Advanced Signal and Image processing

Ali Mohammad-Djafari

Groupe Problèmes Inverses
Laboratoire des signaux and systèmes (L2S)
UMR 8506 CNRS - SUPELEC - UNIV PARIS SUD 11
Supélec, Plateau de Moulon, 91192 Gif-sur-Yvette, FRANCE.

`djafari@lss.supelec.fr`
`http://djafari.free.fr`
`http://www.lss.supelec.fr`

Wuhan University, September 2012

September 2012

Content

1. Introduction: Signals and Images, Linear transformations (Convolution, Fourier, Laplace, Hilbert, Radon, ..., Discrete convolution, Z transform, DFT, FFT, ...)
2. Modeling: parametric and non-parametric, MA, AR and ARMA models, Linear prediction
3. Deconvolution and Parameter Estimation: Deterministic (LS, Regularization) and Probabilistic methods (Wiener Filtering)
4. Inverse problems and Bayesian estimation
5. Kalman Filtering and smoothing
6. Case study: Signal deconvolution
7. Case study: Image restoration
8. Case study: Image reconstruction and Computed Tomography

1. Introduction

1. Representation of signals and images
2. Linear Transformations
3. Convolution
4. Fourier Transform (FT)
5. Laplace Transform (LT)
6. Hilbert, Melin, Abel, ...
7. Radon Transform (RT)
8. Link between Different Linear Transforms
9. Discrete signals and transformations
10. Discrete convolution, Z Transform, DFT, FFT

2. Modeling: parametric and non-parametric, MA, AR and ARMA models

- Modeling ? for what ?
- Deterministic / Probalistic modeling
- Parametric / Non Parametric
- Moving Average (MA)
- Autoregressive (AR)
- Autoregressive Moving Average (ARMA)
- Classical methods for parameter estimation (LS, WLS)

3. Deconvolution and Parameter estimation

- Signal Deconvolution and Image Restoration
- Least Squares method (LS)
- Limitations of LS methods
- Regularization methods
- Parametric modeling
- Examples:
    - Sinusoids in the noise (MUSIC)
    - Antenna Array Processing
    - Mixture models (Gaussian and Cauchy)

4. Inverse problems and Bayesian estimation

- Inverse problems? Why need for Bayesian approach?
- Probability ?
- Discrete and Continuous variables
- Bayes rule and Bayesian inference
- Maximum Likelihood method (ML) and its limitations
- Bayesian estimation theory
- Bayesian inference for inverse problems in signal and image processing
- Prior modeling
- Bayesian computation (Laplace approximation, MCMC, BVA)

5. Kalman Filtering and smoothing

- Dynamical systems and state space modeling
- State space modeling examples
    - Electrical circuit example
    - Radar tracking of object
- Kalman filtering basics
- Kalman Filtering as recursive Bayesian estimation
- Kalman Filtering extensions: Adaptive signal processing
- Kalman Filtering extensions: Fast Kalman filtering
- Kalman Filtering for signal deconvolution

6. Case study: Signal deconvolution

- Convolution, Identification and Deconvolution
- Forward and Inverse problems: Well-posedness and Ill-posedness
- Naîve methods of Deconvolution
- Classical methods: Wiener filtering
- Bayesian approach to deconvolution
- Simple and Blind Deconvolution
- Deterministic and probabilistic methods
- Joint source and canal estimation

7. Case study: Image restoration

- Image restoration
- Classical methods: Wiener filtering in 2D
- Bayesian approach to deconvolution
- Simple and Blind Deconvolution
- Practical issues and computational costs

8. Case study: Image reconstruction and Computed Tomography

- X ray Computed Tomography, Radon transform
- Analytical inversion methods
- Discretization and algebraic methods
- Bayesian approach to CT
- Gauss-Markov-Potts model for images
- 3D CT
- Practical issues and computational costs

Content

1. **Introduction: Signals and Images, Linear transformations (Convolution, Fourier, Laplace, Hilbert, Radon, ..., Discrete convolution, Z transform, DFT, FFT, ...)**
2. Modeling: parametric and non-parametric, MA, AR and ARMA models, Linear prediction
3. Deconvolution and Parameter Estimation: Deterministic (LS, Regularization) and Probabilistic methods (Wiener Filtering)
4. Inverse problems, Regularization and Bayesian estimation
5. Kalman Filtering
6. Case study: Signal deconvolution
7. Case study: Image restoration
8. Case study: Image reconstruction and Computed Tomography

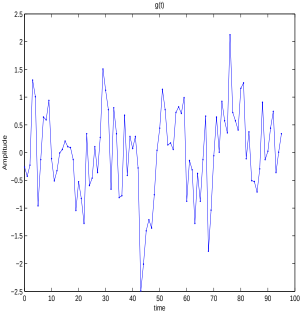
1. Introduction

1. Representation of signals and images
2. Linear Transformations
3. Convolution
4. Fourier Transform (FT)
5. Laplace Transform (LT)
6. Hilbert, Melin, Abel, ...
7. Radon Transform (RT)
8. Link between Different Linear Transforms
9. Discrete signals and transformations
10. Discrete convolution, Z Transform, DFT, FFT

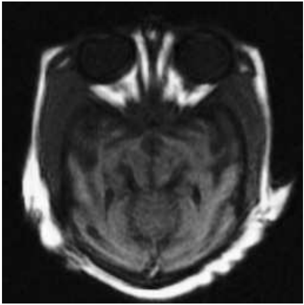
Representation of signals and images

- Signal: $f(t), f(x), f(\nu)$
  - $f(t)$ Variation of temperature in a given position as a function of time t
  - $f(x)$ Variation of temperature as a function of the position x on a line
  - $f(\nu)$ Variation of temperature as a function of the frequency ν
- Image: $f(x,y), f(x,t), f(\nu,t), f(\nu_1,\nu_2)$
  - $f(x,y)$ Distribution of temperature as a function of the position (x,y)
  - $f(x,t)$ Variation of temperature as a function of x and t
  - ...
- 3D, 3D+t, 3D+ν, ... signals: $f(x,y,z), f(x,y,t), f(x,y,z,t)$
  - $f(x,y,z)$ Distribution of temperature as a function of the position (x,y,z)
  - $f(x,y,z,t)$ Variation of temperature as a function of (x,y,z) and t
  - ...

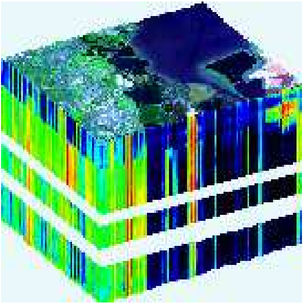
Representation of signals



1D signal



2D signal=image



3D signal

Linear Transformations

$$g(\mathbf{s}) = \int_D f(\mathbf{r})\, h(\mathbf{r},\mathbf{s})\, \mathrm{d}\mathbf{r}$$

$$f(\mathbf{r}) \longrightarrow \boxed{h(\mathbf{r},\mathbf{s})} \longrightarrow g(\mathbf{s})$$

- 1–D :

$$g(t) = \int_D f(t')\, h(t,t')\, \mathrm{d}t'$$

$$g(x) = \int_D f(x')\, h(x,x')\, \mathrm{d}x'$$

- 2–D :

$$g(x,y) = \iint_D f(x',y')\, h(x,y;x',y')\, \mathrm{d}x'\, \mathrm{d}y'$$

$$g(r,\phi) = \iint_D f(x,y)\, h(x,y;r\phi)\, \mathrm{d}x\, \mathrm{d}y$$

Linear and Invariant systems: convolution

$$h(\mathbf{r},\mathbf{r}') = h(\mathbf{r}-\mathbf{r}')$$

$$f(\mathbf{r}) \longrightarrow \boxed{h(\mathbf{r})} \longrightarrow g(\mathbf{r}) = h(\mathbf{r}) * f(\mathbf{r})$$

- 1–D :

$$g(t) = \int_D f(t')\, h(t-t')\, \mathrm{d}t'$$

$$g(x) = \int_D f(x')\, h(x-x')\, \mathrm{d}x'$$

- 2–D :

$$g(x,y) = \iint_D f(x,y)\, h(x-x',y-y')\, \mathrm{d}x'\, \mathrm{d}y'$$

- $h(t)$ impulse response
- $h(x,y)$ Point Spread Function

Linear Transformations: Separable systems

$$g(\mathbf{s}) = \int_D f(\mathbf{r})\, h(\mathbf{r},\mathbf{s})\, \mathrm{d}\mathbf{r}$$

$$h(\mathbf{r},\mathbf{s}) = \prod_j h_j(r_j, s_j)$$

Examples:

- 2D Fourier Transform

$$g(\omega_x,\omega_y) = \iint f(x,y)\, \exp\left\{-j(\omega_x x + \omega_y y)\right\}\, \mathrm{d}x\, \mathrm{d}y$$

$$h(x,y,\omega_x,\omega_y) = h_1(\omega_x x)\, h_2(\omega_y y)$$

$$\exp\left\{-j(\omega_x x + \omega_y y)\right\} = \exp\left\{-j(\omega_x x)\right\}\, \exp\left\{-j(\omega_y y)\right\}$$

- *n*D Fourier Transform

$$g(\boldsymbol{\omega}) = \int f(\mathbf{x})\, \exp\left\{-j\boldsymbol{\omega}'\mathbf{x}\right\}\, \mathrm{d}\mathbf{x}$$

Fourier Transform

[Joseph Fourier, French Mathematicien (1768-1830)]

- 1D Fourier: $\mathcal{F}_1$

$$\left\{\begin{array}{ll} g(\omega) & = \displaystyle\int f(t) \exp\{-j\omega t\}\; \mathrm{d}t \\ f(t) & = \frac{1}{2\pi}\displaystyle\int g(\omega) \exp\{+j\omega t\}\; \mathrm{d}\omega \end{array}\right.$$

- 2D Fourier: $\mathcal{F}_2$

$$\left\{\begin{array}{ll} g(\omega_x,\omega_y) & = \displaystyle\iint f(x,y) \exp\{-j(\omega_x x + \omega_y y\}\; \mathrm{d}x\, \mathrm{d}y \\ f(x,y) & = (\frac{1}{2\pi})^2 \displaystyle\iint g(\omega_x,\omega_y) \exp\{+j(\omega_x x + \omega_y y\}\; \mathrm{d}\omega_x\, \mathrm{d}\omega_y \end{array}\right.$$

- nD Fourier: $\mathcal{F}_n$

$$\left\{\begin{array}{ll} g(\boldsymbol{\omega}) & = \displaystyle\int f(\mathbf{x}) \exp\{-j\boldsymbol{\omega}'\mathbf{x}\}\; \mathrm{d}\mathbf{x} \\ f(\mathbf{x}) & = (\frac{1}{2\pi})^n \displaystyle\int g(\boldsymbol{\omega}) \exp\{+j\boldsymbol{\omega}'\mathbf{x}\}\; \mathrm{d}\boldsymbol{\omega} \end{array}\right.$$

1D Fourier Transform $\mathcal{F}_1$

$$\left\{\begin{array}{ll} g(\omega) & = \int f(t) \exp\{-j\omega t\} \, \mathrm{d}t \\ f(t) & = \frac{1}{2\pi} \int g(\omega) \exp\{+j\omega t\} \, \mathrm{d}\omega \end{array}\right.$$

- $|g(\omega)|^2$ is called the spectrum of the signal $f(t)$
- For real valued signals $f(t)$, $|g(\omega)|$ is symetric

Examples:

$f(t)$	$g(\omega)$
$\exp\{-j\omega_0 t\}$	$\delta(\omega-\omega_0)$
$\sin(\omega_0 t)$	?
$\cos(\omega_0 t)$	?
$\exp\{-t^2\}$	?
$\exp\{-\frac{1}{2}(t-m)^2/\sigma^2\}$	?
$\exp\{-t/\tau\}, t>0$	?
$1 \quad \text{if } \|t\| < T/2$	?

2D Fourier Transform: $\mathcal{F}_2$

$$\left\{\begin{array}{ll} g(\omega_x,\omega_y) & = \iint f(x,y) \exp\left\{-j(\omega_x x + \omega_y y\right\} \, \mathrm{d}x \, \mathrm{d}y \\ f(x,y) & = (\frac{1}{2\pi})^2 \iint g(\omega_x,\omega_y) \exp\left\{+j(\omega_x x + \omega_y y\right\} \, \mathrm{d}\omega_x \, \mathrm{d}\omega_y \end{array}\right.$$

- $|g(\omega_x,\omega_y)|^2$ is called the spectrum of the image $f(x,y)$
- For real valued image $f(x,y)$, $|g(\omega_x,\omega_y)|$ is symetric with respect of the two axis ω_x and ω_y.

Examples:

$f(x,y)$	$g(\omega_x,\omega_y)$
$\exp\left\{-j(\omega_{x0}x + \omega_{y0}y)\right\}$	$\delta(\omega_x - \omega_{x0})\delta(\omega_y - \omega_{y0})$
$\exp\left\{-(x^2+y^2)\right\}$	?
$\exp\left\{-\frac{1}{2}[(x-m_x)^2/\sigma_x^2 + (y-m_y)^2/\sigma_y^2]\right\}$	?
$\exp\left\{-(\|x\|+\|y\|)\right\}$	?
1 if $\|x\| < T_x/2$ & $\|y\| < T_y/2$	?
1 if $(x^2+y^2) < a$	?

nD Fourier Transform: $\mathcal{F}_n$

$$\left\{\begin{array}{ll} g(\boldsymbol{\omega}) & = \int f(\mathbf{x}) \exp\left\{-j\boldsymbol{\omega}'\mathbf{x}\right\}\, \mathrm{d}\mathbf{x} \\ f(\mathbf{x}) & = (\frac{1}{2\pi})^n \int g(\boldsymbol{\omega}) \exp\left\{+j\boldsymbol{\omega}'\mathbf{x}\right\}\, \mathrm{d}\boldsymbol{\omega} \end{array}\right.$$

- $|g(\boldsymbol{\omega})|^2$ is called the spectrum of $f(\mathbf{x})$
- For real valued image $f(\mathbf{x})$, $|g(\boldsymbol{\omega})|$ is symetric with respect of all the axis ω_j.

Examples:

$f(\mathbf{x})$	$g(\boldsymbol{\omega})$
$\exp\left\{-j(\boldsymbol{\omega}_0'\mathbf{x})\right\}$	$\delta(\boldsymbol{\omega}-\boldsymbol{\omega}_0)$
$\exp\left\{-\mathbf{x}'\mathbf{x}\right\} = \exp\left\{-\|\mathbf{x}\|^2\right\}$	$\exp\left\{-\boldsymbol{\omega}'\boldsymbol{\omega}\right\} = \exp\left\{-\|\boldsymbol{\omega}\|^2\right\}$
$\exp\left\{-\|\mathbf{D}\mathbf{x}\|^2\right\}$	?
$\exp\left\{-\frac{1}{2}[(\mathbf{x}-\mathbf{m})'\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\mathbf{m})\right\}$	?
$1 \ \text{ if } \|x\|^2 < R$	?
$1 \ \text{ if } \|x_j\| < R$	?

1D Fourier Transform: Classical definition and properties

Classical Definition for Integrable 1–D signals

$$f(t) \in L_1, \quad \int_{-\infty}^{+\infty} |f(t)| \,\mathrm{d}t < \infty$$

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) \exp\{-j\omega t\} \,\mathrm{d}t$$

Some important properties :

- FT is additive: $f + g \leftrightarrow F(\omega) + G(\omega)$
- FT is homogenious: $cf(t) \leftrightarrow cF(\omega)$
- The spectral amplitude is bounded:

$$|F(\omega)| \leq \int_{-\infty}^{+\infty} |f(t)| \,\mathrm{d}t$$

- $F(\omega)$ is uniformly continuous on $[-\infty, \infty]$
- Rieman–Lebesgue lemma:

$$\lim_{|\omega| \to \infty} |F(\omega)| = 0$$

Convolution

If $f(t), g(t) \in L_1$

$$h(t) = \int_{-\infty}^{+\infty} f(t-\tau)\, g(\tau)\, \mathrm{d}\tau = f * g$$

- The convolution integral $[f * g](t)$ exists almost everywhere for $t \in [-\infty, \infty]$ and, furthermore, $f * g \in L_1$.
- $$\int_{-\infty}^{+\infty} |[f * g](t)|\ \mathrm{d}t \leq \int_{-\infty}^{+\infty} |f(t)|\ \mathrm{d}t \int_{-\infty}^{+\infty} |g(t)|\ \mathrm{d}t$$
- $f * g = g * f$ (commutativity)
- $(f * g) * r = f * (g * r)$ (associativity)
- Convolution and FT: $f(t) * g(t) \leftrightarrow F(\omega) G(\omega)$

FT and Convolution properties

- if $f \in L_1 \cap L_2$, then $F(\omega) \in L_2$ and

$$\int_{-\infty}^{+\infty} |f(t)|^2 \, \mathrm{d}t = \frac{1}{2\pi} \int_{-\infty}^{+\infty} |F(\omega)|^2 \, \mathrm{d}\omega < \infty$$

Lets note by $I_{[-n,n]}(t) = 1, t \in [-n, n]$ and 0 elsewhere

Planchrel Theorem:
Let $f \in L_2$, $f_n(t) = f(t) I_{[-n,n]}(t)$ and $F_n(\omega)$ the FT of $f_n(t)$. Then
:

- There exists a function $F(\omega) \in L_2$ such that $F_n(\omega)$ converges to $F(\omega)$ in the L_2 norm, i.e.,

$$\lim_{n\to\infty} \int_{-\infty}^{+\infty} |F(\omega) - F_n(\omega)|^2 \, \mathrm{d}\omega = 0$$

FT and Convolution properties

- For any $F(\omega) \in L_2$, there exists a unique $f(t) \in L_2$ such that $f(t) \leftrightarrow F(\omega)$, where the double arrow denotes the L_2 Fourier transform.
- If $f \in L_1 \cap L_2$, then

$$F(\omega) = \lim_{n \to \infty} F_n(\omega)$$

  pointwise, and $F(\omega)$ agrees with definition of the FT for L_1 functions.
- For any $f \in L_2$, the Parseval identity holds, i.e.,

$$\int_{-\infty}^{+\infty} |f(t)|^2 \,\mathrm{d}t \frac{1}{2\pi} = \int_{-\infty}^{+\infty} |F(\omega)|^2 \,\mathrm{d}\omega < \infty.$$

FT and Convolution properties

- If $F(\omega) \in L_2$ and we note by

$$F_{[\Omega]}(\omega) = F(\omega) I_{[-\Omega,\Omega]}(\omega),$$

then $F_{[\Omega]}(\omega) \in L_1 \cap L_2$ and we can define

$$f_{[\Omega]}(t) = \mathcal{F}^{-1} F_{[\Omega]}(\omega),$$

and show that $f_{[\Omega]}(t)$ converges to $f(t)$ in the L_2 norm, i.e.,

$$\lim_{\Omega\to\infty} \int_{-\infty}^{+\infty} \left|f(t) - f_{[\Omega]}(t)\right|^2 \, \mathrm{d}t = 0$$

Note the following

$$F_n(\omega) = \int_{-n}^{n} f(t) \exp\left\{-j\omega t\right\} \, \mathrm{d}t$$

$$f_{[\Omega]}(t) = \frac{1}{2\pi} \int_{-\Omega}^{\Omega} F(\omega) \exp\left\{j\omega t\right\} \, \mathrm{d}\omega$$

FT and Convolution properties

If $f, g \in L_2$

$$h(t) = \int_{-\infty}^{+\infty} f(t-x) g(x) \, \mathrm{d}x = f * g$$

- $h(t)$ is bounded, continuous, and converges to zero as t goes to infinity.

Convolution for 2D signals

$$F(\omega_x,\omega_y)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\exp\left\{-j\left(\omega_x x+\omega_y y\right)\right\}\,\mathrm{d}x\,\mathrm{d}y$$

- Rieman–Lebesgue lemma:

$$\lim_{|\omega_x^2+\omega_y^2|\to\infty}|F(\omega_x,\omega_y)|=0$$

$$f(x,y)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(\omega_x,\omega_y)\exp\left\{-j\left(\omega_x x+\omega_y y\right)\right\}\,\mathrm{d}\omega_x\,\mathrm{d}\omega_y$$

If $f,g\in L_1$

$$h(x,y)=\int_{-\infty}^{+\infty} f(x-u,y-v)\,g(u,v)\,\mathrm{d}u\,\mathrm{d}v=f*g$$

Sine and Cosine Transforms

Sine Transform: For un odd function $f(-t) = -f(t)$

$$S(\omega) = 2\int_0^\infty f(t)\sin(\omega t)\;\mathrm{d}t$$

$$f(t) = \frac{1}{\pi}\int_0^\infty S(\omega)\sin(\omega t)\;\mathrm{d}\omega$$

$$S(\omega) = \frac{1}{j}[F(\omega) - F(-\omega)]$$

Cosine Transform: For un even function $f(-t) = f(t)$

$$C(\omega) = 2\int_0^\infty f(t)\cos(\omega t)\;\mathrm{d}t$$

$$f(t) = \frac{1}{\pi}\int_0^\infty C(\omega)\cos(\omega t)\;\mathrm{d}\omega$$

$$C(\omega) = F(\omega) + F(-\omega)$$

Laplace Transforms: $\mathcal{L}$

[Pierre-Simon Laplace, French Mathematicien (1749-1827)]
Let $f(t)$ be a signal with support in $[0,\infty)$ such that $\exp\{-kt\}\, f(t) \in L_1$ for some real number k.

$$F(s) = \int_0^\infty f(t)\, \exp\{-s\, t\}\, \mathrm{d}t$$

- $F(s)$ is defined at least in the right half of the complex plane defined by $\text{Real}\,(s) > k$.
- When the inversion conditions for the FT hold, we also have an inversion for the LT given by

  $$f(t) = \frac{1}{j2\pi} \int_{a-j\infty}^{a+j\infty} F(s) \exp\{+st\}\, \mathrm{d}s, \quad \forall t > 0$$

  where $a > k$ is a real number such that $\exp\{-kt\}\, f(t) \in L_1$
- Suppose $f(t)$ and $g(t)$ have support in $[0,\infty)$, $\exp\{-k_1 t\}\, f(t)$ and $\exp\{-k_2 t\}\, g(t)$ are in L_1, $f \leftrightarrow F$, and $g \leftrightarrow G$. Then

Laplace Transform: few examples

$f(t)$	$g(s)$
t	$1/s$
$\exp\{as\}$	$1/(t-a)$
$\sin\omega t$	$\omega/(t^2+\omega^2)$

Bilateral Laplace Transform

- Definition:

$$L(s) = \int_{-\infty}^{+\infty} f(t) \exp\left\{-s\,t\right\}\,\mathrm{d}t$$

- Whereas the LT integral converges in half–planes $\text{Real}\left(s\right) > k$, the bilateral LT integral converges in infinite strips $k_1 < \text{Real}\left(s\right) < k_2$

$$f(t) = \frac{1}{j2\pi} \int_{a-j\infty}^{a+j\infty} L(s) \exp\left\{+st\right\}\,\mathrm{d}s$$

  where a is a real number $k_1 > a > k_2$, and k_1, k_2 are such that $\exp\left\{-k_1 t\right\} f(t) \in L_1$ and $\exp\left\{-k_2 t\right\} f(t) \in L_1$.

Mellin Transform (MT)

[Hjalmar Melin, Finnish mathematicien (1854-1933)]

- Definition:
  If $t^{\text{Real}(s)-1} f(t) \in L_1$ on $(0,\infty)$,

$$g(s) = \int_0^{\infty} f(t)\, t^{s-1}\, \mathrm{d}t$$

$$f(t) = \frac{1}{j2\pi} \int_{a-j\infty}^{a+j\infty} g(s)\, t^{-s}\, \mathrm{d}s$$

  where $s_1 < a < s_2$.

- Convolution for MT involves a product–type kernel and is given by

$$h(t) = \int_0^{t} f(x)\, g(t/x)\, x^{-1}\, \mathrm{d}x, \quad t > 0$$

$$H(s) = F(s)\, G(s)$$

Abel Transform (AT)

[Niels Henrik Abel, Norwegian mathematician (1802-1829)]

- Definition: If $f(t) \in L_1$,

$$\begin{aligned} g(s) &= 2 \int_s^\infty \frac{f(r)\, r}{r^2 - s^2} \, \mathrm{d}r \\ f(r) &= \frac{-1}{\pi} \int_r^\infty \frac{\mathrm{d}g(s)/\,\mathrm{d}s}{s^2 - r^2} \, \mathrm{d}s \end{aligned}$$

- Generalizations

$$A^\alpha(x) = \frac{1}{\Gamma(\alpha)} \int_a^x f(t)\,(x-t)^{\alpha-1} \, \mathrm{d}t$$

- For $\alpha = n$, a non negative integer, we have

$$A^n(x) = \int_a^x \int_a^{x_1} \cdots \int_a^{x_{n-1}} f(x_n) \, \mathrm{d}x_n \cdots \, \mathrm{d}x_1$$

Abel Transform (AT)

- For $\mathrm{Real}\,(\alpha) > 0$ and $\mathrm{Real}\,(\beta) > 0$

$$A^{\alpha}\left(A^{\beta}(f)\right) = A^{\alpha+\beta}(f)$$

$$A^{1}(f) = \int_{0}^{x} f(t)\,\mathrm{d}t$$

- If f is continuous

$$A^{0}(f) = f$$

Hilbert Transform: $\mathcal{H}$

[David Hilbert, German mathematicien (1862-1943)]

- Definition: If $f \in L_2$ on $(-\infty, \infty)$,

$$g(x) = \frac{1}{\pi} \int_{-\infty}^{+\infty} \frac{f(t)}{t-x} \, \mathrm{d}t$$

$$f(t) = \frac{-1}{\pi} \int_{-\infty}^{+\infty} \frac{g(x)}{x-t} \, \mathrm{d}x$$

  The integrals are interpreted in the Cauchy principal value(CPV) sense at $t = x$.

- Alternate expression useful in signal processing:

$$g(t) = \frac{1}{\pi} \lim_{\epsilon \mapsto 0} \int_{\epsilon}^{\infty} \frac{f(t+\tau) - f(t-\tau)}{\tau} \, \mathrm{d}\tau$$

Hilbert Transform: $\mathcal{H}$

- If $f \in L_2$
  - $\mathcal{H}\left(\mathcal{H}(f)\right) = f$
  - f and $\mathcal{H}(f)$ are orthogonal, i.e.,

$$\lim_{r\to\infty} \int_{-r}^{r} [f\mathcal{H}(f)](u) \, \mathrm{d}u = 0$$

  - The Hilbert transform of a constant is zero.
  - Hilbert and Fourier Transforms

$$\mathcal{H}(f) = f * \frac{-1}{\pi t} \quad \longrightarrow \mathcal{F}\{\mathcal{H}(f)\} = j\mathrm{sgn}(\omega)F(\omega)$$

Radon Transform (RT): $\mathcal{R}$

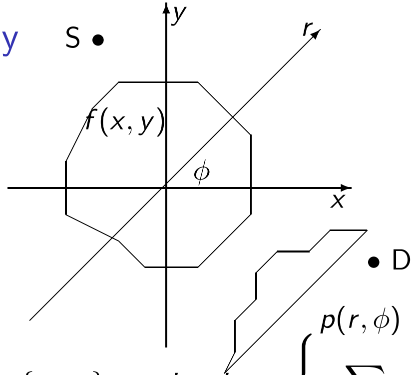
- Definition:
  This transform is defined for the functions in 2 or more dimensions. Here we give the relations only in the 2–D case.

$$
\begin{aligned}
R(r,\phi) &= \int_{L_{r,\phi}} f(x,y)\,\mathrm{d}l \\
&= \iint f(x,y)\,\delta(r - x\cos(\phi) - y\sin(\phi))\,\mathrm{d}x\,\mathrm{d}y
\end{aligned}
$$

- The Radon transform maps the spatial domain $(x,y) \in R^2$ to the domain $(r,\phi) \in R \times [0,\pi]$. Each point in the (r,ϕ) space corresponds to a line in the spatial domain (x,y).
- Note that (r,ϕ) are not the polar coordinates of (x,y). In fact if we note the polar coordinates corresponding to the (x,y) point (ρ,θ), then we have

$$
x = \rho\cos\theta, \quad y = \rho\sin\theta, \quad r = \rho\cos(\phi-\theta)
$$

X ray Tomography



$$I = I_0 \exp\{-\mu x\} \longrightarrow I = I_0 \exp\left\{-\sum_j \mu_j x_j\right\},$$

$$I = I_0 \exp\left\{-\int_L f(x)\,\mathrm{d}x\right\}. \longrightarrow -\ln\frac{I}{I_0} = \int_L f(x,y)\,\mathrm{d}l,$$

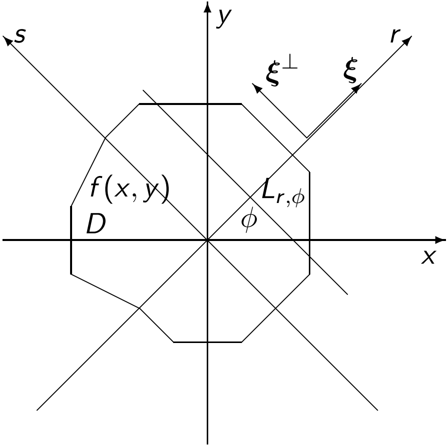
$$p(r,\phi) = \int_L f(x,y)\,\mathrm{d}l = \iint f(x,y)\delta(r - x\cos\phi - y\sin\phi)\,\mathrm{d}x\,\mathrm{d}y$$

$$f(x,y) \longrightarrow \boxed{\text{Radon Transform}} \longrightarrow p(r,\phi)$$

$$p(r,\phi) \longrightarrow \boxed{\text{Image Reconstruction}} \longrightarrow f(x,y)$$

X ray Tomography: Radon Transform

- **Deux dimensions :**



$$\boldsymbol{\xi} = [\cos\phi, \sin\phi]', \quad \boldsymbol{\xi}^{\perp} = [-\sin\phi, \cos\phi]'$$

$$\mathbf{x} = [x,y]' = [\rho\cos\theta, \rho\sin\theta], \qquad L_{r,\phi} : r = x\cos\phi + y\sin\phi = \boldsymbol{\xi}' \cdot \mathbf{x}$$

X ray Tomography: Radon Transform

$$\begin{aligned}
g(r,\phi) &= \int_L f(x,y)\,\mathrm{d}l = \iint_D f(x,y)\delta(r - x\cos\phi - y\sin\phi)\,\mathrm{d}x\,\mathrm{d}y \\
g(r,\boldsymbol{\xi}) &= \iint f(x,y)\,\delta(r - \boldsymbol{\xi}'\cdot\mathbf{x})\,\mathrm{d}x\,\mathrm{d}y \\
g(r,\phi) &= \int_0^{\infty}\int_0^{2\pi} f(\rho,\theta)\delta\big(r - \rho\cos(\phi-\theta)\big)\rho\,\mathrm{d}\rho\,\mathrm{d}\theta
\end{aligned}$$

n-D case:

$$\mathbf{x} = [x_1, x_2, \ldots, x_n]', \quad \boldsymbol{\xi} = [\xi_1, \xi_2, \ldots, \xi_n]' \quad \text{avec} \quad |\boldsymbol{\xi}| = 1$$

$$\mathrm{d}\mathbf{x} = \mathrm{d}x_1\,\mathrm{d}x_2 \ldots \mathrm{d}x_n, \quad r = \boldsymbol{\xi}'\cdot\mathbf{x} = \xi_1 x_1 + \xi_2 x_2 + \ldots + \xi_n x_n$$

$$\boxed{g(r,\boldsymbol{\xi}) = \int_{R^n} f(\mathbf{x})\,\delta(r - \boldsymbol{\xi}'\cdot\mathbf{x})\,\mathrm{d}\mathbf{x}}$$

Radon Transform: Some properties

[Johann K.A. Radon, Austrian mathematician (1887-1956)]

- Definition in cartezian coordinate system:

$$f(x,y) \xrightarrow{\mathcal{R}} g(r,\phi) = \iint f(x,y)\delta(r-x\cos(\phi)-y\sin(\phi))\,\mathrm{d}x\,\mathrm{d}y$$

- Definition in polar coordinate system:

$$f(\rho,\theta) \xrightarrow{\mathcal{R}} g(r,\phi) = \int_0^{\infty}\int_0^{2\pi} f(\rho,\theta)\delta(r-\rho\cos(\phi-\theta)\rho\,\mathrm{d}\rho\,\mathrm{d}\theta$$

- Inversion

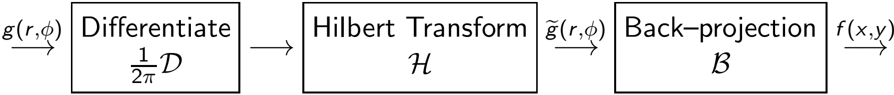
$$f(x,y) = \frac{1}{2\pi}\int_0^{\pi}\int_0^{\infty} \frac{\partial g(r,\phi)/\partial r}{r-x\cos(\phi)-y\sin(\phi)}\,\mathrm{d}r\,\mathrm{d}\phi$$

Radon Transform: Some properties

- Linearity:
  $a\,f_1(x,y) + b\,f_2(x,y) \longrightarrow a\,g_1(r,\phi) + b\,g_2(r,\phi)$
- Symetry:
  $f(x,y) \longrightarrow g(r,\phi) = g(-r,\phi \pm \pi)$
- Periodicity:
  $f(x,y) \longrightarrow g(r,\phi) = g(r,\phi \pm 2k\pi),\ k$ integer
- Shift:
  $f(x-x_0, y-y_0) \longrightarrow g(r - x_0\cos(\phi) - y_0\sin(\phi), \phi)$
- Rotation:
  $f(\rho,\theta+\theta_0) \longrightarrow g(r,\phi+\theta_0)$
- Scaling:
  $$f(a\,x, a\,y) \longrightarrow \frac{1}{|a|} g(ar,\phi), \quad a \neq 0$$
- Mass conservation:
  $$M = \iint_{-\infty}^{\infty} f(x,y)\,\mathrm{d}x dy \longrightarrow M = \int_0^{\pi}\int_{-\infty}^{+\infty} g(r,\phi)\,\mathrm{d}r\,\mathrm{d}\phi$$

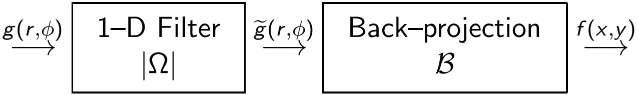
Radon Transform: Inversion

Direct Inverse Radon Transform



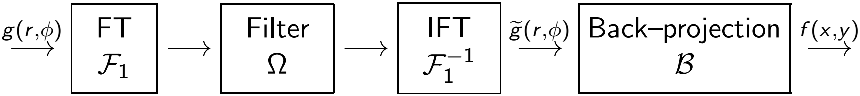
$\xrightarrow{g(r,\phi)}$ Differentiate $\frac{1}{2\pi}\mathcal{D}$ $\longrightarrow$ Hilbert Transform $\mathcal{H}$ $\xrightarrow{\widetilde{g}(r,\phi)}$ Back–projection $\mathcal{B}$ $\xrightarrow{f(x,y)}$

Convolution Back–projection method



$\xrightarrow{g(r,\phi)}$ 1–D Filter $|\Omega|$ $\xrightarrow{\widetilde{g}(r,\phi)}$ Back–projection $\mathcal{B}$ $\xrightarrow{f(x,y)}$

Filter Back–projection method



$\xrightarrow{g(r,\phi)}$ FT $\mathcal{F}_1$ $\longrightarrow$ Filter Ω $\longrightarrow$ IFT $\mathcal{F}_1^{-1}$ $\xrightarrow{\widetilde{g}(r,\phi)}$ Back–projection $\mathcal{B}$ $\xrightarrow{f(x,y)}$

Unitary Transforms: 1D case

$$\begin{aligned} F(u) &= \int f(x)\, h(u,x)\, \mathrm{d}x \\ f(x) &= \int F(u)\, h^*(x,u)\, \mathrm{d}u \end{aligned}$$

$$\begin{aligned} \langle f_1(x), f_2(x) \rangle &= \int f_1(x)\, f_2^*(x)\, \mathrm{d}x \\ \langle F_1(u), F_2(u) \rangle &= \int F_1(u)\, F_2^*(u)\, \mathrm{d}u \end{aligned}$$

$$\begin{aligned} F(u) &= \langle f(x), h_u^*(x) \rangle \\ f(x) &= \langle F(u), h_x(u) \rangle \end{aligned}$$

$$\left\{ \begin{aligned} F(u) &= \langle f(x), h_u^*(x) \rangle \\ f(x) &= \int F(u)\, h_u^*(x)\, \mathrm{d}u \end{aligned} \right.$$

Unitary Transforms: 2D case

$$
\begin{aligned}
F(u,v) &= \int f(x,y)\, h(u,v;x,y)\, \mathrm{d}x\, \mathrm{d}y \\
f(x,y) &= \int F(u,v)\, h^*(x,y;u,v)\, \mathrm{d}u\, \mathrm{d}v
\end{aligned}
$$

$$
\begin{aligned}
\langle f_1(x,y), f_2(x,y)\rangle &= \iint f_1(x,y)\, f_2^*(x,y)\, \mathrm{d}x\, \mathrm{d}y \\
\langle F_1(u,v), F_2(u,v)\rangle &= \iint F_1(u,v)\, F_2^*(u,v)\, \mathrm{d}u\, \mathrm{d}v
\end{aligned}
$$

$$
\begin{aligned}
F(u,v) &= \left\langle f(x,y), h_{u,v}^*(x,y)\right\rangle \\
f(x,y) &= \langle F(u,v), h_{x,y}(u,v)\rangle
\end{aligned}
$$

$$
\left\{
\begin{aligned}
F(u,v) &= \left\langle f(x,y), h_{u,v}^*(x,y)\right\rangle \\
f(x,y) &= \iint F(u,v)\, h_{u,v}(x,y)\, \mathrm{d}x\, \mathrm{d}y
\end{aligned}
\right.
$$

Discrete signals and images

Sampling theory

- Relation between a signal $f(t)$, its samples $f_n = f(n\Delta)$ and sampled signal $f_s(t) = \sum_n f_n \delta(t - n\Delta)$

$$f_s(t) = f(t) \sum_n \delta(t - n\Delta) \longrightarrow F_s(\omega) = F(\omega) * \sum_n \delta(\omega - n/\Delta)$$

- Sampling theorem:
  An Ω band limited signal can be exactly reconstructed from its samples if the sampling interval $\Delta \leq 1/(2\Omega)$.

