kuk-A-droid

1. Cover page

a. Project full title and short title/acronym

Kuk-A-droid: your mobile manipulator buddy

b. Applicant contact data

i. Name of project leader

My name is Arnaud Bertrand and here are my contact informations:

- mail: rnd.bertrand@gmail.com
- mobile phone: +33 6 22 20 74 01
- https://github.com/Arn-O
- https://twitter.com/A_rnO_

ii. Institution of project leader

No institution. The project is proposed as an Open Source project under the MIT license, open to anybody willing to contribute. All code and documentations are maintained on GitHub, in the kuk-A-droid repo.

c. Short summary of project proposal

This project is basically a human-robot interaction (HRI) work designed around the KUKA youBot. The main goal is to be able for anybody to interact with and operate the robot, without using complicated application and even no software interface. One will simply use the speech to send orders.

To develop and foster the relationship between the human and the robot, we will also develop affective behaviors. This kind of feature is required to make the robot engaging and integrated in the neighborhood of humans. The aim is to make the human more comfortable, and finally to develop a more effective interaction thanks to the response to social non verbal expectations. Another key part of the project is that it is based on a Android mobile device. The upper part of the framework (the most cognitive one) will be developed on it, and the sensors of the device will be used as much as possible. The Android app will send generic requests and orders to the youBot.

Those requests will be read and translated to actuators commands by an applicative layer specifically tailored for the youBot. We also target to deliver a methodology to translate generic behaviors to any kind of robotics platform.

2. Team description (max. 1 page)

a. Institution / Laboratory / Group description

With a lot of respect for the traditional scientific institutions, we consider Open Source project as an alternate way of organization, relevant for some research challenges. The project is more open than just the code. The inspiration behind is Open Science and Open Research. The topic of this project is by nature open to a wider audience than just robotics engineer, like animators or social scientist.

A collaboration opportunity has been open the last week of October with Nexus-Computing GmbH Switzerland. They could handle the development part on Android. At the time of writing, this is really new and how we could collaborate is under definition.

b. Background in mobile manipulation (experience, projects, …)

Well, none in fact, except an extensive reading in the past months.

I have interesting assets through, like a strong background in mathematics, physics and project management. I'm skilled in C/C++, Java and Python programming. I am also an experienced user of Linux and Ubuntu in particular.

3. Motivation and objectives

a. General description of industrial and scientific challenges to be tackled (relating to an application of mobile manipulators and/or the development of specific components/algorithms for a mobile manipulator)

The first challenge is to find an **effective way to interact** with the youBot. Above natural language processing and computer vision algorithm, it has to be relevant for the context of robotics. The next challenge will be to **develop behaviors and emotions that are readable and convincing** for a human, considering that the youBot is poor in social attributes (no face for example). The last challenge is to **operate a robot with the sensors from an Android device**.

b. Objectives of the proposed work

Objective 1: build a "synthetic nervous system" based on Android. This part is in a way similar to personal assistant applications that can be command by speech, like Google Now or Siri. Predefined orders will trigger appropriate answers and send the order to the robot. We are not addressing "digital life" here, just a clever engineered design. In a word, we are faking life.

Objective 2: define a methodology to translate generic request from the nervous system into activators commands for any kind of robotic platform. The fun of this part is to work with robots as far as possible of humanoids.

Objective 3: apply the previous methodology to the youBot and develop the applicative layer that will operate the robot based on the generic requests from the nervous system. The behaviors have to be life-like to be readable and convincing. The conception of the behaviors is an endless work, especially for a mobile manipulator. We will focus only on a couple of ones during the timeline of this project.

4. Approach and realization

a. Technical details of proposed solution including a description of the work the project capitalizes on and why the proposed solution is promising

The project capitalizes a lot on the PhD Thesis of Cynthia L. Breazeal, "Sociable Machines: Expressive Social Exchange". This document gives a very structured and detailed framework for designing affective behaviors for a robot, and this is what she did successfully for Kismet.

Reading the +250 pages of the thesis is powerfully inspirational. Here is a short summary of the key points that we will use for this project. There is an example for illustration at the end of this paragraph.

The upper part of the system is dedicated to the perception and the motivation. Several drives are hard-coded into the robot. The robot will use behaviors and emotions to satiate the drives, which will be evaluated by the perception system. This last part is also in charge of constantly evaluating the environment.

