An Autonomous Pong Playing Robot

Uriel Rodriguez uriel@ieee.ece.ufl.edu
John Buisson john@ieee.ece.ufl.edu
Lynette Miller lynette@ieee.ece.ufl.edu
Rolando Panez rolando@ieee.ece.ufl.edu
Eric Schwartz ems@mil.ufl.edu

Machine Intelligence Laboratory
Department of Electrical and Computer Engineering
University of Florida, Gainesville, FL 32611-6200

Abstract

Pong was one of the first video games created and was recreated using robots in the IEEE SoutheastCon 2002 Hardware Competition. This paper discusses the development of an autonomous robot designed to play old-fashioned Pong that placed first in the competition. Starting from the video camera feed, this paper will talk about each component of the robot from the software to the motor controller and finishing with the paddle.

1. Introduction

Every year the IEEE Region 3 holds the SoutheastCon Conference in which different universities can enter a Hardware Competition [1]. The goal of this year's project is to design an autonomous robot that plays old-fashioned Pong on a four-foot by eight-foot table against another robot from a competing school. The objective is to score more points than the opponent by getting more balls into their scoring bin. The robots volley the ball back and forth until one of them misses. The table's highest point is in the center and declines toward either side. This table design places bounds on the minimum and maximum speeds at which an opponent can return the ball across the table. If the ball is hit too fast and does not land in the scoring bin, then neither team scores a point. The match is played for five minutes or until the ten ball supply is exhausted. If the score is tied at the end of regular play, then the match will go into sudden death where an extra ball is added every two minutes. The first team to score in sudden death is declared the winner. Each robot "sees" using an EverFocus video camera that is mounted such that it looks down on the playing field of the table. The robots need to prevent balls from entering the scoring bin by blocking incoming balls and returning them to the other side of the table using an eight-inch paddle. This task is broken into three separate components consisting of the Software Design, the Motor Control System and the Paddle Design.

2. Software Design

The software is broken into three distinct sections. It must track a ball within a frame, predict where the paddle should move and communicate this location to the motor controller.

2.1 Ball Tracking

The most important property of ball tracking code is to ensure that it works under a variety of conditions. For this reason, the tracking software does not simply look for a white cluster of pixels within the image it receives from the video camera. Instead, the software first takes a picture of the search area without any objects and stores it as the Reference Image. Since the camera records in black-and-white, shade values of the reference image's pixels subtract from those of each image as the program receives frames from the camera. A simple filter determines if differences in the two pictures are noise or a ball, and a collection of all balls present in the frame passes from the tracking system to the prediction system. This technique allows both shadows and bright spots to appear on the table's surface without mistaking them for balls as long as they are constant. It also allows ball tracking when a reflection on the table is a brighter shade of white than the ball.

2.2 Ball Prediction

The ball prediction software receives the list of all balls present on the table and decides which ball it will track by finding the ball closest to the paddle. This is necessary due to the judge adding an extra ball to the table every two minutes if the match enters the sudden death phase. At this point, two approaches seemed logical in determining where to move the paddle in order to intercept the ball.

2.2.1 Y-Tracking

The easiest and most obvious solution to predicting where the paddle will move is to use Y-Prediction. This method simply sends the paddle to the y-location that the ball currently occupies, causing the paddle to follow the ball along the table as seen in Figure 1. Although a primitive technique, this first implementation has the desirable property that the paddle is in motion when it makes contact with the ball. This motion causes the ball returns to have more drastic angles as well as increased speed. However, the flaws of this technique far outweigh this benefit since bounces against the wall cause the paddle to slam into the side of the table. The paddle also tends to miss bounces using this technique due to the paddle always racing to catch up to the ball.

