iMorpheus.ai

Weekly Journal Club

An Introduction to the Kalman Filter

Friday, 2nd February, 2018

Weekly Journal Club

Journal club介绍与自动驾驶中定位方案相关的论文， 主要关注的方向有：SLAM算法、点云数据的处理和压缩、 特征地图、传感器数据处理和融合、 GNSS信号处理等。我们一直关注领域前沿技术， 选取得到广泛认可的、或者是在我们的实际使用中结果比较好的论文， 与大家分享， 共同学习成长。

每周五 北京时间12点
http://imorpheus.ai/journalclub

扫码加入无人驾驶技术群

iMorpheus.ai

Outlines

- 1> Definition and Background
- 2> The Kalman Filter equation and derivation
- 3> The Extended Kalman Filter(EKF)
- 4> A Kalman Filter in Action
- 5> discussion

iMorpheus.ai

Definition and Background

- The Kalman Filter is a set of mathematical equations that provides an efficient computational means to estimate the state of a process.
- R.E. Kalman In 1960 for describing a recursive solution to the discrete-data linear filtering problem.
- The Kalman Filter supports estimations of past , present and future states , particularly in the area of autonomous or assisted navigation.

iMorpheus.ai

The Kalman Filter equation and derivation

The Kalman Filter address the general problem by the linear stochastic difference equation：

State equation $x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$

Measurement equation $z_k = Hx_k + v_k$

Parameter interpretation：

state variables : x_k x_{k-1}

measure variables： z_k

process noise： w_{k-1} (be assumed to be independent, white, and with normal probability distributions $p(w) \sim N(0, Q)$)

measure noise： v_k (be assumed to be independent, white, and with normal probability distributions $p(v) \sim N(0, R)$)

iMorpheus.ai

The Kalman Filter equation and derivation

We define $\bar{x}_k$ to be priori state estimate at step k， and $\hat{x}_k$ to be posteriori state estimate at step， The a priori estimate error covariance is then $\bar{P}_k$, and the a posteriori estimate error covariance is $\hat{P}_k$, then

state equation $\quad \bar{x}_k = A\hat{x}_{k-1} + Bu_{k-1} + w_{k-1}$

measure equation $\quad z_k = Hx_k + v_k$

Using Bayes formula to derivate,then the posteriori state estimate is given as below:

$$\hat{x}_k = \bar{x}_k + K_k(z_k - H\bar{x}_k)$$

$$\hat{P}_k = (I - K_k H)\bar{P}_k$$

iMorpheus.ai

The Kalman Filter equation and derivation

$$K_k = \frac{\bar{P}_k H^T}{H \bar{P}_k H^T + R}$$

- When measurement error covariance R approaches zero,then the gain K weight equals to H^{-1}.
- When the a priori estimate error covariance $\bar{P}_k$ approaches zero, then the gain K weight equals to 0.

The Kalman Filter equation and derivation

- Algorithm implementation

Time Update("Predict")

(1) Project the state ahead

$$\bar{x}_k = A\hat{x}_{k-1} + Bu_{k-1}$$

(2) Project the error covariance ahead

$$\bar{P}_k = A\hat{P}_{k-1}A^T + Q$$

Measurement Update("Correct")

(1) Compute the Kalman gain

$$K_k = \frac{\bar{P}_k H^T}{H\bar{P}_k H^T + R}$$

(1) Update estimate with measurement

$$\hat{x}_k = \bar{x}_k + K_k(z_k - H\bar{x}_k)$$

(1) Update the error covariance

$$\hat{P}_k = (I - K_k H)\bar{P}_k$$

iMorpheus.ai

The Extended Kalman Filter(EKF)

- The Extended Kalman Filter addressed the process which is non-linear.

  state equation $x_k = f(x_{k-1}, u_{k-1}, w_{k-1})$

  measurement equation $z_k = h(x_k, v_k)$

Being similar to Kalman Filter, we assume w_k v_k is zero-mean process, then we get new governing equations that linearize an estimate as below:

$$x_k \approx \tilde{x}_k + A(x_{k-1} - \hat{x}_{k-1}) + W w_{k-1}$$

$$z_k \approx \tilde{z}_k + H(x_k - \tilde{x}_k) + V v_k$$

The Extended Kalman Filter(EKF)

- Algorithm implementation

Time Update("Predict")

(1) Project the state ahead

$$\bar{x} = f(\hat{x}_{k-1}, u_{k-1}, 0)$$

(1) Project the error covariance ahead

$$\bar{P}_k = A_K \hat{P}_{k-1} A_k^T + W_k Q_{k-1} W_k^T$$

Measurement Update("Correct")

