Advanced Sensor Data Fusion in Distributed Systems

SS 2019
Felix Govaers,
Dr. rer. nat., Dipl.-Math.
About me

Research interest:

- State estimation and target tracking in multi sensor networks
- In particular
  - Out-of-Sequence Processing and Track-to-Track Fusion
  - Tensor Decomposition Tracking
- Diploma of Mathematics (2007)
- Ph.D. of Computer Science (2012)
- University of Bonn (2007 – 2009)
- Fraunhofer FKIE (since 2009)
  - Former Head of Research Group “Distributed Systems”
  - Deputy of Head of Department “Sensor Data and Information Fusion”
Our institute

FRAUNHOFER FKIE
Department: Sensor Data and Information Fusion

RG Resources Management
RG Integrated Systems
Sensor Data and Information Fusion (SDF)
RG Array Processing
RG Distributed Systems

Location: Fraunhofer FKIE Wachtberg
A motivation for

SENSOR DATA FUSION IN
DISTRIBUTED SYSTEMS
Why do Humans have Sensors?
All Creatures do Sensor Data Fusion (SDF)

- Ambiguous sensing information
- Sensing error
- Complementary sensors
  - visual
  - acoustic
  - smelling
  - etc.
- Environment is a dense situation
- Target behavior can be predicted

Sensor Data Fusion is a cognitive tool for situation awareness.
Distributed Sensor Fusion for Saving Computation
Distributed Sensor Fusion for Network Bandwidth

© intelligent-aerospace.com
Distributed Sensor

\[ x_k|k = \mathbb{E} \left[ x_k|x_1^1, \ldots, x_5^5 \right] =? \]

\[ P_k|k = \text{cov} \left[ x_k|x_1^1, \ldots, x_5^5 \right] =? \]
Centralized vs. Decentralized vs. Distributed

**Centralized:**
- Distinguished Fusion Center (FC)
- Send all measurements to FC
- No cross covariances of tracks
- Doesn’t scale with False Alarms (FA)

**Decentralized:**
- Every node is a FC
- Send data only to neighbours
- Estimate network topology
- Account for “information incest”
- Only approximate solutions exist

**Distributed:**
- Distinguished FC
- Preprocess measurements on nodes
- Send estimates to FC
- Account for X-correlations
- Scales very well in number of sensors and in FA.
Tracking in Distributed Systems

Distributed Systems have the following advantages and drawbacks.

**Advantages:**
- No single point of failure
- Cheap sensor technology
- Complementary sensors
- Spatial distribution
- Distributed computation

**Challenges:**
- Communication of data necessary
- Sensor synchronization
- Sensor registration
- Correlations
How to describe

INSECURE KNOWLEDGE
Probability Density Functions (pdf)

- Pdfs represent imprecise knowledge about the true state of a random variable.
- Pdf $p$ is a real valued function and $p \geq 0$.
- $p$ is normalized
  \[ \int \text{d}x \, p_x(x) = 1 \]
- Imprecise knowledge about the state of an object at time $k$ given all data up to time $k$:
  \[ p(x_k | \mathcal{Z}^k) \]
- Probability that the state is within a region $A$:
  \[ \mathbb{P}[x \in A] = \int_A \text{d}x \, p_x(x) \]
Examples

1 Rolling of a Dice
- RV $x$ represents outcome of one dice.
- Event space: $X = \{1, 2, ..., 6\}$
- $P[x = j] = p(j) = 1/6$

2 Stock value prediction
- RV $x$ represents value at time $t_f$ in the future.
- Event space $X = \mathbb{R}$
- Value at current time: 100
- $P[x < 100] = P[x \in [0, 100)]$

\[
\int_{(0,100)} dx \; p(x) < 0.5 \Rightarrow \text{long!} \\
\text{otherwise} \Rightarrow \text{short!}
\]
How to deal with pdfs?

- **conditional pdf**

\[ p(x|y) = \frac{p(x,y)}{p(y)} \]

Impact of knowledge about a RV y on x

- **marginalization**

\[ p(x) = \int dy \, p(x,y) = \int dy \, p(x|y) \, p(y) \]

Introduce y to the equation

- **Bayes Theorem**

\[ p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\int dx \, p(y|x)p(x)} \]

\[ \text{Posterior} = c \times \text{Likelihood function} \times \text{Prior} \]

- **Certain knowledge**

\[ p(x) = \delta(x - y) \]

Certain knowledge that \( x = y \)
Parameters of Interest: Mean and Covariance

The expectation of a pdf is its centroid:

\[
E[x] = \int_{\mathbb{R}^n} dx \ p(x) \cdot x
\]

The covariance is a measure of spreading around the expectation value:

\[
cov[x] = E[(x - E[x])^2]
\]
\[
\bar{x} = E[x]
\]
\[
cov[x] = E[(x - \bar{x}) \cdot (x - \bar{x})^\top]
\]
STOCHASTIC MOTION
Basic Idea of Sensor Data Fusion

Prior knowledge: target dynamics model, sensor model, context information.

