Perceção e Controlo

Mest. em Robótica e Sistemas Inteligentes

Ano letivo 2024/2025

Nuno Lau (nunolau@ua.pt)

Sensor Fusion

- **Act of combining sensory data** or data derived from sensory data from disparate sources
- **The resulting information is “better“** than it would be possible when the sources were used individually
- **“Better”** is defined according to the context. Can be **more accurate**, **more complete**, **a different view**, etc.
- Using a broader definition, we can speak of **Information Fusion**
- Sensor Fusion is usually considered a subset of information fusion, although the terms are often used with the same meaning
- Another very used term for this task is Multi-Sensor Data Fusion

Bayes Rule

- Determine the probability of a state/event, given the result of other states/events that are related.

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)}$$

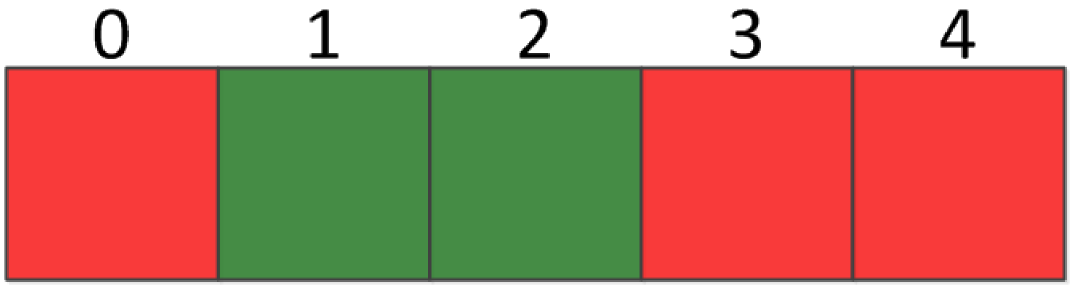
- P(A | B) is the probability of A given that B is true, called **conditional probability**
- P(B | A) is the probability of B given that A is true, called conditional probability
- P(A) and P(B) are the **marginal probabilities** of A and B

Grid World – Measurement Integration

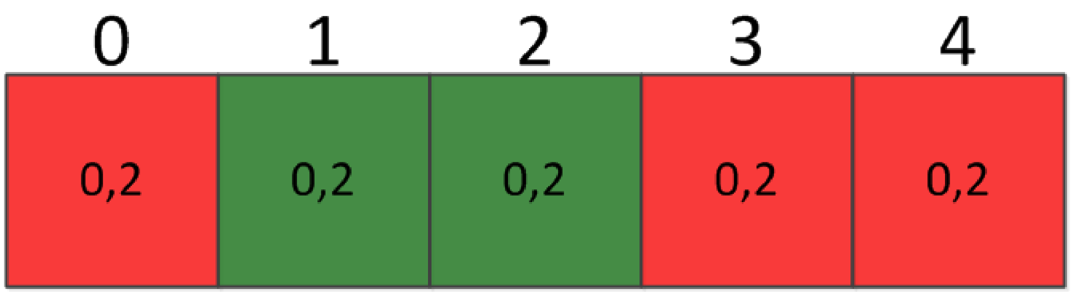
- Bayes' Rule
- $P(Xp|M) = \frac{P(M|Xp)*P(Xp)}{P(M)}$
- $P(X_p)$ is the estimate before measurement integ.
- $P(X_p|M)$ is the posterior estimate
- $P(M)$ does not depend on X_p, so it can be considered as a constant that performs normalization
- **Measurement integration** is performed using the **product** of sensor model and previous estimate

Grid World

- Consider a grid world:



- Position probability distribution:
-



Grid World

- The robot has a color sensor
- Sensor model

  G_M: green measure; R_M: red measure

  G_C: green cell; R_C: red cell

  $P(G_M|G_C)=0.6$; $P(R_M|G_C)=0.4$

  $P(G_M|R_C)=0.2$; $P(R_M|R_C)=0.8$

- The robot measures green
- Which is the new position estimate?

