0. Mathematical Foundation

Probability Density Function:

Normalized: $\int p(x)\, dx = 1$

Positive: $p(x) > 0$

Probability: $P(a < x < b) = \int_a^b p(x)\, dx$

Conditional PDF: $p(x|y) = \dfrac{p(x,y)}{p(y)}$

Marginal Density: $p(x) = \int p(x,y)\, dy$

Bayes Theorem: $p(x|y) = \dfrac{p(y|x)\, p(x)}{p(y)}$

Gaussian Distribution: $p(x|y) = \dfrac{p(y|x)\, p(x)}{p(y)}$

Gaussian Product Formula:

$$\boxed{\begin{aligned} &\mathcal{N}\big(\mathrm{z};\ \mathrm{Hx},\ \mathrm{R}\big)\,\mathcal{N}\big(\mathrm{x};\ \mathrm{y},\ \mathrm{P}\big) \\ &= \underbrace{\mathcal{N}\big(\mathrm{z};\ \mathrm{Hy},\ \mathrm{S}\big)}_{\text{independent of } \mathrm{x}} \times \begin{cases} \mathcal{N}\big(\mathrm{x};\ \mathrm{y} + \mathrm{W}\nu,\ \mathrm{P} - \mathrm{WSW}^\top\big) \\ \mathcal{N}\big(\mathrm{x};\ \mathrm{Q}(\mathrm{P}^{-1}\mathrm{y} + \mathrm{H}^\top\mathrm{R}^{-1}\mathrm{z}),\ \mathrm{Q}\big) \\ \mathcal{N}\big(\mathrm{x};\ \mathrm{y} + \mathrm{W}\nu,\ (\mathrm{I} - \mathrm{WH})\mathrm{P}\big) \end{cases} \\ &\nu = \mathrm{z} - \mathrm{Hy}, \quad \mathrm{S} = \mathrm{HPH}^\top + \mathrm{R}, \quad \mathrm{W} = \mathrm{PH}^\top\mathrm{S}^{-1}, \quad \mathrm{Q}^{-1} = \mathrm{P}^{-1} + \mathrm{H}^\top\mathrm{R}^{-1}\mathrm{H}. \end{aligned}}$$

Idea: Represent a product of two gaussians dependent on x through a different product with one gaussian independent of x

Proof: Interpret $\mathcal{N}(\mathrm{z};\ \mathrm{Hx},\ \mathrm{R})\,\mathcal{N}(\mathrm{x};\ \mathrm{y},\ \mathrm{P})$ as a joint pdf $p(\mathrm{z}|\mathrm{x})p(\mathrm{x}) = p(\mathrm{z},\mathrm{x})$.

Show that $p(\mathrm{z},\mathrm{x})$ is a GAUSSian: $p(\mathrm{z},\mathrm{x}) = \mathcal{N}\left(\binom{\mathrm{z}}{\mathrm{x}};\ \binom{\mathrm{Hy}}{\mathrm{y}},\ \begin{pmatrix} \mathrm{S} & \mathrm{HP} \\ \mathrm{PH}^\top & \mathrm{P} \end{pmatrix}\right)$.

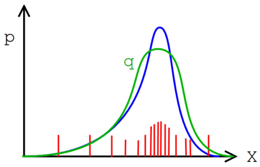
Calculate from $p(\mathrm{z},\mathrm{x})$ the marginal and conditional pdfs $p(\mathrm{z})$ and $p(\mathrm{x}|\mathrm{z})$.

From $p(\mathrm{z},\mathrm{x}) = p(\mathrm{z}|\mathrm{x})p(\mathrm{x}) = p(\mathrm{x}|\mathrm{z})p(\mathrm{z}) = p(\mathrm{x},\mathrm{z})$ we obtain the result.

Definition: Importance Sampling

Sample from a proposal distribution and weight by the ratio between the proposal and target distribution to approximate the target distribution.

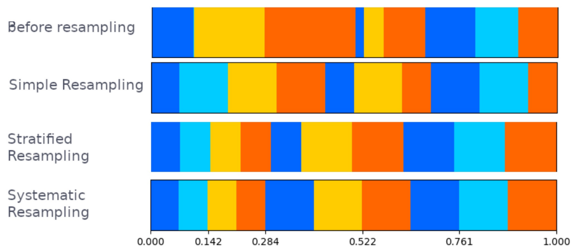
$$\mathbf{x^i} \sim q(\mathbf{x}|\mathbf{Z}),\, w^i \propto \frac{p(\mathbf{x^i}|\mathbf{Z})}{q(\mathbf{x^i}|\mathbf{Z})}$$



Definition: Stratified Sampling

Divide interval into N equally spaced bins and take one sample uniformly from each.

$$\left\{\left(0, \frac{s}{N}\right], \left(\frac{s}{N}, 2 * \frac{s}{N}\right], \ldots, \left((N-1) * \frac{s}{N}, 1\right]\right\}$$



Lemma: Inversion Lemma

$$\mathrm{Q} = \underbrace{(\mathrm{H}^\top \mathrm{R}^{-1}\mathrm{H} + \mathrm{P}^{-1})^{-1}}_{\text{2. Version!}} = \mathrm{P} - \mathrm{P}\mathrm{H}^\top \mathrm{S}^{-1}\mathrm{H}\mathrm{P} \qquad \text{Inversionslemma!}$$

$$= \underbrace{\mathrm{P} - \mathrm{W}\mathrm{S}\mathrm{H}^\top}_{\text{1. Version!}} = \underbrace{(\mathrm{I} - \mathrm{W}\mathrm{H})\mathrm{P}}_{\text{3. Version!}}, \qquad \text{mit:} \quad \mathrm{W} = \mathrm{P}\mathrm{H}^\top \mathrm{S}^{-1}$$

Definition: Moment Matching

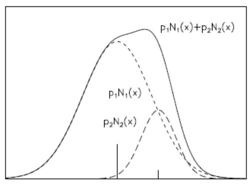
Approach to approximate an arbitrary pdf with a gaussian by matching the first and second moment of the pdf:

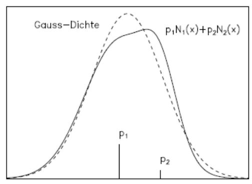
$p(x)$ with $\mathbb{E}[x] = \mathrm{x}$, $\mathbb{C}[x] = \mathrm{P}$ by $p(x) \approx \mathcal{N}(x;\, \mathrm{x},\, \mathrm{P})$

Approximation:

$$\mathrm{x} = \sum_H p_H \mathrm{x}_H$$

$$\mathrm{P} = \sum_H p_H \left\{ \mathrm{P}_H + \overbrace{(\mathrm{x}_H - \mathrm{x})(\mathrm{x}_H - \mathrm{x})^\top}^{\text{spread term}} \right\}$$





Matrix Products

Kronecker product:

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & & \ddots & \vdots \\ a_{n_1 1}\mathbf{B} & a_{n_1 2}\mathbf{B} & \cdots & a_{n_1 n_2}\mathbf{B} \end{pmatrix}$$

Khatri-Rao product:

$$\mathbf{A} \odot \mathbf{B} = (\mathbf{a}_1 \otimes \mathbf{b}_1, \mathbf{a}_2 \otimes \mathbf{b}_2, \ldots, \mathbf{a}_m \otimes \mathbf{b}_m)$$

Hadamard product:

$$\mathbf{A} * \mathbf{B} = \begin{pmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1n}b_{1n} \\ \vdots & & \ddots & \vdots \\ a_{n_1 1}b_{n_1 1} & a_{n_1 2}b_{n_1 2} & \cdots & a_{n_1 n_2}b_{n_1 n_2} \end{pmatrix}$$

Matrix Decomposition with SVD

Decomposition: $\boxed{\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*}$

$\mathbf{U}$ as an unitary matrix

$\mathbf{\Sigma}$ as an matrix of $[[\mathbf{Diag}(\sigma_1 \ldots \sigma_r), \mathbf{0}], [\mathbf{0}, \mathbf{0}]]$ with σ_i as the singular values

$\mathbf{V}^*$ as the adjoint of an unitary matrix $\mathbf{V}$

→ choose k largest eigenvalues for lower rank approximation

Matrix Factorization (Alternating Least Squares)

Goal: Factorize matrix Y in two matrices A, X with a minimal error matrix E:

$$\mathbf{Y} = \mathbf{A}\,\mathbf{X}^{\mathbf{T}} + \mathbf{E}$$

Approach: Minimize the Frobenius norm between the matrices:

$$J = \frac{1}{2}\left\|\mathbf{Y} - \mathbf{A}\mathbf{X}^{\mathbf{T}}\right\|_F^2 \qquad \|A\|_F = \sqrt{\textstyle\sum_{i,j} a_{ij}^2}$$

Algorithm:

0. Initialize the matrices $\mathbf{X}, \mathbf{A}$ randomly
1. Solve for $\mathbf{X}$ with fixed $\mathbf{A}$:
$$\mathbf{X} = \left(\left(\mathbf{A}^{\mathbf{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathbf{T}}\mathbf{Y}\right)^T$$
2. Solve for $\mathbf{A}$ with fixed $\mathbf{X}$:
$$\mathbf{A} = \left(\left(\mathbf{X}\,\mathbf{X}^{\mathbf{T}}\right)^{-1}\mathbf{X}\,\mathbf{Y}\right)^T$$
3. Repeat 1. and 2. until convergence

Stopping criteria:

Use root mean squared error

Test the fit

$$f_r = 1 - \frac{\|Y - A\,X^T\|_F}{\|Y\|_F}$$

Check if the relative fit $\left|f_r^{(n)} - f_r^{(n-1)}\right| < \lambda$

A maximum number of iterations is reached

Tensor Vectorization

Reindex into one running index

$$[\mathrm{vec}(\mathcal{T})]_{I(i_1,\ldots,i_D)} = [\mathcal{T}]_{i_1,\ldots,i_D}$$

$$I(i_1,\ldots,i_D) = 1 + \sum_{k=1}^{D}(i_k - 1)\prod_{l=1}^{k-1} N_l$$

- Example: Consider the order 3 tensor $\mathfrak{X}$ which is a stack of the following matrices:

$$\begin{pmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{pmatrix}.$$

$$\Rightarrow \quad \mathrm{vec}(\mathcal{X}) = (1, \ldots, 24)^{\top}.$$

Tensor Matricisation

Reindex using two indices into a matrix

$$[\mathcal{T}_{(d)}]_{I(i_1,\ldots,i_D)} = [\mathcal{T}]_{i_1,\ldots,i_D}$$

$$I(i_1,\ldots,i_D) = \left(i_d, 1 + \sum_{\substack{k=1 \\ k\neq d}}^{D}(i_k - 1)\prod_{l=1}^{k-1} N_l\right).$$

- Example: Consider the order 3 tensor $\mathfrak{X}$ from the left:

$$\Rightarrow \quad \mathbf{X}_{(1)} = \begin{pmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{pmatrix}$$

$$\mathbf{X}_{(2)} = \begin{pmatrix} 1 & 2 & 3 & 13 & 14 & 15 \\ 4 & 5 & 6 & 16 & 17 & 18 \\ 7 & 8 & 9 & 19 & 20 & 21 \\ 10 & 11 & 12 & 22 & 23 & 24 \end{pmatrix}$$

$$\mathbf{X}_{(3)} = \begin{pmatrix} 1 & 2 & \cdots & 11 & 12 \\ 13 & 14 & \cdots & 23 & 24 \end{pmatrix}$$

Alternating Least Squares for Tensors

Generalize SVD for High dimensional Tensors

$$T \approx [\mathbf{U_1}, \dots, \mathbf{U_D}] \Rightarrow E = T - [\mathbf{U_1}, \dots, \mathbf{U_D}]$$

Minimize the Error Tensor by the cost function $J = \frac{1}{2}||\mathrm{E}||_F^2$

0. Initialize $\mathbf{U}_\mathrm{d}^{(0)} \in \mathbb{R}^{\mathrm{N_d x R}} \twoheadrightarrow \mathbb{R}$ for $d = 1, \dots, D$ with random values.
1. For $d = 1, \dots, D$: Solve for $\mathbf{U}_\mathrm{d}$ with fixed $\mathbf{U}_l$ for $l \neq d$:

$$\mathbf{U}_\mathrm{d}^{(n+1)} = T_{(d)} \left(\bigodot_{l \neq d} \mathbf{U}_l^{(n)} \right) \left(\circledast_{l \neq d} \mathbf{U}_l^{(n)T} \mathbf{U}_l^{(n)} \right)^\dagger$$

2. Repeat 1. until convergence is observed

Alternating Least Squares for Tensor Deflation

Use $\mathbf{T}_{(\mathrm{d})} = \mathbf{A}_\mathrm{d} (\odot_{l \neq d} \mathbf{A}_\mathrm{d})^T$

Use the fact: $(\odot_{l \neq d} \mathbf{A}_l)^T (\odot_{l \neq d} \mathbf{U}_l) = (\circledast_{l \neq d} \mathbf{A}_l^T \mathbf{U}_l)^\dagger$

0. Initialize $\mathbf{U}_\mathrm{d}^{(0)} \in \mathbb{R}^{\mathrm{N_d x R}} \twoheadrightarrow \mathbb{R}$ for $d = 1, \dots, D$ with random values.
1. For $d = 1, \dots, D$: Solve for $\mathbf{U}_\mathrm{d}$ with fixed $\mathbf{U}_l$ for $l \neq d$:

$$\mathbf{U}_\mathrm{d}^{(n+1)} = \mathbf{A}_\mathrm{d} \left(\circledast_{l \neq d} \mathbf{A}_l^T \mathbf{U}_l^{(n)} \right) \left(\circledast_{l \neq d} \mathbf{U}_l^{(n)T} \mathbf{U}_l^{(n)} \right)^\dagger$$

2. Repeat 1. until convergence is observed (Stopping Criteria can be calculated simply)

χ^2- Distribution

Given: n-dimensional gaussian distributed random variable x with: $\mathbb{E}[\mathrm{x}] = \bar{\mathrm{x}}$ $\quad \mathbb{C}[\mathrm{x}] = \mathbf{P}$

Then: $q = (\mathrm{x} - \bar{\mathrm{x}})^\top \mathbf{P}^{-1} (\mathrm{x} - \bar{\mathrm{x}})$ is χ^2-distributed with:

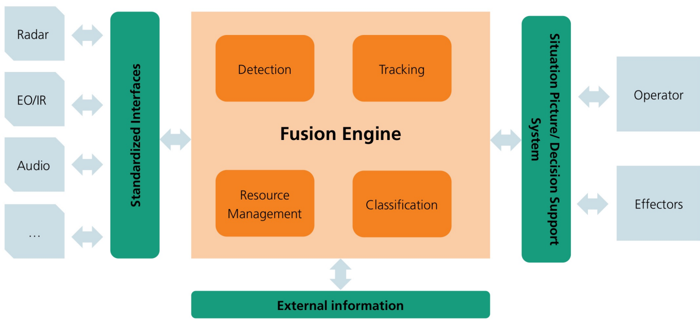
$$\boxed{p(q) = \frac{q^{\frac{n-1}{2}} \mathrm{e}^{-\frac{q}{2}}}{2^{\frac{n}{2}} \Gamma(n/2)}, \quad q \sim \chi_n^2}$$

$n \backslash Q$	0.99	0.975	0.95	0.90	0.75	0.5	0.25	0.10	0.05	0.025	0.01	5E-3	1E-3
1	2E-4	.001	.003	.016	.102	.455	1.32	2.71	3.84	5.02	6.63	7.88	10.8
2	.020	.051	.103	.211	.575	.139	.277	4.61	5.99	7.38	9.21	10.6	13.8
3	.115	.216	.352	.584	1.21	2.37	4.11	6.25	7.81	9.35	11.3	12.8	16.3
4	.297	.484	.711	1.06	1.92	3.36	5.39	7.78	9.49	11.1	13.3	14.9	18.5
5	.554	.831	1.15	1.61	2.67	4.35	6.63	9.24	11.1	12.8	15.1	16.7	20.5
6	.872	1.24	1.64	2.20	3.35	5.35	7.84	10.6	12.6	14.4	16.8	18.5	22.5
7	1.24	1.69	2.17	2.83	4.25	6.35	9.04	12.0	14.1	16.1	18.5	20.3	24.3
8	1.65	2.18	2.73	3.49	5.07	7.34	10.2	13.4	15.5	17.5	20.1	22.0	26.1
9	2.09	2.70	3.33	4.17	5.90	8.34	11.4	14.7	17.0	19.0	21.7	23.6	27.9
10	2.56	3.25	3.94	4.87	6.74	9.34	12.5	16.0	18.3	20.5	23.2	25.2	29.6
11	3.05	3.82	4.57	5.58	7.58	10.3	13.7	17.3	19.7	22.0	24.7	26.8	31.3
12	3.57	4.40	5.23	6.30	8.44	11.3	14.8	18.5	21.0	23.3	26.2	28.3	32.9
13	4.11	5.01	5.90	7.04	9.30	12.3	16.0	19.8	22.4	24.7	27.7	29.8	34.5
14	4.66	5.63	6.57	7.79	10.2	13.3	17.1	21.1	23.7	26.1	29.1	31.3	36.1
15	5.23	6.26	7.26	8.55	11.0	14.3	18.2	22.3	25.0	27.5	30.6	32.8	37.7
16	5.81	6.91	7.96	9.31	11.9	15.3	19.4	23.5	26.3	28.8	32.0	34.3	39.3
17	6.41	7.56	8.67	10.1	12.8	16.3	20.5	24.8	27.6	30.2	33.4	35.7	40.8
18	7.01	8.23	9.40	10.9	13.7	17.3	21.6	26.0	28.9	31.5	34.8	37.2	42.3
19	7.63	8.91	10.1	11.7	14.6	18.3	22.7	27.2	30.1	32.9	36.2	38.6	43.8
20	8.26	9.60	10.9	12.4	15.5	19.3	23.8	28.4	31.4	34.2	37.6	40.0	45.3
25	11.5	13.1	14.6	16.5	19.9	24.3	29.3	34.4	37.7	40.6	44.3	46.9	52.6
30	15.0	16.8	18.5	20.6	24.5	29.3	34.8	40.3	43.8	47.0	50.9	53.7	59.7
40	22.2	24.4	26.5	29.1	33.7	39.3	45.6	51.8	55.8	59.3	63.7	66.8	73.4
50	29.7	32.4	34.8	37.7	43.0	49.3	56.3	63.2	67.5	71.4	76.2	79.5	86.7
60	37.5	40.5	43.2	46.5	52.3	59.3	67.0	74.4	79.1	83.3	88.4	92.0	99.6
70	45.4	48.8	51.7	55.3	61.7	69.3	77.6	85.5	90.5	95.0	100	104	112
80	53.5	57.2	60.4	64.2	71.1	79.3	88.1	96.6	102	107	112	116	125
90	61.8	65.6	69.1	73.3	80.6	89.3	98.6	108	113	118	124	128	137
100	70.1	74.2	77.9	82.4	90.1	99.3	109	118	124	130	136	140	149
$\mathcal{G}()$	-2.33	-1.96	-1.64	-1.28	-.675	0	.675	1.28	1.64	1.96	2.33	2.58	3.09

1. Sensor Data Fusion Basics

Generic Sensor Data Fusion System

Fuses data from heterogenous flows of information into a combined representation of a world state that can be used in downstream tasks. This draws a situational picture in the computer



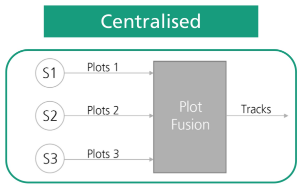
Detection: Detect target measurements in a clutter of data and assign to corresponding targets

Tracking: Fuse measurements along time to estimate and track the state of a target

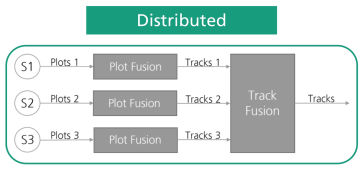
Resource Management: Manage resources of our measurement and fusion plattform based on the tracked targets to allow efficient processing

Classification: Classify what we see based on our estimated states

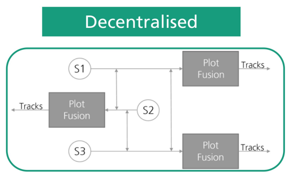
Fusion Architectures



Centralised: Fuse measurements from different sensors in one fusion engine to produce a single track



Distributed: Fuse measurements at sensor level and fuse all tracks together in a centralised fusion engine



Decentralised: Sensors send measurements to other sensors but perform the fusion completly independent

Advanced Sensor Data Fusion System

Level 0: Subobject data association and estimation

Kombination von Daten auf dem Signal Level um grundlegende Informationen über die Charakteristiken des Ziels zu erhalten

Level 1: Object Refinement

Kombination von Sensordaten um Position, Geschwindigkeit, Identität oder weitere Attribute eines Objekts zu schätzen.

Nutzung verfügbarer A-Priori-Information

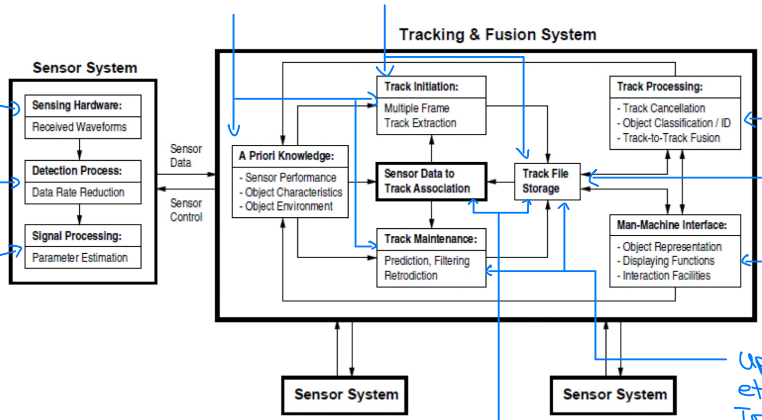
Etablierung neuer Objekte, Wiederaufnahme von Tracks

Fusion bereits vorverarbeiteter Track-Information

Empfangende Zielsignale

Vorgelagerter Entscheidungsprozess zur Datenreduktion

Tracking-Input: geschätzte Zielparameter



Schnittstelle zum Nutzer

Updating bereits etablierter Target Tracks

Kernelement: Zuordnung Sensordaten / etablierte Tracks

Level 2: Situation Refinement

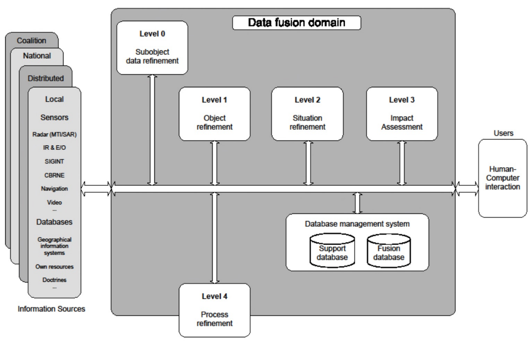
Dynamische Entwicklung von Beziehungen zwischen Objekten und Events im Kontext der Umgebung.

Level 3: Significance Estimation

Projektion des momentanen Zustands in die Zukunft um über Bedrohungen, freundliche und feindliche Schwächen und mögliche Operationen zu schlussfolgern.

Level 4: Process Refinement

Meta-Prozess, der den Sensordatenfusions-Prozess überwacht und versucht die Echtzeit-Performanz zu optimieren.



Range, Azimuth Measurements

Measurements: $z_k = (r_k, \varphi_k)$, $R_k = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\varphi^2 \end{pmatrix}$

Cartesian → Polar: $\boxed{t(z) = r \begin{pmatrix} \cos(\varphi) \\ \sin(\varphi) \end{pmatrix}}$

Transform gaussian random variable: $\mathcal{N}(z; x, R) \xrightarrow{\text{linear transform } z' = t + Tz} \mathcal{N}(z'; t + Tx, TRT^T)$

Approximate through Taylor series expansion

$$\boxed{t(z) \approx t(s') + T(s - z') \text{ with } T = \frac{\partial t(s')}{\partial s'} = \begin{pmatrix} \cos(\varphi) & -r' * \sin(\varphi) \\ \sin(\varphi) & r * \cos(\varphi) \end{pmatrix} = \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & r \end{pmatrix}}$$

Cartesian error covariance: $\boxed{TRT^T = \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix} \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & (r\sigma_\varphi)^2 \end{pmatrix} \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix}^T}$

Ground Moving Target Indication Radar

Goal: Track targets on the ground

Problem: Using a standard radar we get a lot of clutter from the ground that we cannot distinguish target measurements from

Idea: Detect the Doppler shift in frequency induced by moving targets to detect target measurements

Doppler blindness: Targets under a given radial velocity with respect to the radar (MDV) induce a Doppler shift too small to detect

2. Filtering Techniques

Bayesian Filtering

Prediction: Predict next state through an evolution model

$$p(\mathbf{x_k}|\mathbf{Z^{k-1}}) = \int p(\mathbf{x_k}, \mathbf{x_{k-1}}|\mathbf{Z^{k-1}})\, d\mathbf{x_{k-1}} \qquad \text{Marginalization over } x_{k-1}$$

$$= \int p(\mathbf{x_k}|\mathbf{x_{k-1}}, \mathbf{Z^{k-1}}) p(\mathbf{x_{k-1}}|\mathbf{Z^{k-1}})\, d\mathbf{x_{k-1}} \qquad \text{Conditional PDF Rule}$$

$$= \int \underbrace{p(\mathbf{x_k}|\mathbf{x_{k-1}})}_{\text{dynamic model}}\, \underbrace{p(\mathbf{x_{k-1}}|\mathbf{Z^{k-1}})}_{\text{Previous posterior}}\, d\mathbf{x_{k-1}} \qquad \text{Markov property}$$

prediction: $p(\mathrm{x}_{k-1}|\mathcal{Z}^{k-1}) \xrightarrow[\text{road maps}]{\text{dynamics model}} p(\mathrm{x}_k|\mathcal{Z}^{k-1})$

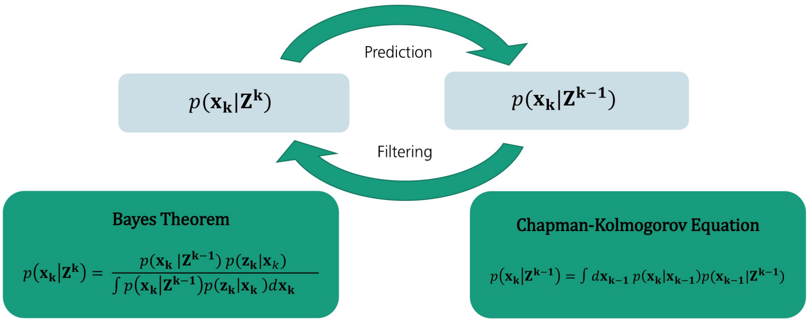
filtering: $p(\mathrm{x}_k|\mathcal{Z}^{k-1}) \xrightarrow[\text{sensor model}]{\text{sensor data } Z_k} p(\mathrm{x}_k|\mathcal{Z}^{k})$

retrodiction: $p(\mathrm{x}_{l-1}|\mathcal{Z}^{k}) \xleftarrow[\text{dynamics model}]{\text{filtering output}} p(\mathrm{x}_l|\mathcal{Z}^{k})$

Filtering: Incorporate the measurement to correct the predicted state

$$p(\mathbf{x_k}|\mathbf{Z^k}) = p(\mathbf{x_k}|\mathbf{Z^{k-1}}, \mathbf{z_k}) \qquad \text{Unpack } Z^k$$

$$= \underbrace{\frac{\overbrace{p(\mathbf{z_k}|\mathbf{x_k}, \mathbf{Z^{k-1}})}^{\text{likelihood}}\, \overbrace{p(\mathbf{x_k}|\mathbf{Z^{k-1}})}^{\text{prior}}}{\int p(\mathbf{z_k}|\mathbf{x_k})\, p(\mathbf{x_k}|\mathbf{Z^{k-1}})\, d\mathbf{x_k}}}_{\text{normalisation Constant}} \qquad \text{Bayes theorem}$$



Retrodiction:

$$p(\mathrm{x}_{l-1}|\mathcal{Z}^{k}) \xleftarrow[\text{dynamics model}]{\text{filtering output}} p(\mathrm{x}_l|\mathcal{Z}^{k})$$

$$p(\mathrm{x}_l|\,\mathrm{x}_{l+1}, \mathcal{Z}^k) = p(\mathrm{x}_l|\,\mathrm{x}_{l+1}, \mathcal{Z}^l) = \frac{p(\mathrm{x}_{l+1}|\,\mathrm{x}_l)\; p(\mathrm{x}_l|\,\mathcal{Z}^l)}{\int d\mathrm{x}_l\; \underbrace{p(\mathrm{x}_{l+1}|\,\mathrm{x}_l)}_{\text{dynamics model}}\; \underbrace{p(\mathrm{x}_l|\,\mathcal{Z}^l)}_{\text{filtering } t_l}}$$

