Progress Report Week 2

Michiel Aernouts
michiel.aernouts@student.uantwerpen.be

May 20, 2017

Abstract

In order to save time and reduce the risk of crashing the ErleCopter, all algorithms should be tested in a simulator before implementing them. For this purpose, we will work with Gazebo. We research multiple SLAM algorithms such as 6D SLAM [1], RGB-D SLAM [2] and OctoSLAM [3] to map an indoor environment.

1 Progress

1.1 Fix simulation errors

Last week, the simulation refused to work after a while. Gazebo stopped with 'exit code 134'. I found that this was probably caused by the graphics card driver that was out of date. Unfortunately, the Ubuntu kernel I installed (14.04.5) was not compatible with the video driver for my graphics card, the AMD Radeon R7 M265. In order to solve the problem, i had to install Ubuntu 14.04.4, update the driver and follow the tutorial I wrote last week.

Edit: It's better to fix this problem by upgrading the kernel and the graphical stack by using this command: `sudo apt-get install --install-recommends linux-generic-lts-utopic xserver-xorg-lts-utopic libqt5gui5 libgles1-mesa-lts-utopic libgles2-mesa-lts-utopic libgl1-mesa-glx-lts-utopic libgl1-mesa-glx-lts-utopic:i386 libglapi-mesa-lts-utopic:i386 libegl1-mesa-drivers-lts-utopic`

(Solution found here!)

1.2 APM Planner

APM Planner is a tool that allows us to operate a robot, whether it is in simulation or in real life. Installation instructions for Ubuntu 14.04 can be found at `https://github.com/`

`ArduPilot/apm_planner#linux-`. Keep in mind that your version of Qt must be Qt 5.4.2 or newer. This version for Ubuntu 14.04 can be found at `https://launchpad.net/~beineri/+archive/ubuntu/opt-qt542-trusty`. Installation instructions:

- `sudo add-apt-repository ppa:beineri/opt-qt542-trusty`
- `sudo apt-get update`
- `sudo apt-get install qt-latest`

After that, you can change your active Qt version to Qt 5.4.2 with this command:

- `export QT_SELECT=opt-qt54 --version`

Now, you can install APM Planner for Ubuntu 14.04! Installation can take a while.

In order to control the ErleCopter that was simulated in Gazebo, I established a local UDP connection in APM Planner on port 14551. After that, I could control the drone using the graphical interface of APM Planner, and even with a PS3 controller.
However, running Gazebo and APM Planner simultaneously does not appear to be a viable option. It significantly slows down the simulation, rendering it vacuous.
Therefore, I will try to write my own tool to control the simulation using a PS3 controller. APM Planner is still serviceable for real-life control of the ErleCopter.

1.3 Propeller guards

Thanks to Eric Paillet, I now have 3D models of propeller guards. These models still have to be modified to be compatible with the ErleCopter and the dimensions of the 3D printer. For this application, it is recommended to print the model in ABS, as this material is more flexible than PLA.

1.4 First test flights

1.4.1 Gazebo

As I mentioned before, controlling the simulation with APM Planner is not a viable option. Therefore, I had to conduct the first test flights in simulation using other means.
In the fourth simulation tutorial, ErleRobotics provides a digital joystick. With this tool, I was able to control the simulation. The drawback is that the joystick has two levers, but obviously I can control only one lever at the time using my computer mouse. A possible solution is to write a ROS node that allows the digital joystick to be controlled with a PS3 controller. Furthermore, it is still possible to control the simulation via terminal commands.

1.4.2 Real world

The first real test with the ErleCopter was no success. For safety, I attached a leash to the landing gear in case it would drift away. During takeoff, it appears that some rotors appear to apply more force, which causes the drone to topple.
In order to find the cause of this problem, I will check if the cabling of the ErleCopter is correct and measure the current at each of the propellers.

1.5 SLAM research

In the past week, I examined different SLAM concepts and algorithms by studying:

- an online SLAM-course by Cyrill Stachniss
- Visual SLAM tutorial
- *'Probabilistic Robotics'* course material by Maarten Weyn

I tried to summarize some of the main concepts and algorithms in mindmaps (respectively figure 1 and figure 2.
Based on this research and the hardware that is available for my thesis, I decided to try out the following algorithms:

- 6D SLAM [1]
- RGB-D SLAM [2]
- OctoSLAM [3]
- (LSD SLAM [4] ?)

2 Planning week 3

- Octomap [5] research
  - Read papers
  - Debug code
  - Compare versions (ROS or compile from source)
- Lay down a set trajectory for the simulation
- Control the ErleCopter via APM Planner

- Test SLAM algorithms in simulation
- Finish and print propeller guard models
- Calibrate the ErleCopter

References

[1] Andreas Nüchter, Kai Lingemann, Joachim Hertzberg, and Hartmut Surmann. 6D {SLAM}{\textemdash}{3D} mapping outdoor environments. *Journal of Field Robotics*, 24(8-9):699–722, 2007.

[2] Felix Endres, Jurgen Hess, Nikolas Engelhard, Jurgen Sturm, Daniel Cremers, and Wolfram Burgard. An evaluation of the {RGB}-D {SLAM} system. *2012 {IEEE} International Conference on Robotics and Automation*, 2012.

[3] Joscha Fossel, Daniel Hennes, Daniel Claes, Sjriek Alers, and Karl Tuyls. {OctoSLAM}: A {3D} mapping approach to situational awareness of unmanned aerial vehicles. *2013 International Conference on Unmanned Aircraft Systems ({ICUAS})*, 2013.

[4] Jakob Engel, Jorg Stuckler, and Daniel Cremers. Large-scale direct {SLAM} with stereo cameras. *2015 {IEEE}/{RSJ} International Conference on Intelligent Robots and Systems ({IROS})*, 9 2015.