Grid World

- Multiply cell values by $P(G_M|G_C)$ and $P(G_M|R_C)$
-

0	1	2	3	4
0,04	0,12	0,12	0,04	0,04

$$\sum P = 0{,}36$$

- Normalize

0	1	2	3	4
0,11	0,33	0,33	0,11	0,11

- **We have just applied Bayes' Rule!**

Grid World - Moving

- $P(1) = P(1 \mid A_R, 0) * P(0) + P(1 \mid A_R, 1) * P(1)$
- Predicted belief

0	1	2	3	4
0,1	0,5	0,4	0,0	0,0

- When the **robot moves** belief is updated through **convolution**

Bayes Filters: Framework

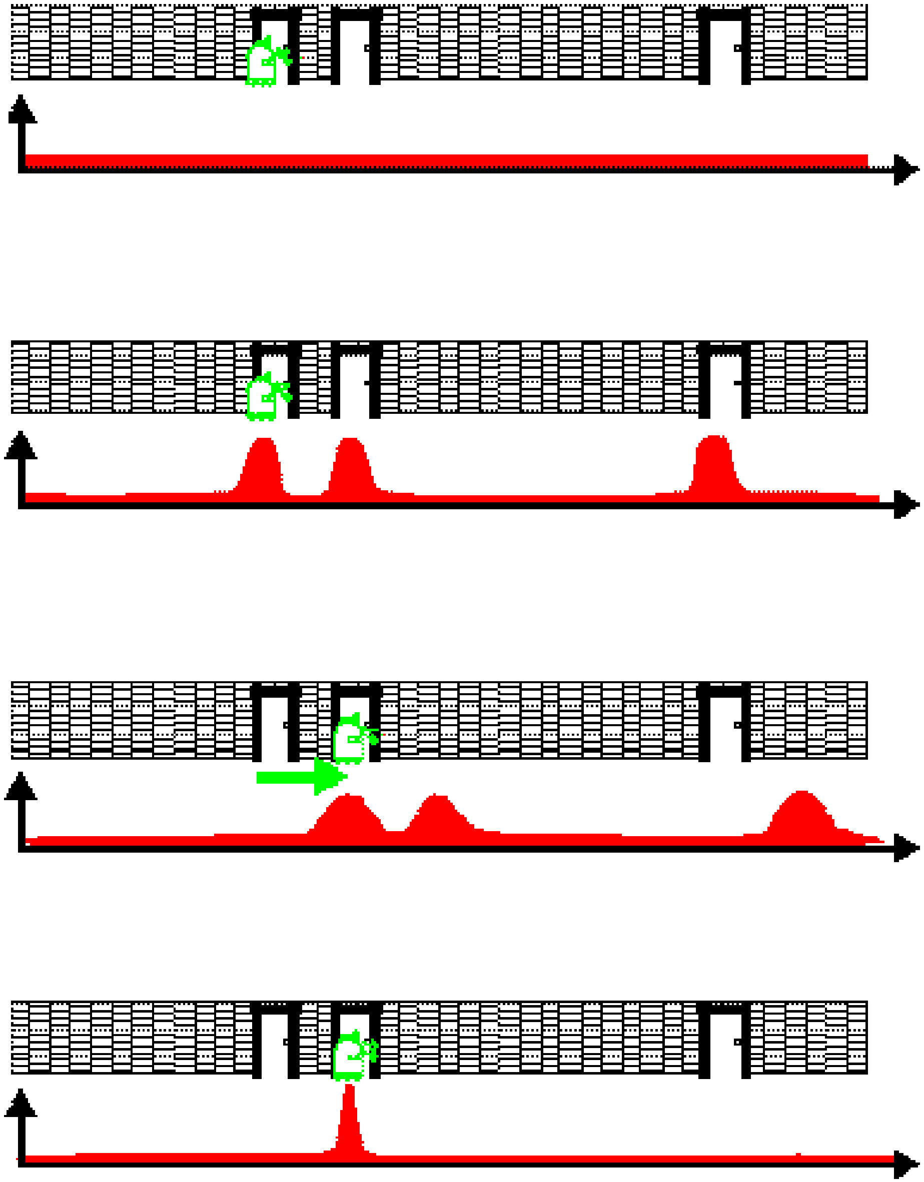
- **Given:**
  - Stream of observations *z* and action data *u:*

$$d_t = \{u_1, z_1 \ldots, u_t, z_t\}$$

  - Sensor model *P(z|x).*
  - Action model *P(x|u,x')*.
  - Prior probability of the system state *P(x).*
- **Wanted:**
  - Estimate of the state *X* of a dynamical system.
  - The posterior of the state is also called **Belief**:

$$Bel(x_t) = P(x_t \mid u_1, z_1 \ldots, u_t, z_t)$$

Bayes Filter Example



Bayes Filter

- Continuous environments
  - Measurement Integration (Product)

$$bel(X)_t = \eta \; p(Z_t|X) \; \hat{bel}(X)_t$$

  - Motion update (Convolution)

$$\hat{bel}(X)_t = \int p(X|U_t, X')bel(X')_{t-1}dX'$$

Bayes Filter Algorithm

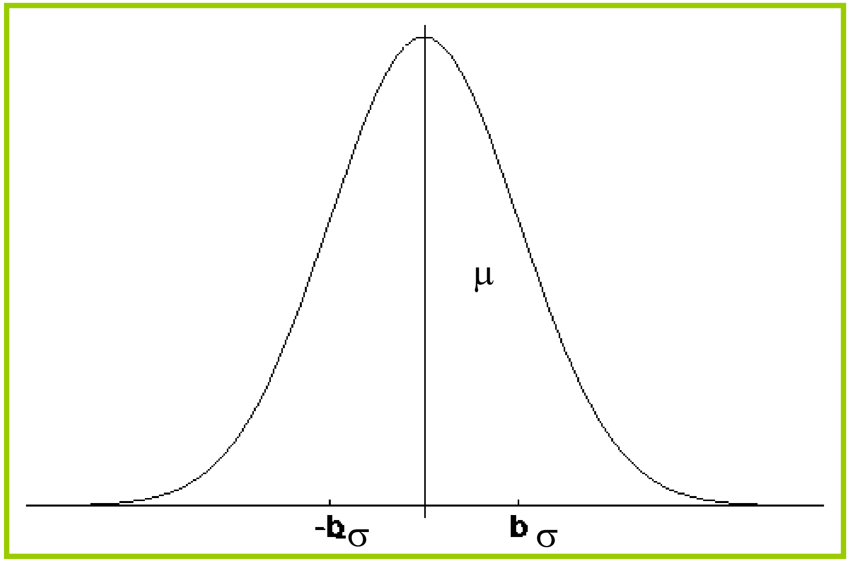
1. Algorithm **Bayes_filter**(*Bel(x),d*):
2. $\eta=0$
3. If *d* is a perceptual data item *z* then
4. For all *x* do
5. $$Bel'(x) = P(z \mid x) Bel(x)$$
6. $$\eta = \eta + Bel'(x)$$
7. For all *x* do
8. $$Bel'(x) = \eta^{-1} Bel'(x)$$
9. Else if *d* is an action data item *u* then
10. For all *x* do
11. $$Bel'(x) = \int P(x \mid u, x')\, Bel(x')\, dx'$$
12. Return *Bel'(x)*

Gaussians

$$p(x) \sim N(\mu, \sigma^2):$$

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

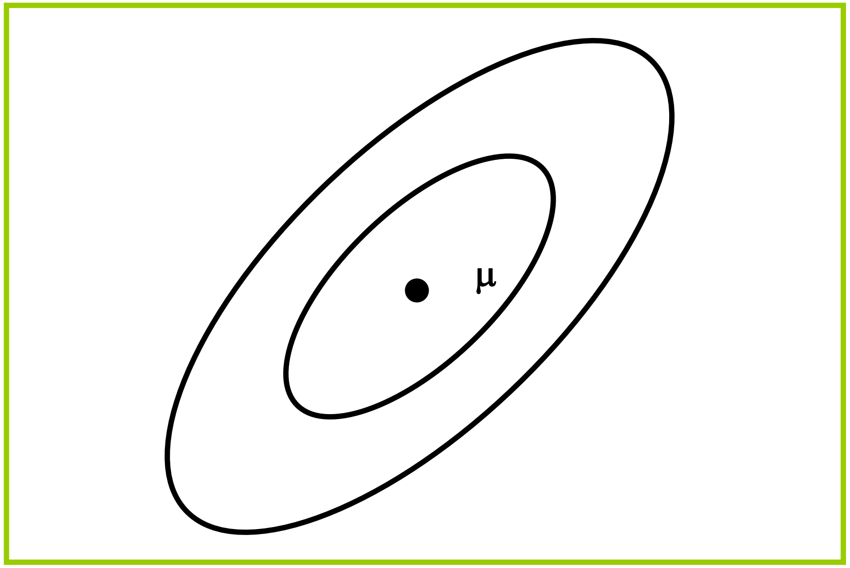
Univariate



$$p(\mathbf{x}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}):$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

