Indoor Localization with Multi-Rate Extended Kalman Filtering Based on Elisa-3

Yiting LI 5281873

January 13 2023

TUDelft

Outlines

1. Research questions
2. System Description
3. Sensor Data Fusion
4. Simulations & Next Tasks

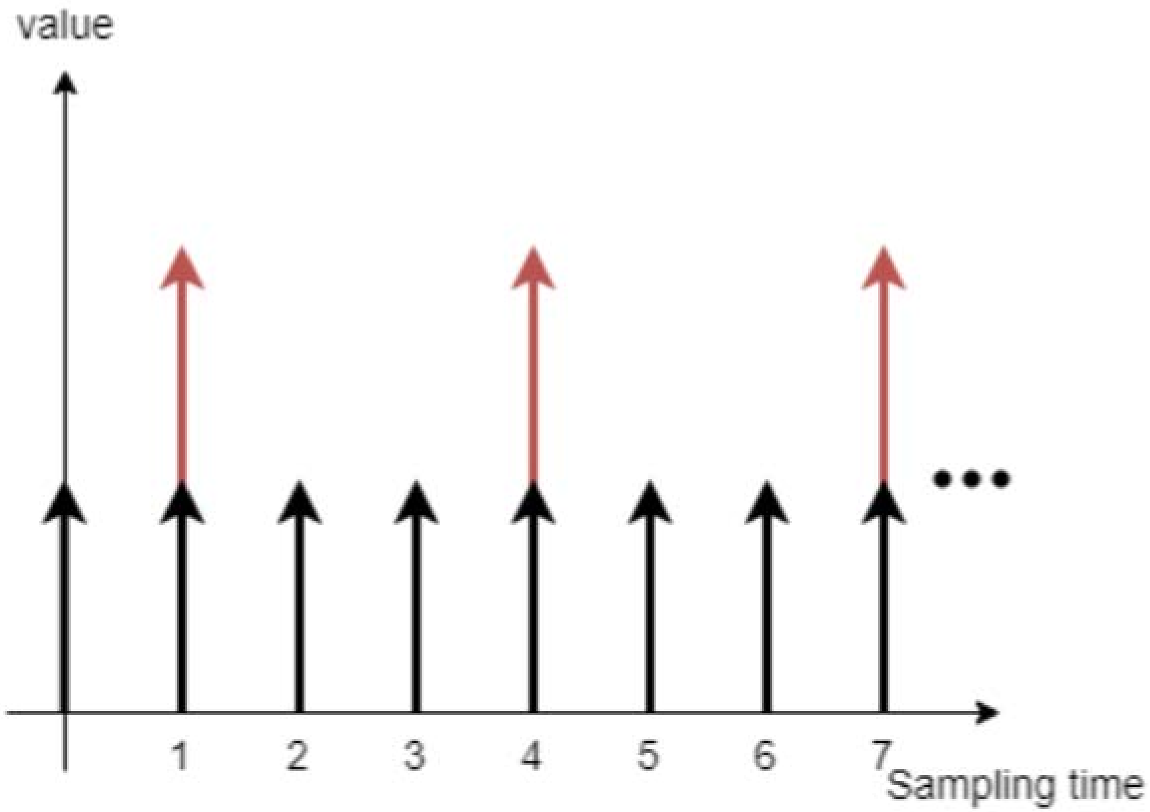
1. Research Questions

1. Localization

- Robot’s state estimation
- Multiple sensors are available, but information is not always trustful
- Data fusion methods were widely applied for state estimations
- Based on data fusion method to construct an indoor localization architecture with specific sensors coming in different sampling rate and missing or biased measurement

2. System Description

1. Multi-sensor



TUDelft

2. System Dynamic Model

- Unicycle-type Robot's Kinematic Model

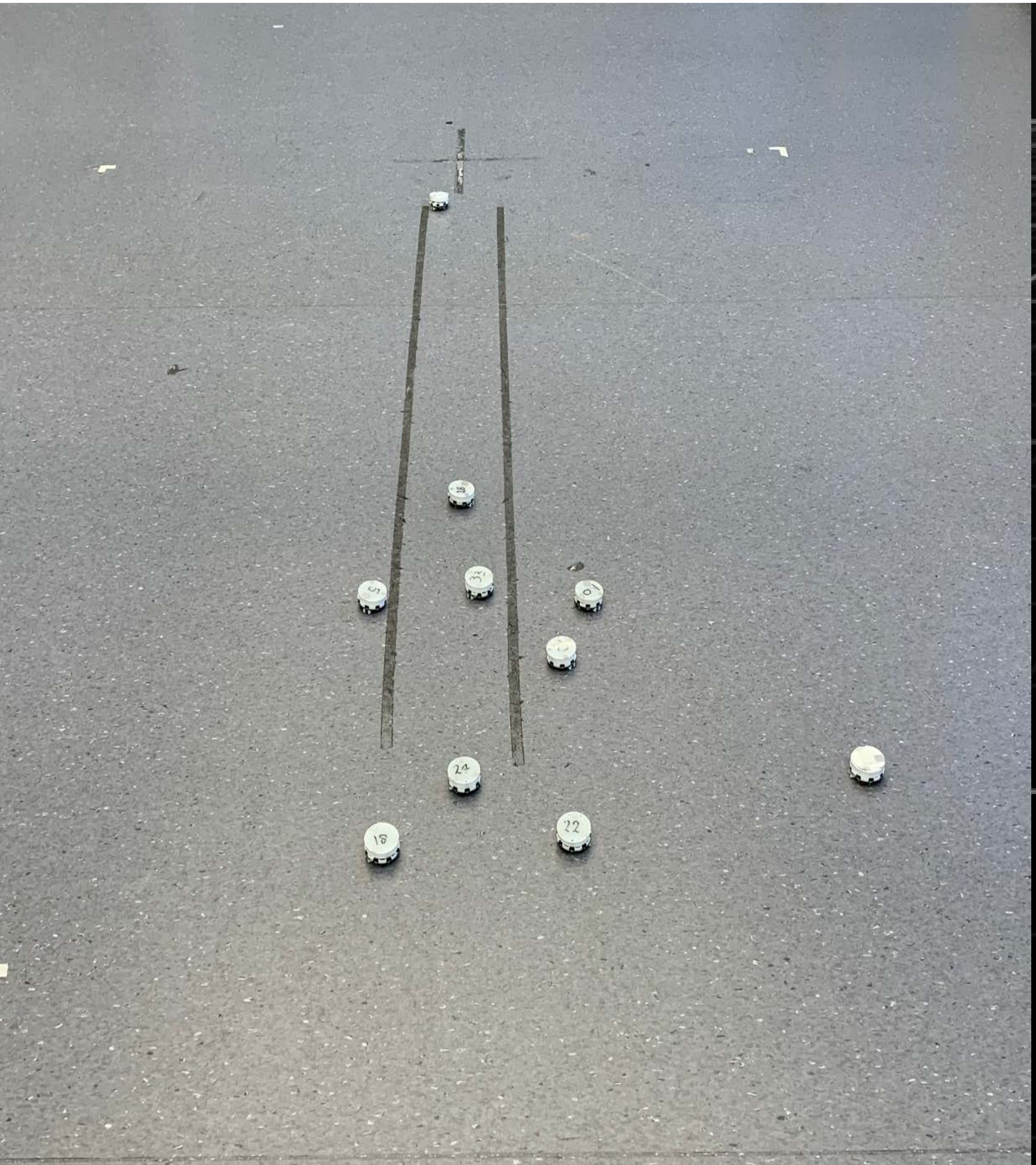
$$x(k+1) = x(k) + v_x(k)cos(\theta(k)) + w_1(k)$$
$$y(k+1) = y(k) + v_y(k)sin(\theta(k)) + w_2(k)$$
$$\theta(k+1) = \theta(k) + \omega(k) + w_3(k)$$

