$$\begin{tikzcd}
{} \arrow[r, "\begin{array}{c}\mathbf{x}(k)= \\ F\left(\mathbf{x}(k-1),\mathbf{u}(k-1)\right)\end{array}", "\textit{prediction}"'] & p_{\mathbf{x}(k)|\mathbf{Y}(k-1)}\left(\mathbf{x}(k)\,|\,\mathbf{Y}(k-1)\right) \arrow[r, "\text{update}", "\begin{array}{c}\text{Apply}\\ \text{observations}\\ \text{performed at time k}\end{array}"'] & p_{\mathbf{x}(k)|\mathbf{Y}(k)}\left(\mathbf{x}(k)\,|\,\mathbf{Y}(k)\right) \arrow[dll] \\
{} \arrow[r, "\begin{array}{c}\mathbf{x}(k+1)= \\ F\left(\mathbf{x}(k),\mathbf{u}(k)\right)\end{array}", "\textit{prediction}"'] & p_{\mathbf{x}(k+1)|\mathbf{Y}(k)}\left(\mathbf{x}(k+1)\,|\,\mathbf{Y}(k)\right) \arrow[r, "\text{update}", "\begin{array}{c}\text{Apply}\\ \text{observations}\\ \text{performed at time k+1}\end{array}"'] & p_{\mathbf{x}(k+1)|\mathbf{Y}(k+1)}\left(\mathbf{x}(k+1)\,|\,\mathbf{Y}(k+1)\right) \arrow[dll] \\
{} \arrow[r, "\begin{array}{c}\mathbf{x}(k+2)= \\ F\left(\mathbf{x}(k+1),\mathbf{u}(k+1)\right)\end{array}", "\textit{prediction}"'] & p_{\mathbf{x}(k+2)|\mathbf{Y}(k+1)}\left(\mathbf{x}(k+2)\,|\,\mathbf{Y}(k+1)\right) \arrow[r, "\text{update}", "\begin{array}{c}\text{Apply}\\ \text{observations}\\ \text{performed at time k+2}\end{array}"'] & p_{\mathbf{x}(k+2)|\mathbf{Y}(k+2)}\left(\mathbf{x}(k+2)\,|\,\mathbf{Y}(k+2)\right) \arrow[dll] \\
{} & &
\end{tikzcd}$$

However, those operations are expensive (in processing and in memory usage, in a computer)

In addition: we usually want this process to operate in real-time.

→ We need to simplify the problem.

A way of achieving that is by using Gaussian PDFs.

With them, it is possible to simplify the general Bayesian process which results in a cheap and feasible implementation.

Gaussian PDFs.

1D case:

Equation

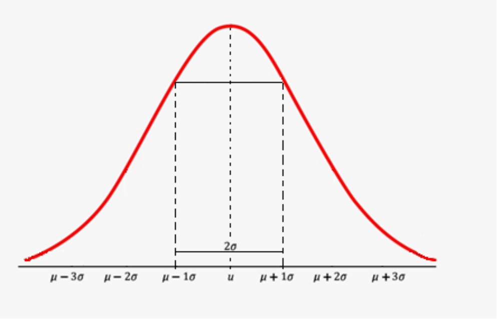
$$p_x(x) = c\cdot e^{-\frac{(x-\hat{x})^2}{2\cdot\sigma_x^2}}$$

Parameters

$\hat{x}$: expected value (scalar)

σ_x^2 : variance (scalar)

Visualization:



2D case:

Equation

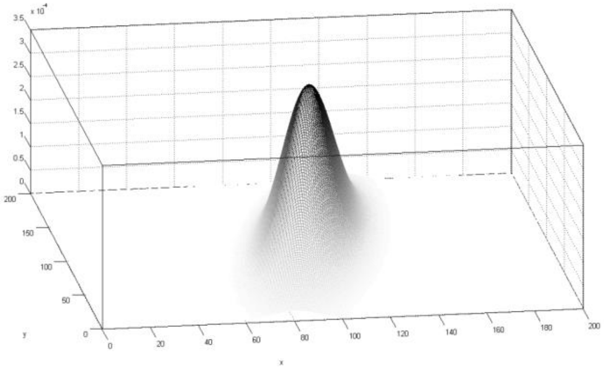
$$p_{\mathbf{X}}(\mathbf{X}) = c\cdot e^{-\frac{1}{2}\cdot(\mathbf{X}-\hat{\mathbf{X}})^T\cdot\mathbf{P}^{-1}\cdot(\mathbf{X}-\hat{\mathbf{X}})}$$

Parameters

$\hat{\mathbf{x}}$: expected value (vector 2x1)

$\mathbf{P}_x$: covariance matrix (matrix 2x2)

Visualization:



n-D case:

Equation

$$p_{\mathbf{X}}(\mathbf{X}) = c\cdot e^{-\frac{1}{2}\cdot(\mathbf{X}-\hat{\mathbf{X}})^T\cdot\mathbf{P}_{\mathbf{X}}^{-1}\cdot(\mathbf{X}-\hat{\mathbf{X}})}$$

Parameters

$\hat{\mathbf{x}}$: expected value (vector $\mathrm{n}\times 1$)

$\mathbf{P}_x$: covariance matrix (matrix $\mathrm{n}\times\mathrm{n}$)

Visualization:

?

Gaussian RVs. Do they actually happen?

YES. The Central Limit Theorem may explain it.

The Central limit theorem (CLT) says that, in many cases, in which a number of independent RVs are linearly combined, the resulting RV will have statistical properties like those of a Gaussian RV (its PDF would be of Gaussian shape)

This situation is also valid even if the linearly combined RVs are not Gaussian! (e.g. uniformly distributed)

CLT is well backed by many practical cases, in which the resulting PDFs are usually unimodal and close to Gaussian. In

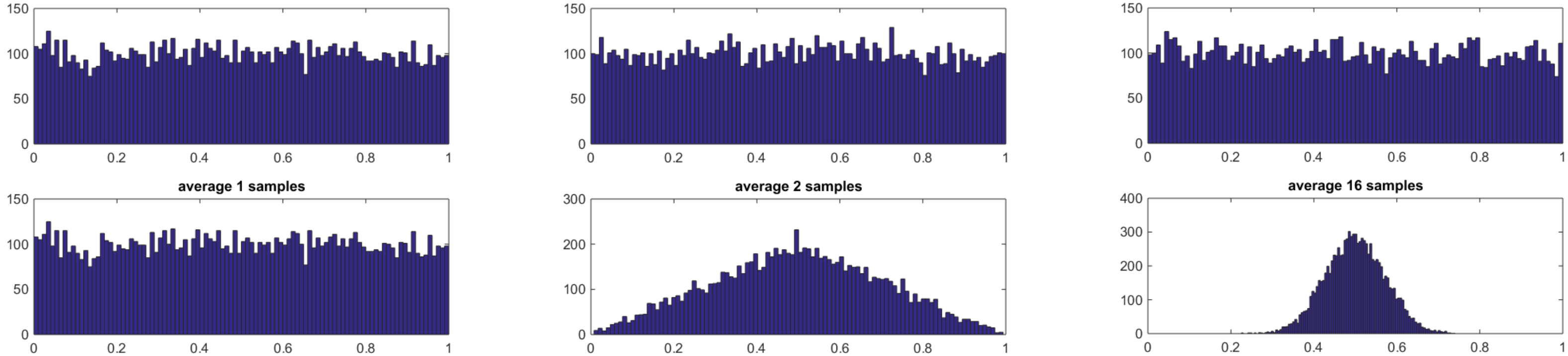
terms of Bayesian estimation, estimation processes do combine multiple sources of information, which are usually polluted by independent sources of uncertainty (not necessarily Gaussian), which results in linear combinations of independent RVs, resulting in estimates whose statistical properties are close to Gaussian.

