# Dog-like robot companion

Michał Frankowicz
Anna Sokołowska

February 9, 2022

*Class:* Intermediate Project

*Instructor:* Witold Paluszyński, Ph.D.

Department of Cybernetics and Robotics

Wrocław University of Science and Technology

## 1 Description

The goal was to create wheeled robot that would mimic the behaviour of a dog in terms of reacting to sound and voice commands. Desired behaviour included following the source of the sound and reacting to taught commands like stay in place or perform some particular move.

## 2 Planned goals

- Hot-word recognition system with pre-learned database of commands
- Executing commands chosen by the user
- Construction of personalized robot body
- Robot following the sound

## 3 Results

All the main planned goals were realized. There was added a distance sensor to the construction, therefore, besides following the sound and voice commands, there are possible operation modes that rely on distance reading.

# 4 Introduction

## 4.1 Hardware

Robot was developed on *Raspberry Pi 4* minicomputer as project includes speech recognition, motor control and usage of sensors and therefore requires quite a lot of resources.
Apart from *Raspberry Pi* hardware of the robot consists of:

**DC motors** – two 5V DC motors one for each drive wheel

**wheels** – two rubber drive wheels, one castor wheel

**H-bridge** – h-bridge motor driver DRV8833 [1]

**sound sensors** – two sound sensors with LM393 comparator used for determining the sound level [2]

**microphone** – *Tracer* microphone used for voice capturing [3]

**distance sensor** – HC-SR04 ultrasonic distance sensor used for determining distance to the object in front of the robot [4]

**buzzer** – used for indicating acceptance of the command from user

**powerbank** – 3A, 5V powerbank for powering *Raspberry Pi*

**battery** – 9V battery for powering the motors

**switch** - switch for powering the motors

**robot case** – 3D-printed custom body skeleton

Robot case is presented on the figure 1. The lowest level holds powerbank and H-brige, middle level holds *Raspberry Pi*, sound sensors, battery for motors with a switch, buzzer and distance sensor. Top level holds the microphone and ADC converter.

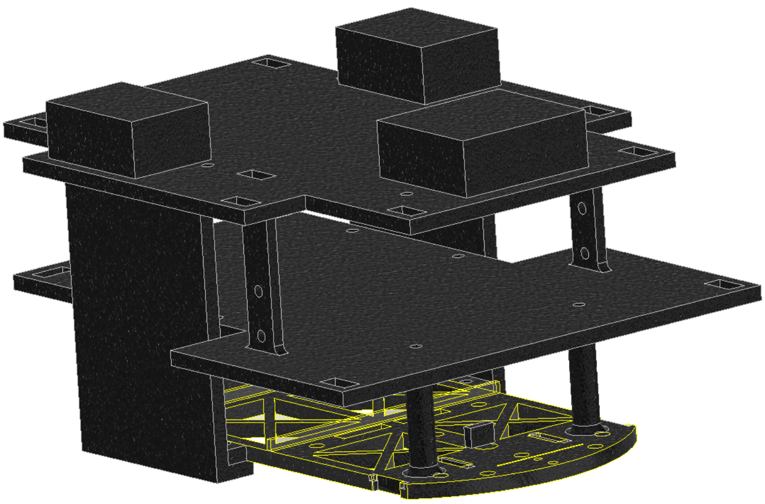

Figure 1: Model of the robot body

Assembled robot is presented on the figure 2.

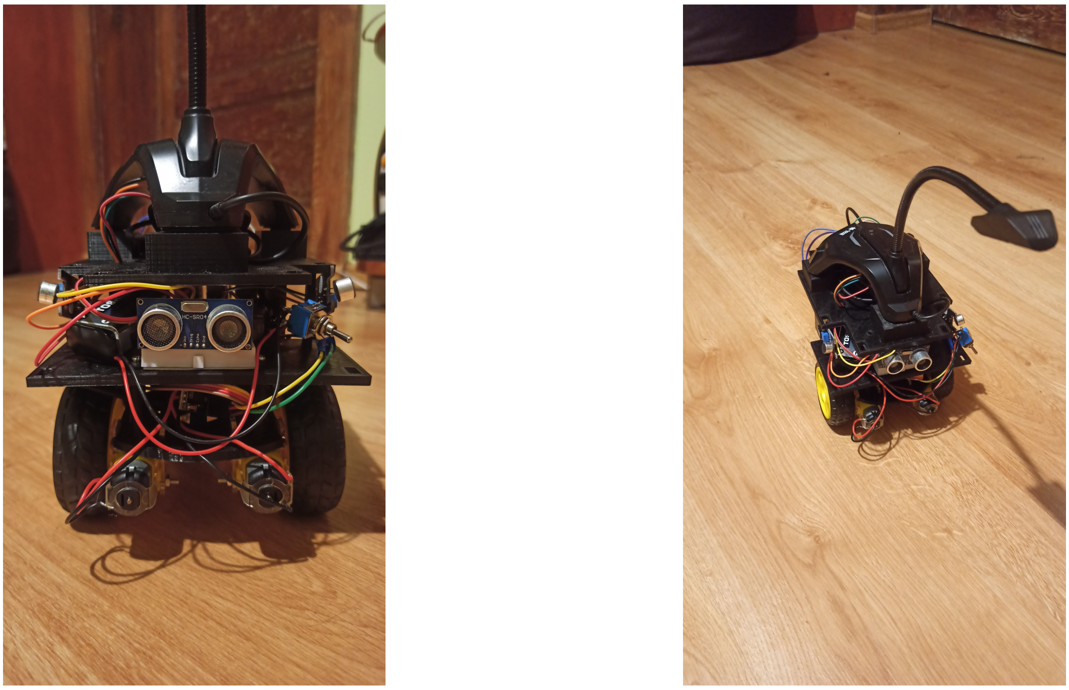

Figure 2: Robot

### 4.2 Software

Software of the robot was created in Python. It can be divided in three elements:

**SOPARE** – the main idea behind the robot was to use speech recognition to trigger some specified robot behaviour. Requirements for the software included the possibility of creating custom words, off-line words recognition and possibility to work on *Apache 2.0* license. There are plenty of solutions available, but most of them are commercial and require a license. After conducting a research, *SOPARE* [7] was chosen – it is an open source Python2 library that meets all of the requirements listed above and has easy to use interface.

