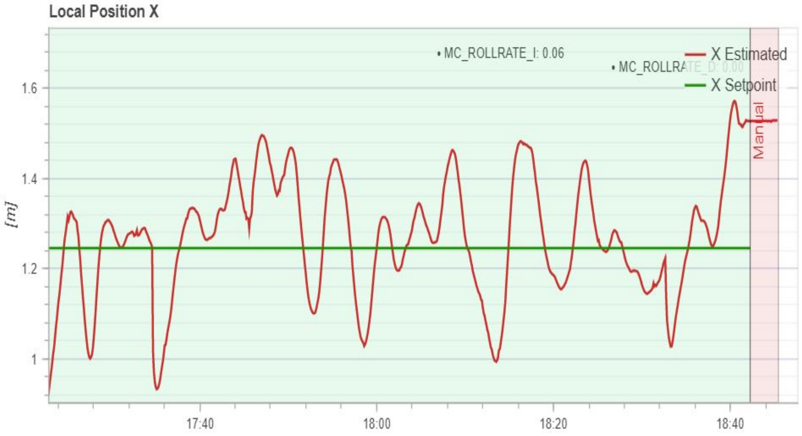
Ultrasound-inertial sensor fusion algorithm implementation

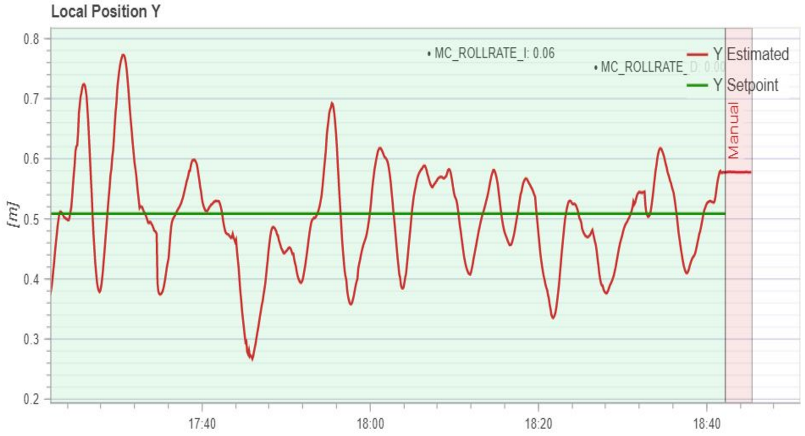
Ruslan Agishev, Marsel Faizullin, Iakov Vasilev, Grigoriy Yashin

December 21, 2018

Background

Accurate indoor localization is a challenging problem. It is worth noting that the use of one type of sensors does not allow performing either precise positioning or robust object tracking.





An example of holding position (XY) by quadrotor: error ± 0.3 m

Sensors for solution

Combination of sensors:

1. MarvelMind ultrasonic positioning system
Accuracy: ± 2 cm
Update rate: about 8 Hz

2. IMU sensor UM6-LT
Accuracy: 0.3 m/s^2,
2° roll/pitch, 5° yaw
Update rate: 20 Hz

3. **Verification** by Vicon motion capture system
Accuracy: 63+-5 microm
Update rate: 100 Hz

Problem

Sensors properties:

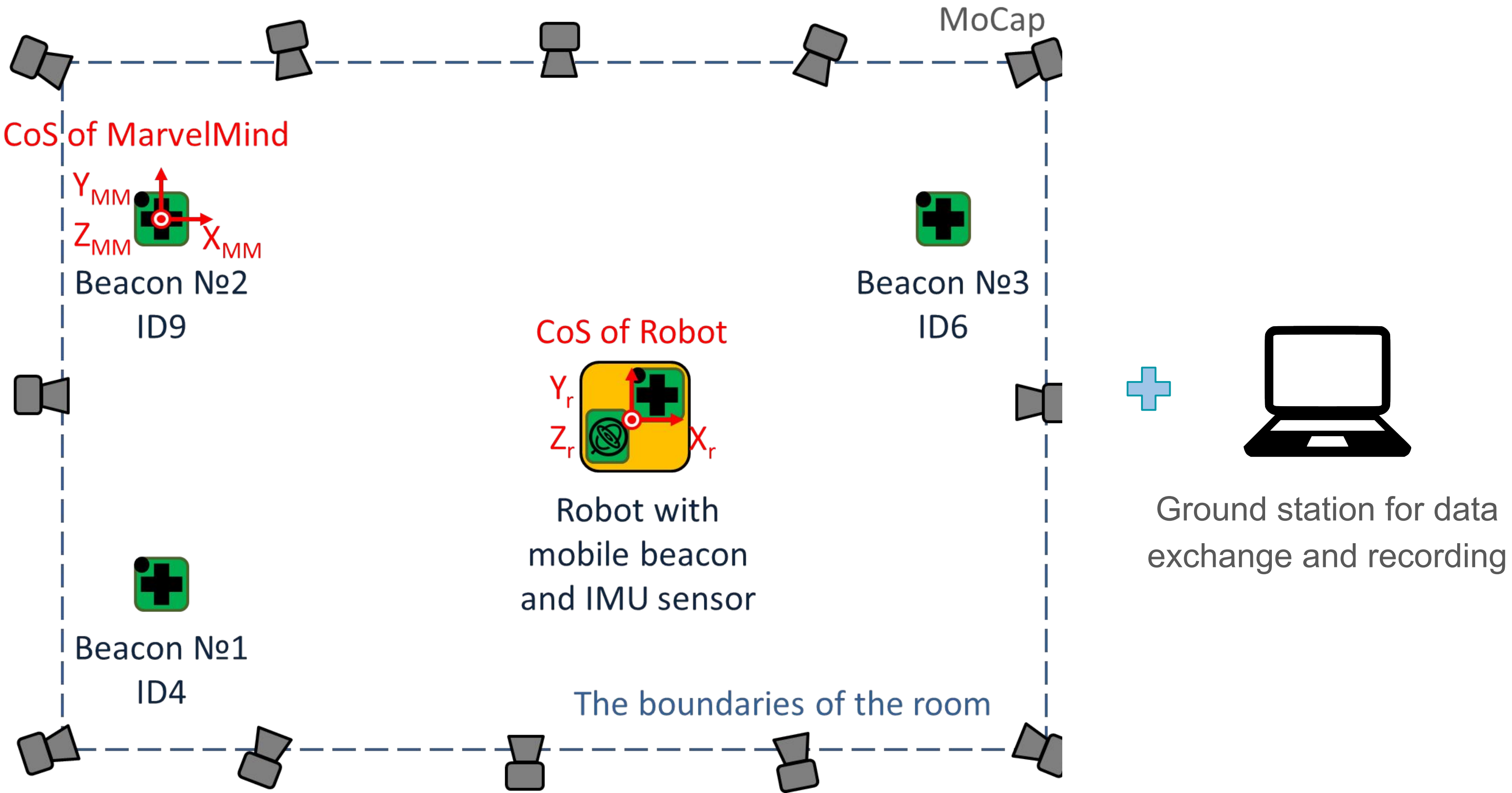
1. Low frequency of update rate of ultrasonic positioning system: [x,y,z]-position
2. Low accuracy IMU sensors: [ax, ay, az] - acceleration

Goal: robust object tracking

Approach: double step KF fusion algorithm

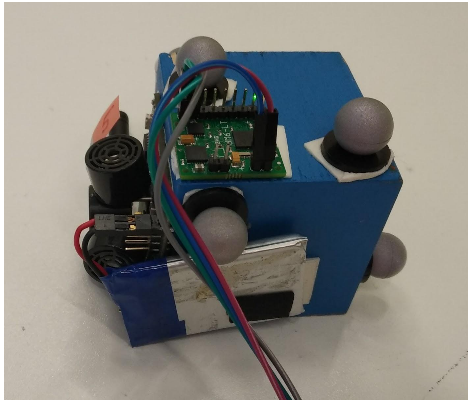
1. Use data from IMU on **prediction step**
2. Use data from Ultrasonic system on **correction step**