Central Limit Theorem (CLT)

The CLT can be demonstrated (but that is not our business here, in AAS)

Numerical example in 1D (adding uniform RVs)

(it gets Gaussian!)



average 1 samples

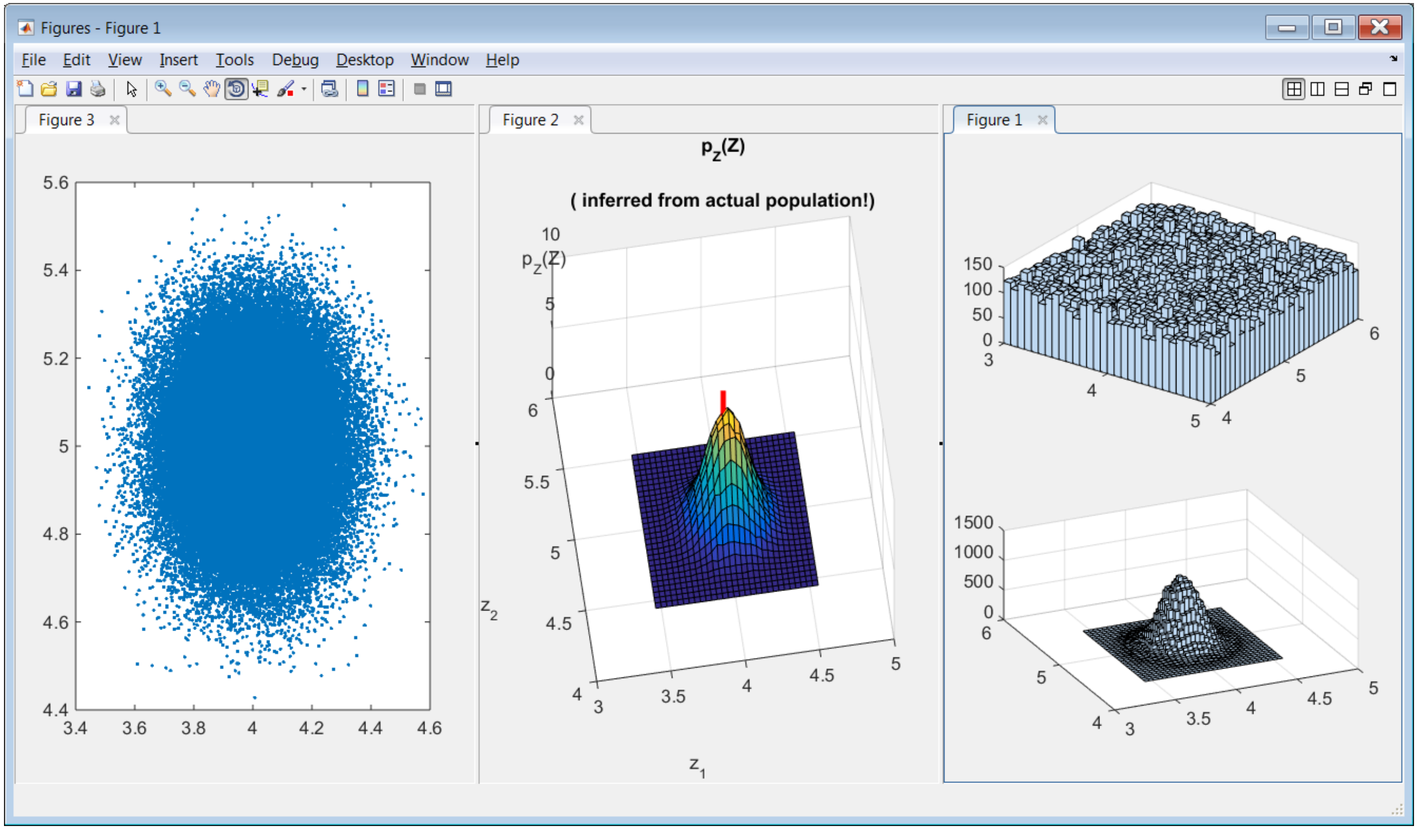
average 2 samples

average 16 samples

Central Limit Theorem (CLT)

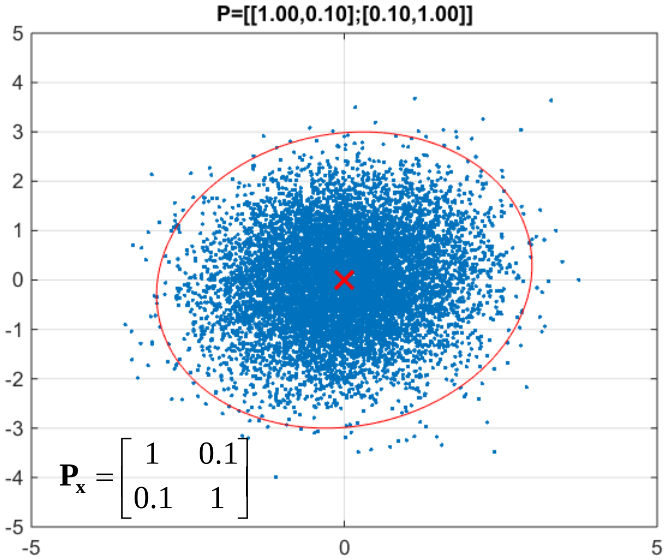
Numerical example in 2D (adding uniform RVs)

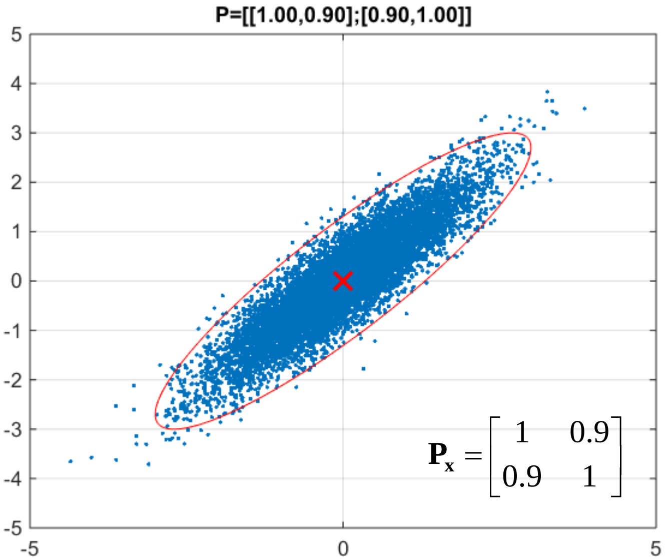
(it gets Gaussian!)

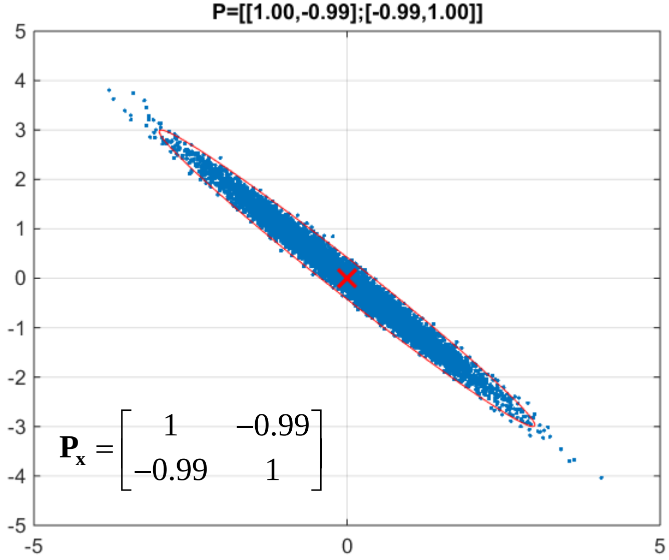


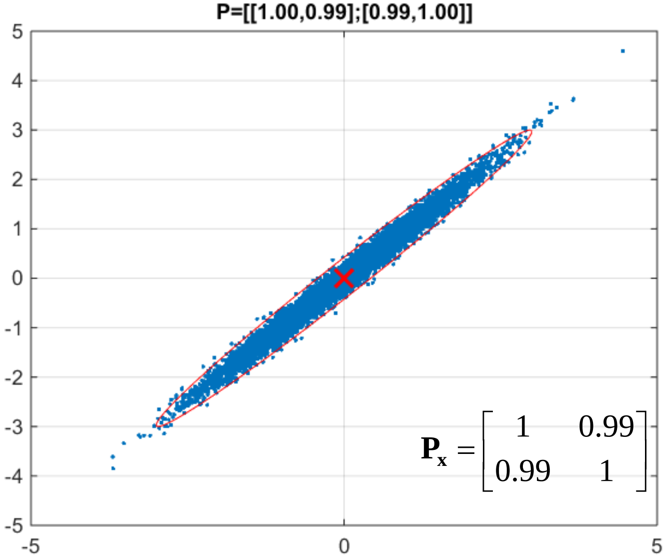
“Feeling” the covariance matrix . Example in 2D.

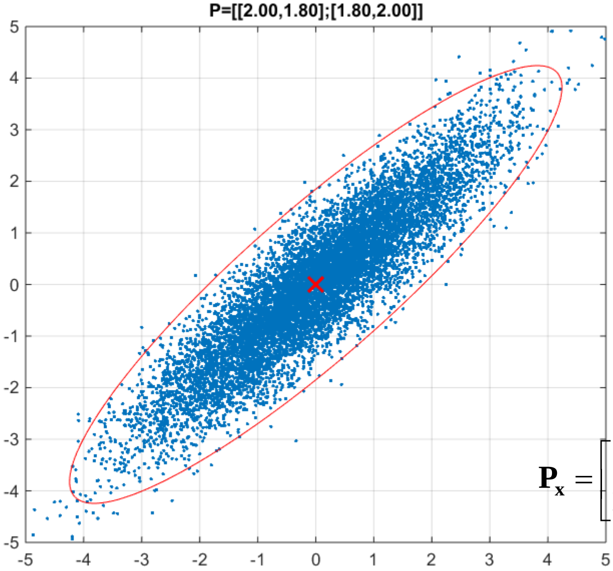
2D Covariance Matrix define the typical population , respect to expected value.



