einem Rechenaufwand von 131, 7 *MFlops* führt. Verglichen mit der Gesamtausgleichung des Sensorfeldes "in einem Rutsch" ergibt sich für den Berechnungsaufwand ein Einsparpotenzial von 94, 3 %. Für die zeitliche Komplexität ergibt sich ein ähnliches Bild (Abbildung 6.8 (b)). Der Vorteil der Cluster wird auch hier ersichtlich. Da mit steigender Anzahl Sensorknoten auch die Anzahl der



Abbildung 6.8: (a) Berechnungsaufwand über steigender Sensorknotenzahl und (b) zeitliche Komplexität für die Berechnung bei b = 25 Beacons.

Beobachtungen und damit die Komplexität des Ausgleichungsproblems exponentiell ansteigt, steigt ebenfalls die Berechnungsdauer exponentiell. In der clusterbasierten Berechnung addiert sich die zeitliche Komplexität, wobei das Erstellen der für die Berechnung notwendigen Matrizen und Beobachtungsvektoren den weitaus größten Teil der Zeit in Anspruch nimmt. In diesem Fall wurden 130 Sensorknoten pro Cluster zugrunde gelegt, was eine Zeiteinsparung von ca. 75 % gegenüber der Gesamtausgleichung ergibt. Es sei erwähnt, dass es in der Literatur zahlreiche Ansätze zur Lösung schwach besetzter Matrizen gibt. Zum Beispiel untersucht Gründig die konjugierte Gradientenmethode im Hinblick auf Rechenzeiteinsparungen für die Ausgleichung großer Netze [Grü80]. Eine Übersicht über Berechnungsverfahren großer schwach besetzter Matrizen ist in George et al. zu finden [GGL93].

Sendereichweite

Erreicht das Sensornetzwerk eine kritische Größe, in der die Sensorknoten nur noch einige und nicht alle Beacons erreichen, muss jeder Cluster eine eigene Vorberechnung nach Kapitel 6.1.4 erhalten. Dafür müssen Clusterbeacons gefunden werden, mittels derer das Sensorfeld möglichst gut und lückenlos abgedeckt wird. Als zweites Kriterium soll die Minimierung der Clusteranzahl dienen, da mit weniger Cluster die Redundanz der Berechnung und damit der Berechnungsaufwand sinkt. Jeder Sensorknoten bestimmt in einem zweiten Schritt die Distanzen zu allen möglichen Beacons und speichert diese ab. Der Sensorknoten ordnet sich dem Cluster zu, dessen Clusterhead dem Knoten am nächsten liegt. Empfängt der Sensorknoten zwei oder mehr Clusterbeacons, d. h., liegt er in einem Überlappungsbereich, werden Positionen in beiden Clustern für den Sensorknoten bestimmt. Das erhöht die Stabilität und Robustheit des Sensornetzes, da eine mögliche Ausgleichung über diese Verknüpfungspunkte zwischen den Clustern berechnet werden kann. Zum Nachteil hat diese Strategie jedoch, dass der Rechenaufwand auf den Sensorknoten ansteigt, was mit den vorgestellten Algorithmen gerade vermieden werden sollte. Aus diesem Grund erfolgt im weiteren Verlauf eine Aufwandsabschätzung. Befindet sich ein Sensorknoten außerhalb der Reichweite einiger Beacons seines Clusters, müssen diese aus der Vorberechnung entfernt werden. Dies kann durch QR-Faktorisierung mit Matrixdown- und upgrading oder der rekursiven Methode der kleinsten Quadrate erreicht werden [Rei07]. Da dies jedoch rechnerisch aufwendig ist, wird an dieser Stelle darauf verzichtet. Es existieren ebenfalls Strategien, die Distanzen zu den Beacons außerhalb der Sendereichweite bestimmen [BSLT09, BST10b]. In diesen Untersuchungen wird allerdings auf die Wahl eines komplexeren Verfahrens verzichtet. Die fehlende Distanz wird durch Summation der Distanz Sensorknoten–Clusterhead–Beacon approximiert:

$$\hat{d}_2 = \hat{d}_1 + \tilde{d}_{P1P2} \tag{6.7}$$

mit \hat{d}_2 der Distanz zwischen Sensorknoten und dem Beacon außerhalb der Reichweite, \hat{d}_1 der Distanz zwischen Sensorknoten und Clusterhead und \tilde{d}_{P1P2} der Distanz zwischen dem Clusterhead und dem Beacon außerhalb der Sendereichweite. Den Einfluss zeigt Abbildung 6.9. Im ersten Fall wird die



Abbildung 6.9: Approximierte Distanz bei Beacons außer Empfangsreichweite der Sensorknoten (a) gute Näherung und (b) schlechte Näherung.

Distanz Sensorknoten–Beacon außer Reichweite sehr gut approximiert. Jedoch ergibt sich für einen Beacon gerade außerhalb der Empfangsreichweite des Sensorknotens die nachteiligste Näherung für die Streckenapproximation. Die Gesamtstrecke addiert sich zum Doppelten der Sendereichweite der Beacons, wohingegen die Länge der wahren Strecke gerade etwas oberhalb der Sendereichweite der Beacons liegt. Das würde für diese Distanz eine Abweichung von 100 % zur Originalstrecke bedeuten. In der Positionsberechnung würde das zu einem Ausreißer führen. Da im Vorfeld keine Aussage über die Geometrie innerhalb des Clusters getroffen werden kann, ist ebenfalls die Abweichung der Strecke unbekannt. Trotzdem hat sie einen Anteil an der Lokalisierung. Darüber hinaus kann es in einigen Szenarien dazu kommen, dass einige Sensorknoten weniger als die für die Lokalisierung nötigen drei Beacons außerhalb der Empfangsreichweite durch aufwendiges Matrixdowngrading auf jedem betreffenden Sensorknoten aus der Vorberechnung eliminiert werden. Aus diesem Grund fließt dieser Beacon mit in die Berechnung mit ein, wird in den nachfolgenden Untersuchungen jedoch mit einem kleineren Gewicht belegt als direkte Streckenbeobachtungen. Dadurch wird der direkte Einfluss dieses Beacons auf die Positionsschätzung des Sensorknotens minimiert.

6.4.2 Nachträgliche Netzausgleichung

In diesem Abschnitt werden die durch *RAL* ermittelten Sensorpositionen einer nachträglichen Netzausgleichung zugeführt. Da diese rechentechnisch sehr aufwendig ist, wird sie auf die Senke ausgelagert. Für den einzelnen Sensorknoten bedeutet dies einen zusätzlichen Kommunikationsaufwand. Es sind folgende Elemente zu übermitteln:

- Kennung des Sensorknotens.
- Die berechnete Knotenposition durch *RAL* (streng genommen werden die Positionen im weiteren Betrieb des Sensornetzwerkes nach der Lokalisierung ohnehin übermittelt. An dieser Stelle werden sie jedoch als zusätzlicher Kommunikationsaufwand eingeführt).
- Die gemessenen und approximierten Strecken zu den Beacons.
- Gemessene Strecken zu den Sensorknoten in Nachbarschaft.
- Da sich die Sendereichweiten der Beacons und der Sensorknoten unterscheiden d. h., die Beacons liegen in Sendereichweite nur weniger Sensorknoten – können nicht alle Werte direkt übermittelt werden, sondern müssen von Sensorknoten zu Sensorknoten zu den Beacons übermittelt werden.

Besonders der letzte Punkt hat zur Folge, dass der Kommunikationsaufwand mit abnehmender Sendereichweite der Sensorknoten ansteigt. Durch ein geeignetes Routingprotokoll kann dieser Effekt allerdings zielgerichtet werden, d. h., es ist kein Fluten aller Messungen und Positionen durch das Netzwerk nötig. In den Simulationen wurde ein "Greedy–Forwarding"–Algorithmus integriert, der die Informationen zielgerichtet durch das Netzwerk leitet. Dabei wird mit jedem Schritt versucht, die Nachricht dichter zu den Beacon hin zu verschicken. Mit den Positionen der Beacons wird die Nachricht an den Sensorknoten weitergeleitet, der aus lokaler Sicht dem Beacon am nächsten ist und somit die Distanz zu ihm verringert. Das hat den Vorteil, dass keine genaue Kenntnis aller Sensorknotenpositionen nötig ist, da ausschließlich aus lokaler Sicht entschieden wird. Als nachteilig hat sich herausgestellt, dass dieses Vorgehen in einer Sackgasse enden kann, d. h., es kann kein Sensorknoten gefunden, der die Distanz zum Beacon verringert. Das heißt weiterhin, dass alle Beacons in dieser Richtung außerhalb der Sendereichweite der Knoten liegen. In diesem Fall wird das Paket in den Simulationen über andere Sensorknoten zum zweitnächsten Beacon weitergeleitet. Aufgrund der hohen Knotendichte in den Simulationen war ein zweites Routing erfolgreich, sodass alle nötigen Informationen für die abschließende Ausgleichung vorlagen.

Da eine Clusterisierung vorangegangen ist, sollen zwei Möglichkeiten miteinander verglichen werden. Zum einen wird für jeden Cluster eine Einzelausgleichung gerechnet. Dabei wird jeder Cluster als eigenes Netzwerk behandelt, d. h. es kommen nur Beobachtungen zu Sensorknoten oder Beacons zum Einsatz, die sich innerhalb des Clusters befinden. Dadurch wird der Berechnungsaufwand heruntergesetzt, allerdings gehen auch Beobachtungen verloren, die das Lokalisierungsergebnis maßgeblich beeinflussen, was gerade in den Randbereichen große Auswirkungen haben kann. Aus diesem Grund soll auch eine andere Herangehensweise beschrieben werden, die alle Elemente des Sensornetzwerkes "in einem Guss" ausgleicht. Da dieses Vorgehen bei großen Sensornetzwerken einen großen Rechenaufwand mit sich bringt, werden die Cluster genutzt, um die Gesamtausgleichung aufzuteilen. Die endgültige Berechnung erfolgt durch eine Blockzerlegung nach Helmert [Höp80, Nie08]. Dabei wird das große Sensornetz in die vorhandenen Cluster zerlegt und mit Reduktion der Normalgleichun-

Parameter	Wert	Parameter	Wert
Sensorfeldgröße [m]	100×100	Sendereichweite Beacons [m]	35
Anzahl Sensorknoten	500/100*	Sendereichweite Sensorknoten [m]	10
Anzahl Beacons	5 % = 25	Anzahl Cluster	4
Varianz der Strecken [%]	25	Varianz der Beacons [%]	5
Gewichtseinheit vor der	5	Wiederholungen	10.000
Ausgleichung $\sigma_{0_{apriori}}$ [m]	5		10.000

Tabelle 6.6: Simulationsaufbau für die Nachausgleichung eines großen Sensornetzwerkes.

gen auf die Verknüpfungspunkte lösbar gemacht. Die Verknüpfungspunkte ergeben sich dabei durch die Überlappung der einzelnen Cluster. Das Ergebnis der Ausgleichung ist dasselbe als wäre das Gesamtnetz "in einem Guss" ausgeglichen worden.

6.4.3 Simulationen

Für die Simulation wird eine Konfiguration nach Tabelle 6.6 gewählt. Um eine gewisse Anschaulichkeit zu erreichen, werden in den Simulationen allerdings nur 100 Sensorknoten dargestellt, was in der Tabelle durch die zweite Ziffer und den Stern gekennzeichnet ist. Bei einer Sendereichweite der Beacons von 35 m und der vorhandenen Beaconkonfiguration ergeben sich vier Cluster, die das Sensornetzwerk möglichst gut abdecken. In jedem Cluster befinden sich im Schnitt 125 Sensorknoten. Wie Abbildung 6.8 (b) zeigt, ist der dabei nötige Zeitaufwand für die Berechnung relativ gering. Der erste Schritt der Berechnung erfolgt nach dem RAL-Algorithmus aus Kapitel 6.1.4. In einem zweiten Schritt werden alle Messungen und die berechnete Knotenposition über die Nachbarknoten an den nächsten Beacon gesendet. Dabei kommt das im vorigen Abschnitt beschriebene "Greedy-Forwarding" zur Anwendung. Abbildung 6.10 (a) zeigt den Simulationsaufbau im Ausgangszustand und (b) mit den ermittelten RAL-Positionen vor der Ausgleichung. Dabei repräsentieren Kreise die exakten Knotenpositionen, die schwarzen Dreiecke die exakten Beaconkoordinaten und die grünen Dreiecke die Clusterheads. Die Sendereichweiten werden als Kreise um die Clusterbeacons dargestellt. Nachdem die Senke alle Messwerte und Positionen erhalten hat, wird eine vermittelnde Ausgleichung nach Kapitel 3.6.3 berechnet. Abbildung 6.11 (a) zeigt die unabhängige Ausgleichung eines jeden Clusters farblich codiert. In Abbildung 6.11 (b) sind die Ergebnisse der Ausgleichung mittels Helmert-Zerlegung für das gesamte Netzwerk dargestellt. In beiden werden die Positionen durch den RAL-Algorithmus durch Punkte repräsentiert und die Ergebnisse der Ausgleichung als Sterne. Die Punktfehler als Fehlerellipsen sind um Faktor 10 vergrößert. Wie zu erwarten war, zeigt die nachträgliche iterative Ausgleichung eine deutliche Verbesserung der Positionsschätzung. Die Positionsschätzung konnte bis zu Faktor 3,34 verbessert werden. Abbildung 6.12 zeigt die durchschnittlichen Punktfehler als farbcodierte Oberfläche über der Fläche des Sensornetzes. In der Grafik ist gut zu erkennen, dass in den Clustern mit geringer Beacondichte oder in den Gebieten mit ungünstiger Netzgeometrie der Fehler maximal wird. Die geringsten Abweichungen ergeben sich in dem Gebiet mit der höchsten Beacondichte. Für den Fall der Ausgleichung mit Blockzerlegung kann der Fehler ebenfalls in den Überlappungsbereichen der Cluster minimiert werden. Die Ergebnisse sind in Tabelle 6.7 zusammengefasst.



Abbildung 6.10: (a) Simulationsaufbau und (b) berechnete RAL-Positionen.



Abbildung 6.11: Sensorpositionen nach (a) Ausgleich pro Cluster und (b) Gesamtberechnung mittels Helmert– Blockzerlegung.



Abbildung 6.12: Mittlerer Punktfehler für RAL ohne Ausgleichung, Ausgleichung mittels Helmert–Blockzerlegung und Einzelausgleichung für jeden Cluster.

6.4.4 Kommunikationsaufwand

Da die Nachausgleichung auf der Senke durchgeführt wird, sind die Sensorknoten nicht an der Berechnung beteiligt. Die Senke unterliegt per definitionem keinen Energie- und Hardwarelimitationen. Aus diesem Grund ist an dieser Stelle auch keine Betrachtung der Rechenkomplexitäten nötig. Wie bereits beschrieben, wird allerdings eine weitere Kommunikationsphase an den prinzipiellen *RAL*–Ablauf angeschlossen. Für das Routing der Datenpakete wird ein "Greedy–Forwarding" benutzt, um ein Fluten der Informationen in das Sensornetzwerk zu vermeiden. Für jeden Sensorknoten müssen n–mögliche Streckenmessungen zu den Beacons, m–mögliche Streckenmessungen zu den Nachbarknoten und die berechneten *RAL*–Position sowie eine Identifizierung des Sensorknotens übermittelt werden. Eine 4 *Byte* Nummer pro Wert vorausgesetzt, ergeben sich so 2 + n + m + 1–Werte für die Kommunikation. In dem untersuchten Szenario können von den 500 Sensorknoten 426 keinen direkten Kontakt zu den Beacons aufbauen. Deren Daten sind über die Nachbarknoten weiterzuleiten. In den Simulationen laufen 53 Routen in eine Sackgasse und müssen neu versendet werden. Pro Sensorknoten werden so Datenpakete von durchschnittlich 20 Knoten weitergegeben. Mit geringerem Abstand zum Beacon nimmt die Kommunikationslast auf den Sensorknoten zu. Die Ergebnisse sind in Tabelle 6.8 aufgelistet.

Mit den gegebenen Beträgen ergibt sich für die zusätzliche Kommunikation für das gesamte Netzwerk ein Mehraufwand von 58.674 *Byte* oder 58,67 *kB*. Dieser hohe Kommunikationsbedarf kommt durch die hohe Sensorknotenanzahl zustande. Da sehr viele Knoten aufgrund ihrer geringen Sendereichweite keinen direkten Kontakt zu den Beacons herstellen können, müssen die Pakete über die anderen Knoten weitergeleitet werden, was den Kommunikationsaufwand enorm erhöht. Um die ermittelten

Parameter	Gesamtes	Ausgleichung	Ohne
	Netz	pro Cluster	Netzausgleichung
Simulationszeit [s]	1.154	201	13
Durchschnittlicher	0.00	0.05	2 19
Fehler [%]	0,99	0,95	5,10
Gewichtseinheit nach der	5.2	18	_
Ausgleichung $\sigma_{0_{aposteriori}}[m]$	5,2	4,0	_
Durchschnittliche	13	8	
Anzahl Iterationen	15	0	_

Tabelle 6.7: Ergebnisse der Nachausgleichung eines großen Sensornetzwerkes.

Größen energetisch analysieren zu können, muss der Kommunikation eine Energie zugewiesen werden. Nach Reichenbach [Rei07] berechnet sich die Sendeenergie nach Gleichung 6.8

$$E_{send} = 1,3235 \cdot 10^{-10} [J] \cdot r^2 - 9,42 \cdot 10^{-10} [J] \cdot r + 6,2181 \cdot 10^{-7} [J] \cdot r^2 [m]$$
(6.8)

mit *r* als Sendereichweite. In diesem Fall wird *r* gleich der Sendereichweite der Sensorknoten, also 10 m gesetzt. Der berechnete Energiebetrag ist dabei die benötigte Energie, die zum Versenden eines Datenpakets von 1 bit Länge aufgewendet werden muss. Da auch der Empfang der Pakete Energie verbraucht, muss auch dieser Betrag berücksichtigt werden. Dieser ergibt sich zu $E_{rec} = 2,3375 \,\mu J/bit$ und wird dem Energiebedarf für die Kommunikation hinzu addiert [Rei07]. Abbildung 6.13 (b) zeigt den Energieverbrauch im Netzwerk über einer steigenden Anzahl von Beacons. Des Weiteren ist in



Abbildung 6.13: (a) Anzahl Knoten ohne Verbindung zu Beacons und (b) Energieverbrauch über steigender Beaconzahl bei 500 Sensorknoten.

Abbildung 6.13 (a) die Anzahl von Sensorknoten dargestellt, die keine Verbindung zu den Beacons direkt aufbauen kann, d. h., alle Daten müssen über die Nachbarknoten zu den Beacons gesendet werden. Die Anzahl der Sensorknoten ohne Verbindung zu den Beacons sinkt mit zunehmender

Parameter		Clu	ster		Gesamtnetz
	1	2	3	4	
Anzahl	140	07	112	140	500
Sensorknoten	142	97	115	140	500
Anzahl	5	7	11	10	25
Beacons	5	7	11	10	23
Knoten ohne					
direkten Kontakt	126	89	97	114	426
zu den Beacons					
Kommunikations-					
aufwand [Byte]	1.846	1.455	2.147	2.664	15.000
(direkt)					
Kommunikations-					
aufwand [Byte]	11.414	11.505	15.029	15.156	58.674
(indirekt)					
Energie-					
aufwand [Joule]	0,043	0,034	0,050	0,062	0,352
(direkt)					
Energie-					
aufwand [Joule]	0,158	0,128	0,179	0,207	0,911
(indirekt)					

Tabelle 6.8: Kommunikationsaufwand für die Nachausgleichung pro Cluster und im gesamten Netzwerk.

Beaconanzahl. Es müssen weniger Daten über die Nachbarknoten weitergesendet werden, da immer mehr Pakete direkt vom Sensorknoten zu den Beacons gesendet werden können. Der energetische Aufwand steigt jedoch weiter an, da zusätzliche Distanzbeobachtungen verschickt werden müssen. Auch erhöht sich die Anzahl der Nachbarknoten in Sendereichweite, womit ebenfalls die Anzahl zusätzlicher Streckenbeobachtungen zu den direkten Nachbarn ansteigt. Eine einfache Erhöhung der Beaconanzahl genügt also nicht, um das Kommunikationsaufkommen zu senken.

6.5 Fazit

Der in diesem Kapitel erläuterte *RAL* bietet die Möglichkeit, Sensorknoten in einem Sensornetzwerk mit einer hohen Genauigkeit zu lokalisieren, ohne die limitierten Energieressourcen der Knoten zu sehr zu belasten. Weiterhin ist dieser Algorithmus auf große Sensornetzwerke ausgedehnt und es sind Möglichkeiten zur Clusterisierung vorgeschlagen worden, um den Berechnungs- und vor allem den

Kommunikationsaufwand zu senken. Darüber hinaus wurden die berechneten Sensorpositionen einer Gesamtnetzausgleichung zugeführt, was eine signifikante Genauigkeitssteigerung zufolge hat. Die dafür notwendigen Kommunikationkosten erhöhen sich bei diesem Ansatz. Jedoch ist diese Erhöhung nicht so groß, als dass sie die Lebensdauer des Netzwerkes signifikant verkürzt.

7 Genauigkeitsbetrachtungen zu ausgewählten Lokalisierungsalgorithmen

"Die Statistik ist ein Verfahren, welches es gestattet, geschätzte Größen mit der Genauigkeit von Hundertstelprozent auszudrücken."

(Helmar Nahr, dt. Mathematiker)

In diesem Kapitel werden einige ausgewählte Lokalisierungsalgorithmen hinsichtlich ihrer Genauigkeit bei der Positionsbestimmung untersucht und miteinander verglichen. Dabei werden vor allem der Einfluss der Geometrie des Netzwerkes und grob fehlerhafter Streckenbeobachtungen analysiert. Die nachfolgenden Analysen erfolgten projektübergreifend und in enger Kooperation mit Dipl.-Ing. Frank Niemeyer aus dem BMBF–Projekt *SLEWS* (vgl. Kapitel 2.4) der Professur für Geodäsie und Geoinformatik der Universität Rostock.

7.1 Untersuchte Algorithmen

Als zu untersuchende Algorithmen dienen der "Weighted Centroid Localization"–Algorithmus (*WCL*), der "Distributed Least Squares"–Algorithmus (*DLS*) sowie der neu eingeführte "Resource Aware Localization"–Algorithmus (*RAL*) (vgl. Kapitel 4.2, 4.3, 6.1). Bevor die Methoden zur Untersuchung beschrieben werden, sollen die Grundprinzipien der zu analysierenden Algorithmen kurz aufgeführt werden. In der Literatur existieren für *DLS* und *WCL* diverse Abwandlungen (vgl. Kapitel 4). Da jedoch alle auf denselben Grundalgorithmen beruhen, wird der ursprüngliche Ansatz in den Simulationen verwendet.

7.1.1 "Weighted Centroid Localization"-Algorithmus

Wie bereits in Kapitel 4.2 beschrieben, handelt es sich bei dem *WCL* um einen approximativen Algorithmus. In der ersten Phase übermitteln alle Beacons ihre Position $B(x_j, y_j)$ an alle Sensorknoten in Sendereichweite. Während des Empfangs bestimmen die Knoten die Distanzen zu den Ankerknoten und speichern diese zusammen mit den empfangenen Koordinaten. Nachdem alle Positionen und Strecken gespeichert wurden, berechnet jeder einzelne Sensorknoten seine ungefähre Position durch eine gewichtete Schwerpunktbestimmung mit allen empfangenen n-Beacons nach Gleichung 7.1.

$$\hat{P}(\hat{x}_N, \hat{y}_N) = \frac{\sum_{j=1}^n (w_j \cdot \tilde{B}(\hat{x}_j, \hat{y}_j))}{\sum_{j=1}^n w_j}$$
(7.1)

Dabei benutzt der *WCL* die gemessenen Strecken zu den Beacons als Gewicht w_j . Die Distanzen \hat{d}_j fließen umgekehrt proportional in die Gewichtsberechnung ein (Gleichung 7.2).

$$w_j = \frac{1}{\hat{d}_j^g} \tag{7.2}$$

Die Distanz wird mit einem Wert *g* potenziert. Diese Potenz wurde in einigen Arbeiten untersucht und soll in dieser Analyse hinsichtlich des Einflusses der Netzgeometrie weiter betrachtet werden [BRT05b, Blu08]. Die Gewichtung der Distanzen hat nach Gleichung 7.1 und 7.2 den Effekt, dass nahe Beaconpositionen einen größeren Einfluss auf die errechneten Sensorpositionen haben als weiter entfernte. In Kapitel 4.4.1 wurde dazu detailliert auf die Streckenbestimmung in Sensornetzwerken eingegangen.

