

EchoBot: Facilitating Data Collection for Robot Learning with the Amazon Echo

Rishi Kapadia



Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/Eecs-2017-40

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/Eecs-2017-40.html>

May 9, 2017

Copyright © 2017, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

**EchoBot: Facilitating Data Collection for Robot Learning with the
Amazon Echo**


by Rishi Kapadia

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II.**

Approval for the Report and Comprehensive Examination:

Committee:



Professor Ken Goldberg
Research Advisor

7 May 2017

(Date)



Professor Anca Dragan
Second Reader

9 May 2017

(Date)

EchoBot: Facilitating Data Collection for Robot Learning with the Amazon Echo

by

Rishi Kapadia

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ken Goldberg, Chair

Professor Anca Dragan

Spring 2017

Abstract

EchoBot: Facilitating Data Collection for Robot Learning with the Amazon Echo

by

Rishi Kapadia

Master of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Ken Goldberg, Chair

The Amazon Echo and Google Home exemplify a new class of home automation platforms that provide intuitive, low-cost, cloud-based speech interfaces. We present EchoBot, a system that interfaces the Amazon Echo to the ABB YuMi industrial robot to facilitate human-robot data collection for Learning from Demonstration (LfD). EchoBot uses the computation power of the Amazon cloud to robustly convert speech to text and provides continuous speech explanations to the user of the robot during operation. We study human performance with two tasks, grasping and “Tower of Hanoi” ring stacking, with four input and output interface combinations. Our experiments vary speech and keyboard as input interfaces, and speech and monitor as output interfaces. We evaluate the effectiveness of EchoBot when collecting infrequent data in the first task, and evaluate EchoBot’s effectiveness with frequent data input in the second task. Results suggest that speech has potential to provide significant improvements in demonstration times and reliability over keyboards and monitors, and we observed a 57% decrease in average time to complete a task that required two hands and frequent human input over 11 participants.

Contents

Contents	i
Acknowledgements	iii
1 Introduction.....	1
1.1 Motivation.....	1
1.2 System Criteria.....	2
1.3 System Overview	2
2 Related Work	4
2.1 Data Collection for Learning from Demonstration.....	4
2.2 Home Automation Systems	4
2.3 Robots and Speech.....	5
3 System Design.....	7
3.1 System Architecture.....	8
3.2 Communication with Amazon	8
3.3 EchoBot Audio Output	9
3.4 Data Collection for Classifier Calibration	10
4 Extended Applications.....	12
4.1 Camera Calibration	12
4.2 Interaction	13
5 Human Performance Studies	15
5.1 Study Procedures	15
5.2 Subject Population	16
5.3 Risks and Discomforts	16
6 Grasp Task	18
6.1 Study Setting.....	18
6.2 Study Procedure	18
6.3 Results.....	20
6.4 Reflections and Enhancements	23
7 Ring-Stacking Task.....	25
7.1 Study Setting.....	25
7.2 Study Procedure	26
7.3 Results.....	26
7.4 Reflections and Enhancements	29
8 Conclusion	31

9 Appendix	32
9.1 EchoBot Setup	32
9.2 Using EchoBot.....	32
9.3 Extending EchoBot.....	35
Bibliography	36

Acknowledgements

I would like to thank my research advisor, Professor Ken Goldberg, for all his advice and guidance throughout this project. I am grateful for his support and counsel throughout this project, and for teaching me the skills to become a better researcher. I thank Professor Goldberg for showing me how to effectively articulate both the strengths and weaknesses of the results of my research, and his feedback on how to write an academic paper. I thank Assistant Professor Anca Dragan for being on my masters' thesis committee, and the resources from her class on Algorithmic Human-Robot Interaction.

I would also like to thank the members of the AUTOLAB at UC Berkeley for their advice and support throughout my years in this research lab. I thank Dr. Stephen McKinley for being my first mentor in this lab and inviting me to collaborate with him on my first paper. I thank Dr. Animesh Garg and Dr. Lauren Miller for introducing me to applications of surgical robotics, and allowing me to explore my interests in and computer vision and 3D geometry. I thank Jeff Mahler for his help in integrating my work on EchoBot with the YuMi robot. I thank Sanjay Krishnan for his help in how to research related academic works. I would also like to thank Jeff Mahler, Michael Laskey, Sanjay Krishnan, and Roy Fox, for their advice on the directions of my research with the Amazon Echo and teaching me what the academic community looks for in a research paper. I thank Sam Staszak for her support in formulating and running experiments, and working with me until the submission deadline on my first 1st-author paper. I would like to thank Raul Puri for his help in getting started with and how to use the Amazon Echo as a tool for development. I thank Lisa Jian for her help in analyzing and presenting the results in my paper and in this thesis. I thank my colleagues who provided helpful feedback and suggestions leading to this report, in particular Ken Goldberg, Sam Staszak, Jeff Mahler, Michael Laskey, Sanjay Krishnan, Roy Fox, Daniel Seita, Nate Armstrong, and Christopher Powers. I would like to thank my parents and sister for their love and support, for encouraging me to pursue my interests outside of my comfort zone, and for having believing in me throughout my education.

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab, the Real-Time Intelligent Secure Execution (RISE) Lab, and the CITRIS "People and Robots" (CPAR) Initiative.

1 Introduction

With the emergence of voice activation systems, a new class of home automation platforms has appeared in the commercial market. These systems utilize speech recognition and natural language processing to facilitate interactions with devices. Although voice-enabled interfaces are still in their early stages, they have been used to interact with speakers, smartphones, television sets, and other electronics. Speech interfaces also have potential to enhance the efficiency of interactions with robots. A common interaction between humans and robots is to train the robot to perform a task according to a desired policy. One approach to training robots is Learning from Demonstration (LfD), where a human provides several demonstrations of a task to the robot, and the robot learns to perform that task. These demonstrations may consist of segments of arm trajectories or keyframes of robot poses at periodic time intervals, and are specified to the robot as input. The robot uses this collected data to learn a policy to perform the task.

We explore how a voice activation system may improve data collection for LfD. Using the Amazon Echo [4], we implemented EchoBot^{1,2}, a 2-way speech interface for communication between humans and robots during data collection for robot learning.

1.1 Motivation

Learning from Demonstration (LfD) is a promising approach to teaching robots via demonstrations of a desired behavior. There currently exist several methods for providing these demonstrations, including kinesthetic teaching, whereby the robot's passive joints are guided through the performance of the desired motion [5]. The studies in [2, 43] rely on dialogue systems that enable subjects to provide speech commands to effectively control the collection of kinesthetic demonstrations. We use a commercial home automation system as our voice interface to the robot to aid in collecting data for LfD.

A motivation for involving speech systems into robotics is to facilitate interactions between human workers and robots in the workplace. A speech interface may help to create a persona for the robot, and establish common ground and likability. It may possibly be

¹ Code is available at <https://github.com/rishikapadia/echoyumi>

² Video available at <https://www.youtube.com/watch?v=XgaGeCsERU8>

more scalable in the future to have workers specify instructions to the robot using voice commands as opposed to typing directions, or safer than pointing to objects in the workplace. Another important motivation for introducing an audio interface is that humans have a limited working memory. However, humans are able to integrate multiple modalities, such as visual and auditory. EchoBot uses the auditory modality to assist the user in the visual and kinesthetic task of data collection.

1.2 System Criteria

We focus on a speech interface as a medium for input and output. We assume that there is only one person speaking during the command transmission and that the user knows what keywords issue each command. To be usable and convenient to the user, response times to user commands must have a latency similar to the delay between two conversing humans.

1.3 System Overview

We integrated the Amazon Echo and the YuMi industrial robot from ABB [1] into EchoBot. The Echo is a wireless speaker and voice command device. It is a low-cost, commercially-available product with a text-to-speech interface for natural language processing. We use the Echo for its speech recognition system, which is robust to voice from different locations and at different pitches and intonations. The YuMi is a dual-arm, human-safe industrial robot with flexible joints and grippers, and offers state-of-the-art robot control. EchoBot as a system allows users to utter commands to the Echo that are relayed as actions to the YuMi robot, and communicates vocal feedback from the robot back to the user while performing those actions.

We evaluate EchoBot in 2 experiments as an input and output interface to perform data collection using robots, and find that it increases collection efficiency when the user needs to input data frequently, and both of the user's hands are occupied with the task.

This paper contributes:

- 1) The first implemented system architecture interfacing the Amazon Echo home automation speech interface to the ABB YuMi industrial robot.
- 2) Experiments with two robot tasks, grasping and "Tower of Hanoi" ring stacking, comparing four interface combinations varying input and output with keyboard, monitor, and speech.

