Autonomous and Mobile Robotics

Prof. Giuseppe Oriolo

Localization

Landmark-Based and SLAM

Dipartimento di Ingegneria Informatica
Automatica e Gestionale Antonio Ruberti

Sapienza
Università di Roma

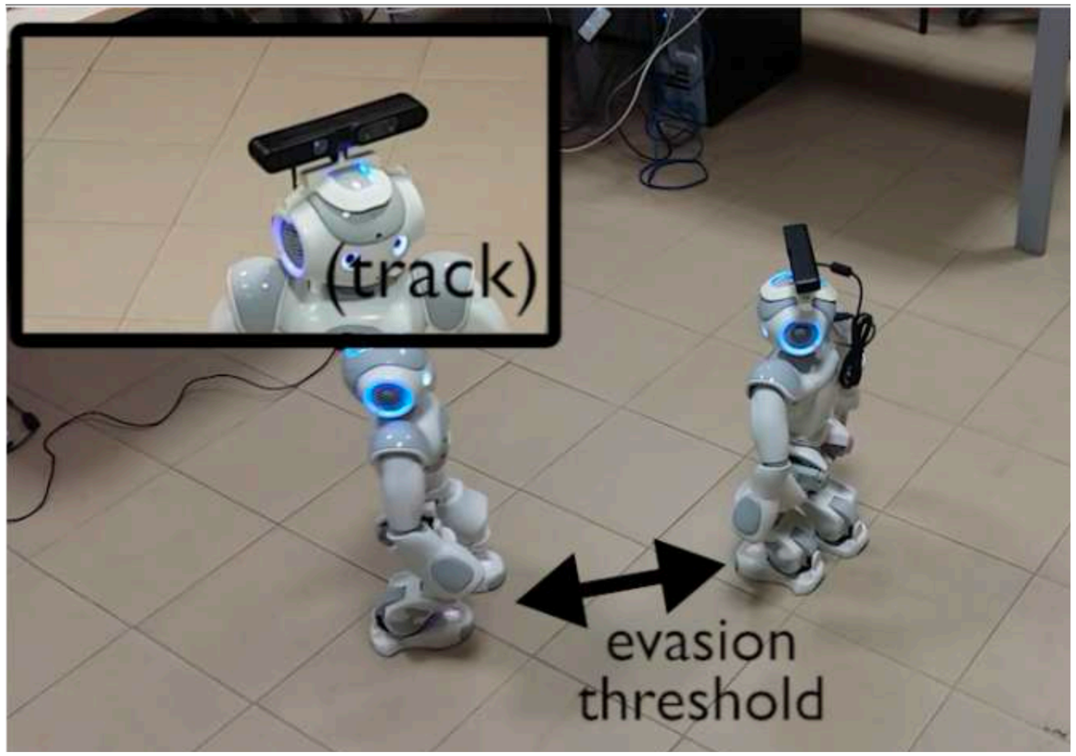
EKF localization with landmarks

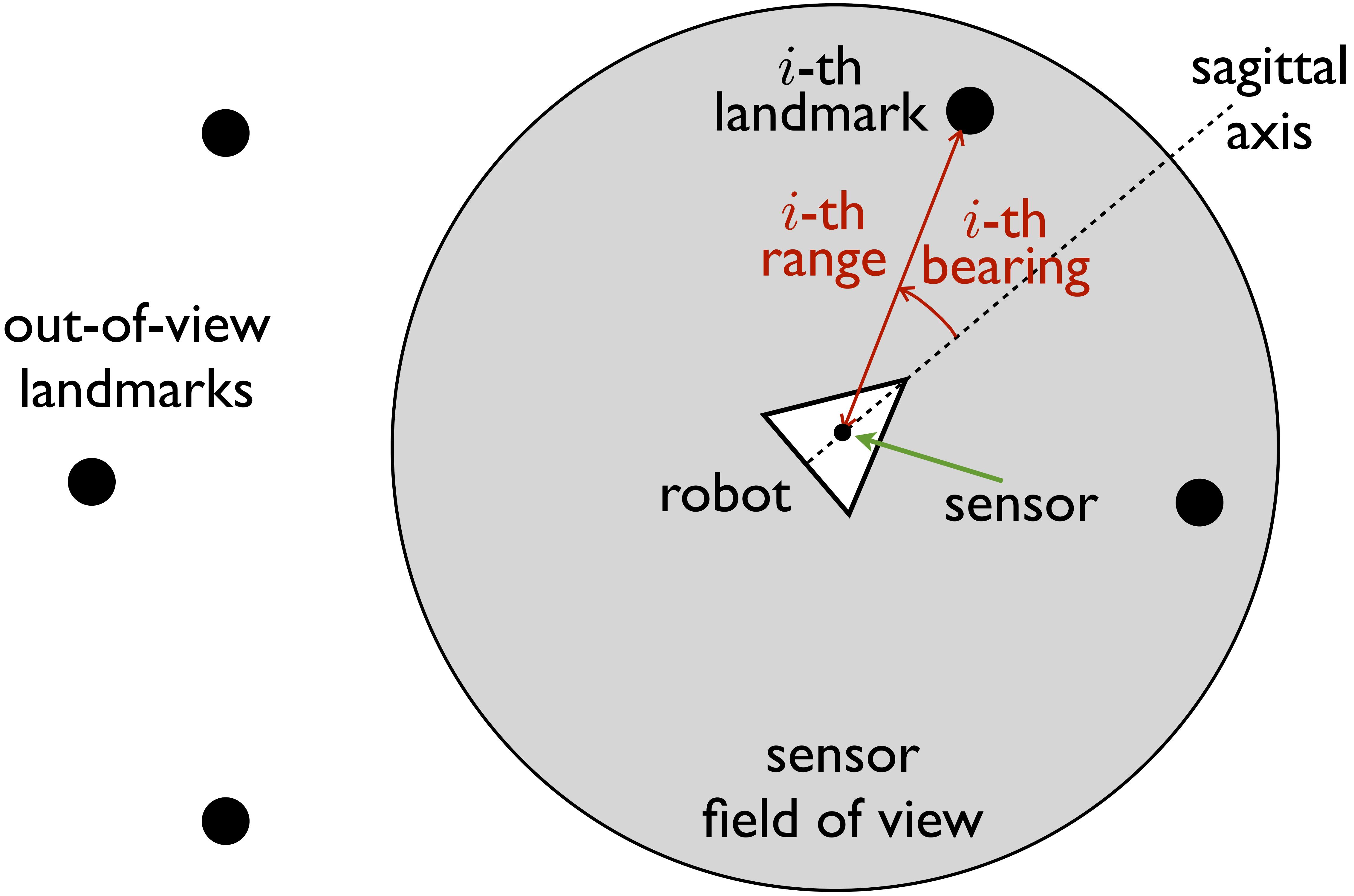
- assume that a unicycle-like robot is equipped with a sensor that measures range (relative distance) and bearing (relative orientation) to certain landmarks
- landmarks may be artificial or natural
- the position of the landmarks is fixed and known
- depending on the robot configuration, only a subset of the landmarks is actually visible
- suitable sensors are laser rangefinders (LIDARs), depth cameras or RFID sensors