7.1.2 Standardlokalisierung

Für eine zweidimensionale Positionsbestimmung werden mindestens drei Beacons benötigt. Die weitere Berechnung erfolgt dann über das Lösen eines entstehenden überbestimmten Gleichungssystems, welches mit dem Gauß-Markow-Verfahren (vgl. Kapitel 3.6.3) gelöst wird. Der Algorithmus wird in dieser Arbeit als Standardlokalisierung (*SL*) eingeführt und läuft wie folgt ab: alle Sensoren bestimmen die Distanzen zu den bekannten Beacons in Kommunikationsreichweite. Mit der Euklidischen Distanz \hat{d}_j , den Positionen der Sensorknoten $\hat{P}(\hat{x}_N, \hat{y}_N)$ und den fehlerfreien Koordinaten der Beacons $\hat{B}(\hat{x}_j, \hat{y}_j)$ kann das überbestimmte Gleichungssystem formuliert werden

$$\hat{d}_{j}^{2} = (\hat{x}_{N} - \hat{x}_{j})^{2} + (\hat{y}_{N} - \hat{y}_{j})^{2}.$$
(7.3)

Bevor dieses nichtlineare Gleichungssystem mit der Methode der kleinsten Quadrate gelöst werden kann, ist es in eine lineare Form zu überführen. Dieses erfolgt über eine Taylorreihenentwicklung und Abbruch nach dem ersten Glied (vgl. Kapitel 3.2.1). Da es sich um ein iteratives Verfahren handelt, sind Startwerte für die Linearisierung erforderlich. In einem Sensornetzwerk sind die Positionen der Knoten jedoch anfänglich unbekannt (vgl. Kapitel 2). Aus diesem Grund muss diesem Algorithmus ein approximativer vorgeschaltet werden. In Born et al. wurden einzelne Algorithmen dazu untersucht [BRBT08]. Der iterative Charakter dieses Verfahrens führt dazu, dass die Standardlokalisierung als der genaueste der hier untersuchten Algorithmen einzuschätzen ist. Deswegen dient er als Referenzalgorithmus für die nachfolgenden Untersuchungen.

7.1.3 "Distributed Least Squares"–Algorithmus

Der "Distributed Least Squares"–Algorithmus basiert wie der *FGL* ebenfalls auf der Lösung eines überbestimmten Gleichungssystems mit Hilfe der Methode der kleinsten Quadrate auf Basis der Euklidischen Distanzen zwischen den Beacons und den Sensorknoten [RBTB06]. Allerdings wird das entstehende nichtlineare Gleichungssystem nicht mittels Taylorreihenentwicklung linearisiert. Um auf die vorangehende Berechnung von Startwerten, wie etwa beim *FGL*–Algorithmus, zu verzichten, werden die Beobachtungsgleichungen 7.3 durch Termerweiterung und Umstellen in ein lineares Gleichungssystem umgeformt. Es ergibt sich Gleichung 7.4

$$(\hat{x}_N - \tilde{x}_j) \cdot (\tilde{x}_i - \tilde{x}_j) + (\hat{y}_N - \tilde{y}_i) \cdot (\tilde{y}_i - \tilde{y}_j) = \frac{1}{2} \left(\hat{r}_j^2 - \hat{r}_i^2 + \hat{d}_{ij}^2 \right) = b_{ij}$$
(7.4)

Diese Umstellung führt zu der bekannten Matrixdarstellung Ax = b mit der Koeffizientenmatrix A, dem Vektor b und dem Lösungsvektor x. Die Anwendung der Methode der kleinsten Quadrate führt zu der Form $x = (A^TA)^{-1}A^Tb$. Die Termerweiterung und die Umformung führen nun analog zu dem in Kapitel 6.1 beschriebenen Vorteil, dass die Gesamtberechnung in einen komplexen und einen einfachen Teil aufgespalten und getrennt voneinander berechnet werden kann.

7.1.4 "Resource Aware Localization"–Algorithmus

Prinzipiell folgt der "Resource Aware Localization"–Algorithmus dem *DLS*, mit dem Unterschied, dass hier lediglich eine Substitution vorgenommen wird, um die nichtlinearen Terme im Gleichungssystem zu eliminieren (vgl. Kapitel 6.1). Das entstehende lineare Gleichungssystem hat die Form:

$$\hat{d}_{i}^{2} - \tilde{x}_{i}^{2} - \tilde{y}_{i}^{2} = \hat{w} - 2\hat{x}_{N}\tilde{x}_{i} - 2\hat{y}_{N}\tilde{y}_{i} = b_{i} \quad mit \quad \hat{w} = \hat{x}_{N}^{2} + \hat{y}_{N}^{2}.$$
(7.5)

Der Unterschied zwischen *DLS* und *RAL* ist nicht sehr groß. Allerdings ergeben sich für die Positionsgenauigkeit Unterschiede, die im Folgenden genauer untersucht werden sollen.

7.2 Simulationen und Ergebnisse

Zur Untersuchung des Genauigkeitsverhaltens der Lokalisierungsalgorithmen wird eine empirische Herangehensweise gewählt. Aussagekräftige Verhaltensmuster können damit nur über eine hohe Anzahl von zufällig verfälschten Beobachtungssätzen gezeigt werden. Als Voraussetzung wird ein Beaconfeld und eine Sensorposition gewählt, aus denen sich die wahren euklidischen Distanzen berechnen lassen. Zusätzlich ist ein Varianzbereich für die Streckenbeobachtungen vorgegeben, in dem die wahren Distanzen zufällig verfälscht werden. Dieser zufällig verfälschte Beobachtungssatz sowie die Beaconund Neupunktkoordinaten werden den einzelnen Algorithmen zugeführt. Die Beaconpositionen gehen unverfälscht in die Untersuchungen ein. Im zweiten Schritt erfolgt ein Vergleich der daraus berechneten Neupunktkoordinaten mit der wahren Position. Der mittlere Fehler der Einzelmessung wird nach jedem Simulationsschritt berechnet und dargestellt. Es wird erwartet, dass bei zunehmender Anzahl von Simulationsdurchläufen eine Stabilisierung des mittleren Fehlers erfolgt. Das heißt, die empirischen Effekte werden minimiert, sodass sich der empirisch ermittelte Fehler seinem theoretischen Wert annähert.

AccuLoc

Bestehenden Netzwerksimulatoren ist gemein, dass keiner die Lokalisierungsfehler oder die statistische Sicherheit von Lokalisierungsalgorithmen evaluiert. Aus diesem Grund wurde im Zuge der Arbeit der Simulator *AccuLoc* (**Accu**racy Analyzer for **Loc**alization in Wireless Sensor Networks) entwickelt, der diese Lücke schließt und Genauigkeitsanalysen aus geodätischer Sicht ermöglicht. Ebenfalls kann vor Ausbringung des Sensornetzwerkes eine Vorabanalyse der erwartbaren Genauigkeiten und eine Netzplanung nach Kapitel 3.6.5 erfolgen. Aufgrund des Umfangs wird auf eine grundlegende Beschreibung von *AccuLoc* im Rahmen dieser Arbeit verzichtet. Alle folgenden Simulationen wurden, soweit nicht anders beschrieben, mit *AccuLoc* durchgeführt. Für detaillierte Informationen zu dem Simulator sei jedoch auf [BN09] verwiesen.

Erzeugung von Zufallszahlen für die Simulation fehlerbehafteter Beobachtungsgrößen

Die Simulation von Lokalisierungsalgorithmen soll aussagekräftige Ergebnisse unter realitätsnahen Bedingungen ermöglichen. Das heißt vielmehr, die Modellierung umweltspezifischer Einflüsse, welche das Ergebnis der Positionsbestimmung beeinflussen. Speziell der Generierung von Messwerten kommt dabei eine große Bedeutung zu. Dabei gestaltet sich die automatische Generierung von echten Zufallszahlen schwierig. Zu deren Erzeugung greift jeder Rechner auf Algorithmen zurück. Deren Systematik kann wiederum die Erzeugung von echten Zufallszahlen beeinträchtigen. In der Literatur hat sich dafür der Begriff Pseudo-Zufallszahl manifestiert.

Im Detail heißt das, sind nun die Initialisierungsbedingungen bei jedem Programmstart die selben, werden zwangsläufig auch immer die selben Zufallszahlenwerte generiert. Um das zu verhindern, ist es möglich, die Startwerte für den Algorithmus mit Hilfe der inneren Uhr des Rechners zu bestimmen. Dieses Vorgehen garantiert bei jedem neuen Programmstart die Erzeugung "echter" Zufallszahlen, welches auch in *AccuLoc* Anwendung findet.

Die unterschiedlichen Verfahren zur Bestimmung der in *AccuLoc* verwendeten Distanzen unterliegen unterschiedlichen Verteilungsfunktionen, die bei der Simulation von fehlerhaften Messwerten zu berücksichtigen sind. Im Wesentlichen sind das Distanzen aus Laufzeitmessungen, die einer Normalverteilung, sowie Distanzschätzungen aus Signalempfangsstärkemessungen (vgl. Kapitel 4.4.1), die einer Lognormalverteilung unterliegen. Des Weiteren wurden Signalausbreitungsmodelle hinzugefügt. Eine detaillierte Beschreibung dieser Modelle erfolgte bereits in Kapitel 5.2.2.

7.2.1 WCL-Algorithmus

In Kapitel 4.2 wurde der *WCL*–Algorithmus bereits ansatzweise beschrieben. An dieser Stelle erfolgen detailliertere Erläuterungen und eine Fehlerrechnung auf Basis fehlerhafter Streckenmessungen und Beaconkoordinaten.

Da der *WCL* auf der Schwerpunktbestimmung beruht, ist er als approximativer Algorithmus zu klassifizieren. Der "Coarse Grained Localization"–Algorithmus (*CGL*, vgl. Kapitel 4.2) wird um eine Gewichtung erweitert, die die Streckenmessungen zu den verfügbaren Beacons enthält. Die Distanzmessungen fließen dabei umgekehrt proportional in die Berechnung ein. Mit anderen Worten gehen weiter entfernte Festpunkte mit einem geringeren Gewicht in die Schwerpunktbestimmung ein als nahe Beacons. Das hat zur Folge, dass die Neupunktkoordinaten stärker zu den dichten Beacons herangezogen werden. Die gewichtete Schwerpunktmethode berechnet sich nach Gleichung 7.6

$$\hat{x}_{N} = \frac{\sum_{j=1}^{n} (w_{j} \cdot \hat{x}_{j})}{\sum_{j=1}^{n} w_{j}}, \quad \hat{y}_{N} = \frac{\sum_{j=1}^{n} (w_{j} \cdot \hat{y}_{j})}{\sum_{j=1}^{n} w_{j}}$$
(7.6)

mit den Beaconkoordinaten \hat{x}_j und \hat{y}_j , der Position des gewichteten Schwerpunktes \hat{x}_N und \hat{y}_N sowie dem Gewicht w_j vom Neupunkt zu Festpunkt *j*. Für die Gewichte ergibt sich nach [Blu08] die Berechnungsformel 7.7

$$w_j = \frac{1}{\left(\hat{d}_j\right)^g} \tag{7.7}$$

mit der gemessenen und damit fehlerbehafteten Distanz \hat{d}_j zwischen dem Neupunkt und Beacon *j*. Die Berechnungsformel für den *WCL* lautet dann:

$$\hat{x}_{N} = \frac{\sum_{j=1}^{n} \left(\frac{1}{(\hat{d}_{jN})^{g}} \cdot \hat{x}_{j} \right)}{\sum_{j=1}^{n} \frac{1}{(\hat{d}_{jN})^{g}}}, \quad \hat{y}_{N} = \frac{\sum_{j=1}^{n} \left(\frac{1}{(\hat{d}_{jN})^{g}} \cdot \hat{y}_{j} \right)}{\sum_{j=1}^{n} \frac{1}{(\hat{d}_{jN})^{g}}}.$$
(7.8)

Hinzu kommt eine Potenz *g*, mit der die Strecke bei der Gewichtung versehen wird. Diese Potenz übt einen starken Einfluss auf die Neupunktkoordinaten aus und ist im weiteren Verlauf Gegenstand gesonderter Untersuchungen.

7.2.1.1 Varianzfortpflanzung für den WCL

Für die folgenden Genauigkeitsanalysen soll für den *WCL*–Algorithmus die Positionsgenauigkeit nach dem Varianzfortpflanzungsgesetz (vgl. Kapitel 3.5) hergeleitet werden. Der *WCL* kann über Gleichung 7.6 berechnet werden. Die fehlerbehafteten Distanzmessungen können als voneinander unabhängig, also unkorreliert, betrachtet werden. Die Beaconkoordinaten $\hat{B}_j(\hat{x}, \hat{y})$ sind aus vorhergehender Positionsbestimmung ebenfalls fehlerbehaftet. Zwischen den Koordinatenanteilen wird zur Vereinfachung ebenfalls keine Korrelation vorausgesetzt, wobei dieses Vorgehen in der Realität untypisch ist. Allerdings sind die Fehleranteile der Beaconkoordinaten im Verhältnis zu den hoch ungenauen Distanzmessungen in drahtlosen Sensornetzwerken als vernachlässigbar einzustufen (siehe Kapitel 4.4). In diesen Betrachtungen werden jedoch die mittleren Fehler der Beaconkoordinaten mit aufgeführt.

Für die Berechnung der Punktfehler der Schwerpunktkoordinaten ist die Kovarianzmatrix (Gleichung 7.9) der Unbekannten aufzustellen.

$$\mathbf{C}_{\mathbf{FF}} = \begin{bmatrix} \sigma_{\hat{x}_N}^2 & \operatorname{cov}(\hat{x}_N, \hat{y}_N) \\ \operatorname{cov}(\hat{y}_N, \hat{x}_N) & \sigma_{\hat{y}_N}^2 \end{bmatrix}$$
(7.9)

Die Elemente der Kovarianzmatrix der Unbekannten sind auf der Hauptdiagonalen die quadratischen mittleren Fehler $\sigma_{\hat{x}_N}^2$ und $\sigma_{\hat{y}_N}^2$ der Koordinatenanteile \hat{x}_N und \hat{y}_N des Unbekannten und die dazugehörigen Kovarianzen $\operatorname{cov}(\hat{x}_N, \hat{y}_N)$ auf der Nebendiagonalen.

Die Kovarianzmatrix der Unbekannten ist Ergebnis des Varianzfortpflanzungsgesetzes (Gleichung 7.10).

$$\mathbf{C}_{\mathbf{F}\mathbf{F}} = \mathbf{F} \cdot C_{ll} \cdot \mathbf{F}^T \tag{7.10}$$

Die Elemente sind detailliert in Kapitel 3.5 beschrieben. Die F-Matrix beinhaltet die partiellen Ableitungen der Beobachtungsgleichungen 7.6 nach den Beobachtungen:

Da die Beobachtungen als unkorreliert angenommen wurden, sind die Nebendiagonalelemente der C_{II} -Matrix (Gleichung 7.12) null. Auf der Hauptdiagonalen finden sich die Varianzen der Beobachtungen wieder.

	_								_
	$\sigma^2_{\hat{d}_{1N}}$	0	0	0	0	0	0	0	0
	0	·	0	0	0	0	0	0	0
	0	0	$\sigma^2_{\hat{d}_{nN}}$	0	0	0	0	0	0
	0	0	0	$\sigma^2_{\hat{x}_1}$	0	0	0	0	0
$C_{ll} =$	0	0	0	0	$\sigma_{\hat{y}_1}^2$	0	0	0	0
	0	0	0	0	0	·	0	0	0
	0	0	0	0	0	0	·	0	0
	0	0	0	0	0	0	0	$\sigma^2_{\hat{x}_n}$	0
	0	0	0	0	0	0	0	0	$\sigma_{\hat{y}_n}^2$

7.2.1.2 Methode zur Bestimmung der optimalen Potenz g

Der WCL–Algorithmus geht davon aus, dass eine (Funk-)Verbindung zwischen Neupunkt und Beacon besteht. In Abhängigkeit der Lage bzw. der Schnittgeometrie und der Potenz *g* ergeben sich unterschiedliche Lagegenauigkeiten in einem Testfeld. Folgendes Kriterium soll dabei angestrebt werden:

$$\lim\left(\sum_{k=1}^{n}\sqrt{\left(\left(\hat{y}_{N}-\tilde{y}_{N}\right)^{2}+\left(\hat{x}_{N}-\tilde{y}_{N}\right)^{2}\right)}\right)\to\min$$
(7.13)

In Abbildung 7.1 ist der mittlere Punktfehler der Neupunkte über die untersuchte Potenz (in den Grenzen von -10 bis +10) beispielhaft dargestellt. Es ist deutlich ein Minimum zu erkennen, dass bei g = 1, 8 liegt. Da in dieser Untersuchung die optimale Potenz nicht analytisch, sondern lediglich über einen Bereich von -10 bis +10 empirisch hergeleitet wird, kann keine Aussage darüber getroffen werden, ob es sich in der ermittelten Potenz um ein globales oder ein lokales Minimum handelt. Die Potenz g kann daher nur in Abhängigkeit der Schrittweite bestimmten Genauigkeit angegeben werden.

7.2.1.3 Simulationsaufbau

In Tabelle 7.1 sind die untersuchten Beaconkonfigurationen aufgeführt. Die K_R^i bezeichnet die Konfigurationen, bei denen eine Beaconverteilung im Raster und K_Z^i bei denen eine zufällige Beaconverteilung vorliegt. In den dargestellten Grafiken werden die Sensorknoten durch Dreiecke und die Beacons durch Punkte repräsentiert.

Einfluss von fehlerbehafteten Beaconkoordinaten

Mit der beschriebenen Varianzfortpflanzung wird nun der Fehlereinfluss der Lagegenauigkeit der Festpunkte dargestellt. Es sollen dabei unterschiedliche Positionsgenauigkeiten der Beacons untersucht werden (vgl. Tabelle 7.1). In Abbildung 7.2 sind die Positionsfehler der Sensorknoten in einem festen Beaconraster und in Abbildung 7.4 bei zufälliger Beaconverteilung dargestellt. Bei fehlerhaften Beacon-



Abbildung 7.1: Bestimmung der optimalen Potenz g (grünes Dreieck) für WCL (nach F. Niemeyer).



Abbildung 7.2: WCL nach Konfigurationen (a) K_R^1 und (b) K_R^2 .

koordinaten ist zu erkennen, dass der Positionsfehler mit abnehmender Entfernung zu den Beacons zunimmt. Am geringsten fällt dieser in der Nähe des Schwerpunktes im Zentrum des Sensorfeldes aus. Die rot gekennzeichneten Vektoren beschreiben die Abweichung der wahren von der geschätzten Position. Je dichter sich die Sensorknoten am Perimeter des Beaconfeldes (gekennzeichnet durch die Umrandung mit den Beacons als Eckpunkte) befindet, desto größer wird die Abweichung. Dabei fällt der mittlere Punktfehler geringer aus als die eigentliche Verschiebung. Das lässt sich dadurch

Tabelle 7.1: Simulationsaufbau Beacon- und Distanzfehler für WCL, 81 Sensorknoten im Raster (Schrittweite 10 m) und 5 % Beacons. Dabei bezieht sich K_R auf die rasterförmige und K_Z auf die zufällige Anordnung der Beacons. *i gibt die verschiedenen Varianten wieder.*

Konfiguration	Fehler Beacon-	Distanz-	Gewicht
K_R^i ; K_Z^i	koordinaten in \hat{x}_i und \hat{y}_i [m]	fehler [m]	
K_R^1	$\forall \hat{x}_i, \hat{y}_i = 3, 0$	0,0	1,81
K_R^2	$\forall \hat{x}_i, \hat{y}_i = 5, 0$	0,0	1,81
K_R^3	$B_1:5,0;5,0;B_2:1,0;1,0;$	0.0	1 81
	$B_3: 10, 0; 10, 0; B_4: 3, 0; 3, 0$	0,0	1,01
K_R^4	$B_1:5,0;3,0;B_2:1,0;1,0;$	0.0	1 81
	$B_3:5,0;10,0;B_4:3,0;7,0$	0,0	1,01
K_R^5	0,0	5,0	1,81
K_R^6	0,0	10,0	1,81
K_R^7	5,0	5,0	1,81
K_R^8	$B_1:5,0;3,0;B_2:1,0;1,0;$	5.0	1 81
	$B_3:5,0;10,0;B_4:3,0;7,0$	5,0	1,01
K_R^9	0,0	0,0	1,81
K_{R}^{10}	0,0	0,0	10,0
K_{R}^{11}	0,0	0,0	3,0
K_{R}^{12}	0,0	0,0	-10,0
K_Z^1	3,0	0,0	1,83
K_Z^2	5,0	0,0	1,83
K_Z^3	$B_1:5,0;5,0;B_2:1,0;1,0;$	0.0	1 82
	$B_3: 10, 0; 10, 0; B_4: 3, 0; 3, 0$	0,0	1,00
K_Z^4	$B_1:5,0;3,0;B_2:1,0;1,0;$	0.0	1.83
	$B_3:5,0;10,0;B_4:3,0;7,0$	0,0	1,00
K_Z^5	0,0	5,0	1,83
K_Z^6	0,0	10,0	1,83
K_Z^7	5,0	5,0	1,83
K_Z^8	$B_1:5,0;3,0;B_2:1,0;1,0;$	5.0	1 83
	$B_3:5,0;10,0;B_4:3,0;7,0$	5,0	1,00

erklären, dass der *WCL* auf der Schwerpunktbestimmung beruht. Die mittleren Punktfehler resultieren allerdings ausschließlich aus den Positionsungenauigkeiten der Beacons. Da der Positionsfehler in \hat{x}_i

und \hat{y}_i gleich gewählt wurde und alle Beacons den selben Fehler aufweisen, sind die Fehlerellipsen für die Neupunkte in dieser Konfiguration ebenfalls kreisförmig. Bei unterschiedlichen Fehlern in den Beaconkoordinaten werden die Fehlerkreise zu Fehlerellipsen. Dieser Sachverhalt ist in Abbildung 7.3 dargestellt.



Abbildung 7.3: WCL nach Konfigurationen (a) K_R^3 und (b) K_R^4 .

In einer nächsten Simulation werden die Beacons zufällig im Sensorfeld verteilt. Die dafür gewählten Parameter sind in Tabelle 7.1 unter Konfigurationen K_7^1 bis K_7^4 wiedergegeben und die Ergebnisse in den Abbildungen 7.4 und 7.5 aufgezeigt. Es lässt sich die typische Natur einer Schwerpunktbestimmung in den Abbildungen beobachten. Alle Sensorknoten, die außerhalb des Beaconperimeters liegen, werden in ihn hineingezogen. Der mittlere Punktfehler hingegen spiegelt den Einfluss des Koordinatenfehlers der Beacons wieder. Bei gleich genauen Beaconkoordinaten und gleichen Fehleranteilen in den Koordinatenachsen werden die Fehlerellipsen als Kreise abgebildet. Erhalten jedoch die Beaconkoordinaten wie in der ersten Simulation unterschiedliche Fehler, drückt sich das auch in den Fehlern der Sensorknotenpositionen aus (Abbildung 7.5). In allen bisher beschriebenen Simulationen ist zu erkennen, dass der mittlere Punktfehler in der Nähe der Beacons am größten wird. Mit abnehmender Distanz zum Schwerpunkt des durch die Beacons definierten Feldes werden die mittleren Punktfehler kleiner, bleiben aber bei sich ändernden Beaconkonfigurationen gleich. Der Lagefehler, d. h., der systematische Einfluss der Schwerpunktbestimmung, indes ist in Schwerpunktnähe des Beaconfeldes am geringsten, ändert sich allerdings mit der Beaconkonfiguration. Soll also ein Sensorfeld über einem Gebiet von Interesse ausgebracht werden, sollte darauf geachtet werden, dass sich die Beacons möglichst gut einer gleichförmigen oder Rasterverteilung annähern. Sensorknoten außerhalb des Feldes werden durch die Anwendung von WCL in Richtung Schwerpunkt gezogen. Der Positionsfehler wird maximal.



Abbildung 7.4: WCL nach Konfigurationen (a) K_Z^1 und (b) K_Z^2 .



Abbildung 7.5: WCL nach Konfigurationen (a) K_Z^3 und (b) K_Z^4 .

Einfluss von Distanzfehlern

Unter den im vorhergehenden Abschnitt beschriebenen Voraussetzungen wird nun der Einfluss fehlerbehafteter Entfernungen zwischen den Beacons und den Neupunkten dargestellt (Abbildung 7.6). Die Simulationen erfolgen auf Grundlage der in Tabelle 7.1 aufgeführten Konfigurationen K_R^5 , K_R^6 sowie K_Z^5 und K_Z^6 . Die Ergebnisse sind in den Abbildungen 7.6 für eine Rasterverteilung und in Abbildung 7.7 für eine zufällige Verteilung der Beacons wiedergegeben. Bei fehlerbehafteten Streckenmessungen



Abbildung 7.6: WCL nach Konfigurationen (a) K_R^5 und (b) K_R^6 .

wird der große Einfluss der Distanzen sichtbar. Obwohl es sich beim *WCL* im Wesentlichen um eine Schwerpunktbestimmung handelt, beeinflussen die Distanzfehler die Lokalisierung in weit höherem Maße als fehlerbehaftete Beaconkoordinaten. Der mittlere Punktfehler nimmt bereits bei geringen Distanzfehlern von 5 % größere Werte an. Besonders in der Nähe des Beaconperimeters wirken sich diese Effekte stark aus. Die Sensorpositionen an den Rändern erfahren einen deutlichen Zug zu den Beacons, was an den Fehlerellipsen ersichtlich wird. Auch der Lagefehler, hervorgerufen durch die Schwerpunktmethode, zieht sich mehr zu den Festpunkten. Bei einer zufälligen Beaconverteilung, Konfiguration K_Z^5 und K_Z^6 , sind die Distanzabhängigkeiten noch stärker ausgeprägt. Die Fehlerellipsen werden in größerem Maße verzerrt als bei der rasterförmigen Anordnung der Beacons. Gerade in den Randbereichen des Festpunktperimeters erfahren die Sensorknoten starke Punktfehler. Selbst die Knoten, die am Rand innerhalb des Perimeters liegen, werden sehr stark in Richtung Schwerpunkt verschoben. Zusätzlich erfahren sie eine Verschiebung in Richtung des nächsten Beacons. Mit steigendem Distanzfehler (Abbildung 7.7 (b)) sind die Positionsfehler so groß, dass eine zuverlässige Positionsbestimmung nicht möglich ist. Positionen in der Nähe des Schwerpunktes können Lageverschiebungen erfahren, die größer als der angenommene Distanzfehler sind.

