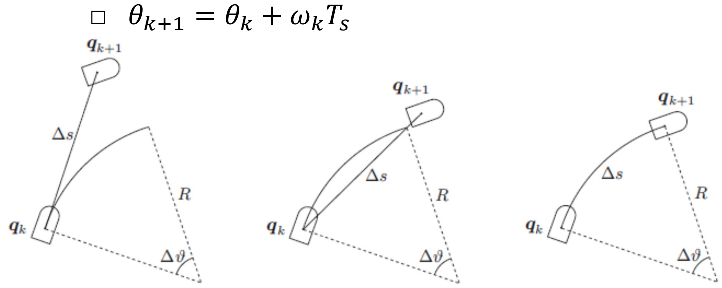
LOCALIZATION

miércoles, 30 de octubre de 2019 9:42 a. m.

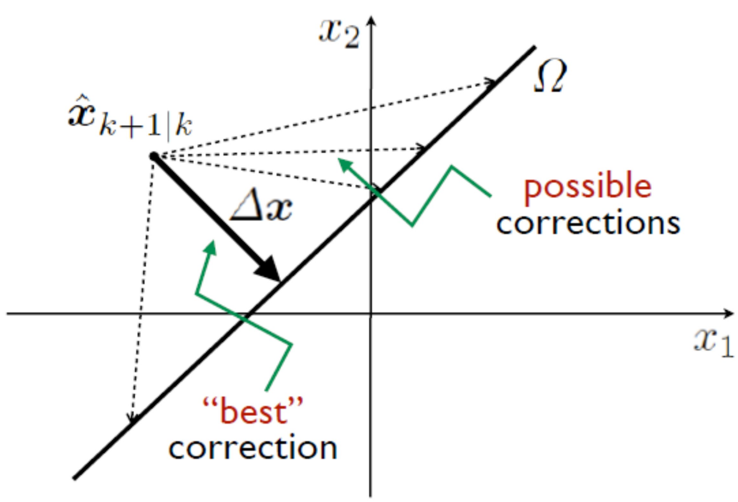
- **Localization** is the proces of estimating the robot configuration q in real time. Which is required for planning and control.
- **Odometric Localization** (dead reckoning):
  - Basic Form of localization using proprioceptive sensors.
  - WMRs are equiped with encoders that allow to measure the rotation of the wheels but not the actual position and orientation.
  - **Integration**: if we assume that time intervals are small enough, then v_k and ω_k are constant during the k-th inverval and knowing the initial configuration q_k allows the computation of the next configuration. Each sampling interval the robot moves along an arc of circle (a line segment if $\omega_k = 0$)
    - Euler Integration:
      - $x_{k+1} = x_k + v_k T_s \cos\theta_k$
      - $y_{k+1} = y_k + v_k T_s \sin\theta_k$
      - $\theta_{k+1} = \theta_k + \omega_k T_s$
      - Where $T_s = t_{k+1} - t_k$
      - Note that x_{k+1} and y_{k+1} are approximate while θ_{k+1} is exact.
    - Runge-Kutta integration (2nd order):
      - $x_{k+1} = x_k + v_k T_s \cos\left(\theta_k + \dfrac{\omega_k T_s}{2}\right)$
      - $y_{k+1} = y_k + v_k T_s \sin\left(\theta_k + \dfrac{\omega_k T_s}{2}\right)$
      - $\theta_{k+1} = \theta_k + \omega_k T_s$
    - Exact Integration:
      - $x_{k+1} = x_k + \dfrac{v_k}{\omega_k}(\sin\theta_{k+1} - \sin\theta_k)$
      - $y_{k+1} = y_k - \dfrac{v_k}{\omega_k}(\cos\theta_{k+1} - \cos\theta_k)$
      - $\theta_{k+1} = \theta_k + \omega_k T_s$



Euler Runge-Kutta exact

  - In practice the commanded inputs are not used for the integration because the actuation is never ideal. Instead we measure the effect of the actual inputs and use those in the integration.
    - $\Delta s = v_k T_s$
    - $\Delta\theta = \omega_k T_s$
    - $\dfrac{\Delta s}{\Delta\theta} = \dfrac{v_k}{\omega_k}$
  - This approach is subject to an error (odometric drift) that grows over time. Some reasons are:

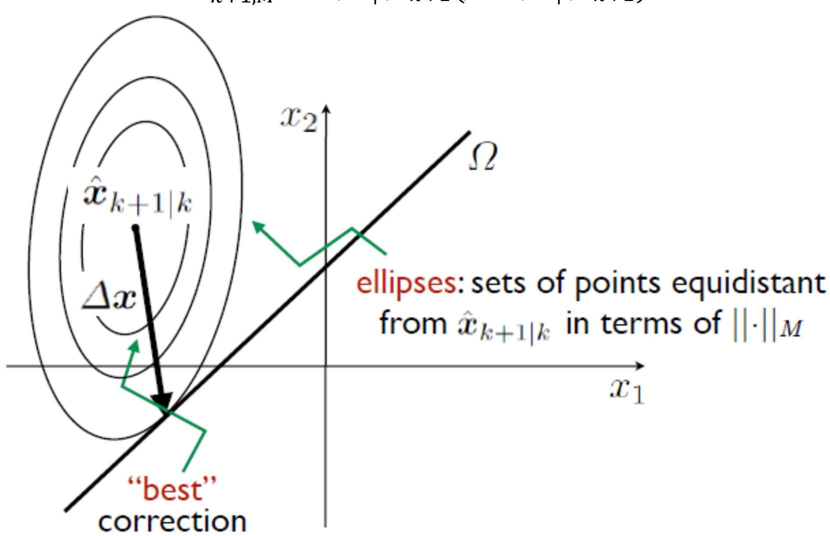
        - Wheel slippage
        - Inaccurate calibration
        - Numerical integration error
    - Effective localization methods use both proprioceptive and exteroceptive sensors
- **Kalman Filter**:
    - Since methods using only proprioceptive sensors are subject to an error that diverges over time. Effective localization methods use also exteroceptive sensors, comparing the external readings with those predicted.
    - **Probabilistic localization**: maintain a probability distribution over the space of all possible hypotheses.
    - Steps:
        - Prediction: use the dynamics to generate an estimate of the next state.
        - Uptade: make a correction of the estimate comparing it with the observation.
    - Without Noise:
        - System:
            - $\hat{x}_{k+1|k} = A_k\hat{x}_k + B_k u_k$
            - $y_k = C_k x_k$
        - Prediction:
            - $\hat{x}_{k+1|k} = A_k\hat{x} + B_k u_k$
        - Uptade:
            - The correction must satisfy:
                - $C_{k+1}(\hat{x}_{k+1|k} + \Delta x) = y_{k+1}$
            - Also, we would like to minimize that correction
                - $\min\|\Delta x\|$
                - $C_{k+1}\Delta x = y_{k+1} - C_{k+1}\hat{x}_{k+1|k}$
                - $\Delta x = C_{k+1}^{\dagger}(y_{k+1} - C_{k+1}\hat{x}_{k+1|k})$



            - Where we define as the innovation
                - $\nu_{k+1} = y_{k+1} - C_{k+1}\hat{x}_{k+1|k}$
            - So that
                - $\Delta x = C_{k+1}^{\dagger}\nu_{k+1}$
            - Finally
                - $\hat{x}_{k+1} = \hat{x}_{k+1|k} + \Delta x$
                - $\hat{x}_{k+1} = \hat{x}_{k+1|k} + C_{k+1}^{\dagger}\nu_{k+1}$
        - In general this approach will not converge because the correction is **naive**. We need a new structure that takes into account the presence of **noise**.
    - With Process Noise:
        - System:

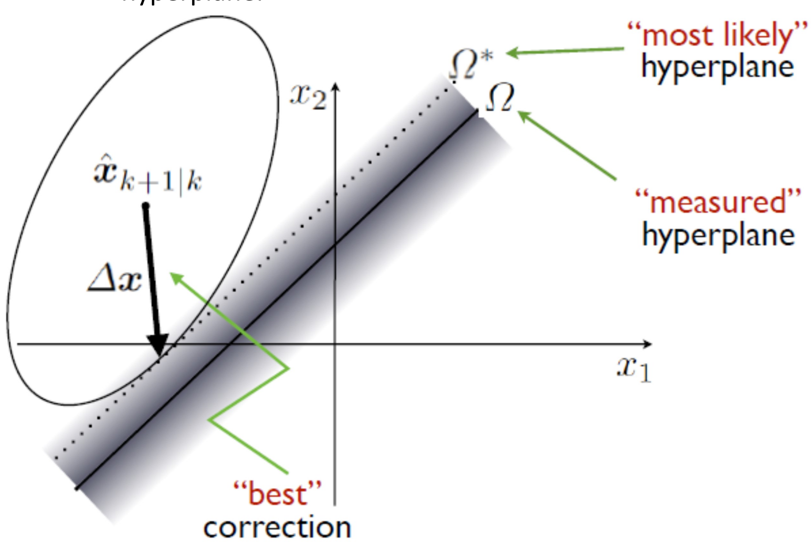
- $\hat{x}_{k+1|k} = A_k \hat{x}_k + B_k u_k + v_k$
- $y_k = C_k x_k$

