

Hochschule für Technik, Wirtschaft und Kultur Leipzig
Fakultät Informatik, Mathematik und Naturwissenschaften



Bachelorarbeit

Ultraschall-Hinderniserkennung im Roboterfußball

Richard Stiller

richard.nao.htwk@gmail.com

26. September 2014

Matrikelnummer: 50403

Gutachter: Rico Tilgner, M. Sc.

Betreuer: Prof. Dr. Sibylle Schwarz

Selbstständigkeitserklärung

Ich erkläre, dass ich die vorliegende Arbeit selbständig, ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen (einschließlich elektronischer Quellen) direkt oder indirekt übernommenen Gedanken sind ausnahmslos als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

Leipzig, den 26. September 2014

Danksagung

Zunächst möchte ich mich an dieser Stelle bei all denjenigen bedanken, die mich während der Anfertigung dieser Bachelorarbeit unterstützt und motiviert haben - vor allem meine Freunde, die ihr zeitliche und personelle Einbußen meinerseits auf euch genommen habt.

Mein besonderer Dank gilt meiner Betreuerin Frau Professorin Sibylle Schwarz. Mit ihrem kritischen Hinterfragen zu inhaltlichen sowie strukturellen Themen gaben sie mir stets wertvolle Hinweise. Diese stellten eine Bereicherung für mich dar und werden mir auch in späteren Arbeiten weiterhelfen.

Weiterhin gilt mein Dank Rico Tilgner, dem teaminternen Betreuer des Nao-Teams der HTWK und dem Nao-Team selbst. Ihr, als Team, habt mich herzlich aufgenommen, mich gut in die Gruppe integriert und mir wundervolle Erlebnisse geschenkt. Ich bin froh und dankbar, ein Mitglied eures Teams zu sein.

Der Weg dahin war nicht leicht und ihr habt mich immer wieder vor Herausforderungen gestellt. Doch standet ihr mir gerne zur Seite, wenn ich schwierige Themen bewältigen musste.

All meinen Korrekturlesern möchte ich danken. Eure aufgewendete Zeit wird von mir nicht vergessen.

Nicht zuletzt gebührt meinen Eltern Dank, dass ihr mich nicht nur während der Schulzeit sondern auch während des Studiums so herzlich unterstützt.

Abstract

Ziel des Roboterfußballs mit dem NAO ist nicht primär die Leidenschaft für den Fußball als Sport, sondern es ist der Austausch von Entwicklungen in der Robotik. Speziell behandelt diese Arbeit die Möglichkeit der Hinderniswahrnehmung durch den Ultraschallsensor des NAOs. Folgende Fragen werden in diesem Dokument behandelt: Welche Sendebereiche deckt der Sensor ab? Liefert der Ultraschall immer sinnvolle Daten? Beeinflusst das Trikot vor dem Sensor die Wahrnehmung? Bei den Untersuchungen werden auf die Ansätze des Bremer NAO Teams B-Human und die Auswertungen empirisch erhobener Daten zurückgegriffen, welche neue Erkenntnisse über den Sendebereich liefern, das Trikot als einen Einfluss auf den Sensor ausschließt und auf weitere problematische Hindernisse hinweist. Letztendlich stellt der Ultraschall ein geeignetes Mittel dar, um am Spielgeschehen erfolgreich mitzuwirken.

Inhaltsverzeichnis

Inhaltsverzeichnis	VI
Abbildungsverzeichnis	VIII
Tabellenverzeichnis	IX
Listings	X
Begriffsverzeichnis	XI
1. Einleitung	1
1.1 Motivation	1
1.2 Anforderungen an die Arbeit	2
1.3 RoboCup	2
1.4 Standard Platform League	3
1.5 Hinderniserkennung durch Ultraschall	4
2. Grundlagen	5
2.1 Der NAO	5
2.2 Ultraschall	6
2.3 Ultraschallsensoren (US-Sensoren)	7
2.3.1 Rückgabewerte	8
2.3.2 Sensor-Modi	9
2.4 Trikot	11
2.5 Aufbau und Integration in die bestehende Software	12
2.5.1 NAOqi (Aldebaran API)	12
2.5.2 Bridge	12
2.5.3 Firmware	12
2.5.4 USTest	13
2.5.5 NaoControl	13
3. Mögliche Verfahren der Hinderniserkennung	15
3.1 Triangulierung durch Schnittpunktberechnung	15
3.1.1 Triangulierung über den Kosinussatz	15
3.1.2 Triangulierung über Kreisschnittpunkte	16
3.1.3 Erkenntnisse	18

3.2	Rasterungsansatz von B-Human	18
3.2.1	Erkenntnisse	19
4.	Allgemeine Informationen für Testsznarien	21
4.1	Positionen der Messungen auf dem Spielfeld	21
4.2	Allgemeiner Aufbau einer Testsoftware	22
4.3	Messergebnisse und Dateinamen	23
4.3.1	Dateinamenkonvention	23
4.3.2	Formatierung des Messergebnisses	24
5.	Aufbau der Testumgebungen und Durchführung der Tests	26
5.1	Testumgebung: leeren Feldes	26
5.1.1	Typische Testfälle	26
5.1.2	Verwendete Software	27
5.1.3	Durchführung und Messergebnisse	28
5.1.4	Auswertung	28
5.2	Test zur Überprüfung der Reichweite eines unbewegten Hindernisses in einem Raster (Rastertest)	30
5.2.1	Typische Testfälle	30
5.2.2	Vorbereitung des Feldes	32
5.2.3	Verwendete Software	33
5.2.4	Durchführung und Auswertung	34
6.	Ausblick und Fazit	38
	Literaturverzeichnis	39

Abbildungsverzeichnis

1.1	NAOs im Spiel	3
2.1	Ansicht eines handelsüblichen NAOs	5
2.2	Schema von Ultraschall mit 2 Hindernissen	6
2.3	Positionen der Ultraschallsensoren	7
2.4	Idealer Öffnungswinkel bei 90 Grad	10
2.5	Trikots der NAOs	11
2.6	Allgemeiner Aufbau der Software	14
3.1	Schema für die Dreiecksberechnung mit dem Kosinussatz	16
3.2	USTest mit Beispielen zur Triangulierung über Kreisschnittpunkte	17
3.3	Aufbau eines dreidimensionalen Rasters	19
3.4	Bilder zum Modell von B-Human	20
4.1	Ausgangspositionen für Messungen auf einen 6x4 Spielfeld	21
5.1	Vorbereitungen des Tests <i>leeres_feld</i>	27
5.2	Die Position des NAOs vor dem Raster im Messbereich	30
5.3	Bilder zur Vorbereitung des Feldes	32
5.4	Umsetzung des Tests und Legende	35
5.5	Visualisierung des Rasters für Modus 4 linker Sensor	36
5.6	Visualisierung des Rasters für Modus 5 linker Sensor	36
5.7	Visualisierung des Rasters für Modus 6 linker Sensor	36
5.8	Visualisierung des Rasters für Modus 7 linker Sensor	37
5.9	Visualisierung der gemeinsamen Bereiche für Modus 4	37
5.10	Visualisierung der gemeinsamen Bereiche für Modus 5	37

Tabellenverzeichnis

2.1	Relative Positionen und Winkel der Ultraschallsensoren	7
2.2	Tabellenausschnitt über die Nutzung des rechten Puffers im einseitigen Modus 0	8
2.3	Werte für die Ansteuerung einseitiger Modi	9
2.4	Werte für die Ansteuerung sequentieller beidseitiger Modi	10
5.1	Abweichungen im Modus 6 bei den Messungen 3 und 4	28
5.2	Computergenerierte Analyse zu <i>Tabelle 12 - hybrid.pos_3.no_jersey.empty_field.0.0.txt</i>	29

Listings

1	Einmaliges Absetzen des Modi 4 mit NAOqi	11
2	<i>struct ultrasound_t</i> für die Rückgabe der Bridge an die Firmware	12
3	Allgemeiner Aufbau der Testsoftware mit 10 Messungen je Modus	22
4	Beispieldaten zur Formatierung eines Messergebnisses	24
5	Allgemeiner Aufbau für spaltenweise Messungen im Rastertest	33
6	Ausgabeoperation und Ausgabebeispiel im Rastertest	34
7	Spiegeln der Werte vom linken zum rechten Messbereich in der Auswertungssoftware	34

Begriffsverzeichnis

Folgende Begriffe und Abkürzungen werden in diesem Dokument verwendet:

NAO	Roboter vom Typ Nao der Firma Aldebaran
SPL	Standard Platform League
US	Abkürzung Ultraschall
US-Sensor	Abkürzung Ultraschallsensor
Transmitter	sendender Ultraschallsensor
Receiver	empfangender Ultraschallsensor
Echo	empfangene Reflexion des Schalls nach dem Aussenden eines Impulses
Nao-Team HTWK	Team der Hochschule für Technik, Wirtschaft und Kultur Leipzig in der Standard Platform League
Hybrid, Helo	Namen der NAOs im Nao-Team HTWK

1 Einleitung

1.1 Motivation

Diese Arbeit erforscht die Nutzungsmöglichkeiten der Ultraschallsensoren (US-Sensoren) des NAOs und die Entwicklung von Lösungsansätzen zur Erkennung von Hindernissen mit Hilfe dieser Sensoren. Der NAO wird aktuell ohne den Ultraschall vom Nao-Team HTWK im Roboterfußball in der Standard Platform League (SPL) eingesetzt.

„Beim Fußballspiel mit humanoiden Robotern geht es nicht nur um die möglichst schnelle Selbstlokalisierung des Roboters auf dem Feld, sondern auch um die Echtzeit-Hinderniserkennung: Wer schneller erkennt, wo die gegnerischen Spieler stehen, wo sich der Ball befindet und wo der Schiedsrichter steht, kann auch schneller reagieren und schießt eher ein Tor. Die Roboter sollen Hindernisse erkennen, um sie anschließend ohne Kollision zu umlaufen.“ ([Forschung2012])

Dieses Zitat beschrieb 2012 den damaligen Stand der visuellen Hinderniserkennung und die Schwierigkeiten zu diesem Problem. Die visuelle Wahrnehmung der Roboter hat sich seitdem verbessert, dennoch ist das Problem der Echtzeit-Hinderniserkennung weiterhin vorhanden. Im Wandel von verschärften Regeln und der Zunahme der Bedeutung von Strategien und Taktiken bei der SPL kommt dieser Rubrik ein größerer Stellenwert zu. Hierfür braucht man eine geeignete Erkennung von Robotern und Hindernissen, um aktives Spielgeschehen zu erfassen und bestmöglich zu nutzen.

Meine Aufgabe ist es, den Ultraschallsensor als möglichen Zusatz zu untersuchen und Lösungsansätze zu erforschen.

1.2 Anforderungen an die Arbeit

Aufgrund der bestehenden Probleme ist das angestrebte Ziel, eine Hinderniserkennung zu evaluieren und in der vorhandenen Software umzusetzen.

Ausgehend von den vorhanden Bedingungen und der Software sind folgende Teilabschnitte umzusetzen, um ein optimales Ergebnis zu erzielen. Das Anlegen einer Grundlagenbasis über den Sensor und dessen Messdaten ist der Ausgangspunkt. Aufbauend darauf sind die Problemstellen des US-Sensors klar zu definieren und Lösungsansätze zu finden. Weiterhin sind Hinderniserkennungen durch verschiedene Methoden und experimentelle Messungen herauszuarbeiten. Abschließend werden die Problemstellungen und Hinderniserkennungen durch die erarbeiteten Ergebnissen evaluiert.

1.3 RoboCup

Auf seiner Homepage wird der RoboCup mit folgenden Worten beschrieben:

„RoboCup is an international scientific initiative with the goal to advance the state of the art of intelligent robots.“ ([Robocup, about-robocup])

Ebenso ist der *RoboCup* der Name des Events, bei dem sich zahlreiche Wissenschaftler und Forscher verschiedener Institutionen aus aller Welt treffen. Dort tauschen sie ihre Fortschritte in Robotik und Künstlicher Intelligenz (KI) aus. Es finden Wettbewerbe in Form von Spielen (Abbildung 1.1a) und Symposien statt. Zum RoboCup zählen ebenso lokale Veranstaltungen wie die *German Open* in Magdeburg.

Mit dem Ziel im Jahr 2050 mit Robotern gegen den amtierenden FIFA Weltmeisterspieler zu lassen, wurden verschiedenste Ligen gegründet oder integriert. Der RoboCup wird in folgende Themen gegliedert:

RoboCupJunior Ein Wettbewerb für Schüler, die sich mit dem Thema Robotik auseinandersetzen wollen

RoboCup@Home Entwicklung von Haushaltsrobotern, um Personen im täglichen Leben zu unterstützen

RoboCupRescue Entwicklung von Rettungsrobotern, um im Krisengebieten auszuhelfen

RoboCupSoccer Autonom und kooperativ spielende Roboter mit Fußball als Sportart

Viele dieser Themen sind noch in Ligen mit unterschiedlichen Anforderungen und Regeln unterteilt, wie z.B. die Standard Platform League (Abschnitt 1.4), welche ein Teil des RoboCupSoccer ist.

