Real-time object tracking with Kalman filters

Week 10 - Topic 2

Outline

1. Concept of Kalman filter
2. Kalman filter for object tracking (a step-by-step)
3. Code example
4. Examples of application
5. Advantages and limitations

Concept of Kalman filter

Where's the car?

Where's the car? Instruments-State Measurements

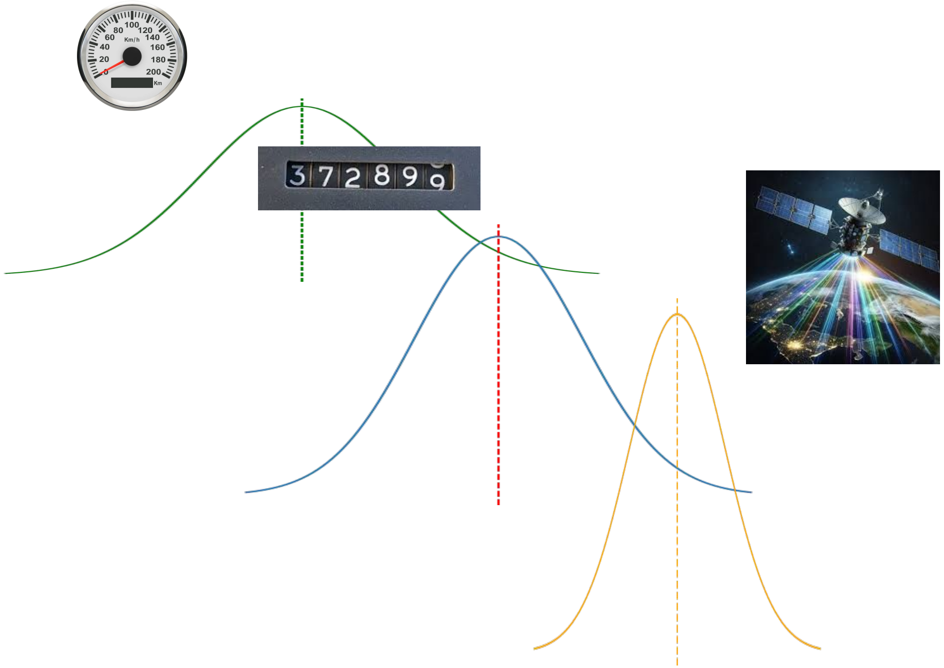
- Speedometer
  - ft/min

- Odometer
  - inches

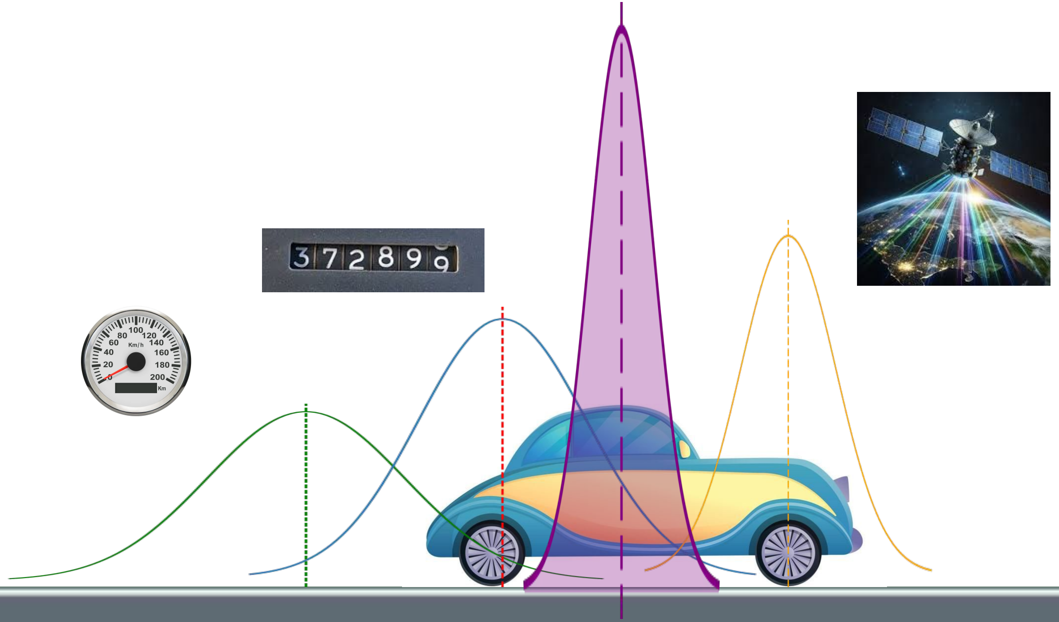
- GPS
  - hz

Where’s the car? Instruments- Gaussian Noise

- Speedometer
  - Mechanical precision
- Odometer
  - Tire slippage
  - Rough road conditions
- GPS
  - Update frequency