(1) Compute the Kalman gain

$$K_k = \frac{\bar{P}_k H_k^T}{H_k \bar{P}_k H_k^T + V_k R_k V_k^T}$$

(1) Update estimate with measurement

$$\hat{x}_k = \bar{x}_k + K_k(z_k - h(\bar{x}_k, 0))$$

(1) Update the error covariance

$$\hat{P}_k = (I - K_k H_k)\bar{P}_k$$

A Kalman Filter in Action

- Experiment equation:

  state equation $\bar{x}_k = \hat{x}_{k-1} + w_{k-1}$

  measurement equation $z_k = \bar{x}_k + v_k$

- We chose a scalar constant x = -0.37727, and assume Q $= 10^{-5}$, $\hat{x}_0$ = 0, $\hat{P}_0 =$ 1.To begin with,we randomly simulate 50 measurements z_k that have error normally distributed arounded zero with a standard deviation of 0.1

iMorpheus.ai

A Kalman Filter in Action

iMorpheus.ai

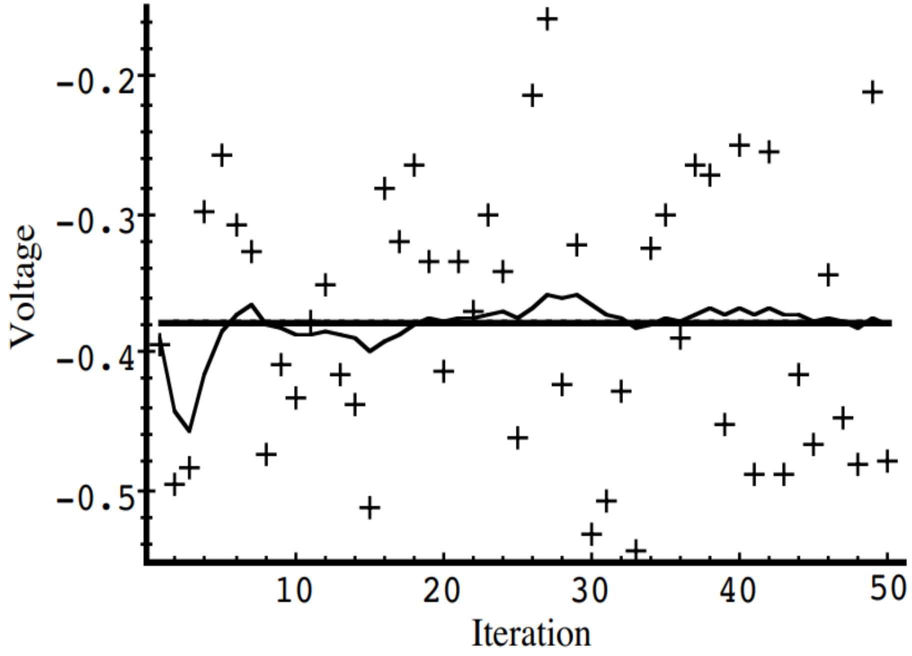

Figure 3-1. The first simulation: $R = (0.1)^2 = 0.01$. The true value of the random constant $x = -0.37727$ is given by the solid line, the noisy measurements by the cross marks, and the filter estimate by the remaining curve.

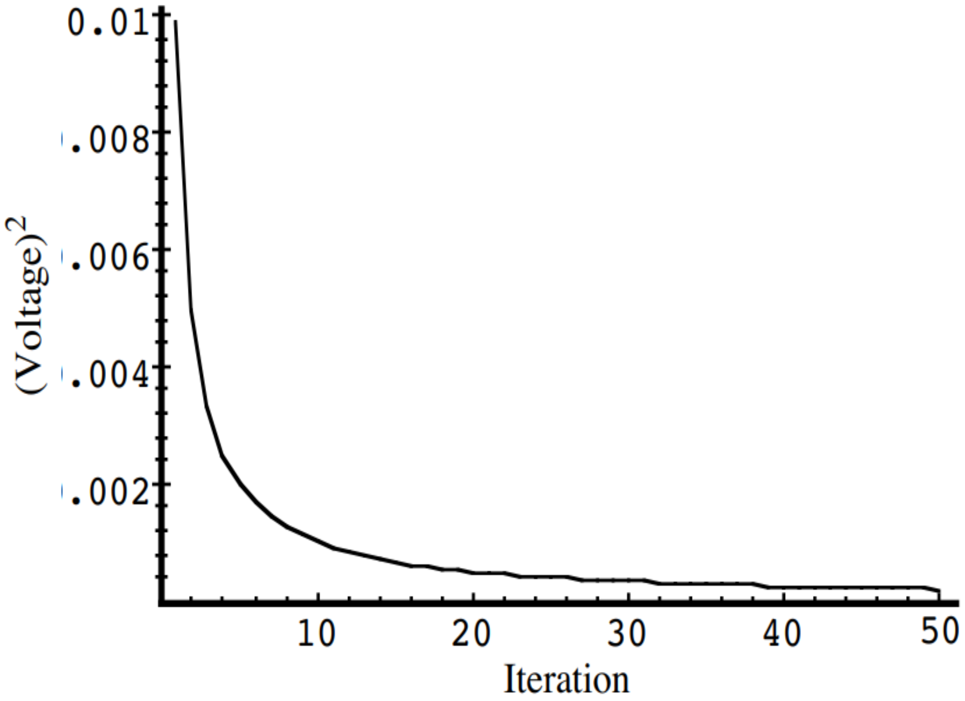

Figure 3-2. After 50 iterations, our initial (rough) error covariance P_k^- choice of 1 has settled to about 0.0002 (Volts^2).

