Application of Image Recognition in Cyber-Physical Mechatronic System Design

Huaching Chang*, Chuchung Chen , Hsiuhong Yeh , Yuwei Chang , Chiaming Yen
Department of Mechanical Design Engineering, National Formosa University, Taiwan
*Email: 40623128@gm.nfu.edu.tw

Abstract

The purpose of this paper is to demonstrate that the image recognition techniques can be applied to cyber-physical mechatronic system design through the creation of a human vs. computer air hockey robot.

All the parts of this robot were designed by using the full-cloud Onshape [1] and printed through a custom built 3D printer. The motion control of the two degree of freedom air hockey robot is implemented by using Arduino, stepping motors and pulley-belt system to position the hockey mallet. The mechatronic simulation of the air hockey robot is completed in open sourced Coppeliasim [2].

This research confirmed that the virtual and physical air hockey robot can not only be controlled by the physical joystick, but also be performed in the full virtual mode. The virtual image sensor above the platform was designed to send the streaming video of the hockey game from the server to user's web browser, which allows multiple users to monitor or participate in the match at the same time.

Keywords: Air Hockey Robot, Cyber-Physical Mechatronic system, Image Recognition

1. Introduction

In recent years, the integration of image recognition and virtual reality become more and more extensive in the mechatronic product development [4-6]. Moreover, due to the rapid progress of Internet information technology, the impact of virtual and physical integration technology on product development becomes more common [19]. This paper developed an air hockey robot that can be played by human and computer program to use the image recognition to play the match.

There are two major different type of air hockey game design, one use disk puck on an air cushion table and another use hover puck on any ordinary flat surface or table. In this research, hover puck design was chosen to allow air hockey robot to be attached to computer table of any size.

This research is divided into four parts. The first step is to create a physical air hockey robot and use Python program for image recognition to allow the web camera to catch the position of the hover puck, and activate the two-degree-of-freedom motion of the mallet accordingly. The second step is to use CoppeliaSim [2] to simulate the mechatronic air hockey scene, and the third part is to use the Flow Simulation functions of Solidworks and COMSOL to analyze the air cushion dynamics of the fan blades inside the hover puck.

In the end, the virtual and physical air hockey scene were streamed into the user web browser to allow multiple users to monitor or participate into the cyber-physical game.

2. Literature Review

The air hockey table was invented in 1969 and filed for US patent in 1973 [3]. Since then, many physical systems and simulations of the air hockey games were build [4-12]. Among them, a bipedal air hockey playing robot was built to study the research issues incorporated with real-time vision-based control in highly dynamic environments [4]. A 2-player virtual air hockey game was developed by using camera to identify the location of two physical mallets on table, and display two virtual hockey game scenes on the computer screen, one for each player [5]. On the other hand, multiple tabletop games, including air hockey were simulated on a big TV screen through various image recognition techniques [6].

The augmented reality [7] and Fuzzy control [8] were also applied to either control or simulate the real game. In [9-10], the collision impulse effects between the mallet and puck were studied extensively. In [11], a planar 2-link robot arm was designed to control one side of the mallet and improve the air hockey robot performance by applying knowledge through observing human player's eye movement. In [12], a four-bar arm robot was used to predict the playing style of human opponent during the air hockey match and autonomously change associated playing strategy.

Recall the research related to the air hockey games in the past two decades, almost all of them use customized simulation methods, and there is no 3D part printing technology at the time. Therefore, from the simulation perspective, it is difficult for other researchers to reuse the research materials. The tasks completed at that time were also difficult to promote or collaborate. Therefore, in this research the public available open source CoppeliaSim [2] is used to create the air hockey dynamic model and available from [20]. All the X-Y table part files designed in Onshape [1] can also be retrieved from [18].

In addition to share our Computer Aided Design parts and dynamic models of the air hockey robot, Arduino control programs are also uploaded to Github repository [21] for reference.

Futhermore, to allow multiple users to participate in the online air hockey game match at the same time, this research specifically refers to the Cyber-Physical architecture and streamed the image recognition and human-machine interactive scene in CoppeliaSim [2] to user web browser.

