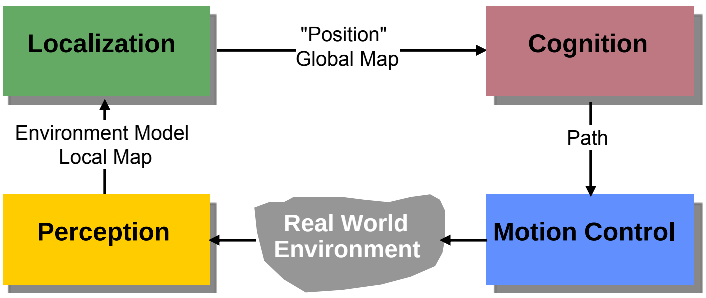


Probabilistic Map Based Localization:

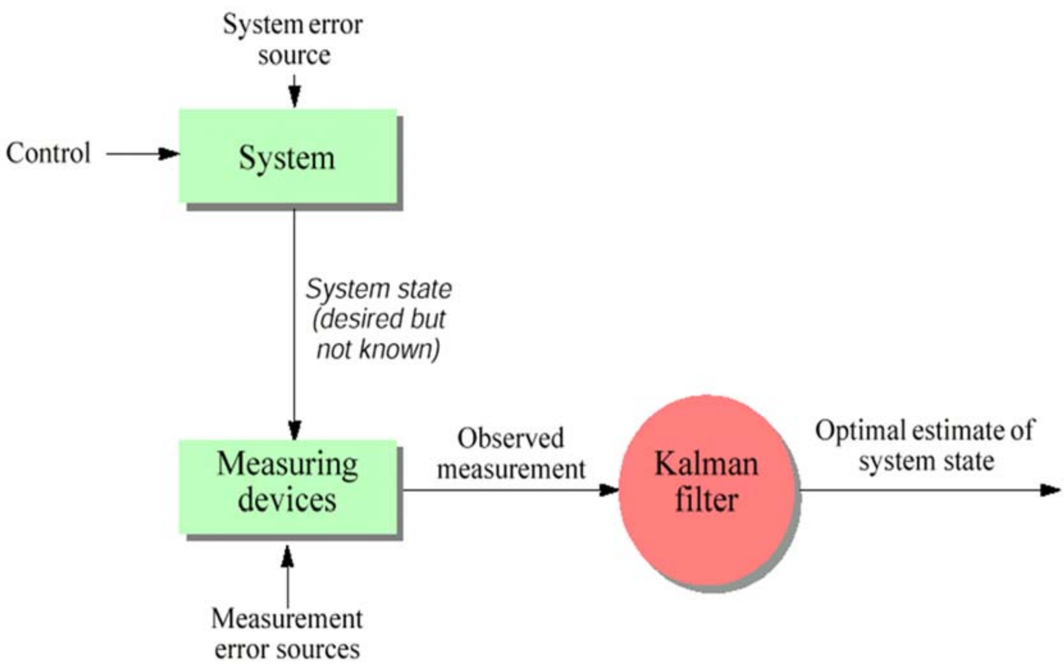
Kalman Filter Localization

Section 5.6.8, pages 322-342 in the textbook

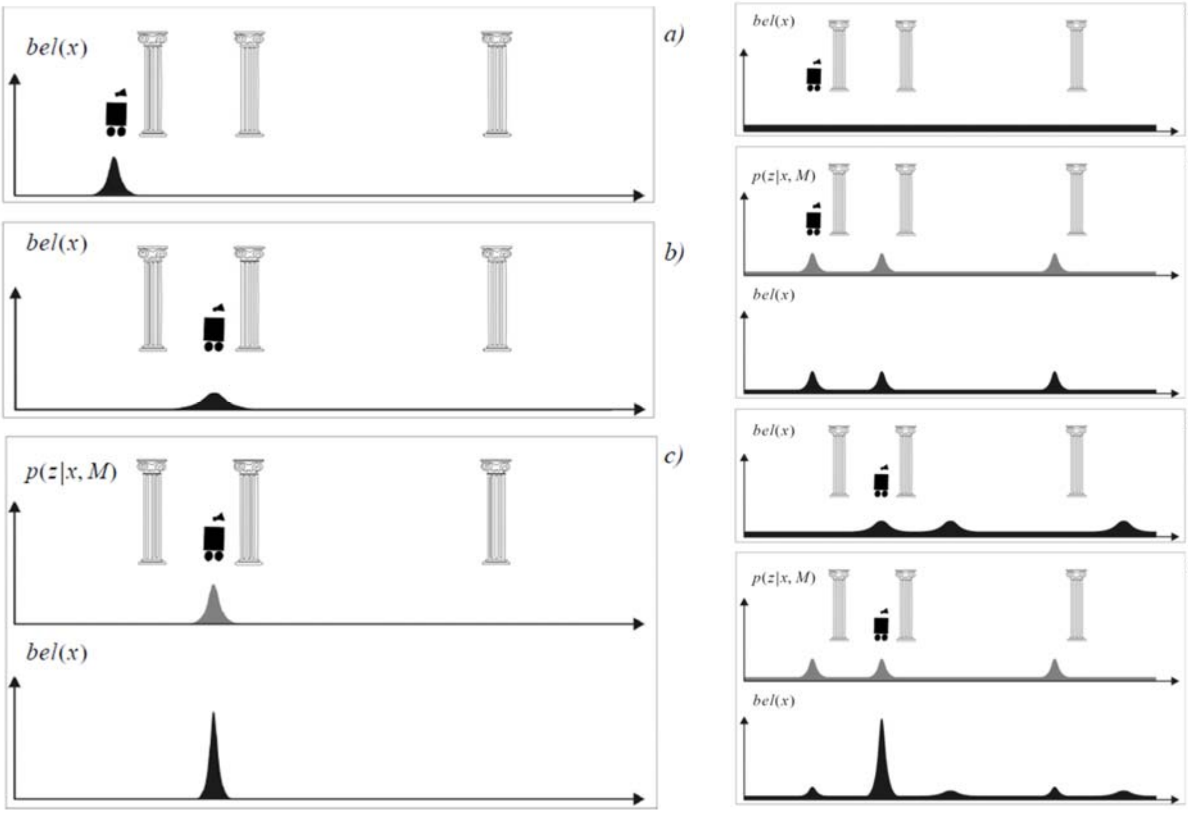
ETH Zürich

Autonomous Systems Lab

2 Kalman Filter Localization

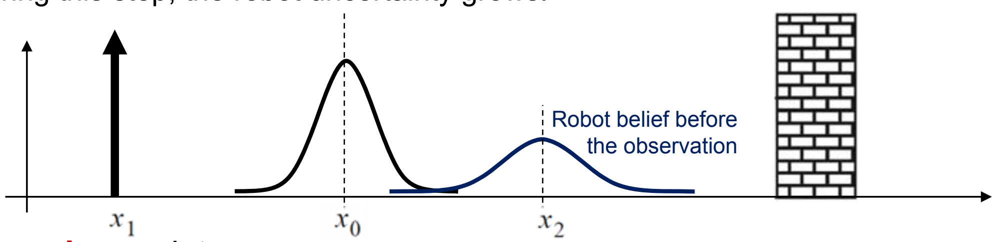


Kalman filter localization

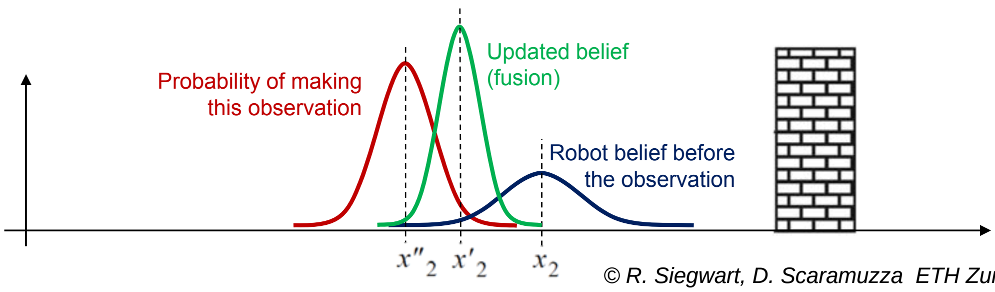


Action and perception updates

- In robot localization, we distinguish two update steps:
  1. **Action** update:
     - the robot moves and estimates its position through its proprioceptive sensors. During this step, the robot uncertainty grows.



