Design and implementation of a Vision-Based Air Hockey robot

presented
by
Abdelrahman Baba, Maysem Mousa
to
The Department of Computer Engineering

in partial fulfillment of the requirements
for the degree of
Bachelor of Science(B.Sc)
in the subject of
Computer Engineering

An-Najah National University
Nablus, Palestine

2022

ACKNOWLEDGMENT

Our deepest and most sincere gratitude goes out to our families, whose continued support played a crucial role and provided invaluable assistance for us and our work on this project as well as our friends, who were there for us when we felt overwhelmed and behind schedule, just to lift us up and get us back on track when we felt overwhelmed and behind schedule.

As well, we are deeply grateful to every single member of our department's family, from teaching assistants to professors, for their contributions and teachings. Their knowledge is priceless, and we will be eternally grateful to them for giving it to us with no expectation of return.

Last but not least, we would like to extend our sincere thanks to our colleagues. Having the opportunity to work with such passionate and great minds was inspiring, and we appreciate each and every contribution they made.

DISCLAIMER STATEMENT

This report was written by students: Abdelrahman Baba and Maysem Mousa at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Design and implementation of a Vision-Based Air Hockey robot

Abstract

In the past decade, robotics revolution has garnered significant attention by researchers and companies because of their ability to perfume human-tasks with better efficiency and higher speed. because of this rapid acceleration in the industry, new robots were developed to do non-critical applications like playing games against humans. Air hockey is one of the most well-known arcade table games, in which, two players try to strike a buck into the opponent's goal. This project proposes the design and implementation of Vision-Based Air-Hooky playing robot. the robot can move in two dimensions which enables it to do more complex movements which was impossible to do in a single dimension. also it has two playing strategies (defense and attack) to give the player more exciting human-like game play. The table used for this project was designed and built from scratch to meet this project needs.The plastic parts were 3d-printed and can be used to any CNC-based system. The robot itself can be detached from the table without much effort so it can be used as a normal air-hockey table. wireless scoreboard system using ESP module is also available to make it easier for arcade halls to adapt this project as a standard arcade game.a spacial camera with high FPS rate was used to achieve the high speed response needed by this real-time application. an Arduino micro-controller was used to process the output of the camera and handle the movement of the motors.

Contents

Listing of figures

1

Introduction

In this era of Technological advancement.Robots has shown the ability to replace humans in every field you can think of. starting with routine-based tasks like sorting ware-horses and factories assembly lines to more critical applications like dissembling bombs and fire fighting. in fact robots are more accurate than humans and can do critical tasks with zero margin error and with less needs. robots don't have any biological needs nor emotions

which can effect their performance and decisions. instead, they do what what you tell them to do exactly.

The first robot to play games in the computer era was IBM's computer named "Deep-blue"[3]. Deepblue was put against the chess world champion Garry Kasparov. Kasparov did manage to beat the first version of Deepblue. but IBM Engineers did many improvements to Deepblue algorithms and asked for a rematch. this time deepblue won the match. It was the first time in history where machine power overcame man-power. After this breaking event robots industry developed rapidly. robots became more powerful more solid more precises and more intelligent and started to take place in all fields.

Because of how easy building a robot can be these days, we started to see robots excessively in the entertainment industry[4]. almost every town in the world has an arcade hall filled with hundreds of robots that you can play against. robot

In this project we developed a robot that can play air hockey table game. this game of skill requires a high level of artificial intelligence for a robot to master. the robot is able to process the table via a high-tech high-speed camera and process it in real time in order to move the motors of the robot in the correct speed and correct position to be able to defense against the buck.

The main operation of the robot goes into phases, with the first is to detect the position of the buck and the second is to serially transfer the data to the micro-controller. The micro-controller process the data and decides the best move to do in order to achieve a human-like game-play with as little delay as possible.

This report is a deep dive into the design process of the robot along with the parts and technologies we used, also the we mentioned all the challenges we faced during this project

and how did we manged to overcome them. finally, a discussion and conclusion of the things we had done and things we aspire to achieve in future work.

2

LITERATURE REVIEW

2.0.1 Motivation

Through automation, processes can be made more efficient while reducing labor costs and time. A whole new level of robotics technology has been achieved as a result of scientific advancements in general. Additionally, engineers have been challenged to overcome technological obstacles by improving these robots. Creating robots that are as human-like as possi-

ble is a challenge for many robot designers. Because of our physical and mental limitations, humans rarely respond to extreme situations optimally in extreme situations. Therefore, it becomes necessary to develop a robot that has the power robots but also takes advantage of human[2] intelligence.

2.0.2 CONSTRAINTS, STANDARDS, AND COURSE WORK

The one major challenge we overcome is the speed of the motors. stepper motors are great for precise moves but its very limited when it comes to speed. this real time application requires a fast processing and response to work probably. to solve this we did calculations and a prediction model to overcome the motors slow response.

3

METHODOLOGY

In this research we developed a robot that is capable of playing human-intended games. the robot was designed to perfume complex moves to give the opponent the feeling of playing a against a human.

Throughout this chapter, we will discuss our system, how we built it, how its different components were connected, and how we achieved real-time response and maximum accu-

racy using it.

3.0.1 System Architecture

In this section, the complete architecture of our system is described in details along with design choices.

