Voice Activation Control with Digital Assistant for Humanoid Robot Torso

Conor Wallace
conorw8@gmail.com
Department of Electrical and Computer Engineering
University of Texas at San Antonio

Berat A. Erol, PhD Candidate
baerol@gmail.com
Department of Electrical and Computer Engineering
University of Texas at San Antonio

***Abstract*—Digital Voice Assistants are an emerging technology due to improvements on mobile communication and computing technologies and are becoming more popular in recent years due to growing marketing strategies on new smart home devices by cloud service providers. Most of the applications on smart environments and assistive robotics are relying on these digital assistants, or Internet of Things (IoT) devices, based on the nature of the system, namely voice activation and control. Smart home assistants, such as Amazon Echo with Alexa and Google Home are the most well-known examples on this manner, relying on the base of processing verbal requests by looking for key words in the conversation as well as the structure of a natural language source of communication. Later, these key words are used to trigger the predefined skills for fulfilling the users' request. A simple structure of the process works both ways between the user and the device by providing a verbal request/order and verbal feedback. In this project, we have built a system, a humanoid robot torso and an IoT device, to control and simulate the process of an interactive functioning assistive robot and tried to improve the effects of Human Robot Interactions.**

Keywords: *Electrical Engineering, Robotics, Artificial Intelligence, Human Robot Interaction, Internet of Things.*

I. INTRODUCTION

This paper outlines the design process for a humanoid robot as well as providing examples of design failure and design success. The purpose of this project was to design the behavior of a robot to study human robotic interactions and the possible application thereof. The paper will delve into the initial problem set, the tools required for this project, the setup required for this project, the design process, the simulation and implementation, problems with the project as well as solutions to those problems, and finally the design drawbacks.

Human robot interaction (HRI) is the study of behavior between humans and robots. There are two essential goals of this field of research which are to improve robot technology and to maintain moral integrity in doing so. Robots have been in mundane factory positions for decades, however, with the rapid improvement of computing power and other necessary technologies robots have been placed in more advanced fields such as bomb-defusal, search and rescue, health care, and law enforcement. It is the study of human robotic interaction that assures both the furthering improvement in these technologies as well as their social competency.

Artificial intelligence can be defined as a machine analyzing its environment and having the ability to identify a best-case plan to achieve a desired goal. With the introduction of smart devices (i.e. smart phones, smart homes, etc.), a new form of interaction has been born. Now there is a deeper interaction between humans and machines aside from simply flipping a switch. There are behavioral elements to be understood for improved user experiences. Smart devices make this otherwise monumental breakthrough, seem ordinary. This is because these devices are now such an important part of our lives; therefore, the interaction with Artificial Intelligence (AI) feels more natural. AI entities such as Amazon's Alexa can add an even more complicated interactive element to any HRI focused studies.

Recent implementations for digital assistants on smart environments in the literature are limited, and few dedicated projects combining it with assistive robotics in HRI scope. The literature offers several metrics to evaluate the human in the loop scenarios based on physical interactions, tele-operating or actively controlling the robots in [1] and [2]. Some suggest new approaches on autonomy for evaluating the efficiency of HRI [3], [4], and [5]. Smart home applications and networked devices, such as smart thermostats, and home

automation tools, are even forcing these boundaries harder. [6] presents an extensive review on smart home applications and devices with robotic applications. For control system aspects, a digital assistant device that controls a manipulator robot framework is proposed in [7] and [8]. For human behavior focus, [9] touches how these devices are becoming so important in our social interactions. [10] proposes how these devices even help to improve our language learning skills. However, due to unexpected popularity on digital assistant devices, affordable home and assistive robotics applications, these metrics are in need of expanding their domains.

The rest of the paper is structured as following, Section II states the foundation of this paper and explain its components. The system setup and preliminary preparations are explained in Section III, followed by the implementation of the proposed system in Section IV. Then, we will be concluding for both hardware and software experiments along with the simulation results in Section V.