  2. **Perception** update:
     - the robot makes an observation using its exteroceptive sensors and correct its position by opportunely combining its belief before the observation with the probability of making exactly that observation. During this step, the robot uncertainty shrinks.



6 Markov versus Kalman localization

Two approaches exist to represent the probability distribution and to compute the convolution and Bayes rule during the Action and Perception phases

Markov	Kalman
• The configuration space is divided into many cells. The configuration space of a robot moving on a plane is 3D dimensional (x,y,θ). Each cell contains the probability of the robot to be in that cell.	• The probability distribution of both the robot configuration and the sensor model is assumed to be continuous and **Gaussian!**
• The probability distribution of the sensors model is also discrete.	
• During Action and Perception, all the cells are updated. Therefore the computational cost is very high.	• Since a Gaussian distribution only described through mean value μ and variance σ^2, we need only to update μ and $\Sigma^{2.}$ Therefore the computational cost is very low!
• Localization can start from any unknown position and recovers from ambiguous situation.	• Localization is tracked from a known positions and recovery from ambiguous situations and after collision is not possible.

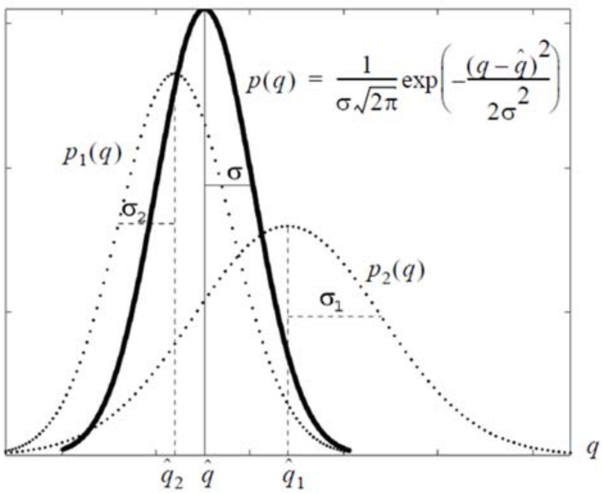
7 Introduction to Kalman Filter (1)

- Two measurements

$$\hat{q}_1 = q_1 \text{ with variance } \sigma_1^2$$
$$\hat{q}_2 = q_2 \text{ with variance } \sigma_2^2$$

- Weighted least-square

$$S = \sum_{i=1}^{n} w_i(\hat{q} - q_i)^2$$



- Finding minimum error

$$\frac{\partial S}{\partial \hat{q}} = \frac{\partial}{\partial \hat{q}} \sum_{i=1}^{n} w_i(\hat{q} - q_i)^2 = 2\sum_{i=1}^{n} w_i(\hat{q} - q_i) = 0$$

- After some calculation and rearrangements

$$\hat{q} = q_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}(q_2 - q_1)$$

Introduction to Kalman Filter (2)

- In Kalman Filter notation

$$\hat{x}_{k+1} = \hat{x}_k + K_{k+1}(z_{k+1} - \hat{x}_k)$$

$$K_{k+1} = \frac{\sigma_k^2}{\sigma_k^2 + \sigma_z^2} \; ; \; \sigma_k^2 = \sigma_1^2 \; ; \; \sigma_z^2 = \sigma_2^2$$

$$\sigma_{k+1}^2 = \sigma_k^2 - K_{k+1}\sigma_k^2$$

Introduction to Kalman Filter (3)

1st hour, 28.4.2008

- Dynamic Prediction (robot moving)

