G54ARS
Autonomous Robotic Systems
Lecture 8

Kalman Filter

Last week

- Guest Lecture
  - Dr Barbara Bruno
  - University of Genoa, Italy
- Human Robot Interaction
  - Tactile Sensors
  - Human Activity Recognition
  - Gesture Based Control
  - Human-Robot Cooperation
  - Culture-aware robots for elderly care

Kahoot!

https://kahoot.it/

Post Assignment 'Localisation'

- Where are we in the module?
- What we have looked at:
  - Foundations of Autonomous Robotic Systems
  - Architectures and Behaviours
    - Brooks' Subsumption Architecture
  - Robot Hardware – Sensors and Actuators
  - PID Control
  - Fuzzy Control
  - Ultrasonic Sensor Models
  - Principles of Localisation and Mapping
  - Sensor and Behaviour Fusion
  - Real World Robotics (guest lecture)
  - Ethics in robotics

+ hands-on experience with real-world sensors/actuators and robots

What is missing, i.e. what next?

This week…

- SET – SEM Feedback
- Kalman Filter - Basic concepts
  - filtering overview
  - system model
  - filter operation
- Worked example
  - model example
  - illustrative examples of operation
- Properties of Kalman filters
  - the optimal filter
  - misconceptions and myths

Evaluate

- When using Evaluate, staff are asked to:
  - Ask students to read the questions carefully, giving special attention to the rating scale. The University has now made this more user-friendly/obvious and in line with NSS rating scale.
  - Assure students that their views are confidential and point out that the data is processed automatically.
  - Inform students that you, as teacher, will read their comments.
  - Make clear that while they may make comments, favourable or unfavourable, responses containing personally offensive comments are unacceptable and will be deleted.

Evaluate Survey Details for Students

To take a survey for Autonomous Robotic Systems please go to

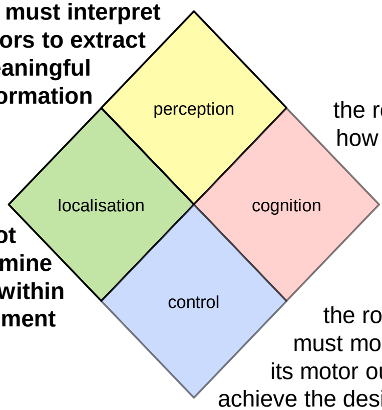
https://bluecastle.nottingham.ac.uk

and log in using your user name and password

Note: Direction of scale

The Kalman Filter: Basic Concepts

Navigation

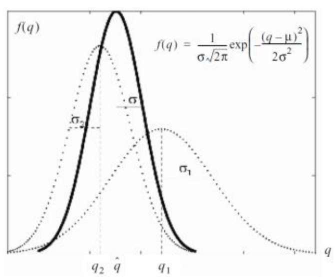


Definition

- A Kalman filter is a recursive filter that estimates the state of a dynamic system from a series of noisy (and possibly incomplete) observations
  - developed by Rudolf Kálmán, c. 1960
- Classic application
  - providing accurate, regularly updated information about the position and velocity of an object, given a sequence of observations about its position, each of which includes some error
    - tracking missiles
    - Apollo moon-landing navigation systems
    - mobile robots

Kalman Filter and Localisation

- The Kalman Filter is an optimal sensor fusion technique.

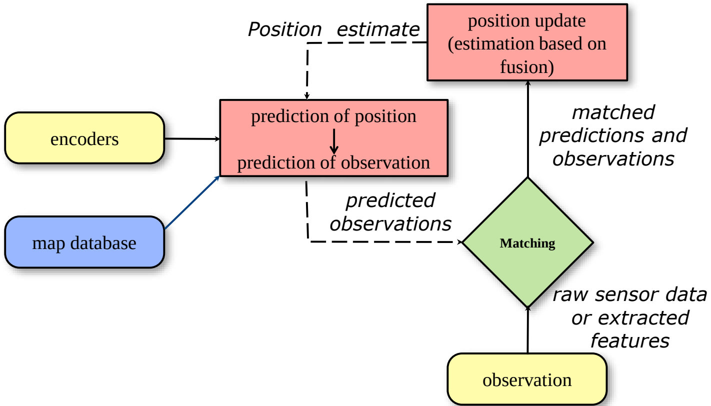


Source: Intro. to Autonomous Mobile Robots, R. Siegwart; I.R. Nourbakhsh (Fig. 5.26)

Kalman Filter and Localisation

- The Kalman Filter is an ‘optimal’ sensor fusion technique.
- In order to apply the Kalman Filter to localisation, the localization problem needs to be expressed as a sensor fusion problem:
  - Proprioceptive Sensors
    - Internal sensing, e.g., wheel encoders
    - E.g., subject to range/turn/drift errors accumulating over time
  - Exteroceptive
    - External sensing, e.g., through ultrasonic sensors
    - Subject to measurement, incl. random noise
- It then enables the incorporation of all information, regardless of precision, to estimate the actual position of the robot.
  - Problem: “precision = ?”