2.2.2 Pure Prediction

This technique sends the current ball position as well as the previous ball position to a recursive function. The function first calculates the equation of the line from the points it receives. Using simple geometry, if this line intersects the line x=xmax between the points (xmax,ymin) and (xmax,ymax), as shown in Figure 2, then the paddle is told to move to that point of intersection and exit the recursion. Otherwise, if the line will intersect the lines y=ymax or y=ymin between xmin and xmax then the function calls itself with the first argument being the point of intersection. The second value is the reflection of the second argument passed in over the line x=(x-value of the point of intersection). The result of this prediction was very precise and the paddle was moved to a point that it would intercept the ball approximately 92% of the time.

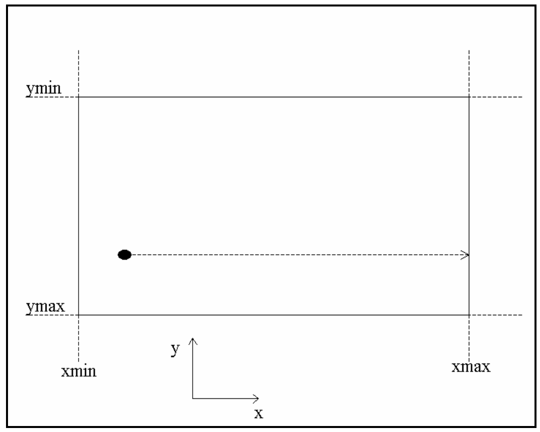

Figure 1: Y-Prediction graphic

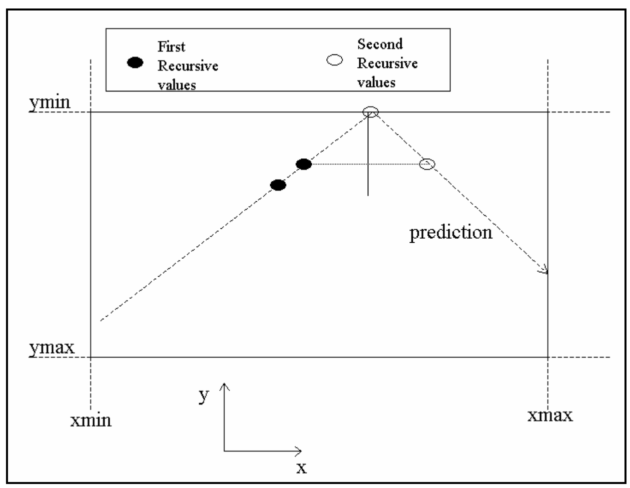

Figure 2: Pure Prediction

2.2.3 Final Result

The increased shot angle of Y-Prediction provides a better offense then the Pure Prediction. The new algorithm morphs these two techniques by using both algorithms at the same time. First, the prediction technique calculates where the ball will eventually cross the goal line. Next, two limits are calculated

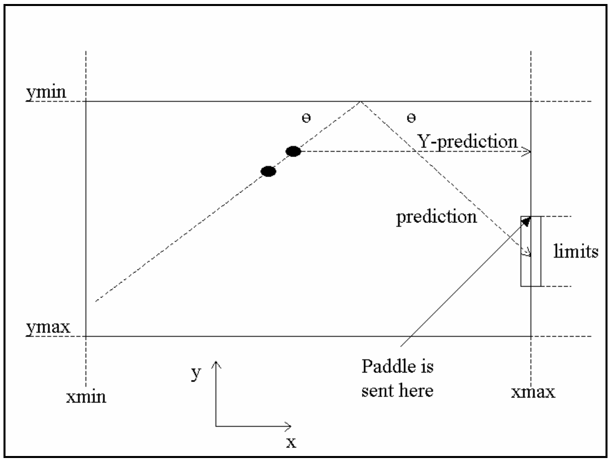

Figure 3: Final Prediction Algorithm

based on this prediction by adding and subtracting three inches worth of pixels from this point. Y-Prediction is then used, but bound between these two limits, as shown in Figure 3. The result is that the paddle can never move more than three inches outside of the prediction, but when the ball strikes the paddle itself, the paddle is moving slightly. By doing these calculations on each frame, the software is accurate and still returns the ball at steeper angles.