Multivariate



Properties of Gaussians

$$\left.\begin{array}{l} X \sim N(\mu,\sigma^2) \\ Y = aX + b \end{array}\right\} \Rightarrow \quad Y \sim N(a\mu + b, a^2\sigma^2)$$

$$\left.\begin{array}{l} X_1 \sim N(\mu_1,\sigma_1^{\,2}) \\ X_2 \sim N(\mu_2,\sigma_2^{\,2}) \end{array}\right\} \Rightarrow p(X_1)\cdot p(X_2) \sim N\left(\frac{\sigma_2^{\,2}}{\sigma_1^{\,2}+\sigma_2^{\,2}}\mu_1 + \frac{\sigma_1^{\,2}}{\sigma_1^{\,2}+\sigma_2^{\,2}}\mu_2, \quad \frac{1}{\sigma_1^{\,-2}+\sigma_2^{\,-2}}\right)$$

Kalman Filter

- Assumptions
  - **Linear model** (transition, action and sensor)
  - **Every estimate is a gaussian**
    - Can be characterized by **mean** and **variance**
  - **Noise is gaussian** (mean=0)
- Integration of measures over time
- **Markovian assumption**
  - The next estimate only depends on the previous estimate
- Considers **action model** and **physics/observation model**

Discrete Kalman Filter

- Estimates the state x of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$\boxed{x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t}$$

- with a measurement

$$\boxed{z_t = C_t x_t + \delta_t}$$

Kalman Filter

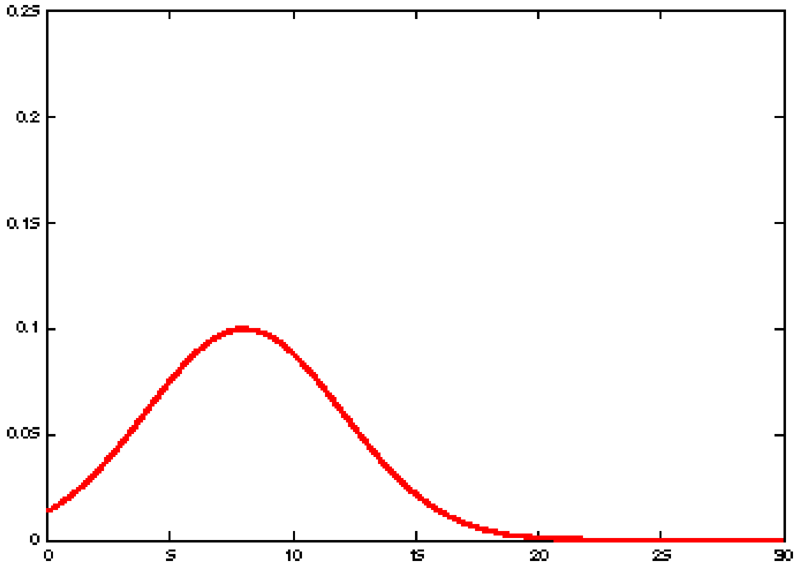
- **Product** of gaussians distributions (**measurement integration**)

