Creating Robotic Characters for Long-Term Interaction

by

Adam Setapen

B.S., University of Texas (2009)
M.S., University of Texas (2010)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2012

Author ....
Program in Media Arts and Sciences,
School of Architecture and Planning
September 1, 2012

Certified by ....
Cynthia Breazeal
Associate Professor
Thesis Supervisor

Accepted by ....
Prof. Patricia Maes
Associate Academic Head
Program in Media Arts and Sciences

Creating Robotic Characters for Long-Term Interaction

by

Adam Setapen

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
on September 1, 2012, in partial fulfillment of the
requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

Researchers studying ways in which humans and robots interact in social settings have a problem: they don't have a robot to use. There is a need for a socially expressive robot that can be deployed outside of a laboratory and support remote operation and data collection. This work aims to fill that need with DragonBot - a platform for social robotics specifically designed for long-term interactions. This thesis is divided into two parts. The first part describes the design and implementation of the hardware, software, and aesthetics of the DragonBot-based characters. Through the use of a mobile phone as the robot's primary computational device, we aim to drive down the hardware cost and increase the availability of robots "in the wild". The second part of this work takes an initial step towards evaluating DragonBot's effectiveness through interactions with children. We describe two different teleoperation interfaces for allowing a human to control DragonBot's behavior differing amounts of autonomy by the robot. A human subject study was conducted and these interfaces were compared through a sticker sharing task between the robot and children aged four to seven. Our results show that when a human operator is able to focus on the social portions of an interaction and the robot is given more autonomy, children treat the character more like a peer. This is indicated by the fact that more children re-engaged the robot with the higher level of autonomy when they were asked to split up stickers between the two participants.

Thesis Supervisor: Cynthia Breazeal
Title: Associate Professor

Creating Robotic Characters for Long-Term Interaction

by

Adam Setapen

The following people served as readers for this thesis:

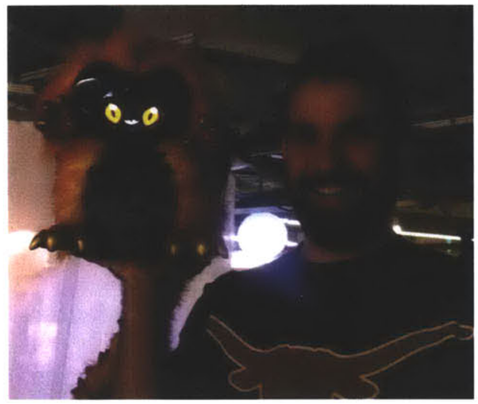
Thesis Reader . . . .

Rosalind W. Picard
Professor of Media Arts and Sciences
Program in Media Arts and Sciences

Thesis Reader . . . .

David DeSteno
Associate Professor of Psychology
Northeastern University

Acknowledgements



I've had the good fortune to work with some of the most talented scientists, engineers, designers, artists, and researchers in the world during my time at MIT. And I learned more in my two years here than I thought was possible. Building a robot is absolutely a team-effort, and the work in this thesis would not have been possible without the hard work, guidance, and support of many talented individuals.

First of all, I would like to thank my advisor Cynthia Breazeal for accepting me into the world-class group of researchers that make up the Personal Robots Group. Cynthia's vision for robotics has been an inspiration to me since I first discovered the field, and it's been an unforgettable two years learning from and working with her. I was stuck in software before I joined Cynthia's group, and she gave me the resources to become a capable roboticist.

Rosalind Picard's scientific expertise and hacker mentality give her a unique blend of practical knowledge and "outside of the box" thinking. It's been a tremendous pleasure to work with her during my time at the Media Lab, and our conversations have instilled a deep appreciation for the possibilities - and complications - of augmenting robots with emotional capabilities.

David DeSteno has been a wonderful mentor through his passion and expertise regarding social psychology. Without his time and effort in helping me analyze the results of this thesis, we wouldn't have found the really interesting information hidden in our data. And a big thanks to Paul Harris for his guidance on how to correctly run a user evaluation with children.

The MIT Media Lab has fostered a truly unique environment for creating, exploring, and learning. Joining the Personal Robots Group family has been amazing, and the members of the group have made my time at MIT unforgettable. First of all, thanks to

our group's administrator, Polly Guggenheim, for keeping us all smiling and laughing even during the most stressful of days. Polly, every time that I step into our lab I look forward to running into your office to see your smiling face.

Having Natalie Freed as a colleague, an officemate, and a friend has been truly awesome. She was instrumental in the creation and execution of DragonBot, and her unique insight into designing technologies for children was a driving force behind the inspirations of this thesis. Natalie's hard work in fabricating the "costumes" for our robots transformed them into truly believable characters.

Siggi, you showed me what it means to be a true roboticist. I don't know how someone your age has both the expertise and patience that you do, but you consistently set the example of how a researcher in our field should behave. David, your passion and work ethic were infectious from day one. You push yourself harder than most people I know, and it truly shows in your work. Nick, you're an awesome friend and an even more awesome researcher. Your enthusiasm for machine learning and your dedication to your ideals have been inspiring to be around, and I'll deeply miss our coffee breaks.

Kris, I've been constantly learning from you since the moment you showed me how to draw a rectangle in SolidWorks. Angela, you are an inexplicable reservoir of positive energy, and the robots group was blessed to have you for so many years. Peter, your fabrication knowledge combined with your keen artistic eye gave me something to strive towards (and you always could give me the right bit for the ShopBot). Jin Joo, it's been great sharing our passion for robotics together, and your generosity with your time and experience make you a blast to work with. Adam, your work skirts the line between art and engineering (pun intended), and your creations made me rethink how to incorporate electronics into my designs. Finally, thanks to Kenton and Phillip for your creative energy and expertise in robotics when they were most needed.

Fardad Faridi is a master character animator and designer, and his expertise and passion are what helped make my robots so expressive. Being able to work with an artist of Fardad's caliber made me rethink how to animate robots, and everything he touches seems to turn into gold. Without the guidance and mechanical expertise of Marc Strauss, this robot would have fallen apart (literally). Marc created the original delta prototype

and empowered me with the mechanical knowledge needed to turn his innovative design into a capable platform. Also, a huge thanks go out to Jesse Grey and Matt Berlin of IF Robots. Jesse and Matt's decades of experience in social robotics helped make this research possible, and their work on the Android framework was essential for turning the "crazy idea of a cellphone-powered robot" into a reality.

Taking Neil Gershelfeld's class, "How to Make (almost) Anything", was probably the most empowering experience of my life. Neil made me realize that I could tinker in the physical world the exact same way I knew how to do in software. Taking and TA'ing his course was a unforgettable part of my experience at MIT. Also, a huge thanks to Tom Lutz and John DiFrancesco for running the shop through the Center for Bits and Atoms, where I did the majority of my fabrication. Tom and John always exhibit tremendous patience when passing on their knowledge to students, and manage to maintain an awesome machine shop at the same time. And to everyone who helps the Media Lab run so fluidly, thanks for being the unsung heroes of the department. Linda Peterson, Kevin Davis, Greg Tucker, Paula Aguilera, and many more make this fantastic program and building fit the needs of its students.

It was incredibly rewarding to be around such a bright group of UROPs during my time at MIT. Special thanks to Sooyeon Jeong for her contributions to the real-time audio pipeline and to Daniel Gonzales for his countless hours soldering motor controllers. Also, a big thanks to Iliya Tsekov for his unparalleled skill in the machine shop.

Thanks to Peter Stone, my previous academic advisor, for preparing me for a graduate career at MIT. And I would be remiss if I didn't mention the gang of RoboCuppers from UT Austin, especially the fantastic Brad Knox. Brad's work ethic and overall awesomeness are something I aspire towards, and he's been an invaluable colleague and friend over the past few years (even if we're just screwing around making interactive pet toys). Thanks to Rachel Lewis, Leah Dickens, and Remi Torracinta for their enthusiasm and patience during operating the robots. And a big shout out to Ytai Ben-Tsvi for coming to the Media Lab and running a workshop on android hacking.

A huge thanks to all my friends. Cranor - you taught me what it means to be a real hardware hacker, and our nights in the green circle will never be forgotten. Edwina - you helped me get through the whirlwind of MIT more than you know. Darthur - the amount of support you give your friends is truly unparalleled. You stayed up late with me fixing things even when you had your own work, and your desire to tackle hard problems has been infectious. Akito and Lakatos - you guys are awesome roommates and colleagues, and now we can finally relax and enjoy our evenings. Mayton - your expertise in just about everything, and your generosity with your time, helped me so many times I can't even count.

Thanks to my older sister Ilana for setting an example of what it means to work hard and follow your passion. And last, but certainly not least, the biggest of thanks go out to my Mom and Dad for their unwavering support, guidance, money, trust, and love throughout my academic career.

Contents

Prologue **21**

1 Introduction **23**

1.1 Contributions 25

1.2 Thesis Overview 26

2 Background and Related Work **29**

2.1 Long-term Human-Robot Interaction 30

2.2 Social robots: A (very) brief overview 31

2.3 Teleoperation of Social Robots 33

2.4 Sharing Task 34

3 Hardware Design and Implementation **35**

3.1 Design goals 37

3.2 Character Design 38

3.3 Mechanical Design 41

3.3.1 Parallel Manipulator 42

3.3.2 Extra degrees of freedom 49

3.3.3 Motors 52

3.3.4 Head design 53

3.3.5 Usability 55

3.4 Electrical Design 56
3.4.1 Controlling a robot with a mobile device 58
3.4.2 IOIO 59
3.4.3 MCBMini motor controllers 61
3.4.4 SEEDpower 63
3.4.5 SenSkin 64
3.5 Aesthetic Design 65
3.5.1 Fur 66
3.5.2 Hands and Feet 69
3.6 Peripherals 70
3.6.1 Charging station 71
3.6.2 KinecTable 72
3.6.3 Travel case 73
3.7 Characters 74
3.7.1 Nimbus: A first character 75
3.7.2 Kombusto: A dragon is born 76
3.7.3 The whole gang: Neon, Pepper, Galli and Nitro 77
3.8 Limitations and Complications 79

4 Software Framework **81**
4.1 Core framework 82
4.1.1 Animation and Motor System 84
4.1.2 Emotion system 86
4.1.3 Synthesizing a lifeform 88
4.1.4 Speech systems 90
4.1.5 Blended Reality 90
4.2 Data collection and load balancing 92
4.3 Auxiliary Systems 92

4.3.1 Teleoperator speech pipeline 93
4.3.2 Augmentation through RGBZ sensor 93
4.4 Communication between modules 94

5 Teleoperator Interfaces **97**
5.1 Commonalities 99
5.1.1 Emotion widgets 100
5.2 Manual Controller 101
5.3 Shared Interface 102

6 Experimental Design **105**
6.1 Introduction 106
6.2 Relevant Measures 107
6.2.1 Cognitive load 107
6.2.2 Social presence 108
6.2.3 Engagement 108
6.2.4 Ease of use of the platform 108
6.2.5 Arousal 108
6.2.6 Child-dependent metrics 109
6.3 Method 109
6.3.1 Participants 109
6.3.2 Teleoperators 110
6.3.3 Task 111
6.3.4 Complicating factors 114
6.3.5 Operator Questionnaires 115

7 Results and Discussion **117**
7.1 Teleoperator's Perspective 118
7.1.1 Cognitive Load 121

7.1.2 Social Presence . . . 122
7.1.3 Engagement . . . 125
7.1.4 Ease of use of the platform . . . 125
7.1.5 Arousal . . . 127
7.1.6 Advice from the Operators . . . 131
7.2 Childs perspective . . . 132
7.2.1 Recall . . . 133
7.2.2 Sharing . . . 134
7.2.3 Re-engagement . . . 135

8 Conclusion **139**
8.1 Contributions . . . 140
8.2 Future Work . . . 141
8.3 Concluding Remarks . . . 142

Appendices **143**

Appendix A Custom Electronics **145**
A.1 SEEDpower power management . . . 145
A.2 SenSkin capacitive sensing . . . 147

Appendix B Linkage Design **151**

Appendix C Study Materials **157**
C.1 Interaction script . . . 157
C.2 Questionnaires . . . 161
C.3 Pictures of robots . . . 172

List of Figures

1-1 *Kombusto*, the first dragon-inspired character created around the platform 25

3-1 Kinematic concept model for original prototype (V1) 36
3-2 DragonBot solid-body model for initial prototype (V1) 36
3-3 Early concept renderings of dragon-inspired characters by Fardad Faridi . 39
3-4 Rigging an animated character with a model of the parallel platform . . . 40
3-5 DragonBot solid-body model for V5 prototype 41
3-6 DragonBot solid-body models for V2, V3, and V4 prototypes 42
3-7 DragonBot prototypes (from left): V1, V3, and V4 42
3-8 Original Deltabot diagram (left) and the Adept Quattro pick-and-place (right) 44
3-9 Kinematics diagram of our modified delta manipulator 45
3-10 Range of motion - extrema on interaction sphere 46
3-11 A pair of precision $\underline{R}$-$(SS)^2$ linkages 49
3-12 The DragonBot V5 prototype without any aesthetic elements 50
3-13 The range of motion for DragonBot's neck tilt 51
3-14 An easily-manufacturable cable-driven tail 52
3-15 Primary motors for DragonBot V4 - V5 53
3-16 A flip-out style cassette deck 53
3-17 Head assembly for DragonBot V5 54
3-18 Control Panel with charging port and power switch 55

3-19 Mechanical mounts for hands and fur (left), and magnetic mounts for the wings (right) . . . 56
3-20 Faceplates designs for different character types (concept and modeling by Fardad Faridi) . . . 57
3-21 The mobile devices used for the DragonBot prototypes (from left to right): Samsung Galaxy Epic (V1), iPod touch (V2), HTC Thunderbolt (v3), HTC Evo 3D (V4 and V5) . . . 57
3-22 DragonBot electrical diagram . . . 58
3-23 The IOIO board for bridging android and embedded systems . . . 60
3-24 The MCBMini control scheme . . . 61
3-25 The SEEDpower board for battery isolation, power regulation, and smart-charging . . . 64
3-26 The SenSkin board for capacitive sensing and driving LEDs . . . 65
3-27 3D printed parts for DragonBot (clockwise from upper left): cranium (FDM), head (FDM), hand (MJM), foot (MJM), faceplate (MJM) . . . 66
3-28 Prototyping squash-and-stretch exteriors for the robot . . . 67
3-29 A prototype of a fur augmented with textile-based capacitive sensors . . . 68
3-30 Dying a synthetic two-way stretch fur . . . 68
3-31 Foot prototype, urethane-iron hybrid material . . . 69
3-32 DragonBot V4 prototype with 3D printed aesthetics . . . 70
3-33 7-piece hand mold from 3D printed acrylic, along with a silicone-cast blue hand . . . 71
3-34 Charging Station for DragonBot . . . 72
3-35 Design of KinecTable for augmenting DragonBot's capabilities . . . 73
3-36 Travel Case for DragonBot . . . 74
3-37 *Nimbus*, an initial character for testing the expressivity of the parallel manipulator . . . 75

3-38 *Nimbus* and a group of preschoolers 75
3-39 *Kombusto*, the first dragon-inspired character (photograph courtesy of Andy Ryan) 76
3-40 Neon, a character for tutoring french designed by Natalie Freed. 77
3-41 Pepper - the character used in the primary user evaluation for this thesis 78
3-42 Galli and Nitro, the Dragon Twins 79

4-1 The core DragonBot interface 83
4-2 Example facial animations for DragonBot, including expressions, eye positions, and visemes 85
4-3 The structure of DragonBot's Emotion system 87
4-4 The modules and hierarchies that comprise DragonBot's lowest-level behaviors 89
4-5 Kombusto as a fully-virtual character 91
4-6 Communication scheme between all the working systems 95

5-1 The tablet-based layout common to both teleoperator interfaces 99
5-2 The valence-based scale of emotional states available to the operator . . 100
5-3 A screen-grab of the DragonBot Manual Controller 101
5-4 A screen-grab of the DragonBot shared interface 103
5-5 The way actions and emotions map to expressive motions within the **Shared Interface** 104

6-1 Age distributions of the participants 110
6-2 The interaction setup with the robot 112
6-3 The five animal stickers used in the experiment 113
6-4 The wide-angle view of the interaction 114

7-1 Video frame from a teleoperator synchronizing the EDA sensor (left) and during interactions (right) . . . 119
7-2 "I felt comfortable during the interaction" . . . 121
7-3 "The other individual paid close attention to me" . . . 124
7-4 "The other individual tended to ignore me" . . . 125
7-5 "How relaxing or exciting was the experience" . . . 127
7-6 EDA from operator #1 . . . 128
7-7 EDA from operator #2 . . . 128
7-8 EDA from operator #3 . . . 129
7-9 Pepper and a child sharing attention . . . 131
7-10 Durations of interactions for both conditions . . . 132
7-11 Tangible interaction with Pepper . . . 133
7-12 Recall percentages for both conditions . . . 134
7-13 A child smiling after sharing 3 stickers with Pepper . . . 135
7-14 Child re-engaging robot during sharing . . . 136
7-15 Two participants re-engaging the robot about the overlapping sticker . . . 137

8-1 Various images from the robot's vision stream . . . 141

A-1 A group of 5 SEEDpower boards ready to be put into robots . . . 145
A-2 Schematics for the SEEDpower management board . . . 146
A-3 2-layer board design for the SEEDpower management board . . . 147
A-4 Schematics for the SenSkin capacitive sensing board . . . 148
A-5 2-layer board design for the SenSkin capacitive sensing board . . . 149

List of Tables

6.1 Teleoperator backgrounds . . . 111

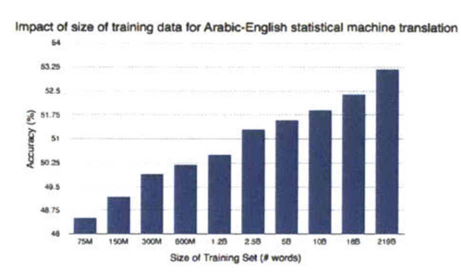
7.1 General questions . . . 120
7.2 Questions comparing the two interfaces . . . 120
7.3 Questions from the NASA-TLX task load index . . . 122
7.5 Questions regarding engagement . . . 126
7.6 Questions about the platform's viability . . . 126
7.7 Results from analysis of electrodermal activity (EDA) . . . 130
7.8 Grouped analysis of electrodermal activity (EDA) . . . 130
7.9 Results from informal post-survey for participants . . . 133
7.10 Sharing 2 vs. 3 stickers . . . 134
7.11 Sharing of overlapping sticker . . . 135

Prologue

Imagine what would happen if you had a robot in your home that was on all the time. What types of things would it see?

Rodney Brooks posits that robotics will be solved when our machines have the *vision* capabilities of a 2-year old, the *language* capabilities of a 4-year old, the *manual dexterity* of a 6-year old, and the social skills of an 8-year old [1]. I agree.

Large vision corpora are becoming widely available, and it has been shown that the sheer quantity of data can be one of the keys in improving the performance of a machine learning algorithm [2]. As for language, the Internet provides a lifetime's worth of text. Every *minute*, 204 million emails are sent, Google receives over 2 million queries, and Twitter users send over 100,000 messages [3]. Throwing more data at language-based tasks has consistently yielded significant performance gains [4], [5]. And *manual dexterity*? Decades of motion capture - now being supplemented with data from cheap 3D depth sensors - have advanced human trajectory planning. And the electromechanical systems are improving every day.



However, where is the data on *social interaction*? Where is the corpus needed to bootstrap this essential sub-problem, which takes children *double* the amount of time needed to grasp their first language? It simply doesn't exist.

And *that's* what inspired this thesis. We need data from real-world social interactions between people and robots. But in order to even start getting the data, a new type of robot had to be built...

Chapter 1

Introduction

Most people do not have the opportunity to interact with robots "in the real world". For those people who do, such robots are typically not designed to be socially capable. When researchers in the Human-Robot Interaction (HRI) community want to learn more about how humans treat robots in social settings, the standard practice is to run a laboratory-based experiment. Human subjects come into the lab, they interact with a robot for 10 or 15 minutes, and then they leave. The subjects are often not completely comfortable in an unfamiliar setting, and there is no possibility for sustaining interactions over longer timescales.

A need exists in the HRI community for a low-cost and expressive platform that is able to support long-term longitudinal studies in the field. This type of robot would enable researchers to deploy more agents outside of the laboratory, leading to a better understanding of how people interact with robots in everyday settings. The robot must be low-cost to facilitate widespread use, and it needs to have a persistent Internet connection. This is important because if we want to really support long-term longitudinal interactions, more and more data analysis will need to happen "in the cloud". The ability to offload computationally expensive problems to cloud-based services has already started stimulating the robotics community [6]. And with the robot's Internet connection, it can be operated by an ordinary person from a remote location.

To address this need, we decided to create an expressive platform for robotics research that is able to support long-term interactions with children outside of the laboratory. And the way we were able to accomplish our goal? Using the robot in your pocket.

The modern smartphone combines a capable touchscreen with a suite of sensors and a powerful processor. These devices also have continuous Internet connections, enabling the use of cloud-based services for solving computationally expensive problems. As robotics matures, more large-scale and powerful analytics will take place "in the cloud", and harnessing a smartphone is the best current method to make use of these emerging paradigms. Furthermore, use of an existing device drives the cost of the robot down, and makes the barrier to entry much lower.

Therefore, this document describes our robot specifically designed for longitudinal interactions outside of the laboratory. A robot that could start generating the amount of data needed to make fundamental advancements in our understanding of social dynamics. A robot that is easy for humans to control, and can exhibit a personality of its own. A robot that can use cloud-based services to distribute the challenge of existing "in the real world". This thesis presents the exploration of building this robot, DragonBot, and some first impressions from data captured from interaction with 4-7 year old children in a social sharing task.

There are a number of challenges that must be overcome to build a robot capable of long-term interactions. An important one is **engagement** - both parties in an interaction must feel like involved social participants. Another is **social presence** - the degree to which people perceive human-like characteristics in the robot [7]. Finally, and probably the most difficult from a technical standpoint, is the ability to partake in interpersonal interaction at human timescales that sustain the child's interest, supporting and fostering a young child's ability to engage the robot as a social actor.

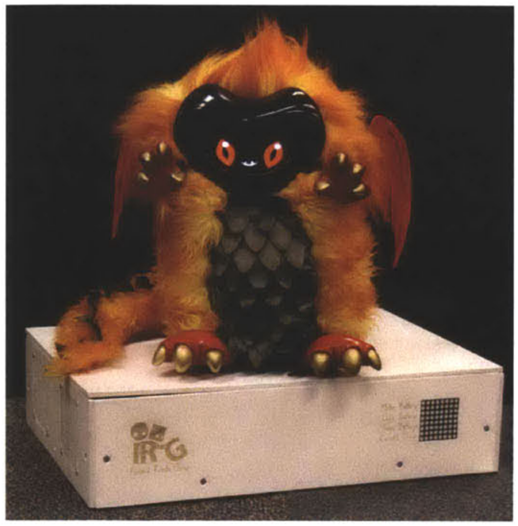


Figure 1-1: *Kombusto*, the first dragon-inspired character created around the platform

1.1 Contributions

This thesis is divided into two parts, each focusing on a set of contributions:

Technical contributions

1. The design and implementation of an expressive social robot with strong appeal to children that can exist outside of the laboratory and support HRI research in the field.
2. A teleoperator interface for non-expert users that enables a person to dynamically shape the robots interaction during a conversation.
3. A data collection framework that can capture a range of relevant data to support HRI studies in the field, including video, audio, 3D depth maps, electrodermal activity, and robot commands, actions and states.

Use case contributions

1. What do kids think about the robot? What do they respond to?
2. What do the human operators think about the control mechanisms?
3. How can different levels of autonomy in teleoperation affect the behavior of children?

To evaluate these contributions, a comprehensive human-subjects study was conducted with 28 children between the ages of four and seven. Because the multimodal data captured in this study spanned over 1.1 Terabytes, this thesis focuses on a few key evaluation metrics obtained from the interactions. However, these initial results from testing the robot illustrate that DragonBot is a powerful and expressive platform for supporting longitudinal research in the field.

1.2 Thesis Overview

This thesis is structured as follows:

Chapter 1, Introduction Informs readers about the overarching goals of this thesis, outlining the structure of the document and enumerating the core contributions of this work.

Chapter 2, Background and Related Work Establishes definitions and frames the research within relevant literature. This chapter also informs readers about previous work in similar directions.

Chapter 3, Hardware Design and Implementation A detailed description of the mechatronic design, character development, and fabrication of the DragonBot hardware platform, as well as the aesthetic to support its characters. This chapter is heavily systems-focused.

Chapter 4, Software Framework An in-depth description of the software architecture that the DragonBot and its auxiliary systems.

Chapter 5, Teleoperator Interfaces A comparison of two different teleoperator interfaces designed for capturing data using DragonBot characters.

Chapter 6, Experimental Design An analysis of two different ways of controlling the robot, comparing a direct manual control interface with a system in which autonomy is shared between human and robot.

Chapter 7, Results and Discussion A detailed analysis and discussion regarding the data-capture evaluation study.

Chapter 8, Conclusion A summary of the contributions of this work, and ideas for future research directions using DragonBot.

Appendix A, Custom Electronics A detailed look into the custom electronics designed for DragonBot.

Appendix B, Linkage Design The mechanical drawings for DragonBot's mechanical linkages.

Appendix C, Study Materials Supplementary materials from the user evaluation, including questionnaires, interaction protocols, and drawings of robots from the participants.

Chapters 3 - 5 contain the technical contributions of this thesis, while Chapters 6 and 7 focus on the contributions from a real-world use case.

Chapter 2

Background and Related Work

Kanzi and some other "enculturated" apes are different. Why? For precisely the reasons you articulate for children: from an early age Kanzi has spent his ontogeny building up a shared world with humans – much of the time through active negotiation. An essential element in this process is undoubtedly the behavior of other beings – i.e., human beings – who, on a daily basis encourage Kanzi to share attention to objects with them, to perform certain behaviors they have just performed, to take their emotional attitudes towards objects, and so on and so forth. *Apes in the wild have no one who engages them in this way – no one who intends things about their intentional states.*

–Michael Tomasello [8]

Pulling in work from the fields of human-robot interaction, telepresence, and human social psychology, this research has been informed through many different lenses. We first discuss long-term interaction with socially-embodied agents, followed by a look at some recent work in designing sociable robots, teleoperation, and shared autonomy. The section is concluded by discussing the prior work regarding the reward allocation behaviors of pre-school aged children.

2.1 Long-term Human-Robot Interaction

One of the first explorations into long-term interactions between humans and robots is CMU's "Roboceptionist" [9]. Although the robot was simply a virtual character's face displayed on a monitor behind a desk, it was the first example of an "always-on" robot intended for long-term social interactivity. After nine months of operation, the authors concluded that visitors often return to interact with the robot on a daily basis, but that few of these interactions last for more than 30 seconds. While the authors argue that endowing their robot with personality, character, and a story that changes over time will keep people interested, the range of expressivity of the "Roboceptionist" is severely limited. Because the robot isn't physically embodied, it cannot exhibit any of the non-verbal behaviors that are so important to the social dynamics of an interaction [10].

Another platform built by the roboticists at CMU is Snackbot - a mobile robot that roams the halls distributing cookies and chips to hungry humans [11]. Although the motivations behind the design of Snackbot are for studying long-term HRI, the robot is primarily a service robot and not a social one. Here at MIT, Cory Kidd built Autom, a socially assistive weight loss coach intended for long-term support [12]. He ran a six-week study that compared the weight-loss habits of subjects interacting with the social robot compared to an analogous non-social software system. He found that participants track their calorie consumption and exercise for nearly twice as long when using the robot than with the purely functional equivalent.

Hiroshi Ishiguru's research group has started to look at the role of social robots in a classroom setting [13]. In a 2007 experiment, their Robovie platform autonomously interacted with elementary-school children in a two-month field trial. Although no significant results were determined from this paper, it informed the research to move in a positive direction. The most recent study from this line of research again used the Robovie platform to teach 6th grade students about Lego Mindstorms [14]. However, this time the robot was teleoperated to teach the children seven lessons, with the human operator exhibiting either social behavior or a purely functional equivalent. Their findings show that the social interactions encouraged children to work more in the first two lessons, but did not affect later performance.

Long-term studies have shed interesting light on socially assistive *virtual* agents, where the hardware problem is sidestepped. In a month-long field trial, Bickmore and Picard demonstrated that users prefer to continue interacting with an agent that "cares" over a functional one that does not exhibit any supportive behavior [15]. Furthermore, this line of research indicates that an agent with social-emotional or relationship-building skills is respected more, liked more, and trusted more than an agent that does not exhibit these traits [16].

2.2 Social robots: A (very) brief overview

There are a number of robots designed for children that have begun to appear over the past few years. Tofu is a small teleoperated robot for children that engages them in collaborative play through expressive motions. Tofu's low cost and beautifully simple design make it an appealing platform for studying interactions between children and robots [17]. However, Tofu has no sensory input, no data-collection capabilities, and no wireless connection, making its deployment outside of the laboratory a problem. Keepon was another socially-expressive robot for kids that suffers from the same problem - deployment outside of the laboratory requires a lot of supplementary equipment [18]. The Huggable project, a robotic teddy bear for children, investigated ways in which touch-sensitive areas on the robot could be used to increase the affective behavior of the robot [19]. While the Huggable project was recently redesigned to be a mobile self-contained platform, the original teleoperated prototype for therapeutic purposes required four external computers and two large power supplies to run [20].

Juan Velasquez's started his insightful look into emotions for virtual agents in 1997 [21], with a flexible and extensible way of modeling the influence of emotion on the behavior of the agent. His method takes into account aspects of both the expressive and experiential components of emotion. Culminating in 2007 [22], Velasquez's "affect programs" were based not upon the desired external behaviors of the robots, but rather on the set of prototypical fundamental situations that the robots will encounter, viewing these as affective situations for which a set of coordinated responses can provide solutions. While this work was primarily evaluated with virtual agents, the social robot Kismet was a pioneering example of how emotion can be successfully integrated into a robot's behaviors to facilitate an appealing social interaction [23].

Roboticists have been using cell phones to augment the capabilities of their plaforms since as early as 2001 [24]. However, when the development of DragonBot started roughly 18 months ago, there were only one or two hacked-together hobby projects that integrated a mobile device as the robot's sole high-level controller. The Amarino toolkit was the first research example of bridging an embedded microcontroller with a mobile device. Amarino connects the popular Arduino environment with Android devices, using a simple message passing protocol over bluetooth [25] to allow phone-to-embedded communication. In the past year, the popularity of smartphone-based robots has skyrocketed, partly due to trends in cloud-based computing. These are primarily appearing in industry, although the applicability of such a system is certainly suitable for robotics research. Some examples of these phone-based robots appearing as products are Cellbots [26], Romo [27], and Botiful [28].

2.3 Teleoperation of Social Robots

Teleoperation research dates back to the 1950's, when interfaces started being developed to assist people working in unsafe conditions. One of the best known early examples of teleoperation is Raymond Goertz's design of a robotic arm that was capable of being operated at a distance in order to safely handle radioactive materials [29]. Interestingly, this was also the first haptic interface, as a passive model of the manipulator was manually positioned by a human to control the robot's configuration. Teleoperation is still widely used in this same manner - as a tool for allowing people to work in difficult situations [30]. Recent work in teleoperation has enabled augmentation of humans capabilities, not just safety mitigation, such as healthcare [31] and robot-assisted surgery systems [32].

Time delay is the most important factor in robot teleoperation [33], [34]. This delay is particularly problematic for conversational agents, as being able to communicate at human timescales is essential to achieve a socially-dynamic interaction. However, teleoperation has been successfully used for control of a robot in social situations, and most robots designed for interaction with children use a human operator [35], [20]. While simultaneous teleoperation of multiple robots is a common theme in robot navigation, Glas et al. have started to bring this concept to social robots. In 2008, they designed an interface for operating a single robot while monitoring several others in the background, with very fast context switching between robots [36].

Shared autonomy is a way of generating a robot's behaviors in which a human operator and robot both affect the control. One of the best examples of a social robot that shares autonomy is Probo [37]. Probo is a therapy robot meant for interaction with hospitalized children, using shared autonomy to control the robot's attention, emotional state, and generation of life-like motions. Shared autonomy is a technique that is used in many fields other than robotics, such as medical diagnosis [38] and recommendation systems [39].

2.4 Sharing Task

Reward allocation, commonly known as sharing, is an active topic in the behavioral psychology community ([40], [41]), specifically relevant to social rapport for pre-school aged children (3-5 years) [42]. A reward allocation task consists of a set of items (often stickers or something else appealing to children) that a child splits up between themselves and one or more participants. Often these studies use an odd number of items, as this forces the children to split up the stickers without simply giving half to themselves and half to others [43], [44]. In the late 1950s, Handlon ran an experiment comparing the allocation behavior of 4 - 16 year olds [45], and since that time the focus has been on younger children.

Lane and Coon studied the reward allocation behavior of 4 and 5-year old children [46] using stickers as rewards. They ran a study where children worked on a task with a partner whose performance they observed to be superior, equal, or inferior to their own. The children were then given rewards to distribute between themselves and their partner. The results showed that 4-year-olds tended to be self-interested with reward distribution, taking significantly more than half for themselves. However, 5-year-olds allocated the rewards consistently with the performance of the two participants during the task.

It's been shown that the sharing behavior of children starts to become consistent with that of adults when kids are only 3.5 years old [47]. Also, resource-allocation decisions made by young children are dependent on the recipient [48], with children preferring friends over strangers, and both over non-friends. Warneken et al. found that 3-year-old children share mostly equally with a peer after they have worked together actively to obtain rewards in a collaboration task, even when those rewards could easily be monopolized [49].

Chapter 3

Hardware Design and Implementation

I really feel like many of the tricks we try to use as animators for creating appeal and life could be directly applied to robots to great effect.

–Doug Dooley

This chapter describes the design and implementation of the DragonBot hardware systems, as well as the suite of dragon-inspired characters built around the underlying platform. Five prototypes were designed and built, and this chapter details the iterative design process that informed the creation of DragonBot. Four models of the final prototype, V5, were built in order to bootstrap a mobile field laboratory for longitudinal HRI.

DragonBot's beginnings are rooted in an innovative inverted parallel manipulator designed by Marc Strauss. As soon as Marc powered on his "Quadrobot" and it began to cycle through it's range of motion, it became apparent that this manipulator offered an unparalleled mechanism for **animating off the screen**. That is, the expressivity possible from this parallel mechanism was enough to provide the basis for an "illusion of life" - the property of inanimate objects that appear to have agency. The flour-sack assignment, which brilliantly illustrates the principle of the illusion of life, is an example given to first-year students in animation school. They are told to make an animation of a flour-sack exhibiting emotion. The purpose? If you can convey emotion with a flour sack, you can convey it with anything. Seeing this parallel manipulator in motion was all the inspiration needed to turn this mechanism into a character, because it could accomplish the flour-sack example - in physical space.

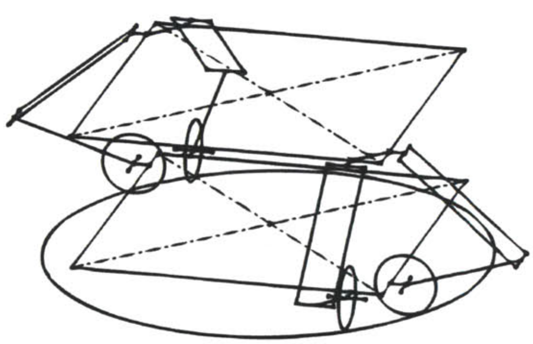


Figure 3-1: Kinematic concept model for original prototype (V1)

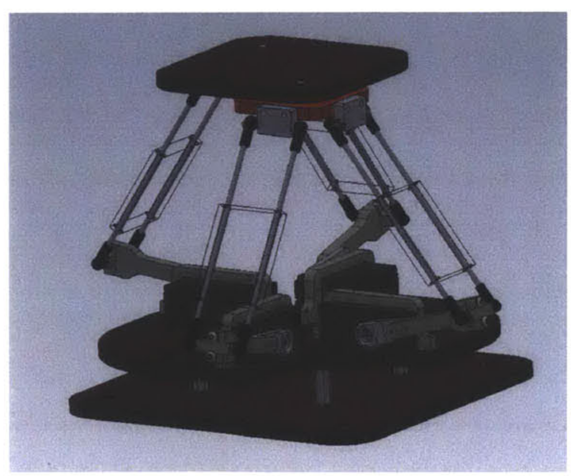


Figure 3-2: DragonBot solid-body model for initial prototype (V1)

3.1 Design goals

As outlined in section 2, there are simply no social robot platforms in existence that can support long-term and longitudinal interactions in the field. Building a social creature is a holistic process, informed by a multitude of diverse disciplines that collide to inform the design of a cohesive character. One can't simply tackle the hardware, the electronics, the design, and finally the software. All of these systems are interrelated, and they must all be considered from the beginning.

