Davide Torielli Fabio Fusaro

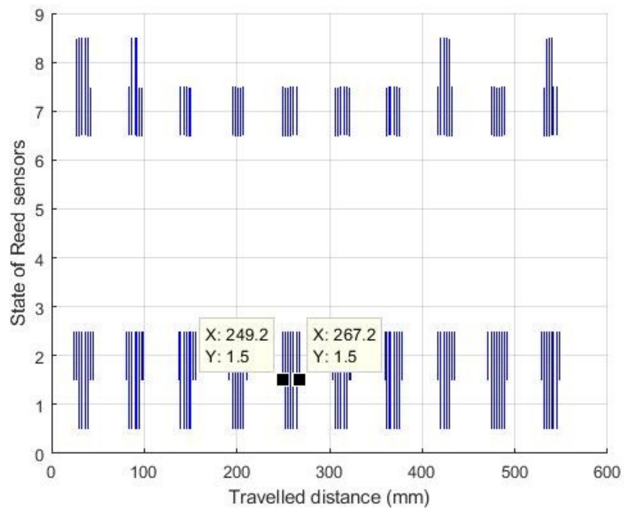
Localization with Kalman Filter

Introduction

The aim of this lab is to understand and studying a Kalman filter for a wheeled robot with magnet sensors which performs certain paths on a plane.
First thing is to create the EvolutionModel.m script which describes the predict states with equations of odometry.
Then we put A, B and C matrix in the MagnetLoc.m, according to our evolution model.

Setting the parameters

Measurement noise (Q_{gamma})
We set all other variances to zero to not have any other influences on this error.
In RobotAndSensorDefinition.m we increase the sampling frequency of the sensors so they detect more frequently the magnets and so it is simpler to understand this variance.
The distribution of error is uniform because each value of the error appears the same time of other values. The variance of this kind of distribution is $(b\text{-}a)^2/12$ where b-a is the interval in which the sensor detects the same magnet.



For $sigma_X$ we use the line2magnets path because in this we have a straight line of magnet in y = 0 while the sensor 2 is at y=2 for the whole path (because the robot moves horizontally). Thanks to this we (almost) can understand how much is the diameter along x in which the magnet is detected by sensor 2. The diameter ranges b-a for the consecutive magnet are all different because nothing is perfect, so we take all ranges in the line and make a mean. The variance find is 29, and so $sigma_X$=5.4

For $sigma_Y$ the reasoning is the same: we know that the distance between 2 sensor is 10 mm vertically, so each sensor can detect a magnet with maximum +-10 error. So b-a is 10 and the wanted standard deviation $sigma_Y$ is 2.9.
Due to the fact that measures are approximated and magnet and sensor are different each other, it isn't important that these values are so precise.

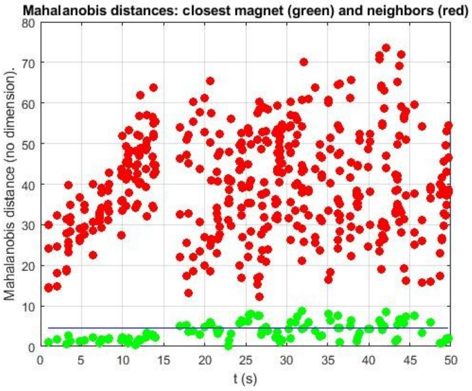
Initial condition uncertainty (P_{init})

These errors are due to the fact that the robot does not start precisely in the initial position wanted (that is the error an operator makes when he put the robot in the chosen initial point), so the distribution is Gaussian.
We set sigma = 2mm for x and y which means that 95% times the error has a maximum value of +-4mm and 99.7% times has a maximum value of +-6mm;
the same for the error in degrees for orientation but we must remember that we have to use radians so $sigma_{theta}$ = 2*pi/180.