[5] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. {OctoMap}: an efficient probabilistic {3D} mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2 2013.

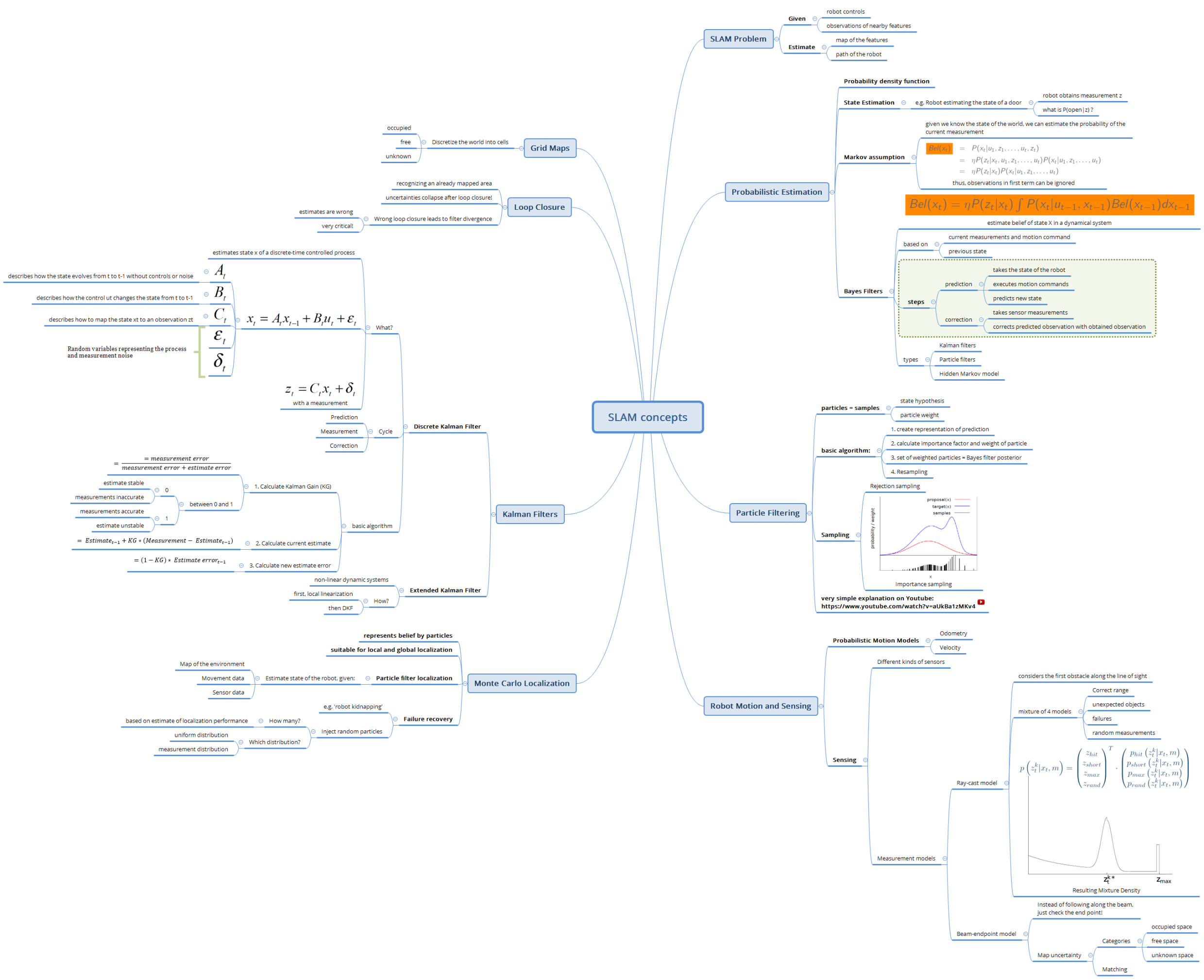

Figure 1: SLAM Concepts

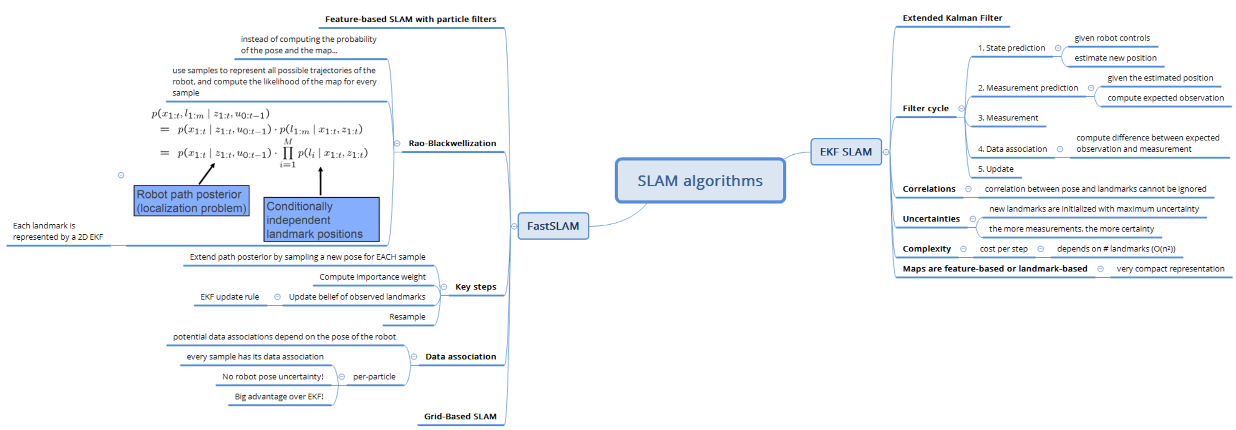

Figure 2: SLAM Algorithms