- v_k is a **white gaussian** noise with zero mean and covariance matrix V_k
- Since the process is now random we must estimate both the state and the associated covariance.
- Prediction:
  - State:
    - $\hat{x}_{k+1|k} = A_k \hat{x}_k + B_k u_k$
    - Because v_k has zero mean
  - Covariance:
    - $P_{k+1|k} = E\left((x_{k+1} - \hat{x}_{k+1|k})(x_{k+1} - \hat{x}_{k+1|k})^T\right)$
    - $P_{k+1|k} = E\left((A_k(x_k - \hat{x}_k) + v_k)(A_k(x_k - \hat{x}_k) + v_k)^T\right)$
    - $P_{k+1|k} = E\left(A_k(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T A_k^T\right) + E\left(v_k v_k^T\right)$
    - $P_{k+1|k} = A_k P_k A_k^T + V_k$
- Uptdate:
  - State:
    - We want to choose Δx so that the probability (gaussian) is maximized
      - $p(x) = \frac{1}{\sqrt{(2\pi)^n |P_{k+1|k}|}} \exp -\frac{1}{2}\left(x - \hat{x}_{k+1|k}\right)^T P_{k+1|k}^{-1}\left(x - \hat{x}_{k+1|k}\right)$
    - $p(x)$ is maximized when the exponent is minimized. So the solution to the problem is
      - $\min \|\Delta x\|_M ; C_{k+1}\Delta x = y_{k+1} - C_{k+1}\hat{x}_{k+1|k}$
    - Where $\|\Delta x\|_M^2 = \Delta x^T P_{k+1|k}^{-1} \Delta x$ is the **Mahalanobis distance**.
    - Finally
      - $\Delta x = C_{k+1,M}^{\dagger} \nu_{k+1}$
    - Where $C_{k+1,M}^{\dagger}$ is the weighted pseudoinverse of C_{k+1}
      - $C_{k+1,M}^{\dagger} = P_{k+1|k} C_{k+1}^T \left(C_{k+1} P_{k+1|k} C_{k+1}^T\right)^{-1}$



  - Covariance:
    - $P_{k+1} = P_{k+1|k} - C_{k+1,M}^{\dagger} C_{k+1} P_{k+1|k}$
- Wrap-up:
  - $\hat{x}_{k+1|k} = A_k \hat{x}_k + B_k u_k$
  - $P_{k+1|k} = A_k P_k A_k^T + V_k$
  - $\hat{x}_{k+1} = \hat{x}_{k+1|k} + C_{k+1,M}^{\dagger} \nu_{k+1}$
  - $P_{k+1} = P_{k+1|k} - C_{k+1,M}^{\dagger} C_{k+1} P_{k+1|k}$

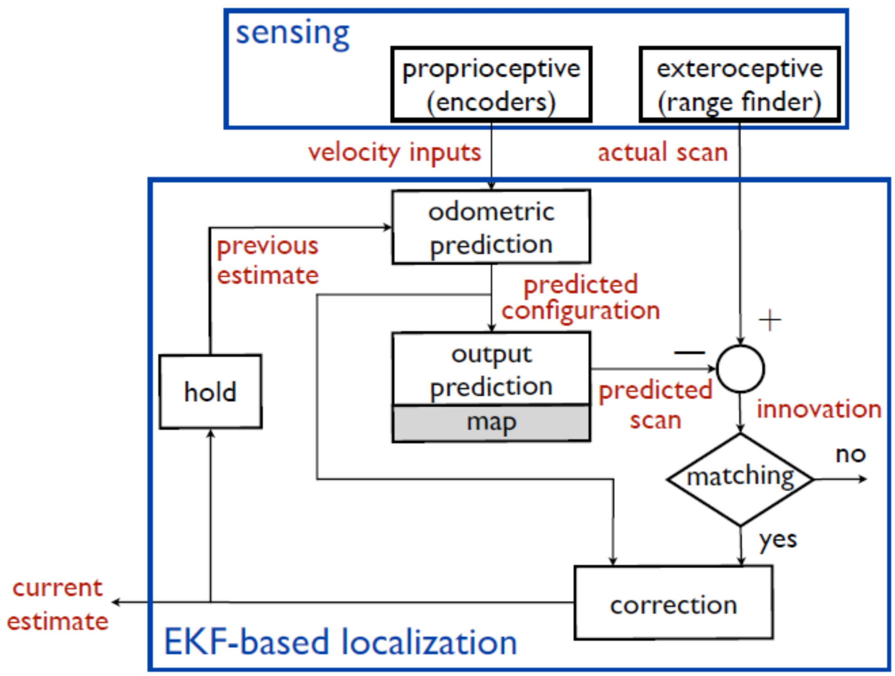
        - The problem with this approach is that there is no measurement noise so the covariance estimate will become singular because there is no uncertainty in the normal direction to the measurement hyperplane.
    - Full Kalman Filter
        - System:
            - $\hat{x}_{k+1|k} = A_k \hat{x}_k + B_k u_k + v_k$
            - $y_k = C_k x_k + w_k$
            - Where v_k and w_k are white gaussian noises with zero mean and covariance matrices V_k and W_k
        - Prediction:
            - $\hat{x}_{k+1|k} = A_k \hat{x}_k + B_k u_k$
            - $P_{k+1|k} = A_k P_k A_k^T + V_k$
        - Update:
            - State: Due to sensor noise the output is no longer certain.
                - First we compute the most likely output given the predictions, then the most likely hyperplane, and finally compute the correction using that hyperplane.



            - $\hat{x}_{k+1} = \hat{x}_{k+1|k} + R_{k+1} v_{k+1}$
            - $P_{k+1} = P_{k+1|k} - R_{k+1} C_{k+1} P_{k+1|k}$
            - Where R_{k+1} is the **Kalman gain matrix.**
                - $R_{k+1} = P_{k+1|k} C_{k+1}^T (C_{k+1} P_{k+1|k} C_{k+1}^T + W_{k+1})$
        - Even if the noises have non-gaussian distributions, the KF is sill the best linear estimator.
        - If the process is **observable**, then the estimate **converges.**
    - **Extended Kalman Filter:**
        - System:
            - $x_{k+1} = f_k(x_k, u_k) + v_k$
            - $y_k = h_k(x_k) + w_k$
        - If the functions are continually differentiable then one way to build a filter is to linearize the system's dynamic equations around the current estimate and then apply the KF equations to the linear approximation.
- **EKF Localization**:
    - Landmark-based:
        - The position of landmarks is fixed and known
        - For each measurement the identity of the landmark is known, i.e. landmarks are tagged.
        - If landmarks are indistinguishable then **data association** must be performed in a way

that minimizes the innovation.

- Map-based:
    - A metric map of the environment is known.
    - Instead of looking for landmarks in the measurements you can use the whole measurement as an output vector. The innovation is computed by comparing the actual readings with the predicted readings given the predicted configuration.
    - This gives a large innovation vector, as large as the measurements.
    - No data association.
    - Problem: One particular measurement can happen in different positions in the map, i.e. the measurements are not unique. This is called **aliasing**.



- **SLAM**: Simultaneous Localization and Mapping
    - If we remove the assumption that the map is known then we need to use the sensors to build a map and localize itself.
    - Can be map-based or landmark-based as was the case with EKF localization, with the difference that the position of the landmarks or the map are not known a priori.

Autonomous and Mobile Robotics

Prof. Giuseppe Oriolo

Localization 1

Odometric Localization

DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

- planning and feedback control require the knowledge of the robot configuration q (e.g., see Motion Control of WMRs: Trajectory Tracking, slide 3)

- in robot manipulators, joint encoders provide a direct measure of q

- WMRs are equipped with incremental encoders that measure only the rotation of the wheels, not the position and orientation of the vehicle

- localization is a procedure for estimating the robot configuration q, typically in real time

- consider a unicycle under constant velocity inputs v_k, ω_k in $[t_k, t_{k+1}]$, as in a digital control implementation; in each sampling interval, the robot moves along an arc of circle of radius v_k/ω_k (a line segment if $\omega_k = 0$)
- assume $\boldsymbol{q}_k$, v_k and ω_k are known; compute $\boldsymbol{q}_{k+1}$ by integration of the kinematic model over $[t_k, t_{k+1}]$
- first possibility: Euler integration (easiest)

$$\begin{aligned} x_{k+1} &= x_k + v_k T_s \cos\theta_k \\ y_{k+1} &= y_k + v_k T_s \sin\theta_k \qquad T_s = t_{k+1} - t_k \\ \theta_{k+1} &= \theta_k + \omega_k T_s \end{aligned}$$

- x_{k+1} and y_{k+1} are approximate; θ_{k+1} is exact

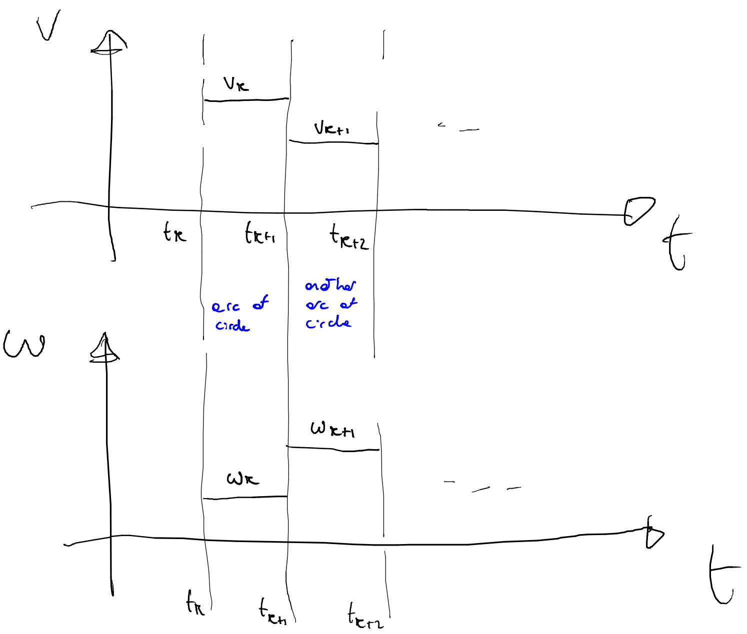
$$\dot x = v\cos\theta$$
$$\dot y = v\sin\theta$$
$$\dot\theta = \omega$$

$$\Downarrow$$

$\theta_k = \theta_k + \omega_k T_s$ exact

for the others we use the Euler integration using θ_k

$$\left.\begin{aligned} x_{k+1} &= x_k + v_k T_s \, c\theta_k \\ y_{k+1} &= y_k + v_k T_s \, s\theta_k \end{aligned}\right\} \text{approximate}$$