Hokuyo laser rangefinder
on a Khepera

SICK laser rangefinder
on a Pioneer

ASUS Xtion depth camera
on a NAO




i-th
landmark
sagittal
axis
i-th
range
i-th
bearing
out-of-view
landmarks
robot
sensor
sensor
field of view

- odometric equations can be used as a discrete-time model of the robot; e.g., using Euler method

$$
\begin{aligned}
x_{k+1} &= x_k + v_k T_s \cos\theta_k + v_{1,k} \\
y_{k+1} &= y_k + v_k T_s \sin\theta_k + v_{2,k} \\
\theta_{k+1} &= \theta_k + \omega_k T_s + v_{3,k}
\end{aligned}
$$

where $\boldsymbol{v}_k = (v_{1,k}\ v_{2,k}\ v_{3,k})^T$ is a white gaussian noise with zero mean and covariance matrix $\boldsymbol{V}_k$

- assume that L landmarks are present, and denote by $(x_{l,i}, y_{l,i})$ the position of the i-th landmark

- let $L_k \leq L$ be the number of landmarks that the robot can actually see at step k

- each of the L_k measurements actually contains two components, i.e., a range component and a bearing component

- assume that for each measurement the identity of observed landmark is known (landmarks are tagged, e.g., by shape, color or radio frequency)

- we build the association map of step k

$$a : \underbrace{\{1, 2, \ldots, L_k\}}_{\text{measurements}} \mapsto \underbrace{\{1, 2, \ldots, L\}}_{\text{landmarks}}$$

hence, $a(i)$ is the index of the landmark observed by the i-th measurement

- the output equation is

$$\boldsymbol{y}_k = \begin{pmatrix} \boldsymbol{h}_1(\boldsymbol{q}_k, a(1)) \\ \vdots \\ \boldsymbol{h}_{L_k}(\boldsymbol{q}_k, a(L_k)) \end{pmatrix} + \begin{pmatrix} \boldsymbol{w}_{1,k} \\ \vdots \\ \boldsymbol{w}_{L_k,k} \end{pmatrix}$$

where

i-th landmark range

$$\boldsymbol{h}_i(\boldsymbol{q}_k, a(i)) = \begin{pmatrix} \overset{\downarrow}{\sqrt{(x_k - x_{l,a(i)})^2 + (y_k - y_{l,a(i)})^2}} \\ \underset{\uparrow}{\text{atan2}(y_{l,a(i)} - y_k, x_{l,a(i)} - x_k) - \theta_k} \end{pmatrix}$$

i-th landmark bearing

and $\boldsymbol{w}_k = (w_{1,k} \, \ldots \, w_{L_k,k})^T$ is a white gaussian noise with zero mean and covariance matrix $\boldsymbol{W}_k$

- we want to maintain an accurate estimate of the robot configuration in the presence of process and measurement noise: this is the ideal setting for KF

- actually, since both process and output equations are nonlinear, we must apply the EKF and, to this end, the equations must be linearized

- process dynamics linearization

$$\boldsymbol{F}_k = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}_k}\right|_{\boldsymbol{q}_k=\hat{\boldsymbol{q}}_k} = \begin{pmatrix} 1 & 0 & -v_k T_s \sin\hat{\theta}_k \\ 0 & 1 & v_k T_s \cos\hat{\theta}_k \\ 0 & 0 & 1 \end{pmatrix}$$