Description of experiment

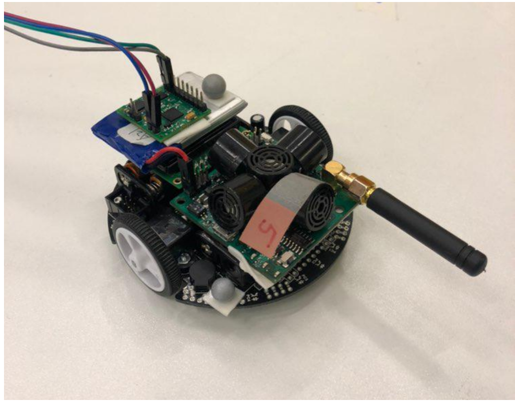


Type of tracked objects

1. Moving a cube with sensors to test the hypothesis

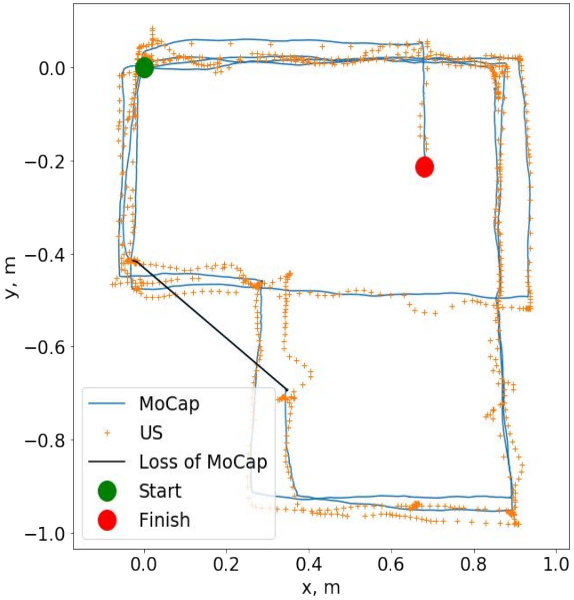


2. Wheel platform Pololu 3pi Robot

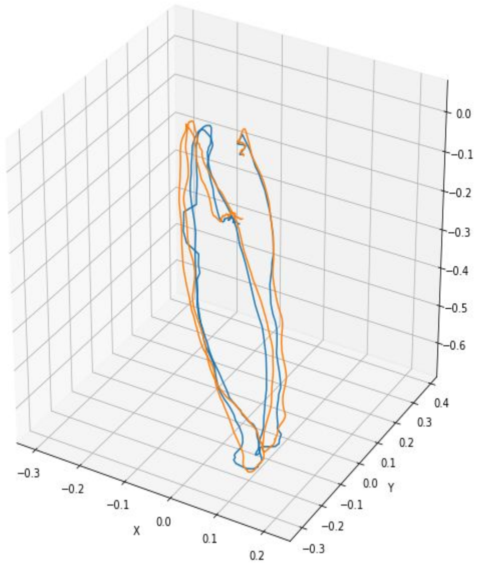


3. Quadrotor DJI F450

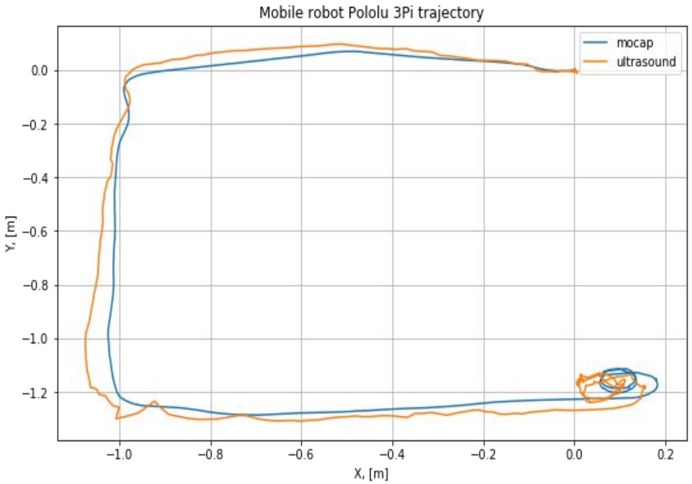
Experimental raw data



2D trajectory of cube



3D trajectory of cube



2D trajectory of Pololu Pi3

Motion model

Predict: IMU

I. $\bar{x}_k = F\,x_{k-1} + Bu_k$

$$x_k = \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} \quad F = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \quad u_k = \begin{bmatrix} a_x \\ a_y \end{bmatrix}$$

