## horizontal line



Final Technical Report (DRAFT)

Robotic Air Hockey System  
2018.07.30

David Eelman - 6365316  
Stanislav Rashevskyi - 7028178  
Thomas Abdallah - 7141518

Conestoga College

Electronic Systems Engineering  
Capstone Project II - EECE74135

Semester 8, Class of 2018

[horizontal line 0](#_Toc520710463)

[Abstract 2](#_Toc520710464)

[1 Introduction 2](#_Toc520710465)

[1.1 Background 2](#_Toc520710466)

[1.2 System Description 2](#_Toc520710467)

[1.3 Literature Review 3](#_Toc520710468)

[Existing Air Hockey Robots 3](#_Toc520710469)

[Methods for Tracking Objects in Motion 3](#_Toc520710470)

[Mechanisms for High Speed Position Control 3](#_Toc520710471)

[1.4 Proposed Solution and Rationale 4](#_Toc520710472)

[2 Design and Test Methods 5](#_Toc520710473)

[2.1 Paddle Controller 5](#_Toc520710474)

[2.2 Master Controller 8](#_Toc520710475)

[2.3 Puck Tracker 10](#_Toc520710476)

[2.4 User Interface 11](#_Toc520710477)

[2.5 Test Methods and Plan 13](#_Toc520710478)

[2.6 Implementation 13](#_Toc520710479)

[2.7 Safety and Impact on Environment and Society 13](#_Toc520710480)

[3 Results and Analysis 14](#_Toc520710481)

[3.1 Qualitative Analysis 14](#_Toc520710482)

[3.2 Measurement Results 14](#_Toc520710483)

[3.3 Quantitative Analysis 14](#_Toc520710484)

[4 Conclusion and Recommendations for Further Work 14](#_Toc520710485)

[5 Appendices 15](#_Toc520710486)

[Appendix A – Coordinate System 15](#_Toc520710487)

[Appendix B – UI Images 16](#_Toc520710488)

[6 References and Bibliography 18](#_Toc520710489)

# Abstract

This document will detail the technical details of the engineering design, integration, and validation of the robotic air hockey system developed for the Capstone design project course during semester 7 and semester 8 of the Electronic Systems Engineering program at Conestoga College.

# 1 Introduction

## 1.1 Background

The Robotic Air Hockey System will be capable of autonomously playing air hockey against a human player. This project will attempt to address the problem of a lack of public knowledge about the ESE program. This problem needs to be solved as there is limited marketing material available that showcases the technical knowledge and capabilities taught in the ESE program that can be understood by both technical and non-technical audiences. Our project will allow group members to develop industry relevant technologies while applying them to a fun and interactive game that will add value to the ESE program through public demonstrations.

## 1.2 System Description

The Robotic Air Hockey System shall be comprised of two major devices, a robotic system, and the supervisory controller system (Figure 1). The robotic system (referred to as the Paddle Controller) is an embedded system responsible for controlling the movements of the air hockey robot based on commands received over CAN. The supervisory controller system is comprised of three software components, each running as a separate process on a single computer, communicating with each other using inter-process communication.

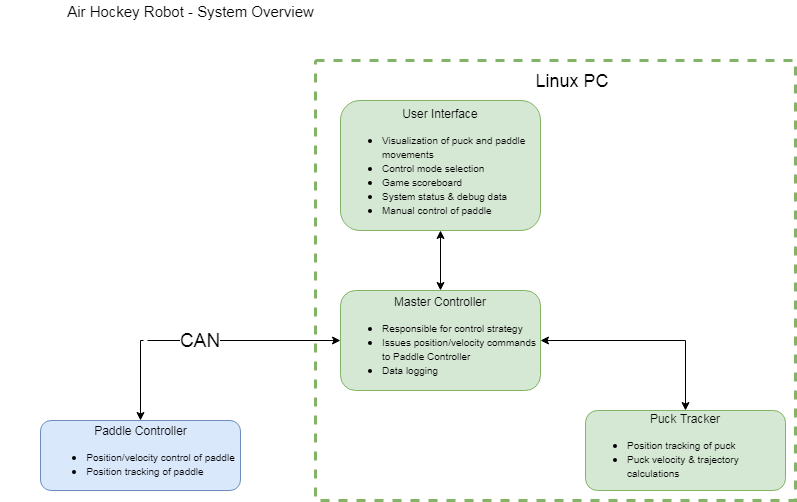


Figure 1 - System Block Diagram

## 1.3 Literature Review

The Robotic Air Hockey System will rely on successfully integrating several unique technologies into one holistic system. This section will review literature covering examples of air hockey playing robots, methods for tracking objects in motion, and mechanisms for high speed position control.