2.3 Communication to Motor Control System

Once the prediction program gets the pixel location it wishes to send to the controller, it converts the pixels to inches, and inches to sectors. The program breaks the table into 120 sectors to allow the software and motor control system to have the same units during communication. The software performs the communication with the motor controller through the parallel port of the computer. When the software first starts, a child process runs in the background waiting for its parent to send it data through a pipe. The child receives the data and extracts the sector it needs to send to the controller via the parallel port. This method of communication is used so that the software runs at 30 frames-per-second to allow the tracking and prediction software to continue executing while the child is busy performing the communication.

3. Motor Control System

The functionality of the motor control system includes receiving data from the software and moving the paddle to the location it receives. This system's components include a power supply, a motor driver, a feedback system (shaft encoder and CPLD), and a control management system (68HC12 microcontroller). The interfacing of these systems is shown in Figure 4.

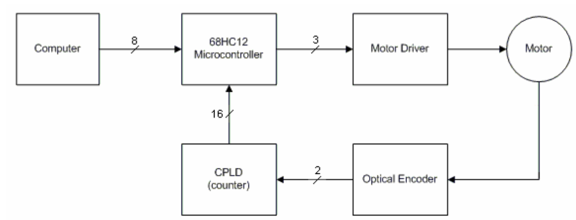

Figure 4: Motor Control System interface.

3.1 Power Supply

The power supply consists of two 12 volt batteries connected in series to provide 24 volts for powering the motor. The 68HC12 uses a 5 volts DC power supply, and the CPLD requires by a 7 volt DC power supply. The shaft encoder and the motor driver share the 5 volt power supply with the 68HC12.

3.2 Motor Driver

The motor driver consists of a standard H-Bridge with logic control. The logic control uses two control lines to select direction along with an enable line which uses a PWM signal to control the speed of the motor. The 68HC12 generates these to actuate the motor.

3.3 Feedback System

The shaft encoder consists of a 2" diameter optical disk with a resolution of 500 lines per revolution and an HEDS optical encoder module. The encoder module outputs two signals that determine the direction and position of the motor. The MAX7000S CPLD contains an up/down counter, that uses this data to update the position of the paddle. Adding a debouncing circuit in the CPLD filters the noise in the signals from the shaft encoder, and a latch component holds a valid current position for the 68HC12 to read.

3.4 Control System Interface

The 68HC12 microcontroller interfaces all other components in the motor control system. This microcontroller accepts data from the software and moves the paddle to the desired location. A proportional control scheme allows for high speed travel of the paddle to the desired location. A derivative compensator can be used to reduce overshoot. However, due to constant change in destination, the compensator will hinder overall speed, and thus, not used in the final implementation.

4. Platform

The platform consists of two wooden stands, placed on either side of the playing field, that support a steel rod used to guide the paddle. The paddle is mounted on an aluminum flanged pillow block that slides along the rod. A spring and collar, located on either end of the rod, protect the paddle from running into the sides of the playing field as shown in Figure 5. A lightweight, #35 ANSI acetal chain attaches to the pillow block with the use of tie straps. A motor equipped with a steel roller chain sprocket drives the chain. On the other side of the motor, a roller chain idler sprocket keeps the chain in place. The motor control system sits on top of one of the stands. The complete platform is shown in Figure 6.

