CENG4480

Lecture 08: Kalman Filter

Bei Yu

byu@cse.cuhk.edu.hk

(Latest update: November 8, 2017)

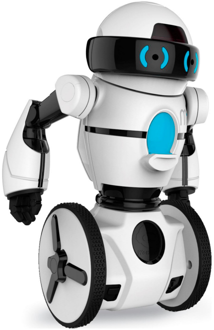
Fall 2017

Overview

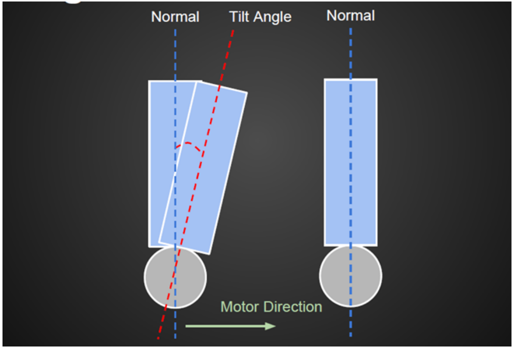
Overview

Self Balance Vehicle / Robot

- http://www.segway.com/
- http://wowwee.com/mip/

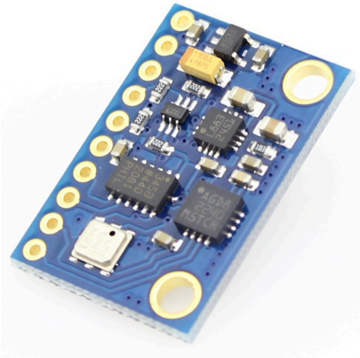


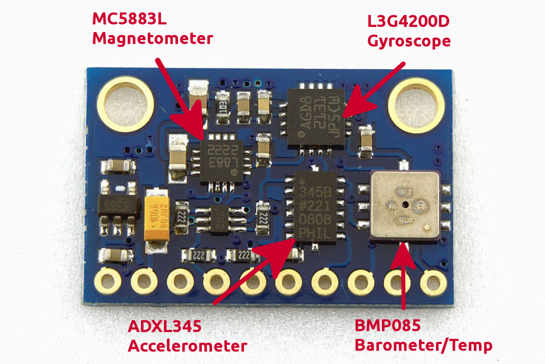
Basic Idea



Motion against the tilt angle, so it can stand upright.

IMU Board



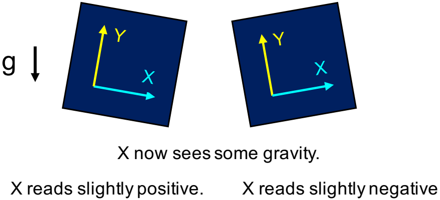


http://www.hotmcu.com/imu-10dof-l3g4200dadxl345hmc5883lbmp180-p-190.html

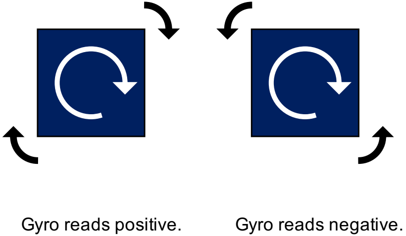
- L3G4200D: gyroscope, measure angular rate (relative value)
- ADXL345: accelerometer, measure acceleration

Overview

Complementary Filter



Accelerometer



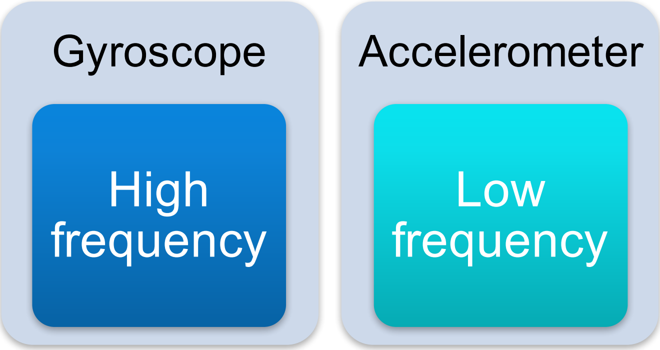
Gyroscope

- Give accurate reading of tilt angle
- Slower to respond than Gyro's
- prone to vibration/noise

