Eagle Knights-RoboBulls 2009: Small Size League

Ernesto Torres[1], Jesús Rodríguez[1], Paolo Aguilar[1], Carlos Alegría[1], Leonardo Ramos [1], Allistair Moses[2], Jeremy Blackburn[2], Marco Morales[1] and
Alfredo Weitzenfeld[123]
1 Robotics Laboratory, Division of Engineering, ITAM, México DF
2 RoboCup Laboratory, College of Engineering, University of South Florida, Tampa, FL
3 RoboCup Laboratory, IT & Eng, University of South Florida Polytechnic, Lakeland, FL

***Abstract*— In this paper we present the design and implementation of our Small Sized League RoboCup Team – Eagle Knights. We explain the three main components of our architecture: Vision System, AI System and Robots. Each element is an independent entity. The explanation focuses on the improvements made to our fifth generation of robots and how these changes interact with the rest of the elements in the team architecture.**

Index Terms: Small-size, robocup, autonomous, vision, architecture.

I. INTRODUCTION

RoboCup [1] is an international joint project to promote AI, robotics and related field. In the Small Size League, two teams of five robots up to 18 cm in diameter play soccer on a 4.05 by 6.05 m carpeted soccer field. Figure 1 shows a schematic diagram of the playing field and computer setup.

Aerial cameras send video signals to a vision system computer that computes robots and ball positioning on the field. This information is then passed to an AI system that produces control commands sent to the robots via wireless communication. Additional information is provided by a referee box indicating the state of the game.

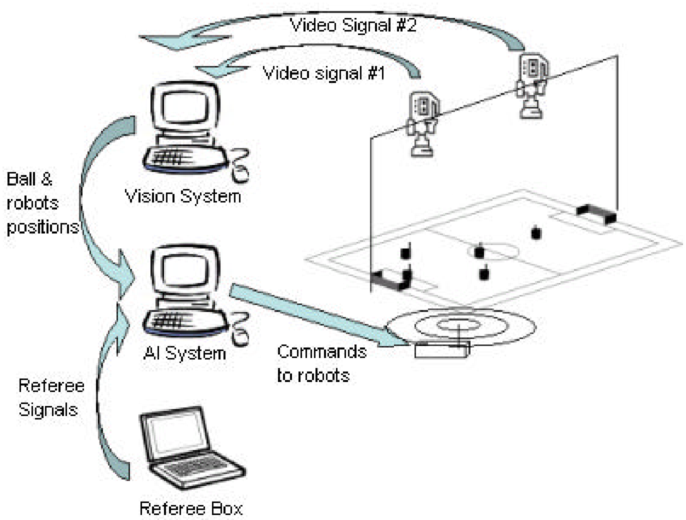

Fig. 1. Typical architecture of a SSL team

The robot architecture of our team in the Small Size League (SSL) consists of four main components: (1) vision system, (2) AI system, (3) robots and (4) referee:

1. The **vision system** digitally processes two video signals from the cameras mounted on top of the field. It computes the position of the ball and robots on

the field, including orientation of robots in our team. Resulting information is transmitted back to the AI system.

2. The **AI system** receives the information from the vision system and makes strategic decisions. The actions of the team are based in a set of roles (goalkeeper, defense, forward) that exhibit behaviors according to the current state of the game. To avoid collision with robots of the opposite team a geometrical exploring tree is used [1]. The decis ions are converted to commands that are sent back to the robots via a wireless link.
3. The **robots**, five in total, execute commands sent from the AI system by generating mechanical actions. This cycle is repeated 60 times per second.
4. The **referee** can communicate additional decisions (penalties, goal scored, start of the game, etc.) sending a set of predefined commands to the AI system through a serial link.

Figure 2 shows an Eagle Knights robot from 2008 taking part in RoboCup held in Suzhou, China. A new enhanced version is being developed for RoboCup 2009. In the next sections we describe in more detail each component of the architecture. The description focuses on the improvements made to our fifth generation of robots and the way these changes interact with the rest of the components in the system architecture. A prior generation can be found in [2].

Fig. 2. Small Size League 2008 Eagle Knights robot.