Where’s the car?



Assumptions

Measurement and state transition equations are linear:

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{w}_k$$

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{v}_k$$

Measurement errors are Gaussian:

$$\mathbf{w}_k \sim \mathcal{N}(\mathbf{w}_k; \mathbf{0}, \mathbf{R}_k)$$

$$p(\mathbf{z}_k | \mathbf{x}_k) = \mathcal{N}(\mathbf{z}_k; \mathbf{H}_k \mathbf{x}_k, \mathbf{R}_k)$$

State transition uncertainty is Gaussian:

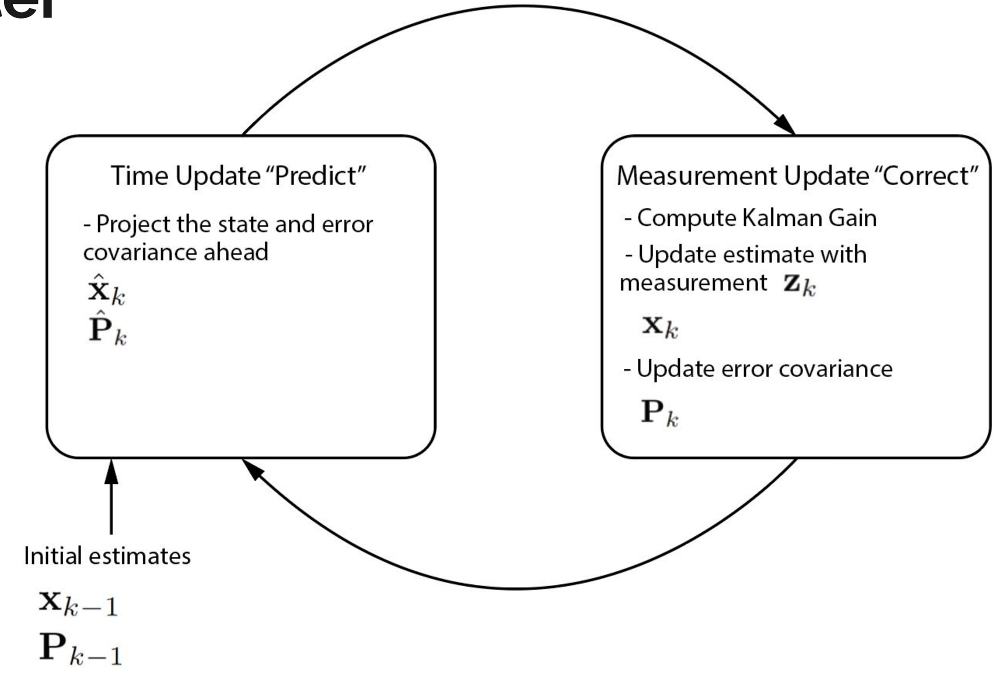
$$\mathbf{v}_k \sim \mathcal{N}(\mathbf{v}_k; \mathbf{0}, \mathbf{Q}_k)$$

$$p(\mathbf{x}_k | \mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_k; \mathbf{F}_k \mathbf{x}_{k-1}, \mathbf{Q}_k)$$

