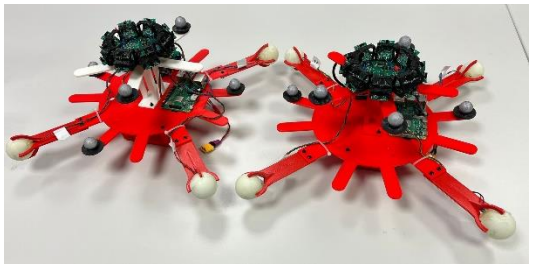
Automated Calibration Algorithm for a 3D Infrared Relative Localization Sensors for Quadrotors

Yacine Derder

Professor : Alcherio Martinoli
Assistant(s) : Kagan Erünsal

Relative localization for UAVs is a central element of several multi-drone missions, such as formation control, optimal exploring or inter-drone collision avoidance. In comparison with centralized localization, the onboard computing of the relative pose makes swarms much more robust to the failure of a single computer, hence allowing a much greater scaling potential. Additionally, infrared-based systems present several unique advantages with respect to popular vision-based systems, such as the ability to work in the dark and much cheaper computational requirements. However, in order to correctly interpolate a position and orientation from processed infrared measurements, we need an accurate estimate of the function that converts the position and power of the beacons to an RSS signal. This estimate is obtained through the process of calibration.

The main objective of this internship was to build upon an existing calibration procedure and to make it more intuitive fast and efficient to use. This includes having a clear understanding of the steps involved in order to combine them in a way that makes the calibration accessible to any user.

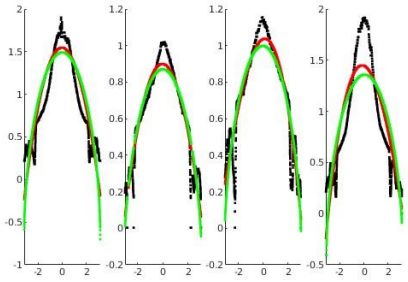


Two assembled setups with the sensors mounted

The following bash scripts were therefore implemented, each of them fulfilling a specific function :

1. `quad98_connect.sh` / `quad99_connect.sh` : these scripts allows an instant connection to the onboard computers of of the setup via SSH, therefore avoiding the need to enter a full address, username and password at every trial.
2. `get_emt_data.sh` : this script runs without arguments to make it as fast and simple as possible. Its function is to copy, transfer and rename all the files on the setup after a trial run that are needed in the context of emitter calibration.
3. `get_rcv_data.sh` : as the previously described one, this script's role it to copy, transfer and rename all the file on the setup, but in the context of the receiver calibration. It takes a single argument : the relative yaw angle between the two setups. This is necessary because many experiments are needed for the receiver calibration.
4. `get_dyn_data.sh` : this script fetches the data of the setups in the context of a generic dynamical experiment with any number of setups. Two arguments are needed, the name of the setup in which we are interested as well as the name of the output file.

Finally, many experiments were realized in varying conditions to identify a strategy that yields clean and usable results as often as possible. The most promising strategy for the emitter calibration way to do a very slow single 360° turn (~1min).



Calibration result raw data (black), polynomial fit (red) and nonlinear fit (green)