Contents

1 Introduction

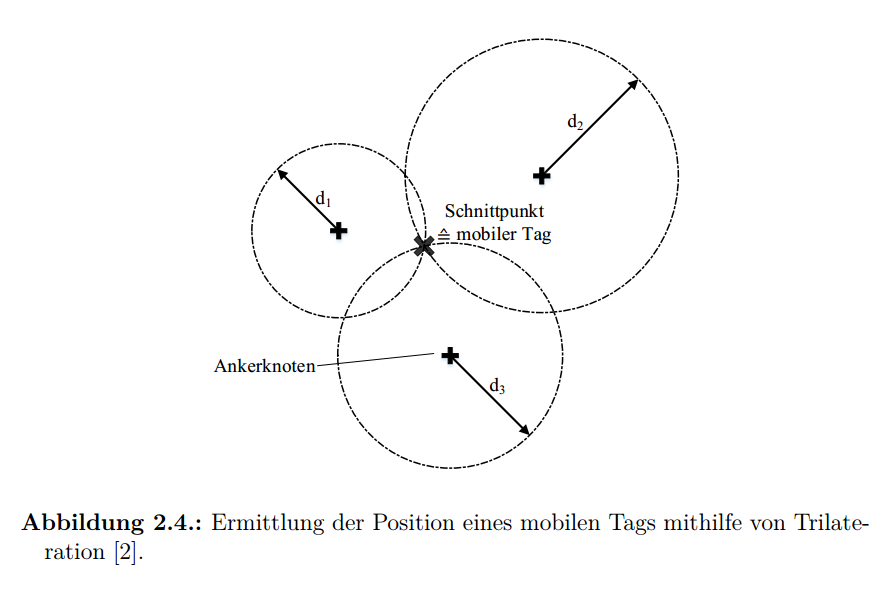
2 Theoretical Background

2.1 localization with distances

In Geometry, the position of a point can be determine by several methods, such as Trilateration and Triangulation. In this thesis, Trilateration is used for location determination. Followed is a brief introduction about Trilateration:

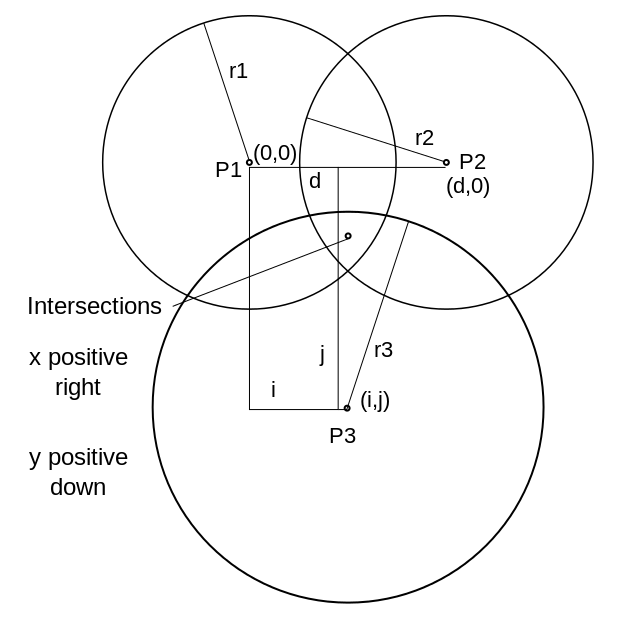
In a two dimensional space, to determine the position of a point, at least 3 distances to 3 different fixed positions are required(these 3 fixed positions should not lie in a same line).

When the exact distances from a mobile Tag to 3 anchor Nodes are known, the position of this mobile Tag can be determined by the intersection point of circles around the Nodes with corresponding radius.



(pic from ‘Patrick thesis P11’)

When the distances from a mobile Tag to 3 anchor Nodes are added with noise due to the inherent error of hardware, the position of this mobile Tag lies within the intersection area of circles.

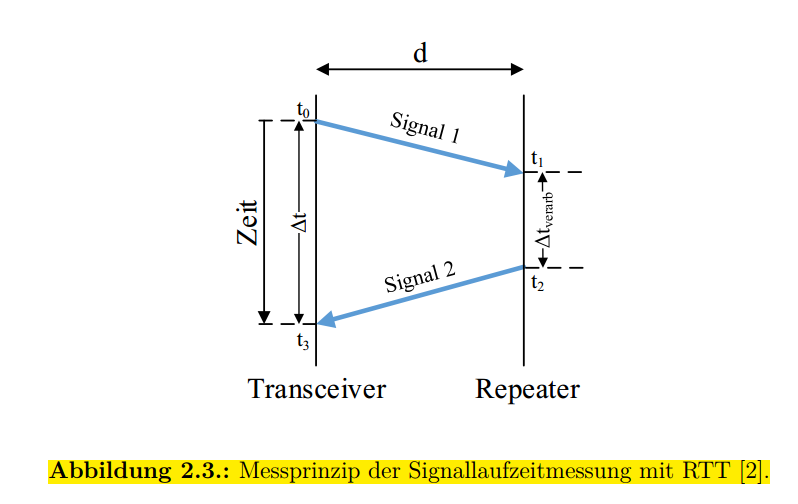


(pic from ‘https://en.wikipedia.org/wiki/Trilateration’)

2.2 distance measurements base on UWB signal

2.2.1 UWB

2.2.2 distance measurements



(pic from ‘Patrick thesis P10’)

For simplicity reason, in the following part of this thesis, transceiver 1 is represented as mobile tag, and transceiver as anchor node.

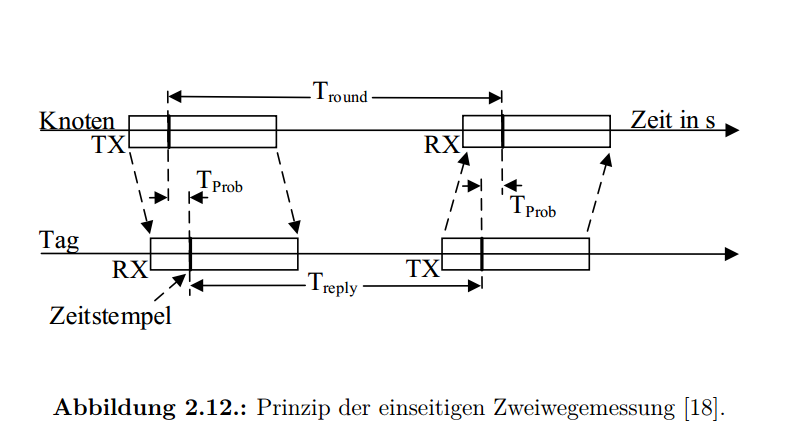
1. implementations of RTT will be introduced below.

Single Sided Two Way Ranging (SSTWR). SSTWR is the basic implementation of RTT. The mobile tag send a message to the anchor node, which responds with another message back to mobile tag to finish the measuring process, as illustrated in XXXX FIG XXX. And each devices record their the transmission and reception  
timestamps of the messages, $T\_round$ is defined as the subtraction of reception timestamp and transmission timestamp in mobile tag, while $T\_reply$ as the subtraction of transmission timestamp and transmission timestamp in anchor node. So the time of flight, $T\_prop$ can be calculated by xxx equ xxx.

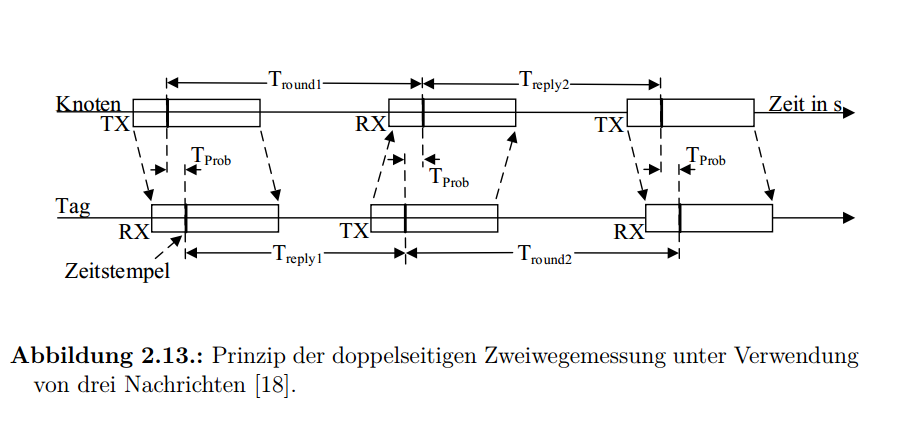
Double Sided Two Way Ranging (DSTWR).

Symmetrical Double Sided Two Way Ranging (SDSTWR).

