Robot Localisation

Extended Kalman Filter

Feras Dayoub

Learning objectives

- Extended Kalman filter.
- Landmark-based localization.
- Range and bearing sensors.

The Kalman Filter Steps

Prediction:

$$\bar{\mu}_t = \mathbf{A}\mu_{x_{t-1}} + \mathbf{B}\mathbf{u}_t$$

$$\bar{\Sigma}_t = \mathbf{A}\Sigma_{t-1}\mathbf{A}^T + \mathbf{R}$$

Update/Correction:

$$\mu_t = \bar{\mu}_t + \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}\bar{\mu}_t)$$

$$\mathbf{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\bar{\mathbf{\Sigma}}_t$$

$$\mathbf{K}_t = \bar{\mathbf{\Sigma}}_t\mathbf{H}^T(\mathbf{H}_t\bar{\mathbf{\Sigma}}_t\mathbf{H}^T + \mathbf{Q})^{-1}$$

Kalman filter assumptions

- The state and the noise are normally distributed.
- The motion and the measurement models are linear.

This lecture tackles the case when this is not true.

If the above is met then Kalman filter is the optimal solution!

Odometry-based state transition function

$f(\mathbf{x}_{t-1}, \mathbf{u}_t)$ Nonlinear

Distance traveled between t-1 and t

$$\mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + \delta d \times cos(\theta_{t-1}) \\ y_{t-1} + \delta d \times sin(\theta_{t-1}) \\ \theta_{t-1} + \delta\theta \end{bmatrix}$$

Change in heading between t-1 and t

Can we still use this motion model?

linear

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{\mathbf{t}-\mathbf{1}} + \mathbf{B}\mathbf{u}_t + \mathbf{v}_t$$

We want this to be Gaussian

odometry

nonlinear

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t) + \mathbf{v}_\mathbf{t},$$

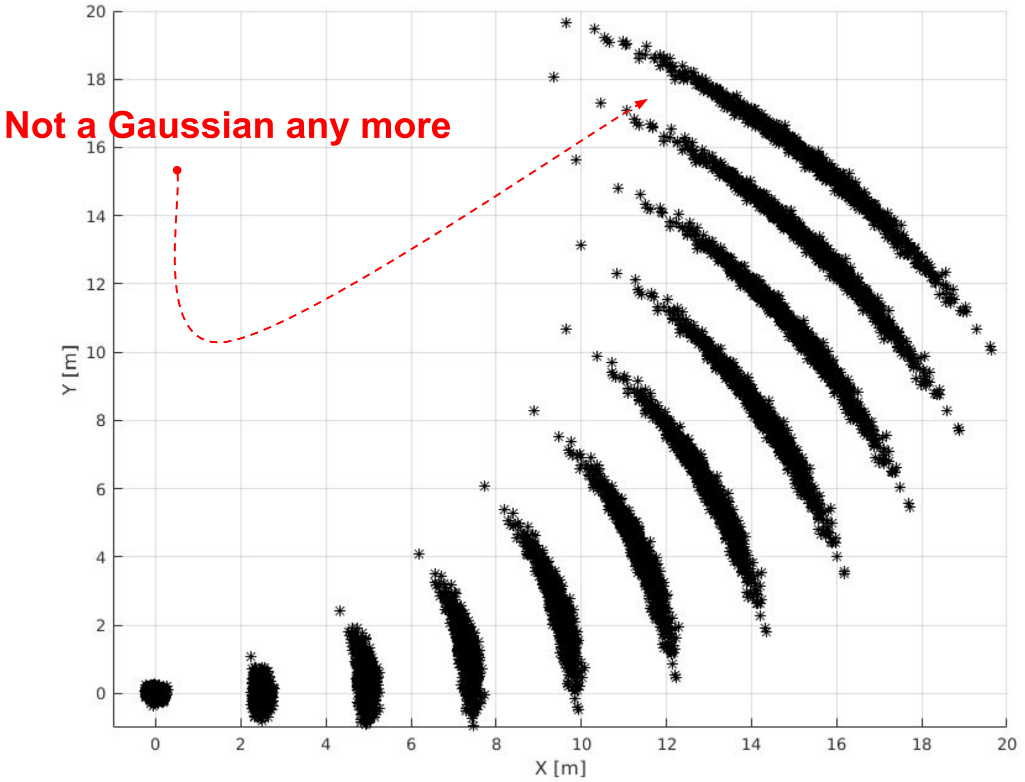
$$\mathcal{N}(\mu_{x_{t-1}}, \boldsymbol{\Sigma}_{t-1})$$

