LEARNING DYNAMIC SYSTEMS : MARKOV MODELS

Markov Process and Markov Chains

Hidden Markov Models

Bayesian Filtering and Kalman Filter

Example of application: Navigation Systems and Data Fusion

Navigation System:
Position (L, l, z)
Angles (yaw, pitch, roll)
Speed
Angular speed

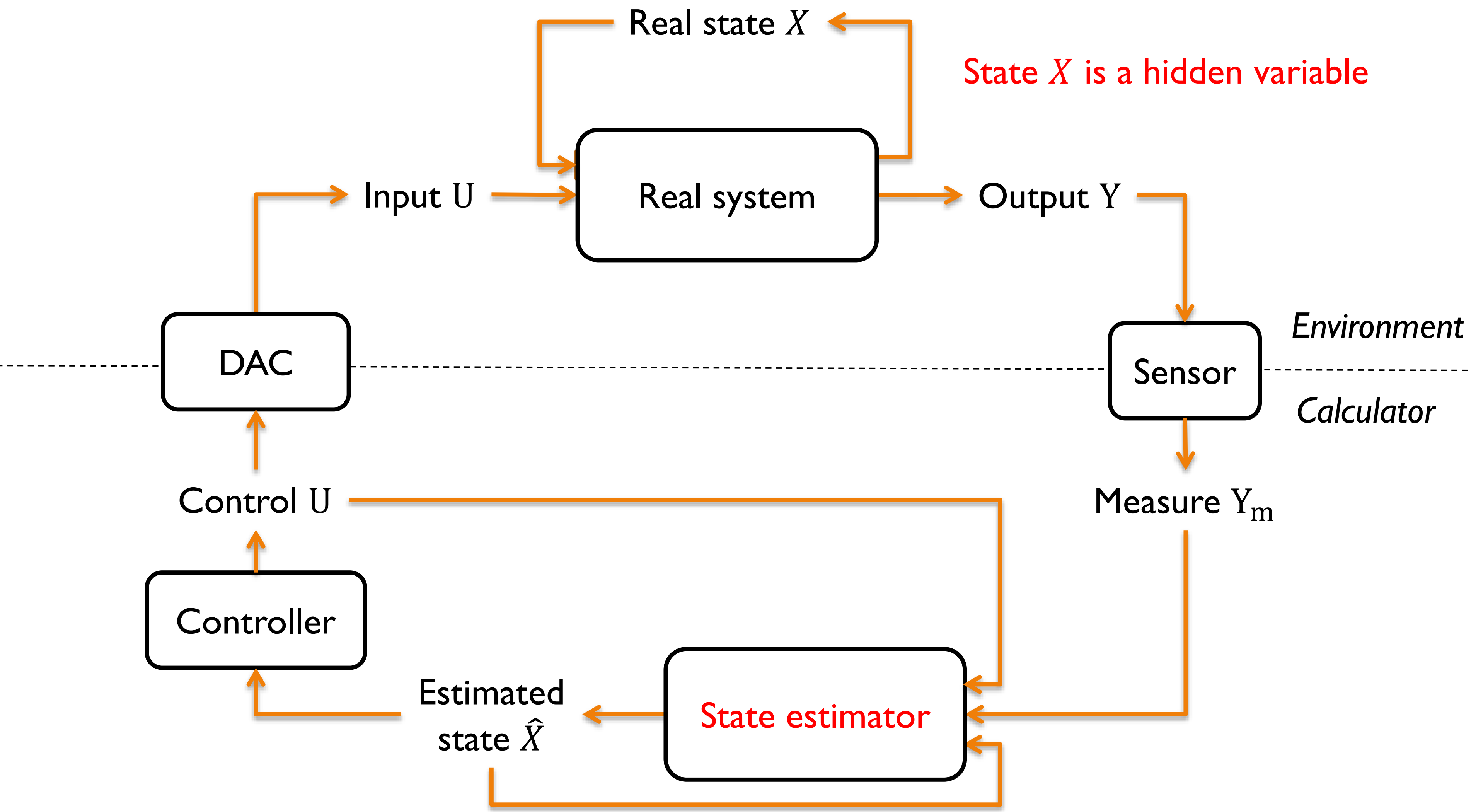
Altimeter:
☺ No drift
☹ Altitude only
☹ Not accurate

Inertial System:
☺ Responsive
☹ Long term drift

GPS:
☺ No drift
😐 Medium accuracy
☹ Not responsive

Tracking example using camshift & Kalman filtering

The whole picture:



Controllable Markov models

Update Markov models with a **command U_t**:

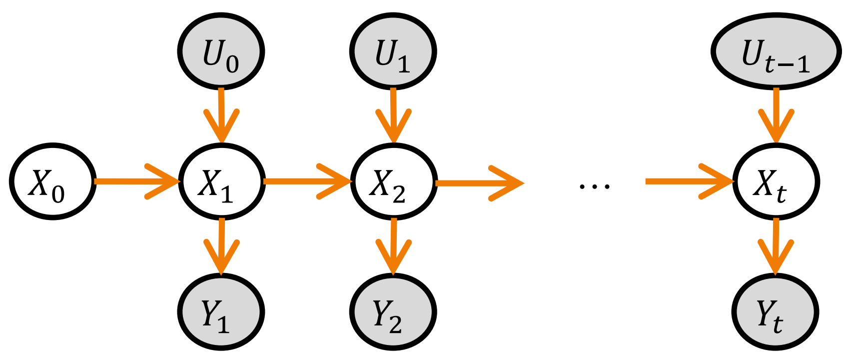
$$P(X_t|x_{t-1}, u_1, \ldots u_{t-1}, y_1, \ldots y_{t-1}) = P(X_t|x_{t-1}, u_{t-1})$$

and

$$P(Y_t|x_t, u_1, \ldots u_{t-1}, y_1, \ldots y_{t-1}) = P(Y_t|x_t)$$

$$P(X_2|x_1, u_0, y_1, u_1) = P(X_2|x_1, u_1)$$

$$P(Y_2|x_1, u_0, y_1, u_1) = P(X_2|x_1)$$



Bayesian Filtering:
State Space Representation

- **Continuous system** (f_t, g_t):

$$\begin{cases} \dfrac{dX_t}{dt} & = & f_t(X_t, U_t) \\ Y_t & = & g_t(X_t, U_t) \end{cases}$$

