Proceedings of 2014 International Conference on
Modelling, Identification and Control, Melbourne, Australia, December 3-5, 2014

Design and Implementation of a Vision-Based Robotic Air Hockey Table

Christopher Marra, Damian Cattenazzi, Adnan Masinovic, and Jinchuan Zheng
School of Software and Electrical Engineering
Swinburne University of Technology, Hawthorn, VIC 3122, Australia
Email: christophermarra86@gmail.com

***Abstract*—Air hockey has been an arcade favourite since the early 70's and has gained as much popularity to even schedule a yearly world championship. Unfortunately, this game requires a minimum of two players making it impossible for users who wish to train or play by themselves. To solve this issue, a robotic air hockey table capable of emulating a human player has to be created. In this paper, we have developed a computer vision system to detect and track the movements of the puck which is used to predict the final position of the puck in real time. Once the system predicts the final location of the puck, an electrical linear actuator capable of high speed and acceleration is activated which drives the mallet to block the goal. This paper will also discuss various problems encountered and possible future solutions to eradicate them.**

I. Introduction

Air hockey has been a popular game since the early 70's. However, it requires a minimum of two players making it impossible for users who wish to train or play by themselves. In this paper, we discuss our implementation of an autonomous system capable of replacing a human opponent in the game of air hockey. To achieve this goal, a custom vision system and an electrical actuation system has been developed and integrated. Compared to a pneumatic actuator, the electrical actuator has the advantage of not only being more portable in size but also in the required infrastructure. This also provides the benefit of being able to replace components more easily and affordably. The implementation of a computer vision system is the best compromise between performance and affordability.

The main concerns in the development of our system stem from the fast motion of an air hockey puck which can travel across the entire table in approximately 0.3 seconds. The fast speed of the puck requires the vision systems calculations must be achieved in real time, and the linear actuator must be able to travel half the width of the table in at least 0.3 seconds. Other issues that may have degraded the system performance have arisen from the noise created by the environment. In particular, artificial lighting indoors and the rapid changes in ambient natural light. In order to resolve these issues, various image processing techniques and algorithms such as applying thresholds, morphologies and limits [1] have been utilised to filter the image from any noise.

The vision-based problems associated with our system were to track the puck and predict its final location. Related work has been reported in various literatures such as the estimation algorithm by using motion blur from two different cameras [2]. In this paper, we describe our employment of canny edge detection with the contour algorithm, which is simple to implement and requires one camera only. This approach has proven to be capable of maintaining the location of the puck at all times, as well as generating a prediction before the puck can move approximately a quarter of the length of the table.

Finally, we present the use of a proportional-integral-derivative (PID) controller [3] for the electrical linear actuator to achieve rapid movement of the mallet in real time. A dual PID control loop has been designed to achieve both fast response time and accurate movements for both small and large displacements. Experimental results are presented to verify its performance.

The implementation results shown by our system is capable of performing real time calculations to move the linear actuator to its required position accurately. The implementation of this system entails architectures required by industries such as high volume manufacturing lines. Our work was able to prove that it is possible to achieve these forms of rapid mechanical movements by utilising affordable options such as computer vision.

II. System Description

The robotic air hockey table we have built is shown in Fig. 1. It consists of a conventional air hockey table, a high speed camera, an electrical linear actuator to drive the mallet, and a standard personal computer for running the controller (not shown in the figure).

The high speed camera (Basler A602f Black and White) has been fixed with a fish-eye lens and has been mounted 1200mm above the table with two side support bars such that it can capture the image of the whole table. The camera communicates with a real time linux operating system which is running software based on OpenCV. A Linux system was chosen due to its performance benefits gained from the minimal amount of computer resources needed, its ability to offload tasks into memory and its fast interrupt switching abilities compared with other OSs. Other reasons include its simple program installation and upgrade capabilities as well as the extensive documentation describing the entire system. OpenCV was chosen as the base for the vision system as it provides easy to use and highly efficient algorithms via its

Fig. 1. Diagram of the robotic air hockey table system.

API. It also provides its entire source code as welll as its highly sufficient documentation [4].

