Autonomous Robot Air Hockey

Merie, Ibrahim

April 28, 2019

***Abstract*—The objective of this lab was to construct and develop a dual-axis motorized robot arm capable of playing air hockey autonomously via closed loop control. To accomplish this, the kinematics of the arm linkage were found by calculating the end effector, or paddle, position in cartesian coordinates based on arm angles. The inverse kinematics of the system were found by calculating paddle position, velocity, and acceleration as a function of arm angles, velocities, and accelerations. The coordinates of the paddle were mapped into their corresponding location in table coordinates by applying an affine transform. The puck was tracked using an overhead camera operating at 522 frames per second. The location of the puck was mapped into its corresponding table coordinates using a bilinear transform. A gameplay strategy was created as a function of puck location and velocity. The robot was tested and yielded a 10% error in position tracking, a 70% defensive efficiency rating, and a 40% offensive efficiency rating.**

***Index Terms*— Error, Feedforward Control, Image Tracking, PD Control**

I. Introduction

The objective of this lab was to construct and develop a dual-axis motorized robot arm capable of playing air hockey autonomously via closed loop control [1]. LabVIEW software was used to create an arm command response as a function of paddle and puck location.

The kinematics of the arm linkage were derived by determining paddle position in the robot's native coordinate system as a function of arm angles. The inverse kinematics of the system were derived by calculating paddle position, velocity, and acceleration as a function of arm angles, velocities, and accelerations.

The coordinates of the paddle were mapped into their corresponding location in table cartesian coordinates by applying an Affine transform from three sampled points at known locations on the table. A bright blue LED was secured to the center of the puck while an overhead camera operating at 522 frames per second tracked its location. The location of the puck was mapped into its corresponding table coordinates using a bilinear transform using four sampled points at known locations on the table. A gameplay strategy was created as a function of puck location and velocity. The robot was tested against a human opponent its performance in offense, defense, and puck tracking was evaluated.

II. Procedure

Hardware

The same dual-axis motorized robot arm described in previous documents was constructed (Fig. 1) [2].

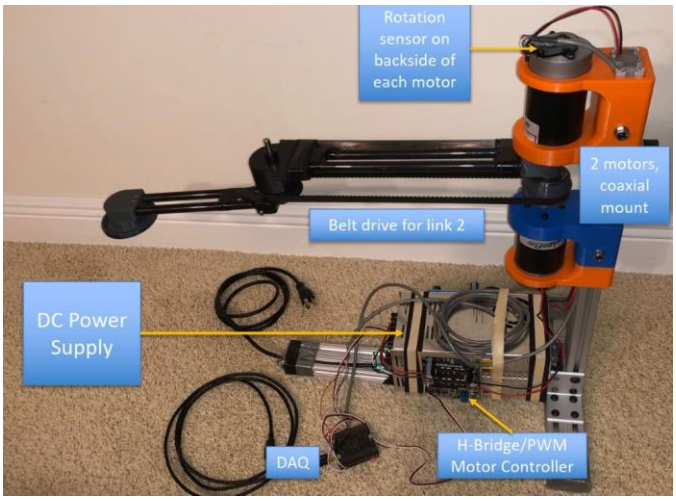

Fig. 1. The robot arm is displayed, and its components are labeled. The power supply used was a 24 V DC battery.

The rotation sensors used to track arm positions were optical encoders. A data acquisition device (DAQ) read position data from the encoders and relayed it into software for analysis. The air hockey table was marked with fiducial points, which were used to calibrate the robot arm and the camera locating the hockey puck (Fig. 2).

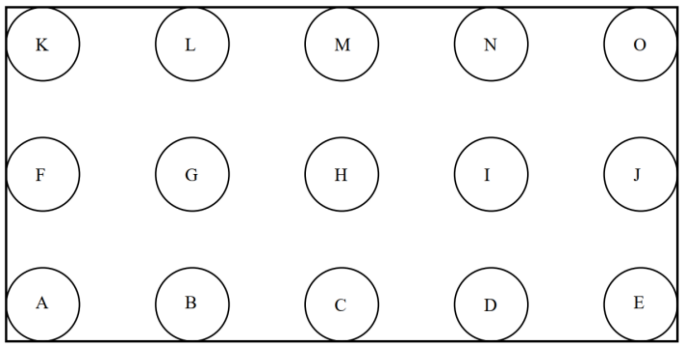

Fig. 2. The air hockey table had 15 fiducial markings, at known locations, used to calibrate the autonomous robot.