A Kalman Filter in Action

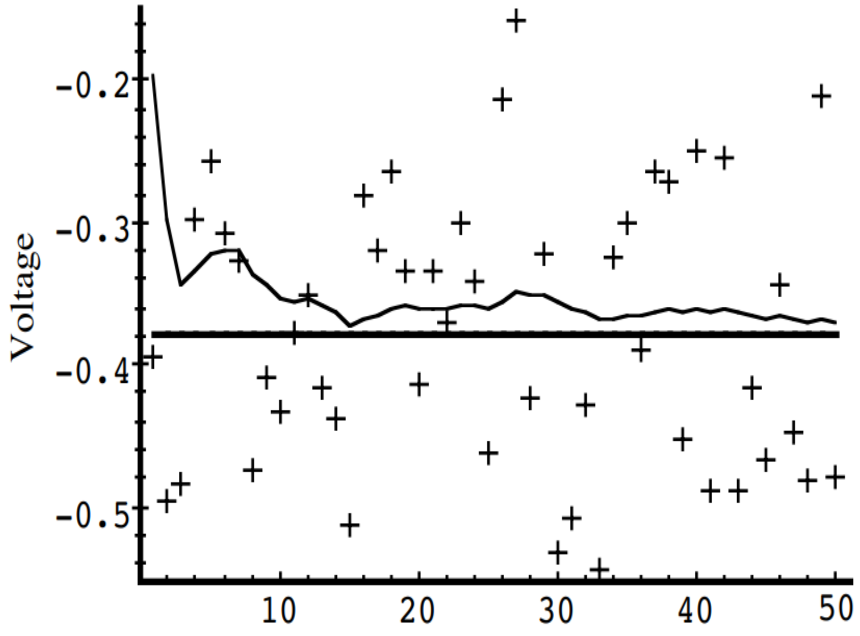

Figure 3-3. Second simulation: $R = 1$. The filter is slower to respond to the measurements, resulting in reduced estimate variance.

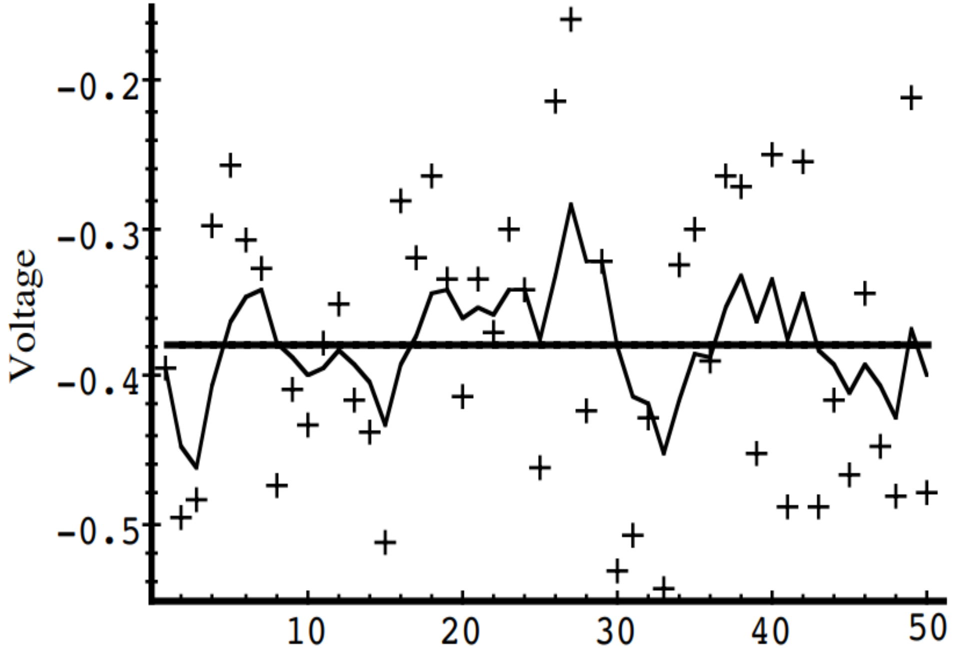

Figure 3-4. Third simulation: $R = 0.0001$. The filter responds to measurements quickly, increasing the estimate variance.

iMorpheus.ai

Discussion

How to get R Q $\hat{x}_0$ $\hat{P}_0$?

Q is the process noise covariance, which is more difficult when we do not have the ability to directly observe the process we are estimating. R is the measurement noise covariance, we usually measured R prior to operation of the filter. $\hat{x}_0$ is our initial estimate value, if $\hat{x}_0$ is correct, we can let $\hat{P}_0 = 0$, else we will let $\hat{P}_0$ with estimate error covariance, then start our filter.

iMorpheus.ai

iMorpheus.ai Weekly Journal Club

Next Friday, 09/02/2018 12:00PM GMT+8

KITTI Odometry Benchmark

关键词：KITTI, 里程计 (Odometry), 测试标准 (Benchmark), 真实数据 (Ground Truth)

Website : **http://imorpheus.ai**
Email Address : **live@imorpheus.ai**

扫码加入无人驾驶技术群

iMorpheus.ai