Discrete systems

- 1–D :

$$g(k) = \sum_{m=0}^{N-1} h(k,m)\, f(m), \quad 0 \leq k \leq N-1$$

- 2–D :

$$g(k,l) = \sum\sum_{m,n=0}^{N-1} h(k,l;m,n)\, f(m,n), \quad 0 \leq k,l \leq N-1$$

◇ **Translation Invariance**

- 1–D :

$$g(k) = \sum_{m=0}^{N-1} h(k-m)\, f(m) = h*f \longrightarrow f(m) \longrightarrow \boxed{h(m)} \longrightarrow g(m)$$

- 2–D :

$$g(k,l) = \sum\sum_{m,n=0}^{N-1} h(k-m,l-n)\, f(m,n) = h*f$$

$$\longrightarrow f(m,n) \longrightarrow \boxed{h(m,n)} \longrightarrow g(m,n)$$

Discrete systems: Z transform

- 1–D :

$$U(z) = \sum_{m=-\infty}^{\infty} u(m)\, z^{-m}$$

$$u(m) = \frac{1}{j2\pi} \oint U(z)\, z^{m-1} \,\mathrm{d}z$$

- 2–D :

$$U(z_1,z_2) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} u(m,n)\, z_1^{-m}\, z_2^{-n}$$

$$u(m,n) = \frac{1}{(j2\pi)^2} \oint \oint U(z_1,z_2)\, z_1^{m-1}\, z_2^{n-1} \,\mathrm{d}z_1 \,\mathrm{d}z_2$$

$$g = h * f \quad \xrightarrow{\mathrm{TZ}} \quad G = H.F$$

h Impulse response
H Transfert function

Unitary Transforms: Vector-Matrix representation

$$\mathbf{v} = \mathbf{A}\mathbf{u} \longrightarrow v(k) = \sum_{m=0}^{N-1} a_k(m)\, u(m), \quad 0 \leq k \leq N-1$$

- Unitary transform: $\mathbf{A}^{-1} = \mathbf{A}^{*t}$

$$\mathbf{u} = \mathbf{A}^{*t}\mathbf{v} \longrightarrow u(m) = \sum_{k=0}^{N-1} v(k)\, a_m^*(k), \quad 0 \leq m \leq N-1$$

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{k=1}^{N} x(k)\, y^*(k) \longrightarrow \left\{ \begin{array}{l} v(k) = \langle \mathbf{u}, \mathbf{a}_k^* \rangle \\ \mathbf{u} = \sum_{k=1}^{N} v(k)\, \mathbf{a}_k^* \end{array} \right.$$

- Basis vector:

$$\mathbf{a}_k^* = \{a_k^*(m)\} \quad \text{column of the matrix} \quad \mathbf{A}^{*t}$$

Unitary Transforms: Vector-Matrix representation

- Orthonormal Basis:

$$\langle \mathbf{a}_k, \mathbf{a}_{k'} \rangle = \sum_{m=0}^{N-1} a_k(m)\, a_{k'}^*(m) = \delta(k-k')$$

$$u_P(m) \overset{\triangle}{=} \sum_{k=0}^{P-1} v(k) a_k^*(m) \quad \text{minimise} \quad \sigma_\epsilon^2 = \sum_{m=0}^{N-1} |u(m) - u_P(m)|^2$$

- Complete Basis:

$$\sum_{k=0}^{N-1} a_k(m) a_k^*(m') = \delta(m-m')$$

$$\sigma_\epsilon^2 = \sum_{m=0}^{N-1} |u(m) - u_P(m)|^2 = 0 \quad \text{if} \quad P = N$$

Unitary Transforms in 2D

$$v(k,l)=\sum\sum_{m,n=0}^{N-1} a_{k,l}(m,n)\,u(m,n),\quad 0\leq k,l\leq N-1$$

- Unitary transforms:

$$u(m,n)=\sum\sum_{k,l=0}^{N-1} v(k,l)\,a^*_{m,n}(k,l),\quad 0\leq m,n\leq N-1$$

- Basis Images:

$$\mathbf{A}^*_{k,l}=\{a^*_{k,l}(m,n)\}$$

- Scalar product:

$$\langle\mathbf{F},\mathbf{G}\rangle=\sum\sum_{m,n=0}^{N-1} f(m,n)\,g^*(m,n)$$

$$v(k,l)=\left\langle\mathbf{U},\mathbf{A}^*_{k,l}\right\rangle,\quad \mathbf{U}=\sum\sum_{k,l=0}^{N-1} v(k,l)\mathbf{A}^*_{k,l}$$

Unitary Transforms in 2D

- Complete Basis:

$$\sum_{k,l=0}^{N-1} a_{k,l}(m,n)\, a_{k,l}^{*}(m',n') = \delta(m-m', n-n')$$

$$\sigma_{\epsilon}^{2} = \sum_{m,n=0}^{N-1} \left|u(m,m) - u_{P,Q}(m,n)\right|^{2} = 0 \quad \text{if} \quad P = Q = N$$

Separable Unitary transforms

$$a_{k,l}(m,n) = a_k(m)\,b_l(n) \overset{\triangle}{=} a(k,m)\,b(l,n)$$

$$\{a_k(m), k = 0, \ldots, N-1\}, \quad \{b_l(n), l = 0, \ldots, N-1\}$$

Two orthonormal and complete basis

$$\mathbf{A}\mathbf{A}^{*t} = \mathbf{A}^{*t}\mathbf{A} = \mathbf{I}, \qquad \mathbf{B}\mathbf{B}^{*t} = \mathbf{B}^{*t}\mathbf{B} = \mathbf{I},$$

$$v(k,l) = \sum\sum_{m,n=0}^{N-1} a(k,m)u(m,n)b(l,n) \Longrightarrow \mathbf{V} = \mathbf{A}\mathbf{U}\mathbf{B}^t$$

$$u(m,n) = \sum\sum_{k,l=0}^{N-1} a^*(k,m)v(k,l)b^*(l,n) \Longrightarrow \mathbf{U} = \mathbf{A}^{*t}\mathbf{V}\mathbf{B}^*$$

$\diamond$ In general, we choose $\mathbf{A} = \mathbf{B}$:

$$\mathbf{V} = \mathbf{A}\mathbf{U}\mathbf{A}^t = \left[\mathbf{A}[\mathbf{A}\mathbf{U}]^t\right]^t$$

$$\mathbf{U} = \mathbf{A}^{*t}\mathbf{V}\mathbf{A}^* = \left[\mathbf{A}[\mathbf{A}\mathbf{V}]^{*t}\right]^t$$

Unitary transforms: Computational cost

- General case:

$$\mathbf{U} = \sum_{k,l=0}^{N-1}\sum v(k,l)\mathbf{A}_{k,l}^{*} \quad \Longrightarrow N^2 \times N^2 = N^4$$

- Separable case:

$$\mathbf{U} = \mathbf{A}_1^{*t}\mathbf{V}\mathbf{A}_2^{*} \quad \Longrightarrow 2N^3$$

- If $\mathbf{A}_1$ and $\mathbf{A}_2$ are separables $\quad \Longrightarrow \log_2 N \times N^2$

Discrete Fourier Transform (DFT)

- Definition:

$$v(k) = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} u(m)\, W_N^{km}, \quad k = 0, \cdots, N-1 \quad \longrightarrow \quad \mathbf{v} = \mathbf{F}\mathbf{u}$$

$$u(m) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} v(k)\, W_N^{-km}, \quad m = 0, \cdots, N{-}1 \quad \longrightarrow \quad \mathbf{u} = \mathbf{F}^{-1}\mathbf{v}$$

with $\quad W_N^{km} = \exp\left\{-j2\pi km/N\right\}$

- When N is a number power of 2, DFT becomes Fast Fourier Transform (FFT).
- In `Matlab` notation: `v=fft(u)` and `u=ifft(v)`

Link between DFT and FT

$$u_e(n) \overset{\triangle}{=} \begin{cases} u(n) & n = 0, \ldots, N-1 \\ 0 & ailleurs \end{cases}$$

$$U_e(\omega) = \sum_{n=-\infty}^{\infty} u_e(n) \exp\{-j\omega n\} = \sum_{n=0}^{N-1} u_e(n) \exp\{-j\omega n\}$$

DFT of $u(n)$:

$$v(k) = \sum_{n=0}^{N-1} u(n) \exp\{-j2\pi kn/N\}$$

$$\Downarrow$$

$$v(k) = U_e(2\pi k/N)$$

Discrete Fourier Transform properties

- Symetry :

$$\mathbf{F}^{-1} = \mathbf{F}^*$$

- Periodic Extension: $u(k)$ and $v(k)$ are periodic

$$u(k+N) = u(k), \quad v(k+N) = v(k)$$

- FFT when N power of 2 : $\Longrightarrow N \log_2 N$ operations
- Real valued signals:

$$v^*(N-k) = v(k) \longrightarrow |v(N/2-k)| = |v(N/2+k)|$$

- **F** is a circulant matrice.
- For any Circulent Matrice **H**:

$$\mathbf{F}\mathbf{H}\mathbf{F}^* = \mathbf{\Lambda} = \operatorname{diag}\left[\lambda_k, k=0,\ldots,N-1\right]$$

and

$$\boldsymbol{\lambda} = \mathbf{F}\mathbf{h} = \text{DFT of the first ligne of } \mathbf{H}$$

Discrete Fourier Transform and Circular Convolution

- Circulent Convolution:

$$y(n) = \sum_{k=0}^{N-1} h_c(n-k)x(k), \quad n = 0, \ldots, N-1$$

$$h_c(n) = h(n \,\mathrm{modulo}\, N)$$

$$\mathrm{DFT}\{y(n)\}_N = \mathrm{DFT}\{h(n)\}_N \, \mathrm{DFT}\{x(n)\}_N$$

Discrete Fourier Transform and Circular Convolution

- Linear Convolution:

$$\{h(n), n = 0, \ldots, N_h - 1\}, \quad \{x(n), n = 0, \ldots, N_x - 1\},$$

$$y(n) = h(n) * x(n)$$

$$\{y(n), n = 0, \ldots, N_h + N_x - 2\}$$

1. Zero padding: $M \geq N_h + N_x - 2$,

$$\widetilde{h}(n) = \begin{cases} h(n) & \text{for} \quad n = 0, \ldots, N_h - 1 \\ 0 & n = N_h, \ldots, M \end{cases}$$

$$\widetilde{x}(n) = \begin{cases} h(n) & \text{for} \quad n = 0, \ldots, N_h - 1 \\ 0 & n = N_h, \ldots, M \end{cases}$$

2. DFT computation :

$$\mathrm{DFT}\{\widetilde{y}(n)\}_M = \mathrm{DFT}\{\widetilde{h}(n)\}_M \, \mathrm{DFT}\{\widetilde{x}(n)\}_M$$

3. Inverse DFT computation:

$$y(n) = \widetilde{y}(n), \quad n = 0, \ldots, N_x + N_h - 2$$

Some References: Signal and image processing

1. V. Smidl, A. Quinn "The Variational Bayes Method in Signal Processing", Springer, ISBN-10 3-540-28819-8 Springer Berlin Heidelberg New York
2. A. K. Jain, "Fundamentals of Digital Image Processing," Prentice Hall, Englewood Cliffs, NJ 07632, (1989).
3. E. R. Dougherty and Ch. R. Giardina "Image Processing-Continuous to Discrete," Prentice Hall, Englewood Cliffs, NJ 07632, Vol. I and II, (1989).
4. R. C. Gonzalez and P. Wintz "Digital Image Processing," Addison–Wesley Publ. Co., (1987).
5. G. Herman, "Image Reconstruction from Projections,"
6. D. E. Dudgeon, R.M. Mersereau, "Multidimensional Digital Signal Processing,"

Some references: Signal and image processing

1. G. E. P. Box, G. C. Tiao, "Bayesian Inference in Statistical Analysis,"
2. A. Blake, A. Zisserman, "Visual Reconstruction,"
3. S. R. Deans "The Radon Transform and some of its Applications,"
4. A.C. Kak, M. Slaney, "Principles of Computerized Tomographic Imaging," IEEE Press, , 1988.
5. A. Tarantola, "Inverse Problem Theory," Elsevier Science Publishers, 1987, Second impression(1988).

Some references: Main Journals

- IEEE transactions on :
    - Signal Processing (SI)
    - Image Processing (SI)
    - Acoustics, Speech and Signal Processing (ASSP)
    - Medical Imaging (MI)
    - Pattern Analysis and Machine Intelligency (PAMI)
- Proceedings of IEEE
- Computer Vision, Graphics, and Image Processing

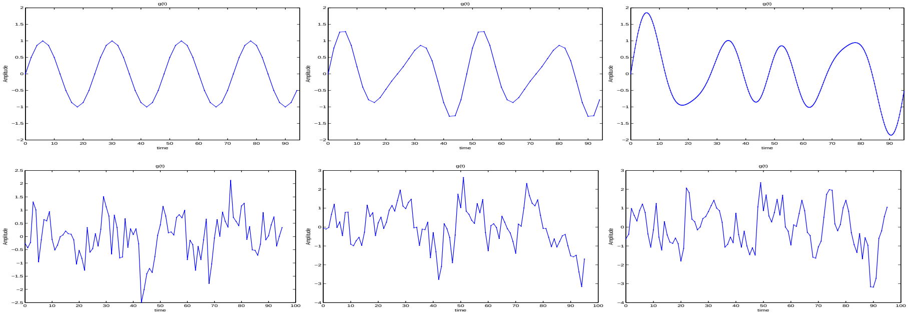
Content

1. Introduction: Signals and Images, Linear transformations (Convolution, Fourier, Laplace, Hilbert, Radon, ..., Discrete convolution, Z transform, DFT, FFT, ...)
2. **Modeling: parametric and non-parametric, MA, AR and ARMA models, Linear prediction**
3. Deconvolution and Parameter Estimation: Deterministic (LS, Regularization) and Probabilistic methods (Wiener Filtering)
4. Inverse problems, Regularization and Bayesian estimation
5. Kalman Filtering
6. Case study: Signal deconvolution
7. Case study: Image restoration
8. Case study: Image reconstruction and Computed Tomography

2. Modeling: parametric and non-parametric, MA, AR and ARMA models

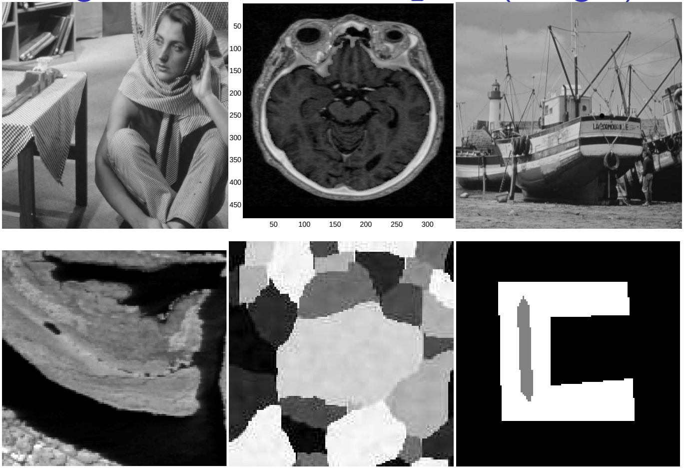
- Modeling ? for what ?
- Deterministic / Probalistic modeling
- Parametric / Non Parametric
- Moving Average (MA)
- Autoregressive (AR)
- Autoregressive Moving Average (ARMA)
- Classical methods for parameter estimation (LS, WLS)

Modeling ? What for ? 1D signals



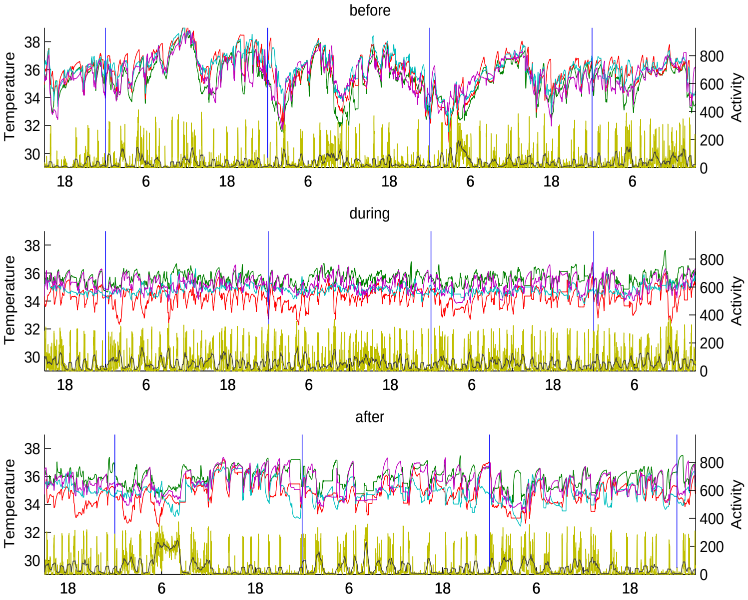
- 1D signals:
  - Is it periodic? What is the period?
  - Is there any structure?
  - Has something changed before, during and after some traitement
  - Can we compress it? How? How much?

Modeling ? What for ? 2D signals (Images)

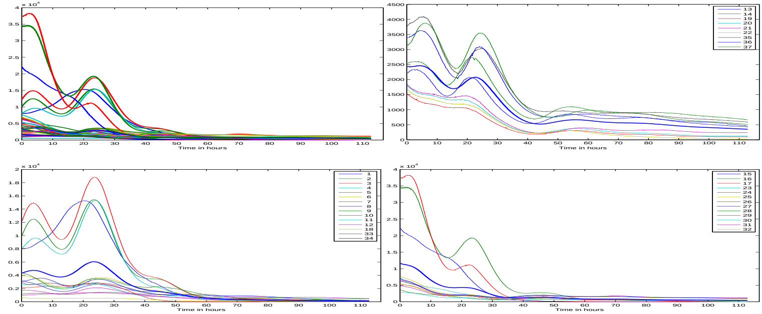


- Images:
  - Is there any structure?
  - Contours? Regions?
  - Can we compress it? How? How much?

Modeling ? What for ? multi dimensional time series



Modeling ? What for ? multi dimensional time series



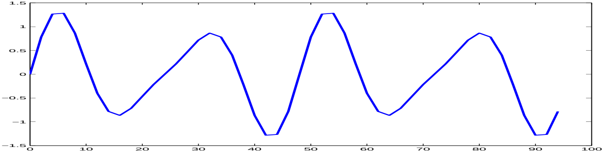
- Multi Dimentionsional signals $g_1(t), \cdots, g_n(t)$
  - Dependancy: Are they all independent?
    If not, which ones are related?
  - Dimensionality reduction: Can we reduce the dimensionality?
  - Principal Components Analysis (PCA):
    What are the principal components?
  - Independent Components Analysis (ICA):
    What are the independent components?
  - Factor Analysis (FA): What are the principal factors?

Deterministic / Probalistic modeling

- **Deterministic:**
  - The signal is a sinusoid $f(t) = a \sin(\omega t + \phi)$. We need just to determine the three parameters a, ω, ϕ.
  - The signal is a periodic $f(t) = \sum_{k=1}^{K} a_k \cos(k\omega_0 t) + b_k \sin(k\omega_0 t)$. If we know ω_0, then, we need just to determine the parameters $(a_k, b_k), k = 1, \cdots, K$.
  - The signal represents a spectra $f(t) = \sum_{k=1}^{K} a_k \mathcal{N}(m_k, v_k)$. We need just to determine the parameters $(a_k, m_k, v_k), k = 1, \cdots, K$.
  - In the last two cases, one great difficulty is determining K
- **Probabilistic:**
  - The shape of the signal is more sophisticated.
  - 1 Sinusoid + noise $f(t) = a \sin(\omega t + \phi) + \epsilon(t)$
  - K Sinusoids + noise $f(t) = \sum_{k=1}^{K} a_k \sin(\omega_k t + \phi_k) + \epsilon(t)$
  - No specific shapes: MA, AR, ARMA, ...

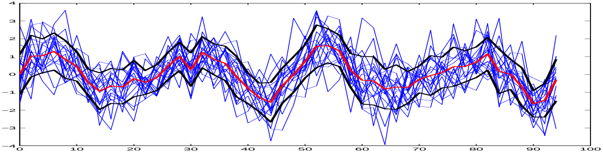
Deterministic and probabilistic modeling of signals

- Deterministe: $f(t)=sin(\omega t)$, the value of f at time t is always the same.



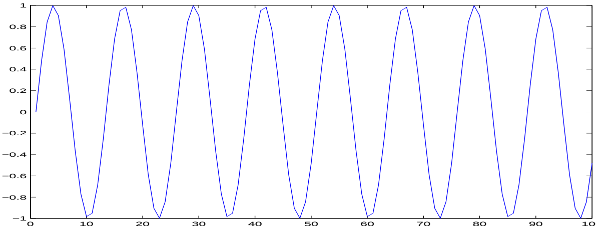
$$f_1(t)=\sin(2\omega t)+.4\sin(3\omega t),\ \omega=\pi/12$$

- Random $f(t)=sin(\omega t)+\epsilon(t)$, the value of f at time t is not always the same.
- For a random signal $f(t)$ for the value of f we can define a probability law, mean, variance, ...

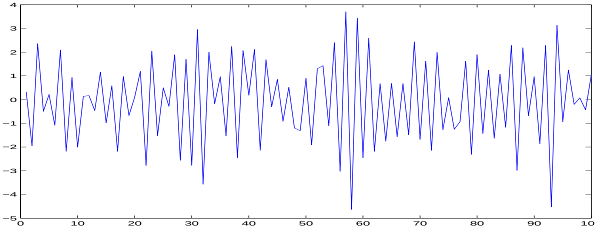


$$f2(t)=f_1(t)+\epsilon(t),\quad \epsilon(t)\sim\mathcal{N}(0,1).$$

Determinist/Probabilist

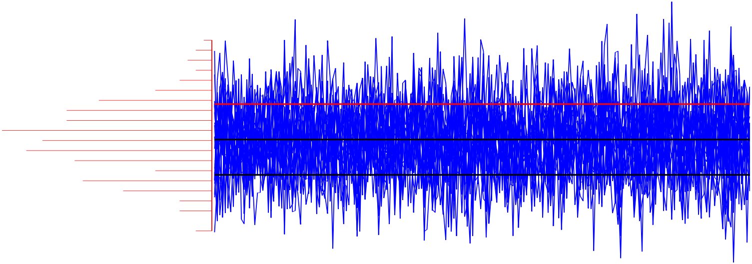


$$f(t) = a\,\sin(\omega t + \phi),\ a = 1, \omega = 2\pi, \phi = 0$$

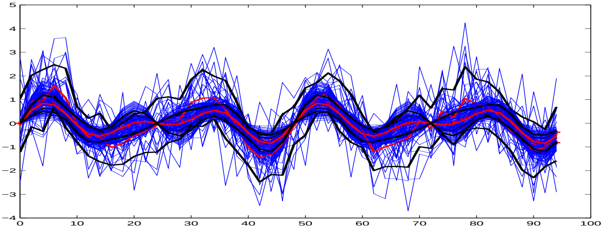


$$\text{ARMA(1,2)},\ q = 1,\ v = 1,\ b_1 = 1,\ a_1 = 1, a_2 = .8$$

Stationary/Non Stationary



$$f(t) \sim \mathcal{N}(0,1), \quad \forall t \longrightarrow \mathrm{E}\{f(t)\} = 0, \mathrm{Var}\{f(t)\} = 1, \forall t$$



$$f_3(t) = a_1 \sin(2\omega t) + a_2 \sim (3\omega t),\ \omega = \pi/12,\ a_1, a_2 \sim \mathcal{U}(0,1)$$

Parametric / Non Parametric

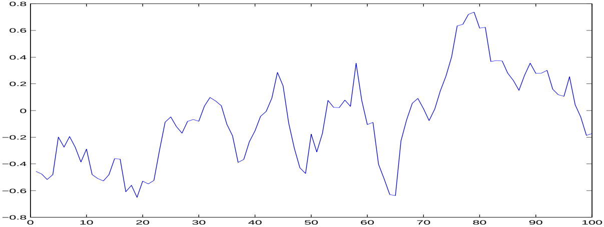
- **Parametric**
  - K Sinusoids + noise $f(t)=\sum_{k=1}^{K} a_k \sin(\omega_k t+\phi_k)+\epsilon(t)$. The parameters are $(a_k,\omega_k,\phi_k), k=1,\cdots,K$
  - K Complex exponentials + noise $f(t)=\sum_{k=1}^{K} c_k \exp\{-j\omega_k t\}+\epsilon(t)$. The parameters are $(c_k,\omega_k), k=1,\cdots,K$
  - Sum of K Gaussian shapes: $f(t)=\sum_{k=1}^{K} a_k\,\mathcal{N}(m_k,v_k)$. The parameters are $(a_k,m_k,v_k), k=1,\cdots,K$.
- **Non-Parametric**
  - The shape of the signal is more sophisticated.
  - The shape is composed of as much as the number of data of Complex exponentials + noise $f(t)=\sum_{n=1}^{N} c_n \exp\{-jn\omega_0 t\}+\epsilon(t)$. If we know ω_0, then, the parameters are $c_n, n=1,\cdots,N$
  - Sum of the Gaussian shapes: $f(t)=\sum_{k=1}^{K} a_n\,\mathcal{N}(m_n,v_n)$. The parameters are $(a_n,m_n,v_n), n=1,\cdots,N$.

Moving Average (MA)

$$f(t) = b(t) * \epsilon(t) = \int b(\tau)\epsilon(t-\tau)\,\mathrm{d}\tau$$

$$f(n) = \sum_{k=0}^{q} b(k)\epsilon(n-k), \quad \forall n$$

$$\epsilon(n) \longrightarrow \boxed{B(z) = \sum_{k=0}^{q} b(k) z^{-k}} \longrightarrow f(n)$$



MA, $q = 15$, $v = 1$, $b_k = \exp\left\{-.05 * k^2\right\}$

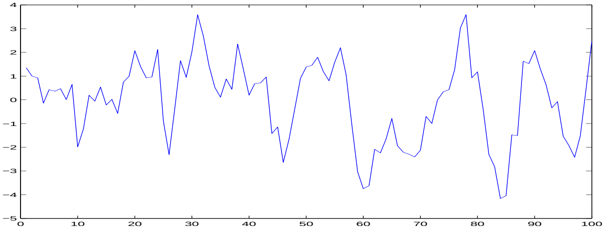
Autoregressive (AR)

$$f(t)=\sum_{k=1}^{p} a(k)\, f(t-k\Delta t)+\epsilon(t) \longrightarrow f(n)=\sum_{k=1}^{p} a(k)\, f(n-k)+\epsilon(n), \quad \forall n$$

$$\mathrm{E}\left\{\epsilon(n)\right\}=0, \quad \mathrm{E}\left\{|\epsilon(n)|^2\right\}=\beta^2,$$

$$\mathrm{E}\left\{\epsilon(n)\, f(m)\right\}=0, \quad m \neq n$$

$$\epsilon(n) \longrightarrow \boxed{H(z)=\frac{1}{A(z)}=\frac{1}{1+\sum_{k=1}^{p} a(k)\, z^{-k}}} \longrightarrow f(n)$$



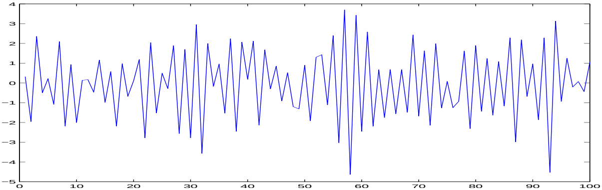
AR1, $a=.7$, $v=1$.

Autoregressive Moving Average (ARMA)

$$f(n) = \sum_{k=1}^{p} a(k)\, f(n-k) + \sum_{l=0}^{q} b(l)\, \epsilon(n-l)$$

$$\epsilon(n) \longrightarrow \boxed{H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{q} b(k) z^{-k}}{1 + \sum_{k=1}^{p} a(k)\, z^{-k}}} \longrightarrow f(n)$$

$$\epsilon(n) \longrightarrow \boxed{B_q(z)} \longrightarrow \boxed{\frac{1}{A_p(z)}} \longrightarrow f(n)$$



ARMA(1,2), $q = 1,\ v = 1,\ b_1 = 1,\ a_1 = 1, a_2 = .8$

Linear prediction and AR modeling

- $\{f(1), \cdots, f(n-1)\}$ observed samples of a signal.
  Predict $f(n)$
- Prediction or innovation Erreur: $\epsilon(n) = f(n) - \widehat{f}(n)$
- Mean Square Errors (MSE):
  $\text{MSE} = \sum_n |\epsilon(n)|^2 = \sum_n |f(n) - \widehat{f}(n)|^2$
- Least Mean Squares (LMS) Error Linear Estimation

$$\widehat{f}(n) = \text{LMSELE}\{f(n)|f(1), \ldots, f(n-1)\} = \arg\min_{f(n)} \{\text{MSE}\}$$

- The linear predictor:

$$\widehat{f}(n) = \sum_{k=1}^{p} a(k)\, f(n-k), \quad \forall n$$

minimizes MSE.

Linear prediction and AR modeling

- $\{f(n)\}$ is a Markov chain of order p

$$\widehat{f}(n) = \mathrm{LMSELE}\{f(n)|f(n-k),\ldots,f(n-1)\} = \sum_{k=1}^{p} a(k)\, f(n-k)$$

- Wihtening Filter

$$f(n) \longrightarrow \boxed{A_p(z)} \longrightarrow \epsilon(n)$$

$$\mathrm{E}\left\{\epsilon(n)\epsilon(m)\right\} = \beta^2\, \delta(n-m)$$

- Link with AR model

$$f(n) = \sum_{k=0}^{N-1} a(k)\, f(n-k) + \epsilon(n), \quad \forall n$$

$$\epsilon(n) \longrightarrow \boxed{H(z) = \frac{1}{A(z)} = \frac{1}{1+\sum_{k=1}^{p} a(k)\, z^{-k}}} \longrightarrow f(n)$$

Minimum Variance Estimation

- $f(n)$, An n sample of a signal
- AR model:

$$\widehat{f}(n) = \sum_{k=0}^{N-1} a(k)\, f(n-k)$$

- Modeling errror

$$\epsilon(n) = f(n) - \widehat{f}(n)$$

- Criterium

$$\beta^2 = \min \mathrm{E}\left\{|\epsilon(n)|^2\right\} = \min \mathrm{E}\left\{[f(n) - \widehat{f}(n)]^2\right\}$$

- Orthogonality Condition

$$\mathrm{E}\left\{[f(n) - \sum_{k=0}^{N-1} a(k)\, f(n-k)]\, f(n-k)\right\} = \beta^2\, \delta(k),\ k = 1, \ldots, p$$

$$r(k) - \sum_{k=0}^{N-1} a(k)\, r(n-k) = \beta^2\, \delta(k)$$

Minimum Variance Estimation

- Correlation matrix

$$\mathbf{R} = \begin{bmatrix} r(0) & r(1) & r(2) & \cdots & \cdots & r(p-1) \\ r(1) & \ddots & \ddots & \ddots & & \vdots \\ r(2) & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & r(2) \\ \vdots & & \ddots & \ddots & \ddots & r(1) \\ r(p-1) & \cdots & \cdots & r(2) & r(1) & r(0) \end{bmatrix}$$

$$\mathbf{r} = [r(1), \ldots r(p)]^t, \quad \mathbf{a} = [a(1), \ldots, a(p)]^t,$$

- Normal equations

$$\begin{aligned} \mathbf{Ra} &= \mathbf{r} \\ r(0) - \mathbf{a}^t\mathbf{r} &= \beta^2 \end{aligned}$$

Causal ou Non-Causal AR models

- **Causal :**

$$f(n) = \sum_{k=1}^{q} a(k)\, f(n-k) + \epsilon(n), \quad \forall n$$

$$A(z) = 1 - \sum_{k=1}^{q} a(k)\, z^{-k} \longrightarrow \epsilon(n) \longrightarrow \boxed{H(z) = \frac{1}{A(z)}} \longrightarrow f(n)$$

- **Non–causal :**

$$f(n) = \sum_{\substack{k=-p \\ k\neq 0}}^{+q} a(k)\, f(n-k) + \epsilon(n), \quad \forall n$$

$$A(z) = 1 - \sum_{\substack{k=-p \\ k\neq 0}}^{+q} a(k)\, z^{-k} \longrightarrow \epsilon(n) \longrightarrow \boxed{H(z) = \frac{1}{A(z)}} \longrightarrow f(n)$$

2D AR Models

$$A(z_1, z_2) = 1 - \sum_{(k,l)} \sum_{\in S} a(k,l) z_1^{-k} z_2^{-k}$$

$$f(m,n) = \sum_{(k,l)} \sum_{\in S} a(k,l)\, f(m-k, n-l) + \epsilon(m,n)$$

- **Non–causal**
  $S = \{l \geq 1, \forall k\} \cup \{l = 0, k \neq 0\}$
- **Semi–ausal**
  $S = \{l \geq 1, \forall k\} \cup \{l = 0, k \geq 1\}$
- **Causal**
  $S = \{(k,l) \neq (0,0)\}$

2D AR Models

- **Causal**
  - $S = \{l \geq 1, \forall k\} \cup \{l = 0, k \neq 0\}$
  - Recursive Filtre
  - Finite Differential Equations with initial conditions
  - Hyperbolic Partial Differential Equations
- **Semi–causal**
  - $S = \{l \geq 1, \forall k\} \cup \{l = 0, k \geq 1\}$
  - Semi–recursif Filters
  - Finite Differential Equations with initial conditions in one dimention and limit conditions in other dimension
  - Parabolic Partial Differential Equations
- **Non–causal**
  - $S = \{(k, l) \neq (0, 0)\}$
  - Non-recursive Filtre
  - Finite Differential Equations with limit conditions in both dimensions
  - Elliptic Partial Differential Equations

Causal/Non-Causal Prediction

- **Causal :**

$$\widehat{f}(n) = \sum_{k=0}^{N-1} a(k)\, f(n-k)$$

- **Non-Causal :**

$$\widehat{f}(n) = \sum_{\substack{k=-p \\ k\neq 0}}^{+q} a(k)\, f(n-k)$$

2D AR models and 2D prediction

$$A(z_1,z_2) = 1 - \sum_{(k,l)} \sum_{\in S} a(k,l) z_1^{-k} z_2^{-k}$$

$$f(m,n) = \sum_{(k,l)} \sum_{\in S} a(k,l)\, f(m-k, n-l) + \epsilon(m,n)$$

- **Non–causal**
- **Semi–ausal**
- **Causal**

2D AR models and 2D prediction

- **Causal**
  - $S = \{l \geq 1, \forall k\} \cup \{l = 0, k \neq 0\}$
  - Recursive Filtering
  - Finite Difference Equation (FDE) with initial conditions
  - Partial Differential Equations (Hyperbolic)
- **Semi–causal**
  - $S = \{l \geq 1, \forall k\} \cup \{l = 0, k \geq 1\}$
  - Filtre semi–récursif
  - FDE with initial conditions in one direction and limit conditions in other direction.
  - Partial Differential Equations (Parabolic)
- **Non–causal**
  - $S = \{(k, l) \neq (0, 0)\}$
  - Non-Recursive Filtering
  - FDE with limit conditions in both directions
  - Partial Differential Equations (Elliptic)

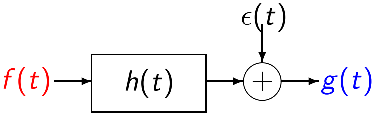
Content

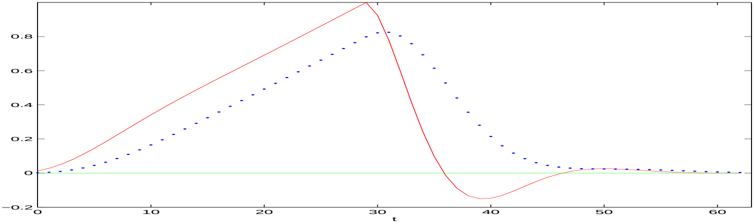
1. Introduction: Signals and Images, Linear transformations (Convolution, Fourier, Laplace, Hilbert, Radon, ..., Discrete convolution, Z transform, DFT, FFT, ...)
2. Modeling: parametric and non-parametric, MA, AR and ARMA models, Linear prediction
3. **Deconvolution and Parameter Estimation: Deterministic (LS, Regularization) and Probabilistic methods (Wiener Filtering)**
4. Inverse problems and Bayesian estimation
5. Kalman Filtering and smoothing
6. Case study: Signal deconvolution
7. Case study: Image restoration
8. Case study: Image reconstruction and Computed Tomography

3. Deconvolution and Parameter estimation

- Signal Deconvolution and Image Restoration
- Least Squares method (LS)
- Limitations of LS methods
- Regularization methods
- Parametric modeling
- Examples:
  - Sinusoids in the noise (MUSIC)
  - Antenna Array Processing
  - Mixture models (Gaussian and Cauchy)

Convolution in signal processing





$$g(t) = \int f(t')h(t-t')\,\mathrm{d}t' + b(t) = \int h(t')f(t-t')\,\mathrm{d}t' + b(t)$$

- $f(t)$, $g(t)$ and $h(t)$ are discretized with $\Delta T = 1$.

Convolution in signal processing: general case

- $h(t) = 0, \forall t < -q\Delta T$, et $\forall t > p\Delta T$.

$$g(m) = \sum_{k=-q}^{p} h(k)\, f(m-k) + b(m), \quad m = 0, \cdots, M \longrightarrow \mathbf{g} = \mathbf{H}\mathbf{f} + \epsilon$$

$$\mathbf{g} = [g(0), g(1), \cdots, g(M)]',$$
$$\mathbf{f} = [f(-p), \cdots, f(0), f(1), \cdots, f(M), \cdots, f(M+q)]'$$

$$\begin{bmatrix}
h(p) & \cdots & h(0) & \cdots & h(-q) & 0 & \cdots & \cdots & \cdots & 0 \\
0 & \ddots & & \ddots & & \ddots & & & & \vdots \\
\vdots & & & & & & & & & \vdots \\
\vdots & & & h(p) & \cdots & h(0) & \cdots & h(-q) & & \vdots \\
\vdots & & & & & & & & & \vdots \\
\vdots & & & & & \ddots & & & \ddots & 0 \\
0 & \cdots & \cdots & \cdots & 0 & h(p) & \cdots & h(0) & \cdots & h(-q)
\end{bmatrix}$$

Convolution in signal processing: Causal systems

Causal system $q = 0$:

$$\begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ g(M) \end{bmatrix} = \begin{bmatrix} h(p) & \cdots & h(0) & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & & & & & & & \vdots \\ \vdots & & & & & & & \vdots \\ \vdots & & h(p) & \cdots & h(0) & & & \vdots \\ \vdots & & & & & & & \vdots \\ \vdots & & & & & & & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & h(p) & \cdots & h(0) \end{bmatrix} \begin{bmatrix} f(-p) \\ \vdots \\ f(0) \\ f(1) \\ \vdots \\ \vdots \\ f(M) \end{bmatrix}$$

Convolution: Causal systems and causal input

Causal system and causal input signal

$$\begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ g(M) \end{bmatrix} = \begin{bmatrix} h(0) & & & & & \\ h(1) & \ddots & & & & \\ \vdots & & & & & \\ h(p) & \cdots & & h(0) & & \\ 0 & \ddots & & & \ddots & \\ \vdots & & & & & \\ 0 & \cdots & 0 & h(p) & \cdots & h(0) \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f(M) \end{bmatrix}$$

if $p = M$:

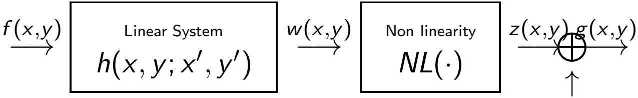
$$\begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ g(M) \end{bmatrix} = \begin{bmatrix} h(0) & & & \\ h(1) & h(0) & & \\ \vdots & & \ddots & \\ h(M) & \cdots & h(1) & h(0) \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(M) \end{bmatrix}$$

Convolution in signal processing: Circulante case

$$
\begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ g(M) \end{bmatrix}
=
\begin{bmatrix}
h(0) & & & & & & h(p) & \cdots & h(1) \\
h(1) & \ddots & & & & & & \ddots & \vdots \\
\vdots & & & & & & & & h(p) \\
h(p) & \cdots & & h(0) & & & & & 0 \\
0 & \ddots & & & \ddots & & & & \vdots \\
\vdots & & & & & & & & \\
0 & \cdots & 0 & h(p) & \cdots & h(0) & 0 & \cdots & 0
\end{bmatrix}
\begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ \vdots \\ \vdots \\ f(M) \\ 0 \\ \vdots \\ 0 \end{bmatrix}
$$

Modeling imaging systems

General observation model:

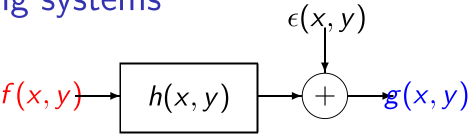


$f(x,y)$ → Linear System $h(x,y;x',y')$ → $w(x,y)$ → Non linearity $NL(\cdot)$ → $z(x,y)$ ⊕ ↑ → $g(x,y)$

If we neglect the Non Linearity, then

$$g(x,y) = \iint f(x',y')\, h(x,y;x',y')\, \mathrm{d}x'\, \mathrm{d}y' + \epsilon(x,y)$$

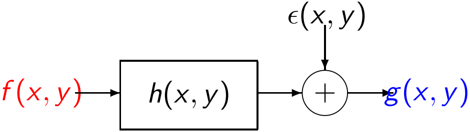
Modeling imaging systems



$$g(x, y) = \iint f(x', y')\, h(x - x', y - y')\, \mathrm{d}x'\, \mathrm{d}y' + b(x, y)$$

System	Point Spread function $h(x, y)$	Transfert Function $H(u, v)$
Diffraction+ Coherent source (Rectangular)	$ab\,\mathrm{sinc}(ax)\mathrm{sinc}(by)$	$\mathrm{rectf}\frac{u}{a}, \frac{v}{b}$
Diffraction+ Incoherent Source (Rectangular)	$\mathrm{sinc}^2(ax)\mathrm{sinc}^2(by)$	$\mathrm{trif}\frac{u}{a}, \frac{v}{b}$
Atmosphoric Turbulance	$\exp\left\{-\pi a^2(x^2 + y^2)\right\}$	$\frac{1}{a^2}\exp\left\{\frac{-\pi(u^2 + v^2)}{a^2}\right\}$
Horizontal movement	$frac1a\mathrm{rectf}\frac{x}{a} - \frac{1}{2}\delta(y)$	$\exp\{-j\pi ua\}\,\mathrm{sinc}(ua)$
CCD interaction	$\sum\sum_{k,l=-1}^{1} \alpha_{k,l}\delta(x - k\Delta, y - l\Delta)$	$\sum\sum_{k,l=-1}^{1} \alpha_{k,l}\exp\{-j2\pi\Delta(ux + vy)\}$

2D Convolution for image restoration



$$g(x,y) = \iint_D f(x',y')\, h(x-x',y-y')\, \mathrm{d}x'\, \mathrm{d}y' + b(x,y)$$

$$g(m\Delta x, n\Delta y) = \sum_{i=-I}^{+I} \sum_{j=-J}^{+J} h(i\Delta x, j\Delta y) f((m-i)\Delta x, (n-j)\Delta y)$$

$$\left\{ \begin{bmatrix} m = 1, \ldots, M \\ n = 1, \ldots, N \end{bmatrix} \right. \qquad \Delta x = \Delta y = 1$$

$$g(m,n) = \sum_{i=-I}^{+I} \sum_{j=-J}^{+J} h(i,j)\, f(m-i, n-j)$$

2D Convolution for image restoration

Two caracteristics:

$$g(m,n)=\sum_{i=-I}^{+I}\sum_{j=-J}^{+J} h(i,j)\, f(m-i,n-j)$$

- $g(m,n)$ depends on $f(k,l)$ for $(k,l)\in\mathcal{N}(k,l)$ where $\mathcal{N}(k,l)$ means the neigborhood pixels around the pixel $'(k,l) \longrightarrow$ No Causality
- The boarding effects cannot be neglected as easily as in the 1D case.