II. $\bar{P}_k = B\,P_{k-1}\,B^\top + Q$

$$Q = B \begin{bmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_a^2 \end{bmatrix} B^\top = \begin{bmatrix} \sigma_p^2 & 0 & \sigma_p\sigma_v & 0 \\ 0 & \sigma_p^2 & 0 & \sigma_p\sigma_v \\ \sigma_p\sigma_v & 0 & \sigma_v^2 & 0 \\ 0 & \sigma_p\sigma_v & 0 & \sigma_v^2 \end{bmatrix} \qquad \sigma_p = \sigma_a \frac{\Delta t^2}{2} \qquad \sigma_v = \sigma_a \Delta t$$

Update: US

III. $K_k = \bar{P}_k H^\top \left(H \bar{P}_k H^\top + R\right)$

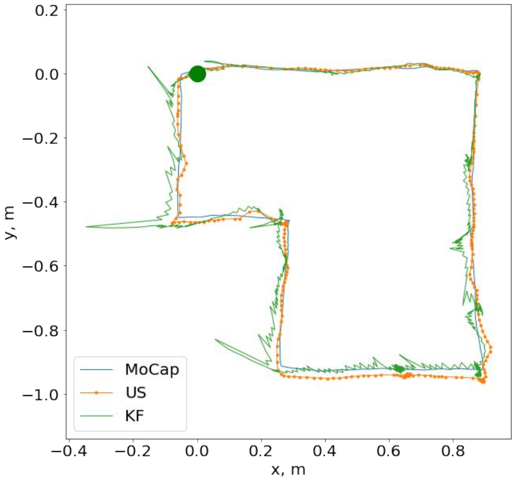
$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad R = \sigma_h^2 I_{2\times 2}$$

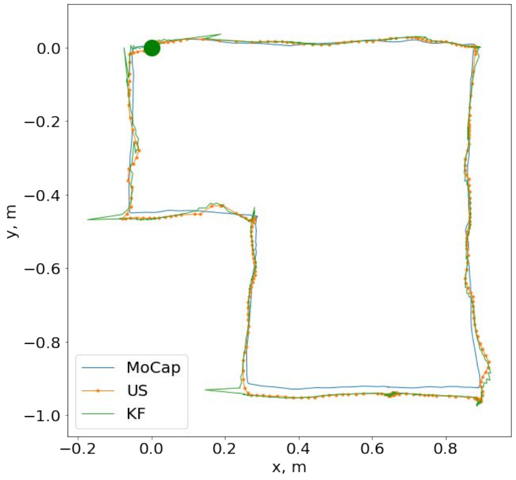
IV. $x_k = \bar{x}_k + K_k\,(z_k - H\bar{x}_k)$

V. $P_k = (I_{4\times 4} - K_k H)\,\bar{P}_k$

Results

Without bandpass filter

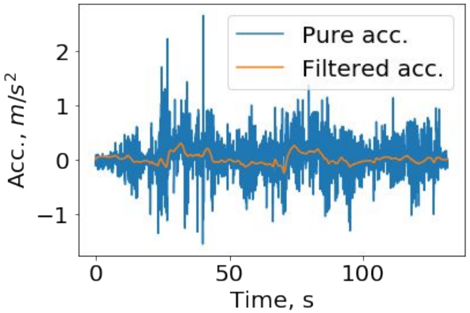




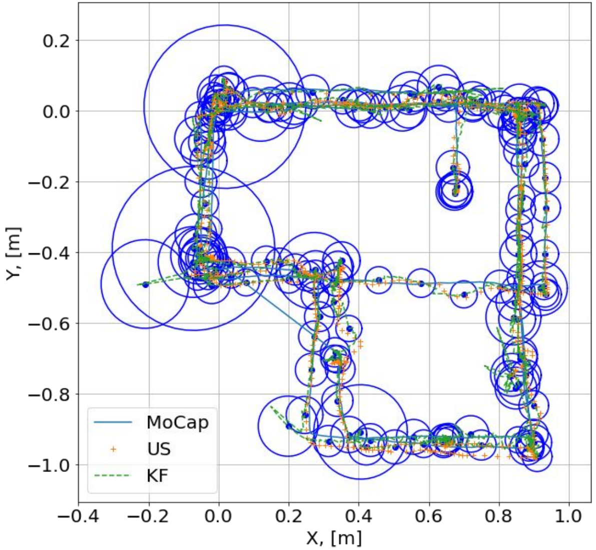
Using bandpass filter

Filter params:

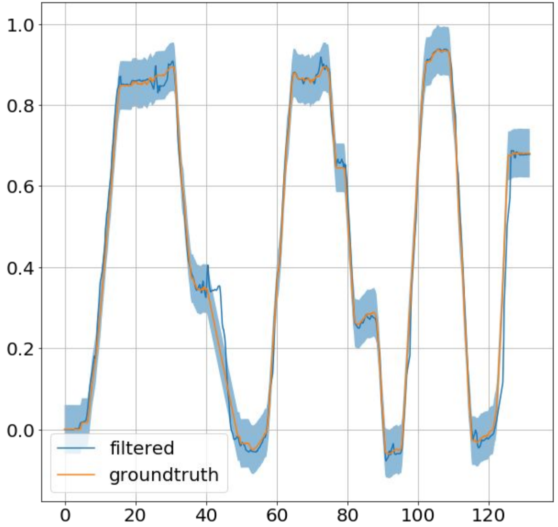
- 1-st order
- Butterworth
- 0.02 – 0.4 Hz



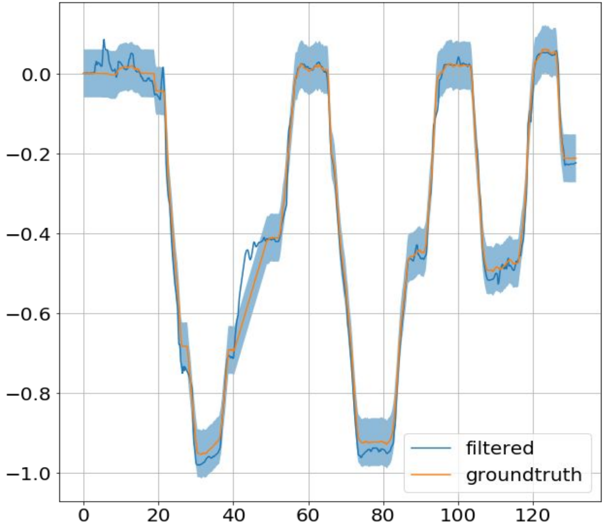
Results



Trajectory of cube with KF

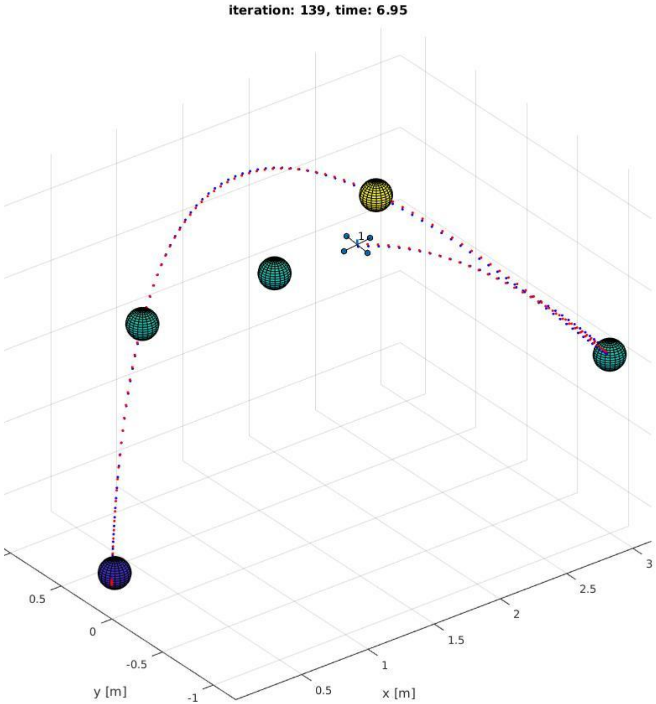


Y trajectory ± 3 σ

