Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Diplomarbeit

Michael Manger

Design und Realisierung einer experimentellen Plattform für Roboterfußball

Fachbereich Elektrotechnik und Informatik
Department of Electrical Engineering and Computer Science

Michael Manger

Design und Realisierung einer experimentellen Plattform für Roboterfußball

Diplomarbeit eingereicht im Rahmen der Diplomprüfung
im Studiengang Technische Informatik
am Fachbereich Elektrotechnik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Kai von Luck
Zweitgutachter : Prof. Dr. Gunter Klemke

Abgegeben am 11. Mai 2004

Michael Manger

Thema der Diplomarbeit

Design und Realisierung einer experimentellen Plattform für Roboterfußball

Stichworte

mobile Roboter, ”omnidirectional” Antriebssystem, RoboCup Junior Soccer League

Kurzzusammenfassung

In dieser Arbeit wird eine experimentelle Plattform realisiert. Vor der Konstruktion des Roboters wurden Untersuchungen der schon verfügbaren Robotersysteme der RoboCup-Junior-Soccer-League durchgeführt und Verbesserungsvorschläge erarbeitet. Die Regeln dieser League bilden die Grundlage bei der Konstruktion des Roboters. Die Plattform ist stabil und in mehreren Ebenen unterteilt, das einen schnellen Auf- und Umbau ermöglicht. Das Antriebssystem ist mit ”omnidirectional” Rädern ausgestattet. Mit diesem Antriebssystem ist der Roboter in der Lage, in jede mögliche Richtung zu fahren, ohne sich drehen zu müssen. Der Sichtwinkel den in der RoboCup-Junior-Soccer-League eingesetzten Ballsensors wird durch eine drehbare Plattform verbessert. Damit wird eine bessere Kontrolle über den Spielball erreicht. Weiterhin sind Umgebungssensoren angebracht, die sich in alle Richtungen drehen können. Dies ermöglicht eine gute Kontrolle über die unmittelbare Umgebung.

Michael Manger

Title of the paper

Design and realization of an experimental platform for robot soccer

Keywords

mobile robots, ”omnidirectional” drive system, RoboCup Junior Soccer League

Abstract

This work describes the realization of an experimental platform. Before the construction of the robot examinations of the already available robot systems of RoboCup-Junior-Soccer-League and suggestions for improvement were worked out. The rules of this League form the basis. The platform is stable and divided into several levels to enable a quick construction and reorganization. The drive system is equipped with "omnidirectional" wheels. This drive system enables the robot to drive in every possible direction without having to turn. A rotatable platform improves the view angle of the ball sensors which are used in the RoboCup-Junior-Soccer-League. This improves the control over the match-ball. Additionally surrounding sensors are placed which are able to turn in every direction. This enables a good control over the surrounding area.

Danksagung

Es gibt viele Diplomarbeiten, die sich mit der Danksagung knapp halten. Doch ich finde, dass man sich bei den viele Leuten, die einem mit “Rat und Tat“ zur Seite standen, im einzelnen Bedanken muss.

Zuerst möchte ich mich bei meinem Betreuer ***Kai von Luck*** bedanken, der mir die Gelegenheit gab mich mit diesem Thema in meiner Diplomarbeit zu beschäftigen.

Ein großer Dank geht auch an meine Eltern, die mich moralisch und auch finanziell unterstützt haben. Denn ohne euch hätte ich es nicht geschafft mein Studium zu Ende zu bringen und bei meinen Großeltern, die immer an mich geglaubt und mich bei jeder Entscheidung unterstützt haben.

Ich möchte mich bei ***Mirco Gerling*** bedanken, der mir die Anregung gegeben hat dieses Antriebssystem mit dem “omnidirectional“ Rädern zu konstruieren.

Ein sehr großer Dank geht auch an die Firma **Interroll**, die mir 3 komplette Allseitenräder für diese Arbeit zur Verfügung stellten.

Auch ein sehr großer Dank geht an die fleißigen Helfer der Zentralen Werkstatt der HAW Hamburg (***Peter Svensson***, ***Bernd Bethke***, ***Hans-Peter Bensemann***, ***Michael Gaedke*** und ***Berthold Witte*** (Fachbereich Physik)), die mir beim Bau des Roboters, trotz nicht immer exakter Darstellung meiner Konstruktionszeichnungen oder Ideen, geholfen haben.

Bedanke muss ich mich auf jeden Fall auch bei ***Ilka Kahl***, ***Kai Steph***, ***Helge Lemcke*** und vor allem ***Clemens Kehren*** und ***Björn Jensen***, die mich bei dem Kampf gegen die vielen Fehlerteufels in dieser Diplomarbeit unterstützt haben.

Ich hoffe, ich habe hier keinen weiter namentlich vergessen und wenn doch dann tu es mir sehr leid...

Also damit bedanke ich mich herzlich bei allen, die mich in irgendeiner Form unterstützt haben.

DANKE, DANKE, DANKE!!!

Inhaltsverzeichnis

Abbildungsverzeichnis

1. Einleitung

Ist Roboterfußball der erste Schritt zur Entwicklung einer neuen, künstlichen Lebensform? Um diese Frage beantworten zu können, muss geklärt werden, was sich hinter Roboterfußball verbirgt und was Roboterfußball ist. Roboterfußball ist eine Disziplin einer jährlich stattfindenden Weltmeisterschaft für (teil-)autonome Roboter, die in verschiedenen Klassen und Softwareagenten unterteilt sind. Es ist eine internationale Entwicklung und Bildungsinitiative mit dem Ziel, die Entwicklung von künstlicher Intelligenz und Robotern zu fördern, sowie eine große Zahl von Technologien zu untersuchen und zu integrieren.
Die Idee, fußballspielende Roboter zu entwickeln, stammt aus dem Jahr 1992 von Professor Alan Mackworth[1] [AM'04]. Ein japanisches Forscherteam organisierte im Oktober 1992 einen Workshop in Tokyo, in dem erstmalig ernsthaft über die Realisierung dieses Fußballspiels zur Vermarktung von Technik und Wissenschaft diskutiert wurde. Im Juni 1993 startete ein Forschungsteam[2] einen ersten Wettkampf, was innerhalb eines Monats vor allem außerhalb Japans zu einer so großen Resonanz führte. Nach sehr vielen Nachfragen wurde dieses Projekt zu einer internationalen Initiative ausgebaut und die World Cup Initiative (kurz: RoboCup [RC'04]) gegründet. Einige Pioniere[3] auf diesem Gebiet der Forschung und Entwicklung beteiligten sich an dem Projekt und stellten ihre Forschungsergebnisse, ohne die das Projekt nie hätte realisiert werden können, zur Verfügung. Schon September 1993 wurde die erste Software (Soccer Server 0 (LISP), später Version 1.0 (C++)) für die Roboter erstellt.

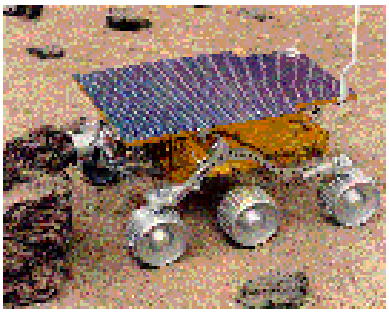


Abbildung 1.1.: Sojourner: autonomes Robotersystem bei der Mars Mission, 1997[NASA'04]

1997 war dann der Wendepunkt in der Geschichte der Robotik und Agentensysteme, denn im Mai diesen Jahres besiegte der IBM Großrechner Deep Blue [DB'97] den Schachweltmeister (Kasparov), am 4. Juli landete das erste autonome Robotersystem (Sojourner, siehe

[1] University of British Columbia, Canada
[2] In diesem Forscherteam sind unter anderem Minoru Asada, Yasuo Kuniyoshi und Hiroaki Kitano vertreten gewesen
[3] Itsuki Noda, Professor Minoru Asada, Professor Manuela Veloso, Peter Stone

Abbildung 1.1) der NASA [NASA'04] auf dem Mars (Pathfinder Mission), und die ersten offiziellen RoboCup Spiele fanden in Nagoya statt. Bereits 1998 gab es allerdings dann zwei konkurrierende Initiativen, die beide den Anspruch erhoben, Roboterfußball zu organisieren. Der von Japan (Dr. Hiroaki Kitano, Sony) organisierte RoboCup und die von Korea (Prof. Jong-Hwan Kim, KAIST) geleitete FIRA [FIRA'04]. Beide unterscheiden sich in den Größen der Roboter und in den Spielregeln. Aber unabhängig von einander ist das Ziel 2050 die richtigen Fußballweltmeister mit menschenähnlichen Robotern zu schlagen.

Diese Diplomarbeit ist in mehrere Kapiteln aufgeteilt, die auf einander aufbauen. Nach der im zweiten Kapitel angesprochenen Aufgabenstellung und Motivation des Verfassers dieser Arbeit werden im dritten Kapitel dem Leser die nötigen Grundlagen über Robotfußball vermittelt, um diesen mit der Materie vertraut zu machen. Im vierten Kapitel werden die beiden größten Initiativen (RoboCup, FIRA), die sich mit (teil-)autonome Roboter beschäftigen, vorgestellt. Anschließend wird im fünften Kapitel besondere, ausgewählte, existierende Robotersysteme vorgestellt. Daraufhin werden im sechsten Kapitel die Ziele einer Neuentwicklung mit Hilfe der vorgestellten Robotersysteme des fünften Kapitel erarbeitet und diskutiert. Des weiteren wird im siebten Kapitel einige Hardwarebauteile vorgestellt, die für eine Neukonstruktion eines Roboters interessant erscheinen könnte. Im achten Kapitel wird die Konstruktion und Realisierung des Roboterprototypen näher beschrieben, wobei auf den Hardware- und den Softwareanteil eingegangen wird. Zum Abschluss wird ein Ausblick auf weitere Entwicklungsmöglichkeiten des Projektes gegeben.

Hinweis zu Markennamen

2. Allgemeine Aufgabenstellung

2.1. Aufgabenstellung

In dieser Diplomarbeit wird eine experimentelle Plattform für einen Roboter realisiert, der in der Lange ist, an einem Roboterfußballspiel vollständig autonom teilzunehmen. Dabei muss der Roboter so konstruiert werden, dass er die Regeln der RoboCup-Junior-Soccer-League [RCJ'04] einhält und die bestehenden Roboter für diese League mit neuartigen Entwicklungsvorschlägen übertrift. Weiterhin wird versucht, eine Kommunikationsmöglichkeit zwischen Robotern zu implementieren. Die Kommunikation der Roboter wird mit Hilfe von PDA's verwirklicht werden. Dabei wird der PDA mit einem seriellen Anschluss an dem Controller-Board angeschlossen. Des Weiteren soll der PDA die kognitiven Fertigkeiten des Roboters übernehmen und das Controller-Board nur die motorischen Fertigkeiten ausführt. Jedoch soll in Betracht gezogen werden, dass die Roboter sich auch mit Hilfe einer externen Quelle (Kamera) orientieren können.

2.2. Motivation

In meiner Studienarbeit [MM'03], sowie im Rahmen von Projekten der Hochschule für Angewandte Wissenschaften Hamburg (HAW Hamburg) [Luck'04] habe ich versucht, mit dem Motorola Board (MIT-Board 6.270) [MIT'04] einen fußballspielenden Roboter zu verwirklichen. Bei der Studienarbeit habe ich durch unterschiedliche Design- und Programmieransetze versucht, die Ballerkennungsroutine zu perfektionieren. In dieser Diplomarbeit möchte ich mein erlerntes Wissen ergänzen und einen Roboter aufbauen, der sich an den Richtlinien der RobCup-Junior-Soccer-League [RCJ'04] orientiert und die bisherigen Probleme der verfügbaren Roboter in dieser League verbessert.

3. Grundlagen

In diesem Kapitel bekommt der Leser nach einer einführenden Beschreibung von Multiagentensystemen und mobilen Robotern die wichtigsten Grundlagen der einbezogenen Technologiebereiche für eine spätere Diskussion von geeigneten Hard- und Softwarekomponenten vermittelt. Auf Grund der hohen Komplexität eines mobilen Roboterentwurfs können die beschriebenen Grundlagen nur bis zu einem gewissen Detailliertheitsgrad vorgestellt werden. Für weitere Informationen wird auf weiterführende Literatur verwiesen [JJAF'96].

3.1. Kooperative autonome Multiagentensysteme

Dieser Begriff setzt sich zusammen aus

- "Agenten": Bezeichnet allgemein ein System, das Informationen aus seiner Umgebung empfangen kann, diese auswertet und in Aktionen umsetzt.
- "Autonome Agenten": So werden solche Agenten bezeichnet, die sich selbständig in ihrer Umgebung bewegen können und dabei unter wechselnden Bedingungen anpassen können.
- "Kooperativen autonomen Multiagentensystemen": Darunter versteht man mehrere von einander unabhängige Systeme, die mit ihrer Umwelt interagieren und die Ziele durch Zusammenarbeit erreichen.

3.2. Roboter

Das Wort "Roboter" stammt vom tschechischen Wort für Leibeigener bzw. Arbeiter ab. Benutzt wurde der Begriff erstmals in diesem Kontext in Karel Capeks 1921 erschienenen Theaterstück "Rossum's Universal Robots" [KC'20]. Nach Definition des Robot Institute of America 1979 ist ein Roboter ein programmierbares Mehrzweckhandhabungsgerät für das Bewegen von Material, Werkstücken, Werkzeugen oder Spezialgeräten. Der frei programmierbare Bewegungsablauf macht ihn für verschiedenste Aufgaben einsetzbar.
In der heutigen Zeit gibt es unterschiedliche Arten von Robotern, die z.B. in der Industrie eingesetzt werden, wo unzumutbare Arbeitsbedingungen herrschen oder dort, wo es für den Menschen gefährlich ist. Moderne Industrieroboter, wie sie zum Beispiel beim Bau von Autos eingesetzt werden, erledigen stupide Fließbandarbeit wesentlich genauer als ein Mensch und vor allem schneller. Selbst ein moderner Mikroprozessor wäre ohne einen Roboter nicht

mehr herstellbar. In anderen Bereichen des Lebens sind Roboter nun mehr ständige Begleiter der Menschen geworden, denn Forschungsroboter erkunden ferne Planeten oder Katastrophengebiete, dringen in Vulkane ein oder Abwasserrohren vor, erleichtern den Alltag, in dem sie staubsaugen, den Boden wischen oder Rasen mähen. Der Trend geht sogar dahin, dass im Kinderzimmer auch elektronische Haustiere wie der Roboter-Hund Aibo von Sony [AIBO] zu finden sind.
Viele angesehene Forscher sehen im Bereich der Unterhaltungselektronik ein großes Potential, um neuartige Roboter zu konstruieren und zu verbessern. Deshalb sind einige Forscher an Projekten, wie z.B. das Roboterfußballspiel beteilig. Nach der Definition des Begriffes “kooperative autonomen Multiagentensysteme“ handelt es sich deshalb nicht ausschließlich um Roboter, jedoch ist in der Welt des Roboterfußballs der Begriff “Roboter“ und “kooperative autonome Multiagentensysteme“ zu benutzen, denn im Roboterfußball handelt es sich um simulierte oder reale Roboter. Weiterhin kann man den Begriff “Roboter“ noch weiter spezifizieren, in “teilautonome“ und “(voll)autonome Roboter“. Ein “teilautonomer Roboter“ kann nicht alle seine Aufgaben gänzlich autark lösen, denn er ist auf externe Unterstützung angewiesen. So können seine Entscheidungen von Daten abhängig sein, die sich nicht über seine lokale Sensorik ermitteln lassen. Dies sind im Roboterfußball beispielsweise Positionsdaten, die durch eine globale Bildverarbeitung ermittelt und von einem Serverrechner zur Verfügung gestellt werden. Die “(voll)autonomen Roboter“ verlassen sich gänzlich auf ihre lokale Sensorik.

3.3. Navigation

Die Navigation eines autonomen mobilen Roboters erfordert Verfahren zur Bestimmung und Messung der Roboterposition, die im Idealfall vom Roboter selbst ausgeführt werden können. In diesem Abschnitt soll ein kurzer Überblick über verschiedene Verfahren und die dafür notwendige Sensorik gegeben werden. Die möglichen Verfahren zur Positionsbestimmung lassen sich unterteilen in absolute und relative Navigation, sowie Navigation mit Hilfe von Landmarken. Verfahren der absoluten Navigation wertet Sensordaten aus, die auf die absolute Position des Roboters in einem Koordinatensystem schließen lassen. Relative Navigation verwendet Sensordaten, welche die Veränderung der Roboterposition innerhalb eines bestimmten Zeitabschnitte beschreiben. Durch diese Veränderungen kann man auf die absolute Roboterposition zurück greifen. Beim Landmarkenverfahren besitzt der Roboter eine Karte seiner Umgebung mit gegebenen Hindernissen (siehe Kapitel 3.3.3).

3.3.1. Absolute Verfahren

Ein gutes Beispiel für ein absolutes Verfahren zur Positionsbestimmung stellt sicherlich das GPS (global positioning system) dar. Ein GP-System ermöglicht durch den Empfang von GPS-Satellitendaten unter anderem eine Positionsbestimmung auf der Erdoberfläche mit einer üblichen Genauigkeit von 10 bis 20 Metern. Es wird der Zeitversatz beim Empfang der Daten von mindestens drei Satelliten gemessen, so dass auf den Abstand des Empfän-

gers von den Sendepunkten und somit auf die Position des Empfängers geschlossen werden kann. Dieses Verfahren nennt man Trilateration. Falls nicht die Entfernung, sondern die Richtung der Sender bestimmt wird, spricht man von Triangulation.
Für dieses Verfahren werden zumeist Infrarot-Sender benutzt. Ein rotierender Sensorempfänger bestimmt die Winkel zu den einzelnen Sendern und kann daraus die absolute Position des Sensors ermitteln.

3.3.2. Relative Verfahren

Ein Beispiel für ein relatives Verfahren ist die deduktive Berechnung (Dead reckoning oder deduction reckoning), welches aus der Seefahrt kommt. Hier kann man mit Hilfe von einfachen, Messinstrumenten durch wiederholte Messung relativer Bewegungen die absolute Position berechnen. Dies geschieht in dem man mit den Messinstrumenten die Geschwindigkeit, die Ausrichtung und die Zeit misst. Aus Geschwindigkeit und Zeit lässt sich die zurückgelegte Strecke bestimmen, die mit Hilfe der Fahrtrichtung auf den Bewegungsvektor im gemessenen Zeitraum schließen lässt. Wird der auf diese Weise bestimmte relative Bewegungsvektor zur bisher berechneten Position addiert, lässt sich die aktuelle Position annäherungsweise bestimmen.
Für dieses Verfahren eignen sich alle Sensoren, die Daten aufgrund einer Positionsänderung des Objekts liefern, zum Beispiel Bewegung eines Antriebsrades oder die Drehung des Objekts. Ein häufig verwendeter Sensortyp sind sogenannte Encoder (siehe Kapitel 3.7.2), die an Rädern oder Motoren angebracht sind.

3.3.3. Navigation mit Landmarken

Bei dieser Art Navigation orientiert sich der Roboter mit Hilfe von Landmarken, die er in seiner Umgebung identifizieren kann. Dabei besitzt er eine Karte seiner Umgebung mit allen notwendigen Landmarken. Falls der Roboter zunächst keinerlei Wissen über seine Umgebung hat und sich erst in ihr orientieren muss, bedient sich der Roboter bestimmte Merkmale in seiner Umgebung und definiert sie als eine Landmarke, um so nach und nach ein Modell seiner Welt zu erstellen.
Für dieses Verfahren eignen sich zur Erkennung von Landmarken berührungslose, wie zum Beispiel Kameras, Entfernungssensoren oder Infrarotempfänger, sowie taktile Sensoren, die z.B. aus Mikroschaltern bestehen. Kameras sind in der Lage, ihre Umgebung detailliert zu erfassen und gegebenenfalls vorgegebene Landmarken zu erkennen oder neue Merkmale als Landmarke zu definieren. Falls Landmarken mit Infrarotleuchtfeuer ausgestattet sind, kann sich der Roboter mit Hilfe von Infrarotsensoren orientieren. Die Entfernungssensoren sind in der Lage ihre Umgebung mit Hilfe von Abstandsdaten zu interpretieren. Diese Sensoren werden oftmals mit Hilfe von Radencodern (siehe Kapitel 3.7.2) beim Aufbau der Karte unterstützt.

3.4. Antrieb

Im Bereich der mobilen Roboter erfolgen Bewegungen meist nicht geradlinig und mit konstanter Geschwindigkeit, sondern sind durch häufiges Abbremsen, Beschleunigen und Änderungen der Richtung geprägt. Daher müssen die Motoren des Roboters genügend leistungsstark sein, um den Roboter möglichst schnell beschleunigen und abbremsen zu können. Für den Antrieb von mobilen Robotern werden Gleichstrommotoren verwendet, wie sie auch im Modellbau eingesetzt werden. Für ihre Ansteuerung werden Elektronikkomponenten eingesetzt, die eine Motorsteuerung mit Hilfe der Pulsweitenmodulation ermöglichen.

3.4.1. Pulsweitenmodulation

Es gibt in dem Bereich Modellbau zwei unterschiedliche Verfahren, um eine Steuerung der Antriebsmotoren derart ermöglichen, dass eine Leistungsregelung der Motoren realisiert werden kann. Diese Verfahren sind als mechanische und elektronische Fahrregler bekannt. Ein mechanischer Fahrregler enthält einen niederohmigen Spannungsteiler, welcher die Versorgungsspannung vor dem Motor unterteilt. Dabei gilt, je höher der Anteil des Motors an der Versorgungsspannung ist, desto mehr Leistung steht dem Motor zu Verfügung. Der Nachteil des Verfahrens liegt im Verlust der restlichen Versorgungsspannung, die im Fahrregler in Wärme umgesetzt wird. Das in dem Bereich mobile Roboter häufig eingesetzte Verfahren ist ein elektronische Fahrregler, welche die Versorgungsspannung mit einer festen Frequenz pulst. Dabei wird dem Motor in schnellem Takt kurzzeitig die volle Versorgungsspannung zur Verfügung gestellt. Durch die Trägheit des Motors wird die Wirkung der Spannungsimpulse über die Zeit gemittelt. Die Veränderung des Verhältnisses zwischen den beiden Zuständen Spannung und keine Spannung wirkt für den Motor wie eine Gleichspannungsversorgung zwischen 0 V und der maximalen Betriebsspannung. Die elektronische Regelung benutzt als Eingangssignal ein PWM-Signal (PWM, pulse width modulation) und schaltet mit Hilfe dieses Signals die Versorgungsspannung der Motoren.

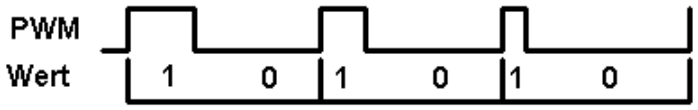


Abbildung 3.1.: Pulsweitenmodulation (PWM)

Ein PWM-Signal besteht aus einem Wechsel von logisch 1 und logisch 0 mit der Möglichkeit, die Breite des logisch Pulses zu verändern (siehe Abbildung 3.1). Mit diesem Verfahren kann die gewünschte Motorspannung digital kodiert und von der Leistungselektronik der Motorsteuerung ausgewertet werden.

3.4.2. Motorenauswahl

Bevor man einen Motor für den Antrieb des Roboters an einem Controllerboard anschließt, muß man einige Berechnungen und Informationen auswerten, denn bei dem Kauf eines Motors sind Angaben wie zum Beispiel Drehmoment, Drehzahl, Leistung usw. zu beachten. Um aus diesen Informationen den richtigen Motor zu ermitteln, müssen vorher bestimmte Eckdaten des Roboters erfasst werden. Dazu gehören Informationen wie Geschwindigkeit [v], Gewicht [m] und Raddurchmesser [d].

Zu erst muß betrachtet werden, welche Kräfte auf einem Roboter wirken, wenn dieser mit konstanter Geschwindigkeit fährt (siehe Abbildung 3.2). Dabei ist F_V die Kraft, die den Roboter vorwärts treibt (Vortriebskraft). F_R bezeichnet die Reibungskraft, die der Vortriebskraft entgegengesetzt wirkt. F_G ist die sogenannte Anziehungskraft, die sich auch mit ($F_G = m \cdot g$) vereinfachen lässt, wobei [g] die Fallgeschwindigkeit [$g = 9.81 \frac{m}{s^2}$] angibt.

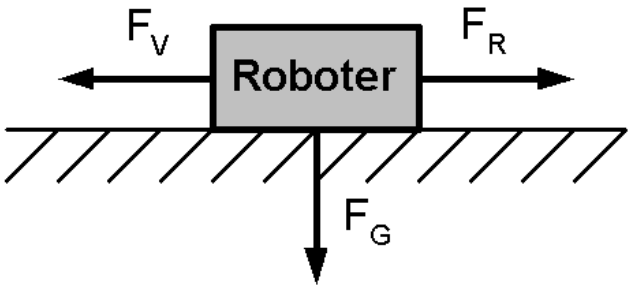


Abbildung 3.2.: Kräfte am Roboter bei konstanter Geschwindigkeit

Allgemein gilt:

$$F_R = \mu \cdot F_N \tag{3.1}$$

[μ] ist die Reibungszahl. Welche Werte man hier einsetzen muss, kann man in einem Tabellenbuch nachschlagen [MS'04]. In diesem Fall entspricht sie der Gleitreibung von Reifen auf Asphalt [$\mu = 0.5$]. F_N ist die sogenannte Normalkraft und berechnet sich aus ($F_N = F_G \cdot \cos\alpha$). Dabei ist $\cos\alpha$ der Neigungsfaktor und hier ± 0, da der Roboter beim Fußballspielen keine geneigte Ebene herauf bzw. herunter fahren muss, sondern stets in der Waagerechten bewegt. Damit gilt also:

$$\begin{aligned} F_R &= \mu \cdot F_G \cdot \cos 0 \\ &= \mu \cdot m \cdot g \end{aligned} \tag{3.2}$$

Da sich der Roboter mit gleichbleibender Geschwindigkeit bewegt, heben sich F_V und F_R gerade gegenseitig auf

$$F_V = F_G = \mu \cdot m \cdot g \tag{3.3}$$

Um nun die Leistung des Motors zu berechnen gilt:

$$P = F_V \cdot v \tag{3.4}$$

[v] steht für die zu erreichende Höchstgeschwindigkeit. Daraus resultiert:

$$P = \mu \cdot m \cdot g \cdot v \tag{3.5}$$

Die Anzahl der Umdrehungen pro Sekunde, die die Antriebswelle machen muss, damit man mit dem vorhandenen Raddurchmesser auf die gewünschte Geschwindigkeit kommt, ergibt sich aus:

$$n = \frac{v}{\pi \cdot d} \tag{3.6}$$

[n] gibt die erforderliche Drehzahl in Umdrehungen pro Sekunde an. Um mit diesen Angaben die Leistung berechnen zu können, gilt diese Formel:

$$P = M \cdot \omega \tag{3.7}$$

[M] ist dabei das Drehmoment bei der Winkelgeschwindigkeit [ω]. Die Winkelgeschwindigkeit ist eine andere Darstellungsform für die Drehzahl und berechnet sich aus:

$$\omega = 2 \cdot \pi \cdot n \tag{3.8}$$

Daraus resultiert:

$$P = M \cdot 2 \cdot \pi \cdot n \tag{3.9}$$

3.4.3. Servomotoren

Für viele Anwendungen im Bereich “mobile Roboter“ sind Servomotoren[1] durch ihre Fähigkeit, eine vorgegebene Position sehr schnell mit hoher Genauigkeit anzufahren und auch gegen Belastung zu halten, sehr interessant. Servomotoren bestehen aus einem kleinem Gleichstrommotor, einem Getriebe, Endanschlägen, über die die Welle nicht hinweg drehen kann, einem Potentiometer, um die aktuelle Position festzustellen und einem IC zur Positionssteuerung. Servos besitzen neben den üblichen Leitungen für Masse und Phase eine dritte Leitung, die sogenannte Steuer- oder Signalleitung. Über diese Leitung wird dem Servo die anzufahrende Position mitgeteilt. Dies geschieht nach der oben angesprochenen PWM-Methode: Eine bestimmte Impulslänge steht für eine bestimmte Position.

[1] Um einen Servomotor zu einem Antriebsmotor umzubauen, muss man den Servo leichten baulichen Veränderungen unterziehen. Eine gute Anleitung dazu ist in “Applied Robotics“ [EW’99] zu finden.

3.5. Antriebskonzepte

Beim Entwurf eines Roboters kann man aus vielen verschiedenen Antriebskonzepten wählen, von denen jedes seine eigenen Vor- und Nachteile besitzt. Hier werden die wichtigsten Antriebsformen vorgestellt.

3.5.1. Differenzialantrieb

Dieses Antriebssystem besteht aus zwei separat angetriebenen Rädern. Der Roboter hat damit die Möglichkeit, geradeaus zu fahren, auf der Stelle zu drehen und sich auf einer Kreisbahn zu bewegen. Bei diesem Antrieb ist die Balance ein Problem, denn neben den beiden Antriebsrädern müssen zusätzlich ein oder zwei Freilauf- oder Stützräder angebracht werden. Ob nun ein oder zwei Stützräder nötig sind, hängt von der Gewichtsverteilung des Roboters ab. Wenn sich der Roboter dreht, muss das Freilaufrad frei schwenkbar sein und einen möglichst großen Nachlauf haben, damit es leicht zur Seite wegklappen kann. Anstelle eines Freilaufrades kann man auch eine Kugel oder ähnliches als dritten Auflagepunkt benutzen. Die Kugel sollte auf einer Achse montiert sein, so dass sie sich bei Vorwärtsfahrt drehen kann. Bei seitlichen Bewegungen des Roboters rutscht die Kugel dann über den Untergrund. Man sollte generell darauf achten, dass der Großteil des Gewichtes auf den Antriebsrädern lastet. Ein Problem beim Differentialantrieb ist die Geradeausfahrt. Auf Grund bauartbedingten Toleranzen der Motoren, einer ungleichmäßigen Gewichtsverteilung des Roboters oder unterschiedlichem Untergrund unter dem einem und dem anderen Antriebsrad drehen sich die Motoren manchmal mit unterschiedlichen Geschwindigkeiten, obwohl an beiden die gleiche Spannung anliegt. Der Roboter driftet dann zu einer Seite ab. Die Motorgeschwindigkeit muss dynamisch angepasst werden. Dies geschieht durch eine Regelung, welche die Motorgeschwindigkeit während der Fahrt überwacht und gegebenenfalls ändert. Eine solche Regelung kann in Soft- oder Hardware realisiert werden.

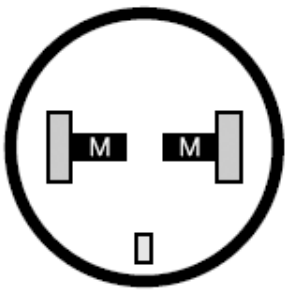


Abbildung 3.3.: Differentialantrieb

3.5.2. Kettenantrieb

Hier wird der Roboter nicht von Rädern, sondern, wie ein Panzer, von zwei Ketten angetrieben. Diese Antriebsform entspricht im Wesentlichen dem Differentialantrieb, nachteilig ist der größere Realisierungsaufwand und das höhere Gewicht. Zu beachten ist auch der zumeist größere Schlupf der Ketten gegenüber Rädern.

3.5.3. Dreiradantrieb

Ein Motor treibt über ein Differentialantrieb die beiden Hinterräder an, ein zweiter Servo- / Motor steuert das Vorderrad. Bei dieser Art des Antriebs muss die Geschwindigkeit der Antriebsräder nicht kontrolliert werden, um eine Geradeausfahrt zu erreichen. Es reicht aus, das Vorderrad geradeaus zu stellen. Nachteilig ist hier, das der Roboter nicht auf der Stelle drehen kann.

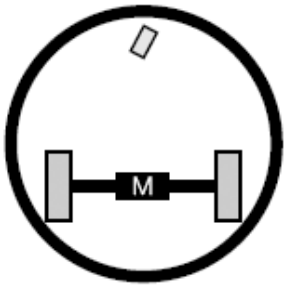


Abbildung 3.4.: Dreiradantrieb

3.5.4. Ackermannlenkung

Die von Autos bekannte Lenkung ist analog zum Dreiradantrieb, nur dass hier eben zwei synchron lenkbare Vorderräder vorhanden sind.

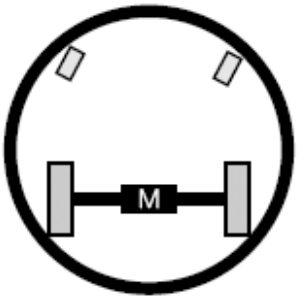


Abbildung 3.5.: Ackermannantrieb

3.5.5. “Omnidirectional“-Antrieb

Bei dieser Art Antriebssystem handelt es sich um einen Drei-Achsen-Antrieb, wobei die Achsen symmetrisch im Winkel von 120° angeordnet sind. Die Räder sind sogenannte “omnidirectional wheels“, die hier in Deutschland auch “Allseitenrollen“ genannt werden. Diese Räder haben die Fähigkeit, aktiv über den Motor angetrieben zu werden und zusätzlich passiv, über die integrierten Rollen in dem Rad, quer zur Antriebssachse zu rollen. Der Vorteil bei diesem System ist, dass durch eine unterschiedliche Geschwindigkeitsregelung der Motoren der Roboter in jede beliebige Richtung fahren kann, ohne dass er sich drehen muss. Ein weiterer Vorteil liegt darin, dass die Gewichtsverteilung des kompletten Roboters nun auf drei Rädern liegt und somit die Balance des Systems erhöht.

Abbildung 3.6.: “Omnidirectional“-Antrieb

3.6. Sensorik

Dieser Abschnitt behandelt die Grundlagen verschiedener Sensoren.

3.6.1. Reflexkoppler

Reflexkoppler ermöglichen die Unterscheidung von verschiedenfarbigen Oberflächen. So ist zum Beispiel das Erkennen einer weißen Fläche auf schwarzem Untergrund problemlos möglich. Auch die Unterscheidung verschiedener anderer Farbkombinationen (rot auf weiß, grün auf rot, usw.) kann mit geeigneten Reflexkopplern realisiert werden. Reflexkoppler erkennen, Unterschiede im Reflexionsverhalten verschiedenfarbiger Oberflächen. Wird eine farbige Fläche mit Licht bestrahlt, so wird nur der Teil des einfallenden Lichtes reflektiert, dessen Wellenlänge der Farbe der Oberfläche entspricht. Eine grüne Oberfläche reflektiert also nur 'grünes' Licht mit Wellenlängen von etwa 550 nm. Fällt Licht anderer Wellenlängen (zum Beispiel rotes Licht mit 700 nm) auf eine grüne Fläche, so wird nichts reflektiert. Das rote Licht wird absorbiert. Normales Umgebungslicht, sei es Sonnenlicht oder Licht von Lampen, ist aus Licht vieler verschiedener Wellenlängen zusammen gesetzt. Durch die Überlagerung der vielen einzelnen Spektralfarben erscheint es uns als weiß. Entsprechend reflektieren weiße Oberflächen alles einfallende Licht, egal welcher Wellenlänge. Oberflächen, die uns dunkelgrau oder schwarz erscheinen, reflektieren nur wenig oder gar kein Licht, sondern absorbieren nahezu die gesamte einfallende Strahlung.