Existing Air Hockey Robots

This project was inspired by an open-source project built by jjrobots [1] [2]. The jjrobots design uses an Arduino to drive three stepper motors from a 3D printer to control the movements of the air hockey paddle. In the version 1 of the jjrobots design, a PC running OpenCV is used to track the positions of the puck and paddle which are sent to the Arduino with a serial connection. In version 2 of the design, an Android smartphone running a custom application was used to track the puck and paddle positions while also providing a user interface for the system. The jjrobots system implemented three distinct control strategies: defense, offense, and offense + defense. Our project would implement a simple defense strategy at first, and more advanced strategies once the core platform is stabilized. Our proposed project was inspired by the jjrobots design, but our implementation will be our own work with the detailed system design being driven by system level performance requirements.

Methods for Tracking Objects in Motion

This project will require accurate real-time position tracking of the air hockey puck in order to calculate its trajectory and velocity. There are a multitude of methods that can be used to track the position of an object. Table 1 summarizes several technologies that may be considered for this project. At this phase in the project we have not committed to any specific technical solution for this problem.

Table 1 - Technologies to Track Objects in Motion

|  |  |  |  |
| --- | --- | --- | --- |
| Technology | Description | Pros | Cons |
| Magnets [3] | Array of sensors directed toward moving magnetic object. | * Low cost * Low power | * Inaccurate measurements * High complexity to track small object on large surface |
| Computer Vision [4] [5] [6] | Camera images processed by a computer to locate specific shapes or colours. | * Open-source solutions available * Accurate tracking of objects in motion | * Requires large amount of processing power * Susceptible to changes in lighting conditions |
| Ultrasonic Sensors [7] | Array of sensors directed toward moving object. | * Low cost * Readily available | * High complexity to track small object on large surface |

Mechanisms for High Speed Position Control

The air hockey paddle will need to move very quickly in order to compete with the best human players. Research into high speed linear position control mechanisms was conducted in order to prepare for the detailed design phase of the project (Table 2). System level performance targets developed in the next phase of the project will drive the design of our chosen position control mechanism.

Table 2 - Technologies for High Speed Position Control

|  |  |  |  |
| --- | --- | --- | --- |
| Technology | Application Examples | Pros | Cons |
| Belt Drive [1] [2] | * Air hockey robot (jjrobots) * 3D printers | * Low cost * Scales to larger/smaller travel distances * Off the shelf components available | * High speed rotating components * Moderate mechanical complexity * Low mass |
| Linear Electric Motors [8] | * Maglev trains * High speed pick and place machines | * High speed * Few moving parts | * Custom designed * High cost * High mass |
| Linear Actuators [9] [10]  (Electromechanical, pneumatic, hydraulic) | * Industrial automation * Heavy equipment * Active suspension | * Few moving parts * High force output | * High cost * Fixed stroke lengths * High mass |

## 1.4 Proposed Solution and Rationale

This project will improve upon an existing open-source design that was implemented using hardware and software from a 3D printer [1]. The proposed system shall utilize the open-source computer vision library OpenCV to track the position of the puck on the playing surface and calculate its speed and trajectory. A mechanical system capable of moving an air hockey paddle in two-dimensions shall be implemented that can accelerate the paddle to speeds high enough to defend against pucks travelling at speeds up to 10m/s and attack the puck by colliding with it while in motion. The electrical system to control the movement of the air hockey paddle will be implemented using a combination of off-the-shelf hardware and custom designed electronics. The software to control the system shall be our own. A user interface shall be implemented to allow the system to be demonstrated to both technical and non-technical audiences. Thorough documentation and a debugging interface shall be provided to enable ESE staff members to maintain and demonstrate the project after the group members have graduated.

This solution should be capable of competing against an average human player in a game of air hockey and should represent a dramatic performance improvement over the open-source design that was constructed using 3D printer parts [1]. Our solution will also feature many relevant technologies used in industry such as motor driver design and construction, tracking high-speed objects in real-time and intercepting objects in motion in real-time.

# 2 Design and Test Methods

## 2.1 Paddle Controller

The Paddle Controller is comprised of three major components, an electro-mechanical system, embedded hardware that interfaces with the electro-mechanical system and the microcontroller, and embedded software running on the microcontroller.

