TIADPE

Positioning fusion by the Kalman estimator

Christian Fischer Pedersen
Assistant Professor
cfp@eng.au.dk

Section of Electrical and Computer Engineering
Department of Engineering
Aarhus University

December 15, 2014

Outline

Outline

Multisensor data fusion

Kalman filter overview

Kalman filter details

Example: Location in one dimension

Multisensor data fusion

- The process of **combining** observations from a **number** of different sensors to provide a **robust** and **complete** description of an environment or process of interest
- The **combining** of sensory data from **disparate** sources such that the resulting information is in some sense **better**, e.g. more accurate, complete or dependable, than would be possible if these sources were used **individually**

Application areas

- Mobile robotics, **autonomous vehicles**, **object tracking** ...
- Denoising, **positioning**, e.g. for location based services

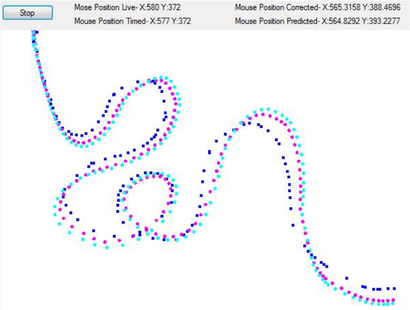


Figure : Mouse tracking. True, predicted and corrected tracks

Application example: Object tracking

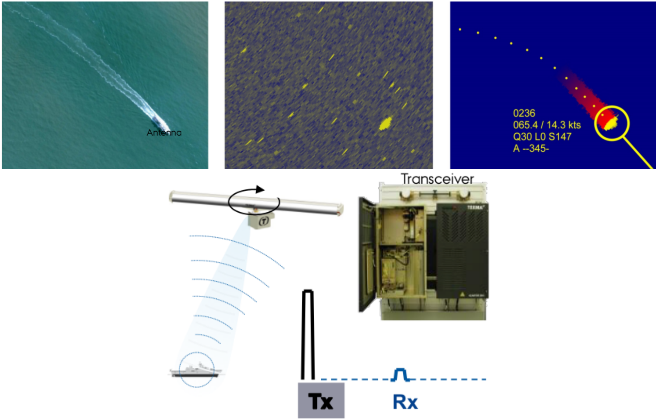


Figure : Locating and tracking objects by Tx/Rx signatures

Application example: Autonomous vehicle

- Autonomous VW Passat at Stanford University
- Onboard sensors include:
  Radar, lidar, GPS, compass, gyro, odometer, camera, ...
- Fuse the sensor information:
  Gather a complete overview over the car and its environment

Reasons and benefits of multiple sensors

- Distributed sensing needs data fusion
- Alleviate sensor imperfections and malfunctions
- Overcome technical limitations of sensors
- Indirect measuring in complex and/or occluded environments
- Representation, e.g improve resolution
- Certainty, e.g. improve likelihood
- Accuracy, e.g. minimize effects of outliers

Outline

Multisensor data fusion

Kalman filter overview

Kalman filter details

Example: Location in one dimension

Ubiquity of the Kalman filter

Figure : Few examples of books on Kalman filtering

Figure : Kalman merchandise...

Kalman is a notable scientist and engineer

Figure : Rudolf Emil Kálmán. Born in Budapest, Hungary, 1930. Received the National Medal of Science in 2008.

Kalman filter applications

The Kalman Filter is over 50 years old but is still one of the most important and common data fusion algorithms in use today

- Apollo navigation to the moon and back
- Self driving cars and other autonomous vehicles
- In satellite navigation devices
- In every modern smart phone
- In many computer games
- ...

Kalman filter: From a noise filtering perspective

Given a signal picked up by one or more **sensors**

- E.g. sound, image, radar or GPS
- The measurements are contaminated with **noise**

How to discard the noise?

