

# Detektion von Messfehlern eines relativen Lokationssystems mittels ANFIS-Methoden

Studienarbeit am  
Telecooperation Office  
Institut für Telematik  
Fakultät für Informatik  
Universität Karlsruhe (TH)

von

cand. inform.  
**Christian Bittel**

Betreuer:

Prof. Dr. W. Juling  
Dipl.-Inform. Martin Bechtold  
Dipl.-Inform. Till Riedel

Tag der Anmeldung: 1. August 2007  
Tag der Abgabe: 29. Februar 2008

---



---

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Karlsruhe, den 29. Februar 2008



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Zielsetzung der Arbeit . . . . .	2
1.3	Gliederung der Arbeit . . . . .	2
<b>2</b>	<b>Fehlerquellen in Lokationssystemen</b>	<b>3</b>
2.1	Systematische Fehler . . . . .	3
2.2	Zufällige Fehler . . . . .	4
<b>3</b>	<b>Absolute und relative Lokationssysteme</b>	<b>7</b>
3.1	Entfernungs- und Positionsbestimmung . . . . .	7
3.1.1	Laufzeitmessungen . . . . .	7
3.1.1.1	GPS . . . . .	7
3.1.2	Signalstärkemessungen . . . . .	9
3.1.2.1	RSSI . . . . .	9
3.1.3	Koppelnavigationssysteme . . . . .	10
3.1.3.1	Inertialnavigation . . . . .	10
3.1.3.2	Koppelnavigation . . . . .	10
3.2	Kombinierte Entfernungs- und Orientierungsbestimmung . . . . .	10
3.2.1	Infrarot . . . . .	10
3.2.2	Ultraschall . . . . .	12
<b>4</b>	<b>Zustandsvorhersage, Fehlerabschätzung und Fehlermodelle</b>	<b>15</b>
4.1	Kalman-Filter . . . . .	15
4.2	Partikel-Filter . . . . .	16
4.3	Distributed Precision Based Localization . . . . .	16
4.4	Sensorvalidierung mittels neuronaler Netze . . . . .	17
4.5	Kontextklassifikation mittels ANFIS-Methoden . . . . .	17
<b>5</b>	<b>Architektur des Systems und Systemidentifikation</b>	<b>19</b>
5.1	Architektur . . . . .	19
5.2	Systemidentifikation . . . . .	20
5.2.1	Clustering . . . . .	21
5.2.1.1	Subtraktives Clustering . . . . .	21
5.2.2	Methode der kleinsten Fehlerquadratrate . . . . .	22
5.2.3	ANFIS . . . . .	23
5.2.4	Hybrides Lernen . . . . .	24
5.3	Online-System zur Qualitätsanalyse . . . . .	24
5.3.1	Fuzzy-Inferenz-System . . . . .	24

---

5.3.1.1	TSK-FIS . . . . .	25
5.3.1.2	Semantische Normalisierung . . . . .	25
<b>6</b>	<b>Versuchsaufbau</b>	<b>27</b>
6.1	Verwendete Hardware . . . . .	27
6.2	Erster Messaufbau zur Sammlung diskreter Werte . . . . .	28
6.3	Zweiter Messaufbau zur Sammlung kontinuierlicher Werte . . . . .	29
<b>7</b>	<b>Aufbau des ANFIS-Trainings und dessen Ergebnisse</b>	<b>33</b>
7.1	Datensätze aus dem Messaufbau . . . . .	33
7.2	Merkmalsauswahl . . . . .	33
7.3	Trainingsablauf . . . . .	34
7.4	Evaluierung der Trainingsergebnisse . . . . .	35
7.4.1	Diskrete Werte . . . . .	35
7.4.2	Kontinuierliche Werte . . . . .	36
<b>8</b>	<b>Anwendung des Qualitätssystems zur Ergänzung von Lokations- Algorithmen</b>	<b>39</b>
8.1	Hyperbelnavigation . . . . .	39
8.2	Praxisanwendung der Hyperbelnavigation . . . . .	40
8.3	Methode der kleinsten Fehlerquadrate . . . . .	42
8.4	Gefilterte Methode der kleinsten Fehlerquadrate . . . . .	42
8.5	Gewichtete Methode der kleinsten Fehlerquadrate . . . . .	42
<b>9</b>	<b>Zusammenfassung und Ausblick</b>	<b>45</b>
<b>A</b>	<b>Anhang</b>	<b>47</b>
<b>B</b>	<b>Anhang - Matlab</b>	<b>57</b>
	<b>Literatur</b>	<b>59</b>

# 1. Einleitung

## 1.1 Motivation

Seit einigen Jahren gibt es Forschungsprojekte, die sich mit der Allgegenwärtigkeit der digitalen Informationsverarbeitung im Leben der Menschen beschäftigen, dem sogenannten „Ubiquitous Computing“. Ziel ist es Computersysteme so in den Alltag zu integrieren, dass diese überall verfügbar sind und gar nicht mehr direkt wahrgenommen werden. Ubiquitäre Systeme, also für das Ubiquitous Computing entwickelte Computersysteme, versuchen nun diesem Ziel nahe zu kommen.

Die Anwendungsmöglichkeiten sind vielfältig. So kann man sich einen durch ein Computersystem ergänzten Kühlschrank vorstellen, der den Besitzer über Produkte informiert die das Mindesthaltbarkeitsdatum überschritten haben, der selbstständig über das Internet Ersatz bestellt, wenn bestimmte Produkte verbraucht sind und Rezeptvorschläge macht, die sich mit den aktuell vorhandenen Produkten umsetzen lassen.

Ein weitere Anwendung ist ein ubiquitäres System, das in einer Wohnung die Personen erkennt, die sich in einem Zimmer aufhalten und beispielsweise die Beleuchtung oder die Musikbeschallung an die Vorlieben der Personen anpasst.

Da in einem ubiquitären System der Benutzer im Mittelpunkt steht, ist es wichtig, dass das System den Kontext, also den Zustand der Umgebung eines Systems oder Benutzers, richtig erkennen kann. Zum Kontext gehören Wissen über den Benutzer, sein soziales Umfeld und seine Aufgaben. Außerdem Informationen über die physische Umgebung, also die Infrastruktur, physikalische Bedingungen und den Aufenthaltsort.

Gerade der Aufenthaltsort eines Benutzers oder Systems ist in vielen ubiquitären Systemen eine der wichtigsten Informationen. Beispielsweise in einem Navigationssystem das dem Benutzer Richtungsangaben macht um, ausgehend von der aktuellen Position und Orientierung des Benutzers, diesen an das gewünschte Ziel zu bringen. Hier kann das System nur sinnvoll funktionieren, wenn es die Position und Orientierung des Benutzers möglichst exakt bestimmen kann.

Um den Aufenthaltsort einer Person oder eines Systems zu bestimmen gibt es verschiedene Verfahren, die jeweils versuchen eine möglichst exakte Positionsangabe zu ermitteln. Solche Positionsangaben sind aber immer auch fehlerbehaftet. Dieser Fehler kommt durch verschiedene Störeinflüsse in der Umgebung zustande,

beispielsweise wenn Gegenstände oder Gebäude Funksignale blockieren oder umlenken.

Zur Verringerung dieses Fehlers können bessere Sensoren oder komplexere Algorithmen eingesetzt werden. Dadurch werden die Systeme aber teurer, benötigen mehr Rechenleistung und haben einen größeren Energieverbrauch. Ein Restfehler, wie beispielsweise das natürliche Rauschen, lässt sich dabei dennoch nicht vermeiden.

Hätte man eine Einschätzung über die Verlässlichkeit der ermittelten Positionsangaben, so könnte man auch weniger komplexe Systeme einsetzen und z.B. nur Positionsangaben mit einer hohen Verlässlichkeit verwenden. Eine Aussage über die Verlässlichkeit wird von bestehenden Verfahren dabei nicht oder nur in allgemeiner Weise gemacht. In dieser Arbeit wird ein System vorgestellt, das eine Detektion von Messfehlern ermöglicht und für die einzelnen Messungen jeweils eine Verlässlichkeits- bzw. eine Qualitätsaussage macht.

## 1.2 Zielsetzung der Arbeit

Das Ziel dieser Arbeit ist es, ein System zur Qualitätsanalyse von relativen Lokationssystemen zu erstellen, es mit Hilfe von Ultraschall-Sensorknoten, kleinen drahtlos kommunizierenden Computersystemen, auf einen praktischen Einsatz hin zu optimieren und die Ergebnisse der Qualitätsanalyse zu evaluieren. Das System soll dabei für jede Entfernung- oder Winkelmessung eines relativen Lokationssystems eine individuelle Verlässlichkeitsaussage liefern, die eine Tendenz angibt, ob ein zuvor festgelegter Fehler überschritten wird.

Die Erstellung des Systems zur Qualitätsanalyse soll dabei kein Wissen über die Arbeitsweise des zu beurteilenden Lokationssystems benötigen. Um dies zu erreichen muss die Erstellung automatisiert ablaufen. Die Eignung der Algorithmen des in dieser Arbeit beschriebenen Systems für diese Aufgabe wurde in [Berc07] gezeigt.

Schließlich soll anhand einer Praxisanwendung die Einbeziehung der Ergebnisse der Qualitätsanalyse in einen Algorithmus gezeigt und das Verbesserungspotential demonstriert werden.

## 1.3 Gliederung der Arbeit

Im folgenden Kapitel werden zunächst die Fehlertypen und Fehlerquellen die in Lokationssystemen auftreten können vorgestellt. Kapitel 3 betrachtet absolute und relative Lokationssysteme und ihre Verwendung in der ubiquitären Informationstechnologie. In Kapitel 4 werden Verfahren zur Zustandsvorhersage und Fehlerabschätzung in Lokationssystemen vorgestellt.

Kapitel 5 gibt einen Überblick über die Systemarchitektur, die Systemidentifikation und ein Online-System zur Qualitätsanalyse. Kapitel 6 beschreibt die eingesetzte Hardware und den Versuchsaufbau zur Sammlung von Messergebnissen mit den verwendeten Ultraschall-Sensorknoten. In Kapitel 7 werden das Vorgehen zum Training und die Evaluierungsergebnisse des Systems beschrieben.

Kapitel 8 schildert den Einsatz der Qualitätswerte aus dem System in einer Praxisanwendung und die dadurch erreichbare Verbesserung der Ergebnisse. Das letzte Kapitel enthält eine Zusammenfassung der Arbeit und einen Ausblick auf zukünftige Erweiterungen und Fortsetzungen der Fragestellungen.



## 2. Fehlerquellen in Lokationssystemen

Im Umgang mit Lokationssystemen ist immer zu bedenken, dass die Positionsbestimmung fehlerbehaftet ist. Bei jedem System gibt es je nach Typ, technischen Daten, verwendeten Algorithmen und auch abhängig von der Umgebung, in der es eingesetzt wird, unterschiedlich große Fehler, also Abweichungen vom gesuchten Wert, in der ermittelten Positions- oder Orientierungsangabe. Im Folgenden werden die beiden gebräuchlichsten Fehlerklassen, die systematischen und die zufälligen Fehler, vorgestellt.

### 2.1 Systematische Fehler

Systematische Fehler treten reproduzierbar auf und ergeben einen konstanten oder proportionalen Fehler in der Positionsbestimmung. Dieser lässt sich nicht durch eine Mittelwertbildung aus mehreren Wiederholungen der Messungen verringern. Im Gegensatz dazu ist eine Mittelung bei zufälligen Fehlern möglich, um den Fehler zu reduzieren.

Zu den systematischen Fehlern gehört die Dekalibrierung der verwendeten Sensoren im Laufe der Zeit, eine Veränderung ihrer Ausrichtung, der Einfluss konstanter Umweltbedingungen am Einsatzort, wie Wände die Signale blockieren, die Verteilung und die Anzahl der eingesetzten Sensorknoten, Fehler im eingesetzten Algorithmus oder im eingesetzten Modell, Rückwirkung der Messinstrumente auf das zu untersuchende System oder auch Ablesefehler, beispielsweise bei einem schrägen Blick auf eine Skala.

Um diesem Fehlertyp zu begegnen kann man versuchen, die Umstände der Messungen zu verändern, zum Beispiel durch Messungen bei anderen Temperaturen, eine Veränderung der Orientierung von Sensorknoten zueinander oder das Bestimmen der Lokation zweier Sensorknoten durch die umliegenden Knoten, siehe dazu [KrHB07].

Nachfolgend werden systematische Fehler näher vorgestellt, die bei mit Ultraschall arbeitenden Sensorknoten, wie sie in dieser Arbeit Verwendung finden, auftreten können. Die Ausbreitungsgeschwindigkeit des Schalls in der Luft ist abhängig von Temperatur und in geringerem Maße von der Luftfeuchtigkeit [Harr66]. Im Algorithmus eines mit Ultraschall arbeitenden Systems muss dies berücksichtigt werden.

Hilfreich sind dabei Sensorknoten, die Temperatur und/oder Luftfeuchtigkeit messen und in den Algorithmus einspeisen können.

Der Schalldruckpegel eines Ultraschallsignals der Frequenz 170kHz wird in der Luft um bis zu 10dB/m gedämpft. Ein Signal der Frequenz 40kHz dagegen nur um etwa 1dB/m [Ashe97]. Im Algorithmus ist also eventuell die verwendete Frequenz der Ultraschallsignale zu berücksichtigen. Zusätzlich gilt unabhängig von der Frequenz für Schallimpulse das Abstandsgesetz das besagt, dass der Schalldruckpegel bei verdoppelter Entfernung um 6dB abnimmt.

Damit ergibt sich auch eine Beschränkung des nutzbaren Frequenzspektrums nach oben hin. Nach unten wird dieses Spektrum dadurch begrenzt, dass Töne im Bereich um die 20kHz für empfindliche menschliche Ohren hörbar sind. Mit Signalen im bei Ultraschallsystemen häufig genutzten Frequenzbereich von 40kHz bis 70kHz lassen sich Entfernungsmessungen im Bereich bis zu 10 Metern durchführen [Ward98].

Durch Objekte in der Umgebung eines Ultraschallsystems kann es zu verschiedenen Effekten kommen. Zum einen können die Signale an Objekten umgelenkt werden, wodurch sie eventuell auch ohne direkte Sichtverbindung ihr Ziel erreichen. Allerdings verringert sich in diesem Fall die Genauigkeit der Messung, die von einer direkten Verbindung ausgeht. Zum anderen können Objekte wie Wände, massive Möbelstücke oder auch Bildschirme die Ultraschallsignale reflektieren. Solche Signale erreichen den Empfänger auf einem längeren Weg als ein direktes Signal und sorgen somit ebenfalls für eine Verringerung der Genauigkeit. Wird ein Signal zu verschiedenen Zeitpunkten empfangen, so kann man davon ausgehen, dass das erste Signal den direkten Weg genommen hat.

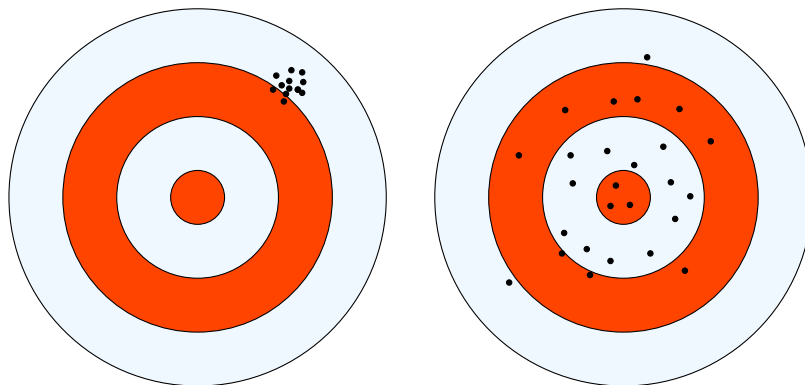


Abbildung 2.1: Systematischer (links) und zufälliger/stochastischer Fehler (rechts) am Beispiel einer Zielscheibe.

## 2.2 Zufällige Fehler

Zufällige Fehler, auch als stochastische Fehler bezeichnet, treten zufällig, also nicht in eine bestimmte Richtung auf. Sie verändern sich bei der Wiederholung einer Messung und folgen aufgrund des Zentralen Grenzwertsatzes, der aussagt, dass die Summe unabhängiger Variablen mit endlicher Varianz annähernd normalverteilt ist, häufig einer Normalverteilung um den wahren Wert einer Messung und lassen sich durch die Standardabweichung oder Varianz beschreiben.

Zu den zufälligen Fehlern gehören das natürliche Rauschen, das Quantisierungsrauschen und zufällige Störungen die einmalig auftreten. Durch mehrmalige Wiederholungen der Messungen und eine Bildung des Mittelwertes lässt sich die Auswirkung dieses Fehlertypes verringern.

Das natürliche Rauschen ist ein unvermeidbarer zufälliger Fehler und beeinflusst die Qualität der Messungen. Die empfangenen Signale werden dadurch verändert und auch die Sensorwerte werden durch ein Rauschen beeinflusst. Als Gegenmaßnahme kann man versuchen in einem in der jeweiligen Umgebung möglichst rauscharmen Frequenzbereich zu arbeiten.

Quantisierungsrauschen tritt auf wenn analoge Messwerte in eine digitale Repräsentation überführt werden. Die digitalen Werte sind im Gegensatz zu den analogen diskret, so dass es eine Rundungsdifferenz gibt, den Quantisierungsfehler. Um diesen Fehler zu verkleinern wird die Anzahl der möglichen diskreten Werte in der digitalen Repräsentation erhöht.



## 3. Absolute und relative Lokationssysteme

Zur Bestimmung einer Lokation lassen sich verschiedene Techniken einsetzen, die eine absolute oder relative Position bzw. Entfernung ermitteln. Einige Systeme bestimmen auch die Orientierung von Personen oder Objekten. Die einzelnen Techniken unterscheiden sich unter anderem in ihrer Komplexität, der Ausbreitungsgeschwindigkeit ihrer Signale, ihrem Preis und der Einsatzmöglichkeit innerhalb von Gebäuden. Im Folgenden werden einige Techniken kurz vorgestellt und ihre Besonderheiten erläutert.

### 3.1 Entfernung- und Positionsbestimmung

#### 3.1.1 Laufzeitmessungen

In diesem Abschnitt wird mit GPS (Global Positioning System) ein System vorgestellt, das durch Laufzeitmessungen von Satellitensignalen eine absolute Position liefert. Bei der Laufzeitmessung wird der Umstand genutzt, dass ein Signal sich gleichmäßig in einem Medium ausbreitet. Die Verzögerung zwischen Aussenden und Empfangen eines Signals hängt damit direkt von der zwischen zwei Systemen liegenden Entfernung ab. Durch die Bestimmung der Ausbreitungsverzögerung lässt sich mit dem Wissen über die Ausbreitungsgeschwindigkeit des Signals im vorliegenden Medium die Entfernung bestimmen.

##### 3.1.1.1 GPS

GPS ist die bekannteste und am weitesten verbreitete Technik zur Bestimmung einer absoluten Position. Das System besteht aus 29 Satelliten, die sich gleichzeitig in 20180 km Höhe im Orbit befinden und als Sender dienen. Dazu kommt eine Vielzahl individueller Empfänger, die durch Signale der Satelliten ihre Position auf der Erde und die exakte Zeit bestimmen können. Diese Positionsbestimmung wird aus den Laufzeitmessungen mittels der Hyperbelnavigation durchgeführt (siehe auch Kapitel 8.1).

