An Arduino-Based Wearable Gesture Device as a Controller for Quadcopter Drones

Clement Ong
De La Salle University
Manila, Philippines
clement.ong@dlsu.edu.ph

Jon Bryan Dela Cruz
De La Salle University
Manila, Philippines
jb_delacruz@dlsu.edu.ph

Yasmin Audrey Tutanes Datario
De La Salle University
Manila, Philippines
yasmin_audrey_datario@dlsu.edu.ph

Juan Christian Geronimo
De La Salle University
Manila, Philippines
juan_geronimo@dlsu.edu.ph

Justin Thomas Lanting
De La Salle University
Manila, Philippines
justin_thomas_lanting@dlsu.edu.ph

Lester Andrei Opulencia
De La Salle University
Manila, Philippines
lester_andrei_opulencia@dlsu.edu.ph

ABSTRACT

Drones are typically controlled using a joystick-based controller. Due to its weight and ergonomics, this type of controller requires practice, dexterity, and attention to successfully pilot a drone. Wearable technology has surfaced as an unobtrusive mechanism to provide sensors and tracking devices for varying domains. In this paper, we describe the design and characterization of a camera-free, Arduino-based wearable gesture device that fits onto a person's hand and wrist. The device is designed to track hand movements and finger curl, which are then translated to drone flight commands. Experiments to validate the performance of a prototype drone controller were executed using an obstacle course to test several drone movements. Results show that for a total novice, the prototype controller achieved comparable navigability, and reduced collision frequency, slightly longer test course completion times but less drone crashes.

KEYWORDS

Quadcopter drones, drone controllers, wearable gesture devices, Arduino

1 INTRODUCTION

A drone or unmanned aerial vehicle (UAV) is an aircraft that can be remotely controlled. It can be classified according to its characteristics such as the number of rotors, level of autonomy, size and weight, and energy source [18]. Multirotor systems usually consist of four or more rotary wings or rotors to create lift and to keep them flying. Originally only utilized for military purposes, surveillance, and environmental monitoring [17], drones have found increasing usage in mainstream commercial and personal applications for asset inspection, traffic monitoring, aerial mapping, delivery services, photography [15]. Drones help in disaster management, through weather forecasting of storms, geographic mapping of inaccessible locations, and thermal sensor drones for search and rescue operations [11]. Use of drones for spatial social sciences [6] and in healthcare settings has also seen steady increase in recent years [14].

Drones are operated through remote ground control systems. The most common type are handheld joystick controllers that communicate through radio signals. The joysticks are used to execute motions on the horizontal plane such as forward, backward, left, and right, and positive or negative rotations around the vertical axis and changes in the altitude of the aircraft [1]. Despite its heavy weight, joysticks are robust and allow users to maneuver the movement of the drone, thus becoming the preferred controller despite alternatives offered by smartphones. Operating joystick-based controllers, however, are complicated and unnatural for humans [9]. They require refined motor skills, practice, and dexterity to maneuver the miniature aircraft around successfully [1]. Those with hand malformations or people with hindrances in learning new things would find it challenging to use a joystick-based controller, especially for first-time users of quadcopter drones.

The utilization of more intuitive inputs such as those afforded by wearable devices as remote drone controllers may lead to ease in piloting these aircraft. Wearable technologies are pieces of clothing or accessories with embedded electronics that can be worn with ease since they can be used hands-free [7, 19]. They carry fully functioning portable computers and can execute a task by detecting gestures. Chanda [3] claims that "the output and the functioning of machines will be more intuitive if they are communicated using human gestures". Thus, transforming controllers into a more natural way of operation using wearable gesture control devices may improve their usability.

2 MOTIVATION

In this paper, we describe our work in designing an Arduino-based prototype wearable gesture device for controlling quadcopter drones - the intent is to develop a controller that allows a novice to fly the aircraft indoors as well as outdoors while reducing the inherent incidences of mishaps. Gestures detected by the wearable device are translated into a set of commands for the drone. We then evaluate the viability of the alternative controller through several quantitative and qualitative measures, including navigability, collision frequency, number of crashes, and course completion time with results comparable to the traditional joystick-based controller.

3 RELATED WORK

Previous research investigated various strategies in controlling drones, from handheld joystick controllers to touchscreen input and body suits that mimic the motions of the human body, to motion sensing cameras for tracking hand gestures. Motion-based gesture controllers have been found useful in controlling devices and robots wirelessly through sensors.

An open-source library for vision-based flight control was developed [13] recognizing eight poses with good accuracy. Although intended as a means to control a drone with a camera facing the user, the system was developed with a camera on a computer. As is typical with vision-based approaches, the recognition accuracy depends highly on environment, i.e the amount of background clutter and lighting conditions, and the target (hand) being in the FOV at a (system-trained) distance of approximately 1 meter. It is not hard to imagine outdoor situations where such an approach may not work well, such getting glare from the sun, the hand in shadow, or low-light in the early morning or sunset.

