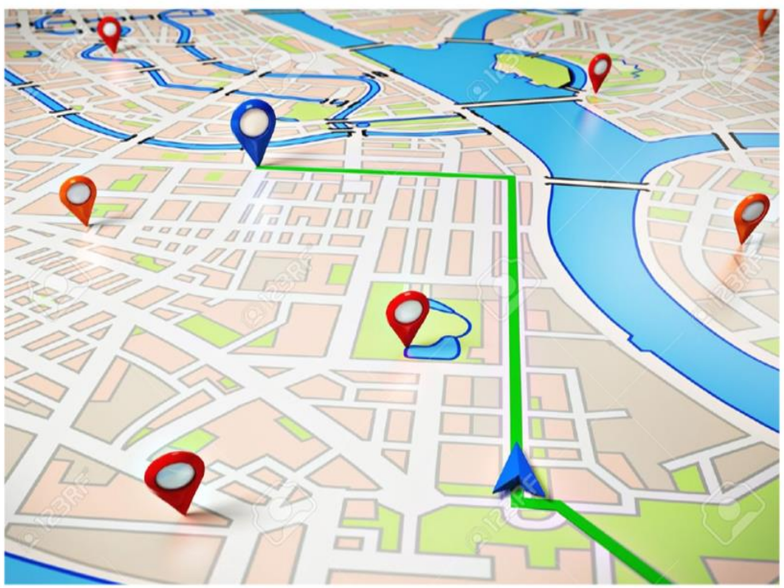
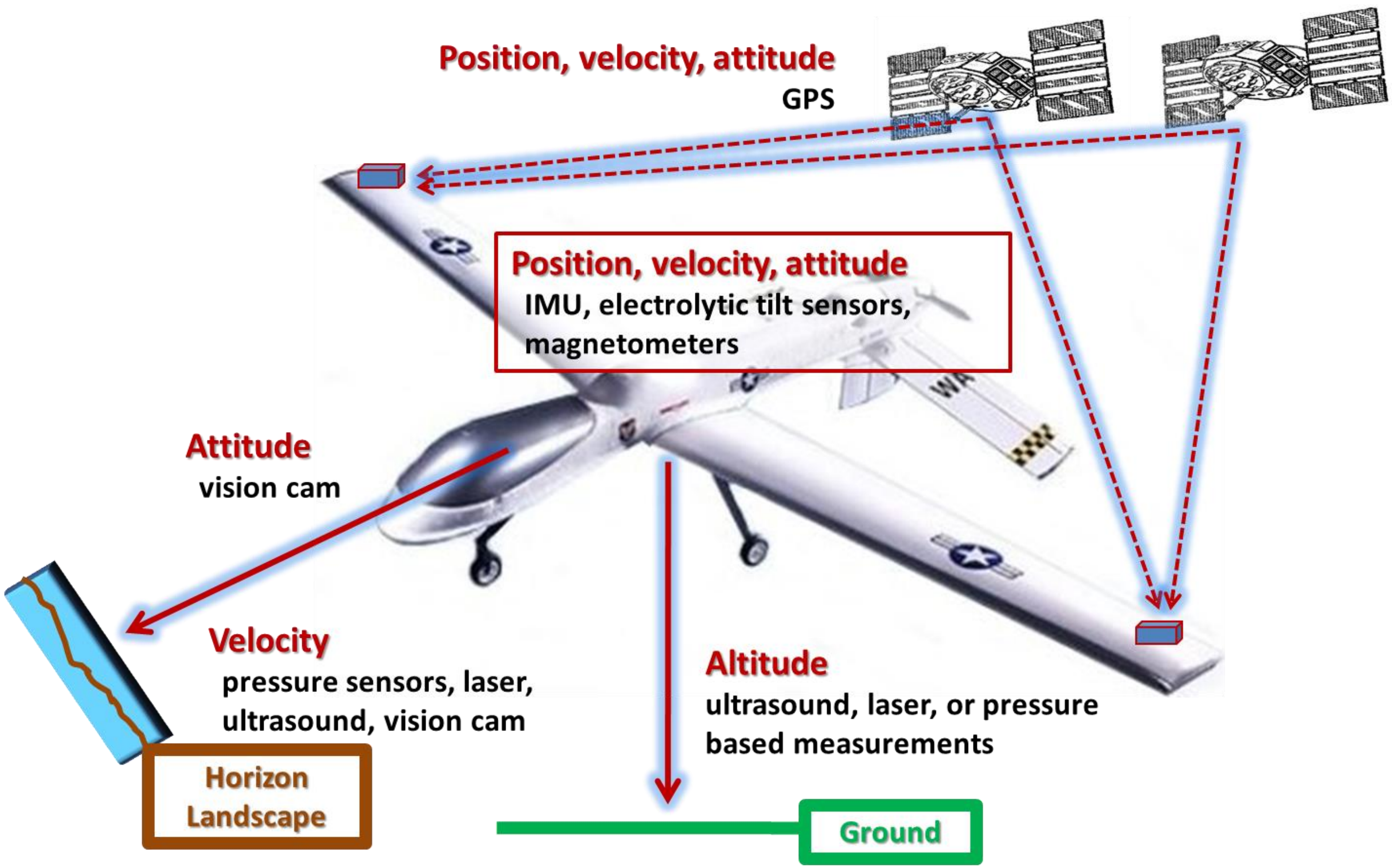


Lecture

KALMAN FILTERING

- INTRODUCTION -

Motivation – why to fuse data? vs what are the sources?



Kalman filtering
plan of the lectures

- **Introduction**
  - Assignment – a constant estimation problem
- **Practical considerations in the KF model tuning**
- **PVA estimation**
  - loosely-coupled, tightly-coupled & ultra-tightly coupled modeling,
  - direct vs indirect model,
- **PNT systems designs**
  - dynamic tuning,
  - practical considerations.

Kalman filtering - intro

What is a Kalman Filter and What Can It Do?
A Kalman filter is an ***optimal estimator*** – i.e. infers parameters of interest from indirect, inaccurate and uncertain observations. It is ***recursive*** so that new measurements can be processed as they arrive.

Optimal in what sense?
If all noise is Gaussian, the Kalman filter minimizes the mean square error of the estimated parameters.

What if the noise is NOT Gaussian?
Given only the mean and standard deviation of noise, the Kalman filter is the best linear estimator. Non-linear estimators may be better.

Why is Kalman Filtering so popular?

- Good results in practice due to optimality and structure.
- Convenient form for *online real time* processing.
- Easy to formulate and implement given a basic understanding.
- Measurement equations need not be inverted.

Least-square solution for overdetermined linear systems

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1n} \\ h_{21} & h_{22} & h_{23} & \cdots & h_{2n} \\ h_{31} & h_{32} & h_{33} & \cdots & h_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{l1} & h_{l2} & h_{l3} & \cdots & h_{ln} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_m \end{bmatrix}$$

$$Hx = z$$

$$\varepsilon^2(\hat{x}) = |H\hat{x} - z|^2 = \sum_{i=1}^{m} \left[\sum_{j=1}^{n} h_{ij}\hat{x}_j - z_i \right]^2$$

$$0 = \frac{\partial \varepsilon^2}{\partial \hat{x}_k} = 2\sum_{i=1}^{m} h_{ik} \left[\sum_{j=1}^{n} h_{ij}\hat{x}_j - z_i \right]$$

$$H^{\mathrm{T}}H\hat{x} = H^{\mathrm{T}}z$$

$$\hat{x} = (H^{\mathrm{T}}H)^{-1}H^{\mathrm{T}}z$$

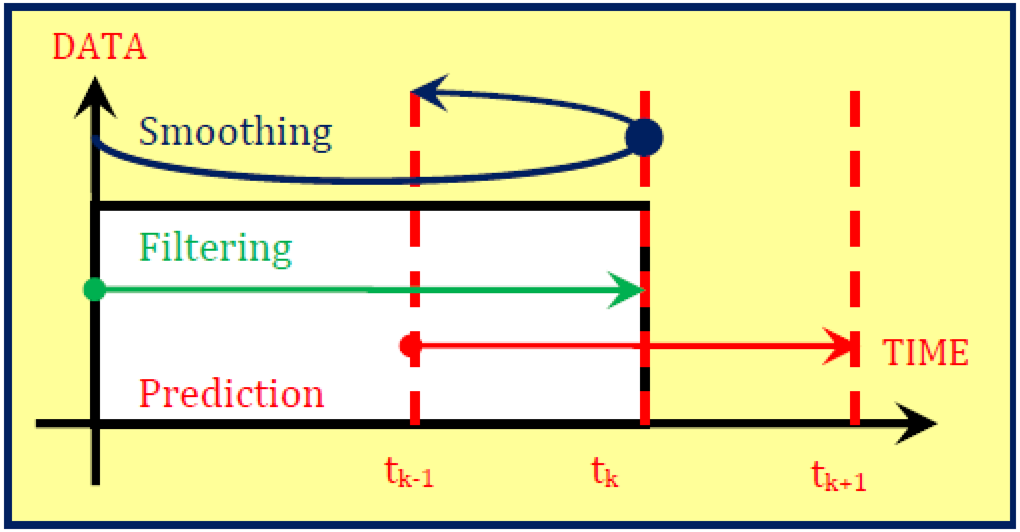
Kalman filtering - intro

Why use the word "Filter"?