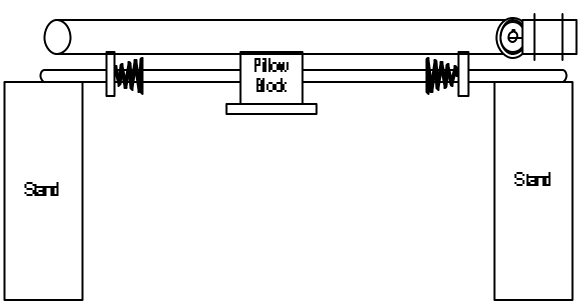

Figure 5: Block diagram of platform

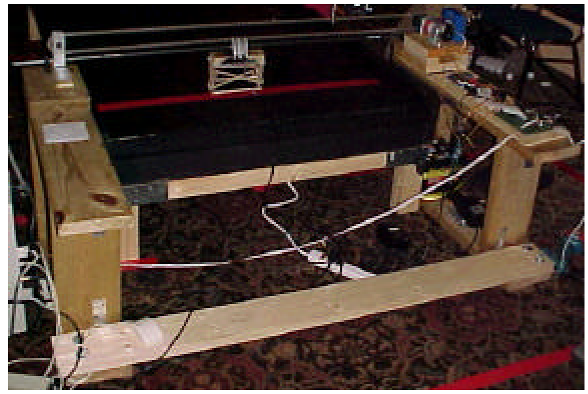

Figure 6: Actual platform setup

5. Paddle Design

The requirements of the paddle as set by the competition rules are:

- It cannot trap the ball
- It cannot alter the momentum of the ball as long as the ball is in the playing field
- The paddle cannot be wider than eight inches and cannot cross into the playing field or scoring bin (must stay within the paddle zone).

5.1 First Ideas

The original idea was to create a funnel shape object that would trap and channel the ball to the hitting mechanism. A break beam would be used to trigger the hitting mechanism which would consist of a powerful push solenoid once the ball was in place as shown in Figure 7. A prototype was built using the funnel shape, but after running a few tests with different types of surfaces and angles, the idea was discarded due to the inconsistency of the ball making its way down to the hitting mechanism.

5.2 First Implementation

The first paddle implementation was mounted vertically, hanging from the rod, and swung from a pivot point, a hinge, on the top of the paddle. A push solenoid was used to swing the paddle as shown in Figure 8 However, this paddle design presented the new problem of