- **Discrete system** (f_t, g_t):

$$\begin{cases} X_{\color{red}{t+\tau}} & = & f_t(X_t, U_t) \\ Y_t & = & g_t(X_t, U_t) \end{cases}$$

- **Stationary** system:

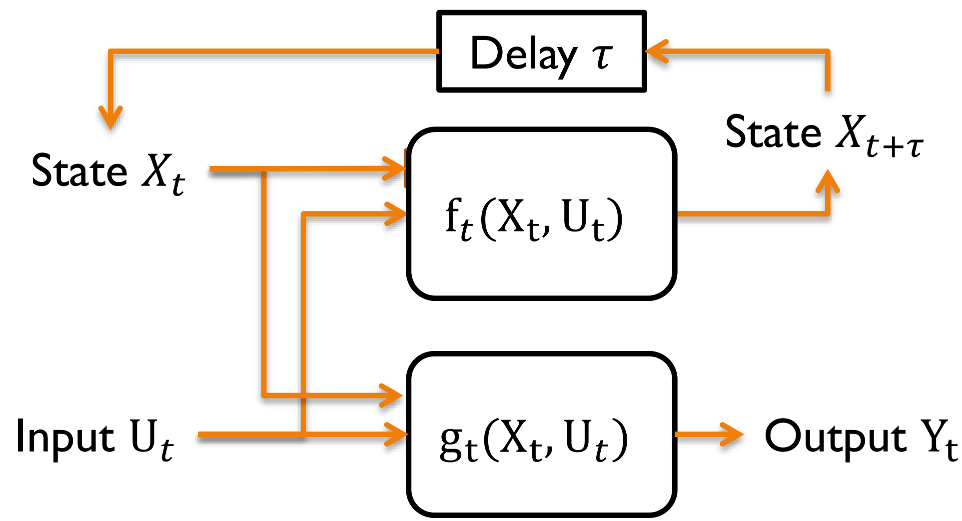
$$f_t = f, g_t = g$$

- **Non deterministic** system:

$$\begin{cases} X_{t+\tau} & = & f_t(X_t, U_t) \color{red}{+ \mathcal{E}_t^X} \\ Y_t & = & g_t(X_t, U_t) \color{red}{+ \mathcal{E}_t^Y} \end{cases}$$

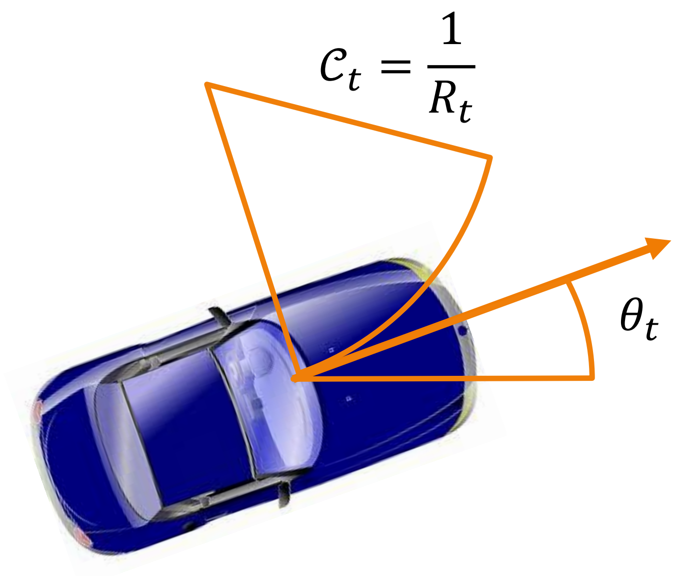
- **Linear** system $(\mathrm{A_t}, \mathrm{B_t}, \mathrm{C_t}, \mathrm{D_t})$:

$$\begin{cases} X_{t+\tau} & = & \color{red}{A_t X_t + B_t U_t} \\ Y_t & = & \color{red}{C_t X_t + D_t U_t} \end{cases}$$



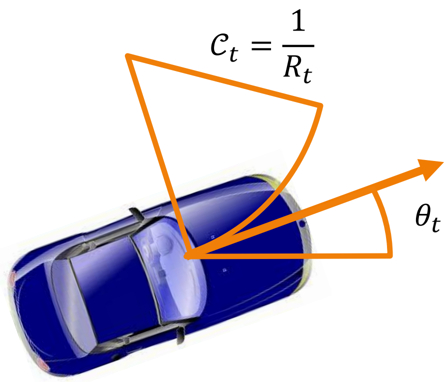
State space representation ensures Markov property.

