Bisio Isabella-Sole
Roncagliolo Serena

Localization Lab

INITIAL CONDITION UNCERTAINTY

These errors are due to the robot not starting at the exact initial position wanted. We set the values:

- sigmaX = 3
- sigmaY = 3
- sigmaTheta = 3*pi/180 (this measure must be in radians)

We chose 3mm for x and y, because 10mm is the max error.

INPUT NOISE

We must set the value *sigmaTuning* in DefineVariances.m, that is the standard deviation of the error for the wheels. This value takes into consideration wrong parametrizations of the wheels and differences from normal conditions due to deformation, fabrication errors, wear for the use and so on.

Since the wheels are almost identical, we can use only one sigma

$$Q_\beta = K\, Q_{\dot{q}}\, K^T \qquad \text{with} \qquad K = \begin{bmatrix} r_r/2 & r_l/2 \\ r_r/e & -r_l/e \end{bmatrix}$$

A reasonable form for $Q_{\dot{q}}$:

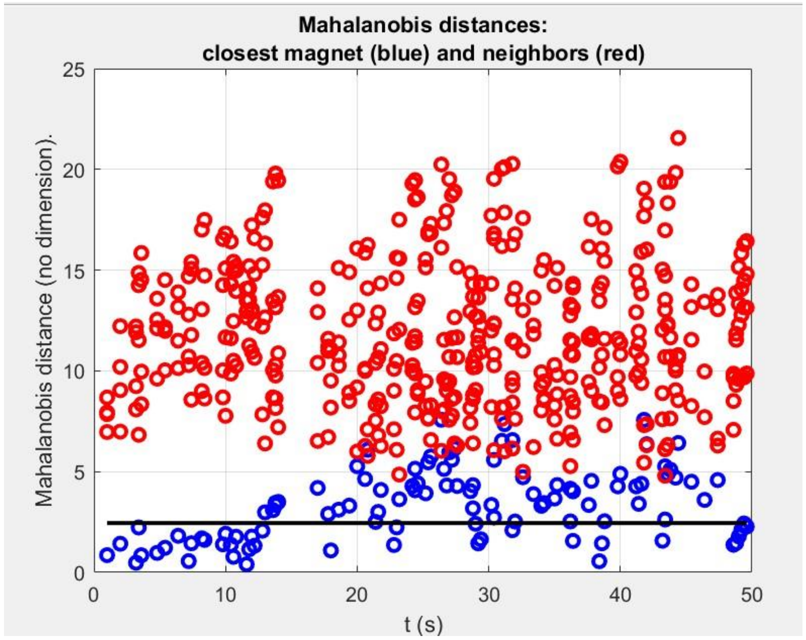
$$Q_{\dot{q}} = \begin{bmatrix} \sigma_{\dot{q}}^2 & 0 \\ 0 & \sigma_{\dot{q}}^2 \end{bmatrix}$$

We set all other variances and try to find a correct value for sigmaTuning such that in the plot of the Mahalanobis the red dots (neighbors magnets) are never under the threshold value. If they were, it would mean that we are not able to determine one and only one magnet compatible with the measurements.

The Mahalanobis distance depends on the term Y – Ŷ; it also considers the measurement noise Qgamma and the P matrix, where P is the error propagation. P depends also on Qbeta (the one we need to compute) and Qalfa, set to zero in our case.

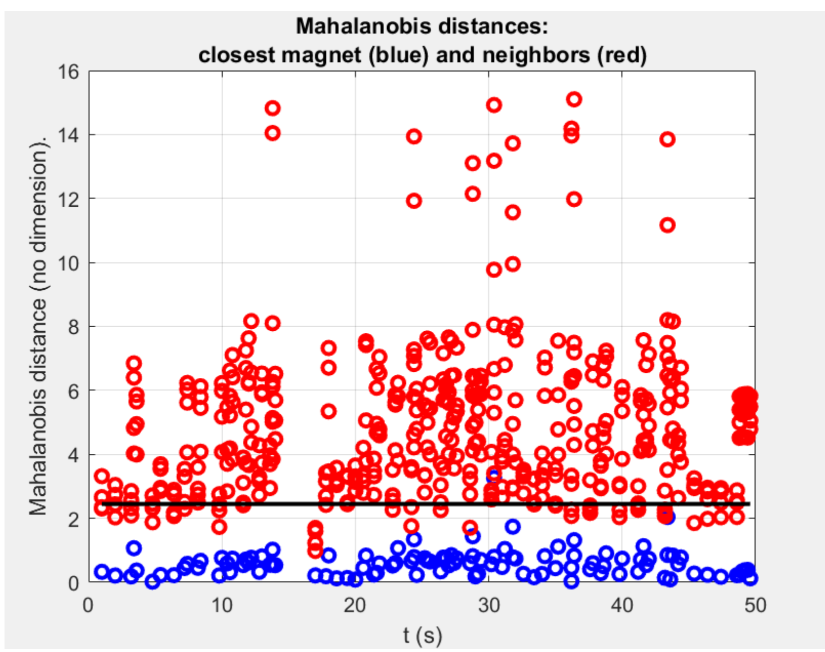
We test the sigmaTuning values on the dataset *twoloops*.

