CADi

ROS as a Development Platform

Kalman Filter

Renato Galluzzi

renato.galluzzi@tec.mx

July 2020

Why State Estimation?

- Determination of variables that are not directly measured
- Redundancy
- Filtering

Discrete Probability

- Probability that the random variable X has an outcome x:

$$p(X = x) > 0$$

- In the discrete case:

$$\sum_x p(X = x) = 1$$

- Brief notation:

$$p(X = x) \to p(x)$$

Continuous Probability

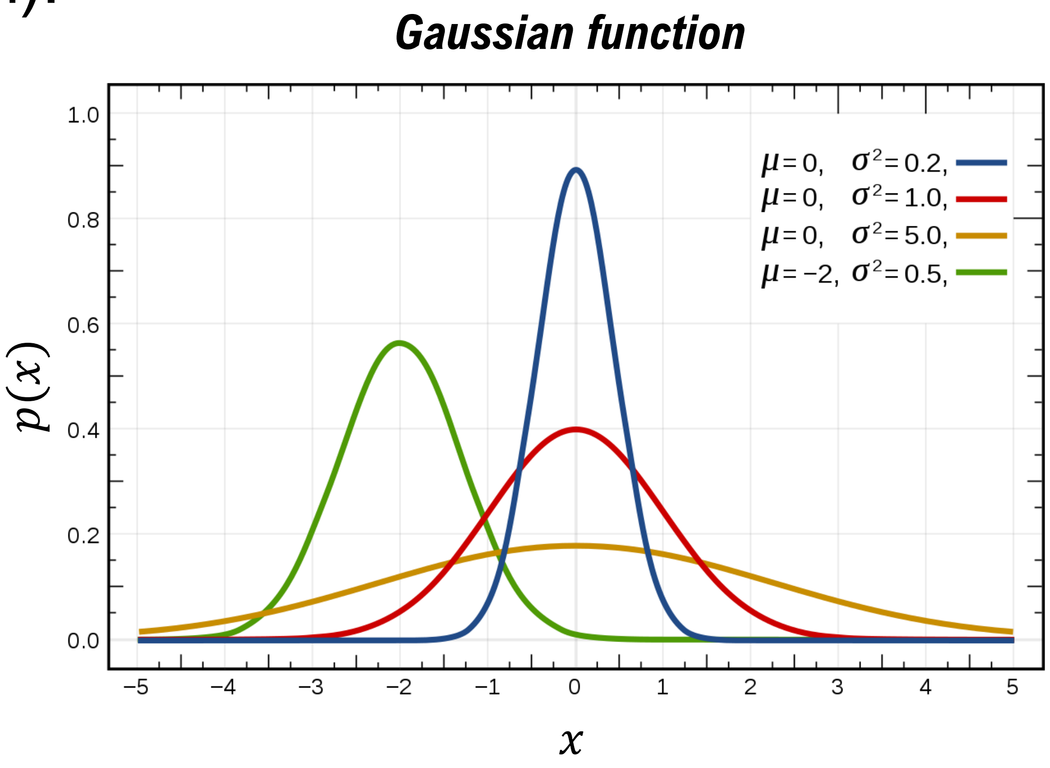
- Normal distribution (prob. density function):

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

  - Mean μ
  - Covariance σ^2

- Similarly to the discrete case:

$$\int p(x)dx = 1$$



Kalman Filter Assumptions

I. State transition

$$x_k = F_k x_{k-1} + G_k u_k + w_k$$

Process uncertainty $[n \times 1]$

State vector $[n \times 1]$

State matrix $[n \times n]$

Input matrix $[n \times m]$

Input vector $[m \times 1]$

- Process uncertainty has *zero* mean and covariance denoted by matrix Q_k.

Multivariate normal distribution

$$\mu = F_k x_{k-1} + G_k u_k$$

$$p(x_k | u_k, x_{k-1}) = \det(2\pi Q_k)^{-1/2} \exp\left[-\frac{1}{2}(x_k - \mu)^T Q_k^{-1}(x_k - \mu)\right]$$

Kalman Filter Assumptions

II. Measurement

$$z_k = H_k x_k + v_k$$

Measurement uncertainty
$[q \times 1]$

Meas. vector
$[q \times 1]$

Meas. matrix
$[q \times n]$

State vector
$[n \times 1]$

- Measurement uncertainty has *zero* mean and covariance denoted by matrix R_k.

