ROBOBO: A MOBILE ROBOT CONTROLLED BY A SMARTPHONE

S. DONCIEUX, F. BELLAS, A. PRIETO AND G. FERNANDEZ

1. Introduction

The ROBOBO is a mobile robot controlled by a smartphone. This robot has been developed at Coruna University in Spain. It is meant to be a low cost robotic platform. This platform will be used in particular in the DREAM project (`http://www.robotsthatdream.eu/`) in order to promote the work done in the project to the general public. See the overview document for details about its features.

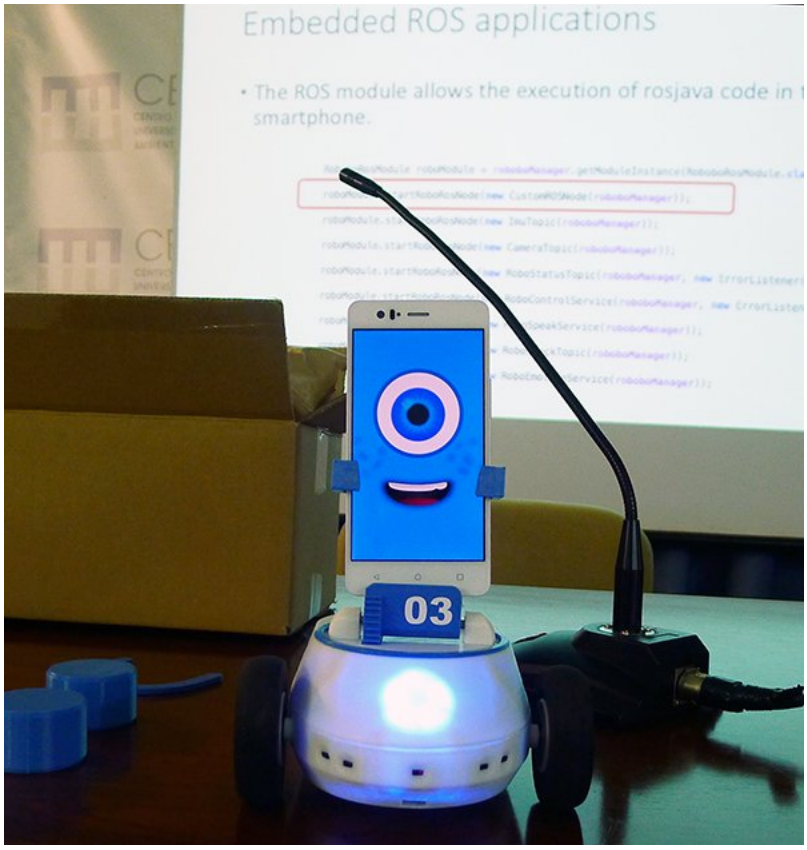

Figure 1. The ROBOBO.

To program the ROBOBO, a dedicated software framework has been defined. It is a small library that allows developers to automatically load/unload ROBOBO modules, and access to their functionality. The core functionality of the robot is provided by the modules. Currently there are 5 different released, but much will be shortly available:

- ROB interface module: Implements the communication (via Bluetooth) with the robotic platform, provoding developers with methods to move the ROB, access to the sensors, etc. The code and some usage examples can be found at:`https://bitbucket.org/mytechia/robobo-rob-interface-module`
- ROS module: Makes the ROBOBO compatible with ROS. It implements a proxy that translates the functionatlity of the ROBOBO to ROS (Robot Operating Systems) concepts like topics and services. The code and some usage examples can be found at:`https://bitbucket.org/mytechia/robobo-ros-module`
- Emotion module: Shows the robot face in the smartphone, allowing the programmer to change it to show different robot emotions. The code and some usage examples can be found at:`https://bitbucket.org/mytechia/robobo-misc-modules`
- Speech production module: A text-to-speech module that allows developers to make the robot talk using Android TTS. The code and some usage examples can be found at:`https://bitbucket.org/mytechia/robobo-misc-modules`
- Shock detection module: An example module that uses the smartphone sensors to detect shocks with objects or walls. The code and some usage examples can be found at:`https://bitbucket.org/mytechia/robobo-misc-modules`

The ROBOBO is made of two parts: the ROB, the robotic platform, and the OBO, a smartphone that can be mounted on top of the ROB. They are connected through a Bluetooth connection[1].