II. VISION SYSTEM

The vision system is the only source of feedback in the system architecture. If data returned by the vision system is inaccurate or incorrect the overall performance of the team will be severely affected. That is why the vision system should be robust enough to compensate for possible errors.

The main object characteristics used by the vision system are the colors defined in the rules of the SSL [3]. The ball is a standard orange golf ball. The robots of one team must have on top of them a 50 mm blue colored circle while the other team must have a yellow patch.

The main tasks of the vision system are:

- Capture video in real time from cameras mounted on top of the field.
- Recognize set of colors specified by the rules to objects of interest in the field (robots and ball).
- Identify, compute the orientation and position of robots in the team.
- Compute the position of robots of the opposite team.
- Track the objects in the field and get its moving vector.
- Transmit information to the AI system.
- Adapt to different light conditions (color calibration procedure).

We have implemented a number of algorithms for adaptability to different light conditions including the use of a neural network to classify camera image pixels to a discrete set of color classes that is robust under different light conditions [4][5].

The vision system consists of several modules; where each module is a functional block with a specific task as shown in Figure 3:

- CAPTURE MODULE. We use two AVT Guppy 1/3" progressive scan CCD cameras with an IEEE 1394 link. The frame capture is the AVT software that allows us to configure the resolution of the image, space color and frame rate. By default we capture RGB images with a 640x480 resolution at 60 fps.
- PREPROCESSING MODULE. The preprocessing module is used to improve the quality of the image, such as brightness, contrast, gamma, exposure, white balance, etc.
- OBJECT CALIBRATION MODULE. This module is a tool to establish the thresholds of each color component according to the space color defined for every object of interest (robots and ball). The calibration is done in HSV color space where selected thresholds are more robust to changes in field lighting and color changes than in previous versions. The HSV thresholds are transformed to RGB values to improve segmentation speed and avoid costly color space transformations.

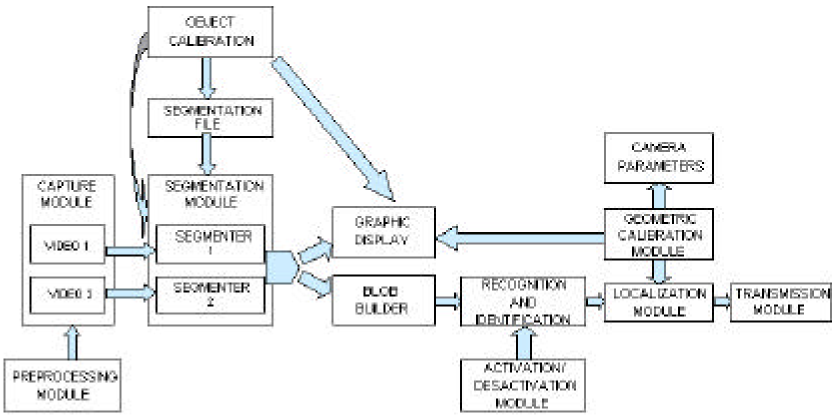

Fig. 3. Vision System Architecture.

- SEGMENTATION MODULE. This module assigns image pixels into object classes. The module consists of two segmenters, each one using separate thresholds values assigned for each camera for every object of interest. The HSV thresholds are mapped to a complete RGB color space cube in such a way that a 32 color segmentation can be done with one memory access.
- BLOB BUILDER MODULE. This module links segmented pixels with blobs. Before reaching this module the image is composed of separate pixels. Once a blob is constructed relevant information can be easily computed, such as color areas, centroids, bounding boxes, etc. A Run Length Encoding [6] and four neighbor search are computed. A joint list of blobs for the two cameras is generated for each color.
- ACTIVATION/DEACTIVATION MODULE. This module enables or disables the use of a particular robot. Sometimes a team can play with a smaller number of robots.
- RECOGNITION MODULE. This module identifies each object in the field selecting the color regions that better adjust to objects searched. Each object in the field is searched within a small square area. This square moves with each frame according to the prediction returned from the Kalman filter in the case of the ball and the opposite's robots. Our robots are tracked using their moving vector provided by the IA system. With this method the processing prediction time and the noise is reduced. We have specific selection criteria for every kind of object. For the ball we select the biggest orange blob (close to 85 pixels with an image resolution of 640x480). For the robots of the opposite team the selection criteria looks for blobs with corresponding central patch