Concept of Kalman filter

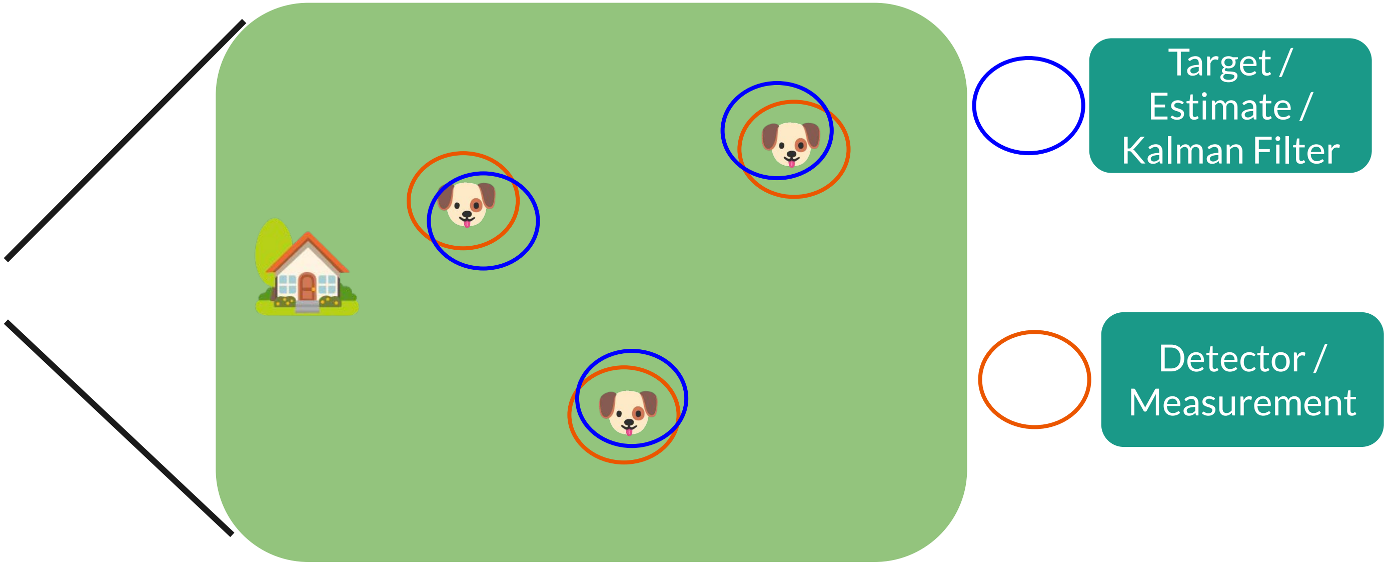
The Kalman filter updates the estimate of the state of a system based on the current measurement and the previous estimate.

Optimal MMSE estimator for a linear dynamic system with a linear measurement equation and Gaussian noises.



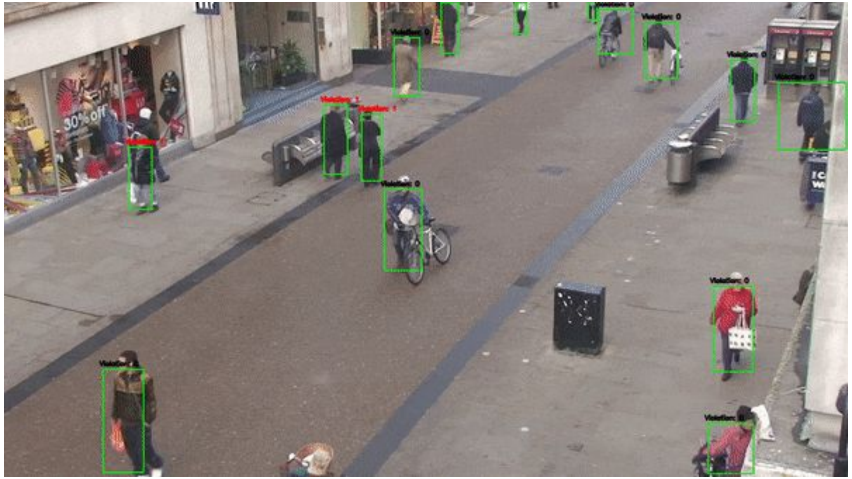
Kalman filter for object tracking

Where is Damiano's dog?



After we took care of Damiano’s dog

- Detect single or multiple objects in an image using detection methods like CNNs
- Track object(s) as they move using Kalman Filter
- Popular in computer vision applications for autonomous cars or robots or in video surveillance



Recipe to follow Damiano’s dog with Kalman Filter

1. Decide on an object detection method
2. Setup Kalman Filter to track the object
3. Decide on assignment method to assign a detection to a target (multiple objects)
4. Tune your detection method, assignment algorithm and Kalman Filter
5. (Follow Damiano’s dog around the garden.)

