DEVELOPMENT OF A LOW–COST SOCIAL ROBOT FOR PERSONALIZED HUMAN–ROBOT INTERACTION

by

CHRISTIAN PUEHN

Submitted in partial fulfillment of the requirements for the degree of Master of Science

Department of Mechanical and Aerospace Engineering

CASE WESTERN RESERVE UNIVERSITY

May, 2015
CASE WESTERN RESERVE UNIVERSITY
SCHOOL OF GRADUATE STUDIES

We hereby approve the thesis of

CHRISTIAN PUEHN

candidate for the degree of Master of Science*.

Committee Chair

Kiju Lee

Committee Member

Francis Merat

Committee Member

Roger Quinn

Date of Defense

March 30, 2015

*We also certify that written approval has been obtained
for any proprietary material contained therein.
Contents

List of Tables ii
List of Figures iii
Acknowledgments iv
Abstract v

1 Introduction 1
  1.1 Research Goals ................................................. 1
  1.2 Related Works ................................................. 3

2 Philos V.1 11
  2.1 Previous Work .................................................. 11
    2.1.1 Mechanical Design ........................................ 11
    2.1.2 Human Robot Interaction .................................. 13
    2.1.3 Identified Problems ........................................ 15
  2.2 Technical Improvements ........................................ 16
    2.2.1 Hardware Updates ........................................... 16
    2.2.2 Human Robot Interaction ................................... 23
    2.2.3 Integration with Wearable Health Sensors .................. 30
    2.2.4 GUI Update .................................................. 31
3 Wearable Sensor

3.1 Tech. Description ........................................... 37
    3.1.1 Embedded Sensors and Measurable Data ............... 38

3.2 Graphical User Interface .................................. 46

3.3 Integration with Philos ................................... 47

4 Philos V.2

4.1 Goals ....................................................... 50

4.2 Mechanical Design ....................................... 52
    4.2.1 Internal Structure .................................. 52
    4.2.2 External Structure .................................. 57
    4.2.3 Electronics ......................................... 60

4.3 HRI Algorithms ........................................... 64
    4.3.1 Face Detection ....................................... 64
    4.3.2 Face Recognition ..................................... 66
    4.3.3 Speech Generation/Recognition ....................... 67
    4.3.4 Tablet Integration .................................. 68

4.4 Servo Control ............................................. 68

4.5 Preliminary Cost Analysis ................................. 71

5 Future Work

5.1 Vision Upgrades ........................................... 73

5.2 Tablet Apps ............................................... 74

5.3 Mobile Platform .......................................... 75

5.4 Behavior .................................................... 75
# List of Tables

<table>
<thead>
<tr>
<th></th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Detailed specifications for Raspberry Pi Model B computer.</td>
<td>21</td>
</tr>
<tr>
<td>2.2</td>
<td>List of audio clips stored on Philos</td>
<td>24</td>
</tr>
<tr>
<td>2.3</td>
<td>Philos V.1 actions based on user input</td>
<td>25</td>
</tr>
<tr>
<td>2.4</td>
<td>Images of Philos V.1 actions</td>
<td>26</td>
</tr>
<tr>
<td>2.5</td>
<td>Images of Philos V.1 actions continued</td>
<td>27</td>
</tr>
<tr>
<td>2.6</td>
<td>Detection time with various sizes of the searching area applied.</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>PA: Potential Area, SA: Searching Area [34].</td>
<td></td>
</tr>
<tr>
<td>2.7</td>
<td>List of actions for manual control of Philos V.1</td>
<td>32</td>
</tr>
<tr>
<td>4.1</td>
<td>Philos 2.0 Specifications.</td>
<td>52</td>
</tr>
<tr>
<td>4.2</td>
<td>Material properties of Aluminum 6061 and Makerbot 3D printed ABS.</td>
<td>54</td>
</tr>
<tr>
<td>4.3</td>
<td>Detailed AX–12A servo specifications.</td>
<td>63</td>
</tr>
<tr>
<td>4.4</td>
<td>Range of motion from Philos V.2.</td>
<td>71</td>
</tr>
<tr>
<td>4.5</td>
<td>Comparison of commercially available social robots.</td>
<td>72</td>
</tr>
<tr>
<td>5.1</td>
<td>Comparison of Raspberry Pi Models</td>
<td>77</td>
</tr>
</tbody>
</table>
List of Figures

1.1 The Huggable robot .............................................. 4
1.2 Kismet robot .................................................. 5
1.3 NeCoRo robot ................................................... 5
1.4 NAO robot ..................................................... 6
1.5 Paro robot ..................................................... 6
1.6 iCat robot ..................................................... 7
1.7 Keepon robot ................................................... 8
1.8 Darwin robot ................................................... 8
1.9 Kaspar robot ................................................... 9

2.1 CAD representation of Philos V.1 arm assembly ................. 12
2.2 CAD representation of Philos V.1 head assembly ................. 12
2.3 Robotis Dynamixel servo control ................................ 13
2.4 Philos V.1 original exterior concept art and actual cover .... 13
2.5 Original GUI for Philos V.1 platform .......................... 16
2.6 Different colored outfits of Philos V.1 ........................ 17
2.7 Philos V.1 eyebrows portraying different emotions ............ 18
2.8 FSR clusters located on head and chest of Philos V.1 ......... 18
2.9 Circuit diagram of FSR clusters in Philos V.1 ................ 20
2.10 Philos V.1 power circuit ...................................... 22
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.11</td>
<td>WTV020-SD audio module pinout</td>
<td>23</td>
</tr>
<tr>
<td>2.12</td>
<td>Control circuit for WTV020SD-20S audio breakout board from Sparkfun</td>
<td>24</td>
</tr>
<tr>
<td>2.13</td>
<td>Face detection algorithm showing detected face outlined</td>
<td>30</td>
</tr>
<tr>
<td>2.14</td>
<td>Philos V.1 motion detection algorithm</td>
<td>31</td>
</tr>
<tr>
<td>2.15</td>
<td>Updated Philos V.1 monitoring GUI</td>
<td>33</td>
</tr>
<tr>
<td>3.1</td>
<td>Wearable health sensor with and without cover</td>
<td>38</td>
</tr>
<tr>
<td>3.2</td>
<td>Circuit diagram of wearable health sensor</td>
<td>41</td>
</tr>
<tr>
<td>3.3</td>
<td>CAD representation of cover for pulse sensing module to secure it to a user's wrist</td>
<td>43</td>
</tr>
<tr>
<td>3.4</td>
<td>GUI of wearable health sensor</td>
<td>48</td>
</tr>
<tr>
<td>4.1</td>
<td>CAD representation and actual assembled Philos V.2</td>
<td>53</td>
</tr>
<tr>
<td>4.2</td>
<td>CAD representation of the body of Philos V.2</td>
<td>55</td>
</tr>
<tr>
<td>4.3</td>
<td>CAD representation of the head structure of Philos V.2</td>
<td>56</td>
</tr>
<tr>
<td>4.4</td>
<td>CAD representation of the arm structure of Philos V.2</td>
<td>56</td>
</tr>
<tr>
<td>4.5</td>
<td>CAD representation of tablet docking section of Philos V.2 shell</td>
<td>60</td>
</tr>
<tr>
<td>4.6</td>
<td>Graphical representation of Dynamixel smart servo control protocol</td>
<td>61</td>
</tr>
<tr>
<td>4.7</td>
<td>Philos V.2 power board schematic</td>
<td>64</td>
</tr>
<tr>
<td>4.8</td>
<td>Face detection algorithm showing the detected face outlined</td>
<td>65</td>
</tr>
<tr>
<td>4.9</td>
<td>Recognized face in embedded Philos V.2 algorithm with user’s name displayed above the recognized face</td>
<td>67</td>
</tr>
<tr>
<td>4.10</td>
<td>Pinout of 74LS241 H-bridge IC</td>
<td>69</td>
</tr>
<tr>
<td>4.11</td>
<td>Wiring schematic of Raspberry Pi and H-bridge IC for servo control</td>
<td>69</td>
</tr>
<tr>
<td>4.12</td>
<td>Velocity of servo with and without servo smoothing</td>
<td>70</td>
</tr>
</tbody>
</table>
Acknowledgments

I would like to thank my advisor, Dr. Kiju Lee, for her guidance, support, and for giving me the opportunity to perform research in her lab. I would also like to thank my colleagues, Sahil Kothari, Tao Liu, Yixin Feng, Jianan Sheng, and especially Donghwa Jeong for their advice and for creating an enjoyable work environment. Also, I would like to thank my parents for their support and encouragement every step of the way.
Development of a Low–Cost Social Robot for Personalized Human–Robot Interaction

Abstract

by

CHRISTIAN PUEHN

This thesis presents two social robotic platforms, Philos V.1 and Philos V.2, which are designed for personalized human–robot interaction (HRI). Personalized HRI is realized through programmable expressive (i.e. gestures and sound/speech generation) and perceptive functions (i.e. human face tracking and recognition, touch detection, and speech recognition) of the robots. In addition, a wearable sensor device was developed for integration with these social robots to further improve the health monitoring and assessment functions of the robots as well as personalizing the robots social behavior based on the user’s biobehavioral data. Among the many potential applications of a social robot, this thesis focuses on elderly healthcare where positive health outcomes are expected by providing companionship, assisting with daily activities, and monitoring health status.
Chapter 1

Introduction

1.1 Research Goals

Social robotics has been receiving growing interest for a broad scope of clinical, educational, and entertaining purposes. Social robots are designed to entertain, assist, or provide service to humans through vision, touch, and sound-based interaction. Therefore, human–robot interaction (HRI) often resembles the way humans interact with each other. Recently, the potential for social robots to serve as a long-term health care solution has been receiving increased research. A social robot can serve as a companion for older people by helping them maintain independent living [1]. In addition, recent studies have demonstrated the strong potential of social robots in behavioral training for children with developmental disabilities [2], [3], [4].

As healthcare and living standards continue to improve, the population of elderly in the United States is growing at an incredible rate. A recent survey in 2009 counted 39.6 million elderly in the United States, and that number is expected to almost double to 72.1 million persons by 2030 [5]. The cost of healthcare for this demographic is unsustainably large and will continue to grow as the population grows. Medicare costs alone are expected to balloon from $555 billion in 2011 to $903 billion in 2020 [6].
Therefore, a technological solution must be found to address this growing issue. Social robots and the integration of wearable health sensor could drastically reduce costs associated with the elderly population by automating part of the health monitoring process and reducing the need or frequency of nurse interaction and enable a more independent lifestyle. In addition to monitoring of health, social robots can act as a companion for the elderly who may not be able to care for a pet and have been shown to increase quality of life.

Similarly, the number of children diagnosed with autism spectrum disorder (ASD) or other behavioral disorder has increased dramatically in recent years. A recent report from the Centers for Disease Control and Prevention has stated that the prevalence of ASD in children aged 8 years is as high as one in 68 [7]. The medical costs for these children is very high due to the high cost of tools and therapeutic techniques used by doctors. The estimated difference in cost per year between raising a child with ASD compared with a child without ASD is at least $17,000 [8]. Social robots have been shown to be successful at interacting with children with ASD or similar behavioral disorders and have even been used to encourage children to interact with other humans easier. A low cost social robot could help increase the availability of this technology for children in need.

Focusing on the healthcare applications of social robots, the major challenges are in 1) difficulties in assessing the health outcomes, 2) relatively high cost, and 3) lack of user–friendly interface for personalizing the robot’s function and behavior based on the user’s needs. In an attempt to address these challenges, the overall goal of my thesis research is to develop a low–cost hardware platform for personalized human–robot social interaction and health monitoring that can be used in both domestic and professional environments. To accomplish this goal, the following specific aims were pursued:

- **Aim 1:** Building on the previous prototype, Philos V.1, identify technical

- **Aim 2:** Develop a wearable biosensor for integration with the Philos platform.

