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

$$\begin{aligned}
\mathrm{cov}(x_1, x_2) &\equiv E[(x_1 - \bar{x}_1)(x_2 - \bar{x}_2)] \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x_1 - \bar{x}_1)(x_2 - \bar{x}_2)\,p(x_1, x_1)\,dx_1 dx_2 \\
&\equiv s^2_{x_1 x_2}
\end{aligned}$$

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

The ***correlation coefficient*** is the normalised quantity

$$r_{12} \equiv \frac{s^2_{x_1 x_2}}{s_{x_1} s_{x_2}}, \quad -1 \leq r_{12} \leq +1$$

The covariance of a *column vector* $\mathbf{x}=[x_1 .. x_n]'$ is defined as

$$\begin{aligned} \operatorname{cov}(\mathbf{x}) &\equiv E[(\mathbf{x}-\bar{\mathbf{x}})(\mathbf{x}-\bar{\mathbf{x}})'] \\ &= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} (\mathbf{x}-\bar{\mathbf{x}})(\mathbf{x}-\bar{\mathbf{x}})' \, p(\mathbf{x}) \; dx_1 .. dx_n \\ &\equiv \mathbf{P}_{\mathbf{xx}} \end{aligned}$$

and is a *symmetric* n by n matrix and is *positive definite* unless there is a linear dependence among the components of $\boldsymbol{x}$.

The (i,j)th element of $\mathbf{P}_{\mathbf{xx}}$ is $\sigma^2_{x_i x_j}$

Interpreting a covariance matrix:

diagonal elements are the variances, off-diagonal encode correlations.

Diagonalising a Covariance Matrix

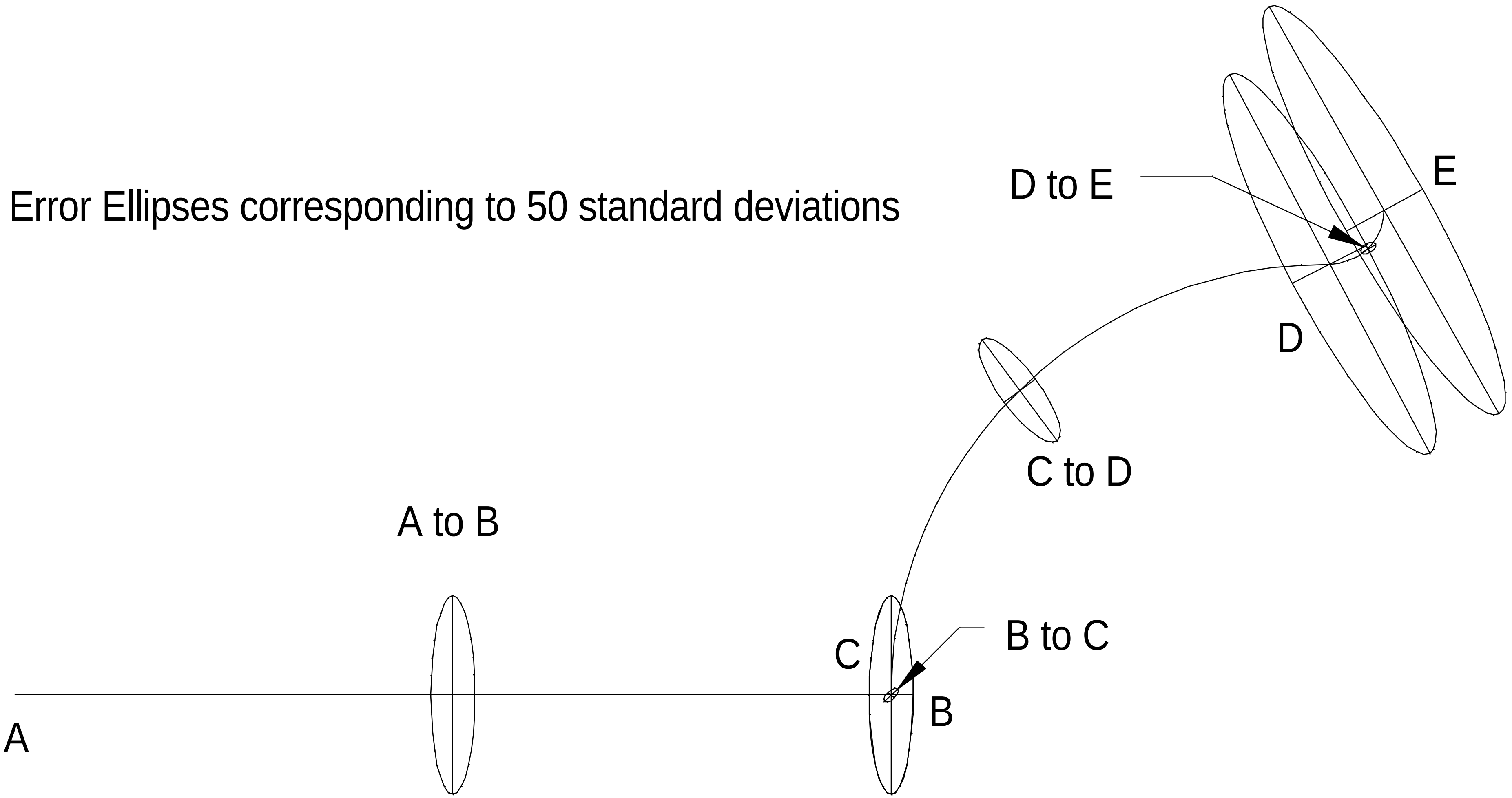
cov(**x)** is symmetric => can be ***diagonalised*** using an ***orthonormal*** basis.

