# Autonomous Systems
# - Kalman filter for map-based localisation -

Dr Alexandru Stancu
Dr Mario Martinez

# Kalman filter localisation

- Motion model for the mobile robot:

$$\mathbf{s}_k = \mathbf{h}(\mathbf{s}_{k-1}, \mathbf{u}_k) + \mathbf{q}_k \qquad \mathbf{q}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$$

- Robot observation model:

$$\mathbf{z}_{i,k} = \mathbf{g}(\mathbf{m}_i, \mathbf{s}_k) + \mathbf{r}_{i,k} \qquad \mathbf{r}_{i,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$$

- If $\mathbf{h}$ and $\mathbf{g}$ are linear functions with respect to their variables, the Kalman filter can be used directly and optimality is guaranteed. Otherwise, Extended Kalman filter is used where both functions $\mathbf{h}$ and $\mathbf{g}$ are linearised around current robot pose estimate using first-order Taylor expansion.

# Kalman filter localisation

- Linearisation (theoretical aspects):

$$\hat{\boldsymbol{\mu}}_k = \mathbf{h}(\boldsymbol{\mu}_{k-1}, \mathbf{u}_k)$$

$$\mathbf{H}_k = \nabla_{\mathbf{s}_{k-1}} \mathbf{h}(\mathbf{s}_{k-1}, \mathbf{u}_k)\Big|_{\mathbf{s}_{k-1}=\boldsymbol{\mu}_{k-1}}$$

$$\mathbf{s}_k \approx \hat{\boldsymbol{\mu}}_k + \mathbf{H}_k(\mathbf{s}_{k-1} - \boldsymbol{\mu}_{k-1})$$

$$\hat{\mathbf{z}}_{i,k} = \mathbf{g}(\mathbf{m}_i, \hat{\boldsymbol{\mu}}_k)$$

$$\mathbf{G}_k = \nabla_{\mathbf{s}_k} \mathbf{g}(\mathbf{m}_i, \mathbf{s}_k)\Big|_{\mathbf{s}_k=\hat{\boldsymbol{\mu}}_k}$$

$$\mathbf{z}_{i,k} \approx \hat{\mathbf{z}}_{i,k} + \mathbf{G}_k(\mathbf{s}_k - \hat{\boldsymbol{\mu}}_k)$$

# Kalman filter localisation

- Based on Extended Kalman filter theory, the aim of the linearised model is to propagate covariance matrices based on Gaussian distributions.

- Thus, provided that $\mathbf{s}_{k-1} \sim \mathcal{N}(\boldsymbol{\mu}_{k-1}, \boldsymbol{\Sigma}_{k-1})$ is available, the ***prediction step*** of the extended Kalman filter is done in a similar manner to motion-based localisation.

$$\widehat{\boldsymbol{\mu}}_k = \mathbf{h}(\boldsymbol{\mu}_{k-1}, \mathbf{u}_k)$$

$$\widehat{\boldsymbol{\Sigma}}_k = \mathbf{H}_k \boldsymbol{\Sigma}_{k-1} \mathbf{H}_k^{\mathrm{T}} + \mathbf{Q}_k$$

# Kalman filter localisation

- The ***correction step*** of extended Kalman filter is carried out as follows:

$$\hat{\mathbf{z}}_{i,k} = \mathbf{g}(\mathbf{m}_i, \widehat{\boldsymbol{\mu}}_k)$$

$$\mathbf{Z}_k = \mathbf{G}_k \widehat{\boldsymbol{\Sigma}}_k \mathbf{G}_k^T + \mathbf{R}_k$$

$$\mathbf{K}_k = \widehat{\boldsymbol{\Sigma}}_k \mathbf{G}_k^T \mathbf{Z}_k^{-1}$$

$$\boldsymbol{\mu}_k = \widehat{\boldsymbol{\mu}}_k + \mathbf{K}_k (\mathbf{z}_{i,k} - \hat{\mathbf{z}}_{i,k})$$

$$\boldsymbol{\Sigma}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{G}_k) \widehat{\boldsymbol{\Sigma}}_k$$

