Understanding and Applying Kalman Filtering

Lindsay Kleeman

Department of Electrical and Computer Systems Engineering

Monash University, Clayton
Introduction

Objectives:

1. Provide a basic understanding of Kalman Filtering and assumptions behind its implementation.
2. Limit (but cannot avoid) mathematical treatment to broaden appeal.
3. Provide some practicalities and examples of implementation.
4. Provide C++ software overview.
What is a Kalman Filter and What Can It Do?

A Kalman filter is an optimal estimator - ie infers parameters of interest from indirect, inaccurate and uncertain observations. It is recursive so that new measurements can be processed as they arrive. (cf batch processing where all data must be present).

Optimal in what sense?

If all noise is Gaussian, the Kalman filter minimises the mean square error of the estimated parameters.
What if the noise is NOT Gaussian?

Given only the mean and standard deviation of noise, the Kalman filter is the best linear estimator. Non-linear estimators may be better.

Why is Kalman Filtering so popular?

- Good results in practice due to optimality and structure.
- Convenient form for online real time processing.
- Easy to formulate and implement given a basic understanding.
- Measurement equations need not be inverted.
**Word examples:**

- Determination of planet orbit parameters from limited earth observations.
- Tracking targets - eg aircraft, missiles using RADAR.
- Robot Localisation and Map building from range sensors/ beacons.

**Why use the word “Filter”?**

The process of finding the “best estimate” from noisy data amounts to “filtering out” the noise.

However a Kalman filter also doesn’t just clean up the data measurements, but also *projects* these measurements onto the state estimate.
What is a Covariance Matrix?

The covariance of two random variables $x_1$ and $x_2$ is

$$\text{cov}(x_1, x_2) \equiv E[(x_1 - \bar{x}_1)(x_2 - \bar{x}_2)]$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x_1 - \bar{x}_1)(x_2 - \bar{x}_2) p(x_1, x_1) dx_1 dx_2$$

$$\equiv \sigma_{x_1 x_2}^2$$

where $p$ is the joint probability density function of $x_1$ and $x_2$.

The correlation coefficient is the normalised quantity

$$\rho_{12} \equiv \frac{\sigma_{x_1 x_2}^2}{\sigma_{x_1} \sigma_{x_2}}, \quad -1 \leq \rho_{12} \leq +1$$
The covariance of a column vector \( x = [x_1 \ldots x_n]' \) is defined as

\[
\text{cov}(x) \equiv E[(x - \overline{x})(x - \overline{x})']
\]

\[
= \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} (x - \overline{x})(x - \overline{x})' p(x) \, dx_1 \ldots dx_n
\]

\[
\equiv P_{xx}
\]

and is a symmetric \( n \) by \( n \) matrix and is positive definite unless there is a linear dependence among the components of \( x \).

The \((i,j)\)th element of \( P_{xx} \) is \( \sigma^2_{x_ix_j} \).

Interpreting a covariance matrix:

- diagonal elements are the variances, off-diagonal encode correlations.
**Diagonalising a Covariance Matrix**

cov(\( x \)) is symmetric \( \Rightarrow \) can be **diagonalised** using an **orthonormal** basis.

By changing coordinates (pure rotation) to these unity orthogonal vectors we achieve **decoupling** of error contributions.

The basis vectors are the eigenvectors and form the axes of **error ellipses**.

The lengths of the axes are the square root of the eigenvalues and correspond to standard deviations of the **independent** noise contribution in the direction of the eigenvector.