A simple example: localization of a wheeled vehicle



$$X_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \\ v_t \end{bmatrix}, U_t = \begin{bmatrix} \mathcal{C}_t \\ \alpha_t \end{bmatrix}, Y_t^{odo} = [v_t^{odo}], Y_t^{gps} = \begin{bmatrix} x_t^{gps} \\ y_t^{gps} \end{bmatrix}$$

A simple non-linear example: localization of a wheeled vehicle



$$X_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \\ v_t \end{bmatrix}, U_t = \begin{bmatrix} \mathcal{C}_t \\ \alpha_t \end{bmatrix}$$

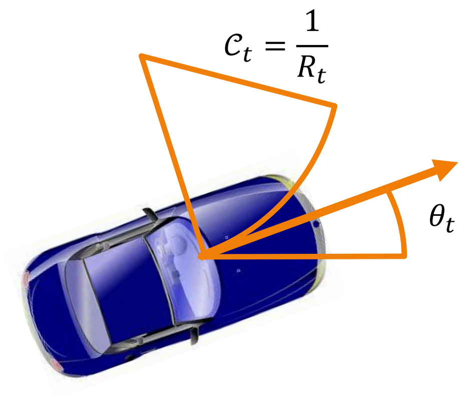
Continuous model:

$$\frac{dX_t}{dt} = \frac{d}{dt}\begin{bmatrix} x_t \\ y_t \\ \theta_t \\ v_t \end{bmatrix} = \begin{bmatrix} v_t \cos(\theta_t) \\ v_t \sin(\theta_t) \\ v_t\, \mathcal{C}_t \\ \frac{1}{M}\alpha_t - \frac{f}{M} v_t \end{bmatrix}$$

Discretized model:

$$\Rightarrow X_{t+\tau} = X_t + \begin{bmatrix} v_t \sin(\theta_t) \\ v_t \cos(\theta_t) \\ v_t\, \mathcal{C}_t \\ \frac{1}{M}\alpha_t - \frac{f}{M} v_t \end{bmatrix} \cdot \tau$$

A simple example: the problem of open looped



$$X_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \\ v_t \end{bmatrix}, U_t = \begin{bmatrix} \mathcal{C}_t \\ \alpha_t \end{bmatrix}$$

Non linearities

Uncertainties

$$\frac{dX_t}{dt} = \frac{d}{dt}\begin{bmatrix} x_t \\ y_t \\ \theta_t \\ v_t \end{bmatrix} = \begin{bmatrix} v_t \times \cos(\theta_t) + \varepsilon_t^x \\ v_t \times \sin(\theta_t) + \varepsilon_t^y \\ v_t \times \mathcal{C}_t + \varepsilon_t^\theta \\ \frac{1}{M}\alpha_t - \frac{f}{M} v_t + \varepsilon_t^v \end{bmatrix}$$

Slippy road, skid

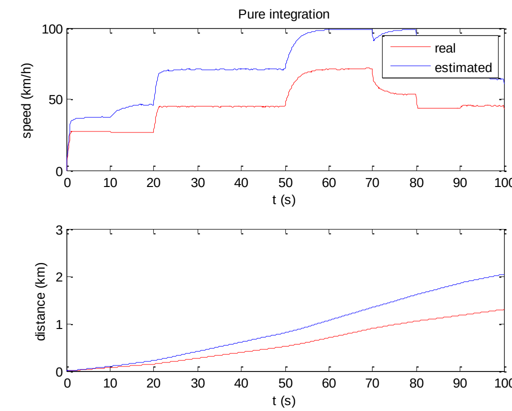
Wind, slope, collision, load, etc

A simple linear example:

Problem: measuring speed $v(t)$ and travelled distance $\mathrm{l}(t)$ of a vehicle with odometer.

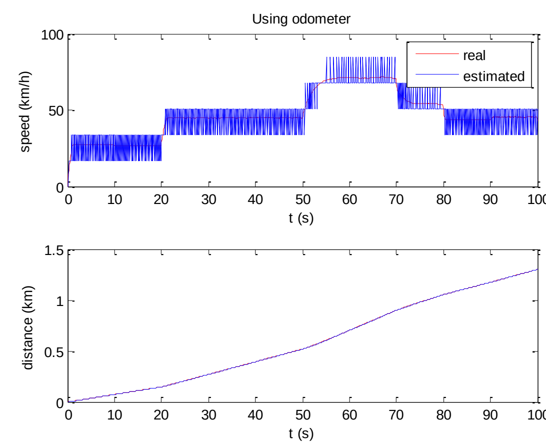
Pure model integration:

$$\begin{cases} \dfrac{dl}{dt}(t) = v(t) + \varepsilon_t^l \\ m\dfrac{dv}{dt}(t) = -f_0\,\mathrm{sign}\big(v(t)\big) - k_f v(t) + a(t) + \varepsilon_t^v \end{cases}$$



Using sensor: odometer counting wheel turn pulses

$$y_t = l_t + \varepsilon_t^y$$



Simulation with varying slope and wind

Linear State Space Representation

State integration:

$$X_t = \begin{bmatrix} v_t \\ l_t \end{bmatrix}, U_t = \begin{bmatrix} 1 \\ a_t \end{bmatrix}, \mathcal{E}_t^X = \begin{bmatrix} \varepsilon_t^v \\ \varepsilon_t^l \end{bmatrix}$$

$$X_{t+1} = X_t + \begin{bmatrix} \frac{1}{m}\left(-f_0 - k_f v_t + a_t + \varepsilon_t^v\right) \\ v_t + \varepsilon_t^l \end{bmatrix} \tau \Leftrightarrow X_{t+1} = A\, X_t + B\, U_t + \mathcal{E}_t^X$$

$$\text{with } A = \begin{bmatrix} 1 - \frac{k_f}{m}\tau & 0 \\ \tau & 1 \end{bmatrix}, B = \frac{\tau}{m}\begin{bmatrix} -f_0 & 1 \\ 0 & 0 \end{bmatrix},$$

Output equation:

$$Y_t = C X_t + D U_t + \mathcal{E}_t^Y$$

$$\text{with } \mathrm{C} = \begin{bmatrix} 0 & 1 \end{bmatrix}, \mathrm{D} = \begin{bmatrix} 0 & 0 \end{bmatrix}, \mathcal{E}_t^Y = \begin{bmatrix} \varepsilon_t^{odo} \end{bmatrix}$$

