Extended Kalman Filter & Wifi Localization

Daniel Lavado, Tomáš Kello, Martin Skaldebø and Tore Øystese

MEEC

Instituto Superior Técnico, Universidade de Lisboa

Project summary

The assignment states that a localization process of a mobile robot should be concluded. The localization is based on WiFi signals and odometry measurements. The WiFi signals is read by a computer connected to the robot, while the odometry is read from the robot itself. The robot used in this assignment is the Pioneer 3DX. The localization is done by applying the Extended Kalman Filter. The localization process is conducted indoor, on one floor. A map of the floor is created using Gmapping.

To be able to use the WiFi signals for localization, a database of WiFi signals is created along side the odometry. The results retrieved shows that the implementation of the EKF-filter can successfully locate the robot. However, the results shows also that the implementation can further be improved, the main of which is adding a part that discards readings which would suggest that the robot have moved a long distance in a short amount of time.

METHOD AND ALGORITHM

- **The Robot:**

The setup for this project is the robot ActivMedia Pioneer 3DX. To control the robot, and run the programs, a computer is connected to the robot via USB. The computer is running Linux with the ROS software implemented.

- **Extended Kalman Filter:**

Model:

$$x_k = f(x_{k-1}, u_k) + w_k$$

$$z_k = h(x_k) + v_k$$

Predict:

$$\hat{x}_k = f(\hat{x_{k-1}}, u_k)$$

$$P_k = F_{k-1} P_{k-1} F'_{k-1} + Q_{k-1}$$

Update:

$$G = P_k H_k^T \cdot (H_k P_k H_k^T + R)^{-1}$$

$$\hat{x}_k = \hat{x}_k + G_k(z_k - h(\hat{x}_k))$$

$$P_k = (I - G_k H_k) \cdot P_k$$

The extended Kalman filter (EKF) is a nonlinear version of the Kalman filter. The odometry from the robot is modelled as the state, while the WiFi signals is the measurement, z.
In the predict step, xˆ is predicted, as well as the prediction error, P. After they have been predicted they are updated in the update step. The gain, G, is calculated and will have a value between 0 and 1. In the end, before a new loop, the prediction error is updated.

- **Real Time Operating System – ROS:**

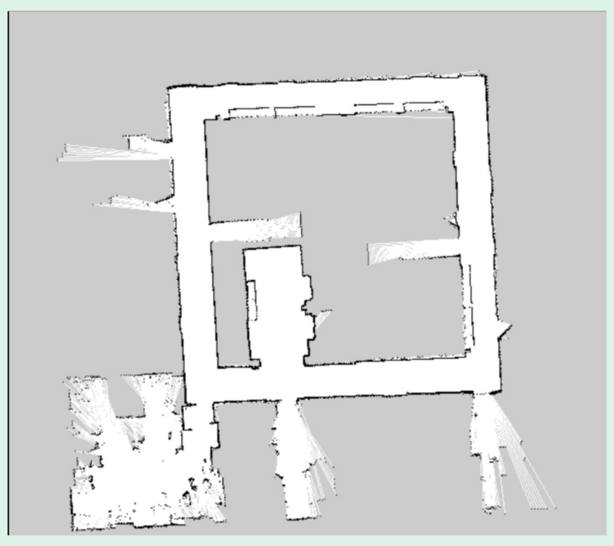
ROS provides functionality to communicate between peripheries like a robot. ROS is running on the operating system Linux, which is a distribution of Unix. After connecting to the robot, is broadcasts all the measured data like position, rotation and outputs from sensors, to ROS.

For this project only the odometry will be taking into consideration, which means the robots position and orientation. After processing this data, the correction between the robot and map base position are calculated and broadcast to ROS. When these connections are broadcast, the movement can be illustrated on a map using the graphical program RVIZ.