- **Aim 3:** Develop the hardware of Philos V.2 to serve as a hardware platform for a wide range of HRI applications.

The research presented in this paper explores the use of social robots as a low cost solution for elderly healthcare and for use in therapy for children with behavioral disorders. The use of the social robots in elderly healthcare are two–fold serving to both improve emotional well–being and incorporating a wearable sensor for health monitoring. Emotional well–being is encouraged by a combination of vision, auditory, and touch human–robot interaction that creates responses from the robot based on how the user interacts with the robot. These reactions can promote emotional and mental wellness by acting as a companion to the user. Integration with a wearable sensor also allows the robot to monitor critical vitals such as pulse and activity levels and can warn if some anomaly is detected. The implementation of software that allows non-technical users or clinicians to control and monitor the robot in real-time is beneficial to therapy for children with behavioral or developmental disorders. The social robots presented in this paper offer a low cost solution that could help to solve the growing healthcare crisis of the elderly and children with behavioral and developmental disorders.

### 1.2 Related Works

Over the past several decades, a number of socially interactive robots have been developed, covering a range of design and functionality objectives. Among many, nine social robots that are highly related to the presented work are reviewed in detail.
Firstly, the Huggable, a robot with the outer appearance of a teddy bear, focuses on implementing a sophisticated touch-sensitive skin, allowing for therapeutic interaction through physical touch inputs and responses [9]. The Huggable uses a complex combination of pressure, temperature, and electric field sensors to achieve a novel system to distinguish touch with a human from touch with an object. By distinguishing between human and inanimate object touches, the Huggable aims to assist in therapeutic care by acting in a similar aspect as a therapy dog might by nuzzling or cuddling with a human. The Huggable robot can be seen in Figure: 1.1.

![Figure 1.1: Huggable robot [9]](image)

Secondly, Kismet is an anthropomorphic head with 21 degrees of freedom that emphasizes producing complex facial expressions in response to user inputs [10]. Kismet uses the large number of degrees of freedom that are available to create a deeper level of interaction with people. Kismet feature fifteen degrees of freedom in the facial features: the ears, eyebrows, lips, mouth, and eyelids [11]. Using these degrees of freedom Kismet is capable of displaying nine different emotions through generating different facial expressions: happiness, sadness, surprise, anger, calm, displeasure, fear, interest, and boredom. The Kismet robot is shown in Figure: 1.2.

![Figure 1.2: Kismet robot](image)

Thirdly, NeCoRo is another social robot designed to interact with older people [13]. NeCoRo is a social robot designed to resemble a cat. Traditionally, pet therapy has been utilized for nursing home residents that suffer from dementia. A study of
1.2. RELATED WORKS

Figure 1.2: Kismet robot courtesy of [12].

NeCoRo interacting with residents suffering from dementia in a nursing home environment showed that residents who interacted with the robot had decreased agitation and showed increased interest. An example of NeCoRo being used to facilitate communication between two people can be seen in Figure: 1.3.

Figure 1.3: NeCoRo robot [13].

NAO is a commercial robotic platform, shown in Figure: 1.4, that is often employed for various research and education applications. For example, some recent studies employed NAO in social training for children with autism spectrum disorders. NAO has a combination of lights, vocal cues, and motions to interact with children. While simple behaviors are autonomously generated, more complex motions were controlled by researchers monitoring the process. In addition, NAO has the capability to record video and touches on its head that allows it to collect information based on the session with a child. By utilizing an intuitive GUI, NAO allows clinicians to interact wirelessly with children that suffer from ASD and has shown success in clinical studies [14].

Paro, a social robot designed to provide companionship to older people shown in Figure: 1.5, is designed as a seal-like robot mainly due to that interaction with
1.2. RELATED WORKS

CHAPTER 1. INTRODUCTION

animals has been shown to be emotionally beneficial to people [16], [17]. Paro is a social robot that is designed to look like a seal and boasts a number of touch sensors across its body that enable it to detect when a user is petting it or touching it in some manner. Paro was tested in an elderly care facility and it was shown interaction with the robot improved the mood of residents and many of them looked forward to interacting with the robot.

Another commercially available platform, iCat, can recognize objects and faces, recognize speech and sound, and generate various facial expressions [18]. iCat achieves
various facial expressions through the use of thirteen servos to actuate the lips, eyes, and eyebrows. Similar to Paro, iCat was also tested for its potential benefits to the elderly population showing that older people are more comfortable and more expressive with a more sociable robot than with a less social one. In the design of iCat, all emotion expressions are enabled through facial movement and voice generation. However, one existing problem is that while interacting with iCat, there will be no direct body contact between iCat and the human user, which may limit the range and type of interaction. An experimental setup with iCat is shown in Figure: 1.6.

Keepon is a small snowman-like robot that interacts with humans in a novel way [20], [21]. Keepon interacts with humans rhythmically by dancing. This dancing motion is controlled by software that determines the dominant rhythm of the surrounding environment and converts it into a dance for the robot to perform. Preliminary testing with children has shown that the dancing of the robot did in fact have some effect on the children’s behavior and has shown some promise with therapy for children with behavioral disorders. Figure: 1.7 shows the Keepon robot.

DARwIn-OP is an open humanoid platform that is commercially available from Robotis. DARwIn is a humanoid robot with 20 degrees of freedom and an onboard camera. DARwIn is capable of performing complex motions like walking, kicking a ball, and moving from lying down to standing up. The robot is also able to track
objects such as a ball with the embedded camera. DARwIn is highly customizable as all source code and design files are available for free. DARwIn-OP is available for purchase at the price of $12,000 and can be seen in Figure: 1.8.

Lastly, Kaspar is a child–sized humanoid robot that has been used with children with ASD. Kaspar has been utilized by researchers as a tool to promote cooperative play between children with autism and another person. Kaspar is capable of autonomously playing a collaborative game as described in Wainer, et al. [23]. In this experiment, Kaspar would encourage a child to play by following their lead, initiating a move and asking for assistance if the child did not react, and praising the child when they selected a shape. Communication with the children is achieved through
gameplay as well as through buttons that the child can press. Although the study was limited in size, the children found play with Kaspar to be fun and enjoyable and were more likely to initiate moves then in play sessions with another human. **Figure:** 1.9 shows the Kaspar robot.

![Kaspar robot](image)

**Figure 1.9:** Kaspar robot [23].

Face detection and recognition are also potentially very important in the development of social robots. Face detection and recognition allow for a deeper, more immersive, level of interaction between user and robot when the robot is able to react to a user’s movements. The ability to follow a user, recognize someone that has previously interacted with the robot, or react to gestures and motions that the user might make to the robot creates a more meaningful interaction with the user. Currently, facial recognition and detection has been shown to be very successful using a variety of methods such as Principal Component Analysis (PCA) and Local Binary Patterns (LBP) [24], [25]. Face detection has also been realized through the use of skin color to locate a face shaped region [26]. Combinations of these methods have also been shown to be practical, often with a skin color filter applied first to narrow the region that is searched for a potential face [27]. Neural network algorithms have shown promise with face detection as well as gesture recognition in large part to their
ability to adapt and learn [28], [29].

Studies have shown that two important factors in capturing a user’s interest in a social robot are facial expressiveness and movement of the robot that shows attention is being paid to the user, such as through a user tracking algorithm [30]. In addition, work has been done on robots that adapt to a user over time [31]. A user might exhaust all interactions with a robot, but if the robot can change its personality or behavior it will be more interactive for a user and will capture their interest for a longer period of time.
Chapter 2

Philos V.1

2.1 Previous Work

2.1.1 Mechanical Design

The original design of Philos V.1 can be separated into two major sections: the base of the robot, and the upper body [32]. The robot weighs approximately 2.9 kg and measures roughly 20 × 24 cm at the base and stands about 40 cm tall. The base is composed of a flat plate of 1/2 inch 6061 aluminum with a 1½ inch square tube of 6063 aluminum that is 1/s” thick and four inches long. Inside the square tube are two linear actuators; the motherboard and power board are located on standoffs on the base of the robot.

The upper body is comprised of two 1/4” 6061 aluminum plates. One servo for each arm is mounted on the lower of the two plates. Connected to this servo is another servo that allows rotation in a separate direction. The arms are connected to this second servo and are comprised of a 1/2” x 1/2” 6063 aluminum u-channel and is 4½” long. One end is connected directly to the servo horn and the other end is connected to the hand of Philos V.1 which is a small 6/6 nylon that rotates about a clevis pin with attached torsion spring. This allows for a more realistic give to the hand of Philos V.1.
Philos V.1. The orientation of the servos give the arms two degrees of freedom by allowing for rotation about both the Z and X axes.

![Figure 2.1: CAD drawing of arm assembly [32]](image)

Attached to the top plate of the upper body are two servos that are connected to each other to allow panning and tilting of the head. The first servo is connected by a mounting bracket that is provided by the manufacturer and tilts the head of the robot, while the second servo is mounted on top of the servo horn of the first servo and connects to a small plate that the cameras and the top of the head are attached to. This setup gives the head two degrees of freedom by rotating about the X and Y axes.

![Figure 2.2: CAD drawing of head assembly [32]](image)

The servos used are the Robotis Dynamixel AX-12A, which were chosen because
2.1. PREVIOUS WORK

2.1.2 Human Robot Interaction

Face tracking was enabled by capturing images using the webcams that are installed in the head of the robot in the place of eyes and processing the images using a BeagleBoard–xM with an ARM Cortex–A8 core. This face recognition and detection
algorithm, developed by Yan Zhang, utilizes Haar features with an adaptive boosting algorithm as well as a skin color filter. Once a face is detected, coordinates are sent to the mbed microcontroller and depending on the tracking parameter the robot may or may not decide to track the face. The head is panned in small increments towards the direction of the human face due to some significant delay because of hardware limitations on an embedded system.

The exterior design of Philos V.1 also included a few force sensitive resistors (FSRs) to enable touch-based interaction with a user. There were four FSRs located on Philos: one on each hand, one on the belly, and one on the top of the head. The magnitude of the touch could then be compared with experimentally obtained threshold values to determine the relative pressure of the touch and whether it corresponded to a positive or negative input from the user. As the amount of force applied to the FSR increases, the resistance of the FSR decreases. This has the unfortunate side effect of increasing the current across the FSR as the voltage stays constant, therefore precautions must be taken. The digital pins on the mbed have a current draw limit of 50mA so current limiting resistor is added in series to ensure that more current is not drawn. The resistor value can be calculated by Ohms Law: $V = I \times R$. Where the voltage is 5 V and $I = 50$ mA. Solving this yields a current limiting resistor of 10 kΩ which also serves as a pulldown resistor so that values at low levels of pressure are more accurate and less susceptible to noise.

Embedded in the robot were ten distinct behaviors: Raise hand, Withdraw hand, Nod head, Pan head, Wave, Wave arms, Roll arms, Lift arms, Lower Arms, and Bounce. These behaviors and the frequency with which they presented themselves were determined based on the personality of the robot [32].
2.1.3 Identified Problems

There were a few problems identified with Philos V.1. One of the major problems was the overall appeal of the robot. The costume that was created for Philos was ill fitting and did not match the original concept art as seen in Figure: 2.4. Another major problem that was noted by collaborators with the Cleveland Clinic Center for Autism was the insufficient number of FSRs and the difficulty of locating them when Philos V.1 had its cover on. Due to there only being one FSR on both the chest and the top of the head and no discernible marking to show where they were located it was difficult for new users to activate an FSR and interact with Philos V.1 via touch. Thirdly, it became apparent that another level of human robot interaction could be achieved by adding some form of sound generation or playback.