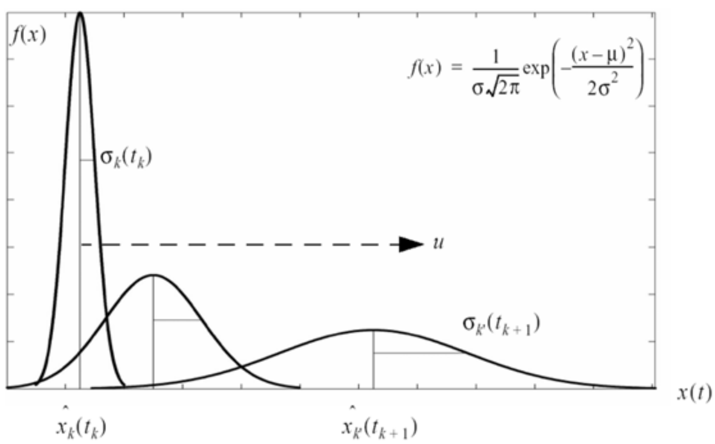
$$\frac{dx}{dt} = u + w$$

u = velocity
w = noise

- Motion

$$\hat{x}_{k'} = \hat{x}_k + u(t_{k+1} - t_k)$$

$$\sigma_{k'}^2 = \sigma_k^2 + \sigma_w^2[t_{k+1} - t_k]$$

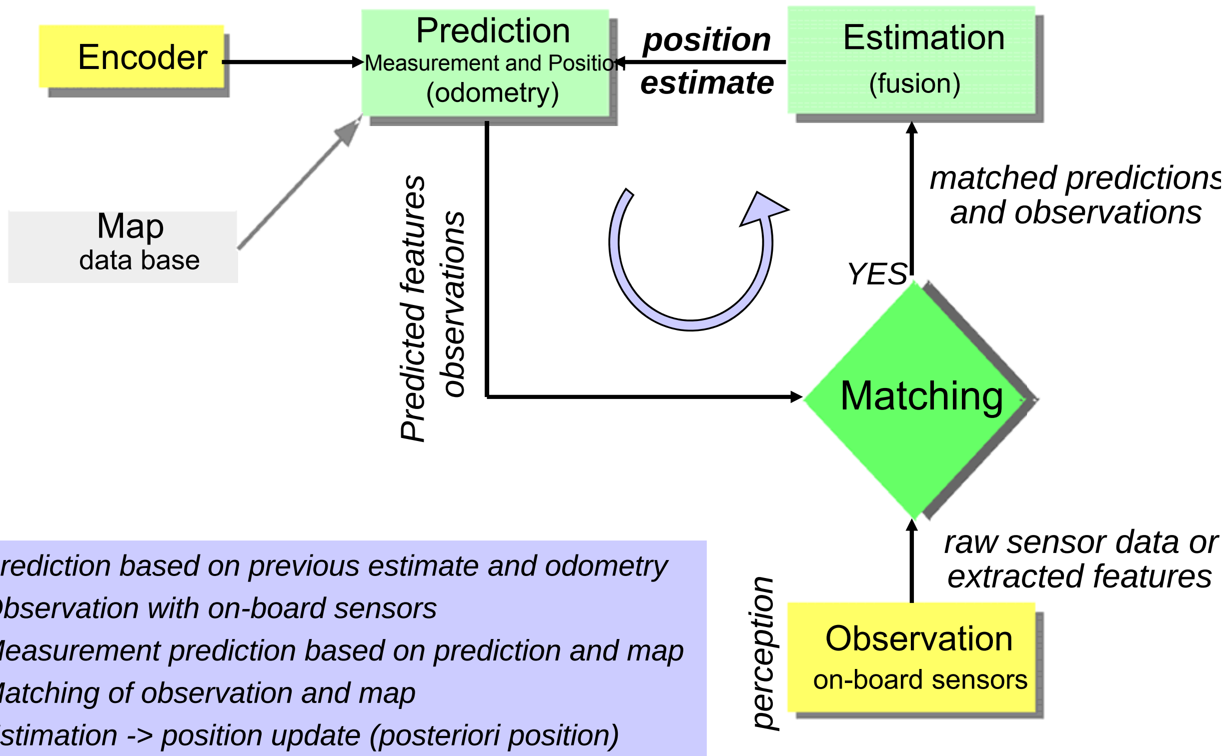


- Combining fusion and dynamic prediction

$$\begin{aligned}\hat{x}_{k+1} &= \hat{x}_{k'} + K_{k+1}(z_{k+1} - \hat{x}_{k'}) \\ &= [\hat{x}_k + u(t_{k+1} - t_k)] + K_{k+1}[z_{k+1} - \hat{x}_k - u(t_{k+1} - t_k)]\end{aligned}$$