3. Design Process

3.1 Physical Design

In order to collaboratively design all parts of the X-Y table system to manipulate the planar motion of the mallet, the full cloud computer aided design system, Onshape [1] was used. The motion control of the hockey mallet is implemented by using Arduino、stepping motors and pulley-belt system to position the mallet. The designed air hockey Onshape [1] assembly file can be accessed from [18].

The mechatronic simulation of the air hockey machine is completed in Coppeliasim [2]. The design of the hover puck uses the air flow generated by the built-in fan to levitate from the platform to reduce the friction, so that it can move smoothly

The control uses the Arduino circuit board to receive the binary data transmitted by the Python module Pyserial for corresponding control, so that the motor can move along X and Y coordinates.

The air hockey robot completed in this research can be assembled on a common computer desk. The hover puck can float on the desk by the downward negative pressure driven by the fan. And the two-degree-of-freedom movement range of the X-Y table can adapt to the size of the table. In achieving the levitation effect, the hover puck needs three AAA batteries, which are arranged into a triangular layout around the motor to achieve dynamic balance.

3.2 Program Development

The image recognition program is written in Python combined with the OpenCV module, and interacts with the virtual environment through the remote api interface of Coppeliasim. The hockey mallet in the physical model is also guided through the same image recognition program. The structure that controls the action of the hockey mallet can be simultaneously used on physical and virtual air hockey game platforms.

In order to accurately estimate the speed and direction of the hover puck motion, we calculate the position difference from each image sensor frame in the image recognition program.

4. Image Recognition

4.1 Design Process

4.1.1 Color Conversion

RGB is the three primary colors of light, namely red, green and blue. The image output is mostly in this form, but this method is very difficult to capture a specific color in settings, so it is converted to HSV Display, HSV stands for Hue, Saturation and Value respectively. Use Hue to adjust to the desired color, and then set saturation and lightness.

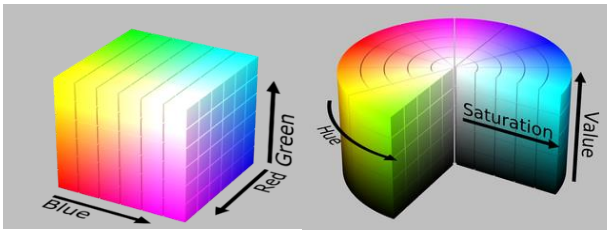

Figure 1: RGB and HSV [22]

4.1.2 Filtration of Color

Use the values of the three colors contained in each pixel to leave the required range, and remove the rest to get the filtered image. When filtering, give the upper and lower limits of the color range. The filtering effect is shown in Figure 2. The filtering range is the lower limit (-10,100,100) and the upper limit (10,255,255).

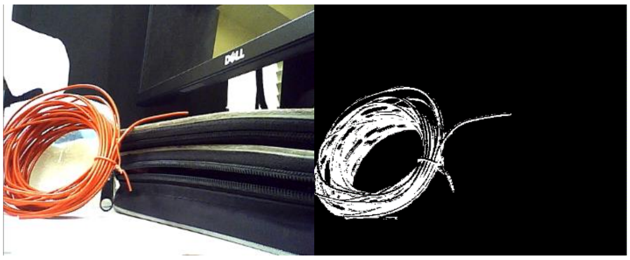

Figure 2: Difference between the unfiltered and the filtered

4.1.3 Noise Removal Process

When using a webcam to take pictures in real time, the image will accompany with noise due to the propagation of electronic signals or other external factors. This noise is likely to cause a lot of unnecessary output during image processing, so it is necessary to perform noise filtering to prevent affecting the accuracy of the output results.

The image noise filtering in this paper has two steps, that is, first mask the noise and then remove the mask noise. Noise masking algorithms mainly use thresholding and segmentation according the color of the hover puck.