The project is divided into two main parts: the mechanical structure and the robot.each part is described in details in the coming subsections.

Mechanical Structure

For the table. we used MDF wood sheets. The table main body consists of a two layers of 60cm X 100cm thin sheet of wood. the two layers is separated using a wooden frame. the height of the frame is 3cm. the top layer which represents the table surface has holes all over it. these holes will allow air to flow to the top surface. the air reduces fraction between the table surface and the buck to increase the smoothness. two fans is mounted at the bottom layer to push air to the upper layer.

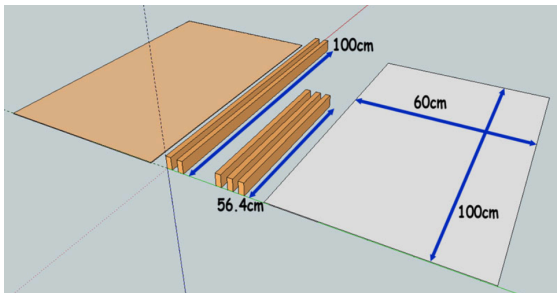


Figure 3.1: Table dimensions

For the robot. we built a two dimensions Coordinate system. this system uses two motors to move the robot in the y-axis and one motor for the x-axis. two motors is needed to

handle the extra weight of the x-axis motor. Timing belts and steel rods are used to frame the main robot body.

The plastic parts used to mount the timing belt and the rods are publicly and was 3D printed. the filament percent was calculated based on the needed force for each part to minimize printing cost. These calculations are out of this project's scope.

Figure 3.2: End pulley

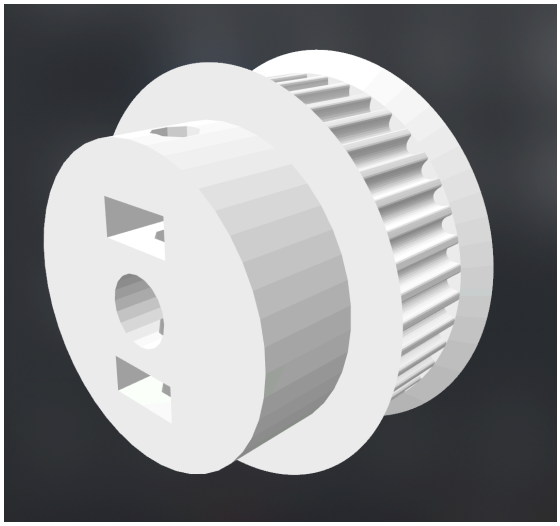


Figure 3.3: Timing pulley

Figure 3.4: X-Axis Carriage

Figure 3.5: Pusher part I

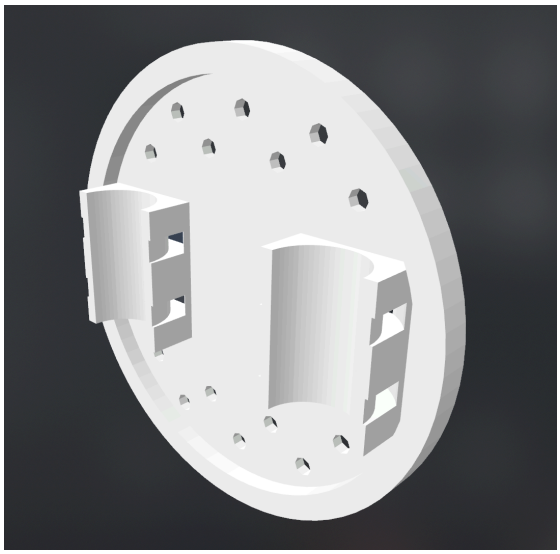


Figure 3.6: Pusher part 2

Figure 3.7: X-Axis Motor Mount

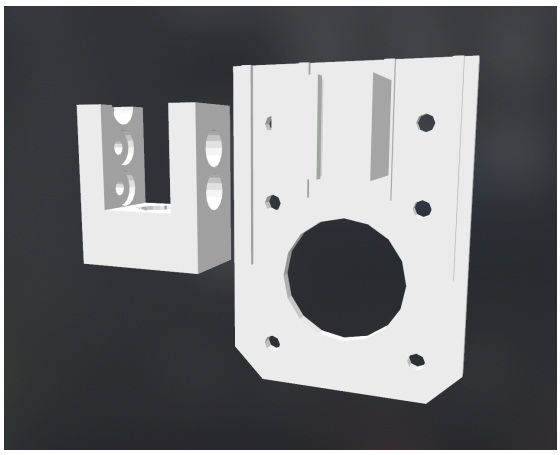


Figure 3.8: Y-Axis Motor Mount

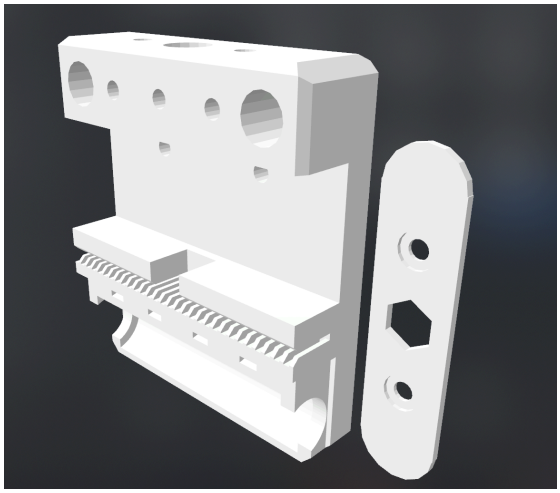


Figure 3.9: Y-Axis Carriage

Figure 3.10: The Buck

The Robot

The Robot has four main components: the vision system, the processing unit and the motors. the data flows from the camera readings to the process unit.based on many factors

including the puck speed and direction the motors then moves to tackle the puck.