$$K_{k+1} = \frac{\sigma_{k'}^2}{\sigma_{k'}^2 + \sigma_z^2} = \frac{\sigma_k^2 + \sigma_w^2[t_{k+1} - t_k]}{\sigma_k^2 + \sigma_w^2[t_{k+1} - t_k] + \sigma_z^2}$$

10 The Five Steps for Map-Based Localization



1. *Prediction based on previous estimate and odometry*
2. *Observation with on-board sensors*
3. *Measurement prediction based on prediction and map*
4. *Matching of observation and map*
5. *Estimation -> position update (posteriori position)*

11 Kalman Filter for Mobile Robot Localization: Robot Position Prediction

- In a first step, the robots position $\hat{x}_t$ at time step *t* is predicted based on its old location (time step *t*-1) and its movement due to the control input u_t:

$$\hat{x}_t = f(x_{t-1}, u_t)$$

f: Odometry function

Jacobians of *f*

$$\hat{P}_t = F_x \cdot P_{t-1} \cdot F_x^T + F_u \cdot Q_t \cdot F_u^T$$

Covariance of previous robot state

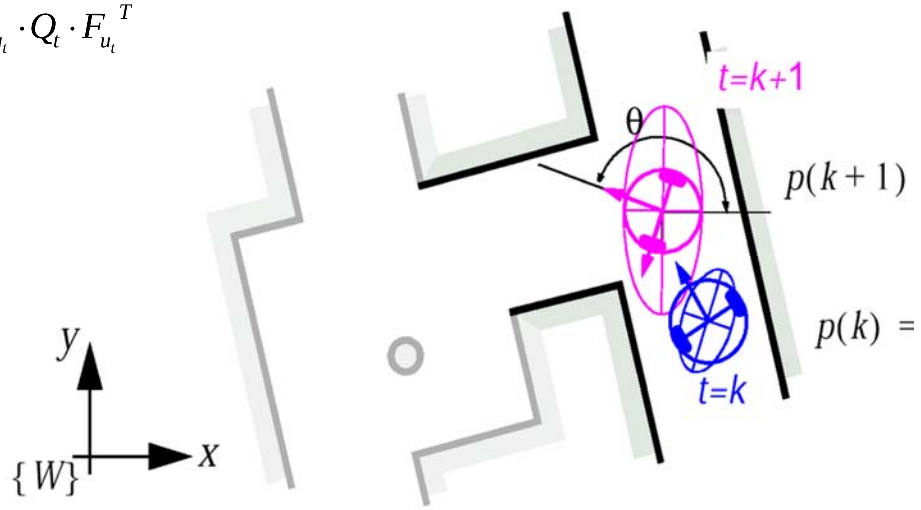
Covariance of noise associated to the motion

12 Kalman Filter Localization: Prediction update

$$\hat{x}_t = f(x_{t-1}, u_t) = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos(\theta_{t-1} + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin(\theta_{t-1} + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix} \qquad Q_t = \begin{bmatrix} k_r |\Delta s_r| & 0 \\ 0 & k_l |\Delta s_l| \end{bmatrix}$$

$$\hat{P}_t = F_{x_{t-1}} \cdot P_{t-1} \cdot F_{x_{t-1}}^{\ T} + F_{u_t} \cdot Q_t \cdot F_{u_t}^{\ T}$$