Vectorial Forme:

$$g(m,n)=\sum_{i=-I}^{+I}\sum_{j=-J}^{+J} h(i,j)\, f(m-i,n-j)$$

$$g(m,n)\left\{\begin{bmatrix} m=1,\ldots,M \\ n=1,\ldots,N\end{bmatrix}\right. \quad f(k,l)\left\{\begin{bmatrix} k=1,\ldots,K \\ l=1,\ldots,L\end{bmatrix}\right. \quad h(i,j)\left\{\begin{bmatrix} i=1,\ldots,I \\ j=1,\ldots,J\end{bmatrix}\right.$$

$$\mathbf{g}=\mathbf{H}\mathbf{f}$$

2D Convolution for image restoration

$$\mathbf{g} = [\quad \underset{---1---}{g_{(1,1)}, \ldots, g(M,1)}, \quad \underset{---2---}{g_{(1,2)}, \ldots, g_{(M,2)}}, \quad \ldots, \quad \underset{---N---}{g_{(1,N)}, \ldots, g_{(M,N)}}]^t$$

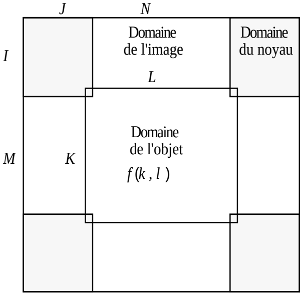
$$\mathbf{f} = [\quad \underset{---1---}{f_{(1,1)}, \ldots, f_{(K,1)}}, \quad \underset{---2---}{f_{(1,2)}, \ldots, f(K,2)}, \quad \ldots, \quad \underset{---L---}{f_{(1,L)}, \ldots, f_{(K,L)}}]^t$$

The structure of the matrix **H** depends on the domaines D_h, D_f and D_g .

Matrix Form H :

- Image $>$ Object
- Image$=$Object
- Image $<$ Object

2D Convolution for image restoration: Image > Object



$$D_g > D_f \quad \left\{ \begin{bmatrix} M = K + I - 1 \\ N = L + J - 1 \end{bmatrix} \right.$$

$$g(m,n) \left\{ \begin{bmatrix} m = 1, \ldots, M \\ n = 1, \ldots, N \end{bmatrix} \right. \quad f(k,l) \left\{ \begin{bmatrix} k = 1, \ldots, K \\ l = 1, \ldots, L \end{bmatrix} \right. \quad h(i,j) \left\{ \begin{bmatrix} i = 1, \ldots, I \\ j = 1, \ldots, J \end{bmatrix} \right.$$

$$\mathbf{g} = [\quad \underset{---1---}{g_{(1,1)}, \ldots, g(M,1)}, \quad \underset{---2---}{g_{(1,2)}, \ldots, g_{(M,2)}}, \quad \ldots, \quad \underset{---N---}{g_{(1,N)}, \ldots, g_{(M,N)}}]^t$$

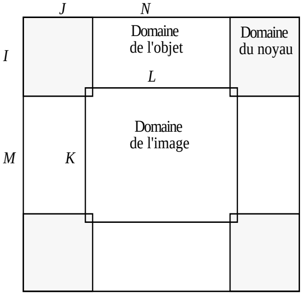
$$\mathbf{f} = [\quad \underset{---1---}{f_{(1,1)}, \ldots, f_{(K,1)}}, \quad \underset{---2---}{f_{(1,2)}, \ldots, f(K,2)}, \quad \ldots, \quad \underset{---L---}{f_{(1,L)}, \ldots, f_{(K,L)}}]^t$$

2D Convolution for image restoration: Image > Object

$$\mathbf{H} = \begin{bmatrix} H_I & \cdot & \cdots & \cdots & \cdots & \cdot \\ \vdots & H_I & \ddots & & & \vdots \\ \vdots & & & \ddots & & \vdots \\ \vdots & & & & \ddots & \vdots \\ H_1 & & & & H_I & \cdot \\ \cdot & H_1 & & & & H_I \\ \vdots & \ddots & H_1 & & & \vdots \\ \vdots & & \ddots & & & \vdots \\ \vdots & & & \ddots & & \vdots \\ \cdot & \cdots & \cdots & \cdots & \cdot & H_1 \end{bmatrix} H_i = \begin{bmatrix} h(i,J) & \cdot & \cdots & \cdots & \cdots & \cdot \\ \vdots & h(i,J) & \ddots & & & \vdots \\ \vdots & & & \ddots & & \vdots \\ \vdots & & & & \ddots & \vdots \\ h(i,1) & & & & h(i,J) & \cdot \\ \cdot & h(i,1) & & & & h(i,J) \\ \vdots & \ddots & h(i,1) & & & \vdots \\ \vdots & & \ddots & & & \vdots \\ \vdots & & & \ddots & & \vdots \\ \cdot & \cdots & \cdots & \cdots & \cdot & h(i,1) \end{bmatrix}$$

Toeplitz-Bloc-Toeplitz

2D Convolution for image restoration: Image < Object



$$D_g < D_f \quad \left\{ \begin{bmatrix} M = K - I - 1 \\ N = L - J + 1 \end{bmatrix} \right.$$

$$g(m,n) \left\{ \begin{bmatrix} m = 1, \ldots, M \\ n = 1, \ldots, N \end{bmatrix} \right. \quad f(k,l) \left\{ \begin{bmatrix} k = 1, \ldots, K \\ l = 1, \ldots, L \end{bmatrix} \right. \quad h(i,j) \left\{ \begin{bmatrix} i = 1, \ldots, I \\ j = 1, \ldots, J \end{bmatrix} \right.$$

$$\mathbf{g} = [\quad \underset{---1---}{g_{(1,1)}, \ldots, g_{(M,1)}}, \quad \underset{---2---}{g_{(1,2)}, \ldots, g_{(M,2)}}, \quad \ldots, \quad \underset{---N---}{g_{(1,N)}, \ldots, g_{(M,N)}}]^t$$

$$\mathbf{f} = [\quad \underset{---1---}{f_{(1,1)}, \ldots, f_{(K,1)}}, \quad \underset{---2---}{f_{(1,2)}, \ldots, f_{(K,2)}}, \quad \ldots, \quad \underset{---L---}{f_{(1,L)}, \ldots, f_{(K,L)}}]^t$$

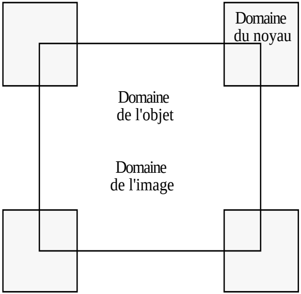
2D Convolution for image restoration: Image $<$ Object

$$\mathbf{H} = \begin{bmatrix} H_1 & H_2 & \cdots & H_I & \cdot & \cdots & \cdot \\ \cdot & H_1 & & & H_I & \ddots & \vdots \\ \vdots & \ddots & H_1 & & & H_I & \cdot \\ \vdots & & & & & & \vdots \\ \cdot & \cdots & \cdot & H_1 & H_2 & & H_I \end{bmatrix}$$

with

$$H_i = \begin{bmatrix} h(i,1) & h(i,2) & \cdots & h(i,J) & \cdot & \cdots & \cdot \\ \cdot & h(i,1) & & & h(i,J) & \ddots & \vdots \\ \vdots & \ddots & h(i,1) & & & h(i,J) & \cdot \\ \vdots & & & & & & \vdots \\ \cdot & \cdots & \cdot & h(i,1) & h(i,2) & & h(i,J) \end{bmatrix}$$

2D Convolution for image restoration: Image=Object



$$D_g = D_f \quad \left\{ \begin{bmatrix} M = K \\ N = L \end{bmatrix} \right.$$

$$g(m,n) \left\{ \begin{bmatrix} m = 1,\ldots,M \\ n = 1,\ldots,N \end{bmatrix} \right. \quad f(k,l) \left\{ \begin{bmatrix} k = 1,\ldots,K \\ l = 1,\ldots,L \end{bmatrix} \right. \quad h(i,j) \left\{ \begin{bmatrix} i = 1,\ldots,I \\ j = 1,\ldots,J \end{bmatrix} \right.$$

$$\mathbf{g} = [\quad \underset{---1---}{g_{(1,1)},\ldots,g_{(M,1)}}, \quad \underset{---2---}{g_{(1,2)},\ldots,g_{(M,2)}}, \quad \ldots, \quad \underset{---N---}{g_{(1,N)},\ldots,g_{(M,N)}}]^t$$

$$\mathbf{f} = [\quad \underset{---1---}{f_{(1,1)},\ldots,f_{(K,1)}}, \quad \underset{---2---}{f_{(1,2)},\ldots,f_{(K,2)}}, \quad \ldots, \quad \underset{---L---}{f_{(1,L)},\ldots,f_{(K,L)}}]^t$$

2D Convolution for image restoration: Circulante forme

$$g(m,n)\left\{\begin{bmatrix} m=1,\ldots,M \\ n=1,\ldots,N\end{bmatrix}\right. \quad f(k,l)\left\{\begin{bmatrix} k=1,\ldots,K \\ l=1,\ldots,L\end{bmatrix}\right. \quad h(i,j)\left\{\begin{bmatrix} i=1,\ldots,I \\ j=1,\ldots,J\end{bmatrix}\right.$$

$$\left\{\begin{bmatrix} P=K+I-1 \\ Q=L+J-1\end{bmatrix}\right.$$

$$\tilde{f}(m,n)=\begin{bmatrix} f((m,n)) & \vdots & 0 \\ \cdots\cdots & \cdots & \cdots\cdots \\ 0 & \vdots & 0 \end{bmatrix} \qquad \dim(\tilde{f})=[P,Q]$$

$$\tilde{g}(k,l)=\begin{bmatrix} g(k,l) & \vdots & 0 \\ \cdots\cdots & \cdots & \cdots\cdots \\ 0 & \vdots & 0 \end{bmatrix} \qquad \dim(\tilde{g})=[P,Q]$$

$$\tilde{h}(i,j)=\begin{bmatrix} h(i,j) & \vdots & 0 \\ \cdots\cdots & \cdots & \cdots\cdots \\ 0 & \vdots & 0 \end{bmatrix} \qquad \dim(\tilde{h})=[P,Q]$$

2D Convolution for image restoration: Circulante forme

$$H=\begin{bmatrix} H_1 & H_2 & \cdots & \cdots & \cdots & H_P \\ H_P & H_1 & H_2 & \cdots & \cdots & H_{P-1} \\ \vdots & \vdots & \vdots & & & \vdots \\ \vdots & \vdots & \vdots & & & \vdots \\ H_P & H_{P-1} & \cdots & \cdots & \cdots & H_1 \end{bmatrix} \quad \text{bloc-circulante}$$

$$H_i=\begin{bmatrix} h(i,1) & h(i,2) & \cdots & \cdots & \cdots & h(i,P) \\ h(i,P) & h(i,1) & h(i,2) & \cdots & \cdots & h(i,P) \\ \vdots & \vdots & \vdots & & & \vdots \\ \vdots & \vdots & \vdots & & & \vdots \\ h(i,P) & h(i,P-1) & h(i,P-2) & \cdots & \cdots & h(i,1) \end{bmatrix} \quad \text{circulante}$$

Circulante-Bloc-Circulante

Classification of the signal and image restoration methods

- **Analytical methods**

$$f(t) \longrightarrow \boxed{\quad\mathcal{H}\quad} \longrightarrow g(t) = \mathcal{H}[f(t)]$$

$$f(x,y) \longrightarrow \boxed{\quad\mathcal{H}\quad} \longrightarrow g(x,y) = \mathcal{H}[f(x,y)]$$

$\mathcal{H}$ Linear Operator

$$g(t) \longrightarrow \boxed{\quad\mathcal{G}\quad} \longrightarrow \widehat{f}(t) = \mathcal{H}^{-1}[f(t)]$$

$$g(x,y) \longrightarrow \boxed{\quad\mathcal{G}\quad} \longrightarrow \widehat{f}(x,y) = \mathcal{H}^{-1}[f(x,y)]$$

$\mathcal{G}$ Linear Operator approximating $\mathcal{H}^{-1}$

  - Inverse Filtring
  - Pseudo–inverse Filtering
  - Wiener Filtering

Classification of the signal and image restoration methods

- **Algebraic methods**

$$g(t) = \mathcal{H}[f(t)] \longrightarrow \quad \text{Discretization} \quad \longrightarrow \mathbf{g} = \mathbf{H}\mathbf{f}$$

$$g(x,y) = \mathcal{H}[f(x,y)] \longrightarrow \quad \text{Discretization} \quad \longrightarrow \mathbf{g} = \mathbf{H}\mathbf{f}$$

Ideal case : H invertible $\longrightarrow \widehat{\mathbf{f}} = \mathbf{H}^{-1}\mathbf{g}$
More general case : H is not invertible

  - Pseudo-inverse
  - Generalized Inversion
  - Least Squares (LS) and Minimum norm LS
  - Regularization

- **Probabilistic methods**
  - Wiener Filtering
  - Kalman Filtering
  - General Bayesian approach

Analytical methods: Inverse Filtering

$$f(t) \longrightarrow \boxed{\quad h(t) \quad} \longrightarrow g(t) = h(t) * f(t)$$

$$f(x,y) \longrightarrow \boxed{\quad h(x,y) \quad} \longrightarrow g(x,y) = h(x,y) * f(x,y)$$

$$F(u,v) \longrightarrow \boxed{\quad H(u,v) \quad} \longrightarrow G(u,v) = H(u,v) \cdot F(u,v)$$

- Inverse Filtering

$$H^{-1}(u,v) = \frac{1}{H(u,v)}$$

$$G(u,v) \longrightarrow \boxed{\quad \frac{1}{H(u,v)} \quad} \longrightarrow F(u,v)$$

- **Difficulties:**
  - What to do when $H(u,v) = 0$ for sme values of (u,v)?

Analytical methods: Inverse Filtering

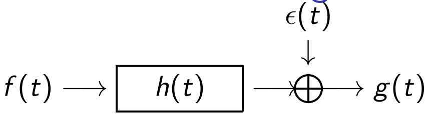
- Noise Amplification

$$\widehat{F}(u,v) = \frac{\widehat{G}(u,v)}{H(u,v)} = \frac{(H(u,v)\,F(u,v) + N(u,v))}{H(u,v)} = F(u,v) + \frac{N(u,v)}{H(u,v)}$$

- Pseudo–inverse Filtering

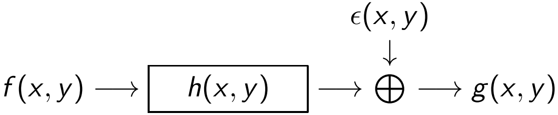
$$H^{-1}(u,v) = \begin{cases} \dfrac{1}{H(u,v)}, & H \neq 0 \\ 0, & H = 0 \end{cases}$$

Analytical methods: Wiener Filtering



$$g(t) = \iint f(t')\, h(t-t')\, \mathrm{d}t' + \epsilon(t)$$

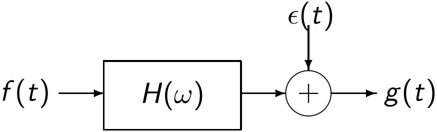
- $f(t)$, $g(t)$ and $\epsilon(t)$ are modelled as Gaussian random signal



$$g(x,y) = \iint f(x',y')\, h(x-x',y-y')\, \mathrm{d}x'\, \mathrm{d}y' + \epsilon(x,y)$$

- $f(x,y)$, $g(x,y)$ and $\epsilon(x,y)$ are modelled as homogeneous and Gaussian random fields

Wiener Filtering



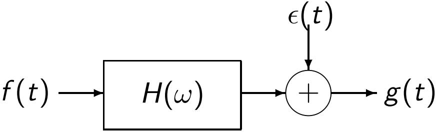
$$g(t) = h(t) * f(t) + \epsilon(t)(t)$$

$$\mathrm{E}\left\{g(t)\right\}, \quad \mathrm{E}\left\{\epsilon(t)\right\} \quad \text{and} \quad \mathrm{E}\left\{f(t)\right\}$$

$$R_{gg}(\tau) = \mathrm{E}\left\{g(t)\, g(t+\tau)\right\}$$
$$R_{ff}(\tau) = \mathrm{E}\left\{f(t)\, f(t+\tau)\right\}$$

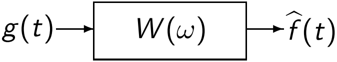
$$R_{bf}(\tau) = R_{fb}(-\tau) = \mathrm{E}\left\{\epsilon(t)\, f(t+\tau)\right\}$$
$$R_{gf}(\tau) = R_{fg}(-\tau) = \mathrm{E}\left\{g(t)\, f(t+\tau)\right\}$$

Wiener Filtering



$$
\begin{aligned}
\mathrm{E}\left\{g(t)\right\} &= h(t) * \mathrm{E}\left\{f(t)\right\} + \mathrm{E}\left\{\epsilon(t)\right\} \\
R_{gg}(\tau) &= h(t) * h(t) * R_{ff}(\tau) + R_{\epsilon\epsilon}(\tau) \\
R_{gf}(\tau) &= h(t) * R_{ff}(\tau)
\end{aligned}
$$

$$
\begin{aligned}
S_{gg}(\omega) &= |H(\omega)|^2 S_{ff}(\omega) + R_{\epsilon\epsilon}(\omega) \\
S_{gf}(\omega) &= H(\omega) S_{ff}(\omega) \\
S_{fg}(\omega) &= H^*(\omega) S_{ff}(\omega)
\end{aligned}
$$



$$
\widehat{f}(t) = w(t) * g(t)
$$

Wiener Filtering

$$EQM = \mathrm{E}\left\{[f(t) - \widehat{f}(t)]^2\right\} = \mathrm{E}\left\{[f(t) - w(t) * g(t)]^2\right\}$$

$$\frac{\partial EQM}{\partial f} = -2\mathrm{E}\left\{[f(t) - w(t) * g(t)] * g(t+\tau)\right\} = 0$$

$$\mathrm{E}\left\{[f(t) - w(t) * g(t)]\, g(t+\tau)\right\} = 0 \quad \forall t, \tau \longrightarrow$$

$$R_{fg}(\tau) = w(t) * R_{gg}(\tau)$$

$$W(\omega) = \frac{S_{fg}(\omega)}{S_{gg}(\omega)} = \frac{H^*(\omega)\, S_{ff}(\omega)}{|H(\omega)|^2\, S_{ff}(\omega) + S_{\epsilon\epsilon}(\omega)}$$

$$\boxed{W(\omega) = \frac{H^*(\omega) S_{ff}(\omega)}{|H(\omega)|^2 S_{ff}(\omega) + S_{\epsilon\epsilon}(\omega)} = \frac{1}{H(\omega)} \frac{|H(\omega)|^2}{|H(\omega)|^2 + \frac{S_{\epsilon\epsilon}(\omega)}{S_{ff}(\omega)}}}$$

Analytical methods: Wiener Filtering

- Linear Estimation: $\widehat{f}(x,y)$ is such that:
  - $\widehat{f}(x,y)$ depends on $g(x,y)$ in a linear way:

$$\widehat{f}(x,y)=\iint g(x',y')\,w(x-x',y-y')\,\mathrm{d}x'\,\mathrm{d}y'$$

    $w(x,y)$ is the impulse response of the Wiener filtre
  - minimizes MSE: $\mathrm{E}\left\{|f(x,y)-\widehat{f}(x,y)|^2\right\}$
- Orthogonality condition:

$$(f(x,y){-}\widehat{f}(x,y))\perp g(x',y') \quad\longrightarrow\quad \mathrm{E}\left\{(f(x,y)-\widehat{f}(x,y))\,g(x',y')\right\}=0$$

$$\widehat{f}=g{*}w \;\longrightarrow\; \mathrm{E}\left\{(f(x,y)-g(x,y)*w(x,y))\,g(x+\alpha_1,y+\alpha_2)\right\}=0$$

$$R_{fg}(\alpha_1,\alpha_2)=(R_{gg}{*}w)(\alpha_1,\alpha_2)\longrightarrow \mathrm{TF}\longrightarrow S_{fg}(u,v)=S_{gg}(u,v)W(u,v)$$

$$\Downarrow$$

$$\boxed{W(u,v)=\frac{S_{fg}(u,v)}{S_{gg}(u,v)}}$$

Wiener filtering

Signal	Image
$W(\omega) = \frac{S_{fg}(\omega)}{S_{gg}(\omega)}$	$W(u,v) = \frac{S_{fg}(u,v)}{S_{gg}(u,v)}$

Particular Case:

$f(x,y)$ and $b(x,y)$ are assumed to be centered and non correlated

$$S_{fg}(u,v) = H'(u,v)\,S_{ff}(u,v)$$

$$S_{gg}(u,v) = |H(u,v)|^2\,S_{ff}(u,v) + S_{\epsilon\epsilon}(u,v)$$

$$W(u,v) = \frac{H'(u,v)S_{ff}(u,v)}{|H(u,v)|^2 S_{ff}(u,v) + S_{\epsilon\epsilon}(u,v)}$$

Signal	Image
$W(\omega) = \frac{1}{H(\omega)} \frac{\|H(\omega)\|^2}{\|H(\omega)\|^2 + \frac{S_{\epsilon\epsilon}(\omega)}{S_{ff}(\omega)}}$	$W(u,v) = \frac{1}{H(u,v)} \frac{\|H(u,v)\|^2}{\|H(u,v)\|^2 + \frac{S_{\epsilon\epsilon}(u,v)}{S_{ff}(u,v)}}$

Wiener Filtering: Implementation

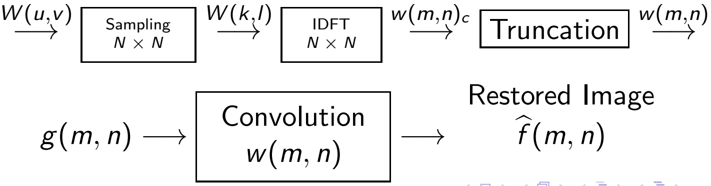
- No noise ⟶ $W(u,v) = 1/H(u,v)$
- $S_{\epsilon\epsilon}(u,v)/S_{ff}(u,v)$ have to be given.
- No blur case: $h(x,y) = \delta(x,y)$ (denoising):

$$W(u,v) = \frac{1}{1 + \frac{S_{\epsilon\epsilon}(u,v)}{S_{ff}(u,v)}}$$

Numerical implementation:

- Direct Implentation by convolution

$$W(u,v) = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + \frac{S_{\epsilon\epsilon}(u,v)}{S_{ff}(u,v)}}$$

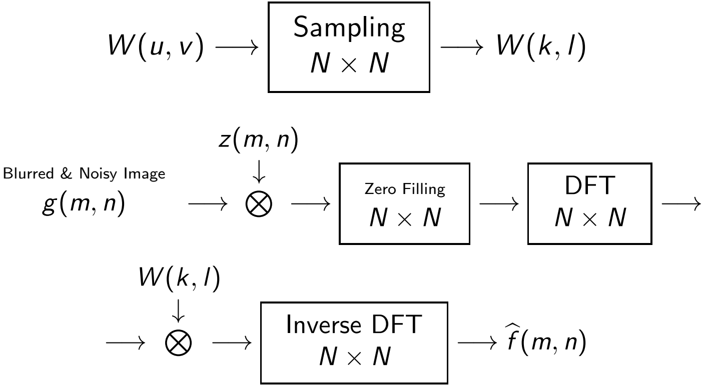


$$\xrightarrow{W(u,v)} \boxed{\begin{array}{c}\text{Sampling}\\ N \times N\end{array}} \xrightarrow{W(k,l)} \boxed{\begin{array}{c}\text{IDFT}\\ N \times N\end{array}} \xrightarrow{w(m,n)_c} \boxed{\text{Truncation}} \xrightarrow{w(m,n)}$$

$$g(m,n) \longrightarrow \boxed{\begin{array}{c}\text{Convolution}\\ w(m,n)\end{array}} \longrightarrow \begin{array}{c}\text{Restored Image}\\ \widehat{f}(m,n)\end{array}$$

Wiener Filtering: Implementation

- Fourier domaine Implentation



$W(u,v) \longrightarrow$ Sampling $N \times N$ $\longrightarrow W(k,l)$

Blurred & Noisy Image

$g(m,n)$ $\longrightarrow \otimes$ ($z(m,n)$) $\longrightarrow$ Zero Filling $N \times N$ $\longrightarrow$ DFT $N \times N$ $\longrightarrow$

$\longrightarrow \otimes$ ($W(k,l)$) $\longrightarrow$ Inverse DFT $N \times N$ $\longrightarrow \widehat{f}(m,n)$

Algebraic Approches

Signal	Image
$f(t) \longrightarrow \boxed{h(t)} \longrightarrow g(t)$	$f(x,y) \longrightarrow \boxed{h(x,y)} \longrightarrow g(x,y)$

Discretization

$$\Downarrow$$

$$\mathbf{g} = \mathbf{H}\mathbf{f}$$

- **Ideal case: H** invertible $\longrightarrow \widehat{\mathbf{f}} = \mathbf{H}^{-1}\mathbf{g}$
- $M > N$ **Least Squares**:

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \epsilon$$

$$e = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 = [\mathbf{g} - \mathbf{H}\mathbf{f}]'[\mathbf{g} - \mathbf{H}\widehat{\mathbf{f}}]$$

$$\widehat{\mathbf{f}} = \arg\min_{\mathbf{f}} \{e\}$$

$$\nabla e = -2\mathbf{H}'[\mathbf{g} - \mathbf{H}\mathbf{f}] = 0 \longrightarrow \mathbf{H}'\mathbf{H}\mathbf{f} = \mathbf{H}'\mathbf{g}$$

- If $\mathbf{H}'\mathbf{H}$ is invertible $\widehat{\mathbf{f}} = (\mathbf{H}'\mathbf{H})^{-1}\mathbf{H}'\mathbf{g}$

Algebraic Approches: Generalized Inversion

$$\widehat{\mathbf{f}} = \mathbf{W}\mathbf{g}$$

$\mathbf{W}$ is such that: $e = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 = [\mathbf{g} - \mathbf{H}\mathbf{f}]'[\mathbf{g} - \mathbf{H}\mathbf{f}]$ is minimum.
The solution may not be unique.

Generalized Inversion $\widehat{\mathbf{f}} = \mathbf{H}^{+}\mathbf{g}$
$\mathbf{H}^{+}$ is such that:

$$e = [\mathbf{g} - \mathbf{H}^{+}\mathbf{f}]'[\mathbf{g} - \mathbf{H}^{+}\mathbf{f}] = \|\mathbf{g} - \mathbf{H}^{+}\mathbf{f}\|^2$$

is minimum and that $\|\mathbf{f}\|^2$ minimum too.

- $\widehat{\mathbf{f}} = \mathbf{H}^{+}\mathbf{g}$ also minimizes

$$e = [\mathbf{f} - \widehat{\mathbf{f}}]'[\mathbf{f} - \widehat{\mathbf{f}}] = \text{Tr}\left\{[\mathbf{f} - \widehat{\mathbf{f}}][\mathbf{f} - \widehat{\mathbf{f}}]'\right\} = \text{Tr}\left\{\mathbf{f}\mathbf{f}'[\mathbf{I} - \mathbf{H}^{+}\mathbf{H}]'\right\}$$

- Estimation is perfect if $\mathbf{H}^{+}\mathbf{H} = \mathbf{I}$.

Algebraic Approches: Generalized Inversion

General case of $[M, N]$ matrix $\mathbf{H}$:

- if $M = N$ and $\operatorname{rang}\{\mathbf{H}\} = N$ then $\mathbf{H}^{+} = \mathbf{H}^{-1}$
- if $M > N$ and $\operatorname{rang}\{\mathbf{H}\} = N$ then $\mathbf{H}^{+} = (\mathbf{H}'\mathbf{H})^{-1}\mathbf{H}'$
- if $M < N$ and $\operatorname{rang}\{\mathbf{H}\} = M$ then $\mathbf{H}^{+} = \mathbf{H}'(\mathbf{H}\mathbf{H}')^{-1}$
- if $\operatorname{rang}\{\mathbf{H}\} = K < \inf M, N$ then
  - Singular Value Decomposition (SVD)
  - Iterative methods
  - Recursive methods

Generalized inversion by Singular Value Decomposition (SVD)

$$\mathbf{g} = \mathbf{H}\,\mathbf{f}$$

$$\begin{array}{lll} \mathbf{H}'\mathbf{H}\,\mathbf{v}_j = \lambda_j^2\mathbf{v}_j, \quad j = 1,\ldots,n & \mathbf{v}_i \text{ Eigen Vectors of } \mathbf{H}'\mathbf{H} \\ \mathbf{H}\mathbf{H}'\,\mathbf{u}_i = \lambda_i^2\mathbf{u}_k, \quad i = 1,\ldots,m & \mathbf{u}_j \text{ Eigen Vectors of } \mathbf{H}\mathbf{H}' \end{array}$$

$$\begin{array}{l} \mathbf{H}\,\lambda_j\,\mathbf{u}_j, \quad j = 1,\ldots,n \\ \mathbf{H}'\,\mathbf{u}_i = \lambda_i\,\mathbf{v}_i, \quad i = 1,\ldots,m \end{array}$$

$$\mathbf{H} = \mathbf{U}\,\boldsymbol{\Lambda}\,\mathbf{V}'$$

$$\mathbf{U} = \left[\mathbf{u}_1, \vdots, \mathbf{u}_2, \vdots, \cdots\right], \quad \mathbf{V} = \left[\mathbf{v}_1, \vdots, \mathbf{v}_2, \vdots, \cdots\right], \quad \boldsymbol{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \cdots)$$

- Truncation order r

$$\mathbf{H} = \sum_{m=1}^{r} \lambda_m \mathbf{v}_m \mathbf{u}_m' \longrightarrow \mathbf{H}^+ = \sum_{m=1}^{r} \frac{1}{\lambda_m} \mathbf{v}_m\,\mathbf{u}_m' \longrightarrow \mathbf{f}^+ = \sum_{m=1}^{r} \frac{\mathbf{v}_m'\,\mathbf{g}}{\lambda_m} \mathbf{u}_m$$

Generalized inversion: Iterative Algorithms

- **Bialy Algorithm** If $M = N$

$$\begin{cases} \mathbf{f}_0 = \mathbf{0} \\ \mathbf{f}_{n+1} = \mathbf{f}_n + \alpha(\mathbf{g} - \mathbf{H}\mathbf{f}_n), \quad \text{with} \;\; 0 < \alpha < 2/\|\mathbf{H}\| \end{cases}$$

- **Landweber Algorithm**

$$\begin{cases} \mathbf{f}_0 = \mathbf{0} \\ \mathbf{f}_{n+1} = \mathbf{f}_n + \alpha\mathbf{H}'(\mathbf{g} - \mathbf{H}\mathbf{f}_n), \quad \text{with} \;\; 0 < \alpha < 2/\|\mathbf{H}'\mathbf{H}\| \end{cases}$$

- **Van–Cittert Algorithm**

$$\begin{cases} \mathbf{f}_0 = \mathbf{0} \\ \mathbf{f}_{n+1} = \mathbf{f}_n + \alpha\mathbf{H}'(\mathbf{g} - \mathbf{H}\mathbf{T}\mathbf{f}_n), \quad \text{with} \;\; 0 < \alpha < 2/\|\mathbf{T}'\mathbf{H}'\mathbf{H}\mathbf{T}\| \end{cases}$$

where $\mathbf{T}$ is a truncation operator

$$\mathbf{f}_n = \alpha \sum_{k=0}^{n} [\mathbf{I} - \alpha\mathbf{H}'\mathbf{H}\mathbf{T}]^k \mathbf{H}'\mathbf{g}$$

n filters $[\mathbf{I} - \alpha\mathbf{H}'\mathbf{H}\mathbf{T}]$

Generalized inversion: Iterative Algorithms

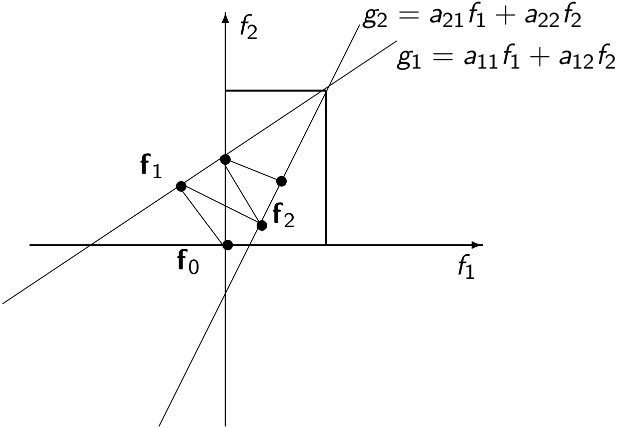
- **Kaczmarz Algorithm**
  - uses the data one by one as they arrived
  - is based on the more general Projection On Convex Sets (POCS)
  - $\mathbf{g} = \mathbf{H}\,\mathbf{f} \longrightarrow y_i = [\mathbf{Hf}] = \sum_j H_{ij}\, f_j =< \mathbf{h}_i, \mathbf{f} >$
  - each data is considered as a constraint
  - in Compted Tomography, it is called Algebraic Reconstruction Technics (ART)

$$\begin{cases} \mathbf{f}_0 = \mathbf{0} \\ \mathbf{f}_{n+1} = C(\mathbf{g}, \mathbf{f}_n) = c_m(\cdots c_2(c_1(\mathbf{f}))\cdots) \\ c_i(\mathbf{f}^{k+1}) = \mathbf{f}^{k+1} + \frac{g_i - \langle \mathbf{f}, \mathbf{h}_i \rangle}{\langle \mathbf{h}, \mathbf{h}_i \rangle}\, \mathbf{h}_i \quad i = 1, 2, \ldots, M \end{cases}$$

where $\mathbf{h}_i$ is the ith line of the matrix $\mathbf{H}$.

C is a set of other constraints such as positivity

Kaczmarz or ART Algorithms



Regularization

Regularization of an **ill-posed** inverse problem is its transformation to a **well-posed** problem, *i.e.*defining a **unique solution** for **any possible data** and insure that this solution is **stable** with respect to **errors** in those data.

A **regularizer** of $\mathbf{g} = \mathbf{H}\,\mathbf{f}$ is a family of of operators

$$\{R_\alpha; \alpha \in \Lambda \subset \mathbf{R}^+\}$$

such that:

- $\forall \mathbf{f} \in \mathcal{F}, \lim_{\alpha\to 0} \|R_\alpha \mathbf{H}\mathbf{f} - \mathbf{f}\|^2 = 0$
- $\forall \alpha \in \Lambda, R_\alpha$ is a continuous operator of $\mathcal{G}$ to $\mathcal{F}$

$$\Downarrow$$

Approximate Solution

$$\mathbf{f}_\epsilon = R_\alpha \mathbf{g}_\epsilon, \quad \text{with} \quad \mathbf{g}_\epsilon = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon}$$

$$\|\mathbf{f}_\epsilon - \mathbf{H}^{-1}\mathbf{g}\| \leq \underbrace{\|R_\alpha \mathbf{g} - \mathbf{H}^{-1}\mathbf{g}\|}_{\text{regularization error}} + \underbrace{\|R_\alpha(\mathbf{g}_\epsilon - \mathbf{g})\|}_{\text{error due to the noise}}$$

Regularization by Truncated SV Decomposition

$$\mathbf{g} = \mathbf{H}\,\mathbf{f}$$

$$\mathbf{H}'\,\mathbf{H}\,\mathbf{v}_j = \lambda_j^2 \mathbf{v}_j, \quad j = 1, \ldots, n \qquad \{\mathbf{u}_i\} \text{ eigen values of } \mathbf{H}'\mathbf{H}$$
$$\mathbf{H}\,\mathbf{H}'\,\mathbf{u}_i = \lambda_i^2 \mathbf{u}_k, \quad i = 1, \ldots, m \qquad \{\mathbf{v}_j\} \text{ eigen values of } \mathbf{H}\mathbf{H}'$$

$$\mathbf{H}\,\mathbf{v}_j = \lambda_j\,\mathbf{u}_j, \quad j = 1, \ldots, n$$
$$\mathbf{H}'\,\mathbf{u}_i = \lambda_i\,\mathbf{v}_i, \quad i = 1, \ldots, m$$

$$\mathbf{H} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}'$$

$$\mathbf{U} = \left[\mathbf{u}_1, \vdots, \mathbf{u}_2, \vdots, \ldots\right], \quad \mathbf{V} = \left[\mathbf{v}_1, \vdots, \mathbf{v}_2, \vdots, \ldots\right], \quad \boldsymbol{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \cdots)$$

- Truncation order k

$$\Lambda^+ = \{\alpha_i\}, \quad \begin{cases} \alpha_i = \frac{1}{\lambda_i} & \text{si} \quad \lambda_i \neq 0 (\text{significatif}) \\ \alpha_i = 0 & \text{si} \quad \lambda_i \simeq 0 (\text{non–significatif}) \end{cases}$$

$$\mathbf{g} + \delta\mathbf{g} = H(\mathbf{f} + \delta\mathbf{f}) \longrightarrow \frac{||\delta\mathbf{f}||}{||\mathbf{f}||} \leq \left(\frac{\lambda_{max}}{\lambda_{min}}\right)^2 \frac{||\delta\mathbf{g}||}{||\mathbf{g}||}$$

Regularization

- Stopping rule in Iterative method
  - Establish a stopping rule based on the residual error

$$\|\mathbf{H}\widehat{\mathbf{f}}^{k} - \mathbf{g}_{\epsilon}\| > \delta \text{ pour } i < k$$

$$\|\mathbf{H}\widehat{\mathbf{f}}^{k} - \mathbf{g}_{\epsilon}\| < \delta \text{ pour } i > k$$

  - $\frac{1}{k}$ plays the role of the regularization parameter
- Introduction of additional constraints: $\mathbf{C}$

$$\mathbf{f} = \mathbf{C}\mathbf{f} \quad \text{iff} \quad \mathbf{f} \quad \text{satisfait la contrainte}$$

Example : Positivity

$$\mathbf{C}\mathbf{f} = \begin{cases} \mathbf{f} & \text{if} \quad \mathbf{f} > 0 \\ 0 & \text{sinon} \end{cases}$$

$$\mathbf{g} = \mathbf{H}\,\mathbf{f} \longrightarrow \mathbf{g} = \mathbf{H}\mathbf{C}\mathbf{f}$$

At each iteration replace **H** by **HC**

Regularization for Continuous / Discretized problems

- **Tikhonov** (Regularization for continuous problems)
- **Philipps and Twomey** (Regularization for discretized problems)

Continuous case:

$$J_\alpha(f) = \|\mathcal{H}f - g\|_Y^2 + \alpha\Omega(f) \quad \alpha > 0$$

$$\Omega(f) = \int \sum_i w_i(t)[f^{(i)}(t)]^2 \,\mathrm{d}t$$

where $f^{(i)}(t)$ is the ith derivative of $f(t)$ and $w_i(t)$ postive, continuous and derivable functions.

