ETH zürich

Embedded Systems with Drones

The State Estimator and the Extended Kalman Filter

Project-based Learning Center | ETH Zurich

Tommaso Polonelli tommaso.polonelli@pbl.ee.ethz.ch

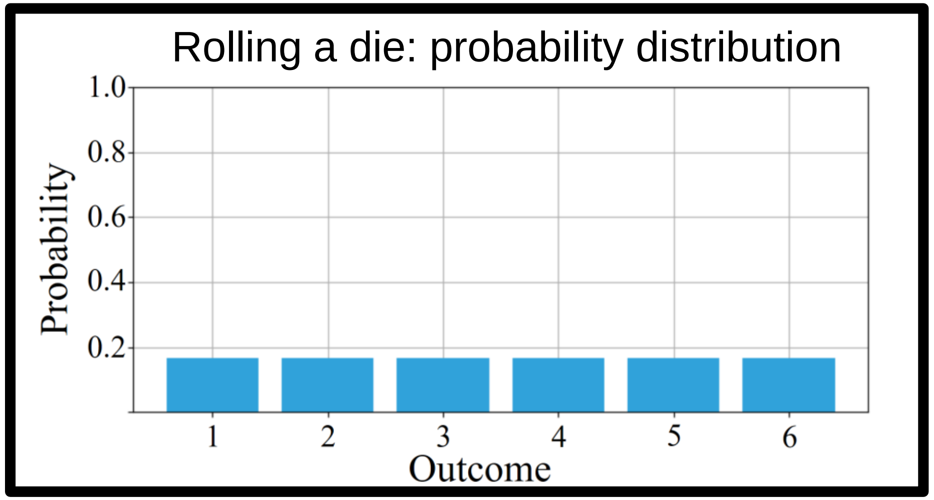
Michele Magno michele.magno@pbl.ee.ethz.ch

Vlad Niculescu vladn@iis.ee.ethz.ch

Credits: STMicroelectronics

Basic statistics knowledge

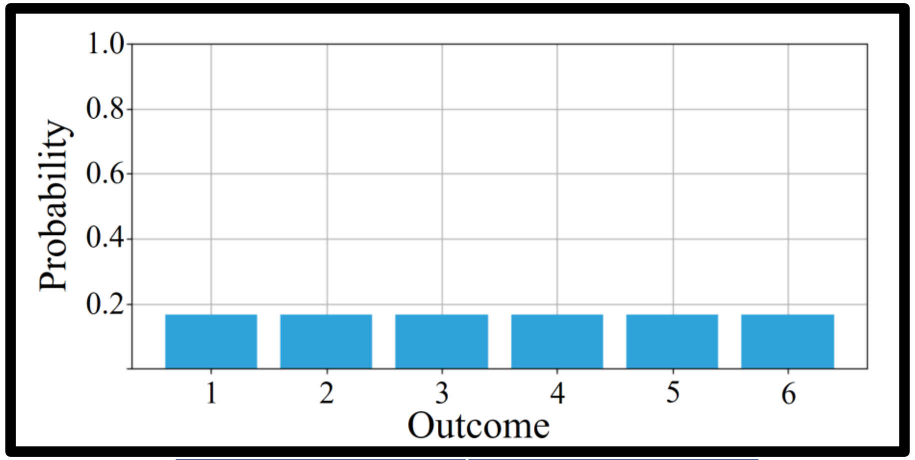
- Each time you roll a die the ***outcome*** will be between 1 and 6.
- ***Random variable***: the combination of values and the associated probabilities.
- The range of values is called the ***sample space*** (die: 1, 2, 3, 4, 5, 6).
- The ***probability distribution***: gives the probability for the random variable to take any value in a sample space.



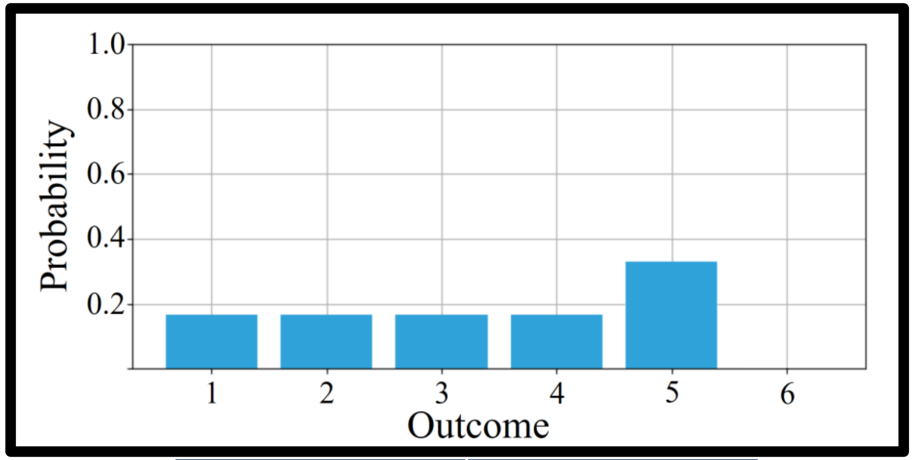
Value	Probability
1	1/6
2	1/6
3	1/6
4	1/6
5	1/6
6	1/6

Basic statistics knowledge

What if the die is biased? What if the face containing a 6, in fact is another 5?



Value	Probability
1	1/6
2	1/6
3	1/6
4	1/6
5	1/6
6	1/6



Value	Probability
1	1/6
2	1/6
3	1/6
4	1/6
5	2/6
6	0

Basic statistics knowledge

- The ***Expected Value*** (mean) of a random variable: the average over an infinite number of samples**.**

$$\mu = E(X) = \sum_{i=1}^{n} x_i p(x_i)$$

- The ***Standard Deviation:*** how much the sampled numbers vary from the mean.

$$\sigma^2 = Var(x) = E[(x-\mu)^2] = \sum_{i=1}^{n} (x_i - \mu)^2 p(x_i)$$

Example:

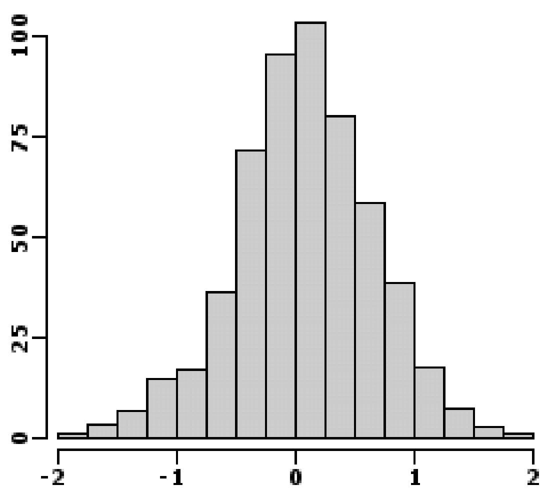
Value	Probability
1	1/6
2	1/6
3	1/6
4	1/6
5	2/6
6	0

$$\mu = 1\cdot\frac{1}{6} + 2\cdot\frac{1}{6} + 3\cdot\frac{1}{6} + 4\cdot\frac{1}{6} + 5\cdot\frac{2}{6} + 6\cdot\frac{0}{6} = 3.33$$

$$\sigma = \frac{(1-3.33)^2 + (2-3.33)^2 + (3-3.33)^2 + (4-3.33)^2 + 2\cdot(5-3.33)^2}{6} = 1.49$$

