2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023
The Harbor of Engineering Education for 130 Years
Paper ID #37399

Ping Pong Robot with Dynamic Tracking

Asad Yousuf, Savannah State University

Asad Yousuf is the Chair of the Department of Engineering Technology at Savannah State University

Md Rakibul Karim Akanda, Savannah State University

Md Rakibul Karim Akanda is the Coordinator and Assistant Professor of Electronics Engineering Technology at Savannah State University. He received his PhD in Electrical Engineering from University of California, Riverside.

Ping Pong Bot: Player Tracking Training

Abstract

With the growing interest in Modeling and Simulation and lack of institutions with a degree program in the discipline in our region, designing, promoting and implementing a degree program in an institution that traditionally attracts students from the minority population will represent a viable pathway to increasing the participation of underrepresented minorities in this emerging field. After an extensive search we could not identify any Minority Serving Institution (MSI) that offers a degree program in Modeling and Simulation in our region. This paper will present the Modelling Simulation project which will help students learn the concepts of Modeling and Simulation. The modeling and simulation project work is supported by the grant from the Department of Education. To successfully train in table tennis, players must practice returning precise shots and developing a consistent play style to deal with the game's varied paces. Table tennis robots are used by players who want a guaranteed series of shots or a pattern of shots that do not change in angle or speed. These robots fire ping pong balls at the player in a controlled manner, typically by using a motor to fire the ping pong balls at certain angles in sequence. There are also robots that can make fixed and random serves, providing a comparable sensation to competing against a real player. The Ping Pong Bot* project tries to make this procedure more dynamic by using a model trained to recognize human torsos and hand signals to track the player. The goal is for the robot to engage the player in three different modes. The first mode involves the robot tracking the player and serving ping pong balls to them at their location. The second phase involves the robot purposefully shooting balls at areas where the player is not present. The third mode alternates between serving ping pong balls at the player's position and serving ping pong balls at areas where the player is not present. For this reason, a convoluted neural network is utilized, and the model created was simulated with matplotlib, where the model's accuracy was tested. The raspberry pi 4 b+ serves as the robot's brain, on which a generalized neural network was trained to detect the shape of a human player as well as hand signals for numbers one, two, and three. Servos are used to feed the firing chamber, agitate, and position the ping pong balls, and adjust the angle at which the firing mechanism is faced. The ping pong balls are fired by a brushless outrunner motor.

* Supplies for this project could not be delivered on time. Therefore, this paper is written as astudy as to how the Ping Pong Bot would be developed.

Tasks to Be Accomplished

The project's first goal is to successfully train a neural network to detect the torsos of ping pong players. Hundreds to thousands of photos of various ping pong players' torsos would be utilized to feed the machine learning process. This would be performed using tensor flow libraries, which would deconstruct the photos and store the data in arrays. A convolutional neural network would be utilized to achieve generalized learning, allowing it to deal with novel input.

To implement the three modes mentioned in the abstract, in which the robot shoots ping pong balls at the player, then away from the player, and finally shoots randomly, the neural network must be trained to recognize three distinct hand signals: (i) the standard pointer finger will be used to represent the number one; (ii) the standard pointer and middle finger will be used to represent the number two; and (iii) the thumb, standard pointer, and middle finger will be used to represent the number three (Figure 1.0).

Once the neural network can detect torsos and hand signals accurately, it will be necessary to specify that the torso must be in the center of the camera during the mode designed for shooting the

ping pong balls at the player, and at the edges of the camera during the mode designed for shooting balls away from the player. Furthermore, some method of sensing the angle at which the firing chamber is directed must be applied, because if the firing chamber is pointed at an extreme angle, the fired ping pong ball will completely miss the table. By rotating the firing chamber, the robot will be able to alter the angle at which the ping pong balls are fired. The firing chamber will be connected to a flat gear-based frame. Because of the impact of another servo-attached gear, the gear will rotate.