A commercially available drone, Spark developed by DJI [1], flies off the user's hand and is controlled through hand gestures. The Leap Motion device from Gubsci and Zsedrovits [5] functions as a motion sensing camera with the palm position being used for determining if the drone should keep its position idle or to specify velocity commands. Similarly, [16] uses the Leap motion controller to pilot an AR Parrot drone, however paper does not elucidate on the gesture to control mapping, except for takeoff and landing sequences. Weaknesses of the LEAP controller include that it is limited to hand and finger gesture, a small working zone and must remain within approximately 1 meter distance of the user's hands. Moreover, because it uses infrared technology, it is susceptible performance degradation in sunlight, and thus cannot be used outdoors.

[20] describes a web-based wireless interactive control system based on hand gestures , which follows the same control commands as the Leap controller but with different hand gestures and does away with the need for any device to control the drone. However, the lack of precision of the solution restricts it to straightforward and specific activities, mainly recreational, where the user must be the focus of the quadcopter's attention. A gesture controller that uses a wearable glove technology instead of an external sensor is reported in [2]. The glove tracks regions of interest based on the trajectory projections of hands.

Motion-based gesture controllers are based on accelerometer technology to control the device wirelessly [3]. These controllers measure the acceleration of a body in space, may it be in the form of movement or vibration. They can also be designed to read body language or any action that a human performs to control several types of technologies. Mapping the motion of human body parts such as fingers, limbs, and torso, for operation and maneuver makes these devices more intuitive to use, requiring less skill and practice to maneuver the miniature devices, such as drones [4]. Still, refined motor skills, practice, and dexterity are needed to feel the ease of using such controllers [1].

Smartphones and tablets integrate well-developed, finger-based "gestures" in the form of various single and multi-finger actions, such as tap, double-tap and drag. Moreover, their touchscreens can also provide a sense of finger pressure information. Drone controllers have been developed around such devices, i.e. [8], which "abstracts the drone and camera into a single flying camera object". Meant for aerial photography, motion and attitude control of the drone/camera are fused into smartphone/tablet's attitude and touchscreen interface. This system's target application is fairly challenging and goes well beyond casual flying. Simpler finger-touch based controllers are installable applications as entry-level flight controls for several commercially-available drones, such as the Parrot Bebop. The DJI Fly app controls several of DJI's drones, such as the Mini, Air and Mavic[1]. The limitations of touchscreen-based controllers include the lack of tactile feedback and the arbitrary mapping of motions such as tap and double-tap and multi-finger / hand requirements, i.e. use of smartphones and tablets, as commercially implemented, is paired with entry-level drones.

Single-hand controllers have also been developed for drone piloting. Normally associated with controlling wheeled RC vehicles, the additional degrees of freedom for drone attitude and movement are input via an embedded accelerometer or intertial measurement unit (IMU) in the controller and joystick, such as the DJI RC Motion 2[2]. A fair number of flight functions are mapped into this controller, with concurrent inputs required for piloting. The joystick is mapped to up/down and left/right motion; the trigger (accelerator) moves the craft forward and backward proportionately. Pitch and yaw is controlled by the attitude of the motion controller. Some motions may not necessarily be intuitive, such as tilting its body 90° to make the drone land. The price of the controller is reflective of its intended application and users.

Wearable gesture devices are often integrated with everyday clothing and accessories to serve varying purposes [19]. Fitness watches, for instance, are designed to track vital body signs such as heart rate and temperature. Gloves and exoskeletons can track body movements and have been applied to control machines and robots. These devices can be programmed to detect gestures that enables controlling other devices [2, 10, 12]

In summary, many approaches to a "better" flight controller have been developed and commercialized. Some designs are meant for indoor use only, with mobility limitations. Image processing and extraction of gestures is compute-intensive, although hardware-specific advancements have almost made this a non-issue. However, challenges remain when environmental conditions make image registration difficult. Commercially-available controllers have advanced features that the novice may not need, and can result in a steeper learning curve, with a price-point that also reflects these advanced features. In our opinion there is room for a controller catered specifically to the novice to allow the pilot to simply enjoy the experience, indoor or outdoor, while reducing the possibility of severe mishaps that could result in expensive repairs or loss of the craft.

4 PROTOTYPE DEVELOPMENT

The design of the wearable controller device to be worn as a glove is depicted in Fig. 1. It incorporates pockets to hold the sensors in place while preventing the sensors from being subjected to outside factors that could cause damage due to extreme exposure. The resulting prototype has three components: (i) the Raspberry Pi (RPI) shown in Fig. 2, (ii) the sensors shown in Fig. 3, and (iii) the Arduino shown Fig. 4.