Example : $i = 1$: first derivative

$$J_\alpha(f) = \|\mathcal{H}f - g\|_Y^2 + \alpha \int |f'(t)|^2 \,\mathrm{d}t$$

Regularization for Continuous / Discretized problems

Discrete case:

$$J_\alpha(\mathbf{f}) = [\mathbf{Hf}-\mathbf{g}]'[\mathbf{Hf}-\mathbf{g}] + \alpha[\mathbf{Df}]'[\mathbf{Df}] = \|\mathbf{Hf}-\mathbf{g}\|^2 + \alpha\|\mathbf{Df}\|^2$$

$$DCb = \begin{bmatrix} 1 & 0 & \cdots & & \cdots & 0 \\ -1 & 1 & \ddots & & & \vdots \\ 0 & -1 & 1 & \ddots & & \vdots \\ & 0 & -1 & 1 & \ddots & \\ & & & & & \\ 0 & & & 0 & -1 & 1 \end{bmatrix} \text{ or } \mathbf{D} = \begin{bmatrix} 1 & 0 & \cdots & & \cdots & 0 \\ -2 & 1 & \ddots & & & \vdots \\ 1 & -2 & 1 & \ddots & & \vdots \\ & 1 & -2 & 1 & \ddots & \\ & & & & & \\ 0 & & & 1 & -2 & 1 \end{bmatrix}$$

$$\nabla J_\alpha = 2\mathbf{H}'[\mathbf{Hf}-\mathbf{g}]' + 2\alpha\mathbf{D}'\mathbf{Df} = 0$$

$$[\mathbf{H}'\mathbf{H} + \alpha\mathbf{D}'\mathbf{D}]\widehat{\mathbf{f}} = \mathbf{H}'\mathbf{g} \longrightarrow \widehat{\mathbf{f}} = [\mathbf{H}'\mathbf{H} + \alpha\mathbf{D}'\mathbf{D}]^{-1}\mathbf{H}'\mathbf{g}$$

Regularization Algorithmes

$$\text{minimize} \quad J(\mathbf{f}) = Q(\mathbf{f}) + \lambda\Omega(\mathbf{f})$$

$$\text{with} \quad Q(\mathbf{f}) = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 = [\mathbf{g} - \mathbf{H}\mathbf{f}]'[\mathbf{g} - \mathbf{H}\mathbf{f}]$$

$$=$$

minimize $\Omega(\mathbf{f})$ subj. to the constraint

$$e = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 = [\mathbf{g} - \mathbf{H}\mathbf{f}]'[\mathbf{g} - \mathbf{H}\mathbf{f}] < \epsilon$$

A priori Information:

- Smoothnesse

$$\Omega(\mathbf{f}) = [\mathbf{D}\mathbf{f}]'[\mathbf{D}\mathbf{f}] = \|\mathbf{D}\mathbf{f}\|^2$$

$$\widehat{\mathbf{f}} = [\mathbf{H}'\mathbf{H} + \lambda\mathbf{D}'\mathbf{D}]^{-1}\mathbf{H}'\mathbf{g}$$

- Positivity:
  $\Omega(\mathbf{f}) =$ a nonquadratique function of $\mathbf{f}$
  No explicite solution

Regularization Algorithmes: 3 main approaches

$$\widehat{\mathbf{f}} = [\mathbf{H}'\mathbf{H} + \lambda \mathbf{D}'\mathbf{D}]^{-1}\mathbf{H}'\mathbf{g}$$

Computation of $\widehat{\mathbf{f}} = [\mathbf{H}'\mathbf{H} + \lambda \mathbf{D}'\mathbf{D}]^{-1}\mathbf{H}'\mathbf{g}$

- Circulante matrix approximation:
  when **H** and **D** are Toeplitz, they can be approximated by the circulant matrices
- Iterative methods:

$$\widehat{\mathbf{f}} = \arg\min_{\mathbf{f}} \left\{ \| J(\mathbf{f}) = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 + \lambda \mathbf{D}\mathbf{f}\|^2 \right\}$$

- Recursive methods:
  $\widehat{\mathbf{f}}$ at iteration k is computed as a function of $\widehat{\mathbf{f}}$ at previous iteration with one less data.

Regularization algorithms: Circulant approximation

1D Deconvolution:

$$\mathbf{g} = \mathbf{H}\,\mathbf{f} + \epsilon$$

H Toeplitz matrix

$$\widehat{\mathbf{f}} = \arg\min_{\mathbf{f}} \{\mathbf{f}\}\, J(\mathbf{f}) = Q(\mathbf{f}) + \lambda\Omega(\mathbf{f})$$

$Q(\mathbf{f}) = \|\mathbf{g}{-}\mathbf{H}\mathbf{f}\|^2 = [\mathbf{g}{-}\mathbf{H}\mathbf{f}]'[\mathbf{g}{-}\mathbf{H}\mathbf{f}]$ and $\Omega(\mathbf{f}) = \|\mathbf{C}\mathbf{f}\|^2 = [\mathbf{C}\mathbf{f}]'[\mathbf{C}\mathbf{f}]$

C a convolution matrix with the following impulse response

$$\mathbf{h}_1 = [1, -2, 1] \quad \longrightarrow \quad x(i) = x(i+1) - 2x(i) + x(i-1)$$

$$\Omega(\mathbf{f}) = \sum_{j=1}^{N} (x(i+1) - 2x(i) + x(i-1))^2 = \|\mathbf{C}\mathbf{f}\|^2 = \mathbf{f}'\mathbf{C}'\mathbf{C}\mathbf{f}$$

Solution :

$$\widehat{\mathbf{f}} = [\mathbf{H}'\mathbf{H} + \lambda\mathbf{C}'\mathbf{C}]^{-1} H'\mathbf{g}$$

Regularization algorithms: Circulant approximation

Main Idea : expand the vectors $\mathbf{f}$, $\mathbf{h}$ and $\mathbf{g}$ by the zeros to obtain $\mathbf{g}_e = \mathbf{H}_e \mathbf{f}_e$ with $\mathbf{H}_e$ a circulante matrix

$$\mathbf{f}_e(i) = \begin{cases} x(i) & i = 1, \ldots, Nx \\ 0 & i = Nx + 1, \ldots, P \geq Nx + Nh - 1 \end{cases}$$

$$y_e(i) = \begin{cases} y(i) & i = 1, \ldots, Ny \\ 0 & i = Ny + 1, \ldots, P \end{cases}$$

$$h_e(i) = \begin{cases} h(i) & i = 1, \ldots, Nh \\ 0 & i = Nh + 1, \ldots, P \end{cases}$$

$$y_e(k) = \sum_{i=0}^{Nh-1} x_e(k-i) h_e(i) \quad \longrightarrow \quad \mathbf{g}_e = \mathbf{H}_e \mathbf{f}_e$$

with $\mathbf{H}_e$ a circulante matrix whcich can diagonalized by FFT

Regularization algorithms: Circulant approximation

$$\mathbf{H}_e = \mathbf{F}\Lambda\mathbf{F}^{-1} \text{ with } \mathbf{F}[k,l] = \exp\left\{j2\pi\frac{kl}{P}\right\} \quad \mathbf{F}^{-1}[k,l] = \frac{1}{P}\exp\left\{-j2\pi\frac{kl}{P}\right\}$$

$$\boldsymbol{\Lambda} = \text{diag}[\lambda_1, \ldots, \lambda_P] \text{ and } [\lambda_1, \ldots, \lambda_P] = \text{TFD } [h_1, \ldots, h_{Nh}, 0, \ldots, 0]$$

$$c = [1, -2, 1] \quad c_e(i) = \begin{cases} c(i) & i = 1, \ldots, 3 \\ 0 & i = 4, \ldots, P \end{cases}$$

$$\widehat{\mathbf{f}} = [\mathbf{H}'\mathbf{H} + \lambda\mathbf{C}'\mathbf{C}]^{-1}\mathbf{H}'\mathbf{g} \longrightarrow \mathbf{F}\widehat{\mathbf{f}}_e = [\Lambda_h'\Lambda_h + \lambda\Lambda_c'\Lambda_c]^{-1}\Lambda_h'\mathbf{F}\mathbf{g}$$

$$\text{TFD } \{\mathbf{f}_e\} = [\Lambda_h'\Lambda_h + \lambda\Lambda_c'\Lambda_c]^{-1}\Lambda_h'\text{TFD } \{\mathbf{g}\}$$

$$\widehat{\mathbf{f}}(\omega) = \frac{1}{H(\omega)}\frac{|H(\omega)|^2}{|H(\omega)|^2 + \lambda|C(\omega)|^2}\, y(\omega)$$

Link with Wiener filter: $C(\omega) = N(\omega)/X(\omega)$

Image Restoration

C Convolution matrix with the following impulse response:

$$H_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{aligned}\Omega(\mathbf{f}) = \sum\sum \quad & (f(i+1,j) + f(i-1,j) \\ & + f(i+1,j+1) + f(i-1,j+1) - 4f(i,j))^2\end{aligned}$$

$$x_e(k,l) = \begin{cases} x(k,l) & k = 1,\ldots,K \qquad l = 1,\ldots,L \\ 0 & k = K+1,\ldots,P \quad l = L+1,\ldots,P \end{cases}$$

Regularization: Iterative methods: Gradient based

$$\widehat{\mathbf{f}} = \arg\min_{\mathbf{f}} \{J(\mathbf{f}) = Q(\mathbf{f}) + \lambda\Omega(\mathbf{f})\}$$

Let note : $\mathbf{g}^k = \nabla J\left(\mathbf{f}^k\right)$ gradient, $\mathbf{H}^k = \nabla^2 J\left(\mathbf{f}^k\right)$ Hessien.

First order gradient methods

- fixed step:

$$\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \alpha \mathbf{g}^{(k)} \qquad \alpha \quad \text{fixe}$$

- Optimal or steepest descente step:

$$\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \alpha^{(k)} \mathbf{g}^{(k)}$$

$$\alpha^{(k)} = -\frac{\mathbf{g}^{(k)t}\mathbf{g}^{(k)}}{\mathbf{g}^{(k)t}\mathbf{H}^k\mathbf{g}^{(k)}} = \frac{||\mathbf{g}^k\|^2}{\|\mathbf{g}^k\|_H^2}$$

Regularization: Iterative methods: Conjugate Gradient

- **Conjugate Gradient (CG)**

$$\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)} \quad \alpha^{(k)} = -\frac{\mathbf{d}^{(k)t} \mathbf{g}^{(k)}}{\mathbf{d}^{(k)t} \mathbf{H}^{k} \mathbf{d}^{(k)}}$$

$$\mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} + \beta^{(k)} \mathbf{g}^{(k)} \quad \beta^{(k)} = -\frac{\mathbf{g}^{(k)t} \mathbf{g}^{(k)}}{\mathbf{g}^{(k-1)t} \mathbf{g}^{(k-1)}}$$

- **Newton method**

$$\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + (\mathbf{H}^{(k)})^{-1} \mathbf{g}^{(k)}$$

- Advantages : $\Omega(\mathbf{f})$ can be any convexe function
- Limitations : Computational cost

Regularization: Recursive algorithms

$$\widehat{\mathbf{f}} = [\mathbf{H}'\mathbf{H} + \lambda \mathbf{D}]^{-1}\mathbf{H}'\mathbf{g}$$

Main idea: Express $\mathbf{f}_{i+1}$ as a function of $\mathbf{f}_i$

$$\mathbf{f}_{i+1} = (\mathbf{H}'_{i+1}\mathbf{H}_{i+1} + \alpha \mathbf{D})^{-1}\mathbf{H}'_{i+1}\mathbf{g}_{i+1}$$

$$\mathbf{f}_i = (\mathbf{H}_i^t\mathbf{H}_i + \alpha \mathbf{D})^{-1}\mathbf{H}_i^t\mathbf{g}_i$$

$$\Downarrow$$

$$\mathbf{f}_{i+1} = (\mathbf{H}_i^t\mathbf{H}_i + \mathbf{h}_{i+1}\mathbf{h}'_{i+1} + \alpha \mathbf{D})^{-1}(\mathbf{H}_i^t\mathbf{g}_i - \mathbf{h}_{i+1}\mathbf{g}_i + 1)$$

Noting:

$$\mathbf{P}_i = (\mathbf{H}_i^t\mathbf{H}_i + \alpha \mathbf{D})^{-1} \quad \text{and} \quad \mathbf{P}_{i+1}^t = \mathbf{P}_i^t + \mathbf{h}_{i+1}\mathbf{h}'_{i+1}$$

$$\Downarrow$$

$$\mathbf{f}_{i+1} = \mathbf{f}_i + \mathbf{P}_{i+1}\mathbf{h}_{i+1}(\mathbf{g}_{i+1} - \mathbf{h}'_{i+1}\mathbf{f}_i)$$

$$\mathbf{P}_{i+1} = \mathbf{P}_i - \mathbf{P}_i\mathbf{h}_{i+1}(\mathbf{h}'_{i+1}\mathbf{P}_i\mathbf{H}_{i+1} + \alpha)^{-1}\mathbf{h}'_{i+1}\mathbf{P}_i$$

Regularization: Wiener Filtering with FIR

$$g(m,n) = \sum_{i,j=-N}^{N} h(i,j)\, f(m-i,n-j) + b(m,n)$$

$$\widehat{f}(m,n) = \sum_{i,j=-M}^{M} w(i,j)\, f(m-i,n-j)$$

$$\mathrm{EQM} = \mathrm{E}\left\{(\widehat{f}(m,n) - f(m,n))^2\right\}$$

Orthogonality condition:

$$\mathrm{E}\left\{(\widehat{f}(m,n) - f(m,n))\, g(m-k,n-l)\right\} = 0, \quad \forall k,l = -M,\ldots,M$$

$(2M+1)^2$ linear equations

$$R_{fg}(k,l) - \sum_{i,j=-M}^{M} w(i,j)\, R_{ff}(k-i,l-j) = 0, \quad \forall k,l = -M,\ldots,M$$

Regularization: Wiener Filtering

Linearity and white Gaussian noise variance σ_b^2 :

$$R_{gg}(k,l) = R_{ff}(k,l) * a(k,l) + \sigma_b^2 \delta(k,l)$$

$$a(k,l) \overset{\triangle}{=} h(k,l) \star h(k,l) = \sum_{i,j=-\infty}^{\infty}\sum h(i,j)\, h(i+k, j+k)$$

$$R_{fg}(k,l) = h(k,l) \star R_{ff}(k,l) = \sum_{i,j=-\infty}^{\infty}\sum h(i,j) R_{ff}(i+k, j+k)$$

$$r_0(k,l) \overset{\triangle}{=} \frac{R_{ff}(k,l)}{R_{ff}(0,0)} = \frac{R_{ff}(k,l)}{\sigma_f^2}$$

$$r_0(-k,-l) = r_0(k,l)$$

$$\left[\frac{\sigma_b^2}{\sigma_f^2}\delta(k,l) + r_0(k,l) \star a(k,l)\right] * w(k,l) = h(k,l) * r_0(k,l), \ \forall k,l = -M, \ldots, M$$

Regularization: Wiener Filtering

- Equation

$$\left[\frac{\sigma_b^2}{\sigma_f^2}\delta(k,l) + r_0(k,l) \star a(k,l)\right] * w(k,l) = h(k,l) * r_0(k,l)$$

- Matrix-Vector form:

$$\left[\frac{\sigma_b^2}{\sigma_f^2}\mathbf{I} + \mathbf{R}\right]\mathbf{g} = \mathbf{r}$$

- **R** is a Bloc–Toeplitz of $(2M+1) \times (2M+1)$ blocs, each bloc has the dimensions $(2M+1) \times (2M+1)$.
- **g** and **r** are two vectors of dimensions $(2M+1)^2 \times 1$ containing $g(k,l)$ and $h(k,l) * r_0(k,l)$.

A more direct presentation

$$g(m,n)=\sum_{i,j=-N}^{N}\sum h(i,j)\,f(m-i,n-j)+\epsilon(m,n)\longrightarrow \mathbf{g}=\mathbf{H}\mathbf{f}+\boldsymbol{\epsilon}$$

$$\widehat{f}(m,n)=\sum_{i,j=-M}^{M}\sum w(i,j)\,f(m-i,n-j)\longrightarrow \widehat{\mathbf{f}}=\mathbf{W}\mathbf{f}$$

$$\mathrm{EQM}=\mathrm{E}\left\{\sum_{i,j=-M}^{M}\sum(\widehat{f}(m,n)-f(m,n))^2\right\}=\mathrm{E}\left\{[\widehat{\mathbf{f}}-\mathbf{f}]'[\widehat{\mathbf{f}}-\mathbf{f}]\right\}$$

Orthogonality:

$$\mathrm{E}\left\{[\widehat{\mathbf{f}}-\mathbf{f}]\mathbf{g}'\right\}=0$$

$$\mathbf{W}=\mathrm{E}\left\{\mathbf{f}\mathbf{g}'\right\}[\mathrm{E}\left\{\mathbf{g}\mathbf{g}'\right\}]^{-1}=\mathbf{R}_{ff}\mathbf{H}'[\mathbf{H}\mathbf{R}_{ff}\mathbf{H}'+\mathbf{R}_{\epsilon\epsilon}]^{-1}$$

$$\mathbf{R}_{ff}=\mathrm{E}\left\{\mathbf{f}\mathbf{f}'\right\},\quad \mathbf{R}_{\epsilon\epsilon}=\mathrm{E}\left\{\boldsymbol{\epsilon}\boldsymbol{\epsilon}'\right\}\quad \mathbf{R}_{f\epsilon}=\mathrm{E}\left\{\mathbf{f}\boldsymbol{\epsilon}'\right\}=0$$

- Matrix Inversion Lemma:

$$\mathbf{W} = \mathbf{R}_{\mathit{ff}}\mathbf{H}'[\mathbf{H}\mathbf{R}_{\mathit{ff}}\mathbf{H}' + \mathbf{R}_{\epsilon\epsilon}]^{-1} = [\mathbf{H}'\mathbf{R}_{\epsilon\epsilon}^{-1}\mathbf{H} + \mathbf{R}_{\mathit{ff}}^{-1}]^{-1}\mathbf{H}'\mathbf{R}_{\epsilon\epsilon}^{-1}$$

- $\mathbf{R}_{\epsilon\epsilon} = \boldsymbol{\sigma}^2\mathbf{I}$

$$\mathbf{W} = \mathbf{R}_{\mathit{ff}}\mathbf{H}'[\mathbf{H}\mathbf{R}_{\mathit{ff}}\mathbf{H} + \sigma_b^2\mathbf{I}]^{-1} = [\mathbf{H}\mathbf{H}' + \sigma_b^2\mathbf{R}_{\mathit{ff}}^{-1}]^{-1}\mathbf{H}'$$

- Particular Case: $\sigma_b^2 = 0$

$$\mathbf{W} = \begin{cases} [\mathbf{H}'\mathbf{H}]^{-1}\mathbf{H}' & \longrightarrow \widehat{[\mathbf{f}} = \mathbf{H}'\mathbf{H}]^{-1}\mathbf{H}'\mathbf{g} \\ \mathbf{R}_{\mathit{ff}}\mathbf{H}'[\mathbf{H}\mathbf{R}_{\mathit{ff}}\mathbf{H}']^{-1} \longrightarrow & \longrightarrow \widehat{\mathbf{f}} = \mathbf{R}_{\mathit{ff}}\mathbf{H}'[\mathbf{H}\mathbf{R}_{\mathit{ff}}\mathbf{H}']^{-1}\mathbf{g} \end{cases}$$

Generalized Wiener filter using unitary transforms

$$\mathbf{W} = [\mathbf{H}'\mathbf{H} + \sigma_b^2 \mathbf{R}_{ff}^{-1}]^{-1}\mathbf{H}'$$

$$\mathbf{P} \overset{\triangle}{=} [\mathbf{H}'\mathbf{H} + \sigma_b^2 \mathbf{R}_{ff}^{-1}]^{-1}$$

Consider a unitary transform **F**, *i.e.* $\mathbf{F}'\mathbf{F} = \mathbf{F}\mathbf{F}' = \mathbf{I}$

$$\widehat{\mathbf{f}} = \mathbf{F}'[\mathbf{F}\mathbf{P}\mathbf{F}']\mathbf{F}\mathbf{H}'\mathbf{g} \overset{\triangle}{=} \mathbf{F}'\bar{\mathbf{P}}\mathbf{z}$$

$$\bar{\mathbf{P}} \overset{\triangle}{=} [\mathbf{F}\mathbf{P}\mathbf{F}'], \quad \mathbf{z} \overset{\triangle}{=} \mathbf{F}\mathbf{H}'\mathbf{g}$$

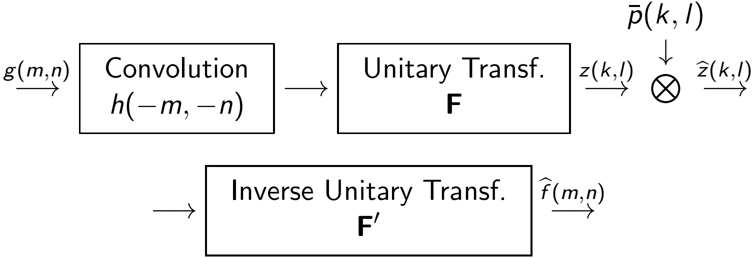
$$\mathbf{g} \longrightarrow \boxed{\mathbf{H}'} \longrightarrow \boxed{\mathbf{F}} \longrightarrow \mathbf{z} \longrightarrow \boxed{\bar{\mathbf{P}} = [\mathbf{F}\mathbf{P}\mathbf{F}']} \longrightarrow \widehat{\mathbf{z}} \longrightarrow \boxed{\mathbf{F}'} \longrightarrow \widehat{\mathbf{f}}$$

For an appropriate unitary transforms $\bar{\mathbf{P}}$ becomes an almost diagonal matrix

$$\widehat{\mathbf{z}} = \bar{\mathbf{P}}\mathbf{z} \quad \Longrightarrow \quad \widehat{z}(k,l) \simeq \bar{p}(k,l)\, z(k,l)$$

Generalized Wiener filter using unitary transforms

If **H** corresponds to convolution matrix $h(m,n)$, $\mathbf{H}'$ is also a convolution matrix with $h(-m,-n)$



$\bar{p}(k,l)$

$\xrightarrow{g(m,n)}$ Convolution $h(-m,-n)$ $\longrightarrow$ Unitary Transf. $\mathbf{F}$ $\xrightarrow{z(k,l)}$ $\bigotimes$ $\xrightarrow{\widehat{z}(k,l)}$

$\longrightarrow$ Inverse Unitary Transf. $\mathbf{F}'$ $\xrightarrow{\widehat{f}(m,n)}$

Content

1. Introduction: Signals and Images, Linear transformations (Convolution, Fourier, Laplace, Hilbert, Radon, ..., Discrete convolution, Z transform, DFT, FFT, ...)
2. Modeling: parametric and non-parametric, MA, AR and ARMA models, Linear prediction
3. Deconvolution and Parameter Estimation: Deterministic (LS, Regularization) and Probabilistic methods (Wiener Filtering)
4. **Inverse problems, Regularization and Bayesian estimation**
5. Kalman Filtering
6. Case study: Signal deconvolution
7. Case study: Image restoration
8. Case study: Image reconstruction and Computed Tomography

4. Inverse problems and Bayesian estimation

- Inverse problems? Why need for Bayesian approach?
- Probability ?
- Discrete and Continuous variables
- Bayes rule and Bayesian inference
- Maximum Likelihood method (ML) and its limitations
- Bayesian estimation theory
- Bayesian inference for inverse problems in signal and image processing
- Prior modeling
- Bayesian computation (Laplace approximation, MCMC, BVA)

Inverse problems : 3 main examples

- Example 1:
  Measuring variation of temperature with a therometer
    - $f(t)$ variation of temperature over time
    - $g(t)$ variation of length of the liquid in thermometer
- Example 2:
  Making an image with a camera, a microscope or a telescope
    - $f(x,y)$ real scene
    - $g(x,y)$ observed image
- Example 3: Making an image of the interior of a body
    - $f(x,y)$ a section of a real 3D body $f(x,y,z)$
    - $g_\phi(r)$ a line of observed radiographe $g_\phi(r,z)$

- Example 1: Deconvolution
- Example 2: Image restoration
- Example 3: Image reconstruction

Measuring variation of temperature with a therometer

- $f(t)$ variation of temperature over time
- $g(t)$ variation of length of the liquid in thermometer

- Forward model: Convolution

$$g(t) = \int f(t')\, h(t-t')\, \mathrm{d}t' + \epsilon(t)$$

$h(t)$: impulse response of the measurement system

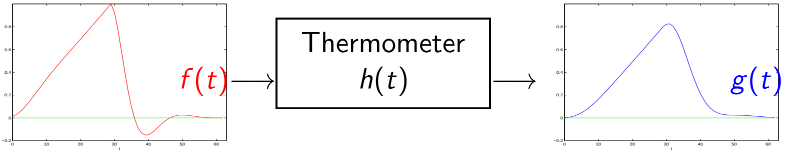
- Inverse problem: Deconvolution

Given the forward model $\mathcal{H}$ (impulse response $h(t)$))
and a set of data $g(t_i), i = 1, \cdots, M$
find $f(t)$

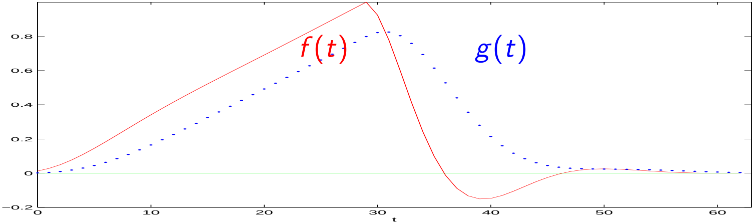
Measuring variation of temperature with a therometer

Forward model: Convolution

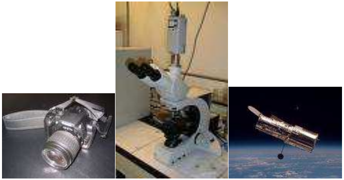
$$g(t) = \int f(t') \, h(t - t') \, \mathrm{d}t' + \epsilon(t)$$



Inversion: Deconvolution



Making an image with a camera, a microscope or a telescope



- $f(x,y)$ real scene
- $g(x,y)$ observed image
- Forward model: Convolution

$$g(x,y) = \iint f(x',y')\, h(x-x',y-y')\, \mathrm{d}x'\, \mathrm{d}y' + \epsilon(x,y)$$

  $h(x,y)$: Point Spread Function (PSF) of the imaging system

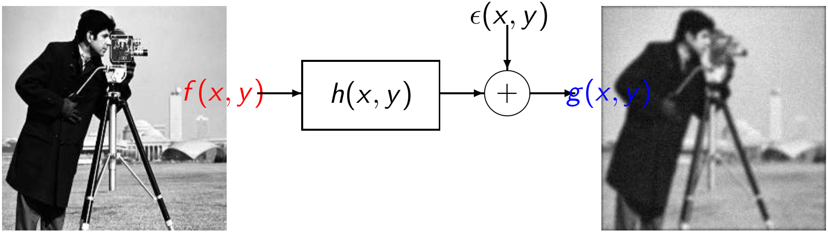
- Inverse problem: Image restoration

  Given the forward model $\mathcal{H}$ (PSF $h(x,y)$))
  and a set of data $g(x_i,y_i), i=1,\cdots,M$
  find $f(x,y)$

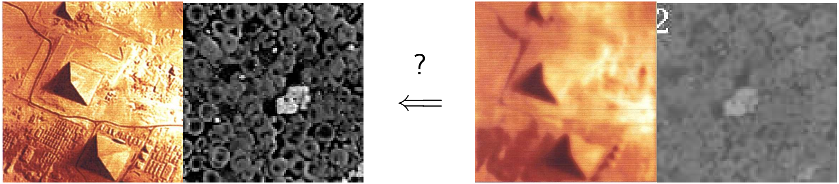
Making an image with an unfocused camera

Forward model: 2D Convolution

$$g(x,y)=\iint f(x',y')\,h(x-x',y-y')\,\mathrm{d}x'\,\mathrm{d}y'+\epsilon(x,y)$$

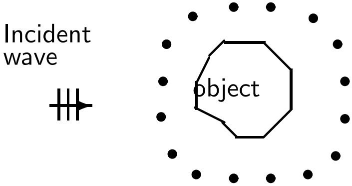


Inversion: Deconvolution

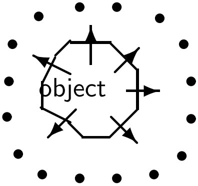


Making an image of the interior of a body

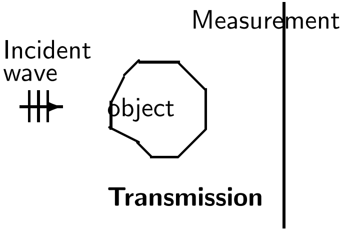
Different imaging systems:



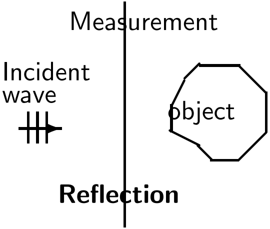
Active Imaging



Passive Imaging



Transmission



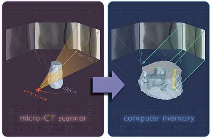
Reflection

Forward problem: Knowing the object predict the data

Inverse problem: From measured data find the object

Making an image of the interior of a body

- $f(x,y)$ a section of a real 3D body $f(x,y,z)$
- $g_\phi(r)$ a line of observed radiographe $g_\phi(r,z)$



- Forward model:
  Line integrals or Radon Transform

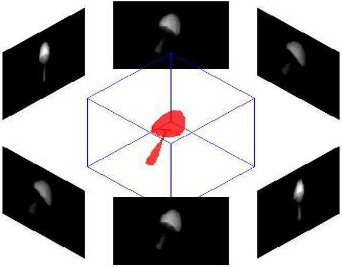
$$
\begin{aligned}
g_\phi(r) &= \int_{L_{r,\phi}} f(x,y)\,\mathrm{d}l + \epsilon_\phi(r) \\
&= \iint f(x,y)\,\delta(r - x\cos\phi - y\sin\phi)\,\mathrm{d}x\,\mathrm{d}y + \epsilon_\phi(r)
\end{aligned}
$$

- Inverse problem: Image reconstruction

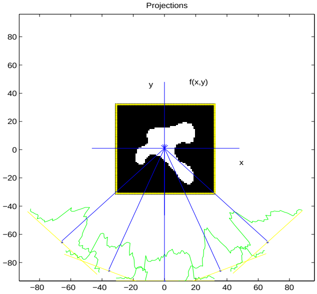
  Given the forward model $\mathcal{H}$ (Radon Transform) and
  a set of data $g_{\phi_i}(r), i = 1, \cdots, M$
  find $f(x,y)$

2D and 3D Computed Tomography

3D



2D



$$g_\phi(r_1, r_2) = \int_{\mathcal{L}_{r_1,r_2,\phi}} f(x, y, z) \, \mathrm{d}l \qquad g_\phi(r) = \int_{\mathcal{L}_{r,\phi}} f(x, y) \, \mathrm{d}l$$

Forward probelm: $f(x, y)$ or $f(x, y, z) \longrightarrow g_\phi(r)$ or $g_\phi(r_1, r_2)$

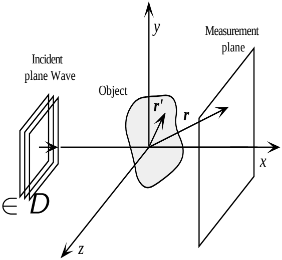
Inverse problem: $g_\phi(r)$ or $g_\phi(r_1, r_2) \longrightarrow f(x, y)$ or $f(x, y, z)$

Microwave or ultrasound imaging

Measurs: diffracted wave by the object $g(\mathbf{r}_i)$

Unknown quantity: $f(\mathbf{r}) = k_0^2(n^2(\mathbf{r}) - 1)$

Intermediate quantity : $\phi(\mathbf{r})$

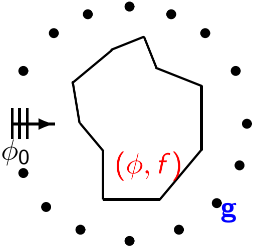


$$g(\mathbf{r}_i) = \iint_D G_m(\mathbf{r}_i, \mathbf{r}')\phi(\mathbf{r}') f(\mathbf{r}') \, \mathrm{d}\mathbf{r}', \; \mathbf{r}_i \in S$$

$$\phi(\mathbf{r}) = \phi_0(\mathbf{r}) + \iint_D G_o(\mathbf{r}, \mathbf{r}')\phi(\mathbf{r}') f(\mathbf{r}') \, \mathrm{d}\mathbf{r}', \; \mathbf{r} \in D$$

Born approximation $(\phi(\mathbf{r}') \simeq \phi_0(\mathbf{r}'))$ **):**

$$g(\mathbf{r}_i) = \iint_D G_m(\mathbf{r}_i, \mathbf{r}')\phi_0(\mathbf{r}') f(\mathbf{r}') \, \mathrm{d}\mathbf{r}', \; \mathbf{r}_i \in S$$



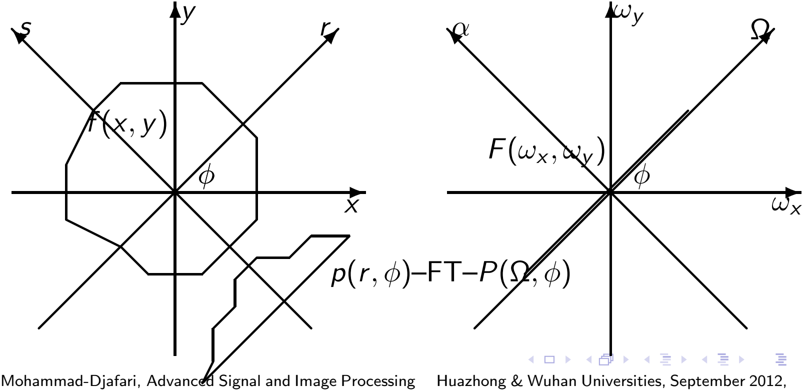
Discretization :

$$\begin{cases} \mathbf{g} = \mathbf{G}_m \mathbf{F} \phi \\ \phi = \phi_0 + \mathbf{G}_o \mathbf{F} \phi \end{cases} \longrightarrow \begin{cases} \mathbf{g} = \mathbf{H}(\mathbf{f}) \\ \text{with} \quad \mathbf{F} = \mathrm{diag}(\mathbf{f}) \\ \mathbf{H}(\mathbf{f}) = \mathbf{G}_m \mathbf{F} (\mathbf{I} - \mathbf{G}_o \mathbf{F})^{-1} \phi_0 \end{cases}$$

Fourier Synthesis in X ray Tomography

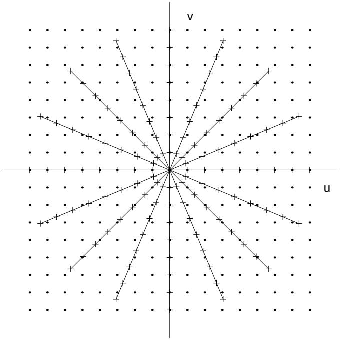
$$\begin{aligned} g(r,\phi) &= \iint f(x,y)\,\delta(r - x\cos\phi - y\sin\phi)\,\mathrm{d}x\,\mathrm{d}y \\ G(\Omega,\phi) &= \int g(r,\phi)\exp\{-j\Omega r\}\;\mathrm{d}r \\ F(\omega_x,\omega_y) &= \iint f(x,y)\exp\{-j\omega_x x,\omega_y y\}\;\mathrm{d}x\,\mathrm{d}y \end{aligned}$$

$$F(\omega_x,\omega_y) = G(\Omega,\phi) \qquad \text{for} \quad \omega_x = \Omega\cos\phi \quad \text{and} \quad \omega_y = \Omega\sin\phi$$



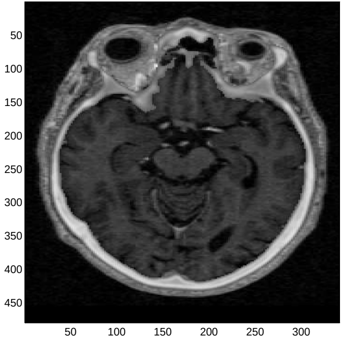
Fourier Synthesis in X ray tomography

$$G(\omega_x, \omega_y) = \iint f(x, y) \exp\left\{-j\left(\omega_x x + \omega_y y\right)\right\} \, \mathrm{d}x \, \mathrm{d}y$$



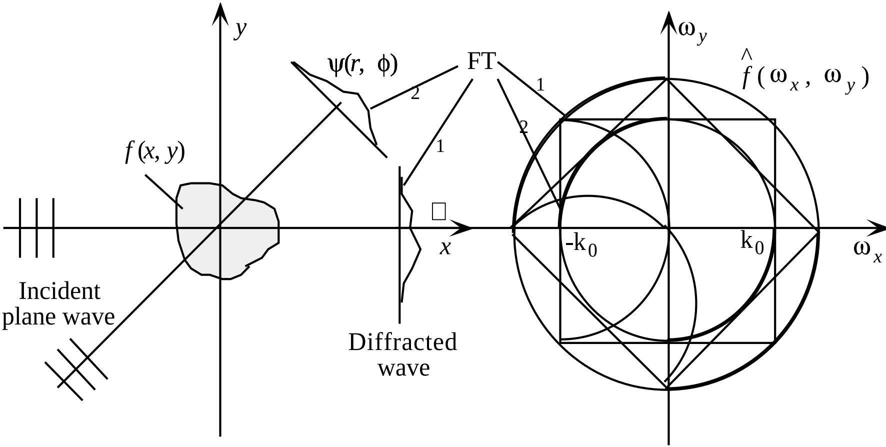
?

$\Longrightarrow$



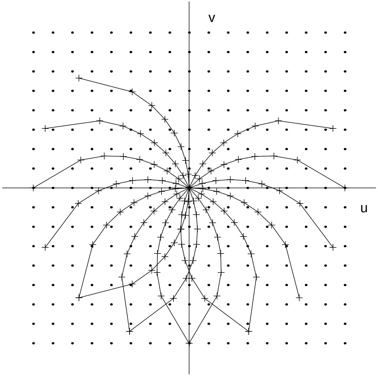
Forward problem: Given $f(x, y)$ compute $G(\omega_x, \omega_y)$
Inverse problem: Given $G(\omega_x, \omega_y)$ on those lines estimate $f(x, y)$

Fourier Synthesis in Diffraction tomography



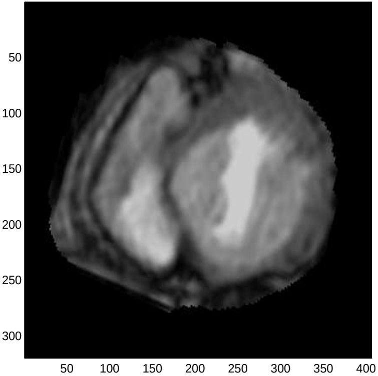
Fourier Synthesis in Diffraction tomography

$$G(\omega_x, \omega_y) = \iint f(x, y) \exp\left\{-j\left(\omega_x x + \omega_y y\right)\right\} \, \mathrm{d}x \, \mathrm{d}y$$



?