The ANDROID apps to control the robot are in the Google Play Store:

- `https://play.google.com/store/apps/details?id=com.mytechia.robobo.rob.application`
- `https://play.google.com/store/apps/details?id=com.mytechia.robobo.ros.application`

More information in the wiki page: `https://bitbucket.org/mytechia/robobo-framework/wiki/Home`

2. Objectives of the project

The current version of the ROBOBO development framework runs under ANDROID. It allows to build a native ROBOBO app. The ROBOBO ROS app offers an interface to control the robot with ROS, a standard robotics middleware (`http://wiki.ros.org/`). The application framework and the modules are developed in Java.

The objectives of the project are to develop:

(1) an interface to monitor several ROBOBOs

[1]They need then to be paired before launching the ANDROID app. The platform id is UPMC_XX, and the code 1234.

(2) a social network to share behaviors
(3) an iOS version of the ROBOBO framework
(4) new behaviors for the robot

Another objective is to debug and optimise the framework and current modules.

It is not expected that all functionalities are developed during the project. The selection of which to focus on is to be done at the beginning of the project.

3. Interface to monitor several ROBOBOs

It is proposed to develop a program to monitor the state of ROBOBOs and to control them. The program may be a web site running on a network to which the OBOs are connected. The development of this interface will imply the development of new modules to be run on the OBOs.

The interface will first propose to scan the network in the search for ROBOBOs. To this end, the interface will broadcast a message asking ROBOBOs connected to the same network to send their id and features:

- version of the ROB platform
- version of the ROBOBO framework
- smartphone device used
- modules installed
- modules currently active

For security reasons, a ROBOBO will not directly answer to such requests the first time it receives it. It will display a message on its screen telling that it has received this request. The IP and eventually name of the computer that has sent the request will also be displayed and the user will be asked to confirm whether he accepts or not the connection. This information will be stored and if the user has accepted, the ROBOBO will answer all id requests coming from this computer. The ROBOBO module will offer the possibility to ignore such requests. It will also allow to browse through the IP of computers that have sent one in the past in order to show their status (accepted or rejected). The module will also allow to change the status of an IP and to remove all IP from its database.

The interface will display the connected ROBOBOs and give the possibility to look at their features.

It is proposed to rely on ROS (`http://wiki.ros.org/`) for the interface implementation. ROS (Robot Operating System) is a standard middleware in robotics. It has a modular structure in which asynchronous modules communicate through dedicated communication channels (called topics). Modules may also propose on-request services. Messages have already been defined for standard sensors, including video cameras. A ROBOBO-ROS module already exists.

3.1. **Monitoring.**

The interface will propose a monitoring mode. When this mode is activated, sensor states will be displayed in the interface. In the list view mode, a compact version will be displayed showing only the IR sensor states as well as the battery level. Once a specific ROBOBO is selected, a detailed view will be proposed in which all available

sensors appear. Sensors requiring a large network bandwidth (like video cameras) can be unwatched. In this case, the ROBOBO will be notified and will not send the corresponding data. By default, these sensors will be in the unwatched mode.

The video camera image could be also inserted in the compact mode. The problem is that it requires a significant network bandwidth as there may be a large number of connected robots. An estimation of the available bandwidth will be done to check if it is compatible with the number of connected ROBOBOs. The possibility to display the video will not be offered if the bandwidth is not sufficient. An alternative is to have a low resolution video transmission mode.

When in monitoring mode, the ROBOBOs will send continuously send their sensor information. The update rate will be empirically estimated and could be adapted to the ROBOBO CPU load and to the network bandwidth.

3.2. **Control.** The interface will propose to control the robot through a limited set of actions:

- STOP
- TURN 45deg left
- TURN 45deg right
- MOVE FORWARD 10cm

The detailed view will offer more possibilities, including the possibility to turn with any angle or to activate or stop a set of behaviors available on the robot (see section 5).