Types of probability distributions

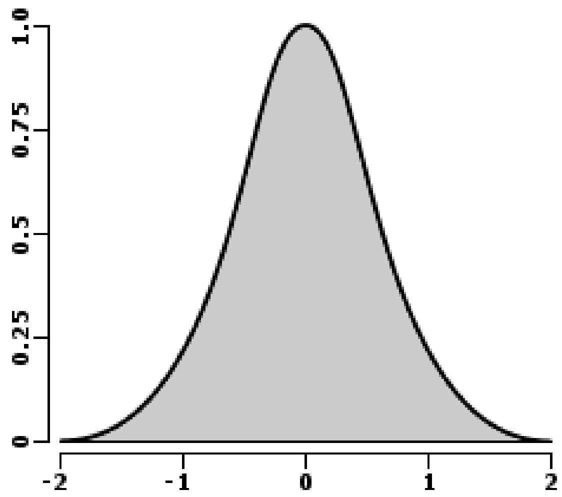
Discrete probability distributions



$$E(X)=\sum_{i=1}^{n} x_i\, p(x_i)$$

$$Var(x)=\sum_{i=1}^{n}(x_i-\mu)^2\, p(x_i)$$

Continuous probability distributions



$$E(X)=\int_{-\infty}^{\infty} x_i\, p(x_i)dx$$

$$Var(X)=\int_{-\infty}^{\infty}(x_i-\mu)^2\, p(x_i)dx$$

Gaussian Distributions

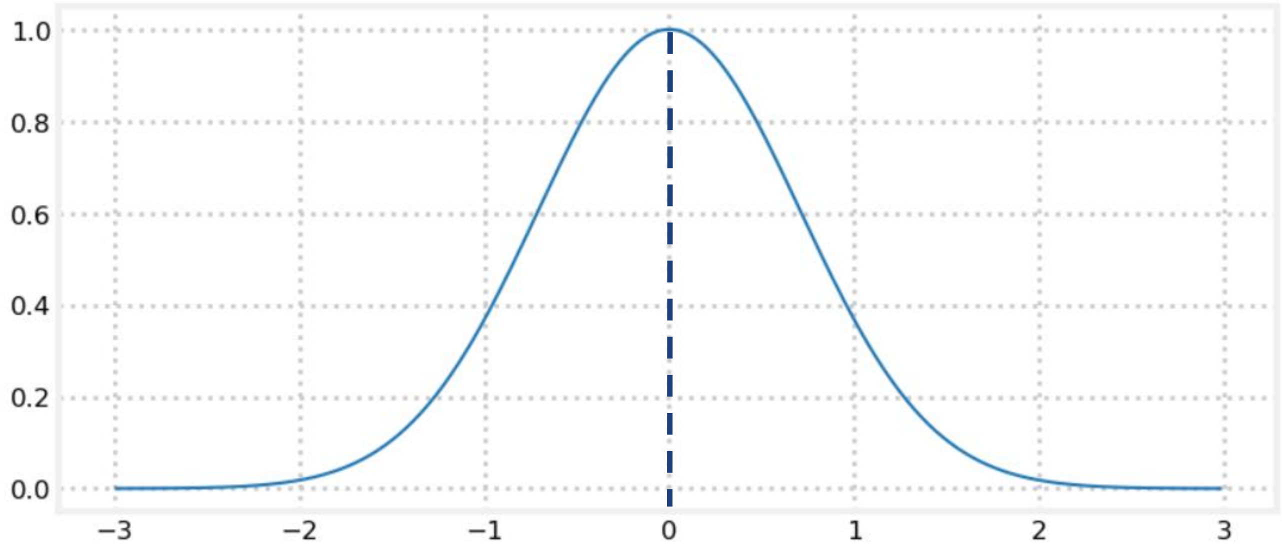
- A Gaussian is a continuous probability distribution completely described by the ***mean*** (μ) and ***variance*** (σ^2)

- μ – mean (Expected Value) is the center value of the plot

- σ – standard deviation dictates how “spread” the outcomes can be w.r.t. the mean

- A Gaussian distribution is also called a ***Normal Distribution***

$$f(\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$



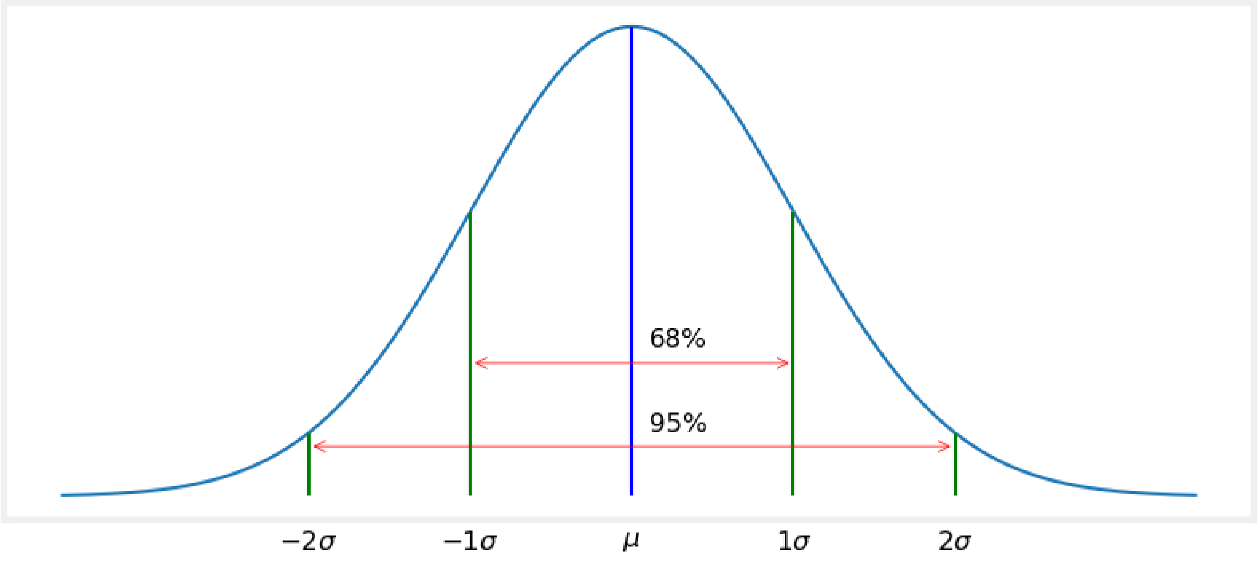
Gaussian Distributions

- A Gaussian is a continuous probability distribution completely described by the ***mean*** (μ) and ***variance*** (σ^2)

$$f(\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

Within a normally distributed batch:

- 68.27% of the values are found within $\pm\sigma$
- 95.45% of the values are found within $\pm 2\sigma$
- 99.73% of the values are found within $\pm 3\sigma$



Why discussing about Gaussians?

Temperature sensor

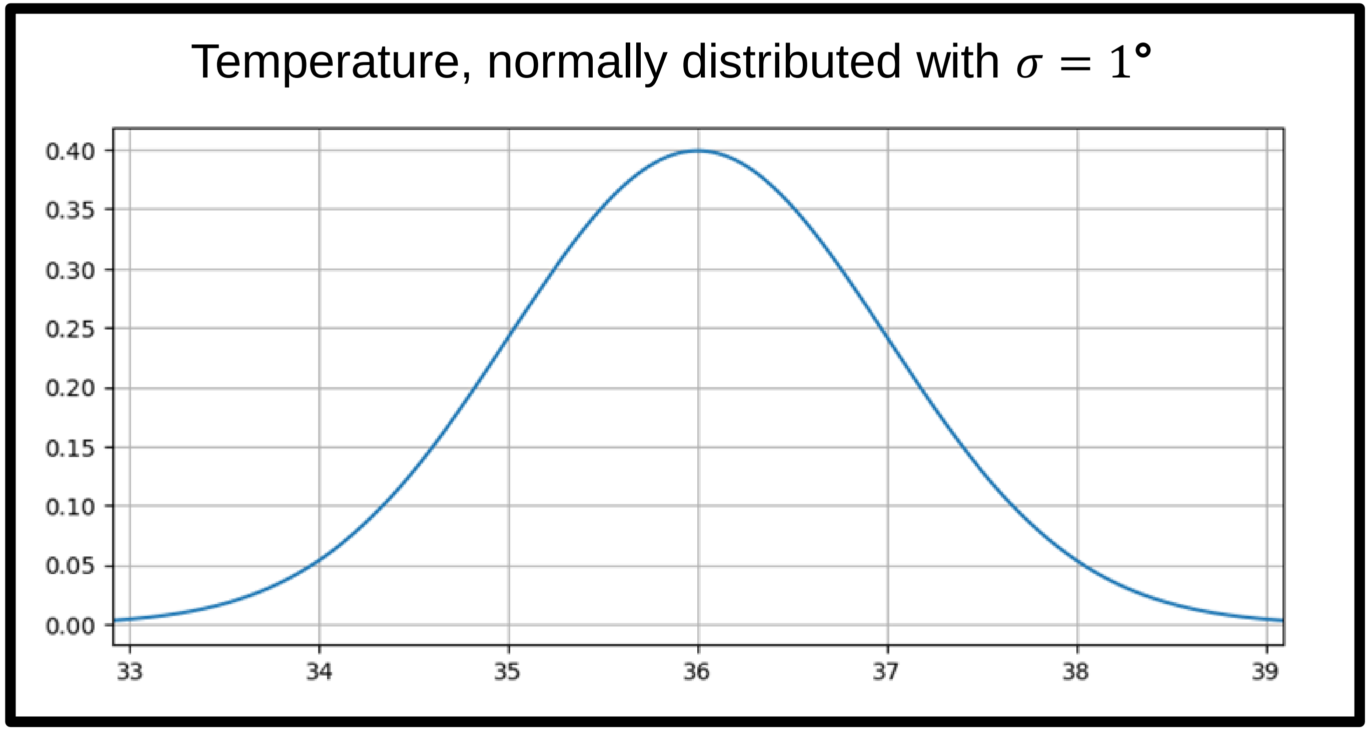
36° **How precise is this value?**

- **Any sensor has noise!**
- **In practice the sensor noise is modeled as a normally distributed variable.**

Why discussing about Gaussians?