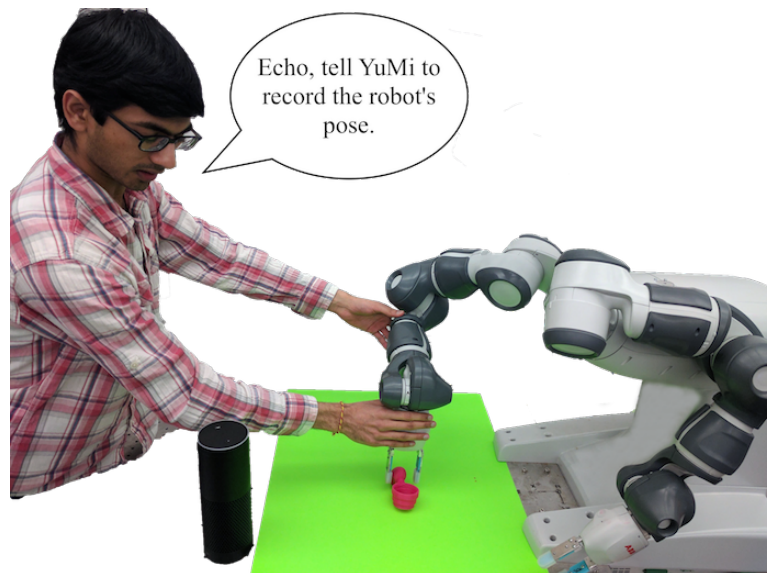


Figure 1: EchoBot integrates the Amazon Echo speech interface with the ABB YuMi industrial robot for data collection.

2 Related Work

2.1 Data Collection for Learning from Demonstration

LfD is a promising approach to teach policies to robots through demonstrations of a desired behavior. There currently exist several methods for providing demonstrations to robots, including teleoperation and kinesthetic teaching [5]. In teleoperation, the demonstrator controls the end-effector position or joint angles of the robot using a device such as a joystick or game controller [25]. In kinesthetic teaching, the human physically guides the robot's arms and grippers to complete the task. The robot uses several such demonstrations of the task as a sampling of the policy intended by the human, and attempts to find the underlying policy to perform the task. Both data collection methods often require the demonstrator to use both hands, which can complicate denoting the start and end of a demonstration using a game controller or button press.

Several studies have utilized voice commands to facilitate data collection of kinesthetic demonstrations in LfD systems. In [43], subjects were tasked with using voice commands to start and end demonstrations, and afterward the robot reproduced the learned skill. The speech interface of Akgun et. al [2] had similar functionality. The kinesthetic trajectories were provided in keyframes, where voice commands were used to indicate where the demonstration was segmented by the subject. In addition to keyframe and segmentation functionality, EchoBot also provides the user with prompts and continuous audio output detailing the status of the robot.

2.2 Home Automation Systems

The Echo has been used as an interface for products by Uber, StubHub, Fitbit, Domino's Pizza, and many others. Samsung revealed in late 2016 that all of its WiFi-enabled robotic vacuums can now be controlled using the Echo. Other voice interfaces include Apple's Siri in 2011 and Homekit in 2014, a collection of smart devices for users to control around the house. In November 2016, Google Home was introduced, which offers the capability of Google, Inc.'s search engine. Other smart assistants include ivee

Voice, Cubic, Mycroft, Sonos Play, Hal, Comcast Xfinity TV remote, and many others. To our knowledge, EchoBot is the first of these home automation systems that provides voice interaction to an industrial robot for data collection.

Prior work has studied voice activation to control robots and other machines. An early instance of voice recognition used in surgical robotic assistants controlled the end effector location of a robotic arm during laparoscopic surgery [34], where the surgeon could command the arm to move in the 3 axial directions or to predefined locations at a constant speed. Furthermore, the constraints or failure modes of the robot were conveyed audibly to the surgeon, such as when a joint had exceeded its limits, to prevent damage to the robot and patient. Dean et al. [10] used a speech interface to control the da Vinci, a tele-operation robotic surgical system, to perform simple tasks like measuring the distance between two locations and manipulating visual markers on the display. Henkel et al. [19] describes an open-source voice interaction toolkit to serve as a medium between dependent victims, such as trapped earthquake survivors, and the outside world. Gesture and voice interfaces were developed to help disabled people operate a remote controller for home automation [21], to facilitate rehabilitation for people with disabilities [26], and to help children with autism [33]. Other examples of using voice control for robots include [16, 18, 29, 42].

2.3 Robots and Speech

Ray et al. [32] found that humans prefer to interact with robots using speech. Cha et al. [8] have shown that human perception of robot capability in physical tasks can be affected by speech. Takayama et al. [37] found that perception of robots is influenced not only by robots showing forethought, but also by the success outcome of the task and showing goal-oriented reactions to task outcomes. The acceptance of robots is important in making robots a part of workplaces and homes [6, 9], and the perceived capability of robots largely influences robot acceptance [9]. Srinivasa et al. [36] used a speech synthesis module on their robotic butler to interact with humans while completing tasks such as collecting mugs. When humans engaged with robots using speech, their confidence that the robot was a reliable source of information was shown to increase [28]. It has also been shown that there are noticeable drops in trust as reliability of the robot decreases [11, 13, 14], and only once reliability recovers does trust start to increase monotonically [12, 27]. These works may suggest that in the event of failures, conversational speech might be able to help restore trust in the robot's capability, and that the content of the speech has an impact on the effectiveness of a robot.

Kollar et al. [24] extracted a sequence of directional commands from linguistic input for a humanoid robot or drone to follow. Tellex et al. [38] trained an inference model with a crowdsourced corpus of commands to allow humans to manipulate an autonomous robotic forklift with natural speech commands. In cases where the robot was told to perform an action that it did not understand, Cantrell et al. [7] demonstrated an algorithm where a human could explain the meaning of that action to the robot, and the robot would then be able to carry out instructions involving that action. There are also examples of robots that modify their behavior based on the circumstance, such as [35, 40, 41], and studies of how humans expect to interact with a robot [15, 17, 22, 23, 30].

3 System Design

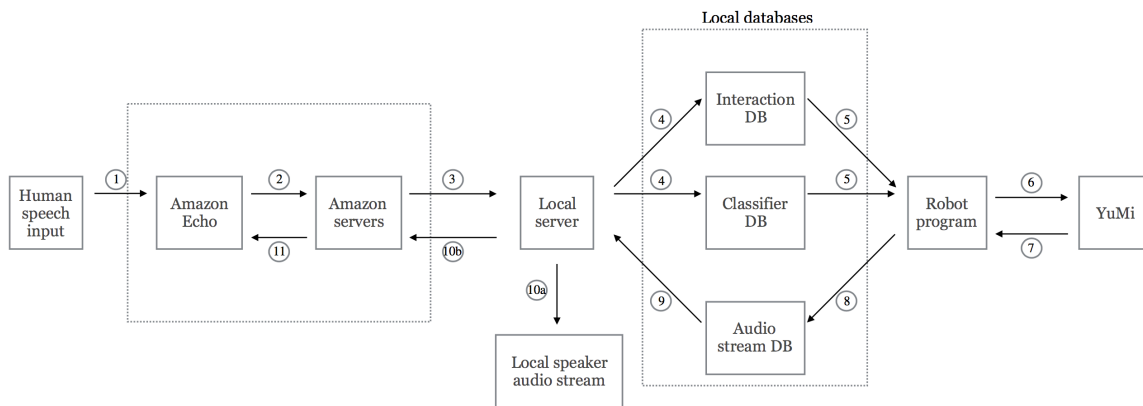


Figure 2: System diagram. When (1) a user asks the Echo a question, (2) a request is sent to Amazon's servers over WiFi, which converts the speech to a robot command. (3) Amazon's servers send an HTTPS request to our server running on our local computer. (4) Our local server communicates actions to either the classifier or interaction database, depending on the command. (5) The robot manipulation program polls that database for a new command, and (6-7) communicates the actions and responses via Ethernet to the YuMi robot. (8) The robot program logs messages to the audio stream database, (9) which is polled by the local server. (10a) That message is played to the user through the computer's speakers in an audio stream. After the local server received the HTTPS request in step (2) and logs to the appropriate database (3), it returns an HTTPS response back to Amazon's server (10b), which relays the "end of request" command to the Echo (11).

3.1 System Architecture

To enable the Amazon Echo to communicate with the ABB YuMi robot, we implemented a web server on a local Linux desktop computer using the Django web framework, implemented in the Python programming language.

Our local Django web server is based on the `pycontribs/django-alexa` repository, which is publicly available on GitHub.com. We modified the code in the public repository to support the current format of Amazon's JSON messages and to handle our own custom application on the Amazon Echo.

Our web application exposes a REST API endpoint at the relative address `/alexa/ask` of our server to communicate with the Amazon Echo (see Figure 2). This endpoint handles all incoming HTTPS requests and dispatches them to the appropriate Python functions that we define for various commands to EchoBot. We specify the public web address of our local server on the Alexa Skills Kit [3] web portal.

3.2 Communication with Amazon

Since we created a custom application on the Echo, Amazon requires that we specify a comprehensive, textual list of commands on the Alexa Skills Kit web portal beforehand. One or more invocation phrases, or human speech commands, must be specified for each robot command, and providing more phrases increases the robustness of Amazon's speech-to-command correspondence algorithm. Given this set of predefined possible human phrases to robot commands, Amazon's servers then compute the closest match of the speech to the corresponding command. These commands can contain parameters, or arguments, which are words or phrases that are variable in a given command. However, the set of possible parameters must be defined beforehand as well, which means that the system is unable to handle wildcard phrases for custom commands.