Multivariate normal distribution

$$\mu = H_k x_k$$

$$p(z_k | x_k) = \det(2\pi R_k)^{-1/2} \exp\left[-\frac{1}{2}(x_k - \mu)^T R_k^{-1}(x_k - \mu)\right]$$

Kalman Filter Assumptions

III. Initial state

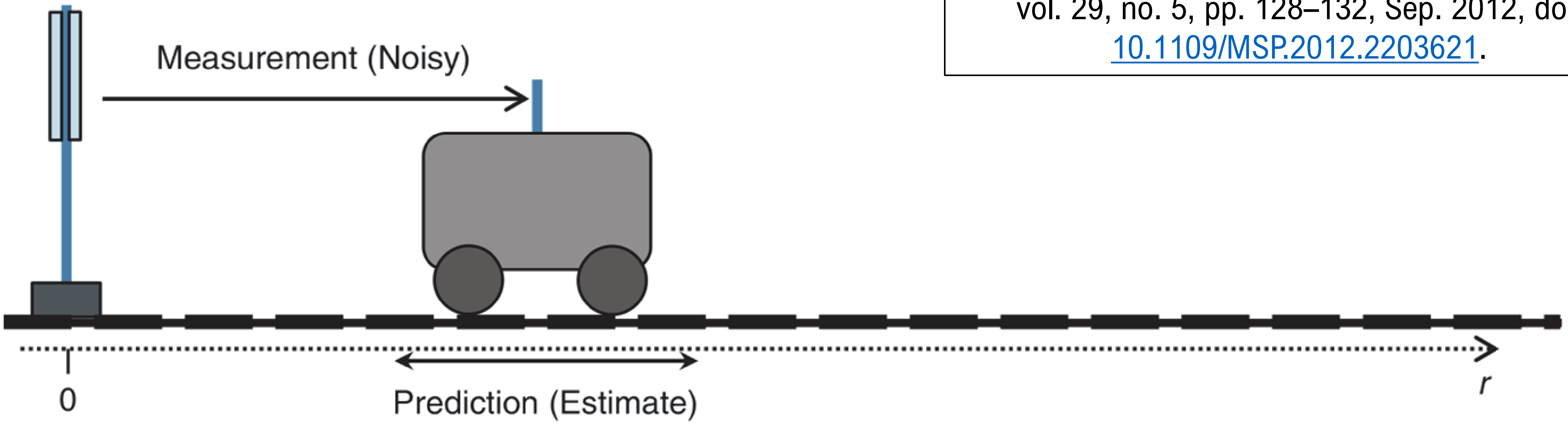
- The initial state probability is given by a normal distribution as well:

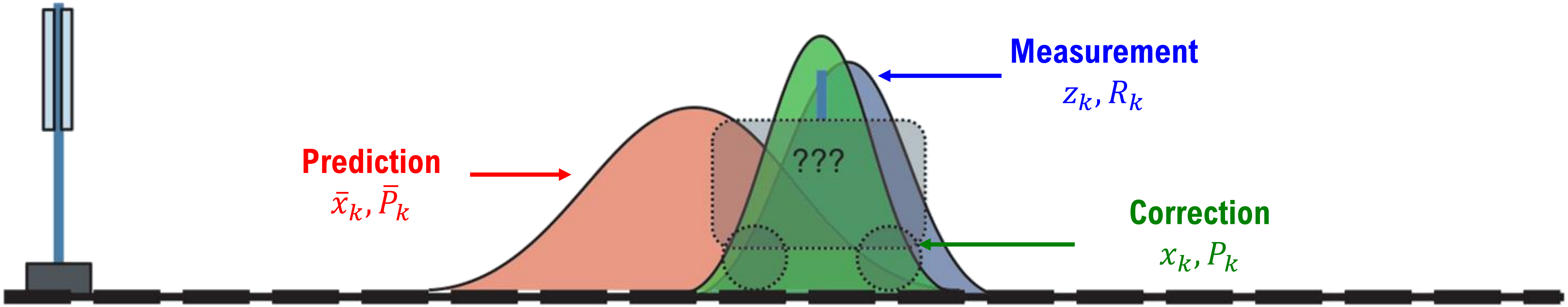
$$p(x_0) = \det(2\pi\Sigma_0)^{-1/2} \exp\left[-\frac{1}{2}(x_k - \mu_0)^T \Sigma_0^{-1} (x_k - \mu_0)\right]$$

- These three assumptions ensure that subsequent estimates will also be Gaussian.

Kalman Filter Algorithm

R. Faragher, “Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation [Lecture Notes],” *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 128–132, Sep. 2012, doi: [10.1109/MSP.2012.2203621](https://doi.org/10.1109/MSP.2012.2203621).