3.6.2. Radencoder - Sensor

In manchen Disziplinen in Bereich der autonomen Roboter wird die genaue Messung des vom Roboter zurückgelegten Weges gefordert. Schon das einfache Geradeausfahren kann unter Umständen große Schwierigkeiten bereiten, wenn man keine Sensoren zur Bestimmung des zurückgelegten Weges einsetzt. So kann es passieren, dass die Roboter statt geradeaus zu fahren, eine Kurve beschreiten. Solche Probleme treten öfter bei Robotern mit zwei unabhängigen Antriebsrädern (Differentialantrieb) auf (siehe Kapitel 3.5.1). Dieses Fehlverhalten kann aber durch die Bestimmung der tatsächliche Drehgeschwindigkeit bzw. den tatsächlich zurückgelegten Weg für jedes der beiden Antriebsräder ausgeglichen werden. Treten Differenzen auf, fährt der Roboter offensichtlich nicht geradeaus und muss mittels Software nachgeregelt werden, indem man den einen Motor mit etwas höherer Spannung versorgt. Soll sich der Roboter um einen bestimmten Winkel drehen, ist es ebenfalls notwendig, den gefahrenen Weg für jedes Rad exakt zu kennen. Anhand des Radabstandes kann man die zu fahrende Strecke für jedes Rad bestimmen, wenn sich der Roboter um einen bestimmten Winkel drehen soll. Kennt man zu jeder Zeit den gefahrenen Weg und den aktuellen Drehwinkel des Roboters, ist auch eine Positionsbestimmung und Navigation möglich. Aus gefahrenem Weg und Winkel lässt sich eindeutig eine (x,y)-Koordinate errechnen, an der sich der Roboter im Moment befindet.
Sensoren zur Wegmessung werden als Radencoder bezeichnet und sind normalerweise auf der Antriebswelle eines Motors oder einer Achse befestigt ist. Es kann auch ein Getriebe

zwischen Motor und Rad gesetzt werden. Man unterscheidet grob zwei Arten von Radencodern: absolute Messwertgeber und inkrementelle Messwertgeber.
Absolute Messwertgeber liefern ein Signal zurück, das eindeutig die aktuelle Position der Antriebswelle bestimmt. Inkrementelle Messwertgeber liefern eine Folge von Impulsen. Jedes mal, wenn sich die Welle oder das Rad ein wenig dreht, ändert sich der Ausgangszustand von High auf Low oder umgekehrt. Die Anzahl der Impulse pro Zeiteinheit entspricht also der Drehgeschwindigkeit des Rades.
Die nachfolgend betrachteten Sensoren zählen alle zu dieser Kategorie, wobei die inkrementellen Radencoder sich noch einmal in mechanische und optische Sensoren unterteilen lassen.

3.6.2.1. Mechanische Radencoder

Die Mechanische Radencoder besitzen eine rotierende Scheibe, in die Schlitze gestanzt sind, welche zur Erzeugung eines elektrischen Kontaktes dienen. Bei der Drehung der Scheibe entsteht eine Folge von Low-High-Low Impulsen. Meist besitzen mechanische Encoder zwei solcher Scheiben, deren Schlitze etwas versetzt voneinander angebracht sind, denn dadurch ist eine Unterscheidung der Drehrichtung (vor- oder rückwärts) möglich. Bild 3.7 verdeutlicht die Funktionsweise.

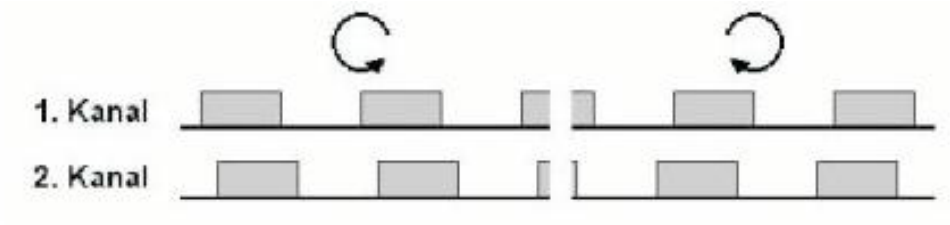


Abbildung 3.7.: Impulsfolge beim mechanischen Radencoder

3.6.2.2. Optische Radencoder

Optische Encoder stellen eine Alternative zu den mechanischen Sensoren dar. Es gibt zwei Arten optischer Radencoder: Photounterbrecher und Photoreflektoren.

3.6.2.3. Photounterbrecher

Photounterbrecher arbeiten wie eine Lichtschranke, indem ein Lichtsignal ausgesandt und empfangen wird. Zwischen Sender und Empfänger befindet sich eine rotierende Scheibe, die den Lichtstrahl abwechselnd unterbricht und hindurchlässt. Dadurch entstehen wiederum Impulse, die gezählt werden können. Die Scheibe kann eine Metall- oder Plastikscheibe sein, in die Löcher oder Schlitze gefräst sind. Man kann auch eine durchsichtige Kunststoffscheibe verwenden, bei der bestimmte Bereiche geschwärzt, also lichtundurchlässig sind. Üblichweise wird Infrarotlicht verwendet, um weniger anfällig gegen Umgebungslicht zu sein.

Trotzdem sollte man auf eine Abschirmung des Empfängers achten, so dass möglichst keine Fremdstrahlung empfangen wird. Man kann natürlich auch eine normale, im sichtbaren Bereich arbeitende LED verwenden. Fotowiderstände sind aber als Empfänger meist ungeeignet, da sie zu träge reagieren. Hier sollte also ein Fototransistor oder eine Fotodiode verwendet werden. Zur Erkennung der Drehrichtung muss eine zweite, etwas versetzt angebrachte Lichtschranke verwendet werden.

3.6.2.4. Photoreflektoren

Photoreflektoren arbeiten nach einem ähnlichem Prinzip, nur wird hier der Strahl nicht unterbrochen, sondern das reflektierte Signal ausgewertet. Als Sensor kann man einen Reflexkoppler verwenden, da die Reflexkoppler Hell und Dunkel unterscheiden können. Lässt man also eine Scheibe mit Schwarz-Weiß-Muster vor dem Reflexkoppler rotieren, erhält man die gewünschte Impulsfolge.

3.6.3. Entfernungssensoren

Ein entscheidender Punkt in nahezu allen Anwendungsbereichen mobiler Robotik ist das Messen von Entfernungen zu bestimmten Objekten, um eine Kollisionen mit Hindernissen zu vermeiden. Darüber hinaus gibt es aber noch andere Anwendungsmöglichkeiten, wie Suchen und Finden von Objekten im Raum oder der Orientierung und Positionsbestimmung in der Umgebung. Entfernungssensoren werden in drei Klassen unterteil:
Die einfachste Methode, ein vor dem Roboter liegendes Hindernis zu erkennen, ist der Einsatz eines Berührungssensors, welches im Roboterbereich auch oft als Bumper bezeichnet wird. Berührungssensoren sind häufig Mikroschalter, die einen Kurzschluss bzw. eine Unterbrechung des Kurzschlusses bei einer Berührung des Schalters verursachen. Die zweite Klasse von Abstandsmessern besteht aus Sensoren, die Objekte innerhalb eines bestimmten Entfernungsbereiches erkennen können und ein Signal abgeben, wenn die Entfernung des Objekts einen bestimmten Schwellwert unterschreitet. Die Sensoren liefern also nur ein binäres Ausgangssignal: Entweder “Objekt erkannt“ oder “kein Objekt“.
Die dritte Klasse liefert nun auch Informationen über die tatsächliche Entfernung eines Objektes. Sensoren dieser Kategorie kommen in den meisten Anwendungen zum Einsatz, denn sie sind flexibel einsetzbar und können mitunter die Aufgaben der Sensoren der ersten beiden Kategorien übernehmen.

3.6.4. Infrarot-Sensor

Infrarotnäherungs-Sensoren können in ihrer unmittelbaren Umgebung lichtreflektierende Objekte erkennen. Sie besitzen meist eine Sendediode, die Licht im infraroten Spektrum aussendet, und eine Empfängerdiode, welche die Intensität des reflektierten Lichts misst. Eine Klassifizierung der Objekte ist allerdings schlecht möglich, da der Sensor nur die Nähe eines Objekts ermittelt und keine Informationen über dessen Beschaffenheit liefert.

3.6.5. Ultraschall-Sensor

Ultraschall-Sensoren (auch Sonarsensoren genannt) basieren auf dem selben Prinzip, das auch von Fledermäusen verwendet wird. Ein kurzer, hochfrequenter Schallimpuls wird ausgestoßen und dessen Reflexion an Objekten in der Umgebung über einen Empfänger aufgenommen. Die Entfernung [s] kann man aus der Zeit [t] zwischen Aussenden und Empfangen des Impulses errechnen, da die Schallgeschwindigkeit [v] bekannt und konstant ist.
Es gilt: $s = v \cdot \frac{t}{2}$
Die Schallgeschwindigkeit [v] beträgt $343\frac{m}{s}$ bei $20°C$ Lufttemperatur. Sie steigt mit zunehmender Temperatur an. Bei $25°C$ ist [$v = 346\frac{m}{s}$] , bei $0°C$ nur noch $331\frac{m}{s}$.

3.6.6. Laser-Sensoren

Sensoren auf Laserbasis existieren in diversen Ausführungen. Grundsätzlich bestehen diese aus einer Laserdiode und einer Empfängerdiode. Die Laserdiode sendet entweder gepulstes oder stetiges Laserlicht aus. Die Empfängerdiode mißt die Reflexion des ausgesendeten Lichts auf entfernten Objekten und liefern auf diese Art Informationen über die Umgebung des Sensors. Durch eine Messung der Signallaufzeit kann bei gepulsten Lasern zudem die Entfernung zum Objekt bestimmt werden. Durch die große Reichweite der Laserdioden können Laser-Sensoren eine Reichweite von mehreren hundert Metern besitzen. Üblicherweise sind diese Sensoren aufgrund ihrer Komplexität größer und damit für den Einsatz in Miniaturrobotern ungeeignet.

3.6.7. Kamera-Sensor

Kamera-Sensoren liefern ein Bild ihrer Umgebung, das mit Hilfe von bildverarbeitenden Systemen ausgewertet werden kann. Kamera-Sensoren werden üblicherweise zur Objekterkennung eingesetzt. Zwei grundsätzliche Sensortechnologien werden für Kamera-Sensoren eingesetzt: Einerseits werden die üblichen CCD-Kameras (charge coupeled device) verwendet und andererseits existieren neuere Kameras in CMOS-Technologie, die eine geringere Versorgungsspannung benötigen als die CCD-Kameras.

3.6.8. Kompass-Sensor

Mit Hilfe eines Kompass-Sensors kann die absolute Ausrichtung des Roboters bestimmt werden, so fern man von einem beschränkten Arbeitsbereich auf der Erdoberfläche außerhalb der Umgebung der beiden magnetischen Pole, ausgeht. Es existieren sowohl mechanische als auch elektronische Sensoren. Mechanische Sensoren sind grundsätzlich träge und in ihren Abmessungen zu groß für den Einsatz in kleinen mobilen Robotern. Elektronische Sensoren sind kostengünstig in Form von integrierten Schaltkreisen verfügbar, die das Erdmagnetfeld durch integrierte Spulen ermitteln.

3.7. Echtzeitbedingungen

Ein mobiler Roboter muss während des Betriebs verschiedenste (Teil-) Aufgaben erfüllen. Dazu gehört vor allem die Wahrnehmung seiner Umwelt durch Sensoren, die Auswertung der resultierenden Sensordaten, die Steuerung und Regelung seiner Bewegungen und die Kommunikation mit anderen Robotern. Je nach Art der Aufgabe können die Ergebnisse von der Einhaltung gewisser Zeitbedingungen abhängig sein. Man unterscheidet:

- Harte Echtzeitbedingungen: erfordern die garantierte Einhaltung vorgegebener Zeitbedingungen, da bei Verletzung der Bedingungen eine Erfüllung der Aufgabe nicht möglich ist. Das heißt, dass ein Roboter zum Beispiel seine Sensordaten ständig aktualisiert haben muss, um sich seiner Umgebung anpassen zu können und so gegebenenfalls Kollisionen zu vermeiden.
- Weiche Echtzeitbedingungen: erfordern lediglich die statistische Einhaltung vorgegebener Zeitbedingungen. Eine zeitweilige Verletzung der Bedingung führt nicht direkt zu Fehlverhalten, sondern nur zu einer verminderten Qualität bei der Lösung der Aufgabe. Hier ist mit gemeint, dass ein Roboter zum Beispiel den Auftrag bekommt zu einer bestimmten Position zu fahren. Wie dieses Ziel erreicht wird ist nicht zeitspezifisch festgelegt.

3.8. Aspekte aus dem Software Engineering

3.8.1. Wasserfallmodell

Eines der häufigsten eingesetzten Projektmodelle ist das Wasserfall-Modell, welches die Zerlegung des gesamten Entwicklungsprozesses in mehrere, in sich abgeschlossene Phasen beinhaltet. Ein Beispiel einer solchen Phaseneinteilung zeigt Abbildung 3.8.

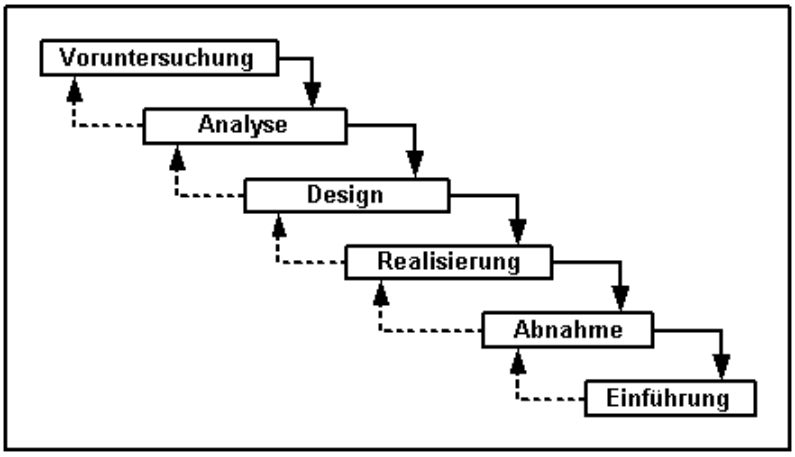


Abbildung 3.8.: Phaseneinteilung eines Entwicklungsprozesses

Diese Phasen werden nacheinander durchlaufen und mit einer Neueren kann erst dann begonnen werden, wenn die vorhergehende abgeschlossen ist. Der Phasenverlauf ist also strikt sequentiell. Zurückgeführt wird das Wasserfallmodell auf die Publikation von Royce [WR'70]. Der Autor verwendet dabei selbst nicht das Wort "waterfall model", allerdings stellt er sein Vorgehensmodell mit Hilfe von Grafiken dar, die intuitiv an einen Wasserfall erinnern. Der Autor fordert dabei explizit auf, den Schritt von einer Phase zur nächsten mindestens zweimal zu durchlaufen und damit gesicherte und abgenommene Ergebnisse zu schaffen, auf die man sich bei späteren Problemen wieder beziehen kann. Ebenso sieht er ein nochmaliges Durchlaufen der Phasen, je nach Komplexität der Aufgabenstellung vor.

3.8.2. Extreme Programming

Ein weiteres Modell ist das "Extreme Programming" [eXtrP'02], was ein relativ neues Grundvorgehen in der Softwaretechnik ist. Das Modell hat die Erfahrung zum Hintergrund, dass der Kunde die wirklichen Anforderungen zum Projektbeginn meist noch nicht komplett kennt. Er fordert Features, die er nicht braucht, und vergisst solche, die benötigt werden. Dabei wird auf einen strikten Anforderungskatalog des Kunden verzichtet, dennoch werden Kundenwünsche berücksichtigt, die sich während der Entwicklung noch ergeben. Statt des klassischen Wasserfallmodells durchläuft der Entwicklungsprozess immer wieder die Zyklen von Implementierung eines kleinen Schrittes, Tests und eventuellen Änderungen der Anforderungen (ständig verbesserte Prototypen). Nur die im aktuellen Iterationsschritt benötigten Features werden implementiert.

Es handelt sich um ein Gemisch aus verschiedenen Ideen, insbesondere

- Pair-Programming (Zwei Programmierer teilen sich eine Tastatur und Monitor - einer codiert, einer denkt mit)
- Integration in kurzen Abständen
- Ständiger Test (Alle, auch Kunden testen laufend)
- Laufende Refaktorisierung, ständige Architektur-Verbesserung

4. Roboterfussball

Roboterfußball hat sich in den letzten Jahren zu einer Standardproblemstellung der Wissenschaft entwickelt, wobei Anwendungsbereiche wie künstliche Intelligenz, Robotik, autonome Multiagentensysteme, Bildverarbeitung sowie Steuerungs- und Regelungstechnik zu erforschen sind. Beim Roboterfußball arbeiten intelligente Roboter kooperativ und koordiniert zusammen. Sie sind in der Lage, eine Aufgabe untereinander aufzuteilen, miteinander zu kommunizieren und autonom zu agieren. Die einzelnen “Spieler“ verfolgen gemeinsam ein konkretes Ziel: möglichst hoch zu gewinnen. Das Spiel findet in einem höchst dynamischen Umfeld statt. Jeder Roboter muss seine Aufgabe zwischen festen (Spielfeldbegrenzung) und bewegten (andere Roboter, Ball) Hindernissen ausführen. Es existieren mehrere Vereinigungen für den Roboterfußball, die für verschiedene Klassen Reglements festlegen und die Durchführung von Turnieren organisieren. In diesem Abschnitt werden die beiden größten Initiativen näher vorgestellt. Zum einen die RoboCup - Initiative [RC'04], an der sich weltweit ca. 150 Universitäten und Forschungszentren beteiligen und die damit die größte Vereinigung dieser Art ist, zum anderen die FIRA (Federation of International Robotsoccer Association) [FIRA'04], die sich für die kleinsten teil- bzw. vollautonomen Roboter verantwortlich schreiben.

4.1. RoboCup

Die Robot-World-Cup-Initiative (kurz: RoboCup) ist eine internationale Initiative für Forschung und Lehre. Diese wurde 1993 zuerst unter dem Namen Robot J-League von der Forschergruppe Minoru Asada, Yasuo Kuniyoshi und Hiroaki Kitano gegründet. Neben der Organisation und Durchführung von Wettkämpfen richtet RoboCup jährlich Konferenzen aus und bietet “educational programs“ für die Lehre an. Die Initiative gliedert sich in drei Bereiche. Der größte Bereich in dieser Initiative ist das Robo-Cup-Soccer. Es existieren die folgenden verschiedene Roboterklassen, darunter auch reine Simulatorklassen.
Die Regeln dieser Initiative wurden in den Jahren immer wieder verbessert und werden hier in der aktuellen Version 2004 kurz dargestellt. Für detailliertere Informationen der Spielregeln wird am Ende der jeweiligen vorgestellten Liegen ein Veiweis drauf hingewiesen.

4.1.1. Middle-Size-League

Ein Match, das maximal zwei mal 45 Minuten dauert, besteht aus mindestens zwei und höchstens 11 Spielern pro Team, wobei einer der Spieler die Torwartaufgabe übernehmen muss. Die Anzahl der zugelassenen Spieler hängt von der Größe der Roboter ab, die zwischen 30 cm x 30 cm und 50 cm x 50 cm variiert und diese Maße nicht unter- bzw. überschreiten darf. Die Höhe der Roboter beträgt mindestens 40 cm und höchstens 80 cm. Die grundlegende Farbe der Roboter sollte schwarz und nicht mit Material versehen sein, das Reflexionen verursacht. Kommunikation zwischen den Robotern in einem Team wird prinzipiell erlaubt, sowie zwischen den Robotern und entfernten Rechnersystemen. Es darf jedoch während des Spieles keine menschliche Einmischung stattfinden.Das Spielfeld, welches einen grünen Untergrund aufweist, hat eine Länge zwischen mindestens 8 m und höchstens 16 m und eine Breite von mindestens 6 m und höchstens 12 m. Üblicherweise wird auf einem Feld von 12 m x 8 m gespielt. Es wird meistens mit einem Lederball mit einem Durchmesser von 68-70 mm und einem Gewicht von 410-450 g gespielt. Regeln 2004: [RC-MSL'04]

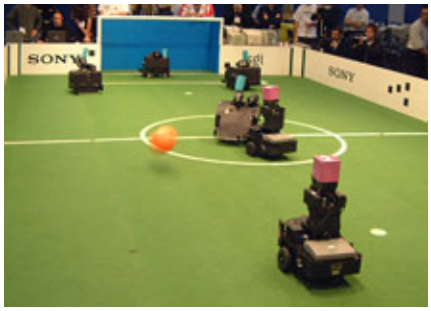


Abbildung 4.1.: Roboter der Middle-Size-League, [RC'04]

4.1.2. Small-Size-League

Bei dieser League, die auch unter der Bezeichnung F180 bekannt ist, stehen sich nicht mehr als fünf Roboter pro Mannschaft auf einem grünen, harten Platte mit einer Länge von 4.90 m und einer Breite von 3.40 m gegenüber [RC-SSLField'04]. Einer der Roboter muss die Funktionen des Torwart übernehmen und alle Roboter müssen eindeutig nummeriert sein, so dass der Schiedsrichter die Spieler identifizieren kann. Der Durchmesser der Roboter wurde, im Gegensatz zu den Anfangszeiten der Small Size League, von 30 cm auf 18 cm reduziert. Jedes Team kann auf ein globales Visionssystem zugreifen, welches oberhalb des Spielfeldes angebracht wird. Wenn die Kamera verwendet wird, dann dürfen die Roboter eine Höhe von 15 cm nicht überschreiten. Wie bei der Middle-Size-League ist eine Kommunikation zwischen den Robotern oder einer Kommunikationsschnittstelle außerhalb des Spielfeldes erlaubt. Auch hier ist es untersagt sich während des Spiels einzumischen. Ein Match kann bis zu zwei mal 10 Minuten dauern. Es wird mit einem orangefarbigen Golfball mit einem Durchmesser von etwa 43 mm und einem Gewicht von etwa 46 g gespielt. Regeln 2004: [RC-SSL'04]

Abbildung 4.2.: Roboter der Small-Size-League, [RC'04]

4.1.3. Sony-Legged-Robot-League

In dieser Liga kämpfen vierbeinige Roboter [AIBO] seit 1999 mit vier Spielern pro Mannschaft in der Sony-Legged-Robot-League um den WM-Titel. Das Besondere an dieser League ist, dass alle Roboter genormt sind. Der AIBO besitzt an jedem Bein jeweils drei Gelenke, der Kopfbereich kann um alle drei Achsen des Raumes bewegt werden, ein Gelenk am Maul und selbst der Schwanz besitzt zwei weitere Gelenke, die alle sowohl motorisch angesteuert als auch sensorisch ausgelesen werden können. Derzeit gibt es drei Typen von Robotern: Sony AIBO ERS-210, Sony AIBO ERS-210A und Sony AIBO ERS-7, die alle für diese League geeignet sind. Der Sony AIBO ERS-7 wird dennoch überwiegend benutzt, da er einen schnelleren Prozessor und eine deutlich bessere Kameraauflösung besitzt. Da der Hauptsensor eine Kamera ist, sind alle Objekte auf dem Spielfeld farbig codiert. Dabei sind die Roboter entweder blau oder rot zu verkleiden. Die grüne Spielfläche hat eine Länge von 4.20 m und eine Breite von 2.70 m. und wird mit einer 30 cm hohen Wand umrandet. Die einzige erlaubte Umbauarbeit beim AIBO ist es, eine drahtlose Kommunikationskarte einzubauen, damit sich die Roboter untereinander verständigen können. Ein Match ist in zwei Hälften á 10 Minuten aufgeteilt, wobei zwischen den beiden Hälften eine 10 minütige Pause eingehalten wird. Es gibt bei dieser League kein Unentschieden, deshalb wird bis zum Elfmeterschießen nach einer 5 Minuten Verlängerung weiter gespielt. Bei Unterbrechungen wird die Zeit angehalten. Es wird mit einem orangefarbigen Ball gespielt. Regeln 2004: [RC-S4L'04]

Abbildung 4.3.: Roboter der Sony-Legged-Robot-League, [RC'04]

4.1.4. Humanoid-League

In dieser recht neuen Liga sollen humanoide Roboter auf zwei Beinen gegeneinander spielen, zuerst in Mannschaften zu je drei Spielern, später aber in Mannschaften mit elf Spielern. Die League wird unterteilt in die Kategorien bis 60 cm, 120 cm und 180 cm Größe. Jedoch mussten bei der RoboCup Weltmeisterschaft in Padua 2003 diese Art Roboter zuerst nur einfachere Aufgaben bewältigen, z.B. auf einem Bein stehen oder den Ball erkennen. Neuere Regeln sind zu diesem Zeitpunkt noch nicht zu finden [RC-HL'04].

Abbildung 4.4.: Roboter der Humanoid-League, [RC'04]

4.1.5. Simulation-League

In der Simulation League treten elf so genannte Agentenprogramme pro Mannschaft auf dem von einem "Soccer Server" simulierten Spielfeld (105 x 69 m) an. Dabei können sich die virtuellen Spieler ganz auf Strategie und Kooperation konzentrieren. Der virtuelle Schiedsrichter pfeift auch Abseits, was heute noch eine Besonderheit der Simulationsliga ist.

4.1.6. RoboCup-ELeague

2003 wurde bei der RoboCup Weltmeisterschaft in Padova diese neue Soccer-League gegründet. Diese hat das Ziel auch kleinere Teams, die nicht die Ressourcen der großen Teams der Middle Size League oder der Small Size League besitzen, eine Chance zu geben bei solchen Weltmeisterschaften mit zu spielen. Die neuen Regeln beruhen zum größten Teil auf den Regeln der Small Size League. Um die Kosten und den Aufwand bei dieser League zu reduzieren, stehen beiden Teams die Nutzung einer gemeinsamen Bildverarbeitung, welche am Rande des Spielfeldes steht, zur Verfügung. Dabei sollen nicht nur die Bildverarbeitung standardisiert werden, sondern auch die Kommunikationsschnittstelle zwischen den Robotern. Die Plattform der Roboter ist bis zum heutigen Tag noch nicht ganz geklärt, jedoch versucht man auch hier, eine Standardplattform zu integrieren, um die Kosten zu reduzieren. Dabei wird überlegt, eine Einschränkung über Ausstattung und Prozessorleistung des Roboters zu verhängen. Es sollen in einem Spiel, welches 2x10 Minuten dauert, nicht mehr als vier Roboter pro Team zu gelassen sein, wobei einer der vier Roboter die Aufgabe des Torwartes übernehmen muss. Die Roboter dürfen einen Durchmesser von 22 cm und eine

Höhe von 15 cm nicht überschreiten. Vorläufig hat das Spielfeld eine Länge von 2.74 m und eine Breite von 1.52 m. Der Spielball ist ein orangefarbiger Golfball mit einem Durchmesser von 45 mm. Regeln 2004: [RC-EL'04].

4.1.7. RoboCup-Junior-League

Der RoboCup-Junior [RCJ'04] ist eine projektorientierte Bildungsinitiative der RobotWorldCup-Initiative, die lokale, nationale und internationale Roboter-Veranstaltungen sponsert. Sie soll Schüler und Studenten, die sonst nicht die Möglichkeit haben, mit Robotern zu arbeiten, an Roboter heranführen. Die Roboter werden aus handelsüblichen Baukästen, wie z.B. LEGO MINDSTORMS [LEGO] gebaut. Das Hauptziel dieses Projektes liegt in der Erziehung und Bildung von Kindern. Dieses wird durch die Zusammenarbeit in Teams und das gemeinsame Entdecken interessanter Technologien reicht. Auch die RoboCup-Junior-League ist in verschiedene Klassen aufgeteilt:

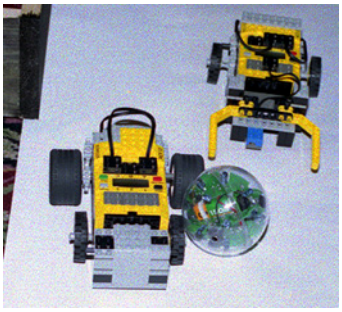


Abbildung 4.5.: Roboter der RoboCup-Junior-Soccer-League, [RC'04]

4.1.7.1. RoboCup-Junior-Soccer:

Hier spielen autonome Roboter allein oder in Zweierteams gegeneinander Fußball. Das Feld hat beim Spiel 1 gegen 1 eine Länge von 1.19 m und eine Breite von 0.87 m. Beim Spiel 2 gegen 2 ist die Spielfläche 1.83 m x 1.23 m groß. Beide Spielflächengrößen haben einen Graustufenverlauf als Untergrund und sind mit einer Wand mit einer Höhe von 14 cm umgeben. Der 105 g leichte Ball ist ein Infrarot Leuchtball (Roboball MK2) von der Firma Wiltronics und hat einen Durchmesser von ca. 8 cm. Der Durchmesser der Roboter darf bei einem Match 1 gegen 1 nicht mehr als 18 cm überschreiten, jedoch beim Match 2 gegen 2 ist ein Durchmesser von bis zu 22 cm erlaubt. Ein Spiel dauert 2x10 Minuten mit einer 5 Minutenpause zwischen den Spielhälften. Regeln 2004: [RC-JS'04]

4.1.7.2. RoboCup-Junior-Rescue:

In dieser League werden Opfer von Robotern aus künstlichen Katastrophenszenarien gerettet. Dabei gibt es verschiedene Schwierigkeitsstufen zu bewältigen, von einfache Linienverfolgung auf flachem Untergrund bis zu jährlich wechselnden, unebenen Geländeaufbauten,

wo die Robotern ihren Weg an Hindernissen vorbei suchen müssen. Die Roboter sind meistens Konstruktionen aus LEGO MINDSTORMS. Dennoch sieht man in den letzten Jahren auch verstärkt Robotersysteme, die nicht aus den üblichen Komponentenbauteile bestehen. Regeln 2004: [RC-JR'04]

4.1.7.3. RoboCup-Junior-Dance:

Bei dieser Veranstaltung sollen die Teams einen oder mehrere Roboter konstruieren, die nach Musik durch eine nicht mehr als 2 Minuten andauernde Vorführung die Jury cleveres Design und Bewehgungsideen beeindrucken. Regeln 2004: [RC-JD'04]

Abbildung 4.6.: Roboter der RoboCup-Junior-Dance-League, [RC'04]

4.1.8. RoboCup Rescue League - Die Retter

RoboCup-Rescue, welches seit 1992 im RoboCup durchgeführt wird, ist eine Simulationsumgebung zur Entwicklung und Beurteilung von Rettungsstrategien bei größeren Katastrophen. Im Gegensatz zu interaktiven Spielen, in denen die Aktivitäten von Agenten (Avatare) durch den User gesteuert werden, ist RoboCup Rescue ein Spielszenario, in dem Roboter autonom handeln. Dazu gilt es, Roboter zu implementieren, die z.B. nach einem Erdbeben als Ambulanz-, Feuerwehr- und/oder Polizeiteams in der Lage sind, Zivilisten zu retten, Brände zu löschen und Straßen zu räumen. Die Ziele dieser League ist es, die Forschung und Entwicklung im Bereich des Katastrophenschutzes zu fördern und autonome, intelligente Agentensysteme für Katastrophenszenarios zu entwickeln. Dabei müssen die autonomen Roboter bei diesen Veranstaltungen der RoboCup-Rescue-League in einem eingestürzten Haus, das mit Hindernissen wie z.B. zugeschütteten Türen, Geröll oder umgestürzten Möbeln ausgestattet ist, Opfer finden, ihre Situation, Zustände und Standorte den Teams mit Hilfe einer Karte übermitteln, wobei die Opfer durch Puppen mit Körperwärme und menschlichen Geräuschen dargestellt werden. Die Szenarien, die nach Referenztestarenen von der "U.S. National Institute of Standards and Technology (NIST)" [NIST'04] für städtische Such- und Rettungsoperationen entwickelt wurden, sind in drei Schwierigkeitstufen aufgeteilt, wobei im leichten Szenario die Positionen der Opfer und die Umgebung detailliert bekannt ist bis zu einem unbekannten und mit zusätzlichen Überraschungen bestückten Szenario. Jedes Team hat für die Szenarien 20 Minuten Zeit, um so viele Opfer wie möglich zu finden,

ohne weitere Opfer durch ungeschickte Navigation im Katastrophengebiet zu verursachen. Auch in dieser League existiert neben der Liga für reale Roboter eine reine Simulationsliga. Hier haben die Teams die Aufgabe, in einem virtuellen Katastrophenszenario, z.B. einer simulierten Erdbebensituation in einer Großstadt, Einsätze von Rettungskräften in Echtzeit zu planen und zu koordinieren, um möglichst effektiv verletzte Opfer zu bergen, blockierte Straßen zu räumen und die Ausbreitung von Bränden zu verhindern.Regeln 2004: [RC-R'04], [RC-RSim'04]

Abbildung 4.7.: Roboter der RoboCup-Junior-Rescue-League, [RC'04]

4.2. FIRA

Die FIRA (Federation of International Robotsoccer Association) wurde im September 1995 von Jong-Hwan Kim am KAIST (Korea Advanced Institute of Science and Technology) in Korea gegründet und ist im asiatischen Raum weit verbreitet. In dieser Initiative werden Spiele in 6 verschiedenen Kategorien abgehalten: MiroSot (Small, Middle und Large League), NaroSot, RoboSot, HuroSot, KheperaSot, and SimuroSot (Large und Middle League). Alle Regeln der hier aufgeführten League findet man auf der offiziellen Seite der FIRA [FIRA'04].

4.2.1. MiroSot

In dieser Kategorie sehen die Roboter aus wie Würfel mit einer maximalen Kantenlänge von 7.5 cm. Als Ball dient ein orangefarbener Golfball. Ein Spiel dauert 2x5 Minuten netto, das heißt bei Spielunterbrechungen läuft die Zeit nicht weiter. Es gibt derzeit drei MiroSot Klassen:
In der "Small League" spielen 3 gegen 3 auf einem 150 cm x 130 cm großen Spielfeld.
In der "Middle League" spielen 5 gegen 5 auf einem 220 cm x 180 cm großen Spielfeld.
In der "Large League" spielen 7 gegen 7 auf einem 280 cm x 220 cm großen Spielfeld.
Die Roboter werden durch ein Hostsystem unterstützt, welches hauptsächlich für die globale Bildverarbeitung zuständig sein soll. Über dem Spielfeld in einer Höhe von ca. 2 m ist zu diesem Zweck eine Farb-CCD-Kamera montiert, die das gesamte Spielfeld überblickt und pro Sekunde 30 bis 60 Bilder macht. Zur Unterscheidung der Roboter sind an deren Oberfläche Farbmarkierungen angebracht. Die Kommunikation zwischen dem Hostsystem und den Robotern wird üblicherweise durch ein Funksystem realisiert, bei dem sich die Mannschaft im Voraus auf die zu verwendenden Funkfrequenzen einigen. Während des Spiels darf nicht in den Spielverlauf eingegriffen werden. Die Systeme, bestehend aus den Robotern plus dem Hostcomputer, müssen das Spiel autonom durchführen. Das Spiel wird allerdings durch einen Schiedsrichter überwacht, der bei Regelverstößen eingreifen kann und für einen fairen Spielverlauf sorgen soll. Im diesem Fall dürfen die Teambetreuer die Systeme manuell stoppen und die resultierenden Feldpositionen der Roboter setzen.

