Library

Eigen has everything we need:

- Easy intuitive initialization
- All the operations
- Easy to override operations
- Intuitive and semantic

Save parameters

R.O.S. already includes it

- Less error prone
- More intuitive for students
- Documentation already exists
- Not a good idea in the launch file

Plot results

R.O.S. rqt_plot

- Good for real time
- Not yet easy to use
- No-sense for end-results

Plot results

MatplotLib

- Ok for static
- Easy to use

Plot results

Use of Images

- Requires OpenCv
- Not really the smartest idea
- Not easier than MatplotLib

Versions

- Please start simple stupid
- Update or save new version?
- Pull to GitHub (shared with Garcia)
- Write down how we want to proceed
- Write down achievements of every iterations
- Update README accordingly

Interface

Launch file the application core

- List all the parameters: saving allowed (dump)
- MOBRO fac-simile parameters list
- Saving folder and file name (error handling)
- All the args and an include, hide complexity
- Easy and to the point modifications

Procedure

- Finish first all the diagrams
- Implement it
- Alpha test, if ok Beta test
- New version: new diagrams
- Iterate

Procedure, further application

- Implement flexible interface (keyboard)
- Move to real time loop simulation
- With final solution, acceptation test (UNIGE, CORO M1)

How much we can use rviz?

A lot, actually

- We can control the robot
- We can plot the odometry (directly)
- We can plot the Kalman filter result (line)
- Keep it simple and ligth