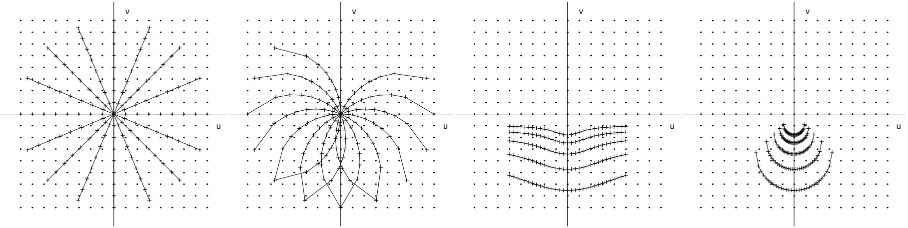
$\Longrightarrow$



Forward problem: Given $f(x, y)$ compute $G(\omega_x, \omega_y)$
Inverse problem : Given $G(\omega_x, \omega_y)$ on those semi cercles estimate $f(x, y)$

Fourier Synthesis in different imaging systems

$$G(\omega_x,\omega_y) = \iint f(x,y) \exp\left\{-j\left(\omega_x x + \omega_y y\right)\right\}\, \mathrm{d}x\, \mathrm{d}y$$



X ray Tomography   Diffraction   Eddy current   SAR & Radar

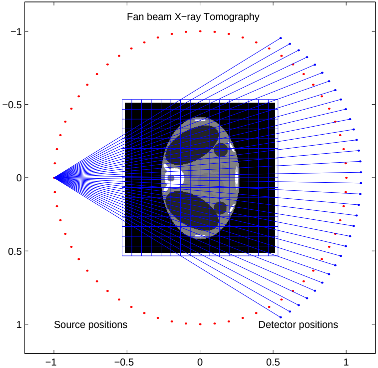
Forward problem: Given $f(x,y)$ compute $G(\omega_x,\omega_y)$
Inverse problem : Given $G(\omega_x,\omega_y)$ on those algebraic lines, cercles or curves, estimate $f(x,y)$

Invers Problems: other examples and applications

- X ray, Gamma ray Computed Tomography (CT)
- Microwave and ultrasound tomography
- Positron emission tomography (PET)
- Magnetic resonance imaging (MRI)
- Photoacoustic imaging
- Radio astronomy
- Geophysical imaging
- Non Destructive Evaluation (NDE) and Testing (NDT) techniques in industry
- Hyperspectral imaging
- Earth observation methods (Radar, SAR, IR, ...)
- Survey and tracking in security systems

Computed tomography (CT)

A Multislice CT Scanner



$$g(s_i) = \int_{L_i} f(\mathbf{r}) \; \mathrm{d}l_i + \epsilon(s_i)$$

Discretization

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \epsilon$$

Positron emission tomography (PET)

Magnetic resonance imaging (MRI)

Nuclear magnetic resonance imaging (NMRI), Para-sagittal MRI of the head

Radio astronomy (interferometry imaging systems)

The Very Large Array in New Mexico, an example of a radio telescope.

General formulation of inverse problems

- General non linear inverse problems:

$$g(\mathbf{s}) = [\mathcal{H} f(\mathbf{r})](\mathbf{s}) + \epsilon(\mathbf{s}), \quad \mathbf{r} \in \mathcal{R}, \quad \mathbf{s} \in \mathcal{S}$$

- Linear models:

$$g(\mathbf{s}) = \int f(\mathbf{r})\, h(\mathbf{r},\mathbf{s})\, \mathrm{d}\mathbf{r} + \epsilon(\mathbf{s})$$

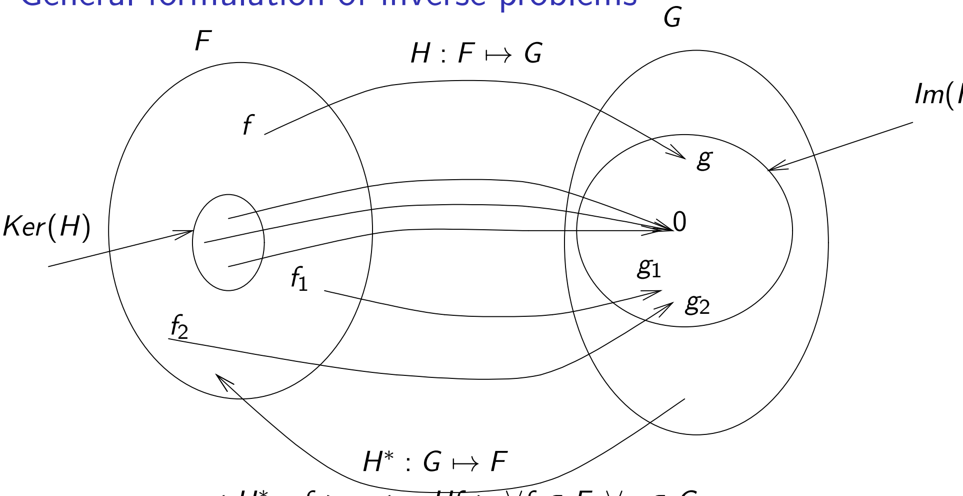
  If $h(\mathbf{r},\mathbf{s}) = h(\mathbf{r}-\mathbf{s}) \longrightarrow$ Convolution.

- Discrete data:

$$g(\mathbf{s}_i) = \int h(\mathbf{s}_i,\mathbf{r})\, f(\mathbf{r})\, \mathrm{d}\mathbf{r} + \epsilon(\mathbf{s}_i), \quad i = 1,\cdots,m$$

- Inversion: Given the forward model $\mathcal{H}$ and the data $\mathbf{g} = \{g(\mathbf{s}_i),\, i = 1,\cdots,m)\}$ estimate $f(\mathbf{r})$
- Well-posed and Ill-posed problems (Hadamard): existance, uniqueness and stability
- Need for prior information

General formulation of inverse problems



$$< H^* g, f > = < g, Hf > \forall f \in F, \forall g \in G$$

Analytical methods (mathematical physics)

$$g(\mathbf{s}_i) = \int h(\mathbf{s}_i, \mathbf{r})\, f(\mathbf{r})\, \mathrm{d}\mathbf{r} + \epsilon(\mathbf{s}_i), \quad i = 1, \cdots, m$$

$$g(\mathbf{s}) = \int h(\mathbf{s}, \mathbf{r})\, f(\mathbf{r})\, \mathrm{d}\mathbf{r}$$

$$\widehat{f}(\mathbf{r}) = \int w(\mathbf{s}, \mathbf{r})\, g(\mathbf{s})\, \mathrm{d}\mathbf{s}$$

$w(\mathbf{s}, \mathbf{r})$ minimizing a criterion:

$$\begin{aligned} Q(w(\mathbf{s}, \mathbf{r})) &= \left\| g(\mathbf{s}) - [\mathcal{H}\,\widehat{f}(\mathbf{r})](\mathbf{s}) \right\|_2^2 = \int \left| g(\mathbf{s}) - [\mathcal{H}\,\widehat{f}(\mathbf{r})](\mathbf{s}) \right|^2 \, \mathrm{d}\mathbf{s} \\ &= \int \left| g(\mathbf{s}) - \int h(\mathbf{s}, \mathbf{r})\, \widehat{f}(\mathbf{r})\, \mathrm{d}\mathbf{r} \right|^2 \, \mathrm{d}\mathbf{s} \\ &= \int \left| g(\mathbf{s}) - \int\int h(\mathbf{s}, \mathbf{r}) w(\mathbf{s}, \mathbf{r})\, g(\mathbf{s})\, \mathrm{d}\mathbf{s}\, \mathrm{d}\mathbf{r} \right|^2 \, \mathrm{d}\mathbf{s} \end{aligned}$$

Trivial solution: $h(\mathbf{s}, \mathbf{r}) w(\mathbf{s}, \mathbf{r}) = \delta(\mathbf{r})\delta(\mathbf{s})$

Analytical methods

- Trivial solution:

$$w(\mathbf{s},\mathbf{r}) = h^{-1}(\mathbf{s},\mathbf{r})$$

  Example: Fourier Transform:

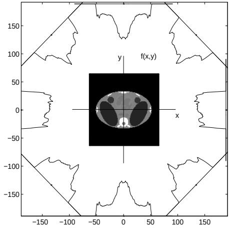
$$g(\mathbf{s}) = \int f(\mathbf{r}) \exp\left\{-j\mathbf{s}.\mathbf{r}\right\} \, \mathrm{d}\mathbf{r}$$

$$h(\mathbf{s},\mathbf{r}) = \exp\left\{-j\mathbf{s}.\mathbf{r}\right\} \longrightarrow w(\mathbf{s},\mathbf{r}) = \exp\left\{+j\mathbf{s}.\mathbf{r}\right\}$$

$$\hat{f}(\mathbf{r}) = \int g(\mathbf{s}) \exp\left\{+j\mathbf{s}.\mathbf{r}\right\} \, \mathrm{d}\mathbf{s}$$

- Known classical solutions for specific expressions of $h(\mathbf{s},\mathbf{r})$:
  - 1D cases: 1D Fourier, Hilbert, Weil, Melin, ...
  - 2D cases: 2D Fourier, Radon, ...

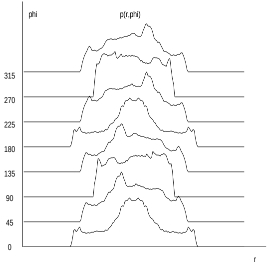
X ray Tomography



$$g(r,\phi) = -\ln\left(\frac{I}{I_0}\right) = \int_{L_{r,\phi}} f(x,y)\,\mathrm{d}l$$

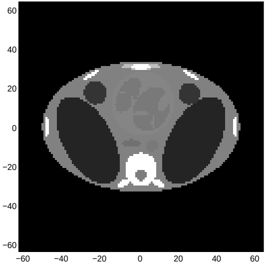
$$g(r,\phi) = \iint_D f(x,y)\,\delta(r - x\cos\phi - y\sin\phi)\,\mathrm{d}x\,\mathrm{d}y$$

$$f(x,y) \longrightarrow \boxed{\text{RT}} \longrightarrow g(r,\phi)$$

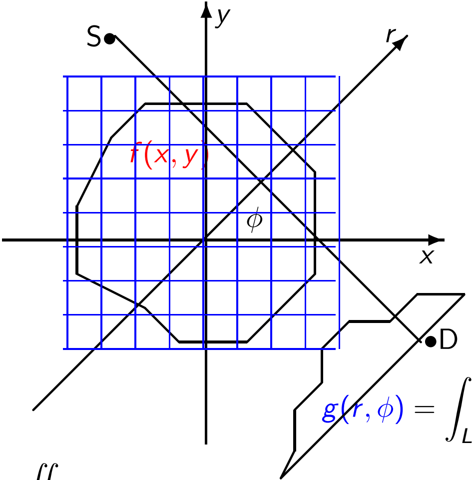


IRT ?

$$\Longrightarrow$$



Analytical Inversion methods



$$g(r,\phi) = \int_L f(x,y)\, \mathrm{d}l$$

Radon:

$$g(r,\phi) = \iint_D f(x,y)\, \delta(r - x\cos\phi - y\sin\phi)\, \mathrm{d}x\, \mathrm{d}y$$

$$f(x,y) = \left(-\frac{1}{2\pi^2}\right) \int_0^{\pi} \int_{-\infty}^{+\infty} \frac{\frac{\partial}{\partial r} g(r,\phi)}{(r - x\cos\phi - y\sin\phi)}\, \mathrm{d}r\, \mathrm{d}\phi$$

Filtered Backprojection method

$$f(x,y) = \left(-\frac{1}{2\pi^2}\right) \int_0^{\pi} \int_{-\infty}^{+\infty} \frac{\frac{\partial}{\partial r} g(r,\phi)}{(r - x\cos\phi - y\sin\phi)} \, \mathrm{d}r \, \mathrm{d}\phi$$

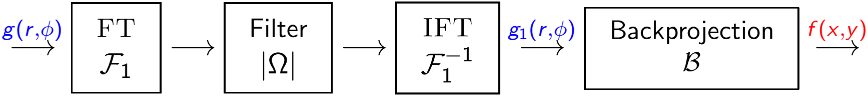
$$\text{Derivation } \mathcal{D} : \qquad \overline{g}(r,\phi) = \frac{\partial g(r,\phi)}{\partial r}$$

$$\text{Hilbert Transform} \mathcal{H} : \qquad g_1(r',\phi) = \frac{1}{\pi} \int_0^{\infty} \frac{\overline{g}(r,\phi)}{(r - r')} \, \mathrm{d}r$$

$$\text{Backprojection } \mathcal{B} : \ f(x,y) = \frac{1}{2\pi} \int_0^{\pi} g_1(r' = x\cos\phi + y\sin\phi, \phi) \, \mathrm{d}\phi$$

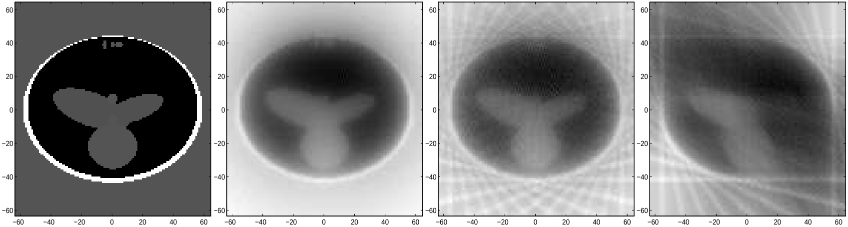
$$f(x,y) = \mathcal{B}\, \mathcal{H}\, \mathcal{D}\, g(r,\phi) = \mathcal{B}\, \mathcal{F}_1^{-1}\, |\Omega|\, \mathcal{F}_1\, g(r,\phi)$$

- Backprojection of filtered projections:



$$\xrightarrow{g(r,\phi)} \boxed{\begin{matrix}\text{FT} \\ \mathcal{F}_1\end{matrix}} \longrightarrow \boxed{\begin{matrix}\text{Filter} \\ |\Omega|\end{matrix}} \longrightarrow \boxed{\begin{matrix}\text{IFT} \\ \mathcal{F}_1^{-1}\end{matrix}} \xrightarrow{g_1(r,\phi)} \boxed{\begin{matrix}\text{Backprojection} \\ \mathcal{B}\end{matrix}} \xrightarrow{f(x,y)}$$

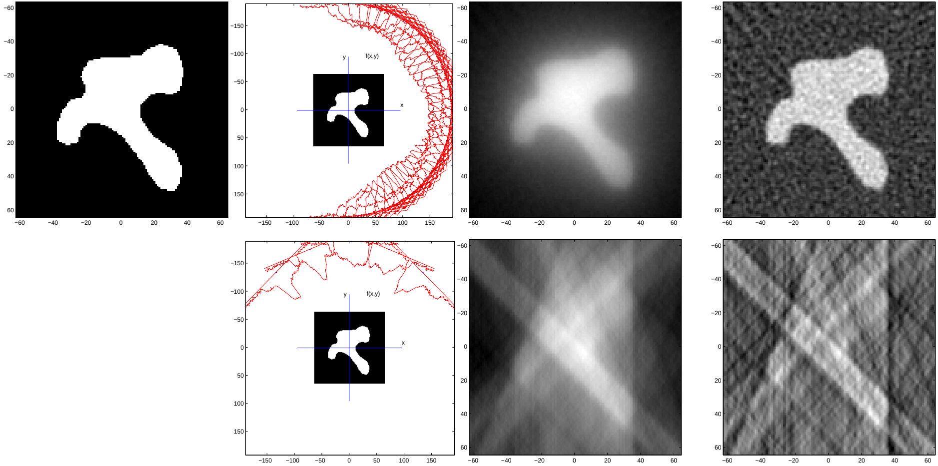
Limitations : Limited angle or noisy data



Original 64 proj. 16 proj. 8 proj. $[0, \pi/2]$

- Limited angle or noisy data
- Accounting for detector size
- Other measurement geometries: fan beam, ...

Limitations : Limited angle or noisy data



Original Data Backprojection Filtered Backprojectio

Parametric methods

- $f(\mathbf{r})$ is described in a parametric form with a very few number of parameters $\boldsymbol{\theta}$ and one searches $\widehat{\boldsymbol{\theta}}$ which minimizes a criterion such as:
- Least Squares (LS): $Q(\boldsymbol{\theta}) = \sum_i |g_i - [\mathcal{H} f(\boldsymbol{\theta})]_i|^2$
- Robust criteria : $Q(\boldsymbol{\theta}) = \sum_i \phi(|g_i - [\mathcal{H} f(\boldsymbol{\theta})]_i|)$
  with different functions ϕ $(L_1$, Hubert, ...).
- Likelihood : $\mathcal{L}(\boldsymbol{\theta}) = -\ln p(\mathbf{g}|\boldsymbol{\theta})$
- Penalized likelihood : $\mathcal{L}(\boldsymbol{\theta}) = -\ln p(\mathbf{g}|\boldsymbol{\theta}) + \lambda\Omega(\boldsymbol{\theta})$

Examples:

- Spectrometry: $f(t)$ modelled as a sum og gaussians
  $f(t) = \sum_{k=1}^{K} a_k \mathcal{N}(t|\mu_k, v_k)$ $\quad \boldsymbol{\theta} = \{a_k, \mu_k, v_k\}$
- Tomography in CND: $f(x, y)$ is modelled as a superposition of circular or elleiptical discs $\quad \boldsymbol{\theta} = \{a_k, \mu_k, r_k\}$

Non parametric methods

$$g(\mathbf{s}_i) = \int h(\mathbf{s}_i, \mathbf{r})\, f(\mathbf{r})\, \mathrm{d}\mathbf{r} + \epsilon(\mathbf{s}_i), \quad i = 1, \cdots, M$$

- $f(\mathbf{r})$ is assumed to be well approximated by

$$f(\mathbf{r}) \simeq \sum_{j=1}^{N} f_j\, b_j(\mathbf{r})$$

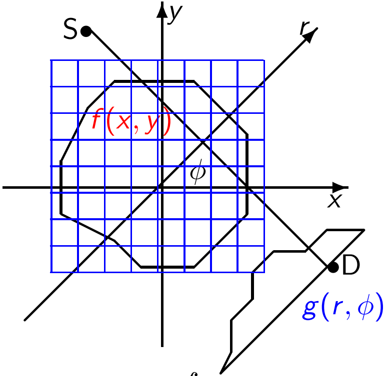
with $\{b_j(\mathbf{r})\}$ a basis or any other set of known functions

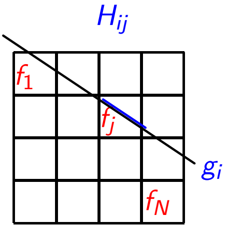
$$g(\mathbf{s}_i) = g_i \simeq \sum_{j=1}^{N} f_j \int h(\mathbf{s}_i, \mathbf{r})\, b_j(\mathbf{r})\, \mathrm{d}\mathbf{r}, \quad i = 1, \cdots, M$$

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon} \ \text{ with } \ H_{ij} = \int h(\mathbf{s}_i, \mathbf{r})\, b_j(\mathbf{r})\, \mathrm{d}\mathbf{r}$$

- $\mathbf{H}$ is huge dimensional
- LS solution : $\widehat{\mathbf{f}} = \arg\min_{\mathbf{f}} \{Q(\mathbf{f})\}$ with $Q(\mathbf{f}) = \sum_i |g_i - [\mathbf{H}\mathbf{f}]_i|^2 = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2$ does not give satisfactory result.

Algebraic methods: Discretization





$$f(x,y)=\sum_j f_j\, b_j(x,y)$$

$$b_j(x,y)=\begin{cases}1 & \text{if } (x,y)\in \text{ pixel } j\\ 0 & \text{else}\end{cases}$$

$$g(r,\phi)=\int_L f(x,y)\,\mathrm{d}l$$

$$g_i=\sum_{j=1}^{N} H_{ij}\; f_j+\epsilon_i$$

$$\mathbf{g}=\mathbf{H}\mathbf{f}+\epsilon$$

Inversion: Deterministic methods

Data matching

- Observation model
  $g_i = h_i(\mathbf{f}) + \epsilon_i, \quad i = 1, \ldots, M \longrightarrow \mathbf{g} = \mathbf{H}(\mathbf{f}) + \boldsymbol{\epsilon}$
- Misatch between data and output of the model $\Delta(\mathbf{g}, \mathbf{H}(\mathbf{f}))$

$$\widehat{\mathbf{f}} = \arg\min_{\mathbf{f}} \{\Delta(\mathbf{g}, \mathbf{H}(\mathbf{f}))\}$$

- Examples:

– LS $\quad \Delta(\mathbf{g}, \mathbf{H}(\mathbf{f})) = \|\mathbf{g} - \mathbf{H}(\mathbf{f})\|^2 = \sum_i |g_i - h_i(\mathbf{f})|^2$

– L_p $\quad \Delta(\mathbf{g}, \mathbf{H}(\mathbf{f})) = \|\mathbf{g} - \mathbf{H}(\mathbf{f})\|^p = \sum_i |g_i - h_i(\mathbf{f})|^p, \quad 1 < p < 2$

– KL $\quad \Delta(\mathbf{g}, \mathbf{H}(\mathbf{f})) = \sum_i g_i \ln \frac{g_i}{h_i(\mathbf{f})}$

- In general, does not give satisfactory results for inverse problems.

Regularization theory

Inverse problems = Ill posed problems

$\longrightarrow$ Need for prior information

Functional space (Tikhonov):

$$g = \mathcal{H}(f) + \epsilon \longrightarrow J(f) = ||g - \mathcal{H}(f)||_2^2 + \lambda||\mathcal{D}f||_2^2$$

Finite dimensional space (Philips & Towmey): $\mathbf{g} = \mathbf{H}(\mathbf{f}) + \boldsymbol{\epsilon}$

- Minimum norme LS (MNLS): $J(\mathbf{f}) = ||\mathbf{g} - \mathbf{H}(\mathbf{f})||^2 + \lambda||\mathbf{f}||^2$
- Classical regularization: $J(\mathbf{f}) = ||\mathbf{g} - \mathbf{H}(\mathbf{f})||^2 + \lambda||\mathbf{D}\mathbf{f}||^2$
- More general regularization:

$$J(\mathbf{f}) = \mathcal{Q}(\mathbf{g} - \mathbf{H}(\mathbf{f})) + \lambda\Omega(\mathbf{D}\mathbf{f})$$

or

$$J(\mathbf{f}) = \Delta_1(\mathbf{g}, \mathbf{H}(\mathbf{f})) + \lambda\Delta_2(\mathbf{f}, \mathbf{f}_\infty)$$

Limitations:

- Errors are considered implicitly white and Gaussian
- Limited prior information on the solution
- Lack of tools for the determination of the hyperparameters

Inversion: Probabilistic methods

Taking account of errors and uncertainties ⟶ Probability theory

- Maximum Likelihood (ML)
- Minimum Inaccuracy (MI)
- Probability Distribution Matching (PDM)
- Maximum Entropy (ME) and Information Theory (IT)
- Bayesian Inference (BAYES)

Advantages:

- Explicit account of the errors and noise
- A large class of priors via explicit or implicit modeling
- A coherent approach to combine information content of the data and priors

Limitations:

- Practical implementation and cost of calculation

Bayesian estimation approach

$$\mathcal{M}: \qquad \mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon}$$

- Observation model $\mathcal{M}$ + Hypothesis on the noise $\boldsymbol{\epsilon} \longrightarrow$
  $p(\mathbf{g}|\mathbf{f};\mathcal{M}) = p_{\boldsymbol{\epsilon}}(\mathbf{g} - \mathbf{H}\mathbf{f})$
- A priori information $\quad p(\mathbf{f}|\mathcal{M})$
- Bayes : $\quad p(\mathbf{f}|\mathbf{g};\mathcal{M}) = \dfrac{p(\mathbf{g}|\mathbf{f};\mathcal{M})\, p(\mathbf{f}|\mathcal{M})}{p(\mathbf{g}|\mathcal{M})}$

Link with regularization :

Maximum A Posteriori (MAP) :

$$\begin{aligned} \widehat{\mathbf{f}} &= \arg\max_{\mathbf{f}} \{p(\mathbf{f}|\mathbf{g})\} = \arg\max_{\mathbf{f}} \{p(\mathbf{g}|\mathbf{f})\ p(\mathbf{f})\} \\ &= \arg\min_{\mathbf{f}} \{-\ln p(\mathbf{g}|\mathbf{f}) - \ln p(\mathbf{f})\} \end{aligned}$$

with $\quad Q(\mathbf{g}, \mathbf{H}\mathbf{f}) = -\ln p(\mathbf{g}|\mathbf{f}) \quad$ and $\quad \lambda\Omega(\mathbf{f}) = -\ln p(\mathbf{f})$

Case of linear models and Gaussian priors

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon}$$

- Hypothesis on the noise: $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_\epsilon^2 \mathbf{I}) \longrightarrow$
  $p(\mathbf{g}|\mathbf{f}) \propto \exp\left\{-\frac{1}{2\sigma_\epsilon^2}\|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2\right\}$
- Hypothesis on $\mathbf{f}$: $\mathbf{f} \sim \mathcal{N}(0, \sigma_f^2 (\mathbf{D}'\mathbf{D})^{-1}) \longrightarrow$
  $p(\mathbf{f}) \propto \exp\left\{-\frac{1}{2\sigma_f^2}\|\mathbf{D}\mathbf{f}\|^2\right\}$
- A posteriori:
  $$p(\mathbf{f}|\mathbf{g}) \propto \exp\left\{-\frac{1}{2\sigma_\epsilon^2}\|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 - \frac{1}{2\sigma_f^2}\|\mathbf{D}\mathbf{f}\|^2\right\}$$
- MAP : $\widehat{\mathbf{f}} = \arg\max_{\mathbf{f}} \{p(\mathbf{f}|\mathbf{g})\} = \arg\min_{\mathbf{f}} \{J(\mathbf{f})\}$
  with $J(\mathbf{f}) = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 + \lambda \|\mathbf{D}\mathbf{f}\|^2, \qquad \lambda = \frac{\sigma_\epsilon^2}{\sigma_f^2}$
- Advantage : characterization of the solution

$$\mathbf{f}|\mathbf{g} \sim \mathcal{N}(\widehat{\mathbf{f}}, \widehat{\mathbf{P}}) \;\text{ with }\; \widehat{\mathbf{f}} = \widehat{\mathbf{P}}\mathbf{H}'\mathbf{g}, \quad \widehat{\mathbf{P}} = \left(\mathbf{H}'\mathbf{H} + \lambda \mathbf{D}'\mathbf{D}\right)^{-1}$$

MAP estimation with other priors:

$$\widehat{\mathbf{f}} = \arg\min_{\mathbf{f}} \{J(\mathbf{f})\} \quad \text{with} \quad J(\mathbf{f}) = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 + \lambda\Omega(\mathbf{f})$$

Separable priors:

- Gaussian: $p(f_j) \propto \exp\left\{-\alpha|f_j|^2\right\} \longrightarrow \Omega(\mathbf{f}) = \alpha \sum_j |f_j|^2$
- Gamma: $p(f_j) \propto f_j^{\alpha} \exp\left\{-\beta f_j\right\} \longrightarrow \Omega(\mathbf{f}) = \alpha \sum_j \ln f_j + \beta f_j$
- Beta:
  $p(f_j) \propto f_j^{\alpha}(1-f_j)^{\beta} \longrightarrow \Omega(\mathbf{f}) = \alpha \sum_j \ln f_j + \beta \sum_j \ln(1-f_j)$
- Generalized Gaussian:
  $p(f_j) \propto \exp\left\{-\alpha|f_j|^p\right\}, \quad 1 < p < 2 \longrightarrow \quad \Omega(\mathbf{f}) = \alpha \sum_j |f_j|^p,$

Markovian models:

$$p(f_j|\mathbf{f}) \propto \exp\left\{-\alpha \sum_{i \in N_j} \phi(f_j, f_i)\right\} \longrightarrow \quad \Omega(\mathbf{f}) = \alpha \sum_j \sum_{i \in N_j} \phi(f_j, f_i),$$

MAP estimation with markovien priors:

$$\widehat{\mathbf{f}} = \arg\min_{\mathbf{f}} \{J(\mathbf{f})\} \quad \text{with} \quad J(\mathbf{f}) = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 + \lambda\Omega(\mathbf{f})$$

$$\Omega(\mathbf{f}) = \sum_j \phi(f_j - f_{j-1})$$

with $\phi(t)$:

Convex functions:

$$|t|^\alpha, \ \sqrt{1+t^2} - 1, \ \log(\cosh(t)), \ \left\{ \begin{array}{ll} t^2 & |t| \le T \\ 2T|t| - T^2 & |t| > T \end{array} \right.$$

or Non convex functions:

$$\log(1+t^2), \quad \frac{t^2}{1+t^2}, \quad \arctan(t^2), \quad \left\{ \begin{array}{ll} t^2 & |t| \le T \\ T^2 & |t| > T \end{array} \right.$$

Main advantages of the Bayesian approach

- MAP = Regularization
- Posterior mean ? Marginal MAP ?
- More information in the posterior law than only its mode or its mean
- Meaning and tools for estimating hyper parameters
- Meaning and tools for model selection
- More specific and specialized priors, particularly through the hidden variables
- More computational tools:
  - Expectation-Maximization for computing the maximum likelihood parameters
  - MCMC for posterior exploration
  - Variational Bayes for analytical computation of the posterior marginals
  - ...

Content

1. Introduction: Signals and Images, Linear transformations (Convolution, Fourier, Laplace, Hilbert, Radon, ..., Discrete convolution, Z transform, DFT, FFT, ...)
2. Modeling: parametric and non-parametric, MA, AR and ARMA models, Linear prediction
3. Deconvolution and Parameter Estimation: Deterministic (LS, Regularization) and Probabilistic methods (Wiener Filtering)
4. Inverse problems, Regularization and Bayesian estimation
5. **Kalman Filtering**
6. Case study: Signal deconvolution
7. Case study: Image restoration
8. Case study: Image reconstruction and Computed Tomography

5. Kalman Filtering and smoothing

- Dynamical systems and state space modeling
- State space modeling examples
  - Electrical circuit example
  - Radar tracking of object
- Kalman filtering basics
- Kalman Filtering as recursive Bayesian estimation
- Kalman Filtering extensions: Adaptive signal processing
- Kalman Filtering extensions: Fast Kalman filtering
- Kalman Filtering for signal deconvolution

State space model: Continuous case

Dynamic systems:

- Single Input Single Output (SISO) system:

$$\left\{\begin{array}{lll} \dot{x}(t) & = A\,x(t) + B\,u(t) & \text{State equation} \\ y(t) & = C\,x(t) + D\,v(t) & \text{Observation equation} \end{array}\right.$$

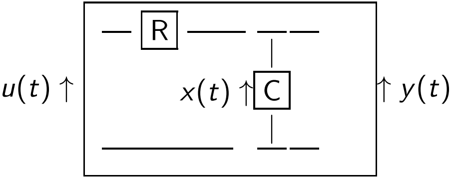
- Multiple Input Multiple Output (MIMO) system:

$$\left\{\begin{array}{lll} \dot{\mathbf{x}}(t) & = \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}\,\mathbf{u}(t) & \text{State equation} \\ \mathbf{y}(t) & = \mathbf{C}\,\mathbf{x}(t) + \mathbf{D}\,\mathbf{v}(t) & \text{Observation equation} \end{array}\right.$$

$\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are the matrices of the system.

State space modeling

- A SISO system
- A simple electric system



- State space model

$$u(t) = R\, i(t) + v_c(t) = RC\, \dot{v}_c(t) + v_c(t) = RC\, \dot{x}(t) + x(t)$$

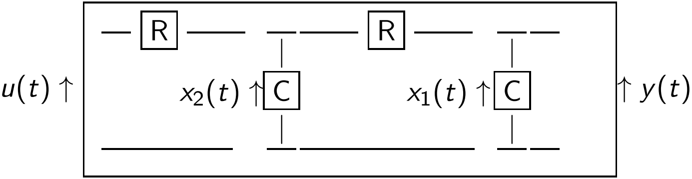
$$\begin{cases} \dot{x}(t) & = \left(\frac{-1}{RC}\right) x(t) + \left(\frac{-1}{RC}\right) u(t) \\ y(t) & = x(t) \end{cases}$$

- $RC = 1$:

$$\begin{cases} \dot{x}(t) = -x(t) - u(t) \\ y(t) = x(t) \end{cases} \rightarrow LT \rightarrow \begin{cases} pX(p) + X(p) = U(p) \rightarrow X(p) = \frac{1}{p+1} U(p) \\ y(t) = e^{-t} * u(t) = h(t) * u(t) \end{cases}$$

State space modeling

- A more complex electric system example



- State space model

$$u(t) = RC\,\dot{x}_2(t) + x_2(t), \quad x_2(t) = RC\,\dot{x}_1(t) + x_1(t)$$

- $RC = 1$:

$$\left\{ \begin{array}{lcl} \left[\begin{array}{c} \dot{x}_1(t) \\ \dot{x}_2(t) \end{array}\right] & = & \left[\begin{array}{cc} -1 & 1 \\ 0 & -1 \end{array}\right] \left[\begin{array}{c} x_1(t) \\ x_2(t) \end{array}\right] + \left[\begin{array}{c} 0 \\ 1 \end{array}\right] u(t) \\ y(t) & = & \left[\begin{array}{c} 1 \\ 0 \end{array}\right] x_1(t) \end{array} \right.$$

Input-Output model

- Linear Systems
  - Single Input Single Output (SISO) systems

$$y(t) = \int h(t,\tau)\, u(\tau)\, \mathrm{d}\tau$$

  - Multi Input Multi Output (MIMO) systems

$$\mathbf{y}(t) = \int \mathbf{H}(t,\tau)\, \mathbf{u}(\tau)\, \mathrm{d}\tau$$

- Linear Time Invariant System
  - SISO Convolution

$$y(t) = h(t) * u(t) = \int h(t-\tau)\, u(\tau)\, \mathrm{d}\tau$$

  - MIMO Convolution

$$\mathbf{y}(t) = \int \mathbf{H}(t-\tau)\, \mathbf{u}(\tau)\, \mathrm{d}\tau$$

  - Impulse response $h(t)$ or $\mathbf{H}(t) = \left[\begin{array}{ccc} . & . & . \\ . & h_{ij}(t) & . \\ . & . & . \end{array}\right]$

Link between the two modeling

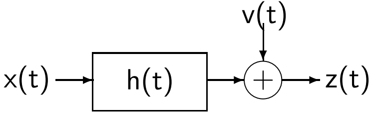
$$\begin{cases} \dot{\mathbf{x}}(t) & = \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}\,\mathbf{u}(t) \quad \text{State equation} \\ \mathbf{y}(t) & = \mathbf{C}\,\mathbf{x}(t) + \mathbf{D}\,\mathbf{v}(t) \quad \text{Observation equation} \end{cases}$$

- $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$
- $\mathbf{x}(t+\Delta t) \simeq \mathbf{x}(t) + \Delta t\,\mathbf{A}\mathbf{x}(t) = (\mathbf{I} + \Delta t\,\mathbf{A})\mathbf{x}(t)$
- $\mathbf{x}(t+n\Delta t) \simeq (\mathbf{I} + \Delta t\,\mathbf{A})^n\,\mathbf{x}(t)$
- $\lim_{\Delta t \mapsto 0}(\mathbf{I} + \Delta t\,\mathbf{A})^{t/\Delta t} = \exp\{t\mathbf{A}\}$
- $\dot{\mathbf{x}}(t) = \mathbf{A}\,\mathbf{x}(t)$ with $\mathbf{x}(0) = 0 \longrightarrow \mathbf{x}(t) = \exp\{t\mathbf{A}\}$

State space and input-Output modeling

Transmission of a AR1 signal through a perturbed channel

$$u(t) \longrightarrow \boxed{\quad \text{AR1} \quad} \longrightarrow x(t)$$



$$x(t) = a\,x(t-1) + u(t) \qquad\qquad z(t) = \sum_k h_k\,x(t-k) + v(t)$$

- A FIR channel
  $\mathbf{h} = [h_0, h_1, \cdots, h_{p-1}]' \longrightarrow z(n) = \sum_{k=0}^{p} h_k x(n-k) + v(n)$

$$\left\{\begin{array}{lcl} x_{n+1} & = & a\,x_n + u_n \\ z_n & = & \displaystyle\sum_{k=0}^{p-1} h_n\,x_{n-k} + v_n \end{array}\right. \rightarrow ? \rightarrow \left\{\begin{array}{lcl} \mathbf{x}_{n+1} & = & \mathbf{F}\,\mathbf{x}_n + \mathbf{G}u_n \\ z_n & = & \mathbf{H}\,\mathbf{x}_n + v_n \end{array}\right.$$

State space and input-Output modeling

- A FIR channel $z(n) = \sum_{k=0}^{p} h_k x(n-k) + v(n)$

$$\begin{aligned} \mathbf{x}_n &= [x_n, x_{n-1}, \cdots, x_{n-p+1}]' \\ \mathbf{x}_{n+1} &= [x_{n+1}, x_n, \cdots, x_{n-p+2}]' \end{aligned}$$

$$\begin{bmatrix} x_{n+1} \\ x_n \\ x_{n-1} \\ \vdots \\ \vdots \\ x_{n-p+1} \end{bmatrix} = \begin{bmatrix} a & 0 & \dots & \dots & 0 \\ 1 & 0 & \dots & \dots & 0 \\ & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \\ \vdots & \vdots & \vdots & \vdots & \\ \vdots & \vdots & \vdots & 1 & 0 \end{bmatrix} \begin{bmatrix} x_n \\ x_{n-1} \\ x_{n-2} \\ \vdots \\ \vdots \\ x_{n-p+1} \\ x_{n-p+2} \end{bmatrix}$$

$$\begin{aligned} \mathbf{G} &= [1, 0, \cdots, 0]' \\ \mathbf{H} &= [h_0, h_1, \cdots, h_n, 0]' \end{aligned} \longrightarrow z_n = \mathbf{H}\,\mathbf{x}_n + v_n$$

State space and input-Output modeling

- A FIR channel $z(n) = \sum_{k=0}^{p} h_k x(n-k) + v(n)$

$$\left\{ \begin{array}{lcl} \mathbf{x}_{n+1} & = & \mathbf{F}\,\mathbf{x}_n + \mathbf{G} u_n \\ z_n & = & \mathbf{H}\,\mathbf{x}_n + v_n \end{array} \right.$$

$$\mathbf{x}_n = [x_n, x_{n-1}, \cdots, x_{n-p+1}]'$$

$$\mathbf{x}_{n+1} = [x_{n+1}, x_n, \cdots, x_{n-p+2}]'$$

$$\mathbf{G} = [1, 0, \cdots, 0]'$$

$$\mathbf{H} = [h_0, h_1, \cdots, h_n]'$$

$$\mathbf{F} = \begin{bmatrix} a & 0 & \ldots & \ldots & 0 \\ 1 & 0 & \ldots & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \\ \vdots & \vdots & \vdots & \vdots & \\ \vdots & \vdots & \vdots & 1 & 0 \end{bmatrix}$$

- A perfect but noisy channel $h(t) = h_0\,\delta(t) \longrightarrow p = 1$

$$\left\{ \begin{array}{lcl} x_{n+1} & = & a\,x_n + u_n \\ z_n & = & h_0\,x_n + v_n \end{array} \right.$$

State space modeling: Examples

$$\begin{cases} x_{n+1} & = & a\, x_n + u_n \\ z_n & = & h\, x_n + v_n \end{cases}$$

$$\begin{cases} u_n & \sim & \mathcal{N}(0, q) \\ v_n & \sim & \mathcal{N}(0, r) \\ x_0 & \sim & \mathcal{N}(m_0, p_0) \end{cases}$$

- Try to obtain $\widehat{x}_{n+1|n}$ as a function of $\widehat{x}_{n|n}$ recursively

$$\begin{cases} \widehat{x}_{n+1|n} & = & a\, \widehat{x}_{n|n} \\ \widehat{x}_{n|n} & = & \widehat{x}_{n|n-1} + k_n\, (z_n - h\, \widehat{x}_{n|n-1}) \\ k_n & = & \dfrac{p_{n|n-1}\, h}{h^2 p_{n|n-1} + r} = \dfrac{1}{h}\, \dfrac{p_{n|n-1}}{p_{n|n-1} + r/h^2} \end{cases}$$

$$\begin{cases} p_{n+1|n} & = & a^2\, p_{n|n} + q \\ p_{n|n} & = & \dfrac{1}{h}\, \dfrac{p_{n|n-1}}{\frac{r}{h^2}\, p_{n|n-1+1}} \end{cases}$$

State space modeling of the systems

- Time Varying systems:

$$\left\{\begin{array}{lcll} \mathbf{x}_{n+1} & = & \mathbf{f}_n(\mathbf{x}_n, \mathbf{u}_n) & \text{state equation} \\ \mathbf{z}_n & = & \mathbf{h}_n(\mathbf{x}_n, \mathbf{v}_n) & \text{observation equation} \end{array}\right.$$

- Time Variiying but Linear system

$$\left\{\begin{array}{lcl} \mathbf{x}_{n+1} & = & \mathbf{F}_n \, \mathbf{x}_n + \mathbf{G}_n \, \mathbf{u}_n \\ \mathbf{z}_n & = & \mathbf{H}_n \, \mathbf{x}_n + \mathbf{v}_n \end{array}\right.$$

- Time Invariant and Linear system

$$\left\{\begin{array}{lcl} \mathbf{x}_{n+1} & = & \mathbf{F} \, \mathbf{x}_n + \mathbf{G} \, \mathbf{u}_n \\ \mathbf{z}_n & = & \mathbf{H} \, \mathbf{x}_n + \mathbf{v}_n \end{array}\right.$$

State space modeling examples

- One-dimensional motion
  Track-While-Scan (TWS) Radar

$$\begin{cases} X_t = \frac{\partial P_t}{\partial t} \\ A_t = \frac{\partial V_t}{\partial t} \end{cases} \longrightarrow \begin{cases} X_{n+1} \simeq X_n + T\, V_n \\ V_{n+1} \simeq V_n + T\, A_n \end{cases}$$

$$\begin{cases} \mathbf{x} = \begin{bmatrix} X \\ V \end{bmatrix} \\ u = A \\ z = X + v \end{cases} \longrightarrow \begin{cases} \mathbf{x}_n = \begin{bmatrix} X_n \\ V_n \end{bmatrix} \\ u_n = A_n \\ z_n = X_n + v_n \end{cases}$$

$$\begin{cases} \mathbf{x}_{n+1} & = \mathbf{F}\mathbf{x}_n + \mathbf{G} u_n \\ Z_n & = \mathbf{H}\mathbf{x}_n + v_n \end{cases} \text{ with } \quad \mathbf{F} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} 0 \\ T \end{bmatrix}, \mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$\begin{cases} \mathbf{f}_n(\mathbf{x}_n, \mathbf{u}_n) = \mathbf{F}\, \mathbf{x}_n + \mathbf{G}\, \mathbf{u}_n \\ \mathbf{h}_n(\mathbf{x}_n, \mathbf{v}_n) = \mathbf{H}\, \mathbf{x}_n + \mathbf{v}_n \end{cases}$$

State space modeling examples

- 1D motion of heavy targets
  Track-While-Scan (TWS) Radar with dependent acceleration sequences
- heavy target:

$$A_{n+1} = \rho A_n + W_n, \quad n = 0, 1, \ldots$$

- ρ near to 0: low inertia target
- ρ near to 1: high inertia target.

$$\begin{bmatrix} X_{n+1} \\ V_{n+1} \\ A_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & T \\ 0 & 0 & \rho \end{bmatrix} \begin{bmatrix} X_n \\ V_n \\ A_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} W_n$$

$$Z_n = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} X_n \\ V_n \\ A_n \end{bmatrix} + e_n$$

Kalman Filtering: Recursive Linear Filtering

$$\underset{\text{non observable}}{x(n)} \quad \longrightarrow \boxed{\text{Linear System}} \longrightarrow \quad \underset{\text{observable}}{z(n)}$$

- Objective: Estimate $x(n)$ from the observed values of $\{z(n), n = 1, \ldots, k\}$.
- $\widehat{x}(n)$ is then a function of the data $\{z(n), n = 1, \ldots, k\}$

$$\widehat{x}(n \mid z(1), z(2), \ldots, z(k)) \overset{\triangle}{=} \widehat{x}(n \mid k)$$

- $\widehat{x}(n + k|n)$ is called the k-th *order prediction* of $z(n)$ and the estimation procedure is called *prediction*.
- $\widehat{x}(n|n)$ is the *filtered value* of $z(n)$ and the estimation procedure is called *filtering*.
- $\widehat{x}(n|n + l)$ is the *smoothed value* of $z(n)$ and the estimation procedure is called *smoothing*.