Abbildung 4.8.: Roboter der MiroSot-League, [FIRA'04]

4.2.2. NaroSot

Hier handelt es sich um die kleinsten Vertreter der Roboter, denn sie sind in dieser Klasse nur 4 cm x 4 cm x 5.5 cm groß und spielen in einer Teamstärke von Robotern auf einem 130 cm x 90 cm großen Spielfeld gegeneinander. Als Ball dient ein orangefarbener Tischtennisball. Die Steuerung der Roboter ist identisch mit der MiroSot-Klasse ab.

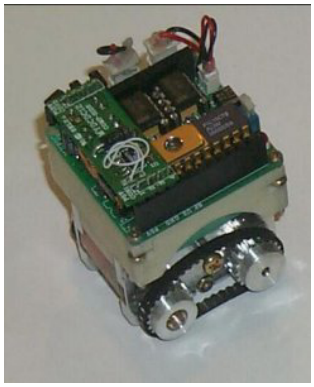


Abbildung 4.9.: Roboter der NaroSot-League, [FIRA'04]

4.2.3. RoboSot

Die Spieler sind in dieser Klasse maximal 20 cm x 20 cm x 40 cm groß. Ein Team kann aus 1 bis 3 Robotern bestehen. Das Spielfeld hat eine Größe von 220 cm x 180 cm. Die Roboter können vollständig oder teilweise autonom sein. Als Ball dient ein gelber Tennisball.

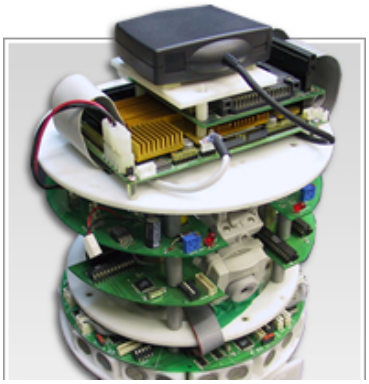


Abbildung 4.10.: Roboter der RoboSot-League, [FIRA'04]

4.2.4. KheperaSot

Bei dieser Kategorie dürfen die Roboter maximal einen Durchmesser von 6 cm haben. Sie müssen vollständig autonom agieren. Es spielt 1 gegen 1. Das Spielfeld ist 105 cm x 68 cm groß. Als Ball dient ein weißer oder gelber Tennisball.

Abbildung 4.11.: Roboter der KheperaSot-League, [FIRA'04]

4.2.5. HuroSot

Die Roboter in dieser Klasse müssen sich auf zwei Beinen fortbewegen und dürfen eine Höhe von 40 cm und einen Durchmesser von 15 cm nicht überschreiten. Es dürfen maximal 3 Spieler in einem Team sein.

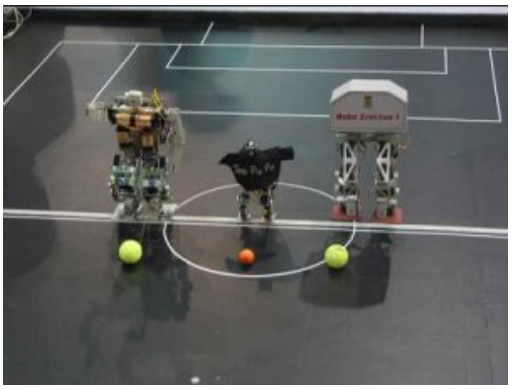


Abbildung 4.12.: Roboter der HuroSot-League, [FIRA'04]

4.2.6. SimuroSot

In dieser Liga werden Roboter wie auch bei der RoboCup-Initiative durch Softwareagenten repräsentiert, die mit einem Simulatorprogramm verbunden werden und einen Spieler innerhalb des Simulators steuern. Der Simulator vermittelt den Softwareagenten ein funktionierendes, reales Roboterfußballsystem. Aufgrund dieser Idealisierung ist es wesentlich einfacher möglich, Algorithmen zu testen und komplexe Lernstrategien zu erproben, die zum Beispiel während der Entwicklungs- und Trainingsphase eine hohe Zahl von Lernzyklen benötigen.

Abbildung 4.13.: Roboter der SimuroSot-League, [FIRA'04]

5. Existierende Robotersysteme

In diesem Kapitel soll ein Überblich über verschiedene Robotertypen, deren Fähigkeiten und Funktionsweisen dargestellt werden. Dabei werden im Speziellen auf die Robotersystemen der RoboCup-Junior-Soccer-League eingegangen. Des Weiteren werden hier auch Roboter einer anderen League kurz vorgestellt. Dies kann und soll nur bis zu einem gewissen Grad geschehen, um die vom Autor später getroffenen Design-Entscheidungen zu verdeutlichen und nachvollziehbar zu machen.

5.1. Roboterbaukästen von LEGO MINDSTORMS

In der RoboCup-Junior-Soccer-League [RCJ'04] werden hauptsächlich Roboter mit den Roboterbaukästen von LEGO MINDSTORMS [LEGO] konstruiert. Diese Baukästen beinhalten einen programmierbaren LEGO-Baustein (auch Programmable Brick oder RCX-Baustein genannt) im typischen LEGO-Gehäuse, einen Lichsensor, zwei Berührungssensoren, zwei 9V-Motoren und diverse Bauelemente. Der RCX-Baustein ist ein multitaskfähiger Hitachi H8/3292-Microcontroller mit 16-KB-ROM und 32-KB-RAM. Dieser Baustein verfügt über drei Analogeingänge mit 10-Bit-A/D-Wandlern für Sensoren, drei pulsweitenmodulierte Ausgänge für Motoren oder Lampen, fünf frei wählbare Programme, ein LCD-Display, vier Steuertasten und eine Infrarotschnittstelle zur Kommunikation. Am Anfang sind es recht einfache Modelle, die meistens mit dem Konstruktionshandbuch des Roboterbaukasten erstellt werden. Später werden diese Typen immer ausgeklügelter. Jedoch sind diese Art Roboter durch ihr geringe Anzahl von Ein- und Ausgängen für Sensoren und Motoren recht begrenz. Durch diese begrenzten Sensormöglichkeiten der Robotertypen wird häufig die Spielstrategie des "Bolzeifers" eines Kindes nachgeahmt, d.h. dass alle Roboter zu erst den Ball suchen und direkt darauf zu steuern, um gegebenenfalls das Tor zu treffen, welches meistens auf der falschen Seite ist. Schon beim einfachen Spiel 1 gegen 1 wird die Limitierung der Fähigkeiten der Roboterbaukästen deutlich und demonstriert, dass diese Roboter der "rechte Ballinstinkt" fehlt. Beim Spiel 2 gegen 2 wird versucht, den zweiten Roboter als Torwart zu programmieren, der gegebenenfalls nur vorm Tor hin und her fährt. Dabei kann es passieren, dass, wenn der Torwart mit einem Gegen- / Mitspieler kollidiert, er vom rechten Weg abkommt und quer über das Spielfeld fährt. Viele der Roboter drehen sich orientierungslos im Kreis auf der Suche nach dem Ball, schießen dabei Eigentore oder blockieren die Mitspieler der eigenen Mannschaft. Um das Problem der in ihrer Fähigkeit/Mächtigkeit begrenzten Roboterbaukästen der LEGO MINDSTORMS-Familie zu verstehen, muss man genauer darauf eingehen, warum die Konstruktion eines Roboters, der Fußballspielen soll, so schwierig ist.
Ein Roboter ist, im Gegensatz zu einem Computer, der sich vollständig deterministisch verhält (sprich: alle Daten liegen in digitaler Form mit beliebiger Genauigkeit vor), ein dynami-

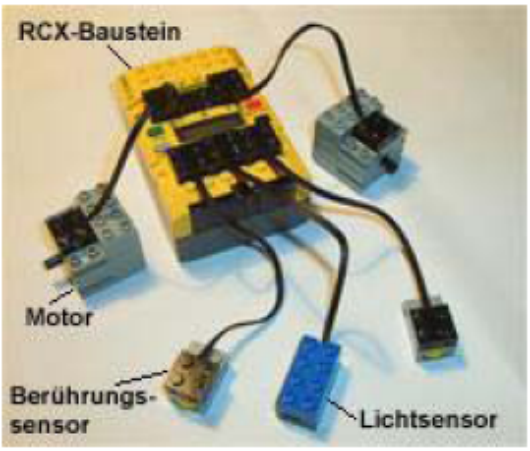


Abbildung 5.1.: Der RCX-Baustein von LEGO MINDSTORMS mit Motoren und Sensoren, [LEGO]

sches System, welches sich in einer hoch komplexen dynamischen Umgebung bewegt. Alle Daten werden von Sensoren gemessen, deren Genauigkeit bei guter technischer Ausstattung sehr hoch sein kann, jedoch trotzdem Messfehleren unterliegen. Auch die Aktionsausführung bewirkt Fehler, welche durch die mechanische Bewegung der Motoren entstehen. Da sich die Fehler summieren, kann niemals davon ausgegangen werden, dass die Messung von Sensorenwerten genau zu dem gewünschten Ergebnis führt. Sowohl die Sensoren als auch die Motoren der verwendeten Roboterbaukästen der LEGO MINDSTORMS-Familie stellen sich als anfällig heraus. So erhalten die Roboter zum Beispiel ungenaue Werte von den Lichtsensoren oder fahren nicht geradeaus, obwohl beide Motoren in der Programmierung mit der gleichen Leistung angesteuert werden. Auch die Mechanik ist nicht einfach zu handhaben: zum Einen müssen die Roboter innerhalb der vorgegebenen Maße der RoboCup-Junior-Soccer-League konstruiert werden, zum anderen müssen sie so stabil konstruiert sein, damit bei einer Berührung mit einem anderen Roboter auf dem Spielfeld keine Schäden entstehen. Bei dem überwiegenden Teil der Roboter in der RoboCup-Junior-Soccer-League basiert die Plattform auf einer Differentialsteuerung (siehe Kapitel 3.5). Mit diesen Antriebsrädern kann der Roboter angetrieben und gelenkt werden, wodurch der Roboter sich um den Mittelpunkt der Antriebsachse drehen kann und dadurch sehr wendig ist. Da beim RoboCup-Junior-Soccer-League mit einem besonderen Ball (siehe Kapitel 7.3) gespielt wird, benötigen die Roboter noch spezielle Sensoren, um den Ball zu erkennen. Dadurch reduziert sich die Anzahl der Sensorenvielfalt dieser Roboter noch weiter und ist nun begrenzt auf einem Berührungssensor, ein Lichtsensor (um die Grauschattierung der Spielfläche zu identifizieren) und den Sensor für die Ballerkennung.

5.2. Erweiterte Bauteile für das RCX

Das größte Manko der Roboterbaukästen der LEGO MINDSTORMS-Familie liegt darin, dass dem RXC nur wenige Ein- und Ausgänge zur Verfügung stehen. Dieses Problem wurde von einigen Diplomanden, darunter Dietmar Cordes und Gunther Lemm [DCGL'04], durch die Entwicklung eines Multiplexer behoben. Dieser Multiplexers[1] hat als Prozessoreinheit einen MSP-430F1121A. Die Hauptplatine des Lepomux besitzt eine Schnittstelle, um verschiedene Module für unterschiedliche Anwendungszwecke zu integrieren. Dabei beinhaltet das Sensor-Modul vier Anschlussmöglichkeiten für Infrarot Entfernungssensoren, die speziell für die Sharp-Sensoren GP2D12 [SHARP] angepasst sind, je vier LEGO-kompatible Anschlüsse für diverse Sensoren und Motoren der Firma LEGO, drei Eingänge für Beacon-Sensoren (siehe Kapitel 7.3) zur Erkennung von IR-Leuchtfeuern und vier Servo Ausgänge. Zusätzlich sind 8 LEDs auf dem Modul angebracht, die den Zustand der Motoranschlüsse anzeigen und für Hinweis und/oder Fehlerbeseitigung verwendet werden können. Das Controller-Board kann mit Hilfe eines Programmieradapters mit neuer Firmware programmiert werden, um noch andere Module auf das Controller-Board zu integrieren.

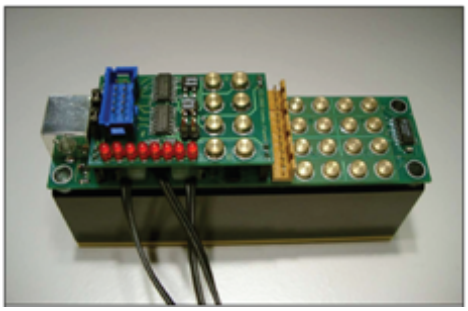


Abbildung 5.2.: Lepomux v3.0 mit Sensor- und Motorboard, [DCGL'04]

Durch diese Erweiterung des RCX stehen nun mehr Ein- und Ausgänge zur Verfügung und er kann mit weiteren sinnvollen Sensoren bestückt werden. Ein Prototyp eines Roboters, der mit diesem Lepomux ausgestattet ist, wurde von den Entwicklern des Multiplexer erstellt und getestet. Dieser Roboter besteht überwiegend aus Standard-Bauteilen des LEGO-Technik-Sortiments und wird mit je zwei Motoren auf beiden Seiten des Differentialantriebes gesteuert. An den Achsen der Antriebsräder sind zwei LEGO-Shaft-Encoder, die zur Messung der zurückgelegten Wegstrecke dienen, und vier Entfernungssensoren der Firma Sharp so angebracht, dass in jeder Richtung eine Abstandsmessung erfolgen kann. Weiterhin wurde ein Beacon-Sensor angebracht, der auf einem Servo montiert ist und zur Lokalisierung des gegnerischen Tores dient. Zusätzlich ist ein zweiter Sevo für den Schussmechanismus im Einsatz, um den Ball zu beschleunigen.

[1] Wird von seinen Erfinder LEPoMux genannt, was aus der Produktbezeichnung ***LE****GO-****Po****rt-****Mu****ltiplex****er*** entstanden ist

5.3. Kombination von LEGO-Technik und MIT-Board

Diese Art Roboter haben nichts mehr mit dem RCX zu tun, denn ihre Steuereinheit ist nun ein MIT-Board 6.270 [MIT'04] oder ein ähnliches Controller-Board der Baureihe, welche im späteren Verlauf näher beschrieben werden. Diese Boards besitzen genügend Ein- und Ausgänge und ausreichend Prozessorleistung. Weiterhin kann man durch eine Arbeit eines Diplomanden (Mirco Gerling) [MG'03] einige Controller-Boards an einem PDA anschließen und rechenintensive Programmteile auf den PDA auslagern.

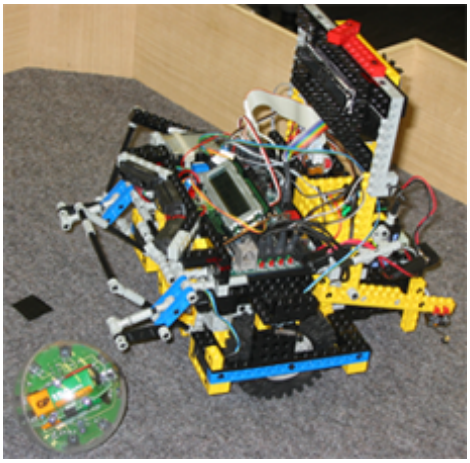


Abbildung 5.3.: Beispiel für eine Kombination von LEGO und MIT-Board 6.270

5.4. Frauenhofer-AiS-Roboter

Die Fußballroboter des Instituts Autonome intelligente Systeme der Frauenhofer - Gesellschaft [ASI'04] sind für den Einsatz in der RoboCup Middle-Size-Liga konzipiert und damit wesentlich größer als die bisher vorgestellten Roboter. Die Roboter sind als vollautonome Systeme konzipiert und besitzen eine Reihe von Sensoren, die ein vollautonomes Verhalten ermöglichen. Darunter ist eine um 360° schwenkbare Kamera, die ihre Umgebung nach Objekten durchsucht und auf diese Weise den Ball sowie eigene und generische Roboter lokalisiert. Die Navigation des Roboters wird zusätzlich durch einen Kreisel-Sensor (Gyroskop) unterstützt, der Daten über die Drehung des Roboters liefert. Die Vermeidung von Zusammenstößen mit anderen Robotern wird durch Entfernungs-Sensoren auf Infrarot- sowie Sonar-Basis und Bumper realisiert. Der Roboter verfügt zudem über einen eigenen pneumatischen Schussmechanismus. Die Datenverarbeitung übernimmt ein Notebook, das auf den Roboter montiert wird. Dies ermöglicht auch die Kommunikation zwischen den Robotern mit Hilfe von wireless LAN-Schnittstellen.

Abbildung 5.4.: Fußballroboter der Frauenhofer Gesellschaft, [ASI'04]

5.5. Computer Science Freiburg - Roboter

Bei diesem Roboter [CS-F'04] handelt es sich ursprünglich um Pioneer-I-Roboter, wie sie von der Firma ActivMedia Robotics hergestellt werden. Allerdings wurde die Hardware deutlich erweitert und das Controller Board durch ein Pioneer II Board ersetzt. Außerdem wurde eine Schussvorrichtung entwickelt, die es nicht nur erlaubt, über das ganze Feld zu schießen, sondern auch ein Führen des Balls möglich macht. Dazu sind an den Seiten Flipper angebracht, die bei Bedarf heruntergeklappt werden. Der Pioneer-Roboter besitzt einen Differentialantrieb, welcher mit Sensoren unterstützt wird. Dabei werden die Drehbewegungen der beiden Antriebsräder gemessen, wodurch Rückschlüsse auf die Positionsänderungen des Roboters möglich sind. Allerdings sammeln sich dennoch mit der Zeit Fehler an, so dass diese Art Positionsbestimmung durch andere Sensoren wie Laser-Scanner und Kamera unterstützt werden müssen.

Abbildung 5.5.: Fußballroboter der CS Freiburg, [CS-F'04]

Der Laser-Scanner vom Typ SICK LMS200 gehört zur Klasse der sogenannten Time-Of-Flight-Sensoren. Das Gerät sendet Lichtstrahlen aus, die von den Objekten in der Umgebung reflektiert werden. Aus der Zeit, die das Licht benötigt, um zum Sensor zurückzukehren, lassen sich die Entfernungen zu den Objekten berechnen. Diese liefern Entfernungsmessungen für einen Bereich von $180°$ vor dem Roboter mit einer Auflösung von $0,5°$ und einer Genauigkeit von 1 cm. Mit dem Laser-Scanner bestimmt der Roboter seine eigene Position auf dem Feld und misst die Position anderer Roboter. Die Kamera ist ein unverzichtbarer Sensor für alle Fußball-Roboter. Beim CS Freiburg werden Digitalkameras des Modells DFW-V500 von Sony verwendet. Sie dienen dazu, den Ball anhand seiner Farbe zu erkennen. Eine weitere Hardwarekomponente der Roboter sind die Notebooks, auf denen ein Linux-Betriebssystem läuft. Die Kamera ist über eine Firewire-Schnittstelle direkt mit dem Computer verbunden. Über USB können die Motoren und der Laser-Scanner angesteuert werden und eine WLan-Karten unterstützt die Kommunikation unter den Mitspielern und mit einem Server außerhalb des Spielfeldes. Weiterhin gibt es Robotertypen dieser Größenordnung, die ihre Umgebung mit einem omni-direktionalen Spiegel[2] beobachten. Die Kombination der Sensorik ermöglicht dem Roboter eine eigenständige Navigation mit Hilfe der Bildverarbeitung, dem Gyroskop und den Radantrieben. Die Erkennung des Balles und anderer Roboter wird ebenfalls mit Hilfe der Kamera durchgeführt. Die notwendige Kommunikation zwischen den Robotern und evtl. vorhandenen Server ist bidirektional mit Hilfe eines drahtlosen Netzwerkes möglich. Der Frauenhofer Roboter und der Roboter von der CS-Freiburg stellen ein gelungenes Beispiel für vollautonome Fußballroboter dar.

[2] Hierbei handelt es sich um eine kleine, nach oben gerichtete Kamera, welche über einen Spiegel eine komplette Rundumsicht des Spielfeldes einfängt

6. Ziel der Neuentwicklung

Um einen Roboter für ein Roboterfußballspiel tauglich zu machen, stehen spezielle Ziele bei der Neuentwicklung für die Konstruktion kooperativer autonomer mobiler Roboter im Vordergrund. Da Untersuchungen im Roboterfußball generalisierbar sein sollten und dadurch auch in anderen Anwendungsgebieten verwendbar sind, sollten die Ziele in einzelne Bereiche unterteilt werden. Die Entwicklung zielt grundsätzlich auf den Entwurf einer Forschungsplattform für autonome mobile Roboter und kooperative Multiagentensysteme am Beispiel des Roboterfußballs ab und nicht primär auf die Erstellung eines möglichst guten Fußballroboters. In diesem Kapitel werden Ziele für die einzelnen Bereiche (Navigation, Kommunikation etc.) vorgestellt, in dem Fähigkeiten der bestehenden Robotersysteme der RoboCup-Junior-Soccer-League [RCJ'04] gesammelt und diskutiert werden. Da die Realisierung eines kompletten Roboterfußballsystems aus zeitlichen Gründen weit über die Aufgabenstellung dieser Arbeit hinausgeht, können und sollen diese Anforderungen nur teilweise als Ziele dieser Arbeit gelten. Daher werden nur ausgewählte Fähigkeiten der bestehenden Robotersysteme genauer betrachtet.

6.1. Navigation

Ein mobiler autonomer Roboter muss in der Lage sein, innerhalb seiner Arbeitsumgebung eigenständig zu navigieren. Üblicherweise besitzt ein solcher Roboter ein Weltmodell seiner Umgebung und kann seine Position und Ausrichtung innerhalb dieses Modells mit einer gewissen Genauigkeit bestimmen. Dies ermöglicht ihm eine kontrollierte Bewegung, so dass z.B. ein Anfahren bestimmter Zielpunkte möglich ist.

Positionsbestimmung

Der Roboter benötigt ein gutes Steuerungssystem, um seine Position bestimmen zu können. Dabei darf bei der Positionsbestimmung die Diverenz zwischen Ist- und Soll-Position einen gewissen Wert nicht überschreiten. Die Messung von Position und Ausrichtung muss ausreichend schnell erfolgen, um bei der gewünschten Maximalgeschwindigkeit eine kontrollierte Bewegung zu ermöglichen. Teilautonome Roboter erfüllen diese Anforderung bei der Positionsbestimmung in gewisser Weise nicht und sind bei Steuerung oder Sensorik eingeschränkt und daher auf Unterstützung von außen angewiesen.

Bewegung

Der Roboter sollte sich mit vergleichbarer oder schnellerer Geschwindigkeit als die gegnerischen Roboter über das Spielfeld bewegen können. Jedoch sollte die Geschwindigkeit nicht zu hoch sein, da der Roboter sonst schlechter kontrolliert werden kann. In Grenzbereichen setzen Schlupfeffekte ein, welche durch die begrenzte Haftreibung der Räder entstehen. Je nach Boden- und Radbeschaffenheit wird die maximale Beschleunigung und der minimale Kurvenradius begrenzt. Das Gewicht des Roboters hat einen großen Einfluss auf die Haftreibung der Räder, denn bei steigendem Gewicht ergibt sich eine größere Haftreibung, die größere Beschleunigungen ohne Haftungsverlust ermöglicht. Der Roboter darf aber auch nicht zu schwer werden, da der Roboter dann einen größeren Energiebedarf hat.

Drehung

Der Roboter muß schnell in der Lage sein, seine Fahrtrichtung zu ändern, damit auf veränderte Spielsituationen sofort reagiert werden kann. Deshalb ist eine schnelle und genaue Lenkbarkeit eine essentielle Voraussetzung. Roboter, die bezogen auf ihre Lenkachse unsymmetrisch aufgebaut sind und somit Vorwärtsfahrt und Rückwärtsfahrt unterscheiden, müssen Drehungen bis 180° bewerkstelligen. Das sind Robotersysteme, die meistens mit einer Differentialsteuerung ausgestattet sind. Die jenigen Roboter, die in dieser Hinsicht symmetrisch aufgebaut sind, können diese Drehungen auf maximal 90° beschränken, indem sie zusätzlich ihre Fahrtrichtung umkehren.

Ballführung

Ein Roboter, der fußballspielen soll, sollte in der Lage sein, den Ball auf seinem Weg zu beeinflussen. Es ist nicht ausreichend, wenn der Roboter versucht, aus einem bestimmten Winkel auf den Ball zu zu fahren und ihn durch die Berührung in die gewünschte Richtung zu lenken. Einerseits ist es unter Umständen nicht möglich, aus allen Richtungen den Ball anzufahren, so dass die gewünschte Zielrichtung nicht erreicht werden kann. Dies trifft meistens dann zu, wenn der Ball an der Bande liegt. Andererseits erfordern veränderte Spielsituationen eine längerfristige Kontrolle des Balles. Um die Fähigkeit des Ballführens zu ermöglichen, verfügen viele Fußballrobotertypen über Vorrichtungen wie Schaufeln oder Einkerbungen.

Schuss

Der Roboter muss neben der Ballführung auch in der Lage sein, den Ball stärker zu beschleunigen und somit einen Schuss auszuführen. Dies kann entweder durch schnelles Auftreffen des Roboters auf den Ball oder durch eine separate Schussvorrichtung erreicht werden. Schussvorrichtungen des Roboters können den Anlauf ersetzen, der normalerweise verwendet wird, um Geschwindigkeit aufzunehmen und somit kann der Roboter direkt aus der Ballführung schießen. Die Schussvorrichtungen bestehen in der RoboCup-League meistens aus pneumatischen Konstruktionen oder rotierenden Flächen / Walzen, die im Kontaktfall einen Teil ihrer kinetischen Energie dem Ball übertragen. Der Nachteil derartiger Vorrichtungen ist der zusätzliche Platz- und Energiebedarf.

6.1.1. Diskussion der bisherigen Umsetzungen

In dieser Diskussion werden alle Arten der bisher konstruierten Robotertypen für die RoboCup-Junior-Soccer-League in Betracht gezogen, wobei speziell auf die Robotertypen eingegangen wird, die mit den Roboterbaukästen der LEGO MINDSTORMS-Familie [LEGO] konstruiert werden und in dieser League hauptsächlich Verwendung finden. Weiterhin werden auch die Robotertypen vorgestellt, die zu den üblichen LEGO MINDSTORMS noch die technischen Erweiterungen beinhalten, wie zum Beispiel den Lepomux für den RCX oder die LEGO-Konstruktionen mit dem MIT-Board 6.270 [MIT'04].
Wenn man die Navigation der Robotertypen in Betracht zieht, fällt auf, dass fast alle Roboterkonstruktionen etwas gemeinsam haben: Sie beruhen auf dem Prinzip der Differentialsteuerung (siehe Kapitel 3.5.1). Mit dieser Art Lenkung hat der Roboter, durch eine geschickte Ansteuerung seiner Räder, die Möglichkeit, sich in jede Richtung zu drehen und das gewünschte Ziel anzusteuern.
Bei der Positionsbestimmung der Roboter gibt es zwischen den verschiedenen Robotertypen deutliche Unterschiede. Die Roboterbaukästen haben, wie schon beschrieben, einen großen Nachteil durch die geringen Anschlussmöglichkeiten für die Sensorik. Deshalb sieht man in den offiziellen Roboterwettkämpfen oft Robotertypen, die nur einen Lichtsensor für die Bodenbeschaffenheit enthalten. Die Roboter bekommen ihre Positionsbestimmung über die Graustufenverteilung des Bodenbelags, welche vom eigenen Tor zum gegnerischen Tor verläuft. Die Robotertypen, die einen Multiplexer oder das MIT-Board 6.270 benutzten, haben zusätzlich zu dem Bodensensor auch noch Umgebungssensoren. Damit kann die nähere Umgebung erfasst werden, um Wände oder Hindernisse (Gegen- oder Mitspieler) zu umgehen. Bei diesen Robotertypen sind häufig 4 Infrarot Sharp Sensoren (siehe Kapitel 7.5.1.2) zu finden, wobei üblicherweise 3 Sensoren nach vorne ausgerichtet sind und 1 Sensor nach hinten zeigt. Die vorderen Sensoren sind überwiegend so angebracht, dass die beiden äußeren Sensoren in einem ca. 45° Winkel zur Seite ausgerichtet sind und der mittlere Sensor direkt nach vorne zeigt. Diese Anordnung hat den Vorteil, dass der Roboter mit den schräg angebrachten Sensoren die Möglichkeit erhält, einen viel größeren Bereich seiner Umgebung zu kontrollieren und bei einem "wall-follow" eine bessere Kontrolle besitzt.

“wall-follow“ heißt, dass der Roboter mit einem im Programm festgelegten Abstand an einer Wand entlang fährt und mit seinem Antriebsrädern diesen Abstand regulieren kann. Mit diesen Umgebungssensoren haben die Roboter auch die Möglichkeit, festzustellen, auf welcher Seite des Spielfeldes sie sich befinden. Auf Grund der Boden- und der Umgebungssensoren kann der Roboter seine Position im Spielfelde ermitteln. Falls die Tore mit “Beacons“ markiert sind, haben die Roboter die Möglichkeit, festzustellen, in welche Richtung sie fahren müssen. Bei einem“Beacon“ handelt es sich um Infrarot-Leutfeuer (siehe Kapitel 7.3).
Die Robotertypen, die das MIT-Board 6.270 installiert haben, besitzen zusätzlich die Möglichkeit, einen PDA anzuschließen, der über eine globale Kamera die “exakte“ Position des Roboters übermitteln kann. Somit könnte man die bestehenden Sensoren für die Kontrolle der ermittelten Position einsetzten. Dieses Verfahren der “exakten“ Positionsbestimmung über eine globale Kamera kann auch mit einem RCX ausgewertet werden, welches in früheren Projekten der HAW Hamburg [Luck’04] zum Einsatz kam.
Wenn man nun die Bewegungen der Roboter betrachtet, fällt auch hier auf, dass es zwischen den einfachen Roboterbaukästen der LEGO MINDSTORMS-Familie und den Erweiterungen große Unterschiede gibt. Der einfache RCX besitzt nur zwei Anschlussmöglichkeiten für die Antriebsmotoren, wobei die von LEGO MINDSTORMS mitgelieferten Motoren sich bei nähere Betrachtung nicht all zu leistungsfähig erweisen. Bei längerer Benutzung verschleißen die Motoren unterschiedlich stark (Problem: Kurvenfahrt). Viele Robotertypen werden mit einer Untersetzung[1] ausgestattet, um die Geschwindigkeit zu erhöhen. Jedoch sollte die Untersetzung nicht zu klein gewählt werden, da der Roboter bei einer zu hohen Geschwindigkeit nicht mehr kontrollierbar ist. Viele Roboter werden bei der Verwendung von nur zwei Motoren auch recht klein gehalten, damit die Motoren es schaffen, das Gewicht des Roboters zu tragen, ohne ein Durchdrehen der Räder zu verursachen. Bei dem Einsatz des Lepomux kann man Konstruktionen beobachten, wo für jede Antriebsseite zwei Motoren zur Verfügung stehen, um mehr Leistung auf die Räder zu übertragen. Diese Konstruktionen mit vier LEGO MINDSTORMS Motoren sieht man auch, wenn ein MIT-Board 6.270 benutzt wird. Es ist aber auch mit diesem Board möglich, leistungsstärkere Motoren des Modellbaues einzusetzen. Es muss noch berücksichtig werden, dass, je mehr Motoren eingesetzt werden, desto höher auch der Energiebedarf des Roboters ist, sowie das Gewicht der Roboter durch zusätzliche Motoreninstallation erhöht wird und durch mangelnden Boden- und Radbeschaffenheit zu Schlupfeffekten führen kann. Ein gutes Mittelmass spielt für die Bewegungen der Roboter eine entscheidende Rolle.
Der Bereich Drehung in der RoboCup-Junior-Soccer-League ist recht einfach gestrickt, denn da fast alle Robotertypen mit der Differenzialsteuerung arbeiten, sind all diese Roboter unsymmetrisch zur Lenkachse ausgerichtet und so gezwungen, eine Drehung bis $180°$ “in Kauf“ zu nehmen. Diese Konstruktion der Differenzialsteuerung verlängert den zu fahrenden Weg des Roboters, um gezielt hinter dem Ball zu kommen erträglich. Dabei kann es vorkommen, dass der Roboter kurze Zeit die Kontrolle des Balles verliert, da der Ball evtl. in Bewegung ist und ihn nicht an der vermuteten Position wieder findet.

[1] Eine Untersetzung besteht hauptsächlich aus einem großen Zahnrad, welches am Motor angeschlossen wird und einem im Durchmesser kleineren Zahnrad, welches an der Achse des Rades montiert ist

In der RoboCup-Junior-Soccer-League ist es üblich, dem Roboter eine Ballführung zu ermöglichen, die über eine Vorrichtung wie Schaufel oder Einkerbungen zu erreichen ist. Jedoch sind diese Robotertypen nicht so feinfühlig, um den Ball am Roboter zu fixieren, denn bei Kurvenbewegungen neigt der Ball aus der Führung zu gleiten. Wenn man sich nicht ganz an die Regeln der RoboCup-Junior-Soccer-League hält, sieht man auch Robotertypen, die eine Fangvorrichtung für den Ball installiert haben, um dem Ball bei der Drehung des Roboters unter Kontrolle zu halten. Bei vielen Robotertypen in der RoboCup-Junior-Soccer-League ist eine Schussvorrichtung aus bautechnischen Gründen selten zu sehen, da die Roboter einen zusätzlichen Motor bzw. Servo benötigen und diese Vorrichtung zu viel Platz einnimmt. Deshalb wird meistens der Ball entweder durch schnelles Auftreffen oder durch schnelles Drehen des Roboters und abruptes Stoppen beschleunigt, um somit dem Ball in die entsprechende Richtung zu schießen.

Fazit

Die Nachteile der Roboterbaukästen von LEGO MINDSTORMS liegen klar auf der Hand, denn das größte Manko des RCX ist, dass er nicht die nötige Anzahl von Sensoren und Motoren Ein- / Ausgänge hat, um eine “exakte“ Positionsbestimmung und Navigation zu einem gewählten Ziel zu gewährleisten. Die Sensoren für die Ballerkennung sind bei vielen Robotertypen starr nach vorne ausgerichtet, wobei diese Art Sensorik bei einer falschen Installation ungenaue Informationen über die Ballposition übermitteln. Durch die Differenzialsteuerung entsteht auch ein großer Nachteil, da der Roboter längere Wege zurück legen muss, um gezielt den Ball oder die gewünschte Position anzusteuern. Dabei tritt es häufig auf, dass der Roboter den Ball nicht mehr an der gewünschten Position vorfindet, weil er durch eine ungenaue Drehung des Roboters die Kontrolle über den Ball verliert. Diese ungenauen Drehungen treten vermehrt auf, wenn die LEGO MINDSTORMS Motoren länger im Betrieb waren, denn sie neigen dazu, durch Verschleiß ungenauer zu werden, so dass der Roboter bei einer geraden Fahrt eher zu einer Bogenfahrt neigt. Roboter nur mit dem Bodensensoren ausgestattet, sind oftmals nicht in der Lage, zu entscheiden, in welcher Richtung sich das gegnerische Tor befindet und schießen verstärkt auch Eigentore. Einige Probleme konnten durch die Erweitung mit dem Lepomux für den RCX oder die Kombination zwischen MIT-Board 6.270 und LEGO-Technik von vorne rein vermieden werden, dennoch treten auch hier Probleme auf. Die mangelnde Verfügbarkeit von Umgebungssensoren kann man mit diesen Erweiterungen zwar beheben, jedoch sind die Sensoren in den meisten Fällen starr vorne am Roboter angebracht und können nur ein Teil ihrer näheren Umgebung kontrollieren. Viele der Roboter, die mit den Erweiterungen ausgestattet sind, haben ein recht “dürftiges“ Bumper-System, welches dem Roboter bei Kollisionen mit der Wand oder Hindernisse vor weiteren Schäden warnen kann. Es ist ein Vorteil, dass die Robotertypen mit einer LEGO-Plattform recht schnell und kompakt aufzubauen sind. Dennoch sind viele Robotertypen recht labil, so das es bei Kollisionen zu Schäden am Roboter selbst kommen kann.