Es stellt sich die Frage, warum Fußball im RoboCupSoccer als Forschungsumgebung geeignet ist. Er hat einige Vorteile wie übersichtliche Spielregeln. Der Fußball ist zudem sehr populär. Faktoren in der Umwelt, wie Licht und Bodenverhältnisse, spielen im Roboterfußball ebenso eine Rolle. So kann die kleinste nicht erkannte Abweichungen der Position des Balls zu Einbußen in der Präzision eines Schusses führen. Es entstehen immer neue Spielsituation, die zu bewältigen sind. Im Blick auf andere Sportarten ist es möglich, Fußball in nur zwei Dimensionen (auf dem Fußboden ohne hohe Schüsse) zu spielen.

1.4 Standard Platform League



(a) Spielsituation HTWK vs. TJArk (RoboCup/SPL, Brasilien 2014)



(b) Tackling zweier NAOs

Abbildung 1.1: NAOs im Spiel (Quelle: Nao-Team HTWK)

Die Standard Platform League (SPL) ist seit 1999 ([RobocupWiki, Standard Platform League]) teil des RoboCups. Ursprünglich wurde auf der Hardwareplattform AIBO, dem Roboterhund (siehe <http://www.sony-aibo.co.uk/>) gespielt. Dieser wurde 2008 durch die Plattform NAO der Firma Aldebaran (siehe <http://www.aldebaran.com/en/humanoid-robot/nao-robot>) ersetzt.

Die SPL zeichnet sich darin aus, dass die gleiche Hardwareplattform von allen Teams verwendet wird und Änderungen an der Hardware nach den Regeln ([Rules2014]) verboten sind. Die erzielten Erfolge lassen sich nur durch die geschriebene Software gewinnen und so kann man in der Liga von einem Kampf der Algorithmen sprechen.

Gegenüber den Simulationsligen, die einen virtuellen NAO zum Spielen nutzen, wird in der SPL auf reale Umweltbedingungen und Hardware aufgebaut, welche ebenso Probleme ins Spiel bringen können.

1 Einleitung

Die SPL stellt eine der größeren Ligen im RoboCup. Im Jahr 2013 in Eindhoven (Holland) nahmen 171 Teilnehmer in 22 Teams ([Final2013, S. 19]) allein in dieser Liga zur Weltmeisterschaft teil. Daten zum Jahr 2011/4 liegen zum Zeitpunkt der Arbeit nicht vor.

1.5 Hinderniserkennung durch Ultraschall

Wie schon in der Motivation beschrieben, ist die Wahrnehmung der Umgebung wichtig, um im Spielgeschehen ballführend zu sein. Bereiche wie Passspiel, Pfadfindung und Tackling, einem robusten aber fairen Abnehmen des Balls vom Gegenspieler, sind weiterführende Themen, die auf Hinderniserkennung aufbauen oder durch diese verbessert werden können.

Dem NAO stehen eine Vielzahl an Sensoren zur Verfügung. Vom Nao-Team HTWK wird zur aktiven Erfassung der Umgebung aktuell jedoch nur die visuelle Wahrnehmung via Kamera eingesetzt. Dies führt zu Schwächen, welche möglicherweise durch den Ultraschall ausgebessert werden können.

Im Fall der visuellen Erkennung der Position von gegnerischen Spielern als Hindernisse nutzt das Team die Füße der Gegner, um deren Abstand zu sich selbst zu bestimmen. Dieses Verfahren funktioniert auf Distanzen von 25cm bis 1,20m schlecht und führt zu keinen Ergebnissen, da primär andere Bauteile des Roboters visuell wahrgenommen werden. Hindernisse wie Beine herum eilender Schiedsrichter und liegende NAOs sind in dem Verfahren nicht beachtet. Ein weiterer Fall für Verbesserungen ist das Tackling. In den Regeln der SPL (siehe Abschnitt 1.4) wurde die Strafe für Fouls beim Tackling (Player Pushing) seit 2014 verschärft und kann zum Platzverweis für Roboter führen, siehe [Rules2014, S. 25].

Der Vorteil der US-Sensoren ist, dass er laut Hersteller einen breiteren Sichtbereich vor dem NAO hat als die Kamera. Die Kamera kann dies nur durch Kopfbewegungen ausgleichen. Doch in schnellen und stressigen Spielzügen ist es wichtig, sich nicht immer umsehen zu müssen, sondern sich eher auf den Ball und das Torschießen zu konzentrieren. Im Zusammenspiel von Kamera im Kopf und US-Sensoren (Abschnitt 2.3) im Körper könnten noch breitere Bereiche des Spielfelds abgedeckt werden.

Der Ultraschall scheint für diese Probleme eine geeignete Lösung zu sein. Ob der Ultraschall die geeignete Lösung ist, wird in diesem Dokument herausgearbeitet.

2 Grundlagen

Die Grundlagen des Ultraschalls, des NAOs und seinen Bestandteilen, besonders die der Ultraschallsensoren werden in diesem Kapitel erklärt. Zudem wird die Einbettung der Software in die bestehende Architektur beschrieben.

2.1 Der NAO

Der Roboter NAO ist ein Produkt der Firma Aldebaran (siehe <http://www.aldebaran.com/en>) aus Paris (Frankreich). Mit seinen 58cm und seinen 4,3kg gehört er zu den mittelgroßen Robotern im Robocup (siehe Abschnitt 1.3).



Abbildung 2.1: Ansicht eines handelsüblichen NAOs (Quelle: Nao-Team HTWK)

Die äußere Erscheinung (Abbildung 2.1) ist einem Menschen nachempfunden und vom Körperbau mit einem Kleinkind vergleichbar. Der Körper ist aber nicht perfekt abgebildet. So besitzt er unter anderem nur 3 Finger und keine Ballen an den Füßen.

Aktuell existieren 3 fußballtaugliche Versionen des NAOs: v3, v4 und v5. Die Unterschiede im Erscheinungsbild sind minimal, dagegen in der Ausstattung der Hardware

2 Grundlagen

erheblich. Von Version v4 auf v5 wurde der verbaute WLAN-Adapter von 2,4GHz auf 5GHz aufgerüstet und stabilere Zahnräder verbaut. Zusätzlich existieren verschiedene Editionen. Die Robocup Edition v4 besitzt beispielsweise nur 21 statt 25 Freiheitsgrade (Gelenke), hat weniger Mikrofone verbaut und ist mit einer grauen Abdeckungen statt mit einer blauen (Abbildung 2.1) ausgestattet. Ein NAO mit 25 Freiheitsgraden wird H25 bezeichnet.

Die Hardware besteht aus Sensoren, Servomotoren und Computerhardware in Form von RAM, Speicher und CPU.

Die Sensorenausstattung besteht u.a. aus Kamera, Infrarot-Sensor, Accelerometer (Beschleunigungssensor), Gyrometer (Drehbewegungssensor) und Ultraschallsensoren (siehe Abschnitt 2.3).

Weitere Details sind unter [AldV2.1, index_h25.html] nachzulesen.

Ausgeliefert wird er mit einem angepassten Linux Betriebssystem und einer Programmierschnittstelle (API) dem NAOqi für die Entwicklung von Software (siehe Abschnitt 2.5.1). Unterstützte Programmiersprachen sind C++, Python und Java.

2.2 Ultraschall

Der Ultraschall ist ein Schall ab 16kHz und ist nicht für den Menschen hörbar. In der Luft bei 20 Grad besitzt er eine Geschwindigkeit von $343 \frac{m}{s}$. Für die Messung mit Ultraschall wird die Reflexion vom Schall an festen Hindernissen genutzt und über die Zeitdifferenz wird der Abstand errechnet.

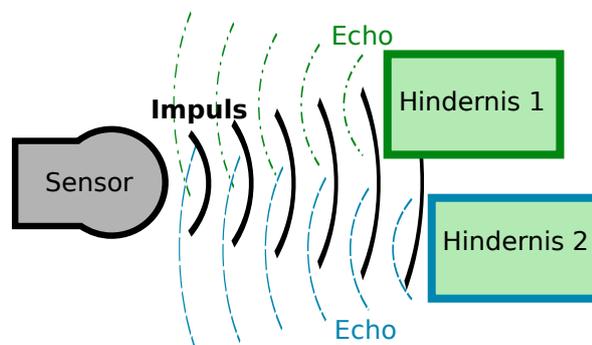


Abbildung 2.2: Schema von Ultraschall mit 2 Hindernissen

Abbildung 2.2 stellt ein Schema zu der Funktionsweise vom Ultraschall dar. In diesem wird ein Impuls (Signal) ausgesendet und als Echo von Hindernis 1 und 2 reflektiert. Die Anzahl der Echos hängt im Realversuch von der Anzahl der Hindernisse und von dem Winkel der Reflexion ab, welcher im Schema nicht beachtet wird.

2.3 Ultraschallsensoren (US-Sensoren)

Der NAO besitzt jeweils 2 Ultraschallsensoren (US-Sensoren) in der Brust, unterteilt in 1 Transmitter (Sender) und 1 Receiver (Empfänger).

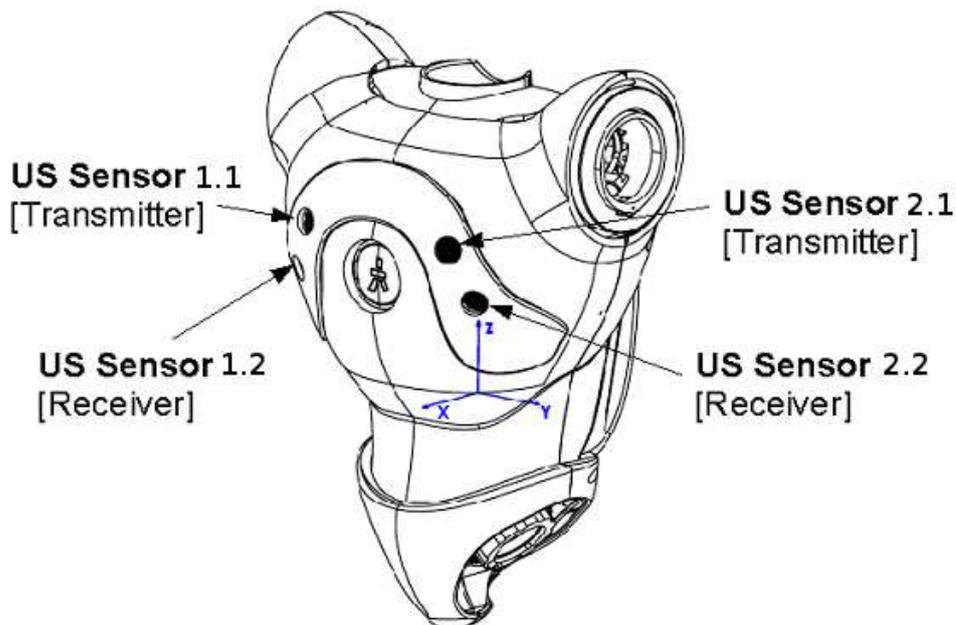


Abbildung 2.3: Positionen der US-Sensoren (Quelle: [AldV2.1, sonar_robot.html, vom Autor geändert])

Gesendet wird in einer Frequenz von 40kHz. Der vom Hersteller beschriebene messbare Winkel beträgt 60 Grad (effective cone [AirUltra]). Die Fehlertoleranz liegt laut Aldebaran bei 1cm und der Empfangsbereich umfasst den Bereich von 0,25m bis 2,55m für NAO v3 und v4. Für NAO v5 wird in einer Entfernung von 0,01m bis 3m eine distanzabhängige Fehlertoleranz von 1cm bis 5cm angegeben. Im Originaldokument spricht Aldebaran von 4 Sensoren. In diesem Dokument wird Sensor 1 und 2 in Sensor 1.1 und 1.2 und Sensor 3 und 4 in 2.1 und 2.2 zusammengefasst und umbenannt (siehe Abbildung 2.3).

Seite	Sensorname	X(m)	Y(m)	Z(m)	X(°)	Y(°)	Z(°)
Rechts	US-Sensor 1.1	0,0537	-0,0341	0,0698	0	-10	-20
	US-Sensor 1.2	0,0477	-0,0416	0,0509	0	15	-25
Links	US-Sensor 2.1	0,0537	0,0341	0,0698	0	-10	-20
	US-Sensor 2.2	0,0477	0,0416	0,0509	0	15	-25

Tabelle 2.1: Relative Positionen und Winkel der US-Sensoren im Körper des NAOs (vgl. [AldV2.1, sonar_robot.html])

2 Grundlagen

Tabelle 2.1 beschreibt die US-Sensoren im Körper des NAOs, Ausgangspunkt für die Positionen ist der „Toros Frame“ (TF). Bezierend auf [AldV2.1, masses_robot.html] können die Werte aus Gleichung 2.3 genutzt werden, um die Höhe der Sensoren im stehenden Roboter zu berechnen.

$$\begin{bmatrix} X_{TF} \\ Y_{TF} \\ Z_{TF} \end{bmatrix} = \begin{bmatrix} -0,00413 \\ 0 \\ 0,04342 \end{bmatrix} \quad (2.1)$$

Softwareintern können die Werte über die Funktion des NAOqi (Abschnitt 2.5.1) „AL-MotionProxy::positionInterpolations“ ([AldV2.1]) abgerufen werden.