- second possibility: 2nd order Runge-Kutta integration

$$
\begin{aligned}
x_{k+1} &= x_k + v_k T_s \cos\left(\theta_k + \frac{\omega_k T_s}{2}\right) \\
y_{k+1} &= y_k + v_k T_s \sin\left(\theta_k + \frac{\omega_k T_s}{2}\right) \\
\theta_{k+1} &= \theta_k + \omega_k T_s
\end{aligned}
$$

- the average orientation during $[t_k, t_{k+1}]$ is used

- as a consequence, x_{k+1} and y_{k+1} are still approximate, but more accurate

- third possibility: exact integration

$$
\begin{aligned}
x_{k+1} &= x_k + \frac{v_k}{\omega_k}(\sin\theta_{k+1} - \sin\theta_k) \\
y_{k+1} &= y_k - \frac{v_k}{\omega_k}(\cos\theta_{k+1} - \cos\theta_k) \\
\theta_{k+1} &= \theta_k + \omega_k T_s
\end{aligned}
$$

- for $\omega_k=0$, x_{k+1} and y_{k+1} are still defined and coincide with the solution by Euler and Runge-Kutta

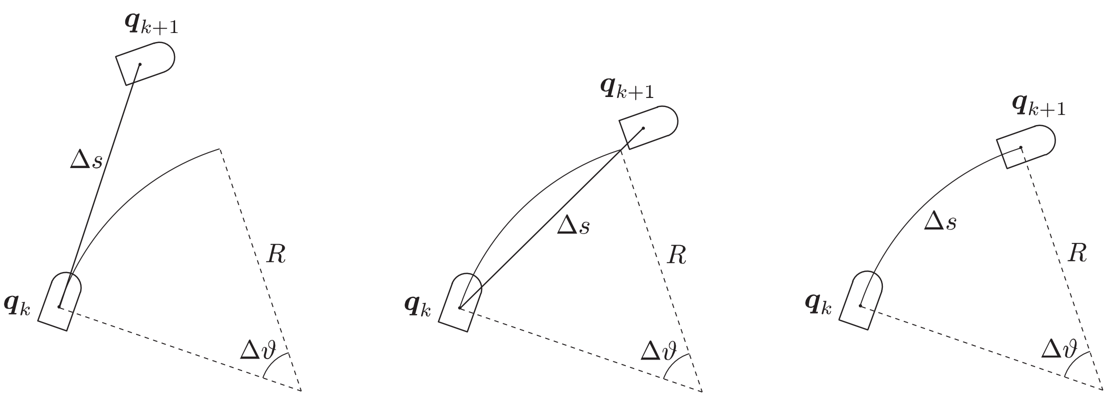
- for $\omega_k \approx 0$, a conditional instruction may be used in the implementation

$\theta(t) = \theta_k + \omega_k (t - t_k) \qquad t \in [t_k, t_{k+1}]$

replacing $\theta(t)$ in $\dot{x}, \dot{y} \ldots$

$\dot{x} = v \cos(\theta_k + \omega_k (t - t_k)) \Rightarrow x(t) = x_k + v_k \int_{t_k}^{t_{k+1}} \cos(\theta_k + \omega_k (t - t_k))\, dt$

geometric comparison



Euler Runge-Kutta exact

- in practice, due to the non-ideality of any actuation system, the commanded inputs v_k and ω_k are not used
- instead, measure the effect of the actual inputs:

$$v_k T_s = \Delta s \qquad \omega_k T_s = \Delta\theta \qquad \frac{v_k}{\omega_k} = \frac{\Delta s}{\Delta\theta}$$

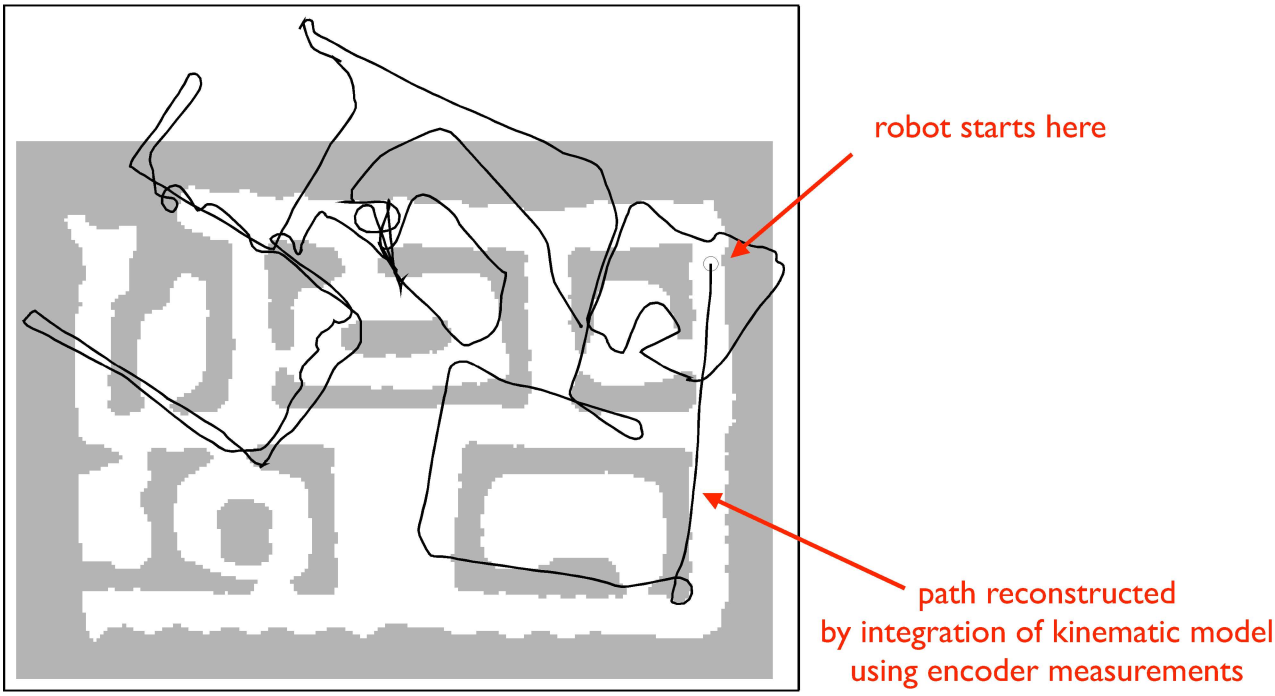
  Δs (traveled length) and $\Delta\theta$ (total orientation change) are reconstructed via proprioceptive sensors

- for example, for a differential-drive robot

$$\Delta s = \frac{r}{2}(\Delta\phi_R + \Delta\phi_L) \qquad \Delta\theta = \frac{r}{d}(\Delta\phi_R - \Delta\phi_L)$$

  where $\Delta\phi_R$ and $\Delta\phi_L$ are the total rotations measured by the wheel encoders

- maintaining an estimate of the robot configuration by iterative integration of the kinematic model is called odometric localization or dead reckoning
- subject to an error (odometric drift) that grows over time, becoming significant over sufficiently long paths
- causes include wheel slippage (model perturbation), inaccurate calibration of, e.g., wheel radius (model uncertainty) or numerical integration error
- effective localization methods use proprioceptive as well as exteroceptive sensors



a typical dead reckoning result

Autonomous and Mobile Robotics

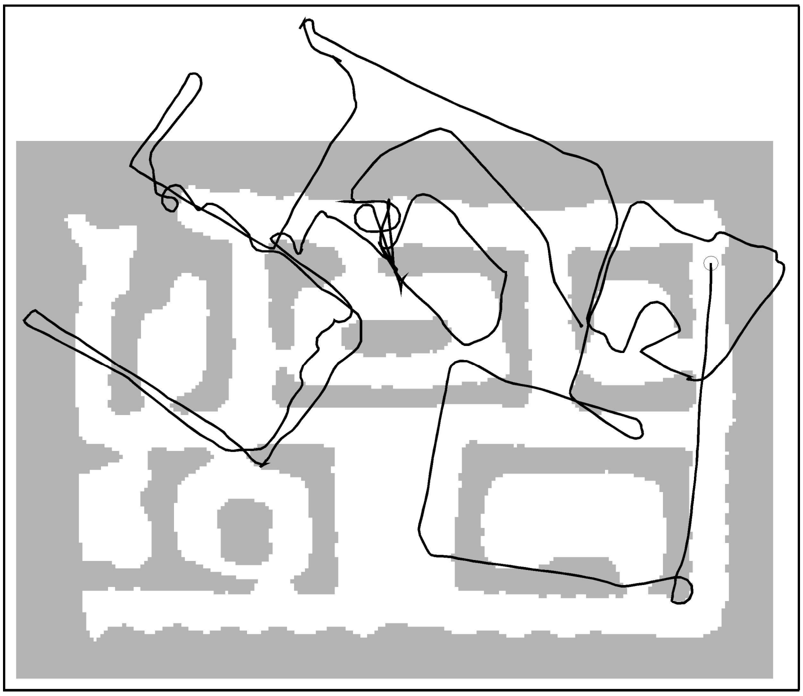
Prof. Giuseppe Oriolo

Localization 2
Kalman Filter

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

- recall: estimating the robot configuration by iterative integration of the kinematic model (dead reckoning) is subject to an error that diverges over time
- effective localization methods use proprioceptive as well as exteroceptive sensors: if an environment map is known, compare the actual sensor readings with those predicted using the current estimate
- probabilistic localization: instead of maintaining a single hypothesis on the configuration, maintain a probability distribution over the space of all possible hypotheses
- one possible approach: use a Kalman Filter



a typical dead reckoning result

basic concepts

- given a vector random variable $\boldsymbol{X}$ with probability density function $f_{\boldsymbol{X}}(\boldsymbol{x})$, its expected (or mean) value is

$$E(\boldsymbol{X}) = \bar{\boldsymbol{X}} = \int_{\boldsymbol{x}\in\mathbb{R}^n} \boldsymbol{x} \underbrace{f_{\boldsymbol{X}}(\boldsymbol{x})}_{\text{probability density function}} d\boldsymbol{x}$$