Bayesian Filtering:
Kalman filter

Hypothesis of Kalman filters:

- The state representation is linear.
- Every "input" of the model $(X_0, \mathcal{E}_t^X, \mathcal{E}_t^Y, U_t)$ is **normally distributed**

Consequences:

- Every state $X_t|X_0, Y_0$, and output $Y_t|X_0, \mathcal{O}_t$ is normally distributed
- State estimation problem is defined by:
  - Model parameters: A_t, B_t, C_t, D_t
  - Initial state distribution: $\hat{X}_0 = E(X_0)\ P_0 = cov(X_0)$
  - State noise distribution: $R_t = cov(\mathcal{E}_t^X)$ $\quad E(\mathcal{E}_t^X) = 0$
  - Output noise distribution: $Q_t = cov(\mathcal{E}_t^Y)$ $\quad E(\mathcal{E}_t^Y) = 0$
  - Output samples: y_t
  - Input samples: u_t

Multivariate normal distribution: definition

Generalization to $\mathbb{R}^{\mathbf{m}}$:

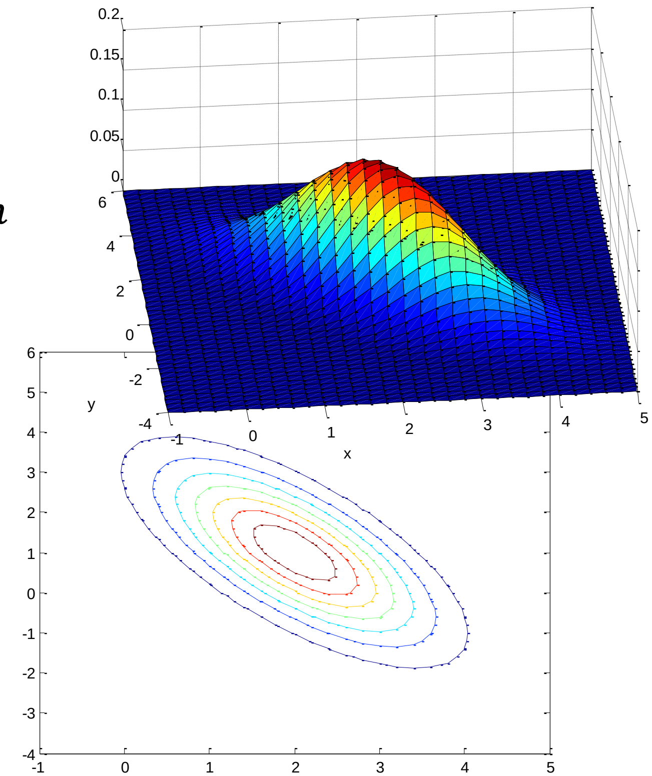
$$X_m \sim \mathcal{N}(\mu_m, \Sigma_{mm}) \Leftrightarrow f_{X_m}(X) = \frac{1}{\sqrt{(2\pi)^m |\Sigma_{mm}|}} e^{-\frac{1}{2}(X-\mu_m)^T \Sigma_{mm}^{-1}(X-\mu_m)}$$

Basic properties:

- $E(X_m) = \mu_m$
- $cov(X_m) = E\big((X-\mu_m)^T(X-\mu_m)\big) = \Sigma_{mm}$

Example:

- $\mu_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \Sigma_{22} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$



Multivariate normal distribution: fundamental properties

Closed under linear transformation:

$$X_m \sim \mathcal{N}(\mu, \Sigma), \mu \in \mathbb{R}^m, \Sigma \in M_{mm}, A \in M_{nm}, B \in M_{n1} \Longrightarrow$$

$$\boxed{AX + B \sim \mathcal{N}(A\mu + B, A\Sigma A^T)}$$

Particular cases:

Given $\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12}^T & \Sigma_{22} \end{bmatrix}\right)$

- Addition: $\boxed{X_1 + X_2 \sim \mathcal{N}\left(\mu_1 + \mu_2, \Sigma_{11} + \Sigma_{22} + \Sigma_{12} + \Sigma_{12}^T\right)}$
- Marginalization: $\boxed{X_1 \sim \mathcal{N}(\mu_1, \Sigma_{11})}$

Kalman Filter:
Prediction and update steps

Prediction step:

- Predicts future state given command:

$$X_{t|t-1} \stackrel{\text{def}}{=} X_t | X_{t-1}, U_{t-1}$$

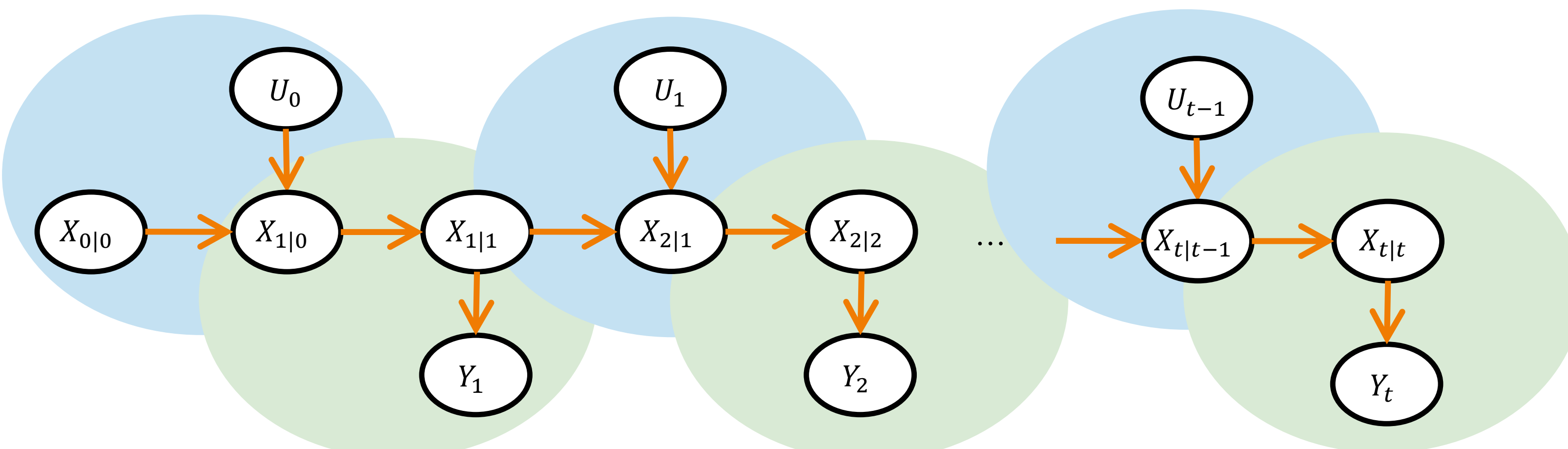
- Model integration

Update step:

- Adjust state given observations:

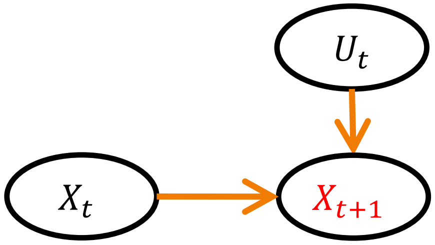
$$X_{t|t} \stackrel{\text{def}}{=} X_t | X_{t-1}, U_{t-1}, Y_t$$

- Model correction



Kalman Filter: Prediction step

Hypothesis:



- $X_{t|t}$ is assumed to be $\mathcal{N}\left(\hat{X}_{t|t}, P_{t|t}\right)$
- $\varepsilon_t^X \sim \mathcal{N}(0, Q_t)$ is a white noise

Compute $\mathrm{P}\left(X_{t+1|t}\right) \stackrel{\text{def}}{=} P(X_{t+1}|X_t, U_t)$:

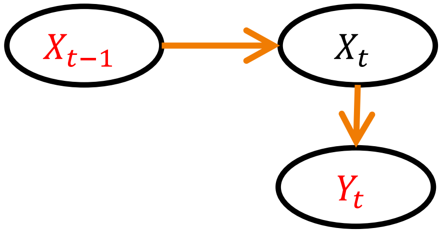
$$X_{t+1|t} = M \begin{bmatrix} X_{t|t} \\ \varepsilon_t^X \end{bmatrix} + V \text{ with } \begin{cases} M = [A_t \;\; I] \\ V = B_t U_t \end{cases}$$

$$\begin{bmatrix} X_{t|t} \\ \varepsilon_t^X \end{bmatrix} \sim \mathcal{N}(\mu, \Sigma) \text{ with } \mu = \begin{bmatrix} \hat{X}_{t|t} \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} P_{t|t} & 0 \\ 0 & Q_t \end{bmatrix} \Rightarrow$$

$$X_{t+1|t} \sim \mathcal{N}(M\mu + V, \mathrm{M}\Sigma M^T)$$

$$\Rightarrow X_{t+1|t} \sim \mathcal{N}\left(\hat{X}_{t+1|t}, P_{t+1|t}\right) \text{ with } \begin{cases} \hat{X}_{t+1|t} = A_t \hat{X}_{t|t} + B_t U_t \\ P_{t+1|t} = A_t P_{t|t} A_t^T + Q_t \end{cases}$$

Kalman Filter:
Output prediction (intermediate step)



Hypothesis:

- $X_{t|t-1}$ is assumed to be $\mathcal{N}\left(\hat{X}_{t|t-1}, P_{t|t-1}\right)$
- $\mathcal{E}_t^Y \sim \mathcal{N}(0, R_t)$ is a white noise

Compute $\boldsymbol{P\left(X_{t|t-1}, Y_{t|t-1}\right) = P(X_t, Y_t | X_{t-1}, U_{t-1})}$ **with** $\boldsymbol{Y_{t|t-1} \stackrel{\text{def}}{=} P(Y_t | X_{t-1}, U_{t-1})}$**:**

$$\begin{bmatrix} X_{t|t-1} \\ Y_t \end{bmatrix} = M \begin{bmatrix} X_{t|t-1} \\ \mathcal{E}_t^Y \end{bmatrix} + V \text{ with } M = \begin{bmatrix} I & 0 \\ C_t & I \end{bmatrix}, V = \begin{bmatrix} 0 \\ D_t U_t \end{bmatrix}$$

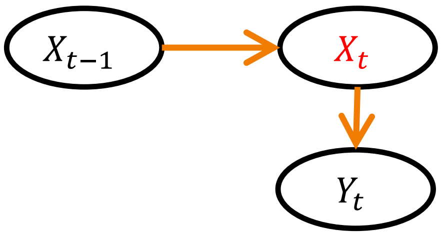
$$\begin{bmatrix} X_{t|t-1} \\ \mathcal{E}_t^Y \end{bmatrix} \sim \mathcal{N}(\mu, \Sigma) \text{ with } \mu = \begin{bmatrix} \hat{X}_{t|t} \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} P_{t|t} & 0 \\ 0 & Q_t \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} X_{t|t-1} \\ Y_t \end{bmatrix} \sim \mathcal{N}(M\mu + V, \mathrm{M}\Sigma M^T)$$

$$\Rightarrow \begin{bmatrix} X_{t|t-1} \\ Y_{t|t-1} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \hat{X}_{t|t-1} \\ \hat{Y}_{t|t-1} \end{bmatrix}, \begin{bmatrix} P_{t|t-1} & P_{t|t-1} C_t^T \\ C_t P_{t|t-1} & S_{t|t-1} \end{bmatrix} \right) \text{ with } \begin{cases} \hat{Y}_{t|t-1} = C_t \hat{X}_{t|t-1} + D_t U_{t-1} \\ S_{t|t-1} = C_t P_{t|t-1} C_t^T + R_t \end{cases}$$