The component holding the RPI in Fig. 2 is attached to the portion of the arm above the elbow. Using the same materials as the wristband, this part of the device is attached to a smaller band for breathability. A battery pack is added to supply power, which is

[1]https://www.dji.com/downloads/djiapp/dji-fly
[2]https://www.dji.com/rc-motion-2

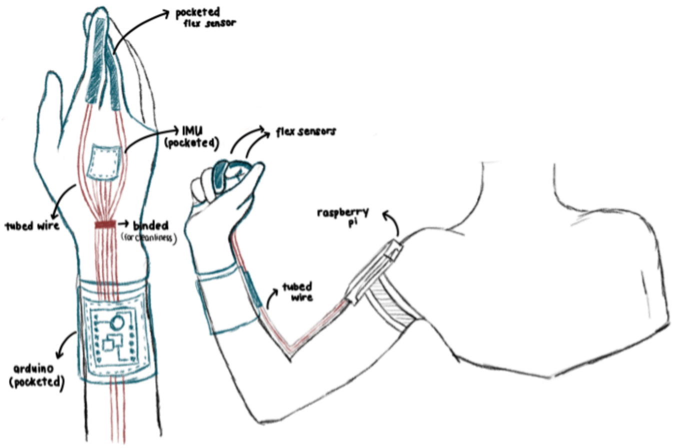

Figure 1: Preliminary design of the wearable device.

placed on the other shoulder alongside the RPI. The actual glove shown in Fig. 3 is cut open on the thumb, ring, and pinky fingers to allow for better ventilation. The glove houses several sensors: the index and middle fingers contain the flex sensors, while the IMU is located on the dorsal side of the hand. Shown in Fig. 4 is the wristband made from leather. It holds the Arduino and utilizes different clasps for its fastening to universally fit the hands of a larger population base. The wristband provides paddings between the Arduino's shape and the wrists' natural shape for comfort.

The wearable prototype was then programmed to connect and interact with the drone to allow the latter to be controlled with the poses indicated in Table 1. The Arduino digitizes the flex sensor voltage outputs and communicates with the IMU, then through this data it determines the pose. The pose is communicated to the RPI through the connection between the digital pins on the Arduino

Figure 2: Raspberry PI.

Figure 3: The sensors.

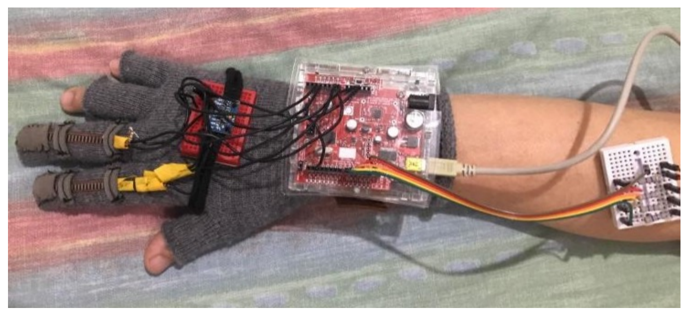

Figure 4: Arduino.

and general-purpose input/output pins on the RPI. When a pose is done, it activates a combination of pins that the RPI recognizes to send the corresponding command to the drone.

5 METHOD

We conducted two types of validation: controller experiment and end-user experiment. Controller experiment involves testing the

Table 1: Hand poses programmed for the wearable drone controller.

Pose	Drone Movement
Hand tilted forward	Pitch forward
Hand tilted backward	Pitch backward
Hand tilted right	Roll right
Hand titled left	Roll left
Straight hand pointed to the right	Rotate right
Straight hand pointed to the left	Rotate left
Fist	Decrease altitude
Index finger pointed forward	Increase altitude

drone connection with the wearable controller device to determine its recognition accuracy and response time. For each pose, a minimum of five trials were performed to check the pose recognized based on the values from the sensors. End-user experiment involves the collection of quantitative measurements and qualitative feedback. Quantitative measurements include the average response time and the accuracy of the controller in recognizing different hand gestures. Flights on an obstacle course are used to determine the usability of the controller, based on the number of times the drone crashed and the time it took for the drone to be maneuvered from one sector of the obstacle course to another. On the other hand, qualitative feedback are used to determine users' perception of the controller in terms of its intuitiveness and ease of use.

The *Obstacle Course Assessment* 5 was developed to help the participants familiarize themselves with the programmed gestures and to track their progress with the specific controls. It has two parts: practice session and main obstacle course. The practice session familiarizes the participant with the eight (8) flight control functions and the response time of the drone to the controller commands. The control functions make the drone to move upward, downward, to the left, right, forward and backward, as well as rotate left and right. Once participants exhibit adequate mastery of these basic flight functions, they are allowed to proceed to the obstacle course.

