

**Aktive Schall-Lokalisation in Teams autonomer, mobiler
Roboter und Ansätze zur kooperativen Selbstlokalisierung**

Diplomarbeit

im Studiengang Informatik
angefertigt an der Fachgruppe
Simulation, Systemoptimierung und Robotik
am Fachbereich Informatik
der Technischen Universität Darmstadt

von

David Becker

Darmstadt, März 2008

Betreuer: Prof. Dr. O. v. Stryk, Dipl.-Inf. Max Risler

Diese Arbeit behandelt die aktive Schall-Lokalisation innerhalb eines Teams von Robotern in einer akustisch verrauschten Umgebung. Dabei wird versucht, mit Hilfe von ausgesendeten *akustischen* Wellenformen Distanzen und Richtungen zwischen Robotern zu messen. Hierfür werden drei Ansätze theoretisch diskutiert. Einer der Ansätze verwendet Kreuzkorrelationsfunktionen in Verbindung mit modulierten, binären Codefolgen. Dieser Ansatz wird in der vorliegenden Arbeit ausführlich für den praktischen Einsatz auf autonomen Robotern diskutiert. Die verschiedenen Ansätze wurden außerdem auf dem SONY Aibo Roboter implementiert und getestet. Die Diskussion erfolgt auch vor dem Hintergrund eines RoboCup-Fußballspiels.

Zusätzlich werden in dieser Arbeit drei Ansätze theoretisch vorgestellt, wie mit Hilfe von relativen Positionsmessungen in einem Team von Robotern die Selbstlokalisierung einzelner Roboter verbessert werden kann.

Inhaltsverzeichnis

1	Einführung	1
1.1	Aktive Schall-Lokalisation	1
1.1.1	Motivation	1
1.1.2	Ansätze zur aktiven Lokalisation	2
1.2	Kooperative Selbstlokalisierung	3
1.2.1	Motivation	3
1.2.2	Anwendungsbeispiel	4
2	Stand der Forschung	5
2.1	Akustische Lokalisation	5
2.2	Kooperative Selbstlokalisierung	8
3	Aktive Schall-Lokalisation	11
3.1	Begriffsklärungen	11
3.2	Schall-Lokalisation beim Menschen	12
3.3	Grundlagen der Schall-Lokalisation	14
3.3.1	Richtungsmessung	14
3.3.2	Distanzmessung	18
3.4	Erster Ansatz: Fourier-Transformation	18

3.4.1	Diskrete Fourier-Transformation	18
3.4.2	Eigenschaften der Spektralzerlegung	19
3.4.3	Lokalisation	20
3.4.4	Reflexion, Brechung und Interferenz	23
3.4.5	Einfluss von Störgeräuschen	25
3.4.6	Zusammenfassung	26
3.5	Zweiter Ansatz: Verwendung eines digitalen Bandpassfilters	27
3.5.1	Kurze Einführung	27
3.5.2	Lokalisation	27
3.5.3	Entwurf des Bandpassfilters	27
3.5.4	Filtereigenschaften	28
3.5.5	Probleme des Ansatzes	30
3.5.6	Zusammenfassung	31
3.6	Dritter Ansatz: Kreuzkorrelation	31
3.6.1	Kurze Einführung	31
3.6.2	Begriffsklärungen	32
3.6.3	Kreuz- und Autokorrelation: Definitionen	33
3.6.4	Kreuzkorrelation bei GPS	34
3.6.5	Modulation von Codefolgen	35
3.6.6	Auswahl und Erzeugung der Codefolgen	39
3.6.7	Auswahl der Grundfrequenz	45
3.6.8	Bewegungsunschärfe	47
3.6.9	Signalvorverarbeitung	48
3.6.10	Reduktion des Berechnungsaufwands	49
3.7	Aspekte und Probleme in der Praxis	50
3.7.1	Quantisierung, digitale Abtastung	50
3.7.2	Uhrendrift und Synchronisierung	50
3.7.3	Beeinträchtigung der Umgebung	53

4	Ansätze zur kooperativen Selbstlokalisierung	54
4.1	Einleitung	54
4.1.1	Beschränkung auf ein bestimmtes Szenario	55
4.1.2	Überblick über die verwendeten Ansätze	56
4.2	Paarweise Auswertung der Messungen	56
4.2.1	Bewertung von Partikeln	58
4.2.2	Einstreuung neuer Partikel	58
4.3	Vermittelnde Ausgleichung	58
4.3.1	Gesamtausgleichung	58
4.3.2	Untergruppen als Positionsreferenzen	60
5	Implementierung und Ergebnisse der Schall-Lokalisation	68
5.1	Übersicht	68
5.2	Verwendete Plattform	68
5.2.1	Plattformspezifische Geometrie	68
5.2.2	Plattformspezifische Eigenschaften der digitalen Klangverarbeitung	71
5.3	Einsatzszenario	71
5.4	Auswahl der Codefolgen	72
5.5	Auswahl der Grundfrequenz	75
5.6	Experiment: Brechung an Hindernissen	76
5.7	Auswahl der Methode der Modulation	77
5.8	Uhrendrift und Synchronisierung	79
5.9	Lautstärkeverhältnisse	81
5.10	Berechnungsaufwand	82
5.11	Abwechselndes Aussenden von Codes	83

5.12	Einbindung in den GermanTeam-Code	87
5.13	Experimente zur Distanzmessung	91
5.13.1	Distanzmessung ohne Synchronisation	91
5.13.2	Distanzmessung mit Synchronisation	92
5.13.3	Ergebnisse einer Spielsituation im Roboterfußball	94
5.14	Experiment zur Richtungsmessung	96
6	Fazit	98
7	Weitere Arbeit	100
7.1	Schall-Lokalisation	100
7.2	Kooperative Selbstlokalisierung	101
A	Fehler bei der Richtungsmessung	a
B	Bewertungsfunktionen im Partikelfilter	b
C	Verwendete Abkürzungen	c
D	Verwendete Codefolgen	d
E	Pseudocode der FFT	e

Kapitel 1

Einführung

1.1 Aktive Schall-Lokalisation

1.1.1 Motivation

Die aktive, akustische Lokalisation von autonomen Robotern untereinander bietet einige Vorteile gegenüber der *visuellen* Ortung. Ein bedeutender Vorteil ist die Möglichkeit *fortlaufender* Messungen. Die Roboter können sich auch dann akustisch lokalisieren, wenn zwischen ihnen keine Sichtverbindung besteht. Dieser Fall tritt in einer Gruppe autonomer Roboter häufig auf, nämlich wenn Roboter durch Hindernisse, wie z.B. weitere Roboter, verdeckt werden. Außerdem nimmt ein gewöhnliches Raummikrophon Signale aus allen Richtungen auf, während ein omnidirektionales Sehen im Allgemeinen nicht möglich oder nur aufwendig zu realisieren ist.

Ein weiterer, wichtiger Vorteil liegt darin, dass sich die Roboter gegenseitig akustisch *identifizieren* können. Eine Richtungs- oder Distanzmessung kann also stets einem bestimmten Roboter der Gruppe zugeordnet werden. Eine visuelle Identifikation hingegen kann schwierig oder sogar unmöglich sein, wenn sich die Roboter der Gruppe ihrem Aussehen nach kaum oder gar nicht unterscheiden. Dasselbe gilt für die Verwendung von Laserscannern.

Ein weiterer Vorteil ist der Preis: Die Verwendung z.B. von Laserscannern ist sehr kostspielig. Die Verwendung akustischer Aktuatoren und akustischer Sensoren ist hingegen *sehr kostengünstig*. Weder Lautsprecher noch Mikrofone müssen für die hier erforschten Ansätze von besonderer Qualität sein. Lautsprecher und

Mikrophone des SONY Aibo, der als Plattform für Implementierung und Tests diente, sind von sehr einfacher Qualität bei einem Geldwert von schätzungsweise unter 10 Euro. Die benötigte Wiedergabeleistung der Lautsprecher hängt von den maximal zu messenden Distanzen und vom Umgebungsrauschen bzw. von Störsignalen ab.

Die Auswertung akustischer Signale ist verglichen mit der Auswertung von (Stereo-)Kamerabildern einfacher zu realisieren und erfordert normalerweise deutlich *weniger Rechenoperationen*.

1.1.2 Ansätze zur aktiven Lokalisation

Es sollen hier kurz die verschiedenen Ansätze zur aktiven Schall-Lokalisation, welche im Rahmen dieser Arbeit verfolgt wurden, vorgestellt werden.

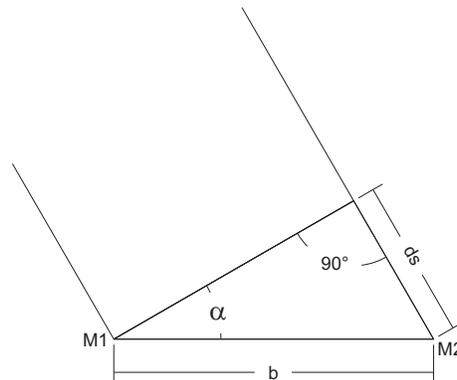


Abbildung 1.1: *Richtungsmessung durch Wegdifferenz ds (Sicht von oben).*

1. Richtungs- und Distanzmessung durch Berechnung der Phase eines reinen Sinustons: Der Sender sendet durchgehend einen reinen Sinuston einer bekannten Frequenz.

Der Empfänger bestimmt die Ursprungsrichtung α des ausgesendeten Signals anhand der Wegdifferenz ds zwischen Schallquelle und den beiden Empfangsmikrofonen $M1$ und $M2$ (siehe Abbildung 1.1). Die Wegdifferenz ds wird dabei durch den *Phasenunterschied* des Sinustons in beiden Eingangssignalen berechnet. Die Bestimmung der Phase des eintreffenden Sinustons erfolgt mit Hilfe einer Fourier-Transformation vom Zeit-/Amplitudenraum in den Frequenzraum.

Durch Integration der *Phasenänderung* kann zudem eine relative *Distanzänderung* berechnet werden. Die *absolute* Distanz kann nur dann berechnet werden, falls zu Beginn ein Initialisierungswert für die Distanz bekannt ist.

2. Richtungs- und Distanzmessung durch Bandpass-Frequenzfilterung:

Der Sender sendet sehr kurze Sinuspulse einer bekannten Frequenz f_0 , gefolgt von ausreichend langen Pausen. Der Empfänger filtert die Eingangssignale im Frequenzraum möglichst dicht um f_0 . Hierzu wird ein nicht-rekursiver Bandpass-Filter verwendet. Der Empfänger versucht dann im gefilterten Eingangssignal das kurze Sinusschema des Senders zu finden.

Aus der Differenz der Empfangszeitpunkte in den beiden Eingangssignalen kann wiederum ds , und damit die Ursprungsrichtung bestimmt werden (Abbildung 1.1).

Die *Distanz* zum Sender kann aus dem Empfangszeitpunkt bestimmt werden, falls dem Empfänger der genaue *Absendezeitpunkt* a priori bekannt ist. Hierfür ist eine exakte Synchronisation zwischen Sender- und Empfängeruhr erforderlich.

3. Richtungs- und Distanzmessung durch Kreuzkorrelationsfunktionen:

Ein Roboter sendet fortlaufend ein langes Sinuswellen-*Schema*, das einer modulierten binären Codefolge entspricht. Der empfangende Roboter berechnet die Kreuzkorrelationsfunktion zwischen dem Eingangssignal und dem bekannten Schema. Die Kreuzkorrelationsfunktion liefert für ein bestimmtes Zeit-Offset einen Wert für die *Ähnlichkeit* zwischen Eingangssignal und dem bekannten Schema. Der Zeit-Offset, an dem die Kreuzkorrelationsfunktion maximal wird, bestimmt dann den Empfangszeitpunkt.

Aus der Differenz der Empfangszeitpunkte der beiden Eingangssignale kann wiederum die Richtung bestimmt werden.

Zur Distanzmessung müssen die Absendezeitpunkte des Schemas *nicht* bekannt sein, falls jeder Roboter gleichzeitig Sender *und* Empfänger ist. Die Zeitdifferenz entspricht dabei der Hälfte einer Umlaufzeit, d.h. der Zeit, die das Signal von A nach B und zurück nach A benötigt (siehe Kapitel 3.7.2.2. Auch eine zusätzliche Initialisierung ist dann nicht erforderlich.

1.2 Kooperative Selbstlokalisierung

1.2.1 Motivation

Selbstlokalisierung ist die Bestimmung der eigenen, absoluten Position und Orientierung im Raum. Eine genaue und zuverlässige Selbstlokalisierung ist unabdingbar für fast alle Anwendungsszenarien autonomer Roboter. Das gilt zum Beispiel

für Assistenzroboter, die innerhalb von bekannten Umgebungen tätig sind, für autonome Roboter im freien Gelände oder auch für den speziellen Fall fußballspielender Roboter. Die Selbstlokalisierung erfolgt normalerweise durch das sensorische Wahrnehmen von Landmarken mit bekannter Position, oder auch durch externe Systeme wie z.B. GPS.

Unter *kooperativer* Selbstlokalisierung wird die Möglichkeit verstanden, dass sich *mehrere* Roboter einer Gruppe gegenseitig bei der Bestimmung ihrer Position und Orientierung unterstützen. Zu der *absoluten* Positionsschätzung, die jeder Roboter unabhängig durchführt, kommen die gemessenen *relativen* Positionen der Roboter untereinander hinzu.

Die eigentliche Kooperation besteht darin, dass sich die Roboter untereinander ihre Positionsschätzungen, sowie Positionskonfidenzen und Robotersichtungen mitteilen. Diese Nachrichten müssen über einen *separaten* Kommunikationskanal, wie z.B. eine drahtlose Funkverbindung, geschickt werden.

Sind relative Positionen untereinander bekannt, so entsteht auf das Gesamtsystem aller Roboterpositionen ein Zwang. Die relative Positionsinformation wird im Allgemeinen nämlich im Widerspruch zu den geschätzten Positionen stehen. Dies kann genutzt werden, um schlechte Positionsschätzungen zu erkennen und ggf. zu korrigieren.

1.2.2 Anwendungsbeispiel

Für ein spezielles Anwendungsbeispiel betrachten wir das Szenario eines Robo-Cup Roboter-Fußballspiels. Die Roboter lokalisieren sich zunächst nur durch die Sichtungen von Landmarken und Feldlinien. Wenn ein Spieler im Besitz des Balls ist, kann dieser nicht auf dem Spielfeld umher sehen. Die Sicht konzentriert sich stattdessen auf den Ball. Dadurch werden oft über längere Zeit wenige oder keine Landmarken gesichtet, wodurch dieser Roboter sich nur noch ungenau und unzuverlässig lokalisieren kann.

Angenommen, es stehen zwei gut lokalisierte Roboter in hörbarer Reichweite. Der bedrängte Roboter kann dann seine absolute Position auf dem Spielfeld aus den Distanz- und Richtungsmessungen zu diesen beiden Robotern in Verbindung mit deren geschätzten Positionen ausrechnen. Er kann sich sozusagen ‘blind’ lokalisieren.

In der Praxis entstehen Probleme vor allem dadurch, dass a priori nicht klar ist, *welche* der Roboter gut lokalisiert sind. Die Berechnung eines Konfidenzwertes für die eigene Position ist im Allgemeinen schwierig und ungenau.

Kapitel 2

Stand der Forschung

2.1 Akustische Lokalisation

Akustik spielt in der Robotik bisher eher eine geringe Rolle. Recht verbreitet sind Ultraschallsensoren, die nach dem Vorbild etwa der Fledermäuse oder Delphine zur Erkennung von Hindernissen eingesetzt werden. Das Prinzip entspricht dabei dem eines Echo-Lots. Die Zeit, die ein Ultraschallpuls benötigt, um vom Sender zum Hindernis und wieder zurück zu gelangen, dient als Basis für eine Distanzschätzung.

Das binaurale Richtungshören, d.h. das Richtungshören mit zwei Ohren, wurde für verschiedene Einsatzzwecke in der Robotik dem biologischen Vorbild nachgeahmt. Die Richtungsortung erfolgt durch Messung des Zeit- bzw Phasenversatzes, sowie des Pegelunterschieds eines Signals an beiden Ohren. Der Vorgang wird in Kapitel 3.2 kurz erläutert. Das binaurale Richtungshören *beim Menschen* wird in [1] sehr ausführlich untersucht. Bei der Richtungsortung beim Menschen gibt es danach zusätzlich zum oben Genannten weitere, kompliziertere Verfahrensweisen auf neuronaler Ebene. Das Richtungshören eines Menschen basiert demnach auf einem individuellen, komplexen Lernvorgang und ist insbesondere abhängig von der individuellen Form der Ohrmuschel.

Ein zuverlässiges Richtungshören in verrauschten Umgebungen ist auf mobilen Robotern schwierig zu implementieren. Steht nur ein Paar Mikrophone zur Verfügung, kann eine Schallrichtung in der Ebene lediglich zweideutig ermittelt werden. Es gibt bereits Verfahren, die Richtung robust mit Hilfe einer größeren Anzahl von Mikrophonen (so genannte Mikrophon-Arrays) zu bestimmen. In [2] wird ein solcher Ansatz unter Verwendung von 8 Mikrophonen vorgestellt. Durch

die große Überbestimmung an Eingangssignalen kann die Genauigkeit von Richtungsmessungen auch im dreidimensionalen Raum erheblich verbessert werden. Die Technik kommt bereits in der Praxis zum Einsatz, unter anderem um undichte Stellen einer Schallisolierung, z.B. an Fahrzeugen oder Bauwerken, zu finden. Die vorliegende Arbeit behandelt aufgrund der zur Verfügung stehenden Plattform nur das Richtungshören mit *zwei* Mikrofonen in der Ebene. Die Genauigkeit dieser Richtungsmessungen ist durch Reflexionen und Interferenzen stark begrenzt. Es liegt hierbei keine Überbestimmung vor, so dass ungenaue Messungen nicht erkannt werden können. Probabilistische Methoden sind notwendig, um die Messungen robust zu filtern.

Im Rahmen der mobilen Robotik gab es bereits Ansätze, Schall zur *Informationsübertragung* zu verwenden. In [3] wird gezeigt, wie anhand von Sinuspulsen verschiedener Frequenzen eine Informationsübertragung erfolgen kann. Der Empfänger filtert die Töne bekannter Frequenz mittels einer Fourier-Transformation aus dem üblicherweise sehr verrauschten Eingangssignal heraus. Die Information kann hierbei entweder in der Frequenz oder in der zeitlichen Abfolge von Pulsen codiert sein. Die Übertragung ist dabei durch starke Störgeräusche recht unzuverlässig. Die Technik ermöglicht außerdem nur geringe Übertragungsraten. In [4] wird ein ähnlicher Ansatz verwendet.

Seit dem Durchbruch drahtloser Netzwerke, und der damit verbundenen Entwicklung sehr kleiner und sparsamer Funknetzwerk-Adapter, spielt die akustische Informationsübertragung im Prinzip keine Rolle mehr für die mobile Robotik. Die akustische *Messung* von Distanz und Richtung ist nach wie vor von Interesse. Ein entsprechendes System, das auf *elektromagnetischen* Wellen basiert, ist aufgrund der viel höheren Ausbreitungsgeschwindigkeit elektromagnetischer Wellen deutlich aufwendiger umzusetzen. Empfangszeitpunkte müssen um den Faktor 10^6 genauer gemessen werden. Dies erfordert im Allgemeinen kostspielige Spezialhardware.

Das aktive Richtungshören auf einem mobilen Roboter kann recht einfach unter Verwendung reiner Sinustöne und der Fourier-Transformation implementiert werden. Dieser Ansatz wird in Kapitel 3.4 behandelt.

In [5] wird ein Verfahren vorgestellt, wie eine mobile Empfangseinheit Distanzen zu einer Basis messen kann. Dabei wird von der Basis ein moduliertes, akustisches Codeschema ausgesandt, dessen genauer Eintreffzeitpunkt durch den mobilen Empfänger durch Kreuzkorrelation festgestellt wird. Zusätzlich sind dem Empfänger die genauen Absendezeitpunkte bekannt, so dass dieser direkt die

Distanz zur Basis bestimmen kann. Der Empfänger wurde dann in einem Experiment an verschiedenen Distanzen aufgestellt und die Genauigkeit der gemessenen Distanzen untersucht.

Der dritte hier vorgestellte Ansatz zur akustischen Lokalisation basiert auf demselben Prinzip wie dem in [5] vorgestellten Ansatz (siehe Kapitel 3.6). Der Hauptteil der vorliegenden Arbeit beschäftigt sich mit der Entwicklung eines Konzepts, wie solche Distanzmessungen in einem *Team autonomer und mobiler Roboter* durchgeführt werden können. Dadurch, dass die Roboter mobil sind, ergeben sich durch die relativen Bewegungen besondere Anforderungen. In einem Team autonomer Roboter entsteht außerdem das Problem nicht synchron laufender Uhren. In der vorliegenden Arbeit werden daher Möglichkeiten einer Uhrensynchronisation untersucht. Außerdem werden sehr ausführlich weitere praktische Aspekte wie laute Störgeräusche der Umgebung sowie die durch Brechung und Reflexionen auftretenden Effekte diskutiert.

Dieser dritte Ansatz kann als Versuch gesehen werden, ein so genanntes CDMA-System auf Basis akustischer Wellen aufzubauen. CDMA-Systeme (engl. ‘*code division multiple access*’) ermöglichen allgemein die *gleichzeitige* Kommunikation mehrerer Sender und Empfänger über einen einzigen Übertragungskanal, d.h. auf nur einer Übertragungsfrequenz. In einem so genannten WCDMA-System (engl. ‘*Wide-CDMA*’) werden zwar mehrere Frequenzen verwendet, die aber jeweils von mehreren Sendern gleichzeitig benutzt werden, siehe [6].

Es gibt bereits sehr viele CDMA-Kommunikationssysteme auf Basis elektromagnetischer Wellen. Allgemeine Beschreibungen zu solchen Systemen sind in [7] sowie in [8] zu finden. Bekannte Vertreter solcher Systeme sind z.B. GPS [9], Bluetooth [10], UMTS [11] und WiFi. Drahtlose Kommunikationsnetzwerke wie z.B. UMTS oder Bluetooth werden zur Übertragung von Daten verwendet. Die in dieser Arbeit behandelte Ansatz ist vorrangig durch GPS inspiriert, da hier die Messung von Distanzen im Vordergrund steht. Allen diesen Systemen gemeinsam ist die Verwendung von Kreuzkorrelationsfunktionen zur Identifikation eines bekannten Wellenmusters in einem verrauschten Eingangssignal.

Für ein CDMA-System ist es erforderlich, dass jeder Sender oder jeder Empfänger (je nach Anwendung) eine eindeutige binäre Codefolge zugewiesen bekommt. Die *Erzeugung* von Mengen solcher Codefolgen, die für die Verwendung in einem CDMA-System geeignet sind, wird bereits seit den 1960er Jahren ausführlich erforscht. Aufgrund der hohen Bedeutung von CDMA-Systemen heute erscheinen zu diesem Thema nach wie vor viele Arbeiten, z.B. [12]. Zwei ältere Vorschläge zur Erzeugung solcher Codemengen, die in dieser Arbeit verwendet werden, stammen von Gold [13] und Kasami [14].

In [15] wird ein Verfahren vorgestellt, bei dem die Position einer beweglichen Schallquelle mit Hilfe eines Partikelfilters verfolgt wird. Dadurch sollen Messungenauigkeiten durch akustisch verhaltene Umgebungen gefiltert werden. Die Schwierigkeit besteht darin, dass besonders bei Verwendung hoher Frequenzen *Echos* des ursprünglichen Signals mit höherer Signalstärke empfangen werden können als das nicht-reflektierte Signal. Die Grundidee hinter dem vorgestellten Ansatz ist es, dass sich Echoanteile im Signal über die Zeit weniger konstant verhalten als das direkte Signal eines sich linear bewegenden Senders. Die Peaks zu solchen Echoanteilen entstehen und verschwinden eher sporadisch, während sich die Peaks des direkten Signals entsprechend der Bewegung des Senders *linearer* verhalten. Der vorgeschlagene Partikelfilter stellt eine Möglichkeit dar, mit wenig Rechenaufwand aus den Messungen über längere Zeiträume eine (partielle) Wahrscheinlichkeitsverteilung zu berechnen. In der vorliegenden Arbeit wird ein solcher Ansatz noch nicht verwendet. Für das Distanz-Tracking könnte im Rahmen weiterer Arbeit ein ähnlicher, eindimensionaler Partikelfilter untersucht werden.

2.2 Kooperative Selbstlokalisierung

Die Kooperative Selbstlokalisierung ist bereits verschiedenartig untersucht worden.

In [16] wird ein Verfahren vorgestellt, bei dem die *relativen* Positionen der anderen Roboter des Teams mit einem probabilistischen Filter geschätzt werden. Die Idee ist, dass jeder Roboter nur die in der näheren Umgebung befindlichen Roboter lokalisieren kann. Die Roboter erkennen und identifizieren sich dabei visuell unter Verwendung von Kameras und farbigen Marken. Alle Roboter tauschen dann ihre Messungen aus und berechnen anhand dieser ein probabilistisches globales Modell der relativen Lagen aller Roboter zueinander, inclusive eines stochastischen Modells in Form von Kovarianz-Ellipsen. Dieser Ansatz könnte auch auf Basis der hier untersuchten akustischen Lokalisation Anwendung finden.

In der vorliegenden Arbeit wird ein Szenario untersucht, bei dem vor allem die *absoluten* Positionen der Roboter von Interesse sind. Es werden hier daher geschätzte, *absolute* Positionen unter dem Systemzwang relativer Positionsmessungen ausgeglichen. Die in [16] vorgestellte probabilistische Berechnung eines vollständigen, relativen Positionsmodells könnte mit anschließender Ausgleichung aller absoluten Positionen eine Erweiterung zu den hier vorgestellten Ansätzen darstellen.

In [17] wird ebenfalls die Berechnung eines globalen, relativen Positionsmodells, allerdings unter Verwendung von Fuzzy-Logic gezeigt.

In [18] wird ausführlich ein Ansatz einer probabilistischen Selbstlokalisierung gegeben. In diesem Ansatz besitzt jeder Roboter lediglich ein Modell für die *eigene* Position und Konfidenz, berechnet durch einen partikelbasierten Markov-Lokalisierer. Die Roboter lokalisieren sich zunächst unabhängig anhand von Landmarkensichtungen. Die Roboter besitzen *keine* Annahmen über die Positionen und Konfidenzen der anderen Roboter des Teams. Sobald ein anderer Roboter gesichtet wird, tauschen die beiden Roboter ihre eigenen Positionsschätzungen aus. Anhand der relativen Positionsmessung, z.B. die Distanz zwischen beiden Robotern, aktualisieren dann beide Roboter die eigene Positionsschätzung. Als Folge daraus verbessern sich die Positionskonfidenzen erheblich, ähnlich wie bei der Sichtung einer Landmarke. Der Ansatz sieht also immer dann eine *paarweisen* Kooperation vor, wenn ein Roboter durch einen anderen gesehen wird. Die Funktionsweise wird anhand von Experimenten belegt. Die Arbeit beschreibt außerdem ein Beispielszenario eines Teams von Robotern mit *heterogener Sensorik*, in dem die Kooperation die Selbstlokalisierung überhaupt erst ermöglicht.

Die vorliegende Arbeit stellt unter anderem auch einen Ansatz einer paarweisen Kooperation vor. Dabei werden die Partikel des Monte-Carlo-Filters paarweise anhand einer Robotersichtung bewertet. In einem Roboter-Fußballspiel geschieht es recht häufig, dass ein Roboter völlig falsch lokalisiert ist, z.B. wenn er geschoben, blockiert oder manuell umgesetzt wurde. Der Monte-Carlo-Partikelfilter benötigt normalerweise einige Landmarkensichtungen, um dies auch zu erkennen. Für die paarweise Kooperation kann dies bedeuten, dass die Sichtung eines völlig falsch lokalisierten Roboters die Positionsgenauigkeit eines zunächst gut lokalisierter Roboters *verschlechtert*. Die vorliegende Arbeit versucht für dieses spezielle Szenario Ansätze für robustere Methoden zu entwickeln, bei der eine *paarweise* Kooperation nicht ausreicht. Voraussetzung hierfür ist allerdings, dass sich häufig *mehrere* Roboter eines Teams gleichzeitig oder mit geringer zeitlicher Differenz sichten können. Die Idee ist, dass eine zuverlässige akustische Schall-Lokalisation die Grundlage hierfür schafft. Zwei Roboter können sich auch dann akustisch relativ lokalisieren, wenn keine direkte Sichtverbindung besteht oder die Roboter sich aus anderen Gründen nicht visuell wahrnehmen können.

In [19] findet sich ein weiteres Beispiel kooperativer Lokalisierung. Hier erforscht ein Team von Robotern eine unbekannte Umgebung. Dabei gibt es stationäre Roboter, die ausschließlich die relativen Positionen zu den anderen Robotern messen. Die mobilen Roboter erforschen z.B. mit Laserscannern ihre direkte Umgebung

erforschen. Die Messungen der mobilen Roboter fügen sich mit Hilfe der relativen Messungen zu einem globalen Umweltmodell zusammen.

Kapitel 3

Aktive Schall-Lokalisation

3.1 Begriffsklärungen

- Lokalisation, Ortung

Als *Lokalisation* wird die Bestimmung von Distanz und Richtung zu einer Landmarke oder einer beweglichen Einheit bezeichnet. In der Literatur *impliziert* der Begriff Lokalisation häufig ein *akustisches Verfahren*. Um Verwechslungen vorzubeugen wird in dieser Arbeit der Terminus ‘Schall-Lokalisation’ verwendet. Schall-Lokalisation bezeichnet die Lokalisation basierend auf akustischen Richtungs- und Distanzmessungen.

Hinweis: Verfahren zur *Lokalisierung* werden häufig auch als Lokalisation bezeichnet. Durch den uneindeutigen Terminus ‘localization’ im Englischen entsteht eine weitere Verwechslungsgefahr.

Der Begriff *Ortung* wird in der Fachliteratur vor allem in Verbindung mit auf elektromagnetischen Wellen basierenden Verfahren gebraucht, wie zum Beispiel Radar oder GPS. Die vorliegende Arbeit wird dieser Bezeichnungsweise folgen.

- *Aktive* Schall-Lokalisation

Der Begriff *Aktive Schall-Lokalisation* bezeichnet in dieser Arbeit ein aus Sendern und Empfängern bestehendes System, in dem die Sender Schallmuster über einen Lautsprecher aussenden. Die Empfänger können die Sender dann anhand der empfangenen Schallmuster akustisch lokalisieren.

- Störgeräusch

Als *Störgeräusch* wird im Rahmen dieser Arbeit ein zufälliges, a priori unbekanntes, akustisches Signal bezeichnet, das nicht planmäßiger Teil der Lokalisation ist.

- Akustisches Rauschen

Akustisches Rauschen ist die Summe aller in der Umgebung vorhandenen Störgeräusche. Akustisches Rauschen ist im Allgemeinen ein kontinuierliches, zufälliges und breitbandiges Signal. Im Folgenden wird Akustisches Rauschen auch kurz als *Rauschen* bezeichnet.

- Störsignal

Mit *Störsignal* wird im Rahmen diese Arbeit ein nicht-zufälliges, im Allgemeinen kontinuierliches Signal bezeichnet, dass den Vorgang der Lokalisation behindert. Es gibt zwei Arten von Störsignalen:

- Störsignale, die zum Zwecke *der Sabotage* der Lokalisation erzeugt werden;
- Störsignale, die durch die Aktive Lokalisation *selbst* erzeugt werden. Solche Störsignale treten z.B. auf, wenn mehrere Sender gleichzeitig ihre Signale aussenden. Bei der Lokalisation eines bestimmten Senders sind dann die Signale der anderen Sender als Störsignale anzusehen.

- Nutzsinal

Als *Nutzsignal* wird ein Signal bezeichnet, das planmäßig im Rahmen der aktiven Lokalisation hervorgerufen wurde und der Lokalisation dienlich ist.

- Signal-Rausch-Verhältnis

Als *Signal-Rausch-Verhältnis* (kurz: *SNR*, von ‘signal to noise ratio’) wird das Verhältnis der mittleren Lautstärken von Nutzsinal und Rauschen bezeichnet.

3.2 Schall-Lokalisation beim Menschen

Die Ansätze zur Lokalisation, die in dieser Arbeit behandelt werden, sind teilweise inspiriert durch den Hörsinn beim Menschen. Der menschliche Hörsinn verwendet vor allem zwei Methoden zur Lokalisation eines akustischen Signals.

Bis zu einer Grenzfrequenz f_{G_1} können Sinustöne anhand von Phasendifferenzen lokalisiert werden. Das leicht zeitversetzte Eintreffen eines Sinustons der Frequenz

f an den Ohren geht mit einem proportionalen Fortschreiten der Phase des Sinustons einher. Anhand dieser Phasendifferenz $d\phi$ kann dann die Wegdifferenz ds bestimmt werden (siehe Abbildung 3.1):

$$ds = \frac{c}{f} \cdot \frac{d\phi}{2\pi}.$$

c ist dabei die temperaturabhängige Schallgeschwindigkeit in der Luft, die hier mit $c = 343 \frac{m}{s}$ angenommen wird (bei 20 Grad Celsius). $\frac{c}{f} =: \lambda$ entspricht einer Wellenlänge des Sinustons.

Die Methode liefert die Wegdifferenz ds nur dann *eindeutig*, wenn der Abstand b zwischen den Ohren, die sogenannte *Basis*, kleiner ist als eine halbe Wellenlänge des Sinustons $\frac{\lambda}{2}$. Ist $b > \frac{\lambda}{2}$, so ist die Wegdifferenz ds nur zwei- oder mehrdeutig bestimmbar.

Ist ds *eindeutig* bekannt und $b < \frac{\lambda}{2}$, so kann die Richtung α dennoch nur *zweideutig* bestimmt werden, da die Richtung $\alpha' = \pi - \alpha$ die *gleiche* Wegdifferenz ds zur Folge hat. Der Mensch kann die Schallrichtung dennoch eindeutig bestimmen, indem er den Kopf leicht dreht. Die Kombination aus zwei Messungen von ds mit unterschiedlicher Kopfstellung ermöglicht die *eindeutige* Bestimmung der Richtung.

Der Ohrenabstand beim Menschen beträgt $b_{Mensch} \cong 18cm$, die Grenzfrequenz ergibt sich damit zu

$$f_{G_1} = 0.5 \frac{c}{b_{Mensch}} \cong 950Hz.$$

Experimente zeigen, dass die effektive Grenzfrequenz beim Menschen mit rund $850Hz$ noch etwas niedriger liegt als der theoretische Wert.

Die zweite wichtige Methode für die Lokalisation beim Menschen ist die Verwendung von Amplitudendifferenzen. Diese Methode funktioniert erst ab einer Grenzfrequenz von $f_{G_2} \cong 1500Hz$. Sinuswellen tieferer Frequenzen bewegen sich mit zu geringer Amplitudenabschwächung durch Hindernisse, wie z.B. auch den menschlichen Kopf, hindurch.

Die Bestimmung der Richtung zu einer Signalquelle durch Amplitudendifferenzen ist beim Menschen äußerst komplex und durch die spezifische Form der Ohrmuschel bedingt, die bei jedem Menschen unterschiedlich ausfällt. Eine Berechnungsformel kann nicht direkt angegeben werden. Vielmehr gründet sich diese Fähigkeit

beim Menschen auf Erfahrung. In [1] wird das menschliche Richtungshören auf neuronaler Ebene genauer erforscht.

Schallquellen reiner Sinuswellen im Bereich $f_{G_1} < f < f_{G_2}$ können durch den menschlichen Hörsinn tatsächlich kaum lokalisiert werden.

Eine akustische *Distanzmessung* findet im menschlichen Hörsinn nicht statt. Grund dafür ist, dass eine genaue Distanz akustisch nur durch die Laufzeit eines Signals bestimmt werden kann. Um also eine Distanz zu bestimmen, müsste der *genaue Zeitpunkt* der Signalerzeugung bekannt sein. Dazu muss aber normalerweise die Schallquelle, z.B. eine zufallende Tür, auch *gesehen* werden. Dann kann jedoch die Distanz auch direkt *rein visuell* bestimmt werden. Eine akustische Distanzmessung bringt hier also keinerlei neue Information.

Ein bekanntes Alltagsbeispiel ist hier die Distanzbestimmung zu einem Gewitter. Genähert wird angenommen, dass die Sichtung des Blitzes mit dem Aussenden des Donners zeitlich koinzidiert. Durch das Zählen der Sekunden bis zum Hören des Donners kann dann die Distanz geschätzt werden. (Man beachte, dass diese Berechnung im Gehirn auf einer hohen kognitiven Ebene geschieht und nicht dem Hörsinn selbst zugeschrieben werden kann.)