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
D to E
E
D
C to D
A to B
C
B to C
B
A

10000 Monte-Carlo runs for $k_L = k_R = 10^{-3} m^{½}$, *B=0.5 m*

	Means	Covariance Matrix	Stand dev/ Corr Matrix
Theor-etical results	0 0 0	3.032e-5 -4.763e-5 -2.817e-5 -4.763e-5 8.974e-5 4.700e-5 -2.817e-5 4.700e-5 3.4849e-5	0.005506 -0.913208 -0.8667 -0.9132 0.009473 0.8404 -0.8667 0.8404 0.005903
Run 1 10000 samples	-1.997e-5 5.321e-5 3.184e-5	2.980e-5 -4.651e-5 2.761e-5 -4.651e-5 8.708e-5 4.585e-5 -2.761e-5 4.585e-5 3.437e-5	0.005459 -0.9130 -0.8627 -0.9130 0.009332 0.8380 0.8627 0.8380 0.005862

Formulating a Kalman Filter Problem

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

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:

$$\mathbf{x}(k+1)=\mathbf{F}(k)\mathbf{x}(k)+\mathbf{G}(k)\mathbf{u}(k)+\mathbf{v}(k) \qquad k=0,1,\ldots$$

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

$$E[\mathbf{v}(k)\mathbf{v}(k)']=\mathbf{Q}(k)$$

Measurement equation:

$$\mathbf{z}(k)= \quad (k) \;\; (k)+ \quad (k) \qquad k=1,\ldots.$$

***w**(k)* is unknown zero mean white *measurement* noise with known covariance

$$E[\mathbf{w}(k)\mathbf{w}(k)']=\mathbf{R}(k)$$

FALLING BODY EXAMPLE

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

$$\ddot{y}(t) = -g$$

$$\Rightarrow \dot{y}(t) = \dot{y}(t_0) - g(t - t_0)$$

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

As a discrete time system with time increment of $t\text{-}t_0\text{=}1$

$$y(k+1) = y(k) + \dot{y}(k) - \frac{g}{2}$$

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

We can define the state as

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

and then the state equation becomes

$$
\begin{aligned}
\mathbf{x}(\mathrm{k}+1) &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \mathbf{x}(\mathrm{k}) + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} (-g) \\
&= \mathbf{F}\,\mathbf{x}(k) + \mathbf{G}\,\mathbf{u}
\end{aligned}
$$

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

$$
\begin{aligned}
\mathbf{z}(\mathrm{k}) &= [1 \;\; 0]\; \mathbf{x}(\mathrm{k}) + \mathrm{w}(\mathrm{k}) \\
&= \mathbf{H}\,\mathbf{x}(k) + w(k)
\end{aligned}
$$

The variance of *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:

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

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

and the error covariance matrix of the state estimate

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

$$\mathbf{P}(k+1|k) \ - \text{estimate of } \mathbf{x}(k+1) \text{ given } z(k),\ z(k-1),\ldots$$

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

State Estimation

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

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

Time update

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

measurement update

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

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

where **W**(k+1) is called the Kalman Gain defined next in the state covariance estimation.