2.3.1 Rückgabewerte

In einer Messung wird eine Gleitpunktzahl als Distanz zum Hindernis in Metern zurückgegeben. Wird kein Hindernis wahrgenommen, ist der Wert 2,55m die Rückgabe. Im minimalen Empfangsbereich von 25cm, wird einmalig der Wert 0,25m ausgegeben, egal wie viele Hindernisse in diesem Bereich gemessen werden. Es können bis zu 10 Echos (Abschnitt 2.2) auftreten. Die Werte sind monoton steigend.

Unterscheidungen bei der Rückgabe sind zwischen einseitigen und beidseitigen Modi (Abschnitt 2.3.2) zu treffen.

Im beidseitigen Modus werden die Werte im Pfad *Device/SubDeviceList/US/Left/Sensor/* für den linken Sensor (Sensor 2 , vgl. Abbildung 2.3) und *Device/SubDeviceList/US/Right/Sensor/* für den rechten Sensor (Sensor 1 , vgl. Abbildung 2.3) abgelegt. Die Datei *Value* (unter dem Pfad) enthält das erste Echo, alle weiteren Echos sind in *Value1* bis *Value9* gespeichert. Um das 3. Echo *Value2* der rechten Seite abzurufen, muss man folgenden Pfad abfragen: *Device/SubDeviceList/US/Right/Sensor/Value2*.

Im einseitigen Modus wird laut Dokumentation nur das *Device/SubDeviceList/US/Sensor/Value* gefüllt. In der Praxis hat sich gezeigt, dass *Sensor/Value* eine Referenz auf *Device/SubDeviceList/US/Right/Sensor/Value* ist. Weitere Echos wurden ebenfalls gemessen, wie Tabelle 2.2 zeigt.

Modus	Messung	...	r0	r1	r2
		...			
0	3	...	0.42	0.51	2.19
		...			

Tabelle 2.2: Tabellenausschnitt über die Nutzung des rechten Puffers im einseitigen Modus 0 mit r0 = Value, r1 = Value1 und r2 = Value2, weitere Information unter [Test:matrix-0-down, Tabelle 2]

In C++ sind die Daten mit der NAOqi-Funktion (Abschnitt 2.5.1) „ALMemoryProxy::getData“ ([AldV2.1]) auszulesen.

2.3.2 Sensor-Modi

Unterschieden wird bei den Modi grundlegend, ob ein einseitiger Impuls (einseitiger Modus) oder zwei sequentielle (beidseitiger Modus) Impulse abgegeben werden sollen.

Im einseitigen Modus wird nur ein Transmitter verwendet. Eine genaue Aufschlüsselung der Möglichkeiten ist in Tabelle 2.3 enthalten. Zur Erklärung: Wert 1 bedeutet, der Transmitter der linken Seite sendet das Signal und der Receiver der rechten Seite empfängt die Daten. Abbildung 2.4 visualisiert die Modi.

Wert	Transmitter (TM)	Receiver (REC)	Binär
0	Links	Links	00
1	Links	Rechts	01
2	Rechts	Links	10
3	Rechts	Rechts	11

Legende:

Wert, welcher an NAOqi (Abschnitt 2.5.1) übergeben wird

Binär Jeweiliger Binärcode der Zahl in Wert

Transmitter/Receiver der jeweiligen Seite (siehe Abschnitt 2.3)

Tabelle 2.3: Werte für die Ansteuerung einseitiger Modi (siehe [AldV2.1, actuator_sensor_names.html])

Im Binärcode wird ersichtlich, dass *Links* als Bit den Wert 0 und *Rechts* den Wert 1 hat. Weiterhin geht hervor, dass das 1. Bit (von rechts) für den Receiver steht und das 2. Bit für den Transmitter genutzt wird.

Im sequentiellen beidseitigen Modus (Wert 4+) kann die Ansteuerung nur vermutet werden, da die Dokumentation eine genaue Beschreibung offen lässt. Meine Theorie dazu ist in Tabelle 2.4 niedergeschrieben. Sie überschneidet sich mit der aktuellen Theorie von B-Human: „In the modes 4 and 7, the receivers measure the pulse sent by the transmitters on the same side. In the modes 5 and 6, they measure the pulse sent by the transmitters on the opposite side.“ ([B-Human13, S. 75])

2 Grundlagen

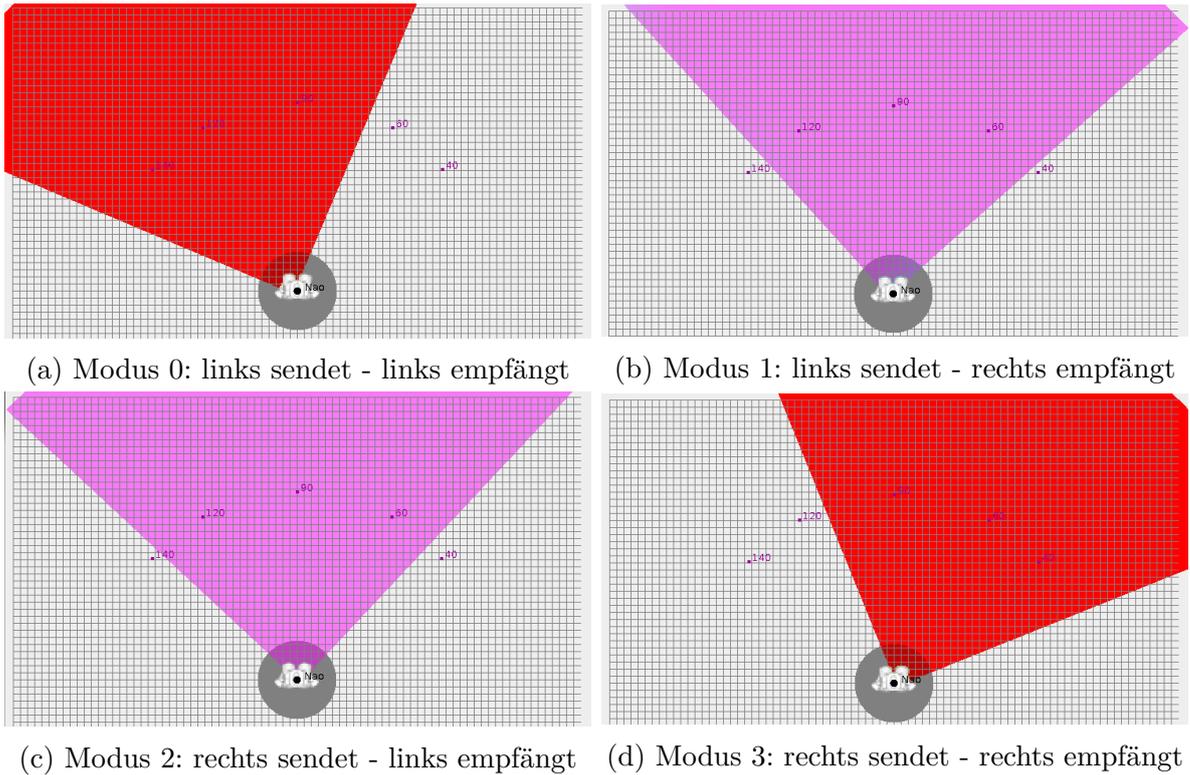


Abbildung 2.4: Idealer Öffnungswinkel bei 90 Grad in Modus 0 bis 3 basierend auf [B-HumanCode, usObstacleModelProvider.cfg, Stand 2013]

Wert	TM 1	REC 1	TM 2	REC 2	Binär 1	Binär 2
4	Links	Links	Rechts	Rechts	100	011
5	Links	Rechts	Rechts	Links	101	010
6	Rechts	Links	Links	Rechts	110	001
7	Rechts	Rechts	Links	Links	111	000

Tabelle 2.4: Werte für die Ansteuerung sequentieller beidseitiger Modi

Wie in Tabelle 2.4 dargestellt, ist in Binär 1 ersichtlich, dass Bit 2 dafür zuständig ist, beidseitig zu senden bzw. eine zweite zu erfolgende Messung anzuzeigen. Dagegen sehen Bit 0 und Bit 1 analog mit Binär aus Tabelle 2.3 aus. Auch geht daraus hervor, dass Binär 1 und Binär 2 inverse voneinander sind.

In der Dokumentation werden noch weitere Bits beschrieben. Auf diese wird in der Arbeit nicht näher eingegangen. Informationen zu den Modi sind unter [AldV1.14, pref_file_architecture.html] oder [AldV2.1, actuator_sensor_names.html] zu finden.

Zu den Vorteilen von beidseitigen Modi zählen, dass nur ein Ansteuerungsbefehl genutzt wird und eine erhöhte Menge an Rückgabewerten (Abschnitt 2.3.1) existiert. So reichen zwei Befehle, um alle Möglichkeiten der Sensoren abzudecken.

Um einen Modus zu setzen, wird die Abfolge aus Listing 1 verwendet.

```

1 | ALValue commands;
2 | commands.arraySetSize(3);
3 | commands[0] = string("Device/SubDeviceList/US/Actuator/Value");
4 | commands[1] = string("Merge"); // doesn't work with "ClearAll"
5 | commands[2].arraySetSize(1);
6 | commands[2][0].arraySetSize(2);
7 | commands[2][0][0] = ALValue(4.0f); // set Modenumber here
8 | commands[2][0][1] = al_dcm->getTime(0); // DCMProxy::getTime()
9 | al_dcm->set(commands); // DCMProxy::set()

```

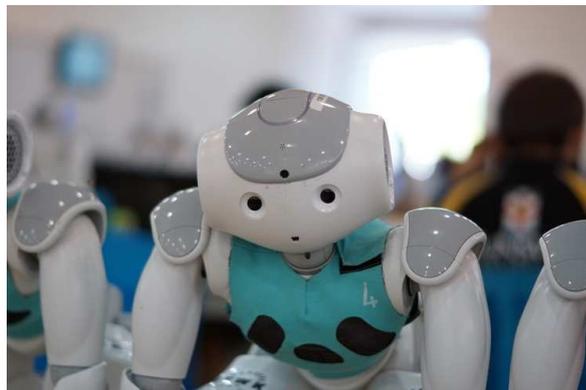
Listing 1: Einmaliges Absetzen des Modi 4 mit NAOqi

2.4 Trikot

Wie im normalen Fußball auch trägt der NAO ein Trikot. Dies erfüllt den Zweck, dass die Schiedsrichter, Teams und Zuschauer die Roboter auseinander halten können. Auf der linken Schulter und groß auf dem Rücken ist die Spielernummer zu sehen. Die verwendeten Farben nach den Regeln „The official jersey shirt colors are cyan and magenta (referred to as blue and red throughout this document).“ ([Rules2014, S. 8]) sind Magenta und Cyan, oft vereinfacht mit Rot und Blau.



(a) Blaues Trikot mit der Nummer 4



(b) Der NAO mit Trikot (Quelle: Nao-Team HTWK)

Abbildung 2.5: Trikots der NAOs

In das Trikot wurden mit Netz besetzte Schlitze in Brusthöhe eingearbeitet, um das Passieren des Ultraschalls zu ermöglichen. Es ergibt sich die Frage, inwieweit das Netz Einfluss auf die Messungen hat bzw. der Trikotstoff beim Verrutschen des Trikots (siehe rechter Roboter in Abbildung 1.1b).

Dieser Tatsache wird im Abschnitt 5.1 nachgegangen.

2.5 Aufbau und Integration in die bestehende Software

In der nachfolgenden Abbildung 2.6 wird der schematische Aufbau der verwendeten Programme dargestellt.

2.5.1 NAOqi (Aldebaran API)

NAOqi ist die mitgelieferte API des Herstellers Aldebaran und stellt als Betriebssystem des NAOs grundlegende Funktionen zur Verfügung.

Die benötigten Klassen sind *ALSonarProxy*, um sich als Nutzer bei den Sensoren anzumelden/abzumelden (subscribe/unsubscribe), *DMCProxy* für das Setzen des Modi (siehe Abschnitt 2.3.2) und *ALMemoryProxy* für das Auslesen der Rückgabedaten (siehe Abschnitt 2.3.1)

2.5.2 Bridge

Die Bridge ist ein Modul, welches den Datenzugriff vom Rest der Software abtrennt und auf NAOqi aufsetzt. Es wurde vom Nao-Team HTWK konzipiert, um im Falle eines Absturzes der Firmware einen Crash des gesamten Roboters abzuwenden.

```
1 | #ifndef ULTRASOUND_ARRAY_SIZE
2 | #define ULTRASOUND_ARRAY_SIZE 10
3 | #endif
4 | struct ultrasound_t {
5 |     float normal;
6 |     float left[ULTRASOUND_ARRAY_SIZE];
7 |     float right[ULTRASOUND_ARRAY_SIZE];
8 | };
```

Listing 2: *struct ultrasound_t* für die Rückgabe der Bridge an die Firmware

Für die Kommunikation mit NAOqi wird auf deren Funktionalität zurückgegriffen. Für die Kommunikation mit der Firmware für Ultraschall liegt die Struktur aus Listing 2 zugrunde.