Example: Error ellipses for mobile robot odometry derived from covariance matrices:
Error Ellipses corresponding to 50 standard deviations
10000 Monte-Carlo runs for $k_L = k_R = 10^{-3} \, m^{\frac{1}{2}}$, $B=0.5 \, m$

<table>
<thead>
<tr>
<th></th>
<th>Means</th>
<th>Covariance Matrix</th>
<th>Stand dev/ Corr Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Theoretical results</strong></td>
<td>0</td>
<td>3.032e-5 -4.763e-5 -2.817e-5</td>
<td>0.005506 -0.913208 -0.8667</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>-4.763e-5 8.974e-5 4.700e-5</td>
<td>-0.9132 0.009473 0.8404</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>-2.817e-5 4.700e-5 3.4849e-5</td>
<td>-0.8667 0.8404 0.005903</td>
</tr>
<tr>
<td><strong>Run 1 10000 samples</strong></td>
<td>-1.997e-5 5.321e-5 3.184e-5</td>
<td>2.980e-5 -4.651e-5 2.761e-5</td>
<td>0.005459 -0.9130 -0.8627</td>
</tr>
<tr>
<td></td>
<td>5.321e-5</td>
<td>-4.651e-5 8.708e-5 4.585e-5</td>
<td>-0.9130 0.009332 0.8380</td>
</tr>
<tr>
<td></td>
<td>3.184e-5</td>
<td>-2.761e-5 4.585e-5 3.437e-5</td>
<td>0.8627 0.8380 0.005862</td>
</tr>
</tbody>
</table>
Formulating a Kalman Filter Problem

We require \textit{discrete time linear dynamic system} description by vector difference equation with \textit{additive white noise} that models unpredictable disturbances.

\textbf{STATE DEFINITION} - the state of a deterministic dynamic system is the smallest vector that summarises the past of the system in full.

Knowledge of the state allows theoretically prediction of the future (and prior) dynamics and outputs of the deterministic system in the absence of noise.
STATE SPACE REPRESENTATION

State equation:
\[ x(k + 1) = F(k)x(k) + G(k)u(k) + v(k) \quad k = 0,1, \ldots \]
where \( x(k) \) is the \( n_x \) dimensional state vector, \( u(k) \) is the \( n_u \) dimensional known input vector, \( v(k) \) is (unknown) zero mean white process noise with covariance
\[ E[v(k)v(k)'] = Q(k) \]

Measurement equation:
\[ z(k) = (k)(k) + (k) \quad k = 1, \ldots \]
\( w(k) \) is unknown zero mean white measurement noise with known covariance
\[ E[w(k)w(k)'] = R(k) \]
FALLING BODY EXAMPLE

Consider an object falling under a constant gravitational field. Let $y(t)$ denote the height of the object, then

$$y(t) = -g$$

$$\Rightarrow y(t) = y(t_0) - g(t - t_0)$$

$$\Rightarrow y(t) = y(t_0) + y(t_0)(t - t_0) - \frac{g}{2}(t - t_0)^2$$

As a discrete time system with time increment of $t-t_0=1$
\[ y(k+1) = y(k) + \dot{y}(k) - \frac{g}{2} \]

the height \( y(k+1) \) depends on the previous velocity and height at time \( k \).

We can define the state as

\[ x(k) \equiv [y(k) \quad \dot{y}(k)]' \]

and then the state equation becomes
\[
\begin{align*}
\mathbf{x}(k+1) &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}(-g) \\
&= \mathbf{F} \mathbf{x}(k) + \mathbf{G} \mathbf{u}
\end{align*}
\]

Assuming we observe or measure the height of the ball directly. The measurement equation is:

\[
\begin{align*}
\mathbf{z}(k) &= [1 \ 0] \mathbf{x}(k) + \mathbf{w}(k) \\
&= \mathbf{H} \mathbf{x}(k) + \mathbf{w}(k)
\end{align*}
\]

The variance of \( \mathbf{w}(k) \) needs to be known for implementing a Kalman filter.