The process of finding the "best estimate" from noisy data amounts to "filtering out" the noise.

However a Kalman filter also doesn't just clean up the data measurements, but also *projects* these measurements onto the state estimate.

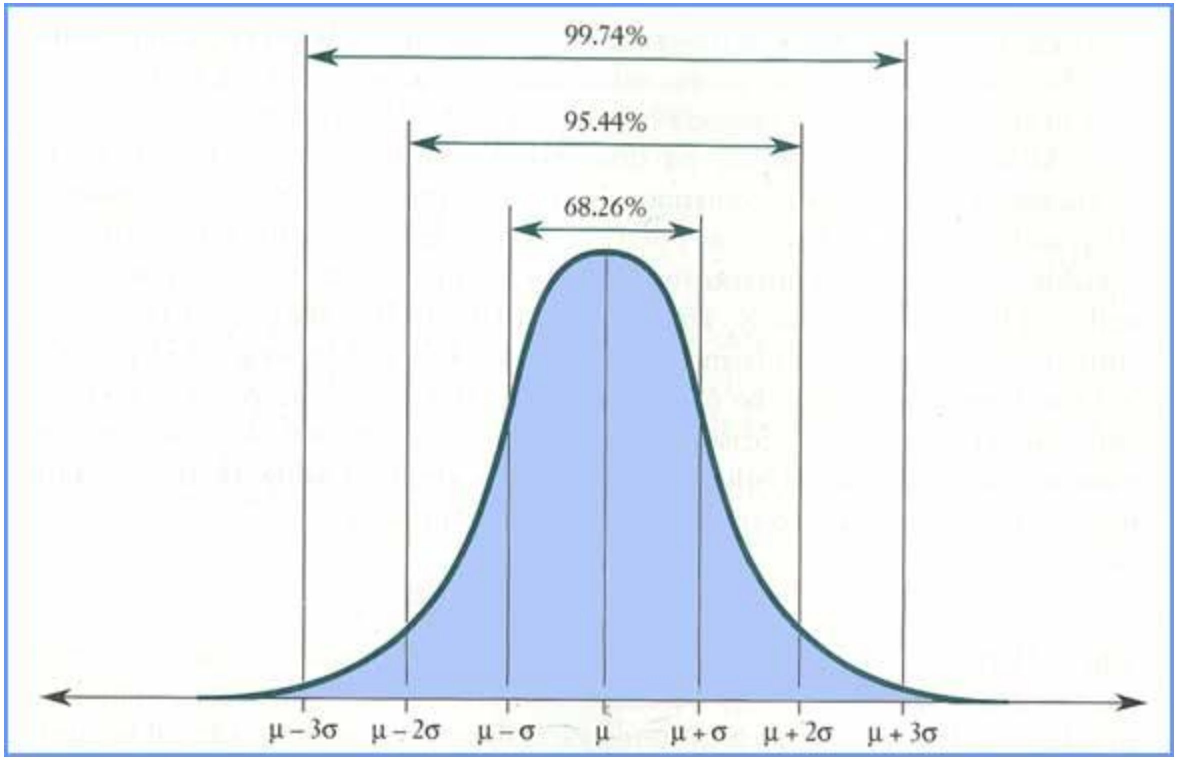
Filtering vs. predicting vs. smoothing



Normal or Gaussian probability density function

Given a random process $\boldsymbol{x} \approx N(\bar{x}, \sigma^2)$, i.e. a continuous random process X that is normally distributed with mean $\bar{x}$ and variance σ^2 (standard deviation σ), the probability density function for $\boldsymbol{x}$ is given by

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\frac{(x-\bar{x})^2}{\sigma^2}\right] \qquad \bar{x} = E\langle x \rangle$$



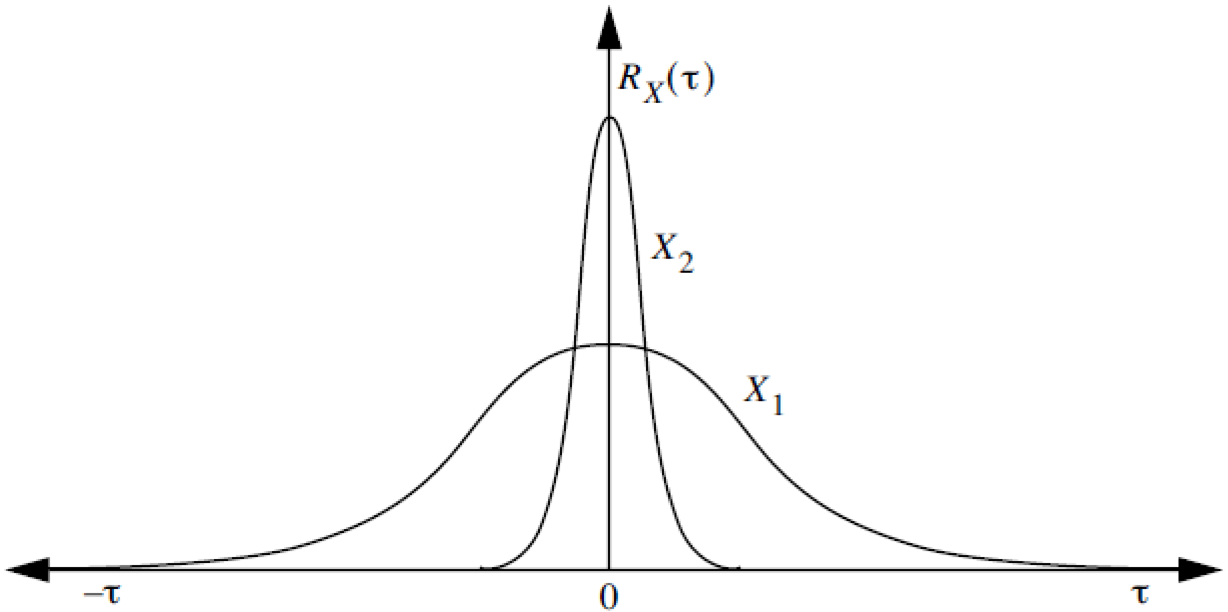
Spatial vs. Spectral Signal Characteristics

useful time-related characteristic of a random signal is its ***autocorrelation function***.

$$R_X(t_1, t_2) = E[X(t_1)X(t_2)]$$

If the process is *stationary* (the density is invariant with time) it depends only on the difference $\tau = t_1 - t_2$