Tatsächlich aber kann der Mensch im Alltag Entfernungen zu Schallquellen durchaus einschätzen. Wird z.B. in 100 Metern Entfernung der Motor eines Autos gestartet, so hat der Mensch *rein akustisch* eine recht gute Vorstellung von dieser Distanz. Der Grund liegt darin, dass der Klang und auch die Lautstärke eines startenden Motors dem Menschen in etwa bekannt sind. Im Laufe des Lebens erlernt der Mensch, Veränderungen des Klangmusters mit der Schallausbreitung im Raum in Verbindung zu bringen. Für die Schätzung der Entfernung ist es dann unerheblich, ob man den Schall in Natur oder z.B. über Kopfhörer hört.

3.3 Grundlagen der Schall-Lokalisation

In diesem Abschnitt werden die Grundlagen der Schall-Lokalisation mit zwei Mikrofonen erläutert. Diese Grundlagen sind allen in dieser Arbeit vorgestellten Ansätzen zur aktiven Schall-Lokalisation gemeinsam.

3.3.1 Richtungsmessung

Um die Richtung, aus der ein erkanntes Signal kommt, zu berechnen, wird die Wegdifferenz ds zwischen der Schallquelle und den beiden Empfangsmikrofonen

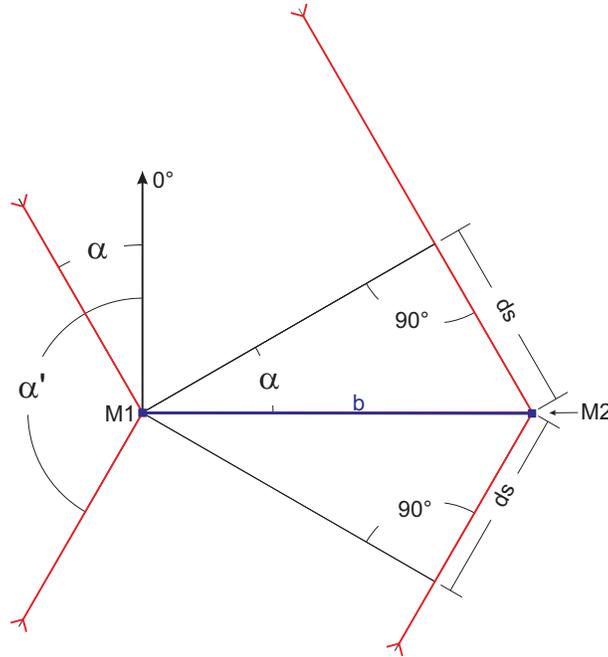


Abbildung 3.1: *Richtungsmessung anhand der Wegdifferenz ds (Sicht von oben). M_1 und M_2 sind die Positionen der beiden Mikrophone, b bezeichnet die Basis. ds sei die gemessene Wegdifferenz. α und α' sind dann die beiden möglichen Richtungswinkel. Diese werden relativ zur eingezeichneten 0° -Linie angegeben.*

bestimmt. Zusätzlich muss die Länge der Basis b , d.h. der Abstand zwischen den Empfangsmikrophonen bekannt sein. Der Richtungswinkel α wird dann anhand folgender Formel berechnet (siehe auch Abbildung 3.1):

$$\alpha = \arcsin\left(\frac{ds}{b}\right)$$

Die Berechnung des Winkels ist nicht eindeutig. Für jedes α ist auch $\alpha' = \pi - \alpha$ eine mögliche Lösung. Dies geht auch geometrisch aus Abbildung 3.1 hervor.

Man beachte, dass hier vereinfachend angenommen wurde, dass die beiden eintreffenden Schallwellenfronten *parallel* zueinander verlaufen, was normalerweise nicht der Fall ist.

Um α *exakt* zu berechnen, muss der absolute Abstand zwischen Sender und Empfänger bekannt sein, was im Allgemeinen nicht der Fall sein wird.

Abbildung 3.2 verdeutlicht die Situation bei nicht als parallel angenommenen Strecken. Sei R der geometrische Bezugspunkt des Empfängers. R befindet sich

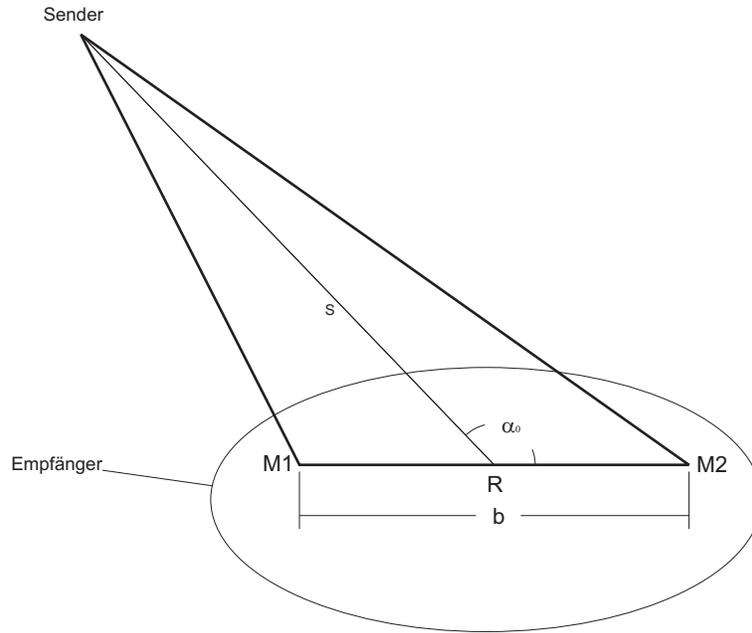


Abbildung 3.2: *Reale Situation der Richtungsmessung: Nicht-parallel verlaufende Schallwellenfronten.*

genau in der Mitte zwischen den Empfangsmikrophonen M_1 und M_2 . Sei s der Abstand zwischen R und der punktförmig angenommenen Schallquelle S . Der resultierende Fehler ist zum einen abhängig vom tatsächlichen Richtungswinkel α_0 und zum anderen vom Verhältnis $\frac{s}{b}$. Der Fehler ist dabei umso geringer, je größer $\frac{s}{b}$ ist. Abbildung 3.3 zeigt den *maximalen* Fehler in Abhängigkeit vom Verhältnis $\frac{s}{b}$. (Ein ausführlicheres Diagramm ist im Anhang A zu finden.)

Ab einem Abstand von $2 \cdot b$ beträgt der maximale Fehler bereits nur noch weniger als 1° . Dieser Fehler ist deutlich geringer als die Fehler, die durch digitale Abtastung (siehe 3.7.1), Interferenz und Brechung (siehe 3.4.4) auftreten. Der Fehler durch die Annahme, dass die Strecken parallel sind, ist für $s > 2 \cdot b$ also hinnehmbar. In der Praxis gilt normalerweise $s \gg 2 \cdot b$. Der Fehler ist *unvermeidbar*, wenn der Abstand s nicht bekannt ist.

Die Wegdifferenz ds wird in den diskutierten Ansätzen anhand der *zeitlichen* Differenz zwischen dem Eintreffen eines Signals an den beiden Mikrofonen bestimmt. Das Erkennen eines bestimmten Signals und die Bestimmung des genauen *Empfangszeitpunktes* an einem Mikrofon wird daher die eigentliche Hauptaufgabe für die Richtungsmessung sein.

Der Fehler bei der Berechnung der Richtung ist direkt vom Fehler bei der Bestimmung von ds abhängig. Abbildung 3.4 zeigt, wie stark sich dieser Fehler bei

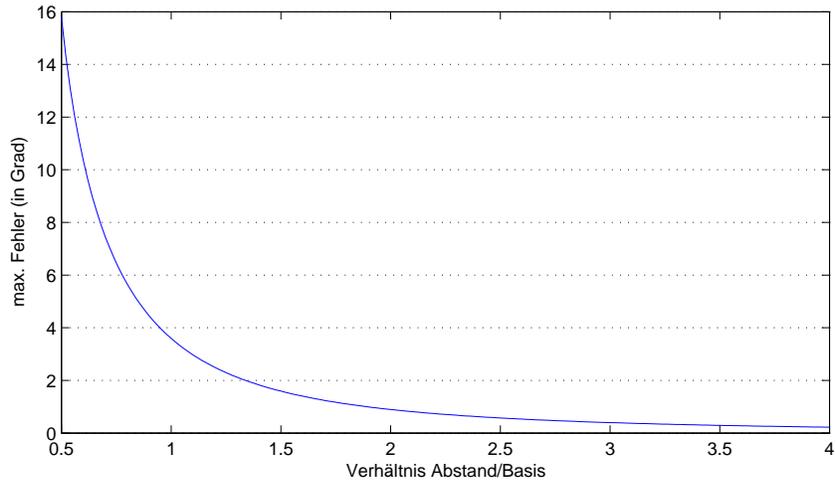


Abbildung 3.3: *Maximaler Fehler durch die Annahme, dass die Strecken zwischen Schallquelle und den Mikrofonen parallel verlaufen.*

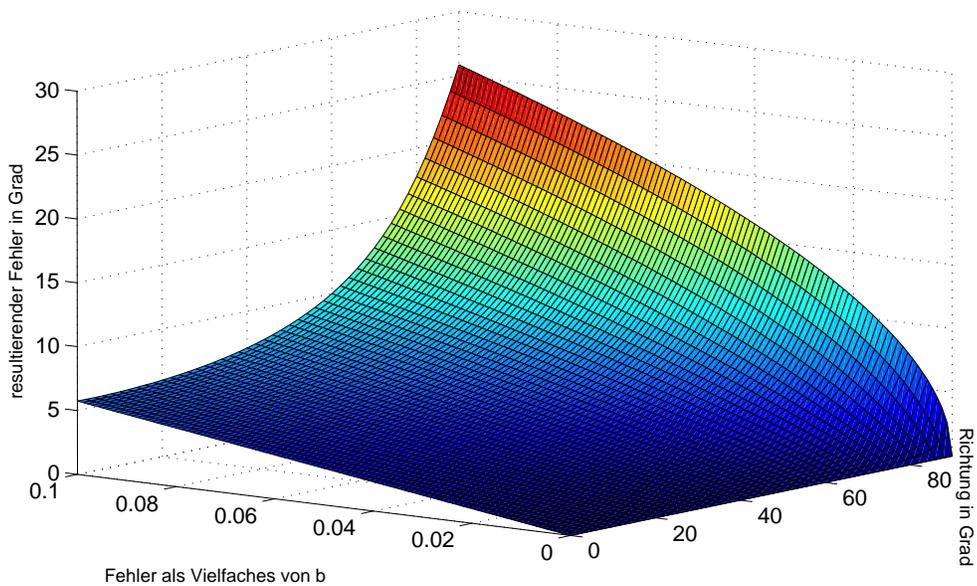


Abbildung 3.4: *Fehlerfortpflanzung bei der Richtungsmessung: Der resultierende Fehler ist abhängig von der Richtung des Signals und dem Messfehler von d_s . Die Fehlerfortpflanzung ist symmetrisch für positive bzw. negative Eingangswerte. Hier wird der Fall positiver Winkel und Messfehler dargestellt.*

der Berechnung der Richtung fortpflanzt. Besonders bei Richtungswinkeln in der Nähe von $\pm 90^\circ$ macht sich ein Messfehler von ds deutlich bemerkbar. Normalerweise ist gerade in diesen Bereichen der Einfluss von Interferenzen auf die Bestimmung von ds besonders hoch (siehe Kapitel 3.4.4). Dies muss in der Anwendung unbedingt berücksichtigt werden.

3.3.2 Distanzmessung

Die Berechnung der Distanz erfolgt durch die Messung der Zeit, die zwischen dem Aussenden und dem Empfang einer akustischen Welle verstreicht. Dazu muss dem Empfänger der *genaue* Zeitpunkt des Aussendens bekannt sein. Außerdem muss er den genauen Empfangszeitpunkt ermitteln. Die Distanz s zwischen Sender und Empfängermikrofon ergibt sich dann mit der Schallgeschwindigkeit c zu

$$s = c \cdot dt = c \cdot (t_{Empfang} - t_{Senden}).$$

Ein Problem bei der Distanzbestimmung unter autonomen Robotern ist, dass die Roboter normalerweise keine synchronisierten Uhren haben. Selbst, wenn ein unabhängiger Kommunikationskanal zur Verfügung steht, können sich die Roboter daher nicht über den Absendezeitpunkt verständigen, da sie unterschiedliche Uhren als Referenz verwenden. Im Rahmen dieser Arbeit werden drei Verfahren zur Synchronisation der Uhren diskutiert, siehe Kapitel 3.7.2.

3.4 Erster Ansatz: Fourier-Transformation

Der erste im Rahmen dieser Arbeit diskutierte Ansatz verwendet die Fourier-Transformation (kurz: FT). Die Fourier-Transformation basiert auf der Feststellung, dass jedes endliche Signal durch eine unendliche Summe von Sinus- oder Kosinuswellen beschrieben werden kann.

3.4.1 Diskrete Fourier-Transformation

Im Falle zeitdiskreter Signale, die im Rahmen dieser Arbeit ausschließlich vorkommen, verwendet man die Diskrete Fourier-Transformation (kurz: DFT), die im Folgenden näher betrachtet wird.

Sei $x_k(t)$ ein endliches, kontinuierliches Signal. Sei $x(t)$ das zeitdiskrete, endliche Signal, das durch Abtastung von $x_k(t)$ entsteht. Weiter seien f_A die verwendete Abtastrate, und L die Anzahl an Abtastpunkten von $x(t)$.

Die Frequenz $f_1 = \frac{f_A}{L}$ wird als *Grundfrequenz* bezeichnet. Sinuswellen mit dieser Frequenz werden als Grundschnwingungen bezeichnet. Die Wellenlänge einer Grundschnwingung entspricht also gerade der Signallänge, L . Vielfache der Grundfrequenz werden als *Harmonische* bezeichnet, die zugehörigen Sinuswellen als *Oberschnwingungen*. Das zeitdiskrete, endliche Signal $x(t)$ kann als Summe aus einem konstanten Summanden a_0 , einer Grundschnwingung und $\frac{L}{2} - 1$ Oberschnwingungen dargestellt werden:

$$x(t) = a_0 + \sum_{k=1}^{\frac{L}{2}} [a_k \cdot \sin(k \cdot 2\pi f_1 \cdot t + \varphi_k)].$$

Das Signal $x(t)$ kann also äquivalent auch durch die Amplituden a_k und die Phasen φ_k ($k = 0 \dots \frac{L}{2}$) beschrieben werden. Man spricht dann von einer Darstellung im Frequenzraum. Die DFT leistet genau diese Umrechnung.

Die Berechnung der DFT ist sehr aufwendig und liegt in der Komplexitätsklasse $O(n^2)$, wobei n die Länge des zu transformierenden Intervalls ist. Eine effizientere Variante des DFT-Algorithmus ist die Schnelle Fourier-Transformation (kurz: *FFT*, Fast Fourier Transformation). Die FFT ist ein rekursiver, sogenannter ‘divide and conquer’-Algorithmus. Im Zuge der Berechnung können bei der FFT viele Werte geschickt wiederverwendet werden. Die Komplexitätsklasse verringert sich gegenüber der DFT auf $O(n \cdot \log n)$. Als Voraussetzung muss die Länge des Intervalls n eine Zweierpotenz sein.

Der FFT-Algorithmus im Pseudocode befindet sich im Anhang E.

3.4.2 Eigenschaften der Spektralzerlegung

Die Transformation in den Frequenzraum kann als eine Art Filterung angesehen werden. Sei f_1 die Grundfrequenz. Ist das Signal eine reine Oberschnwingung, d.h. eine Sinuswelle der Frequenz $f_{x,1} = q \cdot f_1$ mit $q \in \mathbb{N}$ und $q \geq 1$, so gilt $a_k = 0$ für alle $k \neq q$, und $a_q > 0$. Abbildung 3.5 zeigt ein einfaches Beispiel.

Handelt es sich bei dem Signal *nicht* um eine reine Oberschnwingung, also $f_{x,2} = q \cdot f_1$, mit $q \notin \mathbb{N}$ und $q > 1$, so sind im Allgemeinen *alle* $a_k > 0$, das heißt das Spektrum umfasst sämtliche Oberschnwingungen. Die beiden von $f_{x,2}$ nächstgelegenen

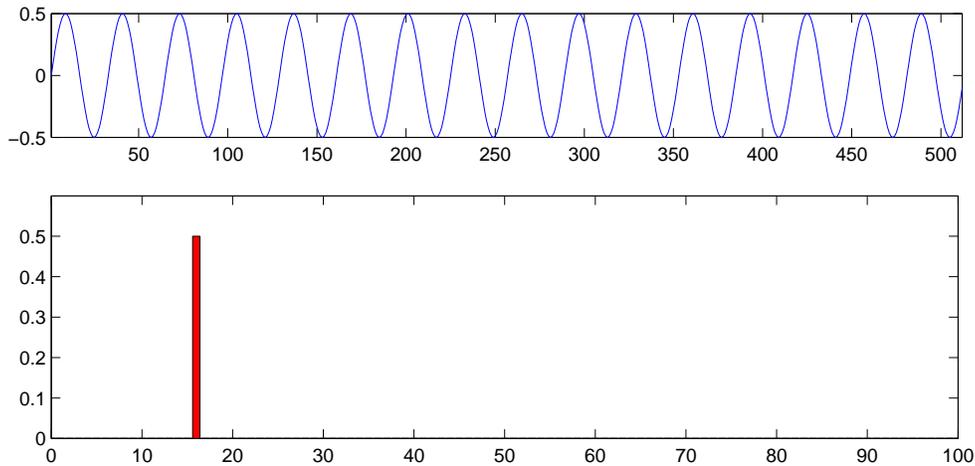


Abbildung 3.5: Oben: Eine durchgehende Oberschwingung der Frequenz $f_{x,1} = 16 \cdot f_1$. Das Signal hat genau die Länge eines Transformationsintervalls (512 Abtastpunkte). Unten: zugehöriges Amplitudenspektrum bis a_{100} .

Oberschwingungen $f_{oben} = \lceil q \rceil \cdot f_1$ und $f_{unten} = \lfloor q \rfloor \cdot f_1$ sind im Spektrum allerdings mit deutlich höheren Amplituden vertreten als die übrigen Oberschwingungen. Abbildung 3.6 gibt hierzu ein Beispiel. Um optimale Ergebnisse zu erreichen ist es also sinnvoll, als Nutzsignale stets reine Oberschwingungen zu verwenden.

Die DFT basiert auf periodischen Schwingungen. Die spektrale Aufspaltung des Signals erfolgt daher in reine Oberschwingungen, die sich *über das gesamte Transformationsintervall* erstrecken. *Nichtperiodische* Signalanteile bewirken eine Verwischung im Frequenzspektrum. Ein Sinuspuls, der sich nicht über das gesamte Intervall erstreckt, ist ein Beispiel für ein nichtperiodisches Signal. Abbildung 3.7 zeigt ein solches Signal mit dem dazugehörigen Amplitudenspektrum. Praktisch bietet sich daher vor allem die Verwendung durchgehender Sinustöne an.

3.4.3 Lokalisation

Die Lokalisation in einer Gruppe von autonomen Robotern kann mit Hilfe der FFT so erfolgen: Jedem Roboter wird eine eindeutige Oberschwingung zugeordnet. Den Robotern der Gruppe sind diese Zuordnungen bekannt. Die Auswahl der Oberschwingungen sollte dabei unter Berücksichtigung der Übertragungseigenschaften der verwendeten Hardware erfolgen. Außerdem sind die Frequenzen möglichst so zu wählen, dass in den entsprechenden Frequenzbereichen

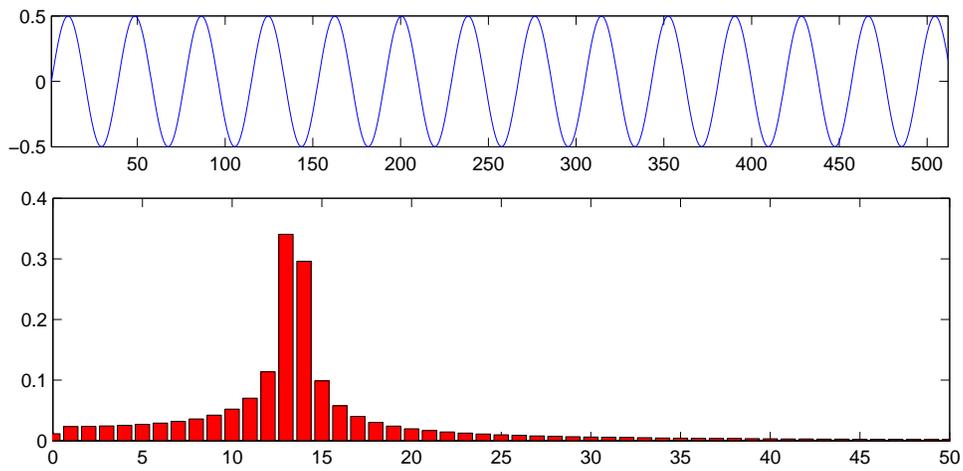


Abbildung 3.6: *Oben: Ein durchgehende Schwingung der Frequenz $f_{x,2} = 13.47 \cdot f_1$. Das Signal hat genau die Länge eines Transformationsintervalls (512 Abtastpunkte). Unten: zugehöriges Amplitudenspektrum bis a_{50} .*

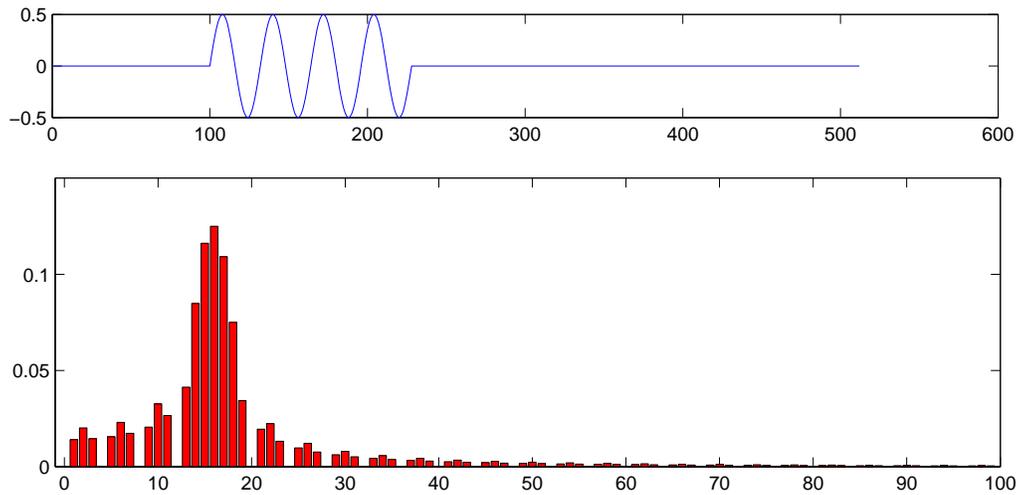


Abbildung 3.7: *Oben: Ein kurzer Sinuspuls der harmonischen Frequenz $f_{x,1} = 16 \cdot f_1$. Das dargestellte Signal hat genau die Länge eines Transformationsintervalls (512 Abtastpunkte). Unten: zugehöriges Amplitudenspektrum bis a_{100} .*

möglichst wenige Störgeräusche in der Umgebung vorkommen. Dazu kann man die Geräusche der Umgebung aufnehmen und eine Spektralanalyse vornehmen.

Ein Roboter A sendet *durchgehend* den Sinuston seiner spezifischen Frequenz. Ein zweiter Roboter B kann dann die Richtung, aus der das Signal kommt, berechnen. Dazu führt B eine FFT der beiden Eingangssignale seiner Mikrophone durch und berechnet so für beide Kanäle *die Phase* von As spezifischer Oberschwingung, φ_1 und φ_2 . Die Wellenlänge λ_f einer akustischen Sinuswelle der Frequenz f kann einfach mit Hilfe der Schallgeschwindigkeit c berechnet werden als

$$\lambda_f = \frac{c}{f}.$$

Die Wegdifferenz ds , wie in Abbildung 3.1 definiert, kann dann aus der Phasendifferenz und der Wellenlänge berechnet werden zu

$$ds = \frac{\varphi_2 - \varphi_1}{2\pi} \cdot \frac{c}{f} = \frac{d\varphi}{2\pi} \cdot \lambda_f.$$

Die Berechnung der Richtung erfolgt dann wie in Kapitel 3.3.1 beschrieben.

Die Berechnung der *Distanz* zu einem Roboter ist im Allgemeinen nicht eindeutig. Die Phase am geometrischen Bezugspunkt sei mit φ_B bezeichnet. Liegt der Bezugspunkt genau zwischen den beiden Mikrophenen, so gilt $\varphi_B = \frac{1}{2} \cdot (\varphi_1 + \varphi_2)$.

Aus φ_B kann die Distanz nur dann eindeutig berechnet werden, falls die Distanz zwischen zwei Robotern *nie größer ist* als eine einzige Wellenlänge λ_f . Außerdem muss für die Berechnung der Distanz eine Initialisierung mit Distanz d_0 und Phase $\varphi_{B,0}$ bekannt sein. Die Distanz kann dann in Abhängigkeit von φ_B berechnet werden:

$$d = \left(\frac{\varphi_B - \varphi_{B,0}}{2\pi} \cdot \lambda_f - d_0 \right) \bmod \lambda_f.$$

Der modulo-Operator bewirkt, dass die berechnete Distanz stets zwischen 0 und λ_f liegt.

Im Allgemeinen werden die maximalen Distanzen allerdings deutlich größer sein als λ_f . Zum Beispiel beträgt die maximale Distanz in einem RoboCup Fußballspiel bis zu 12 Meter. In sinnvoll verwendbaren Frequenzbereichen (siehe Kapitel 3.6.7) ist λ_f kleiner als ein Meter. In diesem Fall ist die genaue Distanz also unklar. Sie kann nur modulo λ_f bestimmt werden, was normalerweise eine wertlose Information darstellt.

Dieses Problem kann umgangen werden, falls Distanzsprünge um mehr als $\frac{\lambda_f}{2}$ zwischen zwei Messzeitpunkten ausgeschlossen werden können. Ausgehend von einem Initialwert kann die Distanz durch einfaches Addieren der nach oben bzw. nach unten durchlaufenen Phasen-Nulldurchgänge über einen längeren Zeitraum verfolgt werden ('Distanz-Tracking').

Dieser recht einfache Ansatz birgt allerdings große Risiken. Grobe Messfehler durch Störgeräusche oder Bewegungen um mehr als $\frac{\lambda_f}{2}$ zwischen zwei Messzeitpunkten können zu groben, *systematischen* Fehlern *aller* darauffolgenden Distanzmessungen führen.

Der Ansatz, die Fourier-Transformation zur Distanzmessung zu verwenden, wurde daher nicht weiter verfolgt.

3.4.4 Reflexion, Brechung und Interferenz

Wie in Kapitel 3.3.1 gezeigt wurde, erfordert die Richtungsmessung eine hohe Genauigkeit für die Bestimmung von ds . In diesem Abschnitt soll der Einfluss von Reflexion, Brechung und Interferenz von Schallwellen auf die Genauigkeit der Messungen verdeutlicht werden. Die tatsächlichen Resultate sind in hohem Maße umgebungsbedingt und können hier nur qualitativ beschrieben werden.

Eine kurze Schallwelle werde von Roboter A ausgesendet. Angenommen, die Schallwelle erreicht die Mikrophone des empfangenden Roboters B zunächst auf dem direkten Wege. Zusätzlich gelangt die Schallwelle aber auch an Hindernisse der Umgebung, wie z.B. Wände oder andere Roboter. Die Schallwelle wird an diesen Hindernissen reflektiert und kann über diesen Umweg zusätzlich an die Mikrophone von B gelangen. Da diese reflektierte Welle einen weiteren Weg zurückgelegt hat als die direkte Verbindung von A nach B, ist ihre Phase schon weiter fortgeschritten als die direkt eingetroffene Welle. Die reflektierte Schallwelle wird durch den weiteren Ausbreitungsweg und die Reflexion gegenüber der direkt eintreffenden Welle stärker gedämpft. (Der genaue Zusammenhang zwischen zurückgelegter Distanz und Amplitudenrückgang ist vor allem von der Frequenz und der akustischen Charakteristik des Raumes abhängig.)

Entscheidend ist nun die Interferenz der direkten mit der reflektierten Schallwelle am Mikrophon des Empfängers B. Die Interferenz zweier Schallwellen entspricht einer Addition der sich überlagernden Schallwellen. Handelt es sich bei den Schallwellen, wie im vorliegenden Fall, um reine Sinusschwingungen mit gleicher Frequenz und unterschiedlicher Phase, so ist die resultierende Welle ebenfalls eine Sinuswelle derselben Frequenz.

Entscheidend für die Richtungsmessung ist hier die Verschiebung der Phase der resultierenden Welle. Die Phase der reflektierten Welle hängt dabei von der Länge des Umwegs ab, und ist daher normalerweise unbekannt.

Ein weiteres Problem entsteht durch die Brechung der Schallwellen, besonders in der Nähe der Mikrophone, z.B. am Chassis des Roboters. Dieser Effekt macht sich dann besonders stark bemerkbar, wenn sich Teile des Chassis zwischen Schallquelle und Empfangsmikrofon befinden. Die Schallwelle wird dann am Chassis gebeugt/gebrochen und überlagert sich am Mikrofon mit der direkt eintreffenden Schallwelle. Die gebrochene Welle ist gegenüber der direkten zwar durch den meist sehr kleinen Wegunterschied nur leicht phasenverschoben, kann aber eine Amplitude besitzen, die *größer* ist als die der direkten Welle.

Wie bei der Reflexion ist der Einfluss der Interferenz mit an Hindernissen gebrochenen Wellen in hohem Maße umgebungsabhängig.

Qualitativ wird der Einfluss von Reflexion und Brechung durch folgende Umstände begünstigt:

- Die Empfangsmikrophone weisen eine schmale Richtcharakteristik auf.
- Ein Mikrofon ist in entgegengesetzter Richtung zur Schallquelle ausgerichtet.
- Hohe Frequenzen sind anfälliger für Brechung und Reflexion als tiefe.
- Zwischen den Mikrophenen befinden sich Hindernisse, d.h. es besteht nicht von beiden Mikrophenen aus direkte Sichtverbindung zur Schallquelle.
- Die Basis, d.h. der Abstand zwischen den Mikrophenen, ist zu klein.
- Die direkte Umgebung von Sender und Empfänger weist Hindernisse oder Wände auf. (Normalerweise nicht vermeidbar.)

3.4.4.1 Messung des Einflusses von Reflexion und Brechung

Zur Verdeutlichung der Situation wurde in einem Experiment die Phasenverschiebung zwischen den beiden Empfangsmikrophenen eines Roboters aufgezeichnet. Der kurze Puls bestand dabei aus 16 Sinus-Halbwellen. Zur Berechnung der Phasen wurde für jeden Abtastzeitpunkt nach Eintreffen der Wellenfront eine *sehr kurze* Fourier-Transformation über nur 32 Abtastpunkte durchgeführt.

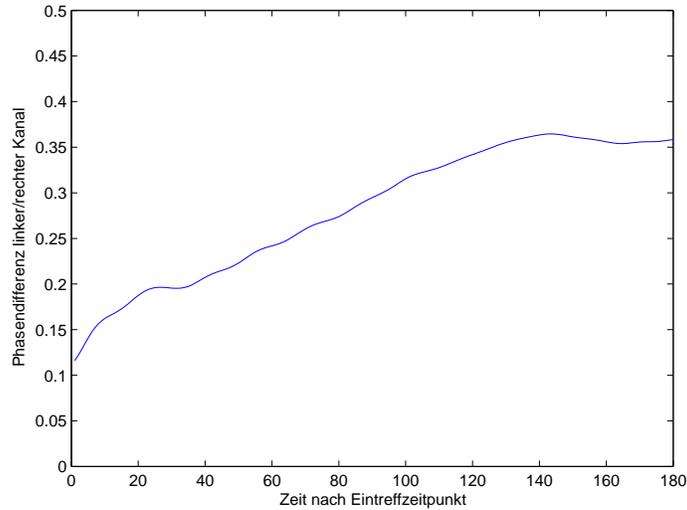


Abbildung 3.8: *Experiment zur Messung des Einflusses von Reflexion und Brechung. Die Phasendifferenz (aufgetragen in rad) zwischen rechtem und linkem Kanal vergrößert sich nach Eintreffen der vordersten Wellenfront durch Interferenz mit gebrochenen und reflektierten Wellen.*

Abbildung 3.8 zeigt das Ergebnis der Messungen. Der initiale Wert von etwa 0.12rad entspricht dabei mit guter Genauigkeit der tatsächlichen Schallrichtung. Mit fortschreitender Zeit wird der Einfluss von Brechung und Reflexion sichtbar. Die Phase des rechten Kanals verlagert sich durch Interferenz stärker nach hinten, so dass sich die Phasendifferenz, und damit auch der resultierende Winkel zur Schallquelle, scheinbar vergrößert. Die *qualitative* Struktur dieser Verzögerung der Phase ist sehr gut reproduzierbar: Die Phasendifferenz vergrößert sich betragsmäßig bei gleichbleibendem Vorzeichen. Der genaue Verlauf variiert stark in Abhängigkeit von Hindernissen und Wänden der Umgebung sowie dem tatsächlichen Richtungswinkel.

Hinweis: In einem Eingangssignal mit erheblichen Störgeräuschen ist es nicht praktikabel, über die ersten 32 Abtastpunkte des eintreffenden Pulses eine kurze FT durchzuführen und damit die Richtung zu ermitteln. Die wichtige *filternde* Eigenschaft der FT funktioniert nur über deutlich größere Transformationsintervalle (siehe oben). Die Vorgehensweise war in diesem Experiment nur aufgrund der völlig störfreien Umgebung möglich.

3.4.5 Einfluss von Störgeräuschen

Zusätzlich zu den Problemen durch Reflexion und Interferenz kommt noch der Einfluss von Störgeräuschen. Ein Störgeräusch entspricht normalerweise einer

Überlagerung von vielen Sinuswellen unterschiedlicher Frequenz.

Zu den Störgeräuschen gehören neben den Störgeräuschen der Umgebung auch solche Störgeräusche, die durch den Roboter selbst hervorgerufen werden.

Besonders in Bewegung treten im Roboter oft Störgeräusche durch Motoren auf. Sind die Mikrophone starr am Chassis aufgehängt, so gelangen diese intern verursachten Störgeräusche über die starre, mechanische Kopplung an die Membrane der Mikrophone. Die Lautstärke dieser Störgeräusche ist dann *deutlich höher* als es auf Grund der *äußeren* Geräuschentwicklung anzunehmen wäre.

Vor allem bei *laufenden* Robotern sind normalerweise viele Elektromotoren gleichzeitig in Betrieb. Dazu kommen noch die Störgeräusche, die durch das Auftreffen der am Laufen beteiligten Glieder auf den Untergrund entstehen.

Im Allgemeinen kann die Amplitude der Störgeräusche die Amplitude des eintreffenden Nutzsignals um ein Vielfaches übertreffen. Die Trennung von Nutzsignal und Störgeräuschen gleicher Frequenz ist nicht möglich, falls man als Nutzsignal reine Sinuswellen verwendet. Der in Kapitel 3.6 behandelte Ansatz wird diesem Problem Rechnung tragen. Die dort verwendeten Nutzsignale sind *keine* reinen Sinuswellen.

Die Frequenzen der Nutzsignale sollten sich also möglichst wenig mit den Frequenzen der Störgeräusche überlagern. Zur Ermittlung der typischen Störgeräusch-Frequenzen werden die Geräusche der Umgebung aufgenommen und danach analysiert. Der Roboter sollte dabei typische Tätigkeiten ausführen, damit auch der Einfluss von Gelenkmotoren und von vom Roboter verursachten Umgebungsgeräuschen erfasst wird.

3.4.6 Zusammenfassung

Die Spektralzerlegung mit Hilfe der FFT ist ein einfach umzusetzender Ansatz. Der Rechenaufwand für die fortlaufende Transformation zweier Signale ist verhältnismäßig hoch.

Der Ansatz ist zur *Distanzmessung* nur dann geeignet, wenn eine hohe Zuverlässigkeit der Messungen gegeben ist.

Die *Richtungsmessung* ist zudem aus den beschriebenen Gründen im Allgemeinen sehr ungenau und zudem sehr unzuverlässig. Um die Spektralzerlegung zur Richtungsmessung zuverlässig zu verwenden, bedarf es einiger Voraussetzungen in Bezug auf die verwendete Hardware und die Umgebung. Insbesondere muss der Einfluss von Reflexion und Interferenz gering gehalten werden, indem stets eine direkte Sichtverbindung zwischen Mikrophenen und Sender besteht.

3.5 Zweiter Ansatz: Verwendung eines digitalen Bandpassfilters

3.5.1 Kurze Einführung

Dieser Ansatz verwendet einen digitalen Bandpassfilter. Die Idee ist, dass der Sender sehr kurze Sinuspulse einer bekannten Frequenz aussendet. Der Empfänger führt eine möglichst schmale Bandpassfilterung um die Übertragungsfrequenz durch. Schmal bedeutet hier, dass nur Frequenzen in der nahen Umgebung der Filterfrequenz passieren dürfen. Damit soll der Einfluss von Störgeräuschen vermindert werden. Dadurch, dass die Pulse nur sehr kurz sind, soll zudem der Einfluss von Reflexion und Brechung vermindert werden.

