

AIR HOCKEY PLAYING ROBOT

Design Document

ECE 445 Fall 2026

Group 32

Matthew Feder (msfeder2)

Micah Wehler (mwehler2)

Rohan Kapur (rohan21)

University of Illinois Urbana-Champaign

Contents

1	Introduction	1
1.1	Problem	1
1.2	Solution	1
1.3	Visual Aid	2
1.4	High Level Requirements	2
2	Design	3
2.1	Block Diagram	3
2.2	Physical Design	4
2.3	Subsystem Design	5
2.4	Tolerance Analysis	10
3	Cost and Schedule	18
3.1	Cost Analysis	18
3.2	Schedule	19
4	Ethics and Safety	21
4.1	Ethics Safety and Societal Impact	21
	References	23

List of Figures

1	System concept and component locations.	2
2	System block diagram with power, sensing, vision, and motor-control interfaces. . . .	3
3	Mechanical assembly and motor-mount arrangement.	4
4	Dimensioned plan view of the table and H-bot assembly.	5
5	Dimensioned isometric view of the H-bot gantry.	9
6	Playing geometry used in the timing analysis.	11
7	Nominal timing budget at a puck speed of 4 m/s.	13
8	Available paddle movement time versus puck speed.	13
9	Required peak paddle speed for 5, 10, and 15 cm movements.	15
10	Required paddle acceleration for 5, 10, and 15 cm movements.	15
11	Required acceleration versus combined image and computation latency.	16

List of Tables

1	Power subsystem requirements and verification	5
2	Motor-drive requirements and verification	6
3	Sensor subsystem requirements and verification	7
4	Perception and control requirements and verification	8
5	Actuation requirements and verification	10
6	Parameters and assumptions for the tolerance analysis	11
7	Calculated motion requirements for a 10 cm movement	14
8	Estimated bill of materials	18
9	Project schedule and responsibilities	20

1 Introduction

1.1 Problem

Air hockey requires a player to visually track a fast-moving puck, predict its trajectory, and physically intercept it, all in a fraction of a second. An autonomous opponent therefore requires tightly coupled, low-latency perception and control. Racket and paddle sports have been used as benchmarks for real-time robotics since the 1980s because they require sensing, prediction, and rapid actuation [1].

Recent results in table tennis show how demanding this problem remains even with substantial engineering resources. In 2026, Sony AI researchers published an autonomous table tennis system in *Nature* that competed with elite human players. The work connects high-speed perception and control to broader applications in dynamic physical environments [1]¹.

For this project, air hockey provides a compact tabletop setting for studying perception and control within the scope, budget, and schedule of a senior design project. We consider straight and single-bounce puck trajectories in the plane of the table. A missed intercept can result in a goal. The broader motivation is to investigate fast responses to human-driven motion, with potential relevance to collision avoidance and human-robot interaction [1].

Our goal is to build a robotic air hockey opponent that autonomously tracks and intercepts an incoming puck to defend against and return shots, using vision-based perception and a stepper-driven actuation system, on a compact tabletop air hockey table.

Additionally, this project aims to address the issue of accessibility in the sport of air hockey. Currently, the sport requires two players to play, which can make practice and general enjoyment difficult when no peers are present. This requirement creates a barrier to entry for people who lack consistent playing partners.

1.2 Solution

Our solution uses an overhead camera to track the puck, a Raspberry Pi to estimate its position and velocity, and an STM32 to predict an interception point and control a paddle mounted on an H-bot mechanism. The detailed design separates power, motor drive, sensing, perception and control, and mechanical actuation.

The actuation subsystem moves the paddle to the intercept point. Two stationary NEMA 17 stepper motors drive a shared belt in an H-bot configuration, providing two degrees of freedom. The STM32 configures the motor drivers over SPI and generates motion commands, as described in Section 2.3.2.

The sensing subsystem adds a pair of IR break-beam sensors at fixed, known positions on the table, independent of the camera. These sensors provide ground-truth measurements of when the puck crosses a given point, which are used to validate and correct the vision system's position and velocity estimates, and to help isolate failures to either the perception or actuation subsystem during testing.

In addition to addressing the robotics issue spoken about above, this device allows users to practice and enjoy the game independently. By providing a robotic opponent, the issue of finding a second player is removed. This increases accessibility to the sport for individuals who desire to practice or play alone.

¹The study was published in April 2026; the reported matches were conducted in April 2025.

1.3 Visual Aid

Figure 1 shows the proposed arrangement. All figures and charts are original project-team work.

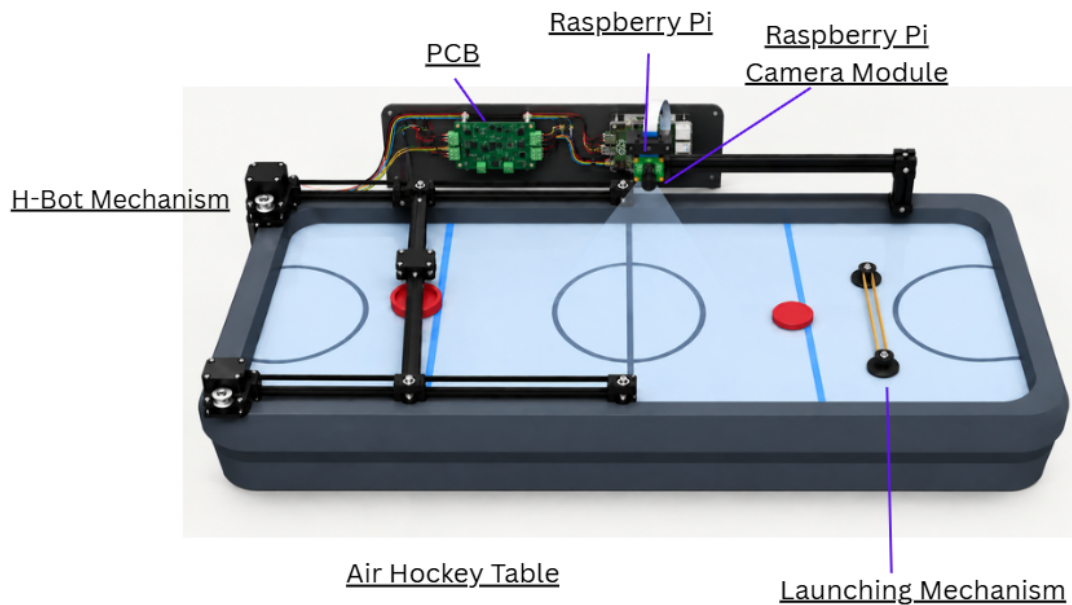


Figure 1: System concept and component locations.

Launching Mechanism Used for velocity repeatability for the puck.

Raspberry Pi Camera Module Used to record the table.

Raspberry Pi Used to detect and calculate puck movement and velocity.

PCB Houses STM32 and motor drivers, which calculate intercept point and control motor movement.

H-Bot Mechanism Used to move the paddle to defend shots.

Air Hockey Table Table used to play against our mechanism.

1.4 High Level Requirements

- HLR1.** The system must correctly detect the puck's position in at least 95% of frames recorded by the Raspberry Pi camera module.
- HLR2.** The system must calculate the puck's velocity vector and command the paddle to its predicted intercept point within 30 ms of the puck entering the camera's field of view, to leave sufficient time for the paddle to physically reach the target before the puck crosses the goal line.
- HLR3.** The system must make paddle contact with at least 75% of incoming straight shots traveling at speeds up to 4 m/s, and must make paddle contact with at least 50% of incoming single-bounce shots traveling at speeds up to 4 m/s.
- HLR4.** The system must move the paddle to within 5 mm of the predicted intercept point on at least 95% of shots.

2 Design

2.1 Block Diagram

The system interfaces are shown in Figure 2.