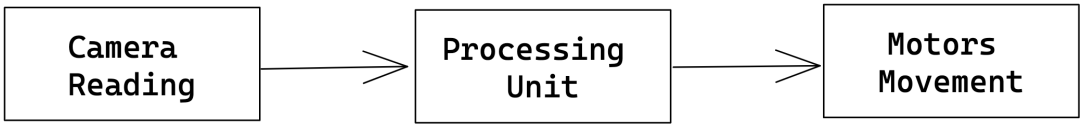


Figure 3.11: Data flow in the system

The Camera

The robot relays heavily on a high-performance camera that acts as complete vision system to the robot. we did extensive market research to find the best camera to meet our needs. the main criteria were as follows:

- The camera should have high frame rate. higher than 60 fps to be able to detect the exact position of the puck with high accuracy and low delay.
- The market price should be as low as possible to meet our budget.
- Some cameras has built-in sensors to farther enhance color and object detection like Microsoft Kienect[8].
- Some cameras do initial image processing which make the images more optimized for and easier to deal with.

One of the best choices available is pixy2 cam[1]. This camera satisfies all the mentioned criteria and was chosen for this project. It has multiple advanced sensors that enhance color detection functionality also it has built-in processor that do some initial processing which farther helped to improve the overall performance.

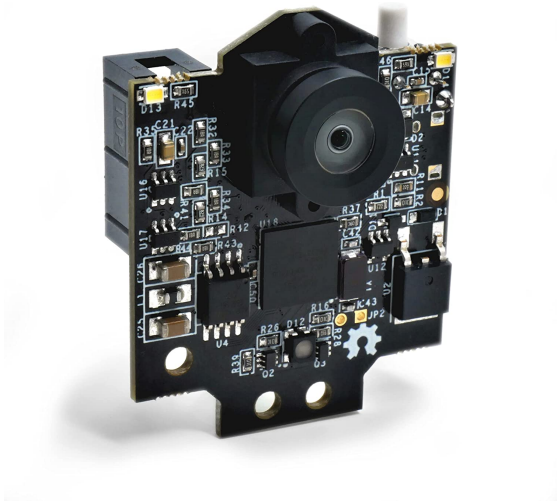


Figure 3.12: Pixy2 Cam

The camera has many of libraries which made it easier to interface with many microcontrollers and it supports a lot of protocols. we interfaced the camera with Arduino UNO in our case.

The camera is connected with the arduino using the serial port and it sends data at 1 Mbits/second.

with almost perfect lighting condition, the camera managed to locate the puck very preciously at around 60 frame per second.

The Processing unit

We went with the Arduino Uno for this project. it can interface easily with the camera our decision is based on many factors which are:

- The price is fair for the functionality it provides.
- Easier to program than other devices like Raspberry pi but, and can still meet our

needs.

- Very compact in size.
- Can be connected with 5v directly as it has internal voltage regulator.

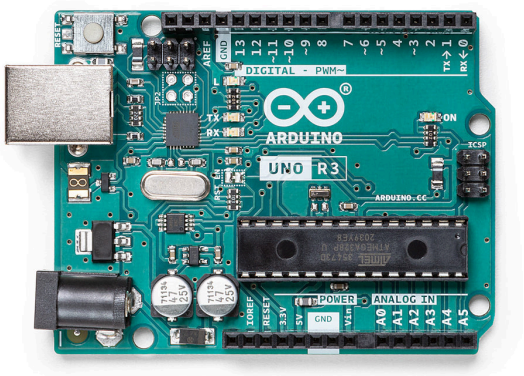


Figure 3.13: Arduino Uno Micro controller

The motors are controlled by the Arduino. its responsible for processing the puck position planning an attack or defensing the goal. the Arduino code that does all this is provided in appendix A the algorithm was developed by the researchers based on many factors and after many trial and error.

The calculations that determine the position of the robot are based on many factors including:

- The speed of the puck.
- The direction of the puck
- The current location of the puck (robot side or player side).

- Whether the puck is stationary or moving.
- the current position of the robot.

Many other corner cases are covered including whether the puck will hit one of the walls and reflect, the distance and the angle of the reflection is taken into consideration. Also if the camera failed to detect the puck due to some errors the robot do some movements to defence the goal.

The Motor Movement

There are many motors type in the market. we went with NEMA 17 stepper motors for many reasons, including:

- Its industry-standard motor type when it comes to CNC systems. because it gives high precision movement and can has a very high torque compared to DC and servo motors.
- NEMA 17 family provides enough torque and high speed at the same time which is optimal for our case.
- Can be accelerated and decelerated i=easily without hard calculation while also maintaining an accurate distance as its only controlled by a step pin w, unlike dc motors which can lose high accuracy due to acceleration and require hard calculations.

