POLITECNICO
MILANO 1863

DIPARTIMENTO DI ELETTRONICA
INFORMAZIONE E BIOINGEGNERIA

LOCALIZATION, NAVIGATION AND SMART MOBILITY PROJECT

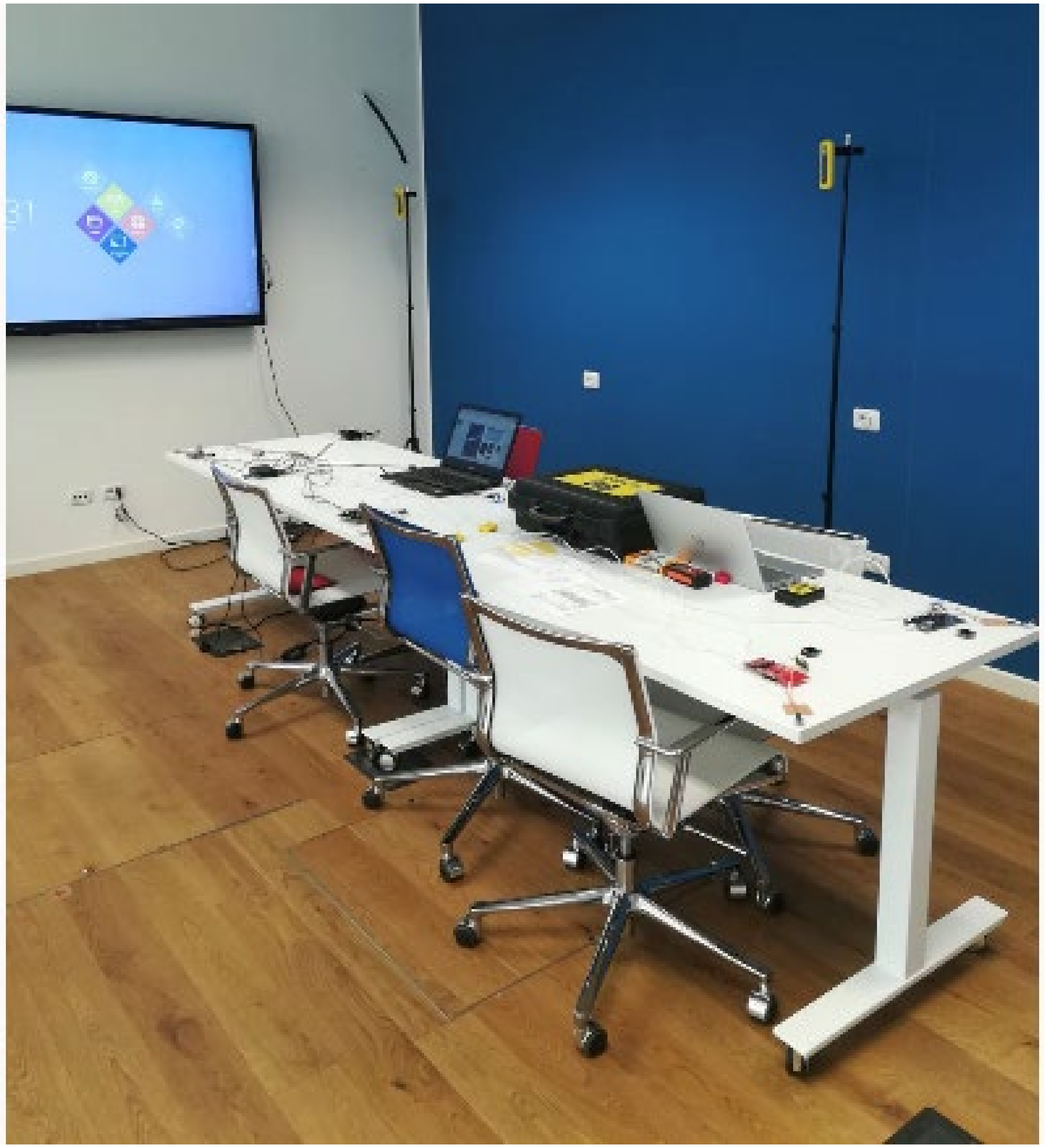
Alessandro De Luca - 10676114

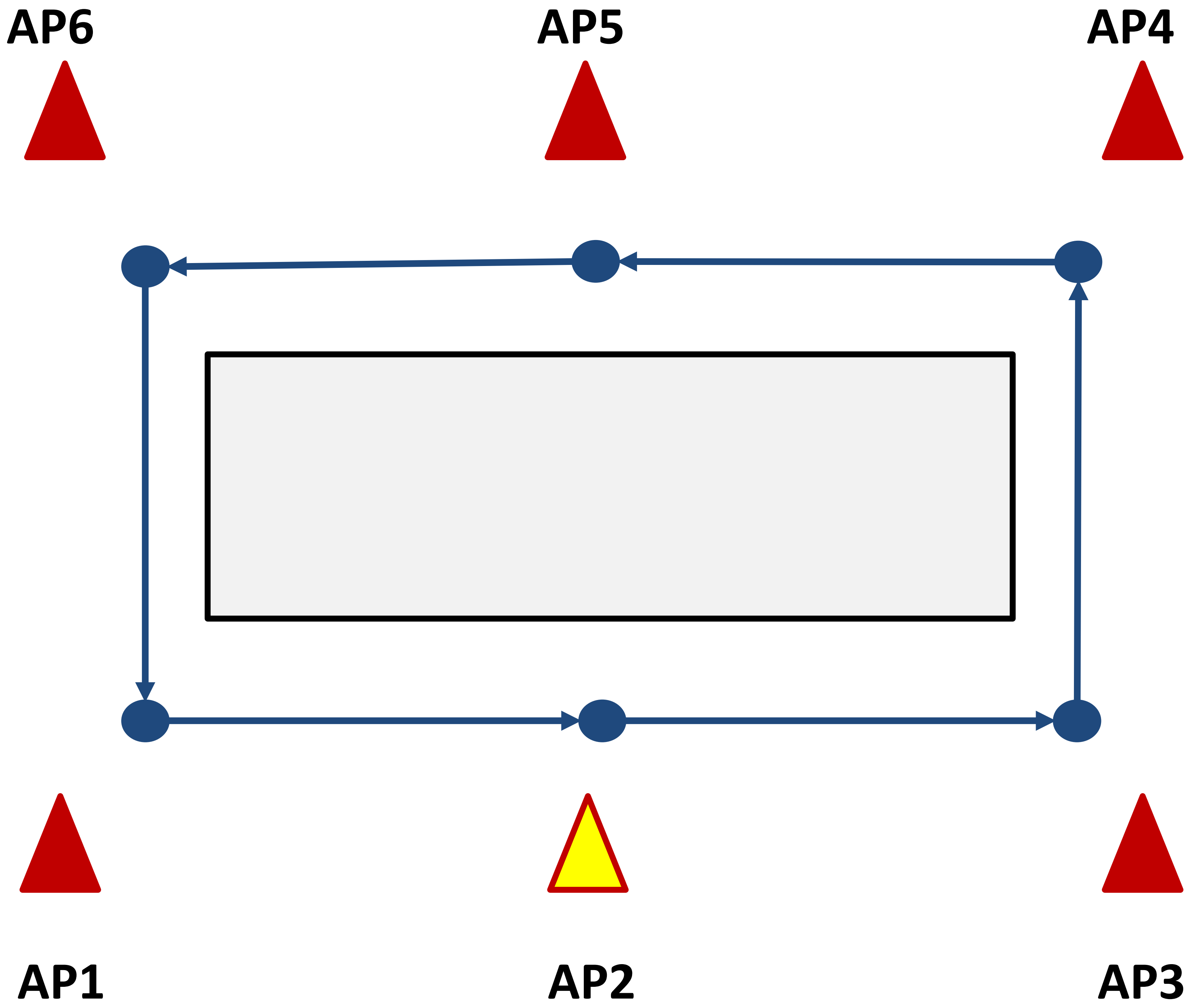
The aim of the project was to implement a localization algorithm and a tracking filter on Time Difference Of Arrival data gathered by an UWB tag

In particular the project was executed in 3 steps:

- **Cleaning of the dataset** to remove noise and outliers
- Implementation of the **Maximum Likelihood localization algorithm**
- Implementation of the **Extended Kalman Filter** and **Particle filter**

Scenario

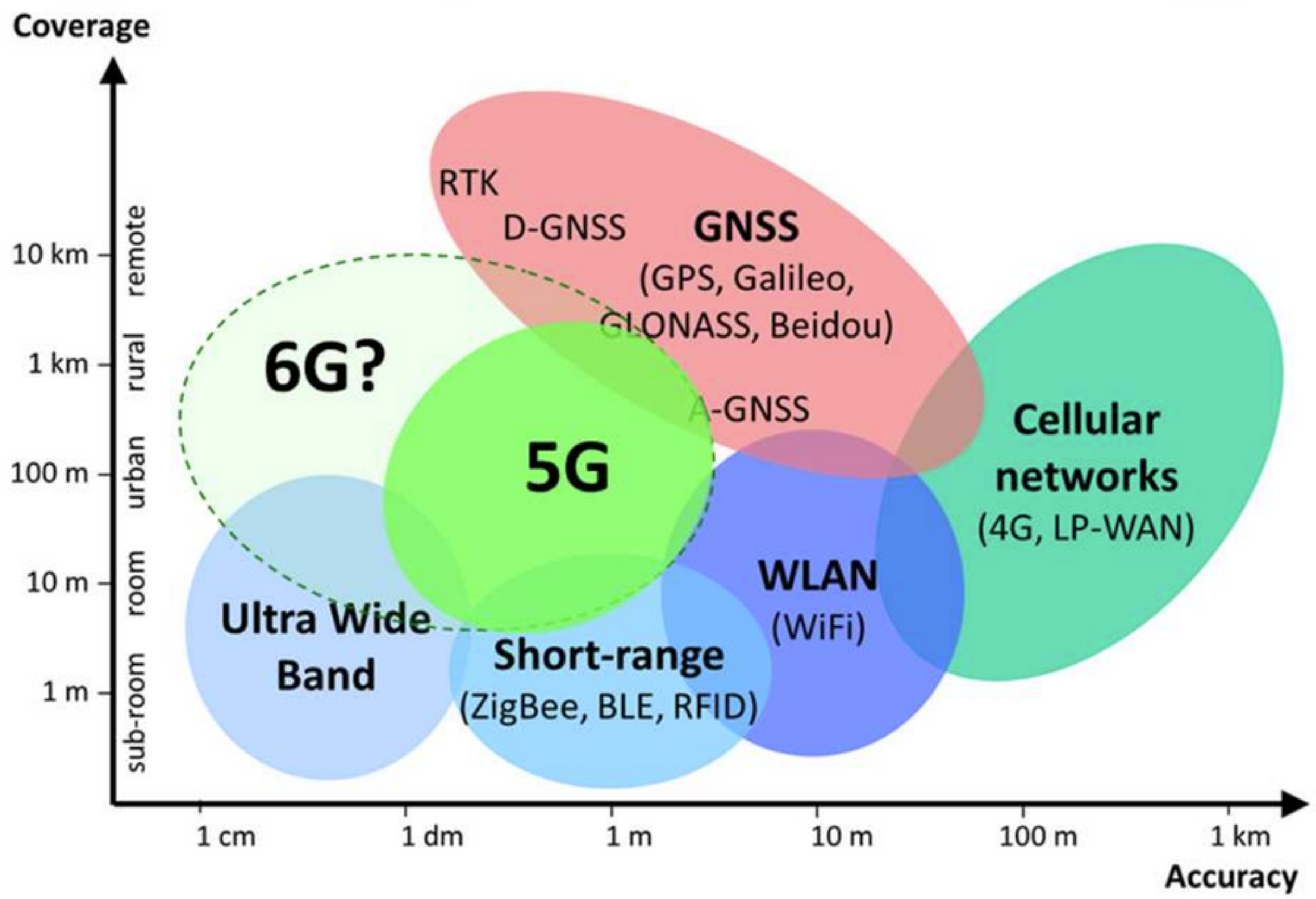




Scenario

The parameters used are:

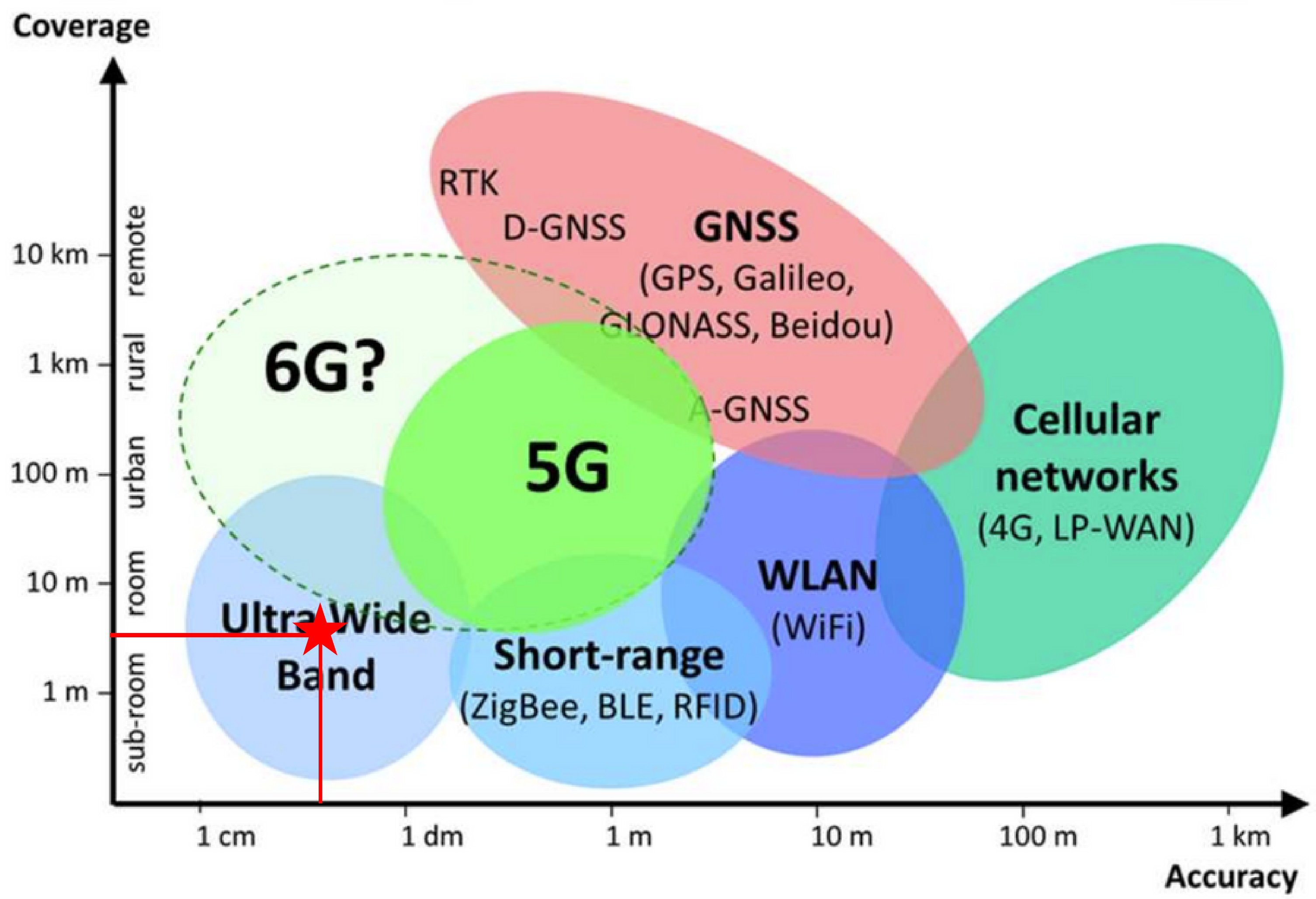
- x_min = 0 m and x_max = 6 m (from dataset)
- y_min = 0 m and y_max = 3.5 m (from dataset)
- z_min = 0 m and z_max = 2 m (from dataset)
- sigma_TDOA = 0.05 m
    - Choice made due to the fact that Ultra Wide Bandwidth are very precise especially in room and sub room enviromnets



Scenario

The parameters used are:

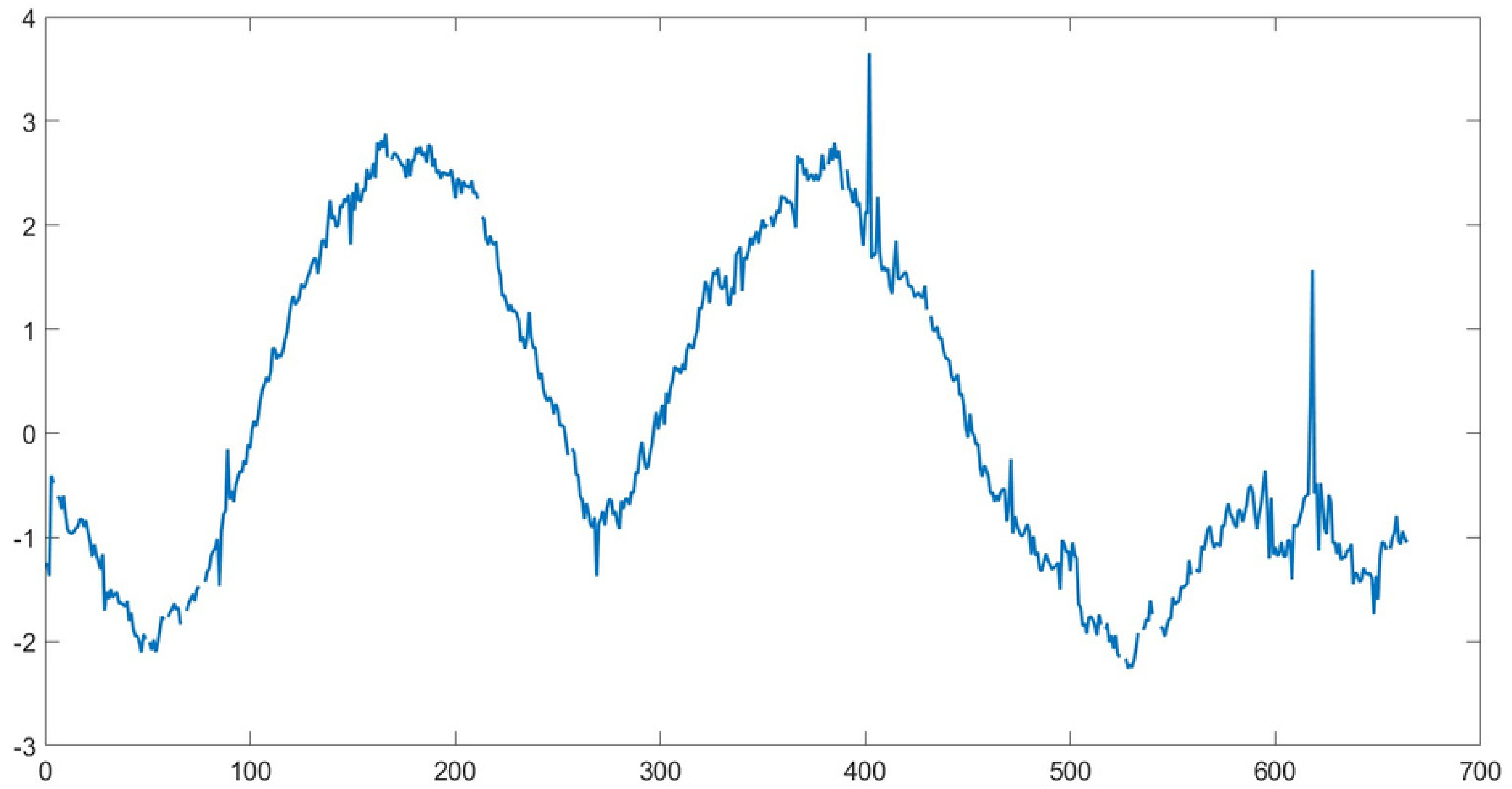
- x_min = 0 m and x_max = 6 m (from dataset)
- y_min = 0 m and y_max = 3.5 m (from dataset)
- z_min = 0 m and z_max = 2 m (from dataset)
- sigma_TDOA = 0.05 m
    - Choice made due to the fact that Ultra Wide Bandwidth are very precise especially in room and sub room enviromnets



Data Cleaning

The process of Data Cleaning is aimed at filling NaN values from the dataset and then removing noise and outliers. The steps taken were:

1. Fill NaN values by interpolating subsequent samples with fillmissing(dataset, ‘linear’) function

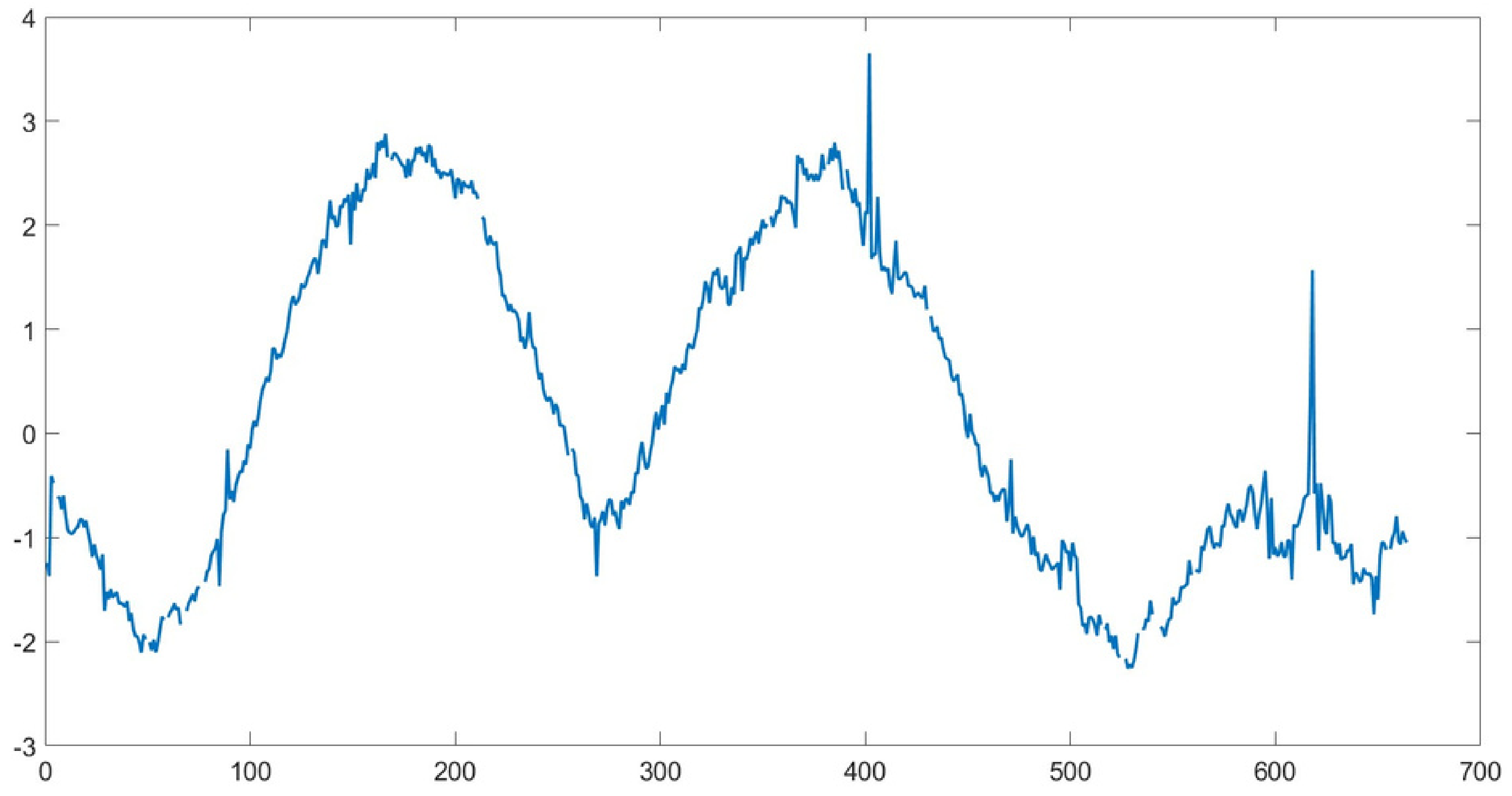


Data Cleaning

The process of Data Cleaning is aimed at filling NaN values from the dataset and then removing noise and outliers. The steps taken were:

1. Fill NaN values by interpolating subsequent samples with fillmissing(dataset, ‘linear’) function

2. Compute average difference between subsequent samples and use this average as tolerance treshold

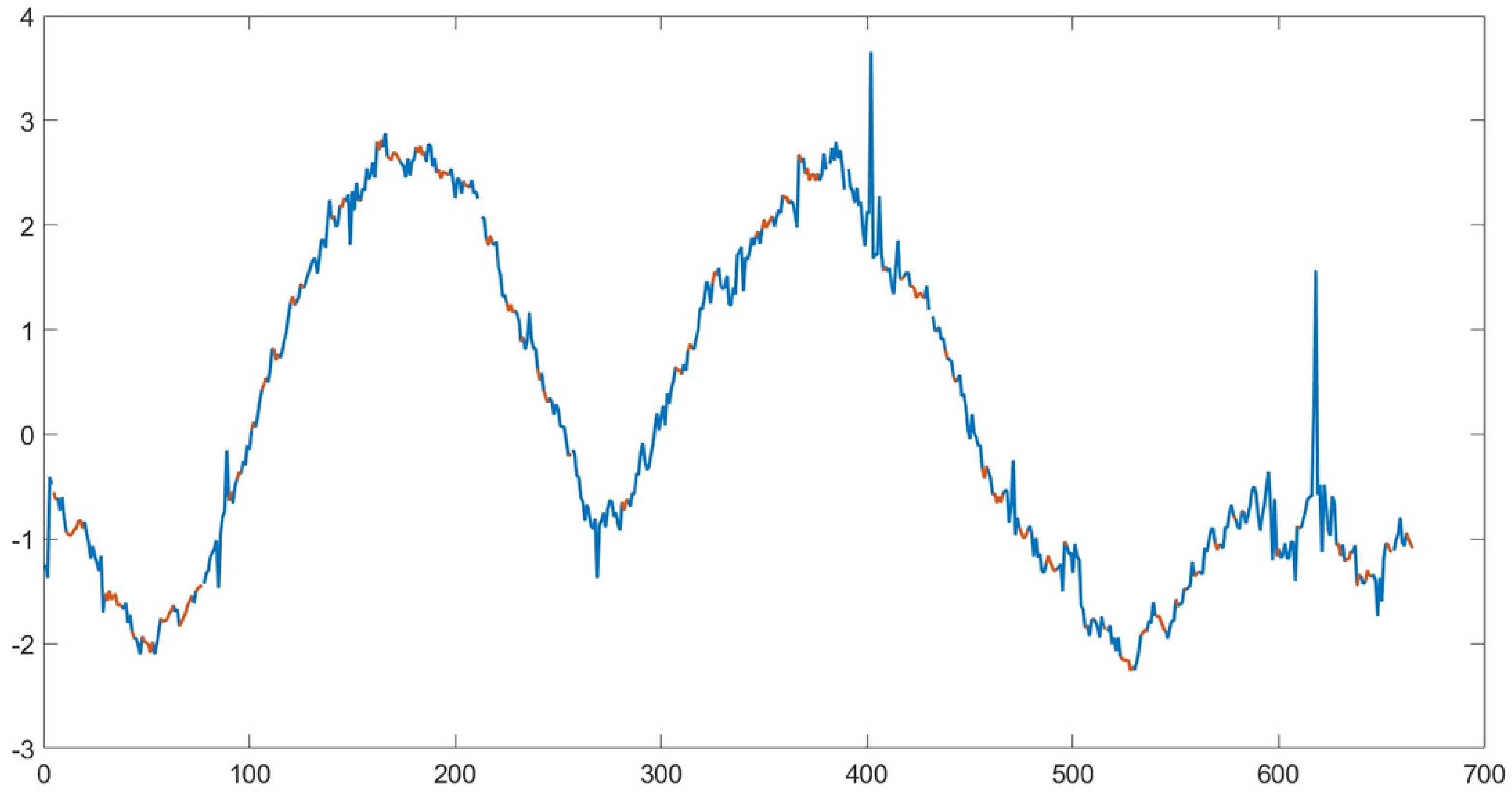


The process of Data Cleaning is aimed at filling NaN values from the dataset and then removing noise and outliers. The steps taken were:

1. Fill NaN values by interpolating subsequent samples with fillmissing(dataset, ‘linear’) function

2. Compute average difference between subsequent samples and use this average as tolerance treshold

3. Deleting all samples that lie outside this treshold, this produces the “features” of the data



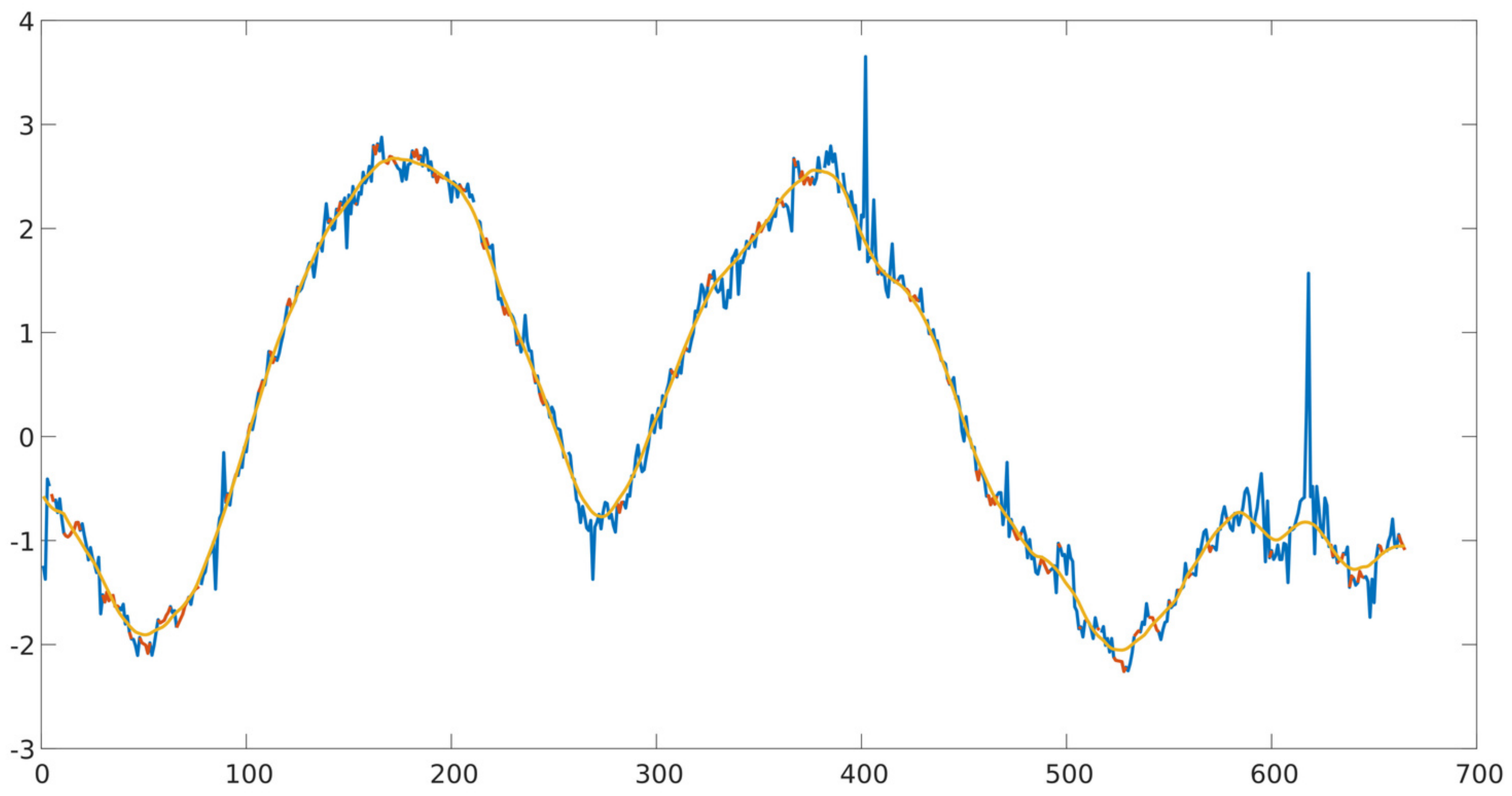
The process of Data Cleaning is aimed at filling NaN values from the dataset and then removing noise and outliers. The steps taken were:

1. Fill NaN values by interpolating subsequent samples with fillmissing(dataset, ‘linear’) function

2. Compute average difference between subsequent samples and use this average as tolerance treshold

3. Deleting all samples that lie outside this treshold, this produces the “features” of the data

4. Fill missing data with moving avarage of window 20 of the interpolated data

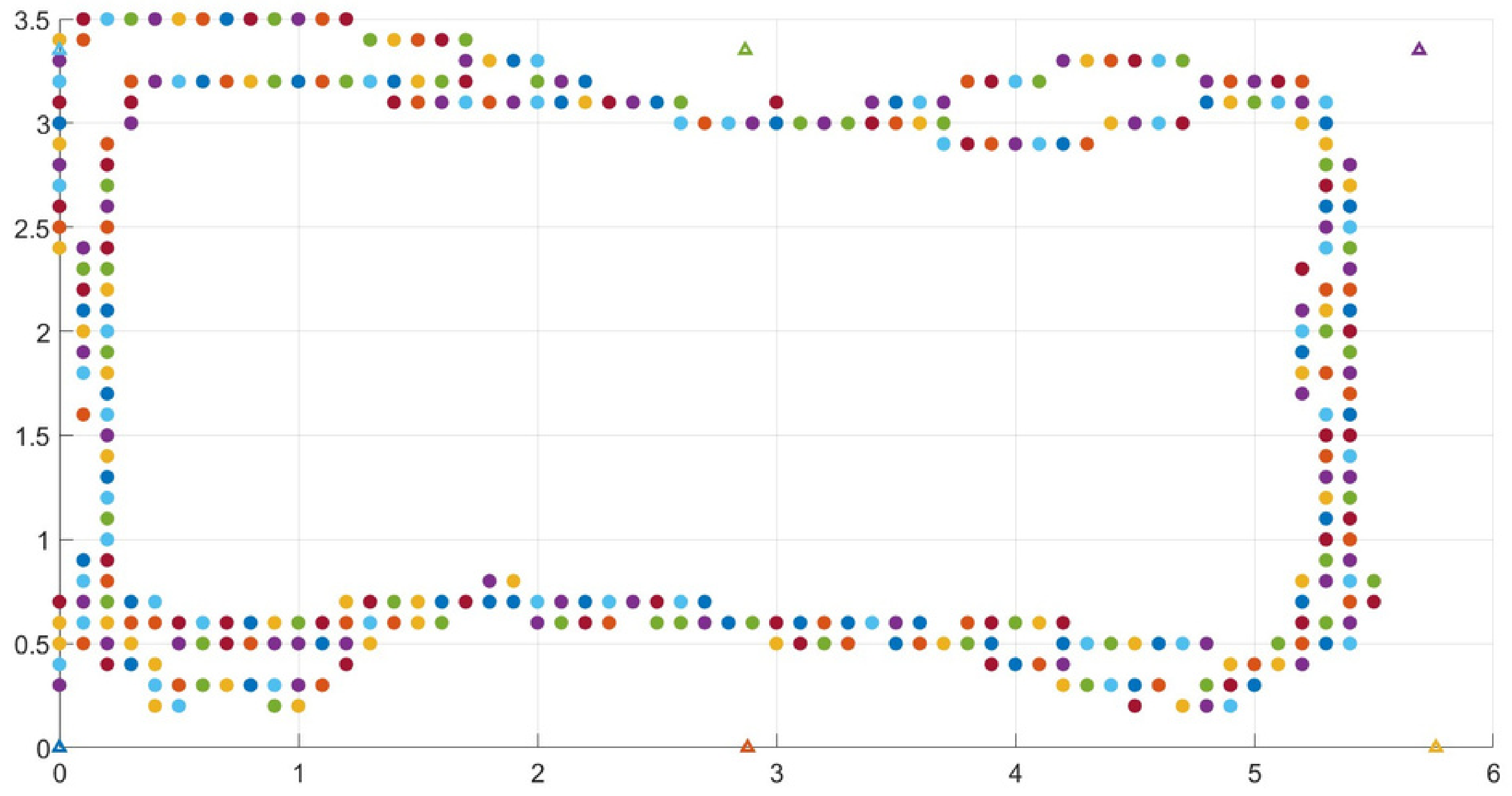


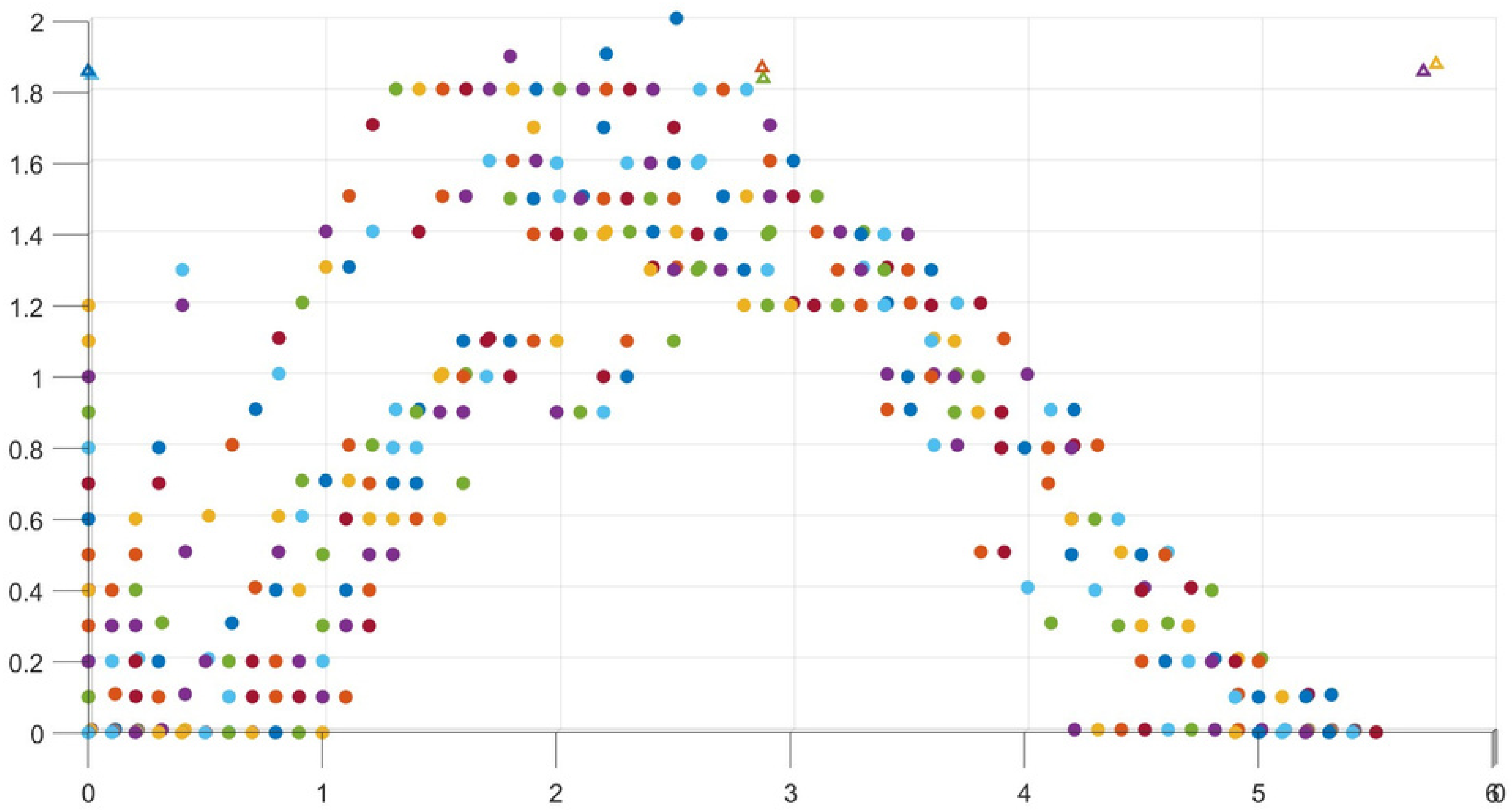
Localization - Mamimum Likelihood

As for the localization with Maximum likelihood I have split 46116 evaluation points each one at 0.1 m from its neighboring evaluation points in a room of dimension:

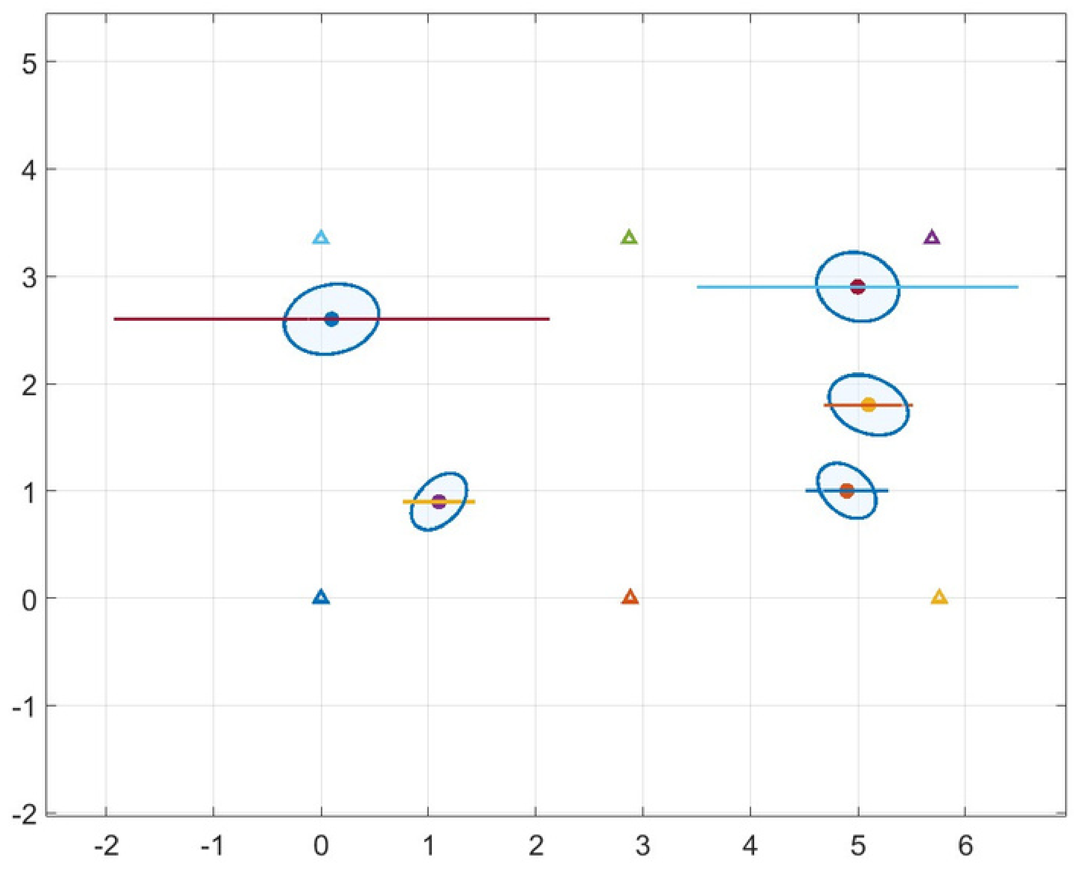
- x = 6 m, thus 61 evaluation values for x;
- y = 3.5 m, thus 36 evaluation values for y;
- z = 2m, thus 21 evaluation values for z;