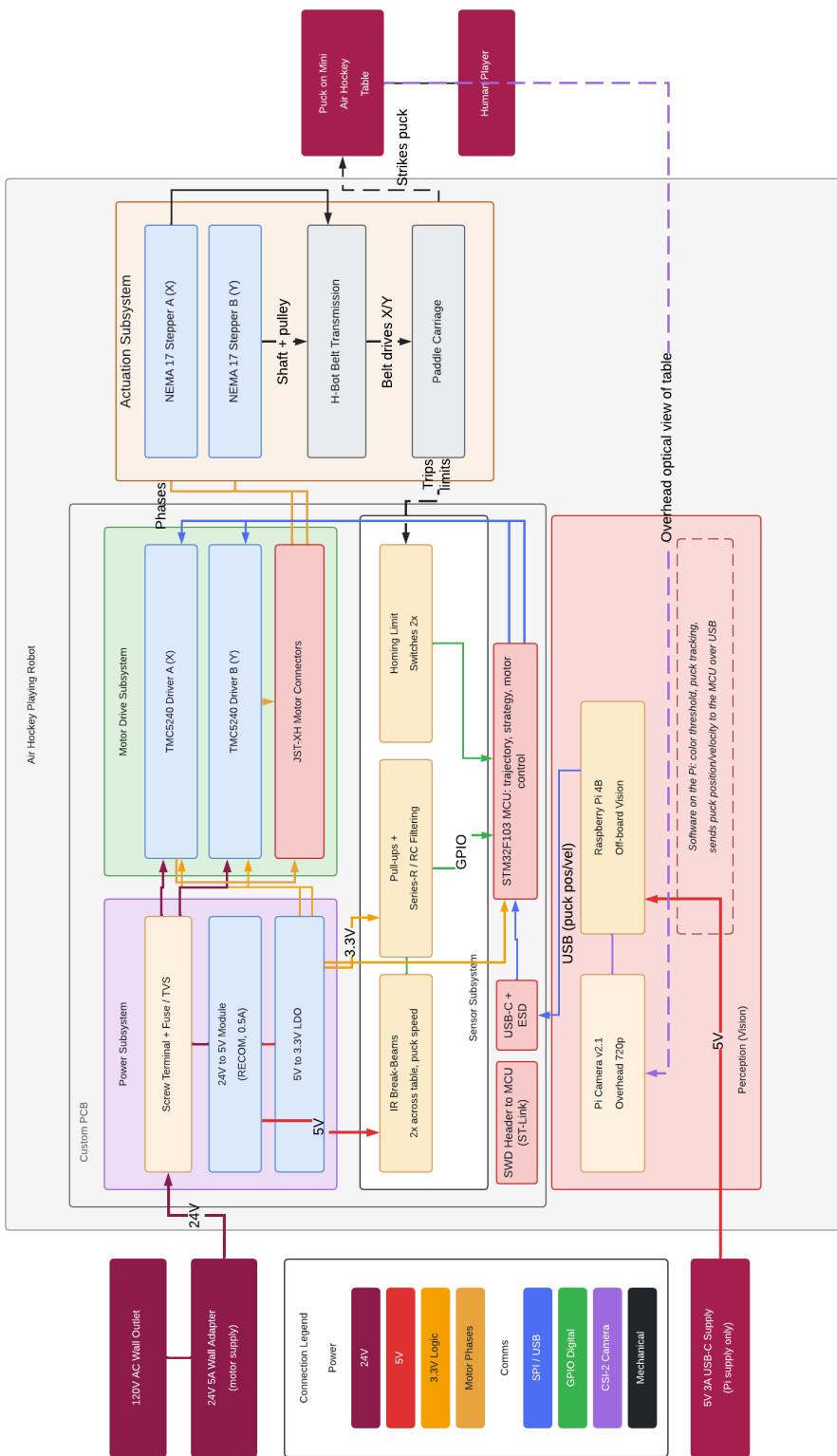


Figure 2: System block diagram with power, sensing, vision, and motor-control interfaces.

2.2 Physical Design

The paddle movement system uses an H-bot XY gantry driven by two NEMA 17 stepper motors. The custom motor mounts shown in Figures 3 and 4 rigidly attach the motors to the frame and align each pulley with the belt path while leaving clearance for the motor body and belt. The two motors remain fixed to the frame, while the shared belt system moves the paddle carriage in both the X and Y directions depending on the relative direction of motor rotation. Keeping the motors stationary reduces the moving mass of the gantry, which helps the paddle reach the acceleration and speed required for fast puck interception. The mount geometry is also designed to maintain consistent pulley alignment and belt tension during operation.

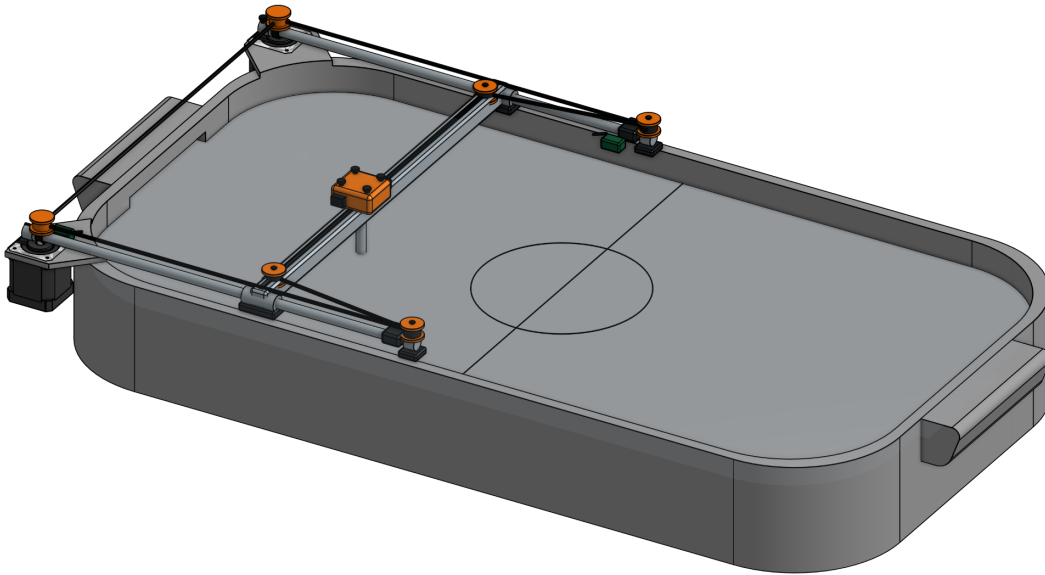


Figure 3: Mechanical assembly and motor-mount arrangement.

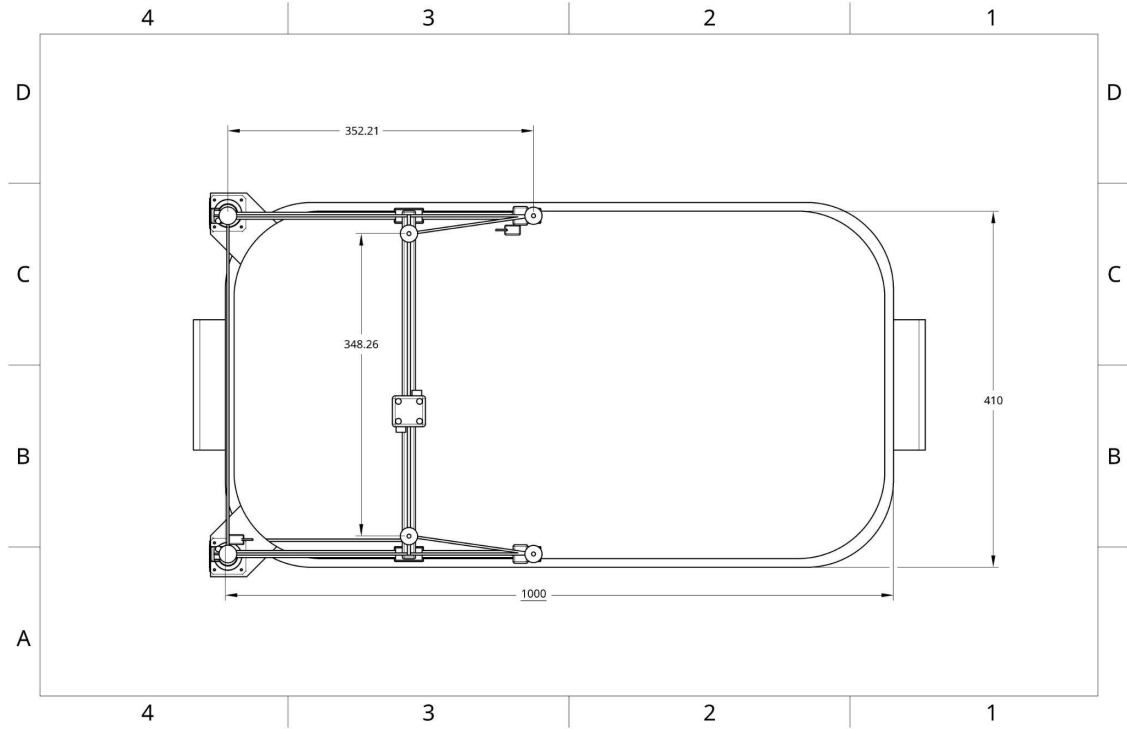


Figure 4: Dimensioned plan view of the table and H-bot assembly.

