Attitude estimation with Extended Kalman Filter

Mirko Mazzoleni

Corso di IDENTIFICAZIONE DEI MODELLI E ANALISI DEI DATI

A.A. 2018/2019

Control Automation Lab

UNIVERSITÀ DEGLI STUDI DI BERGAMO

Outline

- Introduction
- Model
- Preprocessing and implementation of EKF

Outline

- **Introduction**
- Model
- Preprocessing and implementation of EKF

Scope of the laboratory

- Use the Kalman Filter to perform attitude estimation of a smartphone, using inertial platform measurements:
    - Gyroscope $\omega(k)$
    - Accelerometer $a(k)$
    - Magnetometer $m(k)$

Reference ➔ Orientation Estimation using Smartphone Sensors – Linkoping university

- LPT: Access SAMSUNG smartphone sensors with the combination: *#0*#

Frames

Reference → Robotica: modellistica, simulazione e controllo
Siciliano, Sciavicco, Villani, Oriolo

- We will use two coordinate systems: the world frame and the sensor frame
- These coordinate systems are related via linear transformations

$$p^{\mathcal{W}} = R^{\mathcal{W}/\mathcal{S}} p^{\mathcal{S}} + t^{\mathcal{W}/\mathcal{S}}$$

where p^S is a point expressed in the sensor frame and p^W is the same point expressed in the world frame

- The relative rotation between the two frames is given by $R^{W/S}$, the **rotation matrix**

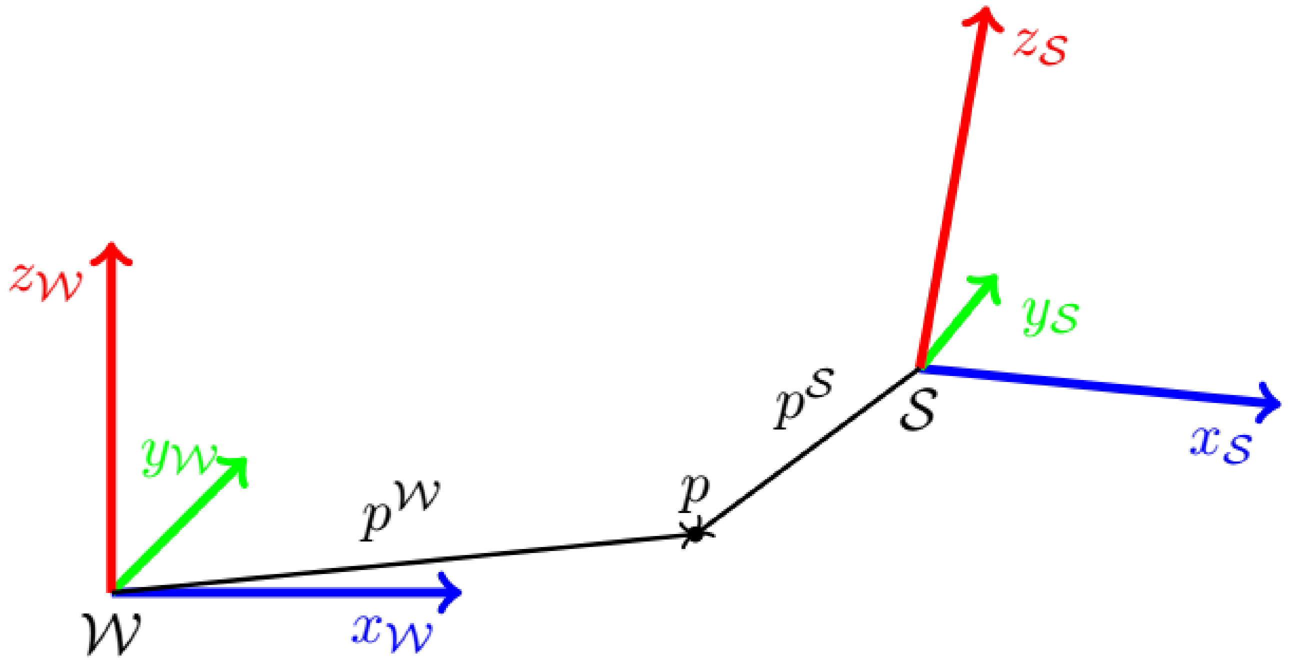
- Since we consider only rotation, the term $t^{W/S}$ is discarded