The design of DragonBot was informed and driven by classical and modern animation principles, as a long-term companion must be a believable character. The primary concerns we kept in mind during the iterative design process of DragonBot were:

- The ability to support long-term interactions
- The capabilities for remote field testing, operation, and data capture
- A cohesive design of a social character
- An extensible of behavior system
- Low-cost

Mechanically, the final DragonBot prototype has six physical degrees of freedom (DOFs). Four of these DOFs (cartesian X, Y, and Z + head rotation) come from our novel parallel manipulator that connects the stationary base of the robot to a "floating head" platform. The fifth degree of freedom is a neck-tilt on the robot's head, allowing the creature to look down at objects on a table or gaze up at people. The sixth DOF is a wagging tail to increase the perceived animacy of the robot. However, the face of a social robot is certainly the most expressive and important feature [50]. To address this issue, we use the screen of the mobile device as an animated face for the character. The smart phone is "inserted" into the head of DragonBot, bringing the robot to life through the use of over 180 unique virtual degrees of freedom.

All actions DragonBot takes are implemented using procedurally blended animations (described further in Section 4.1.1). This enables the user to take the phone out of the robot's head and still interact with the character through a virtual agent. This technique - called *blended reality* [51] - is paradigm for DragonBot's long-term interaction abilities, because we can allow users to interact with a virtual character when they don't have access to the physical robot. The phone's integrated display, camera, microphone, and speaker allow for an engaging interactive agent, regardless of the state of the character. While this thesis does not focus on blended reality, we feel it is an important concept for supporting long-term interactions with characters centered around mobile devices. We discuss the blended reality aspects of DragonBot in Section 4.1.5.

3.2 Character Design

We will now discuss the high-level animation techniques that informed the electromechanical design of DragonBot. Taking inspiration from the pioneering animation techniques of Disney's Thomas and Johnston [52], we analyzed the twelve principles of animation when designing our platform. Just as these principles paved the way for advances in modern computer animation [53], we feel they are essential for bringing a socially-embodied robot to life. Although many of these conventions have started to inform the user-facing behaviors of robots in recent years [54], we still feel they are often overlooked during the **design** process, and therefore merit discussion at this time.

The most important principle of animation is *squash and stretch*, which provides an object with a sense of weight and flexibility without changing the volume of the object. Taking inspiration from Tofu [17] and KeepOn[18], DragonBot's fur attaches to the top and bottom platform of the parallel mechanism, so vertical movement is organic, like a balloon being smothered and pulled. Many fur prototypes were developed as the design of DragonBot matured, in order to support this notion of squash and stretch. The design and fabrication of the robot's fur is discussed at length in Section 3.5.1.

Figure 3-3: Early concept renderings of dragon-inspired characters by Fardad Faridi

Another very important principle is *staging* - being able to clearly direct the audience's attention to an object. One envisioned use for DragonBot is second-language learning, so it is fundamental that the robot be able to direct attention to objects in the environment even when verbal communication might be a barrier. The DragonBot mechanical platform supports rotation up to 70 degrees off-center in each direction, so the robot's gaze can attend to the majority of a tabletop. The neck tilt added in the V4 prototype allows the robot to look down at objects on a table or up at people standing overhead. The virtual face also gives us tremendous flexibility with *staging* - the robot can easily look around the environment or even explicitly display what it is thinking about using the device's screen.

The concepts of *follow through and overlapping action* give objects a greater sense of motion by having passive elements that continue to move after the skeletal movement has stopped. To convey this sense of continuity, the final characters have floppy silicone hands, flexible dragon-wings, and flowing fur on their head. The exaggerated passive motion of these elements during acceleration and deceleration of the platform adds to the suspension of disbelief experienced by the user.

The principle of *secondary action* is a way of strengthening a primary action with secondary attributes. The most important thing about this principle is that the secondary actions emphasize the main action, rather than take attention away from it. This principle is addressed by the novel gaze and emotion systems of the robot, which will be described in Sections 4.1.1 and 4.1.2. These systems allow the robot to perform an action while looking anywhere in its range-of-view, and to perform an action in a certain emotional state with unique outward behavior.

Building upon the framework of the Synthetic Characters and Robotic Life groups at the MIT Media Lab [55], the software animation system addresses *straight-ahead action and pose-to-pose action*, *slow-in and slow-out*, and *timing*. The principle of *arcs* reflects the fact that animal actions typically occur along arched trajectories. This principle is particularly easy to harness with our parallel manipulator - creating parabolic motions is trivial because the platform can move in all three translational directions. Finally, the *appeal* of the character is analogous to charisma in an actor. An appealing character is believable, interesting, and engaging.

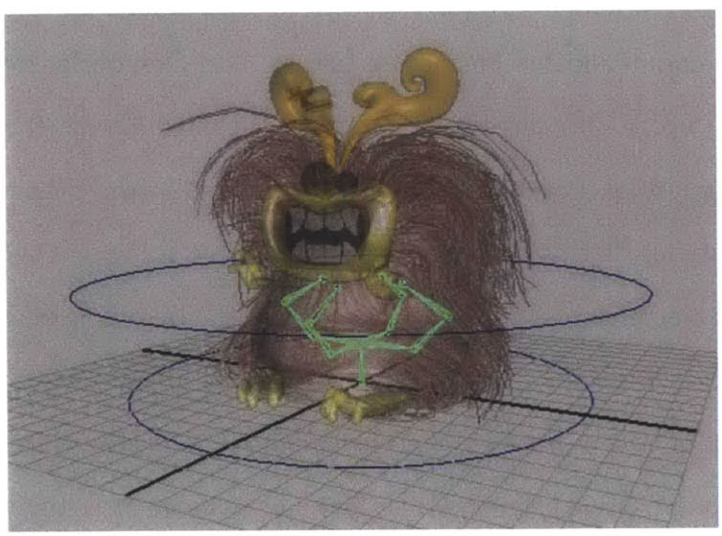


Figure 3-4: Rigging an animated character with a model of the parallel platform

The design of the robot - along with the software principles that bring it to life - were strongly influenced by the pioneering character designs of Disney. Also, the innovative puppeteers and crafters at Jim Henson's creature shop informed this work through their expert control and attention to detail when dealing with physical characters. Finally, the brilliant characters in many of Pixar's movies, such as Monsters, Inc. and Wall-E, inspired the deep integration of emotion into the core behavioral framework, which will be discussed in Sections 4.1.2 and 4.1.3.

3.3 Mechanical Design

This section provides a detailed analysis into the mechanical design and fabrication of DragonBot. Five separate prototypes were designed and built, transforming the innovative "Quadrobot" prototype (seen in Figure 3-2) into the capable platform for character animation seen in Figure 3-5. Three of the intermediate designs (V2, V3, and V4) can be seen in Figure 3-6.

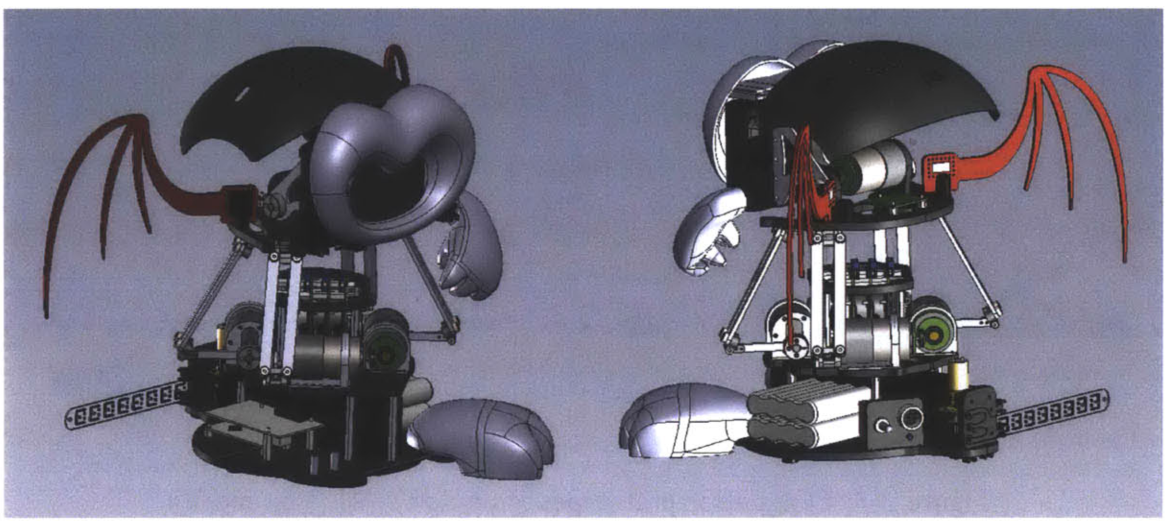


Figure 3-5: DragonBot solid-body model for V5 prototype

The mechanical portions of the robot were designed using solid-body parametric modeling in SolidWorks (see Figures 3-2, 3-6, and 3-5). The aesthetic elements, such as the hands and feet, were modeled in Maya and Rhinoceros. The most challenging aspect of the mechanical design was its very compact nature. The mechanics of the robot's main manipulator (described in the following section) restrict the useable areas to house non-mechanical components. Also, because the phone-as-face concept is so fundamental to the robot, a functional solution for connecting the main mobile device to the internal electronics (see Section 3.3.4) took a considerable amount of design effort and iteration.

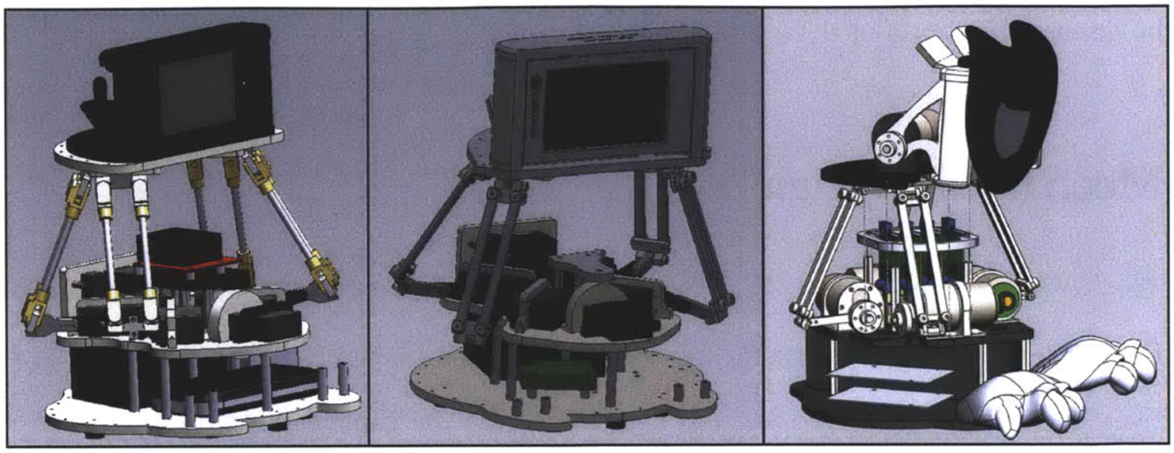


Figure 3-6: DragonBot solid-body models for V2, V3, and V4 prototypes

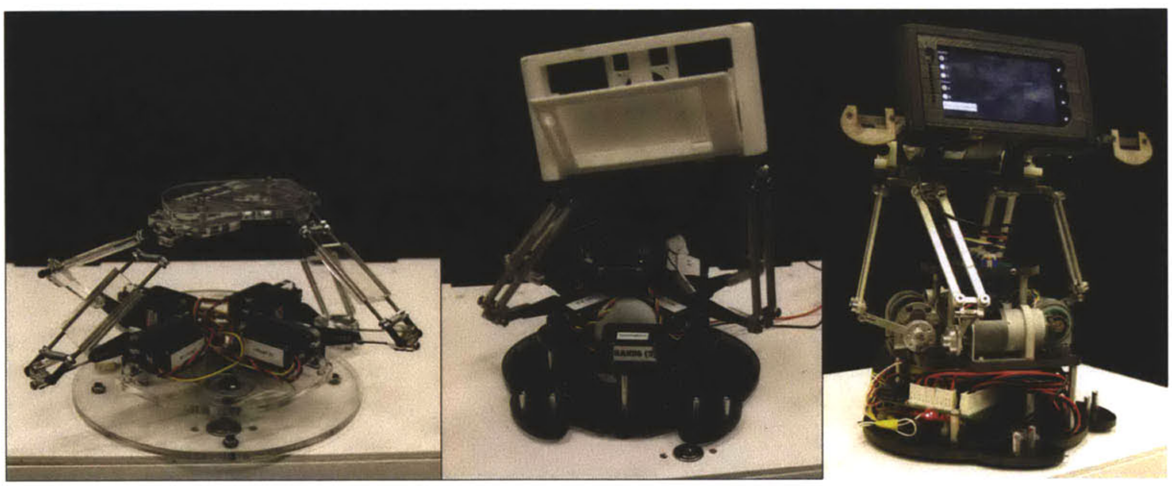


Figure 3-7: DragonBot prototypes (from left): V1, V3, and V4

3.3.1 Parallel Manipulator

The mechanical design of the robot centers around a parallel manipulator - a closed-loop kinematic mechanism whose end-effector is linked to a base by several independent kinematic chains. Parallel mechanisms are abundant in biology. From tissue arrangements to the mechanism by which an animal supports itself on legs, a parallel mechanism is simply one where several constraints simultaneously determine the relations of one body relative to another.

Parallel manipulators have higher accuracy than serial manipulators, support precise force sensitivity, and have the advantage of housing all motors on a stationary platform [56]. The heightened force sensitivity is well suited for physical interactions with children, where the forces presented may be minimal but the robot must exhibit some form of compliance in order to sustain the illusion of life. One drawback of parallel manipulators is that they inherently contain singularities - configurations of a mechanism where the subsequent behavior cannot be predicted or becomes nondeterministic. While particularly daunting in theory, these singularities can be minimized and handled in the physical world with a carefully informed design.

The best-known parallel robots are the Stewart platform and the Delta robot. The Stewart platform, which contains six linear actuators, has a platform that can be moved in the six degrees of freedom for which it is possible for a freely-suspended body to move. These are often implemented using hydraulic jacks for actuation, and therefore are widely used in configurations where extremely large loads must be supported. The simplest Delta robot has a platform that can be moved in all three cartesian directions, with the constraint that the stationary and mobile platforms are always parallel. This mechanical behavior is typically achieved using a system of mechanical linkages with joints that tightly couple the actuators with the moving platform. This simplest configuration uses three parallel $\underline{\mathbf{R}} - (\mathbf{SS})^{\mathbf{2}}$ chains - an actuated revolute joint ($\underline{\mathbf{R}}$) with a four-bar linkage made up of four spherical joints ($(\mathbf{SS})^{\mathbf{2}}$).

To extend this simple Delta robot with rotation of its actuator, the traditional solution is to add a fourth "telescopic" leg that rotates the mobile platform. However, recent research into parallel manipulators has shown that a fourth $\underline{\mathbf{R}} - (\mathbf{SS})^{\mathbf{2}}$ chain can been added to achieve the exact same rotational movement [57]. One advantage of this configuration is that it can be completely free of internal singularities, although external forces can still affect the stability of the motion. This four-chained $\underline{\mathbf{R}} - (\mathbf{SS})^{\mathbf{2}}$ mechanism is at the core of the mechanical design, with one caveat: the mechanism is upside down.

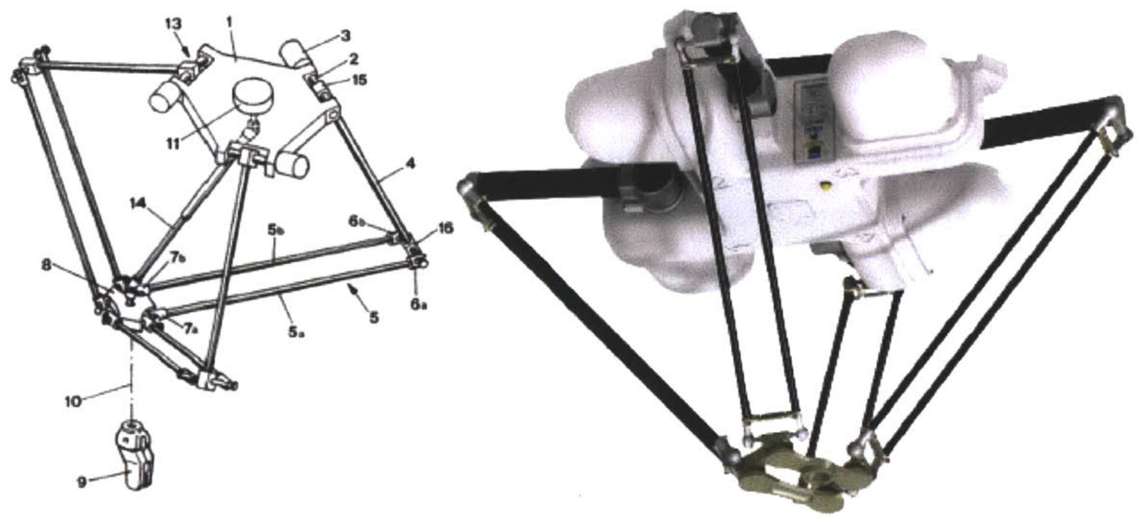


Figure 3-8: Original Deltabot diagram (left) and the Adept Quattro pick-and-place (right)

The platform that controls the motion of the robot's head is the mobile end-effector and the stationary platform (where the robot's feet are) can be placed on any surface. This gives us a 4 DOF upper platform that can translate motion in all three directions and rotate around the robot's dorsal axis. A high-level representation of the robot's mechanical configuration, including a set of dimensions that was calculated to stabilize the small platform's motion, can be found in Figure 3-9.

A Delta robot contains all of its motors in the stationary base, minimizing the mass of the moving parts and making the robot extremely efficient. Delta robots are able to support high velocities and high accelerations because the single mobile platform is driven by the multiple actuators working in concert. Because these manipulators are extremely fast and work best with minimal loads, Delta robots are primarily used in industry for pick-and-place operations. In these settings, the robots are mounted from the ceiling and the mobile platform below is able to pick up and put down components with extreme precision. This limits the working range of the robots, so a conveyor belt is often placed underneath the robot with a tightly coupled control system.

One of the biggest hurdles in the implementation of DragonBot was that parallel manipulators typically don't carry a large load, and they are mounted upside down where

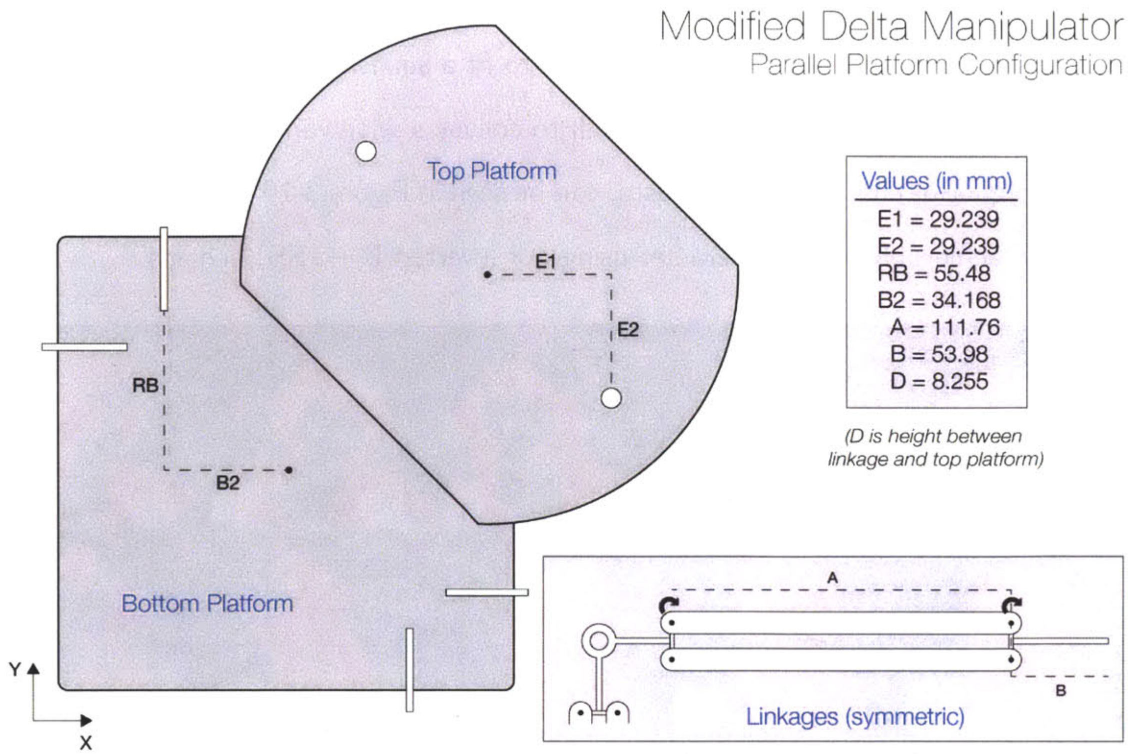


Figure 3-9: Kinematics diagram of our modified delta manipulator

the effects of weight and gravity are minimized. To achieve the stability we needed for the mobile platform of the parallel mechanism, it was necessary to center the weight as much as possible on the mobile platform. The singularities in this parallel manipulator occur when the rank of the Jacobian matrix mapping from joint space to Euclidian space decreases below 6 [58]. External forces are the main cause of these singularities, which is problematic in DragonBot because the mobile platform is carrying the weight of a smart phone. We want the robot to be able to have "snappy" motions without becoming unstable, and to achieve this the weight cannot extend too far outside of the robot's center of gravity. A low-level kinematic safety module ensures that the robot's center of gravity isn't radically changed in a short amount of time, which prevents the robot from becoming unstable in practice. However, this sensitivity to weight and center of gravity was a design constraint that was considered with the placement of every single component in the DragonBot mechanical system.

The major drawback of parallel manipulators is their limited working range, but DragonBot's linkage lengths were carefully crafted to fit a *squash and stretch* character. The range of the robot's motion is large enough to convey a wide variety of body poses. One of the DragonBot characters, Kombusto, can be seen in Figure 3-10 showcasing the range of motion achievable from a character using our inverted $\underline{\mathbf{R}} - (\mathbf{SS})^{\mathbf{2}}$ chains.

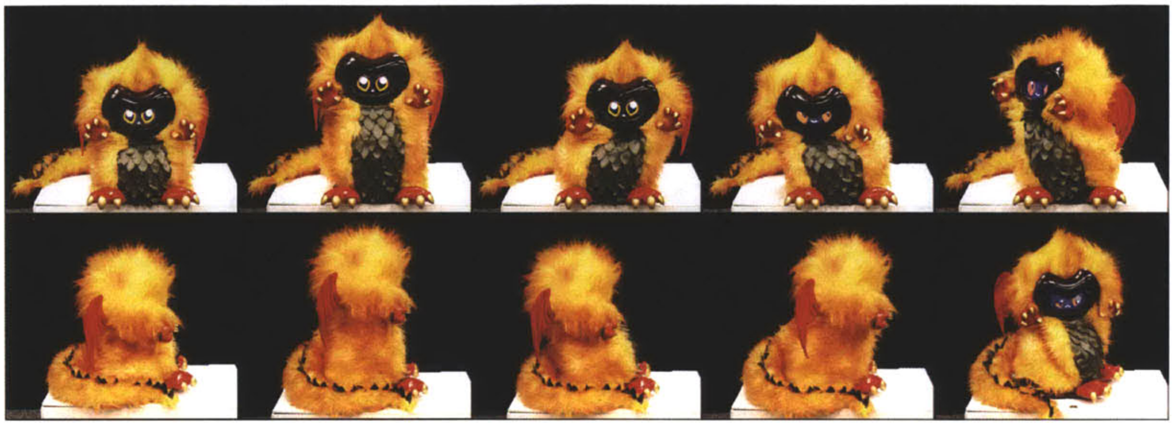


Figure 3-10: Range of motion - extrema on interaction sphere

Inverse Kinematics

Inverse kinematics is the process of calculating a set of joint parameters for a kinematic chain, given the desired end-effector position. This is the primary method of "animating" our characters, as the end-effector is the "floating head" of our robot. At the lowest level, the inverse kinematics for our parallel mechanism ensures that any pose sent to the robot stays within an *interaction sphere* - a solution-space known to be reachable and stable on the physical platform. The interaction sphere is smaller than the theoretical working area of the robot, but it protects the robot from entering states in which excessive forces of gravity or acceleration would yield unpredictable motion.

The algorithm for solving the inverse kinematics of the system is quite straightforward, and can be found in Algorithm 1. From a practical standpoint, this algorithm was able to run on a 20MHz 8-bit ATTiny85, so the computational overhead is quite low. Assuming the sin, cos, $acos$, and $atan2$ are implemented using lookup tables, the only significant computation will happen through the exponential operation, $Math.pow()$.

```
Input: (X, Y, Z, Theta) (desired end effector position in mm)
Output: J[0 : 3] (motor positions in radians)
// Kinematic constants (in mm)
E1 = 29.239; E2 = 29.239;
RB = 55.48; B2 = 34.1675;
A = 111.76; B = 53.98; D = 8.255;
for m = 0; m < 4; m++ do
    if m == 0 then
        xg = -Y - E1*(sin(T) - cos(T)) + E2;
        yg = X + E1*(cos(T) + sin(T)) - B2;
        zg = Z;
    end
    if m == 1 then
        xg = X + E1*(cos(T) + sin(T)) + E2;
        yg = Y + E1*(sin(T) - cos(T)) + B2;
        zg = Z;
    end
    if m == 2 then
        xg = Y - E1*(sin(T) - cos(T)) + E2;
        yg = -X + E1*(cos(T) + sin(T)) - B2;
        zg = Z;
    end
    if m == 3 then
        xg = -X + E1*(cos(T) + sin(T)) + E2;
        yg = -Y + E1*(sin(T) - cos(T)) + B2;
        zg = Z;
    end
    aPrimeSquared = (Math.pow( (A - 2*D)*(A - 2*D) - yg*yg, 0.5 ) + 2*D) *
    (Math.pow( (A - 2*D)*(A - 2*D) - yg*yg, 0.5 ) + 2*D);
    c = pow((xg-RB)*(xg-RB) + zg*zg, 0.5);
    alpha = acos((-aPrimeSquared + B*B + c*c)/(2*B*c));
    beta = atan2(zg, xg-RB);
    J[m] = beta - alpha;
end
```

Algorithm 1: Inverse kinematics

The forward kinematics is not presented because it involves solving an eighth-order polynomial, and it's not of any particular use in controlling the robot.

Linkages

The linkages of our parallel platform are the mechanical structures that connect the robot's head to it's body. The quality and precision of the linkages are the most important aspect in the kinematic stability and reproducibility of the platform. The first prototype, V1, used a very simple linkage mechanism - two threaded rods connected to spherical joints at both ends with a piece of acrylic and two springs in the center to stabilize the bars of the linkage in the same plane. Even though these linkages are simply an approximation to the exact kinematics of an $\underline{\mathbf{R}} - (\mathbf{SS})^2$ chain, this mechanism works remarkably well without any load on the mobile platform. However, with the weight of a mobile device, these linkages led to extreme instability of the top platform. Therefore, we set out to create a set of precision linkages that could support an inverted delta robot with a load on the mobile platform.

The first redesign of the linkages (seen on V2 in the middle panel of Figure 3-6), was attempted by connecting steel rods to brass clevis fasteners. These acted as the bars of the linkage, with a precision-machined steel connecting plate at the top and bottom. This concept was a complete failure because the entire mechanical linkage was not precision machined. There was too much "wiggle" in the parts, and due to the nature of parallel mechanisms, any tolerance issues in a single linkage are exaggerated through the entire kinematic structure. These linkages were used on prototype V2, and rendered it completely useless.

A complete redesign of the linkages using precision-tolerance parts and processes solved these excessive "wiggle problems". This set of linkages, used on prototypes V4 and V5 (and seen in Figure 3-11), combines shoulder screws (having tolerances of +0", 0.001") with hand-reamed aluminum struts to create a standard four-bar linkage. The bottom of this linkage uses an orthogonal shoulder screw and a steel hinge bracket to connect to the rotational joint of the motor ($\underline{\mathbf{R}}$). The top of the linkage connects to the mobile platform using the same orthogonal shoulder screw and hinge bracket, but combined with a steel rotator with 90-degree connection points. The rotator contains a standard ball-bearing with a washer on each side, enabling the top platform to rotate when the kinematics allow.

The tolerances of all machined parts are +.005", -0", and this precision at the inter-

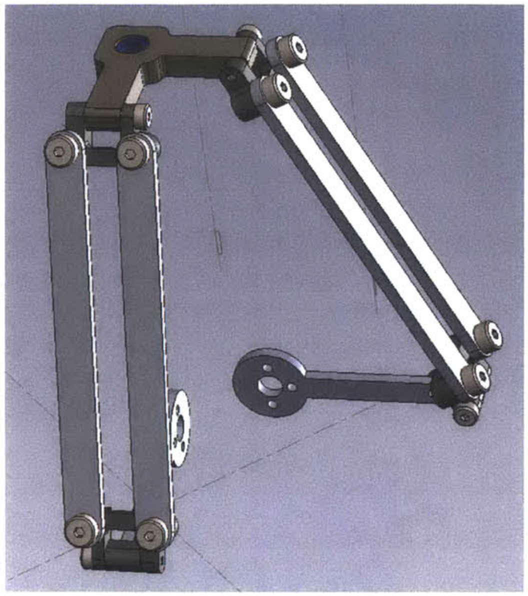


Figure 3-11: A pair of precision $\underline{\mathbf{R}} - (\mathbf{SS})^2$ linkages

section of the moving parts within the linkage lead to very little ambiguity in the position of the parallel kinematic chain. Even with this precision, the backlash inherent in low-cost motors leads to a slight bit of "wiggle" in the mobile platform. However, this backlash can't be reduced without extremely high-quality motors which would dominate the cost of the robot. In practice, the backlash of the motors is not a significant issue - the stability of the parallel platform with these precision linkages is good enough to support the weight of a mobile phone (even at maximum accelerations).

The mechanical drawings to recreate these linkages can be found in Appendix B.

3.3.2 Extra degrees of freedom

Two extra degrees of freedom were added to the robot beyond the parallel mechanism - a neck tilt and a wagging tail. The reasoning for these extra DOFs, in addition to the design decisions in implementing them, are described in the following two sections. The resulting mechanical prototype, V5, can be seen in Figure 3-12.

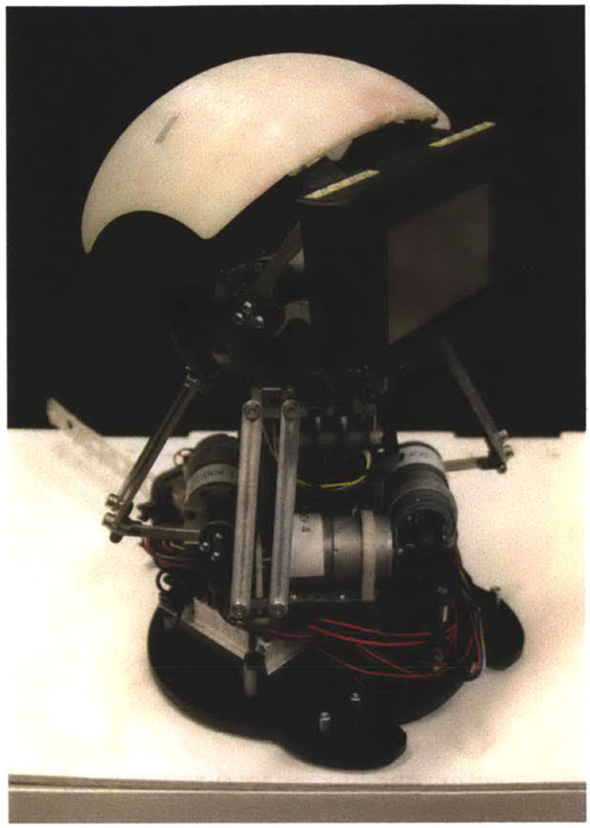


Figure 3-12: The DragonBot V5 prototype without any aesthetic elements

Neck Tilt

As soon as the design of the parallel manipulator was finished, it was apparent that another degree of freedom - a neck tilt - needed to be added. Because DragonBot is not a mobile robot, its interactions will tend to take place on a tabletop or on the floor. The ability to look down at an object, or up at a person, is essential. The iterative design process was extremely helpful here. Our first character, Nimbus (see Section 3.7.1), did not have a neck tilt. During our first interactions with kids, it was obvious that this DOF must be added to support an effective gaze system, which works as a transparency device for humans [59]. Head nodding is also known to be an effective route for mirroring and synchrony, improving the rapport of an interaction [60].

The range of motion of the neck tilt can be seen in Figure 3-13. The motor of the neck was positioned in the center of the platform for weight distribution, and a set of

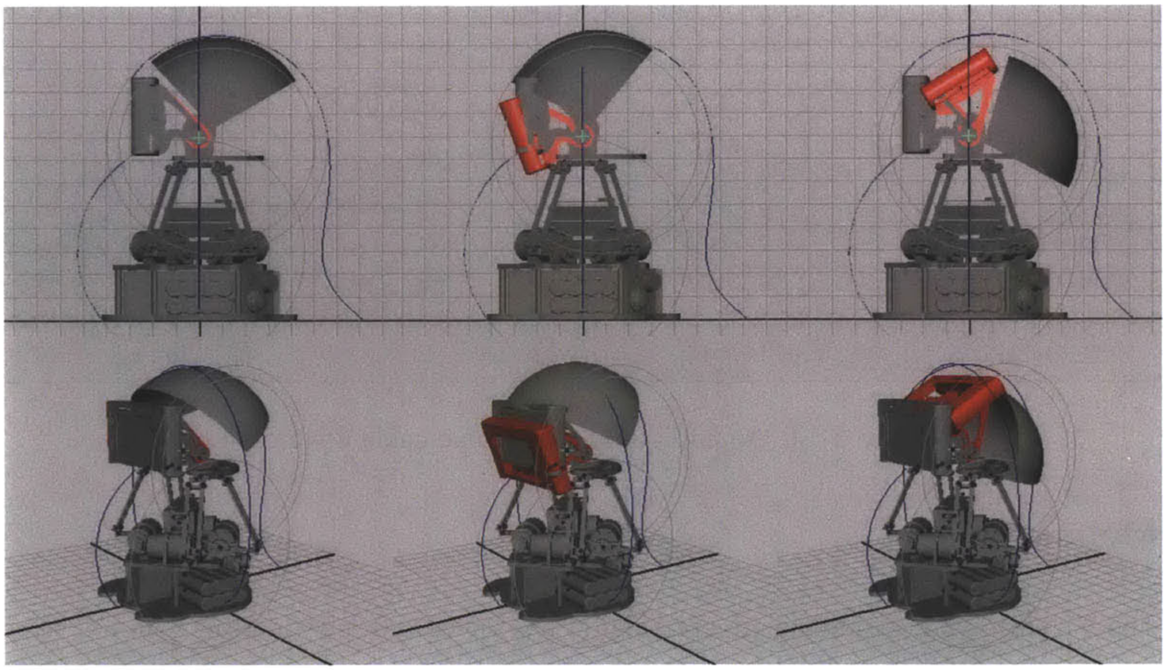


Figure 3-13: The range of motion for DragonBot's neck tilt

aluminum brackets extends the head of the robot so it can lower its gaze up to 15 degrees. Because the robot might need to look up at people from the ground (and there are no kinematic constraints restricting an upward gaze), it can raise its orientation up to 60 degrees.