Odometry

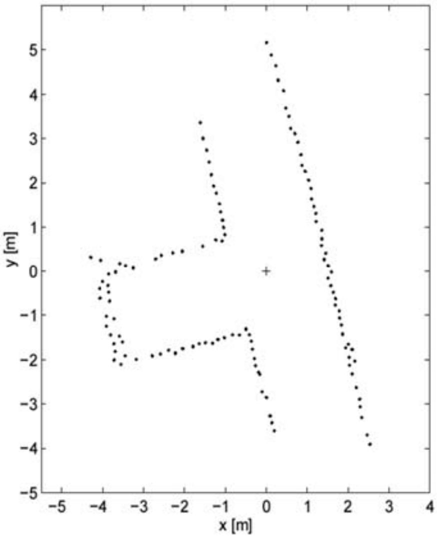


13 Kalman Filter for Mobile Robot Localization: Observation

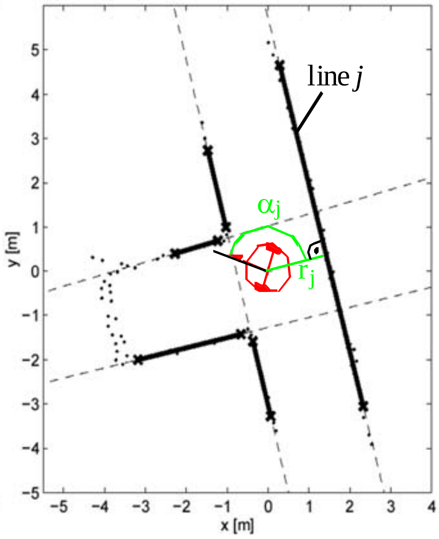
- The second step it to obtain the observation Z (measurements) from the robot's sensors at the new location
- The observation usually consists of a set n_0 of single observations z_j extracted from the different sensors signals. It represents *features* like *lines*, *doors* or *any kind of landmarks*.
- The parameters of the targets are usually observed in the sensor/robot frame *{R}*.
  - Therefore the observations have to be transformed to the world frame {W} or
  - the measurement prediction have to be transformed to the sensor frame {R}.
  - This transformation is specified in the function h_j (see later).

14 Observation

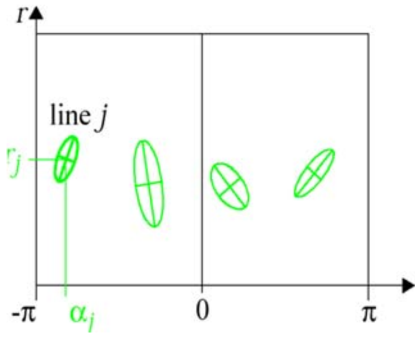
Raw Date of Laser Scanner



Extracted Lines



Extracted Lines in Model Space



$$z_t^i = {}^{R}\begin{bmatrix} \alpha_t^i \\ r_t^i \end{bmatrix}$$

Sensor (robot) frame

$$R_t^i = \begin{bmatrix} \sigma_{\alpha\alpha} & \sigma_{\alpha r} \\ \sigma_{r\alpha} & \sigma_{rr} \end{bmatrix}_t^i$$

15 Measurement Prediction

- In the next step we use the predicted robot position $\hat{x}_t$ and the features m^j in the map M to generate multiple predicted observations $\hat{z}_t^j$
- They have to be transformed into the sensor frame

$$\hat{z}_t^j = h^j\left(\hat{x}_t, m^j\right)$$

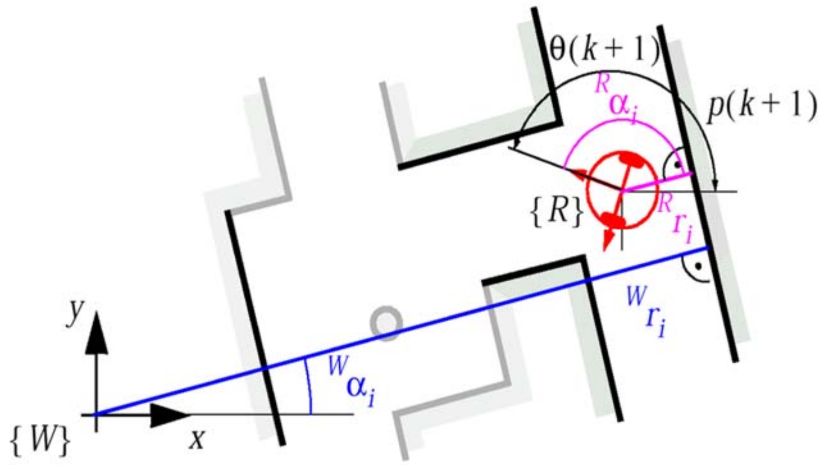
- We can now define the measurement prediction as the set containing all n^j predicted observations

