Obtain C

$$C = \frac{\partial g}{\partial X}(\hat{X}_{k+1/k})$$

④ find g: $Y = g(X)$

„output as function of state"

The sensor measures the two coordinates of the beacon in the robot frame.

$$g(x) = {}^{m}M$$ measurement in frame m

$$= {}^{m}T_{0}\,{}^{0}M \quad ; \quad {}^{0}M = \begin{pmatrix} x_{II} \\ y_{II} \\ 1 \end{pmatrix}$$

$${}^{m}T_{0} = {}^{0}T_{m}^{(-1)} \quad ; \quad {}^{0}T_{m} = \begin{pmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ 0 & 0 & 1 \end{pmatrix}$$

$${}^{0}T_{m} = \begin{pmatrix} \cos\theta & \sin\theta & -x\cos\theta - y\sin\theta \\ -\sin\theta & \cos\theta & x\cdot\sin\theta - y\cos\theta \\ 0 & 0 & 1 \end{pmatrix}$$

$$g(x) = \begin{pmatrix} \cos\theta\cdot x_{II} + \sin\theta\cdot y_{II} - x\cdot\cos\theta - y\cdot\sin\theta \\ -\sin\theta\, x_{II} + \cos\theta\cdot y_{II} + x\cdot\sin\theta - y\cdot\cos\theta \\ 1 \end{pmatrix}$$

$$\frac{\partial g(x)}{\partial X} = \begin{pmatrix} -\cos\theta & -\sin\theta & -\sin\theta(x_{II}-x) \\ \sin\theta & -\cos\theta & \cos\theta(y_{II}-y) \\ \cancel{0} & \cancel{0} & \cancel{0} \end{pmatrix}$$

$$X = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$$

$$\text{oRealMagnet} = \begin{pmatrix} x_{II} \\ y_{II} \end{pmatrix}$$

MOBRO-LAB: Report 1

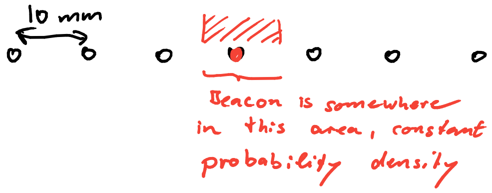
name: Filip Hesse

ID: 488 9393

Compute Std-dev of measurement in y

We observe the characteristics of the sensor for estimating σ_y. When one reed sensor detects a magnet, then its location must be somewhere in the range of [-5 mm, 5 mm], where each location is equaly probable.

=> uniform distribution



Set $\mu_y = 0$ and $f(y) = \begin{cases} 0 & \text{for } y < 5 \\ 0{,}1 & \text{for } 5 \leq y < 5 \\ 0 & \text{for } \geq 5 \end{cases}$, where $f(y)$ is the probability density function of y