$$\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R})$$

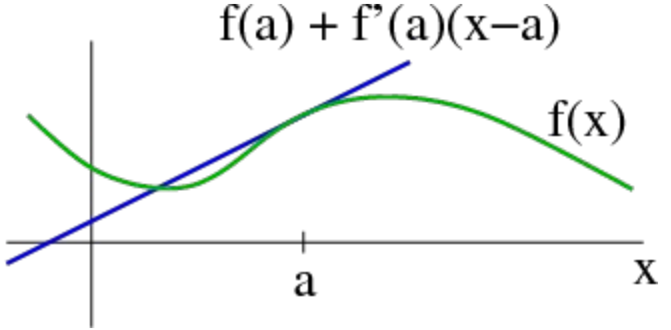
What happens to Gaussians when they pass through nonlinear functions?

```
clear all
N = 1000; % sample N points from a Gaussian
X = mvnrnd([0,0,0],[0.01 0 0;
                    0 0.01 0;
                    0 0 0.01] ,N);
figure(1)
clf
hold on
axis([-1 20 -1 20])
scatter(X(:,1),X(:,2),'k*')

Y = zeros(N,3);
for s = 1:10 % do 10 steps
    delta_d = 2.5; % move
    delta_theta = 10 * pi / 180; % turn
    for i = 1:N % pass all the points through f
        Y(i,1) = X(i,1) + delta_d * cos(X(i,3));
        Y(i,2) = X(i,2) + delta_d * sin(X(i,3));
        Y(i,3) = X(i,3) + delta_theta;
    end
scatter(Y(:,1),Y(:,2),'k*')
X = Y;
end
```



Linearization:
First Order Taylor Series Expansion



$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t) + \mathbf{v_t},$$

$$f(\mathbf{x}_{t-1}, \mathbf{u}_t) = f(\mu_{t-1}, \mathbf{u}_t) + \frac{\partial f(\mu_{t-1}, \mathbf{u}_t)}{\partial \mathbf{x}_{t-1}}(\mathbf{x}_{t-1} - \mu_{t-1})$$

$$f(\mathbf{x}_{t-1}, \mathbf{u}_t) = f(\mu_{t-1}, \mathbf{u}_t) + \mathbf{J}_{xt}(\mathbf{x}_{t-1} - \mu_{t-1})$$

Jacobian matrix calculated at each time step

The <u>extended</u> Kalman filter: **Prediction step**

$$\bar{\mu}_t = f(\mu_{t-1}, \mathbf{u}_t)$$
$$\bar{\Sigma}_t = \mathbf{J}_{xt}\mathbf{\Sigma}_{t-1}\mathbf{J}_{xt}^{T} + \mathbf{R}$$

What if the noise in the odometry is not simply additive

Jacobian matrix w.r.t pose

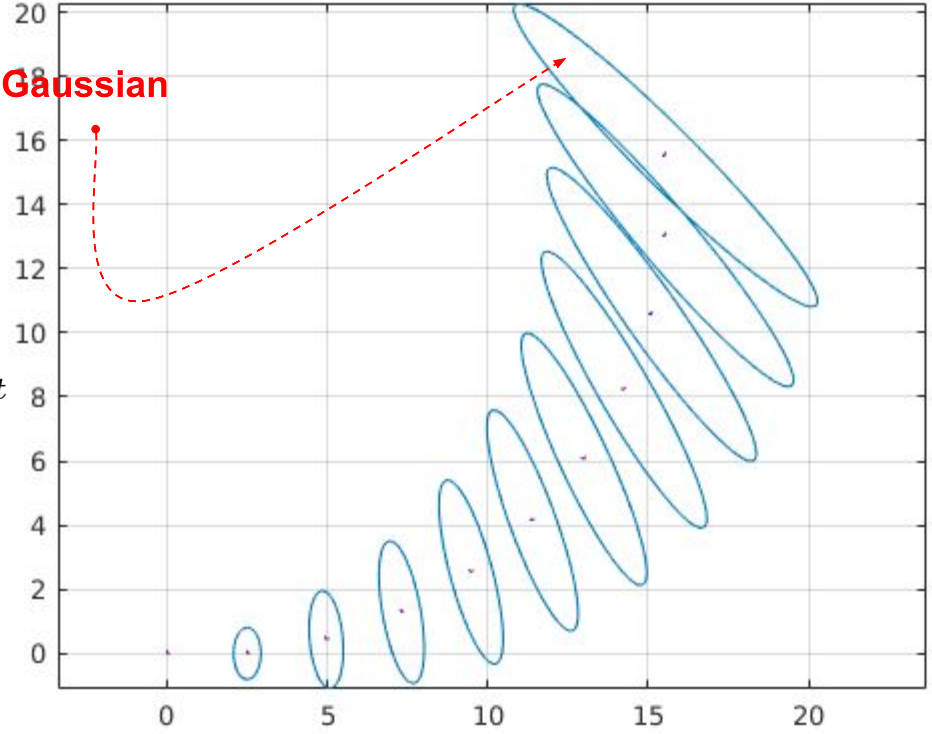
$$\bar{\Sigma}_t = \mathbf{J}_{x_t}\mathbf{\Sigma}_{t-1}\mathbf{J}_{x_t}^{T} + \mathbf{J}_{u_t}\mathbf{R}\mathbf{J}_{u_t}^{T}$$

