Table of Contents

Components

Lessons learned

Archive

Lessons Learned

1. Group 6: 2019 Semester 1

These are the lessons we learned when developing Willy during the 2019 1st semester.

1.1. Development

- When running into a problem, try to make a desision about this as fast as possible. The group ran into problems with the IMU. Instead of scrapping the issue, the group continued to make no progress for a few weeks. This costed the group a lot of time which could have been avoided by not continuing with the IMU.
- Starting with a development method late. Scrum was implemented in week 3 after the group struggled to work methodically. This caused a bit of delay that could've been avoided by implementing Scrum from the start.

1.2. Software

- Get the group to learn ROS early on in the project. With everyone up to speed on ROS it makes working with ROS a lot easier.

1.3. Hardware

- IMU had been removed due to it having issues. This can be caused by faulty hardware and it could be an idea to check out an IMU of better quality.

2. Group 5: 2018-2019

These are the lessons we learned when developing Willy during the 2018/2019 semesters.

2.1. Driving

- It is quite hard to deploy and undeploy the brakes.
- When the battery voltage drops below a certain level, Willy will stop driving at irregular intervals. This is probably an internal security system. The brakes are also deployed causing an abrupt stop.
- When driving manually, Willy uses characters pressed on the keyboard to move. When a lot of characters are pressed in a short time, Willy will try to process the complete list of characters, seeming to be uncontrollable.
- Willy has difficulty driving through doors when driving autonomously. This is caused by Willy being quite wide and limitations in the software for autonomous driving.
- The ultrasonic sensors have a lot of blind spots. Since the LIDAR can't scan the back of Willy, the ultrasonic sensors are the only sensors controlling the back of Willy. When Willy is driving backwards, this can cause problems.

2.2. Design

- Because of the width of Willy with garbage bins attached to the side, it is impossible to go through a door without detaching the bins, including the detachable buffers with ultrasonic sensors in it.

garbage bins are no longer attached.

2.3. Software

- The motor controller is a device that is used to control the cmd_vel topic created by the ROS navigation stack. The ROS navigation stack published geometry::Twist messages
- ROS is complicated and difficult to learn. But if everything is set up right, ROS makes it really easy to communicate between your hardware.
- If you want to create files to start multiple projects? Do not use bash files. ROS can’t handle bash files properly. use ROS launch files instead.
- The making of the DrivingController was a complex situation. That is because of the fact every ROS node has only one ROS nodehandler. To make it possible to subscribe and advertise everywhere in the DrivingWilly code, we must send the nodehandler to every sub-controller by using pointers. This request a detailed knowledge of C++ and ROS.
- The setup of the ROS navigation stack is difficult because of the fact that every robot is different. A lot of components needs to be set up on your own. As example the move_base, the transformations and rotations and the cmd_vel topic.

2.4. Hardware

- The current batteries should be powerful enough to power the current Willy.
- The 230 volt touch-screen in combination with the power converter is replaced by a 19v display.
- The brakes are not easy to deploy. That’s because the levers on the side of willy are too loose to deploy the brake. We can’t tight them because the screw is malformed.
- The GPS sensor and compass only work outside and are controlled using the GPSController.
- The kinect cannot be used outside. The IR camera can’t handle the bright sunlight.

3. Navigation

- Move_base as of this moment is not aware of the shape of the robot. This results in issues where the robot thinks it can pass between objects or rotate while in reality it can’t. It advisable to make move_base aware of the shape of the robot.
- The global planner has not been tuned yet. As of this moment it tries to create a route which is very close to the walls because it wants the fastest route. It would be better for the robot to be driving in the middle of a hallway then hug the wall. The global planner als makes very sharp corners which the robot can’t handle very well. A custom planner or tweaking the standard planner can help a lot.
- The the default default planner which is currently in optimized for a robot with powered rear wheels and a steering front wheels like a car. However willy has two powered wheels which steer using differential power and two swivel wheels.

3.1. Social interaction

- The ambient noise can make it hard to speak with Willy. We used the mute function on the microphone provided with Willy.
- The speech of Willy can be rather robotic.