State Covariance Estimation

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

2. Measurement prediction covariance:

$$\mathbf{S}(k+1) = \mathbf{H}(k+1)\mathbf{P}(k+1|k)\mathbf{H}(k+1)' + \mathbf{R}(k+1)$$

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

4. Updated state covariance

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{W}(k+1)\mathbf{S}(k+1)\mathbf{W}(k+1)'$$

Page 219 Bar-Shalom ANATOMY OF KALMAN FILTER

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:

$$\mathbf{P}(k+1|k)=\mathbf{F}(k)\left\{\begin{array}{l}\mathbf{P}(k|k-1)-\mathbf{P}(k|k-1)\mathbf{H}(k)'\left[\mathbf{H}(k)\mathbf{P}(k|k-1)\mathbf{H}(k)'+\mathbf{R}(k)\right]^{-1}\\ .\mathbf{H}(k)\mathbf{P}(k|k-1)\end{array}\right\}\mathbf{F}(k)'+\mathbf{Q}($$

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 (ie the state is visible from the measurements alone).

{F, H} is completely observable if and only if the ***observability matrix***

$$Q_0 = \begin{bmatrix} F \\ FH \\ \dots \\ FH^{n_x - 1} \end{bmatrix}$$

has full rank of n_x.

The convergent solution to the Riccati equation yields the ***steady state gain*** for the Kalman Filter.

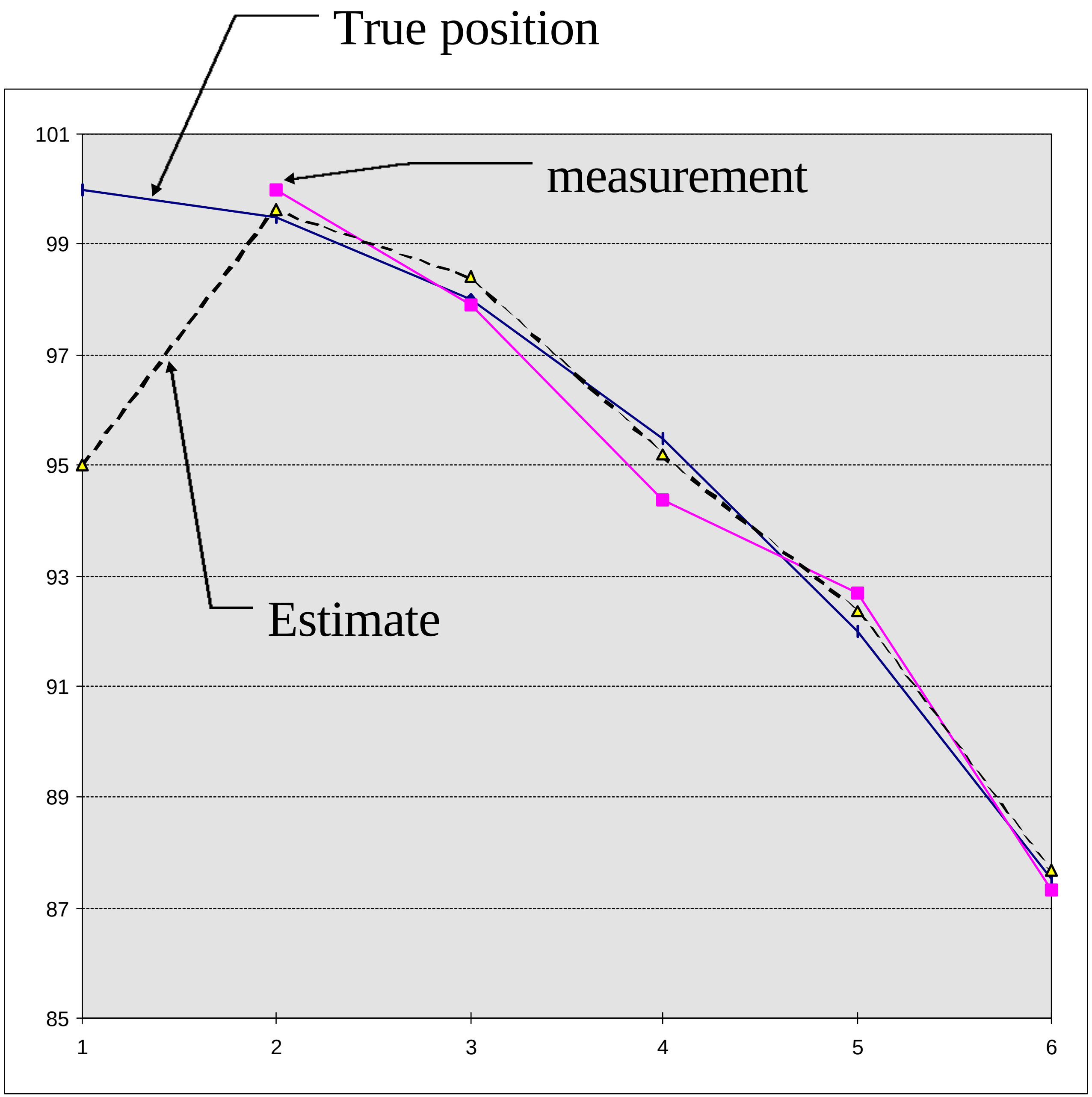
FALLING BODY KALMAN FILTER (continued)