There were also consistency issues with the Philos V.1 platform. The Philos platform had crashing issues where the robot would stop working and would reset on average within 30 to 45 minutes of being turned on. There were also issues with the wiring of Philos V.1 that required a full rewiring of the robot because when the arms or head of the robot moved some of the wires would break and there were some shorts being caused by the wires. These issues were magnified by the challenges of performing maintenance on the internals of Philos. In order to make modifications or perform maintenance to Philos V.1, it is necessary to remove the outer clothing and then the hard plastic shell, which takes a significant amount of time.

Another potential room for improvement in the design of Philos V.1 was for there to be some method for a clinician or observer to manually control the robot rather than for it to follow its autonomous behavior only. Manual control would allow for an observer or clinician to use servo motions or behaviors from the robot to try to prompt a reaction from the person interacting with the robot. Another topic for improvement from our collaboration with researchers at the Cleveland Clinic was the expressiveness of the face. Lastly, the Graphical User Interface (GUI) of Philos
V.1 did not have many monitoring capabilities that were desired such as real time FSR information. The GUI (as shown in Figure: 2.5) is capable of tweaking the parameters of the adaptive behavioral algorithm but real–time monitoring of a user’s interaction with Philos is not possible with this GUI.

![Original GUI for Philos V.1 platform](image)

Figure 2.5: Original GUI for Philos V.1 platform [32].

## 2.2 Technical Improvements

### 2.2.1 Hardware Updates

**Exterior Changes**

In order to increase the appeal of Philos V.1, a new outer fabric cover was designed. An animal inspired design was chosen in order to make the robot resemble a stuffed animal or children’s toy. With the help of Elizabeth Mrugacz, a student of the theater department at Case Western Reserve University, a penguin costume was created with a variety of accessories to allow customization of Philos’ appearance. Accessories included bowties, hats, and vests that allowed a user to dress up Philos in their favorite color or look. Such customizations allow each individual user to change Philos’ appearance to their taste which can help make the robot feel more personal.
2.2. TECHNICAL IMPROVEMENTS

Figure 2.6: Philos V.1 in three different colors of vest and bowtie accessories that enable customization based on a user’s preference.

to the user and can make them feel more comfortable with the robot. Examples of Philos V.1 in some of the different colors available are shown in Figure: 2.6

A penguin was chosen as the target animal to emulate due to the body shape and actions of Philos V.1 and in order to keep the current hardware structures and minimize the amount of redesigning necessary. Philos V.1 has long, thin arms and a thick body which resembles the skeletal structure of penguins. Also, Philos V.1 interacts with users through motions such as waving, flapping its arms up and down, and nodding its head. These behaviors mimic actions that one might expect to see a penguin perform.

In order to add more expressiveness to the face of the robot, eyebrows were added. Two simple servo motors are mounted above the eyes and small holes were drilled for the servo shaft to stick out of the plastic. Fabric was sewn around a four point horn that is then attached to the shaft allowing for actuation of the eyebrow. This servo allows for the robot to move it’s eyebrows to display feelings of happiness or sadness depending on the desired emotional state. Three emotional states are expressed with the eyebrows: happy, angry, and neutral. These three emotions were chosen as they can easily conveyed and are on very different emotions which means that they are distinct in appearance. The eyebrow positions and corresponding emotion can be seen in Figure: 2.7.
2.2. TECHNICAL IMPROVEMENTS

Figure 2.7: Philos V.1 eyebrows in different positions to display a) happy, b) neutral, and c) angry emotions.

Figure 2.8: FSR clusters located on head and chest of Philos V.1

Touch sensing

After preliminary evaluation by clinical collaborators it was found that Philos’ touch sensors were difficult to find and interact with (especially the FSRs located on the head and chest of the robot where one sensor was responsible for the entire body section). In order to rectify this problem, clusters of FSRs were implemented on the head and chest. Five FSRs were used on the head and five FSRs comprised the chest cluster, the location of these cluster can be seen in Figure: 2.8. These FSR clusters were created by wiring FSRs in parallel so that the overall resistance followed:

\[
\frac{1}{R_{total}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \ldots 
\]  

(2.1)
Since all resistors should be equivalent we can simplify:

\[
\frac{1}{R_{\text{total}}} = \frac{1}{R} \times N
\]  

(2.2)

A single FSR when under heavy load has a minimum resistance of roughly 250 Ohms and increases as less load is applied. Due to the inverse relation of current and resistance as shown in Ohm’s Law, the current limiting resistor must be designed around the lowest possible resistance of the circuit. This would occur on an occasion when all five resistors were under heavy load, so we plug in a value of 250 Ohms for R and set N = 5. This gives the head cluster an overall minimum resistance \( R_{\text{total}} \) of 50 Ohms and the body cluster as well. In order to limit the current draw on the analog pin to less than 50 mA, a resistor of 10 Kohm must be added in series to the cluster circuit. These FSR clusters drastically increased the surface area of Philos that was covered by FSRs by approximately 300%. In addition, two new FSR regions were introduced in the feet of the exterior covering of Philos. A circuit diagram of an FSR cluster is shown in Figure: 2.9.

Initial testing shows that users unfamiliar with Philos V.1 were able to successfully trigger the FSR regions much more dependably. When instructed to touch the chest of Philos V.1, 70% of touches were successful, and for the head 50% of touches were successful (n=3 subjects performing five to ten tests each). The head sensor results are still quite low, but this is due to the design of the head rather than the surface area covered. The head shell of Philos V.1 that the FSRs are attached to is attached in the front only which has the unintended consequence that the head of the robot has significant give when pressed on the middle or rear of the head. This give or flexing of the head can cause the user’s touch of the head to simply bend the head and not register as a light touch. This can cause a new user to have difficulty when trying to touch the head of the robot until they become more familiar with how the head touch sensing works. These results show that the incorporation of FSR clusters
2.2. TECHNICAL IMPROVEMENTS

Figure 2.9: Circuit diagram of FSR clusters that were added to the chest and head of Philos V.1.

Figure 2.9: Circuit diagram of FSR clusters that were added to the chest and head of Philos V.1.

Adding FSR clusters to Philos V.1 was successful as the ability for a user to find the FSRs on the head and chest of the robot increased substantially, compared to when there was only one FSR on both the chest and head and preliminary user tests had very low chance of triggering the sensors.

Electronics

A Raspberry Pi Model B micro-computer is embedded within Philos to perform image processing onboard. The Raspberry Pi is a very popular new microcomputer developed by the Raspberry Pi Foundation that has a large population much like Arduino which makes it ideal since there are many libraries available. Unlike microprocessors like Arduino or mbed, the Raspberry Pi is a fully computer that can run Linux distribution and OpenCV and is much more powerful. This makes the Raspberry Pi ideal for integration for image processing. The Raspberry Pi communicates with the mbed
### Table 2.1: Detailed specifications for Raspberry Pi Model B computer.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Raspberry Pi Model B</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Speed</td>
<td>700 MHz</td>
</tr>
<tr>
<td>Core Architecture</td>
<td>ARM 11</td>
</tr>
<tr>
<td>GPU</td>
<td>Dual Core VideoCore IV</td>
</tr>
<tr>
<td>Memory</td>
<td>512MB SDRAM</td>
</tr>
<tr>
<td>OS</td>
<td>Linux Distribution</td>
</tr>
<tr>
<td>Power</td>
<td>USB 5V, 1.2A</td>
</tr>
<tr>
<td>Dimensions</td>
<td>85.6 x 53.98 x 17mm</td>
</tr>
<tr>
<td>USB Ports</td>
<td>2</td>
</tr>
</tbody>
</table>

A microprocessor that controls Philos V.1 via serial communication. The Raspberry Pi image processing is discussed in detail in Section 2.2.2. The Raspberry Pi Model B was chosen for image processing because it offers a powerful chipset, onboard graphics card, small size, and is low-cost. Detailed specifications are shown in Table 2.1.

The linear actuators that allowed Philos V.1 to move about the Y axis were removed. The linear actuators decreased the stability of the robot when pressure was applied on the chest of the robot and would cause it to wobble back and forth. Although this diminishes the total degrees of freedom of Philos V.1 to six degrees of freedom from the original seven, the increased stability makes the platform more appropriate for the children demographic.

Due to changes in the hardware embedded on Philos V.1, changes had to be made to the power board. The removal of linear actuators and addition of the Raspberry Pi and audio playback module changed the power requirements of the board. The new power board circuit is shown in Figure 2.10. The updated board has 12 V power supplied from a wall outlet that powers the AX12 servos. A LM7805 regulator provides a regulated 5 V source for the mbed microcontroller. This regulator is capable of providing up to 1A of current draw at 5V. However, this is not enough current to power a Raspberry Pi reliably as well so a separate 5 V line is used to power the Raspberry Pi. This power supply provides 5 V 1.2 A for the Raspberry Pi.
Figure 2.10: Schematic of updated power circuit for Philos V.1. There are two inputs for power needs from wall outlets a 12 V line and a 5 V, 1.2 A line. In addition, there is a 5 V regulator that regulates from the 12 V line to provide power for the mbed microcontroller.

from a wall outlet. This new board, in addition to rewiring of crucial components, increased the reliability of the platform, lowering the crash rate of Philos from 20% to 12% (a crash is defined as a halting of normal function within 40 minutes of turning on). In addition, the average time without crash rose to 150 minutes.

Where the schematic components are as follows:

- **J1**: Power jack for 12 V power into the board, which is split between powering the servos and the 5 V regulator.

- **LM7805**: A 5 V regulator that takes the input 12 V and regulates it as a 5 V output up to 1 A of current.

- **JP1**: Connector that supplies the regulated 5 V to the mbed microcontroller that controls Philos.
2.2. TECHNICAL IMPROVEMENTS

- **JP2, JP3, JP4, JP5:** Connectors to provide 5 V power for the various FSRs on Philos (on the left hand, right hand, head cluster, and body cluster).

- **JP6:** Connector to supply 12 V of power from the power jack to the AX-12 servos of Philos at high enough amperage to move the servos even under load.

- **J2:** Power jack for 5 V, 1.2 A line.

- **JP7:** Connector to provide the necessary 5 V, 1.2 A power that is necessary for the Raspberry Pi model B used for image processing.

### 2.2.2 Human Robot Interaction

In order to enable auditory feedback from Philos, the WTV020SD sound playback chip was used. This chip allows for playback of pre-recorded audio via serial communication with the mbed controller over a stereo that’s incorporated into Philos hardware. The WTV020SD operates in serial mode enabling full audio control with only two input wires.

![Figure 2.11: Pinout for WTV020-SD audio module courtesy of [33]](image)

The pinout of the WTV020-SD chip is shown in **Figure 2.11**. Audio playback is controlled by the mbed microcontroller through the use of the CLOCK and DATA pins, which are referred to as P04 and P05, respectively. Audio files are played over a 0.5W (8 ohm) speaker. This speaker is controlled by the audio playback breakout board using the SPK+ and SPK- pins. The speaker and audio playback circuit are located in the midsection of Philos in the area between the two plates above where
2.2. TECHNICAL IMPROVEMENTS  

<table>
<thead>
<tr>
<th>ID</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hello</td>
</tr>
<tr>
<td>2</td>
<td>Hi my name is Philos</td>
</tr>
<tr>
<td>3</td>
<td>Want to play?</td>
</tr>
<tr>
<td>4</td>
<td>Touch my belly</td>
</tr>
<tr>
<td>5</td>
<td>Pet my head</td>
</tr>
<tr>
<td>6</td>
<td>Shake my hand</td>
</tr>
</tbody>
</table>

Table 2.2: List of audio clips stored on Philos

the linear actuators used to reside. The control circuit for the WTV020SD breakout board from Sparkfun is shown in Figure: 2.12.

![Control circuit for WTV020SD-20S audio breakout board from Sparkfun.](image)

Figure 2.12: Control circuit for WTV020SD-20S audio breakout board from Sparkfun.