Kalman Filter Algorithm

Kalman_filter($x_{k-1}, P_{k-1}, u_k, z_k$)

$$\bar{x}_k = F_k x_{k-1} + G_k u_k$$
$$\bar{P}_k = F_k P_{k-1} F_k^T + Q_k$$

Prediction: Estimate

- State vector $\bar{x}_k$
- Covariance matrix $\bar{P}_k$

$x_k \rightarrow \mu$
$P_k \rightarrow \sigma^2$

$$K_k = \bar{P}_k H_k^T \left(H_k \bar{P}_k H_k^T + R_k\right)^{-1}$$
$$x_k = \bar{x}_k + K_k(z_k - H_k \bar{x}_k)$$
$$P_k = (I - K_k H_k)\bar{P}_k$$

Correction:

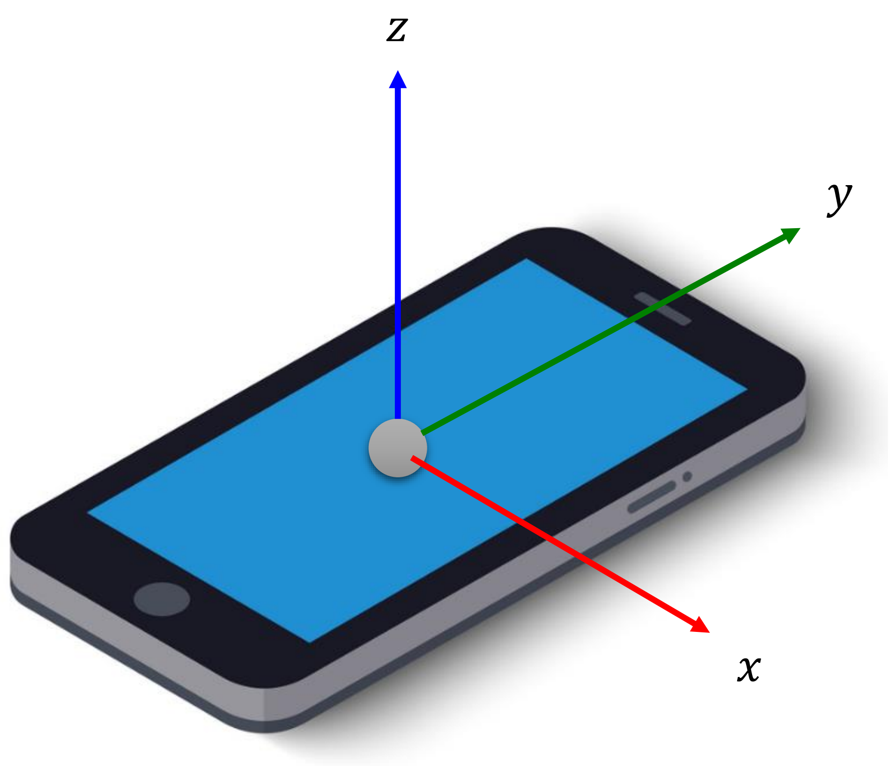
- Compute Kalman gain K_k
- Use K_k to correct $\bar{x}_k, \bar{P}_k$
- The correction of x_k requires the measurement vector z_k

return x_k, P_k

Kalman Filter Example

Estimation of Roll-Pitch-Yaw Angles

Problem Statement



MEMS Accelerometer

$$a_x, a_y, a_z$$

MEMS Gyroscope

$$\omega_x, \omega_y, \omega_z$$

Magnetometer

$$B_x, B_y, B_z$$

Desired output

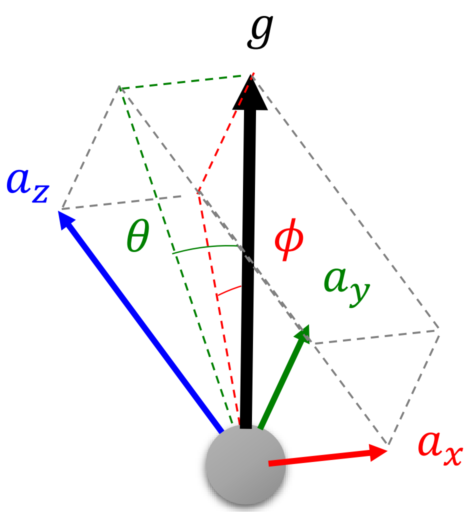
- ϕ rotation around x
- θ rotation around y
- ψ rotation around z

How to obtain them?

Single-Sensor Approach

Accelerometer

- The accelerometer is used as a tilt sensor.
- We find the gravity vector direction from acceleration measurements:



$$\phi = \arctan\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right)$$

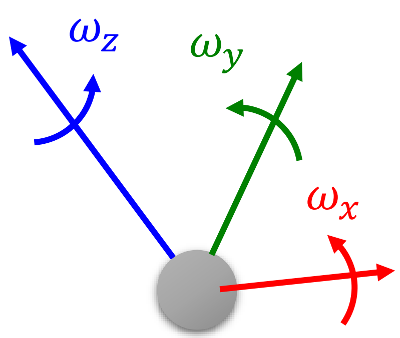
$$\theta = -\arctan\left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}}\right)$$

- Drift-free measurement.
- Yaw angle is unobservable.
- Measurement is sensitive to additional motion or collisions.