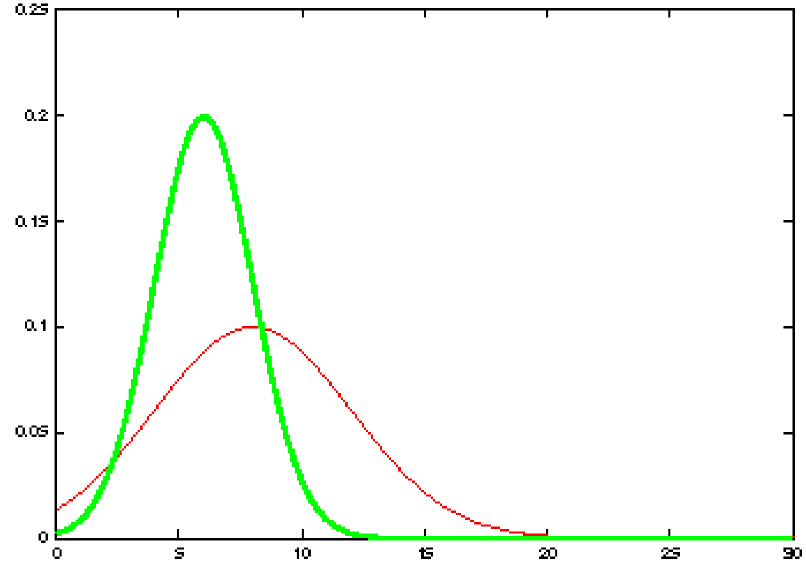
$$\mu_P = \frac{\mu_1 . {\sigma_2}^2 + \mu_2 . {\sigma_1}^2}{{\sigma_1}^2 + {\sigma_2}^2} \qquad {\sigma_P}^2 = \frac{{\sigma_1}^2 . {\sigma_2}^2}{{\sigma_1}^2 + {\sigma_2}^2}$$

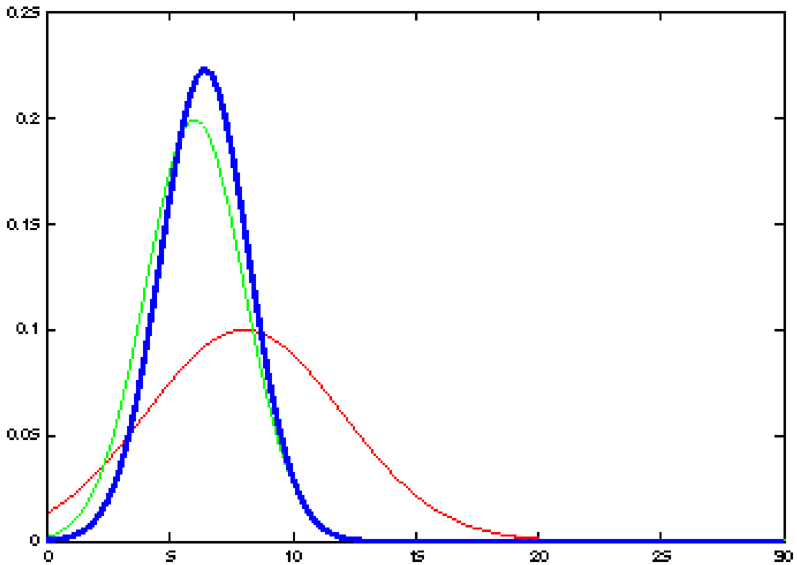
- **Convolution** of gaussians distributions (**motion forecast**)

$$\mu_C = \mu_1 + \mu_2 \qquad {\sigma_C}^2 = {\sigma_1}^2 + {\sigma_2}^2$$

Kalman Filter Updates in 1D





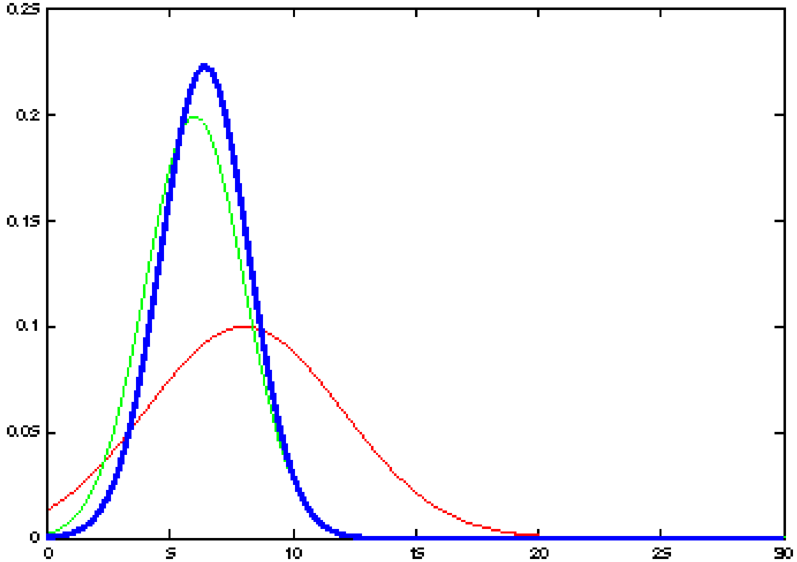


Kalman Filter Updates in 1D

- Measurement integration

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - \bar{\mu}_t) \\ \sigma_t^2 = (1 - K_t)\bar{\sigma}_t^2 \end{cases} \quad \text{with} \quad K_t = \frac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + \bar{\sigma}_{obs,t}^2}$$