State space modeling: General case

$$\begin{cases} \mathbf{x}_{k+1} & = \mathbf{F}_k \, \mathbf{x}_k + \mathbf{G}_k \, \mathbf{u}_k \quad \text{state equation,} \\ \mathbf{z}_k & = \mathbf{H}_k \; \mathbf{x}_k + \mathbf{v}_k \quad \text{observation equation} \end{cases}$$

- $\mathbf{x}_k$ N-dimensional *state vector*
- $\mathbf{z}_k$ P-dimensional *observations vector*
- $\mathbf{v}_k$ P-dimensional *observations error vector*
- $\mathbf{u}_k$ M-dimensional *state representation error*
- $\mathbf{F}_k$, $\mathbf{G}_k$ and $\mathbf{H}_k$ with respective dimensions of (N, N), (N, M) and (P, N) are the *state transition*, the *state input* and the *observation matrices* and are assumed to be known.
- The noise sequences $\{\mathbf{u}_k\}$ and $\{\mathbf{v}_k\}$ and the *initial state* $\mathbf{x}_0$ are assumed to be centered, white and jointly Gaussian.

$$\mathrm{E}\left\{ \begin{bmatrix} \mathbf{v}_k \\ \mathbf{x}_0 \\ \mathbf{u}_k \end{bmatrix} \left[\mathbf{v}_l^t, \mathbf{x}_0^t, \mathbf{u}_l^t\right] \right\} = \begin{bmatrix} \mathbf{R}_k & 0 & 0 \\ 0 & \mathbf{P}_0 & 0 \\ 0 & 0 & \mathbf{Q}_k \end{bmatrix} \delta_{kl}$$

Kalman Filtering: Prediction, Filtering and Smoothing

Objective: Find the best estimate $\widehat{\mathbf{x}}_{k|l}$ of $\mathbf{x}_k$ from the observations $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_l$.

$$\widehat{\mathbf{x}}_k(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_l) \overset{\triangle}{=} \widehat{\mathbf{x}}(k \mid l)$$

- If $k > l$ prediction. For example $l = n, k = n + 1$:

$$\widehat{\mathbf{x}}_{n+1}(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n) \overset{\triangle}{=} \widehat{\mathbf{x}}(n + 1 \mid n)$$

- If $k = l$ filtering. For example $l = n, k = n$:

$$\widehat{\mathbf{x}}_n(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n) \overset{\triangle}{=} \widehat{\mathbf{x}}(n \mid n)$$

- If $k < l$ smoothing. For example $l = n + 1, k = n$:

$$\widehat{\mathbf{x}}_n(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{n+1}) \overset{\triangle}{=} \widehat{\mathbf{x}}(n \mid n + 1)$$

Kalman Filtering: 3 Interpretations

Three different approaches can be used to obtain the Kalman filtering equations:

- Least Mean Square (LMS) estimation:

  $$\widehat{\mathbf{x}}_{k|l} \overset{\triangle}{=} \mathrm{LMS}(\mathbf{x}_k \mid \mathbf{z}_1, \ldots, \mathbf{z}_l)$$

  which minimizes

  $$\mathrm{E}\left\{[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|l}]^t \, \mathbf{W}_k \, [\mathbf{x}_k - \widehat{\mathbf{x}}_{k|l}]\right\}$$

- Maximum A posteriori (MAP) estimate:

  $$\widehat{\mathbf{x}}_{k|l} = \arg\max_{\mathbf{x}} \{p(\mathbf{x}_k \mid \mathbf{z}_1, \ldots, \mathbf{z}_l)\}$$

- Bayesian MSE estimate:

  $$\widehat{\mathbf{x}}_{k|l} = \mathrm{E}\{\mathbf{x}_k \mid \mathbf{z}_1, \ldots, \mathbf{z}_k\}$$

Kalman Filtering: Bayesian approach

The main procedure is to apply the Bayes rule recursively to find the expression of the posterior law $p(\mathbf{x}_k \mid \mathbf{z}_1, \ldots, \mathbf{z}_l)$.

1. All variables are assumed Gaussian ;
2. All the conditional laws such as $p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})$ and $p(\mathbf{z}_{k+1}|\mathbf{z}_{1:k})$ are Gaussian. So, the posterior law
$$p(\mathbf{x}_{k+1}|\mathbf{z}_{1:k+1}) = p(\mathbf{x}_{k+1}|\mathbf{z}_{1:k}) \frac{p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})}{p(\mathbf{z}_{k+1}|\mathbf{z}_{1:k})}$$
is also Gaussian.

To obtain the equations in the general case we note

- $\widehat{\mathbf{x}}_{k|k}$ the estimate of the state vector at time k from the observations up to time k ;
- $\widehat{\mathbf{x}}_{k+1|k}$ the estimate of the state vector at time $k+1$ from the observations up to the instant k ;
- $\mathbf{e}_{k+1} = \mathbf{z}_{k+1} - H_{k+1}\,\widehat{\mathbf{x}}_{k+1|k}$ the *innovation process* of the observations at the instant $k+1$

Kalman Filtering: Bayesian approach

- The covariance matrix of the innovation by

  $$\mathbf{R}^e_{k+1} = \mathrm{E}\left\{\mathbf{e}_{k+1|k}\,\mathbf{e}^t_{k+1|k}\right\}$$

  which is diagonal;

- The covariance matrix of the prediction error by

  $$\mathbf{P}_{k+1|k} = \mathrm{E}\left\{\left[\mathbf{x}_{k+1} - \widehat{\mathbf{x}}_{k+1|k}\right]\left[\mathbf{x}_{k+1} - \widehat{\mathbf{x}}_{k+1|k}\right]^t\right\},$$

- The posterior covariance matrix of the estimation error by

  $$\mathbf{P}_{k+1|k+1} = \mathrm{E}\left\{\left[\mathbf{x}_{k+1} - \widehat{\mathbf{x}}_{k+1|k+1}\right]\left[\mathbf{x}_{k+1} - \widehat{\mathbf{x}}_{k+1|k+1}\right]^t\right\}$$

  which is also called the covariance matrix of the filtering error.

Kalman Filtering: Bayesian approach

$$
\begin{aligned}
\mathrm{E}\left\{\mathbf{x}_k | \mathbf{z}_{1:k}\right\} &\stackrel{\triangle}{=} \widehat{\mathbf{x}}_{k|k} \\
\operatorname{cov}[\mathbf{x}_k | \mathbf{z}_{1:k}] &\stackrel{\triangle}{=} \mathbf{P}_{k|k} \\
\mathrm{E}\left\{\mathbf{x}_{k+1} | \mathbf{z}_{1:k}\right\} &\stackrel{\triangle}{=} \widehat{\mathbf{x}}_{k+1|k} \\
\operatorname{cov}[\mathbf{x}_{k+1} | \mathbf{z}_{1:k}] &\stackrel{\triangle}{=} \mathbf{P}_{k+1|k}
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{E}\left\{\mathbf{z}_{k+1} | \mathbf{x}_{k+1}\right\} &= \mathbf{H}_{k+1}\, \widehat{\mathbf{x}}_{k+1} \\
\operatorname{cov}[\mathbf{z}_{k+1} | \mathbf{x}_{k+1}] &= \mathbf{R}_{k+1}
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{E}\left\{\mathbf{x}_{k+1} | \mathbf{z}_{1:k}\right\} &= \mathbf{F}_k\, \widehat{\mathbf{x}}_{k|k} \\
\operatorname{cov}[\mathbf{x}_{k+1} | \mathbf{z}_{1:k}] &= \mathbf{F}_k\, \mathbf{P}_k \mathbf{F}_k^t + \mathbf{G}_k\, \mathbf{Q}_k\, \mathbf{G}_k^t
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{E}\left\{\mathbf{z}_{k+1} | \mathbf{z}_{1:k}\right\} &= \mathbf{H}_{k+1}\, \mathbf{F}_k\, \widehat{\mathbf{x}}_{k|k} \\
\operatorname{cov}[\mathbf{z}_{k+1} | \mathbf{z}_{1:k}] &= \mathbf{H}_{k+1}\, \mathbf{P}_{k+1|k} \mathbf{H}_{k+1}^t + \mathbf{R}_{k+1}
\end{aligned}
$$

Kalman Filtering: Bayesian approach

Replacing the expressions of $p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})$ and $p(\mathbf{z}_{k+1}|\mathbf{z}_{1:k})$:

$$p(\mathbf{x}_{k+1}|\mathbf{z}_{1:k+1}) = A \exp\left\{-\frac{1}{2}[\mathbf{x}_{k+1} - \widehat{\mathbf{x}}_{k+1|k}]^t \mathbf{P}_{k+1|k+1}^{-1} [\mathbf{x}_{k+1} - \widehat{\mathbf{x}}_{k+1|k}]\right\}$$

with

$$A = \frac{1}{(2\pi)^{n/2}} \left\{ \left| \mathbf{H}_{k+1}\, \mathbf{P}_{k+1|k}\, \mathbf{H}_{k+1}^t + \mathbf{R}_k \right|^{1/2} \left| \mathbf{R}_k \right|^{-1/2} \left| \mathbf{P}_{k+1|k} \right| \right\}^{-1/2}$$

$$\begin{aligned}
\widehat{\mathbf{x}}_{k+1|k} &= \mathbf{F}_k\, \widehat{\mathbf{x}}_{k|k} + \mathbf{P}_{k+1|k}\, \mathbf{H}_{k+1}^t \\
&\quad \left[\mathbf{R}_{k+1} + \mathbf{H}_{k+1}\, \mathbf{P}_{k+1|k}\, \mathbf{H}_{k+1}^t\right]^{-1} \left[\mathbf{z}_{k+1} - \mathbf{H}_{k+1}\, \mathbf{F}_k\, \widehat{\mathbf{x}}_{k|k}\right] \\
\mathbf{P}_{k+1|k} &= \mathbf{F}_k\, \mathbf{P}_{k|k}\, \mathbf{F}_k^t + \mathbf{G}_k\, \mathbf{Q}_k\, \mathbf{G}_k^t \\
\mathbf{P}_{k+1|k+1}^{-1} &= \mathbf{P}_{k+1|k}^{-1} + \mathbf{H}_{k+1}^t \mathbf{R}_{k+1}^{-1}\, \mathbf{H}_{k+1}
\end{aligned}$$

Kalman Filtering: Different forms of equation

- **Prediction-Correction form :**
  - **Prediction (Time update) :**

$$\begin{aligned} \widehat{\mathbf{x}}_{k+1|k} &= \mathbf{F}_k \, \widehat{\mathbf{x}}_{k|k} \\ \mathbf{P}_{k+1|k} &= \mathbf{F}_k \, \mathbf{P}_{k|k} \, \mathbf{F}_k^t + \mathbf{G}_k \, \mathbf{Q}_k \, \mathbf{G}_k^t \end{aligned}$$

  - **Correction (measurement update) :**

$$\begin{aligned} \widehat{\mathbf{x}}_{k+1|k+1} &= \widehat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}^g [\mathbf{z}_{k+1} - \mathbf{H}_{k+1} \, \widehat{\mathbf{x}}_{k+1|k}] \\ \mathbf{K}_{k+1}^g &= \mathbf{P}_{k+1|k} \, \mathbf{H}_{k+1}^t (\mathbf{R}_{k+1}^e)^{-1} \\ \mathbf{R}_{k+1}^e &= \mathbf{R}_{k+1} + \mathbf{H}_{k+1} \, \mathbf{P}_{k+1|k} \, \mathbf{H}_{k+1}^t \\ \mathbf{P}_{k+1|k+1} &= [\mathbf{I} - \mathbf{K}_{k+1}^f \mathbf{H}_{k+1}] \mathbf{P}_{k+1|k} \end{aligned}$$

Kalman Filtering: Different forms of equation

- **Compact form for prediction :**

$$
\begin{aligned}
\widehat{\mathbf{x}}_{k+1|k} &= \mathbf{F}_k\,\widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{R}^e_k)^{-1}[\mathbf{z}_k - \mathbf{H}_k\,\widehat{\mathbf{x}}_{k|k-1}] \\
\mathbf{R}^e_k &= \mathbf{R}_k + \mathbf{H}_k\,\mathbf{P}_{k|k-1}\,\mathbf{H}^t_k \\
\mathbf{K}_k &= \mathbf{F}_k\,\mathbf{P}_{k|k-1}\,\mathbf{H}^t_k \\
\mathbf{P}_{k+1|k} &= \mathbf{F}_k\,\mathbf{P}_{k|k-1}\,\mathbf{F}^t_k + \mathbf{G}_k\,\mathbf{Q}_k\,\mathbf{G}^t_k - \mathbf{K}_k(\mathbf{R}^e_k)^{-1}\,\mathbf{K}^t_k
\end{aligned}
$$

- In all cases the initialization is :

$$
\widehat{\mathbf{x}}_{0|-1} = 0 \quad \mathbf{P}_{0|-1} = \mathbf{P}_0
$$

Kalman Filtering: Different forms of equation

- **Compact form for filtering :**

$$\begin{array}{rcl} \widehat{\mathbf{x}}_{k|k} & = & \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k^g[\mathbf{z}_k - \mathbf{H}_k\,\widehat{\mathbf{x}}_{k|k-1}] \\ \mathbf{K}_k^g & = & \mathbf{P}_{k|k-1}\,\mathbf{H}_k^t[\mathbf{R}_k + \mathbf{H}_k\,\mathbf{P}_{k|k-1}\,\mathbf{H}_k^t]^{-1} \\ \mathbf{P}_{k|k} & = & \mathbf{P}_{k|k-1} - \mathbf{K}_k^g\mathbf{H}_k\mathbf{P}_{k|k-1} \\ \mathbf{P}_{k+1|k} & = & \mathbf{F}_k\mathbf{P}_{k|k}\,\mathbf{F}_k^t + \mathbf{G}_k\,\mathbf{Q}_k\,\mathbf{G}_k^t \end{array}$$

- **Very compact form for prediction :**

$$\begin{array}{rcl} \widehat{\mathbf{x}}_{k+1|k} & = & \mathbf{F}_k\widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k^g[\mathbf{z}_k - \mathbf{H}_k\,\widehat{\mathbf{x}}_{k|k-1}] \\ \mathbf{K}_k^g & = & \mathbf{P}_{k|k-1}\,\mathbf{H}_k^t[\mathbf{R}_k + \mathbf{H}_k\,\mathbf{P}_{k|k-1}\,\mathbf{H}_k^t]^{-1} \\ \mathbf{P}_{k+1|k} & = & \mathbf{F}_k\mathbf{P}_{k|k-1}\,\mathbf{F}_k^t - \mathbf{F}_k\mathbf{K}_k^g\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{F}_k^t + \mathbf{G}_k\,\mathbf{Q}_k\,\mathbf{G}_k^t \end{array}$$

where $\mathbf{K}_k$ is called the *Kalman filter gain* and $\mathbf{K}_k^g = \mathbf{K}_k(\mathbf{R}_k^e)^{-1}$ the generalized *Kalman filter gain*.

Kalman Filtering equations: 1D case

An AR1 signal is observed through a perturbed chanel:

$$\left\{\begin{array}{lcl} x_{n+1} & = & f\, x_n + u_n \\ z_n & = & h\, x_n + v_n \end{array}\right. \qquad \left\{\begin{array}{lcl} u_n & \sim & \mathcal{N}(0, q) \\ v_n & \sim & \mathcal{N}(0, r) \\ x_0 & \sim & \mathcal{N}(m_0, p_0) \end{array}\right.$$

$$\left\{\begin{array}{lcl} \widehat{x}_{n+1|n} & = & f\, \widehat{x}_{n|n} \\ \widehat{x}_{n|n} & = & \widehat{x}_{n|n-1} + k_n\, (z_n - h\, \widehat{x}_{n|n-1}) \\ k_n & = & (h^2 p_{n|n-1} + r)^{-1} p_{n|n-1} h = \frac{p_{n|n-1} h}{h^2 p_{n|n-1} + r} = \frac{1}{h}\, \frac{p_{n|n-1}}{p_{n|n-1} + r/h^2} \end{array}\right.$$

$$\left\{\begin{array}{lcl} p_{n+1|n} & = & f^2\, p_{n|n} + q \\ p_{n|n} & = & \frac{1}{h}\, \frac{p_{n|n-1}}{\frac{r}{h^2}\, p_{n|n-1+1}} \end{array}\right.$$

We can eliminate the coupling between these equations and obtain

$$p_{n+1|n} = \frac{f^2\, p_{n|n-1}}{\frac{h^2}{r}\, p_{n|n-1} + 1} + q$$

As n increases, $p_{n+1|n}$ and so the gain k_n approaches a constant.

Kalman Filtering equations: 1D case

If $p_{n+1|n}$ does approach a constant, say p_∞, then p_∞ must satisfy

$$p_\infty = \frac{f^2\, p_\infty}{\frac{h^2}{r}\, p_\infty + 1} \; + \; q$$

This equation is quadratic and has a unique positive solution

$$p_\infty = \frac{1}{2}\left\{\left[\frac{r}{h^2}(1-f^2) - q\right]^2 + \frac{4rq}{h^2}\right\}^{1/2} - \frac{r}{2h^2}(1-f^2) + q$$

$$\begin{aligned} \left|p_{n+1|n} - p_\infty\right| &= f^2 \left| \frac{p_{n|n-1}}{\frac{h^2}{r}\, p_{n|n-1} + 1} - \frac{p_\infty}{\frac{h^2}{r}\, p_\infty + 1} \right| \\ &\leq f^2 \left|p_{n|n-1} - p_\infty\right| \end{aligned}$$

$$\left|p_{n+1|n} - p_\infty\right| \leq f^{2(n+1)} \left|p_0 - p_\infty\right|$$

$$\leq \quad f^2 \left|p_{n|n-1} - p_\infty\right|$$

- If $|f| < 1$ then $p_{n+1|n}$ converges to p_∞.
- $|f| < 1$ is a sufficient condition for Kalman-Bucy filter to approach a steady state.

Kalman Filtering: Applications

Track-While-Scan (TWS) Radar

$$\begin{aligned}\begin{bmatrix} X_{n+1} \\ V_{n+1} \end{bmatrix} &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_n \\ V_n \end{bmatrix} + \begin{bmatrix} 0 \\ T \end{bmatrix} A_n \\ Z_n &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} X_n \\ V_n \end{bmatrix} + e_n\end{aligned}$$

- For a more general case in 3D we have a state vector with 6 components (3 positions and 3 velocities).
- But, if we assume that the measurement noise in 3 dimensions are independent of one another and independent to the components of the acceleration, the problem can be treated as 3 independent one-dimensional moving target.

Kalman Filtering: Applications

Track-While-Scan (TWS) Radar

$$\begin{aligned}
\begin{bmatrix}\widehat{X}_{n+1|n}\\ \widehat{V}_{n+1|n}\end{bmatrix} &= \begin{bmatrix}\widehat{X}_{n|n}+T\widehat{V}_{n|n}\\ \widehat{V}_{n|n}\end{bmatrix}\\
\begin{bmatrix}\widehat{X}_{n|n}\\ \widehat{V}_{n|n}\end{bmatrix} &= \begin{bmatrix}\widehat{X}_{n|n-1}\\ \widehat{V}_{n|n-1}\end{bmatrix}+\begin{bmatrix}K_{n,1}\\ K_{n,2}\end{bmatrix}\begin{bmatrix}Z_n-\widehat{X}_{n|n-1}\end{bmatrix}\\
\begin{bmatrix}K_{n,1}\\ K_{n,2}\end{bmatrix} &= \begin{bmatrix}\frac{P(1,1)}{P(1,1)+r}\\ \frac{P(2,1)}{P(1,1)+r}\end{bmatrix}
\end{aligned}$$

where $P(k,l)$ is the $(k-l)$th component of the matrix $P_{n|n-1}$. To reduce the computation, the time varying elements of the Kalman gain vector can be replaced with some constants (the steady states values) to obtain

$$\begin{bmatrix}\widehat{X}_{n|n}\\ \widehat{V}_{n|n}\end{bmatrix} = \begin{bmatrix}\widehat{X}_{n|n-1}\\ \widehat{V}_{n|n-1}\end{bmatrix}+\begin{bmatrix}\alpha\\ \beta/T\end{bmatrix}\begin{bmatrix}Z_n-\widehat{X}_{n|n-1}\end{bmatrix}$$

with constatnt values for α and β.

Kalman Filtering: Applications

Track-While-Scan (TWS) Radar with dependent acceleration sequences

- heavy target:

$$A_{n+1} = \rho A_n + W_n, \quad n = 0, 1, \ldots$$

- ρ near to 0: low inertia target and ρ near to 1: high inertia target.

$$\begin{bmatrix} X_{n+1} \\ V_{n+1} \\ A_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & T \\ 0 & 0 & \rho \end{bmatrix} \begin{bmatrix} X_n \\ V_n \\ A_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} W_n$$

$$Z_n = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} X_n \\ V_n \\ A_n \end{bmatrix} + e_n$$

Kalman Filtering: Applications

- **Track-While-Scan (TWS) Radar**

$$\begin{bmatrix} \widehat{X}_{n+1|n} \\ \widehat{V}_{n+1|n} \\ \widehat{A}_{n+1|n} \end{bmatrix} = \begin{bmatrix} \widehat{X}_{n|n} + T\widehat{V}_{n|n} \\ \widehat{V}_{n|n} + T\widehat{A}_{n|n} \\ \rho\widehat{A}_{n|n} \end{bmatrix}$$

$$\begin{bmatrix} \widehat{X}_{n|n} \\ \widehat{V}_{n|n} \\ \widehat{A}_{n|n} \end{bmatrix} = \begin{bmatrix} \widehat{X}_{n|n-1} \\ \widehat{V}_{n|n-1} \\ \widehat{A}_{n|n-1} \end{bmatrix} + \begin{bmatrix} K_{n,1} \\ K_{n,2} \\ K_{n,3} \end{bmatrix} \begin{bmatrix} Z_n - \widehat{X}_{n|n-1} \end{bmatrix}$$

$$\begin{bmatrix} K_{n,1} \\ K_{n,2} \\ K_{n,3} \end{bmatrix} = \begin{bmatrix} \frac{P(1,1)}{P(1,1)+r} \\ \frac{P(2,1)}{P(1,1)+r} \; \frac{P(3,1)}{P(1,1)+r} \end{bmatrix}$$

where $P(k,l)$ is the $(k-l)$th component of the matrix $P_{n|n-1}$.

- To reduce the computation:

$$\begin{bmatrix} \widehat{X}_{n|n} \\ \widehat{V}_{n|n} \\ \widehat{A}_{n|n} \end{bmatrix} \begin{bmatrix} \widehat{X}_{n|n-1} \\ \widehat{V}_{n|n-1} \\ \widehat{A}_{n|n-1} \end{bmatrix} + \begin{bmatrix} \alpha \\ \beta/T \\ \gamma/T^2 \end{bmatrix} \begin{bmatrix} Z_n - \widehat{X}_{n|n-1} \end{bmatrix}$$

Kalman Filtering: Fast Kalman Algorithms

$$\begin{aligned}
\widehat{\mathbf{x}}_{k+1|k} &= \mathbf{F}\widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{R}^e_k)^{-1}[\mathbf{z}_k - \mathbf{H}\widehat{\mathbf{x}}_{k|k-1}] \\
\mathbf{R}^e_k &= \mathbf{R} + \mathbf{H}\mathbf{P}_{k|k-1}\,\mathbf{H}^t \\
\mathbf{K}_k &= \mathbf{F}\mathbf{P}_{k|k-1}\,\mathbf{H}^t \\
\mathbf{P}_{k+1|k} &= \mathbf{F}\mathbf{P}_{k|k-1}\,\mathbf{F}^t + \mathbf{G}\mathbf{Q}\mathbf{G}^t - \mathbf{K}_k(\mathbf{R}^e_k)^{-1}\,\mathbf{K}^t_k
\end{aligned}$$

$$\begin{aligned}
\delta\mathbf{P}_k &= \mathbf{P}_{k|k-1} - \mathbf{P}_{k-1|k-2} \\
\delta\mathbf{K}^g_k &= \mathbf{K}^g_k - \mathbf{K}^g_{k-1} \\
\delta\mathbf{R}^e_k &= \mathbf{R}^e_k - \mathbf{R}^e_{k-1}
\end{aligned}$$

$$\begin{aligned}
\delta\mathbf{P}_{k+1} &= [\mathbf{F} - \mathbf{K}^g_{k-1}\mathbf{H}][\delta\mathbf{P}_k - \delta\mathbf{P}_k\,\mathbf{H}^t(\mathbf{R}^e_k)^{-1}\,\mathbf{H}\mathbf{P}_k][\mathbf{F} - \mathbf{K}^g_{k-1}\mathbf{H}]^t \\
&= [\mathbf{F} - \mathbf{K}^g_k\mathbf{H}][\delta\mathbf{P}_k + \delta\mathbf{P}_k\,\mathbf{H}^t(\mathbf{R}^e_{k-1})^{-1}\,\mathbf{H}\mathbf{P}_k][\mathbf{F} - \mathbf{K}^g_k\mathbf{H}]^t
\end{aligned}$$

Kalman Filtering: Fast Kalman Algorithms

- If $\delta\mathbf{P}_k$ can be factorized as:

$$\delta\mathbf{P}_1 = \mathbf{z}_0\mathbf{M}_0\mathbf{z}_0^t, \longrightarrow \delta\mathbf{P}_{k+1} = \mathbf{z}_k\,\mathbf{M}_k\,\mathbf{z}_k^t$$

$$\begin{aligned}
\delta\mathbf{P}_{k+1} &= \mathbf{z}_k\,\mathbf{M}_k\,\mathbf{z}_k^t \\
\mathbf{z}_k &= [\mathbf{F} - \mathbf{K}_k^g\mathbf{H}]\mathbf{z}_{k-1} \\
\mathbf{M}_k &= \mathbf{M}_{k-1} + \mathbf{M}_{k-1}\,\mathbf{z}_{k-1}^t\mathbf{H}^t(\mathbf{R}_{k-1}^e)^{-1}\,\mathbf{H}\mathbf{z}_{k-1}\,\mathbf{M}_{k-1} \\
\mathbf{R}_{k-1}^e &= \mathbf{R}_k^e + \mathbf{H}\mathbf{z}_k\,\mathbf{M}_k\,\mathbf{z}_k^t\mathbf{H}^t \\
\mathbf{K}_{k+1} &= \mathbf{K}_{k+1}^g\,\mathbf{R}_{k+1}^e = \mathbf{K}_k + \mathbf{F}\mathbf{z}_k\,\mathbf{M}_k\,\mathbf{z}_k^t\mathbf{H}^t \\
\mathbf{P}_{k|k-1} &= \mathbf{P}_0 + \sum_{j=0}^{k-1}\mathbf{z}_j\mathbf{M}_j\mathbf{z}_j^t
\end{aligned}$$

which are called *Chandrasekhar* equations.

Kalman Filtering: Fast Kalman Algorithms

- Note that if $\alpha = \text{rang}\,\{\delta \mathbf{P}_1\}$ where

  $$\delta \mathbf{P}_1 = \mathbf{F}\mathbf{P}_0\mathbf{F}^t + \mathbf{G}\mathbf{Q}\mathbf{G}^t - \mathbf{K}_0(\mathbf{R}_0^e)^{-1}\mathbf{K}_0^t - \mathbf{P}_0$$

  then $\mathbf{z}_k$ has dimensions (N, α), $\mathbf{M}_k$ has dimensions (α, α).
- So, in place of updating $\mathbf{P}_k$ with dimensions (N, N) we only have to update the matrixes $\mathbf{z}_k$ and $\mathbf{M}_k$ with dimensions (N, α) and (α, α).
- Note also that $\mathbf{M}_0$ is the signature matrix of $\delta \mathbf{P}_1$ and the value of α depends on the choice of the initial covariance matrix $\mathbf{P}_0$.
- It is not unusual to have $\alpha = 1$ which greately reduces the computation cost.

Kalman Filter equations for signal deconvolution

$$z(k) = \sum_{i=0}^{p-1} h(i)x(k-i) + \epsilon(k)$$

$$\begin{bmatrix} z(1) \\ \vdots \\ z(k) \\ \vdots \\ z(M) \end{bmatrix} = \begin{bmatrix} h_{(p-1)} & \cdots & h_{(0)} & \cdots & \cdots \\ \vdots & & & & \vdots \\ 0 & h_{(p-1)} & \cdots & h_{(0)} & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & h_{(p-1)} & \cdots & h_{(0)} \end{bmatrix} \begin{bmatrix} x(-p) \\ \vdots \\ x(0) \\ \vdots \\ x(M) \end{bmatrix} + \begin{bmatrix} \epsilon(1) \\ \vdots \\ \epsilon(k) \\ \vdots \\ \epsilon(M) \end{bmatrix}$$

- **Constant state vector model**

$$\left\{ \begin{array}{lll} \mathbf{x}_{k+1} & = & \mathbf{x}_k = \mathbf{x} = [x_{-p}, \ldots, x_{-1}, x_0, x_1, \ldots, x_n]^t \\ z_k & = & \mathbf{h}_k^t \cdot \mathbf{x}_k + v_k \end{array} \right.$$

$$\mathbf{h}_k = \begin{bmatrix} 0 & 0 & 0 & h_{p-1} & \ldots & h_0 & 0 & 0 \end{bmatrix}^t$$

where coefficient h_0 is in the k-th position.

Kalman Filter equations for signal deconvolution

- **Constant state vector model**

$$\left\{\begin{array}{lcl}\mathbf{u}_k & = & \mathbf{0} \\ \mathbf{F}_k & = & \mathbf{G}_k = \mathbf{I} \\ \mathbf{h}_{k+1}^t & = & \mathbf{D}\mathbf{h}_k^t\end{array}\right. \quad \text{with } \mathbf{D} = \begin{bmatrix} 0 & \dots & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \vdots & \vdots \\ \vdots & & & \vdots & \vdots \\ 0 & & \dots & 1 & 0 \end{bmatrix}$$

$$\mathrm{E}\left\{\mathbf{x}\right\} = \mathbf{x}_0 \qquad \mathrm{E}\left\{[\mathbf{x}-\mathbf{x}_0][\mathbf{x}-\mathbf{x}_0]^t\right\} = \mathbf{P}_0$$

$$\mathrm{E}\left\{v_k\right\} = 0 \qquad \mathrm{E}\left\{v_k v_j\right\} = r\,\delta_{kj}$$

$$\begin{array}{rcl}\widehat{\mathbf{x}}_{k|k} & = & \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (r_k^e)^{-1} [y_k - \mathbf{h}_k^t \cdot \widehat{\mathbf{x}}_{k|k-1}] \\ r_k^e & = & r + \mathbf{h}_k^t \mathbf{P}_{k|k-1}\,\mathbf{h}_k \\ \mathbf{K}_k & = & \mathbf{P}_{k|k-1}\,\mathbf{h}_k^t \\ \mathbf{P}_{k+1|k} & = & \mathbf{P}_{k|k-1} - \mathbf{K}_k (r_k^e)^{-1}\,\mathbf{K}_k^t\end{array}$$

- y_k and r_k^e are scalar.
- $\mathbf{x}$ is a N-dimensional.
- The covariance matrix $\mathbf{P}$ has the dimensions $(N \times N)$.

Kalman Filter equations for signal deconvolution

- **Non constant state space model**

$$\mathbf{h} = [h_0, \ldots, h_{p-1}]^t, \quad \mathbf{x}_k = [x_k, x_{k-1}, \ldots, x_{k-p+1}]^t$$

$$z(k) = \mathbf{h}^t \mathbf{x}_k + \epsilon(k), \quad \text{dimension of } \mathbf{x}_k = p \leq N$$

$$x(n+1) = \sum_{i=1}^{q} a(i)\, x(n-i+1) + u(n+1)$$

$$\mathrm{E}\left\{u_n\right\} = 0, \quad \mathrm{E}\left\{|u_n|^2\right\} = \beta^2, \quad \mathrm{E}\left\{u_m u_n\right\} = 0, \quad m \neq n$$

$$\begin{bmatrix} x(n+1) \\ x(n) \\ \vdots \\ \vdots \\ x(n-q+2) \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & \ldots & \ldots & a_q \\ 1 & 0 & \ldots & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \\ \vdots & \vdots & \vdots & \vdots & \\ \vdots & \vdots & \vdots & 1 & 0 \end{bmatrix} \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ \vdots \\ x(n-q+1) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix} u(n+1)$$

$$\begin{cases} \mathbf{x}_{k+1} & = & \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{u}_{k+1} \\ y_k & = & \mathbf{h}^t \cdot \mathbf{x}_k + \epsilon_k \end{cases}$$

Kalman Filter equations for signal deconvolution

- **Non constant state space model**

$$\begin{cases} \mathbf{x}_{k+1} & = & \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{u}_{k+1} \\ y_k & = & \mathbf{h}^t \cdot \mathbf{x}_k + \epsilon_k \end{cases}$$

$$\mathbf{F} = \begin{bmatrix} a_1 & a_2 & \ldots & \ldots & a_q \\ 1 & 0 & \ldots & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \\ \vdots & \vdots & \vdots & \vdots & \\ \vdots & \vdots & \vdots & 1 & 0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

- Advantages: **F**, **G** and **H** are constant and we can use fast algorithms.
- Drawbacks: Determination of q and a_k, $k = 1, \ldots, q$.

Kalman Filter equations for signal deconvolution

- If the input signal is also assumed to be causal, we obtain :

$$\begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ g(M) \end{bmatrix} = \begin{bmatrix} h(0) & & & & & \\ h(1) & \ddots & & & & \\ \vdots & & & & & \\ h(p) & \cdots & & h(0) & & \\ 0 & \ddots & & & \ddots & \\ \vdots & & & & & \\ 0 & \cdots & 0 & h(p) & \cdots & h(0) \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f(M) \end{bmatrix}$$

and finally if $p = M$ we have :

$$\begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ g(M) \end{bmatrix} = \begin{bmatrix} h(0) & & & \\ h(1) & h(0) & & \\ \vdots & & \ddots & \\ h(M) & \cdots & h(1) & h(0) \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(M) \end{bmatrix}$$

Remark that, in all cases matrix **H** is TOEPLITZ.

Kalman Filter equations for signal deconvolution

- Circulante form when system and input signal are both causal.

$$\begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ g(M) \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} h(0) & 0 & \cdots & & & 0 & & h(p) & \cdots & h(1) \\ h(1) & \ddots & & & & & & & \ddots & \vdots \\ \vdots & & & & & & & & & h(p) \\ h(p) & \cdots & & h(0) & 0 & & & & & 0 \\ 0 & \ddots & & & \ddots & & & & & \vdots \\ \vdots & & & & & & & & & \\ 0 & \cdots & 0 & h(p) & \cdots & & & h(0) & & 0 \\ \vdots & & & & & & & & & \vdots \\ & & & & \ddots & & & & \ddots & 0 \\ 0 & \cdots & & \cdots & 0 & h(p) & \cdots & & & h(0) \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f(M) \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

- **f** and **g** have been completed artificially by some zeros. This operation is called zero-filling
- The main advantage to do so is that the matrix **H** is now a circulant matrix.