The Amazon voice service also places constraints on what a user must say to convey a human speech command. First, the Echo must be triggered by a "wake word", which can be either "Alexa", "Echo", or "Amazon", where we have chosen to use "Echo". Then, the user must specify their command in the form of `<action> <connecting word> <application name> <command>`, where we have named our application "YuMi", the connecting word is optional, and "command" refers to an invocation phrase. For example, to issue the command for the robot to grasp all parts, the user could say, "Echo, ask YuMi to pack all of the parts" or "Echo, tell YuMi to pack all of the parts". The command prefix must be included in the human speech command because we created a custom application

with the Echo, rather than a native Amazon feature such as time or weather reports. If users want to issue frequent voice commands to the Echo, they may only say *<command>* for all following robot commands, provided that each command is issued within 5 seconds of the previous command.

When a user speaks a command to the Echo, an HTTPS request is sent over WiFi to Amazon's servers to convert the speech to text. The Amazon server then makes another HTTPS request in JSON format to our local server with the name of that command and potentially any parameters. The mechanics from the Amazon Echo to Amazon's servers are an abstraction, and the interface Amazon provides is speech as input and JSON data as output. Our local server parses the received JSON request and calls the function corresponding to that command name with any required or optional parameters. That function communicates the appropriate actions to a robot manipulation program via a database connected to our local server and accessible from anywhere on that same machine. The robot manipulation program communicates directly to the robot via Ethernet. Upon completion, the function on our local server sends an HTTPS response, also in JSON format, back to Amazon's servers to be sent back to the Echo. This response may contain a phrase to be spoken through the Echo's speakers to the user, a signal for the Echo to continue listening for more commands, or a signal indicating the end of the command (see Figure 2).

The delay between the time the user finishes speaking to the Echo and the time our local server receives the HTTPS request is on average 2.1 seconds (see Figure 2, steps 1-3). The delay between the time the user finishes speaking to the Echo and the time the Echo receives the HTTPS response is 2.2 seconds (see Figure 2, steps 1-3, 10b-11). This includes 0.5 seconds of delay after the user finishes speaking for the Echo to register that there is no more speech to send to Amazon's servers.

3.3 EchoBot Audio Output

The Amazon Echo API does not allow for the Echo to speak a series of phrases unless the user actively queries each one. Therefore, we routed the audio stream to the local server's computer speakers instead, to give the user explanation updates almost in real-time (see Figure 2).

To launch the audio stream, a user states, "Echo, ask YuMi to explain what it is doing." (See Figure 2 steps 1-3, 10a.) Then, messages are logged from the main robot script to the audio stream database used by the local server (see Figure 2 step 8). As a message arrives into the audio stream database queue, the server converts the message into an audio

signal using a text-to-speech library based on Google Translate's web API [39]. The server polls the database every 0.04 seconds, so the user is presented explanations with imperceptible latency as the messages arrive (see Figure 2 step 9). In the case where several messages are logged at the same time, only the last one is played, to reduce queuing delays. We have found that polling the database scales well as the total number of messages grows, since only the last message in the database needs to be checked at each poll.

Google Translate provides speech audio that sounds very natural, but because the text-to-speech library makes an HTTP request to convert the audio, translations would incur an average latency of approximately 2.5 seconds. Therefore, our local server caches these text-to-audio translations for immediate access, and defaults to a different text-to-speech library [31] that incurs latency on the order of 10 milliseconds.

We also utilized speech, music, and sound effects in an attempt to humanize the robot using EchoBot. When there are no explanations in the queue to play to the user, relaxing but interesting background music is played to fill in the silent gaps between messages. If the background music is not desired, there is also a command to disable it, or play a different song using the built-in functionality of the Amazon Echo.

3.4 Data Collection for Classifier Calibration

Many LfD demonstrations require collecting poses or trajectories of the robot's arm positions as a human physically guides the arms and grippers. These trajectories may include several segments, where the endpoints of each segment are recorded. An example segment may be moving an arm's end effector from one location to another, or the actions of closing and opening the gripper. Robot arm trajectories can be recorded using buttons to specify the endpoints of each trajectory segment, where one button maps to a "start recording" command and another button maps to a "stop recording" command. This method has several shortcomings, including that demonstrators:

- 1) Often need both hands to control the robot's movements and can't stop to press a button.
- 2) Have to remember the mapping of buttons to commands.
- 3) Have to check the monitor display for cues on when the button press has been registered and the robot is prepared to record a trajectory.

EchoBot allows the user to speak commands such as "Start recording" and "Stop recording" to the Echo and achieve more intuitive control over data collection. Internally, when our server receives the user command, it sends the command to the main robot script via the classifier database (see Figure 2, step 4). Concurrent accesses are not a major issue

for this database. It is modified only when EchoBot receives a data collection command, which is limited by the rate of incoming speech, and when the robot script retrieves it, where the database is only cleared once a command has been written to it, and therefore also limited by the rate of incoming speech. The classifier database is polled at least every 0.1 seconds by the robot program (see Figure 2, step 5).

4 Extended Applications

In addition to being a tool to interface directly with data collection methods for LfD training, EchoBot can also help users to collect and use data in tasks tangential to the data collection tasks described above. In the following subsections, we describe applications of EchoBot that help parts of the robot development work flow that may benefit from a speech-driven interface.

4.1 Camera Calibration

Calibrating the camera is the first task to be done before starting any development on the robot. In case someone needs help with calibration, EchoBot can be of service. This is helpful in teaching inexperienced roboticists how to calibrate the camera used by the robot to more accurately register 2D points in the image to 3D points in the workspace. After the user asks EchoBot, "How do I calibrate the camera?", the Echo gives a step-by-step walkthrough of how to do so, and runs the calibration scripts in the background between each step.

For example, EchoBot first instructs the user to place a checkerboard in a specific location for camera registration, and waits for the user to respond with a "continue" command. Users may also respond with "OK, what's next?", "Sounds good, what should I do now?", or various other phrases to advance to the next step. After the user prompts EchoBot for the next step, the Unix command for the appropriate calibration script will run. Then, EchoBot will ask the user to place the checkerboard in a different but deterministic location, waits for the user to indicate he or she is ready to move on, and runs a different calibration script. After the user reaches the last step, the Echo provides a confirmation message to the user that the walkthrough has been completed.

In totality, the input interface of EchoBot for this task consists of both human speech commands and the image of the workspace from the camera. EchoBot is able to parse the human's command and respond according to the current step of calibration and what it sees in the workspace. As output, it then relays the next step to the user through a speech prompt, or indicates to the user that the scene does not appear as expected, such as if the checkerboard cannot be detected.

In addition to being a tool for tutorials, this shows that EchoBot can be used as a tool for storing help information or other data about robots, similar to a manual page for a Unix command. EchoBot may also be used to communicate other kinds of data from robots to the user, including robot states not visible to the user.

4.2 Interaction

Another form of data collection is to interact with the robot to run experiments. In an experiment to grasp specific object parts successfully with the YuMi robot arm and gripper, it may be useful to use voice commands to identify objects to grasp by color. EchoBot allows users to specify colors by speech for the robot to grasp those objects and package them into a box. For example, a user may say, "Echo, tell YuMi that I want the blue, white, and orange parts." We found that specifying more than three color parameters to the Echo resulted in inaccurate speech-to-command registration. Thus, for reliability and consistency, we limit the number of colors to a maximum of three, with an additional command to specify to the robot to grasp all objects in the workspace.

The interface of EchoBot on this task from the user's perspective is human speech commands as input, and robot status updates in the form of speech output. After the user prompts EchoBot to grasp specific colored objects, EchoBot begins speaking the messages logged from the robot grasping script. This continuous stream of messages can last from between 20 seconds to 2 minutes, depending on how many objects were specified and how difficult they are to grasp. Internally, checkpoints in the execution of the robot script mark when and which messages should be logged to the audio stream database for speech output to the user.

After the message is passed to our local server, with parameters as the colors of the objects, our local server uses a table to convert the colors to part names, and relays the names to the robot program using the interaction database (see Figure 2, step 4). Just as for classifier calibration, concurrent accesses are not a major issue for this database either. The database is modified only when EchoBot receives a grasp-by-color command and when the robot script retrieves it. This message appears in the database within 2.2 seconds of when the command is finished being spoken to the Echo. The interaction database is polled by the robot script as soon as it is free to perform a new action (see Figure 2, step 5). This typically ranges anywhere from between 0.04 seconds, if the script is waiting for a new command, to about 2 minutes, if the script is currently grasping and packing the maximum 3 objects by color. The robot grasping program then picks out the colored parts that were specified and places them into a box. The YuMi robot can poke at the pile with one gripper

to singulate the parts from each other, and looks at the parts from right to left to grasp with the other gripper. Any miscellaneous parts that are singulated and identified first are placed in a different box. This presents the user with the ability to delegate actions to the robot and run experiment trials by voice.