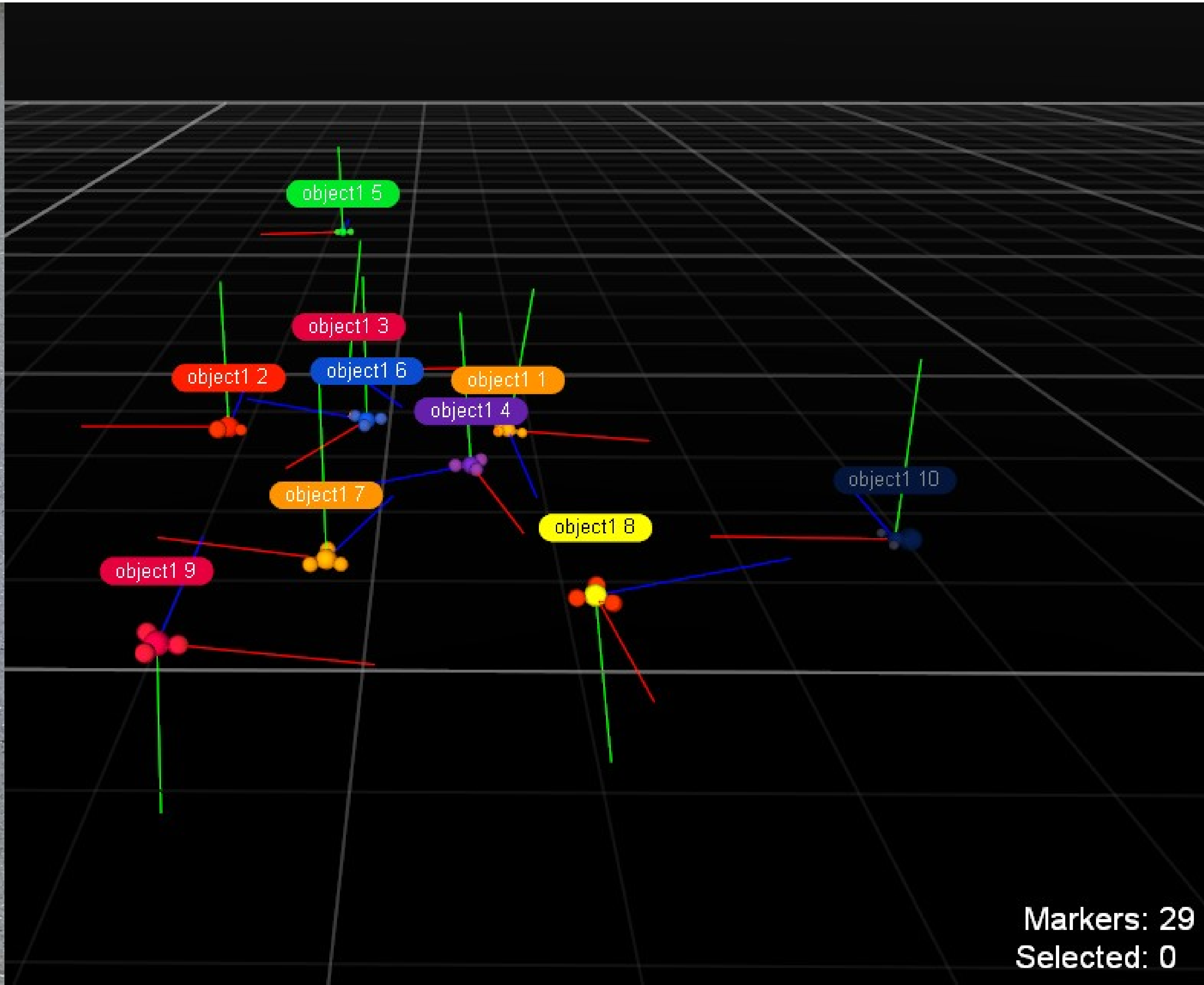
- Go-to-goal P controller

$$e(k) = [X - x(k), Y - y(k)]$$
$$V(k) = \|K_{P1}e(k)\|$$
$$\phi(k) = \arctan\left(X - x(k), Y - y(k)\right)$$
$$\omega(k) = K_{P2}\arctan(sin(\phi(k) - \theta(k)), cos(\phi(k) - \theta(k)))$$