- output equation linearization

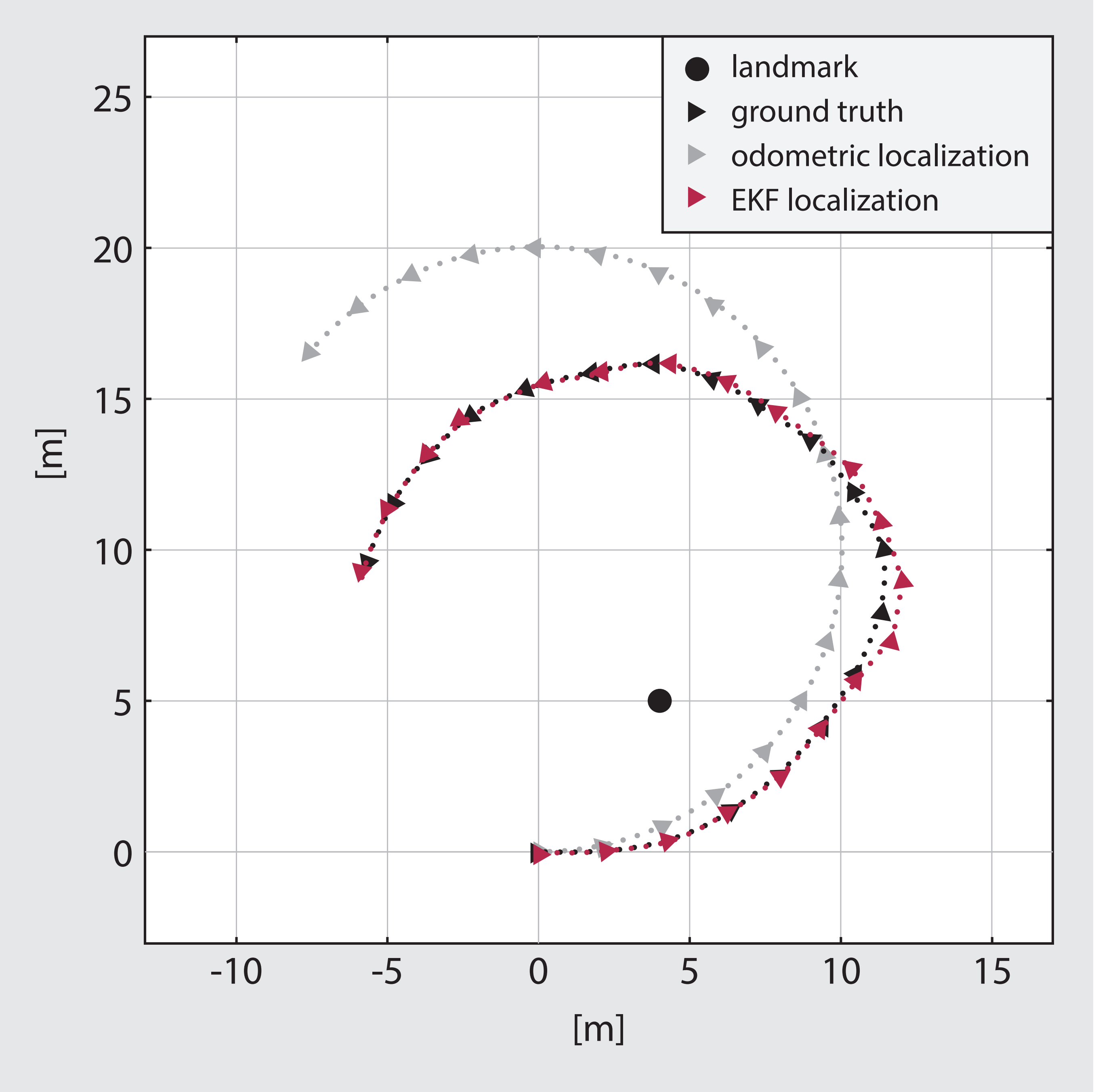
$$\boldsymbol{H}_{k+1} = \begin{pmatrix} \left.\dfrac{\partial \boldsymbol{h}_1}{\partial \boldsymbol{q}_k}\right|_{\boldsymbol{q}_k = \hat{\boldsymbol{q}}_{k+1|k}} \\ \vdots \\ \left.\dfrac{\partial \boldsymbol{h}_{L_k}}{\partial \boldsymbol{q}_k}\right|_{\boldsymbol{q}_k = \hat{\boldsymbol{q}}_{k+1|k}} \end{pmatrix}$$

where

$$\left.\frac{\partial \boldsymbol{h}_i}{\partial \boldsymbol{q}_k}\right|_{\boldsymbol{q}_k = \hat{\boldsymbol{q}}_{k+1|k}} = \begin{pmatrix} \frac{\hat{x}_{k+1|k} - x_{l,a(i)}}{\sqrt{(\hat{x}_{k+1|k} - x_{l,a(i)})^2 + (\hat{y}_{k+1|k} - y_{l,a(i)})^2}} & \frac{\hat{y}_{k+1|k} - y_{l,a(i)}}{\sqrt{(\hat{x}_{k+1|k} - x_{l,a(i)})^2 + (\hat{y}_{k+1|k} - y_{l,a(i)})^2}} & 0 \\ \frac{-(\hat{y}_{k+1|k} - y_{l,a(i)})}{(x_{k+1|k} - x_{l,a(i)})^2 + (\hat{y}_{k+1|k} - y_{l,a(i)})^2} & \frac{\hat{x}_{k+1|k} - x_{l,a(i)}}{(x_{k+1|k} - x_{l,a(i)})^2 + (\hat{y}_{k+1|k} - y_{l,a(i)})^2} & -1 \end{pmatrix}$$

- at this point, just crank the EKF engine

a typical result



- nominal trajectory is a circle, but due to motion+measurement noise the actual trajectory (ground truth) is different

data association

- remove the hypothesis that the identity of each observed landmark is known: in practice, landmarks can be undistinguishable by the sensor