Durch die Positionierung der Satelliten auf 6 verschiedenen Umlaufbahnen mit einer Neigung von  $55^\circ$  gegenüber dem Äquator ist sichergestellt, dass an jedem Ort der Erde jederzeit die Signale von mindestens 4 Satelliten gleichzeitig empfangen werden können. Für jeden dieser Satelliten wird die Laufzeit der ausgesendeten Signale zum Empfänger gemessen. Daraus lässt sich die Entfernung zu den Satelliten

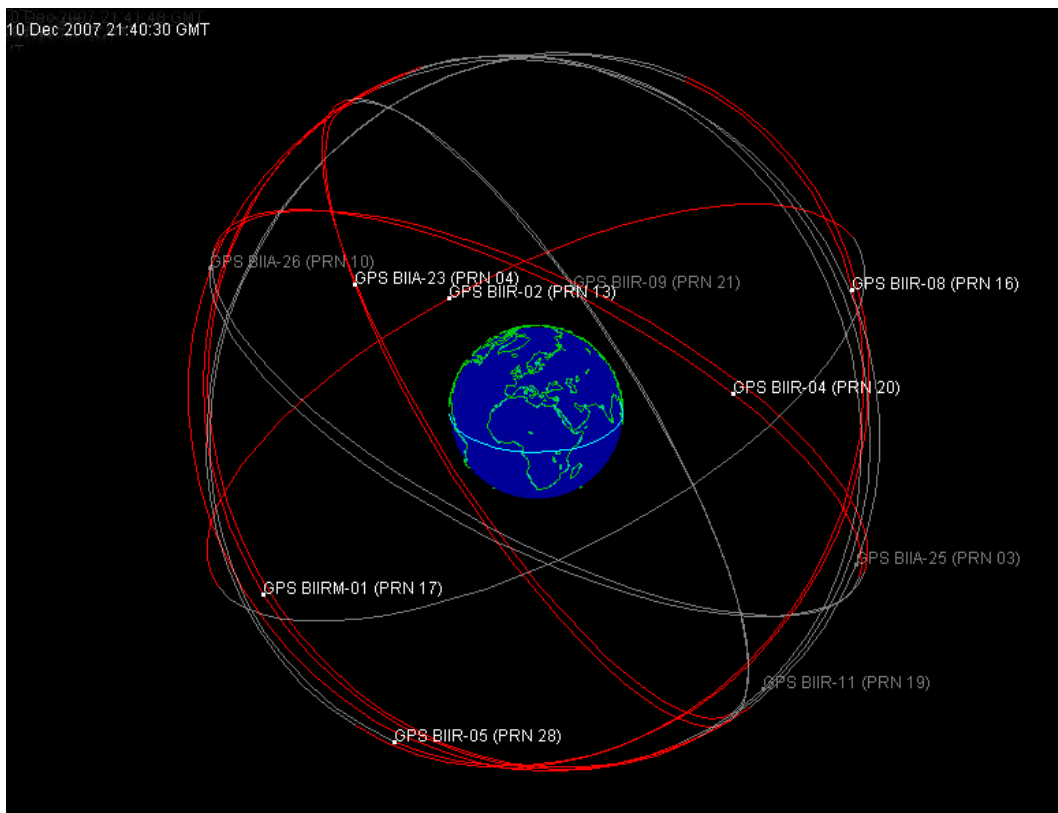


Abbildung 3.1: Illustration der Umlaufbahnen einiger GPS-Satelliten.

bestimmen. Die Uhren der Empfänger weisen gegenüber den in den Satelliten synchron arbeitenden Atomuhren einen Zeitfehler auf. Deswegen benötigt man nicht nur drei Satelliten für die Positionsbestimmung, sondern einen vierten, der den Zeitfehler korrigiert und somit eine exakte Laufzeitmessung und die Bestimmung der exakten Uhrzeit ermöglicht.

Mit dem russischen GLONASS und dem europäischen GALILEO befinden sich mit GPS konkurrierende System im Aufbau bzw. in Planung. Weitere Informationen zur Satellitennavigation finden sich in [Zogg06].

Aufgrund der globalen Verfügbarkeit und der Genauigkeit der Positionsangaben, mit Fehler von weniger als 10 Metern, haben sich GPS-Systeme in vielen Bereichen der Lokationsbestimmung durchgesetzt. Zufällige Fehler treten durch das Messrauschen und Zeitverzögerungen vor allem im GPS-Empfänger auf. Hinzu kommt noch die Ungenauigkeit der gespeicherten Satellitenbahnen und der Satellitenuhren. Zu den systematischen Fehlern gehören der Einfluss von Ionosphäre und Troposphäre, der Mehrwegempfang und die Satellitenkonstellation. Bis auf den Mehrwegempfang und das Messrauschen im GPS-Empfänger lassen sich alle Fehler durch den Einsatz von Differential-GPS (DGPS) reduzieren, siehe dazu [Zogg06].

Für den Einsatz in ubiquitären Systemen gibt es dennoch eine Reihe von Einschränkungen. So ist die Technik eines GPS-Empfängers für kleine Sensorknoten zu groß, verbraucht zu viel Strom und ist zu teuer. Die Genauigkeit ist je nach Anwendung möglicherweise nicht ausreichend. Entscheidend ist aber der fehlende Empfang innerhalb von Gebäuden. Lokationsbestimmung mittels GPS funktioniert somit nur außerhalb von Gebäuden. Für viele Anwendungen der ubiquitären Informationstechnik kommt GPS damit nicht in Frage.

### 3.1.2 Signalstärkemessungen

In ubiquitären Systemen spielt der Kontext eine wichtige Rolle. Für den örtlichen Kontext ist dabei in vielen Fällen nicht die absolute Position entscheidend, sondern die Nähe verschiedener Objekte oder Benutzer zueinander. Deswegen kommen häufig relative Lokationssysteme zum Einsatz.

Im Folgenden wird mit RSSI eine Technik vorgestellt, die eine Entfernungsmessung durch das Bestimmen der Signalstärken erreicht. Dabei wird versucht, eine gemessene Signalstärke auf eine Entfernung abzubilden. Um dies zu erreichen ist entweder Wissen über die Dämpfung der Signale im vorliegenden Medium nötig oder eine umfangreiche Datensammlung mit nachgemessenen Entfernungen und zugehörigen Signalstärken für den gewünschten Einsatzort anzulegen.

#### 3.1.2.1 RSSI

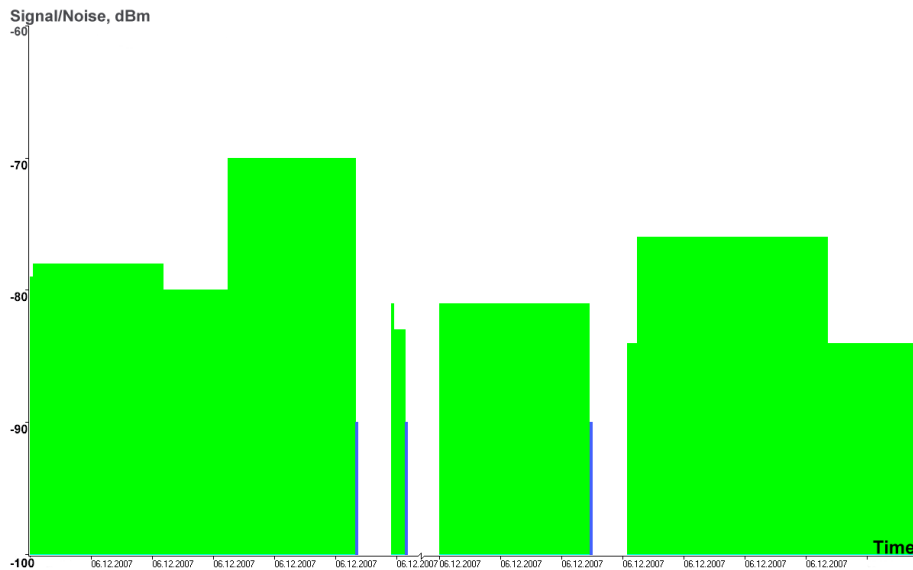


Abbildung 3.2: Einfache RSSI Darstellung auf einem Laptop mit Wireless-LAN.

RSSI steht für Received Signal Strength Indication, also das Messen der Signalstärke eines empfangenen Funksignals. Besonders attraktiv ist die Entfernungsmessung per RSSI in Systemen, in denen die Kommunikation bereits mittels eines Funksignals arbeitet. Wireless LAN und GSM-Netze sind verbreitete Beispiele. Durch das Hinzufügen einer zusätzlichen Softwarekomponente lässt sich mit der vorhandenen Hardwarebasis eine Entfernungsbestimmung durchführen.

Dazu wird die Signalstärke des empfangenen Signals bestimmt und mit einem Modell der Signaldämpfung oder einer Tabelle von Beispielmessungen auf einen möglichst passenden Entfernungswert abgebildet. Die ermittelten Entfernungen sind sehr ungenau und sehr stark von den Umweltbedingungen abhängig. Um die Genauigkeit etwas zu verbessern kann eine Kalibrierung für ein konkretes Einsatzgebiet durchgeführt werden. Siehe dazu [HLIN01].

Zufällige Fehler beim Einsatz von RSSI sind das Rauschen in der Umgebung und im Empfänger. Zu den systematischen Fehlern gehört die Signaldämpfung durch Hindernisse. Letztere lassen sich ohne explizites Wissen über den Einfluss der Umgebung auf die Signalstärke nicht reduzieren. Aufgrund dieser Charakteristik ist die Lokationsbestimmung mittels RSSI nur bei geringen Anforderungen an die Qualität der bestimmten Entfernungen geeignet oder in funkbasierten Systemen in die keine weitere Lokationstechnik eingebaut werden kann oder soll.

### 3.1.3 Koppelnavigationssysteme

#### 3.1.3.1 Inertialnavigation

Ein inertiales Navigationssystem, auch als Trägheitsnavigationssystem bezeichnet, ist ein mechanisches System und gehört zur Gruppe der Koppelnavigationssysteme, die eine fortlaufende Ortsbestimmung durch Messen der Geschwindigkeit und der Bewegungsrichtung durchführen. Es dient zur Erfassung von Geschwindigkeit, Position und Orientierung eines bewegten Objektes unabhängig von äußeren Eingaben. Verwendung findet die Inertialnavigation vor allem in Flugzeugen, Raumschiffen und Militärfahrzeugen. Einen Überblick über Technik und Geschichte der Inertialnavigation bietet [King98].

Das Verfahren ermittelt mittels Laser-Sensoren für die drei Raumachsen die Beschleunigung und die Drehung. Durch Integrieren der Beschleunigungswerte über die Zeit wird die Geschwindigkeit ermittelt, durch eine weitere Integration die Wegstrecke. Auf gleiche Weise erhält man aus den Drehraten die Orientierung des Systems. Durch das Integrieren entsteht dabei ein kumulativer, systematischer Fehler in der Bestimmung von Position und Orientierung.

Die Mechanik für die Ermittlung der Beschleunigungs- und Rotationsraten ist sehr aufwendig und teuer. Ein modernes, in sich geschlossenes, inertiales Navigationssystem kommt deshalb nur in größeren Systemen zum Einsatz. Für den Einsatz in einem kleineren relativen Lokationssystem wird eine vereinfachte Variante der Koppelnavigation benötigt.

#### 3.1.3.2 Koppelnavigation

Eine gegenüber der Inertialnavigation vereinfachte Variante der Koppelnavigation wird in [RaDM03] vorgestellt. Die Genauigkeit der bestimmten Position und Orientierung fällt durch weniger komplexe Hardware etwas geringer aus, als bei der Inertialnavigation. Dafür sind diese Koppelnavigationssysteme günstiger und kleiner. Für größere Lokationssysteme bietet sich die Koppelnavigation damit als unterstützende Technologie an. Sie kann bei einem zeitweisen Ausfall des Hauptlokationssystems für einen gewissen Zeitraum recht präzise Messungen liefern.

Da die Charakteristik der kumulativen, systematischen Fehler dafür sorgt, dass über einen längeren Zeitraum ohne Kalibrierung der Fehler der Messungen stetig zunimmt, lässt sich über die verstrichene Zeit seit der letzten Kalibrierung eine ungefähre Abschätzung über die Qualität der Messungen vornehmen. Solange eine neue Aufgabenstellung die Arbeitsbereiche der Sensoren nicht überschreitet lässt sich die Koppelnavigation recht einfach für diese Aufgabenstellung anpassen.

## 3.2 Kombinierte Entfernungs- und Orientierungsbestimmung

In diesem Abschnitt werden mit einem infrarot-basierten System von Nakamura et al. und den ultraschall-basierten BRICK-Sensorknoten zwei Systeme vorgestellt, die neben einer Entfernungs- oder Positionsbestimmung auch eine Orientierung von Personen oder Objekten ermitteln.

### 3.2.1 Infrarot

Es gibt verschiedene relative Lokationssysteme auf Infrarot-Basis, wie z.B. Active Badge [WHFG92]. Diese sind zumeist für den Einsatz in geschlossenen Räumen gedacht und verwenden kleine Sender, die auf dem zu lokalisierenden Objekt angebracht oder von der zu lokalisierenden Person getragen werden.



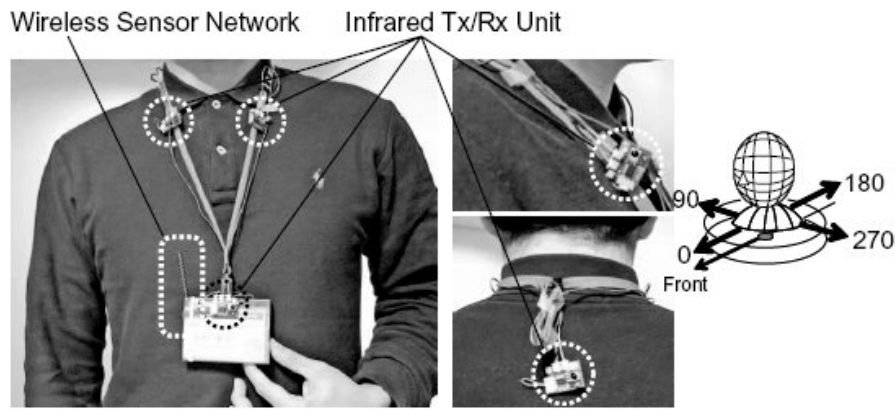


Abbildung 3.3: Beispielfähige Anordnung der Nametag Infrarotelemente. (Quelle: [NNMM<sup>+</sup>07])

Nakamura et al. stellen in [NNMM<sup>+</sup>07] ein System vor, das mit Infrarotsendern und -empfängern bestückte umhängbare Namensschilder („Nametags“) verwendet, um die Position und Orientierung von Personen zu bestimmen. Jedes Namensschild enthält dabei vier Infrarotsender und -empfänger, deren Ausrichtung beim Tragen des Namensschildes den Bereich vor und hinter der tragenden Person sowie große Teile an den beiden Seiten abdecken (siehe dazu Abbildung 3.3).

Jeder Infrarotsender sendet dabei periodisch eine Unit-ID für seine Position im Nametag und eine Modul-ID, die das Nametag identifiziert. Ein anderes Nametag, das diesen Infrarotimpuls empfängt leitet diese Informationen über ein im Nametag integriertes aktives RFID-Tag an einen zentralen Rechenknoten weiter. Dort wird mittels eines als topologische Annäherung („topological approach“) bezeichneten Verfahrens eine Abschätzung für Position und Orientierung vorgenommen.

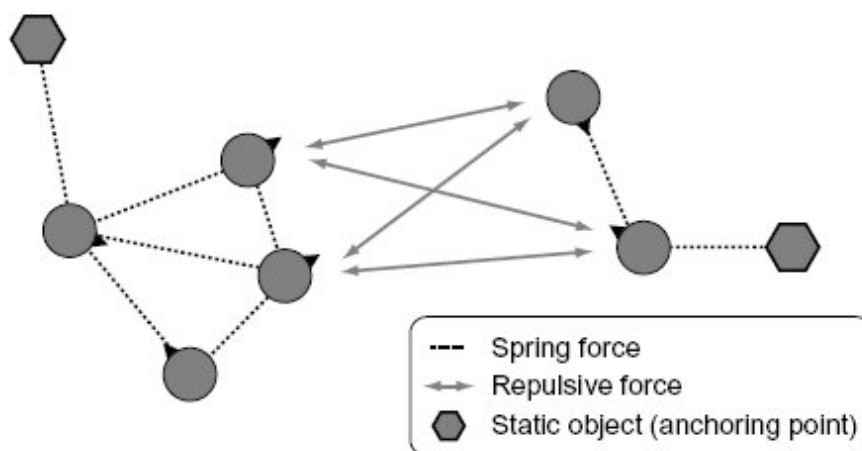


Abbildung 3.4: Ein dynamisches Federkraft-Modell. (Quelle: [NNMM<sup>+</sup>07])

Das Verfahren beinhaltet ein Federkraft-Modell (siehe Abbildung 3.4), das für je zwei Nametag Unit-IDs verschiedener Nametags eine gewichtete Verbindung herstellt. Dieses Gewicht simuliert, vergleichbar einer Feder, eine anziehende Kraft, wenn eine Empfangsmeldung für diese beiden IDs vorliegt und eine abstoßende Kraft, wenn keine solche Meldung empfangen wurde. Für die Bestimmung der Ori-

entierung wird nun noch eine dynamische Drehkraft eingeführt, die von den lokalen Unit-IDs Empfangsmeldungen abhängt.

Von einer zufälligen Anfangspositionierung aus werden nun alle im System eingebuchten Nametags abhängig von den Federkräften in ihrer Position verschoben und in der Orientierung gedreht, um die Summe der Kräfte zu minimieren. Die so ermittelten Positionen dienen als Ausgangsanordnung für den nächsten Schritt mit neuen Empfangsmeldungen.

Nach jeder Positions- oder Orientierungsänderung der Nametags nähert sich dieses Verfahren nach einer Phase des Einschwingens den gesuchten Werten für Positionierung und Orientierung an. Die Genauigkeit des Verfahrens verbessert sich dabei mit steigender Anzahl der Nametags im System. Auf diese Weise lassen sich zufällige Fehler zuverlässig reduzieren. Da allerdings nur mit der recht groben Information über die relative Nähe zweier Nametags zueinander gearbeitet wird, sind die ermittelten Positionen und Orientierungen für kleinere Nametag-Anzahlen mit einem spürbaren Fehler behaftet.

### 3.2.2 Ultraschall

In einem Lokationssystem auf Ultraschall-Basis werden Schallsignale im Ultraschall-Bereich ausgesendet und die Ausbreitungszeit und/oder die Intensität der Schallwellen bestimmt. Ein Beispiel für ein System auf Ultraschall-Basis sind die BRICK-Sensorknoten (siehe [Beus07]), die jeweils über vier kombinierte Ultraschallsender und -empfänger verfügen. Sie ermöglichen eine Entfernungsmessung über das Bestimmen der Ausbreitungszeit von Ultraschallsignalen.

Dazu wird mittels eines zusätzlichen Funksignals, dessen Ausstrahlung anderen Knoten im Empfangsbereich als Zeitreferenz dient, die Übertragung eines Ultraschallimpulses angekündigt.

Die Ausbreitungsverzögerung des Funksignals lässt sich aufgrund des großen Unterschiedes in der Ausbreitungsgeschwindigkeit zwischen Schallwellen und Funkwellen vernachlässigen. Jeder Knoten ermittelt nun die Differenz zwischen dem Eintreffen des Funksignals und dem Eintreffen des Ultraschallsignals für jeden der vier Ultraschallempfänger.

Mit dem Wissen über die Umweltbedingungen, die die Geschwindigkeit des Schalles während der Übertragung beeinflussen, lässt sich aus der Ausbreitungszeit die Entfernung zweier Systeme zueinander bestimmen. Die größten Einflussfaktoren sind dabei vor allem die Temperatur und die Luftfeuchtigkeit. Durch die parallele Messung der Entfernungen mit mehreren Empfängern lässt sich mittels der Hyperbelnavigation eine relative Positionsangabe ermitteln.

Neben einer Entfernungsmessung wird auch eine Winkelmessung durchgeführt. Dazu wird die Signalstärke der zur Entfernungsberechnung ausgesendeten Schallwellen gemessen und für jeden Empfänger eine Validitätsprüfung durchgeführt. Jeder Empfänger steht für einen bestimmten Winkel, wobei sich die Winkel entsprechend der Positionen der Empfänger um 90 Grad unterscheiden. Zunächst wird der Winkel durch den Ultraschallempfänger bestimmt, der die kürzeste Signallaufzeit gemessen hat. Gibt es weitere gültige Signalstärken, so wird zwischen diesen interpoliert um den Winkel weiter einzugrenzen.

Die BRICK-Sensorknoten wurden als Hardwareplattform für diese Studienarbeit ausgewählt, da die Ultraschallsignale eine recht genaue Entfernungsbestimmung mit geringen Schwankungen erlauben, ohne dass für den Einsatz in den Räumen eine aufwendige Infrastruktur installiert werden muss. Ein Vorteil der BRICK-Knoten ist

auch die Kombination aus Laufzeitmessungen und Signalintensität, so dass sich mit ihnen die Eignung der Qualitätsanalyse für die wichtigsten Typen der Lokationssysteme zeigen lässt. Auch das Auftreten systematischer Fehler, bei Mehrwegeempfang, unter verschiedenen Temperatureinflüssen und bei unterschiedlicher Ausrichtung der Knoten zueinander, liefert Möglichkeiten für eine Qualitätsanalyse die über das Mitteln zufälliger Fehler hinausgeht.