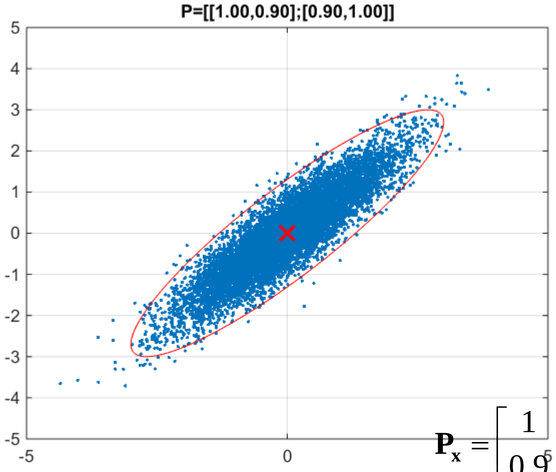




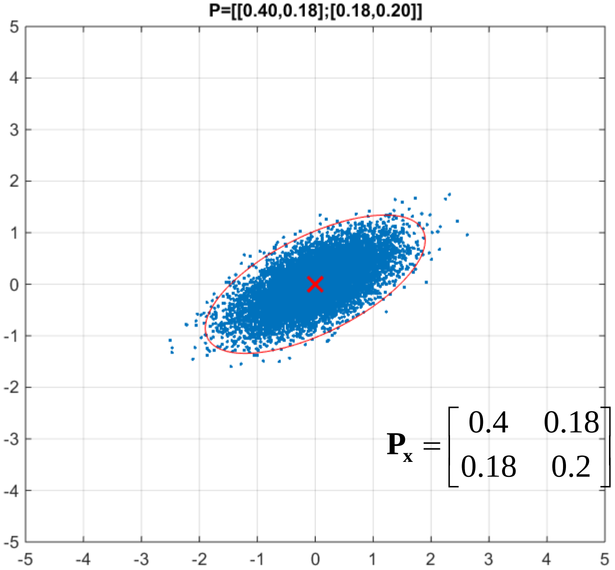




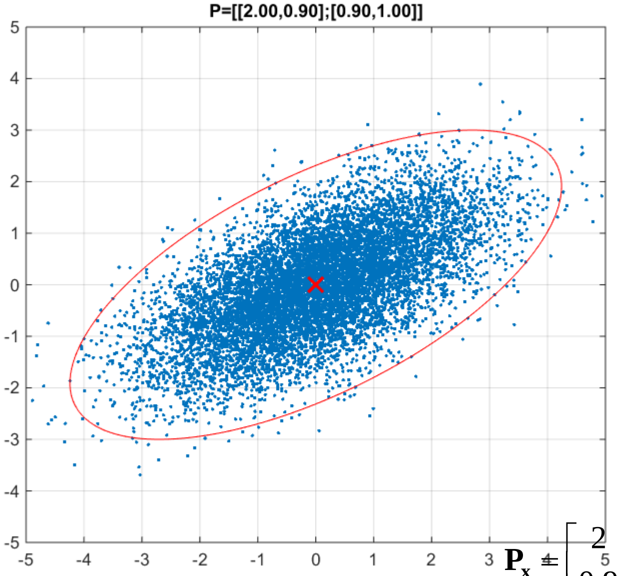
$$\mathbf{P_x} = \begin{bmatrix} 2 & 1.8 \\ 1.8 & 2 \end{bmatrix}$$



$$\mathbf{P_x} = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}$$



$$\mathbf{P_x} = \begin{bmatrix} 0.4 & 0.18 \\ 0.18 & 0.2 \end{bmatrix}$$



$$\mathbf{P_x} = \begin{bmatrix} 2 & 0.9 \\ 0.9 & 1 \end{bmatrix}$$

(Larger variances → higher uncertainty)

Gaussian PDF

A Bayesian Estimation process can be dramatically simplified if the involved PDF functions are of the Gaussian family.

Now, we will show how Gaussian RVs “behave” under certain operations.

Linear Transformation Applied on Gaussian RVs

Certain usual operations (linear and affine transformations), are easily treated when applied to Gaussian RVs.

If we add two Gaussian RVs:

- Resulting RV is Gaussian
- Expected value and covariance matrix can be easily evaluated,

Adding two independent Gaussian RVs

$$\mathbf{x} \in \mathrm{R}^n, \mathbf{y} \in \mathrm{R}^n, \quad \mathbf{x} \sim N\left(\mathbf{x}\,;\hat{\mathbf{x}},\mathbf{P}_{\mathbf{x}}\right), \quad \mathbf{y} \sim N\left(\mathbf{y}\,;\hat{\mathbf{y}},\mathbf{P}_{\mathbf{y}}\right),$$

$$\mathbf{x},\mathbf{y} \text{ statistically independent}$$

$$\mathbf{z} = \mathbf{x} + \mathbf{y}$$

$$\Downarrow$$

$$\mathbf{z} \sim N\left(\mathbf{z}\,;\hat{\mathbf{z}},\mathbf{P}_{\mathbf{z}}\right) \quad : \qquad \boxed{\hat{\mathbf{z}} = \hat{\mathbf{x}} + \hat{\mathbf{y}}, \quad \mathbf{P}_{\mathbf{z}} = \mathbf{P}_{\mathbf{x}} + \mathbf{P}_{\mathbf{y}}}$$

→ Very “cheap” (in terms of processing and memory requirements, when we implement it in a program)

→ Very important/useful, for when we implement PREDICTIONS

→ (we easily obtain the parameters that correspond to the PDF which is the result of the expensive convolution operation we had seen before for general Bayesian cases)

Linear transformation of a Gaussian RV

$$\mathbf{z} = \mathbf{A} \cdot \mathbf{x}$$

$$\mathbf{z} = \begin{bmatrix} z_1 & z_2 & .. & z_m \end{bmatrix}^T \in R^m, \quad \mathbf{x} = \begin{bmatrix} x_1 & x_2 & .. & x_n \end{bmatrix}^T \in R^n, \quad \mathbf{A} \in R^{m \times n}$$

$$\mathbf{x} \sim N\left(\mathbf{x}\,;\hat{\mathbf{x}},\mathbf{P}_{\mathbf{x}}\right)$$

$$\Downarrow$$

$$\mathbf{z} \sim N\left(\mathbf{z}\,;\hat{\mathbf{z}},\mathbf{P}_{\mathbf{z}}\right)$$

$$\boxed{\begin{gathered} \hat{\mathbf{z}} = \mathbf{A} \cdot \hat{\mathbf{x}} \\ \mathbf{P}_{\mathbf{z}} = \mathbf{A} \cdot \mathbf{P}_{\mathbf{x}} \cdot \mathbf{A}^T \end{gathered}}$$

→ Very important, for when we implement PREDICTIONS

→ Very efficient, again.

also valid for affine cases

$$\mathbf{z} = \mathbf{A} \cdot \mathbf{x} + \mathbf{b}$$ (in which $\mathbf{b}$ is deterministic)

$$\boxed{\hat{\mathbf{z}} = \mathbf{A} \cdot \hat{\mathbf{x}} + \mathrm{b}; \quad \mathbf{P}_{\mathbf{z}} = \mathbf{A} \cdot \mathbf{P}_{\mathbf{x}} \cdot \mathbf{A}^T}$$

Product of Gaussian PDFs

If we multiply functions that are of the Gaussian family, the result is a Gaussian function as well.

$$g_1(\mathbf{x}) = \frac{1}{c_1} e^{(\mathbf{xx}-\boldsymbol{\mu}_1)^T \cdot \mathbf{M}_1 \cdot (\mathbf{x}-\boldsymbol{\mu}_1)}$$

$$g_2(\mathbf{x}) = \frac{1}{c_2} e^{(\mathbf{x}-\boldsymbol{\mu}_2)^T \cdot \mathbf{M}_2 \cdot (\mathbf{x}-\boldsymbol{\mu}_2)}$$

$$\Downarrow$$

$$g_3(\mathbf{x}) = g_1(\mathbf{x}) \cdot g_2(\mathbf{x}) = \frac{1}{c_3} e^{(\mathbf{x}-\boldsymbol{\mu}_3)^T \cdot \mathbf{M}_3^{-1} \cdot (\mathbf{x}-\boldsymbol{\mu}_3)}$$