- the association map must be estimated as well

- basic idea: associate each observation to the landmark that minimizes the magnitude of the innovation

- at the $k+1$-th step,consider the i-th measurement $\boldsymbol{y}_{i,k+1}$ and compute all the candidate innovations

$$\boldsymbol{\nu}_{ij} = \underbrace{\boldsymbol{y}_{i,k+1}}_{\text{actual measurement}} - \underbrace{\boldsymbol{h}_i(\hat{q}_{k+1|k}, j)}_{\text{expected measurement if } \boldsymbol{y}_{i,k+1} \text{ referred to the } j\text{-th landmark}}$$

- the smaller the innovation $\boldsymbol{\nu}_{ij}$, the more likely that the i-th measurement corresponds to the j-th landmark
- however, the innovation magnitude must be weighted with the uncertainty of measurement; in the EKF, this is encoded in the matrix

$$\boldsymbol{S}_{ij} = \boldsymbol{H}_i(k+1,j)\boldsymbol{P}_{k+1|k}\boldsymbol{H}_i(k+1,j)^T + \boldsymbol{W}_{i,k+1}$$

measurement uncertainty
due to prediction uncertainty

measurement uncertainty
due to sensor noise

- to determine the association function, let

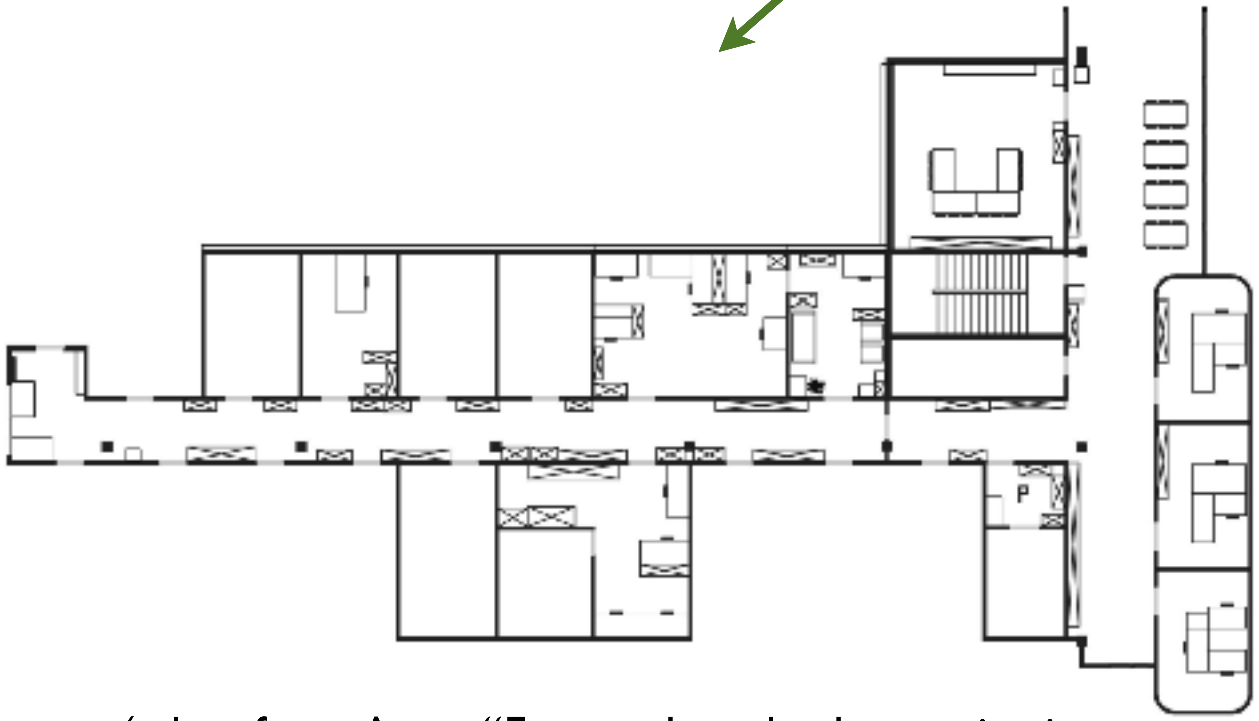
$$\chi_{ij} = \boldsymbol{\nu}_{ij}^T \boldsymbol{S}_{ij}^{-1} \boldsymbol{\nu}_{ij}$$