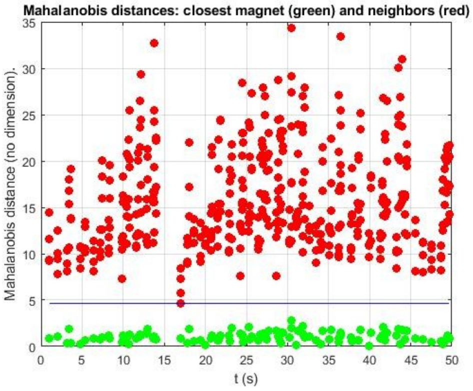
Input Noise($Q_{ß}$)

We have to put a value for $sigma_{wheels}$, 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, that is good also because we are using a trial and error method.
We use a "trial and error" method because it is the only method for us to estimate this parameter.
Having set all the other variances we have to find one suitable for $sigma_{wheels}$:
We set this value such that in the plot of Mahalanobis distance the green dots (closest detected magnets) are well distanced from the red dots (neighbours).
The Mahalanobis distance depends on the innovation term $Y - \hat{Y}$, that is the error between the real position of the magnet and the estimated one; it takes into account also the measurement noise Q_{gamma} and the P matrix. P is the error propagation which depends also on Q_{beta} (the one we are calculating) and Q_{alfa} , that is the noise on the state (set to zero for the entire experiment).
With too low value we may discard some magnet that are good (green dots) because we are too much optimistic on the wheels, and the Mahalanobis distances increase.
With too high value we accept detections of magnet that are neighbours (red dots) and not real detected magnet. In this case we are too pessimistic on the wheels and a lot of red dots are under the Mahalanobis threshold.

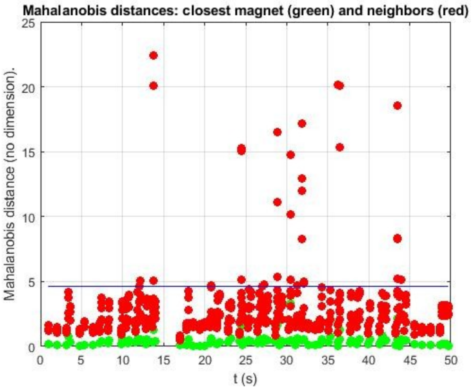
We found a suitable value as 0.1



sigma = 0.01



sigma = 0.1

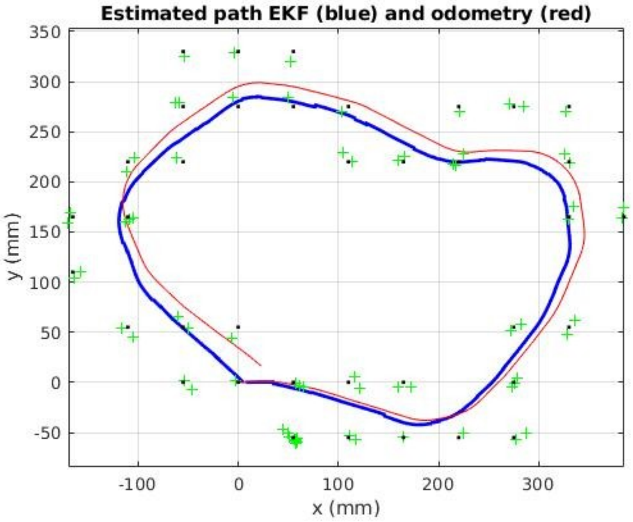


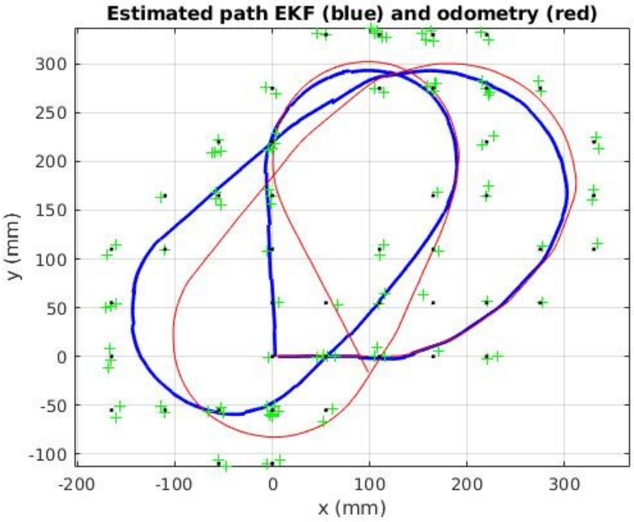
sigma = 1

Mahalanobis threshold

We put this value with chi2inv(0.90, 2) which computes the inverse cumulative distribution function with 2 degree of freedom (the two measures: x coordinate and y coordinate) for the corresponding probability in 90%.