Temperature sensor

36°



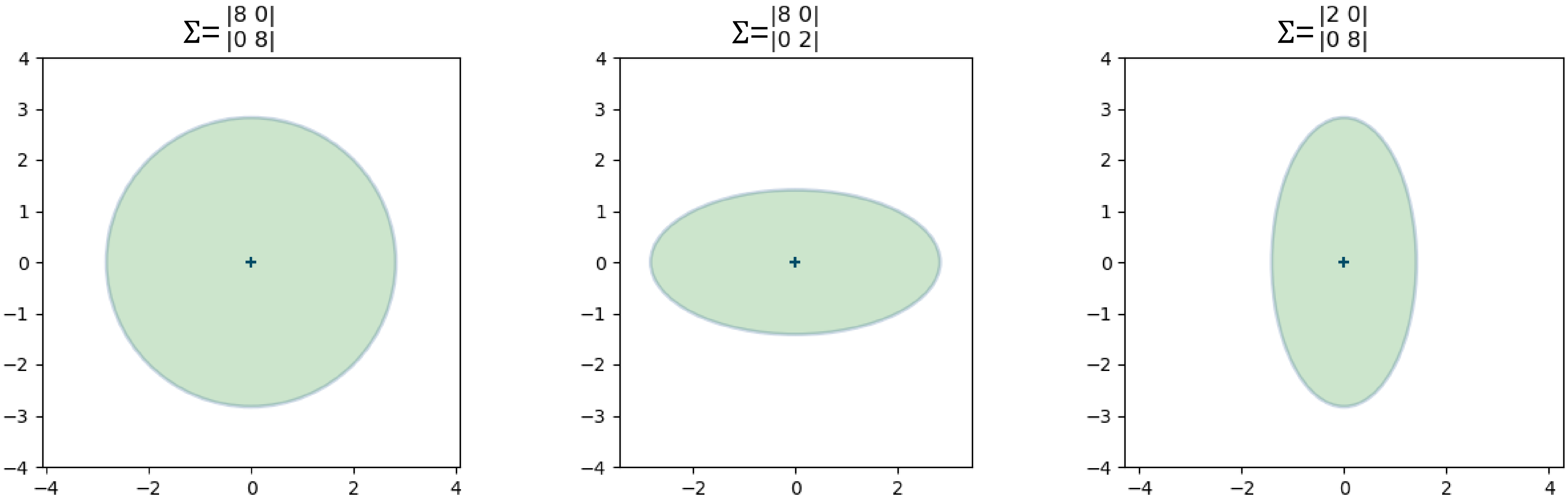
How to determine σ in practice?

1. Look it up in the datasheet.
2. Acquire a batch of samples and compute μ and σ.

Multivariate Gaussian distributions

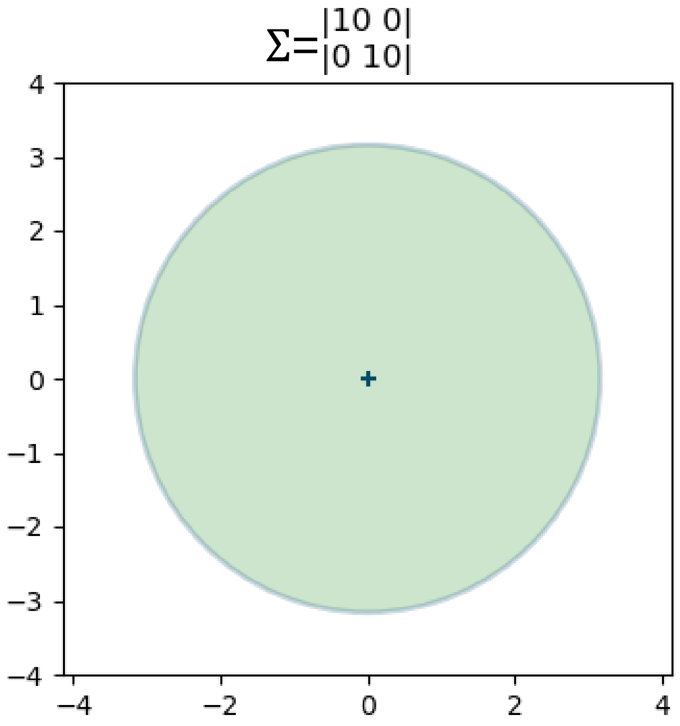
Representing a normal distribution with multiple dimension:

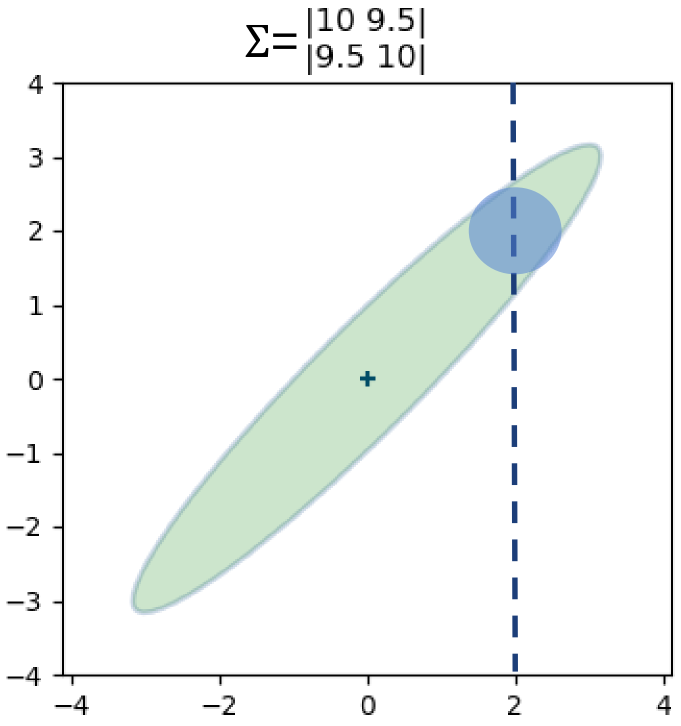
$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix}$$



Multivariate Gaussian distributions

Now suppose I were to tell you that we know that x=20. What can we infer about the value for y?
We can infer the position in y based on the correlation between x and y





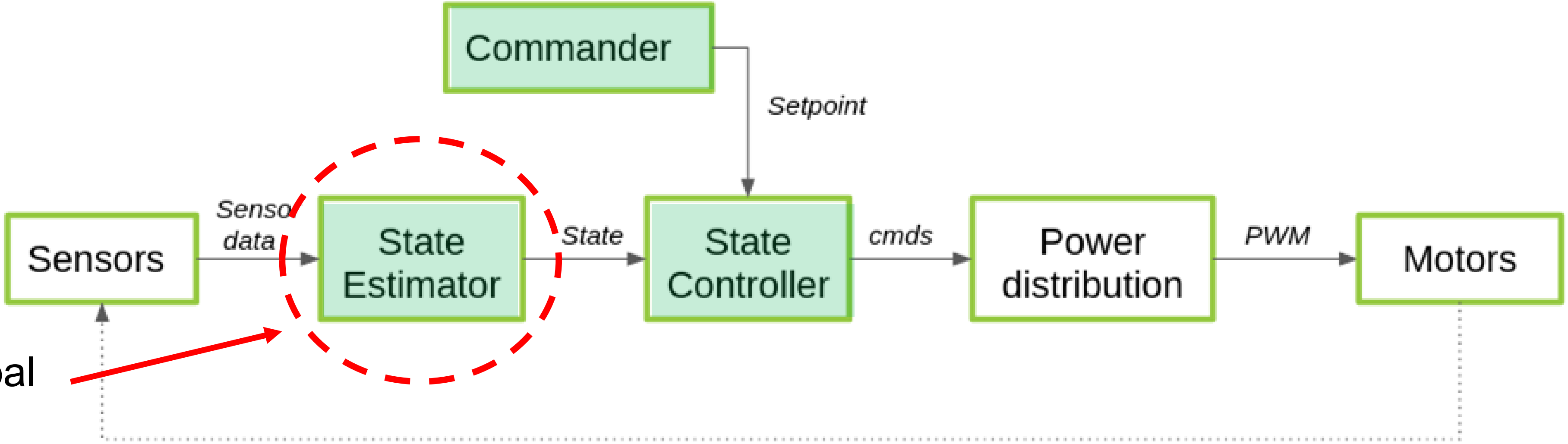
Going back to drones...