# Kalman filter localisation

**Algorithm 1** EKF Localisation with known data association

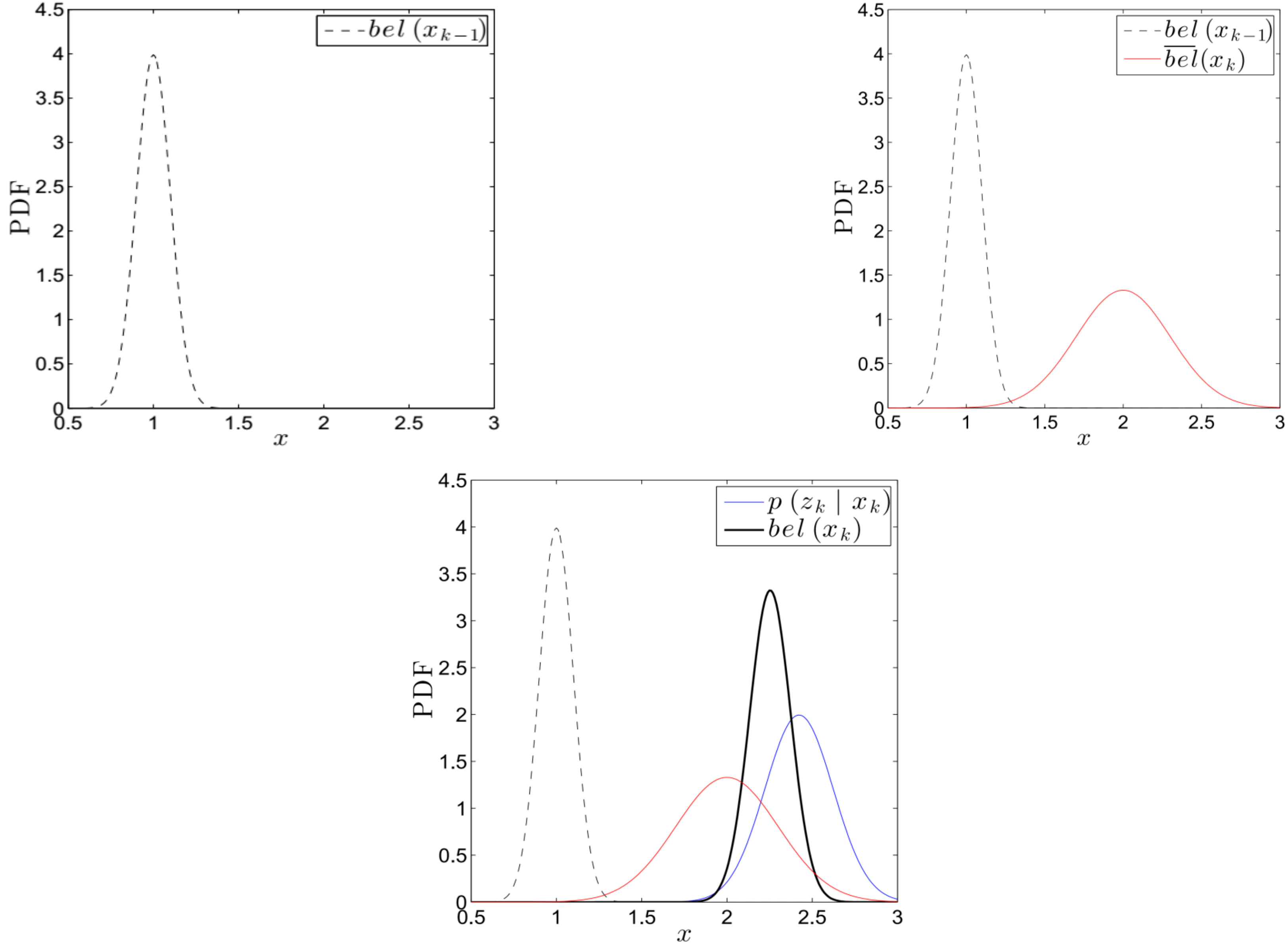
1. **function** EKF−Localisation ($M$, $\boldsymbol{\mu}_{k-1}$, $\boldsymbol{\Sigma}_{k-1}$, $\mathbf{u}_k, \mathbf{z}_{i,k}, \mathbf{Q}_k, \boldsymbol{R}_k$)
2. $\hat{\boldsymbol{\mu}}_k \leftarrow \mathbf{h}(\boldsymbol{\mu}_{k-1}, \mathbf{u}_k)$
3. $\mathbf{H}_k \leftarrow \nabla_{\mathbf{s}_{k-1}} \mathbf{h}(\mathbf{s}_{k-1}, \mathbf{u}_k)|_{\mathbf{s}_{k-1}=\boldsymbol{\mu}_{k-1}}$
4. $\hat{\boldsymbol{\Sigma}}_k \leftarrow \mathbf{H}_k \boldsymbol{\Sigma}_{k-1} \mathbf{H}_k^T + \mathbf{Q}_k$
5. **if** $\mathbf{z}_{i,k}$ **corresponds to landmark** $\mathbf{m}_i \in \boldsymbol{M}$
6. $\hat{\mathbf{z}}_{i,k} \leftarrow \mathbf{g}(\mathbf{m}_i, \hat{\boldsymbol{\mu}}_k)$
7. $\mathbf{G}_k \leftarrow \nabla_{\mathbf{s}_k} \mathbf{g}(\mathbf{m}_i, \mathbf{s}_k)|_{\mathbf{s}_k=\hat{\boldsymbol{\mu}}_k}$
8. $\mathbf{Z}_k \leftarrow \mathbf{G}_k \hat{\boldsymbol{\Sigma}}_k \mathbf{G}_k^T + \mathbf{R}_k$
9. $\mathbf{K}_k \leftarrow \hat{\boldsymbol{\Sigma}}_k \mathbf{G}_k^T \mathbf{Z}_k^{-1}$
10. $\boldsymbol{\mu}_k \leftarrow \hat{\boldsymbol{\mu}}_k + \mathbf{K}_k(\mathbf{z}_{i,k} - \hat{\mathbf{z}}_{i,k})$
11. $\boldsymbol{\Sigma}_k \leftarrow (\mathbf{I} - \mathbf{K}_k \mathbf{G}_k) \hat{\boldsymbol{\Sigma}}_k$
12. **return** $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$