2.3 Subsystem Design

2.3.1 Power Subsystem

The power subsystem distributes and regulates power for the motor drivers, STM32, and sensing electronics. A 24 V bench power supply is used as the primary input and supplies the two TMC5160A motor drivers directly. A buck converter generates a regulated 5 V rail, while a 3.3 V regulator supplies the STM32 and other logic-level components. The Raspberry Pi is powered independently to prevent high-current motor transients from directly affecting the vision computer. Input protection is included to protect the PCB from overcurrent and voltage transients.

Table 1: Power subsystem requirements and verification

Requirement	Verification
The subsystem shall accept a 24 V $\pm 5\%$ DC input during normal operation.	Use a bench power supply to vary the input from 22.8 V to 25.2 V. Measure each output rail using a DMM and record the results in a table.
The 5 V rail shall remain between 4.75 V and 5.25 V throughout its expected operating load range.	Apply loads from minimum operating load through the calculated maximum load and measure the rail with a DMM. Plot output voltage versus load current.

Continued on next page

Table 1 continued

Requirement	Verification
The 3.3 V rail shall remain between 3.14 V and 3.47 V while the STM32, sensors, and motor-driver logic are active.	Measure the 3.3 V rail with an oscilloscope while both motors repeatedly accelerate and decelerate. Record the minimum and maximum observed voltage.
Both TMC5160A drivers shall receive at least 22.8 V while both motors operate simultaneously.	Probe the VM input of each driver while executing the maximum-load motion sequence and record the minimum measured voltage.
Motor operation shall not cause either regulated logic rail to leave its specified range.	Monitor the 5 V and 3.3 V rails with an oscilloscope during repeated motor starts, stops, and direction changes. Record maximum transient deviations.

2.3.2 Motor Drive Subsystem

The motor drive subsystem converts motion commands generated by the STM32 into the phase currents required by the two NEMA 17 stepper motors. Two TMC5160A drivers receive 24 V motor power from the power subsystem and are configured by the STM32 over SPI. The STM32 generates independent STEP and DIR signals for each motor according to the desired H-bot trajectory. This subsystem therefore provides the electrical interface between the low-voltage control system and the high-current actuation system.

Table 2: Motor-drive requirements and verification

Requirement	Verification
The subsystem shall independently control two NEMA 17 stepper motors.	Command Motor 1 while Motor 2 remains stationary, then repeat for Motor 2. Record whether each motor responds independently and in the commanded direction.
Each TMC5160A shall correctly receive 3.3 V SPI and STEP/DIR signals from the STM32.	Connect a logic analyzer to SCK, MOSI, MISO, CS, STEP, and DIR. Send known commands and verify correct logic levels and communication.
SPI communication shall succeed for at least 99.9% of 10,000 transactions during normal motor operation.	Repeatedly write and read known TMC5160A registers while both motors operate. Log failed transactions and calculate the success rate.
Motor phase current shall remain at or below the selected motor's 2.0 A/phase rating.	Measure phase current using a current probe or the driver sense resistor while commanding the highest-current operating condition. Record the peak current.
The subsystem shall execute 100 consecutive direction changes without an incorrect motor direction or unrecoverable driver fault.	Run a scripted forward/reverse motion sequence 100 times and log the driver status and resulting motor direction.

2.3.3 Sensor Subsystem

The sensor subsystem consists of two IR break-beam sensors and two mechanical homing switches connected directly to STM32 GPIO. The two IR sensors are mounted at fixed, known locations

and provide independent puck-crossing measurements that can be compared with the camera-based position and velocity estimates. The two homing switches establish a repeatable X- and Y-axis reference position for the H-bot before gameplay begins. Connecting these sensors directly to the STM32 allows the same processor responsible for physics calculations and movement to react to sensor events without relying on the Raspberry Pi.

Table 3: Sensor subsystem requirements and verification

Requirement	Verification
The subsystem shall contain two IR break-beam sensing locations at fixed positions known to within ± 2 mm.	Measure each beam position relative to the table coordinate system three times using a ruler or calipers. Record the measured coordinates and maximum variation.
Each IR beam shall successfully detect at least 99 of 100 puck crossings for puck speeds from 0.5 to 4.0 m/s.	Launch the puck through each beam 100 times across the operating speed range. Compare GPIO detections against high-speed video and record missed crossings.
The delay between beam interruption and the corresponding STM32 GPIO transition shall be ≤ 1 ms.	Probe the sensor output and STM32 GPIO input simultaneously with an oscilloscope. Measure propagation delay over at least 20 events.
The subsystem shall contain two independent homing switches, providing X- and Y-axis reference information.	Move the carriage away from home and execute the homing procedure. Verify that the STM32 independently detects both switches.
Repeated homing shall return the carriage to within ± 1 mm of its reference position over 20 cycles.	Perform 20 complete homing cycles and measure the final carriage location relative to a fixed reference. Record the maximum deviation.
Sensor signals presented to the STM32 shall remain between 0–0.8 V for LOW and 2.0–3.3 V for HIGH.	Measure each sensor output in both logical states with an oscilloscope and record the maximum LOW and minimum HIGH voltages.

2.3.4 Perception and Control Subsystem

The perception and control subsystem is divided between the Raspberry Pi and STM32. The overhead camera captures the playing surface at a target resolution of 720p and 120 FPS. The Raspberry Pi performs the vision pipeline, including frame acquisition, color-threshold puck detection, and estimation of the puck position and velocity. The resulting puck state is transmitted to the STM32 over USB. The STM32 then applies the table physics model, predicts the desired interception point, generates the required H-bot trajectory, and sends SPI and STEP/DIR commands to the motor-drive subsystem. The two IR sensors and two homing switches are also connected directly to the STM32. This division allows the Raspberry Pi to handle computationally intensive vision while the STM32 performs the time-critical physics and motor-control tasks. The project-level timing requirement is for perception through movement-command generation to complete within 30 ms.

The USB message from the Raspberry Pi to the STM32 contains the estimated puck position (x, y) , velocity (v_x, v_y) , a timestamp, and a validity indicator. The timestamp allows the STM32 to reject stale measurements, while the validity indicator prevents motion commands from being generated when the puck has not been reliably detected. The STM32 uses the received state to propagate the puck trajectory forward in time, account for wall collisions when necessary, and determine the paddle position required for interception.

Table 4: Perception and control requirements and verification

Requirement	Verification
The camera and Raspberry Pi shall process images at a target rate of 120 FPS during normal tracking.	Log camera frame timestamps for at least 10 s while the final vision pipeline is operating. Calculate the achieved frame rate and report dropped frames.
The vision pipeline shall successfully detect the puck in at least 95% of usable frames under the lighting conditions used for testing.	Manually label at least 500 frames in which the puck is visible and compare them against software detections. Calculate the detection percentage.
For straight shots, the Raspberry Pi velocity estimate shall be within $\pm 10\%$ of the velocity measured using the two IR sensors over the 0.5–4.0 m/s operating range.	Determine ground-truth velocity from the known IR-beam separation and crossing times. Compare this against the camera estimate for at least 20 shots and present measured versus estimated velocity.
Raspberry Pi to STM32 USB communication latency shall be ≤ 2 ms for at least 99% of 1,000 puck-state messages.	Timestamp each message immediately before transmission and at STM32 reception. Record 1,000 measurements and report the latency distribution.
Every valid puck-state message shall contain position, velocity, timestamp, and validity information.	Send at least 1,000 known test messages and log the values decoded by the STM32. Compare transmitted and received values and report any malformed packets.
The STM32 shall calculate the intercept and desired movement trajectory within 5 ms of receiving a valid puck state.	Toggle one STM32 GPIO upon receiving a valid state and another when trajectory generation is complete. Measure the delay with an oscilloscope over at least 100 calculations.
Total time from obtaining sufficient camera information to generating the corresponding motor command shall be ≤ 30 ms.	Instrument the Raspberry Pi and STM32 software with timestamps and/or GPIO timing signals. Run at least 100 shot trials and report average, maximum, and 99th-percentile latency.

