Spring 2014 Haptics Class Project Paper presented at the
University of South Florida, April 30, 2014

Kinect Control of a Quadrotor UAV

Mason Chilmonczyk[1]

***Abstract*— The work presented here covers the gesture based control of a quadroter helicopter and compares its advantages and disadvantages to traditional control methods. The particular focus of this study was to create a control system that was intuitive to even the most novice users, unlike many dual analog stick based remotes that are difficult for users to master. Microsoft's Kinect camera for the XBOX gaming platform transmitted the movements of users to a JRC 3D 4CH RC Sky Walker 2.4GHz Quadcopter.**

***Keywords*— gesture-control, quadcopter, Microsoft Kinect, UAV, user interfaces**

I. INTRODUCTION

Humans have a tendency to naturally use their body to communicate in a variety of ways, and when controlling something remotely it is common to see body movement alongside the fine tuned thumb movements associated with a classic analog joystick based remote. In recent years in mainstream electronics and entertainment, such as the XBOX Kinect, entire games can be played based off of the user's gestures and body motion.

The reason research is growing in the field of gesture based interfaces is because a human inherently knows how to communicate with the body. It has been shown that children often "speak" with their bodies before they learn to actually communicate in any verbal form [6]. This form of communication is close to a universal language which makes controls based off of these same motions more intuitive to inexperienced users, especially as technology progresses so quickly that many cannot keep up with the advances.

Early stages of research in this area utilized gloves with microcontrollers, and many times wires to the cameras, in order to communicate the gestures between the gloves and the cameras [6]. However, the XBOX Kinect is able to detect a variety of parameters just based off of pattern recognition (i.e. all humans have a similar shape). The modern advances in these technologies represents a chance to implement them into new fields that were previously unheard of.

The real world application of a gesture based UAV goes beyond just a play thing. For instance, many military and search and rescue operations require unmanned vehicles, whether they be by air or land, and it is possible that gesture based controls can create very simplistic controls for volunteers or new members of these task forces. Another opportunity would be freeing up the operator to do other things, for instance a user could verbally prompt the gesture based control system to take new orders, and then carry on doing more advanced tasks that require human interaction. Another possible field for these body motion controls is the medical industry. With more accurate and with proper safety features an operator could control new devices to conduct surgery or other medical procedures based on the movements of their own body. In this study the focus was not to have this level of automation an accuracy, but to create a closed feedback loop between the user, the kinect, the computer, the controller (Arduino Uno) and the UAV.

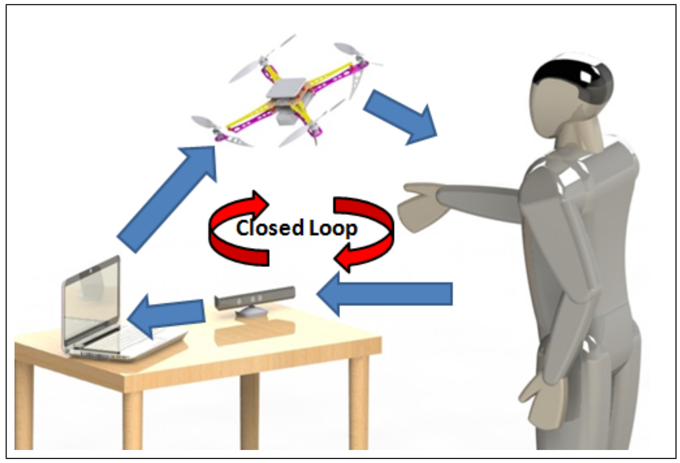

Fig. 1: Diagram demonstrating closed feedback nature of project. User analyzes visual cues from quadrotor to create a gesture to adjust quadrotor that the kinect reads and transmits to the computer which then maps the gesture to a number (see control specifications section) which is sent to the quadroter via Arduino (not pictured).

Fig. 1 represents what the closed feedback loop in this project is comprised of. The Kinect takes gesture based information from the human, then feeds signals to the computer which then sends commands to the UAV via an RF chip on an Arduino (not pictured). The feedback loop is adjusted based off of visual feedback between the user and the UAV.

II. PREVIOUS WORK

Numerous studies have been done involving the control of a quadcopter using a Kinect [1][2] which makes the process of actually controlling this particular vehicle less complicated. What is unique about this study is the comparison of gesture control to traditional control, and to see how well an effective replacement can be designed.