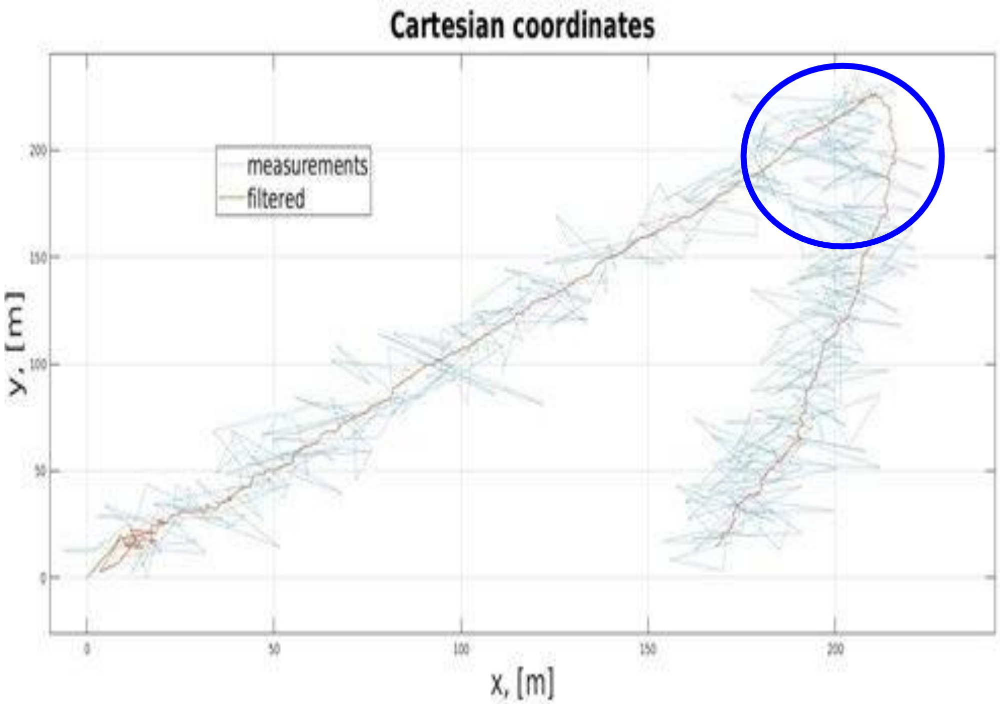


X trajectory ± 3 σ

Results



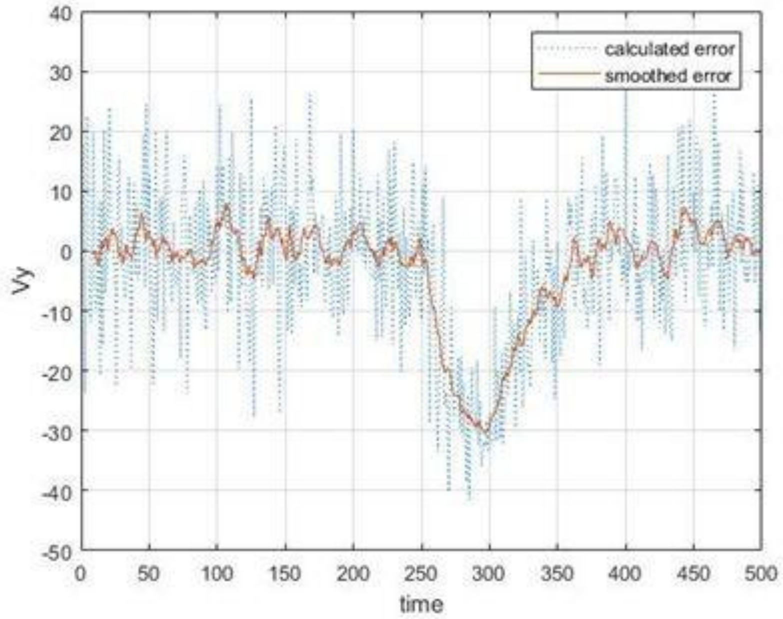
Drone simulated trajectory



Sharp moving direction change

Results

Vy = Zy - Y

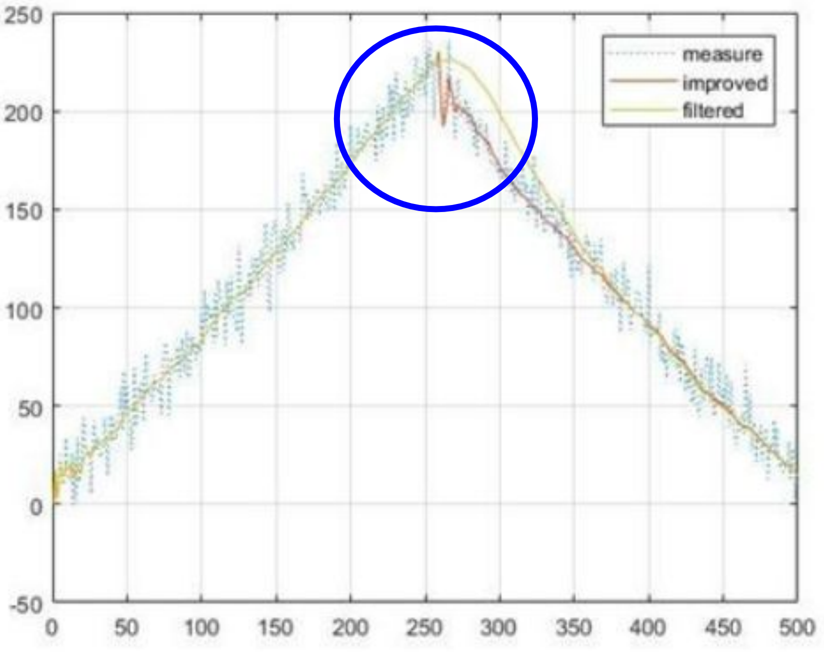


