UNIVERSITÀ di VERONA

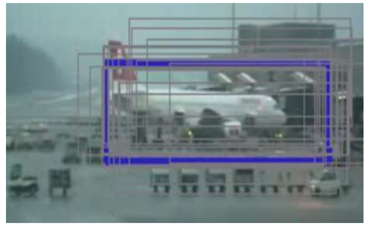
Dipartimento di INFORMATICA

Laurea magistrale in ingegneria e scienze informatiche

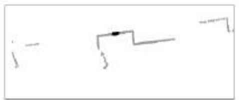
Corso di Robotica
Parte di Laboratorio

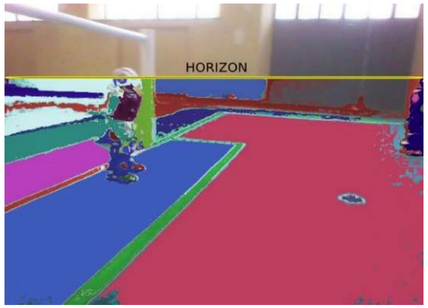
Localizzazione
Kalman Filter

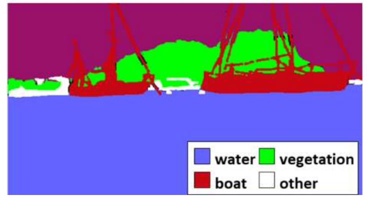
Docente:
Domenico Daniele Bloisi



Dicembre 2017







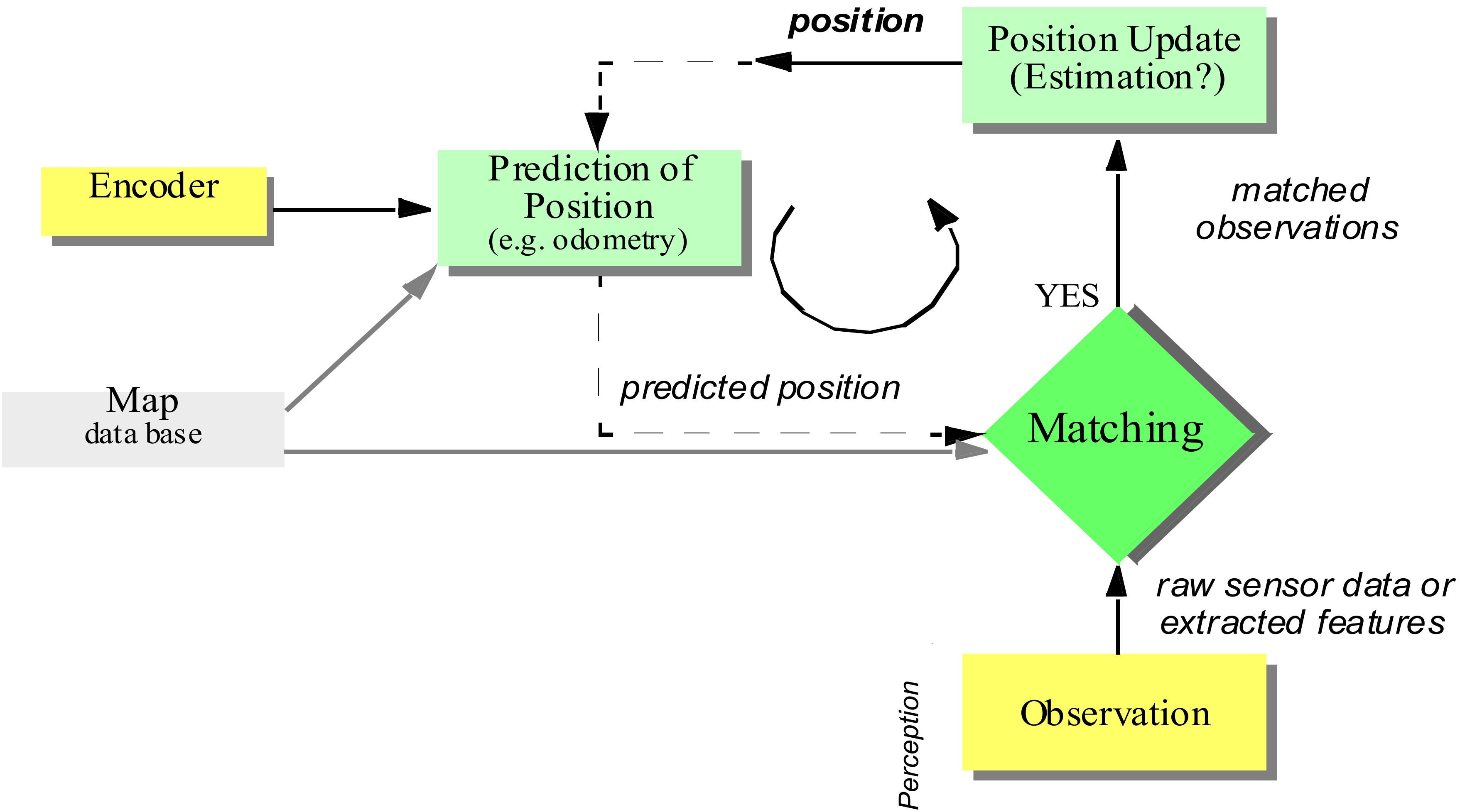
References and credits

Slides derived/borrowed from:

Introduction to Autonomous Mobile Robots
R. Siegwart and I. Nourbakhsh

Autonomous Mobile Robots
Roland Siegwart, Margarita Chli, Martin Rufli

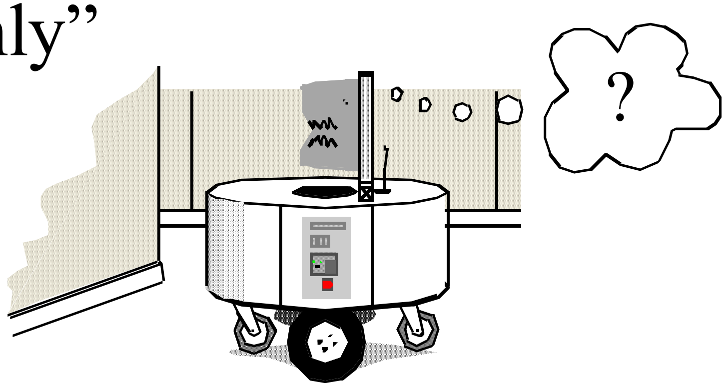
Localization, Where am I?



Localization, definitions

- **Global localization**
  The robot is not told its initial position
  → Its position must be estimated from scratch

- **Position Tracking**
  A robot knows its initial position and "only" has to accommodate small errors in its odometry as it moves

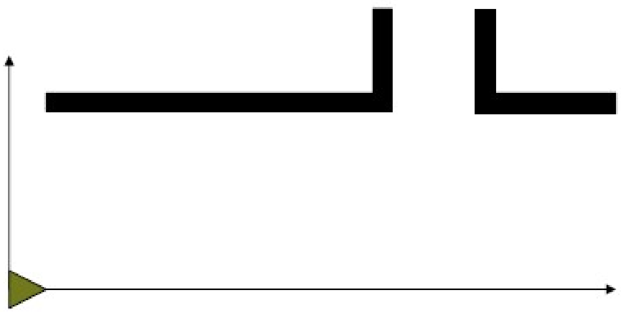


How to localize?