We observed that choosing a low value, for example 0.01, many possible compatible magnets (blue dots) are rejected



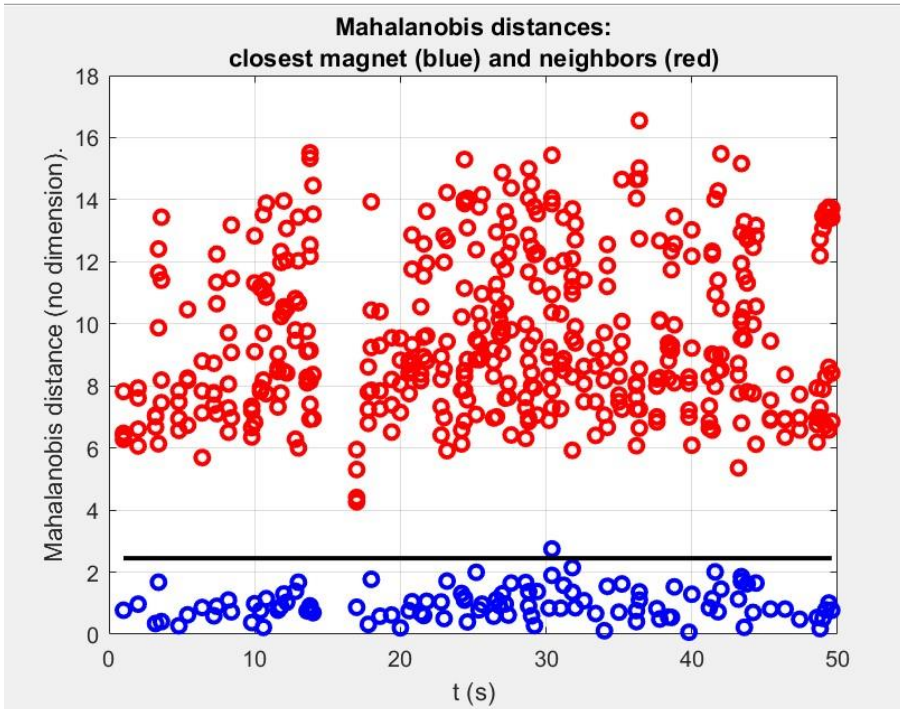
Tuning value 0.01

On the other hand, if we choose high values, for example 0.5, we may detect the red dots instead of the blue ones, which means we would not detect a correct magnet



Tuning value 0.01

A possible suitable value for *sigmaTuning* is 0.1



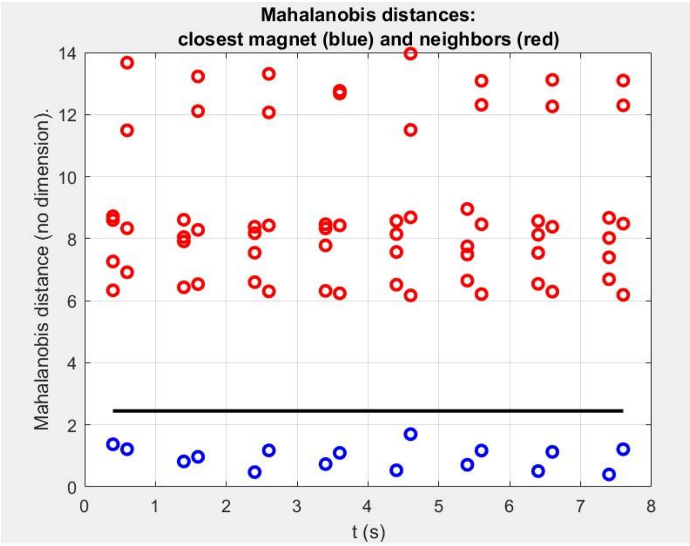
Tuning value 0.01

If Y – Ŷ were Gaussian variables we would have 5% of the measurements to be rejected.

