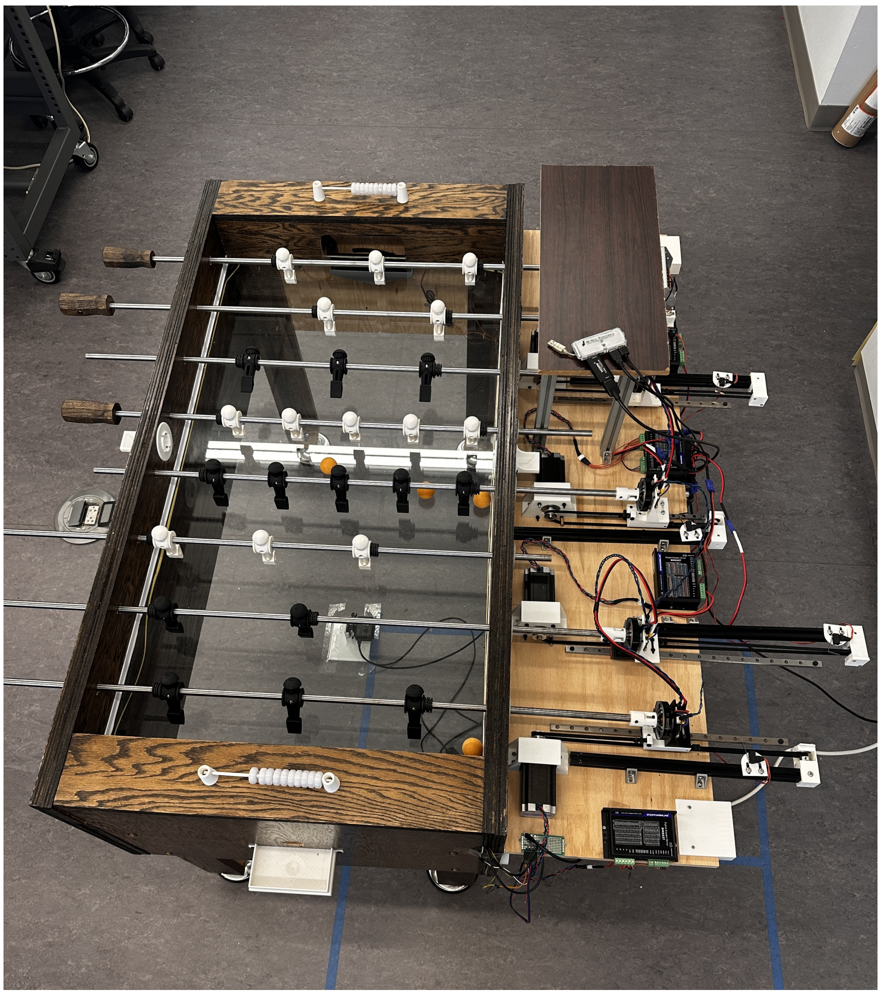
MAE 6900 Independent Design Project Report

Design of a Semi-Autonomous Foosball Table



By:

Julian Prieto (jap465)

Report Submitted On:

February 10, 2025

Project Advisor:

Matthew Ulinski

1 Letter of Transmittal

This document details the construction of the Autonomous Foosball Table project, developed as part of my Master of Engineering program at Cornell University. This project is a solo endeavor that represents an integration of mechanical, electrical, and software systems, designed to transform a traditional foosball table into an autonomous and interactive machine capable of playing against human opponents.

The objective was to automate a task. As a result, my robotics skills were thoroughly tested and I was able to learn new fields through self-study. By automating gameplay, the project explored the challenges of hardware decision-making, computer vision, electrical wiring, and plenty of debugging—all critical aspects of modern robotics and automation. It also served as an opportunity to combine theoretical knowledge with practical application.

Beyond its technical merits, the project aims to demonstrate robotics' potential to enhance recreational activities. Automation does not necessarily need to be isolated from human interaction, and this platform tests how engineering concepts can be integrated in entertainment that is immersive because it permeates the real world. With a budget of around a thousand and five hundred dollars (about five hundred dollars less than a high-end foosball table), I was able to produce a product that exceeds commercial alternatives.

I hope you find the insights and conclusions detailed in the accompanying report valuable. Please do not hesitate to reach out with any questions or to discuss further applications of the work presented.

Thank you for your time and interest in this project.