SDSTWR is a special case of DSTWR where $$ and $$ are required to be equal. Then



(pic from ‘Patrick thesis P24’)



(pic from ‘Patrick thesis P25’)

(pic from ‘Patrick thesis P50’)

2.3 KF

But in the cases of, for example, some of the measurements are missing, or the measurements from different devices have different noise level, or the measurements are too noisy, the basic implementation of \autoref{3\_1} would has difficulty to solve the localization problem. To tacle this difficulty, Kalman filter is used in this thesis.

method like

2.4 EKF

2.5 Self-Calibrations for Anchor-Nodes

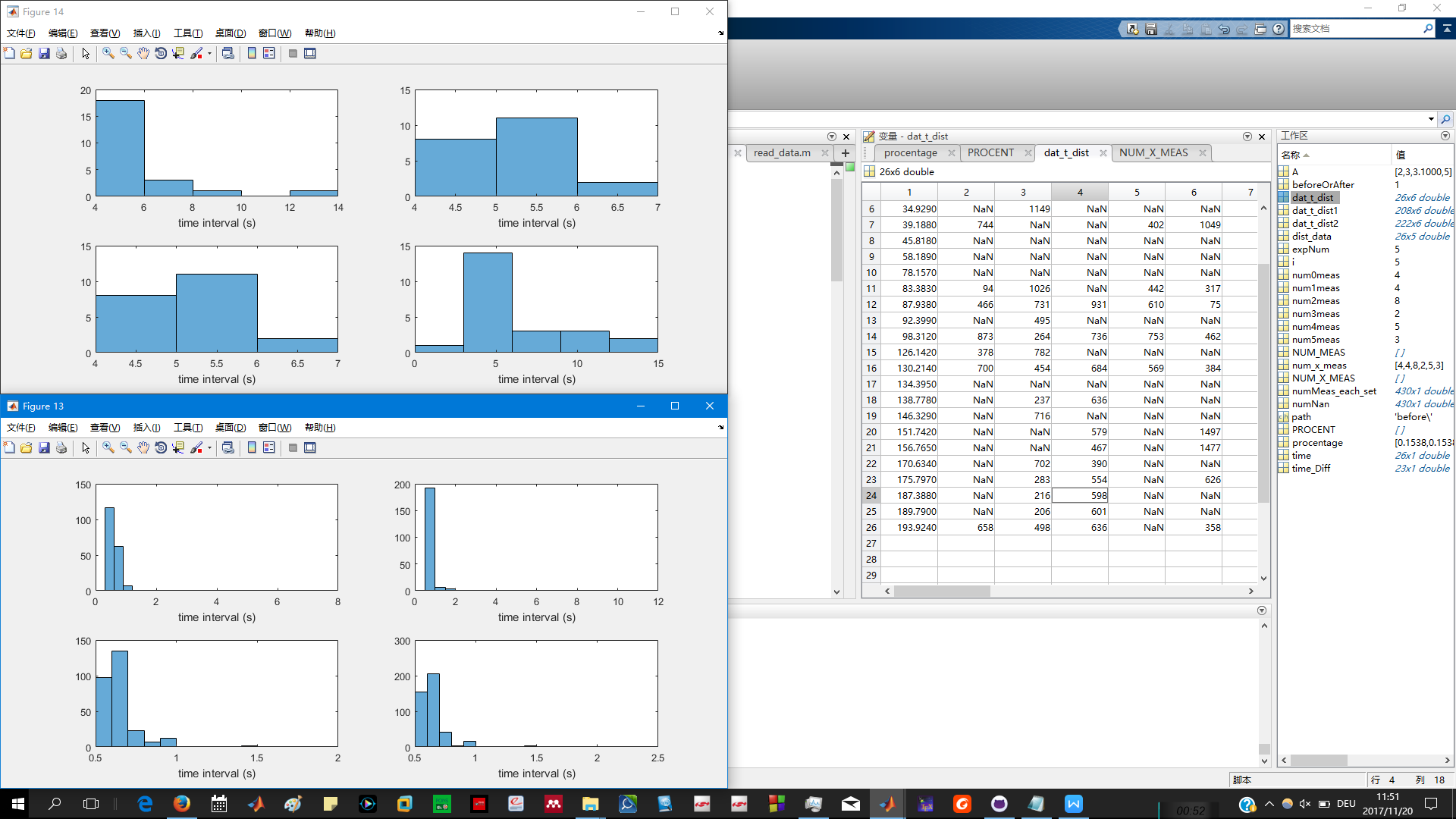
2.6 State of the Art

1. Improvement of hardware

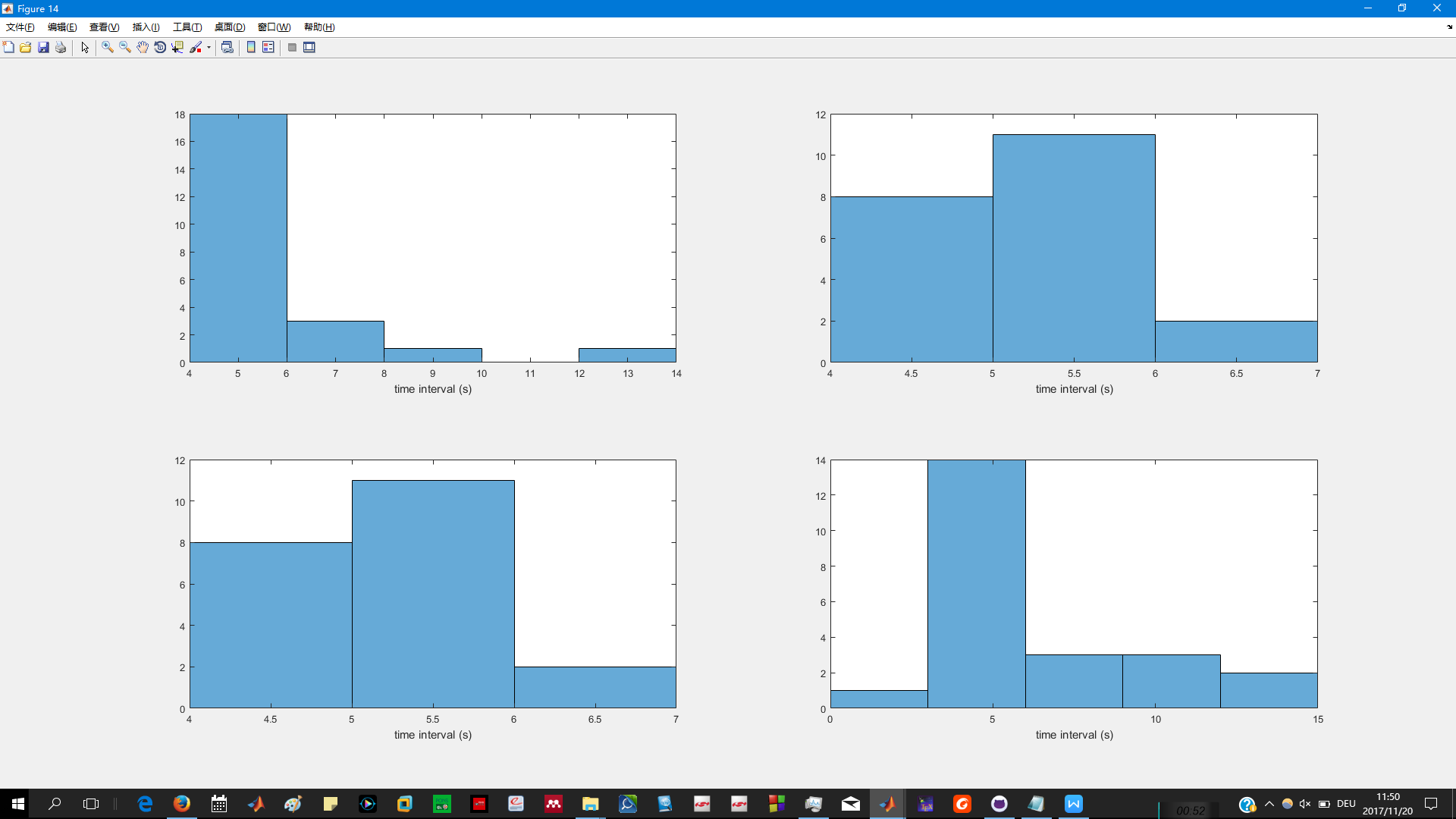
Loop jumper(tag), timing, state machine(loop between nodes)