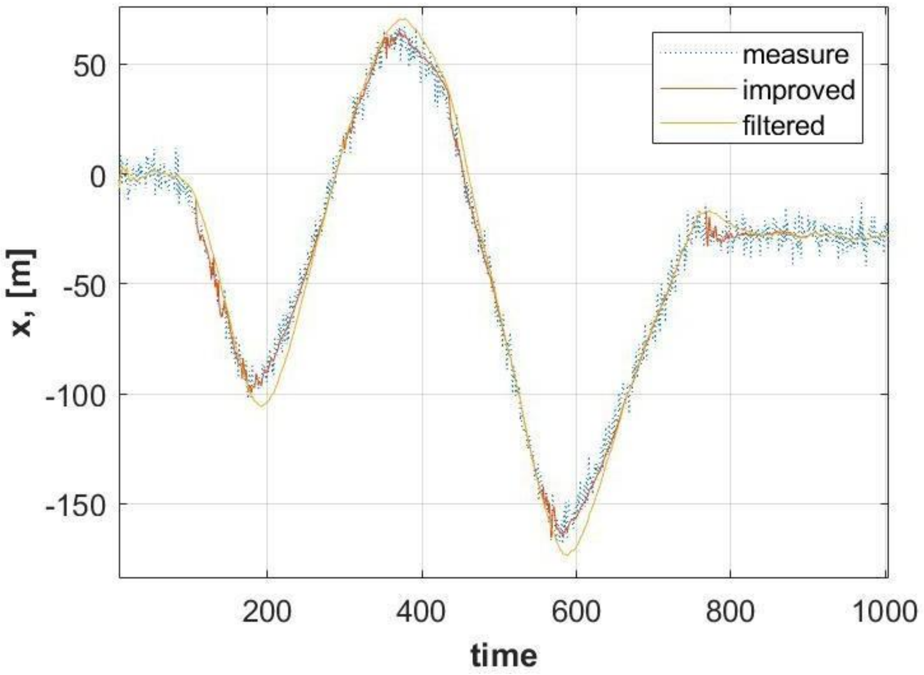
Residual's pick at the moment of turn

$$\Sigma_{update} = \begin{bmatrix} 10^5 & 0 & 0 & 0 \\ 0 & 10^5 & 0 & 0 \\ 0 & 0 & 10^5 & 0 \\ 0 & 0 & 0 & 10^5 \end{bmatrix}$$