**Electro-Mechanical System**

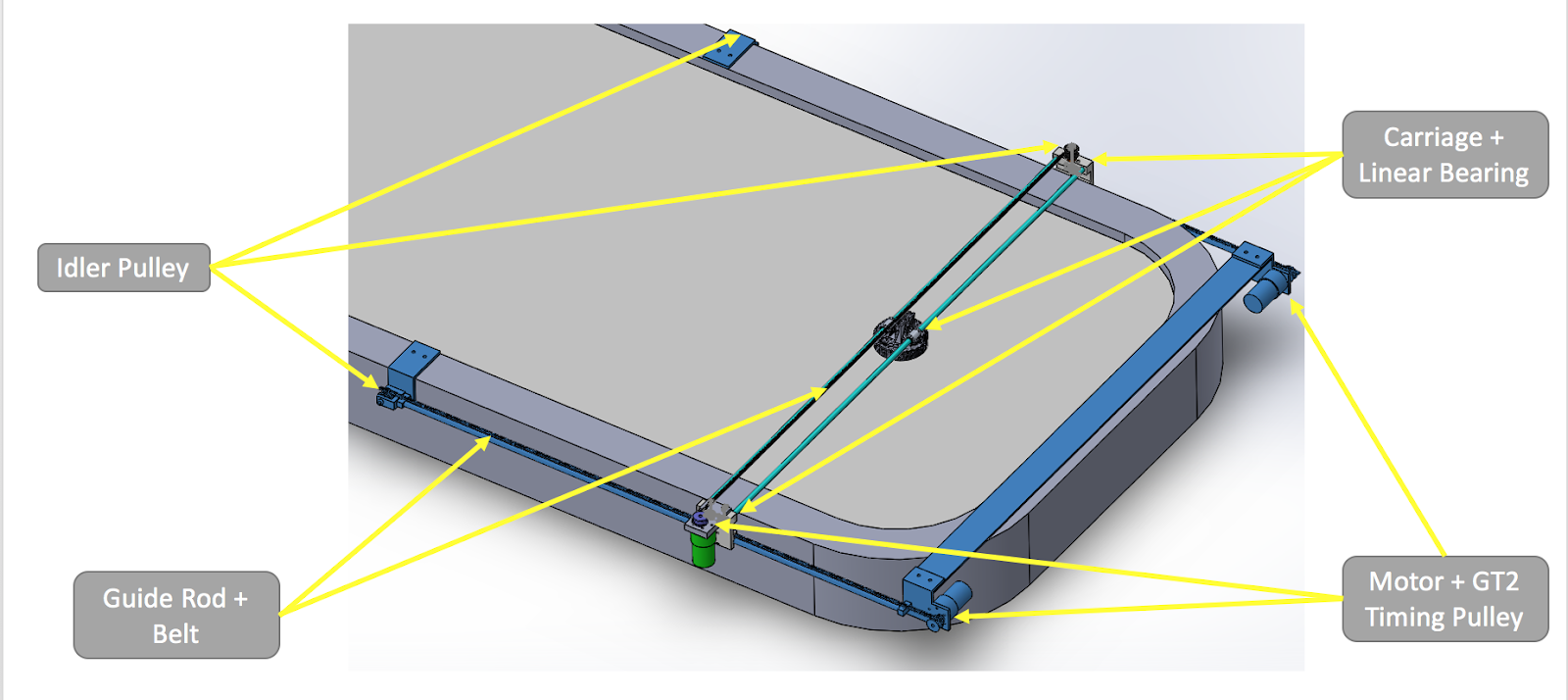


Figure 2 - Electro-Mechanical System Major Components

The electro-mechanical system (Figure 2) is capable of moving the air hockey paddle in two-axes (Figure 9 - Coordinate System). A single brushed DC motor drives a timing belt to move the air hockey paddle left and right in the X-Axis. Two brushed DC motors driving timing belts are used to move the entire X-Axis assembly forward and backward in the Y-Axis.

A quadrature encoder is mounted to the output shaft of each motor enabling accurate control of speed and position for each motor.

Limit switches are mounted at the end of travel on both ends of each of the three linear rails. If the robot travels outside of the playable area and strikes any limit switch, power to all three motors is shut off to stop the robot from moving any further.

A home position switch (simple micro-switch) is mounted on each of the three linear rails allowing the microcontroller to locate a known “home” position after a reset or during a calibration routine.

Infrared break-beam sensors are installed around the perimeter of the electro-mechanical system to detect intrusions into the robots area of movement. If any of the four break-beam sensors is tripped power to all three motors is shut off to stop the robot from moving any further.

An emergency-stop is installed on the human side of the air hockey table. When pressed, power to all three motors is shut off to stop the robot form moving any further.

**Embedded Hardware**

A custom circuit board serves as the interface between the electro-mechanical system and an off-the-shelf microcontroller (Figure 3). Schematics for the circuit board can be found at the end of this document.

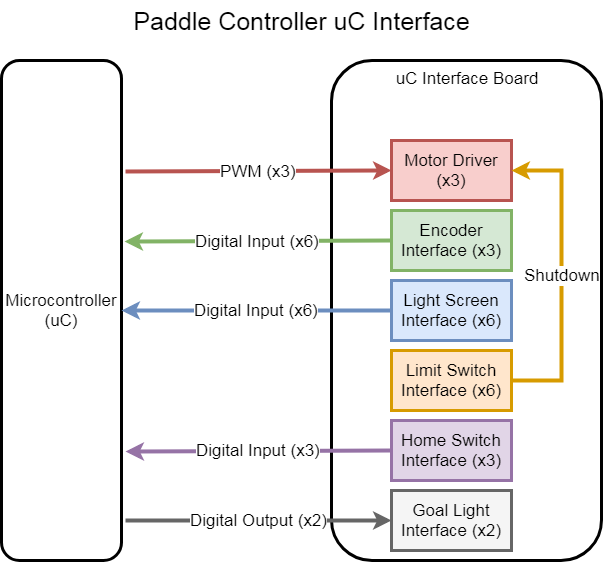


Figure 3 - Microcontroller Interface Board

The circuit board includes three motor driver circuits allowing each brushed DC motor to be controlled using a single PWM signal from the microcontroller.