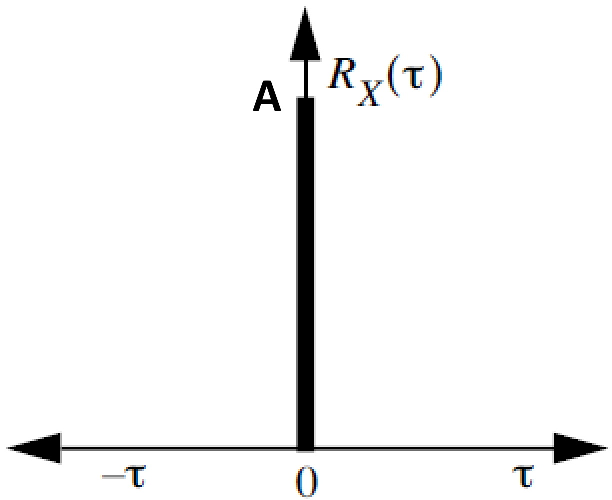
$$R_X(\tau) = E[X(t)X(t+\tau)]$$

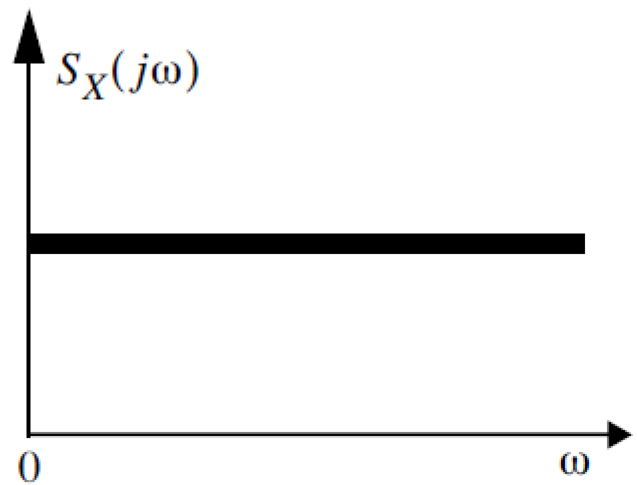


White noise

An important case of a random signal is the case where the autocorrelation function is a *dirac delta* $\delta(\tau)$ function which has zero value everywhere except when $\tau = 0$.

$$R_X(\tau) = \begin{cases} \text{if } \tau = 0 \text{ then } A \\ \text{else } 0 \end{cases}$$

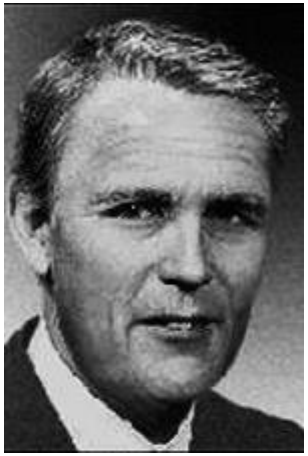




Kalman filter

The Kalman filter is named after Rudolph E. Kalman, who in 1960 published his famous paper describing a recursive solution to the discrete-data linear filtering problem.

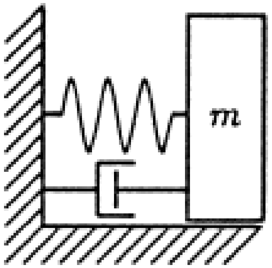
http://www.cs.unc.edu/~welch/kalman/



- The Kalman filter (KF) was introduced as a recursive algorithm for state estimation, which is optimal in the sense of minimum variance or least square error.
- Recursive algorithm had the advantage of being easily implementable in digital computers.
- **Previous non-recursive estimation methods used the entire measurement set, whereas the recursive estimation of the KF use current measurements as well as prior estimates to propagate the states from an initial estimate. The KF is therefore more computational efficient as it can discard previous measurements and update the state estimates with only the present measurements.**

Dynamic systems represented by differential equations

Example: Harmonic resonator with a linear damping



$$F(t) = ma(t) = m\left[\frac{d^2\delta}{dt^2}(t)\right] = -k_s\delta(t) - k_d\frac{d\delta}{dt}(t)$$

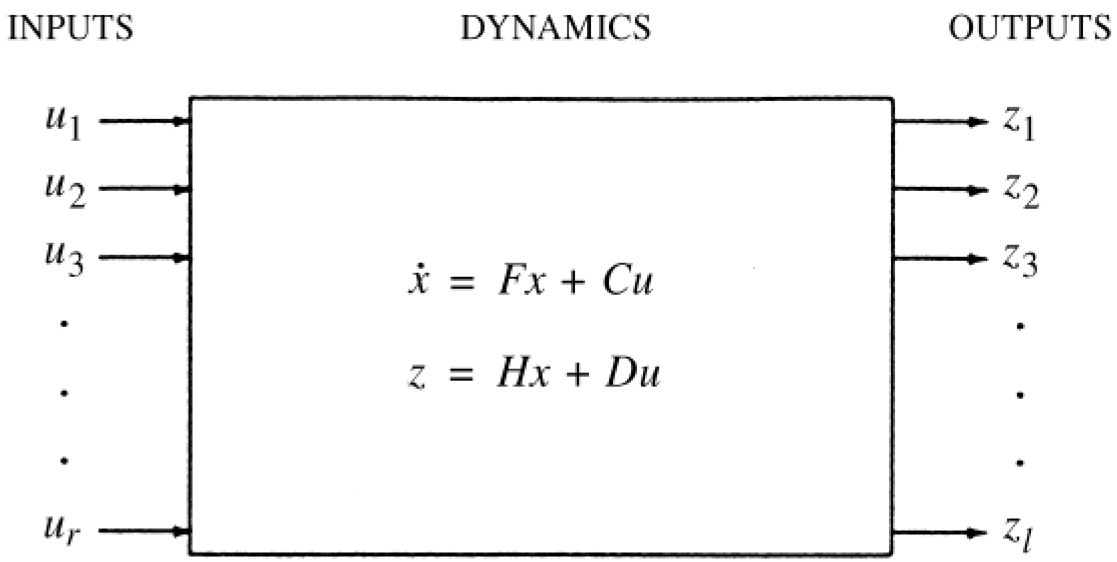
k_s - the spring constant, k_d - a drag coefficient of the dashpot
δ – displacement of the mass from its position at rest

2nd order differential equation

2x 1st order differential equations $x_1 = \delta, x_2 = \frac{d\delta}{dt}$

$$\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k_s}{m} & \frac{k_d}{m} \end{bmatrix}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

Continuous form of the model



- **INPUTS** – are under the control and thus known(at least statistically) or at least measurable.
- **STATE VARIABLES** – generally they cannot be measured directly but must be somehow inferred from what can be measured.
- **OUTPUTS** – can be known through measurements.

Continuous form of the model in III^c

$$
\begin{aligned}
\dot{x}(t) &= A(t)x(t) + B(t)u(t) + G(t)w(t) \\
z(t) &= C(t)x(t) + D(t)u(t) + v\,(t)
\end{aligned}
$$

x(t) = n x 1 – state vector
z(t) = l x 1 – measurement vector
u(t) = r x 1 – deterministic input vector
A(t) = n x n – state transition matrix (time-invariant OR time-varying)
B(t) = n x r – input coupling matrix
G(t) = n x r – process noise coupling matrix
C(t) = l x n – measurement sensitive matrix
D(t) = l x r – output coupling matrix
w(t) = r x 1 – zero-mean process noise
v(t) = l x 1 – zero-mean measurement noise
Q(t) = n x r – process noise covariance matrix
R(t) = l x l – measurement noise covariance matrix

$$
\begin{aligned}
E\langle w(t)\rangle &= 0, \\
E\langle v(t)\rangle &= 0, \\
E\langle w(t_1)w^{\mathrm{T}}(t_2)\rangle &= Q(t_1)\delta(t_2 - t_1), \\
E\langle v(t_1)v^{\mathrm{T}}(t_2)\rangle &= R(t_1)\delta(t_2 - t_1). \\
E\langle v(t_1)v^{\mathrm{T}}(t_2)\rangle &= M(t_1)\delta(t_2 - t_1).
\end{aligned}
$$

Matrix dimensions

Symbol	Vector Name	Dimensions	Symbol	Matrix Name	Dimensions Row	Dimensions Column
x	System state	n	Φ	State transition	n	n
w	Process noise	r	G	Process noise coupling	n	r
u	Control input	s	Q	Process noise covariance	r	r
z	Measurement	ℓ	H	Measurement sensitivity	ℓ	n
v	Measurement noise	ℓ	R	Measurement noise covariance	ℓ	ℓ

