

Robot Butler Project: Development and Handover Report

Author: Friðfinnur M. Þrastarson, University of Iceland.

Date: May 2023

Introduction

This handover report is intended to get you up to speed on the Robot Butler project, which was undertaken with the objective of transforming a decommissioned, non-operational robot into a functional, autonomous butler. The project involved a comprehensive examination of the existing robot, planning and designing new features, implementing modifications and additions, and testing the revamped robot's capabilities. This report provides an overview of the work carried out, information on the final state of the robot, instructions for its use, and recommendations for future improvements.

Initial Assessment

The robot was located in the robotics lab on the second floor.

An initial visual assessment showed that key components had been removed and it was not in an operable state.

Stakeholders were asked about prior work relating to the robot which only turned up a BS thesis from 2017 [1]. At the conclusion of that thesis the robot had functioning manual and autonomous modes of operation including obstacle avoidance.

Removing the outer shell of the robot revealed the electronics package: an acrylic plate mounted with twisted wire to the central column. On this plate were a Raspberry Pi 3 computer, a 5V UBEC spliced to a micro-USB cable powering the Raspberry Pi, a PCA9685 PWM generator, and a custom PCB. The only connections were some jumpers connecting the RPi and PCA9685, and very long jumpers connecting the three ultrasonic sensors to the custom pcb. No motor drivers were installed, and the PCB did not match anything from [1].

The filesystem of the Raspberry Pi was explored but only third-party libraries and one unfinished python script were found in the home directory. My initial conclusion was thus that the previously written code had somehow been lost, and work proceeded on that basis.

However, while writing this report and reviewing [1] in more detail I discovered that the code was likely located in a less obvious directory that I hadn't searched. More importantly, one of the thesis' appendices included a complete code listing. That this was missed originally is unfortunate and highlights the importance of thorough literature review, but having reviewed it it ultimately would not have greatly affected the course of my work.

Beyond the electronics package, the two 12v sealed lead acid batteries were found mounted to the chassis although without tightened nuts securing them down. Their open-circuit voltages were measured and found to be nominal, and they appeared to be in good condition.

The motors, belts, pulleys, and wheels were all present and securely mounted, and appeared to be in good condition. Turning the wheels by hand did indicate that the belt tension was less than optimal and that the pulley alignment could be improved, but it was within useable limits.

Design and planning

Brainstorming took place shortly after the project kickoff meeting, and resulted in several possible goals. Their feasibility was explored and they were categorized into two possible levels of functionality which were presented to faculty.

The first level of functionality consisted of three primary objectives:

1. **Autonomous Driving:** The robot was expected to navigate autonomously within a crowded room without hitting any obstacles, using the sensors already installed in it.
2. **Illumination:** The robot was expected to light up in different colors through its translucent shell. This was designed to make the robot more visually appealing and noticeable in a crowded room.
3. **Verbal Interaction:** When encountering an obstacle, the robot should be able to verbally communicate by saying "Pardon me". This feature was aimed at making the robot more interactive and polite, creating a friendly atmosphere during its operation.

The second level of functionality aimed to extend the robot's capabilities, including:

1. **Path Navigation:** The robot was expected to navigate a preset path that spanned multiple rooms. This would allow the robot to serve multiple areas during an event without requiring manual control. This function required the

development of a more sophisticated navigation system, possibly using a combination of more expensive onboard sensors and external reference points.

2. **Drink Detection and Return to Base:** The robot was also expected to be able to detect when drinks had been removed from its platter and then return to a designated home base. This feature was aimed at making the robot more self-sufficient and reducing the need for human intervention. This required the development of a weight or presence detection system for the drinks, as well as a method for the robot to identify and navigate to its home base.

Ultimately it was agreed to pursue only the first level of functionality, so a detailed bill of materials for it was prepared. Unfortunately, supplier issues caused a considerable delay in the delivery of the necessary parts

In addition, faculty requested specifically that the drinks platter be relocated from the side of the robot to the top.

Implementation

To achieve the set goals the following things were done:

Hardware:

- A main fuse was added and most chassis wiring was redone.
- Encoders were mounted to the motors using 3D-printed encoder mounts.
- An Arduino Uno microcontroller was added to the hardware architecture to read the encoders and handle timing critical functions of the robot, and permit full utilization of the Raspberry Pi in the future. It communicates with the RPi via 115200 baud serial over USB.
- A dual motor driver from Pololu was procured and wired to the motors.
- An LED driver board was designed and fabricated and an LED strip was mounted inside the robot.
- A power distribution board was made to simplify the wiring of the robot.
- The electronics plate was modified to hold the Raspberry Pi, Arduino, PCA9685, motor driver, LED driver, and power distribution board. Pipe clamps were 3D-printed and used to securely attach it to the central column of the robot.
- 3D printed parts were designed to allow mounting of the platter on top of the robot and modifications to the platter were made to accommodate the new design.

Software:

- Libraries were written for the Arduino to read the encoders, read the ultrasonic sensors, and interface with the PWM generator.
- An attempt was made to write a PID controller to drive the motors at a consistent speed using the encoders.
- Python code to interface with the arduino and drive around while avoiding obstacles was written for the Raspberry Pi.