**There is an efficient way for obtaining $\boldsymbol{\mu}_3$ and $\mathbf{M}_3$

(we will see it later)

→Very useful, for multiplying Likelihood functions and prior PDFs (“updates”)

Examples: (adding Independent Gaussian RVs)

Scalar RVs:

$$\mathbf{x} \sim N\left(\mathbf{x};\, \hat{\mathbf{x}}, \mathbf{P}_\mathrm{x}\right) = N\left(\mathbf{x};\, 5\ ,\ 1\right)$$

$$\mathbf{y} \sim N\left(\mathbf{y};\, \hat{\mathbf{y}}, \mathbf{P}_\mathrm{y}\right) = N\left(\mathbf{y}; 3\ ,\ 4\right)$$

$$\mathbf{z} = \mathbf{x} + \mathbf{y} \qquad \Rightarrow \qquad \mathbf{z} \sim N\left(\mathbf{z}; 5+3\ ,\ 1+4\right) = N\left(\mathbf{z}; 8\ ,\ 5\right)$$

Multivariate RV’s

$$\mathbf{x} \sim N\left(\mathbf{x};\, \hat{\mathbf{x}}, \mathbf{P}_\mathrm{x}\right) \quad ; \quad \hat{\mathbf{x}} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \mathbf{P}_\mathrm{x} = \begin{bmatrix} 4 & 1 \\ 1 & 2 \end{bmatrix}$$

$$\mathbf{y} \sim N\left(\mathbf{y};\, \hat{\mathbf{y}}, \mathbf{P}_\mathrm{y}\right) \quad ; \quad \hat{\mathbf{y}} = \begin{bmatrix} 3 \\ 6 \end{bmatrix}, \mathbf{P}_\mathrm{y} = \begin{bmatrix} 6 & 0 \\ 0 & 9 \end{bmatrix}$$

$$\mathbf{z} = \mathbf{x} + \mathbf{y} \qquad \Rightarrow \qquad \hat{\mathbf{z}} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 3 \\ 6 \end{bmatrix} = \begin{bmatrix} 4 \\ 8 \end{bmatrix}; \quad \mathbf{P}_\mathrm{z} = \begin{bmatrix} 4 & 1 \\ 1 & 2 \end{bmatrix} + \begin{bmatrix} 6 & 0 \\ 0 & 9 \end{bmatrix} = \begin{bmatrix} 10 & 1 \\ 1 & 11 \end{bmatrix}$$

Examples:

$$\hat{\mathbf{z}} = \mathbf{A} \cdot \hat{\mathbf{x}}$$

$$\mathbf{P}_{\mathbf{z}} = \mathbf{A} \cdot \mathbf{P}_{\mathbf{x}} \cdot \mathbf{A}^T$$

Scalar RVs:

$$\mathbf{x} \sim N\left(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P}_{\mathbf{x}}\right) = N\left(\mathbf{x}; 5\ ,\ 1\right), \quad \mathbf{z} = 3 \cdot \mathbf{x}$$

$$\Rightarrow \qquad \mathbf{z} \sim N\left(\mathbf{z};\ 3 \cdot 5\ ,\ 3 \cdot 1 \cdot 3\right) = N\left(\mathbf{z};\ 15\ ,\ 9\right)$$

Multivariate

$$\mathbf{x} \sim N\left(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P}_{\mathbf{x}}\right) \quad ; \quad \hat{\mathbf{x}} = \begin{bmatrix} 1.1 \\ 2.1 \end{bmatrix}, \quad \mathbf{P}_{\mathbf{x}} = \begin{bmatrix} 4 & 0.2 \\ 0.2 & 1 \end{bmatrix}$$

$$\mathbf{z} = \begin{bmatrix} 4 & 5 \end{bmatrix} \cdot \mathbf{x} \qquad \Rightarrow \qquad \hat{\mathbf{z}} = \begin{bmatrix} 4 & 5 \end{bmatrix} \cdot \begin{bmatrix} 1.1 \\ 2.1 \end{bmatrix}; \quad \mathbf{P}_{\mathbf{z}} = \begin{bmatrix} 4 & 5 \end{bmatrix} \cdot \begin{bmatrix} 4 & 0.2 \\ 0.2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 4 & 5 \end{bmatrix}^T == [\ldots]_{1 \times 1}$$

$$\mathbf{z} \in R^1$$

$$\mathbf{w} = \begin{bmatrix} -3 & 0 \\ 1 & 1 \end{bmatrix} \cdot \mathbf{x} \qquad \Rightarrow \qquad \hat{\mathbf{w}} = \begin{bmatrix} -3 & 0 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1.1 \\ 2.1 \end{bmatrix}; \quad \mathbf{P}_{\mathbf{w}} = \begin{bmatrix} -3 & 0 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 4 & 0.2 \\ 0.2 & 1 \end{bmatrix} \cdot \begin{bmatrix} -3 & 0 \\ 1 & 1 \end{bmatrix}^T = [\ldots]_{2 \times 2}$$

Nonlinear transformation of a Gaussian RV

$$\mathbf{z} = \mathbf{a}(\mathbf{x})$$

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \ldots \\ z_m \end{bmatrix} \in R^m, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix} \in R^n, \quad \mathbf{a}(\ \ ): R^n \to R^m$$

In many cases (*), the PDF of z, can be well approximated by a Gaussian PDF.

(*) under certain conditions. This is similar to those linearizations discussed/applied in Math and in MMAN3200.

$$\mathbf{z} = \mathbf{a}(\mathbf{x})$$

$$\mathbf{x} \sim N\left(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P}_{\mathbf{x}}\right), \quad \mathbf{z} \in R^m, \quad \mathbf{x} \in R^n, \quad \mathbf{a}(\quad) : R^n \to R^m$$

$$\boxed{\begin{gathered} \mathbf{x} \sim N\left(\mathbf{z}; \hat{\mathbf{z}}, \mathbf{P}_{\mathbf{z}}\right) \\ \hat{\mathbf{z}} = \mathbf{a}\left(\hat{\mathbf{x}}\right) \\ \mathbf{P}_{\mathbf{z}} = \mathbf{A} \cdot \mathbf{P}_{\mathbf{x}} \cdot \mathbf{A}^T \end{gathered}}$$

$$\mathbf{A} = \left[\frac{\partial \mathbf{a}}{\partial \mathbf{x}}\right]_{\mathbf{x}=\hat{\mathbf{x}}}$$

$$\frac{\partial \mathbf{a}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial a_1}{\partial x_1} & \frac{\partial a_1}{\partial x_2} & \cdots & \frac{\partial a_1}{\partial x_n} \\ \frac{\partial a_2}{\partial x_2} & \cdots & \cdots & \frac{\partial a_2}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial a_m}{\partial x_1} & \cdots & \cdots & \frac{\partial a_m}{\partial x_n} \end{bmatrix}$$

(Jacobian Matrix)

(this is based on the linear approximation

$$\mathbf{a}(\mathbf{x}) \cong \mathbf{a}(\hat{\mathbf{x}}) + \left[\frac{\partial \mathbf{a}}{\partial \mathbf{x}}\right]_{\mathbf{x}=\hat{\mathbf{x}}} \cdot \left(\mathbf{x} - \hat{\mathbf{x}}\right)$$

)

Example:

..

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \sim N\left(\mathbf{x}\,; \hat{\mathbf{x}}, \mathbf{P}_\mathbf{x}\right), \quad \hat{\mathbf{x}} = \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix}, \quad \mathbf{P}_\mathbf{x} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

$$\mathbf{z} = h\left(\mathbf{x}\right) = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} h_1\left(x_1, x_2, x_3\right) \\ h_2\left(x_1, x_2, x_3\right) \end{bmatrix} = \begin{bmatrix} \sin\left(1 + \dfrac{x_1 + 2 \cdot x_2}{100}\right) + 5 \cdot x_3 \\ \left(1 + \dfrac{x_1}{100}\right)^3 + x_2 \end{bmatrix}$$

$$\mathbf{z} \sim N\left(\mathbf{z}\,; \hat{\mathbf{z}}, \mathbf{P}_\mathbf{z}\right)$$

$$\hat{\mathbf{z}} = \mathbf{h}\left(\hat{\mathbf{x}}\right)$$

$$\mathbf{P}_\mathbf{z} = \mathbf{H} \cdot \mathbf{P}_\mathbf{x} \cdot \mathbf{H}^T$$

$$\mathbf{H} = \left.\frac{\partial \mathbf{h}\left(\mathbf{x}\right)}{\partial \mathbf{x}}\right|_{\mathbf{x} = \hat{\mathbf{x}}}$$

..