Kalman Filter Result

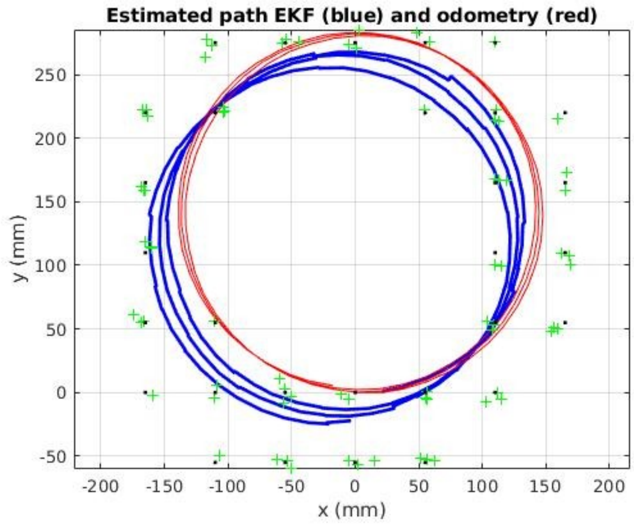


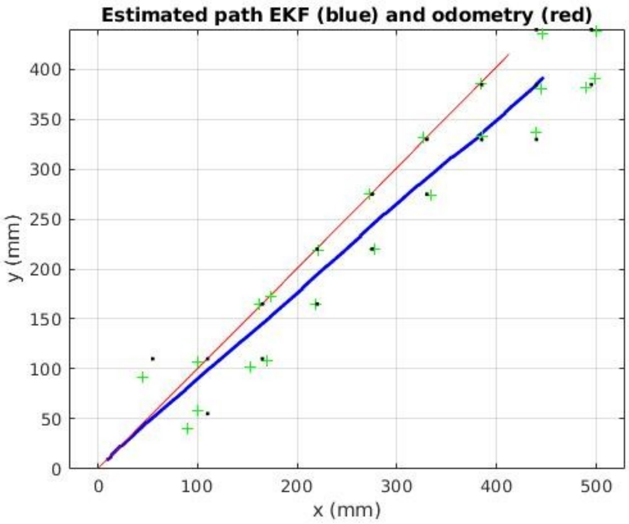


These images show how good the Kalman filter is: the final position is almost the same as the initial position, instead the odometry is clearly not good as well. Corrections on blue trajectory are visible: they are due to the fact that a magnet is detected and the robot path is instantly modified.

With greater value of $sigma_{wheels}$ (here is at 0.1) these corrections are more evident because we are less confident on the wheels so the influence of the magnet detection on the trajectories must be bigger.

So with low value of $sigma_{wheels}$ the blue trajectories are more similar to the red ones.



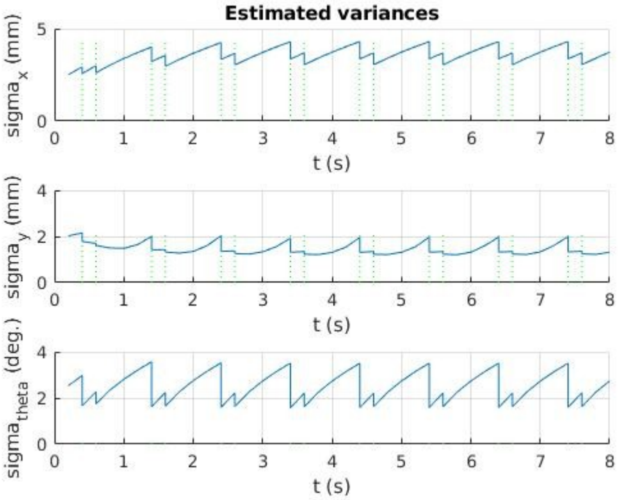


In others trajectories the Kalman filter is still good: the odometry seems better but we must remember that it is only a reference, while the Kalman path is (almost) the real one followed by the robot.