The linear actuator (Macron Dynamics MSA-628) [5] was chosen due to its compromise of affordability, speed and accuracy. It is comprised of a linear rail driven by an integrated 36V NEMA23 motor which is capable of 3500rpm at load with a continuous torque of 3kg-cm. The mallet is attached to the linear rail, which is capable of 5m/s travel speed and acceleration up to 5G. Our calculations yield that the maximum linear speed achievable by the actuator is 4.37m/s. One of the most important features of the linear actuator is its accuracy and repeatability, which ensures accurate positioning relative to the input position reference from the vision system. The actuator can provide positional accuracy of ± 0.4mm and repeatability of ± 0.025mm. The aforementioned parts have been chosen specifically for their ample amount of speed and agility, which matches the high speed output of the vision system. The mechanical setup is completely modular which allows interchangeability of any part if upgrade or replacement is needed.

As will be shown later, the linear actuator is controlled by a dual PID feedback loop such that it can drive the mallet to quickly reach the predicted position by the vision system to block the puck.

III. Computer Vision

A. Overview

The vision system is an integral part of the robotic air hockey table. Its main goal is to interpret the image feedback from the high speed camera in order to detect the puck's position. Furthermore, it will generate a prediction of the puck position in real time, which is then used as the reference value for the linear actuator control system to track.

As proven by tests, the time that a high powered shot takes to travel from one end of the table to the other is approximately 300ms. If the camera is to operate at its maximum specification of 100 frames per second (fps), it will produce approximately 30 frames during that period. Since the linear actuator takes 250ms to move from the middle of the goal to the edge of the table, this date requires the vision system to generate a prediction within one quarter of the table or approximately 50ms to allow for the mallet to reach the puck. This translates to the requirement of a prediction within 5 frames. In order to achieve this goal, the program structure of vision system consists of the following subsystems:

1) Initialisation: This will initialise critical camera settings such as the fps, exposure, shutter, gamma, distortion values and kernel sizes. These are needed to create the most optimised image that can be properly filtered. It is very important that this will take place outside the image loop to minimize the processing involved.

2) Image filtering: Filtering requires the image to be remapped to remove fish eye distortion, crop the image to remove noise and speed up processing. Various filtering functions are used such as erosion, dilation, thresholding, blurring and binary conversion. The reason for these techniques is to remove all objects in the image except for the puck.

3) Detection: In order to detect the circular shape of the puck, the vision system utilises canny edge detection. Using the contour algorithm the system will give the position of the centre of mass. During this stage the system will also apply a set of limits to disregard objects that do not contain the pucks features.

4) Prediction: By using previous and current position values, we can predict the total movement of the puck by

$$T = \left|(C_y - P_y)\left|\frac{P_x - G_x}{C_x - P_x}\right| + P_y\right| \quad (1)$$

and the final puck position (Goal y) by

$$G_y = \left|T - \left(2(B_h - B_l)\lfloor\frac{T}{2(B_h - B_l)}\rceil\right)\right|, \quad (2)$$

Where

T = Total amount of movement
G_y = Goal y position
G_x = Goal x position
P_x = Previous frame's x position
P_y = Previous frame's y position
C_x = Current frame's x position
C_y = Current frame's y position
B_l = Lowest board value
B_h = Highest board value

5) Sending information: A serial value is sent to the linear actuator which represents the distance the linear actuator will need to move from the start position to the prediction. There will be a trigger point which may change according to the speed of the puck. This trigger point is used to determine when the hit back of the mallet should be triggered.

The aforementioned program structure is demonstrated by an UML activity diagram as shown in Fig. 2.