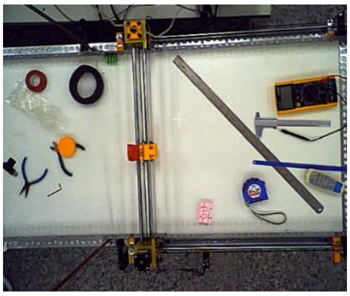

Figure 3: Table with various objects

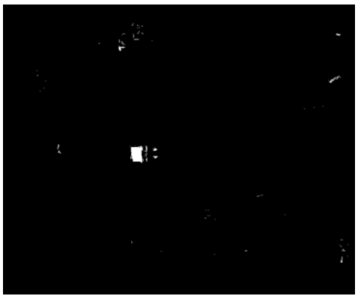

Figure 4: Color filtered image

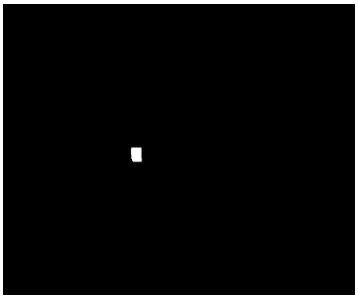

Figure 5: Image after noise removal

4.2 Adjustment of Image

Since the camera used in this study is not directly aligned with the platform, the image needs to transform to parallel with the X-Y table through coordinate conversion. Figure 6 is the desktop image obtained through image sensor and Figure 7 is the adjusted image to align with the X-Y axial coordinates of the table.

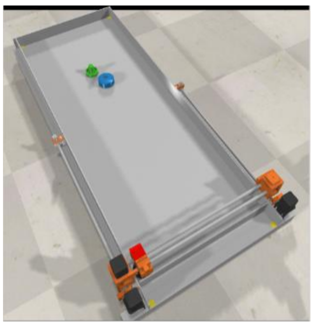

Figure 6: Desktop image obtained by image sensor

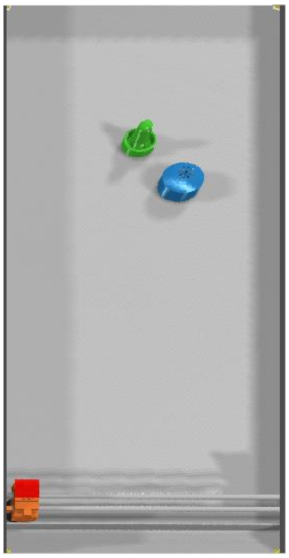

Figure 7: Image adjusted to X-Y table coordinate

5. Python Program

5.1 Translational Velocity of the hover Puck

It takes about 60 milliseconds for the image input to the end of the processing to be calculated by the program itself. Therefore, the number of frames of the processed image is about 17 frames per second. Using the program to calculate the time difference between the processing and the position change of the before and after images, the unit time movement can be obtained. The number of pixel grids is calculated by marking points on the surroundings of the ball table, calculating the length of the unit pixel, and converting the two data obtained to get the moving speed of the hover puck.

5.2 Path Prediction of the hover Puck

5.2.1 Linear Movement

Use image recognition to get the position of the ball, add the position of the previous point, solve the equation of the motion trajectory, and predict the position of the next point from the movement speed calculated above. The image has multiple frames per second, and the predicted path is updated every frame.

5.2.2 Rebound Calculation

The bounce path after hitting the wall is assisted in judging the position of the wall by the mark points around the table, and assuming that the incident angle is equal to the reflection angle, it can be known that the difference between the equations before and after hitting the wall is the opposite slope of the slope. The point of impact is calculated from the equation predicted by the straight line to get the movement path of the ball after rebound.

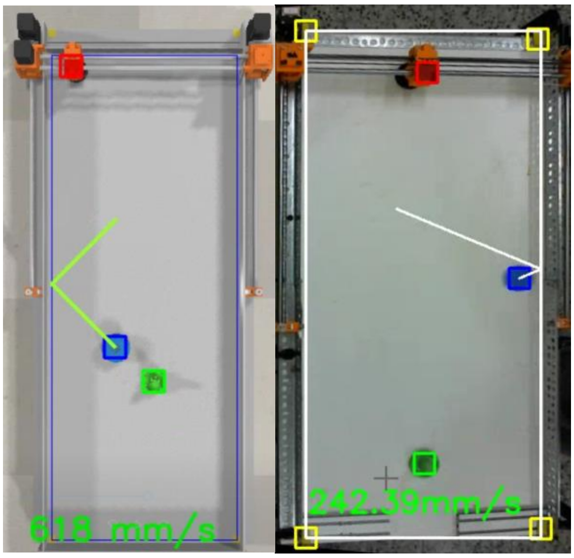

Figure 8: Difference between simulation and modeling

6. Design of the hover puck

6.1 Control Circuit

The circuit system installed inside the hover puck using resistors to make the system current and voltage proportional to avoid overheating of the motor due to the excessive current. The capacitor is used allowing the fan to have a stable output while the fan is running, and the inductance (motor) converts the potential energy into kinetic energy.