Frames

- The world frame is fixed

- The sensor frame rotates

- p^S and p^W express the point coordinates in the two frames



- The focus of this lab is to estimate the rotation $\hat{R}(W/S)$ based on the sensors measurements available from a smartphone

- By estimating $\hat{R}(W/S)$ we are able to obtain the object attitude in the fixed frame

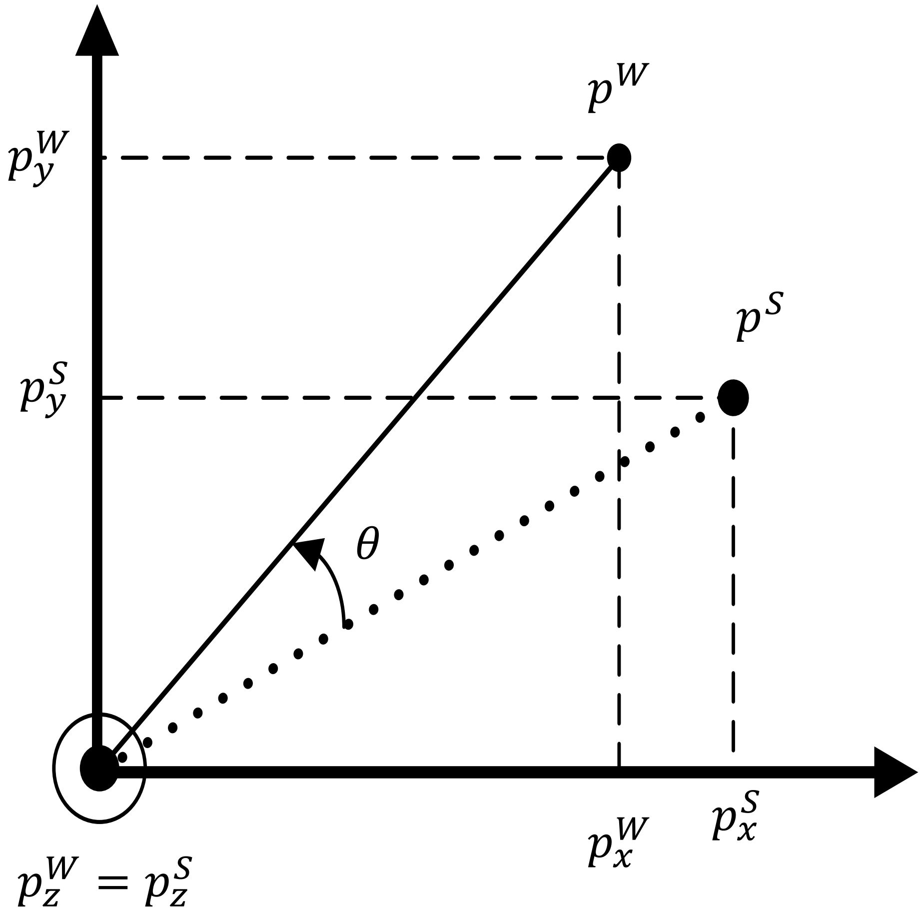
Frames

- A rotation matrix has 3 meanings

- Gives the orientation of one set of coordinates with respect to another

- Represents a transformation of coordinates that relates the coordinates of the same point in two different frames

- Is the operator which rotates a vector in the same frame



Attitude representation

- An object attitude can be expressed via rotation matrices (9 elements)