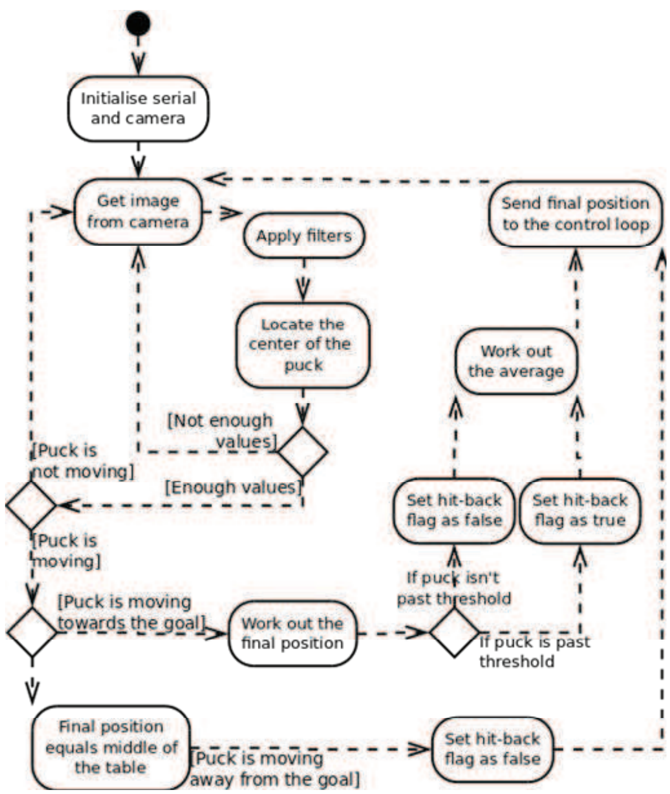

Fig. 2. UML activity diagram for program structure of the vision system.

B. Filtering

In our system, the captured images contain heavy distortions caused by the fish eye lens on the camera. In order to remove these distortions, our system utilises an undistortion method explained in the OpenCV documentation [6]. This method gathers an array of offset values which are created by capturing 30 images using a chessboard as a guide to determine which lines are straight. Using these offsets the image can then be remapped to fix the distortions. The result of using this technique is shown in Fig. 3. Once the image is undistorted, the unneeded areas are cropped out leaving only the table surface in the image.

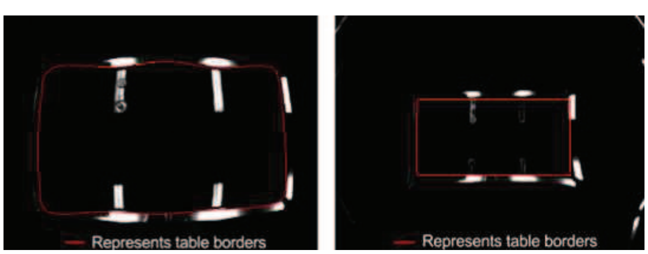

Fig. 3. Distortion processing of the original image (Left: distorted; Right: undistorted).

We are then required to remove the noise produced from the environment. The main impact of the environmental noise is due to the rapidly changing ambient light. Since the lighting conditions change at various times throughout the day, using a simple lower and upper threshold was not suffice and required constant manual adjustments throughout testing. Therefore, we used adaptive threshold values to provide a more robust performance. This new method will have each pixel check against its corresponding neighbours. If a large edge is detected, it will parse them to values between the table and the puck. We have also implemented automatic image adjustment, which will check the brightest and darkest areas of the table and apply different thresholds to each section. It will then readjust them to an average value which is gained from the entire table. Fig. 4 presents the result of thresholding.

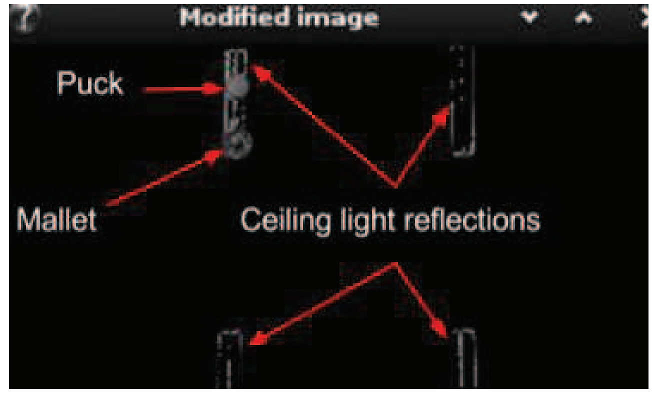

Fig. 4. Image after thresholding.