#_x * #_y * #_z = 46116 evaluation points

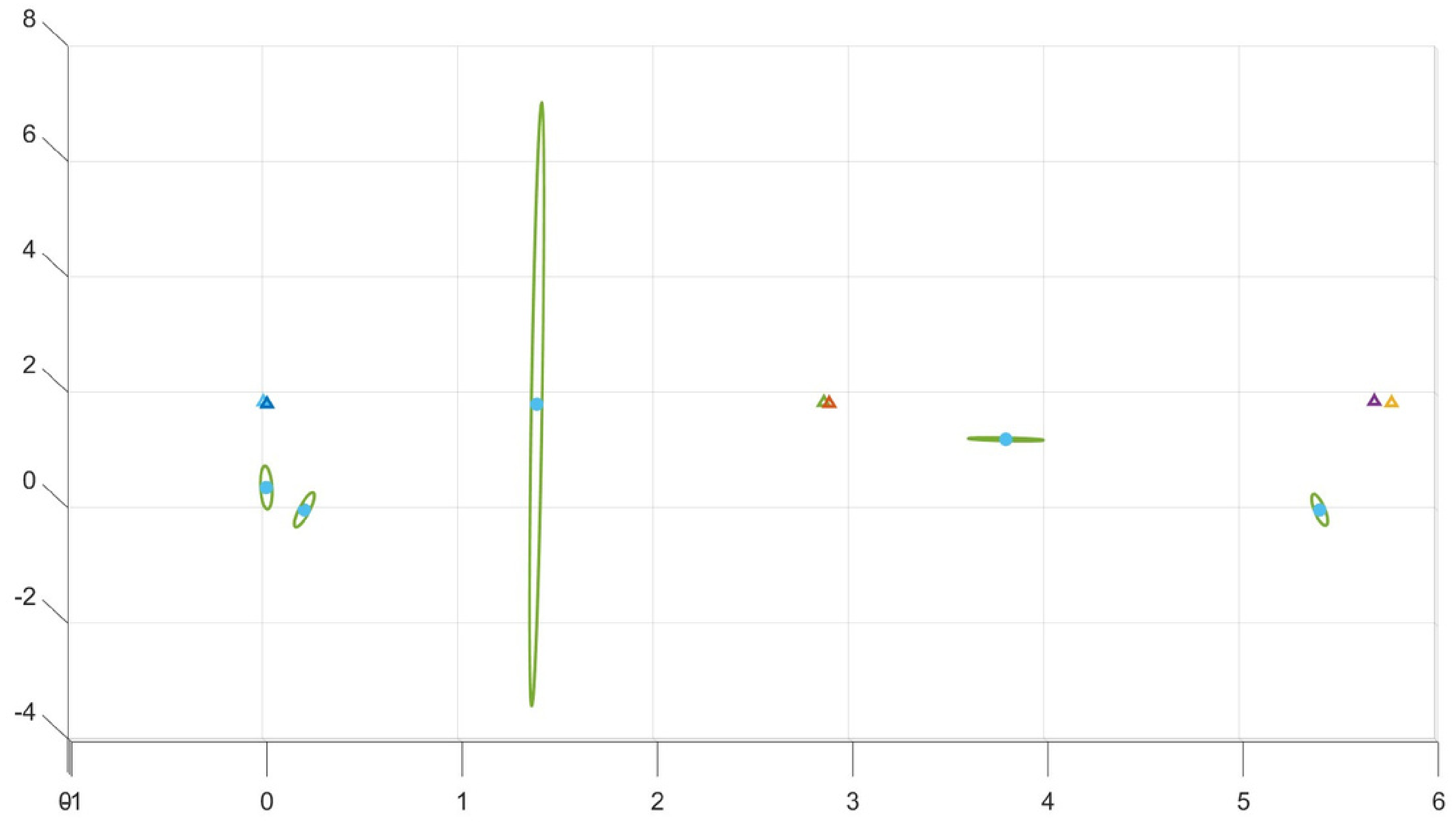




CRB for x-y localization



CRB for x-z localization

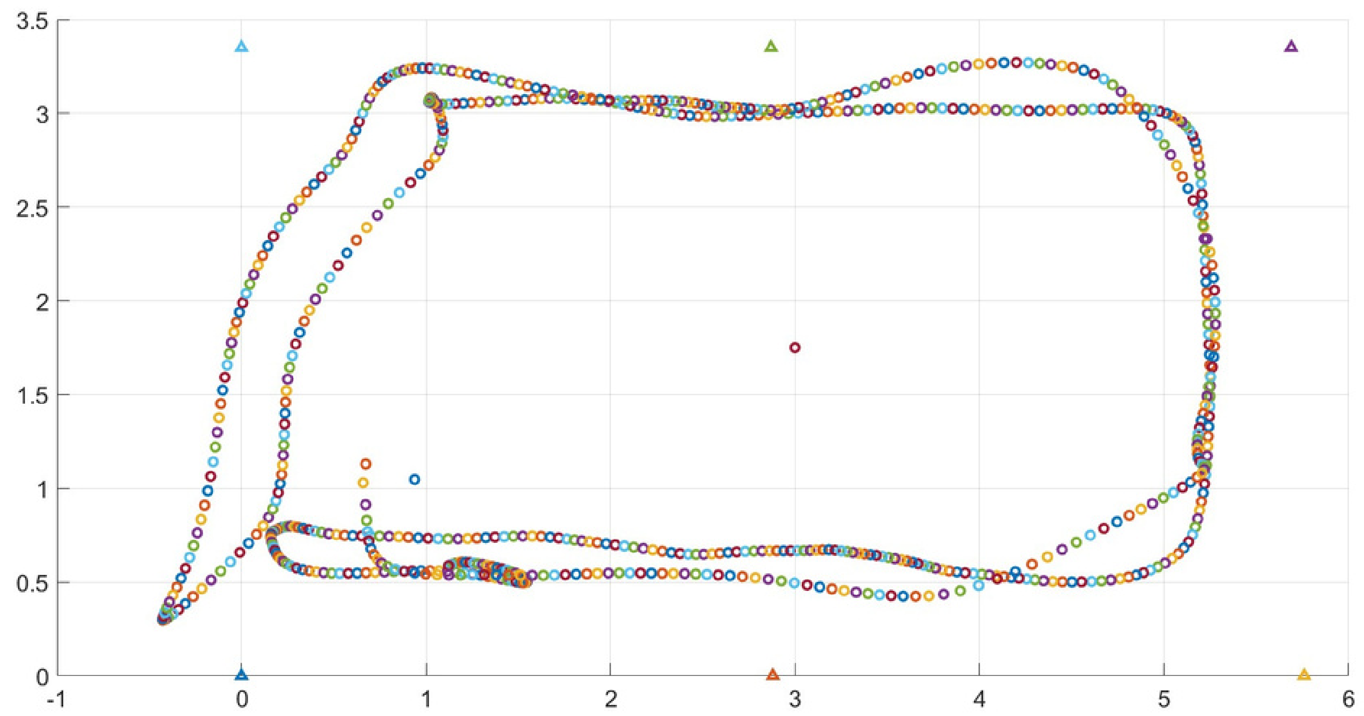


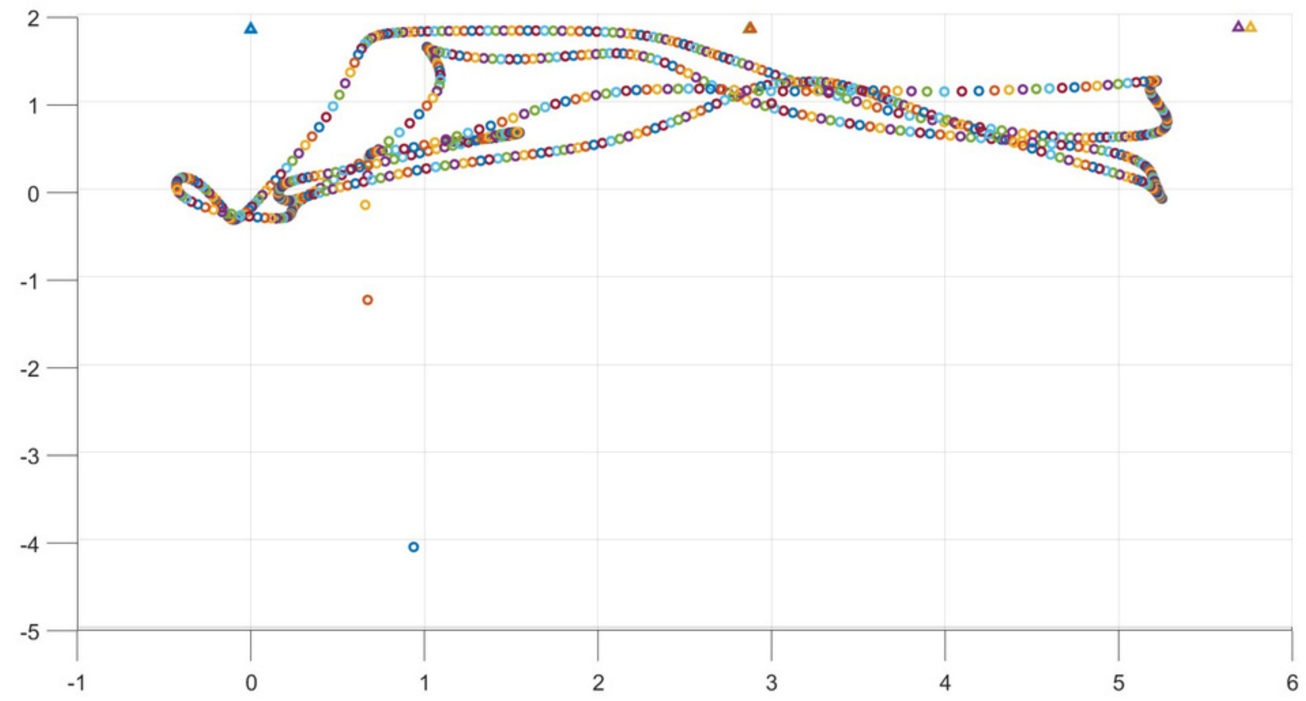
As we can see the error can be much larger in the estimation of the z coordinate

Tracking - Extended Kalman Filter

Parameters of EKF:

- sigma_TDOA = 0.05 m;
- sigma_Driving = 0.1 m/s;
- samplingTime = 0.1 s;
- R = diag(repmat(0.0025 , 1 , 5));
- L = samplingTime * I;
- Q = sigma_Driving^2 * L * L';
- F = I ; (nearly constant veloity model)
- UE_init = [3, 1.75 ,1]; (middle of the room)
- UE_init_COV = diag([10, 10, 10]);

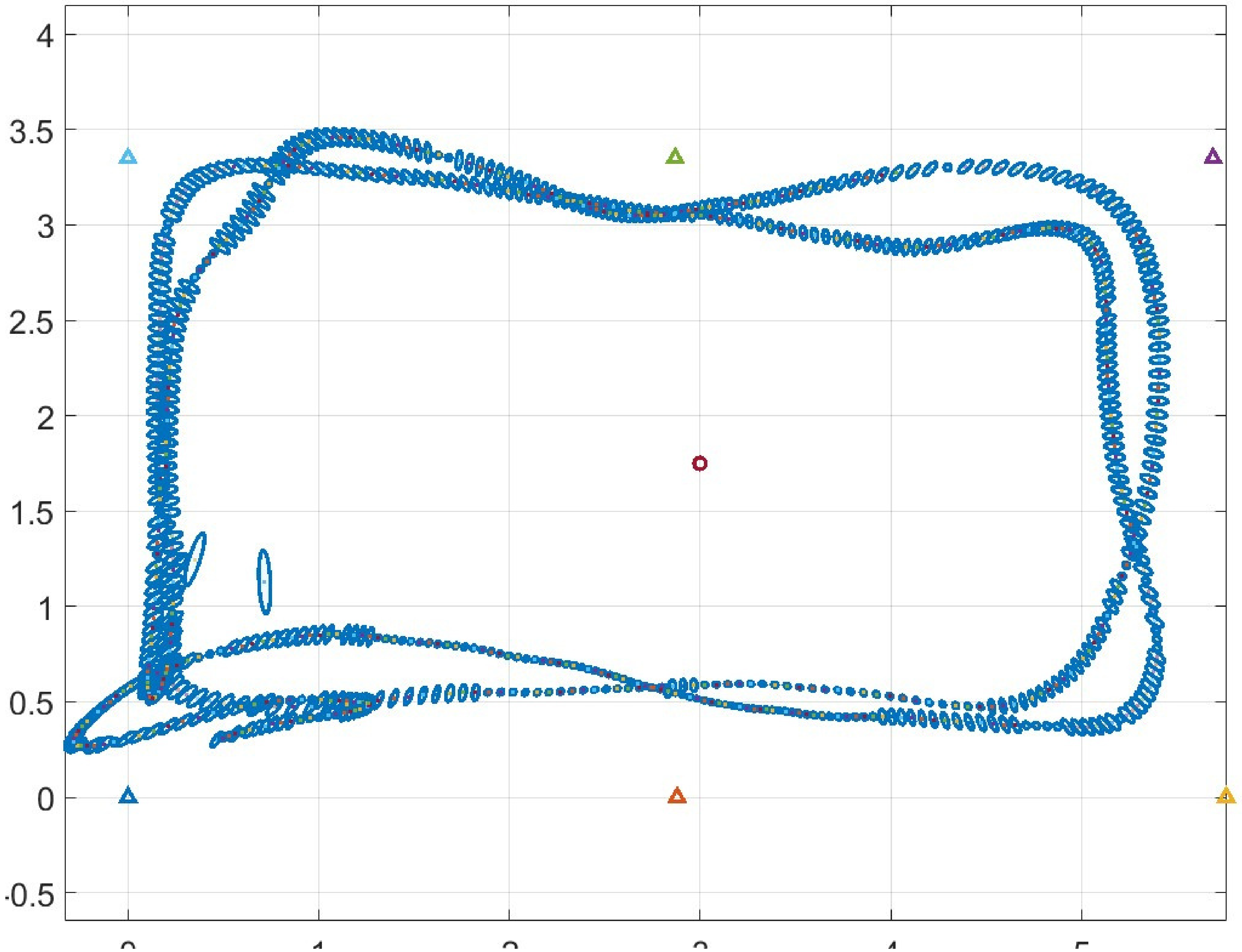




Tracking - Extended Kalman Filter

Parameters of EKF:

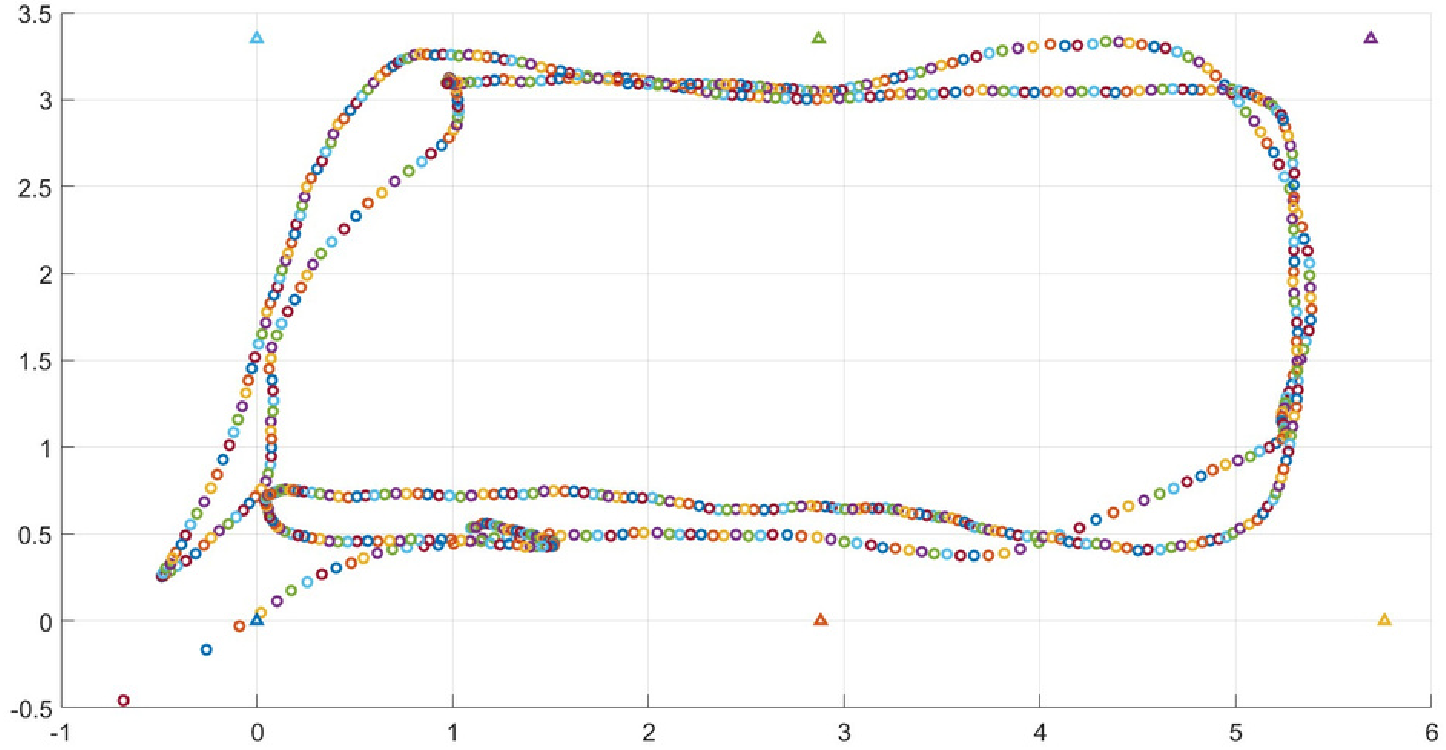
- sigma_TDOA = 0.05 m;
- sigma_Driving = 0.1 m/s;
- samplingTime = 0.1 s;
- R = diag(repmat(0.0025 , 1 , 5));
- L = samplingTime * I;
- Q = sigma_Driving^2 * L * L’;
- F = I ; (nearly constant veloity model)
- UE_init = [3, 1.75 ,1]; (middle of the room)
- UE_init_COV = diag([10, 10, 10]);