Der Ansatz wurde insgesamt aufgrund absehbarer Probleme nicht allzu ausführlich verfolgt. Die einzelnen Sektionen fallen daher recht knapp aus. In [20], Kapitel 5, und [21] finden sich ausführlichere Beschreibungen zum Entwurf digitaler Filter, die für die vorliegende Ausarbeitung verwendet wurden.

3.5.2 Lokalisation

Es wird zunächst kurz erläutert, wie die Lokalisation mit Hilfe eines Frequenzfilters ablaufen kann.

Die Roboter senden regelmäßig sehr kurze Sinustöne (z.B. 8 nur Sinus-Halbwellen) der Frequenz f_0 aus. Der empfangende Roboter muss zur Bestimmung der Distanz den Aussendezeitpunkt kennen. Der digitale Bandpassfilter des empfangenden Roboters wird auf die Frequenz f_0 eingestellt. Die Idee ist, dass dadurch Störgeräusche herausgefiltert werden können. Der genaue Empfangszeitpunkt des Sinuspulses kann dann ermittelt werden, indem z.B. der mittlere Nulldurchgang des kurzen Sinustons bestimmt wird.

3.5.3 Entwurf des Bandpassfilters

Die digitale Filterung entspricht einer Faltung im Zeit-/Amplitudenraum. Da praktisch nicht über unendlich große Intervalle gefaltet werden kann, muss eine

endliche Fenstergröße n gewählt werden. Der Filter selbst entspricht dann einem Intervall der Größe n . Die diskrete Faltung wird nach folgender Formel berechnet:

$$a(t) = \sum_{k=1}^n f(k) \cdot x(t - k)$$

Um den Filter an den Rändern zu glätten, werden sogenannte Fensterfunktionen eingesetzt. Dazu wird der Filter mit der Fensterfunktion multipliziert. Gängige Fensterfunktionen sind z.B. das Hamming-Fenster oder das von-Hann-Fenster (siehe z.B. in [20], Kapitel 5.3.3). Abbildung 3.9 a) zeigt das verwendete Hamming-Fenster der Länge $n = 512$ Abtastpunkte. Abbildung 3.9 b) zeigt den resultierenden, geglätteten Bandpassfilter.

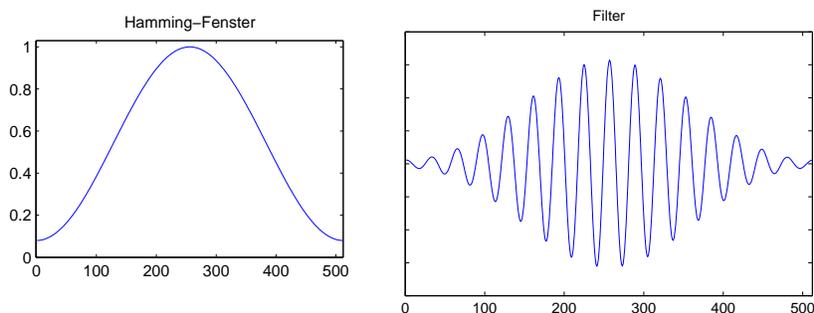


Abbildung 3.9: links: a) verwendetes Hamming-Fenster, rechts: b) resultierender Bandpassfilter. Die Fenstergröße beträgt $n = 512$.

Die zu filternde Frequenz sei f_0 . Die Grenzfrequenzen sind $\Omega_l = f_0 - df$ und $\Omega_2 = f_0 + df$. Die in Abbildung 3.9 b) verwendete Frequenz f_0 ist abhängig von der verwendeten Abtastfrequenz f_A und beträgt hier $f_0 = \frac{f_A}{32}$. (Wellen der Frequenz f_0 besitzen hier also eine Wellenlänge von 32 Abtastpunkten.)

Ein digitaler Filter kann nie völlig scharf um einen bestimmten Frequenzbereich filtern. Wählt man $df = 0$ so ist das Ergebnis der Filterung die Nullfunktion. An den Rändern des Frequenzbereichs gibt es stets Toleranzen. Die unterschiedlichen Verfahren zum Filterentwurf weisen unterschiedliche Eigenschaften insbesondere in diesen Randbereichen auf. Verwendet wurde ein optimaler, nicht-rekursiver Bandpassfilter.

3.5.4 Filtereigenschaften

Abbildung 3.10 zeigt Beispiele für die Filterung. Das gefilterte Signal entspricht dem um $\frac{n}{2}$ Abtastpunkte verzögerten, gefilterten Eingangssignal (n ist dabei die

Fenstergröße). In der Abbildung wurde das gefilterte Signal daher um $\frac{n}{2}$ nach links verschoben, so dass man unmittelbar Eingangssignal und gefiltertes Signal vergleichen kann.

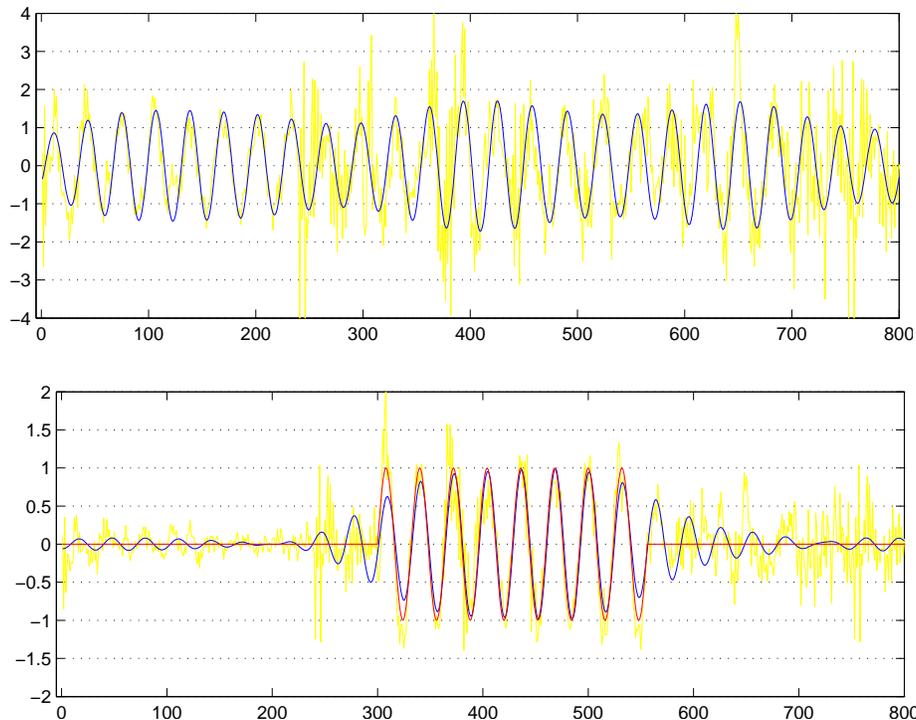


Abbildung 3.10: Beispiele für die Bandpassfilterung. Zu sehen ist jeweils das Eingangssignal (gelb) sowie das gefilterte Signal (blau). Das Nutzsignal ist jeweils mit leichtem Rauschen überlagert.

oben: a) Das Nutzsignal ist ein durchgehender Sinuston gleichbleibender Amplitude mit der Frequenz f_0 . Das Rauschen verursacht hier zufällige Amplitudenschwankungen.

unten: b) Das Nutzsignal (rot eingezeichnet) ist ein kurzer Sinuston der Frequenz f_0 .

Abbildung 3.10 a): Das Signal ist eine durchgehende Welle der Frequenz f_0 , überlagert mit einem leisen Rauschen. Die Filterung funktioniert sehr gut, das Rauschen wurde erfolgreich herausgefiltert.

Abbildung 3.10 b): Das Signal ist ein kurzer Sinuspuls, wie er für die Lokalisation verwendet werden soll, überlagert mit leisem Rauschen. Am gefilterten Signal erkennt man gut die Verrechnung der Eingangswerte über die zurückliegenden n Werte. Statt eines kurzen, gefilterten Pulses entsteht ein anschwellender und abklingender Puls. Um die Position des Pulses zu bestimmen, kann z.B. die Pulsmitte als Referenzposition dienen. Zur Ermittlung der Pulsmitte kann das gefil-

terte Signal an verschiedenen Nulldurchgängen gespiegelt werden. Es wird dann die Trennstelle mit maximaler Ähnlichkeit zwischen den entstehenden Hälften bestimmt. Auf Details wird hier nicht weiter eingegangen.

Der nächste Abschnitt wird zeigen, dass der digitale Bandpassfilter bei *sehr lauten* Störgeräuschen nicht mehr sinnvoll mit kurzen Sinustönen zu verwenden ist.

3.5.5 Probleme des Ansatzes

Das Hauptproblem des Ansatzes besteht darin, dass die robotereigenen Störgeräusche stets auch Anteile mit der Grundfrequenz besitzen. Diese Störanteile können dann nicht vom Nutzsignal unterschieden werden.

Außerdem ist die beschriebene Bandpass-Filterung recht unscharf, d.h. es werden auch Frequenzen, die klar von f_0 abweichen (z.B. $f_1 = f_0 \pm 100Hz$), nicht völlig herausgefiltert. Ein sehr lautes Störsignal der Frequenz f_1 führt in der gefilterten Wellenform dann zu zusätzlichen Sinuswellen der Frequenz f_0 .

Das eigentliche Nutzsignal der Frequenz f_0 kann in der resultierenden Überlagerung jeweils nicht mehr zuverlässig identifiziert werden.

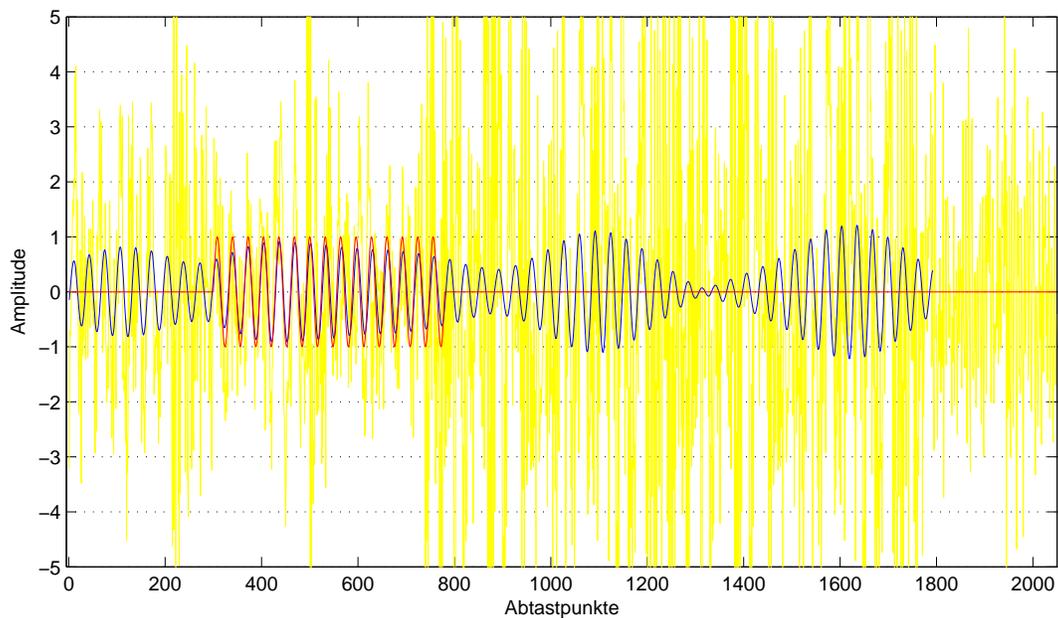


Abbildung 3.11: *Beispiel für die Filterung eines stark verrauschten Eingangssignals. Rot: Nutzsignal, gelb: Laufgeräusche des empfangenden Roboters, blau: gefilterte Wellenform.*

Abbildung 3.11 zeigt eine Beispielsituation. Das Nutzsignal ist ein kurzer Sinuston der Frequenz f_0 (rot). Überlagert damit sind aufgenommene Laufgeräusche eines Roboters (gelb), die deutlich lauter sind, als das Nutzsignal. Es wurde mit einer Fenstergröße von 512 Abtastpunkten gefiltert. Das Ergebnis der Filterung (blau) zeigt, dass das Nutzsignal gar nicht mehr identifiziert werden kann. Die Störgeräusche erzeugen in der gefilterten Wellenform zusätzliche Wellen der Frequenz f_0 .

3.5.6 Zusammenfassung

Der Ansatz ist aufgrund der genannten Gründe unzureichend für die Lokalisation in Teams autonomer Roboter in verrauschten Umgebungen. Die hier durch Elektromotoren und Umgebung auftretenden Störgeräusche sind zu laut für eine ausreichend gute Filterung. Ein recht leises Nutzsignal kann bereits bei mäßigen Störgeräuschen nicht mehr in der Wellenform identifiziert werden. Der Ansatz wurde daher nicht weiter verfolgt.

Der Ansatz der Kreuzkorrelationsfunktionen (KKF) des nächsten Abschnitts ist einer Faltung durchaus ähnlich. Hierbei werden allerdings keine reinen Sinuswellen als Nutzsignal verwendet, was das Herausfiltern von sinuswellenförmigen Störgeräuschen der Grundfrequenz ermöglicht.

3.6 Dritter Ansatz: Kreuzkorrelation

3.6.1 Kurze Einführung

Der dritte Ansatz zur Lokalisation diskutiert die Verwendung von *Kreuzkorrelationsfunktionen* (kurz: *KKF*).

KKFs werden allgemein dazu verwendet, um die Ähnlichkeit zwischen zwei Signalen zu bestimmen. Dabei wird z.B. die Ähnlichkeit zwischen einem Eingangssignal und einem vorab bekannten Schema berechnet. Ein Empfänger kann dadurch den Empfangszeitpunkt des von einem Sender ausgesandten Schemas bestimmen. Mehrere Schemata können von verschiedenen Sendern gleichzeitig ausgesendet werden, wenn die einzelnen Schemata gewisse Eigenschaften zueinander aufweisen. Die Schemata sind hierbei nicht sinuswellenförmig, wodurch Störgeräusche besser herausgefiltert werden können.

Diese Technik wird bereits seit etwa 1970 für das *Global Positioning System*, GPS, verwendet (siehe [9]). Sender sind hierbei die im Orbit befindlichen GPS-Satelliten. Empfänger sind beispielsweise Navigationsgeräte. Weitere bekannte Anwendungsbeispiele sind vor allem moderne, drahtlose Netzwerke zur Datenübertragung, wie Bluetooth [10] oder UMTS [11].

Diese Anwendungen basieren alle auf elektromagnetischen Wellen. Im Rahmen dieser Arbeit soll die grundlegende Technik auf die Verwendung *akustischer* Wellen übertragen werden. Der Entwurf wird außerdem in Hinblick auf die Lokalisation in kleinen Gruppen autonomer, mobiler Roboter erfolgen. Praktische Aspekte, wie Uhrendrift und Bewegungsunschärfe müssen beim Entwurf berücksichtigt werden.

Bedingt durch die unterschiedlichen Ausbreitungseigenschaften von akustischen und elektromagnetischen Wellen, insbesondere der Ausbreitungsgeschwindigkeit, ergeben sich neue Anforderungen für einen erfolgreichen Einsatz.

3.6.2 Begriffsklärungen

Im Folgenden werden einige Definitionen und Begriffsklärungen gegeben, wie sie *im Rahmen dieser Arbeit* verwendet werden. Die Termini sind in der deutschsprachigen Fachliteratur teils uneinheitlich. Im Rahmen dieser Arbeit wurde versucht, die verwendeten Begriffe möglichst intuitiv zu halten.

- Eine **Codefolge** (kurz: **Code**) ist eine binäre Folge bestehend aus -1 und 1. Die Verwendung von -1 und 1 anstatt 0 und 1 ist beim Umgang mit KKF's üblich. Der Grund wird später deutlich werden.
- Eine **Codemenge** ist eine Menge von mehreren Codefolgen, die normalerweise zueinander gewisse Eigenschaften aufweisen.
- Ein **Chip** ist ein einzelnes Element einer Codefolge. Ein Chip hat stets den Wert -1 oder 1. Der Terminus Chip wird in der Fachliteratur durchgehend gebraucht. Im Text werden Chips eines bestimmten Wertes als (-1)Chip bzw. (1)Chip bezeichnet. (*Hinweis*: Der Begriff des *Bit* wird im Kontext von KKF's normalerweise nur im Zusammenhang mit der Übertragung von *Nutzdaten* gebraucht. Der Unterschied zwischen Chips und Bits wird in Kapitel 3.6.4 deutlich werden.)
- **Modulation** ist die Umwandlung von Codefolgen in Wellenformen. Verschiedene Ansätze der Modulation werden in Kapitel 3.6.5 behandelt.

- Ein **modulierter Chip**, kurz **mChip**, ist die *Wellenform*, die bei Modulation eines einzelnen Chips entsteht. Es gibt (-1)mChips und (1)mChips. Ein mChip ist zunächst nicht als akustisch oder elektromagnetisch typisiert.
- Als **Codeschema** (kurz: *Schema*) wird die Wellenform bezeichnet, die bei *Modulation einer Codefolge* entsteht. Ein Codeschema ist also stets eine Folge von (-1)mChips und (1)mChips.

3.6.3 Kreuz- und Autokorrelation: Definitionen

Gegeben sei ein Codeschema $s(\tau)$ der Länge η . Gegeben sei außerdem ein Eingangssignal beliebiger Länge $x(t)$. Die Kreuzkorrelationsfunktion $K_s(x, t)$ liefert in Abhängigkeit des Zeitpunktes t einen Wert für die *Ähnlichkeit* zwischen dem Eingangssignal im Intervall $t \dots t + \eta$ und dem Codeschema $s(\tau)$. Die Kreuzkorrelationsfunktion ist für kontinuierliche Signale definiert als

$$\hat{K}_s(x, t) = \int_0^\eta x(t + \tau) \cdot s(\tau) d\tau.$$

Im Folgenden werden ausschließlich *zeitdiskrete* Signale und Codeschemata verwendet. Die Kreuzkorrelationsfunktion für zeitdiskrete Signale lautet dann

$$K_s(x, t) = \sum_{\tau=0}^{\eta-1} x(t + \tau) \cdot s(\tau).$$

Häufig wird dem Integral bzw. der Summe noch der Dämpfungsfaktor $\frac{1}{\eta}$ vorangestellt, so dass die Werte unabhängig von der Schemalänge η sind. Da im Folgenden aber stets *nur die Verhältnisse* zwischen Werten von K_s , nicht aber die absoluten Größen von Bedeutung sind, wird dieser Dämpfungsfaktor nicht verwendet.

Kreuzkorrelationen *zwischen zwei gleichlangen Codeschemata* spielen eine besondere Rolle bei der Bewertung von Codemengen. Seien s_1 und s_2 zwei diskrete Codeschemata der Länge η . Die Veränderliche δt entspricht dann der *Verschiebung* zwischen den beiden Schemata. Mit Hilfe der Modulo-Funktion wird die Kreuzkorrelationsfunktion zweier Schemata wie folgt definiert:

$$\bar{K}_{s_1, s_2}(\delta t) = \sum_{\tau=0}^{\eta-1} s_1(\tau) \cdot s_2((\delta t + \tau) \bmod \eta).$$

Der Modulo-Operator sorgt dabei für die *zyklische* Verschiebung von s_2 um δt . Die Funktion \bar{K}_{s_1, s_2} ist daher zyklisch und für alle $\delta t \in \mathbb{IN}$ definiert.

Im Folgenden wird verkürzend $s^{\delta t}$ für das um δt zyklisch nach rechts verschobene Schema s geschrieben.

Der *betragsmäßig größte Wert* der Kreuzkorrelation zweier Schemata wird definiert als

$$\bar{K}_{s_1, s_2}^{max} := \max |\bar{K}_{s_1, s_2}(\delta t)|,$$

mit $\delta t \in [0 \dots \eta - 1]$.

Die *Autokorrelation* entspricht der Kreuzkorrelation eines diskreten Schemas s mit sich selbst. Die Autokorrelationsfunktion ist daher definiert als

$$A_s(\delta t) := \bar{K}_{s, s}(\delta t).$$

Für $1 \leq \delta t < \eta$ handelt es sich um eine *verschobene* Autokorrelation. $A_s(\delta t)$ ist offensichtlich stets maximal für $\delta t = 0$. Im Folgenden wird folgende Schreibweise verwendet:

$$A_s^0 := A_s(0).$$

Der maximale Wert der *verschobenen* Autokorrelation wird definiert als

$$A_s^{max} := \max |A_{s_B}(\delta t)|,$$

mit $\delta t \in [1 \dots \eta - 1]$.

3.6.4 Kreuzkorrelation bei GPS

Der hier beschriebene Ansatz zu Lokalisation ist vor allem motiviert durch das *Global Positioning System*, GPS. Die Funktionsweise von GPS soll hier vereinfacht und mit wenigen Worten beschrieben werden.

Jedem Satellit ist ein eindeutiger Code zugewiesen. Die Modulation der Codefolgen erfolgt durch Phasensprünge (siehe Kapitel 3.6.5). Alle Satellitenuhren werden mit Hilfe von Bodenstationen sehr exakt regelmäßig synchronisiert. Die Satelliten senden durchgehend, und zwar genau gleichzeitig ihr Schema aus. Das Senden erfolgt dabei auf nur einer Frequenz.

Dem Empfänger sind die Schemata der verschiedenen Satelliten bekannt. Zusätzlich kennt der Empfänger die Umlaufbahnen der einzelnen Satelliten, d.h. er kann zu einem bestimmten Zeitpunkt die Position des Satelliten ausrechnen. Der Empfänger führt zur Positionierung eine Kreuzkorrelation des Eingangssignals

mit den Schemata aller (erreichbaren) Satelliten durch, und kann dadurch für jeden Satelliten einen *Empfangszeitpunkt* bestimmen. Aus der berechneten Laufzeit der einzelnen Signale können die Abstände zu den Satelliten ermittelt werden. Zusammen mit den bekannten Positionen der Satelliten zum Absendezeitpunkt kann dann die eigene Position durch Triangulation berechnet werden.

Für die Positionierung im dreidimensionalen Raum genügt der Schnitt von drei Kugeloberflächen. Gemessene Abstände zu drei Satelliten genügen also theoretisch zur Positionierung (vorausgesetzt, die Satelliten liegen nicht auf einer Geraden).

In der Praxis muss der *Uhrenfehler* der Empfängeruhr mit berechnet werden. Dies ist nötig, da kleine Ungenauigkeiten der Empfängeruhr bereits deutliche Auswirkungen auf die berechneten Wegstrecken haben. Die Lokalisierung erfolgt daher im *vierdimensionalen* Raum. Somit sind Abstandsmessungen zu mindestens vier Satelliten erforderlich. Für eine *genaue* Positionsbestimmung ist zusätzlich eine möglichst große Überbestimmung mit anschließender *vermittelnder Ausgleichung* erforderlich.

3.6.5 Modulation von Codefolgen

Für die Modulation der Codefolgen werden im Folgenden verschiedene Varianten beschrieben.

Dabei müssen die Ausbreitungseigenschaften akustischer Wellen berücksichtigt werden. Zudem sind Lautsprecher weniger dynamisch als elektromagnetische Sender, was z.B. die Erzeugung von *Phasensprüngen* in Wellenformen betrifft.

Wichtig für die Qualität einer Modulation vor dem Hintergrund der Kreuzkorrelation ist außerdem, dass zwei *unterschiedliche* mChips einen betragsmäßig großen, *negativen* Korrelationswert zueinander besitzen. Angenommen der Korrelationswert eines mChips mit sich selbst beträgt +1. Dann beträgt der kleinstmögliche Wert der Kreuzkorrelation von unterschiedlichen mChips -1. Dies ist der theoretisch beste erreichbare Wert.

Welcher der im Folgenden beschriebenen Ansätze zur Modulation in der Praxis am besten abschneidet, hängt stark von der Wiedergabecharakteristik des verwendeten Lautsprechers ab. Kapitel 5.7 zeigt Messungen mit den verschiedenen Ansätzen.

Zu jedem der folgenden Ansätze wird in Abbildung 3.12 als Beispiel die Modulation der Codefolge [+1 -1 +1 +1 -1 -1 -1 +1] gezeigt.

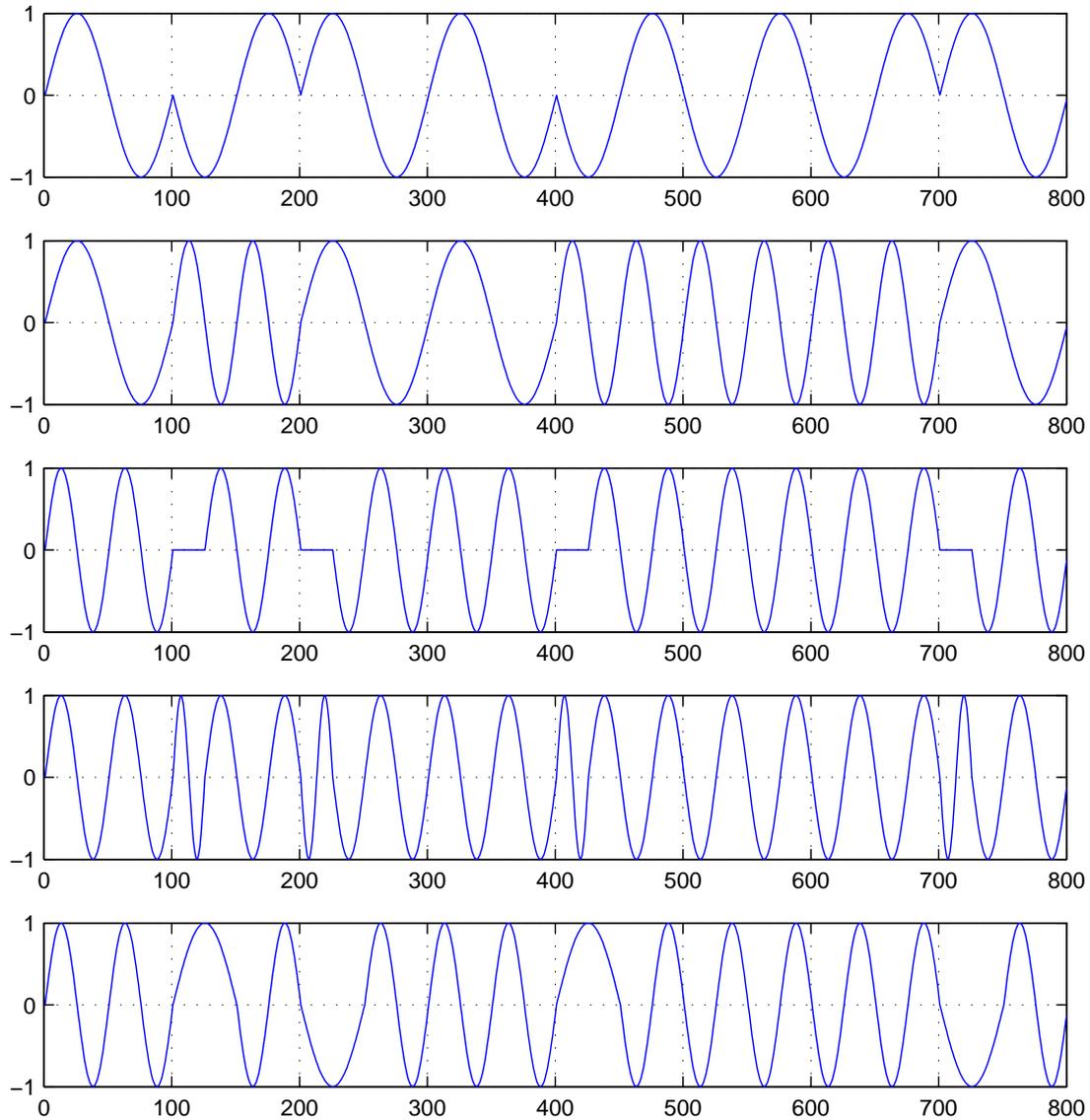


Abbildung 3.12: *Verschiedene Ansätze zur Modulation von Codefolgen. Es wird exemplarisch jeweils die Codefolge $[+1 -1 +1 +1 -1 -1 -1 +1]$ moduliert. Von oben nach unten: a) durch Phasensprünge, b) durch Frequenzsprünge; Glättung von Phasensprüngen c) durch Pausen, d) durch hochfrequente Pulse, e) durch niederfrequente Pulse.*

3.6.5.1 Modulation durch Phasensprünge

Dieser Ansatz ist der einfachste und, zumindest theoretisch, optimal. Ein (+1)mChip entspricht einem vollständigen Sinus-Durchlauf. Ein (-1)mChip besteht aus einem um 180° *phasenverschobenen* Sinus-Durchlauf. Der Ansatz ist optimal, da

$$-\int_0^{2\pi} \sin(\alpha)\sin(\alpha)d\alpha = \int_0^{2\pi} \sin(\alpha)\sin(\pi + \alpha)d\alpha = -2.$$

Ein Beispiel ist in Abbildung 3.12 a) zu finden.

Das Problem des Ansatzes ist, dass ein Phasensprung in einer akustischen Welle normalerweise schlecht übertragen wird. Im Bereich des Phasensprungs flachen die benachbarten Wellenberge stark ab. Des Weiteren können Phasensprünge von einer Lautsprechermembran nur schlecht *wiedergegeben* werden. Die Membran erreicht beim Durchschwingen durch die Nulllage maximale Geschwindigkeit. Die Bewegungsrichtung der Membran müsste aber genau dann in die *entgegengesetzte* Richtung umschlagen. Um dies in einem kurzen Zeitintervall zu schaffen, muss eine sehr starke Gegenbeschleunigung auf die Membran wirken. Die durch die Spule eines gewöhnlichen Lautsprechers induzierte Kraft reicht dafür normalerweise nicht aus, bzw. erfordert eine große Verstärkerleistung.

3.6.5.2 Modulation durch Frequenzsprünge

Dieser Ansatz verwendet für die unterschiedlichen Chips unterschiedliche Frequenzen. Der Ansatz basiert auf der Beobachtung, dass sich Sinus-Wellen unterschiedlicher Frequenzen bei Überlagerung auslöschen:

$$\int_0^\infty \sin(\varphi_0 + 2\pi f_0 t) \cdot \sin(\varphi_1 + 2\pi f_1 t) dt = 0,$$

falls $f_0 \neq f_1$. Dies gilt unabhängig von den Phasen φ_0 und φ_1 . Für endliche Integrationsintervalle ist dies im Allgemeinen *nicht* erfüllt.

Ist die Länge des Intervalls ein Vielfaches von 2π , sowie $f^- = n \cdot f^+$ mit $n \in \mathbb{N}$, so folgt ebenfalls eine vollständige Auslöschung:

$$\int_{t_0}^{t_0+k \cdot 2\pi} \sin(\varphi_0 + 2\pi f^+ t) \cdot \sin(\varphi_0 + 2\pi n f^+ t) \cdot dt = 0,$$

für $n, k \in \mathbb{N}$, $n \geq 2$ und $k \geq 1$.

Das $(+1)$ mChip sei ein einzelner Sinus-Durchgang der Frequenz f^+ . Das (-1) mChip entspricht dann n Sinus-Durchgängen gleicher Phase, mit der Frequenz $f_{-1} = n \cdot f_{+1}$. So ist auch sichergestellt, dass die mChips gleiche Länge haben. Abbildung 3.12 b) zeigt ein entsprechendes Beispiel für $n = 2$.

Der Vorteil des Ansatzes liegt in den guten Eigenschaften bei Wiedergabe und Ausbreitung der akustischen Wellen. Der Nachteil des Ansatzes liegt in den schlechten Korrelationseigenschaften. Die Kreuzkorrelation von (-1) mChip und $(+1)$ mChip ergibt 0. Der Ansatz erwies sich dadurch für die Praxis als untauglich.

3.6.5.3 Geglättete Phasensprünge

Aufgrund der Wichtigkeit einer guten Korrelationseigenschaft wird zur Verwendung von Phasensprüngen zurückgekehrt. Es gibt mehrere Möglichkeiten, 180° -Phasensprünge zu glätten, so dass die Welle bessere Ausbreitungs- und Wiedergabeeigenschaften aufweist. Die Güte der Modulation sollte dabei möglichst nah am Optimum bleiben.

Ein erster, einfacher Ansatz sieht eine Pause von einer halben Wellenlänge nach jedem Chip-Wechsel vor. Um den Einfluss der Pausen zu verringern, sollten pro mChip mehrere Sinus-Wellen verwendet werden. Abbildung 3.12 c) zeigt ein Beispiel mit vier anstatt zwei Halbwellen pro mChip. Da die Pausen *nur* direkt nach Chip-Wechseln verwendet werden, unterscheidet sich der Ansatz kaum noch von dem in Abschnitt 3.6.5.1. In einer pseudozufälligen Codefolge tritt im Durchschnitt alle zwei Chips eine *Chipänderung* auf. Im Mittel stimmen also noch $\frac{7}{8}$ aller Halbwellen mit denen einer Modulation durch Phasensprünge überein. Die Korrelationseigenschaft ist dementsprechend gut.

Eine andere Methode zur Glättung verwendet eine höherfrequente Sinuswelle hinter jedem Phasensprung. Abbildung 3.12 d) zeigt ein Beispiel. Es werden vier Sinus-Halbwellen pro mChip verwendet. Der hier gezeigte höherfrequente Puls verwendet die *doppelte* Grundfrequenz. Wiederum stimmen im Mittel $\frac{7}{8}$ aller Halbwellen mit denen einer Modulation durch Phasensprünge (Abschnitt 3.6.5.1) überein.

Ein letzter Ansatz verwendet im Bereich des Chip-Wechsels eine niederfrequenter Halbwelle *halber* Grundfrequenz. Abbildung 3.12 e) zeigt das entsprechende Beispiel. Es wurden vier Halbwellen pro mChip verwendet. Im Experiment konnte diese Modulation nicht überzeugen (siehe 5.7).

3.6.6 Auswahl und Erzeugung der Codefolgen

Die Auswahl geeigneter Codefolgen ist für die Qualität und Leistungsfähigkeit der Lokalisation von entscheidender Bedeutung. Die Güte der Auto- und Kreuzkorrelation und die Codelänge werden in den folgenden Abschnitten diskutiert. Es werden auch Algorithmen zur Erzeugung geeigneter Codemengen vorgestellt.

Die Art der Modulation muss sicherstellen, dass sich die Korrelationseigenschaften der *Codefolgen* möglichst stark auf die Korrelationseigenschaften der *Codeschemata* übertragen. Dies wurde durch die verschiedenen Ansätze in Abschnitt 3.6.5 versucht. Die Güte der Auto- bzw. Kreuzkorrelation von Codefolgen gilt dann auch für die modulierten Codeschemata.

3.6.6.1 Autokorrelationseigenschaften

Angenommen ein Eingangssignal enthält neben Störgeräuschen *genau ein* durchgehendes Codeschema s , das z.B. durch einen Roboter gesendet wurde. Wird das Eingangssignal $x(t)$ mit diesem Codeschema s selbst korreliert, so ist es wünschenswert, dass die zeitliche Position im Eingangssignal, d.h. der *Eintreffzeitpunkt*, *eindeutig* ermittelt werden kann. Man beachte, dass der Eintreffzeitpunkt durch den maximalen Wert der *Kreuzkorrelation* zwischen Eingangssignal und Codeschema identifiziert wird.

Anmerkung: Man spricht hierbei von *Kreuzkorrelation*, da das Eingangssignal normalerweise nicht *identisch* mit dem Codeschema ist. Die Verwendung des Begriffs ist möglicherweise verwirrend, da in gewisser Weise durchaus das Schema *mit sich selbst* korreliert wird. Das Eingangssignal unterscheidet sich aber vom reinen Schema, nämlich durch

- Störgeräuschanteile,
- Amplitude,
- weitere, überlagerte Schemata, falls mehrere Roboter gleichzeitig senden und
- eine andere Gestalt der Wellenform, bedingt durch die Charakteristik der verwendeten Hardware.

Die eindeutige Identifikation des Eintreffzeitpunktes erfordert für die Gestalt der *Autokorrelationsfunktion* $A_s(\delta t)$, dass es genau einen Funktionswert maximaler Korrelation geben muss, während alle übrigen Werte deutlich kleiner sind. Man beachte, dass der Funktionswert bei $\delta t = 0$ stets maximal ist. Es kann theoretisch allerdings noch weitere Verschiebungen $\delta t \neq 0$ geben, mit $A_s(\delta t) = A_s^0$. In diesem Fall wäre $A_s^0 = A_s^{max}$.