Mathematical models of dynamic systems

	Continuous	Discrete
Time invariant		
Linear	$\dot{x}(t) = Fx(t) + Cu(t)$	$x_k = \Phi x_{k-1} + \Gamma u_{k-1}$
General	$\dot{x}(t) = f(x(t), u(t))$	$x_k = f(x_{k-1}, u_{k-1})$
Time varying		
Linear	$\dot{x}(t) = F(t)x(t) + C(t)u(t)$	$x_k = \Phi_{k-1} x_{k-1} + \Gamma_{k-1} u_{k-1}$
General	$\dot{x}(t) = f(t, x(t), u(t))$	$x_k = f(k, x_{k-1}, u_{k-1})$

Standard symbols used in KF theory

Definition of Notational Usage

x	$\underline{x}$ $\vec{x}$ $\mathbf{x}$	Vector
x_k		The kth component of the vector x
x_k	$x[k]$	The kth element of the sequence $\ldots, x_{k-1}, x_k, x_{k+1}, \ldots$ of vectors
$\hat{x}$	$E\langle x \rangle$ $\bar{x}$	An estimate of the value of x
$\hat{x}_k(-)$	$\hat{x}_{k\|k-1}$ $\hat{x}_{k-}$	A priori estimate of x_k, conditioned on all prior measurements except the one at time t_k
$\hat{x}_k(+)$	$\hat{x}_{k\|k}$ $\hat{x}_{k+}$	A posteriori estimate of x, conditioned on all available measurements at time t_k
$\dot{x}$	x_t dx/dt	Derivative of x with respect to t (time)

Standard symbols used in KF theory

Symbols			Symbol Definition
I[a]	II[b]	III[c]	
F	F	A	Dynamic coefficient matrix of continuous linear differential equation defining dynamic system
G	I	B	Coupling matrix between random process noise and state of linear dynamic system
H	M	C	Measurement sensitivity matrix, defining linear relationship between state of the dynamic system and measurements that can be made
$\overline{K}$	Δ	K	Kalman gain matrix
P	P		Covariance matrix of state estimation uncertainty
Q	Q		Covariance matrix of process noise in the system state dynamics
R	0		Covariance matrix of observational (measurement) uncertainty
x	x		State vector of a linear dynamic system
z	y		Vector (or scalar) of measured values
Φ	Φ		State transition matrix of a discrete linear dynamic system

From continuous to a discrete form of the model

State transition matrix $$\Phi_k = I + A\Delta t + \frac{A^2 \Delta t^2}{2!} + \cdots$$

Control vector matrix $$\Gamma_k = \left(\int_{\tau=0}^{\Delta t} e^{A\tau} d\tau \right) B$$

Control vector $$u_k = \Phi_k \int_{\tau=0}^{\Delta t} \Phi^{-1}(\tau) C(\tau) u(\tau)\, d\tau$$

Measurement matrix $$H_k = C$$

Process noise covariance $$Q_k = \int_{\tau=0}^{\Delta t} e^{A\tau} Q_c e^{A^T \tau} d\tau$$

$$Q_k \approx G_k Q_c G_k^T \Delta t$$ First-order approximation

$$Q_k = \frac{1}{2} \Delta t \left(\Phi_k G_k Q_c G_k^T + G_k Q_c G_k^T \Phi_k^T \right)$$

Measurement noise covariance $$R_k = R_c / \Delta t$$

Van Loan's method $$M = \begin{bmatrix} -A & G Q_c G^T \\ 0 & A^T \end{bmatrix} \Delta t \qquad G = exp(M) = \begin{bmatrix} \cdots & \Phi^{-1} Q_k \\ 0 & \Phi^T \end{bmatrix}$$

Kalman filtering – a discrete model

$$x_k = \Phi_{k-1} x_{k-1} + G_{k-1} w_{k-1},$$
$$z_k = H_k x_k + v_k,$$
$$P_k = \Phi_{k-1} P_{k-1} \Phi_{k-1}^{\mathrm{T}} + G_{k-1} Q_{k-1} G_{k-1}^{\mathrm{T}}$$

where *x* denotes a state vector,
z a measurement vector,
Φ a state transition matrix,
H a measurement matrix giving the ideal connection between the measurement and the state vector,
w, v vectors of the process and measurement noise,

$$w_k \sim N(0, Q_k) \qquad v_k \sim N(0, R_k)$$

$$E\left[w_k w_i^T\right] = \begin{cases} Q_k, & i = k \\ 0, & i \neq k \end{cases} \qquad E\left[v_k v_i^T\right] = \begin{cases} R_k, & i = k \\ 0, & i \neq k \end{cases}$$

$$E\left[w_k v_i^T\right] = 0, \qquad \text{for all } k \text{ and } i.$$

Observability of the time-invariant system – a discrete model

$$
\begin{aligned}
\mathbf{y}_k &= \mathbf{H}_k \mathbf{x}_k \\
\mathbf{y}_{k+1} - \mathbf{H}_{k+1}\boldsymbol{\Gamma}_k \mathbf{u}_k &= \mathbf{H}_{k+1}\boldsymbol{\Phi}_k \mathbf{x}_k \\
\vdots \quad &\quad \vdots \\
\mathbf{y}_{k+N} - \mathbf{H}_{k+N} \sum_{j=0}^{n-1} \prod_{i=j+1}^{n-1} \boldsymbol{\Phi}_{k+i}\boldsymbol{\Gamma}_{k+j}\mathbf{u}_{k+j} &= \mathbf{H}_{k+N}\boldsymbol{\Phi}_{k+N-1}\cdots\boldsymbol{\Phi}_k \mathbf{x}_k
\end{aligned}
$$

$$
\mathbf{Z} = \begin{bmatrix} \mathbf{H}_k \\ \mathbf{H}_{k+1}\boldsymbol{\Phi}_k \\ \vdots \\ \mathbf{H}_{k+N}\boldsymbol{\Phi}_{k+N-1}\cdots\boldsymbol{\Phi}_k \end{bmatrix} \mathbf{x}_k
$$

The system is observable if ***M*** matrix has a rank ***n*** corresponding to the dimension of the system state vector.

$$
M = [H^{\mathrm{T}} \quad \Phi^{\mathrm{T}} H^{\mathrm{T}} \quad (\Phi^{\mathrm{T}})^2 H^{\mathrm{T}} \quad \cdots \quad (\Phi^{\mathrm{T}})^{n-1} H^{\mathrm{T}}]
$$

Observability - examples

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \quad M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{rank of } M = 2$$

$$z(t) = [1 \quad 0] x(t).$$

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \quad M = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}, \quad \text{rank of } M = 1$$

$$z(t) = [0 \quad 1] x(t).$$

---

acceleration is available

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x}(t) \\ y(t) &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{x}(t) \end{aligned} \qquad \mathcal{O} = \begin{bmatrix} \mathbf{H} \\ \mathbf{H\Phi} \\ \vdots \\ \mathbf{H\Phi} \cdots \mathbf{\Phi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ & \vdots & \\ 0 & 0 & 1 \end{bmatrix}.$$

$$\begin{aligned} \mathbf{x}_{k+1} &= \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k \\ y_k &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k. \end{aligned} \qquad \mathcal{O} = \begin{bmatrix} \mathbf{H} \\ \mathbf{H\Phi} \\ \mathbf{H\Phi}^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & T & T^2/2 \\ 1 & 2T & 2T^2 \end{bmatrix}$$

position is measured instead

Discrete time Kalman filter equations

System dynamic model:

$$x_k = \Phi_{k-1} x_{k-1} + w_{k-1}$$
$$w_k \sim \mathcal{N}(0, Q_k)$$

Measurement model:

$$z_k = H_k x_k + v_k$$
$$v_k \sim \mathcal{N}(0, R_k)$$

Initial conditions:

$$E\langle x_0 \rangle = \hat{x}_0$$
$$E\langle \tilde{x}_0 \tilde{x}_0^{\mathrm{T}} \rangle = P_0$$

Independence assumption:

$$E\langle w_k v_j^{\mathrm{T}} \rangle = 0 \quad \text{for all } k \text{ and } j$$

