Eagle Knights 2014: Small Size League Team Description Paper

Marco Morales, Alberto Candela, Alejandro Escalante, Javier Sagastuy, Luis Eduardo Pérez, Andre Possani

Robotics Laboratory, Department of Digital Systems, College of Engineering
Instituto Tecnológico Autonómo de México - ITAM, Mexico City, Mexico.

Abstract. In this paper we describe the architecture of our RoboCup Small Size League 2014 team. This architecture has improvements in hardware and a radically different design in software from what we had presented until 2013. In hardware, we improved the PID robot controller, and we are in the progress of changing motor types. In software, we now have a modular system that uses ROS in order to integrate modules.

Keywords: Small Size League, RoboCup, ROS

1 Introduction

RoboCup [1] is an international joint project to advance research on artificial intelligence and robotics through a grand challenge: design a robotics soccer team able to defeat the FIFA world champion by 2050. The Small Size League takes this challenge by promoting research on multi-agent cooperation and control. Two teams of six mobile robots up to 18 cm in diameter play soccer on a 4.05 by 6.05 m carpeted soccer field (optionally doubled this year). Aerial cameras send video signals to a shared vision system[2] that estimates the position of the robots and of the ball on the field. This information is then passed to an AI system that produces control commands that are sent to each of the robots through a wireless link. An external referee box indicates the state of the game to the central computer.

The Eagle Knights SSL team was founded in 2003 and participated officially for the first time in Robocup 2005. Our team was the first Latin American team consistently obtaining top results in all its regional RoboCup participation, 3rd and 2nd place in US Open 2003 and 2004, respectively, and 1st place in Latin American Open 2004 and 2005. We have also participated in the following RoboCup competitions: Osaka, Japan 2005; Bremen, Germany 2006; Atlanta, USA 2007; Suzhou, China 2008; Graz, Austria 2009 ; Singapore 2010; Istanbul, Turkey 2011; and Mexico City, Mexico 2012.

Eagle Knights official website http://robotica.itam.mx/ssl
Qualification video URL http://youtu.be/tG5Roagbjzs

2 Team Constitution

Our team is integrated by Faculty and undergraduate students from ITAM (Instituto Tecnológico Autónomo de México).

- **Faculty director** Prof. Marco Morales, PhD.
- **Faculty member** Prof. Andre Possani, PhD.
- **College student members** Alberto Candela Garza (ME&IE), Alejandro Escalante Arrieta, Luis Eduardo Pérez Estrada (CE), Javier Sagastuy Breña (CE&BAM), Frida Gail Rojas Contreras (ME), Karen L. Poblete Rodríguez (CE&TE), David Bosch (IE), Christian Peter Hopf Valenzuela (IE), and Ricardo Alonso Arrieta (ME).

3 Overview of the Eagle Knigths Small Size League System EK-bots

We currently have six robots that satisfy the constraints set in the SSL rules:

The height of each robot is 140 mm
The maximum diameter of its projection to the ground is 178 mm
The maximum percentage of ball coverage is 19%.

We have been making changes in our robots since 2012 and now they are fully functional based on the Arduino MEGA 2560 microcontroller. We are trying different wheels also, with some improvements in motion quality. We have also refined the PID controllers for the motors. Our motors are still regular DC motors.

In software, we started a new system from scratch. Now, we are using ROS as our integration backbone and we have developed nodes to support the different functions required. A vision node is a wrapper for the SSL vision system. A referee box node is a wrapper for the SSL referee box. A strategy node defines the actions that each player should make according to the state of the game, the role of the player, and the configurations of all the robots and of the ball. An action node defines the next position for each robot according to the action it is performing. A trajectory node receives the current and goal configuration for each robot and computes the velocity command to be sent to them. A communications node receives velocity commands for the robots and passes them through an XBee radio. Currently the strategy, action and trajectory nodes are integrated, but they will be completely independent by RoboCup 2014.

4 Software

As shown in Fig. 1 our software comprises the following modules implemented as ROS nodes: Vision, Referee Box, Strategy, Action, Trajectory, and Communications.