5 Human Performance Studies

5.1 Study Procedures

We evaluate the effectiveness of EchoBot with respect to data collection in two human performance studies, outlined in the following two sections. We study the problem of efficiently collecting human demonstration data through kinesthetic (where an expert moves the robot through contact) interfaces with the aid of visual and audio feedback. We perform a 2-by-2 human-robot interaction study to analyze how the using the Echo and the keyboard as the input interface and a speaker and a text-based GUI as the output interface affects the efficiency of data collection.

Each subject watched and repeated demonstrations on the robot. First, the subject watched and occasionally interact with the robot during demonstrations. Examples of robot interaction include voicing commands to a speech interface; pressing a keyboard button; and physically guiding a robot arm. The robot's arm moves very slowly and safely, with minimal risk to subjects. Examples of data that were collected are robot joint angles, end-effector poses, motor torques, and anonymous questionnaire results about the demonstration. No personally-identifiable information was collected. The duration of the demonstration for each participant was about 10 to 30 minutes, and did not exceed 40 minutes. Data collected outside of the demonstration, such as questionnaire data, was also anonymous and not contain any personally-identifiable information, and took about 2 to 5 minutes to complete. We did not compensate participants for their time. Any data, such as timestamps during checkpoints in the experiments, was stored on password-secured lab computers. The collected data was analyzed in aggregate over subjects for each of our test groups, and we present the aggregated data in our research paper and this thesis, with no personally-identifiable information.

We ran 2 different experiments, each with a different group of subjects. Participants were identified using an experimenter ID, which is an integer starting at 1 and increasing sequentially for each next subject. Participants were gathered voluntarily by asking members of our research lab (for the first experiment) or members of the computer science department (for the second experiment) whether they would like to participate in a robotics experiment to evaluate different input and output interfaces for interacting with a robot, and were assured that they had full autonomy to decline participation. Participants signed a consent form in the lab before their participation began.

In the first experiment, subjects used either a keyboard or the Amazon Echo as an input device, and either a monitor or the Amazon Echo as an output device. The subject used the input and output interfaces to tell the YuMi robot to capture a picture of an object in the robot's workspace, then guided the robot's arm to within grasp of the object, and told the robot to record the arm pose of the robot. The experimenter first explained the task to the subject, had the subject practice for 2 minutes prior to the actual experiment, and then had the subject perform the same data collection task for ten minutes. After the experiment finished, the subject completed a survey questionnaire. The purpose of this experiment is to evaluate the efficiency of data collection using each of the 4 combinations of input and output interfaces.

In the second experiment, subjects were presented with 3 rings stacked in size order, largest ring on the bottom, next to a vertical rod. Their goal was to stack the 3 rings on a rod in order of size, with the largest ring on the bottom. They must use the robot's 2 grippers to do so, which they can manipulate physically. In addition, they used either the Amazon Echo for input and output (EchoBot), or the keyboard and monitor for input and output. After the experiment was finished, the subject completed a survey questionnaire. The purpose of this experiment is to evaluate the efficiency of completing the sequence using each of the 2 combinations of input and output interfaces.

We obtained approval for human research from the Office for Protection of Human Subjects (OPHS), under CPHS Protocol Number 2017-04-9796, with approval issued under University of California, Berkeley Federalwide Assurance #00006252.

5.2 Subject Population

All participants were 18 years or older and a registered undergraduate or graduate student at UC Berkeley. We did not perform any experiments with individuals of protected populations (children, elderly, prisoners, etc.). Our experiments included 21 participants in total.

5.3 Risks and Discomforts

Kinesthetic teaching involves human subjects moving the arms of the ABB YuMi robot through direct contact. This procedure risks minimal harm to the user because under kinesthetic teaching mode, the robot moves passively. This means that the robot will only move if it is being moved by the user, as it is not executing its own trajectories. Furthermore, the ABB YuMi is a human-safe robot [3]: it is wrapped in soft paddings to absorb impacts, has no pinch points, and can shut down its motors within milliseconds if a collision is detected.

Interacting with Echo may cause slight stress, because of unfamiliarity of conversing with a robotic speech interface. In the unlikely event that an unintended breach of confidentiality was to occur, magnitude of potential harm to study participants would be minimal. Knowledge of participation in our experiments has minimal potential harm to the subject.

We conducted post-experiment surveys, and data was collected anonymously and contained no personally-identifiable information. We did not collect names, pictures, videos, or audio of subjects. Participants also had the option to opt out of the experiment. In the unlikely event where anonymity is not secured, participation does not imply anything about the subject that would place him/her at risk of civil or criminal liability, or cause damage to their financial standing, employability, or reputation. The only identifiable information about participants would be that they are part of our subject population, i.e. students at UC Berkeley of at least 18 years of age.

6 Grasp Task

6.1 Study Setting

We evaluated EchoBot in a 2x2 study as an input and output interface to better understand the system's effectiveness in facilitating data collection for training a robot to grasp objects. The two input interfaces we compared were a keyboard button and EchoBot, and the two output interfaces were a monitor screen and EchoBot. The grasp task exemplifies a method to collect data for LfD robot learning. Given images of an object placed in various locations in the workspace and the poses of the robot to grasp those locations, LfD methods can be used to train the robot to grasp the object in unseen locations (see Figure 1 and Figure 3).

We used a factorial design to compare the four possible combinations of keyboard presses or EchoBot commands as input and a text-based monitor screen or EchoBot responses as output. Our subjects included 10 volunteers from our lab, who were randomly assigned to two of the four experimental conditions. Each condition had 5 subjects perform the experiment. For this task, we asked each subject to provide repeated grasp pose demonstrations on the YuMi robot. Each subject performed 10-minute experiments with two of the interfaces.

We measure the average durations of individual grasp trials and the percentage of failed grasps per condition.

6.2 Study Procedure

The experimenter provided each subject with the necessary information to perform the task using the I/O interface, including how to respond to potential errors that may occur. Subjects performed repeated trials for 10 minutes for each of two of the interface conditions. The order of the two conditions was randomized to mitigate learning effects. Subjects were given two minutes prior to each condition to be familiarized with the task and interface, to alleviate the effects of initial learning time.

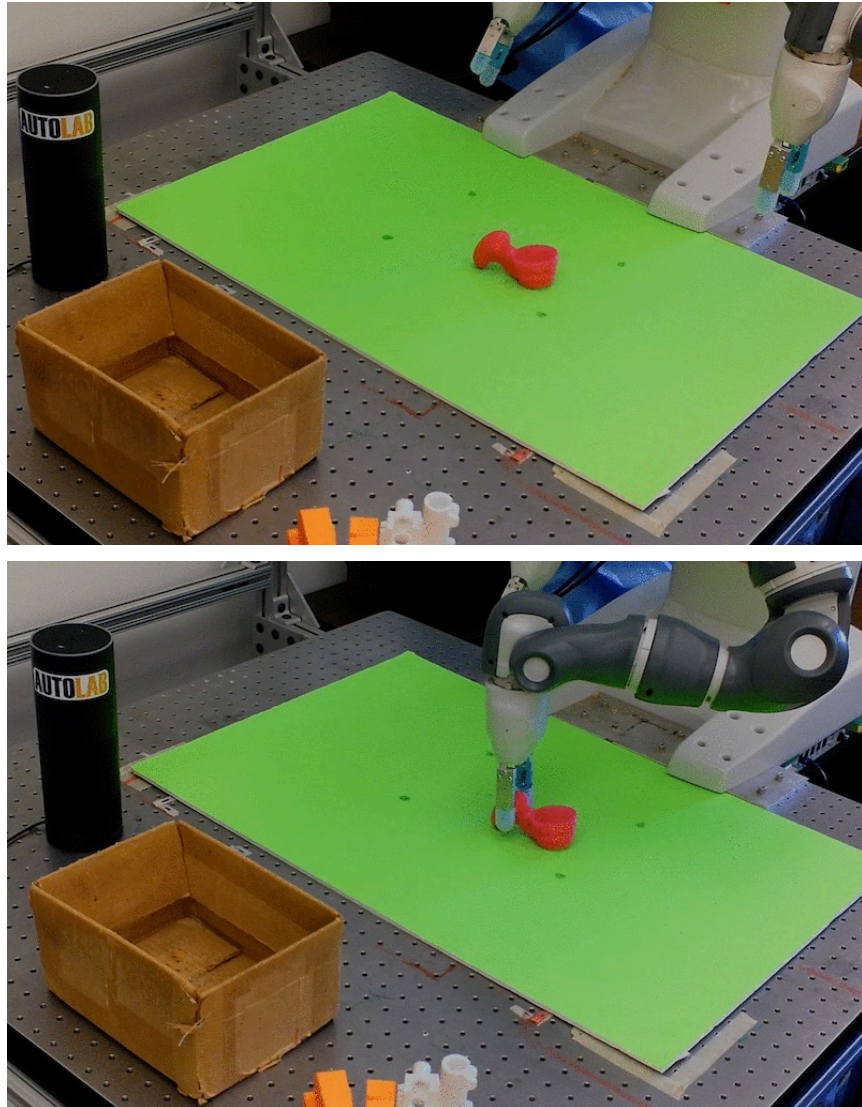


Figure 3: In the grasp task, users first place the object in a random pose in the workspace using the box and ask EchoBot to capture an image using the robot's overhead camera (*top*). Then, the user physically guides the robot's arm to a position where it is able to grasp the object, and asks EchoBot to record the robot arm's pose (*bottom*).