and let $a(i) = j$, where j minimizes χ_{ij}

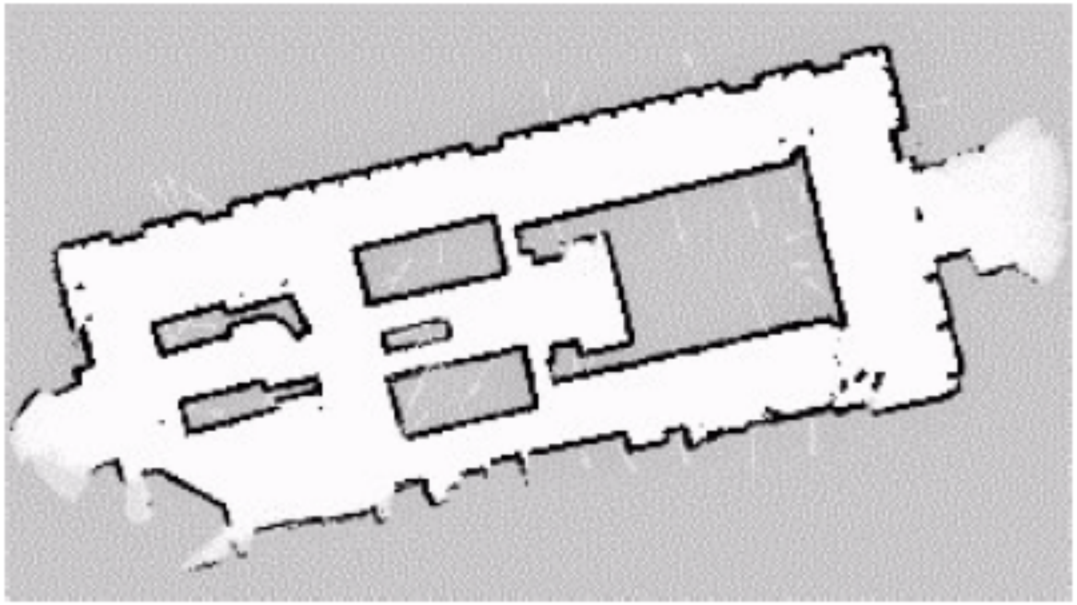
EKF localization on a map

- assume that a metric map $\mathcal{M}$ of the environment is known to the robot

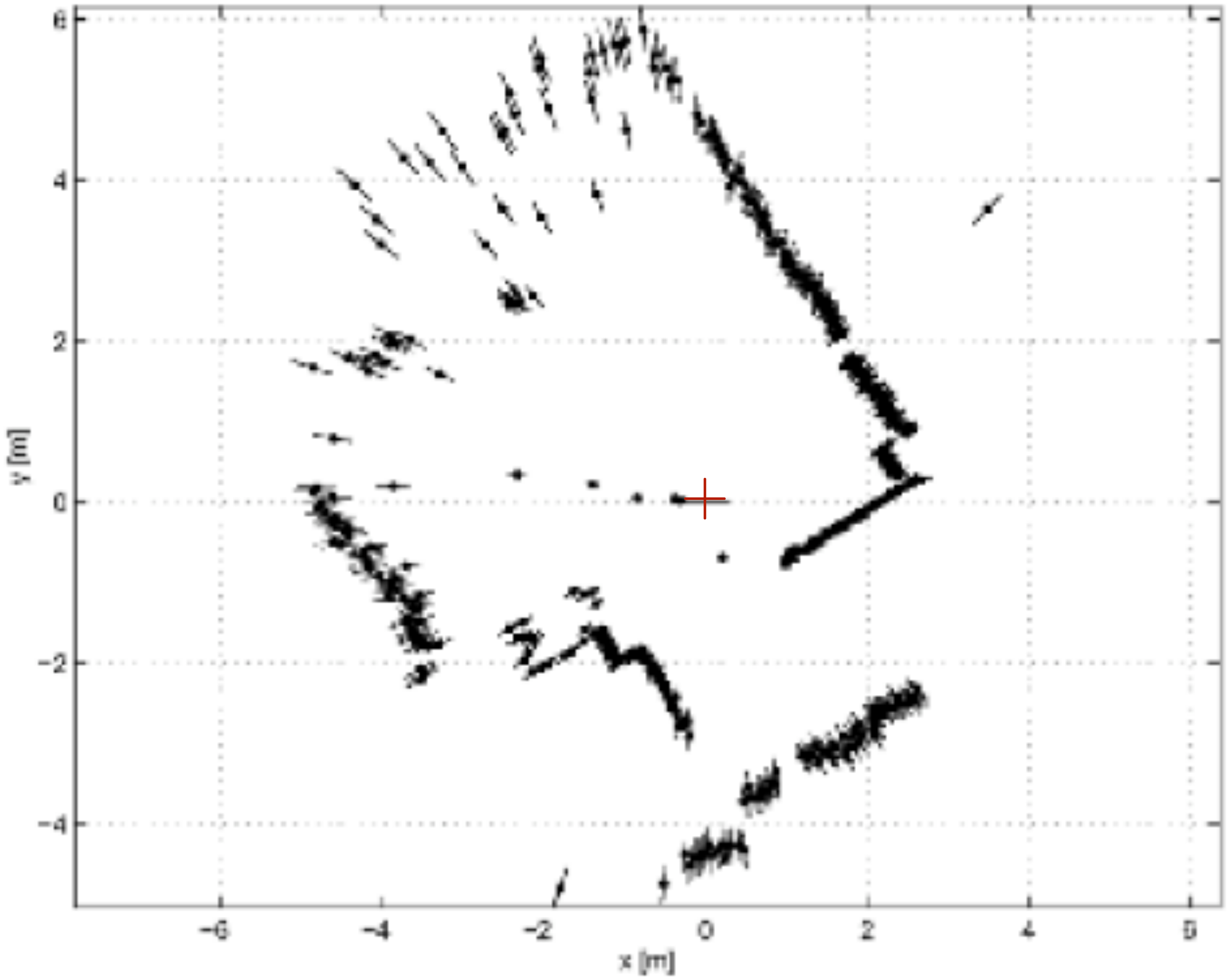
- this may be a line-based map or an occupancy grid



(taken from: Arras, "Feature-based robot navigation in known and unknown environments", 2003)



- assume that the robot is equipped with a range finder; e.g., a laser sensor, whose typical scan looks like this (note the uncertainty intervals)



(taken from: Arras, "Feature-based robot navigation in known and unknown environments", 2003)

- use the whole scan as output vector: its components are the range readings in all available directions

- the innovation is then computed as the difference between the actual scan and the predicted scan