The motors are interfaced using A4988[7] Stepper motor driver. this driver can provide up to 2 amp for each coil without over-heating or causing any problems. our motors are rated

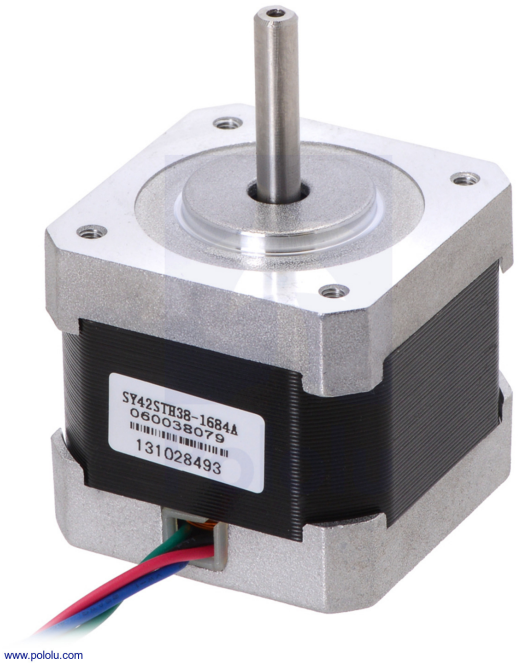


Figure 3.14: NEMA 17 Stepper Motor

1.7 amp so the driver is sufficient. it provides an easy interface which allows the motor to be controlled using step and direction pins only.

An open source Arduino library called FlexyStepper was used to handle the acceleration of the motors. it also provides some useful functions to move the motors in different units.

3.0.2 The implementation

The table is divided in half. CNC system gives the robot arm free moving ability over the whole xy plane. this measures to 40cm on the y axis and 57 cm on the x-axis.

The CNC system used involves two motors that manages the y axis movement. the x axis is controlled by one motor only. this simple architecture has many advantages over some other famous ones like the H bot systems including:

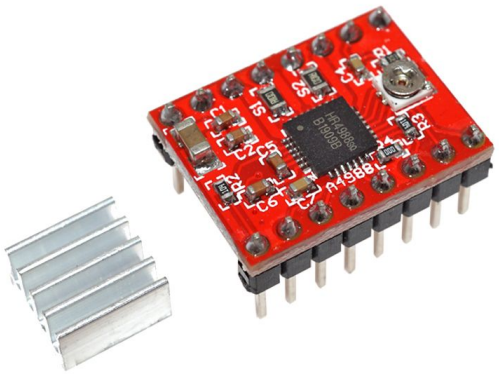


Figure 3.15: A4988 Stepper Motor driver

- Timing belt is more protected and less likely to break because its shaped into simple one direction loops, unlike H bot which has single long H shaped loop which can break easily if was over tighten or built incorrectly.
- The motors are easier to move and control because each axis is secreted and controlled by one motor only in a linear motion.
- The H-bot configuration uses a single long timing belt which can be hard to find. shorter length belts are more available on the market.

On the other hand it has some downsides. including the need for an extra motor to distribute the weight of the x axis motor which is moving over the y axis, unlike the h bot configuration where the motors are in fixed places.

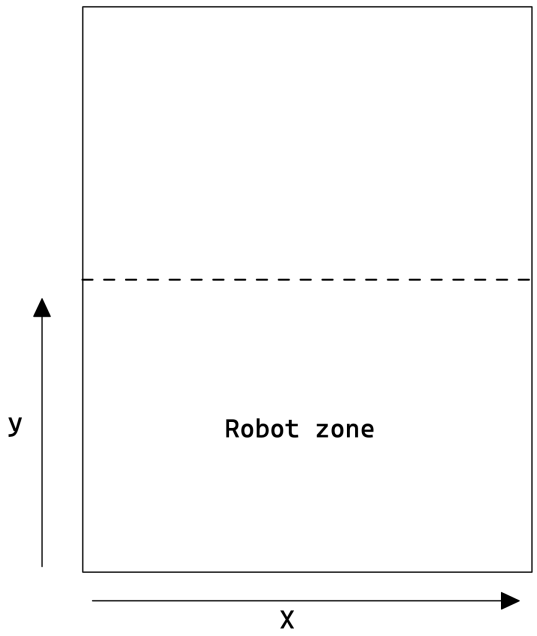


Figure 3.16: The table with robot zone

The camera has to recognize the object perfectly at high frame rate, for which extensive trial and error was done to find the perfect configuration that yields the best results. the configurations include lighting conditions, camera brightness, contrast object signature range and white correction.