Sampling rate

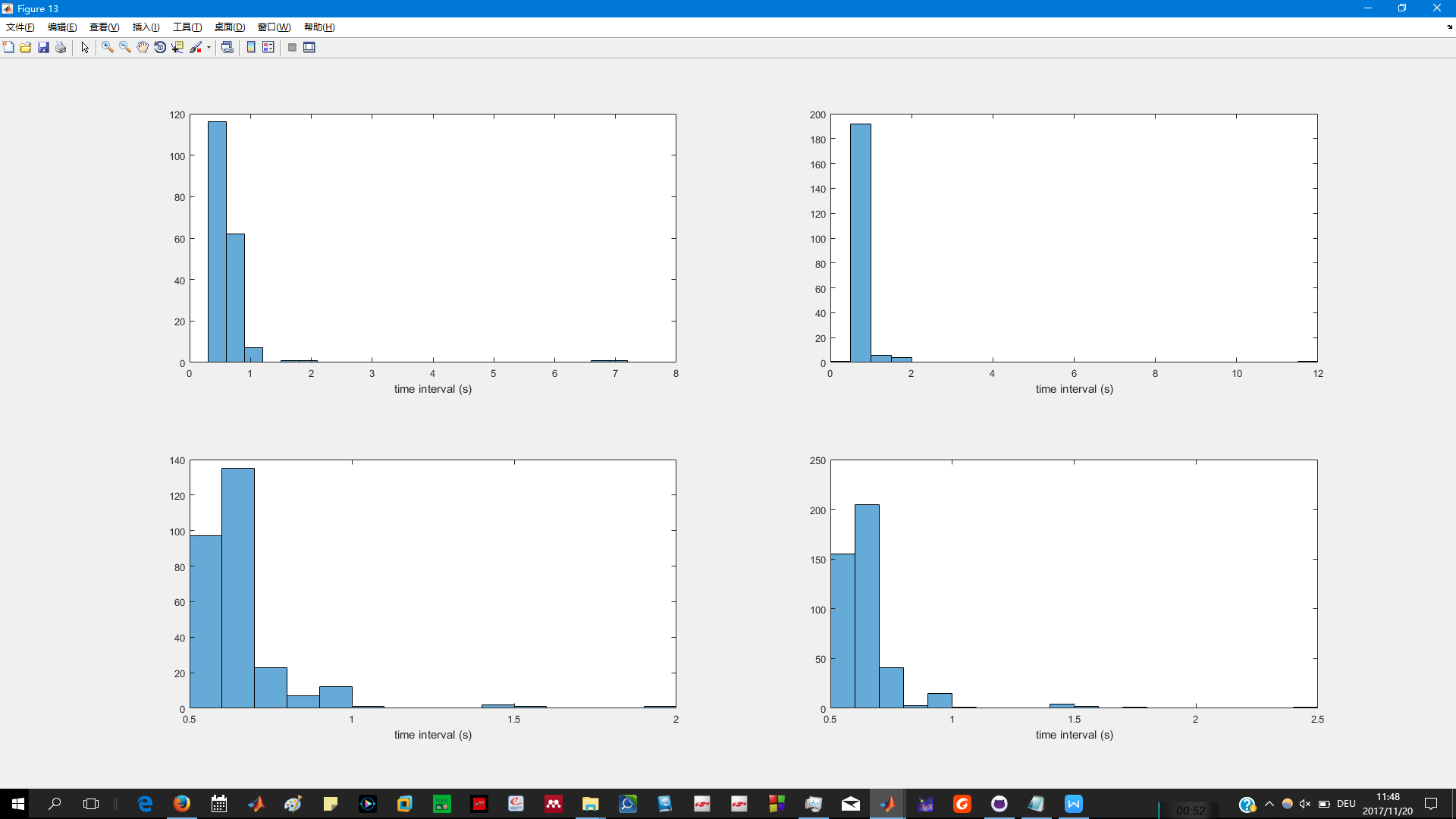
Before and after



Before



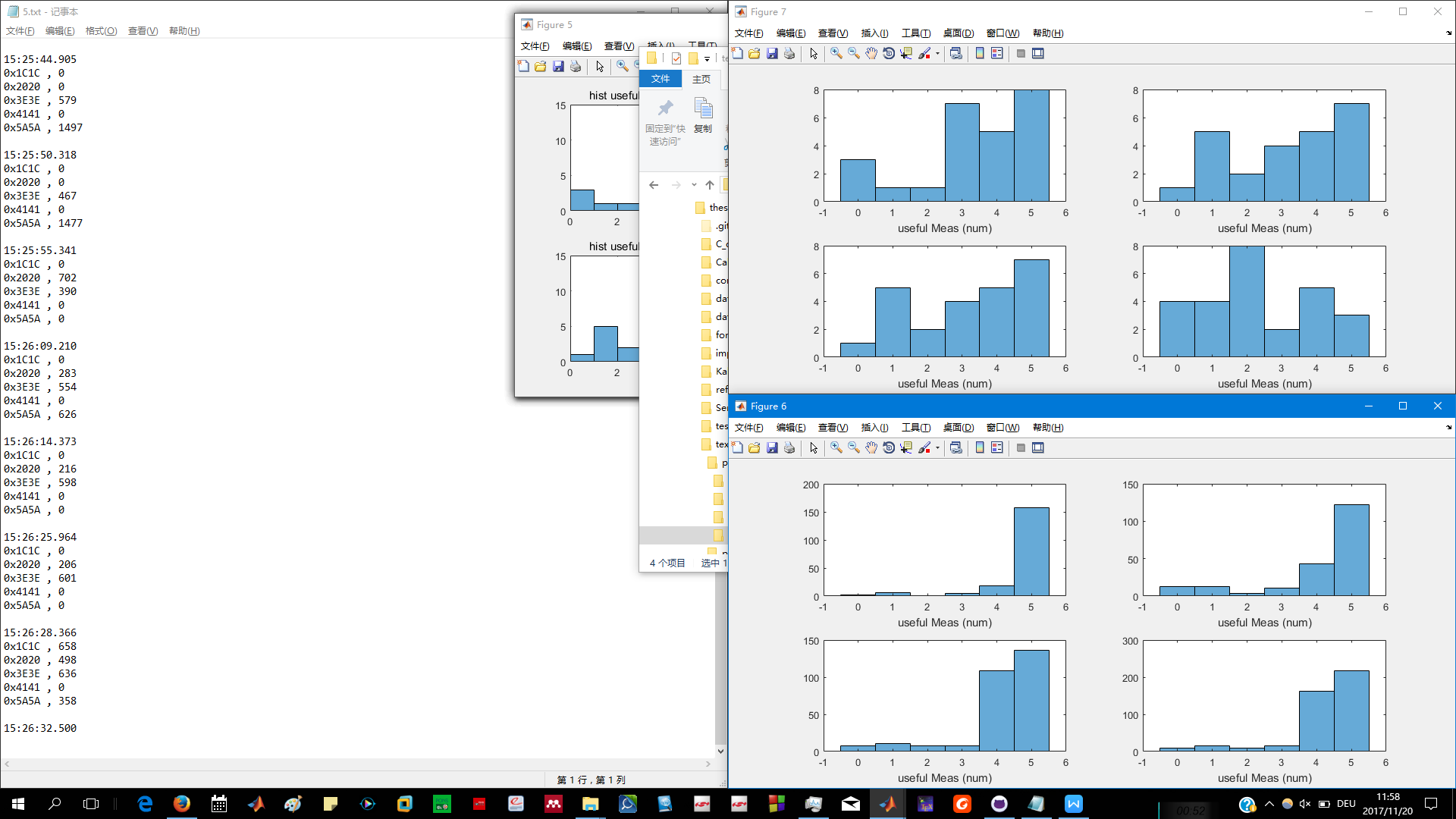
After



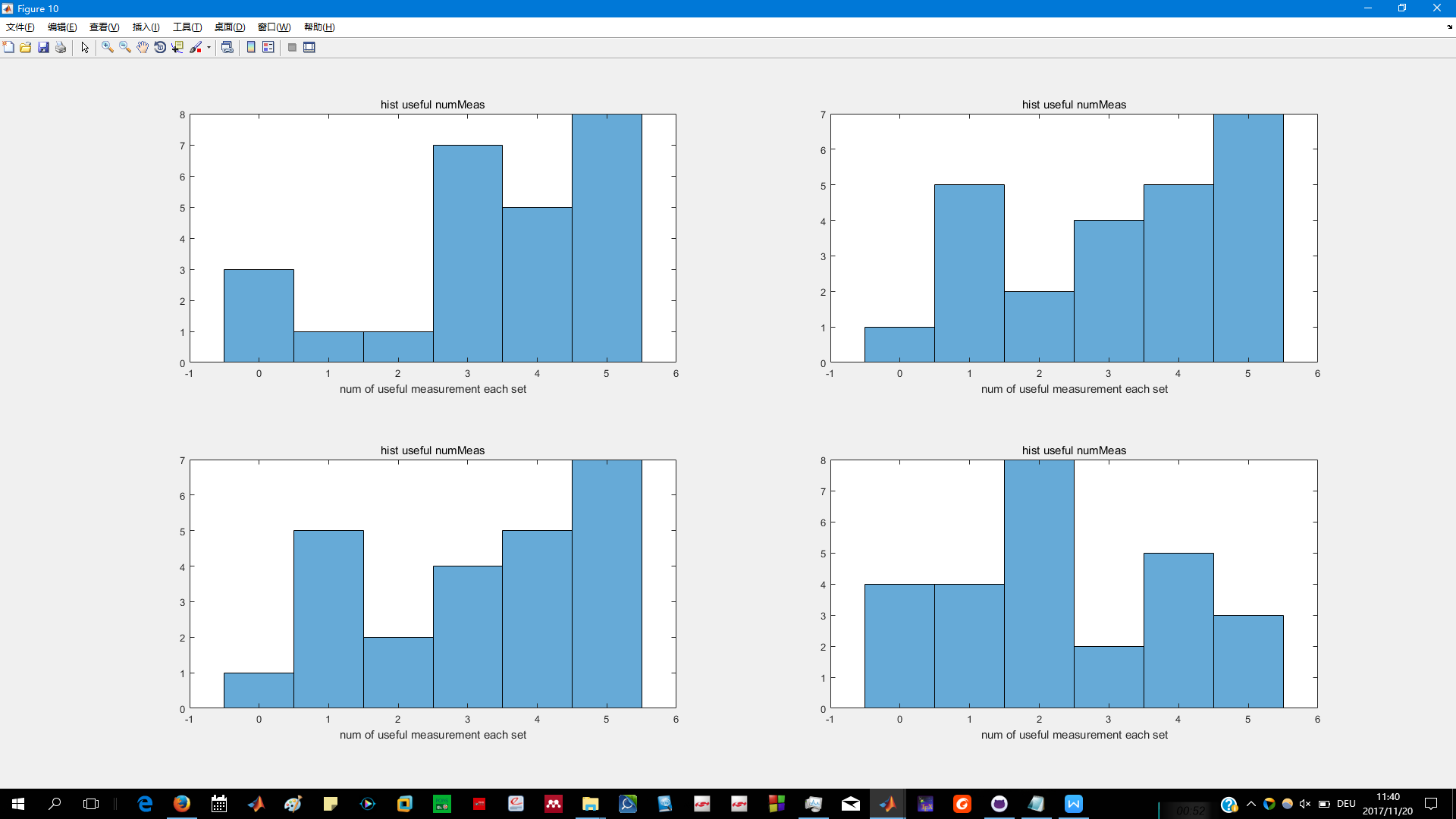
Mat file ‘C:\Users\Yitong\Documents\GitHub\thesis\_indoorLocalization\improvment in hareware’ *hist(time\_Diff,100);*