Einfluss bei gleichzeitigem Auftreten von Beaconkoordinaten- und Distanzbeobachtungsfehlern

In einem letzten Schritt werden die Positionsfehler der Neupunktkoordinaten bei gleichzeitigem Auftreten von Distanzfehlern und Fehlern in den Beaconkoordinaten untersucht. Die Konfiguration ist für diese Simulationen in Tabelle 7.1 unter K_R^7 , K_R^8 für Beacons im Raster sowie K_Z^7 und K_Z^8 für eine zufällige



Abbildung 7.7: WCL nach Konfigurationen (a) K_Z^5 und (b) K_Z^6 .

Beaconverteilung gegeben. Die Ergebnisse sind in den Abbildungen 7.8 und 7.9 wiedergegeben. Bei einer Beaconanordnung im Raster erfolgt eine Überlagerung der einzelnen Fehlereinflüsse. Bei



Abbildung 7.8: WCL mit fehlerbehafteten Streckenbeobachtungen und fehlerbehafteten Beaconkoordinaten nach Konfiguration (a) K_R^7 und (b) K_R^8 .

mittleren Punktfehlern, die bei in Abbildung 7.2 noch kreisrunde Fehlerellipsen aufwiesen, ist bedingt

durch den Einfluss des Streckenfehlers eine Ellipsenform zu erkennen (Abbildung 7.8). Allenfalls im Schwerpunkt des Beaconfeldes ist ein Fehlerkreis verblieben. Bedingt durch die Schwerpunktmethode und die gewählte Beaconverteilung, wirken alle Festpunkte gleich stark auf den Sensorknoten, was die Kreisausbildung zur Folge hat. Im Fall der zufällig verteilten Beacons steigt der mittlere Punktfehler



Abbildung 7.9: WCL mit fehlerbehafteten Streckenbeobachtungen und fehlerbehafteten Beaconkoordinaten nach Konfiguration (a) K_Z^7 und (b) K_Z^8 .

an (Abbildung 7.9). Die Lagefehler in Nähe des Schwerpunktes werden ebenfalls größer. In diesem Szenario werden sämtliche Sensorknotenpositionen verschoben, selbst wenn sie sich in Nähe des Schwerpunktes befinden. Der Positionsfehler der Sensorknoten steigt stark an und die resultierenden Positionsschätzungen sind sehr unzuverlässig.

Ermittlung der optimalen Potenz g

Für die vorangegangenen Simulationen wurde die optimale Potenz *g* für die Streckengewichtung im Vorfeld bestimmt und für die Berechnungen festgelegt. In den einzelnen Konfigurationen blieb die Gewichtung jedoch gleich, um verlässliche Aussagen über die Einflüsse der jeweiligen Parameter treffen zu können. Sie wurde nur geändert, wenn eine wechselnde Netzgeometrie dies erforderte. Diese Annahme soll in Bezug auf sich ändernde Netzwerkgeometrien untersucht und Aussagen über die optimale Wahl des Gewichts und deren Folgen betrachtet werden.

Mit dem in Gleichung 7.13 eingeführten Kriterium soll der Gewichtseinfluss untersucht werden. Zu diesem Zweck wird die optimale Potenz in dem Intervall $g = -10 \le g \le 10$ gesucht, welches zwischen -10 und 10 in 2.000 Teile diskretisiert wird. Somit ergibt sich eine Berechnungssprungweite von 0, 01. Das heißt, die ermittelte Potenz wird auch nur mit der Genauigkeit in der Größe der Berechnungssprungweite angegeben.

Für das Beispielfeld aus Konfiguration K_R^9 wird die optimale Potenz mit g = 1,81 nach Gleichung 7.13 berechnet und mit der gewählten Potenz aus $K_R^{10} - K_R^{12}$ verglichen. Die Ergebnisse sind in Abbildung 7.10 dargestellt. Die Strecken- und Beaconkoordinatenfehler sind dabei auf 0 gesetzt, um einen direkten Vergleich der unterschiedlichen Gewichte zuzulassen. Die Konfiguration K_R^9 in Abbildung 7.10 stellt



Abbildung 7.10: Vergleich (a) optimale Potenz g = 1,81 aus K_R^9 , (b) mit gewählten Potenzen g = 10,0 aus K_R^{10} , (c) g = 3,0 aus K_R^{11} sowie (d) g = -10,0 aus K_R^{12} .

das Optimalgewicht dar, das nach Gleichung 7.13 bestimmt wurde. Die resultierenden Lagefehler im Sensorfeld fallen relativ gering aus. Nur zu den Beacons hin wird der Fehler größer, was durch

die Gewichtung mit zunehmender Nähe zu den Beacons bewirkt wird. Im Fall K_R^{11} liegt die Potenz des Gewichts bei g = 3,0 und damit nur etwas oberhalb der optimalen Schätzung. Für den Fall g = 0,0 wird die Gewichtete Schwerpunktbestimmung zur normalen Schwerpunktbestimmung (*CL*– Algorithmus) und alle Positionen werden im Schwerpunkt lokalisiert. Als nächste Gewichte werden das obere und untere Limit (K_R^{10} und K_R^{12}) der Intervallgrenzen gewählt. Die hohe Potenz bewirkt eine Verschiebung aller Sensorknotenpositionen zu den Beacons. Die Sensorknoten auf den Halbierenden des Beaconperimeters werden auf den Perimeter verschoben, der Sensorknoten im Schwerpunkt des Beaconfeldes bleibt im Zentrum.

Konfiguration K_R^{12} hingegen bewirkt eine Verschiebung der Sensorpositionen auf den entgegengesetzten Rand des Beaconperimeters. Somit werden sämtliche Sensorknoten an den Rändern des Umringpolygons lokalisiert. Für alle der gewählten Extremkonfigurationen sind die berechneten Positionen unbrauchbar.

Bei der zufälligen Beaconverteilung ergibt sich ein ähnliches Bild. Je nach Verteilung ergibt sich ein anderes optimales Gewicht. In Abbildung 7.11 werden für bestimmte Beaconkonfigurationen optimale Gewichte gesucht. Aus diesem Grund finden sich die Konfigurationen nicht in Tabelle 7.1 wieder.

In Abbildung 7.11 (a) wird das zufällig ausgebrachte Beaconfeld aus den ersten Konfigurationen K_Z^i gewählt. Es ergibt sich ein optimales Gewicht von g = 1,83. Damit ist es in einer vergleichbaren Größenordnung wie Konfiguration K_R^9 mit einem rasterförmig angeordnetem Beaconfeld.

Abbildung 7.11 (b) beschreibt ein Beaconfeld, bei dem alle Sensorknoten außerhalb des Perimeters liegen. Hier wird das Gewicht zu g = 9,99 geschätzt, was gleichbedeutend mit dem Umstand ist, dass kein lokales Minimum gefunden werden konnte. Aufgrund des Schwerpunktverhaltens werden auch alle Sensorpositionen in den Perimeter gezogen. Die Einflüsse der jeweiligen Distanzmessungen und der Gewichtungen sind in diesem Szenario vernachlässigbar, da die Schwerpunktbestimmung überwiegt.

Für den Fall (c) in Abbildung 7.11 werden die Beacons innerhalb des Sensorknotenfeldes platziert. Das dazugehörige optimale Gewicht ergibt sich zu g = 3,78. Im Fall (d) befindet sich das Sensorfeld teilweise innerhalb, teilweise außerhalb des Perimeters mit dem optimalen Gewicht g = 6,29. Weitere Simulationen ergaben eine deutliche Abhängigkeit des zu wählenden Gewichtes von der Netzgeometrie. Liegen die Sensorknoten innerhalb des sie einschließenden Beaconperimeters, ist das Gewicht $g \leq 2$. In den Fällen, in denen das Beaconfeld teilweise oder ganz innerhalb des Sensorfeldes liegt, steigt das optimale Gewicht $g \gtrsim 2$. Befinden sich sämtliche Sensorknoten außerhalb des Perimeters, versagt das gewählte Modell zur Gewichtsschätzung. Das liegt daran, dass das Schwerpunktverhalten des *WCL* in diesem Fall überwiegt und mit keinem Gewicht ausgeglichen werden kann.

7.2.2 DLS-Algorithmus

Der *DLS*–Algorithmus berechnet ein überbestimmtes Streckennetz nach der Methode der kleinsten Quadrate (vgl. Kapitel 3.3) und ist damit den exakten Lokalisierungsverfahren im Sinne der Klassifizierung aus Kapitel 4.1 zuzuordnen. Sein Berechnungsalgorithmus ist so aufgebaut, dass die Gesamtberechnung des Gleichungssystems in einen komplexen Teil, der auf die leistungsstarke Senke ausgelagert werden kann, und in eine einfache Nachberechnung, die auf den ressourcenschwachen Sensorknoten durchgeführt wird, aufgeteilt werden kann.



Abbildung 7.11: Bestimmung optimale Potenz g für WCL bei zufälligen Beaconkonfigurationen.

Der Formelapparat für den *DLS*–Algorithmus soll an dieser Stelle hergeleitet werden [Rei07]. Wie bereits erwähnt, berechnet der Algorithmus die Neupunktposition über einen Bogenschnitt. Damit lässt sich der funktionale Zusammenhang der Koordinaten über die Euklidischen Distanzgleichungen 7.14 herstellen.

$$\hat{d}_i = \sqrt{\left(\hat{x}_N - \tilde{x}_i\right)^2 + \left(\hat{y}_N - \tilde{y}_i\right)^2} \quad (i = 1, 2, ..., n)$$
(7.14)

mit *n* Beacons $\tilde{B}(\tilde{x}_i, \tilde{y}_i)$ und $\hat{P}_N(\hat{x}_N, \hat{y}_N)$ den Koordinaten des Neupunktes. Für eine Berechnung mit der Methode der kleinsten Quadrate gilt es, dieses nichtlineare Gleichungssystem in eine lineare Form zu überführen. Hierfür gibt es in der Literatur einige Verfahren (vgl. Kapitel 3 und 6.1). Für den *DLS*–Algorithmus hingegen wird ein anderes Verfahren benutzt. Durch Quadrieren von Gleichung 7.14 und Subtraktion von Gleichung *j* von den übrigen n - 1 Gleichungen erhält man Gleichung 7.15:

$$\hat{d}_{i}^{2} = \left(\hat{x}_{N} - \tilde{x}_{j} + \tilde{x}_{j} - \tilde{x}_{i}\right)^{2} + \left(\hat{y}_{N} - \tilde{y}_{j} + \tilde{y}_{j} - \tilde{y}_{i}\right)^{2} \quad (i = 1, 2, ..., j - 1, j + 1, ..., n)$$
(7.15)

Der festgehaltene Beacon wird, wie in Kapitel 3.2.2 bereits eingeführt, als Linearisierungswerkzeug bezeichnet. Durch Auflösen der binomischen Formeln, Kürzen und Umstellen der Gleichung 7.15 ergibt sich das nunmehr lineare Gleichungssystem 7.16:

$$(\hat{x}_N - \tilde{x}_j) (\tilde{x}_i - \tilde{x}_j) + (\hat{y}_N - \tilde{y}_i) (\tilde{y}_i - \tilde{y}_j) = \frac{1}{2} \left[(\hat{x}_N - \tilde{x}_j)^2 + (\hat{y}_N - \tilde{y}_j)^2 - \hat{d}_i^2 + (\tilde{x}_i - \tilde{x}_j)^2 + (\tilde{y}_i - \tilde{y}_j)^2 \right] = b_{ij}$$

$$(7.16)$$

Der Übersichtlichkeit wegen wird die Distanz $\tilde{d}_{ij} = \sqrt{(\tilde{x}_i - \tilde{x}_j)^2 + (\tilde{y}_i - \tilde{y}_j)^2}$ zwischen Beacon *j* und den übrigen n - 1 Beacons zu Vektor d zusammengefasst. Daraus ergeben sich die Beobachtungsgleichungen mit Beacon $\tilde{B}_1(\tilde{x}_1, \tilde{y}_1)$ als Linearisierungsbeacon, die für die Ausgleichung genutzt werden, zu Gleichung 7.17:

$$b_{i} = (\hat{x}_{N} - \tilde{x}_{1}) (\tilde{x}_{i} - \tilde{x}_{1}) + (\hat{y}_{N} - \tilde{y}_{1}) (\tilde{y}_{i} - \tilde{y}_{1}) = \frac{1}{2} \left[d_{1}^{2} - d_{i}^{2} + d_{i1}^{2} \right] \quad (i = 2, 3, ..., n)$$
(7.17)

Oder in Matrizenschreibweise der bekannten Form Ax = b mit

$$\mathbf{A} = \begin{pmatrix} \tilde{x}_2 - \tilde{x}_1 & \tilde{y}_2 - \tilde{y}_1 \\ \vdots & \vdots \\ \tilde{x}_n - \tilde{x}_1 & \tilde{y}_n - \tilde{y}_1 \end{pmatrix}, \quad \mathbf{\Delta}\mathbf{x} = \begin{pmatrix} \hat{x}_N - \tilde{x}_1 = \Delta \hat{x} \\ \hat{y}_N - \tilde{y}_1 = \Delta \hat{y} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_{21} \\ \vdots \\ b_{n1} \end{pmatrix}$$
(7.18)

mit (i = 2, ..., n). Dieses Gleichungssystem kann jetzt nach der Methode der kleinsten Quadrate gelöst werden:

$$\mathbf{A}^{T}\mathbf{A}\Delta\mathbf{x} = \mathbf{A}^{T}\mathbf{b}$$

$$\Delta\mathbf{x} = (\mathbf{A}^{T}\mathbf{A})^{-1}\mathbf{A}^{T}\mathbf{b}$$
(7.19)

Da die Koeffizientenmatrix A ausschließlich aus bekannten Beaconkoordinaten besteht, kann mit der Aufteilung

$$\mathbf{A}_{\mathbf{p}} = (\mathbf{A}^{T}\mathbf{A})^{-1}\mathbf{A}^{T}$$

$$\Delta \mathbf{x} = \mathbf{A}_{\mathbf{p}}\mathbf{b}$$

(7.20)

die Berechnung in einen komplexen Teil A_p , der auf einem leistungsstarken Beacon/einer Senke ausgeführt wird, und einen einfachen Teil $\Delta x = A_p b$, der auf jedem Sensorknoten berechnet wird, unterteilt werden. Bis auf einen letzten Schritt $\hat{x}_N = \Delta x + \tilde{x}_1$ und $\hat{y}_N = \Delta y + \tilde{y}_1$ für die Berechnung der endgültigen Koordinaten $\hat{P}_N(\hat{x}_N, \hat{y}_N)$ verlaufen *DLS* und *RAL* nach der Linearisierung identisch.

7.2.2.1 Betrachtungen zum stochastischen Modell für den DLS-Algorithmus

Das stochastische Modell für den *DLS*–Algorithmus wird im Wesentlichen mit der Kovarianzmatrix für die Beobachtungsgleichungen b_{ij} beschrieben. Da in jeder Beobachtungsgleichung die Koordinaten eines

Linearisierungswerkzeugs verwendet werden, sind alle Beobachtungsgleichungen 7.17 miteinander korreliert. Für die Kovarianzmatrix ergibt sich damit die folgende Form:

$$\mathbf{C}_{\mathbf{FF}} = \begin{pmatrix} \sigma_{b_{1j}}^2 & \cdots & \operatorname{cov}(b_{1j}, b_{ij}) & \cdots & \operatorname{cov}(b_{1j}, b_{nj}) \\ \vdots & \ddots & \vdots & & \vdots \\ \operatorname{cov}(b_{1j}, b_{ij}) & \cdots & \sigma_{b_{ij}}^2 & \cdots & \operatorname{cov}(b_{ij}, b_{nj}) \\ \vdots & & \vdots & \ddots & \vdots \\ \operatorname{cov}(b_{1j}, b_{nj}) & \cdots & \operatorname{cov}(b_{ij}, b_{nj}) & \cdots & \sigma_{b_{nj}}^2 \end{pmatrix}$$
(7.21)

mit den Standardabweichungen der Beobachtungen σ_b und den Kovarianzen zwischen den Beobachtungen $cov(b_{ij}, b_{nj})$. Die Kovarianzmatrix wird mit der Gleichung 7.10 nach dem Varianzfortpflanzungsgesetz gebildet. Dafür werden die Beobachtungsgleichungen b_{ij} auf nicht korrelierte Beobachtungen zurückgeführt. Für die Funktionalmatrix F ergeben sich hierbei die Ableitungen der Beobachtungsgleichungen nach den unkorrelierten Größen. Die Distanzen zwischen dem Neupunkt und den Festpunkten werden als unkorreliert mit eigenem mittleren Fehler aufgenommen. Es ergibt sich der funktionale Zusammenhang nach Gleichung 7.22.

$$b_{ij} = \frac{1}{2} \left[\hat{d}_j^2 - \hat{d}_i^2 + \hat{d}_{ij}^2 \right] = \frac{1}{2} \left[\hat{d}_j^2 - \hat{d}_i^2 + (\tilde{x}_i - \tilde{x}_j)^2 + (\tilde{y}_i - \tilde{y}_j)^2 \right]$$
(7.22)

Sollen die Beaconpositionen einen Fehler aus einer vorangegangenen Positionsbestimmung erhalten, gehen sie nach dem Prinzip der "weichen Lagerung" als unkorrelierte Beobachtungen ebenfalls in die Ausgleichung ein. Für die Varianzfortpflanzung ergibt sich folgende Funktionalmatrix F.

		\hat{d}_1	•••	\hat{d}_i	•••	\hat{d}_n	\hat{x}_1	\hat{y}_1	• • •	\hat{x}_i	\hat{y}_i		\hat{x}_n	\hat{y}_n	
	b_{1j}	$\frac{\partial b_{1j}}{\partial \hat{d}_1}$	•••	$\frac{\partial b_{1j}}{\partial \hat{d}_i}$	•••	$\frac{\partial b_{1j}}{\partial \hat{d}_n}$	$rac{\partial b_{1j}}{\partial \hat{x}_1}$	$\tfrac{\partial b_{1j}}{\partial \hat{y}_1}$	•••	$rac{\partial b_{1j}}{\partial \hat{x}_i}$	$\frac{\partial b_{1j}}{\partial \hat{y}_i}$	•••	$rac{\partial b_{1j}}{\partial \hat{x}_n}$	$\frac{\partial b_{1j}}{\partial \hat{y}_n}$	
-	÷	÷		÷		÷	÷	÷		÷	÷		÷	÷	(7.00)
F =	b _{ij}	$rac{\partial b_{ij}}{\partial \hat{d}_1}$		$rac{\partial b_{ij}}{\partial \hat{d}_i}$		$rac{\partial b_{ij}}{\partial \hat{d}_n}$	$rac{\partial b_{ij}}{\partial \hat{x}_1}$	$rac{\partial b_{ij}}{\partial \hat{y}_1}$		$rac{\partial b_{ij}}{\partial \hat{x}_i}$	$rac{\partial b_{ij}}{\partial \hat{y}_i}$		$rac{\partial b_{ij}}{\partial \hat{x}_n}$	$rac{\partial b_{ij}}{\partial \hat{y}_n}$	(7.23)
	÷	÷		÷		÷	÷	÷		:	÷		÷	÷	
	b _{nj}	$rac{\partial b_{nj}}{\partial \hat{d}_1}$		$rac{\partial b_{nj}}{\partial \hat{d}_i}$		$rac{\partial b_{nj}}{\partial \hat{d}_n}$	$rac{\partial b_{nj}}{\partial \hat{x}_1}$	$rac{\partial b_{nj}}{\partial \hat{y}_1}$		$rac{\partial b_{nj}}{\partial \hat{x}_i}$	$rac{\partial b_{nj}}{\partial \hat{y}_i}$		$rac{\partial b_{nj}}{\partial \hat{x}_n}$	$rac{\partial b_{nj}}{\partial \hat{y}_n}$	

Mit den quadratischen mittleren Fehlern der unkorrelierten Beobachtungen auf der Hauptdiagonalen ergibt sich Matrix C_{11} .



Mit diesen Matrizen ergibt die Varianzfortpflanzung für den Beobachtungsvektor **b** die für das stochastische Modell nötige Kovarianzmatrix der Unbekannten C_{FF} nach Gleichung 7.25.

$$\mathbf{C}_{\mathbf{F}\mathbf{F}} = \mathbf{F}\mathbf{C}_{\mathbf{I}\mathbf{I}}\mathbf{F}^{T} = \begin{pmatrix} (\hat{d}_{1}\sigma_{\hat{d}_{1}})^{2} + (\hat{d}_{j}\sigma_{\hat{d}_{j}})^{2} & \cdots & (\hat{d}_{j}\sigma_{\hat{d}_{j}})^{2} & \cdots & (\hat{d}_{j}\sigma_{\hat{d}_{j}})^{2} \\ \vdots & \ddots & \vdots & & \vdots \\ (\hat{d}_{j}\sigma_{\hat{d}_{j}})^{2} & \cdots & (\hat{d}_{i}\sigma_{\hat{d}_{i}})^{2} + (\hat{d}_{j}\sigma_{\hat{d}_{j}})^{2} & \cdots & (\hat{d}_{j}\sigma_{\hat{d}_{j}})^{2} \\ \vdots & & \vdots & \ddots & \vdots \\ (\hat{d}_{j}\sigma_{\hat{d}_{j}})^{2} & \cdots & (\hat{d}_{j}\sigma_{\hat{d}_{j}})^{2} & \cdots & (\hat{d}_{n-1}\sigma_{\hat{d}_{n-1}})^{2} + (\hat{d}_{j}\sigma_{\hat{d}_{j}})^{2} \end{pmatrix}$$
(7.25)

Diese Kovarianz der Unbekannten fließt in den Ausgleichungsprozess ein. Das Ergebnis ist der Differenzvektor aus den Koordinaten des Unbekannten und des Linearisierungsbeacons (Gleichung 7.26).

$$\begin{pmatrix} \Delta \hat{x} \\ \Delta \hat{y} \end{pmatrix} = \begin{pmatrix} \hat{x}_N - \hat{x}_j \\ \hat{y}_N - \hat{y}_j \end{pmatrix} = (\mathbf{A}^T \mathbf{C}_{\mathbf{F}\mathbf{F}}^{-1} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{C}_{\mathbf{F}\mathbf{F}}^{-1} \mathbf{b}$$
(7.26)

Die Position des Unbekannten errechnet sich abschließend zu

$$\begin{pmatrix} \hat{x}_N \\ \hat{y}_N \end{pmatrix} = \begin{pmatrix} \Delta \hat{x} + \hat{x}_j \\ \Delta \hat{y} + \hat{y}_j \end{pmatrix}.$$
(7.27)

7.2.2.2 Simulationsaufbau

Für die nachfolgenden Untersuchungen wird ein $100 \ m \times 100 \ m$ großes Sensorfeld definiert, auf dem 500 Sensorknoten mit einem Beaconanteil von 5 % zufällig verteilt werden. In den unterschiedlichen Analysen steigen die Varianzen der Distanzen und/oder der Beaconpositionen von 0 bis 100 %. Die Analysen werden dabei 10.000 Mal wiederholt, um empirische Effekte zu minimieren. Es wird vorausgesetzt, dass jeder Sensorknoten eine Funkverbindung zu jedem Beacon herstellen kann, d. h., alle Beacons für die Positionsberechnung für jeden unbekannten Knoten genutzt werden können.

Einfluss bei gleichzeitigem Auftreten von Beaconkoordinaten- und Distanzbeobachtungsfehlern



Das Testfeld ist in Abbildung 7.12 (a) und in (b) der durchschnittliche Positionsfehler aller Sensorknoten über der steigenden Varianz der Beaconkoordinaten und der Strecken aufgezeichnet. Die Beacons

Abbildung 7.12: (a) DLS-Testfeld und (b) Lokalisierungsfehler über steigender Varianz der Strecken und Beaconpositionen.

sind als grüne Dreiecke, die Sensorknoten als schwarze Punkte dargestellt. Der für *DLS* notwendige Linearisierungsbeacon ist rot eingefärbt. Dessen Einfluss wird an späterer Stelle gesondert untersucht. Wie aus Abbildung 7.12 (b) ersichtlich ist, haben Distanz- und Beaconpositionsfehler einen großen Einfluss auf die Positionsgenauigkeit der Sensorknoten. Bei einer maximalen Varianz der Streckenund Beaconkoordinaten liegt der Positionsfehler bei 11 %.