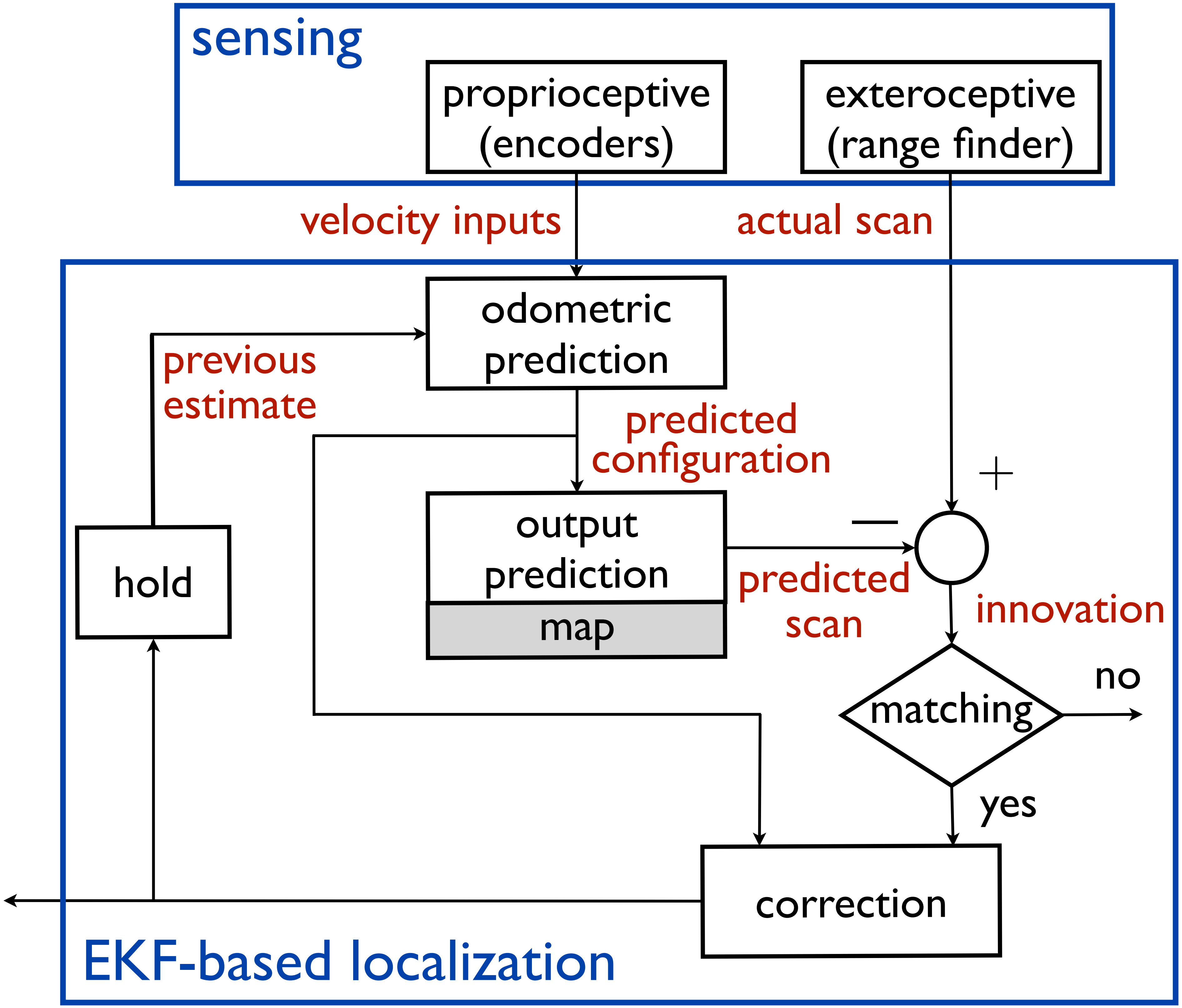
$$\boldsymbol{\nu}_{k+1} = \boldsymbol{y}_{k+1} - \boldsymbol{h}(\hat{\boldsymbol{q}}_{k+1|k}, \mathcal{M})$$

where $\boldsymbol{h}(\ )$ computes the predicted scan by placing the robot at a configuration in the map