- Localization based on **external sensors, beacons or landmarks**
- **Odometry**
- **Map** based Localization
  without external sensors or artificial landmarks, just use robot onboard sensors

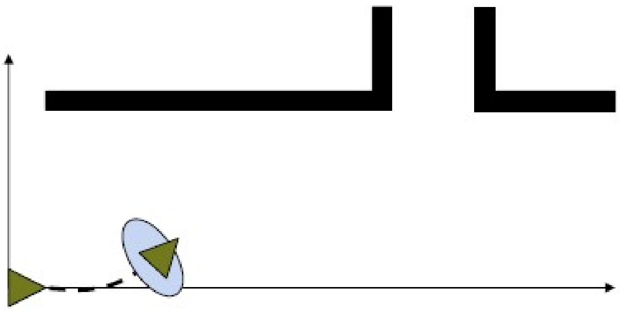
Example: Map based localization

- Consider a mobile robot moving in a known environment



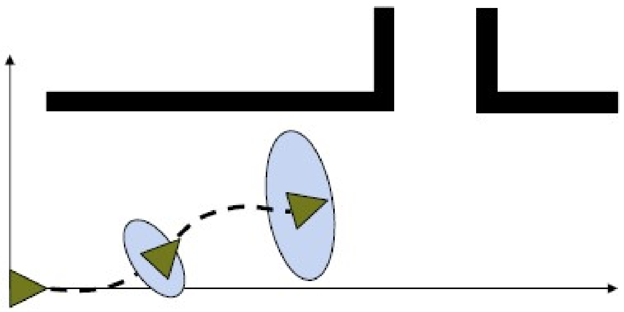
Example: Map based localization

- Consider a mobile robot moving in a known environment
- As it starts to move, say from a precisely known location, it can keep track of its motion using odometry



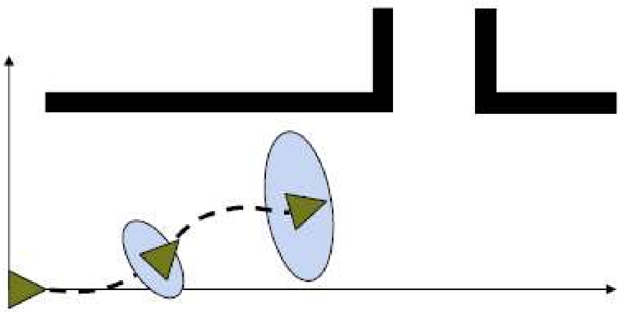
Example: Map based localization

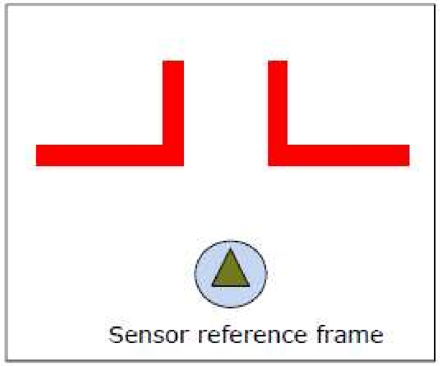
- Consider a mobile robot moving in a known environment
- As it starts to move, say from a precisely known location, it can keep track of its motion using odometry



Example: Map based localization

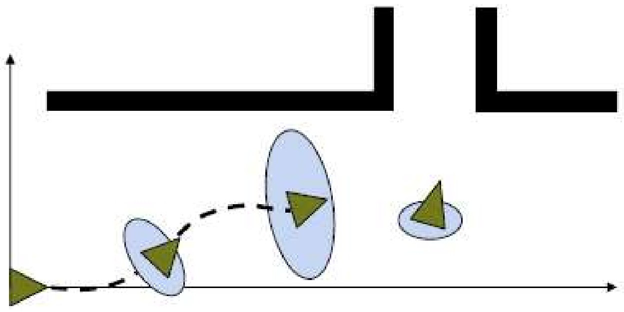
- Consider a mobile robot moving in a known environment
- As it starts to move, say from a precisely known location, it can keep track of its motion using odometry

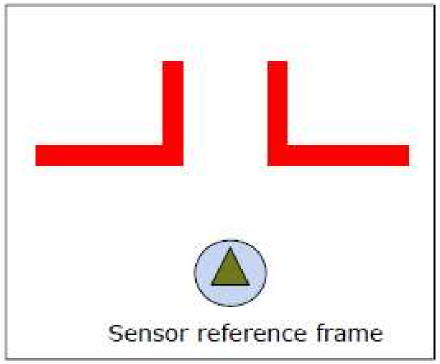




Example: Map based localization

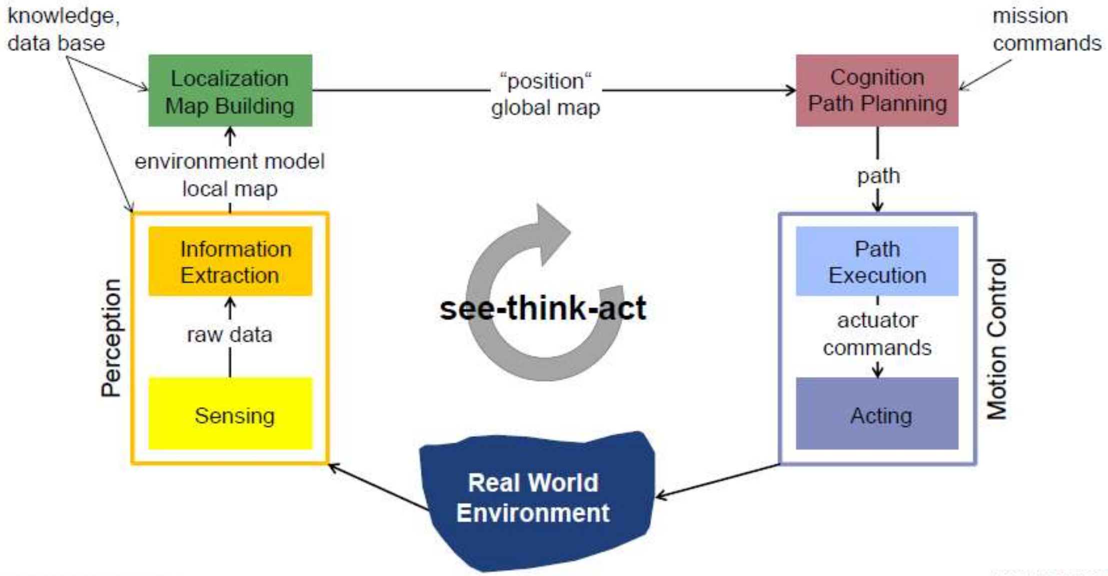
- Consider a mobile robot moving in a known environment
- As it starts to move, say from a precisely known location, it can keep track of its motion using odometry
- The robot makes an observation and updates its position and uncertainty





