

# Roboterfußball

## Ein studentisches Projekt im Studiengang Technische Informatik

Prof. Dr. Oliver Bittel,  
Fachbereich Informatik  
Fachhochschule Konstanz  
bittel@fh-konstanz.de

### Einleitung

In diesem Jahr fand in Portugal nicht nur die Fußball-Europameisterschaft statt, sondern es kämpften dort auch Fußball-Roboterteams aus aller Welt gegeneinander. Die nun seit 1997 jährlich stattfindende Weltmeisterschaft im Roboterfußball [RoboCup] hat eine enorme Anziehungskraft sowohl im akademischen als auch im industriellen Forschungsumfeld. Damit die Roboter beispielsweise in der *Middle Size League* (maximal 6 Spieler pro Mannschaft mit einer Spielfeldgröße von 8 auf 10m) überhaupt in der Lage sind, den Ball zu dribbeln, zu schießen, zu blocken oder gar zu passen, sind zahlreiche Probleme in der Sensorverarbeitung, Navigation und Kooperation mobiler Roboter zu lösen. Inzwischen gibt es auch eine Liga für humanoide Roboter. Das für 2050 ausgegebene Ziel, gegen den amtierenden Weltmeister (echte Spieler!) zu bestehen, mutet aufgrund der Tatsache, dass die humanoiden Roboter gerade erst das Laufen gelernt haben, doch etwas utopisch an.

Der Studiengang Technische Informatik setzt bereits seit mehreren Jahren mobile Roboter in verschiedenen Projekten ein. Es war daher nahe liegend, sich auch einmal mit Roboterfußball zu beschäftigen. Im Rahmen eines studentischen Projekts wurden zwei Fußball-Teams, bestehend aus jeweils 3 Spielern, realisiert. Studentische Projekte sind Bestandteil der Ausbildung in der Informatik im 7. und 8. Semester. Im Studiengang Technische Informatik mit dem Schwerpunkt Intelligente Automatisierungssysteme geht es dabei meistens um die Entwicklung realzeitfähiger Steuerungen von technischen Prozessen.

### Spieler und Fußballfeld

Aus Kostengründen kamen *Lego-Mindstorms*-Roboterbaukästen zum Einsatz. Zentraler Baustein ist der RCX [KnNo00], der aus einem Mikrocontroller besteht und über 3 Sensoreingänge und 3 Motorausgänge verfügt (Abb. 1). Der RCX kann mit einem Satz von Batterien autonom betrieben werden. Der Baukasten enthält 2 Berührungssensoren, einen Lichtsensor und 2 Motoren. Die Motoren werden typischerweise zum Antrieb von Rädern verwendet. Zahlreiche Bausteine erlauben die Konstruktion ganz unterschiedlicher Roboter. Nach einer Reihe von Experimenten hat sich das Modell aus Abbildung 1 durchgesetzt, das sich durch Wendigkeit und eine besonders robuste Bauweise auszeichnet. Alle Roboter sind identisch aufgebaut und bewegen sich auf dem in Abb. 2 gezeigten Spielfeld.

Der RCX kann beispielsweise mit den Programmiersprachen *Legos* (Teilmenge von C++) und *Lejos* (Teilmenge von Java) programmiert werden. Die Programme werden auf einem PC

entwickelt, cross-compiled und über eine Infrarotschnittstelle auf den RCX herunter geladen. Durch Tastendruck auf dem RCX lässt sich das Programm dann starten.



Abbildung 1: Lego Mindstorms Roboter. Der RCX enthält einen Mikrocontroller, mit dem die beiden Antriebsmotoren angesteuert werden.



Abbildung 2: Spielfeld mit 6 Lego-Mindstorms-Robotern. Das Spielfeld, das mit einer Bande versehen ist, misst die Größe 170 cm auf 110 cm. Gespielt wird mit einem Golfball.

Mit den im Baukasten vorhandenen Sensoren können nur einfache Verhalten wie Linienverfolgung oder durch Berührung ausgelöstes Ausweichen implementiert werden. Außerdem lässt der sehr begrenzte Programmspeicher des RCX (ca. 15 Kb) den Ablauf nur sehr kleiner Programme zu. Um diese Grenzen zu überwinden, wird das Spielgeschehen von einer an der Decke montierten Kamera erfasst. Die Position jedes Spielers und des Balles wird in regelmäßigen Abständen ermittelt. Die eigentliche Spielerintelligenz wird als Programm auf einem PC realisiert, das sich über die Position des eigenen und der anderen Spieler und über die Ballposition informieren kann und an den eigentlichen Lego-Roboter einfache Bewegungsbefehle über den Infrarot-Sender schickt, die der Roboter entgegen nimmt und nur auszuführen braucht.