PID Control

A. Proportional Controller

A PID controller was created within the VI. The link was commanded to oscillate between a position of -45° and +45° at a frequency of 0.25 Hz. The integral gain (K_I) and derivative gain (K_D) were both set to zero. The controller was tested at proportional gains (K_P) of 0.00075, 0.00100, and 0.00125 for at least four 90° square wave cycles.

B. Observational Test

The link was commanded to remain at a position of 0°. The user manually displaced the link from its desired position (by grabbing it) and observed the resistance of the system. Proportional control was first observed at K_P values of 0.00010, 0.00025, and 0.00050. Integral control was next observed at K_I values of 0.005, 0.006, and 0.007. Derivative control was lastly observed at K_D values of 0.0001, 0.0005, and 0.001. Observations were recorded in a notebook.

C. Manual Tuning

All gains were initialized to zero. K_P was then increased until the system reached a stable speed. A K_P of 0.002 was selected. Next, K_D was increased until the system reached a desired dampening. A K_P of 0.0003 was selected. K_I was lastly increased until the overshoot of the system became less than 10%. A K_I of 0.002 was selected.

The link was commanded to oscillate between a position of -45° and +45° at a frequency of 0.25 Hz for at least four full cycles.

D. Full State Feedback

The velocity of the link was recorded using a constant command output of 10%. A normalized step response was selected and plotted (Fig. 1).

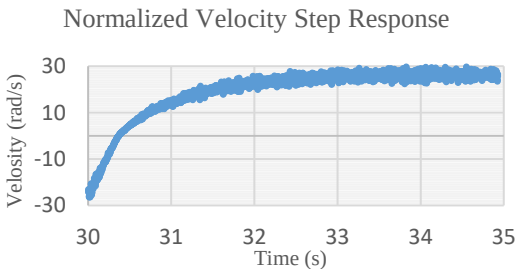

Fig. 1. One normalized step response cycle.

From the graph, a time constant (τ) was found from the point at which the step response reached 63.2% of its final value. Furthermore, a normalized step amplitude (A_n) was found by dividing the change in velocity by the change in voltage of the cycle.

The system dynamics were derived, and the open loop transfer function was found (1)(2).

$$J'\omega = V - b'\theta' \quad (1)$$

$$H(s) = (1/b')/[(J'/b')s + 1] \quad (2)$$

The experimentally determined time constant and normalized amplitude were matched to the poles of the system (3)(4).

$$b' = 1/A_n \quad (3)$$

$$J' = \tau * b' \quad (4)$$

These parameters were recorded and used to create a state space model of the system (Table I).

TABLE I
SYSTEM INPUT PARAMETERS

Symbol	Quantity	Value
J'	Effective Inertia	0.0533
b'	Effective Friction	0.0888
V	Voltage	24 V
Θ'	Rotational Velocity	27 rad/s
A_n	Normalized Amplitude	11.36 rad/s
τ	Time Constant	0.06 s
ζ	Dampening Factor	0.69

Using MATLAB and Ackerman's formula, the PID controller gains were designed to achieve a 2% settling time of less than 0.3 s and an overshoot of less than 5%.

The resulting full state feedback PID control scheme was created (Table II). K_I was set to zero to reduce complexity.

TABLE II
DESCRIPTION OF OPTIMIZED SYSTEM

Symbol	Quantity	Value
K_P	Proportional Gain	0.00140
K_I	Integral Gain	0.0
K_D	Derivative Gain	0.000968
σ	Pole Location (Real)	-13.33 rad
ω	Pole Location (Imaginary)	±13.98i

The control scheme was tested using a step response for a 0.25 Hz, 45° square wave for at least four full cycles.