Piecewise Constant White Acceleration Model

Assume constant acceleration in each step and integrate up

$$\mathbf{F_{k|k-1}} = \begin{pmatrix} \mathbf{I} & \Delta_{t_k - t_{k-1}}\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

$$\mathbf{D_{k|k-1}} = \sigma^2 \begin{pmatrix} \frac{1}{4}\Delta^4_{t_k-t_{k-1}}\mathbf{I} & \frac{1}{2}\Delta^3_{t_k-t_{k-1}}\mathbf{I} \\ \frac{1}{2}\Delta^3_{t_k-t_{k-1}}\mathbf{I} & \Delta^2_{t_k-t_{k-1}}\mathbf{I} \end{pmatrix}$$

Kalman Filter

Prediction:

$$\mathcal{N}(\mathbf{x}_{k-1};\, \mathbf{x}_{k-1|k-1},\, \mathbf{P}_{k-1|k-1}) \xrightarrow[\mathbf{F}_{k|k-1},\, \mathbf{D}_{k|k-1}]{\text{dynamics model}} \mathcal{N}(\mathbf{x}_k;\, \mathbf{x}_{k|k-1},\, \mathbf{P}_{k|k-1})$$

$$\begin{aligned} \mathbf{x}_{k|k-1} &= \mathbf{F}_{k|k-1}\mathbf{x}_{k-1|k-1} \\ \mathbf{P}_{k|k-1} &= \mathbf{F}_{k|k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k|k-1}{}^{\top} + \mathbf{D}_{k|k-1} \end{aligned}$$

Filtering:

$$\mathcal{N}(\mathbf{x}_k;\, \mathbf{x}_{k|k-1},\, \mathbf{P}_{k|k-1}) \xrightarrow[\text{sensor model: } \mathbf{H}_k,\, \mathbf{R}_k]{\text{current measurement } \mathbf{z}_k} \mathcal{N}(\mathbf{x}_k;\, \mathbf{x}_{k|k},\, \mathbf{P}_{k|k})$$

$$\begin{aligned} \mathbf{x}_{k|k} &= \mathbf{x}_{k|k-1} + \mathbf{W}_{k|k-1}\boldsymbol{\nu}_{k|k-1}, & \boldsymbol{\nu}_{k|k-1} &= \mathbf{z}_k - \mathbf{H}_k\mathbf{x}_{k|k-1} \\ \mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{W}_{k|k-1}\mathbf{S}_{k|k-1}\mathbf{W}_{k|k-1}{}^{\top}, & \mathbf{S}_{k|k-1} &= \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k{}^{\top} + \mathbf{R}_k \\ \mathbf{W}_{k|k-1} &= \mathbf{P}_{k|k-1}\mathbf{H}_k{}^{\top}\mathbf{S}_{k|k-1}{}^{-1} & & \text{'KALMAN gain matrix'} \end{aligned}$$

Retrodiction:

$$\mathcal{N}(\mathbf{x}_l;\, \mathbf{x}_{l|k},\, \mathbf{P}_{l|k}) \xleftarrow[\text{dynamics model}]{\text{filtering, prediction}} \mathcal{N}(\mathbf{x}_{l+1};\, \mathbf{x}_{l+1|k},\, \mathbf{P}_{l+1|k})$$

$$\begin{aligned} \mathbf{x}_{l|k} &= \mathbf{x}_{l|l} + \mathbf{W}_{l|l+1}(\mathbf{x}_{l+1|k} - \mathbf{x}_{l+1|l}), & \mathbf{W}_{l|l+1} &= \mathbf{P}_{l|l}\mathbf{F}^{\top}_{l+1|l}\mathbf{P}^{-1}_{l+1|l} \\ \mathbf{P}_{l|k} &= \mathbf{P}_{l|l} + \mathbf{W}_{l|l+1}(\mathbf{P}_{l+1|k} - \mathbf{P}_{l+1|l})\mathbf{W}_{l|l+1}{}^{\top} \end{aligned}$$

Assumptions:

- Markov assumption
- gaussian-distributed random variables
- independent measurements

Prediction derivation:

$$p(\mathbf{x_k}|\mathbf{Z^{k-1}}) = \int p(\mathbf{x_k}|\mathbf{x_{k-1}})\, p(\mathbf{x_{k-1}}|\mathbf{Z^{k-1}})\, d\mathbf{x_{k-1}}$$

$$= \int \mathcal{N}(\mathbf{x_k}; \mathbf{Fx_{k-1}}, \mathbf{D})\, \mathcal{N}(\mathbf{x_{k-1}}; \mathbf{x_{k-1|k-1}}, \mathbf{P_{k-1|k-1}})\, d\mathbf{x_{k-1}}$$

$$= \int \mathcal{N}(\mathbf{x_k}; \mathbf{Fx_{k-1|k-1}}, \mathbf{FP_{k-1|k-1}F^T} + \mathbf{D}) \times$$

$$\mathcal{N}\Big(\mathbf{x_{k-1}}; \mathbf{x_{k-1|k-1}} + \left(\mathbf{P_{k-1|k-1}}\,\mathbf{F^T}\left[\mathbf{FP_{k-1|k-1}F^T} + \mathbf{D}\right]^{-1}\right)(\mathbf{x_k} - \mathbf{Fx_{k-1|k-1}}), \mathbf{P} - \left(\mathbf{P_{k-1|k-1}}\,\mathbf{F^T}\left[\mathbf{FP_{k-1|k-1}F^T} + \mathbf{D}\right]^{-1}\right)\left[\mathbf{FP_{k-1|k-1}F^T} + \mathbf{D}\right]\left(\mathbf{P_{k-1|k-1}}\,\mathbf{F^T}\left[\mathbf{FP_{k-1|k-1}F^T} + \mathbf{D}\right]^{-1}\right)^{\mathbf{T}}\Big)\, d\mathbf{x_{k-1}}$$

$$= \mathcal{N}(\mathbf{x_k}; \mathbf{Fx_{k-1|k-1}}, \mathbf{FP_{k-1|k-1}F^T} + \mathbf{D})$$

Filtering derivation:

$$p(\mathbf{x_k}|\mathbf{Z^k}) = \frac{p(\mathbf{z_k}|\mathbf{x_k})p(\mathbf{x_k}|\mathbf{Z^{k-1}})}{\int p(\mathbf{z_k}|\mathbf{x_k})p(\mathbf{x_k}|\mathbf{Z^{k-1}})d\mathbf{x_k}}$$

$$= \frac{1}{constant} \times p(\mathbf{z_k}|\mathbf{x_k})p(\mathbf{x_k}|\mathbf{Z^{k-1}})$$

$$= \frac{1}{constant} \times \mathcal{N}(\mathbf{z_k}; \mathbf{H}x_k, \mathbf{R_k})\, \mathcal{N}(\mathbf{x_k}; \mathbf{x_{k|k-1}}, \mathbf{P_{k|k-1}})$$

$$= \frac{1}{constant} \times \mathcal{N}(\mathbf{z_k}; \mathbf{Hx_{k|k-1}}, \mathbf{HP_{k|k-1}H^T} + \mathbf{R})\, \mathcal{N}\Big(\mathbf{x_k}; \mathbf{x_{k|k-1}} + \left(\mathbf{P_{k|k-1}H^T}\left[\mathbf{HP_{k|k-1}H^T} + \mathbf{R}\right]^{-1}\right)(\mathbf{z} - \mathbf{Hx_{k|k-1}}),\ \mathbf{P_{k|k-1}} - \left(\mathbf{P_{k|k-1}H^T}\left[\mathbf{HP_{k|k-1}H^T} + \mathbf{R}\right]^{-1}\right)\left[\mathbf{HP_{k|k-1}H^T} + \mathbf{R}\right]\left(\mathbf{P_{k|k-1}H^T}\left[\mathbf{HP_{k|k-1}H^T} + \mathbf{R}\right]^{-1}\right)^{\mathbf{T}}\Big)$$

$$p(\mathbf{x_k}|\mathbf{Z^k}) = \frac{1}{constant} \times \mathcal{N}(\mathbf{z_k}; \mathbf{Hx_{k|k-1}}, \mathbf{HP_{k|k-1}H^T} + \mathbf{R})\, \mathcal{N}\Big(x_k; \mathbf{x_{k|k-1}} + \left(\mathbf{P_{k|k-1}H^T}\left[\mathbf{HP_{k|k-1}H^T} + \mathbf{R}\right]^{-1}\right)(\mathbf{z} - \mathbf{Hx_{k|k-1}}), \mathbf{P_{k|k-1}} - \left(\mathbf{P_{k|k-1}H^T}\left[\mathbf{HP_{k|k-1}H^T} + \mathbf{R}\right]^{-1}\right)\left[\mathbf{HP_{k|k-1}H^T} + \mathbf{R}\right]\left(\mathbf{P_{k|k-1}H^T}\left[\mathbf{HP_{k|k-1}H^T} + \mathbf{R}\right]^{-1}\right)^{\mathbf{T}}\Big)$$

Consider: $\int p(\mathbf{z_k}|\mathbf{x_k})p(\mathbf{x_k}|\mathbf{Z^{k-1}})d\mathbf{x_k} = \mathcal{N}(\mathbf{z_k}; \mathbf{Hx_{k|k-1}}, \mathbf{HP_{k|k-1}H^T} + \mathbf{R})$

$$p(\mathbf{x_k}|\mathbf{Z^k}) = \mathcal{N}\Big[\mathbf{x_k}; \mathbf{x_{k|k-1}} + \left(\mathbf{P_{k|k-1}H^T}\left[\mathbf{HP_{k|k-1}H^T} + \mathbf{R}\right]^{-1}\right)(\mathbf{z} - \mathbf{Hx_{k|k-1}}),$$

$$\mathbf{P_{k|k-1}} - \left(\mathbf{P_{k|k-1}H^T}\left[\mathbf{HP_{k|k-1}H^T} + \mathbf{R}\right]^{-1}\right)\left[\mathbf{HP_{k|k-1}H^T} + \mathbf{R}\right]\left(\mathbf{P_{k|k-1}H^T}\left[\mathbf{HP_{k|k-1}H^T} + \mathbf{R}\right]^{-1}\right)^{\mathbf{T}}\Big]$$

$$\boxed{\begin{aligned} \mathbf{x_{k|k}} &= \mathbf{x_{k|k-1}} + \mathbf{W_{k|k-1}}\boldsymbol{\nu} = \mathbf{W_{k|k-1}}(\mathbf{z} - \mathbf{Hx_{k|k-1}}) \\ \mathbf{P_{k|k}} &= \mathbf{P_{k|k-1}} - \mathbf{W_{k|k-1}}\,\mathbf{S_{k|k-1}}\,\mathbf{W^T_{k|k-1}} \end{aligned}}$$

$$\begin{aligned} \mathbf{W_{k|k-1}} &= \mathbf{P_{k|k-1}H^TS^{-1}_{k|k-1}} \\ \mathbf{S_{k|k-1}} &= \mathbf{HP_{k|k-1}H^T} + \mathbf{R_k} \end{aligned}$$

Extended Kalman Filter

Idea: Use non-linear evolution and measurement model and linearize via the Taylor series expansion using the Jacobian

Prediction	**Filtering**
$\mathbf{x}_{k\|k-1} = f(\mathbf{x}_{k-1\|k-1})$ $\mathbf{P}_{k\|k-1} = \mathbf{F}\mathbf{P}\mathbf{F}^T + \mathbf{D}$ with $\mathbf{F} = \frac{\partial f}{\partial x}\Big\|_{x_{k-1\|k-1}}$	$\mathbf{x}_{k\|k} = \mathbf{x}_{k\|k-1} + \mathbf{W}\left(\mathbf{z}_k - h(\mathbf{x}_{k-1\|k})\right)$ $\mathbf{P}_{k\|k} = \mathbf{P}_{k\|k-1} - \mathbf{W}\mathbf{S}\mathbf{W}^T$ with $\mathbf{H} = \frac{\partial h}{\partial x}\Big\|_{x_{k\|k1}}$ $\mathbf{S} = \mathbf{H}\mathbf{P}_{k\|k-1}\mathbf{H}^T + \mathbf{R}$ $\mathbf{W} = \mathbf{P}_{k\|k-1}\mathbf{H}^T\mathbf{S}^{-1}$

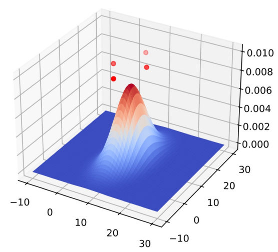
Unscented Kalman Filter

Idea: Approximate the probability distribution of the non-linear models instead of the functions itself.