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \sin\left(1+\dfrac{x_1+2\cdot x_2}{100}\right)+5\cdot x_3 \\ \left(1+\dfrac{x_1}{100}\right)^3 + x_2 \end{bmatrix}$$

$$\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \dfrac{\partial h_1}{\partial x_1} & \dfrac{\partial h_1}{\partial x_2} & \dfrac{\partial h_1}{\partial x_3} \\ \dfrac{\partial h_2}{\partial x_1} & \dfrac{\partial h_2}{\partial x_2} & \dfrac{\partial h_2}{\partial x_3} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{100}\cdot\cos\left(1+\dfrac{x_1+2\cdot x_2}{100}\right) & \dfrac{2}{100}\cdot\cos\left(1+\dfrac{x_1+2\cdot x_2}{100}\right) & 5 \\ \dfrac{3}{100}\cdot\left(1+0.01\cdot x_1\right)^2 & 1 & 0 \end{bmatrix}$$

$$\mathbf{H} = \left.\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\right|_{x=[0\quad 2\quad 1]^T} = \begin{bmatrix} \dfrac{1}{100}\cdot\cos\left(1+\dfrac{0+2\cdot 2}{100}\right) & \dfrac{2}{100}\cdot\cos\left(1+\dfrac{0+2\cdot 2}{100}\right) & 5 \\ \dfrac{3}{100}\cdot\left(1+0.01\cdot 0\right)^2 & 1 & 0 \end{bmatrix}_{2\times 3} =$$

$$= \begin{bmatrix} \dfrac{1}{100}\cdot\cos(1.04) & \dfrac{2}{100}\cdot\cos(1.04) & 5 \\ \dfrac{3}{100} & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0.0051 & 0.0102 & 5 \\ 0.03 & 1 & 0 \end{bmatrix}$$

We obtain expected value and Covariance matrix

..

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \sin\left(1+\frac{x_1+2\cdot x_2}{100}\right)+5\cdot x_3 \\ \left(1+\frac{x_1}{100}\right)^3+x_2 \end{bmatrix}$$

$$\hat{\mathbf{z}} = \begin{bmatrix} \sin(1.04)+5\cdot 1 \\ \left(1+\frac{0}{100}\right)^3+2 \end{bmatrix} = \begin{bmatrix} 5.9624 \\ 3 \end{bmatrix}$$

$$\mathbf{P_z} = \mathbf{H}\cdot\mathbf{P_x}\cdot\mathbf{H}^T =$$

$$= \begin{bmatrix} 0.0051 & 0.0102 & 5 \\ 0.03 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 0.0051 & 0.0102 & 5 \\ 0.03 & 1 & 0 \end{bmatrix}^T =$$

$$= \begin{bmatrix} 50.15 & 5.196 \\ 5.196 & 4.062 \end{bmatrix}_{(\,2\times 2\text{ matrix }\,)}$$

Marginal PDFs / EASY

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix} \in R^n, \qquad \mathbf{x} \sim p_{\mathbf{x}}(\mathbf{x}) = N(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P}_{\mathbf{x}}) \quad ;$$

$$p_{x_2}(x_2) = \int_{\text{domain of } x_1, x_3, .. x_n} p_{\mathbf{x}}(\mathbf{x}) \cdot dx_1 \cdot dx_3 \cdot .. \cdot dx_n \text{ // NO need to to this! (fortunately)}$$

$$p_{x_2}(x_2) = N(x_2; \hat{x}_2, \mathbf{P}_{x_2})$$

// this marginal PDF is Gaussian, and its paramers are easily obteined from those of the joint PDF.

$$\hat{x}_2 = \hat{\mathbf{x}}(2); \quad \mathbf{P}_{x_2} = \mathbf{P}_{\mathbf{x}}(2,2); \quad \text{// using MATLAB notation}$$

We may try the same procedure with any subset of components of **x**.

$$e.g., \quad \mathbf{w} = \begin{bmatrix} x_4 \\ x_6 \end{bmatrix}$$

$$\Downarrow$$

$$p_{\mathbf{w}}(\mathbf{w}) = N(\mathbf{w}; \hat{\mathbf{w}}, \mathbf{P}_{\mathbf{w}}),$$

$$\hat{\mathbf{w}} = \hat{\mathbf{x}}([4,6]); \quad \mathbf{P}_{\mathbf{w}} = \mathbf{P}_{\mathbf{x}}([4,6],[4,6]); \text{// using MATLAB notation}$$

Now,

We exploit those efficient operations

for applying Bayesian estimation.

As we remember,

that involved

a sequence of predictions and updates.

sequence of predictions and updates

$$\xrightarrow[\textit{prediction}]{\mathbf{x}(k)=F(\mathbf{x}(k-1),\mathbf{u}(k-1))} p_{\mathbf{x}(k)|\mathbf{Y}(k-1)}\left(\mathbf{x}(k)\,|\,\mathbf{Y}(k-1)\right) \xrightarrow[\text{Apply observations performed at time } k]{\text{update}} p_{\mathbf{x}(k)|\mathbf{Y}(k)}\left(\mathbf{x}(k)\,|\,\mathbf{Y}(k)\right)$$

$$\xrightarrow[\textit{prediction}]{\mathbf{x}(k+1)=F(\mathbf{x}(k),\mathbf{u}(k))} p_{\mathbf{x}(k+1)|\mathbf{Y}(k)}\left(\mathbf{x}(k+1)\,|\,\mathbf{Y}(k)\right) \xrightarrow[\text{Apply observations performed at time k+1}]{\text{update}} p_{\mathbf{x}(k+1)|\mathbf{Y}(k+1)}\left(\mathbf{x}(k+1)\,|\,\mathbf{Y}(k+1)\right)$$

$$\xrightarrow[\textit{prediction}]{\mathbf{x}(k+2)=F(\mathbf{x}(k+1),\mathbf{u}(k+1))} p_{\mathbf{x}(k+2)|\mathbf{Y}(k+1)}\left(\mathbf{x}(k+2)\,|\,\mathbf{Y}(k+1)\right) \xrightarrow[\text{Apply observations performed at time k+2}]{\text{update}} p_{\mathbf{x}(k+2)|\mathbf{Y}(k+2)}\left(\mathbf{x}(k+2)\,|\,\mathbf{Y}(k+2)\right)$$

Now, the involved PDFs are Gaussian

Prediction step

We have similar operations in our prediction steps

$$\mathbf{x}(k+1)=\mathbf{f}\left(\mathbf{x}(k),\mathbf{u}(k)\right)+\boldsymbol{\xi}(k)$$

In which x(k+1) is generated by transforming x(k) and then adding an independent RV $\boldsymbol{\xi}(k)$

(→it turned to be that the prediction step is easily implemented for the Gaussian cases)

In this step, we have $\mathbf{u}(k),\mathbf{P}_{\xi(k)}$ and $\left(\hat{\mathbf{x}}(k),\mathbf{P}_{\mathbf{x}(k)}\right)$

and we obtain $\hat{\mathbf{x}}(k+1),\mathbf{P}_{\mathbf{x}(k+1)}$

(we know the PDF of x(k), and the prediction model, we also know the PDF of the noise/uncertainty of the model

- → We obtain the PDF of X(k+1)
- → Because the "input" PDFs are Gaussian and **f()** is linear or "smooth enough", the resulting PDF is also a Gaussian PDF, whose parameters we obtain easily.

