Development Of Robotic Foosball As A Versatile Platform For Robotics Research and Contests

Oded Padon

Under the direction of
Oren Zuckerman
Lifelong Kindergarten
Media Lab
MIT

Research Science Institute
September 8, 2003

Abstract

This paper considers the development of a *programmable robotic foosball table* as a versatile platform for robotics research. A *programmable robotic foosball table* is a foosball table with a computer controlled mechanical system to control the rods, and a mean of sensing the position of the ball and the rods. We discuss the design of the system that we've created, and its various implementations. In particular, we suggest its use in an international robotic programming contest.

1 Introduction

Robotics contests are an exciting way of conducting research. One of their unique advantages is their ability to expose the general public to the world of robotics. In particular, they have the potential to attract young and talented minds into robotics research. Moreover, they act as a productive research tool, as the experience and skills aquired by team-members participating in those contests greatly promotes robotics research. Some examples of existing robotics contests are:

- RoboCup [1] - A contest in which each team builds a robotic soccer team.
- Trinity College Fire-Fighting Robot Contest [2] - The teams build independent robots who find a lit candle and then extinguish it.
- 6.270 - MIT's Autonomous Robot Design Competition [3] - The teams build autonomous robots that navigate and recognize objects in the environment.

The great majority of these contests combine building robots and programming them. The advantage to the combination is the fact that we gain experience in both building and programming robots. However, since building robots is often difficult and timeconsuming, it often happens that the building portion hinders the AI programming.

Our vision is to create an interesting and versatile robotic platform that can be used for conducting many kinds of robotic AI research. Thus, in certain cases researchers can use the existing platform instead of building a new robot, enabling them to concentrate on the AI development. In particular, we wish to establish a robotics contest that does not involve building robots, instead requiring only the creating sophisticated robotic AI. Thus, teams from around the world will be able to compete without having to physically attend the contest; Participants can simply submit their algorithm and the match will be held. To

achieve all of this, we chose the platform of Robotic Foosball[1].

The idea of robotic foosball itself is not new. A foosball-playing robotic system called Kiro citeKiro was already built by a team in The Institute for Computer Science, at The University of Freiburg, Germany. Kiro is a foosball playing system with human-like motor skills, that competed with humans in several RoboCup citeRoboCup events. Kiro has achieved a good playing level, and was able to perform well even against advanced players. The system however, was not developed into a platform that allows large scale world wide robotic foosball programming contests.

2 System Overview

The system consists of four main elements:

1. The Foosball Table — A small scale foosball table, with 4 rods for each side.
2. The Robotic Hardware — A set of 16 motors connected to the 8 rods of the table, which enables the computer to control the rods.
3. The Vision System — A camera put above the table, and a vision processing system that feeds the AI with the positions of the ball or balls and of the rods.
4. The Programming Interface & Software Mechanism — A programming interface that allows researchers and amateurs from around the world to create foosball playing algorithms, and a software mechanism that allows matches to be held in which multiple algorithms have simultaneous access to the table.

Let us consider them in more details:

[1]Foosball is also known as table soccer or bar soccer

Figure 1: The foosball table with the motors and the GoGo boards

2.1 The Foosball Table & The Design of the Robotic Hardware

2.1.1 Philosophy

Unlike Kiro, our system is not meant to compete with humans. It is therefore not necessary to try to achieve human-like motor skills. Instead, we will have AI vs. AI matches. So, as long as both AIs have the same motor skills, the match is fair and interesting. We therefore chose to use a small scale table-top foosball field as the foosball table, and Legos for building the mechanical system. These choices greatly simplified the construction process, while keeping the applications (e.g. AI vs. AI matches) interesting.

2.1.2 Details

The foosball table is a table-top foosball field. It has 4 rods for each side, with the following men alignment: 1 men for the goal keeper, 2 for the defense, 4 for mid-fields, and 3 for the offense. Each rod has two degrees of freedom: Rods can be shifted laterally, and can also be rotated. The basic principle of ball manipulations is to shift a rod so that there is a men facing the ball, and then rotate the rod to kick the ball.

A mechanical unit we build out of LEGOs is attached to each rod. Each unit consists of two motors: one stationary motor that controls the lateral shift, and one motor, attached to the rod, that controls the rotation.