1. Choose σ-points so that their mean is the mean of the density and the covariance is the density covariance
2. Push this points through the non-linear function
3. Calculate the new mean and covariance

Sigma points:

$$-1 < W^0 < 1$$
$$x^0 = x$$
$$x^i = x + \left(\sqrt{\frac{n}{1-W^0}P}\right)_i \quad \text{for all } i = 1, \dots, n$$
$$x^{i+n} = x - \left(\sqrt{\frac{n}{1-W^0}P}\right)_i \quad \text{for all } i = 1, \dots, n$$
$$W^j = \frac{1-W^0}{2n} \text{ for all } i = 1, \dots, 2n$$



Prediction:

1. Choose σ-points for the previous posterior density $\boldsymbol{x_i}$
2. Push this points through the non-linear dynamic model $f, \boldsymbol{D}$
3. Calculate the new mean and covariance $\boldsymbol{x^f}, \boldsymbol{P^f}$

$$\mathbf{x}_i^f = f(\mathbf{x}_i)$$
$$\mathbf{x}^f = \sum_{i=0}^{2n} W^i \mathbf{x}_i^f$$
$$\mathbf{P}^f = \sum_{i=0}^{2n} W^i (\mathbf{x}_i^f - \mathbf{x}^f)(\mathbf{x}_i^f - \mathbf{x}^f)^T + \mathbf{D}$$

Filtering:

1. Choose σ-points of the measurement density $\boldsymbol{z_i}$
2. Push this points through the non-linear sensor model with $h, \boldsymbol{R}$
3. Calculate the new mean and covariance $\boldsymbol{x}, \boldsymbol{P}$

$$\mathbf{z}_i^h = h(\mathbf{z}_i)$$
$$\mathbf{z}^h = \sum_{i=0}^{2n} W^i \mathbf{x}_i^h$$
$$Cov(\mathbf{z}^f) = \sum_{i=0}^{2n} W^i (\mathbf{z}_i^h - \mathbf{z}^h)(\mathbf{z}_i^h - \mathbf{z}^h)^T + \mathbf{R}$$
$$Cov(\mathbf{x}^f, \mathbf{z}^h) = \sum_{i=0}^{2n} W^i (\mathbf{x}_i^f - \mathbf{z}^h)(\mathbf{x}_i^f - \mathbf{z}^h)^T$$
$$\mathbf{W} = Cov(\mathbf{x}^f, \mathbf{z}^h) Cov^{-1}(\mathbf{z}^h)$$
$$\mathbf{x} = \mathbf{x}^f + \mathbf{W}(\mathbf{z} - \mathbf{z}^h)$$
$$\mathbf{P} = \mathbf{P}^f - \mathbf{W}\, Cov(\mathbf{z}^h)\mathbf{W}^T$$

Particle Filter

Idea: Employ Monte Carlo methods to solve prediction integral by performing a weighted sum over random paths

$$E[f(\mathbf{x}_{0:k})] = \int p(\mathbf{x}_{0:k}|Z^k) f(\mathbf{x}_{0:k}) d\mathbf{x}_{0:k} \qquad E[f(\mathbf{x}_{0:k})] \approx \sum_{i=1}^{N} w_k^i f(\mathbf{x}_{0:k}^{\mathbf{i}})$$

Sequential Particle Filter

Prediction: Update the position of each particle

$$p(\mathbf{x}_\mathbf{k}|\mathbf{Z}^{\mathbf{k}-1}) \approx \sum_{i=1}^{N} w_{k-1}^i p(\mathbf{x}_\mathbf{k}|\mathbf{x}_{\mathbf{k}-1}^{\mathbf{i}}) \approx \sum_{i=1}^{N} w_{k-1}^i \delta(\mathbf{x}_\mathbf{k} - \mathbf{x}_{\mathbf{k}|\mathbf{k}-1}^{\mathbf{i}}) \text{ with } \mathbf{x}_{\mathbf{k}|\mathbf{k}-1}^{\mathbf{i}} = f_k(\mathbf{x}_{\mathbf{k}-1}^{\mathbf{i}}, \mathbf{v}_\mathbf{k}^{\mathbf{i}})$$

Filtering: Update the weight of each particle

$$p(\mathbf{x}_\mathbf{k}|\mathbf{Z}^\mathbf{k}) \approx \sum_{i=1}^{N} w_k^i \delta(\mathbf{x}_\mathbf{k} - \mathbf{x}_{\mathbf{k}|\mathbf{k}-1}^{\mathbf{i}}) \text{ with } w_k^i \propto p(\mathbf{z}_\mathbf{k}|\mathbf{x}_\mathbf{k})\, w_{k-1}^i$$

$$w_k^i \propto w_{k-1}^i \frac{p(\mathbf{z}_\mathbf{k}|\mathbf{x}_\mathbf{k}^\mathbf{i}) p(\mathbf{x}_\mathbf{k}^\mathbf{i}|\mathbf{x}_{\mathbf{k}-1}^\mathbf{i})}{q(\mathbf{x}_\mathbf{k}^\mathbf{i}|\mathbf{x}_{\mathbf{k}-1}^\mathbf{i}, \mathbf{z}_\mathbf{k})}$$

Sequential Importance Sampling

1. **Initialize** $\{\mathbf{x}_0^\mathbf{i}\}, \{w_0^i\}$
2. **For** $k = 1, 2, \ldots$
   - **Sample N particles:** $\mathbf{x}_\mathbf{k}^\mathbf{i} \sim p(\mathbf{x}_\mathbf{k}|\mathbf{x}_{\mathbf{k}-1}^\mathbf{i})$
   - **Recalculate** $\{w_k^i\}$ **with** $w_k^i \propto p(\mathbf{z}_\mathbf{k}|\mathbf{x}_\mathbf{k}) w_{k-1}^i$

→ easy to implement
→ easy to parallelize
→ typically diverging due to degeneracy

Degeneracy

As the filtering continues less and less particles are representative of the posterior state

Solution: Compute number of effective particles through the inverse variance of the weights and resample, if below a certain threshold

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} (w_k^i)^2}$$

Optimal: $w_k^i = \frac{1}{N}$

Sampling Importance Resampling

1. **Initialize** $\{\mathbf{x}_0^\mathbf{i}\}, \{w_0^i\}$
2. **For** $k = 1, 2, \ldots$
   - **Sample N particles:** $\mathbf{x}_\mathbf{k}^\mathbf{i} \sim p(\mathbf{x}_\mathbf{k}|\mathbf{x}_{\mathbf{k}-1}^\mathbf{i})$
   - **Recalculate** $\{w_k^i\}$ **with** $w_k^i \propto p(\mathbf{z}_\mathbf{k}|\mathbf{x}_\mathbf{k}) w_{k-1}^i$
   - **Resample particles, new weights:** $w_k^i = 1/N$

→ takes care of degeneracy through resampling
→ resampling cannot be parallelized

Auxiliary SIR Particle Filter

Mimic the optimal importance density $q(\mathbf{x}_\mathbf{k}^\mathbf{i}|\mathbf{x}_{\mathbf{k}-1}^\mathbf{i}, \mathbf{z}_\mathbf{k})$ through the importance density: $q(\mathbf{x}_\mathbf{k}, i|\mathbf{Z}^\mathbf{k}) \propto p(\mathbf{z}_k|\boldsymbol{\mu}_k^i) p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i) w_{k-1}^i$

→ particles are drawn from areas of high likelihood

Derivation:

$$\begin{aligned} p(\mathbf{x}_\mathbf{k}, i|\mathbf{Z}^\mathbf{k}) &\propto p(\mathbf{z}_\mathbf{k}|\mathbf{x}_\mathbf{k}) p(\mathbf{x}_\mathbf{k}, i|\mathbf{Z}^{\mathbf{k}-1}) \\ &= p(\mathbf{z}_\mathbf{k}|\mathbf{x}_\mathbf{k}) p(\mathbf{x}_\mathbf{k}|i, \mathbf{Z}^{\mathbf{k}-1}) p(i|\mathbf{Z}^{\mathbf{k}-1}) \\ &= p(\mathbf{z}_\mathbf{k}|\mathbf{x}_\mathbf{k}) p(\mathbf{x}_\mathbf{k}|\mathbf{x}_{\mathbf{k}-1}^\mathbf{i}) w_{k-1}^i \end{aligned}$$

Multiple Hypothesis Tracking

Idea: Build up a hypothesis tree that allows to track multiple hypotheses and keep the relations between them

Prediction:
$$p(\mathbf{x}_k|\mathcal{Z}^{k-1}, h_1) = \sum_{H_{k-1}} p(\mathbf{x}_k|H_{k-1}\mathcal{Z}^{k-1}, h_1)\, p(H_{k-1}|\mathcal{Z}^{k-1}, h_1)$$

Filtering:
$$p(\mathbf{x}_k|\mathcal{Z}^k) = \sum_{H_k} p_{H_k} \mathcal{N}\big(\mathbf{x}_k;\, \mathbf{x}_{H_k},\, \mathbf{P}_{H_k}\big)$$

$\mathbf{x}_{H_k}, \mathbf{P}_{H_k}$: update by KALMAN iteration!

$$p_{H_k} = \frac{p^*_{H_k}}{\sum_{H_k} p^*_{H_k}}, \qquad p^*_{H_k} = p_{H_{k-1}} \begin{cases} (1-P_D)\,\rho_F \\ P_D\, \mathcal{N}(\boldsymbol{\nu}_{H_k}, \mathbf{S}_{H_k}) \end{cases}$$

Retrodiction:
$$p(\mathbf{x}_l|H_k, \mathcal{Z}^k) = \mathcal{N}\big(\mathbf{x}_l;\, \mathbf{x}_{H_k}(l|k),\, \mathbf{P}_{H_k}(l|k)\big) \approx \mathcal{N}\big(\mathbf{x}_l;\, \mathbf{x}_{H_k}(l|l),\, \mathbf{P}_{H_k}(l|l)\big)$$

$$p(\mathbf{x}_l|H_k, \mathcal{Z}^k) \approx \sum_{H_l} p^*_{H_l}\, \mathcal{N}\big(\mathbf{x}_l;\, \mathbf{x}_{H_k}(l|l),\, \mathbf{P}_{H_k}(l|l)\big)$$

New weights: $p^*_{H_k} = p_{H_k}, \quad p^*_{H_l} = \sum p^*_{H_{l+1}}$

→ weak children weaken strong parents
→ strong children strengthen weak parents
→ if children die, the parents die

Growing Memory Disaster

With continuing tracking of multiple hypotheses the number of hypotheses explodes: m data, N hypotheses → N^{m+1} continuations

Mono-hypothesis solution:

Gating: Exclude competing data with $||\boldsymbol{\nu}^i_{k|k-1}|| > \lambda$! (KF)

+ very simple, – λ too small: loss of target measurement

Unique interpretation: look for *smallest statistical distance:* $\min_i ||\boldsymbol{\nu}^i_{k|k-1}||$ (NN)

+ one hypothesis, – hard decision, – not adaptive

Global combining: Merge all hypotheses! (PDAF)

+ all data, + adaptive, – reduced applicability

Adaptive solution:

Individual gating: Exclude irrelevant data for each track before continuing it

Pruning: Kill hypothesis of small weight before filtering

Local combining: Merge similar hypothesis after computing the pdf's

Succesive Local Combining

Similarity: $d(H_1, H_2) < \mu$ with: $d(H_1, H_2) = (\mathbf{x}_{H_1} - \mathbf{x}_{H_2})^\top (\mathbf{P}_{H_1} + \mathbf{P}_{H_2})^{-1} (\mathbf{x}_{H_1} - \mathbf{x}_{H_2})$

Merging: Moment matching: $\boxed{\sum_{H_k \in \mathcal{H}^{k*}} p_{H_k} \mathcal{N}(\mathbf{x}_k; \mathbf{x}_{H_k}, \mathbf{P}_{H_k}) \approx p_{H_k^*} \mathcal{N}(\mathbf{x}_k; \mathbf{x}_{H_k^*}, \mathbf{P}_{H_k^*})}$

Algorithm.

Iterate through hypothesis in descending order according to the hypothesis weights:

Search for similar hypothesis and merge $(H_1, H) \succ H_1^*$

PDAF Filter

Idea: Combine multiple measurements/hypothesis into a single innovation using moment matching

Prediction: $\boxed{p(\mathbf{x}_k | \mathcal{Z}^{k-1}) \approx \mathcal{N}(\mathbf{x}_k;\, \mathbf{x}_{k|k-1}, \mathbf{P}_{k|k-1}) \quad \text{(like Kalman)}}$

Filtering:

$$p(\mathbf{x}_k | \mathcal{Z}^k) \approx \sum_{j=0}^{m_k} p_k^j \mathcal{N}(\mathbf{x}_k;\, \mathbf{x}_{k|k}^j, \mathbf{P}_{k|k}^j) \approx \mathcal{N}(\mathbf{x}_k;\, \mathbf{x}_{k|k}, \mathbf{P}_{k|k})$$

$$\mathbf{x}_{k|k}^j = \begin{cases} \mathbf{x}_{k|k-1} & j=0 \\ \mathbf{x}_{k|k-1} + \mathbf{W}_k \boldsymbol{\nu}_k^j & j\neq 0 \end{cases} \qquad \mathbf{P}_{k|k}^j = \begin{cases} \mathbf{P}_{k|k-1} & j=0 \\ \mathbf{P}_{k|k-1} - \mathbf{W}_k \mathbf{S}_k \mathbf{W}_k^\top & j\neq 0 \end{cases}$$

$$\boldsymbol{\nu}_k^j = \underbrace{\mathbf{z}_k^j - \mathbf{H}\mathbf{x}_k}_{\text{innovation}}, \qquad \mathbf{W}_k = \underbrace{\mathbf{P}_{k|k-1}\mathbf{H}^\top \mathbf{S}_k^{-1}}_{\text{gain matrix}}, \qquad \mathbf{S}_k = \underbrace{\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^\top + \mathbf{R}_k}_{\text{innovation covariance}}$$

$$p_k^j = \underbrace{\frac{p_k^{j*}}{\sum_j p_k^{j*}}}_{\text{Gewichte}}, \qquad p_k^{j*} = \begin{cases} (1 - P_D)\,\rho_F & j=0 \\ \frac{P_D}{\sqrt{|2\pi \mathbf{S}_{H_k}|}} e^{-\frac{1}{2}\boldsymbol{\nu}_{H_k}^\top \mathbf{S}_{H_k}^{-1} \boldsymbol{\nu}_{H_k}} & j\neq 0 \end{cases}$$

Characteristics:

- filtering: processing of *combined innovation*
- *all data* Z_k in the gate are considered
- p_i data dependent! Update *not linear*
- missing measurement: $\mathbf{P}_{k|k-1}$ with weight p_0
- *"usual"* Kalman covarianve according to $(1 - p_0)$
- Spread *positively semidefinite:* larger covariance
- therefore: *data driven adaptivity*
- *non linear estimator:* data dependent error
- Performance prediction *only via simulations*

Interacting Multiple Models

Idea: The measurement model or object dynamics might change over the course of tracking. To model this, we employ a hidden Markov model modeling the transition between those models.