The three main drives are `social`, `stimulation` and `fatigue`. The social drive is the dominant one in case of interaction with a human. The stimulation drive motivates the robot to explore its environment, being the area (since the youBot is mobile) or an object found. Finally the fatigue drive is there to induce some rest.

Technically speaking, those components will be coded on the Android device into an app. Android OS is packed with libraries to identify faces and the position of the eyes (class FaceDetector.Face from the API android.media) and recognize speech (API android.speech). They will be used to localize the humans and process the speech.

At any time, the motivation system promotes the most relevant drive with a continuous evaluation of the context by selecting the appropriate behavior. The behaviors are predefined and organized in a tree, logically structured, to satiate the dominant drive.

The Android app will continuously send an assessment of the behaviors to an application on-boarded on the robot through a USB connector using the Android open Accessory protocol. This application is integrated into the robotics middleware ROS, and will publish the behaviors and emotions activation level (ROS publisher node) using Python and rospy librabry.

The behaviors are generic, and should be translated into activators commands specifically tailored for the youBot. This is done by another applicative layer in ROS, dubbed the "motor skills". This layer is responsible for coordinating the behaviors and the motion of the robot.

Here is a concrete example:

- *youBot is on the `stimulation` drive, and is desperately looking for something to play with (not detailed in this example)*
- *a human gets into the room, the perception detects the event and the motivation system switch the dominant drive to `social`*
- *among several behaviors, the most appropriate one in this context is `call-human`*
- *the activation level of this behavior becomes the highest*
- *the "motor skills" will catch this information and translate the `call-human` behavior into activators commands to demonstrate a posture or a motion. This posture should be read by the human as "I call you", it could be a large waving movement from the arm*
- *if this is successful, the next behavior (with the highest activation level) will be `greet-person`. On the contrary, the next behavior could be `call-human` again, or the motivation system could decide to switch the dominant drive to `stimulation` again and work with another set of behaviors, less social oriented. The robot will then act as if there is nobody in the room.*

Note: the social response, as (too quickly) described above, is a sequence of behaviors that should be closely designed as a subgroup of behaviors. There are also standalone behaviors like "seek-toys" which will trigger autonomous exploration of the environment.

The "motor skills" is a multi-layer application that will be able to collect and arbitrate between several requests to demonstrate a coordinated answer. The lower layer of the application send activators commands using primitives, to control groups of robot parts using a simplified interface (example, a primitive will manage the three planar joints, to make the bend of the arm manageable directly).

b. Work plan (milestones and intermediate results including use of resources)

Milestone 0: definition of the limited set of behaviors that will be implemented. This should include at least a simple social response and an autonomous behavior.

Milestone 1: methodology to translate emotions and behaviors into actuators commands, based on the capabilities of a robotic platform. Apply this methodology to the youBot.

Milestone 2: simple Android app sending behaviors and emotions activation level on user input, like clicking on a button.

Milestone 3: publisher node in ROS connected to the app coded before.

Milestone 4: general structure of the motor skills, and more specifically the primitives.

Milestone 5: one by one implementation of the emotions and the behaviors in the "motor skills". This work should be inspired by animation and is very similar to what is done in computer-animated movies.

c. Used hardware (sensors) and software (libraries, licenses)

The mobile device will be a Samsung Galaxy Nexus. Here are a few characteristics:

- SoC Texas Instruments OMAP 4460 = CPU 1.2 GHz dual-core ARM Cortex-A9 + GPU512 MHz PowerVR SGX540
- rear camera of 5 MP (2592×1936 px) Autofocus, zero shutter lag, 1080p video recording, (1920×1080 @ 24 fps) (angle of view 1.29 rad x 0.97 rad)
- front camera of 1.3 MP, 720p video (1280x720 @ 30 fps)
- dual microphones for active noise cancellation

Moreover, Android devices are equipped with a bunch of sensors for measurement of:

- motion like acceleration and rotation with accelerometers and gyroscopes
- environment like illumination (popular), pressure or temperature (less common)
- position with GPS and orientation with magnetometers
- distance (low range only)

All sensors of this device may be potentially used. To this list should be added the following interesting features:

- 1 HD and 1 SD camera / HD display
- audio microphone and speaker
- full range of wireless connection, WiFI, Bluetooth and mobile phones networks
- NFC (could be used to trick object localization)

The following libraries will be directly used (all Open Source licensed):

- Android 4.3 Jelly Bean
- ROS / OpenCV
- Gazebo (robot simulator)

Additional libraries may be added afterwards, once objectives and milestones studied in detail. Best effort will be done to only involve Open Source software.

