Eagle Knights 2016: Small Size League Team Description Paper

Edgar Granados, Maria Fernanda Borge, Alan Córdova, and Marco Morales

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 2016 team. Both, our hardware and software have evolved since 2014. In hardware, we are now using brushless motors, new computational components based on a micro-controller and an FPGA, and WiFi communications. In software, we have improved all of our ROS modules.

Keywords: Small Size League, RoboCup, ROS

1 Introduction

Here, we present the Eagle Knights system to participate in the RoboCup 2016. The 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 and 15 cm in height play soccer on a 9 m by 6 m carpeted soccer field. Aerial cameras placed 4m above the playing surface 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. We are working to have a competitive team again within the next 2 years.

Eagle Knights official website http://robotica.itam.mx/ssl
Qualification video URL http://robotica.itam.mx/videos/EK-qualification-ssl-2016.mp4

2 Team Constitution

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

- **Faculty advisor** Prof. Marco Morales, PhD.
- **College student members** Edgar Alejandro Granados Osegueda, Maria Fernanada Borge Chávez, and Alan Córdova Posadas.

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

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

The height of each robot is 147 mm
The maximum diameter of its projection to the ground is 170 mm
The maximum percentage of ball coverage is 15%.

We have continued to evolve our robots' hardware to improve their performance. The on-board computing system is a Mojo V3 - FPGA development board. We use Maxon 200142 brushless motors controlled through Trackstar 1:18 scale brushelss systems. We designed our own gear to reduce speed. Our communications use the XBee WiFi Module. Figure 1 shows two of our robots.

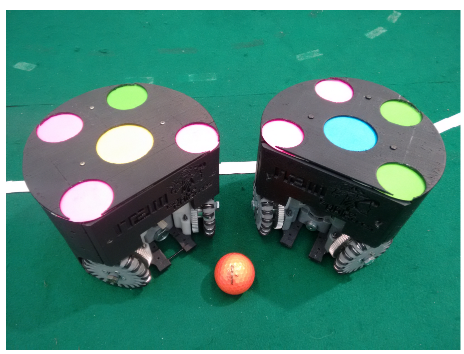

Fig. 1. Eagle Knights SSL Robots

In software, we have continued the development of our ROS nodes that, when finished, will comprise a distributed control system. 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/motion planning node defines the next position for each robot according to the action it is performing and the obstacles (other robots) in the environment. A trajectory node computes the velocity commands to be sent to each robot based on their current and goal configurations. A communications node receives velocity commands for the robots and passes them through an XBee WiFi Arduino shield. A monitor node allows us to visualize the status of the robots and provide manual controls for testing. Currently the nodes that are functional are Vision, Referee Box, Trajectory (still needs improvements), Monitor and Communications. The rest of the nodes will be completed by RoboCup 2016.

4 Software

As shown in Fig. 2 our software comprises the following modules implemented as ROS nodes: Vision, Referee Box, Strategy, Action/Motion Planning, Trajectory, and Communications.