Einfluss von Beaconkoordinaten- oder Distanzbeobachtungsfehlern

Werden Distanz- und Beaconpositionsfehler getrennt voneinander behandelt, stellt sich heraus, dass der Fehler in den Beaconpositionen einen größeren Einfluss auf die Genauigkeit der Positionsbestimmung ausübt als fehlerhafte Distanzmessungen. Abbildung 7.13 (a) stellt die Ergebnisse in fehlerbehafteten Strecken und (b) fehlerbehafteten Beaconpositionen gegenüber. Der Verlauf zeigt ein ähnliches Verhalten wie der *RAL*–Algorithmus in Kapitel 6.3.3. Der Lokalisierungsfehler steigt mit zunehmender Streckenvarianz an. Der Einfluss der Genauigkeit der Beaconpositionen ist auch hier größer als der der Strecken. Es ist jedoch davon auszugehen, dass die Streckenmessgenauigkeit in drahtlosen Sensornetzwerken geringer ist als die der Beaconkoordinaten [Rei07].



Abbildung 7.13: (a) Lokalisierungsfehler für DLS über steigender Varianz der Strecken und (b) der Beaconpositionen.

7.2.3 RAL-Algorithmus

Da die mathematische Herleitung der *RAL*–Grundlagen bereits in Kapitel 6.1.1 detailliert beschrieben wurde, soll an dieser Stelle darauf verzichtet und stattdessen gleich zum stochastischen Modell übergegangen werden. Die Beobachtungsgleichungen für den *RAL* ergeben sich nach Gleichung 6.3 zu:

$$-2\hat{x}_i\hat{x}_N - 2\hat{y}_i\hat{y}_N + \hat{w} = \hat{d}_i^2 - K_i = b_i \quad (i = 1, 2, ..., n)$$
(7.28)

mit der Hilfsunbekannten $\hat{w} = (\hat{x}_N^2 + \hat{y}_N^2)$ sowie der Substitution von $(\hat{x}_i^2 + \hat{y}_i^2) = K_i$.

Das stochastische Modell für den *RAL*–Algorithmus wird ebenfalls mit der Kovarianzmatrix für die Beobachtungsgleichungen b_i beschrieben. Die Beobachtungsgleichungen b_i werden auf nicht korrelierte Beobachtungen zurückgeführt. In der Matrix **F** stehen hierbei die Ableitungen der Beobachtungsgleichungen nach den jeweils unkorrelierten Größen:

		\hat{d}_1	 \hat{d}_n	\hat{x}_1	\hat{y}_1	 \hat{x}_n	\hat{y}_n
F =	b_1	$rac{\partial b_1}{\partial \hat{d}_1}$	 $rac{\partial b_1}{\partial \hat{d_i}}$	$rac{\partial b_1}{\partial \hat{x}_1}$	$rac{\partial b_1}{\partial \hat{y}_1}$	 $rac{\partial b_1}{\partial \hat{x}_i}$	$\frac{\partial b_1}{\partial \hat{y_i}}$
1 -	÷	:	÷	÷	÷	÷	÷
	b_n	$\frac{\partial b_n}{\partial \hat{d}_1}$	 $rac{\partial b_n}{\partial \hat{d}_n}$	$rac{\partial b_n}{\partial \hat{x}_1}$	$rac{\partial b_n}{\partial \hat{y}_1}$	 $\frac{\partial b_n}{\partial \hat{x}_n}$	$\frac{\partial b_n}{\partial \hat{y}_n}$

Die Entfernungen zwischen dem Neupunkt und den Festpunkten können als unkorreliert mit eigenem mittleren Fehler angesehen werden. Die sich ergebende Kovarianzmatrix C_{FF} hat die Form 7.25 und ist in diesem Fall, da keine Korrelationen zwischen den Beobachtungen vorliegen, eine Diagonalmatrix:

$$\mathbf{C}_{\mathbf{F}\mathbf{F}} = \mathbf{F}\mathbf{C}_{\mathbf{l}\mathbf{l}}\mathbf{F}^{T} = \begin{pmatrix} (2\hat{d}_{1} \cdot \sigma_{\hat{d}_{1}})^{2} & 0 & \cdots & 0 \\ 0 & (2\hat{d}_{2} \cdot \sigma_{\hat{d}_{2}})^{2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & (2\hat{d}_{n} \cdot \sigma_{\hat{d}_{n}})^{2} \end{pmatrix}.$$
 (7.30)

Nach vollständiger Berechnung durch RAL erhält man also:

$$\hat{\mathbf{x}} = \begin{pmatrix} \hat{x}_N \\ \hat{y}_N \\ \hat{w} \end{pmatrix} = (\mathbf{A_1}^T \mathbf{C_{FF}}^{-1} \mathbf{A_1})^{-1} \mathbf{A_1}^T \mathbf{C_{FF}}^{-1} \mathbf{b}$$
(7.31)

7.2.3.1 Untersuchung zum vereinfachten RAL-Algorithmus

Berechnungsaufwand bei der Eliminierung einzelner Unbekannter

In der bisherigen Beschreibung von *RAL* wurde nach der Vorberechnung die dritte Zeile in der entstandenen $(3 \times n)$ - Matrix A gestrichen, um den Berechnungsaufwand zu reduzieren. Wie im Abschnitt 6.2.1 ausgeführt wurde, bringt diese Vereinfachung den Vorteil mit sich, dass weniger Berechnungen auf den Sensorknoten ausgeführt und weniger Daten übermittelt werden müssen. Diese Vereinfachung ist mathematisch streng genommen nicht korrekt. Aus diesem Grund soll an dieser Stelle der Berechnungsaufwand mit der in Kapitel 3.6.6 beschriebenen Eliminierung einzelner Unbekannter betrachtet werden.

Für die Verbesserungsgleichungen gilt bei *RAL* nach Gleichung 7.32:

$$\mathbf{v} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 \end{pmatrix} \begin{pmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \end{pmatrix} - \mathbf{b}$$
(7.32)

mit $\hat{\mathbf{x}}_1 = \begin{pmatrix} \hat{x}_N \\ \hat{y}_N \end{pmatrix}$ und $\hat{\mathbf{x}}_2 = \hat{w}$. Da nur eine lineare Variable zu entfernen ist, wird die Submatrix \mathbf{A}_2 zum

Vektor mit Einselementen, was die Berechnungskomplexität reduziert. Die $(n \times 2)$ große Submatrix A₁ ergibt sich zu:

$$\mathbf{A_1} = \begin{pmatrix} -2\hat{x}_1 & -2\hat{y}_1 \\ -2\hat{x}_2 & -2\hat{y}_2 \\ \vdots & \vdots \\ -2\hat{x}_n & -2\hat{y}_n \end{pmatrix}.$$
 (7.33)

Für die Berechnung auf der Senke ergeben sich folgende Schritte:

1. Die Berechnung der Submatrix N_{11}^{-1} .
2. Die Multiplikation der Submatrix N_{11}^{-1} mit N_{12} und N_{22}^{-1} .

Diese Submatrizen, die Matrix A_1 und der Vektor K müssen zu den Sensorknoten verschickt werden, auf denen die Nachberechnung durchgeführt wird. Dafür ergeben sich folgende Operationen:

- 1. Aufstellen des Vektors b erfordert nach Abschnitt 6.2.1.2 2n Flops.
- 2. Berechnung des Vektors n_1 durch Multiplikation der $(2 \times n)$ -Matrix A_1 mit dem Vektor b benötigt 4n 2 *Flops*.
- 3. Berechnung des Skalars n_2 durch Aufaddieren der Elemente im **b**–Vektor erfordert n 1 Flops.
- 4. Multiplikation der inversen Normalgleichungsmatrix \check{N}_{II}^{-1} mit dem Vektor $\mathbf{n_1}$ benötigt 5 *Flops*.
- 5. Multiplikation des (2×1) großen Normalgleichungsvektors N₂₂ mit dem Skalar n_2 verlangt 2 *Flops*.
- 6. Abschließend muss die Subtraktion $\hat{\mathbf{n}}_1 \hat{\mathbf{n}}_2$ mit ebenfalls 2 *Flops* durchgeführt werden.

Die Berechnung mittels Blockzerlegung erfordert zusammen 7n - 3 *Flops*. Damit wären bei der Berechnung durch dieses Verfahren n - 1 mehr Schritte durchzuführen als bei dem in Abschnitt 6.2.1.2 formulierten Algorithmus.

Für die Kommunikation ergeben sich folgende Zahlen:

- 1. Versenden des Vektors K beinhaltet n Elemente.
- 2. Submatrix A_1^T hat 2n Elemente.
- 3. Die Matrix \breve{N}_{11}^{-1} mit jeweils 3 und der Vektor N_{22} mit 2 Elementen.

Damit müssen 3n + 5 Elemente über das Netzwerk versendet werden. Bei 100 Beacons und 4 *Byte* pro Element ergibt sich somit ein Kommunikationsaufkommen von 1.220 *Bytes* und damit lediglich 20 *Bytes* mehr als für den ursprünglichen vereinfachten *RAL*.

Berücksichtigung von Abhängigkeiten zwischen den Unbekannten

In der bisherigen Beschreibung des *RAL* wurde vorausgesetzt, dass \hat{x}_N , \hat{y}_N und \hat{w} unabhängig voneinander sind. Durch die geltende Beziehung $\hat{w} = \hat{x}_N^2 + \hat{y}_N^2$ ist diese Voraussetzung jedoch nicht gegeben. Für eine präzise Berechnung muss die Abhängigkeit der Unbekannten jedoch berücksichtigt werden. Diese kann nach der Positionsbestimmung auf jedem einzelnen Sensorknoten durch eine weitere Ausgleichung aufgelöst werden. Dafür wird eine bedingte Ausgleichung nach Kapitel 3.6.4 angewendet. Bei der bedingten Ausgleichung bewegt man sich im Bereich der Beobachtungen. Ausgehend von der Positionsbestimmung durch *RAL* werden die berechneten Unbekannten \hat{x}_N , \hat{y}_N und \hat{w} als Beobachtungen eingeführt. Als einzige Bedingungsgleichung erhält man dann auf Grundlage des Zusammenhangs $\hat{w} = \hat{x}_N^2 + \hat{y}_N^2$

$$f(l) = \hat{x}_N^2 + \hat{y}_N^2 - \hat{w} = 0.$$
(7.34)

Die Jakobimatrix **B** enthält die partiellen Ableitungen $\frac{\partial f}{\partial \hat{x}_N}$, $\frac{\partial f}{\partial \hat{y}_N}$ und $\frac{\partial f}{\partial \hat{w}}$

$$\mathbf{B}^{T} = \begin{bmatrix} \frac{\partial f}{\partial \hat{x}_{N}} & \frac{\partial f}{\partial \hat{y}_{N}} & \frac{\partial f}{\partial \hat{w}} \end{bmatrix} = \begin{bmatrix} 2\hat{x}_{N} & 2\hat{y}_{N} & -1 \end{bmatrix}.$$
(7.35)

Die Normalgleichungsmatrix errechnet sich mit der Kofaktormatrix Q_{xx} aus der vermittelnden Ausgleichung zu

$$\mathbf{N} = \mathbf{B}^T \mathbf{Q}_{\mathbf{x}\mathbf{x}} \mathbf{B}. \tag{7.36}$$

Da es sich bei **B** um einen $[3 \times 1]$ -Vektor handelt, wird die Normalgleichungsmatrix zu einem Skalar. Somit wird der Widerspruchsvektor ebenfalls zu einem Skalar

$$w = 0 - (\hat{x}_N^2 + \hat{y}_N^2 - \hat{w}). \tag{7.37}$$

Selbiges trifft für den Korrelatenvektor nach Gleichung 3.66 zu. Die Verbesserungen der Beobachtungen als $[3 \times 1]$ -Vektor errechnen sich mit

$$\mathbf{v} = \mathbf{Q}_{\mathbf{x}\mathbf{x}}\mathbf{B}k \tag{7.38}$$

und die ausgeglichenen Bebachtungen

$$\begin{aligned} x_N &= \hat{x}_N + v_x \\ y_N &= \hat{y}_N + v_y. \end{aligned} \tag{7.39}$$

Um den Berechnungsaufwand der nachträglichen bedingten Ausgleichung abzuschätzen, sind folgende Punkte zu berücksichtigen:

- Da für die Berechnung die Kofaktormatrix der Unbekannten Q_{xx} sowie ein Näherungwert für \hat{w} aus der vermittelnden Ausgleichung benötigt wird, ist das Weglassen der dritten Zeile nach Kapitel 6.2.1.2 nicht möglich. Darüber hinaus ist die Vorberechnung $(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ nicht möglich, da die Normalgleichungsmatrix $(\mathbf{A}^T \mathbf{A})^{-1}$ separat benötigt wird, was sowohl den Berechnungsaufwand auf den Sensorknoten als auch den Kommunikationsaufwand innerhalb des Sensornetzwerkes belastet.
- Wird ŵ im Vorfeld nach Kapitel 3.6.6 eliminert, muss zusätzlich die [2 × 1] große Matrix N₂₁ versendet und ŵ nach Gleichung 3.78 nachträglich berechnet werden.

Da der Aufwand nach Punkt 2 geringer ist, soll dieser genauer betrachtet werden. Für die Berechnung von $\hat{w} = N_{22}^{-1}n_2 - N_{22}^{-1}N_{21}x_1$ fallen 10 Multiplikationen und 4 Additionen an. Somit erhöht sich der im letzten Abschnitt ermittelte Aufwand auf 7n + 11 *Flops*. Die bedingte Ausgleichung erfordert folgende Rechenoperationen:

- Die Berechnung der Normalgleichungsmatrix $\mathbf{N} = \mathbf{B}^T \mathbf{Q}_{\mathbf{x}\mathbf{x}} \mathbf{B}$ erfordert 20 *Flops*.
- Das Aufstellen des Widerspruchsvektors erfordert 5 Flops.
- Die Berechnung des Korrelatenvektors erfordert 2 Flops.
- Die Berechnung der Verbesserungen der Beobachtungen \hat{x}_N und \hat{y}_N erfordert 20 *Flops*.
- Die abschließende Berechnung der ausgeglichenen Koordinaten benötigt weitere 2 Flops.

Damit ergibt sich ein Gesamtberechnungsaufwand für die nachträgliche bedingte Ausgleichung von 49 *Flops*. Mit der zusätzlichen Berechnung von \hat{w} ergibt sich ein Gesamtaufwand von 7n + 60 *Flops* auf jedem einzelnen Sensorknoten. Für die Kommunikation fallen nochmals 2 weitere Elemente also 8 *Bytes* für die Matrix N₂₁ an.

Als eine weitere Möglichkeit der Berücksichtigung der Abhängigkeit der Unbekannten ist eine vermittelnde Ausgleichung mit Bedingungen zwischen den Unbekannten denkbar. Damit ist eine Berechnung mit nur einer Ausgleichung möglich, indem Gleichung 7.34 als Bedingungsgleichung in das Anfangsmodell mit aufgenommen wird. Da hierfür im Vorfeld Näherungswerte benötigt werden, die in diesem Fall nicht vorliegen, wird auf eine detaillierte Betrachtung verzichtet. Abschließend ist zu sagen, dass der *RAL*–Algorithmus mit Eliminierung der Unbekannten \hat{w} durch Blockzerlegung sowie mit nachträglicher Beachtung der Abhängigkeiten von \hat{x}_N , \hat{y}_N in \hat{w} eine durchaus performante Technik zur Lokalisierung darstellt. Fällt der Mehraufwand für die Berechnung und Kommunikation nicht zu sehr ins Gewicht, ist diese Technik dem ursprünglichen Algorithmus aus Abschnitt 6.2.1.2 vorzuziehen. In den weiteren Betrachtungen soll jedoch das Modell mit minimalstem Berechnungs- und Kommunikationsaufwand betrachtet werden.

7.2.3.2 Einfluss von fehlerbehafteten Eingangsdaten

In einer weiteren Simulation soll der Einfluss fehlerbehafteter Eingangsdaten betrachtet werden. Darunter fallen zum einen die Distanzmessungen und zum anderen die Beaconkoordinaten. Da die Beacons über ein beliebiges Lokalisierungssystem, wie z. B. GNSS, positioniert werden, sind deren ermittelte Koordinaten ebenfalls Ergebnisse fehlerbehafteter Messungen. Die Distanz- und Beaconkoordinatenfehler werden dafür auf Basis der Ausführungen in Abschnitt 7.2 schrittweise erhöht und der durchschnittliche Lokalisierungsfehler ermittelt. Der Simulationsaufbau ist folgendermaßen gewählt: die Sensorfeldfläche beträgt $(100 \ m \times 100 \ m)$ mit 500 Sensorknoten, von denen 5 % Beacons sind. Jede Simulation wird 10.000 Mal wiederholt, um empirische Einflüsse auszuschließen. Die Ergebnisse sind in Abbildung 7.14 wiedergegeben. Der Lokalisierungsfehler steigt mit sich erhöhender Varianz an (Abbildung 7.14 (a)). Der flache Verlauf weist auf eine hohe Toleranz von *RAL* gegenüber Distanzfehlern hin.

In Abbildung 7.14 (b) ist der Lokalisierungsfehler über die steigende Varianz der Beaconpositionen illustriert. Der Fehler steigt schneller als bei fehlerhaften Distanzwerten.

Abschließend wird der Lokalisierungsfehler betrachtet, wenn sämtliche Eingangsdaten (Distanzen und Beaconpositionen) fehlerbehaftet sind (Abbildung 7.14 (c)). Der Anstieg des Fehler erfolgt in selbem Maße wie in den Abbildungen 7.14 (a) und (b). Derart große Fehler der Beaconpositionen sind jedoch nicht zu erwarten, da die Beacons entweder manuell sehr genau platziert oder durch ein hochgenaues Ortungssystem eingemessen werden können. Aus diesem Grund wird in den nachfolgenden Untersuchungen die Position der Beacons als ebenfalls fehlerfrei angenommen.

7.2.4 Standardlokalisierung

Das mathematische Modell der Standardlokalisierung basiert wie *DLS* und *RAL* auf der Lösung des nichtlinearen Gleichungssystems 7.14 mit der Methode der kleinsten Quadrate. Die Linearisierung erfolgt hier allerdings über eine Taylorreihenentwicklung (vgl. Kapitel 3.2.1). Die Lösung ergibt sich nach $\mathbf{b} = \mathbf{A}\mathbf{x}$. Durch Anwendung der Taylorlinearisierung finden sich in der Koeffizientenmatrix die Ableitungen der Beobachtungsgleichungen nach den Unbekannten wieder [Grü03]:

$$\mathbf{A} = \begin{bmatrix} \frac{\partial \hat{d}_1}{\partial \hat{x}_N} & \frac{\partial \hat{d}_1}{\partial \hat{y}_N} \\ \vdots & \vdots \\ \frac{\partial \hat{d}_i}{\partial \hat{x}_N} & \frac{\partial \hat{d}_i}{\partial \hat{y}_N} \\ \vdots & \vdots \\ \frac{\partial \hat{d}_n}{\partial \hat{x}_N} & \frac{\partial \hat{d}_n}{\partial \hat{y}_N} \end{bmatrix}.$$

(7.40)



Abbildung 7.14: Einfluss von fehlerbehafteten (a) Distanzen und (b) Beaconpositionen und (c) Distanzen und Beaconpositionen auf RAL.

Da für die Linearisierung ein Startwert $\bar{P}_0(\bar{x}_0, \bar{y}_0)$ benötigt wird, ergibt sich der *b*-Vektor zu:

$$\mathbf{b} = \begin{pmatrix} \hat{d}_1 - \bar{d}_1 \\ \vdots \\ \hat{d}_i - \bar{d}_i \\ \vdots \\ \hat{d}_n - \bar{d}_n \end{pmatrix}, \qquad (7.41)$$

mit \hat{d} den gemessenen und den berechneten Distanzen \bar{d} aus dem Näherungswert und den Beaconkoordinaten. Somit ergibt sich für die Berechnung:

$$\Delta \mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T (\hat{\mathbf{d}} - \bar{\mathbf{d}}).$$
(7.42)

Da ein Näherungswert eingeführt wurde, müssen für die errechnete Position abschließend die Startwertkoordinaten zur Lösung des Gleichungssystems hinzuaddiert werden:

$$P_{N} = \begin{pmatrix} x_{N} \\ y_{N} \end{pmatrix} = \Delta \mathbf{x} + \begin{pmatrix} \bar{x}_{0} \\ \bar{y}_{0} \end{pmatrix} = \begin{pmatrix} \Delta x + \bar{x}_{0} \\ \Delta y + \bar{y}_{0} \end{pmatrix}.$$
 (7.43)

Im stochastischen Modell der SL ergibt sich für die unkorrelierte Kovarianzmatrix CFF:

$$\mathbf{C}_{\mathbf{FF}} = \begin{pmatrix} \sigma_{\hat{d}_1}^2 & 0 \\ & \sigma_{\hat{d}_1}^2 & \\ & & \sigma_{\hat{d}_1}^2 \\ & & & \ddots \\ & & & & \\ 0 & & & \sigma_{\hat{d}_n}^2 \end{pmatrix}$$
(7.44)

und die abschließende Berechnung unter Einbeziehung des stochastischen Modells zu:

$$\Delta \mathbf{x} = (\mathbf{A}^T \mathbf{C}_{\mathbf{F}\mathbf{F}}^{-1} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{C}_{\mathbf{F}\mathbf{F}}^{-1} (\hat{\mathbf{d}} - \bar{\mathbf{d}})$$
(7.45)

sowie für die Koordinaten des Neupunktes nach Gleichung 7.42.

Effekte bei der Wahl des stochastischen Modells für DLS und RAL

Wie in den vorigen Absätzen beschrieben, werden bei den Lokalisierungsalgorithmen *DLS* und *RAL* die vereinfachten stochastischen Modelle verwendet, d. h. die Kovarianzmatrix C_{FF} wird zur Einheitsmatrix E. Das geschieht aus dem Grund, als dass die anfänglich unbekannten Distanzen zwischen Sensorknoten und Beacons für die Berechnung der Kovarianzen benötigt werden. Da die Distanzen für die Vorberechnung jedoch nicht vorliegen, wäre eine verteilte Berechnung, wie in Kapitel 6.1.4 formuliert, nicht möglich. Das Gleichsetzen der Kovarianzmatrix mit der Einheitsmatrix hat jedoch einige Nebeneffekte. Für die Varianzen, den Hauptdiagonalelementen, bei *RAL* gilt nach Gleichung 7.30:

$$\sigma_{b_i}^2 = (2\hat{d}_i \cdot \sigma_{\hat{d}_i})^2 \qquad (i = 1, 2, ..., n)$$
(7.46)

Die Varianzen werden in dem vereinfachten Modell zu Eins. Das heißt, alle Hauptdiagonalelemente werden Eins gesetzt:

$$\sigma_{b_i}^2 = (2\hat{d}_i \cdot \sigma_{\hat{d}_i})^2 = 1 \qquad (i = 1, 2, ..., n)$$
(7.47)

und weiter:

$$(2\hat{d}_1 \cdot \sigma_{\hat{d}_1})^2 = (2\hat{d}_2 \cdot \sigma_{\hat{d}_2})^2 = \dots = (2\hat{d}_n \cdot \sigma_{\hat{d}_n})^2 = 1.$$
(7.48)

Unter der Annahme, dass die Strecke \hat{d}_1 doppelt so lang ist wie die Strecke \hat{d}_2 , also $\hat{d}_1 = 2 \cdot \hat{d}_2$ oder $\frac{1}{2}\hat{d}_1 = \hat{d}_2$, heißt das:

$$(2\hat{d}_1 \cdot \sigma_{\hat{d}_1})^2 = (2 \cdot \frac{1}{2}\hat{d}_1 \cdot \sigma_{\hat{d}_2})^2 = (\hat{d}_1 \cdot \sigma_{\hat{d}_2})^2.$$
(7.49)

Damit ergibt sich für die Streckenvarianz $\sigma_{\hat{d_1}}^2$:

$$\sigma_{\hat{d}_1}^2 = \frac{\hat{d}_1^2}{4 \cdot \hat{d}_1^2} \sigma_{\hat{d}_2}^2$$

$$= \frac{1}{4} \sigma_{\hat{d}_2}^2.$$
(7.50)

Das heißt, die Streckenvarianz der längeren Strecke \hat{d}_1 geht mit nur einem Viertel des Gewichtes der kürzeren Strecke \hat{d}_2 in die Berechnung ein. Das bedeutet, dass in diesem vereinfachten mathematischen Modell die längeren Strecken als genauer beobachtet in die Berechnung eingehen, und zwar mit Faktor 4. Dieses Modell spiegelt nicht die Realität wieder, da längere Distanzen mit einem größeren Fehler behaftet sind als kurze (siehe Abschnitt 4.4.1).