Previous work has shown that the Kinect's depth camera is capable of controlling pitch (see fig. 3 for explanation of pitch, roll, yaw) by moving toward and away from the screen [2]. The dynamic nature of flight requires quick

*This work was not supported by any organization

[1]Mason Chilmonczyk is an undergraduate from the Department of Mechanical Engineering at the University of South Florida, Tampa, FL chilmonczyk@mail.usf.edu

updates and fast response, and this was also proven to be handled with ease due to the Kinect's high frame rate and the quadrotor's built in stabilization gyro. The studies referenced in this paper however focus mostly on altitude or pitch control individually; this study aims to control altitude, pitch and roll simultaneously. For this particular quadcopter it was unnecessary to control yaw because the variable required no adjustment during flight, where as the other three required constant updating in order to avoid a failed flight.

III. XBOX Kinect Specifications

The Microsoft Kinect (Fig. 2) is an addendum to the traditional XBOX gaming system with initial release in November 2010. The camera was initially used exclusively for gaming but quickly became popular in developer kits after it's release.

The Kinect is equipped with an Red Green Blue (RGB) camera with 8 bit resolution (0-255), a depth camera with 11 bit resolution (0-2047) and an infrared projector allowing the depth camera to function even in low light situations with 8 bit resolution (0-255) [1][2]. The work presented here focuses on using the infrared projector with a constant depth of operation. This study focused on three axis control: to control "roll" with the left hand moving horizontally, controlling "pitch" by moving the left hand vertically, and updating throttle with the right hand moving vertically. Fig. 5 shows the Kinect interface with control boundaries (see section: Control Specifications and User Interface). With careful calibration yaw would remain constant so the quadcopter would not rotate about its vertical axis.

The Kinect was programmed using C sharp in conjunction with the software development kit version 1.8 (SDK 1.8). The code was modified from a previous year's project which required the mapping of body parts, but required no communication with arduino.

IV. Quadcopter UAV Specifications

Fig. 3 is a picture of the Skywalker 1306 quadrotor helicopter. This is a unique version of a very popular remote controlled helicopter because it has a so called "hamster cage" around it that functions as a protective shield for both users and the copter itself. This cage was designed so the device could be used as a ground vehicle, and most uniquely a wall vehicle, with the ability to fly on walls as if the gravity were modified. In the scope of this project the cage represented a nice way to protect innocent bystanders during testing and flight, and also offered protection of the UAV in the case of an accident. The control system is equipped with emergency shutoff protocols that are intended to scale down the motors for a somewhat smooth landing, but this is not equivalent to a safe landing, so the protective cage allowed for a more aggressive shutdown cycle with no harm.

The quadcopter is equipped with a 2.4 gHz receiver and has a 3.7v 300 mAh battery for approximately 7 minutes of

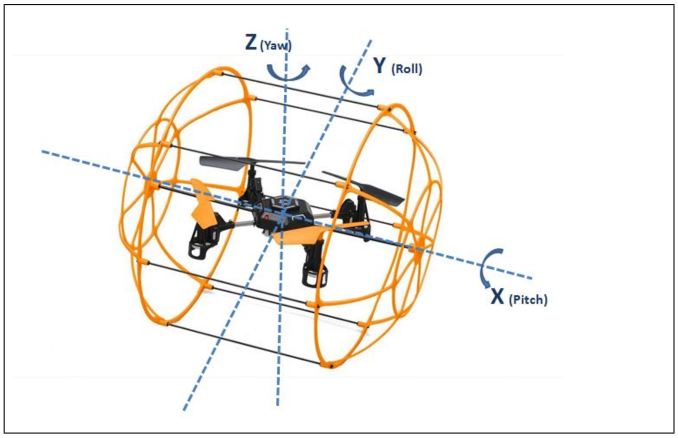

Fig. 3: Front view of the Skywalker 2.4GHz Quadcopter with the protective cage with representative X,Y, and Z axes. Pitch is rotation about the horizontal or Y axis, making the quadcopter go forward and back in the XZ plane. Yaw is about the Z axis, and makes the copter turn in the XY plane. Roll is about the X axis and moves the quadcopter left to right in the ZY plane