Sincerely,

Julian Prieto

Table of Contents

Contents

Acknowledgments

This project has come together over three semesters with the help of the people I'd like to take a moment to thank here:

- Matt Ulinski for his generous support and advising role.
- Professor Jinjie Yeo for his contributions as my first advisor.
- Katherine and Sam Knecht for their contribution and consultation on the physical construction of the table.
- Alex Eagan and Mateo Guynn for donating the temporary space needed to construct and facilitate the table.
- Stephan Wagner for his invaluable coding guidance
- Shae Marks for her coding advice and encouragement
- Those who watched this project come together from nothing with full confidence that the end product would come together. You all gave me the confidence I needed and I thank you for that.

2 Abstract

This project has resulted in a functional foosball table with a glass playing field that is able to successfully identify and track a foosball with robustness to human interaction. The table can predict ball motion and manipulate the rods in a lateral and rotational capacity to successfully play against human opponents.

Figure 1. The current autonomous table prototype.

3 Introduction

Foosball, commonly known as table soccer, is played on a table where two players each manipulate four rods that rotate and move plastic players laterally to score a ball into a goal. The automation of this project involves the dissection of the game into four key components:

- Hardware: Facilitating movement of each rod's two degrees of freedom.
- Electrical Design: Powering and communicating with the motors and sensors.
- Software Design: Translating ball position and player positions into a coordinated strategy.
- Computer Vision: Tracking and predicting the path of the ball.

4 Literary Review

I'm not the first to attempt this project, so I began by reviewing and researching other projects. The primary paper I referenced throughout my project was "The Study of Semi-Automated Football Table" by Mohebi Dani.

This paper gave me invaluable information, such as Figure 2, which detailed the varying sizes of popular foosball table brands. The lack of standardization was not surprising, but made it challenging to determine finalized dimensions. This paper was also the source for key assumptions I use such as the top speed of the ball (10m/s). Additionally, this paper guided me in my selection of actuators. Initially, I considered off-the-shelf DC linear actuators as mentioned in this paper. The low speed and high cost ultimately made me decide that a belt driven by stepper motors was a better choice.

STYLE	Make	Playing area(cm)	3-bar spacing(cm)	Goal width (cm)	Ball wgt(g)
Italian	Garlando	114-120x70.5	18,4	19	17,5-18,5
	Roberto Sport(UCD)	111-118x70.5	16,0	19	17,5
	FABI	120x70	16,8	19	17,5
	FAS	117x70	16,4	19,5	17,5
	Longoni	113x70	16,5	21,5	17,5
	Brighouse/FASNA	108x70	16,2	19,8	17,5
	Sardi	120x69	16,5	19,5	17,5
German	Loewen	119x68.5	16,4	17	20,5
	KCE/TS	118.5x68.5	15,9	18	20,5
French	Bonzini	119x69	17,0	19	14
	Rene Pierre	119.5x69	17,8	20	11
Benlux/Gr	Soccer 2000	116.x69	16,0	19	14,5
	Jupiter	117x68	16,0	19	11,2
	Topper 2000	111x65	16,4	18	15,4
Spanish	Biufca	134x75	n/a	25	28
American	Tornado	120x68.5	16,2	20,5	27
	Sivissidis	119x68.5	18,5	20,5	27

Figure 2. Other Foosball table brands and their varying dimensions. Dani 2022

The paper ”Low latency vision-based control for robotics” by Joshua Lues was also an abundance of information I would view to inspire my implementation. This paper discusses the details of their own physical implementation stressing the importance of low-latency vision systems when designing and testing a foosball table. Specifically, this paper analyzed how the reliability of intercepting the ball is directly dependent on the latency of its vision system. To test this, Joshua Lues experiments across different CPU's and camera's to find that an FPGA-HPS and Terasic D5M camera module performed best with around 26.4ms of latency.