Architecture of a Drone's Stabilizer

The stabilizer represents the path from sensor acquisition to motor control.
It is the entity which ensures that the drone is properly flying. Elements:

- The State Estimator
- The State Controller
- The Commander

This lecture's goal



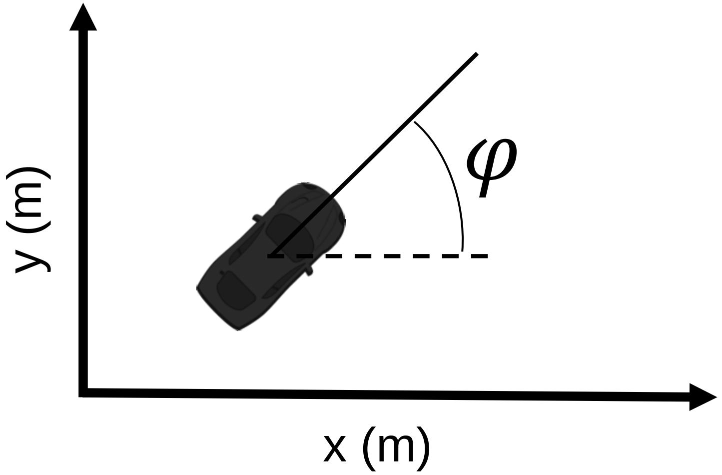
The State Space and the System Dynamics

- **STATE**: A set of variables used to describe the behavior of the system at a particular time.
- **DYNAMICS**: A set of linear/non-linear equations which describe the evolution of the system state.

1D moving object

$$\boldsymbol{x} = \begin{pmatrix} x \\ \dot{x} \end{pmatrix}$$

2D moving object



$$\boldsymbol{x} = \begin{pmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \varphi \end{pmatrix}$$

The State Space and the System Dynamics

- **STATE**: A set of variables used to describe the behavior of the system at a particular time.
- **DYNAMICS**: A set of linear/non-linear equations which describe the evolution of the system state.

$$\boldsymbol{x} = \begin{pmatrix} x \\ \dot{x} \end{pmatrix}$$

Dynamics:

System input

Ideal: $\boldsymbol{x}(\boldsymbol{k}+\mathbf{1}) = f\big(\boldsymbol{x}(\boldsymbol{k}), \boldsymbol{u}(\boldsymbol{k})\big)$

Real: $\boldsymbol{x}(\boldsymbol{k}+\mathbf{1}) = f\big(\boldsymbol{x}(\boldsymbol{k}), \boldsymbol{u}(\boldsymbol{k})\big) + \rho, \quad \rho \sim N(\mu, \sigma)$

timestamp

The State Space and the System Dynamics

- **STATE**: A set of variables used to describe the behavior of the system at a particular time.
- **DYNAMICS**: A set of linear/non-linear equations which describe the evolution of the system state.

$$\boldsymbol{x} = \begin{pmatrix} x \\ \dot{x} \end{pmatrix}$$

Dynamics (ideal case):

$$x(k+1) = x(k) + \dot{x}(k) \cdot \Delta t + \frac{\ddot{x}(k) \cdot \Delta t^2}{2}$$

$$\dot{x}(k+1) = \dot{x}(k) + \ddot{x}(k) \cdot \Delta t$$

The State Space and System Dynamics

Dynamics:

$$x(k+1) = x(k) + \dot{x}(k) \cdot \Delta t + \frac{\ddot{x}(k) \cdot \Delta t^2}{2}$$

$$\dot{x}(k+1) = \dot{x}(k) + \ddot{x}(k) \cdot \Delta t$$

$k = 0$

The State Space and System Dynamics

Dynamics:

$$x(k+1) = x(k) + \dot{x}(k) \cdot \Delta t + \frac{\ddot{x}(k) \cdot \Delta t^2}{2}$$

$$\dot{x}(k+1) = \dot{x}(k) + \ddot{x}(k) \cdot \Delta t$$

$k = 0$

The State Space and System Dynamics

Dynamics:

$$x(k+1) = x(k) + \dot{x}(k) \cdot \Delta t + \frac{(\ddot{x}(k) + \boldsymbol{\rho}) \cdot \Delta t^2}{2}$$

$$\dot{x}(k+1) = \dot{x}(k) + (\ddot{x}(k) + \boldsymbol{\rho}) \cdot \Delta t$$

$$\boldsymbol{\rho} \sim N(\mu, \sigma)$$

$$k = 1$$

The State Space and System Dynamics


Estimated position

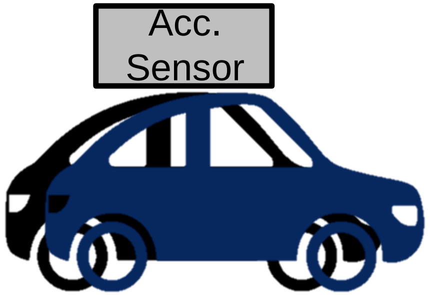

Real position

Dynamics:

$$x(k+1) = x(k) + \dot{x}(k) \cdot \Delta t + \frac{(\ddot{x}(k) + \boldsymbol{\rho}) \cdot \Delta t^2}{2}$$

$$\dot{x}(k+1) = \dot{x}(k) + (\ddot{x}(k) + \boldsymbol{\rho}) \cdot \Delta t$$

$$\boldsymbol{\rho} \sim N(\mu, \sigma)$$



$$k = 1$$

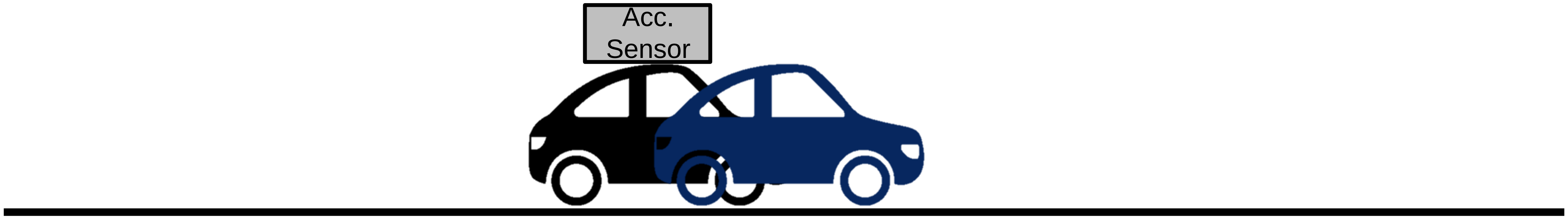
The State Space and System Dynamics

Dynamics:

$$x(k+1) = x(k) + \dot{x}(k) \cdot \Delta t + \frac{(\ddot{x}(k) + \boldsymbol{\rho}) \cdot \Delta t^2}{2}$$

$$\dot{x}(k+1) = \dot{x}(k) + (\ddot{x}(k) + \boldsymbol{\rho}) \cdot \Delta t$$

$$\boldsymbol{\rho} \sim N(\mu, \sigma)$$



$$k = 2$$

The State Space and System Dynamics

Dynamics:

$$x(k+1) = x(k) + \dot{x}(k) \cdot \Delta t + \frac{(\ddot{x}(k) + \boldsymbol{\rho}) \cdot \Delta t^2}{2}$$

$$\dot{x}(k+1) = \dot{x}(k) + (\ddot{x}(k) + \boldsymbol{\rho}) \cdot \Delta t$$

$$\boldsymbol{\rho} \sim N(\mu, \sigma)$$

ERROR GROWS UNBOUNDED!

$$k = 3$$

The State Space and System Dynamics

Dynamics:

$$x(k+1) = x(k) + \dot{x}(k) \cdot \Delta t + \frac{(\ddot{x}(k) + \boldsymbol{\rho}) \cdot \Delta t^2}{2}$$

$$\dot{x}(k+1) = \dot{x}(k) + (\ddot{x}(k) + \boldsymbol{\rho}) \cdot \Delta t$$

$$\boldsymbol{\rho} \sim N(\mu, \sigma)$$

Measurement model (UWB):

$$z(k) = x(k) + \boldsymbol{\rho}_M, \qquad \boldsymbol{\rho}_M \sim N(\mu_M, \sigma_M)$$

$$k = 0$$

The State Space and System Dynamics

Dynamics:

$$x(k+1) = x(k) + \dot{x}(k)\cdot\Delta t + \frac{(\ddot{x}(k)+\boldsymbol{\rho})\cdot\Delta t^2}{2}$$

$$\dot{x}(k+1) = \dot{x}(k) + (\ddot{x}(k)+\boldsymbol{\rho})\cdot\Delta t$$

$$\boldsymbol{\rho} \sim N(\mu,\sigma)$$

Measurement model (UWB):

$$z(k) = x(k) + \boldsymbol{\rho}_M, \qquad \boldsymbol{\rho}_M \sim N(\mu_M,\sigma_M)$$

$k = 1$

The State Space and System Dynamics

Dynamics:

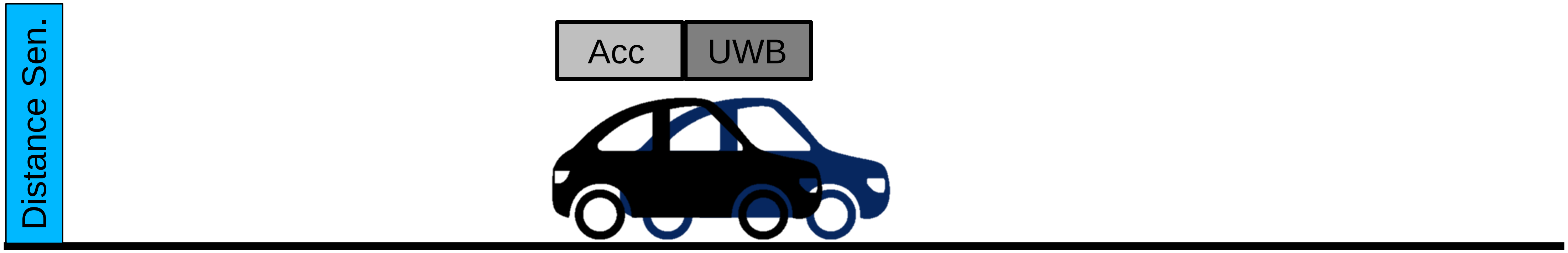
$$x(k+1) = x(k) + \dot{x}(k) \cdot \Delta t + \frac{(\ddot{x}(k) + \boldsymbol{\rho}) \cdot \Delta t^2}{2}$$

$$\dot{x}(k+1) = \dot{x}(k) + (\ddot{x}(k) + \boldsymbol{\rho}) \cdot \Delta t$$

$$\boldsymbol{\rho} \sim N(\mu, \sigma)$$

Measurement model (UWB):

$$z(k) = x(k) + \boldsymbol{\rho}_M, \qquad \boldsymbol{\rho}_M \sim N(\mu_M, \sigma_M)$$



$$k = 2$$

The State Space and System Dynamics

Dynamics:

$$x(k+1) = x(k) + \dot{x}(k)\cdot\Delta t + \frac{(\ddot{x}(k) + \boldsymbol{\rho})\cdot\Delta t^2}{2}$$

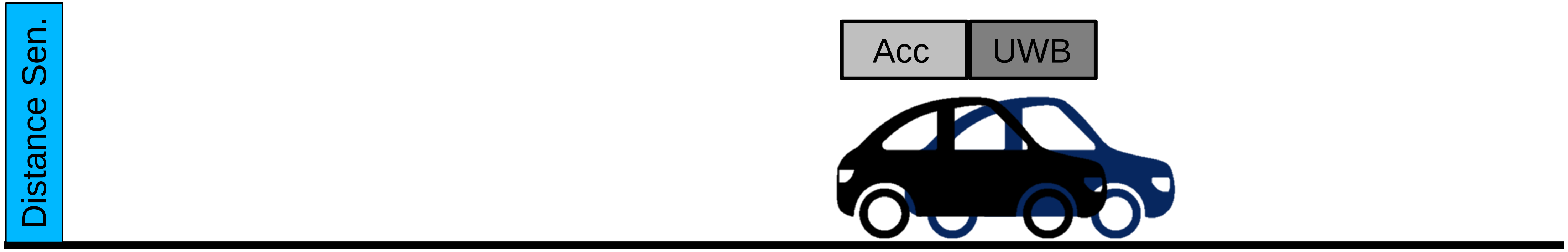
$$\dot{x}(k+1) = \dot{x}(k) + (\ddot{x}(k) + \boldsymbol{\rho})\cdot\Delta t$$

$$\boldsymbol{\rho} \sim N(\mu, \sigma)$$

Measurement model (UWB):

$$z(k) = x(k) + \boldsymbol{\rho}_M, \qquad \boldsymbol{\rho}_M \sim N(\mu_M, \sigma_M)$$

ERROR IS BOUNDED!



$$k = 3$$

The Kalman Filter

In the previous example, we have two sensor:

- The acceleration sensor
- The distance sensor

How do we combine the information from the two?

x_0, P_0 → Predict (dynamics acceleration sensor) → Update (distance sensor) → Predict (dynamics acceleration sensor) → Update (distance sensor) →

x_0, P_0 → **Predict** ⇄ **Update**

The Prediction Step

Dynamics of a linear system:

$$\boldsymbol{x}(k+1) = \underbrace{\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}}_{\text{A}} \boldsymbol{x}(k) + \underbrace{a(k) \begin{bmatrix} 0.5 \cdot \Delta t^2 \\ \Delta t \end{bmatrix}}_{\text{u}} + \underbrace{\boldsymbol{\rho} \begin{bmatrix} 0.5 \cdot \Delta t^2 \\ \Delta t \end{bmatrix}}_{w_p}$$

Predict

Update

$Q = Var(v)$ – process noise covariance matrix

$$x_0, P_0 \longrightarrow \textbf{Predict} \longrightarrow x_p(1), P_p(1) \longrightarrow \textbf{Update}$$

$$x_p(1) = \mathrm{A}(0)\mathrm{x}(0) + \mathrm{u}(0)$$

$$P_p(1) = \mathrm{A}(0)P(0)A^T(0) + Q$$

The Prediction Step

Dynamics of a linear system:

$x(k+1) = A(k)x(k) + u(k) + w_p$

$Q = Var(w_p)$ – process noise covariance matrix

$x_0, P_0 \longrightarrow$ **Predict** $\longrightarrow x_p(1),\, P_p(1) \longrightarrow$ **Update**

Prediction equations:

$x_p(1) = \mathrm{A}(0)\mathrm{x}(0) + \mathrm{u}(0)$

$P_p(1) = \mathrm{A}(0)P(0)A^T(0) + Q$

The Update Step

The measurement update

$z(k+1) = H(k)x(k) + w_m$

$R = Var(w_m)$ – measurement noise covariance matrix

$$x_0, P_0 \longrightarrow \textbf{Predict} \longrightarrow x_p(1),\ P_p(1) \longrightarrow \textbf{Update} \longrightarrow x_m(1),\ P_m(1)$$

Measurement update equations:

$x_m(1) = x_p(1) + \mathrm{K}(1)(\mathrm{z}(1) - \mathrm{H}(1)\ x_p(1))$

$P_m(1) = (I - K(1)H(1))\ P_p(1)(I - K(1)H(1))^T + K(0)RK^T(0)$

The Update Step

The measurement update

$z(k+1) = H(k)x(k) + w_m$

$R = Var(w_m)$ – measurement noise covariance matrix

$x_0, P_0 \longrightarrow$ **Predict** $\longrightarrow x_p, P_p \longrightarrow$ **Update** $\longrightarrow x_m, P_m$ (Posterior)

(Prior)

The Update Step

The measurement update

$z(k+1) = H(k)x(k) + w_m$

$R = Var(w_m)$ – measurement noise covariance matrix

$$x_0, P_0 \longrightarrow \textbf{Predict} \longrightarrow x_p(1),\, P_p(1) \longrightarrow \textbf{Update} \longrightarrow x_m(1),\, P_m(1)$$

$x_m = x_p + \mathrm{K}(\mathrm{z} - \mathrm{H}\ x_p)$

$P_m = (I - KH)\ P_p(I - KH)^T + KRK^T$

Filter Gain (Kalman Gain)

$K = P_p H^T (H P_p H^T + R)^{-1}$

Filter Gain

$$x_m = x_p + \mathrm{K}\overbrace{(\mathrm{z} - \mathrm{H}\ x_p)}^{\text{Correction factor}}$$

Posterior mean

Prior mean

The Kalman Filter

1. Prediction step:

$$x_p = \mathrm{A}x_m + \mathrm{u}$$

$$P_p = \mathrm{A}P_m A^T + Q$$

2. Measurement update step:

$$K = P_p H^T (H P_p H^T + R)^{-1}$$

$$x_m = x_p + \mathrm{K}(\mathrm{z} - \mathrm{H}\, x_p)$$

$$P_m = (I - KH)\, P_p (I - KH)^T + KRK^T$$

Designing a Kalman Filter:

1. Define the System Dynamics and Measurement Models
2. Calculate A, Q, H, R
3. Initialize the state.
4. Iterate between the ***Prediction step*** and the ***Update step***

Note: Within an application, there are typically multiple measurement models

The Kalman Filter: Example 1

System Dynamics:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$x_1(k+1) = 2x_1(k) + x_2(k) + v_1(k)$$
$$x_2(k+1) = 3x_2(k) + x_3(k) + v_2(k)$$
$$x_3(k+1) = 5x_3(k) + v_3(k)$$

$$\boldsymbol{x}(\boldsymbol{k}+\mathbf{1}) = \underbrace{\begin{bmatrix} 2 & 1 & 0 \\ 0 & 3 & 1 \\ 0 & 0 & 5 \end{bmatrix}}_{\text{A}} \boldsymbol{x}(\boldsymbol{k}) + \begin{bmatrix} v_1(k) \\ v_2(k) \\ v_3(k) \end{bmatrix} \longrightarrow Q = \begin{bmatrix} {\sigma_1}^2 & 0 & 0 \\ 0 & {\sigma_2}^2 & 0 \\ 0 & 0 & {\sigma_3}^2 \end{bmatrix}$$

The Kalman Filter: Example 1

Measurement Model

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$m(k+1) = 4x_1(k) + w_m(k)$$

$$\boldsymbol{m}(\boldsymbol{k}+\boldsymbol{1}) = \underbrace{[4 \quad 0 \quad 0]}_{\text{H}}\boldsymbol{x}(\boldsymbol{k}) + w_m(k) \longrightarrow R = std(w_m)$$