Åhnlich verhält es sich mit dem *DLS*-Algorithmus. Da hier ebenfalls die Einheitsmatrix als Kovarianzmatrix gewählt werden muss, um eine verteilte Berechnung ermöglichen können, heißt das, dass nun zusätzlich die Nebendiagonalelemente, die Korrelationskoeffizienten (Gleichung 7.25), zu Null gesetzt werden. Das heißt $(\hat{d}_n \cdot \sigma_{\hat{d}_n})^2 = 0$ für den Fall, dass der letzte Beacon festgehalten wird. Für die Hauptdiagonalenelemente gilt dann:

$$\sigma_{b_i}^2 = (\hat{d}_i \cdot \sigma_{\hat{d}_i})^2 + (\hat{d}_n \cdot \sigma_{\hat{d}_n})^2 = 1 \qquad (i = 1, 2, ..., n - 1)$$
(7.51)

mit $(\hat{d}_n \cdot \sigma_{\hat{d}_n})^2 = 0$ folgt:

$$\sigma_{b_i}^2 = (\hat{d}_i \cdot \sigma_{\hat{d}_i})^2 = 1.$$
(7.52)

Die weiteren Schritte erfolgen analog zu *RAL* und es ergibt sich nach dem obigen Beispiel für die Streckenvarianz $\sigma_{\hat{d}_1}^2$:

$$\sigma_{\hat{d}_1}^2 = \frac{\hat{d}_1^2}{4 \cdot \hat{d}_1^2} \sigma_{\hat{d}_2}^2$$

$$= \frac{1}{4} \sigma_{\hat{d}_2}^2.$$
(7.53)

Für den *DLS* hat das zufolge, dass die längeren Strecken genauso stark gewichtet werden wie bei *RAL*. Hinzu kommt, dass die Korrelationen mit dem festgehaltenen Beacon nicht beachtet werden beachtet. Wird, wie bereits erwähnt, der letzte Beacon n festgehalten, gilt:

$$(\hat{d}_n \cdot \sigma_{\hat{d}_n})^2 = 0 \tag{7.54}$$

Um diese Bedingung zu erfüllen, muss $\sigma_{\hat{d}_n}^2 = 0$. Mit anderen Worten heißt das, dass die Strecke zu dem festgehaltenen Beacon als fehlerfrei in das vereinfachte Modell eingeht.

Nachteilig auf das vereinfachte Modell scheint sich das Vernachlässigen einiger stochastischer Eigenschaften auszuwirken. Jedoch ist nur mit diesem Ansatz eine verteilte und somit ressourceneffiziente Berechnung möglich. Im folgenden Abschnitt sollen die Auswirkungen dieser Vereinfachungen genauer untersucht werden.

7.3 Vergleich der Algorithmen

Für die Erzeugung der verfälschten Beobachtungen wird an dieser Stelle auf Kapitel 7.2 verwiesen. Um die Distanzmessungen möglichst real zu simulieren, sind alle Strecken um einen zufälligen Betrag verfälscht, der einer Gauß'schen Normalverteilung r_{appr} , $B_{appr} \sim \mathcal{N}(\mu_{exact}, \sigma^2)$ mit Erwartungswert μ der exakten Distanz und der Varianz $\sigma_d^2 = 10$ % unterliegt. Der Varianzbereich einer Distanzbeobachtung erstreckt sich somit in einem Intervall von -10 % bis +10 %. Die Potenz des *WCL*–Algorithmus für die Gewichtung wird für die jeweilige Sensorknotenposition nach Gleichung 7.13 aus Abschnitt 7.2.1.2 berechnet sowie die Beaconkoordinaten als fehlerfrei angenommen.

Im Fokus der Untersuchung steht das Verhalten der unterschiedlichen Lokalisierungsalgorithmen bei Einbeziehen ihrer stochastischen Modelle. Die Algorithmen *RAL* und *DLS* basieren auf dem Ansatz, die Positionsberechnung zwischen den Sensorknoten und den Beacons aufzuteilen. Wie in dem letzten Abschnitt gezeigt wurde, ist diese verteilte Berechnung nach Einbeziehen der strengen stochastischen Modelle nicht ohne Weiteres möglich. Ziel dieses Abschnitts ist die Untersuchung der Algorithmen hinsichtlich ihrer Genauigkeit bei der Positionsbestimmung unter Einbeziehen dieser strengen stochastischen Zusammenhänge sowie unter Verwendung des vereinfachten Ansatzes, in dem die Kovarianzmatrix gleich der Einheitsmatrix gesetzt wird. Erste Untersuchungen dazu sind in Niemeyer et al. [NBB10] zu finden und sollen an dieser Stelle näher erläutert bzw. ausgebaut werden. Die Testfelder geben Abbildungen 7.15 (a) für eine zufällige und (b) für eine kreisförmige Verteilung der



Abbildung 7.15: Testfeld der Simulationen für den Vergleich ausgewählter Lokalisierungsalgorithmen (a) bei zufällig verteilten und (b) kreisförmig angeordneten Beacons.

Beacons wieder. In Tabelle 7.2 sind die Simulationsparameter aufgezeigt. Die Beacons und Sensorknoten werden zufällig und einer Gleichverteilung folgend ausgebracht. Der blaue Rahmen markiert einen Zoom–Ausschnitt, in dem die einzelnen Genauigkeitsanalysen beispielhaft für einen Sensorknoten (grüner Kreis in Abbildung 7.15 (a)) vergrößert dargestellt werden. Es wird angenommen, dass sich die Sensorknoten in Kommunikationsreichweite zu allen Beacons befinden. Der Linearisierungsbeacon für

Parameter	Wert	Parameter	Wert
Sensorfeldfläche [m]	100×100	Sendereichweite Beacons [m]	100
Anzahl Sensorknoten	100	Varianz (σ_d^2) der Distanzen [%]	10
Anzahl Beacons	100	Varianz (σ_b^2) der Beaconpositionen [%]	0
Anzahl der Wiederholungen	10.000		

Tabelle 7.2: Parameter der Simulationen für den Vergleich ausgewählter Lokalisierungsalgorithmen.

DLS ist durch ein blaues Dreieck gekennzeichnet. Um die Richtung zum Linearisierungsbeacon im vergrößerten Ausschnitt sichtbar zu machen, verbindet eine Linie den Beacon mit dem Sensorknoten. Die Berechnungen erfolgen mit 100 zufällig platzierten Sensorknoten.

In den Abbildungen 7.16 (a) und (b) sind die Ergebnisse der Simulationen für ein Szenario zufällig und kreisförmig verteilter Beacons mit vereinfachtem stochastischen Modell wiedergegeben. Die Punktwolken bilden sich sowohl bei *SL*, *RAL* und *DLS* ellipsenförmig ab. In Abbildung 7.16 (a) ist gut zu erkennen, dass der *DLS* in Form einer langgezogenen Ellipse einen größeren Streuungsbereich aufweist, als *SL* und *RAL*. Da in der Positionsschätzung die Korrelation aus Gleichung 7.25 des Unbekannten mit dem Linearisierer nicht berücksichtigt wurde, bildet sich die Punktwolke stark gedehnt entlang der Verbindung Sensorknoten–Linearisierungsbeacon aus. Mit zunehmender Entfernung zum Linearisierungsbeacon vergrößert sich auch die Streuung der berechneten Positionen entlang der Verbindung Sensorknoten–Linearisierer. Der Einfluss des Linearisierers wird im zweiten Versuch gesondert untersucht. Die Punktwolken bei *SL* und *RAL* hingegen sind nahezu kreisförmig, was auf die hohe Beacondichte zurückzuführen ist. Die Beobachtungen sind unkorreliert und gleich genau, sodass alle Beacons unabhängig in die Berechnung eingehen. Lediglich der streckenabhängige Fehleranteil wirkt sich auf die Positionsschätzung aus.

Der approximative *WCL*–Algorithmus erzeugt keine Fehlerellipsen im klassischen Sinn. Vielmehr ist eine Überlagerung von zufälligen und systematischen Fehlereinflüssen zu erkennen. Da die Distanzwerte lediglich als Gewichte eingehen, können die berechneten Punkte systematisch nicht außerhalb des umringenden Polygons der Beacons liegen. Durch die hohe Gewichtung nahe liegender Beacons werden die berechneten Neupunkte zu diesen hingezogen. Die weiter entfernten Festpunkte üben einen geringen Einfluss auf die Positionsschätzung aus, weswegen sich die Positionen entlang der Richtungen zu den nächsten Beacons ausbilden.

Die Dimensionen der Punktwolken von *SL* und *RAL* liegen im selben Genauigkeitsbereich. Daher sind die Fehlerdiagramme in Abbildung 7.16 (c) zu Rate zu ziehen. Die Grafik gibt die mittleren Fehler der Einzelmessungen über die Wiederholungen wieder. Ab 1.000 Wiederholungen fluktuiert der Fehler nicht mehr, d. h., es sind ausreichend Simulationen durchgeführt worden, um empirische Effekte auszuschließen.

Die *RAL*-Positionen sind mit einem Positionsfehler von 0, 93 % ca. 33 % schlechter als die des Referenzalgorithmus *SL*. Bei dem *DLS*–Algorithmus hingegen tritt mit 4, 26 % ein mehr als sechs mal größerer mittlerer Punktfehler auf als bei den anderen beiden exakten Lokalisierungsmethoden. Dies lässt sich im Wesentlichen durch die beschriebene Wahl und die Lage des Linearisierungsbeacons erklären. Der



Abbildung 7.16: Berechnete Sensorpositionen mit vereinfachtem stochastischen Modell (a) bei zufälliger und (b) bei Kreisverteilung der Beacons sowie der Lokalisierungsfehler (c) bei zufälliger und (d) bei Kreisverteilung der Beacons.

WCL–Algorithmus liefert, wie eingangs bereits vermutet, den größten mittleren Punktfehler von 8,74 %, was auf das Schwerpunktverhalten des *WCL* zurückzuführen ist.

In einem zweiten Versuch wird das systematische Verhalten der Algorithmen untersucht (Abbildung 7.16 (b)). Dazu wurden 100 Beacons im Radius von 50 m um den Knotenpunkt kreisförmig angeordnet. Es bestehen Distanzbeobachtungen zwischen dem Knotenpunkt und allen Beacons. Der Varianzbereich

für die Distanzbeobachtungen liegt hier ebenfalls zwischen -10% bis +10%. Die optimale Potenz des *WCL*-Algorithmus wird dynamisch berechnet und für jeden Sensorknoten separat festgelegt.

Durch die kreisförmige Verteilung der Beacons um den Unbekannten haben diese denselben Einfluss auf die Position. Die Punktwolken in Abbildung 7.16 (b) bei *SL* und *RAL* sind demzufolge kreisförmig und der mittlere Punktfehler mit 0, 63 % in Abbildung 7.16 (d) nahezu gleich groß. Da sich der Sensorknoten im Schwerpunkt des Beaconfeldes befindet, ist auch die Punktwolke bei *WCL* kreisförmig. Nach Abbildung 7.16 (d) liegt der mittlere Punktfehler mit 0, 64 % im selben Genauigkeitsbereich wie der bei *SL* und *RAL*. Bei *DLS* wird die Abhängigkeit des Linearisierers auf die Positionsschätzung deutlich. Auch hier ist eine lang gezogene Ellipse entlang der Line des Sensorknotens zum Linearisierer zu beobachten. Der mittlere Punktfehler fällt sogar um den Faktor 3 größer aus als der des approximativen *WCL* und der exakten Algorithmen *SL* und *RAL*. Wenn die Distanzmessungen als fehlerfreie vorausgesetzt werden, ergeben sich für die exakten Lokalisierungsmethoden ebenfalls fehlerfreie Positionen. Der approximative *WCL* jedoch liefert auch bei fehlerfreien Distanzen fehlerhafte Positionen, was in seiner Schwerpunktmethode begründet ist. Befindet er sich jedoch im Schwerpunkt, ergeben sich auch exakte Positionsschätzung bei fehlerfreien Streckenbeobachtungen. Tabelle 7.3 fasst die Simulationsergebnisse in den jeweiligen Testfeldern zusammen.

Algo-	Lokalisierungsfehler [%]					
rithmus	zufällig*	kreisförmig*	zufällig*	kreisförmig*		
	$\sigma_d^2 = 10\%$	$\sigma_d^2 = 10 \%$	$\sigma_d^2 = 0\%$	$\sigma_d^2 = 0\%$		
WCL	8,74	0,64	sys.Fehler	0,0		
SL	0,67	0,63	0,0	0,0		
RAL	0,93	0,63	0,0	0,0		
DLS	4,26	2, 18	0,0	0,0		

Tabelle 7.3: Simulationsergebnisse für den Vergleich ausgewählter Lokalisierungsalgorithmen mit vereinfachtem stochastischen Modell (* Beaconverteilung).

Werden jedoch die strengen stochastischen Modelle mit berücksichtigt, ergeben sich für die feinkörnigen Algorithmen vergleichbare Ergebnisse. In Abbildung 7.17 (a) und (b) sind die Punktwolken für *SL*, *RAL* und *DLS* unter Einbeziehung der strengen stochastischen Modelle und in den Abbildungen 7.18 die dazugehörigen Fehlerellipsen dargestellt, Abbildung 7.17 (c) und (d) stellen die sich ergebenden mittleren Punktfehler dar.

Die Fehlerellipsen sind unter Einbeziehung des strengen stochastischen Modells berechnet worden. Bei der zufälligen Beaconverteilung haben die Fehlerellipsen der drei exakten Lokalisierungsalgorithmen eine vergleichbare Größe und Ausrichtung. Für den *WCL* ist diese lang gezogen mit klarer Richtung zu Gebieten mit hoher Beacondichte, was wiederum durch das Schwerpunktverhalten erklärt werden kann. Wird die Kreisverteilung der Beacons zugrunde gelegt, sind die Fehlerellipsen aller Lokalisierungsalgorithmen gleich groß.

Da die Abhängigkeit des *DLS* vom Linearisierungsbeacon Berücksichtigung findet (Gleichung 7.25), werden alle Algorithmen gleich genau. Auch werden die kleineren, und damit genaueren, Distanzen stärker gewichtet. Der mittlere Punktfehler beim *RAL*–Algorithmus verringert sich auf 0,70 %, der des *DLS* auf 0,68 %. Für die kreisförmige Beaconverteilung ergibt sich ein vergleichbares Bild. Alle



Abbildung 7.17: Berechnete Sensorpositionen mit Berücksichtigung des strengen stochastischen Modells (a) bei zufälliger und (b) bei Kreisverteilung der Beacons sowie der Lokalisierungsfehler (c) bei zufälliger und (d) bei Kreisverteilung der Beacons.

Algorithmen liegen mit einem Lokalisierungsfehler von 0, 63 % im selben Genauigkeitsbereich. Da die drei untersuchten exakten Lokalisierungsalgorithmen auf demselben Modell beruhen und sich lediglich in der Art der Linearisierung unterscheiden, ist dieses Ergebnis plausibel.

In Tabelle 7.4 sind die Simulationsergebnisse unter Einbeziehung der strengen stochastischen Modelle aufgeführt. Die Fehler für das gesamte Sensornetzwerk gibt Abbildung 7.19 wieder. In Abbildung 7.19 (a) und (b) ist die Auswirkung der Korrelation des Linearisierungsbeacons gut zu beobachten. In Abbildung (a) sind die Fehler des vereinfachten stochastischen Modells aufgezeigt. Der Fehler bei *SL*



Abbildung 7.18: Fehlerellipsen der untersuchten Algorithmen unter Berücksichtigung der strengen stochastischen Modelle für (a) zufällige und (b) Kreisverteilung der Beacons.

Tabelle 7.4: Simulationsergebnisse für den Vergleich ausgewählter Lokalisierungsalgorithmen unter Berücksichtigung der strengen stochastischen Modelle (* Beaconverteilung).

Algo-	Lokalisierungsfehler [%]				
rithmus	zufällig*	kreisförmig*	zufällig*	kreisförmig*	
	$\sigma_d^2 = 10\%$	$\sigma_d^2 = 10 \%$	$\sigma_d^2 = 0\%$	$\sigma_d^2 = 0\%$	
SL	0,67	0,63	0,0	0,0	
RAL	0,70	0,63	0,0	0,0	
DLS	0,68	0,63	0,0	0,0	

und *RAL* verteilt sich gleichmäßig über das Sensornetz. Die einzelnen Ausschläge sind hauptsächlich auf die Netzgeometrie zurückzuführen. In den Gebieten, in denen die Beacondichte dünner und damit die Distanzen Sensorknoten–Beacon größer werden, ist der Fehler maximal. In den Bereichen mit hoher Beacondichte wird der Fehler minimal. In Einzelgebieten erreicht dieser ein Minimum von 0,21 % und in den dünner besetzten Gebieten ein Maximum von bis zu 1,27 % für *SL*. Bei *RAL* bewegen sich die Genauigkeiten in vergleichbaren Größenordnungen. Bei *DLS* hingegen steigt der Fehler mit zunehmender Distanz zum Linearisierer und erreicht sein Maximum an den entfernten Stellen von bis zu 8,98 %, was sich wiederum auf die stärkere Gewichtung größerer Distanzen bei dem vereinfachten Modell zurückführen lässt. In der Nähe des Linearisierers ist das Minimum von 0,22 % erreicht. Es bewegt sich damit im Genauigkeitsbereich von *DLS* und *RAL*.

Im Fall des approximativen Algorithmus *WCL* ist der Fehler in den Randbereichen des Sensorfeldes am größten. Das liegt an dem in Abschnitt 7.2.1 diskutierten Zusammenhang. Durch den Schwerpunkt-



Abbildung 7.19: Positionsfehler für das gesamte Netzwerk bei zufälliger Beaconverteilung (a) mit vereinfachtem und (b) mit strengem stochastischen Modells und die Fehlerellipsen aller Punkte in Abbildung (c). Die Fehlerellipsen sind für SL, RAL und DLS 6- und für WCL 2-fach vergrößert dargestellt. Die optimale Potenz für das Gewicht g für WCL gibt Abbildung (d) wieder.

charakter wird der Sensorknoten in Richtung des Schwerpunktes gezogen. Gerade an den Grenzen oder außerhalb des einschließenden Polygons der Beacons wird der Fehler maximal. In diesem Fall hat dieser einen Betrag von 32,71 % und erreicht in der Nähe des Schwerpunktes des Sensorfeldes ein Minimum von 3,09 %.

Werden die strengen stochastischen Modelle in die Berechnung mit einbezogen ergibt sich für alle exakten Lokalisierungsalgorithmen ein vergleichbares Ergebnis. Die Berücksichtigung der stochastischen Abhängigkeit bei *DLS* vom Linearisierungsbeacon ist im Ergebnis nicht erkennbar. *SL*, *DLS* und *RAL* haben ihre maximalen Fehler in den selben Bereichen des Sensorfeldes, was sich wiederum auf

die Netzgeometrie zurückführen lässt. Dabei weist *RAL* den größten Fehler auf. Sein Maximalbetrag beträgt 1,21 % und das von *DLS* und *SL* bei 1,03 %.

Im Abbildungsteil 7.19 (c) sind die Fehlerellipsen aller Sensorknoten für die vier untersuchten Algorithmen unter Berücksichtigung des strengen stochastischen Modells abgebildet. Zu beachten ist, dass die Fehlerellipsen für *SL*, *RAL* und *DLS* um Faktor 6 sowie für *WCL* um Faktor 2 vergrößert dargestellt sind. Im Fall der exakten Methoden haben alle Ellipsen die gleichen Ausdehnungen und Richtungen. Bei *WCL* lässt sich das Schwerpunktverhalten beobachten. Alle Fehlerellipsen richten sich zum Schwerpunkt hin aus, d. h., der Positionsfehler wird in Richtung zum Schwerpunkt am größten.

Die Verteilung der errechneten optimalen Potenz für das *WCL*–Gewicht ist in Abbildungsteil 7.19 (d) aufgezeigt. Eine systematische Verteilung lässt sich nicht beobachten, was in der zufälligen Beaconverteilung begründet ist. Allerdings sind die höchsten Potenzen in Nähe hoher Beaconkonzentrationen zu erkennen. Die optimalen Potenzen bewegen sich zwischen 0,5 und 12,2. Am häufigsten wurde die optimale Potenz mit 2,4 berechnet.

7.4 Performanz der untersuchten Algorithmen

Reichenbach führte in seiner Dissertation das Energie-Fehler-Produkt (*EFP*) ein [Rei07]. Dieses Produkt erlaubt es, verschiedene Lokalisierungsalgorithmen direkt miteinander zu vergleichen. Dabei wird die erreichbare Genauigkeit mit dem Lokalisierungsfehler multipliziert. Den Berechnungen liegt der Simulationsaufbau aus Tabelle 7.5 zugrunde. Die Algorithmen *RAL* und *DLS* sind als die performantesten

Parameter	Wert	Parameter	Wert
Sensorfeldfläche [m]	100×100	Sendereichweite Beacons [m]	35
Anzahl Sensorknoten	500	Streckenvarianz [m ²]	25
Anzahl Beacons	100	Varianz Beaconpositionen [m ²]	5
Sendereichweite	10		
Sensorknoten [m]	10		

Tabelle 7.5: Simulationssetup für das Energie-Fehler-Protokoll (EFP) für ausgewählte Lokalisierungsalgorithmen.

Methoden einzustufen, da sie mit einem geringen Energieverbrauch sehr genaue Ergebnisse liefern. Die Standardlokalisierung *SL* liefert ebenfalls sehr genaue Ergebnisse, jedoch ist der Energieaufwand sehr hoch, da sämtliche Berechnungen auf allen Sensorknoten ausgeführt werden müssen. Bei dem "Fine-grained Localization"–Algorithmus ergibt sich ebenfalls ein sehr großes *EFP*. Der *FGL* wird wie die *SL* auf jedem Sensorknoten durchgeführt. Allerdings liefert er einen größeren Lokalisierungsfehler als *SL*, was sich auf das *EFP* auswirkt. Tabelle 7.6 führt die Ergebnisse für ausgewählte Algorithmen auf und Abbildung 7.20 stellt diese grafisch dar.

RAL und *DLS* verlaufen verteilt, die Hauptlast der Berechnung wird auf der Senke ausgeführt, die keinen Limitationen unterliegt und daher nicht in das *EFP* einfließt. Für die Lokalisierung mit *RAL* und nachträglicher Ausgleichung (*RAL+AGL*) ergibt sich ein interessantes Bild. Der zusätzliche Kommunikationsaufwand zum Versenden der Messwerte und berechneten Positionen muss zum Energieverbrauch

	RAL+AGL	RAL	DLS	WCL	SL	FGL
EFP	10,75	25,13	45,66	69,22	287,64	303,81

 Tabelle 7.6: EFP für ausgewählte Lokalisierungsalgorithmen.

für die Lokalisierung hinzuaddiert werden. Damit erhöht sich der durchschnittliche Energieaufwand auf 10,3 *J* (vgl. Tabelle 6.8 in Abschnitt 6.4.4). Die eigentliche Berechnung wiederum wird auf der Senke ausgeführt und zählt somit nicht in die Energiebilanz. Die Ausgleichung erhöht allerdings die Genauigkeit der Positionsbestimmung signifikant, was in einem niedrigeren *EFP* resultiert. Damit erreicht *RAL+AGL* das geringste *EFP* der untersuchten Lokalisierungsalgorithmen.



Abbildung 7.20: Energie-Fehler-Protokoll für Lokalisierungsalgorithmen.

7.5 Fazit

In diesem Kapitel wurden ein approximativer und zwei exakte Lokalisierungsalgorithmen auf erreichbare Genauigkeiten bei der Positionsbestimmung untersucht sowie der Standardlokalisierung, als Referenzalgorithmus, gegenübergestellt. Obwohl sich die exakten Lokalisierungsalgorithmen ausschließlich in der Art der Linearisierung unterscheiden, weisen sie in ihrem Fehlerverhalten teilweise gravierende Unterschiede auf. Besonders der Einfluss der einfangs erwähnten Vereinfachung im stochastischen Modell wirkt sich auf die Genauigkeit der Positionsbestimmung aus. Allerdings ist dieser Einfluss in den durchgeführten Simulationen im überschaubaren Rahmen. Der feinkörnige RAL-Algorithmus stellte sich als der performanteste und genaueste der hier untersuchten Algorithmen heraus. Beide, RAL und DLS reichen in ihrer erreichbaren Genauigkeit nicht ganz an SL heran. Diesem vermeintlichen Nachteil steht jedoch eine äußerst ressourceneffiziente Lokalisierung gegenüber. Damit können redundante Berechnungen vermieden und die Lebensdauer des Netzwerkes erhöht werden. Sie nutzen weiterhin die Eigenschaften des Netzwerkes, wie z. B. leistungsstärkere Beacons und einer Datensenke, die keinen Energielimitationen unterliegt. Damit bieten sich beide Algorithmen für die feinkörnige Lokalisierung in drahtlosen Geosensornetzwerken an und sind komplexen und aufwendigen Algorithmen wie der SL vorzuziehen. Die Genauigkeitseinbußen, deren Ursache in den Vereinfachungen des stochastischen Modells begründet liegt, können in einer nachträglichen Gesamtnetzausgleichung signifikant reduziert werden.

