Data Filtering for Sensor Fusion in the context of orientation computation

Caus Danu

20/11/2017
MEMS

Microelectromechanical Systems

Include Accelerometer, Magnetometer, Gyroscope and other sensors

Have Hardware Accelerators for filters and sensor fusion algorithms

Courtesy of [1]
Accelerometer

Measures Linear Acceleration + Gravity

Sensor Fusion helps split these components

Raw data will always have a bias because of the Gravity component

Used for computing motion gestures

Courtesy of [2]
Magnetometer

Magnetic Declination: Magnetic north vs Geographic north

Magnetic Inclination: Magnetic Field vector is tilted with respect to the Horizontal axis

Lots of noise from: metal components in the device, battery, other electrical and magnetic components

Used in conjunction with accelerometer to correct the drift of the gyroscope

Courtesy of [3]
Gyroscope

Senses angular velocity using the Coriolis Effect

It has good dynamic response (because of low noise/ the noise is integrated)

Suffers from drift (because of noise integration)

Courtesy of [4]

Courtesy of [5]
Data Filtering

- Using high pass and low pass filters (not state of the art)

![Low Pass](image1.png) ![High Pass](image2.png)

( In digital form of course 😊 )

*Courtesy of [6]*

*Courtesy of [7]*

- Using Kalman Filters (state of the art)
Kalman Filter

**Initial State**
- $X_0$
- $P_0$

**Previous State**
- $X_{k-1}$
- $P_{k-1}$

**Predicted New State**
- $X_{kp} = AX_{k-1} + Bu_k + w_k$
- $P_{kp} = AP_{k-1}A^T + Q_k$

**Kalman Gain**
- $K = \frac{P_{kp}H}{(HP_{kp}H^T + R)}$

**Output of updated state**
- $X_k = X_{kp} + K[Y - HX_{kp}]$
- $P_k = (I - KH)P_{kp}$

**Measurement Input**
- $Y_k = CX_{km} + Z_k$

**Notes**
- $X = $ State Matrix
- $P = $ Process Covariance Matrix (Represents error in the estimate)
- $A, B, C = $ Adaptation Matrices
- $u = $ Control Variable Matrix
- $w = $ Predicted state-noise matrix
- $Q = $ Process Noise Covariance Matrix
- $Z = $ Measurement Noise
- $R = $ Sensor Noise Covariance Matrix (Measurement Error)
- $Y = $ Measurement of the state
- $I = $ Identity Matrix
- $K = $ Kalman Gain
Kalman Filter converges very fast, no matter what the initial estimates were (i.e. initial $X_0$)

Kalman Filter ignores outlier measurement points

*Courtesy of [10]*

*Courtesy of [11]*
The Model

AI Algorithm (Ex: Linear Regression)  Physical Model

Prediction Model

Kalman Filter

Measurement Data
Physical Model for 3D Motion

\[ \mathbf{r} = \mathbf{r}_0 + \mathbf{v}t - \frac{1}{2}\mathbf{a}t^2 \]

\[ \mathbf{X}_k = \begin{bmatrix} 1 & 0 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta T \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta T^2 & 0 & 0 \\ 0 & \frac{1}{2}\Delta T^2 & 0 \\ 0 & 0 & \frac{1}{2}\Delta T^2 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \]

*Courtesy of [12]*
Prediction Model for MEMS chip: Madgwick Filter

Courtesy of [13]

Courtesy of [14]

Courtesy of [13]
Madgwick Filter uses Quaternions

Intuition: Vector in 3D that can also be rotated at an angle theta

\[ Q = \langle \cos(\frac{\theta}{2}), \frac{x\sin(\theta)}{2}, \frac{y\sin(\theta)}{2}, \frac{z\sin(\theta)}{2} \rangle \]

Similar concept to an Axis-Angle rotation

*Courtesy of [15]*
Why Quaternions?

Math is nice for expressing complex orientation!

\[ \mathbf{r} \ast e^{i\theta} \] useful for rotation

\[ Q = w + (i\cdot x + j\cdot y + k\cdot z) \]

Interpolation, addition, subtraction,...
Data Fusion and Integration

Courtesy of [15]
Questions ?
References


[7] RC High Pass Filter. URL http://hyperphysics.phy-astr.gsu.edu/hbase/electric/filcap.html


References