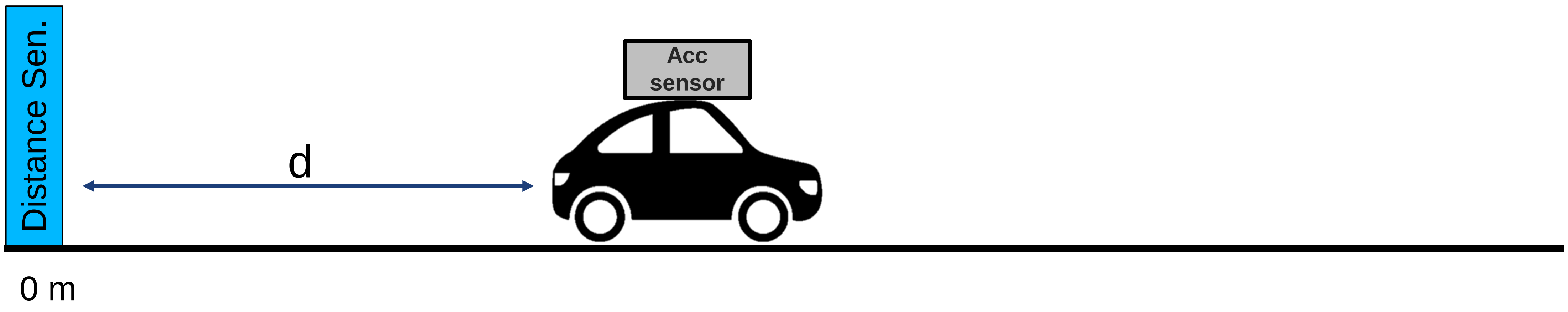
Example 2

Given:

1. The car is equipped with an acceleration sensor
2. A fixed distance sensor measures the distance to the car

Sensors:

- Acceleration sensor: $\boldsymbol{\rho} \sim N(0, \sigma_p)$
- Distance sensor



State space: $\boldsymbol{x} = \binom{x}{v}$

Dynamics:

$$x(k+1) = x(k) + v(k) \cdot \Delta t + \frac{(a(k) + \boldsymbol{\rho}) \cdot \Delta t^2}{2}$$

$$v(k+1) = v(k) + (a(k) + \boldsymbol{\rho}) \cdot \Delta t$$

Example 2

Given:

1. The car is equipped with an acceleration sensor
2. A fixed distance sensor measures the distance to the car

Sensors:

- Acceleration sensor: $\boldsymbol{\rho} \sim N(0, \sigma_p)$
- Distance sensor

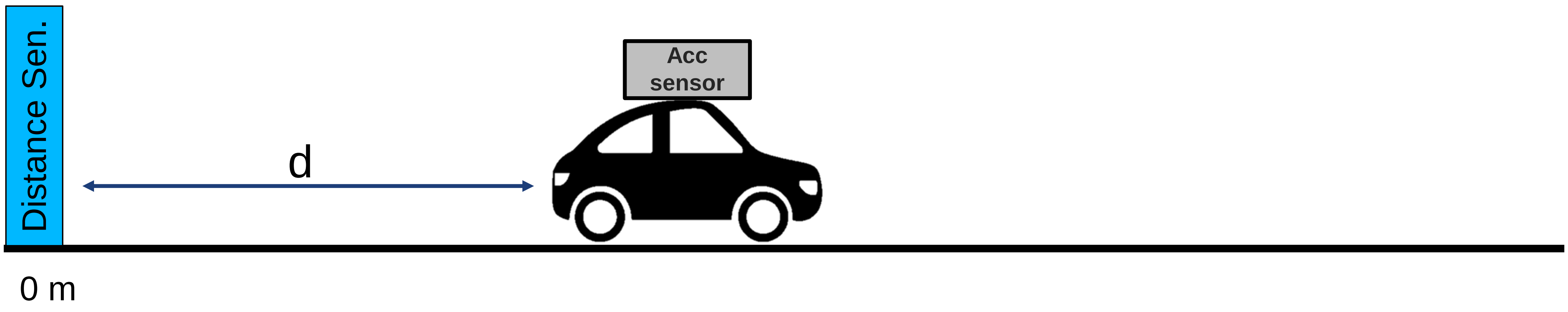
State space: $\boldsymbol{x} = \binom{x}{v}$

Dynamics:

$$x(k+1) = x(k) + v(k) \cdot \Delta t + \frac{a(k) \cdot \Delta t^2}{2} + \frac{\boldsymbol{\rho} \cdot \Delta t^2}{2}$$

$$v(k+1) = v(k) + a(k) \cdot \Delta t + \boldsymbol{\rho} \cdot \Delta t$$

We compute matrix A



Example 2

Given:
1. The car is equipped with an acceleration sensor
2. A fixed distance sensor measures the distance to the car

Sensors:
- Acceleration sensor: $\boldsymbol{\rho} \sim N(0, \sigma_p)$
- Distance sensor

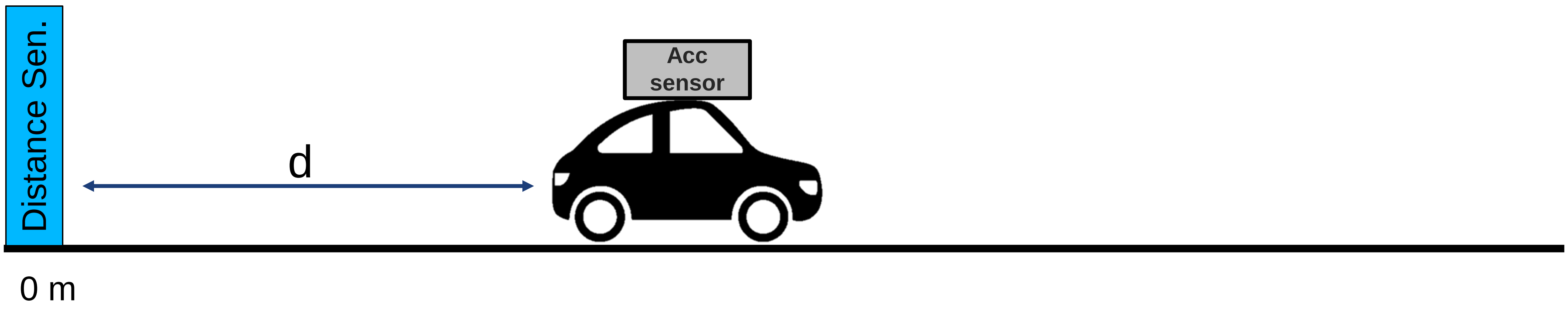
State space: $\boldsymbol{x} = \binom{x}{v}$

Dynamics:

$$x(k+1) = x(k) + v(k) \cdot \Delta t + \frac{a(k) \cdot \Delta t^2}{2} + \frac{\boldsymbol{\rho} \cdot \Delta t^2}{2}$$

$$v(k+1) = v(k) + a(k) \cdot \Delta t + \boldsymbol{\rho} \cdot \Delta t$$

Input



Example 2

Given:
1. The car is equipped with an acceleration sensor
2. A fixed distance sensor measures the distance to the car

Sensors:
- Acceleration sensor: $\boldsymbol{\rho} \sim N(0, \sigma_p)$
- Distance sensor

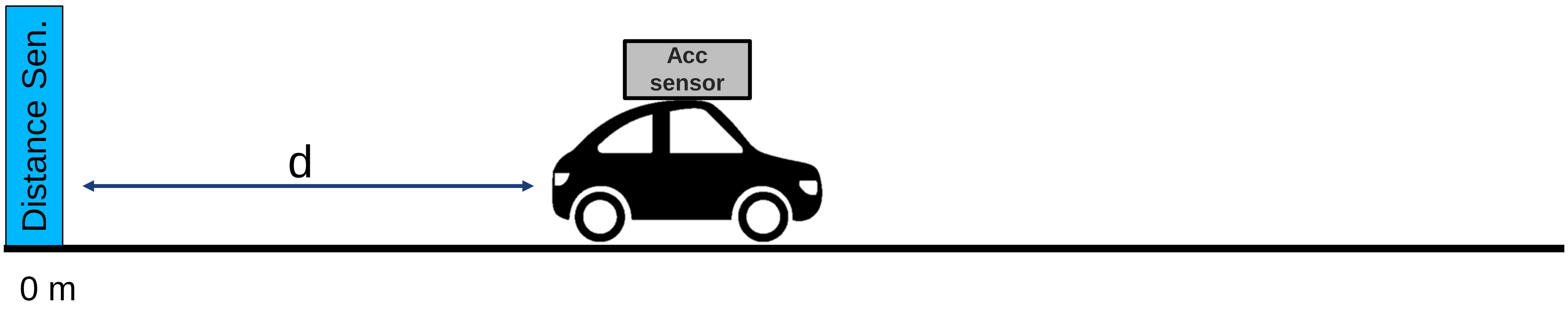
State space: $\boldsymbol{x} = \begin{pmatrix} x \\ v \end{pmatrix}$