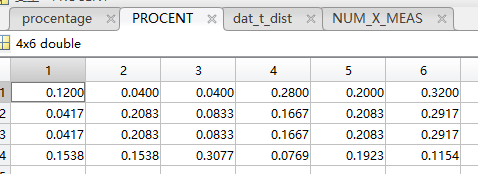
Sampling measurement #

Before and after

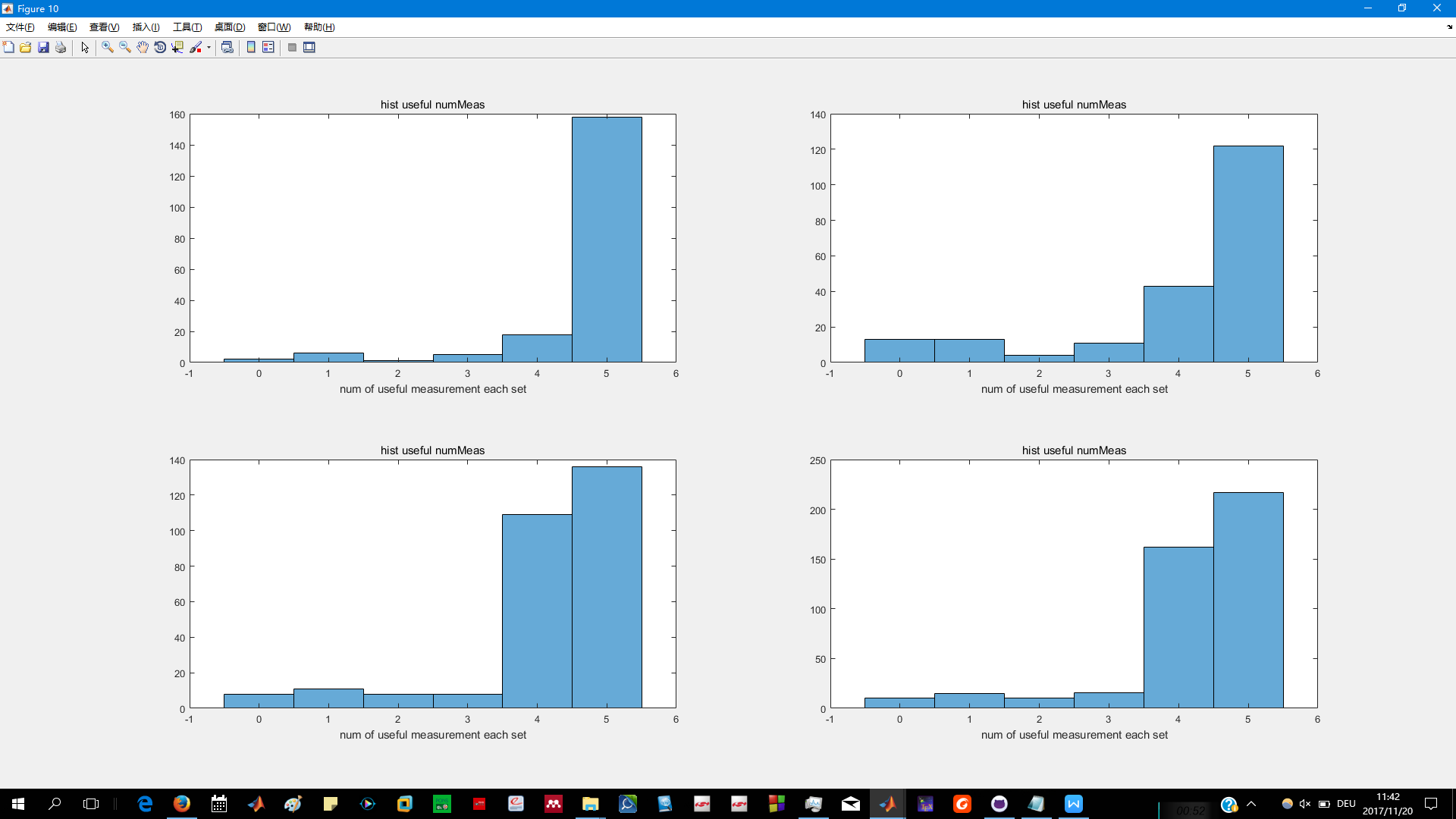


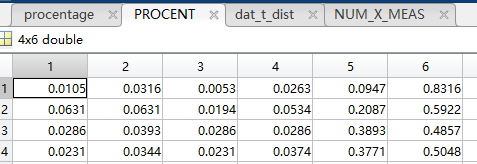
Before



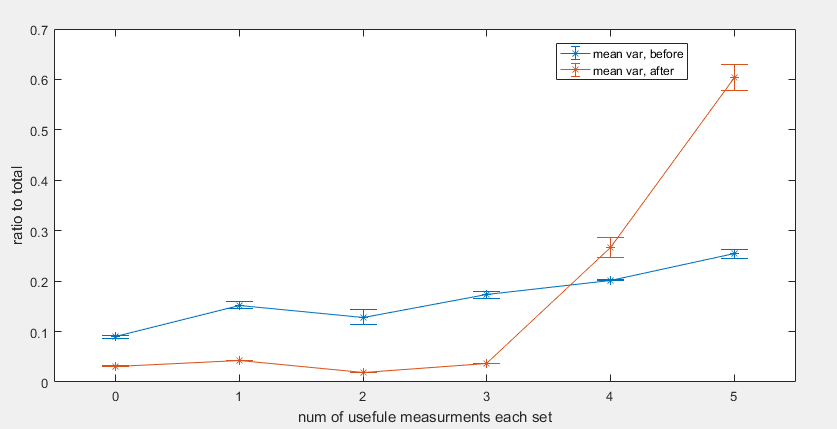


After





Location’C:\Users\Yitong\Documents\GitHub\thesis\_indoorLocalization\improvment in hareware\after’

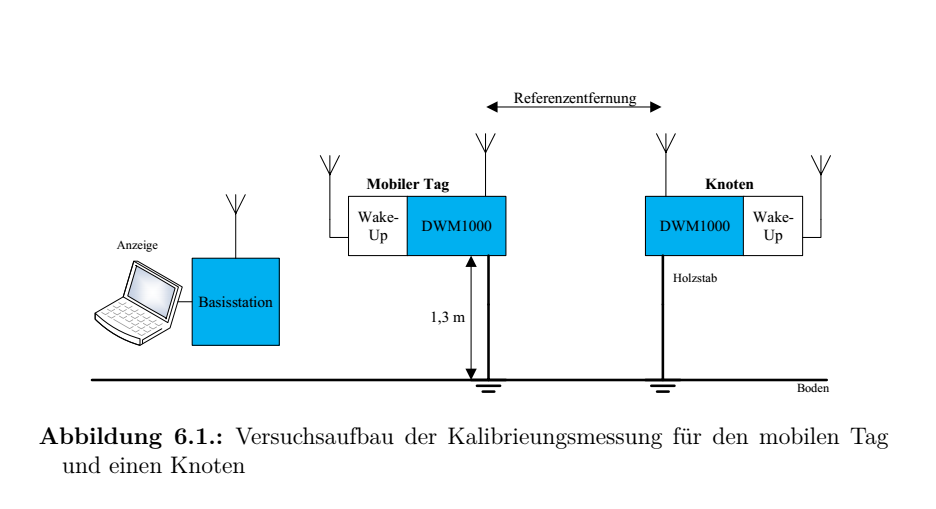


(‘C:\Users\Yitong\Documents\GitHub\thesis\_indoorLocalization\improvment in hareware\compare in numUsafule meas.fig’)

4 Measurements

4.1.1 Calibration

Set up (3,4,5,6,7,8,9m)

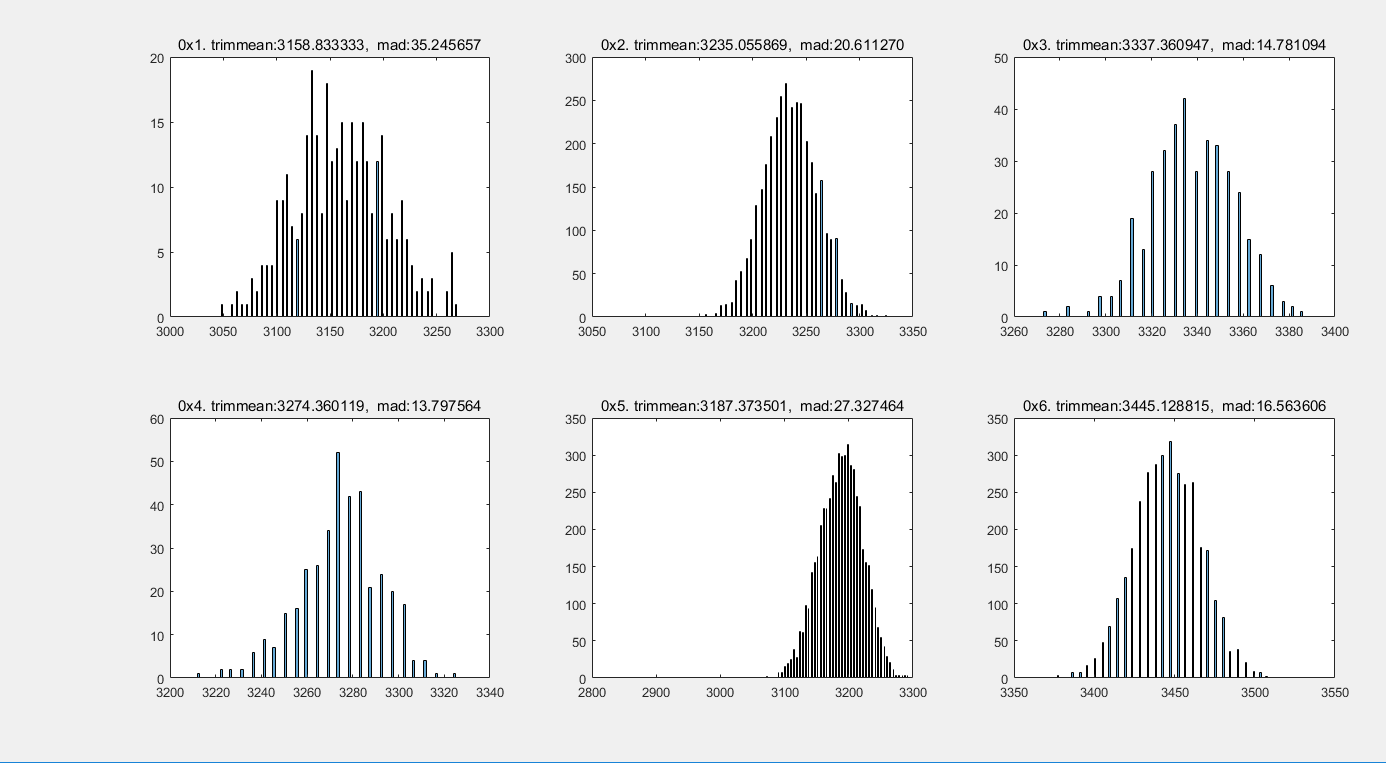


(pic from ‘Patrick thesis P64’)

Measurements Distributions

.................................. heavy tail in front

34568m



...