Consider the case of a LINEAR process model

$$\mathbf{x}(k+1)=\mathbf{A}\cdot\mathbf{x}(k)+\mathbf{B}\cdot\mathbf{u}(k)+\boldsymbol{\xi}(k)$$

In which x(k+1) is generated by transforming x(k) and then adding an independent RV $\boldsymbol{\xi}(k)$

The model parameters {A,B }and the inputs u(k) are well known, and the noise is additive WGN whose covariance is known. WGN has, by definition, zero mean. We also have that the PDF about X(k) is Gaussian , whose expected value and covariance matrix are known.

$$\mathbf{u}(k),\mathbf{P}_{\xi(k)} \text{ and } \left(\hat{\mathbf{x}}(k),\mathbf{P}_{\mathbf{x}(k)}\right)$$

In this step, we obtain:

$$\hat{\mathbf{x}}(k+1)=\mathbf{A}\cdot\hat{\mathbf{x}}(k)+\mathbf{B}\cdot\mathbf{u}(k)+0$$

$$\mathbf{P}_{\mathbf{x}(k+1)}=\mathbf{A}\cdot\mathbf{P}_{\mathbf{x}(k)}\cdot\mathbf{A}^{T}+\mathbf{P}_{\xi(k)}$$

This is also valid for linear time varying models, as well.

$$\hat{\mathbf{x}}(k+1)=\mathbf{A}(k)\cdot\hat{\mathbf{x}}(k)+\mathbf{B}(k)\cdot\mathbf{u}(k)+0$$

$$\mathbf{P}_{\mathbf{x}(k+1)}=\mathbf{A}(k)\cdot\mathbf{P}_{\mathbf{x}(k)}\cdot\mathbf{A}(k)^{T}+\mathbf{P}_{\xi(k)}$$

$$\left(\begin{array}{l}\text{Noise covariance matrix } \mathbf{P}_{\xi(k)} \text{ is usually called } \mathbf{Q} \text{ or } \mathbf{Q}(k) \text{ in the literature,}\\ \text{we will use that name later}\end{array}\right)$$

Update step

“Receiving information”

If at time k, we perform an observation (we measure a system output), then we can define a Likelihood function and obtain a “posterior” PDF. (via a Bayesian Update)

We have the prior PDF, described by $\left(\hat{\mathbf{x}}(k), \mathbf{P}_{\mathbf{x}(k)}\right)$

Then after the update we would obtain the “updated“ parameters $\left(\hat{\mathbf{x}}(k)^{(+)}, \mathbf{P}_{\mathbf{x}(k)}^{(+)}\right)$

(they describe the POSTERIOR PDF)

Both PDFs talk about the same variable, x(k), however the “updated” version used more information that the prior one.

How do we obtain these updated parameter? $\left(\hat{\mathbf{x}}(k)^{(+)}, \mathbf{P}_{\mathbf{x}(k)}^{(+)}\right)$

If the observation is based on a measurement of an output of the system, and the output equation is linear:

$$\mathbf{y}(k) = \mathbf{H} \cdot \mathbf{x}(k) + \boldsymbol{\eta}(k)$$

In which **H** is known, and where the additive noise $\boldsymbol{\eta}(k)$ represents the noise in the measurements, and other errors introduced by our nominal output model.

In addition, if we know that $\boldsymbol{\eta}(k)$ behaves as WGN, having zero mean ($\hat{\boldsymbol{\eta}}(k) = \mathbf{0}$) and covariance matrix

$$\mathbf{P}_{\boldsymbol{\eta}(k)} \qquad (\text{usually called } \mathbf{R}(k)\text{, in the literature})$$

Then, we may obtain the expected value and the covariance matrix of the posterior PDF in this way:

Update

$$\text{We have } \left\{\hat{\mathbf{x}}(k)^{(-)}, \mathbf{P}(k)^{(-)}\right\} \quad \text{PRIOR } \{\text{expected value and , covariance matrix}\}$$

$$\text{We want } \left\{\hat{\mathbf{x}}(k)^{(+)}, \mathbf{P}(k)^{(+)}\right\} \ \text{POSTERIOR } \{\text{expected value and , covariance matrix}\}$$

$$\Downarrow$$

$$\mathbf{z}(k) = \mathbf{y}_{measurement}(k) - \mathbf{H} \cdot \hat{\mathbf{x}}(k)^{(-)}$$

$$\mathbf{S} = \mathbf{H} \cdot \mathbf{P}(k)^{(-)} \cdot \mathbf{H}^{T} + \mathbf{R}(k)$$

$$\mathbf{K}(k) = \mathbf{P}(k)^{(-)} \cdot \mathbf{H}^{T} \cdot \mathbf{S}^{-1}$$

$$\hat{\mathbf{x}}(k)^{(+)} = \hat{\mathbf{x}}(k)^{(-)} + \mathbf{K}(k) \cdot \mathbf{z}(k)$$

$$\mathbf{P}(k)^{(+)} = \mathbf{P}(k)^{(-)} - \mathbf{P}(k)^{(-)} \cdot \mathbf{H}^{T} \cdot \mathbf{S}^{-1} \cdot \mathbf{H} \cdot \mathbf{P}(k)^{(-)}$$

We will always talk about "expected values" and "covariance matrices"
(which implicitly represent their associated Gaussian PDFs)

For a notation consistent with the literature , let's change our notation:

$$\text{PRIOR:} \quad \left\{\hat{\mathbf{x}}(k)^{(-)},\mathbf{P}(k)^{(-)}\right\} \xrightarrow{\text{we use notation}} \left\{\ \hat{\mathbf{x}}(k\,|\,k-1)\ \ ,\ \mathbf{P}(k\,|\,k-1)\ \right\}$$

$$\text{POSTERIOR:} \quad \left\{\hat{\mathbf{x}}(k)^{(+)},\mathbf{P}(k)^{(+)}\right\} \xrightarrow{\text{we use notation}} \left\{\ \hat{\mathbf{x}}(k\,|\,k)\ ,\ \mathbf{P}(k\,|\,k)\ \right\}$$

$$\hat{\mathbf{x}}(k\,|\,k-1),\mathbf{P}(k\,|\,k-1) \quad \text{parameters of PDF} \quad p_{\mathbf{x}(k)|\mathbf{Y}(k-1)}\left(\mathbf{x}(k)\,|\,\mathbf{Y}(k-1)\right)$$

$$\hat{\mathbf{x}}(k\,|\,k),\mathbf{P}(k\,|\,k) \quad \text{parameters of PDF} \quad p_{\mathbf{x}(k)|\mathbf{Y}(k)}\left(\mathbf{x}(k)\,|\,\mathbf{Y}(k)\right)$$

$$\left\{\ \hat{\mathbf{x}}(k\,|\,k-1)\ \ ,\ \mathbf{P}(k\,|\,k-1)\ \right\}: \textit{prior}$$
$$\left\{\ \hat{\mathbf{x}}(k\,|\,k)\ ,\ \mathbf{P}(k\,|\,k)\ \right\}: \textit{posterior}$$

All our calculations will be numerical, always using expected values and covariances matrixes.

→ No need to manipulate PDFs!

Now, our sequence of Bayesian prediction and update steps

$$\xrightarrow[\text{prediction}]{\substack{\mathbf{x}(k)= \\ F(\mathbf{x}(k-1),\mathbf{u}(k-1))}} \left\{ \hat{\mathbf{x}}(k|k-1)\ ,\ \mathbf{P}(k|k-1) \right\} \xrightarrow[\substack{\text{Apply} \\ \text{observations} \\ \text{performed at time k}}]{\text{update}} \left\{ \hat{\mathbf{x}}(k|k)\ ,\ \mathbf{P}(k|k) \right\}$$

$$\xrightarrow[\text{prediction}]{\substack{\mathbf{x}(k+1)= \\ F(\mathbf{x}(k),\mathbf{u}(k))}} \left\{ \hat{\mathbf{x}}(k+1|k)\ ,\ \mathbf{P}(k+1|k) \right\} \xrightarrow[\substack{\text{Apply} \\ \text{observations} \\ \text{performed at time k+1}}]{\text{update}} \left\{ \hat{\mathbf{x}}(k+1|k+1)\ ,\ \mathbf{P}(k+1|k+1) \right\}$$

$$\xrightarrow[\text{prediction}]{\substack{\mathbf{x}(k+2)= \\ F(\mathbf{x}(k+1),\mathbf{u}(k+1))}} \left\{ \hat{\mathbf{x}}(k+2|k+1)\ ,\ \mathbf{P}(k+2|k+1) \right\} \xrightarrow[\substack{\text{Apply} \\ \text{observations} \\ \text{performed at time k+2}}]{\text{update}} \left\{ \hat{\mathbf{x}}(k+2|k+2)\ ,\ \mathbf{P}(k+2|k+2) \right\}$$

sequence continues...