Kalman Filter:
Update step

Hypothesis:



- $\begin{bmatrix} X_{t|t-1} \\ Y_{t|t-1} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \hat{X}_{t|t-1} \\ \hat{Y}_{t|t-1} \end{bmatrix}, \begin{bmatrix} P_{t|t-1} & P_{t|t-1} C_t^T \\ C_t P_{t|t-1} & S_{t|t-1} \end{bmatrix}\right)$
- Observe $Y_t = y_t$

Compute $\mathbf{P}\left(\boldsymbol{X}_{\mathbf{t|t}}\right) \stackrel{\text{def}}{=} \boldsymbol{P}\left(\boldsymbol{X}_{\mathbf{t}} \middle| \boldsymbol{X}_{t|t-1}, \boldsymbol{Y}_t = \boldsymbol{y}_t\right)$

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12}^T & \Sigma_{22} \end{bmatrix}\right) \Longrightarrow$$

$$P(X_1 | X_2 = \vec{x}) \sim \mathcal{N}\left(\mu_1 + \Sigma_{12} {\Sigma_{22}}^{-1} (\vec{x} - \mu_2), \Sigma_{11} - \Sigma_{12} {\Sigma_{22}}^{-1} \Sigma_{12}^T\right)$$

$$\Rightarrow X_{t|t} \sim \mathcal{N}\left(\hat{X}_{t|t}, P_{t|t}\right) \text{ with } \begin{cases} \hat{X}_{t|t} = \hat{X}_{t|t-1} + P_{t|t-1} C_t^T S_{t|t-1}^{-1} (y_t - \hat{Y}_{t|t-1}) \\ P_{t|t} = P_{t|t-1} - P_{t|t-1} C_t^T S_{t|t-1}^{-1} C_t P_{t|t-1} \end{cases}$$

Kalman Filter:
Summary of equations & implementation

Initialisation step:		$\hat{X}_{0\|0} \leftarrow \hat{X}_0,\ P_{0\|0} \leftarrow P_0$	
Prediction step:		run when system time is increased ($t \leftarrow t+1$)	
1.	Predicted state:	$\hat{X}_{t+1\|t} \leftarrow A_t \hat{X}_{t\|t} + B_t U_t$	(model integration)
2.	Prediction covariance:	$P_{t+1\|t} \leftarrow A_t P_{t\|t} A_t^T + Q_t$	(uncertainty increase)
Estimation step:		run when observations are received	
1.	Output prediction:	$\hat{Y}_{t\|t-1} \leftarrow C_t \hat{X}_{t\|t-1} + D_t U_t$	(mean of output posterior)
2.	Output variance:	$S_{t\|t-1} \leftarrow C_t P_{t\|t-1} C_t^T + R_t$	(add observation noise)
3.	Innovation:	$\tilde{Y}_t \leftarrow y_t - \hat{Y}_{t\|t-1}$	(error on output prediction)
4.	Kalman filter gain	$K_t \leftarrow P_{t\|t-1} C_t^T S_{t\|t-1}^{-1}$	(compromise of uncertainty)
5.	Posterior state:	$\hat{X}_{t\|t} \leftarrow \hat{X}_{t\|t-1} + K_t \tilde{Y}_t$	(state correction)
6.	Posterior covariance:	$P_{t\|t} \leftarrow (I - K_t C_t) P_{t\|t-1}$	(uncertainty reduction)

Kalman filter tuning

Initial state distribution:

$$X_0 = \begin{bmatrix} v_0 \\ l_0 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_{v_0}^2 & 0 \\ 0 & 0 \end{bmatrix}\right) \qquad \text{with } \sigma_{v_0} = 1\, m/s$$

State integration noise:

Force uncertainty: $\varepsilon_t^v \sim \mathcal{N}(0, \sigma_v^2)$ with $\sigma_v = m \cdot 1 \cdot 9.8\, m/s^2$

Slip uncertainty: $\varepsilon_t^l \sim \mathcal{N}(0, \sigma_l^2)$ with $\sigma_l = 1\, m/s$

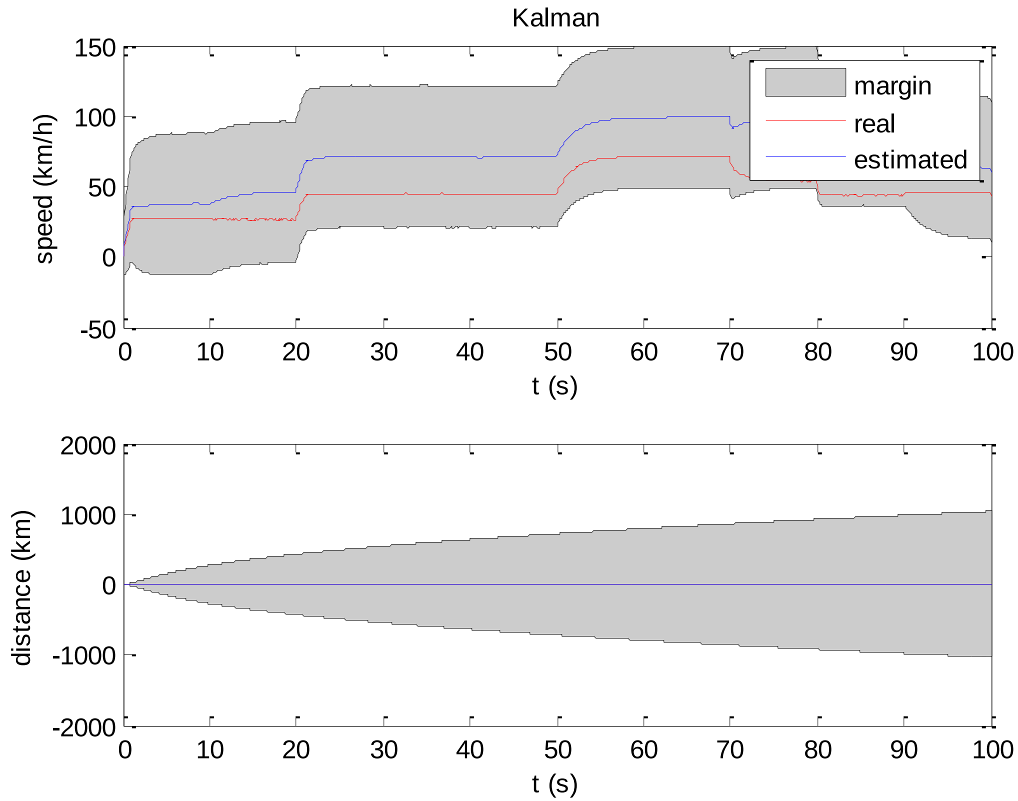
$$\varepsilon_t^v \perp \varepsilon_t^l \Rightarrow \mathcal{E}_t^X \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, Q\right) \text{ avec } \boxed{Q_t = \tau^2 \begin{bmatrix} \frac{\sigma_v^2}{m^2} & 0 \\ 0 & \sigma_l^2 \end{bmatrix}}$$