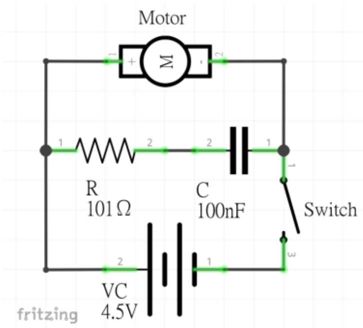

Figure 9: Circuit diagram in fritzing [24]

6.2 Solidworks Air Flow Analysis

The hover puck works like a hovercraft. The fan pushes the air from top to bottom to cause the effect of floating. It mainly depends on the flow at the outlet to float. Flow Q = the product of the flow velocity V times the cross-sectional area A, and the larger the value, the better. The internal pressure of the sphere also has a great relationship. In order to find the best effect between each self-designed shell and fan to verify that the hover puck can reduce friction with the ground when it moves, the finite elements method was used for each analysis item to select the available parameters for combination.

6.2.1 Hover Puck Air-Flow Analysis

To understand the flow of wind when the fan is running, and the direction of flow at the exhaust and inlet holes of the casing, at the same time design a variety of different shapes and perform similar analysis to find the most suitable version of each sub-item, including the hole in the upper cover size, number, size and number of air outlet holes on the inner wall and bottom.

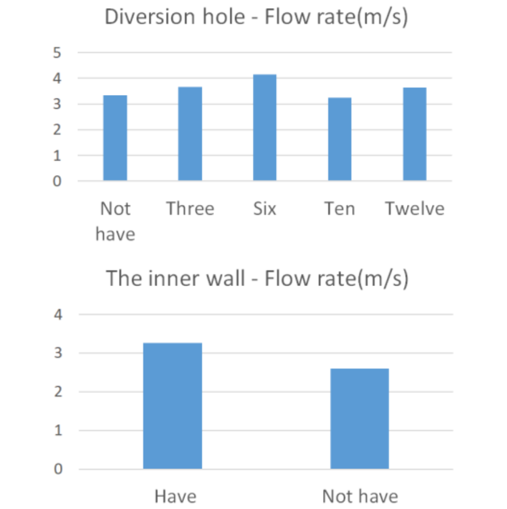

Figure 10, 11: Comparison of each sub-item of the upper cover

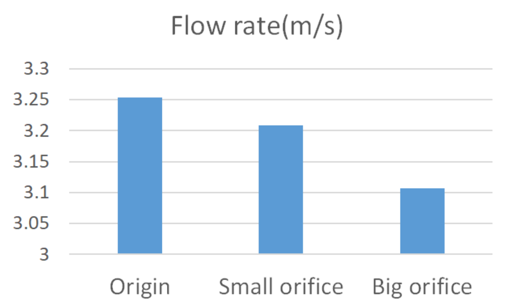

Figure 12: Comparison of sub-items at the bottom

6.2.2 COMSOL Air Flow Analysis

In this section, COMSOL was use to analyze various wind speeds and the number of blades, blade inclination, fillet and type.