Both phases of each of the three quadrature encoders are filtered using a Schmidt trigger. The “A-phase” of each encoder triggers an interrupt on the microcontroller where the “B-phase” of the corresponding encoder is read to determine the direction of rotation for the motor. The “B-phase” of each quadrature encoder is latched on the rising edge of its corresponding “A-Phase” using a D-Flip-Flop to ensure that its value does not change prior to being read during the interrupt-service-routine on the microcontroller.

The infrared (IR) “light screens” (or break-beam sensors) are comprised of an IR LED and an IR sensor that is sensitive to 56 kHz pulses of IR light. The IR LED’s are driven by 555 timers operating in a stable mode outputting a 56 kHz waveform. The IR output of the sensors are filtered using a Schmidt trigger and passed to the microcontroller for processing.

The six limit switch outputs are combined using logical AND-gate, with the output driving the “shutdown” pin on each of the motor driver integrated circuits.

The home switch outputs are filtered using a Schmidt trigger and passed to the microcontroller for processing.

Additional circuitry is included to drive two LED “goal lights” which may be implemented in the future.

**Embedded Software**

An off-the-shelf microcontroller running software written in C is used to interface with the motors and sensors via the embedded hardware. The embedded software communicates with the Master Controller using CAN, transmitting information about the air hockey paddle position, velocity, state, and any error information. The Master Controller sends commands to the embedded software to control the position and velocity of the air hockey paddle, as well as issuing state machine commands (Figure 4).

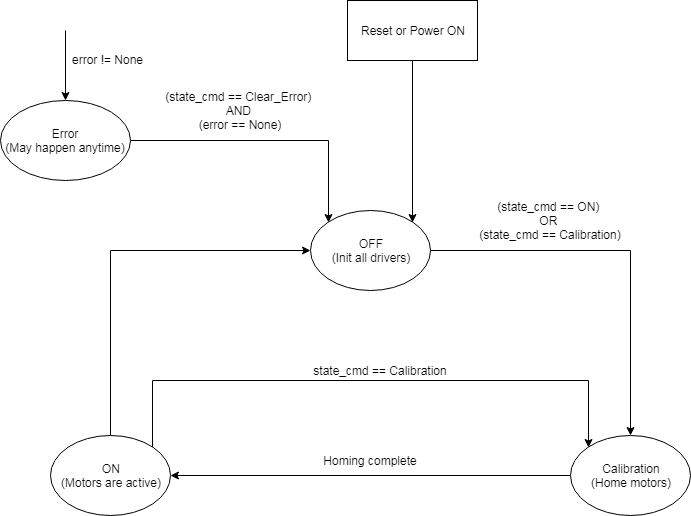


Figure 4 - Paddle Controller State Machine

## 2.2 Master Controller

../5_Master_Controller/MC_flowchart.png

Figure 5 - Master Controller Supervisory flowchart

The Master Controller (MC) is responsible for game strategy control, trajectory predictions, data logging and supervisory control. MC process is written in Python 2.7 and ran on off-the-shelf Linux Computer (i7, 16Gb RAM, Ubuntu 14.04).

Figure 5 demonstrates the constant MC program flow:

**Rx IPC and Tx IPC data**

MC establishes inter-process communication with the Puck Tracker (PT) and the User Interface (UI) processes through Python process-based “threading” interface called *multiprocessing*.

Using multiprocessing, MC creates two arrays (multiprocessing.Array together with enumerations specified in a JSON file) and a queue (multiprocessing.Queue) for communication with the PT.

PT transmit array is mainly used to identify the velocity and position of the puck on the playing surface, as well as to show current states and errors of the process. Receive array is used to configure the web-camera and change PC states. Visualization receive queue is used to pass camera video stream to the MC.

Using multiprocessing, MC creates two arrays (multiprocessing.Array together with enumerations specified in a JSON file) and a queue (multiprocessing.Queue) for communication with the UI. UI transmit array is mainly used to identify current menu page, and send data relevant to that page (e.g. diagnostics interface, manual game paddle positions, settings values, etc), as well as to show current states and errors of the process. Receive array is used to change UI states, list current states and errors of other modules, send scored goals, etc. Visualization transmit queue is used to pass camera video stream with drawn game strategy decisions (puck trajectory predictions, paddle movement, etc.) to the UI.

**Rx CAN and Tx CAN data**

Master Controller communicates with the Paddle Controller board using CAN bus, receiving information about the air hockey paddle position, velocity, goals, state, and any error information. The Master Controller transmits commands to the embedded software to control the position and velocity of the air hockey paddle, as well as issuing state machine commands (Figure 4).