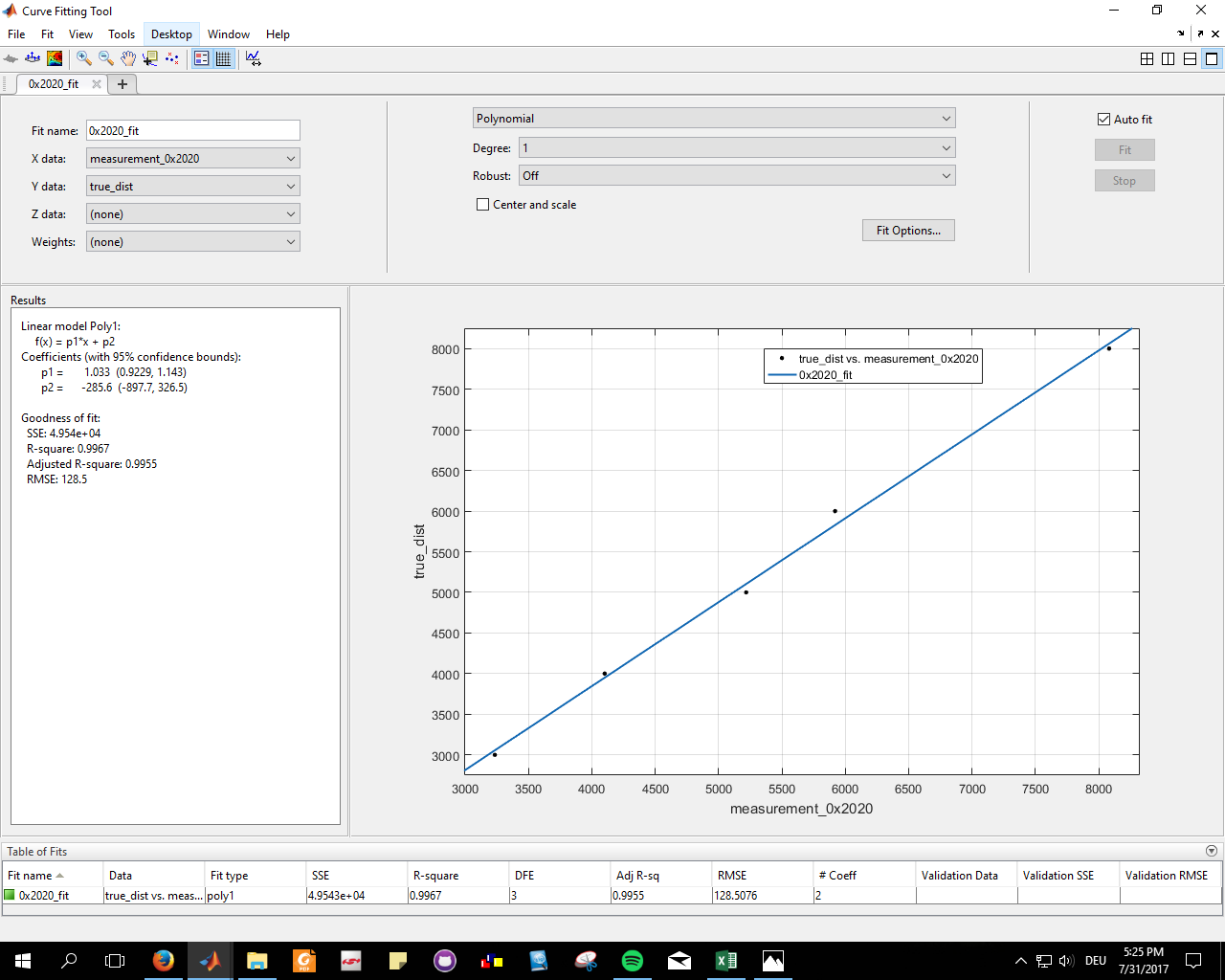
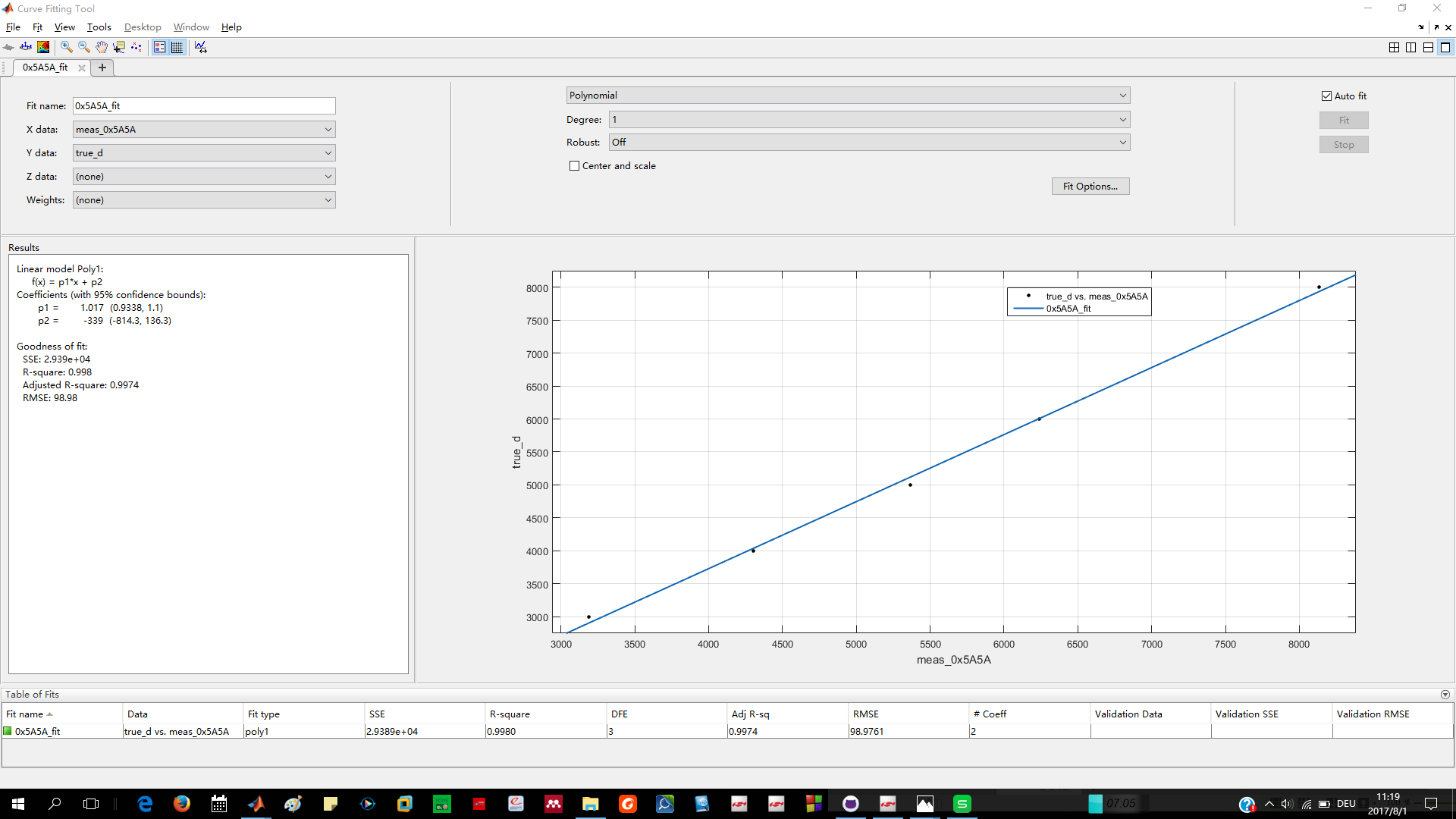
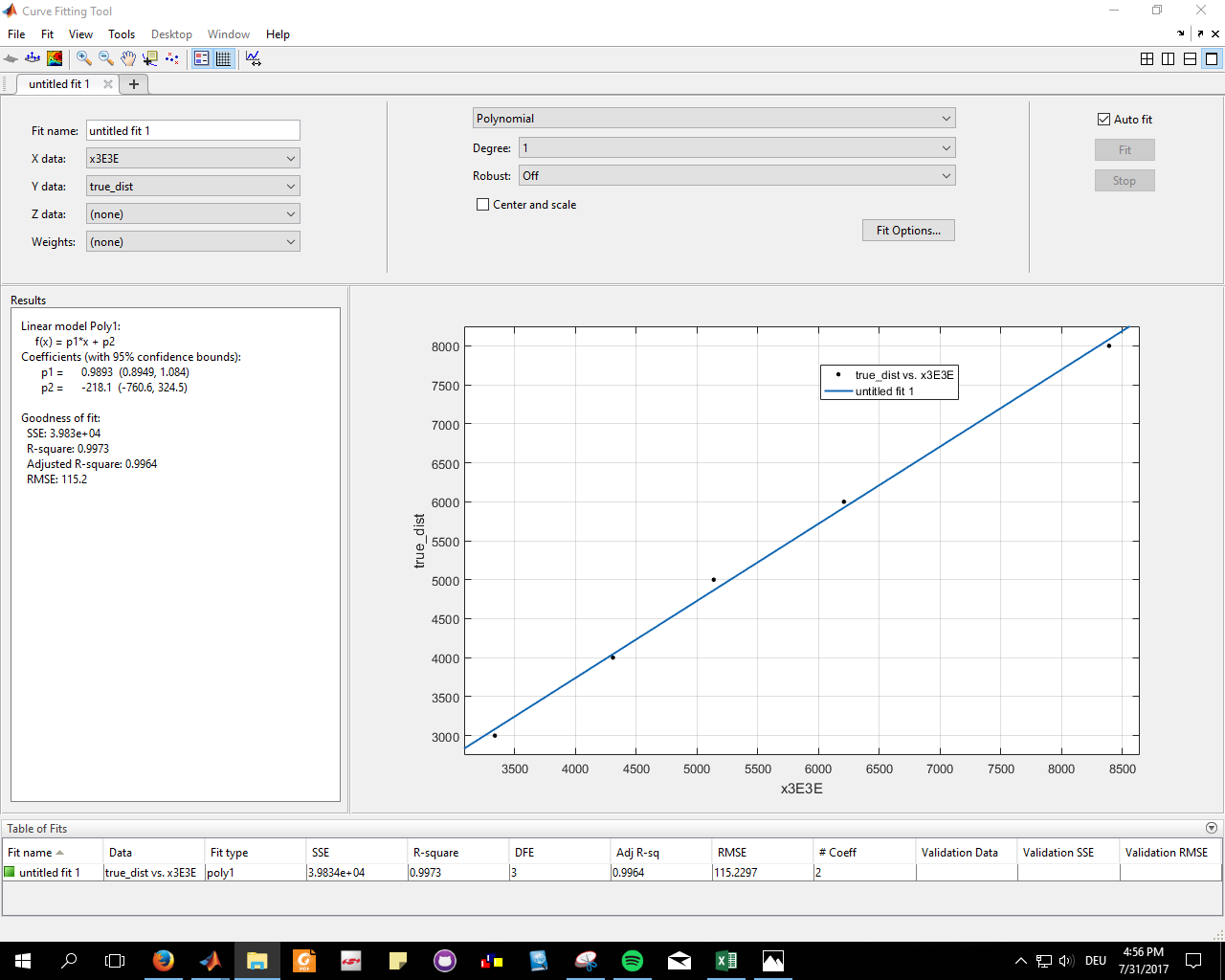
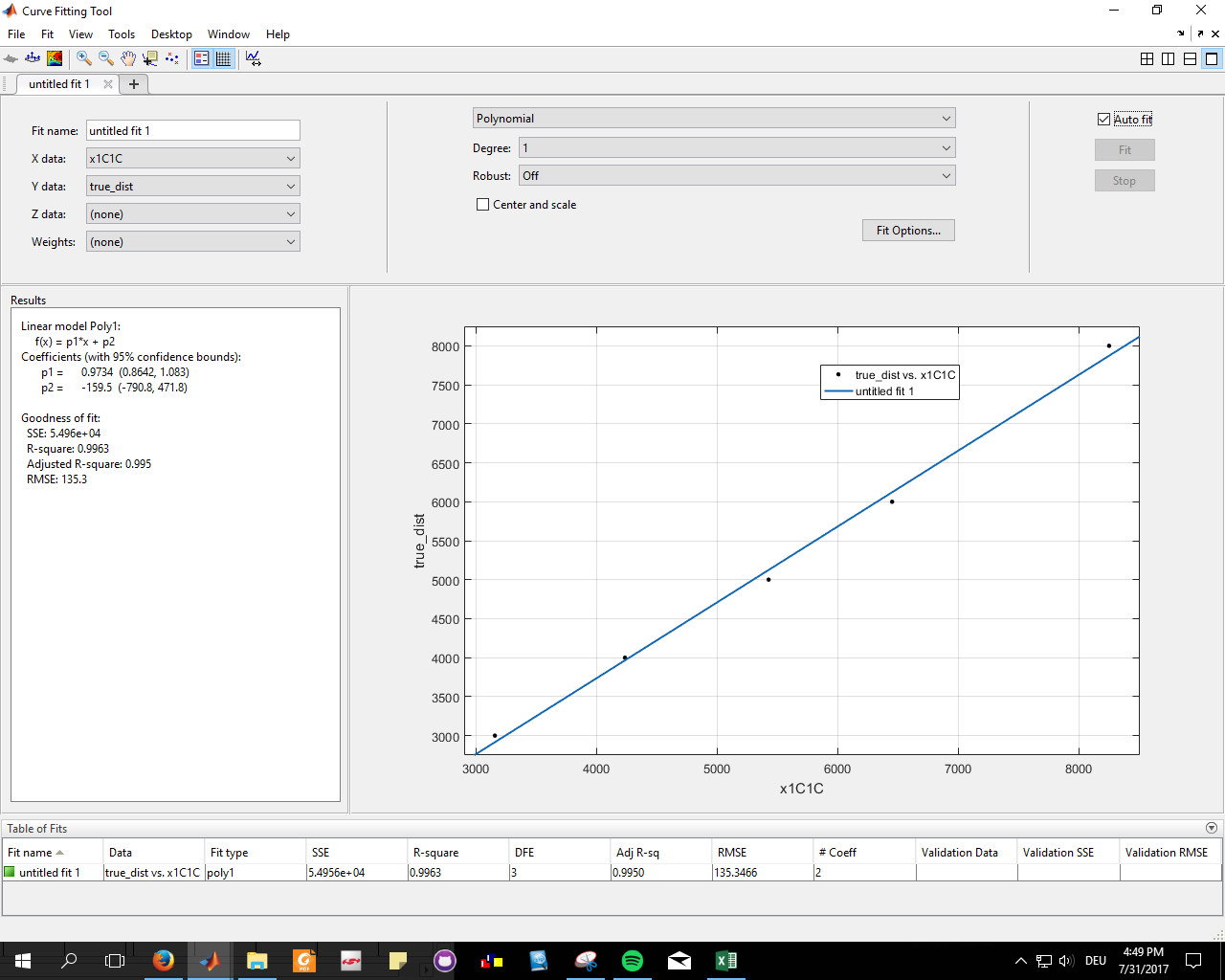
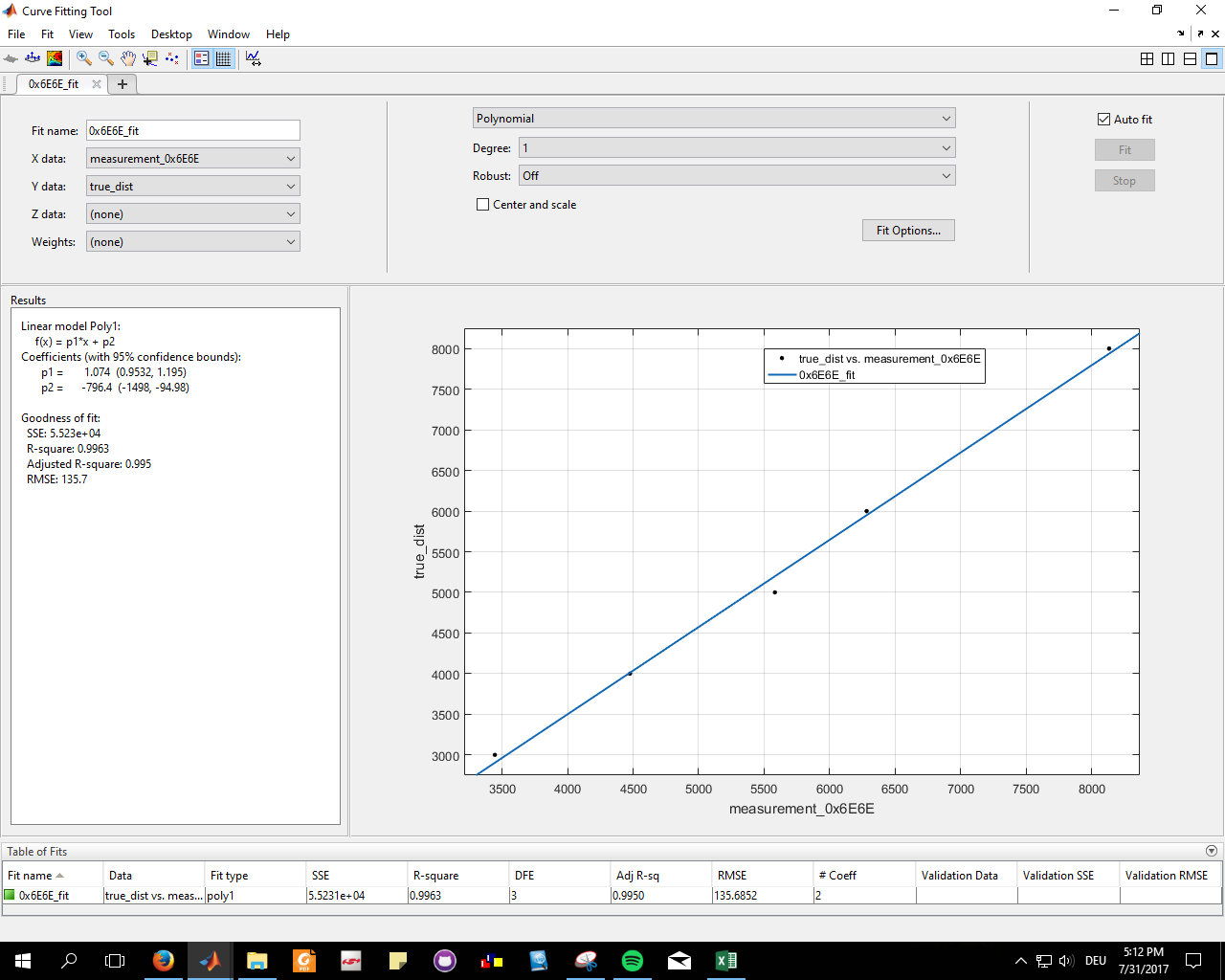