- response faster
- but has drift over time

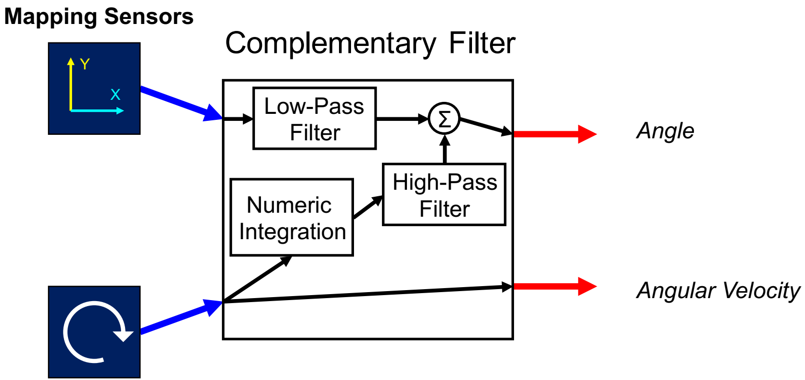
Complementary Filter (cont.)

- Since



- Combine two sensors to find output

Complementary Filter (cont.)




```
Read_acc();
Read_gyro();
Ayz=atan2(RwAcc[1],RwAcc[2])*180/PI;            //angle by accelerometer
Ayz-=offset;                                    //adjust to correct
Angy = 0.98*(Angy+GyroIN[0]*interval/1000)+0.02*Ayz; //complement filter
```

Overview

Rudolf Kalman (1930 – 2016)

- Born in Budapest, Hungary
- BS in 1953 and MS in 1954 from MIT electrical engineering
- PhD in 1957 from Columbia University.

- Famous for his co-invention of the Kalman filter – widely used in control systems to extract a signal from a series of incomplete and noisy measurements.
- Convince NASA Ames Research Center 1960
- Kalman filter was used during Apollo program

Problem Statement

Linear Estimate System

$$\mathbf{x}_t = \mathbf{F}_t \mathbf{x}_{t-1} + \mathbf{B}_t \mathbf{u}_t + \mathbf{w}_t$$

- $\mathbf{x}_t$: state in time t
- $\mathbf{F}_t$: state transition matrix from time $t-1$ to time t
- $\mathbf{u}_t$: input parameter vector at time t
- $\mathbf{B}_t$: control input matrix – apply the effort of $\mathbf{u}_t$
- $\mathbf{w}_t$: process noise, $\mathbf{w}_t \sim N(0, \mathbf{Q}_t)*$

$*\mathbf{w}_t$ assumes zero mean multivariate normal distribution, covariance matrix $\mathbf{Q}_t$

Example of Linear System

Lab6: Angle Measurement

$$\mathbf{x}_t = \mathbf{F}_t \mathbf{x}_{t-1} + \mathbf{B}_t \mathbf{u}_t + \mathbf{w}_t$$

- $\mathbf{x}_t = [x_t, \dot{x}_t]^\top$: x_t is current angle, while $\dot{x}_t$ is current rate
- $\mathbf{F}_t = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix}$
- $\mathbf{B}_t = [\Delta t, 0]^\top$
- $\mathbf{u}_t = [\dot{x}_t, 0]^\top$

Problem Statement (cont.)

System Measurement

$$\mathbf{z}_t = \mathbf{H}_t \mathbf{x}_t + \mathbf{v}_t$$

- $\mathbf{z}_t$: measurement vector
- $\mathbf{H}_t$: transformation matrix mapping state vector to measurement
- $\mathbf{v}_t$: measurement noise, $\mathbf{v}_t \sim N(0, \mathbf{R}_t)\dagger$

†$\mathbf{w}_t$ assumes zero mean multivariate normal distribution, covariance matrix $\mathbf{R}_t$