2.5.3 Firmware

Die Firmware ist das Herzstück der Software des Nao-Teams. In dieser sind alle Funktionalitäten enthalten und sie ist ausschließlich für den NAO konzipiert. Die Quellcodes für die Tests aus Kapitel 5 wurde in diese Software integriert. Für die benötigte Kommunikation zwischen Firmware und Bridge oder NaoControl stehen bereits Funktionen zur Verfügung, welche das Weiterverarbeiten von erzeugten Daten und Ergebnissen begünstigt.

2.5.4 USTest

USTest ist ein im Rahmen dieser Arbeit entwickeltes Programm zur Auswertung der Messdaten und zur Simulation von Triangulierungen mit idealen Sensoren. Ausschnitte sind in Abbildung 3.2 zu sehen. Das Programm liegt auf der CD bei.

2.5.5 NaoControl

NaoControl ist das externe Debugging- und Visualisierungs-Tool des Nao-Teams HTWK. Dank eingebauter Kommunikation mit der Firmware wurde es für Probeversuche mit dem US-Sensor genutzt, um die Messwerte zu visualisieren. Im Nao-Team hat es eine wichtige Stellung als Werkzeug. Es ist zu empfehlen, den USTest in Zukunft in dieses Tool fest zu integrieren.

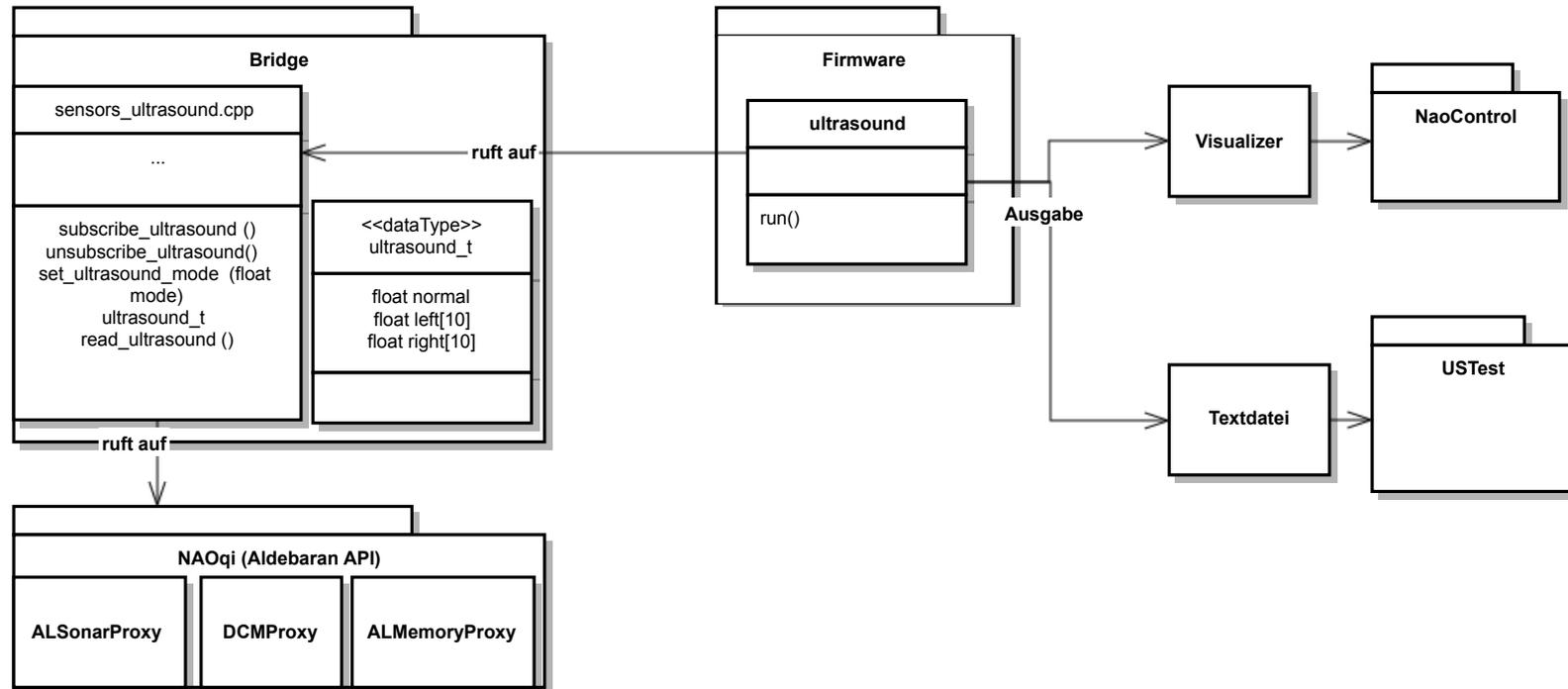


Abbildung 2.6: Allgemeiner Aufbau der Software

3 Mögliche Verfahren der Hinderniserkennung

Dieses Kapitel beschäftigt sich mit ausgewählten theoretischen Ansätzen der Hinderniserkennung und stellt die jeweiligen Schwächen und Stärken dar. Als Teilgebiete werden *Triangulierung durch Schnittpunktberechnung* (Abschnitt 3.1) und *Rasterungsansatz von B-Human* (Abschnitt 3.2) erläutert.

3.1 Triangulierung durch Schnittpunktberechnung

Anmerkung: Es wird in den beiden Methoden davon ausgegangen, dass die Reflexionen und daraus ermittelten Messdaten auf derselben Höhe wie die Sensoren gemessen werden.

3.1.1 Triangulierung über den Kosinussatz

Als einfachste Methode zur Triangulierung eignet sich die Dreiecksberechnung. Über drei Seiten eines Dreiecks ist es möglich, die Winkel zwischen diesen zu errechnen. Sind noch zwei Eckpunkte dieses Dreiecks gegeben, so ist eine Berechnung des dritten Eckpunktes durchführbar.

Wenn eine Messung je einen Wert sowohl für den linken (Sl) als auch für den rechten Sensor (Sr) zurückgibt, liegen zwei Seitenlängen vor. Nach Abbildung 3.1 ist a die Seite mit der gemessene Länge vom Sl und b von Sr. Die dritte Seitenlänge c ist der Abstand zwischen den beiden Sensoren. Der genaue Abstand errechnet sich aus der Summe der Beträge der relativen Positionen aus Y von links und Y von rechts (siehe Tabelle 2.1). Wird von einem Sensor kein Wert zurück gegeben, so ist diese Methode nicht anwendbar.

Die zwei benötigten Eckpunkte sind die relativen Koordinaten der Sensoren (Sl, Sr) zum NAO.

3 Mögliche Verfahren der Hinderniserkennung

$$\begin{aligned}a^2 &= b^2 + c^2 - 2bc \cdot \cos \alpha \\b^2 &= a^2 + c^2 - 2ac \cdot \cos \beta \\c^2 &= a^2 + b^2 - 2ab \cdot \cos \gamma\end{aligned}\tag{3.1}$$

Über den Kosinussatz (Gleichung 3.1) lassen sich die Winkel (Gleichung 3.2 für α) zwischen den Seiten bestimmen und daraus Geraden mit dem eingeschlossenen Punkt Sl oder Sr bilden. Über die Schnittpunktberechnung dieser Geraden wird die Position des Hindernis H (siehe Abbildung 3.1) ermittelt.

$$\alpha = \arccos \left(-\frac{a^2 - b^2 - c^2}{2bc} \right)\tag{3.2}$$

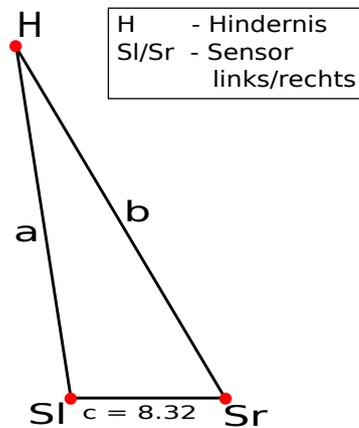


Abbildung 3.1: Schema für die Dreiecksberechnung mit dem Kosinussatz (siehe Gleichung 3.1)

Bei der praktischen Umsetzung in dem Programm USTest (Abschnitt 2.5.4) haben sich die erhaltenen Punkte als fehlerhaft erwiesen und daraufhin wurde diese Methode verworfen.

3.1.2 Triangulierung über Kreisschnittpunkte

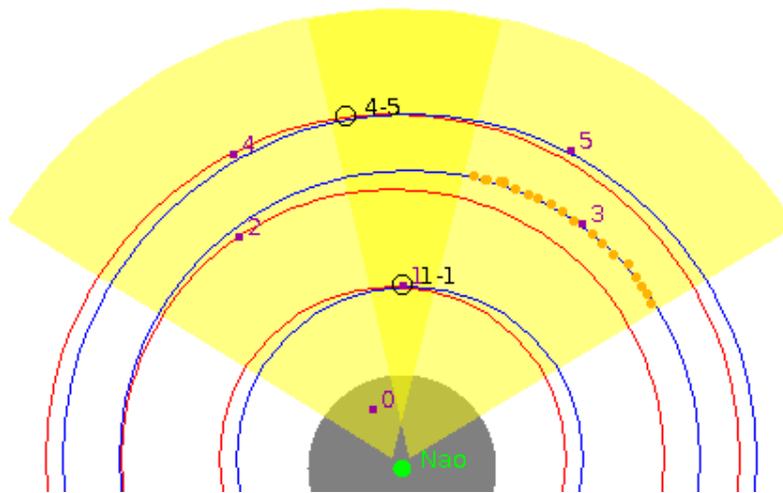
Als geeignetere Lösung gegenüber 3.1.1 bietet sich die Schnittpunktberechnung von Kreisen an. Die jeweiligen Kreise bestehen aus der gemessenen Distanz als Radius und dem Sensor als Mittelpunkt. Unter den unten genannten und in Abbildung 3.2 visualisierten Bedingungen kann ein Schnittpunkt zweier Kreise ein Hindernis sein.

Die Unterscheidung der Schnittpunkte benötigt ein genaues Modell der Messbereiche der Sensoren. Ein Schnittpunkt kann verworfen werden, wenn er außerhalb dieses Modells liegt. So kommt es meistens vor, dass der zweite Schnittpunkt einer Berechnung

3.1 Triangulierung durch Schnittpunktberechnung

zweier Kreise nicht im Modell liegt und nicht als Hindernis gewertet wird. Ein eindeutiges Hindernis kann nur in dem überlappenden Messbereich beider Sensoren erkannt werden. Wie in Abbildung 3.2a verdeutlicht, sind dabei Unterscheidungen zu treffen. Die Nichtexistenz eines Schnittpunktes aus zwei Messdaten sagt aus, dass es sich um zwei unabhängige Hindernisse handelt.

Treten mehr als zwei Hindernisse auf, besteht die Möglichkeit, dass der Algorithmus keine sinnvollen Ergebnisse mehr liefert. Es ergeben sich im Worst Case mit $10^2 = 100$ theoretische Hindernisse aus der Kombination der linken und rechten Messdaten mit je 10 Werten (siehe Abschnitt 2.3.1). Um dieses Modell auch für mehr als zwei Hindernisse nutzen zu können, muss ein geeigneter Filter entwickelt werden.



(a) Möglichkeiten der Schnittpunktbestimmungen (idealer Öffnungswinkel der Sensoren: 72°)

- Legende:
- 1 : unbewegliches Hindernis
 - 2 : bewegliches Hindernis
 - ▲ Öffnungswinkel eines Sensors
 - 1-2 : fester Schnittpunkt von 1 und 2
 - 1-2 : falscher Schnittpunkt von 1 und 2
 - : Radius linker Sensor
 - : Radius rechter Sensor
 - Nao : Mittelpunkt vom NAO
 - : Bereich $\leq 25\text{ cm}$

(b) Legende zu USTest

Abbildung 3.2: USTest mit Beispielen zur Triangulierung über Kreisschnittpunkte mit NAO im Modus 4

Zur Verdeutlichung wurden in Abbildung 3.2a einige Beispiele angeführt, welche bei einer Messung mit maximal zwei Hindernissen auftreten können. Hierbei werden die

3 Mögliche Verfahren der Hinderniserkennung

Paarungen als eigenständige Messungen und nicht als nachfolgende Echos betrachtet.

Hindernis 0 ist im Bereich unter 25cm und wird als solches Hindernis ausgegeben. Genauere Daten sind nicht bestimmbar.

Hindernis 1 ist im überschneidenden Bereich beider Sensoren und kann punktgenau als Hindernis bestimmt werden. Dieses ist mit einem festen Schnittpunkt (siehe Abbildung 3.2b) beschreibbar, der in Abbildung 3.2a mit „1-1“ gekennzeichnet ist.

Hindernis 2 und 3 sind jeweils in einem Sensorbereich gemessen worden und unterscheiden sich von der Distanz stark genug, sodass kein Schnittpunkt in dem gemeinsamen Bereichen zustande kommt. In diesem Fall wird zweimal kein Punkt, sondern ein Abschnitt im jeweiligen Sensorbereich erkannt. Für Hindernis 3 ist dies mit Punkten eingezeichnet worden (siehe Abbildung 3.2a).