2.3.5 Actuation Subsystem

The actuation subsystem physically positions the paddle using a two-degree-of-freedom H-bot gantry driven by two stationary NEMA 17 stepper motors. The H-bot belt arrangement combines the rotations of both motors to generate X- and Y-axis translation of the paddle carriage. Keeping both motors fixed to the frame reduces the moving mass of the mechanism and allows higher paddle acceleration. The STM32 converts the desired Cartesian movement into the corresponding motor-step commands, which are then executed by the motor-drive subsystem. The mechanism must reach all required defensive positions while remaining inside the mechanical travel range established through homing. The gantry geometry appears in Figure 5.

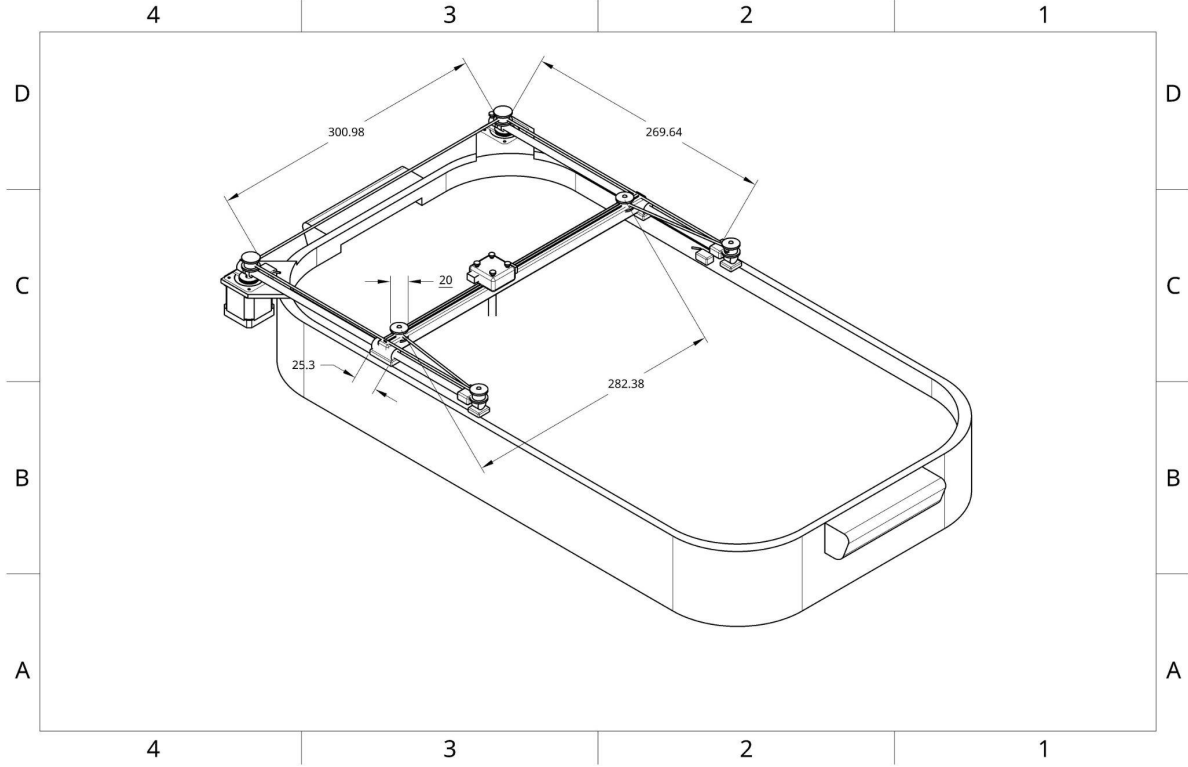


Figure 5: Dimensioned isometric view of the H-bot gantry.

For the selected motor-direction convention, the H-bot motion can be expressed as

$$\Delta M_1 = k(\Delta x + \Delta y) \quad (1)$$

$$\Delta M_2 = k(\Delta x - \Delta y) \quad (2)$$

where k converts linear paddle displacement into motor steps based on the pulley circumference, motor step angle, and selected microstepping setting. The exact motor direction polarity is established during calibration.

From the timing tolerance analysis, a worst-case 10 cm paddle movement for a puck traveling at 4 m/s must be completed in approximately 135 ms. Using a symmetric acceleration/deceleration profile, this corresponds to a required peak paddle speed of approximately

$$v_{\text{peak}} \approx 1.49 \text{ m/s} \quad (3)$$

and a required acceleration of approximately

$$a \approx 22.1 \text{ m/s}^2 \quad (4)$$

These values define the required performance of the completed H-bot rather than the individual stepper motors alone. Equations (1) and (2) define the motor mixing convention.

Table 5: Actuation requirements and verification

Requirement	Verification
The mechanism shall provide controlled movement in two independent translational degrees of freedom.	Command pure X and pure Y movements of at least 100 mm. Measure the resulting displacement and verify unintended perpendicular motion remains below 5 mm.
The paddle shall reach the entire defined defensive workspace without contacting the frame or mechanical hard stops.	Home the mechanism and command the paddle to the corners and boundaries of the defensive region. Measure and record the reachable workspace.
The carriage shall complete a 100 mm stop-to-stop movement within 135 ms under the normal paddle load.	Record at least 20 movements using a high-speed camera at ≥ 120 FPS. Measure the movement duration in each trial and report the mean and maximum.
The paddle shall achieve at least 1.49 m/s peak linear velocity during the 100 mm movement test.	Extract carriage position versus time from the high-speed recording, calculate velocity, and report the measured peak value.
The paddle shall achieve at least 22.1 m/s ² acceleration during the worst-case movement test.	Calculate acceleration from the measured position/velocity data and report the maximum usable acceleration.
After homing, commanded paddle positions shall be accurate to within ± 5 mm throughout the defensive workspace.	Command at least five predetermined X-Y positions, repeat each 10 times, and measure the actual paddle position. Present commanded and measured coordinates in a table.
The mechanism shall complete 100 gameplay-style movements without contacting a hard stop, losing its reference position, or producing an unrecoverable motor-driver fault.	Execute a scripted sequence spanning the defensive workspace while logging motor-driver status and carriage position. Record any failures or collisions.

2.4 Tolerance Analysis

The most critical feature of the air hockey robot is its ability to move the paddle to the predicted puck interception point before the puck reaches the robot. The project requirements specify successful interception of at least 75% of straight shots and 50% of one-bounce shots traveling below 4 m/s. This tolerance analysis focuses on whether the sensing, computation, and H-bot movement pipeline provides sufficient reaction time at the 4 m/s design limit. Trajectory-prediction error for bounced shots will be characterized experimentally once the vision system and physical table are assembled.

For this preliminary analysis, the table is modeled as 1 m long. The puck is assumed to be struck 25 cm from the player’s end of the table and intercepted 10 cm from the robot’s end. These are conservative design assumptions based on the planned physical geometry rather than manufacturer-specified tolerances. The puck therefore travels approximately 0.65 m before interception.