II. Results

Data Analysis

The results of the tests were analyzed and tabulated in the same respective order as they were taken (Table III).

TABLE III
CONTROL SCHEME ANALYSIS

Type	Mean Error (°)	Error Standard Deviation (°)	Mean Command	Command Standard Deviation	RMS Error	RMS Command
Bang-Bang	-2.37	35.1	-0.00157	0.0500	83.5	7.88E-05
	-3.31	45.0	-0.00289	0.100	150	2.89E-04
	2.55	45.1	-0.000381	0.150	115	5.72E-05
Proportional	-2.44	46.4	-0.00183	0.0348	113	6.36E-05
	-4.20	40.20	-0.00421	0.0402	169	1.69E-04
	-9.13	37.4	-0.0114	0.0467	341	5.33E-04
PID Manual	-0.505	21.1	-0.00122	0.0813	10.7	9.98E-05
PID Full State	2.68	16.7	0.0376	0.743	44.9	0.0280

Root Mean Square (RMS) values were calculated by taking the square root of the sum of the squares of the mean and standard deviation of each data set.

Bang-Bang Control

The bang-bang controller resulted in a marginally stable system. The system did not achieve a steady state value. The arm oscillated around the given desired position in each cycle. The rise time of the system was 0.6 s. The performance of the bang-bang controller was poor and resulted in a percent overshoot of over 100% at effort levels of just 15%.

PID Control

A. Proportional Controller

The proportional controller resulted in a stable system. The system achieved a steady state value. Overshoot was present indicating an underdamped system. Percent overshoot was less than bang-bang control but still high at 90% at the smallest proportional gain. This system had a steady state error of under 10% for all gains. It was evident that increasing proportional gain decreased settling time but increased overshoot. This can be seen by increasingly larger RMS errors with higher gains (Table III).

B. Observational Test

A K_P larger than 0.002 resulted in a quicker response. A K_D larger than 0.0003 resulted in an overdamped response. A K_I larger than 0.002 resulted in systems with unstable responses.

C. Manual Tuning

The manually tuned PID controller resulted in a system that was stable. A steady state value was achieved. The steady state error was less than 5% with a rise time of 0.3 s. The system was underdamped. The position of the arm for a full cycle was plotted (Fig. 2).

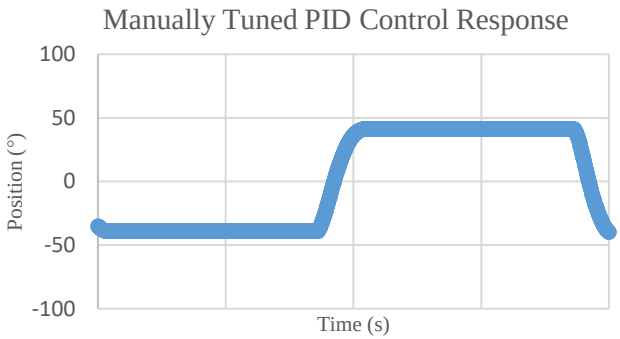

Fig. 2. One cycle response of manually tuned PID control. The cycle is 4 s.

D. Full State Feedback

The full state feedback PID controller resulted in a system that was stable. A steady state value was achieved. The steady state error was less than 3% in most cases with a rise time 0.2 s. The position of the arm for a full cycle was plotted (Fig. 3).

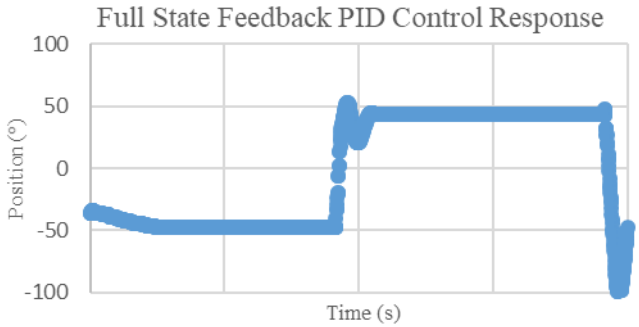

Fig. 3. One cycle response of full state feedback PID control. The cyle is 4 s.