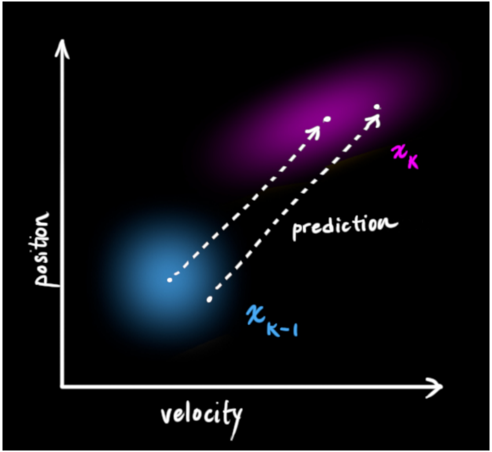
Exercise

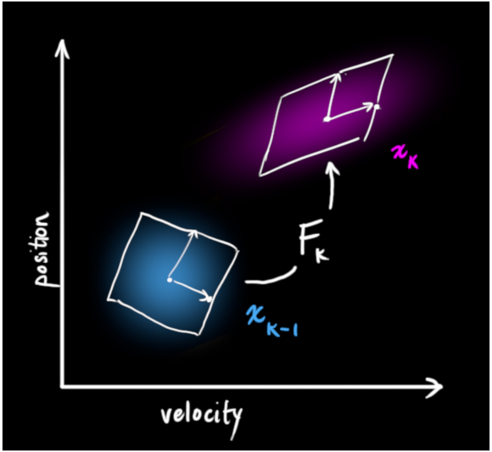
In angle measurement lab, what is the transformation matrix $\mathbf{H}_t$?

$$\mathbf{z}_t = \mathbf{H}_t \mathbf{x}_t + \mathbf{v}_t$$

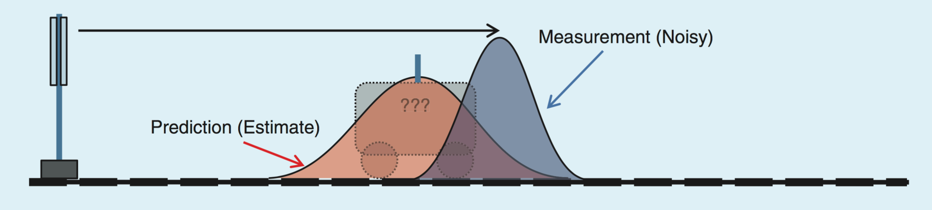
Model with Uncertainty

- Model the measurement w. uncertainty (due to noise $\mathbf{w}_t$)
- $\mathbf{P}_k$: covariance matrix of estimation $\mathbf{x}_t$
- On how much we trust our estimated value – the smaller the more we trust

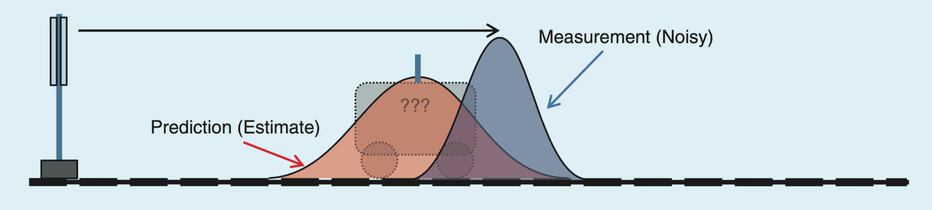


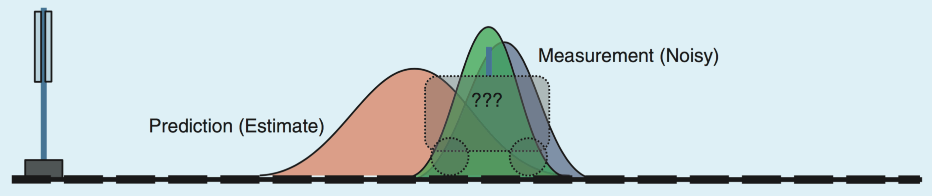


Fuse Gaussian Distributions



Fuse Gaussian Distributions





Exercise

Given two Gaussian functions $y_1(r;\mu_1,\sigma_1)$ and $y_2(r;\mu_2,\sigma_2)$, prove the product of these two Gaussian functions are still Gaussian.

$$y_1(r;\mu_1,\sigma_1) = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(r-\mu_1)^2}{2\sigma_1^2}} \qquad y_2(r;\mu_2,\sigma_2) = \frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{(r-\mu_2)^2}{2\sigma_2^2}}$$