The audio prerecorded includes common social greetings and questions as well as penguin noises. The social greeting sound files were generated by a computer text to speech generator, then the WAV files were converted to .ad4 that is accepted by the chip and stored on a micro-SD card on the breakout board. These responses can be triggered due to an input from a user or by an administrator or clinician that is controlling the robot manually.

The control of Philos’ behavior can be specified as two different modes: fully
autonomous and manual. Philos V.1 can be switched between these two modes by pushing a button that is located on the backside of the robot. The autonomous mode of Philos V.1 is similar to the original behavior that was embedded in the robot. In autonomous mode, Philos V.1 cycles through two main protocols: responding to inputs and communicating with the GUI, and vision detection looking for users. In the first protocol, Philos V.1 reads the inputs from the FSRs and determines if there is a user touch and categorizes these touches as either “gentle” or “harsh.” The threshold values for “gentle” and “harsh” inputs are determined experimentally and can be changed based on the individual user. For instance, the thresholds may be lowered for an elderly or frail person who may have a softer touch and the thresholds can be raised in the instance of a child who is hyperactive and is more aggressive with touch accidentally. There is some significant noise from the FSRs so filtering is used to ensure that a recorded fluctuation in FSR value is in fact a user input by requiring three consecutive touches in a span of 0.2 seconds after an initial touch is detected since electrical noise fluctuations won’t be sustained long enough to record as a human touch. After determining that the input is indeed a user touching a sensor and classifying the touch as “harsh” or “gentle” an appropriate reaction is chosen for the robot to perform.

In addition to reading the input and performing an appropriate reaction, Philos V.1 transmits the FSR data via wireless communication using ZigBee protocol to a host computer running the GUI.

<table>
<thead>
<tr>
<th>Body Part</th>
<th>“Gentle” Response</th>
<th>“Harsh” Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chest</td>
<td>Flap Arms</td>
<td>Shake Head</td>
</tr>
<tr>
<td>Arm</td>
<td>Shake Hands</td>
<td>Withdraw Hand and Shake Head</td>
</tr>
<tr>
<td>Head</td>
<td>Flap Arms and Nod</td>
<td>Cower</td>
</tr>
<tr>
<td>Feet</td>
<td>Flap Arms</td>
<td>Shake Head</td>
</tr>
</tbody>
</table>

Table 2.3: Philos V.1 actions based on user input
### Action Name

<table>
<thead>
<tr>
<th>Action Name</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shake Head</td>
<td><img src="image" alt="Shake Head Image" /></td>
</tr>
<tr>
<td>Cower</td>
<td><img src="image" alt="Cower Image" /></td>
</tr>
<tr>
<td>Withdraw Hand</td>
<td><img src="image" alt="Withdraw Hand Image" /></td>
</tr>
<tr>
<td>Flap Arms</td>
<td><img src="image" alt="Flap Arms Image" /></td>
</tr>
<tr>
<td>Nod Head</td>
<td><img src="image" alt="Nod Head Image" /></td>
</tr>
</tbody>
</table>

Table 2.4: Images of Philos V.1 actions
<table>
<thead>
<tr>
<th>Action Name</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shake Hands</td>
<td><img src="image" alt="Shake Hands" /></td>
</tr>
<tr>
<td>Wave</td>
<td><img src="image" alt="Wave" /></td>
</tr>
<tr>
<td>Lift Head</td>
<td><img src="image" alt="Lift Head" /></td>
</tr>
<tr>
<td>Nod and Flap</td>
<td><img src="image" alt="Nod and Flap" /></td>
</tr>
</tbody>
</table>

Table 2.5: Images of Philos V.1 actions continued.
The second major protocol, vision processing, is controlled by a Raspberry Pi connected serially to the mbed. The location of the user is determined by algorithms embedded on the Raspberry Pi and is transmitted to the mbed which then decides whether to move to face the user and the commands the servos to pan the head to the target location.

To create a smoother, more natural looking motion a servo control algorithm is employed. On non-smoothed motion the start and stop of a rotation can be very jarring or sudden which can seem unnatural to an observer. This algorithm attempts to solve this by breaking a rotation into a series of smaller rotations and scaling the speed of rotation so that it is a smoother acceleration.

To allow for an observer or user to manually control the actions of Philos V.1, a manual mode was added to Philos behavioral program. This mode is controlled by the GUI running on a host computer that communicates with Philos V.1 via wireless XBee. In this mode, a user can select specific actions that are pre-programmed on Philos (such as waving, flapping arms, etc.) as well as having the ability to control the audio playback device to play the pre-recorded sounds that are included in Philos repertoire.

**Vision Processing**

Two different vision processing algorithms were embedded on Philos V.1 depending on the needs of the particular session. A face detection algorithm using Haar classifier and skin color detection, based on the work of Tao Liu (also a member of the Distributed Intelligence and Robotics Lab), can be used when desired or a motion detection algorithm when speed of detection is preferred over accuracy. The face detection algorithm uses the AdaBoost Classifier, Haar classifiers, and a skin color based algorithm. However, these methods proved to be too computationally intensive for embedded, real-time tracking. In order to remedy this, two methods are applied: 1)
### Table 2.6: Detection time with various sizes of the searching area applied. PA: Potential Area, SA: Searching Area [34].

<table>
<thead>
<tr>
<th>Searching area</th>
<th>Original</th>
<th>120% PA &amp; 80% SA</th>
<th>120% PA &amp; 90% SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>50×50</td>
<td>985 ms</td>
<td>433 ms</td>
<td>376 ms</td>
</tr>
<tr>
<td>80×80</td>
<td>1303 ms</td>
<td>553 ms</td>
<td>462 ms</td>
</tr>
<tr>
<td>160×160</td>
<td>2997 ms</td>
<td>739 ms</td>
<td>507 ms</td>
</tr>
<tr>
<td>400×400</td>
<td>7814 ms</td>
<td>1190 ms</td>
<td>696 ms</td>
</tr>
</tbody>
</table>

Reducing the area inside the frame that is being searched for a face and 2) optimizing the area that is being searched by predicting where the face is moving to.

First, the original image is converted from RGB to grayscale and an equalization process is applied to smooth it. Secondly, the future face location that the algorithm searches is estimated by assuming that the user’s face will move without changing speed or direction using the results from the face detection in the previous few frames. The speed and direction of the user’s face can be determined by comparing the location of the face in consecutive frames. Depending on the speed of the detected face, the size of the potential searching area is adjusted. If the face is moving at high speed, the potential searching area is increased, and if the face is moving at a lower speed then the potential searching area is diminished. In addition, the potential area will increase by 20% of the previous detected face and decreases the searching area by 10% if the face is moving closer or farther from the camera. A comparison of detection times with various sizes of searching area and potential areas is shown in Table: 2.6.

Once a face is detected by the algorithm, the relative position of the face is sent to the mbed microcontroller. The relative position of the face is determined by dividing the entire area of the frame into zones where zone 1 correlates to the far left, zone 2 is left of center, zone 3 is center, zone 4 is right of center, and zone 5 is far right. Sending the zone number that the face is detected in to the mbed then allows the robot to rotate its head relative to its current position in order to face the detected
Another option is to use the embedded motion detection algorithm, also being developed by Tao Liu. This option becomes useful for detecting people that are interacting with Philos V.1 from a distance as it may not be possible to obtain images suitable for face detection at a distance of more than ten feet. By comparing two consecutive frames and calculating the difference, it is possible to locate a person interacting with Philos V.1. If two pixels in consecutive images show the same value it is marked as a 0, but if not it is marked as 1. Then the center of the area marked with 1 is calculated as the probable location of the person. In order to reduce processing time, the image is first converted to gray scale. The relative position of the person or moving object is then sent serially to the mbed microcontroller using the same method as in the face detection algorithm with five separate zones.

### 2.2.3 Integration with Wearable Health Sensors

A wearable health sensor is presented in Section 3 that is integrated with Philos. The microcontroller and sensor embedded in the wearable sensors make the sensor capable of determining how the user is moving, which serves as a form of gesture recognition. Wireless communication with Philos is established using wireless XBee technology.
2.2. TECHNICAL IMPROVEMENTS

2.2.4 GUI Update

An updated GUI was created using Visual C#. This GUI allows a host computer to communicate with Philos which enables real–time monitoring of the FSR data as well as enabling manual control of Philos V.1. Communication is enabled by using a wireless XBee module connected to a computer via USB dongle. FSR data is collected by Philos V.1 and is transmitted to the computer. The GUI displays the FSR data in real–time as both the absolute values and a graphical representation of the data. The FSR data is displayed with a timestamp and the analog readings from all six sensor regions: head cluster, chest cluster, left arm, right arm, left foot, and right foot. This FSR data is also displayed graphically for an easier to read representation for real-time interpretation of user interaction. The FSR data is represented by a picture of Philos V.1 with bars that fill up based on the FSR reading, where an empty bar is no pressure and a full green bar is high load. The data recorded and displayed can be exported to a text file so further analysis can be performed at a later time if desired.
Manual override of Philos V.1 is also controlled via the GUI. Manual override can be triggered by actuation of a switch that is on the back of Philos V.1. Triggering this switch creates an interrupt of Philos’ programming and puts Philos V.1 in manual override mode where it listens for commands from the XBee. A user can control Philos V.1 by selecting actions for the robot to perform. The GUI user can select actions by clicking on a body part on a picture of Philos V.1 and selecting an action that is related to that body part from a drop down menu that appears. Alternatively, there is another dropdown list that contains all of the actions, and preprogrammed voice commands that are embedded on Philos V.1 and can select a command to perform. A table of programmed actions and the body parts that are associated with them are listed in Table: 2.7. This manual control allows a clinician or observer to follow a script with another user interacting with Philos V.1 and observe how they react to various prompts from Philos whether visual or auditory.

Additionally, this GUI interacts with the wearable health sensor and displays information on the gestures that are detected by Philos V.1.
2.2. TECHNICAL IMPROVEMENTS

CHAPTER 2. PHILOS V.1

Figure 2.15: Philos V.1 monitoring GUI that is divided into 3 sections: 1) Real-time FSR data that shows the absolute readouts from the FSRs on Philos and timestamps, 2) Real-time graphical FSR representation, 3) Philos V.1 manual control: user can click on a body part to select actions related to that specific body part (See Table: 2.7) or select from any actions and voice commands from the choose action dropdown bar.
2.2.5 Preliminary Results

With the help of residents at the Judson Manor Senior Living community a preliminary test of residents’ interactions with Philos V.1 was performed. Six residents were exposed to Philos V.1 and asked to provide feedback about their interaction with the robot as well as any hopes or functions they would like to see in a social robot. After a brief introduction of the functions of Philos V.1 such as face tracking and how the robot could respond to touches to various parts of its body the residents spent some time interacting with the robot. Most of the residents had positive reactions to the robot after spending some time with it. They responded positively to the reactions of Philos V.1 and projected human–like personalities onto the reactions of Philos V.1 when the robot reacted positively or negatively to their touches.

Although, the majority of residents had a positive experience with Philos, and a few commented they would even like one of their own, there were many suggestions for modifications or features that could make the Philos V.1 platform better suit their needs. One resident mentioned the desire for a social robot that is also mobile and could go on walks outside with her. For her, a social robot would ideally be able to go on walks outside as a companion and would be able to warn her of any broken sidewalks or other treacherous obstacles that might provide treacherous footing.

One of the most requested features was for Philos V.1 to act as a personal assistant or scheduler. Many residents that interacted with Philos V.1 remarked that they would like to be able to schedule events or dates with the robot and be reminded in the future of the event. Similarly, a major concern with residents is remembering to take their medication as prescribed and the ability for Philos V.1 to manage reminders for when to take each medicine would be a very helpful feature.