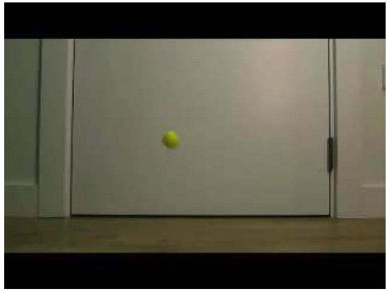
Some practical tips

- Detection methods
  - CNNs -> edge detection and defining a surrounding box or centroids
  - Background subtraction -> background is steady and moving object is the noise to detect, define bounding box or circle
  - Choose an appropriate method regarding the environment
- Kalman Filter
  - Update target state using assigned detected object
  - Try different Kalman Filter to identify the best fit
- Assignment method (for multiple objects)
  - Assign target estimate to correct detection
  - Hungarian algorithm as simple solution
  - Important to handle covered targets and in busy environments where objects can be close

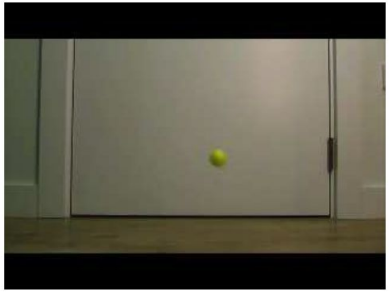
Code example

Code example - Tracking a ball

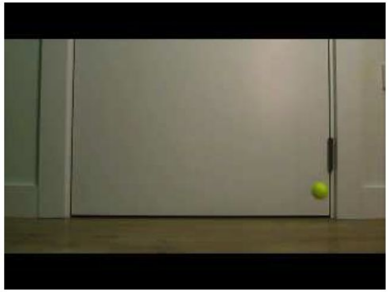