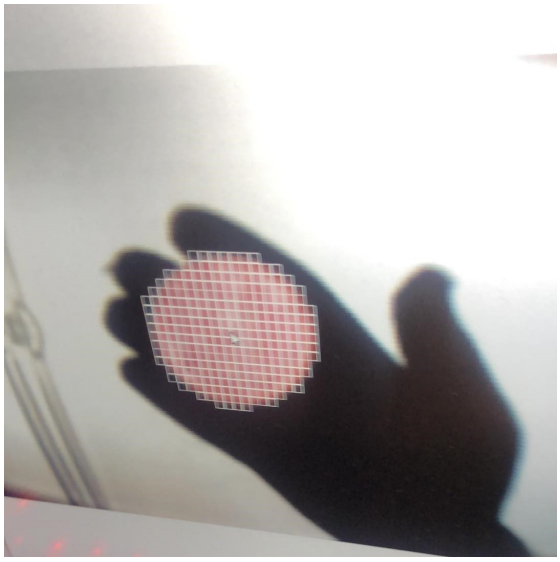


Figure 3.17: Object recognition using pixy2 cam

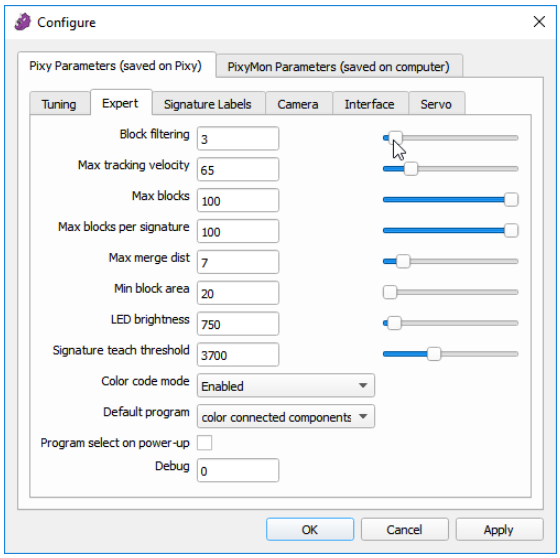


Figure 3.18: Pixy2 camera Configurations

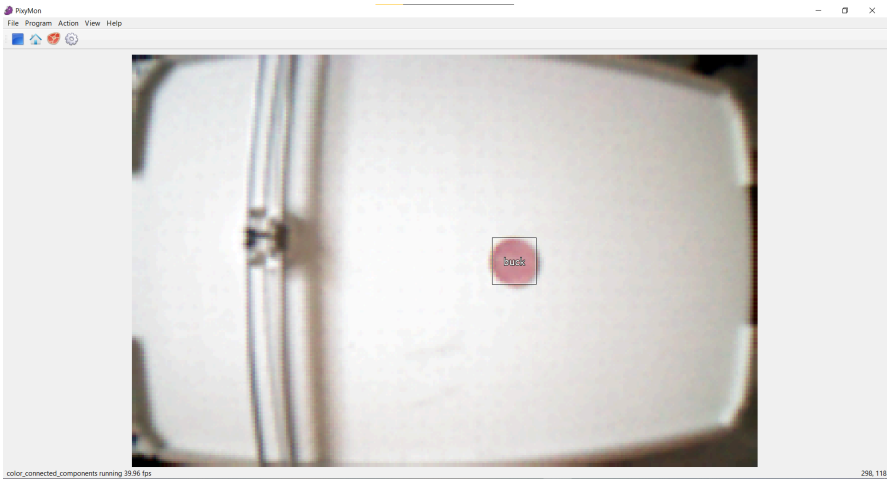


Figure 3.19: Camera view of the table with the puck detected

The camera has the following dimensions, 315 pixels for the x-axis and 207 pixels for the y-axis.

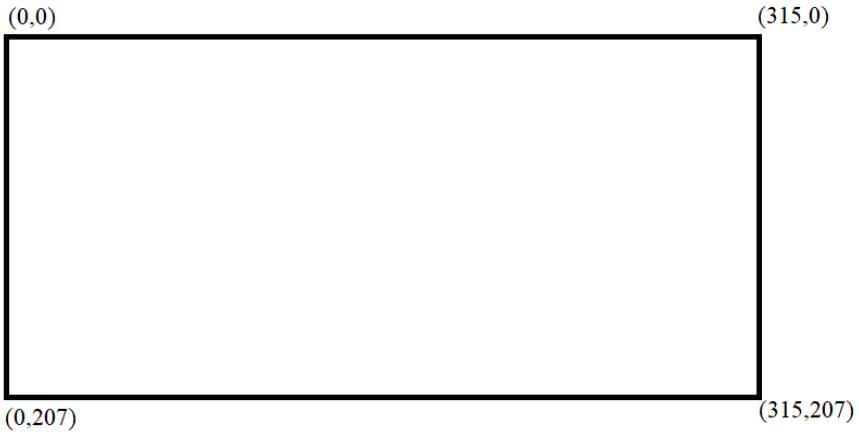


Figure 3.20: The table as seen by the camera

And the robot will be placed at a position in the x-axis, for example at x = 80.