6.1.2. Ziele der Neuentwicklung für die Navigation

Es ergeben sich aus der bisherigen Diskussion folgende Ziele für die Neuentwicklung:

- Die Differenzialsteuerung erwies sich bis jetzt immer als das ausgereifteste System bei einer Roboterplattform, jedoch ist diese Art Antriebssystem recht unflexibel und muss mit einer neueren Idee einer Antriebsplattform verbessert werden. Dabei sollte untersucht werden, ob ein Antriebsystem mit “omnidirectional wheels“ verwirklicht werden kann. “Omnidirectional wheels“ sind Räder, die sich in alle Richtung bewegen können und hier in Deutschland auch als “Allseitenrollen“ bekannt sind.

- Um die Navigation und Positionsbestimmung zu verbessern, sollte der Roboter die Möglichkeit haben, durch eine Rundumsicht die Kontrolle seiner Umgebung zu gewährleisten. Dafür dürfen die Umgebungssensoren nicht starr vorne am Roboter befestigt sein, sondern müssen auch in der Lage sein, in eine andere Richtung schauen zu können.

- Dieses Prinzip gilt auch für den Ballsensor, denn der Roboter sollte es vermeiden, die Kontrolle über den Ball zu verlieren.

- Um den Roboter stabiler zu gestalten, darf die Plattform nicht mehr aus LEGO-Technik bestehen. Weiterhin sollte die Plattform so konstruiert sein, dass sie leicht umzubauen und bei irgendwelchen Schäden schnell auszutauschen ist.

- Eine Ballführung und eine Schussvorrichtung sollte in den Roboter mit integriert sein.

- Da der Roboter laut den Regeln der RoboCup-Junior-Soccer-League nicht größer als 22 cm im Durchmesser und eine Höhe von 22 cm nicht überschreiten darf, muss der Roboter sehr kompakt konstruiert werden.

- Mit den vielen Sensoren und Aktoren benötigt der Roboter deutlich mehr Energie als die Roboter, die nur mit dem Roboterbaukasten von LEGO MINDSTORMS konstruiert wurden. Deshalb sollte darauf geachtet werden, dass die Plattform des Roboters so leicht wie möglich ausfällt und die eine energiefreundliche Programmierung beinhaltet.

6.2. Kommunikation

In diesem Abschnitt soll untersucht werden, welche Kommunikationsanforderungen sich bei der Arbeit mit autonomen Multiagentensystemen und im speziellen bei Fußballrobotern der RoboCup-Junior-Soccer-League ergeben. In einem ersten Schritt werden allgemeine Anforderungen an die Kommunikation zwischen autonomen Multiagentensystemen gesammelt. Für eine kooperative Arbeit ist ein Kommunikation notwendig, um Absichten mitzuteilen und zu koordinieren. Hier spricht man auch von einer Synchronisation der Mutiagentensysteme. Eine weitere Anforderung besteht darin, dass die Roboter die Rollen und Aufgabenverteilung absprechen und Handlungsergebnisse oder sonstige Erkenntnisse dem Partner mitteilen. Hierbei kann die Kommunikation im Roboterfußball zum Beispiel den Austausch der Ball- und Roboterpositionen oder die Abstimmung der Rollenverteilung auf einzelne Roboter (Torwart, Stürmer) bedeuten. Es muss darauf geachtet werden, dass jedes Agentensystem seine zugeteilten Aufgaben abarbeiten und im Fehlerfall durch jedes der anderen gleichartigen Systeme ersetzt werden kann. Deshalb muss jedes System in der Lage sein, eine bidirektionale Kommunikation mit den anderen System zu gewährleisten. Ansonsten wäre eine Neuverteilung der Aufgaben unmöglich.

6.2.1. Diskussion der bisherigen Umsetzung

Bisher gibt es in der RoboCup-Junior-Soccer-League noch keinerlei Umsetzung einer Kommunikation zweier Roboter im Team. Es wurde in einem Semesterprojekt mal versucht die Roboter der LEGO MINDSTORMS-Familie mit einer Kamera zu steuern, jedoch beschränkte sich die Kommunikation nur zwischen der Kamerasoftware und dem RCX. Eine effiziente Kommunikation zweier Roboter wurde weder auf einem RCX noch auf dem MIT-Board 6.270 realisiert.

6.2.2. Ziele der Neuentwicklung für die Kommunikation

Um eine Kommunikation zweier Roboter dennoch zu verwirklichen, wäre eine mögliche Lösung, einen PDA mit dem Controllerboard zu koppeln und die Kommunikation über eine wireless LAN- oder eine Bluetooth-Schnittstelle zu verwirklichen. Dabei sollte man darauf achten, dass die Schnittstellen genügend schnell sind, damit es zu keiner Fehlinformationen bei den Angaben von beispielsweise eigenen, gegnerischen und/oder Ball-Position sowie Aufgabenverteilung (Torwart, Stürmer) kommt.

6.3. Rechenleistung

Autonome mobile Roboter müssen viele verschiedene Berechnungen für z.B. Positionsanalyse ausführen, damit sie ihre Aufgaben erfüllen können. Solche Aufgaben können Prioritäts- und Zeitvorgaben bezüglich ihrer Ausführungsdauer und Häufigkeit beinhalten. Des Weiteren sollte der Roboter eine Fehlerbehandlung haben, um die Funktionsfähigkeit des Roboters auch im Fehlerfall einzelner Module zu gewährleisten. All diese Aufgaben benötigen viel Rechenleistung und deshalb sollte das Ziel bei autonomen mobilen Robotern sein, eine Bearbeitung von notwendigen und sinnvollen Aufgaben zu gewährleisten und Mechanismen für die koordinierte Bearbeitung dieser Aufgaben zu finden.

6.4. Systemspezifische Ziele

Im Folgenden werden weitere detaillierte Ziele gesammelt, die für den zu entwickelnden Fußballroboter spezifisch sind und nicht den bisher angeführten Kategorien zugehören:

6.4.1. Regelkonformität

Wesentliche Eigenschaften des Roboters werden durch das Reglement der RoboCup-Junior-Soccer-League vorgegeben, wobei der Durchmesser und die Höhe nicht über 22 cm hinausgehen darf. Wenn man sich entschließt, eine globale Kamera zum Einsatz zu bringen, darf der Ball nur bis zu 30 % vom Roboter verdeckt werden, da der Ball ansonsten nicht mehr von der globalen Bildverarbeitung zu sehen wäre.

7. Hardware-Auswahl

Es ist ein komplexer Prozess, eine Neuentwicklung eines Fußballroboters zu entwickeln und die damit verbundene Auswahl der zu verwendenden Bauteile - zum Beispiel Controllerboard, Sensoren und Aktoren - zu bestimmen. Dies liegt unter anderem in den vielfältigen Anforderungen an das Gesamtsystem, denn hier spielen die verschiedenen Fähigkeiten und Anforderungen der einzelnen Bauteile, sowie Platz- und Stromversorgungsvorgaben der Elektronik eine große Rolle. Weiterhin können die Designentscheidungen für oder gegen ein bestimmtes Bauteil nicht vom Rest des Designs unabhängig betrachtet werden.
In diesen Kapitel wird die Methodik des Vorgehens während der Entwurfsphase erläutert. Anschließend werden Designentscheidungen, die daraus resultierenden Bauteile und ihre Einbindung in das Gesamtsystem im Detail diskutiert.

7.1. Methodik

Die Entwicklung des Fußballroboters läuft in dieser Diplomarbeit in folgenden Schritten ab. Dabei werden die Projektmodelle aus dem im Kapitel 3.8 mit berücksichtigt:

- In der “Beurteilungsphase“, auch “Evaluierungsphase“ genannt werden Anforderungen an die zu realisierenden Komponenten unter Berücksichtigung der gewünschten Ziele und bisheriger Erfahrungen erarbeitet. Die Rahmenbedingungen und die Möglichkeiten werden zusammengestellt und es wird eine Recherche nach möglichen Bauteilen, Sensor- und Aktorkomponenten durchgeführt.

- In der “Planungsphase“ wird die Verfügbarkeit der gewünschten Bauteile ermittelt. Wenn es jedoch zu keiner Bezugsmöglichkeit der gewünschten Bauelemente kommt, muss in einer wiederholten “Evaluierungsphase“ nach Alternativen gesucht werden.

- Anschließend sollte eine “Konstruktionszeichnung“ des zu entwickelnden Roboters erstellt werden, welches aus Ideen von Handzeichnungen und anschließend mit einem CAD-Programm verwirklicht wird.

- Anhand der erstellten Konstruktionszeichnungen wird im nächsten Schritt ein “Prototyp“ des Roboters entwickelt.

- Nach einer Testphase der zu verwendeten elektronischen Bauteile werden sie in die Konstruktion integriert. Diese Phase nennt man auch “prototypische Implementierung“, welches auch den Test der einzelnen Komponenten des Roboters hinsichtlich ihrer grundlegenden Funktionsfähigkeit ermöglicht. Dies gilt besonders für die Hardwareansteuerung der einzelnen Sensoren und Aktoren.

Auf Grund dieses Vorgehens wird eine frühe Fehlererkennung sichergestellt und ein schrittweiser kontrollierter Aufbau des Roboters ermöglicht. Die abschließende Programmierung des Roboters ermöglicht die Demonstration von ausgesuchten Fähigkeiten im Detail.
Trotz guter Planungen können innerhalb der einzelnen Phasen Probleme auftreten, die in früheren Phasen nur schlecht oder gar nicht zu erkennen waren. In diesem Fall kann ein "Re-Design" notwendig werden, um bestehende Probleme zu beheben. Zusätzlich kann es, auf Grund von Erfahrungen während der Entwicklung, sinnvoll sein, Verbesserungsvorschläge zu verarbeiten und den Roboteraufbau auf diese Weise über mehrere Versionen zu optimieren.

7.2. Designentscheidungen

Um die im vorherigen Kapitel erarbeiteten Ziele erreichen zu können, sind eine Reihe von Designentscheidungen bezüglich der Hardware-Realisierung zu treffen. Unter einer Designentscheidung ist beispielsweise die Auswahl eines Controllerboards aus den momentan auf dem Markt befindlichen Modellen zu verstehen. Diese Entscheidungen sind von Bedeutung, da sie Einfluss auf die späteren Fähigkeiten des Systems und insbesondere auf die Erfüllbarkeit der in vorherigen Kapiteln gesammelten Anforderungen an ein Roboterfußballsystem haben. Entscheidungen für die eine oder andere Komponente begründen sich unter anderem durch die Abwägung des Kosten/Nutzenverhältnisses und beinhalten zum Beispiel den Platzbedarf, den Stromverbrauch, die Kosten sowie die Verfügbarkeit der gewünschten Komponenten. Im folgenden Abschnitt wird eine Auswahl verfügbarer und geeigneter Komponenten wie Controllerboard, Sensoren und Aktoren diskutiert, jedoch wird vorher die Spielfläche und die dazugehörige Laborumgebung vorgestellt.

7.3. Laborumgebung

Das Labor, welches sich im 11. Stock der HAW Hamburg im Raum 11.61 "E.W.Dijkstra" [1] befindet, beinhaltet eine große Auswahl an Material und Hilfsmitteln für die Entwicklung eines Roboters. In diesem Raum sind unter anderem Rechner für die Programmierung der Roboter zu finden. Des weiteren ist eine Spielfläche zu finden, welche mit den Maßen für eine Spielfläche der RoboCup-Junior-Soccer-League [RCJ'04] für Matches 2 gegen 2 Roboter von der Zentralen Werkstatt der HAW Hamburg [Werkstatt] konstruiert wurde. Jedoch wurden die Regeln der RoboCup-Junior-Soccer-League [RC-JS'04] etwas erweitert und verändert, indem die Tore mit je einem IR-Beacon ausgestattet sind. IR-Beacons sind Infrarot-Leuchtfeuer, die ursprünglich zur Orientierung bzw. Markierung von Punkten innerhalb eines Parcours verwendet wurden. Anhand dieser Markierungen können die Roboter ein Ziel leicht auffinden und ansteuern. Die hier verwendeten IR-Beacons (siehe Abbildung 7.1) wurden von Prof. Dr. Klemke [Klemke'04] entwickelt und sind bereits seit geraumer Zeit in Projekten der HAW

[1] Edsger Wybe Dijkstra, niederländischer Informatiker, geboren 11. Mai 1930 in Rotterdam, gestorben am 6. August 2002, in Neunen (Niederlande).

Hamburg [Luck'04] zum Einsatz gekommen. Sie wurden speziell für die Verwendung mit Robotern konzipiert, die auf dem MIT-Board 6.270 [MIT'04] basieren. Um die Tore zu unterscheiden senden diese IR-Beacon Pulse mit Frequenzen von 100Hz bzw. 125Hz aus.Um das Signal unempfindlich gegenüber Umgebungslicht zu machen, wird eine Trägerfrequenz von 40 kHz moduliert.

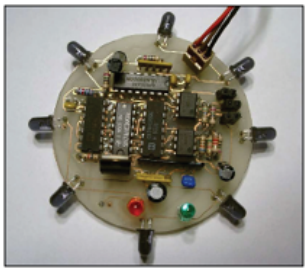


Abbildung 7.1.: IR-Beacon für die Tor Erkennung

Eine weitere Änderung wurde im Bereich des Spielflächenbelags vorgenommen. Der Boden des Spielfelds ist nicht in einer Grauabstufung bedruckt, wie in den Regeln der RoboCup Junior Soccer League vorgesehen ist, sondern wurde mit grauem Fils ausgelegt, um die zufälligen Bewegungsabläufe des Spielballes zu verhindern und dadurch eine bessere Kontrolle über diesen zu haben. Bei dem Ball (Roboball MK2 der Firma Wiltronics) [WRPL'03] (siehe Abbildung 7.2), der in der HAW Hamburg eingesetzt wird und den Regeln der RoboCup-Junior-Soccer-Legaue entspricht, handelt es sich um einen Kunststoffball, der 105 g schwer ist und einen Durchmesser von ca. 8 cm aufweist. In seinem Inneren befinden sich Leuchtdioden (LEDs), die durch eine Batterie gespeist werden und durch die der Ball von den Spielern wahrgenommen werden kann.

Abbildung 7.2.: Roboball MK2 der Firma Wiltronics, [WRPL'03]

Falls Lötarbeiten am Roboter, Sensorik oder Aktoren durchgeführt werden müssen, steht eine kleine Werkstatt im Raum 11.64 mit allem was man für solche Arbeiten braucht, zur Verfügung.

7.4. Aktoren

7.4.1. Motoren

Bei der Auswahl eines geeigneten Motors (siehe Abbildung 7.3) sind einige Informationen und Berechnungen über Drehmoment, Drehzahl, Leistung usw. zu beachten. Wie man diese Eckdaten eines benötigten Motors bekommt, wurde im Kapitel 3.4.2 erläutert.

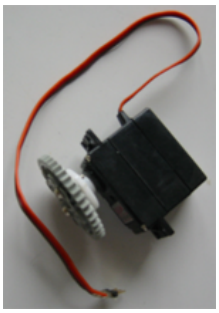


Abbildung 7.3.: Servomotoren; Umbaubar zu einem Antriebsmotor

Wenn man nun alle benötigten Informationen zusammengetragen hat, dann muss noch ein wichtiger Punkt in Betracht gezogen werden: Der Strom- und Spannungsbedarf des Motors. Die meisten Motoren arbeiten mit Spannungen von 6 bis 12 Volt, welches in dem Bereich des benötigten Wertes liegt. Die Motortreiber zum Beispiel des MIT-Board 6.270 dürfen jedoch nur mit maximal 600 mA belastet werden, ansonsten werden sie zerstört. Darum sollte man drauf achten, dass der Dauerstrom weit unterhalb dieses Wertes liegt. Kurzfristige Stromspitzen verkraften die Motortreiber des MIT-Borads, man sollte sie aber nicht ständig “am Limit“ betreiben. Bei der Wahl nach dem passenden Antriebsmotor wurde aus finanziellen Gründen auf die vorhandenen Motoren zurückgegriffen. Diese Motoren sind umgebaute Servo-Motoren (GIGA live ce STANDARD)[2]. Dieser Motor hat ein Kunststoffgetriebe, seine Motordrehzahl liegt bei ca. 40 Umdrehungen pro Minute und die Abmessungen betragen 39,5 mm x 20 mm x 30 mm. Als Servo hat dieser eine Stellgeschwindigkeit von 0,17 sec und eine Winkelgröße von $40°$.

[2] Eine gute Alternative ist der Servo S-71, da er ein hohes Drehmoment besitzt und doppelt kugelgelagert ist, sowie über ein Metallgetriebe verfügt. Für andere Alternativen siehe auch unter [Servos]

7.5. Sensoren

7.5.1. Entfernungsmessung

Wie im Kapitel 3.6.3 schon erwähnt, gibt es drei Klassen.

- Die erste Klasse besteht aus den Berührungssensoren, die auch als Bumper bezeichnet werden und häufig aus Mikroschalter[3] bestehen. Diese Schalter verursachen bei einer Berührung einen Kurzschluss bzw. eine Unterbrechung des Kurzschlusses. Sie können entweder an einem digitalen oder analogen Eingang eines Controller-Board angeschlossen werden. Jedoch erbringen diese Art Sensoren nur wenige Informationen (nur, ob der Roboter ein Hindernis kontaktiert hat). Ziel sollte es aber sein, Hindernisobjekte zu erkennen, bevor der Roboter gegen sie stößt, um im schlimmsten Fall eine Verkeilung zu verhindern. Deshalb kommen Berührungssensoren in der Regel nur als “Notausschalter“ zum Einsatz. Ihre Reichweite kann mit entsprechenden mechanischen Konstruktionen wie Bügeln etc. vergrößert werden, liegt aber im Allgemeinen im Bereich von wenigen Zentimetern um den Roboter.

- Die zweite Klasse von Abstandsmessern besteht aus Sensoren, ähnlich den Berührungssensoren, die Objekte innerhalb eines bestimmten Entfernungsbereiches erkennen können und ein Signal abgeben, wenn die Entfernung des Objekts einen bestimmten Schwellwert unterschreitet. Die Sensoren liefern also nur ein binäres Ausgangssignal: Entweder “Objekt erkannt“ oder “kein Objekt“. Ihre Reichweite kann aber 50 cm und mehr betragen.

- Die dritte Klasse liefert Informationen über die tatsächliche Entfernung eines Objektes. Sensoren dieser Kategorie kommen in den meisten Anwendungen zum Einsatz, denn sie sind flexibel einsetzbar und können mit unter die Aufgaben der Sensoren der ersten beiden Kategorien übernehmen. Je nach Sensortyp gibt es aber in der Regel einen Totbereich in unmittelbarer Nähe des Sensors, in der die Daten unbrauchbar sind. Werden Messungen in diesem Bereich benötigt, muss doch auf die erste oder zweite Kategorie zurückgegriffen werden. Die maximale Reichweite variiert beträchtlich und beträgt bei Infrarotsensoren etwa 80 cm, bei Ultraschallsensoren mehrere Meter und kann bei professionellen Laser-Sensoren mehrere 100 m betragen. In den folgenden Abschnitten werde Entfernungssensoren vorgestellt, die evtl. im Labor vorhanden oder kostengünstig zu erhalten sind.

[3] Die im Labor zur Verfügung stehenden Mikroschalter sind sogenannte Subminiatur-Schalter DB5 der Firma Cherry [Cherry]. Dieser Schalter hat eine maximale Schaltspannung von 250 V und sein maximaler Schaltstrom beträgt 1 A.

7.5.1.1. Objekterkennung auf IR-Basis - Sharp IS471

Diese Art Sensoren [SHARP] gehören in die zweite Kategorie der Entfernungssensoren, denn sie liefern ein binäres Ausgangssignal - je nachdem, ob ein Objekt erkannt wurde, oder nicht. Über die Entfernung des Hindernisses wird keine Angabe gemacht. Der IS471 sendet über eine anzuschließende IR-LED moduliertes Infrarotlicht aus. Befindet sich vor dem Sensor ein Objekt, wird das Licht in den IS471 zurück reflektiert. Durch das modulierte Licht ist der IS471 recht unempfindlich gegen Fremdeinstrahlung. Die Reichweite lässt sich über die Abstrahlleistung der IR-LED verändern und beträgt etwa 10 cm.

7.5.1.2. Entfernungsmessung auf IR-Basis - GP2D1xx

Abbildung 7.4.: Sharp Entfernungsmesser der Serie GP2D1xx

Bei diesem Entfernungssensor handelt es sich um die dritte Kategorie. Der GP2D12 [GP2D12] von der Firma Sharp (siehe Abbildung 7.4) arbeitet mit IR-Strahlung und gibt einen zur gemessenen Entfernung proportionalen Spannungspegel aus (siehe Abbildung 7.5).

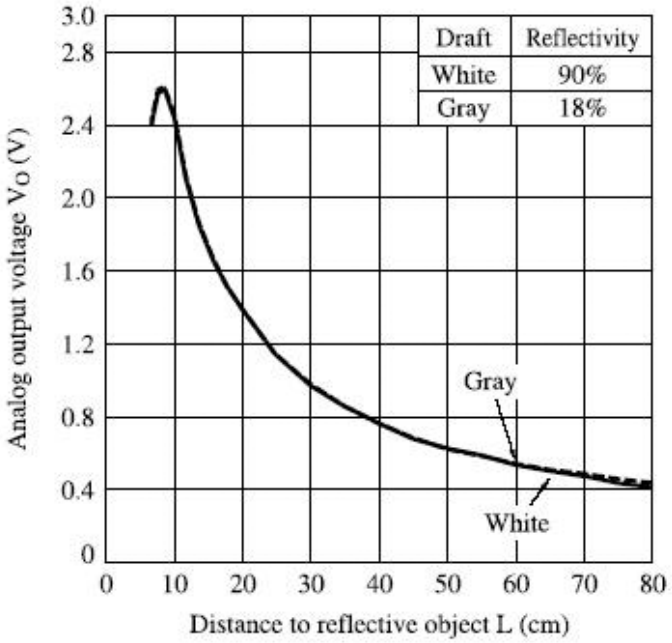


Abbildung 7.5.: Kennlinie eines GP2D12

Im Diagramm ist ein Totbereich bei Entfernungen unter 8 cm zu erkennen. Die maximal auswertbare Entfernung beträgt etwa 50 cm bis 60 cm. Der Sensor verfügt über ein Kabel mit 3 Adern (Betriebsspannung, Mass und Signal), über das er ohne weiteres mit einem Analog-Eingang des Controllers verbunden werden kann. Der Sharp-Sensor besteht im Wesentlichen aus einer IR-LED und einem sogenannten PSD-Chip (Position Sensitive Detector). Dazu kommt noch ein wenig Auswert-Elektronik. Abbildung 7.6 verdeutlicht die Arbeitsweise des Sensors. Das von der LED emittierte IR-Licht wird von Objekten in unterschiedlicher Entfernung zu unterschiedlichen Punkten auf dem PSD reflektiert.

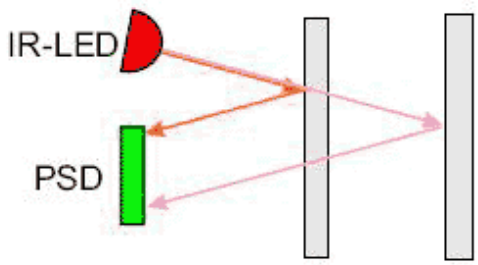


Abbildung 7.6.: Funktionsweise eines PSD

Je nach Auftreffpunkt gibt der PSD über die Elektronik ein anderes Spannungssignal aus. Der GP2D12 arbeitet zuverlässig und lässt sich durch Materialeigenschaften (Farbe, Reflexionsverhalten etc.) nur wenig beeinflussen. Da der Sensor aber mit Infrarot-Strahlung arbeitet, wird er durch andere IR-Quellen wie Sonneneinstrahlung oder starke Scheinwerfer enorm gestört. Dadurch kann es vorkommen, dass die Messungen äußerst unzuverlässig werden. Die fremden IR-Strahlen müssen nicht einmal direkt in den Sensor gestrahlt werden. Es reicht schon eine diffuse Reflexion. Der Sensor reagiert auf diese Fremdstrahlung mit einem sehr niedrigen Ausgangspegel, so als würde er kein Hindernis vor sich sehen. Es gibt noch weitere baugleiche Sensoren, die sich nur in ihrem Messbereich unterscheiden:
Der GP2D120 entspricht dem GP2D12, nur liegt der Messbereich bei etwa 4 cm bis 30 cm. Weiterhin gibt es den GP2D150, der einen Messbereich von 20 cm bis 150 cm besitzt.

7.5.1.3. Entfernungsmessung mit Ultraschall

Ultraschall-Sensoren, wie im Kapitel 3.6.5 schon erklärt wurde, stoßen einen kurzen hochfrequenter Schallimpuls aus und berechnen über die Ankunftszeit der Reflexion auf einem Empfänger die Entfernung zu einem Objekt in der Umgebung. Ultraschall-Sensoren haben eine Reihe von Vor- und Nachteilen. Vorteilhaft ist, dass sie unabhängig vom Umgebungslicht sind und so nicht wie die IR-Sensoren durch Sonneneinstrahlung etc. gestört werden können. Die gemessene Entfernung ist prinzipiell auch extrem genau, je nach verwendeter Auswertelektronik liegen die Standardabweichungen im oder unter dem Millimeterbereich.

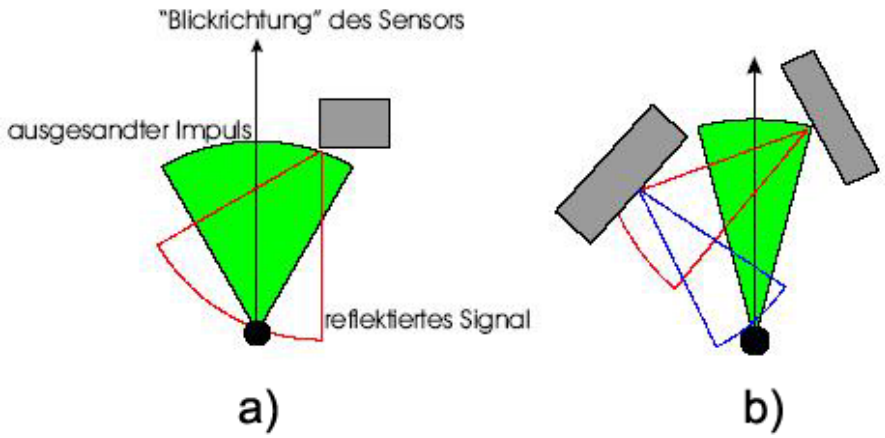


Abbildung 7.7.: Kegelförmige Reflexion bei Sonarsensoren (a) und Spiegelreflexion (b)

Nachteilig ist, dass sich der Schall nicht strahlförmig ausbreitet (wie IR-Licht) sondern kegelförmig (siehe Abbildung 7.7). Je nach Ausbreitungswinkel des Sensors werden also auch Objekte erfasst, die relativ weit außerhalb der Mittelrichtung des Schallkegels liegen. Die präzise Position eines Hindernisses ist also nicht bekannt, es kann sich überall innerhalb des Sonarkegels auf einem Kreisbogen mit Radius s befinden. Ein weiteres Problem sind sogenannte Spiegelreflexionen (siehe Abbildung 7.7). Dabei trifft der Ultraschallimpuls in einem zu flachen Winkel auf das Hindernis, das Signal wird gar nicht mehr oder erst über andere Hindernisse zum Sensor zurück geworfen. In beiden Fällen ergibt sich ein ungültiger oder falscher Messwert. Bei der Verwendung mehrerer Sonarsensoren ist auch das Problem des Übersprechens zu beachten. Dabei nimmt ein Sensor das reflektierte Signal eines anderen Sensors auf und erhält falsche Messergebnisse.
Als Beispiel für einen Sonarsensor wird der SRF04 der Firma Devantech [SRF04] betracht (siehe Abbildung 7.8). Der SRF04 erkennt Objekte in 3 cm bis zu 3 Metern Entfernung. Der Sensor besteht aus Sender und Empfänger sowie einer Auswerteelektronik .

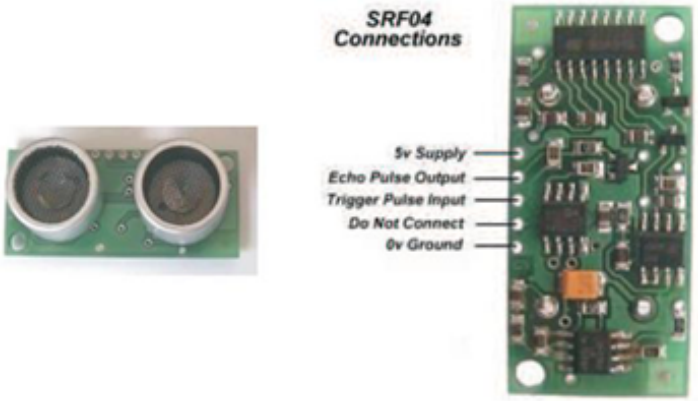


Abbildung 7.8.: Frontansicht und Anschlüsse des SRF04, [SRF04]

Neben Betriebsspannung (5V) und Masse gibt es noch 2 weitere Anschlüsse. Am Trigger-Pulse Input ist ein mindestens 10 ms langer Impuls anzulegen, um eine Messung zu starten. Daraufhin wird ein Ultraschall-Impuls abgestrahlt und der Echo-Pulse-Ausgang auf Betriebsspannung gelegt. Sobald das erste reflektierte Signal wieder beim Sensor eintrifft, wird dieser Ausgang wieder auf Masse gezogen. Die Länge dieses High-Impulses ist also direkt proportional zur gemessenen Entfernung. Der Anschluss des SRF04 an einem Controllerboard muss individuell betrachtet werden. Beim Handyboard [Handy] ist es problemlos. Der Echo-Pulse-Ausgang des Sensors wird mit einem Digital-Eingang des Handyboards verbunden. Der Trigger-Pulse-Eingang kommt an einen Digitalausgang auf dem Expansionboard[4]. Beim Vorgängermodell (MIT-Board 6.270) muss eine zusätzliche Platine auf dem Controllerboard integriert sein, damit die gleichen Ein- und Ausgänge zur Verfügung stehen.

7.5.1.4. Ball-Sensoren

Um den Spielball (Roboball MK2) [WRPL'03] zu erkennen, benötigt man Sensoren, die speziell für diesen Ball entwickelt wurden. Bill Corcoran, Josh Bennion, Geodie MacDonald und Andrew Franzosen konstruierten den Bellarine-RoboBall-Sensor und brachten ihn in der RoboCup-Junior-Soccer-WM 2000 zum Einsatz. Der Sensor ist in zwei geätzt Leiterplatinen aufgeteilt. Auf der einen Platine sind 4 QSD723 Fototransistoren angebracht, die mit zwei Kabeln an die zweite Platine angeschlossen sind, die für die Auswertung der Messinformationen zuständig ist und diese an ein Controllerboard weiterleitet. Für weitere Informationen wird hier auf das im Anhang beigefügten Beiblatt der Sensoren hingewiesen.

7.5.2. Reflexkoppler

Wie im Kapitel 3.6.1 schon erwähnt, ermöglichen sie die Unterscheidung von verschiedenfarbigen Oberflächen. Abbildung 7.9 verdeutlicht, dass ein Reflexkoppler aus einer Fotodiode und einem Fototransistor besteht.

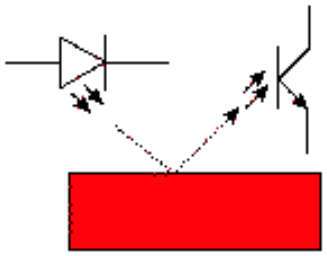


Abbildung 7.9.: Funktionsweise eines Reflexkopplers

[4] Um den Sonar mit dem Handyboard anzusprechen zu können, sollte man die neue Version 4 von Interactive C [InterC] benutzt. Der Befehl *sonar()* führt die Messung und Auswertung durch und liefert die Entfernung in Millimeter zurück. Zwischen den Aufrufen von *sonar()* sollte man mindestens 30 ms warten, damit die Sonarechos der letzten Messung verklungen sind. Dies kann einfach mit sleep(0.03) geschehen.

Die durch die Diode ausgesandte Lichtstrahlung wird vom zu untersuchenden Objekt teilweise reflektiert und gelangt anschließend in den Fototransistor. Je nach Stärke des reflektierten Lichts erhöht sich dadurch der Transitorstrom, so dass ein direkter Zusammenhang zwischen Transistorstrom und reflektierter Lichtmenge besteht. Statt des Foto-Transistors kann auch ein Spannungsteiler mit einem Fotowiderstand benutzt werden. Die Beschaffenheit des Untergrundes beeinflusst die Wahl der Fotodiode und auch des einzusetzenden Transistors erheblich. Ein bereits fertig aufgebauter Reflexkoppler ist ein CNY-70 [CNY70] (siehe Abbildung 7.10), der für die Unterscheidung von einzelnen Graustufen geeignet ist. Diese Art Reflexkoppler haben üblicherweise 4 Anschlusspins und arbeiten mit einer IR-Diode.

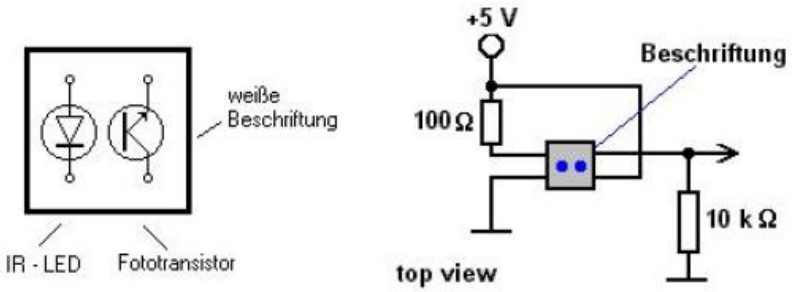


Abbildung 7.10.: Aufbau und Anschluss des CNY70, [CNY70]

Falls eine Graustufenerkennung nicht ausreicht, kann dieser wie folgt modifiziert werden. Die in Abbildung 7.11 vorgestellte Schaltung ermöglicht das sichere Erkennen einer roten Fläche auf weißem Grund.