- E.g. averaging neighbouring samples: Often **not** good results
- We need a more sophisticated approach for real life problems

The Kalman filter is a very good method for discarding the noise

- I.e. **estimates** the noise-free signal

Application of importance in pervasive computing

- E.g. estimate accurate **position** from noisy GPS data

Kalman filter: From an estimator perspective

The Kalman Filter is an **MMSE optimal** linear estimator

- For systems with Gaussian error statistics
- Estimates the state of a system given a set of observations
- Has a theoretical guarantee of **convergence**
- Implemented as an **recursive** algorithm
- Is applicable without understanding **all** theory

Aims to **filter** out noise to **estimate** the underlying truth

Kalman filter: From a data fusion perspective

You have

- An **expectation** of how the system should behave, i.e. a **model** of the system
- **Measured** information actual system behavior, i.e. noisy sensor **data**

You need

- The best possible idea of the system **behavior**

You apply

- An estimator to **fuse measurement** and **model** information

Kalman steps: Predict and correct

1. **Predict** next state using current state and system model
2. **Correct** the predicted state using sensor observations

$$\begin{tikzcd}
\boxed{\text{Estimated state}} \arrow[r, shift left=2, "\text{Prediction}"] & \boxed{\text{Predicted state}} \arrow[l, shift left=2, "\text{Correction}"]
\end{tikzcd}$$

Analogy: Estimation by predict and correct 1/2

Courtesy of M. Bisgaard

Real world example

- You are on a train trip from Aalborg to Copenhagen
- You fall asleep and wake up about 2 hours later
- How far has the train travelled? When will you arrive?

You make a **model based estimation**

Analogy: Estimation by predict and correct 2/2

Courtesy of M. Bisgaard

1 Frederikshavn – Aalborg – Århus – Odense – København

køredage								
Frederikshavn		16.39						17.41
Kvissel		16.46						
Tolne		16.52						
Sindal		16.58						17.58
Hjørring		17.11						18.11
Vrå		17.19						18.19
Brønderslev		17.28						18.28
Lindholm		17.49			18.00			18.49
Aalborg Vestby		17.52			18.03			18.52
Aalborg	o	17.55			18.06			18.55
Aalborg			17.59		18.10			
Skalborg					18.15			
Svenstrup					18.20			
Støvring					18.27			
Skørping					18.34			
Arden					18.40			
Hobro			18.29		18.52			
Randers			18.45		19.12			
Langå					19.22			
Hadsten					19.31			
Hinnerup								
Århus H	o		19.21		19.52			
Århus H			19.32			20.02		
Skanderborg			20.03			20.33		
Horsens			20.20			20.50		
Vejle			20.58			21.28		
Fredericia	o		21.20			21.50		
Fredericia			1 21.36			22.06		
Middelfart				21.48		22.15	22.48	
Odense	o		22.07	22.13		22.43	23.13	
Odense			22.10	22.17		22.46	23.17	
Nyborg				22.32		23.02	23.32	
Korsør				22.45		23.15	23.45	
Slagelse				22.54		23.25	23.54	
Sorø				23.03			0.03	
Ringsted	o			23.11		23.38	0.11	
Roskilde	o			23.28		23.55	0.28	
Høje Taastrup	o		23.14	23.37		0.04	0.37	
Valby	o			23.46			0.46	
København H	o		1 23.29	23.53		0.19	0.53	

- The model: Time table
- The sensor data: Looking out the window
- You **fuse** the two pieces of info to estimate **actual** location

Outline

Multisensor data fusion

Kalman filter overview

Kalman filter details

Example: Location in one dimension

Kalman filter main principles expressed with PDFs

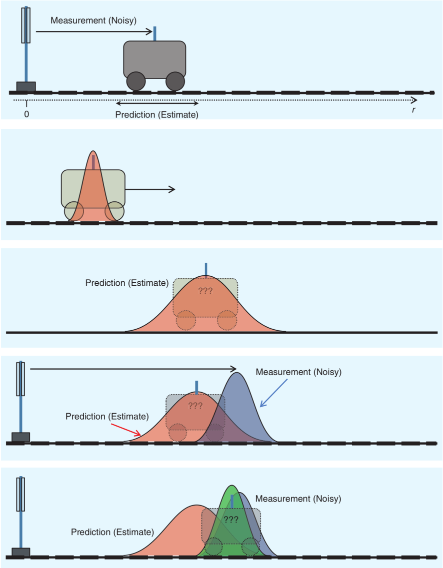
Train w/ positioning receiver

t_0: Estimate of current pos.