**Log Data**

All received and transmitted messages are stored in HDF5 file and analyzed in Matlab. Debug information is stored in a text file using Python *logging* module.

**Analyze Rx data and Make decision**

In data analysis, MC tracks states, errors, quit requests, game modes, and settings. Then MC makes decisions: calculates paddle position for the automated game; handles manual game (paddle position is filtered and sent from the UI touch display); updates settings; calibrates camera; resolves errors and quit requests, etc.

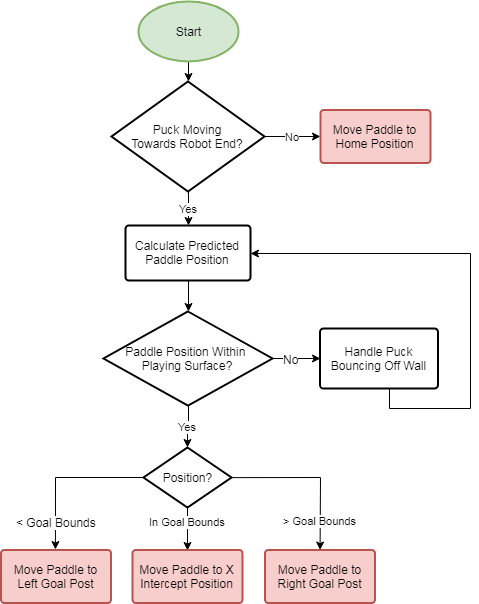


Figure 6 - Complex Defense Algorithm for Automated game

Complex Defense Algorithm for Automated game is shown in Figure 6. Its main purpose is to make sure that paddle defends the goal boundaries at all cost, it has no offensive strategy. An equation of a line (y = mx + b) is a fundamental math concept that lies underneath Calculate Predicted Paddle Position step of the Defense Algorithm. PT from two consecutive camera frames determines two puck position values on the playing surface, this gives us X and Y vectors. Afterward using the vectors and y = mx + b, algorithm predicts the trajectory and final position of the moving puck. Next, each camera frame allows us to average out trajectory and position calculations to improve accuracy.

Offense Algorithm for Automated game is still under development and will depend on the speed, precision, reaction time of the high-speed Paddle Controller system. This algorithm remains the primary risk for the Master Controller, due to the inexperience with advanced control algorithms and dependency on the Paddle Controller electro-mechanical design.

## 2.3 Puck Tracker

The Puck Tracker (PT) is implemented using python and OpenCV - an open source computer vision library. The puck tracker runs as an independent process on a Linux computer and communicates with the Master Controller through inter process communication. The puck tracker utilizes a PlayStation Eye camera mounted above the table and connected to the PC via USB to provide the tracking capabilities. The puck trackers functionality can be split up into two key parts: fiducial locating and tracking of the air hockey puck. See Figure 7 for a detailed block diagram of the puck tracker system.

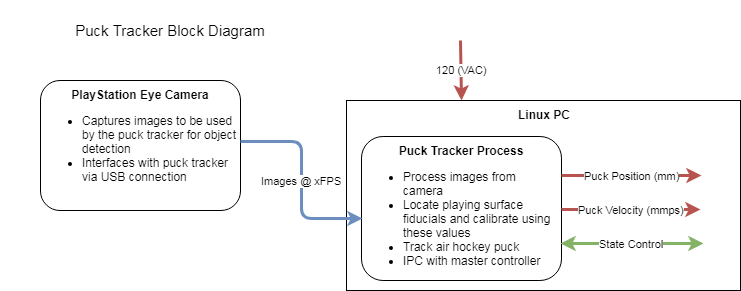


Figure 7 - Puck Tracker Block Diagram

**Fiducial Locating**

The playing surface of the air hockey table has 4 pink two inch circles at the corners. These fiducials (markers) are used by the puck tracker to dynamically frame the playing surface as seen from the camera mounted above the air hockey table. The puck tracker uses OpenCV to detect the fiducials based on a color threshold as well as size, and then saves these fiducial locations to a JSON file. The fiducial locations are then used to compute a transformation matrix which is applied to every incoming frame of the camera to correct the image perspective for the camera. This correction is necessary to make the image appear “top down” to the system even though the camera is mounted at one end of the table. The corrected images are used during puck tracking as explained in the section below.

**Puck Tracking**

When the puck tracker is in normal operation, it is continuously using OpenCV to detect the air hockey pucks location and velocity on the playing surface frame by frame from the incoming video stream of the camera. In order to provide position in mm and velocity in mm per second to the master controller, the puck tracker must have a perspective corrected image of the playing surface (as explained in the fiducial finding section). Once this corrected image is acquired, OpenCV is used to detect the puck within the image. The method to detect the puck involves applying a color threshold to the image in a range that includes just the color of the puck and removes all other colors. The puck tracker then draws contours around all the objects located in the image and checks the radius and area of the located contours to ensure that it is the pucks shape. Once detected, a circle is drawn around the puck and the pixel coordinates are translated to a position in mm relative to the xy coordinate system. Consecutive puck position readings allow us to report velocity of the puck using distance divided by time. The puck position and velocity are reported to the master controller for further use.