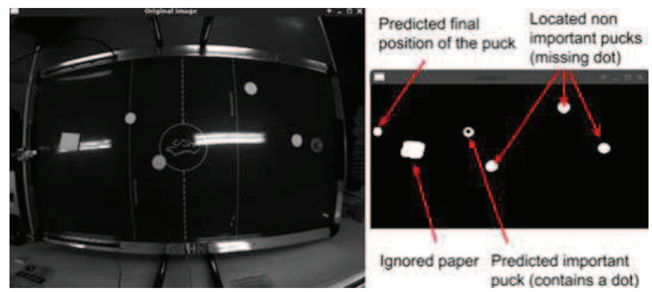

Fig. 5. Filtering result of the original image (Left: original; Right: final with filtering).

The majority of leftover noise is removed through the use of morphological techniques. The first step is the erosion of the image using a rectangular element. The reason for using a rectangular element is due to the pucks circular appearance resulting in it being the last object to be eroded. The next step is the dilation of the image by using an elliptical element. The morphological algorithm has been proven to be effective at removing the majority of noise and reflections on the table. The end result is shown in Fig. 5.

C. Feature Detection

Feature detection is used to detect the location of the puck. Initially the system utilised the Hough circle detection algorithm which worked quite well during the testing phase. However, as complexity increases some issues become prevalent. For example, the excessive amount of jitter caused by the low resolution of the camera, the inflexible parameters which are at its limitations again due to the low resolution of the camera. Its slow running speed and its huge memory usage compared to other algorithms.

Another available method is the Blob detection. This is proved to be the fastest and least memory intensive solution. Blob detection worked very well during initial tests, but as complexity increased there were two main issues. The first is the detection that will be affected by any other white blob that comes in view of the camera. The other was that it does not provide the capability to run multiple pucks at the same time.

The final solution to our system was the use of contour detection. By using this algorithm it is possible to provide identification to each blob, which enables the system to locate multiple pucks and provide useful limits. These can be limits such as eliminating large objects such as peoples' shirts or eliminating objects that are not round enough. Fig. 6 demonstrates the use of contour detection algorithm to identify the center of the puck.

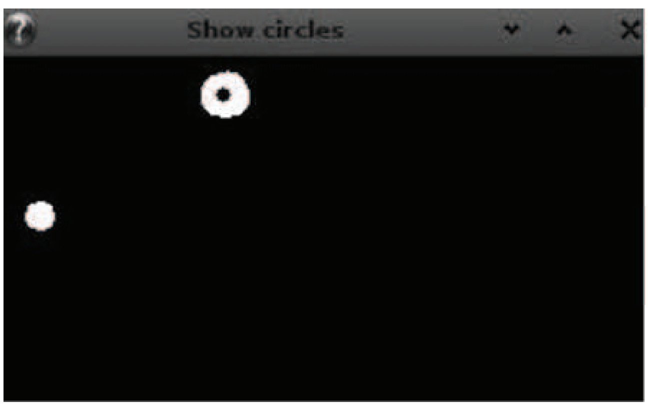

Fig. 6. Final image showing the centre of the circle via contour detection.

D. Prediction Algorithm

The prediction algorithm that was utilised was derived from first principle calculations. The basic structure of the algorithm works by determining the amount of steps till the puck reaches the goal position in the x plane. By using this value as a magnitude for the position in the y plane, the algorithm uses added features to determine locations in case of boundaries.

The advantages of using this algorithm is that it is incredibly fast because it uses no complex math functions or loops. There are steps in place to stop errors occurring with bounces and slow movement. Finally, if the puck gets lost or deviates for some unknown reasons, the code will adapt and create a new prediction.

Unfortunately, while this algorithm works in theory, it fails to take into account the limited resolution of the camera. Because of the small resolution and high frame rate of the camera, the algorithm will sometimes see the puck moving in a 45 degree angle instead of the correct angle. To fix this problem, we collect 4 points (also for satisfying the requirement of 5 frames) before predicting the resulting location. However, this will be slow when compared to many of the original principles. In order to improve the performance, an array is implemented which is updated at every frame. This means that not only will the algorithm gain the needed 4 results, but also it will be able to update the predictions within each frame.