III. Discussion

Data Analysis

The mean of each data set indicated a bias in the data, while the standard deviation gave a metric by which the precision of the arm could be assessed (Table III). Error was used as a metric of how well the given task was performed by the robot arm. A small error was desirable because it indicated a smaller overshoot in the system. The motor command signal was a metric of how much effort was expended for the task. A lower motor command was desirable because it offered increased efficiency. The ideal step transition is plotted (Fig. 4).

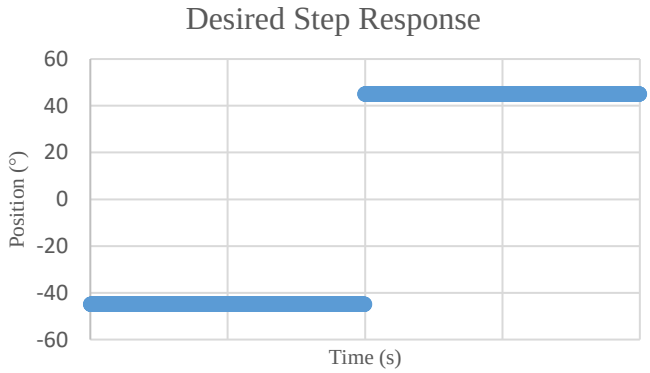

Fig. 4. Optimal step response of system. The cycle is 4 s.

Bang-Bang Control

The bang-bang controller was always delivering a nonzero output command and thus never stabilized. This is evident by the fact that the standard deviation of the command was equal to the given effort level (Table III). The command level from this controller was constant unlike the proportional controller that delivered a variable command. Bang-bang control was therefore less responsive.

PID Control

A. Proportional Controller

The proportional controller was more responsive than the bang-bang controller because it generated a variable command output as opposed to a constant one. The proportional controller also offered a faster rise time. A K_P larger than 0.003 resulted in unstable systems.

B. Observational Test

The output of the proportional controller was directly related to the displacement of the arm. This effect is synonymous with a mechanical spring; as gain values increased by a given factor, the level of command increased by the same factor.

On the other hand, the output of the derivative controller was proportional to the rate of error with respect to position. A higher velocity of the arm resulted in greater rates of error and thus larger output commands were generated. The intensity of such a dampening effect increased with larger gain values. Increasing K_I increased settling time but decreased overshoot.

Lastly, the output of the integral controller was dependent on the length of time at which the arm remained at a position of a non-zero error. The longer the arm was displaced, the larger the output command generated.

C. Manual Tuning

The manually tuned PID controller generated a smooth position response with no overshoot (Fig. 2). The rise time was 0.1 s slower than the full state control system and the steady state error was greater by 2% for most cycles.

The manually tuned PID control system was, however, the most power efficient because it had the lowest RMS command value (Table III). This system was also the most accurate because it had the lowest mean error (Table III).

D. Full State Feedback

The full state feedback PID controller generated a discontinuous position response with large overshoot (Fig. 3). This system offered a reasonable level of accuracy as seen by a mean error of only 2.68° (Table III). However, the system was imprecise as seen by its abnormally large command signal standard deviation (Table III). The variation in command is the reason that the position curve was not smooth. This may have been the result of a noisy derivative control gain exacerbated by the absence of an integral control gain. Although the full state controller was designed to generate a percent overshoot of under 5% this was not the case in practice. The system offered inconsistent results on each cycle. A possible explanation is that the system needed integral control to improve its consistency and decrease its steady state error by decreasing the noise generated by the derivative control factor.

IV. Conclusion

The optimal control scheme as evidenced by the results of this lab was a manually tuned PID controller with K_P, K_I, and K_D values of 0.002, 0.002, and 0.003, respectively. This system offered a quick rise time of 0.3 s, a low steady state error of 3%, and consistent results. Although the full state feedback PID controller was theoretically superior, it did not produce superior results. The full state controller did yield faster rise times by 0.1 s, but it was but it created an underdamped system with an overshoot of more than 100% on most cycles.