- The 9 elements are not independent because of orthogonality constraints 3 indenpendent parameters (3 angles)

Euler angles

PROBLEM: Singularities

$R = R_z(\varphi)R_{y'}(\vartheta)R_{z''}(\psi)$ ZYZ representation (current frame)

$R = R_z(\varphi)R_y(\vartheta)R_x(\psi)$ RPY representation (fixed frame)

Different positions in the same have the same representation

Quaternions

- Non minimal representation (4 parameters) but with no singularities

- Represent a rotation of an angle α around the axis $\hat{v}$, $||\hat{v}|| = 1$

$$q = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} \cos(\frac{1}{2}\alpha) \\ \sin(\frac{1}{2}\alpha) \begin{pmatrix} \hat{v}_x \\ \hat{v}_y \\ \hat{v}_z \end{pmatrix} \end{pmatrix} = \begin{pmatrix} \cos(\frac{1}{2}\alpha) \\ \sin(\frac{1}{2}\alpha)\hat{v} \end{pmatrix} \qquad q_o^2 + q_1^2 + q_2^2 + q_3^2 = 1$$

- We have that q and $-q$ represent the same rotation. So the representation of a rotation must be chosen according to the specific application

Rotation matrix corresponding to a quaternion

$$R = Q(q) = \begin{pmatrix} 2q_0^2 - 1 + 2q_1^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & 2q_0^2 - 1 + 2q_2^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & 2q_0^2 - 1 + 2q_3^2 \end{pmatrix}$$

Time derivative of a quaternion

$$\dot{q} = \frac{1}{2} S(\omega) q = \frac{1}{2} \begin{pmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{pmatrix} q$$

Discretize the derivative $\dot{q} = \frac{q(k+1)-q(k)}{T} \Longrightarrow$

$$q_{k+1} = \left(I + \frac{1}{2} S(\omega_k) T\right) q_k$$

UNIVERSITÀ DEGLI STUDI DI BERGAMO
Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Outline

- Introduction
- **Model**
- Preprocessing and implementation of EKF

Simple model

$$x_k = q_k\,, \qquad u_k = \omega_k\,, \qquad y_k = \begin{pmatrix} y_k^{a,T} & y_k^{m,T} \end{pmatrix}^T$$

- The unknown **states** are the quaternion components. We want to estimate them in order to obtain the object attitude which we do not measure (4-d vector)
- **Input:** gyroscope measurements (3-d vector)
- **Outputs:** accelerometers and magnetometer measurements (6-d vector)

Model state space equations

$$
\begin{cases}
q(k+1) = \left[I_4 + \frac{1}{2} S\big(w(k)\big) \cdot T \right] \cdot q(k) + e^q(k) \\
y^a(k) = Q^T\big(q(k)\big) \cdot g^o + e^a(k) \\
y^m(k) = Q^T\big(q(k)\big) \cdot m^o + e^m(k)
\end{cases}
$$

$$
g^o = \begin{pmatrix} 0 \\ 0 \\ 9.81 \end{pmatrix}
$$

$$
m^0 = \begin{pmatrix} 0 & \sqrt{m_x^2 + m_y^2} & m_z \end{pmatrix}^T
$$

Linearization – state space matrix $F_{4\times 4}$

$$q(k+1) = q(k) + \frac{1}{2}T \cdot \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \cdot \begin{bmatrix} q_0(k) \\ q_1(k) \\ q_2(k) \\ q_3(k) \end{bmatrix}$$