3. Optitrack System

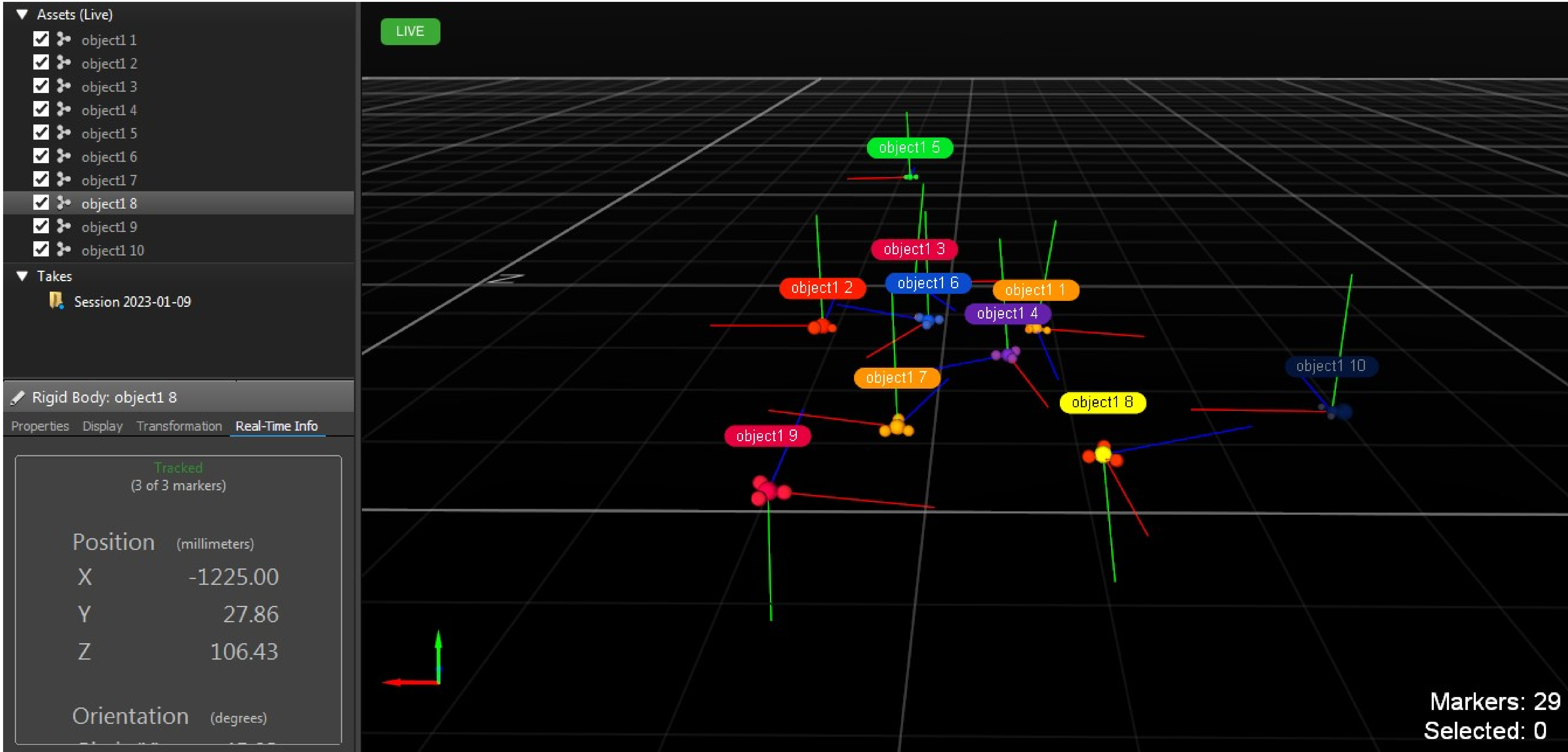
- Overview





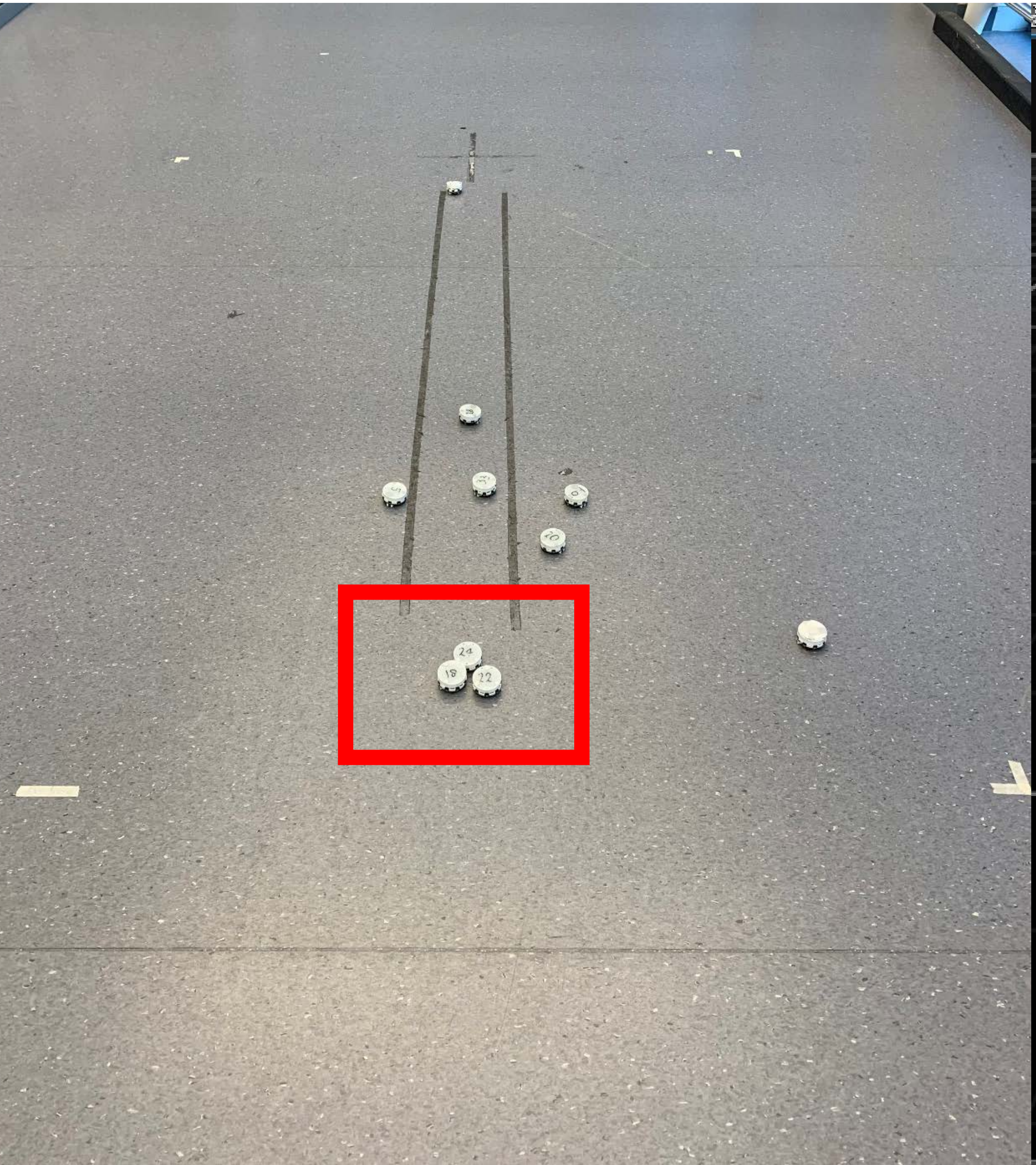
3. Optitrack System

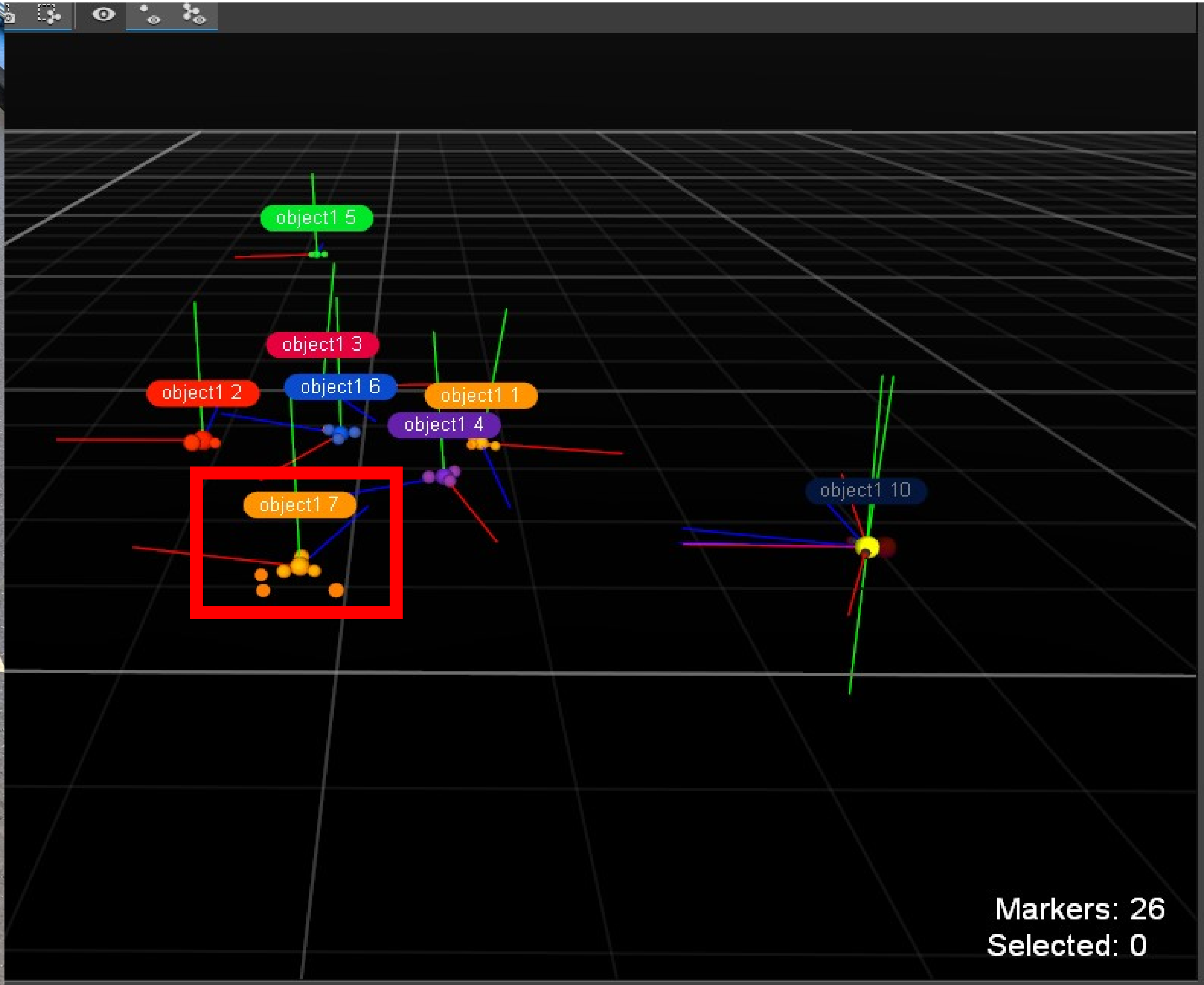
- Overview