Kalman Filter equations for signal deconvolution

$$\left\{\begin{array}{lll} \mathbf{z}_k & = \mathbf{H}_k\,\mathbf{x}_k + \mathbf{v}_k & \text{observation equation} \\ \mathbf{x}_{k+1} & = \mathbf{F}_k\,\mathbf{x}_k + \mathbf{G}_k\,\mathbf{u}_k & \text{state equation} \end{array}\right.$$

$$\begin{array}{rcl} \widehat{\mathbf{x}}_{k+1|k} & = & \mathbf{F}_k\,\widehat{\mathbf{x}}_{k|k} \\ \mathbf{P}_{k+1|k} & = & \mathbf{F}_k\,\mathbf{P}_{k|k}\,\mathbf{F}_k^t + \mathbf{G}_k\,\mathbf{Q}_k\,\mathbf{G}_k^t \\ \widehat{\mathbf{x}}_{k+1|k+1} & = & \widehat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}^f[\mathbf{z}_{k+1} - \mathbf{H}_{k+1}\,\widehat{\mathbf{x}}_{k+1|k}] \\ \mathbf{K}_{k+1}^f & = & \mathbf{P}_{k+1|k}\,\mathbf{H}_{k+1}^t(\mathbf{R}_{k+1}^e)^{-1} \\ \mathbf{R}_{k+1}^e & = & \mathbf{R}_{k+1} + \mathbf{H}_{k+1}\,\mathbf{P}_{k+1|k}\,\mathbf{H}_{k+1}^t \\ \mathbf{P}_{k+1|k+1} & = & [\mathbf{I} - \mathbf{K}_{k+1}^f\mathbf{H}_{k+1}]\mathbf{P}_{k+1|k} \end{array}$$

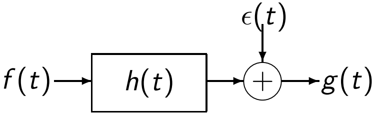
Content

1. Introduction: Signals and Images, Linear transformations (Convolution, Fourier, Laplace, Hilbert, Radon, ..., Discrete convolution, Z transform, DFT, FFT, ...)
2. Modeling: parametric and non-parametric, MA, AR and ARMA models, Linear prediction
3. Deconvolution and Parameter Estimation: Deterministic (LS, Regularization) and Probabilistic methods (Wiener Filtering)
4. Inverse problems and Bayesian estimation
5. Kalman Filtering and smoothing
6. **Case study: Signal deconvolution**
7. Case study: Image restoration
8. Case study: Image reconstruction and Computed Tomography

6. Case study: Signal deconvolution

- Convolution, Identification and Deconvolution
- Forward and Inverse problems: Well-posedness and Ill-posedness
- Naîve methods of Deconvolution
- Classical methods: Wiener filtering
- Bayesian approach to deconvolution
- Simple and Blind Deconvolution
- Deterministic and probabilistic methods
- Joint source and canal estimation

Convolution: Discretization



$$g(t) = \int f(t')\, h(t-t')\, \mathrm{d}t' + \epsilon(t) = \int h(t')\, f(t-t')\, \mathrm{d}t' + \epsilon(t)$$

- The signals $f(t)$, $g(t)$, $h(t)$ are discretized with the same sampling period $\Delta T = 1$,
- The impulse response is finite (FIR) : $h(t) = 0$, for t such that $t < -q\Delta T$ or $\forall t > p\Delta T$.

$$g(m) = \sum_{k=-q}^{p} h(k)\, f(m-k) + \epsilon(m), \quad m = 0, \cdots, M$$

Convolution: Discretized matrix vector forms

$$\begin{bmatrix} g(0)\\ g(1)\\ \vdots\\ \vdots\\ \vdots\\ \vdots\\ \vdots\\ \vdots\\ g(M)\end{bmatrix} = \begin{bmatrix} h(p) & \cdots & h(0) & \cdots & h(-q) & 0 & \cdots & \cdots & \cdots & 0\\ 0 & \ddots & & \ddots & & \ddots & & & & \vdots\\ \vdots & & & & & & & & & \vdots\\ \vdots & & h(p) & \cdots & h(0) & \cdots & h(-q) & & & \vdots\\ \vdots & & & & & & & & & \vdots\\ \vdots & & & \ddots & & & \ddots & & & \vdots\\ \vdots & & & & \ddots & & & & & 0\\ 0 & \cdots & \cdots & & 0 & h(p) & \cdots & h(0) & \cdots & h(-q)\end{bmatrix} \begin{bmatrix} f(-p)\\ \vdots\\ f(0)\\ f(1)\\ \vdots\\ \vdots\\ \vdots\\ f(M)\\ f(M+1)\\ \vdots\\ f(M+q)\end{bmatrix}$$

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \epsilon$$

- $\mathbf{g}$ is a $(M+1)$-dimensional vector,
- $\mathbf{f}$ has dimension $M+p+q+1$,
- $\mathbf{h} = [h(p), \cdots, h(0), \cdots, h(-q)]$ has dimension $(p+q+1)$
- $\mathbf{H}$ has dimensions $(M+1) \times (M+p+q+1)$.

Convolution: Discretized matrix vector form

- If system is causal ($q = 0$) we obtain

$$
\begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ g(M) \end{bmatrix}
=
\begin{bmatrix}
h(p) & \cdots & h(0) & 0 & \cdots & \cdots & 0 \\
0 & & & & & & \vdots \\
\vdots & & & & & & \vdots \\
\vdots & & h(p) & \cdots & h(0) & & \vdots \\
\vdots & & & & & & \vdots \\
\vdots & & & & & & 0 \\
0 & \cdots & \cdots & 0 & h(p) & \cdots & h(0)
\end{bmatrix}
\begin{bmatrix} f(-p) \\ \vdots \\ f(0) \\ f(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f(M) \end{bmatrix}
$$

- $\mathbf{g}$ is a $(M+1)$-dimensional vector,
- $\mathbf{f}$ has dimension $M+p+1$,
- $\mathbf{h} = [h(p), \cdots, h(0)]$ has dimension $(p+1)$
- $\mathbf{H}$ has dimensions $(M+1) \times (M+p+1)$.

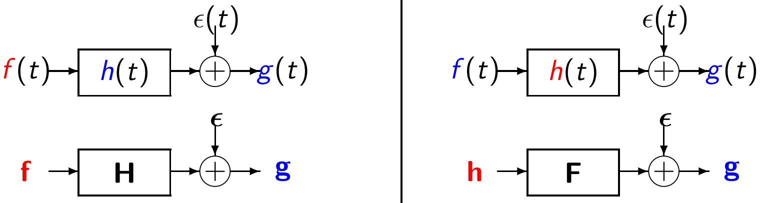
Convolution: Causal systems and causal input

$$\begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ g(M) \end{bmatrix} = \begin{bmatrix} h(0) & & & & & \\ h(1) & \ddots & & & & \\ \vdots & & & & & \\ h(p) & \cdots & & h(0) & & \\ 0 & \ddots & & & \ddots & \\ \vdots & & & & & \\ 0 & \cdots & 0 & h(p) & \cdots & h(0) \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f(M) \end{bmatrix}$$

- $\mathbf{g}$ is a $(M+1)$-dimensional vector,
- $\mathbf{f}$ has dimension $M+1$,
- $\mathbf{h} = [h(p), \cdots, h(0)]$ has dimension $(p+1)$
- $\mathbf{H}$ has dimensions $(M+1) \times (M+1)$.

Convolution, Identification, Simple Deconvolution and Blind deconvolution problems

$$g(t)=\int f(t')\,h(t-t')\,\mathrm{d}t'+\epsilon(t)=\int h(t')\,f(t-t')\,\mathrm{d}t'+\epsilon(t)$$



$$\mathbf{g}=\mathbf{H}\ \mathbf{f}+\boldsymbol{\epsilon} \qquad \mathbf{g}=\mathbf{F}\ \mathbf{h}+\boldsymbol{\epsilon}$$

$$\mathbf{g}=\mathbf{H}\,\mathbf{f}+\boldsymbol{\epsilon}=\mathbf{F}\,\mathbf{h}+\boldsymbol{\epsilon}$$

- Convolution: Given h and f compute g
- Identification: Given f and g estimate h
- Simple Deconvolution: Given h and g estimate f
- Blind Deconvolution: Given g estimate h and f

Convolution: Discretization for Identification

Causal systems and causal input

$$\begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ g(M) \end{bmatrix} = \begin{bmatrix} 0 & & . & 0 & f(0) \\ . & & . & f(0) & f(1) \\ . & & f(0) & f(1) & \vdots \\ . & . & . & . & \vdots \\ f(0) & f(1) & & & f(M-p) \\ f(1) & . & & . & . \\ . & & & . & . \\ . & & . & & . \\ . & . & & & . \\ f(M-p) & . & . & . & f(M) \end{bmatrix} \begin{bmatrix} h(p) \\ h(p-1) \\ \vdots \\ \vdots \\ h(1) \\ h(0) \end{bmatrix}$$

$$\mathbf{g} = \mathbf{F}\,\mathbf{h} + \epsilon$$

- $\mathbf{g}$ is a $(M+1)$-dimensional vector,
- $\mathbf{F}$ has dimension $(M+1) \times (p+1)$,

Identification and Deconvolution

Deconvolution	Identification
$\mathbf{g} = \mathbf{H}\,\mathbf{f} + \epsilon$	$\mathbf{g} = \mathbf{F}\,\mathbf{h} + \epsilon$
$J(\mathbf{f}) = \lVert\mathbf{g} - \mathbf{H}\mathbf{f}\rVert^2 + \lambda_f \lVert\mathbf{D}_f\mathbf{f}\rVert^2$ $\nabla J(\mathbf{f}) = -2\mathbf{H}'(\mathbf{g} - \mathbf{H}\mathbf{f}) + 2\lambda_f \mathbf{D}_f'\mathbf{D}_f\mathbf{f}$ $\widehat{\mathbf{f}} = [\mathbf{H}'\mathbf{H} + \lambda_f \mathbf{D}_f'\mathbf{D}_f]^{-1}\mathbf{H}'\mathbf{g}$ $\widehat{f}(\omega) = \frac{H^*(\omega)}{\lvert H(\omega)\rvert^2 + \lambda_f \lvert D_f(\omega)\rvert^2}\, g(\omega)$	$J(\mathbf{h}) = \lVert\mathbf{g} - \mathbf{F}\mathbf{h}\rVert^2 + \lambda_h \lVert\mathbf{D}_h\mathbf{h}\rVert^2$ $\nabla J(\mathbf{h}) = -2\mathbf{F}'(\mathbf{g} - \mathbf{F}\mathbf{h}) + 2\lambda_h \mathbf{D}_h'\mathbf{D}_h\mathbf{h}$ $\widehat{\mathbf{h}} = [\mathbf{F}'\mathbf{F} + \lambda_h \mathbf{D}_h'\mathbf{D}_h]^{-1}\mathbf{F}'\mathbf{g}$ $\widehat{f}(\omega) = \frac{\lvert F^*(\omega)}{\lvert F(\omega)\rvert^2 + \lambda_h \lvert D_h(\omega)\rvert^2}\, g(\omega)$
$\widehat{f}(\omega) = \frac{H^*(\omega)}{\lvert H(\omega)\rvert^2 + \frac{S_{\epsilon\epsilon}(\omega)}{S_{ff}(\omega)}}\, g(\omega)$	$\widehat{f}(\omega) = \frac{F^*(\omega)}{\lvert F(\omega)\rvert^2 + \frac{S_{\epsilon\epsilon}(\omega)}{S_{hh}(\omega)}}\, g(\omega)$
$p(\mathbf{g}\vert\mathbf{f}) = \mathcal{N}(\mathbf{H}\mathbf{f}, \boldsymbol{\Sigma}_\epsilon)$ $p(\mathbf{f}) = \mathcal{N}(0, \boldsymbol{\Sigma}_f)$	$p(\mathbf{g}\vert\mathbf{h}) = \mathcal{N}(\mathbf{F}\mathbf{h}, \boldsymbol{\Sigma}_\epsilon)$ $p(\mathbf{h}) = \mathcal{N}(0, \boldsymbol{\Sigma}_h)$
$p(\mathbf{f}\vert\mathbf{g}) = \mathcal{N}(\widehat{\mathbf{f}}, \widehat{\boldsymbol{\Sigma}}_f)$ $\widehat{\boldsymbol{\Sigma}}_f = [\mathbf{H}'\mathbf{H} + \lambda_f \mathbf{D}_f'\mathbf{D}_f]^{-1}$ $\widehat{\mathbf{f}} = [\mathbf{H}'\mathbf{H} + \lambda_f \mathbf{D}_f'\mathbf{D}_f]^{-1}\mathbf{H}'\mathbf{g}$	$p(\mathbf{h}\vert\mathbf{g}) = \mathcal{N}(\widehat{\mathbf{h}}, \widehat{\boldsymbol{\Sigma}}_h)$ $\widehat{\boldsymbol{\Sigma}}_h = [\mathbf{F}'\mathbf{F} + \lambda_h \mathbf{D}_h'\mathbf{D}_h]^{-1}$ $\widehat{\mathbf{h}} = [\mathbf{F}'\mathbf{F} + \lambda_h \mathbf{D}_h'\mathbf{D}_h]^{-1}\mathbf{F}'\mathbf{g}$

Blind Deconvolution: Regularization

Deconvolution	Identification
$\mathbf{g} = \mathbf{H}\,\mathbf{f} + \epsilon$	$\mathbf{g} = \mathbf{F}\,\mathbf{h} + \epsilon$
$J(\mathbf{f}) = \Vert\mathbf{g} - \mathbf{H}\mathbf{f}\Vert^2 + \lambda_f \Vert\mathbf{D}_f\mathbf{f}\Vert^2$	$J(\mathbf{h}) = \Vert\mathbf{g} - \mathbf{F}\mathbf{h}\Vert^2 + \lambda_h \Vert\mathbf{D}_h\mathbf{h}\Vert^2$

- Joint Criterion

$$J(\mathbf{f},\mathbf{h}) = \Vert\mathbf{g} - \mathbf{H}\mathbf{f}\Vert^2 + \lambda_f \Vert\mathbf{D}_f\mathbf{f}\Vert^2 + \lambda_h \Vert\mathbf{D}_h\mathbf{h}\Vert^2$$

- iterative algorithm

Deconvolution	Identification
$\nabla_f J(\mathbf{f},\mathbf{h}) = -2\mathbf{H}'(\mathbf{g} - \mathbf{H}\mathbf{f}) + 2\lambda_f \mathbf{D}_f'\mathbf{D}_f\mathbf{f}$	$\nabla_h J(\mathbf{f},\mathbf{h}) = -2\mathbf{F}'(\mathbf{g} - \mathbf{F}\mathbf{h}) + 2\lambda_h \mathbf{D}_h'\mathbf{D}_h\mathbf{h}$
$\widehat{\mathbf{f}} = [\mathbf{H}'\mathbf{H} + \lambda_f \mathbf{D}_f'\mathbf{D}_f]^{-1}\mathbf{H}'\mathbf{g}$	$\widehat{\mathbf{h}} = [\mathbf{F}'\mathbf{F} + \lambda_h \mathbf{D}_h'\mathbf{D}_h]^{-1}\mathbf{F}'\mathbf{g}$
$\widehat{f}(\omega) = \frac{1}{H(\omega)} \frac{\vert H(\omega)\vert^2}{\vert H(\omega)\vert^2 + \lambda_f \vert D_f(\omega)\vert^2}\, g(\omega)$	$\widehat{f}(\omega) = \frac{1}{F(\omega)} \frac{\vert F(\omega)\vert^2}{\vert F(\omega)\vert^2 + \lambda_h \vert D_h(\omega)\vert^2}\, g(\omega)$

Blind Deconvolution: Bayesian approach

Deconvolution	Identification
$\mathbf{g} = \mathbf{H}\,\mathbf{f} + \boldsymbol{\epsilon}$	$\mathbf{g} = \mathbf{F}\,\mathbf{h} + \boldsymbol{\epsilon}$
$p(\mathbf{g} \mid \mathbf{f}) = \mathcal{N}(\mathbf{H}\mathbf{f}, \boldsymbol{\Sigma}_\epsilon)$	$p(\mathbf{g} \mid \mathbf{h}) = \mathcal{N}(\mathbf{F}\mathbf{h}, \boldsymbol{\Sigma}_\epsilon)$
$p(\mathbf{f}) = \mathcal{N}(0, \boldsymbol{\Sigma}_f)$	$p(\mathbf{h}) = \mathcal{N}(0, \boldsymbol{\Sigma}_h)$
$p(\mathbf{f} \mid \mathbf{g}) = \mathcal{N}(\widehat{\mathbf{f}}, \widehat{\boldsymbol{\Sigma}}_f)$	$p(\mathbf{h} \mid \mathbf{g}) = \mathcal{N}(\widehat{\mathbf{h}}, \widehat{\boldsymbol{\Sigma}}_h)$
$\widehat{\boldsymbol{\Sigma}}_f = [\mathbf{H}'\mathbf{H} + \lambda_f \mathbf{D}_f'\mathbf{D}_f]^{-1}$	$\widehat{\boldsymbol{\Sigma}}_h = [\mathbf{F}'\mathbf{F} + \lambda_h \mathbf{D}_h'\mathbf{D}_h]^{-1}$
$\widehat{\mathbf{f}} = [\mathbf{H}'\mathbf{H} + \lambda_f \mathbf{D}_f'\mathbf{D}_f]^{-1}\mathbf{H}'\mathbf{g}$	$\widehat{\mathbf{h}} = [\mathbf{F}'\mathbf{F} + \lambda_h \mathbf{D}_h'\mathbf{D}_h]^{-1}\mathbf{F}'\mathbf{g}$

- Joint posterior law:

$$p(\mathbf{f}, \mathbf{h} \mid \mathbf{g}) \propto p(\mathbf{g} \mid \mathbf{f}, \mathbf{h})\, p(\mathbf{f})\, p(h\mathbf{h})$$

$$p(\mathbf{f}, \mathbf{h} \mid \mathbf{g}) \propto \exp\left\{-J(\mathbf{f}, \mathbf{h})\right\}$$

with

$$J(\mathbf{f}, \mathbf{h}) = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 + \lambda_f \|\mathbf{D}_f \mathbf{f}\|^2 + \lambda_h \|\mathbf{D}_h \mathbf{h}\|^2$$

- iterative algorithm

Blind Deconvolution: Bayesian Joint MAP criterion

- Joint posterior law:

$$p(\mathbf{f},\mathbf{h}|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f},\mathbf{h})\, p(\mathbf{f})\, p(h\mathbf{h})$$

$$p(\mathbf{f},\mathbf{h}|\mathbf{g}) \propto \exp\left\{-J(\mathbf{f},\mathbf{h})\right\}$$

with

$$J(\mathbf{f},\mathbf{h}) = \|\mathbf{g}-\mathbf{H}\mathbf{f}\|^2 + \lambda_f\|\mathbf{D}_f\mathbf{f}\|^2 + \lambda_h\|\mathbf{D}_h\mathbf{h}\|^2$$

- iterative algorithm

Deconvolution	Identification
$p(\mathbf{g}\|\mathbf{f},\mathbf{H}) = \mathcal{N}(\mathbf{H}\mathbf{f},\boldsymbol{\Sigma}_\epsilon)$	$p(\mathbf{g}\|\mathbf{h},\mathbf{F}) = \mathcal{N}(\mathbf{F}\mathbf{h},\boldsymbol{\Sigma}_\epsilon)$
$p(\mathbf{f}) = \mathcal{N}(0,\boldsymbol{\Sigma}_f)$	$p(\mathbf{h}) = \mathcal{N}(0,\boldsymbol{\Sigma}_h)$
$p(\mathbf{f}\|\mathbf{g},\mathbf{H}) = \mathcal{N}(\widehat{\mathbf{f}},\widehat{\boldsymbol{\Sigma}}_f)$	$p(\mathbf{h}\|\mathbf{g},\mathbf{F}) = \mathcal{N}(\widehat{\mathbf{h}},\widehat{\boldsymbol{\Sigma}}_h)$
$\widehat{\boldsymbol{\Sigma}}_f = [\mathbf{H}'\mathbf{H}+\lambda_f\mathbf{D}_f'\mathbf{D}_f]^{-1}$	$\widehat{\boldsymbol{\Sigma}}_h = [\mathbf{F}'\mathbf{F}+\lambda_h\mathbf{D}_h'\mathbf{D}_h]^{-1}$
$\widehat{\mathbf{f}} = [\mathbf{H}'\mathbf{H}+\lambda_f\mathbf{D}_f'\mathbf{D}_f]^{-1}\mathbf{H}'\mathbf{g}$	$\widehat{\mathbf{h}} = [\mathbf{F}'\mathbf{F}+\lambda_h\mathbf{D}_h'\mathbf{D}_h]^{-1}\mathbf{F}'\mathbf{g}$

Blind Deconvolution: Marginalization and EM algorithm

- Joint posterior law:
- Marginalization

$$p(\mathbf{f},\mathbf{h}|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f},\mathbf{h})\, p(\mathbf{f})\, p(h\mathbf{h})$$

$$p(\mathbf{h}|\mathbf{g}) = \int p(\mathbf{f},\mathbf{h}|\mathbf{g})\, \mathrm{d}\mathbf{f}$$

$$\widehat{\mathbf{h}} = \arg\max_{\mathbf{h}} \{p(\mathbf{h}|\mathbf{g})\} \longrightarrow \widehat{\mathbf{f}} = \arg\max_{\mathbf{f}} \left\{p(\mathbf{f}|\mathbf{g},\widehat{\mathbf{h}})\right\}$$

- Expression of $p(\mathbf{h}|\mathbf{g})$ and its maximization are complexes
- Expectation-Maximization Algorithm

$$\ln p(\mathbf{f},\mathbf{h}|\mathbf{g}) \propto J(\mathbf{f},\mathbf{h}) = \|\mathbf{g}-\mathbf{H}\mathbf{f}\|^2 + \lambda_f\|\mathbf{D}_f\mathbf{f}\|^2 + \lambda_h\|\mathbf{D}_h\mathbf{h}\|^2$$

  - Iterative algorithm
  - Expectation: Compute

$$Q(\mathbf{h},\mathbf{h}^{k-1}) = \mathrm{E}_{p(\mathbf{f},\mathbf{h}^{k-1}|\mathbf{g})}\{J(\mathbf{f},\mathbf{h})\} = \langle \ln p(\mathbf{f},\mathbf{h}|\mathbf{g})\rangle_{p(\mathbf{f},\mathbf{h}^{k-1}|\mathbf{g})}$$

  - Maximization:

$$\mathbf{h}^k = \arg\max_{\mathbf{h}} \left\{Q(\mathbf{h},\mathbf{h}^{k-1})\right\}$$

Blind Deconvolution: Variational Bayesian Approximation algorithm

- Joint posterior law:

$$p(\mathbf{f},\mathbf{h}|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f},\mathbf{h})\, p(\mathbf{f})\, p(h\mathbf{h})$$

- Approximation: $p(\mathbf{f},\mathbf{h}|\mathbf{g})$ by $q(\mathbf{f},\mathbf{h}|\mathbf{g}) = q_1(\mathbf{f})\, q_2(\mathbf{h})$
- Criterion of approximation: Kullback-Leiler

$$\mathrm{KL}(q|p) = \int q \ln \frac{q}{p} = \int q_1\, q_2 \ln \frac{q_1\, q_2}{p}$$

$$\begin{aligned} \mathrm{KL}(q_1\, q_2|p) &= \int q_1 \ln q_1 + \int q_2 \ln q_2 - \int q \ln p \\ &= -\mathcal{H}(q_1) - \mathcal{H}(q_2) + \langle -\ln p((\mathbf{f},\mathbf{h}|\mathbf{g})\rangle_q \end{aligned}$$

- When the expression of q_1 and q_2 are obtained, use them.

Variational Bayesian Approximation algorithm

- Kullback-Leibler criterion

$$\begin{aligned} \mathrm{KL}(q_1\, q_2|p) &= \int q_1 \,\ln q_1 + \int q_2 \,\ln q_2 + \int q \ln p \\ &= -\mathcal{H}(q_1) - \mathcal{H}(q_2) + \langle -\ln p((\mathbf{f},\mathbf{h}|\mathbf{g})\rangle_q \end{aligned}$$

- Free energy

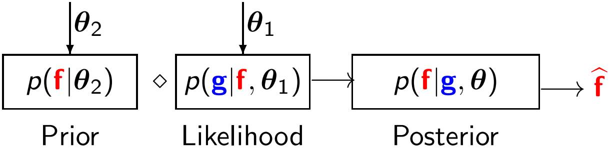
$$\mathcal{F}(q_1\, q_2) = -\left\langle \ln p((\mathbf{f},\mathbf{h}|\mathbf{g})\right\rangle_{q_1 q_2}$$

- Equivalence between optimization of $\mathrm{KL}(q_1\, q_2|p)$ and $\mathcal{F}(q_1\, q_2)$
- Alternate optimization:

$$\begin{aligned} \widehat{q}_1 &= \arg\min_{q_1}\left\{\mathrm{KL}(q_1\, q_2|p)\right\} = \arg\min_{q_1}\left\{\mathcal{F}(q_1\, q_2)\right\} \\ \widehat{q}_2 &= \arg\min_{q_2}\left\{\mathrm{KL}(q_1\, q_2|p)\right\} = \arg\min_{q_2}\left\{\mathcal{F}(q_1\, q_2)\right\} \end{aligned}$$

Summary of Bayesian estimation for Deconvolution

- Simple Bayesian Model and Estimation for Deconvolution

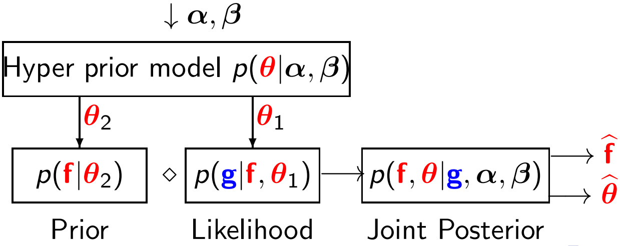


θ_2 θ_1

$p(\mathbf{f}|\theta_2)$ $\diamond$ $p(\mathbf{g}|\mathbf{f},\theta_1)$ $\longrightarrow$ $p(\mathbf{f}|\mathbf{g},\theta)$ $\longrightarrow \widehat{\mathbf{f}}$

Prior Likelihood Posterior

- Full Bayesian Model and Hyperparameter Estimation for Deconvolution



$\downarrow \alpha,\beta$

Hyper prior model $p(\theta|\alpha,\beta)$

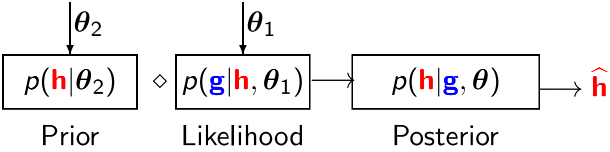
θ_2 θ_1

$p(\mathbf{f}|\theta_2)$ $\diamond$ $p(\mathbf{g}|\mathbf{f},\theta_1)$ $\longrightarrow$ $p(\mathbf{f},\theta|\mathbf{g},\alpha,\beta)$ $\longrightarrow \widehat{\mathbf{f}}$ $\longrightarrow \widehat{\theta}$

Prior Likelihood Joint Posterior

Summary of Bayesian estimation for Identification

- Simple Bayesian Model and Estimation for Identification

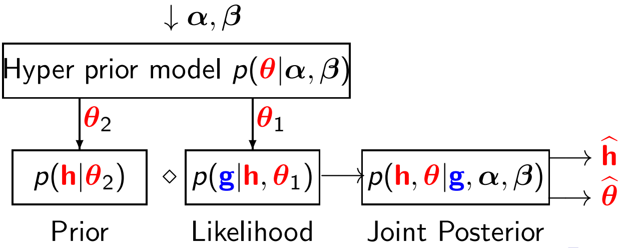


$\boldsymbol{\theta}_2$ $\boldsymbol{\theta}_1$

$p(\mathbf{h}|\boldsymbol{\theta}_2) \diamond p(\mathbf{g}|\mathbf{h},\boldsymbol{\theta}_1) \longrightarrow p(\mathbf{h}|\mathbf{g},\boldsymbol{\theta}) \longrightarrow \widehat{\mathbf{h}}$

Prior Likelihood Posterior

- Full Bayesian Model and Hyperparameter Estimation for Identification



$\downarrow \boldsymbol{\alpha},\boldsymbol{\beta}$

Hyper prior model $p(\boldsymbol{\theta}|\boldsymbol{\alpha},\boldsymbol{\beta})$

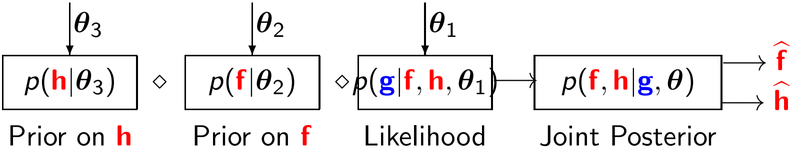
$\boldsymbol{\theta}_2$ $\boldsymbol{\theta}_1$

$p(\mathbf{h}|\boldsymbol{\theta}_2) \diamond p(\mathbf{g}|\mathbf{h},\boldsymbol{\theta}_1) \longrightarrow p(\mathbf{h},\boldsymbol{\theta}|\mathbf{g},\boldsymbol{\alpha},\boldsymbol{\beta})$ $\longrightarrow \widehat{\mathbf{h}}$ $\longrightarrow \widehat{\boldsymbol{\theta}}$

Prior Likelihood Joint Posterior

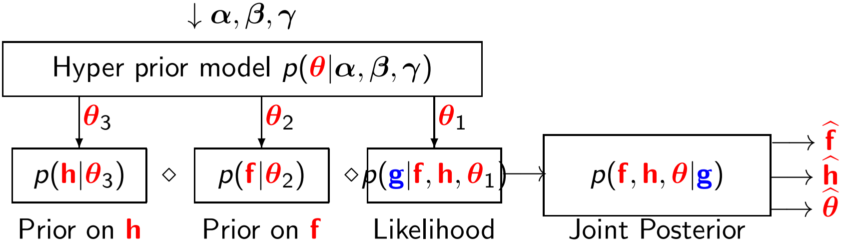
Summary of Bayesian estimation for Blind Deconvolution

Known hyperparameters $\boldsymbol{\theta}$



$$\begin{tikzcd} \boxed{p(\mathbf{h}|\boldsymbol{\theta}_3)} \arrow[r, phantom, "\diamond"] & \boxed{p(\mathbf{f}|\boldsymbol{\theta}_2)} \arrow[r, phantom, "\diamond"] & \boxed{p(\mathbf{g}|\mathbf{f},\mathbf{h},\boldsymbol{\theta}_1)} \arrow[r] & \boxed{p(\mathbf{f},\mathbf{h}|\mathbf{g},\boldsymbol{\theta})} \arrow[r] \arrow[r] & \widehat{\mathbf{f}},\ \widehat{\mathbf{h}} \end{tikzcd}$$

Unknown hyperparameters $\boldsymbol{\theta}$



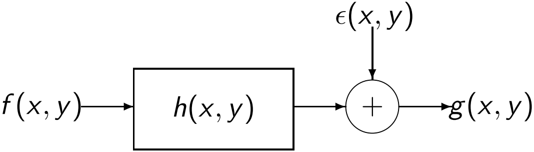
Content

1. Introduction: Signals and Images, Linear transformations (Convolution, Fourier, Laplace, Hilbert, Radon, ..., Discrete convolution, Z transform, DFT, FFT, ...)
2. Modeling: parametric and non-parametric, MA, AR and ARMA models, Linear prediction
3. Deconvolution and Parameter Estimation: Deterministic (LS, Regularization) and Probabilistic methods (Wiener Filtering)
4. Inverse problems and Bayesian estimation
5. Kalman Filtering and smoothing
6. Case study: Signal deconvolution
7. **Case study: Image restoration**
8. Case study: Image reconstruction and Computed Tomography

7. Case study: Image restoration

- Image restoration
- Classical methods: Wiener filtering in 2D
- Bayesian approach to deconvolution
- Simple and Blind Deconvolution
- Practical issues and computational costs

Convolution in 2D: Discretization



$$g(x,y) = \int f(x',y')\, h(x-x')\, \mathrm{d}x\, \mathrm{d}y + \epsilon(x,y)$$

- Images $f(x,y)$, $g(x,y)$, $h(x,y)$ are discretized with the same sampling period $\Delta x = \Delta y = 1$,
- The impulse response is finite (FIR) : $h(x,y) = 0$, for $|x - p\Delta x| < 0$ and $|y - p\Delta y| < 0$.

$$g(m,n) = \sum_{k=-p}^{p} \sum_{l=-p}^{p} h(k,l)\, f(m-k, n-l) + \epsilon(m,n)$$

Convolution: Discretized matrix vector forms

$$g(m,n)=\sum_{k=-p}^{p}\sum_{l=-p}^{p} h(k,l)\, f(m-k,n-l)+\epsilon(m,n)$$

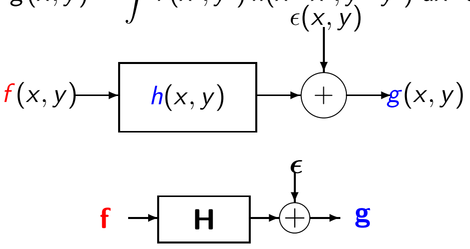
- Matlab notation `g=conv2(h,f);`

$$\mathbf{g}=\mathbf{H}\mathbf{f}+\epsilon$$

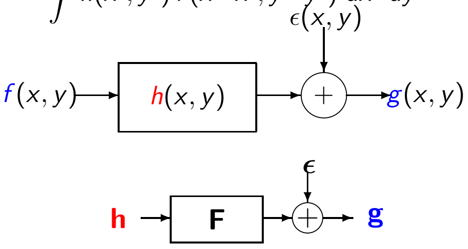
- $\mathbf{g}$ is a vector of dimension $(M \times M)$.
- $\mathbf{f}$ is a vector of dimension $(M+2p+1)\times(M+2p+1)$.
- $\mathbf{H}$ is a Toeplitz-Bloc-Toeplitz matrix of dimensions $(M \times M)\times((M+2p+1)\times(M+2p+1))$.

Convolution, Identification, Simple Deconvolution and Blind deconvolution problems

$$g(x,y) = \int f(x',y')\, h(x-x',y-y')\, \mathrm{d}x'\, \mathrm{d}y' = \int h(x',y')\, f(x-x',y-y')\, \mathrm{d}x'\, \mathrm{d}y'$$



$\mathbf{g} = \mathbf{H}\ \mathbf{f} + \epsilon$



$\mathbf{g} = \mathbf{F}\ \mathbf{h} + \epsilon$

$$\mathbf{g} = \mathbf{H}\,\mathbf{f} + \epsilon = \mathbf{F}\,\mathbf{h} + \epsilon$$

- Convolution: Given h and f compute g
- Identification: Given f and g estimate h
- Simple Deconvolution: Given h and g estimate f
- Blind Deconvolution: Given g estimate h and f

Identification and Deconvolution

Deconvolution	Identification
$\mathbf{g} = \mathbf{H}\,\mathbf{f} + \epsilon$	$\mathbf{g} = \mathbf{F}\,\mathbf{h} + \epsilon$
$J(\mathbf{f}) = \Vert\mathbf{g} - \mathbf{H}\mathbf{f}\Vert^2 + \lambda_f \Vert\mathbf{D}_f\mathbf{f}\Vert^2$ $\nabla J(\mathbf{f}) = -2\mathbf{H}'(\mathbf{g} - \mathbf{H}\mathbf{f}) + 2\lambda_f \mathbf{D}_f'\mathbf{D}_f\mathbf{f}$ $\widehat{\mathbf{f}} = [\mathbf{H}'\mathbf{H} + \lambda_f \mathbf{D}_f'\mathbf{D}_f]^{-1}\mathbf{H}'\mathbf{g}$	$J(\mathbf{h}) = \Vert\mathbf{g} - \mathbf{F}\mathbf{h}\Vert^2 + \lambda_h \Vert\mathbf{D}_h\mathbf{h}\Vert^2$ $\nabla J(\mathbf{h}) = -2\mathbf{F}'(\mathbf{g} - \mathbf{F}\mathbf{h}) + 2\lambda_h \mathbf{D}_h'\mathbf{D}_h\mathbf{h}$ $\widehat{\mathbf{h}} = [\mathbf{F}'\mathbf{F} + \lambda_h \mathbf{D}_h'\mathbf{D}_h]^{-1}\mathbf{F}'\mathbf{g}$
$\widehat{f}(u,v) = \frac{H^*(u,v)}{\lvert H(u,v)\rvert^2 + \lambda_f \lvert D_f(u,v)\rvert^2}\, g(u,v)$ $\widehat{f}(u,v) = \frac{H^*(u,v)}{\lvert H(u,v)\rvert^2 + \frac{S_{\epsilon\epsilon}(u,v)}{S_{ff}(u,v)}}\, g(u,v)$	$\widehat{f}(u,v) = \frac{F^*(u,v)}{\lvert F(u,v)\rvert^2 + \lambda_h \lvert D_h(u,v)\rvert^2}\, g(u,v)$ $\widehat{f}(u,v) = \frac{F^*(u,v)}{\lvert F(u,v)\rvert^2 + \frac{S_{\epsilon\epsilon}(u,v)}{S_{ff}(u,v)}}\, g(u,v)$
$p(\mathbf{g}\mid\mathbf{f}) = \mathcal{N}(\mathbf{H}\mathbf{f}, \mathbf{\Sigma}_\epsilon)$ $p(\mathbf{f}) = \mathcal{N}(0, \mathbf{\Sigma}_f)$	$p(\mathbf{g}\mid\mathbf{h}) = \mathcal{N}(\mathbf{F}\mathbf{h}, \mathbf{\Sigma}_\epsilon)$ $p(\mathbf{h}) = \mathcal{N}(0, \mathbf{\Sigma}_h)$
$p(\mathbf{f}\mid\mathbf{g}) = \mathcal{N}(\widehat{\mathbf{f}}, \widehat{\mathbf{\Sigma}}_f)$ $\widehat{\mathbf{\Sigma}}_f = [\mathbf{H}'\mathbf{H} + \lambda_f \mathbf{D}_f'\mathbf{D}_f]^{-1}$ $\widehat{\mathbf{f}} = [\mathbf{H}'\mathbf{H} + \lambda_f \mathbf{D}_f'\mathbf{D}_f]^{-1}\mathbf{H}'\mathbf{g}$	$p(\mathbf{h}\mid\mathbf{g}) = \mathcal{N}(\widehat{\mathbf{h}}, \widehat{\mathbf{\Sigma}}_h)$ $\widehat{\mathbf{\Sigma}}_h = [\mathbf{F}'\mathbf{F} + \lambda_h \mathbf{D}_h'\mathbf{D}_h]^{-1}$ $\widehat{\mathbf{h}} = [\mathbf{F}'\mathbf{F} + \lambda_h \mathbf{D}_h'\mathbf{D}_h]^{-1}\mathbf{F}'\mathbf{g}$

Blind Deconvolution: Regularization

Deconvolution	Identification
$\mathbf{g} = \mathbf{H}\,\mathbf{f} + \epsilon$	$\mathbf{g} = \mathbf{F}\,\mathbf{h} + \epsilon$
$J(\mathbf{f}) = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 + \lambda_f \|\mathbf{D}_f \mathbf{f}\|^2$	$J(\mathbf{h}) = \|\mathbf{g} - \mathbf{F}\mathbf{h}\|^2 + \lambda_h \|\mathbf{D}_h \mathbf{h}\|^2$

- Joint Criterion

$$J(\mathbf{f}, \mathbf{h}) = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 + \lambda_f \|\mathbf{D}_f \mathbf{f}\|^2 + \lambda_h \|\mathbf{D}_h \mathbf{h}\|^2$$

- iterative algorithm

Deconvolution	Identification
$\nabla_f J(\mathbf{f}, \mathbf{h}) = -2\mathbf{H}'(\mathbf{g} - \mathbf{H}\mathbf{f}) + 2\lambda_f \mathbf{D}_f' \mathbf{D}_f \mathbf{f}$	$\nabla_h J(\mathbf{f}, \mathbf{h}) = -2\mathbf{F}'(\mathbf{g} - \mathbf{F}\mathbf{h}) + 2\lambda_h \mathbf{D}_h' \mathbf{D}_h \mathbf{h}$
$\widehat{\mathbf{f}} = [\mathbf{H}'\mathbf{H} + \lambda_f \mathbf{D}_f' \mathbf{D}_f]^{-1} \mathbf{H}' \mathbf{g}$	$\widehat{\mathbf{h}} = [\mathbf{F}'\mathbf{F} + \lambda_h \mathbf{D}_h' \mathbf{D}_h]^{-1} \mathbf{F}' \mathbf{g}$
$\widehat{f}(u, v) = \frac{H^*(u,v)}{\|H(u,v)\|^2 + \lambda_f \|D_f(u,v)\|^2}\, g(u, v)$	$\widehat{f}(u, v) = \frac{F^*(u,v)}{\|F(u,v)\|^2 + \lambda_h \|D_h(u,v)\|^2}\, g(u, v)$