IMM model:

$$p(x_k, j_k|x_{k-1}, j_{k-1}) = p(j_k|j_{k-1})\ \mathcal{N}\big(\mathrm{x}_k;\ \mathrm{F}^{j_k}_{k|k-1}\mathrm{x}_{k-1},\ \mathrm{D}^{j_k}_{k|k-1}\big)$$

Prediction:

$$p(\mathrm{x}_k|\mathcal{Z}^{k-1}) = \sum_{j_k=1}^{r}\sum_{j_{k-1}=1}^{r} p(j_k|j_{k-1})\ p(j_{k-1}|\mathcal{Z}^{k-1})\ \mathcal{N}\big(\mathrm{x}_k;\ \mathrm{x}^{j_k j_{k-1}}_{k|k-1},\ \mathrm{P}^{j_k j_{k-1}}_{k|k-1}\big)$$

$$\mathrm{x}^{j_k j_{k-1}}_{k|k-1} = \mathrm{F}^{j_k}_{k|k+1}\mathrm{x}^{j_{k-1}}_{k-1|k-1}, \qquad \mathrm{P}^{j_k j_{k-1}}_{k|k-1} = \mathrm{F}^{j_k}_{k|k+1}\mathrm{x}^{j_{k-1}}_{k-1|k-1}\mathrm{F}^{j_k\top}_{k|k+1} + \mathrm{D}^{j_k}_{k|k+1}$$

Mixing step:

$$p(\mathrm{x}_k|\mathcal{Z}^{k-1}) \approx \sum_{j_k=1}^{r} p(j_k|\mathcal{Z}^{k-1})\ \mathcal{N}\big(\mathrm{x}_k;\ \mathrm{x}^{j_k}_{k|k-1},\ \mathrm{P}^{j_k}_{k|k-1}\big)$$

$$p(j_k|\mathcal{Z}^{k-1}) = \sum_{j_{k-1}=1}^{r} p(j_k|j_{k-1})\ p(j_{k-1}|\mathcal{Z}^{k-1})$$

$$\mathrm{x}^{j_k}_{k|k-1} = \frac{1}{p(j_k|\mathcal{Z}^{k-1})}\sum_{j_{k-1}=1}^{r} p(j_k|j_{k-1})\ p(j_{k-1}|\mathcal{Z}^{k-1})\mathrm{x}^{j_k j_{k-1}}_{k|k-1}, \quad \mathrm{P}^{j_k}_{k|k-1} = \frac{1}{p(j_k|\mathcal{Z}^{k-1})}\sum_{j_{k-1}=1}^{r} p(j_k|j_{k-1})\ p(j_{k-1}|\mathcal{Z}^{k-1})\big(\mathrm{P}^{j_k j_{k-1}}_{k|k-1} + (\mathrm{x}^{j_k j_{k-1}}_{k|k-1} - \mathrm{x}^{j_k}_{k|k-1})(\cdots)^\top\big)$$

→ apply moment matching to approximate r gaussians that build a mixture of gaussians

→ each gaussian represents the prediction given the specific hidden model

Filtering:

$$p(\mathrm{x}_k|\mathcal{Z}^{k}) = \sum_{j_k=1}^{r} p(j_k|\mathcal{Z}^{k})\ \mathcal{N}\big(\mathrm{x}_k;\ \mathrm{x}^{j_k}_{k|k},\ \mathrm{P}^{j_k}_{k|k}\big)$$

with: $$p(j_k|\mathcal{Z}^k) = \frac{\mathcal{N}\big(\mathrm{z}_k;\ \mathrm{H}_k\mathrm{x}^{j_k}_{k|k-1},\ \mathrm{H}_k\mathrm{P}^{j_k}_{k|k-1}\mathrm{H}_k + \mathrm{R}_k\big)}{\sum_{j'_k=1}^{r} p(j'_k|\mathcal{Z}^{k-1})\ \mathcal{N}\big(\mathrm{z}_k;\ \mathrm{H}_k\mathrm{x}^{j'_k}_{k|k-1},\ \mathrm{H}_k\mathrm{P}^{j'_k}_{k|k-1}\mathrm{H}_k + \mathrm{R}_k\big)}$$ (mixture coefficients)

$$\mathrm{x}^{j_k}_{k|k} = \mathrm{x}^{j_k}_{k|k-1} + \mathrm{W}^{j_k}_{k|k}(\mathrm{z}_k - \mathrm{H}_k\mathrm{x}^{j_k}_{k|k-1}), \quad \mathrm{P}^{j_k}_{k|k} = \mathrm{P}^{j_k}_{k|k-1} - \mathrm{W}^{j_k}_{k|k-1}\mathrm{S}^{j_k}_{k|k}\mathrm{W}^{j_k}_{k|k-1}, \quad \mathrm{W}^{j_k}_{k|k} = \mathrm{P}^{j_k}_{k|k-1}\mathrm{H}_k^\top\mathrm{S}^{j_k\,-1}_{k|k} \quad \mathrm{S}^{j_k}_{k|k} = \mathrm{H}_k\mathrm{P}^{j_k}_{k|k-1}\mathrm{H}_k + \mathrm{R}_k$$

Retrodiction:

$$p(\mathrm{x}_l|\mathcal{Z}^k) \approx \sum_{i_l} \mu^k_{i_l}\ \mathcal{N}\big(\mathrm{x}_l;\ \mathrm{x}^k_{i_l},\ \mathrm{P}^k_{i_l}\big)$$

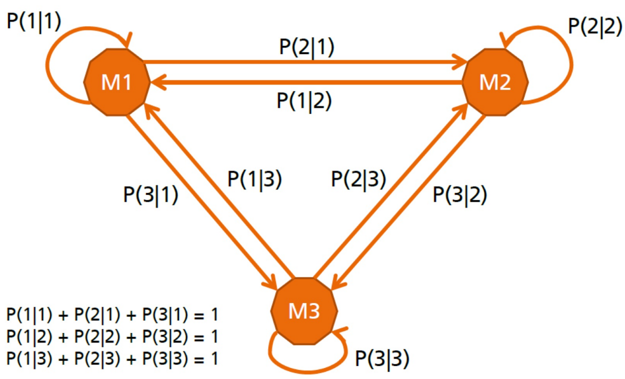
$$p(\mathrm{x}_l|\mathcal{Z}^k) = \sum_{i_{l+1}, i_l}\int d\mathrm{x}_{l+1}\ \underbrace{p(\mathrm{x}_l, i_l|\mathrm{x}_{l+1}, i_{l+1}, \mathcal{Z}^k)}_{\text{calculated!}}\ \underbrace{p(\mathrm{x}_{l+1}, i_{l+1}|\mathcal{Z}^k)}_{\text{retrodiction in } t_{l+1}}$$

insert, product formula!

$$= \sum_{i_{l+1}, i_l} \mu^k_{i_{l+1}i_l}\ \mathcal{N}\big(\mathrm{x}_l;\ \mathrm{x}^k_{i_{l+1}i_l},\ \mathrm{P}^k_{i_{l+1}i_l}\big)$$

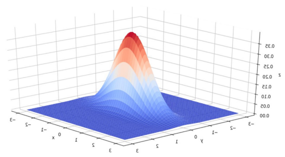
exponential growth of dynamics histories $i_{l+1}i_l\ldots$!

$$= \sum_{i_l}\underbrace{\sum_{i_{l+1}} \mu^k_{i_{l+1}i_l}\ \mathcal{N}\big(\mathrm{x}_l;\ \mathrm{x}^k_{i_{l+1}i_l},\ \mathrm{P}^k_{i_{l+1}i_l}\big)}_{\text{approximation: moment matching!}}$$



Tensor Based Tracking

Idea: Tensor based tracking discretizes the state space and can represent arbitrary PDFs. Thus, we make no gaussian assumption and can directly employ non-linear evolution and measurement models



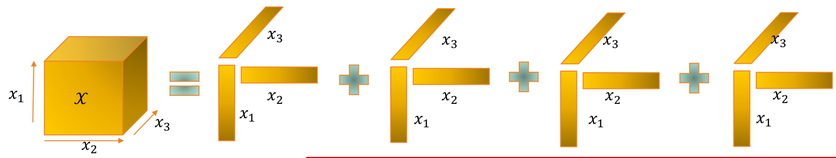
CANDECOMP/PARAFAC Format

Motivation: Tensors quickly become too large to save. Thus, we compute a low rank approximation to work more efficiently with them.

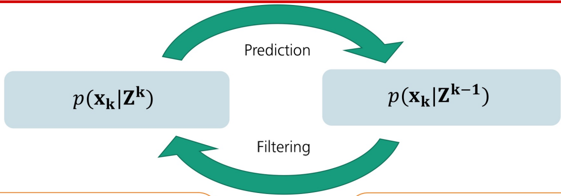
Idea: Represent a tensor through outer products of vectors that build the "basis" for the space that the tensor spans. The rank of this approximation is variable.

Approximation:

$$\mathcal{Y} = \sum_{i=0}^{R} \mathbf{y}_{1,i} \circ \ldots \circ \mathbf{y}_{D,i} = [\mathbf{U}_1, \ldots, \mathbf{U}_D]$$



Tracking Formalism:



Bayes Theorem

$$p(\mathbf{x_k}|\mathbf{Z^k}) = \frac{p(\mathbf{x_k}\,|\mathbf{Z^{k-1}})\, p(\mathbf{z_k}|\mathbf{x_k})}{\int p(\mathbf{x_k}|\mathbf{Z^{k-1}}) p(\mathbf{z_k}|\mathrm{x_k}) d\mathbf{x_k}}$$

Fokker-Planck Equation (FPE)

$$\frac{\partial p}{\partial t} = -\sum_{i=1}^{D} \frac{\partial([\boldsymbol{f}]_i p)}{\partial x_i} + \frac{1}{2}\sum_{i,j=1}^{D} \frac{\partial^2([\mathbf{GQG^T}]_{i,j} p)}{\partial x_i \partial x_j}$$

Prediction:

Fokker-Planck Equation (FPE)

$$\partial_t p = -\sum_{i=1}^{D} \partial_{x_i}([\boldsymbol{f}]_i p) + \frac{1}{2}\sum_{i,j=1}^{D} \partial_{x_i}\partial_{x_j}([\boldsymbol{B}]_{i,j} p)$$

$$[\boldsymbol{f}]_i(x) = \sum_{k=1}^{K_i} \prod_{d=1}^{D} f_{d,k}^{i}(x_d)$$

$$[\boldsymbol{B}]_{i,j}(x) = \sum_{k=1}^{K_{i,j}} \prod_{d=1}^{D} B_{d,k}^{i,j}(x_d)$$

- Prediction by solving Fokker-Planck equation
  - Use Fokker-Planck operator [1]

$$\partial_t vec(p) = (\boldsymbol{\mathcal{L}}_{drift} + \boldsymbol{\mathcal{L}}_{diffusion}) vec(\mathrm{p})$$

  - With matrix exponential

$$\mathrm{vec}(p_{k|k-1}) = \exp\left(\Delta_t(\boldsymbol{\mathcal{L}}_{dritft} + \boldsymbol{\mathcal{L}}_{diff})\right) vec(p_{k-1|k-1})$$

Filtering:

Filtering by Bayes Formula

Discretize likelihood

Apply ALS

Compute hadamard product of the likelihood an the prior

Normalize

$$p(\mathbf{x_k}|\mathcal{Z}^{\mathbf{k}}) = \frac{\mathrm{L}(\mathbf{z_k}|\mathbf{x_k})\, p(\mathbf{x_k}|\mathcal{Z}^{\mathbf{k-1}})}{\int \mathrm{L}(\mathbf{z_k}|\mathbf{x_k})\, p(\mathbf{x_k}|\mathcal{Z}^{\mathbf{k-1}})\, d\mathbf{x_k}}$$

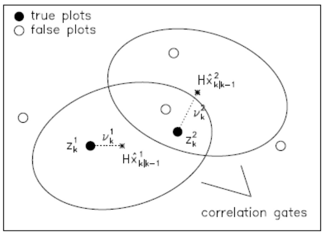
$$\Rightarrow [\mathbf{U_1}, \ldots \mathbf{U_D}] = \frac{[\mathbf{L_1}, \ldots, \mathbf{L_D}] * [\mathbf{U_1}, \ldots, \mathbf{U_D}]}{\int [\mathbf{L_1}, \ldots, \mathbf{L_D}] * [\mathbf{U_1}, \ldots, \mathbf{U_D}]\, d\mathbf{x_k}}$$

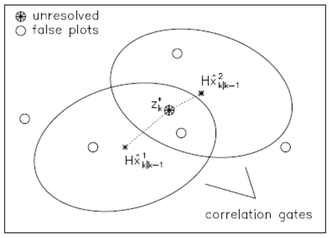
3. Measurement Modeling

Expected Measurements

The innovation covariance draws an expectation gate which can be used to sort out unprobable measurements using the Mahalanobis distance.

innovation:	$\boldsymbol{\nu}_{k\|k-1} = \mathbf{z}_k - \mathbf{H}_k \mathbf{x}_{k\|k-1}$,
innovation covariance:	$\mathbf{S}_{k\|k-1} = \mathbf{H}_k \mathbf{P}_{k\|k-1} \mathbf{H}_k^\top + \mathbf{R}_k$
expectation gate:	$\boldsymbol{\nu}_{k\|k-1}^\top \mathbf{S}_{k\|k-1}^{-1} \boldsymbol{\nu}_{k\|k-1} \leq \lambda(P_c)$
MAHALANOBIS	ellipsoid containing $\mathbf{z}_k$ with certain probability P_c
	Choose $\lambda(P_c)$ ("gating parameter") properly!
	Can be looked up in a χ^2-table





Derivation:

$$p(\mathbf{z}_k | \mathcal{Z}^{k-1}) = \int d\mathbf{x}_k \, p(\mathbf{z}_k, \mathbf{x}_k | \mathcal{Z}^{k-1}) = \int d\mathbf{x}_k \, p(\mathbf{z}_k | \mathbf{x}_k) \, p(\mathbf{x}_k | \mathcal{Z}^{k-1})$$

$$= \int d\mathbf{x}_k \, \underbrace{\mathcal{N}(\mathbf{z}_k;\, \mathbf{H}_k \mathbf{x}_k,\, \mathbf{R}_k)}_{\text{likelihood: sensor model}} \, \underbrace{\mathcal{N}(\mathbf{x}_k;\, \mathbf{x}_{k|k-1},\, \mathbf{P}_{k|k-1})}_{\text{prediction at time } t_k}$$

$$= \mathcal{N}(\mathbf{z}_k;\, \mathbf{H}_k \mathbf{x}_{k|k-1},\, \mathbf{S}_{k|k-1}) \qquad \text{(product formula)}$$

Detection Process Modeling

Detector: receive signal and decide on object existence

Processor: process the received measurement and produce measurement

Detector errors:

'D': detector detects an object	error of 1. kind: $P_{\mathrm{I}} = P(\neg\text{'}D\text{'}\|D)$
D: object actually existent	error of 2. kind: $P_{\mathrm{II}} = P(\text{'}D\text{'}\|\neg D)$

Detector characteristics: detection probability $P_D = 1 - P_{\mathrm{I}}$
false alarm probability $P_F = P_{\mathrm{II}}$

Detector design: Maximize P_D for a given P_F

Swerling I Model: $P_D = P_D(P_F, \text{SNR}) = P_F^{1/(1+\text{SNR})}$

Modeling of False Measurements

Number n of false measurements: poisson distributed

$$p_F(n) = \frac{\bar{n}^n}{n!} e^{-\bar{n}} \qquad \text{expectation: } \mathbb{E}[n] = \bar{n}, \qquad \text{variance: } \mathbb{V}[n] = \bar{n}$$

Mean in FoV: $\bar{n} = \rho_F |\text{FoV}|$.

