5dpo Team Description for RoboCup 2006

Paulo Costa\textsuperscript{1,2}, António Paulo Moreira\textsuperscript{1,2}, Armando Sousa\textsuperscript{1,2}, Pedro Costa\textsuperscript{1}, Susana Gaio\textsuperscript{1}, Paulo Marques\textsuperscript{3}, Paulo Malheiros\textsuperscript{4}

\textsuperscript{1} FEUP - Faculdade de Engenharia da Universidade do Porto  
Rua Roberto Frias, s/n, 4200-465 Porto, Portugal  
\textsuperscript{2} ISR – Instituto de Sistemas e Robótica – Porto  
\textsuperscript{3} FluidInova, Engenharia de Fluidos, SA  
\textsuperscript{4} Software Development Department - Grupo PIE, SA

\{paco, amoreira, pmalheiros, asousa, pedrogc, sgaio\}@fe.up.pt  
\texttt{pmarques@grupopie.com}  
\texttt{paulo.malheiros@fluidinova.pt}  
http://www.fe.up.pt/~robosoc

Abstract. This paper very briefly describes the current software architecture and the mechanical and hardware design of the 5dpo. The overall hardware and software architecture, is presented and so are the main modules. The design of a Robotic soccer Team is an evolving process where the weaknesses and strengths of previous years drive new enhancements. The inevitable rule changes also take an important role on design.

1 Introduction

The 5dpo team, since 1998, has already been present in 10 international Robocup competitions in the Small size League (SSL) and a few local events. Since then, the main research topics have been related to the modeling, control and sensorial problems posed by mobile robots. Of course, other problems like the coordination between multiple robots, the design of realistic simulators and other technological issues like reliable radio and IR links where also tackled. This effort payed off very early as it was possible to transfer the acquired know-how to other project, apparently unrelated to mobile robotics.

This presentation starts with a brief description of the 5dpo team robots. Some design lessons learned from previous robots' generations are presented. It must be stressed that is a continuous, almost evolutionary, process.

Most of the mechanical, hardware and software design were done by team members.
2 Robot Design

It is normal for robots to exchange roles frequently in a SSL game. As such, it is advantageous to have the same design for all of them. The goal keeper could be one possible exception to this uniformity rule. Previous versions of the team had a specially designed goal keeper. The small benefit from that design did not offset its disadvantages: special code for the goal keeper, special code for the case where a “normal” robot had to perform the goal keeper role and the extra hardware and mechanical problems to maintain a goal keeper substitute. In the present design, a normal 5dpo robot is perfectly capable of performing the goal keeper role. Another advantage of this uniformity is that it eases the maintenance and the availability of spare parts or even substitute robots.

The SSL player has omni-directional locomotion and two kicker devices. Both kickers are based on electromagnetic solenoids. The applied voltage pulse can have its width adjusted to push the ball with controlled power. One of the devices is able to lift the ball within the rules of the competition and the other is more powerful and pushes the ball close to the floor. This second kicker is faster.

As all the sensors in the robots are used locally, it is not necessary a radio uplink so the radio protocol and the local software can be optimized to one way communications.
Mechanical Evolution

In previous years the omni-directional wheels had a nylon body and small metallic wheels. These small wheels were standard washers and as they were very thin, a lot of them could be fitted on a wheel. This configuration allows a smooth movement as the gaps are very small. Unfortunately, there are also some disadvantages: these robots can only work on some types of carpet and there is a considerable amount of friction when the wheels have simultaneously traction and slippage. This year the wheel are being upgraded. The washers are being replaced by Teflon wheels with an O-Ring. These wheels show less axial friction even under heavy traction and, due to the rubber O-Ring, better grip.

Electronics and Communications

The robots use 3 packs constituted by 5 Ni-MH batteries to provide 18V. A low drop regulator tapped from the lower pack (6 Volts) is used to drive digital circuits. All other parts of the robot use the full 18V. Both electromagnetic kickers require an higher voltage. That is achieved by a pump based circuit. Energy from the batteries is transferred to capacitors while the voltage is multiplied in order to strengthen the kick. Carefully controlling kicking times allows for controlled passing/kicking strength. The signals for the motors and the kickers are generated by a microcontroller an Atmel ATMega8

The radio protocol requires a second ATMega8 that handles all real time communications. There are two small single frequency Radio Frequency (RF) modules (433 MHz and 418MHz). As both modules are always present An additional communication channel is provided through an InfraRed (IR) link. The emitting LEDs are at the camera level, pointing down. The information transmitted via IR and RF is the same.

Vision System

Like previous years, the hoods of the robots have a circular non-repeating barcode that allows robot tracking of robot identity, position and angle.

The vision system is based on two Firewire cameras, one for each mid-field. Two vision boards in the same computer receive synchronized frames. The Vision Server communicates with the Strategy Server over an Ethernet 100 Mb/s network.
It uses a fuzzy system to classify the colour of each pixel and aggregates them in contiguous groups. The observed groups are matched and that information is incorporated in the state resorting to a set of Kalman filters tuned to the dynamics of each kind of object. The Strategy Server uses the global system state as well as strategy rules to generate motor orders and send them to the players[7][8][9].

**Decision and Control System**

The Strategy Server is based on a hierarchical state machine engine. Global team tactics depend on a state machine that switches roles according to the actual game state and a few parameters that control the overall strategy. Each robot has a Role that selects the Tasks to perform. Each Task is decomposed in a series of different Actions.