Just to mention them, the following tools/OS/application will be used:

- Android Studio for Android development
- Windows 7 / Chrome / Google Drive
- Ubuntu 13.04
- Java / Python / C++
- GitHub

5. Results and measures of success

a. Results

For **emotions and social behaviors**, the readability and believability will be evaluated with people out of project using quizzes, video of simulations or of the real robot, or demonstration with the real robot directly. The duration of the interaction for a non-trained user is also a good measure of the success of the project.

The measure of success for this part of the project is a task that can be typically **crowd sourced**.

For **standalone behaviors**, like the ones from the stimulation drive, the evaluation will be based on completion rate of a given task.

b. Assessment of technology readiness level of the proposed solution including its scalability and the reusability of used and developed components

The Android platform is completely "ready" and widely deployed. The option to develop a part of the project on this platform is the key for the scalability and the reusability of the developments. All functions directly and autonomously supported by the mobile device could be transferred to another robot.

ROS is the leading Open Source middleware for robotics and is continuously under active development. There are now more than 2.000 packages in it, that cover most aspect of robotics, from robot description, low level control to motion planning with MobeIt!. Even if ROS lacks industrial references and is only used as a package manager in the youBot (we plan to push it further), it has been used on a couple of famous robotics platform, like NAO from Aldebaran Robotics or STAIR I and II from Stanford's AI lab.

6. Economic analysis of proposed solution

a. Analysis of economic impact

At a first level, the economic impact could be good, considering the rising number of robots available today. Take for examples the Roomba from iRobot or the Pioneer 3-AT from Adept Mobilerobots. The development proposed in this project could be translated to them. Let's imagine that you order by speech your Roomba to clean a given room or to complete other fun tasks.

Mobile phones will become cheap sensors for such robots, with advanced functionalities. This is a low-cost entry for anyone interested in robotics. You can start facing robotics challenge just with your phone.

For the long term, I strongly believe that robots should and will acquire social skills, that will help them to be more accepted. This will leverage the symbiosis between them and the humans. The evolution will be similar to the one of the computer, from a monolithic and distance machine to a versatile and handy companion.

c. Analysis of competitive advantage

The main competitive advantage is the mobile device in a central position and the Android OS. Even if the sensors are not the best, they cover a large spectrum of features, especially video and microphones.

Android OS is open. The development kits are free and based on standard technology. The number of developers is huge and pulled by Google Play (the Android app market).

I am obviously not the first to have thought about that. Google and Hasbro have released 2 years ago a robot-toy based on Android. Cellbots is an Open Source project that succeeded in controlling low-cost robots with cellphones. Finally, during last May, Apple announced sponsoring Anki, which planned to deliver robots controlled by iPhone and bring to the entertainment industry advanced techniques from robotics.

7. First results of the simulation part

a. Data exchange between and Android device and a Python script

This step has been validated using a proof of concept. With a proper set of commands, the PyUSB library is able to negotiate the communication with an Android device using the Android open Accessory protocol.

Read "Python Android Accessory USB communication [proof of concept]" and "Turn your (Linux-) Desktop / Arduino Yún / Raspberry Pi into an USB Android Accessory: How to use the Android Accessory Protocol with pyusb" from Nexus-Computing GmbH.

b. Methodology to translate generic behavior to actuators commands

The wiki page "Social expectations from the KUKA youBot" on the kuk-A-droid GitHub repo is a first attempt to do this work. This page is dedicated to the youBot, and is not generic as we plan to do it in this project.

c. ROS nodes

Stubs of the nodes are already available on the repo. The script `nervous_publiser.py` will publish emotions and behaviors activation levels using specific ROS messages. The activation levels can be easily visualized using `rtq_plot`.

The script `motor_skills.py` plays a sequence of emotions in `rviz`. It does not currently read the messages published by the `nervous_publiser.py`. This sequence can be watched in this video.