Additionally, multiple residents mentioned that they like the idea of Philos V.1 watching over them and getting assistance in case of emergency. Falls in one’s own home or room is a serious risk for senior citizens and the ability for Philos V.1 to
be able to determine if a resident had fallen and then alert a nurse, or to call 911 if requested by the user would give the residents peace of mind. Implementation of such functionality would also make the lives of nurses and staff at a resident facility much easier by keeping a watchful eye over residents and allowing them to get assistance immediately if an emergency were to ever arise.

2.2.6 Commercialization

The estimated cost of Philos V.1 is projected to be less than $3,000 after commercialization. When compared to other social robots mentioned in the introduction that have similar functionality as Philos V.1 the cost is much less prohibitive than other social robots. Paro, the seal-like robot, has a commercial price of about $6,000 while Nao costs over $15,000 and DARwIn retails for $12,000. Philos V.1 has similar functionality to these robots, but could be commercially available at only a fraction of the price.

2.2.7 Future Work on Philos V.1

Although the upgrades to Philos V.1 addressed many of the problems that were noted with the original platform, through testing and collaborative meetings it became apparent that there was still much room for improvement. Some areas for improvements that were discovered were:

- The sturdiness of the platform
- The high difficulty in making adjustments to the software and hardware
- Lack of compatibility with a tablet
- Not much room for future expansion
Due to the rate of crashing in Philos V.1 as mentioned previously, despite various attempts to repair the problem, a new more robust platform is required for Philos to be more effective as a social robot, especially when dealing with people that may accidentally play rough with the robot. Another problem that was noticed with Philos V.1 is how difficult it was to reprogram or get to the internal components for maintenance. The hard plastic shell and fabric cloth covering were not designed with easy assembly and disassembly in mind. Another request from our collaborators at the Cleveland Clinic was for integration of a tablet into Philos so that they could have applications such as discretized training (DT training) with children with ASD. In order to perform these improvements to Philos V.1, it was necessary to build an entirely new robot to address these concerns and to have room for future expansion of abilities.
Chapter 3

Wearable Health Sensor

3.1 Technology Description

The wearable health sensor is manufactured via 3D printing in–house with a Makerbot Replicator 2x. Two different filaments are used in production: an ABS filament and a flexible filament. The design of the device is that there are two small boxes that house the components of the wearable device (Figure: 3.1). These boxes are joined by ribbon cable which allows the boxes to move with the user’s wrist rather than constraining the user’s motion or feeling uncomfortable. Each box has a top and bottom half. The bottom half holds the circuit board with the components and is printed from ABS filament so that it is rigid and protects the circuit. The top half is designed to cover the components from the environment and is printed with Ninjaflex, a brand of thermoplastic polyurethane filament for 3D printing. Ninjaflex is a flexible, high tear resistant material that was chosen because the flexibility of the material improves the comfortableness of the device and also creates a tight seal with the bottom half that makes it easy to assemble but difficult for the top to fall off during use.

Each box measures $1.85'' \times 1.85'' \times 0.875''$ with a small slot in the side of the
3.1. TECH. DESCRIPTION

CHAPTER 3. WEARABLE SENSOR

box that houses the battery that allows for toggling of the power of the sensor with a pen or a finger. Two boxes were used in this design since a single box with a long circuit board would be very rigid and was found to be too uncomfortable to wear for an extended period of time. By using two small boxes, the wearable sensor can twist with the user’s arm and does not restrict motion. The device is secured using velcro straps that thread through attachment slots that extend off the side of the bottom half of the boxes. A small cover for the pulse sensor allows for the velcro strap to thread through it so that the pulse sensor can be held in place as well.

A separate hub is designed for communication with a host computer and serves as a node for social data monitoring. The hub measures 2.50” × 3.55” × 1.05” and contains the circuit for communication with the computer. There is a slot on the top of the hub that allows for FTDI reprogramming and serial communication with the microcontroller.

![Figure 3.1: Wearable health sensors a) with cover on, b) without cover on.](image)

3.1.1 Embedded Sensors and Measurable Data

The wearable sensor consists of the following components:

- **ATMega328**: The processor used was an Arduino ATMega328.

- **Tri-axial Accelerometer**: ADXL345 tri-axial accelerometer used to measure the motion of the device.
• **Infrared Temperature Sensor**: TMP006 infrared temperature sensor breakout from Sparkfun is used to measure the skin temperature of the user.

• **Pulse sensor**: Pulse/SPO2 sensor from Modern Device based on Silicon Labs Si1143 chip that measures the pulse of the user when placed on the skin over an artery.

• **Microphone**: Simple electret microphone to record sound and interactions between people.

• **XBee Module**: XBee module used for RSSI and for communication with hub computer.

• **Humidity Sensor**: RHT03 humidity sensor used to measure ambient temperature and humidity.

• **Tri-color LED**: NeoPixel RGB LED changes color to signal that a heartrate has been detected and when it is between 60 and 75 beats per minute.

• **Micro-SD Card Writer**: Enables all data to be recorded locally so that it can be uploaded to a computer later for analysis.

• **7.4V battery**: Rechargeable lithium polymer battery provides power to wearable device.

• **Power Switch**: Allows power to be toggled with the flip of a switch.

• **Recharging Port**: Allows for charging of the batteries with a power source of 8.4V.

These components are split between the two boxes of the wearable sensor: the box closest to the wrist contains the environmental and biological sensors, as well as the XBee module for wireless communication between devices and the hub. The box
further up the forearm contains the microcontroller, SD card reader, power switch, battery, and charging port. The two boxes communicate via wires connecting the two boxes. The component placement can be seen in Figure: 3.1.

The wearable health sensor project is a collaborative project with two other members of the Distributed Intelligence and Robotics Lab: Sahil Kothari and Donghwa Jeong. Selection and calibration of most of the modules was the responsibility of Sahil Kothari and the RSSI algorithm was embedded by Donghwa Jeong.

Microcontroller

An Arduino Pro Mini microcontroller based on an ATMega328 chip is chosen for control and interpretation of sensor data. This microcontroller was chosen due to the high number of GPIO pins and there are open source libraries provided by the manufacturers of the pulse sensing module. Programming of the board, as well as direct serial communication with a computer is enabled by a FTDI to USB device. Power is provided for the microcontroller and the modules by the batteries that are included. A detailed circuit diagram for the wearable device is shown in Figure: 3.2.

Biological Measuring Modules

The wearable health sensor gathers biological data on the skin temperature, the pulse of the user, and acceleration data. Skin temperature is measured with the TMP006 infrared temperature sensor breakout board from Sparkfun. The module communicates with the Arduino microcontroller using I2C communication to return the temperature of the user’s skin.

Two pulse sensing modules were evaluated for use in the wearable sensor: the Pulse Sensor Amped, and an oxygen saturation module from Modern Device that conveniently can be used for measuring a user’s pulse. The Pulse Sensor Amped is attractive in that it is an open source module that is designed specifically for
Figure 3.2: Circuit diagram of the wearable health sensor that shows the wiring of the biological, environmental, and social monitoring modules.
Arduino and is relatively inexpensive. However, through testing there are multiple shortcomings to this device. Firstly, the sensor is designed to be attached to a finger or earlobe of the user which is not ideal for this project as long term wearing of this module could inhibit a user’s actions if attached to the finger or become uncomfortable if attached to the earlobe. Another significant problem with the Pulse Sensor Amped is the design of the module itself; the module is not enclosed and the soldered leads are exposed to the environment. This makes the module very unattractive since there is a potential of a short if the user’s skin is oily or sweaty. Due to the low current draw of the device (4.5mA), the severity of any injury due to a short is minimal but it could cause discomfort over a long period of time. Due to these severe drawbacks, the Pulse Sensor Amped was not used in the wearable health sensor, and the module from Modern Device is used.

The module from Modern Device was designed to measure the oxygen saturation of a user’s blood, but the device can also double as a pulse sensor. The one drawback to this module is that there is a sweet spot that it is most successful at detecting pulse and if the module is not placed in this location it can give no reading or inaccurate data. In order to achieve the best pulse sensing, the module must be pressed closely against the skin on the underside of a person’s forearm against the radial artery. Due to this, a very thin cover is designed for the pulse sensor that allows the wrist strap to hold it in place and allows the LEDs to make direct contact with the user’s skin on the underside of the wrist. The cover can be seen in Figure: 3.3.

The cover is very thin (0.05”) so that the LEDs of the module are as close as possible to the user’s wrist to limit the amount of noise caused by ambient light hitting the sensor. The slots on the side of the main body allow the velcro wrist strap to be threaded through and fasten the module securely in place. A small path that intersects one of the slots allows for the necessary wiring of the module to the Arduino microprocessor.
Some filtering of the pulse data that is gathered is required due to the fact that the wrist strap cannot hold the pulse module perfectly in place which will cause some deviation in the measured pulse. To counter this deviation, Donghwa Jeong and Sahil Kothari developed a filtering algorithm. The pulse measurements are filtered by discarding data that are considered to be unrealistic or impossible. Data values under 35 bpm (beats per minute) and above 120 bpm are considered to be outliers due to poor connection with the skin and are thus discarded since a human does not normally experience a pulse rate in these ranges unless there is some serious problem.

To provide pulse feedback in real-time to the wearer of the device, a tri-color NeoPixel LED is embedded in the cover of the box that is farther up the forearm. This LED is initially turned off on startup of the device but illuminates when a pulse is detected. When a pulse is detected to be between 60 and 75 bpm the LED glows green. However, if a pulse is detected but is not within this preferred pulse range the LED turns red.

Acceleration data is gathered using a tri-axial, low power MEMS accelerometer. Communication with the ATMega328 is achieved over I2C. Raw acceleration data is gathered for the X, Y, and Z directions. In order to convert this value to a more standard unit of g values a conversion is necessary. Unit conversion is possible by
3.1. TECH. DESCRIPTION

CHAPTER 3. WEARABLE SENSOR

multiplying the raw value by a scalar which is calculated using Equation: 3.1.

\[
Scalar = \frac{Range}{2^{10}} \quad (3.1)
\]

The ADXL345 is configured for measurement in 10 bits and the range of values has been set to 8g. Therefore the scalar is equal to \(\frac{8}{2^{10}}\), or 0.0078. Acceleration values in terms of g can then be attained by:

\[
x_g = x_{\text{raw}} \times 0.0078 \quad (3.2a)
\]
\[
y_g = y_{\text{raw}} \times 0.0078 \quad (3.2b)
\]
\[
z_g = z_{\text{raw}} \times 0.0078 \quad (3.2c)
\]

The change in the accelerometer data can be used for gesture recognition. Gesture recognition is made possible by comparing the accelerometer changes with experimentally obtained values. Preliminary work on gesture detection using acceleration data has been conducted by Sahil Kothari based on the work of Kwapisz, et al. [35]. Sahil Kothari experimentally gathered these values by wearing the sensor while performing the desired motions for detection. The motions chosen for gesture recognition are waving, typing or writing, running, and no activity due to the distinctiveness of the gestures and the intended use of the wearable sensor. The waving gesture is chosen for integration with the Philos platform to enable further human–robot interaction, typing or writing is chosen because a large number of potential users that work in an office setting will spend a large portion of their day typing or writing at their desk, the running gesture allows for the wearable sensor to be used while a user is exercising and monitor how their vitals change over the course of a workout, and no activity captures any unknown motions as well as being used to determine periods of
inactivity. Accelerometer values were measured in real-time using a FTDI to USB device for serial communication with a computer. In addition, they were recorded for analysis. By analyzing the accelerometer data it is possible to determine trends for these chosen motions.

With these values, the embedded microcontroller can determine if the current values being gathered match any of the gestures that have been experimentally tested. If these gestures are detected, the sensor wirelessly communicates this to a GUI and records the gesture on the embedded micro-SD card. The gestures that are recognizable by the wearable health sensor are: waving, typing or writing, running, and no activity. The current algorithm uses the parameters that were obtained from a specific person’s acceleration data, so additional data from other persons is needed for generalizing the algorithm.