The performance of the full state model could be improved. Viscous friction within the system is the main reason why the full state model did not perform as well as intended. The control scheme could be improved by building a model that more accurately predicts the dynamics of the system by considering factors such as viscous friction and variable belt tension. An integral control factor would also potentially improve the system.

The performance of the bang-bang controller could be also be improved with the use of a dead zone centered around a zero error. This would reduce the oscillation in the system, increase its stability, and offer improved power efficiency.

Furthermore, this lab was conducted with hardware that was used extensively and vigorously. The imperfect rigidity of the robot arm frame, the wear of components, and vibrations of the setup all contributed to the limitations of this lab. The results could have been better if the robot arm was assembled and maintained with better care.

References

[1] Banks, Scott, “EML 4314C - Spring 2019 - Lab 1 Assignment” University of Florida, 2019

Forward Kinematics

The robot arm was modeled in a two-dimensional coordinate axis (Fig. 1).

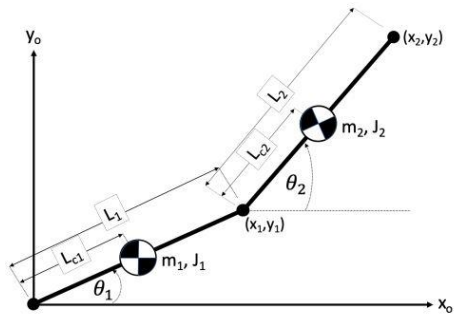

Fig. 1. A simplified model of the dual-axis robot arm [4].

The paddle position on the x-axis and y-axis were found with respect to arm geometry (1)(2). Subscripts indicate the link that each term describes as shown in the diagram.

$$x_2 = L_1 cos\theta_1 + L_2 cos\theta_2 \qquad (1)$$

$$y_2 = L_1 sin\theta_1 + L_2 sin\theta_2 \qquad (2)$$

Inverse Kinematics

The geometry of the arm was analyzed to quantify the angular positions of each link (Fig. 2).

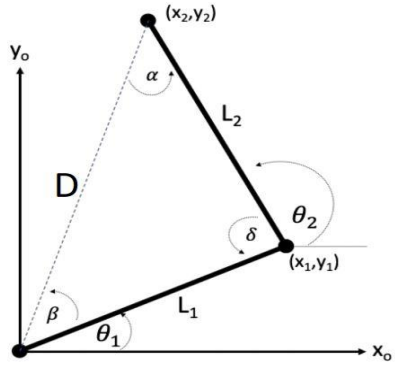

Fig. 2. A model of the robot arm depicting angular positions [4].

The distance from the origin and the paddle was derived using the Pythagorean theorem (3).

$$D = \sqrt{{x_2}^2 + {y_2}^2} \qquad (3)$$

The angular position of the paddle relative to the origin was calculated (4).

$$\beta = \cos^{-1}\left(\frac{D^2 + {L_1}^2 - {L_2}^2}{2DL_1}\right) \qquad (4)$$

The angular positions of the links relative to the origin were calculated (5)(6).

$$\theta_1 = \tan^{-1}\left(\frac{y_2}{x_2}\right) - \beta \qquad (5)$$

$$\theta_2 = \tan^{-1}\left(\frac{y_2 - L_1 sin\theta_1}{x_2 - L_1 cos\theta_1}\right) \qquad (6)$$

The relation between the velocity of the paddle is found as the product of the Jacobian and joint velocities (7).

$$\begin{bmatrix}\dot{x_2}\\ \dot{y_2}\end{bmatrix} = J(\theta)\begin{bmatrix}\dot{\theta_1}\\ \dot{\theta_2}\end{bmatrix} = \begin{bmatrix}-L_1 sin\theta_1 & -L_2 sin\theta_2\\ L_1 cos\theta_1 & L_2 cos\theta_2\end{bmatrix}\begin{bmatrix}\dot{\theta_1}\\ \dot{\theta_2}\end{bmatrix} \qquad (7)$$

Taking the time derivative of the previous equation and solving for joint velocity yields desired joint accelerations (8).