Tail

The final prototype was augmented with a tail to add to the illusion of life conveyed by the robot. DragonBot's tail was an exploration into low-cost and easily-manufacturable cable-driven actuation. As can be seen in Figure 3-14, the tail attaches to the body using snap-fit flexures, and the tail can be configured to wag either vertically or horizontally by simply "unsnapping" and "resnapping" it in the desired orientation. The only parts in this cable-driven actuator are the motor, a timing pulley for coupling to the motor's shaft, a cable, a small piece of laser-cut 1/16p̈olypropylene, and standard screw terminals. The alternating tabs of the tail provide it with a natural curving motion during actuation.

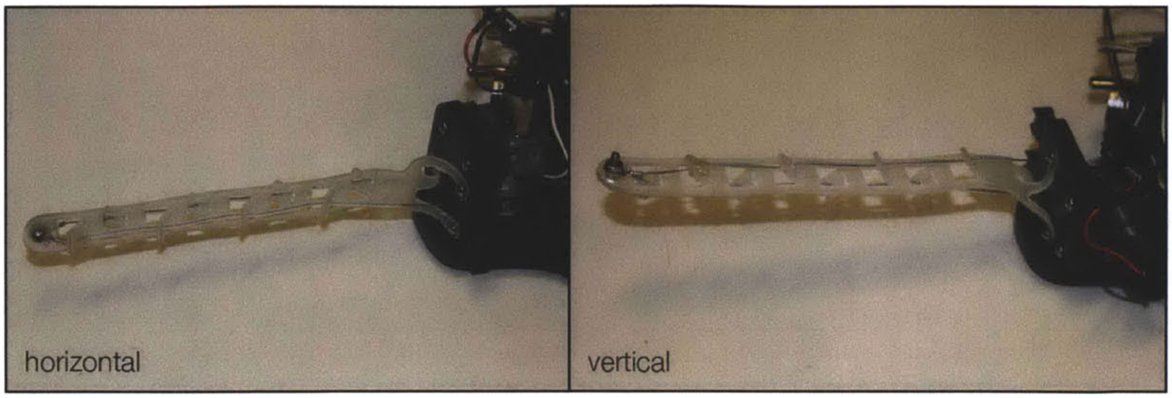


Figure 3-14: An easily-manufacturable cable-driven tail

3.3.3 Motors

The V1 prototype used standard-sized digital RC servos with plastic gears. An RC servo is a small brushless DC motor with a gearbox, a potentiometer on the output shaft and an integrated position controller. While these inexpensive and easy-to-control servos were powerful enough to articulate the acrylic-based V1 prototype, they could not support the load of a modern smartphone on the parallel platform. The next set of actuators were updated high-torque digital servos, which improved the stability of the motion by using metal gearboxes, and the accuracy of the coordinated motion through high-precision magnetic encoders. These high-torque DC servos were used in both V2 and V3 prototypes, and solved the weight problems encountered using the $20 plastic-geared servos.

The upgraded servos allowed us to quickly create our first character around the platform (see Section 3.7.1). However, the motor noise generated when under load was significantly too loud - around 60 decibels when the robot wasn't even moving. We know from experience that children on the autism spectrum can have trouble interacting when there is excessive noise in the environment. And furthermore, these servos cost about $120 each which would make the motors dominate the price of the hardware. They were not a feasible solution for a low-cost robot designed for social interaction.

The motor pipeline in V4 and V5 represents a complete redesign of the robot's actuation. Instead of hobby servos, 12V DC brushed motors with integrated magnetic quadrature encoders were used [61]. The gear ratio of the motors is 131:1, helping the actuators to support the weight of the top platform and minimizing the backdrivability of the platform. Further, these motors are silent when not moving, and generate only 35 decibels of sound when moving. And to top it off, they cost less than $40 a piece.

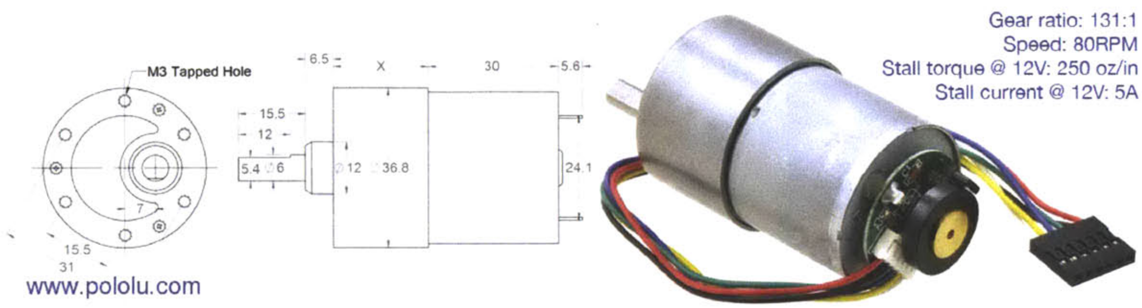


Figure 3-15: Primary motors for DragonBot V4 - V5

These motors are used to actuate the parallel platform and the neck tilt. The actuator used for the tail is a 16mm diameter 12V Copal motor with a 50:1 gearing, resulting in 330rpm and 53.25 oz/in of stall-torque.

3.3.4 Head design

The robot's head is based on a "flip-out" mechanism, drawing inspiration from the classic tape-decks of the 80s and 90s (see Figure 3-16). This allows the phone to be inserted into the head of the robot, making a physical connection between the phone and the onboard electronics. This detailed design was accomplished by developing the head with the assumption that it could be 3D printed. This mechanism is held closed by the character's faceplate (described in the following section), which along with the head is embedded with Neodymium magnets. These magnets allow the phone to

Figure 3-16: A flip-out style cassette deck

be quickly removed from the robot, and the lack of moving parts makes inserting and removing the phone safe for children.

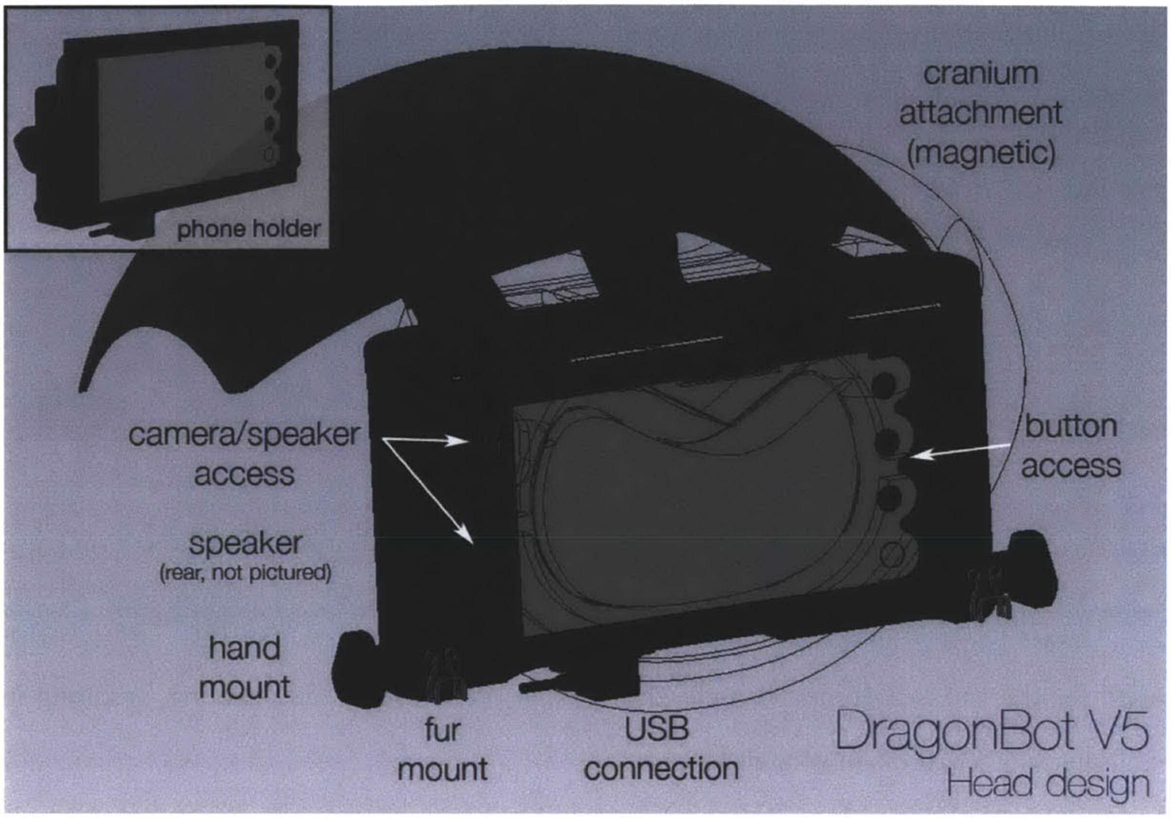


Figure 3-17: Head assembly for DragonBot V5

The head contains cutouts for the phone's camera, speaker, and buttons. There is a mount for a standard 4W speaker behind the phone, should the robot need to speak in a loud environment. A USB connection was facilitated using a 90 degree micro-USB cable and a small 3D printed cap that places the cable in a location where a connection with the phone is guaranteed. Small fabric hooks and large triangular nubs are included to provide mechanical connections to the fur and hands (respectively). The top of the head has three prongs with embedded neodynium magnets, which connect to a separate 3D printed cranium.

The fact that the head is 3D printed is very important for the forward-compatibility of DragonBot, as well as its potential use outside of the laboratory. Because the design of the phone holder within the head is coupled to a specific device, it's desirable to be able to use the newest phones as they are released. Because this part was parametrically modeled, the dimensions of it can easily be changed to support a new phone. So when you upgrade your phone, you simply tweak a few parameters and print out a new head for your robot. This enables the entire platform to advance along with mobile technology, and mitigates the need for a complete mechanical redesign.

3.3.5 Usability

From a user-facing standpoint, the usability of the platform is extremely straightforward. A control panel on the back of the robot exists with three components: an on/off switch, an LED indicator, and a receptacle to connect the robot to its charging station (discussed in Section 3.6.1) (the charging station can be seen in Figure 3-34). Every aspect of the electronics and mechanics ensure that there can be no short-circuits that result from plugging in the robot.

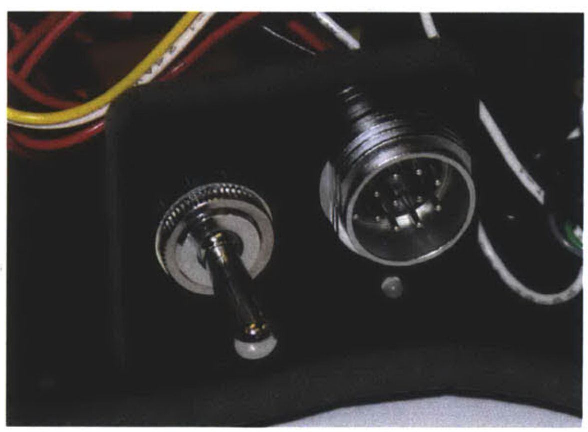


Figure 3-18: Control Panel with charging port and power switch

Multiple characters were developed around the DragonBot hardware, and the aesthetic elements of these characters' "costumes" are able to attach to and detach from the robot very simply. The hands of the robot press-fit onto triangular pegs sticking out from the 3D printed head. The triangular support ensures that the hands are always oriented correctly. The feet press-fit onto two small pegs on the base of the platform that similarly constrain their orientation. The fur has standard fabric hooks sewn in, that are latched onto the 3D printed head.

Figure 3-19: Mechanical mounts for hands and fur (left), and magnetic mounts for the wings (right)

All other aesthetic elements are attached with magnets. This includes the dragon's wings and the robot's "faceplate". The faceplate is a way to frame the character's face and secure the phone from falling out. Each faceplate conveys a different type of character, and an assortment of design concepts can be seen in Figure 3-20. These faceplates are 3D printed and embedded with high-strength (N52) neodynium magnets. Because there is material separating these high-strength magnets from the ones embedded in the head, the force isn't strong enough to hurt the small finger of a child. However, the magnets provide enough force to stay connected, even during particularly vigorous motions.

3.4 Electrical Design

The electrical design of DragonBot is centered around an Android smartphone. The platform will support any device with a front-facing camera running Android 1.5 and above. All that must be done to integrate a new phone into the system is change a few variables in the head design (see Section 3.3.4), and print out a new one.

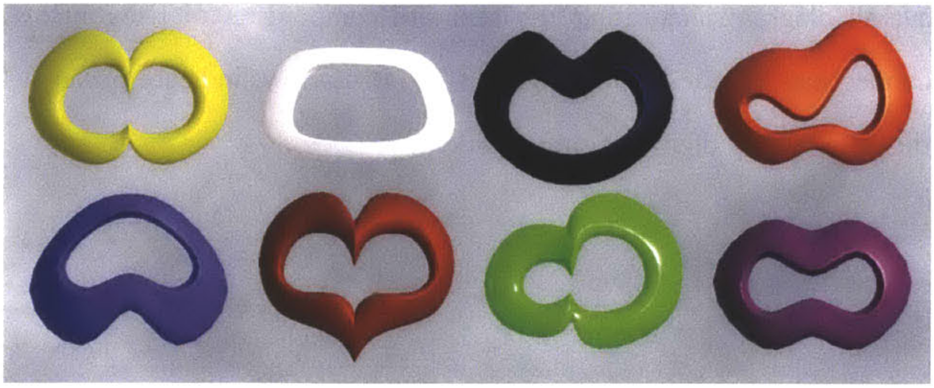


Figure 3-20: Faceplates designs for different character types (concept and modeling by Fardad Faridi)

While the Android platform provided the best platform on the market for our specific needs, we also assessed the viability of using other classes of mobile devices along the way. The first device, used in the V1 prototype, was a Samsung Galaxy Epic running Android 2.1. The second, used in the failed V2 prototype, was an iPod touch. The V3 prototype used an HTC Thunderbolt running Android 2.2. Both V4 and V5 prototypes make use of the HTC Evo 3D with Android 2.3.4. The Evo 3D has a 1.2GHz dual-core processor and 32GB of storage, a 4.3 inch touchscreen display, a 1.3 megapixel front-facing camera, WiFi/cellular connectivity, and an integrated microphone and speaker. The mobile devices used can be seen in Figure 3-21.

Figure 3-21: The mobile devices used for the DragonBot prototypes (from left to right): Samsung Galaxy Epic (V1), iPod touch (V2), HTC Thunderbolt (v3), HTC Evo 3D (V4 and V5)

The overall diagram illustrating the separate components of DragonBot and their required connections can be seen in Figure 3-22. Red arrows indicate power, while black arrows indicate analog / digital communication.

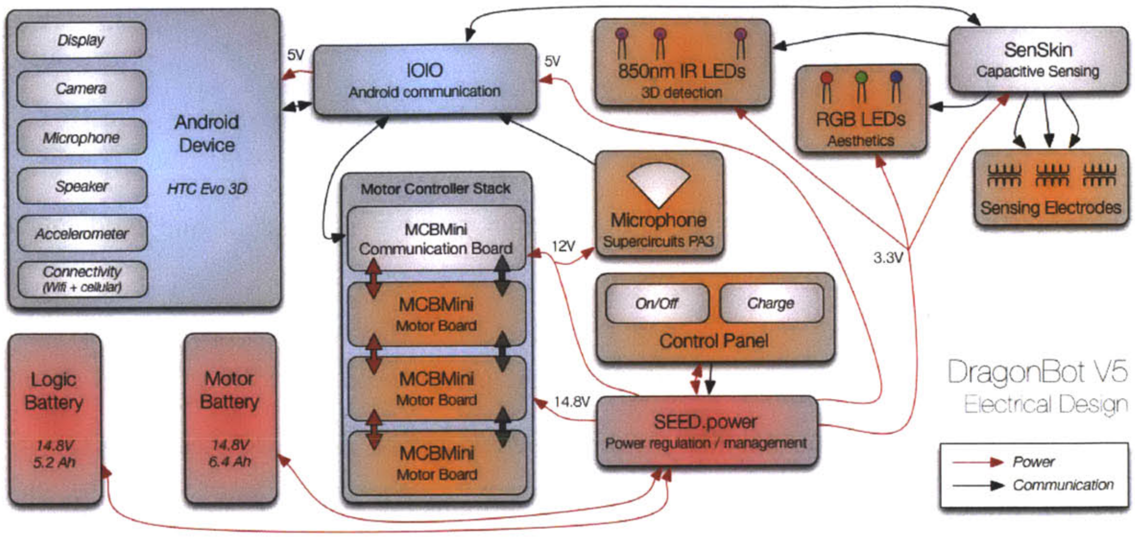


Figure 3-22: DragonBot electrical diagram

The core software framework for the Android device (discussed in Section 4) uses the phone's display, camera, microphone, speaker, accelerometer, and networking capabilities (WiFi). In addition to the phone, one off-the-shelf board and six custom PCBs provide the functionality for the robot to operate.

3.4.1 Controlling a robot with a mobile device

The first challenge in using a mobile phone in lieu of a traditional computer was a method of communicating data from the host machine (the phone) to an embedded microprocessor with access to the robot's sensors and actuators. This problem is largely solved for interfacing between robots and standard desktop/laptop computers. A USB cable is connected between the computer and the robot, and the computer acts as a "host" for the robot. Much like with a printer or hard drive connected over USB, the computer handles connecting with and communicating to the robot, using an interface like RS-232 or

CAN. However, modern smart phones were never intended to have external "peripherals" connected to them, so they are typically capable of only running as a USB device, not a host. [1]

The first three prototypes tried to bypass this problem by using a wireless interface instead of a direct wired connection. V1, V2, and V3 all used an Arduino prototyping board to generate and send signals to the RC servos. The motion trajectories were storied on the microcontroller's non-volatile memory, and a bluetooth module was connected to the Arduino in order to communicate with the smart phone. While this was a nice way to trigger animations from the phone and quickly prototype characters, a very limited amount of motion trajectories could be stored on the Arduino's memory. Furthermore, bluetooth cannot support the framerates required to produce smooth motor movements, so this solution needed to be improved.

Although a prototype was developed using a WiFi module instead of bluetooth, it soon became apparent that the motor server (the module that generates the motor trajectories from a desired target position) required a direct wired connection for smooth and lifelike motion. First we tried the official Android ADK developer board, which yielded poor performance and restrictive functionality. A custom solution would have taken months of development, but then we discovered a tiny board developed by Ytai Ben-Tsvi called the IOIO. And suddenly our problem was solved.

3.4.2 IOIO

The IOIO [62] (pronounced "yo-yo") is an embedded board that provides robust connectivity to any Android device running version 1.5 or greater. The IOIO connects to the phone through a physical USB connection, or optionally without any cables using a bluetooth dongle. The IOIO has nearly 50 general purpose pins, and supports up to four

[1] As of writing this document, the newest Android phones support USB host mode natively, as non-traditional use of phones is becoming more widespread.

communication channels over SPI, I^2C, or $UART$. One of the nicest features of the IOIO is it's lack of a programming interface. To be more specific, the IOIO never requires programming, and it is controlled solely from a high-level Java library through any ordinary Android application. This pipeline makes it trivial to read a value from a specific sensor or write a PWM value to an output device, all from the high-level without ever needing to change the embedded code on the microcontroller.

The IOIO board contains a single PIC microcontroller that acts as a USB host and interprets commands from an Android application. The IOIO can use either the Android Debug Bridge (adb) or the official OpenADK specification with Android devices running OS 2.3 and higher (providing higher framerates through improved hardware support). It also supports Android application notifications on connection or disconnection, so the application can be notified when the phone is inserted or removed. Finally, use of the IOIO doesn't require the user to make any hardware or software modifications to their Android device, so the robot can be used without the fear of violating the warranty of the mobile device. A picture of the IOIO can be seen in Figure 3-23.

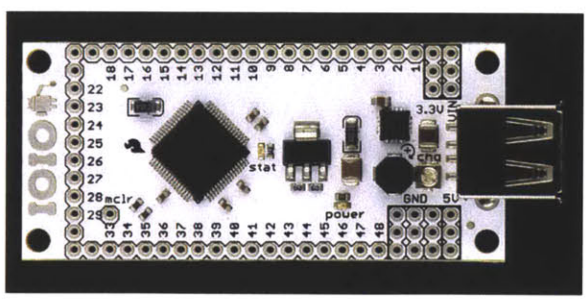


Figure 3-23: The IOIO board for bridging android and embedded systems

In DragonBot, the IOIO is primarily used to delegate communication from the motor server on the phone to the embedded motor boards (which will be discussed in the following section). It is also connected to the SenSkin board (discussed in Section 3.4.5) and a high-sensitivity directional microphone. It was initially planned to combine the audio stream from this high-quality microphone with the smartphone's integrated sensor, running a signal processing algorithm to remove any background noise from a speech signal.

However, the IOIO's PIC microcontroller wasn't fast enough to support motor control AND the audio requirements, so the signal from this directional microphone is currently unused.

3.4.3 MCBMini motor controllers

The MCBMini motor stack [63] was created by Sigurdur Orn Adalgeirsson during his development of MeBot, a socially-embodied telepresence robot [64]. Sigurdur's expertise in electrical engineering has driven the development of MCBMini to yield a compact solution for elegantly solving the most common issues in coordinated motor control.

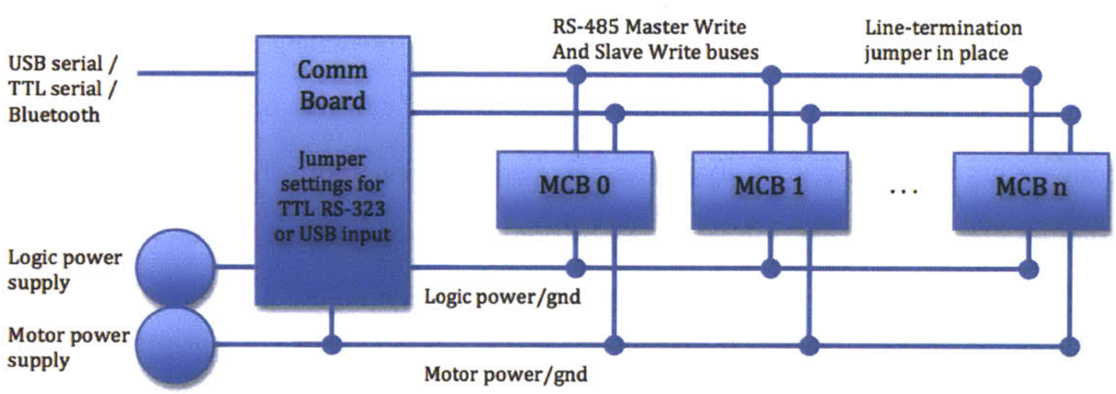


Figure 3-24: The MCBMini control scheme

From the highest level, MCBMini represents a stack of boards for interfacing with and controlling a set of brushed DC motors between 9V and 24V. There are two boards that make up the MCBMini stack. The first is the **MCBMini communication board**, which is used to delegate commands from an offboard host (such as a computer or a IOIO) to the individual motor controllers. A single communication board is used to link together a chain of many MCBMini motor boards. The **MCBMini motor board** combines an ATMega328 microprocessor with a pair of Cirrus Logic SA57AHU-FH H-bridge chips to generate the waveforms needed to interface with a DC motor. Each MCBMini motor board is just 68mm x 43mm, a tiny footprint for a powerful board that can drive two large

DC motors (8A maximum current limit).

The communication board and motor boards talk using the RS-485 protocol - a standard communication scheme where a balanced (differential) transmission line is used in a multi-party configuration (more than one device). RS-485 supports high transmission rates (up to 10Mbps at less than 40 feet) and supports up to 32 nodes (for a theoretical limit of 62 motors on the MCBMini stack). A typical RS-485 network can provide reliable communications in electrically noisy environments and can operate in the presence of reasonable ground differential voltages, making it particularly suitable for motor control where inductive loads can cause significant voltage spikes.

MCBMini is capable of operating motors through both position mode (signaling the motor to turn to a specific *angle*) and velocity mode (signaling the motor to turn at a specific *speed*). Furthermore, the motor controllers can use both encoders and potentiometers for feedback. A magnetic encoder is typically an extremely precise sensor, although it has no concept of absolute position. A potentiometer is often much less precise (because it's placed on an output shaft), but has an absolute positioning. Combining these sensors together results in the most accurate and reliable control.

The real magic of the MCBMini is in the software running on the microprocessor. Most importantly, the control scheme guarantees synchronization on the RS-485 bus, meaning that each motor will always move a single "frame" at the same time. MCBMini's software revolves around a tunable PID controller that runs at an update frequency of 200Hz (with an internal PWM-update rate of 20kHz). All analog values - potentiometer feedback and current values - are averaged over 5 measurements before being integrated in the PID feedback. In position mode, the controller uses a target dead-band with hysteresis, stabilizing the motor and preventing it from oscillating around its target value. There is also a maximum step parameter that prevents the motor from hurting itself by attempting to move too far during a single timestep. Finally, a communication timeout of approximately 1 second will disable the motor channels. This effectively makes the robot disable its motors if the host (the smartphone) malfunctions or is shut down unexpectedly. In this case, when the host returns, the boards will broadcast their presence and reconnect automatically without the need to be reinitialized.

DragonBot's six motors require a single communication board and three motor boards. DragonBot only uses position control, as the robot is non-mobile and would not benefit from velocity control. The motors for the parallel mechanism and neck rely on magnetic encoders for feedback, while the tail uses a potentiometer on the output shaft for absolute position.

3.4.4 SEEDpower

To support long-term interactions, it is essential that DragonBot can power itself for at least a few hours. In order to provide the robot with an acceptable runtime, we used high-capacity 14.8V lithium-polymer (LiPo) batteries. Because the motors controlling the parallel mechanism were susceptible to high torques, current-spikes in the power sources rendered it challenging to control both the logic and motor channels of the robot using a single supply. The problem arises when a motor draws a sizable amount of current (in our case about 3 Amperes), causing the voltage on the logic side to drop (in our case up to 50%). This volatility was frequently enough to reset the microcontroller on the logic boards, which made the robot power down and lose connectivity.

As there was not a viable power management solution for a compact robot, the SEED-power management boards were developed. These compact and general-purpose boards output regulated 12V, 5V, and 3.3V power, provide isolation of logic and motor sources, and support integrated charging of LiPo batteries while powering the robot externally.

SEEDpower is an integrated solution for power management and regulation on small-to-medium sized robots. With full isolation of logic and motor power sources, the board supports 3-channel input (up to 3 batteries) and 4-channel output (Raw motor power, +12V, +5V, and +3.3V). Any of the two input batteries may be placed in series or parallel (using on-board jumpers), and the output is fully protected with both fuses and flyback diodes. The board supports "plug-and-play" charging, using an onboard relay to switch sources an external supply whenever the robot is plugged in. For motor configurations that will not generate significant current spikes, a single battery may be used with mild power isolation.

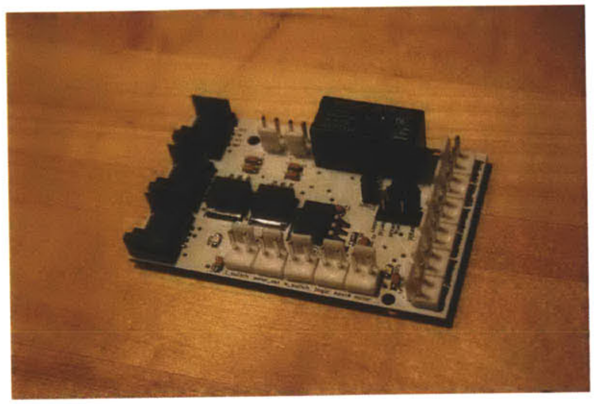


Figure 3-25: The SEEDpower board for battery isolation, power regulation, and smart-charging

The final power configuration for DragonBot can be seen in Figure 3-22. The motors are powered by a dedicated 14.8V LiPo battery with a 6.5Ah capacity, and the logic is powered by a similar LiPo with a 5.2Ah capacity. The logic battery trickle charges the smart phone through the IOIO when the device's internal battery runs low. This battery configuration was able to power DragonBot for 6.81 hours, when the robot was streaming it's video / audio stream to an offboard computer over WiFi and performing a random action every 20 seconds.

3.4.5 SenSkin

SenSkin is a general-purpose board for small robots, designed for capacitive sensing and driving LEDs. Combining an ATMega328 microprocessor with a Freescale MPR121Q capacitive sensing circuit, SenSkin can detect touch and pre-touch from up to eight electrodes. The board can also directly drive up to six LEDs, and takes commands over an I^2C socket. The board is operated at 3.3V, and the microcotroller and MPR121Q both run in low-power mode. The MPR121Q is also an interrupt-driven device, so the overall power footprint of the board is less than 10mA. The board can be seen in Figure 3-26.

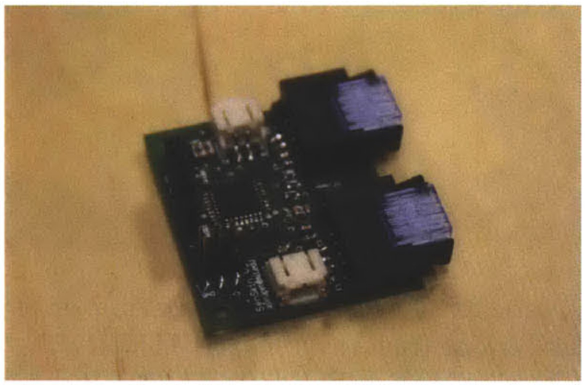


Figure 3-26: The SenSkin board for capacitive sensing and driving LEDs

SenSkin was never fully integrated into the DragonBot behavioral system, but was simply an exploration into designing electronics for soft-sensor capacitive sensing. The sensor-focused aspects of this will be described in Section 3.5.1.

3.5 Aesthetic Design

As indicated by the title of this thesis, the robots based around DragonBot are characters. Because principles from animation have guided the design of underlying hardware and software, they too hold for the aesthetic design of the robot. Because DragonBot is a robot designed with children in mind, the general appeal and liveliness of the aesthetic was important to capture the user's attention. Furthermore, a convincing aesthetic enables a *suspension of disbelief* – the willingness of an observer to overlook a medium's limitations.

Figure 3-27: 3D printed parts for DragonBot (clockwise from upper left): cranium (FDM), head (FDM), hand (MJM), foot (MJM), faceplate (MJM)

3.5.1 Fur

Bi-directionally stretchy synthetic fur was used to connect the two platforms of the parallel manipulator, and longer-pile fur was used to cover the head. The stretchy fur comprising the torso of the robot supports the *squash-and-stretch* principle, which was discussed in Section 3.2. Small pockets of light cotton were sewn into the fur to enhance the round shape and make the character soft and touchable for kids. Natalie Freed was instrumental in the design and creation of the furs, combining her mastery of textiles with a character-driven approach. Also, the author was lucky enough to make some friends in the soft-goods division at Hasbro, who were of tremendous help with creating a sewing pattern that conveyed the expressivity desired.

The furs are colored using acetic-acid based dyes intended for synthetic materials. When attempting to dye these furs in the washing machine - the standard method of coloring fabrics - the fur develops clumps and loses a significant amount of aesthetic appeal. Because of this, the dye is mixed with boiling water and sprayed directly onto the fur. This process allows our textile to be colored without sacrificing the passive motion

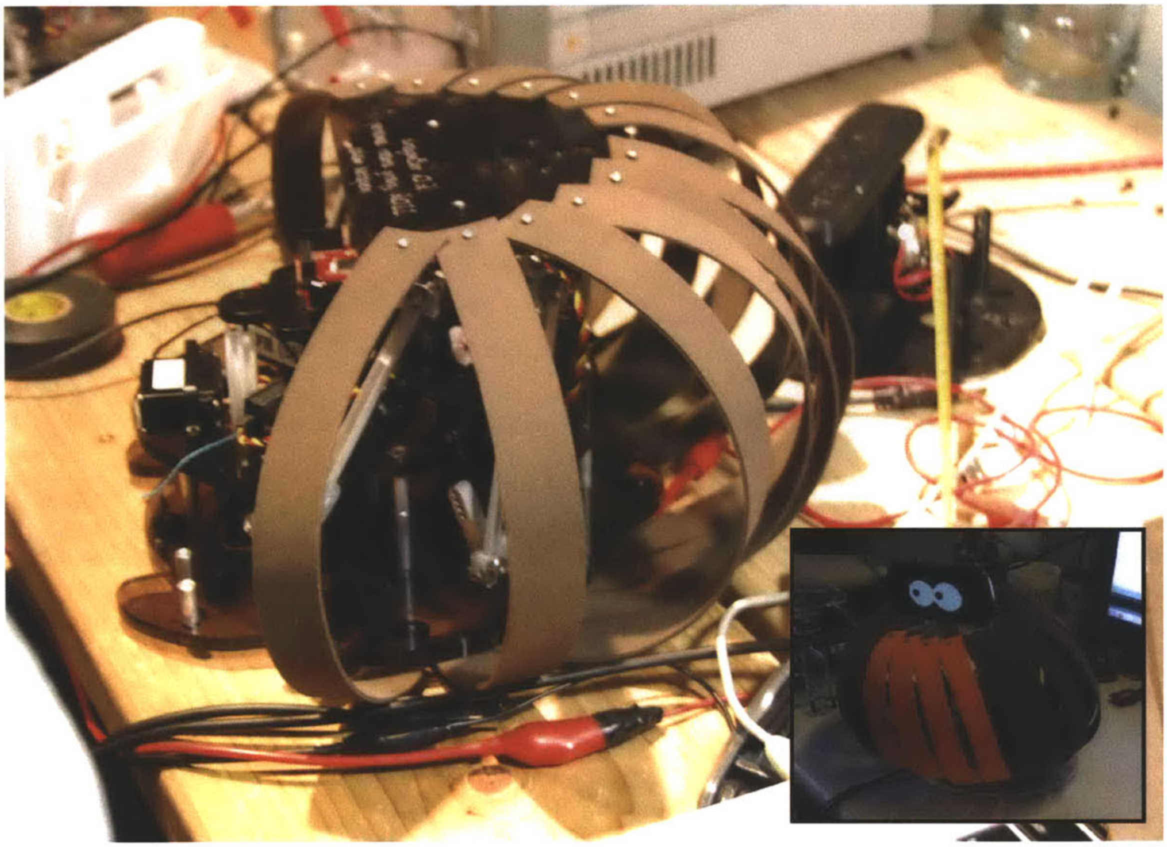


Figure 3-28: Prototyping squash-and-stretch exteriors for the robot

from a "fluffy" long-pile fur.

As an exploration into augmenting the furs with touch-sensitive areas, six soft-sensor electrodes were sewn into an early aesthetic prototype. It's important that all fabric and thread connected to the structure of the fur can stretch in both directions. For this reason, the electrodes were made from MedTex130 - a silver-plated nylon that is stretchy in both directions. It has a surface resistivity of less than 1 ohm per square foot, making it an excellent electrode capable of detecting pre-touch up to 6 inches.

The complication of creating a sensing fur lies in the fact that the electrodes must be well shielded from both the inside and the outside. The prototype pictured in Figure 3-29 shielded the MedTex130 by sandwiching it on both sides with a layer of spandex, a ground plane made up of more MedTex130, and another neutral layer of spandex for protection from any potential short-circuits. This was all sewn on with wooly nylon thread, which is

Figure 3-29: A prototype of a fur augmented with textile-based capacitive sensors

a nylon-based thread intended for use with a serger.

