

# Task A. Theoretical part

## 1. Finding Q

Covariance  $Q$  characterizes the observation noise. In our case, observation is the vector:

$$z = \begin{bmatrix} bearing \\ ID \end{bmatrix}$$

Therefore, as variables in observation are not correlated:

$$Q = \begin{bmatrix} \sigma_{bearing}^2 & 0 \\ 0 & \sigma_{ID}^2 \end{bmatrix}$$

We know that observation of landmark ID has error equal to zero, and for bearing:

$$\sigma_{bearing} = 0.35$$

So we can reduce the covariance matrix to:

$$Q = [0.35^2]$$

## 2. Finding Jacobians

We have the following odometry model:

$$x = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

$$u = \begin{bmatrix} \delta_{rot1} \\ \delta_{trans} \\ \delta_{rot2} \end{bmatrix}$$

$$x_t = g(x_{t-1}, u_t, \varepsilon_t)|_{\varepsilon_t=0} = \begin{bmatrix} x_{t-1} + \delta_{trans} \cos(\theta + \delta_{rot1}) \\ y_{t-1} + \delta_{trans} \sin(\theta + \delta_{rot1}) \\ \theta_{t-1} + \delta_{rot1} + \delta_{rot2} \end{bmatrix}$$

According to it we can find state and control jacobians:

$$G_t = \frac{\partial g(x_{t-1}, u_t, \varepsilon_t)}{\partial x_{t-1}} \Big|_{\mu_{t-1}, \varepsilon_t=0} = \begin{bmatrix} 1 & 0 & -\delta_{trans} \sin(\theta + \delta_{rot1}) \\ 0 & 1 & \delta_{trans} \cos(\theta + \delta_{rot1}) \\ 0 & 0 & 1 \end{bmatrix}$$

$$V_t = \frac{\partial g(x_{t-1}, u_t, \varepsilon_t)}{\partial u_t} \Big|_{\mu_{t-1}, \varepsilon_t=0} = \begin{bmatrix} -\delta_{trans} \sin(\theta + \delta_{rot1}) & \cos(\theta + \delta_{rot1}) & 0 \\ \delta_{trans} \cos(\theta + \delta_{rot1}) & \sin(\theta + \delta_{rot1}) & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Our observations can be converted to features of landmarks:

$$landmark_i = [m_{i,x} \quad m_{i,y}]^T$$

$$f_i = \begin{bmatrix} range \\ bearing \\ signature \end{bmatrix} = \begin{bmatrix} r_{i,t} \\ \phi_{i,t} \\ s_{i,t} \end{bmatrix} = \begin{bmatrix} \sqrt{(m_{i,x} - x_t)^2 + (m_{i,y} - y_t)^2} \\ atan2(m_{i,x} - x_t, m_{i,y} - y_t) - \theta \\ 0 \end{bmatrix}$$

and observation jacobian is:

$$H_t = \frac{\partial g(x_{t-1}, x_t, \varepsilon_t)}{\partial x_t} \Big|_{\mu_{t-1}, \varepsilon_t=0} = \begin{bmatrix} -\frac{m_{i,x} - \bar{\mu}_{t,x}}{r_{i,t}} & \frac{m_{i,y} - \bar{\mu}_{t,y}}{r_{i,t}} & 0 \\ \frac{m_{i,x} - \bar{\mu}_{t,x}}{r_{i,t}^2} & -\frac{m_{i,y} - \bar{\mu}_{t,y}}{r_{i,t}^2} & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

Reducing to bearing only:

$$H_t = \begin{bmatrix} \frac{m_{i,x} - \bar{\mu}_{t,x}}{r_{i,t}^2} & -\frac{m_{i,y} - \bar{\mu}_{t,y}}{r_{i,t}^2} & -1 \end{bmatrix}$$

Given the initial mean state and the initial control input:

$$\mu_1 = [180 \quad 50 \quad 5]^T$$

$$u = [0 \quad 10 \quad 0]^T$$

we obtain the following values of jacobians:

$$G_t^{\mu_1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{bmatrix}$$

$$V_t^{\mu_1} = \begin{bmatrix} 0 & 1 & 0 \\ 10 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