Challenges of Localization

- Knowing the absolute position (e.g., GPS) is not sufficient
- Localization in human-scale in relation with environment
- Planning in the *Cognition* step requires more than only position as input



Challenges of Localization

- Perception and motion plays an important role
  - *Sensor noise*
  - *Sensor aliasing*
  - *Effector noise*
  - *Odometric position estimation*

Sensor Noise

- Sensor noise in mainly influenced by environment
  e.g., surface, illumination …
- or by the measurement principle itself
  e.g., interference between ultrasonic sensors
- Sensor noise drastically reduces the useful information of sensor readings.
  The solution is:
  - *to take multiple reading into account*
  - *employ temporal and/or multi-sensor fusion*

Sensor Aliasing

- In robots, non-uniqueness of sensors readings is the norm

- Even with multiple sensors, there is a **many-to-one** mapping from environmental states to robot's perceptual inputs

- Therefore the amount of information perceived by the sensors is generally insufficient to identify the robot's position from a single reading
  - *Robot's localization is usually based on a series of readings*
  - *Sufficient information is recovered by the robot over time*

Example of Sensor Aliasing in Humans

http://www.verona.net/it/monumenti/giardino_giusti.html

Effector Noise: Odometry, Dead Reckoning

- Odometry and dead reckoning:
  Position update is based on proprioceptive sensors
  - *Odometry: wheel sensors only*
  - *Dead reckoning: also heading sensors*
- The movement of the robot, sensed with wheel encoders and/or heading sensors is integrated to the position.
  - *Pros: Straight forward, easy*
  - *Cons: Errors are integrated*
- Using additional heading sensors (e.g., gyroscope) might help to reduce the cumulated errors, but the main problems remain the same

Odometry: Error sources

deterministic (systematic) ⟷ non-deterministic (random)

- *deterministic errors can be eliminated by proper calibration of the system.*
- *non-deterministic errors have to be described by error models and will always leading to uncertain position estimate.*

- Major Error Sources:
  - *Limited resolution during integration (time increments, measurement resolution …)*
  - *Misalignment of the wheels (deterministic)*
  - *Unequal wheel diameter (deterministic)*
  - *Variation in the contact point of the wheel*
  - *Unequal floor contact (slipping, not planar …)*
  - …

Odometry: Classification of Integration Errors

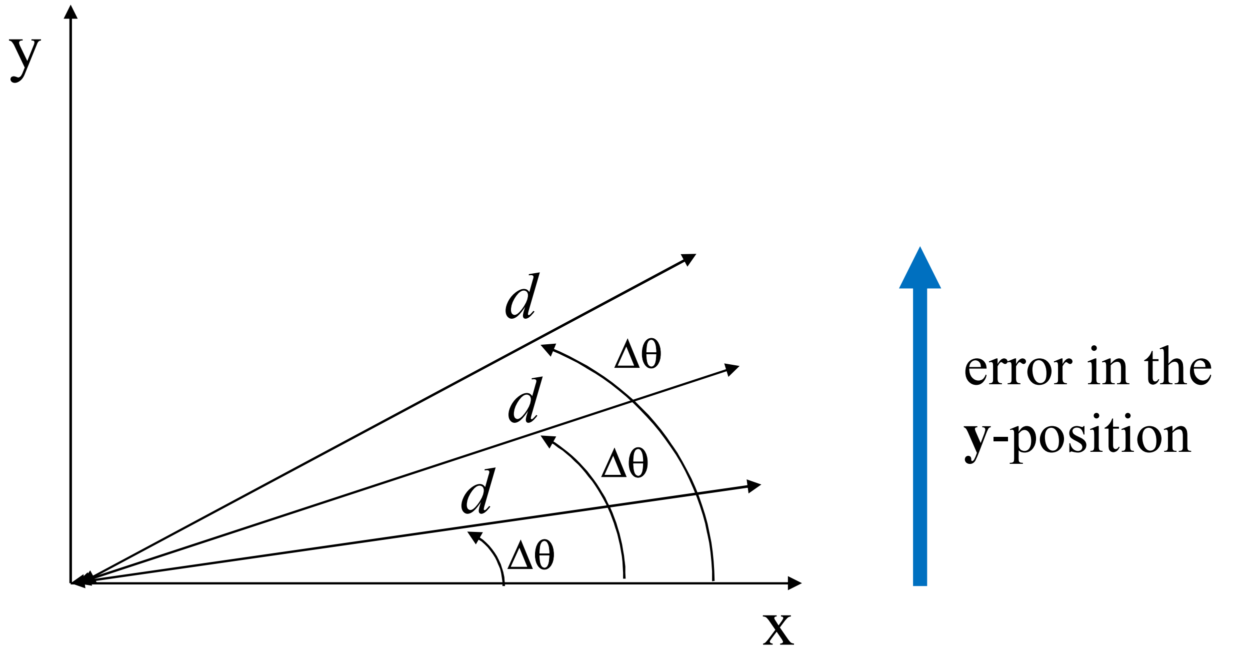
- Range error: integrated path length (distance) of the robots movement
  - *sum of the wheel movements*
- Turn error: similar to range error, but for turns
  - *difference of the wheel motions*
- Drift error: difference in the error of the wheels leads to an error in the robots angular orientation

Over long periods of time, turn and drift errors
far outweigh range errors!

Odometry: Classification of Integration Errors

Consider moving forward on a straight line along the **x** axis.

The error in the **y**-position introduced by a move of **d** meters will have a component of **dsinΔθ**, which can be quite large as the angular error Δθ grows