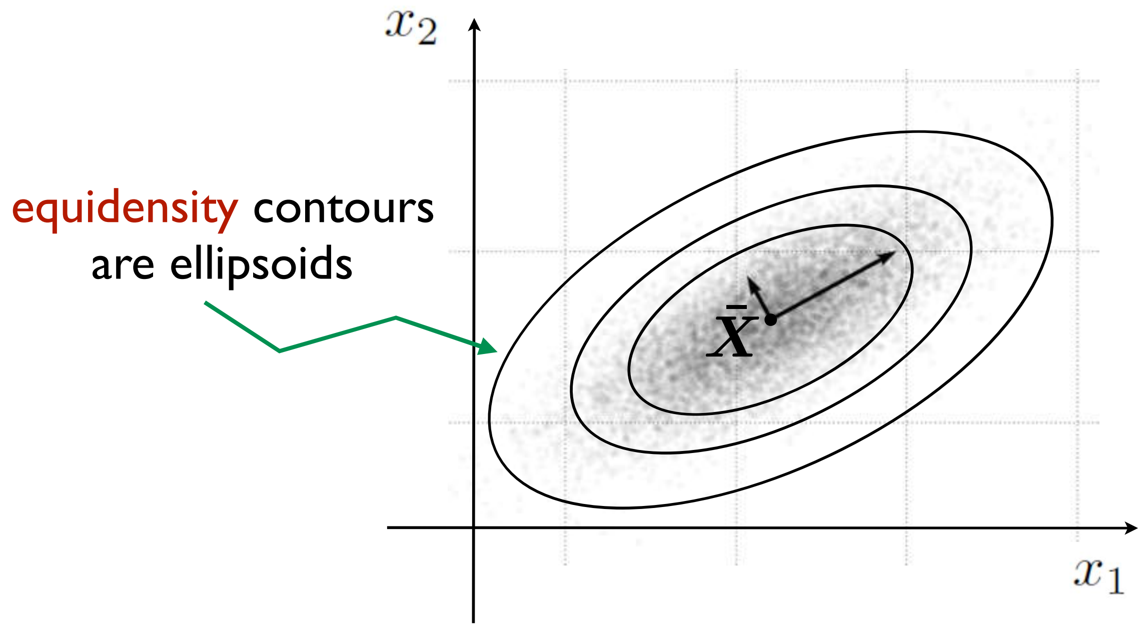
- its covariance matrix is

$$\boldsymbol{P}_{\boldsymbol{X}} = E\left((\boldsymbol{X}-\bar{\boldsymbol{X}})(\boldsymbol{X}-\bar{\boldsymbol{X}})^T\right)$$

- $\boldsymbol{X}$ has a multivariate gaussian distribution if

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^n|\boldsymbol{P}_{\boldsymbol{X}}|}} e^{-\frac{1}{2}(\boldsymbol{x}-\bar{\boldsymbol{X}})^T\boldsymbol{P}_{\boldsymbol{X}}^{-1}(\boldsymbol{x}-\bar{\boldsymbol{X}})}$$

- geometric interpretation (of this Gaussian Distribution)



- the principal axes are directed as the eigenvectors of $\boldsymbol{P_X}$
- their squared relative lengths are given by the corresponding eigenvalues

Kalman Filter ...without noise (or observer)

- consider a linear discrete-time system without noise

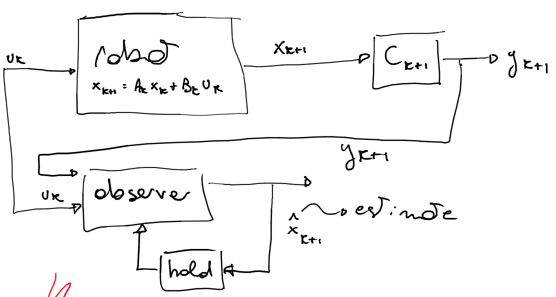
$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{B}_k \boldsymbol{u}_k$$
$$\boldsymbol{y}_k = \boldsymbol{C}_k \boldsymbol{x}_k$$

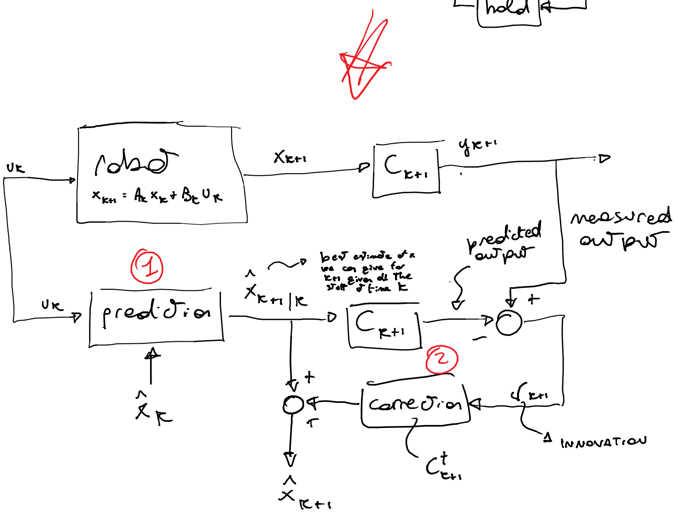
- build a recursive observer that computes an estimate $\hat{\boldsymbol{x}}_{k+1}$ of $\boldsymbol{x}_{k+1}$ from $\boldsymbol{u}_k$, $\boldsymbol{y}_{k+1}$ and previous estimate $\hat{\boldsymbol{x}}_k$

- two steps:

  1. prediction: generate an intermediate estimate $\hat{\boldsymbol{x}}_{k+1|k}$ by propagating $\hat{\boldsymbol{x}}_k$ using the process dynamics
  2. correction (update): correct the prediction on the basis of the difference between the measured and the predicted output

Assumption:
u_k is known





Constraint:

$C_{k+1}\hat{x}_{k+1} = y_{k+1}$

$C_{k+1}(\hat{x}_{k+1|k}+\Delta x) = y_{k+1}$

$C_{k+1}\Delta x = y_{k+1} - C_{k+1}\hat{x}_{k+1|k}$

- prediction

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

assuming that we know $\boldsymbol{A}_k$, $\boldsymbol{B}_k$ and $\boldsymbol{u}_k$

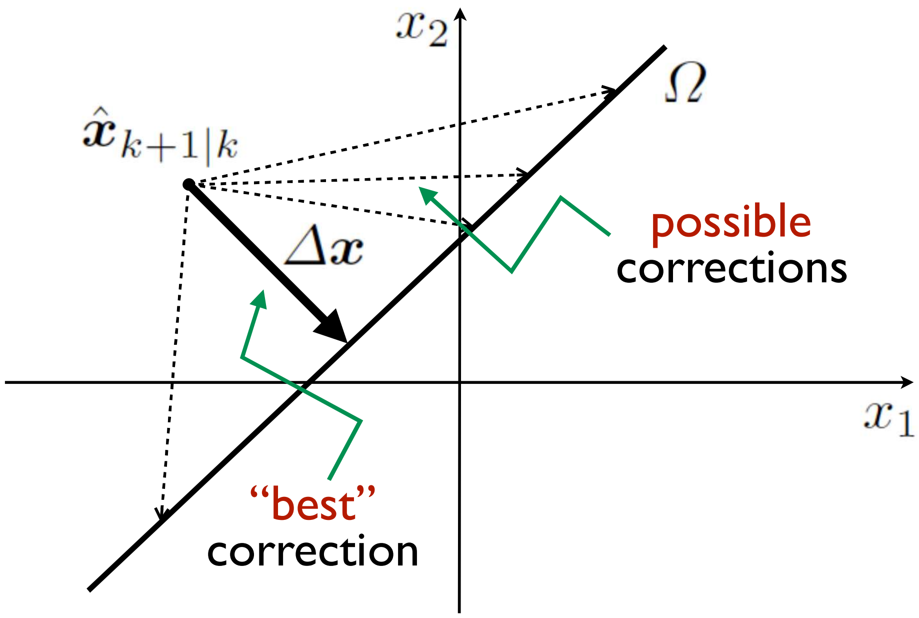
- correction: to be consistent with the measured value of the output, $\boldsymbol{x}_{k+1}$ must belong to the hyperplane

$$\Omega = \{\boldsymbol{x} \,:\, \boldsymbol{C}_{k+1}\boldsymbol{x} = \boldsymbol{y}_{k+1}\}$$

hence the correction $\Delta\boldsymbol{x}$ must satisfy

$$\boldsymbol{C}_{k+1}(\hat{\boldsymbol{x}}_{k+1|k} + \Delta\boldsymbol{x}) = \boldsymbol{y}_{k+1}$$

- geometric interpretation



intuitively, the "best" correction $\Delta\boldsymbol{x}$ is the closest to the prediction, which we believe is accurate

- $\Delta\boldsymbol{x}$ is then the solution of an optimization problem

$$\min \|\Delta\boldsymbol{x}\|$$

minimum norm vector which satisfies the condition

$$\text{s.t. } \boldsymbol{C}_{k+1}\Delta\boldsymbol{x} = \boldsymbol{y}_{k+1} - \boldsymbol{C}_{k+1}\hat{\boldsymbol{x}}_{k+1|k}$$

unknown

- it is well known that

$$\Delta\boldsymbol{x} = \boldsymbol{C}_{k+1}^{\dagger}(\boldsymbol{y}_{k+1} - \boldsymbol{C}_{k+1}\hat{\boldsymbol{x}}_{k+1|k}) = \boldsymbol{C}_{k+1}^{\dagger}\boldsymbol{\nu}_{k+1}$$

where

$$\boldsymbol{C}_{k+1}^{\dagger} = \boldsymbol{C}_{k+1}^{T}(\boldsymbol{C}_{k+1}\boldsymbol{C}_{k+1}^{T})^{-1}$$ pseudoinverse of $\boldsymbol{C}_{k+1}$

$$\boldsymbol{\nu}_{k+1} = \boldsymbol{y}_{k+1} - \boldsymbol{C}_{k+1}\hat{\boldsymbol{x}}_{k+1|k}$$ innovation

note that we have assumed $\boldsymbol{C}_{k+1}$ to be full row rank

- wrapping up, the resulting two-step observer is

$$\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{A}_k \hat{\boldsymbol{x}}_k + \boldsymbol{B}_k \boldsymbol{u}_k$$
$$\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{C}^{\dagger}_{k+1} \boldsymbol{\nu}_{k+1}$$