Subjects were asked to provide input as a means to record demonstrations of correct poses for robotic grasps of the same object. They either stated "Echo, tell YuMi to record" for the EchoBot interface, or pressed the "r" key for the keyboard interface. By using the "r" key instead of the "Enter" key, the two input methods were more similar in terms of cognitive memory load on the user. The subject provides this input periodically every 20-30 seconds. The output messages of the monitor and EchoBot were exactly the same, with the function of giving the subject information about successes and failures while recording grasp poses. When the system detected a failure, such as if the gripper was outside the workspace, the subject had to re-demonstrate the grasp based on feedback from the output. After the subject completed both conditions, the subject was administered a short questionnaire, loosely based on [20], to understand user opinions of the interfaces.

6.3 Results

To ascertain that initial learning effects did not cause subjects to increase in speed over time, we analyzed the durations of individual sample collections for each subject. We did not find a significant difference in these durations over time for any of the 4 interfaces.

We report the average durations of trials and number of failures per condition (see Figure 4), along with survey results. Each condition had between 74 and 110 total grasp trials across all participants. Using the keyboard-monitor as the input-output interface saw both more successes and failures, which means that this condition resulted in many more attempts at samples than the other conditions. However, survey results suggested that the Echo-speaker (EchoBot) and keyboard-speaker conditions were more intuitive and more enjoyable to use than the Echo-monitor and keyboard-monitor conditions, respectively. Although the most samples were collected with the keyboard-monitor interface, most users preferred to use an interface with either speech input or audio output.

From our survey responses, we also found that 40% of users found the keyboard-monitor interface harder to use than the keyboard-speaker interface, and 60% of users found the Echo-monitor interface harder to use than the Echo-speaker (EchoBot) interface. Moreover, users experienced both the most physical fatigue and mental fatigue in the keyboard-monitor condition, and the least of both kinds of fatigue in the Echo-speaker (EchoBot) condition. However, this finding is not statistically significant, and may be correlated with the fact that the former condition collected the highest number of data samples in the given 10-minute experimental period and the latter condition collected the smallest.

The difference in successful grasps may be explained by the length of speech input and output in comparison with pressing a button or reading a sentence on the monitor. We tried to optimize both our EchoBot and baseline implementations as much as possible for a fairer comparison. However, it takes a user longer to speak a 5-word command to the Echo than to press a key on average. Moreover, it was also observed that subjects would wait until EchoBot had finished its speech transmission before attempting the next trial, whereas they would immediately begin after the text on the monitor changed. The value of a speech interface may not have been apparent in this task, aside from user preferences, because the task did not require the user to engage both hands.

We also saw that some users experienced frustration at the Amazon Echo as an input device when it would not detect their voice command as expected. For certain people, such as those with quieter voices or those with voices close to the ambient frequency, the Echo would either not detect any human voice command at all, detect a completely different command, or detect a mixture between the user's voice and someone else's voice in the room. As a result, we do not think that EchoBot is ready to be used in a home, commercial, or factory setting as of now. Possible directions to improve the robustness and reliability of EchoBot would be to integrate noise cancellation and voice separation, or integrate a microphone or headset that may be used in place of speaking directly to the Amazon Echo device.

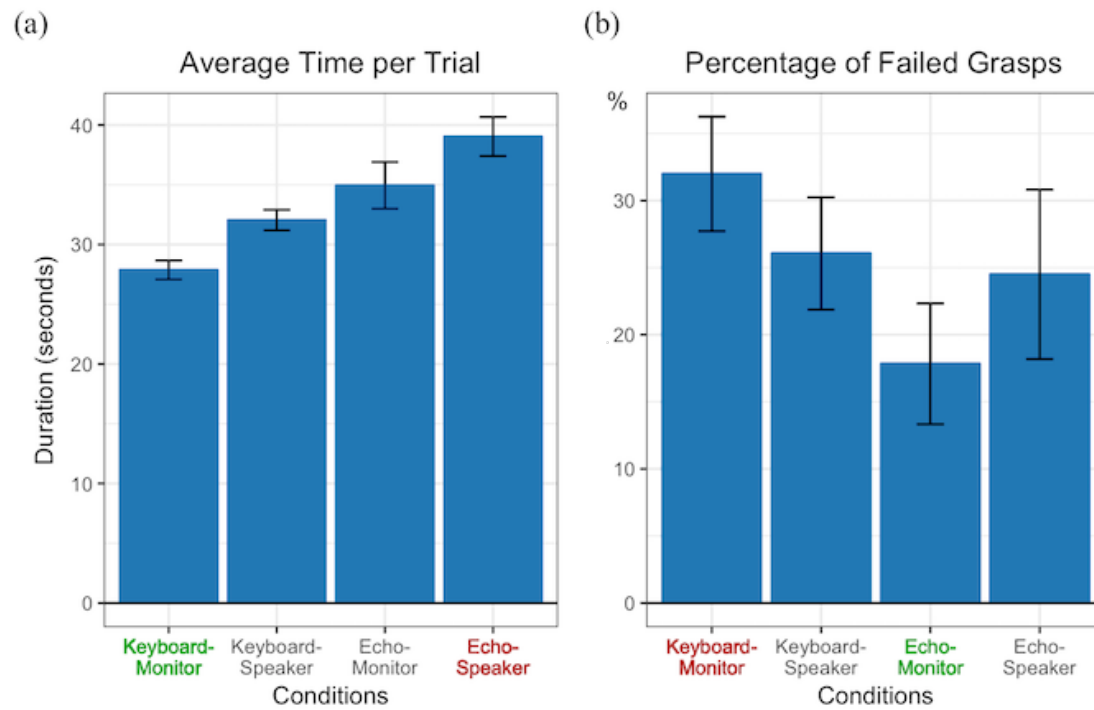


Figure 4: Results for the grasp task, with 5 subjects per condition. The best condition is highlighted in green, and the worst in red. (a) Average durations of trials per condition. (b) Percent of failed grasps per condition in 10-minute tasks.

6.4 Reflections and Enhancements

A common area of difficulty that we saw in the experiments was that after several of these repetitive iterations, the user would lose track of which step they were currently administering. We added several more features to EchoBot for this task to alleviate this problem:

1. Correlate short audio sound effects with the prompt before each step, for the trial.
2. Have longer, explanatory phrases in the first few trials, and shorten the phrases in the following iterations.
3. Switch between phrases used by EchoBot to communicate the same message to the user, to maintain the user's focus.
4. Use more humanizing phrases for EchoBot to connect with users.

As described above, each trial iteration consists of two steps: placing the object randomly in the workspace and capturing an image, and guiding the robot's arm to grasp the object and recording the arm's pose. For the first two iterations of these two steps, EchoBot will give formal instructions for each step, preceded by a sound effect. All sound effects have a duration of less than a second.

1. (sound effect #1) + "Step 1: Please place the object in the workspace and capture an image."
2. (sound effect #2) + "Step 2: Please guide my arm to the object and record my arm's pose."

For the next 2 iterations, the user may not need detailed instructions, so EchoBot would provide a shorter prompt. The motivation behind switching to shorter prompts is increase the efficiency of data collection and to not belabor the user with lengthy and repetitive instructions. This prompt would also be in the form of a rhyme, to help them get into the two-step rhythm and focus better. Trial iterations 3 and 4 have the following prompts:

1. (sound effect #1) + "Step 2 is through, now run step 1."
2. (sound effect #2) + "Step 1 is done, now do step 2."

By the 5th iteration of the 2 steps, we make the assumption that the user knows which sound effect correlates with which step, so the prompt is shortened to just the sound effect before each step. In the steady state, where the user has completed many sample collection iterations, adding the sound effects is just a slight increase in trial times. Moreover, utilizing sound effects may be beneficial to the user, and may even save redoing

work if users forget which step they are on. This hypothesis has yet to be evaluated with experiments.

We also improved EchoBot's phrases for failure modes of the experiment (see Table 1). In the case of a user error, EchoBot would have details on what exactly the human needs to fix. Each phrase is preceded by an "error" sound effect so the user can immediately distinguish error phrases from prompts or successes. For a given error, EchoBot randomly rotates between a set of phrases so that the user does not tire of hearing the same message. It also includes assurances and encouragement to the user when they make several mistakes, to alleviate any anxiety or pressure the user may experience in dealing with a speech interface.

Sample EchoBot Error Speech Output
Looks like the gripper needs to be rotated 180 degrees.
I think the gripper needs to be rotated 180 degrees.
Oh no! The gripper is too high. Let's try that again.
That's odd, I can't see the object anymore. Did you move it outside of the workspace?
That's ok! Let's try that again.
Don't worry, why don't we try that again?

Table 1: Examples of phrases EchoBot communicates to the user in the case of human errors. Each phrase is preceded by an "error" sound effect so the user can immediately distinguish error phrases from prompts or successes. EchoBot provides details on what actions the user needs to take, and assurances or encouragements to alleviate stress on the user.