The obstacle course has an oval-shaped layout with five sectors, each requiring certain combination of drone flight, shown symbolically in , along with the required flight path of the drone. Sector A is a "hall" is made of corrugated carton, forming a passage with a width of 40cm. Targeting the pitch and roll control, this sector assesses the ability to control the drone to move horizontally and vertically. Sector B has two vertical pipe structures, and is navigated by moving forward and tilting at the same time. This evaluates the ability to control the drone to move forward and rotate clockwise and counterclockwise at the same time. Sector C focuses on the combination of simultaneously moving forward while moving left and right, while Sector D determines the drone operator's ability of going through a narrow passage with "backward" (relative to the drone operator) flight pattern. Sector E is composed of three rings that measure 40cm and are placed at differing locations and heights, requiring concentration and skill since the operator must have the ability to move upward and downward while simultaneously moving left and right to "thread" the drone through rings. The time of flight per sector, the average time, and the whole obstacle courses' total time are recorded alongside the number of crashes per sector.

We invited five (5) participants to perform experiments with the traditional and our prototype wearable controller: three (3) used the traditional controller and two (2) used the prototype wearable controller. The participants had minimal to no experience with flying drones. After the experiments, we administered surveys to collect their individual evaluation of the controllers' intuitiveness and usability attributes using a 5-point Likert scale. We also conducted debriefing sessions with the two (2) participants who used the wearable controller. The questions focused on collecting their perception of comfort and overall experience in performing the drone control actions.

6 RESULTS

We present the results from our experiments with the prototype controller, end-user experiments, and users' evaluation of the prototype controller.

6.1 Prototype Controller Tests

We performed three types of tests on the controller: recognition accuracy, program response time, and controller delay.

Table 2 shows a sample of the experiments that were conducted to evaluate the accuracy of recognizing the poses, in this case. For each pose, a minimum of five trials was performed and we recorded the detected pose and the values from the sensors. All poses recorded a recognition accuracy of 100%.

The response time of the components of the wearable controller was also measured for each of the pose. Table 3 shows the sample results collected from measuring the response time in milliseconds for the *forward* drone command. The average time for the pose to be recognized by the Arduino is 3.2 ms, while the overall average response time is 4.0824 ms. However, the response time varies depending on the pose for the RPI to read the data from the Arduino and return from the command task, due to the positioning or order in the code. In this instance, the forward pose is among the bottom at the order in the code thus the time to read is greater than the time to return. On average the read time is 0.8370 ms and the return time is 0.0118 ms.

The delay for each controller was tested as well. The input command was recorded alongside the drone response. The video was then slowed down to check the delay between the input to the response. This was done five times for each controller, the results can be seen in Table 4. The difference between the traditional, 27.60 ms, to the wearable, 28.60 ms, is just 1.00 ms which is insignificant.

6.2 Practice Session

A total of eight (8) drone actions were performed by the participants using the traditional (TRAD) and the wearable (WEAR) controllers. Fig. 6 shows the average times in performing each of the actions. TRAD participants completed the actions faster by an average of 8.45s. The largest time difference is 11.92s for the *roll left* action, while the *rotate left* has the smallest average time difference of 5.68s.

In Fig. 7, it can be seen that TRAD participants were more effective in performing drone actions on the practice session, requiring fewer attempts from the participants using WEAR. An average of 1.375 attempts on each drone action was performed by TRAD participants and none exceeded 2. The WEAR participants, on the other hand, had an average of 2.5 tries.

6.3 Obstacle Course

The results in the obstacle course experiment showed improvements in the performance of WEAR. As seen in Fig. 8, WEAR participants completed the first three sectors in a much shorter time than those using TRAD. For sector 4 however, TRAD was ahead by as much as 20s, while the time difference for sector 5 was 4s, in favor of TRAD. Overall, TRAD was 12s faster than WEAR with an average course completion time of 76s.