Frame No. : 3



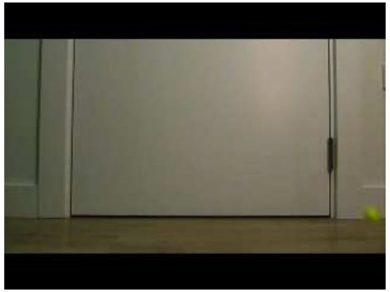
Frame No. : 15



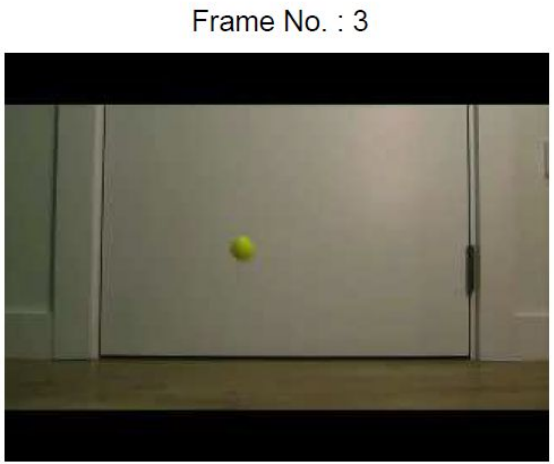
Frame No. : 45

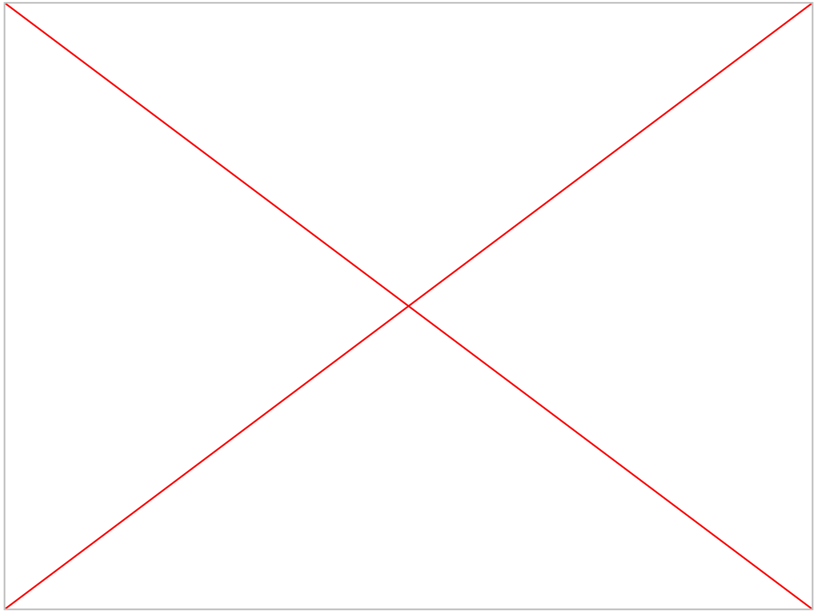


Frame No. : 80



Simulating Data





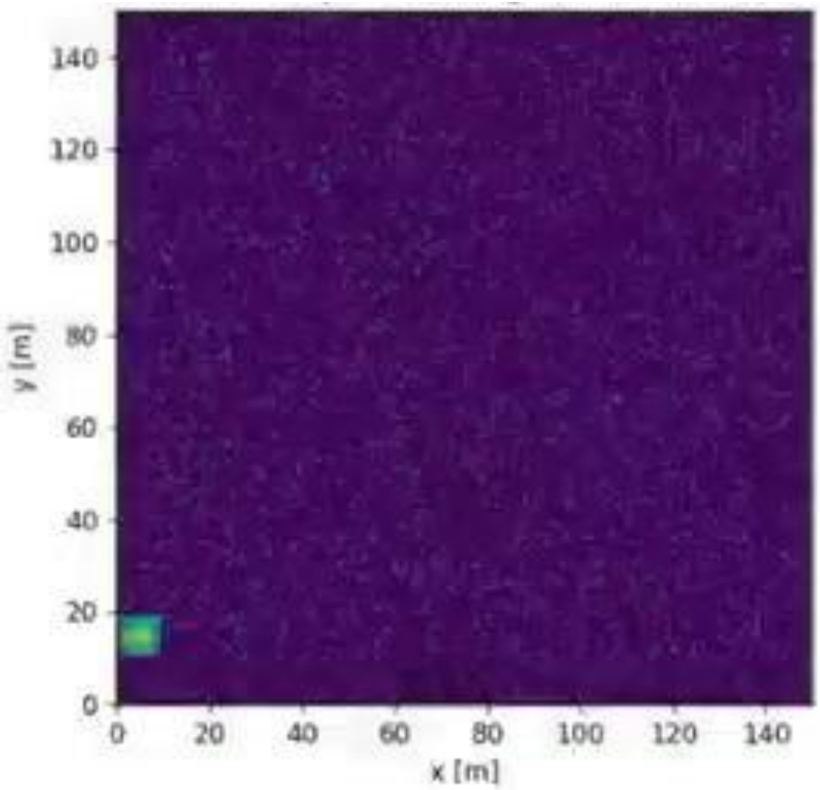
Blob Detector

What is a Blob Detector:

- Identifies distinct shapes (“blobs”) in images
- Shape, size, brightness

Performance:

+ Generally ok
- Misses some detections
- Gets confused by background noise



Kalman Filter

Implementation:

- Initializes using balls actual pose
- Predicts next pose
- Compares prediction and measured position from blob detector
  - *With detections:* Uses them to make estimate of ball position
  - *No detections:* Relies on previous predictions to estimate
- A lot of tuning for both detector and tracker

Performance:

- More stable than when using only a detector
- “Lags” a bit behind the detector

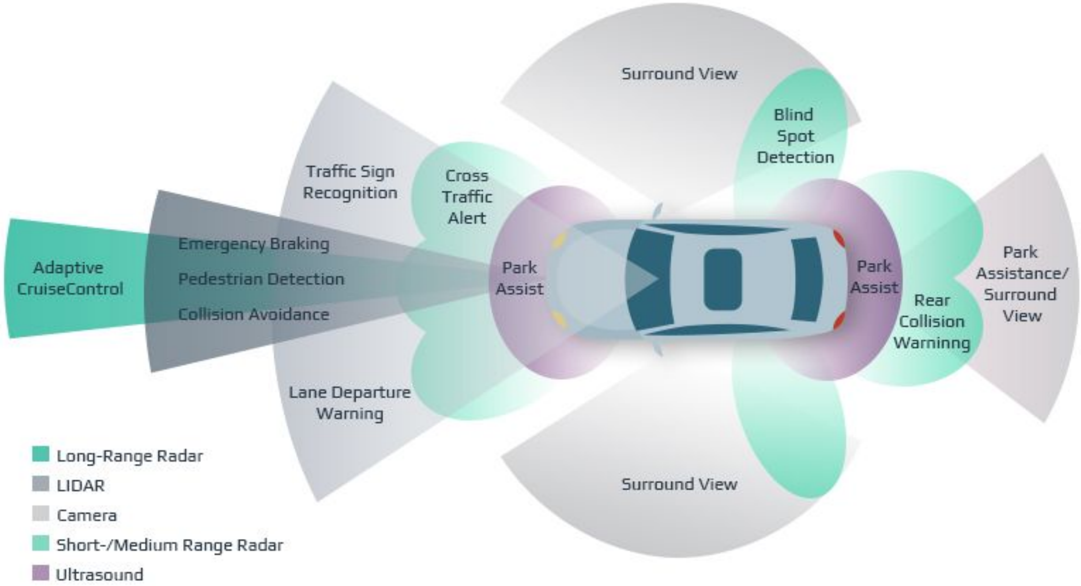
Examples of application

Examples of application: Crash prevention in autonomous driving

- **Tracking and Prediction**: Kalman filters are used in autonomous driving to track and predict the positions and speeds of nearby vehicles, pedestrians, and obstacles.

- **Sensor Data Fusion**: They combine data from multiple sensors—LIDAR, radar, and cameras—creating a more accurate, reliable estimate of each object's location and movement.

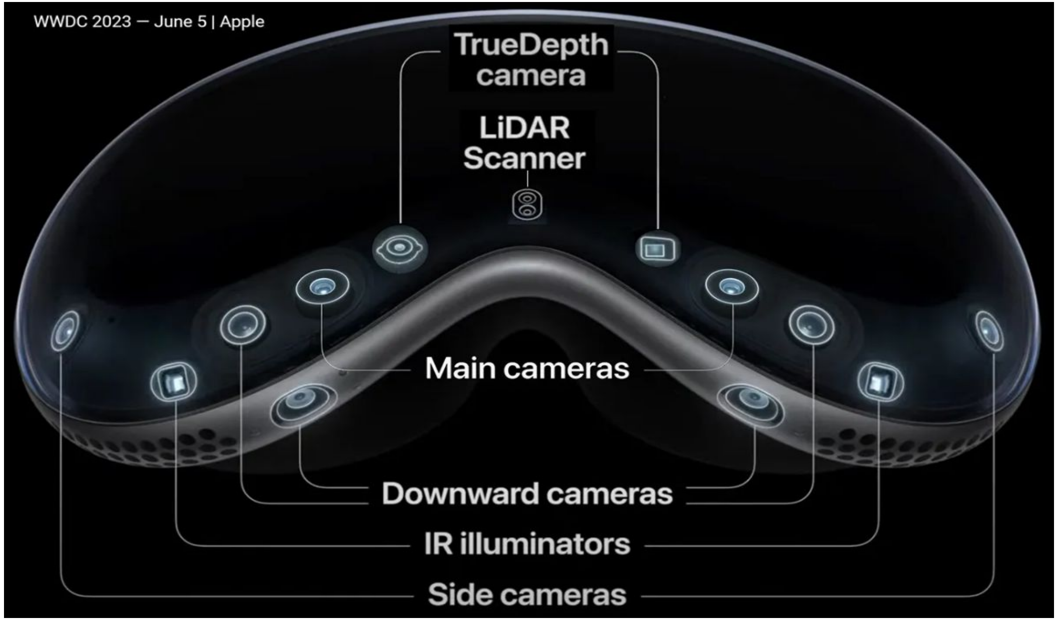
- **Collision Avoidance**: With these precise estimates, the vehicle can plan safe braking or evasive manoeuvres to avoid potential collisions.