# Map based localisation

| **Dead-reckoning localisation:** | **Map-based localisation:** |
| --- | --- |
| 1. Kinematic/dynamic model | 1. Kinematic/dynamic model |
| 2. Proprioceptive sensors: | 2. Proprioceptive sensors: |
| – Encoders | – Encoders |
| – Inertial measurement unit (IMU) | – Inertial measurement unit (IMU) |
| | 1. Exteroceptive sensors: |
| | – LiDAR |
| | – Camera |
| | 1. Map of the environment |
| (a) | (b) |

Requirement for: (a) dead-reckoning localisation, (b) map-based localisation.

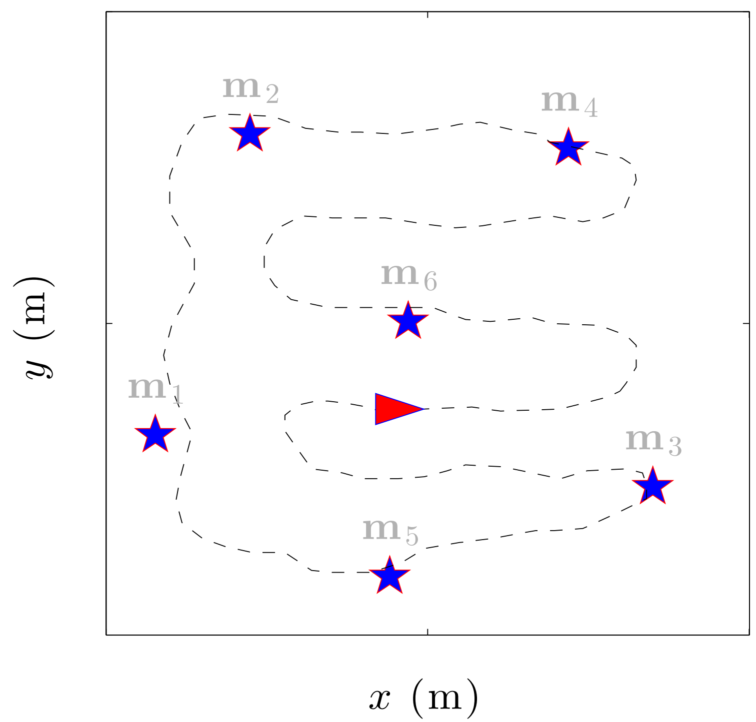
# Map based localisation



# Kalman filter localisation – Case study

- Consider a robot moving in 2D fully observable environment with 6 distinct landmarks. The robot pose represents position and orientation of the robot such that $\mathbf{s}_k = [s_{x,k} \quad s_{y,k} \quad s_{\theta,k}]^T$, and each landmark is a stationary point feature denoted by

$$\mathbf{m}_i = [m_{x,i} \quad m_{y,i}]^T$$



# Kalman filter localisation – Case study

- The robot has two control inputs $\mathbf{u}_k = [v_k \quad \omega_k]^T$, where $v_k$ is the robot linear velocity and $\omega_k$ is the robot angular velocity. The robot motion model is defined as follows:

$$s_{x,k} = s_{x,k-1} + \Delta t \cdot v_k \cdot \cos s_{\theta,k-1} + q_{x,k}$$

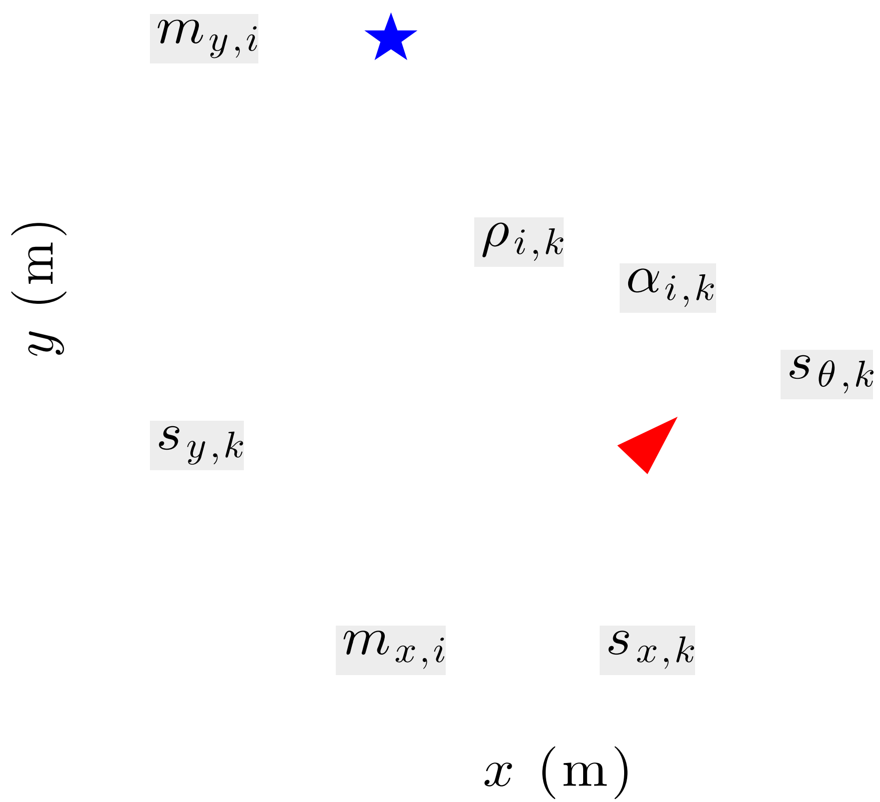
$$s_{y,k} = s_{y,k-1} + \Delta t \cdot v_k \cdot \sin s_{\theta,k-1} + q_{y,k}$$

$$s_{\theta,k} = s_{\theta,k-1} + \Delta t \cdot \omega_k + q_{\theta,k}$$

where $\Delta t$ is the temporal length of consecutive time steps, and the vector $\mathbf{q}_k = [q_{x,k} \quad q_{y,k} \quad q_{\theta,k}]^T$ represents the motion noise.

# Kalman filter localisation – Case study

- The robot is equipped with a 360° range-bearing exteroceptive sensor.



# Kalman filter localisation – Case study

The robot observation model is defined as follows:

$$z_{\rho,i,k} = \sqrt{(m_{x,i} - s_{x,k})^2 + (m_{y,i} - s_{y,k})^2} + r_{\rho,i,k}$$

$$z_{\alpha,i,k} = \text{atan2}(m_{y,i} - s_{y,k}, m_{x,i} - s_{x,k}) - s_{\theta,k} + r_{\alpha,i,k}$$

where $\mathbf{z}_{i,k} = [\rho_{i,k} \quad \alpha_{i,k}]^{\boldsymbol{T}}$ is the measurement vector that consists of landmark angle and bearing in robot frame $\{R\}$, and $\mathbf{r}_{i,k} = [r_{\rho,i,k} \quad r_{\alpha,i,k}]^{\boldsymbol{T}}$ is the measurement noise vector. The noise of both motion and observation models is Gaussian.

# Kalman filter localisation – Case study

- Since the robot motion and observation models are nonlinear, Extended Kalman Filter (EKF) is used where the Jacobian matrix $\mathbf{H}_k$ was defined in lecture 1 (equation 6.10 in the lecture notes).
- The observation model Jacobian matrix $\mathbf{G}_k$ is computed as follows:

$$\Delta x = m_{\mathrm{x},i} - \hat{s}_{x,k} \qquad \Delta y = m_{y,i} - \hat{s}_{y,k} \qquad p = \Delta x^2 + \Delta y^2$$

$$\mathbf{G}_{\boldsymbol{k}} = \begin{bmatrix} \frac{\partial g_1}{\partial s_{x,k}} & \frac{\partial g_1}{\partial s_{y,k}} & \frac{\partial g_1}{\partial s_{\theta,k}} \\ \frac{\partial g_2}{\partial s_{x,k}} & \frac{\partial g_2}{\partial s_{y,k}} & \frac{\partial g_2}{\partial s_{\theta,k}} \end{bmatrix} = \begin{bmatrix} -\frac{\Delta x}{\sqrt{p}} & -\frac{\Delta y}{\sqrt{p}} & 0 \\ \frac{\Delta y}{p} & -\frac{\Delta x}{p} & -1 \end{bmatrix}$$