



Secure Autonomous Systems

CSCI 6907/3907 86

Spring 2024

Prof. Sibin Mohan

<https://bit.ly/secureauto-spring24>

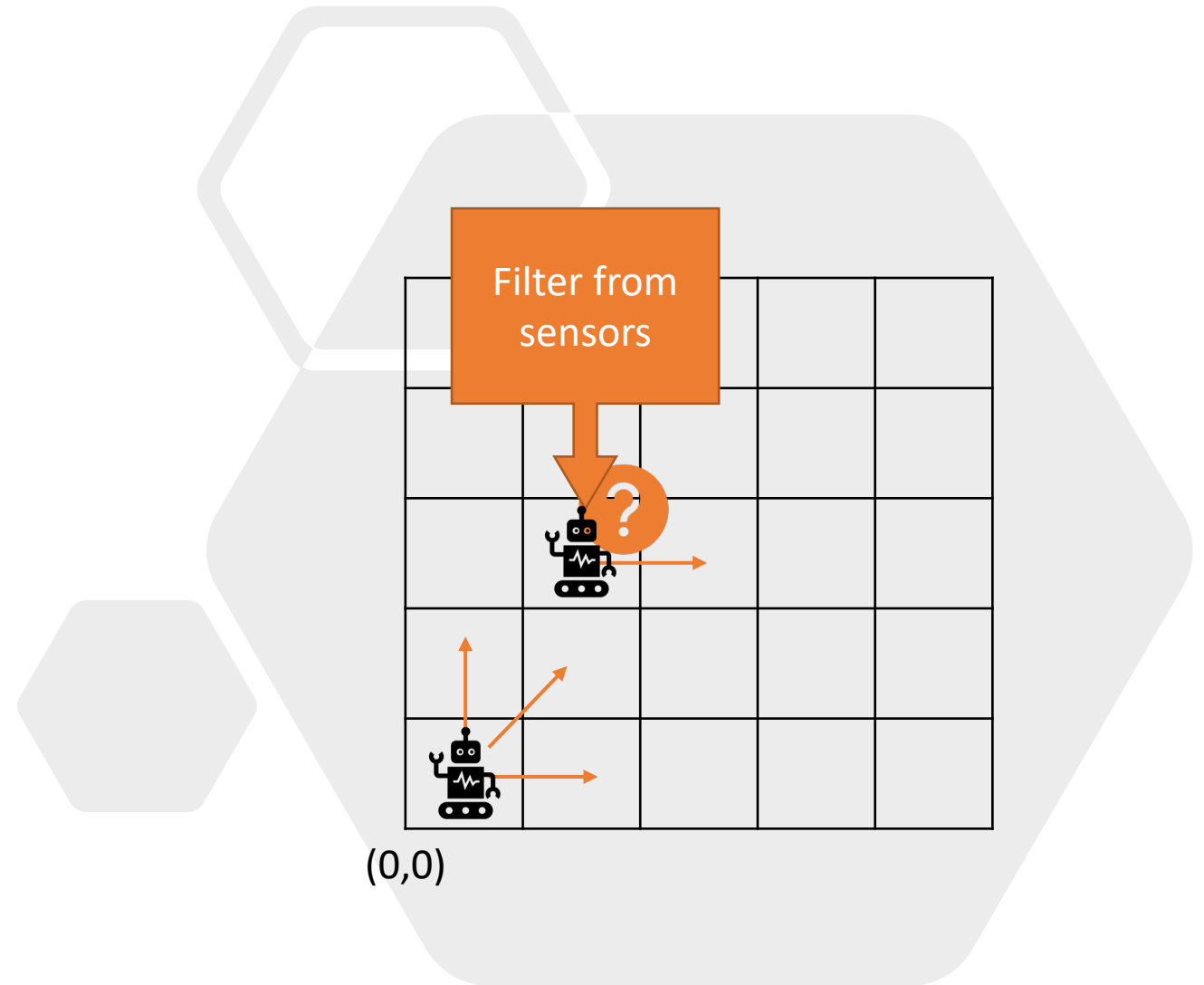
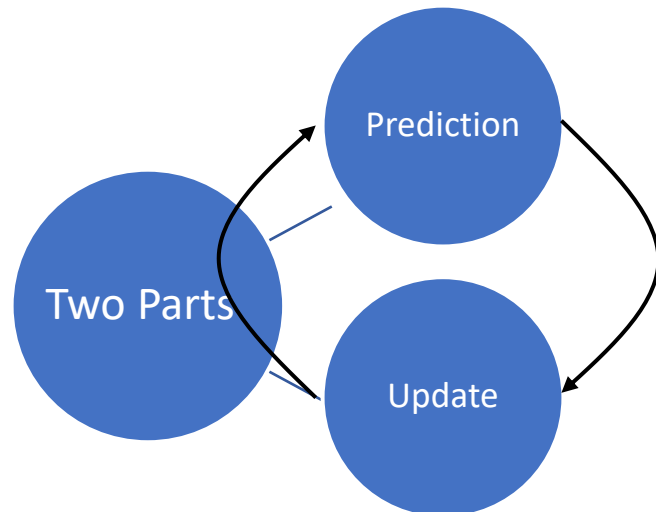


Kalman Filter

- For data fusion → estimate the state of a dynamic system
 - In the present (filtering)
 - past (smoothing)
 - future (prediction)
- Estimate the **state** of a robot from odometry data+observations
- Bayesian Filter