State estimate extrapolation

$$\hat{x}_k(-) = \Phi_{k-1} \hat{x}_{k-1}(+)$$

Error covariance extrapolation

$$P_k(-) = \Phi_{k-1} P_{k-1}(+) \Phi_{k-1}^{\mathrm{T}} + Q_{k-1}$$

! **symmetric & positively definite**

State estimate observational update

$$\hat{x}_k(+) = \hat{x}_k(-) + \overline{K}_k [z_k - H_k \hat{x}_k(-)]$$

Error covariance update

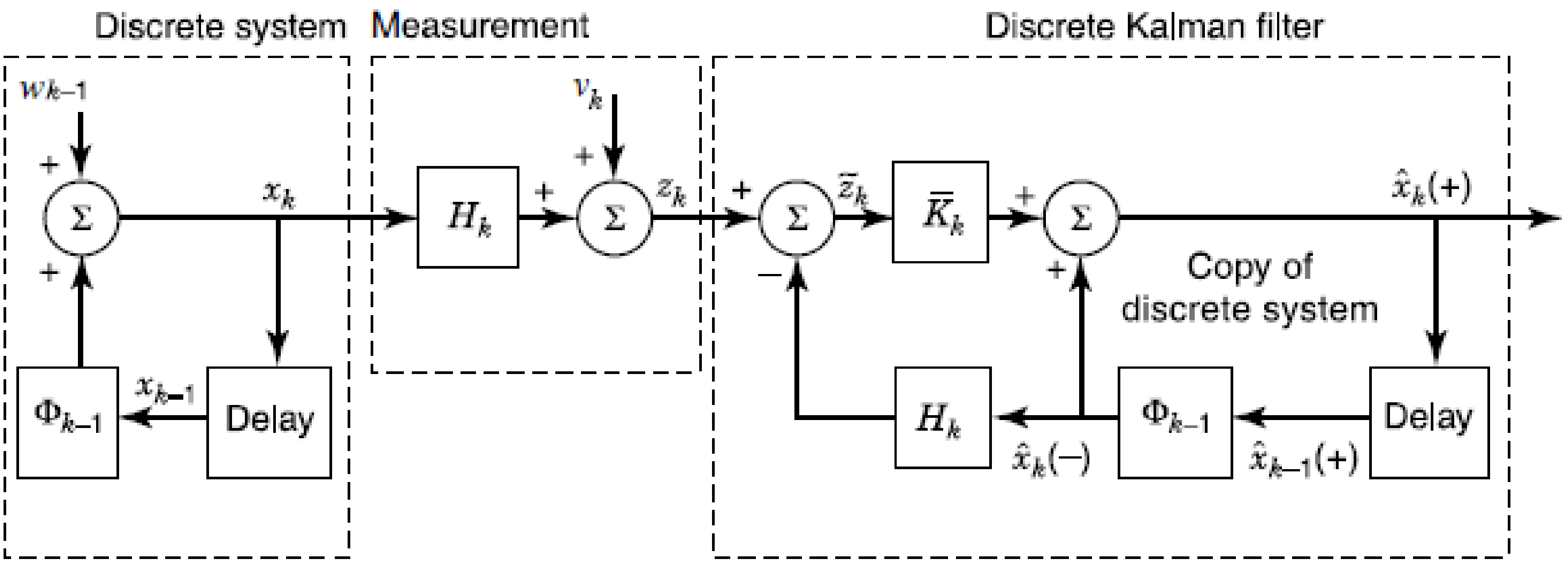
$$P_k(+) = [I - \overline{K}_k H_k] P_k(-)$$

! **symmetric & positively definite**

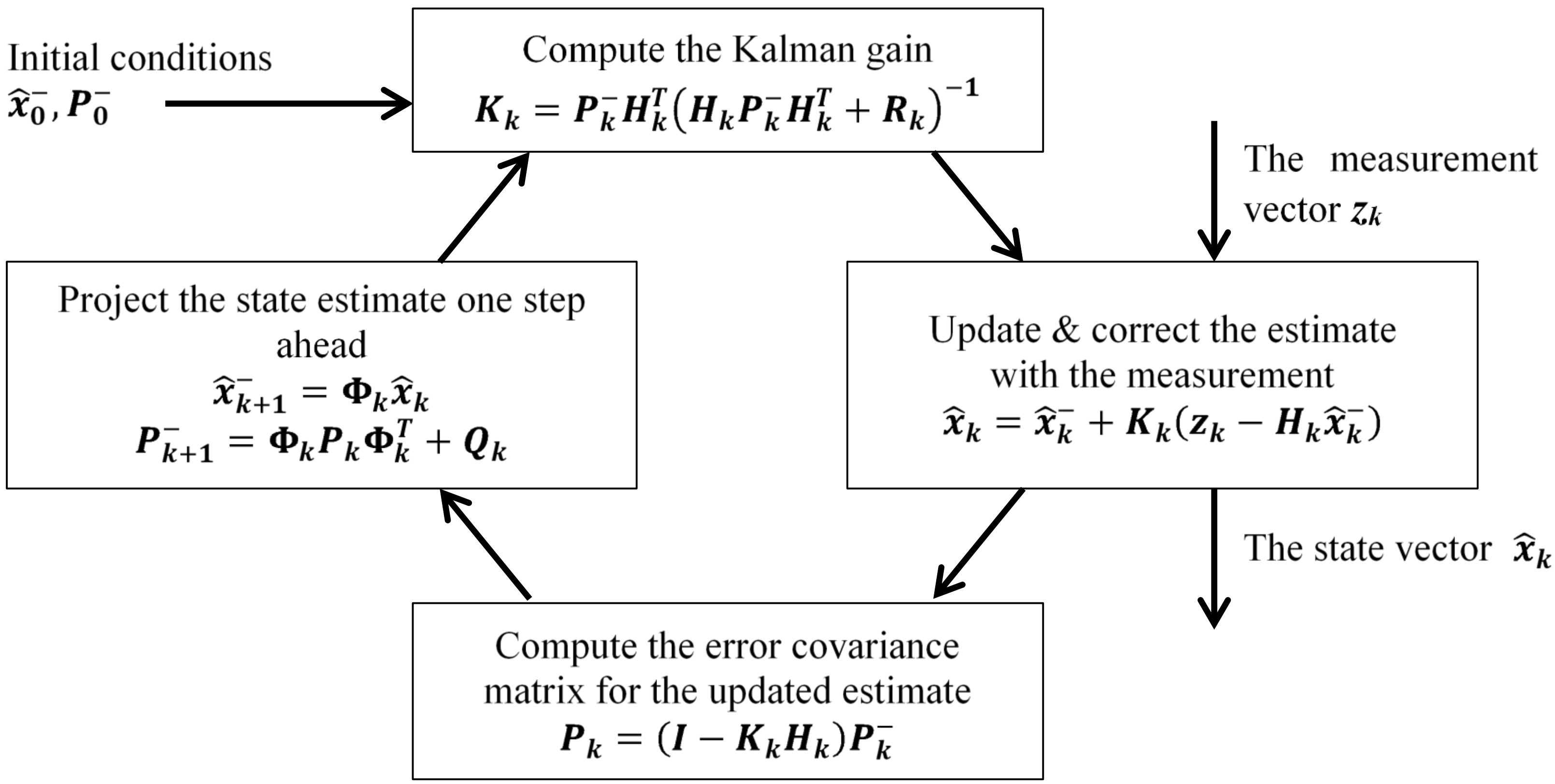
Kalman gain matrix

$$\overline{K}_k = P_k(-) H_k^{\mathrm{T}} [H_k P_k(-) H_k^{\mathrm{T}} + R_k]^{-1}$$