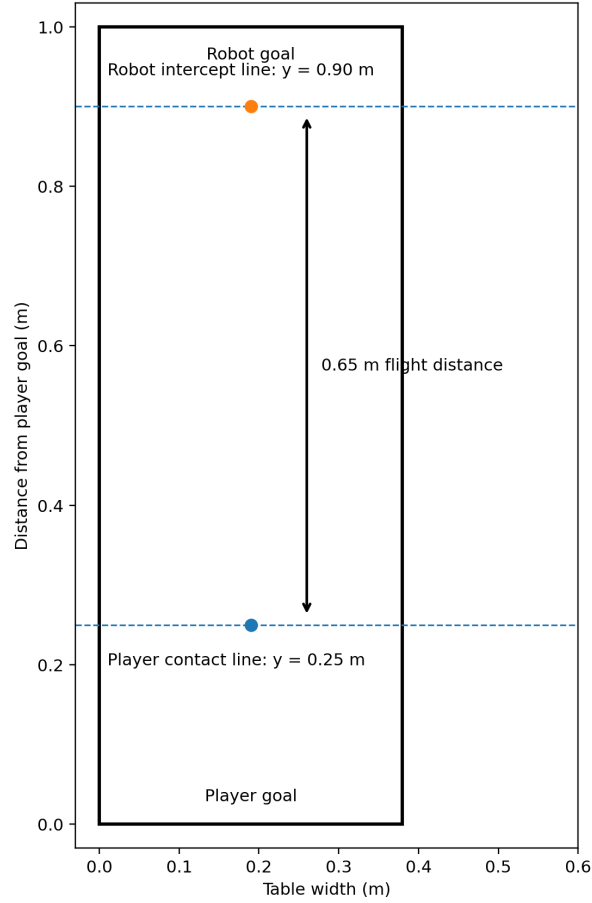


Figure 6: Playing geometry used in the timing analysis.

The parameters used in the analysis are summarized in Table 6. Values describing the physical playing geometry and desired puck speed are project requirements or design assumptions. Hardware values are taken from manufacturer specifications where available.

Table 6: Parameters and assumptions for the tolerance analysis

Parameter	Value Used	Basis
Table length	~1.0 m	Project mechanical design
Player contact position	0.25 m	Conservative analysis assumption
Robot intercept position	0.90 m	Selected defensive intercept line
Puck travel distance	0.65 m	Calculated from geometry
Maximum puck speed	4.0 m/s	Project requirement
Camera frame rate	120 FPS	Camera capability / design target [2]
Image acquisition budget	18 ms	Estimated from frame period + overhead
CV + trajectory calculation	10 ms	Preliminary software estimate
Maximum lateral paddle movement	0.10 m	Motion-planning design constraint
Stepper motor	17HE19-2004S	Selected component [3]

Continued on next page

Table 6 continued

Parameter	Value Used	Basis
Stepper driver	TMC5160A	Selected component [4]
Loaded H-bot speed/acceleration	To be measured	Experimental verification required

The selected 17HE19-2004S stepper motor is specified as a 1.8° bipolar NEMA 17 motor rated for 2.0 A per phase with a holding torque of 55 N·cm [3]. The TMC5160A motor controller supports STEP/DIR and SPI control and an 8–60 V motor supply, making it compatible with the planned 24 V motor power system [4].

2.4.1 Timing Analysis

For a puck traveling at velocity v_p , the time between the player's strike and the defensive interception line is

$$t_{\text{flight}} = \frac{d_{\text{puck}}}{v_p} \quad (5)$$

For the assumed puck travel distance of 0.65 m and the project maximum puck velocity of 4 m/s,

$$t_{\text{flight}} = \frac{0.65}{4} = 0.1625 \text{ s} = 162.5 \text{ ms} \quad (6)$$

The planned camera system operates at up to 120 FPS. For reference, Raspberry Pi's Camera Module 3 supports a 720p120 video mode, and Raspberry Pi documentation lists approximately 120 FPS sensor modes for the IMX708 camera [2]. At 120 FPS, one frame period is²

$$t_{\text{frame}} = \frac{1}{120} \text{ s} \approx 8.33 \text{ ms} \quad (7)$$

Velocity estimation requires observations from at least two frames. Depending on when the puck enters the camera frame relative to image acquisition, two observations can therefore require approximately two frame periods. An 18 ms acquisition budget is used to account for these frame periods plus a small amount of capture overhead. This 18 ms value is an engineering estimate and will be replaced with measured end-to-end camera latency on the completed system.

An additional 10 ms is allocated for puck detection, coordinate transformation, trajectory estimation, and intercept calculation. This is also a preliminary design estimate and must be experimentally measured across the Raspberry Pi and STM32 once the final vision pipeline is implemented.

The remaining time available for physical movement is

$$t_{\text{move}} = t_{\text{flight}} - t_{\text{camera}} - t_{\text{CV}} \quad (8)$$

giving

²The cited 720p120 mode describes Camera Module 3 capture capability. Achieved end-to-end tracking rate depends on acquisition settings and vision processing and must be measured on the completed system.

$$t_{\text{move}} = 162.5 - 18 - 10 = 134.5 \text{ ms} \quad (9)$$

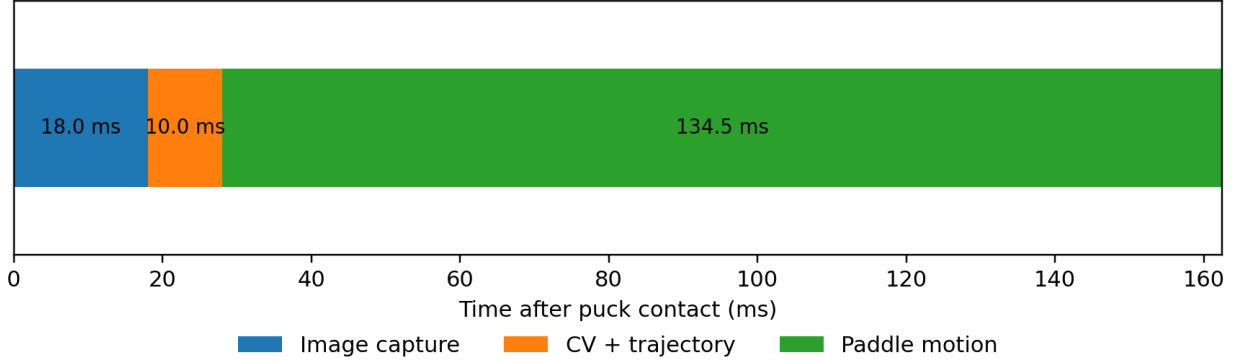


Figure 7: Nominal timing budget at a puck speed of 4 m/s.

The available movement time decreases as puck velocity increases. At the 4 m/s design limit, approximately 134.5 ms remains for paddle movement after the current sensing and computation budgets. Figures 7 and 8 illustrate this budget and its dependence on puck speed.

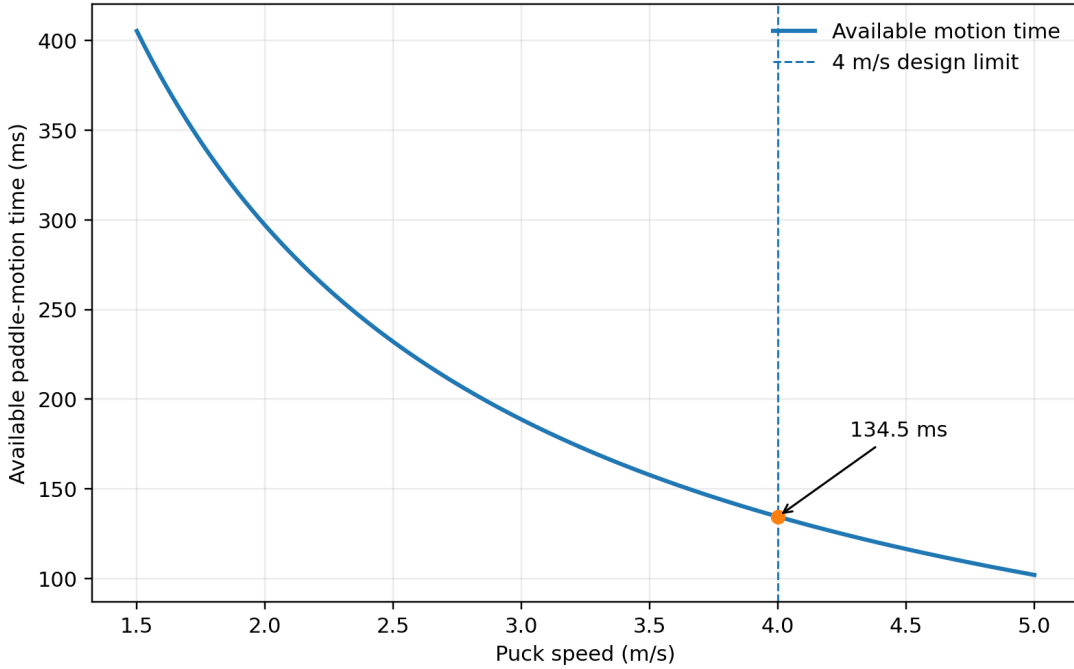


Figure 8: Available paddle movement time versus puck speed.