$$\begin{bmatrix}\ddot{\theta_{1d}}\\ \ddot{\theta_{2d}}\end{bmatrix} = J(\theta)^{-1}\left\{\begin{bmatrix}\ddot{x_{2d}}\\ \ddot{y_{2d}}\end{bmatrix} - \frac{dJ(\theta)}{dt}\begin{bmatrix}\dot{\theta_{1a}}\\ \dot{\theta_{2a}}\end{bmatrix}\right\} \qquad (8)$$

A subscript of ‘a’ indicates a measured value from the encoders. A subscript of ‘d’ indicates a desired value generated by the VI.

The voltages sent to each motor is found using these relations and inertial and centripetal terms [2].

$$V_1 = \frac{R_M}{K_T}\left[d_{11}\ddot{\theta_{2d}} + d_{12}\ddot{\theta_{2d}} + c_{221}\theta_{2a}\dot{} + \dot{\theta_{1a}}(b_1 + \frac{{K_T}^2}{R_M})\right] \qquad (9)$$

$$V_2 = \frac{R_M}{K_T}\left[d_{21}\ddot{\theta_{1d}} + d_{22}\ddot{\theta_{2d}} + c_{112}\theta_{1a}\dot{} + \dot{\theta_{2a}}(b_2 + \frac{{K_T}^2}{R_M})\right] \qquad (10)$$

Proportional Derivative Control

The PD controller was tested four times. The test was designed to evaluate the ability of the robot arm to follow a circular path of a known radius and known location. Four tests were conducted with varying circle sizes, locations, and frequencies [Table II].

TABLE I
ROBOT TRAJECTORY TEST

Run	X Coordinate (m)	Y Coordinate (m)	Radius of Circle (m)	Frequency of Trace (Hz)
1	0.24	0.24	0.01	0.5
2	0.24	0.24	0.10	0.5
3	0.24	0.24	0.10	1.5
4	0	0.439	0.10	0.5

Feedforward Control

The feedforward controller was tested four times using the same four trajectories as the PD controller test.

Combined Control

The combined controller was tested four times using the same four trajectories as the PD controller test.

RESULTS

The error in position and the command signals were analyzed [Table II].

TABLE II
CONTROL SCHEME ANALYSIS

Type	Mean Error (rad)	Error Standard Deviation (rad)	Mean Command (V)	Command Standard Deviation (V)	RMS Error (rad)	RMS Command (V)
PD	0.250	0.509	0.132	5.85	0.624	2.08
Feed-forward	0.019	0.475	0.969	13.4	0.709	5.85
Combined	0.424	0.311	-0.183	6.02	0.942	2.38

PD Control

Firstly, the VI calculated the error in position as a difference between the measured and actual arm positions. Using a math tool, the integral and derivative of this error signal was calculated. The IV used this information to implement a Proportional Derivative (PD) open loop control scheme. The gain values of this controller were initialized to zero and gradually increased until acceptable performance was achieved [2].

Feedforward Control

The second objective that the VI accomplished was implementing a feedforward closed loop control scheme. To do this, the kinematics of the arm were analyzed. The end effector, or paddle, position was found as a function of arm angles and lengths [3]. The kinematics of the system were found in the inverse direction. The rotational inertia and centripetal components of the arm were calculated. Using this data, the required torque, and therefore voltage, required by the system for a given response was found and a feedforward controller was created [4].

Affine Transformation

The third objective of the IV was to track the hockey puck and interrelate the coordinate systems of the robot to the table [5]. In other words, robot space was mapped to world (table) space using an Affine transform (1).

$$\begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} = TX_r = \begin{bmatrix} \cos\theta_Z & -\sin\theta_Z & x_o \\ \sin\theta_Z & \cos\theta_Z & y_o \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ 1 \end{bmatrix} \quad (1)$$

A subscript of "w" indicates a world coordinate. A subscript of "r" indicates a robot coordinate. The angular displacement from one coordinate system to the next is defined as θ_Z. The distance of offset from origin to origin in the x direction and y direction is defined as x_o and y_o, respectively. Three points at known locations on the table, or fiducial points, were used to generate the Affine transform matrix, T [6].

