Our First Robot Soccer Player:Pele2

Machine Intelligence Lab
University of Florida -32611

Ashish Jain
Researcher
ashishj@mil.ufl.edu

A.Antonio Arroyo
Director
arroyo@mil.ufl.edu

Michael Nechyba
Assistant Professor
nechyba@mil.ufl.edu

Eric M. Schwartz
Assistant Director
ems@mil.ufl.edu

***Abstract*— This project evolved from the desire to create a robot which could be used in the international Robosoccer Competition. The goal is to build a team of robots for playing soccer on a specialized playing field. We have built a platform and the necessary hardware for communication and control. Since vision forms the primary sensor, we have written the necessary software to serve for ball detection, robot detection and the virtual boundary detection. Upon assembling our first prototype (Pele 2), we tested the vision and control algorithms. During the test the robot chased a ball while staying within the virtual boundary. The response of Pele2 to the vision-control algorithm was fairly smooth. The next steps are to add a kicking mechanism and build Pele3, a teammate for Pele2.**

I. Introduction

Pele2 is our first soccer playing robot. Pele2's control is handled offboard with a PC. The PC acts as a coach while the robot serves as a player. The primary sensor for Pele2's control system is a camera which captures real time images and sends them to the PC. These images are then used by the computer to detect the location of the robot and the ball. The computer then sends control signals to dictate the motion of the robot.

II. Platform

The entire platform design of Pele2 was done using Auto-Cad. This design was cut out, on a modified T-tech machine, from balsa wood. For Pele2 we also used a new injection molded wheel with urethane tires for better traction and control. The new wheels perform much better due to improved mounting on the servo head of the motors.

III. Hardware

We developed all the electronic hardware that went into Pele2. Protel was used for designing the circuit boards, and T-tech machine was used to cut them out. The electronic design for Pele2 consist of four boards: the controller, motor driver, transmitter and reciever.

A. Control Board

Pele2's control board was developed using the Atmel Mega8L microcontroller, which has one input capture, 3 I/O ports, 6 A/D converters , 2 Timers, and 2 PWM ports. The control board uses both of the PWM ports to signal the motor drivers. One I/O port is used for providing direction signals to the motor driver board. The other IO port is used for receiving 8 bit parallel signal from the receiver unit. Two A/D converters are reserved for future use. Atmel's ISP (similar to JTAG) was used to program the processor.

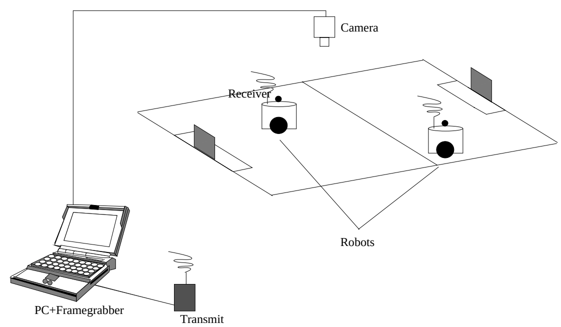

Fig. 1. System Design

B. Motor Driver Board

Since Pele2 required support for both forward and reverse actuation of the driving motors we tried using modified servos as motors. The servos we used had full 360 degree rotation, and the duty cycle decided the direction of the motion. To modify these servos we had to remove certain fixtures from the commercial servos. We had to calibrate the variable resistor, in the servos, to get the desired motion from them.

We noticed after long time of testing that servos started behaving differently from what was intended. Due to heating up of variable resistor and a continuous change in direction of motion the calibration of the motors changed while still in motion. This made controlling them very difficult as the algorithm for motion required a moderate level of precision.

In order to get the precise motion, we moved on to motor driver board. Since the DC motors we were using drew within an Ampere of current, we were able to design a simple motor driver board. The motor driver board consisted of a quad-optoisolator and UDN-2993. To control the driving motors, the motor driver takes four inputs from the control board, two for direction and two for PWM. However, we later found the chip UDN-2993 had been discontinued. As a result we switched to the more accessible Texas Instrument SN754410 for future robots.

C. Communication Boards

Pele2 uses a one way communication system, the ground station has a transmitter connected to a PC, while the receiving unit sits on top of Pele2. The transmitting unit consists of a TWS-434A RF Transmitter and HT640 encoder chip. The HT640 encoder takes 8 bit input from the PC's parallel port and sends it to the transmitter serially. The transmitter emits an RF signal at a frequency of 433.94 MHz. The data rate for the RF link is 3000 baud which is more than sufficient for Pele2's control signals, as well as for future additions.
The receiving unit on Pele2 consists of RWS-434RF receiver and HT648L decoder chip. The decoder chip decodes the receiver signal and sends it as an 8 bit parallel signal to Pele2's control board.

IV. Communication

A simple wireless protocol is set up between computer and Pele2 for sending control commands. The 8 bit control signal allows for 255 control commands. Since we wanted to extend the protocol for atleast four different robots that allows support for 63 possible decisions. We are currently making five decisions: turn left, move forward, move backward, turn right and stop.