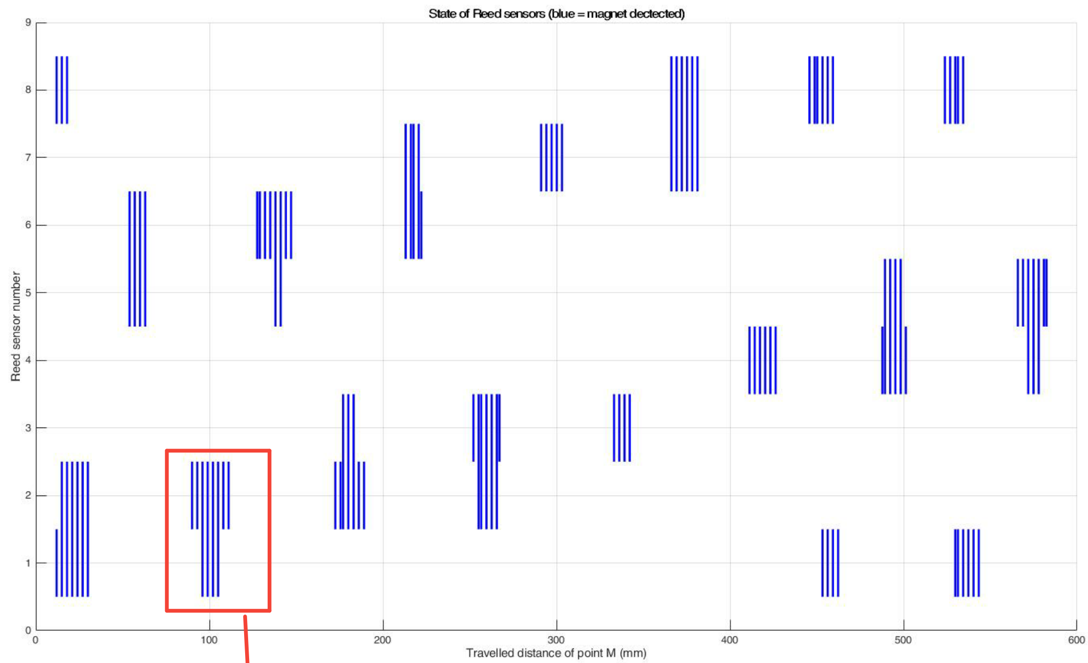
$$Var(y) = \int_{-\infty}^{\infty} (y-\mu_y)^2 \cdot f(y)\, dy$$

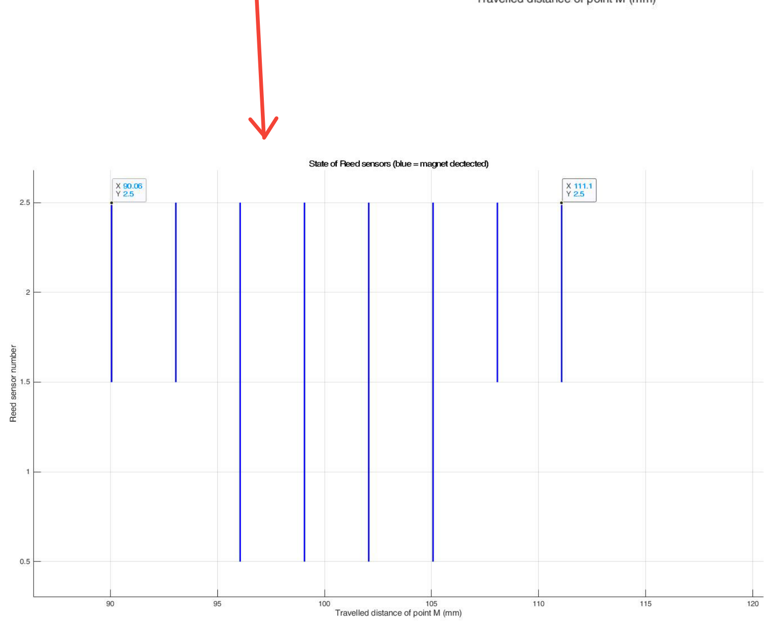
$$= \int_{5}^{5} y^2 \cdot 0{,}1\, dy = \left[0{,}1 \cdot \frac{1}{3} y^3\right]_{-5}^{5} = \frac{1}{30} \cdot (125 + 125) = 8{,}\overline{3} \text{ mm}^2$$

$$\sigma_y = \sqrt{Var(y)} = 2{,}887 \text{ mm}$$

Compute Std-dev of distribution in x

For computing σ_x, we can look at the distance travelled while detecting the same sensor. We can assume, that the true sensor position is in the center of this area and the error around it is uniformly distributed. Optimally, we would need to take into account every single measurement of many different trajectories, but it is much easier estimating a worst case uncertainity, which will be a slight over-estimate of the true one





The variance can be computed just like in the previous case or simply by the formula

$$Var(x) = \frac{1}{12}(b-a)^2 = \frac{1}{12}(111{,}1 - 90{,}06)^2$$

$$= 36{,}89 \text{ mm}^2$$

$$\Rightarrow \sigma_x = \sqrt{36{,}89} = 6{,}07 \text{ mm}$$

MOBRO-Lab : Report 2

Filip Hesse
488 9393

To define a threshold for the Mahalanobis distance, we use the function chi2inv(0.9, 2).