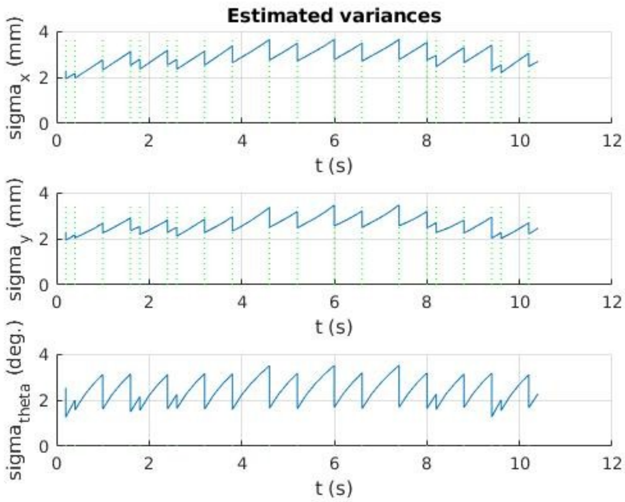
Note that in the circles the path is shifting on the left for the wheels slippage, due to the fact that we are performing circles counter-clockwise. Odometry trajectory is only a reference so it does not take into account this.

Compare variances on line1magnet and diagonal45degrees

Line1Magnet

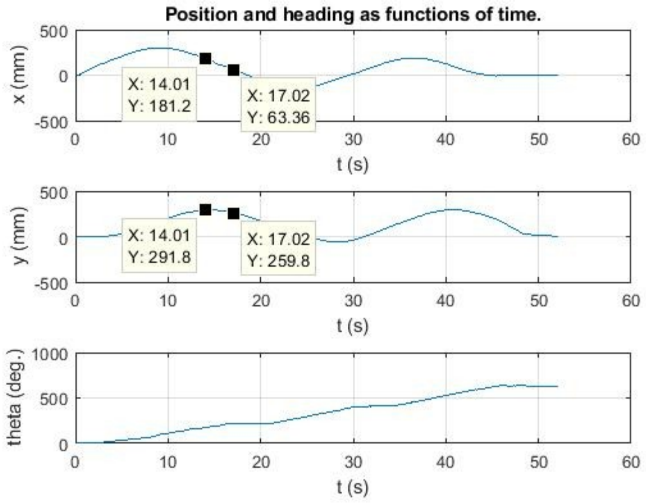


Diagonal45degrees

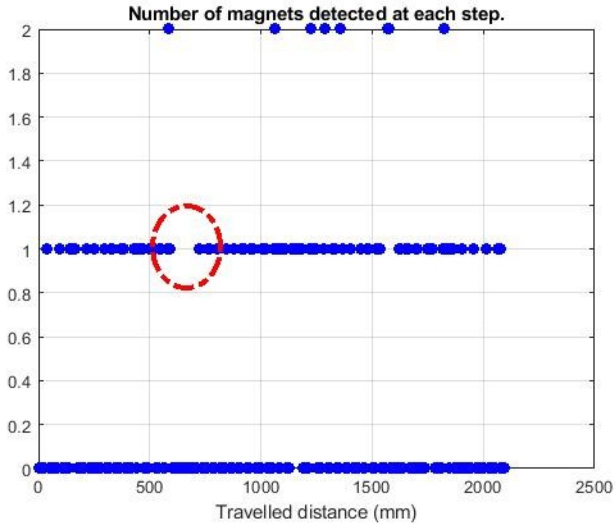


Here we see some sort of “sawtooth” curve. This because errors increase while robot goes on by odometry, but in the instant when the robot detects a magnet (green vertical dotted lines) the position is corrected and so the variances instantly decrease a bit.
In diagonal45degree the variances on x and y are the same because going 45 degrees we perform same distance on x and y.
In line1magnet the direction of the motion is only on x so they are different.