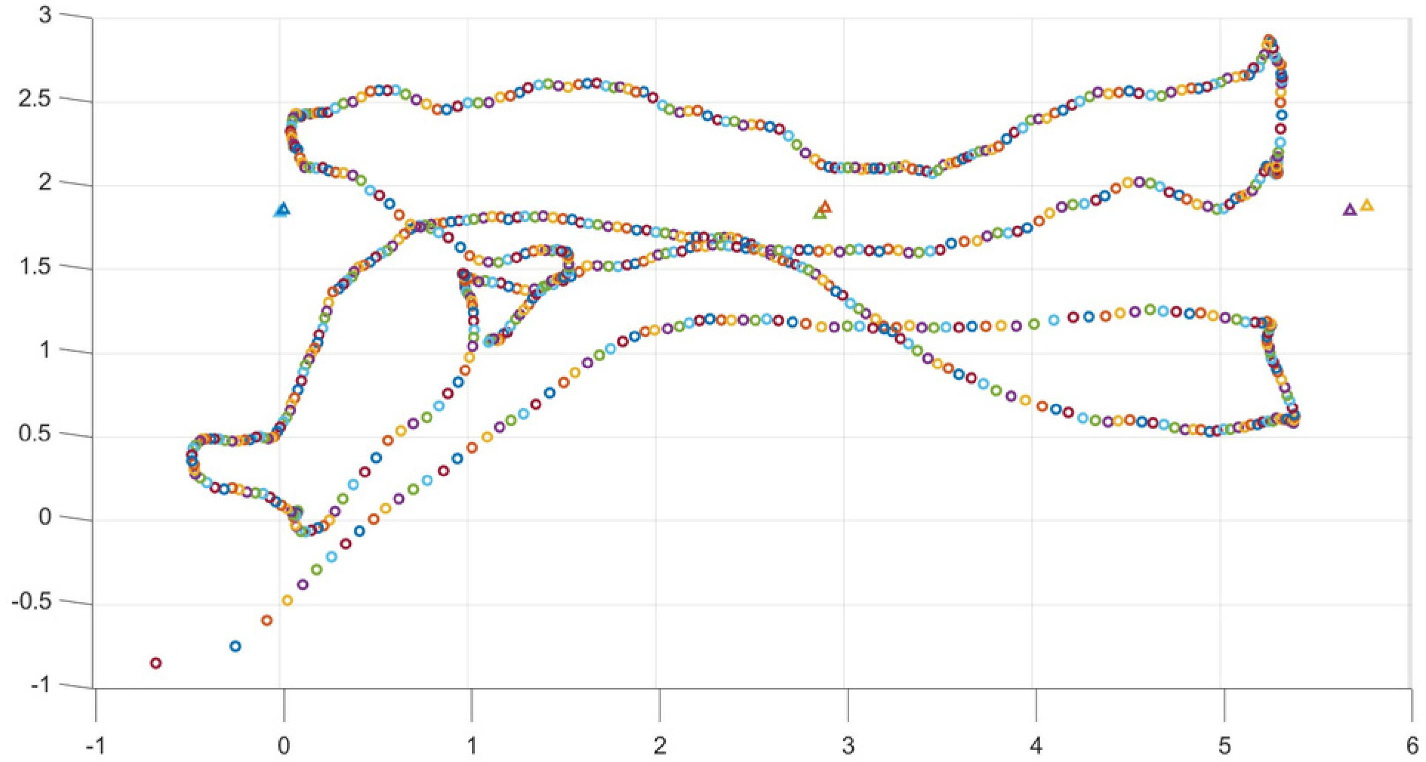


Tracking - Particle Filter

Parameters of PF:

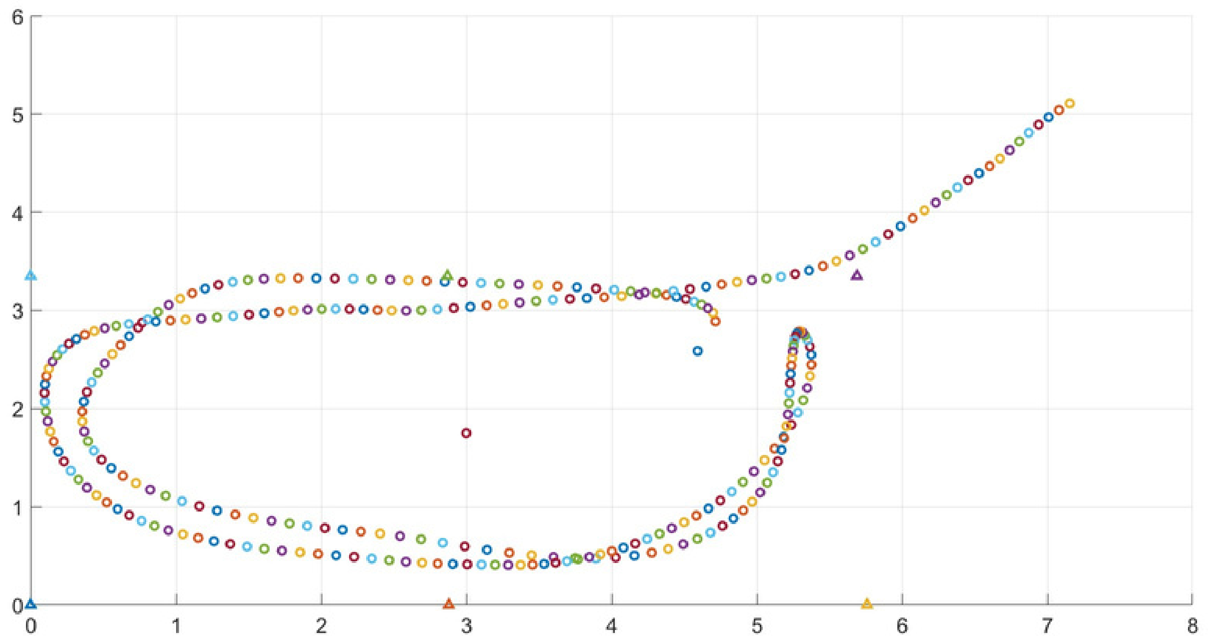
- sigma_TDOA = 0.05 m;
- sigma_Driving = 0.1 m/s;
- samplingTime = 0.1 s;
- number_Of_Particles = 1000;