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - C_t\bar{\mu}_t) \\ \Sigma_t = (I - K_t C_t)\bar{\Sigma}_t \end{cases} \quad \text{with} \quad K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$

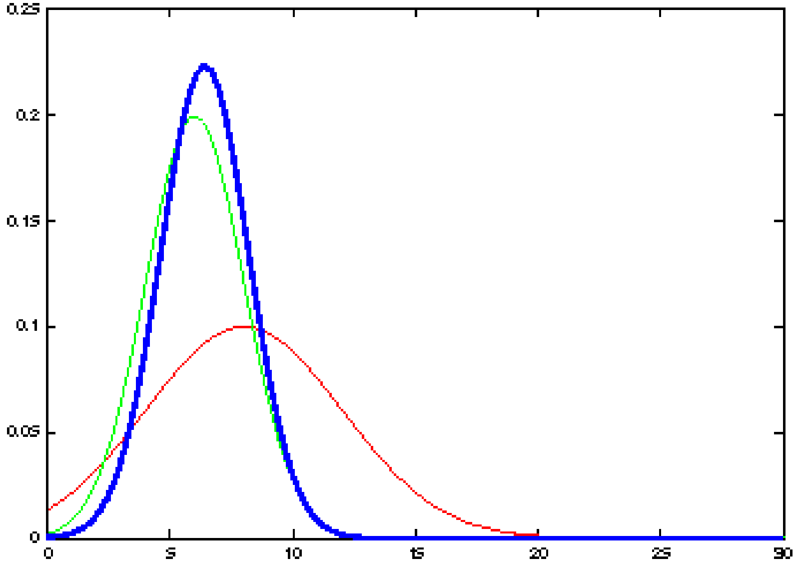


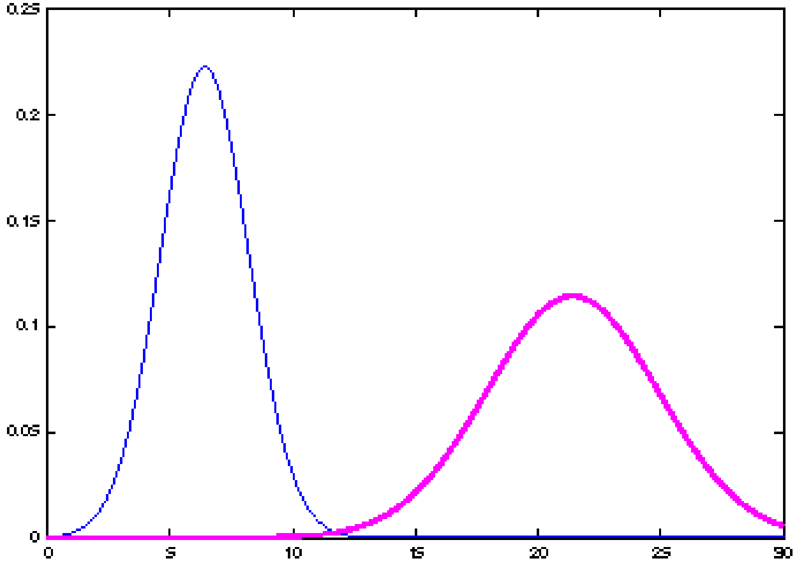
Kalman Filter Updates in 1D

- Motion

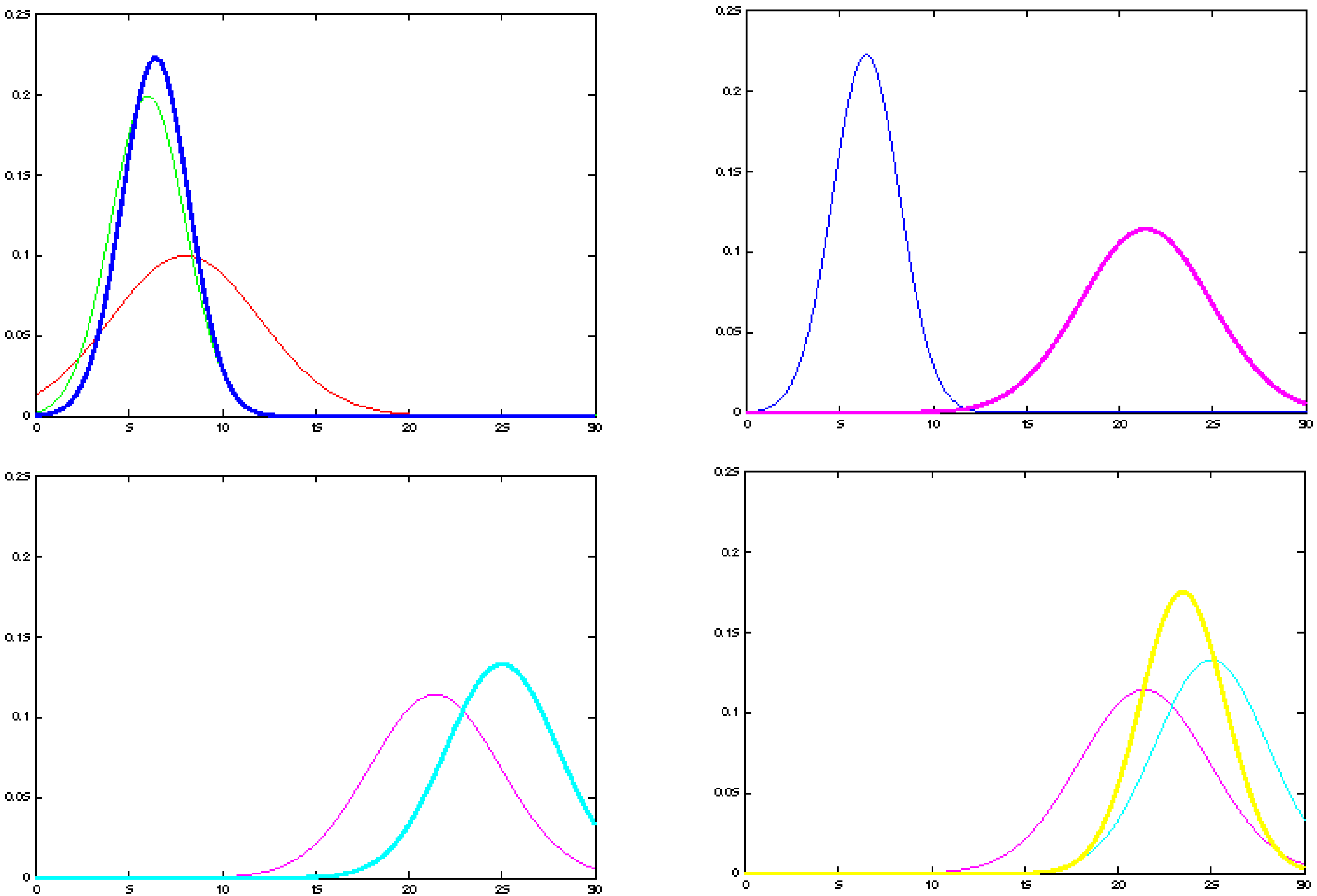
$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = a_t \mu_{t-1} + b_t u_t \\ \bar{\sigma}_t^2 = a_t^2 \sigma_t^2 + \sigma_{act,t}^2 \end{cases}$$

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$





Kalman Filter Updates



Linear Gaussian Systems: Initialization

- Initial belief is normally distributed:

$$\boxed{bel(x_0) = N\left(x_0; \mu_0, \Sigma_0\right)}$$

Linear Gaussian Systems: Dynamics

- Dynamics are linear function of state and control plus additive noise:

$$\boxed{x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t}$$

$$\boxed{p(x_t \mid u_t, x_{t-1}) = N\left(x_t; A_t x_{t-1} + B_t u_t, R_t\right)}$$

$$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) \qquad bel(x_{t-1})\, dx_{t-1}$$

$$\Downarrow \qquad\qquad \Downarrow$$

$$\sim N\left(x_t; A_t x_{t-1} + B_t u_t, R_t\right) \quad \sim N\left(x_{t-1}; \mu_{t-1}, \Sigma_{t-1}\right)$$