Some loose cotton was added in the layers to provide more body, and a standard fabric-snap was used to connect the sensing electronics to the sensing electronics, allowing the fur to be easily "donned" and "doffed". The prototype used shielded cables, and the grounds were all internally connected using conductive thread. Even with the amount of effort put into shielding, the signal received by any capacitive sensing circuit connected to the fur was always noisy when the robot was operating.

Figure 3-30: Dying a synthetic two-way stretch fur

The wings of the robot consist of .2mm aluminum sandwiched by standard 2mm craft foam. Slots were cut in the main supports of the wings in order to thin them out and increase the amount of passive motion generated. A magnet is embedded into the base of the wing and a fabric enclosure is sewn around everything to provide both the structure and aesthetics of the wing.

3.5.2 Hands and Feet

Figure 3-31: Foot prototype, urethane-iron hybrid material

This section describes the molding and casting design processes for creating the robot's soft appendages. The hands and feet of the robot were envisioned from the beginning as both "squishy" and "floppy", adding to the lifeform-inspired design and encouraging tangible interactions with kids.

The first exploration into soft-appendages used an acrylic 3D printed mold in which urethane rubber was cast. These urethane-based hands and feet, eventually used for the Nimbus character, had the nice property that you can add a small amount of iron filings to the casting mixture without affecting the curing process. Through this simple trick, inspired by a recent breakthrough in micromechanics [65], we were able to turn the entire foot into a pressure sensor. By sticking a wire into the foot and directly connecting it to a microcontroller, we were able to simply use a step-response function for a beautiful response to pressure. Because of safety concerns with the urethane rubber, as well as a generally "unpleasant" feel of the material, the urethane rubber was abandoned in favor of something more appropriate.

The V4 prototype used hard plastic aesthetic elements, which can be seen in Figure 3-32. These acrylic appendages could be painted with extreme detail, and looked quite nice paired with the right aesthetic. However, they just didn't have the right feel.

After many failed attempts at molding and casting with countless different materials, an appropriate pipeline - and rubber - for creating our soft appendages was found. Em-

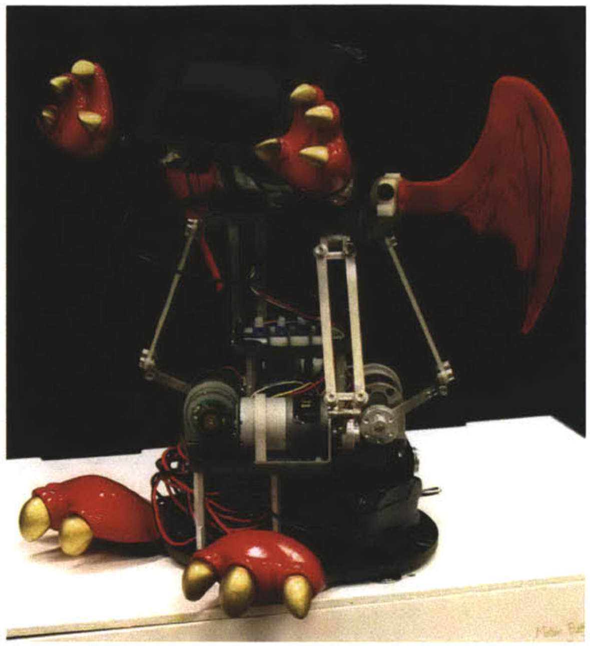


Figure 3-32: DragonBot V4 prototype with 3D printed aesthetics

ployed on the V5 prototypes, the feet are made out of a very soft silicone (0050 shore) and the hands are made from a slightly-less-squishy 20A shore silicone. Both of the silicones are platinum-cure, which is a process that doesn't react at all with the acrylic 3D printed molds, and yield parts that are harmless for kids to handle. Unfortunately, the addition of iron filings into a platinum-cure silicone yielded some rather explosive results, so the hands and feet are not active pressure sensors.

3.6 Peripherals

Three peripherals were designed and built to support the use of DragonBot outside of the laboratory: the charging station, the KinecTable, and a travel case. The charging station allows the robot to run through external power and charge its internal batteries without any expert knowledge or need for disassembly. The KinecTable augments the robot's capabilities with a 3D depth sensor, and the travel case allows the robot to easily be transported.

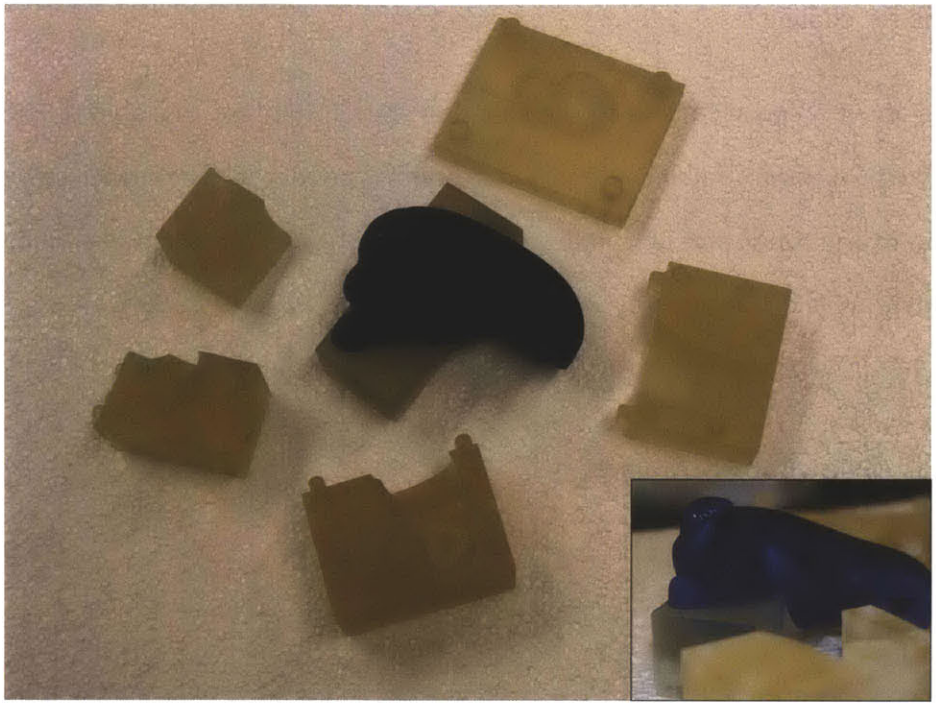


Figure 3-33: 7-piece hand mold from 3D printed acrylic, along with a silicone-cast blue hand

3.6.1 Charging station

The DragonBot charging station is a kid-friendly device that charges DragonBot's three Lithium-Polymer batteries while simultaneously providing external power to the robot. The charging station is tightly coupled with the SEEDpower boards, so one must simply plug in the power umbilical and the robot starts charging. This can all be done while the robot is fully powered and all behaviors remain uninterrupted as the Android phone runs continuously on it's own integrated battery.

The power station is a lockable enclosure that holds the following:

- Three lithium polymer battery chargers
- Two 120W power supplies (for powering the robot during charging)
- an ATMega328 microcontroller with custom current sensing circuit
- 8x8 LED array for displaying battery levels, controllable using I^2C

- fans for cooling

There are very few elements to the charging station that are available to the user - an on/off switch, a power umbilical, and a battery display meter. The power station can be connected to the robot and powered on in any order. A custom enclosure made of Baltic Birch houses all the components, which can be seen in figure TODO.

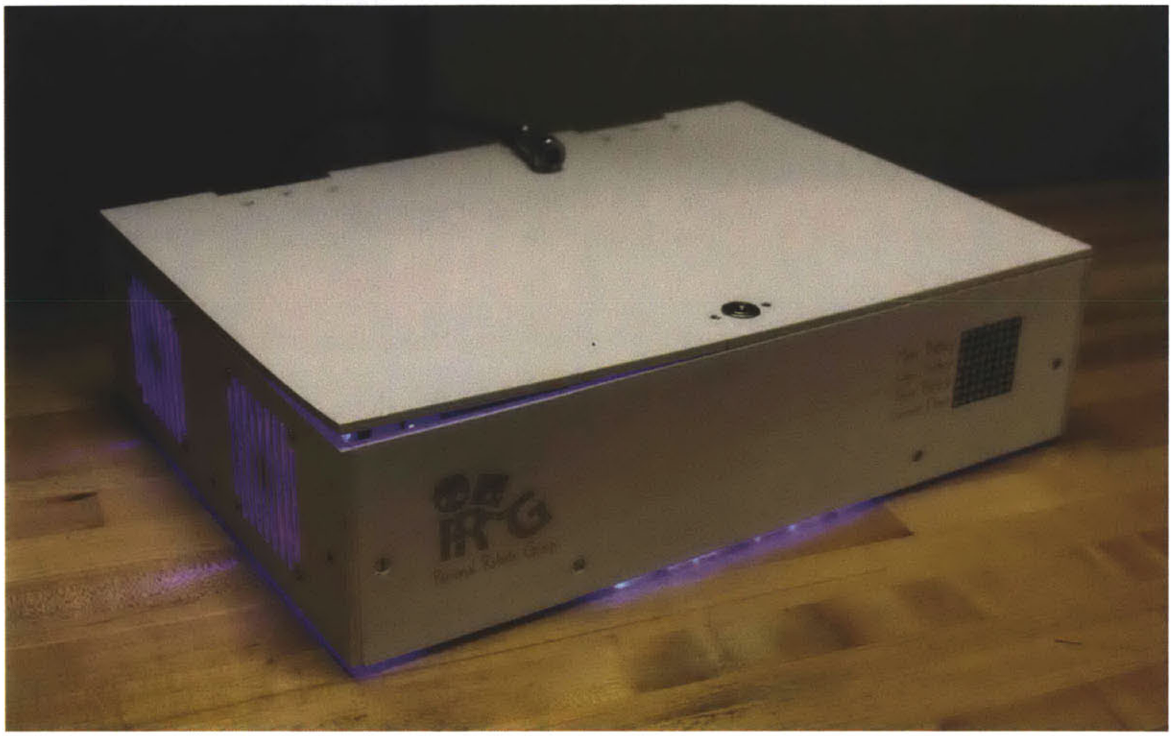


Figure 3-34: Charging Station for DragonBot

The charging system was designed to be extremely easy to use and safe, supporting the platforms use by children and viability in a remote lab. To charge the batteries and concurrently supply external power to the robot, one must simply plug in a single cable from the charging station to the robot, and SEEDpower board takes care of the rest.

3.6.2 KinecTable

A custom surface for table-top interactions was designed to augment DragonBot's abilities. Dubbed KinecTable, the interaction table has a large surface area that fits the robot's gaze range and still allows a child to be within arms reach of the robot. The table has a vertical structure for mounting a Microsoft Kinect (or other camera/depth sensor) that enables the Kinect to capture the depth and camera data of the tabletop and the seated user. The table also has a slot for a small computer (a Mac Mini) and an integrated speaker, both which are hidden underneath the surface. The robot sits on top of a speaker grill, which can allow the robot's sounds and voice to be audible in noisy environments. Because the robot sits directly on top of the speaker, the perception of the user is that the sound is coming directly from the robot.

The top of the table was painted white to allow for better image recognition and color segmentation. A render of KinecTable can be seen in Figure 3-35, and the actual table can be found in Figure 6-2.

Figure 3-35: Design of KinecTable for augmenting DragonBot's capabilities

3.6.3 Travel case

A compact hard-shell suitcase was converted into a travel case suitable for transporting DragonBot. This case, which can be seen in Figure 3-36, can safely house the robot and a spare set of batteries. This case was used to bring DragonBots to local preschools and museums, across the country for a robotics expo, and even overseas for a conference. And it can even be stored in an overhead bin!

Figure 3-36: Travel Case for DragonBot

3.7 Characters

As the title of this thesis suggests, DragonBot is intended as a platform upon which to create convincing **characters**. In this light, many different characters were developed around the various prototypes along the design process. We now describe these characters, including links to supplemental videos when available.

3.7.1 Nimbus: A first character

Nimbus (seen in Figures 3-37 and 3-38) was the first character developed around the parallel platform, using the V3 prototype. The metal-geared RC servos of this version offered the snappiest movement of all 5 prototypes. Some software was written for Nimbus that uses a 3D depth sensor to mirror people's motions when they step in front of the robot. For example, if the person squats down, the robot would also squash itself to become shorter.

Figure 3-37: *Nimbus*, an initial character for testing the expressivity of the parallel manipulator

Figure 3-38: *Nimbus* and a group of preschoolers

Using Nimbus as a character in interactions illustrated the ways in which the platform needed to improve. The fur design needed work to maintain organic motions and the RC servos proved to be much too loud to facilitate social interactions. And we realized that a neck tilt was needed so the robot could gaze up and down in its environment. A video of Nimbus, including autonomous mirroring behavior, can be found at `http://vimeo.com/39997345`.

3.7.2 Kombusto: A dragon is born

Kombusto (seen in Figure 3-39) was the first dragon-inspired character built around the platform, and was constructed around the V4 prototype. Kombusto was explicitly modeled to look like a brique of charcoal exploding into flames, resulting in a dark belly and flame-like fur. Kombusto's fur was colored using seven layers of dye to create a gradient, moving from a base of yellow to mids made of oranges and finally red tips. Instead of soft-silicone appendages, Kombusto's hands and feet were 3D printed and hand-painted. Although aesthetically these parts looked preferable to silicone-casted ones, they were not suitable for tangible interactions with children. The mounting mechanism for this style of hands resulted in them frequently falling off when kids touched them.

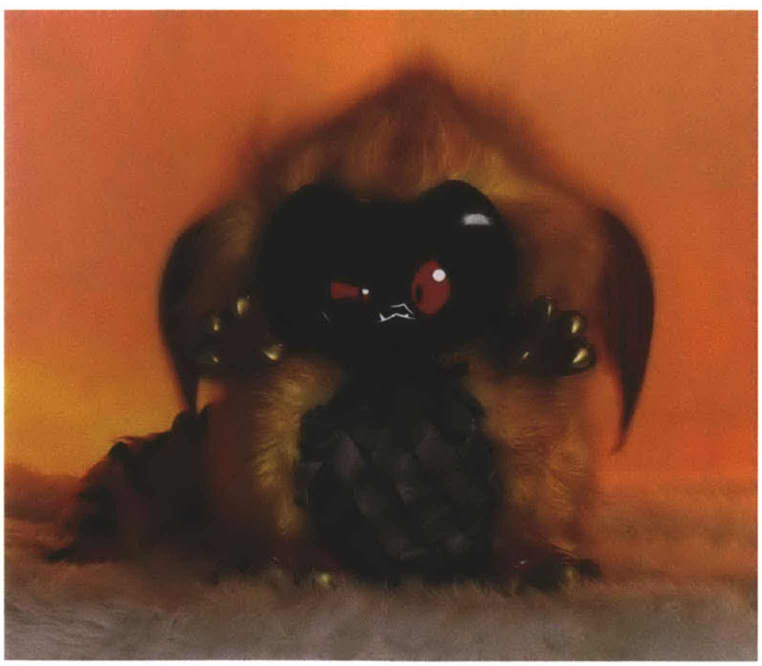


Figure 3-39: *Kombusto*, the first dragon-inspired character (photograph courtesy of Andy Ryan)

Kombusto is a feisty character with "grabby" hands and a very bright appearance.

This is particularly suitable for a certain type of character, but many young girls were frightened of his look. We feel that this is primarily due to the placement of Kombusto's hands and his color scheme. This prompted the hand design of the future characters to be at a much more neutral pose. Finally, it should be noted that Kombusto is overwhelmingly the favorite character of the boys who came in to play with the robots. A video showcasing Kombusto and the DragonBot platform can be found at
`http://vimeo.com/31405519`.

3.7.3 The whole gang: Neon, Pepper, Galli and Nitro

Neon, Pepper, Galli and Nitro were all built on the V5 platform. Aesthetically, they differ from Kombusto through their soft-silicone hands and feet, their actuated tails, and the in the mounting mechanisms for of all of the robot's aesthetic parts.

Figure 3-40: Neon, a character for tutoring french designed by Natalie Freed.

Neon (seen in Figure 3-40), a character designed by Natalie Freed, is intended to be used as a French-tutor for children that are learning second languages. Through Natalie's work with DragonBot, she developed the Neon character and a virtual tabletop built around an Android tablet. Neon uses the same facial rig as Kombusto, only with a different set of textures overlaid on top.

Pepper (seen in Figure 3-41) is the character that was used in the evaluation of this thesis. This dragon-inspired has yellow eyes, yellow spots on its faceplate and wings, and a gender-neutral aesthetic. The speed of Pepper's actions were slowed down by 15%

compared to Kombusto's, as we wanted the character to be a bit less lively than Kombusto. Finally, Pepper has a special "Sneeze" animation that is unique to its character.

Figure 3-41: Pepper - the character used in the primary user evaluation for this thesis

Galli and Nitro (seen in Figure 3-42) are twin characters built on the DragonBot platform that are currently being used to investigate the ways in which contingent behaviors can affect social interaction with children. By designing the two robots to have a very similar aesthetic appearance - one using blue and one using green - we are able to use both of these robots at the same time in studies where their looks could bias the participants. For example, if a study was run with Kombusto, it's likely that boys would have a more positive experience with the robot than girls. However, by designing these two characters in a similarly-neutral way, their actions can be subtly tweaked and any results can be attributed to the robot's behavior - not the aesthetic.

Figure 3-42: Galli and Nitro, the Dragon Twins

3.8 Limitations and Complications

While the parallel mechanism provides an expressive four DOF movement, it is not meant to carry a heavy load. The industrial configurations that use parallel manipulators typically orient the stationary platform above the mobile one, so the manipulator isn't fighting gravity. The inverted configuration used in DragonBot can struggle when a singularity is encountered. And the problem of singularity analysis for parallel robots is far from solved. Last year, Jean-Pierre Merlet ran a summer school focusing on theoretical and practical research into handling the singularities that arise with parallel manipulators [66]. The singularity handling used in the implementation of DragonBot was described in section 3.3.1. However, this is still a very active research area, with active work on generating indices for singularity closeness, development of fast algorithms for detecting singularities, and algorithms for working space calculation using a given set of singularity regions.

While the robot's power system does a fantastic job of supporting long-term interactions with a 7-hour battery life, the robot's fur severely restricts the airflow to the electronics during operation. Therefore, if the robot isn't "vented" every 2 hours or so, the motors can overheat and the potentiometers begin to drift, leading to very unstable motion. While the parallel mechanism for controlling DragonBot's body restricts all the motors to a single stationary platform, it also requires that all motors are perfectly synchronized.

Because particular care must be taken with blended reality characters in preserving representations across mediums [51], a proper 3D character skinned with the same look as the physical unit was never developed. If the phone is removed from the current system, a virtual character is rendered on the display with a digital representation of the mechanical structure and a "floating head". Simulating and rendering the fluid motion of fur and rubber is beyond the scope of this thesis, but would only add to the potential for developing long-term relationships between children and synthetic characters.

Chapter 4

Software Framework

Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain.

–Alan Turing

This chapter outlines the fundamental software paradigms designed to support engaging long-term interactions with DragonBot. As mentioned previously, DragonBot is controlled through a modern smart-phone. While this removes the need for an offboard computer and an external internet connection, it also introduces a lot of challenges. Modern smart phones are not designed to support the paradigms in which many roboticists operate, and it's not possible to just "throw more hardware at the problem". Because of this, a lot of context-switching and computational delegation happens behind the scenes to ensure the robot's motion never drops below an acceptable framerate.

Even with these challenges, the smartphone gives us tremendous flexibility for long-term interactivity. The internet connection of a mobile device allows data to be pushed to and pulled from the cloud at runtime. We can also offload computationally intensive tasks to the cloud, receiving and integrating results as they are available. Finally, the availability of all the integrated components in a modern smartphone - a screen, a microphone, a camera, a speaker, an accelerometer - enable a wide variety of applications to be supported within the context of robotics.

4.1 Core framework

A software framework for a robot "in the wild" should allow its developers to quickly implement and test new high-level behaviors without requiring that a physical robot is present. This allows *anyone* to develop for the platform - not just people who have access to the hardware. For this reason, a cross-platform environment was created that allows the robot to run either physically or in a virtual environment.

For the purposes of rapid development and expandability, every piece of the DragonBot core code is connected to a top-level interface that allows any behavioral module to run on all implemented platforms. The currently implemented platforms include the native framework on the Android phone or as a simulated virtual character that can be run directly from the development environment. This allows the behavioral code to extend across platforms, making character development a portable process. The core functionality can be seen in Figure 4-1.

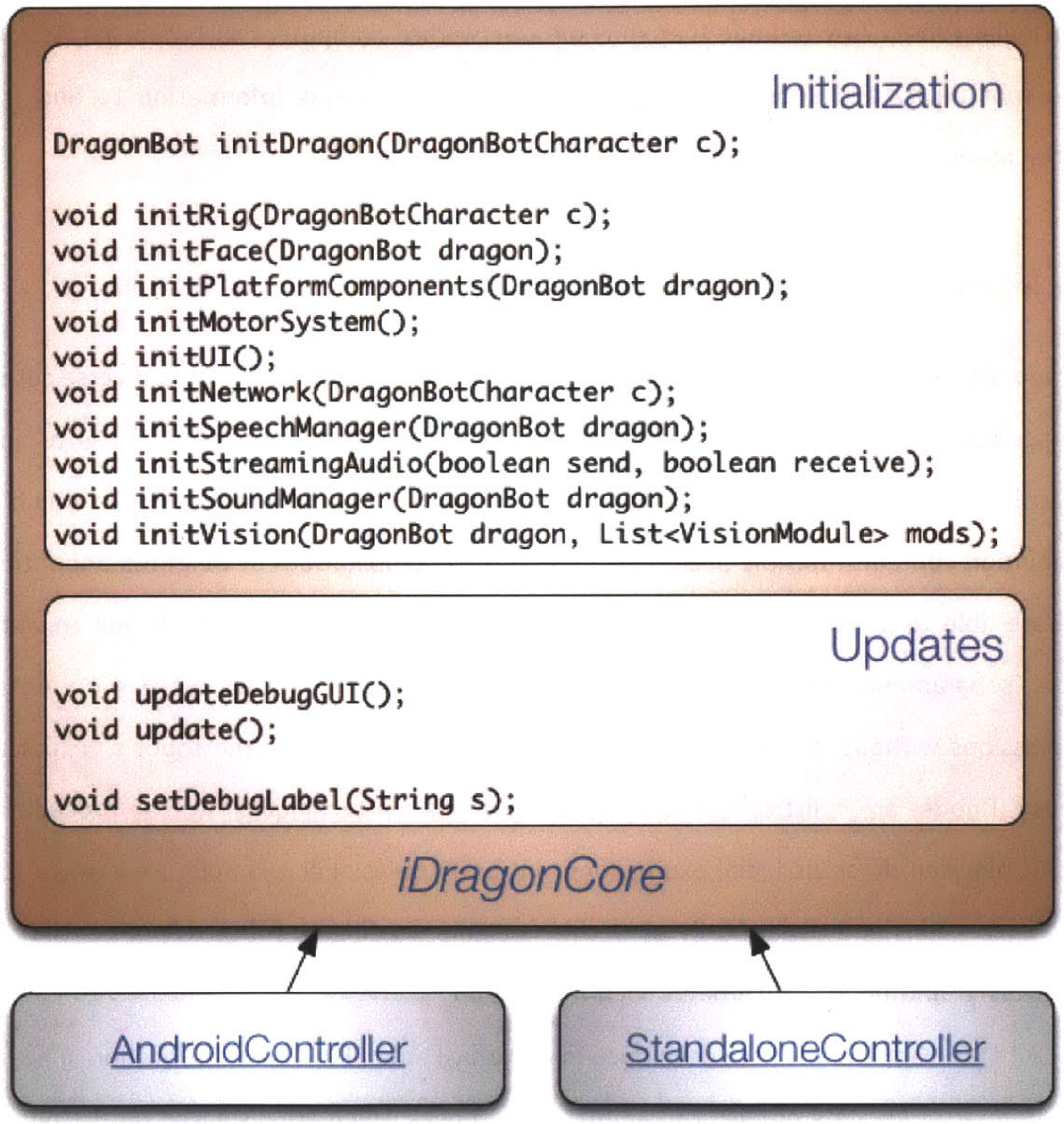


Figure 4-1: The core DragonBot interface

A python script was developed that quickly generates an Android project, compiles the code, and pushes the application bundle onto the device. This script also transfers any content-based assets such as images, animations, or sounds.

The implementation of the android device's primary sensors and device-specific settings were completed by Jesse Grey and Matt Berlin of IF Robots. The networking aspect is called IRCP, or intra-robot communication protocol. IRCP is a thin layer on top of UDP that abstracts away IP addresses and introduces robot IDs and module IDs for any particular robot. Any module that runs on a networked computer can broadcast its robot and module ID and subsequently start sending and receiving information to and from other modules on the same network.

4.1.1 Animation and Motor System

DragonBot's actions are entirely controlled through procedurally blending a diverse catalog of pose-to-pose and straight-ahead animations. A pose-to-pose animation is a single snapshot of the robot's physical state, while a straight-ahead animation is one in which the animation contains multiple poses over time. Using combinations of weighted animations, we are able to generate a vast set of dynamic behaviors for the robot during run-time. This is paramount for providing an extensible method of creating novel motions and expressions without recompiling any code. Furthermore, because the robot's virtual and physical nodes are holistically controlled using animation techniques, we can procedurally blend between different facial expressions and body poses. For example, we can combine two expressions and dynamically alter their weights, leading to robot responses that can be evolved and improved through each interaction.

All of the animations were developed in Maya by Fardad Faridi and the author. The face of DragonBot consists of over 180 animated 3D nodes which are displayed on the phone's screen through a two-dimensional orthogonal projection. The use of such a complicated facial rig within the robot results in a tremendous amount of expressivity possible without any moving parts.

The **motor system** of the robot encapsulates the procedural blending of many different animations to achieve lifelike and dynamic motions. The robot's low-level motor system consists of the following components:

Viseme Sets the mouth position of the robot

IK Directly sets the robot's pose

Lookat Moves the robot to look at a relative location in Cartesian space

Expression Controls the robot's facial expression, can be fully expressed or suppressed

Motion Handles the main "actions" of the robot, which are straight-ahead animations

Idle Handles the robot's breathing and tail wagging

Blink Causes the robot to blink

The higher an element is in this list, the higher it's priority is in updating. For example, a viseme request should *always* result in triggering a mouth movement - no matter what animation the robot is currently playing. Similarly, a blink event should only trigger if there is no expression or motion that is affecting the robot's eyes. Some sample facial expressions, eye positions, and visemes can be seen in Figure 4-2.

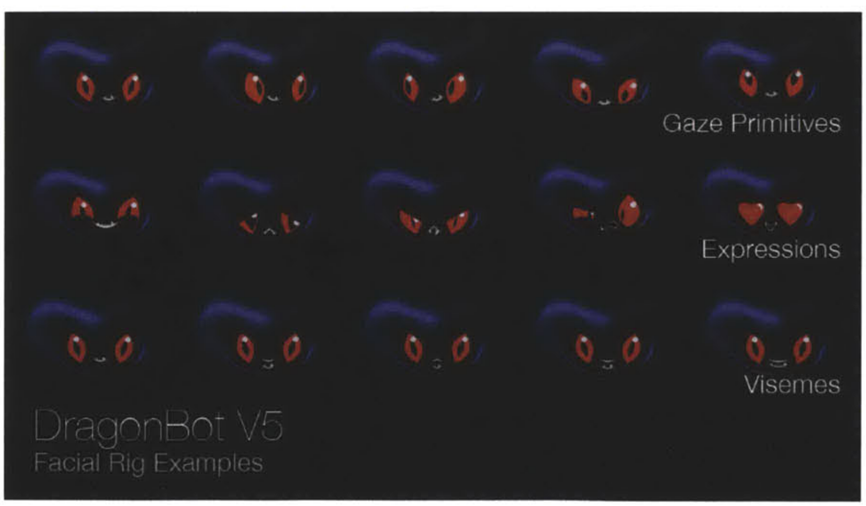


Figure 4-2: Example facial animations for DragonBot, including expressions, eye positions, and visemes

4.1.2 Emotion system

The emotion system of DragonBot was largely inspired by a presentation given by Doug Dooley at the MIT Media Lab. In his presentation, Dooley focused on applying animation principles for robotic characters to robots in the real world. His thesis was that combining a robot's motivation and emotional state leads to a compelling character. Similar to how actors consider their character's intentions, motivation forms the foundation of building a believable character. An agent trying to guilt will look much different than an agent trying to command.

However, motivation is not enough to make a compelling creature. The emotions of a character are what the viewer notices most. Integrating different emotional states with a character's motivation should lead to noticeably different behavior. Using the command example, different characters will want to command in varying ways. Some characters will command others with an emotion of happiness, while other characters will command with an emotion of anger or nervousness. How a character's motivations combine with their emotions defines who that character is.

A social robot must exhibit emotional responses to be a believable character. A robot that is capable of long-term social interactions should not only react with emotional responses, but it should store all interactions and allow for dynamic adaptation of the mapping from internal state to external action.

To be clear, DragonBot does not have emotions. Nor does any robot. And quite frankly, we don't really even know what the whole story with emotions is [67]. However, we definitely know that a person's emotional state has a strong effect on their outward behaviors, and that emotion is a fundamental principle in classical character animation [52]. Therefore, a social robot should be able to dynamically act and react in a way that humans in similar emotional states would behave.

The emotion-based action system of the robot can be seen in Figure 4-3. At the top level is the motivation manager, which handles the robot's high-level goals and plans.

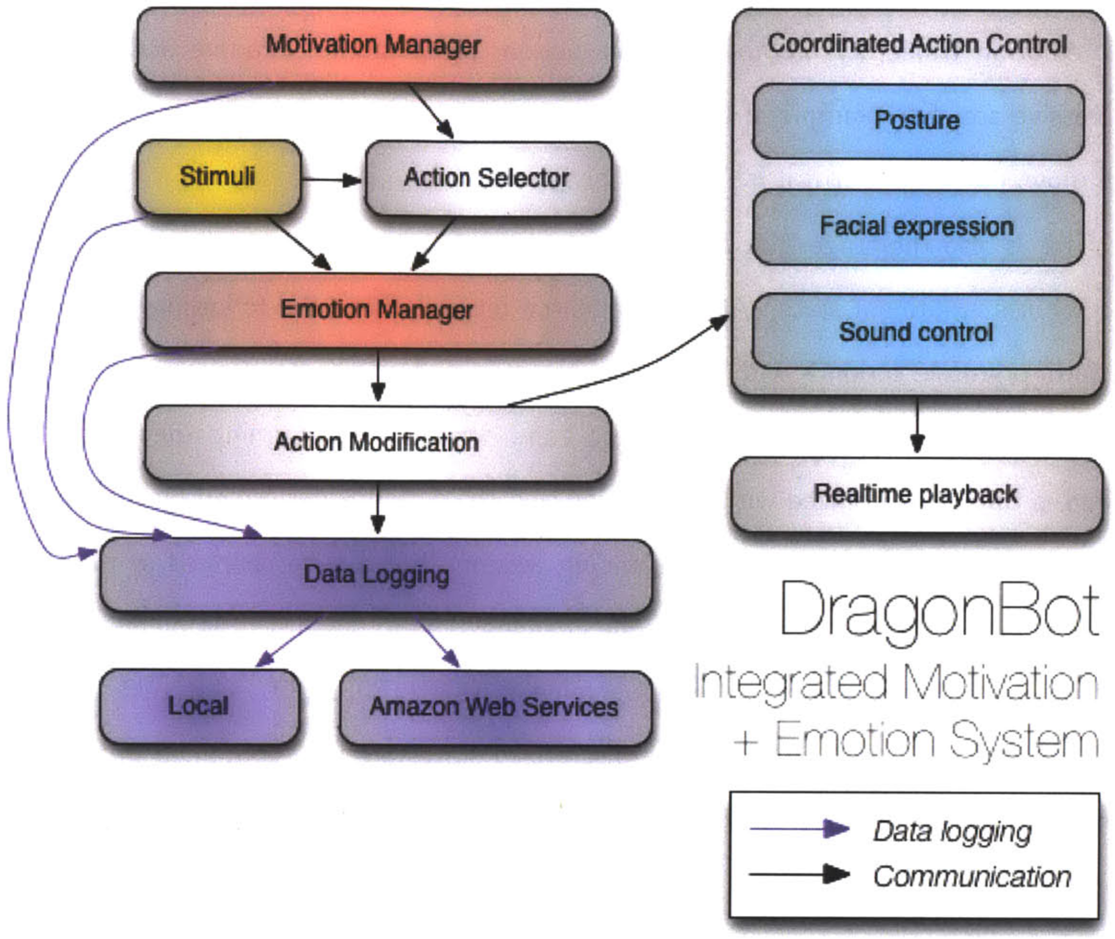


Figure 4-3: The structure of DragonBot's Emotion system

These motivations can be simple tasks like "find faces" or more complex ones like "make people smile". When stimuli are encountered they are run through a naive planner that selects an action to use given the current motivations. However, if the robot is being teleoperated, the motivation manager is disabled, as we are relying on the human operator to directly trigger the robot's actions.

After an action is selected, it is sent to the emotion manager which regulates a graph of the robot's emotional state over time. The emotion manager uses the emotion annotation representation language (EARL) XML schema, and it supports both categorized emotions (like "happiness" or "boredom") and emotions based on valence, arousal, and power.

Next comes action modification, which considers the primitive action selected and maps it to a new action by considering the current emotion. This step basically applies an "emotion trajectory" to the primitive action, blending the primitive action with an expressive arc. For example, if the primitive action is "lean towards object X", the robot will move much quicker if it is "excited" vs "bored".

This is also the stage that adds facial expressions and sounds to an action based on the robot's emotional state. At this stage, we basically have the following information: perform primitive action X with emotion Y by applying MAP(X, Y). We log all data about motivations, stimuli, action selection, etc. and commit to performing the action. The sound is time-synchronized with the motion and the action is performed by the robot.

The mapping function contains blending weights that can be changed at runtime to allow dynamic behavior without any code changes. The robot's list of implemented emotions is (sorted from positive to negative valence): `EXCITED`, `HAPPY`, `CURIOUS`, `NEUTRAL`, `SAD`, `FRIGHTENED`, and `ANGRY`. These discretized states are fully contained in the EARL schema, enabling the use of existing analysis software and allowing simple extension of the system with new emotional states.

4.1.3 Synthesizing a lifeform

Even though DragonBot is primarily intended for teleoperation, it's low-level behaviors are strongly inspired by biological creatures. A simplified nervous system was created for the robot, allowing it to behave as expected without any input from the operator. This system is responsible for things like making the robot jump away from objects that will collide with it, or triggering actions that convey the robot's illusion of life (such as breathing, blinking, or wagging the tail).