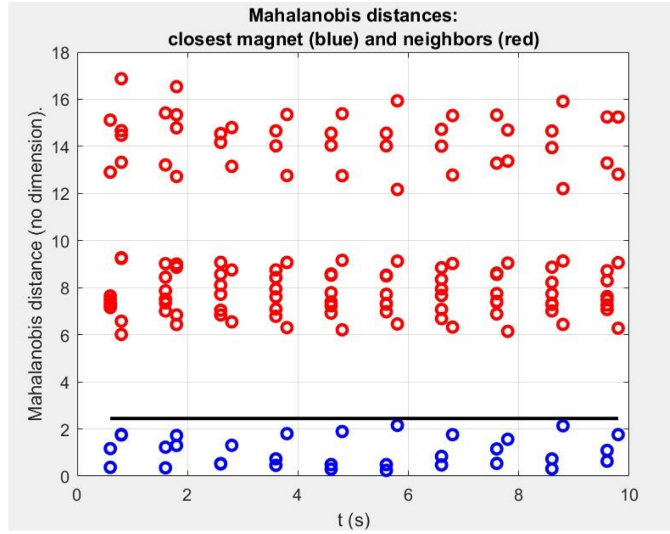
In our code, we have set 0.95% to compute the Mahalanobis threshold (using chi2inv(..)), it means we would have some magnets rejected most of the time, which is less than the expected 5%. This happens because Y – Ŷ is not Gaussian, but it has a uniform distribution.

An important characteristic of this distribution is that large errors have zero probability to happen, this means it is normal that we do not obtain 5% rejection. It is not a problem to reject a few measurements, while it would be if too many measurements were because the robot would end up lost (incorrect odometry)

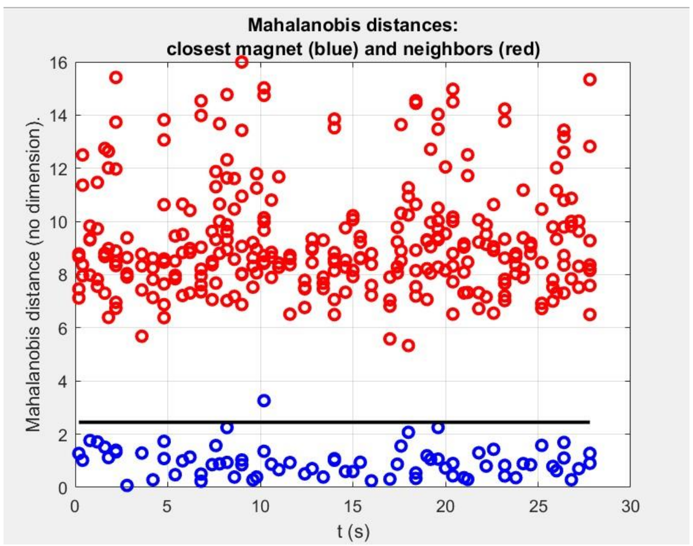
We have also verified this value of 0.1 to be suitable for all the different paths.



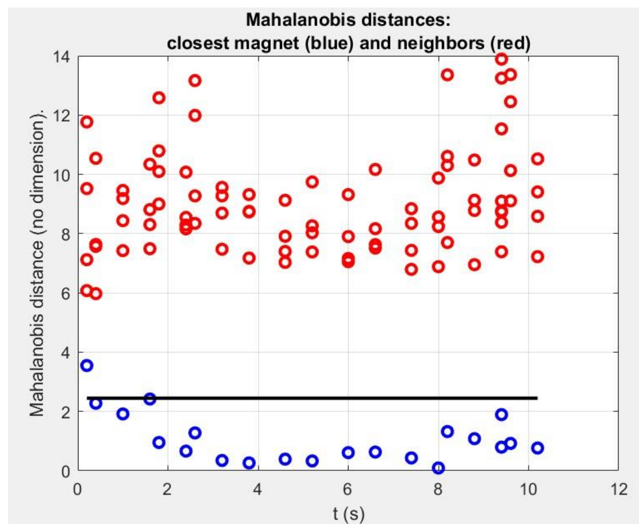
line1magnet.txt



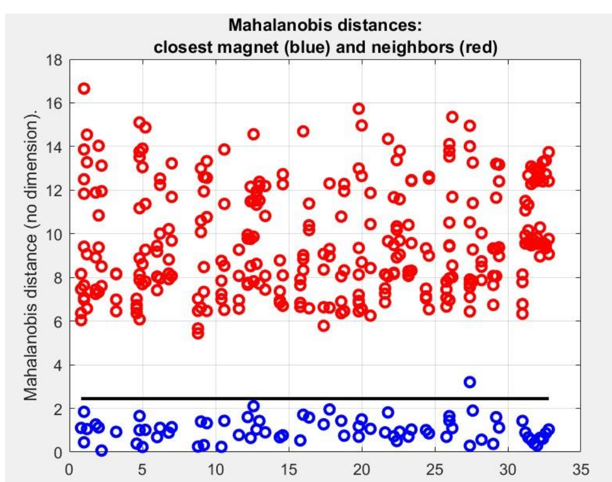
line2magnets.txt



circles.txt



diagonal45degrees.txt



oneloop.txt