Distribution in the Field of View: uniformly distributed

$$p(\mathbf{z}_1^f, \ldots, \mathbf{z}_n^f | \text{FoV}) = \prod_{i=1}^{n} p(\mathbf{z}_i^f | \text{FoV}) \qquad p(\mathbf{z}_i^f | \text{FoV}) = |\text{FoV}|^{-1}$$

Modeling Ambiguous Sensor Data

Idea: If we have a non-ideal sensor ($P_D < 1$, $P_F > 0$), we have different possible data associations/interpretations

Let: $n_k + 1$ possible interpretations of the sensor data $Z_k = \{\mathbf{z}_k^j\}_{j=1}^{n_k}$!

Data interpretations: E_0: the object was not detected; n_k false data in the Field of View (FoV)

E_j, $j = 1, \dots, n_k$: Object detected; $\mathbf{z}_k^j$ is object measurement; $n_k - 1$ false plots

Measurement Model: $$\boxed{p(Z_k, n_k|\mathbf{x}_k) \propto (1 - P_D)\rho_F + P_D \sum_{j=1}^{n_k} \mathcal{N}(\mathbf{z}_k^j;\, \mathbf{H}\mathbf{x}_k,\, \mathbf{R})}$$

Derivation:

$$p(Z_k, n_k|\mathbf{x}_k) = p(Z_k, n_k, \neg D|\mathbf{x}_k) + p(Z_k, n_k, D|\mathbf{x}_k) \qquad D = \text{“object was detected”}$$
$$= p(Z_k, n_k|\neg D, \mathbf{x}_k)\, P(\neg D|\mathbf{x}_k) + p(Z_k, n_k|D, \mathbf{x}_k)\, p(D|\mathbf{x}_k)$$
$$= p(Z_k|n_k, \neg D, \mathbf{x}_k)\, p(n_k|\neg D, \mathbf{x}_k)\, (1 - P_D) + P_D \sum_{j=1}^{n_k} p(Z_k, n_k, j|D, \mathbf{x}_k)$$
$$= |\text{FoV}|^{-n_k}\, p_F(n_k)\, (1 - P_D) + P_D \sum_{j=1}^{n_k} p(Z_k|n_k, j, D, \mathbf{x}_k)\, p(j|n_k, D)\, p(n_k|D)$$
$$= \frac{e^{-\rho_F|\text{FoV}|}}{n_k!}\, \rho_F^{n_k-1}\Big((1 - P_D)\rho_F + P_D \sum_{j=1}^{n_k} \mathcal{N}(\mathbf{z}_k^j;\, \mathbf{H}\mathbf{x}_k,\, \mathbf{R})\Big)$$

Example: Well-separate targets

filtering (at time t_{k-1}): $\quad p(\mathbf{x}_{k-1}|\mathcal{Z}^{k-1}) = \sum_{H_{k-1}} p_{H_{k-1}}\, \mathcal{N}(\mathbf{x}_{k-1};\, \mathbf{x}_{H_{k-1}},\, \mathbf{P}_{H_{k-1}})$

prediction (for time t_k):

$$p(\mathbf{x}_k|\mathcal{Z}^{k-1}) = \int d\mathbf{x}_{k-1}\, p(\mathbf{x}_k|\mathbf{x}_{k-1})\, p(\mathbf{x}_{k-1}|\mathcal{Z}^{k-1}) \qquad \text{(MARKOV model)}$$
$$= \sum_{H_{k-1}} p_{H_{k-1}}\, \mathcal{N}(\mathbf{x}_k;\, \mathbf{F}\mathbf{x}_{H_{k-1}},\, \mathbf{F}\mathbf{P}_{H_{k-1}}\mathbf{F}^\top + \mathbf{D}) \qquad \text{(IMM also possible)}$$

measurement likelihood:

$$p(Z_k, m_k|\mathbf{x}_k) = \sum_{j=0}^{m_k} p(Z_k|E_k^j, \mathbf{x}_k, m_k)\, P(E_k^j|x_k, m_k) \qquad (E_k^j\text{: interpretations})$$
$$\propto (1 - P_D)\, \rho_F + P_D \sum_{j=1}^{m_k} \mathcal{N}(\mathbf{z}_k^j;\, \mathbf{H}\mathbf{x}_k,\, \mathbf{R}) \qquad (\mathbf{H}, \mathbf{R}, P_D, \rho_F)$$

filtering (at time t_k):

$$p(\mathbf{x}_k|\mathcal{Z}^k) \propto p(Z_k, m_k|\mathbf{x}_k)\, p(\mathbf{x}_k|\mathcal{Z}^{k-1}) \qquad \text{(BAYES' rule)}$$
$$= \sum_{H_k} p_{H_k}\, \mathcal{N}(\mathbf{x}_k;\, \mathbf{x}_{H_k},\, \mathbf{P}_{H_k}) \qquad \text{(Exploit product formula)}$$

Simple Radar Beam and Detection Model

Beam positioning error: $\Delta b_k^2 = |\mathbf{x}_k - \mathbf{b}_k|^2 / B^2$

$\mathbf{x}_k$: target direction, $\mathbf{b}_k$: current beam position, B: radar beam width

Signal-to-Noise ratio: $s(\mathbf{x}_k; \mathbf{b}_k) = s_0\, e^{-\log 2 \Delta b_k^2}$

False alarm probability: P_F — SWERLING model

Detection probability: $P_D(\mathbf{x}_k; \mathbf{b}_k) = P_F^{\frac{1}{1+s(\mathbf{x}_k; \mathbf{b}_k)}}$

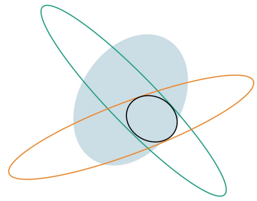
Out-of-Sequence Measurements

Measurements might not reach the fusion system in temporal sequential order due to:

- communication delays
- multi-path delivery
- local data caching

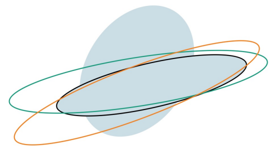
Fusion of Measurements

Fuse measurement to improve certainty through complementary covariances



$$\mathbf{R} = \left(\sum_{i=1}^{n} \mathbf{R}_i^{-1}\right)^{-1}$$

$$\mathbf{z} = \mathbf{R}\sum_{i}^{n} \mathbf{R}_i^{-1}\mathbf{z}_i$$



Derivation:

$$p(\mathbf{z}_1, \mathbf{z}_2|\boldsymbol{x}) = p(\mathbf{z}_1|\boldsymbol{x})\, p(\mathbf{z}_2|\boldsymbol{x})$$

$$= \mathcal{N}(\mathbf{z}_1; \mathbf{Hx}, \mathbf{R}_1)\, \mathcal{N}(\mathbf{z}_2; \mathbf{Hx}, \mathbf{R}_2)$$

$$= \mathcal{N}(\mathbf{Hx}; \mathbf{z}_1, \boldsymbol{R}_1)\, \mathcal{N}(\mathbf{z}_2; \mathbf{Hx}, \boldsymbol{R}_2)$$

$$\propto \mathcal{N}\left(\mathbf{Hx}; \underbrace{\mathbf{R}(\mathbf{R}_1^{-1}\mathbf{z}_1 + \mathbf{R}_2^{-1}\mathbf{z}_2)}_{\mathbf{z}}, \underbrace{(\mathbf{R}_1^{-1} + \mathbf{R}_2^{-1})^{-1}}_{\mathbf{R}}\right)$$

GMTI Detection Model

Idea: The detection probability should depend on the target kinematics and target/sensor geometry since these might induce Doppler blindness. The signal-to-(noise & interference) ratio builds the basis for this:

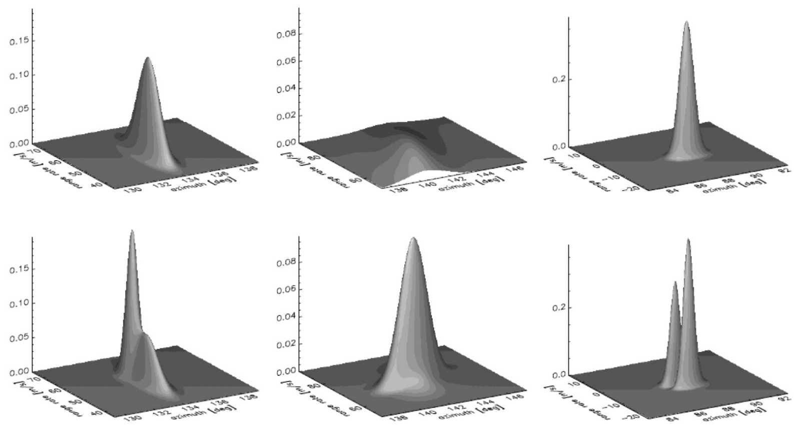
$$\boxed{\text{snir} = \text{snir}_0 \underbrace{\left(\frac{\bar{\sigma}_k}{\sigma_0}\right)}_{\text{rcs}} \underbrace{\left(\frac{r_k}{r_0}\right)^{-4}}_{\text{propagation}} \underbrace{D(\varphi_k)}_{\text{directivity}} \underbrace{\left[1 - \mathrm{e}^{-\log 2\left(\frac{n_c(r_k,\varphi_k,\dot{r}_k)}{v_m}\right)^2}\right]}_{\text{clutter notch } (<\frac{1}{2} \text{ for } |n_c|<v_m)}}$$

Detection probability:

$$\boxed{\begin{aligned} P_D(r_k, \varphi_k, \dot{r}_k) = P_{\text{FA}}^{\frac{1}{1+\text{snir}}} &\approx P_d\left(1 - \mathrm{e}^{-\log 2\left(\frac{n_c(r_k,\varphi_k,\dot{r}_k)}{v_m}\right)^2}\right) \\ &\approx P_d\left(1 - 2\pi \text{mdv}\, \mathcal{N}(0;\ h_n(\mathbf{x}_k),\ \text{mdv}^2)\right) \end{aligned}}$$

Measurement variance:

$$\boxed{\sigma_{r,\varphi,\dot{r}}(r_k, \varphi_k, \dot{r}_k) = \Sigma_{r,\varphi,\dot{r}} / \sqrt{\text{snir}(r_k, \varphi_k, \dot{r}_k)}}$$



Missing detection occurred near the clutter notch

Several missing detections in the clutter notch

Detection occurred near the clutter notch

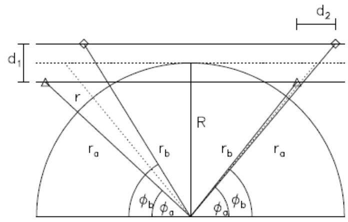
Measurement Resolution

Idea: Radars can only resolve targets up to a given angular and range resolution. Close targets will be measured with a single measurement which needs to be modeled

Resolution: α_r = 200 m, α_φ = 2°, $\alpha_{\dot r}$ = 2 m/s

very low resolution for: $\Delta r < \alpha_r$, $\Delta\varphi < \alpha_\varphi$, $\Delta\dot r < \alpha_{\dot r}$

no resolution phenomena: $\Delta r \gg \alpha_r$, $\Delta\varphi \gg \alpha_\varphi$, $\Delta\dot r \gg \alpha_{\dot r}$



Irresolved measurement: $\mathbf{H}_g \mathbf{x}_k = \frac{1}{2}\mathbf{H}(\mathbf{x}_k^1 + \mathbf{x}_k^2)$ "center of gravity"

Measurement likelihood: $\ell(Z_k, n_k | \mathbf{x}_k) = \sum_{E_k} \ell(Z_k, n_k, E_k | \mathbf{x}_k)$

Irresolved, detected as group:

E_k^{ii}: Objects irresolved, detected as a group, $\mathbf{z}_k^i \in Z_k$ being the plot:

$$\ell(Z_k, n_k, E_k^{ii} | \mathbf{x}_k) = \text{const.} \times P_u(\mathbf{x}_k)\, \mathcal{N}(\mathbf{z}_k^i; \mathbf{H}_k^g \mathbf{x}_k, \mathbf{R}_k^g)$$

$$= \text{const.}' \times \mathcal{N}\left(\begin{pmatrix} \mathbf{z}_k^i \\ \mathbf{o} \end{pmatrix}; \begin{pmatrix} \mathbf{H}_k^g \\ \mathbf{H}_u \end{pmatrix} \mathbf{x}_k, \begin{pmatrix} \mathbf{R}_k^g & \mathbf{O} \\ \mathbf{O} & \mathbf{R}_u \end{pmatrix}\right)$$

Assuming E_k^{ii}, we process a (real) measurement $\mathbf{z}_k^i$ of the center $\frac{1}{2}\mathbf{H}(\mathbf{x}_k^1 + \mathbf{x}_k^2)$ and a (fictitious) measurement "zero" of the distance $\mathbf{H}(\mathbf{x}_k^1 - \mathbf{x}_k^2)$ between the targets. $\mathbf{R}_u$ defines the resolution capability.

Neither resolved nor detected:

E_k^{00}: Objects neither resolved, nor detected; all plots are false

$$\ell(Z_k, n_k, E_k^{00} | \mathbf{x}_k) = P_u(\mathbf{x}_k)\,(1 - P_D^u)\, \frac{p_F(n_k)}{|\text{FoV}|^{n_k}}$$

$$= \text{const.} \times \mathcal{N}\big(\mathbf{O};\, \mathbf{H}(\mathbf{x}_k^1 - \mathbf{x}_k^2),\, \mathbf{R}_u\big)$$

Assuming E_k^{00}, a fictitious zero-distance measurement is processed.