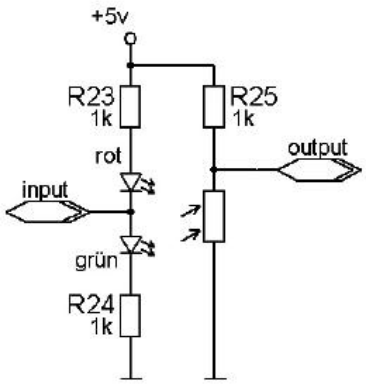


Abbildung 7.11.: Farberkennung mit Reflexkoppler

Der CNY-70 kann diese Aufgabe nicht zufriedenstellend meistern, da die sich die Menge an reflektiertem IR-Licht bei hellgrauen bzw. weißen und roten Flächen nicht stark genug unterscheidet. Die Schaltung verdeutlicht, dass der Reflexkoppler aus zwei LEDs besteht, eine rote und eine grüne. Je nach Zustand des Input-Signals leuchtet eine der beiden LEDs, die jeweils andere ist im Sperrzustand und dunkel. Der rechte Zweig der Schaltung besteht

aus einem einfachen Spannungsteiler, wobei einer der Widerstände ein Fotowiderstand ist. Durch das abwechselnde Schalten der roten und der grünen LED, indem man den Input-Pin abwechselnd auf Plus und Masse legt, wird ein Vergleich des Spannungsabfalls über dem Fotowiderstand zur Erkennung der Farben angestellt. Der Output-Pin wird an einen Analog/Digital-Eingang des Controllerboards angeschlossen. Nun ist zu beachten, dass weißer Untergrund rotes und grünes Licht gleich stark reflektiert. Die über dem Fotowiderstand abfallende Spannung ist also in beiden Fällen gleich groß. Roter Untergrund hingegen reflektiert kaum oder gar kein grünes Licht, aber sehr viel rotes. Es sollte sich also ein großer Spannungsunterschied am Output-Pin messen lassen. Auf die gleiche Weise lassen sich also auch grüne Flächen erkennen. Die Schwarz/Weiß-Erkennung funktioniert mit diesem Reflexkoppler ebenfalls ohne Probleme. Wie alle optischen Sensoren sind auch Reflexkoppler höchst empfindlich gegen direkte Sonneneinstrahlung oder sonstige starke Licht- und IR-Quellen. Je nach Umgebungsbedingungen können sich Messwerte auf der gleichen Fläche relativ stark ändern.

7.5.3. Radencoder

Im Kapitel 3.6.2 wurde erklärt, wie diese Wegmesssensoren funktionieren und dass man sie grob in zwei Arten unterteilen kann: absolute und inkrementelle Messwertgeber. Die absoluten Messwertgeber sind für Bewegungsanalysen wenig geeignet, da sie nur die aktuelle Position der Welle als Signal zurück liefern und überwiegend für die Bestimmung der Position von Roboterarmen dienen. Für den Einsatz im Robotfußball sind inkrementelle Radencoder eher geeignet und lassen sich noch einmal in mechanische und optische Sensoren unterteilen. Die mechanischen Radencoder besitzen 2 Signalausgänge, sind recht groß und teuer. Daher sind sie für Kleinstroboter ungeeignet. Außerdem neigen solche Sensoren zum Prellen, das heißt, anstatt einer klaren Impulsflanke werden mehrere schnell aufeinanderfolgende Impulse erzeugt, bevor sich das Signal stabilisiert. Das Ausgangssignal muss also in der Regel noch geglättet bzw. entprellt werden. Dies kann in Hard- oder Software geschehen. Wird keine Entprellung vorgenommen, sind die gemessenen Wegstrecken zu ungenau.

Abbildung 7.12.: optischer Radencoder (Photoreflektor)

Bei den optischen Encoder sind die Photoreflektoren recht gut für die Wegmessung geeignet, wobei der CNY70 (siehe Abschnitt “Reflexkoppler“) im Labor zur Verfügung steht oder recht günstig zu erhalten ist. Um diesen Sensor für die Wegmessung in das System zu integrieren, kann man zum Beispiel ein Hell-Dunkel-Muster auf der Radinnenseite anbringen (Abbildung 7.12), wobei der Reflexkoppler von unten an die Radinnenseite “schaut“. Man kann auch eine Lochscheibe verwenden, die unmittelbar vor dem Sensor rotiert.

7.5.4. Kompass-Sensoren

Die Integration eines Kompass-Senors stellt eine äußerst interessante Möglichkeit dar, die Richtungsbestimmung und damit die Orientierung des Roboters zu verbessern. Ein Roboter mit Kompass-Sensor wäre in der Lage, seine Ausrichtung autonom, ohne globale Datenquellen, zu bestimmen. Ein geeigneter Sensor KMZ52 des Herstellers Philips [KMZ52] ermöglicht eine Ausrichtungsbestimmung mit einer Genauigkeit von unter 1°. Es handelt sich dabei um einen analogen Sensor, dessen Integration aufgrund seiner Störanfälligkeit als aufwendig angesehen werden muss. Störungen können sowohl von der Digital- oder Leistungselektronik des Roboters sowie von den schnell wechselnden magnetischen Feldern der Motoren ausgehen.
Die Entwicklung und Einbindung dieses Sensors ist aus zeitlichen/finanziellen Gründen nicht im Rahmen dieser Arbeit möglich.

7.5.5. Kamera-Sensoren

Die Integration von Kamera-Sensoren und die damit verbundene Möglichkeit einer lokalen Bildverarbeitung kann den Roboter zu einem vollautonomen System erweitern. Miniaturkameras, die sich für die Integration in einen Fußballroboter eignen, setzen Eigenschaften voraus, welche die Auswahl an geeigneten Sensoren stark einschränken. Dabei benötigt die Kamera einen Ausgang, den man an ein Controllerboard anschließen kann und somit den direken Zugriff auf die Bilddaten erlaubt. Weiterhin muss darauf geachtet werden, dass die Kamera eine gewisse Größe nicht überschreitet, damit eine Integrierung in das Gehäuse des Roboters möglich ist. Wichtig ist auch, dass die Versorgungsspannung der Kamera dem der Betriebsspannung der Roboterelektronik entspricht. Kameras mit CMOS-Technologie, die auch in Webcams Verwendung finden, sind durch ihre geringe größe und Versorgungsspannung geeignet.

7.6. Controllerboard

Das Controllerboard des Roboters stellt sicherlich die schwierigste Designentscheidung dar. Die Recheneinheit muss in der Lage sein, mit allen ausgewählten Peripheriebauteilen effizient zusammenzuarbeiten. Dazu gehören die Anschlussmöglichkeiten für die Sensoren und Aktoren, sowie eine Möglichkeit, eine Kommunikationsschnittstelle zu beinhalten.
An Hand der Planungsphasen zur Konstruktion des Roboters wurde eine Liste von möglichen verwendbaren Peripheriebauteilen erstellt. Darunter sind Anschlussmöglichkeiten für 4 Sharp-, 2 Ball-, 2 Boden-, 1 Kompass- und 4 Beacon-Sensoren, sowie 6 Bumper, 3 Servos, 3 leistungsstarke Motoren, 1 Resetschalter und 3 Infrarot-Sensoren für eine genauere Ballerkennung vorhanden sein. Weiterhin sollte die Rechenleistung des Controllerboards ausreichend sein, um interne Aufgaben, wie z.B. das gleichzeigte Ablaufenlassen mehrerer Tasks zu lösen. Außerdem sollte das Controllerboard energiesparend sein, denn je weniger Energiezellen gebraucht werden, desto leichter ist der Roboter und desto schneller wird der Roboter. Des weiteren sollte das Controllerboard die Möglichkeit besitzen, einen PDA anzuschließen, um die rechenintensiven Algorithmen auf den PDA auszulagern und eine Kommunikation zwischen den Robotern zu verwirklichen. Der Speicher des Controllerboards sollte groß genug sein, um umfangreichen Programmcode zu verwalten.

Durch diese Vorgaben fiel die Konstruktion des Roboters mit dem RCX sowie seines Multiplexer von vornherein aus dem Konzept, da diese Einheit nicht die nötigen Spezifikationen bezüglich der Anschlussmöglichkeiten für die Sensoren und Aktoren besitzt. Weiterhin ist es zur Zeit nicht möglich, eine Verbindung zwischen dem PDA und dem RCX herzustellen.

Die HAW Hamburg besitzt schon seit geraumer Zeit ein Controllerboard (MIT-Board 6.270)[5] [MIT'04] (siehe Abbildung 7.13), welches in den Seminarkursen durch seine vielen Anschlussmöglichkeiten große Beliebtheit errungen hat.

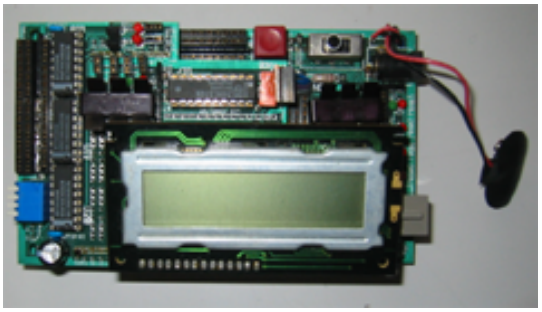


Abbildung 7.13.: MIT-Board 6.270

Die CPU besteht aus einem Motorola 68HC11 mit 2 MHz Taktfrequenz und 32 KByte RAM. Das Board besitzt 8 digitale und 16 analoge Eingänge, einen Servo und vier Motoren-Ausgänge. Die Motoren können über spezielle Treiberstufen, sog. Motortreiber vom Typ

[5] Das MIT-Board 6.270 ist eine Konstruktion von Fred Martin, Pankaj Oberoi und Randy Sargent

L293D (im Board integriert), direkt mit der Akkuspannung versorgt werden. Bei der Auswahl der Motoren ist auch auf die maximale Stromaufnahme zu achten, denn sie liegt bei 600 mA pro Motor. Fließen höhere Ströme, werden die Motortreiber und/oder sogar das gesamte Board zerstört. Für Kontrollen hat das Board ein Zwei-Zeilen-Display. Siehe dazu im Anhang unter MIT-Board 6.270.

Ein weiterer interessanter Kandidat für ein passendes Controllerboard ist das Handy-Board [Handy] (siehe Abbildung 7.14), der Nachfolger des MIT-Board 6.270.

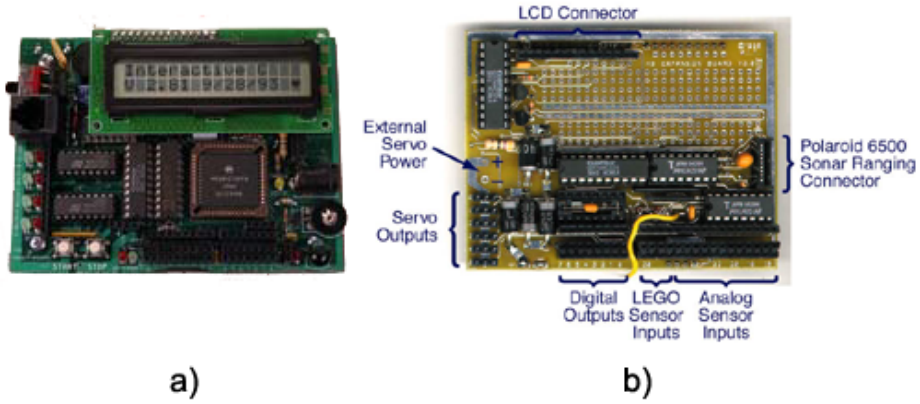


Abbildung 7.14.: Handy Board (a) und Expansion Board für das Handy Board (b), [Handy]

Dieses Board besitzt wie das MIT-Board 6.270 4 Motoren-Ausgänge, jedoch zusätzlich 6 Servo-Ausgänge. Weiterhin sind 6 digitale Ausgänge und 8 digitale Eingänge sowie 19 analoge Eingänge verfügbar. Das Board hat speziell für LEGO-Sensoren 4 analoge Eingänge und ein IR-Ausgang. Ein obligatorisches Display für die Kontrolle der Programme ist im Board auch integriert (Siehe dazu Anhang Handy-Board).

Während der Diplomarbeit besorgte die HAW Hamburg ein Brandenburg-Board[6] [Aksen], welches auch einige interessante Anschlussmöglichkeiten aufweist:

Abbildung 7.15.: Aksen - Das Aktor- und Sensorboard, [Aksen]

[6] Aksen-Board, das Aktor- und Sensorboard

Dieses Board verfügt über vier Motor-Treiber, wobei das Besondere ist, dass hier max. 1000mA zur Verfügung stehen, was eine Steigerung der Motorenauswahl beinhaltet, da die anderen Controllerboards bei ihren Motor-Treiber auf max. 600mA begrenzt sind. Es gibt 3 Servo-Ausgänge, 16 digitale Ein-/Ausgänge und 15 analoge Eingänge. Weiterhin besitzt das Board ein CAN-Interface, 4 Lämpchen-Treiber (max. 600mA), 1 modulierten IR-Ausgang (40 KHz), sowie 3 Encoder-Eingänge für positionsgesteuerte DC-Motoren. Eine LCD-Anzeige ist im Board optional integriert (Siehe Anhang Aksen-Board).

Seit einiger Zeit gibt es auf dem Markt auch sogenannte Mini-ITX Boards [Mini-ITX], welche wie gewöhnliche PCs ausgestattet sind, jedoch nur einen sehr kleinen Platzbedarf haben. Diese Boards sind üblichweise 17 cm x 17 cm groß und mit unterschiedlichen Prozessortypen, sowie zusätzlicher Hardware zu bekommen. Diese Boards sind auf jeden Fall leistungsstärker als die oben angesprochenen Controllerboards. Über die parallele Schnittstelle stehen viele digitale Ein- und Ausgänge zu Verfügung. Weiterhin kann das Board mit spezielle Zusatzhardware für Robotik ihre Anzahl an Anschlussmöglichkeiten stark erhöhen. In der Entwicklung und vereinzelt verfügbar sind die noch kleineren Varianten von 7 cm x 7 cm, die sogenannten Micro-ITX Boards, zu erwähnen. Diese Boards werden mit einer Versorgungsspannung von 12 V betrieben und benötigen daher einen größeren Energiebedarf als die schon vorgestellten Boards.

Abbildung 7.16.: Typisches Mini-ITX Board, [Mini-ITX]

Aus finanziellen Gründen wurde auf die Neuanschaffung einer geeigneten Prozessoreinheit verzichtet und man konzentrierte sich auf das vorhandene und schon lang erprobte System. Dieses System beinhaltet das MIT-Board 6.270, welches nahezu genügend Anschlussmöglichkeiten aufweist. Daher muss bei diesem Board auf einige Hardwarekomponenten verzichten und die erstellte Liste der gewünschten Komponenten reduzieren. Die nun verwendeten Hardwarekomponenten werden im nächsten Kapitel genauer vorgestellt.

7.7. Energieversorgung

Nahezu alle auf dem Markt verfügbaren Sensoren und integrierten Halbleiterschaltungen (ICs) für den Bereich Kleinrobotik benötigen eine Betriebsspannung von 5 V. Zum Betreiben von Motoren und Servomotoren ist aber eine höhere Spannung notwendig. Deshalb haben alle Controllerboards einen passenden Spannungswandler integriert, um die benötige Spannung für den Prozessor sowie für Aktoren und Sensoren zu gewährleisten.
Das MIT-Board6.270 hat einen eigenen Anschluss für die Versorgung des Prozessors und die Sensoren sowie einen zusätzlichen Anschluss für die Motoren und den Servo. Üblicherweise wird an den Prozessor und den Sensoren eine Spannung zwischen 5,5 V - 6 V angelegt. Die Motoren und der Servo erhalten eine Spannung von 6 V - 7,2 V.
Auf dem Handyboard ist ein Spannungswandler vom Typ L78S05 integriert, der die angelegte Akkuspannung auf die benötigten 5 V herunterregelt. Die Motoren werden direkt mit der anliegenden Akkuspannung betrieben. Servomotoren arbeiten nur mit Spannungen von 4,8 V bis 6 V. Auf dem Handyboard befinden sich 5 Dioden, die zusammen die Akkuspannung um ca. 3,5 V verringen. An jeder Diode fallen etwa 0,7 V ab. Damit beträgt die maximal verwendbare Akkuspannung 9,6 V, denn 9,6 V Versorgungsspannung minus 3,5 V Diodenspannung bleiben 6,1 V für die Servos übrig, was gerade noch zulässig ist.

7.7.1. Hinweise zu Auswahl und Anschluss der Akkus

Es gibt prinzipiell 4 Arten von Akkus: NiCd (Nickel-Cadmium), NiMH (Nickel- Metallhybrid), Blei-, bzw. Bleigel-Akkus und Lithium-Ionen-Akkus. Empfehlenswert sind ausschließlich NiMHAkkus mit einer Spannung von 1,2 V. Blei- bzw. Bleigel-Akkus sind nur für Spannungen von 6 V bzw. 12 V erhältlich, sind recht schwer und für einige Boards auch nicht geeignet. NiCd-Akkus haben einen ausgeprägten Memory-Effekt, d.h., sie müssen immer vollständig entladen werden, bevor man sie, ohne die Lebensdauer zu reduzieren, wieder aufladen kann. NiMH-Akkus haben nahezu keinen Memory-Effekt und zusätzlich eine höhere Energiedichte. Die teuere Alternative sind die Lithium-Ionen-Akkus, die wesentlich empfindlicher auf falsche Behandlung als andere Akkus reagieren. Entscheidend für die Laufzeit des Roboters ist die Kapazität der Akkus. 1600 mAh stellen hier das Minimum dar.

7.8. Zusätzliche Hardware

Für die Interaktion eines PDA's mit dem Controllerboard steht diesem Projekt ein HP[7] zur Verfügung.
Durch seine Abmessung von 8.4 cm x 1.6 cm x 13.8 cm und einem Gewicht von nur 207 g ist dieser Pocket PC für die integration in das Robotersystem geeignet. Er verfügt über einen Prozessor von Intel (PXA250) mit 400 MHz Taktfrequenz, sein ROM ist mit 48 MB groß und der RAM liegt bei 128 MB. Weiterhin ist ein 3.8“ TFT - Aktivmatrix - 16 Bit (64k Farben) Display integriert sowie diverse Extras, wie z.B. Lautsprecher, Audioein- und -ausgang.

[7] Hewlett-Packard [HP'04]: iPAQ Pocket PC H5550

Dieser Pocket PC unterstützt sowohl Bluetooth als auch WLAN (Wireless Local Area Network) nach 802.11b, um sich mit Funknetzen zu verbinden. Als Betriebssystem kommt Microsoft Windows Mobile 2003 Premium Software for Pocket PC [MpocketPC] zum Einsatz. Es handelt sich um ein Betriebssystem, was multitaskfähig ist.

Abbildung 7.17.: Hewlett-Packard: iPAQ Pocket PC H5550

8. Konstruktion und Realisierung

In diesem Kapitel geht es um die Konstruktion und Realisierung des prototypischen Roboters, wobei genau auf die einzelnen Konstruktionsteile des Roboters unter Berücksichtigung der beschriebenen Zielsetzungen und getroffener Hardwareauswahl eingegangen wird. Im Weiteren wird auf den Bereich Software eingegangen, deren Entwicklungsumgebungen, die Programmarchitektur, Bibliothek des Antriebssystems und Spielstrategien. Zum Schluss wird das Ergebnis dieser Realisierung diskutiert und ein Ausblick auf zukünftige Arbeiten gegeben.

8.1. Konstruktion und Auswahl der Hardware

8.1.1. Plattform des Roboters

Vor Beginn des Entwurfs des Roboters wird noch mal eine Liste der Punkte zusammengetragen, die der Roboter erfüllen soll. Der wichtigste Punkt ist, dass der Roboter die konstruktionsspezifischen Maße der RoboCup-Junior-Soccer-League einhalten soll. Das heißt, dass der Roboter nicht die 22 cm Durchmesser überschreiten darf und eine Höhe von 22 cm einhält. Diese Angaben zwingen dem Konstrukteur, den Roboter so klein wie möglich zu gestalten und jeden Zentimeter des zur Verfügung stehenden Platzes optimal auszunutzen. Deshalb war in der Anfangsphase der Konstruktion die Plattform rund, um die maximale Fläche mit 22 cm Durchmesser auszunutzen. Da für eine Ballführung gerade Kanten benötigt wurden, mußte der Kreis damit versehen werden. Diese Kanten müssen zwischen den Antriebsrädern liegen. Dies wird durch eine Grundform der Platform in Form eines gleichmäßigen Sechsecks erreicht. Für den Schutz der Antriebsrädern bleiben an deren Seiten die Rundungen erhalten (siehe Skizze 8.1).

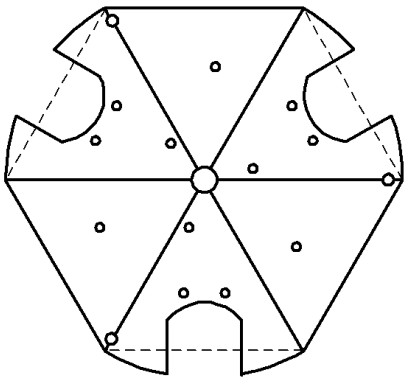


Abbildung 8.1.: Skizze der Grundform der Roboterplattform

Weiterhin soll die Plattform möglichst leicht umzubauen sein, um evtl. Konstruktionsteile zu ersetzen bzw. bei der Installation der Elektronik besser an die Teilbereiche des Roboters zu gelangen. Die Umgebungssensoren sowie die Ballsensoren werden auf Plattformen installiert, die durch Drehung der Plattform in jede mögliche Richtung schauen können. Ein weiterer Punkt ist, dass der Roboter ein Rundum-Bumper-System bekommt, das bei Kollision des Roboters mit Hindernissen wie z.B. der Wand oder bei Ausfall der Umgebungssensoren größere Schäden am Roboter verhindern soll. Der Roboter soll bzgl. seiner Bewegung recht flexibel sein, deshalb wird ein neues Antriebssystem realisiert: das im Kapitel 3 erwähnte Antriebssystem des “omnidirectional“ - Antrieb. Dies soll für diesen Roboter entwickelt und realisiert werden, da es durch seine vielseitigen Bewegungsmöglichkeiten dem anderen Antriebssystemen überlegen ist.

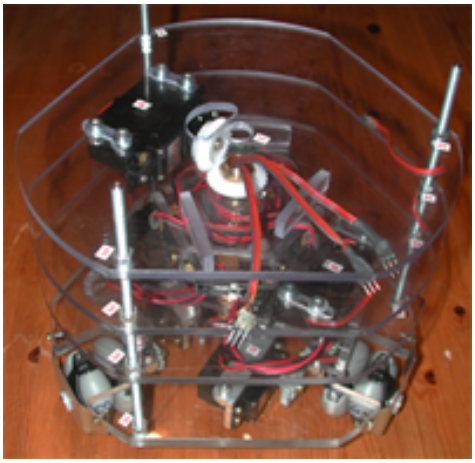


Abbildung 8.2.: Prototypischer Aufbau der Plattformen

Durch diese Vorgaben wird die Überlegung angestellt, das Robotersystem in verschiedene Ebenen zu unterteilen, um eine Unabhängigkeit der verschiedenen Plattformen zu gewährleisten. Dabei beinhaltet die untere Ebene das Antriebssystem und die Ball-Sensoren. Die nächste Ebene soll nur für die Umgebungssensoren zuständig sein. Danach wird die Elektronik- und Batterie-Ebene entstehen. Die oberste Ebene ist für das Controllerboard und gegebenenfall den PDA zuständig. Bevor genauer über die Konstruktion und Installation der einzelnen Ebenen eingegangen wird, soll vorher über die zu verwendete Hardware geredet werden.

8.1.2. Peripheriebauteile des Roboters

Wie in Kapitel 7 erwähnt, wird der Roboter mit dem MIT-Board 6.270 verwirklicht. Dabei muss die Anzahl der gewünschten Peripheriebauteile reduziert werden, denn die Anschlussmöglichkeiten des MIT-Board 6.270 sind nicht so ausgeprägt wie bei manch anderen Controllerboards. Deshalb werden nur die wichtigsten Bauteile installiert.
Für das “omnidirectional“ - Antriebssystem benötigt der Roboter 3 gleichwertige Motoren. Es werden 2 Ballsensoren auf einer drehbaren Plattform direkt in der Antriebsebene installiert,

so dass für die drehbare Plattform ein Servo benötigt wird. Die Umgebungssensoren werden mit 4 Sharp-Sensoren ausgestattet, die über einen Verstärker[1] (siehe im Anhang) direkt an das Controllerboard angeschlossen und im System drehbar gehalten werden. Da das MIT-Board 6.270 jedoch nur einen Anschluss für Servomotoren zur Verfügung stellt, muss diese Plattformdrehung mit einem normalen Motor realisiert werden. Es muss auf der Ebene der Umgebungssensoren eine Art Kontaktsystem installiert werden, so dass das Controllerboard die momentane Position der Umgebungssensoren ermitteln kann. Um den Roboter mit dem angesprochenen Bumper-System auszustatten, benötigt der Roboter 6 Mikroschalter, die an jeder Seite des Roboter befestigt werden. Weiterhin braucht der Roboter 3 Beacon-Empfänger, um das Tor zu erkennen. Ein Reset-Schalter, bestehend aus einem Mikroschalter, soll für den Notfall oder das Beenden des Programms für den Benutzer gut zugänglich sein. Da die Spielfläche keine Grauabstufung mehr besitzt, kann man auf die Bodensensoren (Reflexkoppler) verzichten. Auch auf Radencoder muss man verzichten, da die “omnidirectional wheels“ und die zur Verfügung stehenden Motoren viel Platz auf der Antriebsebene benötigen, vor allem, da die Motoren noch eine “Untersetzung“ benötigen, um ein ausreichend hohes Drehmoment der Räder zu gewährleisten. Auf einen Kompass- und Kamera-Sensor wurde aus Kostengründen verzichtet. Durch diese Angaben ist die folgende Liste entstanden:

- 3 “omnidirektional wheels“
- 3 Motoren für das Antriebsystem
- 1 Motor für die Umgebungssensoren
- 1 Servo für die Ballsensoren
- 2 Ballsensoren
- 6 Mikroschalter für das Bumpersystem
- 1 Mikroschalter für den “Notausschalter“
- 4 Sharp-Sensoren
- 1 Verstärker für die Sharp-Sensoren
- 3 Beacon-Sensoren
- 1 IR-Sensor (Sender und Empfänger), um den Ball am Roboter zu identifizieren

[1] Dient auch als Schutz für das Controllerboard und versorgt die Sharp-Sensoren mit Energie.

8.1.3. Antriebs-Ebene

Die Antriebs-Ebene erwies sich während der Konstruktion als der schwierigste Teil des Roboters. Trotz früher Entscheidung, das “omnidirektional“ Antriebssystem zu integrieren, gab es bei der Konstruktion immer wieder Probleme mit der Umsetzung, da es bei dieser Ebene sehr wichtig ist, dass die Motoren alle symmetrisch auf die Plattform angeordnet sind. Bei dem ersten Versuchen wurden die Motoren direkt an den “Allseitenräder“ (“omnidirection wheel“) angebracht. Die Motoren, die für die Entwicklung zur Verfügung stehen, haben jedoch ein sehr geringes Drehmoment, wodurch der Roboter durch die Direktmontage sehr langsam wurde. Deshalb konzentrierte man sich auf ein Konzept der “Untersetzung“. Dabei wurde versucht, die Motoren und die Räder über einen Riemensystem anzutreiben, um das Getriebe im Inneren des Motors zu entlasten bzw. dies vor Beschädigung zu schützen. Am Motor selbst wurde eine größere Antriebsscheibe befestigt als an dem Rad selber, wodurch der Roboter theoretisch schneller wird. Dieses Konzept wurde aus Mangel an Leistungsübertragung verworfen, denn der Riemen rutschte bei Belastung durch. Man konzentrierte sich deshalb auf eine Zahnraduntersetzung. Bei der Wahl der möglichen Untersetzungen wurde mehr Wert auf Kosten der Geschwindigkeit als auf Kraftübertragung gelegt. Das “Allseitenrad“ bekam ein kleines Zahlrad mit 16 Zähnen und der Motor ein größeres Zahnrad mit 24 Zähnen. Dies entspricht einem Untersetzungsverhältnis von 1:1,5.

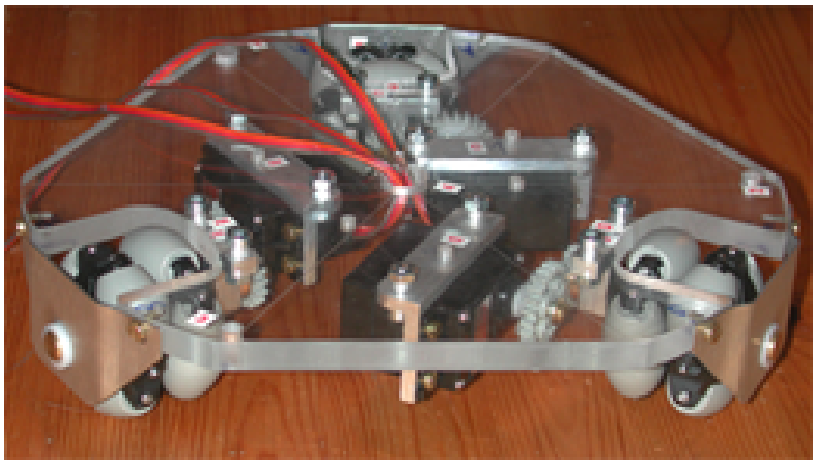


Abbildung 8.3.: Antriebsebene

Dieses Antriebssystem mit drei “omnidirectional“ Allseitenräder erlaubt dem Roboter, viele verschiedene Richtungen direkt aus dem Stand anzufahren. Dabei sind die wichtigsten Himmelsrichtungen Norden, Nordosten, Südosten, Süden, Südwesten und Nordwesten. Dies liegt an der 120° Anordnung der Antriebsachsen. Weitere Informationen über die resultierenden möglichen Fahrtrichtungen erhält man im Abschnitt 8.2.3. Nach der Installation des Antriebssystems montierte man auf der Platte eine weitere kleinere Zwischenplattform für die Ballsensoren. Diese Sensorenplattform wurde mit einer Elektrostange mit einem Durchmesser von M13 in der Mitte der Antriebsplattform verbunden, so dass sie sich auf der Ebene drehen kann. (siehe Skizze 8.4) Am beiden Enden der Zwischenplattform wurden die Ball-

sensoren angebracht und die Verbindungskabel in die Elektrostange eingeführt, so dass sich die Kabel bei der Drehung des Systems nicht aufwickeln. Diese Anordnung der zwei Ballsensoren hat den Vorteil, dass der Roboter durch die Drehung der Sensoren den Ball schneller finden kann, als wenn das System nur einen Ballsensor besitzt. Weiterhin müsste die Plattform mit einem Ballsensor komplett umgebaut werden, da der Servo, der das System antreiben soll, nur Winkel von $0°$ bis $180°$ anfahren kann. Durch diese Einschränkung müsste eine Untersetzung installiert werden, welches sehr viel Platz beansprucht.

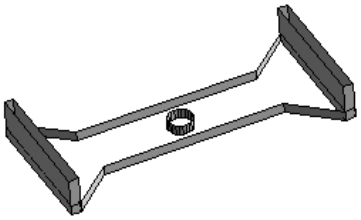


Abbildung 8.4.: Skizze für die Ebene der Ballsensoren

8.1.4. Sensoren-Ebene

Direkt über der Antriebsebene wurde mit drei Gewindestangen die zweite Ebene befestigt. Auf der Unterseite dieser Ebene wurde der Servo angebracht, der für die Drehung der Ballsensorplattform zuständig ist und über zwei Winkelzahnräder angetrieben wird. Auf der Oberseite der zweiten Ebene wurde, wie auch schon auf der Antriebsebene, eine weitere Zwischenplattform für die Umgebungssensoren angebracht. Dabei entspricht diese Plattform ein Sechseck, wobei drei Sharpsensoren nebeneinander angebracht sind und ein Sensor nach hinten ausgerichtet ist (siehe Skizze 8.5).

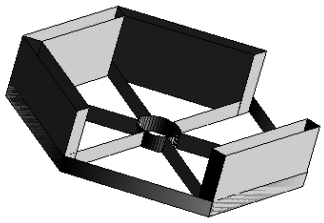


Abbildung 8.5.: Skizze für die Ebene der Umgebungssensoren

Da das MIT-Board 6.270 nur einen Servo-Anschluss zur Verfügung steht, muss diese Umgebungssensorplattform mit Hilfe eines Motor angetrieben werden. Damit nun die Sharpsensoren die geforderte Position einnehmen können, wurde auf der zweiten Ebene sechs Kontaktstellen in der Mitte der wichtigsten Himmelsrichtungen(N, NO, SO, S, SW und NW) angebracht. Da unterhalb der Umgebungssensorenplattform auch eine Kontaktstelle angebracht ist und bei Drehung dieser Plattform die Kontaktstellen beider Ebenen einen Kurzschluss verursachen, kann das System erkennen, dass sie in der korrekten Position sind.

Die sechs Kontaktstellen sind unterteilt in Norden, gerade und ungerade Kontaktstellen (siehe Skizze 8.6).

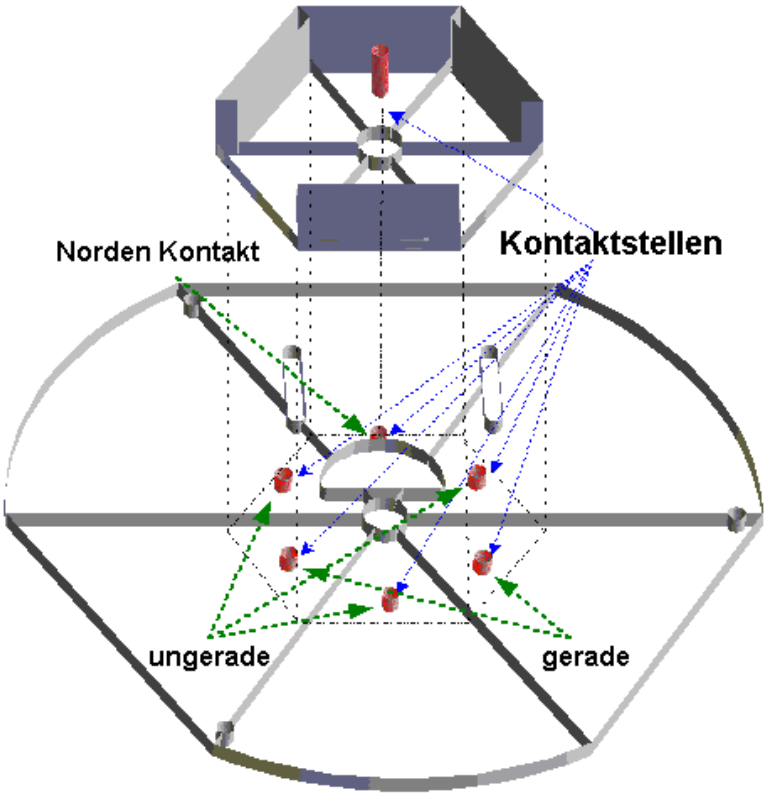


Abbildung 8.6.: Kontaktstellen für die Kontrolle bei der Drehen der Umgebungssensoren

Dadurch kann das System auch genau identifizieren, in welche Richtung die Sharp-Sensoren gerade schauen. Nach einigen Testläufen dieser Sensorplattform stellte man fest, dass die Kontaktstellen nicht richtig identifiziert werden und es zu Überdrehungen der Kabel der Sharp-Sensoren kam. Dies führte wiederum zu einem Bruch der Kabel. Deshalb wurden Stopper installiert, die ein Überdrehen der Plattform verhindern sollen. Der Motor, der die Umgebungssensoren über Winkelzahnräder antreibt, ist aus Platzgründen auf der nächsten Ebene integriert. Auf dieser Ebene sind auch die Akkus für das Controllerboard, die Akkus für Sharp-Sensoren und den benötigten Verstärker für die Sharp-Sensoren zu finden. Die letzte Ebene umfasst das Controllerboard, gegebenenfalls einen PDA und Beacon-Empfänger für die Torkennung. Die Beacon-Sensoren sollten nach den ersten Überlegungen wie die Ball- und Umgebungssensoren auch drehbar gehalten werden, jedoch wurde dieses Konzept aus Mangel an Platz und Servo-Anschlussmöglichkeiten verworfen. Die Beacon-Sensoren wurden zwischen zwei Antriebsachsen fest nach vorne hin ausgerichtet angebracht. Auf der Höhe der zweiten Ebene wurden die Mikroschalter für die Kollisionserkennung in der Mitte der wichtigsten Himmelsrichtungen angebracht. Dünne Metallstücke, die auf Federn gelagert sind, dienen dem Mikroschalter als größere Berührungsfläche.