t_0: Prediction of pos. at t_1

t_1: Position measurement

t_1: Predict and meas. fusion



Kalman filter main principles expressed with PDFs

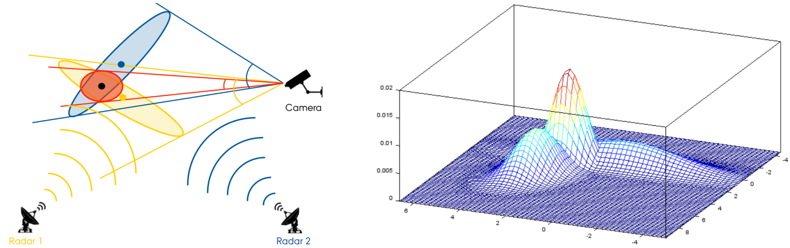


Figure : The Kalman filter can be used for fusing data from different sensors. Each sensor provides a state estimate; location estimates in this example. PDFs of estimates are fused (multiplied).

Kalman filter model 1/2: Estimation model

Estimate the process, $\mathbf{x} \in \mathbb{R}^n$, governed by

$$x_k = \mathbf{A}x_{k-1} + \mathbf{B}u_{k-1} + w_{k-1}$$

where

- x: state vector (e.g., position, velocity, heading)
- $\mathbf{A}$: state transition matrix (applies the effects of x_{k-1} on x_k)
- u: control vector (e.g., steering angle, throttle, braking force)
- $\mathbf{B}$: control input matrix (applies the effects of u_{k-1} on x_k)
- w: process noise; $\mathsf{p}(w) \sim \mathcal{N}(0, \mathbf{Q})$. ($\mathbf{Q}$ must be guessed)
  Better noise models lead to better estimates

Kalman filter model 2/2: Measurement model

Systems **measurements**, $\mathbf{z} \in \mathbb{R}^m$, can be performed according to

$$z_k = \mathbf{H} x_k + v_k$$

where

- z: measurement vector (e.g., position, velocity, heading)
- x: state vector (e.g., position, velocity, heading)
- $\mathbf{H}$: state-domain to measurement-domain transformation
- v: measurement noise; $\mathrm{p}(v) \sim \mathcal{N}(0, \mathbf{R})$. ($\mathbf{R}$ can be measured) Better noise models lead to better estimates

Covariances between terms in the state vector

The covariance matrix, $\mathbf{P}$, needed to describe the Gaussians

- Terms along the main diagonal are variances of the corresponding terms in the state vector
- Off-diagonal terms are the covariances between terms in the state vector

$$\begin{bmatrix} \mathrm{E}[(X_1-\mu_1)(X_1-\mu_1)] & \mathrm{E}[(X_1-\mu_1)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_1-\mu_1)(X_n-\mu_n)] \\ \mathrm{E}[(X_2-\mu_2)(X_1-\mu_1)] & \mathrm{E}[(X_2-\mu_2)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_2-\mu_2)(X_n-\mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{E}[(X_n-\mu_n)(X_1-\mu_1)] & \mathrm{E}[(X_n-\mu_n)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_n-\mu_n)(X_n-\mu_n)] \end{bmatrix}$$

Kalman steps: Predict and correct

1. **Predict** next state using current state and system model

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k \qquad \text{State propagation}$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}_k \mathbf{P}_{k-1|k-1} \mathbf{A}_k^T + \mathbf{Q}_k \qquad \text{Covariance propagation}$$

2. **Correct** the predicted state using sensor observations

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}) \qquad \text{State update}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k|k-1} \qquad \text{Covariance update}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \qquad \text{Kalman gain}$$

3. **Recursively** carry out step 1 and 2 above

$$\begin{tikzcd}
\boxed{\text{Estimated state}} \arrow[r, shift left, "\text{Prediction}"] & \boxed{\text{Predicted state}} \arrow[l, shift left, "\text{Correction}"]
\end{tikzcd}$$

Outline

Multisensor data fusion

Kalman filter overview

Kalman filter details

Example: Location in one dimension

Problem formulation

- Estimate a scalar const, a, e.g. location on 1D path in meters
- A GPS gives us the location along the dimension of interest
- The readings are however noisy, i.e. above/below the target
- Assume stdev of white measurement noise is $\mathbf{R} = 0.1$m

Measurements

T [ms]	1	2	3	4	5	6	7	8	9	10
L [m]	0.39	0.50	0.48	0.29	0.25	0.32	0.34	0.48	0.41	0.45