II. PROBLEM STATEMENT

There are two major components of this project that pose problems requiring solutions:

A. RowdyBot: A Humanoid Robot Torso

During this project timeline, we have been working on an open-sourced project for low cost humanoid platform called Poppy [11]. This robot, as the name suggests, is a robotic implementation of the upper torso of the human body. It is comprised of several 3D-printed segments that are attached via Dynamixel servo motors that act as joints, as shown in Figure 1. The collection of these components works together to create the functionality of the human body. It has all the same limitations of a human body meaning that the joints can only turn between positive ninety degrees and negative ninety degrees. This means that each segment of the torso can only move in the fashion its human counterpart is able to. The movements are done by first releasing the motors from its compliant position, meaning that the motors are stable and are unable to be moved my simply moving them with one's hands, then sending the robot values between positive ninety degrees and negative ninety degrees. By combining these movements of each of the motors we are able to mimic human behavior.

The problem here is figuring out which motors are necessary for each human behavior we want to mimic, as well as the values to send to these motors, as well as the time we want each of these actions to take place in.

B. Alexa Skill: Voice Interaction and Activation

Alexa is a digital voice assistant introduced by Amazon and integrated with Amazon Echo devices. It is an artificial intelligence software that allows its users to vocally interact with various services and systems. Alexa functions via Alexa Skills which are essentially like web based apps that can be activated by vocation. This capability is particularly useful for this project as it helps to improve human-robotic interaction to a more sentient interaction. This can be done by way of a few simple lines of code and proper software capabilities. Namely, network tunneling, which is used to connect and convert the code in the Amazon data base to the local network being used by the computer controlling the RowdyBot. This will further be explained in the following section.

III. PREPARING THE TEST-BED

A. System Setup

Before starting the coding stage for controlling the system, the development environment must be constructed. We mostly used the Python programming language regarding the humanoid robot torso and digital assistant back-end. We will need at least Python 2.7 or greater for the compatibility of the native Linux environment. Once the base language package installed, many post-packages will be required eventually, such as numpy, scipy, notebook, jupyter, and matplotlib packages and libraries for the Python environment.

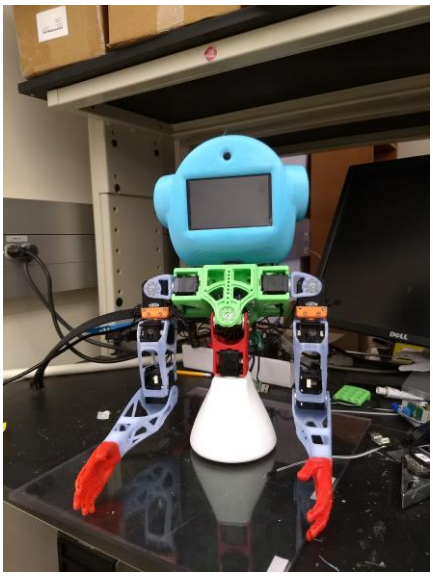

Figure 1. Humanoid robot torso- RowdyBot is built by using 3D printing technology. It has 13 degree of freedom that gives the robot the ability to mimic upper body movements A touchscreen LCD monitor has added onto the forehead to improve HRI.

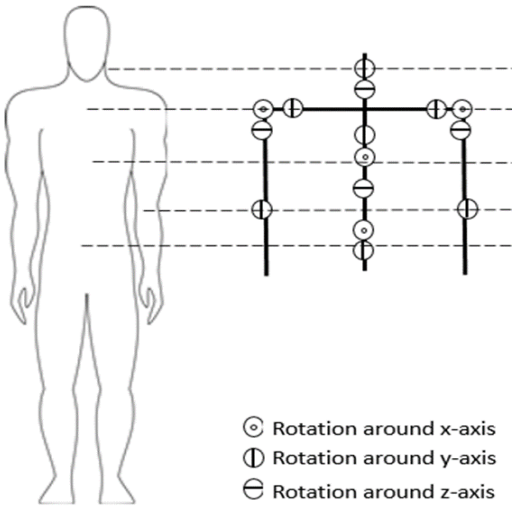

Figure 2. A representation for the humanoid robot torso based on the human upper body and its joints.

Then, the default open-source software has to be installed, which contains any functions required to control the robot including, motor calls, timing commands etc. These actions can be done via command line instructions which can be found in the open-source documentation [11]. In the case for this project, the bulk of the work was done using the VREP simulation environment, a robotics simulator as opposed to the actual robot as an object [12].