5 Methodology

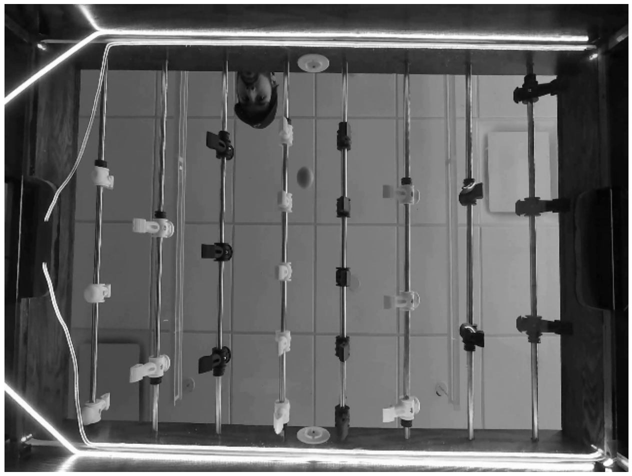


Figure 3. A camera frame from under the table used to detect the ball.

The design of my table holds up to the widely considered tournament standard brand Tornado, which means it is approximately 120 x 68.5 cm or approximately 47 x 26 inches with the key feature being its tempered glass bottom. This clear field allows a GoPro camera mounted below to detect and predict the ball location in real life.

Using the position stored over time in a circular buffer, a python script sends commands through Serial communication to an Arduino Mega, which moves stepper motors to create lateral motion. Brushless DC motors are commanded directly through python for rotational motion. The main script is written to carry out the basic strategy outlined in Figure 4. This strategy, though simplistic, can be effective with fast enough actuation and accurate ball tracking.

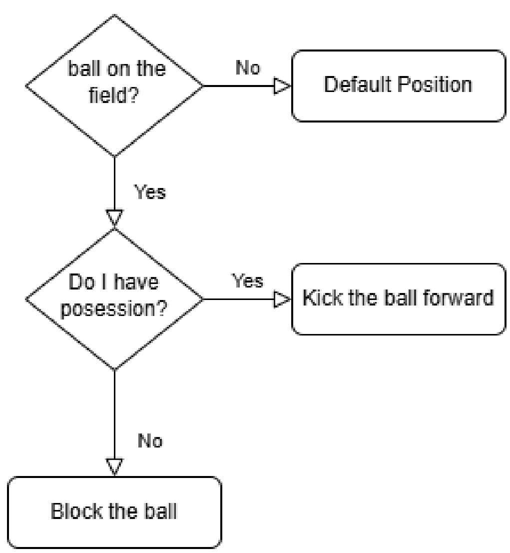


Figure 4. A simple strategy to be used in the most basic version of the game.

6 Design

6.1 Overview

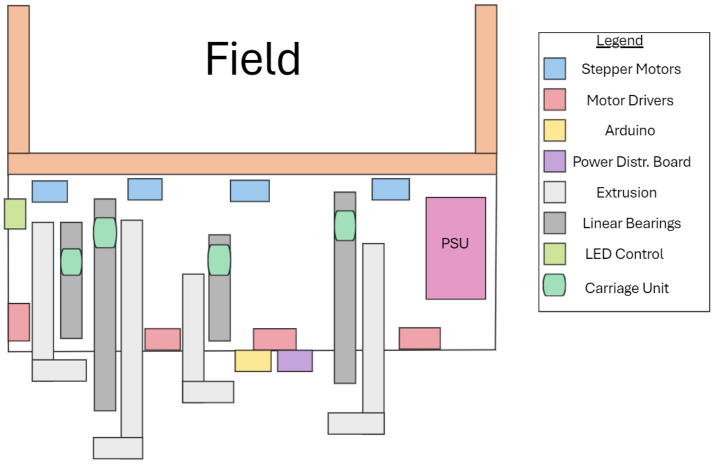
As a brief overview and emphasis of the unique choices in my design, I present the following as key elements of this project:

Subsystem	Feature	Explanation
Mechanical		
	Glass Field	Allows for unobstructed view of the ball at all times
	Hinged Platform	Allows for easy transportation and reversion to a normal table
	Linear Belt Drive	Custom and provides lateral movement
	Belt Tensioner	Keeps the belt from skipping allowing accurate control
Electrical		
	Headers	Every cable has a polarity enforced header
	Arduino Shield	Allows easy connection to motor drivers and limit switches
	LEDs and Control Board	Homogenizes the ball image and is isolated from main power
Software		
	Dual-process System	Allows control to operate quicker and predict ball position
	Ball Prediction	Provides estimates of ball location in between sensor inputs
Computer Vision		
	Thresholding	Finds LEDs accurately and limits ball search space
	Template Matching	Finds the position of the ball quickly without interference