The structure of this hierarchical nervous system can be found in Figure 4-4. The most important module (and the parent node in the hierarchy of the nervous system) is the robot's emotional state. Within this module are specific submodules, all of which occur in the context of a specific emotion. For example, there is a module that makes the robot respond to the poking of its face, and one for making the robot sleepy when its low on batteries. But these behaviors will manifest in different ways based on the current emotional state. This enables the creation of dynamic behaviors simply by having the current emotion (set by an operator) trickle through its submodules without any intervention. This makes implementing new behaviors straightforward, as every submodule in this hierarchy is solely dependent on *the context of the robot's current emotional state.*

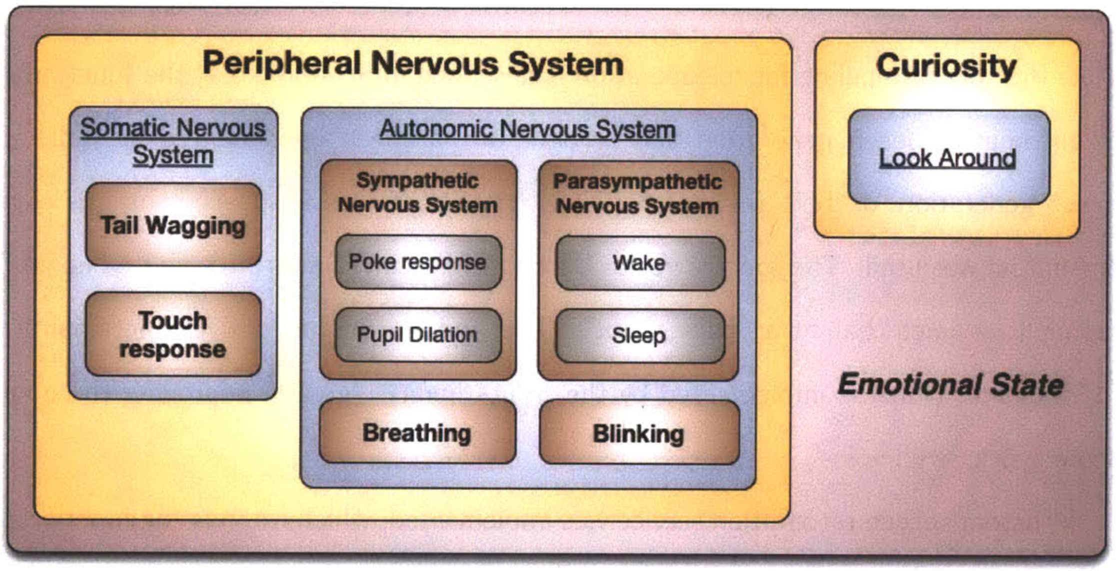


Figure 4-4: The modules and hierarchies that comprise DragonBot's lowest-level behaviors

The largest submodule is the robot's "Peripheral Nervous System" (PNS), which is structured and inspired from a mammal's peripheral nervous system. The first submodule in the PNS is the Somatic Nervous System (SoNS), which is responsible for all low-level voluntary actions. For DragonBot, this includes responding to touch and modulating the speed and frequency of tail wagging. The PNS's second submodule is the Autonomic Nervous System (ANS), which controls the involuntary actions of the robot. As in humans, the ANS regulates the robots blinking and breathing, and contains a Parasympathetic Nervous System (PSNS) and a Sympathetic Nervous System (SNS). The PSNS submodule controls the functions of the body when it is at rest, and overrides the robot's behavior when receiving a wake or sleep command. The SNS submodule controls the involuntary "fight or flight" actions during times of arousal, like responding to pokes and appropriately dilating pupils when excited or bored.

Each submodule can support an arbitrary number of emotional states. For example, the Curiosity module is only triggered when the robot's emotional state is `CURIOUS`, but the Breathing module is affected by any emotional state.

4.1.4 Speech systems

While the speech pipeline for teleoperation is described in Section 4.3.1, the functionality for automatic speech generation and recognition has been implemented as well. For speech generation, or Text-To-Speech (TTS), the SVOX Classic speech synthesis engine for Android was used. The specific voice (US English "Benny") was pitch shifted up 140%, leading to a voice that fit a "baby robot". The Android OS has native functionality for TTS built in, which is implemented by the SVOX library - greatly improving the quality of the synthesized voice.

A custom speech recognition service was implemented, which extends the functionality of Android's built-in speech recognition capabilities through Google. Typically, applications are able to use the speech service with a small dialog that appears on the phone. However, because the phone is acting as the robot's face, this was not a suitable solution. Therefore, the API calls to the Google speech recognition service were reverse engineered, and a custom background service interfaces with the Google servers directly. This background service forwards any results to the main application, and does it without a dialog window covering up the robot's face. This speech recognition functionality requires the phone to have an internet connection, as the Google cloud-based services are used.

4.1.5 Blended Reality

While DragonBot is focused on physically coupling the smartphone with the robot as the "face" of the character, it's entirely capable of running as a purely virtual character on the screen of the phone. This ability to have a "blended-reality" character [51], one that can seamlessly transition between a virtual and physical representation, allows for longer-term interactions. Allowing a user to have access to their personalized character "on the go" opens up the possibilities for interactive conversations even when the physical robot isn't physically accessible. A rendering of the Kombusto character on a cellphone can be seen in Figure 4-5.

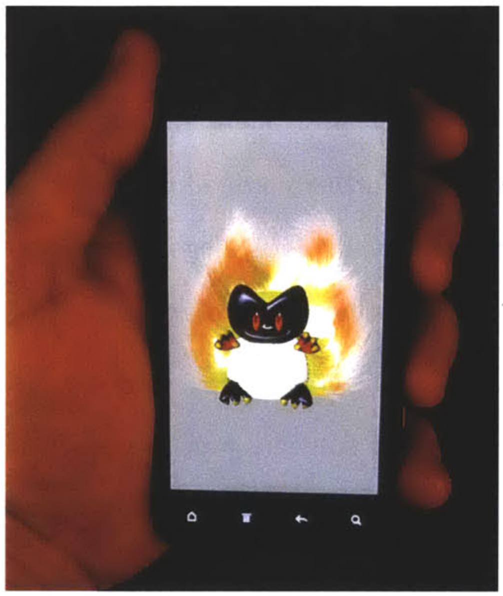


Figure 4-5: Kombusto as a fully-virtual character

The primary sensory inputs to DragonBot are the Android phone's front-facing camera and the microphone, and these are available whether the robot is embodied or purely virtual. All behaviors still stem from animations, so the actions of the creature are consistent in both the virtual and physical realms, and the character's motion remains cohesive. However, because the computational modeling and rendering of "fur" is not able to achieve realtime performance, the blended reality capabilities of our robot were not finished. Because the model of the virtual robot had to be simplified in comparison to the physical character, it destroyed the cohesiveness of the character when switching mediums.

4.2 Data collection and load balancing

Every time something needs to be logged - whether it be a command from a human, an audio data stream, or a face detected in a vision frame - the data is sent to a static logger that gets initialized at launch. Using this singleton design pattern, any piece of code on the robot can log relevant data. The data type of the requesting module is appended to the log, along with a timestamp. When the robot receives a command from a new module, a time-synchronization event is stored for all modules, enabling quick analysis of the data without having to manually align all of the streams.

The logger works on video streams, audio streams, captured images by the robot, Kinect data, and EDA activity. Furthermore, all of the robot's actions, and any teleoperator commands, are similarly saved. These logs can be stored locally or automatically uploaded to Amazon Web Services for offboard analysis.

Any time a new module is registered on the network, the robot queries the module for its supported subsystems. The software framework offloads tasks from the robot to external subsystems if possible, attempting to balance the computational load among all devices. For example, the robot will run a Viola-Jones facial detection algorithm every 100 frames if there are no other computers on the network. While not computationally expensive, analyzing an image every frame can slow the robot's motion to an unacceptable level. However, as soon as another module is available on the network that supports facial detection, the robot streams the data to the computer and receives high-level results every 3 frames.

4.3 Auxiliary Systems

This section describes the auxiliary systems that enhance DragonBot's onboard capabilities.

4.3.1 Teleoperator speech pipeline

First, let's define a few terms. A **phoneme** is a basic element of sound for a spoken language or dialect. A phoneme lives in the *acoustic* domain - it is the smallest segmental unit of sound employed to form meaningful contrasts between utterances. A **viseme** is a representational unit used to classify speech in the *visual* domain. A viseme can be thought of as a mouth position. The DragonBot facial rig supports nine distinct visemes, with the following sounds: `AA/AH`, `AO/AW`, `CH/SH/ZH`, `EH/AE/AY`, `EY`, `L`, `M/B/P`, `N/NG/D/Z`, and `R/ER`.

At the highest level, the teleoperator speech pipeline takes a stream of audio data (the teleoperator's voice), guesses the current **phoneme** of the speaker based on the speech input, and estimates the current **viseme** given the most likely phoneme. Finally, the speech signal is shifted so the person's voice becomes a higher pitch. The pitch-shifted audio and the current viseme are sent to the robot, allowing DragonBot to talk in a character-like voice with appropriate mouth movements.

The phoneme and viseme estimation was made possible through use of the Realtime Lipsync SDK created by Annosoft [68]. This C++ based software, which runs on a Windows PC, works best when the user is wearing a headset with a high-quality microphone positioned approximately 6 inches from the mouth. The source of the Realtime Lipsync SDK was augmented with a library that wraps the IRCP communication protocol, allowing the windows application to send data directly to the robot. Finally, the pitch-shifting of the audio signal is accomplished by using a Short Time Fourier Transform for changing the perceived pitch of the signal by representing it as a sum of sinusoids and scaling their frequency.

4.3.2 Augmentation through RGBZ sensor

The KinecTable's primary sensor, the Microsoft Kinect, is capable of capturing RGB images and 3D depth maps (RGBZ). As a time-of-flight sensor, the Kinect measures the depth of a scene by quantifying the changes that an emitted infrared light signal encounters when it bounces back from objects in a scene. As a wrapper on top of this, the OpenNI module's NITE algorithm is able to detect and track people in the scene using a proprietary skeleton-tracking algorithm. The skeletons, along with the raw data, are used as inputs to the powerful Kinect framework developed by Jin Joo Lee [10] which correlates the RGB images with depth data and skeletons. Furthermore, this framework integrates logging capabilities with real-time access, enabling data to be analyzed and quickly saved to disk. Additionally, the Kinect is able to determine the direction of the loudest sound in the environment using a linear array of microphones.

A module was developed that uses the aforementioned capabilities of the Kinect to enhance the robot's perceptual capabilities. This module was designed specifically around the task presented through our evaluation (described in Section 6), and provides four capabilities:

1. Logging of correlated data (RGBZ + skeleton)
2. Color blob-tracking of five colors
3. Detection of loud sounds
4. Detection of when a child is touching the robot

It should be noted that the color blob-tracking was finely tuned to the illumination conditions of the room and a set of five very distinct colors.

4.4 Communication between modules

We conclude this section by describing the communication of the different modules in the DragonBot system. We also include relevant information regarding the networking scheme used in the first DragonBot data-capture study (described in Section 6).

Although the robot functions without any of the auxiliary systems, there are four modules that are tied together to provide the full capabilities presented in this section: the robot, the KinecTable, the Teleoperator tablet, and the SpeechBox. A single consumer-grade 802.11N wireless router was used to bridge these four systems without any significant packet loss. As minimization of delay is such a central issue in quality of teleoperation [34], the KinecTable and SpeechBox - both desktop computers - communicated with the router using a wired ethernet connection. The robot and the teleoperator tablet both connect to the wireless router, using the WPA2 security protocol.

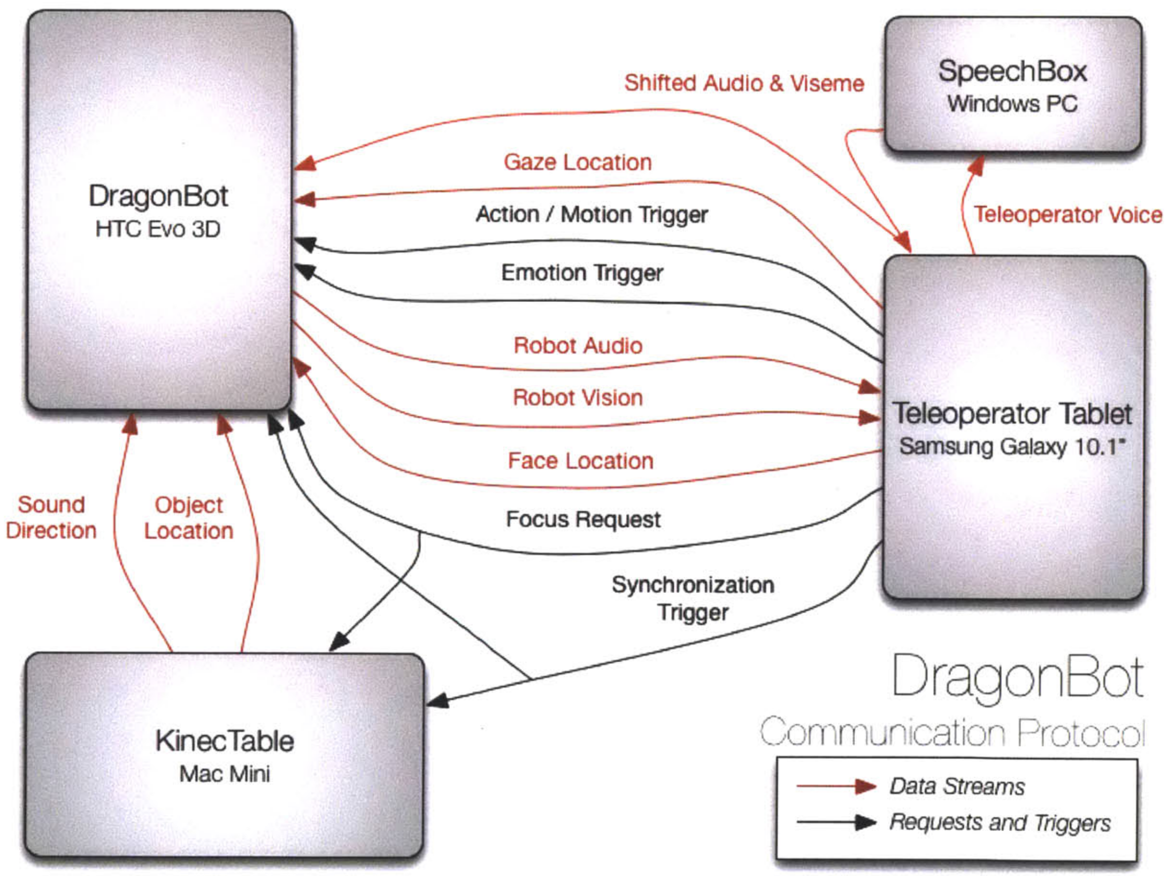


Figure 4-6: Communication scheme between all the working systems

The communication scheme between these submodules can be seen in Figure 4-6. The protocol consists of two high-level data types, **Data Streams** and **Requests and Triggers**. A trigger is simply that - it signals another module to change its state without expecting any return communication. However, a request is a way to notify one module that it's data stream is desired by another module. Upon receiving a request, the module begins streaming it's packets to the module that requested the data.

Chapter 5

Teleoperator Interfaces

This chapter describes the two teleoperator interfaces that were designed to control DragonBot. The first interface, which we will refer to as the **Manual Controller**, gives the human operator nearly full control of the robot's actions. The second interface, the **Shared Interface**, presents a novel method for sharing autonomy with a socially expressive robot. The **Shared Interface** was designed because a remote robot operator having a conversation should be primarily focused on the social aspects of the interaction, not on controlling the robot. If most of the operator's cognitive load is focused on the social dynamics, we can allow the robot to take initiative through intelligently selecting appropriate actions based on the current context. The terminology controller and interface are used explicitly - the **Manual Controller** is just that, a direct and exclusive method for triggering actions on the robot. However, the **Shared Interface** is simply a point of interaction between components - the human operator and the robot's autonomous capabilities.

There are many ways a robot can augment a human's capabilities during teleoperation, which were outlined in Chapter 2. The specific autonomous behaviors that were implemented to supplement the human's control of DragonBot are:

1. intelligently focusing the robot's gaze (Section 4.3.2)

2. integrating the robot's emotional state into its outward representation (Section 4.1.2)

3. expressing emotions and actions when appropriate

4. mirroring the child's body language and facial features

We first describe the common points between the two interfaces, followed by a description of the unique elements from each design.

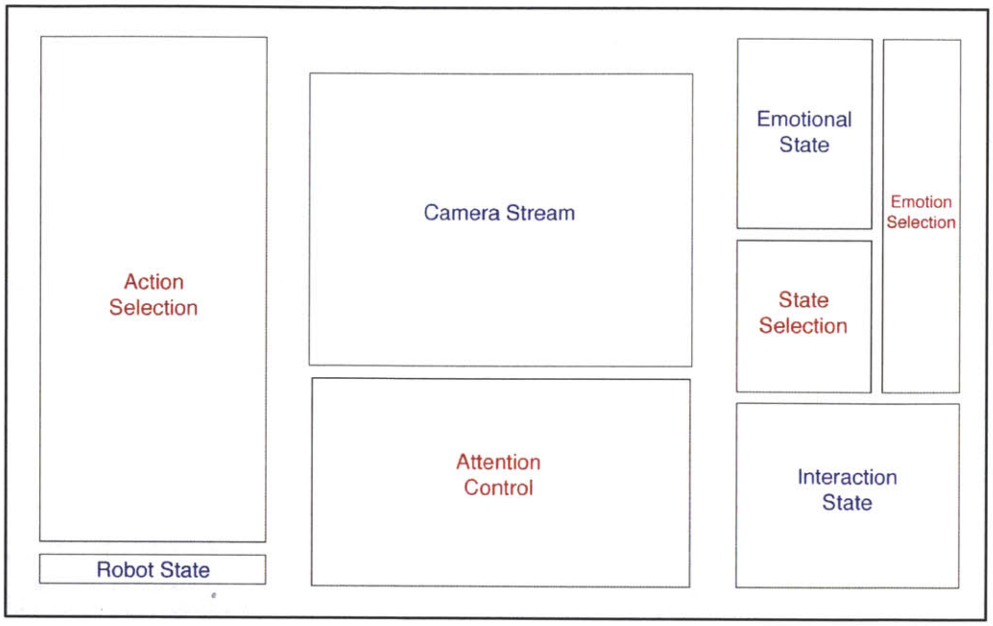


Figure 5-1: The tablet-based layout common to both teleoperator interfaces

5.1 Commonalities

Both robot control interfaces were designed as tablet-based applications, using a Samsung Galaxy 10.1 running Android 3.1. The portability of tablets offer more flexibility to the robot operator than a desktop application, but provide the interaction area needed for on-the-fly control of a robot's actions. Both control schemes were designed from the start to have a nearly-identical interaction from the operator's standpoint. A holistic layout was first designed that groups the desired information and controls for both schemes into logical chunks that are appropriate for tablet-based teleoperation of a social robot. This layout, which can be seen in Figure 5-1, highlights the robot's vision stream and the widget for attentional focus by centering them in the middle of the screen.

The layout also contains input areas for selecting an action, selecting an emotional state, and for triggering general actions that are specific to the current task. Output groups exist for displaying the state of the robot's motion, the state of the robot's emotional system, and any contextual information specific to the task.

While the individual components inside each of these logical groups differ, the locations of the groups are consistent across both interfaces. Furthermore, the look and feel of both applications is nearly identical, using the same color schemes and graphical assets where possible.

The Camera Stream area always contains a black and white snapshot of the robot's vision stream, updating at 20 FPS. If the android phone had an extra core (or there was nothing else happening on the phone), a 30FPS color stream could be used. However, for practical use of the robot in the field, the black and white solution is much more appropriate.

The state selection area contains action buttons specific to the current task. In the case of our experimental analysis, these were simply a prologue and an epilogue. The interaction state area displays context-specific information for the current interaction. For example, the overall interaction time is always displayed here, as well as gaze locations and any potential connection problems with the robot. Finally, the Robot State area displays the current action that the robot is performing.

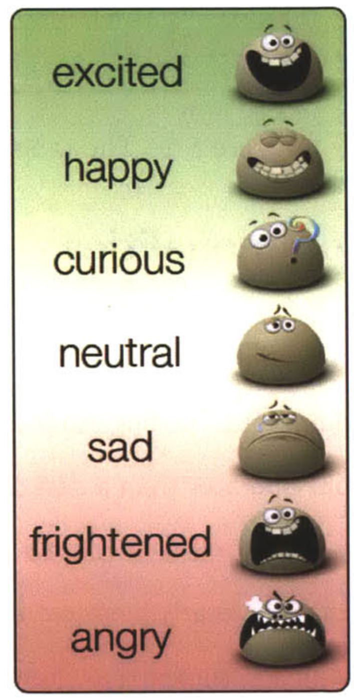


Figure 5-2: The valence-based scale of emotional states available to the operator

5.1.1 Emotion widgets

The widgets for selecting and viewing the emotional state of the robot are consistent between both interfaces. These widgets consist of selection (a vertical slider that lets a user choose the robots emotional state) and state (a display widget that notifies the user of the current emotional state of the robot). When the operator touches the selection widget, the state widget updates with a preview of the currently selected emotion. However, the emotion is only sent to the robot when the operator removes their finger from the screen. This allows the operator to quickly preview and select emotions, without the risk of sending an inappropriate emotion to the robot.

The slider has positive valence at the top and negative valence at the bottom, and the emotions (and icons used to represent emotional states to the operator) can be found in Figure 5.1. The ways in which the robot's current emotional state changes its behavior are discussed in Section 4.1.2.

5.2 Manual Controller

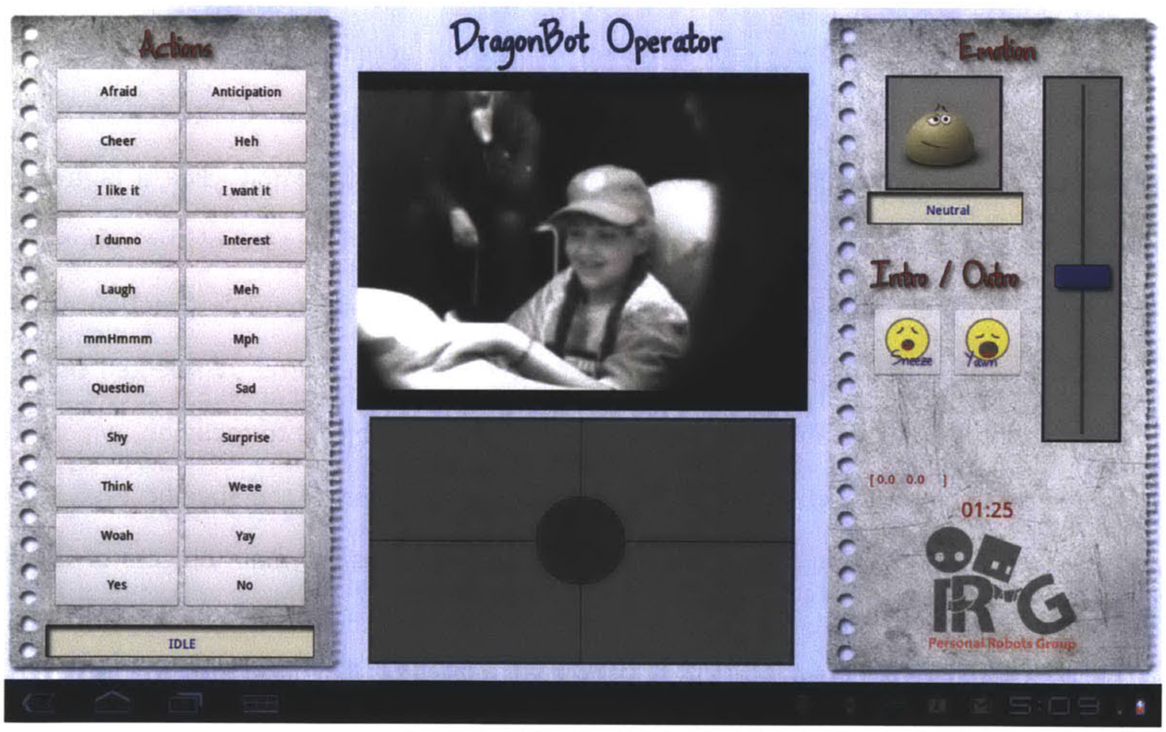


Figure 5-3: A screen-grab of the DragonBot Manual Controller

Let's now discuss the unique aspects of the **Manual Controller**, which can be seen in Figure 5-3. The Attention Control area has a manual gaze widget. When the user touches this area, the circle and crosshair follow the user's finger and send gaze commands to the robot based on the X and Y position of their finger. 1250 milliseconds after the operator releases their finger, the robot's gaze is reset and the manual gaze widget flashes to indicate a reset. The Action Selection panel is simply a grid of 22 alphabetically-organized text buttons that directly trigger the corresponding motions on the robot. When a user touches one of these 22 buttons, the corresponding explicit action is sent to the robot. Finally, when an emotion is selected, the robot uses a simple facial expression to signal its change in emotional state.

5.3 Shared Interface

Now we turn our attention to the **Shared Interface**, which can be found in Figure 5-4. Both interfaces have the exact same set of possible actions, but they are organized and evoked differently using the Shared Interface. The Attention Control area has a simplified "Attention Widget" that enables the teleoperator to simply click a high-level object. Instead of relying on the operator to directly control the gaze of the robot, the robot's facial detection module (Section 4) and auxiliary RGBZ module (Section 4.3.2) are used by the robot to automatically direct attention to the currently-selected object. The Attention Widget enables the robot with six high-level objects to focus on, which are triggered by a touch event from the teleoperator. In the case of the specific evaluation for this thesis (see Section 6), we used five animal stickers as the main objects of focus. In addition, the sixth object of focus is the child's face, and there is a "cancel" button in the upper right corner which sets the robots focus to "explore mode".

The front-end of emotion selection is common to both interfaces. However, when changing the robot's emotional state within the shared controller, the autonomous aspects of the robot's emotion system and nervous system (described in Sections 4.1.2 and 4.1.3) control which actions are selected. This can include facial expressions, body posture, pupil dilation, the speed at which the robot wags its tail, and overall speed of performing motions. This results in the emotional state of the robot to be persistently observable to

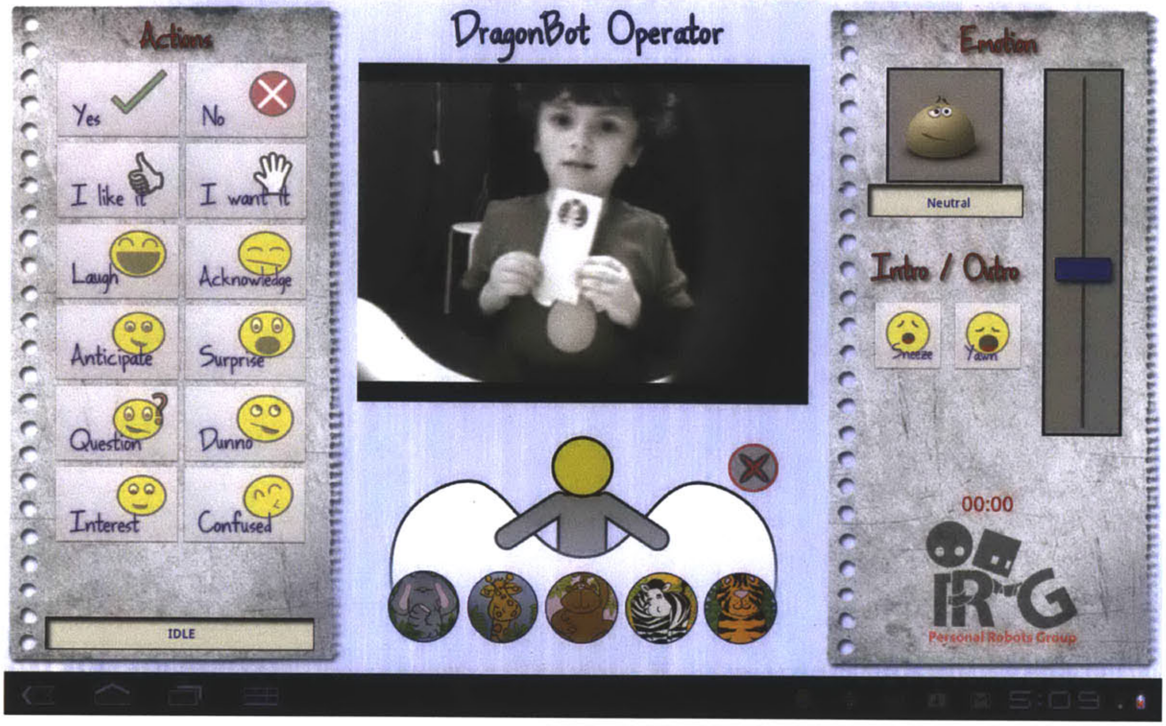


Figure 5-4: A screen-grab of the DragonBot shared interface

the user, and it also adds variability to the interaction.

The Action Selection area of the shared interface is implicitly clustered to make the selection process easier for the teleoperator. The 12 actions map to the exact same 22 actions from the Manual Controller, using the contextual state of which the robot is in to provide a more dynamic experience. The 12 action buttons use simple smiley-face emoticons to indicate their meaning, and they are logically grouped by row. These buttons do not directly trigger motions on the robot, but there is a set of possible motions that are triggered from each action. The robot attempts to use a novel motion from the set if it hasn't been used recently, and then picks randomly once all motions have been performed. The mapping from action to motion can be found in Figure 5-5.

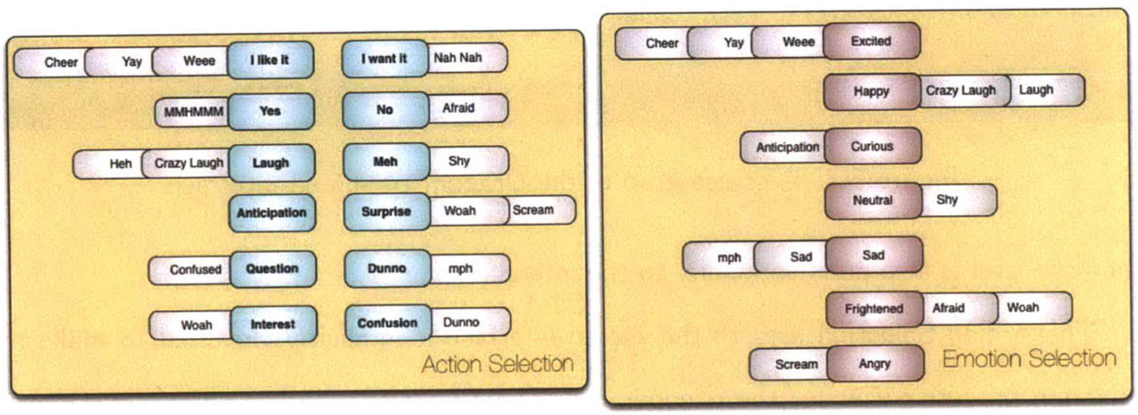


Figure 5-5: The way actions and emotions map to expressive motions within the **Shared Interface**

Chapter 6

Experimental Design

6.1 Introduction

There were multiple options to choose from in terms of which aspects of the DragonBot system to evaluate. We could have deployed a robot in the field for a long-term interaction with a few children. However, it was decided to run an in-house evaluation with a larger population size. Running the first experiment in-house lets us test the data-capture framework and ensure proper performance with a sufficient network. It also lets us carefully control the remote-operator interface to narrow down the correct paradigms for teleoperation of a social robot. And most importantly, we can see what a lot of kids think of the robot.

The scenario that we chose to evaluate was one that would tell us about the usefulness of shared autonomy for operating socially expressive robots. The analysis is **both** looking at the child's behavior as well as the remote operator's ability to control the robot. We decided to run a between-subjects experiment where the participants would have a conversation about five animal stickers with the robot. After talking about the stickers for approximately 10 minutes, the child will be prompted to indicate their three favorite stickers, followed by the robot. The robot will always overlap with one of the child's favorites, and then the participant is prompted to split up the stickers for themselves and the robot to keep.

The two conditions of the study are:

Condition M the operator is using the **Manual Controller** to directly control the robot's actions

Condition S the operator is using the **Shared Interface**, working in synchrony with the robot's autonomous modules to produce the appropriate behavior

The details of the experiment are further explained in Section 6.3.

Before running the study, we posited the following hypotheses:

1. The children will recall more stickers from interactions with the **Shared Interface** than with the **Manual Controller**

2. The children will share more stickers with a robot being operated through the **Shared Interface** than through the **Manual Controller**

3. The operators will prefer the **Shared Interface** over the **Manual Controller**

4. The operators' engagement in the interaction will be higher with the **Shared Interface** than with the **Manual Controller**

5. The operators' cognitive load will be lower with the **Shared Interface** than with the **Manual Controller**

6. The operators' arousal will be lower with the **Shared Interface** than with the **Manual Controller**

6.2 Relevant Measures

There were four relevant measures to the teleoperator: Cognitive Load, Social Presence, Engagement, Ease of use of the Platform, and Arousal. Furthermore, there are two relevant measures for the participant: recall of robot's favorite stickers and sharing behavior.

6.2.1 Cognitive load

It is very important in the evaluation of a teleoperation system to pay attention to how difficult the control interfaces were to use. It's essential to minimize the cognitive load on the operator, as this will allow them to focus on the social interaction. For evaluating the cognitive load experienced while operating the robots, the standardized NASA-TLX [69] task-load index questionnaire was used. Furthermore, specific questions were asked in the operator questionnaires that attempt to figure out which subset of the interface was too cognitively demanding. The number of touch events registered on each interface can also be used to gain some insight into the operator's overall load.

6.2.2 Social presence

One of the main goals of this thesis is to provide an embodied medium for long-distance communication. Social presence attempts to measure the level of "being there" that a remote operator experiences. This idea - letting people be present in a space that is separated from them by distance - is an important one for long-term interactions between children and human-operated robots. Social presence has frequently been used as a quality measure of different communication media [70], [71]. For evaluating this metric for the robot's operator, the Networked Minds Social Presence [72] questionnaire was used.

6.2.3 Engagement

The engagement of the operator is important to ensure a similarly engaging experience for the child. A set of questions around engagement were compiled using inspiration from Van Baren's work on Measuring Presence [73].

6.2.4 Ease of use of the platform

The ease of use of the platform is very relevant for allowing non-technical users to operate the robot. A set of questions was constructed that asks the operator about any distractions or complications encountered during use. These issues are things like motor noise, quality of the video and audio, and the ability to control the robot's gaze.

6.2.5 Arousal

The primary way we will determine the operator's arousal is using a skin-conductance sensor that measures the operator's Electrodermal Activity (EDA). Coupled with the video stream of the operator's face, the set of touch interactions with the interface, and the self-report questionnaires and interviews conducted on the operators, the EDA signal now has a good deal of contextual information, which is essential in correctly interpreting data from most biometric signals.

6.2.6 Child-dependent metrics

The child will be asked to recall the robot's favorite stickers after the interaction. This will give some indication of how much the child remembers from the interaction. Furthermore, the child's sharing behavior - which stickers were given to which participant - will also be used to indicate how much the child "liked" the robot, and whether or not the robot is being treated as a peer.

6.3 Method

6.3.1 Participants

The call for participation in the study advertised for 4 - 7 year olds, and recruitment was done through the Boston Museum of Science, MIT daycare, and the Harvard School of Education. A total of 28 people participated in the study, and four subjects' data were excluded leaving an 86% success rate and a total population size of $n = 24$ participants. The average age was 5.11, and the median age was 5, and the age distribution can be found in Figure 6-1. There were 19 boys and 9 girls.

It should be noted that all four of the omitted participants were four years old. These participants were excluded for the following reasons:

- Participant 7 could not concentrate on talking to the robot - he mainly kept "roaring" and complaining that he was hungry.
- Participant 11 had fun and engaged the robot, but wouldn't complete the task.