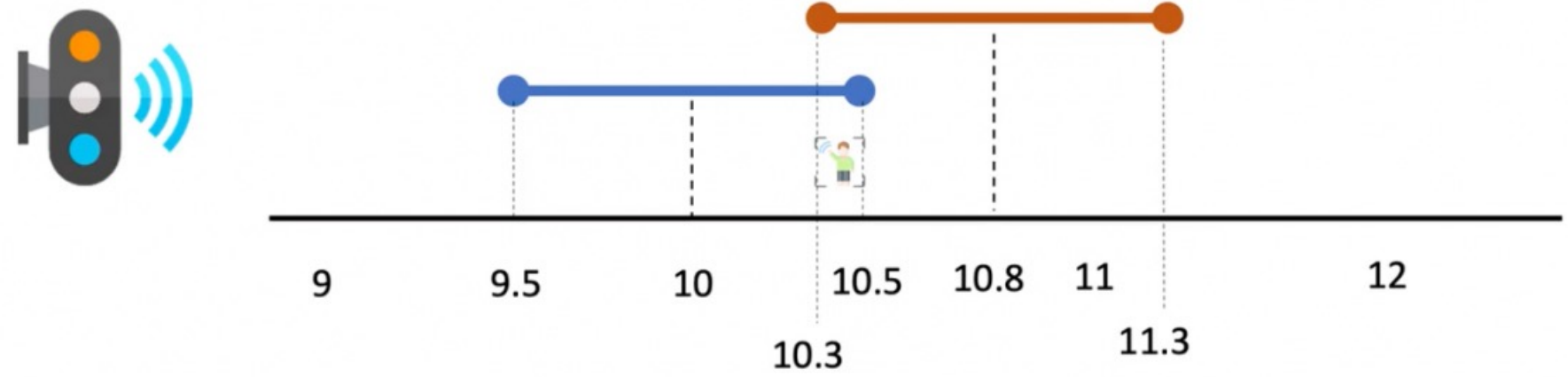
Bayes Filter

- probabilistic approach
- estimating an **unknown PDF**
 - recursively over time using
 - incoming **measurements** and
 - a mathematical process **model**