Linear Gaussian Systems: Dynamics

$$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) \qquad bel(x_{t-1})\, dx_{t-1}$$

$$\Downarrow \qquad\qquad\qquad \Downarrow$$

$$\sim N\left(x_t; A_t x_{t-1} + B_t u_t, R_t\right) \quad \sim N\left(x_{t-1}; \mu_{t-1}, \Sigma_{t-1}\right)$$

$$\Downarrow$$

$$\overline{bel}(x_t) = \eta \int \exp\left\{-\frac{1}{2}(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1}(x_t - A_t x_{t-1} - B_t u_t)\right\}$$

$$\exp\left\{-\frac{1}{2}(x_{t-1} - \mu_{t-1})^T \Sigma_{t-1}^{-1}(x_{t-1} - \mu_{t-1})\right\} dx_{t-1}$$

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$

Linear Gaussian Systems: Observations

- Observations are linear function of state plus additive noise:

$$\boxed{z_t = C_t x_t + \delta_t}$$

$$\boxed{p(z_t \mid x_t) = N\left(z_t ; C_t x_t, Q_t\right)}$$

$$\boxed{\begin{array}{cccc} bel(x_t) = & \eta & p(z_t \mid x_t) & \overline{bel}(x_t) \\ & & \Downarrow & \Downarrow \\ & & \sim N\left(z_t ; C_t x_t, Q_t\right) & \sim N\left(x_t ; \overline{\mu}_t, \overline{\Sigma}_t\right) \end{array}}$$

Linear Gaussian Systems: Observations

$$bel(x_t) = \quad \eta \quad p(z_t \mid x_t) \qquad\qquad \overline{bel}(x_t)$$

$$\Downarrow \qquad\qquad\qquad \Downarrow$$

$$\sim N\left(z_t; C_t x_t, Q_t\right) \qquad \sim N\left(x_t; \bar{\mu}_t, \bar{\Sigma}_t\right)$$

$$\Downarrow$$

$$bel(x_t) = \eta \exp\left\{-\frac{1}{2}(z_t - C_t x_t)^T Q_t^{-1}(z_t - C_t x_t)\right\} \exp\left\{-\frac{1}{2}(x_t - \bar{\mu}_t)^T \bar{\Sigma}_t^{-1}(x_t - \bar{\mu}_t)\right\}$$

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t) \\ \Sigma_t = (I - K_t C_t)\bar{\Sigma}_t \end{cases} \qquad \text{with} \quad K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$

Kalman Filter – Motion Model

$$\hat{X}_t = F_t \hat{X}_{t-1} + B_t U_t + \omega_t$$

- $\hat{X}_t$ is the **estimated state**
- F_t is the **state transition model**
- B_t is the **control-input model**
- U_t is the **control vector**
- w_t is the **process noise** with covariance

$$Q_t : \omega_t \sim N(0, Q_t)$$

Kalman Filter – Observation Model

$$\hat{Z}_t = H_t X_t + \upsilon_t$$

- $\hat{z}_t$ is the **measurement** taken at time t
- H_t is the **observation model** of the state/event
- v_t is the **observation noise** with covariance:

$$R_t : \upsilon_t \sim N(0, R_t)$$

Kalman Filter – Implementation

- The filter state is represented by two variables:
  - X_t is the **estimate of the state** at time t
  - P_t is the **measure of estimated accuracy** of the process
- The filter works in **two steps**:

Forecast integration

$$\overline{X}_t = F_t X_{t-1} + B_t U_t$$

$$\overline{P}_t = F_t P_{t-1} F_t^T + Q_t$$

Measurement

$$K_t = \frac{\overline{P}_t H_t^T}{H_t \overline{P}_t H_t^T + R_t}$$

$$X_t = \overline{X}_t + K_t (Z_t - H_t \overline{X}_t)$$

$$P_t = (I - K_t H_t) \overline{P}_t$$

Kalman Filter Algorithm

1. Algorithm **Kalman_filter**($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):

2. Prediction:
3. $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
4. $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$

5. Correction:
6. $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
7. $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
8. $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$

9. Return μ_t, Σ_t