Step 1: Prediction

$$\mathbf{x}_{t|t-1} = \mathbf{F}_t \mathbf{x}_{t-1|t-1} + \mathbf{B}_t \mathbf{u}_t \quad (1)$$

$$\mathbf{P}_{t|t-1} = \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^\top + \mathbf{Q}_t \quad (2)$$

Step 1: Prediction

$$\mathbf{x}_{t|t-1} = \mathbf{F}_t \mathbf{x}_{t-1|t-1} + \mathbf{B}_t \mathbf{u}_t \quad (1)$$

$$\mathbf{P}_{t|t-1} = \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^\top + \mathbf{Q}_t \quad (2)$$

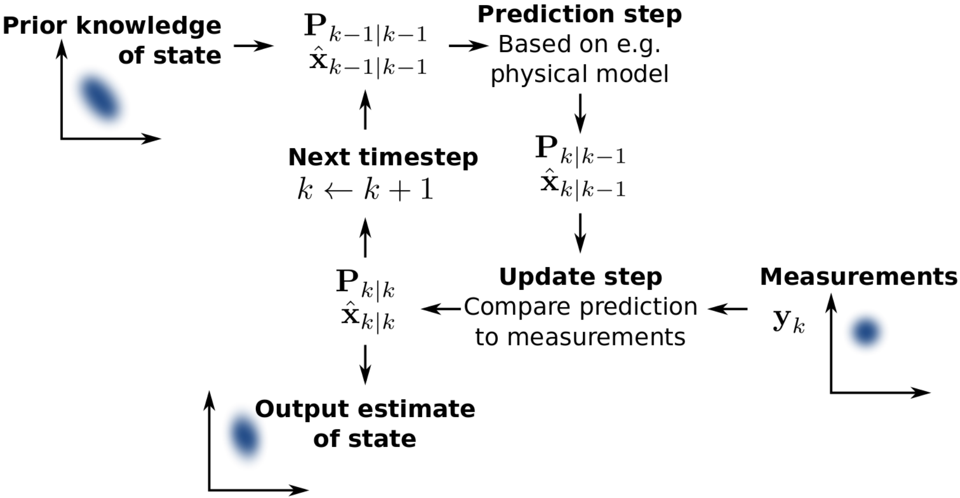
Step 2: Measurement Update

$$\mathbf{x}_{t|t} = \mathbf{x}_{t|t-1} + \mathbf{K}_t (\mathbf{z}_t - \mathbf{H}_t \mathbf{x}_{t|t-1}) \quad (3)$$

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t \mathbf{H}_t \mathbf{P}_{t|t-1} \quad (4)$$

$$\mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{H}_t^\top (\mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^\top + \mathbf{R}_t)^{-1} \quad (5)$$

Basic Concepts



Overview

C Implementation

```c
// Kalman filter module
float Q_angle  =  0.001;
float Q_gyro   =  0.003;
float R_angle  =  0.03;

float x_angle = 0;
float x_bias = 0;
float P_00 = 0, P_01 = 0, P_10 = 0, P_11 = 0;
float dt, y, S;
float K_0, K_1;
```

- **Q**:
- **R**:
- **P**:

C Implementation (cont.)

```
float kalmanCalculate(float newAngle, float newRate,int looptime)
{
    dt = float(looptime)/1000;
    x_angle += dt * (newRate - x_bias);
    P_00 +=  - dt * (P_10 + P_01) + Q_angle * dt;
    P_01 +=  - dt * P_11;
    P_10 +=  - dt * P_11;
    P_11 +=  + Q_gyro * dt;

    y   = newAngle - x_angle;
    S   = P_00 + R_angle;
    K_0 = P_00 / S;
    K_1 = P_10 / S;

    x_angle +=  K_0 * y;
    x_bias  +=  K_1 * y;
    P_00 -= K_0 * P_00;
    P_01 -= K_0 * P_01;
    P_10 -= K_1 * P_00;
    P_11 -= K_1 * P_01;

    return x_angle;
}
```

Summary

- Complementary Filter
- Kalman Filter