General Schematic for Kalman Filter Localization



Kalman Filter for Localisation in Brief

1. Predict robot position based on encoder input, old position and control input. (Compute variance based on known error of plant and process)
2. Obtain sensor measurements (features) (e.g., using sonars) and transform measurements to sensor frame using a function (h_i).
3. Predict sensor measurements (features) based on predicted robot position and map.
4. Do matching of predicted measurements/features and actually observed features. →Find "good enough" matches to determine which actual measurements/features to use in the next step, others are discarded.
5. Estimate robot position based on position prediction and validated measurements/features. Compute variance.

More formally…

System Model

- State, x

$$x \in \Re^n$$

- Discrete linear process

Relate previous state at k-1 to current state k for position and control signal respectively

$$x_k = \overbrace{A_k}\underbrace{x_{k-1}}_{\text{Previous position}} + \overbrace{B_k}\underbrace{u_k}_{\text{control signal}} + w_k$$

- Process noise

$$w_k \sim N(0, \underbrace{Q_k}_{\text{Process noise covariance matrix}})$$

Measurements

- Measurement, z

$$z \in \Re^m$$

- Measurement of the true state x_k given by

$$\underbrace{z_k}_{\text{Feature}} = \overbrace{H_k}^{\text{Transform measurement to sensor frame}} \underbrace{x_k}_{\text{Measurement}} + v_k$$

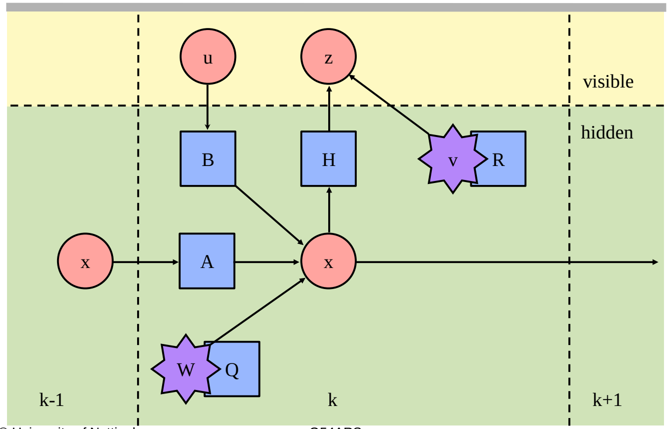
Features can be "actual features" such as doors, chairs, people or a raw sensor value

- Measurement noise

$$v_k \sim N(0, \underbrace{R_k}_{\text{Measurement noise covariance matrix}})$$

 G54ARS

Model Summary



 G54ARS

Break

 G54ARS

Predict

- Project the state ahead (from previous state and control signal)

$$\underbrace{\hat{x}_k^-}_{\text{The "prior estimate", i.e. before correction based on measurement}} = A\hat{x}_{k-1} + Bu_k$$

- Project the error covariance ahead
  - a measure of the estimated accuracy of the state estimate

$$\underbrace{P_k^-}_{\text{The "prior error covariance"}} = AP_{k-1}A^T + Q$$

 G54ARS

Correct

- Compute the Kalman gain
  - It tells us how much to adjust the state given the difference between the measurement and the predicted measurement

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$

i.e.: if R (~measurement noise) is large, the Kalman gain is small

Correct continued

- Update estimate with measurement z_k

$$\hat{x}_k = \hat{x}_k^- + K_k \boxed{(z_k - H\hat{x}_k^-)}$$

the difference between the actual measurement and the predicted measurement is known as the *innovation* or *residual*

- Update the error covariance

$$P_k = (I - K_k H) P_k^-$$

an identity matrix of size n x n (like 1 in the numeric case)

Note: $\hat{x}_k$ and P_k are posterior values.

Operation Summary

Predict (Time Update)

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$$

$$P_k^- = A P_{k-1} A^T + Q$$

Correct (Measurement Update)

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-)$$

$$P_k = (I - K_k H) P_k^-$$

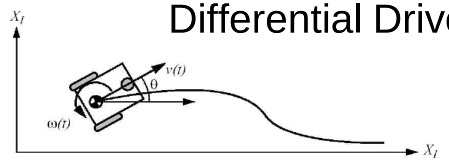
Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

Filter Parameters and Tuning

- The measurement noise covariance, R
  - the size of the error in the sensors
  - can be experimentally determined prior to operation
- The process noise covariance, Q
  - *may* also be determined experimentally, but it is often more difficult to directly observe the process
  - can sometimes be tuned off-line through another Kalman filter, performing *system identification*
- The error covariance, P
  - starting conditions known exactly: initialise with zeros
  - conditions unknown: initialise with large diagonal