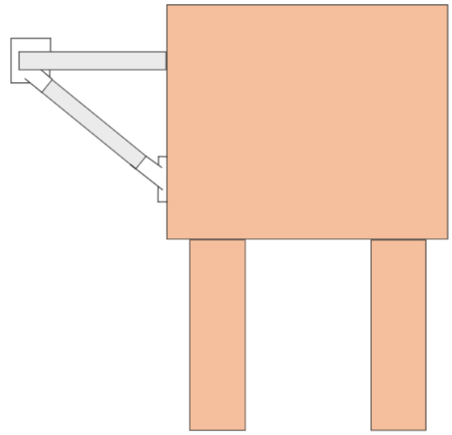
Table 1

6.2 Mechanical Subsystem

The mechanical subsystem is designed to integrate seamlessly with the table itself through a hinged wooden platform. This supplies a sturdy surface that allows for easy attachment of the components necessary for automation. When not locked in place by the extrusion braces, the mechanical subsystem can swing down for reversion to a standard table. The placement of each component follows the design in Figure 5.



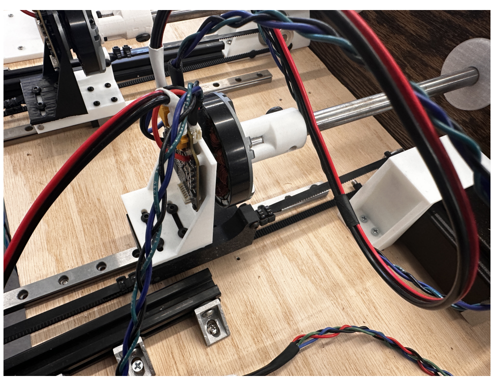
(a) The layout of the Mechanical Subsystem.



(b) A diagram displaying the side of the table and the platform support.

Figure 5

The carriage unit is a 3D printed block that holds an mj5208 brushless motor which is controlled via an MJBots Moteus r4.11 module.



(a) The carriage unit that moves on a Linear rail bearing.

(b) The belt tensioner seen above is 3D printed from three pieces and utilizes a standard skate bearing to tension the belt. This prevents the belt from skipping under the large torque from the motors.

Figure 6

Each stepper motor is fitted with a 22mm GT2 pulley that drives a belt, moving the carriage unit along a linear rail. The belt tensioner, seen in Figure 6b, ensures that the prototype linear actuator maintains accurate open loop control of lateral positioning. This way the foosball players can move at speeds up to 8.33 m/s without skipping steps. Limit switches are fitted to one end for calibration on startup.

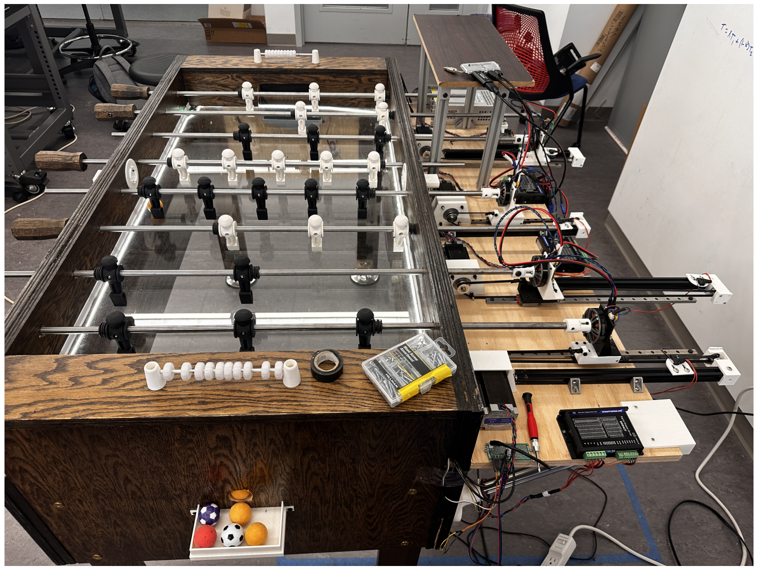
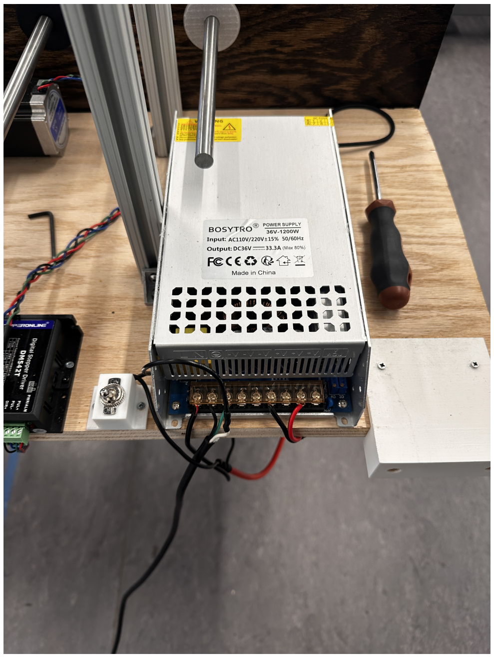