color with an area closest to 115 pixels (the area of the patch is bigger than the ball) . For our own robots the procedure is similar to the one used for robots of the opposite team, although in addition to the central patch, a search for extra patches is necessary. The extra patches are employed for identification and orientation computation. In the case where one object is not found inside its square area a sub sample segmentation is made in the entire field in order to relocate all objects and resume the tracking step.

- GEOMETRIC CALIBRATION MODULE. This module computes the internal and external parameters of the cameras using the Tsai method [7]. These parameters are used to correct the distortion produced by the camera lenses and to convert camera coordinates to world coordinates and vice versa.
- LOCALIZATION MODULE. This module computes the position and orientation of objects in the field. It uses camera parameters obtained in the Geometric Calibration module to correct distortions in the image. Computation uses a set of pre-defined points, each one representing a well known landmark in the field (corners, midline point, etc). When a point appears in both camera images, corresponding coordinates are used to match and discard duplicate pixel data.
- KALMAN FILTER MODULE. This module consists on a Kalman Filter used to reduce the noise generated when the centroid of the blobs are calculated. The prediction of the following frame is used to define the position of the square where the ball and the opposite's robots will be searched on the next frame. It is also used to obtain the moving vector of each object to be used in the AI system.
- GRAPHIC DISPLAY MODULE. This module is responsible for displaying video images in the screen and for generating basic drawing functions such as lines, circles, etc.
- TRANSMISSION MODULE. This module consists of UDP network link used for communication between the vision system and the AI system that process on separate machines. The information transmitted from the AI system are robots positions and orientations, ball, opposite robots position, and the moving vector of all objects.

III. AI SYSTEM

The AI System comprises eight modules: Artificial Intelligence, Simulation System, Collision Detection, Transceiver Communication, Robot Control, User Interface, Vision System Communication and Game Control. The system includes a dynamics simulator in two and three dimensions in order to test system functions initially in the computer, including collision detection, AI and robot control.

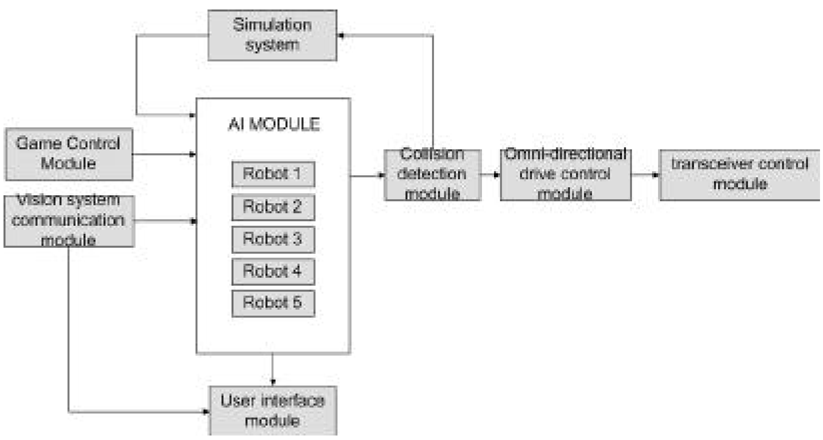

Fig. 4. Artificial Intelligence System Architecture.

The artificial intelligence system includes a main thread that loops and calls each of the different modules like shown in Figure 4. The detailed description for each of the modules is as follows:

- VISION SYSTEM COMMUNICATION MODULE. This module provides via packets the vision system commands representing the game scenario corresponding to robots and ball coordinates, angles and moving vector.
- GAME CONTROL MODULE. This module receives referee commands through a serial interface and returns the game state of the game. For 2008 RoboCup we succesfuly tested the Ethernet interface both modes are available.
- AI MODULE. This module receives the robots and ball positions, robots orientations, game state, robots roles, shooting direction and field configuration. With all this information the system calculates the future position and actions to take by each robot. The strategy used depends on the configuration of a tree that contains all the possible actions. The actions are classified according to their importance. For each node of the tree one or more evaluations are used. Each evaluation has a group of possible results associated with a particular score. During the loop of the program the tree is evaluated. The trajectory to take from the root to leaf (final action) depends on the highest score of the evaluation result on each level using a Best First Search method. Once the system has reached a final action like passing, shooting, or blocking, the robot moving vector, its linear and angular velocity and the use of the kicker and dribbler devices is defined. The robots also include a roll motion to coordinate them in joint actions. The robots are coordinated through different roles: Goalkeeper, Defense, First, Second and Third Forward. The final action for each role is defined using the position of the robots and the ball. With the implementation of the Kalman Filter in the vision system it is possible to know where the ball is moving and the defenses and goalkeeper can block its moving path. It is also used to intercept the ball from the other team, to give or receive passes and to shoot to goal. To define more efficient trajectories we implemented a spline based method to develop softer changing routes fo r every robot.
- USER INTERFACE MODULE. This module constantly displays important programmed using MFC control. The information includes the robot's position, orientation and speed, the game referee, the control commands to the robots and the configuration of the AI system. The spline trajectory module is implemented in this section to test the result of these new routes.
- SIMULATION SYSTEM. This module tests the operation of the artificial intelligence system without using the real vision system or the actual robots. It is useful to debug and test actions in the artificial intelligence module. The field is visualized with a two or a three dimension graphics OpenGL [8] interface.
- COLLISION DETECTION MODULE. This module receives actual and final position of the robots and generates a new path generated by a GET (geometrical exploring tree). With this method it is possible to easily avoid the opposite's robots and goals. The method works in real time (five robots in more than 60 fps). Because the space is constantly changing and the configuration is constantly changing we decided to use exploring trees. A GET constructs a tree during every process iteration. It can combine different types of obstacles with geometrical figures. The robots are represented like circles and the goal like rectangles. To generate the planner the tree starts with a root in the initial point and it is classified as an exploring node. The final point has already been defined by the AI module. The steps of the algorithm are as follows. First select an exploring node of the tree and try to reach the goal advancing a small predefined distance. If there are no obstacles interfering with the new point then the tree is extended and the new point replaces the last exploring node. In order to know if an obstacle interferes between the initial point and the goal the geometry must be considered. A vector "A" is defined from the initial point to the goal. In the case of a circular obstacle a possible

intersection between the circle and the vector “A” is calculated like shown in Figure 5. In case of an intersection the distance between the obstacle and the new extension is validated to be smaller than a radio “R”.

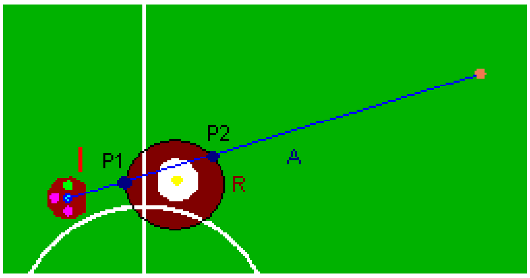

Fig. 5 Collision detection of a circular obstacle and the robot 1 trying to reach the ball.

In the example shown in Figure 5 there are two intersections: “P1” and “P2”. When the tree reaches the radio “R” , it will generate two possible routes one to each side of the obstacle. These two points are now considered as exploring nodes. While the exploring node can not freely reach the goal then it continues rounding the obstacle until there is not intersection or another obstacle is found. In this case a new obstacle is defined as the exploring node obstacle and it continues surrounding the new obstacle as shown in Figure 6.

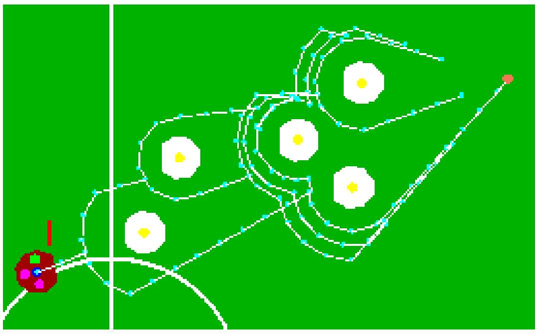

Fig 6. The tree finds the goal avoiding multiple obstacles.