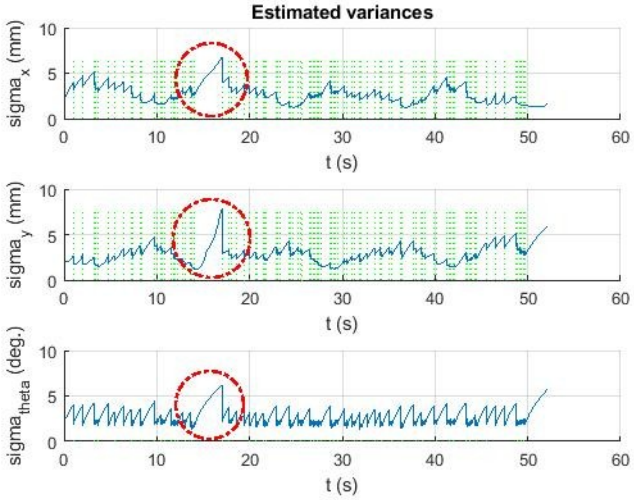
TwoLoops between 14 and 17 seconds

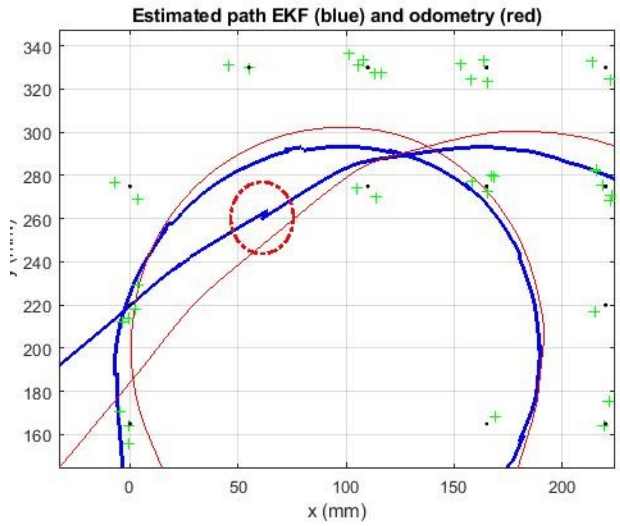


From the plot above, we see what the positions of the robot from 14 to 17 seconds are:
14s : [181mm, 291mm, 173°]
17s : [63.36mm, 259mm, 210°]



In the above graph we see that the 14-17 is a long period where no magnet are detected. This causes the robot to go on with only odometry and no correction by the Kalman filter.





In fact we see above how much the estimated variances increase in this period:
After a magnet is again detected the correction is strong: in the estimated variances we see a big instantly decrease, and in the path plot a evident zigzag section is visible.

Sensor with only one measure

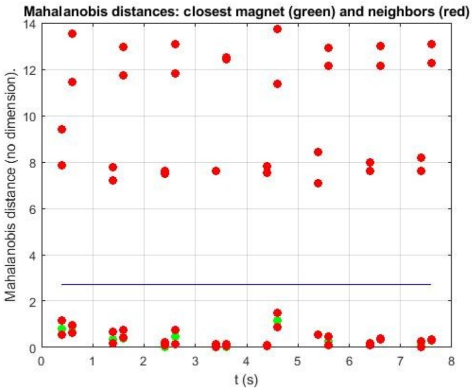
Changes respect to original model

Now we assume we have the robot sensors with only one measure, we choose to keep the one on y because it is the one more reliable (the variance of the error on y is lower). So the measure on x is no more taken into consideration.
The model has to be changed:

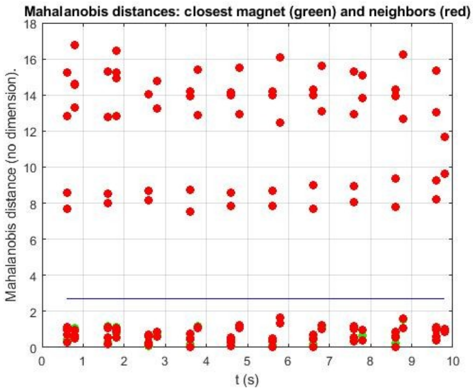
- In magnetLoc2.m now C has only the second row and so innovation term is only a scalar.
- In defineVariances2 Qgamma is a scalar and no more a 2x2 matrix, and for Mahalanobis threshold the chi2inv has 1 as second argument.
- SigmaWheels hasn't to be changed because the wheels remain the same. We may look for better estimation to reduce the overall errors, but it is useless: for example in the line1magnet the red dots are too near the green ones to be well spaced, and in other path this would not improve the things so much