Mit dem *WCL* wurde ein approximativer Lokalisierungsalgorithmus untrersucht, der sich aufgrund seines grobkörnigen Ansatzes nicht mit den feinkörnigen Algorithmen vergleichen lässt. Jedoch zählt dieser zu den meist zitierten Algorithmen in der Literatur, wobei im Wesentlichen Einsparung energetischer Ressourcen im Vordergrund standen, jedoch weniger das Fehlerverhalten bei der Positionsbestimmung diskutiert wurde. Aufgabe dieses Kapitels war es, diese Lücke zu schließen. Dabei wurde für den *WCL* eine ausführliche Fehlerfortpflanzung gerechnet und der Einfluss der Netzgeometrie auf die Genauigkeit der Positionsbestimmung genauer untersucht. Da der *WCL* auf der Schwerpunktbestimmung aufbaut, haben die durchgeführten Veränderungen in diesem Algorithmus teilweise erhebliche Konsequenzen. So bietet sich mit dem *WCL* die Möglichkeit, bei bestimmten Netzgeometrien eine hohe Lokalisierungsgenauigkeit, die sich im Bereich der feinkörnigen Algorithmen bewegen kann.

8 Zusammenfassung und Ausblick

"Ein Mathematiker ist ein Mensch, der einen ihm vorgetragenen Gedanken nicht nur sofort begreift, sondern auch erkennt, auf welchem Denkfehler er beruht."

(Helmar Nahr, dt. Mathematiker)

8.1 Zusammenfassung

Das Forschungsgebiet drahtloser Geosensornetzwerke ist noch nicht abgeschlossen und wird auch auf längere Sicht ein wichtiger Gegenstand wissenschaftlicher Fragestellungen bleiben. Vor allem auf dem Gebiet der Hardware, und dort im Besonderen in der Verbesserung aktueller Energiespeicher, lässt sich ein enormer Forschungsbedarf beobachten. Ziel muss es sein, Energieträger mit höherer Energiedichte zu entwickeln, um den hohen Bedarf der Kommunikations- und Sensormodule gerecht zu werden. Neben der Entwicklung neuer und besserer Energiespeicher und energieeffizienter Hardware ist die Softwareentwicklung ressourcensparender Algorithmen ein wichtiger Zweig der Forschung. Die Fragestellung dieser Arbeit war es daher, effiziente Lokalisierungsalgorithmen zu entwerfen und ihre Genauigkeiten unter Berücksichtigung des dafür aufzuwendenden energetischen Aufwands abzuschätzen. Die Fragestellung lässt sich unter folgenden Punkten zusammenfassen:

- 1. Verwendung von Sensormesswerten für die Lokalisierung: Sensorknoten sind mit einer Anzahl Sensoren bestückt, um physikalische Parameter zu messen. Können diese Messwerte benutzt werden, um die Positionsschätzung zu verbessern?
- 2. Entwicklung von Lokalisierungsalgorithmen: In der Literatur existieren vielfältige Lokalisierungsalgorithmen, die nach ihrem Ansatz in exakte (feinkörnige) und approximative (grobkörnige) gegliedert werden. Kann durch einen einfachen Ansatz die Genauigkeit bisheriger Techniken erhöht werden, ohne dabei den Berechnungs- und Kommunikationsaufwand zu erhöhen?
- 3. Adaption der geodätischen Netzausgleichung: Die geodätische Ausgleichung beschäftigt sich mit Genauigkeitsanalysen und Berechnungsmethoden, die es erlauben, Aussagen über erreichbare Genauigkeiten zu treffen und fehlerbehaftete Messwerte derart auszugleichen, dass die Lösung die beste, lineare, unverzerrte Schätzung der unbekannten Parameter darstellt. Damit ist die geodätische Ausgleichung das optimale Werkzeug für die Lokalisierung in drahtlosen Sensornetzwerken mit einer hohen Anzahl von Messungen. Kann sie dazu genutzt werden, eine signifikante Erhöhung der Genauigkeit zu erzielen?

8.2 Ergebnisse der Arbeit

Verwendung von Sensormesswerten und Signalausbreitungsmodellen für die Lokalisierung

In ausführlichen Simulationen konnte gezeigt werden, dass die Anwendung des *ACL*–Algorithmus den mittleren Fehler der Lokalisierung mittels Trilateration um bis zu 30 % verbessern kann, wenn die gemessenen Distanzen eine Standardabweichung von 10 % aufweisen. Bei 5 % Abweichung beträgt die Verbesserung noch etwa 12 %. Diese Ergebnisse wurden bei drei bekannten Sensorintervallen erreicht, welche etwa ein Drittel des Messfeldes abdeckten. Mit weiteren Vorbedingungen kann die Lokalisierung weiter verbessert werden. Lediglich bei sehr geringen Distanzfehlern mit Abweichungen von unter 1 % ist die Anwendung des Algorithmus nicht angebracht, da hier zum Teil ein erhöhter Lokalisierungsfehler festgestellt wurde.

Es hat sich gezeigt, dass eine Verwendung von Sensormesswerten zur Lokalisierung durchaus sinnvoll ist, da diese Daten ohnehin zur Verfügung stehen. Mit geringem Mehraufwand für die Sensorknoten ist eine deutliche Verbesserung der Ergebnisse erzielbar. Der konkrete Nutzen ist jeweils sehr stark davon abhängig, wie genau gemessen werden kann und wie viele Vorbedingungen festlegbar sind. Ein Vorteil dieses Verfahrens ist, dass keine zusätzlichen Fehler erzeugt werden, denn es wird lediglich eine Filterung der Eingangswerte durchgeführt. Dies kann sich zwar vereinzelt ungünstig auswirken, doch global gesehen wird das Ergebnis dadurch genauer. Ein weiterer Vorteil des geschaffenen Modells ist die Möglichkeit, theoretisch beliebig komplexe Vorbedingungen zu formulieren.

Zusammenfassend kann der *ACL*–Algorithmus als effiziente Zusatzmethode zur Lokalisierung angesehen werden. Schon mit wenigen Vorbedingungen ist es möglich, die Lokalisierung zu verbessern oder zumindest Positionen zu verifizieren. Vor allem in Kombination mit approximativen Algorithmen lassen sich gute Ergebnisse erzielen, jedoch auch mit exakter Lokalisierung bei entsprechend großen Distanzfehlern.

Im nächsten Schritt wurde dieser Ansatz für Indooranwendungen erweitert. Da in diesen Umgebungen Einflüsse stärker in Erscheinung treten, die die Signale auf dem Weg vom Sender zum Empfänger beeinträchtigen und verfälschen, muss die Signalausbreitung stärker berücksichtigt werden.

In den Simulationen konnte gezeigt werden, dass Hindernisse und deren Beschaffenheit die Genauigkeit der Positionsbestimmung maßgeblich beeinflussen. Im Gegensatz zu Outdooranwendungen, in denen zumeist nur die Freiraumdämpfung das Signal beeinflusst, ist dieser Effekt bei Indoorszenarien besonders stark. Durch Einbeziehen der auftretenden Dämpfungen, die das Signal bei dem Durchlaufen der Hindernisse erfährt, können diese Einflüsse jedoch minimiert werden. Dazu wurde ein Algorithmus vorgestellt, der spezielle Dämpfungseffekte berücksichtigt. In Simulationen konnte eine Verbesserung der Genauigkeit der Positionsbestimmung bis zu Faktor 4 erreicht werden. Bei sehr geringen Streckenfehlern, bedingt durch geringe Dämpfung, war die Anwendung von *ACL* ausreichend. Bei Holzwänden mit relativ geringem Dämpfungsvermögen lieferten beide Algorithmen Genauigkeiten im selben Bereich. Stieg jedoch die Streckenungenauigkeit bedingt durch ein höheres Dämpfungsvermögen der Hindernisse, verbesserte sich auch die Position signifikant, die mittels *eACL* korrigiert wurde. Die auftretenden Abweichungen in den Positionsberechnungen sind auf die Einflüsse der Reflexion zurückzuführen, die in der Simulation der Streckenmessung nur teilweise berücksichtigt wurde. *eACL* berücksichtigt allein die Dämpfungseffekte durch Hindernisse auf der Sichtverbindung Sender-Empfänger. Ein weiterer Ausbau des Algorithmus wäre denkbar, ist allerdings aufgrund der Komplexität der Reflexion und der verbleibenden Effekte sehr aufwendig und höchstwahrscheinlich nicht zielführend, da dafür auf den ressourcenlimitierten Sensorknoten aufwendige Rechen- und Interpolationsverfahren notwendig wären, um eine realitätsnahe Signalausbreitungskarte in Echtzeit zu erstellen. Eine Vorberechnung und Speicherung auf den Knoten wäre denkbar, ist allerdings nicht in dynamischen Szenarien zu empfehlen, da die Strahlungsmuster recht stark auf Veränderungen reagieren. Dämpfungen können bei Anwendungen des "Path Attenuation Factor"–Modells jedoch recht genau vorhergesagt werden und sind für die Anwendung bei *eACL* hinreichend, wie in den Simulationen gezeigt werden konnte.

Entwicklung und Vergleich von Lokalisierungsalgorithmen

Entwicklung

In einem ersten Schritt wurde mit dem "Resource Aware Localization"–Algorithmus *RAL* ein neuer, feinkörniger Lokalisierungsalgorithmus auf Basis des "Distributed Least Squares"–Algorithmus *DLS* vorgeschlagen. Durch eine alternative Form der Linearisierung des nichtlinearen Gleichungssystems kann der Berechnungsaufwand (und damit der Energieverbrauch im Sensornetzwerk) weiter gesenkt werden. Im Wesentlichen ergeben sich für den *RAL* folgende Vorteile:

- Geringe Lokalisierungsfehler von unter $1\,\%$ in Simulationen.
- Hohe Skalierbarkeit und Robustheit des Sensornetzes bei Ausfällen oder Hinzunahme von Beacons, da mit Orthogonalisierungsmethoden ohne großen Aufwand Beacons aus der Berechnung entfernt oder hinzugefügt werden können.
- Durch Aufteilung der Berechnungen ergeben sich geringe Ressourcenanforderungen an die Sensorknoten:
 - **Berechnungsaufwand** Es ergibt sich eine Komplexität von 6n 2 *Flops* für die Normalgleichungen und 8n + 3 *Flops* für die QR-Faktorisierung.
 - **Kommunikationsaufwand** Empfang von $2 \cdot n + n$ Gleitkommazahlen für die Normalgleichung und $3 \cdot n + 3 \cdot n + n$ bei Verwendung der QR-Faktorisierung.

Leider konnte der neu entwickelte *RAL*–Algorithmus ausschließlich im Simulator getestet werden. Eine Erprobung in der Realität konnte nicht umgesetzt werden, da die Hardwareausstattung nicht gegeben war.

Vergleich

Obwohl *FGL*, *RAL* und *DLS* auf demselben Beobachtungssatz und der Methode der kleinsten Quadrate basieren, ist ihr Fehlerverhalten unterschiedlich. Dabei haben die Beaconpositionen (Netzgeometrie) und der Varianzbereich der Beobachtungen einen sehr großen Einfluss. Beim *DLS* kommt zusätzlich der Einfluss der fehlerhaften Distanzbeobachtung zum festgehaltenen Beacon hinzu, was sich als starke Korrelation im stochastischen Modell niederschlägt. Eine Berücksichtigung dieser in der Vorberechnung

ist allerdings nicht ohne Weiteres umsetzbar, da die Distanzbeobachtungen in das stochastische Modell mit einfließen und diese apriori nicht vorliegen.

DLS und *RAL* eignen sich sehr gut für die Positionsberechnung in Geosensornetzwerken. Besonders die Möglichkeit, den Berechnungsaufwand in eine komplexe Vorberechnung auf der Senke und eine simple Nachberechnung auf den Sensorknoten aufzuteilen, macht die Anwendung in drahtlosen Sensornetzwerken interessant. Durch diese Aufteilung kann der Berechnungsaufwand im Sensornetz minimiert werden, was gleichbedeutend einer signifikanten Energieeinsparung ist. Dabei liefert der *RAL* durch den funktionalen Ansatz per Substitution genauere Ergebnisse als *DLS*. Wird jedoch das stochastische Modell mit berücksichtigt oder als Linearisierer ein Beacon in unmittelbarer Nähe des Sensorknotens gewählt, liefern *DLS* und *RAL* Ergebnisse im selben Genauigkeitsbereich. Allerdings ist im ersten Fall keine verteilte Berechnung möglich, da in das stochastische Modell die Streckenbeobachtungen mit einfließen.

Sollte die Sensorposition in der Nähe des Schwerpunktes des Sensornetzwerkes liegen, ist es möglich, dass der approximative *WCL*–Algorithmus bessere Ergebnisse liefern kann als der *DLS*. Da diese Möglichkeit apriori, d. h., vor dem Ausbringen des Sensorfeldes nicht vorausgesetzt werden kann, ist diese Möglichkeit als eher theoretisch anzusehen.

Clusterisierung und Geodätische Netzausgleichung

In einem weiteren Schritt wurde die Möglichkeit untersucht, die geodätische Netzausgleichung auf ein drahtloses Sensornetzwerk zu adaptieren.

Mit der geodätischen Ausgleichungsrechnung kann die Genauigkeit der Sensorknotenposition signifikant gesteigert werden. Als nachteilig erweist sich jedoch der zusätzliche Kommunikationsaufwand, der nötig ist, um alle notwendigen Informationen über die Sensorknoten zu den Beacons und weiter zur Senke zu verschicken. Dadurch werden die Energieressourcen des gesamten Netzwerkes zusätzlich belastet. Da die Knotenpositionen ohnehin zur Senke übermittelt werden müssen, können diese eigentlich aus der Aufwandsberechnung gestrichen werden. Den größten Anteil an der Kommunikation übernimmt jedoch das Versenden aller gemessenen Distanzen.

Die Frage, ob das Sensornetzwerk abschließend ausgeglichen werden soll oder nicht, kann an dieser Stelle nicht beantwortet werden. Es bleibt eine Frage des Anwendungsszenarios. Werden hoch genaue Positionen (< 1 %) gefordert, wird der Anwender nicht um eine Netzausgleichung, und somit einem weiteren Energieaufwand, umhin kommen. Sind jedoch relativ genaue Knotenpositionen (1 % < x < 10 %) ausreichend, sollte mit Hinblick auf den hohen Ressourcenbedarf und zugunsten einer längeren Lebensdauer des Netzwerkes auf eine nachträgliche Netzausgleichung verzichtet werden.

Liegen große Netzwerke vor, ist eine Clusterisierung in jedem Fall empfehlenswert. Zum einen erhöht diese die Robustheit des Netzwerkes und verringert den Energieverbrauch, da nicht alle Beacons in die Berechnung einfließen und sich mögliche Ausfälle nur auf wenige Cluster auswirken. Eine abschließende Nachausgleichung mit Blockzerlegung oder Transformation ermöglicht einen einfachen Zusammenschluss mehrerer Einzelcluster.

8.3 Ausblick

Im Fokus nachfolgender Arbeiten sollte die experimentelle Umsetzung und Erprobung der in dieser Arbeit entwickelten Algorithmen in realen Szenarien stehen. Der *DLS*–Algorithmus wurde bereits erfolgreich erprobt, was darauf schließen lässt, dass der ähnliche *RAL* ebenfalls erfolgreich eingesetzt werden kann. Für *ACL* und *eACL* sind jedoch ausgedehnte Untersuchungen erforderlich.

In einem weiteren Schritt ist die Übertragung aller entwickelten Algorithmen auf mobile Sensornetze zu realisieren. Mit dem *mDLS* ist ein erster Schritt getan. Das Problem bei diesem Ansatz ist vielmehr, dass die Position mit der Zeit divergiert und große Fehler aufweist. Eine Kombination mit einer IMU auf Hardwareebene wäre denkbar, um die berechnete Position mit Messwerten der IMU zu stützen. Nachteil bei diesem Ansatz ist ebenfalls die Divergenz der IMU über die Zeit. Auf Algorithmenebene können Bayes'sche Filter oder Partikelfilter implementiert werden, um eine Positionsbestimmung in mobilen Umgebungen zu gewährleisten.

Abschließend ist zu sagen, dass für die Lokalisierung auf Algorithmenebene ein rasantes Fortschreiten der Entwicklungen zu beobachten ist. Die größte Herausforderung liegt in der Weiterentwicklung der in dieser Arbeit häufig erwähnten Energieträger, um eine lange Lebensdauer in der Praxis zu ermöglichen.

Literaturverzeichnis

- [AAA⁺07] ARNHARDT, C. ; ASCH, K. ; AZZAM, R. ; BILL, R. ; FERNANDEZ-STEEGER, T.M. ; HORN-FELD, S.D. ; KALLASH, A. ; NIEMEYER, F. ; RITTER, H. ; TOLOCZYKI, M. ; WALTER, K.: Sensor based Landslide Early Warning System - SLEWS, Development of a Geoservice Infrastructure as Basis for Early Warning Systems for Landslides by Integration of Real-Time Sensors. In: STROINK, L. (Hrsg.): *Early Warning Systems in Earth Management: Kick-Off-Meeting.* Bd. 10. Potsdam : Koordinationsbüro GEOTECHNOLOGIEN, 2007, S. 75–88 28
- [AAG] ARC-ADVISORY-GROUP: Wireless technology report. http://www.processingtalk. com/news/ajw/ajw112.html. - Zuletzt aufgerufen: 31.01.2011 9
- [AKT09] ATHANASIOU, G.; KORAKIS, T.; TASSIULAS1, L.: An 802.11k Compliant Framework for Cooperative Handoff in Wireless Networks. In: EURASIP Journal on Wireless Communications and Networking (2009), S. 1–14 59
- [Ame53] AMENT, W.S.: Toward a Theory of Reflection by a Rough Surface. In: *Proceedings of the IRE* Bd. 41, 1953, S. 142–146 79
- [ASS05] AHMED, A.A.; SHI, H.; SHANG, Y.: SHARP: A New Approach to Relative Localization in Wireless Sensor Networks. In: *Proceedings of the 2nd International Workshop on Wireless Ad Hoc Networking*, 2005, S. 892–898 56, 57
- [Ban] BANNER ENGINEERING: Wireless Industrial I/O Sensor Network Applications. http:// www.bannerengineering.com/en-US/wireless/surecross_web_appnotes.-Zuletzt aufgerufen: 31.01.2011 28
- [BB05] BLAZEVIC, L.; BOUDEC, J.-Y. L.: A Location-Based Routing Method for Mobile Ad Hoc Networks. In: *IEEE Transactions on Mobile Computing* (2005), S. 97–110 12
- [BB10] BORN, A.; BILL, R.: Reducing the Calculation for Precise Localization in Wireless Sensor Networks. In: Proceedings of the 2010 IEEE 71st Vehicular Technology Conference: VTC2010-Spring, 2010, S. 1–5. – Taipei, TW 98
- [BCKM04] BILL, R.; CAP, C.; KOFAHL, M.; MUNDT, T.: Indoor and Outdoor Positioning in Mobile Environments, A Review and some Investigations on WLAN Positioning. In: Geographic Information Sciences (2004), S. 91–98 11
- [Ber] BERKELEY UNIVERSITY: Smart Dust Autonomous sensing and communication in a cubic millimeter. http://robotics.eecs.berkeley.edu/~pister/SmartDust/. - Zuletzt aufgerufen: 31.01.2011 11
- [BHM⁺09] BOGENA, H. R. ; HUISMAN, J. A. ; MEIER, H. ; ROSENBAUM, U. ; WEUTHEN, A.: Hybrid wireless underground sensor networks: Quantification of signal attenuation in soil. In: *Vadose Zone Journal* (2009), August, S. 755 – 761 27

- [Bil10] BILL, R.: Grundlagen der Geo-Informationssysteme. Wichmann Verlag, 2010 9, 10
- [Bjö96] BJÖRCK, A.: *Numerical Methods for Least Squares Problems*. Mathematics Society for Industrial and Applied Mathematics, 1996 31
- [BKK⁺04] BARBEAU, M. ; KRANAKIS, E. ; KRIZANC, D. ; MORIN, P. ; NIKOLAIDIS, I.: Improving Distance Based Geographic Location Techniques in Sensor Networks. In: *Proceedings* of the 3rd International Conference on Ad-hoc, Mobile, and Wireless Networks, 2004. – ISBN 3540225439, S. 197–210 56
- [Blu] BLUETOOTH SPECIAL INTEREST GROUP (SIG): Bluetooth. http://www.bluetooth. com/English/Technology/Pages/Basics.aspx. - Zuletzt aufgerufen: 31.01.2011 19
- [Blu08] BLUMENTHAL, J.: Ressourcenarme und dezentrale Lokalisierung autonomer Sensorknoten in Sensornetzwerken, Universität Rostock, Diss., 2008 122, 124
- [BN09] BORN, A.; NIEMEYER, F.: A New Localization and Accuracy Analyzer for Wireless Sensor Networks using Defective Observations. In: 5th Workshop on Positioning, Navigation and Communication (WPNC2009), 2009. – ISBN 978–1–4244–3292–9, S. 83 – 88 123
- [BNSB10] BORN, A. ; NIEMEYER, F. ; SCHWIEDE, M. ; BILL, R.: Using Signal Propagation Models to Improve Distance Estimations for Localisation in Wireless Geosensor Networks. In: *Proceedings of Ninth International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences: Accuracy 2010*, 2010, S. 45–48. – Leicester, UK 84
- [BNW09] BLANKENBACH, J.; NORRDINE, A.; WILLERT, V.: Ultra Wideband Based Indoor Positioning: A Localization Prototype for Precise 3D Positioning and Orientation. In: Proceedings of the 9th Conference on Optical 3-D Measurement Techniques, 2009, S. 1–6 64
- [BP00] BAHL, P. ; PADMANABHAN, V.N.: RADAR: An In-Building RF-Based User Location and Tracking System. In: *IEEE Infocom* (2000), S. 775–784. – Tel-Aviv, Israel 54
- [BR09] BILL, R. ; RESNIK, B.: Vermessungskunde für den Planungs-, Bau- und Umweltbereich.
 Wichmann Verlag, Heidelberg, 2009. ISBN 978–3–879–074884 32
- [BR10] BORN, A. ; REICHENBACH, F.: Converting the Nonlinear Least Squares Problem for Localization in Wireless Sensor Networks. In: Proceedings of the ICCCN 2010 WiMAN Workshop, 2010, S. 1–6 98
- [BRBT06] BORN, A. ; REICHENBACH, F. ; BILL, R. ; TIMMERMANN, D.: Bestimmung Optimaler Startwerte zur Exakten Lokalisierung mittels Geodätischer Ausgleichung. (2006), Juli, S. 93–98. – Stuttgart, Deutschland 54, 95
- [BRBT08] BORN, A. ; REICHENBACH, F. ; BILL, R. ; TIMMERMANN, D.: Lokalisierung in Ad Hoc Geosensornetzwerken mittels geodätischer Ausgleichungstechnik. In: GIS–Zeitschrift für Geo-Informationssysteme (2008), April, S. 4–16. – ISSN 14303663. – Heidelberg, Germany 13, 26, 56, 122
- [BRT05a] BLUMENTHAL, J. ; REICHENBACH, F. ; TIMMERMANN, D.: Position Estimation in Ad hoc Wireless Sensor Networks with Low Complexity. In: *Proceedings of the 2nd Workshop on Positioning, Navigation and Communication and 1st Ultra-Wideband Expert Talk (WP-*

NC2005), Shaker Verlag, März 2005. – ISBN 3832237461, S. 41–49. – Hannover, Deutschland 52

- [BRT05b] BLUMENTHAL, J. ; REICHENBACH, F. ; TIMMERMANN, D.: Precise Positioning with a Low Complexity Algorithm in Ad hoc Wireless Sensor Networks. In: *PIK - Praxis der Informationsverarbeitung und Kommunikation* 28 (2005), Juni, S. 80–85. ISBN 3598012527 122
- [BRT06] BLUMENTHAL, J. ; REICHENBACH, F. ; TIMMERMANN, D.: Minimal Transmission Power vs. Signal Strength as Distance Estimation for Localization in Wireless Sensor Networks. (2006), Juni, S. 761–766. – New York, USA 59
- [BSB10] BORN, A.; SCHWIEDE, M.; BILL, R.: On Distance Estimation based on Radio Propagation Models and Outlier Detection for Indoor Localization in Wireless Geosensor Networks. In: *Proceedings of Conference for Indoor Positioning and Indoor Navigation: IPIN2010*, 2010, S. 1–6 84
- [BSG⁺09] BEHNKE, R. ; SALZMANN, J. ; GROSSMANN, R. ; LIECKFELDT, D. ; TIMMERMANN, D. ; THUROW, K.: Strategies to overcome Border Area Effects of Coarse Grained Localization. In: Proceedings of the 6th Workshop on Positioning, Navigation and Communication. WPNC2009. (2009), S. 95–102 53
- [BSLT09] BEHNKE, R. ; SALZMANN, J. ; LIECKFELDT, D. ; TIMMERMANN, D.: sDLS- Distributed Least Squares Localization for Large Wireless Sensor Networks. In: *Proceedings of International Conference on Ultra Modern Telecommunications & Workshops. ICUMT09.*, 2009, S. 1–6 57, 112
- [BST10a] BEHNKE, R. ; SALZMANN, J. ; TIMMERMANN, D.: Improvements on Scalable Distributed Least Squares Localization for Large Wireless Sensor Networks. In: Proceedings of 5th IEEE International Symposium on Wireless Pervasive Computing. ISWPC10., 2010. – ISBN 978–1–4244–6857–7, S. 273–277 57
- [BST10b] BEHNKE, R. ; SALZMANN, J. ; TIMMERMANN, D.: sDLS^{ne} Improved Scalable Distributed Least Squares Localization with minimized Communication. In: Proceedings of the 21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2010) (2010), September 57, 112
- [BTB⁺06] BLUMENTHAL, J. ; TIMMERMANN, D. ; BUSCHMANN, C. ; FISCHER, S. ; KOBERSTEIN, J. ; LUTTENBERGER, N.: Minimal Transmission Power as Distance Estimation for Precise Localization in Sensor Networks. In: *Proceedings of the International Wireless Communications and Mobile Computing Conference*, 2006. – ISBN 15959333069, S. 1331–1336. – Vancouver, Canada 59
- [Bul00] BULUSU, N.: GPS-Less Low Cost Outdoor Localization for Very Small Devices. In: *IEEE Personal Communications Magazine* 7 (2000), S. 28–34 34, 51, 52
- [Bus99] BUSINESSWEEK: 21 ideas for the 21st century. (1999), August, S. 78–167 9
- [BW03] BRUNNER, F.K.; WIESER, A.: Berechnung des Varianzfaktors bei Mehrfachbeobachtungen im Gauss–Markov–Modell. In: AVN–Allgemeine Vermessungs-Nachrichten (2003), Nr. 3, S. 87–90 109