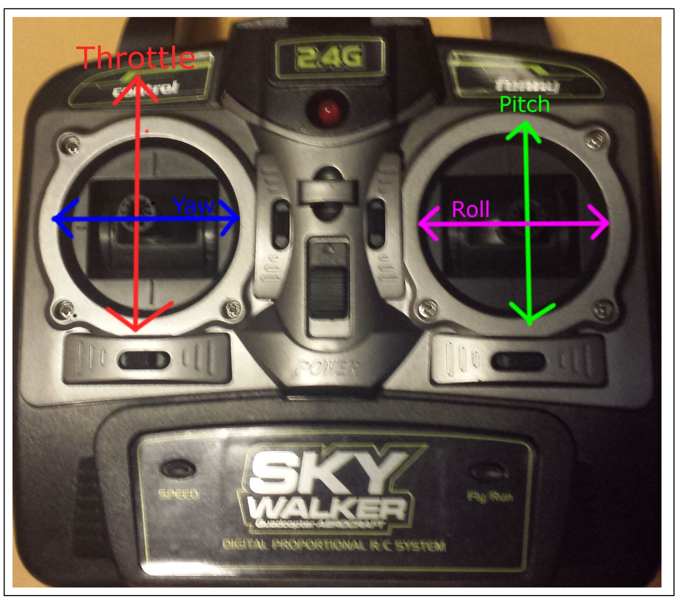

Fig. 4: Quadcopter standard remote with corresponding control protocol. The left analog stick handles throttle and yaw, while the right analog stick handles pitch and roll.

Fig. 2: Front view of an XBOX Kinect and its three sensors. From left to right, the IR projector, the RGB camera, and the monochrome camera (used for depth imaging) [1][2]

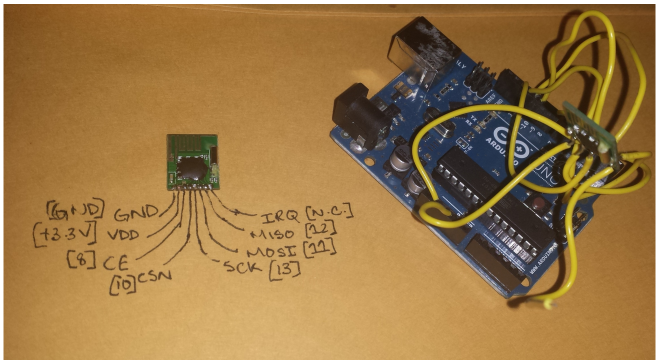

Fig. 5: Arduino Uno with RF chip connected. Diagram to the left shows pin layout for RF chip and corresponding Arduino connection written with [brackets] around it.

continuous flight. On board the helicopter is a 6 axis gyro that stabilizes the copter mid fight to account for fluctuations in balance during flight, but the technology is not advanced enough to let the quadcopter hover in place which is why pitch and roll had to be updated via Kinect to keep the quadcopter from losing balance. The copter itself is 24 x 21.5 x 21.5 cm, making it a quite compact package.

The remote control for the device is shown in fig. 4. The controller is similar to most, allowing the user to adjust roll and pitch together while keeping yaw and throttle separate. The fact that yaw and throttle are coupled on the left hand (generally a less dominate hand) indicates that pitch and roll require more fine muscle control in order to keep them steady. In the Kinect based version the handedness of the controls was switched (left hand controls pitch and roll, right hand controls throttle) to try and create a completely gesture based control that would be intuitive, so handedness would not come into play.

V. Arduino and RF Transceiver Setup

The Arduino Uno was the microcontroller used in this project. The HopeRF RFM73 2.4GHz Nordic Nrf24L01+ RFIC is the RF transceiver used to communicate with the quadcopter. The chip is designed to operate at 3.3V logic but will handle the 5V logic of the Arduino, allowing for a direct connection between the two devices. One serious disadvantage of this RF chip is the antenna's range; being very short and fabricated from a standard wire, the quadcopter easily left the zone of communication. In addition, the soldering was done in house and therefore was not professional, leading to increased resistance and decreased performance. Fig. 5 shows the physical setup and wiring schematic of the hardware. The pin connections are as follows: RF GND to Arduino GND, RF VDD to Arduino +3.3V, RF CE to Arduino pin 8, RF CSN to Arduino pin 10, RF IRQ no connect, RF MISO to Arduino pin 12, RF MOSI to Arduino 11, and RF SCK to Arduino pin 13.

