Kalman filter notes

March 5, 2018

Kjeld Jensen, MSc, PhD
Associate Professor
Email: kjen@mmmi.sdu.dk

SDU UAS Center
Faculty of Engineering
University of Southern Denmark
Campusvej 55
5230 Odense M
Denmark

Please notice that you may download an updated version of this document at:

http://kjen.dk/download/kalman_filter_notes.pdf

Contents

Introduction

The purpose of this paper is to give a brief introduction to Kalman Filters and their applications within field robot state estimation. The current paper is a draft version, please email me any comments or questions.

Kalman Filters

The state estimator estimates the current vehicle position and orientation as well as other relevant state variables that together describe the field robot state.

A Kalman Filter (KF) is typically used as state estimator. The KF estimates the state for a dynamical system based on a model of the system and measurements from one or more sensors [**?**, **?**, **?**]. The KF algorithm uses knowledge about the system and measurement noise to optimize the state estimation. The KF is a recursive algorithm with two elements:

- A prediction (time update) where the estimate from the previous time step is used to calculate an estimate of the current state.
- A correction (measurement update) where measurement data from sensors are used to update the predicted state.

By recursive is ment that the Kalman Filter does not need to archive and process a history of measurement data during the correction.

The following describes a discrete time Kalman Filter where $\mathbf{x}_k$ is the discrete $\mathbf{x}(t)$ sampled at the time $k\,T$, where T is the sampling time.

System model

The system model describes how the system state $\mathbf{x}$ evolves as a function of time:

$$\mathbf{x}_k \quad = \quad \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{v}_{k-1}) \tag{1}$$

Where the function $\mathbf{f}$ relates the state at the previous time step $k-1$ to the current time step k, $\mathbf{u}$ represents input to the system and $\mathbf{v}$ represents the system random noise with covariance matrix $\mathbf{Q}$. The KF uses the system model during the prediction to estimate the system state.

Measurement model

The measurement model is based on models of the individual sensors, that describe the measurement data $\mathbf{z}$ as a function of the system state $\mathbf{x}$:

$$\mathbf{z}_k \quad = \quad \mathbf{h}(\mathbf{x}_k, \mathbf{w}_k) \tag{2}$$

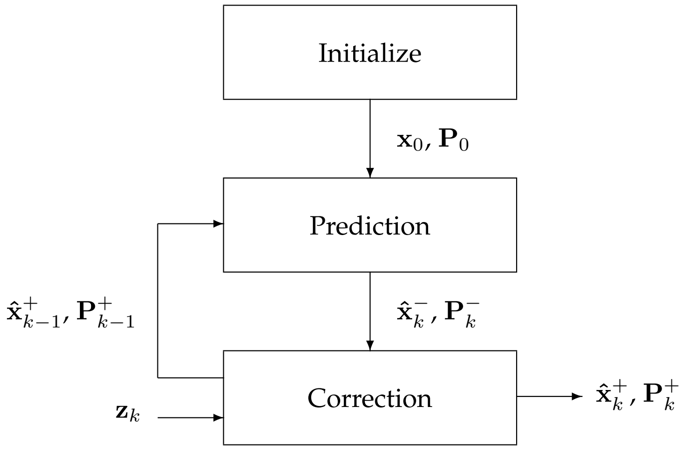

Figure 1: Kalman Filter algorithm

Where the function $\mathbf{h}$ relates the system state $\mathbf{x}_k$ to the measurement $\mathbf{z}_k$, $\mathbf{w}$ represents the measurement noise with covariance matrix $\mathbf{R}$. The KF uses the measurement model during the correction to update the system state estimate when new measurement data are available.

Assumptions

The KF assumes that the system and measurement models are perturbed by noise that is independent, white and normal distributed with zero mean and covariance matrix $\mathbf{Q}$ and $\mathbf{R}$. KF assumes further that the estimate can be represented by a Gaussian.

$$\begin{aligned} \mathbf{v} &\in NID(0, \mathbf{Q}) \\ \mathbf{w} &\in NID(0, \mathbf{R}) \end{aligned} \tag{3}$$

Kalman Filter algorithm

Figure 1 shows the recursive KF.

A scalar Kalman filter

Initialization

$$\hat{x}_0 = c_1 \tag{4}$$

$$P_0 = c_2 \neq 0 \tag{5}$$

Prediction

$$\hat{x}_k^- = \hat{x}_{k-1} + u_k + v_k \quad (6)$$

$$P_k^- = P_{k-1} + Q_k \quad (7)$$

$v_k \equiv 0$ since the system noise is assumed to be normal distributed with zero mean.

Correction

$$z_k = h(x_k) + w_k \quad (8)$$

$$K_k = \frac{P_k^-}{P_k^- + R_k} \quad (9)$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(z_k - \hat{x}_k^-) \quad (10)$$

$$P_k^+ = P_k^-(1 - K_k) \quad (11)$$

$w_k \equiv 0$ since the measurement noise is assumed to be normal distributed with zero mean.

Pseudocode

The pseudocode 1 shows an implementation example of a scalar KF for a field robot navigating in-row in row crops or orchards using a gyro and a lidar (laser range scanner):

Algorithm 1 Scalar Kalman Filter example

Input
T
$gyroVelocity$
$gyroVar$
$lidarAngle$
$lidarVar$

Output
$estAngle$
$estVar$

Initialization
$estAngle = 0$
$estVar = 3.14$
$gyroVarAccumulated = 0$

loop
  Prediction
  if new gyro data available **then**
    $predAngle = estAngle + gyroVelocity * T$
    $gyroVarAccumulated = gyroVarAccumulated + gyroVar$
    $predVar = estVar + gyroVarAccumulated * T$
    $estAngle = predAngle$
    $estVar = predVar$
  end if

  Correction
  if new lidar data available **then**
    $K = \frac{predVar}{predVar + lidarVar}$
    $corrAngle = predAngle + K * (lidarAngle - predAngle)$
    $corrVar = predVar * (1 - K)$
    $estAngle = corrAngle$
    $estVar = corrVar$
    $gyroVarAccumulated = 0$
  end if
end loop