Der Grund, weshalb A_s^{max} *deutlich* kleiner sein sollte als A_s^0 liegt darin begründet, dass Störgeräusche im Eingangssignal $x(t)$ die Werte der Kreuzkorrelationsfunktion verfälschen. Die resultierenden Werte liegen dadurch zufällig höher oder niedriger. Ein Autokorrelationswert $A_s(\delta t)$, der nur knapp unter dem Maximum A_s^0 liegt, kann bei der Korrelation mit dem Eingangssignal leicht zum *maximalen* Wert führen: Sei t_0 der tatsächliche Eintreffzeitpunkt. Dann wäre $K_s(x, t_0 + \delta t) > K_s(x, t_0)$, obwohl $A_s(\delta t) < A_s^0$. Der errechnete Eintreffzeitpunkt des Signals ist dann falsch. Er liegt um δt später als der tatsächliche Eintreffzeitpunkt t_0 .

Abbildung 3.13 a) zeigt die Autokorrelation einer Codefolge mit guter Autokorrelationseigenschaft. Die Werte sind betragsmäßig klein. Das Verhältnis zwischen A_s^0 und dem betragsmäßig nächst größten Wert, A_s^{max} beträgt 7:1.

Abbildung 3.13 b) zeigt die Autokorrelation einer Codefolge mit schlechter Autokorrelationseigenschaft. Bei dieser kann man erkennen, dass A_s^{max} fast so groß ist wie A_s^0 . Der Ursache dafür ist, dass die Codefolge aus 3 nahezu identischen Abschnitten der Länge 21 besteht.

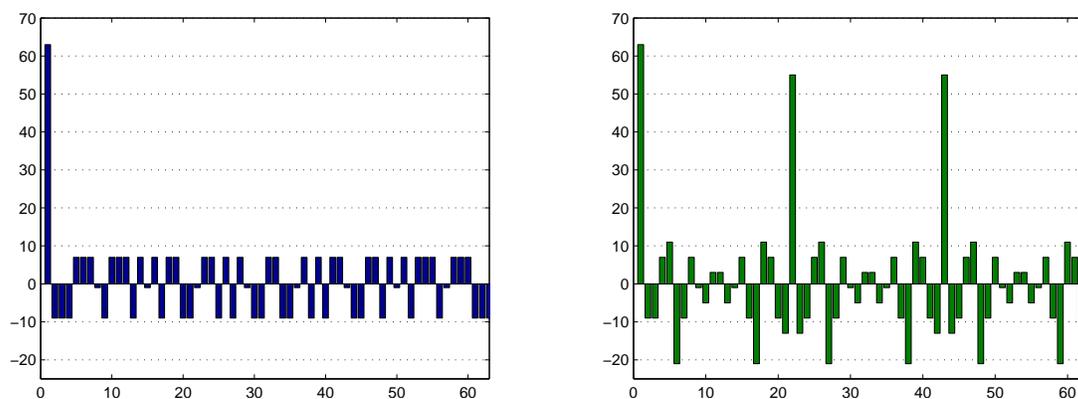


Abbildung 3.13: a) links: Autokorrelation einer Codefolge mit sehr guten Autokorrelationseigenschaften; b) rechts: Autokorrelation einer Codefolge mit schlechten Eigenschaften. Die 63-Chips langen Codefolgen sind im Anhang D zu finden.

3.6.6.2 Kreuzkorrelationseigenschaften

Bei der *gleichzeitigen* Lokalisation innerhalb einer Gruppe von Robotern ist die Qualität der Kreuzkorrelation der verwendeten Codes wichtiger als gute Autokorrelationseigenschaften. Ein großes Problem entsteht in der Praxis dadurch, dass ein Roboter normalerweise sein *eigenes* Codeschema *deutlich lauter* empfängt als die Schemata weiter entfernter Roboter, da sich der eigene Lautsprecher viel näher an den Mikrofonen des Roboters befindet.

Angenommen, ein Roboter A sendet selbst das Codeschema s_A . Ein anderer, entfernter Roboter B sendet das Codeschema s_B . Beide Schemata haben eine Länge von η . Das *Lautstärkeverhältnis* zwischen s_A und s_B betrage am Mikrophon von A q_v . Normalerweise ist $q_v > 1$, d.h. A hört sein eigenes Codeschema lauter als das von B. Angenommen, es gibt keine Störgeräusche.

Gemäß Kapitel 3.6.3 ist \bar{K}_{s_A, s_B}^{max} der maximale Wert der Kreuzkorrelation zwischen den beiden Codeschemata. $A_{s_B}^0$ ist der Wert der nicht verschobenen Autokorrelation von s_B , $A_{s_B}^{max}$ ist der maximale Wert der verschobenen Autokorrelation von s_B .

Es wird im Folgenden eine Bedingung hergeleitet, die für eine erfolgreiche und zuverlässige Ermittlung des Empfangszeitpunktes des Schemas von B *notwendig* ist. Da hierbei andere Einflussgrößen, z.B. Störgeräusche, vernachlässigt werden, ist diese Bedingung *nicht* hinreichend.

Die Kreuzkorrelationsfunktion ist linear für zusammengesetzte Signale $x(t)$:

$$x(t) = \alpha x_1(t) + \beta x_2(t) \Rightarrow K_s(x, t) = \alpha K_s(x_1, t) + \beta K_s(x_2, t).$$

Sei t_0 der *tatsächliche* Empfangszeitpunkt des Schemas s_B . Das Eingangssignal entspreche der Summe aus dem nicht verschobenen Codeschema s_B^0 und dem um q_v -fach lauterem, um δt verschobenen Codeschema $s_A^{\delta t}$. Die *absolute* Lautstärke wird hier nicht berücksichtigt, da ohnehin nur das *Verhältnis* der Lautstärken von Belang ist. Das Eingangssignal ist somit

$$x(t) = q_v \cdot s_A^{\delta t}(t - t_0) + s_B^0(t - t_0)$$

für $t_0 \leq t < t_0 + \eta$. Für $t \geq \eta$ und $t < t_0$ wiederhole sich das Eingangssignal periodisch.

Im *ungünstigsten* Fall ist die Verschiebung δt zwischen den beiden Schemata so, dass $\bar{K}_{s_B, s_A}(\delta t)$ *maximal negativ* ist. Die Kreuzkorrelation des Eingangssignals mit s_B zum Empfangszeitpunkt $t = t_0$ liefert dann im ungünstigsten Fall

$$K_{s_B}(x, t_0) = q_v \cdot \bar{K}_{s_B, s_A}(\delta t) + \bar{K}_{s_B, s_B}(0) = -q_v \cdot \bar{K}_{s_A, s_B}^{max} + A_{s_B}^0.$$

Der Empfangszeitpunkt wird durch das *Maximum* aller Kreuzkorrelationswerte identifiziert. Die übrigen Kreuzkorrelationswerte des Eingangssignals erreichen im *maximal positiven* Fall das q_v -fache des maximalen Kreuzkorrelationswerts der beiden Schemata. Hinzu kommt im ungünstigsten Fall noch der größte Wert der verschobenen Autokorrelation von s_B , $A_{s_B}^{max}$.

Um den Empfangszeitpunkt *zuverlässig* zu erkennen ist es daher erforderlich, dass

$$K_{s_B}(x, t_0) > A_{s_B}^{max} + q_v \cdot \bar{K}_{s_A, s_B}^{max}.$$

Durch Einsetzen und Umstellen erhält man die *notwendige* Bedingung

$$A_{s_B}^0 - A_{s_B}^{max} > 2 \cdot q_v \cdot \bar{K}_{s_A, s_B}^{max}.$$

Die Differenz $A_s^0 - A_{s_B}$ sollte in der Praxis *deutlich* größer sein als der rechte Teil der Ungleichung, um einen robusten Einsatz mit verrauschten Eingangssignalen zu gewährleisten.

Stellt man die Ungleichung um, so ergibt sich allgemein für zwei gegebene Codefolgen s_A und s_B ein *maximales* Lautstärkeverhältnis q_v^{max} :

$$q_v^{max} = \frac{A_{s_B}^0 - A_{s_B}^{max}}{2 \cdot \bar{K}_{s_A, s_B}^{max}}.$$

Die Messung bei der Verwendung bestimmter Codefolgen ist also nur bis zu einem maximalen Lautstärkeverhältnis q_v^{max} möglich. Bei Vergrößerung der Distanz zwischen A und B vergrößert sich auch das Lautstärkeverhältnis q_v . Als Folge hieraus ist die *maximale Distanz*, bis zu der zuverlässig Messungen durchgeführt werden können, durch q_v^{max} begrenzt.

Anmerkung: Man beachte, dass diese Begrenzung aufgrund des Lautstärkeverhältnisses, nicht aber wegen der absoluten Lautstärke auftritt. Eine gleichzeitige Erhöhung der Sendelautstärke sowohl bei A als auch B bringt daher keinen Gewinn.

Die Kreuzkorrelationen beliebiger Paare von Codes einer bestimmten Codemenge sollten daher betragsmäßig kleine Maxima aufweisen, falls Roboter *gleichzeitig* ihre Schemata senden. Dies ist dann die zentrale Eigenschaft bei der Auswahl der Codemengen.

Das *gleichzeitige* Messen von Distanzen ist zum einen vorteilhaft für eine ständige Synchronisierung der Roboteruhren (siehe Kapitel 3.7.2.2), sowie zum anderen für eine Gesamtauswertung *aller* Messungen in einer Gruppe von sich bewegenden Robotern (siehe Kapitel 4.3). Dennoch wurde aufgrund der hier gezeigten Begrenzungen auch die Möglichkeit untersucht, dass die Roboter abwechselnd ihre Schemata senden, siehe dazu Kapitel 5.11.

3.6.6.3 Codelänge, Codeschemalänge und Korrelationsintervalle

Das bestmögliche Verhältnis von $A_{s_A}^0$ bzw. $A_{s_B}^0$ und \bar{K}_{s_A, s_B}^{max} hängt von der *Codelänge* η ab. Welch untersucht in [22] das theoretische Optimum dieses Verhältnisses. Je länger die Codes sind, desto besser ist dieses theoretische Optimum. Zusätzlich kann es für bestimmte Algorithmen zur Codeerzeugung Einschränkungen für die Codelänge η geben (siehe 3.6.6.4). Man beachte, dass eine geeignete Modulation die Korrelationseigenschaften annähernd unverändert lässt, siehe Kapitel 3.6.5.

Neben der Codelänge η_s des Codes s ist auch die *zeitliche Länge* des zugehörigen modulierten *Codeschemas* χ_s von Bedeutung. χ_s und η_s stehen über die verwendete Grundfrequenz f_0 und die Anzahl an Sinuswellen-Durchläufen pro mChip κ in folgendem Zusammenhang:

$$\chi_s = \frac{\eta_s \cdot \kappa}{f_0}.$$

Außerdem wird die *räumliche Länge* eines sich als akustische Welle ausbreitendes Codeschemas definiert als

$$\psi_s = \chi_s \cdot c = \frac{\eta_s \cdot \kappa}{f_0} \cdot c = \eta_s \cdot \kappa \cdot \lambda_0.$$

Dabei bezeichnet c die Schallgeschwindigkeit und λ_0 eine Wellenlänge der Grundfrequenz.

Beim *fortlaufenden* Aussenden eines Codes ist die absolute Distanz zunächst *nicht eindeutig* bestimmbar. Die Distanz kann nur auf ψ_s genau bestimmt werden:

Sei die absolute Distanz $d_a = k \cdot \psi_s + d_R$, mit $0 \leq k \in \mathbb{N}$ und der Restdistanz $d_R < \psi_s$. Bei der Distanzmessung kann nur d_R bestimmt werden. Um die *absolute* Distanz zu bestimmen, muss k a priori bekannt sein. Hierfür kommen zwei sinnvolle Möglichkeiten in Frage:

- k kann mit Hilfe anderer Sensorik bestimmt werden.
- Die Schemalänge ψ_s wird lang genug gewählt, so dass für die *maximale Distanz* d^{max} in einer bestimmten Anwendung gilt: $d^{max} < \psi_s$. Dann gilt stets $k = 0$.

Ist eine dieser Bedingungen erfüllt, kann die Distanz auch dann absolut bestimmt werden, wenn das Schema fortlaufend gesendet wird.

Für die maximale Schemalänge gibt es auch begrenzende Faktoren:

Zum einen kann pro Schema nur eine Messung durchgeführt werden. Sind pro Zeiteinheit viele Messungen erforderlich, begrenzt dies die maximale zeitliche Schemalänge χ_s .

Zum anderen wirkt sich der Effekt der Bewegungsunschärfe stärker bei der Verwendung längerer Schemata aus. Die maximale zeitliche Schemalänge χ_s wird somit auch durch die maximale *relative Bewegungsgeschwindigkeit* zwischen zwei Robotern begrenzt. Die Bewegungsunschärfe wird ausführlich in Kapitel 3.6.8 behandelt.

Unabhängig von der Schemalänge kann die Länge des Korrelationsintervalls gewählt werden. Intervalle, die sich über mehrere Schemalängen erstrecken, können sinnvoll sein, um den Einfluss von Störgeräuschen zu verringern. Längere Korrelationsintervalle sorgen also für zuverlässigere Messungen bei ungünstigeren Signal-Rausch-Verhältnissen. Beispielsweise wird bei GPS ein Korrelationsintervall gewählt, das sich über 10 Schemalängen erstreckt. Grundsätzlich kommen nur ganzzahlige Vielfache der Schemalänge in Frage, damit die Kreuzkorrelationseigenschaften erhalten bleiben. Bei Verwendung langer Korrelationsintervalle nimmt ebenfalls der Einfluss der Bewegungsunschärfe zu.

3.6.6.4 Erzeugung der Codefolgen

Die Erzeugung von Codemengen mit guten Korrelationseigenschaften ist seit vielen Jahren ein wichtiges Thema in der Signalverarbeitung. Kasami und Gold haben bereits in den 1960er Jahren Algorithmen zur Erzeugung geeigneter Codemengen angegeben. Praktisch kann die Erzeugung mit Hilfe von einfachen Addierern und Schieberegistern durchgeführt werden.

Die Algorithmen von Kasami und Gold erzeugen jeweils Mengen von Codefolgen, die gute Auto-, sowie zueinander gute Kreuzkorrelationseigenschaften aufweisen. Eingabeparameter sind jeweils eine Größe n (siehe unten) und binäre Erzeuger-Polynome. Die Arbeiten von Gold und Kasami zur Codeerzeugung finden sich unter [13] und [14]. Eine einfache Beschreibung zur Erzeugung von Gold-Codemengen unter Verwendung einer logischen Schaltung mit Schieberegistern ist in [23] zu finden. Im Folgenden werden einige Eigenschaften von Kasami-Codemengen und Gold-Codemengen aufgelistet.

Kasami-Codemengen haben folgende Eigenschaften: Eine Codemenge beinhaltet genau $2^{\frac{n}{2}}$ Codefolgen für ein gegebenes, *gerades* $0 \leq n \in \mathbb{N}$. Ein einzelner

Code hat dabei eine Länge von $\eta = 2^n - 1$. Beispielsweise besteht eine Kasami-Codemenge für $n = 6$ aus 8 Codes der Länge $\eta = 63$. Die in Abbildung 3.13 a) gezeigte Autokorrelation gehört zu einem solchen Code. Die nächst größere Kasami-Codemenge besteht bereits aus 16 Codes der Länge $\eta = 255$.

Eine Gold-Codemenge besteht für ein beliebiges $n \in \mathbb{N}$ aus $2^n + 1$ Codefolgen der Länge $\eta = 2^n - 1$. Eine Gold-Codefolge ist ausgeglichen, d.h. die Anzahl von (1)Chips und (-1)Chips unterscheidet sich, aufgrund der ungeraden Codelänge, um genau 1. Beliebige Paare von Codefolgen einer Gold-Codemenge weisen allerdings unterschiedliche Korrelationsgüten auf. Es muss daher eine Teilmenge von Codes ermittelt werden, die bestmögliche Korrelationseigenschaften zueinander haben. GPS verwendet eine Gold-Codemenge (bzw. Teilmenge) mit $n = 10$ für die öffentlichen Codes. Jeder GPS-Satellit erhält einen eindeutigen Gold-Code.

Welch erforscht in [22] das theoretische Minimum für \bar{K}_{s_1, s_2}^{max} der Kreuzkorrelation zweier Codefolgen s_1 und s_2 . Seien die Codefolgen von der Länge η . Dann beträgt jeweils $A^0 = \eta$, da das Produkt an jeder Chip-Position +1 ergibt. Nach Welch beträgt das kleinstmögliche Maximum der Kreuzkorrelationsfunktion dann

$$\bar{K}_{s_1, s_2}^{max} = \lceil \frac{\eta}{\sqrt{\eta+1}-1} \rceil = \lceil \frac{\eta \cdot (\sqrt{\eta+1}+1)}{\eta+1-1} \rceil = \lceil \sqrt{\eta+1}+1 \rceil.$$

Diese Eigenschaft konnte für Kasami-Codemengen für alle hier relevanten Werte von $n \in \{4, 6, 8, 10\}$ gezeigt werden, d.h. Kasami-Codefolgen erreichen diese von Welch definierte untere Schranke. Mit $\eta = 2^n - 1$ ergibt sich dann:

$$\bar{K}_{s_1, s_2}^{max} = \lceil \sqrt{\eta+1}+1 \rceil = \sqrt{2^n} + 1 = 2^{\frac{n}{2}} + 1.$$

Es werden im Rahmen dieser Arbeit vorwiegend Kasami-Codefolgen verwendet, da diese bei gleicher Codelänge bessere Kreuzkorrelationseigenschaften als Gold-Codes aufweisen. Zudem weisen verschiedene Paare von Kasami-Codes derselben Kasami-Codemenge stets dieselbe Korrelationsgüte auf.

3.6.7 Auswahl der Grundfrequenz

In diesem Abschnitt sollen die verschiedenen Faktoren für die Auswahl der Grundfrequenz f_0 noch einmal übersichtlich aufgelistet werden.

- Die verwendete Grundfrequenz f_0 kann aufgrund des Abtasttheorems höchstens die Hälfte der Abtastrate f_A betragen, also $f_0 \leq \frac{f_A}{2}$. Details zum Abtasttheorem finden sich z.B. in [20], Kapitel 2.4.

- Eine hohe Lautstärke ist für die Messung auf großer Distanz sehr wichtig. Die Grundfrequenz sollte so gewählt sein, dass die verwendeten Lautsprecher die Grundfrequenz gut wiedergeben können. Dazu kann eine Messung des charakteristischen Wiedergabespektrums des Lautsprechers durchgeführt werden.
- Eine hohe Grundfrequenz f_0 bewirkt, dass eine Sinus-Halbwellen im Signal nur noch aus sehr wenigen Abtastpunkten besteht. Z.B. bestehen Halbwellen der Grundfrequenz $f_0 = \frac{f_A}{4}$ nur aus zwei Abtastpunkten. Schon eine Verschiebung innerhalb des Korrelationsintervalls um nur einen Abtastpunkt bewirkt dann, dass die Kreuzkorrelation kein verwertbares Ergebnis mehr liefert. Die Verwendung solcher hoher Frequenzen ist daher ungeeignet, falls sich Sender bzw. Empfänger in ständiger Bewegung befinden.
- Höhere Frequenzen führen zu einem stärkeren Einfluss von Reflexion und Brechung bei der Ausbreitung der akustischen Welle. Die Ermittlung des Empfangszeitpunktes durch Korrelation wird dadurch ungenauer. Insbesondere für Richtungsmessungen führt dies zu großen Fehlern, da sich kleine Messfehler hier sehr stark auswirken (siehe auch Kapitel 3.3.1).
- Eine höhere Grundfrequenz erhöht die Genauigkeit der Korrelation dadurch, dass das Maximum der Kreuzkorrelationsfunktion *lokal* eindeutiger ist. Die dem Maximum benachbarten Korrelationswerte sind dann bereits deutlich geringer. (Man spricht auch von einem ‘scharfen Peak’.) Bei niedrigeren Frequenzen ist der Peak sehr flach. Die um das Maximum liegenden Korrelationswerte sind also annähernd gleich groß. Störgeräusche können dann bewirken, dass ein Korrelationswert maximal wird, der sich in der direkten Umgebung des theoretischen Maximums befindet. In Experimenten mit $f_0 \approx \frac{f_A}{20}$ entstanden hierdurch Fehler in der Größenordnung von bis zu zwei Abtastpunkten. (*Anmerkung*: Dieser Effekt ist hinnehmbar verglichen mit den verschiedenen Problemen, die bei der Verwendung hoher Grundfrequenzen auftreten.)
- Verschiedene Frequenzen werden von Menschen und Tieren in der Umgebung als unterschiedlich störend empfunden. Insbesondere können sehr hohe Frequenzen beim Menschen ein Gefühl von Unbehagen auslösen (siehe auch Kapitel 3.7.3).
- *Hinweis*: Der Einfluss der Bewegungsunschärfe (Kapitel 3.6.8) verhält sich *invariant* gegenüber einer Veränderung der Grundfrequenz. Eine höhere Grundfrequenz erhöht zwar die Anfälligkeit der Korrelation bei Bewegung, jedoch wird auch die Schemalänge ψ in gleichem Maße kürzer.

Für die Auswahl der Grundfrequenz müssen die oben genannten Punkte für die spezifische Anwendung untersucht werden. Beispielsweise können in einem Szenario, in dem während der Messung Sender und Empfänger still stehen, durchaus hohe Frequenzen verwendet werden. Dabei muss allerdings der höhere Einfluss von Brechung und Reflexionen berücksichtigt werden. In einem Anwendungsfall, bei dem vor allem die maximale *Reichweite* der Distanzmessungen entscheidend ist, sollte besonders eine Frequenz mit maximaler Wiedergabeleistung gewählt werden.

3.6.8 Bewegungunschärfe

Sind die Roboter in Bewegung, so ändert sich normalerweise ständig die Distanz zwischen ihnen. Wenn sich die Distanz *während des Aussendens* eines Codeschemas verändert, so wird sich beim Empfänger die Wellenform von der am Sender ausgesendeten Welle unterscheiden. Ursache hierfür ist der Dopplereffekt.

Angenommen die relative Geschwindigkeit zwischen den Robotern v_r sei während der Zeit des Aussendens konstant. Durch den Dopplereffekt verändert sich die effektive Grundfrequenz des empfangenen Codeschemas. Die Folge ist, dass die empfangene Wellenform, je nach Vorzeichen von v_r , auseinandergezogen oder zusammengestaucht wird.

Für die Korrelation macht sich dieser Effekt negativ bemerkbar. Ist die Differenz zwischen gesendeter und empfangener Schemalänge in der Größenordnung einer halben Wellenlänge der Grundfrequenz f_0 , so kann der Empfangszeitpunkt weniger zuverlässig bestimmt werden. Die Differenz sollte daher möglichst geringer bleiben:

$$\frac{\eta \cdot \kappa \cdot c}{f_0} - \frac{\eta \cdot \kappa \cdot (c - v_r)}{f_0} = \frac{\eta \cdot \kappa \cdot v_r}{f_0} < \frac{1}{2} \cdot \frac{c}{f_0} \Rightarrow v_r < \frac{c}{2 \cdot \eta \cdot \kappa}$$

c ist dabei die Schallgeschwindigkeit, η die Codelänge und κ die Anzahl von Sinuswellen-Durchgängen pro mChip. Man beachte, dass die resultierende Forderung *unabhängig von f_0* ist.

Abbildung 3.14 zeigt ein Beispiel. Die Verschiebung durch den Dopplereffekt betrug hierbei eine halbe Wellenlänge pro gesendetem Codeschema, d.h. es liegt der Fall $v_r = \frac{c}{2 \cdot \eta \cdot \kappa}$ vor. Man sieht, dass die Peaks der Korrelation geringer sind. In einer Situation mit erheblichen Störgeräuschen und kleinen Nutzsignallautstärken kann dies die Ermittlung des Empfangszeitpunktes erschweren.

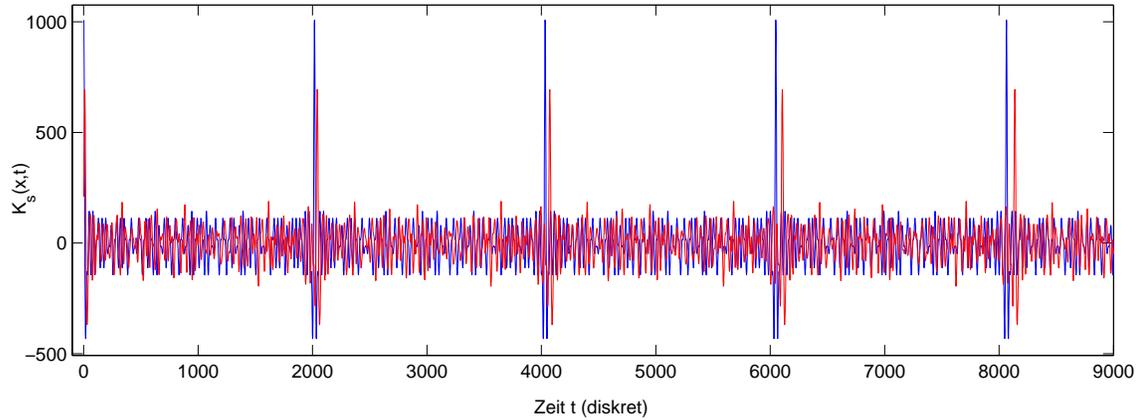


Abbildung 3.14: blau: Korrelation eines Schemas bei $v_r = 0$; rot: Korrelation desselben Schemas mit $v_r = \frac{c}{2 \cdot \eta_s \cdot \kappa}$. Man erkennt den geringeren Peak bei der Korrelation. Die relative Verlagerung der Peaks nach rechts spiegelt die allmähliche Vergrößerung der Distanz wider.

3.6.9 Signalvorverarbeitung

Eine Signalvorverarbeitung kann sinnvoll sein, um den Einfluss von Störgeräuschen zu verringern. Insbesondere ist es sinnvoll, den Einfluss von einzelnen, sehr lauten Störgeräuschen zu verringern. Bei der Korrelation können wenige Abtastpunkte den Wert der *gesamten* Korrelationssumme stark verändern. Ein Korrelationswert kann also auch durch kurze und laute Störgeräusche stark verfälscht werden.

Eine einfache und effektive Methode ist es, das Eingangssignal oberhalb einer gewissen Amplitude abzuschneiden. Ist beispielsweise bekannt, dass die Summe aus Rauschen und allen gleichzeitig empfangenen Codeschemata selten lauter ist als die Amplitude a^{max} , kann man alle Eingangswerte die deutlich größer sind als a^{max} auf a^{max} oder auch auf 0 herabsetzen. Die Information der empfangenen Schemata zu diesen Zeitpunkten geht dabei verloren. Dieser Verlust ist hinnehmbar, falls es deutlich weniger solcher herabgesetzten Werte gibt als Abtastpunkte pro Codeschema. Der Gewinn ist eine größere Robustheit gegenüber einzelnen, lauten Störgeräuschen.

Eine andere Methode der Signalvorverarbeitung ist die Reduktion der Lautstärke des eigenen, ausgesendeten Codeschemas. Dies ist dann von Vorteil, wenn mehrere Codeschemata gleichzeitig ausgesendet werden. Senden z.B. Roboter A und B gleichzeitig ihr Schema aus, empfängt Roboter A sein eigenes Schema normalerweise deutlich lauter als das Schema von B. Für die Kreuzkorrelation zwischen Eingangssignal und dem Schema von B kann dieser Umstand die maximal messbare Distanz stark begrenzen (siehe Kapitel 3.6.6.2).

Um die Lautstärke des eigenen Codeschemas zu reduzieren kann Roboter A von seinem Eingangssignal sein *eigenes* Schema *subtrahieren*. Dies ist nur dann praktikabel, wenn die Lautstärke und Position des *eigenen* Schemas zu jedem Zeitpunkt bekannt ist. Wenn die relative Position von Lautsprecher und Mikrofon unverändert bleibt, können Position und Lautstärke des eigenen Schemas einmal, z.B. im Vorhinein, ermittelt werden.

In der Praxis konnte diese Methode nicht überzeugen, da die Wellenform und die genaue Position des eigenen Schemas zu stark in Abhängigkeit der (verhallten) Umgebung variieren.

3.6.10 Reduktion des Berechnungsaufwands

Mit Hilfe verschiedener Ansätze kann der Berechnungsaufwand bei der Evaluation der KKF verringert werden.

Der wichtigste Ansatz ist der, dass nicht zu *jedem* Zeitpunkt die KKF ausgewertet werden muss, wenn die Schemata regelmäßig, z.B. fortlaufend, gesendet werden.

Angenommen, Roboter B sendet durchgehend sein Schema s_B der Länge χ . Roboter A registriert den Empfang von s_B zum Zeitpunkt t_0 . Folgt das Aussenden der Codes einem bekannten Plan, so kann Roboter A den nächsten Empfangszeitpunkt *abschätzen*. Sendet B zum Beispiel fortlaufend sein Schema, kann Roboter A den nächsten Empfangszeitpunkt mit $t_1 = t_0 + \chi$ abschätzen.

Der Zeitpunkt t_1 variiert nur in dem Maße, wie sich die Distanz zwischen A und B zwischen diesen beiden Messungen verändert. Angenommen, die maximale, relative Geschwindigkeit v_r^{max} zwischen A zu B sei bekannt. Die maximale Distanzänderung zwischen zwei Aussendezeitpunkten beträgt dann $\chi \cdot v_r^{max}$. Roboter A muss daher lediglich im Intervall $t_1 \pm v_r^{max} \cdot \chi \cdot \frac{1}{c}$ die Kreuzkorrelationsfunktion evaluieren. c ist dabei die Schallgeschwindigkeit. In der Praxis ist dieses Intervall sehr klein verglichen mit χ .

Liegen die Aussendezeitpunkte zeitlich zu weit auseinander, so kann alternativ die Kenntnis über die maximale Distanz d^{max} zwischen den Robotern verwendet werden, um das auszuwertende Intervall der KKF zu reduzieren. Das auszuwertende Intervall der KKF reduziert sich dann auf $t_1 \pm \frac{d^{max}}{c}$.

3.7 Aspekte und Probleme in der Praxis

3.7.1 Quantisierung, digitale Abtastung

Entscheidend für die Genauigkeit der verschiedenen Ansätze ist die Abtastfrequenz f_0 . Aus dieser kann man direkt die *Auflösung* der Distanzmessung berechnen zu $\varrho = \frac{c}{f_0}$. Besonders bei der Richtungsmessung ist eine hohe Auflösung bei der Bestimmung von ds erforderlich.

Mit verschiedenen Techniken ist es möglich, die Auflösung durch *Interpolation* künstlich zu erhöhen. Der genaue Empfangszeitpunkt bei der Auswertung der KKF kann dabei z.B. durch das Verhältnis zwischen dem Wert des Maximums und der dem Maximum benachbarten Korrelationswerte bestimmt werden. Dies ist deshalb möglich, weil der Korrelationspeak die Gestalt einer in y -Richtung stark gestreckten Sinus-Halbwellen hat. Auf Details wird hier nicht eingegangen.

3.7.2 Uhrendrift und Synchronisierung

Die in autonomen mobilen Robotern verwendeten Uhren basieren normalerweise auf einem Quarzkern. Die Vorteil dieser Uhren liegt vor allem im sehr günstigen Preis, kleinen Ausmaßen und geringem Energieverbrauch. Bei der Distanzmessung wirken sich Fehler in den Uhren stark auf die Distanz aus.

Angenommen, der Uhrendrift der verwendeten Uhr betrage ϵ (ohne Einheit), d.h. in einem Zeitraum von dt beträgt der absolute Uhrenfehler $\epsilon \cdot dt$. Der resultierende Fehler bei der Distanzmessung beträgt $c \cdot \epsilon \cdot dt$.

Ein Zahlenbeispiel soll die Wichtigkeit synchroner Uhren aufzeigen: Der Uhrendrift einer für gewöhnlich in Robotern verwendeten Uhr beträgt etwa $\epsilon = 1 \cdot 10^{-5}$, teilweise auch mehr. Das entspricht einem Uhrenfehler von etwa 26 Sekunden pro Monat. In einem Zeitraum von $dt = 10 \text{min}$ beträgt der Fehler bei der Distanzmessung durch diesen Uhrendrift bereits *über 2 Meter*. Eine fortlaufende Synchronisation ist dann erforderlich.

Im Folgenden werden drei Ansätze für die Synchronisation der Roboteruhren diskutiert.

3.7.2.1 Einmalige Synchronisation

Zu einem Zeitpunkt wird ein Ereignis, z.B. ein bestimmtes Geräusch, hervorgehoben, dass alle Roboter zeitgleich erreicht und dessen Empfangszeitpunkt genau bestimmt werden kann. Jeder Roboter merkt sich diesen Zeitpunkt als Referenzzeitpunkt. Das Problem hierbei ist, dass ab dem Zeitpunkt der Synchronisation die Uhren teilweise erheblich driften können. Große Fehler bei der Distanzmessung sind die Folge.

3.7.2.2 Fortlaufende Synchronisation

Wenn zwei Roboter A und B *zeitgleich* die Distanz zueinander bestimmen, kann indirekt über die beiden gemessenen Distanzen eine Synchronisation durchgeführt werden.

Erfolgen die Messungen nicht zeitgleich entsteht ein Fehler durch die zwischen den Messungen erfolgte Distanzänderung zwischen A und B. Stehen die Roboter still besteht diese Forderung also nicht.

Im Allgemeinen liegen Lautsprecher und Mikrophon eines Roboters nicht an derselben Stelle. Dadurch ist die wahre Distanz von A nach B bzw. von B nach A *nicht* identisch. Dies führt zu Fehlern bei der Distanzmessung und der Synchronisation.

Man beachte, dass hier von Messungen in der Ebene ausgegangen wird. Der *auf den Untergrund projizierte* Abstand zwischen Lautsprecher und Mikrophon sei $d_{\overrightarrow{LM}}$. Liegen Lautsprecher und Mikrophon projiziert auf derselben Position, dann *ist* der gemessene Abstand in beiden Richtungen identisch.

Sei $d_{\overrightarrow{LM}} = 0$. Abbildung 3.15 zeigt ein Zeitdiagramm der Sende- und Empfangsvorgänge. Die Offset zwischen den beiden Uhren, t_{offset} , ist zunächst unbekannt. t_d ist die Zeitdauer, die das Signal tatsächlich benötigt, um von A nach B zu gelangen. Die Distanz zwischen den Robotern ist dann $t_d \cdot c$. Durch den Drift der Uhren verändert sich t_{offset} allmählich. Sowohl t_d als auch t_{offset} können die Roboter berechnen:

Dazu müssen beide den eigenen Sendezeitpunkt t_S und den Empfangszeitpunkt t_R kennen. Über einen separaten Kommunikationskanal werden dann alle Sende- und Empfangszeiten t_{R_A} , t_{S_A} , t_{R_B} und t_{S_B} ausgetauscht. Beide Roboter können dann t_d wie folgt berechnen:

$$t_d = \frac{1}{2}[(t_{R_B} - t_{S_B}) + (t_{R_A} - t_{S_A})].$$

Man beachte, dass hier nur *Zeitdifferenzen* lokaler Zeiten verwendet werden. Z.B. ist eine Berechnung nach der Formel $t_d = t_{R_A} - t_{S_B}$ *nicht korrekt*, weil die einzelnen Zeitpunkte jeweils *in lokaler Zeit* gemessen sind. Der Zusammenhang zwischen den beiden lokalen Zeitskalen wird über den Offset der Uhren hergestellt, den beide Roboter ebenfalls berechnen können:

$$t_{offset} = t_{R_A} - t_d - t_{S_B} = -(t_{R_B} - t_d - t_{S_A}).$$

Die Berechnung von t_{offset} entspricht dabei der eigentlichen Synchronisation. Bei darauf folgenden Messungen kann das korrigierte t_{offset} verwendet werden, um *direkt* aus dem Empfangszeitpunkt die Distanz zu berechnen. Eine erneute Synchronisation sollte in regelmäßigen Abständen erfolgen, um den Effekt des Uhrendrifts gering zu halten.

Die Synchronisation bei diesem Ansatz hat nur Geltung für Roboter A und Roboter B. In einem Team von Robotern muss jedes Paar von Robotern separat synchronisiert werden. Jeder Roboter muss also zu jedem der übrigen Roboter des Teams einen separaten Wert t_{offset} führen.

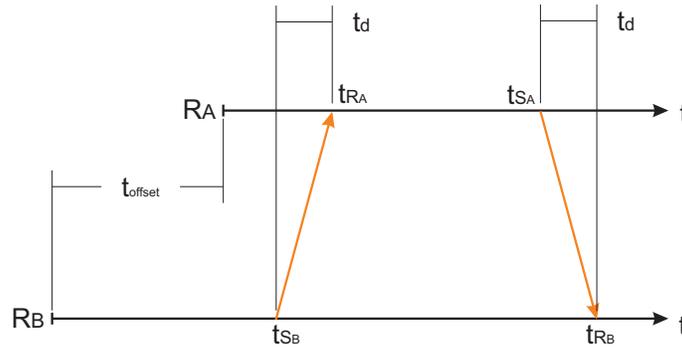


Abbildung 3.15: Berechnung von effektiver Distanz und Uhren-Offset bei gleichzeitiger Distanzmessung, siehe Text. R_A zeigt den lokalen Zeitstrahl von Roboter A, R_B den von Roboter B.