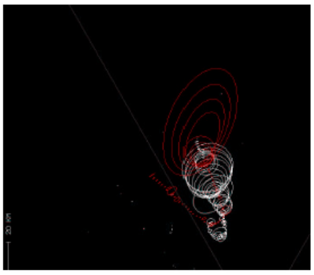
Resolved and individually detected:

E_k^{ij}: Objects resolved and individually detected, $\mathbf{z}_k^i, \mathbf{z}_k^j$ being the plots

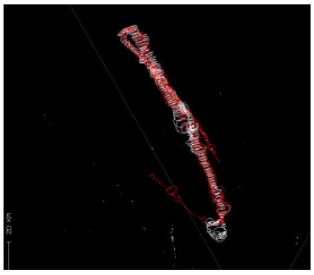
$$\ell(Z_k, n_k | E_k^{ij}, \mathbf{x}_k) = \text{const.} \times [1 - P_u(\mathbf{x}_k)]\, \mathcal{N}\left(\begin{pmatrix} \mathbf{z}_k^i \\ \mathbf{z}_k^j \end{pmatrix}; \begin{pmatrix} \mathbf{H}_k \\ \mathbf{H}_k \end{pmatrix} \mathbf{x}_k, \begin{pmatrix} \mathbf{R}_k & \mathbf{O} \\ \mathbf{O} & \mathbf{R}_k \end{pmatrix}\right)$$

Mixtures with *negative* coefficients occur! Interpretation: Resolved targets keep a minimum distance, otherwise they were irresolvable.

radar raw data



no resolution model



with resolution model

4. Advanced Tracking

Sequential Track Extraction

To initiate tracks, we first have to detect objects appropriately in the data which is not trivial due to background clutter.

Hypothesis:

h_1: Z^k contains a real object measurement
h_0: there is no object measurement in Z^k

Statistical Decision Errors:

$P_1 = P(\text{accept } h_1 | h_1)$: detect an existing object (corresponds to P_D)
$P_0 = P(\text{accept } h_1 | h_0)$: detect a non-existing object (corresponds to P_F)

Goal: For given P_0 and P_1 decide on track initiation as fast as possible

Sequential Likelihood Ratio Test

Iteratively compute the ratio between h_1 and h_0 being true given our data.

Ratio:

$$\frac{p(h_1|\mathcal{Z}^k)}{p(h_0|\mathcal{Z}^k)} = \frac{p(\mathcal{Z}^k|h_1)}{p(\mathcal{Z}^k|h_0)}\frac{p(h_1)}{p(h_0)} \quad \text{a priori: } p(h_1) = p(h_0)$$

Decision:

If	$\text{LR}(k) < A$,	accept hypothesis h_0 (i.e. no target is existing)!
If	$\text{LR}(k) > B$,	accept hypothesis h_1 (i.e. target exists in FoV)!
If	$A < \text{LR}(k) < B$,	wait for new data Z_{k+1}, repeat with $\text{LR}(k+1)$!

Thresholds:

$$A \approx \frac{1-P_1}{1-P_0} \quad \text{and} \quad B \approx \frac{P_1}{P_0}$$

Iterative calculation:

initiation: $k = 0, \quad \mathrm{j}_0 = 0, \quad \lambda_{\mathrm{j}_0} = 1$

recursion: $$\text{LR}(k+1) = \sum_{\mathrm{j}_{k+1}} \lambda_{\mathrm{j}_{k+1}} = \sum_{\mathrm{j}_{k+1}=0}^{m_{k+1}} \sum_{\mathrm{j}_k} \lambda_{\mathrm{j}_{k+1}\mathrm{j}_k} \lambda_{\mathrm{j}_k}$$

with: $$\lambda_{\mathrm{j}_{k+1}\mathrm{j}_k} = \begin{cases} 1 - P_D & \text{for } \mathrm{j}_{k+1} = 0 \\ \frac{P_D}{\rho_F}\mathcal{N}(\boldsymbol{\nu}_{\mathrm{j}_{k+1}\mathrm{j}_k}, \mathbf{S}_{\mathrm{j}_{k+1}\mathrm{j}_k}) & \text{for } \mathrm{j}_{k+1} \neq 0 \end{cases}$$

innovation: $\boldsymbol{\nu}_{\mathrm{j}_{k+1}\mathrm{j}_k} = \mathbf{z}_{\mathrm{j}_{k+1}} - \mathbf{H}_{\mathrm{j}_{k+1}}\mathbf{x}_{\mathrm{j}_{k+1|k}}$

innov. cov.: $\mathbf{S}_{\mathrm{j}_{k+1}\mathrm{j}_k} = \mathbf{H}_{\mathrm{j}_{k+1}}\mathbf{P}_{\mathrm{j}_{k+1|k}}\mathbf{H}_{\mathrm{j}_{k+1}}^\top + \mathbf{R}_{\mathrm{j}_{k+1}}$

state update: $\mathbf{x}_{\mathrm{j}_{k+1|k}} = \mathbf{F}_{\mathrm{j}_{k+1}}\mathbf{x}_{\mathrm{j}_k} \qquad \mathbf{x}_{\mathrm{j}_k} = \mathbf{x}_{\mathrm{j}_{k|k-1}} + \mathbf{W}_{\mathrm{j}_k\mathrm{j}_{k-1}}\boldsymbol{\nu}_{\mathrm{j}_k\mathrm{j}_{k-1}}$

covariances: $\mathbf{P}_{\mathrm{j}_{k+1|k}} = \mathbf{F}_{\mathrm{j}_{k+1}}\mathbf{P}_{\mathrm{j}_k}\mathbf{F}_{\mathrm{j}_{k+1}}^\top + \mathbf{D}_{\mathrm{j}_{k+1}} \qquad \mathbf{P}_{\mathrm{j}_k} = \mathbf{P}_{\mathrm{j}_{k|k-1}} - \mathbf{W}_{\mathrm{j}_k\mathrm{j}_{k-1}}\mathbf{S}_{\mathrm{j}_k\mathrm{j}_{k-1}}\mathbf{W}_{\mathrm{j}_k\mathrm{j}_{k-1}}^\top$

Track Initiation:

Normalize coefficients λ_{j_k}: $p_{\mathrm{j}_k} = \frac{\lambda_{\mathrm{j}_k}}{\sum_{\mathrm{j}_k}\lambda_{\mathrm{j}_k}}$!

$$(\lambda_{\mathrm{j}_k}, \mathbf{x}_{\mathrm{j}_k}, \mathbf{P}_{\mathrm{j}_k}) \rightarrow p(\mathbf{x}_k|\mathcal{Z}^k) = \sum_{\mathrm{j}_k} p_{\mathrm{j}_k}\mathcal{N}(\mathbf{x}_k; \mathbf{x}_{\mathrm{j}_k}, \mathbf{P}_{\mathrm{j}_k})$$

→ use LR coefficients to initiate track

Track Maintaining:

track confirmation: $\text{LR}(k) > \frac{P_1}{P_0}$: reset $\text{LR}(0) = 1$!
track deletion: $\text{LR}(k) < \frac{1-P_1}{1-P_0}$; ev. track re-initiation

→ after initiation monitor track using the LR test

Phased Array Radar

Instead of scanning the complete FoV, selectively position the radar beam to the expected position of the object. This requires precise control of the radar through the tracking system. If the object is lost, it can be locally searched through a range of dwells.

Bayesian Beam Positioning:

1. first dwell $\to$ point radar beam to predicted target position $\to \mathbf{b}_k^1$
2. successful detection $\to$ filtering/prediction/revisit time $\to$ GOTO 1!
3. no success $\to$ Bayesian processing of this 'negative' output ($\neg D_1$):

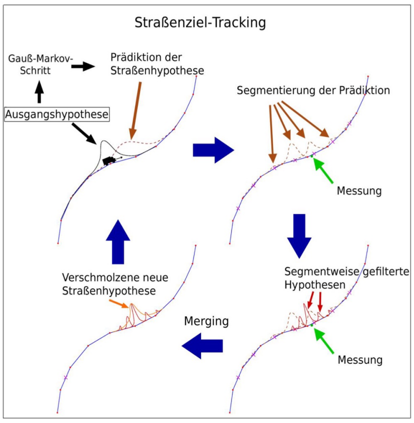
$$\boxed{p(\mathbf{x}_k|\neg D_1, \mathcal{Z}^{k-1}) \propto \left[1 - P_D(\mathbf{x}_k; \mathbf{b}_k^1)\right] p(\mathbf{x}_k|\mathcal{Z}^{k-1})}$$

4. next dwell $\to$ point beam to the maximum of this conditional pdf!
5. successful detection $\to$ filtering/prediction/revisit time $\to$ GOTO 1!
6. no success $\to$ calculate: $p(u_k, v_k|\neg D_1, \neg D_2, \mathcal{Z}^{k-1}) \to$ GOTO 4!

Road Assisted Tracking

Idea: Perform the prediction step in the road network:

$$p(\mathbf{x}_{k-1}^r|\mathcal{Z}^{k-1}) \xrightarrow{\text{Dyn.}} p(\mathbf{x}_k^r|\mathcal{Z}^{k-1})$$



Then transform this to sensor coordinates by marginalizing over the road network:

$$\underbrace{p(\mathbf{x}_k^r|\mathcal{Z}^{k-1})}_{\text{road coordinates}} \xrightarrow[\text{mapping errors } \mathbf{R}_i^j]{\text{road network}} \underbrace{p(\mathbf{x}_k^s|\mathcal{Z}^{k-1})}_{\text{sensor coordinates}}$$

And finally filter in the sensor coordinate frame:

$$p(\mathbf{x}_k^s|\mathcal{Z}^{k-1}) \xrightarrow[Z_k]{\text{Sen.}} p(\mathbf{x}_k^s|\mathcal{Z}^{k})$$

Road Model:

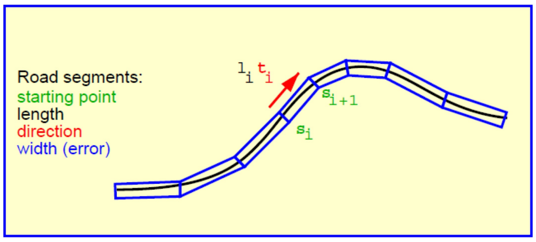
A road is defined through a continuous curve. To integrate this into our filtering framework, we can subdivide roads into linear segments, which can be represented through a start vector and a tangential vector:

$$\mathcal{R}: l \in [l_1, l_{n_r}) \mapsto \mathcal{R}(l) = \sum_{m=1}^{n_r} \left[\mathbf{s}_m + (l - l_m)\mathbf{t}_m\right] \chi_m(l)$$

arc length l, node vector $\mathbf{s}_m = \mathcal{R}(l_m)$, tangential vector $\mathbf{t}_m$, # of nodes n_r

accuracy of $\mathbf{s}_m$: covariance matrix $\mathbf{R}_m$, $\chi_m(l) = \begin{cases} 1 & \text{for } l \in [l_m, l_{m+1}) \\ 0 & \text{else} \end{cases}$

$$t_m = \frac{s_{m+1} - s_m}{\|s_{m+1} - s_m\|}$$



Discretization errors: $\|\mathbf{s}_m - \mathbf{s}_{m-1}\| \leq l_m - l_{m-1} =: \lambda_m$

$\to$ arc length of object is longer than segment length

Transformation operator:

$$\mathcal{T}^{m}_{g\leftarrow r}\left[p(\mathrm{x}^r_k|\mathcal{Z}^{k-1})\right] = p(\mathrm{x}^g_k|m, \mathcal{Z}^{k-1}) = \int d\mathrm{x}^r_k\, p(\mathrm{x}^g_k, \mathrm{x}^r_k|m, \mathcal{Z}^{k-1}) = \int d\mathrm{x}^r_k\, \underbrace{p(\mathrm{x}^g_k|\mathrm{x}^r_k, m)}_{\text{transformation}}\, \underbrace{p(\mathrm{x}^r_k|m, \mathcal{Z}^{k-1})}_{\text{Prior}}.$$

m: road segment

Transformation:

$$p(\mathrm{x}^g_{k+1}|\mathrm{x}^r_{k+1}) = \mathcal{N}\left(\mathrm{x}^g_{k+1};\, \mathrm{t}_{g\leftarrow r}[\mathrm{x}^r_{k+1}],\, \sigma^2_m\right)$$
$$\mathrm{t}_{g\leftarrow r}[\mathrm{x}_r] = \begin{pmatrix} \mathrm{t} & 0 \\ 0 & \mathrm{t}\end{pmatrix}\mathrm{x}_r + \begin{pmatrix} \mathrm{s} - l\mathrm{t} \\ 0\end{pmatrix}$$

→ simple affine transform along the segment $\mathcal{R}(l) = \mathrm{s} + l\mathrm{t}$.

Prior:

$$p(\mathrm{x}^r_k|m, \mathcal{Z}^{k-1}) = \frac{p(m|\mathrm{x}^r_k)\, p(\mathrm{x}^r_k|\mathcal{Z}^{k-1})}{\int d\mathrm{x}^r_k\, p(m|\mathrm{x}^r_k)\, p(\mathrm{x}^r_k|\mathcal{Z}^{k-1})} = p^m_{k-1}\, \mathcal{N}\left(\mathrm{x}^r_k;\, \mathrm{x}^{rmj}_{k|k-1},\, \mathbf{P}^{rmj}_{k|k-1}\right)$$

→ Model different errors for each segment
→ Kalman-type update function

Prob. of road segment:

$$p(m|\mathrm{x}^r_k) = \chi_m(\mathbf{H}_r\mathrm{x}^r_k) \approx \exp[-\tfrac{1}{2}(z^m_r - \mathbf{H}_r\mathrm{x}_r)^2/\lambda^2_m]$$

→ approximate via moment matching

Data Augmentation using EM

Idea: Directly estimating the state from the given measurements might not be feasible:

$$p(\mathcal{X}|\mathcal{Z})$$

Thus, we can incorporat additional information and augment our pdf to better estimate the state:

$$p(\mathcal{X}|\mathcal{A}, \mathcal{Z})$$

Employ expectation maximization to first incorporate the additional information and then re-compute the filters

Iterate until $|\mathcal{Q}(\mathcal{X}^{l+1}; \mathcal{X}^l, \mathcal{P}^l) - \mathcal{Q}(\mathcal{X}^l; \mathcal{X}^l, \mathcal{P}^l)| < \epsilon$:

E-step: Incorporate the additional information into the state estimate:

$$\mathcal{Q}(\mathcal{X}; \mathcal{X}^l, \mathcal{P}^l) = \int_{\mathrm{dom}\mathcal{A}} d\mathcal{A} \log[p(\mathcal{X}|\mathcal{A}, \mathcal{Z})]p(\mathcal{A}|\mathcal{X}^l, \mathcal{P}^l, \mathcal{Z})$$

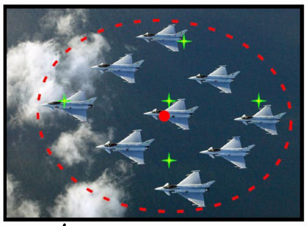
M-step: Given the estimates recompute filters

$$\mathcal{X}^{l+1} = \mathrm{argmax}_{\mathcal{X}}\mathcal{Q}(\mathcal{X}; \mathcal{X}^l, \mathcal{P}^l), \quad p(\mathcal{X}^{l+1}|\mathcal{Z}) > p(\mathcal{X}^l|\mathcal{Z})$$

Extended Target Tracking

Idea: Eventually we wish to track not a single point but rather a target (or multiple targets) with a given extension. For this, we augment the object state and track state and extension separately

$$X_k = (\mathbf{x}_k, \mathbf{X}_k)$$



Measurement Model:

Individual measurements are modeled as measurements of the target center. Thus the measurement error is proportional to the extension of the object.

