Soccer Teleworkbench for Development and Analysis of Robot Soccer

Emad Monier, Ulf Witkowski, and Andry Tanoto
Heinz Nixdorf Institute, System and Circuit Technology
University of Paderborn, Germany
{monier, witkowski, tanoto}@hni.upb.de

Abstract

This paper describes a platform for assisting researchers and/or students to perform and validate experiments using real robots in robot soccer domain. Some important features are: live-video of the experiment, recording video for the experiment, data logging of exchanged messages, robots positioning system, downloading user-specific program to each robot, and post-experiment analysis tool. The platform can be used locally or remotely via the Internet.

1 Introduction

Robot soccer is an attractive domain for researchers and students working in the field of autonomous robots. However developing (coding, testing, and debugging) robots for such domain is a rather complex task, especially for Kheperasot which requires programs to run onboard the robots.

Robot simulator is commonly used during the development since it can help robot developer to analyze and debug the robot program. However, simulation, in most cases, just simplifies the real situation. Due to noises from real sensors and non-linearity in hardware components, programs that run well in simulator may fail in experiments using real robots. Thus, for validation purpose, experiments are still necessary.

However, doing experiments with real robots is a tedious task. Moreover, it is also repetitive. Before experiments, we need to prepare the robots, set up the environment, and so on. During the experiments, we need to record some important events and messages as well as the video of the experiments if necessary. After that, we need to analyze the data generated during the experiments. Although the result of an experiment is satisfying, we need to repeat it several times to make sure that the result is valid and not just by chance. For example, to ensure the developed strategy algorithm working robustly in varying situations, we need to test the program by putting the robot at some different initial positions and orientations.

For these reasons there is a need for a tool that can complement robot simulator for developing robot programs in robot soccer domain which can ease and assist robot developers in analyzing and debugging experiments using real robot.

In this paper we present a tool called Soccer Teleworkbench which we will refer through out the paper as STWB, which is actually a smaller version of a system called Teleworkbech [1], developed in our research group at Heinz Nixdorf Institute. It has several features that can simplify the execution, analysis, and debugging of robot soccer experiment for Kheperasot. The features include web-based interface, live-video of the experiment, recording video for the experiment, data logging of all messages exchanged such as sensor data, communication message, etc., robots tracker to calculate position and orientation of the robots on the field, downloading user-specific program to each robot, and post-experiment analysis tool to help robot developer to analyze the experiment.

The rest of this paper is organized as follows. In the second section, we present more detailed information of our system. Afterward, an overview of the robot platform we are using will be described. Later on, we present an experiment to demonstrate STWB and other tools developed in our research group. At the end, a short conclusion will conclude this paper.

2 STWB System Description

The proposed system of the STWB, shown in Figure 1, comprises one field used in Kheperasot leagues, equipped with a webcam above it, a wireless communication system using Bluetooth and one computer (PC or Laptop) which performs tasks such as image processing, video broadcasting, and web hosting.