The next stage was preparing the environment for developing an Alexa Skill, which was proposed as the main tool to control the RowdyBot. This can be done several ways, but there is one thing that is absolutely required and that is an Amazon Developer account from Amazon Web Services (AWS) [13]. It is free to everyone if the development is for non-profit individuals and/or educational reasons. In the AWS environment, one can develop all the backend and user interface for the Amazon Echo Devices [15], a skill that is required so that the digital assistant (Alexa) responds appropriately. From here another package needs to be installed, Flask. This package streamlines the interaction between the Skill and the Python script so that less backend programming is required [16]. Finally, a network-tunneling software named Ngrok will need to be used [17]. It is a very efficient approach to connect the Skill running on Amazon's servers to the local computer that is connected to the robot. This package is the most essential piece of the setup as the project would not function without it. After having all packages installed and the software is running properly, the development of the testbed environment can begin.

B. Robotic Controls

The servo motors that link the human limbs and mimics the joints control the movement of the robot, and can only move between the -90 and +90 degrees, and the directions as shown in Figure 2. With the combination of movements of these motors, human movement can be mimicked. The process for moving a motor is as follows:

- turn motor compliance off so it is stiff
- call a function to rotate the motor to a certain degree
- wait until the motor has completed its movement

The motors' compliance is a feature that, when activated, allows the motor to be freely moved by hand. To activate the motor, the compliance must be turned off. To activate the motor, a function must be called. There are several different functions that can be used, but the two implemented in this project were:

- motor.goal_position = x
- motor:goto_position(x, t, wait=True)

In the two functions, motor is referring to the particular motor that is being controlled. In the first function, x is the angle (-90 to +90 degrees) the motor needs to travel to. In the second function, x is also referring to the angle, t is referring to the time to wait for, and wait=True is a command to wait until time t has expired. Using these two functions, we can operate each of the motors in tandem to create human body movement. To make sure the motors complete their respective movements; a certain amount of time must be allowed to pass. In the second function, this characteristic is already in place. For the first function, the sleep function must be called after each movement, including tandem movements.

- time.sleep(t)

Time, in this function, is the library that should be included in the script and t is referring to the amount of time that should pass.

C. RowdyBot Humanoid Torso Alexa Skill

The Alexa Skill is the voice application used to interact with the robot as well as what is driving the Python script which controls the robot. The Skill is largely comprised of backend code that is mainly generated automatically by Amazon's servers. This backend code is responsible for handling voice input and response, as well as keeping track of the state of the session. A session is the name for the time from when the Skill is opened to when it is closed. A session is only

relevant in that time frame, meaning that each session is uniquely generated upon start up each time the Skill is invocated.

An Alexa Skill has a specific code structure with general elements. These elements include an invocation name, an intent name, and a sample utterance. The invocation name is the phrase that, when heard, triggers the Python script. The invocation name is the root name for the entire Skill, much like the name of a mobile application. The intent name is the argument that tells the invocation name what to do. It is a function inside of the Python script that will be specifically triggered when one of its sample utterances are heard. Sample utterances are possible phrases that are routed to a specific intent name. These phrases will be stated along with the invocation name when addressing Alexa. The basic structure of an Alexa phrase is to first address Alexa, then the invocation name to trigger the Python script, then a sample utterance so the script knows which intent to trigger.

For this project we used a program called Flask that uses Python code to streamline the connection between the backend code generated by Amazon's servers and the Python script. This program has two libraries that are very useful for this project named statement and question. Statement takes in a string of text that is sent to the Amazon servers and then spoken by Alexa. This library, when triggered, automatically ends the current session because it assumes that there is no further interaction to be had. Question takes in a string of text as well and does the same thing as statement except that instead of ending the session, it keeps the session open so that further interaction can be had. Using question, we can write the Python script so that the user can have a conversation with the robot as opposed to Alexa and increase the interaction value.

D. Network Tunneling

To link the backend code and the interface for the Alexa Skill to the Python script, requires one of two methods:

- Lambda Function
- Tunneling Program

The latter will be used in this project in the form of an Ngrok platform. Ngrok is a tunneling program that creates a blank https address intended to make a path from the Amazon Web Services servers to the Python script being run on a local IP address, described by a simple system flow diagram shown in Figure 3.