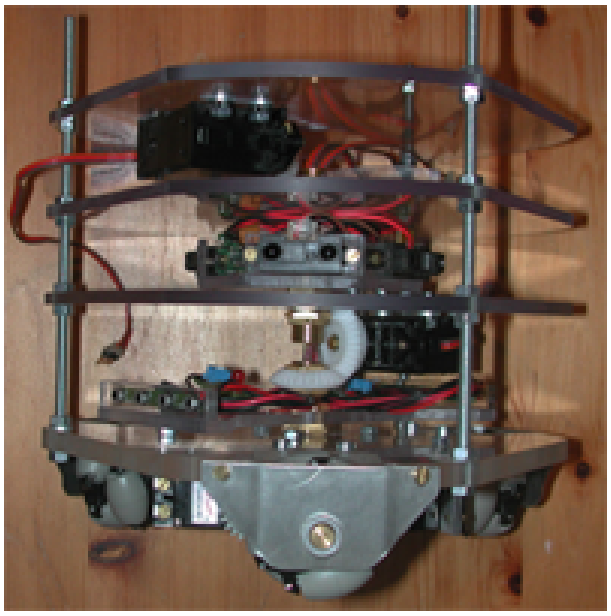


Abbildung 8.7.: Alle Ebenen des Roboters

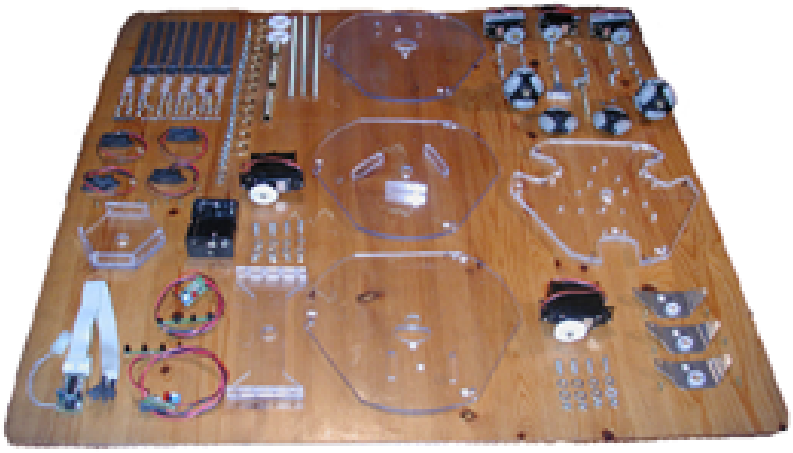


Abbildung 8.8.: Einzelteile der Roboterplattform

8.2. Software

8.2.1. Entwicklungsumgebung

8.2.1.1. Die Programmiersprache Interactive C

Zum Programmieren des Controllerboards (MIT-Board 6.270) wird das Programm Interactive C Version 3.2 von Newton Research Labs [Newton] verwendet, welches eine abgespeckte Version der Programmiersprache C ist. Interactive C bietet eine Bibliothek mit diversen Funktionen, die speziell für das MIT-Board 6.270 entwickelt wurde. Jedoch bietet sie leider nur einfache Kontrollstrukturen wie *for*, *while*, *if-else* und *break* sowie ein einfaches Multitasking[2].

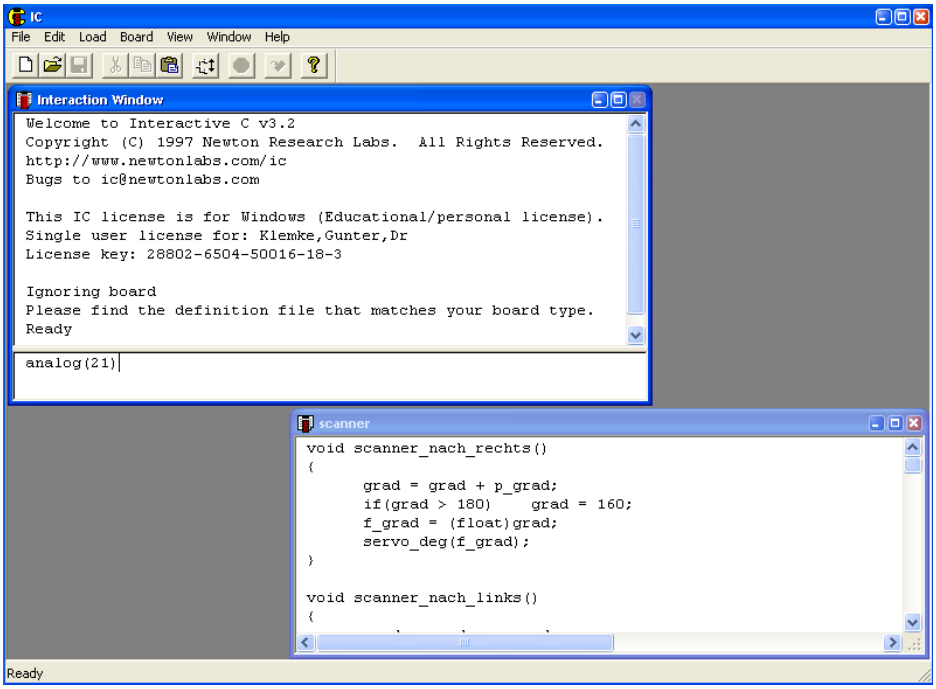


Abbildung 8.9.: Entwicklungsumgebung der Programmiersprache Interactive C

In der Interactive C-Bibliothek gibt es Funktionen für das Einlesen von Sensordaten und die Bereitstellung von Steuerdaten auf die Ausgänge. Es gibt Zeit-, Ton-, Menü- und Diagnose-Funktionen. Die für die Steuerung des Roboters verwendete Programmiersprache Interactive C bietet diverse Funktionen, darunter die Funktionen für die Beacon-Erkennung. Die Funktionen sind für den Empfang von Infrarot Signalen bestimmt.

```
void ir_receive_on()
void ir_receive_off()
void set_ir_receive_frequency(int f)
int ir_counts(int p)
```

[2] was versteht man hier unter einfach?

Die Entwicklungsumgebung von Interactive C bietet eine grafische Oberfläche zur Entwicklung und Übertragung von Programmen auf den Roboter. Diese Oberfläche beinhaltet die Möglichkeit, Statusmeldungen während der Übertragung von Code zu überwachen, wobei es im unteren Bereich der Oberfläche einen kleinen Abschnitt gibt, in dem man direkt Funktionen aufrufen und damit die Funktionstüchtigkeit des Roboters überprüfen kann.
Beim Aufruf von Interactive C versucht die Entwicklungsumgebung das Board des Roboters anzusprechen, um den PCode für den Grundbetrieb des Boards herunterzuladen. Falls kein Board an die serielle Schnittstelle angeschlossen ist, kann man in den Einstellungen angeben, dass das Board ignoriert werden soll, um das Arbeiten am Programmcode auch ohne Anschluss des Boards zu ermöglichen. Weitere Funktionalitäten der Entwicklungsumgebung von Interactive C sind einfache Editorfunktionen wie *Find*, *Replace* und *Goto Line*. Eine Hilfe zu den verschiedenen zur Verfügung stehenden Funktionen wird ebenfalls angeboten.

8.2.2. Programme für den Pocket PC

Für die Entwicklung der Software für den PDA wird die kostenlose Entwicklungsumgebung Visual C++ 3.0 von Microsoft Embedded benutzt. Für weitere Informationen wird auf die Diplomarbeit von Mirco Gerling [Gerling] hingewiesen, der die Schnittstelle zwischen dem Pocket PC und dem Controllerboard entwickelte.

8.2.3. Bibliothek des “omnidirectional“ - Antriebssystems

Das Besondere an diesem Roboter ist sein Antriebssystem, welches es ihm erlaubt, in jede beliebige Richtung zu fahren, ohne sich vorher in die entsprechende Fahrtrichtung drehen zu müssen. Da es die Bibliothek der Motorsteuerung von Interactive C erlaubt, den Motor in 8 Geschwindigkeitsstufen zu betreiben und er entweder nach rechts oder links drehen kann, gibt es daher 3374[3] verschiedene Bewegungsmöglichkeiten, die der Roboter bewältigen kann, zuzüglich dem Stillstand des Roboters. Da sich jedoch einige Bewegungsmöglichkeiten nur durch die Geschwindigkeit unterscheiden, ist die Anzahl der Richtungen, die der Roboter einschlagen kann, geringer als die genannten Möglichkeiten. Bei der Erstellung der Bibliothek für die eigentliche Aufgabe des Fußballspielens wurden nur 27 mögliche Bewegungsabläufe deklariert (siehe Abbildung 8.10), da man davon ausgeht, dass der Motor entweder mit voller Leistung angesprochen wird oder still steht.

[3] In der Interactive C - Bibliothek sind für die Motoreinstellung 8 Geschwindigkeitsstufen vorgesehen (von 0% – 100% Leistung). Das entspricht für jeden Motor 15 Geschwindigkeitseinstellung, da der Motor rechtsrum, sowie linksrum laufen kann (7 rechtsrum, 7 linksrum und Stopp). Wenn man nun die 3 Motoren mit je 15 Möglichkeiten mit einander multipliziert (15 x 15 x 15), erhält man die möglichen Kombinationen von 3375

	Motor 1	Motor 2	Motor 3	Bewegungsabläufe
1	-100	-100	-100	im Stand linksrum drehen
2	-100	-100	0	Radius um Rad 3 / linksrum
3	-100	-100	100	Radius ca. 48 cm um Rad 3 / linksrum
4	-100	0	-100	Radius um Rad 2 / linksrum
5	-100	0	0	Radius ca. 7.5 cm über Rad 2 und 3 / linksrum
6	-100	0	100	gerade aus Richtung SW (Südwest)
7	-100	100	-100	Radius ca. 48 cm um Rad 2 / linksrum
8	-100	100	0	gerade aus Richtung S (Süden)
9	-100	100	100	Radius ca. 48 cm um Rad 1 / rechtsrum
10	0	-100	-100	Radius um Rad 1 / linksrum
11	0	-100	0	Radius ca. 7.5 cm über Rad 1 und 3 / linksrum
12	0	-100	100	gerade aus Richtung NW (Nordwest)
13	0	0	-100	Radius ca. 7.5 cm über Rad 1 und 2 / linksrum
14	0	0	0	Roboter steht still
15	0	0	100	Radius ca. 7.5 cm über Rad 1 und 2 / rechtsrum
16	0	100	-100	gerade aus Richtung SO (Südost)
17	0	100	0	Radius ca. 7.5 cm über Rad 1 und 3 / rechtsrum
18	0	100	100	Radius um Rad 1 / rechtsrum
19	100	-100	-100	Radius ca. 48 cm um Rad 1 / linksrum
20	100	-100	0	gerade aus Richtung N (Norden)
21	100	-100	100	Radius ca. 48 cm um Rad 2 / rechtsrum
22	100	0	-100	gerade aus Richtung NO (Nordost)
23	100	0	0	Radius ca. 7.5 cm über Rad 2 und 3 / rechtsrum
24	100	0	100	Radius um Rad 2 / rechtsrum
25	100	100	-100	Radius ca. 48 cm um Rad 3 / rechtsrum
26	100	100	0	Radius um Rad 3 / rechtsrum
27	100	100	100	im Stand rechtsrum drehen

Abbildung 8.10.: Bewegungsmöglichkeiten des Antriebssystems

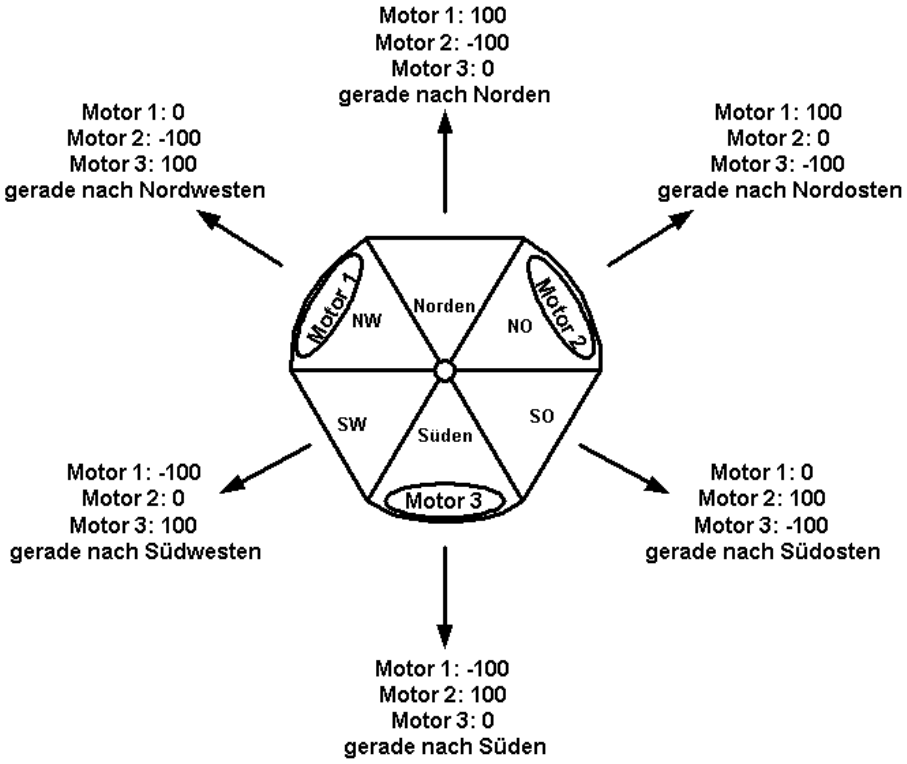


Abbildung 8.11.: Bewegungsmöglichkteiten: Himmelsrichtungen

Dies wurde dadurch entschieden, da das Antriebssystem nicht die notwendige Geschwindigkeit aufweist, um mit kleineren Leistungen der Motoren zu arbeiten. Die wichtigsten Richtungen dieser Bibliothek sind die Himmelsrichtungen Norden, Nordosten, Südosten, Südwesten und Nordwesten(siehe Abbildung 8.11), sich im Stand nach rechts oder links drehen oder sich bei Dauerbetrieb bei bestimmten Motoreinstellungen um verschiedene Radien um einem dadurch definierten Punkt zu drehen (siehe Abbildung 8.12 und 8.13). Hinzu kommt der Stillstand des Motors.

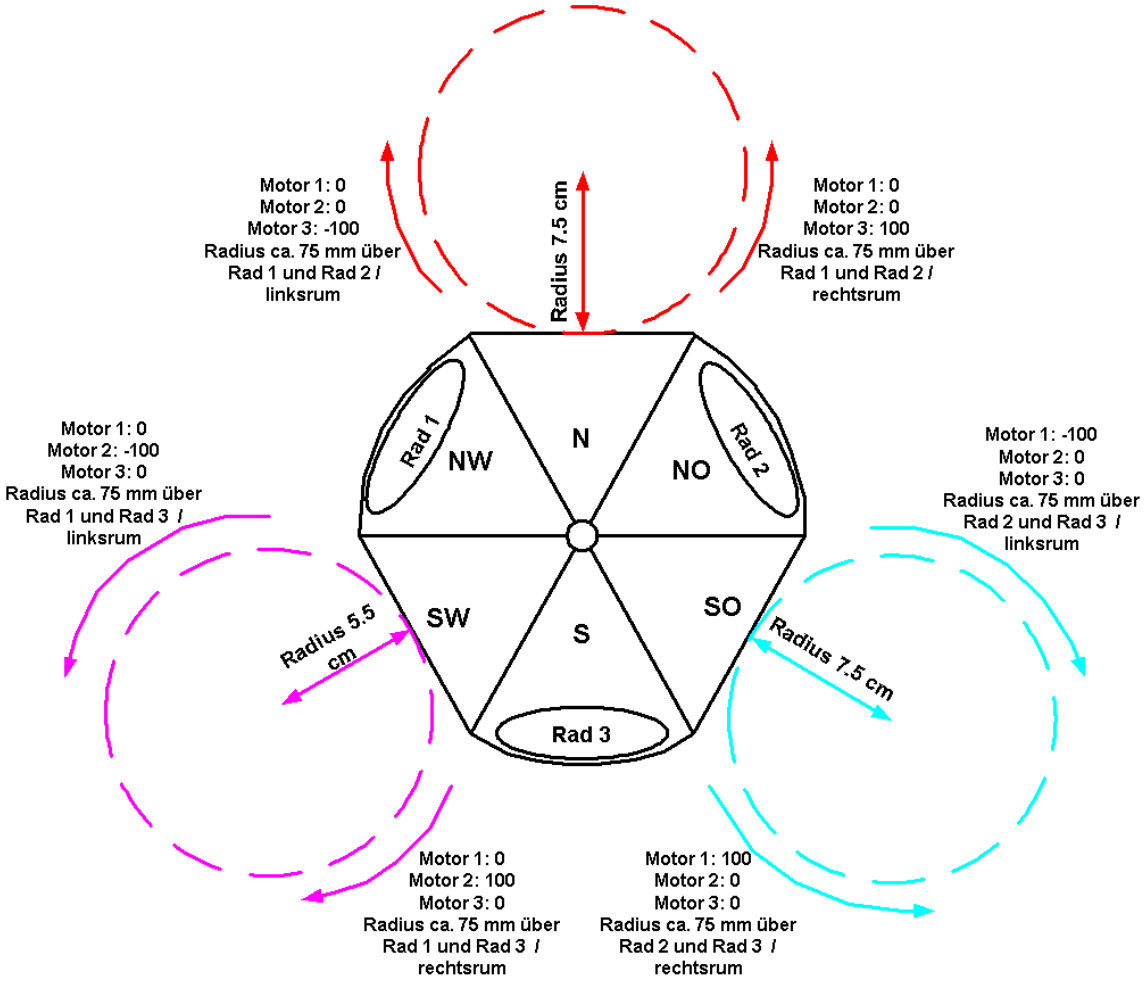


Abbildung 8.12.: Bewegungsmöglichkeiten: Radius 7.5 cm über Rad X und Rad Y

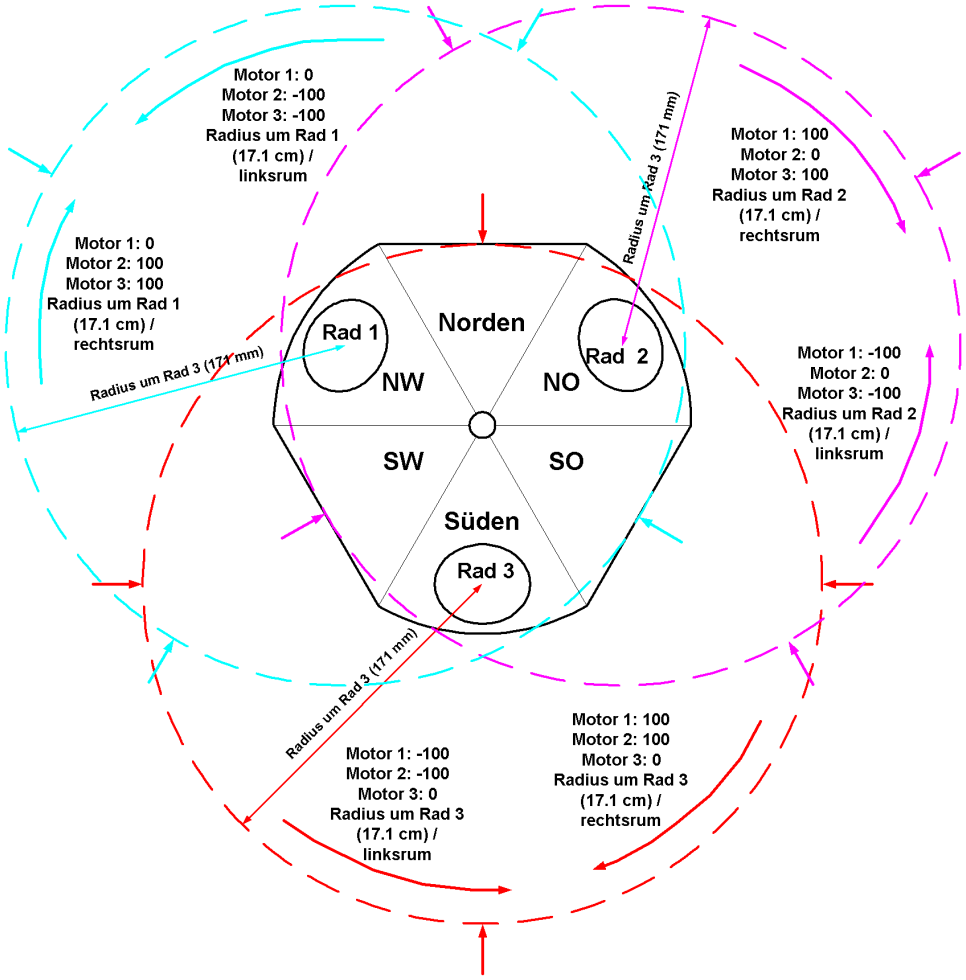


Abbildung 8.13.: Bewegungsmöglichkeiten: Radius um Rad X

Es gibt nun verschiedene Möglichkeiten, eine Bibliothek für dieses Antriebssystem zu erstellen. Eine Möglichkeit besteht darin, alle 3375 Bewegungsmöglichkeiten in einer Datei “Bewegung“ zu deklarieren. Dies kann wie folgt aussehen:

Möglichkeit 1:

```
void stop()
{
motor(mot_1,stufe0);
motor(mot_2,stufe0);
motor(mot_3,stufe0);
}

void norden()
{
motor(mot_,stufe7);
motor(mot_,stufe7);
}

void sueden()
{
motor(mot_,stufe7);
motor(mot_,stufe7);
}
```

Diese Funktionen können dann im eigentlichen Programm aufgerufen werden, in dem man zum Beispiel alle möglichen Situation, die der Roboter meistern muss, in einem Zustands-Automaten unterbringt.

```
/* action 0 : Stop - undefinierte Situation */
if(state == 0) stop();

/* action 1: fahre nach Norden */
else if(state == 1) norden();

/* action 2: fahre nach Sueden */
else if(state == 2) sueden();
```

Während der Programmierung wurde jedoch aus Mangel an Hauptspeicher im MIT-Board 6.270 in der Datei “Bewegung“ nur eine Funktion deklariert, die die Leistung der einzelnen Motoren erwartet.

Möglichkeit 2:

```
void antrieb(int m1, int m2, int m3)
{
motor(mot_1,m1);
motor(mot_2,m2);
motor(mot_3,m3);
}
```

Um nun die erwünschten Bewegungsmöglichkeiten zu bekommen, wird im Programm die Funktion folgendermaßen aufgerufen:

```
/* action 0 : Stop - undefinierte Situation */
if(state == 0) antrieb(stufe0, stufe0, stufe0);

/* action 1: fahre nach Norden */
else if(state == 1) antrieb(stufe7, -stufe7, stufe0);

/* action 2: fahre nach Sueden */
else if(state == 2) antrieb(-stufe7, stufe7, stufe0);
```

Diese Art Bibliothek geht davon aus, dass der Programmierer die verschiedenen Bewegungsmöglichkeiten im Einzelnen kennt und sie entsprechend aufrufen kann.
Eine weitere Möglichkeit ist es mit Hilfe von Formeln die Motorleistungen genau zu regeln, um bestimmte Richtungen einzuschlagen (siehe Abbildung 8.14).

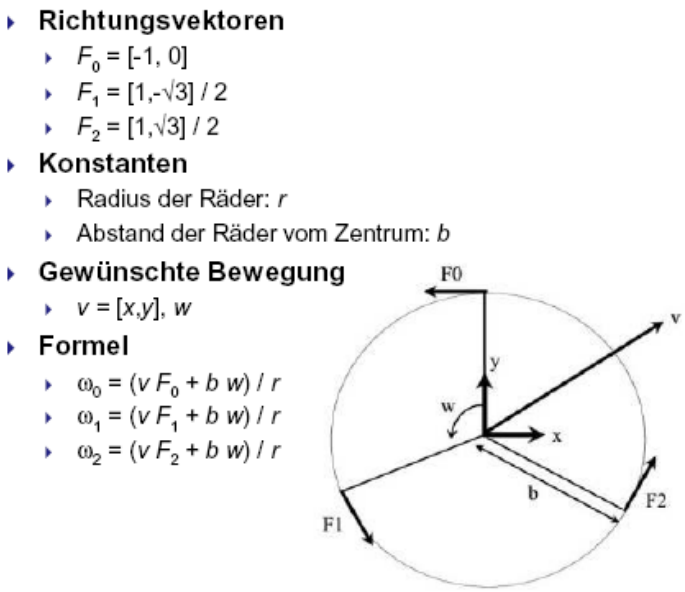


Abbildung 8.14.: Möglichkeit den Roboter mit Formeln zu steuern, [TR'02]

Eine solche Bibliothek zu entwickeln, bei der der Programmierer nur anzugeben braucht, in welchem Winkel sich der Roboter bewegen soll, ist aus konstruktionellen Gründen nicht einfach zu verwirklichen. Es erweist sich als problematisch, dass die im Rahmen dieser Diplomarbeit verwendeten Motoren nicht die benötigten, verschiedenen Geschwindigkeitsstufen umsetzten können. Falls die Motoren nur geringfügig voneinander unterschiedliche Leistungen aufweisen oder das Gewicht des Roboters nicht exakt ausbalanciert ist, kann der Roboter zudem nicht die gewünschten Bewegungsrichtungen einhalten. Dieses Manko könnte man mit Hilfe von Radencodern zwar nicht beseitigen, jedoch verbessern. Aber da weder Platz noch Anschlussmöglichkeiten für die Radencoder vorhanden ist, wird auf die speichersparende Möglichkeit zurückgegriffen und das Manko "in Kauf" genommen.

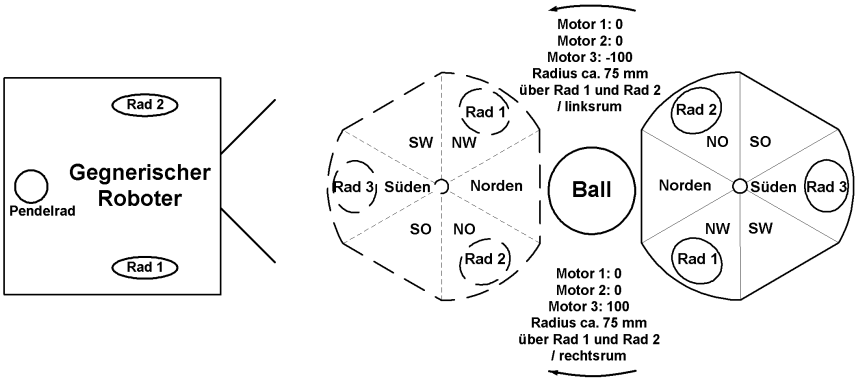


Abbildung 8.15.: Möglicher Spielzug: Ballabschirmen

Im folgenden Abschnitt werden Spielsituationen graphisch dargestellt, um die Flexibilität des Antriebssystems zu verdeutlichen. In der Abbildung 8.15 ist eine typische Situation illustriert, bei der zwei Roboter den Ball unter Kontrolle bekommen möchten. Dabei ist der gegnerische Roboter, welcher über ein normalen Differentialantrieb verfügt, dem Roboter mit der experimentellen Plattform unterlegen, da das Drei-Achsen-Antriebsystem in der Lage ist, durch geschicktes Umrunden des Balls diesen vor dem gegnerischen Roboter abzuschirmen.
Ein weiterer Vorteil der experimentellen Antriebssystem ist, dass der Roboter beim “Wall Follow“[4] (siehe Abbildung 8.16) bzw. “Line Follow“[5] die maximale Geschwindigkeit beibehalten kann und nur mit dem Rad, welches für das Geradeausfahren nicht verantwortlich ist, den Abstand zur Wand bzw. zur Linie steuert. Beim Differenzialantrieb müssen die beiden Räder unterschiedliche Geschwindigkeiten aufweisen, um gegebenenfalls den Abstand zur Wand bzw. zur Linie zu korrigieren.

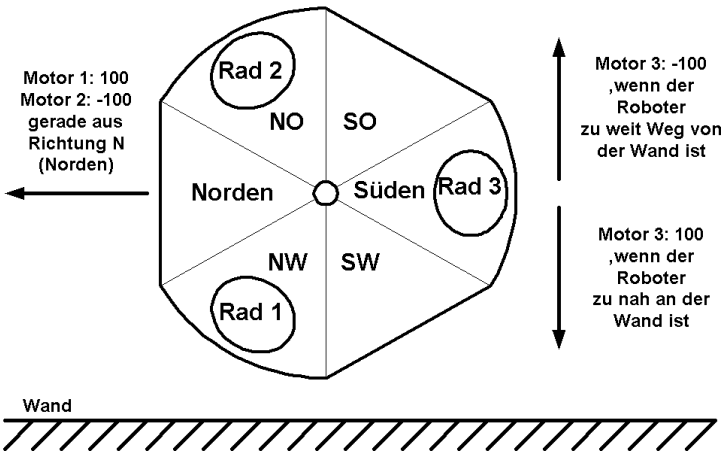


Abbildung 8.16.: Bewegungsablauf beim “Wall Follow“

[4]“Wall Follow“ Der Roboter fährt in einem bestimmten Abstand an der Wand lang.
[5]“Line Follow“ Der Roboter fährt auf einer Linie entlag, wobei der Roboter mit einem Reflexkoppler den Farbunterschied der Linie und dem Untergrund analysieren kann

Abbildung 8.17 verdeutlicht weiterhin, dass der Roboter durch geschickte Ansteuerung der Motoren in der Lage ist, schneller als die üblichen Antriebssysteme hinter dem Ball zu kommen, um dann weiter zum Tor vorzudringen. Dies ist jedoch nur durch die Kombination von Ballsensorplattform und Antriebsystem möglich, denn der Roboter muss die Kontrolle des Balls immer aufrecht halten, um an ihm gegebenenfalls vorbei zu fahren.

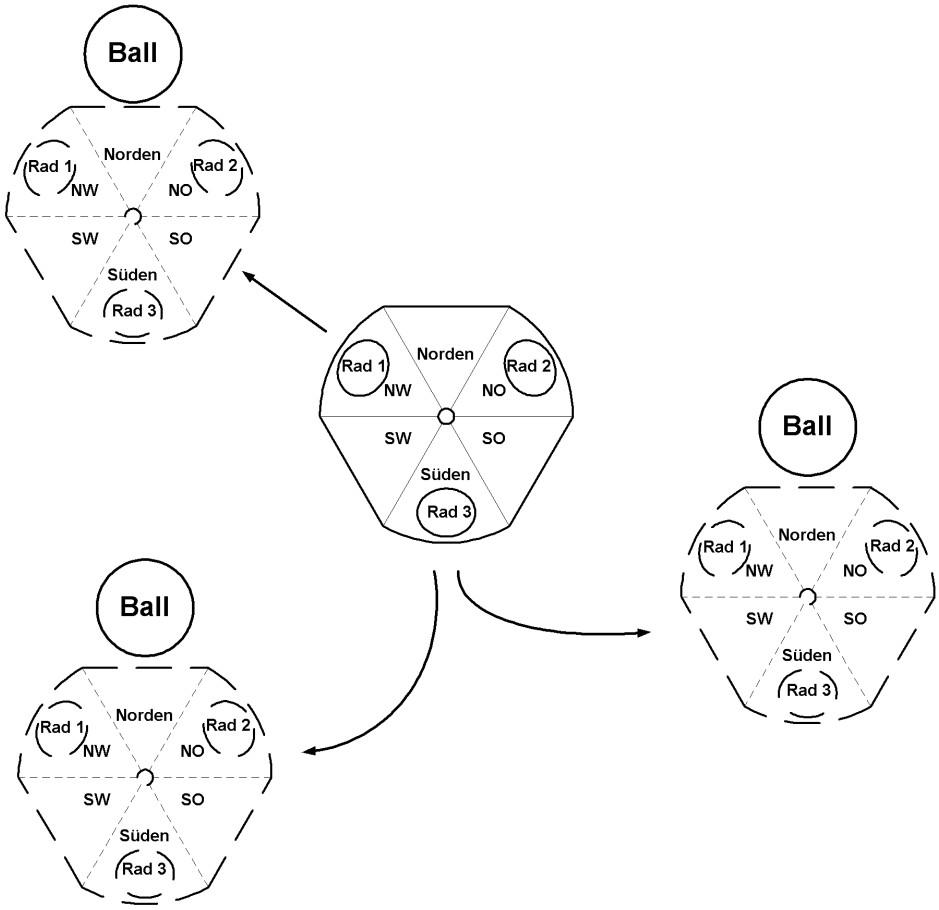


Abbildung 8.17.: Mögliche Spielzüge: Schnell hinter dem Ball kommen

8.2.4. Programm-Strategien

Durch die Möglichkeit, dass sich die Sensoren, ob nun Ball- oder Umgebungssensoren, in alle Richtungen drehen können, liegt die Spielstrategie darin, den Ball so schnell wie möglich zu finden und dann immer unter Kontrolle zu halten. Dies geschieht, indem die Funktion für die Ballsensoren die Plattform zwischen $0°$ und $180°$ hin und her drehen lässt. Bei Erkennung des Balles korrigiert die Funktion die Stellung des Servo so weit, bis der Ball wieder aus dem Sensorenbereich austritt und der Scanner sofort wieder an die letzte gesehene Position des Balles zurückkehrt. Der Scanner scheint durch die “Wackelbewegung“ den Ball unter Kontrolle zu halten, auch wenn sich der Ball von der ursprünglichen Position fortbewegt. Da der Beacon-Empfänger für die Torerkennung starr am Roboter installiert wurde, ist es für die Strategie auch wichtig, die Kontrolle zum Tor aufrecht zu halten. Falls der Roboter den Ball verliert, sollte er sich zum eigenen Tor zurück orientieren und dann einen erneuten Angriff starten. Die Umgebungssensoren geben die Information, ob in der Richtung, in die der Roboter fahren möchte, Wände oder Gegner zu finden sind. Falls also der Roboter nach Südwesten fahren möchte, drehen sich die Umgebungssensoren nach Südwesten und kontrollieren die Umgebung. Dabei versucht der Roboter gegebenenfalls Hindernissen auszuweichen.

8.2.5. Architektur des Programms

In den Anfangzeit der Roboterprogrammierung herrschte allgemein die Ansicht, dass ein Kontroll-System für autonome, mobile Roboter im Wesentlichen aus 3 funktionellen Elementen bestehen sollte:

- Sensing System, welches Sensordaten verarbeitet und ein Modell der Welt erzeugt.
- Planning System, dass aus diesem Modell der Welt und einem bestimmten Ziel einen Plan für die weitere Vorgehensweise macht.
- Execution (Action) System, dass aus diesem Plan die Befehle für den Roboter generiert.