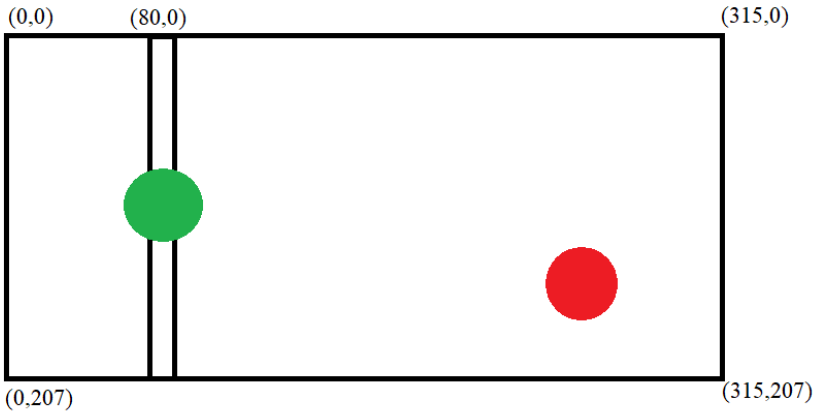


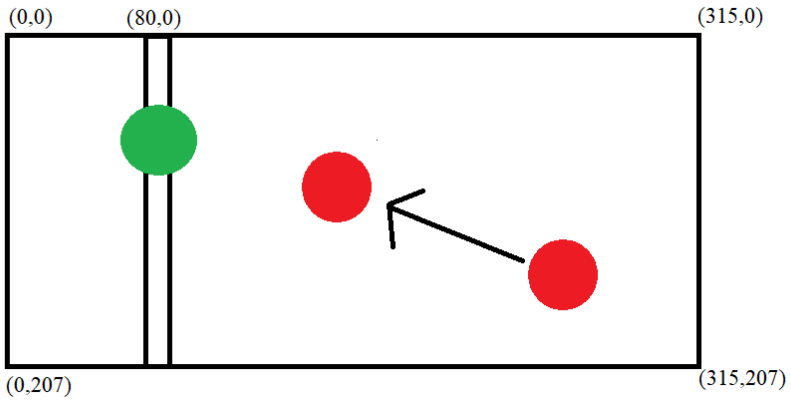
Figure 3.21: Robot initial position

We will be having the puck (in red color) moving freely in the gird and the camera gives us the coordinates (x,y) of the red puck, and we are calculating and trying to predict where the puck will cross the line that the machine is currently on (pixel 80 on the x-axis) and we will be moving the machine to that predicted point.

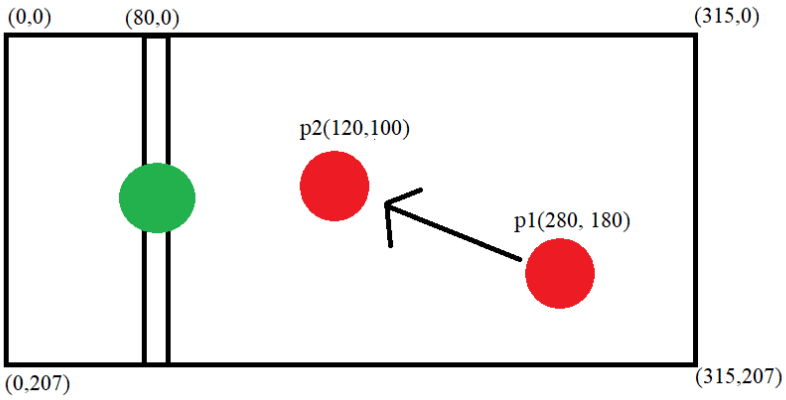
Our calculation is all about simple math calculations, we are using the linear equation to find the cross point of the puck with the current position for the machine. For example, if

the puck is moving toward the machine and that is represented in p1 and p2 then we could find where the machine (green circle) should be positioned by the following equation y – yo = m (x – xo) where m = (y2-y1) / (x2-x1).

and choosing xo,yo in the equation as any point from p1, and p2, and substituting the current position of the machine with x in the equation which in our example equals to 80 then we are finding the y coordinates that the machine should move to.



Calculations: M = (180 - 100) / (280 - 120) = 80 / 160 = 0.5 And choosing p2 as a point to use in the linear equation Y – 100 = 0.5 (x - 120), substituting x with the current position of the machine (80) Y – 100 = 0.5 (80 - 120) Y = -20 + 100 = 80 Then the green circle (machine) should be moving up to reach 80 pixels in the y-axis



The scoreboard

As the project is targeted to arcade halls mainly, a smart scoreboard system is needed, the researches developed a smart scoreboard that keeps track of both the player and the robot score.

The smart scoreboard consists of IR transceiver module which detects obstacles and send them to Arduino. The module is mounted on the goal area so when the buck passes this area, a signal is sent to the ESP module.

Figure 3.22: IR transceiver module

The ESP8266 is used to as a web server[5]. its configured in AP mode. when running, the ESP then can be connected to as a WiFi network, any device with WiFi connectivity can then connect and has access to scoreboard interface. the scoreboard is updated in real time with the sensors readings.

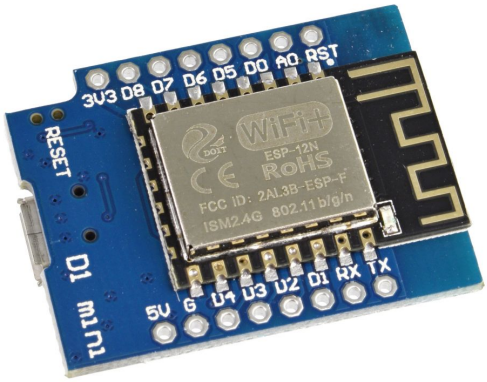


Figure 3.23: ESP 12-f with nodeMCU

4

RESULTS AND DISCUSSION

4.0.1 Testing

The testing process starts with finding the perfect parameters for the camera depending on the environment lighting conditions. after that the arduino code is loaded and the motors is tested to work independently of the camera. when the motors are assured to be working successfully the camera is connected and every this is connected with each other. this

modularity helps the debugging process.