Source: https://doi.org/10.3390/s20154126

Examples of application: Mixed reality (MR) goggles

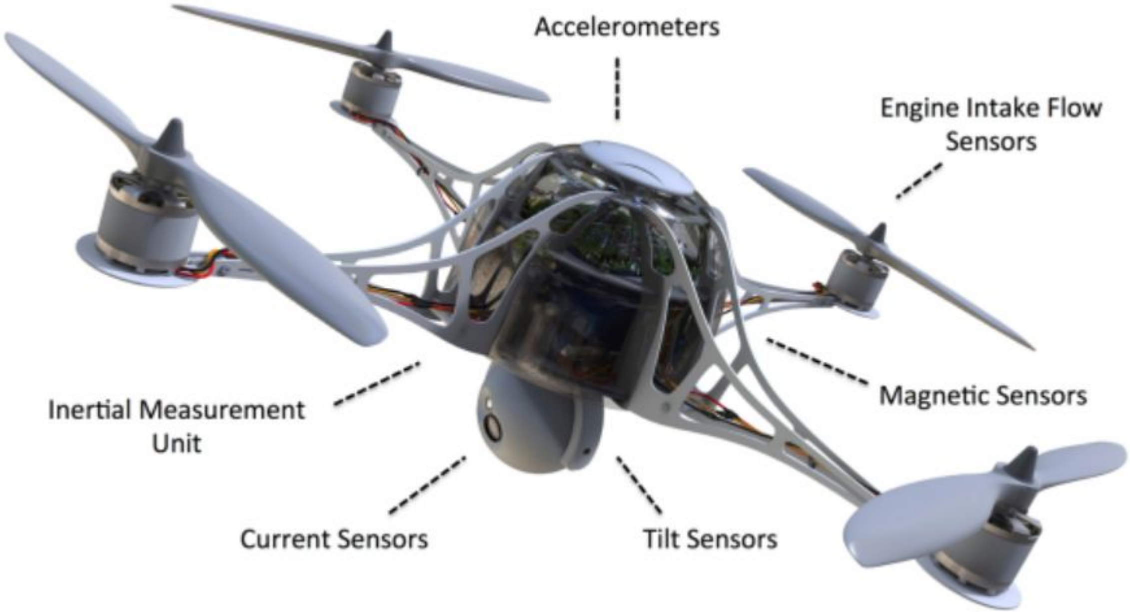
- **Head and Eye Tracking**: Kalman filters process data from various sensors to track head, eye, and surrounding movements, ensuring virtual objects stay in the user's view as they move.
- **Object Stabilisation**: By predicting head movements, Kalman filters help virtual elements remain anchored in the real world, reducing any “drifting” effect.
- **Image Stabilisation**: They filter out minor jitters from head movements, making the MR experience smoother and preventing shakiness in virtual overlays.



Source: https://doi.org/10.3390/s20051444

Examples of application: Drone flight stabilization

- **Position and Altitude Estimation**: Kalman filters combine data from various sensors to provide stable altitude and position readings, even with sensor noise.
- **Orientation Stabilisation**: By processing accelerometer and gyroscope data, Kalman filters maintain steady drone orientation, ensuring balance in turbulent conditions.
- **Flight Path Prediction**: They predict and adjust the drone's path to counteract environmental disturbances, helping the drone stay on course.



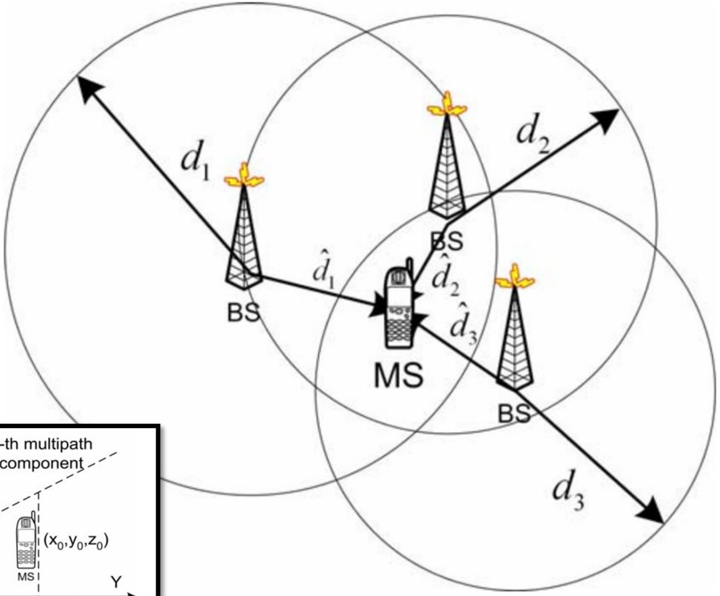
Source: **https://doi.org/10.3390/s22041677**

