# Eagle Knights 2006: Small Size League

Alfredo Weitzenfeld, Ernesto Torres, Aldo Faba, Christian Chavarría, Yuichi Murrieta, Leonardo Espinosa, Octavio Ponce, Monserrat Muñoz, Jesús Silva, Misael Soto, and Luis A. Martinez-Gomez

***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 and therefore the explanation focuses on the improvements made to our third 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 by 5.4 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 sate of the game.

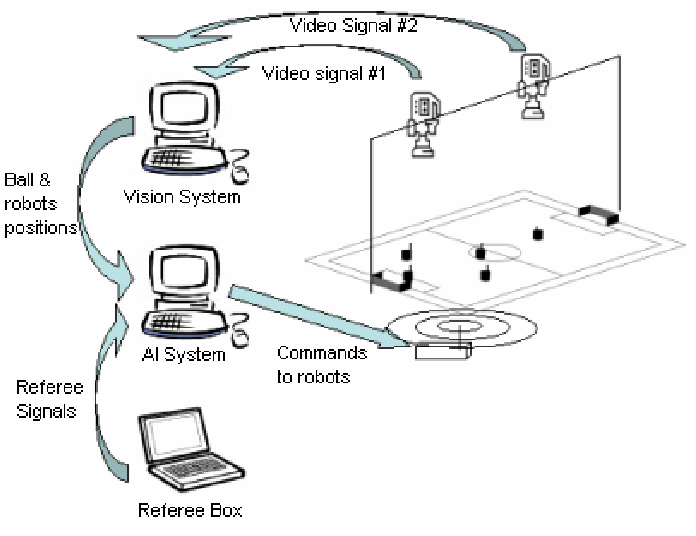


Fig. 1. Typical architecture of a SSL team

Thus, the robot architecture of the Eagle Knight 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 collision free tangential path is used [1]. The decisions 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 2006. A new enhanced version is being developed for RoboCup 2006. In the next sections we describe in more detail each component of the architecture. The explanation focuses on the improvements made to our third generation of robots [2] and the way these changes interact with the rest of the components in the system architecture.


Fig. 2. Small Size League 2006 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.

1-4244-0537-8/06/$20.00 ©2006 IEEE

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 in diameter blue colored circle patch while the other team must have a similarly sized 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 and compute orientation and position of robots in the team.
- Compute position of robots of the opposite team.
- Transmit information back to the AI system.
- Adapt to different light conditions (color calibration procedure).

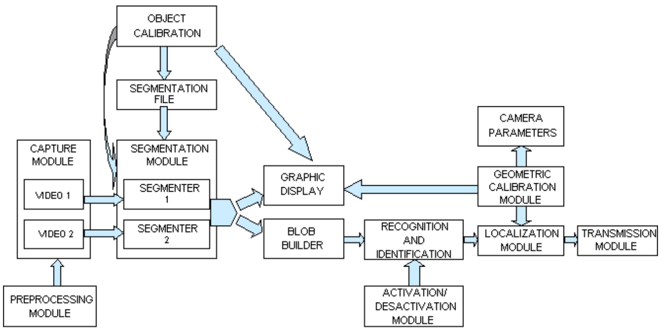


Fig. 3. Vision System Architecture.

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 done with MS DirectShow that allows us to configure the resolution of the image, space color and frame rate. By default we capture RGB images with a 720x480 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. Although this module is crucial for a proper operation, its impact has been reduced thanks to a more robust color calibration procedure.
- 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. This procedure is the most important improvement to our previous vision system [4] because it allows to accurate color separations. With this change the errors in the localization of the robots has been significantly reduced.
- SEGMENTATION MODULE. This module assigns each image pixel 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 just one access to memory.
- 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 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 selects the color regions that better adjust to objects searched. We have specific selection criteria for every kind of object. For the ball we select an orange blob that is closest to an area of 85 pixels (with an image resolution of 720x480). 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). The number of blobs selected is determined in the Activation/Deactivation module. 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.
- GEOMETRIC CALIBRATION MODULE. This module computes the internal and external parameters of the cameras using the Tsai method [5] in OpenCV [6]. These parameters are used to correct the distortion produced by the camera lenses.
- 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.
- 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. in the video image.
- 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 module builds a structure appropriate for data transmission. In practice the vision system can perform communication with two or more hosts

allowing for distributed AI system processing if necessary.