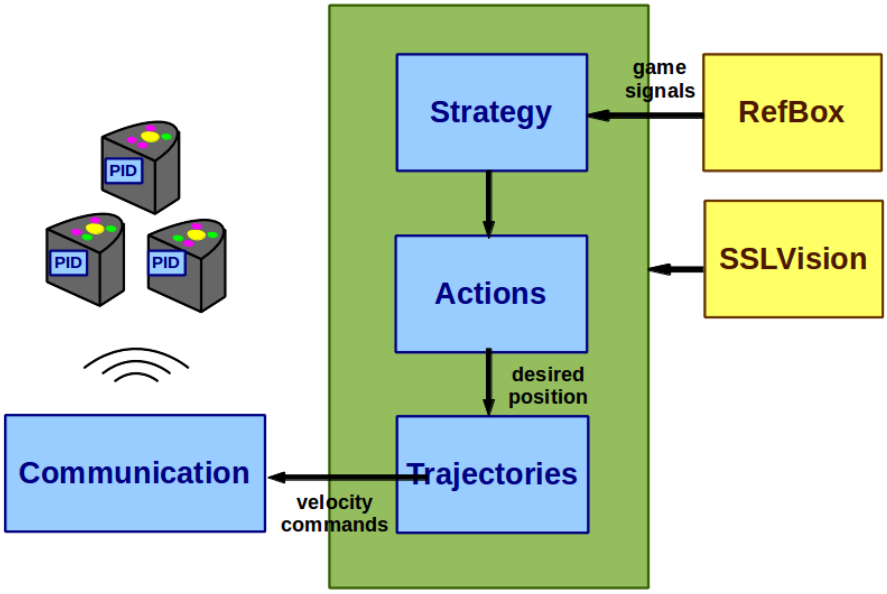

Fig. 1. Eagle Knights SSL System Architecture

4.1 Vision

A vision node is a wrapper for the SSL vision system. The Shared Vision System [2] digitally processes two video signals from the cameras mounted on top of the field. Its main tasks are:

Capture video in real time from cameras mounted on top of the field.
Recognize the colored patterns specified by the rules of the robots and ball. These patterns distinguish each robot and its team (yellow or blue) through 50-mm circular patches arranged in a unique way as defined in the rules of the SSL [1]. The ball is a standard orange golf ball.
Adapt to different lighting conditions through color calibration.
Localize the position and orientation of robots of both teams and the position of the ball.

In the past, when we used our own vision system, we implemented a number of algorithms to make our system more robust to different light conditions [3]. In our current system we implemented a ROS client that produces a ROS topic per robot and another one for the ball.

4.2 Referee Box

A referee box node is a wrapper for the SSL referee box. This module controls the flow of the game, the robots are restricted to obey its commands. The Referee

communicates the state of the game as decided by the referee through an ethernet link. These messages are converted into ROS topics that are used by the role node, the action node, and the trajectory node.

4.3 Strategy

A strategy node defines the actions that each player should make according to the state of the game, the role of the player, and the configurations of all the robots and of the ball.

The role players we currently support are goalkeeper, defense, and forward. Currently, our strategies are very basic. The goalkeeper is always blocking the ball within the goal area. The defense is blocking the ball outside the goal area. The forward follows the ball until it is close to it, and shoots towards the opposite goal.

In the future we are exploring to apply motion planning and task planning techniques in order to identify potentially succesful strategies.

4.4 Action

An action node defines the next position for each robot according to the action it is performing. Although several actions consist only of one move, some of them may include several distinct moves that are encoded as a state machine. At any given time, the action node will produce a next position for the robot according to the state machine of the action it is performing. The actions that can be currently performed are block the ball, follow the ball, and shoot the ball.

When blocking the ball, the straight line between the ball and our team goal is computed, and the robot should move to a point over that straight line depending on its role. If it is a goalie, it will move within the goal area, while if it is a defense, it will move outside the goal area.

When following the ball, the straight line between the robot and the ball is computed and the robot will move to a point within a small distance of the ball.

When shooting the ball, the actions are encoded in a state machine. If the robot is not close to the ball, then it will move closer to the ball. If it is close to the ball, it will compute the straight line between the ball and the opposite team's goal and it will move behind the ball. If it is already behind the ball, it will shoot the ball.

4.5 Trajectory

A trajectory node receives the current and goal configuration for each robot and computes the velocity command to be sent to them. This node takes in consideration the dynamic limitations of the robot to define a maximum acceleration and speed.

Also, this node avoids obstacles (other robots) through a potential functions [4]. We are planning to integrate this approach with the one we used in the past based on a geometrical exploring tree (GET) [5].

4.6 Communications

A communications node receives velocity commands for the robots and passes them through a zigbee radio. More details on the radio are provided in Section 5.1.

4.7 Auxiliary tools

In order to speed the development, we frequently test our software replacing our robots with the ROS turtlesim. This simulator lacks some functionality of our robots, but allows us to quickly debug behaviors.