- in general, the estimate $\hat{\boldsymbol{x}}_{k+1}$ will not converge to the true value $\boldsymbol{x}_{k+1}$ because the correction is naive: estimation errors directed as Ω are not corrected

- we need to modify the above structure to take into account the presence of noise; in doing so, we will fix the above problem

Kalman Filter ...with process noise only

- now include process noise

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

  where $\boldsymbol{v}_k$ is a white gaussian noise with zero mean and covariance matrix $\boldsymbol{V}_k$

- since this is now a random process, we estimate both the state $\boldsymbol{x}_{k+1}$ and the associated covariance $\boldsymbol{P}_{k+1}$

- we keep the prediction/correction structure

- state prediction: as before

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

because $\boldsymbol{v}_k$ has zero mean

- covariance prediction: by definition

$$\begin{aligned}
\boldsymbol{P}_{k+1|k} &= E\left((\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k})(\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k})^T\right) \\
&= E\left((\boldsymbol{A}_k(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k) + \boldsymbol{v}_k)(\boldsymbol{A}_k(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k) + \boldsymbol{v}_k)^T\right) \\
&= E\left(\boldsymbol{A}_k(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k)(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k)^T \boldsymbol{A}_k^T\right) + \\
&\quad E\left(\boldsymbol{A}_k(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k)\boldsymbol{v}_k^T + \boldsymbol{v}_k(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k)^T \boldsymbol{A}_k^T)\right) + E\left(\boldsymbol{v}_k \boldsymbol{v}_k^T\right)
\end{aligned}$$

- now use the linearity of E plus the independence of $\boldsymbol{v}_k$ on $\hat{\boldsymbol{x}}_k$ and $\boldsymbol{x}_k \Rightarrow$ the second term in the rhs is zero

$$x_{k+1} - \hat{x}_{k+1|k} = A_k x_k + \cancel{B_k u_k} + v_k - A_k \hat{x}_k - \cancel{B_k u_k} = A_k(x_k - \hat{x}_k) + v_k$$

$$E\left(A_k(x_k - \hat{x}_k)v_k^T\right) = A_k E\left(x_k - \hat{x}_k\right) \underbrace{E\left(v_k^T\right)}_{0} = 0$$

constant matrix

because the noise has zero mean

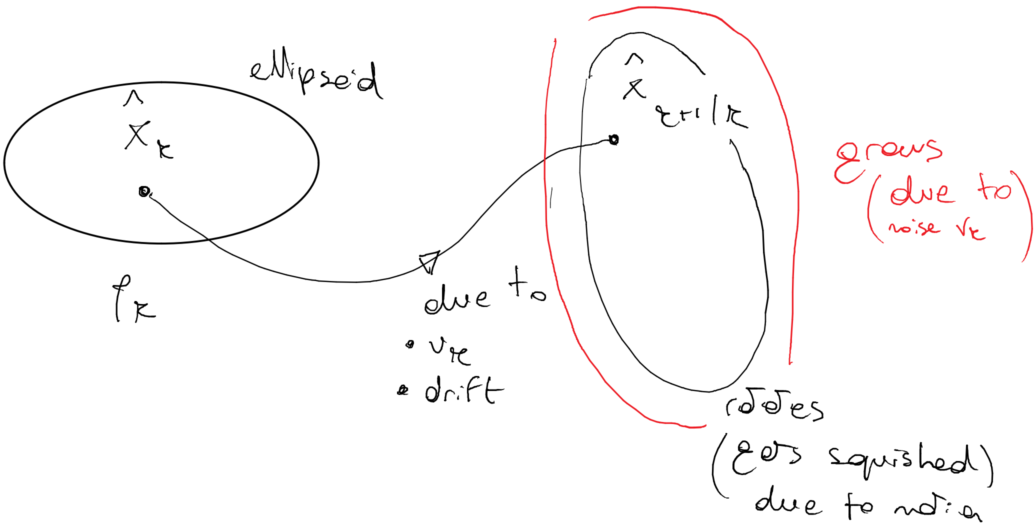
- finally the covariance prediction is

$$\begin{aligned}\boldsymbol{P}_{k+1|k} &= \boldsymbol{A}_k E\left((\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k)(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k)^T\right)\boldsymbol{A}_k^T + E\left(\boldsymbol{v}_k \boldsymbol{v}_k^T\right)\\ &= \boldsymbol{A}_k \boldsymbol{P}_k \boldsymbol{A}_k^T + \boldsymbol{V}_k\end{aligned}$$

- state correction: we should choose $\Delta\boldsymbol{x}$ so as to get the most likely $\boldsymbol{x}$ in Ω, i.e., the $\boldsymbol{x}$ that maximizes the gaussian distribution defined by $\hat{\boldsymbol{x}}_{k+1|k}$ and $\boldsymbol{P}_{k+1|k}$

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{P}_{k+1|k}|}}\, e^{-\frac{1}{2}(\boldsymbol{x} - \hat{\boldsymbol{x}}_{k+1|k})^T \boldsymbol{P}_{k+1|k}^{-1} (\boldsymbol{x} - \hat{\boldsymbol{x}}_{k+1|k})}$$

$p(\boldsymbol{x})$ is maximized when the exponent is minimized



- define the (squared) Mahalanobis distance

$$\Delta \boldsymbol{x}^T \boldsymbol{P}_{k+1|k}^{-1} \Delta \boldsymbol{x} = \|\Delta \boldsymbol{x}\|_M^2$$

- $\Delta \boldsymbol{x}$ is the solution of a new optimization problem

$$\min \|\Delta \boldsymbol{x}\|_M \quad \leadsto$$ minimize no more the euclidean norm but the Mahalanobis norm

$$\text{s.t. } \boldsymbol{C}_{k+1} \Delta \boldsymbol{x} = \boldsymbol{y}_{k+1} - \boldsymbol{C}_{k+1} \hat{\boldsymbol{x}}_{k+1|k}$$

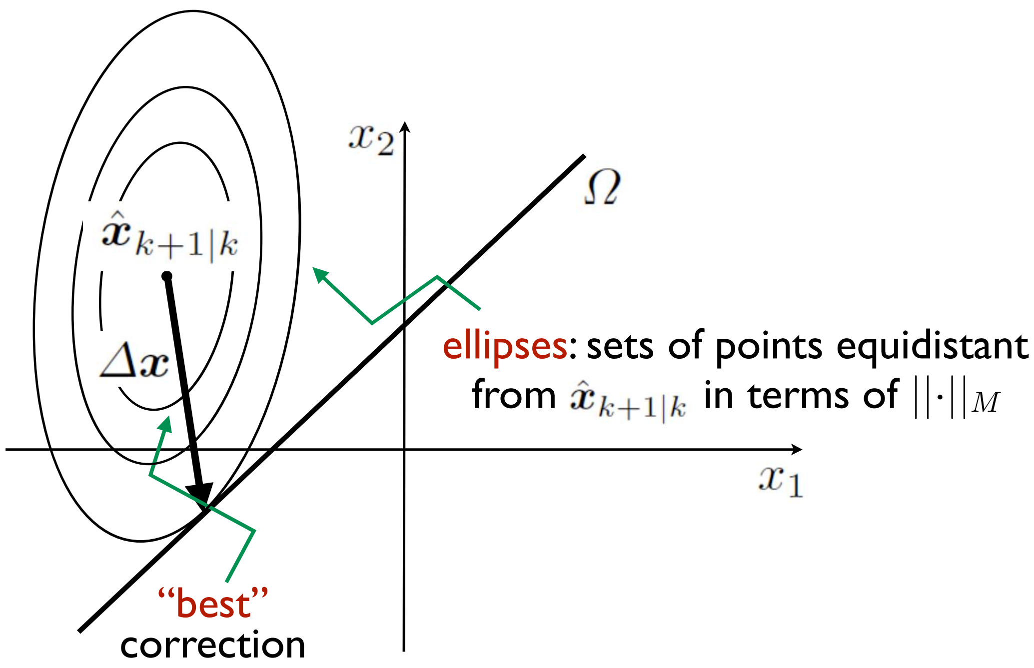
- it is well known that

$$\Delta \boldsymbol{x} = \boldsymbol{C}_{k+1,M}^{\dagger} (\boldsymbol{y}_{k+1} - \boldsymbol{C}_{k+1} \hat{\boldsymbol{x}}_{k+1|k}) = \boldsymbol{C}_{k+1,M}^{\dagger} \boldsymbol{\nu}_{k+1}$$

where $\boldsymbol{C}_{k+1,M}^{\dagger}$ is the weighted pseudoinverse of $\boldsymbol{C}_{k+1}$

$$\boldsymbol{C}_{k+1,M}^{\dagger} = \boldsymbol{P}_{k+1|k} \boldsymbol{C}_{k+1}^T (\boldsymbol{C}_{k+1} \boldsymbol{P}_{k+1|k} \boldsymbol{C}_{k+1}^T)^{-1}$$

- geometric interpretation



the "best" correction is the closest to the prediction according to the current covariance estimate