## III. AI SYSTEM

The AI System comprises eight modules: Artificial Intelligence, Simulation System, Collision Detection, Transceiver Communication, Omni-directional Drive Control, User Interface, Vision's System Communication and Game Control. This system is designed in a way that the user can test each module separately. It does not require that all modules be connected all the time to simplify testing. The system also includes a dynamics simulator in order to test system functions initially in the computer, including collision detection, AI and robot control.

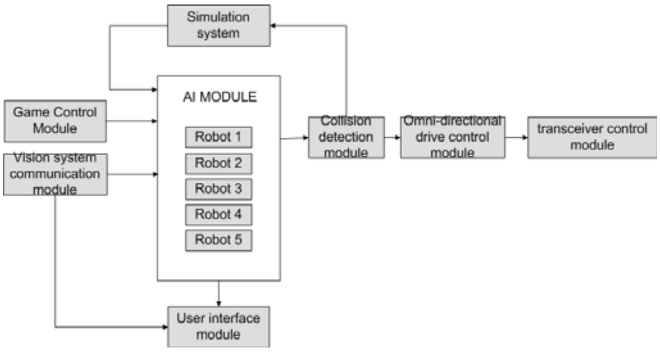


Fig. 4. Artificial Intelligence System Architecture.

The artificial intelligence system includes a main thread that loops and calls each of the different modules. The main system thread first communicates with the vision system to know position and orientation of robots In addition to ball position. The system then checks the game state that is controlled by the referee. Afterwards, the system calls the AI module function that returns the desired position for robot movement together with additional actions to take. Once movements are specified, the system computes collision avoidance trajectories to avoid bumping into other robots. The system then calculates the speed for the four wheels in each of the robots. Finally the system transmits via the transceiver communication packets corresponding to commands to take.

The detailed description for each of the modules shown in Figure 4 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, and robots angles.
- GAME CONTROL MODULE. This module receives referee commands through a serial interface and returns the game state of the game.
- 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 task of the goalie is to defend the goal. It follows a block path when the ball is far and kicks it when the ball is near. Moving is restricted to the area around the goal. The defense is in charge of helping the goalkeeper defend the goal against shots from far way while also including joint strategies with the three forwards. Defenders also clear the ball out when close to their own area and follow opposite robots to prevent a pass and shot to goal. The three forwards have a common objective but with different priority. They coordinate different types of passing and shooting and they move over the complete field. They can move in blocks when needed.
- USER INTERFACE MODULE. This module constantly displays information for each robot including positions, orientations, motor speeds, desired positions, ids, actions, game state, and referee commands. Robot positions, orientations, desired positions and actions are displayed graphically in the GUI programmed using OpenGL [7].
- 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.
- COLLISION DETECTION MODULE. This module receives the movement vector, the robots positions and orientations, ball position, and returns the new movement vector that avoids collisions while moving. Avoidance follows a tangential route around the obstacle.
- TRANSCEIVER COMMUNICATION MODULE. This module receives the speed for each of the robot motors and actions to take. This module builds the packets sent using our transceiver. It also checks communication is active at all times.
- OMNI-DIRECTIONAL DRIVE CONTROL MODULE. This module receives the movement vector including linear and angular velocities and returns the speed for each one of the four robot motors. Since the robot has four omnidirectional wheels, this module computes the speed for each motor in order to move in the desired direction.

## IV. ROBOTS

We design and built five omnidirectional robots. Each robot has five Faulhaber [8] 2224006SR motors with gearheads 30.7:1 (four motors for the wheels and one for the dribbler), 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 130 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 manufactured using a CNC ABC plastic machine at ITAM´s facilities.