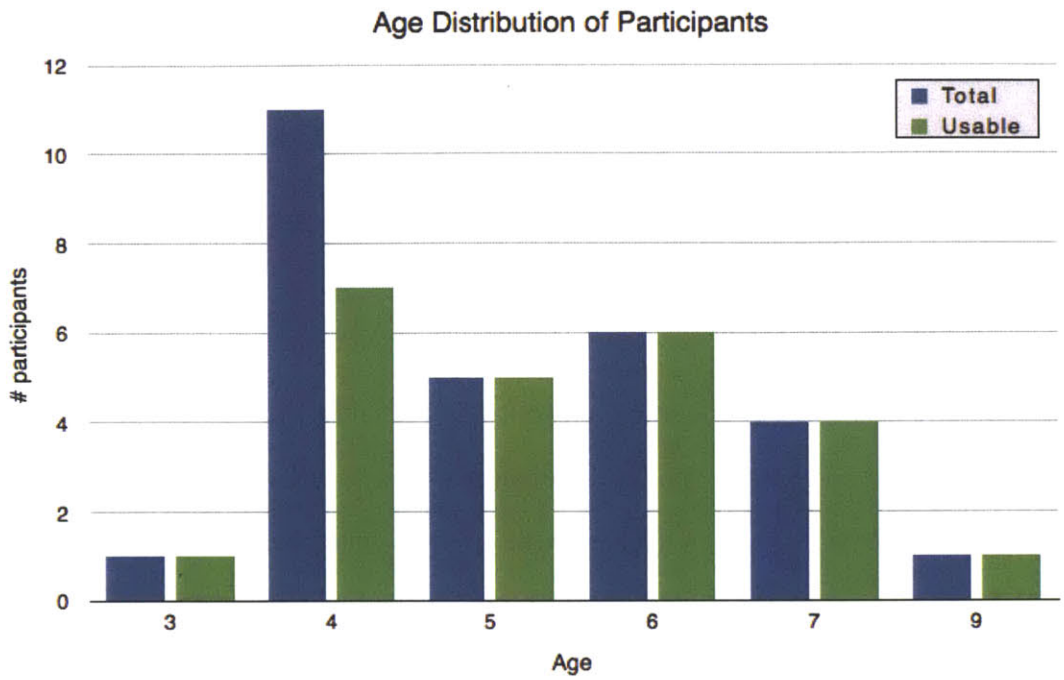


Figure 6-1: Age distributions of the participants

- Participant 15 would not stay interested in the interaction, although he really liked the robot. His interactions with the robot were the most tangibly-oriented of any participant. He couldn't complete the task.
- Participant 17 would not engage the robot as a social actor, and barely spoke a word. In a post-interview, she said she was afraid of dragons.

6.3.2 Teleoperators

Three teleoperators were used to test the interfaces. The operators first filled out a survey with questions relating to their background and video game habits. Also, they were asked questions on a seven-point Likert scale regarding their familiarity with computers, familiarity with robotics, and comfortability in social situations. The resulting responses can be found in Table 6.1.

Table 6.1: Teleoperator backgrounds

Question	**Operator #1**	**Operator #2**	**Operator #3**
Level of Education	College Degree	Some college	Some graduate school
How much do you know about robotics, compared to the average US citizen?	5	3	5
How much do you know about computers, compared to the average US citizen?	6	7	5
How comfortable do you typically feel in social situations when you meet somebody new?	6	5	6
How many video-games do you play in a week?	Zero	More than 2 hours a week	Zero

Contrary to the norm in teleoperator training, the operators were not allowed to use the interfaces before their first interaction. It's quite important that non-technical users are able to operate DragonBot, and it was desirable to have the "first-time" interaction on record to see how quickly people are able to learn the control paradigms. The operators were given screenshots of the two interfaces 72 hours before their first interaction, so they could familiarize themselves with the layouts and different components. They were also given the interaction script, which can be found in Section C.1.

The operators were told the following immediately before their first interaction:

The main task through the interaction is to have an engaging conversation with the child regarding a set of stickers on a tabletop. While having this conversation, you should direct the robots gaze to share attention with the childs. Also, you should express contingent behaviors and mirror the affect of the child when appropriate. In short, try to be an engaging playmate for the child through the robot.

6.3.3 Task

The task consists of a tabletop sticker-sharing scenario between the participant and the teleoperated robot. When the parent and child arrived in the staging area, the child was told: "You're going to be talking to a robot today, and they really love pictures of other robots. Can you draw a picture of your favorite robot?" While the child was drawing this picture, their guardian filled out consent forms and a video release. The participant's name was told to the operator, so the robot could immediately capture the child's attention by knowing their name. The child was told the robot is ready to see them, and they were allowed to bring their parent if they didn't want to go alone. They were also told that it was completely fine if they wanted to stop at any time during the interaction.

Figure 6-2: The interaction setup with the robot

The robot's interaction room, which can be seen in Figure 6-2, was made to be a comfortable space for kids. In addition to colorful robot stickers on the walls, the room contained a "glowing dragon egg", a few festive plants, and brightly-colored chinese lanterns overhead. One wall is a chalkboard, so there were colorful pictures drawn all over. Another wall was corkboard, and the child's robot picture was immediately hung up on the wall. There were two extra seats behind the child - one for the experimenter, and one for a parent.

When the child sat down, the robot asked "Ooh hi there! Are you `NAME?`", inserting the specific child's name for `NAME`. After the child responded, the robot followed up with: "Its great to meet you, `NAME`! Im so excited that youre here to visit me! My name is Pepper." The experimenter than said the following: "Here are a set of 5 stickers for you and Pepper to share with each other. Why don't you take a few minutes to talk about the stickers with each other? Maybe talk about which stickers you like or dont like to help you decide how you might want to divide them up."

Figure 6-3: The five animal stickers used in the experiment

At this point, the experimenter places the five animal stickers, seen in Figure 6-3, on the table. This marks the beginning of the open-ended interaction, which lasted between 5 - 15 minutes, depending on the engagement and talkativeness of the participant.

After talking for a sufficient amount of time, the experimenter asked the participant to say which three stickers they liked the best. Pepper was then asked the same question, and the operator always picked the two stickers that the child didn't pick **and** a single overlapping sticker from the child's preference. The child was then asked to divide the stickers between themselves and Pepper. After dividing, the robot behaved appropriately based on the number of stickers it received. The full interaction script can be found in Section C.1 for reference.

The robot thanked the child for visiting, and the child was escorted back out to the staging area away from the robot. The following questions were asked of the child:

- Did you have fun?

- Do you think the robot had fun?
- Which Animals did the robot like best?
- Why did you split up the stickers the way you did?
- What was your favorite part about Pepper?
- Was there anything you didn't like about Pepper? I promise not to tell...
- Do you think Pepper has feelings?
- Would you want to talk more with Pepper in the future?

For participants who didn't understand the question, "Do you think Pepper has feelings?", the question was clarified by adding "do you think Pepper can be happy or sad?"

Figure 6-4: The wide-angle view of the interaction

6.3.4 Complicating factors

Because of a technical failure early in the study, the operators did not randomize their interactions between the Manual and Shared interfaces. At first, the manual controller was used, followed by the shared interface. At the end, the manual controller was used a final time, followed by a last round with the shared interface.

Another complicating factor, which will help explain some results in the following section, is that of *re-engaging* the robot during the sticker sharing. When we say *re-engaging*, we mean asking the robot it's preferences after the child has been asked to split up the stickers. Because we hadn't anticipated the children to re-engage the robot, the teleoperators weren't given any instructions as to the correct way to handle this situation. The first time a child asked the robot if they wanted the shared preference sticker, the operator said "You should take it, you liked that one!". After this, the operators were instructed to encourage the child to take the sticker if re-engaged during the delegation of stickers.

6.3.5 Operator Questionnaires

The operators filled out short questionnaires after every interaction and long questionnaires after their first, middle, and final interactions. See Section C.2 for the full questionnaires. The operators were also verbally interviewed after random interactions, to gain more insight into their experiences operating the robots.

Chapter 7

Results and Discussion

The results of the evaluation, as well as a discussion of their significance, are presented in this section. The measures we will use to evaluate our hypotheses from Section 6.1 are:

1. Length of interactions
2. Self report questionnaires/interviews of the operators
3. Self report interview of the participants
4. Video captured from the interaction
5. Electrodermal activity from the robot operator
6. Logs from the tablets that contain a list of all the commands issued by the operator
7. The # of stickers the participant gives robot, and whether any were shared preferences
8. The set of stickers the participants recall as the robot's favorites

Although 28 children participated in the experiment, four were excluded from the analysis (discussed in Section 6.3.1). This leaves a total of $n = 24$, with $n_{manual} = 14$ and $n_{shared} = 10$.

7.1 Teleoperator's Perspective

The questionnaire results were analyzed using a method called Analysis of Variance (ANOVA), a fairly well known method to determine if samples from two groups actually originate from the same population or if there is a statistically significant difference in their means. One of the outcomes of this analysis is the p value which tells us the likelihood of this particular outcome of differences in the group means. If p is close to 1 then there is high likelihood that this difference would show up at random, if $p \leq 0.05$

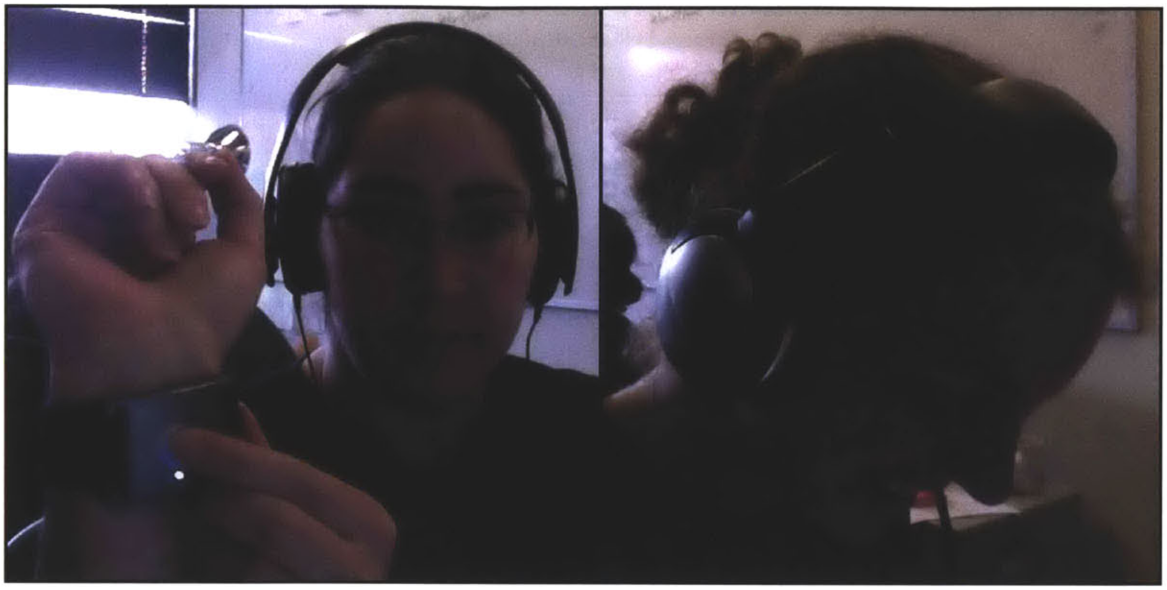


Figure 7-1: Video frame from a teleoperator synchronizing the EDA sensor (left) and during interactions (right)

then there is less than 5% probability that this result could be caused by chance. The only processing performed on the questionnaire data is that some of the scales were reversed for the questions which require a low score to signal a more desirable outcome.

Table 7.1 contains the means, standard deviations, and cross-conditional p-values from the general questionnaire. The questions are asked on a seven point Likert scale.

The results from the general questionnaire show that both groups felt present and able to focus on the interaction, and generally enjoyed the experience. Neither of the conditions had subjects that were easily distracted (combined $\mu = 1.292$, $\sigma = .735$), and neither group felt detached from the interaction (combined $\mu = 1.5$, $\sigma = .5918$). Both groups had a favorable enjoyment level (combined $\mu = 5.310$, $\sigma = .656$) and had a strong sense of presence during the interaction (combined $\mu = 5.643$, $\sigma = .878$). Surprisingly, the average for question 1 ("I was able to easily look around the environment") is higher for the **Shared Interface**, where the human actually has less control over the robot's gaze than in the **Manual Controller**.

Table 7.1: General questions

No.	Question	M μ	M σ	S μ	S σ	p-value
1.	I was able to easily look around the environment.	4.286	1.604	4.667	1.528	.7695
2.	I felt aware of the real world surroundings while operating the robot.	5.857	0.690	4.667	1.528	.1411
3.	I was easily distracted.	1.429	0.787	1.667	1.155	.6836
4.	*I felt comfortable during the interaction.*	*4.257*	*1.215*	*5.333*	*1.155*	*.0126*
5.	The child felt comfortable during the interaction.	3.857	1.215	5.333	2.082	.08
6.	I felt detached from the interaction.	2.000	0.816	2.000	1.000	.3676
7.	When the interaction ended, I felt like I came back to the "real world" after a journey.	2.571	1.718	4.333	2.887	.2637
8.	What was your overall enjoyment level in the interaction?	5.286	0.488	5.333	2.082	.8247
9.	How strong was your sense of presence, "being there", when interacting with the child?	5.286	0.756	6.000	1.000	.1192
	Group Average	**4.907**	**1.018**	**5.148**	**0.818**	

Table 7.2: Questions comparing the two interfaces

No.	Question	μ	σ
1.	I found the manual interface easy to use.	5.333	1.155
2.	I found the shared interface easy to use.	6.000	1.000
3.	I preferred the shared interface over the manual interface.	4.333	2.309
4.	I think the child preferred the shared interface over the manual interface.	3.667	0.577

As can be seen, the only metric that was statistically significant was "I felt comfortable during the interaction". This preference was in favor of the **Shared Interface**. Figure 7-2 shows a graph of the means and standard deviations of these two conditions (left is Manual, right is Shared).

Questions were asked of the operators to determine which interface they preferred, and which interface they think the child preferred. The results are in Table 7.2.

We see that the shared interface was slightly easier to use than the manual controller, however when we compare the two preferences there is no significance with ANOVA (p =

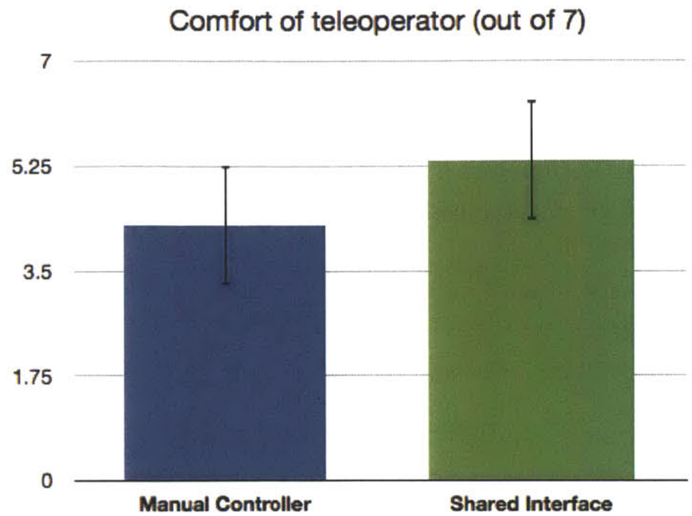


Figure 7-2: "I felt comfortable during the interaction"

.208). The shared interface was also preferred over the manual controller, although with a large standard deviation. This is primarily because the teleoperator who was an avid video gamer strongly preferred the manual gaze widget over the focus-based widget (see Chapter 5 for differences).

7.1.1 Cognitive Load

To measure cognitive load, the NASA-TLX questionnaire was used. Table 7.3 contains the means, standard deviations, and cross-conditional p-values from the general questionnaire. The questions are asked on a twenty point Likert scale.

While the p-values tell us that nothing is statistically significant, there is an information contained within this data. Mental demand monotonically decreased for all three participants as they used the interfaces, from 13 to 9, 11 to 2, and 15 to 11 (on a 20 point Likert scale). This decrease was independent of the interface used, indicating that there was an order effect present.

Table 7.3: Questions from the NASA-TLX task load index

No.	Question	M μ	M σ	S μ	S σ	p-value
1.	How mentally demanding was the task?	11.8	2.280	8.667	5.859	.1768
2.	How physically demanding was the task?	1.8	1.304	2.333	1.528	.5012
3.	How hurried or rushed was the pace of the task?	6.4	1.517	9.333	4.726	.1117
4.	How successful were you in accomplishing what you were asked to do?	7.4	5.320	7.000	5.292	.894
5.	How hard did you have to work to accomplish your level of performance?	10.8	2.864	9.667	3.055	.4999
6.	How insecure, discouraged, irritated, stressed, and annoyed were you?	5.4	5.320	7.000	5.196	.5951
	Group Average	**13.733**	**3.665**	**13.667**	**2.700**	

7.1.2 Social Presence

The Networked Minds questionnaire was used to measure social presence. Table 7.4 contains the means, standard deviations, and cross-conditional p-values from the social presence questions. The questions are asked on a seven point Likert scale.

No.	Question	M μ	M σ	S μ	S σ	p-value
	Isolation / Aloneness					
1.	I often felt as if I was all alone.	1.4	0.548	1.333	0.577	.8327
2.	I think the other individual often felt alone.	2.6	2.191	1.667	1.155	.3981
	Group Average	**6.000**	**0.849**	**6.500**	**0.236**	
	Mutual Awareness					
3.	I hardly noticed another individual.	1.4	0.548	1.000	0.000	.1411
4.	The other individual didnt notice me in the room.	1	0.000	1.667	1.155	.1053
	Group Average	**6.800**	**0.283**	**6.667**	**0.471**	
	Attentional Allocation					
5.	I sometimes pretended to pay attention to the other individual.	2.8	1.304	2.000	1.732	.3466
6.	The other individual sometimes pretended to pay attention to me.	2.6	1.342	3.333	2.082	.4342
7.	*The other individual paid close attention to me.*	*6*	*0.707*	*4.667*	*1.528*	*.0497*

8.	I paid close attention to the other individual.	5.2	1.643	6.000	1.000	.3466
9.	My partner was easily distracted when other things were going on around us.	2	1.414	3.667	2.517	.1397
10.	I was easily distracted when other things were going on around me.	1.4	0.548	2.333	2.309	.2602
11.	*The other individual tended to ignore me.*	*1.2*	*0.447*	*2.000*	*1.000*	*.05*
12.	I tended to ignore the other individual.	1.6	1.342	1.000	0.000	.3466
	Group Average	**5.950**	**0.630**	**5.542**	**0.907**	
	Empathy					
13.	When I was happy, the other was happy.	6.2	1.095	4.667	2.517	.1395
14.	When the other was happy, I was happy.	6	1.000	6.333	0.577	.5097
15.	The other individual was influenced by my moods	5.6	0.548	5.667	1.155	.8832
16.	I was influenced by the others moods	5.8	0.447	5.000	2.000	.2623
17.	The others mood did NOT affect my mood/emotional-state	1.6	0.894	1.667	0.577	.8832
18.	My mood did NOT affect the others mood/emotional-state	1.6	0.894	2.667	1.528	.1274
	Group Average	**6.067**	**0.327**	**5.556**	**0.689**	
	Mutual Understanding					
19.	My opinions were clear to the other	5.6	1.673	5.667	0.577	.9332
20.	The opinions of the other were clear	5.8	0.837	5.333	2.082	.5548
21.	My thoughts were clear to the other	5.4	1.342	5.667	0.577	.6819
22.	The other individuals thoughts were clear to me	5.6	0.894	5.000	2.000	.4458
23.	The other understood what I meant	5	1.581	5.667	1.155	.4265
24.	I understood what the other meant	6.2	0.837	6.000	1.000	.6938
	Group Average	**5.600**	**0.400**	**5.556**	**0.344**	
	Behavioral Interdependence					
25.	My actions were dependent on the others actions	5	1.871	6.000	1.000	.296
26.	The others actions were dependent on my actions	5.6	0.894	5.667	1.155	.9051

27.	My behavior was in direct response to the others behavior	5	1.871	5.667	1.155	.486
28.	The behavior of the other was in direct response to my behavior	5.8	0.837	5.667	1.528	.8327
29.	What the other did affected what I did	6.2	0.837	6.000	1.000	.6938
30.	What I did affected what the other did	5.8	1.643	5.667	1.15	.875
	Group Average	**5.567**	**0.480**	**5.778**	**0.172**	
	Total Average	**5.887**	**0.560**	**5.733**	**0.663**	

Interestingly, the two questions that yielded statistically significant results were: "The other individual paid close attention to me" (Figure 7-3) and "The other individual tended to ignore me" (Figure 7-4). Both of these favored the **Manual Controller**, although it's worth noting that the overall scores from *both* conditions are above average.

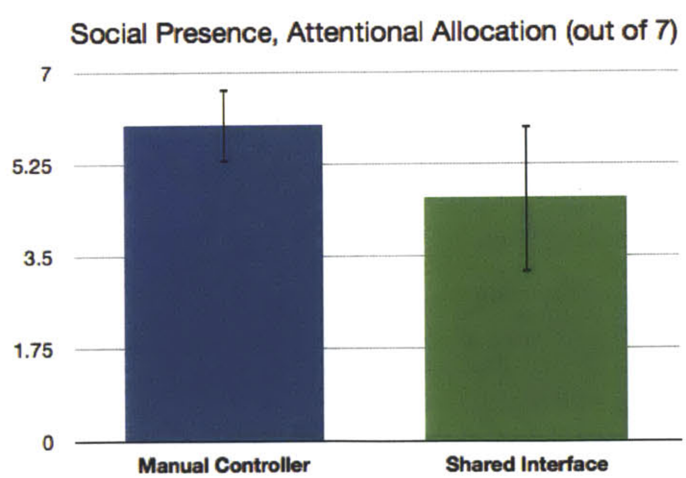


Figure 7-3: "The other individual paid close attention to me"

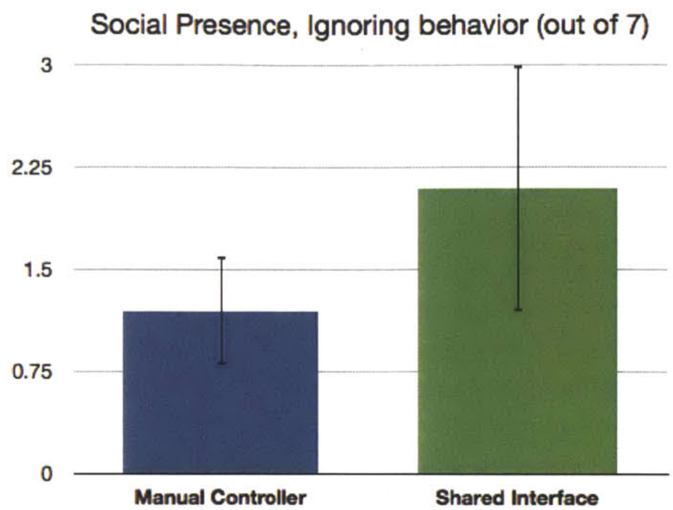


Figure 7-4: "The other individual tended to ignore me"

7.1.3 Engagement

The questionnaire to assess engagement was informed by Van Baren's article on Measuring Presence [73]. Table 7.5 contains the means, standard deviations, and cross-conditional p-values from the social presence questions. The questions are asked on a seven point Likert scale.

There was a single question that was statistically significant: "How relaxing or exciting was the experience" (Figure 7-5). The results indicate that the **Shared Interface** was significantly more relaxing than the **Manual Interface**. The

7.1.4 Ease of use of the platform

The final questionnaire attempts to assess the usability of the platform. Table 7.6 contains the means, standard deviations, and cross-conditional p-values from the social presence questions. The questions are asked on a seven point Likert scale.

Table 7.5: Questions regarding engagement

No.	Question	**M** μ	**M** σ	**S** μ	**S** σ	p-value
1.	How engaging was the interaction?	5.8	1.789	5.333	0.577	.5852
2.	*How relaxing or exciting was the experience?*	*5.8*	*0.837*	*4.000*	*1.732*	*.0265*
3.	How completely were your senses engaged?	5.8	0.447	5.667	1.155	.7570
4.	The experience caused real feelings and emotions for me.	6	0.707	5.333	0.577	.1053
5.	I was so involved that I lost track of time.	6	0.707	5.667	0.577	.3880
6.	I found the control mechanism distracting.	2.6	0.548	2.667	0.577	.8327
7.	How well could you concentrate on the interaction rather than on the mechanisms used to perform the interaction?	5	2.345	6.333	0.577	.2458
	Group Average	**5.686**	**0.363**	**5.381**	**0.705**	

Table 7.6: Questions about the platform's viability

No.	Question	**M** μ	**M** σ	**S** μ	**S** σ	p-value
1.	How enjoyable was your experience controlling DragonBot?	5.6	0.894	6.000	0.000	.3466
2.	I found that the robots motor noise distracted me from the interaction.	1.8	1.789	2.333	2.309	.6357
3.	I found that the robots movement distracted me from the interaction.	1.4	0.548	1.333	0.577	.8327
4.	I found the robot to be intuitive to control.	5.8	0.447	5.667	0.577	.6357
	Group Average	**6.050**	**0.443**	**6.000**	**0.471**	

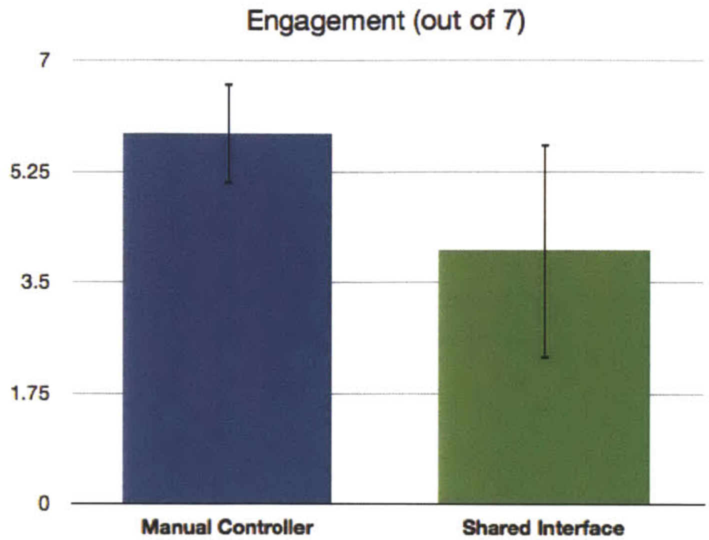


Figure 7-5: "How relaxing or exciting was the experience"

Although nothing here is significantly different between the two conditions, we see that people found the control mechanisms intuitive, enjoyed operating the robots, and weren't affected by typical hardware-based issues in teleoperation interfaces.

7.1.5 Arousal

To measure the operator's arousal level, we used their electrodermal activity and combined it with contextual information from videos of the interaction. The results for the three operators can be found in Figures 7-6, 7-7, and 7-8.

As can be seen, the signals look very different for the unique operators. There were about 10 samples that contained very noisy data from the EDA sensor, which were thrown out for analysis. Combined with a few user-errors where the sensor was accidentally powered off, there are 10 usable EDA samples that are included in this analysis. A rough pass at analyzing the averages and peaks in the data can be seen in Table 7.7. The **End / Beginning ratio** is the average of the second half of the interaction divided by the average from the first half.

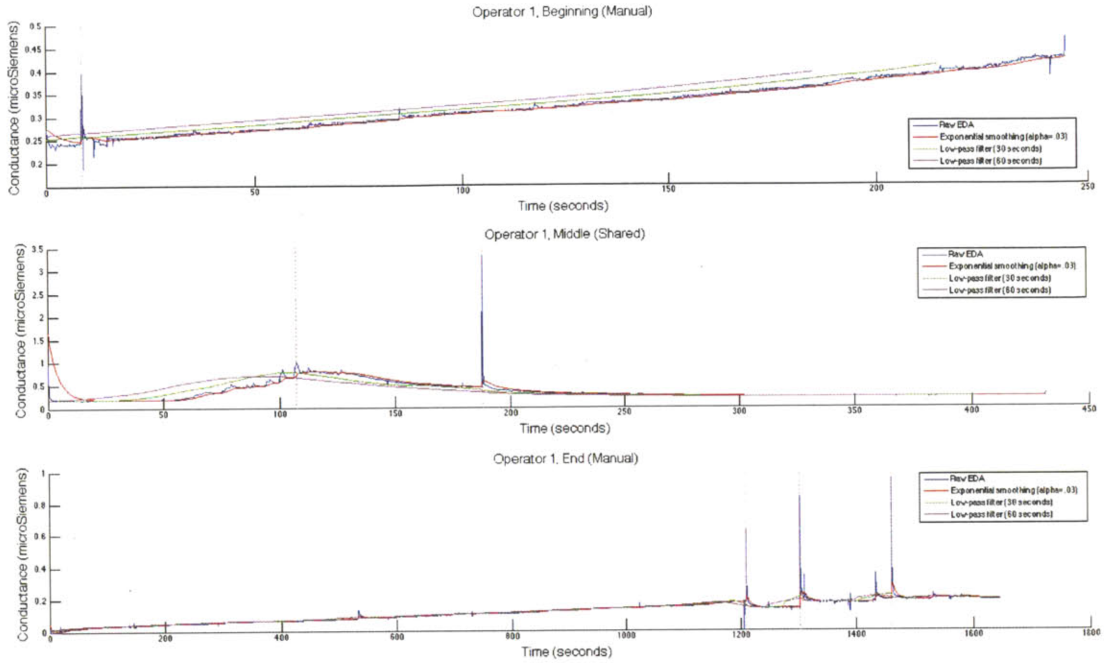


Figure 7-6: EDA from operator #1

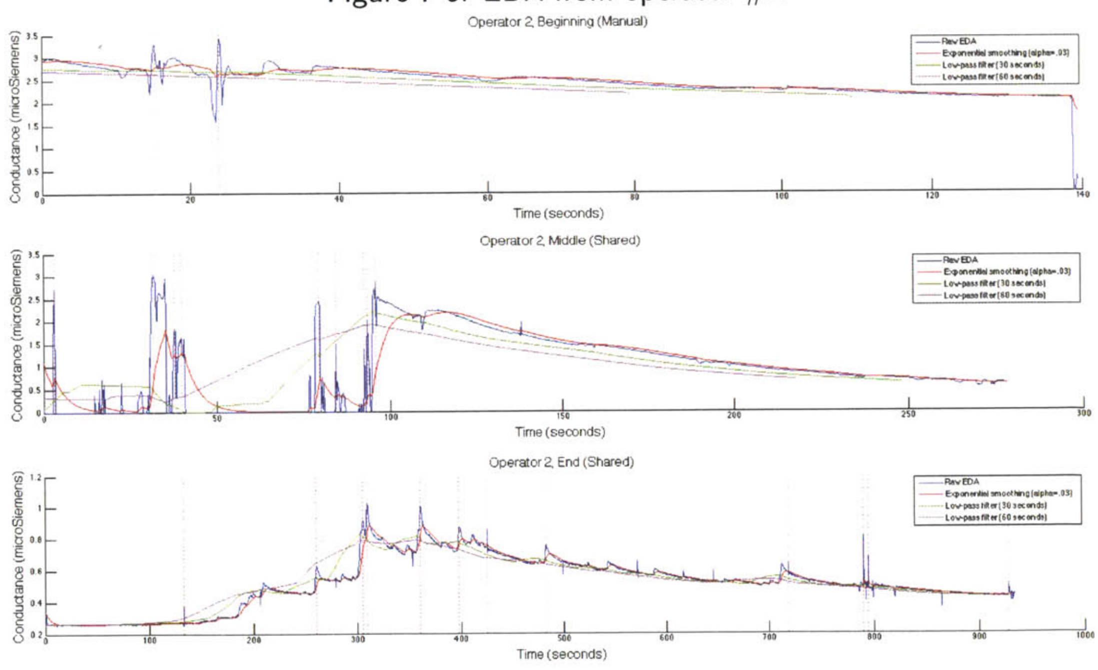


Figure 7-7: EDA from operator #2

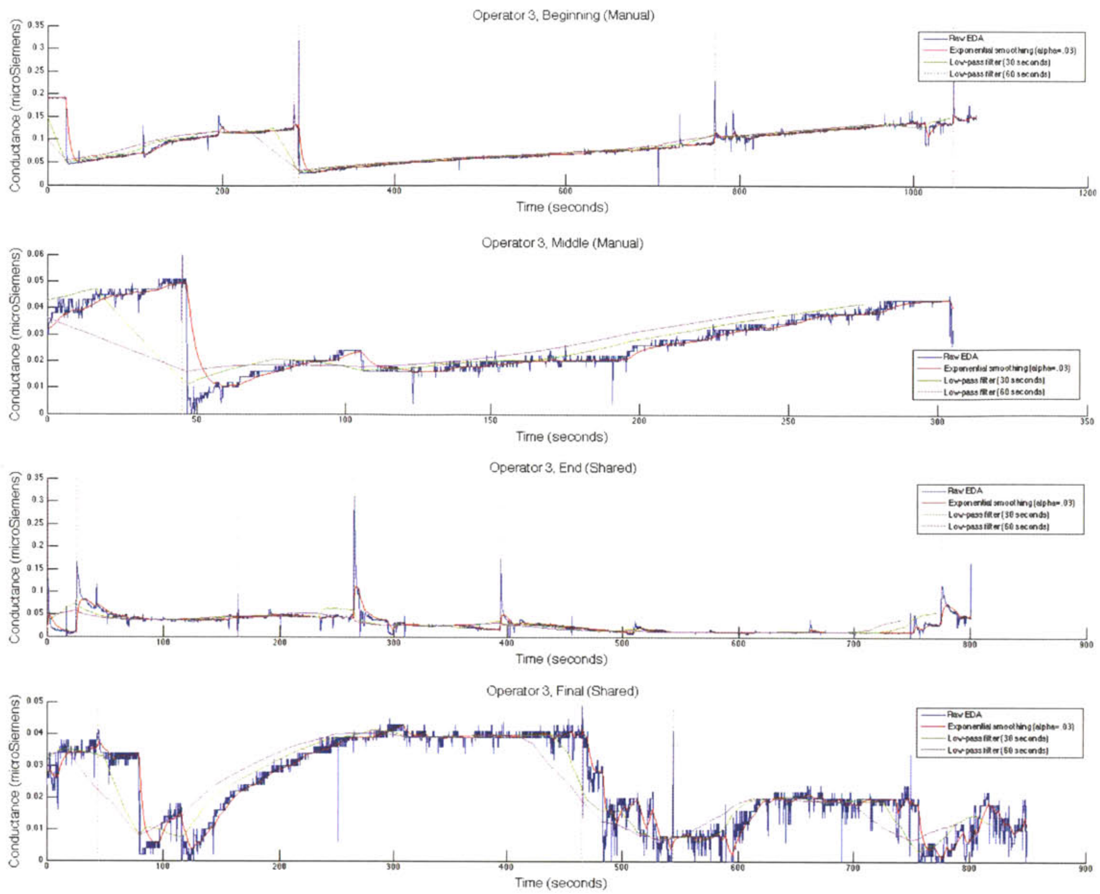


Figure 7-8: EDA from operator #3

We now group the analysis temporally and by condition and provide a few results comparing the EDA mean, ratio, and peaks between these groupings. These results can be found in Table 7.8. Because there were so few samples, nothing here is statistically significant and all p-values are omitted. However, we see that the mean EDA response steadily decreases as the operators gain more experience, and the Shared interface has a lower EDA response than the Manual controller. All of the cross-participant means were normalized based on the participants' baseline and maximum EDA responses.

Table 7.7: Results from analysis of electrodermal activity (EDA)

Subject	Condition	Average EDA	End / Beginning ratio	# Peaks
Operator 1, Beginning	Manual	0.324	1.301	1
Operator 1, Middle	*Shared*	0.350	0.537	3
Operator 1, End	Manual	0.114	2.452	3
Operator 2, Beginning	Manual	2.477	0.820	2
Operator 2, Middle	*Shared*	0.922	1.164	10
Operator 2, End	*Shared*	0.513	1.061	12
Operator 3, Beginning	Manual	0.089	1.379	5
Operator 3, Middle	Manual	0.028	1.226	1
Operator 3, End	*Shared*	0.029	0.467	6
Operator 3, Final	*Shared*	0.024	0.542	3

Table 7.8: Grouped analysis of electrodermal activity (EDA)

	Mean, average EDA response	Std. Dev.	Mean, End/Beginning Ratio	Std. Dev.	Mean, # Peaks	Std. Dev.
Beginning	0.963	1.316	1.166	0.303	2.667	2.082
Middle	0.433	0.453	0.975	0.381	4.667	4.726
End	0.17	0.233	1.13	0.92	6	4.243
Manual	0.606	1.052	1.435	0.608	2.4	1.673
Shared	0.367	0.375	0.754	0.33	6.8	4.087

7.1.6 Advice from the Operators

A lot of useful advice was obtained from the robot operators through results from open-ended comments on the questionnaire and transcriptions from video interviews. The general consensus was that audio quality and minimizing lag were the most important factors in maintaining a fluid social interaction. Two of the participants largely preferred the shared interface, the first saying it was "much easier to use than the manual EXCEPT when face and object recognition don't work". The second user said that "...if an item was out of the view of the camera, it couldn't be found and tracked. Other than that, the interface was really comfortable". The third user, who preferred the manual controller, said "I liked the face tracking of the child, but wish I could've looked around a bit more freely.".