7 Ring-Stacking Task

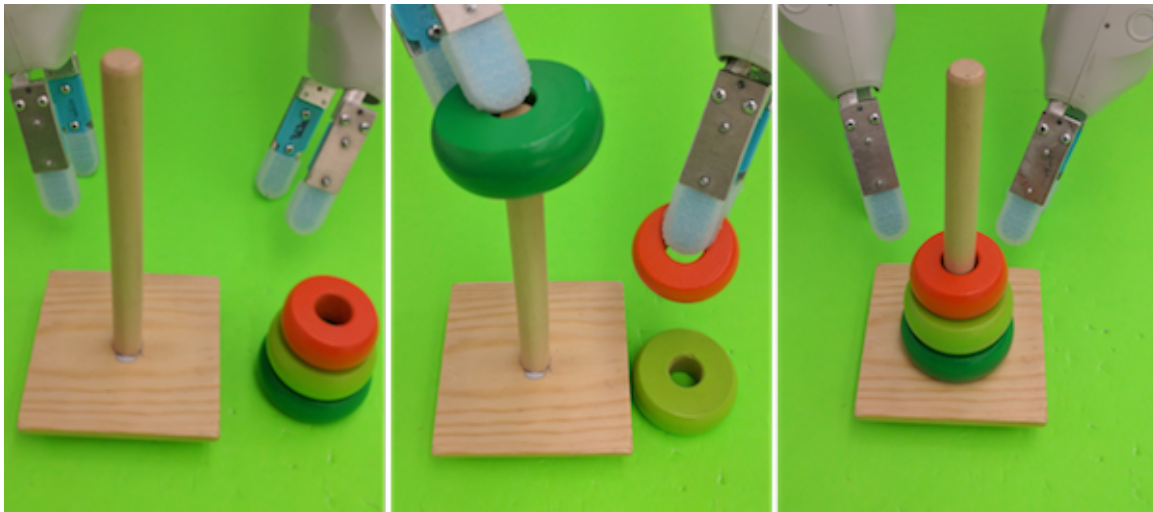


Figure 5: The ring-stacking task (left to right). Subjects guide the robot grippers to pick up the rings to stack onto the rod in size order.

7.1 Study Setting

Motivated by the findings of the grasping task, we designed a second task in which subjects provided full-trajectory, kinesthetic demonstrations of a ring stacking task with the YuMi robot. The task is similar to the Tower of Hanoi puzzle, where the objective is to move 3 rings from a pile to a rod in size order using both robot grippers (see Figure 5). During the demonstrations, the user was asked to record every instance of the grippers being open or closed. Our aim was to compare the EchoBot interface with the keyboard-monitor interface for a task where the collection of human input was more frequent and occurred at inconsistent time intervals, and demonstrations required the concurrent involvement of both hands. While we did not use the data collected in this experiment, this task is representative of other data collection methods that require constant input from a human. Frequent human input may be used in data collection for LfD methods in a dynamic

workspace, and concurrent use of both hands allows for a larger range of tasks with the robot.

We used a within-subjects design, assigning 11 UC Berkeley computer science student volunteers to both conditions, randomly perturbing the order of the conditions. We measured the time to complete each demonstration and the percentage of human errors in record commands for each condition.

7.2 Study Procedure

The experimenter provided each subject with the necessary information to perform the task using the I/O interface. The experimenter demonstrated a sequence to move the 3 rings from one pile to the rod and guided the subject to repeat the sequence twice prior to the experimental trials. This was to help the subject memorize the exact sequence of moves, and to reduce the effects of not knowing how to perform the task. During the trials, subjects repeated the same sequence and also indicated using the keyboard or EchoBot whenever either gripper opens or closes. This was done by saying "left opened", "left closed", "right opened", or "right closed" to EchoBot, or by typing "lo", "lc", "ro", or "rc" on the keyboard. If the subject made a mistake, the monitor or speaker informed the subject to restart the sequence before the trial can be successful, yet will continue to accept inputs. Thus, ignoring the output would cause the subject to perform unnecessary work. After the subject completed both conditions, the subject was administered a short questionnaire, loosely based on [20], to understand user opinions of the interfaces.

7.3 Results

Subjects using EchoBot were able to complete the ring-stacking task in less time than with the keyboard-monitor interface (see Figure 6). We observed a 57% decrease in average time to complete the task. The average ratio for a given participant between using the keyboard-monitor interface and EchoBot interface across all participants was 1.76 ± 0.4 . All subjects were able to complete the ring-stacking demonstration with EchoBot in equal or less time than with the keyboard-monitor interface. In addition, subjects committed fewer errors with EchoBot than with the keyboard-monitor.

While the confidence intervals in Figure 6 show high variance, this is mainly a result of how the data has been presented. Because all of the subjects that we used in this experiment had limited experience with a robot, some users were inherently faster or slower than others. This means that while the difference in durations between the 2 conditions for a given subject were apparent, averaging these durations across all participants discards the relative improvement of the EchoBot condition versus the keyboard-monitor condition. A more informative metric would be the average ratio of durations across participants, reported above as 1.76 ± 0.4 .

Survey results indicate that subjects found EchoBot to be more intuitive and enjoyable than the keyboard-monitor interface. They also felt more efficient with EchoBot than with the keyboard-monitor. Subjects reported that neither interface was harder to use than the other.

Even though EchoBot outperformed the keyboard-monitor interface according to many metrics, there are still areas where it can be improved. Some subjects indicated that the 5-second listening timeout on the Amazon Echo was too short, and it was inconvenient to reactivate the Echo whenever they were unable to issue the next command within that time limit. EchoBot performed much better in this task than in the grasp task because this task required repeated input from the user every 5 seconds, and because it occupied both of the subject's hands. Frequent input, as opposed to every 20-30 seconds as in the previous experiment, allowed the user to reduce the size of the human speech command to the Echo by 4 words because the Echo can continue listening for input up to 5 seconds after a human speech command, a significant improvement for user interactivity. Moreover, we found that in the keyboard-monitor condition, subjects would often first speak the command (e.g. "left closed", "right opened"), think about which buttons to press, and then type the correct input, even if they had not yet experienced the EchoBot condition.

For some of the subjects that indicated they had no prior experience with the Amazon Echo or any other voice interface system, they had difficulty knowing when to pause and when to speak to the Amazon Echo. Some participants appeared to feel personally responsible when the Echo did not understand their command, and had to be reassured that it was not their fault. For this reason, we tried to expand on the vocabulary of EchoBot in an attempt to humanize the robot and make users feel more comfortable, discussed in the following subsection.

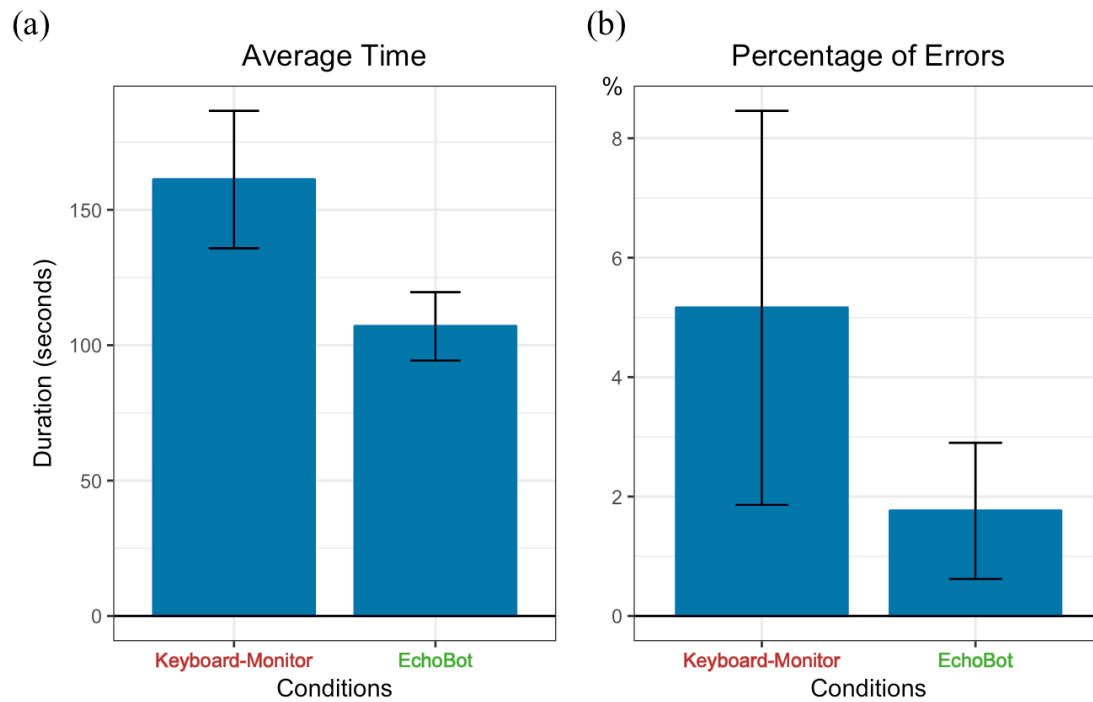


Figure 6: Results from the ring-stacking experiment, with 11 trials per condition. The best condition is highlighted in green, and the worst in red. (a) Average durations of experiments across conditions. (b) Percentage of human errors across conditions.