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

We choose an initial estimate state estimate $\hat{\mathbf{x}}(0)$ and initial state covariance $\mathbf{P}(0)$ 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:


True position
measurement
Estimate
101
99
97
95
93
91
89
87
85
1
2
3
4
5
6

	True values			Estimates		Errors in Estimate	
	Position	Velocity	Meas.	Position	velocity	Position	velocity
t=kT	x_1	x_2	z(k)	$\hat{x}_1(k)$	$\hat{x}_2(k)$	$P_{11}(k)$	$P_{22}(k)$
0	100.0	0		95.0	1.0	10.0	1.0
1	99.5	-1.0	100.0	99.63	0.38	0.92	0.92
2	98.0	-2.0	97.9	98.43	-1.16	0.67	0.58
3	95.5	-3.0	94.4	95.21	-2.91	0.66	0.30
4	92.0	-4.0	92.7	92.35	-3.70	0.61	0.15
5	87.5	-5.0	87.3	87.68	-4.84	0.55	0.08

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:

$$\mathbf{S}(k+1) = \mathbf{H}(k+1)\mathbf{P}(k+1|k)\mathbf{H}(k+1)' + \mathbf{R}(k+1)$$

and the measurement prediction: $\hat{\mathbf{z}}(k+1|k) = \mathbf{H}(k)\,\hat{\mathbf{x}}(k+1|k)$

measurement update

and measurement residual: $\mathbf{v}(k+1) = \mathbf{z}(k+1) - \hat{\mathbf{z}}(k+1|k)$

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

$$e^2 = \mathbf{v}(k+1)\mathbf{S}^{-1}(k+1)\mathbf{v}'(k+1) \le g^2$$

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

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{W}(k+1)\mathbf{S}(k+1)\mathbf{W}(k+1)'$$

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

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

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

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

Bar-Shalom and Xiao-Rong Li**, Estimation and Tracking: Principles, Techniques and Software**, Artech House Boston, 1993.

Jazwinski, A. H. . **Stochastic Processes and Filtering Theory**. New York, Academic Press, 1970.

Bozic, S M, **Digital and Kalman Filtering**, Edward Arnold, London 1979.

Maybeck, P. S. “The Kalman filter: An introduction to concepts.” Autonomous Robot Vehicles. I. J. Cox and G. T. Wilfong. New York, Springer-Verlag: 194-204, 1990.

Odometry Error Covariance Estimation for Two Wheel Robot Vehicles ***(Technical Report MECSE-95-1, 1995)***

A *closed form* error covariance matrix is developed for

(i) straight lines and

(ii) constant curvature arcs

(iii) turning about the centre of axle of the robot.

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.