Initial estimations

- $\hat{x}_{k=0} = 0$, $\mathbf{P}_{k=0} = 1$, and $\mathbf{Q} \approx 0$
- Choose $\mathbf{P}_{k=0} \neq 0$ to introduce noise in environment
  - Otherwise value of $\hat{x}_k$ would remain as in init. state

Model building: Signal model

About the signal model

- Problem is 1D, i.e. model entities are scalars not matrices
- There is no control signal so $u_k = 0$
- As signal is constant set $\mathbf{A} = 1$
- Even if other linear nature often reasonable to assume $\mathbf{A} = 1$

Build a signal/process model

$$
\begin{aligned}
x_k &= \mathbf{A}x_{k-1} + \mathbf{B}u_{k-1} + w_k \\
&= x_{k-1} + w_k
\end{aligned}
$$

Model building: Measurement model

About the measurement model

- Problem is 1D, i.e. model entities are scalars not matrices
- Measurements are directly of state and noise: $\mathbf{H} = 1$
- Often reasonable to assume in real life examples $\mathbf{H} = 1$

Build a measurement model

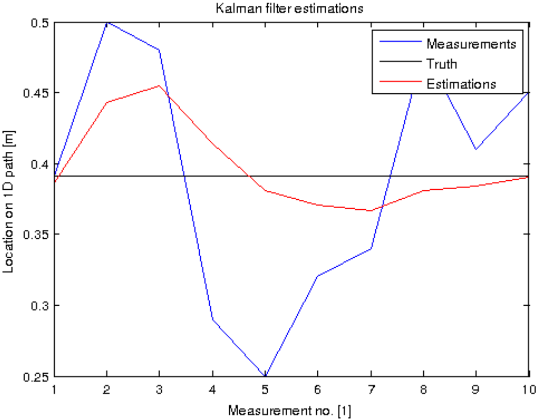
$$\begin{aligned} z_k &= \mathbf{H}x_k + v_k \\ &= x_k + v_k \end{aligned}$$

Computations: First 3 iterations

Remember the initial estimations: $\hat{x}_{k=0} = 0$, $\mathbf{P}_{k=0} = 1$, and $\mathbf{Q} \approx 0$

	k	1	2	...
	z_k	0.39	0.50	...
Predict	$\hat{x}_{k-} = \hat{x}_{k-1}$	initial estimation: 0	0.35	...
	$\mathbf{P}_{k-} = \mathbf{P}_{k-1}$	initial estimation: 1	0.09	...
Correct	$\mathbf{K}_k = \mathbf{P}_{k-}(\mathbf{P}_{k-} + \mathbf{R}_k)^{-1}$	$\frac{1}{1+0.1} = 0.91$	0.47	...
	$\hat{x}_k = \hat{x}_{k-} + \mathbf{K}_k(z_k - \hat{x}_{k-})$	$0 + 0.91(0.39 - 0) = 0.35$	0.42	...
	$\mathbf{P}_k = (1 - \mathbf{K}_k)\mathbf{P}_{k-}$	$(1 - 0.91) \cdot 1 = 0.09$	0.05	...

Plot: All 10 iterations

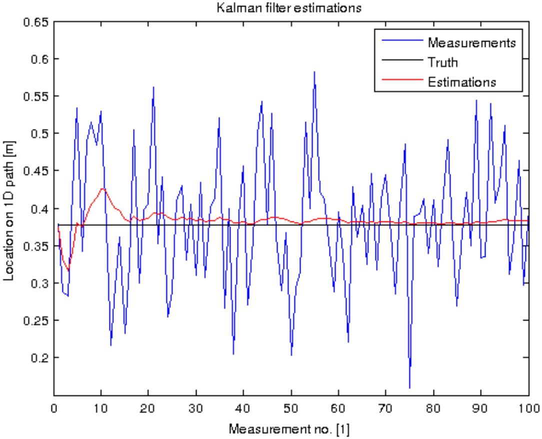


To enable the convergence in fewer steps

- Precise system model and noise estimations

Extended 1D example: The Kalman estimations

- Measurements are a const, 0.37727, w/ Gaussian noise added
- Gaussian noise w/ mean $= 0$ and stdev $= 0.1$



Extended 1D example: The added Gaussian noise

- Measurements are a const, 0.37727, w/ Gaussian noise added
- Gaussian noise w/ mean $= 0$ and stdev $= 0.1$