7.4 Reflections and Enhancements

We noticed in our experiments that some users had initial difficulty knowing when to pause and when to speak to the Amazon Echo, when they were relaying successive commands to EchoBot. This was especially apparent in users with no prior experience with the Amazon Echo. We had incorrectly assumed that the flashing LED lights intrinsic to the Amazon Echo would be enough to cue the user for the next command. We added several more features to EchoBot for this task to alleviate this problem:

1. Use a sound effect as a confirmation that EchoBot had processed a human command and is ready for the next command.
2. Use a sound effect in addition to the error message when the user makes a mistake inputting the sequence of commands.
3. Switch between phrases used by EchoBot to communicate the same message to the user, to maintain the user's focus.
4. Use more humanizing phrases for EchoBot to connect with users.

When the user starts or restarts the task, EchoBot will say a random phrase out of the following:

- Hello, friend! Let's get started.
- Howdy, partner! Ready when you are.
- Nice to see you!

Then, the user begins inputting the sequence as they perform the sequence using the robot's grippers. Every time they input a step of the sequence into EchoBot correctly, such as "left closed" or "right opened", a short sound effect will be played as a confirmation that the Echo received the message. The sound effects we used in EchoBot all have a duration of less than a second. We have seen from our experiments that there have been about 10 times as many successes as failures in the experiments. Thus, playing a sound effect for confirmation will be more efficient than a speech phrase and no less efficient than simply the flashing LED lights on the Amazon Echo, in terms of total time for the user to complete the sequence.

If the user makes a mistake in inputting the sequence, EchoBot will play a sound effect corresponding to an error, and then prompt the user to restart the sequence:

- Hmm, that's not quite right. Let's reset the sequence and try that again.
- Hmm, that's not what I expected. Let's restart the demonstration.

In the case of human errors, we can afford to have a slightly longer phrase for descriptive and casual speech because errors are not very common. We also considered informing the user specifically what they did wrong and then prompt for a restart, although we found in the experiments that users often knew exactly what mistake was made in the sequence.

After the entire ring-stacking task is completed, EchoBot plays a "success" sound effect, and says a phrase of encouragement chosen at random from a set of phrases. (See Table 2 for a subset of the phrases.)

Sample EchoBot Success Speech Output
You figured it out!
All done! You're really good at this!
Well done! You made it look easy!
Congratulations! You've finished the experiment.
Well aren't you a problem solver! Good work!

Table 2: Examples of phrases EchoBot says after the user successfully completes the ring-stacking sequence. EchoBot randomly chooses one of the phrases of encouragement to speak to the user. Each phrase is preceded by a "success" sound effect.

8 Conclusion

We present EchoBot, a system to facilitate data collection for LfD with the Amazon Echo. EchoBot enables users to record robot demonstrations using a speech interface. EchoBot utilizes the speech processing of the Amazon Echo to communicate with the ABB YuMi industrial robot. We implemented a local server to dispatch human speech commands from the Echo to either a local speaker audio stream or the robot program. The audio stream notifies the user in real-time of explanations of the robot actions and feedback while the user is collecting data to train robots using Learning from Demonstration (LfD). We evaluated the use of EchoBot as both an input and output interface in grasping and ring-stacking tasks. Experiments suggest that EchoBot can be more efficient when inputs are frequent and both hands of the user are occupied.

This paper presents an initial experimental study of the effects of using a voice interface to collect data on a robot. Limitations of our work include:

- Predefining human phrase inputs to the Amazon Echo, which constrains user-robot interaction.
- Robustness to multiple voices speaking at once.
- Length of human phrases to relay infrequent commands to EchoBot.
- Short timeout duration of Echo inputs for frequent commands.

We speculate that decreasing the length of human phrase inputs will increase user efficiency for data collection. We also speculate that including more humanizing phrases spoken by EchoBot will enhance the likability of the robot, and we look forward to addressing these issues in future work.

9 Appendix

9.1 EchoBot Setup

To run the local server for EchoBot, clone our GitHub repository (see Footnote 1). Follow the instructions outlined in the file `README.md` to install the python package dependencies using the `pip` installer, and adding your Amazon Alexa App ID to your Unix environment. You will receive an App ID after creating a new Alexa application on Amazon's developer website.

To run the Django server locally and access the server externally, run the main script:

```
$ ./main
```

The external URL will be in the form of `https://*.ngrok.io`. If you want the external URL to be constant throughout restarts, upgrade to the paid `ngrok` plan, and replace the last command in `main.sh` with:

```
$ ./ngrok-linux-64 http -subdomain=your_subdomain 8000
```

Then, log into the Alexa Skills Kit online developer portal, and copy your server's `ngrok` URL to the appropriate field in the configuration settings. You will also have to define custom utterances, or human speech phrases, to trigger a given command on the server.

9.2 Using EchoBot

Logging a message to the local server database from an external Python script is straightforward. First, copy the file `robot_logger.py` into your workspace. Then add the following lines of code into your external Python script:


```
>>> import robot_logger
>>> robot_logger.log("Message")
```

It is also possible to retrieve messages from EchoBot, such as when the user issues a command such as “take a picture” or “record the pose” to the Amazon Echo for data collection. After a user starts the data collection interface as described above, have your code call the following function. Then whenever you say a command, it will be accessible via that function. Currently supported return values are `None`, “start”, “record”, “stop”, “pause”, and “finish”. This will not block execution.

Start by saying “Echo, tell YuMi to ...” and then your first command. Subsequent commands can be one word. Say “Quit” after you are done to stop data collection mode.

```
>>> import robot_logger as r
    "Echo, tell YuMi to record."
>>> print r.getDataCommand()
Record
>>> print r.getDataCommand()
None
    "Stop."
>>> print r.getDataCommand()
Record
    "Record."
>>> print r.getDataCommand()
None
    "Quit."
```

For the interaction task, where a user specifies to EchoBot which object to grasp by color, there are 2 possible methods to use, depending on if you want to retrieve objects one at a time or all at once. To grab the colors after a user has asked the Echo to grasp them, you can use either of the next two functions in `robot_logger.py`: `getSingleGraspCommand()` or `getGraspCommands()`. Here is sample usage of those functions:

```
>>> import robot_logger as r
    "Echo, tell YuMi that I want the orange, yellow, red, and blue parts."
>>> print r.getSingleGraspCommand()
bar_clamp
>>> print r.getSingleGraspCommand()
pawn
>>> print r.getSingleGraspCommand()
vase
>>> print r.getSingleGraspCommand()
nozzle
```

```
>>> print r.getSingleGraspCommand()
None
```

```
    “Echo, tell YuMi that I want the orange, yellow, red, gold, and blue parts.”
>>> print r.getGraspCommands()
['bar_clamp', 'pawn', 'vase', 'part1', 'nozzle']
```

Here is a subset of the list of additional human phrases we used for EchoBot:

Starting Audio Stream:

"Echo, ask YuMi to explain what it is doing."
"Echo, tell YuMi to speak."

Ending Audio Stream:

"Echo, tell YuMi to end the audio stream."
"Echo, tell YuMi to end stream."

Help with Camera Calibration:

“Echo, ask YuMi how do I calibrate the camera?”
“Echo, ask YuMi how to calibrate the camera.”

To advance to the next step, say “continue”, “next”, or “what next?”.

Grasping Mode:

The {param*} below represent one of the following 8 colors: orange, red, white, yellow, gold, black, pink, blue.

All:

“Echo, tell YuMi that I want all parts.”
“Echo, tell YuMi that I want all the parts.”
“Echo, tell YuMi that I want all of the parts.”

One:

“Echo, tell YuMi that I want the {paramOne} part.”

“Echo, tell YuMi that I want the {paramOne} parts.”

Two:

“Echo, tell YuMi that I want the {paramOne} and {paramTwo} parts.”

Three:

“Echo, tell YuMi that I want the {paramOne}, {paramTwo}, and {paramThree} parts.”

“Echo, launch YuMi.” (wait for the response “Welcome.”) “I want the {paramOne}, {paramTwo}, and {paramThree} parts.”

9.3 Extending EchoBot

The code for our local EchoBot server is available on GitHub (see Footnote 1). To add a new command to EchoBot, create a new function in `RobotThoughtApp/alexa.py`, consistent with the other defined functions. Your new function should also have an `@intent` decorator to be recognized by the framework as an Alexa intent. If your command requires the use of parameters, you also should define Slots, which are how the Alexa framework defines their parameter names and types. See the other Slots in that file for example usage.

You must also define the names of your new intents in a different file for the Alexa framework to recognize them. In `django_alexa/internal/intents_schema.py`, append your new intent to the end of the `MY_INTENTS` list, near the top of the file.