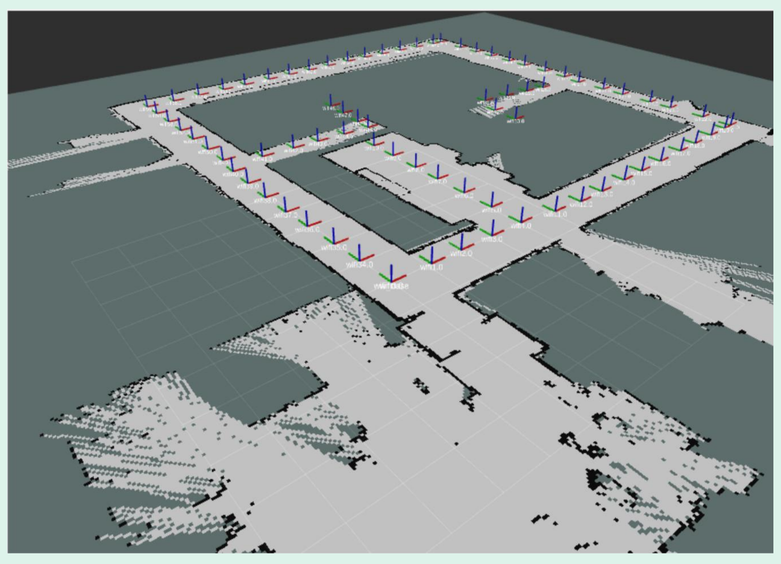
IMPLEMENTATION

- **Wifi**

For this project WiFi was used as the measurement sensor to localize the robot. To localize the robot, a map of signals in area where the robot is going to be localized had to be generated. The map of the floor was created using Gmapping. The map is of the sort, occupancy map, which represents the probability of the point to be occupied on the gray-scale.



The gray map is aligned to the center. Meaning x=0 and y=0 are located in the center of the picture. However the same values from the WiFi database are located where the first measurement was executed. With calculation of the position of the corners of the map and the WiFi database, a transformation and rotation between these points were computed. After this was applied it resulted in a fully working map, with the points of WiFi measurements.



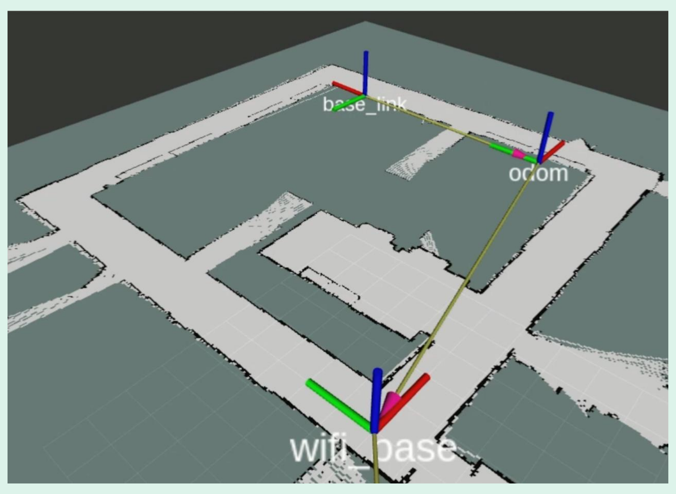
- **Filter Implementation**

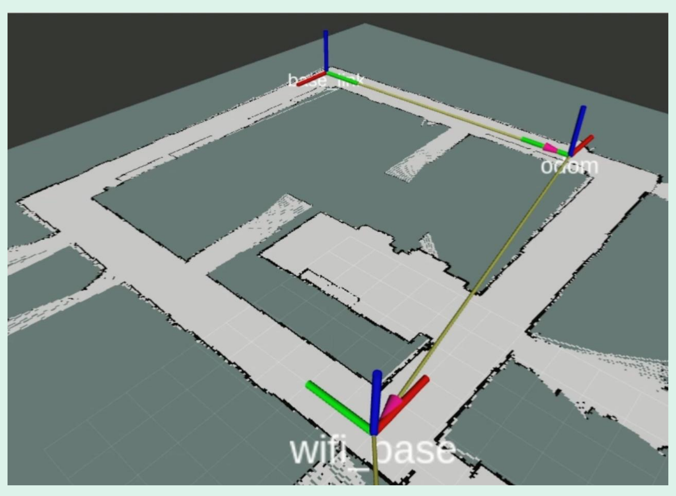
Some parts of the filter had to be adjusted and optimized to make it optimal for the case of this project. The odometry was calculated as

$$odometry_k = position_k - position_{k-1}$$

The wifi signal that are received when scanning are in RSSI format. This means the received signal consisted of the mac addresses and the corresponding signal strengths, in decibel, of all available wifi routers. In the update step of the Extended Kalman filter the measurement z and the prediction xˆ has to be in the same format. Therefore a function was created to transform the RSSI signals to the a corresponding location in coordinates.

RESULTS





The final code calculate the robots location using odometry as state input and WiFi readings as measurements input. The odometry was continually tracked, while each WiFi reading was done with a few seconds intervals. This was because the readings needed time to get accurate readings.
The figures illustrates the robot when it moves around a corner.
The distance from the *wifi_base* to *odom* is the initial difference in odometry position and map origin. This means that the readings in the figures started in the lower right corner.
The red part of the indicators shows the rotation of the robot. It can then be seen how the robot moves around the corner from indicator *base_link*.

Prof. Rodrigo Ventura, Prof. Pedro Lima, Prof. Oscar Lima