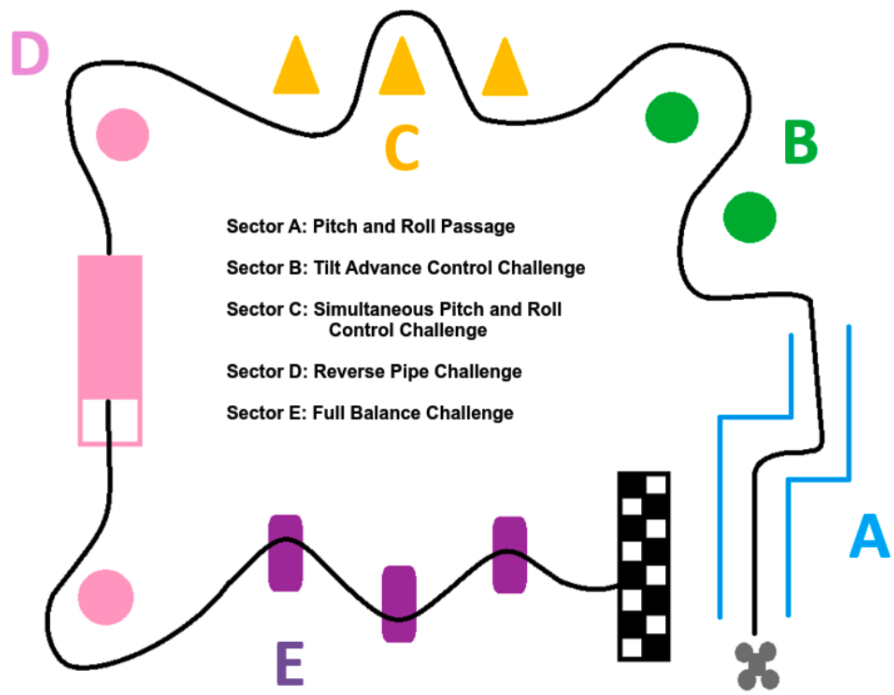

Figure 5: Design of the obstacle course.

Table 2: Sample results of gesture recognition accuracy test for the general purpose input/output connection.

	Trial 1	**Trial 2**	**Trial 3**	**Trial 4**	**Trial 5**
Actual pose	**Forward/Positive Pitch (Hand Tilted Forward)**				
Pose recognized	Forward	Forward	Forward	Forward	Forward
Value of *AngleY* from IMU	-31	-44	-25	-50	-22
Threshold	AngleY <-20				
Deviation from the threshold	-11	-22	-5	-30	-2

Table 3: Sample results in measuring the program response test (in milliseconds) for the different poses.

	Trial 1	**Trial 2**	**Trial 3**	**Trial 4**	**Trial 5**	**Average**
Pose to Arduino recognition	3.0000	3.0000	3.0000	4.0000	3.0000	**3.2000**
Drone command	**Forward/Positive Pitch**					
Arduino to RPI drone command	0.8569	0.8263	0.8187	0.8650	0.8154	**0.8370**
RPI return from command	0.0119	0.0112	0.0126	0.0122	0.0110	**0.0118**
Total response time	3.866	3.8375	3.8313	4.8772	3.8264	**4.0824**

Table 4: Results from the controller delay test.

Call to Drone response	**Trial 1**	**Trial 2**	**Trial 3**	**Trial 4**	**Trial 5**	**Average**
WEAR	29.00	30.00	27.00	30.00	27.00	**28.60**
TRAD	30.00	28.00	26.00	26.00	28.00	**27.60**

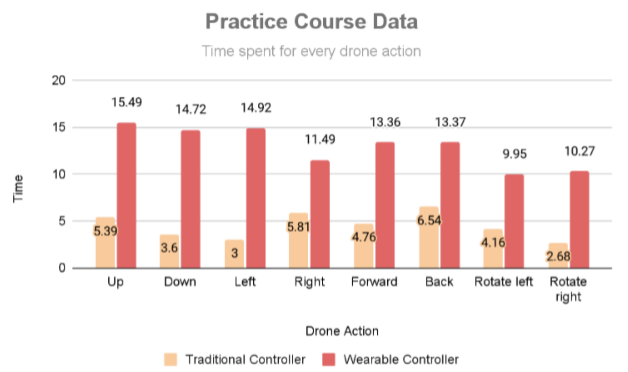

Figure 6: Average time (in seconds) spent on each drone action.

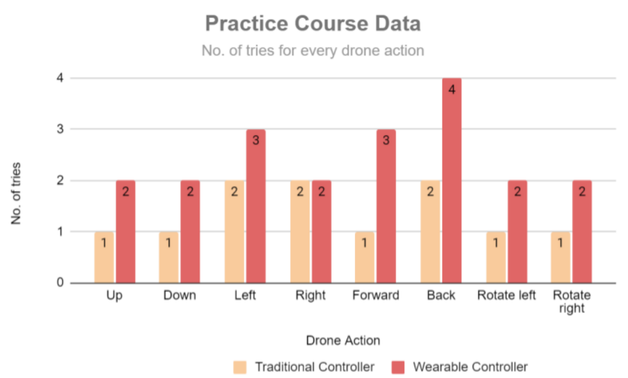

Figure 7: Average number of tries spent on each drone action.

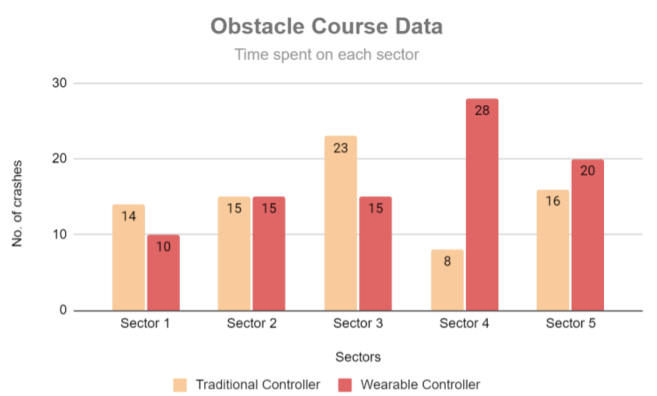

Figure 8: Time (in seconds) spent to complete each sector of the obstacle course.