**Environmental Measuring Modules**

Environmental data is collected on the ambient temperature and humidity. Ambient temperature and humidity are measured with the RHT03 sensor. This sensor has small power and voltage requirements but has very high precision. Communication is enabled through a single wire for data which is convenient as the large number of modules embedded use up a lot of GPIO pins. Relative humidity measurement is accurate ±2% and the temperature measurement is accurate within ±0.5 °Celsius.

**Social Data Monitoring**

Quantitative social data monitoring is achieved by determining the distance between user’s and the amount of time that users spend in close proximity with each other. A RF–based signal strength method is adopted to measure the proximity among users. Using the XBee modules embedded in each device a Received Signal Strength Indication (RSSI) technique is applied to obtain a relative measure of distance between
users. Threshold values for close proximity in terms of RSSI values are experimentally obtained by having three users wear a device and examining the values measured with respect to the physical distance between the users in a variety of settings including inside and outside as well as crowded environments. This experiment showed that there are distinct thresholds that can determine if users are close together in a potentially social environment even in crowded or indoor areas. There can be issues with measuring the physical distance if there is an object like a wall or a door between users as the signal will not be as strong, but this is not particularly problematic for this device as a wall or door between users means they are probably not interacting socially. Using these threshold values it is possible to determine when multiple users are in close proximity that would suggest some social interaction between the users. One drawback of this technique is that each device must measure the distance between each other device that is within communication range, so far large scale uses of more than six devices there may be some delay due to a large amount of communication traffic between XBee modules. This RSSI technique was mainly carried out and embedded by Donghwa Jeong.

Future work will focus on qualitative social data assessment by adding voice processing module into the wearable sensor that can detect the emotional status, prosody, and stress level by combining the user’s biological and voice data.

3.2 Graphical User Interface

The wearable sensor has a standalone GUI that displays real-time information from the embedded sensors. The wearable sensor GUI is shown in Figure: 3.4. The wearable sensor GUI is composed of six different sections. The first section in the upper left of the window displays user information such as name, gender, and age. The upper middle section displays which activities the user has been performing since
the session began by keeping count whenever the wearable sensor detects that the user has performed one of the known gestures. In the top right corner of the window there is a graphical pie chart representation of the activity monitoring section that gives a percentage breakdown of how much time the user spent performing each action. The environmental monitoring section displays real-time updates on the ambient temperature and humidity. A user’s pulse is displayed in two ways: firstly the current user pulse rate can be displayed by selecting the button to “Calculate BPM” or by viewing the line graph that displays the user’s pulse over the last four minutes. In order to create a more accurate pulse line graph obviously inaccurate data points are filtered out if they are extremely high to the point of being impossible or extremely low which are potentially due to errors in recording such as a poor connection with the radial artery. A more complete analysis of data can be performed by removing the micro-SD card that is onboard the device and reading the .txt file that serves as a data log.

3.3 Integration with Philos

The wearable sensor is designed to also integrate with the Philos V.1 platform. Communication between the sensor and Philos V.1 is achieved using wireless XBee modules that are embedded on both platforms. Integration of the wearable sensor and Philos V.1 is best realized in a one sensor environment rather than when there are multiple users with wearable devices. This is due to the large amount of traffic that must occur if the wearable sensors are communicating and determining proximity using RSSI and it would limit the ability for Philos V.1 and the wearable device to communicate. The wearable sensor can serve as a tool to aid Philos V.1 in interacting with the user through use of the accelerometer in the wearable sensor. The wearable sensor is capable of reading the accelerometer data and determining if the user is
3.3 INTEGRATION WITH PHILOS

CHAPTER 3. WEARABLE SENSOR

Figure 3.4: Graphical User Interface of the wearable health sensor. Sections 1) User information 2) Activity monitoring counts the number of times each gesture is detected by the wearable sensor 3) Graphical representation of activity monitoring in Section 2 4) Environmental monitoring of ambient temperature and humidity 5) Current BPM of the user (updated on a user clicking “Calculate BPM” 6) History of the last ten measured pulses of the user.
performing certain gestures, such as waving. If the user waves at Philos V.1, that information is sent via XBee and Philos V.1 will wave back at the user. Philos V.1 could be programmed to work in conjunction with the wearable sensor to monitor a user’s biological data as well, warning a user if their pulse or temperature is reaching some unsafe level.
Chapter 4

Philos V.2

4.1 Goals

There were five main goals that were identified as means of improvement on the old design of Philos:

- A sturdier platform
- Ease of maintenance
- Compatible with tablets
- Room for expansion
- Universality and reprogrammability

One of the major goals in the redesign of Philos was to create a more robust platform. Since one of the target demographics of Philos is children it is necessary to design a social robot that can withstand the rigors of play with a child. The original Philos had issues with rough play causing the internal electronics to get jostled which caused a significant amount of crashing, a more robust platform would drastically increase the reliability of the robot.
4.1. GOALS

Maintenance of Philos is also a significant problem currently, as any maintenance or updating of code requires disassembly of the outer covers of Philos including the outer cloth covering and the hard plastic shell. This process is tedious and time consuming for an average user that may want to use Philos for personal or clinical use that needs to change parameters or update behavioral reactions. Therefore, there are built-in ports on the shell of the robot for HDMI and USB connections with the Raspberry Pi that is controlling the robot so that maintenance can be performed easily on the robot without any disassembly required.

One of the improvements suggested through collaboration with clinicians at the Cleveland Clinic was for there to be some way to make Philos compatible with a tablet device. Compatibility with a tablet would allow clinicians who work with children with ASD to use programs such as DT trainer which has shown some success with behavioral training. In addition, this compatibility could allow for the integration of many different applications that have been suggested. From an initial exposure group with elderly and their reaction with Philos one of the suggestions was the ability to use Philos as a calendar or appointment reminder. These capabilities could be achieved through integration with a tablet and creation of multiple applications that run on this tablet.

Room for expansion was also a large concern in the design of Philos V.2, which would allow for modification of Philos in terms of both hardware and software capabilities as the needs or target demographics of Philos evolves. To account for this possibility, Philos V.2 is designed with a large interior body that both allows for the tablet to fit into the shell as well as allowing for the addition of hardware as it is required. Hardware such as additional microcontrollers or modules that enhance the capabilities of the social robot could fit in the interior of the robot easily as opposed to the limited space in Philos V.1.
### 4.2 Mechanical Design

Another major goal of the Philos V.2 project was to make a social robot platform that is universal and reprogrammable. This goal is to design a platform that can be used for a wide range of applications and can be easily redesigned or reprogrammed for a specific goal. This platform should easily allow for the embedding of algorithms in the future and have room for future adaptations.

#### 4.2.1 Internal Structure

In order to create a lightweight yet strong structure two materials are used in the design of the internal structure. The majority of parts are made from 3D printed ABS due to its low density but relatively high strength, whereas certain important structural components are machined from aluminum 6061–T6 (material properties shown in Table 4.2). In addition, the 3D printed parts are not solid parts as the software that prepares the parts for printing converts them to save on material which

<table>
<thead>
<tr>
<th>Measurements</th>
<th>15.5” × 16” × 14”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degrees of Freedom</td>
<td>5</td>
</tr>
<tr>
<td>Materials</td>
<td>3D printed ABS and Aluminum 6061</td>
</tr>
<tr>
<td>Controller</td>
<td>Raspberry Pi Model B</td>
</tr>
<tr>
<td>Servos</td>
<td>3 × Dynamixel AX-12A servos</td>
</tr>
<tr>
<td>Actuation</td>
<td>2 × Firgelli linear actuators</td>
</tr>
</tbody>
</table>

Table 4.1: Philos 2.0 Specifications.
Figure 4.1: a) CAD representation of Philos V.2 and b) actual hardware assembly with open space in the center where the tablet inserts.
Table 4.2: Material properties of Aluminum 6061 and Makerbot 3D printed ABS.

<table>
<thead>
<tr>
<th>Material</th>
<th>Yield Strength</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aluminum 6061</td>
<td>35,000 psi</td>
<td>0.0975 lb/in(^3)</td>
</tr>
<tr>
<td>3D printed ABS</td>
<td>5,532 psi</td>
<td>0.0376 lb/in(^3)</td>
</tr>
</tbody>
</table>

Further decreases the weight of printed parts. The internal structure of Philos V.2 can be broken into two distinct segments: the body and head.

**Figure: 4.2** shows a CAD representation of the body of the new Philos robot. A $\frac{1}{10}$” thick baseplate connects to four structural posts. The baseplate has eight holes drilled for attaching of the robot shell and screw holes to attach a standoff for a Raspberry Pi that serves as the brains of the robot. The baseplate was initially machined from aluminum 6061, however the weight of the baseplate was found to be much too large, therefore the baseplate was printed in-house out of Makerbot 3D printed ABS. This switch in material significantly lowered the overall weight of the robot and still provides sufficient stability for the base.

Connected to this base plate are four structural posts that connect to the top plate as well as serving as a mounting point for the arms. These posts are made from $\frac{1}{4}$” × $\frac{1}{4}$” aluminum 6061 bar stock. These posts were fabricated from aluminum because the posts are intended to support the top plate and head as well as the arm structures and so had to be fairly strong. In addition, 3D printing of the posts would not have been possible in house because of their length of ten inches; this would have necessitated printing the posts as a series of parts and attaching them together which was not desired due to the need for some structural strength. Mounted to the top of the four posts is the top plate of Philos that is a rectangular plate with mounting holes for the head assembly of Philos. The body and head of the robot mount together but do not screw or lock in as the weight of the head holds it in place and can be removed for easy maintenance. Also mounted to the four posts are the arm structure of Philos which are described more in depth in **Section: 4.2.1.**
The head assembly of Philos is designed to fit onto the top plate of the body and rotate the mounted camera and screen that displays the face of the robot. The overall head assembly can be seen in Figure: 4.3. The head assembly is composed of a circular plate with protrusions that slot into the mounting holes on the top plate of the body structure. Attached to this plate with a mounting bracket is the AX–12 servo motor that allows the rest of the head structure to rotate about the Z axis. Attached to the horn of the servo is a u–shaped bracket that connects to the face mount as well as to the outer shell for the head of the robot. Coming off the u-shaped bracket is an L–shaped connector to the plate that the LCD screen that serves as the face of the robot and a small bracket for the attachment of a Raspberry Pi camera module.

Arm Structure

The two arms of Philos are designed to be extendable. When the robot is not in use, the arms of the robot retract to make the hands flush with the shell of the robot. The structure that enables the actuation of the arms connects to the support posts of the body and can be seen in Figure: 4.4.

In order to realize the actuation of the arm there are three major components: the
4.2. MECHANICAL DESIGN

Figure 4.3: CAD representation of the head structure of Philos V.2.

Figure 4.4: CAD representation of the arm structure of Philos V.2.

linear actuator, the guide hole, and the shaft. The linear actuator is a Firgelli brand actuator that is mounted off of the support posts of the body. This linear actuator drives the shaft that links the actuator with the hand of Philos. The driving of the shaft extends the arms outwards or retracts them back into the body. However, this structure is not very stable and may not retract correctly back into place. Therefore, a guide was designed that the shaft travels through. This guide limits the movement of the shaft to one direction allowing the arm structure to reliably extend and retract back to its original position.

Connected to the end of the shaft is the hand of Philos. The hand consists of a servo motor with mount, a shield, and the hand itself. The servo motor mounts onto
a ledge of the shaft structure with a mounting bracket. Connected to the shaft is also a shield; this shield serves to protect the internals of Philos from liquids or from the eyes of observers or users to promote a more aesthetic appearance. The arm itself is a slender 3D printed concave shape that mounts to the horn of the servo motor for the hand. This is the part that is rotated by the servo to perform gestures and that is directly interacted with by the user.