Figure 7-9: Pepper and a child sharing attention

The larger buttons of the shared interface were praised by all three operators, and the variability in the robot's motions was noticed and preferred ("Having several different laughs/giggles was nice for interaction with kids" and "the different motions from the same action seemed to make the kids more interested"). One teleoperator thought that the platform would greatly benefit from a high-resolution camera, as "some children prefer showing over telling". One operator thought that there were too many emotions to keep track of, and only changed the robot's emotional state an average of .7 times per interaction. All three of the operators mentioned that they thought a "manual override" when in the shared interface would be the best solution. For the majority of the time, the simpler interface was preferred, but the biggest issues with its use were concerning automatic object and face recognition.

7.2 Childs perspective

The average duration of the interaction was longer in the shared condition by an average of 1 minute and 18 seconds. Although this was not a statistically significant difference ($p = .172$), it is certainly trending. A graph comparing the durations can be seen in Figure 7-10.

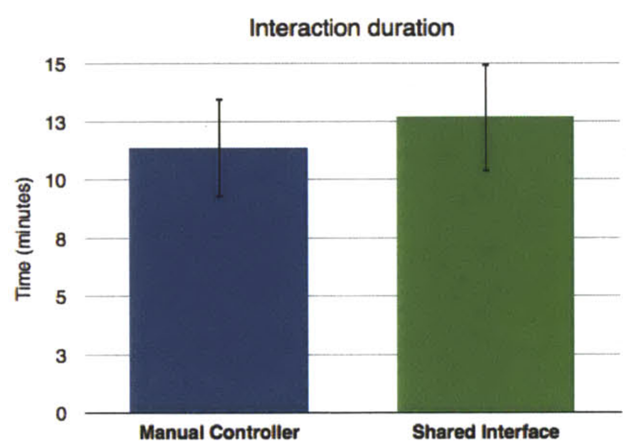


Figure 7-10: Durations of interactions for both conditions

When we look at the results from the informal post-survey, we see that nearly everybody had fun, thought the robot had fun, and wanted to speak to the robot again in the

Table 7.9: Results from informal post-survey for participants

Question	% "Yes", Manual	% "Yes", Shared	p-value
Did you have fun?	100%	100%	1.0
Do you think the robot had fun?	86%	100%	.4928
Do you think Pepper has feelings?	93%	90%	1.0
Would you want to talk to Pepper again in the future?	93%	90%	1.0

future. There was no significant difference in the two conditions.

7.2.1 Recall

Figure 7-11: Tangible interaction with Pepper

We hypothesized that the children would recall more stickers about the robot's preferences from interactions with the **Shared Interface** than with the **Manual Controller**. However, we actually see the opposite, which can be seen through Figure 7-12. A student's paired t-test was used to determine whether this difference was significant, and the p-value was $p = .225$ (with $t = 1.25$). Although the results are strange, it's possible that the children were more interested in the dynamic behavior of the robot and payed less attention to the stickers during Shared interface interactions. A detailed analysis of the child's focus of attention could shed more light on this perplexing result, but is beyond the scope of this thesis.

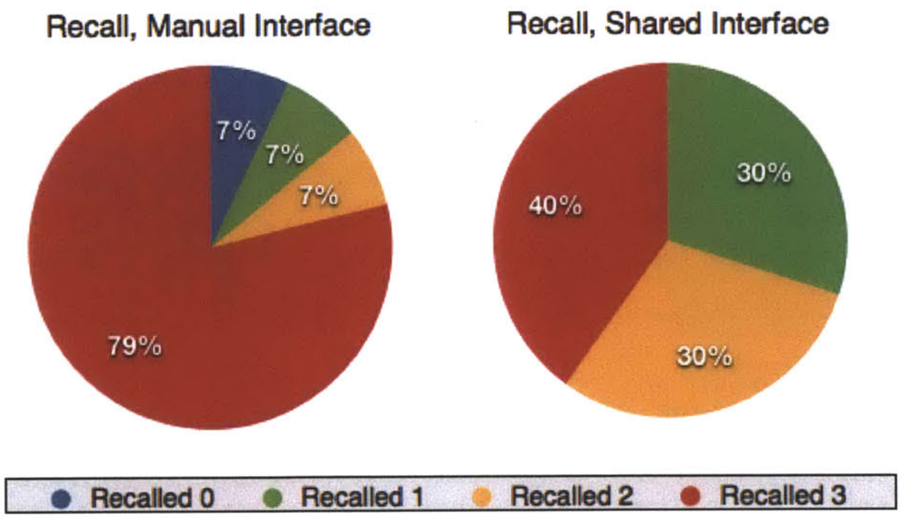


Figure 7-12: Recall percentages for both conditions

Table 7.10: Sharing 2 vs. 3 stickers

Condition	Gave robot 2 stickers	Gave robot 3 stickers
Manual	.286	.714
Shared	.6	.4

7.2.2 Sharing

We initially thought that children would share more stickers with the robot when it was controlled by the dynamic **Shared Interface** instead of the **Manual Controller**. Again, we see the opposite, with more children sharing stickers with the manually-controlled interface. These results can be found in Table 7.10.

While there is a trend to give the manually-controlled robot more stickers, this is not statistically significant ($p = .211$ with Fisher's exact test). We then decided to look only at the sharing behavior of the sticker which both the child and robot preferred. We see similar results here, with more children sharing the overlapping sticker with the manually-controlled character (seen in Table 7.11). Again, there is not a statistically significant difference here ($p = .204$ with Fisher's exact test).

Table 7.11: Sharing of overlapping sticker

Condition	Gave robot overlapping sticker	Kept overlapping sticker
Manual	.786	.214
Shared	.5	.5

These odd results invalidate both of our hypotheses regarding the child's behavior. When looking at the videos, it was obvious that the children were more engaged with the shared interface, so we decided to keep looking for the reason behind this confounding result.

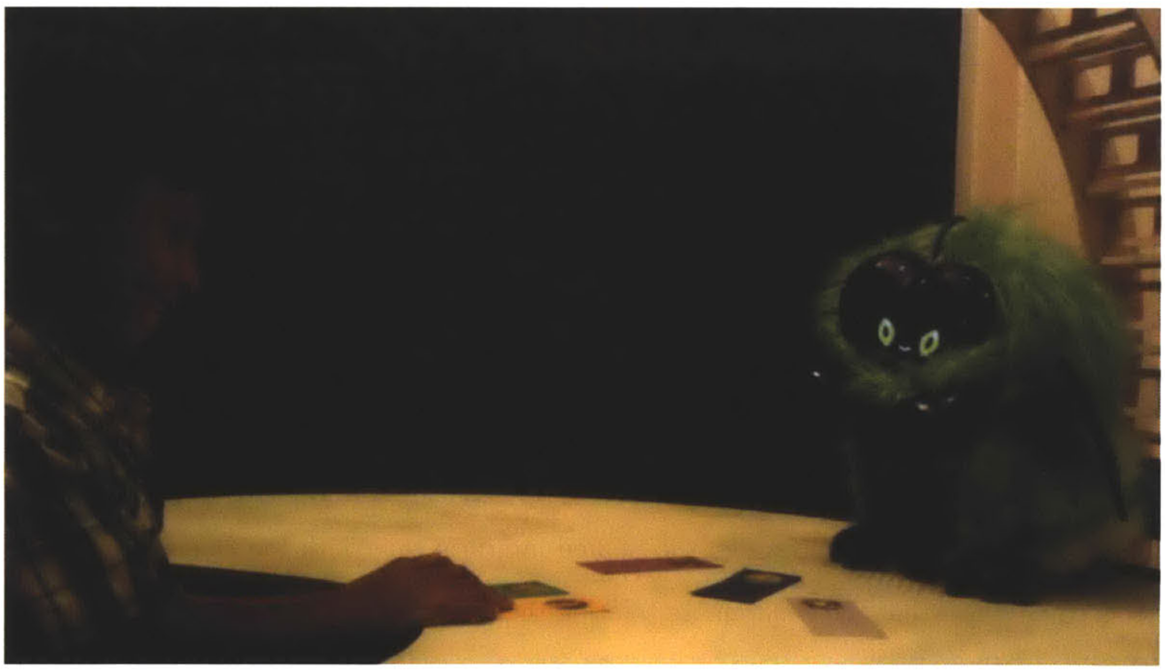


Figure 7-13: A child smiling after sharing 3 stickers with Pepper

7.2.3 Re-engagement

After spending much time thinking about why more children shared stickers with the manually-controlled robot, a very interesting result became apparent regarding re-engagement of the robot when the child is splitting up the stickers. When we compare the number of children who re-engaged the robot in the manual condition (2 out of 14) with the number of re-engagements with the shared condition (6 out of 10), we find that this is a statistically significant result. This was determined using Fisher's exact test, resulting in a p-value of $p = 0.0324$. This explains why more children shared with the manual condition, because the operator told the child to take the shared sticker when re-engaged (see Section 6.3.4).

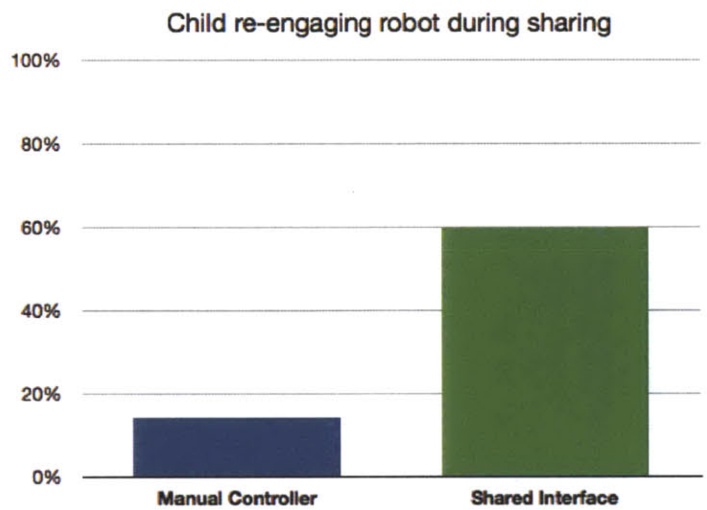


Figure 7-14: Child re-engaging robot during sharing

This tells us that children are considering the robot's preferences more when it's behavior is more dynamic through the Shared Interface than with the Manual controller. This could mean that the children are treating the robots more like peers instead of social outsiders. Pictures of two children re-engaging the robot during the sharing portion of the task can be seen in Figure 7-15.

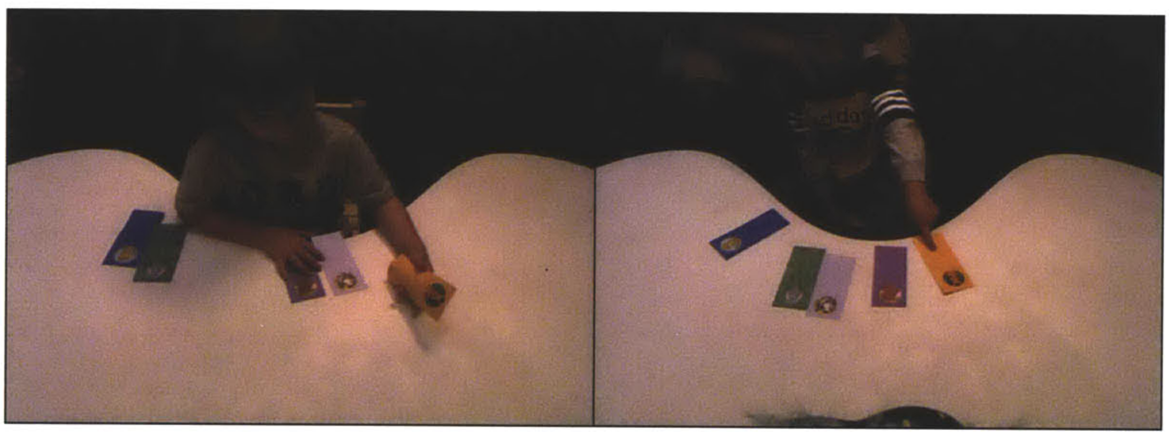


Figure 7-15: Two participants re-engaging the robot about the overlapping sticker

Chapter 8

Conclusion

8.1 Contributions

This thesis has taken initial strides towards enabling the use of social robots in the field. To the best of our knowledge, this work represents one of the most capable platforms for studying social interactions "in the wild" between robots and children. We have accomplished this by harnessing mobile phones for driving down cost and providing reliable data-capture capabilities.

This thesis presented the following technical contributions:

1. The design and implementation of an expressive social robot with strong appeal to children that can exist outside of the laboratory and support HRI research in the field.

2. A teleoperator interface for non-expert users that enables a person to dynamically shape the robots interaction during a conversation.

3. A data collection framework that can capture a range of relevant data to support HRI studies in the field, including video, audio, 3D depth maps, electrodermal activity, and robot commands, actions and states.

Furthermore, we analyzed the results from an initial user study with 4-7 year old kids. The most interesting finding is that when the robot was controlled using the novel shared autonomy interface, children re-engaged the robot significantly more during the reward-allocation task. This indicates that children are treating the robot more like a peer when it's behaviors are more dynamic. While the operators didn't significantly prefer one of the interfaces over the other, they steadily improved their performance and became less aroused as they became familiar with the control schemes.

8.2 Future Work

While we have found some interesting results, there is much future work that could make the experience of controlling DragonBot better, both from a user-standpoint and an operator perspective. Following the advice from the teleoperators, we feel that the most significant improvement that could be made to the **Shared Interface** is adding an override for manually controlling the gaze of the robot. This aspect, which is present in the **Manual Controller**, would be useful to add a more finely-tuned control mechanism when it's appropriate.

Figure 8-1: Various images from the robot's vision stream

We have just skimmed the surface of the data that was collected from our user evaluation. A more detailed analysis could yield interesting information about shared attention between the robot and the child. Also, the electrodermal activity of the operator could be carefully analyzed using all of the contextual data streams to determine whether any individual aspect of controlling the robot was particularly arousing for the operator.

Finally, although it was beyond the scope of this thesis, this robot was designed for long-term interactions in the field. The time constraints of a master's thesis combined with the desire for a large sample size led us to run our initial evaluation in the laboratory, and without any long-term interaction. However, now that we know that the robot is an effective platform for social interaction with kids, we are looking forward to sending the robots "out into the world" where they can interact with children for extended periods of time.

8.3 Concluding Remarks

The smartphone revolution has put a computer in everybody's pocket, and the DragonBot platform represents a new way to harness these phones for long-term and longitudinal studies. By developing a socially-expressive low cost robot with advanced data-capture capabilities, we offer a glimpse of the possibilities from having human-operated robots in the home or classroom. Through creating this platform, we hope to start gathering the amounts of data needed to build and train autonomous models for understanding social dynamics.

Appendices

Appendix A

Custom Electronics

This chapter presents the details and designs for two custom pieces of electronics that made DragonBot possible.

A.1 SEEDpower power management

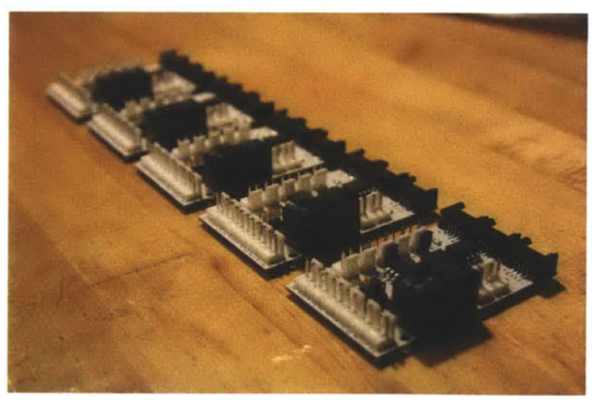


Figure A-1: A group of 5 SEEDpower boards ready to be put into robots

SEEDpower is an integrated solution for power management and regulation on small-to-medium sized robots. With full isolation of logic and motor power sources, the board supports 3-channel input (up to 3 batteries) and 4-channel output (Motor voltage, +12V, +5V, and +3.3V). Any of the two input batteries may be placed in series or parallel (using on-board jumpers), and the output is fully protected with both fuses and flyback diodes. The board supports "plug-and-play" charging, using an onboard relay to switch to an external supply whenever the robot is plugged in.

Name	Minimum	Maximum
Input Voltage	4.8V	24V
Motor Current (Amps)	N/A	7A
+12V Current (Amps)	N/A	5A
+5V Current (Amps)	N/A	5A
+3.3V Current (Amps)	N/A	5A
Operating temperature	-40°C	125°C

The schematic for SEEDpower is in Figure A-2 and the 2-layer board layout can be found in Figure A-3.

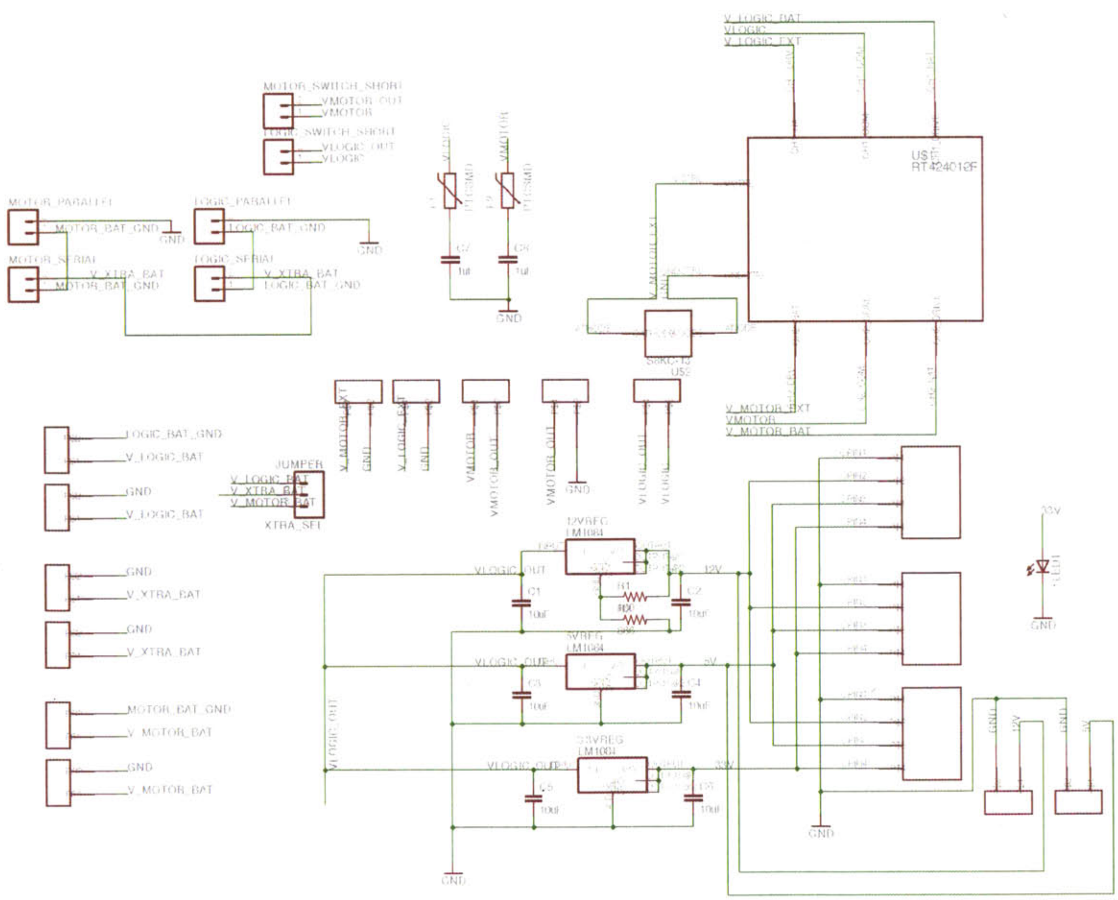


Figure A-2: Schematics for the SEEDpower management board

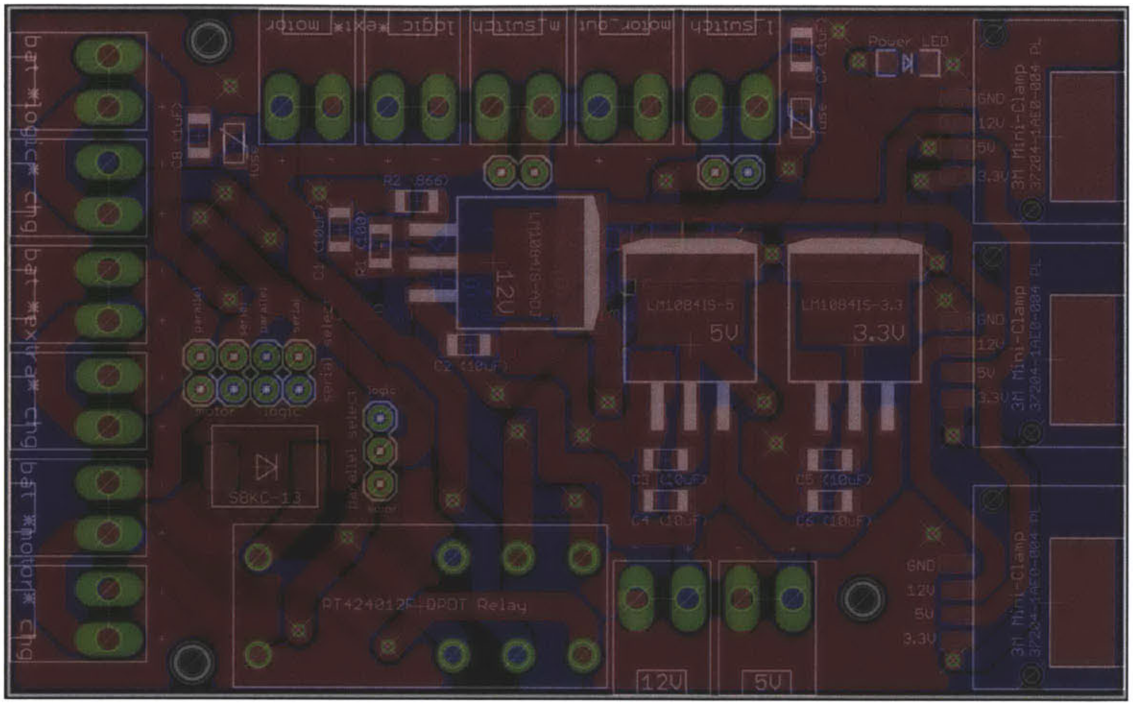


Figure A-3: 2-layer board design for the SEEDpower management board

A.2 SenSkin capacitive sensing

The SenSkin schematics can be found in figure A-4 and the two-layer board layout is in A-5.

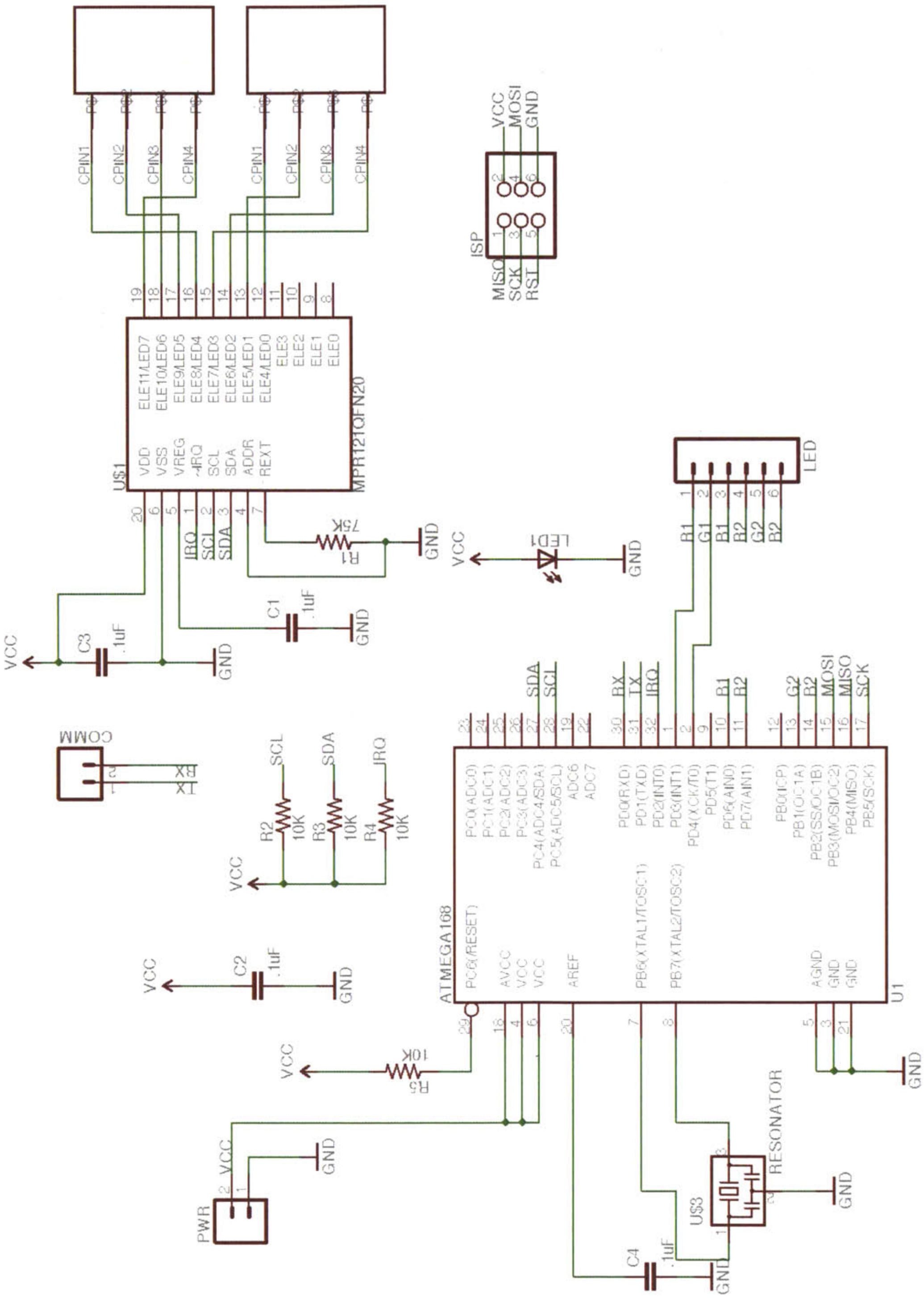


Figure A-4: Schematics for the SenSkin capacitive sensing board

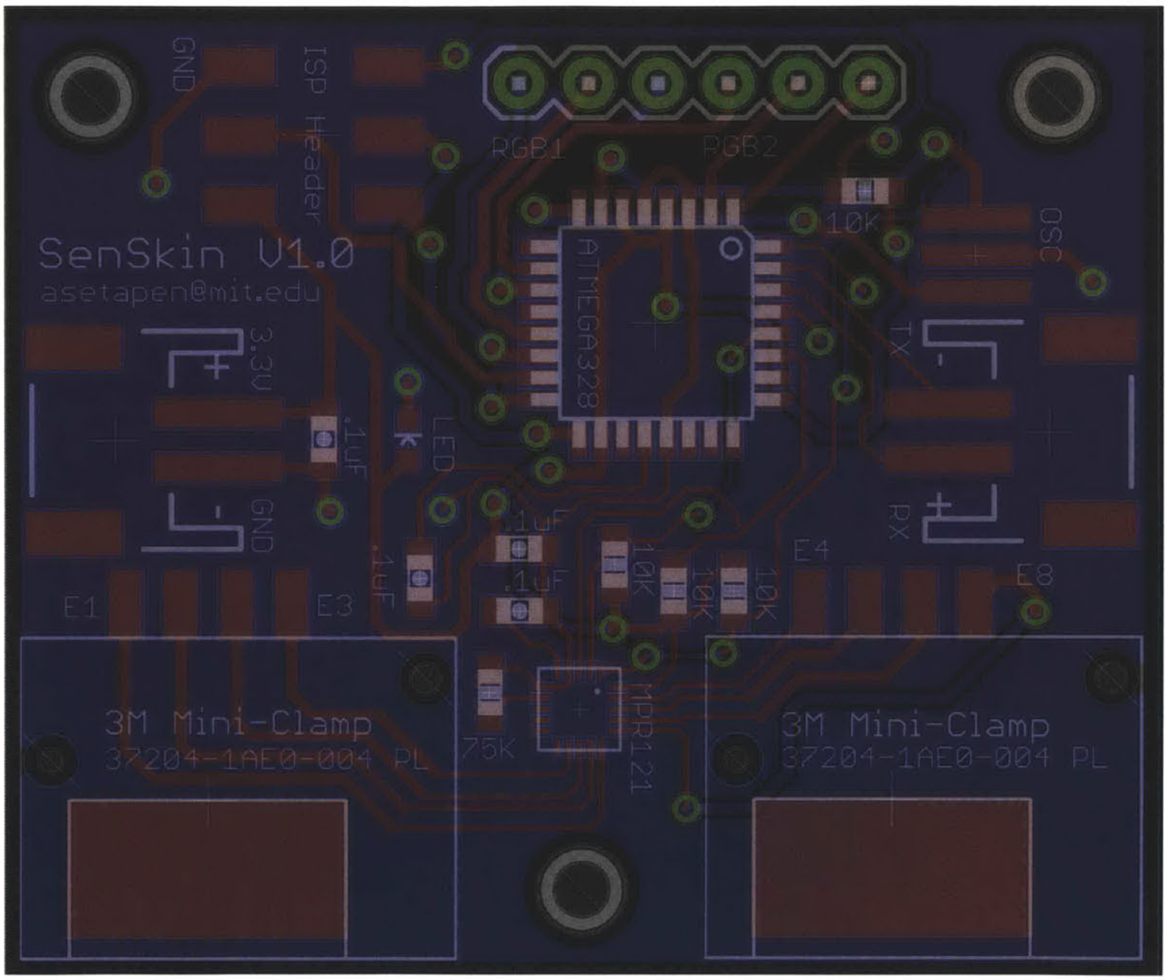
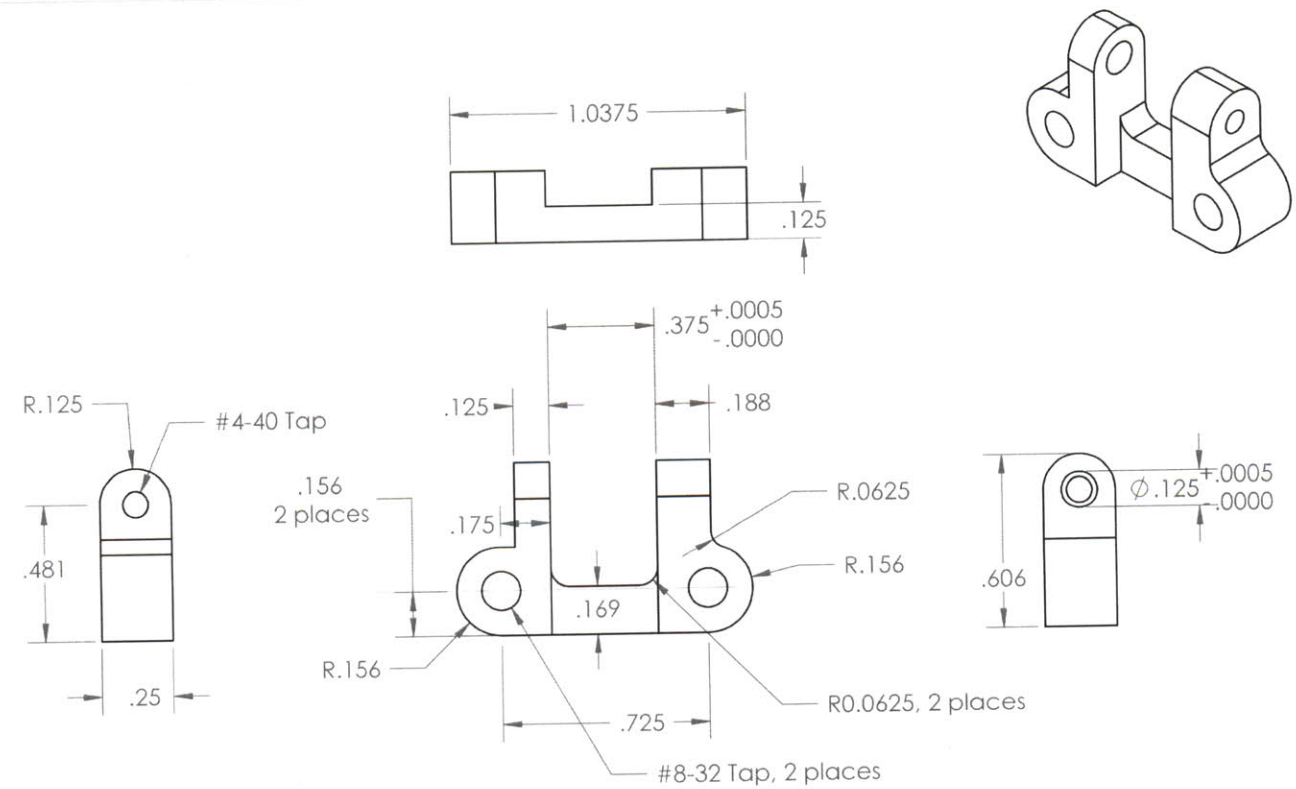


Figure A-5: 2-layer board design for the SenSkin capacitive sensing board

Appendix B

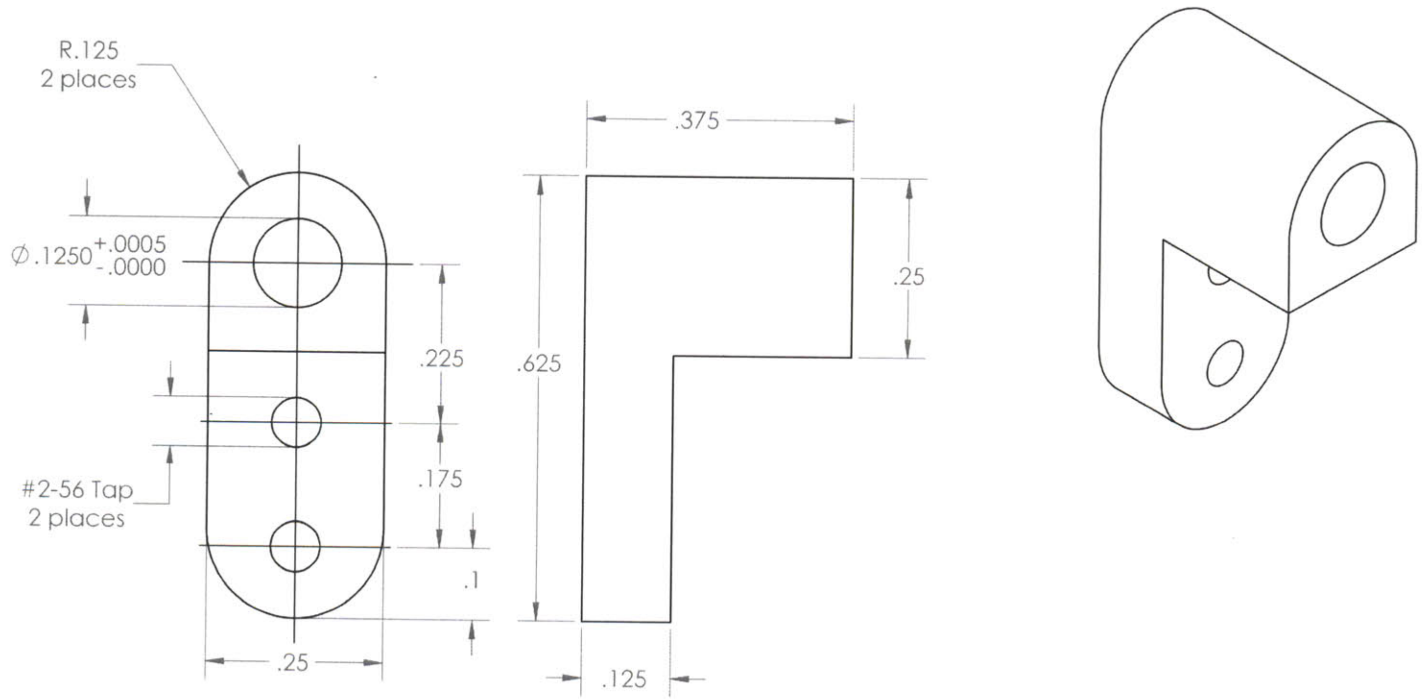
Linkage Design

This appendix contains the mechanical diagrams needed to recreate the linkages of a four-chained $\underline{\mathbf{R}} - (\mathrm{SS})^2$ mechanism.