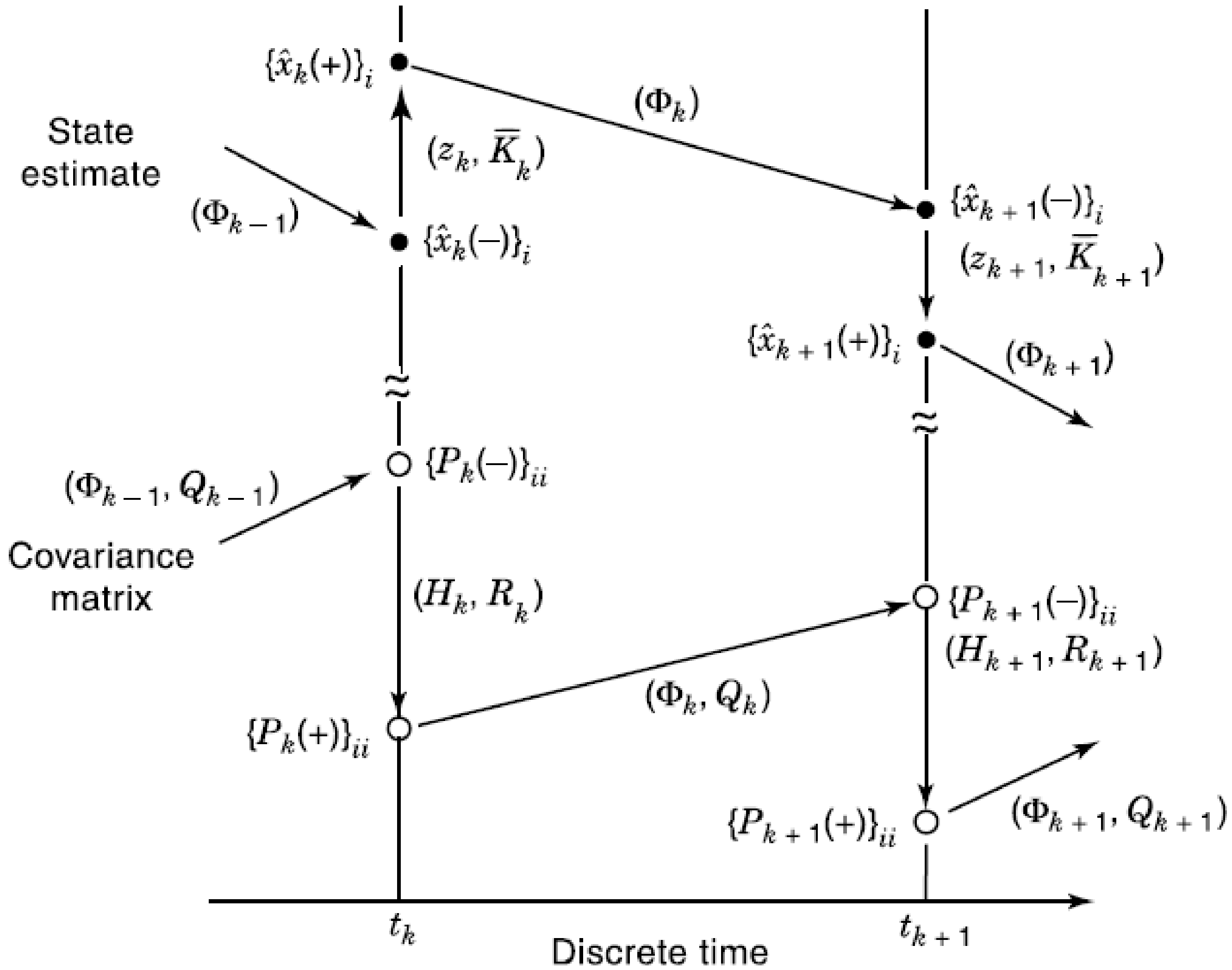
Discrete time Kalman filter equations



Kalman filtering – Kalman filter cycle



Progressions in the Kalman cycle



Extended Kalman Filter – EKF

Nonlinear model of the system:

$$\boldsymbol{x}_k = \boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}) + \boldsymbol{w}_{k-1}\,,\ \boldsymbol{w}_k \sim N(0, \boldsymbol{Q}_k)$$

Nonlinear measurement model:

$$\boldsymbol{z}_k = \boldsymbol{h}_k(\boldsymbol{x}_k) + \boldsymbol{v}_k\,,\ \boldsymbol{v}_k \sim N(0, \boldsymbol{R}_k)$$

Extrapolation of the system state estimation:

$$\hat{\boldsymbol{x}}_k(-) = \boldsymbol{f}_{k-1}\left(\hat{\boldsymbol{x}}_{k-1}(+)\right)$$

Measurement prediction:

$$\hat{\boldsymbol{z}}_k = \boldsymbol{h}_k\left(\hat{\boldsymbol{x}}_k(-)\right)$$

Linearized equations (with respect to the previous system state):

$$Jacobian\ computation\colon \boldsymbol{\Phi}_{k-1}^{[1]} \approx \left.\frac{\partial \boldsymbol{f}_{k-1}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k-1}(-)}$$

$$Jacobian\ computation\colon \boldsymbol{H}_k^{[1]} \approx \left.\frac{\partial \boldsymbol{h}_k}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_k(-)}$$

A priori covariance matrix estimation:

$$\boldsymbol{P}_k(-) = \boldsymbol{\Phi}_{k-1}^{[1]} \boldsymbol{P}_{k-1}(+) \boldsymbol{\Phi}_{k-1}^{[1]\mathrm{T}} + \boldsymbol{Q}_{k-1}$$

Kalman gain computation:

$$\boldsymbol{K}_k = \boldsymbol{P}_k(-)\boldsymbol{H}_k^{[1]\mathrm{T}} \left[\boldsymbol{H}_k^{[1]} \boldsymbol{P}_k(-) \boldsymbol{H}_k^{[1]\mathrm{T}} + \boldsymbol{R}_k\right]^{-1}$$

System state estimation:

$$\hat{\boldsymbol{x}}_k(+) = \hat{\boldsymbol{x}}_k(-) + \boldsymbol{K}_k[\boldsymbol{z}_k - \hat{\boldsymbol{z}}_k]$$

A posteriori covariance matrix estimation:

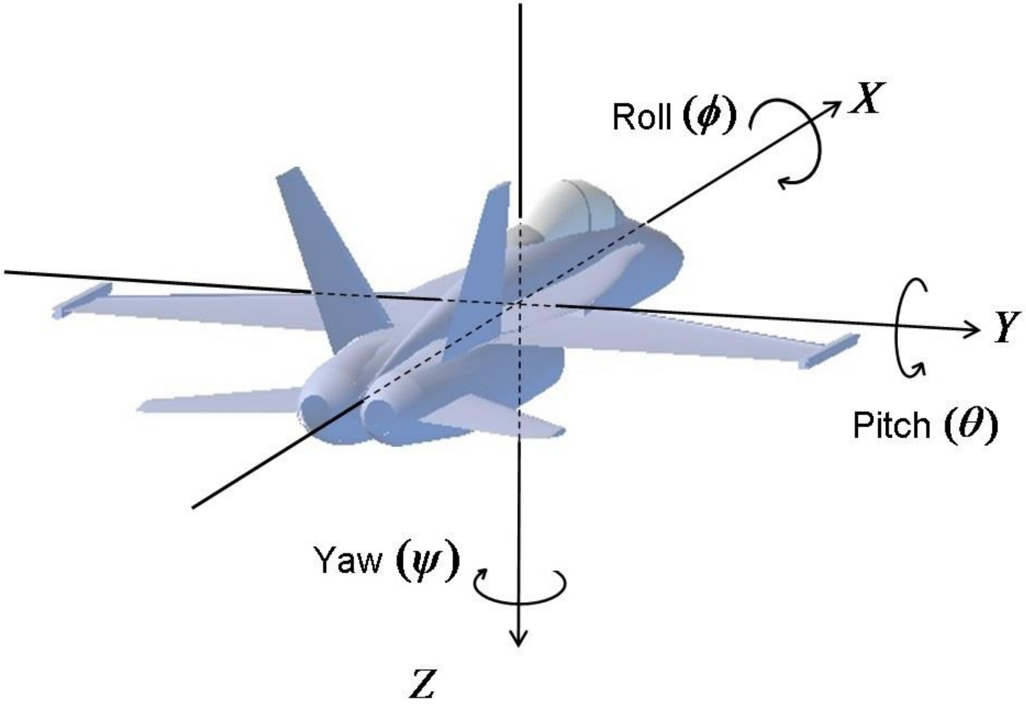
$$\boldsymbol{P}_k(+) = \left[\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k^{[1]}\right] \boldsymbol{P}_k(-)$$