# 4. Zustandsvorhersage, Fehlerabschätzung und Fehlermodelle

In Kapitel 2 wurde eine Reihe von möglichen Fehlerquellen in Lokationssystemen genannt. Um diesen Fehlern zu begegnen wurden im Laufe der Jahre einige Verfahren zur Zustandsvorhersage entwickelt. Dabei gibt es verschiedene Ansätze eine möglichst exakte Lösung in möglichst vielen Fällen zu erhalten. Manche der Verfahren ermöglichen auch eine Aussage über die Qualität, also den Grad der Fehlerminimierung, der ermittelten Zustände.

Einige der Verfahren werden in diesem Kapitel kurz vorgestellt. Dabei werden auch die jeweils möglichen Qualitätseinschätzungen betrachtet und diese mit den Anforderungen an das in dieser Arbeit vorgestellte System verglichen. Dazu gehört beispielsweise eine explizite Qualitätseinschätzung für einzelne Messwerte und eine einfache Anpassung an veränderte Problemstellungen.

## 4.1 Kalman-Filter

Eine breite Anwendung haben Systeme gefunden, die Kalman-Filter einsetzen. Vorgestellt wurden diese 1960 von Rudolf Kálmán in [Kalm60]. Für den praktischen Einsatz hat es seither verschiedene Erweiterungen gegeben. Ein Kalman-Filter ist ein stochastischer Zustandsschätzer und liefert in einem dynamischen System eine iterative Schätzung von Zuständen oder Parametern.

Es gibt zwei sich abwechselnde Schritte. Zum einen die Prädiktion, in der ausgehend vom vorherigen Zustand mittels einer Zustandsübergangsmatrix und, falls verfügbar, unter Nutzung eines Regeleingriffes ein neuer Zustand prädiziert wird. Aus der Zustandsübergangsmatrix, einem Modellfehler und der Fehlerkovarianzmatrix des vorherigen Zustandes wird dann eine Fehlerkovarianzmatrix des aktuellen Zustandes bestimmt.

Zum anderen gibt es die Korrektur, in der die Prädiktion mit einer aktuellen Messung verglichen und zu einer neuen Schätzung zusammengefasst wird. Die Gewichtung der Messung gegenüber der Prädiktion hängt dabei von der Prädiktionsfehlerkovarianzmatrix und einer Messfehlerkovarianzmatrix ab. Schließlich wird aus der vorherigen eine neue Fehlerkovarianzmatrix bestimmt, die zusammen mit der

Schätzung als Eingabe für eine erneute Prädiktionsphase dient. Für eine detaillierte Beschreibung siehe [WeBi01].

Eine Möglichkeit zur Qualitätseinschätzung bietet ein Kalman-Filter nicht. Es wird in der Modellierung nur das Prozessrauschen erfasst, systematische Fehler lassen sich nicht erkennen und beseitigen. Die Qualität der Schätzung hängt davon ab, dass passende Parameter für das Modell und die Zustandsübergänge gewählt wurden. Diese Parameter müssen für eine neue Problemstellung speziell angepasst werden.

## 4.2 Partikel-Filter

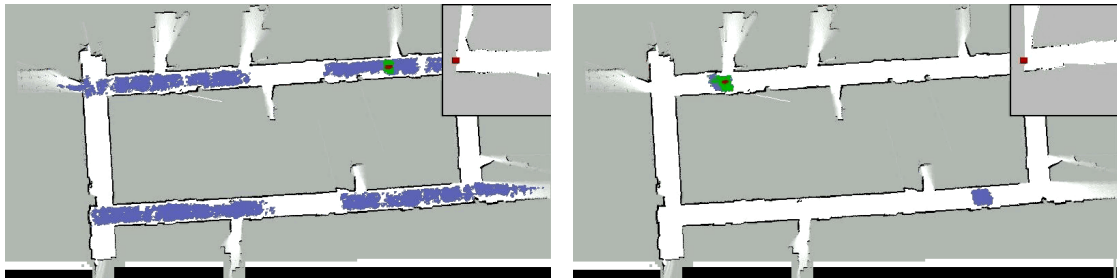


Abbildung 4.1: Einsatz von Partikel-Filtern zur Lokationsbestimmung eines Roboters am USC Robotics Research Lab. Links: kurz nach der Initialisierung, rechts: nach einigen Durchläufen. (Quelle: [http://robotics.usc.edu/~howard/projects\\_wifi.php](http://robotics.usc.edu/~howard/projects_wifi.php))

Weitere stochastische Zustandsschätzer sind Partikel-Filter die wie Kalman-Filter auch iterativ arbeiten, aber flexibler einzusetzen sind als diese, da sie in der Approximation beliebige Dichten ermöglichen. Allerdings ist der Rechenaufwand, auch abhängig von der zur Problemlösung benötigten Partikel Menge, zum Teil deutlich größer als bei den Kalman-Filtern.

Ein Partikel-Filter besteht aus zwei Gleichungssystemen. Das Erste modelliert die Zustandsübergänge. Wie der Kalman-Filter enthält es noch für jeden Übergang ein zufälliges Rauschen, dessen Intensität und Verteilung man von der erwarteten Änderung im System abhängig machen kann. Das zweite Gleichungssystem dient dazu die Wahrscheinlichkeiten der ermittelten Partikel zu bestimmen, abhängig von der erwarteten Fehlerverteilung des Systems, siehe dazu [MuMR06].

Ein Überblick über die verschiedenen stochastischen Verfahren zur Zustandsschätzung gibt es in [Krum03]. Kalman- und Partikel-Filtern ist gemeinsam, dass sie darauf abzielen, immer ein möglichst optimales Ergebnis zurückzuliefern. Dies gilt auch wenn ein Ergebnis von schlechter Qualität ist. Ein Ergebnis das vom tatsächlichen gesuchten Wert stark abweicht lässt sich dabei kaum oder gar nicht von einem Ergebnis unterscheiden, das nahe am gesuchten Wert liegt. Wie beim Kalman-Filter wird nur das Prozessrauschen modelliert und es ist nicht möglich systematische Fehler zu erkennen und zu beseitigen. Einzig die Verteilung der Partikel lässt beim Partikel-Filter einen ungefähren Rückschluss zu, ob ein Ergebnis verlässlich ist oder nicht. Möchte man einen Partikel-Filter für eine neue Problemstellung anpassen, so muss man ähnlich wie beim Kalman-Filter die Zustandsübergänge neu modellieren und die passende Fehlerverteilung bestimmen.

## 4.3 Distributed Precision Based Localization

Beim Distributed Precision Based Localization Algorithm handelt es sich um ein verteiltes Verfahren von Evers et al. zur Lokationsbestimmung das in [EvDH04]

vorgestellt wurde. Verwendung findet die gewichtete Methode der kleinsten Fehlerquadrate um die Position eines Sensorknotens aus zu anderen Knoten gemessenen Entfernungen zu bestimmen. Das Verfahren arbeitet dabei iterativ, um die zuletzt abgeschätzten Positionen anderer Knoten im nächsten Schritt zur Verbesserung der Abschätzung einzubeziehen.

Als Gewichte für die einzelnen Messungen dient eine Qualitätseinschätzung, die aus einer Gaussverteilung von Messungen um ihren korrekten Wert gewonnen wird. Eine gaussverteilte Zufallsvariable kennzeichnet dabei die Wahrscheinlichkeit, dass sich die korrekte Position des Knotens in einer bestimmten Reichweite zur ermittelten Position befindet. Ausgedrückt wird dies durch die Standardabweichung der Fehlerverteilung. Eine solche hängt von der zugrundeliegenden Hardware ab und wird vor dem Einsatz eines Systems einmalig festgelegt.

Problematisch ist dabei die Annahme eines generellen Modelles für die Fehlerprognose. Dieses Modell muss abhängig vom Einsatzort des Systems erstellt werden, da sich die Fehlerverteilung beim Einsatz an verschiedenen Orten unterscheiden kann. Ein solches generelles Modell liefert aber nur eine identische Qualitätseinschätzung für alle Messungen eines Systems, beispielsweise eines Sensorknotens. Eine Qualitätseinschätzung einer einzelnen Messungen ist mit diesem Ansatz nicht möglich und es lassen sich auch nur zufällige Fehler und solche systematischen Fehler modellieren, die beim Ermitteln der Fehlerverteilung bereits vorlagen.

## 4.4 Sensorvalidierung mittels neuronaler Netze

Mattern et al. stellen in [MJGG<sup>+</sup>98] ein für die NASA entwickeltes Verfahren vor, das neuronale Netze verwendet, um aus einer Menge redundanter Sensoren einen fehlerhaften Sensor zu erkennen. Eine Sensorvalidierung lässt sich in drei Schritte aufteilen. Zunächst das Feststellen, dass eine Messung nicht korrekt ist. Dann die Eingrenzung des Problemursprungs und abschließend die Anpassung an das Problem.

Verwendet wird ein Auto-Associative Neural Network (AANN) das bei Ausfall eines Sensors aus den Werten der übrigen Sensoren eine Abschätzung Messwerte vornimmt. Es handelt sich um ein Feed-Forward Netz das aus zwei durch eine Engpassschicht verbundenen Abbildungsschichten besteht. Die erste Schicht verringert die Dimension der Daten durch das Entfernen redundanter Merkmale und das Extrahieren der Hauptkomponenten bzw. der Schlüsselmerkmale der Daten. In der Abbildungsschicht nach der Engpassschicht wird aus den kodierten Hauptkomponenten die komplette Information wiederhergestellt. Die Ausgabeschicht bildet die Eingabeschicht nach.

In einem ersten Schritt wird das neuronale Netz mit gültigen Daten trainiert. Die Größe der Engpassschicht ist dabei so anzupassen, dass die gewünschte Redundanzeliminierung ermöglicht wird. In einem zweiten Schritt wird mit einem Datensatz trainiert der auch fehlerhafte Informationen enthält, um dem neuronalen Netz das Filtern von fehlerhaften Werten anzutrainieren.

Der Fokus des Verfahrens liegt nicht auf einer Qualitätsanalyse, wie sie in dieser Arbeit angestrebt wird, sondern auf dem Erkennen falscher Sensorwerte und dem Umgehen des Problems durch das neuronale Netz.

## 4.5 Kontextklassifikation mittels ANFIS-Methoden

Die Grundlage für diese Studienarbeit bildet das in [BDRZ<sup>+</sup>07] vorgestellte System, das eine Qualitätsanalyse von bestehenden Systemen zur Erkennung von kon-

textuellen Zuständen ermöglicht. Das System beruht auf einer unscharfen Logik, deren Grad der Zugehörigkeit ein Maß der Qualität repräsentiert.

Zur qualitativen Analyse der kontextuellen Werte, die durch einen beliebigen Algorithmus zur Bestimmung von kontextuellen Zuständen berechnet werden, wird ein unscharfes Regelsystem eingesetzt.

Dieses unscharfe Regelsystem ist vom Typ der TSK-Regelsysteme, die von Takagi, Sugeno und Kang in [TaSu83] und [SuKa88] entwickelt wurden. Die Analyse der berechneten kontextuellen Zustände erfolgt auf Basis der Eingabewerte, die auch zur Bestimmung der kontextuellen Zustände benutzt werden, und des Identifikators, der den kontextuellen Zustand identifiziert.

Zur Spezifizierung des Sugeno-Systems wird mit ANFIS eine von Roger Jang in [Jang93] vorgestellte Methode verwendet, die die Parameter, die in einer Sugeno-Fuzzy-Logik verwendet werden, aufstellt und durch Training eines korrespondierenden neuronalen Netzes adaptiert. Für Detaillierte Ausführungen zu ANFIS siehe 5. An einem Stift-Artefakt, das eine Klassifikation für die Zustände „liegend“, „schreibend“ und „bewegt“ durchführt, wurde das vorgestellte System getestet. Es zeigte sich, dass sich durch die Qualitätsanalyse die richtig von den falsch klassifizierten Kontexten mit hohen Wahrscheinlichkeiten trennen lassen.



# 5. Architektur des Systems und Systemidentifikation

In Kapitel 4 wurden bestehende Verfahren zur Zustandsvorhersage in Lokationssystemen betrachtet. Es zeigte sich, dass die damit möglichen Qualitätseinschätzungen nicht die Anforderungen eines dynamischen Fehlermodelles erfüllen, sondern nur eine allgemeine Aussage über die Qualität einer Lokationsbestimmung liefern können. Für diese Arbeit soll nun im Folgenden ein System vorgestellt werden, das es ermöglicht dynamisch für jede Entfernungs- oder Winkelmessung eines Sensor-knotens eine Qualitätsaussage zu treffen. Ziel ist es dabei nicht nur mit zufälligen sondern gerade auch mit systematischen Fehlern umgehen zu können.

## 5.1 Architektur

Ein zentraler Designaspekt der Systemarchitektur ist es, die Qualitätsanalyse unabhängig vom zu bewertenden Algorithmus, in dieser Arbeit dem Algorithmus zur Lokationsbestimmung, zu gestalten. Dadurch ist eine einfache Übertragbarkeit auf unterschiedliche Systeme möglich. In diese Systeme wird nicht direkt eingegriffen und es ist auch nicht nötig und sinnvoll, dass diese Systeme an die Qualitätsanalyse angepasst werden.

Für das zu bewertende System erscheint die Qualitätsanalyse als eine „Blackbox“, genauso erscheint auch umgekehrt das zu bewertende System der Qualitätsanalyse als eine „Blackbox“. Beide lassen sich unabhängig voneinander entwickeln, verbessern und an neue Gegebenheiten anpassen. Eine enge Verzahnung der Lokationsbestimmung und Qualitätsanalyse würde dies erschweren.

Ein wichtiges Designziel ist es die Qualitätsanalyse ohne Expertenwissen über das zu bewertende System durchführen zu können, um zum einen den Einsatz des Systems zu erleichtern und um zum anderen unabhängig von den konkreten Algorithmen im beurteilten System zu bleiben.

Ein weiterer Designaspekt ist es keine generellen (statistischen) Qualitätsaussagen zu machen, wie es in einem statischen Fehlermodell der Fall ist, sondern die Qualitätsanalyse nach einem dynamischen Fehlermodell für jede einzelne Messung durchzuführen um eine umfassendere und flexiblere Qualitätsaussage zu erhalten, die neben zufälligen auch mit systematischen Fehlern umgehen kann.

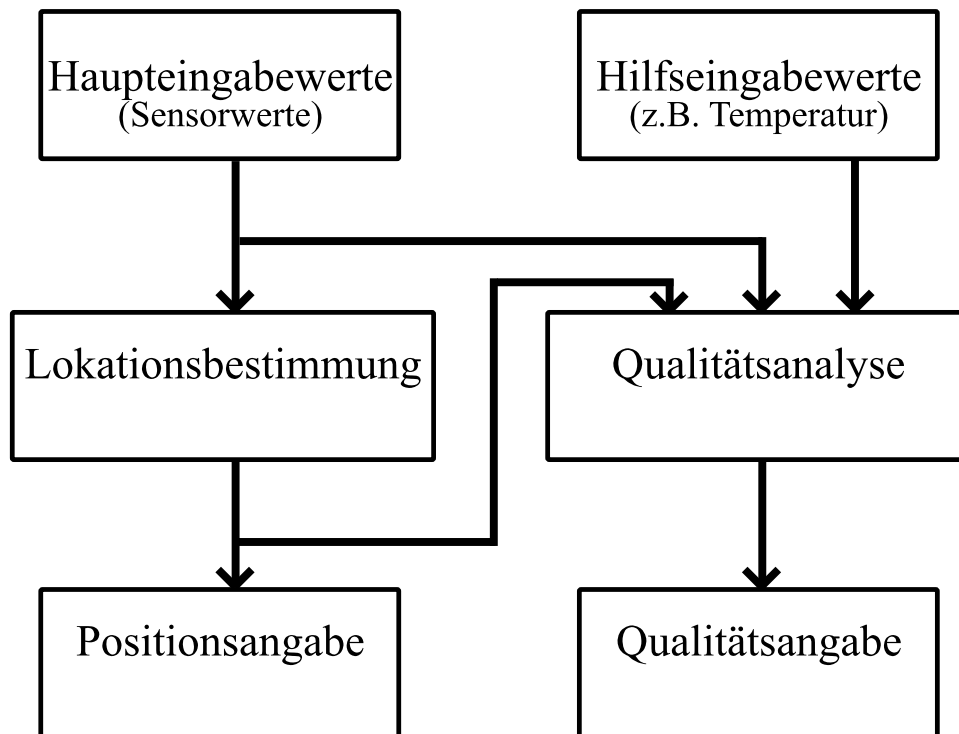


Abbildung 5.1: Systemarchitektur

Da das System auch auf Sensorknoten mit geringer Rechenleistung eingesetzt werden soll ist es notwendig das System so zu gestalten, dass die Qualitätsanalyse für die einzelnen Messungen keine hohe Performanz erfordert. Dies wird durch eine Aufteilung in eine aufwendige Offline-Systemidentifikation und eine Online-Qualitätsanalyse mit geringem Rechenaufwand erreicht.

Die Qualitätsanalyse erhält die gleichen Eingabewerte die auch das zu bewertende System verwendet, da diese die Grundlage für die im bewerteten System eingesetzten Algorithmen bilden. Zusätzlich lassen sich weitere Hilfeingabewerte verwenden, die möglichst alle erfassbaren Rahmenbedingungen abdecken sollten, die einen Einfluss auf die Qualität der Systemausgaben haben können.

Zuletzt wird noch die Ausgabe des zu bewertenden Systems als Eingabe in die Qualitätsanalyse eingebracht, da es sich dabei um die Aussage des zu bewertenden Systems handelt, die darüber bestimmt, ob das System das Ziel der Fehlerminimierung erreicht hat.

## 5.2 Systemidentifikation

In dieser Arbeit wurde die Systemidentifikation mittels ANFIS-Methoden durchgeführt, da diese in der Lage sind die gewünschten Designaspekte des Systems zu erfüllen. Dazu gehören Takagi-Sugeno-Kang (TSK) Regelsysteme, die es ermöglichen für jede einzelne durchgeführte Messung eine Qualitätsabschätzung durchzuführen. Dabei lässt sich das Verfahren unabhängig von der konkreten Implementierung des zu beurteilenden Systems und ohne Expertenwissen verwenden und lernt durch das Training mit Beispielmessungen aus den anfänglichen Fehlern in der Qualitätsbeurteilung.

ANFIS ist die Abkürzung für das von Roger Jang [Jang93] entwickelte Adaptive-  
Network-Based Fuzzy Inference System. Es handelt sich um eine Fuzzy-Logik, die

auch als neuronales Netz interpretiert wird. Durch das Trainieren als neuronales Netz werden dabei die Parameter der Fuzzy-Logik angepasst und für eine gegebene Problemstellung optimiert. Eine detaillierte Darstellung des ANFIS findet sich in [Berc07] und [BDRZ<sup>+</sup>07].

### 5.2.1 Clustering

Eine Problematik für den praktischen Einsatz der Fuzzy Logik besteht darin, die Regeln zu bestimmen, die benötigt werden, um eine gute Abdeckung der Eingabedaten zu erreichen und eine passende Ausgabe zu liefern. Mit unendlich vielen unscharfen Regeln lässt sich eine infinitesimal genaue Approximation erreichen, aber für eine eingeschränkte Regelanzahl, die sich praktisch in einem System verwenden lässt, muss eine Möglichkeit gefunden werden, die am besten passenden Regeln ausfindig zu machen. Ausprobieren scheitert an der Vielzahl der möglichen Kombinationen. Für komplexe Eingabedaten ist es einem Menschen auch kaum mehr möglich sinnvolle Regeln zu erkennen. Deswegen ist es wünschenswert, die Regeln vollständig automatisiert aufstellen zu lassen.

Eine mögliche Lösung zur Bestimmung geeigneter Regeln ist ein vorgeschaltetes Clustering. Damit lassen sich zum einen die Regeln automatisiert aufstellen und gleichzeitig kann auch die Komplexität des resultierenden Fuzzy-Inferenz-Systems (FIS) verringert werden. Für das Clustering gibt es eine Reihe von Algorithmen die allerdings zumeist mit einigen Nachteilen für den Einsatz mit einem FIS behaftet sind. Beispielhaft sei hier k-Means Clustering [MacQ76] genannt. Bei diesem recht schnell konvergierenden Verfahren werden scharf voneinander abgegrenzte Cluster gebildet. Für die unscharfe Logik ist diese Eigenschaft aber gerade unerwünscht, da hier die Übergänge zwischen den Mengen fließend ausfallen. Hinzu kommt beim k-Means Clustering die Notwendigkeit für das Initialisieren des Algorithmus die Anzahl der Cluster,  $k$ , zu bestimmen.