Bilinear Mapping

The location of the puck on the table was tracked using an overhead high frame rate camera and a bright blue LED secured to the center of the puck. The location of the puck was mapped into world space using a bilinear transform. Four fiducial points were used to generate the bilinear transform matrix, C (2).

$$[C] = [P]^{-1}[W] \quad (2)$$

The P matrix is composed of pixel locations of the LED while the W matrix is composed of table coordinates at four known locations.

The velocity of the puck was then calculated (3).

$$\begin{aligned} x_p &= x_i + v_x t \\ y_p &= y_i + v_y t \end{aligned} \quad (3)$$

The predicted location is a function of x and y velocities, (v_x, v_y), and initial x and y positions, (x_i, y_i).

Gameplay Strategy

The last objective of the lab was to implement an air hockey gameplay strategy. To ensure that the equipment was not damaged, limits were placed on the robot's movement. The limits ensured that the paddle remained over the playable hockey table surface and did not hit the walls hard while tracking the puck's movement. A desirable range was mapped out on the table and the LabVIEW VI was programmed to only have the robot operate in the desired area. The robot was programmed to play with both defensive and offensive strategies. The defensive strategy was to track the puck in the y-direction and match the pucks movement, while remaining at a constant position in the x-direction (Fig. 3).

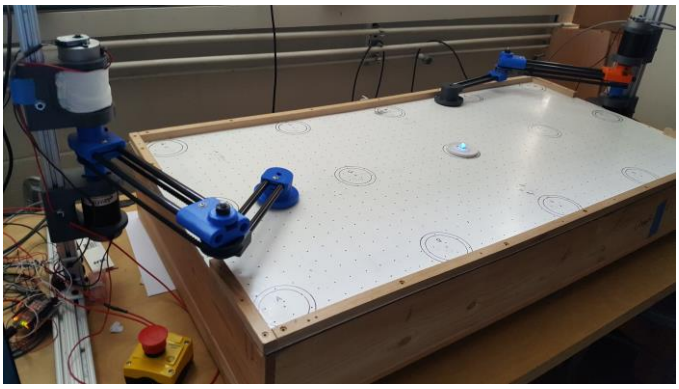

Fig. 3. The robot operating in defensive mode (left) is shown. In defensive mode, it mirrors the puck's position in the y-direction and maintains a constant x-direction position

The offensive strategy was similar to the defensive strategy in how the robot was programmed to mirror the pucks position in the y-direction. In offensive mode, though, the robot was programmed to hit the puck if it entered a certain range in the x-direction. To strike the puck, the robot paddle would move to a position past the known location of the puck to move through the puck and push it in the opposite direction.

Testing

To quantify the accuracy and performance of the autonomous robot, multiple tests were conducted. The control method selected was a combined controller (PD and Feedforward control). The proportional gain values were initialized to zero and manually increased until an acceptable level of performance was observed. The derivative gain was also initialized to zero and gradually increased until the system achieved an appropriate level of dampening. Control loop time was set to 3 ms.

To characterize the accuracy of the robot's end-effector positioning, the expected and actual position data was taken at four locations on the air hockey table. To characterize the camera-based puck positioning system, the expected and actual position data was taken at six locations on the air hockey table. The autonomous robot's air hockey gameplay performance was tested by pushing the puck toward the robot at a variety of potential angles and speeds. The puck was sent twice towards the robot's goal in five directions using an average human strike. The robot's defensive and offensive strategies were tested separately using this gameplay simulation technique.

RESULTS

The system was evaluated in its ability to accurately track the location of the puck (Table I).

TABLE I
PUCK LOCATION ACCURACY TEST

Fiducial Location (m)	Projected x-coordinate (m)	Projected y-coordinate (m)	Total Distance Error (m)
A (0,0)	0.000	-0.011	0.011
E (1.108,0)	1.115	0.023	0.007
F (0,0.248)	-0.005	0.234	0.014
G (0.278,0.248)	0.263	0.253	0.008
K (0,0.498)	-0.004	0.483	0.015
O (1.108,0.498)	1.122	0.472	0.002

The error in puck location was assessed as the distance from the desired point to the actual point. The average error in puck location was found to be 0.010 m.