Hindernis 4 und 5 sind ebenso wie Hindernis 2 und 3 jeweils in einem Sensorbereich gemessen worden und zeigen eine Schwierigkeit dieses Ansatzes auf: die beiden Hindernisse sind etwa auf der selben Distanz zum Sensor und ergeben den Kreisschnittpunkt „4-5“, welcher nicht real existiert. Somit wird geschlussfolgert, dass es sich um ein anstatt um zwei Hindernisse handelt.

3.1.3 Erkenntnisse

Der Vorteil der Methode *Triangulierung über Kreisschnittpunkte* ist, dass es in dem gemeinsamen Bereichen (Kernbereich) sehr genaue und schnelle Ergebnisse liefert. Doch der Kernbereich ist in der Theorie mit einem idealen Öffnungswinkel gegenüber dem gesamten Messbereich der Sensoren relativ klein und ist in der Praxis nicht spezifiziert.

Als weitere Nachteile erweisen sich, dass er außerhalb der gemeinsamen Bereiche keine genauen Ergebnisse in Form von Punkten liefern kann und wie bereits beschrieben, mit steigender Anzahl an Hindernissen die echten Hindernisse nicht eindeutig erkannt werden können.

Ergänzt wird die Methode mit den Ergebnissen des Tests aus Abschnitt 5.2.

3.2 Rasterungsansatz von B-Human

Diese Methode wird seit 2010 im RoboCup von B-Human, einem Team der Universität Bremen, eingesetzt und kontinuierlich weiterentwickelt (siehe [B-Human10], [B-Human12] und [B-Human13]). Der Quellcode ist veröffentlicht und auf [B-HumanCode] erhältlich.

Das zugrunde liegende Modell ist ein dreidimensionales Raster (visualisiert in Abbildung 3.3) mit Namen *USObstacleGrid*. In den Abmaßen X und Y beträgt es je 80 Zellen

und in der Z-Achse 16 Ebenen, welche die Intervalle der Messungen darstellen. In der Praxis wird es als Matrix mit Ringpuffer für jeder Zelle umgesetzt.

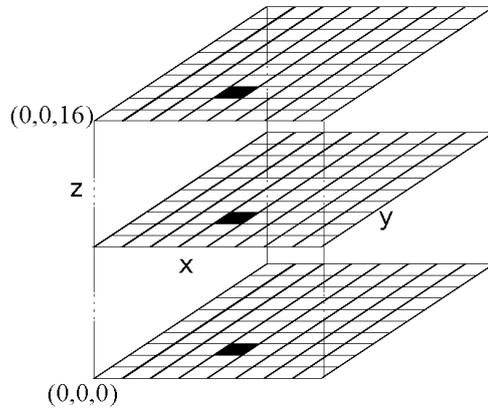


Abbildung 3.3: Aufbau eines dreidimensionalen Rasters, die schwarzen Felder beschreiben Messungen einer Zelle über die Zeit (Quelle: [Becker2002, vom Autor geändert])

Im Laufe der Zeit hat sich die Methode verändert. So ist die gröbere Auflösung („The grid currently used has a resolution of 45×45 cells with each cell having a size of $60\text{mm} \times 60\text{mm}$.“ ([B-Human10, S. 56])) einem feineren, fast doppelt so großes Raster aus dem Jahr 2013 („[...] approximately $2.7\text{m} \times 2.7\text{m}$ grid of 80×80 cells.“ ([B-Human13, S. 74])) gewichen. Die weiteren Ausführungen beziehen sich auf den Stand von 2013.

Ein detektiertes Hindernis wird mit einer Dicke von 100mm in dem idealen Sendebereich des Sensors von 90° eingetragen (siehe [B-Human13, S. 74]). Damit wird unter anderem versucht, die Fläche des NAOs abzudecken. Es ist auch ein Sicherheitsbereich, um zusätzliche Abweichungen in den Messungen vorzubeugen.

Für die Bewertung müssen 10 der 16 letzten Messungen ein Hindernis wahrgenommen haben, dann gilt die Zelle für den NAO als Hindernis. Der erste Rückgabewert (Abschnitt 2.3.1) jedes Sensors wird stärker gewichtet als der zweite und dritte. Die restlichen Messergebnisse werden verworfen.

Mit dem *ObstacleCombinator* werden die Zellen, die ein Hindernis für den NAO darstellen, zu einer Gruppe von Zellen zusammengefasst. Dies kann dazu führen, dass mehrere NAOs zu einem Hindernis verschmelzen.

3.2.1 Erkenntnisse

Der primäre Vorteil des Rasterungsansatzes ist, dass er erprobt ist und aktiv eingesetzt wird. Auch hat er schon Themen wie Speicherung der Daten (über 16 Zeitschritte)

3 Mögliche Verfahren der Hinderniserkennung

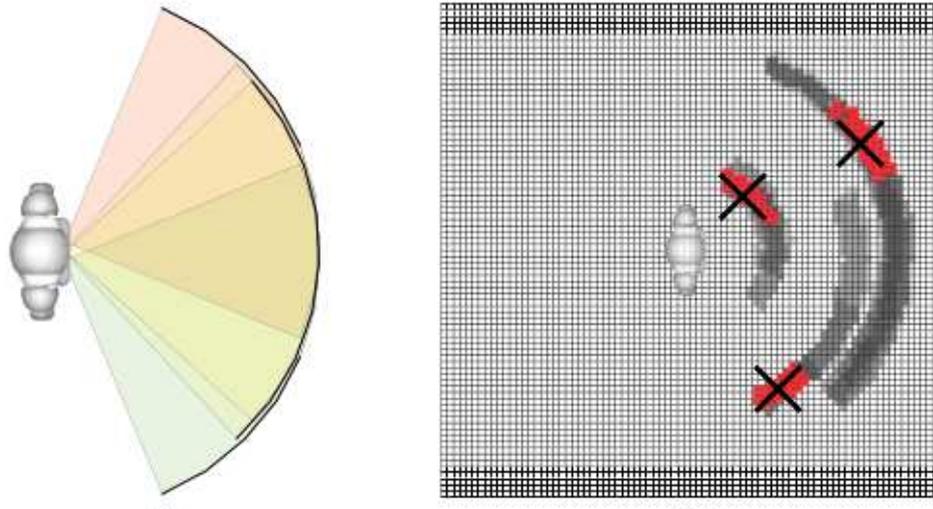


Abbildung 3.4: Aufbau der Ultraschallbereiche (links) von B-Human und ein Beispiel aus einem Spiel (rechts) (Quelle: [B-Human13, S. 76])

und langsame Fortbewegung des Messenden, worauf in dieser Arbeit nicht eingegangen wurde, integriert (siehe [B-Human13, S. 76]).

Ein Zeitbedarf von 1,12 Sekunden ($16 \text{ Messungen} \times 70 \text{ms}$) ist für das Spiel im akzeptablen Bereich, um ein Hindernis zu identifizieren. In der Software sind die Möglichkeiten nicht ausgereizt. Nach einem Sturz des NAOs im Spiel werden die Daten der Sensoren nutzlos und müssten erst wieder neu gesammelt und berechnet werden.

Im Gegensatz zur *Triangulierung über Kreisschnittpunkte* (Abschnitt 3.1.2) benötigt diese Methode für jeden Zeitschritt mindestens 6400 Operationen ($80 \text{ Zellen} \times 80 \text{ Zellen}$).

Ergänzt wird die Methode mit den Ergebnissen des Tests aus Abschnitt 5.2.

4 Allgemeine Informationen für Testszzenarien

In diesem Kapitel werden die allgemeinen Informationen zum Aufbau der Tests beschrieben und auf selbst aufgestellte Konventionen, wie der Speicherung von Messdaten in Dateien, eingegangen.

4.1 Positionen der Messungen auf dem Spielfeld

Ein Spielfeld nach den Regeln der Standard Platform League beziehend auf das Jahr 2012 (siehe [Rules2012, S. 1]) bietet eine optimale Umgebung für Messungen mit dem Ultraschall. Zum einen ist ausreichend Platz vorhanden, um den Sensor nach den Spezifikationen des Herstellers voll auszureizen. Zum anderen sind die Umgebung und Spielelemente in Form von Untergrund (Teppich und Spielfeldmarkierungen) und Toren vorhanden.

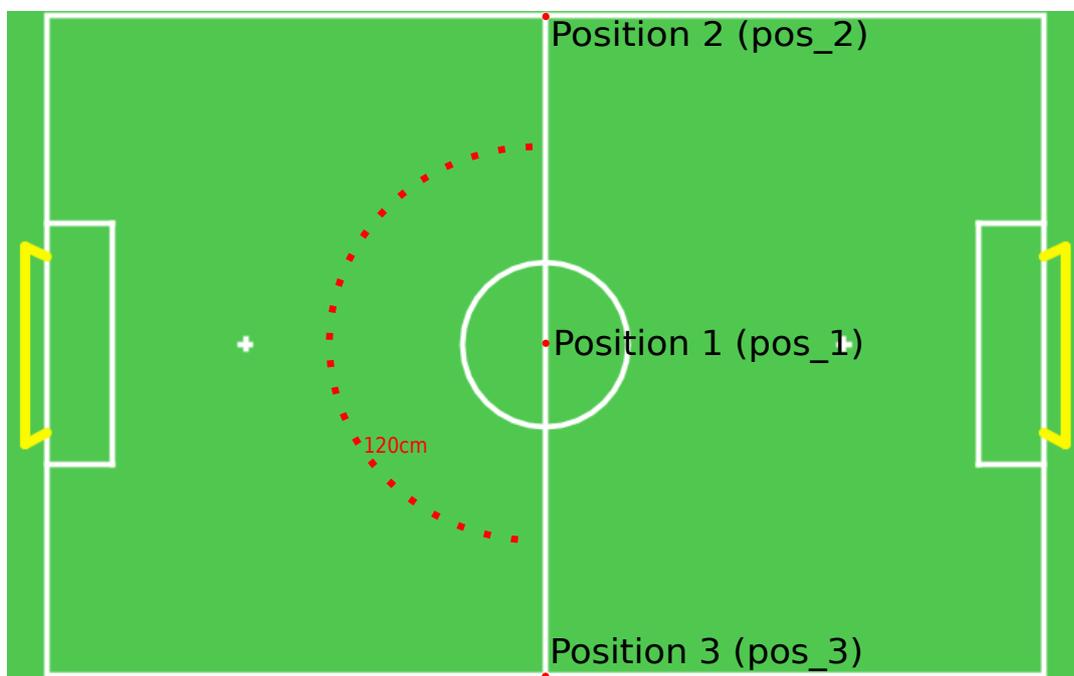


Abbildung 4.1: Ausgangspositionen für Messungen auf einen 6x4 Spielfeld

In der Größe ist das Feld 6m lang mal 4m breit. Es ergeben sich folgende Position für messende Roboter, wie in Abbildung 4.1 eingetragen.

Bei Position 1 (pos_1) steht der NAO auf dem Anstoßpunkt mit Brust in Richtung des eigens gestalteten 1,2m-Halbkreises (nicht maßstabsgetreue rote gestichelte Linie). Die Markierung des Halbkreises entstand, um das Problem aus Abschnitt 1.5 zu lösen. Deshalb wird die maximale Distanz auf 1,2m begrenzt. Die Skalierung innerhalb der Halbkreismarkierungen beträgt jeweils 10° zur besseren Bestimmung des Winkels zum Messenden.

Nachteilig ist, dass der Abstand vom Messenden in seiner Position zum Spielfeldrand nur 2m statt 2,55m besitzt und der Sensor nicht maximal ausgereizt werden kann. Zwar beschreiben die Regeln (siehe [Rules2012, S. 1]), dass die Randflächen außerhalb der Außenmarkierungen zusätzlich noch 0,7m betragen sollen. Doch dies war im Labor des Nao-Teams nicht umsetzbar.

Daher werden Positionen 2 (pos_2) und 3 (pos_3) aus Abbildung 4.1 für Messungen für den maximalen Bereich der Sensoren genutzt. Die Brust des NAOs ist zum Anstoßpunkt ausgerichtet, welcher der Position 1 entspricht.

4.2 Allgemeiner Aufbau einer Testsoftware

Im Rahmen der Arbeit werden alle Tests aus Kapitel 5 in die Firmware (Abschnitt 2.5.3) eingebettet und ausgeführt.