#### 5.2.1.1 Subtraktives Clustering

Ein zur automatischen Erstellung gut geeignetes Verfahren ist das subtraktive Clustering [Chiu94]. Dieses berechnet die Anzahl der Cluster selbstständig, liefert unscharfe Cluster und benötigt auch nicht wie manche andere Verfahren eine möglichst günstige Gitterauflösung. Jeder Vektor des Datensatzes dient als potentielles Cluster-Zentrum. Für alle Vektoren wird das Potential bestimmt, also die Distanz zu jedem anderen Vektor des Datensatzes. Es folgt ein iteratives Vorgehen, wobei das gewichtete Potential des zuletzt bestimmten Cluster-Zentrums abgezogen und für die verbleibenden Potentiale ein neues Cluster-Zentrum gesucht wird. Größter Vorteil des subtraktiven Clusterings ist die selbstständige Arbeitsweise, die es ermöglicht das Clustering als unüberwachte Vorstufe der FIS-Erstellung einzusetzen.

Aus den Clustern ergeben sich für die Eingabe  $\vec{v}$  die initialen Zugehörigkeitsfunktionen  $F_{ij}(\vec{v})$  mit den Parametern  $\sigma_{ij}^2$  und  $\mu_{ij}$ , die nichtlinearen Antezedenzen  $w_j$  und aus der Anzahl der Cluster die Anzahl  $o$  der implikativen Regeln des Sugeno-Regelsystems  $\hat{S}$ .

Beim subtraktiven Clustering ist zu beachten, dass die Gaussfunktionen beim Clustering nur für solche Werte ein Cluster-Zentrum finden können, die bereits im Datensatz vorhanden sind. Enthält dieser Datensatz Lücken, so werden bei der Auswertung des mittels Clustering erzeugten FIS diejenigen Werte, die in diesem Bereich liegen, bei der Abbildung auf das Zielintervall  $[0..1]$  immer auf Null abgebildet, siehe zur Veranschaulichung Abbildung 5.2. Das FIS liefert also für „unbekannte“ Werte

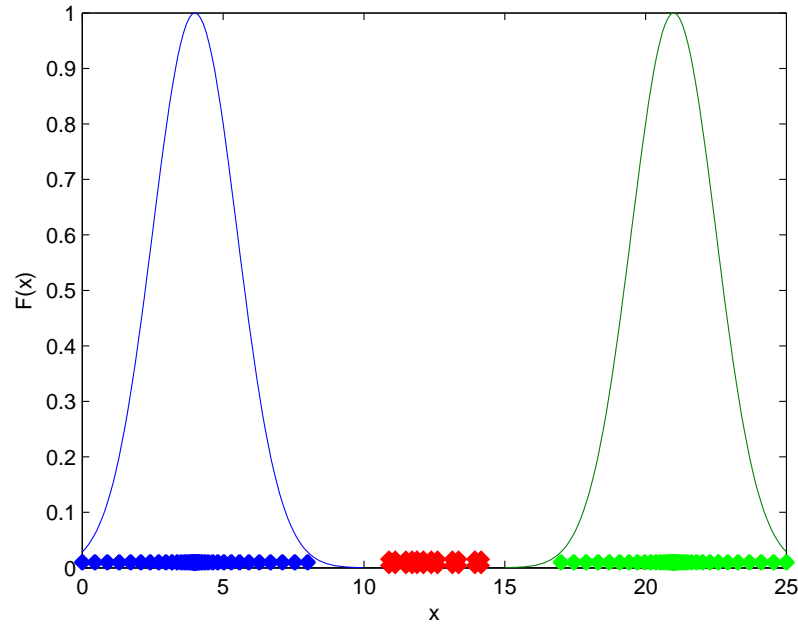


Abbildung 5.2: Beim Durchführen des Clusterings nicht erfasste Datensätze zwischen zwei Cluster-Zentren.

eine definierte Ausgabe. Dennoch sollte man für das Clustering bedenken, dass diskrete Eingabewerte eine vollständige Abdeckung erschweren, da es dabei zu den oben beschriebenen Lücken im Datensatz kommen kann.

### 5.2.2 Methode der kleinsten Fehlerquadrate

Die Methode der kleinsten Fehlerquadrate versucht in einen Datensatz eine Funktion so einzupassen, dass die Summe der quadratischen Fehler, also der Abstände der Funktion zu den einzelnen Daten, minimal wird. Die Datensätze sind dabei die dem System vorgegebenen gesuchten Ausgabewerte. Die Abstände werden über ein lineares Gleichungssystem formuliert, dessen Lösung das Resultat der Methode ist. Da das resultierende Gleichungssystem überbestimmt ist, erfordert es eine numerische Lösung.

Beim Sugeno-System  $\hat{S}$  werden mittels der Methode der kleinsten Fehlerquadrate die linearen Konsequenzen  $f_j$ , in Abhängigkeit von den nichtlinearen Antezedenzien  $w_j$ , an die Eingabedaten  $\vec{v}$  angepasst. Für die linearen Konstanten  $a_{ij}$  ergeben sich die linearen Konsequenzen als

$$f_j(\vec{v}) := a_{1j}v_1 + a_{2j}v_2 + \dots + a_{nj}v_n + a_{(n+1)j}.$$

Da

$$\hat{S}(\vec{v}) = \frac{\sum_{j=1}^o w_j(\vec{v})f_j(\vec{v})}{\sum_{j=1}^o w_j(\vec{v})}$$

und die nichtlinearen Antezedenzien  $w_j$  aus dem Clustering bekannt sind, lassen sich unter Verwendung der Eingabedaten die linearen Konsequenzen  $f_j$  als überbestimmtes lineares Gleichungssystem schreiben und numerisch lösen um die initialen linearen Konstanten  $a_{ij}$  zu erhalten. Mit den initialen Zugehörigkeitsfunktionen  $F_{ij}(\vec{v})$ , den linearen Konsequenzen  $f_j(\vec{v})$  und Regeln  $j$  lässt sich nun das Sugeno-System  $\hat{S}$  konstruieren.

### 5.2.3 ANFIS

Im ANFIS wird nun die Methodik eines Fuzzy-Inferenz-Systems mit dem Trainieren eines neuronalen Netzes verknüpft. Für Takagi-Sugeno-Kang (TSK) Regelsysteme ist ein ANFIS dabei die funktional identische Repräsentation eines FIS (siehe dazu [Berc07]). Nachdem durch subtraktives Clustering die Regeln und initialen Zugehörigkeitsfunktionen  $F_{ij}(v_i)$  bestimmt und durch die Methode der kleinsten Fehlerquadrate die initialen linearen Konsequenzen  $f_j$  festgelegt wurden lässt sich daraus nun das Sugeno-System  $\hat{S}$  generieren und als neuronales Netz interpretieren.

Das resultierende neuronale Netz besteht aus mehreren Schichten. Abbildung 5.3 veranschaulicht den Aufbau des neuronalen Netzes.

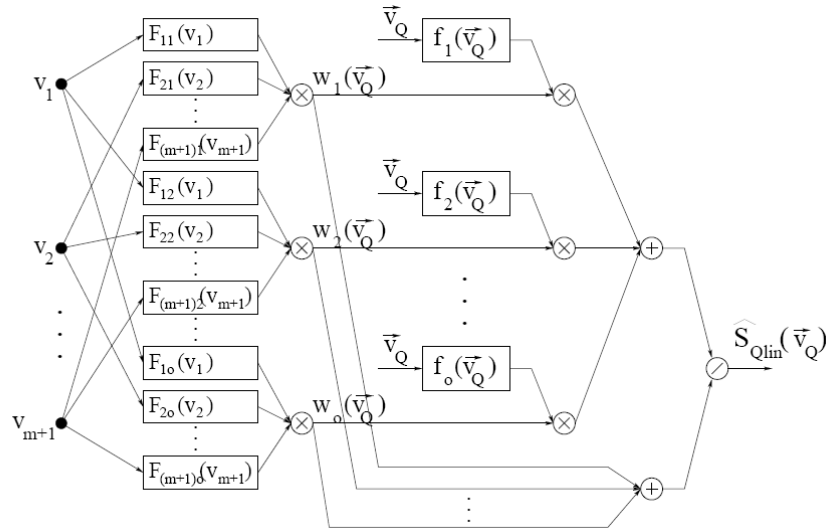


Abbildung 5.3: TSK-FIS repräsentiert durch ein neuronales Fuzzy-Netz (Quelle: [Berc07]).

In der ersten Schicht wird der Vektor  $\vec{v}$  in seine Elemente aufgeteilt und an die einzelnen Neuronen übergeben, welche die Elemente an die Zugehörigkeitsfunktionen in der zweiten Schicht übergeben. In der dritten Schicht werden die Funktionswerte der Zugehörigkeitsfunktionen multipliziert. Dies ergibt die Gewichte  $w_j$ . Diese werden einmal direkt an die fünfte Schicht weitergeleitet und einmal an die vierte Schicht.

In der vierten Schicht wird aus den Gewichten  $w_j$  und den linearen Konsequenzen  $f_j$  die Multiplikation  $w_j(\vec{v})f_j(\vec{v})$  durchgeführt. Die fünfte Schicht besteht aus zwei Neuronen. Das eine bildet die gewichtete Summe  $\sum_{j=1}^o w_j(\vec{v})f_j(\vec{v})$ , das andere, dessen Werte direkt aus der dritten Schicht übergeben wurden, die Summe  $\sum_{j=1}^o w_j(\vec{v})$ .

Für die Ausgabe wird im Neuron in der letzten Schicht eine Division der beiden Summen durchgeführt.

Für die Parameter der Zugehörigkeitsfunktionen  $(\sigma_{ij}^2, \mu_{ij})$  und die linearen Konstanten  $a_{ij}$  gilt es nun eine Feinabstimmung vorzunehmen. Dazu wird das Sugeno-System  $\hat{S}$  als neuronales Netz interpretiert und mit der Methode des Gradientenabstiegs trainiert. Dies kann zum einen durch Backpropagation gemacht werden, also das Korrigieren der Gewichte der Neuronen ausgehend von den Fehlern in den Ausgabewerten bis in die vorderste Neuronenschicht, zum anderen durch hybrides Lernen.

### 5.2.4 Hybrides Lernen

Hybrides Lernen ist eine Kombination aus Backpropagation und der Methode der kleinsten Fehlerquadrate. In der Rückwärtsrichtung wird mittels Backpropagation der Fehler zwischen gewünschter und tatsächlicher Ausgabe durch die Neuronenschichten nach vorne übertragen um dort die Parameter  $(\sigma_{ij}^2, \mu_{ij})$  der nichtlinearen Zugehörigkeitsfunktionen  $F_{ij}$  anzupassen. Durch den Gradientenabstieg wird dazu ein globales Minimum für den Fehler auf einer Fehlerfläche für die Parameter  $\sigma_{ij}^2$  und  $\mu_{ij}$  bestimmt.

Mit den neu bestimmten nichtlinearen Zugehörigkeitsfunktionen  $F_{ij}$  wird in der Vorwärtsrichtung ein weiterer Durchlauf der Methode der kleinsten Fehlerquadrate vorgenommen und die linearen Konstanten  $a_{ij}$  ausgehend von den neuen  $F_{ij}$  bestimmt.

Das hybride Lernen stoppt nach einer festgelegten Epochenanzahl oder wenn der Fehler eines Checking-Datensatzes ansteigt. Das Ergebnis ist ein TSK-FIS, das sich für eine schnelle Online-Auswertung eignet.

## 5.3 Online-System zur Qualitätsanalyse

### 5.3.1 Fuzzy-Inferenz-System

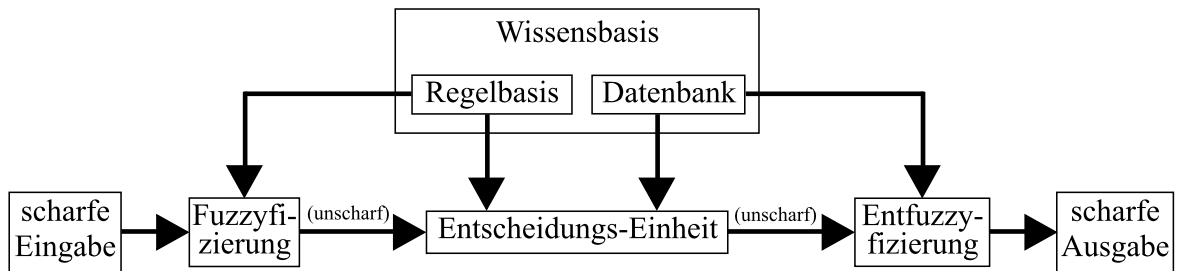


Abbildung 5.4: Schematischer Aufbau eines Fuzzy-Inferenz-Systems.

Ein Fuzzy-Inferenz-System (FIS) besteht aus den folgenden Blöcken:

- Eine Schnittstelle, die klare, eindeutige Eingabewerte in den Grad ihrer Übereinstimmung mit den unscharfen Mengen überführt (Fuzzyfizierung).
- Eine Regelbasis, die eine Reihe von unscharfen Wenn-Dann-Regeln enthält.
- Eine Datenbank, die die Zugehörigkeitsfunktionen für die unscharfen Mengen in den Fuzzy-Regeln definiert.
- Eine Entscheidungs-Einheit, die auf den Regeln die Deduktion durchführt.
- Eine Schnittstelle, die unscharfe Ergebnisse in klare, eindeutige Ausgabewerte überführt (Entfuzzyfizierung).

Bei den unscharfen Wenn-Dann-Regeln handelt es sich um Ausdrücke der Form *WENN A DANN B*, wobei A und B unscharfe Mengen bezeichnen, die durch ihre Zugehörigkeitsfunktionen charakterisiert werden. Für jedes Element einer unscharfen Menge lässt sich der Grad seiner Zugehörigkeit zu dieser Menge angeben. Die verbreitetste Zugehörigkeitsfunktion ist die Gaussfunktion.

Abbildung 5.4 zeigt den Ablauf eines FIS von der scharfen Eingabe über unscharfe Zwischenwerte zur scharfen Ausgabe. Zunächst werden die scharfen Eingabewerte durch die Fuzzifizierung unter Verwendung der Zugehörigkeitsfunktionen aus der Datenbank in unscharfe Mengen überführt. In der Entscheidungs-Einheit werden unter Nutzung von Regelbasis und Datenbank die regelbasierten Ableitungen der unscharfen Werte durchgeführt. Die resultierende funktionalen Zugehörigkeit wird anschließend in der Entfuzzifizierung auf einen reellen Wert abgebildet.

### 5.3.1.1 TSK-FIS

Eine etwas modifizierte Form der Wenn-Dann-Regeln, die sich besonders für die automatische Generierung eignet, wurde von Takagi und Sugeno in [TaSu83] bzw. Sugeno und Kang in [SuKa88] vorgestellt. Dabei enthält nur die Prämisse unscharfe Mengen. Ein Beispiel dafür ist ein Ausdruck der Form *WENN Geschwindigkeit groß DANN Kraft = k \* Geschwindigkeit<sup>2</sup>*. Die unscharfe Menge *groß* wird charakterisiert durch eine passende Zugehörigkeitsfunktion. Als Implikation wird eine nicht-lineare Gauss-Funktion verwendet.

Beim mittels ANFIS-Methoden erstellten TSK-FIS handelt es sich bei der Prämisse um die Zugehörigkeitsfunktionen  $F_{ij}(v_i)$  und bei der Implikation um die linearen Konsequenzen  $f_j(v_i)$ . Und das Sugeno-System  $\hat{S}$  lässt sich schreiben als

$$\hat{S}(\vec{v}) = \frac{\sum_{j=1}^o w_j(\vec{v}) f_j(\vec{v})}{\sum_{j=1}^o w_j(\vec{v})}$$

Beim TSK-FIS wird die Projektion von der Eingabe auf die Ausgabe durch einen gewichteten Summendurchschnitt vorgenommen. Dabei handelt es sich um eine Kombination aus unscharfer Schlussfolgerung und Entfuzzifizierung, die aus den Zugehörigkeitsfunktionen und den linearen Konsequenzen die Ausgabe bestimmt.

### 5.3.1.2 Semantische Normalisierung

Da sich bei einem TSK-FIS die Grenzen einer Abbildung nicht bestimmen lassen benötigt man noch eine Normalisierungsfunktion, um den Wertebereich auf das gewünschte Zielintervall  $[0..1]$  einzuschränken. Werte in der Nähe von 0 und 1, sowie dazwischen sollen dabei gültig sein, alle übrigen Werte sollen auf einen Fehlerzustand  $e$  abgebildet werden. Eine solche Normalisierungsfunktion  $L$  kann für die Eingabe  $x = \hat{S}(\vec{v})$  folgende Gestalt haben:

$$L(x) := \begin{cases} |x| & \text{wenn } -0,5 \leq x < 0 \\ x & \text{wenn } 0 \leq x \leq 1 \\ 2 - x & \text{wenn } 1 < x \leq 1,5 \\ e & \text{sonst.} \end{cases}$$

Ein ANFIS-System lässt sich wie jedes neuronale Netz auf bestimmte Anforderungen hin trainieren. Es ist dabei möglich eine komplexe Funktion zu approximieren die zu Eingabewerten eine gewünschte Ausgabe liefert. Wird für jede Gruppe an Eingabewerten der gewünschte Ausgabewert so festgelegt, dass er entweder ein richtiges oder ein falsches Ergebnis repräsentiert und das ANFIS-System auf diese Zuordnung hin trainiert, so lässt sich die Ausgabe des TSK-FIS als Qualitätswert interpretieren.

In Kapitel 4 wurde gezeigt, dass bestehende Verfahren zur Zustandsvorhersage und Fehlerabschätzung in relativen Lokationssystemen keine umfassende Qualitätsbeurteilung liefern können. Durch den Einsatz eines mittels ANFIS-Methoden aufgestellten und dann als neuronales Netz trainierten TSK-FIS dagegen lässt sich eine

solche Qualitätsbeurteilung individuell für einzelne Messwerte durchführen, wobei der Rechenzeitbedarf vor allem im Offline-Bereich der Systemidentifikation anfällt und die spätere Online-Auswertung sich als schlankes, schnelles System realisieren lässt.



## 6. Versuchsaufbau

Um das in Kapitel 5 beschriebene ANFIS zu trainieren und zu testen werden Beispielmessungen benötigt. In diesem Kapitel wird die verwendete Hardware zur Sammlung von Entfernungs- und Winkelmesswerten in der Praxis beschrieben und der Versuchsaufbau zur Gewinnung von Messwerten für das Trainieren des ANFIS dargestellt.

### 6.1 Verwendete Hardware

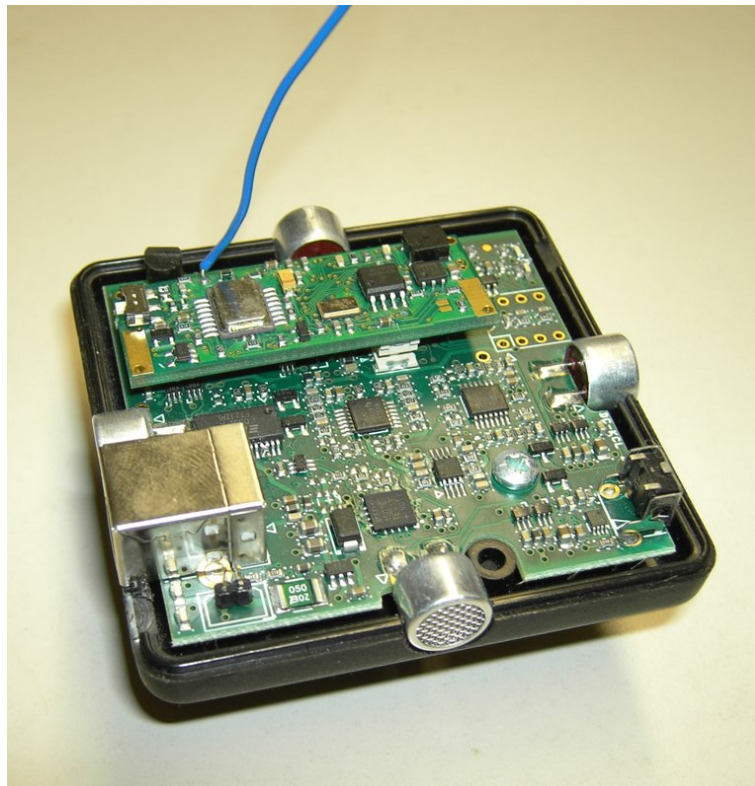


Abbildung 6.1: BRICK Sensorknoten mit aufgestecktem Particle zur Kommunikation mit dem PC.