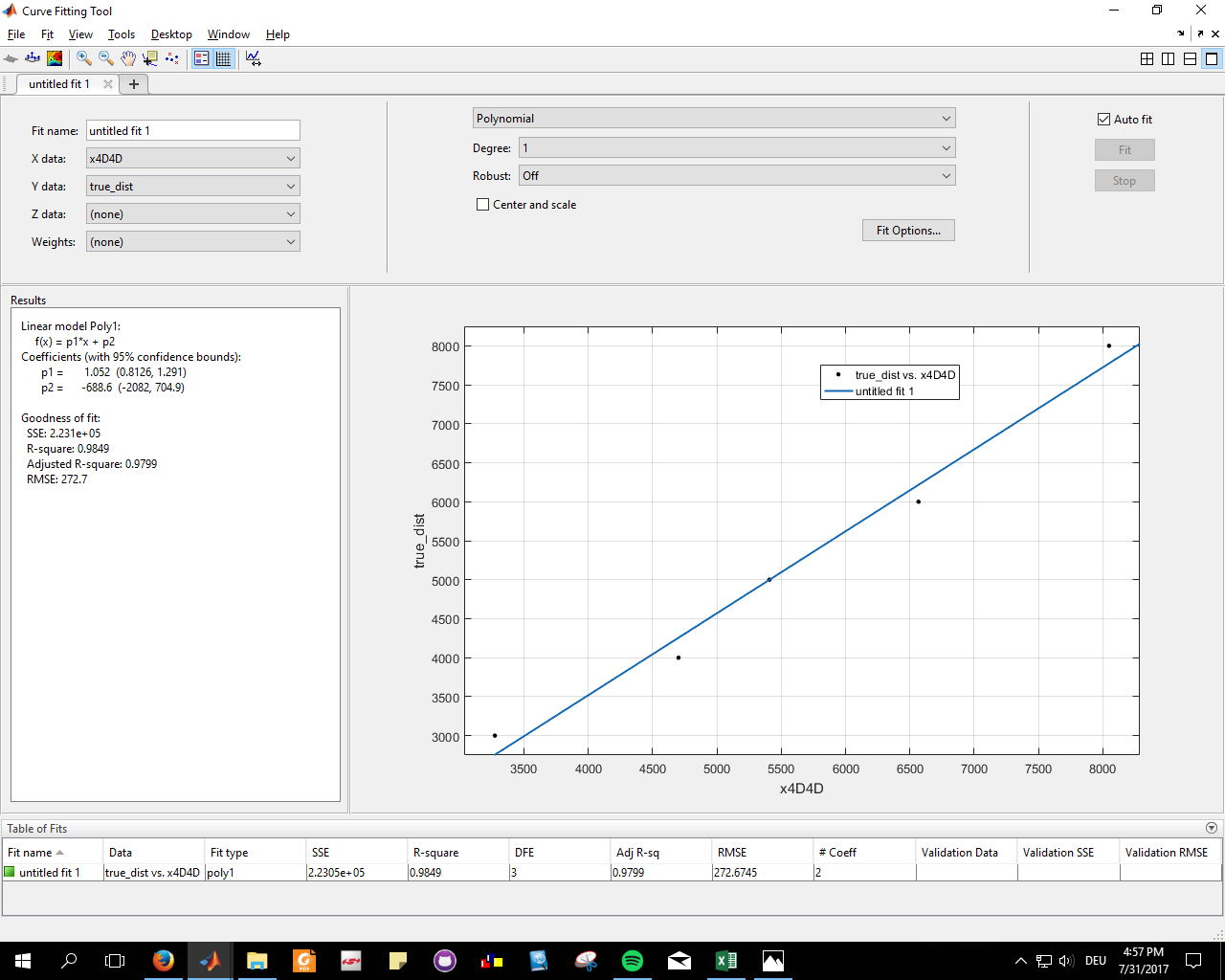
‘thesis\_indoorLocalization\communicate\_with\_basis\_station\_in\_matlab\calibrations\outdoor\OUTDOOR\_MEASUREMENT\_07\_27\_2017\histogram’

