

**UNIVERSIDAD DE COSTA RICA  
FACULTAD DE INGENIERÍA  
ESCUELA DE CIENCIAS DE LA  
COMPUTACIÓN E INFORMÁTICA**

**CI2657- ROBÓTICA  
Prof. Kryscia Daviana Ramírez Benavides**

**Tarea #3: Q.bo**

**Elaborado por:  
Jose Víquez B37638  
Geovanny Zúñiga B37822  
Luis Diego Hernández B23188  
Emmanuel Arias B30640  
Gloriana Garro B32831**

**5 de Septiembre del 2016**

## Introducción

Una habilidad que deben tener los robots es poder interactuar correctamente con los usuarios y esto implica que puedan comunicarse de forma correcta. La forma en que las personas nos comunicamos diariamente es a través de la voz, nos hablamos y mantenemos conversaciones y es la forma en que podemos dar y recibir órdenes, por eso es importante que los robots sean capaces de entender lo que los usuarios les dicen y también que puedan expresarse con los usuarios de la misma manera (“hablar”).

El robot Q.bo posee una unidad de procesamiento central con un sistema operativo basado en la distribución de linux Ubuntu. Este sistema operativo posee las herramientas necesarias para el desarrollo de aplicaciones que utilicen sistemas de reconocimiento y procesamiento de voz. Entre sus herramientas está la plataforma ROS, la cual vamos a usar para explicar y detallar el proceso de programación de una aplicación de reconocimiento de voz.

### Etapas de un programa de reconocimiento y procesamiento de voz para robots Q.Bo

1. Reconocimiento de voz
2. Procesamiento de texto
3. Síntesis de voz

Al estar el sistema operativo del Qbo basado en linux, este funciona con diferentes aplicaciones de reconocimiento de voz. Dos sistemas de reconocimiento y sintetización de voz populares para el robot Qbo son Julius y Festival respectivamente. Estos lo que hacen es aprender de una serie de datos suministrados para que el robot sea capaz de entender cuando se le habla, así como poder contestar acorde a lo que se le está diciendo. Para poder aprender, ambos dependen de grabaciones de textos que son procesadas y compiladas para permitirle al robot entender los diferentes fonemas de los idiomas soportados. Con esto, el robot va aprendiendo a reconocer diferentes palabras y frases. Ambos sistemas son open source y permiten que los usuarios colaboren con grabaciones de textos en específico para ayudar a mejorar la sintetización de texto.

Cabe mencionar que el software de reconocimiento de voz, *Julius*, permite decodificar lo que se le dice casi en tiempo real. Este programa incorpora diferentes técnicas de búsqueda que facilitan la decodificación del discurso. El software *Julius* ha sido modularizado de manera en la cual es independiente de otros modelos. Este último dato permite que sea más eficiente.

Además de esto, el *framework Festival Speech Synthesis* es un sistema que permite producir un discurso de manera artificial. Este sistema toma texto, lo procesa y lo dice, como si fuera discurso. Este programa ha sido desarrollado en C++ y cuenta con gran número de APIs.

Por otro lado, VoxForge es un proyecto completamente open source liderado por Ken McLean que busca mejorar el análisis y reconocimiento de voz. La idea es que los usuarios colaboren suministrando grabaciones de los mismos leyendo un texto en específico. Estas grabaciones pueden ser hechas por cualquiera en la misma página del proyecto. La idea es que las grabaciones puedan ser procesadas y agregadas a la base de datos de fonemas y luego cualquiera pueda usar la misma para desarrollar una aplicación que reconozca y entienda voz.

El reconocimiento de voz consiste en grabar lo que el usuario está diciendo. Esta grabación se procesa para que pueda ser entendida y para esto se le hacen diferentes procedimientos, como normalizar, se le elimina el ruido y inclusive se puede separar por frecuencias. Luego, la grabación es separada en segmentos de centésimas de segundo, y estos segmentos son emparejados con los conocidos fonemas que se habían analizado previamente. Esta es la conversión de una señal analógica a una digital. Luego que se tiene una señal digital, sigue el análisis para poder encontrarle un sentido a la grabación que se tiene, se interpreta.

## **Referencias**

<http://openqbo.org/wiki/doku.php?id=start>

<http://www.ros.org/>

<http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition.htm>

<http://thecorpora.com/blog/?p=411&lang=es>

<http://www.voxforge.org/>

[http://julius.osdn.jp/en\\_index.php?q=index-en.html#feature](http://julius.osdn.jp/en_index.php?q=index-en.html#feature)

<http://cmusphinx.sourceforge.net/wiki/tutorial>

<http://www.cstr.ed.ac.uk/projects/festival/>

<http://www.explainthatstuff.com/voicerecognition.html>

[http://julius.osdn.jp/en\\_index.php](http://julius.osdn.jp/en_index.php)

<http://www.cstr.ed.ac.uk/projects/festival/>