4.0.2 Challenges Faced

Stepper Motors

Due to a known issue with stepper motor, named step loss[6], the current position become inaccurate over long periods of time. this can be avoided by monitoring the position of the robot by the camera and re position the robot periodically. for a normal match. this problem can be ignored

The Camera

The camera failed to detect the object in bad lighting conditions. this can cause the robot to act unexpectedly. the its recommended to provide a good lighting condition to obtain best results. the robots stops if that happen.

4.0.3 Summary

The robot was able to predict the direction of the ball in real time and with great precision. however, the attack is not strong enough to beat an adult person. But can beat a kid easily. the defense system is so strong that its pretty hard to score goals to it. the robot game play proofs the concept of the ability of robots to play hard to play games like air hockey.

5

Conclusion

The project uses a vision-system to create a robot that can play air hockey. The robot is moved using stepper motor. and its controlled by arduino.

The smart scoreboard uses ESP module along with NodeMCU to interface with the IR sensors. The ESP module acts as an access point that can be connected to from any device through WIFI.

5.0.1 Things We Learned

To begin the project. The we had to learn Arduino programming. we studied the UNO, MEGA and nano types to determine the best fit for the project. also we learnt alot about image processing and cameras, modules and sensors interfacing.

On the other hand. we learnt many things about mechanical parts. including CNC systems and timing. also we learnt about different types of motors.

5.0.2 Future Work

This project and its design principles can be used in a variety of situations. One of the project's limitations, which could be addressed in future work, is that the attacking algorithm is not "smart enough" to beat an adult person more advanced and attacking algorithms can be implemented in the future to farther enhance the playing skills of the robot.

The CNC system implemented is rather simple, but it has weight distribution issues that increases the load on the motors. also it needs three motors to operate. Implementing an H bot system can give better performance and requires only two motors.