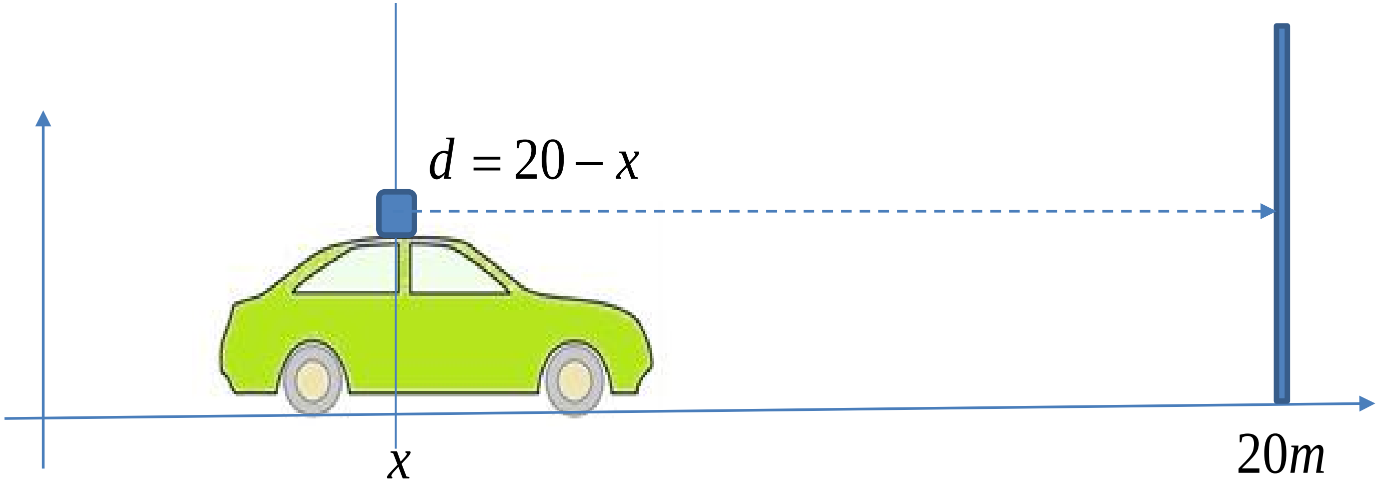
prediction and update steps

$$\begin{tikzcd}
.. \left\{\begin{matrix}\hat{\mathbf{X}}(i\,|\,i-1)\\ \mathbf{P}(i\,|\,i-1)\end{matrix}\right\} \arrow[r, "\mathbf{y}(i)=h(\mathbf{x}(i))", "update"'] & \left\{\begin{matrix}\hat{\mathbf{X}}(i\,|\,i)\\ \mathbf{P}(i\,|\,i)\end{matrix}\right\} \arrow[r, "\mathbf{x}(i+1)=F(\mathbf{x}(i)\text{,}\mathbf{u}(i))", "prediction"'] & \left\{\begin{matrix}\hat{\mathbf{X}}(i+1\,|\,i)\\ \mathbf{P}(i+1\,|\,i)\end{matrix}\right\} \arrow[r, "\mathbf{y}(i+1)=h(\mathbf{x}(i+1))", "update"'] & \left\{\begin{matrix}\hat{\mathbf{X}}(i+1\,|\,i+1)\\ \mathbf{P}(i+1\,|\,i+1)\end{matrix}\right\} \arrow[r] & ..
\end{tikzcd}$$

We never stop, even if sometimes there are no observations (no update)

$$\begin{tikzcd}
.. \left\{\begin{matrix}\hat{\mathbf{X}}(i\,|\,i-1)\\ \mathbf{P}(i\,|\,i-1)\end{matrix}\right\} \arrow[r, "\text{no available observations}", "\text{no update}"'] & \left(\begin{matrix}\text{no}\\ \text{update}\\ \\ \{\hat{\mathbf{X}},\mathbf{P}\}\ \text{unchanged}\end{matrix}\right) \arrow[r, "\mathbf{x}(i+1)=F(\mathbf{x}(i)\text{,}\mathbf{u}(i))", "prediction"'] & \left\{\begin{matrix}\hat{\mathbf{X}}(i+1\,|\,i)\\ \mathbf{P}(i+1\,|\,i)\end{matrix}\right\} \arrow[r, "\mathbf{y}(i+1)=h(\mathbf{x}(i+1))", "update"'] & \left\{\begin{matrix}\hat{\mathbf{X}}(i+1\,|\,i+1)\\ \mathbf{P}(i+1\,|\,i+1)\end{matrix}\right\} \arrow[r] & ..
\end{tikzcd}$$

Example: The 1D car.

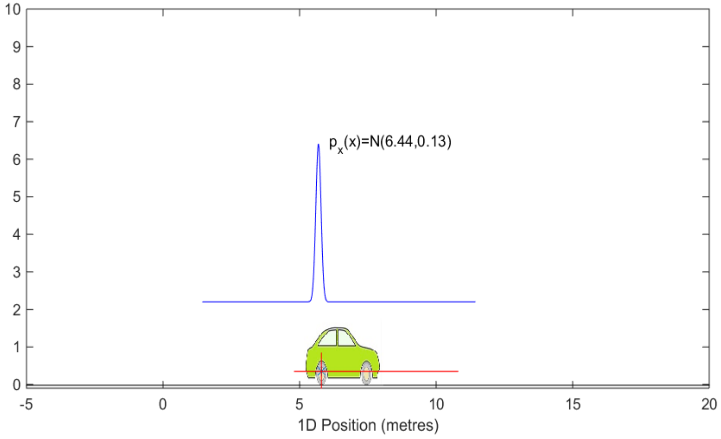


Consider this case: 1D localization of a vehicle.

Its initial 1D position is x=3m, , which we know with an uncertainty of 0.05m standard deviation.

Our model is based on speed, from sensor's measurements, which are polluted by WGN , having standard deviation of 0.15m/s.

The sampling time is $\tau =$**10ms**.



$$x(k+1) \cong x(k) + \tau \cdot v\ (k)$$

$$v_{sensor}(k) = v(k) + \xi_v(k)$$

$$\xi_v(k) \sim N\left(0, \sigma^2_{\xi_v}\right)$$

$$\sigma_{\xi_v} = 0.15$$

$$\sigma^2_{x(0)} = 0.05^2$$

$$\hat{x}(0) = 3$$

$$x(k+1) = x(k) + \tau \cdot v_{sensor}(k) - \tau \cdot \xi_v(k) =$$
$$= x(k) + \tau \cdot v_{sensor}(k) + \xi(k)$$

$$\left(\xi(k) = -\tau \cdot \xi_v(k)\right)$$

$$\mathbf{Q} = \tau \cdot \sigma^2_{\xi_v} \cdot \tau = (0.15)^2 \cdot 0.01^2$$

$$\Downarrow$$

Our prediction

$$\hat{\mathbf{x}}(k+1|k) = \hat{\mathbf{x}}(k|k) + \tau \cdot v_{sensor}(k)$$
$$\mathbf{P}(k+1|k) = 1 \cdot \mathbf{P}(k|k) \cdot 1 + \mathbf{Q}$$

(In this example the meaning of $\mathbf{P}$ is $\sigma^2_{\mathbf{x}}$)

Observation:

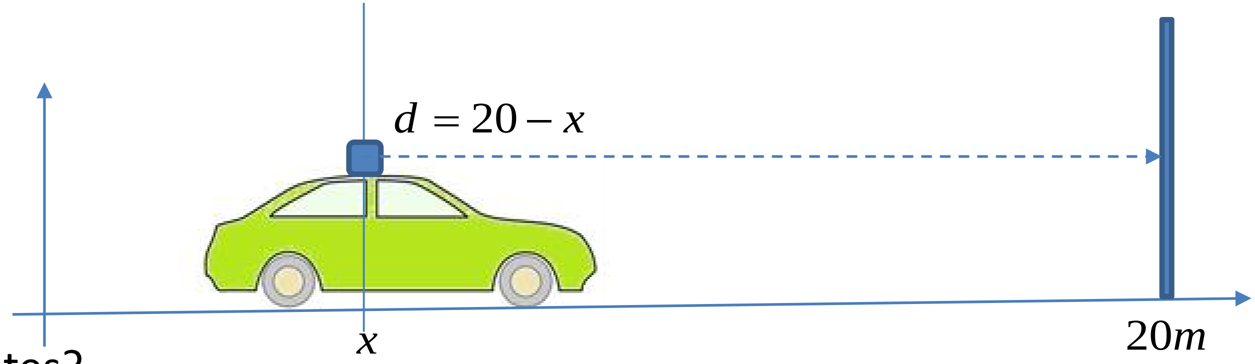
Suppose we use a 1D Lidar to measure the distance to a pole, which is located at position 20m (in this 1D universe)

The error in the measurements behaves as WGN, having standard deviation 1cm.

$$d = 20 - x \quad \text{(variable } d \text{ is function of the state!)}$$

$$d_{measured}(k) = 20 - x(k) + \eta(k)$$

$$\eta(k) \sim N\left(0\ ,\ 0.01^2\right)$$



Can we implement updates?