Worked Example

Remember the position update for a Differential Drive robot



$$\Delta\theta = \frac{\Delta s_r - \Delta s_l}{b}$$

$$\Delta s = \frac{\Delta s_r + \Delta s_l}{2}$$

where Δs_l and Δs_r are the distances travelled for the left and right wheel respectively and b is the distance between the two wheels of the differential-drive robot

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

Example System

- Mobile robot with pose $x = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$
- Moving with speed s in direction Θ $u = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix}$

$$x_k = \overset{A}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}} \overset{x_{k-1}}{\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}} + \overset{B}{\begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{bmatrix}} \overset{u}{\begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix}} + w_k$$

$$z_k = \overset{H}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + v_k$$

Covariances

$$Q = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix}$$

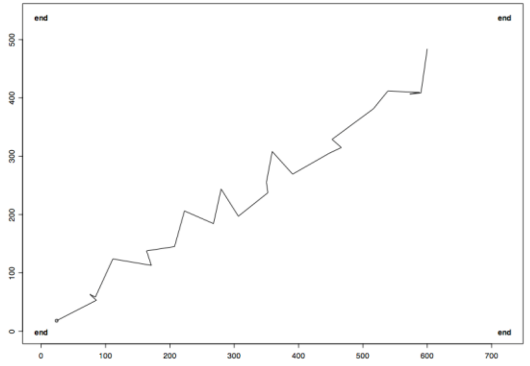
$$R = \begin{bmatrix} \sigma_{x_m}^2 & 0 \\ 0 & \sigma_{y_m}^2 \end{bmatrix}$$

m sensors

$$P_0 = \begin{bmatrix} ? & 0 & 0 \\ 0 & ? & 0 \\ 0 & 0 & ? \end{bmatrix}$$

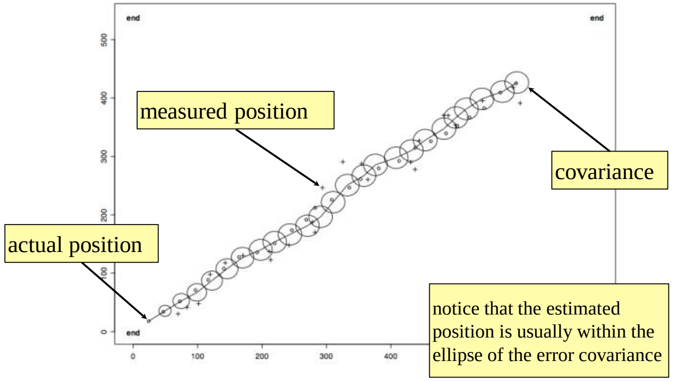
- Terms along diagonal are variances.
- Terms off the diagonal are covariances (not used here)

Unfiltered



$\sigma_x^2 = \sigma_y^2 = 4$ $\sigma_{x_m}^2 = \sigma_{y_m}^2 = 16$

Kalman Filter



Properties of Kalman Filters

Optimality

- Assumptions
  - linear process
    - Gaussian process noise with zero mean
  - linear observation model
    - Gaussian observation noise with zero mean
  - independent process noise and observation noise
- Under its specific assumptions, the Kalman filter is the **guaranteed optimal** matching method
- guaranteed to provide the estimate with the minimum (root-) mean-squared-error (MSE or RMSE)

Example Situations

Optimal Situations (Assumptions Obeyed)	
(simulated) mobile robot moving at constant velocity, with position given by “GPS”	
(simulated) mobile robot moving at constant velocity, with positions given by overhead camera system	

Non-Optimal Situations (Assumptions Violated)	
non-linear process	robots subject to ‘random’ collisions (robot soccer)
non-linear observations	range and bearing from a remote beacon
non-independent process & observation noise	position error proportional to robot velocity

Misconceptions and Myths

A KF provides the correct position

no: it provides an estimate with the optimal (minimum) expected error

A KF cannot be beaten

it can be beaten by lucky guesses:on average, a KF provides the best estimate;another method may be lucky & beat it on a single run

A KF is *always* optimal

no: the KF is only optimal if the assumptions hold;if not, other filters may provide lower RMSE

KFs are complex and require advanced maths

no: the matrix formulation is straight-forward;matrix operations available in many software libraries

Summary

- Summary of this lecture
  - basic concepts
    - filtering overview
    - system model
    - filter operation
  - worked example
    - model example
    - illustrative examples of operation
  - properties of Kalman filters
    - the optimal filter
    - misconceptions and myths
- Next lecture
  - Particle Filtering
- Next lab
  - A new type of sensor!