Figure 7. A side view of the current prototype.

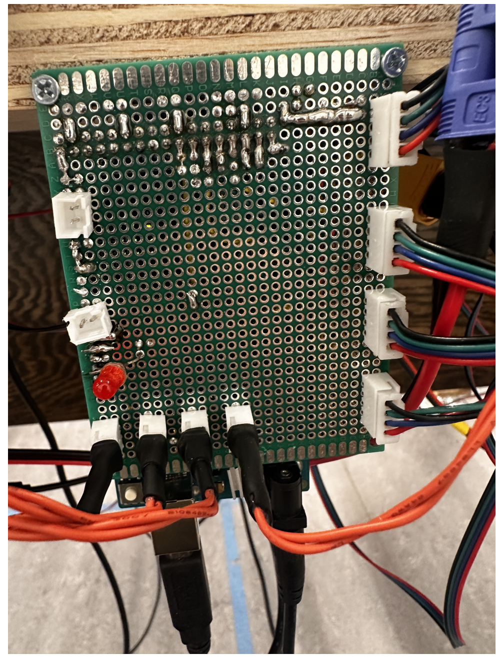
A small auxiliary table extends above the platform. This smaller table facilitates the placement of a laptop running the control script for the foosball table and requires only a single USB-port connection.

6.3 Electrical Subsystem

The electrical subsystem serves as the backbone of its operation. There are a total of eight motor drivers, four limit switches, and a GoPro that all need power and communication. Power distribution is handled through parallel connections to a 1200-Watt, 36-Volt power supply shown in Figure8 utilizing EC3 headers. This ensures error-free connections and simplifies component replacement. Communication wires utlize JST headers of various sizes which also enforce polarity. An emergency stop button has been mounted near the opponent which is connected in series to the main power supply.



(a) The main power supply unit of the table that can provide up to 1200W at 36V. This gives plenty of power to the system.



(b) The shield for the Arduino Mega that receives four pin headers for communication with the stepper motors and two pin headers for communication with the limit switches.

Figure 8

The Arduino Mega in Figure 8b facilitates communication with the stepper motors and uses the limit switches to calibrate positional information. A shield for the Arduino was created from male headers and a proto-board in order to facilitate connections to the stepper motors and limit switches. Messages are received via Serial to the Arduino at a BAUD rate of 115200 bits per second, supporting a stepper motor control frequency of about 200 Hz.

The mj5208 brushless motors are driven and controlled in a closed-loop using the r4.11 Moteus module. This module uses a magnetic encoder and CAN-FD communication to operate brushless DC motors. These motors are capable delivering up to 1.7 Nm of instantaneous Torque and speeds up to 7500 RPM. For video capture, a GoPro Hero 5 camera streams data live via a micro-HDMI cable and a capture card to USB. This gives the vision system 60 frames per second at a resolution of 480p x 640p.

6.4 Software Design

To maximize control frequency, I split my code between two separate python processes that run in a loop at vastly different speeds.

- Controls Process: Handles the manipulation of the rods and the communication with the Arduino. Runs about every 5 ms or 200Hz

- Vision Process: Receives data from the GoPro and processes the frame using computer vision to determine the ball position. Runs about every 30 ms or 33 Hz

In addition to these python processes, I implement a script to control the stepper motors on the Arduino which can receive messages at a baud rate of 115200 bits per second.

By separating these scripts, I can allow the ball position to be predicted through the controls routine. By saving the last locations of the ball into a buffer, I can estimate the speed and trajectory of the ball in each direction.

$$v_x = \frac{x_2 - x_1}{t_2 - t_1} v_y = \frac{y_2 - y_1}{t_2 - t_1}$$

Figure 9. A screenshot predicting the ball trajectory generated from the past data.