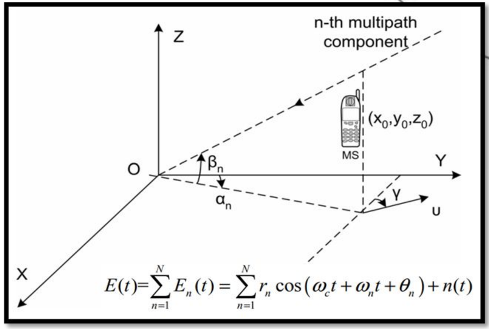
Examples of application: Phone's position and velocity tracking

EKF : Extended Kalman Filters

- **Fusion of Sensor Data :** The EKF combines data from the GPS and Inertial Measurement Unit (IMU) sensors . GPS provides absolute position data but is often slow and less accurate indoors, while the IMU provides high-frequency data on the phone's movement and orientation

- **Nonlinear Motion Modeling and Prediction**: EKF is designed to handle these nonlinearities by applying a linearization technique to approximate the motion model at each time step. continuously corrects its position and velocity estimates.



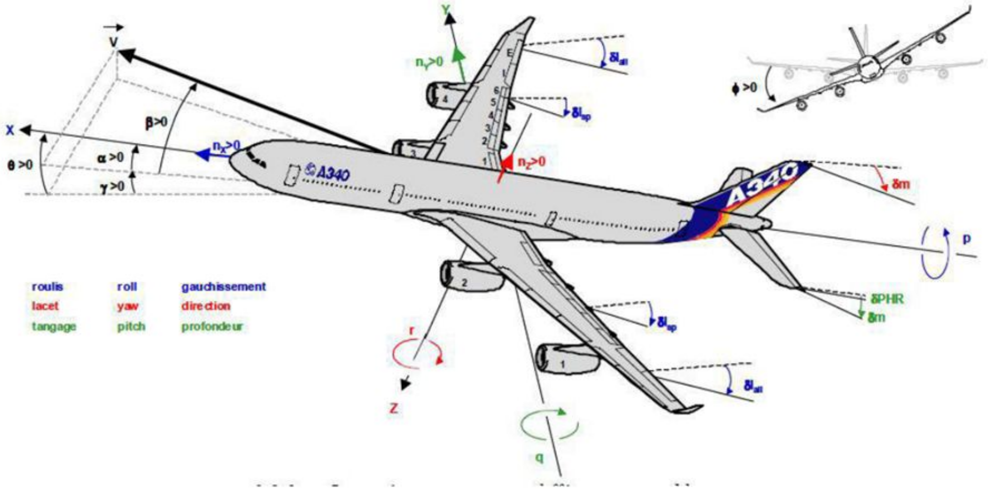


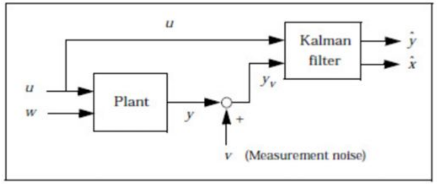
Source: ***https://doi.org/10.1109/WCNC.2005.1424911***

Examples of application: Airplane estimation control

- **Control Purpose :** to determine the velocity of an aircraft or sideslip angle, one could use a Doppler radar, the velocity of navigation system, or the relative wind information

- **Balancing the navigation:** Kalman Estimator could be built to combine all of this data and knowledge of the various systems dynamics to generate an overall best estimate of pitch, roll and sideslip angle.





Kalman estimator

Source: ***https://doi.org/10.1016/j.proeng.2014.09.084***

Advantages and limitations

Advantages and limitations

Advantages:

1. Predictive capabilities during observation gaps
2. Smooth predictions
3. Computationally efficient: suitable for real-time applications

Limitations:

1. Assumes gaussian noise
2. Depends on good modeling of the process
3. Assumes linearity
4. Slow response to abrupt changes
5. Increased complexity in higher dimensional spaces /multiple objects