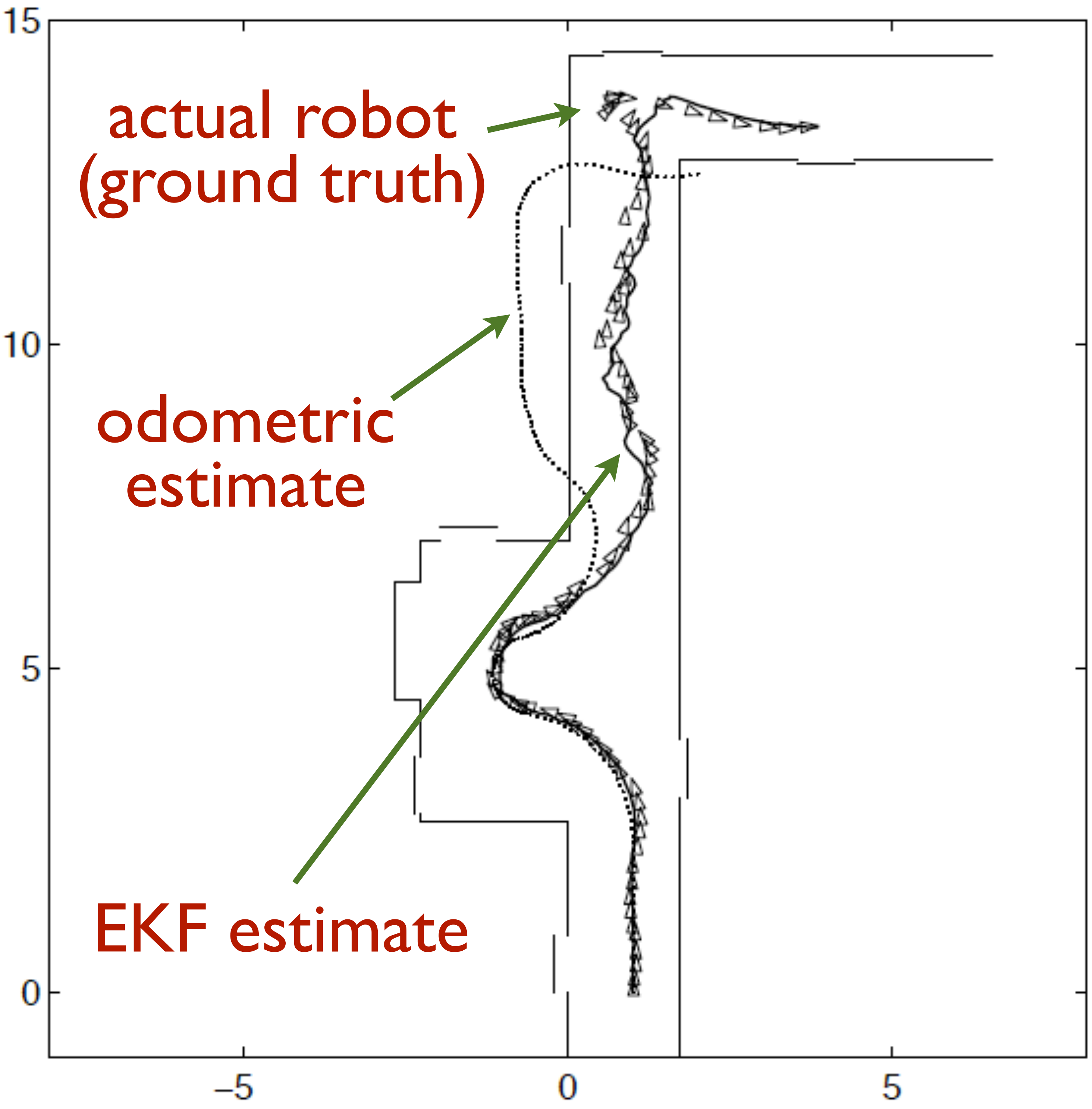
- note that no data association is needed; on the other hand, aliasing may severely displace the estimate

- both the process dynamics (i.e., the robot kinematic model) and the output function $\boldsymbol{h}$ are nonlinear, and therefore the EKF must be used

architecture



a typical result



actual robot (ground truth)

odometric estimate

EKF estimate

- robotized wheelchair with high slippage
- 5 ultrasonic sensors with 2 Hz rate
- shadow zone behind the robot

EKF SLAM

- remove the hypothesis that the environment is known a priori: as it moves, the robot must use its sensors to build a map and at the same time localize itself

- SLAM: Simultaneous Localization And Map-building

- in probabilistic SLAM, the idea is to estimate the map features in addition to the robot configuration

- here, we discuss a simple landmark-based version of the problem which can be solved using KF or EKF

- assumptions:
  - the robot is an omnidirectional point-robot, whose configuration is then a cartesian position
  - L landmarks are distributed in the environment (their position is unknown)
  - the robot is equipped with a sensor that can see, identify and measure the relative position of all landmarks wrt itself (infinite FOV + no occlusions)
- define an extended state vector to be estimated

$$\boldsymbol{x} = \begin{pmatrix} \underset{\text{robot position}}{x \quad y} & \underset{\text{landmark 1 position}}{x_{l1} \quad y_{l1}} & \underset{\dots}{\dots} & \underset{\text{landmark } L \text{ position}}{x_{lL} \quad y_{lL}} \end{pmatrix}^T$$

- since the landmarks are fixed, the discrete-time model of the robot+landmarks system is ($T_s = 1$ w.l.o.g.)

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u_{x,k} \\ u_{y,k} \end{pmatrix} + \begin{pmatrix} v_{x,k} \\ v_{y,k} \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

where $u_{x,k}$ and $u_{y,k}$ are the robot velocity inputs while $\boldsymbol{v}_{xy,k} = (v_{x,k}\ v_{y,k})^T$ is a white gaussian noise with zero mean and covariance matrix $\boldsymbol{V}_{xy,k}$

- this is clearly a linear model of the form

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{v}_k$$

where $\boldsymbol{u}_k = (u_{x,k}\ u_{y,k})^T$ is the input vector while $\boldsymbol{v}_k = (v_{1,k}\ v_{2,k}\ \ldots\ v_{2+2L,k})^T$ is a white gaussian noise with zero mean and covariance matrix

$$\boldsymbol{V}_k = \begin{pmatrix} \boldsymbol{V}_{xy,k} & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \end{pmatrix}$$

- the i-th measurement contains the relative position of the i-th landmark wrt the sensor

$$\boldsymbol{y}_{i,k} = \begin{pmatrix} x_{li,k} - x_k \\ y_{li,k} - y_k \end{pmatrix} + \boldsymbol{w}_{i,k}$$

where $\boldsymbol{w}_{i,k}$ is a white gaussian noise with zero mean and covariance matrix $\boldsymbol{W}_{i,k}$