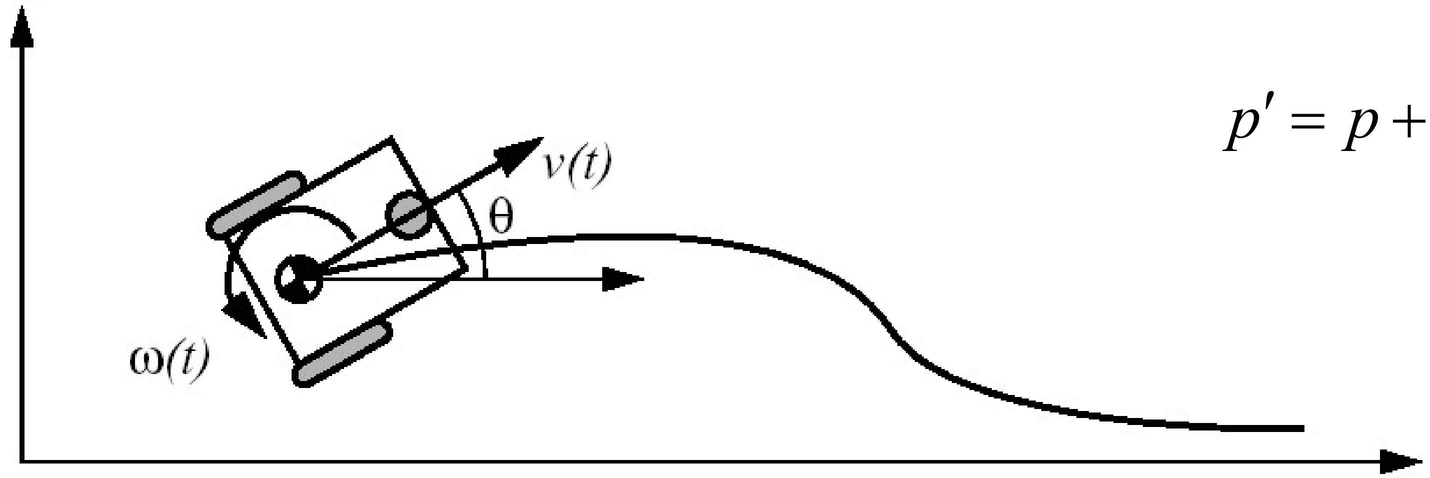


Odometry: The Differential Drive Robot

Robot Pose $p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$

For a differential-drive robot the position can be estimated starting from a known position by integrating the movement (summing the incremental travel distances)

$$p' = p + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix}$$



Kinematics

For a discrete system with a fixed sampling interval Δt, the incremental travel distances $(\Delta x; \Delta y; \Delta\theta)$ are

$$\boxed{\begin{aligned} \Delta x &= \Delta s \cos(\theta + \Delta\theta / 2) \\ \Delta y &= \Delta s \sin(\theta + \Delta\theta / 2) \end{aligned}}$$

$$\Delta\theta = \frac{\Delta s_r - \Delta s_l}{b} \qquad \Delta s = \frac{\Delta s_r + \Delta s_l}{2}$$

This term comes from the application of the Instantaneous Center of Rotation

$(\Delta x; \Delta y; \Delta\theta)$ = path traveled in the last sampling interval;

$\Delta s_r; \Delta s_l$ = traveled distances for the right and left wheel respectively;

b = distance between the two wheels of differential-drive robot.

Kinematics

$$p' = p + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix}$$

$$\Downarrow$$

$$p' = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = p + \begin{bmatrix} \Delta s \cos(\theta + \Delta\theta/2) \\ \Delta s \sin(\theta + \Delta\theta/2) \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta s \cos(\theta + \Delta\theta/2) \\ \Delta s \sin(\theta + \Delta\theta/2) \\ \Delta\theta \end{bmatrix}$$

Kinematics

$$p' = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta s \cos(\theta + \Delta\theta/2) \\ \Delta s \sin(\theta + \Delta\theta/2) \\ \Delta\theta \end{bmatrix} \qquad \Delta\theta = \frac{\Delta s_r - \Delta s_l}{b} \qquad \Delta s = \frac{\Delta s_r + \Delta s_l}{2}$$

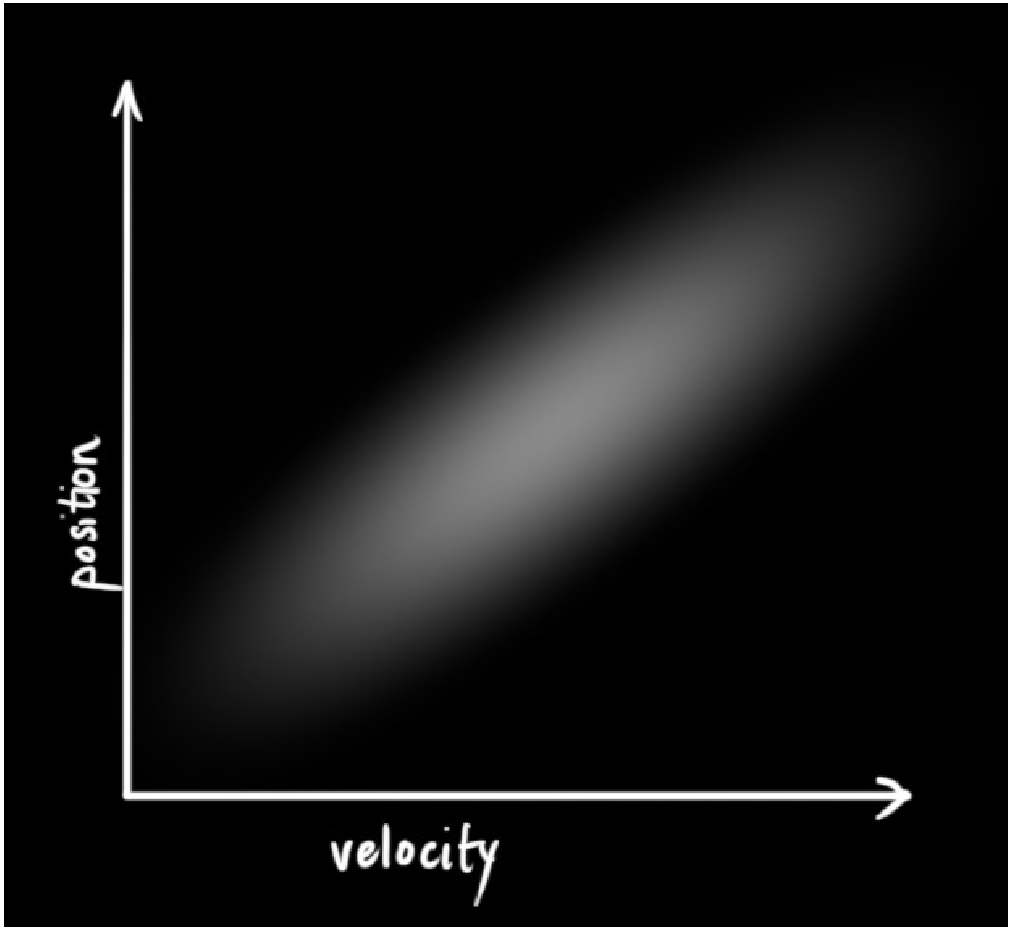
$$\Downarrow$$

$$p' = f(x, y, \theta, \Delta s_r, \Delta s_l) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos\left(\theta + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin\left(\theta + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix}$$

Error Model

covariance is a measure of the joint variability of two random variables

For example, position and velocity



http://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/

Error Model for the Differential Drive Robot

Goal: we want to establish an error model for the integrated position p' to obtain the covariance matrix $\Sigma_{p'}$ of the odometric position estimate

- We assume that at the starting point the initial covariance matrix Σ_p is known

Error Model

For the motion increment we assume the following covariance matrix:

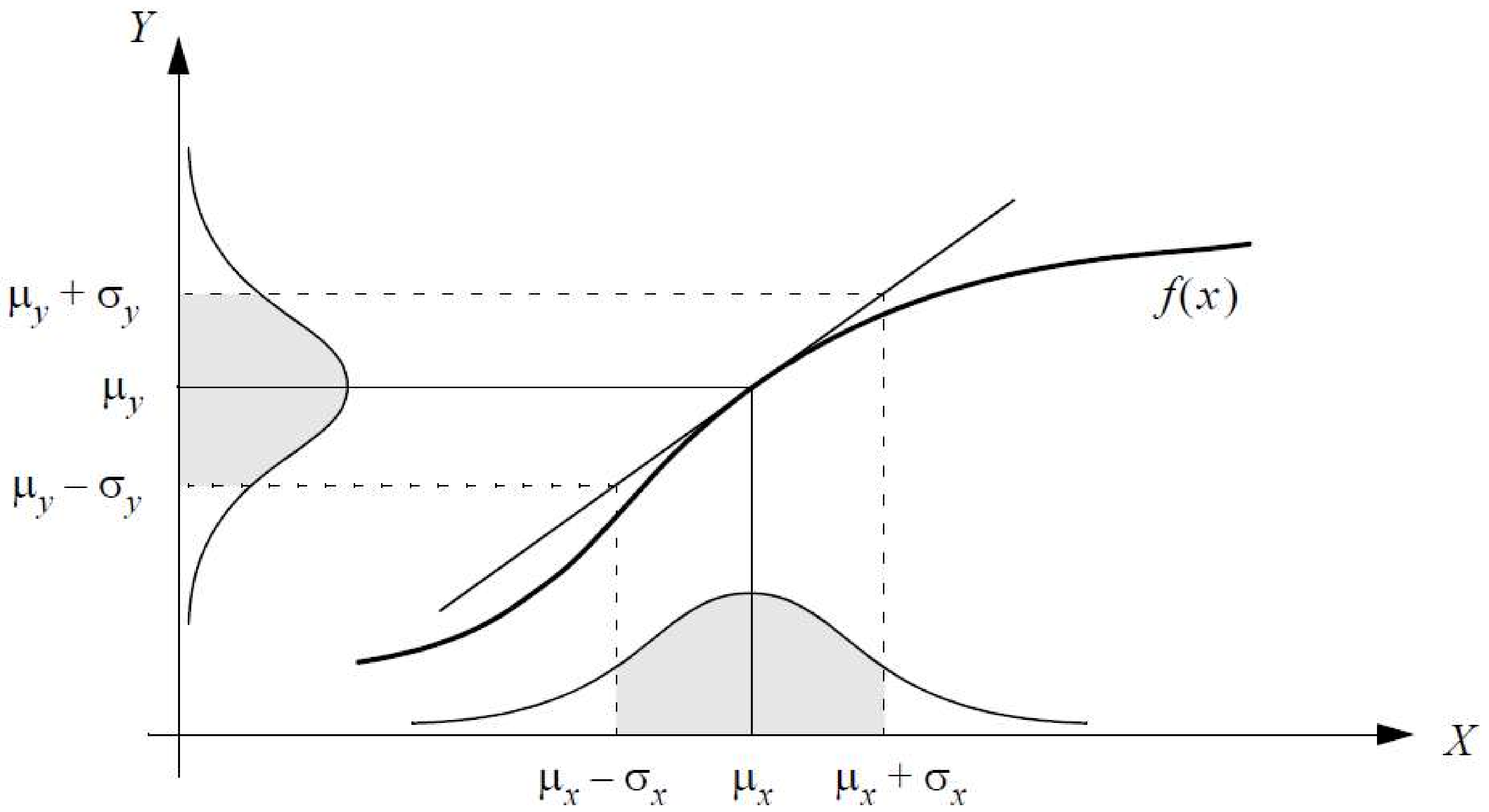
$$\Sigma_\Delta = covar(\Delta s_r, \Delta s_l) = \begin{bmatrix} k_r|\Delta s_r| & 0 \\ 0 & k_l|\Delta s_l| \end{bmatrix}$$

where Δs_r and Δs_l are the distances traveled by each wheel, and k_r, k_l are error constants representing the nondeterministic parameters of the motor drive and the wheel-floor interaction

Assumptions

- The two errors of the individually driven wheels are independent
- The variance of the errors (left and right wheels) are proportional to the absolute value of the traveled distances $(\Delta s_r;\Delta s_l)$

Error Propagation



One-dimensional case of a nonlinear error propagation problem

Error Propagation Law

The output covariance matrix C_Y is given by the error propagation law:

$$C_Y = F_X C_X F_X^T,$$

where

C_X = covariance matrix representing the input uncertainties;

C_Y = covariance matrix representing the propagated uncertainties for the outputs;

F_x is the *Jacobian* matrix defined as

$$F_X = \nabla f = \begin{bmatrix} \frac{\partial f_1}{\partial X_1} & \cdots & \frac{\partial f_1}{\partial X_n} \\ \vdots & \cdots & \vdots \\ \frac{\partial f_m}{\partial X_1} & \cdots & \frac{\partial f_m}{\partial X_n} \end{bmatrix}.$$

Error Propagation

$$p' = f(x, y, \theta, \Delta s_r, \Delta s_l)$$

$$p \qquad\qquad \Delta_{rl} = [\Delta s_r, \Delta s_l]^T$$

$$\Sigma_{p'} = \boxed{\nabla_p f} \cdot \boxed{\Sigma_p} \cdot \boxed{\nabla_p f^T} + \boxed{\nabla_{\Delta_{rl}} f} \cdot \boxed{\Sigma_\Delta} \cdot \boxed{\nabla_{\Delta_{rl}} f^T}$$