Wenn nichts anderes in den Tests definiert ist, werden in allen zu untersuchenden Modi (0 bis 7, siehe Abschnitt 2.3.2) Messwerte erhoben. Dabei sind immer 10 Messungen je Modus durchzuführen. Die Messwerte werden, egal welcher Modus gesetzt ist, immer mit allen Rückgabewerten (Abschnitt 2.3.1) abgerufen. So ergeben sich insgesamt 1600 Messwerte in 80 Messungen für jeden durchgeführten Test.

```
1 | subscribe()
2 | foreach i in Modes do
3 |     for j = 0 to 10 do
4 |         set_us_mode(i)
5 |             usleep(70000)
6 |             results += read_us_sensor_data()
7 |             usleep(70000)
8 | Write(results)
9 | unsubscribe()
```

Listing 3: Allgemeiner Aufbau der Testsoftware mit 10 Messungen je Modus

Listing 3 beschreibt den allgemeinen Ablauf in einem Programm, nach den oben genannten Bedingungen. Der US-Sensor ist generell deaktiviert, um im Spielverlauf Strom zu sparen und eine längere Akkulaufzeit zu erreichen. Mit *subscribe* meldet sich das

Programm als Nutzer beim Sensor an und aktiviert ihn gegebenenfalls. Die Variable *Modes* enthält alle zu messende Modi. Der Modus *i* wird mit dem Befehl *set_us_mode(i)* gesetzt.

Danach folgt ein *usleep(70000)*, eine Pause von 70ms, um die Hardware arbeiten zu lassen. Der Wert 70ms basiert auf der Beobachtung von B-Human: „We also found out that only 70ms after firing the sensors, the readings seem to be correct.“ ([B-Human13, S. 75]). Mathematisch würden für das Senden und Empfangen über 2,55m 14,87ms ausreichen, wie in Gleichung 4.1 zu sehen ist. Hinzukommen wahrscheinlich Hardwarezugriffe und -kommunikation, genaue Angaben sind in der Dokumentation ([AldV2.1]) nicht enthalten.

Weiterhin sei zu erwähnen, dass die Sensoren untereinander eine zeitliche Pause von 50ms bei der Ansteuerung eines beidseitigen Modi haben: „This suggests that the two ultrasound sensors are fired one after the other with a delay of about 50ms.“ ([B-Human12, S. 30]). Die Durchführung ergänzender Untersuchungen der zeitlichen Abläufe würde zu präziseren Ansteuerungen verhelfen.

$$\frac{2 \times 2,55\text{m}}{343\frac{\text{m}}{\text{s}}} = 0,01487\text{s} = 14,87\text{ms} \tag{4.1}$$

Der Befehl *read_sensor_data()* liest anschließend die Daten vom Sensor aus und speichert sie in der Variable *results* zwischen.

Die letzte Pause von 70ms wurde gewählt, um mögliche Reflexionen in den Daten von früheren Messungen zu vermeiden, wie B-Human in Ihrer Veröffentlichung mit „Reflections from obstacles far away can be registered as very close obstacles because the reflected signal arrives in later measurement cycles.“ ([B-Human12, S. 30]) beschreiben.

Abschließend wird *results* ausgegeben und der Sensor meldet sich ab (*unsubscribe*).

4.3 Messergebnisse und Dateinamen

Dieser Abschnitt enthält die Konventionen für Dateinamen und die Ausgabe der Messungen in Dateien für die computergestützte Weiterverarbeitung via USTest (siehe Abschnitt 2.5.4).

4.3.1 Dateinamenkonvention

Schema: name.position.jersey.to.number1.number2[.extra].txt

name Name des zu messenden Roboters, dabei werden Namen der NAOs der HTWK verwendet

position Position des messenden Roboters (siehe Abschnitt 4.1)

jersey Informationen, ob und wie der Roboter ein Trikot trägt (siehe Abschnitt 5.1.1)

to ist eine Umschreibung des Hindernisses

number1 wird für die Auswertung in USTest benutzt, enthält meistens den Abstand vom Messenden zum Hindernis

number2 wird für die Auswertung in USTest benutzt, enthält meistens den Winkel vom Messenden zum Hindernis

extra (optional) enthält zusätzliche Informationen zum Test

Beispiel: *hybrid.pos2.no_jersey.empty_field.0.0.txt*

Hier misst ein Roboter mit Namen *Hybrid* auf Position 2 ohne Trikot auf einem leeres Feld.

4.3.2 Formatierung des Messergebnisses

Mit dem Beispiel in Listing 4 werden die nach dem Test entstehenden Daten und deren Formatierung in einer Datei erklärt.

```
1 | %metadata=wert (optional)
2 | %column=0
3 | # optionales Kommentar
4 | mode: 0 # Modus 0
5 | →0 # Messung 0
6 | →n→0.25 # normal (US/Sensor/Value)
7 | → # 10 Messungen links (US/Left/Sensor/Value)
8 | →l→2.55 | 2.55 | 2.55 | 2.55 | 2.55 | 2.55 | 2.55 | 2.55 | 2.55
   | ↪ | 2.55 |
9 | → # 10 Messungen rechts (US/Right/Sensor/Value)
10| →r→0.25 | 0.50 | 0.74 | 2.21 | 2.55 | 2.55 | 2.55 | 2.55 | 2.55
   | ↪ | 2.55 |
11| →1 # Messung 1
12| →n→0.51
```

Listing 4: Beispieldaten zur Formatierung eines Messergebnisses

Das Schlüsselwort *mode* leitet einen neuen Modus (Abschnitt 2.3.2) in der Ausgabe ein, in Listing 4 ist es Modus 0. Weiterführend ist die Zeile darunter eine fortlaufende Zahl zur Zählung der Messungen in dem aktuellen Modus. Wieder darunter sind drei Zeilen mit Messergebnissen. Die Zahlen aus „0.25 | 0.50 | ...“ sind die Werte aus „Value | Value1 | ...“ der jeweiligen Seite (siehe Abschnitt 2.3.1 und Listing 2).

Optional können Kommentare mit *# Kommentar* oder auch Metainformationen mit *%metadata=wert* angegeben werden. Stellvertretend sei hier mit der Metainformationen *%column=0* gemeint, dass *column* den Wert 0 hat.

4.3 Messergebnisse und Dateinamen

Alle erhobenen Messdaten sind unter [Test:GitHub] zu finden und wurden mit USTest verarbeitet.

5 Aufbau der Testumgebungen und Durchführung der Tests

Die Tests der entworfenen Testumgebungen werden in diesem Kapitel empirisch untersucht und ausgewertet.

5.1 Testumgebung: leeren Feldes

Diese Testumgebung sieht vor, mit dem Sensor auf einem leeren Feld Messungen durchzuführen. Beachtet wird dabei der Einfluss der Umgebung. Für die Testumgebung werden verschiedene Phasen untersucht, um die Auswirkung des Trikots auf die Messwerte zu ermitteln (siehe Abschnitt 2.4).

Um die Problematik zu verdeutlichen, wäre folgende Szene in einem Spiel denkbar: Der Tormann steht allein auf seiner Spielfeldhälfte im Tor. Die gesamte Fläche vor ihm ist leer. Alle anderen Feldspieler befinden sich vor dem gegnerischen Tor. Nun erkennt der Tormann weit rechts von ihm ein Hindernis. Daraufhin versucht er das Tor vor dem vermeintlichen Gegenspieler abzudecken. Aber dabei öffnet er die linke Torseite für Fernschüsse. In dieser Spielszene wird die Zusammenarbeit mit anderen Sensoren ignoriert, um das Problem als solches zu betrachten.

Visuell ist das vom Tormann wahrgenommene Hindernis nicht erkennbar, es gilt mögliche Ursachen zu bestimmen. Unebenheiten des Fußbodens oder ein (verrutschtes) Trikot könnten der Grund sein. Laut Aldebaran ist es durchaus möglich, über die volle Distanz den Fußboden als Hindernis zu messen: „The maximum distance for detection is approximately 1.20m (for the old version), but you’ll likely detect the ground before.“ ([AldV2.1, actuator_sensor_names.html]). Das Ausmaß dieses Problems sollte für die Berücksichtigung in der Hinderniserkennung bekannt sein, um darauf reagieren zu können.

5.1.1 Typische Testfälle

Ziel ist herauszufinden, wie fehlerbehaftet die Rückgabe des Sensors ist und Störungen oder unerwünschte Echos zu identifizieren. Es wird getestet, welche Messwerte bei einem leeren Feld auftreten können und ob ein Trikot Einfluss nehmen kann.



(a) Vorbereitung der Messungen für Position 2 (b) Der NAO trägt Trikot 6 verkehrt herum

Abbildung 5.1: Vorbereitungen des Tests *leeres_feld*

Im Aufbau wird der sendende NAO vor eine freie Fläche gestellt. Hierfür eignen sich Position 2 und 3 (siehe Abbildung 5.1a). In drei Testphasen unterteilt, wird der NAO entweder mit keinem (Phase: *ohne Trikot*), einem richtig herum angezogenen (Phase: *mit Trikot*, siehe Abbildung 2.5b) oder einem verkehrt herum angezogenen (Phase: *verkehrtes Trikot*) Trikot bekleidet. Der NAO mit dem verkehrt herum getragenen Trikot hat die Sensorschlitze auf dem Rücken und den durchgängigen Trikotstoff vor dem Sensor (siehe Abbildung 5.1b).

Das einzige vorstellbare Hindernis auf dem Feld wäre ein getragenes Trikot. Unter der Annahme, das Trikot würde keine Impulse durchlassen, müsste ein zurückgegebener Messwert 0,25m sein. Würde das Trikot störend auf die Messwerte wirken, werden sich schlechtere Werte als ohne Trikot aufzeigen. Hat das Trikot keinen Einfluss, der Unterschied wäre nicht statistisch signifikant.

Die Erwartung ist, dass der Sensor keine Hindernisse trotz Trikot auf einem leeren Feld erkennt und somit kein Messwert unter dem Wert 2,55m vorliegt.

5.1.2 Verwendete Software

Die im Abschnitt *Allgemeiner Aufbau einer Testsoftware* (Abschnitt 4.2) beschriebene Software wurde verwendet.

Die Dateinamenkonvention (Abschnitt 4.3.1) wurde erweitert:

no_jersey Der NAO trägt kein Trikot. (Phase: *ohne Trikot*)

wrongside_jersey Das Trikot wird verkehrt herum getragen. (Phase: *verkehrtes Trikot*)

jersey Der Roboter trägt das Trikot wie in den Regeln ([Rules2014]) beschrieben. (Phase: *mit Trikot*)

empty_field Das Hindernis trägt den Namen *empty_field*, das *leeres Feld* bedeutet.

5.1.3 Durchführung und Messergebnisse

Der Test trägt den Namen *leeres_feld* und ist unter [Test:leer] einsehbar.

5.1.4 Auswertung

Die kürzeste Distanz von 1,63m in der gesamten Testumgebung ist auf Position 1 gemessen worden. In allen 10 Messungen des Modi 6 wurden Abweichungen in der Phase *ohne Trikot* erfasst. Die Schlussfolgerung daraus war, diese Position nicht für weitere Phasen zu nutzen und Position 2 und 3 aus Platzgründen zu präferieren.

Modus	Messung	n	l0	r0
6	3	2.13	2.55	2.13
6	4	2.13	2.55	2.13

Tabelle 5.1: Abweichungen im Modus 6 bei den Messungen 3 und 4 im erstem Rückgabewert von Rechts(r0), *Tabelle 3 - Datei gefiltert - hybrid.pos_2.jersey.empty_field.0.0.txt* aus Test:leer. Die Werte sind in Metern.

Auf Position 2 wurden in allen drei Phasen mit insgesamt 240 Messungen (80 Messungen je Test mal 3 Phasen) zu 99,16% keine Hindernisse ermittelt. Die Ausnahme bilden zwei Messwerte der Phase *mit Trikot* in der Tabelle 5.1. Diese könnten durch Zufallseffekte als auch durch Störungen durch das Trikot entstanden sein.

Die Position 3 lieferte die schlechtesten Messdaten der Testumgebung, da viele Hindernisse gesehen wurden. Das Auftreten dieser Werte ist ab Modus 3 zu beobachten.

Die Analyse in Tabelle 5.2 zeigt Abweichungen mit rund 0,43m links und 0,41m rechts. Ähnliche Werte kommen in den anderen Phasen vor, sogar mit weniger Streuungen (siehe [Test:leer, Tab. 9 - 14]). Eine Beobachtung während der Durchführung lässt den Eindruck zu, dass der Fußboden das Hindernis war. Beim Versetzen des NAOs von Position 2 zum Mittelkreis wurden die gemessenen Daten kleiner.

Als Lösungsansatz für das Eliminieren unerwünschter Daten bietet sich an, alle erkannten Hindernisse über der Obergrenze von 1,20m entsprechend des Problems in Abschnitt 1.5 zu ignorieren. Weiterführend wäre eine Beobachtung durch eine Langzeitmessung an mehreren Positionen oder in Bewegung sinnvoll.

Modus	Links			Rechts		
	Soll	ØIst	Differenz	Soll	ØIst	Differenz
3				2,55	2,33	-0,22
4	2,55	2,07	-0,48	2,55	1,93	-0,62
5	2,55	1,91	-0,64	2,55	2,17	-0,38
6	2,55	2,49	-0,06	2,55	1,93	-0,62
7	2,55	1,98	-0,57	2,55	2,33	-0,22

Tabelle 5.2: Computergenerierte Analyse zu *Tabelle 12 - hybrid.pos_3.no_jersey.empty_field.0.0.txt*. Bestimmen der Durchschnittsabweichung der jeweils ersten gemessenen Werte einer Seite je Modus. Verglichen mit dem Abstand in dem theoretischen Modell Abbildung 2.4 mit einem Öffnungswinkel von 72°. Die Werte sind in Metern angegeben. Weitere Information sind unter [Test:leer] ersichtlich.