Als Hardwareplattform zum Sammeln von Ultraschall Entfernungs- und Winkelmessungen dienten BRICK-Sensorknoten [Beus07]. Diese sind quadratisch mit

etwa 6,5 cm Seitenlänge und enthalten an jeder Seitenkante mittig einen Ultraschallsensor, der sowohl senden als auch empfangen kann. Auf der Platine befindet sich außerdem ein Steckplatz für ein Particle (Abbildung 6.2), welches die Funkkommunikation untereinander und mit einem Empfänger an einem PC ermöglicht. Zwei dieser BRICK-Sensorknoten dienen als Basis für die Entfernung- und Winkelmessungen. Zur vereinfachten Handhabung war ein Knoten als Sender konfiguriert, der andere als Empfänger. Als Empfänger am PC arbeitete eine Particle-USB-Bridge.

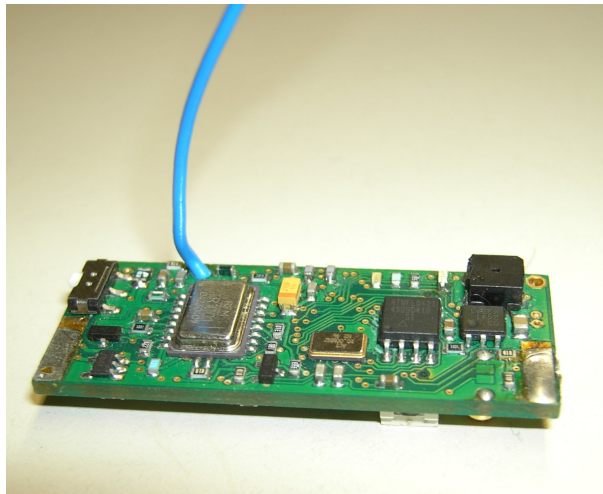


Abbildung 6.2: Particle communication board.

Als verbindendes Hardwareelement dienen Particles. Sie ermöglichen das Auslesen verschiedener Parameter der übrigen Hardware und ermöglichen die geräteübergreifende Kommunikation und die Anbindung an einen PC. Es handelt sich bei dem Particle Computer System um ein sensorbasiertes, vernetztes eingebettetes System, dessen kleine Sensorknoten sowohl im Ad-Hoc-Betrieb untereinander als auch mit einem PC-Backend kommunizieren können. Für eine detailliertere Vorstellung der Particles siehe [DKBZ05].

## 6.2 Erster Messaufbau zur Sammlung diskreter Werte

Der anfängliche Versuchsaufbau bestand aus zwei BRICK-Sensorknoten. Es wurde versucht, die Vermutung einer Winkelabhängigkeit der Entfernungsmessungen zu bestätigen. Dazu wurden die beiden Sensorknoten in verschiedenen Abständen und in unterschiedlichen Winkeln zueinander positioniert. Nach einigen Messungen wurde der Winkel leicht verändert. Nach einer kompletten Umdrehung wurde eine neue Position gewählt. Durch den großen Zeitaufwand für das exakte Positionieren ergeben sich kleine Datensätze mit nur wenigen Entfernungen oder Datensätze über mehrere Entfernungen, aber mit recht großen Lücken zwischen den Winkeln.

Der Wunsch nach möglichst kontinuierlichen Datensätzen, die viele Positionen in geringem Abstand mit einer guten Abdeckung der Winkelmenge enthalten, lässt sich auf manuellem Wege nur mühsam erreichen. So ergab auch eine Auswertung der Datensätze (im Folgenden und im Anhang als Datensatz M1 bis M15 bezeichnet), dass die gewonnenen Werte zu diskret sind, wodurch es dem ANFIS-System nur möglich ist, eine Zuordnung der Messwerte zu den einzelnen Entfernungen herzustellen,



Abbildung 6.3: Zwei BRICK-Sensorknoten manuell positioniert und ausgemessen.

und somit die Trainingswerte „auswendig zu lernen“. Mehr dazu in Kapitel 7.4. Es war also eine Methode gefragt viele Messungen durchzuführen, ohne die korrekten Werte (Ground Truth) bei jeder Messung manuell bestimmen zu müssen.

### 6.3 Zweiter Messaufbau zur Sammlung kontinuierlicher Werte

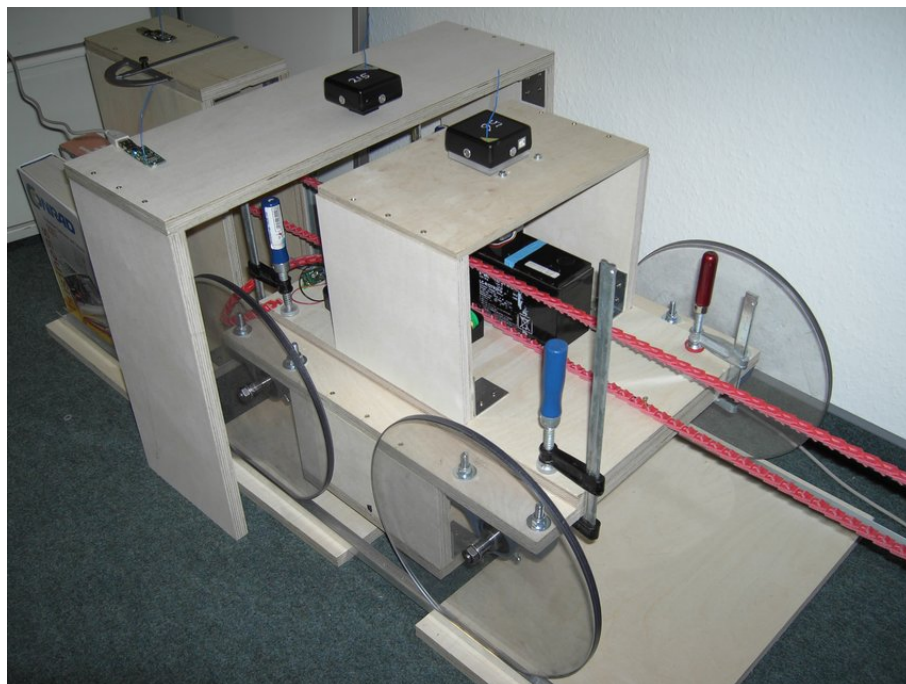


Abbildung 6.4: Roboterwagen für die automatischen Messungen.

Um der Problematik des großen Zeitaufwandes für umfassende kontinuierliche Messungen zu begegnen wurde ein Roboter eingesetzt. Bei diesem handelt es sich um einen Wagen, der auf Schienen eine etwa 180cm lange Strecke zwischen zwei Lichtschranken zurücklegen kann. Gesteuert wird dieser Roboter über Particles an den Motoren. Der Senderknoten befindet sich an einem Ende der Schienen, der



Empfängerknoten bewegt sich zusammen mit dem Roboter. Ein Aufbau mit einem Schrittmotor ermöglicht es, den Empfängerknoten auf dem Roboter in Schritten von weniger als einem halben Grad zu drehen. Durch einen zweiten Aufbau lässt sich der Wagen auch für eine andere Arbeit mit RFID-Messungen an einem Regal verwenden.

Um die Dauer einer Messreihe auf 1-2 Tage zu beschränken wurde eine Schrittweite von  $1.8^\circ$  für das Drehen des Empfängerknotens gewählt. Dies entspricht 200 Drehbewegungen für eine komplette Umdrehung. Die Messpositionen des Roboters liegen für jede komplette Umdrehung des Empfängerknotens etwa 2.5cm auseinander. In jeder Position wurden mindestens 5 Messwerte gesammelt um sicherzustellen, dass der Datensatz möglichst vollständig ist. Der Schematische Aufbau der Messungen ist auf Abbildung 6.5 zu sehen.

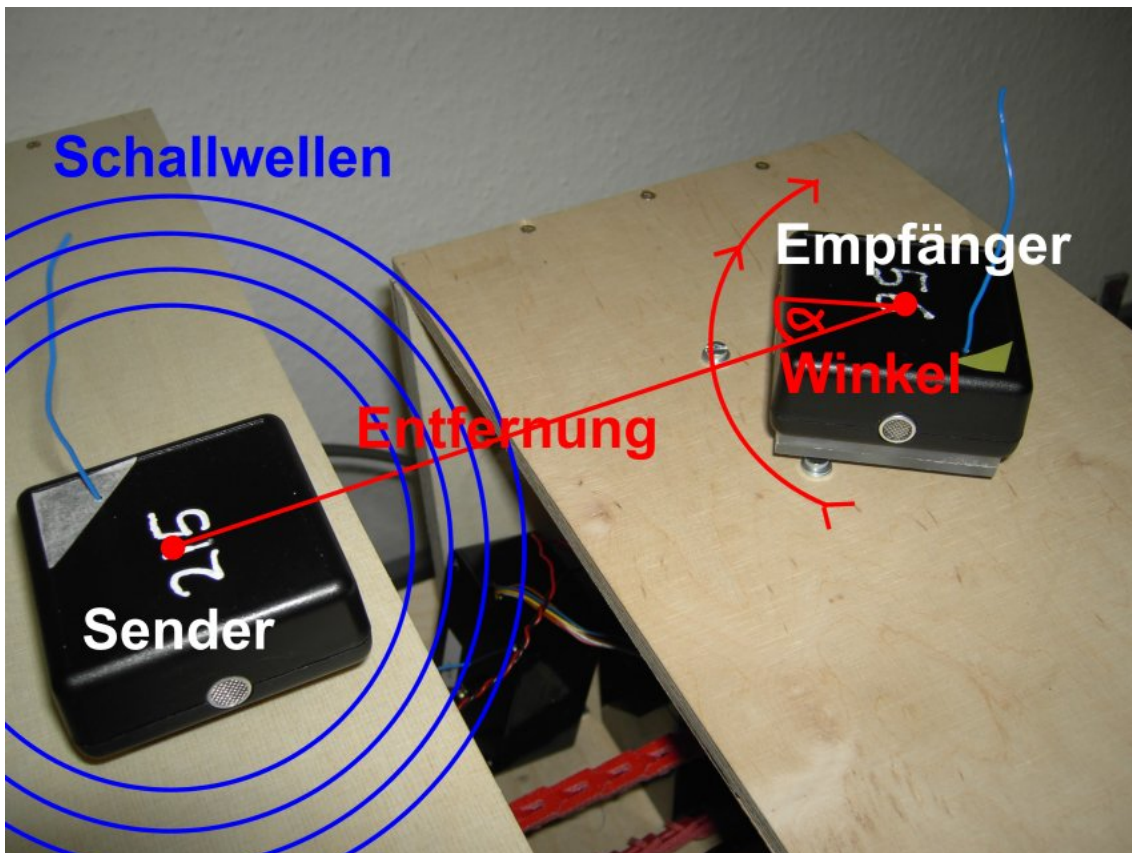


Abbildung 6.5: Schematischer Aufbau der Robotermessungen.

Eine Messung über die gesamte Strecke, die der Roboter auf den Schienen zurücklegen kann, dauerte etwa einen Tag, wobei es immer wieder zu Verzögerungen kam, wenn der Roboter eine Position erreichte, in der die Funksignale der Particles nicht mehr am PC ankamen, wogegen eine Veränderung der Ausrichtung und Position des Empfängers half. Probleme ergaben sich auch durch die Lage der Schienen parallel zu einer Wand, wodurch Messungen für Entfernungen zwischen 120cm und 190cm, bei denen der Wagen sich in mittlerer Position auf den Schienen befand, durch ungünstige Reflexionen fast durchgängig viel zu geringe Entfernungen ergaben.

Um diesem Problem zu begegnen wurden die Schienen von der Wand wegbewegt und mitten im Raum positioniert. Insgesamt kam es so zu einem Datensatz für Entfernungen von 15cm bis 190cm (im Folgenden und im Anhang als Datensatz A

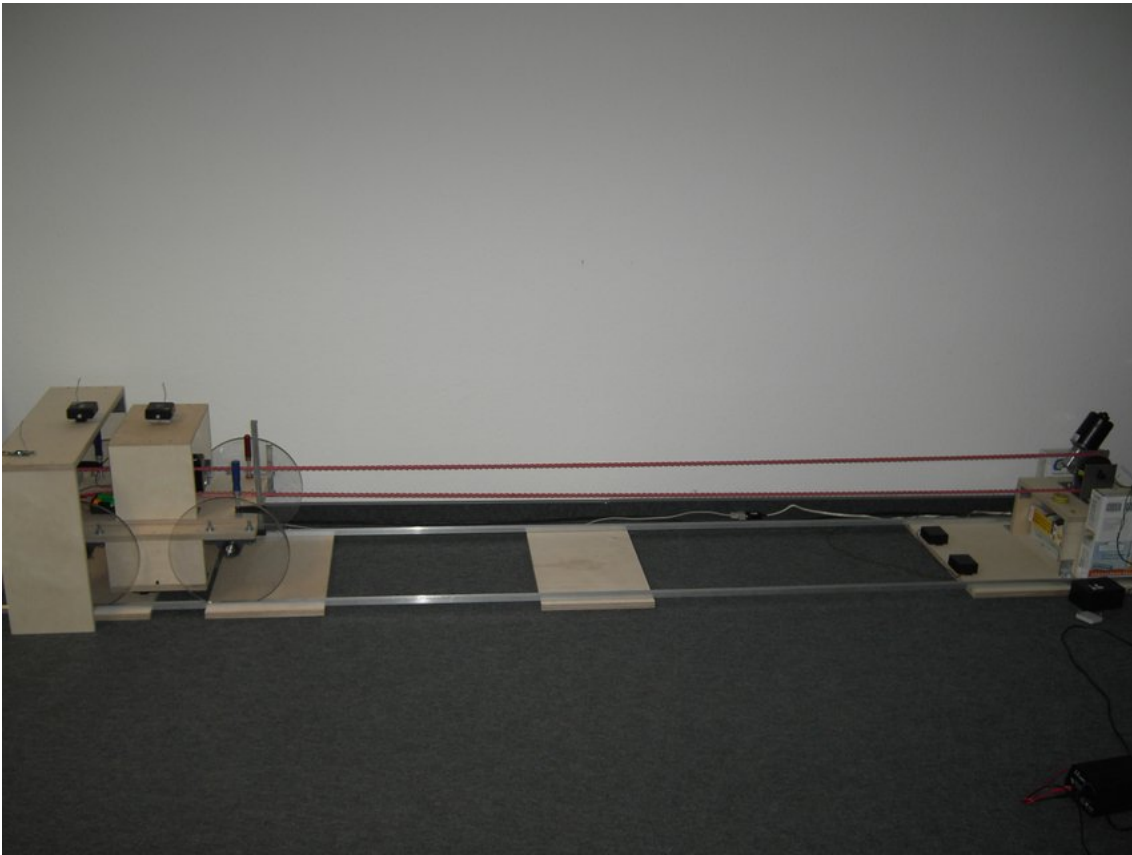


Abbildung 6.6: Roboterwagen - Schienenstrecke.

bezeichnet), wobei sowohl Hinweg als auch Rückweg des Roboters enthalten sind. Ein weiterer Datensatz wurde für Entfernungen von 180cm bis etwa 300cm erzeugt (im Folgenden und im Anhang als Datensatz B bezeichnet). Auch darin sind einmal die Messungen für den Hin- und einmal für den Rückweg enthalten. Entfernungen zwischen 300cm und etwa 350cm waren ebenfalls enthalten, bestanden allerdings zum größten Teil aus ungültigen Messungen und wurden deswegen verworfen.



# 7. Aufbau des ANFIS-Trainings und dessen Ergebnisse

In diesem Kapitel werden der Aufbau der aus den Praxisversuchen gewonnenen Datensätze und die Auswahl der für das Training verwendeten Merkmale erläutert. Außerdem wird der Trainingsablauf geschildert, sowie die Ergebnisse des Trainings evaluiert.

## 7.1 Datensätze aus dem Messaufbau

Ein Datensatz besteht aus einer Anzahl von Messungen an einer oder mehreren Positionen der Sensorknoten, wobei jeweils verschiedene Ausrichtungen gemessen wurden. Stellt man den Datensatz tabellarisch dar, so ergibt jede Messung eine Zeile. Diese enthält in den Spalten die berechnete Entfernung, den berechneten Winkel, die gemessenen Signallaufzeiten der vier Ultraschallempfänger, die vier gemessenen Signalstärken, vier valid-Bits aus dem Algorithmus zur Berechnung von Entfernung und Winkel, die Temperatur zum Zeitpunkt der Messung und die ground truth, also nachgemessene Entfernung und Winkel.

Ein Datensatz mit manueller Positionierung und Ausrichtung enthält zwischen 750 und 10.000 Messungen. Die Entfernungen bleiben dabei innerhalb eines Datensatzes konstant, so dass erst die Kombination mehrerer solcher Datensätze einen größeren Entfernungsbereich abdeckt. Die gemessenen Entfernungen liegen zwischen 15cm und 200cm. Ein Datensatz aus der automatisierten Messung mit Hilfe des Roboters enthält etwa 200.000 Messungen. Datensatz A deckt Entfernungen von 15cm bis 190cm ab. Datensatz B Entfernungen von 190cm bis etwa 300cm.

## 7.2 Merkmalsauswahl

Für die Weiterverarbeitung der Datensätze stellt sich die Frage, ob es sinnvoll ist alle Merkmale eines Datensatzes zu verwenden, oder ob die Qualitätsbestimmung vielleicht sogar verbessert wird, wenn die Anzahl der einbezogenen Merkmale reduziert wird. Um dieser Frage auf den Grund zu gehen wurden von allen Datensätzen verschiedene Varianten erstellt, die verschiedene Kombinationen der Merkmale enthalten. Dazu gehören die reinen Signallaufzeiten, die reinen Signalstärken und Kombinationen mit Entfernungs- und Winkelangaben. Eine weitere Frage in die-

sem Zusammenhang ist die erlaubte Toleranz in den berechneten Entfernungs- und Winkelangaben. Es wurden Varianten mit 5 %, 8 %, 10 % und 20 % Toleranz erstellt.

### 7.3 Trainingsablauf

Das Training wurde mit den ANFIS-Funktionen des Mathematikprogrammes Matlab durchgeführt. Dazu benötigt man einen Datensatz um das neuronale Netz zu trainieren und einen der als Testdatensatz dient, um in jedem Schritt die Auswertungsfehler zu bestimmen und durch das weitere Training zu minimieren. Zusätzlich kann noch ein weiterer Datensatz verwendet werden, der Checkingdatensatz. Dieser fließt nicht in das Training ein, sondern wird nach jedem Trainingsschritt erneut auf Auswertungsfehler überprüft.

Normalerweise sinkt der Auswertungsfehler des Testdatensatzes mit jedem Trainingsschritt. Allerdings kann es passieren, dass ein neuronales Netz "übertrainiert" wird und sich die Auswertungsfehler für unbekannte Datensätze erhöhen, da sich das neuronale Netz auf die bekannten Trainings- und Testdaten "spezialisiert". Mit einem Checkdatensatz soll nun dieser Zeitpunkt erkannt werden. Wenn sich der Auswertungsfehler des Checkingdatensatzes nicht mehr verbessert, dann wird das Training abgebrochen, auch wenn sich beim Testdatensatz noch Verbesserungen ergeben würden. Mittels der Matlab-Funktion `evalfis` lässt sich ein zuvor trainiertes

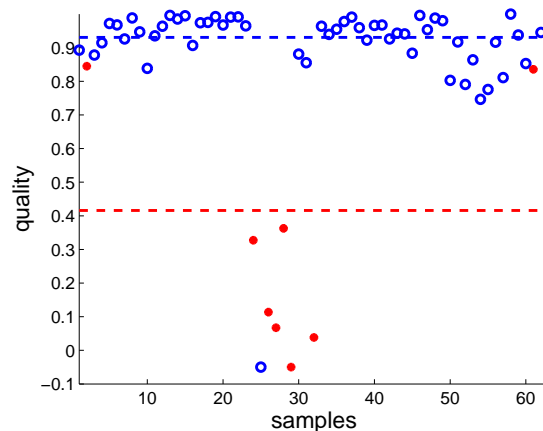


Abbildung 7.1: Ein gut separierbarer Beispieldatensatz. Gültige Werte als blaue Kreise, ungültige als rote Punkte. Die Mittelwerte als gestrichelte Linien.

FIS mit einem Datensatz auswerten.

Das Ergebnis besteht aus einer Matrix mit zwei Spalten, zum einen die durch den Vergleich der ground truth mit den Messergebnissen unter Berücksichtigung einer Toleranzschwelle ermittelten Werte, ob eine Messung ein gültiges oder ein ungültiges Ergebnis geliefert hat und zum anderen der durch das ANFIS ermittelte Qualitätswert für jede Messung.