### 3. Finding R

Covariance  $R$  characterizes the noise added to transition function and can be found via following equation:

$$R_t = V_t M V_t^T$$

where M is covariance of control noise and is found as following:

$$M = \begin{bmatrix} \alpha_1 \delta_{rot1}^2 + \alpha_2 \delta_{trans}^2 & 0 & 0 \\ 0 & \alpha_3 \delta_{trans}^2 + \alpha_4 (\delta_{rot1}^2 + \delta_{rot2}^2) & 0 \\ 0 & 0 & \alpha_1 \delta_{rot2}^2 + \alpha_2 \delta_{trans}^2 \end{bmatrix}$$

$$\alpha_1 = 0.05^2$$

$$\alpha_2 = 0.001^2$$

$$\alpha_3 = 0.05^2$$

$$\alpha_4 = 0.01^2$$

## Task B. Filters implementation

Implementations of EKF and PF are located in 'filters/ekf.py' and 'filters/pf.py' files respectively. Resulting data is located in 'pf\_data' and 'ekf\_data' folders with corresponding videos in 'video' folder. A proof that robot lies within  $3\sigma$  ellipse more than

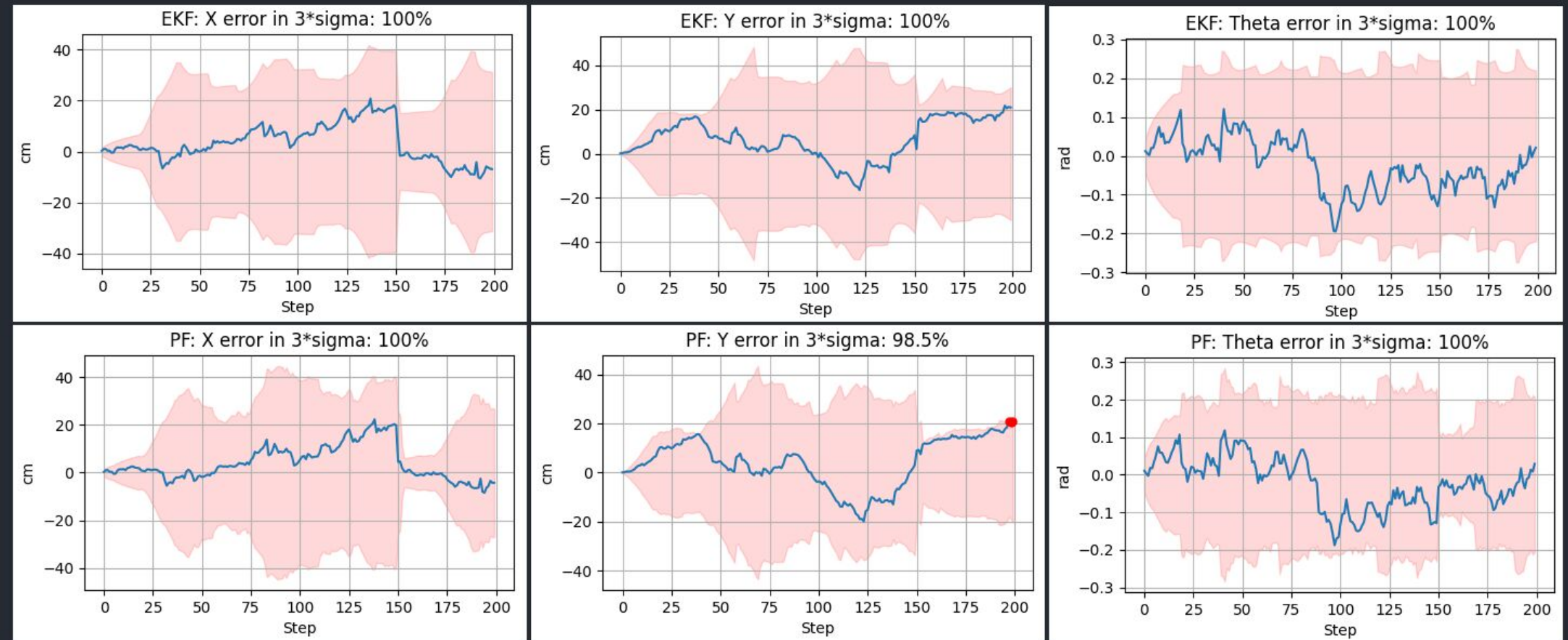
- 98.89% of the time for EKF,
- 99.73% of the time for PF

will be given in the next part of the report.