We also counted the number of crashes and near hits. A *crash* occurs when the drone comes to a complete stop after a collision, while a *near hit* is counted when the drone collides with an object but continues onward. After a crash, the drone will be positioned at

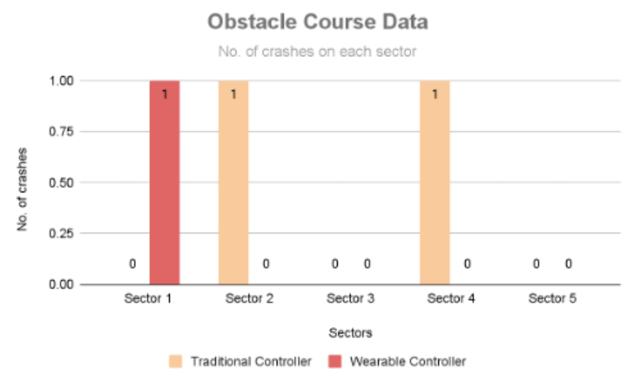

Figure 9: Number of crashes per sector.

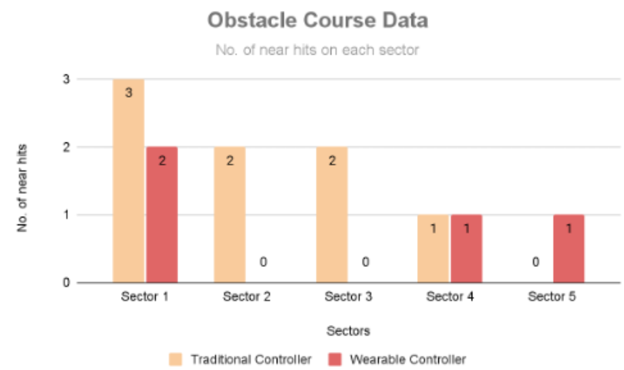

Figure 10: Number of near hits per sector.

the starting point of the current sector. In Fig. 9, it can be observed that a total of two crashes occurred using TRAD while only one crash using WEAR. In Fig. 10, WEAR had a total of four near hits in sectors 1 (2 near hits), 4 and 5 (1 each) as opposed to the eight near hits using TRAD.

6.4 User Evaluation

Fig. 11 shows the participants' evaluation of the intuitiveness of the two controllers by rating the following items:

(1) It was easy to learn how to use the controller.
(2) The controller actions are simple to perform.
(3) Performing the actions using the controller is natural for me.
(4) I had no trouble executing the actions and gestures to command the drone.
(5) I did not notice a delay in the drone's response to my command.

The wearable controller received a higher rating compared to the traditional controller for all items except item 4: because the prototype controller uses a third-party WiFi radio module to communicate with the drone, WEAR participants reported encountering connection problems during the experiment. Item 3 received the most significant difference of 1.7 in the rating, underscoring the perceived intuitiveness of the wearable controller.

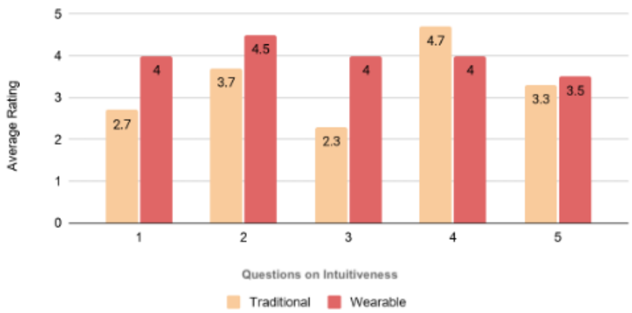

Figure 11: User perception on the intuitiveness of the controllers.

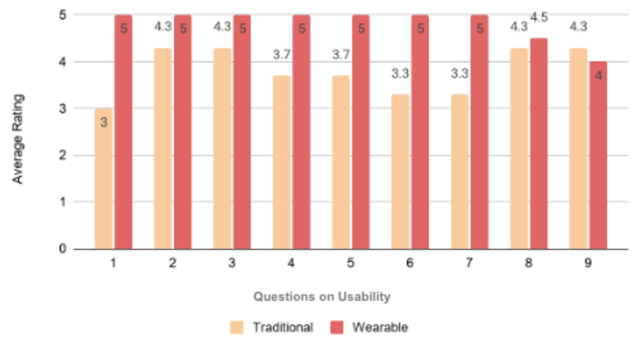

Figure 12: User perception on the usability of the controllers.