Measurement noise:

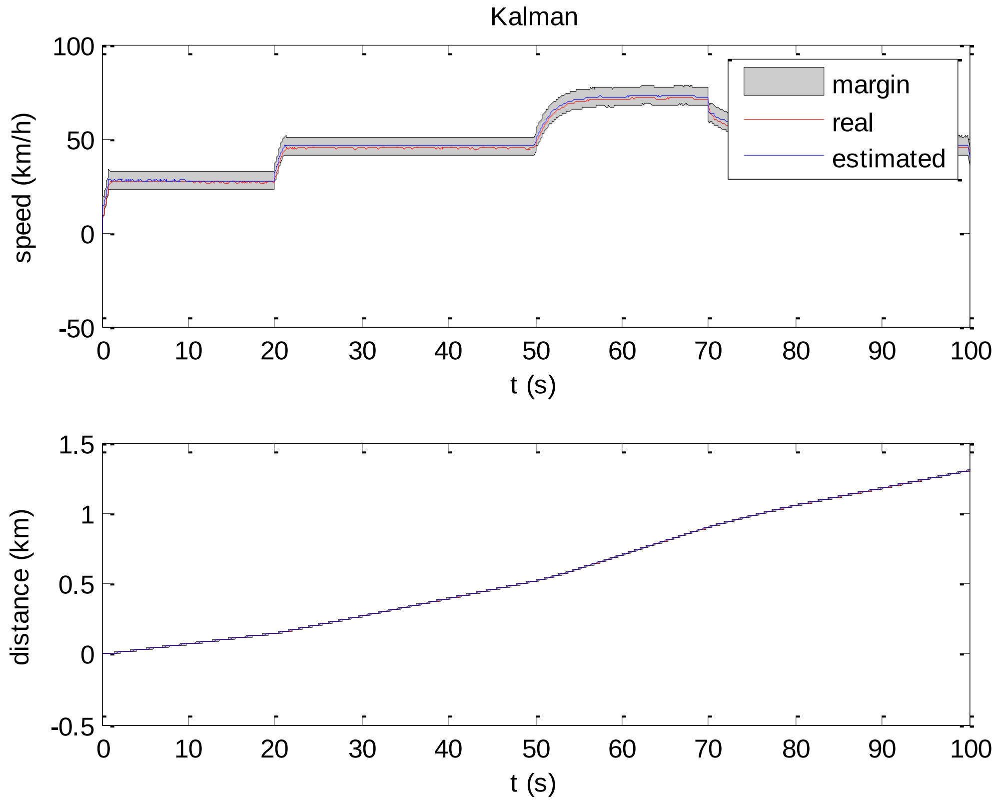
$$\varepsilon_t^y \sim \mathcal{N}(0, \sigma_y^2) \qquad \text{with } \sigma_y = \sqrt{\frac{q^2}{12}}, q = 50\, cm$$