Dieses System wird allgemein als SPA-System (Sense-Plan-Act) (siehe Abbildung 8.18) bezeichnet. Dieses System des Planning und World-Modelling ist sehr komplex, zu rechenaufwendig und zu langsam angesichts einer sich dauernd verändernden Umwelt.

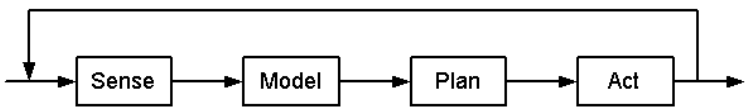


Abbildung 8.18.: Sense-Plan-Act - Architektur

Bei der Programmierung des Roboters dieser Diplomarbeit wurde auf die Programmier-Architektur von Rodney Brooks [Brooks'85] Subsumption-Architektur) (siehe Abbildung 8.19) zurück gegriffen, die er Mitte der achtziger Jahre am MIT (Massachusetts Institute of Technology) entwickelte. Bei dieser Art Architektur ist es möglich, verschiedene Verhaltensweisen zu kapseln. Dabei arbeiten verschiedene Schichten konkurrierend und asynchron an unterschiedlichen Aufgaben, wobei die Koordination über Prioritäten erfolgt. Das heißt, dass der Roboter nur ein Verhalten einer bestimmten Schicht abarbeitet. Dadurch ist es möglich, z.B. der Kollisionserkennung die höchste Priorität zu geben und anderen Aufgaben, die in diesem Moment eine weniger große Rolle spielen, eine niedrigere Priorität zuzuweisen. Theoretisch ist die Subsumption-Architektur so aufgebaut, dass die Schicht mit der höchsten Priorität ganz oben befindet und somit in der Lage ist, die anderen Schichten zu unterdrücken. Alle Schichten haben unabhängig davon, ob sie gerade die Kontrolle über den Roboter haben, Zugriff auf einen Pool von Sensordaten.

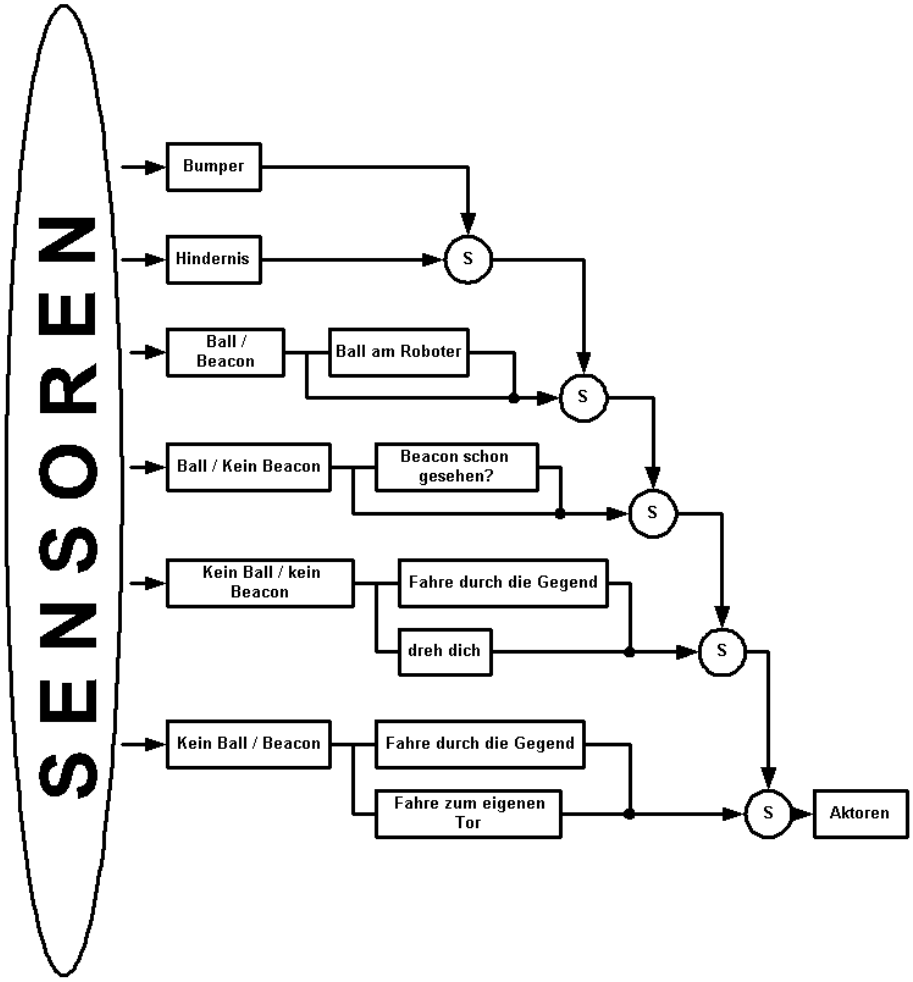


Abbildung 8.19.: Subsumption-Architektur

8.2.6. Allgemeines über die Programmierung des Roboters

Während der Programmierung des Roboters wurde versucht, einen PDA in das System zu integrieren. Dies erwies sich als nicht so revolutionär wie erwartet, da sich die serielle Schnittstelle zwischen dem Controllerboard (MIT-Board 6.270) und den PDA als sehr langsam erwies. Eine schnelle Verbindung ist jedoch notwendig, da der Roboter zeitspezifische Sensordaten an den PDA übermitteln, diese dann dort ausgewertet und dem Controllerboard die notwendigen Befehle wieder zu übermittelt werden muss. Deshalb wurde in dieser Arbeit der PDA nur soweit verwendet, um das Programm auf dem Controllerboard zu starten und einige Informationen, die keine zeitspezifischen Anforderungen benötigen, auf dem Display des PDAs darzustellen. Es gibt dennoch eine Version des Programms, die auf den Anschluss des PDA vollkommend verzichtet, um den knappen Speicher des Boards nicht vollkommen auszuschöpfen. In dieser Version ist auch ein Sensor- und Funktionstest integriert, um vor dem eigentlichem Fußballspiel alle Funktionen des Roboters kontrollieren zu können.

8.3. Ergebnis der Arbeit

In dieser Diplomarbeit wurde versucht, eine experimentelle Plattform für einen Roboter zu konstruieren, der in der Lange ist, an einem Roboterfußballspiel teilzunehmen und dabei die offiziellen Regeln der RoboCup Junior Soccer League einhält. Weiterhin sollte der Roboter einige Verbesserungen gegenüber existierenden Robotersystemen dieser League aufweisen. Diese gestellten Zielsetzungen wurden durch die Installation des “omnidirectional“ Antribssystems erfühlt, da der Roboter dadurch flexibler gegenüber den bisherigen Systemen ist. Durch den symmetrischen Aufbau der Plattform ist der Roboter nun in der Lage, nicht nur in alle möglichen Richtungen direkt zu fahren, sondern mit seinen Sensoren in alle möglichen Richtungen zu schauen. Es erwies sich bei Erkennung und Kontrolle des Balles, dass sich der Ballsensorscanner bestens bewert hat. Das sich die Umgebungssensoren in alle Richtung drehen lassen, erhöht die Flexibilität des Roboter um ein vielfaches. Die Plattform ist stabil und zum Teil in unabhängige Ebenen unterteilt, was einen schnellen Auf- und Umbau des Roboters ermöglicht. Das Koppeln eines PDAs mit dem Controllerboard (MIT-Board 6.270) ermöglicht nicht nur eine Kommunikation zwischen Robotern und/oder einem Kamera-Server, sondern die Auslagerung rechenintensiver Programmteile auf den PDA.

8.4. Ausblick

Während der Diplomarbeit wurde eine Idee/Vorstellung des Roboters entwickelt, welche jedoch aus zeitlichen und vor allem aus finanziellen Gründen nicht komplett umgesetzt werden konnte. Um den Roboter noch konkurrenzfähiger für die RoboCup Junior Soccer League zu gestalten, ist es sinnvoll, das Antriebssystem mit leistungsstärkeren Motoren und/oder einer größeren Untersetzung auszustatten. Die Geschwindigkeit, die diesem Prototyp zur Verfügung steht, ist nicht ausreichend. Es ist denkbar, eine bessere Kontrolle des genauen Bewegungsverlaufes mit Hilfe von Radencoder zu erhalten, um evtl. Unterschiede bei den

Motorenleistungen auszugleichen. Weiterhin könnte darüber nachgedacht werden, die Plattform mit leichteren Materialien zu konstruieren, um die Motoren zu entlasten. Zum Beispiel ist es sinnvoll, die drei Gewindestangen, die die Ebenen von einander trennen, aus leichterem Material zu fertigen und sie zu stückeln, damit man die Plattform evtl. leichter auf- bzw. umbauen kann.
Die Sensoren drehbar zu halten ist zwar eine ganz interessante Möglichkeit, die Umgebung im Auge zu halten, jedoch wäre es interessant, über den Einsatz einer “omnidirectional“ Kamera nachzudenken (Patrick Schmider: Einsatz einer “omnidirectional“ Kamera im RoboCup, 2002). Diese Kamera ist senkrecht nach oben auf einen kegelförmigen Spiegel ausgerichtet, welche dadurch das komplette Spielfeld unter Kontrolle hat. Um jedoch das Kamerabild auszuwerten, benötigt der Roboter ein leistungsstärkeres Controllerboard als das, was in der Diplomarbeit zur Verfügung stand.

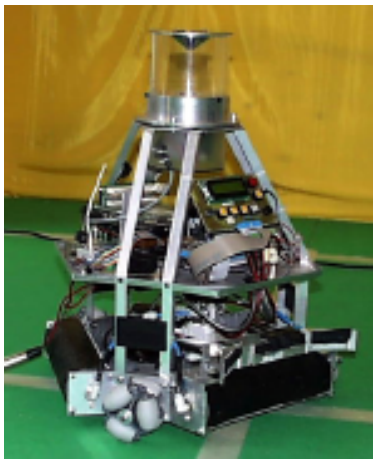


Abbildung 8.20.: Roboter mit omnidirektionaler Kamera

Es ist vorstellbar, falls bei weiteren Arbeiten mit dieser Plattform, die genügend Platz für neue Technologien bereitstellt, mehr finanzielle Mittel zur Verfügung stehen, nicht nur die Integration einer “omnidirectional“ Kamera in Betracht zu ziehen, sondern auch ein leistungsstärkeres Controllerboard wie zum Beispiel ein Mini-ITX oder Micro-ITX einzusetzen. Dadurch ergeben sich neue Möglichkeit bei der Programmierung. Denn bisher war die Programmierung auf eine funktionsorientierte Programmiersprache begrenzt, die nun durch eine objektorientierte Sprache ersetzt werden könnte.
Falls man sich weiterhin mit der bisherigen zur Verfügung stehenden Hardware beschäftigen möchte, dann sollte der Leser einen Blick auf Kapitel 6.3 der Diplomarbeit von Mirco Gerling (Gerling, 2003) richten, in dem man einige interessante Informationen erhält, wie man den Roboter mit Hilfe eines PDAs verbessern könnte.

Zum Schluss sollte der Leser noch einen kurzen Ausblick bekommen, was es für Möglichkeiten für den Einsatz eines “omnidirectional“ Antriebssystem außerhalb des Roboterfußballs gibt. Es ist denkbar, solche Antriebssysteme im Bereich der Lagerarbeiten einzusetzen, denn solche Systeme sparen Zeit und Platz, da keine Wendeaktionen durchgeführt werden muss. Weitere interessante Einsatzmöglichkeiten liegen im Bereich der Überwachung. Denn durch eine symmetrische Plattform ist zum Beispiel ein Überwachungsroboter in der Lage, in jede noch so enge Umgebung ohne größere Schwierigkeiten zu navigieren. Man könnte auch versuchen, dieses Antriebsystem Geländetauglich zu machen, dann könnten solche Roboterkonstruktionen im Bereich der militärischen Aufklärung oder bei Rettungsarbeiten in Katastrophengebieten eingesetzt werden. Eine weitere Überlegung wäre, solche Antriebssystem im Bereich der Fahrzeugtechnik zu integrieren. Dies würde das Einparken in manch enge Parklücke sehr erleichtern.

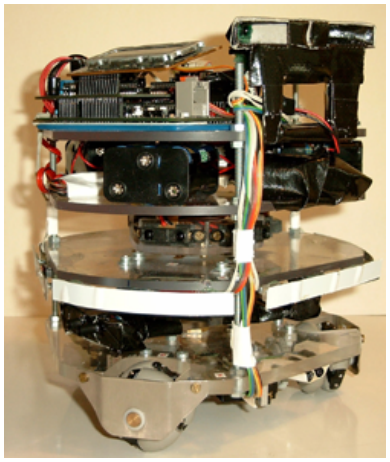
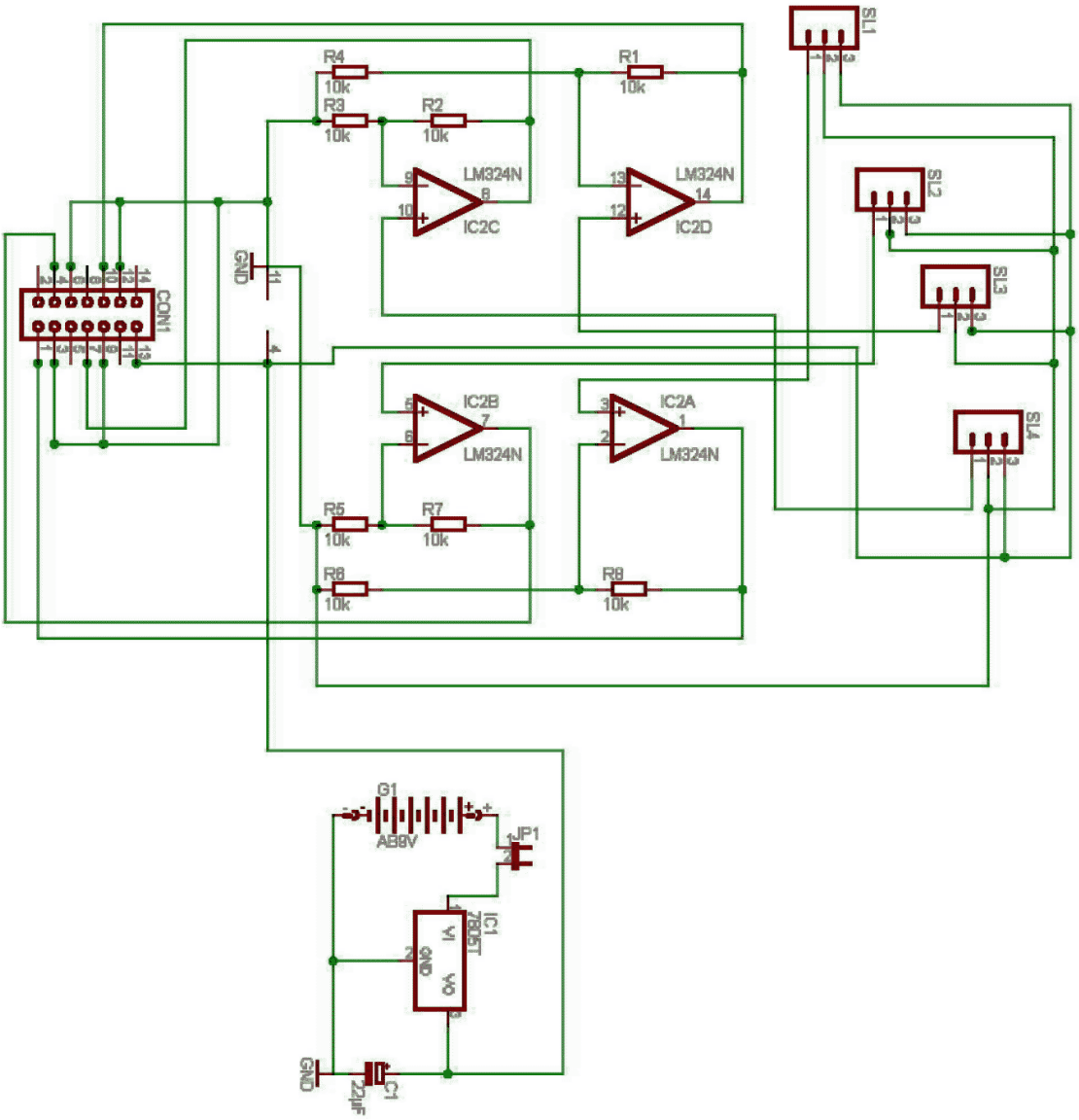


Abbildung 8.21.: Die Experimentelle Roboterplattform

A. Schaltplan Verstärker

Der Verstärker für die Sharp Sensoren mit vier Eingängen ist eine Weiterentwicklung von Gunther Lemm. Das Layout der Platine wurde von Timo Storjohann erstellt. Dieser Verstärker beruht auf einer Schaltung die im Roboter Labor erstmals von Rainer Balzerowski gebaut wurde. Informationen zu der Schaltung gibt es auf den Seiten der HAW Hamburg [HAWLego].



B. Datenblatt Sharp Sensor GP2D12

GP2D12/GP2D15

General Purpose Type Distance Measuring Sensors

■ Features

1. Less influence on the color of reflective objects, reflectivity
2. Line-up of distance output/distance judgement type
   Distance output type (analog voltage) : **GP2D12**
   Detecting distance : 10 to 80cm
   Distance judgement type : **GP2D15**
   Judgement distance : 24cm
   (Adjustable within the range of 10 to 80cm)
3. External control circuit is unnecessary
4. Low cost

■ Applications

1. TVs
2. Personal computers
3. Cars
4. Copiers

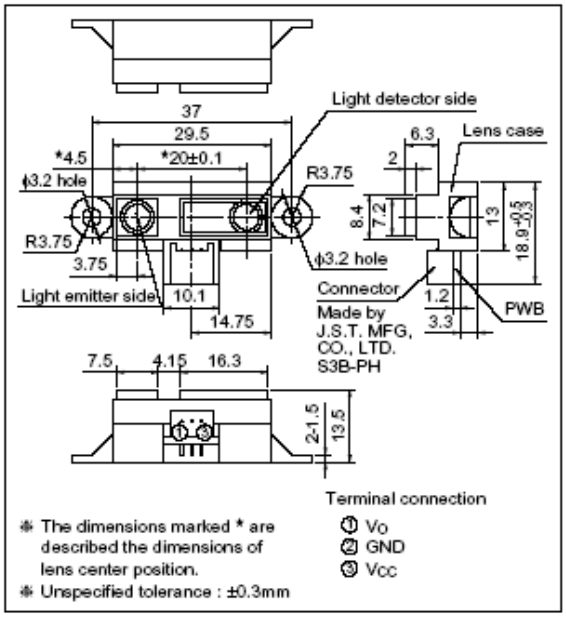
■ Absolute Maximum Ratings

(Ta=25°C, Vcc=5V)

Parameter	Symbol	Rating	Unit
Supply voltage	V_{CC}	−0.3 to +7	V
Output terminal voltage	V_O	−0.3 to Vcc +0.3	V
Operating temperature	T_{opr}	−10 to +60	°C
Storage temperature	T_{stg}	−40 to +70	°C

■ Outline Dimensions

(Unit : mm)



* The dimensions marked * are described the dimensions of lens center position.
* Unspecified tolerance : ±0.3mm

■ Recommended Operating Conditions

Parameter	Symbol	Rating	Unit
Operating supply voltage	V_{CC}	4.5 to +5.5	V

■ Electro-optical Characteristics

(Ta=25°C, V_{CC}=5V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Distance measuring range		ΔL	*1 *3	10	–	80	cm
Output terminal voltage	GP2D12	V_O	L=80cm *1	0.25	0.4	0.55	V
	GP2D15	V_{OH}	Output voltage at High *1	Vcc –0.3	–	–	V
		V_{OL}	Output voltage at Low *1	–	–	0.6	V
Difference of output voltage	GP2D12	ΔVo	Output change at L=80cm to 10cm *1	1.75	2.0	2.25	V
Distance characteristics of output	GP2D15	V_O	*1 *2 *4	21	24	27	cm
Average Dissipation current		I_{CC}	L=80cm *1	–	33	50	mA

Note) L : Distance to reflective object.

*1 Using reflective object : White paper (Made by Kodak Co. Ltd. gray cards R-27 · white face, reflective ratio ; 90%).
*2 We ship the device after the following adjustment : Output switching distance L=24cm±3cm must be measured by the sensor.
*3 Distance measuring range of the optical sensor system.
*4 Output switching has a hysteresis width. The distance specified by Vo should be the one with which the output L switches to the output H.

Fig.1 Internal Block Diagram

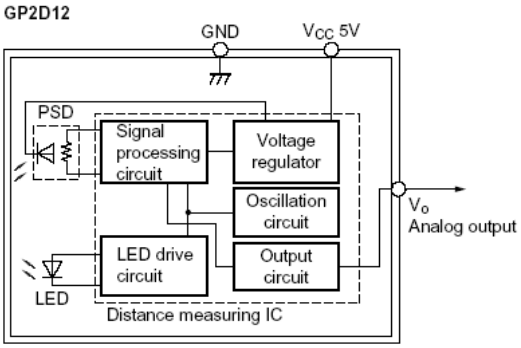


Fig.2 Internal Block Diagram

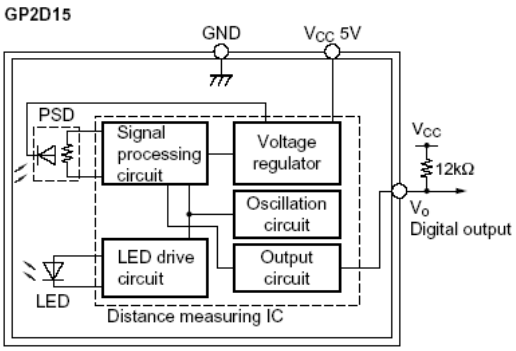


Fig.3 Timing Chart

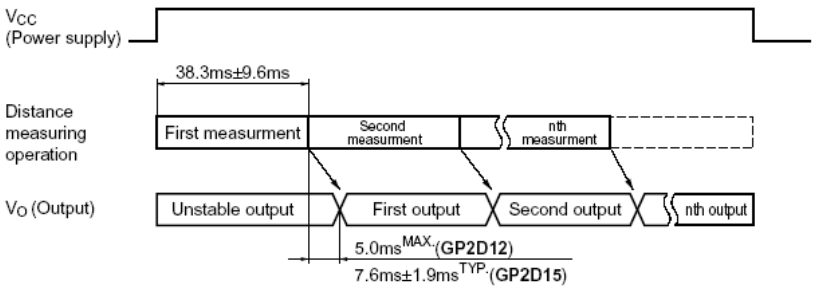


Fig.4 Distance Characteristics

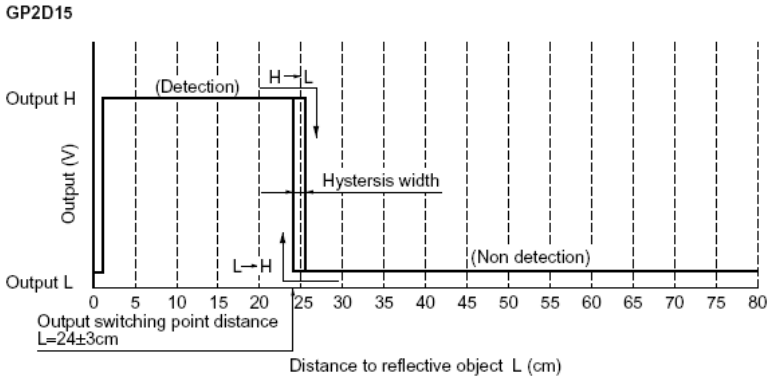


Fig.5 Analog Output Voltage vs. Surface Illuminance of Reflective Object

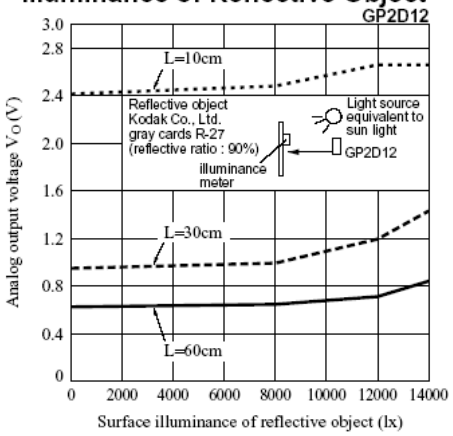


Fig.6 Analog Output Voltage vs.Distance to Reflective Object

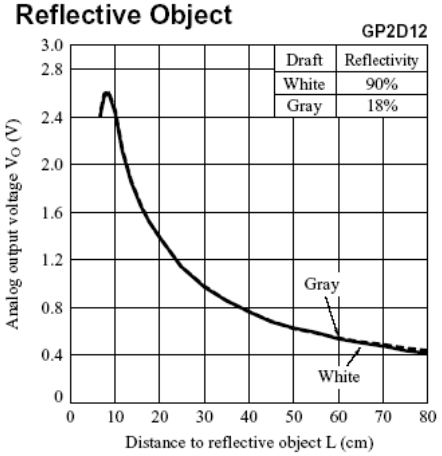


Fig.7 Analog Output Voltage vs.Ambient Temperature

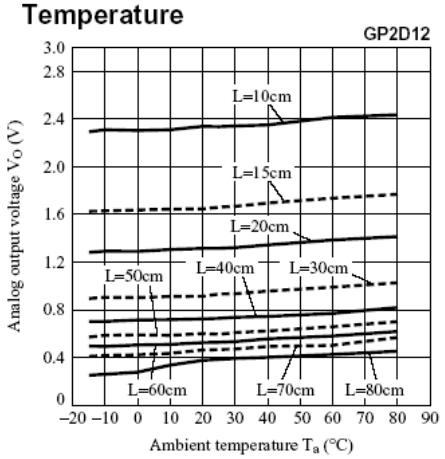
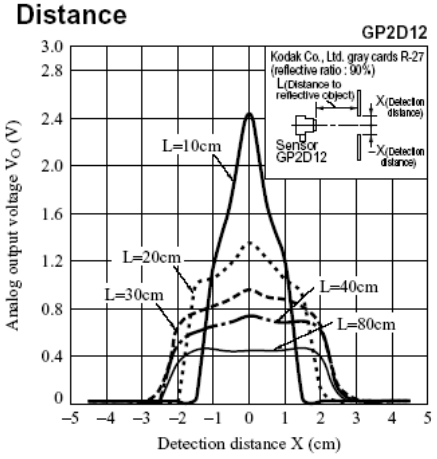


Fig.8 Analog Output Voltage vs.Detection Distance



Application Circuits

NOTICE

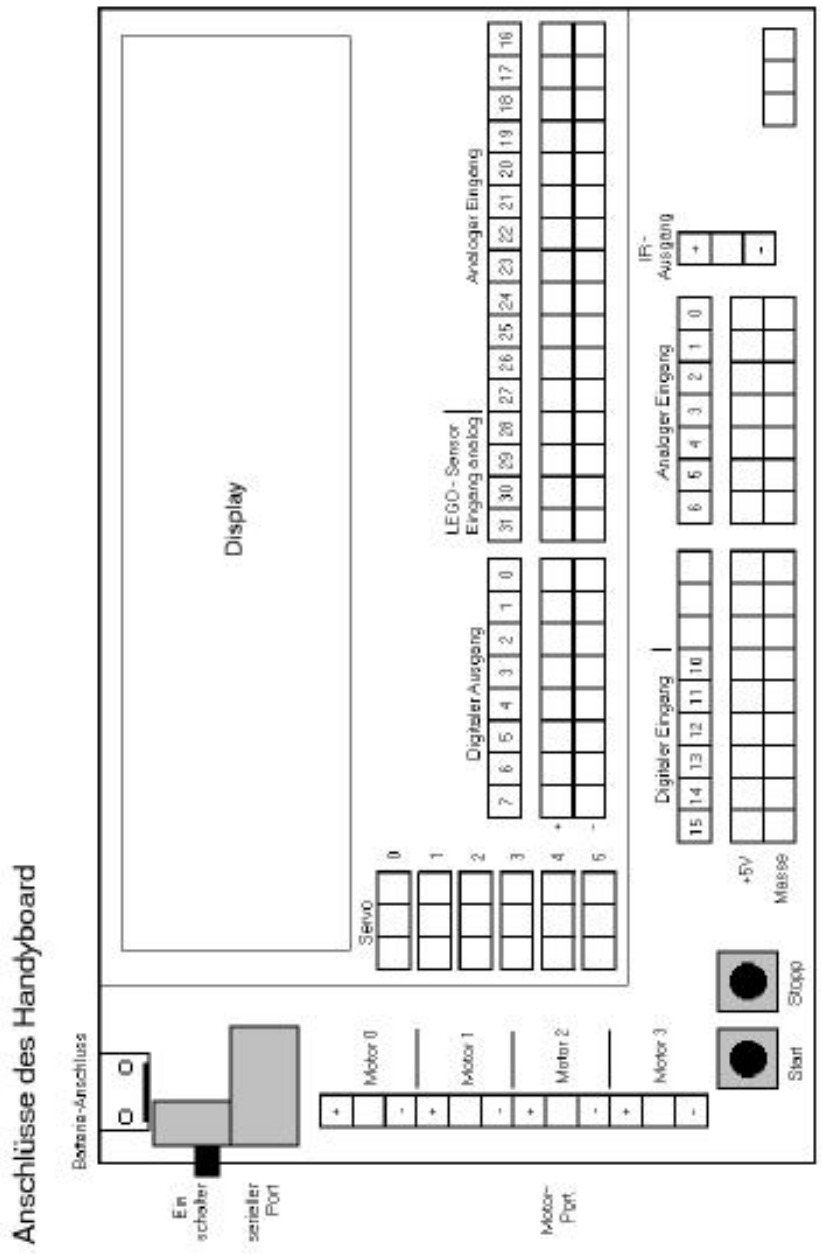
- The circuit application examples in this publication are provided to explain representative applications of SHARP devices and are not intended to guarantee any circuit design or license any intellectual property rights. SHARP takes no responsibility for any problems related to any intellectual property right of a third party resulting from the use of SHARP's devices.
- Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device. SHARP reserves the right to make changes in the specifications, characteristics, data, materials, structure, and other contents described herein at any time without notice in order to improve design or reliability. Manufacturing locations are also subject to change without notice.
- Observe the following points when using any devices in this publication. SHARP takes no responsibility for damage caused by improper use of the devices which does not meet the conditions and absolute maximum ratings to be used specified in the relevant specification sheet nor meet the following conditions:

  (i) The devices in this publication are designed for use in general electronic equipment designs such as:
  - Personal computers
  - Office automation equipment
  - Telecommunication equipment [terminal]
  - Test and measurement equipment
  - Industrial control
  - Audio visual equipment
  - Consumer electronics

  (ii) Measures such as fail-safe function and redundant design should be taken to ensure reliability and safety when SHARP devices are used for or in connection with equipment that requires higher reliability such as:
  - Transportation control and safety equipment (i.e., aircraft, trains, automobiles, etc.)
  - Traffic signals
  - Gas leakage sensor breakers
  - Alarm equipment
  - Various safety devices, etc.

  (iii) SHARP devices shall not be used for or in connection with equipment that requires an extremely high level of reliability and safety such as:
  - Space applications
  - Telecommunication equipment [trunk lines]
  - Nuclear power control equipment
  - Medical and other life support equipment (e.g., scuba).
- Contact a SHARP representative in advance when intending to use SHARP devices for any "specific" applications other than those recommended by SHARP or when it is unclear which category mentioned above controls the intended use.
- If the SHARP devices listed in this publication fall within the scope of strategic products described in the Foreign Exchange and Foreign Trade Control Law of Japan, it is necessary to obtain approval to export such SHARP devices.
-
- Contact and consult with a SHARP representative if there are any questions about the contents of this publication.

C. Handyboard



D. Ballsensoren der Firma Wiltronics

BELLARINE RoboBall™ SENSOR KIT (RO218) Instruction Sheet

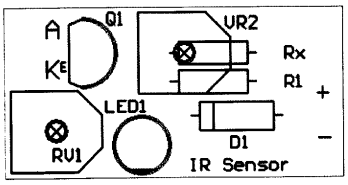
RO218DC1V1

The Bellarine RoboBall™ Sensor is designed to provide improved performance in locating the Infrared IR output of the Wiltronics RoboBall™. It is specifically designed for use with robots competing in the RoboCup Junior* competitions. The sensors also provide a visual output, when the sensors see the ball.

This sensor was designed and used successfully by Bill Corcoran, Josh Bennion, Geordie MacDonald and Andrew French, the winning team in RoboCup Junior* 2000.

The sensor is divided into 2 printed circuit boards (PCBs), a control PCB and a Sensor block PCB.

Fig.1 Control PCB Component Overlay



The Control PCB
Load and solder the parts into the control PCB referring to the component overlay, the component list and the circuit diagram. You will note the area on the PCB containing VR2, Rx and R1, this area is provided to assist with modifications and experimentation with the circuit. See section titled Modifications. The basic kit is supplied with resistor R1 only, Rx and VR2 do not need to be fitted. A LEGO® lead cut to 10cm should be fitted to the +ve and the –ve points marked on the PCB.
Note: LEGO® leads are not supplied in the kit.

Fig. 2 Sensor Block PCB Component Overlay

The Sensor Block PCB
Load and solder the QSD723 photo transistors on to the Sensor Block PCB, the long lead goes to the positive rail marked 'A' and the short lead goes to the negative rail marked 'K'. Connect the cable from the control PCB 'A' to 'A' and 'K' to 'K' soldering direct to the pads on the track side of the Sensor Block PCB. This will allow the QSD723's to be inserted into the holes of a single piece of LEGO®.

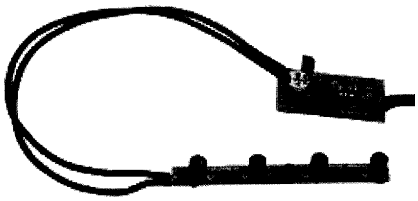


Fig.3 Assembled RoboBall™ Sensor Kit

Modifications
The array of QSD723's can be impacted by ambient light, it was determined that 4 pcs was the maximum that could be used in an array. If ambient light is a problem the array could be reduced to 3 pcs or extra shading could be used. A trimming potentiometer (trimpot) RV1 is provided to adjust the sensitivity. For those wishing to experiment further a resistor position Rx is provided so that an extra resistor can be added in parallel with R1 to reduce it's value. Alternately R1 could be replaced with a trimpot at VR2.

Make sure that the red LED is visible, this will assist with programming. It is surprising how often the sensor "sees" the ball, but the programming does not respond. This is usually due to the robot scanning too quickly.