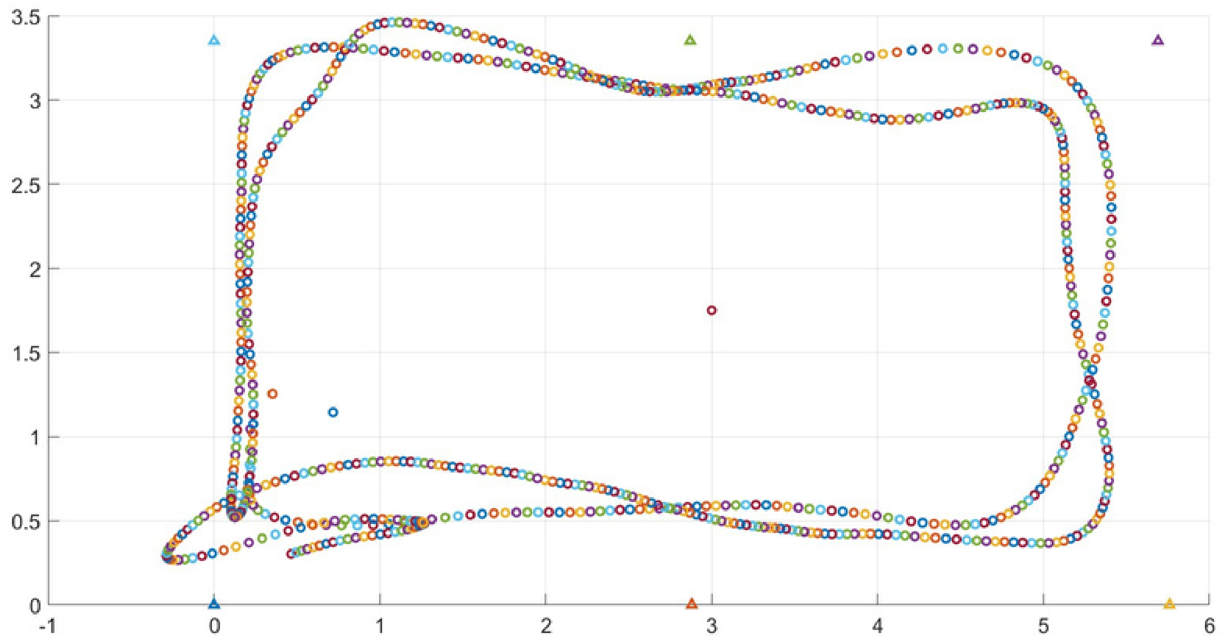


Issues and Conjectures

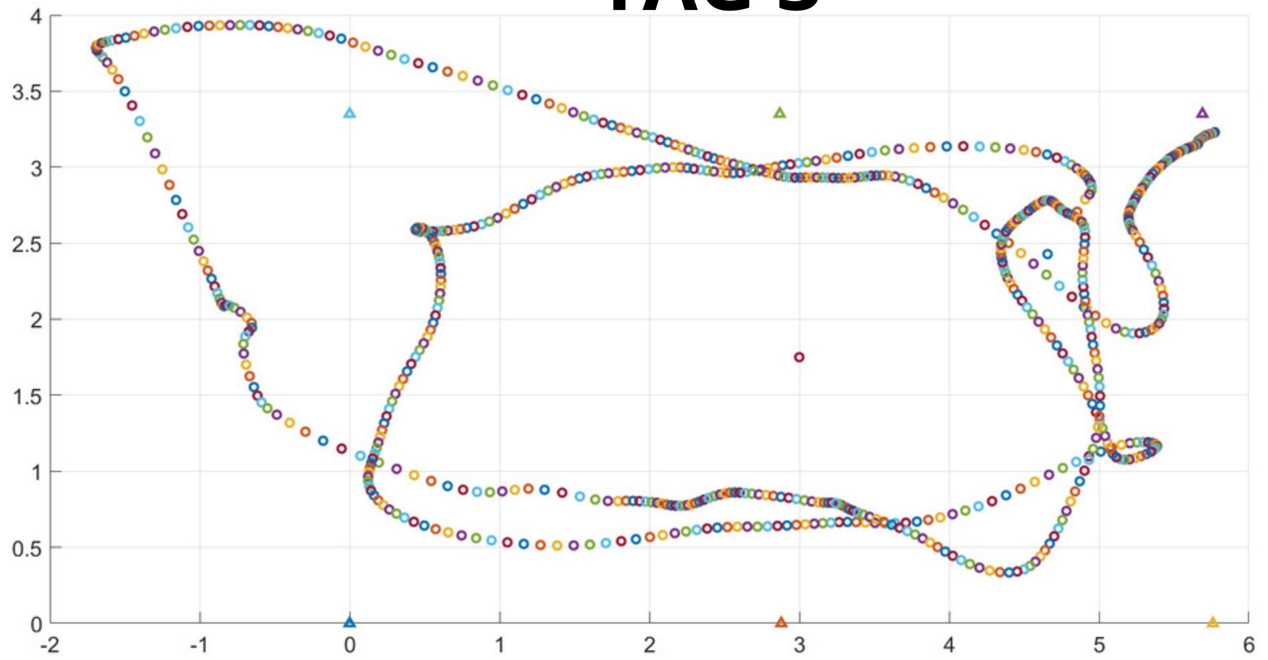
TAG 1



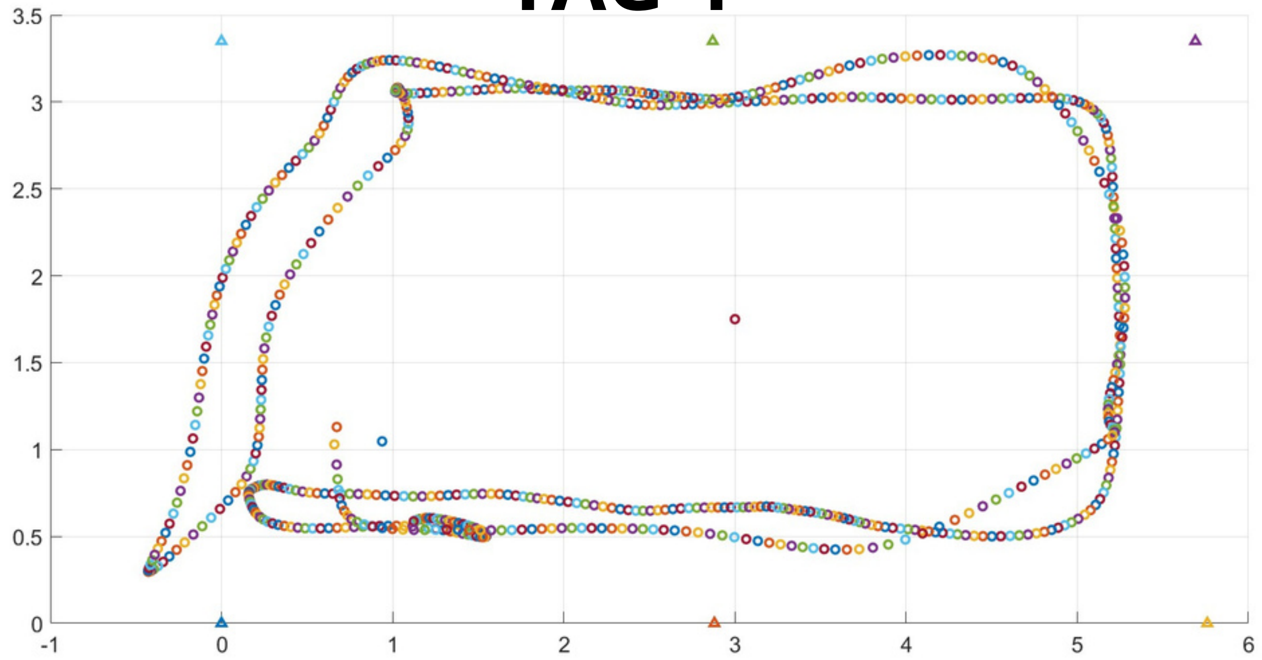
TAG 2



TAG 3



TAG 4



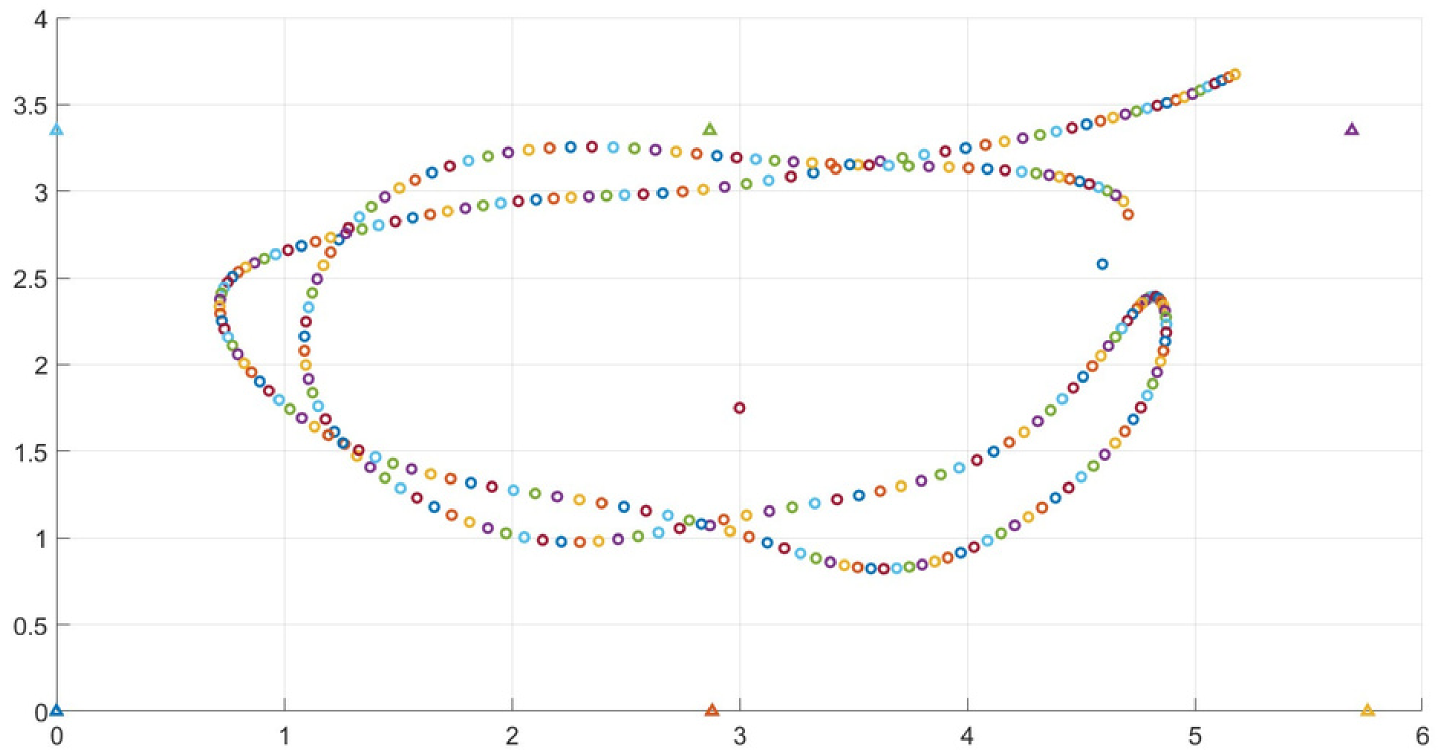
Processes tried that didn’t work:

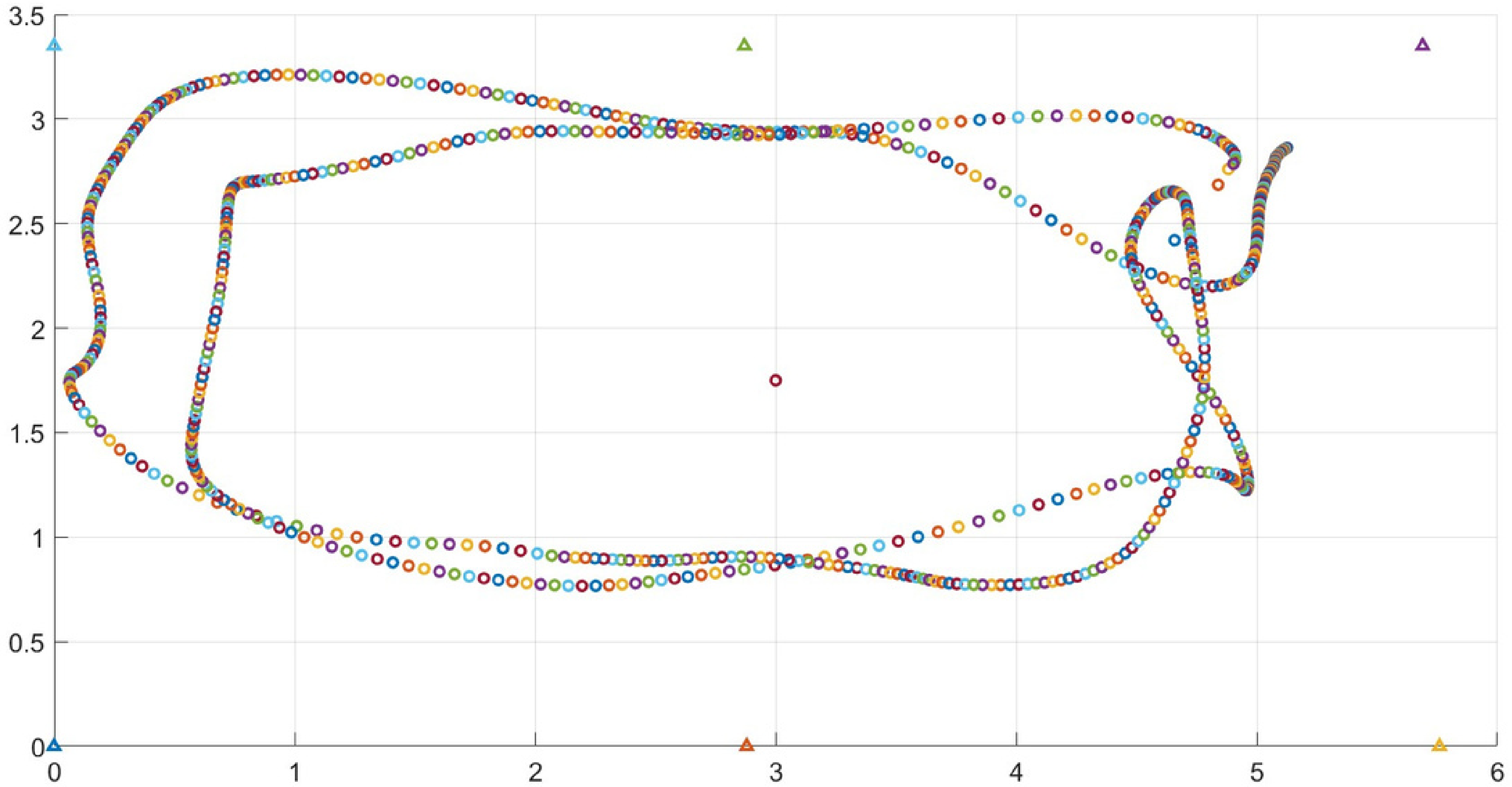
- Changing starting Position
- Changing starting Covariance
- Changing model

Processes tried that did work:

- Changing sigma_Driving

sigma_Driving changed from 0.1 to 0.05





Even though measurements are inside the expected room, an error of 0.05 m/s doesn’t seem reasonable