- covariance correction: using the covariance matrix definition and the state correction one obtains

$$\boldsymbol{P}_{k+1} = \boldsymbol{P}_{k+1|k} - \boldsymbol{C}^{\dagger}_{k+1,M}\boldsymbol{C}_{k+1}\boldsymbol{P}_{k+1|k}$$

- wrapping up, the resulting two-step filter is

$$\begin{aligned}
\hat{\boldsymbol{x}}_{k+1|k} &= \boldsymbol{A}_k\hat{\boldsymbol{x}}_k + \boldsymbol{B}_k\boldsymbol{u}_k \\
\boldsymbol{P}_{k+1|k} &= \boldsymbol{A}_k\boldsymbol{P}_k\boldsymbol{A}_k^T + \boldsymbol{V}_k \\
\hat{\boldsymbol{x}}_{k+1} &= \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{C}^{\dagger}_{k+1,M}\boldsymbol{\nu}_{k+1} \\
\boldsymbol{P}_{k+1} &= \boldsymbol{P}_{k+1|k} - \boldsymbol{C}^{\dagger}_{k+1,M}\boldsymbol{C}_{k+1}\boldsymbol{P}_{k+1|k}
\end{aligned}$$

- problem: no measurement noise $\Rightarrow$ the covariance estimate will become singular (no uncertainty in the normal direction to the measurement hyperplane)

Kalman Filter ...full

- finally include also measurement (sensor) noise

$$\begin{aligned}
\boldsymbol{x}_{k+1} &= \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{B}_k\boldsymbol{u}_k + \boldsymbol{v}_k \\
\boldsymbol{y}_k &= \boldsymbol{C}_k\boldsymbol{x}_k + \boldsymbol{w}_k
\end{aligned}$$

where $\boldsymbol{v}_k$, $\boldsymbol{w}_k$ are white gaussian noises with zero mean and covariance matrices $\boldsymbol{V}_k$, $\boldsymbol{W}_k$

- the dynamic equation is unchanged, therefore the predictions are the same

$$\begin{aligned}
\hat{\boldsymbol{x}}_{k+1|k} &= \boldsymbol{A}_k\hat{\boldsymbol{x}}_k + \boldsymbol{B}_k\boldsymbol{u}_k \\
\boldsymbol{P}_{k+1|k} &= \boldsymbol{A}_k\boldsymbol{P}_k\boldsymbol{A}_k^T + \boldsymbol{V}_k
\end{aligned}$$

- state correction: due to the sensor noise, the output value is no more certain; we only know that $\boldsymbol{y}_{k+1}$ is drawn from a gaussian distribution with mean value $\boldsymbol{C}_{k+1}\,\boldsymbol{x}_{k+1}$ and covariance matrix $\boldsymbol{W}_{k+1}$

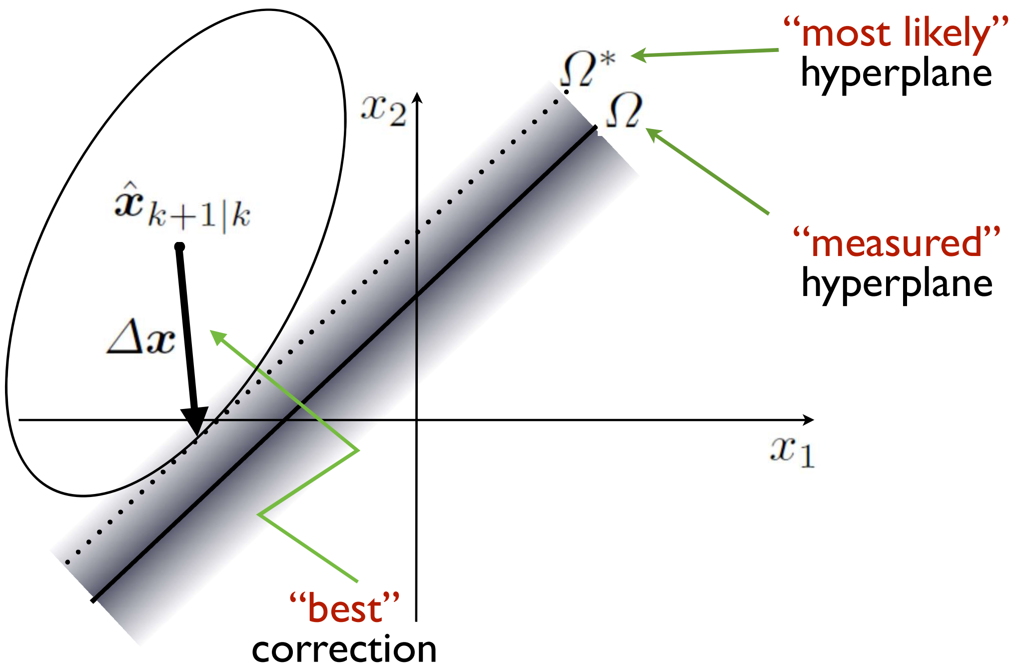
- first we compute the most likely output value $\boldsymbol{y}^*_{k+1}$ given the predictions and the measured output $\boldsymbol{y}_{k+1}$

- then compute the associated most likely hyperplane

$$\Omega^* = \{\boldsymbol{x} \,:\, \boldsymbol{C}_{k+1}\boldsymbol{x} = \boldsymbol{y}^*_{k+1}\}$$

- finally compute the correction $\Delta\boldsymbol{x}$ as before but using Ω^* in place of Ω

- geometric interpretation



the "best" correction is still the closest to $\hat{\boldsymbol{x}}_{k+1|k}$ according to $\boldsymbol{P}_{k+1|k}$, but now it lies on Ω^*

- the resulting Kalman Filter (KF) is

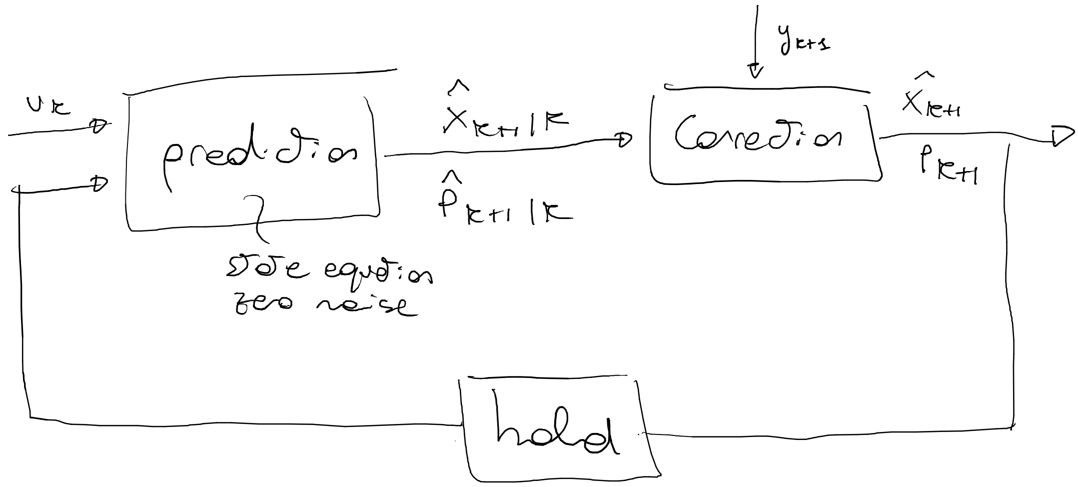
$$
\begin{aligned}
\hat{\boldsymbol{x}}_{k+1|k} &= \boldsymbol{A}_k \hat{\boldsymbol{x}}_k + \boldsymbol{B}_k \boldsymbol{u}_k \\
\boldsymbol{P}_{k+1|k} &= \boldsymbol{A}_k \boldsymbol{P}_k \boldsymbol{A}_k^T + \boldsymbol{V}_k \\
\hat{\boldsymbol{x}}_{k+1} &= \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{R}_{k+1} \boldsymbol{\nu}_{k+1} \\
\boldsymbol{P}_{k+1} &= \boldsymbol{P}_{k+1|k} - \boldsymbol{R}_{k+1} \boldsymbol{C}_{k+1} \boldsymbol{P}_{k+1|k}
\end{aligned}
$$

with the Kalman gain matrix

$$
\boldsymbol{R}_{k+1} = \boldsymbol{P}_{k+1|k} \boldsymbol{C}_{k+1}^T (\boldsymbol{C}_{k+1} \boldsymbol{P}_{k+1|k} \boldsymbol{C}_{k+1}^T + \boldsymbol{W}_{k+1})^{-1}
$$

- matrix $\boldsymbol{R}$ weighs the accuracy of the prediction vs. that of the measurements
  - $\boldsymbol{R}$ "large": measurements are more reliable
  - $\boldsymbol{R}$ "small": prediction is more reliable

KF



- the KF provides an optimal estimate in the sense that $E(\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1})$ is minimized for each k

- the KF is also correct, i.e., it provides mean value and covariance of the posterior gaussian distribution

- if the noises have non-gaussian distributions, the KF is still the best linear estimator but there might exist more accurate nonlinear filters

- if the process is observable, the estimate produced by the KF converges, in the sense that $E(\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1})$ is bounded for all k

Extended Kalman Filter

- consider a nonlinear discrete-time system with noise

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{v}_k$$
$$\boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{x}_k) + \boldsymbol{w}_k$$

  where $\boldsymbol{f}_k$ and $\boldsymbol{h}_k$ are continuously differentiable for each k

- one simple way to build a filter is to linearize the system dynamic equations around the current estimate and then apply the KF equations to the resulting linear approximation