For any two given locations of the ball, I can stretch a trajectory path in intervals of the time it takes for a single control loop and predict when and where that trajectory intercepts with a rod. This allows the table to move much faster than it can sense. Once the location the rods should go is determined, a string is sent to the arduino that holds their coordinates. The arduino processes the messsage and steps each motor at a speed that proportionally scales according to the current distance to the destination. The greater the distance the greater the speed.

6.5 Computer Vision

Arguably the most difficult and involved task I had to accomplish was finding the ball's location. Given pixel locations for four points that have corresponding coordinates in the real world, one can find a homography matrix H that transforms image coordinates to real coordinates according to Equation 1.

$$\begin{bmatrix} x_{real} \\ y_{real} \\ 1 \end{bmatrix} = H \begin{bmatrix} x_{image} \\ y_{image} \\ 1 \end{bmatrix} \tag{1}$$

To establish four points in real life that I know the location of, I place four green LEDs on each corner of the table. Getting the pixel locations involves implementing either of two methods:

- Thresholding: Using a specific range of values across the HSV color spectrum to isolate an object and calculate its center.
- Template Matching: Calculate the correlation of each pixel and its neighbors to a template of the ball. The pixel that matches the most should be the location of the ball offset by half the height and width of the template.

The ”thresholding” technique in computer vision is well-suited for applications where only a single prominent color is present. This could effectively identify the green LEDs in each corner very quickly and accurately. The limited region of interest where each LED is located also greatly speeds up the process.

The template matching operation was applied to find the ball. By isolating the saturation channel, I get an image like Figure 10. With a 25 x 25 pixel image of the ball, I could very accurately and quickly determine the pixel position. By combining this with a more broad thresholding range, I can be even more confident of the ball's position.

With the current GoPro and capture card setup, the streamed framerate was capped by the capture card at 60 fps and the resolution streamed at 640 x 480 pixels. This meant that I could find the position of the ball within a 2mm radius of where it actually was at a maximum of 60 Hz (once every 17 ms). These frames are stored into a buffer and only the most recent frame is used when detecting the ball so as long as the camera detection routine takes longer than 17ms, it will always have the most recent frame. The complexity of the template matching routine is the bottleneck of the whole process due to large matrix multiplication across the whole image, and it brings the overall vision routine speed to 33 Hz (30ms).

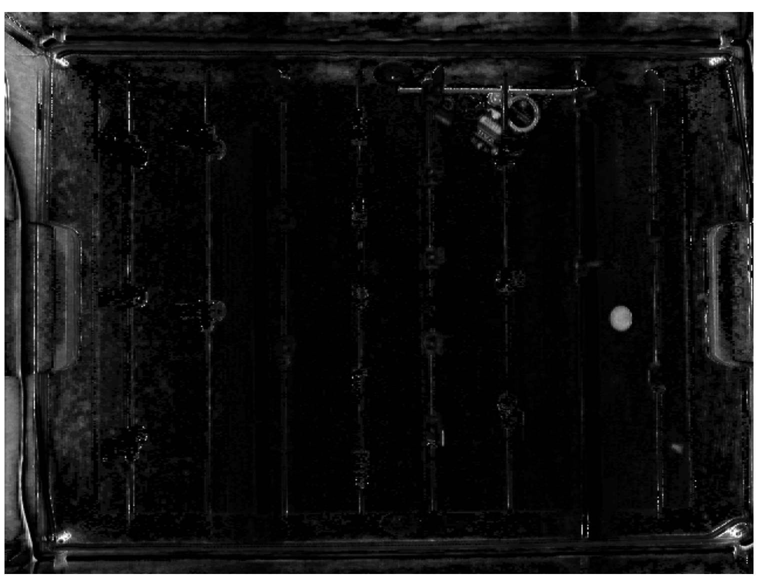


Figure 10. An example of the search frame for the ball.

7 Results

Putting all of these components together, I was able to play my first foosball game against a robot. The system demonstrates great robustness to external visual interference and accurately moves to the correct positions. The dynamic speed and interception routine works effectively to successfully block shots with clear trajectories.

In a real game, however, everything happens very quickly, and quick trajectory changes can be difficult to catch at a camera frequency of 33 Hz. This limitation results in occasional mistakes in the tables play. The system is easily tricked with strategic ball movement into blocking incorrect areas.

