**KNUCKLES, THE ROBOTIC ARM**

Senior Design Proposal 2018-2019

**Faculty Advisor:** Dr. Aaron T. Becker

**Sponsor:** ECE Department & Makerspace

**Team Captain:** Andrew Blanchard

**Team:** Matthew van Zuilekom, Paola Hernandez, Rym Benchaabane

**INTRODUCTION**

Background:

Our senior design team aims to assemble Knuckles, an assistive robotic arm, that will operate using high-end technology on a desk or in a lab space. Knuckles is being designed to identify, acquire, and provide an item to an user, with the use of object and voice recognition. The arm scans the environment periodically to create a 3D map, where it identifies its surroundings. Our team is manipulating object and voice detection software to control the robot with voice recognition and provide commands. The robot will use these senses to improve the accuracy of recognition with a training. Furthermore, Knuckles should be able to function autonomously to remap the room and assist at the user’s request.

**PROJECT DESCRIPTION**

The **mechanical properties** of Knuckles are divided into two components, the arm and the gripper. The **arm’s 5-axis movement** and mechanics will be developed using ROS on Linux Ubuntu 16.04 LTS. We are connecting the arm’s joints with a total of **6 stepper motors** (2 for the base). These motors will be processed by **ROS** and controlled by **Arduino**. Programming the ability to move autonomously allows the robot to decide how to move without external input beyond voice commands or manual input on ROS. The gripper is a three-finger design; each finger will have a set of **tactile sensors** to give the robot feedback on the pressure applied to the object it is attempting to grasp. The gripper will be controlled by a **DC motor** with an H-bridge. The gripper can therefore respond accordingly and autonomously adjust its grip with sensors. Once functioning in conjunction with the software components, it will be able to calculate and complete the best path to the item and its return path to the user. The robotic arm and the gripper will be 3D printed using an AutoCAD design. The design is inspired by the open source robotic arm [BCN3D MOVEO](https://www.youtube.com/watch?v=XSY0kieEL8A) (3D printed), with the gripper designed from an [IEEE research paper](https://ieeexplore.ieee.org/document/7109102/?part=1%7Ctable1#).

For this project, we are programming in Python 3.5 and C programming. The two main software applications are visual and voice recognition.

The **visual processing** for Knuckles, including mapping and object recognition will be handled using **OpenCV, Rviz, and TensorFlow**. Knuckles will utilize the Intel RealSense Depth Camera D435 using the mentioned software packages in ROS. **OpenCV** analyzes the video feed to **detect, identify, and log the location of objects** as the room is mapped. The recognition features will be executed through **TensorFlow**. **Rviz** is our 3D visualization application for ROS that will work with the Intel Camera. These libraries will allow us to access and process images, while training with TensorFlow and improving the visual recognition. The user will activate Knuckles with a voice command which will begin the mapping of its surroundings and the execution of the command.

The robot will be able to receive **voice commands** to perform actions such as handing objects including phones, pencils, etc. The microphone will be incorporated with an [Intel® Speech Enabling Developer Kit](https://click.intel.com/intelr-speech-enabling-developer-kit.html). This add-on module is designed to fit the Raspberry Pi3 board. The **Raspberry Pi3** sends the audio to the [Amazon Alexa Voice Service](https://developer.amazon.com/alexa-voice-service) and receives the text request to be processed using a listening script connected to **ROS**. The command will be translated into the object data detected from the map created via **RViz**. The robot’s arm motors will move and execute the motion command. A voice command will be used to “wake-up” Knuckles from its standby state, which then allows Knuckles to execute commands.

**ROLES OF THE TEAM**

* Andrew Blanchard - Team captain, ROS, object recognition, hardware
* Matthew van Zuilekom - ROS, object recognition, hardware
* Paola Hernandez - ROS, object recognition
* Rym Benchaabane - ROS, voice recognition, hardware

**LIST OF ROS PACKAGES**

* Rviz
* MoveIt!
* Gazebo
* ros\_control package
* find\_object package
* rtabmap package
* uvc\_camera package
* ROSgui to build menus using qt\_designer

**CURRENT PROGRESS:**

* ur3 and ur5 simulation (Rviz, MoveIt!, Gazebo) and object tracking (find\_object package) working within Rviz. Arm and object tracking have not communicated.

**TIMELINE**

**Milestone 0 - September 24th 2018**

* Finalize arm and gripper design

**Milestone 1 - October 22nd 2018**

* 3D print arm and gripper design
* Implement motors to the assembled robotic arm

**Milestone 2 - Thanksgiving Break - November 20th 2018**

* Design the mount and mount camera on the arm
* Mount microphone on the arm
* Mount tactile pads on the gripper

**Milestone 3 - December 13th 2018**

* Simulation for robotic arm and gripper mechanical functions working properly
* Manual control of robotic arm

**Milestone 4 - January 20th 2019**

* Implement 3D mapping
* Simulate object detection and implement it on the physical robot. Robot will be able to locate and pick up objects in its environment
* Use commands through text

**Milestone 5 - February 15th 2019**

* Improved object recognition and user position recognition
* Implement voice recognition software to accept voice commands

**Milestone 6 - Last Milestone before IEEE conference - Friday March 8th 2019**

* Implement voice commands reference library to the robot

**Milestone 7 - April 15th 2019**

* Fine tuning

**Final Milestone - Senior Design presentation**