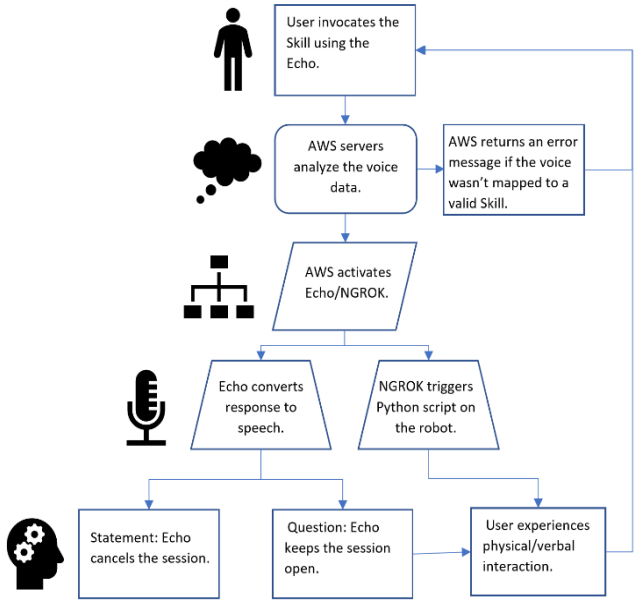

Figure 3. The system flow diagram based on the network tunneling. Note that the Alexa Skill is called first, then, the Skill, if it is available in the AWS data base, is matched with the corresponding controller to trigger the system component. Finally, the Skill passes through the network-tunneling to move the RowdyBot to fulfill the user's requests.

This is done by writing data to a blank port number (i.e. 192. 254. XXXX). This way when something triggers in the Python script, the data is immediately written to the blank port and then to the blank port and then to the AWS servers.

IV. IMPLEMENTATION

A. RowdyBot Humanoid Torso

As stated in previous sections, the RowdyBot is a humanoid torso robotics platform based upon the open-source Poppy Project [11]. The idea is to build simple human body movement and behavior being triggered by conversation with its own Alexa Skill. This is done entirely in a local Python script that will be executed when the Skill is invocated. In this script, several libraries are imported to help simplify the code. The two most notable libraries are the open-sourced library, which contain the functions used for changing motors and so on, and Flask-Ask, which is a program that streamlines the backend integration for the Alexa Skill. With these two libraries in place, behavior can be implemented. There are several functions implemented for this project including, but not limited to:

- RowdyLeftArm():

- RowdyInit():
- RowdyHello():
- RowdyHowAreYou():
- RowdyWeather():

Each one of these functions has a specific behavioral attribute and verbal response. For explanatory reasons, we will examine the say hi function. This function makes use of four motors to create its behavior. The behavior for this function is simply a right-hand wave. The motors in use are:

- r_elbow_y
- r_arm_z
- r_shoulder_y
- r_shoulder_x

First, the time.sleep function is called to prepare the RowdyBot to create the movement. The four motors' compliance parameters are then simultaneously set to false to allow control. Then each of the motors are set to a degree to which raise the right arm. Next, the r_shoulder_y motor is waved back and forth three times for half second time intervals. Before executing the next line of code, wait=True confirms that the motors have successfully reached their target positions. Finally, RowdyBot is set back to a stationary position.The exact code to exectue this behavior can be seen in Figure 4. This is the general structure for all of the functions in the program.

B. RowdyBot Humanoid Torso Robot Alexa Skill

The Alexa Skill that is coupled with the RowdyBot software, is similarly structured as the robot's software. It has numerous functions, also known to Alexa as intents. Each one of these intents is used to trigger its corresponding function in the Python script. After the RowdyBot software has been executed, a response is sent back to Alexa in the form of a statement or a question as stated in the section III. C before. In the code segment above, the last line is the response that is sent to Alexa. After successful execution of the code segment, Alexa utters, "Hello User! How are you today?" Notice that this is a response of type question meaning that the session stays open and waits for a response from the user to determine what to do next. An example of the code that controls this Skill can be seen in Figure 4.