VI. Control Specifications and User Interface

The control was comprised of three essential components: 1) Acer Aspire V laptop computer 2) Arduino Uno 3)HopeRF RFM73 2.4GHz Nordic Nrf24L01+ RFIC, which is the transceiver that replaced the remote. The programming and control was inspired through an Internet blog which used a different RF chip and communicated directly with the Arduino through the analog ports [8].

TABLE I
8 Byte Control Package Contents Sent To Quadcopter [5]

Byte 0	Throttle	0-255
Byte 1	Yaw	0-255
Byte 2	Yaw Trim	0-128
Byte 3	Pitch	0-255
Byte 4	Roll	0-255
Byte 5	Pitch Trim	0-128
Byte 6	Roll Trim	0-128
Byte 7	Fly/Run Toggle	0=Fly 16=Run

The Arduino was programmed using the open source coding software available through the Arduino website with libraries designed for binding to remote devices operating in the 2.4 gHz range. Flying data is sent to the UAV in its normal operation in an 8 byte package. Table 1 represents the package information. In the spectrum of this project the only bytes that needed to be sent were byte 0 (throttle), byte 3 (pitch) and byte 4 (roll). All other bytes were set so they had no effect on the flight path.

The control of the quadcopter was done as follows. Step 1: calculate the position of the left and right hands in terms of the pixel location on the Kinect screen. Step 2: map the horizontal position of the left hand from its calculated pixel location to 0-255 for roll, the vertical position of the left hand to 0-255 for pitch, and the vertical position of the right hand to 0-255 for throttle. In the case of throttle, the

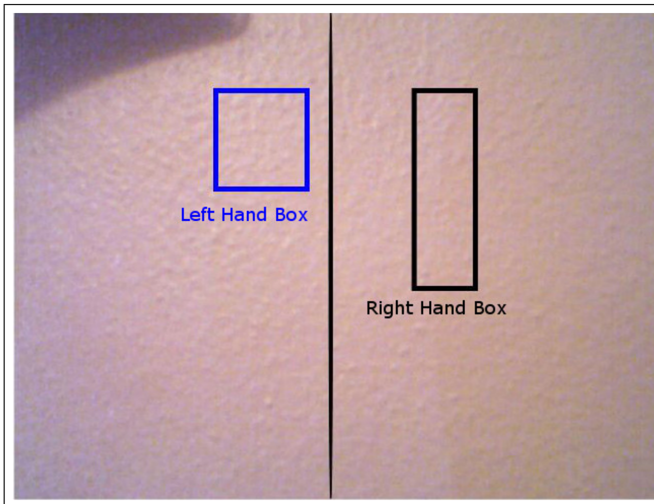

Fig. 6: Screen capture of the Kinect camera interface. The right hand controls throttle with the top of the box being full on, the left hand controls pitch with vertical motion and roll with horizontal motion. For no adjustment on roll or pitch the user centered their hand in the blue box. The vertical black line represents the center line of the screen.

actual window was smaller, approximately 0-200, to avoid full on throttle which is not necessary for indoor flight. In the case of roll and pitch, the actual window was 40-205. For roll 0 represents fully rolled left, and 255 is fully rolled right, so the window was closed on both sides to avoid drastic over compensation. Pitch is similar to roll in that 0 represents pitching back and 255 is pitching all the way forward. For both pitch and roll, the default value was 128 which correlated to no adjustment.

One serious disadvantage of the Kinect version of the control software was the time lag for commands. In the case of the standard remote there was nearly no delay detected and it is claimed the delay is less than 20 ms, but for the Kinect there was a serious delay on the order of on average 150 ms. This number was determined by creating a test program to send a shutoff command and see how long the timer ran until the quadcopter shut down. This delay is due in part to a number of issues: the Arduino is a low power module and the RF transceiver was soldered by hand. These two issues meant that there was lost power between the command and the RF signal. The slow baud rate (19200) also meant that signals could not be sent at the max clocking speed of the processing programs. The hand held remote however has very little microprocessing going on and has a larger power source (8 double A batteries) than the Kinect version. Future work might address some of these problems to alleviate the control lag.

