Universiteit Antwerpen

Progress Report Week 1

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.

1 Progress

1.1 Assembly of the ErleCopter

The ErleCopter itself was already assembled. The only thing that had to be done this week was mounting the ErleCopter on the landing gear. Figure 1 shows the result.

Figure 1: The ErleCopter

1.2 Make a global thesis planning

I made a global planning for the first seven weeks of the semester. Based on this planning, I will document my weekly progress and report it to my supervisors.

1.3 ErleCopter Simulation

In order to test our algorithms without being limited by the drone's battery life or the risk of possible crashes, it is important to simulate the tests before implementing them. Therefore, I will use a ROS/Gazebo simulator in which we can spawn an ErleCopter and command it using MAVProxy or APM Planner.

ErleRobotics provides a step-by-step tutorial to install and use the simulator. However, this tutorial appears to be incomplete at some points. Eventually, I was able to configure the environment and launch an ErleCopter in simulation, based on ErleRobotics' tutorials.

EDIT: ErleRobotics has updated its configuration tutorial, so you can just follow their tutorial at `http://docs.erlerobotics.com/simulation/configuring_your_environment`.

1.3.1 Environment configuration

I initially installed Ubuntu 16.04 and ROS Kinetic. However, I was not able to run the simulator in this configuration. ErleRobotics recommends to install Ubuntu 14.04 and ROS Indigo, which I eventually did.
Based on the ErleRobotics tutorials at `http://docs.erlerobotics.com/simulation/configuring_your_environment`, I executed the following steps on a native Linux machine (NOT on a virtual machine, because the virtual video drivers could not handle the simulation).

1. **Install the base packages**
   - `sudo apt-get update`
   - `sudo apt-get install gawk make git curl cmake`
2. **MAVProxy**
   - Install MAVProxy dependencies
     ```
     sudo apt-get install g++ python-pip python-matplotlib python-serial
     python-wxgtk2.8 python-scipy python-opencv python-numpy python-pyparsing
     ccache realpath libopencv-dev
     ```
   - Install MAVProxy
     ```
     sudo pip2 install pymavlink catkin_pkg --upgrade
     sudo pip install MAVProxy==1.5.2
     ```

3. **Download and install ArUco**

   - Download ArUco at `https://sourceforge.net/projects/aruco/files/1.3.0/aruco-1.3.0.tgz/download`
   - Install ArUco
     ```
     cd ~/Downloads
     tar -xvzf aruco-1.3.0.tgz
     cd aruco-1.3.0/
     mkdir build && cd build
     cmake ..
     make
     sudo make install
     ```

4. **Compile ArduPilot**

   - `mkdir -p ~/simulation`
   - `cd ~/simulation`
   - `git clone https://github.com/erlerobot/ardupilot -b gazebo`

5. **Get JSBSim**
   In order to install JSBSim, you may have to install libtool first.
   `sudo apt-get install libtool`
   After that, you can install JSBSim.

   - Install JSBSim
     ```
     cd ~/simulation
     git clone git://github.com/tridge/jsbsim.git
     sudo apt-get install libtool automake autoconf libexpat1-dev
     cd jsbsim
     ./autogen.sh --enable-libraries
     make -j2
     sudo make install
     ```

6. **Install ROS Indigo**

   - Setup your computer to accept software from packages.ros.org
     ```
     sudo sh -c 'echo deb http://packages.ros.org/ros/ubuntu $(lsb_release
     -sc) main> /etc/apt/sources.list.d/ros-latest.list'
     sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net --recv-key
     0xB01FA116
     sudo apt-get update
     ```

- Install ROS package
  `sudo apt-get install ros-indigo-ros-base`
- Initialize rosdep
  ```
  sudo rosdep init
  rosdep update
  ```
- Add ROS environmental variables to bash
  ```
  echo "source /opt/ros/indigo/setup.bash" >> ~/.bashrc
  source ~/.bashrc
  ```
- Get rosinstall and some additional dependencies
  ```
  sudo apt-get install python-rosinstall ros-indigo-octomap-msgs
  ros-indigo-joy ros-indigo-geodesy ros-indigo-octomap-ros
  ros-indigo-mavlink ros-indigo-control-toolbox unzip
  ```