**Servo Placement and Actuation**

Philos contains three Dynamixel AX–12 servos and 2 Firgelli linear actuators for the rotation and actuation of moving structures. The selection criteria for these specific motors are discussed in **Section: 4.2.3.** Two of the Dynamixel servos are part of the arm structures of Philos. These two servos are mounted onto a ledge of the driving shaft for the extension of the arms and enable rotation of the hands around the X axis. The third Dynamixel servo is mounted onto the base plate of the head structure. This servo enables the head with the face and camera to rotate around the Z axis and pan around the room as needed. The two Firgelli linear actuators are used in the arm structures of the robot. The linear actuators drive the shaft that extends the arms from the body and retracts them back to being flush with the shell.

The combination of these motors gives the robot five degrees of freedom:

- Rotation about the X axis for each arm
- Rotation about the Z axis for the head structure
- Extension along the X axis for each arm

**4.2.2 External Structure**

The face of Philos is intended to look futuristic or robotic. This is achieved by using a LCD screen from 4D Systems. This LCD screen module is designed specifically to be
controlled by a Raspberry Pi via GPIO ports. One of the most attractive qualities of this LCD screen is that it has its own graphics processor, which drastically cuts down on the burden that is placed on the Raspberry Pi to display images such as faces on the screen. This onboard graphics can be utilized by designing the faces desired using 4D Systems provided software for PC and then uploading the images onto the screen via FTDI to USB cable. Once these images are loaded and pre–rendered on the LCD’s memory they can be easily toggled between through serial communication from the Raspberry Pi by calling the ID of the different faces, rather than controlling the image generation directly with the Pi. One drawback to the use of this LCD screen is that the shape of the LCD screen creates some space between the shell of the robot and the screen itself as it cannot become flush with the curved inner surface of the shell. One potential solution to remedy this would be to instead use an LED matrix with a dome shaped diffuser to project faces to the user. This solution might require a separate controller such as an Arduino that communicates serially with the main controller due to the extra load that it may place on the Raspberry Pi.

**Shell**

In order to contain and protect the internal electronics and structure of Philos V.2, a shell structure was designed. This shell structure gives Philos V.2 its look and is the outward facing structure to the user. This shell was designed to roughly resemble an egg, with the arms of the robot that can come out or withdraw back into the shell and a head that can rotate freely. Due to the large nature of the robot and the need for the shell to be strong but have some give to it there was one natural selection for material. The shell of the Philos V.2 prototype was printed using Makerbot ABS filament with a Makerbot Replicator 2X. It was decided to use 3D printing for the manufacturing of the shell in in order to reduce cost. The size of the shell meant that manufacturing via a third party or even in Thinkbox’s Fortus 3D printers could cost
over $3000, whereas printing it in house using a Makerbot Replicator 2X cost merely $250 in filament.

The downside to the Makerbot is the size of the build plate that requires for large pieces like the shell to be split into many smaller pieces and then assembled together using an adhesive. However, even with the large build plate of the Fortus, the shell is so large that the shell would still have to be split into a few pieces and reassembled. This can cause some problems with the appearance of the robot’s outer shell since the pieces may have some warping or imperfections that cause them to not align perfectly. This is a definite problem with the Makerbot printed parts because up until a recent firmware update there were often imperfections or lifting of the part off the plate that would cause pieces of the shell to not align well. These imperfections could be covered up by using some paint to even out the appearance. Alternatively, if a cloth covering is used, the imperfections in the shell will not be noticeable by the user. Another drawback of this manufacturing technique is the time it took to print. The shell is divided into three separate parts: a head shell and two shells for the body.

The finished shell of the robot has some visible seams and cracks, but some polishing or finishing of the surface should make it look more aesthetically pleasing. In the center of the shell is a shelf that is of sufficient size to dock a Samsung Galaxy tablet. This shelf also has holes running to the interior of the robot for routing of power or communication wires to the tablet as needed. The docking region for the Galaxy tablet can be seen in Figure: 4.5.

In order to make the robot more mass producible, it may be necessary to use a different manufacturing process such as vacuum forming or molding. These manufacturing techniques would be significantly faster than 3D printing and would also yield more consistent results especially if the shell can be made in one single piece rather than multiple pieces assembled together.

The shell for the head is separate from the rest of the shell to allow the head
to freely rotate since the camera is mounted in the head and it is necessary for the head to be able to pan around the room and track users. The shell that covers the main body of the robot is split into two parts: the back and front. It was decided to split the body shell into two parts to allow for easy maintenance and assembly. The body shell connects to the body through screws to the bottom of the base plate of the body structure and the two halves secure themselves to each other using rare earth magnets that are mounted in the shell. Another feature of the back half of the body shell is that there are programming ports that align with the Raspberry Pi that is sitting on the standoff. These ports allow for easy programming, troubleshooting, and data analysis or transfer through HDMI and USB ports.

4.2.3 Electronics

Controller

Philos V.2 is controlled by a Raspberry Pi model B. The Raspberry Pi is a small credit card-sized single-board computer developed by the Raspberry Pi Foundation. The model B Pi boasts a 700 MHz ARM1176 processor with 512 MB RAM. The Raspberry Pi also has a built-in graphics processor. This board was chosen for Philos
2 because of its high computational power with respect to its relatively small size and power consumption. The power of the Pi allows for implementation of embedded image processing algorithms. The Raspberry Pi also boasts 26 GPIO pins that are adequate for control of the servos and actuators of Philos 2.

**Servos**

The servos used in Philos V.2 are the Robotis Dynamixel AX–12A servo. These servos were chosen for a variety of reasons. These smart servos were very attractive for use with a Raspberry Pi because they communicate with a half-duplex UART and utilize a daisy-chain system. This was important in selection because the one major downside of the Raspberry Pi is the limited number of GPIO pins for controlling the robot. The ability to control all three servos via one data pin was a deciding factor in the choice of using Dynamixel smart servos. Robotis (the makers of the Dynamixel servos) offers a variety of different servos that utilize this unique protocol. In order to choose the correct servo for Philos V.2 the torque requirements must be calculated.

In order to make torque calculations easier, the arms are modeled as cantilever beams:

\[ \tau_{arm} = W r_{cm} \sin \theta + F r_f \cos \phi \]  

(4.1)
Where \( W \) is the weight of the arm, \( r_{cm} \) is the distance to the arm’s center of mass, \( F \) is the applied force, and \( r_f \) is the distance to the applied force. \( \theta \) and \( \phi \) are the angles between these two vectors. In order to calculate the torque requirement for the arm servos, the max torque can be calculated with the above equation when \( \theta \) and \( \phi \) are 90° and the force is applied to the end of the arm.

The head is modeled as a solid cylinder for the sake of a rough estimate of the torque requirements:

\[
\tau_{head} = \frac{mr^2\alpha}{2}
\]

Where \( m \) and \( r \) are the mass and radius of the arm, respectively, and \( \alpha \) is the angular acceleration that will be produced.

The AX–12A is the smallest of the Dynamixel servo, which provides approximately 1.5 N m of torque operating at 12 V. Setting this stall torque as the value of both \( \tau_{head} \) and \( \tau_{arm} \) it is possible to find the angular acceleration that this servo can produce, as well as the maximum load that can be applied to the arm. For the arm \( r_{cm} \) is 0.164 cm, \( r_f \) is 8.16 cm, and \( W = 0.03 \) kg. It can be seen that \( Wr_{cm}sin\theta \) will be miniscule compared to the other term, however for the sake of a better estimate both terms were left in. Solving for the maximum applied force yields 18.37 N as shown in Equation: 4.3. For the head torque calculations \( r = 2.536 \) cm and \( m = 0.6973 \) kg. By rearranging the head torque equation as seen in Equation: 4.4 the calculated value for angular acceleration is 117 deg/sec². From these calculations, it has been shown that the AX–12A servo is appropriate for use in Philos 2. Detailed specifications of the AX–12A servo can be seen in Table: 4.3. From these calculated force and angular acceleration abilities of the AX–12 servos it was determined that
4.2. MECHANICAL DESIGN

<table>
<thead>
<tr>
<th>Weight</th>
<th>54.6 g</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions</td>
<td>32 mm × 50 mm × 40 mm</td>
</tr>
<tr>
<td>Stall Torque</td>
<td>1.5 N m</td>
</tr>
<tr>
<td>Communication Speed</td>
<td>7343 bps - 1 Mbps</td>
</tr>
<tr>
<td>Operating Voltage</td>
<td>9 - 12 V</td>
</tr>
<tr>
<td>Resolution</td>
<td>0.29°</td>
</tr>
</tbody>
</table>

Table 4.3: Detailed AX–12A servo specifications.

these servos would be adequate for use in Philos V.2.

\[ F = \frac{\tau - Wr_{cm}}{r_f} \]  \hspace{1cm} (4.3)

\[ \alpha = \frac{2\tau}{mr^2} \]  \hspace{1cm} (4.4)

Power Requirements

The various components of Philos V.2 require different voltages for operation, so a power board was designed to supply power. The Raspberry Pi requires 5 V 1.2 A supply, whereas the servos and actuators require 12 V. The board connects to a 12 V supply by a molex connector, this 12 V supply then branches into a 5 V regulator to power all modules that work on 5 V and the H–bridge ICs that are used to control the AX–12 servos and linear actuators. These ICs are necessary to provide power to the servos as the Raspberry Pi is not capable of providing the necessary voltage or high current required. A schematic of the power board can be seen in Figure: 4.7.

Where the components in the schematic are:

- **J1**: Power jack for 12 V power into the board, which is split between powering the servos and the 5 V regulator.

- **LM7805**: A 5 V regulator that takes the input 12 V and regulates it as a 5 V output up to 1 A of current.
4.3 HRI Algorithms

Human–robot interaction is realized through a combination of audio and vision algorithms. Philos V.2 algorithms include face detection, face recognition, and speech recognition algorithms. These algorithms are achieved with the use of a Raspberry Pi model B running the Raspbian Linux distribution.

4.3.1 Face Detection

Philos V.2 achieves face detection using OpenCV and using the Local Binary Patterns (LBP) approach. LBP cascade was chosen over the Haar classifiers because from experimentation LBP cascade performed at a higher speed with less latency than
the Haar cascade, at the cost of slightly lower accuracy. Speed was prioritized over accuracy for the embedded algorithms because of the limitations of a Raspberry Pi versus a computer. In addition, to increase performance of the algorithm, the images are first converted from RGB to grayscale to further speed up the detection process.

One of the largest factors in FPS improvements and CPU usage decreases is the use of new hardware for image capture. By using a Raspberry Pi camera module, from the Raspberry Pi Foundation, as opposed to a USB webcam the strain on the Raspberry Pi is greatly diminished. This is a byproduct of the camera module connecting directly to the GPU of the Raspberry Pi, whereas a USB camera takes up a significant amount of CPU usage to move the data over the USB. This frees up CPU cycles for further processing of images and other computations. The effect on the face detection and image stream using this algorithm and the Pi camera is that the FPS increases from roughly 6 FPS to 20 FPS (15 when a face is detected). Also, the delay in the image shrinks from roughly 250ms to only 75ms.

Figure 4.8: Face detection algorithm showing the detected face outlined.
4.3.2 Face Recognition

Upon startup, the database of learned faces is loaded. The database of learned faces is populated by taking ten pictures of a subject and recording their name. The pictures are taken automatically with a half of a second interval between pictures while the user provides a variety of poses such as directly looking at the camera and slightly profile views. The pictures are then automatically cropped based around the location of the detected user’s face and then a script written in Python automatically populates the CSV file that is used for training the faces. One issue is that the face detection sometimes receives false positives in addition to the human face that should be detected by the detection software. These false positives could affect the training of faces for the person so the false positives are filtered out with a simple algorithm. It was determined that the false positives are always secondary faces detected so the cropping algorithm saves the first detected face for a picture and then deletes subsequent detected faces as these are false positives. Using the LBP cascade we compare the detected face to the faces in our database to determine a match. If no match is found it is returned as an unknown person and, if desired, they can be added to the database and have their pictures taken.