Fig. 4 Component List

Control PCB

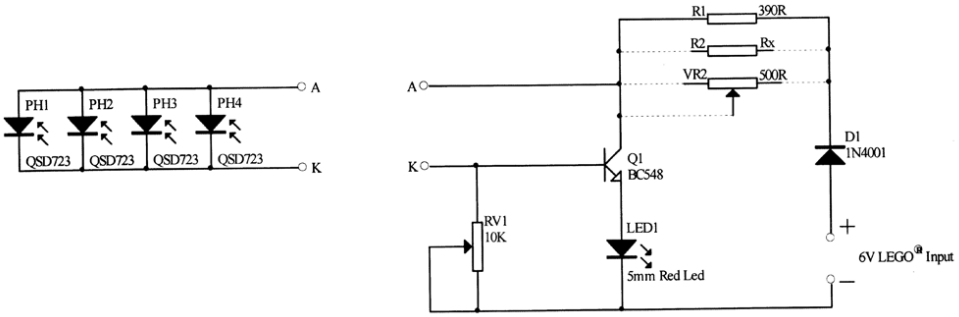
R1	RS1435 390R 0.25W 5% resistor	1
Rx	Not Fitted (? ohm 0.25W 5% Resistor)	
VR2	Not Fitted (500R TL1 Trimpot)	
RV1	PT8150 10K TL1 Trimpot	1
D1	1N4004 1 Amp Diode	1
LED1	LED-5MM/R 5mm Red Led	1
Q1	BC548 transistor	1
PCB1	213pc6v1 Printed Circuit Board	1
Lead	LEGO® lead, cut to 10cm (not supplied)	1

Sensor Block PCB

PH1-4	QSD723 Photo transistors	4
PCB2	213pc7v1 Printed Circuit Board	1
Lead1	12cm Red Hookup wire	1
Lead2	12cm Black Hookup wire	1

WILTRONICS RESEARCH PTY LTD PO Box 43, Alfredton 3350, AUSTRALIA
Ph. +61 3 5334 2513 Fax +61 3 5334 1845 http:/ www.wiltronics.com.au

Fig. 5 RO218 Bellarine RoboBall™ Sensor Circuit Diagram



WILTRONICS RESEARCH PTY LTD PO Box 43, Alfredton 3350, AUSTRALIA
Ph. +61 3 5334 2513 Fax +61 3 5334 1845 http:/ www.wiltronics.com.au

E. Konstruktionszeichnungen

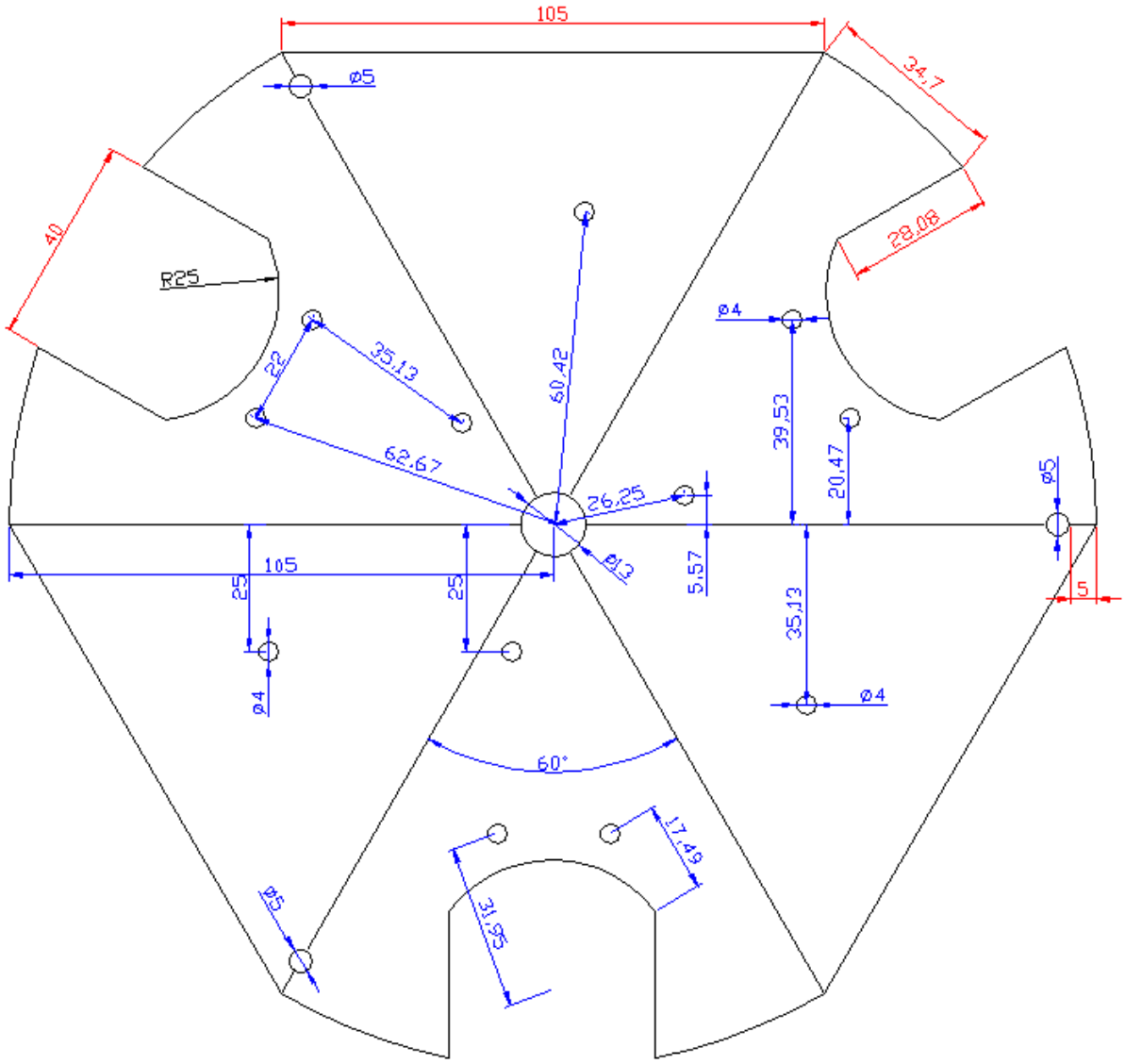


Abbildung E.1.: Antriebsebene

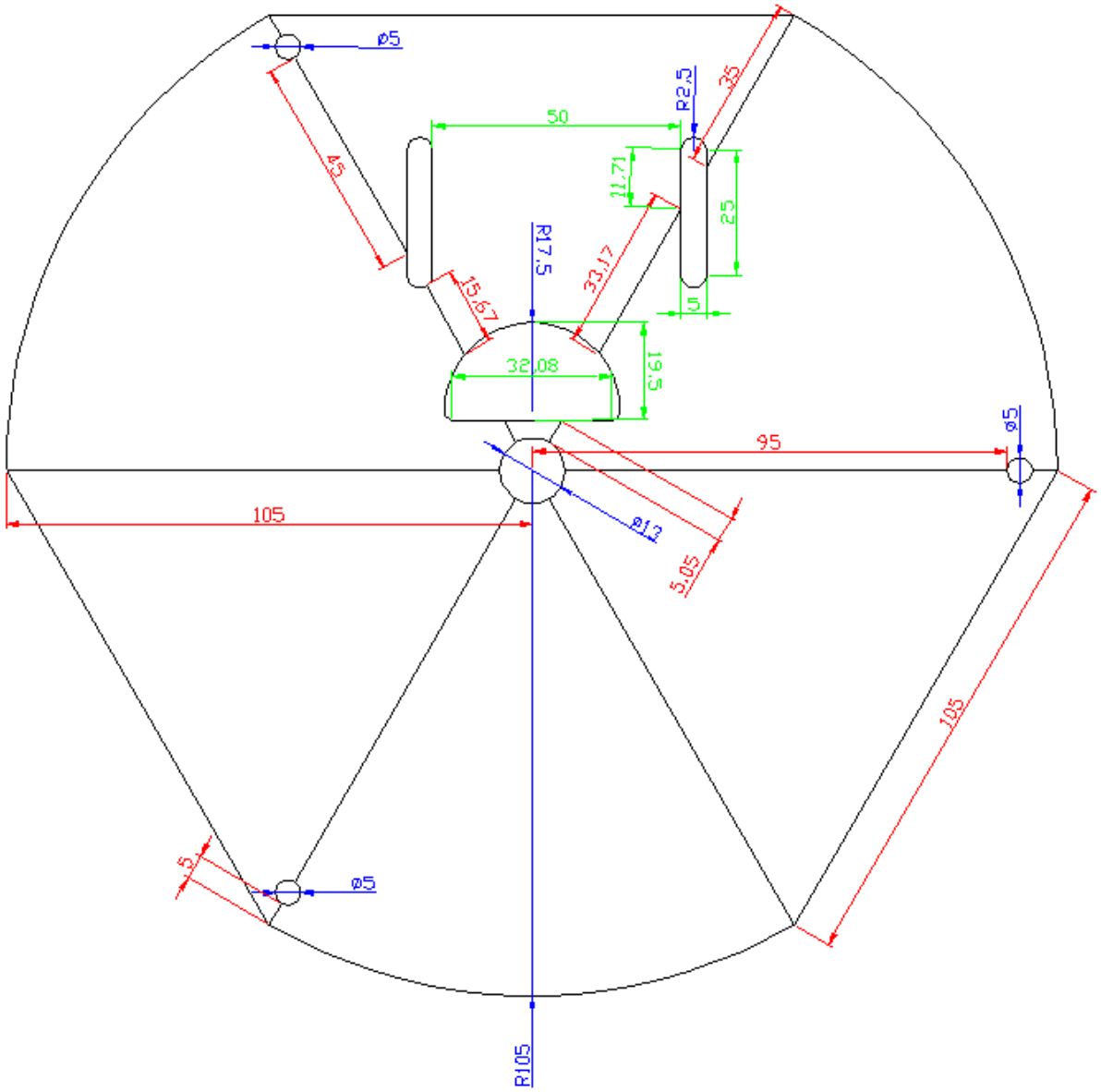


Abbildung E.2.: Ebene 2 bzw. Ebene 4

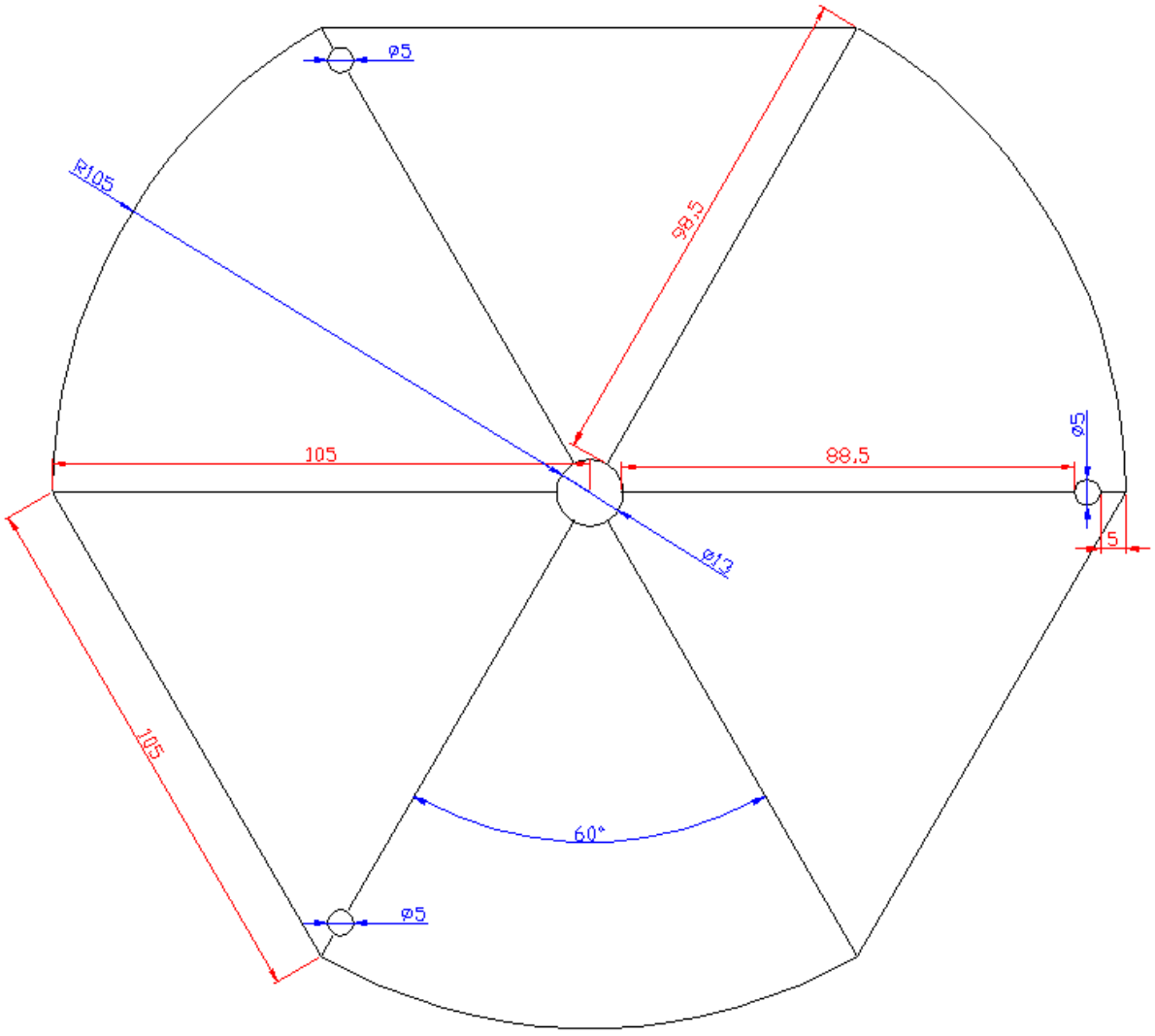


Abbildung E.3.: Ebene 3

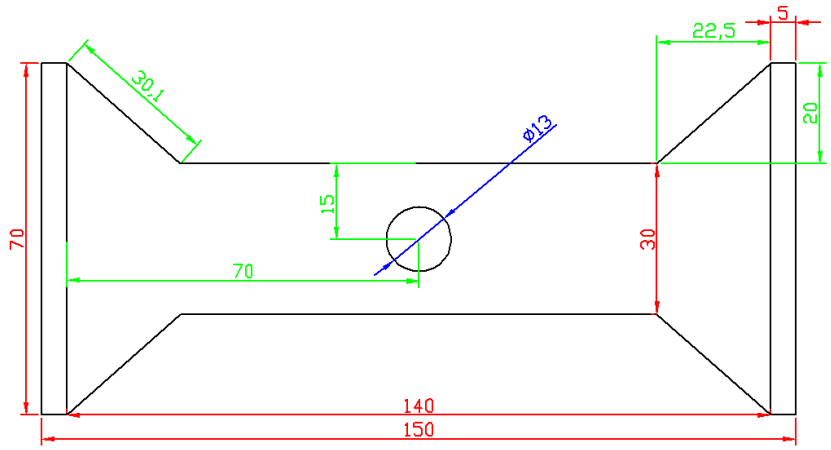


Abbildung E.4.: Ball-Sensor-Ebene

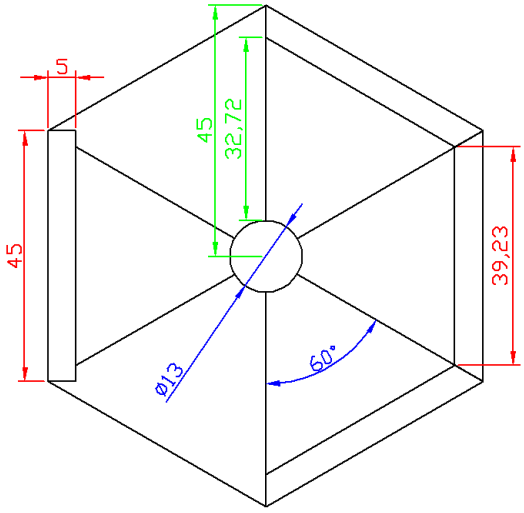


Abbildung E.5.: Umgebung-Sensor-Ebene

Alle hier dargestellten Konstruktionszeichnungen wurden mit AutoCAD erstellt und sind in digitaler Form auf der begleitenden CD zu finden.

F. Aufbau eines Servo

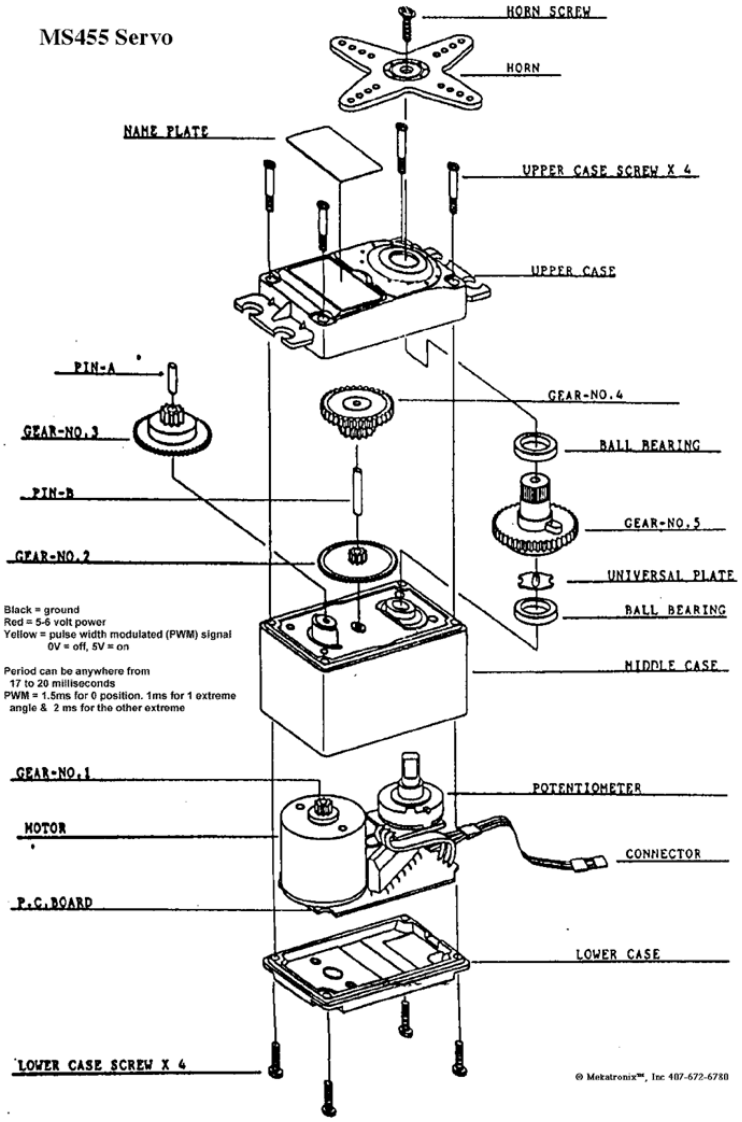


Abbildung F.1.: Servoschema von Mr. Robot [MrR]

G. Inhalt der CD-Rom

Die CD-ROM zu dieser Diplomarbeit enthält diese Arbeit als Datei *Diplomarbeit.pdf*. Außerdem findet man hier Dokumentationen über das Achsen- und MIT-Board sowie von einigen Sensoren, diverse Bilder, Videos und Konstruktionszeichnungen, verschiedene Regel für die einzelnen Disziplinen der beiden Initiativen (RoboCup, FIRA), diverse Tools und das Programm für den Roboter.

Hier der Inhalt der Verzeichnisse im einzelnen:

\Diplomarbeit
Das Verzeichnis enthält diese Diplomarbeit als Datei *Diplomarbeit.pdf*.

\Documentation\Achsen
Hier findet man das Benutzerhandbuch vom AKSEN-Board als Datei *handbuch.pdf* und diverse Datenblätter als PDF-Dateien.

\Documentation\MIT
In diesem Verzeichnis findet man The 6.270 Robot Builders Guide *6270-guide-92.pdf*, M68HC11E Family Technical Data *M68hc11e.pdf* und Replacing 68HC11A Series MCUs with 68HC11E Series MCUs *Eb193.pdf*.

\Documentation\SenInf
Datenblätter über den Reflexkoppler CNY70 *CNY70.pdf*, über die Sharp-Sensoren GP2D12/GP2D15 *SharpGP2D12-15.pdf*, über die Subminiatur-Schalter *Subminiatur-Schalter.pdf* und über das Ultraschall-Modul SRF04 von der Firma Devantech *Ultraschall - SRF04.pdf* sind hier zu finden.

\Pictures\DiplomBilder
Das Verzeichnis beinhaltet diverse Bilder (PNG-Format), die in dieser Diplomarbeit Verwendung finden.

\Pictures\KonstZeichnung\AutoCAD
Hier findet man detailierte Konstruktionszeichnungen die mit AutoCAD erstellt wurden. Dabei sind Dateinen mit der Endung *.DWF* und *.DWG* sowie Abbildungen mit *.PNG* zu finden.

\Pictures\KonstZeichnung\Handzeichnungen
In diesem Verzeichnis sind handgezeichnete Skizzen im PNG-Format zu finden.

\Pictures\KonstZeichnung\Visio
Konstruktionszeichnungen und Abbildungen für die Diplomarbeit erstellt mit VISIO sowie die dazu gehörigen PNG-Bilder sind hier zu finden.

\Pictures\RoboterBilder
Detaillierte Bilder (PNG-Format) über den Roboter bei der Konstruktionsphase sind hier zu finden.

\Pictures\Verschiedenes
In diesem Verzeichnis sind einige Bilder (PNG-Format) enthalten, die in der Diplomarbeit keine Verwendung gefunden haben.

\Pictures\Video
Das Verzeichnis beinhaltet 2 Videos, die sich mit der RoboCup-Junior-Soccer-League beschäftigen.

\RoboProgram
Enthält den Quellcode für den Roboter.

\Rules\Fira
Hier findet man von der FIRA-Initiative offizielle, erstellte Regeln der HuroSot, KheperaSot, MiroSot, NaroSot, RoboSot und SimuroSot.

\Rules\RoboCup
In diesem Verzeichnis findet man von der RoboCup-Initiative offizielle, erstellte Regeln der Middle-Size-, Rescue-, RoboCup-Junior-, Small-Size- und Sony-Leeged-Robot-League.

\Tools\InteractiveC
Hier ist das Interactive C - Programm für das Programmieren des Roboters mit dem MIT-Board 6.270 zu finden. *Achtung*! Dieses Programm beinhaltet nicht die benötigte Key-Datei.

Literaturverzeichnis

[AIBO] Sony Entertainment Rot Europe: **AIBO Europa**, Offiziele Webseite *URL:* *www.aibo-europe.com/index.asp?language=de*

[Aksen] I. Boersch: **KI-Labor der FH Brandenburg**, *URL:* *http://ots.fh-brandenburg.de/aksen*

[AM'04] **Official Homepage of Alan Mackworth**, University of British Columbia, 2004, *URL:* *www.cs.ubc.ca/spider/mack/*

[ASI'04] **Frauenhofer - Institut Autonome Intelligente Systeme**, 2004 *URL:* *www.ais.fraunhofer.de/*

[Brooks'85] Rodney A. Brooks: **A Robust Layered Control System for a Mobile Robot**, 1985, *URL:* *www.ai.mit.edu/people/brooks/publications.shtml*

[Cherry] Cherry Corporation: **Cherry DB Series Sub-Miniature Snape-Action Switch**, 2001, *URL:* *www.cherrycorp.com/english/switches/submini/db.htm*

[CNY70] Vishay: **Reflective Optical Sensor with Transistor Output**, *URL:* *www.vishay.com/docs/83751/83751.pdf*

[CS-F'04] Institut für Informatik: **CS Freiburg**, Albert-Ludwigs-Universität Freiburg *URL:* *www.cs-freiburg.de/*

[DB'97] IBM: **Kasparov vs Deep Blue**, 1997, *URL:* *www.research.ibm.com/deepblue/*

[DCGL'03] D.Cordes, G.Lemm: **Entwicklung einer mikroprozessorbasierten Sensor-/Aktorerweiterung für das LEGO-Mindstorm System**, Studienarbeit 2003, *URL:* *http://users.informatik.haw-hamburg.de/~kvl/cordes/diplom.pdf, 2004*

[DCGL'04] Dietmar Cordes, Gunther Lemm: **Entwicklung einer mikroprozessorbasierten Sensor- / Aktorerweiterung für das LEGO-Mindstorm System**, Diplomarbeit 2004, *URL:* *http://users.informatik.haw-hamburg.de/~kvl/cordes/diplom.pdf*

[EW'99] Edwin Wise: **Applied Robotics**,Prompt Publications, 1999, ISBN 0-7906-1184-8

[eXtrP'02] Martin Lippert, Stefan Roock, Henning Wolf: **Software entwickeln mit eXtreme Programming**, dpunkt.verlag, Januar 2002, ISBN 3-89864-107-4

[FIRA'04] **FIRA - Federation of International Robot-soccer Association**, *URL:* http://fira.net/

[GP2D12] SHARP: **GP2D12/GP2D15**, *URL:* www.acroname.com/robotics/parts/SharpGP2D12-15.pdf

[Handy] **The Handy Board**, *URL:* http://handyboard.com/

[HP'04] **Hewlett-Packard**, 2004, *URL:* www.hp.com/

[InterC] Newton Research Labs: **Interactive C**, *URL:* www.newtonlabs.com/ic/

[Interroll] **Interroll - Rollen, Antriebe und Module für die Fördertechnik**, *URL:* www.interroll.de/

[JJAF'96] Joseph L. Jones, Anita M. Flynn: **Mobile Roboter**,Addison- Wesley, Bonn, Germany, 1996

[KC'20] Wikipedia, the free encyclopedia: **Karel Capek**, *URL:* http://en.wikipedia.org/wiki/Karel_Capek

[Klemke'04] **Gunter Klemke**, Hochschule für Angew. Wissenschaften Hamburg, Fachbereich Elektrotechnik/Informatik, *URL:* http://users.informatik.haw-hamburg.de/~gk/

[KMZ52] Philips Semiconductors: **Electronic Compass Design using KMZ51 and KMZ52**, 2004, *URL:* www.semiconductors.philips.com/

[LEGO] **LEGO MINDSTORMS**, *URL:* http://mindstorms.lego.com/eng/default.asp

[Lepomux-Homepage] Dietmar Cordes, Gunther Lemm: **Official Homepage of Lepomux**, *URL:* www.lepomux.com/

[Luck'04] Kai von Luck: **IKS-project (Integration Kognitiver Systeme)**, *URL:* http://users.informatik.haw-hamburg.de/~kvl/

[MG'02] Mirco Gerlin: **Agentenarchitekturen**, Studienarbeit 2002, *URL:* http://users.informatik.haw-hamburg.de/~kvl/gerling/studien.pdf

[MG'03] Mirco Gerling: **PDA-gestützte Robotersteuerung mit funkbasierter Serveranbindung**, Diplomarbeit 2003, *URL:* http://users.informatik.haw-hamburg.de/~kvl/gerling/diplom.pdf

[Mini-ITX] Mini-ITX.com and respective owners: **mini-itx.com - The Next Small Thing**, *URL:* www.mini-itx.com/

[MIT'04] Massachusetts Institute of Technology: **6.270 - MIT's Autonomous Robot Design Competition**, 2004, *URL:* www.mit.edu:8001/courses/6.270/home.html

[MM'03] Michael Manger: **Design und Realisierung von "kostengünstigen" fussballspielenden Robotern**, Studienarbeit 2003 *URL: http://users.informatik.haw-hamburg.de/~kvl/manger/studienarbeit.pdf*

[MpocketPC] **Microsoft Corporation**, 2004 *URL: www.microsoft.com/*

[MrR] Mr. Robot: **Robots & Microcontrollers for the Future**, *URL: www.mrrobot.com/*

[MS'04] Michael Schröer: **Physik-Lexikon**, 2004 *URL: www.physik-lexikon.de/index.php*

[NASA'04] Jim Wilson: **National Aeronautics And Space Administration (NASA)**, 2004, *URL: www.nasa.gov/home/*

[NIST'04] **National Institute of Standards and Technology**, *URL: www.nist.gov/*

[RC'04] The RoboCup Federation: **RoboCup Official Site**, *URL: www.robocup.org/*

[RCJ'04] RoboCup Technical Committee: **RoboCcup Junior Official Site**, *URL: http://satchmo.cs.columbia.edu/rcj/*

[RCS4L'04] **RoboCup 2004 - Sony Four-Legged Robot League Lisbon, Portugal**, *URL: www.openr.org/robocup/index.html*

[Servos] ModellFlugKlub Pattensen: **Servos verschiedener Hersteller**, *URL: www.mfk-pattensen.de/servos.htm*

[SHARP] **SHARP Corporation**, *URL: http://sharp-world.com/index.html*

[SRF04] Devantech: **Ultraschall-Modul SRF04**, *URL: www.roboter-teile.de/datasheets/srf04.pdf URL: www.robot-electronics.co.uk/*

[TR'02] Thomas Röfer: **Motorik**, Universität Bremen, 2002, *URL: www.informatik.uni-bremen.de/~roefer/kr00/03s.pdf*

[Werkstatt] HAW-Hamburg: **Zentrale Laborwerkstatt**, *URL: www.haw-hamburg.de/mp/zw/*

[WR'70] W.W.Royce: **Managing the Development of Large Software Systems**, Proceedings of IEEE WESCON, August 1970

[WRPL'03] AV Interactive: **Wiltronics Research Pty Ltd**, 2003, *URL: www.wiltronics.com.au/*

Weitere Links

[AC'02] Adam Currie: **The History of Robotics**, University of California, 2002 *URL:* *http://cache.ucr.edu/~currie/roboadam.htm*

[AG'] Alexander Gloye: **Projektgruppe RoboCup - FU Berlin**, *URL:* *http://robocup.mi.fu-berlin.de/*

[Anhalt] Hochschule Anhalt (FH): **RoboCup-Ligen**, *URL:* *www.et.hs-anhalt.de/robocup/ligen.html*

[AZ'04] Andreas Zell: **GI Gesellschaft für Informatik - Lexikon: Roboter-Fußball**, 2004 *URL:* *www.gi-ev.de/informatik/lexikon/inf-lex-roboter-fussball.shtml*

[Droids] Universität Dortmund: **Dortmund Droids**, *URL:* *www.robosoccer.de/*

[DS'99] Dirk Stüker: **Projektstudie zum Aufbau autonomer Roboter**, Studienarbeit 1999, *URL:* *www.informatik.uni-oldenburg.de/~trebla/Harold/StA_html/StA.html*

[Fulda] Fachbereich Angewandte Informatik Fulda: **Leagues**, *URL:* *www.informatik.fh-fulda.de/unimatrix/league.html*

[GRCR'01] Gerhard Röthlin, Christoph Rüegg: **Der Bau und die Programmierung eines omnidirektionalen Roboters**, Maturaarbeit 2001 *URL:* *http://cdrnet.ch/projects/xbot/doc.pdf*

[IRSI] International Rescue System Institute: **RoboCup-Rescue Official Web Page**, *URL:* *www.rescuesystem.org/robocuprescue/*

[ISD] Asada Laboratory: **Intelligent Systems Division - Performance Metrics and Test Arenas for Autonomous Mobile Robots**, *URL:* *www.isd.mel.nist.gov/projects/USAR/competitions.htm*

[Koch'03] Birgit Koch: **Einsatz von Robotikbaukästen in der universitären Informatikausbildung am Fallbeispiel "Hamburger Robocup: Mobile autonome Roboter spielen Fußball"**, Diplomarbeit 2003, *URL:* *ftp://ftp.informatik.uni-hamburg.de/pub/unihh/informatik/TIS/koch/da.zip*

[OD'03] Oliver Dahlmann: **Entwicklung eines kostengünstigen Distanzmesssystems**, Diplomarbeit 2003, *URL:* *http://users.informatik.haw-hamburg.de/~kvl/dahlmann/diplom.pdf*

[PS'02] Patrick Schmider: **Einsatz einer omnidirektionalen Kamera im RoboCup**, 2002 *URL:* *www.informatik.uni-stuttgart.de/ipvr/bv/lehre/HS_robocup/hs_ss2002_schmider.pdf*

[RCMSL'04] RoboCup-MSL **RoboCup2004 Middle Size League**, *URL:* *www.er.ams.eng.osaka-u.ac.jp/rc2004msl/index.cgi*

[RCPor'04] **Official Page Of RoboCup 2004 - Portugal**, *URL:* *www.robocup2004.pt/*

[RCSSL'04] Brett Browing: **RoboCup Small Size Leage**, *URL:* *www-2.cs.cmu.edu/~brettb/robocup/*

[SJL'00] Elizabeth I. Sklar, Jeffrey H. Johnson, Henrik Hautop Lund: **Children Learning From Team Robotics - RoboCup Junior 2000**, *URL:* *http://demo.cs.brandeis.edu/papers/rcj2000.pdf*

[TR] Thomas Röfer: **Motorik**, *URL:* *www.informatik.uni-bremen.de/~roefer/kr00/03.pdf*

[Unimatrix] Marc Siewert, Marcus Feßler: **Unimatrix Fachhochschule Fulda**, *URL:* *http://unimatrix.it-spirit.net/*

[Wien] Institutes für Handhabungsgeräte und Robotertechnik (IHRT), Technischen Universität Wien (TU Wien): **Roboterfußball-Homepage**, *URL:* *www.roboterfussball.at/info/frameset-deu.html*

Verweise auf Regeln der RoboCup Leagues

[RC-EL'04] Gaurav Singal: **RoboCup-ELeague**, *URL:* http://agents.cs.columbia.edu/eleague/

[RC-HL'04] RoboCup Technical Committee: **RoboCup 2004 Humanoid League**, *URL:* www.ais.fraunhofer.de/robocup/HL2004/rules.html

[RC-JD'04] RoboCup Technical Committee: **RoboCupJunior 2004 DANCE rules**, *URL:* http://satchmo.cs.columbia.edu/rcj/rcj2004/dance-rules.html

[RC-JR'04] RoboCup Technical Committee: **RoboCupJunior 2004 RESCUE Rules**, *URL:* http://satchmo.cs.columbia.edu/rcj/rcj2004/rescue-rules.html

[RC-JS'04] RoboCup Technical Committee: **RoboCupJunior 2004 SOCCER rules**, *URL:* http://satchmo.cs.columbia.edu/rcj/rcj2004/soccer-rules.html

[RC-MSL'04] MSL Technical Committee 1997-2004: **Middle Size Robot League Rules and Regulations for 2004**, *URL:* www.tcsi.de/ROBOCUP/_DOCUMENTS/MSL/msl-rules-2004.pdf

[RC-R'04] RoboCup Technical Committee - Adam Jacoff, Brian Weiss: **RoboCupRescue Robot League Rules**, *URL:* www.isd.mel.nist.gov/RoboCup2003/RoboCupRescue2003Rules.pdf

[RC-RSim'04] RoboCup Technical Committee: **RoboCup 2004 Rescue Simulation League Rules V1.01**, *URL:* http://robot.cmpe.boun.edu.tr/rescue2004/rules2004.pdf

[RC-S4L'04] RoboCup Technical Committee: **Sony Four Legged Robot Football League Rule Book**, *URL:* www.tzi.de/~roefer/Rules2004/Rules2004.pdf

[RC-SSL'04] Gordon Wyeth: **Laws of the F180 League 2004 - Release 3.00a**, *URL:* www.itee.uq.edu.au/~wyeth/F180%20Rules/f180rules300a.pdf

[RC-SSLField'04] Gordon Wyeth: **Drawing of the Field for the F180 League 2004**, *URL:* www.itee.uq.edu.au/~wyeth/F180%20Rules/f180_2004_field.pdf

Alle in diesem Literaturverzeichnis angegebenen Links auf Webseiten sind am 08.05.2004 auf ihre Funktionfähigkeit überprüft worden.

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Ort, Datum	Unterschrift des Studenten