Trimmean and mad useage.

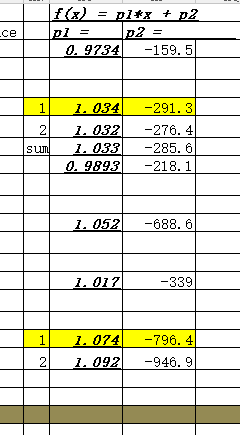
Trimmean for calibration

Mad for R

Calibration results



‘thesis\_indoorLocalization\communicate\_with\_basis\_station\_in\_matlab\calibrations\outdoor\OUTDOOR\_MEASUREMENT\_07\_27\_2017\fitting\_0x1C1C.PNG’



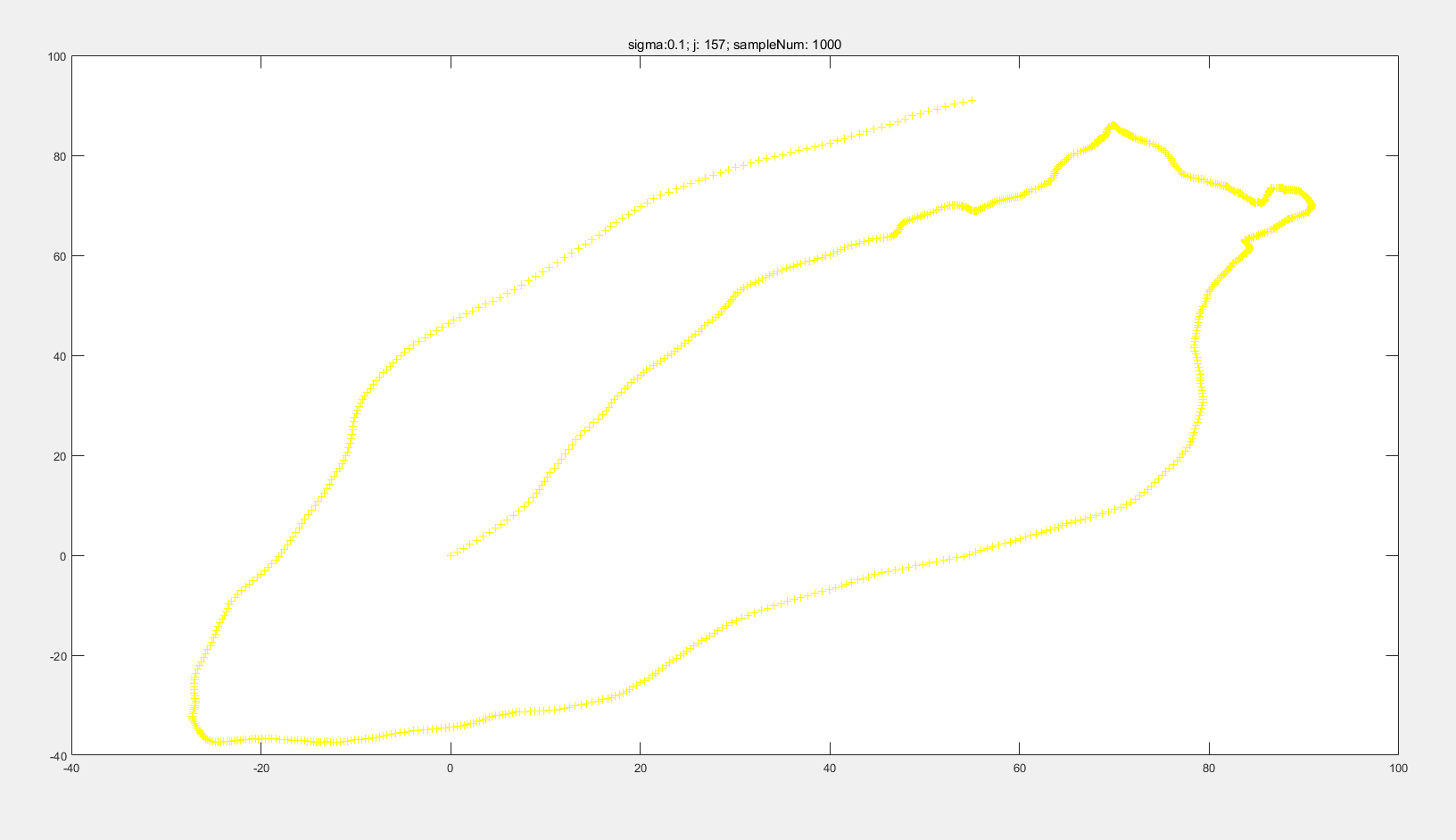
**(mad, trimmean)**

‘thesis\_indoorLocalization\communicate\_with\_basis\_station\_in\_matlab\calibrations\outdoor\OUTDOOR\_MEASUREMENT\_07\_27\_2017\outdoor\_calibration\_results.xlsx’

4.1.2 Data Collections

5 Experiments Base on Simulations data

5.1 generating moving trajectories

(unit m)

(‘\thesis\_indoorLocalization\trajectory\goodTraj01’)

Algo::Code:(\thesis\_indoorLocalization\trajectory\TrajectoryGenerator\_plot\_onlyONE\_qualified\_with\_changing\_Sigma.m)

5.2 choosing Positions of Anchor-Nodes

>2m

Cover the traj in the acceptable range, which the signal still can be reached to most of the nodes

5.3 EKF without noisy distance data

5.3.1 choosing the right parameter for EKF

Q/R big trusts more on measurements, vercise wise...the motion model is randon, not so well simlated , so try to make the Q big.

Q/R big or small results

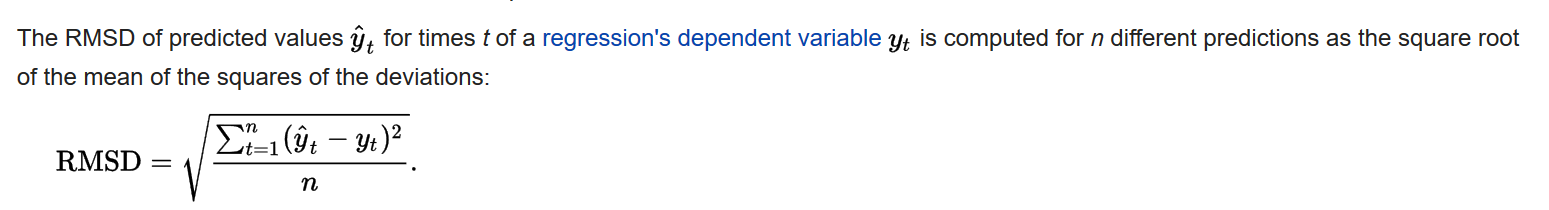
5.3.2 different strategies of processing incoming data

5.3.2.1 evenly distributed sampling in time domain

5.3.2.2 uneven distributed sampling in time domain

5.3.3 Results

**root-mean-square error (RMSE)** is a frequently used measure of the differences between values (sample and population values) predicted by a model or an estimator and the values actually observed.(wikipedia)

(wiki)

**RMSE v.s. ratio of Q/R**

(C:\Users\Yitong\Documents\GitHub\thesis\_indoorLocalization\KalmanFilter\KF\_traj\results\same R to generate noise\data not missing) (results are not calculated by RMSE, but this way[*mis\_match = sum(mis\_pos(1,:).^2 + mis\_pos(2,:).^2) / size(X,2) / area\_of\_map;*]) convert back to RMSE