## Task C. Evaluation of errors

Source code for plotting errors versus time is in 'taskC.py' file. Red area on plots stands for  $3\sigma$  ellipse range. Blue line is state error with red markers indicating spots where it is out of the ellipse.

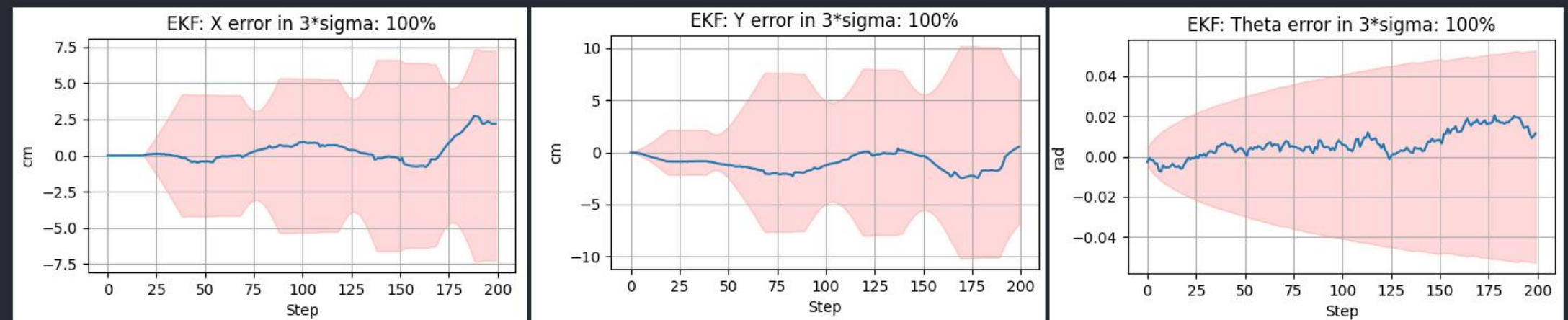
The following figures demonstrate results of errors evaluation.



# Task D. Filter properties

## 1. Sensor or motion noise goes towards zero (EKF)

Sensor motion approaches zero:



Sensor motion approaches zero:



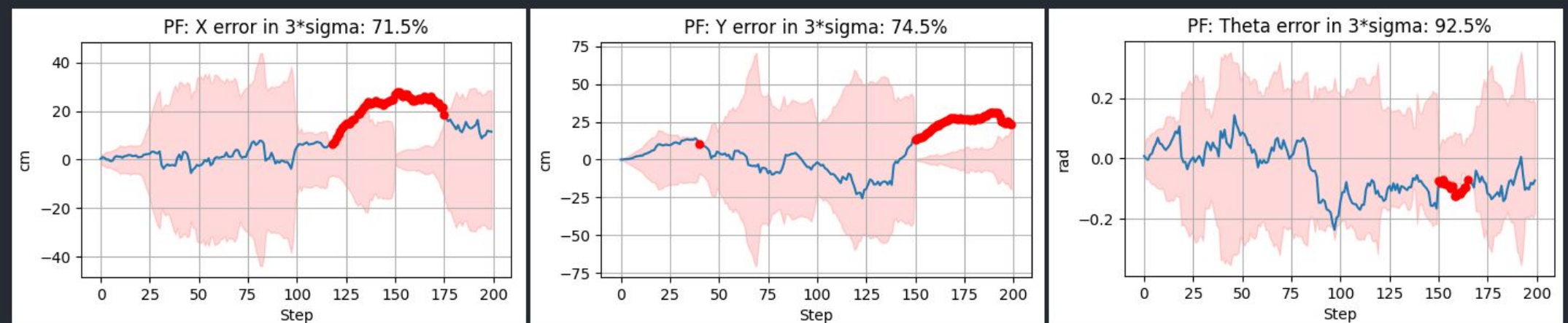


In both cases we observe a significant decrease of error value. That happens because if motion or state noise is close to zero, Kalman filter changes its gain so this value becomes "ground truth" for it. Then KF:

- discards prediction results (robot moves according to the sensor data) if state noise is zero
- discards update results (robot moves according to the math model) if motion noise is zero

## 2. Small number of particles (PF)

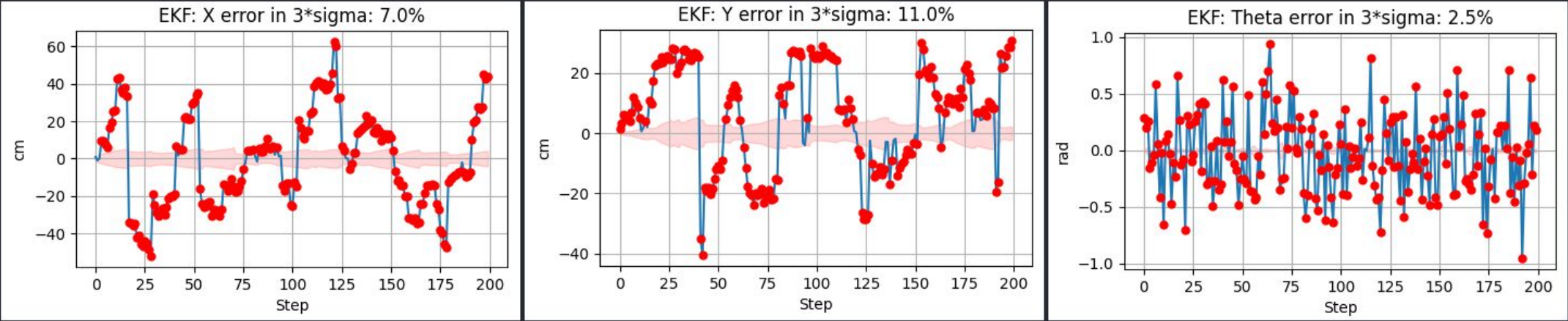
In this experiment particle filter has performed with 20 particles. Below there are error plots similar to ones in Task C, but for such filter.



We can see huge increase of incorrect estimations (ones outside of  $3\sigma$  ellipse) of robot position for every parameter.

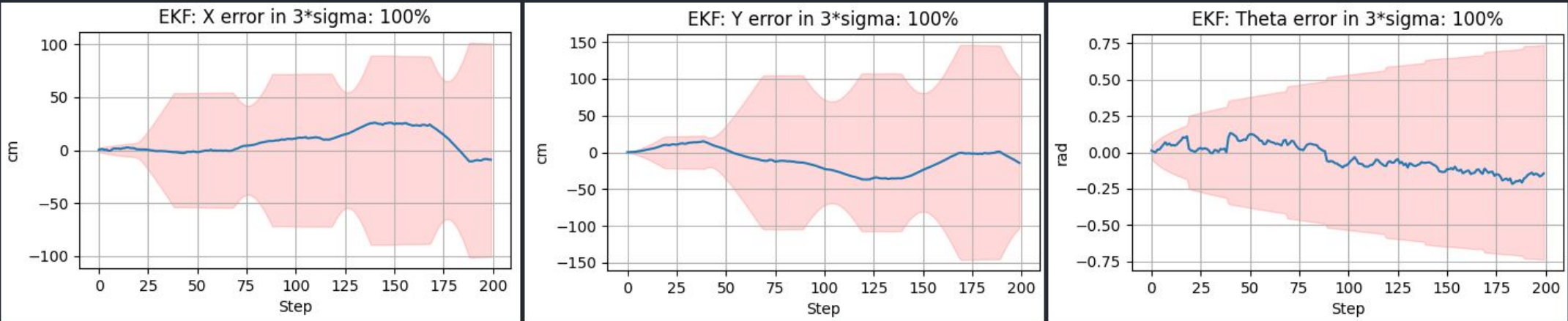
### 3. Noise parameters different from true ones (EKF)

Underestimation of true noise - when true noise covariance is bigger than the estimated one.



Evaluated covariance stays normal, however, error goes out of  $3\sigma$  ellipse almost on every single point of the path.

Overestimation of true noise - when true noise covariance is smaller than the estimated one.



Error always stays inside  $3\sigma$  (actually even inside  $1\sigma$ ) ellipse. However, evaluated covariance becomes significantly bigger.