3. Optitrack System

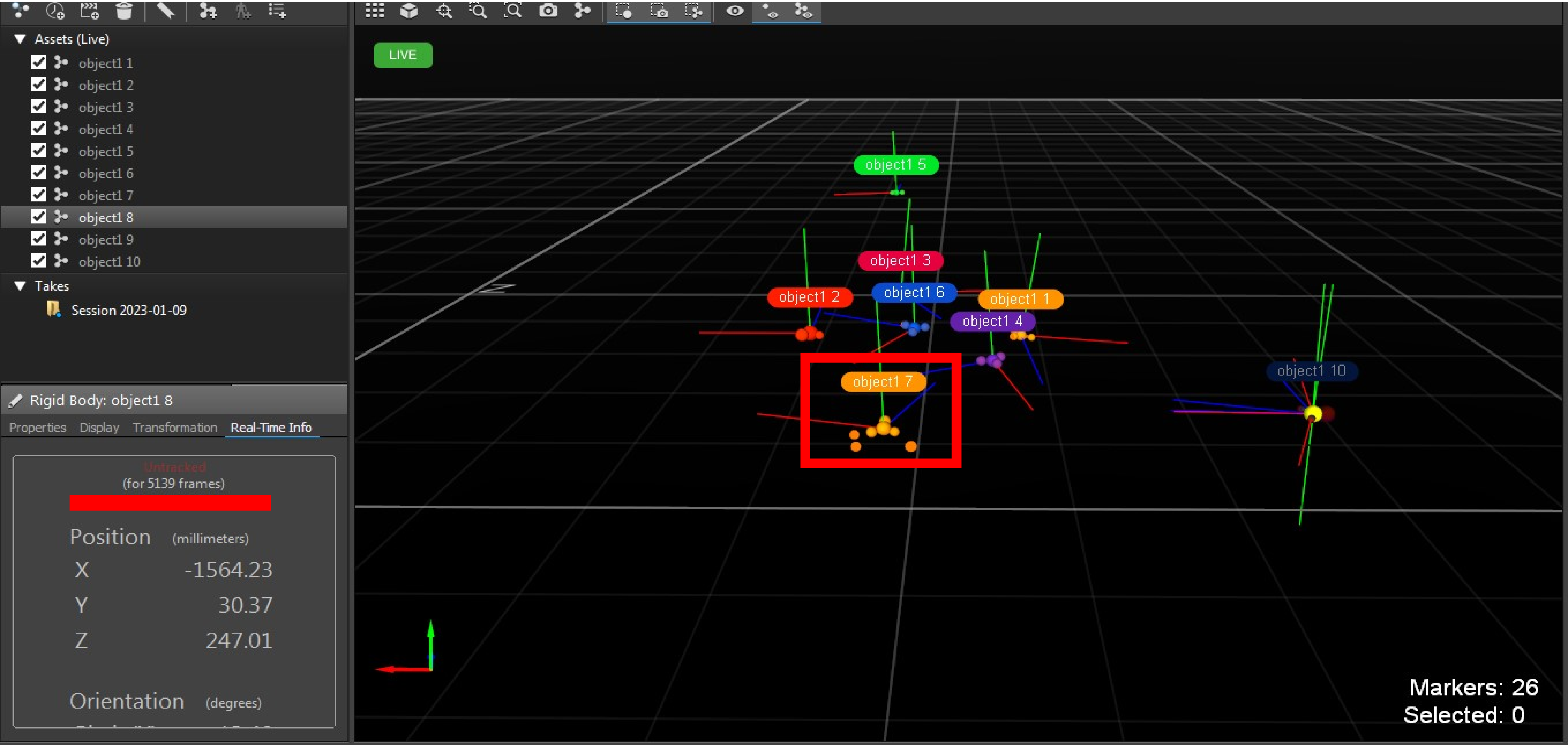
- Merging





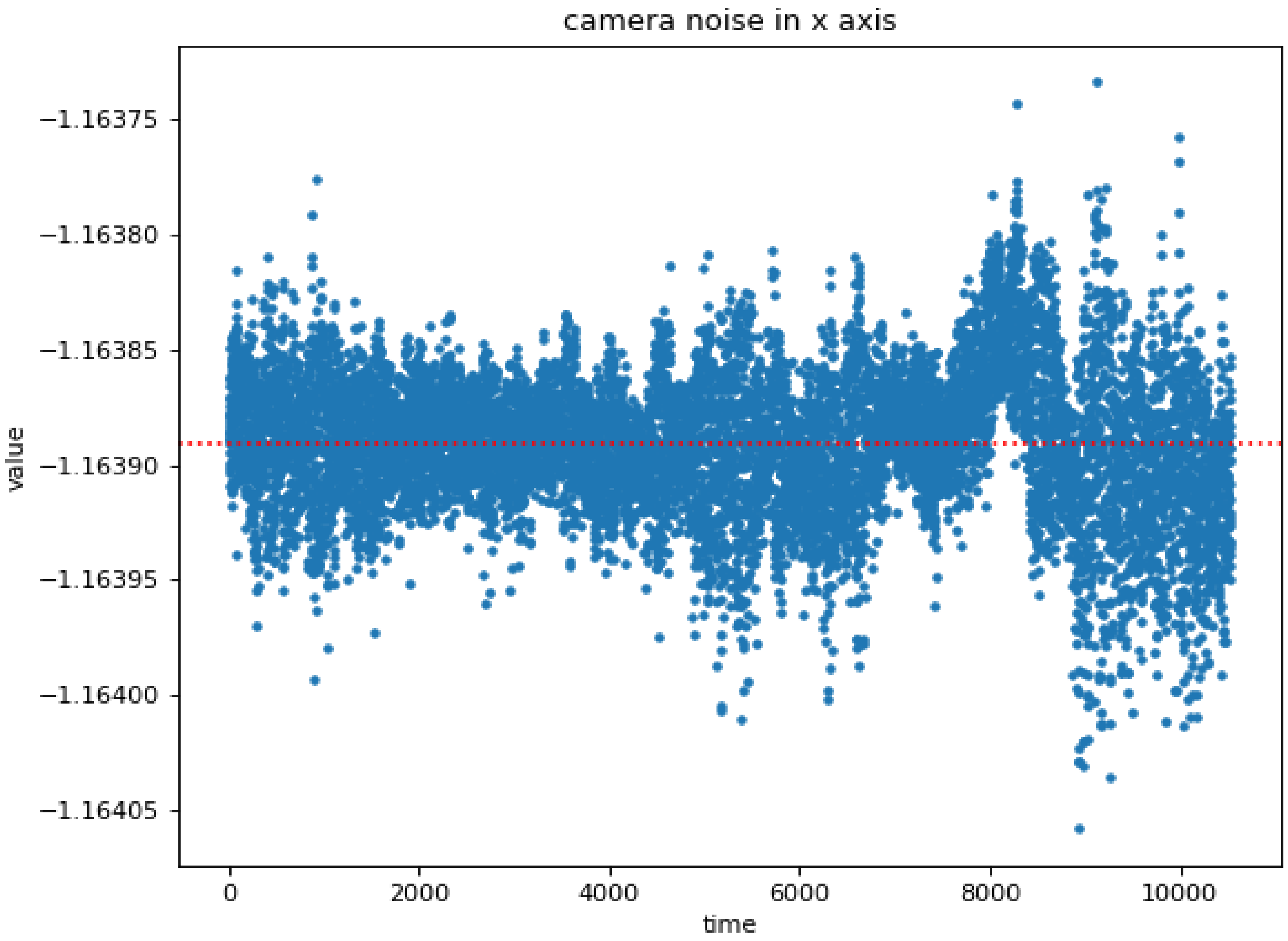
3. Optitrack System

- Merging



3. Optitrack System

- Noise