We used GoGo Boards [5] to power and control the motors. A GoGo Board is a computer-interfacing micro-controller that, among other features, enables the computer to control up to six motors. It enables simple two-directional on-off, as well as PWM (pulse-width modulation), which allows to move the motor in different speeds. Since each GoGo Board controls up to six motors, we had to use three boards to control our sixteen motors. The GoGo board has a very simple serial protocol which we implemented when we created the table server (section 3.3).

2.2 The Vision System

2.2.1 Philosophy

We believe that vision should always be preferred over other methods of sensing the environment. Other methods such as rotation sensors and magnetic field sensors could be used in a foosball table (rotation sensors to sense the position of the rods, and magnetic field sensors to sense the position of the ball). However, using any of these methods would have been a specialized solution for a specific problem. Vision however, is a much more general mean of sensing, and the development of vision systems not only solves a specific problem, it also makes a small contribution to the field of vision systems. We therefore chose to use vision as the only input from the environment.

2.2.2 Details

- A webcam is placed above the table.
- We have painted all the parts of the table in black.
- We used the openCV library for the vision processing.
- We position the ball or balls using standard blob techniques.
- Colorful stickers are placed on each rod, and they are used to determine the rods' lateral position.
- The rods' rotation is not yet determined by the camera. We will perform this in the future, using two color stickers on the rods to determine rotation.

2.3 The Programming Interface & Software Mechanism

The software part of the system are discussed in the next section.

3 Software

3.1 Philosophy

We wished to create a mechanism that will allow people from around the world to create algorithms for the foosball table. The programming interface should be as simple-to-use as possible. The mechanism also needs to allow the holding of matches, in which multiple algorithms have simultaneous access to the table. Also, during the algorithm developing process, the participants will require a simple way to experiment with their ideas, without submitting a full algorithm. Remote access to the table is therefore required. To summarize, the software system has to meet three main requirements:

1. Allow programmers to easily program the table.
2. Allow the holding of matches, in which multiple algorithms have simultaneous access to the table.
3. Allow remote access to the table.

3.2 Architecture

In order to meet the requirements above, we have created two main components: the programming interface and the table server. In order to allow for remote access we implemented an IP-based client server system.

When a programmer creates an algorithm using the programming interface, the result is a client application that connects to the table server over an IP-based network. The client application does not perform any vision processing or motor control. It only sends commands to the server and receives data from it. The table server runs on the computer which is connected to the camera and to the GoGo boards. It is the server who performs

image processing and sends orders to the motors, according to commands he receives from clients.

This architecture enables developers to experiment with different ideas, by controlling the table over the Internet. This also enables the holding of matches, as the server can deal with multiple client simultaneously. Real matches, where real-time processing has to be done, will be performed locally, with the algorithms and the server running on an IP based LAN.

3.3 The Table Server

The table server runs on the computer which is connected to the camera and to the GoGo boards. The server performs all the vision processing and motor control. The server can handle multiple clients at a time, sending data about the balls' and rods' position to all of the clients, and performing rod moving commands received from all clients.

Whenever the server retrieves a new frame from the camera, it processes it into a set of 18-20 numbers:

- The X and Y coordinates of the ball or balls.
- The 8 lateral positions of all the rods.
- The 8 rotational positions of all the rods (not implemented yet).

The server then sends the 18 numbers to all the clients who finished processing the previous frame. Whenever a client finishes processing a frame it notifies the server, so that it will get the next frame when it's processed.

The server also stays alert for incoming motor orders. The clients only send very simplified orders, specifying which rod to move or rotate, and the requested direction and speed. The server the figures out which GoGo board it has to contact, and which motor to activate.

This way, the mechanical and electronic system can be changed, without having to change all the clients.

3.4 The Programming Interface

We wished to create a very simple-to-use programming interface, which will allow programmers to program the table easily. The programming interface takes care of all of the technicalities, and provides the programmers with easy to use ways for:

- Retrieving the position of the ball or balls, and of the rods.
- Moving the rods to play foosball.

The entire processes of motor control and vision processing, as well as the network communication, are completely transparent for the programmers. That way, we hope to minimize the overhead that exists in all robotics research, and allow researchers to focus on innovation and the creation of smart foosball AIs, and not on technicalities.