To make the system more flexible, various difficulty levels have been added into the robotic system. This can be realised by manually adjusting the prediction accuracy. There are a few goals that the new AI levels will have to accommodate. First, the mistakes must appear to be human-like behaviour. Secondly, it must not be repeatable. Finally, a switch between difficulty levels is available.

The most appropriate method that can facilitate these goals is through the use of a random normal distribution value. This method will deviate from the predicted position and set the range of the standard deviation different for each level. This allows for a more human-like appearance as the robot will always intend to hit the puck. In the event that the robot completely misses the puck, it would generally miss by a smaller deviation compared to a larger value. This is due to the nature of a normal distribution because the larger the deviation from the normal, the lower the probability that the number would appear. To verify this method, we recorded the time a testing player took to score a goal and repeated the experiment for 10 turns. Table I summarizes the experimental results, which show that the time used for different AI levels has proven to deviate as predicted.

TABLE I
AI LEVEL SCORING TEST

AI Level	Shortest Time (sec)	Longest Time (sec)	Total Time (sec)	Average Time (sec)
1 (Easy)	0.68	7.88	48.37	4.84
2 (Medium)	5.05	47.77	275.5	27.55
3 (Hard)	29.23	199.34	1356.96	135.70

E. Hit-Back Trigger

The hit back trigger is tripped by setting a varying threshold that will grow and shrink depending on the speed at which the puck moves. Once the puck passes this threshold, the system will send a hit back flag along with the final location of the puck. When both the trigger has been set and the linear actuator arrives at the predicted location, the punch back system would then trigger to push the mallet to hit the puck.

F. Discussions

Throughout testing we found that the overall performance could be further improved significantly by using a camera of a higher resolution. The present camera has a resolution of 656×491, which means that the maximum resolution that can be used while keeping the entire table in the image is approximately 344 pixel wide. Such a low resolution greatly reduces the maximum accuracy that the hit back can aim for. More specific, we have

Table length = 2100mm
Table width = 1100mm
Puck = 80mm diameter.
Thus, $1100/344 = 3.2$ implies 3.2mm per pixel.

A simple calculation shown below gives

$$\tan^{-1}\frac{3.2}{40} = 4.574° \tag{3}$$

$$2100\tan(4.574°) = 168.0029 \tag{4}$$

where 40 indicating the radius of the puck, we can derive that the maximum accuracy that can be achieved with the present camera is around 168mm.

Since the goal is 250mm wide, even with the best possible aim and a direction straight towards the middle of the goal there is still a chance of missing by 43mm.

The resolution needed to guarantee a goal scoring opportunity will be worked out below. The mallet is 80mm and the goal is 250mm, therefore there is 85mm of scoring area on either side of the mallet. This means in order to guarantee hitting an empty space an accuracy of 42.5mm would be required. In order to get the accuracy of 42.5mm, the resolution that the camera requires can be derived by the following procedure

$$\tan^{-1}\frac{42.5}{2100} = 1.159° \tag{5}$$

$$40\tan(1.159°) = 0.80925 \tag{6}$$

$$\frac{1100}{0.80925} = 1359.283. \tag{7}$$

Therefore, an ideal camera resolution would have to be greater than 2600×1360. However, the higher resolution comes at a higher cost of the system as a whole.

The other thing to consider when selecting a camera is the fps. By using the results gained through extensive testing, the fps would have to be greater than 70, but it would not need to be larger than 100.

The last important specification of the camera is its colour capabilities. Due to the ambient lighting, which can remove the majority of the contrast between the table and puck. It would be highly beneficial if the camera is able to capture colour. This would make the filtering and feature detection trivial and quick. The system will be able to look for a red puck over a blue table resulting in simple RGB thresholding and therefore bypassing the need for memory intensive algorithms.