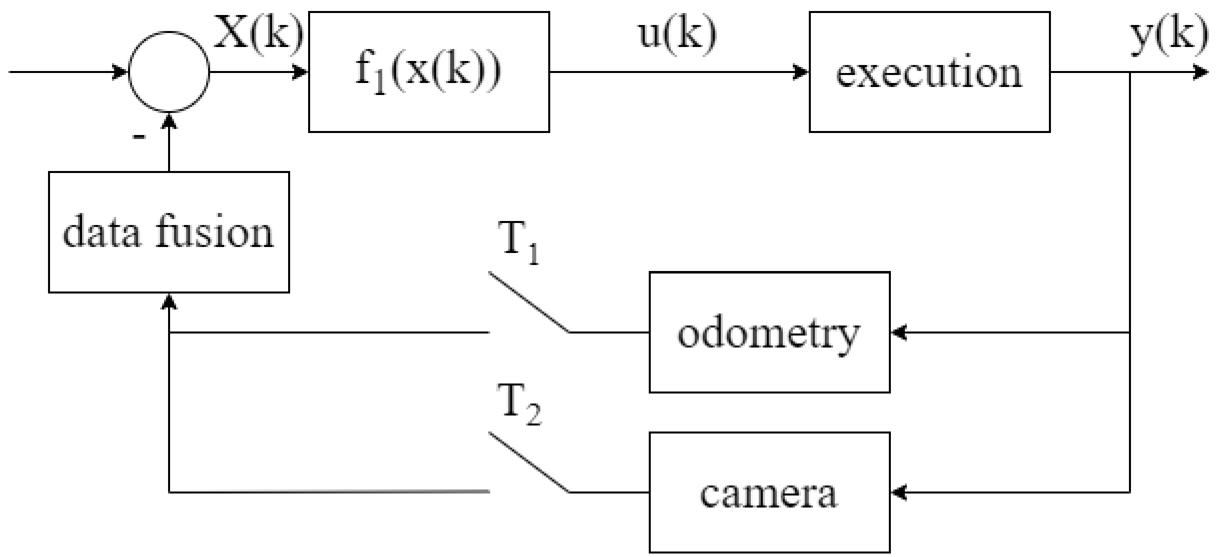
TU Delft

4. State-space model & System Block Diagram

- State-space model

$$X(k+1) = f(X(k), u(k), w(k))$$
$$Z_i(k+1) = \gamma_i(k) h_i(X(k), v(k)), i = 1, \ldots, N$$