- **prediction:** $p(x_{k-1}|z^{k-1})$ \[\xrightarrow{\text{dynamics model}}\] $p(x_k|z^{k-1})$

- **filtering:** $p(x_k|z^{k-1})$ \[\xrightarrow{\text{sensor data } z_k}\] $p(x_k|z^k)$

- **retrodiction:** $p(x_{l-1}|z^k)$ \[\xleftarrow{\text{filtering output}}\] $p(x_l|z^k)$
Prediction – Filtering - Cycle

Retrodiction (optional) → Prediction → Filtering → Retrodiction (optional)
A stochastic process is a set of random variables, which represent the state of a non-deterministic state:

Continuous

\[ \{X_t\}_{t \geq 0} \]

\( t = \text{time} \)

Discrete

\[ \{X_k\}_{k \in \mathbb{N}} \]

\( k = \text{time step} \)
Example

Discrete, $\Delta_t = 1/8$
Stochastic Motion: Diffusion in 1D

For a state $x_k$ in $\mathbb{R}$, consider the following transition kernel:

$$p(x_k | x_{k-1}) = \begin{cases} \frac{1}{2}, & \text{if } |x_k - x_{k-1}| = h \\ 0, & \text{else.} \end{cases}$$

This implies for the pdf of the succeeding state that

$$p(x_{k+1}) = \sum_{x_k \in hZ} p(x_{k+1} | x_k)p(x_k)$$

$$= \frac{1}{2} p(x_k + h) + \frac{1}{2} p(x_k - h).$$

As a consequence, one has

$$\frac{p(x_{k+1}) - p(x_k)}{\tau} = \frac{h^2}{2\tau} \frac{p(x_k + h) - 2p(x_k) + p(x_k - h)}{h^2}$$

with $\tau = \text{time step}$ and $D = \frac{h^2}{2\tau}$.

The diffusion equation is obtained:

$$\partial_t p(x_k) = D \partial_{x}^2 p(x_k)$$

$$\tau \to 0 \text{ and } \tau \to 0$$
Example
Generalization to Arbitrary Dimensions

A random walk in $n$ dimensions is described by the transition model

$$ p(x_{k+1}) = \frac{1}{2n} \sum_{i=1}^{n} (p(x_k + he_i) + p(x_k - he_i)) $$

Analogously to the 1D case, we may write

$$ \frac{p(x_{k+1}) - p(x_k)}{\tau} = \frac{h^2}{2n\tau} \sum_{i=1}^{n} \frac{p(x_k + h) - 2p(x_k) + p(x_k - h)}{h^2} $$

Taking the limit leads to the general **Diffusion Equation**:

$$ \partial_t p(x_k) = D \Delta_x p(x_k) $$

$$ D = \frac{h^2}{2\tau n} $$

$$ \Delta_x = \sum_{i=1}^{n} \partial^2_{x_i} $$
Example Trajectories
Example
Bernoulli Model

We may modify our model to a Bernoulli process, which assigns probabilities $u$ and $v$ such that $u + v = 1$ to the options:

$$p(x_{k+1}) = up(x_k + h) + vp(x_k - h).$$

Using the fact that $v = 1 - u$, this can be rewritten as

$$p(x_{k+1}) = (u - \frac{1}{2})p(x_k + h) + \frac{1}{2}p(x_k + h) - (u - \frac{1}{2})p(x_k - h) + \frac{1}{2}p(x_k - h)$$

Therefore

$$\frac{p(x_{k+1}) - p(x_k)}{\tau} = \frac{h^2}{2\tau} \frac{p(x_k + h) - 2p(x_k) + p(x_k - h)}{h^2} + \frac{u - \frac{1}{2}}{\tau}p(x_k + h) - \frac{u - \frac{1}{2}}{\tau}p(x_k - h)$$

$$= \frac{h^2}{2\tau} \frac{p(x_k + h) - 2p(x_k) + p(x_k - h)}{h^2} - \frac{2h(\frac{1}{2} - u)}{\tau} \frac{p(x_k + h) - p(x_k - h)}{2h}$$

Taking the limit yields

$$\partial_t p(x_k) = \left( D \partial_x^2 - \nu \partial_x \right) p(x_k) \quad \nu = \frac{2h}{\tau} \left( \frac{1}{2} - u \right) \quad D = \frac{h^2}{2\tau}$$
Example
Multi Dimensional Bernoulli Model

We consider the multivariate walk from above in $\mathbb{R}^n$. However the direction now is not uniformly distributed over all $2n$ options, but parameters $u_1, \ldots, u_n$ are given such that

$$p(x_{k+1}) = \frac{1}{2n} \sum_{i=1}^{n} u_i p(x_k + he_i) + (1 - u_i) p(x_k - he_i).$$