Fig. 12 shows the participants' evaluation of the controller's usability based on the execution of specific actions. WEAR outperformed TRAD in eight out of nine usability items, with participants successfully executing the actions that instruct the drone to maintain its position (1), go up (2), down (3), forward (4), backward (5), towards the left (6), and towards the right (7). Items 8 and 9 evaluate the ease in using the controller to command the drone to perform rotations, achieving the lowest in commanding the drone to rotate to the right (9).

7 DISCUSSION

Practice sessions took longer to master with the WEAR controller as these are calculated based on the accumulated time of the attempts to complete the required action, i.e. for the "Up" action, TRAD averaged one attempt in 5.39s while WEAR required 2 attempts to do the same. These time differences are significant primarily due to the small number of attempts that were needed to accomplish the action and do not necessarily reflect that the WEAR controller is harder to master. It might also be helpful to clarify that the TRAD controller is an app that runs on a smartphone and displays virtual buttons on the phone's touch screen that executes the eight poses required for the practice session. It might not be unreasonable to expect that TRAD users will require only one attempt for each action.

Being able to make a drone go to a particular pose (position and orientation) in a specific flight pattern in three-dimensional space requires the pilot to mix the eight basic flight actions from the practice session, and this is assessed by the obstacle course. Flight times for sectors 1, 2 and 3 show that WEAR is at least as fast, or faster than TRAD, which means that mixed actions are easier to accomplish on the WEAR controller. The time difference for sector 5, considered the most challenging section of the course, is 4s, where WEAR is 25% slower. This sector requires precise 3D positioning of the drone and may indicate that simultaneous up/down and left/right movements are more difficult to encode with the poses chosen for WEAR. It can be noted however that this is the same time difference for sector 1, where WEAR takes a shorter time.

The significant difference between the two controllers is in sector 4 where the drone is required to avoid a vertical pole and fly into a tunnel backward relative to the viewpoint of the pilot. This is a combination of precise 3D positioning while in flight as well as reversing the vision-to-control-requirement. Since sector 5 has shown that 3D positioning is more challenging for the WEAR pilot, the even-larger time discrepancy for sector 4 may point to a larger difficulty in reversing the eye-movement coordination. We posit that this may have been brought about by having a more intuitive (forward movement) interface.

Examining the collision rate for crash and near-hits, WEAR has half as many accidents versus TRAD, 2 vs. 4 and 4 vs. 8, respectively. This lower collision rate indicates that the what the pilot sees the drone doing versus the required flight path (specifically to avoid hitting an object) is more easily corrected using WEAR. It might be argued that the lower collision rate is due to the pilots using WEAR flying slower - however this is countered by the equal or shorter flight times for sectors 1, 2 and 3. Given this, the WEAR controller may be interpreted to be more intuitive to use.

The above assertion is substantiated by the results of the user evaluation on the controller's intuitiveness, where 4 out of 5 questions had the WEAR controller favored. The results for each of the 8 basic flight actions reflect that the pose required to rotate the drone left, and even worse, right, may have to be recast, or perhaps the thresholds required for recognizing the pose have to be adjusted.

8 CONCLUSION

We developed a prototype wearable controller that recognizes eight hand poses, to control a quadcopter, with the objective of providing a more intuitive interface for flying the aerial vehicle. The hand pose instrumentation uses flex sensors positioned on the index and middle fingers, and an inertial measurement unit at the back of the hand. The pose is recognized by an Arduino and sent to a battery-powered Raspberry Pi which then sends the flight control command via WiFi to the drone.

Results show that users typically take two to four attempts to learn the proper pose for each flight action. The wearable controller exhibited comparable albeit slightly longer completion times against a traditional controller for an obstacle course whose sectors require various combined flight actions to accomplish, while having half the number of mishaps. Moreover, users reported a positive experience with the wearable controller, scoring higher in both intuitiveness and usability versus the traditional controller. The wearable controller however did require a longer time to accomplish individual flight actions during familiarization. Despite this, overall we believe the controller has good potential, in line with the use-case stated in the Motivation section.

Future work will include refinement of recognizing existing poses and tuning the magnitude of the response to the pose to possibly accomplish more responsiveness without losing controllability. A larger set of participants for testing is desirable, and these can be carried out in an area that has less radio-frequency interference to avoid connectivity issues originally experienced during the development of this prototype.

REFERENCES

[1] 2018. *UAV for Everyone: An Intuitive Control Alternative for Drone Racing Competitions*. https://doi.org/10.1109/TISHW.2018.8559538

[2] Vladislav Berezhnoy, Dmitry Popov, Ilya Afanasyev, and Nikolaos Mavridis. 2018. The Hand-Gesture-Based Control Interface with Wearable Glove System. In *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics*, Vol. 2. 448–455. https://doi.org/10.5220/0006909304480455

[3] Premangshu Chanda, Pallab Kanti Mukherjee, Subrata Modak, and Asoke Nath. 2016. Gesture Controlled Robot using Arduino and Android. *International Journal of Advanced Research in Computer Science and Software Engineering* 6, 6 (June 2016).