$$C_Y : \qquad F_X C_X F_X^T \qquad\qquad F_X C_X F_X^T$$

Error Propagation

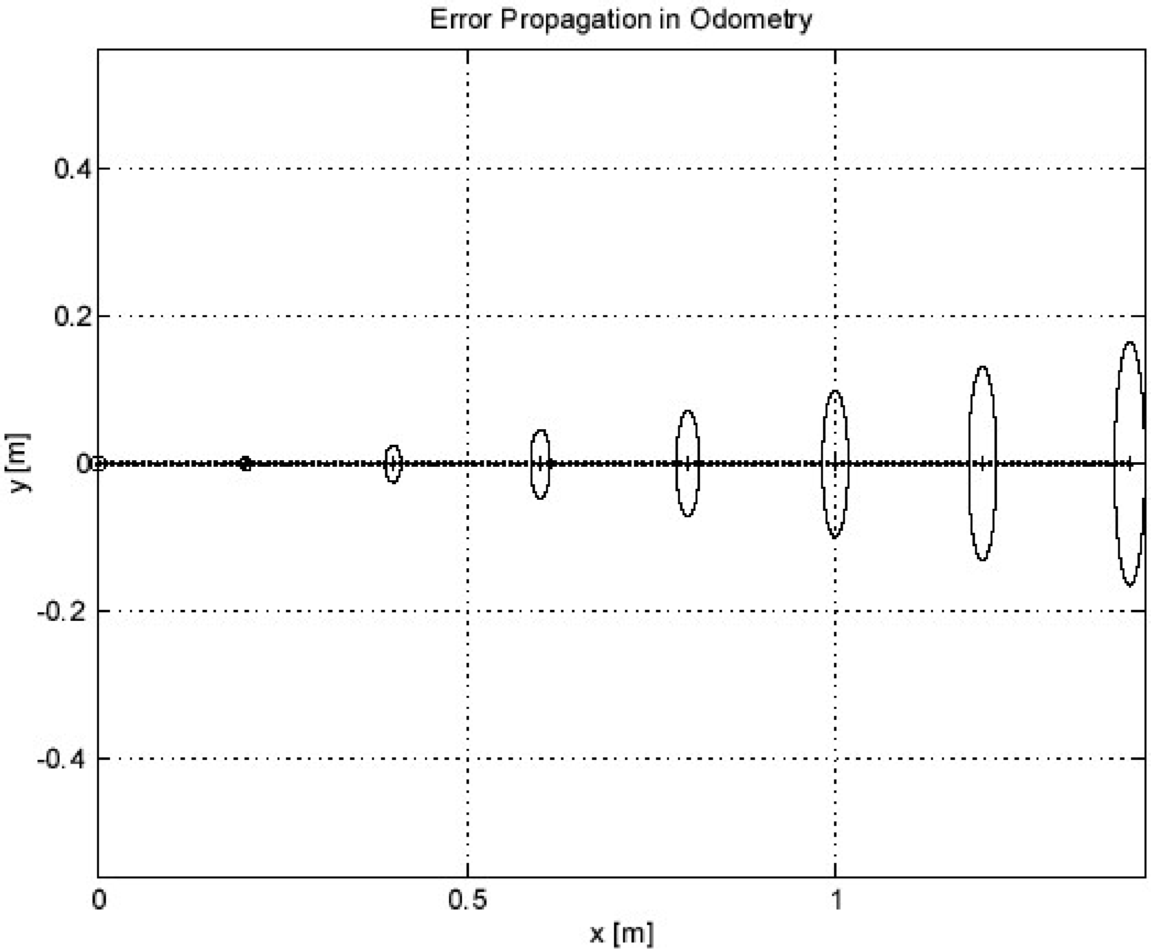
$$\Sigma_{p'} = \nabla_p f \cdot \Sigma_p \cdot \nabla_p f^T + \nabla_{\Delta_{rl}} f \cdot \Sigma_\Delta \cdot \nabla_{\Delta_{rl}} f^T$$

$$F_p = \nabla_p f = \nabla_p(f^T) = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta s \sin(\theta + \Delta\theta/2) \\ 0 & 1 & \Delta s \cos(\theta + \Delta\theta/2) \\ 0 & 0 & 1 \end{bmatrix}$$

$$F_{\Delta_{rl}} = \begin{bmatrix} \frac{1}{2}\cos\left(\theta + \frac{\Delta\theta}{2}\right) - \frac{\Delta s}{2b}\sin\left(\theta + \frac{\Delta\theta}{2}\right) & \frac{1}{2}\cos\left(\theta + \frac{\Delta\theta}{2}\right) + \frac{\Delta s}{2b}\sin\left(\theta + \frac{\Delta\theta}{2}\right) \\ \frac{1}{2}\sin\left(\theta + \frac{\Delta\theta}{2}\right) + \frac{\Delta s}{2b}\cos\left(\theta + \frac{\Delta\theta}{2}\right) & \frac{1}{2}\sin\left(\theta + \frac{\Delta\theta}{2}\right) - \frac{\Delta s}{2b}\cos\left(\theta + \frac{\Delta\theta}{2}\right) \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix}$$

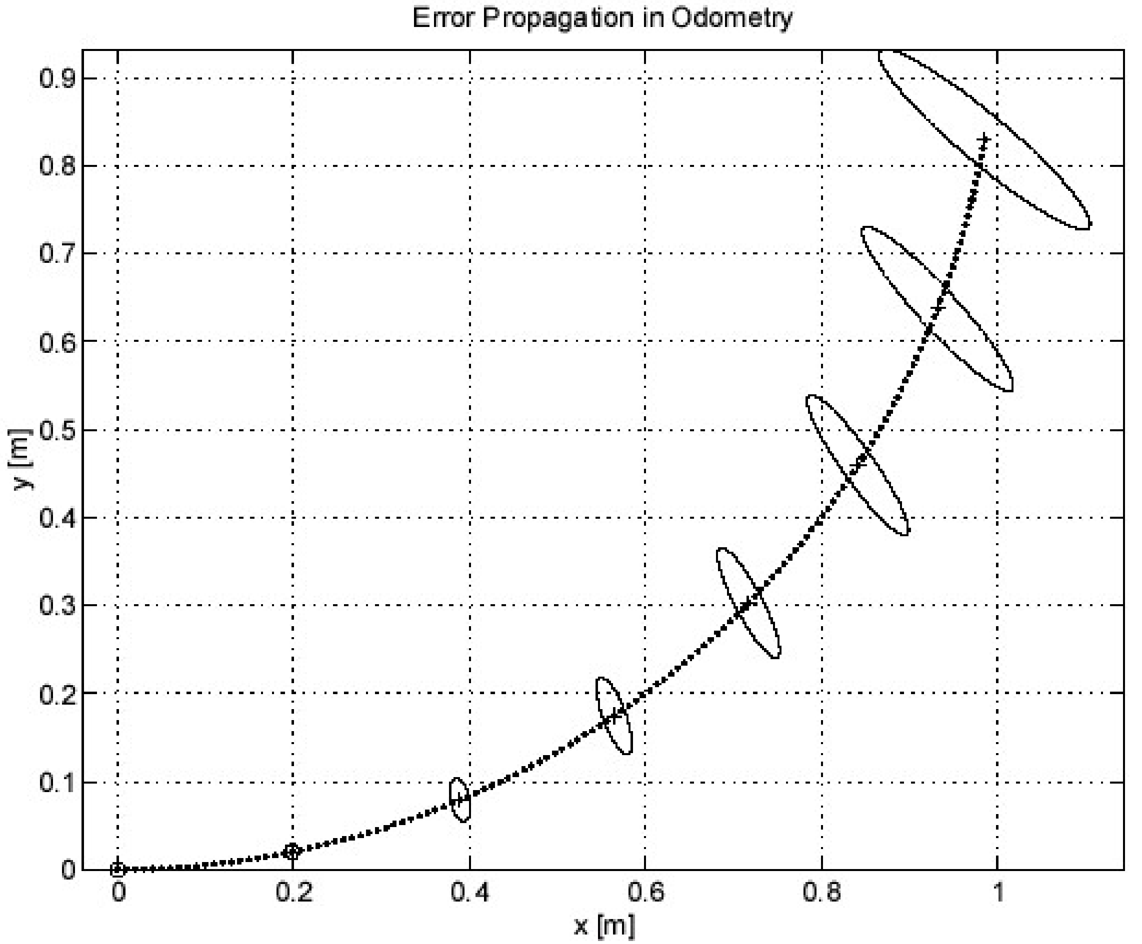
Odometry: Growth of Pose uncertainty for Straight Line Movement

- Note: Errors perpendicular to the direction of movement are growing much faster!