- System Block Diagram



TU Delft

3. Sensor Data Fusion

1. Single-rate Extended Kalman Filter

prediction part:

$$\hat{\boldsymbol{x}}(k+1 \mid k) = f(X(k), u(k), w(k))$$

$$\boldsymbol{P}(k+1 \mid k) = \boldsymbol{F}(k+1)\boldsymbol{P}(k \mid k)\boldsymbol{F}(k+1)^{\top} + \boldsymbol{Q}(k+1)$$

correction part:

$$\tilde{\boldsymbol{y}}(k+1) = \boldsymbol{z}(k+1) - h(\hat{\boldsymbol{x}}(k+1 \mid k))$$

$$\boldsymbol{S}(k+1) = \boldsymbol{H}(k+1)\boldsymbol{P}(k+1 \mid k)\boldsymbol{H}(k+1)^{\top} + \boldsymbol{R}(k+1)$$

$$\boldsymbol{K}(k+1) = \boldsymbol{P}(k+1 \mid k)\boldsymbol{H}(k+1)^{\top}\boldsymbol{S}(k+1)^{-1}$$

output:

$$\hat{\boldsymbol{x}}(k+1 \mid k+1) = \hat{\boldsymbol{x}}(k+1 \mid k) + \boldsymbol{K}(k+1)\tilde{\boldsymbol{y}}(k+1)$$

2. Multi-rate Extended Kalman Filter

predicted part:

$$\hat{\mathbf{x}}(k+M \mid k) = f(X(k), u(k), w(k))$$

$$\boldsymbol{P}(k+M \mid k) = \boxed{\boldsymbol{F}(k+M)^{M}}\boldsymbol{P}(k \mid k)\boldsymbol{F}(k+M)^{M^{\top}} + \boxed{B_M \mathbf{Q}_M(k) B_M^T}$$

correction part:

$$\tilde{\mathbf{y}}(k+M) = \boldsymbol{z}(k+M) - h(\hat{\mathbf{x}}(k+M \mid k))$$

$$\mathbf{S}(k+M) = \boldsymbol{H}_i \boldsymbol{P}(k+M \mid k)\boldsymbol{H}_i^{\top} + \boldsymbol{R}_i(k+M)$$

$$\mathbf{K}(k+M) = \boldsymbol{P}(k+M \mid k)\boldsymbol{H}_i^{\top}\mathbf{S}(k+M)^{-1}$$

output:

$$\hat{\mathbf{x}}(k+M \mid k+M) = \hat{\mathbf{x}}(k+M \mid k) + \mathbf{K}(k+M)\tilde{\mathbf{y}}(k+M)$$

TU Delft

2. Multi-rate Extended Kalman Filter

- Extend the state space model

$$
\begin{aligned}
X(k+2) &= A(X(k+1)) + Bu(k+1) + w(k+1) \\
&= A(A(X(k)) + Bu(k) + w(k)) + Bu(k+1) + w(k+1) \\
&= A^2(X(k)) + \begin{bmatrix} AB & B \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \end{bmatrix} + \begin{bmatrix} A & I \end{bmatrix} \begin{bmatrix} w(k) \\ w(k+1) \end{bmatrix}
\end{aligned}
$$

$$
A^M(X(k)) + B_M B u_M(k) + B_M w_M(k)
$$

- The covariance matrix of the block noise

$$
E\left\{w_M(k) w_M^T(m)\right\} = Q_M(k)\delta(k+m)
$$

$$
Q_M(k) = \begin{bmatrix} Q(k+1) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & Q(k+M) \end{bmatrix}
$$

TU Delft

2. Multi-rate Extended Kalman Filter

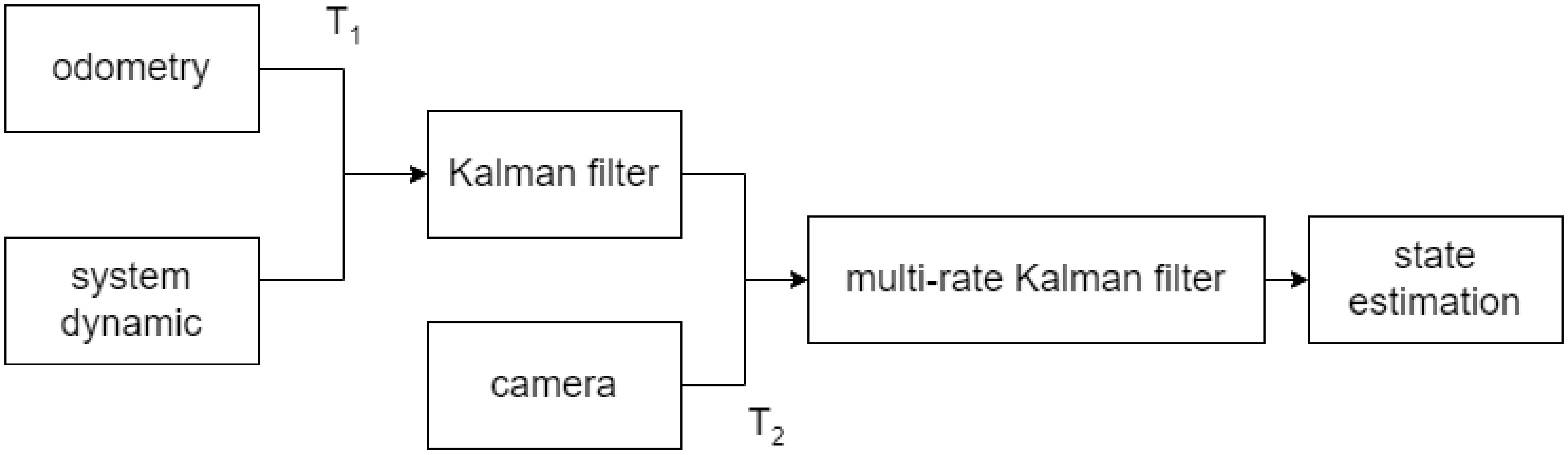
- Dealing with missing data

$$\hat{x}(k+M \mid k+M) = \hat{x}(k+M \mid k) = f(X(k), u(k), w(k))$$

$$\boldsymbol{P}(k+M \mid k+M) = \boldsymbol{P}(k+M \mid k) = \boldsymbol{F}(k+M)^M \boldsymbol{P}(k \mid k) \boldsymbol{F}(k+M)^{M^\top} + \boldsymbol{Q}_M(k+1)$$