Observation:

$$d = d(x) = 20 - x \quad \text{(it is function of the state!)}$$

$$d_{measured}(k) = 20 - x(k) + \eta(k)$$

$$\eta(k) \sim N\left(0\ ,\ 0.01^2\right)$$

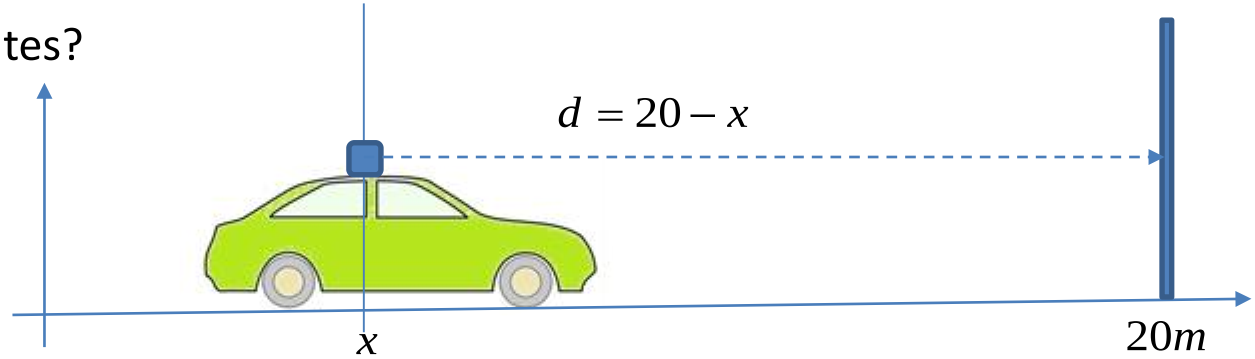
$$\Downarrow$$

$$\mathbf{H} = -1$$

$$\mathbf{R} = 0.01^2$$

Can we implement updates?

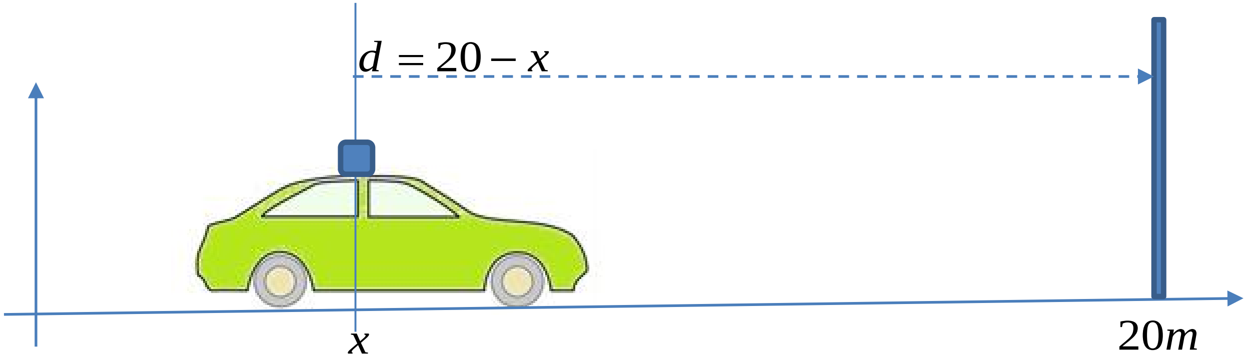
YES



Full sequence of prediction and update steps.

Firstly, we need to initialize our belief about the state, $\hat{x}(0|0)=3, \quad \mathbf{P}(0|0)=0.05^2$

```
t=t0;
While running
{          "do sleep" during $\tau$
           wake up! and predict  one step $\tau$   ("dt" since last  prediction.
           Any available observation?  (i.e. measurement from LiDAR)
                                              if yes: → perform update
}
```



Full sequence of prediction and update steps.

initial, at k=0: $\hat{x}(0|0) = 3, \quad \mathbf{P}(0|0) = 0.05^2$

k=1 : do prediction (use proces model, inputs values) $\{\hat{x}(1|0), \quad \mathbf{P}(1|0)\}$

k=1 : if observations are available $\Rightarrow$ perform update : $\{\hat{x}(1|1), \quad \mathbf{P}(1|1)\}$

if not keep current parameters

go to sleep for τ

wake up!

k=2 : do prediction $\{\hat{x}(2|1), \quad \mathbf{P}(2|1)\}$

k=2 : if observations are available $\Rightarrow$ perform update : $\{\hat{x}(2|2), \quad \mathbf{P}(2|2)\}$

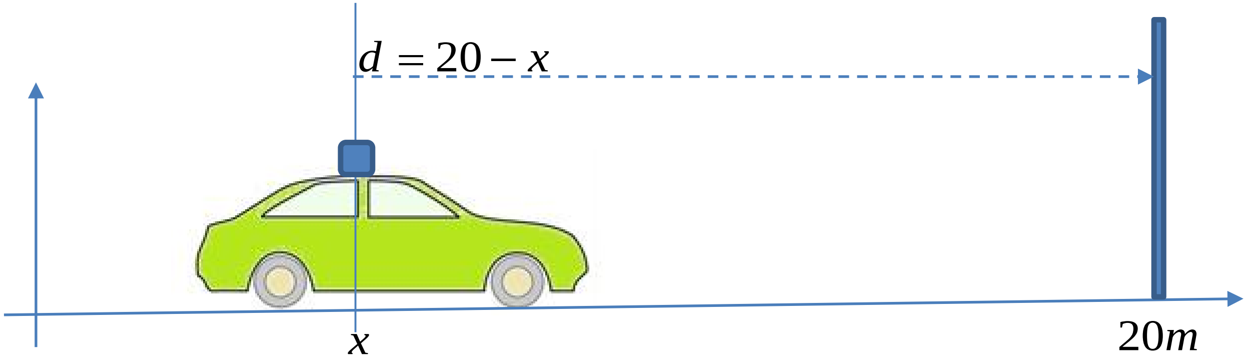
if not keep current parameters

go to sleep for τ

Example: Prediction and update steps (simulation)

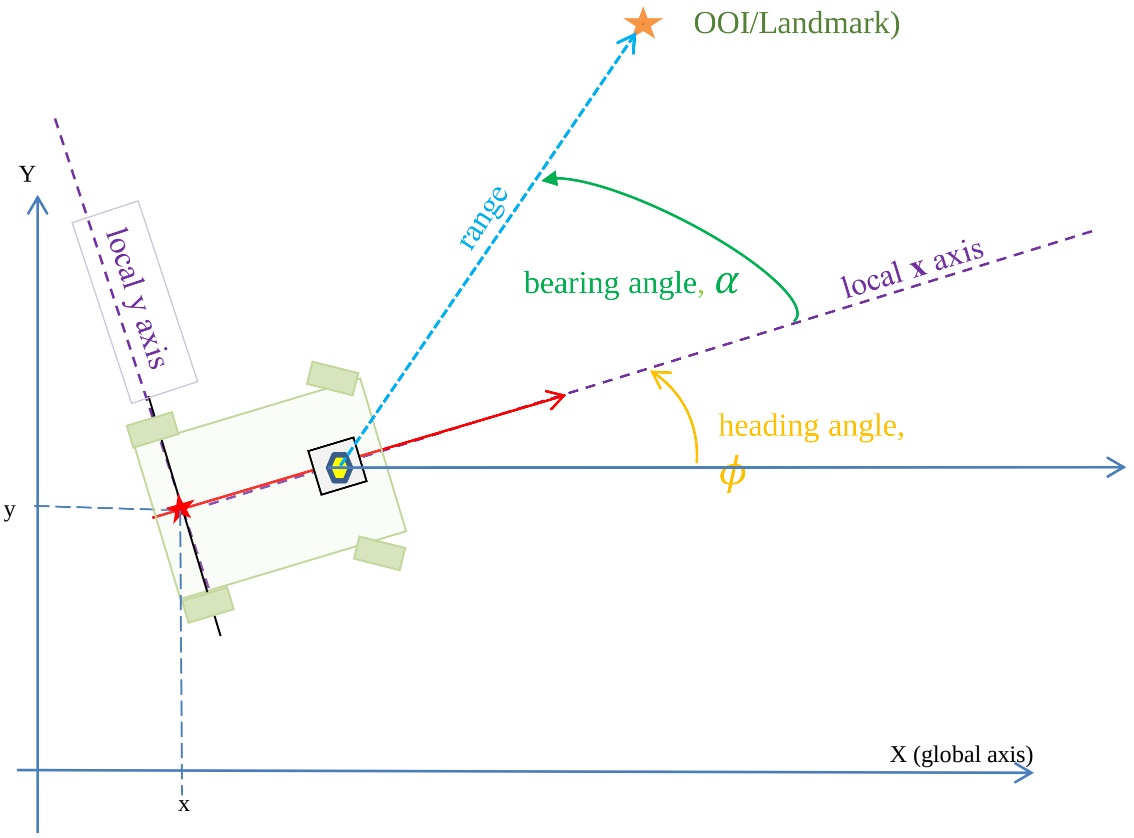
In this simulation we simulate measurements from the noisy sensors

We see the code in class, and we see its results.



(Related tutorial problems to be discussed now.)

How would it be, for estimating our 2D pose?



A bit more difficult, but still easy enough.

We will see it in class.

→ Defining our process model, our observation model, etc. Inspecting program code, visualizing results.

(then we end here)