This yields

$$\partial_t p(x_k) = \left( - \sum_{i=1}^{n} v_i \partial_x + D\Delta_x \right) p(x_k)$$

with

$$v_i = \frac{2h}{\tau} \left( \frac{1}{2} - u_i \right)$$
Example
TARGET TRACKING USING SENSOR DATA
Difficult Operational Conditions

- Target detection
  - Small objects \( P_D < 1 \)
  - Fading
  - Minimum detectable radial velocity
- Measurements
  - False returns (clutter, birds, clouds, waves)
  - Measurement error
  - Data to target assignment

- Sensor resolution
  - Group measurements
  - Resolution probability model
- Target behavior
  - High maneuvering capability
  - Distinct maneuvering phases
  - Dynamic object parameters unknown
Examples of a Target State

- Position and velocity
  \[ \mathbf{x}_k = \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix} \]

- Position, velocity and acceleration
  \[ \mathbf{x}_k = \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{pmatrix} \]

- Kinematics and classification
  \[ \mathbf{x}_k = \begin{pmatrix} x \\ y \\ \cdots \\ c \end{pmatrix} \]

- Single parameter estimation
  \[ \mathbf{x}_k = \begin{pmatrix} x \end{pmatrix} \]
Basic Idea of Sensor Data Fusion

Prior knowledge: target dynamics model, sensor model, context information.

- **prediction:** $p(x_{k-1} | Z^{k-1})$ \(\xrightarrow{\text{dynamics model}}\) road maps $p(x_k | Z^{k-1})$

- **filtering:** $p(x_k | Z^{k-1})$ \(\xrightarrow{\text{sensor model}}\) $p(x_k | Z^k)$

- **retrodiction:** $p(x_{l-1} | Z^k)$ \(\xleftarrow{\text{filtering output}}\) $p(x_l | Z^k)$
Prediction – Filtering - Cycle

- Prediction
- Filtering
- Retrodiction (optional)
Important Family of PDFs

- wanted: Probability mass concentrated around a center \( \mu \).
- Parameter for width of the pdf: \( \sigma \).

\[
\mathcal{N}(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)
\]

- Calculations show

\[
\mathbb{E}[x] = \int_{-\infty}^{\infty} dx \ x \ \mathcal{N}(x; \mu, \sigma) = \mu
\]

\[
\mathbb{V}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \sigma^2
\]
Delta Peak: a Limiting Case of a Gaussian

- Secure knowledge: delta peaks.

\[
\delta(x; \mu) \equiv \begin{cases} 
\infty & x = \mu \\
0 & x \neq \mu 
\end{cases}
\]

- Can be interpreted as a limiting case:

\[
\delta(x; \mu) \equiv \lim_{\sigma \to 0} \mathcal{N}(x; \mu, \sigma)
\]
Generalization to Multivariate (MV) RVs

- Accumulate parameters of interest into a vector state:

\[ \mathbb{R}^n \ni \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \]

- Probability can be obtained via volume integral:

\[ P\{\mathbf{x} \in V\} = \int_V dx_1 \ldots dx_n p(x_1, \ldots, x_n) \]

For example:

\[ x_1 = x \]
\[ x_2 = y \]
\[ x_3 = \dot{x} \]
\[ x_4 = \dot{y} \]
Expectation and Covariance of a MV Gaussian

- A MV Gaussian is defined as

\[
p(x) = \mathcal{N}(x; \bar{x}, P) = \frac{1}{\sqrt{\det(2\pi P)}} e^{-\frac{1}{2}(x - \bar{x})^T P^{-1} (x - \bar{x})}
\]

- where
  - \( \mathbb{E}[x] = \bar{x} \in \mathbb{R}^n \)
  - \( \text{cov}[x] = P \in \mathbb{R}^{n \times n} \)
Correlation of Random Variables

Let \( x \) and \( y \) be real valued random variables with a normal distributed joint density:

\[
p(x, y) = \mathcal{N}\left( \begin{pmatrix} x \\ y \end{pmatrix} ; \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}, \begin{pmatrix} P_{xx} & P_{xy} \\ P_{xy} & P_{yy} \end{pmatrix} \right)
\]

The covariance matrix consists of

\[
P_{xx} = \text{cov} [x] = \text{cov} [x, x] = E \left[ (x - E[x])^2 \right]
\]

\[
P_{xy} = \text{cov} [x, y] = E \left[ (x - E[x])(y - E[y]) \right]
\]

Assume \( x \) and \( y \) are zero-mean. That is

\[
E[x] = E[y] = 0.
\]

Then:

\[
P_{xx} = E[x^2]
\]

\[
P_{xy} = E[x \cdot y]
\]
Uncorrelated and Independent Random Variables

If the joint density has the following form:

\[
p(x, y) = \mathcal{N}\left(\begin{pmatrix} x \\ y \end{pmatrix}; \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}, \begin{pmatrix} P_{xx} & 0 \\ 0 & P_{yy} \end{pmatrix}\right)
\]

Then \(x\) and \(y\) are called „uncorrelated“.