Overall, while the system performs well under controlled conditions, its performance emphasizes a need for improvements in processing speed, complex control algorithms and adaptable game strategies.

8 Discussion of Future Works

The Autonomous Foosball Table is an effective prototype and leaves plenty of room for improvement.

One of the biggest improvements that could be made is the vision routine frequency. The vision routine I use is highly susceptible to the size of the image. If I split the image into smaller sections and search these in an order that is changes based on the balls likeliness to be there, then the template matching routine could be sped up significantly.

The current sensors do a good job at executing accurate and fast motions but one way to make the system even better would be to close the loop on the stepper motors. If I had an encoder or potentiometer that could be attached to the stepper motors, that could safeguard the system from any belt skips and would also allow the code to read exactly where the rods are at any given time. The current prototype doesn't wait for any feedback from the Arduino and thus assumes rod positions are reached instantly. Additionally, the current prototype cannot detect player positions, which could supplement the knowledge it uses to execute strategy.

The control portion of the code could benefit from a more complex algorithm. Given more time, a feedback control system could be developed that takes in the current rod positions, angles, angular velocities, and the ball position and velocity to draw a more accurate picture of the balls trajectory that can be influenced by the foosmen. This could allow more complex shooting and ball control algorithms that could direct a ball to a location. If the game could be successfully simulated in this manner, it would open the door to the development of more complex behavioral strategies like passing through machine learning development algorithms.

9 Conclusion

The current prototype effectively operates as an autonomous foosball table. The behavior of the table stands to be improved but significant progress has been made. This project encompasses a wide breadth and depth of fields. My time working on this has taught me a lot about the automation process and the different steps required to divide and conquer particular tasks. I look forward to improving my design and hope to continue my career testing my knowledge and skills through automation.

10 Bill of Materials

Purchased Components				
Part	No.	Supplier	Quantity	Cost
Physical Table Construction				
Red Oak Boards	1042805	Menards	2	80
Oak Plywood 3/4", 4'x8'	796766	Lowes	2	179.56
Wood Finish	91864A018	Lowes	1	25.00
Wood Stain	90128A108	Lowes	1	13.48
Tempered Glass 1/4", 27"x49"	N/A	G and G Glass	1	100.00
Angle Brackets	3632116	Lowes	1 pack	9.16
Binding Post Screws	148274	Lowes	16	16.00
1/4-20 Threaded Inserts	N/A	Amazon	2 packs	7.99
2" OD Oak Rod, 36" Long	96825K86	McMaster Carr	1	52.32
Rubber Grommets	9600K318	McMaster-Carr	1	15.76
Motors				
mj5208 BLDC Motors	N/A	MJBots	4	260.00
Moteus r4.11 Controller	N/A	MJBots	4	319.96
mjcanfd-usb	N/A	MJBots	1	39.00
Nema23 Stepper Motor	23HS45-4204S	StepperOnline	4	102.84
Stepper Drivers	DM542T	StepperOnline	4	65.46
Miscellaneous				
Metric hardware kit	N/A	Amazon	1	9.98
Yellow Foosballs	N/A	Amazon	1	9.98
Capture Card	N/A	Amazon	4	9.98
Power Supply	N/A	Amazon	1	50.00
Linear Rails	N/A	Amazon	1	28.88
LED light strip	N/A	Amazon	1	12.99
Sum	1412.16			

Many components were obtained at no cost from sources including: broken donated foosball table, scrapped parts from previous projects, parts available from the lab I worked in. They are listed in the following table

Procured Components		
Part	Source	Quantity
1/2" Solid Steel Rods	Emerson Manufacturing Shop	8
Leg levelers	Broken Foosball Table	4
Foam Window Insulation	Fluids Laboratory	2 strips
GoPro Hero 5 Black	Past Project	1
Arduino Uno	Past Project	1
Aluminum Extrusion (500mm)	Past Project	10
Green LED	Lab	1
Various EC3 and JST Headers	Lab	31 Total
GT2 Pulley and Belt	Past Project	8 pulleys
Extrusion Brackets	Lab and Past Projects	25 Total
Heatshrink	Lab	N/A
Variety of Wires	Lab and Past Projects	N/A
22 x 7 x 8 Skate Bearings	Past Projects	12