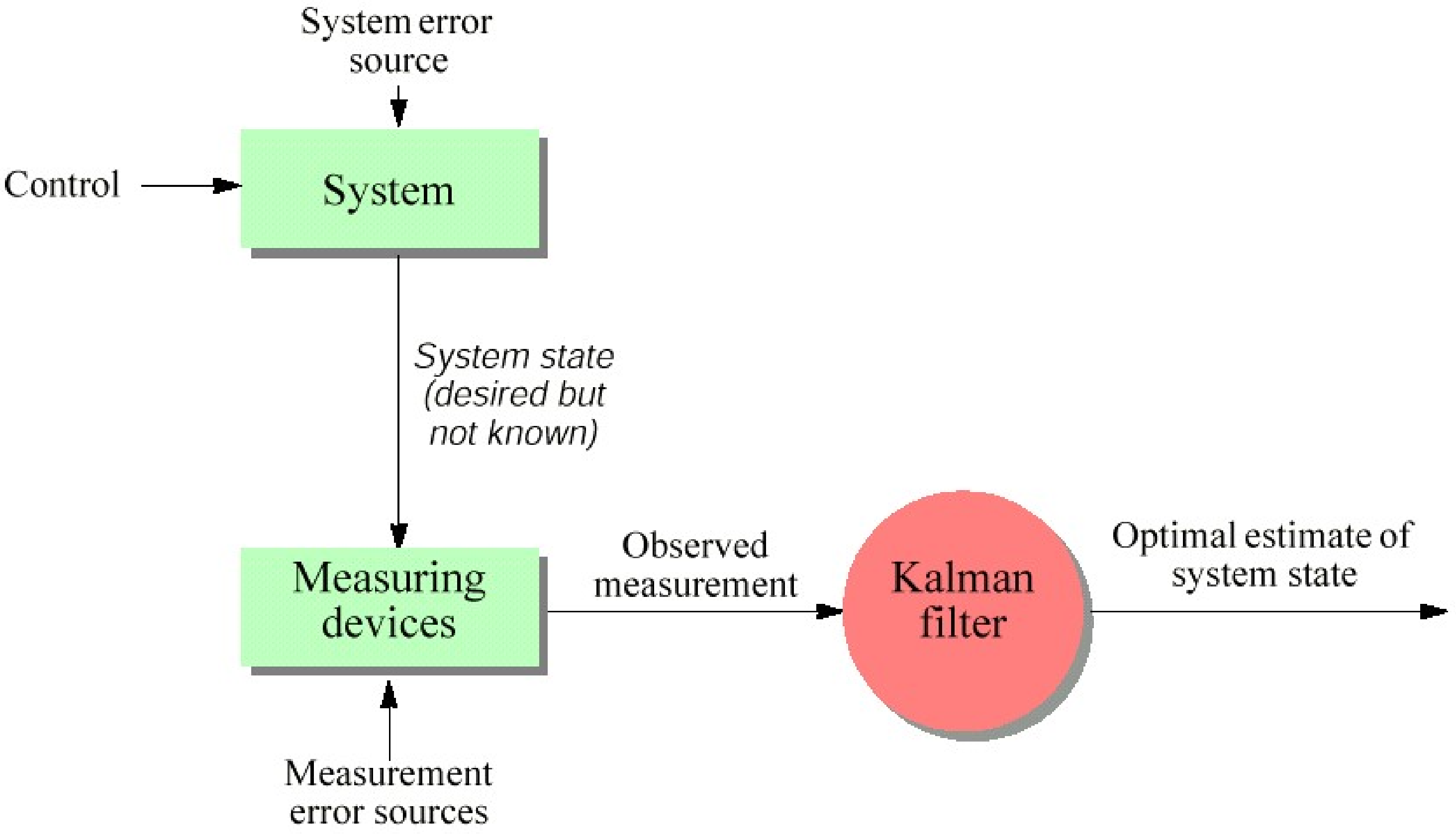


Odometry: Growth of Pose uncertainty for Movement on a Circle

- Note: Errors ellipse does not remain perpendicular to the direction of movement!



Kalman Filter Localization



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

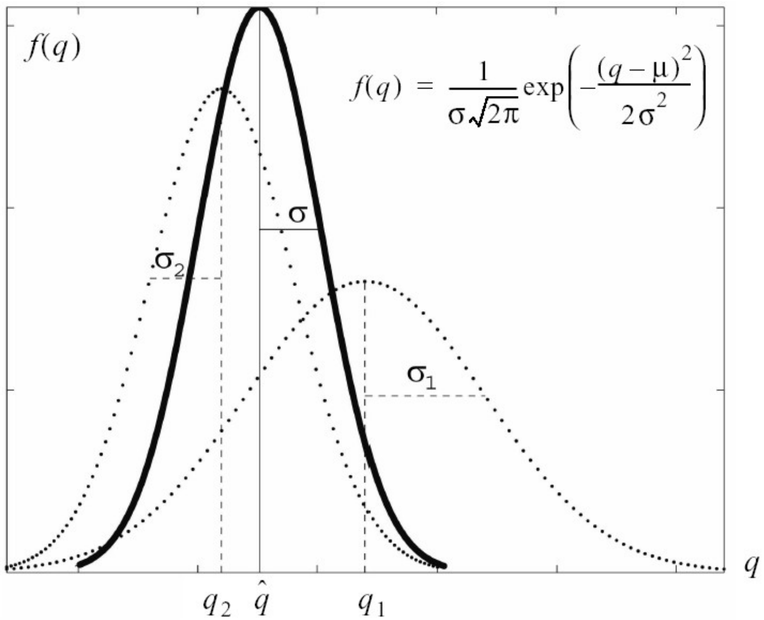
$$S = \sum_{i=1}^{n} w_i (\hat{q} - q_i)^2$$

- 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

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

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

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

the best estimate $\hat{x}_{k+1}$ of the state x_{k+1} at time $k+1$ is equal to the best prediction of the value $\hat{x}_k$ before the new measurement z_{k+1} is taken, plus a correction term of an optimal weighting value times the difference between z_{k+1} and the best prediction $\hat{x}_k$ at time $k+1$

Introduction to Kalman Filter (3)