Jacobian matrix w.r.t odometry

Linearization keeps it normal!

$$\bar{\mu}_t = f(\mu_{t-1}, \mathbf{u}_t)$$

$$\bar{\Sigma}_t = \mathbf{J}_{x_t} \mathbf{\Sigma}_{t-1} \mathbf{J}_{x_t}^T + \mathbf{J}_{u_t} \mathbf{R} \mathbf{J}_{u_t}^T$$



Exercise

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u})$$

$$\mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + a\sin(y_{t-1}) \\ b\sin(x_{t-1}) + y_{t-1}^2 \end{bmatrix} \begin{matrix} \Leftarrow f_1 \\ \Leftarrow f_2 \end{matrix}$$

$$\mathbf{J_x} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix} =? \qquad \mathbf{J_u} =?$$

$$\mathbf{u} = \begin{bmatrix} a \\ b \end{bmatrix}$$

Nonlinear measurement model

Most of the time the measurement model is not linear as well

linear $\Rightarrow$ $$\mathbf{z}_t = \mathbf{H}\mathbf{x_t} + \mathbf{w}_t$$

nonlinear $\Rightarrow$ $$\mathbf{z}_t = h(\mathbf{x}_t) + \mathbf{w}_t$$

$$\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q})$$

Linearization: First Order Taylor Series Expansion

$$\mathbf{z}_t = h(\mathbf{x}_t) + \mathbf{w}_t$$

$$h(\mathbf{x}_t) = h(\mu_t) + \frac{\partial h(\mu_t)}{\partial \mathbf{x}_t}(\mathbf{x}_t - \mu_t)$$

$$h(\mathbf{x}_t) = h(\mu_t) + \mathbf{G}_t(\mathbf{x}_t - \mu_t)$$

Jacobian matrix calculated at each time step

The extended Kalman filter: **Update step**

$$\mu_t = \bar{\mu}_t + \mathbf{K}_t(\mathbf{z}_t - h(\bar{\mu}_t))$$

$$\mathbf{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{G}_t)\bar{\mathbf{\Sigma}}_t$$

$$\mathbf{K}_t = \bar{\mathbf{\Sigma}}_t\mathbf{G}_t^T(\mathbf{G}_t\bar{\mathbf{\Sigma}}_t\mathbf{G}_t^T + \mathbf{Q})^{-1}$$

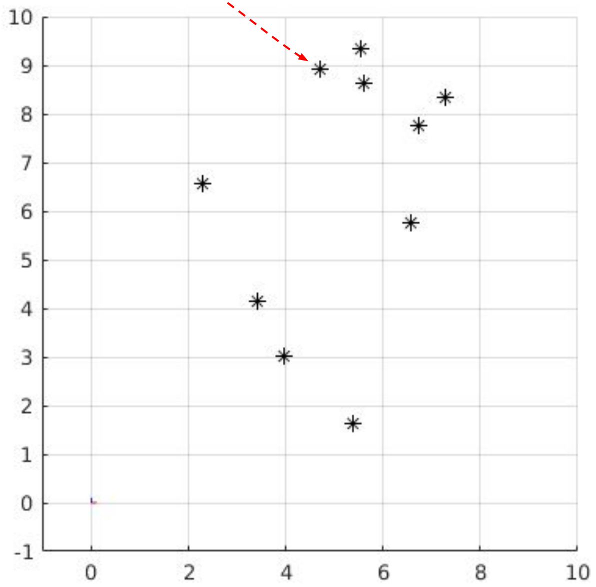
Landmark-based localization

- Given a set of landmarks with known positions (we will call this set our map).
- Given that we have a sensor onboard the robot that can detect these landmarks (we will assume the sensor can give us the range and bearing to each landmark in the map relative to the robot).
- Given that we have an idea about the motion of the robot through the odometry information.