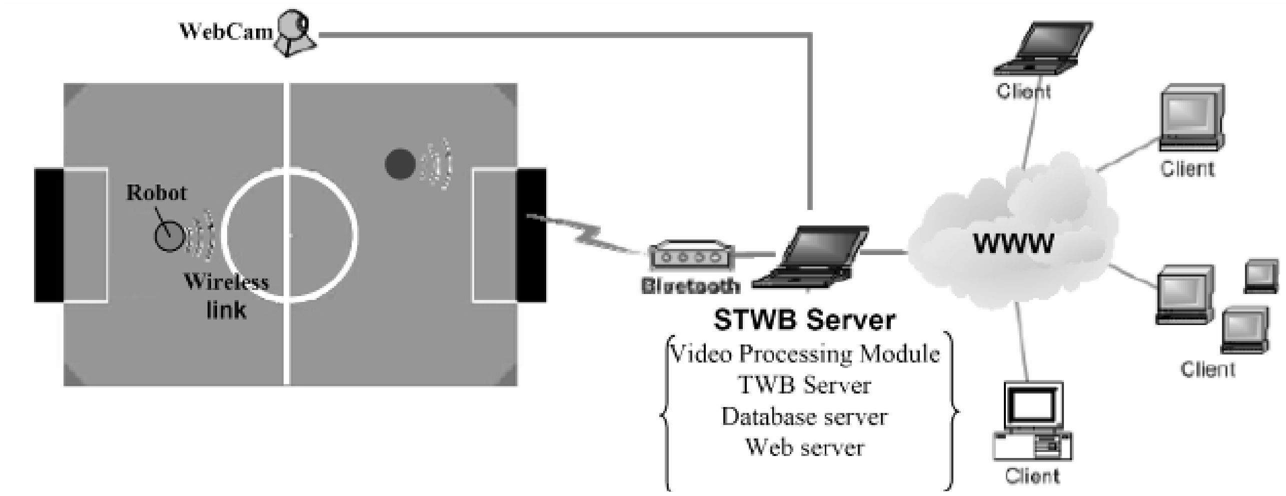

Figure 1: The Soccer Teleworkbench diagram.

The field of the STWB is the same as the fields used in the Kheperasot. Further details of the field as well as the rules of the game can be found in [2]. A webcam is fixed over the field to provide STWB with raw video data needed for image processing and live video. More details of the system will follow in the next subsections.

2.1 STWB server

The STWB server is responsible for downloading programs to robots, routing information between users through WWW server and robots, and logging all events as well as communication messages. The data in the log file will be used as inputs to the post-experiment analysis module.

2.2 Video Processing Module

The video processing module has two roles. First is to provide positions as well as orientations of the robots on the field. The second is to broadcast the video data to the streaming server. As the robot position provider, the video processing module captures the raw frame data from the camera and processes it by detecting objects on the image, and classifying them into two groups: robots and non-robots. To enable classification, we use an identifier which is unique so that it will get recognized easily. The identifier is of circle-form consisting of two colors, each of which covers a half-circle area, and a white box stretching from the center to the edge. This circle is surrounded by a black border for easier identification (See Figure 2). After the position and orientation information is calculated, the robot position extractor will send the information to the STWB server module.

As a video broadcaster, it encodes the raw video data and streams it to the streaming server. In our STWB system, we use open-source software called MP4LIVE which is part of the MPEG4IP project [3] as a broadcaster.

2.3 WWW Server

The WWW server is used to host the web interface for the STWB system. The web interface written in PHP is used to enable remote users to access the STWB. With this interface the users can do the following tasks: Watch the live video of the soccer field, command the STWB, and download the soccer program to a robot.

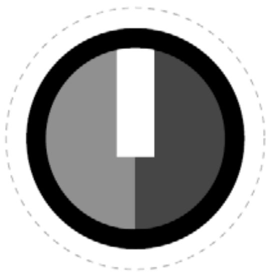

Figure 2: The color mark for Khepera identification.

2.4 Video Streaming Server

In our STWB, the heart of the video streaming server is software called Darwin Streaming Server (DSS), which is the open source version of Quick Time Streaming Server from Apple [4]. One important advantage of using this program is that we can stream hinted QuickTime, MPEG-4, and 3GPP files over the internet Real-Time Transport Protocol (RTP) and Real-Time Streaming Protocol (RTSP) protocols. The first standard is used to transmit real-time video content and the latter is used to control the stream of real-time video content.

2.5 Post Experiment Analysis and Visualization Tool

The log files that contain all information of the experiment such as robot positions, sensor data, and messages between robots and so on, are used as an input to an experiment post-analysis and visualization tool developed in our group [5]. This tool uses MPEG-4 standard to visualize information acquired during an experiment. On top of the video recorded by the webcam, some computer generated objects representing the information needed to be visualized are overlaid and the appearance of these objects can be controlled by users.