Mahalanobis distances

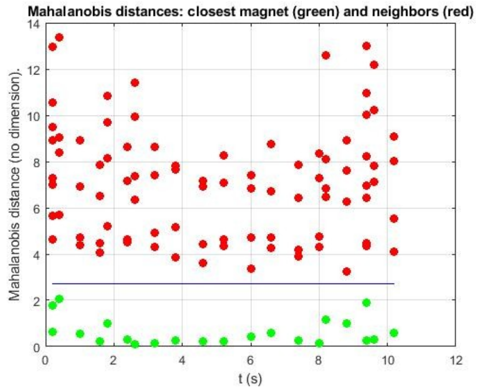
Line1Magnet



Line2Magnet



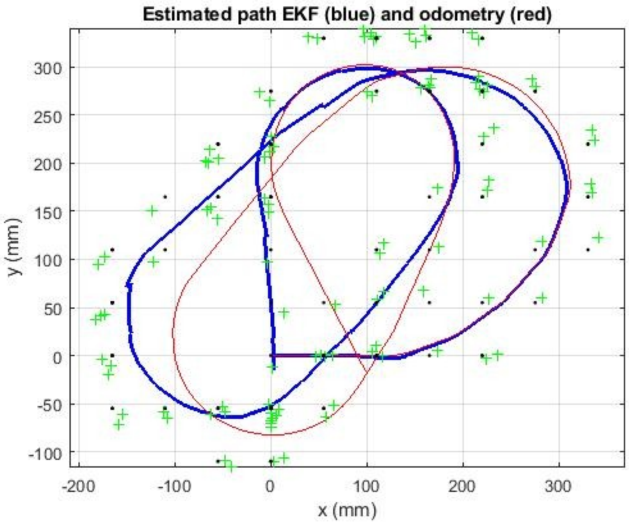
Diagonal45degrees



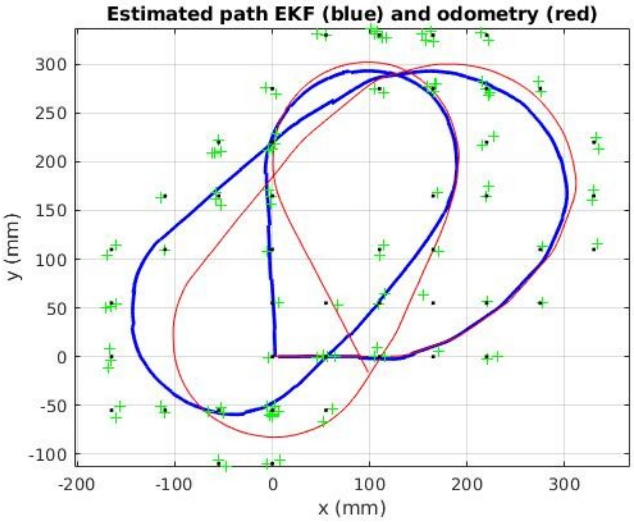
In line1magnet and line2magnets the magnets are along a line (and 2 lines) parallel to y. Some of the magnets near the one detected have a little Mahalanobis distance, too little to be differentiate with the green dots. This is because the measure is only along y so the x distances (that are the only which changes because y coordinates are always the same) are not taken into considerations.
In the diagonal45degrees, instead, the y coordinate of the magnets changes so the computations of Mahalanobis distance are more reliable with correct discharged neighbours (red dots) (but obviously worse than the version with 2 measures).