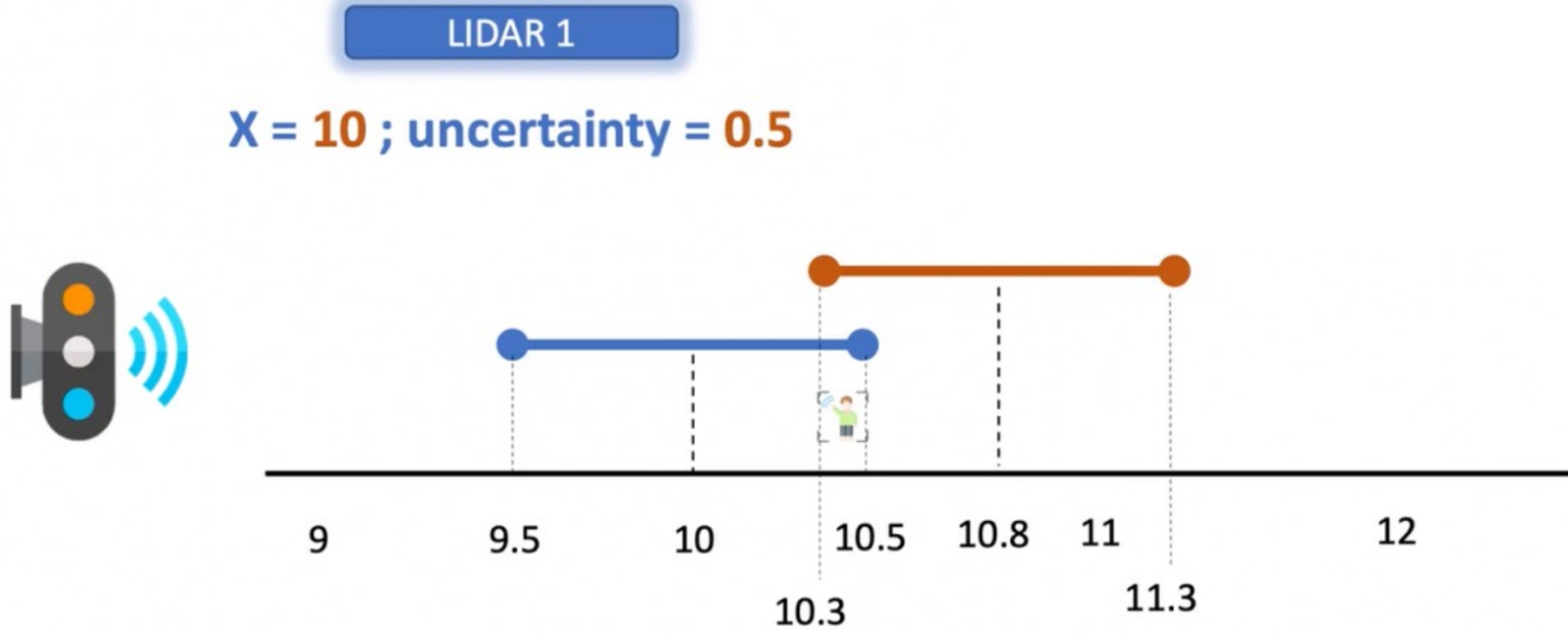
Kalman Filter

Sensors capture incomplete or noisy information



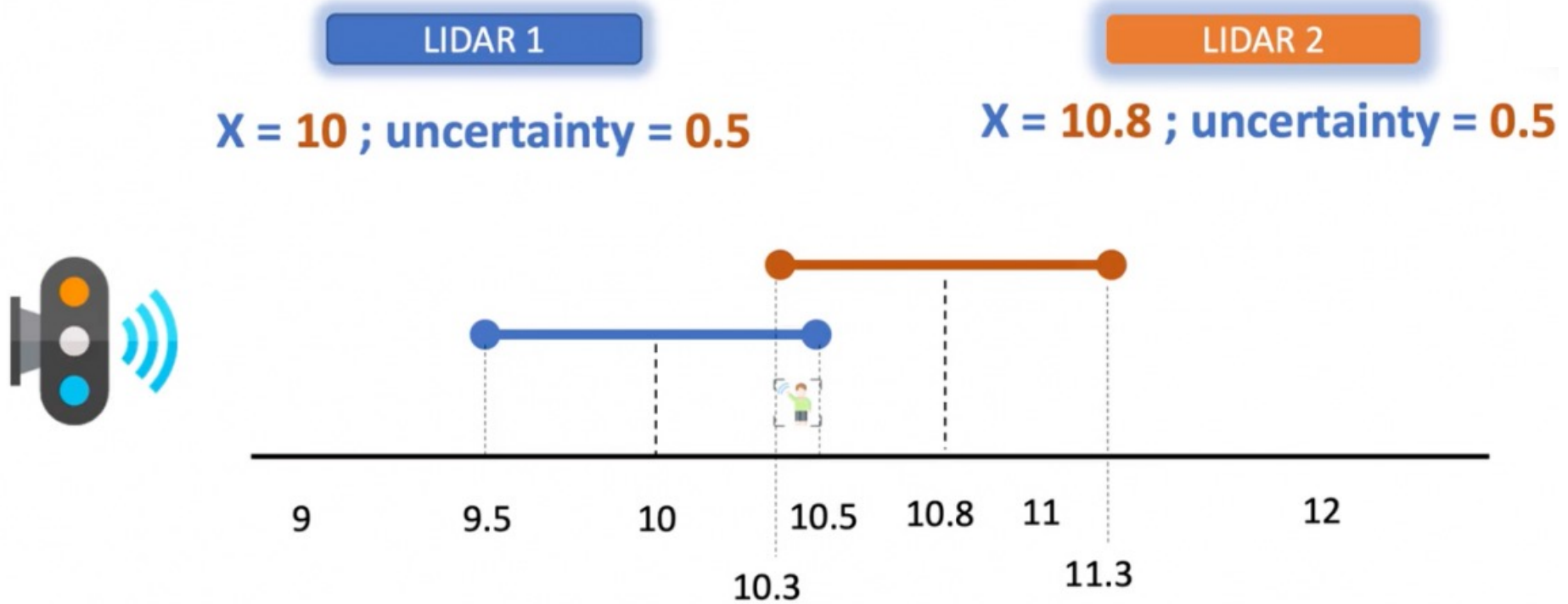
Kalman Filter

Sensors capture incomplete or noisy information



Kalman Filter

Sensors capture incomplete or noisy information



Kalman Filter

Sensors capture incomplete or noisy information

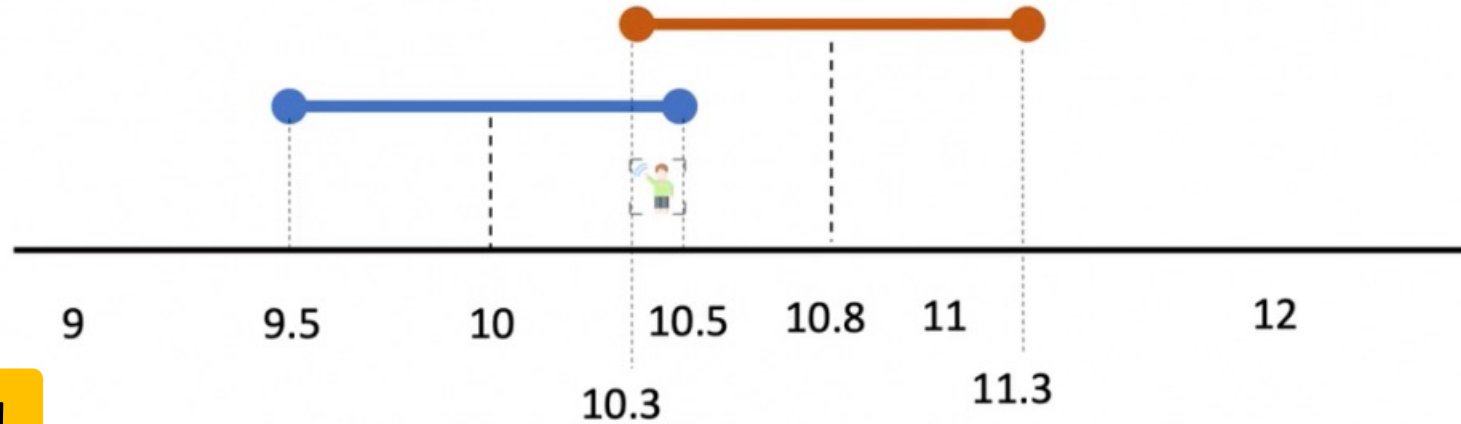


LIDAR 1

$X = 10$; uncertainty = 0.5

LIDAR 2

$X = 10.8$; uncertainty = 0.5



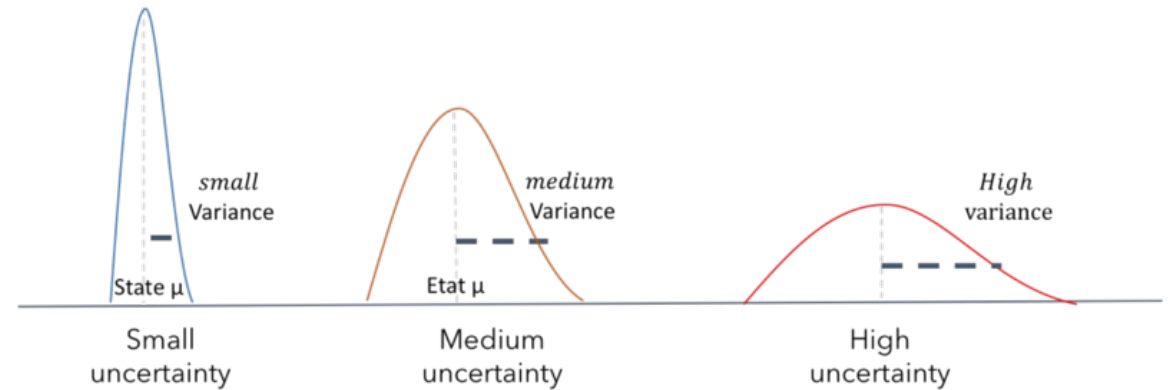
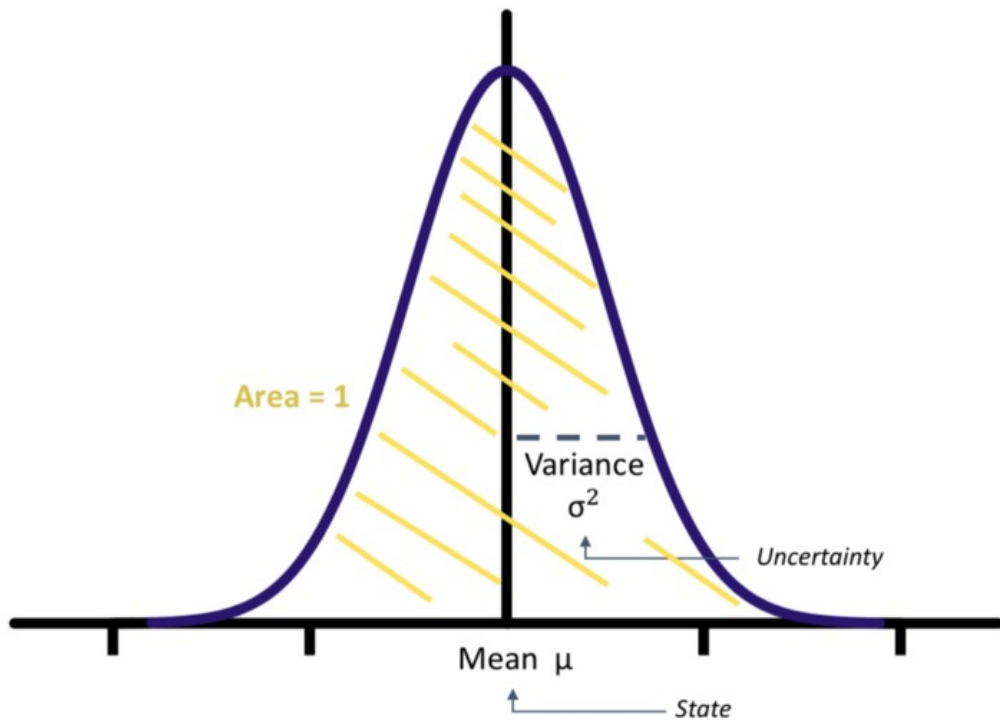
Both sensors equally certain!

Pedestrian \rightarrow 10.4
probability \rightarrow 0.98

The pedestrian is between 10.3 and 10.5

Gaussians

Kalman Filters express state and uncertainty using **Gaussians**



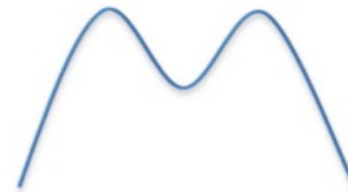
Kalman Filter and Gaussians

- Kalman Filter is unimodal
 - Single peak each time

Unimodal



Bimodal

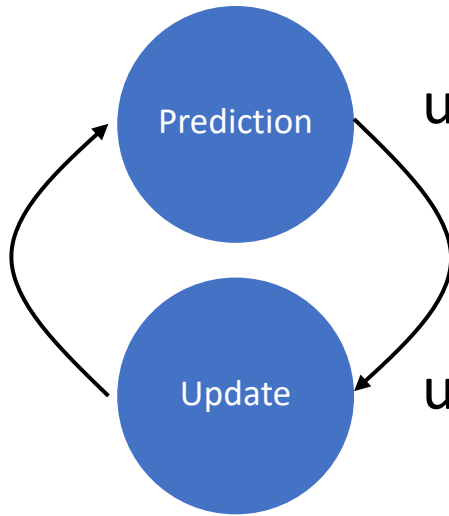


Multimodal



- An obstacle is **not** both, 10m away with 90% and 8m away with 70% probability
- **9.7m away with 98% or nothing**

Kalman Filter | Under the Hood



use estimated state to **predict future** state+uncertainty

use observations of sensors to **correct** prediction+**improve** accuracy

state of system

$$x = \begin{pmatrix} p \\ v \end{pmatrix}$$

position

velocity

Only needs **current observations**
and **previous predictions**

Kalman Filter | Steps