Dynamics:

$$x(k+1) = x(k) + v(k) \cdot \Delta t + \frac{a(k) \cdot \Delta t^2}{2} + \frac{\boldsymbol{\rho} \cdot \Delta t^2}{2}$$

$$v(k+1) = v(k) + a(k) \cdot \Delta t + \boldsymbol{\rho} \cdot \Delta t$$

Process noise



Example 2

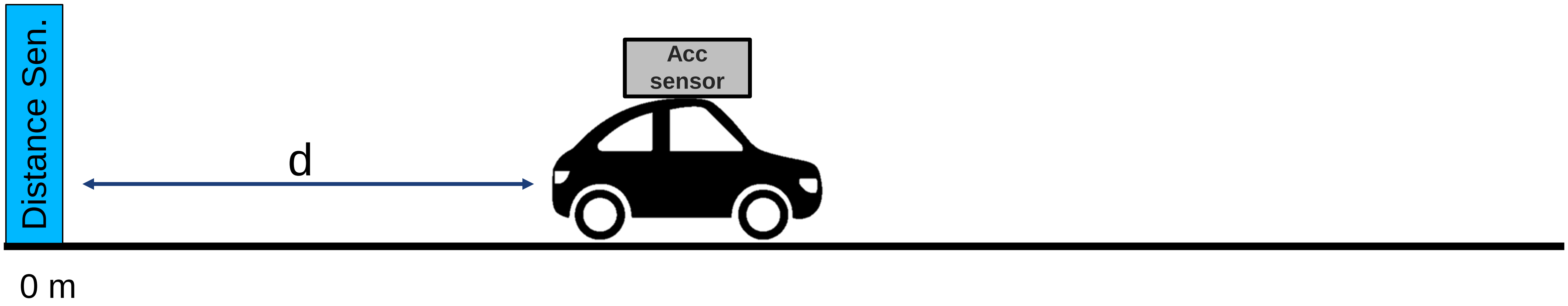
Given:
1. The car is equipped with an acceleration sensor
2. A fixed distance sensor measures the distance to the car

Sensors:
- Acceleration sensor: $\boldsymbol{\rho} \sim N(0, \sigma_p)$
- Distance sensor

State space: $\boldsymbol{x} = \begin{pmatrix} x \\ v \end{pmatrix}$

Dynamics:

$$\boldsymbol{x}(k+1) = \underbrace{\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}}_{\text{A}} \boldsymbol{x}(k) + \underbrace{a(k) \begin{bmatrix} 0.5 \cdot \Delta t^2 \\ \Delta t \end{bmatrix}}_{\text{u}} + \underbrace{\boldsymbol{\rho} \begin{bmatrix} 0.5 \cdot \Delta t^2 \\ \Delta t \end{bmatrix}}_{w_p}$$



Example 2

Given:
1. The car is equipped with an acceleration sensor
2. A fixed distance sensor measures the distance to the car

Sensors:
- Acceleration sensor: $\boldsymbol{\rho} \sim N(0, \sigma_p)$
- Distance sensor

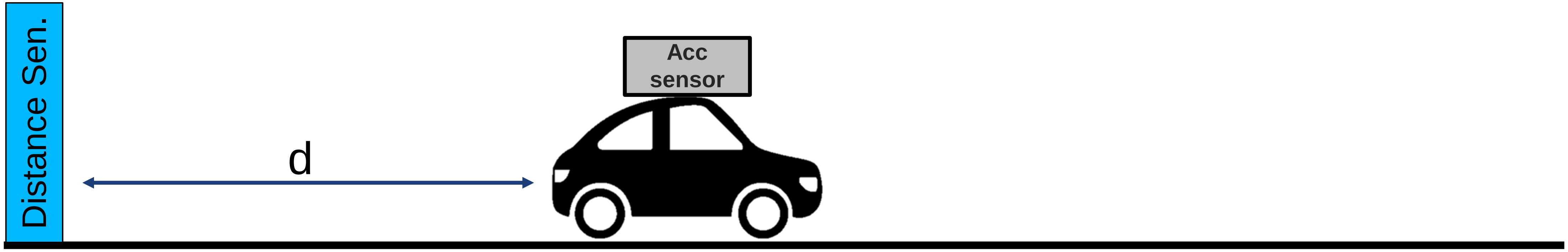
State space: $\boldsymbol{x} = \begin{pmatrix} x \\ v \end{pmatrix}$

Dynamics:

$$\boldsymbol{x}(k+1) = \underbrace{\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}}_{\text{A}} \boldsymbol{x}(k) + \underbrace{a(k) \begin{bmatrix} 0.5 \cdot \Delta t^2 \\ \Delta t \end{bmatrix}}_{\text{u}} + \underbrace{\boldsymbol{\rho} \begin{bmatrix} 0.5 \cdot \Delta t^2 \\ \Delta t \end{bmatrix}}_{w_p}$$

Calculating Q:

1. $\text{Q} = \text{Var}\left(\boldsymbol{\rho} \begin{bmatrix} 0.5 \cdot \Delta t^2 \\ \Delta t \end{bmatrix}\right) = \begin{bmatrix} 0.25 \cdot \Delta t^4 & 0.5 \cdot \Delta t^3 \\ 0.5 \cdot \Delta t^3 & \Delta t^2 \end{bmatrix} {\sigma_p}^2$

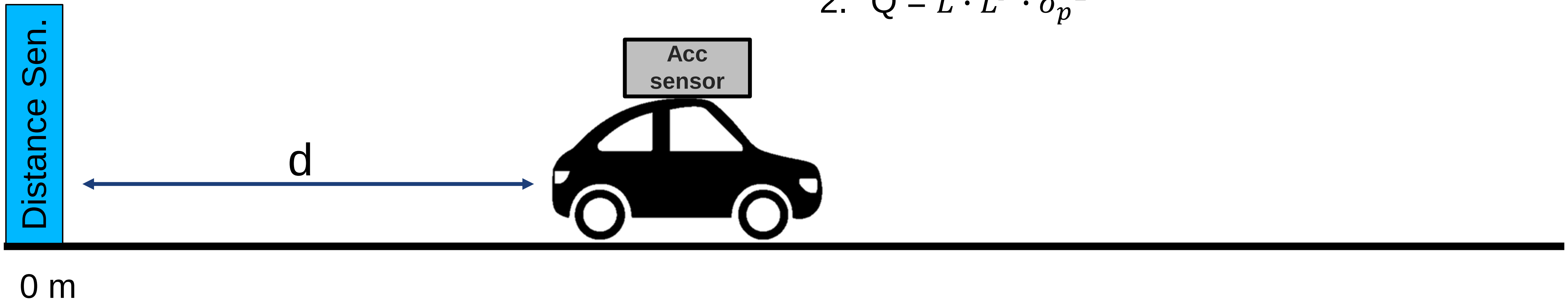


0 m

Example 2

Given:
1. The car is equipped with an acceleration sensor
2. A fixed distance sensor measures the distance to the car

Sensors:
- Acceleration sensor: $\boldsymbol{\rho} \sim N(0, \sigma_p)$
- Distance sensor



State space: $\boldsymbol{x} = \binom{x}{v}$

Dynamics:

$$\boldsymbol{x}(k+1) = \underbrace{\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}}_{\text{A}} \boldsymbol{x}(k) + \underbrace{a(k) \begin{bmatrix} 0.5 \cdot \Delta t^2 \\ \Delta t \end{bmatrix}}_{\text{u}} + \underbrace{\boldsymbol{\rho} \overbrace{\begin{bmatrix} 0.5 \cdot \Delta t^2 \\ \Delta t \end{bmatrix}}^{\text{L}}}_{w_p}$$

Calculating Q:

1. Q = $\mathrm{Var}\left(\boldsymbol{\rho} \begin{bmatrix} 0.5 \cdot \Delta t^2 \\ \Delta t \end{bmatrix}\right) = \begin{bmatrix} 0.25 \cdot \Delta t^4 & 0.5 \cdot \Delta t^3 \\ 0.5 \cdot \Delta t^3 & \Delta t^2 \end{bmatrix} {\sigma_p}^2$
2. Q = $L \cdot L^T \cdot {\sigma_p}^2$

Example 2

Given:
1. The car is equipped with an acceleration sensor
2. A fixed distance sensor measures the distance to the car

Sensors:
- Acceleration sensor: $\boldsymbol{\rho} \sim N(0, \sigma_p)$
- Distance sensor: $\eta \sim N(0, \sigma_m)$