Covariance matrix reinitialization

Results



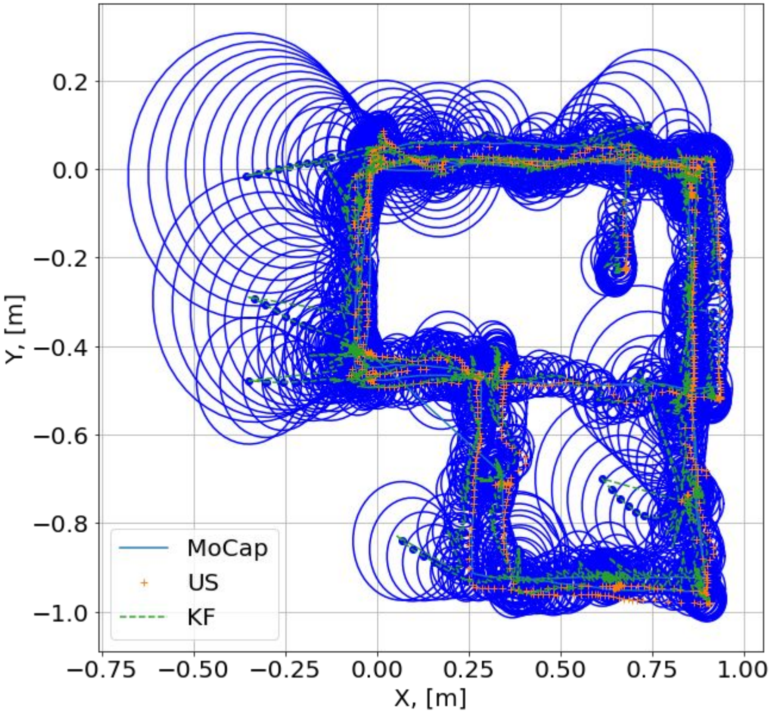


Better filtration results for sharp turns

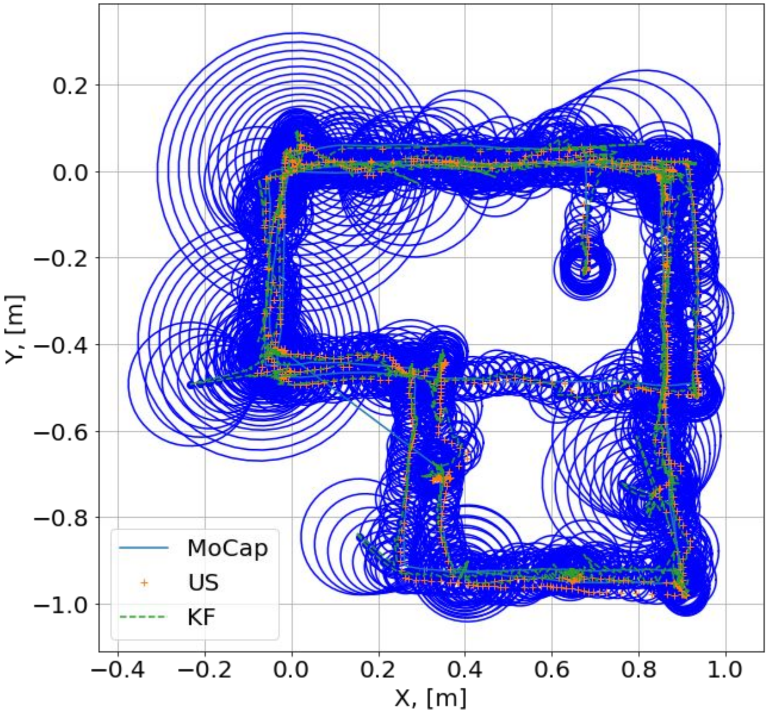
Conclusions

1. Fusion algorithm was implemented using sensors of different properties and tasks
2. Data fusion was utilized for object localization

Thank you for attention



no filt



filt