Explanation: The innovation can (simplyfying a bit) be assumed as normally distributed in dimension x and y. Under this assumption, the mahalanobis distance is chi-squared distributed. This is why we need an inverse function of the chi-square distribution to define a threshold for mahalanobis distances.
Our dimension of the measurement (and innovation) is 2 (x and y). We pick a relatively high value for the percentile of the threshold (0,9), because there is no influence on the reed sensors other than the actual magnets. So the probability of a false positive measurement is very low.

MOBRO Lab - Report 3

Filip Hesse
4889395

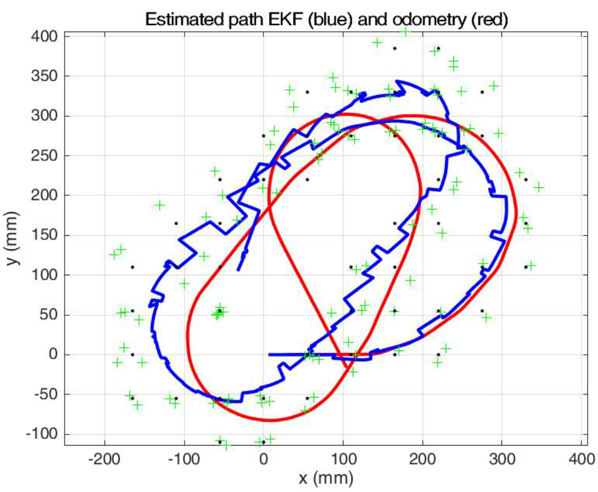
For detemining sigmaTuning, I used the path „TwoLoops". The value necessarily needs to be a very small value, because it is an input standard deviation, which is taken into account at each step, so each 0,2 s (5 Hz).

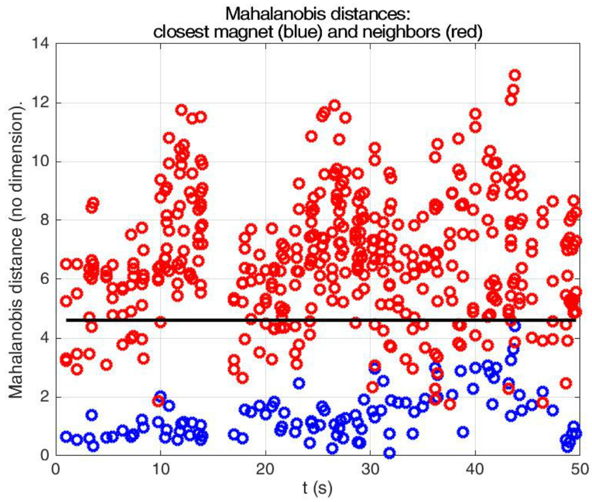
I try out different values for sigmaTuning and look at the estimated path, which is supposed to end at the same coordinates, where it started. I also evaluate the plot with the Mahalanobis distances, because it shows how definite/safe the choice of the corresponding magnets is.

After trying out different values, I decided sigmaTuning = 0,03 is a suitable value.

The following graphics show the paths and Mahalanobis distances for a too big value, for a too small value and for the correct value:

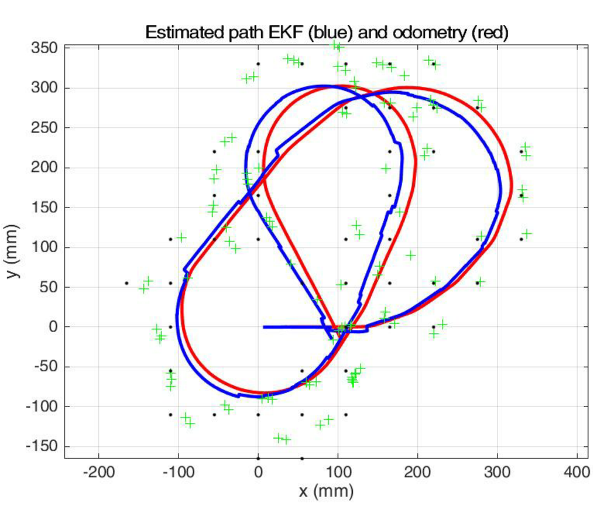
(1) Sigma Tuning = 0,15 => TOO BIG

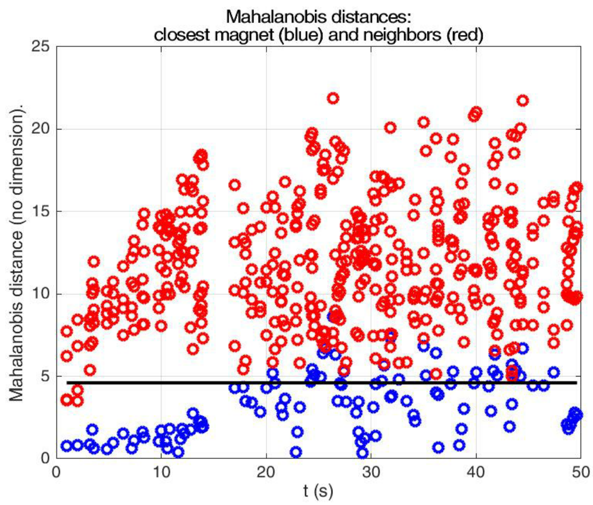




The path goes too far away from the odometry because some wrong choices about the corresponding magnets have been made. The path does not end in the starting point. The Mahalanobis distances of neighbor magnets are not always clearly separated from the distances of the closest magnet.

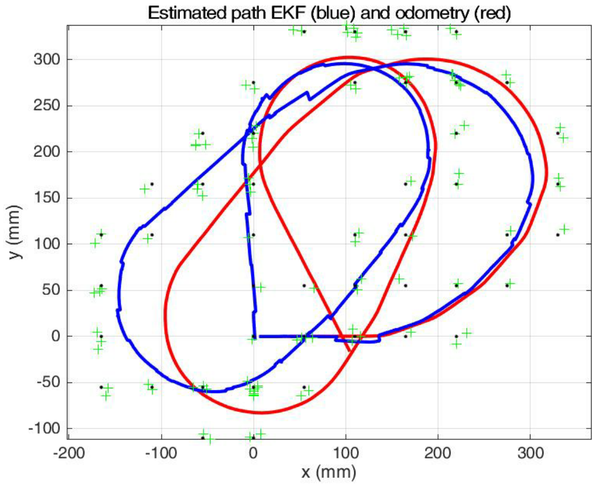
(2) Sigma Tuning = 0,01 => too SMALL

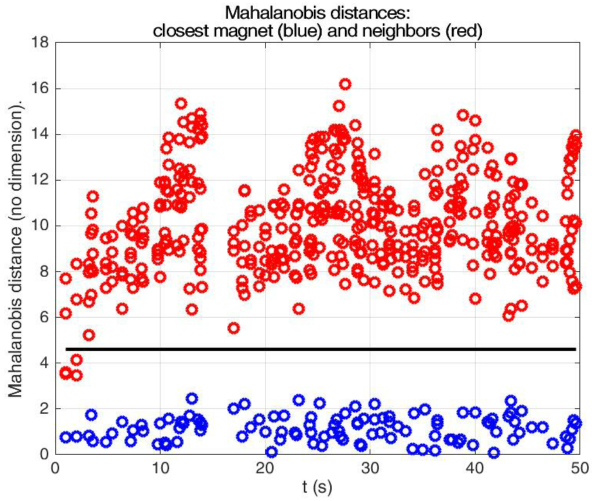




The path stays too close to odometry.

SigmaTuning = 0,03 => correct Value





The path ends at the starting point and the Mahalanobis distances are clearly separated

=> value works fine