- it is a linear equation

$$\boldsymbol{y}_{i,k} = \boldsymbol{C}_i \boldsymbol{x}_k + \boldsymbol{w}_{i,k}$$

with

$$\boldsymbol{C}_i = \begin{pmatrix} -1 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 0 & \dots & 0 & \underset{\uparrow}{0} & 1 & 0 & \dots & 0 \end{pmatrix}$$

$(2i+1)$-th column

- stack all measurements to create the output vector

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{w}_k$$

where

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{C}_1 \\ \vdots \\ \boldsymbol{C}_L \end{pmatrix} \qquad \boldsymbol{w}_k = \begin{pmatrix} \boldsymbol{w}_{1,k} \\ \vdots \\ \boldsymbol{w}_{L,k} \end{pmatrix}$$

and the covariance of the measurement noise is

$$\boldsymbol{W}_k = \begin{pmatrix} \boldsymbol{W}_{1,k} & 0 & \dots & 0 \\ 0 & \boldsymbol{W}_{2,k} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \boldsymbol{W}_{L,k} \end{pmatrix}$$

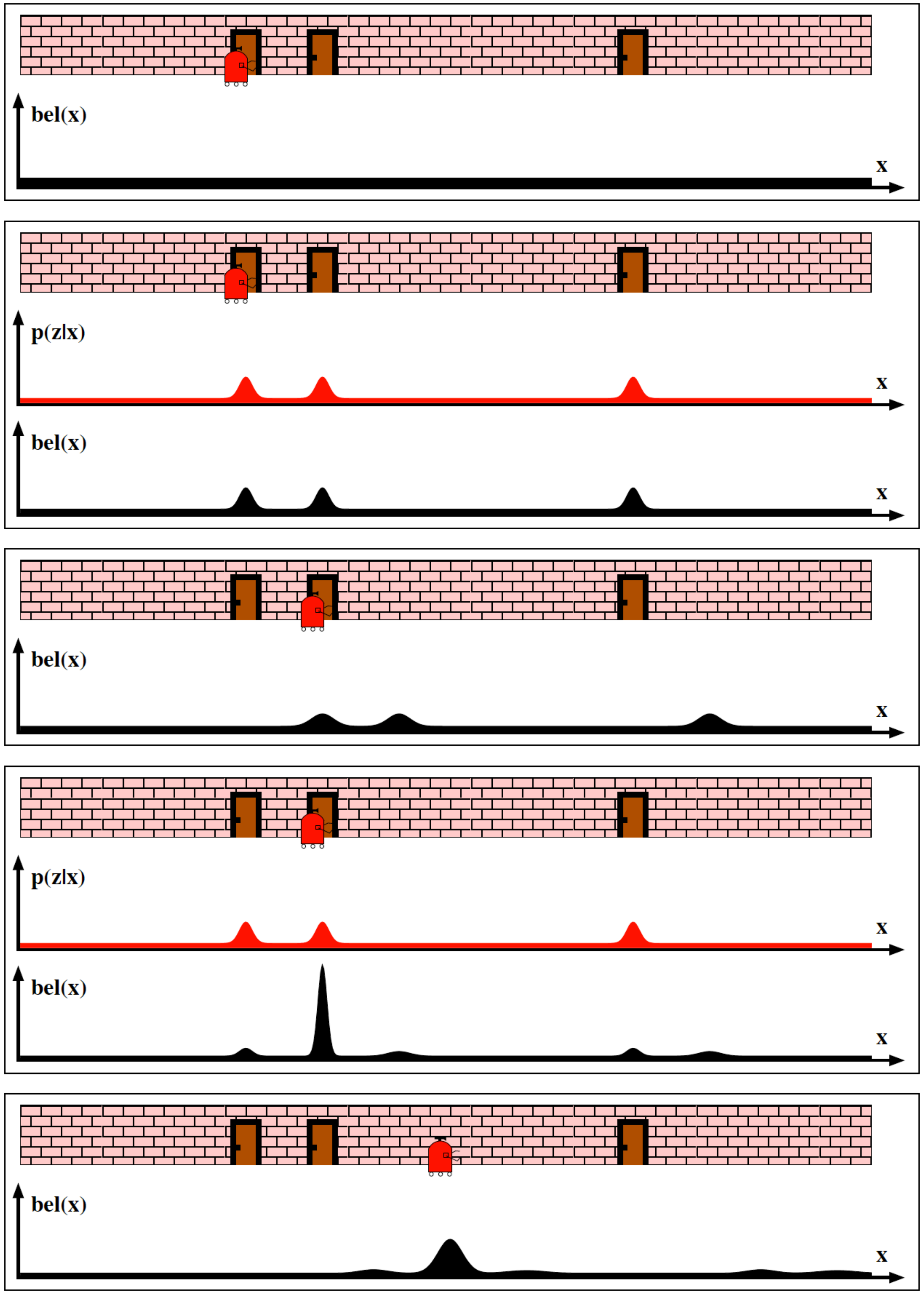
- at this point, just crank the KF engine

how realistic is KF/EKF localization?

- KF/EKF assume that the probability distribution for the state is unimodal, and in particular a gaussian

- this requires an accurate estimate of the robot initial configuration and also relatively small uncertainties (position tracking problem)

- however, if the robot is released at an unknown (or poorly known) position, the probability distribution for the state becomes multimodal in the presence of aliasing (kidnapped robot problem)



- need to track multiple hypotheses

- more general Bayesian estimators (e.g., particle filters) must be used