Zum schnellen Beurteilen der Separierbarkeit bietet es sich an, diese Ergebnisse zu plotten. Es lassen sich dabei, z.B. durch unterschiedliche Farben, die Messungen die ein gültiges Ergebnis geliefert haben von denen abgrenzen, die ein ungültiges Ergebnis geliefert haben. Für alle Messungen wird nun der Qualitätswert aus dem Intervall  $[0...1]$  aufgetragen. Gültige Messungen sollten dabei einen Qualitätswert in der Nähe von 1 aufweisen, ungültige Messungen möglichst einen Qualitätswert von 0. Abbildung 7.1 zeigt dies beispielhaft für einen Datensatz. Messungen denen



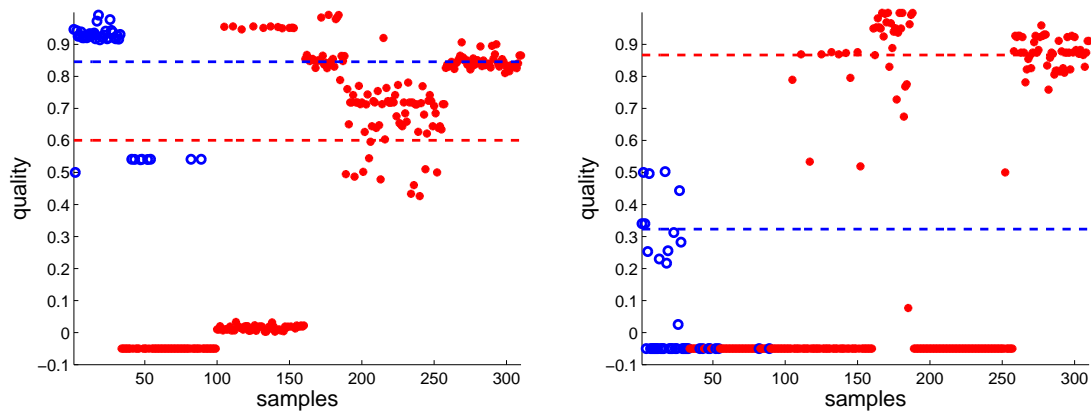


Abbildung 7.2: Diskrete Datensätze. Links: normalisiert, rechts: min. max.

durch das ANFIS kein Qualitätswert zugeordnet werden konnte bekommen einen Wert kleiner 0 zugewiesen.

## 7.4 Evaluierung der Trainingsergebnisse

Auf die in diesem Kapitel beschriebene Weise wurden nun hunderte verschiedene Varianten pro Datensatz mit ANFIS-Methoden trainiert und die Auswertung mit einem Testdatensatz durchgeführt. Die geplotteten Qualitätsverteilungen galt es nun auf verschiedene Fragestellungen zu untersuchen.

- Macht es Sinn, alle zur Verfügung stehenden Merkmale in den Datensätzen ins Training einzubeziehen?
- Sind die Messungen vom Winkel abhängig und lassen sie sich durch das Einbeziehen der gemessenen Winkel die Qualitätsangaben verbessern?
- Ist es sinnvoll Temperaturmessungen in das Training einzubeziehen?
- Ist es sinnvoller mit den Signalstärken zu arbeiten, oder mit den Signallaufzeiten?
- Lässt sich nach dem Training mit einem Datensatz ein anderer separieren?

Für die diskreten Werte des manuellen Versuchsaufbaus war es zunächst nicht möglich diesen Fragestellungen nachzugehen.

### 7.4.1 Diskrete Werte

Es zeigte sich, wie in Kapitel 6.2 beschrieben, dass die manuell durchgeführten Messungen für das Trainieren mittels ANFIS-Methoden schlecht geeignet sind. Es ergeben sich dabei trainierte FISe, die nicht in der Lage sind für einen zweiten Datensatz aus dem gleichen Entfernungsbereich ein separierbares Ergebnis zu liefern. Stattdessen lassen sich für viele der Messungen keine Qualitätswerte zuordnen (siehe dazu Abbildung 7.2, Werte kleiner als 0 ließen sich nicht zuordnen).

Zwischen den einzelnen ausgemessenen Entfernungen und Winkeln gibt es Lücken, die es dem ANFIS-System erschweren die allgemeine Zuordnung der schlechten Messungen zu den zugehörigen Sensorwerten zu approximieren. Stattdessen wird eine Zuordnung der Qualität zu den Entfernungen hergestellt. Das System lernt sozusagen die Entfernungen "auswendig", die gute Qualität ergeben und diejenigen, die

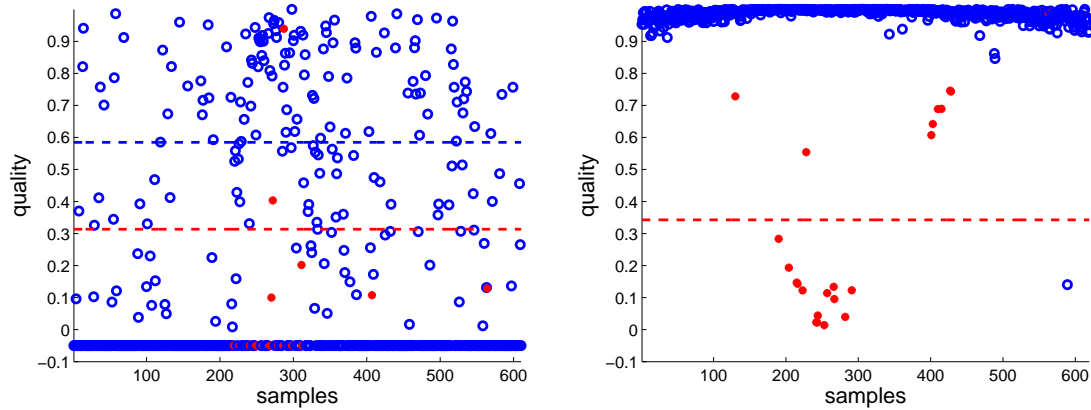


Abbildung 7.3: Links: Signalstärke+Signallaufzeit, rechts: Signallaufzeit

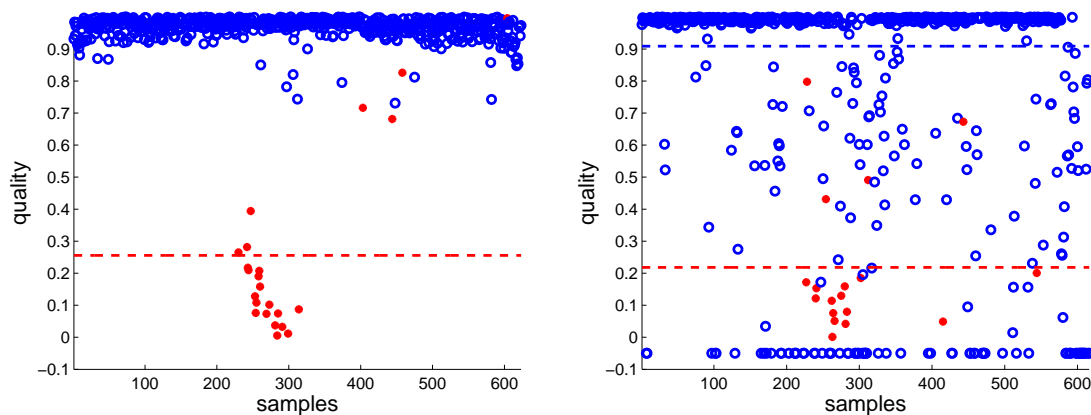


Abbildung 7.4: Links: Ohne Temperaturwerte, rechts: Mit Temperaturwerten

schlechte Qualität ergeben. Versucht man nun einen anderen Datensatz für den gleichen Entfernungsbereich damit zu beurteilen, so enthält das System nicht die passenden Merkmale und kann viele Werte nicht zuordnen. Das Ergebnis ist dann auch unmöglich zu separieren.

### 7.4.2 Kontinuierliche Werte

Wie in Kapitel 6.3 beschrieben, wurden mittels eines Roboters zwei Datensätze mit Messungen erzeugt, die wesentlich kleinere Abstände zwischen den einzelnen Messpunkten haben. Dadurch sollte sichergestellt werden, dass ein Datensatz ausreichend Informationen für jeden enthaltenen Entfernungsbereich besitzt um dem trainierten ANFIS-System zu ermöglichen, auch für Messungen aus einem zweiten Datensatz des gleichen Entfernungsbereichs eine passende Qualität zuzuordnen.

Die oben genannte Frage, ob es Sinn macht, alle in einem Datensatz zur Verfügung stehenden Merkmale ins Training einzubeziehen, lies sich anhand der Auswertung der Testingsdatensätze recht schnell klären. In den meisten Fällen verschlechtert sich die Separierbarkeit der Messungen deutlich gegenüber eingeschränkten Trainingsätzen. Die große Fülle der Merkmale erschwert es dem ANFIS auf die signifikanten Merkmale zu trainieren, so dass sich auch auffallend viele Messungen nicht zuordnen ließen. Abbildung 7.3 zeigt dies durch Werte kleiner 0 exemplarisch für einen Teilbereich von 65 bis 120cm des Datensatz A.

Die Einbeziehung der Temperaturmessungen in die Trainingsmerkmale führte zumeist zu einer Verschlechterung der Separierbarkeit, beispielhaft zu sehen in Ab-

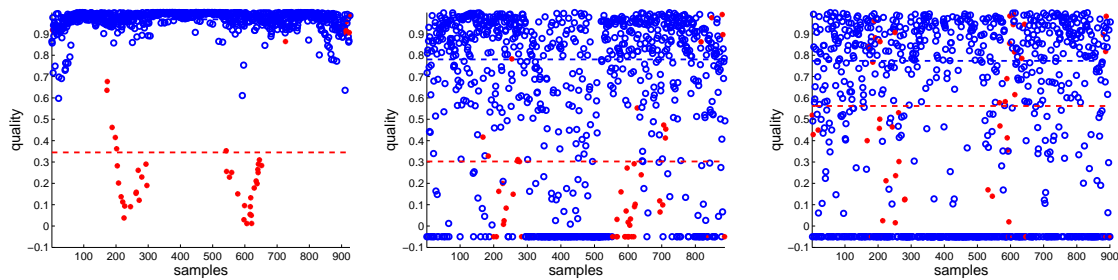


Abbildung 7.5: Signallaufzeiten Links: +Entfernung, Mitte: +Entfernung & Winkel, rechts: +Winkel

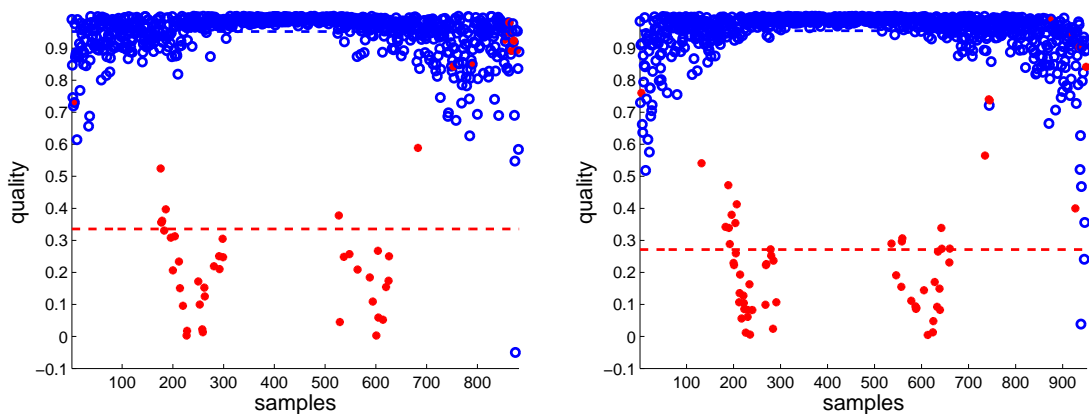


Abbildung 7.6: Signalstärken Links: +Entfernung, rechts: +Entfernung & Winkel

Abbildung 7.4. Da die Temperatur über die Messungen zumeist sehr konstant geblieben ist und ein Datensatz somit kaum unterschiedliche Temperaturwerte enthält, liegt die Vermutung nahe, dass die Temperaturwerte nur die Komplexität des ANFIS erhöhten ohne konkrete Ansatzpunkte für eine Klassifizierung der Messwerte zu bieten.

Beim Einsatz des Systems mit größeren Temperaturunterschieden, die sich dann tatsächlich auf die Qualität der Messungen auswirken, könnte die Einbeziehung der Temperaturwerte als Trainingsmerkmale doch noch einen positiven Einfluss haben.

Nutzt man die Signallaufzeiten als Merkmal, so bietet es sich an die Entfernungsmessungen, aber nicht die Winkelmessungen einzubeziehen. Signallaufzeiten und Entfernungen ergeben in einigen Fällen ein besser separierbares Ergebnis als Signallaufzeiten, Entfernungen und Winkel. Signallaufzeiten und Winkel wiederum ergeben nochmals ein schlechter separierbares Ergebnis. Abbildung 7.5 zeigt dies an einem der auffälligsten Beispiele, einem Teilbereich des Datensatzes A bis 190cm im Bereich 15-35cm bei normalisierten Signallaufzeiten.

Der Effekt der Winkelmessungen als Merkmal zusammen mit Entfernungsmessungen und Signalstärken ist sehr gering. Zumeist ergibt sich nur eine minimale positive oder negative Veränderung gegenüber Signalstärken und Entfernungsmessungen alleine. Ein typisches Beispiel ist in Abbildung 7.6 zu sehen. Ein wichtiger Punkt für den Praxiseinsatz einer Qualitätsbeurteilung mittels ANFIS-Methoden ist die Übertragbarkeit des trainierten FIS auf eine neue Problemstellung. Dazu galt es die Frage zu beantworten, ob sich nach dem Training mit einem Datensatz ein anderer Datensatz mit diesem FIS separieren lässt. Dies gelang mit dem Training des ersten kontinuierlichen Datensatzes A für Entfernungen von 65 bis 120cm und

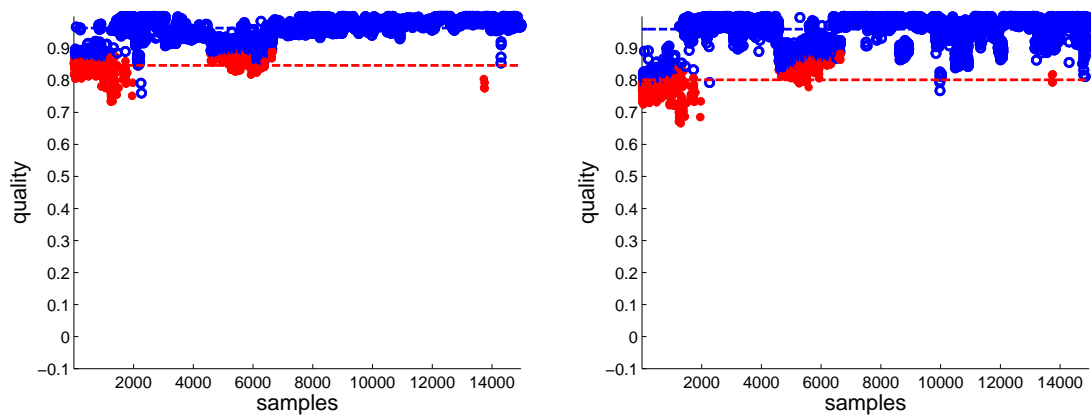


Abbildung 7.7: Training mit kontinuierlichem Datensatz, Auswertung mit diskretem Datensatz (80cm).

der Auswertung eines der manuell erstellten Datensätze (M15), der Messungen für verschiedene Winkel in einer Entfernung von 80cm enthält. Abbildung 7.7 zeigt zwei Varianten die eine gute Separierbarkeit erreichten. Die Qualitätswerte der als ungültig klassifizierten Messungen fallen zwar recht hoch aus, doch durch eine geeignete Wahl des Schwellwertes lässt sich dennoch eine gute Separierbarkeit erreichen.

Die im Anhang aufgelisteten Daten zeigen außerdem, dass auf den Fehlerzustand vor allem solche Messungen abgebildet werden, die einen größeren als den maximal akzeptierten Fehler enthalten.

## 8. Anwendung des Qualitätssystems zur Ergänzung von Lokations-Algorithmen

Für sich alleine betrachtet ist eine Qualitätsangabe in einem Lokationssystem nicht sonderlich nützlich. Eine mögliche Anwendung ist die Filterung von Messungen, um die Zahl der in einem Sensornetzwerk zu kommunizierenden Daten zu reduzieren, statistisch z.B. indem nur Messungen, die über einem bestimmten Schwellwert liegen, an die anderen Knoten im Netzwerk übertragen werden. Wirklich deutlich wird der Wert der Qualitätsangaben erst durch das Einbinden in einen Algorithmus (z.B. in einen Fusions-Algorithmus). Im Folgenden soll dies am Beispiel der Hyperbelnavigation gezeigt werden.

### 8.1 Hyperbelnavigation

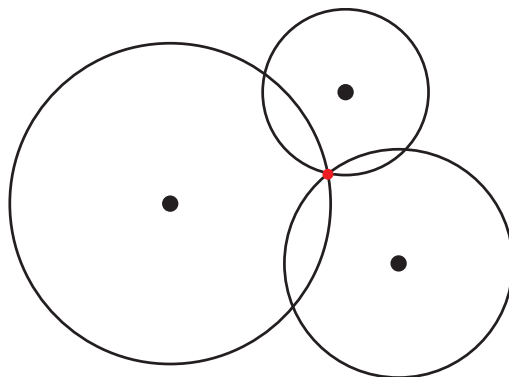


Abbildung 8.1: Bestimmung einer Position mittels Hyperbelnavigation.

Durch die Hyperbelnavigation wird die Position z.B. eines Sensorknotens bestimmt, indem die Signallaufzeiten zu Sendern bzw. Empfängern deren Position bekannt ist gemessen werden. Bekannt ist dieses Verfahren durch den Einsatz in GPS-Systemen. Dort dienen Satelliten mit bekannten Umlaufbahnen als Sender und

jeder Empfänger bestimmt aus den Signallaufzeiten und den mitgesendeten Positionen der Satelliten seine Position. Durch die gemessenen Signallaufzeiten ergibt sich um jeden Sender mit bekannter Position ein kugelförmiger Bereich für die gesuchte Position des Empfängers. Dieser kann nun mit den ermittelten Bereichen der übrigen Sender geschnitten werden. Aus dem Schnitt von vier Bereichen ergibt sich ein eindeutiger Punkt im dreidimensionalen Raum. Für einen GPS-Empfänger würden sogar drei Satelliten ausreichen, da von den dann möglichen zwei Positionen eine nicht auf der Erdoberfläche liegt. Allerdings wird für die Bestimmung der exakten Zeit im Empfänger wiederum ein vierter Satellit benötigt. Beschränkt man sich auf eine Ebene, also einen zweidimensionalen Raum, so ist jeder Bereich nicht mehr kugel-, sondern kreisförmig. Damit erhält man schon durch den Schnitt von drei Bereichen eine eindeutig bestimmte Position (siehe Abbildung 8.1).

## 8.2 Praxisanwendung der Hyperbelnavigation

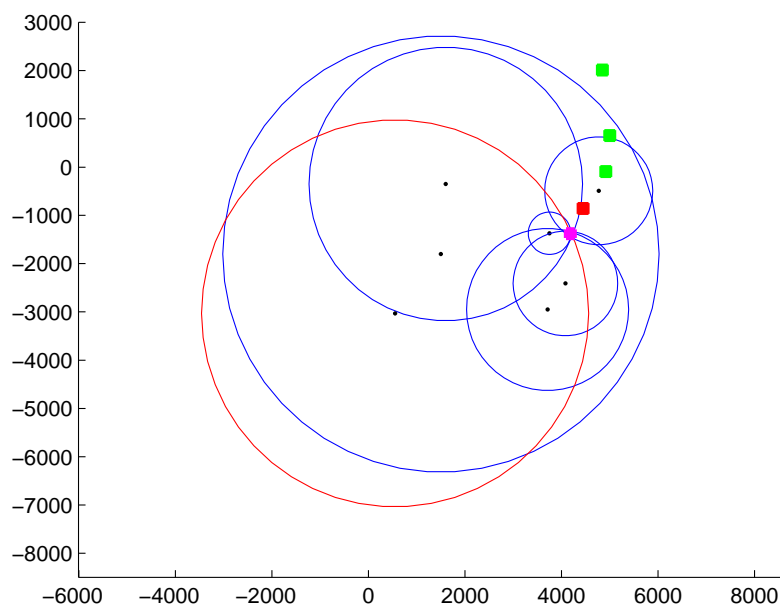


Abbildung 8.2: 7 Knoten, kleinste Fehlerquadrate

Auf diese Weise soll nun aus den Entfernungsmessungen von stationären Sensorknoten mit bekannter Position die Position eines vierten Sensorknotens bestimmt werden, der innerhalb der Reichweite von mindestens drei der stationären Sensorknoten liegt [BRDB<sup>+</sup>08]. Als Grundlage für diese Anwendung diente ein Datensatz aus einem Versuch in einem Feuerwehr-Trainingshaus in Paris. Dabei wurden in einem Übungshaus auf zwei Stockwerken 40 BRICK-Sensorknoten platziert, und deren Entfernungs- und Winkelmessungen, sowie die exakte Position und Orientierung aufgezeichnet. Im so entstandenen Datensatz finden sich für die Hyperbelnavigation verwertbare Sensorknoten, die sich in der Reichweite von 3 bis 11 anderen Sensorknoten befanden. Der Fehler der Entfernungsmessungen schwankte dabei zwischen 1mm und 4516mm, der durchschnittliche Fehler betrug 312mm. Die Hyperbelnavigation liefert auf Grund der fehlerbehafteten Messungen keinen exakten Schnittpunkt, sondern nur einen möglichen Bereich für die gesuchte Position.