Given the initial state and covariance, we have sufficient information to find the optimal state estimate using the Kalman filter equations.
Kalman Filter Equations

The Kalman filter maintains the estimates of the state:

\[
\hat{x}(k|k) \quad - \text{estimate of } x(k) \text{ given measurements } z(k), \ z(k-1),\ldots
\]

\[
\hat{x}(k+1|k) \quad - \text{estimate of } x(k+1) \text{ given measurements } z(k), \ z(k-1),\ldots
\]

and the error covariance matrix of the state estimate

\[
P(k|k) \quad - \text{covariance of } x(k) \text{ given } z(k), \ z(k-1),\ldots
\]

\[
P(k+1|k) \quad - \text{estimate of } x(k+1) \text{ given } z(k), \ z(k-1),\ldots
\]

We shall partition the Kalman filter recursive processing into several simple stages with a physical interpretation:
State Estimation

0. Known are $\hat{x}(k|k)$, $u(k)$, $P(k|k)$ and the new measurement $z(k+1)$.

1. State Prediction
   $$\hat{x}(k+1|k) = F(k)\hat{x}(k|k) + G(k)u(k)$$ \textit{Time update}

2. Measurement Prediction:
   $$\hat{z}(k+1|k) = H(k)\hat{x}(k+1|k)$$ \textit{measurement update}

3. Measurement Residual:
   $$v(k+1) = z(k+1) - \hat{z}(k+1|k)$$

4. Updated State Estimate:
   $$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k+1)v(k+1)$$

where $W(k+1)$ is called the Kalman Gain defined next in the state covariance estimation.
State Covariance Estimation

1. State prediction covariance: \( P(k+1|k) = F(k)P(k|k)F(k)'+Q(k) \)

2. Measurement prediction covariance:
   \[
   S(k+1) = H(k+1)P(k+1|k)H(k+1)'+R(k+1)
   \]

3. Filter Gain \( W(k+1) = P(k+1|k)H(k+1)' S(k+1)^{-1} \)

4. Updated state covariance
   \[
   P(k+1|k+1) = P(k+1|k) - W(k+1)S(k+1)W(k+1)'
   \]
State at $t_k$
$x(k)$
**Matrix Riccati Equation**

The covariance calculations are *independent* of state (not so for EKF later)

=> can be performed *offline* and are given by:

\[
P(k+1|k) = F(k) \left[ P(k|k-1) - P(k|k-1)H(k)'[H(k)P(k|k-1)H(k)'+R(k)]^{-1}\right]F(k)' + Q(k)
\]

This is the *Riccati equation* and can be obtained from the Kalman filter equations above.

The solution of the Riccati equation in a time invariant system converges to steady state (finite) covariance if the pair \{F, H\} is completely observable (i.e., the state is visible from the measurements alone).
\{F, H\} is completely observable if and only if the \textit{observability matrix} 

\[
Q_0 = \begin{bmatrix}
F \\
FH \\
\vdots \\
FH^{n_x-1}
\end{bmatrix}
\]

has full rank of \(n_x\).

The convergent solution to the Riccati equation yields the \textit{steady state gain} for the Kalman Filter.
FALLING BODY KALMAN FILTER (continued)

Assume an initial *true* state of position = 100 and velocity = 0, g=1.

We choose an initial estimate state estimate $\hat{x}(0)$ and initial state covariance $P(0)$ based on mainly intuition. The state noise covariance $Q$ is all zeros.

The measurement noise covariance $R$ is estimated from knowledge of predicted observation errors, chosen as 1 here.

$F, G, H$ are known the Kalman filter equations can be applied:
<table>
<thead>
<tr>
<th>$t=kT$</th>
<th>True values</th>
<th>Estimates</th>
<th>Errors in Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Position $x_1$</td>
<td>Velocity $x_2$</td>
<td>Meas. $z(k)$</td>
</tr>
<tr>
<td>0</td>
<td>100.0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>99.5</td>
<td>-1.0</td>
<td>100.0</td>
</tr>
<tr>
<td>2</td>
<td>98.0</td>
<td>-2.0</td>
<td>97.9</td>
</tr>
<tr>
<td>3</td>
<td>95.5</td>
<td>-3.0</td>
<td>94.4</td>
</tr>
<tr>
<td>4</td>
<td>92.0</td>
<td>-4.0</td>
<td>92.7</td>
</tr>
<tr>
<td>5</td>
<td>87.5</td>
<td>-5.0</td>
<td>87.3</td>
</tr>
</tbody>
</table>
Kalman Filter Extensions