In case of the goals the algorithm works similar but surrounding them with a rectangle. Once the tree has reached a point close enough to the goal the nearest path is chosen and the robot can be sent directly to the first point before intersection or in the direction of the next node towards the goal. This algorithm is repeated every cycle until the robot reaches its goal.

- TRANSCEIVER COMMUNICATION MODULE. This module builds the packets sent using our transceiver. The information sent to each robot is the moving vector and the angular speed of each robot.

IV. ROBOTS

We designed and built six omnidirectional robots. Each robot has five Faulhaber 2224P0212 motors with gearheads 14:1 (four motors for the wheels and one for the dribbler) [9], a low resistance solenoid, a DSP – Digital Signal Processor, a transceiver, a single printed circuit board and two Lithium Polymer batteries. The height of the robot is 140 mm, the maximum diameter of its projection to the ground is 178 mm, and the maximum percentage of ball coverage is 19%. The robots were designed and manufactured using a the e-Machine Workshop software. The robot receives commands from the AI system in the PC. It includes the following functional elements:

- ROBOT ID. Each robot incorporates an identification circuit manually setup with a dipswitch making it easy to modify the robot ID if necessary.
- DSP. The robot micro controller is a Texas Instruments TMS320LF2812 fixedpoint single chip DSP. This device offers low power and high-performance processing capabilities, optimized for digital motor and motion control. The DSP consists of six major blocks of logic: (1) External program and data memory, (2) I/O Interface, (3) Standard I/O, in addition to other modules not currently used in our design. The modules used are:
  External program and data memory. The RAM module is used in debugging the software with the Parallel Port JTAG Controller Interface.
  I/O Interface. It contains different kinds of pins: (i) Capture units used for capturing rising pulses generated by the motor encoders which can be used to measure speed and direction of the moving motor. (ii) PWM outputs having an associated compare unit. A periodic value is established to determine the size of the PWM, and the compare value is used to change the duty cycle.
  Standard I/O. It is used to read and write values for transceiver communication, motor, kicker and dribbler control.
- MOTOR CONTROL. The motor encoders generate a number of square pulses for each completed turn. Each pulse is captured using the DSP and the feedback speed is computed into the omni directional-module. To control the motors speed a PWM signal sent back to the motor. This information is obtained by the omni-directional module.
- WIRELESS COMMUNICATION. Wireless communication is controlled by two Radiometrix RPC-914/869-64 transceivers with radio frequency at either 914MHz or 869MHz. The transceiver module is a self-contained plug-in radio incorporating a 64kbit/s packet controller with a parallel port interface. Data is transferred between the RPC and the host (either DSP or PC) four bits at a time using a fully asynchronous protocol. The nibbles are always sent in pairs to form a byte, having the Least Significant Nibble (bits 0 to 3) transferred first, followed by the Most Significant Nibble (bits 4 to 7). Two pairs of handshake lines REQUEST & ACCEPT, control the flow of data in each direction.

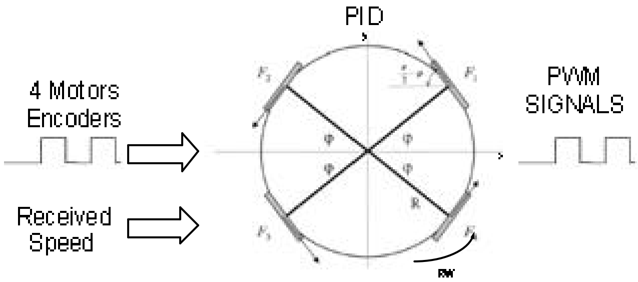

Fig. 7 Motor control using Pulse Width Modulation (PWM) and Proportional-Integral-Derivative controller (PID).