A straight shot is approximately the worst case for this timing calculation. For an angled shot of the same total puck speed, only the longitudinal velocity component moves the puck toward the robot, increasing the time before it reaches the interception line. One-bounce shots are therefore expected to be more difficult primarily because of uncertainty in the predicted trajectory after contact with the wall rather than because of a smaller timing budget.

2.4.2 H-Bot Motion Requirement

The previous preliminary timing calculation considered only average paddle velocity. For a 10 cm lateral movement completed in 134.5 ms,

$$v_{\text{avg}} = \frac{0.10}{0.1345} \approx 0.74 \text{ m/s} \quad (10)$$

This value does not account for the acceleration required to bring the paddle from rest to its commanded velocity. To approximate this effect, the paddle movement is modeled using a symmetric triangular velocity profile, in which the paddle accelerates for the first half of the movement and decelerates for the second half.

For a displacement d completed in time t ,

$$v_{\text{peak}} = \frac{2d}{t} \quad (11)$$

and

$$a_{\text{req}} = \frac{4d}{t^2} \quad (12)$$

For the 10 cm worst-case movement,

$$v_{\text{peak}} = \frac{2(0.10)}{0.1345} \approx 1.49 \text{ m/s} \quad (13)$$

and

$$a_{\text{req}} = \frac{4(0.10)}{(0.1345)^2} \approx 22.1 \text{ m/s}^2 \quad (14)$$

Therefore, the assembled H-bot must be capable of approximately 1.5 m/s peak paddle velocity and 22 m/s² acceleration while carrying the complete paddle assembly. These values are calculated system requirements, not specifications guaranteed by the motor manufacturer. The selected motor's manufacturer provides a motor torque-speed curve, but the achievable linear acceleration also depends on pulley radius, moving mass, belt tension, friction, motor supply voltage, and the coupled H-bot geometry [3]. Consequently, final compliance with this requirement must be established experimentally on the completed mechanism.

Table 7: Calculated motion requirements for a 10 cm movement

Puck speed	Flight time	Motion time	Average speed	Peak speed	Acceleration
2 m/s	325 ms	297 ms	0.34 m/s	0.67 m/s	4.5 m/s ²
3 m/s	217 ms	189 ms	0.53 m/s	1.06 m/s	11.2 m/s ²
4 m/s	163 ms	135 ms	0.74 m/s	1.49 m/s	22.1 m/s ²

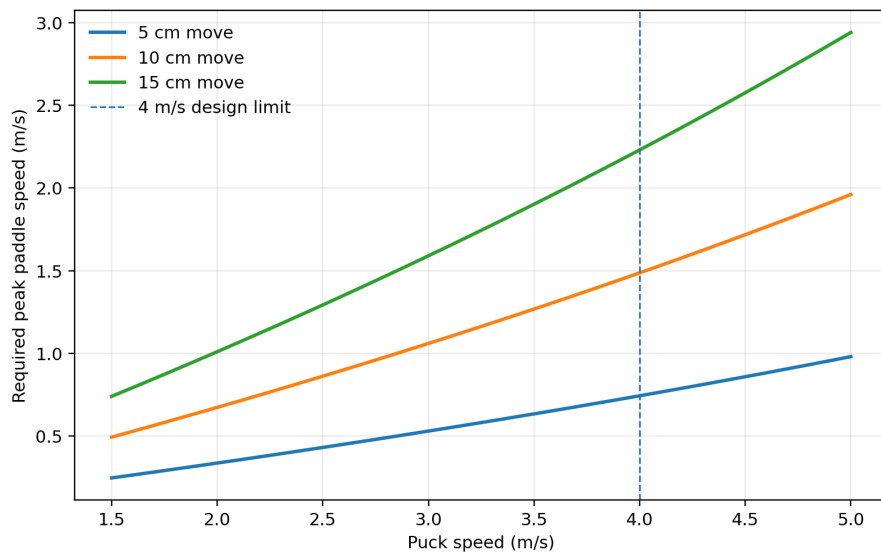


Figure 9: Required peak paddle speed for 5, 10, and 15 cm movements.

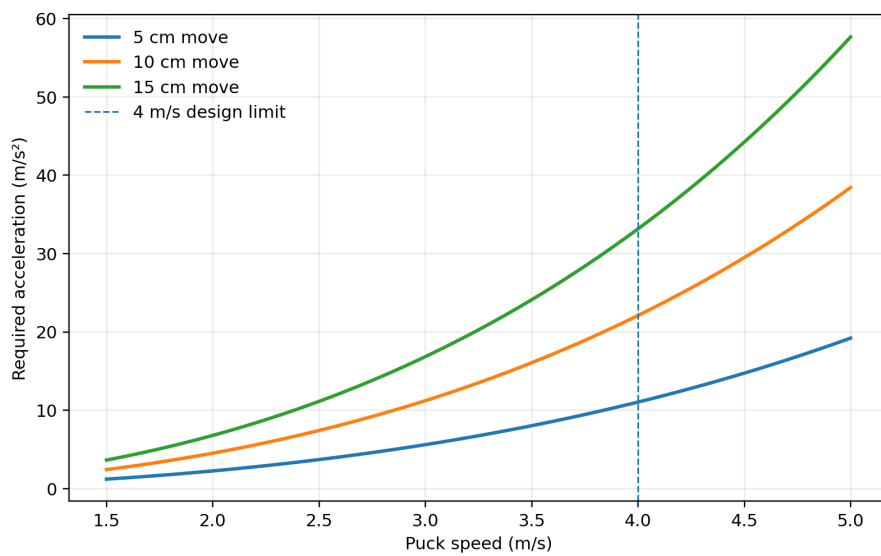


Figure 10: Required paddle acceleration for 5, 10, and 15 cm movements.

Keeping the planned paddle correction below approximately 10 cm substantially reduces the dynamic requirements. At 4 m/s, a 5 cm movement requires approximately 0.74 m/s peak velocity and 11.1 m/s² acceleration. A 10 cm movement requires approximately 1.49 m/s and 22.1 m/s², while a 15 cm movement requires approximately 2.23 m/s and 33.2 m/s². The 10 cm maximum movement is therefore treated as a motion-planning constraint rather than a physical motor specification. See Figures 9 and 10.