VII. Experimental

For this initial "feeler" study the experiment was qualitative with a relatively small sample size. The goal was to compare the different flight methods. Every participant was at least allowed to fly the quadcopter with the Kinect. In most (13 of the 16 total) cases participants were asked to fly the quadcopter with both the Kinect and the classic remote. During flight a fishing line was attached to keep the quadcopter in some level of physical limit which allowed participants to focus on just balancing and maneuvering the quadcopter, rather than trying to also avoid obstacles. After the trial was completed each participant filled out a survey with four questions. Questions 1 and 2 asked participants to rate the intuitive nature of both controls (or of just the Kinect if applicable) from 1-5 with 1 being non-intuitive. Question 3 asked (yes/no) if the participants had ever flown any type of toy like this before including a helicopter, air plane, etc. Question 4 asked for an input as to how to make the project better with an open ended response.

VIII. Results

Fig. 7 shows the responses of the participants to the remote control. In general, there were few people who responded that the remote control was perfectly intuitive, and those who responded with higher ratings of 4 and 5 also had flown a quadcopter in the past, indicating that it is possible that they found it to be intuitive because they were familiar with the device. Fig. 8 shows the responses to the Kinect based control. Opposite of the hypothesis, this does not indicate any significant difference in control intuitiveness, and since the sample sizes were different (only 13 for both the remote control and Kinect control, while 3 used only the Kinect) it was hard to compare the slight differences.

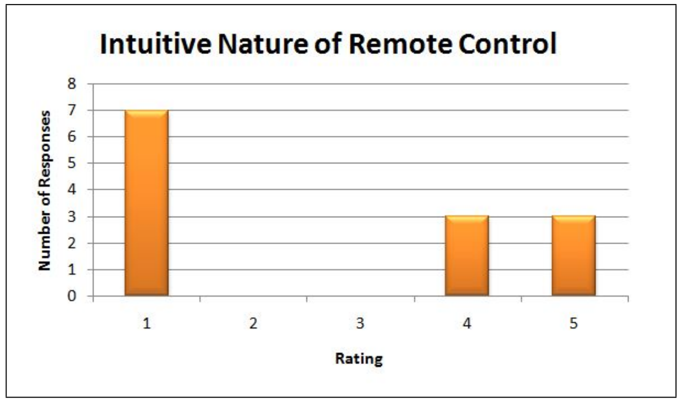

Fig. 7: Intuitive nature of remote control with 1 being nonintuitive and 5 being easy to handle. The total is only 13 because 3 of the participants did not get a chance to use the classic remote.

The trend for the suggested changes (question 4 of the survey) was overwhelmingly pointed towards one factor: the Kinect display. Of those that left comments, 9 of them commented that the boxes in which the hands were placed should be larger to allow for less abrupt control. Other comments included limiting the rotation of the quadcopter (this was attempted, but clearly not evident), as well as changing feedback methods because it was difficult to pay attention to both the quadcopter and the Kinect at the same time.

The estimated response time from 13 of the 16 total participants was in the 50 to 500 ms time, indicating a serious lag in responsiveness.

IX. Discussion

The Kinect based control was completely functional, which was one major goal of this study, but the results

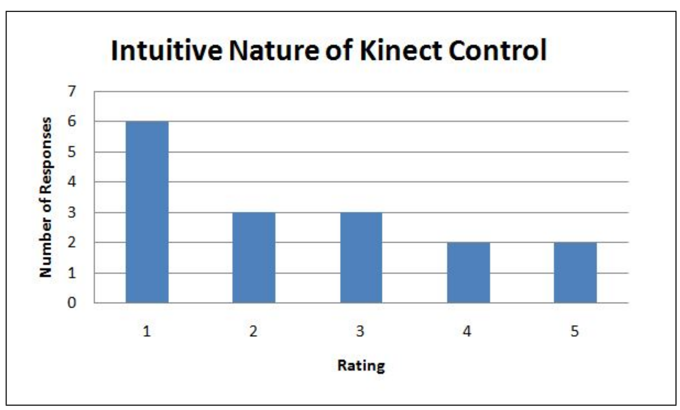

Fig. 8: Intuitive nature of Kinect based gesture control with 1 being nonintuitive and 5 being easy to handle. The total here is 16, indicating the total number of participants in the study.

did not clearly suggest that this new control method was more effective than the classic remote. However, it is hard to determine whether this is a function of the control itself, or of how the control was presented due to the flaws as discussed by the participants. The most obvious issue with the Kinect based control is the response lag. With the classic remote there is a small enough lag that participants did not seem to mind, but when using the Kinect participants commented that the lag was noticeable. By reducing the lag time the control might be significantly better because then the two versions of control (gesture based and remote based) would at least be on the same level of responsiveness, at which point the features of each control could be accurately compared.