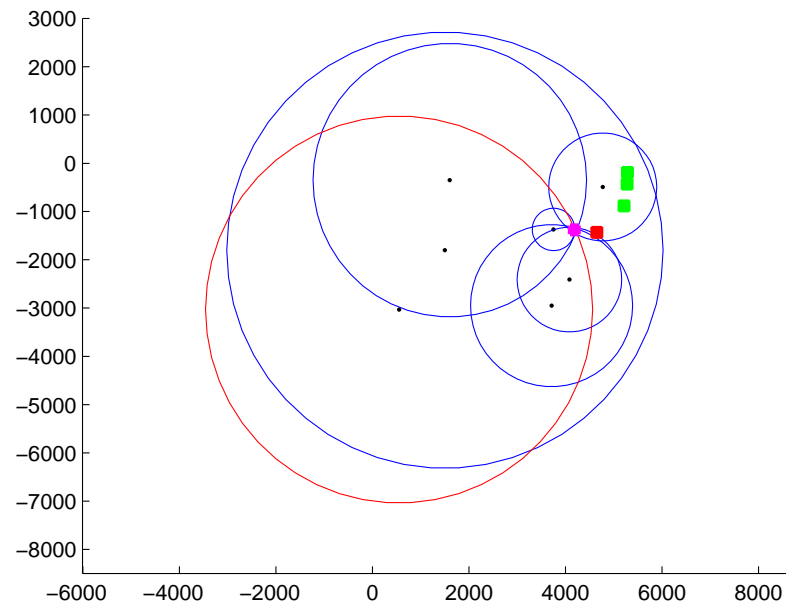


Abbildung 8.3: 7 Knoten, gefilterte kleinste Fehlerquadrate

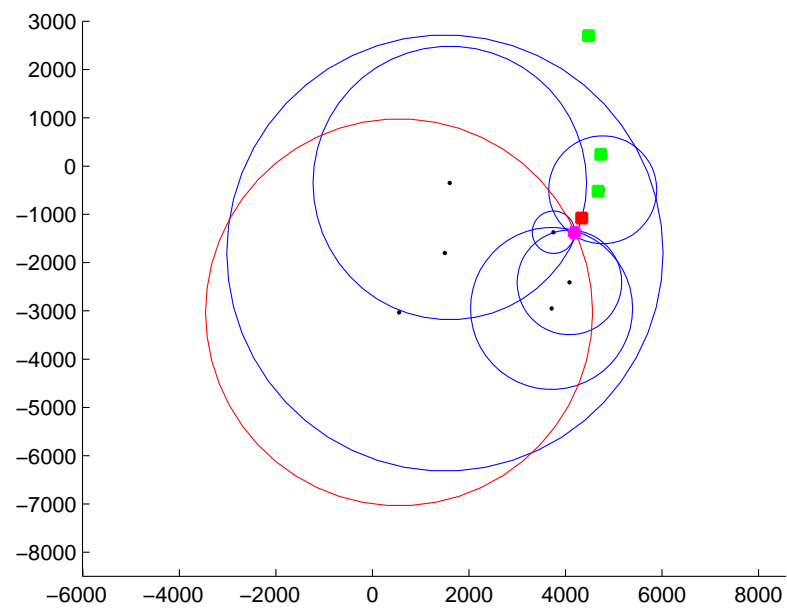


Abbildung 8.4: 7 Knoten, gewichtete kleinste Fehlerquadrate

### 8.3 Methode der kleinsten Fehlerquadrate

Die exakte Position wird dann mit der Methode der kleinsten Fehlerquadrate angenähert. Die Differenz aus so bestimmter Position und der beim Auslegen der Sensorknoten gemessenen Position wurde aufsummiert und der so ermittelte Fehler für alle beim jeweiligen Sensorknoten möglichen Anzahlen an verwertbaren Messungen bestimmt. Diese Werte dienen als Referenz zum Vergleich mit zwei Varianten des Algorithmus, bei denen die Methode der kleinsten Fehlerquadrate durch die Qualitätsangaben aus einem für diesen Datensatz trainierten ANFIS unterstützt wird. Einen Vergleich zwischen den drei Varianten für einen Beispielknoten aus dem Datensatz zeigen die Abbildungen 8.2, 8.3 und 8.4.

### 8.4 Gefilterte Methode der kleinsten Fehlerquadrate

Für die gefilterte Methode der kleinsten Fehlerquadrate wurden zunächst die in Frage kommenden Messungen nach der ihnen zugeordneten Qualitätsangabe sortiert. Berücksichtigt wurden nun für jede Anzahl an verwertbaren Messungen nur solche, deren zugeordnete Qualität größer als 0.5 war. Blieben dadurch keine drei Messungen mehr übrig, so wurden die drei Messungen mit dem größten Qualitätswert verwendet. Durch dieses Vorgehen blieben Messungen unberücksichtigt, deren niedriger Qualitätswert einen größeren Fehler in der Entfernungsbestimmung nahelegt. Der Vergleich mit der einfacheren Referenzvariante zeigt ab vier berücksichtigten Knoten eine Verbesserung sowohl beim durchschnittlichen (Abbildung 8.5), als auch beim maximalen Fehler (Abbildung 8.6).

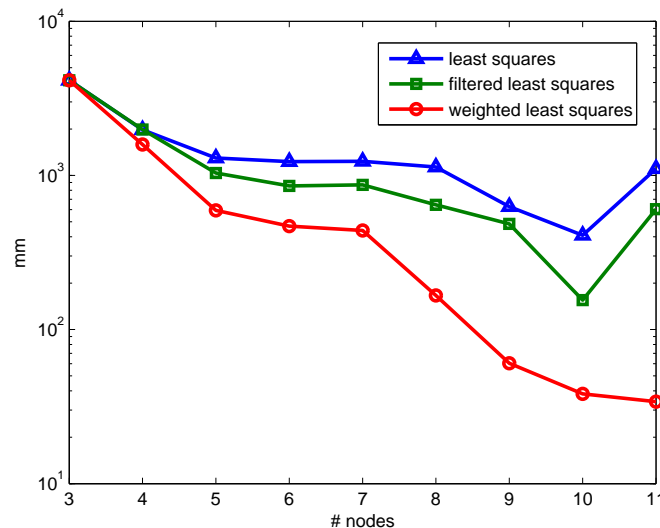


Abbildung 8.5: Durchschnittlicher Fehler, aufgelistet über die Anzahl der Knoten.

### 8.5 Gewichtete Methode der kleinsten Fehlerquadrate

Als letztes kam noch eine gewichtete Methode der kleinsten Fehlerquadrate zum Einsatz, bei der die Qualitätswerte der Messungen als Gewichte dienten. Hierbei werden immer alle zur Verfügung stehenden Messungen einbezogen, allerdings mit



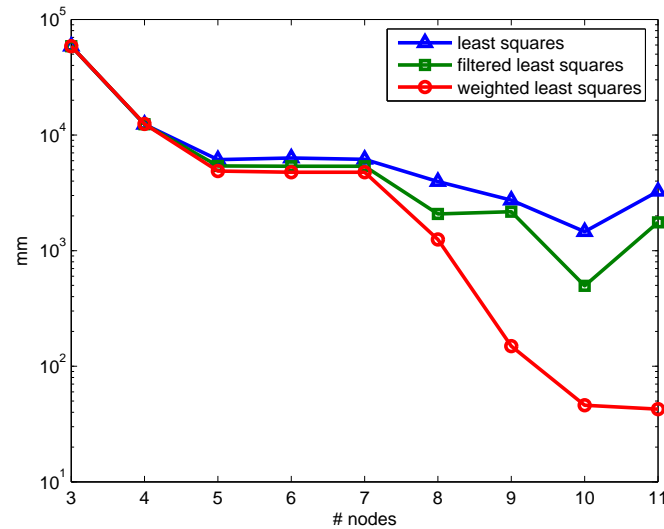


Abbildung 8.6: Maximaler Fehler, aufgelistet über die Anzahl der Knoten.

der Intention durch die Gewichtung den Anteil potentiell fehlerträchtiger Messungen zu reduzieren. Der Vergleich mit den beiden vorherigen Verfahren zeigt eine weitere teils sehr deutliche Verbesserung sowohl gegenüber der einfachen Referenz als auch gegenüber der gefilterten Variante, die besonders bei größeren Knotenzahlen zum Tragen kommt.



## 9. Zusammenfassung und Ausblick

In dieser Arbeit wurde das Problem der Detektion von Messfehlern und der Qualitätsbeurteilung in der relativen Lokationsbestimmung vorgestellt. Es wurden verschiedene Verfahren zur Lokationsbestimmung erläutert und mögliche Fehlerquellen aufgezeigt. Es wurde ein Überblick über Verfahren zur Zustandsvorhersage in Lokationssystemen gegeben und die Verfahren auf ihre Qualitätsaussagen hin untersucht. Es stellte sich heraus, dass die bestehenden Verfahren bestenfalls eine allgemeine Aussage zu den in der Positionsbestimmung enthaltenen Fehlern machen können.

Es wurde mit ANFIS und den TSK-FIS ein Verfahren vorgestellt, um eine individuelle Qualitätsaussage zu jeder Entfernungs- oder Winkelmessung eines Lokationssystems machen zu können. Dieses Verfahren wurde durch Messungen mit BRICK-Sensorknoten getestet und es wurden verschiedene Trainingsmerkmale verglichen. Abschließend wurde die Praxistauglichkeit des Verfahrens anhand einer Beispielanwendung gezeigt, wobei Daten eines echten Einsatzszenarios zum Einsatz kamen. Hier konnte eine deutliche Verbesserung in der Lokationsbestimmung unter Einbeziehung der durch ein FIS ermittelten Qualitätsaussagen zu den einzelnen Entfernungsmessungen erreicht werden.

Ansatzpunkte für zukünftige Arbeiten bieten sich bei der Auswahl der Merkmale für das Training, wo weitere Erkenntnisse wünschenswert wären. Die Übertragung des Systems auf andere Lokationssysteme bietet sich genauso an, wie die Nutzung in einem größeren System mit mehreren Knoten. Schließlich bietet die Erstellung der FIS noch Optimierungspotential, z.B. durch eine Verringerung der Regelanzahl.



# A. Anhang

Erklärungen:

korrekte Messungen: Messungen die einen festgelegten Fehler nicht überschreiten.

falsche Messungen: Messungen die einen festgelegten Fehler überschreiten.

TP: True Positives, die Anzahl der richtig zugeordneten korrekten Messungen.

TN: True Negatives, die Anzahl der richtig zugeordneten falschen Messungen.

FP: False Positives, die Anzahl der falsch zugeordneten korrekten Messungen.

FN: False Negatives, die Anzahl der falsch zugeordneten falschen Messungen.

EP: Errorstate Positives, die Anzahl der auf den Fehlerzustand abgebildeten korrekten Messungen.

EN: Errorstate Negatives, die Anzahl der auf den Fehlerzustand abgebildeten falschen Messungen.

TP, TN, FP, FN, EP, EN werden jeweils als Anteil des betrachteten Datensatzes angegeben. Zusammengekommen ergeben sie den kompletten Datensatz.

Der Schwellwert (Threshold) dient als Grenze zur Bestimmung der True/False Positives/Negatives. Er ergibt sich für jeden Datensatz aus der Mitte zwischen den Durchschnittswerten für die korrekten und die falschen Messungen.

Daten zu den Abbildungen:

Abbildung 7.1:

TP	TN	FP	FN	EP	EN	Threshold
84.13%	7.94%	0.00%	4.76%	1.59%	1.59%	0.67

Signallaufzeiten, Entfernung, Distanzfehler, normalisiert

Training: Datensatz A3, 8% Toleranz, Testing: Datensatz A3

Abbildung 7.2 links:

TP	TN	FP	FN	EP	EN	Threshold
10.32%	32.90%	2.90%	35.16%	18.71%	0.00%	0.72

Signalstärken, Entfernung, Winkel, Winkelfehler, normalisiert

Training: Datensatz M1, 8% Toleranz, Testing: Datensatz M2

Abbildung 7.2 rechts:

TP	TN	FP	FN	EP	EN	Threshold
0.00%	1.29%	4.19%	28.39%	57.10%	9.03%	0.60

Signalstärken, Entfernung, Winkel, Winkelfehler, min.+ max.

Training: Datensatz M1, 8% Toleranz, Testing: Datensatz M2

Abbildung 7.3 links:

TP	TN	FP	FN	EP	EN	Threshold
21.31%	0.82%	10.66%	0.16%	2.13%	64.92%	0.45

Signallaufzeiten, Signalstärken, Entfernung, Distanzfehler, min.+ max.

Training: Datensatz A3, 10% Toleranz, Testing: Datensatz A3

Abbildung 7.3 rechts:

TP	TN	FP	FN	EP	EN	Threshold
96.12%	2.75%	0.16%	0.97%	0.00%	0.00%	0.66

Signallaufzeiten, Entfernung, Distanzfehler, min.+ max.

Training: Datensatz A3, 10% Toleranz, Testing: Datensatz A3

Abbildung 7.4 links:

TP	TN	FP	FN	EP	EN	Threshold
96.31%	3.05%	0.00%	0.64%	0.00%	0.00%	0.61

Signallaufzeiten, Entfernung, Winkel, Distanzfehler

Training: Datensatz A3, 10% Toleranz, Testing: Datensatz A3

Abbildung 7.4 rechts:

TP	TN	FP	FN	EP	EN	Threshold
77.76%	2.60%	7.14%	0.32%	0.00%	12.18%	0.56

Signallaufzeiten, Entfernung, Winkel, Temperaturwert, Distanzfehler

Training: Datensatz A3, 10% Toleranz, Testing: Datensatz A3

Abbildung 7.5 links:

TP	TN	FP	FN	EP	EN	Threshold
94.38%	4.21%	0.32%	1.08%	0.00%	0.00%	0.66

Signallaufzeiten, Entfernung, Distanzfehler, normalisiert

Training: Datensatz A1, 10% Toleranz, Testing: Datensatz A1

Abbildung 7.5 Mitte:

TP	TN	FP	FN	EP	EN	Threshold
65.99%	3.15%	10.59%	0.68%	1.91%	17.68%	0.54

Signallaufzeiten, Entfernung, Winkel, Distanzfehler, normalisiert

Training: Datensatz A1, 10% Toleranz, Testing: Datensatz A1

Abbildung 7.5 rechts:

TP	TN	FP	FN	EP	EN	Threshold
45.73%	2.66%	15.09%	1.89%	0.78%	33.85%	0.67

Signallaufzeiten, Winkel, Distanzfehler, normalisiert

Training: Datensatz A1, 10% Toleranz, Testing: Datensatz A1

Abbildung 7.6 links:

TP	TN	FP	FN	EP	EN	Threshold
93.88%	4.54%	0.45%	1.02%	0.00%	0.11%	0.64

Signalstärken, Entfernung, Distanzfehler

Training: Datensatz A1, 10% Toleranz, Testing: Datensatz A1

Abbildung 7.6 rechts:

TP	TN	FP	FN	EP	EN	Threshold
92.65%	5.88%	0.74%	0.74%	0.00%	0.00%	0.61

Signalstärken, Entfernung, Winkel, Distanzfehler

Training: Datensatz A1, 10% Toleranz, Testing: Datensatz A1

Abbildung 7.7 links:

TP	TN	FP	FN	EP	EN	Threshold
81.00%	11.01%	7.99%	0.00%	0.00%	0.00%	0.91

Signallaufzeiten, Entfernung, Distanzfehler

Training: Datensatz A3, 10% Toleranz, Testing: Datensatz M15

Abbildung 7.7 rechts:

TP	TN	FP	FN	EP	EN	Threshold
78.06%	11.01%	10.93%	0.00%	0.00%	0.00%	0.83

Signallaufzeiten, Entfernung, Winkel, Distanzfehler, min. + max.

Training: Datensatz A3, 10% Toleranz, Testing: Datensatz M15

Weitere Daten:

TP	TN	FP	FN	EP	EN	Threshold
4.36%	0.11%	5.03%	0.22%	4.59%	85.68%	0.514

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler

Training: Datensatz A1, 10% Toleranz, Testing: Datensatz A1

TP	TN	FP	FN	EP	EN	Threshold
18.13%	1.80%	5.30%	0.42%	3.61%	70.73%	0.49

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, normalisiert

Training: Datensatz A1, 10% Toleranz, Testing: Datensatz A1

TP	TN	FP	FN	EP	EN	Threshold
4.46%	0.11%	2.63%	0.00%	5.60%	87.20%	0.37

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, min. + max.

Training: Datensatz A1, 10% Toleranz, Testing: Datensatz A1

TP	TN	FP	FN	EP	EN	Threshold
0.11%	0.00%	0.22%	0.00%	6.65%	93.02%	0.19

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler

Training: Datensatz A1, 10% Toleranz, Testing: Datensatz A1

TP	TN	FP	FN	EP	EN	Threshold
35.74%	3.04%	8.49%	0.73%	1.26%	50.73%	0.53

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler, normalisiert

Training: Datensatz A1, 10% Toleranz, Testing: Datensatz A1

TP	TN	FP	FN	EP	EN	Threshold
12.00%	1.44%	7.67%	1.00%	3.56%	74.33%	0.48

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler, min. + max.

Training: Datensatz A1, 10% Toleranz, Testing: Datensatz A1

TP	TN	FP	FN	EP	EN	Threshold
92.07%	4.67%	1.96%	0.54%	0.32%	0.43%	0.62

Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler

Training: Datensatz A1, 10% Toleranz, Testing: Datensatz A1

TP	TN	FP	FN	EP	EN	Threshold
89.74%	5.93%	1.48%	2.51%	0.11%	0.23%	0.65

Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, normalisiert

Training: Datensatz A1, 10% Toleranz, Testing: Datensatz A1

TP	TN	FP	FN	EP	EN	Threshold
93.82%	5.20%	0.00%	0.98%	0.00%	0.00%	0.64

Training: Datensatz A1, 10% Toleranz, Testing: Datensatz A1

Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, min. + max.

TP	TN	FP	FN	EP	EN	Threshold
78.17%	4.06%	11.17%	1.52%	0.00%	5.08%	0.57

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
54.95%	2.20%	23.08%	3.85%	0.00%	15.93%	0.76

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, normalisiert

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
66.48%	1.10%	26.92%	0.55%	0.55%	4.40%	0.81

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, min. + max.

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
26.49%	0.00%	23.24%	1.08%	2.70%	46.49%	0.70

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
75.42%	2.54%	11.44%	1.27%	0.00%	9.32%	0.64

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler, normalisiert

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
26.29%	1.03%	7.73%	0.00%	2.58%	62.37%	0.56

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler, min. + max.

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
79.68%	3.21%	12.30%	2.67%	0.00%	2.14%	0.73

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
87.13%	0.99%	8.42%	0.99%	0.00%	2.47%	0.68

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, nor-



malisiert

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
46.67%	1.03%	19.49%	0.51%	3.59%	28.72%	0.56

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, min. + max.

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
33.52%	0.57%	25.57%	0.57%	2.27%	37.50%	0.72

Signallaufzeiten, Entfernung, Winkel, Distanzfehler

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
57.92%	3.47%	17.33%	2.97%	1.98%	16.34%	0.73

Signallaufzeiten, Entfernung, Winkel, Distanzfehler, normalisiert

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
74.09%	3.63%	13.47%	1.04%	0.52%	7.25%	0.73

Signallaufzeiten, Entfernung, Winkel, Distanzfehler, min. + max.