4. A social network to share behaviors

The social network is implemented as a web site showing a list of behaviour programs together with an evaluation and associated keywords. The goal of this network is to share robot programs and the behaviors they have generated.

The web site works as google play store, offering the possibility to a connected user to rate and comment behaviour programs. A mobile version of the site also allows to install them on a compatible smartphone.

The difference with the google play store comes from the possibility to upload data corresponding to the behaviour that a ROBOBO has generated when controlled by the corresponding program. The behaviour is a set of robot perceptions (including video camera images). The behaviour program offers the possibility to record its perceptions and actions. As it corresponds to private data, a warning message needs to be printed when starting to record behaviour. The mobile version of the social network can access these data and offer the possibility to upload them to the behaviours of the corresponding behaviour program. The user is offered the possibility to provide some comments or details about the behaviour.

On the social network platform, the behaviours can also be commented and rated. Besides its rating, a behaviour program shows the average rating of the uploaded behaviours, if any.

OPTION: Give the possibility to record the behaviour of a ROBOBO controlled by a smartphone A through another smartphone B and provide the possibility to transfer the corresponding video from B to A so that it is added to the perceptions of A and can thus also be uploaded to the social network.

5. iOS version of the framework

The current version of the ROBO framework runs on ANDROID. An iOS version will be developed to allow the use of iPhones with the ROBOBO. It includes the ROS module. Besides the adaptation to iOS, this work package includes a study of the different alternatives to make maintenance easier, notably to avoid having two completely separate frameworks to maintain.

The ROS part of the framework can reuse or draw inspiration from `https://introlab.3it.usherbrooke.ca/mediawiki-introlab/index.php/ROS4iOS`.

This work package includes a comparison of the relative performance of the ROBOBOs with each OS.

6. New behaviors for the ROBOBO

The behavior of the robot depends on how its control program takes into account its sensors to determine its motor values. Currently there are ten different functionality modules available for developers to build a ROBOBO native app:

- Robot movement;
- Voice detection and production;
- Camera access module;
- Face detection;
- Color detection;
- Emotion module (a predefined set of robot faces to show robot emotions + a predefined set of sounds to signal robot emotions);
- Touch detection module.

It is proposed to implement several behaviours:

- Obstacle avoidance: random movement with obstacle avoidance relaying on the IR sensors. The basic idea is to turn right when an obstacle is detected on the left, and right when an obstacle is detected on the left. The robot makes a U-Turn when there are obstacles in the front;
- Follower 1: keep the front IR sensor at a particular value and turn to face the closest obstacle. This will make the robot follow the front object at a given distance;
- Follower 2: make the robot focus on an object seen by the camera (through the interface), and then make the robot follow the object by trying to keep it at a constant size;
- Line follower: use the IR sensors turned to the ground to perceive the limit of a line and follow it;
- Dancer: make the robot mode depending on the sound it perceives;

- Human interaction 1: Combines face + touch + voice detection with the emotion module and robot movement (pan-tilt).
- Human interaction 2: Based on an emotional state reacts to faces, touch and some words For instance: avoid faces / turn toward faces, become happy / scared when touched, answer when called by its name
- Surveillance: Moves pseudo-randomly in circles looking towards different places trying to find a ball of a predefined color in a room. When detected takes a picture and shows that on the screen (it can also approach the ball until is located at a predefined distance size of the ball in pixels on the lcd-). A voice command can change the color of the ball which tries to find.

7. Debugging and optimization

The ROBOBO is a novel platform that has not been extensively tested yet. It is proposed to make extensive tests of the software and propose modification of the code to increase performance and reduce CPU load and network bandwidth use. For this particular work package, a clear and rigorous methodology will need to be proposed and discussed. It implies to make an experience plan in which the experiments done are described together with their goal. Their results are also documented (how many experiences done, in what context).

Once limitations and bugs have been identified, propositions will be made to remove them and, upon acceptance, the corresponding development will be done.