- OMNI-DIRECTIONAL DRIVE CONTROL MODULE. This module receives the movement vector including linear and angular velocities from the transceiver. To control the motor speeds two steps are completed: (i) Speeds are read from the motor enconders. The speed of each motor generates the actual linear and angular velocities of the robot. (ii) These velocities along with transceiver velocities are used as inputs to the PID algorithm. There are three independent PID algorithms in the process: the linear speed projection in the x and y coordinates of the robot and the angular velocity. The output of the PID is turned into speeds for each motor (using the motors geometry in the robot) and finally they are controlled to the correct speed with a PWM signal. An illustration of the problem is shown in Figure 7.
- KICKER CONTROL SYSTEM. Small Size soccer robots use different kicking designs to push the ball. We use a push type solenoid that kicks the

ball. Solenoid kicker system needs a high power supply. For size restrictions robots have six 7.4V/ 700mA batteries, equivalent to 31 Watts of power. With this amount of power we obtain less than the solenoid requires for a minimum performance. The main idea in power elevation is to store energy, then discharge it when solenoid is activated. To solve this power problem we implement a fourlayer system as follows:

Voltage transformation. The 14.8 dc voltage obtained from the batteries is increased using a (Pico Electronics IRF100S) dc-dc converter [10] to reach 110 volts. The output is used to charge up two 2200mF capacitors. The converter is controlled using a control pin of the DSP with a relay and a transistor. The robot can kick approximately every 10 seconds.

Discharge and solenoid activation. An infrared sensor system in the bottom of the robot senses if the robot has the ball. The DSP sends a high-level output bit when the robot is in score position. To discharge the capacitors into the solenoid, the Discharge layer uses both the DSP kick bit and the infrared ball detector output bit to discharge the capacitors. Because the capacitors charge level is very high, the robot dis charges it using a power MOSFET. A PWM signal is sent to the MOSFET to control the flow of current through it and thus controlling the intensity of the kick.

V. CONCLUSIONS

We presented a software and hardware overview of the SSL Eagle Knights team. The collision detection, the kicker device and the omni-directional robot control have been described in detail. Our team has been 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 last three RoboCup competitions: Osaka, Japan 2005, Bremen, Germany 2006, and Atlanta, USA 2007 and Suzhou, China 2008 improving our game and results each time. We had released to the public the Vision System and documentation of our electronics and DSP software to promote the participation of others teams in this initiative. More information can be found in http://robotica.itam.mx/.

ACKNOWLEDGMENTS

This work is supported by the "Asociación Mexicana de Cultura, A.C." in Mexico and by IT & Engineering at USF Polytechnic.

REFERENCES

[1] Elnagar, A., Basu, A. Local path planning in dynamic environments with uncertainty. Multisensor Fusion and Integration for Intelligent Systems, 1994.

[2] Martínez-Gómez, L.A, Moneo, F., Sotelo, D., Soto, M., Weitzenfeld, A., 2005, Design and Implementation of a Small Size RoboCup Soccer Team, Proc. 2nd IEEE-RAS Latin American Robotics Symposium, Sao Luis, Brasil, Sept 20-23.

[3] RoboCup SSL League, Laws of the F180 League 2009. http://small-size.informatik.uni-bremen.de/rules:main

[4] Martínez-Gómez, L.A., and Weitzenfeld, A., 2004, Real Time Vision System for a Small Size League Team, Proc. 1st IEEE-RAS Latin American Robotics Symposium, ITAM, Mexico City, October 28-29.

[5] Torres, E., and Weitzenfeld, A, 2008, RoboCup Small-Size League: Using Neural Networks to Learn Color Segmentation during Visual Processing, ENRI-LARS 2008, Oct 29-30, Salvador, Brasil.

[6] Abel, J, Run Length Encoding Data Compression, http://www.data-compression.info/Algorithms/RLE/, 2004

[7] Tsai, R.Y. A versatile camera calibration technique for high accuracy 3D machine vision using off-the-shell TV cameras and lenses. IEEE Journal of robotics and Automation, 1987.

[8] OpenGL Open Graphics Library. http://www.opengl.org/

[9] MicroMo, http://www.faulhaber-group.com/uploadpk/e_201_MIN.pdf

[10] Pico Electronics, http://www.picoelectronics.com/dcdclow/pe88_89.htm