We want to track the pose of the robot in the map while it is moving around.

Landmark-based map

$$\mathbf{M} = \begin{bmatrix} x_{l_1} \\ y_{l_1} \\ \vdots \\ x_{l_n} \\ y_{l_n} \end{bmatrix}$$



Range and bearing measurement model

$$h(\bar{\mu}_t, i) = \begin{bmatrix} r^i \\ \beta^i \end{bmatrix} = \begin{bmatrix} \sqrt{(x_r - x_{l_i})^2 + (y_r - y_{l_i})^2} \\ atan2(y_{l_i} - y_r, x_{l_i} - x_r) - \theta_r \end{bmatrix}$$

Nonlinear

Range and bearing with respect to robot's own frame of reference at time step t.

Coordinates of landmark i

Robot's pose

Jacobians Matrices

$$\mathbf{J}_x = \begin{bmatrix} 1 & 0 & -\delta d \times sin(\theta) \\ 0 & 1 & \delta d \times cos(\theta) \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{J}_u = \begin{bmatrix} cos(\theta) & 0 \\ sin(\theta) & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} -\frac{x_l - x_r}{r} & -\frac{y_l - y_r}{r} & 0 \\ \frac{y_l - y_r}{r^2} & -\frac{x_l - x_r}{r^2} & -1 \end{bmatrix}$$

Putting it all together

Prediction step:

$$\bar{\mu}_t = f(\mu_{t-1}, \mathbf{u}_t)$$

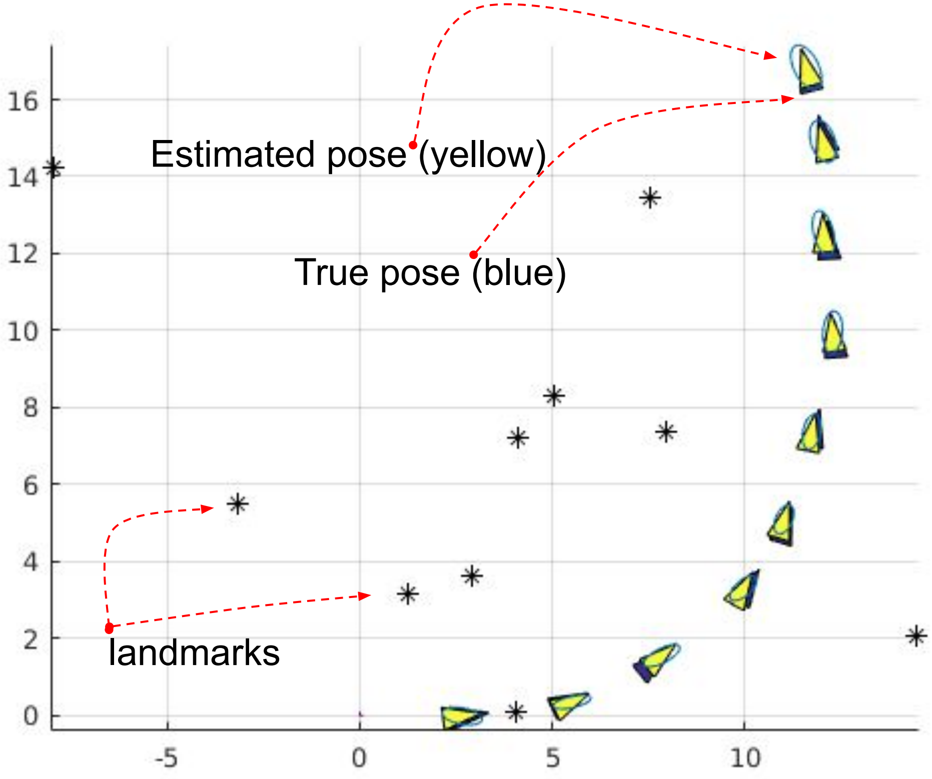
$$\bar{\Sigma}_t = \mathbf{J}_{x_t} \mathbf{\Sigma}_{t-1} \mathbf{J}_{x_t}^T + \mathbf{J}_{u_t} \mathbf{R} \mathbf{J}_{u_t}^T$$