Bibliography

- [1] A. B. B. (ABB). (2017, January). [Online]. Available: <http://new.abb.com/products/robotics/industrial-robots/yumi>
- [2] B. Akgun, M. Cakmak, J. W. Yoo, and A. L. Thomaz, “Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective,” in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*. ACM, 2012, pp. 391–398.
- [3] Amazon. Alexa skills kit. [Online]. Available: <https://developer.amazon.com/alexa-skills-kit>
- [4] ——. Amazon echo. [Online]. Available: <https://www.amazon.com/echo>
- [5] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, “A survey of robot learning from demonstration,” *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.
- [6] J. M. Beer, A. Prakash, T. L. Mitzner, and W. A. Rogers, “Understanding robot acceptance,” *Georgia Institute of Technology*, pp. 1–45, 2011.
- [7] R. Cantrell, P. Schermerhorn, and M. Scheutz, “Learning actions from human-robot dialogues,” in *RO-MAN, 2011 IEEE*. IEEE, 2011, pp. 125–130.
- [8] E. Cha, A. D. Dragan, and S. S. Srinivasa, “Perceived robot capability,” in *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2015, pp. 541–548.
- [9] F. D. Davis, “Perceived usefulness, perceived ease of use, and user acceptance of information technology,” *MIS quarterly*, pp. 319–340, 1989.
- [10] L. A. Dean and H. S. Xu, “Voice control of da vinci,” May 2011, voice integration with the da Vinci surgical robot, Johns Hopkins University. [Online]. Available: <https://ciis.lcsr.jhu.edu/dokuwiki/lib/exe/fetch.php?media=courses:446:2011:446-2011-6:finalreport-team6.pdf>
- [11] M. Desai, P. Kaniarasu, M. Medvedev, A. Steinfeld, and H. Yanco, “Impact of robot failures and feedback on real-time trust,” in *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*. IEEE Press, 2013, pp. 251–258.
- [12] ——, “Impact of robot failures and feedback on real-time trust,” in *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*. IEEE Press, 2013, pp. 251–258.
- [13] M. Desai, M. Medvedev, M. Vázquez, S. McSheehy, S. Gadea-Omelchenko, C. Bruggeman, A. Steinfeld, and H. Yanco, “Effects of changing reliability on trust of

- robot systems,” in *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*. IEEE, 2012, pp. 73–80.
- [14] M. Desai, K. Stubbs, A. Steinfeld, and H. Yanco, “Creating trustworthy robots: Lessons and inspirations from automated systems,” *Proceedings of the AISB Convention: New Frontiers in Human-Robot Interaction*, 2009.
- [15] B. R. Duffy, “Anthropomorphism and the social robot,” *Robotics and autonomous systems*, vol. 42, no. 3, pp. 177–190, 2003.
- [16] T. Fong, C. Thorpe, and C. Baur, “Collaboration, dialogue, human-robot interaction,” in *Robotics Research*. Springer, 2003, pp. 255–266.
- [17] S. R. Fussell, S. Kiesler, L. D. Setlock, and V. Yew, “How people anthropomorphize robots,” in *Human-Robot Interaction (HRI), 2008 3rd ACM/IEEE International Conference on*. IEEE, 2008, pp. 145–152.
- [18] L. Gallardo-Estrella and A. Poncela, “Human/robot interface for voice teleoperation of a robotic platform,” in *International Work-Conference on Artificial Neural Networks*. Springer, 2011, pp. 240–247.
- [19] Z. Henkel, V. Srinivasan, R. R. Murphy, V. Groom, and C. Nass, “A toolkit for exploring the role of voice in human-robot interaction,” in *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2011, pp. 255–256.
- [20] G. Hoffman, “Evaluating fluency in human-robot collaboration,” in *International conference on human-robot interaction (HRI), workshop on human robot collaboration*, vol. 381, 2013, pp. 1–8.
- [21] H. Jiang, Z. Han, P. Scucces, S. Robidoux, and Y. Sun, “Voice-activated environmental control system for persons with disabilities,” in *Proceedings of the IEEE 26th Annual Northeast Bioengineering Conference (Cat. No.00CH37114)*, 2000, pp. 167–168.
- [22] T. Kanda, T. Hirano, D. Eaton, and H. Ishiguro, “Interactive robots as social partners and peer tutors for children: A field trial,” *Human-Computer Interaction*, vol. 19, no. 1, pp. 61–84, June 2004. [Online]. Available: http://dx.doi.org/10.1207/s15327051hci1901&2_4
- [23] S. Kiesler and J. Goetz, “Mental models of robotic assistants,” in *CHI’02 extended abstracts on Human Factors in Computing Systems*. ACM, 2002, pp. 576–577.
- [24] T. Kollar, S. Tellex, D. Roy, and N. Roy, “Toward understanding natural language directions,” in *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2010, pp. 259–266.
- [25] M. Laskey, C. Chuck, J. Lee, J. Mahler, S. Krishnan, K. Jamieson, A. Dragan, and K. Goldberg, “Comparing human-centric and robot-centric sampling for robot deep learning from demonstrations,” 10 2016. [Online]. Available: <https://arxiv.org/abs/1610.00850>

- [26] C. E. Lathan and S. Malley, "Development of a new robotic interface for telerehabilitation," in *Proceedings of the 2001 EC/NSF Workshop on Universal Accessibility of Ubiquitous Computing: Providing for the Elderly*, ser. WUAUC'01. New York, NY, USA: ACM, 2001, pp. 80–83. [Online]. Available: <http://doi.acm.org/10.1145/564526.564548>
- [27] M. K. Lee, S. Kielser, J. Forlizzi, S. Srinivasa, and P. Rybski, "Gracefully mitigating breakdowns in robotic services," in *Proceedings of the 5th ACM/IEEE international conference on Human-robot interaction*. IEEE Press, 2010, pp. 203–210.
- [28] M. K. Lee and M. Makatchev, "How do people talk with a robot?: an analysis of human-robot dialogues in the real world," in *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2009, pp. 3769–3774.
- [29] H. Medicherla and A. Sekmen, "Human-robot interaction via voice-controllable intelligent user interface," *Robotica*, vol. 25, no. 05, pp. 521–527, 2007.
- [30] S. Nikolaidis, S. Nath, A. D. Procaccia, and S. Srinivasa, "Game-theoretic modeling of human adaptation in human-robot collaboration," *IEEE Transactions on Robotics*, January 2017. [Online]. Available: <https://arxiv.org/abs/1701.07790>
- [31] P. Parente. Pyttsx. [Online]. Available: <https://pypi.python.org/pypi/pyttsx>
- [32] C. Ray, F. Mondada, and R. Siegwart, "What do people expect from robots?" in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2008*. IEEE, 2008, pp. 3816–3821.
- [33] B. Robins, K. Dautenhahn, R. Te Boekhorst, and A. Billard, "Effects of repeated exposure to a humanoid robot on children with autism," in *Designing a more inclusive world*. Springer, 2004, pp. 225–236.
- [34] J. M. Sackier, C. Wooters, L. Jacobs, A. Halverson, D. Uecker, and Y. Wang, "Voice activation of a surgical robotic assistant," *The American Journal of Surgery*, vol. 174, no. 4, pp. 406–409, January 1997. [Online]. Available: [http://dx.doi.org/10.1016/S0002-9610\(97\)00128-1](http://dx.doi.org/10.1016/S0002-9610(97)00128-1)
- [35] D. Spiliotopoulos, I. Androutsopoulos, and C. D. Spyropoulos, "Human-robot interaction based on spoken natural language dialogue," in *Proceedings of the European workshop on service and humanoid robots, 2001*, pp. 25–27.
- [36] S. S. Srinivasa, D. Ferguson, C. J. Helfrich, D. Berenson, A. Collet, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, and M. V. Weghe, "Herb: a home exploring robotic butler," *Autonomous Robots*, vol. 28, no. 1, pp. 5–20, 2010.
- [37] 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*. ACM, 2011, pp. 69–76.

- [38] S. A. Tellex, T. F. Kollar, S. R. Dickerson, M. R. Walter, A. Banerjee, S. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," *In Proceedings of the National Conference on Artificial Intelligence (AAAI 2011)*, 2011.
- [39] Google, Inc. Google Text to Speech API. [Online]. Available: <https://pypi.python.org/pypi/gTTS>
- [40] SoftBank Robotics. Pepper, the humanoid robot from Aldebaran, a genuine companion. [Online]. Available: <https://www.ald.softbankrobotics.com/en/cool-robots/pepper>
- [41] A. R. Wagoner and E. T. Matson, "A robust human-robot communication system using natural language for harms," *Procedia Computer Science*, vol. 56, pp. 119–126, 2015.
- [42] B. Wang, Z. Li, and N. Ding, "Speech control of a teleoperated mobile humanoid robot," in *2011 IEEE International Conference on Automation and Logistics (ICAL)*. IEEE, 2011, pp. 339–344.
- [43] A. Weiss, J. Igelsbock, S. Calinon, A. Billard, and M. Tscheligi, "Teaching a humanoid: A user study on learning by demonstration with hoap-3," in *Robot and Human Interactive Communication (RO-MAN)*, 2009. 147–152.