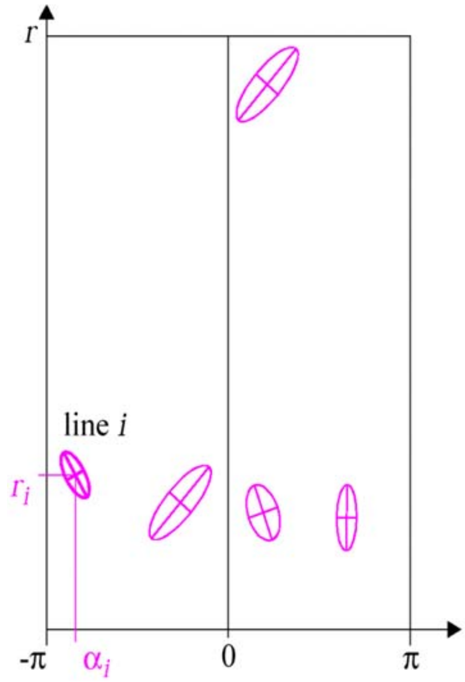
$$\hat{Z} = \left\{\hat{z}_j \middle| \left(1 \le j \le n_i\right)\right\}$$

- The function h^j is mainly the coordinate transformation between the world $\{W\}$ frame and the sensor/robot frame $\{R\}$

16 Measurement Prediction

- For prediction, only the walls that are in the field of view of the robot are selected.
- This is done by linking the individual lines to the nodes of the path





17 Kalman Filter for Mobile Robot Localization: Measurement Prediction: *Example*

- The generated measurement predictions have to be transformed to the robot frame $\{R\}$

$$m^j = {}^{\{W\}}\begin{bmatrix} \alpha_t^j \\ r_t^j \end{bmatrix} \longrightarrow \hat{z}_t^j = \begin{bmatrix} \hat{\alpha}_t^j \\ \hat{r}_t^j \end{bmatrix}$$

- According to the figure in previous slide the transformation is given by

$$\hat{z}_t^j = \begin{bmatrix} \hat{\alpha}_t^j \\ \hat{r}_t^j \end{bmatrix} = h^j(\hat{x}_t, m^j) = \begin{bmatrix} {}^{\{W\}}\alpha_t^j - \hat{\theta}_t \\ {}^{\{W\}}r_t^j - (\hat{x}_t \cos({}^{\{W\}}\alpha_t^j) + \hat{y}_t \sin({}^{\{W\}}\alpha_t^j)) \end{bmatrix}$$

and its Jacobian by

$$H^j = \begin{bmatrix} \frac{\partial \alpha_t^j}{\partial \hat{x}} & \frac{\partial \alpha_t^j}{\partial \hat{y}} & \frac{\partial \alpha_t^j}{\partial \hat{\theta}} \\ \frac{\partial r_t^j}{\partial \hat{x}} & \frac{\partial r_t^j}{\partial \hat{y}} & \frac{\partial r_t^j}{\partial \hat{\theta}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ -\cos({}^{\{W\}}\alpha_t^j) & -\sin({}^{\{W\}}\alpha_t^j) & 0 \end{bmatrix},$$