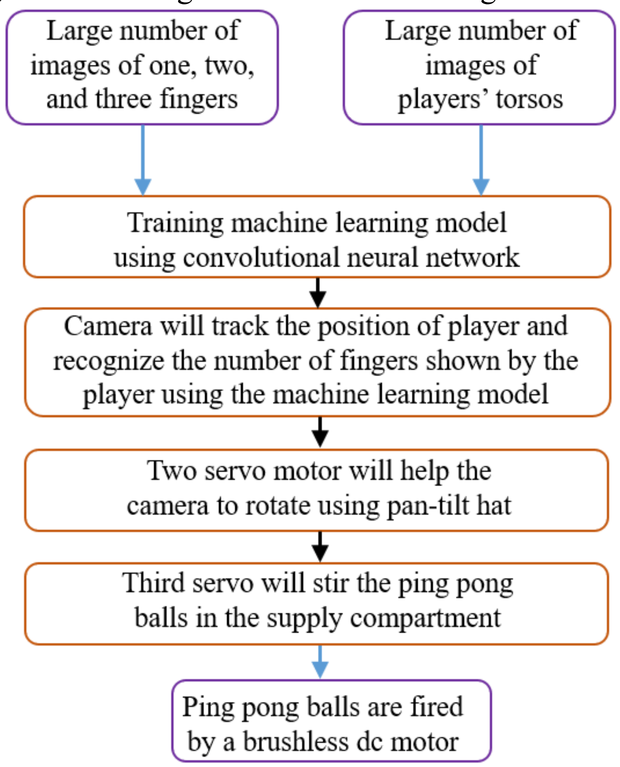

Figure 1.0 (Flow Chart of Designing the Ping Pong Robot)

A cache of ping pong balls will need to be attached to the robot, and the firing chamber will need to be constantly supplied with ping pong balls while it is operating. The ping pong balls must be delivered one at a time, and the technique must not allow the delivery system to become clogged. Furthermore, once in the firing chamber, the ping pong balls must be fired quickly enough to clear the net and fall on the opposing side of the table. The delivery method will be a pipe with a slightly larger diameter than a ping pong ball. The ping pong ball supply compartment will use a servo to stir the ping pong balls until they make their way to a hole just large enough for them to slide into and down the pipe. The ping pong balls will be launched with a brushless dc motor once they have passed through the pipe.

Possible Designs and Reference Projects

Initial Design

Prior to further investigation, the project's original design called for the ping pong robot to have a supply cache from which ping pong balls would fall into a distribution pipe that went into the firing chamber (Figure 2.0). The robot's base was a gear that rotated owing to a secondary gear moved by a servo. The pipe, power source, and chamber were all connected to a chassis that held the raspberry pi and served as the hub for all electronic connections. The camera was mounted atop the firing chamber, which was made up of a rounded piece of piping and two dc motors. Please see the following figure for a graphical representation of the original design:

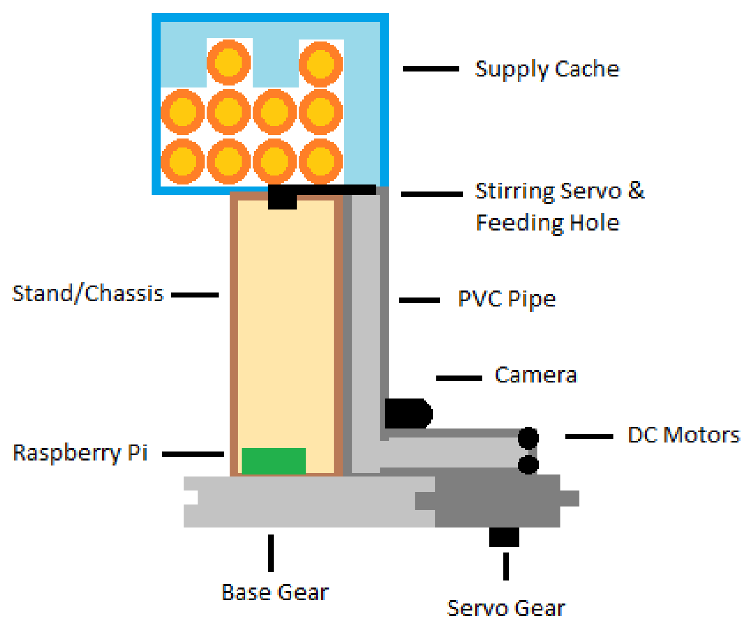

Figure 2.0 (Initial Design of the Ping Pong Robot)

Mechanical Design For Effective Ping Pong Ball Delivery

There was one mechanism and machinery that was mentioned repeatedly when researching the mechanisms and machinery for making an automated project. "Design of a Smart Ping Pong Robot" was the title of the model project [1]. There are numerous methods to improve the design of a ping pong robot. One of the best methods to build a ping pong robot is to use readily available materials that can be quickly replicated and scaled up.

A PVC pipe is the most essential mechanism for keeping a steady stream of ping pong balls. The PVC piping can handle a variety of sizes. A ping pong ball is typically 40 millimeters (mm) in size

[2]. The PVC that would suit the average ping pong ball size is 1.5" PVC piping with an inside diameter of 41.27 mm [3]. This permits any mistakes that occur during the manufacturing process. To contain the ping pong balls, something durable and functional in most normal settings would be required. A hard plastic basket could be used to hold all the ping pong balls because it is malleable but rigid enough to maintain its shape.