- Validation gates - rejecting outlier measurements
- Serialisation of independent measurement processing
- Numerical rounding issues - avoiding asymmetric covariance matrices
- Non-linear Problems - linearising for the Kalman filter.
Validation Gate

Recall the measurement prediction covariance:

\[ S(k + 1) = H(k + 1)P(k + 1|k)H(k + 1)' + R(k + 1) \]

and the measurement prediction:

\[ \hat{z}(k + 1|k) = H(k)\hat{x}(k + 1|k) \]

and measurement residual:

\[ v(k + 1) = z(k + 1) - \hat{z}(k + 1|k) \]

A validation gate can be set up around measurements as follows:

\[ e^2 = v(k + 1)S^{-1}(k + 1)v'(k + 1) \leq g^2 \]

where \( g^2 \) is chosen to for a confidence level. Normalised error \( e^2 \) varies as a Chi-Squared distribution with number of measurements degrees of freedom.
**Sequential Measurement Processing**

If the measurement noise vector components are uncorrelated then state update can be carried out one measurement at a time.

Thus matrix inversions are replaced by scalar inversions.

Procedure: state prediction as before

- scalar measurements are processed sequentially (in any order)
- using scalar measurement equations.
Numerical Rounding Problems

The covariance update

\[ P(k + 1|k + 1) = P(k + 1|k) - W(k + 1)S(k + 1)W(k + 1)' \]

involves subtraction and can result in loss of symmetry and positive definiteness due to rounding errors.

*Joseph’s form covariance update* avoids this at expense of computation burden:

\[ P(k + 1|k + 1) = [I - W(k + 1)H(k + 1)]P(k + 1|k)[I - W(k + 1)H(k + 1)]' \]
\[ + W(k + 1)R(k + 1)W(k + 1)' \]

Only subtraction is “squared” and preserves symmetry.
Extended Kalman Filter (EKF)

Many practical systems have non-linear state update or measurement equations. The Kalman filter can be applied to a linearised version of these equations with loss of optimality:
EKF - p 387 Bar-Shalom
Iterated Extended Kalman Filter (IEKF)

The EKF linearised the state and measurement equations about the predicted state as an operating point. This prediction is often inaccurate in practice.

The estimate can be refined by re-evaluating the filter around the new estimated state operating point. This refinement procedure can be iterated until little extra improvement is obtained - called the IEKF.
C++ Software Library

Matrix class - operators overloaded including:

+ * / ~(transpose) =, +=, -=, *=

() for accessing elements,

|| && vertical and horizontal composition,

<< >> input output with automatic formatting

inverse, determinant, index range checking, begin at 1 (not 0!), complex numbers can be used, noise sources, iterative root finding

Constant matrices, Eye, Zeros etc
Kalman filtering classes, for defining and implementing KF, EKF and IEKF

- allows numerical checking of Jacobian functions

Software source is available to collaborators for non-commercial use provided appropriately acknowledged in any publication. Standard disclaimers apply!

contact Lindsay.Kleeman@monash.edu.au
Further Reading


Odometry Error Covariance Estimation for Two Wheel Robot Vehicles \textit{(Technical Report MECSE-95-1, 1995)}

A \textit{closed form} error covariance matrix is developed for
\begin{itemize}
  \item[(i)] straight lines and
  \item[(ii)] constant curvature arcs
  \item[(iii)] turning about the centre of axle of the robot.
\end{itemize}

Other paths can be composed of short segments of constant curvature arcs.

Assumes wheel distance measurement errors are zero mean white noise.

Previous work incrementally updates covariance matrix in small times steps.
Our approach *integrates* noise over the entire path for a closed form error covariance - more efficient and accurate
Scanned Monocular Sonar Sensing

Small ARC project 1995 - aims:

- To investigate a scanned monocular ultrasonic sensor capable of high speed *multiple* object range and *bearing* estimation.

- Deploy the sensor in these robotic applications:
  - obstacle avoidance,
  - doorway traversal and docking operations,
  - localisation and mapping.