## 2.4 User Interface

The User Interface (UI) is implemented using python and Kivy – an open source Python library for rapid development of multi-touch applications – on a 10.1” capacitive touch screen. The UI runs as an independent process on a Linux computer and communicates with the Master Controller through inter process communication. The UI consists of 6 different screen options which can be navigated to using the main menu (Figure 8) and each screen has an option to return to the main menu. The UI screens are described in detail in the sections below.

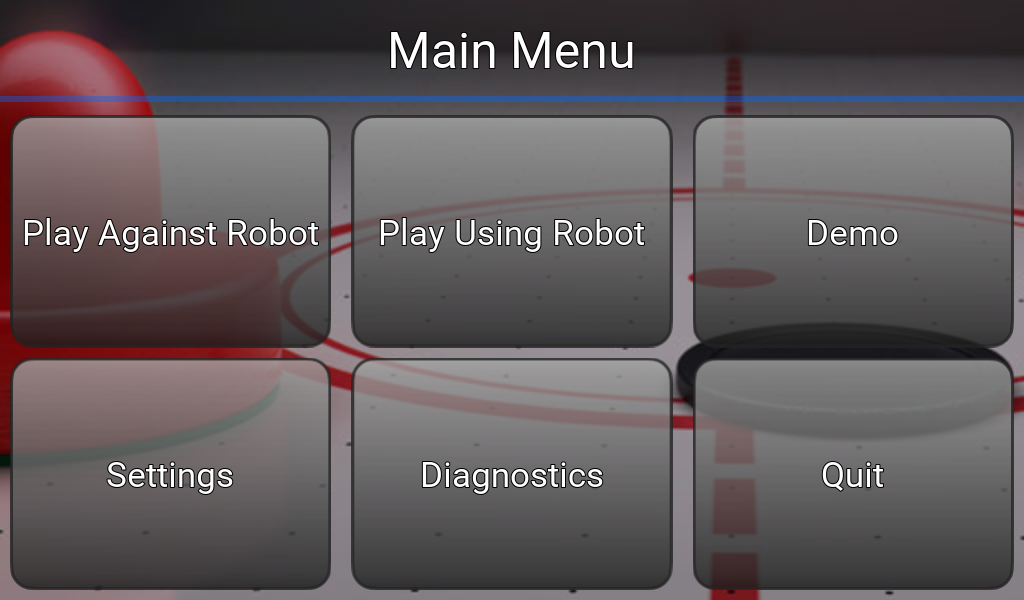


Figure 8 - Main Menu

**Play Against Robot (Appendix B - Figure 10)**

This screen includes a visualization of the puck tracking system and game control for playing against the robot. The visualization portion of this screen shows the tracking of the air hockey pucks movement as well as the predicted path of the puck as it moves towards the robots net and the required paddle controller position to intercept the puck in motion. The game control portion of the screen allows the user to play a timed game against the robot and keeps track of the score using the break-beam sensors of the paddle controller. The user can pause, resume, or reset the game using on screen buttons at any time.

**Play Using Robot (Appendix B - Figure 11)**

The multi-touch screen allows us to implement a method to manually control the paddle controller using a graphical air hockey paddle located on a playing surface on the screen. The screen also includes game control identical to the system described in the section above.

**Demo**

While this screen has yet to be implemented, the motive is to have a predefined routine that the paddle controller can complete to demonstrate its capabilities.

**Settings (Appendix B - Figure 12)**

The settings screen includes options to control the game interface. These options currently include game length (1, 2, or 5 minutes), game mode (defense or offense), and independent control of the x & y axis speeds (slow, medium, or fast).

**Diagnostics (Appendix B - Figure 13)**

The diagnostics screen provides real-time data regarding the state and error reporting of the four core modules (user interface, paddle controller, master controller, and puck tracker). There are buttons to calibrate the paddle controller and puck tracker, as well as a button to clear all errors manually. The paddle controller calibration will be performed entirely by that module; however, the puck tracker calibration includes two additional screens with sliders to adjust the color thresholds of the puck tracking system. These color thresholds occasionally need to be adjusted due to lighting conditions. This calibration routine also includes finding the location of the puck tracker fiducials for when the camera position has been adjusted. All of these settings are saved to a JSON file for long term storage when requested by the user.

**Quit**

The quit option allows the user to quit the User Interface application.

## 2.5 Test Methods and Plan

All modules of the system shall be capable of being tested separately by providing simulated inputs and verifying the correctness of the modules outputs (black-box testing). Detailed component level testing will also be performed where applicable (ex: testing power rails on the Paddle Controller circuit board). The fully integrated system shall be tested under normal operating conditions to verify its function and performance. The fully integrated system shall also be tested while in “error states” to verify correct operation when one or more modules cannot function as designed. All testing shall be non-destructive to maintain the integrity of the prototype. Detailed test plans and results can be found in the Detailed Design Document previously submitted as part of the Capstone project course.