Because the ping pong balls must fall effortlessly, a funnel that can continuously drop the ping pong balls towards the PVC opening is required. With that requirement, the simplest choice is to shape the ball container into a cone or pyramid (Figure 3.0). To feed the ping pong balls through the closed system, the ping pong ball is dropped to the crankshaft, which pushes the ping pong ball

forward with enough force to move it but not shatter it. When it reaches the end of the shaft, force must be applied to ensure that the ping pong ball has the necessary velocity for the player to volley. The ping pong ball's velocity will be generated by the wheel at the shaft's extremity. The wheel gives frictional force and a constant that can be kept constant but also varied if necessary. A voltage supply, such as a 12-volt DC motor, is required to power the system. The engine is responsible for spinning the wheel and moving the actuator.

Nothing can move forward unless the engine is started, and it will remain at the PVC piping intersection. The rotation of the tire allows it to be measured for each serve of the ping pong ball. One example would be completing 60 rotations per minute, which equates to 1 second per ping pong ball launch. This can be adjusted to be slower to accommodate beginners or faster to fit more advanced players. Figure 3.0 depicts this full design as another example of a potential robot design.

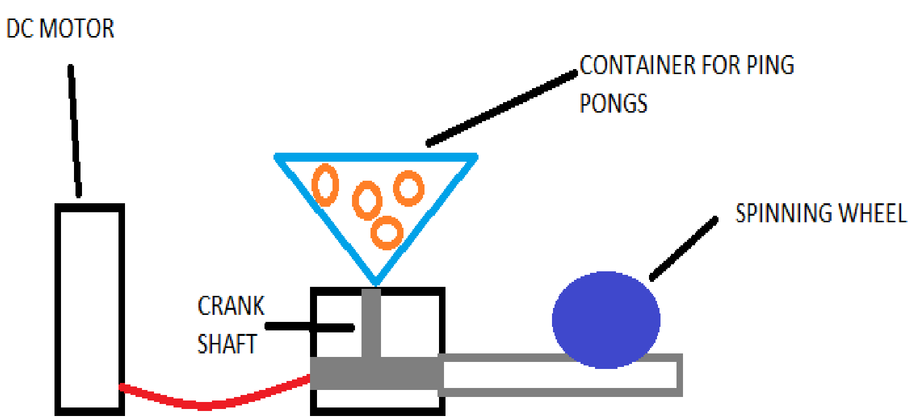

Figure 3.0 (The mechanism for ping pong ball delivery)

Object Tracking Reference Project

A project discovered during the investigation of implementing the targeting of player torsos by the raspberry pi program proved to be a useful reference for future development, and it was a project by Macharla [4]. Image processing was used in the Macharla project to track a face and place it in the center of the camera stream. This was an excellent reference project because the tracked face could be replaced by any object/image that the neural network had been trained to recognize. In our situation, we would be tracking player torsos rather than faces.

A camera was mounted on a fixture made of a horizontal and vertical servo to control its lateral and longitudinal axis (Figure 4.0). The camera's video stream was fed into the raspberry pi where each frame was read, and the servo positions were adjusted so that the face was in the middle [5].

Using this project as a model, the end project will use the same libraries and coding format for targeting. For example, programming for servo positions must be applied, where x and y positions are determined by classes and definitions. The sample of such code is shown below:

Code 1.0 (Controlling x and y servo positions)

```
class servopos():
    def__init__(self):
        self.currentx,self.currenty=7,4
        x.start(self.currentx)
        y.start(self.currenty)
        sleep(1)
        x.ChangeDutyCycle(0)
        y.ChangeDutyCycle(0)

    def setposx(self,diffx):
        self.currentx+=diffx
        self.currentx=round(self.currentx,2)
        if(self.currentx<15 and self.currentx>0):
            x.ChangeDutyCycle(self.currentx)
    def setposy(self,diffy):
        self.currenty+=diffy
        self.currenty=round(self.currenty,2)
        if(self.currenty<15 and self.currenty>0):
            y.ChangeDutyCycle(self.currenty)
    def setdcx(self,dcx):
        x.ChangeDutyCycle(dcx)
    def setdcy(self,dcy):
        y.ChangeDutyCycle(dcy)
```

Another function of the program is to compute the deviation of the targeted object from the camera's center and set the new coordinates to which the servos will point. The error number will be saved and used to reorient the servos in the following section of code:

Code 2.0 (Reorienting Servos)

```
if abs(error_x)<20:
    ser.setdcx(0)
else:
    if abs(valx)>0.5:
        sign=valx/abs(valx)
```

```
            valx=0.5*sign
        ser.setposx(valx)

if abs(error_y)<20:
        ser.setdcy(0)
else:
        if abs(valy)>0.5:
            sign=valy/abs(valy)
            valy=0.5*sign
        ser.setposy(valy)
```

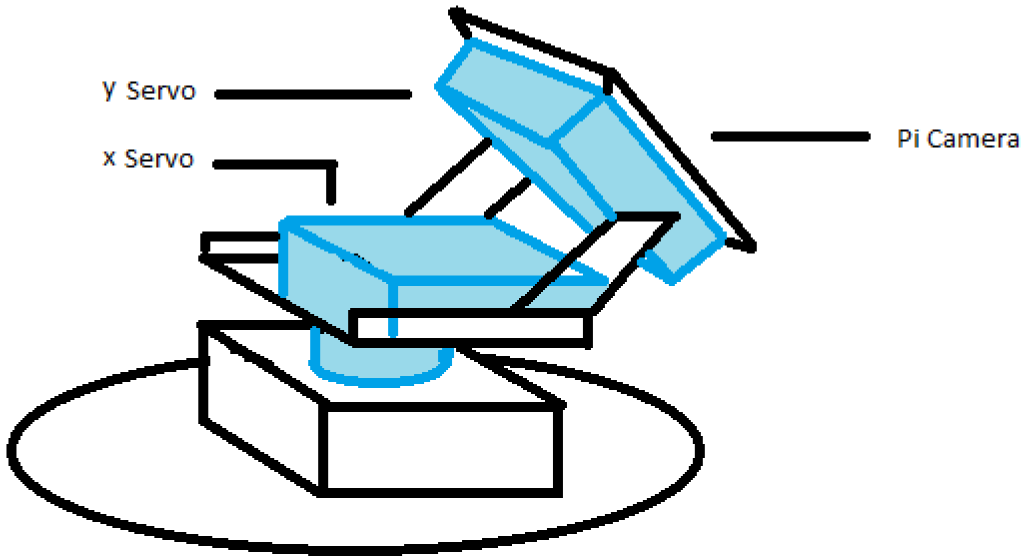

Figure 4.0 (Controlling x and y servo positions to rotate camera. One servo (x Servo) will rotate the camera in the horizontal direction, and another servo (y Servo) will rotate the camera in the vertical direction to track the player.)

Design Conclusion

Following consideration of different design paths, the final design that will be applied in the future is a ping pong robot with a variable longitudinal and lateral positioning system for the firing chamber. This will necessitate the addition of one more servo capable of elevating or lowering the shooting chamber in relation to the table. This is done to get a more precise shot at the player by considering how much of the frame they take up [6]. If the player is close to the table, a lower angle could be used to make it more difficult for the player to respond. In addition, a more aggressive firing mechanism will be used, with an actuator, a relay, and a rpm control mechanism integrated into the system. This changes the pace at which the ping pong balls are fired at the player. Instead of a tub, the supply cache will be in the form of a pyramid to encourage the flow of the ping pong balls in a smoother manner.

References

1. IJIRST - International Journal for Innovative Research in Science and Technology. “Design

of a Smart Ping Pong Robot.” *Academia.edu*, 19 Sept. 2015, https://www.academia.edu/15892332/Design_of_a_Smart_Ping_Pong_Robot.

2. “Ping-Pong Ball: Table Tennis Ball Dimensions & Drawings.” *RSS*, https://www.dimensions.com/element/ping-pong-ball-table-tennis-ball.

3. Author Homestratosphere's Editorial Staff & Writers. “PVC Pipe & Fittings Sizes and Dimensions Guide (Diagrams and Charts).” *Home Stratosphere*, 15 Sept. 2021, https://www.homestratosphere.com/pvc-pipe-and-fittings-sizes/.

4. Macharla, Monisha. “Object Tracking Camera Using Raspberry Pi and Opencv.” IoTEDU, 15 Sept. 2020, https://iot4beginners.com/object-tracking-camera-using-raspberry-pi-and-opencv/.

5. “The Complete Guide to Program Pan-Tilt Servos” Miller, https://www.learnrobotics.org/blog/the-complete-guide-to-program-pan-tilt-servos

6. “How to Perform Object Detection with Tensorflow Lite on Raspberry Pi.” *Digi*, https://www.digikey.com/en/maker/projects/how-to-perform-object-detection-with-tensorflow-lit e-on-raspberry-pi/b929e1519c7c43d5b2c6f89984883588.