Single-Sensor Approach

Gyroscope

- We integrate the angular rate on each axis:



$$\phi = \int \omega_x \mathrm{d}t$$

$$\theta = \int \omega_y \mathrm{d}t$$

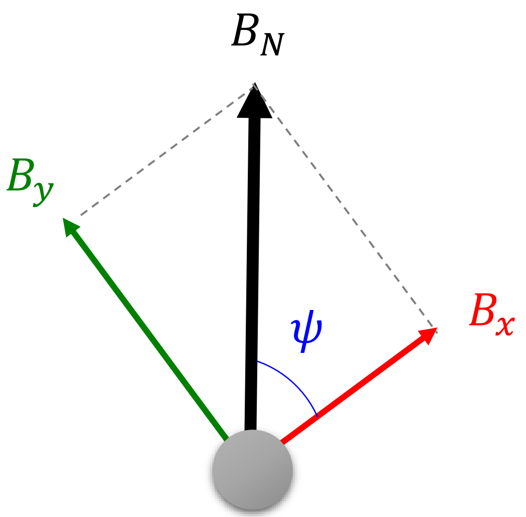
$$\psi = \int \omega_z \mathrm{d}t$$

- Affected by drift.
- Measures the three angles.
- Very clean signal (filtering action).

Single-Sensor Approach

Magnetometer

- If the magnetometer is perfectly flat with respect to the reference:



$$\psi = -\arctan\left(\frac{B_y}{B_x}\right)$$

- Only the yaw angle can be estimated.
- When not perfectly flat, roll and pitch angles are required:

$$\hat{B}_x = B_x \cos\theta + B_y \sin\theta \sin\phi + B_z \sin\theta \cos\phi$$
$$\hat{B}_y = B_y \cos\phi - B_z \sin\phi$$

- The sensor is generally slower than MEMS devices.
- Drift-free, but prone to magnetization.

Sensor Fusion

Idea: Set up an estimator with accelerometer, gyroscope and magnetometer measurements to determine ϕ, θ, ψ.

- Angular speeds provided by gyroscope.
- Angles provided by accelerometer and magnetometer.

Model

$$\omega \rightarrow \theta$$

Integrator Model

$$x_k = F x_{k-1} + w_k$$

$$\begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{bmatrix}_k = \begin{bmatrix} 1 & T_s & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T_s & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{bmatrix}_{k-1}$$

Process Noise Covariance Matrix

Discrete White Noise Acceleration Model

For the single integrator case:

$$x_k = Fx_{k-1} + \Gamma w_k$$

$$F = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}$$

If w_k is a constant acceleration during sample k, the increment in velocity is $w_k T_s$ and the effect in displacement is $w_k T_s^2/2$. Hence

$$\Gamma = \begin{bmatrix} \frac{1}{2}T_s^2 \\ T_s \end{bmatrix} \rightarrow Q = \Gamma\sigma_w^2\Gamma^T = \begin{bmatrix} \frac{1}{4}T_s^4 & \frac{1}{2}T_s^3 \\ \frac{1}{2}T_s^3 & T_s^2 \end{bmatrix}\sigma_w^2, \qquad 0.5\omega_{max} \leq \sigma_w \leq \omega_{max}$$

Measurements

$$z_k = Hx_k + v_k$$

$$H = I_{6\times 6}$$

$$z_k = \begin{bmatrix} \arctan\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right) \\ \omega_x \\ -\arctan\left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}}\right) \\ \omega_y \\ -\arctan\left(\frac{\hat{B}_y}{\hat{B}_x}\right) \\ \omega_y \end{bmatrix}_k$$

Magnetometer measurements are corrected from angles of the previous step.

$$\hat{B}_{x,k} = B_{x,k} \cos \theta_{k-1}$$
$$+B_{y,k} \sin \theta_{k-1} \sin \phi_{k-1}$$
$$+B_{z,k} \sin \theta_{k-1} \cos \phi_{k-1}$$
$$\hat{B}_{y,k} = B_{y,k} \cos \phi_{k-1} - B_{z,k} \sin \phi_{k-1}$$

- R can be a diagonal matrix tuned heuristically
- The sensor datasheet indication is usually a fair starting point

MATLAB Mobile

- Mobile device sensor acquisition feature.
- Sampling frequency up to 100 Hz.
- Exports to MATLAB Drive in .mat (conversion is required).
- Available with your institutional account for Android & iOS.