- Imagine sensor fusion between LiDAR and RADAR
- Only their outputs (late fusion)
- Kalman Filter: unimodals and Gaussians to represent state/uncertainty
- Prediction/update cycle

Kalman Filters | Predictions

- estimate state x' and uncertainty P' at time $t+1$

estimated state of system at $t+1$ ←

← **transition matrix** from t to $t+1$

← old state of system at time t

$$x' = Fx + u$$

← noise

estimated uncertainty of system at $t+1$ ←

← old uncertainty of system at time t

$$P' = FPF^T + Q$$

← covariance matrix including noise

Kalman Filter | Predictions [contd.]

- New position = former position (x) times matrix (F)

• F → motion model

- matrix describing how to move from t to t+1

position (t+1) = position (t) + velocity (t)*time

velocity (t+1) = velocity (t)

constant velocity

- Can consider other motion models for F

- Constant Turn Rate, Constant Velocity, Constant Acceleration

Kalman Filter | Prediction Matrix Example

$$x' = F \cdot x$$

$$x' = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}$$

$$x' = \begin{array}{l} \text{position} + \text{velocity} \cdot \Delta t \\ \dot{x} \end{array}$$

Kalman Filter | Update Measurements

- Adjust position, correct how to update next step

difference between actual measurements and prediction

$$y = z - Hx'$$

actual measurement

observation matrix

prediction

system error

$$S = HP'H^T + R$$

sensor noise

Kalman Gain [between 0 and 1]