Navigation in local-level coordinates – attitude vs. angular rates

$$\dot{\boldsymbol{C}}_b^n = \boldsymbol{C}_b^n(\boldsymbol{\omega}_{ib}^b \times) - (\boldsymbol{\omega}_{in}^n \times)\boldsymbol{C}_b^n$$

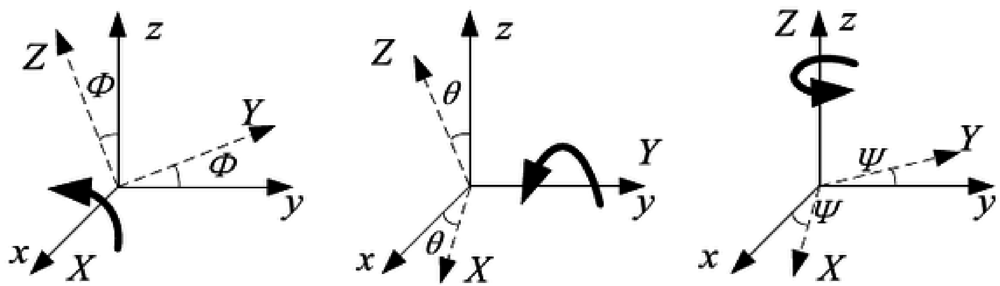
$$\Downarrow$$

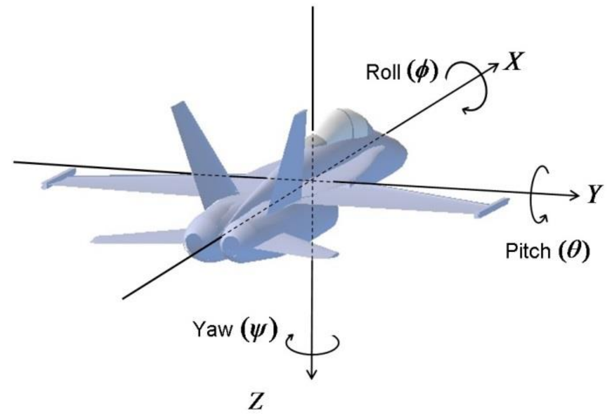
$\boldsymbol{C}_b^n$ - DCM – Direction cosine matrix



DCM - attitude vs. angular rates – for low-cost systems

$$\dot{\boldsymbol{C}}_b^n = \boldsymbol{C}_b^n(\boldsymbol{\omega}_{ib}^b \times)$$





$$\boldsymbol{x}^b = \boldsymbol{A}_x \boldsymbol{A}_y \boldsymbol{A}_z\, \boldsymbol{x}^r$$

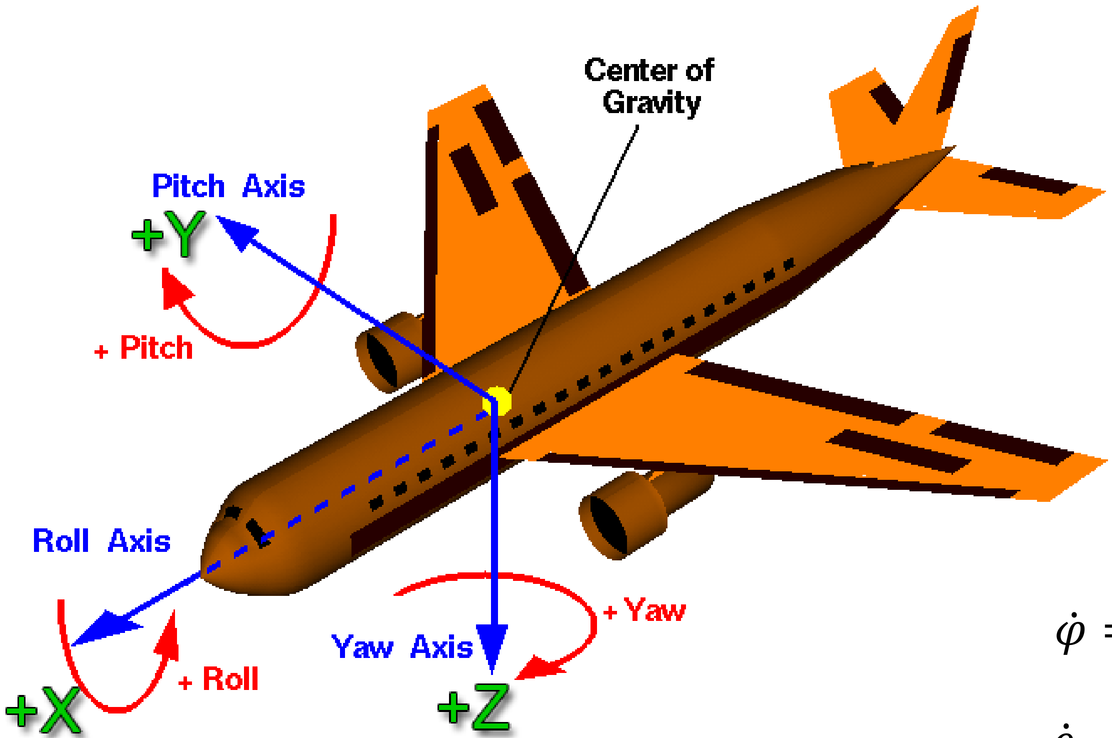
$$\mathbf{A}_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$

$$\mathbf{A}_Y = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$\mathbf{A}_Z = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$C_b^n = \begin{bmatrix} c\theta c\psi & -c\phi s\psi + s\phi s\theta c\psi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & -s\phi c\psi + c\phi s\theta s\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}$$

Navigation principles – strapdown mechanization - attitude



$$\omega_x = \dot{\psi}\ \sin\theta + \dot{\varphi}$$

$$\omega_y = -\dot{\psi}\sin\varphi\cos\theta + \dot{\theta}\cos\varphi$$

$$\omega_z = \dot{\psi}\cos\varphi\cos\theta + \dot{\theta}\sin\varphi$$

$$\Downarrow$$

$$\dot{\varphi} = \omega_x + \omega_y sin\varphi tan\theta + \ \omega_z cos\varphi tan\theta$$

$$\dot{\theta} = \omega_y cos\varphi \ - \ \omega_z sin\varphi$$

$$\dot{\psi} = \omega_y \frac{sin\varphi}{cos\theta} + \ \omega_z \frac{cos\varphi}{cos\theta}$$

State vector & differential equations

State-space modeling

$$\dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu} + \boldsymbol{Gw}$$

$$y = Cx + v$$

Example of a state vector for attitude estimation

$$x = \begin{bmatrix} \varphi & \theta & \psi & \omega_x & \omega_y & \omega_z & b_{\omega x} & b_{\omega y} & b_{\omega z} \end{bmatrix}^T$$

Attitude time update in a continuous form

$$\dot{\varphi} = \omega_x + \omega_y sin\varphi tan\theta + \omega_z cos\varphi tan\theta$$

$$\dot{\theta} = \omega_y cos\varphi - \omega_z sin\varphi$$

$$\dot{\psi} = \omega_y \frac{sin\varphi}{cos\theta} + \omega_z \frac{cos\varphi}{cos\theta}$$

$$\dot{\omega} = -\frac{1}{\tau}\omega + \frac{1}{\tau}w, \text{where } \tau \text{ is a correlation time}$$

$$\dot{b} = 0 + w$$

Covariance matrix of the process

$$Q = \begin{bmatrix} 0_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & \sigma^2{}_{\omega} \times I_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & \sigma^2{}_{b} \times I_{3x3} \end{bmatrix}$$

Measurement vector and its relation to expected measurements

State-space modeling

$$\dot{x} = Ax + Bu + Gw$$

$$\boldsymbol{y = Cx + v}$$

Measurement vector can be formed as

$$z = [f_x \quad f_y \quad f_z \quad \omega_x \quad \omega_y \quad \omega_z \quad m_x \quad m_x \quad m_x]^T$$

Relation between the state vector and the measurement = ***C*** matrix

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}_{body} = 0 - C_n^b \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow \begin{aligned} f_x &= +g \times \sin(\theta) \\ f_y &= -g \times \sin(\varphi)\cos(\theta) \\ f_z &= -g \times \cos(\varphi)\cos(\theta) \end{aligned}$$

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}_{meas} = \omega_{est.} + bias$$

$$C_n^b = \begin{bmatrix} c\theta c\psi & -c\varphi s\psi + s\varphi s\theta c\psi & s\varphi s\psi + c\varphi s\theta c\psi \\ c\theta s\psi & c\varphi c\psi + s\varphi s\theta s\psi & -s\varphi c\psi + c\varphi s\theta s\psi \\ -s\theta & s\varphi c\theta & c\varphi c\theta \end{bmatrix}$$

Transformation matric C_n^b is formed by estimated Euler angles.