2.4.3 Processing Time Tolerance

The mechanical requirement is also sensitive to variation in camera and computation latency. For a 4 m/s puck and a 10 cm paddle movement,

$$a_{\text{req}} = \frac{4d}{(t_{\text{flight}} - t_{\text{processing}})^2} \quad (15)$$

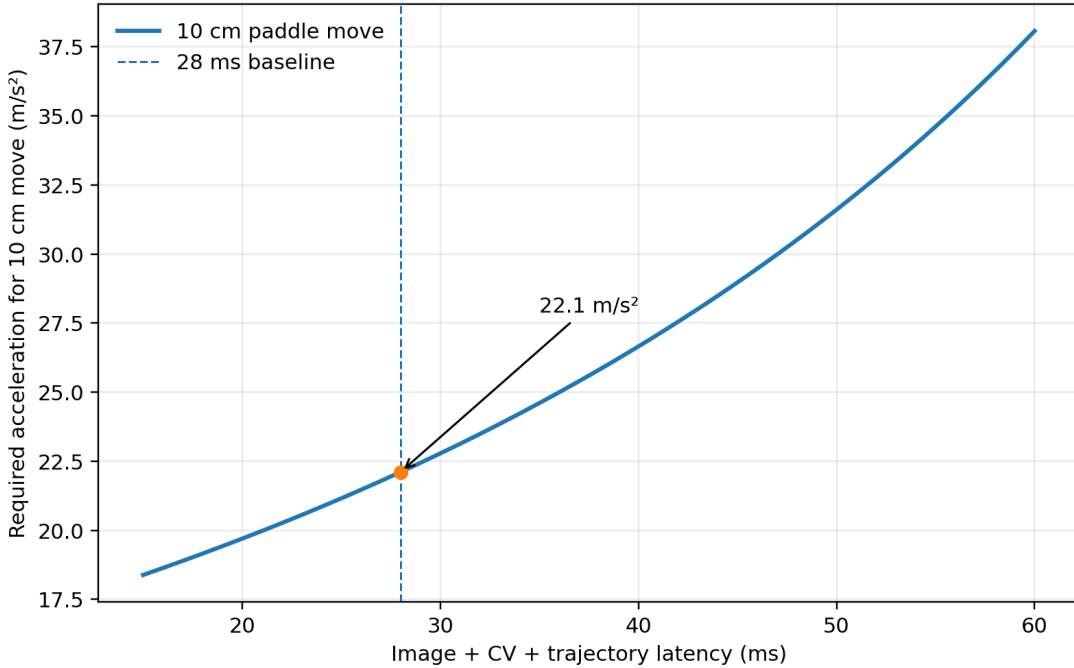


Figure 11: Required acceleration versus combined image and computation latency.

At the nominal 28 ms combined sensing and processing budget, the required acceleration is approximately 22.1 m/s². If this latency increases to 40 ms, the required acceleration rises to approximately 26.7 m/s². At 50 ms, approximately 31.6 m/s² is required. Figure 11 shows this sensitivity.

This illustrates that processing latency is itself an important system tolerance. Because required acceleration is inversely proportional to the square of the remaining movement time, relatively small increases in computation latency cause increasingly large mechanical requirements near the 4 m/s operating limit. The current 18 ms image-capture and 10 ms computation values should therefore be considered provisional until measured on the completed system.

2.4.4 Analysis and Design Implications

The analysis indicates a plausible timing budget for a puck traveling at the project's 4 m/s design limit. Camera Module 3 is specified for a 720p120 mode [2], while the TMC5160A supports a 24 V supply and STEP/DIR and SPI interfaces [4]. These specifications do not guarantee that the assembled system will meet the interception requirement.

At the worst-case operating point analyzed here, the complete image-processing pipeline must remain near its estimated 28 ms budget, and the H-bot must complete a 10 cm stop-to-stop movement in approximately 135 ms. This corresponds to approximately 1.49 m/s peak paddle velocity and 22.1 m/s² acceleration.

The largest remaining uncertainty is the loaded dynamic performance of the physical H-bot. The selected stepper is rated for 55 N·cm holding torque and 2 A per phase [3], but holding torque alone does not determine high-speed linear performance. The actual system will also depend on the motor torque-speed characteristic, paddle and gantry mass, pulley dimensions, belt compliance, friction, and H-bot racking. These effects should therefore be incorporated into the final tolerance analysis using measured maximum acceleration and movement time.

The analysis also supports limiting paddle travel through intercept-point selection. Reducing a worst-case movement from 10 cm to 5 cm approximately halves the required peak speed and acceleration. Therefore, minimizing unnecessary lateral movement provides significant timing margin without requiring changes to the sensing or motor electronics.

This analysis covers the timing-feasibility portion of the project requirement. A second tolerance analysis should characterize camera calibration and puck-position uncertainty, particularly how image-space position and velocity errors propagate into uncertainty in the predicted horizontal interception location. This is expected to be especially important for satisfying the one-bounce interception requirement.

3 Cost and Schedule

3.1 Cost Analysis

3.1.1 Labor

$(\$45.00/\text{ hour}) \times (2.5) \times (80 \text{ hours}) = \$9,000.00 \text{ per partner}$

$(\$9,000.00 \text{ per partner}) \times (3 \text{ partners}) = \$27,000.00$

Machine Shop Hours: $(\$30.00/\text{ hour}) \times (20 \text{ hours}) = \600

3.1.2 Parts

Table 8 summarizes the estimated parts cost.

Table 8: Estimated bill of materials

Description	Manufacturer	Part #	Qty.	Line cost (USD)
Air Hockey Table	MD Sports	—	1	\$34.84
NEMA 17 Stepper Motor	STEPPERONLINE	17HE19-2004S	2	\$12.72
Stepper Motor Driver	Analog Devices	TMC5160A	2	\$21.14
STM32 Microcontroller	STMicroelectronics	NUCLEO-G071RB	1	\$20.21
Raspberry Pi	Raspberry Pi	Pi 4	1	\$67.50
Raspberry Pi Camera	Raspberry Pi	Camera Module v2.1	1	\$25.00
IR Break-Beam Sensor	Adafruit	Product 2168	2	\$11.90
Mechanical Homing Switch	C&K	SDS005	2	\$3.10
24 V Bench Power Supply	Mean Well	LRS-100-24	1	\$15.20
5 V Buck Converter	RECOM	R-78HB5.0-0.5	1	\$19.26
3.3 V Regulator	Diodes Incorporated	AP2112K-3.3TRG1	1	\$0.28
8 MHz Crystal	Abracon	ABM8-8.000MHZ-B2-T	1	\$0.15
MOSFET	Alpha & Omega Semiconductor	AOD4185	1	\$1.40
TVS Protection Array	Littelfuse	SP1003-01WTG	1	\$0.20
Zener Diode	Diodes Incorporated	BZT52C12	1	\$0.23
TVS Diode	Littelfuse	SMBJ26A	1	\$0.50
Fuse	Littelfuse	1206L500	1	\$2.63
Pushbutton Switch	C&K	KMR221GLFS	1	\$0.20
USB-C Receptacle	GCT	USB4085-GF-A	1	\$0.91
2-pin Screw Terminal	Phoenix Contact	1725656	1	\$1.44
2-pin JST-XH Connector	JST	B2B-XH-A	1	\$0.20
3-pin JST-XH Connector	JST	B3B-XH-A	2	\$0.30
4-pin JST-XH Motor Connector	JST	B4B-XH-A	2	\$0.34
3-pin Pin Header	Samtec	TSW-103-07-G-S	1	\$0.27
10-pin Pin Header	Samtec	TSW-105-07-G-D	2	\$0.80
LEDs	Lite-On	LTST-C190KGKT	5	\$0.75
Test Points	Keystone Electronics	5015	9	\$0.90

Continued on next page

Table 8 continued

Description	Manufacturer	Part #	Qty.	Line cost (USD)
Resistor, 100 Ω	Yageo	RC0603FR-07100RL	4	\$0.40
Resistor, 100 Ω	Yageo	RC0603FR-07330RL	3	\$0.30
Resistor, 1 k Ω	Yageo	RC0603FR-071KL	2	\$0.20
Resistor, 1.5 k Ω	Yageo	RC0603FR-071K5L	1	\$0.10
Resistor, 3.3 k Ω	Yageo	RC0603FR-073K3L	2	\$0.20
Resistor, 5.11 k Ω	Yageo	RC0603FR-075K11L	2	\$0.30
Resistor, 10 k Ω	Yageo	RC0603FR-0710KL	6	\$0.60
Resistor, 15 k Ω , 1%	Yageo	RC0603FR-0715KL	2	\$0.20
Resistor, 33 k Ω	Yageo	RC0603FR-0733KL	1	\$0.10
Resistor, 47 k Ω	Yageo	RC0603FR-0733KL	1	\$0.10
Resistor, 82 k Ω	Yageo	RC0603FR-0782KL	1	\$0.10
Capacitor, 10 pF	Samsung	CL10C100JB8NNNC	2	\$0.20
Capacitor, 22 nF, 50 V	Samsung	CL10B223KB8NNNC	2	\$0.20
Capacitor, 100 nF	Yageo	CC0603KRX7R9BB104	15	\$1.35
Capacitor, 1 μ F	Samsung	CL10A105KB8NNNC	5	\$0.70
Capacitor, 2.2 μ F, 10 V	Samsung	CL10A225KQ8NNNC	2	\$0.56
Capacitor, 4.7 μ F	Samsung	CL10A475KP8NNNC	1	\$0.20
Capacitor, 22 μ F	Samsung	CL21A226MAQNNNE	1	\$0.45
Capacitor, 470 μ F, 50 V	Panasonic	EEU-FR1H471	1	\$1.00
Capacitor, 10 μ F, 25 V	Samsung	CL32A106KAJNNNE	4	\$1.56
Total				\$251.19

Total Parts Cost = $231.15 + 11.22 + 8.82 = \251.19

Account for shipping: $251.19 \times 1.05 = \$263.75$

Account for sales tax: $263.75 \times 1.1 = \$290.12$

3.1.3 Total Cost

$27,000.00 + 600 + 290.12 = \$27,890.12$

Once accounting for part costs (including shipping and sales tax), machine shop costs and labor costs, we estimate the total cost to be \$27,890.12.