$$K = P'H^T S^{-1}$$

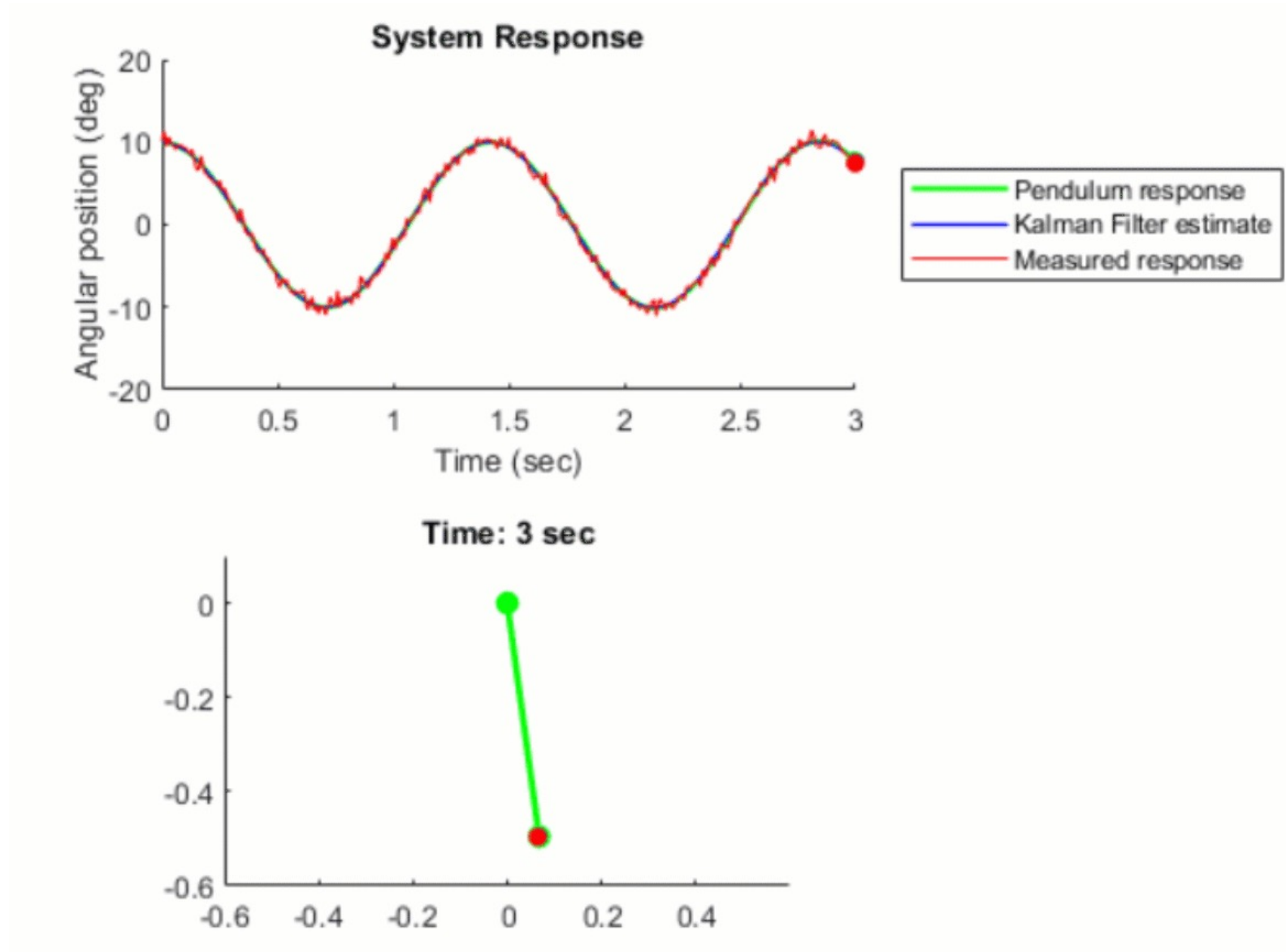
Kalman Filter | Final Update

- Compute new x and P

$$x = x' + Ky$$

$$P = (I - KH)P'$$

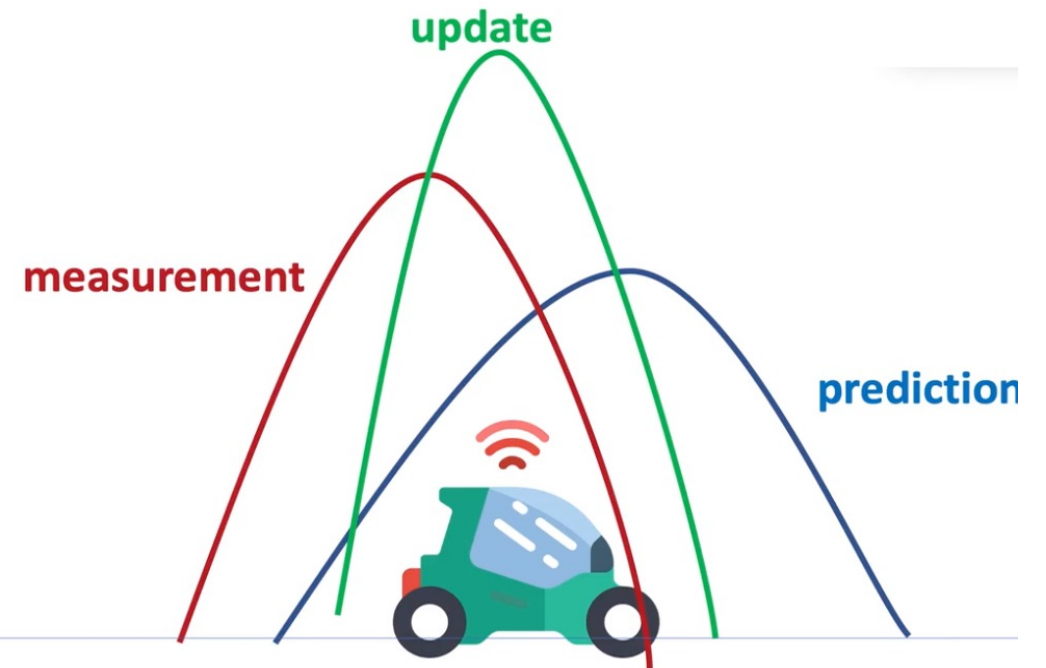
Kalman Filter | Simple Example



Kalman Filter | Bayesian Filtering

- We want to **estimate posterior**
 - posterior = prior (prediction) * likelihood (measurement)

$$P(A|B) \underset{\text{posterior}}{=} \frac{\overset{\text{likelihood}}{P(B|A)} * \overset{\text{prior}}{P(A)}}{\underset{\text{normalizer}}{P(B)}}$$