Die Synchronisation unterliegt noch einem Fehler, falls $d_{\overrightarrow{LM}} \neq 0$ ist. Dieser Effekt ist herausrechenbar, falls man den effektiven Distanzunterschied in Richtung des anderen Roboters aufgrund anderer Sensorik bestimmen kann. Dies kann z.B. mit Hilfe einer bekannten Orientierung im Raum erfolgen. Liegt eine solche Information *nicht* zuverlässig vor, beträgt der Fehler bei der Distanzmessung maximal $e_{LM}^{max} = d_{\overrightarrow{LM}}$. Für viele Anwendungen ist dieser Fehler hinnehmbar verglichen mit den großen Fehlern durch den Uhrendrift.

Zusammenfassend kann der durch Uhrendrift entstehende Fehler durch eine ständige Synchronisation *sehr gering* gehalten werden.

3.7.2.3 Synchronisation mit NTP

Eine Synchronisation der Roboteruhren kann auch mit Hilfe des *Network Time Protocols* (kurz: *NTP*, siehe [24]) durchgeführt werden, falls die Roboter über eine drahtlose Netzwerkverbindung verfügen. Die durch NTP ermittelten Werte sind allerdings oft aufgrund von Schwankungen der effektiven Übertragungszeiten zu ungenau, so dass sie lediglich als Näherung verwendet werden können.

Dieser Näherungswert kann allerdings sehr von Nutzen sein. Ein Roboter kann die grobe Näherung verwenden, um ohne jegliche Kenntnis des Empfangszeitpunktes diesen zumindest grob schätzen zu können. Die *initiale* Ermittlung des Empfangszeitpunktes kann dadurch beschleunigt werden.

3.7.3 Beeinträchtigung der Umgebung

Ein letzter Aspekt der Praxis soll nicht unerwähnt bleiben: Der Einsatz von akustischen Aktuatoren kann für einen Einsatz in einer Umgebung mit Mensch oder Tier ungeeignet sein.

Eine Anwendung im Alltagsbereich ist nur bei Verwendung von Frequenzen im nicht hörbaren Bereich sinnvoll (Infra- oder Ultraschall). Dies erfordert normalerweise andere Lautsprecher und Mikrophone. Alle hier diskutierten Ansätze sind grundsätzlich auch für den Einsatz mit Ultraschallwellen geeignet. Allerdings sind Ultraschallwellen deutlich stärker durch Brechung und Reflexion beeinträchtigt.

Kapitel 4

Ansätze zur kooperativen Selbstlokalisierung

4.1 Einleitung

Als Selbstlokalisierung wird der Prozess bezeichnet, bei dem ein Roboter seine eigene Position aufgrund von Messungen ermittelt. Typische Messungen zur Selbstlokalisierung sind z.B. GPS-Messungen, optische Beobachtung der Umgebung mit Hilfe einer Kamera oder der Einsatz von Laserscannern.

Die genaue Kenntnis der eigenen Position ist im Prinzip in allen Szenarien beim Einsatz autonomer, mobiler Roboter von hoher Wichtigkeit. Hohe Genauigkeit und Zuverlässigkeit sollten gewährleistet sein. In der Praxis können die eigenen Messungen hierfür nicht ausreichend sein.

Bei der kooperativen Selbstlokalisierung sollen, im Rahmen dieser Arbeit, relative Positionsinformationen zu anderen Robotern verwendet werden, um die eigene Position genauer und zuverlässiger bestimmen zu können. Die Kooperation besteht hier zum einen darin, dass die Roboter durch ein aktives Verfahren (z.B. aktive Schall-Lokalisation, siehe Kapitel 3) ihre relativen Positionen ermitteln können. Darüber hinaus senden sich die Roboter untereinander ihre eigenen, geschätzten Positionen sowie entsprechende Konfidenzwerte über eine drahtlose Netzwerkverbindung zu.

Jeder Roboter der Gruppe ermittelt zu einem Zeitpunkt folgende Daten:

- Die geschätzte *eigene* Position mit zugehöriger Konfidenz, aufgrund von Umweltmessungen sowie

- gemessene Richtungen und Distanzen zu anderen Robotern.

Hinzu kommen folgende Daten, die jeder Roboter von den anderen Robotern der Gruppe mitgeteilt bekommt:

- Die geschätzten Positionen und zugehörige Konfidenzen der anderen Roboter sowie
- gemessene Richtungen und Distanzen der anderen Roboter *untereinander*.

Ein Problem der Praxis wird sein, dass der Austausch von Daten zwischen den Robotern Zeit benötigt. Beim Einsatz von drahtlosen Computer-Netzwerken liegt die Latenz in einer Größenordnung von bis zu 500ms.

4.1.1 Beschränkung auf ein bestimmtes Szenario

Eine allgemeine Betrachtung dieses umfangreichen Themas würde den Rahmen dieser Arbeit sprengen. Es werden daher einige Annahmen über das Szenario getroffen.

Die folgenden Ansätze werden für die Verwendung mit relativen Positionsinformationen aus der aktiven, akustischen Lokalisation (Kapitel 3) diskutiert. Es wird daher stets davon ausgegangen, dass

- Richtungsmessungen sehr ungenau sind,
- Distanzmessungen genau sind und
- keine relativen Orientierungen gemessen werden können.

Weiter wird angenommen, dass die Konfidenzen der geschätzten Positionen nur grob geschätzt werden können. Einige Ansätze werden diese Information daher gar nicht verwenden.

Darüber hinaus wird davon ausgegangen, dass sich alle Roboter *nur in der Ebene* bewegen. Die Ansätze wären überwiegend auch auf den dreidimensionalen Raum übertragbar, allerdings wird jeweils ein Eingangsdatum mehr zur eindeutigen Bestimmung der Position nötig sein. Das Szenario des Roboterfußballs, nach dem diese Arbeit ausgerichtet ist, erfüllt diese Forderung.

Die Ansätze werden auch im Hinblick auf die Verwendung eines *Partikelfilters* zur Selbstlokalisierung entwickelt. Die Idee ist, dass ein unabhängig funktionierender Partikelfilter zur Selbstlokalisierung durch den kooperativen Prozess zusätzliche Eingaben erhält. Die Überlegungen zur vermittelnden Ausgleichung sowie geometrische Überlegungen sind aber zunächst allgemein.

4.1.2 Überblick über die verwendeten Ansätze

In den folgenden Kapiteln werden verschiedene Ansätze zur Integration der relativen Positionsinformationen in die Selbstlokalisierung behandelt.

Der erste Ansatz verwendet eine paarweise Auswertung von gemessene Distanzen und Richtungen, jeweils zwischen zwei Robotern des Teams. In Abhängigkeit der Konfidenzwerte der geschätzten Roboterpositionen werden hier direkt Partikel zur Selbstlokalisierung bewertet oder neu eingestreut.

Der folgende Abschnitt 4.3 beschäftigt sich mit der vermittelnden Ausgleichung der verschiedenen Positionen und Messungen. Bei einer Gesamtausgleichung werden alle geschätzten Positionen und Messungen ‘in einem Guss’ ausgeglichen. Der Ansatz ist dann sinnvoll, wenn die Positionen aller Roboter zur Erhöhung der Genauigkeit nur noch leicht korrigiert werden müssen. Kommt es aber vor, dass einige geschätzte Positionen *völlig* falsch sind, so müssen diese identifiziert und aus der Ausgleichung entfernt werden.

Weitere Ansätze in Abschnitt 4.3.2 verwenden Untergruppen von Robotern, deren geschätzte Positionen am besten durch die Messungen *bestätigt* wurden. Diese geschätzten Positionen werden dann als Referenzpunkte verwendet. Die Positionen der übrigen Roboter werden dann *rein geometrisch* relativ zu diesen Referenzpunkten berechnet. Diese Ansätze werden am besten dem Fall gerecht, dass einige Roboter gut lokalisiert sind, während andere Roboter völlig falsch lokalisiert sind. Zudem ist die Berechnung deutlich weniger aufwendig. Die Verwendung von Referenz-Untergruppen der Größe n wird getrennt für die Fälle $n = 2$, $n = 3$ sowie $n > 3$ behandelt werden.

4.2 Paarweise Auswertung der Messungen

In diesem Abschnitt wird gezeigt, wie gemessene Distanzen und Richtungen zu anderen Robotern direkt in einen Partikelfilter zur Selbstlokalisierung integriert werden können. Für diesen Ansatz werden Positionen von Robotern immer nur paarweise verglichen. Vereinfachend wird daher zunächst angenommen, dass Roboter A nur die relativen Positionsinformationen zu Roboter B auswertet. Falls A relative Positionsinformationen zu weiteren Robotern messen konnte, wird jede dieser Messungen nacheinander wie im Folgenden beschrieben ausgewertet.

Das Grundproblem bei der Bewertung der Partikel durch relative Positionsinformationen besteht darin, dass a priori völlig unbekannt ist, ob ein Roboter gut oder

schlecht lokalisiert ist. Passen die relativen Positionsinformationen nicht zu den geschätzten Roboterpositionen ist daher unbekannt, welcher der beiden Roboter schlecht lokalisiert ist, oder ob gar beide Roboter schlecht lokalisiert sind. Doch auch im Falle einer *bestätigenden* Messung kann nicht sicher davon ausgegangen werden, dass beide Roboter tatsächlich gut lokalisiert sind. Es kann ebenso sein, dass völlig falsche Positionen *zufällig* durch die Messung bestätigt wurden.

Für eine paarweise Auswertung ist es daher nötig, dass zu den geschätzten Positionen zusätzlich Konfidenzwerte vorliegen, die die Zuversicht eines Roboters in die eigene Position widerspiegeln. Ein Konfidenzwert sei jeweils durch eine Zahl zwischen 0 und 1 gegeben, der die Wahrscheinlichkeit der Korrektheit der zu schätzenden Position widerspiegelt.

Die praktische Ermittlung bzw. Schätzung der Konfidenz muss an das jeweilige Szenario angepasst werden. Im Allgemeinen ist die Berechnung der Konfidenz schwierig und von vielen Größen abhängig, wie z.B. die verstrichene Zeit seit der letzten Sichtung einer Landmarke.

Bei einer paarweisen Auswertung mit bekannten Schätzwerten für die Konfidenzen ergeben sich dann verschiedene Situationen.

Eine von einem anderen Roboter geschätzte Position mit zu niedriger Konfidenz sollte nicht verwendet werden. Hierfür sollte eine minimale Konfidenz p^{min} vorgegeben werden.

Liegt die Konfidenz für die eigene Position über einem bestimmten Wert p^{max} , ist es normalerweise wünschenswert diese Position nicht weiter zu verändern. Auch in diesem Falle wird also die geschätzte Position von B nicht verwendet.

In den übrigen Fällen gilt also $p_A < p^{max}$ und $p_B > p^{min}$. Intuitiv ist klar, dass Roboter A nur dann die Messung zu B verarbeiten sollte, falls die Positionskonfidenz von B höher ist als die eigene, also $p_B > p_A$. In der Praxis sollte aufgrund der recht unzuverlässigen Schätzung der Konfidenz zusätzlich ein minimaler Faktor μ oder eine minimale Differenz λ gefordert werden, also $p_B > p_A \cdot \mu$ bzw. $p_B > p_A + \lambda$. Der Fall, dass A gut lokalisiert ist, aber die Konfidenz von B fälschlich etwas höher liegt, kann dadurch abgefangen werden.

Im Folgenden wird kurz gezeigt, wie die gemessene Distanz oder Richtung zu einer als korrekt angenommen Position in einen Partikelfilter zur Selbstlokalisierung integriert werden kann.

4.2.1 Bewertung von Partikeln

Partikel können im Partikelfilter aufgrund des Unterschieds zwischen dem gemessenen Abstand und ihrem tatsächlichen Abstand zur bekannten Position bewertet werden. Dieser Unterschied kann entweder als absoluter Streckenfehler oder als relativer Fehler modelliert werden. Normalerweise ist die zweite Variante sinnvoller, da derselbe absolute Fehler bei einer kleinen realen Distanz normalerweise schwerer wiegen sollte, als bei einer sehr großen realen Distanz. Im Anhang B findet sich eine Darstellung dieser Bewertungsfunktionen.

4.2.2 Einstreuung neuer Partikel

Das Neubewertung von Partikeln im Partikelfilter birgt das Risiko, dass aufgrund falscher Positionsschätzungen oder falscher Konfidenzschätzungen eine zuvor gute Position verloren geht.

Werden neue Partikel an passenden Stellen eingestreut, anstatt die Partikel neu zu bewerten, so hat der Vorgang mehr unterstützende Wirkung. Es werden in den Folgeiterationen dann automatisch diejenigen Partikel durch den Filter vervielfältigt bzw. aufgewertet, die durch darauffolgende Umgebungsmessungen bestätigt werden. Dadurch, dass keine Partikel neu bewertet werden, kann eine bereits gute Position nicht verloren gehen.

Die Einstreuung von neuen Partikeln ist dann praktikabel, wenn nur ein kleiner Bereich von Positionen in Frage kommt. Da die Orientierung des Roboters zunächst unbekannt ist, sollten an einer Position mehrere Partikel mit unterschiedlichen Orientierungen eingestreut werden.

4.3 Vermittelnde Ausgleichung

4.3.1 Gesamtausgleichung

Bei einer Gesamtausgleichung nach vermittelnden Beobachtungen werden alle Positionen und realen Distanzen ‘in einem Guss’ geschätzt. Die Eingaben sind dabei die geschätzten Positionen der Roboter sowie alle verfügbaren Messungen von relativen Positionen. Außerdem kann für jede Messung zusätzlich ein Konfidenzwert angegeben werden. Das Resultat der Ausgleichung ist dann ein Vektor

mit allen Korrekturwerten für die Positionen, die so genannten *Verbesserungen*. Die geschätzten Positionen werden also im Zuge der vermittelnden Ausgleichung korrigiert.

Das Problem einer Gesamtausgleichung liegt darin, dass eine grob falsche Schätzung einer Position erkannt werden muss. Diese darf dann nicht in den Prozess mit einbezogen werden. Die Identifikation dieser grob falschen Positionen, der so genannten ‘Ausreißer’, kann aber in der Praxis aufgrund sehr ungenauer Positionskonfidenzen schwierig sein.

Zu Erkennung von Ausreißern existieren so genannte ‘Ausreißertests’. Das sind Verfahren, die im Zuge der Ausgleichung Ausreißer durch stochastische Hypothesen über den a priori zu erwartenden Fehler identifizieren. Sind Ausreißer identifiziert, muss die Berechnung anhand der jetzt reduzierten Eingangsdaten erneut durchgeführt werden. Die Ausgleichung wird also zu einem iterativen Prozess:

- 1. Führe eine Ausgleichung über alle verbleibenden Eingangsdaten aus.
- 2. Falle es mindestens einen Ausreißer gibt:
 - Entferne den größten Ausreißer aus der Menge der Eingangswerte.
 - Fahre mit Schritt 1 fort.

Die Ausreißer müssen deshalb einzeln aus der Menge entfernt werden, da schon ein Ausreißer die gesamte Ausgleichung so sehr verzerren kann, dass andere Eingangsdaten *fälschlich* als Ausreißer identifiziert werden können.

Der *Rechenaufwand* einer großen Ausgleichung ist sehr hoch, da große Matrizen in den Dimensionen von Messdaten und Parametern zu invertieren sind. Auf mobilen Robotern mit knapp bemessener Rechenkapazität ist die Berechnung daher schnell nicht mehr praktikabel, besonders im Hinblick auf Ausreißer und die dadurch notwendigen Neuberechnungen.

Der Ansatz wurde zugunsten der in den nächsten Abschnitten folgenden Ansätze nicht weiter verfolgt. Diese werden besser und einfacher der Situation gerecht, dass einige Roboter völlig falsch lokalisiert sind. Zudem erfordern diese Ansätze deutlich weniger Rechenzeit.

4.3.2 Untergruppen als Positionsreferenzen

Die hier behandelten Ansätze stützen auf der Annahme, dass zu einem Zeitpunkt in einer Gruppe mehrerer Roboter immer wenigstens eine Untergruppe an Robotern gefunden werden kann, bei denen die geschätzten Positionen sehr gut durch die relativen Positionsmessungen bestätigt werden. In RoboCup-Fußballspielen der Aibo-Liga ist diese Annahme sehr häufig erfüllt.

Die geschätzten Positionen dieser Roboter werden dann zunächst in einer ‘kleinen’ Ausgleichung ausgeglichen. Die ausgeglichenen Positionen dienen dann als Referenz, um die Positionen der übrigen Roboter *geometrisch* direkt zu bestimmen.

In Kapitel 3.3.1 wurde erläutert, dass Richtungsmessungen mit Hilfe akustischer Lokalisation recht ungenau und störanfällig sind. Distanzen können mit guter Genauigkeit gemessen werden. Um die kooperative Selbstlokalisierung kompatibel mit den Verfahren der akustischen Lokalisation (Kapitel 3) zu halten, werden die folgenden Ansätze auch die Situation ohne Richtungsmessungen untersuchen. Außerdem werden für die folgenden Ansätze keine Positionskonfidenzen benötigt, welche in der Praxis nur schwierig und ungenau bestimmt werden können.

4.3.2.1 Methode des besten Positions-Paares

In diesem sehr einfachen Ansatz wird das Paar zweier Roboter gesucht, bei dem die gemessene Distanz am besten mit den geschätzten Positionen zusammenpasst. Die geschätzten Positionen dieser beiden Roboter werden dann direkt als geometrische Referenz für alle übrigen Roboter verwendet. Dieser Test muss bei n Robotern maximal $\binom{n}{2} = \frac{1}{2}n(n+1)$ mal durchgeführt werden, nämlich dann, wenn zu jedem Roboterpaar eine Distanzmessung existiert.

Da die relativen Orientierungen der Roboter zueinander im Allgemeinen nicht gemessen werden können, kann die Position eines dritten Roboters noch nicht eindeutig über die Distanzen zu den beiden Referenzpositionen bestimmt werden, sondern es existieren in der Ebene stets genau zwei mögliche Positionen. Auch die zusätzliche Verwendung von Richtungsmessungen bringt hier keine Eindeutigkeit, da die Orientierungen der Roboter als unbekannt anzunehmen sind.

Abbildung 4.1 skizziert die Situation. Die Punkte R_1 und R_2 wurden als Referenzpunkte ausgewählt. Die Distanzen zu einem dritten Roboter an der Position N sind mit d_{1N} bzw. d_{2N} bezeichnet. Die Position des dritten Roboters kann mit den gegebenen Distanzen aber auch bei N' liegen.

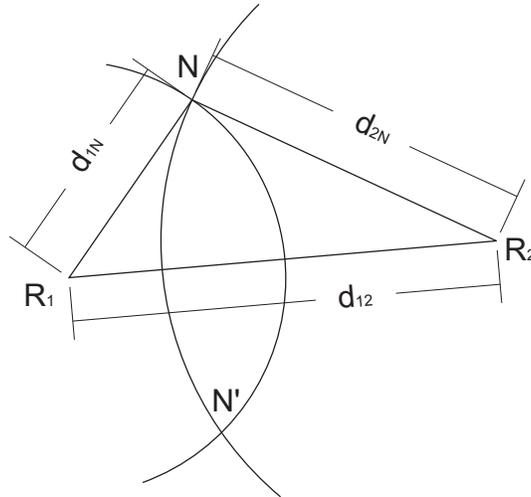


Abbildung 4.1: *Uneindeutige Positionsberechnung anhand zweier Distanzen zu Referenzpunkten.*

Die Zweideutigkeit der Position ist bei Verwendung eines Partikelfilters allerdings unproblematisch. Entweder werden an den beiden möglichen Positionen neue Partikel eingestreut. Alternativ hierzu können die Partikel neu bewertet werden, indem der minimale Abstand zu einem der beiden geometrisch möglichen Punkte als Abweichungswert verwendet wird (Bewertung der Partikel gemäß Anhang B).

Das Problem bei der Verwendung von nur zwei geschätzten Roboterpositionen liegt darin, dass es häufig vorkommen kann, dass zwei *falsche* Positionen nur *zufällig* durch die Messung gestützt werden. Die Folgen sind insofern fatal, weil andere Roboter ihre möglicherweise sehr guten Positionen unnötig verlieren können.

Um diese Fehlerquelle zu reduzieren, werden in den jetzt folgenden Ansätzen *mehr als zwei* Positionen als Referenzpunkte gesucht. Dadurch reduziert sich allerdings auch die Chance, *überhaupt* geeignete Referenzpunkte zu finden.

4.3.2.2 Methode des besten Positions-Tripels

Die Verwendung von drei Referenzpunkten in der Ebene bietet Vorteile gegenüber der Verwendung von zwei Referenzpunkten: Einerseits kann mit höherer Sicherheit davon ausgegangen werden, dass drei Roboter tatsächlich gut lokalisiert sind, wenn ihre geschätzten Positionen durch die gemessenen Distanzen untereinander bestätigt werden. Zum anderen können die Positionen der anderen Roboter relativ zu den Referenzpunkten nun *eindeutig* trianguliert werden, falls Distanzen zu allen drei Referenzpunkten bekannt sind.

Für die Qualität der Triangulation ist es entscheidend, dass das durch die drei Referenzpositionen beschriebene Dreieck drei möglichst gleichgroße Innenwinkel aufweist. Liegen die drei Punkte hingegen (annähernd) auf einer Geraden, so liegt eine Singularität vor. Die Triangulation kann dann nur noch anhand von zwei Referenzpunkten erfolgen.

Auch hier gehen die gemessenen Richtungen nicht in die Berechnung ein, da von ungenauen Orientierungen und ebenso ungenauen Richtungsmessungen ausgegangen werden muss.

Für das Auffinden des besten Positions-Tripels kommen verschiedene Varianten in Frage. Für eine möglichst *exakte* Berechnung müsste jedes in Frage kommende Positionstripel anhand der entsprechenden Messwerte ausgeglichen werden. Das Positionstripel mit der geringsten Summe an Abweichungen zwischen ausgeglichener und geschätzter Position (den so genannten Restklaffen) ist dann als Referenz zu verwenden.

Da unter Umständen sehr viele solcher ‘kleinen Ausgleichungen’ vorzunehmen wären, wird stattdessen eine Näherung verwendet, die den Vorgang vereinfacht und die Berechnung deutlich beschleunigt.

Die Form des Dreiecks, das auf die Roboterpositionen eingepasst werden soll, ist durch die Distanzen bereits vorgegeben. Dadurch, dass keine Richtungen im Dreieck bekannt sind, gibt es allerdings stets zwei mögliche Formen des Dreiecks, nämlich zusätzlich die an einer beliebigen Seite *gespiegelte* Variante.

Angenommen, das Positionstripel P_1 , P_2 und P_3 ist zu testen. Das gegebene Dreieck und seine gespiegelte Kopie müssen auf diese Punkte eingedreht werden. Da die Distanzen jeweils zwischen zwei bekannten Robotern gemessen wurden, ist bereits klar, auf welche Position die jeweilige Ecke des Dreiecks fallen muss. Die Ecken des Dreiecks werden dementsprechend mit E_1 , E_2 und E_3 bezeichnet.

Vereinfachend wird nun E_1 auf P_1 gelegt. Das Dreieck wird so gedreht, dass die Strecke $\overline{E_1E_2}$ auf die Strecke $\overline{P_1P_2}$ fällt. Die Lage des Dreiecks ist in der Ebene dadurch bereits eindeutig bestimmt.

Abbildung 4.2 zeigt ein bereits auf diese Art eingedrehtes Dreieck E_i zusammen mit den geschätzten Roboterpositionen P_i .

Gemäß den Bezeichnungen in Abbildung 4.2 kann dieses ‘Eindrehen’ des Dreiecks

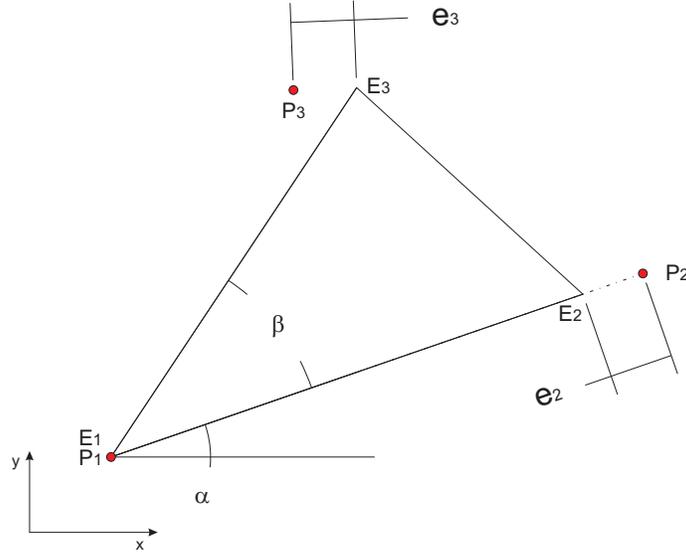


Abbildung 4.2: ‘Eindrehen’ des Dreiecks $(E_1E_2E_3)$ zur Evaluierung eines Positionstripels $(P_1P_2P_3)$. e_2 und e_3 bezeichnen die nach dem Eindrehen verbleibenden Abweichungen.

formelmäßig wie folgt beschrieben werden:

$$\beta = \arccos\left(\frac{d_{E_1,E_2}^2 + d_{E_1,E_3}^2 - d_{E_2,E_3}^2}{2 \cdot d_{E_1,E_2} \cdot d_{E_1,E_3}}\right),$$

$$\alpha_0 = \arctan\left(\frac{P_{2,y} - P_{1,y}}{P_{2,x} - P_{1,x}}\right),$$

$$E_2 = P_1 + \begin{pmatrix} \cos \alpha_0 \\ \sin \alpha_0 \end{pmatrix} \cdot d_{E_1,E_2},$$

$$E_3 = P_1 + \begin{pmatrix} \cos(\alpha_0 + \beta) \\ \sin(\alpha_0 + \beta) \end{pmatrix} \cdot d_{E_1,E_3}.$$

Die einzelnen Punkte P_i und E_i wurden hierbei als zweidimensionale Vektoren behandelt. Die d_{E_i,E_j} entsprechen den gemessenen Distanzen zwischen den entsprechenden Eckpunkten des Dreiecks. Die Berechnung von β nach dem Cosinussatz ist nur einmal durchzuführen, die übrigen Berechnungen für jedes zu testende Positionstripel.

Für jedes Positionstripel muss zusätzlich auch die gespiegelte Variante des Dreiecks geprüft werden. Die Berechnung unterscheidet sich durch die Symmetrie des Arcuscosinus nur durch den Winkel β . Dieser wird dann zu $\beta' = -\beta$. Im maximalen Fall ergeben sich also $2 \cdot \binom{n}{3}$ Tests, von denen aber jeder mit diesen wenigen Operationen auskommt.

Als Maß für die Abweichung wird jeweils die Fehlerquadratsumme der Differenzen an den Ecken E_2 und E_3 verwendet. Mit den Bezeichnungen in Abbildung 4.2 ergibt sich

$$e = \sum_{i=2}^3 |P_i - E_i|^2 = e_2^2 + e_3^2.$$

In der Ecke E_1 liegt konstruktionsbedingt keine Abweichung vor, da $E_1 = P_1$.

Wird kein Positionstripel mit ausreichender Passgenauigkeit gefunden, so wird der gesamte Vorgang abgebrochen, da keine sinnvolle Aussage über die tatsächliche Position der Referenzpunkte getroffen werden kann.

Wird ein Positionstripel mit ausreichender Genauigkeit gefunden, so wird das Dreieck *zusätzlich* durch eine vermittelnde Ausgleichung eingedreht. Die Distanzmessungen d_{E_i, E_j} gehen bei dieser Ausgleichung als konstant ein, die Positionen P_i werden im Zuge der Ausgleichung korrigiert. Die ausgeglichenen \hat{P}_i fallen dann mit den E_i bei minimaler Quadratsumme der Abweichungen zusammen. Ob das Dreieck in gespiegelter Form zu verwenden ist, wurde bereits ermittelt.

Durch die *Ausgleichung*, die zusätzlich zum oben geschilderten Verfahren erfolgt, soll die Lagegenauigkeit der Referenzpunkte erhöht werden. Im Folgenden werden kurz die nötigen Formeln zur Anwendung der vermittelnden Ausgleichung entwickelt. Weitere Informationen zur vermittelnden Ausgleichung sind in [25], Kapitel 4 zu finden.

Das Funktionale Modell stellt den Zusammenhang zwischen den Beobachtungen L und den Parametern X her. L ist dabei ein Vektor bestehend aus den insgesamt 6 Koordinaten der drei Roboterpositionen. Der Vektor der Parameter X beschreibt die Lage des Dreiecks. X besteht dabei aus der Position von E_1 sowie der Ausrichtung:

$$X = \begin{pmatrix} E_{1,x} \\ E_{1,y} \\ \alpha \end{pmatrix}$$

Die Ausgleichung erfordert zunächst gute *Näherungen* für die Parameter. Der entsprechende Vektor wird mit X^0 bezeichnet. Hierfür können die Werte aus der oben beschriebenen Näherung direkt übernommen werden, also

$$X^0 = \begin{pmatrix} E_{1,x}^0 \\ E_{1,y}^0 \\ \alpha_0 \end{pmatrix} = \begin{pmatrix} P_{1,x} \\ P_{1,y} \\ \arctan\left(\frac{P_{2,y} - P_{1,y}}{P_{2,x} - P_{1,x}}\right) \end{pmatrix}.$$

Das Funktionale Modell lautet dann

$$\begin{aligned}
 f_1 : 0 &= E_{1,x} - \hat{P}_{1,x} \\
 f_2 : 0 &= E_{1,y} - \hat{P}_{1,y} \\
 f_3 : 0 &= E_{1,x} + d_{E_1,E_2} \cdot \cos \alpha - \hat{P}_{2,x} \\
 f_4 : 0 &= E_{1,y} + d_{E_1,E_2} \cdot \sin \alpha - \hat{P}_{2,y} \\
 f_5 : 0 &= E_{1,x} + d_{E_1,E_3} \cdot \cos(\alpha + \beta) - \hat{P}_{3,x} \\
 f_6 : 0 &= E_{1,y} + d_{E_1,E_3} \cdot \sin(\alpha + \beta) - \hat{P}_{3,y}
 \end{aligned}$$

Da die Ausgleichung hier nicht linear ist, werden zusätzlich Näherungen für die Beobachtungen L^0 benötigt. Diese müssen die Gleichungen des Funktionalen Modells erfüllen. Hierfür werden die Ecken $(E_1 E_2 E_3)$ des eingedrehten Dreiecks verwendet, so wie oben geschildert.

$$L^0 = \begin{pmatrix} E_1^0 \\ E_2^0 \\ E_3^0 \end{pmatrix}$$

Die Designmatrix A_{ij} ist eine 6x3 Jacobi-Matrix der partiellen Ableitungen der f_i nach den Parametern X_j , ausgewertet an den Näherungsstellen L^0 und X^0 .

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & -d_{E_1,E_2} \cdot \sin(\alpha_0) \\ 0 & 1 & d_{E_1,E_2} \cdot \cos(\alpha_0) \\ 1 & 0 & -d_{E_1,E_3} \cdot \sin(\alpha_0 + \beta) \\ 0 & 1 & d_{E_1,E_3} \cdot \cos(\alpha_0 + \beta) \end{pmatrix}$$

Sind für die einzelnen Roboterpositionen Konfidenzwerte bekannt, so können diese in Form von Standardabweichungen $\sigma_{P_{i,x}}$ bzw. $\sigma_{P_{i,y}}$ in die Ausgleichung mit eingehen. Positionen mit kleinerer Standardabweichung gehen dann mit höherem Gewicht in die Ausgleichung ein. Bei den beobachteten Roboterpositionen kann von unkorrelierten Beobachtungen ausgegangen werden. Die Fehler der einzelnen Werte werden also als statistisch unabhängig angenommen.

Hinzu kommt die sogenannte a-priori-Standardabweichung, die den typischen Beobachtungsfehler beschreibt. Da dieser Wert für Roboterpositionen schwer abzuschätzen ist, kann für die Berechnung auch das arithmetische Mittel der einzelnen Standardabweichungen verwendet werden.

Mit den gegebenen Werten ergeben sich die Varianz-Kovarianz-Matrix Σ , die Kofaktormatrix Q , sowie die Gewichtsmatrix P wie folgt:

$$\Sigma = \begin{pmatrix} \sigma_{P_{1,x}}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{P_{1,y}}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{P_{2,x}}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{P_{2,y}}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{P_{3,x}}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{P_{3,y}}^2 \end{pmatrix}$$

$$Q = \frac{1}{\sigma_0^2} \Sigma$$

$$P = Q^{-1} = \sigma_0^2 \cdot \Sigma^{-1}$$

Liegen keine Werte für die einzelnen Standardabweichungen vor, so wird die Gewichtsmatrix P als Einheitsmatrix der Größe 6 definiert.

Die eigentliche Ausgleichung erfolgt nun gemäß der folgenden Formeln:

$$\begin{aligned} l &= L - L^0 \\ N &= A^T P A \\ n &= A^T P l \\ d\hat{x} &= N^{-1} n \\ \hat{X} &= X^0 + d\hat{x} \\ \hat{L} &= L^0 + A \cdot d\hat{x} \end{aligned}$$

Als Ergebnis erhält man die Parameter der Lage des ausgeglichenen Dreiecks \hat{X} , sowie die ausgeglichenen Roboterpositionen \hat{L} .

Der Berechnungsaufwand liegt vor allem in der Invertierung der 3x3-Matrix N , sowie in den verschiedenen Matrixmultiplikationen. Durch eine geeignete Implementierung kann der Rechenaufwand verringert werden. Beispielsweise kann das Gleichungssystem $N \cdot d\hat{x} = n$ (die sogenannte Normalgleichung) analytisch aufgelöst werden, wodurch die Invertierung von N wegfällt.

Durch ungenaue Näherungswerte L^0 tritt ein Linearisierungsfehler auf. Dieser Fehler kann verringert werden, indem die Ausgleichung nach einem iterativen Schema wiederholt wird. Für die Näherungen der Parameter der folgenden Iteration gilt $X^{0,2} = \hat{X}$. Die Näherungen der Beobachtungen $L^{0,2}$ müssen erneut berechnet werden, so dass sie zusammen mit $X^{0,2}$ die Gleichungen des funktionalen Modells erfüllen.

Da die Ausgleichung ohnehin nur für die drei Roboterpositionen berechnet wird, die sehr gut zu den Beobachtungen passen, ist der Zuschlag zu den Parametern $d\hat{x}$ normalerweise bereits in der ersten Iteration sehr klein. Weitere Iterationen bringen oft nur noch Veränderungen, die deutlich kleiner sind als der zu erwartende Messfehler. Im Regelfall genügt hierbei also eine einzige Iteration. Dadurch bleibt auch der Rechenaufwand gering.

Die drei ausgeglichenen Roboterpositionen $\hat{L} = \left(\hat{P}_1 \ \hat{P}_2 \ \hat{P}_3 \right)^T$ fungieren nun als Referenzpunkte. Damit können nun Positionen weiterer Roboter durch Triangulation berechnet werden. Die Position eines vierten Roboters kann dabei *eindeutig* bestimmt werden, falls die Distanzen zu allen drei Referenzpunkten bekannt sind. Die Position ist *zweideutig* bestimmbar, falls nur Distanzen zu zwei Referenzpunkten bekannt sind, siehe Abbildung 4.1.

Die anhand der Referenzpunkte berechneten Positionen der Roboter können entweder direkt übernommen werden. Alternativ können die triangulierten Positionen aber auch zur Partikelbewertung gemäß Abschnitt 4.2.1 verwendet werden. Zusätzlich können an der errechneten Position gemäß Abschnitt 4.2.2 neue Partikel eingestreut werden.

4.3.2.3 Methode des besten Positions-Tupels

Die Überlegungen aus dem vorherigen Abschnitt sind in ähnlicher Form auf die Verwendung größerer Untergruppen von Referenzpositionen anwendbar.

Hierbei ist aber zu beachten, dass die Form des Referenz-Vielecks durch die Distanzmessungen bereits *überbestimmt* ist. Die Form kann dann entweder genähert werden, oder im Rahmen der Ausgleichung ebenfalls geschätzt werden. Die Spiegelung des Vielecks muss, wie beim Dreieck, zusätzlich berücksichtigt werden.

Der Ansatz wird hier nicht weiter vertieft, da im Rahmen dieser Arbeit ohnehin nur kleine Gruppen von Robotern berücksichtigt werden.