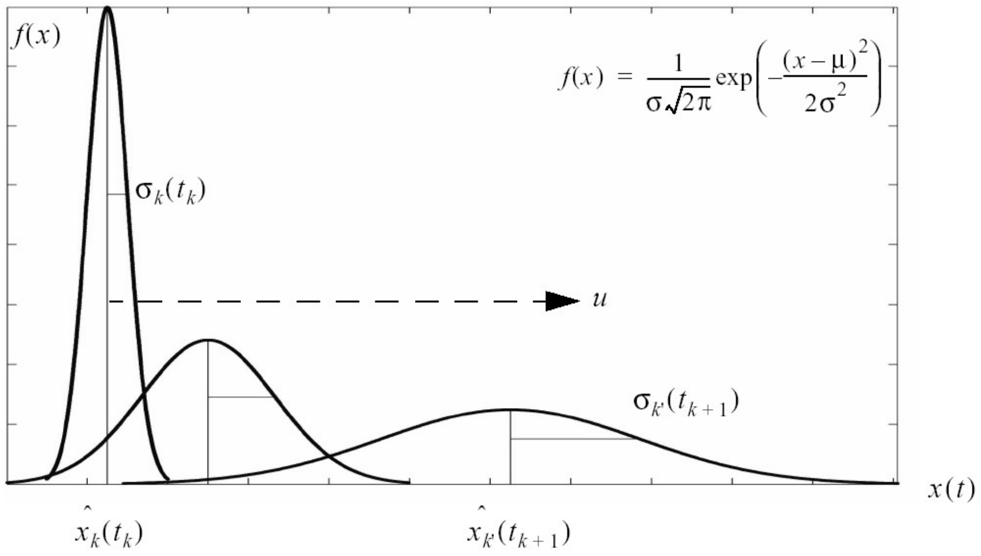
- Dynamic Prediction (robot moving)

$$\frac{dx}{dt} = u + w \qquad \begin{array}{l} u = velocity \\ w = noise \end{array}$$

- Motion

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

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

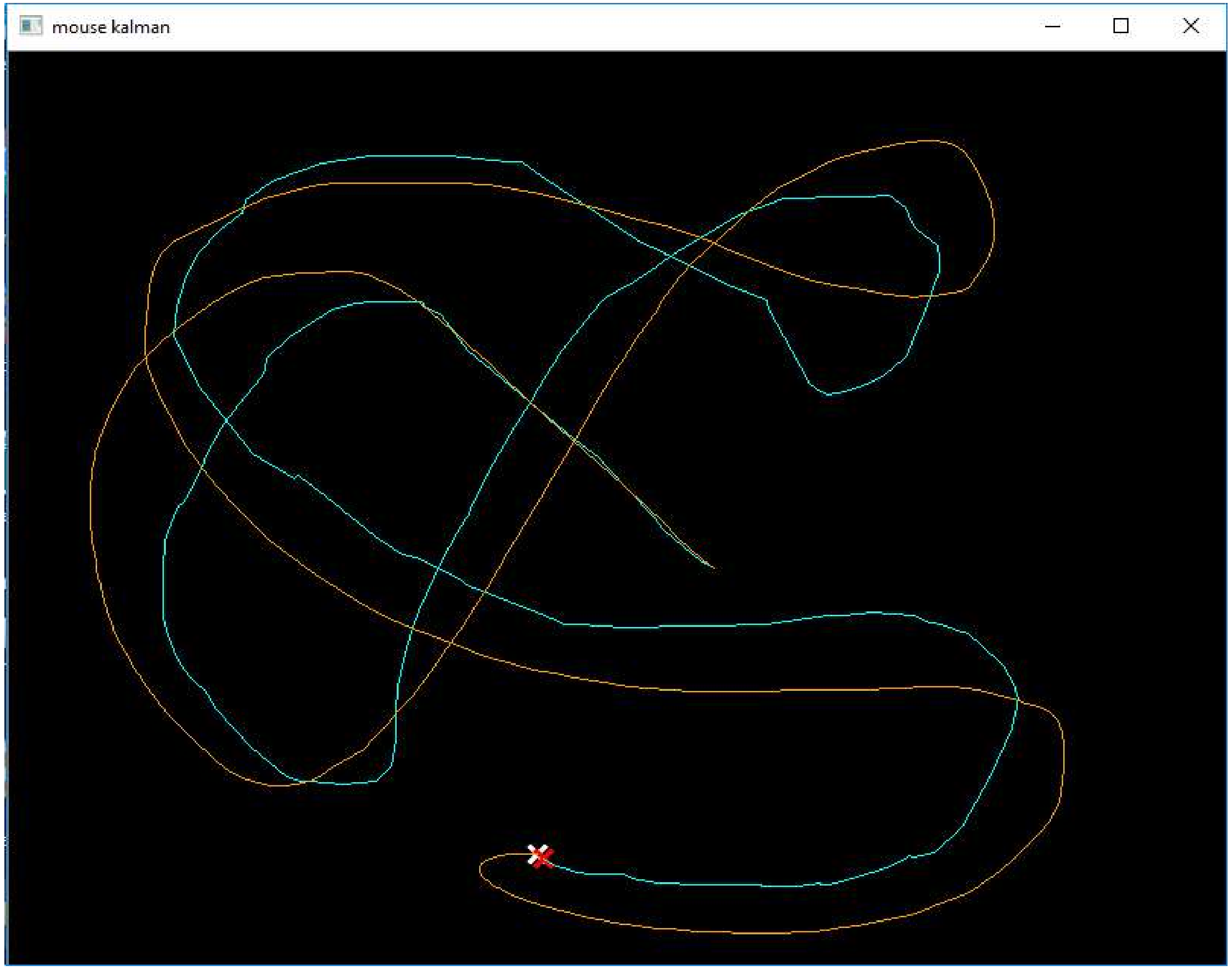


- Combining fusion and dynamic prediction

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

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

Esempio – Mouse Kalman



Quale delle due curve è stata disegnata con il mouse?

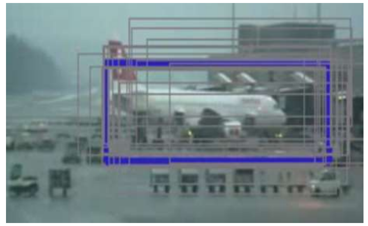
UNIVERSITÀ di VERONA
Dipartimento di INFORMATICA

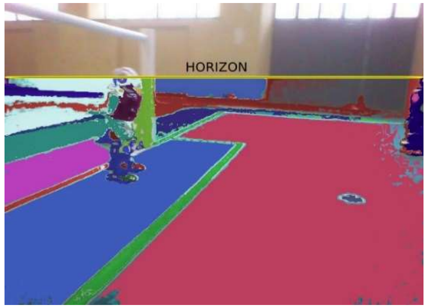
Laurea magistrale in ingegneria e scienze informatiche

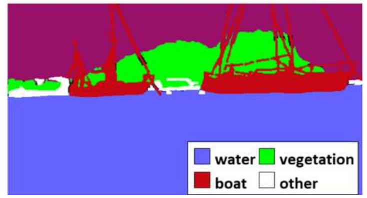
Corso di Robotica
Parte di Laboratorio

Localizzazione
Kalman Filter

Docente:
Domenico Daniele Bloisi







Dicembre 2017