3 Robot Platform

Robot soccer in the Khepera league uses Khepera robot base [6] that is equipped with a camera top module. The KheperaSot rules do not specify the type of the camera, but mostly a linear camera module with 64 pixel with 8 bit gray scale is used [7]. This type of camera limits the object recognition capability due to the missing 2D pixel array and the missing color information. For this reason we have developed a 2D color camera module for the Khepera robot with flexible on-board image processing features. Figure 3 depicts the Khepera robot equipped with the new camera top module and additional image processing module.

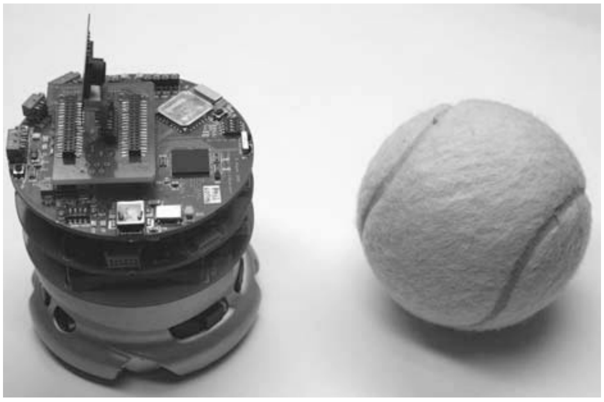

Figure 3: Khepera robot equipped with FPGA module and 2D color camera top turret

The camera is a miniature 2D color CMOS camera from Transchip, model TC5740 with VGA resolution. To control the camera we use an FPGA (field programmable gate array) module that additionally processes captured images. Results of the image processing are symbolic data like position of the ball in the image plane, size of the ball, coordinates (left and right corner) of the goal (if visible), and coordinates of the opponent. In this way, the microcontroller of the robot's base module will not be burdened with image processing tasks. Thus the processing power can be used for realizing soccer strategy, planning, positioning, and drive control.

Besides the image processing capability, the FPGA module offers important features which are very useful during the development phase of the image processing and soccer algorithms. The module is equipped with a Bluetooth transceiver and USB 2.0 port. Via the Bluetooth link, status data of the robot can be transferred wirelessly for analyzing the robot's behavior. This feature is used in the soccer teleworkbench environment to record all actions of the robot along with the video captured by the camera over the pitch. Examples are presented in the experiment section. Via the USB port the programmer is able to see on a computer screen what the robot *sees*. In addition, all (intermediate) results of the image processing performed by the FPGA can be displayed as a real time video. For this, the robot is connected via a very flexible USB cable to a computer. Although the cable is limiting the area of operation of the robot, the video streaming is a very helpful feature during the development and calibration phase. For a soccer match, the USB cable will not be used. Features of the FPGA module (Figure 4) are:

- FPGA Virtex-E XCV300E; flexible programming by user via VHDL; supports dyn. reconfiguration
- Cypress EZ-USB FX2 USB 2.0 micro-controller
- Multi-chip memory package with 128 MBit Flash and 32 MBit SRAM
- Programmable clock (up to 60 MHz)
- ADC and DAC for sound in/output
- Communication links: Bluetooth (point-to-point, piconet scatternet), USB 2.0, RS232, I2C, IrDA

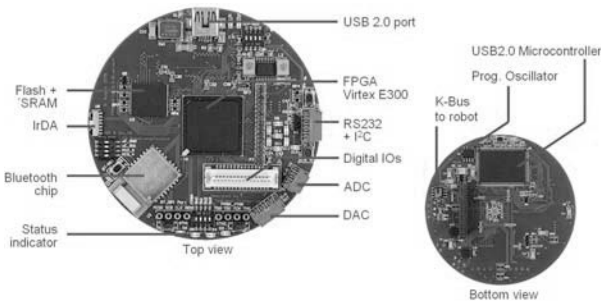

Figure 4: FGPA module for image processing tasks