IV. Control System

Electronic control for the robotic air hockey table is implemented by using an embedded system running off a 32 bit ARM Cortex M4 processor. The processor runs a control loop which is responsible for controlling the actuation of the linear actuator. The 32 bit processor has been chosen as it is more capable of floating point operations as compared to an 8 bit processor. The desired position signals (i.e., reference commands to the control loop) are sent to the processor via serial communication from the vision system. A shaft encoder

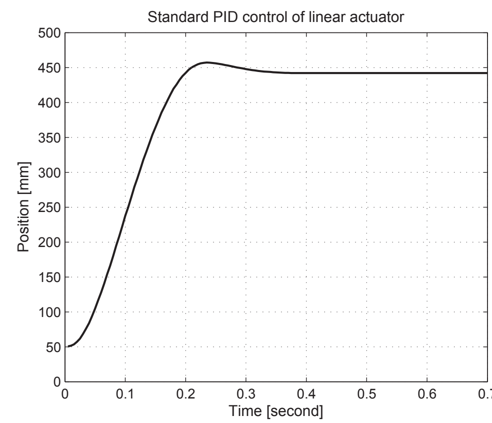

Fig. 7. Standard PID tracking response to a half table width.

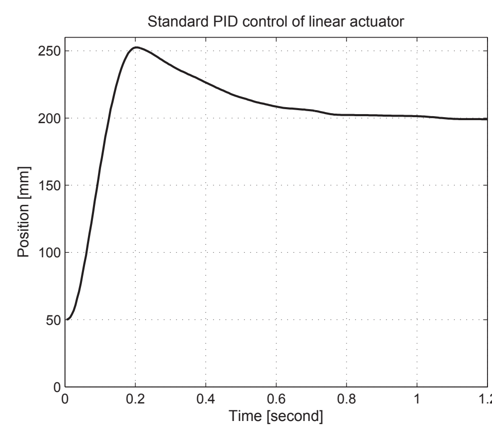

Fig. 8. Standard PID tracking response to a quarter table width.

which measures the linear position of the linear actuator provides feedback signals to the control loop. The control loop runs on a discrete period of 5ms between updates. This allows for sufficient time to calculate the PID controller output.

Initially, we used a standard single PID control loop. However, due to the high power requirements of the linear actuator, the power supply tends to saturate during high acceleration stages of the actuator while changing direction. Because of this, the behaviour of the linear actuator system becomes unstable at times during high exertion movements. More seriously, it may create excessive overshoot which is not allowable as the mallet driven by the actuator may hit the edges of the table. Fig. 7 shows the tracking response curve for a half table width and Fig. 8 shows one for a quarter table width. From the curves, we can see that the overshoots are significant in either response.

To improve performance, we finally designed a dual PID control loop. The block diagram of which is shown in Fig. 9, where r represents the reference command from the vision system, y is the linear actuator position measured by the encoder, and e is the tracking error. The arrow from e to the switch indicates the magnitude of e will determine the switching strategy between the two PIDs. In particular, the two

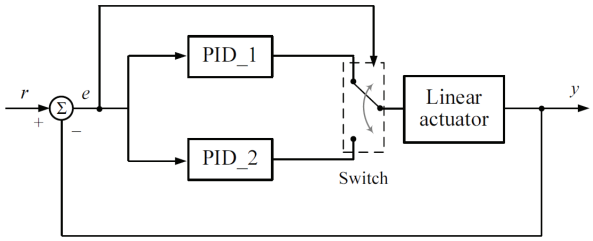

Fig. 9. Block diagram of a dual PID control scheme.

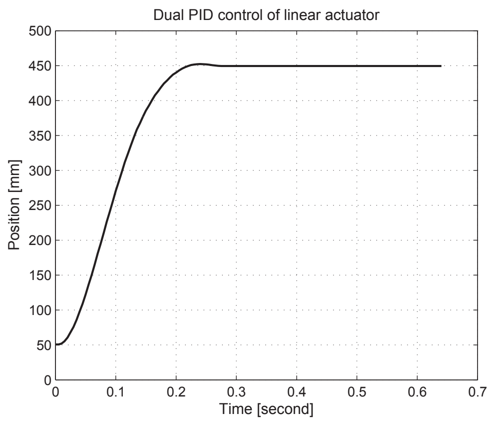

Fig. 10. Dual PID tracking response to a half table width.