RMSE = sqrt( *mis\_match\*area\_of\_map ) =* sqrt( *mis\_match\*140\*140). (PS 140 unit m)*

*When data is not enough , this set can be use ’differ R to generate noise’*

(C:\Users\Yitong\Documents\GitHub\thesis\_indoorLocalization\KalmanFilter\KF\_traj\results\differ R to generate noise\fig\_goodTraj01)



5.4 EKF with noisy distance data

5.4.1 noisy data generatation(add noise / delete data)

(add noise according to the R from mad)

delete data (randon delete; in a specific order to reproduce the cinario of 25ms40Hz case)

5.4.1 choosing the right parameter for EKF (XXX)

5.4.2 Results

25ms40Hz case

‘C:\Users\Yitong\Documents\GitHub\thesis\_indoorLocalization\KalmanFilter\KF\_traj\EKF\_25ms40Hz\25ms\_40HzSamplingRate’

**RMSE v.s. ratio of Q/R**

**RMSE v.s. number of missing meas**

**RMSE (noisy free vs noisy)**

5.5 Self-Calibrations for Anchor-Nodes

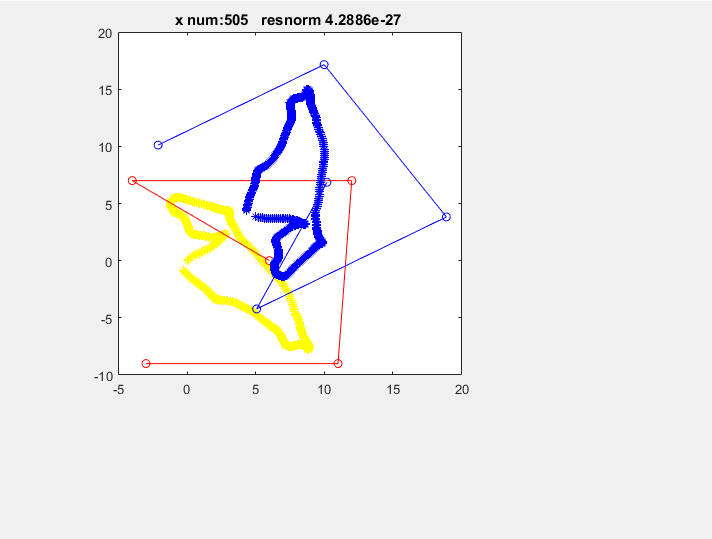
550 principle and method(optimization)

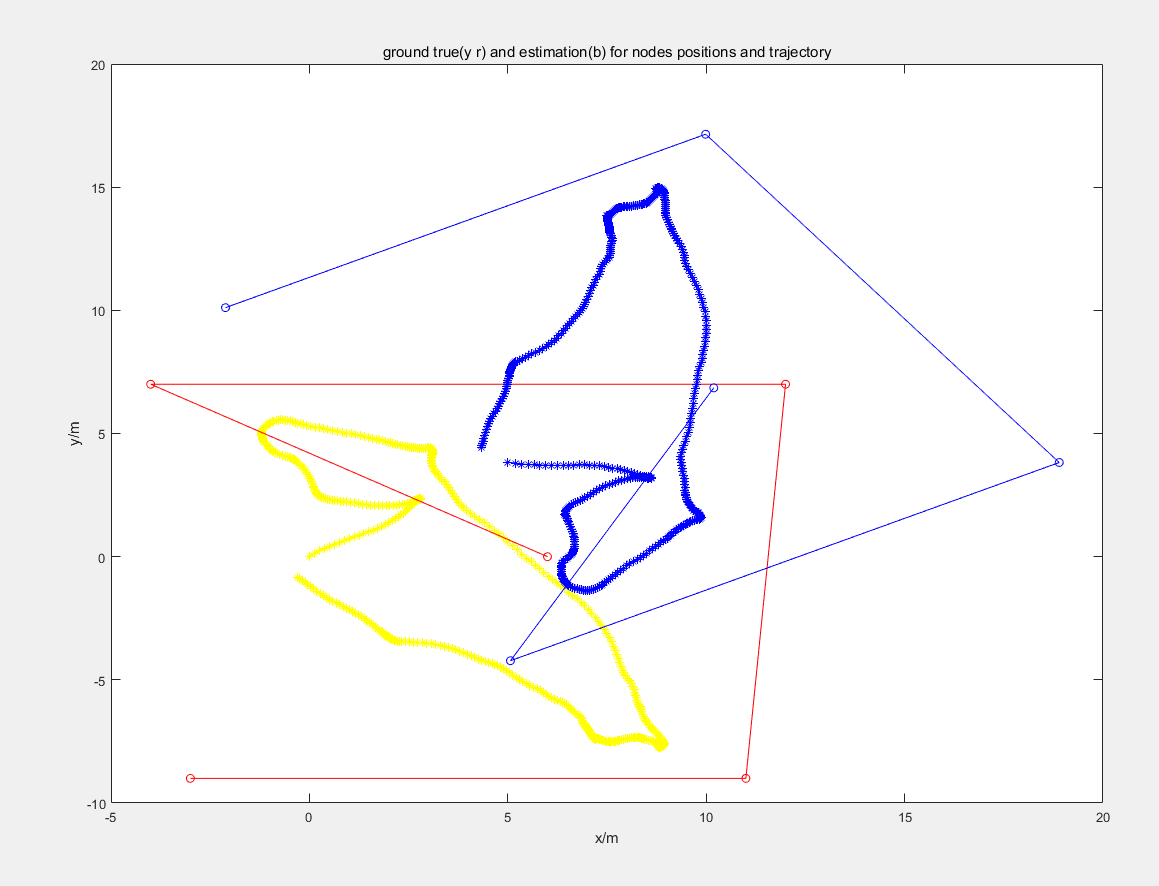
5.5.1 performance effected factors

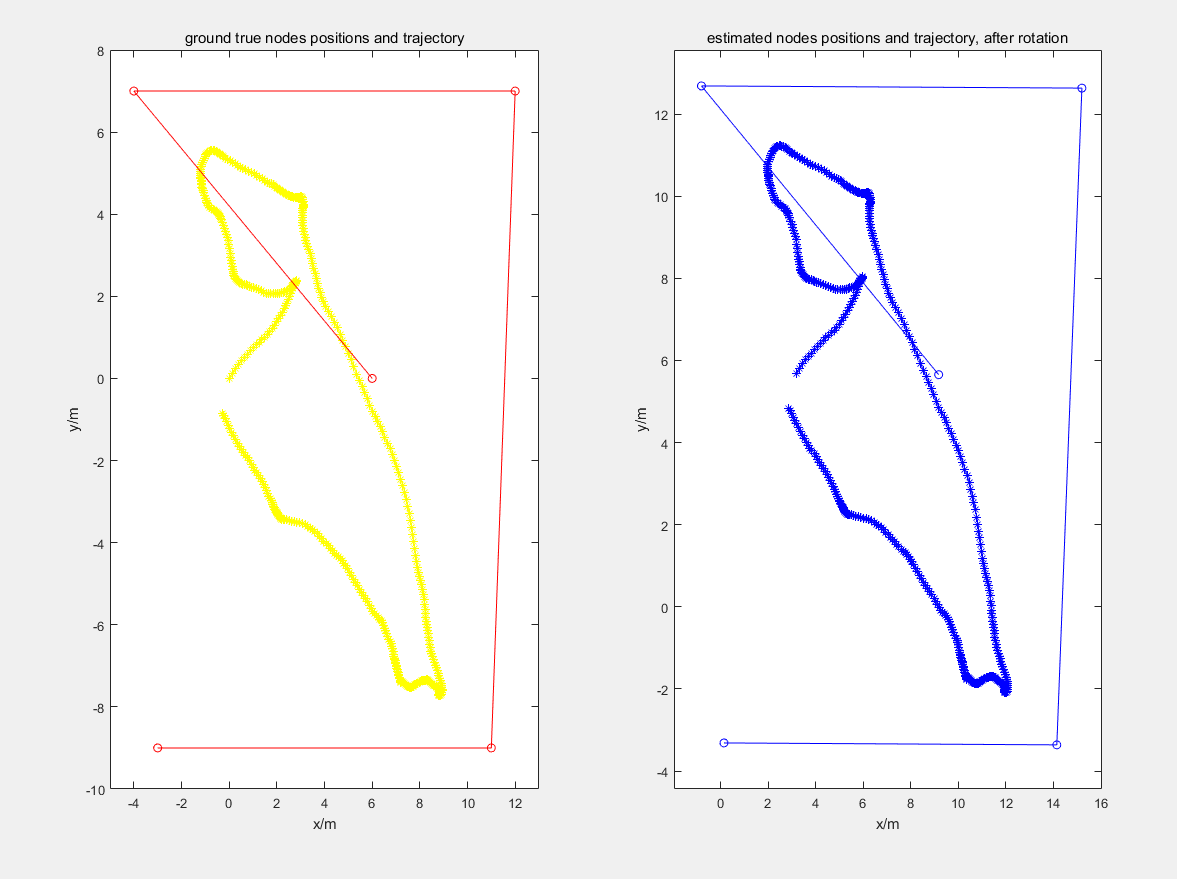
5.5.1.1 number of trajectory points and Anchor-Nodes

5.5.1.2 distributions of the Anchor-Nodes and the trajectory

5.5.2 results







(TODO: apply transform to it)

‘C:\Users\Yitong\Documents\GitHub\thesis\_indoorLocalization\Calibration-Free Localization\simulation\_results’

6 Experiments based on measurement data

6.1 EKF with Measurement data

6.1.1 choosing the right parameter for EKF

6.1.2 Results

6.2 Self-Calibrations for Anchor-Nodes

6.2.1 Results

6.3 25ms40Hz

6.3.1 data mitigation

6.4 first self calibration then EKF

RMSE (VS hand meas resultingEKF)