- the resulting Extended Kalman Filter (EKF) is

$$
\begin{aligned}
\hat{\boldsymbol{x}}_{k+1|k} &= \boldsymbol{f}_k(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k)\\
\boldsymbol{P}_{k+1|k} &= \boldsymbol{F}_k \boldsymbol{P}_k \boldsymbol{F}_k^T + \boldsymbol{V}_k\\
\hat{\boldsymbol{x}}_{k+1} &= \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{R}_{k+1} \boldsymbol{\nu}_{k+1}\\
\boldsymbol{P}_{k+1} &= \boldsymbol{P}_{k+1|k} - \boldsymbol{R}_{k+1} \boldsymbol{H}_{k+1} \boldsymbol{P}_{k+1|k}
\end{aligned}
$$

with

$$
\boldsymbol{F}_k = \left.\frac{\partial \boldsymbol{f}_k}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_k} \qquad \boldsymbol{H}_{k+1} = \left.\frac{\partial \boldsymbol{h}_{k+1}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k+1|k}}
$$

and the gain matrix

$$
\boldsymbol{R}_{k+1} = \boldsymbol{P}_{k+1|k} \boldsymbol{H}_{k+1}^T (\boldsymbol{H}_{k+1} \boldsymbol{P}_{k+1|k} \boldsymbol{H}_{k+1}^T + \boldsymbol{W}_{k+1})^{-1}
$$

Autonomous and Mobile Robotics

Prof. Giuseppe Oriolo

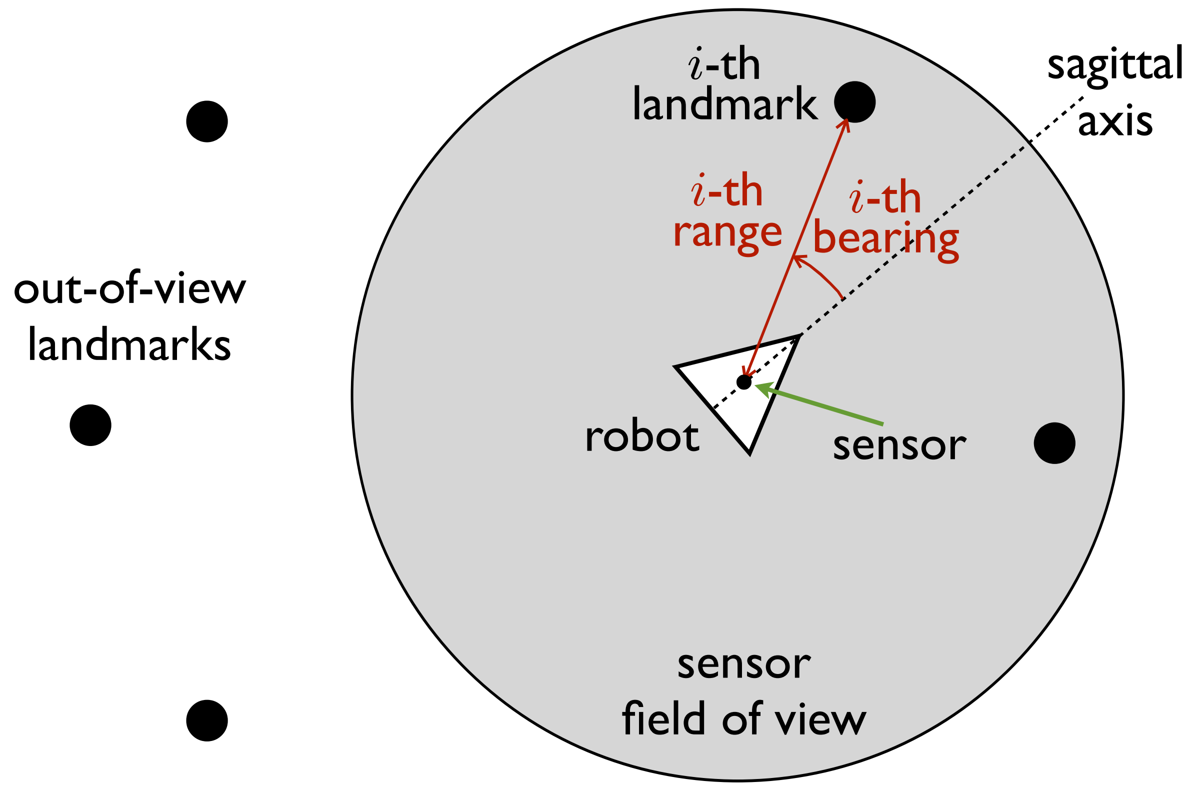
Localization 3

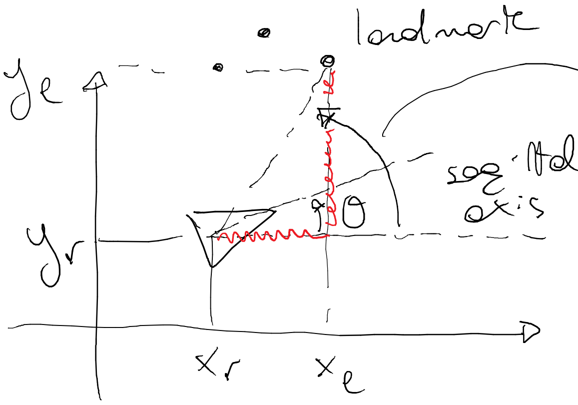
Landmark-Based and SLAM

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, depth cameras or RFID sensors





θ = orientation

ATAN2 $(y_e - y_r,\ x_e - x_r)$

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

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

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

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

where

$$\boldsymbol{h}_i(\boldsymbol{q}_k, a(i)) = \begin{pmatrix} \overbrace{\sqrt{(x_k - x_{l,a(i)})^2 + (y_k - y_{l,a(i)})^2}}^{i\text{-th landmark range}} \\ \underbrace{\operatorname{atan2}(y_{l,a(i)} - y_k, x_{l,a(i)} - x_k) - \theta_k}_{i\text{-th landmark bearing}} \end{pmatrix}$$

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

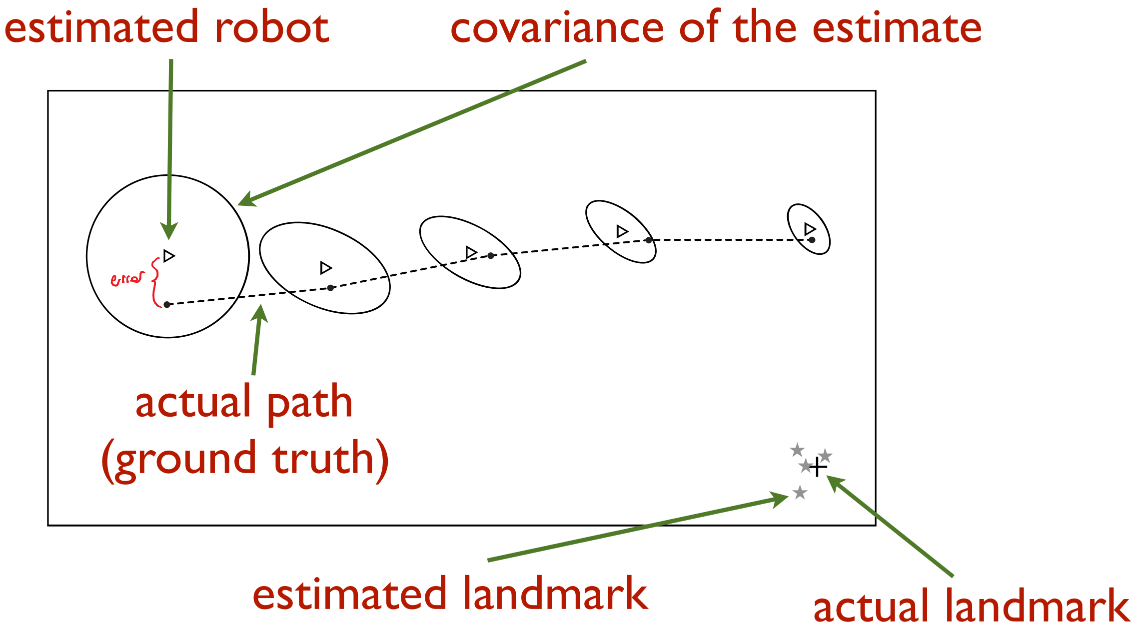
$$
\boldsymbol{H}_{k+1} = \begin{pmatrix} \left.\dfrac{\partial \boldsymbol{h}_1}{\partial \boldsymbol{q}_k}\right|_{\boldsymbol{q}_k=\hat{\boldsymbol{q}}_{k+1|k}} \\ \vdots \\ \left.\dfrac{\partial \boldsymbol{h}_{L_k}}{\partial \boldsymbol{q}_k}\right|_{\boldsymbol{q}_k=\hat{\boldsymbol{q}}_{k+1|k}}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

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



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

$$\boldsymbol{S}_{ij} = \underbrace{\boldsymbol{H}_i(k+1,j)\boldsymbol{P}_{k+1|k}\boldsymbol{H}_i(k+1,j)^T}_{\text{measurement uncertainty due to prediction uncertainty}} + \underbrace{\boldsymbol{W}_{i,k+1}}_{\text{measurement uncertainty due to sensor noise}}$$

- to determine the association function, let

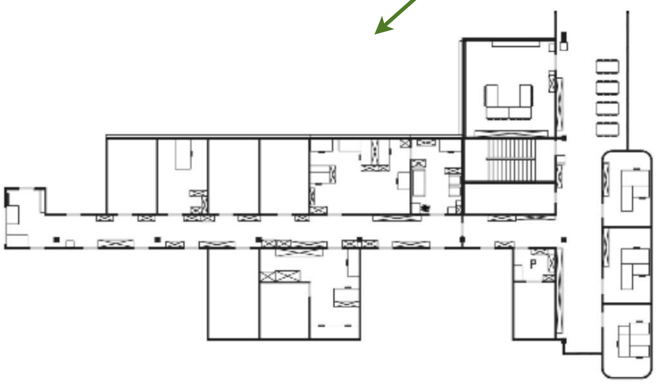
$$\chi_{ij} = \boldsymbol{\nu}_{ij}^T \boldsymbol{S}_{ij}^{-1} \boldsymbol{\nu}_{ij}$$