Abschließend kann gesagt werden, dass das Trikot keinen Einfluss auf die Messungen hat. Es ist aber ersichtliche, dass die Messwerte *ohne Trikot* eine größer Streuung haben als mit Trikot. Das Tragen eines Trikot, im Spielverlauf kann als Fehlerquelle ausgeschlossen werden. Der Ultraschall durchdringt problemlos den Stoff. Wie in Abschnitt 5.1.1 gefordert, weichen die Daten nicht stark genug ab. Auf Position 2 hatte das Trikot keine Auswirkungen auf die Messungen. Weiterführend gewinnt das Hindernis „Fußboden“ mehr Bedeutung, da sich Messwerte kleiner 2,55m ergeben haben.

5.2 Test zur Überprüfung der Reichweite (Rastertest)

Wie in *Testumgebung: leeren Feldes* (Abschnitt 5.1) ist das zu bemessene Feld anfangs ohne Hindernisse und benötigt eine Breite von $2,6\text{m} \times 5,1\text{m}$. Die geeigneten Positionen sind Position 2 und Position 3 (siehe Abschnitt 4.1). Da aus Abschnitt 5.1 bekannt ist, dass Position 2 ungeeignet ist, wurde für den gesamten Test Position 3 gewählt.

Das zu bemessende Feld wird virtuell in $10\text{cm} \times 10\text{cm}$ Zellen (10×10 -Zelle) gerastert. Es entsteht ein Raster mit 26 Zellen in der Tiefe und 52 Zellen in der Breite, dabei wird die angegebene Sensorbreite von $2,55\text{m}$ auf $2,60\text{m}$ aufgerundet. Insgesamt entstehen 1352 Zellen, die symmetrisch in einen linken und rechten Messbereich zu je 676 Zellen aufgeteilt werden.

Der Test sieht vor, von jeder Zelle mehrere Messungen zu machen. Dabei werden verschiedene Hindernisse verwendet. Es darf immer nur ein Hindernis in der zu betrachtenden Zelle des Feldes stehen. Ein optimales Hindernis ist der NAO selbst, weil er das erwartete Hindernis im Spiel ist, das es zu erkennen gilt. Jedoch wäre jeder andere Gegenstand ebenso möglich, dieser sollte aber in etwa die gleiche Höhe wie der NAO besitzen. Negativ ist, dass der NAO zu sperrig für die 10×10 -Zelle ist. Er nimmt stehend grob eine Fläche von $20\text{cm} \times 30\text{cm}$ ein. Dies führte unter anderem zur Entscheidung, ein gröberes Raster für die Messung als definiert zu benutzen.

Um einen NAO als Hindernis in einer Zelle zu platzieren, muss folgendes beachtet werden: der linke untere Fuß muss mit seiner linken unteren Kante an der linken unteren Zellenkante liegen. Zwar führt dies zu Überhängen in andere 10×10 -Zellen, was durch seiner natürliche Größe ohnehin gegeben ist. Dies wird der Einfachheit halber ignoriert.

Der messende NAO steht an der Symmetrieachse (breite Linie in Abbildung 5.2), wobei diese in der Sagittalebene des Roboters liegt. Die Fußspitzen schließen am unteren Rand des Rasters ab (siehe Abbildung 5.3b).

Wie oben beschrieben, sollen mehrere Hindernisse verwendet werden. Dies wird durch zwei Phasen in jeder Zelle simuliert und durch die Veränderung der Ausrichtung des Hindernisses bewirkt. So stellt die Seitenansicht eine andere Hindernisfläche dar als die Frontalansicht des Roboters. Daraufhin wird der Aufbau weiter unterteilt in die Phase: *down* (Frontalansicht mit der Ausrichtung der Brust des Hindernis zum Messenden) und in die Phase: *left* (Seitenansicht mit der Ausrichtung der Brust des Hindernis nach links, vom Messenden aus gesehen). Der Winkel zwischen der Seitenansicht und der Frontalansicht beträgt 45° .

Unter diesen Voraussetzungen werden, wie in Abschnitt 4.2 beschrieben, alle 8 Modi à 10 Messungen gemessen.

Für die Bewertung einer Zelle wird je Sensorseite und Modus gezählt, ob Hindernisse gesehen wurden. Ob sich in dieser Zelle potentiell ein Hindernis befindet, wird nicht überprüft. Als gesehen gilt, wenn der erste Messwert unter $2,55\text{m}$ liegt. Alle anderen Messwerte werden verworfen.

Für jede Sichtung wird der interne Zähler um einen Treffer erhöht. So sind Maximal 10 Treffer in einer Phase je Zelle und Modus zu erreichen. Für die einzelnen Zellen werden die Treffer der einzelnen Phasen für jeden Modus zusammengerechnet. Es ergibt sich ein

5 Aufbau der Testumgebungen und Durchführung der Tests

Maximum von *Messungen je Modus \times Phasen*. In diesem Test sind es 20 Treffer für eine Zelle.

Es empfiehlt sich auf unerwünschte Werte durch den Fußboden (siehe Test in Abschnitt 5.1) zu prüfen, um falsche Messwerte auszuschließen.

Um die Übersicht zu behalten, sollten die Zellen spaltenweise (aufsteigend, siehe Abbildung 5.2) getestet werden. Hier gilt als Abbruchkriterium für die Spalte, dass, wenn kein Hindernis in der vorhergehenden Zelle erkannt wurde, die folgende nicht mehr getestet wird. Analog wird mit den Zeilen verfahren.

Da der NAO nicht nur sperrig ist, sondern auch ein erheblicher Zeitaufwand und Belastung für Mensch und Maschine bestand, wurde nur jede zweite Zelle für Messungen benutzt. Die Ergebnisse der Zelle werden ebenfalls den übersprungenen Zellen zugerechnet und als eine Zelle zusammengefasst. Aus Symmetriegründen wurde auch nur der linke Messbereich ausgemessen. Es ergibt sich ein Messbereich von 169 Zellen aufgrund der Seitenlänge von 20cm.

5.2.2 Vorbereitung des Feldes

Die Rasterung wurde mit Hilfe von einem Ausdruck realisiert, wie Abbildung 5.3a zeigt. Aus umweltfreundlichen Gründen wurden nur 4 Spalten gedruckt und diese wurden spaltenweise weiter versetzt.



(a) Aufbau und Fixierung der Spalten 0-4



(b) Positionierung des NAOs an der Rastergrenze

Abbildung 5.3: Bilder zur Vorbereitung des Feldes

5.2.3 Verwendete Software

Die Dateinamenkonvention (Abschnitt 4.3.1) wurde erweitert:

number1 trägt die Zeilennummer der Zelle.

number1 trägt die Spaltennummer der Zelle.

robot_down entspricht der Phase: *down*.

robot_left entspricht der Phase: *left*.

normal Der Inhalt der Datei hat die Form wie in Listing 4 beschrieben.

Messsoftware

Da während der Durchführung der Messungen das Hindernis verstellt wird, kann es z.B. durch die Beine des Umsetzenden zu falschen Messwerten kommen. Es wird empfohlen, mit zeitlichen Pausen und der Sprachfunktion des Roboters zu arbeiten. Sätze wie „Setze das Hindernis auf Feld x“ oder „Starte Messung“ sind nützlich. Ebenso hat sich ergeben, dass eine Pause von *7sec* für das Drehen oder Umsetzen des Roboters ausreicht.

```

1 | say("start")
2 | subscribe()
3 | foreach row do
4 |   say("set obstacle to row " + row)
5 |   sleep(1)
6 |   foreach orientation do
7 |     say("look at " + orientation);
8 |     sleep(6)
9 |     say("start measurement");
10 |    sleep(1)
11 |    foreach i = modes do
12 |      for j = 0 to 10 do
13 |        set_us_mode( i )
14 |        usleep(70000)
15 |        data = read_sensor_data( )
16 |        usleep(70000)
17 |        if (data.left[0] < 2.55) counter_left += 1
18 |        if (data.right[0] < 2.55) counter_right += 1
19 |        result += data
20 |        counter_all = counter_left + counter_right
21 |        say (count_all)
22 |        sleep(1)
23 |        Write(mode, row, col, orientation, counter_all,
24 |              counter_left, counter_right)
25 |        Write(results)
    say("finish")

```

Listing 5: Allgemeiner Aufbau für spaltenweise Messungen im Rastertest

5 Aufbau der Testumgebungen und Durchführung der Tests

Die folgende Modifikation (Listing 5) von Listing 3 ist auf das spaltenweise arbeiten im Raster ausgelegt. Die Phasen werden im Quelltext *Orientation* genannt.

```
1 | Write(mode, row, col, orientation, counter_all, counter_left,  
  | counter_right)  
2 | //Ausgabebeispiel: 4;8;4;1;14;10;4
```

Listing 6: Ausgabeoperation (aus Listing 5) und Ausgabebeispiel zusätzlicher Zahlen einer Zelle im Rastertest

Alle zusätzlichen Ausgaben sind Zahlenwerte, die durch ein Semikolon getrennt sind. Bei *Orientation* steht eine 0 für *down* und eine 1 für *left*. Das Ausgabebeispiel in Listing 6 ergibt folgenden Inhalt: Für den Modus 4 in der Zeile 8 Spalte 4 mit Ausrichtung *links* wurden 14 Hindernisse gemessen, wobei 10 vom linken und 4 vom rechten Sensor sind.

Auswertungssoftware

Um den kompletten Wirkungsbereich abzubilden, wenn nur ein Messbereich (links oder rechts) gemessen wurde, wird Listing 7 verwendet. In den gespiegelten Zellen des Rasters werden die linken Messwerte zu den rechten oder die rechten werden zu den linken. Dadurch werden auch Zellen abgebildet, die nicht getestet wurden.

```
1 | values = values.mapValues(  
2 |   f => f ++ f.map(g => (g._1._1, -g._1._2 - 2) -> g._2.swap)  
3 | )
```

Listing 7: Spiegeln der Werte vom linken zum rechten Messbereich in der Auswertungssoftware

5.2.4 Durchführung und Auswertung

In Abbildung 5.4a sieht man eine Messung der *Zelle*[10;0] mit Zeile 10, Spalte 0. Diese Daten wurden in Abbildung 5.5 bis Abbildung 5.8 für die Modi 4 bis 7 für den linken Sensor visualisiert. Für den simulierten rechten Sensor sind es gespiegelte Werte. Die Abbildungen zeigen die Anzahl der Treffer pro Zelle farblich untersetzt und mit Zahlen versehen. Die Phasen sind dabei nicht mehr beachtet worden. Die maximale Anzahl an Treffern beträgt 20, die minimale 0. Null gilt zur Kennzeichnung, dass eine Messung ohne Treffer durchgeführt wurde. Verdeutlicht ist dies durch die Legende in Abbildung 5.4b.

Die Messdaten für Spalte 0 mit der Ausrichtung *down* sind unter [Test:matrix-0-down] einsehbar.

Zu beobachten ist, dass die Form keinen idealen Kreisbogen, sondern eher einem Pilz entspricht. Die maximal getesteten Zeilen waren 22, dies entspricht $2, 20m - 2, 40m$ und 18 Spalten mit $1, 80m - 2, 0m$. Die Ausmaße der beiden Sensoren entsprechen demzufolge $2, 40m \times 4m$ Gesamttiefe mal Gesamtbreite.

5.2 Test zur Überprüfung der Reichweite (Rastertest)



0	keine Hindernisse
r	nur rechts Hindernisse
l	nur links Hindernisse
20	20/20 Hindernisse
18	18/20 Hindernisse
10	10/20 Hindernisse
2	2/20 Hindernisse
0	0/20 Hindernisse
↙	keine Messung

(a) Messungen einer Zelle mit der Ausrichtung *down*

(b) Legende: Nach Anzahl der Treffer eingefärbte Zellen

Abbildung 5.4: Umsetzung des Tests und Legende

Für Sensormodus 4 und 7 ergeben sich eindeutige Kernbereiche (Treffer ≥ 10), wo Messwerte empfangen werden können. Zwar gibt es auch Ausreißer (Treffer ≤ 5), doch sind diese im Gegensatz zu Modus 5 und 6 weniger in der Anzahl. Bei den Modi 5 und 6 gibt es ebenfalls Kernbereiche, doch im kleineren Ausmaß und mit Lücken. Dagegen sind die Bereiche mit Ausreißern größer. Diese reichen im Modus 5 weit in den linken Messbereich hinein, im Modus 6 in den rechten.

Modus 4, 6, 7 würden die Theorie aus Abschnitt 2.3.2 verstärken, auch wenn man hierzu die Formen und Reichweiten anpassen müsste. Die Ausreißer im Modus 5 lassen sich hingegen nicht erklären.