Measurement equation:

$$\begin{aligned} \mathbf{z}_k^j &= (h_k^1\mathbf{I}_d, h_k^2\mathbf{I}_d, h_k^3\mathbf{I}_d)\mathbf{x}_k + \mathbf{u}_k, \quad \mathbf{u}_k \sim \mathcal{N}(\mathbf{o}, \mathbf{R}_k) \\ &= (\mathbf{H}_k \otimes \mathbf{I}_d)\mathbf{x}_k + \mathbf{u}_k \end{aligned}$$

Center measurement:

$$\mathbf{z}_k = \frac{1}{n_k}\sum_{j=1}^{n_k} \mathbf{z}_k^j, \quad \mathbf{Z}_k = \sum_{j=1}^{n_k} (\mathbf{z}_k^j - \mathbf{z}_k)(\mathbf{z}_k^j - \mathbf{z}_k)^\top$$

Measurement error:

$\mathbf{R}_k \propto \mathbf{X}_k$ ***unknown!***

Measurement Likelihood:

$$\begin{aligned} p(Z_k | n_k, \mathbf{x}_k, \mathbf{X}_k) &= \prod_{j=1}^{n_k} \mathcal{N}\left(\mathbf{z}_k^j;\ (\mathbf{H}_k \otimes \mathbf{I}_d)\mathbf{x}_k,\ \mathbf{X}_k\right) \quad \text{(independent plots)} \\ &\propto \mathcal{N}\left(\mathbf{z}_k;\ (\mathbf{H}_k \otimes \mathbf{I}_d)\mathbf{x}_k,\ \tfrac{1}{n_k}\mathbf{X}_k\right)\ \mathcal{LW}\left(\mathbf{Z}_k;\ n_k - 1,\ \mathbf{X}_k\right) \\ &= \mathcal{N}\left(\mathbf{z}_k;\ \mathbf{H}_k\mathbf{x}_k,\ \tfrac{1}{n_k}\mathbf{X}_k\right) \prod_{i=1}^{n_k-1} \mathcal{N}\left(\mathbf{z}_k^{i+1};\ \bar{\mathbf{z}}_k^i,\ \tfrac{i+1}{i}\mathbf{X}_k\right) \end{aligned}$$

→ wishart density is a column-wise gaussian and can be represented through a product of gaussians

$$\begin{aligned} \prod_{i=1}^{n_k-1} \mathcal{N}\left(\mathbf{z}_k^{i+1};\ \bar{\mathbf{z}}_k^i,\ \tfrac{i+1}{i}\mathbf{X}_k\right) &\propto |\mathbf{X}_k|^{-\frac{n_k-1}{2}}\ \mathrm{etr}\left[-\tfrac{1}{2}\mathbf{Z}_k\mathbf{X}_k^{-1}\right] \\ &\propto \mathcal{LW}\left(\mathbf{Z}_k;\ n_k - 1,\ \mathbf{X}_k\right). \end{aligned}$$

Evolution Model:

Kinematic state: $\mathbf{x}_k = \left(\mathbf{r}_k^\top, \dot{\mathbf{r}}_k^\top, \ddot{\mathbf{r}}_k^\top\right)^\top$

Temporal evolution:

$$\mathbf{x}_k = \boldsymbol{\Phi}_{k|k-1}\mathbf{x}_{k-1} + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathcal{N}(\mathbf{o}, \boldsymbol{\Delta}_{k|k-1})$$

$$\boldsymbol{\Phi}_{k|k-1} = \begin{pmatrix} \mathbf{I}_d & \Delta t_k \mathbf{I}_d & \frac{1}{2}\Delta t_k^2 \mathbf{I}_d \\ \mathbf{O}_d & \mathbf{I}_d & \Delta t_k \mathbf{I}_d \\ \mathbf{O}_d & \mathbf{O}_d & \mathrm{e}^{-\Delta t_k/\theta}\mathbf{I}_d \end{pmatrix} = \mathbf{F}_{k|k-1} \otimes \mathbf{I}_d$$

Plant noise: $\boldsymbol{\Delta}_{k|k-1} = \mathbf{D}_{k|k-1} \otimes \mathbf{X}_k$

→ allows mutually conjugate pairs of pdfs that produce analytic update equations

→ small extends: low maneuverability → low error
large extend: individual movements → unimportant prediction

Prediction:

$$p(\mathbf{x}_k, \mathbf{X}_k | \mathcal{Z}^{k-1}) = \underbrace{p(\mathbf{x}_k | \mathbf{X}_k, \mathcal{Z}^{k-1})}_{\textit{vector}\text{ variate}} \; \underbrace{p(\mathbf{X}_k | \mathcal{Z}^{k-1})}_{\textit{matrix}\text{ variate}}$$

Kinematic:

$$p(\mathbf{x}_k | \mathbf{X}_k, \mathcal{Z}^{k-1}) = \mathcal{N}\big(\mathbf{x}_k;\, \mathbf{x}_{k|k-1},\, \mathbf{P}_{k|k-1} \otimes \mathbf{X}_k\big)$$

$$\mathbf{x}_{k|k-1} = (\mathbf{F}_{k|k-1} \otimes \mathbf{I}_d)\mathbf{x}_{k-1|k-1}, \quad \mathbf{P}_{k|k-1} = \mathbf{F}_{k|k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k|k-1}^{\top} + \mathbf{D}_{k|k-1}$$

Extend:

$$p(\mathbf{X}_k | \mathcal{Z}^{k-1}) = \mathcal{IW}\big(\mathbf{X}_k;\, \nu_{k|k-1},\, \mathbf{X}_{k|k-1}\big) \propto |\mathbf{X}_k|^{-\frac{\nu_{k|k-1}+d+1}{2}} \operatorname{etr}\big[-\tfrac{1}{2}\mathbf{X}_{k|k-1}\mathbf{X}_k^{-1}\big].$$

$$\mathbb{E}[\mathbf{X}_k] = \frac{\mathbf{X}_{k|k-1}}{\nu_{k|k-1} - d - 1}, \quad \operatorname{etr}[\mathbf{A}] = \exp[\operatorname{tr}\mathbf{A}], \; |\mathbf{A}| = \det \mathbf{A}$$

Idea: Assume inverse Wishart densities which describe that X^{-1} columns are distributed according to a multivariate gaussian

Filtering:

$$p(Z_k | n_k, \mathbf{x}_k, \mathbf{X}_k)\, p(\mathbf{x}_k, \mathbf{X}_k | \mathcal{Z}^{k-1}) \propto \underbrace{\mathcal{N}\big(\mathbf{x}_k;\, \mathbf{x}_{k|k},\, \mathbf{P}_{k|k} \otimes \mathbf{X}_k\big)}_{\text{KALMAN type update}} \; \underbrace{\mathcal{IW}\big(\mathbf{X}_k;\, \nu_{k|k},\, \mathbf{X}_{k|k}\big)}_{\text{extension update}}$$

$$\mathbf{X}_{k|k} = \mathbf{X}_{k|k-1} + \mathbf{N}_{k|k-1} + \mathbf{Z}_k, \quad \nu_{k|k} = \nu_{k|k-1} + n_k.$$

Structure:

exploiting BAYES:

$$p(Z_k | n_k, \mathbf{x}_k, \mathbf{X}_k)\, p(\mathbf{x}_k, \mathbf{X}_k | \mathrm{Z}^{k-1}) \propto \underbrace{\mathcal{N}\big(\mathbf{x}_k;\, \mathbf{x}_{k|k},\, \mathbf{P}_{k|k} \otimes \mathbf{X}_k\big)}_{\text{KALMAN type update}}$$

$$\times \underbrace{|\mathbf{X}_k|^{-\frac{1}{2}} \operatorname{etr}\big[-\tfrac{1}{2}\mathbf{N}_{k|k-1}\mathbf{X}_k^{-1}\big]}_{\text{innovation factor}} \; \underbrace{\mathcal{LW}\big(\mathbf{Z}_k;\, n_k - 1,\, \mathbf{X}_k\big)}_{\text{from measurement likelihood}} \; \underbrace{\mathcal{IW}\big(\mathbf{X}_k;\, \nu_{k|k-1},\, \mathbf{X}_{k|k-1}\big)}_{\text{extension prediction}}$$

$$\mathbf{N}_{k|k-1} = S_{k|k-1}^{-1}\big(\mathbf{z}_k - (\mathbf{H}_k \otimes \mathbf{I}_d)\mathbf{x}_{k|k-1}\big)\big(\mathbf{z}_k - (\mathbf{H}_k \otimes \mathbf{I}_d)\mathbf{x}_{k|k-1}\big)^{\top}$$

Retrieve kinematic state:

→ simply marginalize over extend

$$\begin{aligned} p(\mathbf{x}_k | \mathcal{Z}_k) &= \textstyle\int d\mathbf{X}_k\, p(\mathbf{x}_k, \mathbf{X}_k | \mathcal{Z}^k) \\ &= \textstyle\int d\mathbf{X}_k\, \mathcal{N}\big(\mathbf{x}_k;\, \mathbf{x}_{k|k},\, \mathbf{P}_{k|k} \otimes \mathbf{X}_k\big)\, \mathcal{IW}\big(\mathbf{X}_k;\, \nu_{k|k},\, \mathbf{X}_{k|k}\big) \\ &= \mathcal{T}\big(\mathbf{x}_k;\, \nu_{k|k} + s(1-d),\, \mathbf{x}_{k|k},\, \mathbf{P}_{k|k} \otimes \mathbf{X}_{k|k}\big) \quad \textbf{(standard algebra!)} \end{aligned}$$

multivariate Student-t-density with $\nu_{k|k}$ degrees of freedom

PMHT for extended objects

Idea: Compute estimates of multiple extended targets. The problem is that no assignments are given and it is inherently hard to make the data association

Given: Sequence of Measurements comprising Noise and Target Data

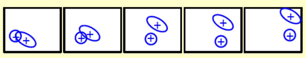
$\mathcal{Z} = [Z_t]_{t=1}^{T}$

$Z_t \equiv$ Data at scan t

Wanted: Sequences of **Joint Target States** (kinematics, extents)

$\mathcal{X} = [X_t]_{t=1}^{T}$

$X_t = [\mathbf{x}_t^m, \mathbf{X}_t^m]_{m=1}^{M}$



E-step: Compute assignments:

$$w_t^{rm(i)} = \frac{\pi_t^{m(i)} \mathcal{N}\Big(\mathbf{z}_t^r; \mathbf{H}\mathbf{x}_t^{m(i)}, \mathbf{X}_t^{m(i)} + \mathbf{R}\Big)}{\sum_{m'=0}^{M} \pi_t^{m'(i)} \mathcal{N}\Big(\mathbf{z}_t^r; \mathbf{H}\mathbf{x}_t^{m'(i)}, \mathbf{X}_t^{m(i)} + \mathbf{R}\Big)}$$

M-step: Given the assignments re-compute extended target filters

1. Run a **Bank of M Bayesian Extended Target Filters** (one per Target) using Synthetic Measurements and Spread Matrices

   Result: $\Big[\mathbf{x}_1^{m(i+1)}, \ldots, \mathbf{x}_T^{m(i+1)}\Big], \Big[\mathbf{X}_1^{m(i+1)}, \ldots, \mathbf{X}_T^{m(i+1)}\Big] \;\forall m$

2. Update **Mixing Proportions** for each Target

   Result: $\Big[\pi_1^{m(i+1)}, \ldots, \pi_T^{m(i+1)}\Big] \;\forall m$

Filter Consistancy

Idea: A filter is consistent, if we observe an error with zero mean and variance corresponding to our predicted covariance

$$\Delta \mathbf{x}_{k|k} = \mathbf{x}_k - \mathbf{x}_{k|k} \quad \text{with:} \quad \begin{aligned} \mathbb{E}[\Delta \mathbf{x}_{k|k}] &= 0 \\ \mathbb{E}[\Delta \mathbf{x}_{k|k} \Delta \mathbf{x}_{k|k}^\top] &= \mathbf{P}_{k|k} \end{aligned}$$

In practice a filter is consistent, if the NEES and NIS are χ^2-distributed with n_x or n_z DoF.

Error Metrics:

Normalized estimation error squared: $\boxed{\epsilon(k) = \Delta \mathbf{x}_{k|k}^\top \mathbf{P}_{k|k}^{-1} \Delta \mathbf{x}_{k|k}}$

Normalized innovation squared: $\boxed{\epsilon_\nu(k) = \boldsymbol{\nu}_{k|k-1}^\top \mathbf{S}_{k|k}^{-1} \boldsymbol{\nu}_{k|k-1}}$

$$\boldsymbol{\nu}_{k|k-1} = \mathbf{z}_k - \mathbf{H}_{k|k-1} \mathbf{x}_{k|k-1}, \quad \mathbf{S}_{k|k-1} = \mathbf{H}_{k|k-1} \mathbf{P}_{k|k-1} \mathbf{H}_{k|k-1}^\top + \mathbf{R}_{k|k-1}$$

Test:

1. Simulate N random, independent time series:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{u}_k$$

2. Compute the errors and error mean:

$$\epsilon^i(k),\ i = 1, \dots, N, \qquad \bar{\epsilon}(k) = \tfrac{1}{N} \textstyle\sum_{i=1}^N \epsilon^i(k)$$

3. Decide on consistancy using a two-side test depending on the significance α

Acceptance interval:
$$\boxed{\begin{aligned} &P(\bar{\epsilon}(k) \in [r_1, r_2] | \text{konsist.}) = 1 - \alpha \\ &r_1 = \chi^2_{Nn_z}(\alpha/2)/N,\ r_2 = \chi^2_{Nn_z}(1-\alpha/2)/N \end{aligned}}$$

Accept, if: $\bar{\epsilon}(k) \in [r_1, r_2]$