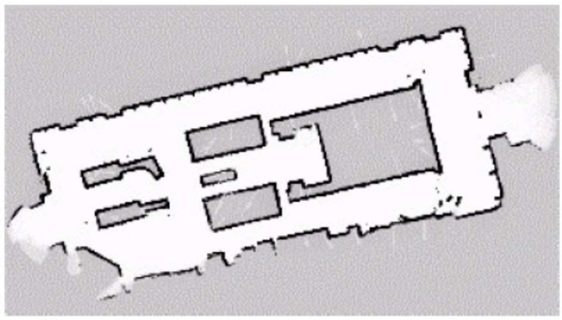
and let $a(i) = j$, where j minimizes χ_{ij}

EKF localization on a map

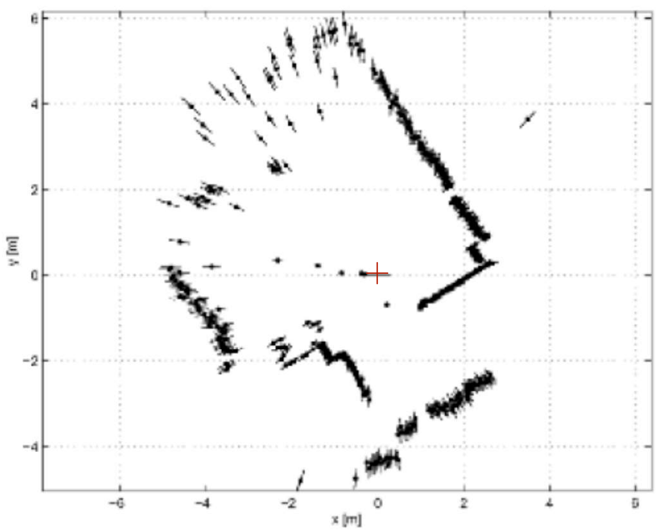
- assume that a metric map $\mathcal{M}$ of the environment is known to the robot

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





- 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)



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

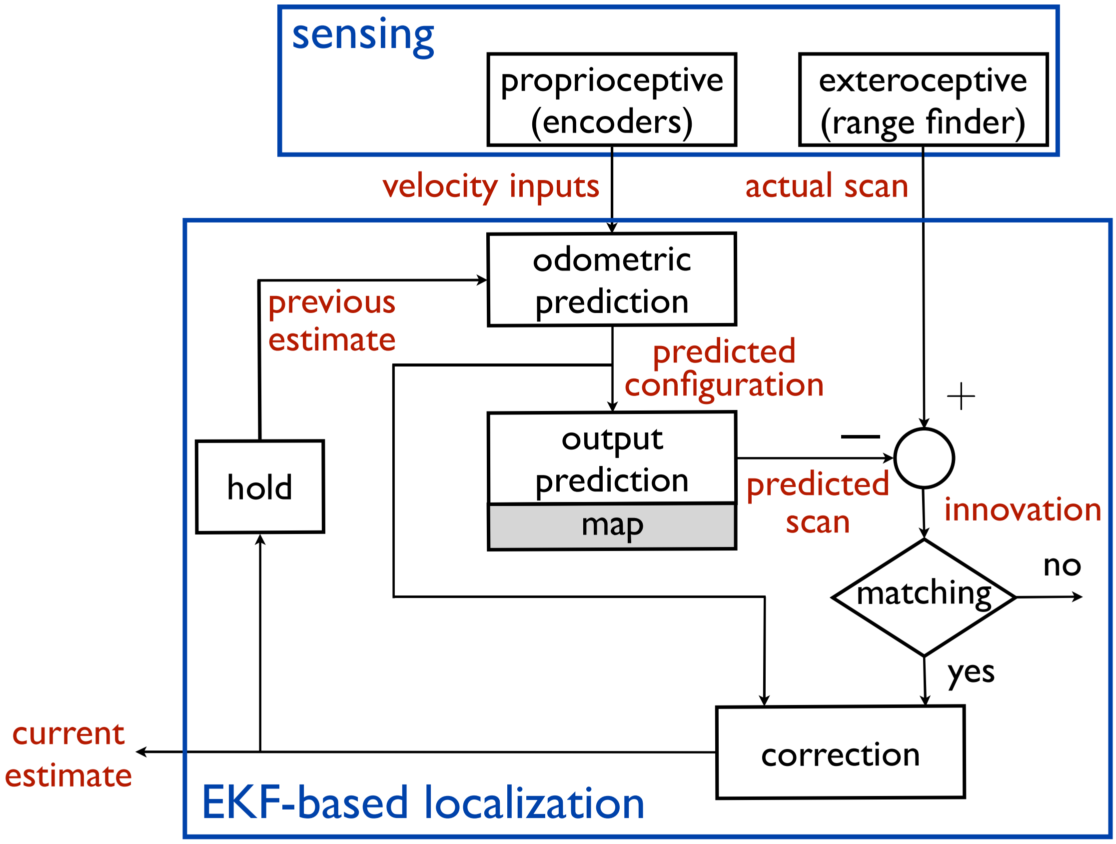
- the innovation is then computed as the difference between the actual scan and the predicted scan

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

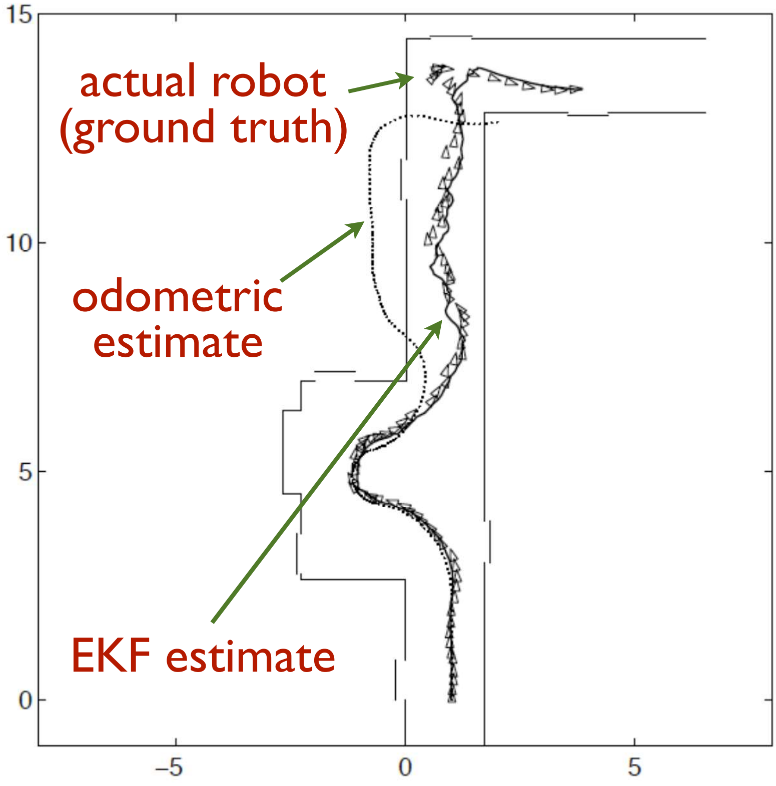
where $\boldsymbol{h}(\;)$ computes the predicted scan by placing the robot at a configuration in the map

- 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



- 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

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

- since the landmarks are fixed, the discrete-time model of the robot+landmarks system is

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

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

- this is clearly a linear model of the form

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

and the covariance of the process noise $\boldsymbol{v}_k$ is

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

where $u_{x,k}$,$u_{y,k}$ are the robot velocity inputs and $\boldsymbol{v}_k = (v_{1,k}\ v_{2,k})^T$ is a white gaussian noise with zero mean and covariance matrix $\boldsymbol{V}_{xy,k}$

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

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

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

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

$(2i+1)$-th column

- stack all measurements to create the output vector

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

where

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

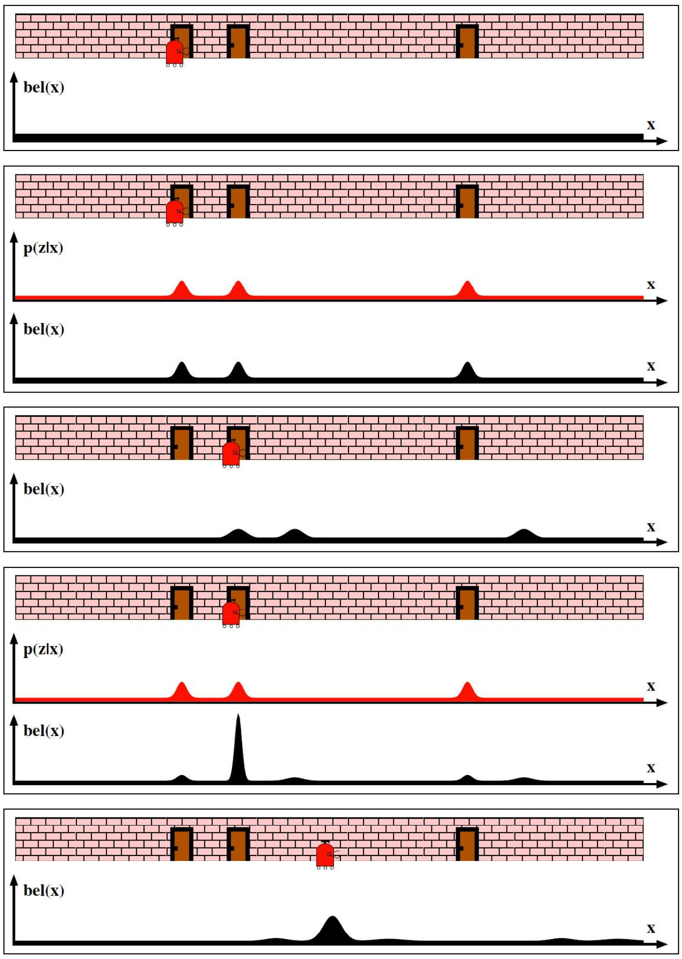
and the covariance of the measurement noise is

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

- 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