## Architektur des Systems

Der Kern des Systems ist der Spiel-Server (Abb. 3). Mit seiner Bildverarbeitungskomponente holt er sich von einer Firewire-Kamera (IEEE 1394) 30-mal pro Sekunde ein Bild ab und wertet es in Echtzeit aus. Um Ort und Ausrichtung jedes Roboters ermitteln zu können, sind auf der Oberseite der Roboter eindeutige Farbmarkierungen angebracht. Eine graphische Benutzeroberfläche (Abb. 4) stellt die Ball- und Roboter-Positionen dar und gestattet außerdem einen Eingriff in das Spielgeschehen (z.B. Spiel anhalten bzw. fortsetzen).

Die Intelligenz jedes Spielers läuft als eigenes Programm (siehe S1 bis S6 in Abb. 3). Die Spielerprogramme werden 30-mal je Sekunde über die Positionen der Roboter und des Balls informiert. Aus diesen Informationen berechnen sie einfache Bewegungsbefehle (fahre gerade aus, drehe links, drehe rechts, stoppe, etc.) und geben diese an den Infrarot-Server weiter. Der Infrarot-Server sendet im Broadcast-Verfahren alle Befehle an die Lego-Roboter R1 bis R6. Jeder Lego-Roboter filtert die für ihn bestimmten Befehle aus und führt sie durch. Damit auch ein Zusammenspiel der Roboter miteinander möglich wird, dürfen die Spielerprogramme einander Nachrichten zuschicken.

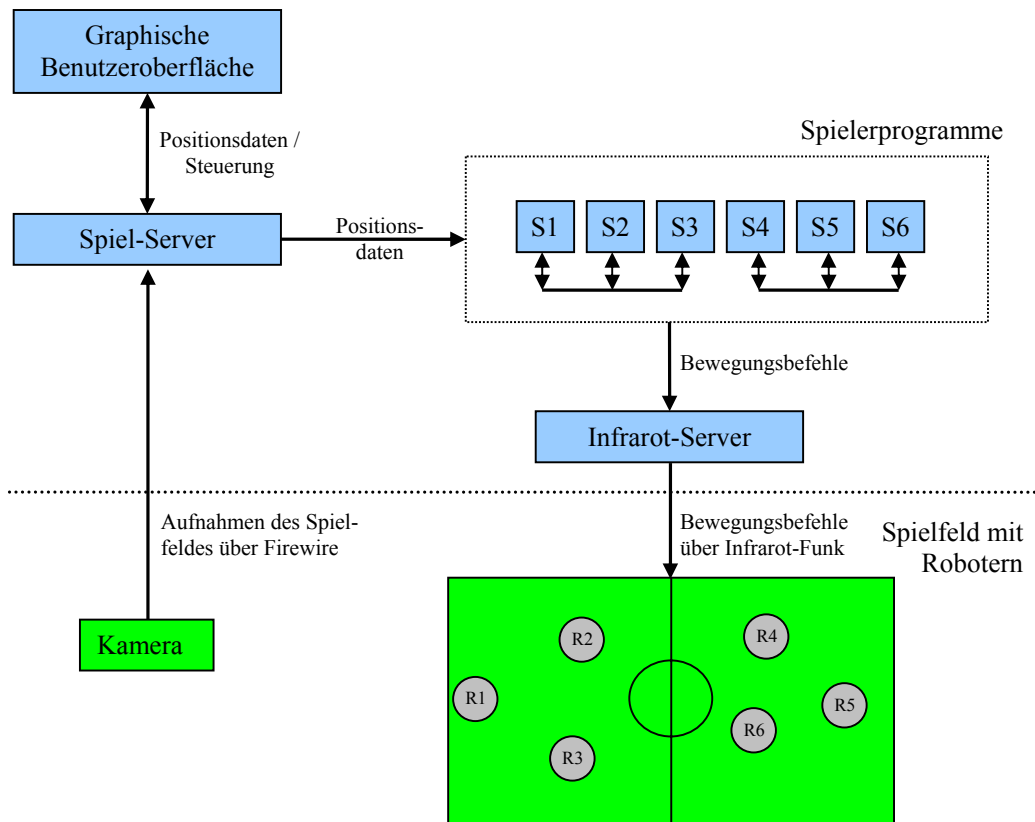


Abbildung 3: Architektur des Systems. Der Infrarot-Server läuft auf einem Windows-Rechner und die restlichen Komponenten auf einem Linux-Rechner. Die Komponenten kommunizieren untereinander über das verbindungsorientierte TCP- bzw. das paketorientierte UDP-Protokoll