Another complaint was the feedback method, or as it was put how difficult it was to keep the user's hands inside the alloted space for control. By expanding the boxes there was a serious downside of the Kinect not responding on the outer limits of the screen, but by keeping the boxes small the user could not tell where the hands were in relation to where they should be. One way to improve this would be to implement a relative position control for pitch, yaw, and roll. Throttle was maintained easily, but if a new control type could be developed which compared where the user's hand was at some moment in time to where it was a moment ago then the command could be sent to adjust in an absolute manner rather than in a limited manner which is what happens with the boxes on the display screen.

One interesting feature of this study is that users did not find the remote control to be necessarily more or less effective than the Kinect based control, which indicates that flying this quadcopter is just difficult in general. This is likely due to the fact that in most people find controlling any flying object to be somewhat challenging, flight has for a long time puzzled the human race. However, this could also be attributed to the quadcopter lacking advanced gyro technologies which automatically balance it, forcing users to have to finely tune the balance prior to actually carrying out any flight maneuvers.

Initially it was proposed that the handedness should not come into play with control, but during trials it became evident that participants were more likely to use their dominant hand to adjust pitch and roll while their nondominant hand stayed put. In the one trial which was left handed it seemed to be more natural, but this is a qualitative assessment and not a something which was investigated further. Future controls may base the angle adjustments around the dominant hand and leave the throttle to the nondominant hand because it generally requires less adjustment.

X. Future Work

This project lends itself to numerous adjustments before true conclusions can be gathered. New hardware with high baud rates and a quadcopter with effective gyro balance can reduce response lag and fine tuned adjustments. Additional feedback methods such as vibrotactile or improved visual through larger displays could improve the issue of hand placement, and increasing the control box size on the Kinect display, or by implementing relative position control for the inputs the focus could be on controlling the quadcopter rather than placing hands. Another option would be to set up physical barriers so the hands could only move so far, but this takes away from the advantage of being able to use this device with only a body and brings in additional hardware which takes away from the freedom of gesture based controls. After the control lag and the feedback methods are addressed more quantitative studies could take place that include obstacle course time trials to determine which control performs better in a real world task.

Acknowledgment

The author would like to thank Ahmad Manashrah for his patient assistance with the Kinect programming, and for the advice for smoothing out the communication. In addition, special thanks to Dr. Kyle Reed for his assistance and guidance along the way, and to my roommates for dealing with the testing phase of this project, at which point it did not fly but only crashed.

References

[1] Stowers, John, Michael Hayes, and Andrew Bainbridge-Smith. ”Altitude control of a quadrotor helicopter using depth map from Microsoft Kinect sensor.” Mechatronics (ICM), 2011 IEEE International Conference on. IEEE, 2011.

[2] Boudjit, K., C. Larbes, and M. Alouache. ”Control of Flight Operation of a Quad rotor AR. Drone Using Depth Map from Microsoft Kinect Sensor.”

[3] H. G. Malabet, R. A. Robles, and K. B. Reed, ”Symmetric Motions for Bimanual Rehabilitation,” Proc. of IEEE Intl. Conf. on Intelligent Robots and Systems (IROS), Taipei, Taiwan, October, 2010.

[4] Wachs, Juan Pablo, et al. ”Vision-based hand-gesture applications.” Communications of the ACM 54.2 (2011): 60-71.

[5] S. McAmis and K. B. Reed. ”Simultaneous Perception of Forces and Motions Using Bimanual Interactions,” IEEE Transactions on Haptics, Special Issue on Haptic Human-Robot Interaction, Vol. 5, Num. 3, pp. 220-230, 2012.

[6] Bhuiyan, Moniruzzaman, and Rich Picking. ”Gesture-controlled user interfaces, what have we done and what's next.” 5th Collaborative Research Symposium on Security, E-Learning, Internet and Networking. 2009.

[7] J. R. Cooper, M. M. Wernke, and K. B. Reed ”The Effects of Incongruent Feedback on Bimanual Task Performance”, Proc. of the Haptics Symposium, Vancouver, Canada, 2012.

[8] Author Unknown, ”Controlling toy quadcopter(s) with Arduino” November 12, 2013. Web. April 1, 2014 http://dzlsevilgeniuslair.blogspot.com/2013/11/more-toy-quadcopter-hacking.html

[9] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.