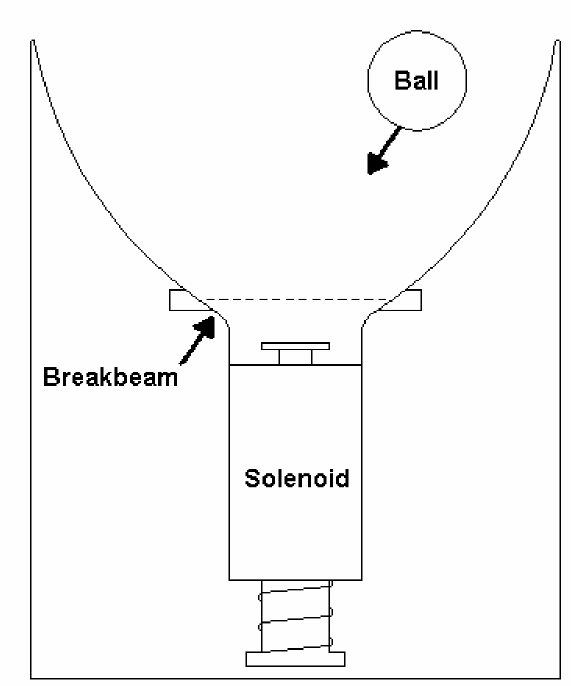

Figure 7: First Paddle Concept

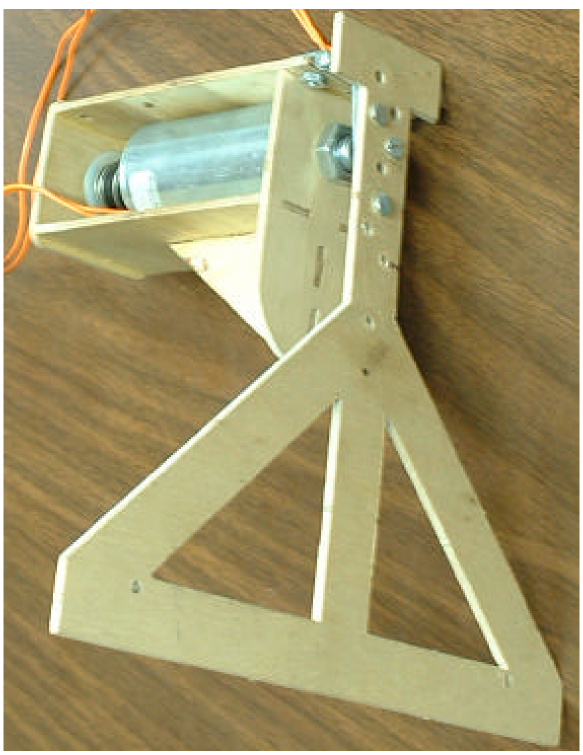

Figure 8: First paddle prototype

determining when to trigger the paddle in order to strike the ball efficiently. One approach to this problem is the implementation of a break beam. Adding a break beam would decrease the paddle size by about 1/2" on each side. Another approach is to use the tracking software to calculate the velocity of the ball and predict when the ball will strike the paddle. A signal from the computer triggers the solenoid. The problem with this approach is calculating the time that the ball contacts with the paddle. If the calculation is inaccurate, this will result in a poor return.

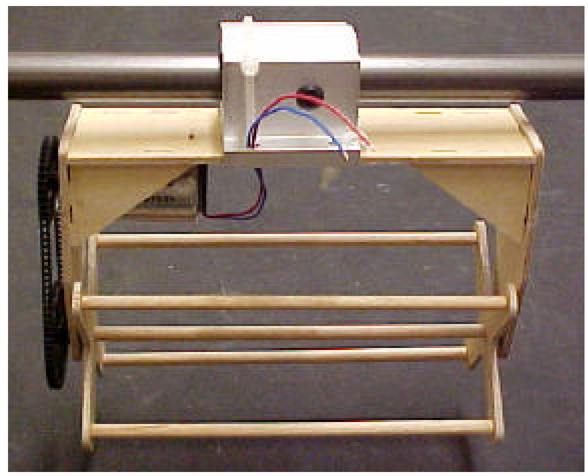

Figure 9: Straight-rod paddle

5.3 Second Implementation

The new design constraint consists of having the ability to hit the ball consistently without any triggering mechanism. To accomplish this task, the new paddle rotates four rods around a center rod in such a way that when the bottom half of the paddle hits the ball. A 5 1/4" floppy drive motor spins the paddle as shown in Figure 9. The ball was consistently hit back across the table whenever it made contact with the paddle.

5.4 Final Version

Replacing the straight rods with angled rods allows the paddle to return the ball in random directions. The rods are alternately angled to the left and the right so that two rods hit towards the left and two rods hit towards the right as shown in Figure 10. The paddle calibrates in two ways: number of revolutions of the paddle and the height of the paddle relative to the table. The best results occur when gears reduce the speed by a factor of two. The paddle can be moved up or down by adjusting the screws to change the point of contact in which the paddle strikes the ball.

Figure 10: Angled-rod paddle

6. Conclusion

The concept of Pong is hardly a new one, but the task of moving this classic game outside of a virtual venue proved to be a challenge. These many challenges were overcome by keeping the physical portions of the robot simple and concentrating on the robot's intelligence. With the strategies mentioned in this paper, the University of Florida IEEE Team entered the SoutheastCon 2002 Hardware Competition [1] and placed first among 26 schools in the southeast [2]. This project shows that a winning pong-playing robot could be built at minimal cost.

7. References

[1] IEEE SoutheastCon 2002 Hardware Competition website [online]. URL: http://www.ee.sc.edu/orgs/Secon2002/Rules. Retrieved on April 9, 2002.

[2] University of Florida IEEE SoutheastCon 2002 Tribute website [online]. URL: http://ieee.ece.ufl.edu/secon2002. Retrieved on April 9, 2002.