The face recognition algorithm has a very small latency from real-time video, but the drawback is that to achieve this the face recognition is not completely reliable. For some people the face recognition may not work unless they are at a specific angle or are facing the camera directly on. This is a problem with the training and perhaps more pictures should be used in the training process. However, this is not a critical failure as the robot can still track a user with simple face detection which is very reliable and behavioral control algorithms can work around the lack of constant facial recognition by remembering who the detected user is throughout a session and acting accordingly.
4.3.3 Speech Generation/Recognition

Auditory interaction was desired on Philos to increase the ability of the robot to interact with humans naturally. Numerous methods were tested to achieve this goal such as PocketSphinx and EasyVR. These devices are commercially available and designed to work with an Arduino or other microprocessor. However, in testing, they were found to be fairly limited in their capabilities. These devices could be trained to recognize a limited number of commands and were found to work consistently, but only with the person who trained the commands. When other users would try to issue the trained commands, the success rate was abysmal which meant these commercial solutions could not be viable for Philos V.2.

Software solutions for voice recognition were too computationally intensive to be embedded on a Raspberry Pi, therefore recognition had to be performed off board using cloud computing. By using a wi–fi connection, Philos can record a user’s input and send the audio file to Google servers for speech to text conversion, much like
how smartphones achieve voice recognition. This is made possible by utilizing the Google Translate API. It is then possible to respond appropriately by comparing the returned text file from the cloud computing with a list of commands or prompts to look for a match. By using this method, robust speech recognition is achieved at a very low computational cost. In addition, audio playback can be achieved with this same method using the API it is possible to query the servers for a speech to text translation and then play that back over a speaker inside Philos. The one shortcoming of this method is that it only works when the Pi can connect to a wireless router.

### 4.3.4 Tablet Integration

Philos V.2 is designed to integrate with a tablet such as a Samsung Galaxy Tablet. The Philos V.2 platform is specifically designed with this tablet in mind for docking with the robot, however an Android tablet will function with the applications. The integration of a tablet allows for the design of a suite of applications that broaden the scope of the robot. Future applications for the tablet to interact with the robot could allow for the robot to serve as a scheduling tool for users that can remind them of appointments or to take their medicine on time, or to enable games that a user can play on the touch screen and communicate wirelessly with Philos V.2 to illicit some response to encourage a user when they perform well in the game.

### 4.4 Servo Control

The AX–12 servos operate on 12 V, whereas the Raspberry Pi utilizes 5 V logic. Therefore, an H–bridge IC is utilized, specifically the 74LS241. The pinout for this IC can be seen in Figure: 4.10.

The H-bridge IC works to step-up the logic voltage from 5 V to 12 V to control the AX–12 servos. The wiring schematic between the Raspberry Pi and the 74LS241
4.4. SERVO CONTROL

Figure 4.10: Pinout of 74LS241 H-bridge IC used for control of AX–12 servos, courtesy of [38].

are shown in Figure: 4.11.

Figure 4.11: Wiring schematic of Raspberry Pi and H-bridge IC for servo control.

In order to smooth the motion of the servos an algorithm is applied. In order to make the servo motions appear smoother, rather than being jerky at the start and stop of the servo’s motion smoothing is applied. This is possible with the AX–12
servos which allows control of both the speed and position of the servos. By starting the servo at a slow speed and ramping up towards the middle of the motion, and then slowing back down near the end the motion appears more natural to users. This servo control scheme was utilized as an improvement in the previous Philos. This servo motion algorithm breaks the initial servo position command into a series of small motions and differing speeds. A graphic representation of this algorithm can be seen in Figure: 4.12. The one downside of the servo smoothing that can be seen in the figure is that the time it take to move to a position is longer due to the time it takes to ramp up the speed as opposed to a near instantaneous full speed.

The three AX–12 servos embedded on the Philos V.2 platform provide the robot with a wide range of motion and enable a variety of gestures for interactions with human users. The arm servos allow the robot to make gestures such as extending a hand for shaking or a high five, waving, or dancing. The head servo allows for the robot to track a user or object as well as to display disagreement by shaking its head. A table showing the range of motion of the robot can be seen in Table: 4.4.
4.5 Preliminary Cost Analysis

The overall cost of Philos V.2 is still very low compared to other social robots available on the market. The estimated commercial price of Philos V.2 is comparable to Philos V.1 at about $3,300, one of the main reasons for the increased price between these two robots is the cost of the tablet that is included in Philos V.2. This relative increase in price between Philos platforms is still very reasonable compared to other social robots that are commercially available that range from $4,000 to over $12,000. A comparison of select commercially available social robots can be seen in Table: 4.5.
<table>
<thead>
<tr>
<th>Specification</th>
<th>Paro</th>
<th>Nao</th>
<th>DarwinOP</th>
<th>Philos V.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degrees of Freedom</td>
<td>8</td>
<td>25</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>Embedded Sensors</td>
<td>Touch sensor, microphone, light sensor</td>
<td>Touch sensors, directional microphones, two cameras, inertial measurement unit, sonar rangefinder</td>
<td>USB camera and microphone, 3-axis gyroscope, 3-axis accelerometer, FSRs (optional)</td>
<td>Microphone, camera, Touch sensor (future work)</td>
</tr>
<tr>
<td>Size</td>
<td>22.4” (Length)</td>
<td>23” (Height)</td>
<td>17.9” (Height)</td>
<td>14” (Height)</td>
</tr>
<tr>
<td>Cost</td>
<td>$6,000</td>
<td>$8,000</td>
<td>$12,000</td>
<td>$3,300</td>
</tr>
</tbody>
</table>

Table 4.5: Comparison of commercially available social robots.
Chapter 5

Future Work

5.1 Vision Upgrades

An embedded gesture recognition algorithm is one planned upgrade to Philos V.2. The reduced CPU load due to the integration of Raspberry Pi camera makes complex vision algorithms much more feasible. Gesture recognition software could be used to teach Philos a variety of commands or allow the robot to recognize common movements like waving and allow it to wave back at the user. Currently, a gesture recognition algorithm is being developed by Tao Liu to recognize hand gestures such as waving or trying to shake hands and could be integrated at a future time. Such an algorithm could also have very practical uses if trained to recognize specific motions. One such motion would be taking a pill or medicine. The ability to tell if a user has forgotten to take their medicine or has already taken their medicine would be a very practical solution for someone who is elderly or perhaps has memory problems.

Another vision algorithm that was considered but not implemented currently in Philos V.1 or V.2 is stereo vision. Stereo vision allows for the robot to perceive the environments in three dimensions similar to how humans and other animals do, with two image sources. By comparing and contrasting the images from two cameras
that are spaced apart it is possible to determine depth. The ability to determine depth could be used in future applications especially if the robot was used in mobile applications.

5.2 Tablet Apps

The incorporation of a tablet into the design of Philos V.2 adds a lot of room for customizability and adaptability for a wide range of uses. Development of new applications could allow Philos V.2 to be used for a wide variety of purposes. One such possible application would be a form of discretized training software (DT training software).

DT training software was one potential use for Philos V.2 that was brought to our attention by clinicians at the Cleveland Clinic Center for Autism. DT training software is widely used by clinicians that work with children with ASD as a fun way to help a child focus and learn. Many DT training programs utilize flashy rewards or sounds to encourage user’s to continue to play and learn; this would be a great opportunity to use Philos V.2. Integration of a commercially available or newly created application for DT training could feature a child or user playing the game and answering question while Philos V.2 would flap it’s arms or give audible encouragement to the child.

Another potential application of tablets is to create a personal assistant type application that integrates with Philos V.2. One particular feedback that was obtained from a pilot study with some elderly people at the Judson Senior Living Community was that they would like to have Philos act as a personal assistant or reminder for them. The ability to manage a calendar to remind them of appointments, dates with friends, and even reminding them to take their medicine regularly would all be very helpful in their day to day life and improve their living conditions.
5.3 Mobile Platform

A lack of mobility is a significant limitation of the current Philos V.2 platform. A mobile platform or pedestal that Philos can be mounted onto would increase the ability for Philos to act as a companion for users. This mobile platform could even be controlled by the onboard microprocessor as there is a port on the bottom of the robot that was included with this future expansion in mind. Some mobility is a feature that would have significant impact on interaction with both children and the elderly as moving toys are stimulating to children and Philos could act as a walking companion for elderly who maybe do not have someone to walk with.

5.4 Behavior

The current behavior embedded on Philos is a fairly simplistic algorithm that gives an output reaction based on the input given. For instance, if a face or touch is detected the robot will perhaps follow the face or wave at the person. A more adaptive behavioral algorithm could be adopted in the future to allow for the robot to learn and evolve based on repeated interactions with a user. One of the graduate students, Jianan Sheng, in the Distributed Intelligence and Robotics Lab is working on an adaptive behavioral algorithm that could be embedded on Philos V.2 one day. An adaptive behavioral algorithm would make interaction with Philos more immersive and feel more like a real pet or companion as the robot would have its own personality and would adapt over time based on how the user interacts with the robot.

5.5 Touch Cloth

Depending on the application, a fabric costume might be desired for Philos V.2. This costume might resemble an animal like in the original Philos, or some other
anthropomorphic shape. Conductive fabric could be used to create a costume that could detect touch anywhere on the robot. This would solve one of the potential drawbacks of Philos V.2, which is that the current design does not allow for the integration of traditional touch sensors such as FSRs as there is nowhere to hide them from the user. Touch fabric could make for an ideal solution as it would provide a cuter facade if desired and can detect touch at any location on the robot rather than just in localized hotspots where sensors are located.

5.6 User Interface

Additionally, a user interface for programming and monitoring of the Philos V.2 platform would be desirable in the future. A graphical user interface could be developed that would allow for a non-technical user to make modifications to the control algorithms of Philos and to allow for customization of Philos. This graphical user interface could either be a standalone computer program or could be developed as a tablet application.

5.7 Microcomputer

When the algorithms for Philos V.2 were being developed, the most powerful version of Raspberry Pi was the model B. Since then, two new models have been released, the model B+ and the model 2. Both of these models present certain advantages over the current model B in use and so these models should be considered in the future as potential upgrades to the computing power of Philos V.2. The detailed specs of these three models are shown in Table: 5.1.

Both of these models present certain advantages over the current model B in use and so these models should be considered in the future as potential upgrades Philos V.2. The increased number of GPIO and USB ports on the B+ would be useful
### Table 5.1: Comparison of Raspberry Pi Models

<table>
<thead>
<tr>
<th>Specification</th>
<th>Model B</th>
<th>Model B+</th>
<th>Model 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
<td>700MHz ARM1176</td>
<td>700MHz ARM1176</td>
<td>900MHz quad–core ARM Cortex-A7</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>512MB</td>
<td>512MB</td>
<td>1GB</td>
</tr>
<tr>
<td><strong>USB ports</strong></td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td><strong>GPIO Pins</strong></td>
<td>26</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td><strong>Power Requirements</strong></td>
<td>5V, 1A</td>
<td>5V, 2A</td>
<td>5V, 2.5A</td>
</tr>
</tbody>
</table>

for future add-ons or improvements to the robot. The model 2 boasts impressive computer power compared to previous models of Raspberry Pi. The model 2 contains a 900MHz quad–core ARM Cortex–A7 which is significantly more powerful than the single core processors on previous versions, which could allow for embedding of more computationally intensive algorithms.
Bibliography