Once an argument that corresponds to a valid intent is received, Alexa will respond with either the appropriate function call, or will signal a standard error message. The structure of the basic interface for the skill can be found in its intent schema, Figure 5. It lists the intents used, as well as a few standard Alexa intents that are not implemented in this program. The intent schema is what is used to map invocations with appropriate predefined responses as well as triggering backend functions.

```
@ask.intent('PoppyHelloIntent')
def poppyHello():
        poppy.r_elbow_y.moving_speed = 50
        poppy.r_arm_z.moving_speed = 50
        poppy.r_shoulder_y.moving_speed = 50
        poppy.r_shoulder_x.moving_speed = 50
        rest_pos = {'head_y': 0,
                    'head_z': 0,
                    'abs_z': 0,
                    'bust_y': 0,
                    'bust_x': 0,
                    'r_shoulder_y': 0,
                    'r_shoulder_x': 0,
                    'r_arm_z': 0,
                    'r_elbow_y': 0,
                    'l_shoulder_y': 0,
                    'l_shoulder_x': 0,
                    'l_arm_z': 0,
                    'l_elbow_y': 0}
        time.sleep(0.5)
        poppy.r_elbow_y.compliant = False
        poppy.r_elbow_y.goal_position = 0
        poppy.r_arm_z.compliant = False
        poppy.r_arm_z.goal_position = 0
        poppy.r_shoulder_y.compliant = False
        poppy.r_shoulder_y.goal_position = -90
        poppy.r_shoulder_x.compliant = False
        poppy.r_shoulder_x.goal_position = -90
        poppy.r_elbow_y.goto_position(30, 0.5, wait=True)
        poppy.r_elbow_y.goto_position(-30, 0.5, wait=True)
        poppy.r_elbow_y.goto_position(30, 0.5, wait=True)
        poppy.goto_position(rest_pos, 0.5, wait=True)
        return question('Hello user! How are you today?')
```

Figure 4. A sample of the control loop for the triggered Alexa Skill. It initiates the system and provides the voice feedback while providing control commands to the RowdyBot to move its limbs. The structure is in the following format: object name, the name of the joint, underscore, the direction, function name and the arguments in the parenthesis.

```
{
  "intents": [
    {
      "intent": "AMAZON.CancelIntent"
    },
    {
      "intent": "AMAZON.HelpIntent"
    },
    {
      "intent": "AMAZON.StopIntent"
    },
    {
      "intent": "PoppyInitIntent"
    },
    {
      "intent": "PoppyHelloIntent"
    },
    {
      "intent": "PoppyHowAreYouIntent"
    },
    {
      "intent": "PoppyWeatherIntent"
    }
  ]
}
```

Figure 5. In addition to default Skills, predefined Skills have added to the Echo device for the digital assistant. They are identified based on the keywords that said by the user; then, a required Skill is sent through the control algorithm, so the robot can act as programmed and visual and vocal feedback can be provided.

V. RESULTS AND DISCUSSION

Once the RowdyBot software, the Skill, and the Ngrok tunnel are all setup and running the total outcome of the project can be simulated. In Figure 6 the *Weather* intent has been illustrated, the user asked the system "how is the weather outside?", and the responses were in both visual and vocal by the system, by the robot acting like a human and looking up to the sky and providing the most current weather conditions from the digital assistant respectively. Just from this short demonstration, the value of this technology can be immediately seen [18].

A. Interpretation

Though, the simulation is a crude representation of the intended design, several key conclusions can be made.

First, the physical behavior that the robot performs is crucial to the robot's humanization. Body movement makes up the majority of human behavior. Even without vocal interaction, the experience can be maintained by simple physical gestures that suggest emotion and behavior. With improvements in software, these gestures will be improved, and more gestures will be added to further sustain the element of disbelief. With these improvements, the physical aspect of human interaction could be astoundingly accurate.

The next conclusion can be made on the robot's voice responses. Though the conversation is short, it can detail the importance of conversation. The most important aspect of a conversation is the connection that is made when true conversation happens. With an entity as powerful as Amazon's Alexa, this conversation can be made more fluid. With deeper networks of responses, the Alexa software can impressively mimic a legitimate conversation. This conversation will further improve the interaction between the user and the robot.

Finally, when the physical behavior is coupled with conversation, the study comes to light. It is an impressive experience when interacting with an artificial being, even in a modest scale such as this. The robot suddenly has somewhat of a conscious personality. The mannerisms of the robot provide information of the robot's physical behavior and even its personality. The vocal responses provide a diction that is specific to the robot just like any human. With the two combined, the robot creates a palpable personality that is essential to a unique interaction.