$$F = \begin{bmatrix} \partial f_1/\partial x_1 & \dots & \partial f_1/\partial x_n \\ \vdots & & \vdots \\ \partial f_n/\partial x_1 & \dots & \partial f_n/\partial x_n \end{bmatrix}_{\substack{x=\bar{x} \\ u=\bar{u}}} \Longrightarrow F = \begin{bmatrix} 1 & -\frac{1}{2}T\cdot\omega_x & -\frac{1}{2}T\cdot\omega_y & -\frac{1}{2}T\cdot\omega_z \\ \frac{1}{2}T\cdot\omega_x & 1 & \frac{1}{2}T\cdot\omega_z & -\frac{1}{2}T\cdot\omega_y \\ \frac{1}{2}T\cdot\omega_y & -\frac{1}{2}T\cdot\omega_z & 1 & \frac{1}{2}T\cdot\omega_x \\ \frac{1}{2}T\cdot\omega_z & \frac{1}{2}T\cdot\omega_y & -\frac{1}{2}T\cdot\omega_x & 1 \end{bmatrix}$$

Linearization – input matrix $G_{4\times3}$

$$G = \begin{bmatrix} \partial f_1/\partial u_1 & \partial f_1/\partial u_2 & \partial f_1/\partial u_1 \\ \partial f_2/\partial u_1 & \partial f_1/\partial u_2 & \partial f_1/\partial u_2 \\ \partial f_3/\partial u_1 & \partial f_1/\partial u_3 & \partial f_1/\partial u_3 \\ \partial f_4/\partial u_1 & \partial f_1/\partial u_4 & \partial f_1/\partial u_4 \end{bmatrix}$$

$$\Longrightarrow \quad G = \begin{bmatrix} -\frac{1}{2}T\cdot q_1 & -\frac{1}{2}T\cdot q_2 & -\frac{1}{2}T\cdot q_3 \\ \frac{1}{2}T\cdot q_0 & -\frac{1}{2}T\cdot q_3 & \frac{1}{2}T\cdot q_2 \\ \frac{1}{2}T\cdot q_3 & \frac{1}{2}T\cdot q_0 & -\frac{1}{2}T\cdot q_1 \\ -\frac{1}{2}T\cdot q_2 & \frac{1}{2}T\cdot q_1 & \frac{1}{2}T\cdot q_0 \end{bmatrix}$$

Linearization – Output matrix $H_{6\times4}$

$$H = \begin{bmatrix} \partial g_1/\partial x_1 & \cdots & \partial g_1/\partial x_4 \\ \vdots & \ddots & \vdots \\ \partial g_6/\partial x_1 & \cdots & \partial g_6/\partial x_4 \end{bmatrix} \Longrightarrow \quad m_{xy} = \sqrt{m_x^2 + m_y^2} \quad \Longrightarrow$$

$$H = \begin{bmatrix} 9.81 \cdot (-2q_2) & 9.81 \cdot 2q_3 & 9.81 \cdot (-2q_0) & 9.81 \cdot 2q_1 \\ 9.81 \cdot 2q_1 & 9.81 \cdot 2q_0 & 9.81 \cdot 2q_3 & 9.81 \cdot 2q_2 \\ 9.81 \cdot 4q_0 & 0 & 0 & 9.81 \cdot 4q_3 \\ m_{xy} \cdot 2q_3 + m_z \cdot (-2q_2) & m_{xy} \cdot 2q_2 + m_z \cdot 2q_3 & m_{xy} \cdot 2q_1 + m_z \cdot (-2q_0) & m_{xy} \cdot 2q_0 + m_z \cdot 2q_1 \\ m_{xy} \cdot 4q_0 + m_z \cdot 2q_1 & m_z \cdot 2q_0 & m_{xy} \cdot 4q_2 + m_z \cdot 2q_3 & m_z \cdot 2q_2 \\ m_{xy} \cdot (-2q_1) + m_z \cdot 4q_0 & m_{xy} \cdot (-2q_0) & m_{xy} \cdot 2q_3 & m_{xy} \cdot 2q_2 + m_z \cdot 4q_3 \end{bmatrix}$$

Outline

- Introduction
- Model
- **Preprocessing and implementation of EKF**

Steps

1. Check sensors frames on the smarthpone
2. Acquire static measurement to compensate for biases in the sensors
3. Calibrate magnetometer
4. Implement the code