If the joint density has the following form:

\[
p(x, y) = p(x) \cdot p(y)
\]

Then \(x\) and \(y\) are called „independent“.

Exercise

If \(x\) and \(y\) are normal distributed and uncorrelated, then they are independent!
Structure of a Covariance Matrix

Let \( \mathbf{x} \) be given as

\[
\mathbb{R}^n \ni \mathbf{x} = \begin{pmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_n
\end{pmatrix}
\]

and \( p(x) = \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{P}) \).

Then:

\[
\mathbf{P} = \text{cov} \begin{pmatrix} X_i, X_j \end{pmatrix}_{i,j=1,...,n}
\]

in particular:

\[
\mathbf{P} = \begin{pmatrix}
  P_{11} & P_{12} & P_{13} & \cdots & P_{1n} \\
  P_{21} & P_{22} & P_{23} & \cdots & P_{2n} \\
  P_{31} & P_{32} & P_{33} & \cdots & P_{3n} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  P_{n1} & P_{n2} & P_{n3} & \cdots & P_{nn}
\end{pmatrix}
\]

Note that:

- Diagonal elements are the \textit{variance} of parameter \( x_i \)
- Off-diagonal elements are the correlation of \( x_i \) and \( x_j \)
- \( \mathbf{P} \) has the following properties:
  - symmetric \( P_{ij} = P_{ji} \)
  - positive definite \( \forall \mathbf{x} \neq 0 : \mathbf{x}^T \mathbf{P} \mathbf{x} > 0 \)
  - has a positive determinant: \( |\mathbf{P}| := \det(\mathbf{P}) > 0 \)
Example 1

\[ P = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \]
Example 2

\[ P = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix} \]
Example 3

\[ P = \begin{pmatrix} 0.1 & 0 \\ 0 & 1 \end{pmatrix} \]
Example 4

\[ P = \begin{pmatrix} 0.1 & 0.3 \\ 0.3 & 1 \end{pmatrix} \]
Statistical Models for SENSORS
Linear Gaussian Case

- The measurement $z$ at time $k$ is modeled as a RV.
- Relationship to state is described by the "measurement equation":
  
  $$z_k = H_k x_k + v_k$$

where
- $H$ is the measurement function
- $v$ is the measurement noise.
Ideal Sensor for Tracking

An ideal sensor is given if

- $H$ is linear.
- $\nu$ is zero-mean and Gaussian distributed, that is
  - $E[\nu_k] = 0$
  - $p(\nu_k) = \mathcal{N}(\nu_k; O, R_k)$
  - in short: $\nu_k \sim \mathcal{N}(\nu_k; O, R_k)$

\[
\iff z_k \sim \mathcal{N}(z_k; H_kx_k, R_k) = p(z_k|x_k)
\]

Why is this sensor ideal?
If transition model is also linear and Gaussian, an optimal closed solution can be given -> Kalman filter
Example 1

Assume the state is given by

\[
x_k = \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix}
\]

then, the measurement equation for a position sensor (GPS e.g.) is

\[
z_k = H_k(x_k) + v_k = \begin{pmatrix} x \\ y \end{pmatrix} + \text{noise}
\]

therefore

\[
H_k = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}
\]
Non-linear Sensor Model

What can we do, if we have a non-linear measurement function:

$$z_k = h(x_k) + v_k$$

- What to do then?
  - e.g. “Be wise, linearize!”

$$\tilde{H}_k := \left. \frac{dh(x)}{dx} \right|_{x=x_k}$$

- More details in section “Non-linear filtering”.

© Fraunhofer FKIE
Example 2

Assume the state is given by

\[
\mathbf{x}_k = \begin{pmatrix}
    x \\
    y \\
    z \\
    \dot{x} \\
    \dot{y} \\
    \dot{z}
\end{pmatrix} \in \mathbb{R}^6
\]

then, the measurement equation of a air surveillance radar is

\[
\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \nu_k
\]

\[
= \begin{pmatrix}
    \text{range } r \\
    \text{bearing } \alpha \\
    \text{elevation } \epsilon
\end{pmatrix} + \text{noise}
\]

\[
= \left( \sqrt{x^2 + y^2 + z^2} \right) + \text{noise}
\]

\[
= \begin{pmatrix}
    \arctan\left(\frac{y}{x}\right) \\
    \arctan\left(\frac{z}{x}\right)
\end{pmatrix} + \text{noise}
\]