Kapitel 5

Implementierung und Ergebnisse der Schall-Lokalisation

5.1 Übersicht

Im Folgenden werden die Details der Implementierung und Ergebnisse der Schall-Lokalisation gezeigt. Die Ausführungen beschränken sich hierbei vor allem auf den dritten Ansatz (KKF, Kapitel 3.6). Abschnitte, die speziell auf diesen Ansatz eingehen, sind 5.4, 5.7, 5.10, 5.12 und 5.13. Die übrigen Kapitel behandeln allgemein Eigenschaften der akustischen Wellenausbreitung.

5.2 Verwendete Plattform

Als Plattform diente der SONY Aibo, Modell ERS-7. Abbildung 5.1 zeigt ein Bild des ERS-7. Dieser Roboter besitzt einen kleinen Lautsprecher in der Brust. Seitlich am Kopf sind zwei Mikrophone mit weiter Öffnungscharakteristik (omnidirektionales Mikrophon, Raummikrophon) installiert. Dies ist für die Lokalisation von Vorteil, da Schall aus beliebigen Richtungen mit nahezu gleicher Lautstärke aufgenommen wird.

5.2.1 Plattformspezifische Geometrie



Abbildung 5.1: Foto des SONY®Aibo ERS-7. Man erkennt den Lautsprecher in der Brust sowie das Mikrofon am linken Ohr. Copyright SONY®.

Der Mittelpunkt des Lautsprechers L befindet sich im lokalen Robotersystem bei $y = 0$, die beiden Mikrophone bei $y_M = \pm 47,712\text{mm}$. Die Basis, d.h. der Abstand zwischen den beiden Mikrophenen beträgt somit $b^0 = 95,424\text{mm}$. Die projizierte Basis b ist die Projektion der Verbindung der beiden Mikrophone auf den Untergrund. Durch verschiedene Lagen des Kopfes gilt allgemein $b \leq b^0$. Der Referenzpunkt R des Mikrophenpaares ist der Schnittpunkt zwischen der Verbindungsgeraden der beiden Mikrophone und der Ebene $y=0$ (siehe auch Abbildung 3.2). Der geometrische Nullpunkt des Roboters liegt auf dem Schnittpunkt zwischen der Ebene $y=0$ und der Drehachse des Hals-Drehgelenks. Dieser Punkt liegt etwa um 2cm in x -Richtung hinter dem Mittelpunkt des Lautsprechers. Abbildung 5.2 zeigt eine Ansicht des ERS-7 von oben mit Nullpunkt, R und den beiden Achsenrichtungen.

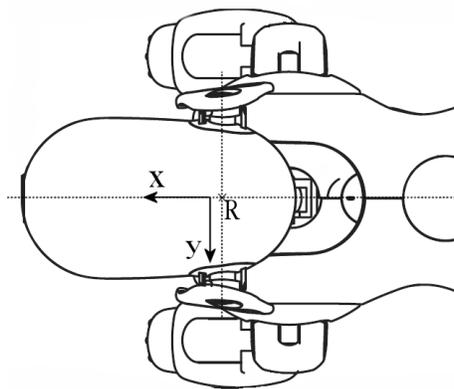


Abbildung 5.2: Ansicht des ERS-7 von oben mit Referenzpunkt der Mikrophone R und den Achsen. Der Nullpunkt ist der Schnittpunkt der Achsen.

Die Drehung des Kopfes erfolgt um eine Drehachse die ebenfalls in der Ebene

$y = 0$ liegt. Ist der Kopf des Roboters relativ zum Rumpf um die lokale z -Achse gedreht, liegt also R nach wie vor in der Ebene $y = 0$. Bei der Messung der Distanz werden daher die Eintreffzeitpunkte am linken und am rechten Ohr gemittelt. Durch die *Drehung* des Kopfes entsteht also kein Fehler bei der Distanzmessung. Die Richtungsmessung erfolgt immer relativ zum möglicherweise gedrehten Kopf.

Ein Fehler entsteht durch die Neigung des Kopfes. Projiziert man R und L auf den Untergrund, so erkennt man einen durch die Kopfneigung bedingten Unterschied (siehe Abbildung 5.3).

Die Lage des Kopfes hat auch Auswirkungen auf die Richtungsmessung. Die Länge der projizierten Basis b kann sich verringern, falls sich die beiden Mikrophone aufgrund der Kopfhaltung in unterschiedlicher Höhe befinden. Mit Hilfe des Kinematischen Modells kann die Lage des Kopfes und damit der momentane Wert von b ungefähr ermittelt werden. Kleinere projizierte Basen b haben ungenauere Messungen zur Folge. Im äußersten Fall liegen die beiden Mikrophone direkt übereinander. Die Länge der projizierten Basis entspricht dann 0. Eine Richtungsmessung in der dann senkrecht dazu stehenden Ebene ist nicht möglich.

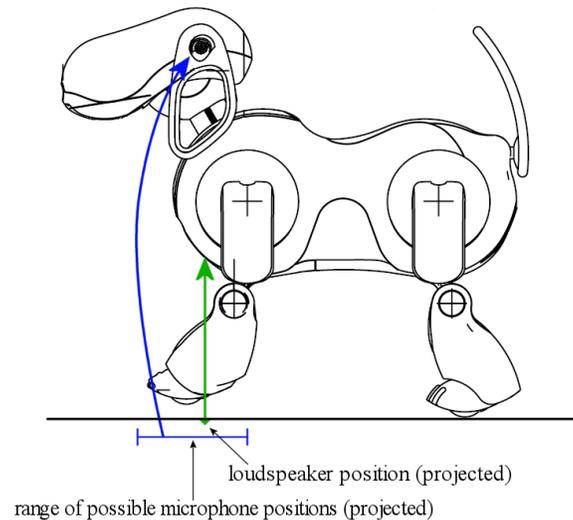


Abbildung 5.3: *Distanzfehler durch die Neigung des Kopfes. Die projizierte Position von R liegt um bis zu 6cm vor L und um bis zu 3cm hinter L .*

5.2.2 Plattformspezifische Eigenschaften der digitalen Klangverarbeitung

Die Aibo-Roboter verfügen über den OPEN-R-Bus, der als Schnittstelle zwischen Soft- und Hardwareseite dient. Über diesen Bus werden neben verschiedenen Sensorwerten und Stellwerten auch die Sounddaten von den Mikrofonen empfangen bzw. zum Lautsprecher gesendet.

Die Abtastrate der Mikrophone beträgt $f_A = 16kHz$. Die einzelnen Abtastpunkte werden mit einer Auflösung von 16 Bit digitalisiert. Die von den Mikrofonen kommenden digitalisierten Sounddaten sind im PCM-Format mit zwei Spuren gespeichert (stereo), das linke Mikrofon auf der ersten, das rechte Mikrofon auf der zweiten Spur.

Das Klangformat der digitalen Sounddaten, die an den Lautsprecher gesendet werden, kann separat gewählt werden. Die möglichen Formate sind 8 Bit Auflösung bei 8kHz Abtastrate oder 16 Bit Auflösung bei 16kHz Abtastrate. Die Klangwiedergabe erfolgt mit einer einzelnen Klangspur, d.h. mono. Für die Klangwiedergabe wurde zugunsten einer höheren Flexibilität das Format 16 Bit bei 16kHz gewählt. Bei 8kHz lassen sich insbesondere höhere Frequenzen nicht fein genug modellieren.

Neben den Sounddaten können über den OpenR-Bus auch verschiedene Werte für die Soundausgabe gesetzt werden. Die Lautstärke wurde stets auf maximaler Einstellung angesteuert. Zusätzlich kann die Soundausgabe jederzeit mit einem ‘Mute’-Schalter auf Software-Ebene unterdrückt werden.

Die Mikrophonsignale können mit Hilfe des sogenannten ‘Automatic Level Control’ (ALC) automatisch angehoben bzw. abgesenkt werden, je nachdem wie laut der einfallende Schall ist. Da die Nutzdaten in der Praxis deutlich unter dem maximalen Pegel liegen, kann ALC ohne den Verlust verwertbarer Information deaktiviert werden. Für die Berechnungen der verschiedenen Ansätze ist normalerweise nur das *Verhältnis* zwischen Nutzsignal und Rauschen bzw. Störgeräuschen entscheidend, welches durch ALC unverändert bleibt. Verwendet man aber zusätzlich auch die *absolute* Lautstärke, also z.B. die absolute Größe eines Korrelationswertes, so muss zum Erhalt der Vergleichbarkeit ALC deaktiviert werden.

5.3 Einsatzszenario

Die Schall-Lokalisation wird für den Einsatz bei einem RoboCup-Fußballspiel in der Sony-Aibo-Liga implementiert. Die zentralen Eigenschaften dieser speziellen

Umwelt sind

- ein ebener Untergrund,
- Teams von bis zu 5 Robotern,
- mögliche Distanzen von bis zu 12 Metern und
- eine maximale Bewegungsgeschwindigkeit von etwa $50 \frac{cm}{s}$.

Der ebene Untergrund ist deshalb wichtig, da näherungsweise davon ausgegangen werden kann, dass sich die verwendeten Lautsprecher und Mikrophone alle auf derselben Höhe befinden. Dadurch müssen sämtliche geometrische Berechnungen, wie Triangulation und Ausgleichungen, lediglich in der Ebene ausgeführt werden. Im Raum wäre zur Triangulation jeweils eine Eingangsmessung mehr erforderlich, was den Einsatz des Systems deutlich einschränken würde.

Effektiv liegen die Mikrophone um bis zu 10cm höher als die Lautsprecher. Bei einer realen Distanz von 1m bewirkt dies einen Fehler von nur 0,5cm. Für höhere Distanzen nimmt der Fehler quadratisch ab. Der Effekt wird daher vernachlässigt.

5.4 Auswahl der Codefolgen

Die Auswahl der Codefolgen erfolgte nach verschiedenen Kriterien. Wie in Kapitel 3.6.6 erläutert werden Kasami-Codefolgen verwendet. Dazu wurde zunächst eine Kasami Codemenge mit $n = 6$ verwendet. Diese Menge besteht aus 8 Codes der Länge 63.

Kürzere Codes haben zu schlechte Eigenschaften bei der Kreuzkorrelation und sind daher nicht zu verwenden. Längere Codes kommen grundsätzlich auch in Frage. Die modulierten Schemata dürfen allerdings nicht zu lang werden, damit der Einfluss der Bewegungsunschärfe (siehe Kapitel 3.6.8) nicht zu groß wird. Zusätzlich muss der Berechnungsaufwand beachtet werden.

Im Anhang D ist die gesamte Menge der 8 erzeugten Codes zu finden, von denen 5 für das Roboter-Team eingesetzt werden. Die Codemenge wurde durch das Erzeugerpolynom $x^6 + x + 1$ generiert. Die initialen Werte der Schieberegister betragen (000001). Die Codefolgen wurden mit Hilfe einer speziellen Bibliothek der Software MATLAB(tm) erstellt.

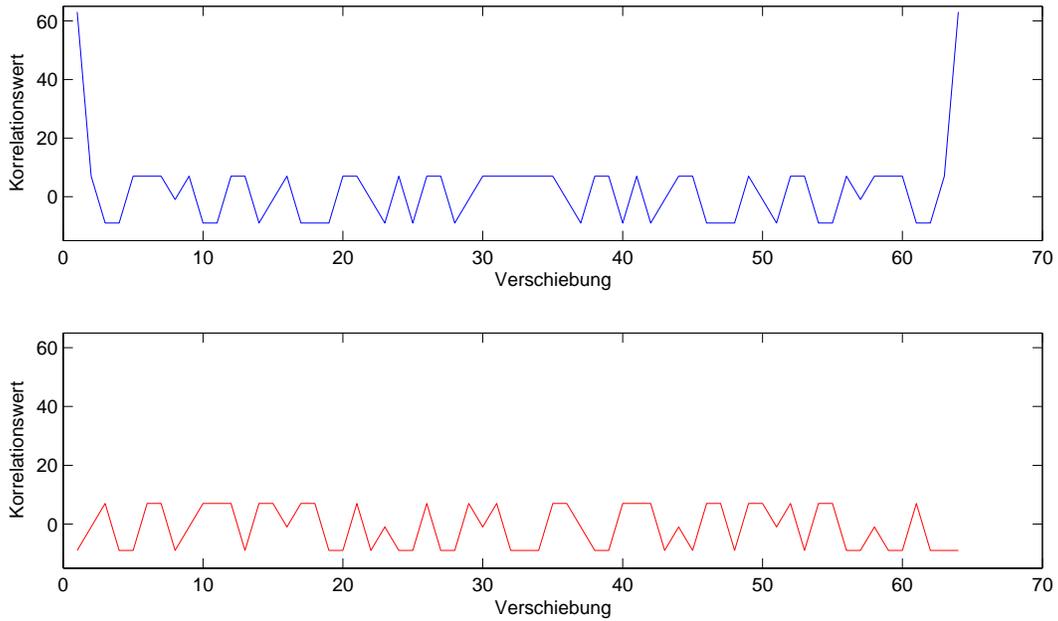


Abbildung 5.4: *Kasami-Codes der Länge 63. Oben: Autokorrelation eines Codes, unten: Kreuzkorrelation zweier Codes.*

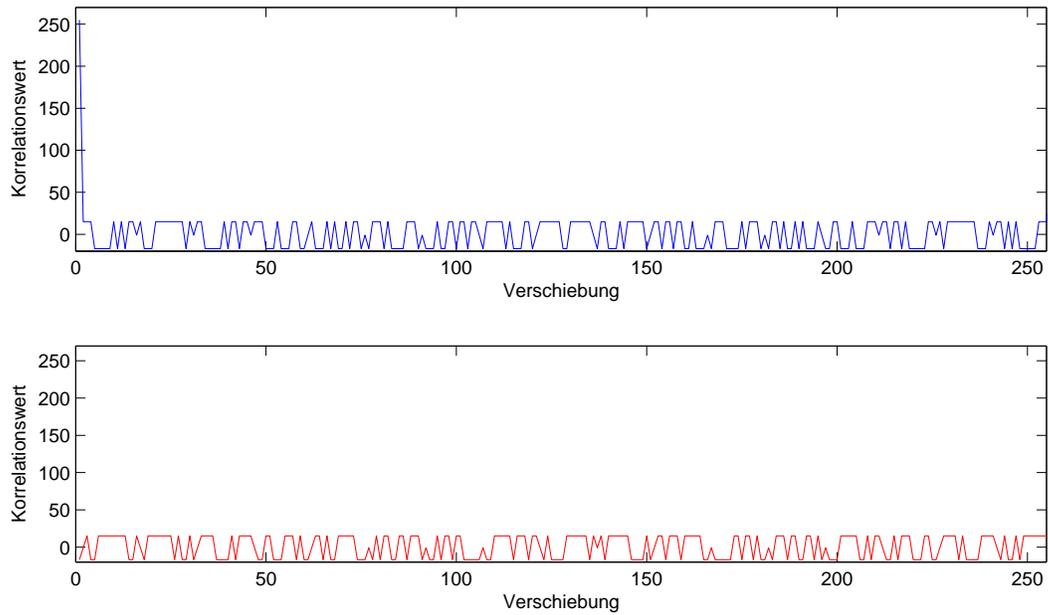


Abbildung 5.5: *Kasami-Codes der Länge 255. Oben: Autokorrelation eines Codes, unten: Kreuzkorrelation zweier Codes.*

Abbildung 5.4 zeigt eine typische Autokorrelationsfunktion eines Codes dieser Kasami-Codemenge, sowie eine typische Kreuzkorrelationsfunktion zweier Codes. Man kann erkennen, dass die Werte der Kreuzkorrelation und die Werte der verschobenen Autokorrelation vorteilhaft klein sind. Die Werte der Kreuzkorrelation und der verschobenen Autokorrelation liegen durchgehend im Bereich zwischen -9 und 7. Gemäß Kapitel 3.6.3 gilt also $\bar{K}^{max} = 9$, $A^{max} = 9$ und $A^0 = 63$. Das entscheidende Verhältnis von A^0 zu \bar{K}^{max} beträgt also genau 1:7. Gemäß Kapitel 3.6.6.2 ergibt sich für das maximale Lautstärkeverhältnis zwischen dem eigenen und einem fremden empfangenen Codeschema

$$q_v^{max} = \frac{A^0 - A^{max}}{2 \cdot \bar{K}^{max}} = \frac{63 - 9}{2 \cdot 9} = 3.$$

Zusätzlich wurde eine Kasami-Codemenge mit $n = 8$ erzeugt. ($n = 7$ ist durch die speziellen Anforderungen der Kasami-Codes nicht möglich.) Es entsteht dabei eine Menge von 16 Codes der Länge 255. Abbildung 5.5 zeigt eine typische Auto- und Kreuzkorrelation von Codes dieser Kasami-Codemenge. Man erkennt die im Verhältnis zur Codemenge mit $n = 6$ noch besseren Eigenschaften. Gemäß Kapitel 3.6.3 ist $\bar{K}^{max} = 17$, $A^{max} = 17$ und $A^0 = 255$. Das entscheidende Verhältnis von A^0 zu \bar{K}^{max} beträgt hier also 1:15. Gemäß Kapitel 3.6.6.2 ergibt sich

$$q_v^{max} = \frac{A^0 - A^{max}}{2 \cdot \bar{K}^{max}} = \frac{255 - 17}{2 \cdot 17} = 7.$$

Man erkennt einige Regelmäßigkeiten in Abhängigkeit von n , die hier hergeleitet werden sollen. (Man beachte, dass n per Definition immer gerade ist.) Die Eigenschaft

$$\bar{K}^{max} = A^{max} = 2^{\frac{n}{2}} + 1$$

wurde für Kasami-Codemengen für die relevanten Werte $n \in \{4, 6, 8, 10\}$ gezeigt und bereits in Kapitel 3.6.6.4 hergeleitet. Dadurch ergibt sich das Verhältnis von A^0 zu \bar{K}^{max} mit Hilfe der dritten binomischen Formel zu

$$\frac{A^0}{\bar{K}^{max}} = \frac{2^n - 1}{2^{\frac{n}{2}} + 1} = \frac{(2^{\frac{n}{2}} - 1)(2^{\frac{n}{2}} + 1)}{2^{\frac{n}{2}} + 1} = 2^{\frac{n}{2}} - 1.$$

Außerdem ergibt sich allgemein für das maximale Lautstärkeverhältnis

$$\begin{aligned} q_v^{max} &= \frac{A^0 - A^{max}}{2 \cdot \bar{K}^{max}} = \frac{(2^n - 1) - (2^{\frac{n}{2}} + 1)}{2 \cdot (2^{\frac{n}{2}} + 1)} \\ &= \frac{(2^{\frac{n}{2}} + 1)(2^{\frac{n}{2}} - 1)}{2 \cdot (2^{\frac{n}{2}} + 1)} - \frac{1}{2} = \frac{2^{\frac{n}{2}} - 1}{2} - \frac{1}{2} = 2^{\frac{n}{2}-1} - \frac{1}{2} - \frac{1}{2} = 2^{\frac{n}{2}-1} - 1. \end{aligned}$$

Diese Gleichung stellt also für Kasami-Codefolgen direkt einen Zusammenhang zwischen dem maximalen Lautstärkeverhältnis und der Codelänge dar. Eine Vervielfachung der Codelänge liefert somit ein etwa doppelt so großes maximales Lautstärkeverhältnis.

5.5 Auswahl der Grundfrequenz

Die Auswahl der Frequenz erfolgte nach den in Kapitel 3.6.7 genannten Kriterien:

In einem Experiment wurden zunächst die Wiedergabeleistungen in Abhängigkeit von Sinustönen unterschiedlicher Frequenzen ermittelt. Diese Töne wurden mit den Mikrofonen eines zweiten Aibos aufgenommen, um unter realistischen Bedingungen zu testen. Abbildung 5.6 zeigt das Ergebnis des Experimentes für Frequenzen zwischen 500Hz und 5000Hz. Frequenzen unterhalb von 500Hz nehmen stark an Lautstärke ab (hier im Bild nicht zu sehen). Töne mit Frequenzen von über 5000Hz weisen aufgrund der sehr wenigen Abtastpunkte pro Sinuswellen-Durchlauf große Anfälligkeiten bei der Korrelation auf und wurden daher nicht untersucht. (Die Abtastrate beträgt 16000Hz.) Außerdem ist die Verwendung sehr hoher Frequenzen wegen der Empfindlichkeit gegenüber Brechung und Interferenz nicht sinnvoll (siehe Abschnitt 5.6).

Man erkennt, dass der Verlauf der Kombination aus Lautsprecher und Mikrofon nicht linear ist. Frequenzen zwischen etwa 600Hz und 800Hz, sowie Frequenzen ab 3500Hz bieten gute Lautstärken. Dazwischen gehen die Lautstärken deutlich zurück.

Die Anfälligkeit höherer Grundfrequenzen gegenüber Brechung wurde in einem separaten Experiment gezeigt (Abschnitt 5.6). Deshalb, und wegen der höheren Störanfälligkeit bei der Korrelation (siehe oben), kommen hier als Grundfrequenz nur Frequenzen im Bereich zwischen etwa 600Hz und 800Hz in Frage.

Die Grundfrequenz wurde aus diesem Bereich so gewählt, dass die modulierten Codefolgen eine zeitliche Länge haben, die ein ganzzahliges Vielfaches von 32ms ist. Dies ist deshalb wünschenswert, da die Sounddaten im Aibo in Abschnitten der Länge 32ms verarbeitet werden. Ist die Länge des Schemas ein Vielfaches von 32ms, so ist die Verarbeitung der Sounddaten für die Implementierung erheblich einfacher und außerdem in der Ausführung effizienter.

Die Codefolgen besitzen Länge $\eta = 63$, und es werden $\kappa = 2$ Sinuswellen pro Chip verwendet. Gemäß Kapitel 3.6.6.3 ergibt sich für ein $\chi = 5 \cdot 32ms = 160ms$ langes Schema die Grundfrequenz von

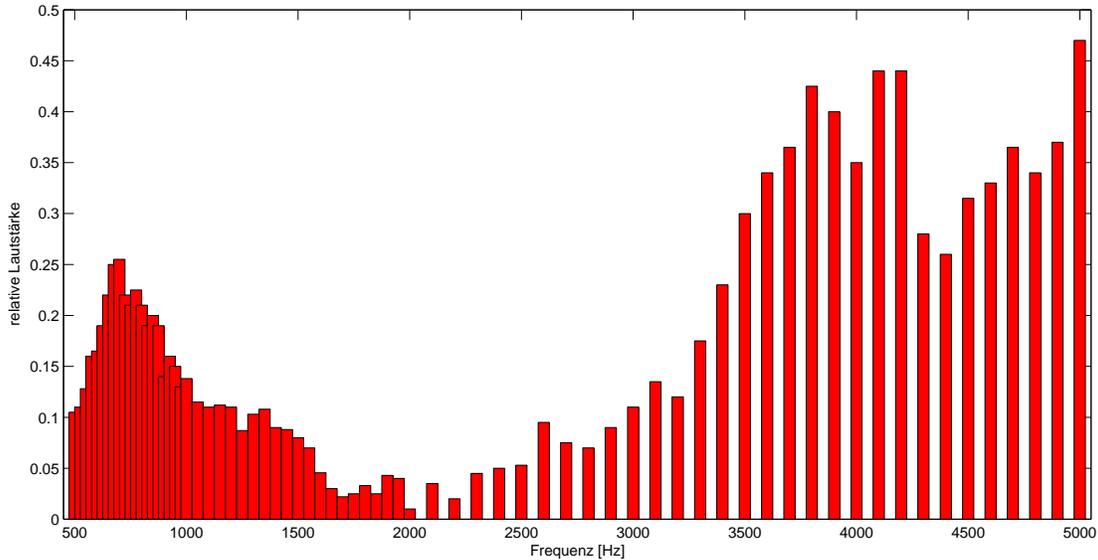


Abbildung 5.6: Experiment zur Ermittlung einer geeigneten Grundfrequenz: Gemessene Lautstärken von Sinustönen verschiedener Frequenzen.

$$f_0 = \frac{\eta \cdot \kappa}{\chi} = \frac{63 \cdot \kappa}{0.16s} = 787.5Hz.$$

5.6 Experiment: Brechung an Hindernissen

Ein einem Experiment soll der Einfluss von Hindernissen ermittelt werden, die sich auf dem Weg zwischen einem Sende- und einem Empfangsroboter befinden.

Zunächst besteht zwischen Sende- und Empfangsroboter eine freie Sichtverbindung. Dann wird ein dritter Aibo als Hindernis nahe vor dem sendenden Roboter aufgestellt. Nachdem dieser dritte Aibo wieder weggenommen wurde, wurde eine ca. 60cm breite und 0,8cm dicke Kunststoffplatte zwischen den Robotern aufgestellt, so dass die Sichtverbindung völlig blockiert war. Am Ende wurde diese Platte wiederum entfernt.

Dieses Experiment wurde jeweils für eine niedrige Grundfrequenz von $f_1 = 787.5Hz$ sowie eine hohe Grundfrequenz von $f_2 = 5 \cdot f_1 = 3937.5Hz$ durchgeführt.

Abbildung 5.7 zeigt die Abweichungen der Distanzmessung über die Zeit aufgetragen. Als Grundfrequenz wurde f_1 verwendet. Man erkennt deutlich die positiven Distanzabweichungen durch die Hindernisse. Der Aibo als Hindernis verursachte dabei einen Fehler von ca. 5cm, die Platte einen Fehler von ca. 10cm.

Abbildung 5.7 zeigt die Abweichungen desselben Experimentes bei Verwendung der Grundfrequenz f_2 . Die durch die Hindernisse hervorgerufenen Fehler betragen hier ca. 10cm und 21cm. In den Übergangsbereichen kann man kurze Schwankungen erkennen. Diese sind vermutlich auf Luftdruckunterschiede zurückzuführen, die beim Aufstellen bzw. Wegnehmen der Hindernisse entstanden. Höhere Frequenzen sind bei der Korrelation für solche Einflüsse ebenfalls anfälliger.

Anmerkung: In beiden Abbildungen erkennt man ein langsames Driften der Distanzabweichungen. Dieser Fehler ist dem Fehler durch Brechung überlagert und entsteht durch Uhrenfehler der beiden Roboter. Während der knapp 30 Sekunden, die das Experiment dauerte, entstand dadurch bereits ein Fehler von etwa 2cm. Wenn beide Roboter gleichzeitig senden und empfangen kann dieser Fehler durch eine regelmäßige Synchronisation eliminiert werden (siehe auch Kapitel 5.8 und 3.7.2).

Töne mit hohen Frequenzen besitzen zusammenfassend eine größere Anfälligkeit gegenüber Brechung und Bewegungsunschärfe. Daher wurde für alle weiteren Experimente mit einer Grundfrequenz zwischen 500Hz und 800Hz gearbeitet.

5.7 Auswahl der Methode der Modulation

Wie bereits in Abschnitt 3.6.5 diskutiert, können Phasensprünge durch den Lautsprecher normalerweise nur unzureichend wiedergegeben werden. In diesem Abschnitt wurden auch verschiedene Verfahren zur Modulation vorgestellt (siehe auch Abbildung 3.12).

Da die Frequenzmodulation bereits in der Theorie unzureichend ist, wurde diese nicht weiter untersucht. Alle übrigen Modulationstypen wurden in einem Experiment getestet und direkt miteinander verglichen. Dabei wurde ein Codeschema mit einer Grundfrequenz von 787.5Hz wiedergegeben. In Abbildung 5.8 sind jeweils die Aufnahmen bei Verwendung der unterschiedlichen Modulationstypen im Bereich eines Phasensprungs aufgetragen. Man erkennt zunächst die Reproduzierbarkeit der Ergebnisse an den beiden Phasensprüngen.

Die verschiedenen Ansätze aus Abbildung 3.12 sind in den Abbildungen 5.8 farblich kenntlich gemacht (siehe Unterschrift der Abbildung 5.8). Die gestrichelte Linie stellt den Soll-Verlauf des Phasensprungs dar. Zur besseren Übersicht wurde die Amplitude des Soll-Verlaufs grob an die Amplituden der Aufnahme angepasst.

Der Ansatz der eingefügten niederfrequenten Halbwelle ergibt bei der Kreuzkorrelation die schlechtesten Ergebnisse und wird daher nicht verwendet. Die anderen

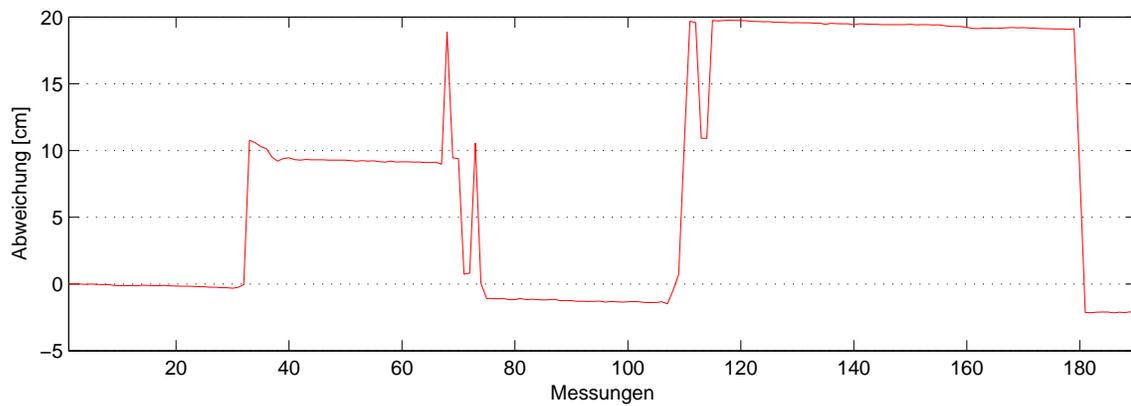
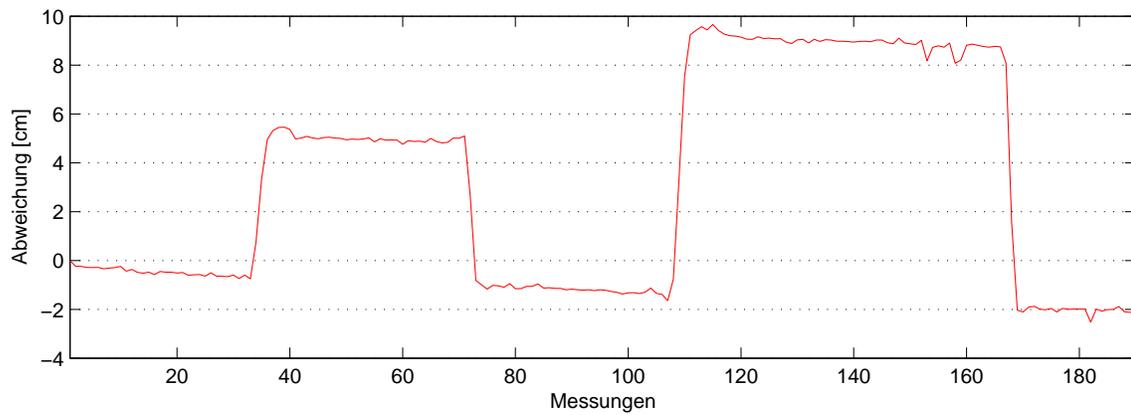


Abbildung 5.7: Gemessene Distanzabweichungen durch Hindernisse. Verwendete Grundfrequenz: oben: $f_1 = 787.5\text{Hz}$, unten: $f_2 = 5 \cdot f_1 = 3937.5\text{Hz}$.

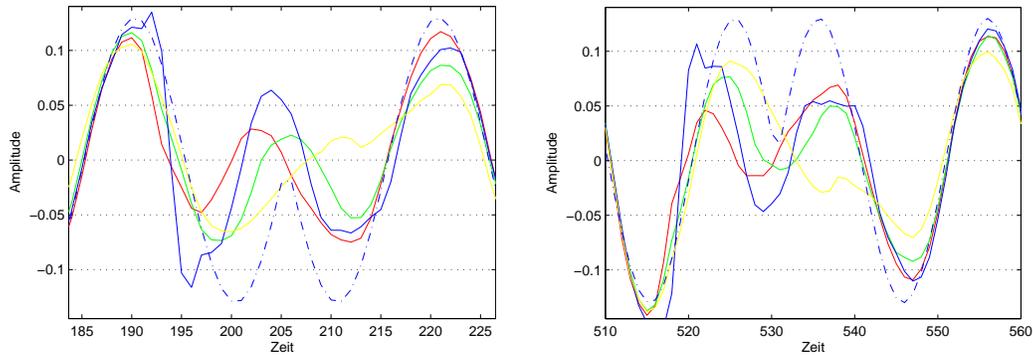


Abbildung 5.8: *Experiment zum Vergleich verschiedener Modulationsarten für Phasensprünge. rot: mit unveränderten Sprüngen, grün: mit kurzen Pausen, blau: mit höherfrequentem Puls, gelb: mit niederfrequentem Puls, blau gestrichelt: Soll-Verlauf exakter Phasensprünge.*

drei Ansätze liefern annähernd gleich gute Ergebnisse. Bei der Implementierung wurde sowohl mit direkten Phasensprüngen, als auch mit höherfrequenten Pulsen gearbeitet.

Anmerkung: Der genaue Wellenverlauf im Bereich der Phasensprünge wurde hier für den Aibo-Lautsprecher unter Verwendung der Grundfrequenz $f_0 = 787.5\text{Hz}$ gemessen. Bei Verwendung anderer Lautsprecher oder Grundfrequenzen können sich abweichende Verläufe ergeben.

5.8 Uhrendrift und Synchronisierung

Der Uhrendrift der Roboter wurde in einem weiteren Experiment untersucht. Es sollte dabei herausgefunden werden, ob der Uhrendrift zwischen einem bestimmten Paar von Robotern über einen längeren Zeitraum konstant verläuft. In diesem Fall könnte der durch den Uhrendrift auftretende Fehler nach einer initialen Synchronisation aus den Messungen herausgerechnet werden.

Das Experiment basiert darauf, dass die Tonwiedergabe auf Hardwareebene und die interne Uhr mit dem selben Taktgeber betrieben werden. Der Uhrenfehler kann also durch die Veränderungen der Frequenz eines durchgehenden Sinustons identifiziert werden. Ist die Uhr beispielsweise zu schnell getaktet, wird ein Sinuston der Sollfrequenz f_0 tatsächlich mit einer *etwas höheren* Frequenz wiedergegeben.

Im Experiment spielte ein Aibo Roboter über einen langen Zeitraum einen Sinuston der Frequenz $f_0 = 750\text{Hz}$ ab. Dieser Sinuston wurde über ein Mikrofon

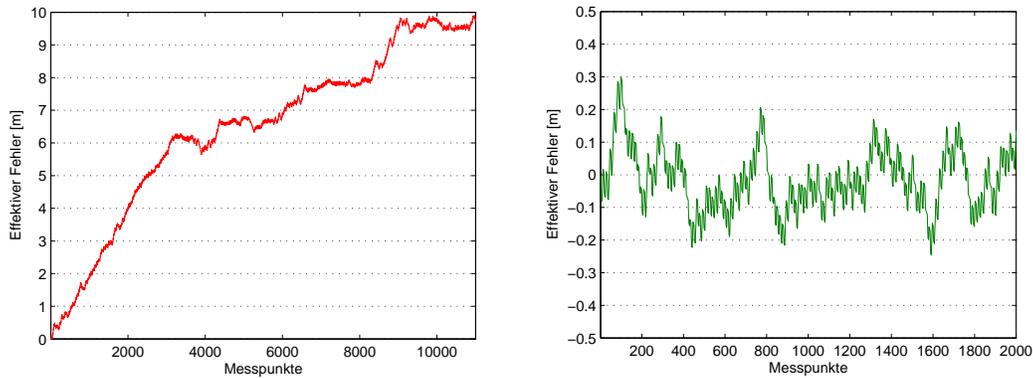


Abbildung 5.9: *Experiment zur Ermittlung der Distanzfehler durch den relativen Uhrenfehler. Die Messpunkte lagen 1.28 Sekunden auseinander. links: Verlauf des gesamten Experiments über fast 4 Stunden. rechts: residuale Abweichungen nach Modellierung eines konstanten Fehleranstiegs über die ersten 2000 Messpunkte (=43 Minuten).*

aufgezeichnet und später ausgewertet. Alle 1.28 Sekunden wurde eine Messung gemacht. Das gesamte Experiment dauerte ca. 4 Stunden. Abbildung 5.9 (links) zeigt die Schwankungen, bereits umgerechnet in den effektiven Fehler, der bei einer Distanzmessung auftritt. Man erkennt einen absoluten Fehler von fast 10m nach 4 Stunden. Man erkennt außerdem den recht konstanten Anstieg zu Beginn. Dieser konstante Anstieg konnte durch ein zweites Experiment reproduziert werden.

Es wurde weiter versucht, den Fehler der ersten 2000 Messpunkte durch einen linearen Anstieg zu modellieren und diesen Anteil heraus zu rechnen. Dazu wurde eine vermittelnde Gerade durch die Werte der ersten 2000 Messungen gelegt. Abbildung 5.9 (rechts) zeigt die residualen Fehler. Der Gesamtumfang der verbleibenden Schwankungen lag bei etwa 55cm. Der residuale Fehler beträgt also noch etwa 10% der ursprünglichen Abweichung.