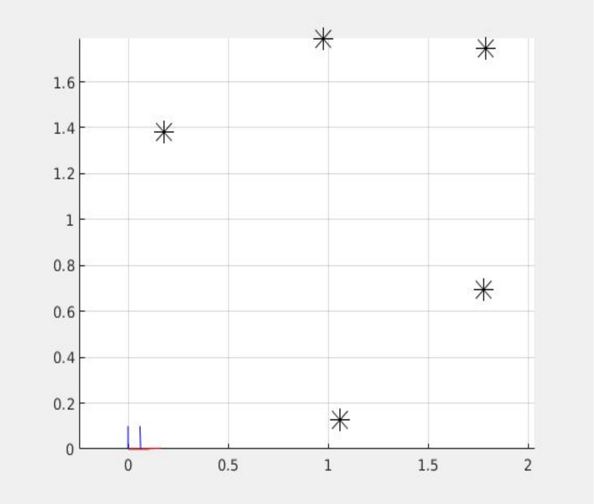
Update step:

For each landmark do:

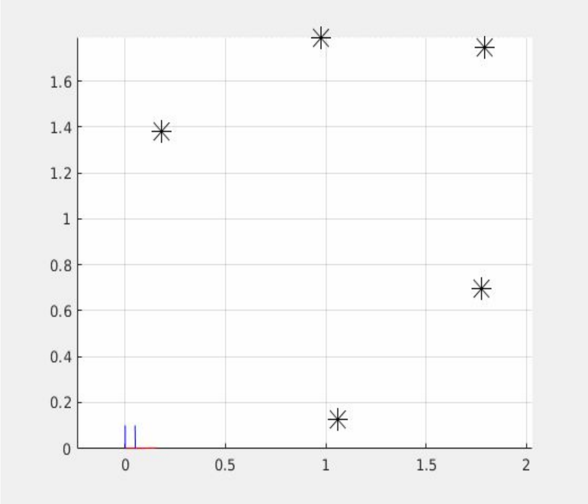
$$\mu_t = \bar{\mu}_t + \mathbf{K}_t(\mathbf{z}_t - h(\bar{\mu}_t))$$

$$\mathbf{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{G}_t)\bar{\mathbf{\Sigma}}_t$$

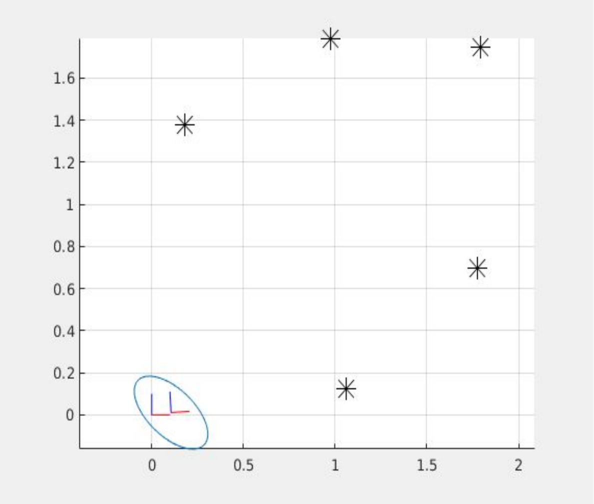




Ground truth (unknown)



Odometry



EKF localization

Known correspondences

Update step:

For each landmark $\mathbf{z}_t^i$ do:

$$\bar{\mu}_t = \bar{\mu}_t + \mathbf{K}_t^i(\mathbf{z}_t^i - h(\bar{\mu}_t, i))$$

$$\bar{\mathbf{\Sigma}}_t = (\mathbf{I} - \mathbf{K}_t^i \mathbf{G}_t^i)\bar{\mathbf{\Sigma}}_t$$

end

$$\mu_t = \bar{\mu}_t$$

$$\mathbf{\Sigma}_t = \bar{\mathbf{\Sigma}}_t$$

Next lecture MAPPING!

What if the pose of the robot is known and we want to estimate the positions of the landmarks?