The system was also evaluated in its ability to accurately place the paddle at a desired location (Table II).

TABLE II
ROBOT PADDLE LOCATION ACCURACY TEST

Fiducial Location (m)	Projected x-coordinate (m)	Projected y-coordinate (m)	Total Distance Error (m)
A (0,0)	0.001	0.001	0.001
F (0,0.248)	0.002	0.255	0.007
G (0.278,0.248)	0.276	0.247	0.002
K (0,0.498)	0.000	0.498	0.000

The error in paddle location was assessed as the distance from the desired point to the actual point. The average error in paddle placement was found to be 0.003 m.

The system was tested in its ability to play air hockey against a human opponent (Table III).

TABLE III
GAMEPLAY PERFORMANCE TEST

Gameplay Strategy	Total Attempted Shots	Successful Strikes	Unsuccessful Strikes	Success Rate
Defense	10	7	3	70%
Offense	10	4	6	40%

The robot was able to successfully deflect the shots of the opponent away from its goal seven times out of ten. The robot was able to strike and send the puck towards the opponent's goal four out of ten times.

DISCUSSION

The robot proved that it could successfully play autonomous air hockey. Although the system did not perfectly place the paddle in any desired location, the level of error in placement was found to be insignificant. During gameplay the 0.003 m error in placement did not inhibit the performance of the robot significantly because it was still able to defend well and strike the puck well.

The error in puck location tracking was, however, four times greater than the error in paddle placement. This error in tracking the position of the puck was the main contributor to the total error of the system at 0.010 m on average. This error in tacking can be attributed to lens distortion of the camera and to noise in the optics of the system due to varying lighting conditions in the lab.

Although the performance of the controller was acceptable, it can be significantly improved. To get the entire table in view of the camera, a focal length of 4 mm was used. Such a lens creates barrel distortion in the captured images (Fig. 4).

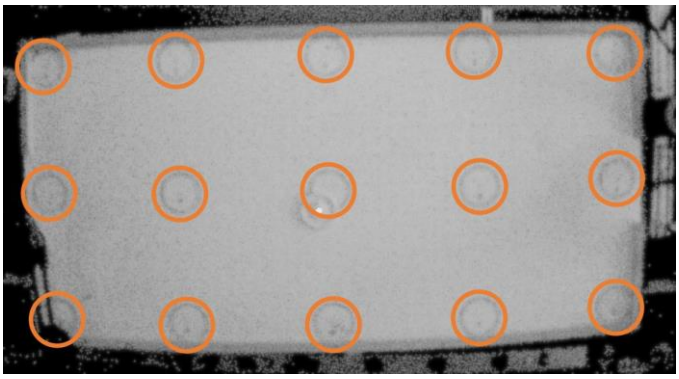

Fig. 4. This is the image that is taken by the overhead camera [7]. The rectangular table is clearly seen as distorted. This barrel distortion is the main contributor to error in the system.

The optimal method for improving tracking performance would be to offset the lens distortion effect by using a higher-order transform matrix thereby eliminating the largest source of error in the system.

REFERENCES

[1] Banks, Scott, "EML 4314C - Spring 2019 - Lab 4 Assignment" University of Florida, 2019
[2] Merie, Ibrahim, "EML4314C_IM_Lab1" University of Florida, 2019
[3] Merie, Ibrahim, "EML4314C_IM_Lab2" University of Florida, 2019
[4] Merie, Ibrahim, "EML4314C_DF_IM_Lab3" University of Florida, 2019
[5] Merie, Ibrahim, "EML4314C_IM_TD_Lab4" University of Florida, 2019
[6] Banks, Scott, "EML 4314C - Spring 2019 – Lecture 13" University of Florida, 2019
[7] Banks, Scott, "EML 4314C - Spring 2019 – Lecture 20" University of Florida, 2019