A

Arduino Code

```
#include <Pixy2.h>
#include <FlexyStepper.h>

Pixy2 pixy;
```

```
FlexyStepper stepperX;
FlexyStepper stepperY;

const int MOTOR_X_STEP_PIN = 3;
const int MOTOR_X_DIRECTION_PIN = 4;

const int MOTOR_Y_STEP_PIN = 5;
const int MOTOR_Y_DIRECTION_PIN = 6;

const int yMinPixel = 0;
const int yMaxPixel = 207;
const int xMinPixel = 0;
const int xMaxPixel = 130;

const int PUCK_NOT_MOVING = 0;
const int PUCK_GOES_AWAY = 1;
const int PUCK_HEADING_TO_ROBOT = 2;
const int ATTACK_PUCK = 3;

const int MAX_Y = 570;
const double UNIT_Y = 2.75;    // (570 / 207)

const int MAX_X = 380;
```

```
const double UNIT_X = 3; // (380 / 125)

int posXDefender = 10;
int posXold = 0;
int posYold = 0;

int posX;
int posY;

int targetYInPixels = 0;
int targetYInMillimeters = 0;
float slope = 1.0;

int puckStatus = PUCK_NOT_MOVING;

unsigned long current_time = 0;
unsigned long old_time = 0;

const int DEFENSE_MODE = 0;
const int ATTACK_MODE = 1;
int mode;

void setpuckStatus(int x, int y, int xOld, int yOld){
```

```
  if(abs(x - xOld) < 5 && abs(y - yOld) < 5){
    puckStatus = PUCK_NOT_MOVING;
    if(x >= xMinPixel + 35 && x <= xMaxPixel){
      puckStatus = ATTACK_PUCK;
    }
  }else if(x > xOld){
    puckStatus = PUCK_GOES_AWAY;
  }else if(x < xOld){
    puckStatus = PUCK_HEADING_TO_ROBOT;
  }
}

void calculateYReflectionInPixels(int x, int y){
  int counter = 3;
  while(!(targetYInPixels >= yMinPixel && targetYInPixels <= yMaxPixel)
    int target_x = 0;
    int target_y = (targetYInPixels < yMinPixel) ?  yMinPixel : yMaxPix

    target_x = ((target_y - y) / (float)slope) + x;

    slope = slope * -1;
    targetYInPixels  = slope * (posXDefender - target_x) + target_y;
```

```
    counter--;
  }

}

void calculateTargetYInPixels(int x, int y, int xOld, int yOld){
    slope = (x - xOld == 0) ? 99999 : (y - yOld) / (float)(x - xOld);
    targetYInPixels  = slope * (posXDefender - x) + y;

    if((targetYInPixels < yMinPixel) || (targetYInPixels > yMaxPixel)){
      calculateYReflectionInPixels(x, y);
    }

}

void calculateTargetYInMillimeters(){
  targetYInMillimeters = targetYInPixels * UNIT_Y;
}

void setYPosition(int pos){
  stepperY.setTargetPositionInMillimeters(pos * -1);
}
```

```
int getYPosition(){
  return stepperY.getCurrentPositionInMillimeters() * -1;
}

void setXPosition(int pos){
  stepperX.setTargetPositionInMillimeters(pos);
}

int getXPosition(){
  return stepperX.getCurrentPositionInMillimeters();
}

void processDefender(){
  if(puckStatus == PUCK_NOT_MOVING || puckStatus == PUCK_GOES_AWAY){
    setYPosition(MAX_Y / 2);
    setXPosition(MAX_X);
  }else if(puckStatus == PUCK_HEADING_TO_ROBOT){
    if(targetYInPixels >= yMinPixel && targetYInPixels <= yMaxPixel){
      setYPosition(targetYInMillimeters);
      if(mode == ATTACK_MODE){
        setXPosition(MAX_X / 2);
      }
    }
```

```
  }else if(puckStatus == ATTACK_PUCK){
    setYPosition(posY * UNIT_Y);
    setXPosition(xMinPixel);
  }

  while ((!stepperY.motionComplete()) || (!stepperX.motionComplete())) {
    stepperY.processMovement();
    stepperX.processMovement();
  }

}

void setup() {
  pixy.init();
  pixy.setLamp(1, 1);

  Serial.begin(115200);
  Serial.print("Starting...\n");
  old_time = micros();

  mode = DEFENSE_MODE;

  stepperY.connectToPins(MOTOR_Y_STEP_PIN, MOTOR_Y_DIRECTION_PIN);
```

```
  stepperY.setStepsPerMillimeter(2);   // 1x microstepping
  stepperY.setSpeedInMillimetersPerSecond(30000.0);
  stepperY.setAccelerationInMillimetersPerSecondPerSecond(8000.0);
  stepperY.setCurrentPositionInMillimeters(0);

  stepperX.connectToPins(MOTOR_X_STEP_PIN, MOTOR_X_DIRECTION_PIN);
  stepperX.setStepsPerMillimeter(2);   // 1x microstepping
  stepperX.setSpeedInMillimetersPerSecond(10000.0);
  stepperX.setAccelerationInMillimetersPerSecondPerSecond(3000.0);
  stepperX.setCurrentPositionInMillimeters(0);
  stepperX.setTargetPositionInMillimeters(0);

}

void loop() {
  pixy.ccc.getBlocks();
  posX = pixy.ccc.blocks[0].m_x;
  posY = pixy.ccc.blocks[0].m_y;

  current_time = micros();

  if(current_time - old_time >= 1000){
    old_time = current_time;
```

```
setpuckStatus(posX, posY, posXold, posYold);

if(puckStatus == PUCK_NOT_MOVING){
  Serial.println("Not Moving");
}else if(puckStatus == PUCK_GOES_AWAY){
  Serial.println("Buck heading away");
}else if(puckStatus == PUCK_HEADING_TO_ROBOT){
  calculateTargetYInPixels(posX, posY, posXold, posYold);
  calculateTargetYInMillimeters();
  Serial.println("Buck comming to us");
}else if(puckStatus == ATTACK_PUCK){
  Serial.println("ROBOT Attacking Puck");
}

if(mode == ATTACK_MODE && puckStatus == PUCK_HEADING_TO_ROBOT){
  posXDefender = 125 / 2;
  stepperX.setSpeedInMillimetersPerSecond(20000.0);
  stepperX.setAccelerationInMillimetersPerSecondPerSecond(5000.0);
}else{
  posXDefender = 10;
  stepperX.setSpeedInMillimetersPerSecond(10000.0);
  stepperX.setAccelerationInMillimetersPerSecondPerSecond(3000.0);
```

```
    }

    processDefender();

//    Serial.print(" X: ");
//    Serial.print(posX);
//    Serial.print(" Y: ");
//    Serial.print(posY);
//    Serial.print(" targetYInPixels: ");
//    Serial.print(targetYInPixels);
//    Serial.print(" targetYInMillis: ");
//    Serial.println(targetYInMillimeters);

    posXold = posX;
    posYold = posY;
  }
}
```

References

[1] Al-Sammarraie, M. A. J. & Özbek, O. (2021). Comparison of the effect using color sensor and pixy2 camera on the classification of pepper crop. *Journal of Mechanical Engineering Research and Developments*, 44(1), 396–403.

[2] Hackel, M. (2007). *Humanoid robots: Human-like machines*. BoD–Books on Demand.

[3] Hsu, F.-h. (1999). Ibm's deep blue chess grandmaster chips. *IEEE micro*, 19(2), 70–81.

[4] Lund, H. H. (2003). Adaptive robotics in the entertainment industry. In *Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation. Computational Intelligence in Robotics and Automation for the New Millennium (Cat. No. 03EX694)*, volume 2 (pp. 595–602).: IEEE.

[5] Mesquita, J., Guimarães, D., Pereira, C., Santos, F., & Almeida, L. (2018). Assessing the esp8266 wifi module for the internet of things. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1 (pp. 784–791).: IEEE.

[6] Schweid, S. A., McInroy, J. E., & Lofthus, R. M. (1995). Closed loop low-velocity regulation of hybrid stepping motors amidst torque disturbances. *IEEE Transactions on Industrial Electronics*, 42(3), 316–324.

[7] Upadhyaya, S. V., Israni, D., Jasani, K., & Shah, A. (2016). A novel approach to precisely control linear movement of sensor by motor using microstepping. In *2016 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)* (pp. 242–246).: IEEE.

[8] Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2), 4–10.