Als Ergebnis kann festgestellt werden, dass die residualen Fehler auch nach der Modellierung eines konstanten Fehleranstiegs zu hoch für eine sinnvolle Verwendung der Messungen ist. Im Experiment befand sich der Aibo in völliger Ruhe. Während eines Spiels ist durch Spannungsschwankungen der internen Spannungsversorgung von noch größeren Schwankungen auszugehen. Eine einzelne initiale Synchronisation ist daher auf dieser Plattform unzureichend. Eine fortlaufende Synchronisation, z.B. gemäß Kapitel 3.7.2.2, muss verwendet werden.

5.9 Lautstärkeverhältnisse

Beim gleichzeitigen Aussenden von Schemata spielen Lautstärkeverhältnisse eine wichtige Rolle. Hört ein Roboter das eigene Codeschema deutlich lauter als das eines anderen Roboters, so schlägt die Kreuzkorrelation schon bei kleinen Distanzen fehl (siehe 3.6.6.2).

In einem Experiment wurde daher der Zusammenhang zwischen Distanz und Lautstärke untersucht.

Dabei sendete ein still stehender Roboter fortlaufend sein Schema aus, während ein anderer Roboter über längere Strecken bewegt wird. Abbildung 5.10 zeigt das resultierende Diagramm, dass die Stärke des Korrelationspeaks über die Entfernung aufträgt. Die absoluten Werte der Peaks sind hier nicht von Belang, sondern ausschließlich der Grad der Abschwächung in Abhängigkeit der Entfernung. Da die Kreuzkorrelationsfunktion invariant gegenüber einer skalaren Multiplikation des Eingangssignals ist, spiegelt der Verlauf der aufgetragenen Korrelationspeaks den qualitativen Verlauf der eintreffenden Lautstärke wider.

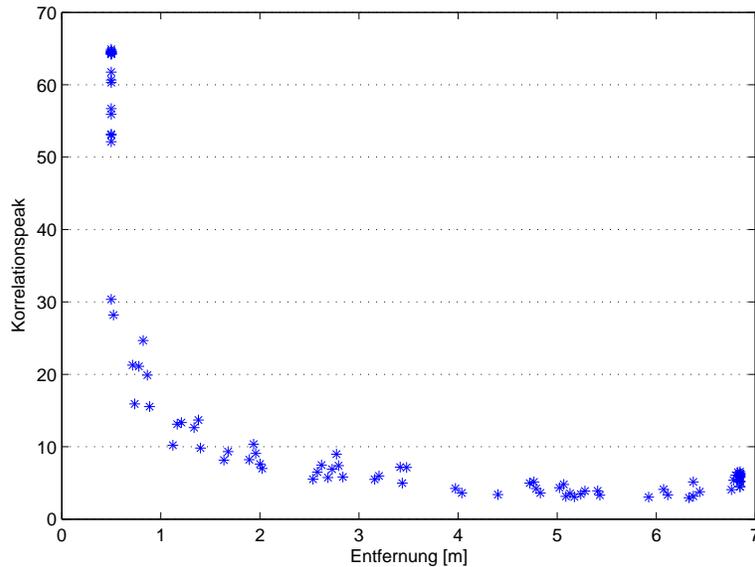


Abbildung 5.10: *Experiment zur Ermittlung des Zusammenhangs zwischen Entfernung und Abschwächung der Lautstärke.*

Der Verlauf der Kurve ist vor allem deshalb nicht völlig regelmäßig, weil die unterschiedlichen Positionen im Raum, an denen sich der Roboter befand, unterschiedliche Interferenzmuster aufweisen. Die Werte sind zusätzlich durch das akustische Rauschen im Raum leicht verfälscht.

Anzumerken ist außerdem, dass der Verlauf der Kurve von der Umgebung abhängig ist. In einer völlig freien Umgebung wird die Lautstärke mit der Distanz stärker abnehmen. In kleineren Räumen nimmt die Lautstärke etwas weniger mit der Distanz ab, da der ausgesendete Schall an Wänden sowie am Boden reflektiert wird. Das Experiment fand in einem freien Raumbereich der Breite 6m und der Tiefe 10m statt.

5.10 Berechnungsaufwand

Für den Einsatz in Echtzeit ist der Berechnungsaufwand von hoher Wichtigkeit. Bei der Implementierung wurde daher darauf geachtet, den Berechnungsaufwand gering zu halten.

Einen großen Vorteil bringt die Verwendung von Ganzzahlen anstatt Fließkommazahlen für die interne Speicherung von Sounddaten und Korrelationswerten. Der Hauptaufwand bei der Lokalisation mit Kreuzkorrelation entsteht durch die Multiplikationen und Additionen bei der Auswertung der Kreuzkorrelationsfunktion. Eine Ganzzahl-Addition oder -Multiplikation geht wesentlich schneller von statten als die Addition oder Multiplikation von Fließkommazahlen.

Die Korrelationswerte werden bei der Verwendung von Ganzzahlen sehr hoch. Ein Überlauf musste durch Skalierung der Werte um den Faktor $\frac{1}{2^9}$ (d.h. durch ein Shift nach rechts um 9 Bit) verhindert werden. Der entstehende Verlust an Information durch Wegfall der letzten Stellen ist dabei hinnehmbar, da die maximalen Korrelationswerte ohnehin *deutliche* Maxima sein müssen, um klar als solche identifiziert zu werden. Sind im Gegenteil zwei Korrelationswerte der Größenordnung 2^{38} bis auf die letzten 9 Bits identisch, so kann das Maximum der beiden Werte *ohnehin* nicht für eine zuverlässige Messung herangezogen werden, da die Differenz zwischen den beiden Werten viel geringer ist als der mittlere Fehler bei der Auswertung der Kreuzkorrelation.

Gemäß Kapitel 3.6.10 genügt eine Auswertung der Korrelationsfunktion im Intervall $t_1 = \pm v_r^{max} \cdot \chi \cdot \frac{1}{c}$. t_1 ist dabei der Zeitpunkt, zu dem das nächste Schema bei $v_r = 0$ empfangen würde. $v_r^{max} = 1 \frac{m}{s}$ wird hier nach oben abgeschätzt. Rechnet man direkt in Abtastpunkten, ergibt sich ein maximaler Abstand zwischen den Korrelationsmaxima von $\pm 1 \cdot 16000 \cdot \frac{1}{c} \approx \pm 47$ Abtastpunkten pro Sekunde. Der zeitliche Abstand zwischen zwei ausgesandten Schemata beträgt 5 Frames, also 0.16 Sekunden. Somit ergibt sich ein maximaler zeitlicher Abstand zwischen zwei Empfangszeitpunkten von $\pm [7.46] = \pm 8$ Abtastpunkten.

Die Gesamtzahl von Additionen und Multiplikationen soll im folgenden abgeschätzt werden. Die Berechnung erfordert in jedem Takt

- 512 Additionen und 512 Multiplikationen pro Korrelationswert,
- die Auswertung von maximal 17 Korrelationswerten (nämlich im Intervall ± 8 Abtastpunkte relativ zur zuletzt gefundenen Position), sowie
- die Auswertung der Korrelationswerte zu k Mitspielern.
- Die Auswertung *beider* Mikrofon-Kanäle liefert einen zusätzlichen Faktor von 2.

Die Zahl 512 ist dadurch begründet, dass ein 32ms-Takt bei 16000Hz Abtastrate genau 512 Abtastpunkten entspricht. Insgesamt ergeben sich also $k \cdot 512 \cdot 17 = k \cdot 8704$ Additionen und Multiplikationen pro 32ms-Takt (bzw. $k \cdot 17408$ bei Evaluierung beider Kanäle).

5.11 Abwechselndes Aussenden von Codes

Bisher wurde vor allem der Fall untersucht, dass alle Roboter gleichzeitig ihre Schemata aussenden. Diese Variante ist einerseits wünschenswert, da es insgesamt sehr viele Messungen pro Sekunde geben kann. Der erhebliche Nachteil besteht darin, dass die maximal messbare Distanz reduziert wird, und zwar bedingt durch das maximale Lautstärkeverhältnis q_v^{max} (siehe Kapitel 3.6.6.2). Das eigene ausgesandte Codeschema wirkt auf die Korrelation wie ein lautes pseudozufälliges Störgeräusch. Die maximale Distanz wird *zusätzlich* noch durch die Störgeräusche der Umgebung und der robotereigenen Motoren reduziert.

Die Alternative besteht darin, dass die Roboter *abwechselnd* ihre Schemata senden. Dies kann z.B. in einer einfachen Round-Robin-Manier geschehen. Ein einzelnes Aussenden sollte dabei so kurz wie möglich sein (z.B. *ein* Schema), damit möglichst viele Roboter pro Zeiteinheit ihr Schema senden können.

Wichtig ist, dass die Roboter sich auf einen gemeinsamen Nullzeitpunkt einigen müssen, der für alle als Referenz dient. Ist allen dieser Nullzeitpunkt bekannt, so können alle Roboter relativ zu diesem anhand des bekannten Turnus ausrechnen, welcher Roboter zu welchem Zeitpunkt sendet und insbesondere auch, wann ein Roboter selbst sein Schema aussenden muss.

Für den Nullzeitpunkt wurde eine einfache Definition gewählt: Der globale Nullzeitpunkt ist der Zeitpunkt, an dem Roboter Nummer 1 das erste Mal den `SoundInProcessor` aufruft (siehe 5.12). Diesen Zeitpunkt teilt Roboter Nummer 1 daraufhin den anderen Robotern über WLAN mit. Jeder Roboter rechnet den Nullzeitpunkt anhand von NTP-Zeitoffsets [24] in lokale Zeit um.

Der Ansatz des abwechselnden Aussendens birgt das Problem, dass die Synchronisation weniger genau ist. Eine Distanz zwischen zwei Robotern A und B wird nicht mehr zeitgleich, sondern mit durchaus erheblicher zeitlicher Differenz gemessen. Um diesen Effekt gering zu halten wurde für den Aussende-Turnus ein abgewandeltes Round-Robin gewählt, bei dem jedes Paar von Robotern mindestens einmal direkt hintereinander sendet. Damit ist einmal pro Turnus die Möglichkeit gegeben mit geringem Fehler eine Synchronisation durchzuführen. Bis zu einer Anzahl von drei Robotern kann der gewöhnliche Round-Robin-Turnus verwendet werden. Bei 4 Robotern ergibt sich z.B. die Folge 1-2-3-4-1-3-2-4, bei 5 Spielern die Folge 1-2-3-4-5-1-3-5-2-4. Diese Folge wird dann fortlaufend wiederholt.

Es gibt gewisse Ungenauigkeiten bei der Bestimmung des Nullzeitpunktes in lokaler Zeit über NTP. Außerdem gibt es beim Aibo einen geringen, schwierig vorhersehbaren Zeitversatz zwischen Aktivieren oder Deaktivieren der Sound-Ausgabe und der *tatsächlichen* Unterbrechung der Klangausgabe. Darüber hinaus ist zu beachten, dass es bis zu 20msec dauern kann, bis ein ausgesandtes Schema bei einem weit entfernten Roboter eintrifft.

Um sicher auszuschließen, dass zwei Roboter in den Übergangsbereichen gleichzeitig senden, bzw. gleichzeitig zwei Schemata bei einem dritten Roboter eintreffen, musste eine gewisse Toleranz gewählt werden. Dabei erhält jedes Zeitfenster eine Gesamtlänge von 288msec, wobei davon für 224msec das Schema ausgesendet wird. Ein einzelnes Schema hat dabei eine Länge von 160msec. Ein gesamter Turnus dauert dann im Falle von 5 Robotern 2880msec, also knapp 2,9s.

Hinweis: In jedem Zeitfenster werden damit also 1,4 Schemata ausgesendet! Tatsächlich muss der empfangende Roboter *innerhalb* dieses Sendezeitraumes einen Zeitraum von der Länge genau *eines* Schemas sicher finden können. Die Korrelation erfolgt dann nur über dieses eine (um Vielfache von 32msec verschobene) Schema. Es wurden stets ganzzahlige Vielfache von 32msec gewählt, was der Länge eines zu verarbeitenden Frames im Aibo entspricht.

Abbildung 5.11 zeigt den zeitlichen Verlauf des abwechselnden Aussendens. Dargestellt wird ein Sende-Turnus für 5 Roboter. Außerdem ist der zeitliche Verlauf *innerhalb* eines Zeitfensters skizziert.

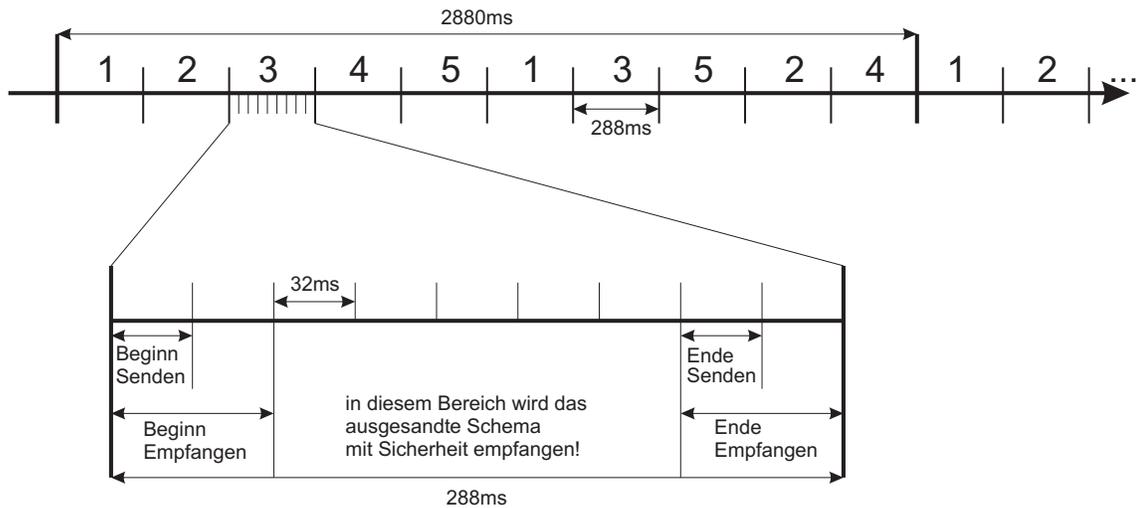


Abbildung 5.11: Zeitlicher Verlauf des abwechselnden Aussendens. Die verwendete Skala ist die globale Zeit. Ein Turnus dauert 2880msec. Ein einzelnes Zeitfenster dauert 288msec.

Oben: Übersicht über ein Turnus. Die Zahl repräsentiert den Roboter des Teams, der in diesem Zeitfenster sendet.

Unten: Ein Zeitfenster ist in 9 Intervalle à 32msec unterteilt. Es wird für stets 7 Intervalle lang gesendet. Da die Aibos in 32msec-Intervallen arbeiten, kann ein lokales Zeitintervall um bis zu 32msec vom globalen Intervall verschoben sein. Das Aussenden geschieht daher frühestens zur globalen Zeit 0ms und spätestens zur Zeit 32msec des aktuellen Zeitfensters. Da der Schall zunächst maximal für 32msec durch den Raum propagiert, kann ein empfangender Roboter frühestens ab 0msec und spätestens ab 64msec des aktuellen Zeitfensters das Signal empfangen. Der Bereich zwischen dem spätesten möglichen Empfangsbeginn und dem frühesten möglichen Empfangsende wird dann für die eigentliche Korrelation verwendet. Dieser Bereich ist 5 Intervalle, d.h. 160msec lang, hat also genau die Länge eines Code-Schemas. So ist sichergestellt, dass die für die Korrelation verwendeten Daten genau ein einzelnes (verschobenes) Schema enthalten.

Das Aktivieren und Deaktivieren der Sound-Ausgabe geschieht über Mute-Befehle. Das heißt, der Roboter sendet das Schema *im Prinzip* weiterhin *fortlaufend* an den Lautsprecher, genau wie im Falle des gleichzeitigen Sendens. Während den Zeiten, zu denen der Roboter nicht senden soll, wird die Sound-Ausgabe einfach durch ein Mute unterdrückt. Der Vorteil dabei ist, dass das Senden völlig linear über die Zeit erfolgt. Die ausgesandten Schemata bleiben also stets ‘im Rhythmus’. Damit kann auch ein empfangender Roboter berechnen, wie die aktuelle Verschiebung eines Schemas eines sendenden Roboters ist. Der empfangende Roboter kennt also, bis auf die eventuelle Distanzänderung, die genaue Position an der das Schema liegen muss, auch wenn die verstrichene Zeit seit dem letzten Empfang *kein Vielfaches* der Schemalänge ist, und auch wenn die Soundausgabe des sendenden Roboters zwischenzeitlich unterbrochen war. Für die Auswertung der KKF ist es nur erforderlich, dass über Sounddaten einer Schemalänge (oder ein ganzzahliges Vielfaches) korreliert wird. Das Schema kann dabei durchaus zyklisch verschoben sein, da der empfangende Roboter anhand der verstrichenen Zeit direkt die ungefähre Verschiebung berechnen kann.

Eine Synchronisation kann dementsprechend nur alle 2,88s erfolgen. Dies ist völlig ausreichend, da der Fehler durch den Uhrendrift in diesem Zeitraum vernachlässigbar gering ist. Entscheidend ist die Zeitdauer von 288msec, die zwischen den zwei Messungen derselben Distanz (A nach B; B nach A) verstreicht. Die relative Distanzänderung der beiden beteiligten Roboter nach diesem Zeitraumes geht halbiert als Fehler in die Synchronisation ein. Dieser Fehler beträgt dann im *ungünstigsten* Falle 12cm. Bei zufälliger Bewegung liegt dieser Fehler im Mittel bei 6cm, im Falle eines RoboCup-Fußballspiels tendenziell geringer, da zu einem Zeitpunkt selten mehrere Roboter mit voller Geschwindigkeit in entgegengesetzte Richtungen laufen. Dieser Synchronisierungsfehler wirkt sich nur bis zur nächsten Synchronisation aus, also z.B. nur auf eine einzelne Messung, falls bei der nächsten Messung bereits erneut synchronisiert werden kann.

Der Rechenaufwand bleibt durch den Ansatz annähernd gleich: Die Korrelation muss zwar jeweils nur für ein Schema ($k = 1$), dafür aber mit entsprechend größeren Intervallen für das Positions-Tracking erfolgen.

Die Kreuzkorrelationseigenschaften der verwendeten Codes sind bei einem abwechselnden Aussenden unwichtig. Bei der Wahl der Codefolge muss allein auf die Autokorrelationseigenschaft Wert gelegt werden. Eine Codefolge mit sehr guten Autokorrelationseigenschaften (besser als bei den bisher verwendeten Kasami-Codefolgen) ist ein 127-Chip langer Gold-Code. Für diesen gilt $A^{max} = 1$ (gegenüber $A^{max} = 9$ bei Kasami-Codefolgen der Länge 63). Dieser ist auch im Anhang D zu finden.

5.12 Einbindung in den GermanTeam-Code

Die Implementierung geschah im Rahmen des GermanTeam-Codes von 2007/-2008, der für Roboter-Fußball im Rahmen der Aibo-Liga im RoboCup entwickelt wurde. Für Beschreibungen zur Struktur des GermanTeam-Codes siehe [26] und [27].

Für die Schall-Lokalisation wurden im Projekt einige Module sowie Representationen neu angelegt oder bearbeitet. Im Folgenden werden die wichtigsten Module und Representationen, sowie der funktionelle Ablauf erläutert. Der Informationsfluss wird außerdem in einem Diagramm (Abbildung 5.12) übersichtlich dargestellt.

- Der `SoundDataSender` übernimmt das Senden von Sound-Daten zum `OPEN-R-Bus`, über den die Sound-Daten letztlich zum Lautsprecher des Aibo gelangen. Die Sound-Daten sind in der Repräsentation `SoundDataOut` abgelegt, welche vom `SoundOutProcessor` zur Verfügung gestellt wird. Die Sound-Daten müssen in maximal 1 KByte großen Paketen zum `OPEN-R-Bus` gesendet werden. Der `SoundDataSender` legt hierzu die Daten in 1 KByte großen Paketen einmalig in speziellen Speicherregionen ab, so dass bei der (wiederholten) Wiedergabe längerer Sounds nicht immer wieder dieselben Daten transferiert werden müssen. Der `SoundDataSender` teilt über den `OPEN-R-Bus` dann lediglich die Speicheradresse mit, an der die als nächstes abzuspielenden Sound-Daten abgelegt sind. Der ursprüngliche `SoundDataSender` des GermanTeam-Codes von 2004 wurde dadurch erheblich performanter im Falle sich ständig wiederholender Sounds.
- Der `SoundOutProcessor` steuert die Ausgabe der Sounddaten. Dieser liest ein Sound-Schema ein, welches als RAW-Datei im `config`-Verzeichnis abgelegt ist. Er stellt die Repräsentation `SoundDataOut` zur Verfügung, in welcher sich neben den eigentlichen Daten auch ein Schalter befindet, der die Ausgabe aktiviert bzw. deaktiviert. Der `SoundOutProcessor` hat neben dem einmaligen Einlesen der externen Datei lediglich die Aufgabe, die Ausgabe von Sound zu aktivieren bzw. deaktivieren. Senden beispielsweise mehrere Roboter abwechselnd nach einer bekannten Reihenfolge, so muss die Soundausgabe ständig aktiviert und deaktiviert werden. Dazu wird ein spezieller Mute-Befehl verwendet, der als `SystemCall` implementiert wurde. Zusätzlich kann die Soundausgabe auch vom Verhalten über die `SoundRequest`-Repräsentation angefordert werden.

Hinweis: Der `SoundInProcessor` übernimmt die zentrale Koordination der eingehenden Sound-Daten. Im Falle eines abwechselndes Aussendens

koordiniert er außerdem *davon abhängig* den *ausgehenden* Sound (siehe Kapitel 5.11). Für die Koordination des ausgehenden Sounds stellt er die `SoundInitInfo`-Repräsentation zur Verfügung. Dieser ‘Umweg’ des Informationsflusses über den `SoundOutProcessor` ist deshalb notwendig, da dieser im ‘Motion’-Prozess des Projektes ausgeführt werden muss (im 8ms-Takt). Der `SoundInProcessor` sowie die meisten anderen Module werden separat im ‘Cognition’-Prozess verarbeitet, der im GermanTeam-Code alle 32ms durchlaufen wird (einmal für jedes Kamerabild). Die Notwendigkeit entstand also durch die speziellen Infrastruktur des Projektes. Die gemeinsame Verwendung durch beide Prozesse ist mit der nur wenigen Bytes großen `SoundInitInfo` *deutlich* performanter als mit der bis zu 10 KBytes großen `SoundDataOut`-Repräsentation.

- Der `SoundInProcessor` wertet die eingehenden Sound-Daten aus. Diese sind im RAW-Format in der Repräsentation `SoundDataIn` gespeichert, welche vom `CognitionDataProvider` zur Verfügung gestellt wird. Der `SoundInProcessor` übernimmt die Auswertung der Kreuzkorrelation. Der `SoundInProcessor` muss dabei entscheiden, ob die Distanz zu einem anderen Roboter verfolgt werden kann (‘Distanztracking’), oder ob diese initialisiert werden muss. Das Ergebnis der Auswertung der Kreuzkorrelationsfunktion sind die gemessenen Distanzen zu anderen Robotern (gemessen in 16kHz-Abtastpunkten). Für Richtungsmessungen können die Distanzen in beiden Kanälen des Stereo-Eingangssignals berechnet werden. Diese Positionen werden im `SoundInPercept` gespeichert.

Der `SoundInProcessor` stellt, wie oben beschrieben, auch die `SoundInitInfo` zur Verfügung. Der `SoundOutProcessor` erhält mit Hilfe dieser Repräsentation wichtige Information wie den modellierten Nullzeitpunkt in lokaler Systemzeit (siehe 5.11) sowie die Spielernummer.

- Der `SoundRobotPerceptProvider` hat die Hauptaufgabe, anhand der Sample-Positionen aus dem `SoundInPercept` die eigentlichen Distanzen und Richtungen auszurechnen. Außerdem übernimmt er die fortlaufende Synchronisation, die den Hauptaufwand dieses Moduls ausmacht. Das Ergebnis der Berechnungen ist das `SoundRobotPercept`, welches die Distanzen und Richtungen zu anderen Robotern beinhaltet.
- Der `TeamDataProvider` stellt die innerhalb des Teams ausgetauschte Information zur Verfügung. In den `SoundSyncStatusMessages` teilen die einzelnen Roboter mit, ob sie im Moment Sound aussenden. Außerdem werden hierüber alle Distanzmessungen der Roboter untereinander sowie der

für alle Roboter zu verwendende Nullzeitpunkt ausgetauscht. Mit Hilfe der `MCSLMessages` werden Informationen des `MonteCarloSelfLocator` zwischen verschiedenen Robotern ausgetauscht.

- Der `MonteCarloSelfLocator` ist das Modul zur Selbstlokalisierung im `GermanTeam`-Code. Es wurden Teile für die kooperative Nutzung der Informationen aus dem `SoundRobotPercept` ergänzt. Über `MCSLMessages` werden spezielle Informationen wie z.B. Positionskonfidenzen zwischen den Robotern ausgetauscht. Im Rahmen dieser Arbeit wurde lediglich der Ansatz der *paarweisen* kooperativen Auswertung der relativen Positionsinformationen implementiert (siehe Abschnitt 4.2).
- Der `OraclePerceptProvider` dient ausschließlich zu Debug-Zwecken. Anhand der `GroundTruthRobotPoses`-Repräsentation, die die *wahren* Positionen aller Roboter beinhaltet, wird ein `SoundRobotPercept` berechnet, das relative Positionsmessungen *simuliert*. Das Modul kann nur *alternativ* zum `SoundRobotPerceptProvider` verwendet werden, wenn die wahren Positionen bekannt sind (z.B. mit Hilfe einer Deckenkamera oder in einer Simulation). Das Modul kommt zum Einsatz, wenn die kooperative Selbstlokalisierung unabhängig von der Güte der Messungen getestet werden soll.

Zusätzlich wurden dem Projekt einige Hilfsklassen hinzugefügt. Die wichtigste davon ist die Klasse `SoundTools`, die unter anderem die statischen Methoden für die Korrelationsfunktion, die Maximums-Suche in Korrelationswerten, die Generierung von Code-Schemata sowie einige Konstanten beinhaltet.

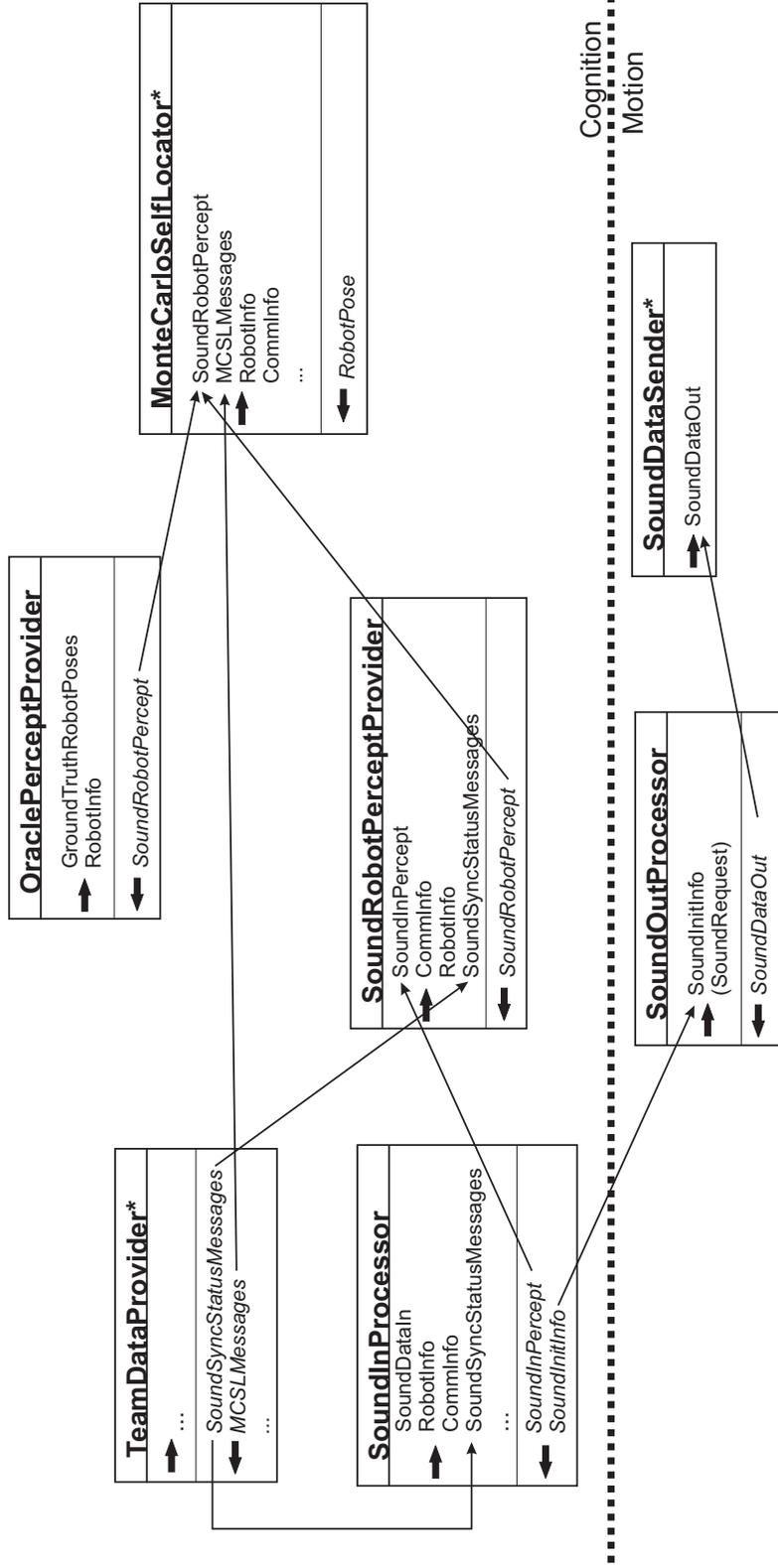


Abbildung 5.12: Informationsfluss im GermanTeam-Code zwischen den Modulen, die an der Schall-Lokalisation beteiligt sind. Jedes Modul besitzt neben dem Namen eine Liste der benötigten Eingaben und der zur Verfügung gestellten Repräsentationen (Ausgaben). *: Das Modul wurde im Rahmen dieser Arbeit lediglich erweitert oder bearbeitet.

5.13 Experimente zur Distanzmessung

5.13.1 Distanzmessung ohne Synchronisation

In einem Experiment wurde die Distanz zu einem einzelnen sendenden Roboter A gemessen. Ein zweiter Roboter B wurde dabei entlang eines Maßbandes bewegt. Die Distanz zwischen den Robotern betrug zu Beginn etwa 50cm. Im Anschluss wurde Roboter B in 50cm-Schritten bis auf 4m Entfernung zu A bewegt. Danach wurde B kontinuierlich bis auf eine Distanz von ca. 50cm wieder A angenähert. Beide Roboter verharrten noch einige Sekunden an ihren Endpositionen. Das Experiment dauerte insgesamt 90 Sekunden.

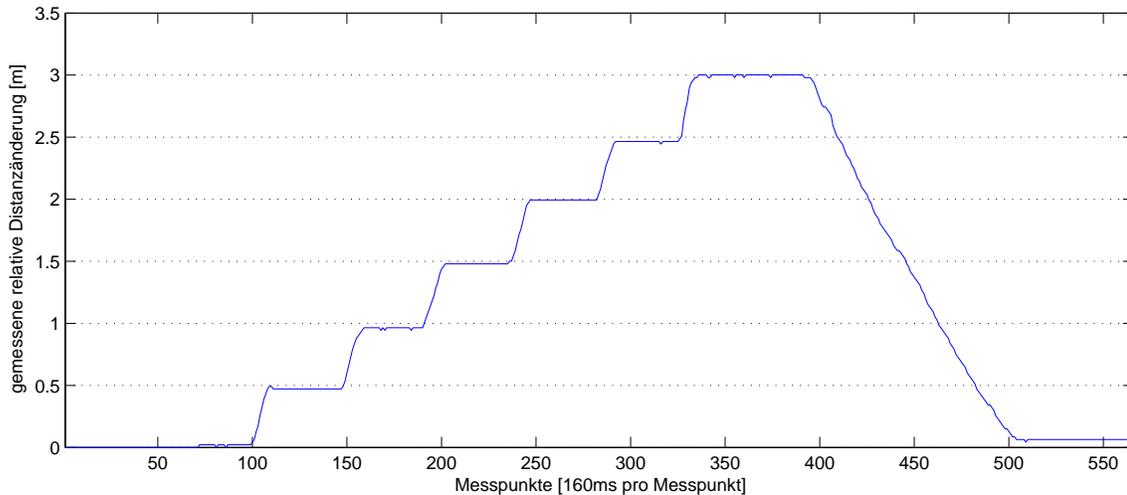


Abbildung 5.13: *Experiment zur Distanzmessung ohne Synchronisation. Siehe Text.*

Abbildung 5.13 zeigt die gemessenen Distanzen im Verlauf des Experiments. Da die Roboter nicht synchronisiert wurden, wird hier lediglich die *relative* Veränderung gezeigt. Daher wird mit einem Startwert von 0m begonnen. Die geringen Abweichungen an den verschiedenen Positionen müssen neben dem Fehler durch Uhrendrift auch auf den ungenau am Maßband aufgestellten Roboter B zurückgeführt werden.

Der Sollwert für die relative Distanzänderung beträgt am Ende des Experiments 0m. Gemessen wurde hier eine relative Distanz von 7cm. Dieser Fehler kann vor allem auf den Uhrendrift zurückgeführt werden, da Roboter B mit hoher Genauigkeit an der Endposition aufgestellt wurde. Das Experiment des nächsten Abschnittes wird diesen Effekt noch deutlicher zeigen.

5.13.2 Distanzmessung mit Synchronisation

5.13.2.1 Bewegungen am Maßband

In einem zweiten Experiment wurden wiederum zwei Roboter A und B an einem Maßband aufgestellt. Roboter A steht still an Position 0m, Roboter B steht zu Beginn bei 0.5m. Diesmal sendeten beide Roboter gleichzeitig, so dass eine fortlaufende Synchronisation möglich ist. Die Roboter wurden wieder in 0.5m-Schritten von 0.5m bis 3m entfernt und anschließend wieder auf 0.5m angenähert. Zusätzlich wurde im Anschluss Roboter B zwischen den Marken bei etwa 0.25m und 1m zügig hin- und herbewegt.

Abbildung 5.14 zeigt den Verlauf dieses Experimentes. In blau eingezeichnet ist die gemessene Distanz bei *fortlaufender* Synchronisierung. Die resultierenden Distanzen, die beide Roboter dabei errechnen, sind erwartungsgemäß identisch (siehe 3.7.2.2).

Die beiden anderen Linien im Diagramm zeigen zum Vergleich die gemessenen Distanzen bei Verwendung einer *einzelnen* Synchronisation zu Beginn. Man erkennt gegen Ende deutlich den durch den Uhrendrift hervorgerufenen Fehler von etwa 7cm. Die von den beiden Robotern gemessenen Distanzen unterscheiden sich dadurch um 14cm.

Man beachte, dass sich die Roboter zwischen den Messpunkten 320 und 560 in völliger Ruhe befanden. Auch hier ist das allmähliche Anwachsen des Fehlers gut zu erkennen.

5.13.2.2 Vergleich mit Deckenkamera

In einem weiteren Experiment wird ein Roboter frei in der Umgebung des stehenden Roboters bewegt. Die realen Distanzen werden diesmal mit Hilfe einer Deckenkamera aufgezeichnet.

Abbildung 5.15 zeigt den Verlauf dieses Experiments. Die Distanzen der Deckenkamera wurden teilweise nicht aktualisiert, z.B. weil die Sicht zum optischen Marker auf dem Roboter blockiert war. Im Diagramm erkennt man dies an gleich bleibenden Werten (Stufen im Kurvenverlauf). In diesen Bereichen liegt *kein* grober Messfehler der akustischen Distanzmessung vor.