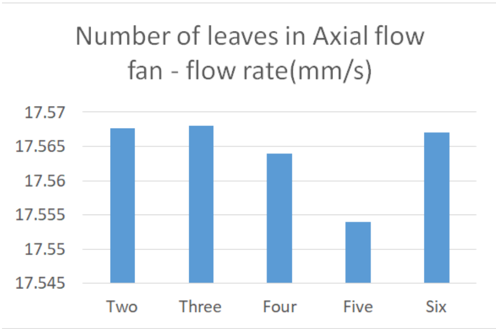

Figure 13: Comparison of the number of leaves

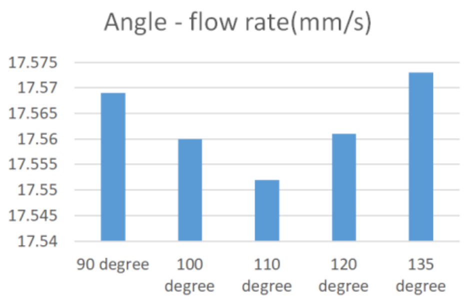

Figure 14: Angle comparison

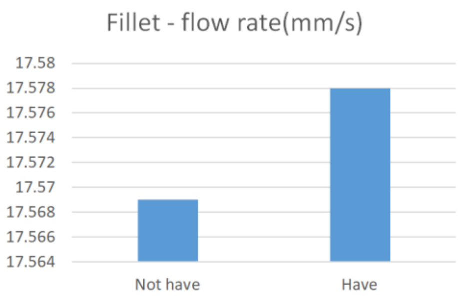

Figure 15: Fillet comparison

6.2.3 Physical Capability

After combining the better parameters in the above analysis, it is tested and found that there is a floating effect and can last for a period of time until the fan loses its rotation function due to insufficient strength.

Figure 16, 17: Before and after puck hovering

7. Cyber-Physical Mechatronic System

The final step of this research is to create an internet interface by using Python Flask [25], and transmit data to the internet interface. The image is transmitted by fast image replacement, and the operation of the web page is retrieved by using JavaScript on the web page to achieve remote control on the web browser.

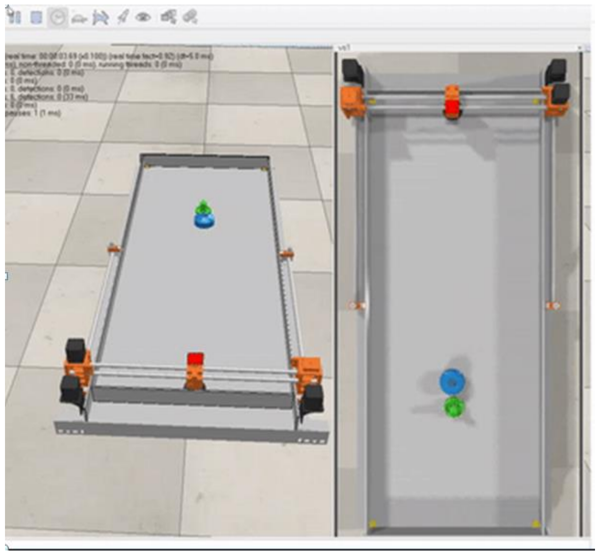

Figure 18: Simulation screen

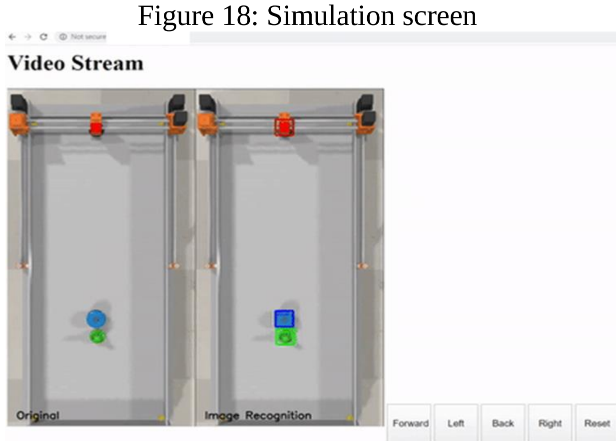

Figure 19: User screen in web browser

The functions shown above can be applied to multi-user collaboration, using real-time web-based streaming to achieve online discussion and speed up the design process. It can also capture real-time control images from web pages, add artificial intelligence, and use the benefits of simulation software. Set up a large number of simulation environments to provide program training, saving a lot of manpower and time costs required during training.

8. Conclusions

This paper completes the design, simulation and manufacturing of ice hockey robot. The goal is to use the latest full cloud computer aided design package and open source mechatronic simulation system to allow reuse of the materials created in this research.

As we commit and push all files into Github repository [21] which include Arduino control programs, Onshape assembly, Coppeliasim scenes and image recognition remote api programs to allow anyone to either recreate this cyber-physical project or extend into other application fields, e.g. machine learning [14-15].

9. Acknowledgements

This work is funded by National Formosa University, Taiwan under the department of Mechanical Design Engineering Project-2020-06-B03004-CD-001 [28].