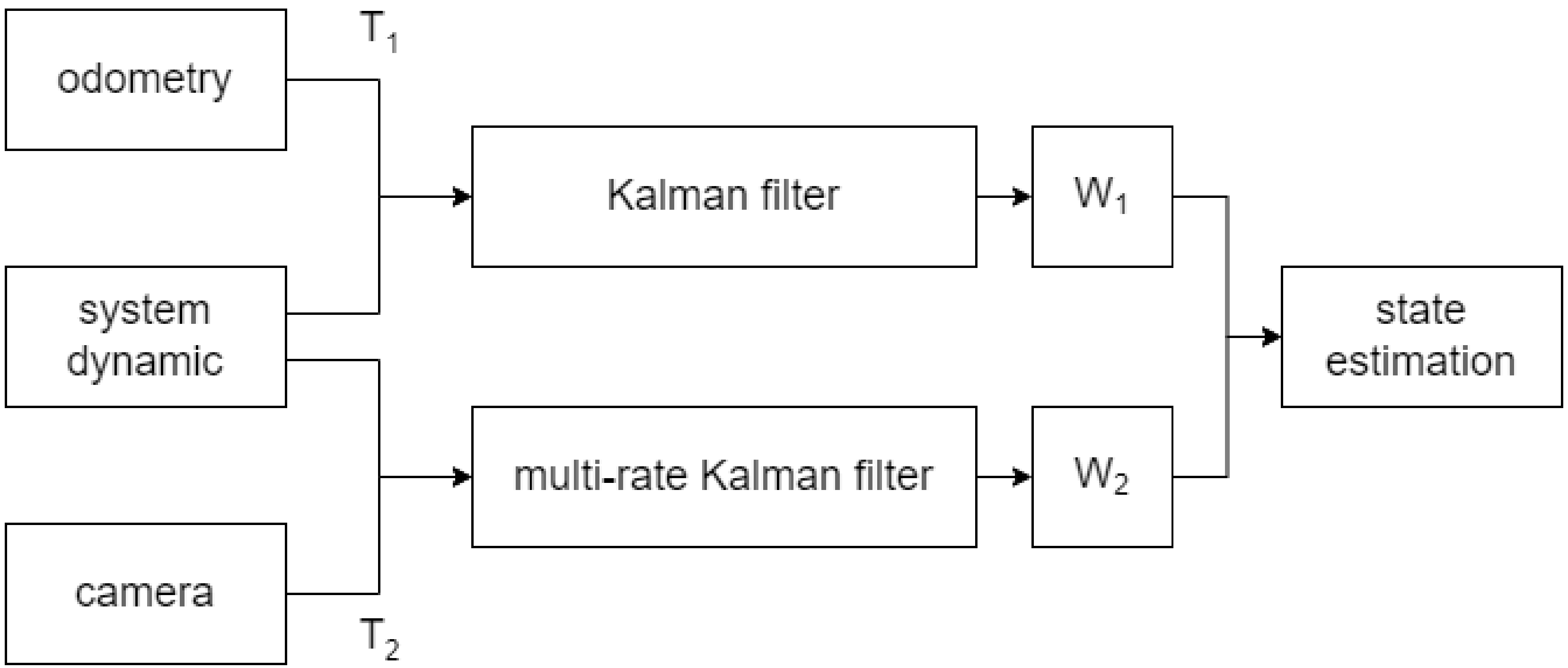
3. State Estimation Architecture: mr-EKF

- Cascade style



3. State Estimation Architecture: mr-EKF

- OWA style



3. State Estimation Architecture: mr-EKF

- OWA style, weight determination

$$W_{i,k} = \hat{C}_i^{-1}(k \mid k)(\sum_{j=1}^{N} \hat{C}_j^{-1}(k \mid k))^{-1}$$

$$\hat{C}_i^{-1}(k) = \frac{1}{l_w}\sum_{i=1}^{l_w} r_i(k+1-i) r_i(k+1-i)^T$$

$$r_i(k) = y_i(k) - H_i((\hat{x}(k \mid k))$$

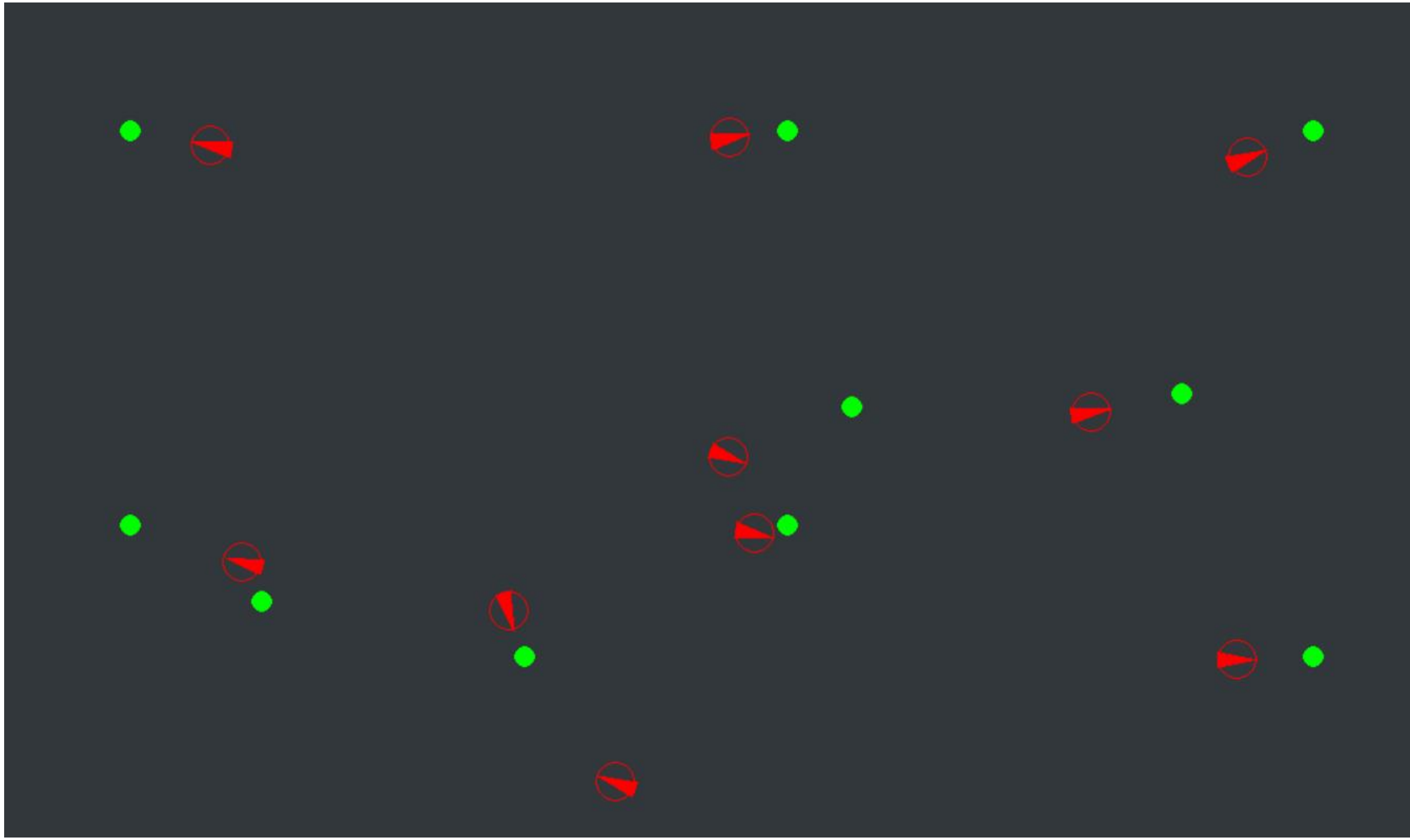
TU Delft

4. Simulation

1. Simulation Setup

- pygame
- 1080x640 ground size, 10 robots, init positions are known, randomly move
- 10 rounds
- Metrics: mean square error (MSE)