18 Kalman Filter for Mobile Robot Localization: Matching

- Assignment from observations m^j (gained by the sensors) to the targets z_t (stored in the map)
- For each measurement prediction for which an corresponding observation is found we calculate the innovation:

$$v_t^{ij} = [z_t^i - \hat{z}_t^j] = [z_t^i - h^j(\hat{x}_t, m^j)] =$$

$$= \begin{bmatrix} \alpha_t^i \\ r_t^i \end{bmatrix} - \begin{bmatrix} {}^{\{W\}}\alpha_t^j - \hat{\theta}_t \\ {}^{\{W\}}r_t^j - (\hat{x}_t \cos({}^{\{W\}}\alpha_t^j) + \hat{y}_t \sin({}^{\{W\}}\alpha_t^j)) \end{bmatrix}$$

and its innovation covariance found by applying the error propagation law:

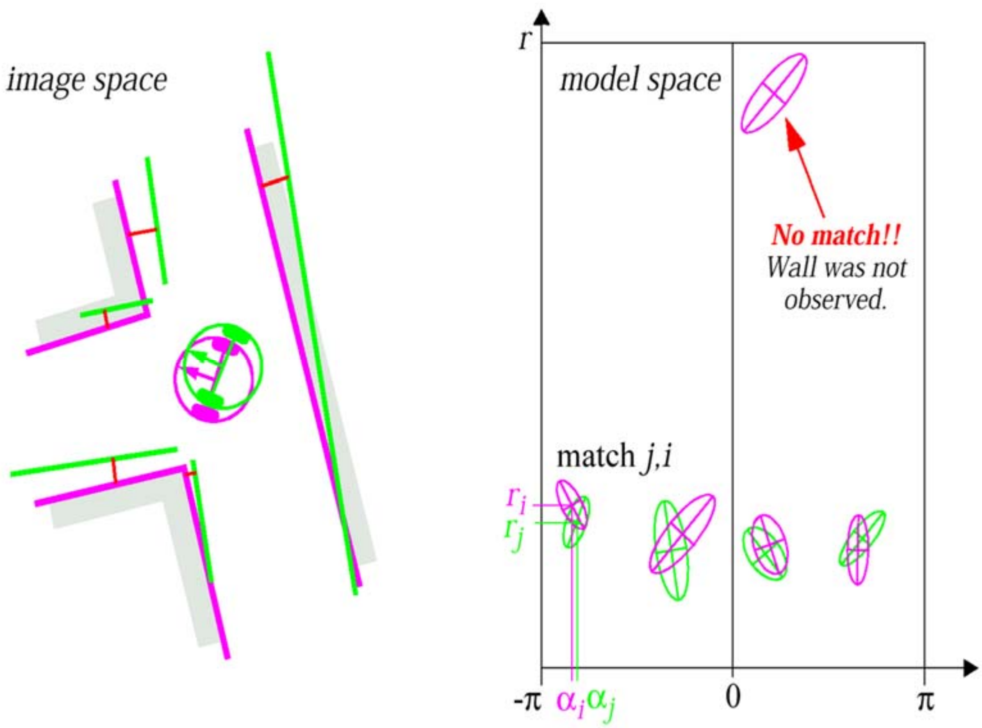
$$\Sigma_{IN_t}^{ij} = H^j \cdot \hat{P}_t \cdot H^{jT} + R_t^i$$

H^j: Jacobian

- The validity of the correspondence between measurement and prediction can e.g. be evaluated through the Mahalanobis distance:

$$v_t^{ijT} \cdot (\Sigma_{IN_t}^{ij})^{-1} \cdot v_t^{ij} \le g^2$$

19 Matching



20 Kalman Filter for Mobile Robot Localization: Estimation: Applying the Kalman Filter

- Kalman filter gain:

$$K_t = \hat{P}_t \cdot H_t^T \cdot (\Sigma_{IN_t})^{-1}$$

- Update of robot's position estimate:

$$x_t = \hat{x}_t + K_t v_t ,$$

- The associate variance

$$P_t = \hat{P}_t - K_t \cdot \Sigma_{IN_t} \cdot K_t^T$$

22 Estimation

- Kalman filter estimation of the new robot position x_t :
  - By fusing the prediction of robot position (magenta) with the innovation gained by the measurements (green) we get the updated estimate of the robot position (red)