[CFP ⁺ 06]	CHINTALAPUDI, K. ; FU, T. ; PAEK, J. ; KOTHARI, N. ; RANGWALA, S. ; CAFFREY, J. ; GOVINDAN, R. ; JOHNSON, E. ; MASRI, S.: Monitoring Civil Structures with a Wireless Sensor Network. In: <i>IEEE Internet Computing</i> 10 (2006), S. 26–34. – ISSN 10897801 28
[CHH02]	CAPKUN, S. ; HAMDI, M. ; HUBAUX, JP.: GPS-free Positioning in Mobile Ad hoc Networks. In: <i>Cluster Computing</i> (2002), S. 157–167 56
[CHWJ10]	CHEN, JS. ; HONG, ZW. ; WANG, NC. ; JHUANG, SH.: Efficient Cluster Head Selection Methods for Wireless Sensor Networks. In: <i>Journal of Networks.</i> Bd. 5, 2010, S. 964–970 108
[DEA06]	DEMIRKOL, I. ; ERSOY, C. ; ALAGOZ, F.: MAC protocols for wireless sensor networks: a survey. In: <i>Communications Magazine, IEEE</i> 44 (2006), April, S. 115–121 21
[Dei06]	DEINERT, F.: <i>Mathematische Modelle zur Wellenausbreitung für die Simulation drahtloser</i> <i>Netze</i> . 2006. – Seminararbeit 82, 83
[DVA ⁺ 04]	DUNKELS, A. ; VOIGT, T. ; ALONSO, J. ; RITTER, H. ; SCHILLER, J.: Connecting Wireless Sensornets with TCP/IP Networks. In: <i>Proceedings of the 2nd International Conference</i> <i>on Wired/Wireless Internet Communications</i> , 2004, S. 143–152. – Frankfurt (Oder), Deutschland 27
[ene]	ENERGYMICRO: <i>EFM32 Tech nology - Ultra Low Power</i> . http://www.energymicro.com/technology/ 31.01.2011 25
[EnO]	ENOCEAN: <i>Produkte</i> . http://www.enocean.com/de/produkte_technologie/ Zuletzt aufgerufen: 31.01.2011 23
[EPF06]	E. PARTON, T. S. ; FIORINI, P.: Innovative Vibrationssensoren - Entwicklung elektrostati- scher Vibrationswandler auf MEMS-Basis. In: <i>E&E Kompendium 2005/06</i> (2006), January, S. 40 – 43 22
[ETH]	ETH ZÜRICH: Sensor Network Museum. http://www.btnode.ethz.ch/Projects/ SensorNetworkMuseum Zuletzt aufgerufen: 31.01.2011 22
[Fai03]	FAIRLEY, P.: 10 Emerging Technologies That Will Change the World. In: <i>MIT Technology</i> <i>Review</i> 106 (2003), Februar, S. 33–49 9
[FBVS10]	FINK, A. ; BEIKIRCH, H. ; VOSS, M. ; SCHRÖDER, C.: RSSI-based indoor positioning using diversity and Inertial Navigation. In: <i>2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)</i> , 2010, S. 1–7 63
[FLB09]	FONT-LLAGUNES, J. ; BATLLE, A.: New Method that solves the Three-Point Resection Problem using Straight Lines Intersection. In: <i>Journal of Surveying Engineering</i> 135 (2009), Nr. 2, S. 39–45 33
[GBY ⁺ 06]	GOLDENBERG, D.K.; BIHLER, P.; YANG, Y. R.; CAO, M.; FANG, J.; MORSE, A. S.; ANDERSON, B.D.O.: Localization in Sparse Networks using Sweeps. In: <i>Proceedings</i> <i>of the 12th Annual international Conference on Mobile computing and networking</i> , ACM Press, 2006. – ISBN 1595932860, S. 110–121. – Los Angeles, CA, USA 55
[GGL93]	GEORGE, A. ; GILBERT, J.R. ; LIU, J.W.H.: <i>Graph Theory and Sparse Matrix Computation</i> . 1993 111

- [Gib96] GIBSON, J.: The Mobile Communications Handbook. CRC Press, 1996. ISBN 3540648364 11
- [GL96] GOLUB, G.H.; LOAN, C.F. V.: *Matrix Computations*. The Johns Hopkins University Press, 1996 40, 99, 100, 110
- [GMF⁺07] GUILFORD, T. C. ; MEADE, J. ; FREEMAN, R. ; M., Perrins C. ; BIRO, D. ; EVANS, T. ; BONADONA, F. ; BOYLE, D.: Tracking the foraging movements of Manx shearwaters from Skomer Island by GPS. In: *Technical report for the Countryside Council for Wales*, 2007, S. 462–473 27
- [GR71] GOLUB, G.H.; REINSCH, C.: Singular Value Decomposition and Least Square Solutions, Linear Algebra, Volume II of Handbook for Automatic Computations. Springer Verlag, 1971 40, 110
- [GRB⁺06] GRASSERT, F. ; REICHENBACH, F. ; BLUMENTHAL, J. ; GOLATOWSKI, J. ; THUROW, K. ; TIMMERMANN, D.: Drahtloses Sensornetzwerk - Messen, was los ist. In: Landestechnologieanzeiger, 1/2006 (2006), April, S. 4–5. – Schwerin, Deutschland 28
- [Gro69] GROSSMANN, W.: *Grundzüge der Ausgleichungsrechnung*. Heidelberg, Deutschland : Springer Verlag, 1969 31
- [Grü80] GRÜNDIG, L.: Feasibility Study of the Conjugate Gradient Method for Solving Large Sparse Equation Sets. In: *NOAA Technical Report NOS 82 NGS 13*, 1980, S. 19 pp. 111
- [Grü96] GRÜNDIG, L.: Fehlerlehre und Statistik. 1996. Skript zur Vorlesung 40
- [Grü03] GRÜNDIG, L.: *Grundlagen der Ausgleichungsrechnung*. 2003. Skript zur Vorlesung 40, 41, 44, 45, 145
- [Han06] HANDY, M.: Energieeffiziente Algorithmen und Protokolle für drahtlose Ad-hoc- und Sensornetze, Universität Rostock, Diss., 2006 19
- [HARa] HART-COMMUNICATION FOUNDATION: HART-Communication Foundation. http:// www.hartcomm.org/. - Zuletzt aufgerufen: 31.01.2011 20
- [Harb] HARWARD UNIVERSITY: MoteTrack: A Robust, Decentralized Approach to RF-Based Location Tracking. http://www.eecs.harvard.edu/~konrad/projects/ motetrack/. - Zuletzt aufgerufen: 31.01.2011 23
- [HC08] HUI, J.W.; CULLER, D.E.: Extending IP to Low-Power, Wireless Personal Area Networks. In: Internet Computing, IEEE 12 (2008), Juli-Aug., Nr. 4, S. 37–45. http://dx.doi. org/10.1109/MIC.2008.79. – DOI 10.1109/MIC.2008.79. – ISSN 1089–7801 20
- [Heu08] HEUNECKE, O.: Geosensornetze im Umfeld der Ingenieurvermessung. In: Proceedings of the Sixth Annual Conference on Communication Networks and Services Research (CNSR2008), 2008, S. 357–364 9, 10
- [HHB⁺03] HE, T. ; HUANG, C. ; BLUM, B.M. ; STANKOVIC, J.A. ; ABDELZAHER, T.: Range-Free Localization Schemes for Large Scale Sensor Networks. In: *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking* (2003), S. 81–95.
 San Diego, CA, USA 53
- [HM06] HART, J.K.; MARTINEZ, K.: Environmental Sensor Networks: A revolution in the earth system science? In: *Earth-Science Reviews* (2006), 177–191. http://eprints.ecs.

	soton.ac.uk/13093/ 27
[HM09a]	HEIDARI, E.; MOVAGHAR, A.: Intelligent Clustering in Wireless Sensor Networks. In: <i>Proceedings of First International Conference on Networks and Communications. NET-COM09</i> , 2009. – ISBN 978–1–4244–5364–1, S. 12–16 108
[HM09b]	HERBERT, S. ; MARCULESCU, D.: Variation-aware dynamic voltage/frequency scaling. In: <i>Proceedings of the IEEE 15th International Symposium on High Performance Computer Architecture, 2009 (HPCA2009)</i> , 2009, S. 301 – 312 24
[Höp80]	HÖPCKE, W.: <i>Fehlerlehre und Ausgleichungsrechnung.</i> Walter de Gruyter, 1980. – ISBN 978–3110075144 113
[IEE]	IEEE: <i>IEEE</i> 802.15 TM : Wireless Personal Area Networks (PANs). http://standards. ieee.org/about/get/802/802.15.html Zuletzt aufgerufen: 31.01.2011 19
[Jac04]	JACOBS, S.: Battery Power for Remote Wireless Sensors. In: <i>SENSORS</i> (2004), S. 42 – 47 24
[JEZ ⁺ 05]	JAFARI, R. ; ENCARNACAO, A. ; ZAHOORY, A. ; DABIRI, F. ; NOSHADI, H. ; SARRAFZADEH, M.: Wireless Sensor Networks for Health Monitoring. In: <i>Proceedings of the 2nd Annual</i> <i>International Conference on Mobile and Ubiquitous Systems</i> , 2005. – ISBN 0769523757, S. 479–481 28
[JMS05]	JÄGER, R.R. ; MÜLLER, T. ; SALER, H.: <i>Klassische und robuste Ausgleichungsverfahren.</i> <i>Ein Leitfaden für Ausbildung und Praxis von Geodäten und Geoinformatikern.</i> Herbert Wichmann Verlag, 2005. – ISBN 9783879073702 31, 36, 40, 41, 44, 45
[jsi07]	J-Sim: A component-based, compositional Simulation Environment, http://www.j-sim.org/. http://www.j-sim.org/. Version: 2007 71
[JZ04]	JI, X. ; ZHA, H.: Sensor Positioning in Wireless Ad-Hoc Sensor Networks using Multidi- mensional Scaling. (2004), S. 2652–2661. ISBN 0780383559 56
[Kab05]	KABARA, J.: <i>Wireless Data Networks - Sensor Networks</i> . 2005. – Präsentation am Department of Information Science and Telecommunications, University of Pittsburgh 57
[Kah05]	KAHMEN, H.: Angewandte Geodäsie: Vermessungskunde. Gruyter, 2005 61
[KF10]	KARAVAS, S. ; FISCHER, T. M.: Voltree Power leading the way with bio-energy harvesting technology. In: <i>Vaisala News</i> (2010), January, S. 24 – 25 22
[KKKP00]	KAHN, J.M. ; KATZ, R.H. ; KATZ, Y.H. ; PISTER, K.S.J.: Emerging Challenges: Mobile Networking for "Smart Dust". In: <i>Journal of Communications and Networks</i> (2000), S. 188–196 11
[KKP99]	KAHN, J.M.; KATZ, R.H.; PISTER, K.S.J.: Mobile Networking for Smart Dust. In: <i>Proceedings of the International Conference on Mobile Computing and Networking</i> , 1999, S. 271–278. – Seattle, USA 18
[KMS ⁺ 05]	KWON, Y. ; MECHITOV, K. ; SUNDRESH, S. ; KIM, W. ; AGHA, G.: Resilient Localization for Sensor Networks in Outdoor Environments. In: <i>Proceedings of the 25th IEEE International</i> <i>Conference on Distributed Computing Systems</i> , 2005, S. 643–652 57

- [KS78] KLEINROCK, L.; SILVESTER, J.: Optimum Transmission Radii for Packet Radio Networks or why Six is a Magic Number. In: *Proceedings of the National Telecomm Conference*, 1978, S. 431–435 62
- [Leg] LEGG, G.: ZigBee: Wireless Technology for Low-Power Sensor Networks. http://www.eetimes.com/design/communications-design/4017853/ ZigBee-Wireless-Technology-for-Low-Power-Sensor-Networks. - Zuletzt aufgerufen: 31.01.2011 20
- [LGF10] LEHSAINI, M. ; GUYENNET, H. ; FEHAM, M.: Intelligent Clustering in Wireless Sensor Networks. In: *Network Protocols and Algorithms* (2010) 108
- [LH05] LANGENDOEN, K. ; HALKES, G.: *Energy-Efficient Medium Access Control*. CRC Press, 2005. ISBN 9780849328244 21
- [Lie10] LIECKFELDT, D.: *Effiziente Lokalisierung von Nutzern und Geräten in Smarten Umgebungen*, Universität Rostock, Diss., 2010 57
- [Lin96] LINMARTZ, JP.M.G.: Radio Propagation Models. 1996. CD-ROM 83
- [LPSC04] LEVIS, P. ; PATEL, N. ; SHENKER, S. ; CULLER, D.: Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks. In: Proceedings of the First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI), 2004, S. 15–28 24
- [LR03] LANGENDOEN, K.; REIJERS, N.: Distributed Localization in Wireless Sensor Networks: A Quantitative Comparison. In: Computer Networks (Elsevier), Special Issue on Wireless Sensor Networks (2003), S. 499–518 36, 57
- [LRWW98] LAGARIAS, J.C.; REEDS, J.A.; WRIGHT, M.H.; WRIGHT, P.E.: Convergence Properties of the Nelder-mead Simplex Methode in Low Dimensions. In: SIAM Journal of Otimization (1998), S. 112–147 34
- [LWG05] LANDSIEDEL, O.; WEHRLE, K.; GOTZ, S.: Accurate Prediction of Power Consumption in Sensor Networks. In: Proceedings of the 2nd IEEE Workshop on Embedded Networked Sensors (2005), S. 37–44. ISBN 0780392469 22
- [LYT08] LIECKFELDT, D.; YOU, J.; TIMMERMANN, D.: An Algorithm for Distributed Beacon Selection.
 In: IEEE International Conference on Pervasive Computing and Communications (2008), S. 318–323. ISBN 978–0–7695–3113–7 57
- [Mau10] MAUTZ, R.: *Positioning techniques.* 2010. Konferenzvortrag: Conference for Indoor Positioning and Indoor Navigation: IPIN2010, Zürich, CH 51
- [MF09] MAO, G.; FIDAN, B.: Localization Algorithms and Strategies for Wireless Sensor Networks.
 Hershey, PA : Information Science Reference Imprint of: IGI Publishing, 2009. ISBN 9781605663968 10, 51, 57
- [MH95] MURPHY, W. ; HEREMAN, W.: Determination of a Position in Three Dimensions using Trilateration and Approximate Distances. (1995). – Colorado, USA 36
- [MHO09] MARTINEZ, K.; HART, J. K.; ONG, R.: Deploying a Wireless Sensor Network in Iceland. In: Lecture Notes in Computer Science, Proc. Geosensor Networks (2009), Juli, 131-137. http://eprints.ecs.soton.ac.uk/17655/ 27

- [Mil] MILLENNIAL NET: Wireless Technology: Meshscape S. http://www.millennialnet. com/technology/. - Zuletzt aufgerufen: 31.01.2011 20
- [MLL⁺08] MESSINA, M.; LIM, Y.Y.; LAWRENCE, E.; DON, M.; KARGL, F.: Implementing and Validating an Environmental and Health Monitoring System. In: *ITNG '08: Proceedings of the Fifth International Conference on Information Technology: New Generations*. Washington, DC, USA : IEEE Computer Society, 2008. – ISBN 978–0–7695–3099–4, S. 994–999 28
- [MLRT04] MOORE, D. ; LEONARD, J. ; RUS, D. ; TELLER, S.: Robust Distributed Network Localization with Noisy Range Measurements. In: *Proceedings of the 2nd International Conference* on Embedded Networked Sensor Systems, 2004. – ISBN 1581138792, S. 50–61. – Baltimore, MD, USA 56
- [MOBK07] MAUTZ, R.; OCHIENG, W.Y.; BRODIN, G.; KEMP, A.: 3D Wireless Network Localization from Inconsistent Distance Observations. In: Ad Hoc & Sensor Wireless Networks (2007) 56
- [MPS⁺02] MAINWARING, A. ; POLASTRE, J. ; SZEWCZYK, R. ; CULLER, D. ; ANDERSON, J.: Wireless Sensor Networks for Habitat Monitoring. In: *Proceedings of the International Workshop* on Wireless Sensor Networks and Applications, 2002. – ISBN 1581135890, S. 88–97. – Atlanta, Georgia, USA 27
- [MS77] MEISSL, P. ; STUBENVOLL, K.: Ein Computer Programmsystem zur Verdichtung trigonometrischer Netze. Allgemeine Beschreibung und Benützeranleitung. In: *Mitteilungen der* geodätischen Institute der Technischen Universität Graz, Folge 25: Statistik, Fehlerrechnung, Methode der kleinsten Quadrate, Teststatistik (1977) 96
- [MZG⁺09] MORITZ, G. ; ZEEB, E. ; GOLATOWSKI, F. ; TIMMERMANN, D. ; STOLL, R.: Web Services to Improve Interoperability of Home Healthcare Devices. In: *Proceedings of the 3rd International Conference on Pervasive Computing Technologies for Healthcare 2009* ISCT, 2009. – ISBN 978–936–9799–42–4, S. 1–4 28
- [NBB07] NASH, E. ; BILL, R. ; BOBERT, J.: Anwendungsfallanalyse für den Einsatz von GDI-Technologien in Precision Farming. In: GIS - Zeitschrift für Geoinformatik, 2007, S. 12–19 27
- [NBB10] NIEMEYER, F.; BORN, A.; BILL, R.: Analysing the Accuracy of Resource Aware Localisation Algorithms for Wireless Geosensor Networks. In: Proceedings of Ninth International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences: Accuracy 2010, 2010, S. 45 – 49. – Leicester, UK 149
- [NC94] NISHIMURA, C.E.; CONLON, D.M.: IUSS dual use: Monitoring Whales and Earthquakes using SOSUS. In: *Mar. Technol. Soc. J.* 27 (1994), S. 13–21 10
- [Net10] NETWORKED AND EMBEDDED SYSTEMS LABORATORY: *The Ad-Hoc Localization System– AHLoS*. http://nesl.ee.ucla.edu/projects/ahlos/. Version:2010 63
- [NHF07] NIYATO, D. ; HOSSAIN, E. ; FALLAHI, A.: Sleep and Wakeup Strategies in Solar-Powered Wireless Sensor/Mesh Networks: Performance Analysis and Optimization. In: *Proceedings* of the IEEE Transactions on Mobile Computing (2007), S. 221–236 22
- [Nie08] NIEMEIER, W.: *Ausgleichsrechnung: Statistische Auswertemethoden*. Walter de Gruyter, 2008. ISBN 978–3–11–019055–7 31, 35, 37, 40, 41, 43, 44, 47, 113

[NL02]	NASIPURI, A. ; LI, K.: A Directionality Based Location Discovery Scheme for Wireless Sensor Networks. In: <i>Proceedings of the International Workshop on Wireless Sensor Networks and Applications</i> , 2002. – ISBN 1581135890, S. 105–111. – Atlanta, Georgia, USA 63
[NLF07]	NAKAMURA, E.F. ; LOUREIRO, A.F. ; FRERY, A.C.: Information fusion for wireless sensor networks: Methods, models, and classifications. In: <i>ACM Comput. Surv.</i> (2007), S. 9. – ISSN 0360–0300 26
[NLPV05]	NECCHI, L. ; LAVAGNO, L. ; PANDINI, D. ; VANZAGO, L.: An Ultra-low Energy Asynchronous Processor for Wireless Sensor Networks. In: <i>Proceedings of the 12th IEEE International</i> <i>Symposium on Asynchronous Circuits and Systems</i> , 2005. – ISBN 0769524982, S. 78 24
[NN01]	NICULESCU, D. ; NATH, B.: Ad hoc positioning system (APS). In: <i>Proceedings of the IEEE Global Telecommunications Conference</i> , 2001. – ISBN 0780372069, S. 2926–2931. – San Antonio, USA 55
[NN03]	NICULESCU, D. ; NATH, B.: Ad Hoc Positioning System (APS) using AOA. In: <i>IEEE Annual Joint Conference Computer and Communications Societies</i> , 2003, S. 1734–1743. – San Francisco, CA, USA 55
[Nor09]	NORRDINE, A.: <i>Präzise Positionierung und Orientierung innerhalb von Gebäuden</i> , TU Darmstadt, Diss., 2009. – Schriftenreihe Fachrichtung Geodäsie, Nr. 29 64
[OYM ⁺ 04]	ONG, K.G. ; YANG, X. ; MUKHERJEE, N. ; WANG, H. ; SURENDER, S. ; GRIMES, C.A.: A Wireless Sensor Network for Long-term Monitoring of Aquatic Environments: Design and Implementation. In: <i>Sensor Letters</i> 2 (2004), S. 48–57 18
[PBDT03]	PRIYANTHA, N.B. ; BALAKRISHNAN, H. ; DEMAINE, E. ; TELLER, S.: Anchor-Free Distributed Localization in Sensor Networks. In: <i>Technischer Report, TR-892</i> , 2003, S. 340–341 56
[PHC04]	POLASTRE, J. ; HILL, J. ; CULLER, D.: Versatile low power media access for wireless sensor networks. In: <i>SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems</i> . New York, NY, USA : ACM Press, 2004. – ISBN 1581138792, 95–107 21
[Rap02]	RAPPAPORT, T.S.: Wireless Communications - Principles and Practice (Prentice Hall Communications Engineering and Emerging Technologies). Prentice Hall International, 2002. – ISBN 9780130422323 77, 78, 79, 80, 81, 82, 84, 85
[RB06]	ROST, S. ; BALAKRISHNAN, H.: Memento: A Health Monitoring System for Wireless Sensor Networks. In: <i>Proceedings of the IEEE Communications Society Conference on</i> <i>Sensor, Mesh and Ad Hoc Communications and Networks</i> , 2006. – ISBN 1424406269, S. 575–584. – Reston, VA, USA 28
[RBN ⁺ 08]	REICHENBACH, F. ; BORN, A. ; NASH, E. ; STREHLOW, C. ; TIMMERMANN, D. ; BILL, R.: Improving Localization in Geosensor Networks Through Use of Sensor Measurement Data. In: <i>Proceedings of the 5th International Conference on Geographic Information</i> <i>Science (GiScience 2008)</i> , 2008, S. 261–273. – Park City, Utah, USA 68

[RBT04a] REICHENBACH, F.; BLUMENTHAL, J.; TIMMERMANN, D.: Drahtlose Sensornetzwerke in der Laborautomatisierung. In: GIT Laboratory IT User Service (LITUS) (2004), November, S. 20–21. – ISSN 16148622. – Darmstadt, Deutschland 28