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 fixed-point 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:
    1. External program and data memory. The RAM module is used in debugging the software with the Parallel Port JTAG Controller Interface.
    2. 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. (
    3. Standard I/O: 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. Feedback speed along with received speed from the transceiver is used as inputs to the PID algorithm in calculating an adjusted PWM signal sent back to the motor as shown in Figure 5.
- 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.

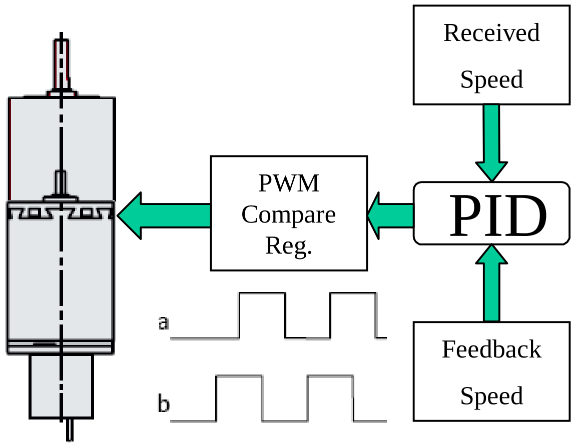


Fig. 5 Motor control using Pulse Width Modulation (PWM) and Proportional-Integral-Derivative controller (PID).

- KICKER CONTROL SYSTEM. Small Size soccer robots use different kicking designs to push the ball towards the opposite team goal. We use a push type solenoid that kicks the ball. Solenoid kicker system needs a high power supply. For size restrictions robots have only two 7.4V/2100mA 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 four-layer system as follows:
    1. Oscillation generation. Voltage transformers need an AC signal but robots batteries are DC. In order to do a voltage transformation, we need an oscillating voltage source. Oscillating voltage can be obtained with the DSP. We can generate a 3.3V PWM signal, 50% duty cycle signal at 100KHz.
    2. Voltage transformation. The oscillating voltage obtained from the DSP is a low power signal, to increase signal power we use an H transistor bridge, L298N. The two direction control bits in L298N are fed from the oscillating signal, each one is the inverse of the other. Enable bit is controlled by the DSP to avoid unnecessary voltage transformation. The input power signal in L298N is obtained from the two serial connected 7.4V/2100mA DC batteries. It represents a 15.8V input signal, so the output in L298N is a 29.6V peak to peak, 50%duty cycle at 100KHz signal. L298N output signal feeds a 24/120V voltage transformer. The voltage in the output transformer signal is a 148V peak to peak, 50% duty cycle at 100KHz signal.
    3. Charge accumulation. The charge in a capacitor is the number of electrons on the two plates. This involves the difference in the quantity of electrons and the unit of quantity is the coulomb,

$$Q = C*E \qquad (1)$$

where, Q = Coulombs, C = Capacitance in Farads, and E = Volts.

We use two 2200mF capacitors. Before charging them, the input signal needs to be rectified, where we use a full wave rectifier diode bridge across a DB106. The full charge capacitor time is 8 seconds approximately. It means solenoid can be activated every 8 seconds.

4. 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 discharges it using a power mosfet, a NTE 2388. 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 present a software and hardware overview of the SSL Eagle Knights team. The functional blocks of the software systems (Vision and AI) have been described in detail, together with the kicker control system. Our team has been the first Latin American team consistently obtaining top results in all its regional RoboCup participation, 2nd and 3rd place in US Open and 1st place in Latin American Open. 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 French-Mexican LAFMI, UC MEXUS CONACYT, CONACYT grant #42440, and “Asociación Mexicana de Cultura, A.C.” in Mexico.


## 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] Verret, Ball, Kiat Ng. Laws of the F180 League - Release 3.00a. http://www.itee.uq.edu.au/~wyeth/F180%20Rules/index.htm.

[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] 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.

[6] Open Computer Vision, Library, http://www.intel.com/research/mrl/research/opencv/

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

[8] MicroMo Product Catalog. http://www.faulhaber-group.com/n41799/n.html.