## 2.6 Implementation

Add some evidence of your implementation.

Example:

• Photograph of hardware you implemented and its experimental setup

• PCB layout you developed

• Video clip to show the operation

• Reference to software code you developed (should be placed as an appendix if small, or separate files if large)

## 2.7 Safety and Impact on Environment and Society

This project should have a positive impact on the ESE program by increasing the programs exposure within the academic community, and by providing outward visibility for potential students, employers, and industry partners. This project may also serve as a platform to allow disabled people to play air hockey, or for professional players to train against. This project has no notable impact on the environment as only one prototype has been constructed and mass-production is not planned.

This solution features a high-speed robot operating in close proximity to humans which may pose several safety hazards. If a human were to reach into the operating area of the robot they would be at risk of pinching, crushing, or cutting their fingers and hands. If the robot were to suffer a catastrophic failure while operating at high speeds components of the robot may become projectiles that could strike bystanders.

To mitigate these safety risks the robot has been designed and constructed with several safety features. First, a virtual barrier has been constructed around the robots operating area using IR LEDs and IR sensors to create a “light screen”. If any object crosses any of these virtual barriers and enters the robots operating area power to the motors will be shut off and the robot will stop moving. Second, the motor control software includes overload protection logic that will shut off power to the motors if excessive resistance is detected while applying torque to the motors. Finally, an emergency stop is mounted prominently on the humans’ side of the air hockey table. When pressed, all power to the robot is shut off and the robot will stop moving.

# 3 Results and Analysis

## 3.1 Qualitative Analysis

Present your qualitative test results here. It is expected that you will present results under different test conditions.

## 3.2 Measurement Results

Collect measured data and present those data here. Data tables are expected here.

## 3.3 Quantitative Analysis

Analyze the data to present the key characteristic features. For example, graphs of different kinds may be appropriate here.

# 4 Conclusion and Recommendations for Further Work

The design and integration of the air hockey robot has been highly successful in the scope of this project. The prototype that has been constructed has met all performance targets and can reliably play air hockey against a human player. The User Interface provides easy access for humans to control the air hockey robot and debug the system at a high level. Robustness of the prototype mechanical and electrical systems has been satisfactory.

Further development of the prototype could be conducted to improve performance and robustness, specifically reducing friction in the mechanical system of the Paddle Controller and improving the strength of the 3D printed components. For mass production the system should be redesigned to have all functions run off of a single embedded controller and eliminate the need for an expensive desktop computer. All 3D printed parts would be replaced with injected molded parts, and the mechanical design could be improved to reduce the number of fasteners used to reduce cost.