Results with one measure

Only one measure

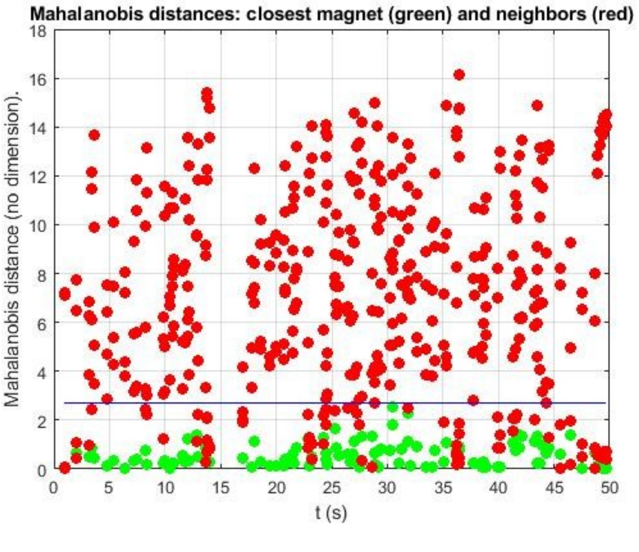


Two measures

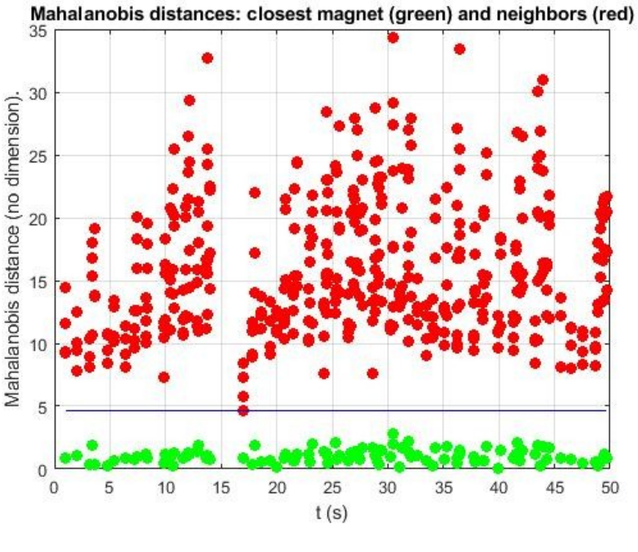


With only one measure the path are similar to the version with two measures.
The only path with visibly bad changes is the most difficult one: the 2 loop (we see that the final position is further from the initial).

Only one measure (2loop)

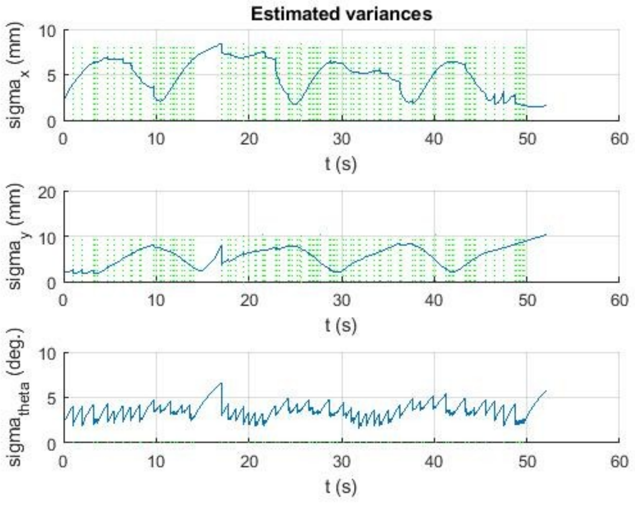


Two measures(2loop)

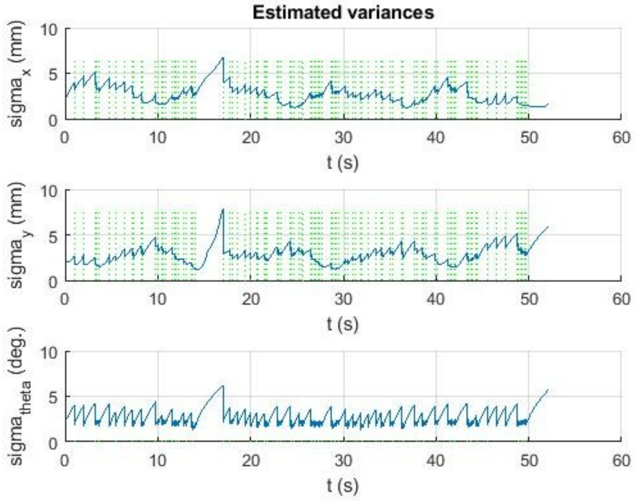


Mahalanobis plots are worse but we see that this worsens only a bit the path.

Only one measure (2loop)



Two measures(2loop)



Regarding variances, we note that in general they are greater with only one measure. This is obvious because we have less information.
Also, this is due to the fact that at every magnet detection the correction is much less strong because we are less confident, having only one measure.