When a programmer creates an algorithm using the programming interface, the result is an application that connects to the table server over an IP-based network. The application waits for the server to send in the data of the first frame. It then calls a function written by the programmer, and passes the data to the function. The function processes the data and decides on an appropriate action. The function then sends the orders to the server, using functions supplied by the programming interface. When the function returns, the programming application sends notifies the server that it is ready to receive data.

4 Possible Applications

The programmable foosball table is a very versatile platforms which can be used in different many ways. In particular, there are two main formats in which interesting contests can be

held:

1. Centralized — In this format, each team creates an algorithm to control the four rods of one sides. Two algorithms of different teams play a foosball match. This format allows experimenting with different AI approaches, and many creative teams working on creating such algorithms will possibly result in new insights in the field of robotic AI.

2. Decentralized — In this format, each team creates a program to control one rod, and eight different algorithms are randomly assigned to form the two sides. That way, when a team conceive their algorithm, the cannot rely on any particular play-style for the other algorithms its algorithms will be teamed with. This is a very interesting format, as it allows research in the field of decentralized processing. This format is perhaps closer to a human soccer team, in which there is no centralized processing done, and it is the individual behavior of all the players which produces the behavior of the team.

Another interesting expansion that could be made on both formats is the introduction of a second ball. This would add a new aspect of complexity to the game and to the AI required. In a foosball game with two balls, the algorithms would have to prioritize, and plan their game. The experience gained by participants in such contests would contribute to our understanding of those aspects of AI.

Although contests are a very exciting way of research, the foosball table can also be used in other ways which aren't necessarily contests. For example, the foosball table also provides a platform to experiment with the concept of learning robots. Researchers could create programs which just play with themselves (control all 8 rods), and have the robot learn the basic principles of foosball. e.g. the way the ball bounces of the walls or the men, different delicate techniques of manipulating the ball etc.

Contests and research in all formats and variations can be held using the same platform — the robotic foosball table and the programming interface. We therefore hope that our Robotic Foosball will become a useful research tool, and that it will provide the scientific community with valuable experience in the field of programming robots.

5 Future Work

- Complete the software system and create a functional version.
- Establish a foosball programming contest, promoting it, and running it.
- Examine the possibility to create an interface that will allow non-programmers to create algorithms for the foosball table, intended to be used by children as an educational tool.

6 Acknowledgments

♥ Tons of thanks and regards to my mentor, Oren Zuckerman, for his helpful mentoring, and for letting me work on an original project.

♥ Many many thanks to the Media Lab's Lego Lab, the wonderful place where I got to work.

♥ A thankful heart to my tutor, Jenny Sendova, for EVERYTHING!

♥ A bulleted heart to all the TAs: Chris, Sasen, Ed, Gabriel, Josh and Balint, for their skillful guiding of me through the unknown mysteries of LaTeX

♥ A thankful yawn to Arrak, my counselor, for not falling asleep as much as he'd like to.

♥ A thankful smile to all the other counselors, Vivek, Robby, Jamie, Malcolm, Doug and Emma, for all the fun and food I've had!

♥ My gratitude, to the Center for Excellence in Education, for giving me this rare and magnificent opportunity.

♥ A special thanks to Matt Paschke, for the entertainment.

♥ WOW, to MIT, for MIT.

References

[1] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawa, and Hitoshi Matsubara — RoboCup: A challenge problem for AI.*AI Magazine, 18(1):73 85, 1997.*

[2] Trinity College Fire-Fighting Robot Contest, refer to the internet URL: `http://www.trincoll.edu/events/robot/philosophyFAQ.htm`

[3] 6.270 - MIT's Autonomous Robot Design Competition, reffer to the internet URL: `http://www.mit.edu:8001/courses/6.270/home.html`

[4] T. Weigel and B. Nebel, KiRo — An Autonomous Table Soccer Player, *Proceedings of the International RoboCup Symposium '02, Fukuoka, Japan, 2002.*

[5] Arnan Sipitakiat, Paulo Blikstein, David P. Cavallo — The GoGo Board: Moving towards highly available computational tools in learning environments. May be retrieved in the internet URL: `http://web.media.mit.edu/~paulo/publications/academic/blikstein_sipitakiat_gogo_icl2002.pdf`