The control loop is closed through the cameras, as seen in figure 3. It must be stressed that while the "sampling" frequency of this loop is 25 Hz, there is some intrinsic lag that also influences its performance. The captured image takes some time to be delivered from the camera to the PC memory, then some time is spent processing it and then the decision is made. There is also the time spent sending the orders over the RF/IR channels to the robots. The controller performance benefits from a prediction layer where the low level control is calculated taking into consideration known system lag.
Fig. 3. A schematic of the control loop. Most of the lag can be attributed to three activities: the transmission of the captured image to the camera, the image processing and the radio transmission from the decision system to the robots.

Knowing the delay, it is possible to tune the dynamics prediction layer, based on a Kalman Filter, and use the ball and robots dynamics to predict the state at the time when the low level commands are to be executed thus improving the team performance.

3 Details Of The Vision System

The vision system plays a crucial role on a team performance. There are a few issues related to it: the global accuracy, the ability to find the robots and the ball most of the time, the absence of ghost measures and the real time performance. This last issue is becoming increasingly important because of the crescent robots’ speed. Considering the closed control loop formed by the robot, the vision and decision systems and the radio link, there is a considerable amount of lag introduced by the vision system that degrades the robots’ performance.

In contrast with other teams[4][6](figure 4) the 5dpo robots only require the use of the official color, plus the black and white pattern[2][3][5]. The pattern can have other colors provided that their luminance is enough different.
Fig. 4. The 5dpo SSL team in a game with the FU-fighter team

This kind of markers have the advantage that ease color calibration because only three colors must be tracked: the orange for the ball, yellow and blue for the teams. If both teams use this approach the calibration intervals can be larger and its easy to track the robots even under non-uniform lightning.

Fig. 5. Bar codes for the blue team and a sample scan circle on number 1 (shown in red)

The round-shaped bar code identifies the robot and also serves to extract its angle. Around the circular color of the team (yellow or blue) there is room for 6 bits of black or white sectors. As only 5 robots are needed on the field at the same time, from all the possible 6 bit combinations, were chosen combinations that had the maximum number of transitions and that are not a rotation of previously chosen codes. From this bar code it is possible to extract a unique binary code for each robot as well as its orientation. Orientation noise is less then 2.5 degrees.
The De-Bayering Problem

A Bayer camera (or one that has that pixel format available) allow us the have along the original signals present on the CCD. Most color CCDs have an array of pixels sensitive to the three color primaries arranged like in figure 4. This is called a Bayer pattern and has twice many green pixels as blue or red. This is an optimization connected with fact that human vision is more sensitive to luminance than chrominance and the green component is the one responsible for most of the perceived luminance. From the Bayer pattern we can estimate the three components RGB for each pixel. One of the simplest ways of doing that is assuming that a pixel is composed by a square of two by two original Bayer pixels. The next neighbor shares two pixels from the previous one, both on the horizontal and in the vertical directions[1]. This is a fast way to estimate the RGB components but is somewhat crude. Better methods, using more neighbors and clever analysis of the actual pattern can improve the quality of the estimation but are costly in terms of processing time. Also, are optimized for human viewing and not to a processing algorithm.

![Bayer Pattern for a 5x5 CCD](image)

Each pixel can be calculated by:

- \( P_{1\text{red}} = R2, \) \( P_{1\text{green}} = (G1+G7)/2, \) \( P_{1\text{blue}} = B6 \)
- \( P_{2\text{red}} = R2, \) \( P_{2\text{green}} = (G7+G3)/2, \) \( P_{2\text{blue}} = B8 \)
- ...
- \( P_{6\text{red}} = R12, \) \( P_{6\text{green}} = (G7+G11)/2, \) \( P_{6\text{blue}} = B8 \)
- ...

This approach has a significant problem in regions where there is a sharp transition, like the white line on the green field. There, the fact that the color sampling is done on different places for each pixel can lead to bad estimates of the RGB values. The pixels there can exhibit colors not present at all on the real image. Worse, some of that color can be mistaken for the ball or even the blue or yellow markers creating a lot of phantom robots.

While that problem can be solved by a more elaborate de-bayering algorithm the extra processing burden can become to high sustain good frame rates. Even if the frame rate is not impaired there is the burden of an additional delay in the control loop. A better approach is to filter out those pixels from the classification stage. These pixels can be always marked as not belonging to any “interesting” color. One hint that a pixel is a “bad” one can be extracted from the two green values. If they are signifi-
cantly different, that means that the pixels belong to a region where there is a sharp transition on the Luminance. This approach was successfully tested and can detect almost all the transitions leading to color artifacts.

This means that:

$$\text{for } P_{\text{green}} = \frac{(G_a + G_b)}{2} \text{ then if } |G_a - G_b| > \text{thresh} \text{ then this is bad pixel}$$

Of course the \text{thresh} value can depend on the noise level and controls the strictness of this filter. A low value can lead to good pixels being discarded. This approach eliminates most of the “bad” pixels present in horizontal or vertical gradients. Most of the cases where this happens, in a typical Robocup setup, fall in this category. The white lines on the field are mostly horizontal or vertical and there, in the green white transitions, is where the bayer pattern is responsible for most of the mis-predicted colors.

References