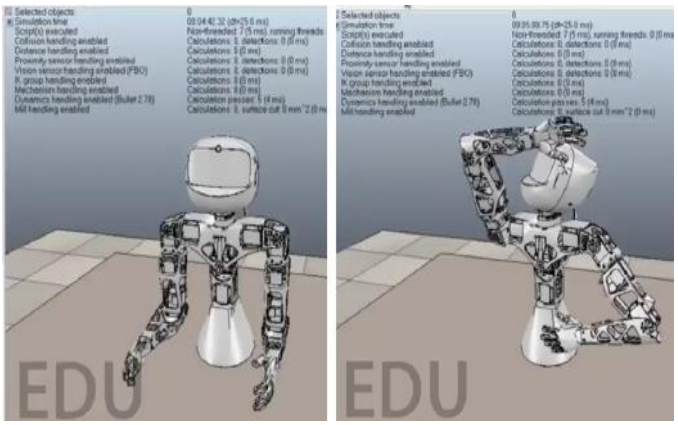

Figure 6. The humanoid robot torso object has been created and implemented in the V-REP environment. For communication purposes, the V-REP environment has connected via network tunneling to the AWS that is waiting for an Alexa Skill to be triggered.

B. Future Direction

This project is only a brief demonstration of what is possible using these tools. In the future RowdyBot could be further developed to include more conversational intelligence as well as more human body movement. RowdyBot could also be given skills that allow it to interact physically with the world as in simple human gestures, or picking objects up and placing them in designated locations and other more intelligent functions.

REFERENCES

[1] D. R. Olsen and M. A. Goodrich, "Metrics for evaluating human-robot interactions," in Proceedings of PERMIS, vol. 2003, 2003, p. 4.

[2] J. Scholtz, "Theory and evaluation of human robot interactions," in System Sciences, Proceedings of the 36th Annual Hawaii International Conference on. IEEE, 2003, pp. 10–pp.

[3] M. Katzenmaier, R. Stiefelhagen, and T. Schultz, "Identifying the addressee in human-human-robot interactions based on head pose and speech," in Proceedings of the 6th international conference on Multimodal interfaces. ACM, 2004, pp. 144–151.

[4] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, and M. Goodrich, "Common metrics for human-robot interaction," in Proceedings of the 1st ACM SIGCHI/SIGART conference on Humanrobot interaction. ACM, 2006, pp. 33–40.

[5] B. D. Argall and A. G. Billard, "A survey of tactile human–robot interactions," Robotics and autonomous systems, vol. 58, no. 10, pp. 1159–1176, 2010.

[6] R. Y. M. Li, H. C. Y. Li, C. K. Mak, and T. B. Tang, "Sustainable smart home and home automation: Big data analytics approach," International Journal of Smart Home, vol. 10, no. 8, pp. 177–187, 2016.
[7] M. Fischer, S. Menon, and O. Khatib, "From bot to bot: Using a chat bot to synthesize robot motion," in 2016 AAAI Fall Symposium Series, 2016.
[8] R. Kapadia, S. Staszak, L. Jian, and K. Goldberg, "Echobot: Facilitating data collection for robot learning with the amazon echo," 2017.
[9] A. Purington, J. G. Taft, S. Sannon, N. N. Bazarova, and S. H. Taylor, "Alexa is my new bff: social roles, user satisfaction, and personification of the amazon echo," in Proceedings of the 2017 CHI Conferencextended Abstracts on Human Factors in Computing Systems. ACM, 2017, pp. 2853–2859.
[10] G. Dizon, "Using intelligent personal assistants for second language learning: A case study of alexa," TESOL Journal, vol. 8, no. 4, pp. 811–830, 2017.
[11] Poppy Team. Poppy project documentation. Available in https://docs.poppy-project.org/en/
[12] "Virtual Robot Experimentation Platform USER MANUAL", version 3.5.0 Available in http://www.coppeliarobotics.com/helpFiles/index.html
[13] "AWS Getting Started Resource Center", Available in https://aws.amazon.com/getting-started/
[14] Control Raspberry Pi GPIO With Amazon Echo and Python. Available in http://www.instructables.com /id/Control-Raspberry-Pi-GPIO-With-Amazon-Echo-and-Pyt/ 2016
[15] Alexa Python Tutorial Build a voice experience in 5 minutes or less. Available in https://developer. amazon.com/alexa-skills-kit/alexa-skill-quick-start-tutorial
[16] John Wheeler. Flask-Ask, Rapid Alexa Skills Kit Development for Amazon Echo Devices. Available in http://flask-ask.readthedocs.io/en/latest/
[17] "NGROK Documentation", 2018, Available in https://ngrok.com/docs
[18] "RowdyBot, Human Robotic Torso" Available in https://www.youtube.com/watch?v=DpchzPXvZjU