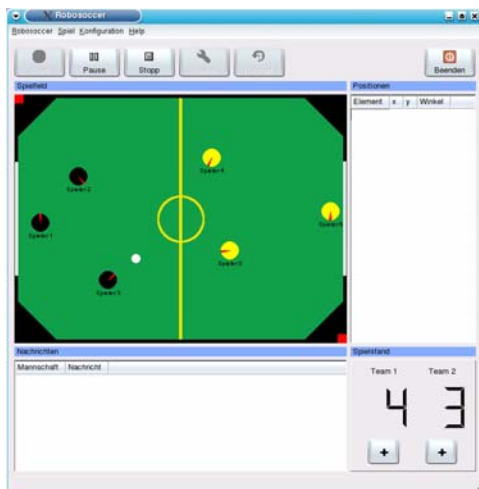


Abbildung 4: Graphische Benutzeroberfläche. Dargestellt sind die Roboter und der Ball aus der Spielsituation in Abb.2

gründen in Java implementiert. Für die Kommunikation auf PC-Seite werden die Protokolle TCP und UDP eingesetzt.

Die Spielerprogramme weisen den Robotern eine der folgenden Rollen zu: Torwart, Verteidiger oder Angreifer. Für jede Rolle wird auf Grundlage aller Spielerpositionen und der Ballposition ein Entscheidungsbaum durchlaufen und die nächste durchzuführende Aktion berechnet. Typische Aktionen sind: Ball schießen, Ball dribbeln, Schusslinie abblocken und Befreiungsschuss.

Das System läuft auf einer heterogenen Plattform und ist mit unterschiedlichen Sprachen implementiert. Der in Java realisierte Infrarot-Server basiert auf einer Fremdentwicklung und läuft unter Windows. Die Bildverarbeitung basiert auf das Bildverarbeitungsprogramm *CMVision* [CMV] und ist aus Effizienzgründen in C++ implementiert. Die Firewire-Kamera wird über einen Linux-Rechner angesteuert. Graphische Benutzeroberfläche, Spiel-Server und Spieler sind aus Komfort-

## Schlussbemerkung

Reizvoll an dem Roboterfußballprojekt war nicht nur der große Spaßfaktor sondern auch der Einsatz einer Vielzahl von Technologien aus dem Bereich der verteilten Systeme, Realzeitsysteme, Bildverarbeitung, graphischen Benutzeroberflächen und objektorientierten Programmierung mit C++ und Java. Zu sehen waren zwar noch keine berausenden Ballstafetten, aber immerhin gelang es den Spielern in aussichtsreichen Positionen erfolgreich auf und auch in das Tor zu schießen oder den Ball abzuwehren. Aufgrund des kleinen Spielfeldes (begrenzter Bildausschnitt der Kamera) war eine Teamgröße von 2 Spielern vorteilhafter. Die anfangs als kritisch eingestufte Bildverarbeitung lief zuverlässig mit 30 Bildauswertungen je Sekunde. Voraussetzung war eine ausreichend diffuse und konstant bleibende Ausleuchtung des Spielfeldes. Als Flaschenhals erwies sich jedoch der Infrarotfunk. Neben Abschattungsproblemen führte vor allem die niedrige Übertragungsrate dazu, dass die Übersendung von nur 15 Befehlen an jeden Roboter je Sekunde möglich war. Für sehr flinke Roboter- und Ballbewegungen ist das zu wenig. Eine überlegungswerte Alternative für die Zukunft wäre der Einsatz eines wesentlich leistungsfähigeren Bluetooth-Moduls.

## Danksagung

Großen Dank geht an die Studenten Tobias Frischknecht, Christian Güttinger, Daniel Hosti, Michael Lohr, Alexander Stolina, Stefan Wedele und Michael Zwick, die mit sehr viel Freude und Engagement die hier dargestellten Konzepte entwickelt und umgesetzt haben. Für wichtige Vorarbeiten möchte ich mich auch bei den Studenten Jonas Ernst, Frank Bernhardt, Heiko Helble und Markus Jäger bedanken.

## Literaturverzeichnis

- [@Rob] <http://www.roboocup.org/>, RoboCup Official Site, Stand 10/2004.
- [@CMV] <http://www-2.cs.cmu.edu/~jbruce/cmvision/>, CMVision Realtime Color Vision, Stand 10/2004.
- [KnNo00] J. B. Knudsen und M. L. Noga, Das inoffizielle Handbuch für Lego Mindstorms Roboter, O'Reilly Verlag, 2000.