- [RBT04b] REICHENBACH, F. ; BLUMENTHAL, J. ; TIMMERMANN, D.: Kostengünstige und präzise funkgestützte Lokalisierung in drahtlosen Sensornetzwerken. In: Landestechnologieanzeiger, 4/2004 (2004), Dezember, S. 19. – Schwerin, Deutschland 12
- [RBTB06] REICHENBACH, F.; BORN, A.; TIMMERMANN, D.; BILL, R.: A Distributed Linear Least Squares Method for Precise Localization with Low Complexity in Wireless Sensor Networks. In: 2nd IEEE International Conference on Distributed Computing in Sensor Systems (2006), S. 514–528. ISBN 03029743 36, 56, 122
- [Rei07] REICHENBACH, F.: Ressourcensparende Algorithmen zur exakten Lokalisierung in drahtlosen Sensornetzwerken, Universität Rostock, Diss., 2007 15, 16, 56, 57, 58, 62, 103, 111, 117, 136, 140, 156
- [RHT05] REICHENBACH, F.; HANDY, M.; TIMMERMANN, D.: Monitoring the Ocean Environment with Large-Area Wireless Sensor Networks. (2005), September, S. 57–58. ISBN 3902457090.
 – Porto, Portugal 27
- [Rie09] RIEGLER, T.: Akkus und Ladegeräte. Baden-Baden : Vth, 2009. ISBN 978–3881807852 22, 23
- [RKT06] REICHENBACH, R.; KOCH, M.; TIMMERMANN, D.: Closer to Reality Simulating Localization Algorithms Considering Defective Observations in Wireless Sensor Networks. (2006), März, S. 59–64. ISBN 3832248625. – Hannover, Deutschland 58, 59
- [RLT08] REICHENBACH, F. ; LIECKFELDT, D. ; TIMMERMANN, D.: Using QR-Updating with Reduced Complexity for Precise Localization in Mobile Sensor Networks. In: Proceedings of the Sixth Annual Conference on Communication Networks and Services Research (CNSR2008), 2008, S. 432–439 57
- [RSPS02] RAGHUNATHAN, V. ; SCHURGERS, C. ; PARK, S. ; SRIVASTAVA, M.: Energy-Aware Wireless Sensor Networks. In: *IEEE Signal Processing* 19 (2002), S. 40–50 25
- [RT05] REICHENBACH, F. ; TIMMERMANN, D.: Großflächige drahtlose Sensornetzwerke zur Überwachung der Meeresumwelt. April 2005 Vortrag, Seminar: Physikalische Ozeanographie, Leibniz-Institut für Ostseeforschung, Warnemünde, Deutschland 27
- [RT07] REICHENBACH, F. ; TIMMERMANN, D.: On Improving the Precision of Localization with Minimum Resource Allocation. In: 16th International Conference on Computer Communications and Networks, International Workshop on Wireless Mesh and Ad Hoc Networks, 2007. – ISBN 142441251X, S. 1093–1098. – Honolulu, Hawaii, USA 57
- [Sam01] SAMPLE, I.: Alternatives to landmines. In: *New Scientist* (2001), April. http://www.newscientist.com 29
- [Sch06] SCHULZ, J.: Entwicklung eines funkgestützten Sensorknotens zur Detektion der Feuchtigkeit in künstlichen Deichen. 2006. – Studienarbeit 28
- [SCV⁺06] SIKKA, P.; CORKE, P.; VALENCIA, P.; CROSSMAN, C.; SWAIN, D.; BISHOP-HURLEY, G.: Wireless Adhoc Sensor and Actuator Networks on the Farm. In: *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, 2006. – ISBN 1595933344, S. 492–499. – Nashville, Tennessee, USA 27

- [SHS01] SAVVIDES, A.; HAN, C.-C.; SRIVASTAVA, M.B.: Dynamic fine grained localization in ad-hoc networks of sensors. In: *Proceedings of the 7th ACM MobiCom* (2001), S. 166–179. – Rome, Italy 50, 54
- [SIG] SIG: Press Releases. http://www.bluetooth.com/Bluetooth/Press/ SIG/SIG_INTRODUCES_BLUETOOTH_LOW_ENERGY_WIRELESS_TECHNOLOGY_THE_ NEXT_GENERATION_OF_BLUETOOTH_WIRELESS_TE.htm. - Zuletzt aufgerufen: 31.01.2011 20
- [SN05] STEFANIDIS, A.; NITTEL, S.: GeoSensor Networks. CRC Press, 2005 9, 10, 51
- [SRB01] SAVARESE, C. ; RABAEY, J. ; BEUTEL, J.: Location in distributed ad-hoc wireless sensor networks. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001, S. 2037–2040 56
- [SRL02] SAVARESE, C. ; RABAEY, J. ; LANGENDOEN, K.: Robust Positioning Algorithms for Distributed Ad-hoc Wireless Sensor Networks. In: *Proceedings of the USENIX Annual Technical Conference*, 2002, S. 317–327 55, 56
- [SS02] SIMIC, S.; SASTRY, S.: Technischer Report, UC Berkeley, UCB/ERL M02/26. (2002) 54
- [Ste00] STEWART, G.W.: *The Decompositional Approach To Matrix Computation*. Computing in Science and Enginieering, 2000 40
- [Tay96] TAYLOR, J.R.: An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements. University Science Books, 1996 33
- [TBBC08] TARRIO, P. ; BERNARDOS, A.M. ; BESADA, J.A. ; CASAR, J.R.: A new positioning technique for RSS-Based localization based on a weighted least squares estimator. In: IEEE International Symposium on Wireless Communication Systems (ISWCS '08), 2008, S. 633–637 36
- [The] THE PERMASENSE CONSORTIUM: *PermaSense*. http://www.permasense.ch/. Zuletzt aufgerufen: 31.01.2011 27
- [Uni] UNIVERSITY OF SOUTHAMPTON: *Glacsweb.* http://envisense.org/glacsweb/ index.html. - Zuletzt aufgerufen: 31.01.2011 27
- [Vik] VIKSTRÖM, P.: WISA Wireless Interface for Sensors and Actuators in Industrial Applications. http://users.tkk.fi/virranko/sensor_networks/vikstrom2.pdf. - Zuletzt aufgerufen: 31.01.2011 20
- [WALW⁺06] WERNER-ALLEN, G. ; LORINCZ, K. ; WELSH, M. ; MARCILLO, O. ; JOHNSON, J. ; RUIZ, M. ; LEES, J.: Deploying a Wireless Sensor Network on an Active Volcano. In: *IEEE Internet Computing* (2006), S. 18–25. – ISSN 10897801 28
- [WG97] WOLF, P.R. ; GHILANI, C.D.: Adjustment Computations: Statistics and Least Squares in Surveying and GIS (Surveying & Boundary Control). Wiley-Interscience, 1997 31
- [Wika] WIKIPEDIA: Huygenssches Prinzip. http://de.wikipedia.org/wiki/ Huygenssches_Prinzip. – Zuletzt aufgerufen: 31.01.2011 79
- [Wikb] WIKIPEDIA: Sound Surveillance System. http://de.wikipedia.org/wiki/SOSUS. - Zuletzt aufgerufen: 31.01.2011 10

Litoraturvorz		hnic
LIIGIAIUIVEIZ	CIUI	11113

[WS]	WLAN STANDARDS, The Working G.: <i>The Working Group for WLAN Standards</i> . http://grouper.ieee.org/groups/802/11/ Zuletzt aufgerufen: 31.01.2011 20
[XHE01]	XU, Y. ; HEIDEMANN, J. ; ESTRIN, D.: Geography-informed Energy Conservation for Ad Hoc Routing. In: <i>ACM MOBICOM</i> , 2001, S. 70–84 18
[Yan06]	YANG, GZ. ; YANG, GZ. (Hrsg.): <i>Body Sensor Networks</i> . Springer, 2006. – ISBN 1846282721 29
[YOD ⁺ 02]	YANG, X. ; ONG, K.G. ; DRESCHEL, W.R. ; ZENG, K. ; MUNGLE, C.S. ; GRIMES, C.A.: Design of a Wireless Sensor Network for Long-term, In-Situ Monitoring of an Aqueous Environment. In: <i>Sensors</i> (2002), S. 455–472. – ISSN 14248220 27
[YR04]	YI, S. ; RUML, W.: Improved MDS-based Localization. (2004), S. 2640–2651. ISBN 0780383559 56
[Zel06]	ZELENKO, D.: Sensornetze / Smart-Dust. In: Seminar Drahtlose Netzwerke: Technologien, Anwendungen, Management (2006) 11
[ZHKS04]	ZHOU, G. ; HE, T. ; KRISHNAMURTHY, S. ; STANKOVIC, J.A.: Impact of Radio Irregularity on Wireless Sensor Networks. In: <i>Proceedings of the International Conference on Mobile</i> <i>Systems</i> , 2004, S. 23–25. – Boston, USA 59
[Zig]	ZIGBEE ALLIANCE: ZigBee. http://www.zigbee.org/ Zuletzt aufgerufen: 31.01.2011 20
[ZXL ⁺ 09]	ZHOU, Z.; XIAO, M.; LIU, L.; CHEN, Y.; LV, J.: An Improved DV-HOP Localization Algorithm. In: <i>Information Science and Engineering (ISISE), 2009 Second International</i> <i>Symposium on</i> (2009), dec., S. 598–602. http://dx.doi.org/10.1109/ISISE. 2009.79. – DOI 10.1109/ISISE.2009.79 62
Abbildungs- und Tabellenverzeichnis

Abbildungsverzeichnis

1.1	"Smart Dust"–Knoten	11
2.1 2.2	Sensorknoten schematisch	16 23
3.1 3.2 3.3	Vorwärts- und Rückwärtseinschneiden durch Winkelmessungen	32 33 35
 4.1 4.2 4.3 4.4 4.5 4.6 4.7 4.8 4.9 	Netzwerkarten	49 50 51 52 53 58 59 61 62
5.1 5.2 5.3 5.4 5.5	ACL–Prinzip	69 71 72 73
5.6 5.7 5.8 5.9 5.10 5.11 5.12 5.13	tervalle sowie Vergleich von <i>ACL</i> mit Mittelung und <i>ACL</i> mit erster gültiger Position Large Scale und Small Scale Fading	74 77 82 83 84 86 87 88
5.14 5.15	Lokalisierungsfehler über <i>PAF</i> Steigende Anzahl Beacons für <i>eACL</i>	89 90

5.16 5.17	Lokalisierungsfehler über steigender Beaconanzahl für eACL90Steigende Anzahl Sensorintervalle für eACL und Positionsfehler91
6.1	Komplexitätsvergleich und Kommunikationsaufwand
6.2	Energieverbräuche bei Anwendung von RAL 104
6.3	Energieverbrauch RAL im Vergleich I
6.4	Energieverbrauch RAL im Vergleich II
6.5	Clusterisierung eines Sensornetzwerkes
6.6	Anzahl Beobachtungen im Sensornetzwerk
6.7	Clusterisierung eines Sensornetzwerkes
6.8	Berechnungsaufwand und zeitliche Komplexität im Sensornetz
6.9	Approximierte Distanz bei Beacons außer Empfangsreichweite der Sensorknoten 112
6.10	Simulationsaufbau und RAL-Positionen
6.11	Sensorpositionen nach Ausgleich
6.12	Mittlerer Punktfehler
6.13	Anzahl Knoten ohne Verbindung zu Beacons und Energieverbrauch über steigender
	Beaconzahl
7.1	Bestimmung der optimalen Potenz g für WCL
7.2	WCL mit fehlerbehafteten Beaconkoordinaten Lund fehlerfreien Streckenbeobachtungen
	(Rasterverteilung)
7.3	WCL mit fehlerbehafteten Beaconkoordinaten II und fehlerfreien Streckenbeobachtungen
	(Rasterverteilung)
7.4	WCL mit fehlerbehafteten Beaconkoordinaten I und fehlerfreien Streckenbeobachtungen
	(zufällige Verteilung)
7.5	WCL mit fehlerbehafteten Beaconkoordinaten II und fehlerfreien Streckenbeobachtungen
	(zufällige Verteilung)
7.6	WCL mit fehlerbehafteten Streckenbeobachtungen und fehlerfreien Beaconkoordinaten I 131
7.7	WCL mit fehlerbehafteten Streckenbeobachtungen und fehlerfreien Beaconkoordinaten II132
7.8	WCL mit fehlerbehafteten Streckenbeobachtungen und fehlerbehafteten Beaconkoordi-
	naten (Rasterverteilung)
7.9	WCL mit fehlerbehafteten Streckenbeobachtungen und fehlerbehafteten Beaconkoordi-
	naten (zufällige Verteilung)
7.10	Vergleich ausgewählter Potenzen g für WCL 134
7.11	Bestimmung optimale Potenz g für WCL bei zufälligen Beaconkonfigurationen 136
7.12	DLS-Testfeld und Lokalisierungsfehler
7.13	DLS-Testfeld und Lokalisierungsfehler
7.14	Einfluss fehlerbehafteter Beaconpositionen und Distanzen auf RAL
7.15	Testfeld der Simulationen für den Vergleich ausgewählter Lokalisierungsalgorithmen 149
7.16	Berechnete Sensorknotenpositionen und Lokalisierungsfehler I
7.17	Berechnete Sensorknotenpositionen und Lokalisierungsfehler II
7.18	Fehlerellipsen der untersuchten Algorithmen 154
7.19	Positionsfehler und WCL-Gewicht für das gesamte Sensornetzwerk
7.20	Energie-Fehler-Protokoll

Tabellenverzeichnis

1.1	Parameter und Unterscheidungsmerkmale von Geosensornetzwerken	10
2.1	Bluetooth-Klassen	19
2.2	Beispiele aktueller Sensormodule	22
2.3	Energiespeicher und Energiedichten	23
2.4	Senorknotenaktivität und Energieverbrauch	24
4.1	Lokalisierungsalgorithmen	65
5.1	Pfadverlustexponenten nach Umgebungstyp	82
5.2	Simulationsparameter für <i>eACL</i>	85
5.3	Simulationsergebnisse zu <i>eACL</i> in Laborraum 2	87
5.4	Simulationsergebnisse zu <i>eACL</i> in Büroraum 1	88
5.5	Anzahl gültiger Positionen über ansteigende Sensorintervalle	92
6.1	RAL-Lösungsmethoden	98
6.2	Vergleich der Komplexitäten für die Methode der kleinsten Quadrate	101
6.3	Kommunikationsaufwand RAL	103
6.4	Simulationssetup RAL	103
6.5	Vergleich der Komplexitäten für Lokalisierungsverfahren	105
6.6	Simulationsaufbau Nachausgleichung	114
6.7	Ergebnisse der Nachausgleichung	117
6.8	Kommunikationsaufwand für die Nachausgleichung	118
7.1	Simulationsaufbau für WCL	128
7.2	Parameter der Simulationen für den Vergleich ausgewählter Lokalisierungsalgorithmen	150
7.3	Simulationsergebnisse für den Vergleich ausgewählter Lokalisierungsalgorithmen I	152
7.4	Simulationsergebnisse für den Vergleich ausgewählter Lokalisierungsalgorithmen II	154
7.5	Simulationssetup für das Energie-Fehler-Protokoll (EFP)	156
7.6	EFP für ausgewählte Lokalisierungsalgorithmen	157

Abkürzungs- und Symbolverzeichnis

Abkürzungsverzeichnis

ABC	Assumption Based Coordinates–Algorithmus
AccuLoc	Accuracy Analyzer for Localization in Wireless Sensor Networks
ACL	Anomaly Correction in Localization–Algorithmus
AoA	Angle of Arrival–Verfahren
APIT	Approximate Point in Triangulation-Test (auch Abkürzung für den Range-
	Free Localization–Algorithmus)
APS	Ad-Hoc Positioning System
BLUE	Best Linear Unbiased Estimator (engl. für Bester, Linearer, Unverzerrter Schätzer)
CC1010	Chipcon CC1010 Sensorknotenplattform
CGL	Coarse Grained Localization–Algorithmus oder Kategorie als approximativer oder grobkörniger Lokalisierungsalgorithmus
CL	Centroid Localization–Algorithmus
CRLB	Cramer-Rao Untergrenze (engl. Cramer-Rao Lower Bound)
DGM	Digitales Oberflächenmodell (engl. Digital Terrain Model)
DLS	Distributed Least Squares–Algorithmus (verteilter, feinkörniger Algorithmus)
DSMAC	Dynamic Sensor Media Access Control
DSN	drahtloses Sensornetzwerk (engl. Wireless Sensor Network (WSN))
DV-Hop	Distance-Vector-Hop
eACL	extended Anomaly Correction in Localization–Algorithmus
EM-Feld	Elektromagnetisches Feld. Eine wichtige Form eines EM-Feldes sind elek- tromagnetischen Wellen.
FGL	Fine-grained Localization-Algorithmus oder gleichnamige Kategorie als ex-
	akter oder feinkörniger Lokalisierungsalgorithmus
Flop	Floating Point Operation
FP	Fingerprinting
GHM	Gauss-Helmert-Modell
GMM	Gauss–Markov–Modell
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GSN	Geosensornetzwerk

iDLS	Iterative Distributed Least Squares–Algorithmus
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial Measurement Unit (engl. für Inertialmesssystem)
iMUL	iterative Multilateration–Algorithmus
LoS	Line-of-Sight (direkte Sichtverbindung zwischen Sender-Empfänger)
LSF	Large Scale Fading
MAC	Media Access Control
MCU	Mikrocontrollereinheit (engl. Microcontorller Unit)
MdkQ	Methode der kleinsten Quadrate (engl. Least Squares Method (LSM))
mDLS	Mobile Distributed Least Squares–Algorithmus
MDS	Multidimensionale Skalierung
MEMS	Micro-Electro-Mechanical Systems
МІТ	Massachusetts Institute of Technology
МТР	Minimal Transmission Power–Verfahren
NB	Nearest Beacon–Algorithmus
NLoS	Non–Line–of–Sight (keine Sichtverbindung zwischen Sender–Empfänger)
PAF	Path Attenuation Factor
PIT	Point in Triangulation-Test
PMT	Pattern Matching Technique
PR	Pattern Recognition
QRF	QR–Faktorisierung
RAL	Resource Aware Localization–Algorithmus
RAL+AGL	Resource Aware Localization–Algorithmus mit nachträglicher Gesamtnetz-
	ausgleichung
RSS	Signalempfangsstärke (engl. Received Signal Strength)
RSSI	Received Signal Strength Indicator
RTT	Round Trip Time-Verfahren
S-MAC	Sensor Media Access Control
SA	Szenenanalyse
sDLS	Scalable Distributed Least Squares-Algorithmus
SL	Standardlokalisierung
SOSUS	Sound Surveillance System
SRD	Short Range Devices
SSF	Small Scale Fading
SVD	Singular-Value-Decomposition
T–MAC	Timeout Media Access Control
TDC	Time-to-Digital-Converter
TDoA	Time Difference of Arrival–Verfahren
ТоА	Time of Arrival–Verfahren
TRAMA	Traffic-Adaptive Media Access Control Protokoll
UWB	Ultra-Wideband
VFG	Varianzfortpflanzungsgesetz

WCL	Weighted Centroid Localization–Algorithmus
WGS 84	World Geodetic System 1984
WISA	Wireless Interface for Sensors and Actuators
WLAN	Wireless Local Area Network bzw. IEEE 802.11

Symbolverzeichnis

α_i	Innenwinkel eines Dreiecks
$ar{P}_0(ar{x}_0,ar{y}_0)$	Näherungskoordinaten für Standardlokalisierung
r	Berechnete Distanz aus Näherungskoordinaten für Standardlokalisierung
β	Pfadverlustexponent
В	Jakobimatrix bei der bedingten Ausgleichung
k	Korrelatenvektor bei der bedingten Ausgleichung
w	Widerspruchsvektor bei der bedingten Ausgleichung
μ	Erwartungswertvektor
ε	Residuenvektor
Γ	Fresnel'scher Reflexionskoeffizient
Γ_{rough}	Modifizierter Fresnel'sche Reflexionskoeffizient
Î	Ausgeglichene Beobachtung
\hat{S}_i	Definition eines lokalen Koordinatensystems für den ABC-Algorithmus
ŵ	Hilfsunbekannte bei RAL
λ	Wellenlänge eines elektromagnetischen Signals
$\lambda_{1,2}$	Eigenwerte zur Halbachsenberechnung für die Fehlerellipsen
λ_{ii}	Singulärwerte bei der SVD
μ_k	Knotenverteilungsdichte
Ω	Verbesserungsquadratsumme
Ω_{Σ}	Gewichtete Verbesserungsquadratsumme
\overline{V}	Verbesserung der Positionsschätzung bei Mittelung der gültigen Positionen
	bei ACL
Φ	Fläche, die ein Sensornetzwerk einnimmt
$ ho_s$	Streuungsverlustfaktor
$ ho_{ij}$	Korrelationskoeffizient zwischen Beobachtungen i und j
σ	Standardabweichung
σ^2	Varianz
σ_b^2	Varianz der Beaconposition
σ_d	Standardabweichung der Strecke
σ_d^2	Streckenvarianz
σ_h	Standardabweichung der Höhe der Oberflächenrauheit
$\sigma_{0_aposteriori}$	Gewichtseinheit nach der Ausgleichung
$\sigma_{0_apriori}$	Gewichtseinheit vor der Ausgleichung
θ	Einfallswinkel einer elektromagnetischen Welle

$\tilde{B}(\tilde{x}_i; \tilde{y}_i)$	Beaconposition in der euklidischen Ebene
\tilde{d}_i, \hat{d}_i	Exakte und fehlerbehaftete euklidische Distanz
$\tilde{P}_i(\tilde{x}_i; \tilde{y}_i), \hat{P}_i(\hat{x}_i; \hat{y}_i)$	Exakte und fehlerbehaftete Koordinaten
φ	Richtungswinkel der großen Halbachse für die Fehlerellipse
A	Kugeloberfläche
a	Große Halbachse der Fehlerellipse
b	Kleine Halbachse der Fehlerellipse
С	Lichtgeschwindigkeit
$C_{\hat{u}\hat{u}}$	Kovarianzmatrix der geschätzten Unbekannten als Ergebnis der Netzplanung
Ε	Erwartungswert
E _{f,ACL}	Lokalisierungsfehler der ersten korrekten Position bei ACL
E_f	Lokalisierungsfehler der Erstposition bei ACL
EFP	Energie-Fehler-Produkt
8	Potenz bei der Gewichtsberechnung für WCL
h	Oberflächenrauheit
h_c	Kritische Höhe für die Struktur einer Oberfläche für das Rayleigh-Kriterium
I_0	Besselfunktion erster Art und nullter Ordnung
k	Sensorknotenanzahl im Netzwerk
1	Stochastische Beobachtung
m	Anzahl notwendiger Beobachtung bei der bedingten Ausgleichung
п	Anzahl der Gleichungen des Gleichungssystems
$P(x_i; y_i)$	Referenzpunkt in der euklidischen Ebene
p _{ij}	Gewichte in der Gewichtsmatrix P
$P_N(x_N;y_N)$	Neupunkt in der euklidischen Ebene
r	Redundanz des Ausgleichungsproblems
R_1	R-Matrix bei der QRF
s _{ij}	Distanz von Punkt P_i nach P_j
t_i^j	Richtungswinkel von Punkt i nach j
U	Anzahl der Unbekannten des Gleichungssystems
υ	Ausbreitungsgeschwindigkeit eines Ultraschallsignals in Luft
v_i	Verbesserungvektor der Beobachtung b_i
V_f	Verbesserung der Positionsschätzung bei erster gültiger Position bei ACL
w_j	Gewicht bei WCL
<i>x</i> ₀	Startwert für die Taylorlinearisierung
ň _i	Reduzierte rechte Seite des Ausgleichungsproblems
х _і	Reduzierter Lösungsvektor des Ausgleichungsproblems
A _p	Matrix mit der Vorberechnung bei RAL
Α	Koeffizientenmatrix bei der Lösung von Gleichungssystemen
b	Absolutglied des Gleichungssystems
C _{FF}	Kovarianzmatrix als Ergebnis der Varianzfortpflanzung
C _{ll}	Kovarianzmatrix des Zufallsvektors
C _{xx}	Kovarianzmatrix der Unbekannten für die Varianzfortpflanzung

F	Finhoitematrix
E	
F	Funktionalmatrix für die Varianzfortpflanzung
Ν	Normalgleichungsmatrix
n	Rechte Seite des Normalgleichungssystems
Р	Gewichtsmatrix für die Netzausgleichung
Q ₁₁	Kofaktormatrix der Beobachtungen
Q	Q-Matrix bei der QRF
Q _{x̂x}	Kofaktormatrix der Unbekannten
S ₁	S-Matrix bei der SVD
U	U-Matrix bei der SVD
V	V-Matrix bei der SVD
v	Verbesserungsvektor der Beobachtungen
x	Lösungsvektor des Gleichungssystems
у	y–Vektor für Zwischenberechnung bei der SVD
Z	z–Vektor für Zwischenberechnung bei der QRF
K	Vektor mit Beaconkoordinaten bei RAL
Ň ₁₁	Reduzierte Normalgleichungsmatrix des Ausgleichungsproblems

Danksagung

Diese Arbeit widme ich meinen Eltern und meiner Freundin. Ohne ihren Rückhalt wäre sie nicht zustande gekommen.

Für die Bereitstellung des Themas und die Möglichkeit, an der Professur für Geodäsie und Geoinformatik wissenschaftlich tätig zu werden, habe ich Prof. Ralf Bill sehr zu danken. Insbesondere seine stete Bereitschaft, mir beratend und mit viel Geduld zur Seite zu stehen, hat mir die Bearbeitung der Problemfelder sehr erleichtert. Großer Dank gebührt ebenfalls Herrn Prof. Dirk Timmermann für die Schaffung des interdisziplinären Umfeldes im Rahmen der DFG-Projekte Geosens und Geosens 2. Weiterhin bedanke ich mich bei Prof. Lothar Gründig für einige fruchtbare Ideen und Anmerkungen.

Einen ganz großen Beitrag zu meiner wissenschaftlichen Arbeit hat als Freund und Kollege Frank Niemeyer geleistet. Ohne seine Hilfe und Unterstützung sowie den fachlichen Diskussionen wären große Teile dieser Arbeit und insbesondere Kapitel 7 wohl nicht möglich gewesen. Großer Dank gebietet ebenfalls meinem guten Freund und ehemaligen Kollegen Dr. Frank Reichenbach, der mich an die wissenschaftliche Arbeit und die Thematik der drahtlosen Sensornetzwerke heranführte und mir die Grundzüge der Elektrotechnik nahe brachte. Ebenfalls möchte ich mich bei Andreas Fink bedanken. Diskussionen mit ihm führten im Wesentlichen zu Kapitel 5.

Für das freundschaftliche Umfeld an der Professur für Geodäsie und Geoinformatik der Universität Rostock bedanke ich mich bei allen Kollegen.