7. **Create ROS workspace**

   - `mkdir -p  /simulation/ros_catkin_ws/src`
   - Initialize workspace
     ```
     cd  /simulation/ros_catkin_ws/src
     catkin_init_workspace
     cd  /simulation/ros_catkin_ws
     catkin_make
     source devel/setup.bash
     ```
   - Download repositories in src:
     ```
     cd src/
     git clone https://github.com/erlerobot/ardupilot_sitl_gazebo_plugin
     git clone https://github.com/tu-darmstadt-ros-pkg/hector_gazebo/
     git clone https://github.com/erlerobot/rotors_simulator -b sonar_plugin
     git clone https://github.com/PX4/mav_comm.git
     git clone https://github.com/ethz-asl/glog_catkin.git
     git clone https://github.com/catkin/catkin_simple.git
     git clone https://github.com/erlerobot/mavros.git
     git clone https://github.com/ros-simulation/gazebo_ros_pkgs.git -b indigo-devel
     # Add Python and C++ examples
     git clone https://github.com/erlerobot/gazebo_cpp_examples
     git clone https://github.com/erlerobot/gazebo_python_examples
     ```

8. **Compile everything**

- Compile
  ```
  cd ~/simulation/ros_catkin_ws
  catkin_make --pkg mav_msgs mavros_msgs gazebo_msgs source devel/setup.bash
  catkin_make -j 4
  ```

At this point, you may encounter some problems. First of all, it is possible that certain packages such as 'transmission_interface' or 'joint_limits_interface' are missing. This can be solved by installing them separately. E.g.:

```
sudo apt-get install ros-indigo-transmission-interface
sudo apt-get install ros-indigo-joint-limits-interface
```

Also, it is possible that cmake will fail because gazeboConfig packages were not found. If that is the case, complete step 9: 'Install Gazebo' first. After that, re-initialize the workspace and try to compile again.

9. **Install Gazebo**

   - Setup your computer to accept software from packages.osrfoundation.org
     ```
     sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/
     ubuntu-stable `lsb_release -cs` main" > /etc/apt/sources.list.d/
     gazebo-stable.list'
     ```
   - Setup keys
     ```
     wget http://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key
     add -
     ```
   - Install Gazebo7
     ```
     sudo apt-get update
     *sudo apt-get remove .*gazebo.*'.*sdformat.*' '.*ignition-math.*'
     && sudo apt-get update && sudo apt-get install gazebo7 libgazebo7-dev
     drcsim7
     ```

   As mentioned at the end of the previous step, you can now re-initialize the ROS workspace and compile everything again. After that, you can continue with step 10 'Download Gazebo models'.

10. **Download Gazebo models**
    ```
    mkdir -p ~/.gazebo/models
    git clone https://github.com/erlerobot/erle_gazebo_models
    mv erle_gazebo_models/*  /.gazebo/models
    ```

1.3.2 First launch

When the environment is configured correctly, launching the ErleCopter in Gazebo should be no problem. I was able to do this by following the tutorial at `http://docs.erlerobotics.com/simulation/vehicles/erle_copter/tutorial_1`.

- Launch MAVProxy
  ```
  source /simulation/ros_catkin_ws/devel/setup.bash
  cd /simulation/ardupilot/ArduCopter
  ../Tools/autotest/sim_vehicle.sh -j 4 -f Gazebo
  ```
- Load parameters in MAVProxy
  ```
  param load /home/michiel/simulation/ardupilot/Tools/Frame_params
  /Erle-Copter.param
  ```
- In a new terminal:
  ```
  source /simulation/ros_catkin_ws/devel/setup.bash
  roslaunch ardupilot_sitl_gazebo_plugin erlecopter_spawn.launch
  ```

It is possible that you get the following error in the MAVProxy terminal:
`APM: PreArm: RC not calibrated`.
To solve this, go to the MAVProxy terminal and type:
`param set ARMING_CHECK 0`

The simulation should be up and running! You can let the ErleCopter take off and rise to 2 metres by typing the following commands in the MAVProxy terminal:

- mode GUIDED
- arm throttle
- takeoff 2

1.3.3 Problems

Unfortunately, the simulation does not launch any more. The cause of the problem is not clear. When launching gazebo with roslaunch, gazebo quits with exit code 134. This indicates problems with graphic card drivers. However, the problem remains after updating these drivers.

2 Planning week 2

- Fix Gazebo exit code 134
- Install APM Planner for simulation
- Create protection for the rotors
- Drone calibration
- First test flights
- Camera
  - Study the camera module
  - Connect the camera to the ErleCopter