RADAR/LiDAR Sensor Fusion Flow

Each time a sensor value comes in → new prediction+update cycle

t=0



INITIALISATION 1ST MEASUREMENT



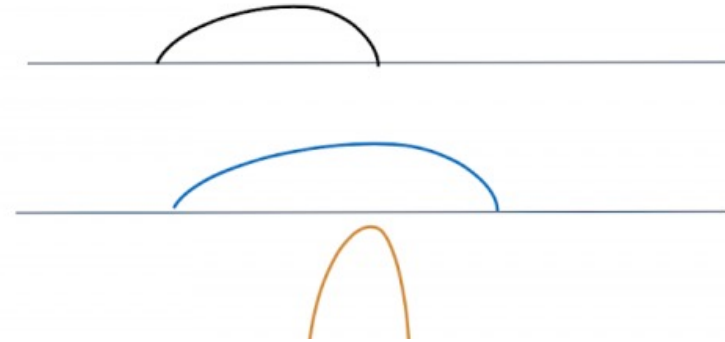
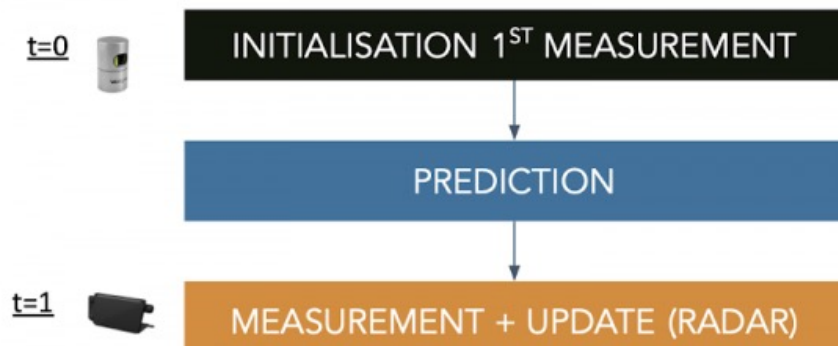
RADAR/LiDAR Sensor Fusion Flow

Each time a sensor value comes in → new prediction+update cycle



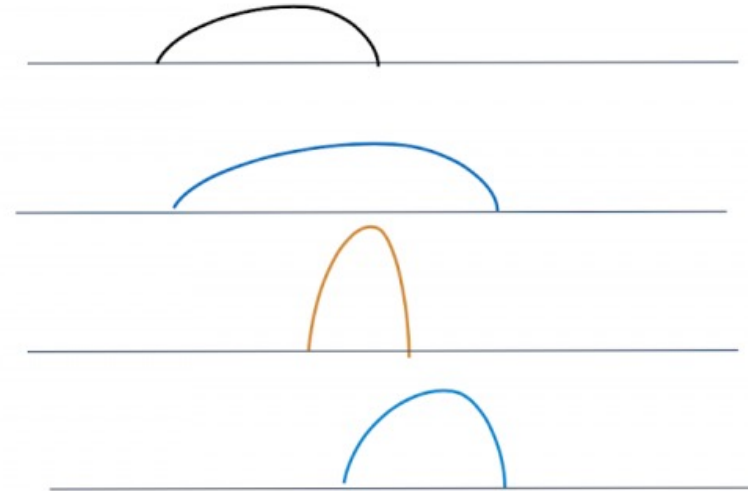
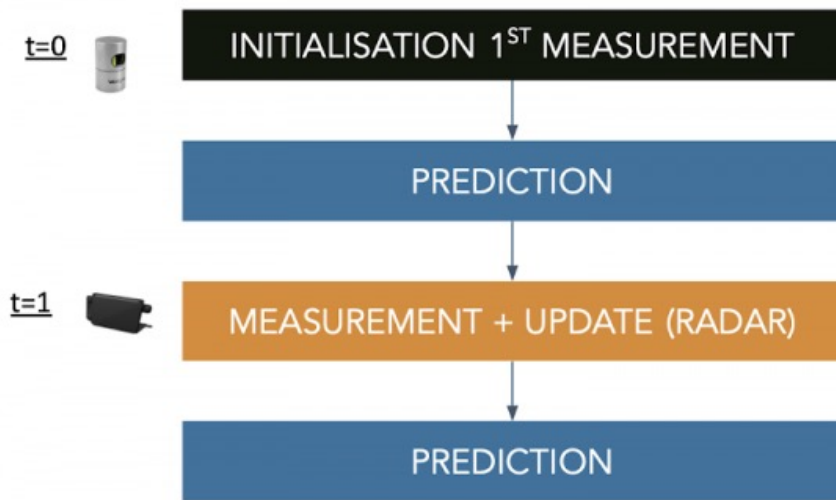
RADAR/LiDAR Sensor Fusion Flow