Im Bereich zwischen Messzeitpunkt 13000 und 14000 erkennt man, dass die Distanz akustisch um über 10cm zu groß gemessen wurde. Dies liegt vor allem an Interferenzeffekten, da hier der empfangende Roboter weg vom sendenden Roboter

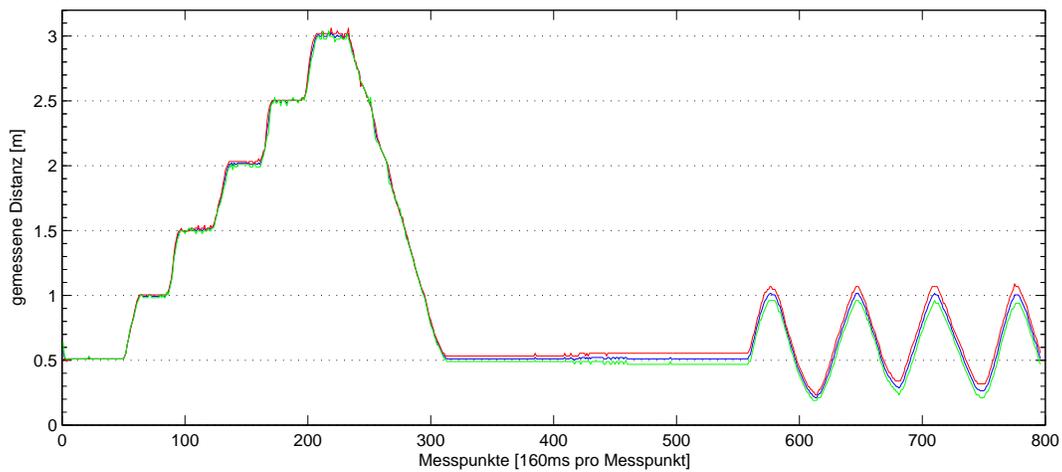


Abbildung 5.14: *Experiment zur Distanzmessung mit einmaliger bzw. fortlaufender Synchronisation. Rot und grün: Gemessene Distanzen unter Verwendung einer einmaligen initialen Synchronisation; blau: unter Verwendung einer fortlaufenden Synchronisation.*

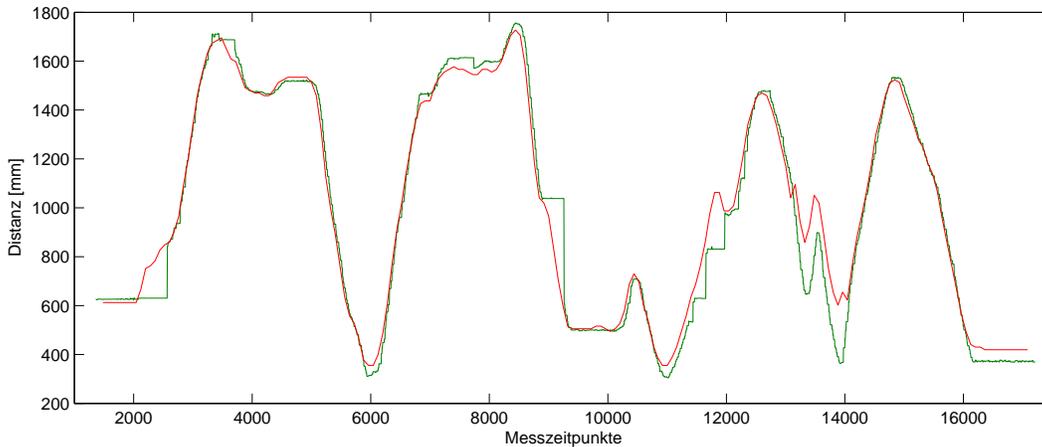


Abbildung 5.15: *Vergleich zwischen Deckenkamera und akustischer Distanzmessung. Die grüne Kurve kennzeichnet den Verlauf der Distanz, wie er von der Deckenkamera registriert wurde. Die rote Kurve kennzeichnet den Verlauf der akustisch gemessenen Distanzen. Hinweis: An Zeitpunkten, an denen die Deckenkamera die Roboter nicht erkennen konnte, bleibt die Distanz im Diagramm gleich (Stufen im Kurvenverlauf). Es ist hier nur der Verlauf unter Verwendung einer fortlaufenden Synchronisation eingezeichnet.*

blickte. Es bestand keine freie Sicht zwischen Sender und Empfängermikrophon. Die Schallwelle wurde daher stark am Chassis gebrochen. In einer Implementierung könnte dies dadurch verbessert werden, indem versucht wird, immer nur das dem Sender zugewandte Mikrophon zu verwenden. Dadurch, dass beide Roboter in dieselbe Richtung blicken, entsteht zusätzlich ein Fehler durch den projizierten Abstand zwischen Sender-Lautsprecher und Empfänger-Mikrophon. Dieser Effekt ist dann relevant, wenn der Kopf des Aibo gesenkt ist, siehe Abbildung 5.3.

In den Bereichen mit geringer Distanz, z.B. bei den Messzeitpunkten 6000, 11000 und ab 16000, kommt ein weiterer Effekt zum Vorschein. Die Distanz wurde hier ebenfalls zu groß gemessen. Der Grund hierfür ist, dass die *vertikale* Distanz zwischen dem Lautsprecher des Sender-Roboters und den Mikrofonen des Empfangs-Roboters ins Gewicht fällt. Dieser Fehler kann mit Hilfe des Satzes von Pythagoras herausgerechnet werden, wenn die ungefähre vertikale Distanz bekannt ist. Bei größeren Distanzen ist dieser Effekt nicht relevant.

5.13.3 Ergebnisse einer Spielsituation im Roboterfußball

In einem letzten Experiment wird der Einsatz in einem Roboter-Fußballspiel getestet. Entscheidend hierbei sind die lauten Störgeräusche, die nun durch die Motoren umher laufenden Roboter auftreten. Im Experiment werden drei Roboter verwendet, die sich frei bewegen können. Die wahren Distanzen zwischen den Robotern wurden wiederum anhand einer Deckenkamera aufgezeichnet.

Damit die maximal messbare Distanz mit diesen stark verrauschten Signalen recht gering ist, wird hier ein *abwechselndes* Aussenden der Code-Schemas gemäß Kapitel 5.11 verwendet. Es wurde dabei die periodisch wiederholte Sendeabfolge '1-2-3' verwendet.

Abbildung 5.16 a) zeigt die gemessenen Distanzen zwischen den Robotern 1 und 2, b) die zwischen 1 und 3 und c) die zwischen 2 und 3. Die Werte der Deckenkamera treffen aufgrund der Latenzen des Funknetzwerks etwas später ein und wurden deshalb zur besseren Vergleichbarkeit zeitlich geringfügig verschoben eingezeichnet. Die Fehler der akustischen Messungen betragen gegenüber den Messwerten der Deckenkamera maximal 14cm. Fehlerursachen sind vor allem die Synchronisation mit zeitlich versetzten Messungen (Abschnitt 5.11), die durch Brechung und Reflexion auftretenden Fehler (Abschnitt 5.6) sowie Fehler durch unterschiedlichen Kopfeigungen (Abschnitt 5.2.1). Im Nahbereich wird wiederum die Distanz aufgrund des *vertikalen* Abstands zwischen Lautsprecher und Mikrofonen zu groß gemessen.

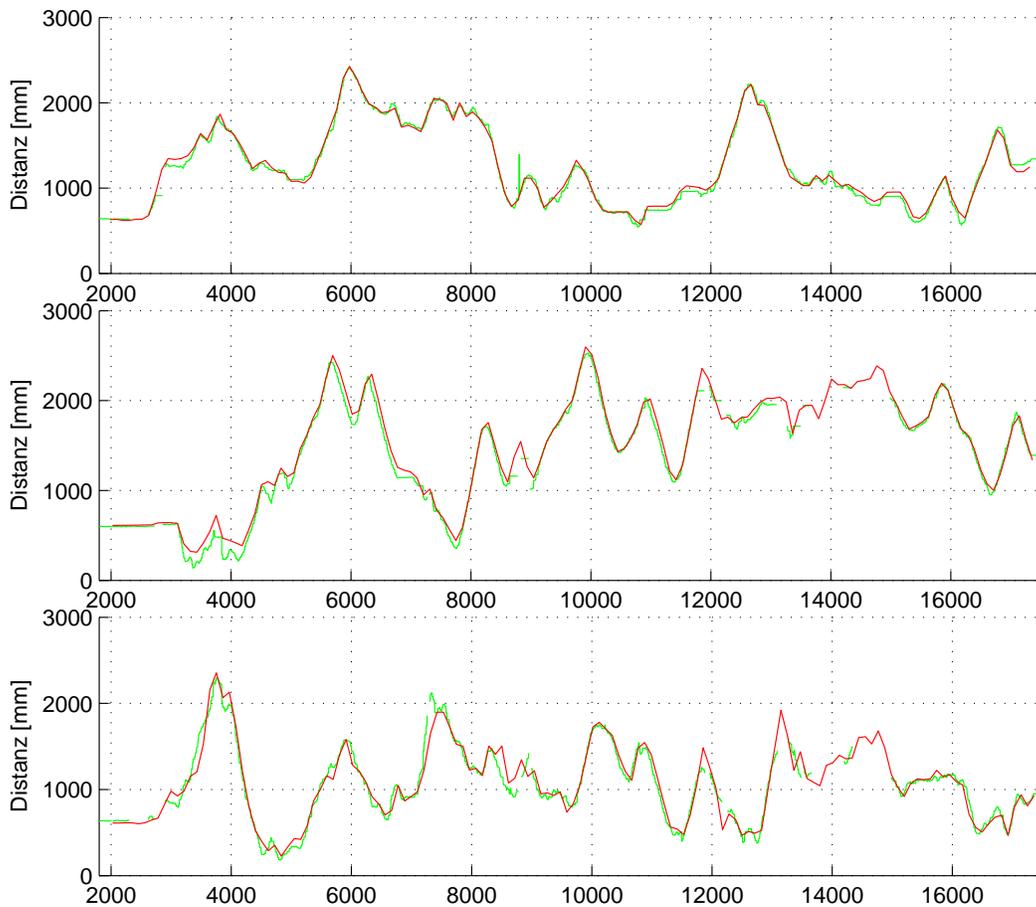


Abbildung 5.16: Ergebnisse einer realen Roboterfußball-Spielsituation. Drei Roboter bewegten sich hierbei frei auf dem Spielfeld. Die akustisch gemessenen Distanzen sind rot eingezeichnet.

Die wahren Distanzen (grün) wurden mit Hilfe einer Deckenkamera aufgezeichnet. Die grüne Kurve ist zu Zeitpunkten unterbrochen, an denen die Deckenkamera nicht die beide beteiligten Roboter sehen konnte.

Abbildung a) (oben): Distanz zwischen den Robotern 1 und 2;

Abbildung b) (Mitte): Distanzen zwischen den Robotern 1 und 3;

Abbildung c) (unten): Distanzen zwischen den Robotern 2 und 3.

5.14 Experiment zur Richtungsmessung

Die Genauigkeit der Richtungsmessung soll in einem Experiment ermittelt werden. Die Genauigkeit bei der Richtungsmessung ist durch den Effekt der digitalen Abtastung begrenzt. ds kann nur mit einer Genauigkeit von ca 2.1cm bestimmt werden (das ist die Differenz zwischen zwei benachbarten Abtastpunkten). Zur Erhöhung der Genauigkeit wurden die Korrelationswerte zwischen den maximalen benachbarten Abtastpunkten interpoliert (siehe Abschnitt 3.7.1).

Ein Roboter A steht still an einer festen Position. Roboter B sendet sein Schema und wird in Schritten von 22.5° auf einer Kreisbahn mit Radius 1m um Roboter A bewegt. An den einzelnen Punkten wurde jeweils der Wegunterschied ds gemessen.

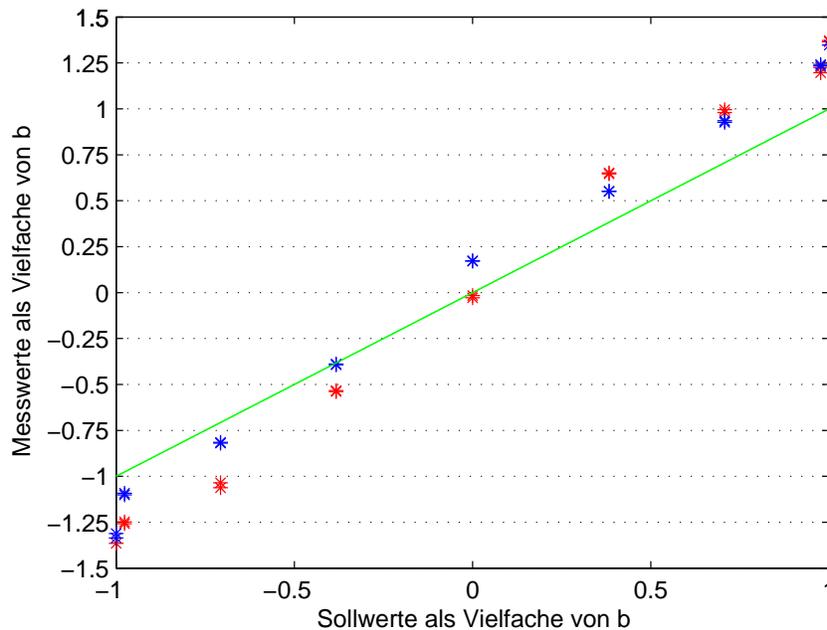


Abbildung 5.17: *Ergebnisse der Richtungsmessung. Die Messwerte werden über die Sollwerte aufgetragen. Man erkennt die durchgehend betragsmäßig zu großen Wegdifferenzen.*

Abbildung 5.17 zeigt die gemessenen Wegunterschiede aufgetragen über die Sollwerte. Die blauen Punkte stellen Messungen dar, bei denen sich Roboter B vor Roboter A befand, also im Bereich $-90^\circ \rightarrow 0^\circ \rightarrow +90^\circ$. Die roten Punkte symbolisieren entsprechend Messungen im Bereich $-90^\circ \rightarrow \pm 180^\circ \rightarrow +90^\circ$. Die Werte waren an den jeweiligen Positionen gut reproduzierbar.

Unterschiede ergeben sich durch die Charakteristik der Testumgebung. Je nach Richtung werden die Schallwellen unterschiedlich an Wänden und Hindernissen

reflektiert. Auch der Weg der Schallwellen über das Chassis des Roboters unterscheidet sich je nach Richtung.

Man erkennt, dass die Beträge der gemessenen Wegdifferenzen durchgehend zu groß ausfallen. Dieses Ergebnis war qualitativ zu erwarten: Die Brechung der Schallwellen am Chassis des Roboters führt zu einer kurzen Verzögerung des Eintreffens der Schallwelle. Trifft die Schallwelle seitlich auf den Kopf, so kann sie direkt in das der Schallquelle zugewandte Mikrofon gelangen. Der Eintreffzeitpunkt am entfernteren, abgewandte Mikrofon ist hingegen durch den Einfluss der Brechung zusätzlich verzögert. Somit ergeben sich bei den Messungen stets betragsmäßig größere Wegdifferenzen.

Die Messungen fielen erkennbar schlechter aus, als sich Roboter B *vor* Roboter A befand. Dieser Umstand ist allein auf die Charakteristik der Umgebung zurückzuführen und ist für verschiedene Positionen im Raum nicht reproduzierbar.

Der Einfluss der Brechung ist in den Randbereichen, das heißt für Winkel um $\pm 90^\circ$, besonders groß. Dies wird erkennbar, wenn man sich die äußere Form des Kopfes des Aibo betrachtet. Aus seitlichen Richtungen wird das abgewandte Mikrofon besonders stark durch das Chassis des Kopfes abgeschirmt. Es gibt dann nur noch geringe Wellenanteile, die direkt in das abgewandte Mikrofon gelangen können. Der Messfehler ist dann entsprechend größer.

Anmerkung: Man beachte, dass zur Richtungsmessung keine Synchronisation erforderlich ist, da hier nur die *Differenz* der Eintreffzeitpunkte an beiden Mikrofonen verwendet wird.

Kapitel 6

Fazit

Mit dieser Arbeit konnte gezeigt werden, wie sich autonome Roboter eines Teams akustisch lokalisieren können.

Es wurde gezeigt, dass Distanzmessungen auch in Umgebungen mit lauten Störgeräuschen, die z.B. durch die Motoren und Laufgeräusche des Roboters selbst erzeugt werden, mit guter Genauigkeit möglich sind. Es traten im Experiment selten Fehler von über 10cm auf. Diese Genauigkeit ist z.B. für die Anwendung in einem Roboter-Fußballspiel ausreichend. Das hier vorgestellte Verfahren zur Distanzmessung funktioniert ohne Einschränkung auch im dreidimensionalen Raum.

Des weiteren wurden auch die Grenzen des Ansatzes aufgezeigt. Besonders beim gleichzeitigen Aussenden von Codes sowie bei lautem Umgebungsgeräuschen ist die maximal messbare Distanz erheblich eingeschränkt. Bei Verwendung des Aibos ergeben sich in störfreier Umgebung maximal 3 Meter bei gleichzeitigem Aussenden und ca. 7 Meter bei abwechselndem Aussenden. Laufgeräusche und Motorengeräusche reduzieren diese maximalen Distanzen auf etwa 2 Meter bzw. 4 Meter. Störgeräusche der Umgebung reduzieren die maximal messbaren Distanzen zusätzlich.

Es wurde gezeigt, dass die Uhrenfehler auf mobilen Robotern bereits nach kurzer Zeit zu erheblichen Fehlern bei akustischen Distanzmessungen führen können. Daher wurde eine Synchronisation entworfen, die akustisch fortlaufend den Offset zwischen den Roboteruhren berechnen kann und so den Fehler durch Uhrendrift minimieren kann. Dabei ist ein separater Kommunikationskanal wie z.B. ein drahtloses Netzwerk notwendig. Diese Methode kam in der Praxis erfolgreich zum Einsatz. Diese paarweise ‘*akustische Synchronisation*’ funktioniert auf dem SONY Aibo bei mittleren Bewegungsgeschwindigkeiten und gleichzeitigem

Aussenden mit einer Genauigkeit von *weniger als eine Millisekunde*. Die Synchronisation mit NTP über ein Funknetzwerk liefert aufgrund der schwankenden Netzwerk-Latenzen nicht selten Abweichungen von 20 Millisekunden und mehr.

Richtungsmessungen wurden im Rahmen dieser Arbeit lediglich in der Ebene und unter Verwendung von zwei Empfängermikrofonen untersucht. Für eine dreidimensionale Richtungsmessung sind bereits drei Mikrophone erforderlich. Die akustische Richtungsmessung erfordert grundsätzlich eine sehr hohe Genauigkeit bei der Bestimmung der Empfangszeitpunkte. Umgebungsbedingt ist normalerweise eine gewisse Ungenauigkeit durch Brechung und Reflexion am Roboterchassis und an Hindernissen unumgänglich. In den durchgeführten Experimenten schwankten die gemessenen Richtungen hierdurch um bis zu 35° . Richtungsmessungen können daher nur sehr eingeschränkt verwendet werden.

Die Messgenauigkeit ist deutlich besser, wenn der Kopf des empfangenden Roboters in Richtung zum Sender ausgerichtet ist. Der Fehler ist dann kleiner als 10° . Daher ist es möglich, dass ein Roboter allmählich seinen Kopf in Richtung der Schallquelle ausrichtet und auf diese Weise die Richtung bestimmen kann. In einem Roboter-Fußballspiel ist dies normalerweise nicht praktikabel, da ein gezieltes Umsehen nach Ball und Landmarken höhere Priorität hat.

Kapitel 7

Weitere Arbeit

7.1 Schall-Lokalisation

Das Tracking der Positionen der anderen Roboter geschieht in der vorliegenden Arbeit recht statisch. Alternativ könnte das Tracking der Distanz- und Richtungsmessungen anhand eines probabilistischen Filters (z.B. Kalman-Filter oder Monte-Carlo-Partikelfilter) durchgeführt werden. Dies könnte eine größere Robustheit gegenüber Störungen bringen.

In der vorliegenden Arbeit wurde der Einsatz mehrerer akustischer Frequenzen nicht berücksichtigt. In einem so genannten Multi-Carrier-CDMA-System wird zeitgleich auf mehreren Frequenzen gesendet. Dadurch können Fehlereinflüsse wie Multipath-Effekte besser herausgerechnet werden. Dies könnte die Fehler bei der Richtungsmessung deutlich reduzieren. Vor allem können dadurch höhere Übertragungsraten erzielt werden, was allerdings im Zusammenhang mit akustischer Lokalisation belanglos ist.

Ebenfalls noch nicht untersucht wurde die Möglichkeit mehrerer Chip-Typen. Kasami-Codes erreichen für die KKF optimale Korrelationseigenschaften *binärer* Sequenzen. Durch verschiedene Modulationen könnten auch Codes mit mehr als zwei Chip-Typen verwendet werden. Ob hierdurch tatsächlich bessere Korrelationseigenschaften erreicht werden können ist allerdings ungewiss. Die Verwendung solcher Codes in CDMA-Systemen spielen in der Praxis bisher keine Rolle.

In auf elektromagnetischen Wellen basierenden CDMA-Systemen findet man üblicherweise deutlich längere Codefolgen als die hier verwendeten (z.B. GPS: 1023, UMTS: über 20.000). Die Verwendung längerer Codefolgen ist deshalb erstrebenswert, da deutlich bessere Korrelationseigenschaften erzielt werden. Damit

wären auch größere Distanzen zwischen den Robotern bei gleichzeitigem Senden möglich. In der vorliegenden Arbeit wurden üblicherweise nur 63-Chip-Sequenzen verwendet, da längere Codeschemata zu einem beträchtlichen Fehler durch Bewegungsunschärfe führen.

Ein Ansatz könnte es sein, eine geschätzte relative Geschwindigkeit zwischen zwei Robotern zu verwenden. Dadurch könnte der durch die Bewegung entstehende Versatz bei der Auswertung der KKF direkt berücksichtigt werden. Dies verringert den Einfluss der Bewegungsunschärfe erheblich, so dass die Verwendung längerer Codefolgen möglich wird. Längere Codefolgen haben dabei allerdings auch einen höheren Aufwand bei der Auswertung der KKF zur Folge. Mit Hilfe der geschätzten relativen Geschwindigkeit kann der Rechenaufwand andererseits reduziert werden, da sich die Anzahl an möglichen Empfangszeitpunkten reduziert. Die KKF muss dann an weniger Punkten ausgewertet werden. (Allein die Kenntnis darüber, ob sich die Roboter annähern oder entfernen, reduziert den Rechenaufwand auf die Hälfte.)

Die Ansätze zur akustischen Lokalisation wurden bisher nur auf dem Aibo als Plattform getestet. Bei Verwendung besserer Hardware sind bessere Ergebnisse zu erwarten. Dazu gehört ein Lautsprecher, welcher größere Lautstärken und eine gleichmäßigere Abstrahlung in alle Richtungen ermöglicht. Der Einfluss von Störgeräuschen und Reflexionen könnte hierdurch verringert werden, was die Messung auch über größere Distanzen zuverlässiger machen würde. Multipath-Effekte, die durch Brechung am Roboter-Chassis entstehen, können verhindert werden, indem die Mikrophone mit freier Sichtverbindung aufgehängt werden und sich in der direkten Umgebung keine Teile des Chassis befinden. Dies könnte vor allem die Qualität der Richtungsmessung deutlich erhöhen.

7.2 Kooperative Selbstlokalisierung

In der vorliegenden Arbeit wurden lediglich die x-/y-Positionen der Roboter in der Ebene verwendet und bestimmt. Zusätzlich hierzu kann untersucht werden, wie die geschätzten Orientierungen und die sehr groben Richtungsmessungen mit in eine Partikelbewertung oder eine Ausgleichung eingehen können. Die Position aller Roboter wären dann in der Ebene eindeutig bestimmt. Um die sehr groben Richtungsmessungen einzubeziehen sind robuste Integrationsverfahren, wie z.B. ein probabilistischer Filter notwendig.

Die gleichzeitige akustische Distanzmessung ist aus den genannten Gründen bisher nur zwischen zwei oder maximal drei sendenden Robotern möglich. Ein Ex-

periment zur Ausglei chung unter Verwendung des ‘besten Dreiecks’ konnte daher bisher noch nicht durchgeföhrt werden.

Der *gezielte* Einsatz einzelner Distanzmessungen kann trotz der Beschränkung auf zwei gleichzeitig sendende Roboter in einem Szenario wie dem Roboterfußball sinnvoll sein. Die statische Sendeabfolge (gemäß Abschnitt 5.11) könnte hierfür zu einem Protokoll erweitert werden, das den gezielten Einsatz von Distanzmessungen optimiert. Beispielsweise könnte dieses Protokoll vorsehen, dass maximal drei Roboter *situationsabhängig* dann gleichzeitig senden, wenn diese zuletzt nicht weiter als eine bestimmte Distanz auseinander lagen. Zusätzlich ist auch denkbar, dass der Einsatz von Distanz- bzw. Richtungsmessungen *taktisch* über das *Verhalten* der Roboter gesteuert wird. Das Protokoll muss dezentral verwendbar sein, so dass fortlaufende Absprachen der Roboter untereinander möglich sind.

Anhang A

Fehler bei der Richtungsmessung

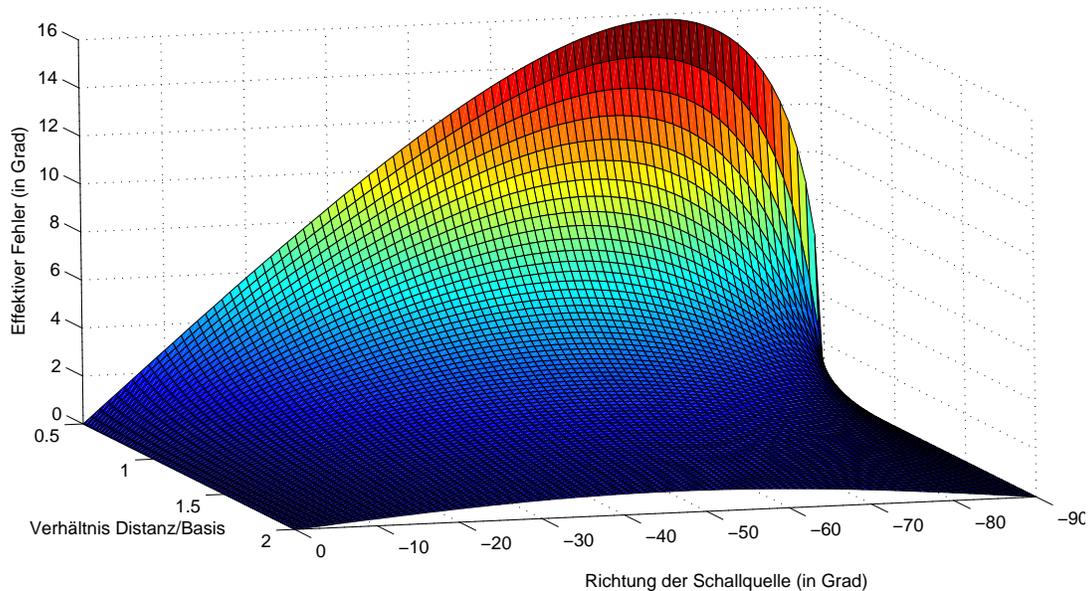


Abbildung A.1: Fehler bei der Richtungsmessung durch die Annahme sich parallel ausbreitender Schallwellen.

Bei der Richtungsmessung entsteht ein Fehler durch die vereinfachende Annahme, dass die Schallwellen-Fronten parallel den Weg zwischen Schallquelle und Empfangsmikrophon zurücklegen. Der durch diese Annahme entstehende effektive Messfehler ist in Abbildung A.1 aufgetragen. Der Fehler hängt von der tatsächlichen Richtung zur Schallquelle und dem Verhältnis zwischen Distanz zur Schallquelle und Länge der Basis ab. Man erkennt, dass sich der Fehler bei einer Richtung von etwa 70° am stärksten auswirkt.

Anhang B

Bewertungsfunktionen im Partikelfilter

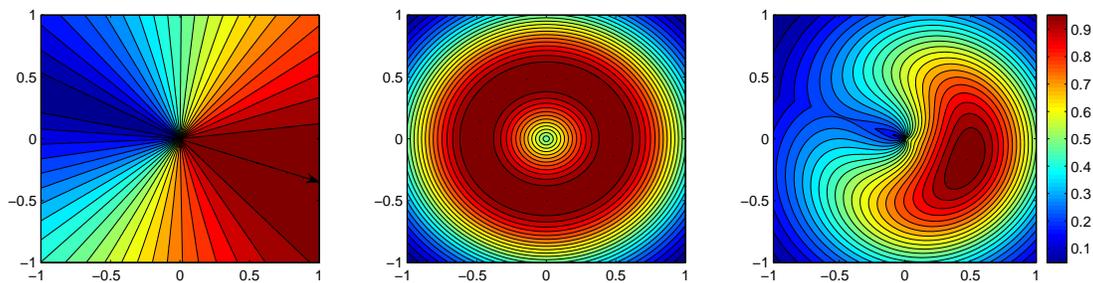


Abbildung B.1: *Bewertung von Partikeln. links: nach Richtung (Pfeil), Mitte: nach Distanz (0.5m), rechts: kombinierte Bewertung. Höhere Werte bedeuten jeweils eine bessere Bewertung. Der Referenzpunkt ist der Nullpunkt in der Diagramm-Mitte.*

Abbildung B.1 zeigt mögliche Bewertungsfunktionen in einem Partikelfilter. Die Bewertungen für Partikel an den entsprechenden Positionen sind in Form von Potenziellinien angegeben. Je nach Gewichtung der Bewertungen nach Distanzen bzw. Richtungen verändert sich die Form der kombinierten Bewertungsfunktion.

Anhang C

Verwendete Abkürzungen

- KKF: Kreuzkorrelationsfunktion
- KK: Kreuzkorrelation
- CDMA: code division multiple access
- FT: Fourier-Transformation
- DFT: Diskrete Fourier-Transformation
- FFT: Fast Fourier-Transformation

Anhang D

Verwendete Codefolgen

Hinweis: Zugunsten einer kompakteren Darstellung werden Chips mit Wert -1 als 0 geschrieben.

Zwei Kasami-Codes der Länge 15 ($n = 4$):

001111110111010; 111001000001100.

Acht Kasami-Codes der Länge 63 (Eine Codemenge mit $n = 6$):

1000001000011000 1010011110100011 1001001011011101 1001101010111111;
0010010101010110 0011101010011001 1110011000110100 010010010001100;
1100110010000101 1001110111010111 0111101100001110 001111011111000;
0001111100100010 1101001101001010 0100000101111010 110101000010001;
1011100001101100 0100111001110000 0011010110010011 000001111000010;
1111011011110001 0111010000000100 1101110001000000 101000001100101;
0110101111001011 0000000011101101 0000111111100111 111011100101011;
0101000110111111 1110100100111110 1010100010101001 011100110110110.

Zwei Kasami-Codes der Länge 255 ($n = 8$):

0010000100110110 1100111100010001 0110100001110101 1000101101110001
0111101111110001 0100000111111111 1010011111101001 0111011111001001
0011111110000000 1110000011100010 1101010010000111 1011000010001100
1000111111100100 0101010010011110 1101101101011001 111010101111010;
0110110001001110 0101010111100000 0101110110010111 1110000010110101
10101100011111000 1110111011101100 1111100111001111 1100101110000100
0100011100011010 0001000111010111 0011011011101100 0111010001011011
0000011001001011 0100011111000000 1111110111100101 101001111000110.

Ein Gold-Code der Länge 127:

1000000100000110 0001010001111001 0001011001110101 0011111010000111
0001001001101101 0110111101100011 0100101110111001 1001010101111111.

Anhang E

Pseudocode der FFT

```
function fft(n,f)
{
  if (n==1)
  {
    return f;
  }
  else
  {
    g = fft(n/2, [f(0),f(2),...,f(n-2)]);
    u = fft(n/2, [f(1),f(3),...,f(n-1)]);
    for k = 0 to n/2-1
    {
      c(k)      = g(k) + u(k) * exp(-2*pi*i*k/n);
      c(k+n/2) = g(k) - u(k) * exp(-2*pi*i*k/n);
    }
    return c;
  }
}
```

Abbildung E.1: Pseudo-Code einer FFT. Die Parameter sind das Array der Eingangsdaten f und die Anzahl an Eingangsdaten n . (Quelle: www.wikipedia.de)

Hinweis: Bei der Implementierung wurde darauf geachtet, die temporären Arrays g und u nur ein einziges Mal, aber mit doppelter Größe anzulegen. Die Ausführung konnte dadurch erheblich beschleunigt werden.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel verfasst habe.

Darmstadt, den 31. März 2008

Literaturverzeichnis

- [1] Jürgen Adamy, Kyriakos Voutsas, V.W.: Ein binaurales richtungshörsystem für mobile roboter in echoarmer umgebung. *Automatisierungstechnik* **51** (2003)
- [2] Valin, J.M., Michaud, F., Rouat, J., Letourneau, D.: Robust sound source localization using a microphone array on a mobile robot. In: *International Conference on Intelligent Robots and Systems, 2003. (IROS 2003)*. Proceedings. 2003 IEEE/RSJ. Volume 2. (2003) 1228–1233
- [3] Chen, S., Siu, M., Vogelgesang, T., et al.: The UNSW RoboCup 2001 sony legged league team report (2001)
- [4] Burkhard, H.D., Düffert, U., Hoffmann, J., Jüngel, M., Löttsch, M., Brunn, R., Kallnik, M., Kuntze, N., Kunz, M., Petters, S., Risler, M., v. Stryk, O., Koschmieder, N., Laue, T., Röfer, T., Spiess, K., Cesarz, A., Dahm, I., Hebbel, M., Nowak, W., Ziegler, J.: GermanTeam 2002 available online: <http://www.tzi.de/kogrob/papers/GermanTeam2002.pdf>.
- [5] Girod, L.: Development and characterization of an acoustic rangefinder. Technical report, University of Southern California, Los Angeles (2000)
- [6] Maloo, S., Xhafa, A.: Wideband CDMA. Technical report (February 1998)
- [7] R. Kohno, R. Meidan, L.M.: Spread spectrum access methods for wireless communications. *IEEE Communication Magazine* **41** (1995)
- [8] Desai, K.: *Wireless communication technologies* (2000)
- [9] Parkinson, B.: *Global positioning system: Theory and applications*. Technical report, American Institute of Aeronautics and Astronautics, Washington, D.C. (1996)
- [10] Miia Hermelin, K.N.: Correlation properties of the bluetooth combiner (feb 1999)

- [11] Baghaie, R.: Applications of transformation techniques in CDMA receivers (August 1999)
- [12] Khoo, K., Gong, G., Stinson, D.R.: Sequences with low cross correlation. Technical report (2003)
- [13] Gold, R.: Optimal binary sequences for spread spectrum multiplexing (corresp.). IEEE Transactions on Information Theory **13**(4) (1967)
- [14] Kasami, T.: Weight distribution formula for some class of cyclic codes. Technical Report No. R-285, Univ. of Illinois (1966)
- [15] Chen, C.E., Ali, A.M., Lorenzelli, F., Hudson, R.E., Yao, K.: Particle filtering approach to localization and tracking of a moving acoustic source in a reverberant room (2006)
- [16] Howard, A., Mataric, M.J., Sukhatme, G.S.: Cooperative relative localization for mobile robot teams: An egocentric approach (2003)
- [17] Sinha, D.K.: Cooperative self-localization in a multi-robot-no-landmark scenario using fuzzy logic (2004)
- [18] Fox, D., Burgard, W., Thrun, S.: In special issue of autonomous robots on heterogeneous multi-robot systems, 8(3), 2000. to appear. a probabilistic approach to collaborative multi-robot localization
- [19] Rekleitis, I., Dudek, G.: Probabilistic cooperative localization and mapping in practice (2003)
- [20] Karl-Dirk Kammeyer, K.K.: Digitale Signalverarbeitung. B. G. Teubner Verlag (2006)
- [21] Leinen, S.: Skriptum zu Signalverarbeitung II (2007) Technische Universität Darmstadt, Institut für Physikalische Geodäsie, URL: http://www.ipg.tu-darmstadt.de/download/ps2_skript.pdf.
- [22] Welch, L.: Lower bounds on the maximum cross correlation of signals. IEEE Transactions on Information Theory **20**(3) (1974)
- [23] Plausinaitis, D.: GPS signals and receiver technology, slideset no. 10 (2006) University of Aalborg, Department of Communication Technology, available online at http://kom.aau.dk/~dpl/courses/mm10_slides.pdf.
- [24] Mills, D.L.: Network time protocol (version 3). Technical report, University of Delaware (1992)

- [25] Leinen, S.: Skriptum zu Grundlagen der Ausgleichsrechnung (2006) Technische Universität Darmstadt, Institut für Physikalische Geodäsie, URL: http://www.ipg.tu-darmstadt.de/download/ps1_skript_v1_1.pdf.
- [26] Röfer, T., Brunn, R., Czarnetzki, S., Dassler, M., Hebbel, M., Jüngel, M., Kerkhof, T., Nistico, W., Oberlies, T., Rohde, C., Spranger, M., Zarges, C.: GermanTeam team report. In: RoboCup 2005: Robot Soccer World Cup IX Preproceedings. RoboCup Federation (2005)
- [27] Röfer, T., Brose, J., Göhring, D., Jüngel, M., Laue, T., Risler, M.: German-team 2007 - the german national robocup team. Technical report, DFKI Bremen, TU Darmstadt, HU Berlin, Universitt Bremen (2007)