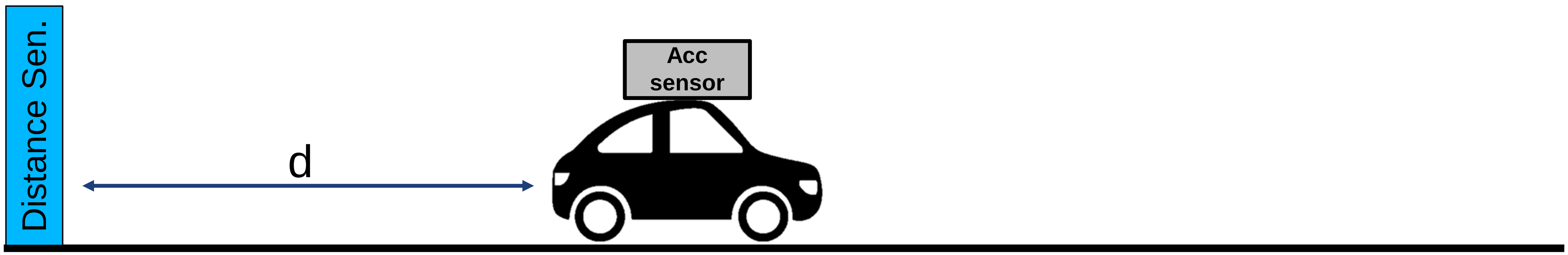
State space: $\boldsymbol{x} = \begin{pmatrix} x \\ v \end{pmatrix}$

Dynamics:

$$\boldsymbol{x}(k+1) = \underbrace{\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}}_{\mathrm{A}} \boldsymbol{x}(k) + \underbrace{a(k) \begin{bmatrix} 0.5 \cdot \Delta t^2 \\ \Delta t \end{bmatrix}}_{\mathrm{u}} + \boldsymbol{\rho} \begin{bmatrix} 0.5 \cdot \Delta t^2 \\ \Delta t \end{bmatrix}$$

Measurement model:

$$d(k) = x(k) + \eta \qquad R = {\sigma_m}^2$$



0 m

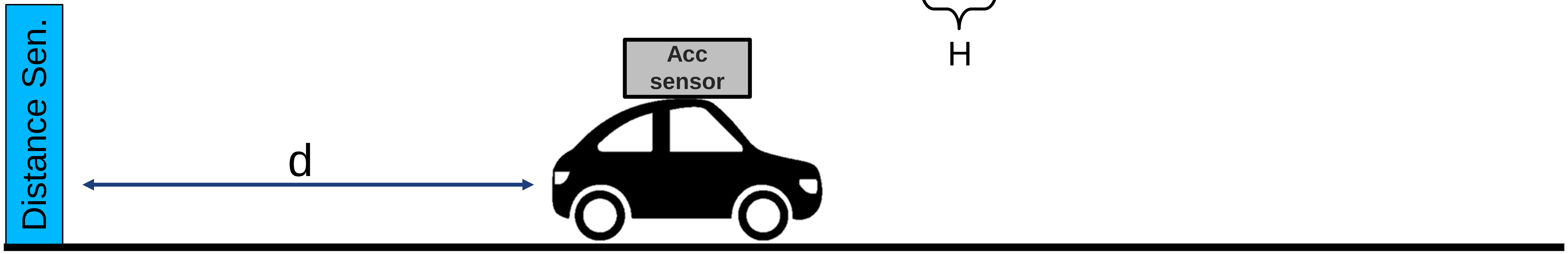
Example 2

Given:

1. The car is equipped with an acceleration sensor
2. A fixed distance sensor measures the distance to the car

Sensors:

- Acceleration sensor: $\boldsymbol{\rho} \sim N(0, \sigma_p)$
- Distance sensor: $\eta \sim N(0, \sigma_m)$



0 m

State space: $\boldsymbol{x} = \begin{pmatrix} x \\ v \end{pmatrix}$

Dynamics:

$$\boldsymbol{x}(k+1) = \underbrace{\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}}_{\text{A}} \boldsymbol{x}(k) + \underbrace{a(k) \begin{bmatrix} 0.5 \cdot \Delta t^2 \\ \Delta t \end{bmatrix}}_{\text{u}} + \boldsymbol{\rho} \begin{bmatrix} 0.5 \cdot \Delta t^2 \\ \Delta t \end{bmatrix}$$

Measurement model:

$$d(k) = \underbrace{[1 \quad 0]}_{\text{H}} \boldsymbol{x}(\boldsymbol{k}) + \eta \qquad R = {\sigma_m}^2$$

The Extended Kalman Filter

The System Dynamics:

$$d(x,v) = x(k+1) = A(k)x(k) + u(k) + w_p$$

The Measurement Model:

$$m(x,w) = z(k+1) = H(k)x(k) + w_m$$

1. Prediction step:

$$A = \frac{\partial d(x_m,0)}{\partial x} \qquad L = \frac{\partial d(x_m,0)}{\partial v}$$

$$x_p = \mathrm{A}x_m + \mathrm{u}$$

$$P_p = \mathrm{A}P_m A^T + LQL$$

2. Update step:

$$H = \frac{\partial m(x_p,0)}{\partial x} \qquad M = \frac{\partial m(x_p,0)}{\partial w}$$

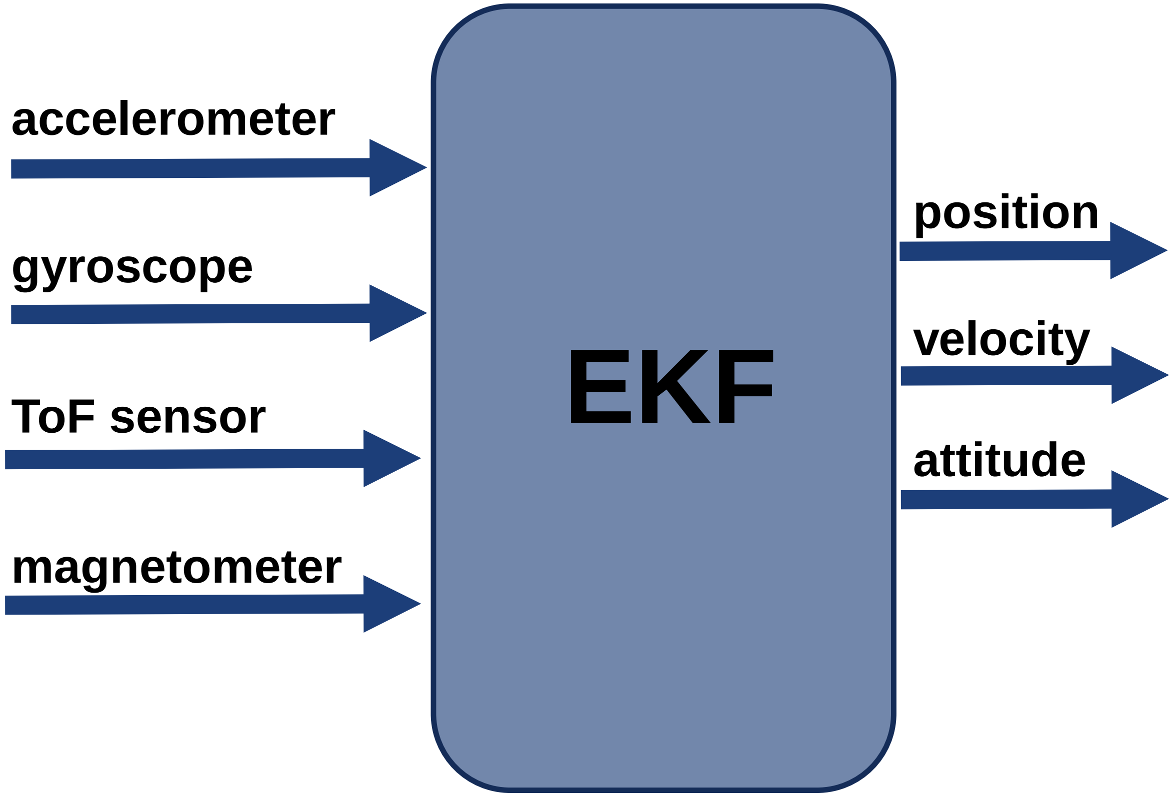
$$K = P_p H^T (H P_p H^T + MRM)^{-1}$$

$$x_m = x_p + \mathrm{K}(\mathrm{z} - \mathrm{H}\, x_p)$$

$$P_m = (I - KH)\, P_p (I - KH)^T + KRK^T$$

The Extended Kalman Filter with Drones

- The accelerometer and gyroscope are typically used for the *Prediction step*.
- To run the *Measurement step*, use the appropriate measurement model (of the corresponding sensor).



Conclusions

- KF and EKF are lightweight implementations which easily allow to fuse a large number of sensors.
- The filter is recursive and it only requires to store the previous state.
- When the Dynamics is Linear and the sensor noise is Gaussian, the KF is the best estimator possible (in the MSE sense).
- Used as estimator in most drone applications.

Hands-on

Download the exercise from the following link and follow the instructions:

https://colab.research.google.com/drive/1rNnwfasbspLDrzsnf8zHzpDwXmxfcerw?usp=sharing