Each time a sensor value comes in → new prediction+update cycle



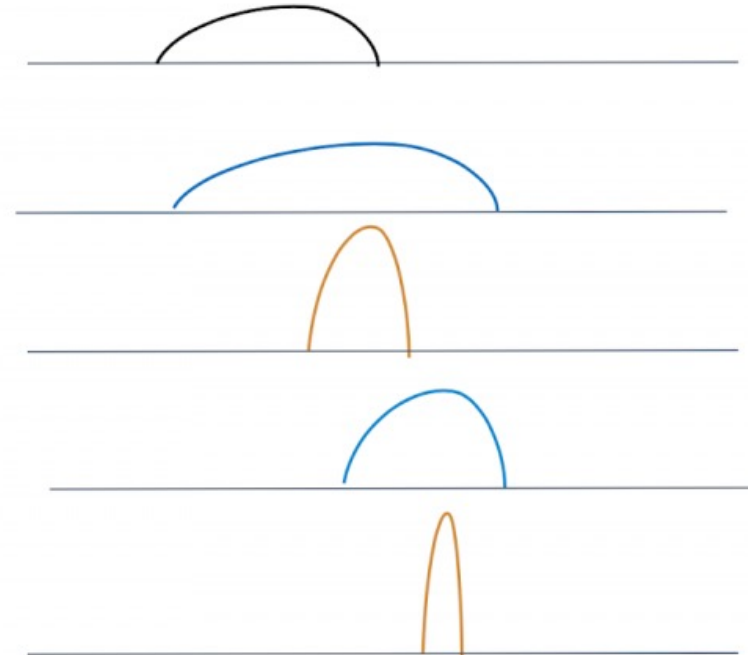
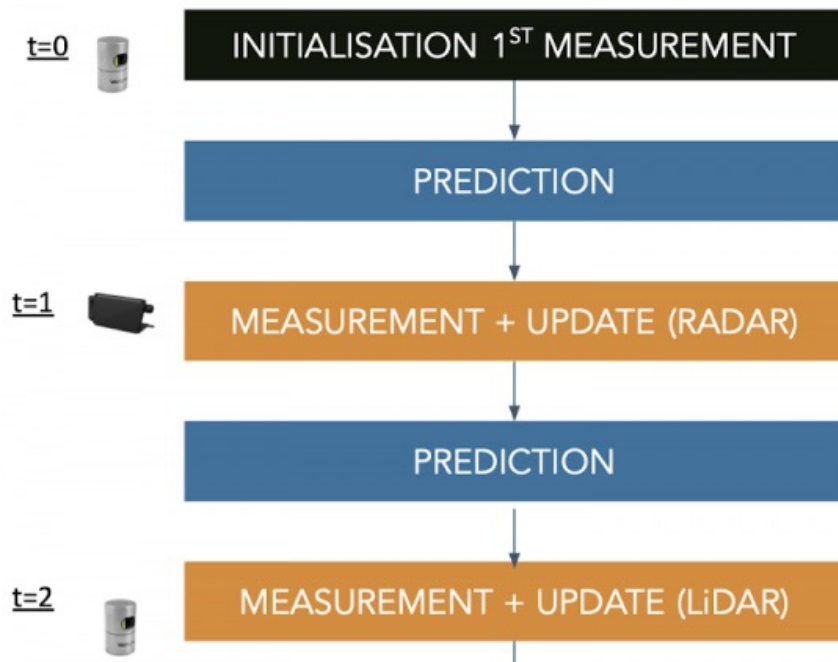
RADAR/LiDAR Sensor Fusion Flow

Each time a sensor value comes in → new prediction+update cycle



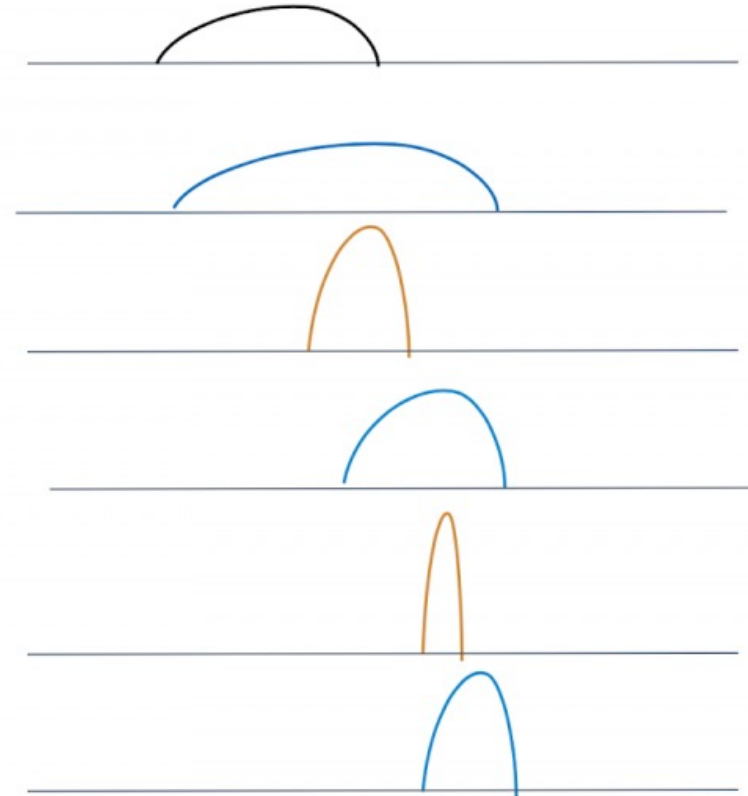
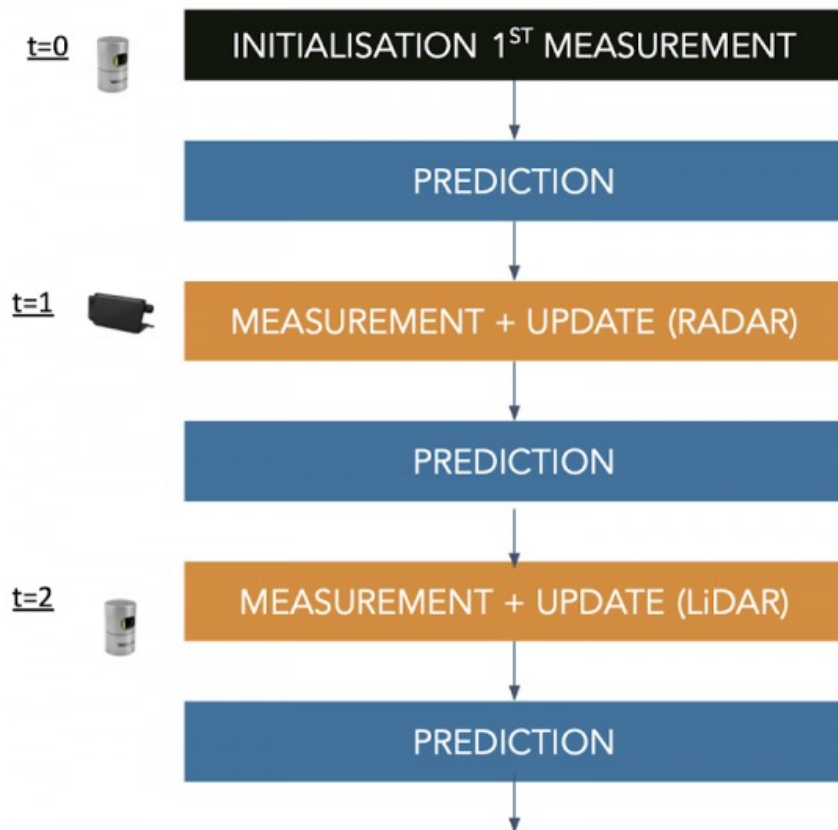
RADAR/LiDAR Sensor Fusion Flow