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
63.11%	0.97%	21.36%	0.97%	0.00%	13.59%	0.69

Signalstärken, Entfernung, Winkel, Distanzfehler

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
57.29%	4.17%	21.35%	4.17%	0.52%	12.50%	0.73

Signalstärken, Entfernung, Winkel, Distanzfehler, normalisiert

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
72.73%	2.67%	17.65%	1.07%	1.60%	4.28%	0.67

Signalstärken, Entfernung, Winkel, Distanzfehler, min. + max.

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
79.23%	1.64%	14.21%	2.19%	0.00%	2.73%	0.79

Signallaufzeiten, Entfernung, Distanzfehler

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
47.34%	3.38%	14.98%	1.93%	0.97%	31.40%	0.69

Signallaufzeiten, Entfernung, Distanzfehler, normalisiert

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
83.33%	2.45%	10.29%	1.47%	0.00%	2.45%	0.76

Signallaufzeiten, Entfernung, Distanzfehler, min. + max.

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
47.47%	1.52%	10.61%	1.01%	3.54%	35.86%	0.67

Signalstärken, Entfernung, Distanzfehler

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
74.64%	2.39%	13.88%	2.39%	2.87%	3.83%	0.71

Signalstärken, Entfernung, Distanzfehler, normalisiert

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
63.03%	2.84%	31.75%	2.37%	0.00%	0.00%	0.94

Signalstärken, Entfernung, Distanzfehler, min. + max.

Training: Datensatz A2, 10% Toleranz, Testing: Datensatz A2

TP	TN	FP	FN	EP	EN	Threshold
64.04%	0.66%	7.22%	0.66%	2.30%	25.12%	0.78

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler

Training: Datensatz A3, 10% Toleranz, Testing: Datensatz A3

TP	TN	FP	FN	EP	EN	Threshold
66.16%	2.12%	5.31%	0.46%	1.52%	24.43%	0.58

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, normalisiert

Training: Datensatz A3, 10% Toleranz, Testing: Datensatz A3

TP	TN	FP	FN	EP	EN	Threshold
88.46%	2.08%	4.17%	0.80%	0.80%	3.69%	0.64

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, min. + max.

Training: Datensatz A3, 10% Toleranz, Testing: Datensatz A3

TP	TN	FP	FN	EP	EN	Threshold
77.76%	2.60%	7.14%	0.32%	0.00%	12.18%	0.56

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler

Training: Datensatz A3, 10% Toleranz, Testing: Datensatz A3

TP	TN	FP	FN	EP	EN	Threshold
93.90%	2.20%	0.51%	1.36%	0.17%	1.86%	0.70

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler, normalisiert

Training: Datensatz A3, 10% Toleranz, Testing: Datensatz A3

TP	TN	FP	FN	EP	EN	Threshold
94.95%	2.60%	0.98%	0.65%	0.16%	0.65%	0.64

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler, min. + max.

Training: Datensatz A3, 10% Toleranz, Testing: Datensatz A3

TP	TN	FP	FN	EP	EN	Threshold
95.02%	1.96%	1.36%	1.06%	0.15%	0.45%	0.71

Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler

Training: Datensatz A3, 10% Toleranz, Testing: Datensatz A3

TP	TN	FP	FN	EP	EN	Threshold
94.34%	2.99%	1.26%	0.79%	0.00%	0.63%	0.63

Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, normalisiert

Training: Datensatz A3, 10% Toleranz, Testing: Datensatz A3

TP	TN	FP	FN	EP	EN	Threshold
95.21%	3.47%	0.17%	1.16%	0.00%	0.00%	0.67

Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, min. + max.

Training: Datensatz A3, 10% Toleranz, Testing: Datensatz A3

TP	TN	FP	FN	EP	EN	Threshold
44.15%	8.22%	5.80%	0.63%	0.84%	40.36%	0.42

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler

Training: Datensatz A4, 10% Toleranz, Testing: Datensatz A4

TP	TN	FP	FN	EP	EN	Threshold
68.30%	5.66%	8.50%	2.07%	0.87%	14.60%	0.51

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, normalisiert

Training: Datensatz A4, 10% Toleranz, Testing: Datensatz A4

TP	TN	FP	FN	EP	EN	Threshold
45.82%	4.56%	11.94%	8.68%	2.39%	34.42%	0.50

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, min. + max.

Training: Datensatz A4, 10% Toleranz, Testing: Datensatz A4

TP	TN	FP	FN	EP	EN	Threshold
11.03%	0.00%	0.92%	0.00%	8.97%	79.08%	0.36

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler

Training: Datensatz A4, 10% Toleranz, Testing: Datensatz A4

TP	TN	FP	FN	EP	EN	Threshold
66.59%	3.53%	10.69%	1.54%	2.87%	14.77%	0.61

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler, normalisiert

Training: Datensatz A4, 10% Toleranz, Testing: Datensatz A4

TP	TN	FP	FN	EP	EN	Threshold
76.60%	6.27%	5.49%	1.12%	3.47%	7.05%	0.61

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler, min. + max.

Training: Datensatz A4, 10% Toleranz, Testing: Datensatz A4

TP	TN	FP	FN	EP	EN	Threshold
81.20%	5.89%	4.76%	1.47%	2.15%	4.53%	0.56

Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler

Training: Datensatz A4, 10% Toleranz, Testing: Datensatz A4

TP	TN	FP	FN	EP	EN	Threshold
89.32%	7.02%	1.68%	0.21%	1.36%	0.42%	0.51

Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, normalisiert

Training: Datensatz A4, 10% Toleranz, Testing: Datensatz A4

TP	TN	FP	FN	EP	EN	Threshold
91.36%	3.02%	0.00%	5.61%	0.00%	0.00%	0.86

Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, min. + max.

Training: Datensatz A4, 10% Toleranz, Testing: Datensatz A4

TP	TN	FP	FN	EP	EN	Threshold
9.78%	0.23%	9.56%	0.23%	10.35%	69.85%	0.55

Signallaufzeiten, Entfernung, Winkel, Distanzfehler

Training: Datensatz A4, 10% Toleranz, Testing: Datensatz A4

TP	TN	FP	FN	EP	EN	Threshold
34.08%	7.76%	5.96%	0.00%	1.35%	50.84%	0.42

Signallaufzeiten, Entfernung, Winkel, Distanzfehler, normalisiert

Training: Datensatz A4, 10% Toleranz, Testing: Datensatz A4

TP	TN	FP	FN	EP	EN	Threshold
26.43%	2.08%	4.68%	0.31%	4.58%	61.91%	0.52

Signallaufzeiten, Entfernung, Winkel, Distanzfehler, min. + max.

Training: Datensatz A4, 10% Toleranz, Testing: Datensatz A4

TP	TN	FP	FN	EP	EN	Threshold
90.71%	8.24%	0.74%	0.21%	0.11%	0.00%	0.55

Signalstärken, Entfernung, Winkel, Distanzfehler

Training: Datensatz A4, 10% Toleranz, Testing: Datensatz A4

TP	TN	FP	FN	EP	EN	Threshold
90.68%	7.04%	0.00%	1.30%	0.98%	0.00%	0.62

Signalstärken, Entfernung, Winkel, Distanzfehler, normalisiert

Training: Datensatz A4, 10% Toleranz, Testing: Datensatz A4

TP	TN	FP	FN	EP	EN	Threshold
91.14%	7.28%	0.42%	1.16%	0.00%	0.00%	0.61

Signalstärken, Entfernung, Winkel, Distanzfehler, min. + max.

Training: Datensatz A4, 10% Toleranz, Testing: Datensatz A4

TP	TN	FP	FN	EP	EN	Threshold
0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	0.5

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler

Training: Datensatz B1, 10% Toleranz, Testing: Datensatz B1

TP	TN	FP	FN	EP	EN	Threshold
20.00%	0.00%	0.87%	0.00%	0.00%	79.13%	0.41

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, normalisiert

Training: Datensatz B1, 10% Toleranz, Testing: Datensatz B1

TP	TN	FP	FN	EP	EN	Threshold
0.87%	0.00%	0.00%	0.00%	0.00%	99.13%	0.45

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, min. + max.

Training: Datensatz B1, 10% Toleranz, Testing: Datensatz B1

TP	TN	FP	FN	EP	EN	Threshold
0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	0.50

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler

Training: Datensatz B1, 10% Toleranz, Testing: Datensatz B1

TP	TN	FP	FN	EP	EN	Threshold
0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	0.50

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler, normalisiert

Training: Datensatz B1, 10% Toleranz, Testing: Datensatz B1

TP	TN	FP	FN	EP	EN	Threshold
0.87%	0.00%	0.00%	0.00%	0.00%	99.13%	0.14

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler, min. + max.

Training: Datensatz B1, 10% Toleranz, Testing: Datensatz B1

TP	TN	FP	FN	EP	EN	Threshold
99.13%	0.00%	0.00%	0.00%	0.00%	0.87%	0.48

Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler

Training: Datensatz B1, 10% Toleranz, Testing: Datensatz B1

TP	TN	FP	FN	EP	EN	Threshold
96.52%	0.00%	2.61%	0.00%	0.00%	0.87%	0.47

Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, normalisiert

Training: Datensatz B1, 10% Toleranz, Testing: Datensatz B1

TP	TN	FP	FN	EP	EN	Threshold
100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.50

Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, min. + max.

Training: Datensatz B1, 10% Toleranz, Testing: Datensatz B1

TP	TN	FP	FN	EP	EN	Threshold
0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	0.50

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler

Training: Datensatz B2, 10% Toleranz, Testing: Datensatz B2

TP	TN	FP	FN	EP	EN	Threshold
16.10%	0.00%	5.93%	0.00%	0.00%	77.97%	0.29

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, normalisiert

Training: Datensatz B2, 10% Toleranz, Testing: Datensatz B2

TP	TN	FP	FN	EP	EN	Threshold
2.54%	0.00%	0.00%	0.00%	0.00%	97.46%	0.30

Signallaufzeiten, Signalstärken, Entfernung, Winkel, Temperatur, Distanzfehler, min. + max.

Training: Datensatz B2, 10% Toleranz, Testing: Datensatz B2

TP	TN	FP	FN	EP	EN	Threshold
11.02%	0.00%	1.69%	0.00%	0.00%	87.29%	0.33

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler, normalisiert

Training: Datensatz B2, 10% Toleranz, Testing: Datensatz B2

TP	TN	FP	FN	EP	EN	Threshold
0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	0.50

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler, min. + max.

Training: Datensatz B2, 10% Toleranz, Testing: Datensatz B2

TP	TN	FP	FN	EP	EN	Threshold
99.15%	0.00%	0.85%	0.00%	0.00%	0.00%	0.49

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler

Training: Datensatz B2, 10% Toleranz, Testing: Datensatz B2

TP	TN	FP	FN	EP	EN	Threshold
100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.49

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler, normalisiert

Training: Datensatz B2, 10% Toleranz, Testing: Datensatz B2

TP	TN	FP	FN	EP	EN	Threshold
99.15%	0.00%	0.00%	0.00%	0.00%	0.85%	0.50

Signallaufzeiten, Entfernung, Winkel, Temperatur, Distanzfehler, min. + max.

Training: Datensatz B2, 10% Toleranz, Testing: Datensatz B2



## B. Anhang - Matlab

Die bei der Erstellung dieser Studienarbeit angefallenen Skripte und Daten finden sich auf einem beiliegenden Datenträger.

Für diese Arbeit geschriebene Matlab-Skripte:

acq\_robot.m - Auslesen der Daten vom Roboter.

annotate\_robot\_data.m - Annotiert die Daten die vom Roboter eingelesen wurden.

batch\_calc\_multilat.m - Berechnet ausgehend von mindestens 3 bekannten Knoten die Position eines vierten Knotens.

batch\_call\_train\_data.m - Ruft call\_train\_data für alle Dateien in einem Verzeichnis auf.

batch\_count\_tp\_tn\_fp\_fn - Zählt Datensätze nach der Separierbarkeit aus.

batch\_sparse\_plot\_evalfis\_output\_plot\_only.m - Ruft plot\_evalfis\_output\_plot\_only mit einem Bruchteil der Daten auf.

calc\_multilat.m - Berechnet die Multilateration für übergebene Knotenmenge.

call\_train\_data.m - Ruft train\_data auf und speichert das Ergebnis in eine Datei.

compare\_multlat\_results.m - Vergleicht die multilateration Ergebnisse der ungewichteten, gefilterten und gewichteten Varianten.

crossover\_evalfis.m - Ruft eval\_data mit unterschiedlichen Datensätzen für Training und Testing auf.

degree\_ground\_truth.m - Berechnet die ground truth für den Datensatz aus Paris.

drawclustering.m - Zeichnet eine Grafik zur Veranschaulichung diskreter Werte beim Clustering.

draw\_multilateration.m - Visualisiert die Ergebnisse der Multilateration.

eval\_data.m - Wertet einen zuvor gespeicherten Datensatz aus.

generate\_variants.m - Erzeugt verschiedene Varianten der Datensätze.

get\_nodes.m - Extrahiert Knoten für die Multilateration.

get\_quality\_from\_nodes.m - Extrahiert aus der Matrix der Knoten die jeweiligen Qualitätswerte.

get\_training\_sets.m - Erzeugt Datensätze für das Training des FIS.

load\_data.m - Erzeugt Datensätze mit verschiedenen Parametern aus den Messwerten.

plot\_evalfis\_output.m - Zeichnet einen ausgewerteten Datensatz und bestimmt die Anzahl richtiger und falscher Werte.

plot\_evalfis\_output\_plot\_only.m - Zeichnet einen ausgewerteten Datensatz.  
remove\_duplicates.m - Entfernt doppelte Einträge aus Datensätzen.  
sbs\_relate\_acq\_raw.m - Liest Daten von zwei Knoten (Sender + Empfänger) ein.  
SendMovePacket.m - Schickt einen Befehl zum Bewegen an den Roboter.  
split\_variants.m - Teilt einen Datensatz in Teilbereiche der Entfernungen auf.  
train\_data.m - Führt das Training eines Datensatzes mit Testing- und optional Checking-Datensatz durch.  
transform\_measurements\_2\_nodes.m - Extrahiert aus den Paris-Messungen die Knotenpositionen.

Verzeichnisstruktur der Daten:

automated - Vom Roboter erzeugte Messwerte.  
automated\_loading - Datensätze in verschiedenen Merkmals-Varianten.  
manuelle Serie - Manuell erzeugte Datensätze.  
Paris - Datensätze aus Paris.  
Robot - Aus den Robotermesswerten erzeugte Datensätze.  
training\_sets - Fürs Training aufbereitete Datensätze.



# Literatur

- [Ashe97] R. Asher. *Ultrasonic Sensors for Chemical Process Plant*. Institute Of Physics Publishing. 1997.
- [BDRZ<sup>+</sup>07] Martin Berchtold, Christian Decker, Till Riedel, Tobias Zimmer und Michael Beigl. Using a Context Quality Measure for Improving Smart Appliances. In *ICDCSW '07: Proceedings of the 27th International Conference on Distributed Computing Systems Workshops*, Washington, DC, USA, 2007. IEEE Computer Society, S. 52.
- [Berc07] Martin Berchtold. Qualitätsmanagement von Kontexten Verlässlichkeit mit Fuzzy Logik. Diplomarbeit, Universität Karlsruhe (TH), Januar 2007.
- [Beus07] Monty Beuster. The RELATE Brick System - Tutorial, November 2007.
- [BRDB<sup>+</sup>08] Martin Berchtold, Till Riedel, Christian Decker, Christian Bittel, Michael Beigl und Monty Beuster. Quality of Location: Estimation, System Integration and Application. Submitted to the „Fifth International Conference on Networked Sensing Systems“, 2008.
- [Chiu94] Stephen L. Chiu. Fuzzy Model Identification based on Cluster Estimation. *Journal of Intelligent and Fuzzy Systems*, Band 2, 1994, S. 267–278.
- [DKBZ05] Christian Decker, Albert Krohn, Michael Beigl und Tobias Zimmer. The particle computer system. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, Piscataway, NJ, USA, 2005. IEEE Press, S. 62.
- [EvDH04] L. Evers, S. O. Dulman und P. J. M. Havinga. A Distributed Precision Based Localization Algorithm for Ad-Hoc Networks. In *proceeding of the Second International Conference on Pervasive Computing, PERVASIVE 2004, Linz/Vienna, Austria*, Band Volume 3001/2004 der *Lecture Notes in Computer Science*, Berlin, March 2004. Springer Verlag, S. 269–286.
- [Harr66] Cyril M. Harris. Absorption of Sound in Air versus Humidity and Temperature. *Journal of the Acoustical Society of America*, Band 40, 1966, S. 148–159.
- [HiBo01] Jeffrey Hightower und Gaetano Borriella. Location Systems for Ubiquitous Computing. *IEEE Computer*, 34(8), 2001, S. 57–66.

- [HLIN01] Marko Helén, Juha Latvala, Hannu Ikonen und Jarkko Niittytahti. Using Calibration in RSSI-Based Location Tracking System. In *The 5th World Multiconference on Circuits, Systems, Communications and Computers (CSCC2001)*, Juli 2001.
- [Jang93] Jyh-Shing Roger Jang. ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems, Man, and Cybernetics*, Band 23, 1993, S. 665–684.
- [Kalm60] Rudolf Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D), 1960, S. 35–45.
- [King98] A. D. King. Inertial Navigation - Forty Years of Evolution. *General Electric Company Review*, 13(3), 1998, S. 140–149.
- [KrHB07] Albert Krohn, Mike Hazas und Michael Beigl. Removing Systematic Error in Node Localisation Using Scalable Data Fusion. In Langendoen und Voigt [LaVo07], S. 341–356.
- [Krum03] John Krumm. Probabilistic Inferencing for Location, Oktober 2003.
- [LaVo07] Koen Langendoen und Thiemo Voigt (Hrsg.). *Wireless Sensor Networks, 4th European Conference, EWSN 2007, Delft, The Netherlands, January 29-31, 2007, Proceedings*. Band 4373 der *Lecture Notes in Computer Science*, Springer, 2007.
- [MacQ76] J. B. MacQueen. Some Methods for classification and Analysis of Multivariate Observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, S. 281–297. University of California Press, 1976.
- [MJGG<sup>+</sup>98] Duane L. Mattern, Link C. Jaw, Ten-Huei Guo, Ronald Graham und William McCoy. Using Neural Networks for Sensor Validation, 1998.
- [MuMR06] Henk L. Muller, Michael McCarthy und Cliff Randell. Particle Filters for Position Sensing with Asynchronous Ultrasonic Beacons. In *Proceedings of LoCA 2006*, S. 1–13. Springer Verlag, Mai 2006.
- [NNMM<sup>+</sup>07] Yoshiyuki Nakamura, Yuko Namimatsu, Nobuo Miyazaki, Yutaka Matsuo und Takuichi Nishimura. A Method for Estimating Position and Orientation with a Topological Approach using Multiple Infra-red Tags. In *Networked Sensing Systems, 2007. INSS apos;07. Fourth International Conference on*, Juni 2007, S. 187–195.
- [RaDM03] C. Randell, C. Djallis und H. Muller. Personal position measurement using dead reckoning, 2003.
- [ScBG99] Albrecht Schmidt, Michael Beigl und Hans-W. Gellersen. There is more to context than location. *Computers and Graphics*, 23(6), 1999, S. 893–901.

- 
- [SuKa88] M. Sugeno und G. T. Kang. Structure identification of fuzzy model. *Fuzzy Sets Syst.*, 28(1), 1988, S. 15–33.
- [TaSu83] T. Takagi und M. Sugeno. Derivation of fuzzy control rules from human operator’s control actions. In *Proc. of the IFAC Symp. on Fuzzy Information, Knowledge Representation and Decision Analysis*, Juli 1983, S. 55–60.
- [Ward98] Andrew Martin Robert Ward. *Sensor-driven Computing*. Dissertation, University of Cambridge, August 1998.
- [WeBi01] Greg Welch und Gary Bishop. *An Introduction to the Kalman Filter*, 2001.
- [WHFG92] Roy Want, Andy Hopper, Veronica Falcao und Jon Gibbons. The Active Badge Location System. Technischer Bericht 92.1, Olivetti Research Ltd (ORL), ORL, 24a Trumpington Street, Cambridge CB2 1QA, 1992.
- [Zogg06] Jean Marie Zogg. *Grundlagen der Satellitennavigation*. Online-Publikation. 2006.