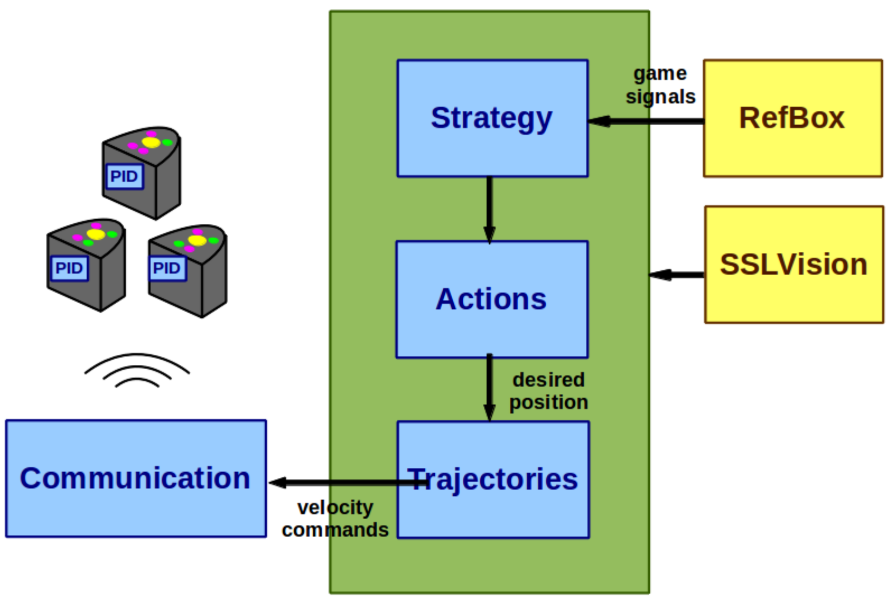

Fig. 2. Eagle Knights SSL System Architecture

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.

We are currently redesigning this node with plans to support the following roles: 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/Motion Planning

An action/motion planning 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. In the future we will support motion planning for avoiding other robots in the field either through potential functions [4] or through a geometric exploring tree (GET) [5].

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. We currently move in a straight line at constant speed between the start and the goal, but we plan to explore using a quintic-polynomial for the speed.

4.6 Communications

A communications node receives velocity commands for the robots and passes them through a WiFi connection.

4.7 Monitor

We have developed a Monitor with a GUI to track the robots, manual control and to display useful debugging output from the robots.

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

Figure 3 shows some of the internal parts of our robots.

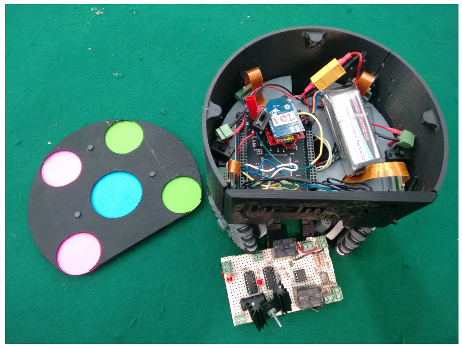

Fig. 3. Some internal parts of one of our robots

5.1 Actuators

Our robots have Maxon 200142 motors that run at an up to 4370 rpm. We have designed our own gear train with a reduction ratio of 3.5. We use a Trackstar 1:18 scale brushless system for driving the motors. This allows our robots to move up to 3.55m/s. Our kicker uses a low resistance solenoid.

5.2 Omni-Directional Drive Microcontroller

We use the Mojo V3-FPGA development board as our on-board computing system. Since it has an ARV ATMega32 microcontroller and a Spartan 6 FPGA, we can distribute computation in the robot itself. The microcontroller handles the communication with the ROS modules and transforms the desired cartesian speeds into motor speeds. Here we implemented a mapping from robot velocities (linear and angular) to motor velocities that are fed into a controller for each motor velocity. Simultaneously, the FPGA handles the actual control of each motor. It reads actual motor speeds and computes speed corrections to adjust the duty cycle of the PWM signals for each motor.

5.3 Kicker Control System

Our kicker uses 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 2016

Since our last competition in RoboCup 2012, we continued with hardware improvements and we started our software from scratch. The current status of our system is that our robots can move in basic ways. We will be working in the following projects in order to get the system fully working by RoboCup 2016:

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.

Action/Motion Planning 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. Also, we still do not support obstacle avoidance.

Trajectory Node Currently, our trajectories go at constant speed. We will try other approaches that handle better acceleration limits, such as a quintic polynomial.

7 Conclusions

Here we described our robotic soccer team in order to participate in RoboCup 2016. Our current capabilities are basic, however we plan to have a team able to play by the time of the competition. Also, we have a long-term plan with the goal of having a competitive team with two years.

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. http://wiki.robocup.org/wiki/Small$_{Size_League}/Rules$.
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.