10. References

1. Onshape, https://onshape.com, accessed September 22, 2020.
2. CoppeliaSim, https://www.coppeliarobotics.com/, accessed September 22, 2020.
3. Air Cushion Table Game, https://bit.ly/36aubhQ, accessed September 26, 2020.
4. Bishop, B., Spong, M., “Vision-based control of an air hockey playing robot,” IEEE Control Syst. Mag. 19(3), pp. 23–32 (1999)
5. Chen, A., Shin, A. and Zhang, Y., "2-Player Virtual Air Hockey", accessed from https://bit.ly/33YM4xj (2014)
6. Basile, M. and Choudhury Z., "La PC-na, " accessed from https://bit.ly/30qWx3N (2016)
7. T. Ohshima, K. Satoh, H. Yamamoto and H. Tamura, "AR/sup 2/Hockey: a case study of collaborative augmented reality," Proceedings. IEEE Virtual Reality Annual International Symposium, pp. 268-275 (1998)
8. Wang, W.J., Tsai, I.D., Chen, Z.D., Wang, G.H., “A vision based air hockey system with fuzzy control,” Proceedings of the International Conference on Control Applications, vol. 2, pp. 754–759 (2002)
9. T. Iguchi, N. Katsuyama, H. Noborio and S. Hirai, "Computing and calibrating collision impulses and its application for air hockey game," IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, pp. 2219-2226 vol.3 (2002)
10. H. Matsuura, N. Abe, K. Tanaka and H. Taki, “Reciprocating vision and auditory sensation through network - constructing virtual air hockey game,” IEEE Conference on Cybernetics and Intelligent Systems, pp. 954-959 (2004)
11. Ogawa, M., et al., “Development of air hockey robot improving with the human players,” IECON 2011, pp. 3364–3369 (2011)
12. Namiki, A., Matsushita, S., Ozeki, T., Nonami, K., “Hierarchical processing architecture for an air-hockey robot system,” IEEE International Conference on Robotics and Automation, pp. 1187–1192. IEEE (2013)
13. Marzano G., Martinovs A., Usca S., “Autonomous Vehicles and Mechatronics Education,” retrieved from https://bit.ly/2FPhDBC, (2019).
14. Taitler, A. and Shimkin, N., “Learning control for air hockey striking using deep reinforcement learning,” International Conference on Control, Artificial Intelligence, Robotics Optimization, pp. 22–27 (2017)
15. Taitler A. and Shimkin N., "Learning to Strike Accurately with DQN-based Algorithms", International Journal of Mathematical Models and Methods in Applied Sciences, Vol11 pp 158-169 (2017)
16. Pürmayr, J., Wolny, B. and Hopf, M., "Low Cost 2D-Hover Puck," Journal of Physics, accessed from https://bit.ly/3i8tS9o (2018)
17. AlAttar A., Rouillard L., Kormushev P., “Autonomous Air-Hockey Playing Cobot Using Optimal Control and Vision-Based Bayesian Tracking,” retrieved from https://bit.ly/3hXHunL, (2019)
18. Onshape Air Hockey Assembly, accessed from https://bit.ly/2SuCNYu September 26, 2020
19. Edward A. Lee, " Cyber Physical Systems: Design Challenges", Technical Report of University of California, Berkeley, US, (2008)
20. Coppeliasim models, accessed from https://bit.ly/36gndrH (2020)
21. Arduino control programs, accessed from https://bit.ly/3339LoX (2020)
22. The RGB Gamut, Retrived from https://en.wikipedia.org/wiki/HSL_and_HSV
23. HSV cylinder, Retrived from https://en.wikipedia.org/wiki/HSL_and_HSV
24. Fritzing, https://fritzing.org/ accessed September 26, 2020
25. Flask, https://flask.palletsprojects.com/ accessed September 26, 2020
26. 張浩祥,“探討影像解析度對於移動物件追蹤之影響,”國立高雄科技大學電腦與通訊工程系碩士論文，台灣 (2018)
27. 陳俊宏,“以彩色影像技術為基礎的動態物件追蹤系統,”國立臺灣大學資訊工程學研究所碩士論文，台灣 (2008)
28. Chang H., Chen, C. , Yeh, H. , Chang, Y., "Application of Image Recognition in Cyber Physical Mechatronic System Design, " Final Project Report, accessed from https://bit.ly/30cR5RE (2020)