$$\Rightarrow \boxed{R_t = [\sigma_y^2]}$$

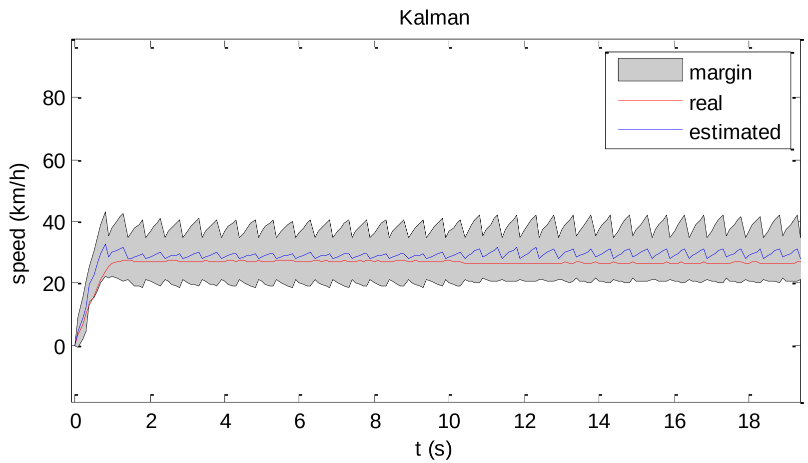
Kalman Filter:
Without measures

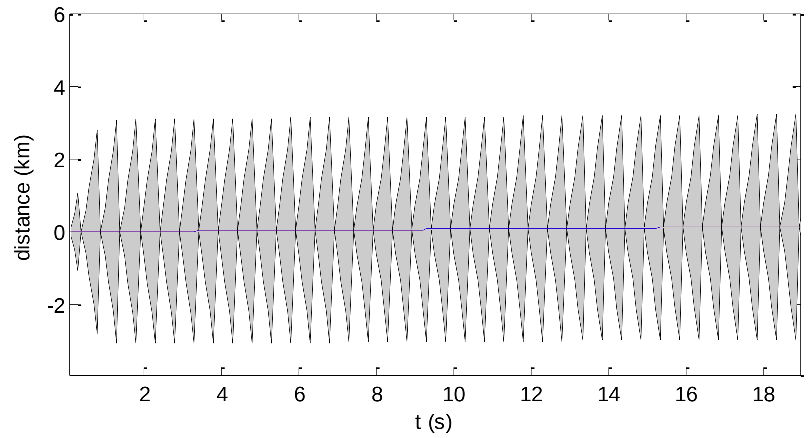


Kalman Filter:
With measures



Kalman Filter:
Slowing down measurement time rate





Kalman Filter: Extensions

- **Continuous state:**

Make $\tau \to 0$: $\frac{d\hat{X}_{|t_o}}{dt}(t) = A_t \hat{X}_{|t_0}(t) + B_t U(t), \ldots$

- **Hybrid filters:**

Combine continuous predictions with discrete updates

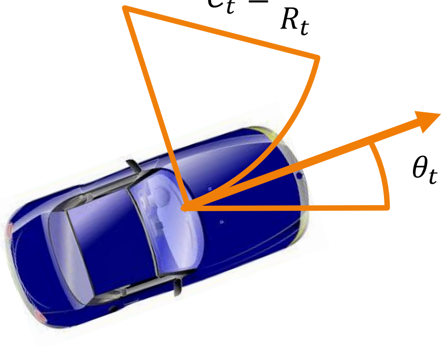
- **Non linear models: extended Kalman filters**

Linearize state space representation (f_t, g_t) around estimated state $\hat{X}_t$:

$$\hat{X}_{t+1|t} = f_t\left(\hat{X}_{t|t}, U_t\right), \hat{Y}_{t+1|t} = g_t\left(\hat{X}_{t|t}, U_t\right),$$

$$A_t = \frac{\partial f}{\partial X}\left(\hat{X}_{t|t}\right), C_t = \frac{\partial g}{\partial X}\left(\hat{X}_{t|t}\right)$$

Example of an extended Kalman filter



$$X_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \\ v_t \end{bmatrix}, U_t = \begin{bmatrix} \mathcal{C}_t \\ \alpha_t \end{bmatrix} \qquad \frac{dX_t}{dt} = \frac{d}{dt}\begin{bmatrix} x_t \\ y_t \\ \theta_t \\ v_t \end{bmatrix} = \begin{bmatrix} v_t \times \cos(\theta_t) + \varepsilon_t^x \\ v_t \times \sin(\theta_t) + \varepsilon_t^y \\ v_t \times \mathcal{C}_t + \varepsilon_t^\theta \\ \frac{1}{M}\alpha_t - \frac{f}{M} v_t + \varepsilon_t^v \end{bmatrix}$$

$$\begin{cases} X_{t+1|t} = X_t + \begin{bmatrix} v_t \times \cos(\theta_t) + \varepsilon_t^x \\ v_t \times \sin(\theta_t) + \varepsilon_t^y \\ v_t \times \mathcal{C}_t + \varepsilon_t^\theta \\ \frac{1}{M}\alpha_t - \frac{f}{M} v_t + \varepsilon_t^v \end{bmatrix} \times \tau \\ A_t = \frac{\partial f}{\partial X}(\hat{X}_{t|t}) = \begin{bmatrix} 1 & 0 & -v_t \times \sin(\theta_t) \times \tau & \cos(\theta_t) \times \tau \\ 0 & 1 & v_t \times \cos(\theta_t) \times \tau & \sin(\theta_t) \times \tau \\ 0 & 0 & 1 & \mathcal{C}_t \times \tau \\ 0 & 0 & 0 & 1 - \frac{f}{M} v_t \tau \end{bmatrix} \end{cases}$$

Kalman Filter:
A summary

- Bayesian filtering updates dynamic state using bayesian inference
  - Relies on Markov property
  - Find best balance between uncertainty of model and sensors
- Kalman filter is a special case of bayesian filtering
  - Relies on a state space representation
  - All input distribution of the models are gaussian (model & observation noises, initial state)
- Wide application scope:
  - Continuous/discrete systems
  - Multiple sensors (data fusion)

Exercise:
hot air balloon

Estimate vertical speed and altitude from:

$$
\begin{cases}
m\ddot{z} = -mg + Vg(\rho_{in} - \rho_{out}) + \varepsilon \\
\rho_{out} = \frac{P_{out}}{RT_{out}} \\
\rho_{in} = \frac{P_{in}}{RT_{in}} \\
P_{out} = P_{in} \\
\dot{T}_{in} = -\frac{S\sigma_b}{Vc_{air}}(T_{out} - T_{in}) + k_1(\Delta T + T_{in} - T_{out})^2 \\
\Delta T = k_2 P \\
T_{out} = 300 - 6.4 \cdot 10^{-3} z + \varepsilon_T \\
P_{out} = 10^5 - 7.15z + \varepsilon_P
\end{cases}
$$

Sensors:

1. Accelerometer: $m_a = \ddot{z} + \varepsilon_a$
2. Altimeter: $m_z = (10^5 - P_{out} + \varepsilon_z)/7.15$
3. Thermometer: $m_T = T_{out} + \varepsilon_{T_{out}}$