# 5 Appendices

## Appendix A – Coordinate System

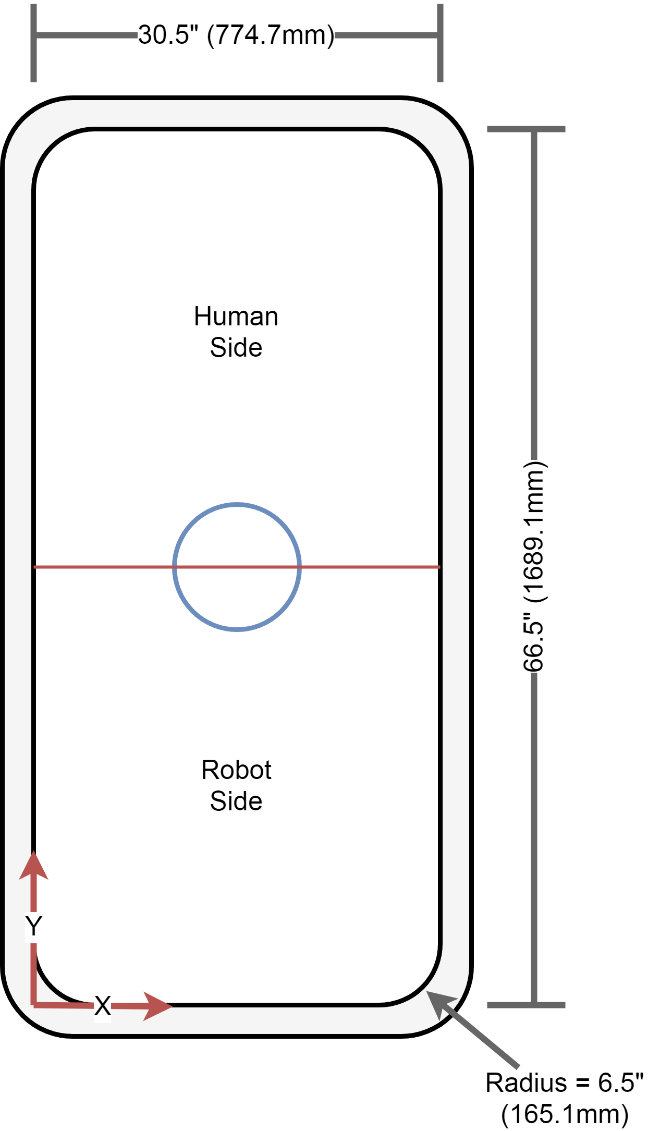


Figure 9 - Coordinate System

## Appendix B – UI Images

****

Figure 10 - Play Against Robot

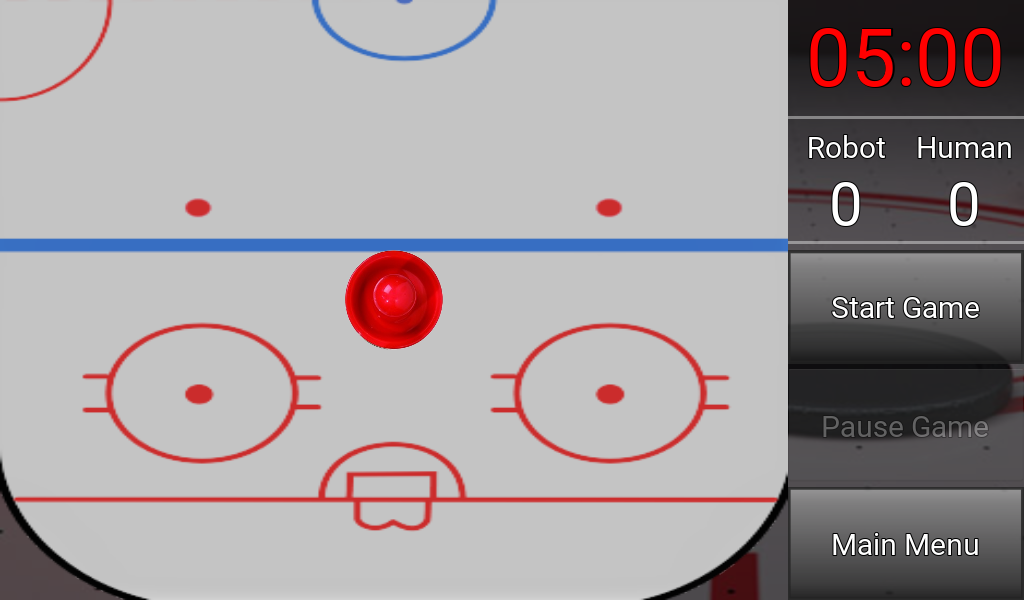
****

Figure 11 - Play Using Robot

****

Figure 12 - Settings

****

Figure 13 - Diagnostics

# 6 References and Bibliography

[1] "Air Hockey Robot (a 3D printer hack)", *jjrobots*, 2018. [Online]. Available: https://www.jjrobots.com/air-hockey-robot-a-3d-printer-hack/. [Accessed: 08- Feb- 2018].

[2] "AIR HOCKEY ROBOT EVO - jjrobots", *jjrobots*, 2018. [Online]. Available: https://www.jjrobots.com/the-open-source-air-hockey-robot/. [Accessed: 08- Feb- 2018].

[3] "Multiple Objects Positioning and Identification Method Based on Magnetic Localization System - IEEE Journals & Magazine", *Ieeexplore.ieee.org*, 2018. [Online]. Available: http://ieeexplore.ieee.org/document/7496898/. [Accessed: 08- Feb- 2018].

[4] H. Parekh, D. Thakore and U. Jaliya, "A Survey on Object Detection and Tracking Methods", *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 2, 2018.

[5] S. A ̊gren, *Object tracking methods and their areas of application: A meta-analysis*. 2018.

[6] "Create an Object Tracking System: Integrating Servo Control with Object Detect", *Allaboutcircuits.com*, 2018. [Online]. Available: https://www.allaboutcircuits.com/projects/create-an-object-tracking-system-integrating-servo-control-with-object-dete/. [Accessed: 08- Feb- 2018].

[7] W. Hamlet and W. Kusewich, "Using The Ultrasonic Sensor to Determine Location", *Engineering.nyu.edu*, 2018. [Online]. Available: http://engineering.nyu.edu/mechatronics/summit/SUMMIT2007/group6-Will\_Bill/Summit2007Project.pdf. [Accessed: 08- Feb- 2018].

[8] "Motors and Selecting the Right One", *Learn.sparkfun.com*, 2018. [Online]. Available: https://learn.sparkfun.com/tutorials/motors-and-selecting-the-right-one. [Accessed: 08- Feb- 2018].

[9] "Linear Positioning - Motion Control Application", *Oriental Motor U.S.A. Corp.*, 2018. [Online]. Available: http://www.orientalmotor.com/applications/linear-positioning.html. [Accessed: 08- Feb- 2018].

[10] "Know Your Pneumatics: Linear Actuators", *Blog.parker.com*, 2018. [Online]. Available:   
 http://blog.parker.com/know-your-pneumatics-hints-tips-for-specifying-linear-actuators.   
 [Accessed: 08- Feb- 2018].