Testing and Debugging:

- A motor and encoder were connected to the electronics assembly to write the PID controller, which revealed that the chosen encoders were too coarse to provide a good velocity signal at low speed. As a result the current incarnation of the hardware interface does not implement any closed-loop control of the motors.
- A problem was identified with the ultrasonic sensors: they did not time out hung waiting indefinitely for an echo when nothing was in range. This was solved partially in software and partially by finding some sensors that worked correctly in the lab.
- The robot was assembled and repeatably let loose on an obstacle course in the robotics lab, where it admirably avoided conveniently sized cardboard boxes but occasionally crashed into tables and people. This is due to two factors.
 1. The robot only has sensors at knee-level and so considers the empty space below tables fair game. However the robot is over a meter tall, leading to collisions.
 2. The robot has three nominally 15° FOV ultrasonic sensors, one pointing directly forward and two pointing 30 degrees off in each direction. Due to the sensors not having quite that wide an FOV and likely exacerbated by the subpar ultrasonic reflectivity of people's trousers, there are dead spots between each of the sensors close to the robot that are roughly the size of a person standing sideways.

This could not be addressed due to time constraints, but should definitely be a primary concern for future work.

Additionally, documentation was written in markdown format regarding all aspects of the robot and hosted on GitHub, in the hopes that it might be updated along with the robot. Currently it's accessible at [2] along with the source code and a journal.

Final State

As of May 8th 2023, the robot has been fully assembled with all its components, including the modified platter, 3D-printed parts, ultrasonic sensors, LED lights, motor controller, and encoders. However due to the difficulties encountered with the encoders they are left unused and so are not connected. The ultrasonic sensors are also connected partially with jumper wires as diodes had to be added inline so that they could share a single wire for their echo pulses. Unfortunately the Verbal Interaction goal was also cut for time and no speaker was installed, so it remains a straightforward goal for future work.

A final test demonstrated that the robot could move around and avoid obstacles in a controlled environment while supporting the weight of several water bottles, and so is useable for its intended purpose. Further work is however suggested to make it more capable and reliable.

Guide for Use

Powering Up the Robot:

1. Ensure that the top emergency stop is reset.
2. Locate the power switch on the bottom of the robot at the back, slightly right of center.
3. Press the switch towards the center to turn everything on.
4. Allow about 1 minute for the Raspberry Pi to fully boot.

Running the Test Script:

1. Connect to the Raspberry Pi via serial or SSH. For instructions on this, refer to [raspberrypi.md](#) in the project git repository [2].
2. Once connected, navigate to `/home/pi/butler_behaviour/` by typing `cd /home/pi/butler_behaviour/` into the terminal.
3. Run the test script by typing `python3 test.py` into the terminal.
4. To stop the robot press Ctrl-C, or hit the emergency stop.

Disassembling the Robot:

1. Remove the platter. It is held in place with three M3 screws from the top. Once the screws are removed, the platter can be lifted off of its mounting flange.
2. Lift off the dome in a similar manner.
3. The main shell is fastened in three places at the bottom. Once these fasteners are removed, the shell too can be lifted off the chassis.

Suggested Future Work

Given the constraints of time, certain aspects of the project were left unattended. It is recommended that these be the first things any future project tackles:

1. **Battery Mounting:** The 12v batteries should be securely fastened, ideally with wingnuts, to prevent any dislocation during movement.
2. **Ultrasonic Sensor Wiring:** The ultrasonic sensors are currently connected partially by a custom made cable and partially by jumper wires, including inline diodes. The diodes should be moved to the interfacing PCB and a new cable fabricated.
3. **Drive Belt Adjustment:** The drive belts should be correctly aligned and tensioned to allow smoother, quieter movement and avoid premature wear.
4. **Encoders and Speed Control:** Higher resolution encoders such as AMT102-V's should be installed and used to implement closed-loop speed control. Without this, the robot tends to drive in circles and get stuck on bumps.
5. **Improved Autonomous Behavior:** The code written for autonomous operation is extremely rudimentary. More sophisticated code should be written, perhaps taking inspiration from that presented in [1].

Additionally, these more open-ended tasks could significantly enhance the robot's capabilities:

- **Sensor Enhancement:** Incorporating additional sensors (e.g., infrared, radar, lidar, cameras) to enhance the robot's obstacle detection and avoidance capabilities.
- **Verbal Interaction Implementation:** Attaining the Verbal Interaction goal by integrating a speaker system and voice lines, would make the robot more interactive and engaging.
- **Level 2 Functionality:** Working towards the proposed functionality level 2, perhaps by setting up ROS on the raspberry pi, creating a URDF model of the robot, and conducting simulations in Gazebo.

References

- [1] A. F. Skúlason, "Sociobot Control System Design and Prototype Development," BS thesis, University of Iceland, Reykjavik, Iceland, 2017.
[Online]. Available: <http://hdl.handle.net/1946/27771>.

[2] F. M. Prastarson, "Robot Butler Project Repository," GitHub repository.
[Online]. Available: https://github.com/fridfinnurm/hi_mekatronik_butler.

Glossary

- **FOV** (Field of View) - Extent of observable world at a given moment.
- **PCB** (Printed Circuit Board) - Board connecting electronic components.
- **PID** (Proportional-Integral-Derivative) - Control loop feedback mechanism.
- **PWM** (Pulse-Width Modulation) - Technique for controlling power.
- **RPI** (Raspberry Pi) - Small, single-board computer.
- **ROS** (Robot Operating System) - Framework for writing robot software.
- **SSH** (Secure Shell) - Protocol for secure network communications.
- **UBEC** (Universal Battery Elimination Circuit) - Compact voltage regulator.
- **URDF** (Unified Robot Description Format) - XML format for robot models.