3.2 Schedule

Table 9 lists the project schedule for Fall 2026.

Table 9: Project schedule and responsibilities

Week of	Task	Responsible
Oct. 4	Bring air hockey table and motors to machine shop	Everyone
	Set up Raspberry Pi and Camera Module	Rohan & Matt
	Begin computer vision and puck-detection	Rohan & Matt
	Review design with TAs/instructors	Everyone
Oct. 11	Continue puck detection	Rohan
	Implement puck velocity vector calculations	Micah
	Implement defensive paddle trajectory calculations	Matt
	Breadboard demo	Everyone
Oct. 18	Receive completed table and assembled H-bot mechanism	Everyone
	Calibrate camera to table coordinates	Rohan
	Verify H-bot mechanical alignment	Rohan
	STM32 setup and communication	Micah
	Test puck velocity and intercept calculations	Matt
	Assemble PCB	Everyone
Oct. 25	Test computer vision on air hockey table	Rohan
	Implement STM32 intercept calculations	Micah
	Begin PCB testing	Everyone
	Test motor functionality	Matt
	Test H-bot mechanism movement	Matt
	PCB edits	Micah
Nov. 1	Integrate motors with STM32	Micah
	Integrate camera vision with physical table	Rohan
	Test PCB power and motor-driver functionality	Matt
	Integrate Raspberry Pi to STM32 communication	Micah & Matt
	Integrate and test homing switches and IR break-beam sensors	Rohan
Nov. 8	Tune H-bot movement and positioning	Matt
	Test H-bot homing repeatability	Micah
	Measure puck detection and velocity accuracy using IR sensors	Rohan
	Test mechanism response to slowly moving pucks	Everyone
	Test straight-shot defense accuracy (goal: 75% at 4 m/s)	Everyone
	Test single-bounce defense accuracy (goal: 50% at 4 m/s)	Everyone
Nov. 15	Adjust puck trajectory calculations as necessary	Rohan
	Adjust paddle movement calculations as necessary	Matt & Micah
	Retest straight-shot defense accuracy (goal: 75% at 4 m/s)	Everyone
	Retest single-bounce defense accuracy (goal: 50% at 4 m/s)	Everyone
	Full working system test	Everyone
Nov. 22	Fall break	N/A
Nov. 29	Final system testing and fix existing bugs	Everyone
	Final documentation and demo preparation	Everyone
Dec. 6	Demo	Everyone

4 Ethics and Safety

4.1 Ethics Safety and Societal Impact

4.1.1 Human Safety Issues

Due to our project consisting of pucks and paddles moving at high velocities, human safety must be taken into consideration. We must prepare for the possibility of the puck being sent back at high velocity to an individual. The IEEE Code of Ethics, Section I.1, prioritizes public safety and welfare [5]. To ensure we do this, we will test the paddle so that it is not accelerated at speeds that could cause injury to an individual if a puck were to hit them. It will also be tested to ensure that, when returning the puck, the puck does not fly off the board when hitting a wall. Ensuring that neither of these scenarios becomes a reality will greatly improve human safety for observers and players.

Another safety concern is our H-bot mechanism. This mechanism can injure individuals, as it accelerates the paddle in a few different ways. It could hit an individual. When moving the paddle to defend the puck, the mechanism could accelerate into the hand of an individual. This poses a risk of injury. To combat this, the range of motion of the machine will be limited, and players will be required to stay on the side of the table that does not consist of the mechanism. This mechanism could also injure someone by moving when someone is adjusting the motor or one of the rails. To avoid this issue, as stated above, players and observers will not be allowed near the side of the table that contains this mechanism, and engineers will not be near it when it is active.

The final safety concern is the launcher. This device is intended to launch a puck at varying velocities. If a puck were to be launched at an individual, this could cause harm. To avoid this scenario, anyone using the launcher will be trained on how to use it and will be required to check the environment to ensure there is no one in the direction of the launcher when it is loaded (even if it is not currently being launched). To ensure total safety, the launcher will only be loaded right before it is used. Pucks will not be sitting in the launcher for extended periods of time.

4.1.2 Human Privacy

Because our system contains a camera, human privacy becomes a concern. Although the camera is pointed at the air hockey table, it is continuously recording video at 120 fps. This can cause individuals to be accidentally captured on camera. The IEEE Code of Ethics, Section I.1, addresses protection of privacy [5]. To ensure that we do this, two things will be done. Firstly, all individuals who participate will be made aware of the camera. Both the intended purpose, recording the table, and the unintentional side effect, that they could end up on camera, will be stated. Also, all footage will be deleted once it is no longer needed for processing intercept points. This ensures that there is no saved footage of any individuals.

4.1.3 Laboratory Safety Compliance

Due to the fact that this project will be developed in laboratories at the University of Illinois Urbana-Champaign, we will adhere to all campus safety and laboratory regulations. This includes following all safety protocols when handling our PCB, Raspberry Pi products, launcher, air hockey table, H-bot mechanism, and any other supplies that end up being used.

4.1.4 Societal Impacts

This project has a couple of positive societal impacts. The first of which is entertainment. Because the game traditionally requires two players, when an individual cannot find someone to play with, they become unable to play. This project solves this issue because it allows air hockey to be played alone. It takes a fun game and makes it more accessible to users, as there are now fewer requirements to play. Another positive impact is that this concept is applicable in other spaces. The concept of tracking movement to predict future movement and respond accordingly has many uses. When applied to vehicles, we could track pedestrians and other vehicles ahead of time to avoid collisions. When applied to sports, we can predict ball paths and intercept those paths, allowing for training automation in sports like soccer and lacrosse (where there could be a robotic goalie blocking shots). These concepts have many applications that can improve safety and efficiency in society.

References

- [1] P. Dürre *et al.*, “Outplaying elite table tennis players with an autonomous robot,” *Nature*, vol. 652, pp. 886–891, Apr. 2026. [Online]. Available: <https://www.nature.com/articles/s41586-026-10338-5>
- [2] Raspberry Pi Ltd., “Raspberry Pi Camera Module 3,” Product specifications, accessed: Oct. 2, 2026. [Online]. Available: <https://www.raspberrypi.com/products/camera-module-3/>
- [3] STEPPERONLINE, “E Series NEMA 17 Bipolar 55 Ncm 2 A Stepper Motor, 17HE19-2004S,” Product specifications, accessed: Oct. 2, 2026. [Online]. Available: <https://www.omc-stepperonline.com/e-series-nema-17-bipolar-55ncm-77-88oz-in-2a-42x48mm-4-wires-w-1m-cable-connector-17he19-2004s>
- [4] Analog Devices, “TMC5160/TMC5160A Stepper Motor Controller and Driver,” Product specifications and datasheet, accessed: Oct. 2, 2026. [Online]. Available: <https://www.analog.com/en/products/tmc5160.html>
- [5] IEEE, “IEEE Code of Ethics,” IEEE Circuits and Systems Society, accessed: Oct. 2, 2026. [Online]. Available: <https://ieee-cas.org/about/ieee-code-ethics>