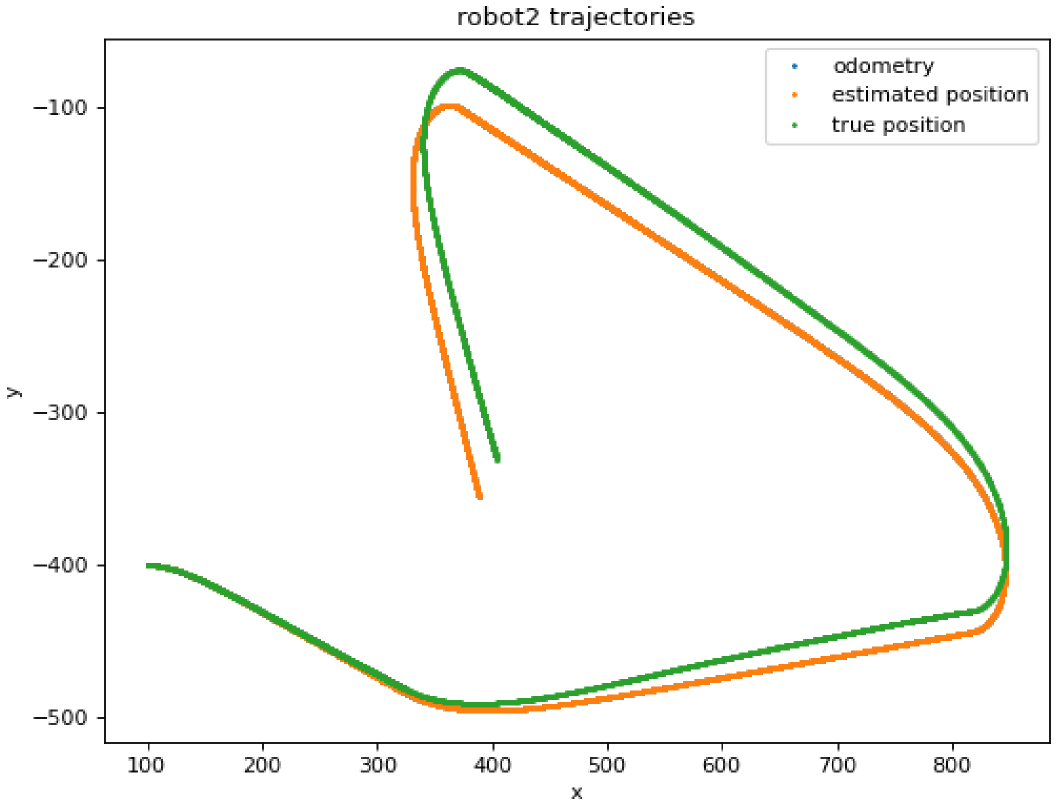
$$\mathrm{MSE} = \frac{1}{l_w}\sum_{i}^{l_w} = 1(\hat{x}_i - x_i)^2$$



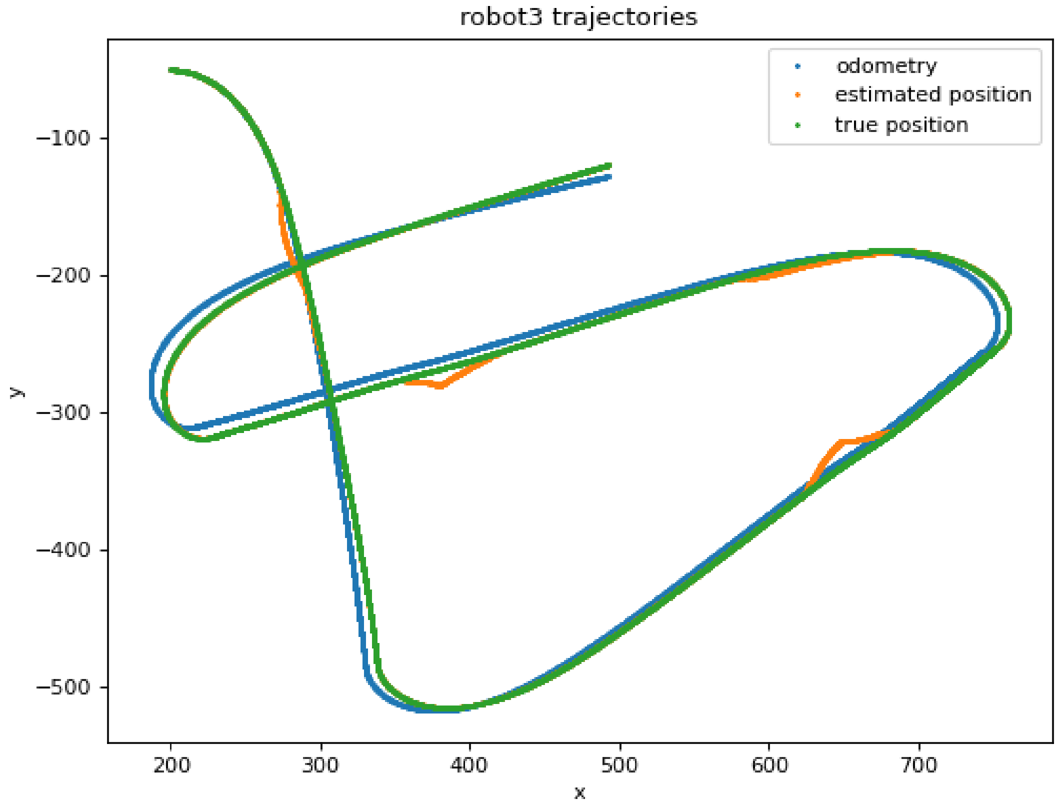
2. Result

- Trajectory

- Odometry only



- OWA mr-EKF



TUDelft

2. Result

- Metrics: mean square error (MSE) comparison

round	1	2	3	4	5	6	7	8	9	10
odometry	82.134	80.727	60.76	65.333	81.47	81.442	73.525	94.816	64.463	75.465
cascade sr-EKF	27.255	39.504	30.509	28.273	35.525	36.163	37.831	30.045	32.721	37.009
OWA sr-EKF	33.052	28.716	27.68	24.549	23.245	23.259	24.675	37.025	27.911	19.106
cascade mr-EKF	45.329	28.137	45.025	46.085	40.93	52.13	58.572	33.651	41.407	41.865
OWA mr-EKF	50.487	19.019	34.113	21.046	19.425	31.429	17.78	21.146	21.616	29.639

- Average MSE comparison(including outliers)

	odometry	cascade sr-EKF	OWA sr-EKF	cascade mr-EKF	OWA mr-EKF
MSE	76.013	33.484	26.921	43.313	26.57

3. Next Tasks

- Implementation on Elisa-3 robots
- More things

Thank you for your attention

Q & A