Fortführend wäre zum Abgleich der Messwerte eine Auswertung mit Triangulation (Abschnitt 3.1) sinnvoll. So ließe sich überprüfen, ob die Daten mit der Position der Zelle übereinstimmen. Dies ist aber nur bedingt mit dem Ansatz aus (Abschnitt 3.1) machbar, da der gemeinsame Bereich der beiden Sensoren nicht groß ist. Für den Modus 4 und 5 (siehe Abbildung 5.9 und Abbildung 5.10) wäre der Bereich mittig gerade mal 40cm groß.

Abschließend halte ich die Hinderniserkennung mit *Triangulierung durch Schnittpunkt-berechnung* (Abschnitt 3.1) nur ergänzend sinnvoll und den Ansatz von B-Human (Abschnitt 3.2) ohne Überarbeitung nicht ausreichend.

5.2 Test zur Überprüfung der Reichweite (Rastertest)

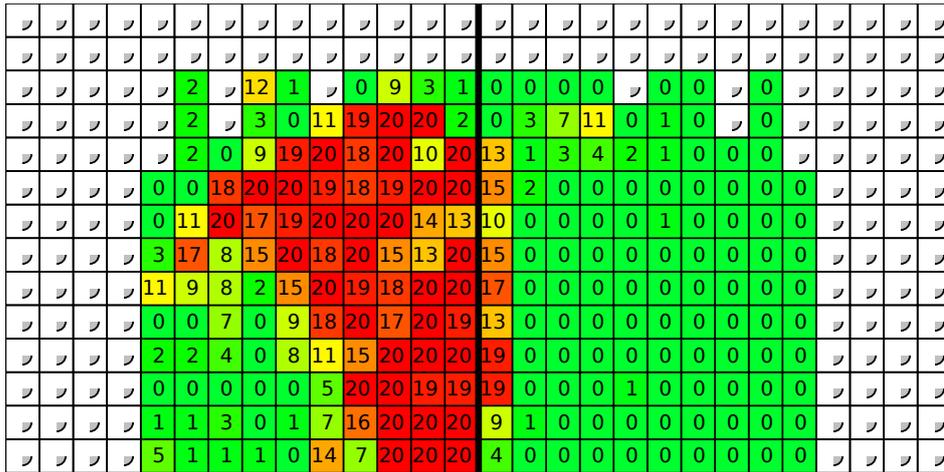


Abbildung 5.8: Visualisierung des Rasters für Modus 7 linker Sensor

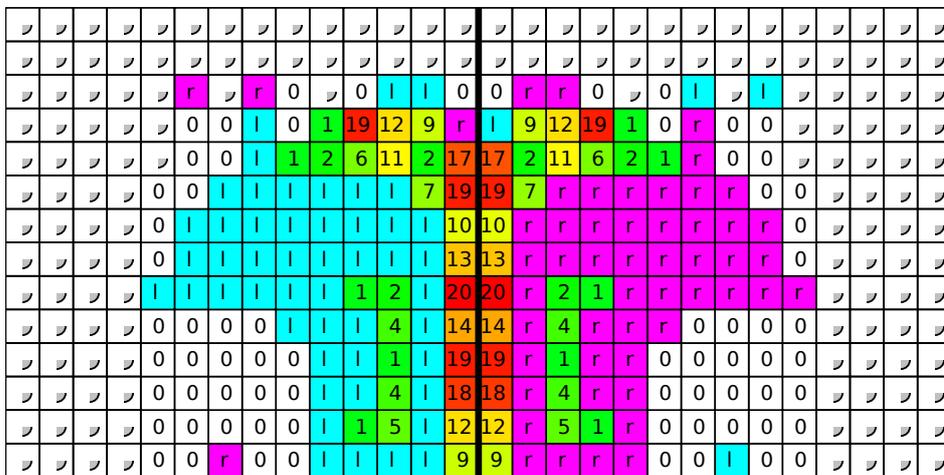


Abbildung 5.9: Visualisierung der gemeinsamen Bereiche für Modus 4

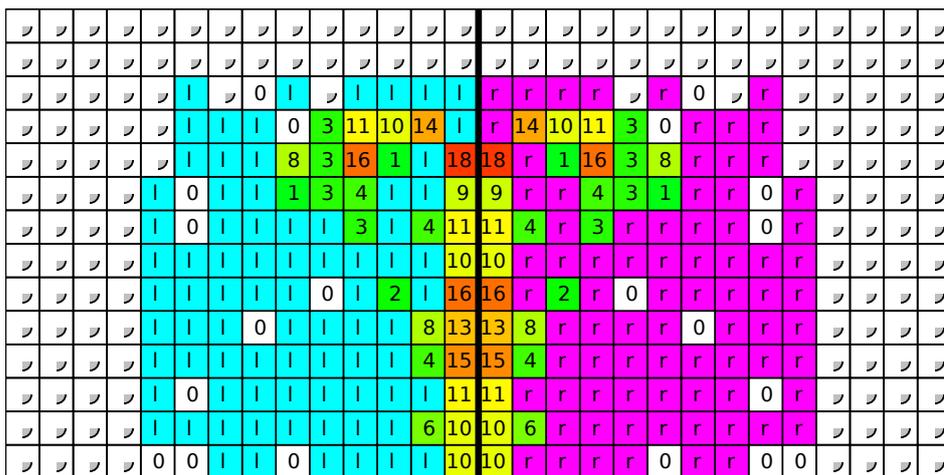


Abbildung 5.10: Visualisierung der gemeinsamen Bereiche für Modus 5

6 Ausblick und Fazit

Wie in der Einleitung beschrieben, ist Wahrnehmung wichtig. Diese Arbeit zeigt eindrucksvoll, dass man sich auch bewusst sein soll, wie das Wahrnehmungsorgan im Körper funktioniert, um es richtig zu nutzen.

Die Arbeit mit dem Ultraschallsensor ist eine Entwicklung in eine sinnvolle Richtung. Doch erst mit dem Verstehen der Grundlagen kann ein weiterführender Ansatz entstehen. In diesem Dokument werden Grundlagen zusammengetragen, daraus resultierende Annahmen überprüft und gegebenenfalls durch Alternativen ersetzt oder revidiert.

So lässt sich nun sagen, dass der Sendebereich eher einem Pils als einem Kegel ähnelt und sich in der Größe abhängig vom Modus unterscheidet. Auch ist der nutzbare Bereich mit $2,40\text{m} \times 4\text{m}$ in Tiefe und Breite kleiner.

Ebenso ist der überschneidende Sendebereich weitaus kleiner als erhofft. Der *Triangulierung über Kreisschnittpunkte* wird zwar Potential als unterstützendes Verfahren eingeräumt, doch genau aus diesem Grund als nicht praxistauglich bezeichnet.

Es wurden Tests entwickelt und Messungen damit durchgeführt, um den Einfluss des Trikots zu bestimmen. Diese Tests zeigten auf, dass die Trikots keinen Einfluss auf die Messwerte haben, jedoch dem Fußboden als Hindernis eine stärkere Beachtung zukommen sollte.

Auf der Grundlage der Testergebnisse lässt sich eine spieltaugliche Hinderniserkennung durch die Ultraschallsensoren entwickeln.

Der Ultraschall stellt ein gutes Mittel dar, um zusätzlich am Spielgeschehen mitzuwirken und mit einem geeigneten Algorithmus, z.B. einer modifizierten Version von B-Human, gute Ergebnisse zu erzielen.

Literaturverzeichnis

- [AirUltra] *Air Ultrasonic Ceramic Transducers 400ST/R100*. Dokument liegt auf der CD bei. URL: https://community.aldebaran.com/resources/project/motor-others-datasheet/US_400SR-ST100%20%28SONAR%29.pdf (besucht am 18. Aug. 2014).
- [AldV1.14] Aldebaran. *Aldebaran Dokumentation v1.14*. 2013. URL: <https://community.aldebaran.com/doc/1-14/index.html> (besucht am 1. Aug. 2014).
- [AldV2.1] Aldebaran. *Aldebaran Dokumentation v2.1*. 2014. URL: <https://community.aldebaran.com/doc/2-1/index.html> (besucht am 18. Aug. 2014).
- [Becker2002] Silvia Becker. *Grundlagen der Positionsbestimmung mit Partikelfiltern - Vertiefer-Seminar Geoinformation WS 02/03*. Nov. 2002. URL: http://www.ikg.uni-bonn.de/vorlesungsarchiv/Vertiefer_Geoinfo_02_03/Seminar_WS02_03/05-Silvia-Becker-Partikelfilter.ppt (besucht am 16. Sep. 2014).
- [B-HumanCode] B-Human. *BHumanCodeRelease*. URL: <https://github.com/bhuman/BHumanCodeRelease> (besucht am 12. Juli 2014).
- [Rules2012] RoboCup Technical Committee. *RoboCup Standard Platform League (NAO) Rule Book*. 2012. URL: <http://www.informatik.uni-bremen.de/spl/pub/Website/Downloads/Rules2012.pdf> (besucht am 18. Sep. 2014).
- [Rules2014] RoboCup Technical Committee. *RoboCup Standard Platform League (NAO) Rule Book*. 2014. URL: <http://www.informatik.uni-bremen.de/spl/pub/Website/Downloads/Rules2014.pdf> (besucht am 30. Aug. 2014).
- [Robocup] The Robocup Federation. *www.robocup.org*. URL: <http://www.robocup.org/> (besucht am 2. Sep. 2014).
- [Forschung2012] Referat Forschung. *Warum Fußballroboter nicht rempeln - Visuelle Hinderniserkennung in mobilen Systemen: die HTWK Leipzig auf der CeBIT 2012*. Jan. 2012. URL: [http://www.htwk-leipzig.de/index.php?id=18&tx_ttnews\[tt_news\]=4233](http://www.htwk-leipzig.de/index.php?id=18&tx_ttnews[tt_news]=4233) (besucht am 1. Sep. 2014).

- [Final2013] René van de Molengraft, Roel Merry, Carel van de Bunt, Ivo Jongsma, Barry van der Meer und Charlotte van den Heuvel. *Final Report RoboCup 2013*. Dez. 2013. URL: <http://www.robocup2013.org/wp-content/uploads/2013/12/Final-report-RoboCup-2013.pdf> (besucht am 2. Sep. 2014).
- [RobocupWiki] *Robocup Wiki*. URL: http://wiki.robocup.org/wiki/Standard_Platform_League (besucht am 1. Sep. 2014).
- [B-Human10] Thomas Röfer, Tim Laue, Judith Müller, Armin Burchardt, Erik Damrose, Alexander Fabisch, Fynn Feldpausch, Katharina Gillmann, Colin Graf, Thijs Jeffry de Haas, Alexander Härtl, Daniel Honsel, Philipp Kastner, Tobias Kastner, Benjamin Markowsky, Michael Mester, Jonas Peter, Ole Jan Lars Riemann, Martin Ring, Wiebke Sauerland, André Schreck, Ingo Sieverdingbeck, Felix Wenk und Jan-Hendrik Worch. *B-Human Team Report and Code Release 2010*. Techn. Ber. Deutsches Forschungszentrum für Künstliche Intelligenz und Universität Bremen, Okt. 2010. URL: <http://www.b-human.de/downloads/coderelease2012.pdf>.
- [B-Human12] Thomas Röfer, Tim Laue, Judith Müller, Michel Bartsch, Malte Jonas Batram, Arne Böckmann, Nico Lehmann, Florian Maaß, Thomas Münder, Marcel Steinbeckand, Andreas Stolpmann, Simon Taddiken, Robin Wieschendorf und Danny Zitzmann. *B-Human Team Report and code Release 2012*. Techn. Ber. Deutsches Forschungszentrum für Künstliche Intelligenz und Universität Bremen, 2012. URL: <http://www.b-human.de/downloads/coderelease2012.pdf>.
- [B-Human13] Thomas Röfer, Tim Laue, Judith Müller, Michel Bartsch, Malte Jonas Batram, Arne Böckmann, Martin Böschen, Martin Kroker, Florian Maaß, Thomas Münder, Marcel Steinbeck, Andreas Stolpmann, Simon Taddiken, Alexis Tsogias und Felix Wenk. *B-Human Team Report and Code Release 2013*. Techn. Ber. Deutsches Forschungszentrum für Künstliche Intelligenz und Universität Bremen, 2013. URL: <http://www.b-human.de/downloads/publications/2013/CodeRelease2013.pdf>.
- [Test:GitHub] Richard Stiller. *GitHub: Ultrasound bachelor thesis Testdata*. Alle Testdaten in Rohfassung und Aufbereitet zum Download verfügbar und liegen auf der CD bei. 2014. URL: https://github.com/Lexanius/ultrasound_ba_testdata.
- [Test:leer] Richard Stiller. *Ultraschall-Messungen zum Test leeres_feld*. Datei: leeres_feld.pdf, liegt auf der CD bei. 2014. URL: https://github.com/Lexanius/ultrasound_ba_testdata/blob/master/leeres_feld.pdf.

[Test:matrix-0-down] Richard Stiller. *Ultraschall-Messungen zum Test matrix_col_0_down*. Datei: matrix_col_0_down.pdf, liegt auf der CD bei. 2014. URL: https://github.com/Lexanius/ultrasound_ba_testdata/blob/master/matrix_col_0_down.pdf.