Each time a sensor value comes in → new prediction+update cycle



RADAR/LiDAR Sensor Fusion Flow

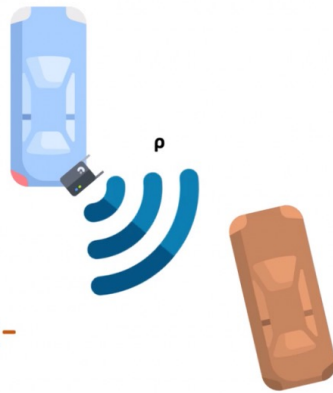
Each time a sensor value comes in → new prediction+update cycle



Kalman Filter | Hang on a minute...

- LiDAR has cartesian (linear) values of type $y=ax+b$
- RADAR → **not linear!**

Measurements are non-linear

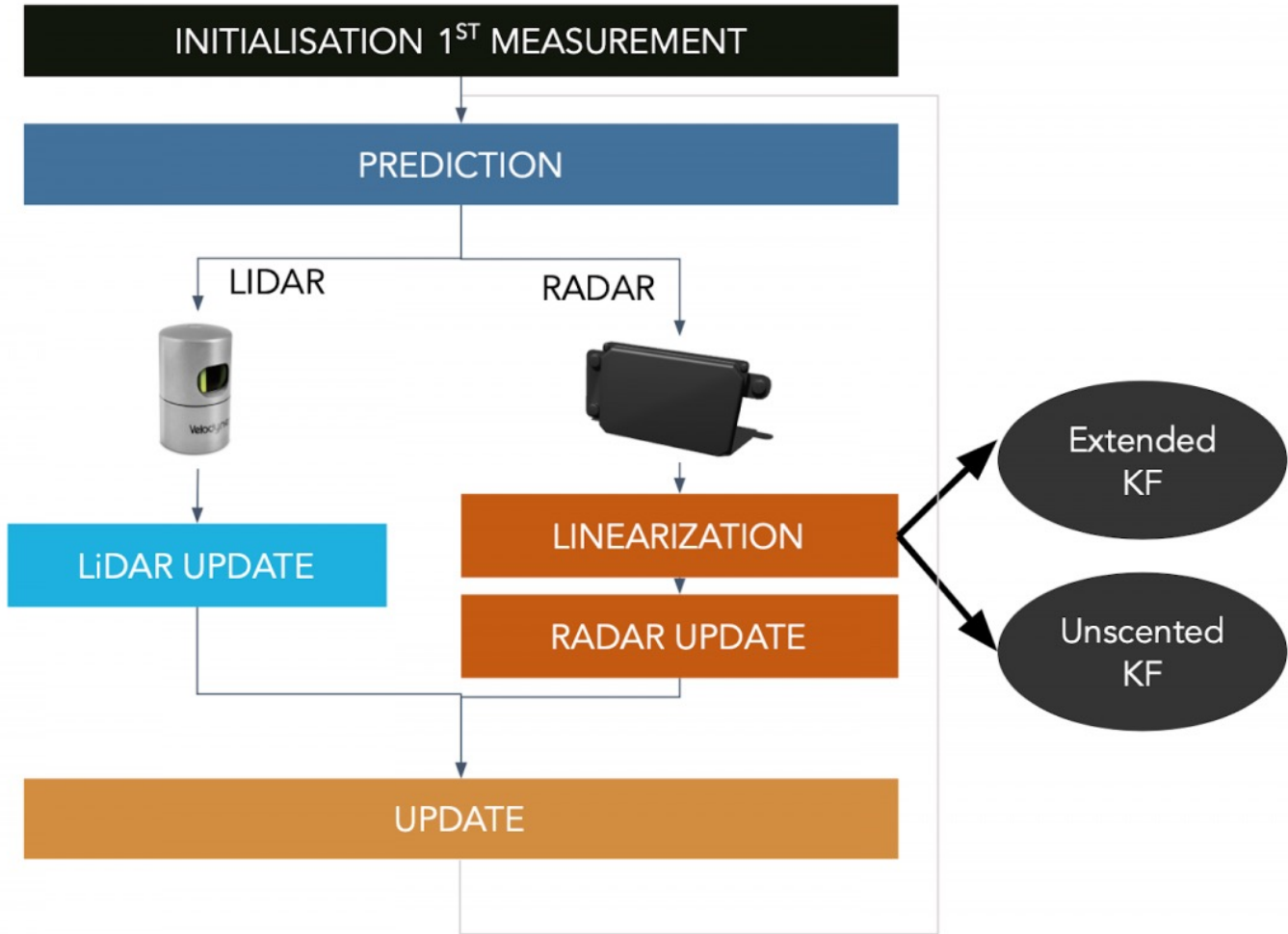


- How to reconcile the two?

Non-Linear Kalman Filters

- The world is **non-linear**
 - Not everything moves in straight lines
 - All sensors work differently
- Two types of Kalman Filters
 - Extended Kalman Filters
 - Unscented Kalman Filters

Final Sensor Fusion Flow



References

- Kalman Filter (in pictures):
<https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/>
- A decent primer on the Kalman Filter
<https://www.thinkautonomous.ai/blog/?p=sensor-fusion>
- Extended Kalman Filter with a nice video
<https://kusemanohar.wordpress.com/2020/04/08/sensor-fusion-extended-kalman-filter-ekf/>
- Some papers that use EKF
 - <https://ieeexplore.ieee.org/document/1338645>
 - <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7435659/>