4.8 Robot Operating System

We use the Robot Operating System (ROS) as the backbone of our system. It provides a framework to develop modular and distributed computation that has allowed us to produce compact modules that are easy to test and to maintain. In ROS, we develop nodes that can communicate with each other through topics. Each node can provide its results as messages published to the appropriate topic. A node that needs a piece of information subscribes to the corresponding topic. Also, if using the right topics we can replace the robot with a robot simulator that allows us to test our algorithms even without real robots.

5 Hardware

Our robots have five Faulhaber 2224P0212 motors with gearheads 14:1 (four motors for the wheels and one for the dribbler) [6], a low resistance solenoid, a microcontroller, an XBee radio, a single printed circuit board and two Lithium Polymer batteries.

5.1 Wireless Communication

We use XBee Series 1 radios, at a frequency of 2.4 GHz, for communications. The communications node broadcasts command velocities through one radio and each robot has its own radio to gather its own command. We used the X-CTU software to configure and test the XBee modules. They use either a 16-bit or 64-bit source address. We use an Arduino library to handle the serial interface required for the communications.

5.2 Omni-Directional Drive Microcontroller

We use the Arduino MEGA 2560 board based on the ATmega 2560 microcontroller. Here we implemented a mapping from robot velocities (linear and angular) to motor velocities that are fed into a PID controller for each motor velocity. The PIDs read actual motor speeds from the motor encoders and compute speed

corrections to adjust the duty cycle of the PWM signals for the control of each motor.

One of the problems we previously faced was in speed computation for the motors. The encoders were attached to processor inputs that generated an interrupt at every pulse generated by each encoder. Since the frequency of interrupts was so high, the processor had little time to manage communications and the PID resulting in very slow control actions for our robots. We now use the Timer/Counter system of the microcontroller in order to compute the frequency of encoder pulses without needing interrupts. Also, we implemented a 4x1 multiplexor that allows us to use a single input for the four motors. Our program alternates the computation of speed for each motor intime.

5.3 Kicker Control System

We are addressing issues that we have had with our kicker. We use a push type solenoid. We have four 7.4V/ 700mA batteries, equivalent to 31 Watts of power. Since this amount of power is not enough to achieve minimum performance with the solenoid, we store energy through a voltage multiplier and discharge it when solenoid is activated. Activation happens when two conditions meet: a kicking signal is activated by the software, and an infrared sensor system in the bottom of the robot senses that the robot has the ball.

6 Research timeline for RoboCup 2014

Since our last competition in RoboCup 2012, we continued with hardware improvements and we started our software from scratch. This approach has already given us good results, because in a relatively short time we have a functional system with basic playing abilities. The current status of our system is that our robots can play with basic strategies. We will be working in the following projects in order to get

Referee Box Node We will make this node functional.

Predictor Node We will complement our SSL Vision node with Kalman Filters to estimate future states of the ball and opposite robots.

Strategy Node We currently only have a basic playing strategy. We will develop strategies for many more game situations than what we already have. Here we will consider to use again an expert system. Also, we currently do not support the signals from the referee box.

Action Node We will work intensely in having a good set of abilities for our robots. For example, we didn't include passing in our qualification video.

Trajectory Node Currently, our trajectories go at constant speed, and we are moving our robots slowly in order not to risk burning our motors. We will incorporate acceleration limits so our robots can go faster.

Motion We already have a design of our robot structures with brushless motors, we will finish testing the electronics and replace our current DC motors.

Structure We have an improved structure design. As soon as our new motors are tested we will migrate to the new structure.

7 Conclusions

Here we described our robotic soccer team in order to participate in RoboCup 2014. This year we have made a radical redesign of our systems and we already have a functional team. We will keep working in our robots in order to be competitive in the RoboCup.

8 Acknowledgements

This work is supported by the Asociación Mexicana de Cultura, A.C. and the Instituto Tecnológico Autónomo de México.

References

1. Laws of the robocup small size league 2013. http://robocupssl.cpe.ku.ac.th/rules:main.
2. S. Zickler, T. Laue, O. Birbach, M. Wongphati, and M. Veloso. SSL-Vision: The Shared Vision System for the RoboCup Small Size League. *RoboCup 2009: Robot Soccer World Cup XIII*, pages 425–436, 2009.
3. Ernesto Torres and Alfredo Weitzenfeld. RoboCup small-size league: Using neural networks to learn color segmentation during visual processing. In *ENRI-LARS*, Salvador, Brasil, 2008.
4. O. Khatib. Real–time obstacle avoidance for manipulators and mobile robots. *Int. J. Robot. Res.*, 5(1):90–98, 1986.
5. Basu A. Elnagar, A. Local path planning in dynamic environments with uncertainty. pages 183 –190, Oct 1994.
6. Micromo. http://www.faulhaber-group.com/uploadpk/e_201_MIN.pdf.