V. Vision

The vision involves a camera, a framegrabber, a computer, and software for detection.

A. Devices

The framegrabber we are using is a PCI card made by Meteor that hooks up to the computer. It is capable of capturing images at 30 frames/sec. The camera is a commercially available Sharp VL-H860. It is an NTSC analog camera and sends the image to the framegrabber via an S-Video signal. The computer is an Intel based linux machine running at 800MHz.

B. Software

The software was created so that we can locate both the ball and Pele2 in real time. In order to keep the processing of images fast and robust, we kept the vision code as simple as possible. The placement of camera plays an important role in the way we process images. The most suitable position for camera is over the top of the field, so that camera captures the two-dimensional view of the field. However this was difficult for us to set up. We chose another option that meets our needs and was easier to implement. The camera was placed outside the field at an angle on a tripod to a significant height. This way the camera was focussed on the entire field. To locate Pele2 in the image captured from the camera we uniquely colored Pele2 so that software can detect it from its surroundings by placed two strips (white and pink) over the top of the robot. This scheme provided us with robot's coordinates, and orientation. We also implemented through software a virtual boundary system for the robot.

1) Virtual Boundary: Since we chose not to implement hardware(IR) based local obstacle detection, we created instead a software based virtual boundary. The virtual boundary was based on the camera's field of view. The virtual boundary could be very easily defined by the user through software. To implement it the user would capture the image in camera's view then draw a box around the playing field using a mouse. This information of the box is then fed into the vision-control software, which eventually uses it to contain Pele2. So whenever Pele2 comes in contact with the boundary it would turn 180 degress before taking any other control commands.

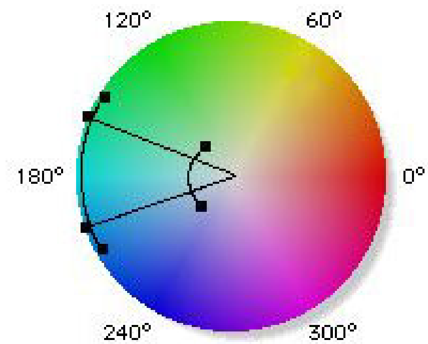

Fig. 2. Hue Variation

2) Color Detection: To detect color we used HSV color space, and chose the hue value for getting the color. The Hue portion of the color space is linear and provides us intensity invariant color detection with this method. Each color orange, green, white and pink can be reliably detected.

3) Location Detection: Since color detection resulted in a region of space and we needed a point for the control algorithm, we took the mean of the color region as the object.

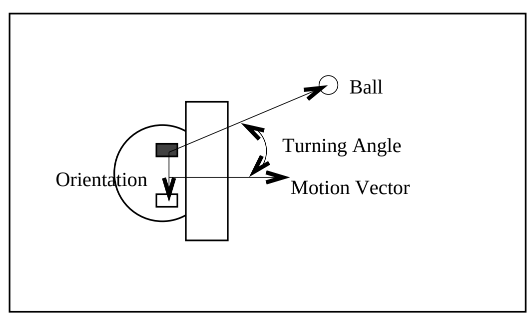

Fig. 3. Pele2's control criteria

4) Orientation: The orientation of robot with respect to the ball was calculated by taking the dot product of the ball vector and the motion vector. The motion vector is calculated as the vector perpendicular to the vector between the pink and white strips. The ball vector is a vector of the ball from the strip.

VI. Controller

A simple proportional feedback controller was implemented for Pele2. The camera provides the location of the ball and the robot, and the controller tries to

- Minimize distance between Pele2 and ball
- Reduce the angle between the ball and motion vector

The distance between Pele2 and the ball is the magnitude of ball vector. The angle between ball vector and motion vector is calculated by taking dot product of the two vectors to find the cosine of angle, cross product to find the sine of angle and then using arctangent to get the angle. At certain positions Pele2's has intermittent uncontrollable behavior, which is possible to be corrected using proportional differential controller.

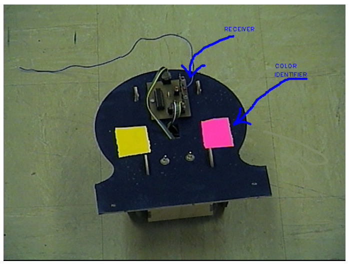

Fig. 4. Pele2

VII. Conclusion

Pele2 passed its initial tests by staying within the boundary that was created for it and chasing the ball wherever it was placed within the boundary. Now that we have sufficient control of Pele2 we plan to implement the kicking apparatus.

References

[1] Veloso M.,Stone P. and Han K. CMUnited-97:RoboCup-97 Small-Robot World Champion Team, AI Mag. 19, 3(1998), 61-69.

[2] Veloso M.,Stone P. and Han K. CMUnited-97 Robotic Soccer Team: Perception and Multiagent Control, Second International Conference on Autonomous Agents, 1998

[3] A.K.Macworth. On seeing robots. In A. Basu and X. Li, editors, Computer Vision: Systems, Theory, and Applications, pages 1-13. World Scientific Press, Singapore, 1993