[4] Mesut Cicek. 2015. Wearable Technologies and Its Future Applications. *International Journal of Electrical, Electronics and Data Communication* 3, 5 (April 2015).

[5] Gergely Gubsci and Tamas Zsedrovits. 2018. Ergonomic Quadcopter Control Using the Leap Motion Controller. In *2018 IEEE International Conference on Sensing, Communication and Networking (SECON Workshops*. 1–5. https://doi.org/10.1109/SECONW.2018.8396348

[6] Ola Hall and Ibrahim Wahab. 2021. The Use of Drones in the Spatial Social Sciences. *Drones* 5, 4 (2021). https://doi.org/10.3390/drones5040112

[7] Seok Chan Jeong, Sang-Hyun Kim, Ji Yeon Park, and Beomjin Choi. 2017. Domain-specific Innovativeness and New Product Adoption: A Case of Wearable Devices. *Telematics and Informatics* 34, 5 (August 2017), 399–412.

[8] Hao Kang, Haoxiang Li, Jianming Zhang, Xin Lu, and Bedrich Benes. 2018. Fly-Cam: Multitouch Gesture Controlled Drone Gimbal Photography. *IEEE Robotics and Automation Letters* 3, 4 (2018), 3717–3724. https://doi.org/10.1109/LRA.2018.2856271

[9] Guan-Chun Luh, Heng-An Lin, Yi-Hsiang Ma, and Chien Jung Yen. 2015. Intuitive Muscle-Gesture Based Robot Navigation Control using Wearable Gesture Armband. In *2015 International Conference on Machine Learning and Cybernetics*. IEEE. https://doi.org/10.1109/ICMLC.2015.7340953

[10] Leonardo Meli, Davide Barcelli, Tommaso Lisini Baldi, and Domenico Prattichizzo. 2017. Hand in Air Tapping: A Wearable Input Technology to Type Wireless. (2017).

[11] Sharifah Mastura Syed Mohd Daud, Mohd Yusmiaidil Putera Mohd Yusof, Chong Chin Heo, Lay See Khoo, Mansharan Kaur Chainchel Singh, Mohd Shah Mahmood, and Hapizah Nawawi. 2022. Applications of Drone in Disaster Management: A Scoping Review. *Science Justice* 62, 1 (2022), 30–42. https://doi.org/10.1016/j.scijus.2021.11.002

[12] Morais. 2016. Application of Myo Armband System to Control a Robot Interface. In *9th International Conference on Bio-inspired Systems and Signal Processing (Biosignals 2016)*. https://doi.org/10.5220/0005706302270231

[13] Kathiravan Natarajan, Truong-Huy D. Nguyen, and Mutlu Mete. 2018. Hand Gesture Controlled Drones: An Open Source Library. In *22018 1st International Conference on Data Intelligence and Security*. 168–175. https://doi.org/10.1109/ICDIS.2018.00035

[14] Mario Poljak and Anja Šterbenc. 2020. Use of drones in clinical microbiology and infectious diseases: current status, challenges and barriers. *Clinical Microbiology and Infection* 26, 4 (2020), 425–430. https://doi.org/10.1016/j.cmi.2019.09.014

[15] James Rennie. 2022. What is a drone used for? Commercial uses and applications in various industries. https://www.auav.com.au/articles/what-is-a-drone-used-for/.

[16] Ayanava Sarkar, Ketul Arvindbhai Patel, R.K. Ganesh Ram, and Geet Krishna Capoor. 2016. Gesture control of drone using a motion controller. In *2016 International Conference on Industrial Informatics and Computer Systems*. 1–5. https://doi.org/10.1109/ICCSII.2016.7462401

[17] Alessia Vacca and Hiroko Onishi. 2017. Drones: military weapons, surveillance or mapping tools for environmental monitoring? The need for legal framework is required. *Transportation Research Procedia* 25 (2017), 51–62. https://doi.org/10.1016/j.trpro.2017.05.209

[18] B. Vergouw, H. Nagel, G. Bondt, and B. Custers. 2016. Drone Technology: Types, Payloads, Applications, Frequency Spectrum Issues and Future Developments. *The Future of Drone Use* (2016), 21–45. https://doi.org/10.1007/978-94-6265-132-6_2

[19] Wright and Keith. 2014. Wearable Technology: If the Tech Fits, Wear It. *Journal of Electronic Resources in Medical Lbraries* 11, 4 (2014), 204–216. https://doi.org/10.1080/15424065.2014.969051

[20] Zhenfei Zhao, Hao Luo, Guang-Hua Song, Zhou Chen, Zhe-Ming Lu, and Xiaofeng Wu. 2018. Web-based interactive drone control using hand gesture. *Review of Scientific Instruments* 89, 1 (2018). https://doi.org/10.1063/1.5004004