**sound sensors** – handling the ADC measurements from sound sensors was done with use of Python3 library `Adafruit-Blinka`.

**main robot program** – main program of the robot was implemented in Python2. Peripherals like buzzer and motors are handled with module `gpiozero`, distance sensor with module `RPi.GPIO`. Main program also communicates with the other two software elements via `multiprocessing` module (*SOPARE*) and `zmq` module (ADC measurements).

### 4.3 Operation

Robot operates in 4 modes that can be switched using voice commands:

**task follow** Robot tries to keep the distance reading fixed so it drives forward when the object in front of it is moving away and backwards when the object is getting closer.

**task stop** Robot goes forward until distance reading is small enough.

**task rotate** Robot goes slowly forward and turns to the direction where louder sound comes from.

**go** Robot drives in direction specified by voice commands (**ahead**, **backwards**, **left**, **right**).

When robot recognizes the voice command, it is signalized by multiple buzzer beeps depending on recognized operation mode.

# 5 Results

Project was successfully developed, all main functionalities work, some extra were added. Robot is assembled on 3D-printed body. All of the components work correctly and simultaneously. Robot can be easily further developed:
Robot body is prepared to have a case added. This way the insides of the robot can be hidden. Program logic is easily extendable by other operations modes. Therefore, any other sensor can be added (e.g. light sensor, so robot can be guided by light).

## 5.1 Encountered difficulties

During the realization of the project there were some difficulties that are worth mentioning in terms of future development and improvement.

**microphone** – choosing the right microphone is very important. The chosen one is quite big and connects via USB what makes it more demanding in terms of power, but it assures high quality recording that has direct effect on performance of words recognition process.

**sound sensors** – sound sensors are very sensitive – during testing it was discovered that sounds that are caused by operation of the robot (motors running, fan from the *Raspberry Pi*) have impact on the readings, therefore it was not possible to move the robot and detect the sound source at the same time. Also the fan had to be dismounted.

**Python versions** – Python provides a lot of ready-to-go modules that help controlling robots like the one developed during the project. Because of that, it is possible to implement all of the functionalities within one program. However, as described in section 4.2, there occurs a difference in Python versions between module that handles ADC measurements and the rest of the program. Because of that ADC reading had to be implemented as separate process and this required implementing communication between processes. At first there was an attempt to use module `multiprocessing` like it is used in *SOPARE* - main communication, but there was an issue related to different serialization methods in Python2 and Python3, so finally communication with ADC program was implemented using `zmq`.

**multiple words recognition** Following advice from `SOPARE` documentation, there was constructed a logic to recognize commands built of two words. Thanks to that, many false positives can be avoided. Firstly, there were used words *operation* and *go* as main commands; one was used with operation modes, second with direction. All the words gave satisfactory results with single word recognition as presented in table 1. However, for word *operation* results of multiple words recognition were surprisingly bad, what is shown in table 2. There were tested different configurations of `SOPARE`, but no satisfying results were obtained. Following the theory that this word is too long in comparison to other commands it was replaced with word *task*, which has worse results of single word recognition but works better for full commands.

| **command** | **true positives** |
|---|---|
| operation | $76,92\%$ |
| rotate | $84,62\%$ |
| follow | $76,92\%$ |
| stop | $76,92\%$ |
| go | $76,92\%$ |
| left | $92,31\%$ |
| right | $84,62\%$ |
| ahead | $76,92\%$ |
| backwards | $84,62\%$ |
| task | $68,75\%$ |

Table 1: Recognition of single words

| command | true positives |
| --- | --- |
| operation follow | 14, 29% |
| operation rotate | 60, 00% |
| operation stop | 23, 08% |
| go left | 66, 67% |
| go right | 75, 00% |
| go ahead | 80, 00% |
| go backwards | 68, 75% |
| task follow | 50, 00% |
| task rotate | 57, 89% |
| task stop | 76, 47% |

Table 2: Recognition of combined commands

## References


[1] H-bridge datasheet. https://html.alldatasheet.com/html-pdf/437529/TI1/DRV8833/54/1/DRV8833.html.

[2] Sound sensor. https://robotdyn.com/sound-detector.html.

[3] Tracer microphone. https://www.tracer.pl/mikrofon-tracer-gamezone-gamer-led-usb.

[4] Ultrasonic distance sensor datasheet. https://datasheetspdf.com/pdf/1380136/ETC/HC-SR04/1.

[5] https://projects.raspberrypi.org/. Physical computing with python. https://projects.raspberrypi.org/en/projects/physical-computing/14.

[6] Bohan Zhou Karl Enoksson. Sound following robot. http://kth.diva-portal.org/smash/get/diva2:1200481/FULLTEXT01.pdf, 2017.

[7] Martin Kauss. Sopare. https://www.bishoph.org/.