PID controllers are individually designed with PID_1 for long distance tracking with fast speed performance while PID_2 for short distance tracking with less control gains to avoid overshoots. In real implementation, the dual control loops are running simultaneously to calculate control input to the actuator. A threshold is set for the tracking error e, i.e., the distance the mallet is away from the final position, to switch the PID controllers. This allows for responsive behaviour regardless of the distance requires, and prevents the saturation of the power supply due to a sudden acceleration. The dual PID tracking response curves are shown in Figs. 10 and 11, respectively. We can clearly see that the overshoots are almost removed and the tracking time is still within the required range.

According to final testing, the performance of the table is greatly improved with the addition of the dual PID control loop. The maximum velocity achieved with this control loop is approximately 4.2 m/s for a long range movement without saturating the power supply.

V. Discussions and Conclusion

The developed robotic air hockey table is capable of providing a solution towards the initial constraints with much more accuracy, speed and functionality than initially expected. The system is able to handle the majority of issues that were found during its implementation through the use of various techniques and algorithms, such as morphology and the dual PID control loop to an exceptional standard.

However, while the system is capable of providing an

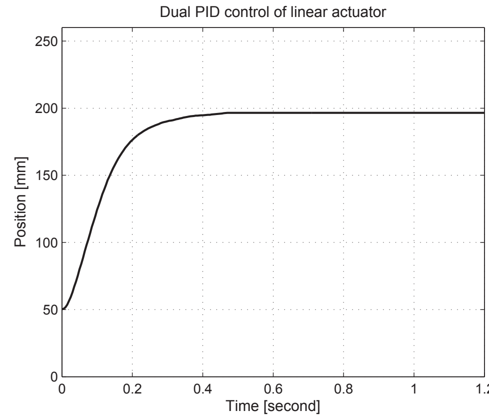

Fig. 11. Dual PID tracking response to a quarter table width.

autonomous opponent for a human player, there are still issues with its consistency. The table is capable of providing a rally against a capable opponent for extended periods of time, but may occasionally fail against the simplest of shots. This is most likely due to compounding errors and jitters caused by both vision system's low resolution and the control system which induces overshoots due to frequent directional changes.

While the system as a whole provides an autonomous player, further development is required to create a system that would be unbeatable and scalable to an expert user's abilities. Some of these further improvements can be achieved through the use of a colour camera with a resolution greater than 2600×1360. The colour component of the camera would increase performance and greatly decrease the difficulty and amount of steps to filter the noise of an image. The higher resolution would increase the accuracy and remove the majority of jitter in the system which would also in turn allow the prediction array to be shortened, resulting in increasing performance.

References

[1] T. Morris, *Computer Vision and Image Processing*, Palgrave Macmillan, 2004.

[2] Sh. Rezvankhah, A. A. Bagherzadeh, H. Moradi, and B. Nadjar Araabi, "A real-time velocity estimation using motion blur in air hockey," in *Proc. International Conference on Robotics and Biomimetics*, 2012, pp. 1767-1772.

[3] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 3rd ed., Reading, MA: Addison-Wesley, 1994.

[4] A. Oliver, "OpenCV vs Matlab vs SimpleCV," Internet: http://simplecv.tumblr.com/post/19307835766/opencv-vs-matlab-vs-simplecv, 2012 [Apr. 24, 2013].

[5] Macron Dynamics, "MSA-628 Actuator — MacSTANDARD Linear Actuators" Internet: http://www.macrondynamics.com/belt-actuator/msa-628, Jul. 17, 2014 [Apr. 10, 2013].

[6] W. Garage, "Camera calibration with OpenCV," Internet: http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html, Apr. 21, 2014 [Apr. 28, 2013].

[7] Basler AG, "Basler Industrial Cameras - A600 Series - A602f," Internet: http://www.baslerweb.com/products/A600.html?model=313, 2014 [Apr. 14, 2013].

[8] W. Garage, "Welcome to opencv documentation! - OpenCV 2.4.9.0 documentation," Internet: http://docs.opencv.org/, Apr. 21, 2014 [Nov. 3, 2013].