UNLESS OTHERWISE SPECIFIED:		NAME	DATE	
DIMENSIONS ARE IN INCHES TOLERANCES: THREE PLACE DECIMAL ±.001	DRAWN	adam	11/11/11	TITLE: HingeBracket
	CHECKED			
	ENG APPR.			
	MFG APPR.			
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.			
MATERIAL Steel	COMMENTS: Break all edges First article inspection			SIZE A · DWG. NO. HingeBracket · REV 5
FINISH				
DO NOT SCALE DRAWING				SCALE: 2:1 · WEIGHT: · SHEET 1 OF 1

5 4 3 2 1



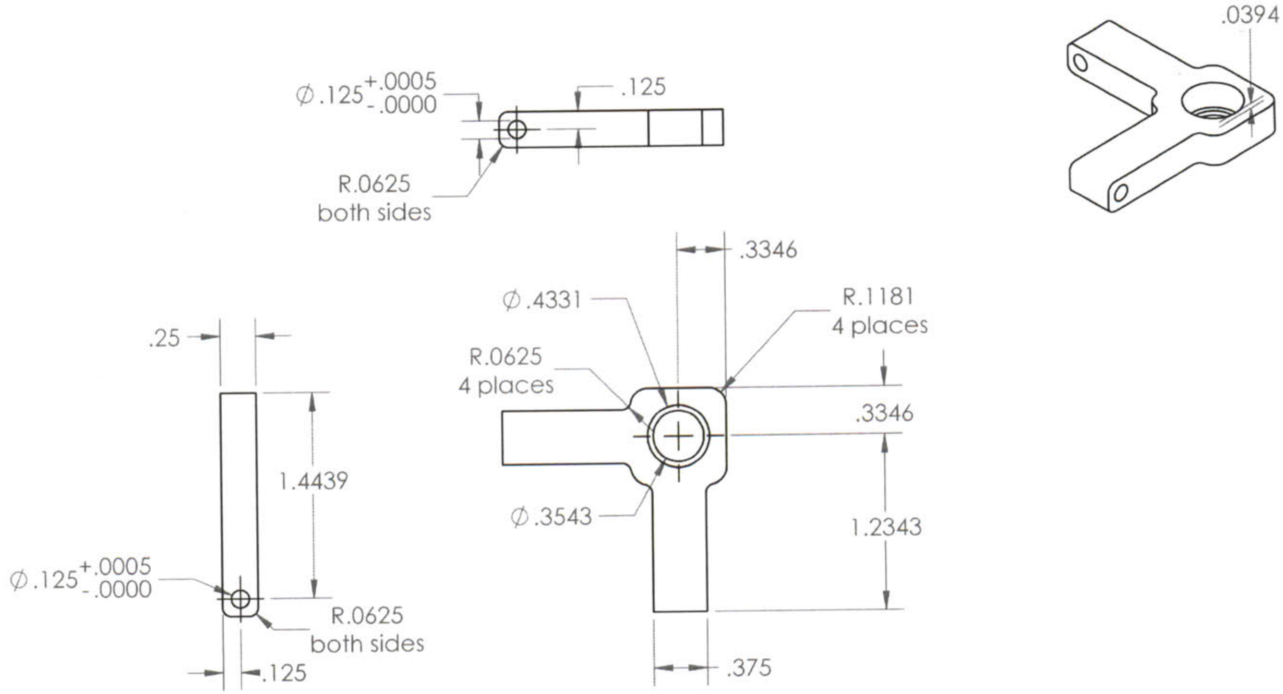
UNLESS OTHERWISE SPECIFIED:		NAME	DATE
DIMENSIONS ARE IN INCHES TOLERANCES: TWO PLACE DECIMAL ±.01 THREE PLACE DECIMAL ±.001	DRAWN	adam	11/28/11
	CHECKED		
	ENG APPR.		
	MFG APPR.		
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.		
MATERIAL Steel	COMMENTS: Break all edges First article inspection		
FINISH			
DO NOT SCALE DRAWING			

TITLE:
HornConnect

SIZE	DWG. NO.	REV
A	HornConnect	5

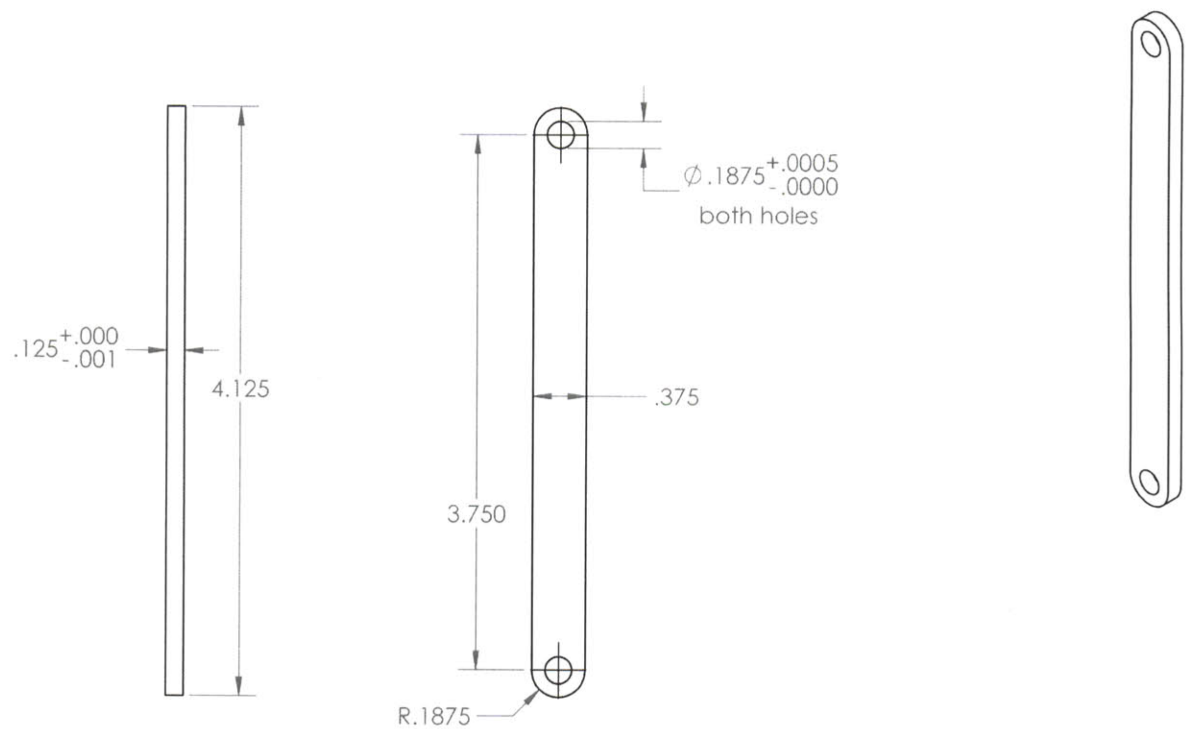
SCALE: 5:1 WEIGHT: SHEET 1 OF 1

5 4 3 2 1



UNLESS OTHERWISE SPECIFIED:		NAME	DATE	
DIMENSIONS ARE IN INCHES TOLERANCES: TWO PLACE DECIMAL ±.01 THREE PLACE DECIMAL ±.001	DRAWN	adam	11/30/11	
	CHECKED			TITLE: Rotator V5
	ENG APPR.			
	MFG APPR.			
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.			
MATERIAL Steel	COMMENTS: Symmetric part Lip on center hole is .0394" thick Break all edges First article inspection			SIZE A · DWG. NO. Rotator · REV 5
FINISH				
DO NOT SCALE DRAWING				SCALE: 1:1 · WEIGHT: · SHEET 1 OF 1

5 4 3 2 1



UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
TWO PLACE DECIMAL ±005
THREE PLACE DECIMAL ±001

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL
6061 Aluminum

FINISH

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	adam	12/28/11
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

COMMENTS:
Symmetric Part
Break all edges
First article inspection

TITLE:
Strut

SIZE DWG. NO. REV
A Strut 5

SCALE: 1:1 WEIGHT: SHEET 1 OF 1

5 4 3 2 1

Appendix C

Study Materials

C.1 Interaction script

The following script was used by the robot operators and the experimenter:

– BEGIN –

Pepper:

ACTION: Anticipate

"Ooh hi there! Are you {NAME}?"

`[ Wait for response ]`

Pepper:

EMOTION: Excited

"Its great to meet you, {NAME}! Im so excited that youre here to visit me! My name is Pepper."

INTRO: Sneeze

Experimenter:

"Here are a set of 5 stickers for you and Pepper to share with each other. Why don't you take a few minutes to talk about the stickers with each other? Maybe talk about which stickers you like or dont like to help you decide how you might want to divide them up."

——————— BEGIN OPEN-ENDED INTERACTION ———————

`[ Child and robot talk about the stickers for about 5 minutes. The robot should attempt to keep shared focus and positively react to at least one of the stickers that the child seems to like. ]`

If the child asks you questions, please answer them. If the child is shy, or you feel like asking a question, you should initiate a question or comment. Here are some possible questions:

- Which animal would you bring to school for show and tell?
- Have you ever been to the zoo? Somebody told me that they have ALL these animals there!
- Which animal do you think is the fastest?
- Which animal is the smartest?
- Which animal would be the best basketball player?
- Which animal would make the best pet?
- Which animal would you like to be for a day?
- Which animal do you think is the scariest?
- What sound does that animal make?
- Have you ever seen that animal in real life?

- Which animal looks most like me?
- Which of these animals are boys and which are girls?

And here are some possible standard responses:

- I think so too!
- Cool! I didn't know that!
- Really? I think that the `[ talk about another animal ]`...
- Are you sure?

——————— CONCLUDE OPEN-ENDED INTERACTION ———————

Experimenter:

"Now that you've had a chance to talk about the different stickers,

{NAME}, why dont you say which 3 stickers you like best?"

`[ Child names 3 stickers ]`

"Thanks, {NAME}. Now Pepper, why dont you tell us which 3 stickers you like best?"

Pepper:

Pick favorites and overlap with exactly 1 of the stickers the child likes best

Experimenter:

"OK. I think youve both found some stickers you really like.

{NAME}, why don't you go ahead and divide up the stickers for you and Pepper to keep."

Pepper:

ACTION: Anticipate

`[ Child divides up stickers ]`

`[ Respond appropriately to sticker if you get one of your favorites ]`

Experimenter:

IF EACH PARTICIPANT HAS STICKERS:

"Wonderful. You both have some really cool stickers."

IF CHILD TOOK ALL STICKERS:

PEPPER → EMOTION: Sad

"Well, I dont think Pepper has much of a use for stickers anyway."

IF CHILD GIVES ROBOT ALL STICKERS:

PEPPER → EMOTION: Excited

"How generous! Look how excited you made Pepper!"

Experimenter:

"OK, Pepper looks tired, so its time to say goodbye."

Pepper:

OUTRO: Yawn

"It was great to meet you, NAME! I hope you come back to visit me soon! Bye!"

Experimenter:

`[ Make sure child takes the stickers allocated to him/herself ]`

`[ Escort the child out of the interaction area ]`

`[ Record self-survey from participant in staging area ]`

– END –

C.2 Questionnaires

The following questionnaires were given to the robot operators. The full questionnaire includes a NASA-TLX task load index assessment [69], questions from the Networked Minds Social Presence survey [72], and our own questions related to the specific robot, interface, and task.

Interaction Time: ________________ Teleoperator ID: __________
Participant ID: __________

DragonBot Teleoperator
Short Questionnaire

Please circle the number that represents your level of agreement, as follows:

1 = Strongly Disagree
2 = Disagree
3 = Somewhat Disagree
4 = Neutral
5 = Somewhat Agree
6 = Agree
7 = Strongly Agree

	Strongly Disagree						*Strongly Agree*
I was able to easily look around the environment.	1	2	3	4	5	6	7
I felt aware of the real world surroundings while operating the robot.	1	2	3	4	5	6	7
I was easily distracted.	1	2	3	4	5	6	7
I felt comfortable during the interaction.	1	2	3	4	5	6	7
The child felt comfortable during the interaction.	1	2	3	4	5	6	7
I felt detached from the interaction.	1	2	3	4	5	6	7
When the interaction ended, I felt like I came back to the "real world" after a journey.	1	2	3	4	5	6	7

	Not enjoyable at all						*Very enjoyable*
What was your overall enjoyment level in the interaction?	1	2	3	4	5	6	7

	Not strong at all						*Very strong*
How strong was your sense of presence, “being there”, when interacting with the child?	1	2	3	4	5	6	7

Teleoperator ID: __________

DragonBot Teleoperator
Full Questionnaire

Part A
(Only fill out once)

Name: ____________________________________

How old are you ? ___________________________

What is your level of education?

____ Some high school
____ High school degree
____ Some college
____ College degree
____ Some graduate school
____ Graduate school degree

	Nothing						*A lot*
How much do you know about robotics, compared to the average US citizen?	1	2	3	4	5	6	7
How much do you know about computers, compared to the average US citizen?	1	2	3	4	5	6	7

	Not comfortable at all						*Very comfortable*
How comfortable do you typically feel in social situations when you meet somebody new?	1	2	3	4	5	6	7

How many video games do you play in a week?

____ Zero
____ Between 10 minutes and 2 hours a week
____ Less than 2 hours a week
____ More than 2 hours a week

Part B

Please circle the number that represents your level of agreement, as follows:
1 = Strongly Disagree
2 = Disagree
3 = Somewhat Disagree
4 = Neutral
5 = Somewhat Agree
6 = Agree
7 = Strongly Agree

	Strongly Disagree						*Strongly Agree*
I was able to easily look around the environment.	1	2	3	4	5	6	7
I felt aware of the real world surroundings while operating the robot.	1	2	3	4	5	6	7
I was easily distracted.	1	2	3	4	5	6	7
I felt comfortable during the interaction.	1	2	3	4	5	6	7
The child felt comfortable during the interaction.	1	2	3	4	5	6	7
I felt detached from the interaction.	1	2	3	4	5	6	7
When the interaction ended, I felt like I came back to the "real world" after a journey.	1	2	3	4	5	6	7

	Not enjoyable at all						*Very enjoyable*
What was your overall enjoyment level in the interaction?	1	2	3	4	5	6	7

	Not strong at all						*Very strong*
How strong was your sense of presence, “being there”, when interacting with the child?	1	2	3	4	5	6	7

Part C

Please circle the number that best represents your answer to each question.

	Strongly Disagree						*Strongly Agree*
I often felt as if I was all alone.	1	2	3	4	5	6	7
I think the other individual often felt alone.	1	2	3	4	5	6	7
I hardly noticed another individual.	1	2	3	4	5	6	7
The other individual didn't notice me in the room.	1	2	3	4	5	6	7
I sometimes pretended to pay attention to the other individual.	1	2	3	4	5	6	7
The other individual sometimes pretended to pay attention to me.	1	2	3	4	5	6	7
The other individual paid close attention to me.	1	2	3	4	5	6	7
I paid close attention to the other individual.	1	2	3	4	5	6	7
My partner was easily distracted when other things were going on around us.	1	2	3	4	5	6	7
I was easily distracted when other things were going on around me.	1	2	3	4	5	6	7
The other individual tended to ignore me.	1	2	3	4	5	6	7
I tended to ignore the other individual.	1	2	3	4	5	6	7

	Strongly Disagree						Strongly Agree
When I was happy, the other was happy.	1	2	3	4	5	6	7
When the other was happy, I was happy.	1	2	3	4	5	6	7
The other individual was influenced by my moods	1	2	3	4	5	6	7
I was influenced by the other's moods	1	2	3	4	5	6	7
The other's mood did NOT affect my mood/ emotional-state	1	2	3	4	5	6	7
My mood did NOT affect the other's mood/ emotional-state	1	2	3	4	5	6	7
My opinions were clear to the other	1	2	3	4	5	6	7
The opinions of the other were clear	1	2	3	4	5	6	7
My thoughts were clear to the other	1	2	3	4	5	6	7
The other individual's thoughts were clear to me	1	2	3	4	5	6	7
The other understood what I meant	1	2	3	4	5	6	7
I understood what the other meant	1	2	3	4	5	6	7

	Strongly Disagree						Strongly Agree
My actions were dependent on the other's actions	1	2	3	4	5	6	7
The other's actions were dependent on my actions	1	2	3	4	5	6	7
My behavior was in direct response to the other's behavior	1	2	3	4	5	6	7
The behavior of the other was in direct response to my behavior	1	2	3	4	5	6	7
What the other did affected what I did	1	2	3	4	5	6	7
What I did affected what the other did	1	2	3	4	5	6	7

Part D

Please circle the number that best represents your answer to each question.

	Not at all engaging						*Extremely engaging*
How engaging was the interaction?	1	2	3	4	5	6	7

	Very relaxing						*Very exciting*
How relaxing or exciting was the experience?	1	2	3	4	5	6	7

	Not at all engaged						*Extremely engaged*
How completely were your senses engaged?	1	2	3	4	5	6	7

	Strongly Disagree						*Strongly Agree*
The experience caused real feelings and emotions for me.	1	2	3	4	5	6	7

	Strongly Disagree						*Strongly Agree*
I was so involved that I lost track of time.	1	2	3	4	5	6	7

	Strongly Disagree						*Strongly Agree*
I found the control mechanism distracting.	1	2	3	4	5	6	7

	Not well at all						*Very well*
How well could you concentrate on the interaction rather than on the mechanisms used to perform the interaction?	1	2	3	4	5	6	7

Part E

Please circle the number that best represents your answer to each question.

	Not enjoyable at all						*Extremely enjoyable*
How enjoyable was your experience controlling DragonBot?	1	2	3	4	5	6	7

	Strongly Disagree						*Strongly Agree*
I found that the robot's motor noise distracted me from the interaction.	1	2	3	4	5	6	7
I found that the robot's movement distracted me from the interaction.	1	2	3	4	5	6	7
I found the robot to be intuitive to control.	1	2	3	4	5	6	7

What features/properties of the current implementation of the robot did you particularly like/dislike:

____

____

____

____

What features/properties do you think would be most important in the future design of a robot teleoperation system like the one you used?

____

____

____

____

Part F

Mental Demand How mentally demanding was the task?

Very Low — Very High

Physical Demand How physically demanding was the task?

Very Low — Very High

Temporal Demand How hurried or rushed was the pace of the task?

Very Low — Very High

Performance How successful were you in accomplishing what you were asked to do?

Perfect — Failure

Effort How hard did you have to work to accomplish your level of performance?

Very Low — Very High

Failure How insecure, discouraged, irritated, stressed, and annoyed were you?

Very Low — Very High

Part G

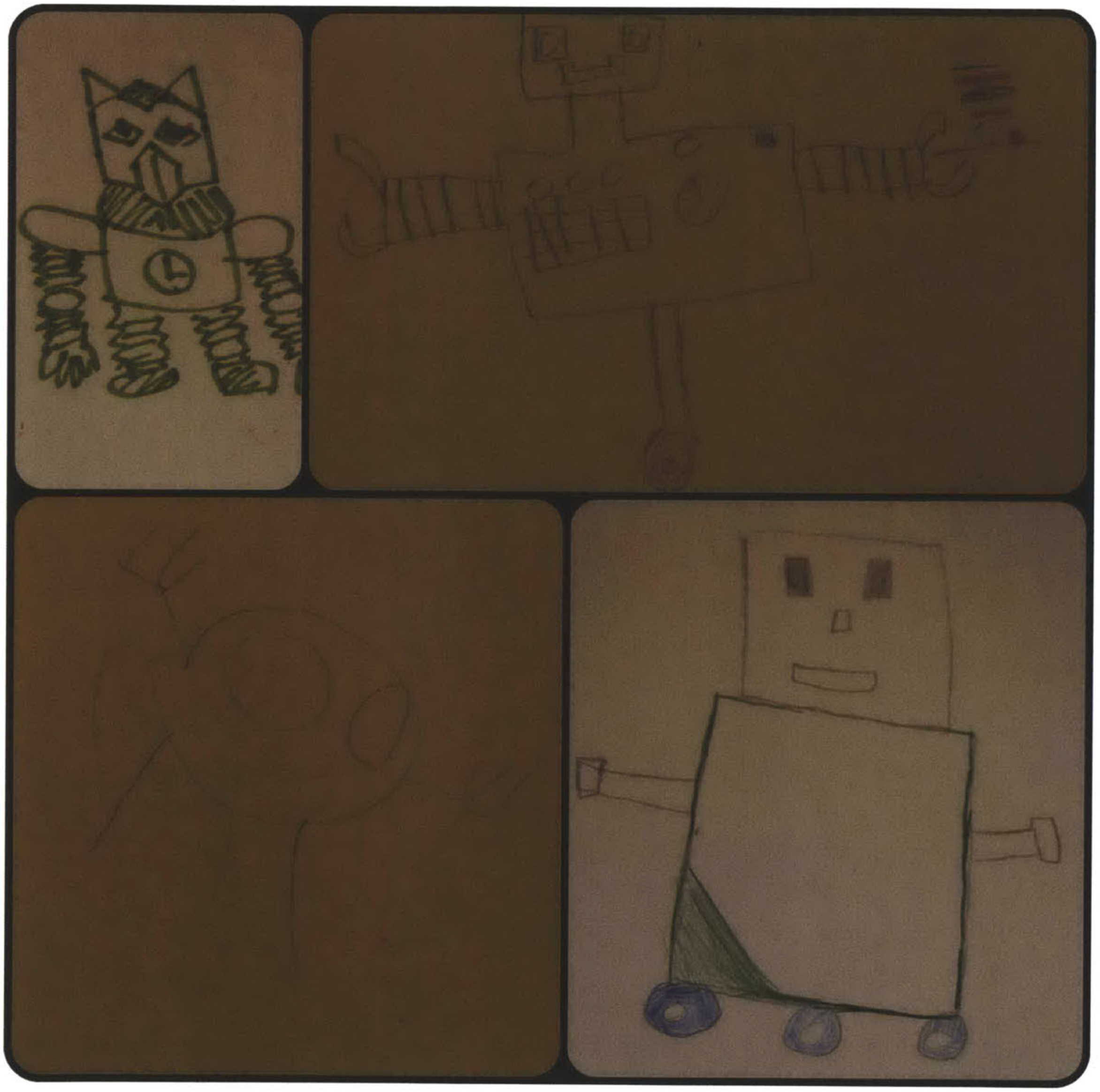
	Strongly Disagree						Strongly Agree
I found the manual interface easy to use.	1	2	3	4	5	6	7
I found the shared interface easy to use.	1	2	3	4	5	6	7
I preferred the shared interface over the manual interface.	1	2	3	4	5	6	7
I think the child preferred the shared interface over the manual interface.	1	2	3	4	5	6	7

C.3 Pictures of robots

The following pictures of robots were drawn by the participants in our user evaluation while their parents were filling out consent forms. We'll let the pictures do the talking.


and turtles
I LUV ROBOTS
FROG BOT


Plex



Bibliography

[1] D. Magazine, "The rise of the machines is not going as we expected," May 2010.

[2] J. Hays and A. A. Efros, "Scene completion using millions of photographs," *ACM Transactions on Graphics (SIGGRAPH 2007)*, vol. 26, no. 3, 2007.

[3] DOMO and C. F. Media, "How much data is created every minute?," 2012.

[4] T. Brants, A. Popat, P. Xu, F. Och, and J. Dean, "Large language models in machine translation," in *In EMNLP*, Citeseer, 2007.

[5] A. Lopez, "Statistical machine translation," 2012.

[6] E. Guizzo, "Cloud robotics: Connected to the cloud, robots get smarter," *IEEE Spectrum. Available via http://spectrum. ieee. org/automaton/robotics/robotics-software/cloud-robotics*, 2011.

[7] J. Short, E. Williams, and B. Christie, "The social psychology of telecommunications," 1976.

[8] C. Moore and P. J. Dunham, *Joint Attention - Its Origins and Role in Development*. Lawrence Erlbaum Associates, 1995.

[9] R. Gockley, A. Bruce, J. Forlizzi, M. Michalowski, A. Mundell, S. Rosenthal, B. Sellner, R. Simmons, K. Snipes, A. Schultz, *et al.*, "Designing robots for long-term social interaction," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 1338–1343, IEEE, 2005.

[10] J. J. Lee, "Modeling the dynamics of nonverbal behavior on interpersonal trust for human-robot interactions," Master's thesis, Massachusetts Institute of Technology, 2011.

[11] M. Lee, J. Forlizzi, P. Rybski, F. Crabbe, W. Chung, J. Finkle, E. Glaser, and S. Kiesler, "The snackbot: documenting the design of a robot for long-term human-robot interaction," in *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pp. 7–14, ACM, 2009.

[12] C. Kidd and C. Breazeal, "Robots at home: Understanding long-term human-robot interaction," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 3230–3235, IEEE, 2008.

[13] T. Kanda, R. Sato, N. Saiwaki, and H. Ishiguro, "A two-month field trial in an elementary school for long-term human–robot interaction," *Robotics, IEEE Transactions on*, vol. 23, no. 5, pp. 962–971, 2007.

[14] T. Kanda, M. Shimada, and S. Koizumi, "Children learning with a social robot," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pp. 351–358, ACM, 2012.

[15] T. Bickmore and R. Picard, "Towards caring machines," in *CHI'04 extended abstracts on Human factors in computing systems*, pp. 1489–1492, ACM, 2004.

[16] T. Bickmore and R. Picard, "Establishing and maintaining long-term human-computer relationships," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 12, no. 2, pp. 293–327, 2005.

[17] R. Wistort and C. Breazeal, "Tofu: a socially expressive robot character for child interaction," in *Proceedings of the 8th International Conference on Interaction Design and Children*, pp. 292–293, ACM, 2009.

[18] H. Kozima, M. Michalowski, and C. Nakagawa, "Keepon," *International Journal of Social Robotics*, vol. 1, no. 1, pp. 3–18, 2009.

[19] W. Stiehl, J. Lieberman, C. Breazeal, L. Basel, L. Lalla, and M. Wolf, "The design of the huggable: A therapeutic robotic companion for relational, affective touch," in *AAAI Fall Symposium on Caring Machines: AI in Eldercare, Washington, DC*, 2005.

[20] W. Stiehl, C. Breazeal, K. Han, J. Lieberman, L. Lalla, A. Maymin, J. Salinas, D. Fuentes, R. Toscano, C. Tong, *et al.*, "The huggable: a therapeutic robotic companion for relational, affective touch," in *ACM SIGGRAPH 2006 Emerging technologies*, p. 15, ACM, 2006.

[21] J. Velásquez, "Modeling emotions and other motivations in synthetic agents," in *Proceedings of the National Conference on Artificial Intelligence*, pp. 10–15, Citeseer, 1997.

[22] J. Velásquez and R. Adviser-Brooks, *When robots weep: a computational approach to affective learning*. PhD thesis, Massachusetts Institute of Technology, 2007.

[23] C. Breazeal and R. Brooks, "Robot emotion: A functional perspective," *Who needs emotions*, pp. 271–310, 2005.

[24] T. Kubik and M. Sugisaka, "Use of a cellular phone in mobile robot voice control," in *SICE 2001. Proceedings of the 40th SICE Annual Conference. International Session Papers*, pp. 106–111, IEEE, 2001.

[25] B. Kaufmann and L. Buechley, "Amarino: a toolkit for the rapid prototyping of mobile ubiquitous computing," in *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, pp. 291–298, ACM, 2010.

[26] C. team, "Cellbots: Using cellphones as robotic control platforms," Aug. 2012.

[27] Romotive, "Romo: The smart phone robot," Aug. 2012.

[28] C. Delaunay, "Botiful, social telepresence robot for android," Aug. 2012.

[29] R. Goertz, "Fundamentals of general-purpose remote manipulators," *Nucleonics (US) Ceased publication*, vol. 10, 1952.

[30] L. Li, B. Cox, M. Diftler, S. Shelton, and B. Rogers, "Development of a telepresence controlled ambidextrous robot for space applications," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 1, pp. 58–63, IEEE, 1996.

[31] F. Michaud, P. Boissy, H. Corriveau, A. Grant, M. Lauria, D. Labonte, R. Cloutier, M. Roux, M. Royer, and D. Iannuzzi, "Telepresence robot for home care assistance," in *AAAI Spring Symposium on Multidisciplinary Collaboration for Socially Assistive Robotics*, 2007.

[32] G. Ballantyne, "Robotic surgery, telerobotic surgery, telepresence, and telementoring," *Surgical Endoscopy*, vol. 16, no. 10, pp. 1389–1402, 2002.

[33] T. Sheridan, "Space teleoperation through time delay: Review and prognosis," *Robotics and Automation, IEEE Transactions on*, vol. 9, no. 5, pp. 592–606, 1993.

[34] J. Corde Lane, C. Carignan, B. Sullivan, D. Akin, T. Hunt, and R. Cohen, "Effects of time delay on telerobotic control of neutral buoyancy vehicles," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 3, pp. 2874–2879, IEEE, 2002.

[35] I. Nourbakhsh *et al.*, "Robots and education in the classroom and in the museum: On the study of robots, and robots for study," 2000.

[36] D. Glas, T. Kanda, H. Ishiguro, and N. Hagita, "Simultaneous teleoperation of multiple social robots," in *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, pp. 311–318, ACM, 2008.

[37] J. Saldien, B. Vanderborght, and D. Lefeber, "The social robotplatform probo," in *Proceedings of the 28th Annual European Conference on Cognitive Ergonomics*, pp. 363–364, ACM, 2010.

[38] A. Kon, "The shared decision-making continuum," *JAMA: The Journal of the American Medical Association*, vol. 304, no. 8, pp. 903–904, 2010.

[39] A. Ansari, S. Essegaier, and R. Kohli, "Internet recommendation systems," *Journal of Marketing research*, vol. 37, no. 3, pp. 363–375, 2000.

[40] A. Rogers-Warren and D. Baer, "Correspondence between saying and doing: teaching children to share and praise," *Journal of Applied Behavior Analysis*, vol. 9, no. 3, p. 335, 1976.

[41] N. Garon, B. Johnson, and A. Steeves, "Sharing with others and delaying for the future in preschoolers," *Cognitive Development*, 2011.

[42] S. Dreman and C. Greenbaum, "Altruism or reciprocity: Sharing behavior in israeli kindergarten children," *Child Development*, pp. 61–68, 1973.

[43] M. Lerner, "The justice motive:" equity" and" parity" among children.," *Journal of Personality and Social Psychology; Journal of Personality and Social Psychology*, vol. 29, no. 4, p. 539, 1974.

[44] C. Moore and S. Macgillivray, "Altruism, prudence, and theory of mind in preschoolers," *New Directions for Child and Adolescent Development*, vol. 2004, no. 103, pp. 51–62, 2004.

[45] B. Handlon and P. Gross, "The development of sharing behavior," *Journal of Abnormal and Social Psychology*, vol. 59, no. 3, pp. 425–428, 1959.

[46] I. Lane and R. Coon, "Reward allocation in preschool children," *Child Development*, pp. 1382–1389, 1972.

[47] K. Olson and E. Spelke, "Foundations of cooperation in young children," *Cognition*, vol. 108, no. 1, pp. 222–231, 2008.

[48] C. Moore, "Fairness in children's resource allocation depends on the recipient," *Psychological Science*, vol. 20, no. 8, pp. 944–948, 2009.

[49] F. Warneken, K. Lohse, A. Melis, and M. Tomasello, "Young children share the spoils after collaboration," *Psychological science*, vol. 22, no. 2, pp. 267–273, 2011.

[50] C. Breazeal, *Designing sociable robots*. The MIT Press, 2004.

[51] D. Robert, "Imaginative play with blended reality characters," Master's thesis, Massachusetts Institute of Technology, 2011.

[52] F. Thomas and O. Johnston, *The Illusion of Life: Disney Animation*. Abbeville Press New York, 1981.

[53] J. Lasseter, "Principles of traditional animation applied to 3d computer animation," *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 35–44, 1987.

[54] L. Takayama, D. Dooley, and W. Ju, "Expressing thought: improving robot readability with animation principles," in *Proceedings of the 6th international conference on Human-robot interaction*, pp. 69–76, ACM, 2011.

[55] J. Gray, G. Hoffman, S. O. Adalgeirsson, M. Berlin, and C. Breazeal, "Expressive, interactive robots: Tools, techniques, and insights based on collaborations," in *HRI 2010 Workshop: What do collaborations with the arts have to say about HRI?*, 2010.

[56] M. Minsky, "Manipulator design vignettes," Technical Memo 276, Massachusetts Institute of Technology, A.I. Laboratory, MIT, Cambridge, USA, Oct. 1972.

[57] F. Pierrot, V. Nabat, S. Krut, P. Poignet, *et al.*, "Optimal design of a 4-dof parallel manipulator: from academia to industry," *Robotics, IEEE Transactions on*, vol. 25, no. 2, pp. 213–224, 2009.

[58] J. Merlet, *Parallel robots*. Springer-Verlag New York Inc, 2006.

[59] A. Thomaz and M. Cakmak, "Learning about objects with human teachers," in *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pp. 15–22, ACM, 2009.

[60] C. Sidner, C. Lee, L. Morency, and C. Forlines, "The effect of head-nod recognition in human-robot conversation," in *Proceedings of the 1st ACM SIGCHI/SIGART conference on human-robot interaction*, pp. 290–296, ACM, 2006.

[61] P. Corporation, "Pololu robotics and electronics," Aug. 2012.

[62] Y. Ben-Tsvi, "Ioio wiki," 2011.

[63]

[64] S. O. Adalgeirsson, "Mebot: A robotic platform for socially embodied telepresence," Master's thesis, Massachusetts Institute of Technology, 2009.

[65] Y. Park, C. Majidi, R. Kramer, P. Bérard, and R. Wood, "Hyperelastic pressure sensing with a liquid-embedded elastomer," *Journal of Micromechanics and Microengineering*, vol. 20, p. 125029, 2010.

[66] I. S. A. Mediterranee, "Singularities of parallel robots," Oct. 2011.

[67] R. Picard, "Affective computing. 1997."

[68] R. L. SDK, *version 4.0.0.0*. Richardson, Texas: Annosoft, 2009.

[69] S. Hart and L. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," *Human mental workload*, vol. 1, pp. 139–183, 1988.

[70] J. Hauber, H. Regenbrecht, A. Hills, A. Cockburn, and M. Billinghurst, "Social presence in two-and three-dimensional videoconferencing," 2005.

[71] S. Adalgeirsson and C. Breazeal, "Mebot: a robotic platform for socially embodied presence," in *Proceeding of the 5th ACM/IEEE international conference on Human-robot interaction*, pp. 15–22, ACM, 2010.

[72] F. Biocca, C. Harms, and J. Gregg, "The networked minds measure of social presence: Pilot test of the factor structure and concurrent validity," in *4th annual International Workshop on Presence, Philadelphia, PA*, 2001.

[73] J. Van Baren and W. IJsselsteijn, "Measuring presence: A guide to current measurement approaches," 2004.