Measurement vector and its relation to expected measurements

Magnetometer readings should be treated the same way as ACC readings; nevertheless, it is necessary to calculate a reference magnetic vector. Just be aware that via magnetometer based compensation there should not be affected roll and pitch angles' estimates, only azimuth/yaw. It means that in the Earth's magnetic field due to nearby power lines or large magnetic objects cannot have a direct effect on the pitch and roll estimates. Therefore, rotation of the measured magnetic vector from the body frame to the reference frame needs to be then modified to share the same inclination as the measured vector (declination is considered to be 0).

$$\begin{bmatrix} m_{xv} \\ m_{yv} \\ m_{zv} \end{bmatrix} = C_b{}^n \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} \rightarrow m_{ref} = \begin{bmatrix} m_{rx} \\ m_{ry} \\ m_{rz} \end{bmatrix} = \begin{bmatrix} \sqrt{m_{xv}{}^2 + m_{yv}{}^2} \\ 0 \\ m_{zv} \end{bmatrix}$$

$$\begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix}_{meas} = C_n{}^b \times m_{ref}$$

This approach limits the magnetometer to influence the yaw estimate only.

Shaping filters

For many physical systems encountered in practice, it may not be justified to assume that all noises are white Gaussian noise processes. Thus it can be useful to generate an autocorrelation function or PSD from real data and develop an appropriate noise model using differential/difference equations.

↓

Shaping filters

↓

Shaping filters are driven by white noise with a flat spectrum and they shape the spectrum into the one obtained by evaluating real data processes.

↓

The state vector can be then "augmented" by the state vector components of the shaping filter with a resulting model having the form of a linear dynamic system driven by white noise.

Assignment – a constant estimation – models available

Noise Type	Autocorrelation Function ψ_x Power Spectral Density Ψ_x	State-Space Formulation And Model	
White noise	$\psi_x(\tau)=\sigma^2\delta^2(\tau)$ $\Psi_x(\omega)=\sigma^2$	Always treated as measurement noise	
Random walk	$\psi_x(\tau)=(undefined)$ $\Psi_x(\omega)\propto\sigma^2/\omega^2$	$\dot{x}=w(t)$ $\sigma_x^2(0)=0$	$x_k=x_{k-1}+w_{k-1}$ $\sigma_x^2(0)=0$
Random constant	$\psi_x(\tau)=\sigma^2$ $\Psi_x(\omega)=2\pi\sigma^2(\omega)$	$\dot{x}=0$ $\sigma_x^2(0)=\sigma^2$	$x_k=x_{k-1}$ $\sigma_x^2(0)=\sigma^2$
Harmonic	$\psi_x(\tau)=\sigma^2\cos(\omega_0\tau)$ $\Psi_x(\omega)=\pi\sigma^2\delta(\omega-\omega_0)$ $+\pi\sigma^2\delta(\omega+\omega_0)$	$\dot{x}=\begin{bmatrix}0 & 1\\ -\omega_0^2 & 0\end{bmatrix}x \quad P(0)=\begin{bmatrix}\sigma^2 & 0\\ 0 & 0\end{bmatrix}$	
Exponentially correlated (1^{st} order GM process)	$\psi_x(\tau)=\sigma^2e^{-\alpha\|\tau\|}$ $\Psi_x(\omega)=\frac{2\sigma^\alpha\alpha}{\omega^2+\alpha^2}$	$\dot{x}=-\alpha x+\sigma\sqrt{2\alpha}w(t)$ $\sigma_x^2(0)=\sigma^2$	$x_k=e^{-\alpha}x_{k-1}$ $+\sigma\sqrt{1-e^{-2\alpha}}w_{k-1}$ $\sigma_x^2(0)=\sigma^2$
p^{th} order GM process	$\psi_x(\tau)=\sigma^2e^{-\alpha_p\|\tau\|}\sum_{n=0}^{p-1}\frac{(p-1)!(2\alpha_p\|\tau\|)^{p-n-1}(p+n+1)!}{(2p-2)!n!(p-n-1)!}$ as defined in [2]		

Correlated process noise

$$\dot{x}(t) = F(t)x(t) + G(t)w_1(t), \qquad z(t) = H(t)x(t) + v(t)$$

where $w_1(t)$ is nonwhite, for example, correlated Gaussian noise. As given in the previous section, $v(t)$ is a zero-mean white Gaussian noise. Suppose that $w_1(t)$ can be modeled by a linear shaping filter[10]:

$$\dot{x}_{SF}(t) = F_{SF}(t)x_{SF}(t) + G_{SF}(t)w_2(t)$$
$$w_1(t) = H_{SF}(t)x_{SF}(t)$$

where SF denotes the shaping filter and $w_2(t)$ is zero mean white Gaussian noise. Now define a new augmented state vector

$$X(t) = [x(t)\ x_{SF}(t)]^T$$

$$\boxed{\begin{aligned} \begin{bmatrix} \dot{x}(t) \\ \dot{x}_{SF}(t) \end{bmatrix} &= \begin{bmatrix} F(t) & G(t)H_{SF}(t) \\ 0 & F_{SF}(t) \end{bmatrix} \begin{bmatrix} x(t) \\ x_{SF}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ G_{SF}(t) \end{bmatrix} w_2(t), \\ \dot{X}(t) &= F_T(t)X(t) + G_T(t)w_2(t), \end{aligned}}$$

$$\boxed{\begin{aligned} z(t) &= [H(t) \quad 0] \begin{bmatrix} x(t) \\ x_{SF}(t) \end{bmatrix} + v(t) \\ &= H_T(t)X(t) + v(t). \end{aligned}}$$

Correlated measurement noise

$$\dot{x}(t) = F(t)x(t) + G(t)w(t),$$

$$z(t) = H(t)x(t) + v_1(t).$$

correlated

$$\dot{x}_{SF}(t) = F_{SF}(t)x_{SF}(t) + G_{SF}(t)v_2(t),$$

$$v_1(t) = H_{SF}(t)x_{SF}(t).$$

$$\boxed{\begin{aligned} \begin{bmatrix} \dot{x}(t) \\ \dot{x}_{SF}(t) \end{bmatrix} &= \begin{bmatrix} F(t) & 0 \\ 0 & F_{SF}(t) \end{bmatrix} \begin{bmatrix} x(t) \\ x_{SF}(t) \end{bmatrix} + \begin{bmatrix} G(t) & 0 \\ 0 & G_{SF}(t) \end{bmatrix} \begin{bmatrix} w(t) \\ v_2(t) \end{bmatrix}, \\ z(t) &= [H(t)\ H_{SF}(t)] \begin{bmatrix} x(t) \\ x_{SF}(t) \end{bmatrix}. \end{aligned}}$$

Assignment – a constant estimation

Please, write a Matlab code for a constant estimation via linear Kalman filter when you have available measurements (about 1000 samples or more) related to the choice of the constant but corrupted by noise with a normal distribution, zero mean value and σ^2 defined.

- For different initial conditions (x_0, P_0, Q, R) observe the character of the innovations, progression of P, x_k, and K.
- In the middle of the input series, change to constant to another value and observe the behavior of the KF.

References used:

- M. S. Grewal and A. P. Andrews. Kalman Filtering: Theory and Practice using MATLAB. John Wiley & Sons, 2011.
- Rogers R. M.: Applied mathematics in integrated navigation systems, 3rd edition, AIAA 2007, ISBN: 978-1-56347-927-4