Blind Deconvolution: Bayesian approach

Deconvolution	Identification
$\mathbf{g} = \mathbf{H}\,\mathbf{f} + \boldsymbol{\epsilon}$	$\mathbf{g} = \mathbf{F}\,\mathbf{h} + \boldsymbol{\epsilon}$
$p(\mathbf{g}\vert\mathbf{f}) = \mathcal{N}(\mathbf{H}\mathbf{f}, \boldsymbol{\Sigma}_\epsilon)$	$p(\mathbf{g}\vert\mathbf{h}) = \mathcal{N}(\mathbf{F}\mathbf{h}, \boldsymbol{\Sigma}_\epsilon)$
$p(\mathbf{f}) = \mathcal{N}(0, \boldsymbol{\Sigma}_f)$	$p(\mathbf{h}) = \mathcal{N}(0, \boldsymbol{\Sigma}_h)$
$p(\mathbf{f}\vert\mathbf{g}) = \mathcal{N}(\widehat{\mathbf{f}}, \widehat{\boldsymbol{\Sigma}}_f)$	$p(\mathbf{h}\vert\mathbf{g}) = \mathcal{N}(\widehat{\mathbf{h}}, \widehat{\boldsymbol{\Sigma}}_h)$
$\widehat{\boldsymbol{\Sigma}}_f = [\mathbf{H}'\mathbf{H} + \lambda_f \mathbf{D}_f'\mathbf{D}_f]^{-1}$	$\widehat{\boldsymbol{\Sigma}}_h = [\mathbf{F}'\mathbf{F} + \lambda_h \mathbf{D}_h'\mathbf{D}_h]^{-1}$
$\widehat{\mathbf{f}} = [\mathbf{H}'\mathbf{H} + \lambda_f \mathbf{D}_f'\mathbf{D}_f]^{-1}\mathbf{H}'\mathbf{g}$	$\widehat{\mathbf{h}} = [\mathbf{F}'\mathbf{F} + \lambda_h \mathbf{D}_h'\mathbf{D}_h]^{-1}\mathbf{F}'\mathbf{g}$

- Joint posterior law:

$$p(\mathbf{f},\mathbf{h}|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f},\mathbf{h})\,p(\mathbf{f})\,p(h\mathbf{h})$$

$$p(\mathbf{f},\mathbf{h}|\mathbf{g}) \propto \exp\{-J(\mathbf{f},\mathbf{h})\}$$

with

$$J(\mathbf{f},\mathbf{h}) = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 + \lambda_f\|\mathbf{D}_f\mathbf{f}\|^2 + \lambda_h\|\mathbf{D}_h\mathbf{h}\|^2$$

- iterative algorithm

Blind Deconvolution: Bayesian Joint MAP criterion

- Joint posterior law:

$$p(\mathbf{f},\mathbf{h}|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f},\mathbf{h})\, p(\mathbf{f})\, p(h\mathbf{h})$$

$$p(\mathbf{f},\mathbf{h}|\mathbf{g}) \propto \exp\{-J(\mathbf{f},\mathbf{h})\}$$

with

$$J(\mathbf{f},\mathbf{h}) = \|\mathbf{g}-\mathbf{H}\mathbf{f}\|^2 + \lambda_f \|\mathbf{D}_f\mathbf{f}\|^2 + \lambda_h \|\mathbf{D}_h\mathbf{h}\|^2$$

- iterative algorithm

Deconvolution	Identification
$p(\mathbf{g}\mid\mathbf{f},\mathbf{H}) = \mathcal{N}(\mathbf{H}\mathbf{f},\mathbf{\Sigma}_\epsilon)$	$p(\mathbf{g}\mid\mathbf{h},\mathbf{F}) = \mathcal{N}(\mathbf{F}\mathbf{h},\mathbf{\Sigma}_\epsilon)$
$p(\mathbf{f}) = \mathcal{N}(0,\mathbf{\Sigma}_f)$	$p(\mathbf{h}) = \mathcal{N}(0,\mathbf{\Sigma}_h)$
$p(\mathbf{f}\mid\mathbf{g},\mathbf{H}) = \mathcal{N}(\widehat{\mathbf{f}},\widehat{\mathbf{\Sigma}}_f)$	$p(\mathbf{h}\mid\mathbf{g},\mathbf{F}) = \mathcal{N}(\widehat{\mathbf{h}},\widehat{\mathbf{\Sigma}}_h)$
$\widehat{\mathbf{\Sigma}}_f = [\mathbf{H}'\mathbf{H}+\lambda_f\mathbf{D}_f'\mathbf{D}_f]^{-1}$	$\widehat{\mathbf{\Sigma}}_h = [\mathbf{F}'\mathbf{F}+\lambda_h\mathbf{D}_h'\mathbf{D}_h]^{-1}$
$\widehat{\mathbf{f}} = [\mathbf{H}'\mathbf{H}+\lambda_f\mathbf{D}_f'\mathbf{D}_f]^{-1}\mathbf{H}'\mathbf{g}$	$\widehat{\mathbf{h}} = [\mathbf{F}'\mathbf{F}+\lambda_h\mathbf{D}_h'\mathbf{D}_h]^{-1}\mathbf{F}'\mathbf{g}$

Blind Deconvolution: Marginalization and EM algorithm

- Joint posterior law:
- Marginalization

$$p(\mathbf{f},\mathbf{h}|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f},\mathbf{h})\, p(\mathbf{f})\, p(h\mathbf{h})$$

$$p(\mathbf{h}|\mathbf{g}) = \int p(\mathbf{f},\mathbf{h}|\mathbf{g})\, \mathrm{d}\mathbf{f}$$

$$\widehat{\mathbf{h}} = \arg\max_{\mathbf{h}} \{p(\mathbf{h}|\mathbf{g})\} \longrightarrow \widehat{\mathbf{f}} = \arg\max_{\mathbf{f}} \left\{p(\mathbf{f}|\mathbf{g},\widehat{\mathbf{h}})\right\}$$

- Expression of $p(\mathbf{h}|\mathbf{g})$ and its maximization are complexes
- Expectation-Maximization Algorithm

$$\ln p(\mathbf{f},\mathbf{h}|\mathbf{g}) \propto J(\mathbf{f},\mathbf{h}) = \|\mathbf{g}-\mathbf{H}\mathbf{f}\|^2 + \lambda_f\|\mathbf{D}_f\mathbf{f}\|^2 + \lambda_h\|\mathbf{D}_h\mathbf{h}\|^2$$

  - Iterative algorithm
  - Expectation: Compute

$$Q(\mathbf{h},\mathbf{h}^{k-1}) = \mathrm{E}_{p(\mathbf{f},\mathbf{h}^{k-1}|\mathbf{g})}\{J(\mathbf{f},\mathbf{h})\} = \langle \ln p(\mathbf{f},\mathbf{h}|\mathbf{g})\rangle_{p(\mathbf{f},\mathbf{h}^{k-1}|\mathbf{g})}$$

  - Maximization:

$$\mathbf{h}^k = \arg\max_{\mathbf{h}}\left\{Q(\mathbf{h},\mathbf{h}^{k-1})\right\}$$

Blind Deconvolution: Variational Bayesian Approximation algorithm

- Joint posterior law:

$$p(\mathbf{f},\mathbf{h}|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f},\mathbf{h})\, p(\mathbf{f})\, p(h\mathbf{h})$$

- Approximation: $p(\mathbf{f},\mathbf{h}|\mathbf{g})$ by $q(\mathbf{f},\mathbf{h}|\mathbf{g}) = q_1(\mathbf{f})\, q_2(\mathbf{h})$
- Criterion of approximation: Kullback-Leiler

$$\mathrm{KL}(q|p) = \int q \ln \frac{q}{p} = \int q_1\, q_2 \ln \frac{q_1\, q_2}{p}$$

$$\begin{aligned}\mathrm{KL}(q_1\, q_2|p) &= \int q_1 \ln q_1 + \int q_2 \ln q_2 - \int q \ln p \\ &= -\mathcal{H}(q_1) - \mathcal{H}(q_2) + \langle - \ln p((\mathbf{f},\mathbf{h}|\mathbf{g})\rangle_q\end{aligned}$$

- When the expression of q_1 and q_2 are obtained, use them.

Variational Bayesian Approximation algorithm

- Kullback-Leibler criterion

$$\begin{aligned} \mathrm{KL}(q_1\, q_2|p) &= \int q_1 \ln q_1 + \int q_2 \ln q_2 + \int q \ln p \\ &= -\mathcal{H}(q_1) - \mathcal{H}(q_2) + \langle -\ln p((\mathbf{f},\mathbf{h}|\mathbf{g})\rangle_q \end{aligned}$$

- Free energy

$$\mathcal{F}(q_1\, q_2) = -\langle \ln p((\mathbf{f},\mathbf{h}|\mathbf{g})\rangle_{q_1 q_2}$$

- Equivalence between optimization of $\mathrm{KL}(q_1\, q_2|p)$ and $\mathcal{F}(q_1\, q_2)$
- Alternate optimization:

$$\begin{aligned} \widehat{q}_1 &= \arg\min_{q_1} \{\mathrm{KL}(q_1\, q_2|p)\} = \arg\min_{q_1} \{\mathcal{F}(q_1\, q_2)\} \\ \widehat{q}_2 &= \arg\min_{q_2} \{\mathrm{KL}(q_1\, q_2|p)\} = \arg\min_{q_2} \{\mathcal{F}(q_1\, q_2)\} \end{aligned}$$

Blind Deconvolution algorithm

$$\begin{tikzcd}
\mathbf{h}^{(0)} \arrow[r] & \mathbf{H} \arrow[r] & \boxed{\left(\mathbf{H}'\mathbf{H} + \lambda_f \mathbf{D}_f' \mathbf{D}_f\right)^{-1} \mathbf{H}'\mathbf{g}} \arrow[r] & \widehat{\mathbf{f}}(t) \arrow[d] \\
 & \widehat{\mathbf{h}} \arrow[u] & \boxed{\left(\mathbf{F}'\mathbf{F} + \lambda_h \mathbf{D}_h' \mathbf{D}_h\right)^{-1} \mathbf{F}'\mathbf{g}} \arrow[l] & \mathbf{F} \arrow[l]
\end{tikzcd}$$

Joint Estimation of **h** and **f** with a Gaussian prior model..

VBA: $p(\mathbf{f},\mathbf{h}|\mathbf{g}) \longrightarrow q_1(\mathbf{f}|\mathbf{h},\mathbf{g})\; q_2(\mathbf{h}|\mathbf{f},\mathbf{g})$

$$q_1(\mathbf{f}) = \mathcal{N}(\widehat{\mathbf{f}}, \widehat{\boldsymbol{\Sigma}}_f)$$
$$\left\{ \begin{array}{l} \widehat{\boldsymbol{\Sigma}}_f = (\mathbf{F}'\mathbf{F} + \lambda_f \boldsymbol{\Sigma}_f)^{-1} \\ \widehat{\mathbf{f}}(t) = (\mathbf{F}'\mathbf{F} + \lambda_f \boldsymbol{\Sigma}_f)^{-1}\mathbf{F}'\mathbf{g}(t), \quad \lambda_f = v_\epsilon / v_f \end{array} \right.$$

$$q_2(\mathbf{h}) = \mathcal{N}(\widehat{\mathbf{h}}, \widehat{\boldsymbol{\Sigma}}_h)$$
$$\left\{ \begin{array}{l} \widehat{\boldsymbol{\Sigma}}_h = (\mathbf{H}'\mathbf{H} + \lambda_h \widehat{\boldsymbol{\Sigma}}_h)^{-1} \\ \widehat{\mathbf{h}} = (\mathbf{H}'\mathbf{H} + \lambda_h \widehat{\boldsymbol{\Sigma}}_h)^{-1} \qquad \lambda_a = v_\epsilon / v_a \end{array} \right.$$

Joint Estimation of h and f with a Gaussian prior model..

- Joint MAP:

$$\begin{tikzcd}
\mathbf{h}^{(0)} \arrow[r] & \mathbf{H} \arrow[r] & \boxed{\left(\mathbf{H}'\mathbf{H} + \lambda_f \boldsymbol{\Sigma}_f\right)^{-1} \mathbf{H}'\mathbf{g}} \arrow[r] & \widehat{\mathbf{f}}(t) \arrow[d] \\
 & \widehat{\mathbf{h}} \arrow[u] & \boxed{\left(\mathbf{F}'\mathbf{F} + \lambda_h \boldsymbol{\Sigma}_h\right)^{-1} \mathbf{F}'\mathbf{g}} \arrow[l] & \mathbf{F} \arrow[l]
\end{tikzcd}$$

- VBA:

$$\begin{tikzcd}
\begin{array}{l} \mathbf{h}^{(0)} \longrightarrow \mathbf{H} \longrightarrow \\ \boldsymbol{\Sigma}_f^{(0)} \longrightarrow \boldsymbol{\Sigma}_f \longrightarrow \end{array} & \boxed{\begin{array}{c} \widehat{\mathbf{f}} = \left(\mathbf{H}'\mathbf{H} + \lambda_f \boldsymbol{\Sigma}_f\right)^{-1} \mathbf{H}'\mathbf{g} \\ \widehat{\boldsymbol{\Sigma}}_f = (\mathbf{H}'\mathbf{H} + \lambda_f \boldsymbol{\Sigma}_f)^{-1} \end{array}} & \begin{array}{l} \longrightarrow \widehat{\mathbf{f}} \\ \longrightarrow \widehat{\boldsymbol{\Sigma}} \end{array} \arrow[d, Rightarrow] \\
\begin{array}{l} \widehat{\mathbf{h}} \longleftarrow \\ \widehat{\mathbf{V}} \longleftarrow \end{array} \arrow[u, Rightarrow] & \boxed{\begin{array}{c} \widehat{\mathbf{h}} = \left(\mathbf{F}'\mathbf{F} + \lambda_h \boldsymbol{\Sigma}_h\right)^{-1} \mathbf{F}'\mathbf{g} \\ \widehat{\mathbf{V}} = (\mathbf{F}'\mathbf{F} + \lambda_h \boldsymbol{\Sigma}_h)^{-1} \end{array}} & \begin{array}{l} \longleftarrow \mathbf{F} \\ \longleftarrow \boldsymbol{\Sigma}_h \end{array}
\end{tikzcd}$$

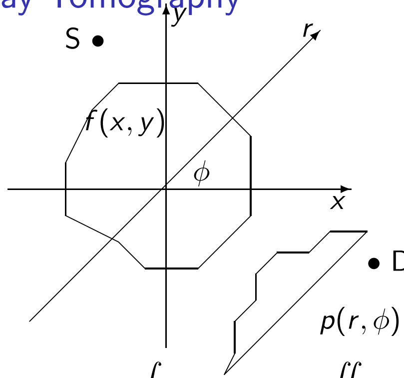
Content

1. Introduction: Signals and Images, Linear transformations (Convolution, Fourier, Laplace, Hilbert, Radon, ..., Discrete convolution, Z transform, DFT, FFT, ...)
2. Modeling: parametric and non-parametric, MA, AR and ARMA models, Linear prediction
3. Deconvolution and Parameter Estimation: Deterministic (LS, Regularization) and Probabilistic methods (Wiener Filtering)
4. Inverse problems and Bayesian estimation
5. Kalman Filtering and smoothing
6. Case study: Signal deconvolution
7. Case study: Image restoration
8. **Case study: Image reconstruction and Computed Tomography**

8. Case study: Image reconstruction and Computed Tomography

- X ray Computed Tomography, Radon transform
- Analytical inversion methods
- Discretization and algebraic methods
- Bayesian approach to CT
- Gauss-Markov-Potts model for images
- 3D CT
- Practical issues and computational costs

X ray Tomography



$$I = I_0 \exp\left\{-\mu x\right\}$$

$$I = I_0 \exp\left\{-\sum_j \mu_j x_j\right\}$$

$$I = I_0 \exp\left\{-\int_L f(x)\,\mathrm{d}x\right\}$$

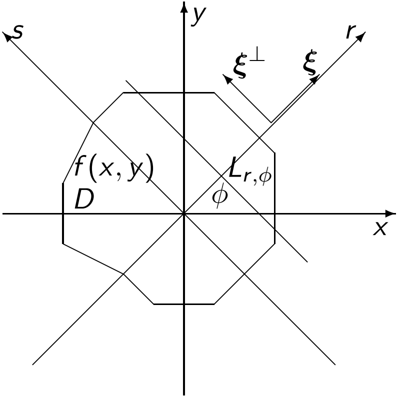
$$-\ln\frac{I}{I_0} = \int_L f(x,y)\,\mathrm{d}l$$

$$p(r,\phi) = \int_L f(x,y)\,\mathrm{d}l = \iint f(x,y)\delta(r - x\cos\phi - y\sin\phi)\,\mathrm{d}x\,\mathrm{d}y$$

$$f(x,y) \longrightarrow \boxed{\text{Radon Transform}} \longrightarrow p(r,\phi)$$

$$p(r,\phi) \longrightarrow \boxed{\begin{array}{c}\text{Reconstruction}\\ = \\ \text{Inversion de la TR}\end{array}} \longrightarrow f(x,y)$$

Radon Transform: 2D case



$$\boldsymbol{\xi} = [\cos\phi, \sin\phi]', \quad \boldsymbol{\xi}^{\perp} = [-\sin\phi, \cos\phi]'$$

$$\mathbf{x} = [x, y]' = [\rho\cos\theta, \rho\sin\theta], \qquad L_{r,\phi} : r = x\cos\phi + y\sin\phi = \boldsymbol{\xi}' \cdot \mathbf{x}$$

Radon Transform: Different formulations

- 2D case

$$g(r,\phi)=\int_L f(x,y)\,\mathrm{d}l=\iint_D f(x,y)\delta(r-x\cos\phi-y\sin\phi)\,\mathrm{d}x\,\mathrm{d}y$$

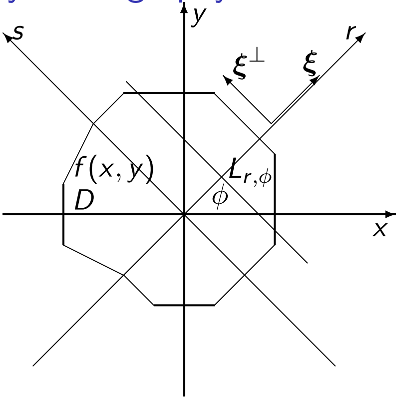
- 2D polar coordinates:

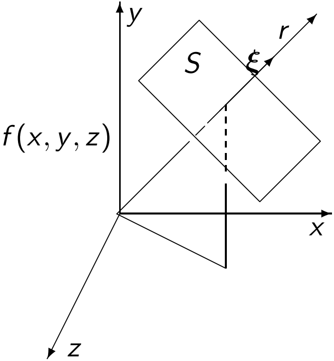
$$g(r,\phi)=\int_0^\infty\int_0^{2\pi} f(\rho,\theta)\delta\big(r-\rho\cos(\phi-\theta)\big)\rho\,\mathrm{d}\rho\,\mathrm{d}\theta$$

- nD case

$$g(r,\boldsymbol{\xi})=\iint f(x,y)\delta(r-\boldsymbol{\xi}'\cdot\mathbf{x})\,\mathrm{d}x\,\mathrm{d}y$$

X ray Tomography: 3D case





$$\mathbf{x}=[x,y,z]', \qquad d\mathbf{x}=\,\mathrm{d}x\,\mathrm{d}y dz, \quad r=\boldsymbol{\xi}'\cdot\mathbf{x}=\xi_1 x+\xi_2 y+\xi_3 z$$

$$g(r,\boldsymbol{\xi})=\int f(\mathbf{x})\delta(r-\boldsymbol{\xi}'\cdot\mathbf{x})\,\mathrm{d}\mathbf{x}$$

$$g(r,\boldsymbol{\xi})=\iiint f(x,y,z)\delta(r-x\sin\theta\cos\phi-y\sin\theta\sin\phi-z\cos\theta)\,\mathrm{d}x\,\mathrm{d}y dz$$

X ray Tomography: *n*-D case

$$\mathbf{x} = [x_1, x_2, \ldots, x_n]', \quad \boldsymbol{\xi} = [\xi_1, \xi_2, \ldots, \xi_n]' \quad \text{avec} \quad |\boldsymbol{\xi}| = 1$$

$$\mathrm{d}\mathbf{x} = \mathrm{d}x_1 \, \mathrm{d}x_2 \ldots \mathrm{d}x_n, \quad r = \boldsymbol{\xi}' \cdot \mathbf{x} = \xi_1 x_1 + \xi_2 x_2 + \ldots + \xi_n x_n$$

$$\boxed{g(r, \boldsymbol{\xi}) = \int_{R^n} f(\mathbf{x}) \delta(r - \boldsymbol{\xi}' \cdot \mathbf{x}) \, \mathrm{d}\mathbf{x}}$$

Radon Transform: some properties

- Definition:

$$f(x,y) \xrightarrow{\mathcal{R}} g(r,\phi) = \iint f(x,y)\delta(r - x\cos\phi - y\sin\phi)\,\mathrm{d}x\,\mathrm{d}y$$

- Definition in Polar Coordinates

$$f(\rho,\theta) \xrightarrow{\mathcal{R}} g(r,\phi) = \int_0^{\infty}\int_0^{2\pi} f(\rho,\theta)\delta(r-\rho\cos(\phi-\theta)\rho\,\mathrm{d}\rho\,\mathrm{d}\theta$$

	function	Radon Transform
Linearity:	$af_1(x,y) + bf_2(x,y)$	$ag_1(r,\phi) + bg_2(r,\phi)$
Symetry:	$f(x,y)$	$g(r,\phi) = g(-r,\phi \pm \pi)$
Periodicity:	$f(x,y)$	$g(r,\phi) = g(r,\phi \pm 2k\pi)$
Translation:	$f(x-x_0, y-y_0)$	$g(r - x_0\cos\phi - y_0\sin\phi, \phi)$
Rotation :	$f(\rho,\theta+\theta_0)$	$g(r,\phi+\theta_0)$
Scale change:	$f(ax,ay)$	$\frac{1}{\|a\|}g(ar,\phi), \quad a \neq 0$
Mass Conservation:	$M = \iint_{-\infty}^{\infty} f(x,y)\,\mathrm{d}x\,\mathrm{d}y$	$M = \int_{-\infty}^{+\infty} g(r,\phi)\,\mathrm{d}r$

Radon Transform: *n*–D case

- Definition

$$\mathcal{R}\{f(\mathbf{x})\} = g(r, \boldsymbol{\xi}) = \int f(\mathbf{x}) \delta(r - \boldsymbol{\xi}' \cdot \mathbf{x}) \, \mathrm{d}\mathbf{x}$$

- Linearity

$$\mathcal{R}\{a\, f(\mathbf{x}) + b\, g(\mathbf{x})\} = a\, \mathcal{R}\{f(x)\} + b\, \mathcal{R}\{g(\mathbf{x})\}$$

- Symetry

$$g(-r, -\boldsymbol{\xi}) = g(r, \boldsymbol{\xi})$$

- Homogeneity

$$g(sr, s\boldsymbol{\xi}) = \frac{1}{|s|} \int f(\mathbf{x}) \delta(r - \boldsymbol{\xi}' \cdot \mathbf{x}) \, \mathrm{d}\mathbf{x}$$

$$g(sr, s\boldsymbol{\xi}) = \frac{1}{|s|} g(r, \boldsymbol{\xi})$$

$$g(r, s\boldsymbol{\xi}) = \frac{1}{s} g(\frac{r}{s}, \boldsymbol{\xi}), \quad s > 0$$

$$g(sr, \boldsymbol{\xi}) = s g(r, \boldsymbol{\xi})$$

- Axis transformation

Radon Transform: *n*–D case

- RT of the derivative of a function
- Derivative:

$$\mathcal{R}\{\frac{\partial f}{\partial x_k}\} = \xi_k \frac{\partial g(r,\boldsymbol{\xi})}{\partial r}$$

$$\nabla f = \left[\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}\right]' \quad \mathbf{a}.\nabla f = \left[a_1 \frac{\partial f}{\partial x_1}, \ldots, a_n \frac{\partial f}{\partial x_n}\right]'$$

$$\mathcal{R}\{\mathbf{a}.\nabla f\} = \mathbf{a}.\boldsymbol{\xi} \frac{\partial g(r,\boldsymbol{\xi})}{\partial r}$$

- Second Derivative:

$$\mathcal{R}\{\frac{\partial^2 f}{\partial x_l \partial x_k}\} = \xi_l \, \xi_k \frac{\partial g(r,\boldsymbol{\xi})}{\partial r}$$

$$\Delta f(\mathbf{x}) = \sum_{k=1}^{n} \frac{\partial f}{\partial x_k}$$

$$\mathcal{R}\{\Delta f\} = |\boldsymbol{\xi}|^2 \frac{\partial g(r,\boldsymbol{\xi})}{\partial r} = \frac{\partial g(r,\boldsymbol{\xi})}{\partial r}$$

Radon Transform: *n*–D case

- Derivative of the RT

$$\frac{\partial g(r,\boldsymbol{\xi})}{\partial \xi_k} = \int f(\mathbf{x}) \frac{\partial}{\partial \xi_k} \delta(r - \boldsymbol{\xi}'.\mathbf{x}) d\mathbf{x}$$

$$\frac{\partial}{\partial \xi_k} \mathcal{R}\{f(\mathbf{x})\} = \frac{\partial g(r,\boldsymbol{\xi})}{\partial \xi_k} = -\frac{\partial}{\partial r} \mathcal{R}\{x_k\, f(\mathbf{x})\}$$

$$\frac{\partial^2}{\partial \xi_l \partial \xi_k} \mathcal{R}\{f(\mathbf{x})\} = \frac{\partial^2 g(r,\boldsymbol{\xi})}{\partial \xi_l \partial \xi_k} = -\frac{\partial}{\partial r} \mathcal{R}\{x_l\, x_k\, f(\mathbf{x})\}$$

- Convolution

$$f(\mathbf{x}) = g * h = \int g(\mathbf{y}) h(\mathbf{x} - \mathbf{y})\, \mathrm{d}\mathbf{y}$$

$$g(r,\boldsymbol{\xi}) = \widetilde{g} * \widetilde{h}$$

Summary

$$\begin{array}{rcl}
f(\mathbf{x}) & \xrightarrow{TR} & g(r,\boldsymbol{\xi}) \\
c_1 f_1(\mathbf{x}) + c_2 f_2(\mathbf{x}) & \xrightarrow{TR} & c_1 g_1(r,\boldsymbol{\xi}) + c_2 g_2(r,\boldsymbol{\xi}) \\
f(\mathbf{A}\mathbf{x}) & \xrightarrow{TR} & |\mathbf{A}^{-1}| g(r, \mathbf{A}^{-t}\mathbf{x}) \\
f(\mathbf{A}^{-1}\mathbf{x}) & \xrightarrow{TR} & |\mathbf{A}| g(r, \mathbf{A}^{t}\mathbf{x}) \\
f(c\,\mathbf{x}) & \xrightarrow{TR} & \frac{1}{c^n}\, g(r, \boldsymbol{\xi}/c) = \frac{1}{c^{n-1}}\, g(cr, \boldsymbol{\xi}) \\
f(\mathbf{x} \pm \mathbf{a}) & \xrightarrow{TR} & g(r \pm \boldsymbol{\xi}'.\mathbf{a}, \boldsymbol{\xi})
\end{array}$$

$$\begin{array}{rcl}
g(sr, s\boldsymbol{\xi}) & = & \frac{1}{|s|}\, g(r,\boldsymbol{\xi}) \\
g(-r, -\boldsymbol{\xi}) & = & g(r,\boldsymbol{\xi}) \\
g(sr, \boldsymbol{\xi}) & = & s\, g(r,\boldsymbol{\xi})
\end{array}$$

Summary

Derivative:	$\frac{\partial f}{\partial x_k}$	$\xi_k \frac{\partial g(r,\boldsymbol{\xi})}{\partial r}$
Gradient :	$\nabla f = \left[\frac{\partial f}{\partial x_1},\ldots,\frac{\partial f}{\partial x_n}\right]'$	$\frac{\partial g(r,\boldsymbol{\xi})}{\partial r}\left[\xi_1,\ldots,\xi_n\right]'$
	$\mathbf{a}.\nabla f = a_1\frac{\partial f}{\partial x_1}+\cdots+a_1\frac{\partial f}{\partial x_n}$	$\mathbf{a}.\boldsymbol{\xi}\frac{\partial g(r,\boldsymbol{\xi})}{\partial r}$
Second Derivative:	$\frac{\partial^2 f}{\partial x_l \partial x_k}$	$\xi_l\xi_k\frac{\partial g(r,\boldsymbol{\xi})}{\partial r}$
Laplacian:	$\Delta f(\mathbf{x}) = \sum_{k=1}^{n}\frac{\partial f}{\partial x_k}$	$\frac{\partial g(r,\boldsymbol{\xi})}{\partial r}$

Link between RT and FT

$$\mathbf{x} = [x_1, \ldots, x_n]', \quad \boldsymbol{\omega} = [\omega_1, \ldots, \omega_n]'$$

$$\left\{ \begin{array}{rcl} \widehat{f}(\boldsymbol{\omega}) & = & \mathcal{F}_n \left\{ f(\mathbf{x}) \right\} = \int f(\mathbf{x}) \exp\left\{-j\boldsymbol{\omega}' \cdot \mathbf{x}\right\} \, \mathrm{d}\mathbf{x} \\ f(\mathbf{x}) & = & \mathcal{F}_n^{-1} \left\{ \widehat{f}(\boldsymbol{\omega}) \right\} = \int \widehat{f}(\omega) \exp\left\{+j\boldsymbol{\omega}' \cdot \mathbf{x}\right\} \, \mathrm{d}\boldsymbol{\omega} \end{array} \right.$$

$$\left\{ \begin{array}{rcl} \widehat{g}(\Omega, \boldsymbol{\xi}) & = & \mathcal{F}_1 \left\{ g(r, \boldsymbol{\xi}) \right\} = \int g(r, \boldsymbol{\xi}) \exp\left\{-j\Omega r\right\} \, \mathrm{d}r \\ g(r, \boldsymbol{\xi}) & = & \mathcal{F}_1^{-1} \left\{ \widehat{g}(\Omega, \boldsymbol{\xi}) \right\} = \int \widehat{g}(\Omega, \boldsymbol{\xi}) \exp\left\{+j\Omega r\right\} \, \mathrm{d}\Omega \end{array} \right.$$

$$\widehat{f}(\boldsymbol{\omega}) = \int g(r, \boldsymbol{\xi}) \exp\left\{-j\Omega r\right\} \, \mathrm{d}r = \widehat{g}(\Omega, \boldsymbol{\xi}) \quad \forall \boldsymbol{\omega} = \Omega \boldsymbol{\xi}$$

$$\left\{ \begin{array}{rcl} \widehat{f} & = & \mathcal{F}_n \left\{ f \right\} = \mathcal{F}_1 \left\{ g \right\} = \mathcal{F}_1 \left\{ \mathcal{R}\{f\} \right\} \\ \widetilde{f} & = & \mathcal{R}\{f\} = \mathcal{F}_1^{-1} \left\{ \widehat{f} \right\} = \mathcal{F}_1^{-1} \left\{ \mathcal{F}_n \left\{ f \right\} \right\} \end{array} \right.$$

Direct inversion of the RT: Odd dimensions

$$f(\mathbf{x}) = C_n \int_{|\boldsymbol{\xi}|=1} \left[(\frac{\partial}{\partial r})^{n-1} g(r, \boldsymbol{\xi}) \right]_{r=\boldsymbol{\xi}'.\mathbf{x}} \mathrm{d}\boldsymbol{\xi}$$

$$C_n = \frac{1}{2} \frac{(-1)^{(n-1)/2}}{(2\pi)^{(n-1)}} = \frac{1}{2} \frac{1}{(2j\pi)^{(n-1)}}$$

Example :
$n = 3,\ C_3 = \frac{-1}{8\pi^2}$, $\mathbf{x} = [x, y, z]'$,
$\boldsymbol{\xi} = [\xi_1, \xi_2, \xi_3]' = [\sin\theta \cos\phi,\ \sin\theta \sin\phi,\ \cos\theta]'$ and
$d\boldsymbol{\xi} = \sin\theta\, d\theta\, d\phi$.

$$f(\mathbf{x}) = \frac{-1}{8\pi^2} \int_{|\boldsymbol{\xi}|=1} \left[\frac{\partial^2}{\partial r^2} g(r, \boldsymbol{\xi}) \right]_{r=\boldsymbol{\xi}'.\mathbf{x}} d\boldsymbol{\xi}$$

Direct inversion of the RT: Odd dimensions

Spherical coordinates : (ρ, θ, ϕ):

$$g(r, \boldsymbol{\xi}) = \iiint f(x, y, z)\delta(r - x \sin\theta \cos\phi + y \sin\theta \sin\phi + z \cos\theta) \, \mathrm{d}x \; \mathrm{d}ydz$$

$$\overline{g}(r, \boldsymbol{\xi}) = \frac{-1}{8\pi^2} \frac{\partial^2 g(r, \boldsymbol{\xi})}{\partial r^2}$$

$$f(x, y, z) = \mathcal{B}\left\{\overline{g}\right\} \overset{\triangle}{=} \int_0^{\pi} \int_0^{2\pi} \overline{g}(r, \boldsymbol{\xi}) \sin\theta \; \mathrm{d}\phi \; \mathrm{d}\theta$$

$$f(x, y, z) \longrightarrow \boxed{\;\mathcal{R}\;} \longrightarrow g(r, \boldsymbol{\xi})$$

$$g(r, \boldsymbol{\xi}) \longrightarrow \boxed{\frac{-1}{8\pi^2} \frac{\partial^2}{\partial r^2}} \longrightarrow \overline{g}(r, \boldsymbol{\xi}) \longrightarrow \boxed{\;\mathcal{B}\;} \longrightarrow f(x, y, z)$$

Even dimensions

$$f(\mathbf{x}) = \frac{C_n}{j\pi} \int_{|\boldsymbol{\xi}|=1} \mathrm{d}\boldsymbol{\xi} \int \mathrm{d}r \, \frac{(\frac{\partial}{\partial r})^{n-1} g(r, \boldsymbol{\xi})}{(r - \boldsymbol{\xi}'.\mathbf{x})}$$

Example : $n = 2,\ C_2 = \frac{-1}{2\pi}$,
$\mathbf{x} = [x, y]'$, $\boldsymbol{\xi} = [\xi_1,\ \xi_2]' = [\cos\phi,\ \sin\phi]',\ d\boldsymbol{\xi} = d\phi.$

$$f(x, y) = \left(-\frac{1}{2\pi^2}\right) \int_0^{\pi} d\phi \int_{-\infty}^{+\infty} \mathrm{d}r \, \frac{\frac{\partial}{\partial r} g(r, \phi)}{(r - x\cos\phi - y\sin\phi)}$$

$$g(r, \phi) = \iint f(x, y) \delta(r - x\cos\phi + y\sin\phi) \, \mathrm{d}x \, \mathrm{d}y$$

Even dimensions

$$\overline{g}(r,\phi) = \frac{-1}{2\pi^2}\frac{\partial g(r,\phi)}{\partial r}$$

$$g(r,\phi) = \mathcal{H}\left\{\overline{g}(r,\phi)\right\} = \int \frac{\overline{g}(r,\phi)}{(r - x\cos\phi - y\sin\phi)}\,\mathrm{d}r$$

$$f(x,y) = \mathcal{B}\left\{g(r,\phi)\right\} \overset{\triangle}{=} \int_0^{\pi} g(r,\phi)\,\mathrm{d}\phi$$

$$f(x,y) \longrightarrow \boxed{\mathcal{R}} \longrightarrow g(r,\phi)$$

$$g(r,\phi) \rightarrow \boxed{\frac{-1}{2\pi^2}\frac{\partial}{\partial r}} \rightarrow \overline{g}(r,\phi) \rightarrow \boxed{\mathcal{H}} \rightarrow g(x,y) \rightarrow \boxed{\mathcal{B}} \rightarrow f(x,y)$$

Cylindrical coordinates: $(\rho,\theta) \longrightarrow x = \rho\cos\theta$ and $y = \rho\sin\theta$,

$$f(\rho,\theta) = \left(-\frac{1}{2\pi^2}\right)\int_0^{\pi} d\phi \int_{-\infty}^{+\infty} \mathrm{d}r\,\frac{\frac{\partial}{\partial r}g(r,\xi)}{(r - \rho cos(\phi-\theta))}$$

Inversion using the FT

$$\begin{tikzcd}
f(x,y) \arrow[rr, "\mathcal{R}"] \arrow[dr, shift right, "\mathcal{F}_2"'] & & p(r,\phi) \arrow[dl, shift left, "\mathcal{F}_1"] \\
& \widehat{f}(\omega_x,\omega_y) \arrow[ul, shift right, "\mathcal{F}_2^{-1}"'] \arrow[ur, shift left, "\mathcal{F}_1^{-1}"] &
\end{tikzcd}$$

$$\mathcal{F}_2\left\{f\right\} = \mathcal{F}_1\left\{\mathcal{R}\{f\}\right\} = \mathcal{F}_1\left\{g\right\}$$

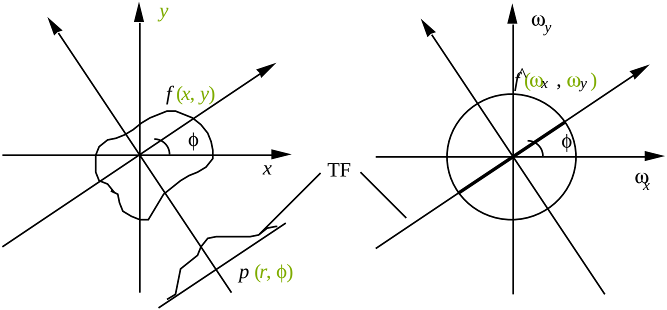
$$G(\Omega,\boldsymbol{\xi}) = G(\Omega,\phi) = \int g(r,\phi)\exp\left\{-j\Omega r\right\}\,\mathrm{d}r$$

$$\widehat{f}(\omega_x,\omega_y) = \iint f(x,y)\exp\left\{-j(\omega_x\,x+\omega_y\,y)\right\}\,\mathrm{d}x\,\mathrm{d}y$$

$$f(x,y) = \iint \widehat{f}(\omega_x,\omega_y)\exp\left\{+j(\omega_x\,x+\omega_y\,y)\right\}\,\mathrm{d}\omega_x\,\mathrm{d}\omega_y$$

$$\widehat{f}(\omega_x,\omega_y) = G(\Omega,\phi) \quad \text{pour} \quad \omega_x = \Omega\sin\phi \quad \text{et} \quad \omega_y = \Omega\cos\phi$$

Inversion using the FT



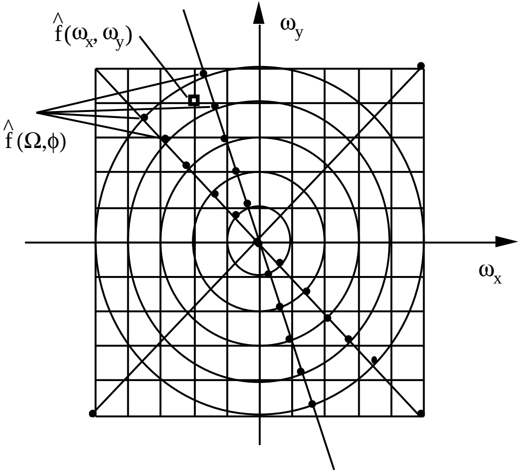
Inversion using FT

$$\begin{tikzcd}
f(x,y) \arrow[r, "\mathcal{R}"] & g(r,\phi) \arrow[d, "\mathcal{F}_1"] \\
\widehat{f}(\omega_x,\omega_y) \arrow[u, "\mathcal{F}_2^{-1}"] & G(\Omega,\phi) \arrow[l, "{\omega_x = \Omega\cos\phi\,,\ \omega_y = \Omega\sin\phi}"']
\end{tikzcd}$$

Algorithm:

- $p(r,\phi_m) \longrightarrow$ 1D FT $\longrightarrow \hat{p}(\Omega,\phi_m), \quad m = 1,\cdots,M$
- Interpolation in the Fourier domaine $\longrightarrow \widehat{f}(\omega_x,\omega_y)$
- $\widehat{f}(\omega_x,\omega_y) \longrightarrow$ 2D FT $f(x,y)$

Inversion using FT



Inversion using Backprojection

For a function $p(r, \phi)$ of $r \in R$ and $\phi \in [0, \pi]$, and
$r = x \cos \phi + y \sin \phi$, $\forall (x, y) \in R^2$

$$\mathcal{B}\left\{p\right\}(x, y) = \int_0^\pi p(x \cos \phi + y \sin \phi, \phi) d\phi$$

This operator is the adjoint operator of the Radon transform $\mathcal{R}$
Polar coordinates:

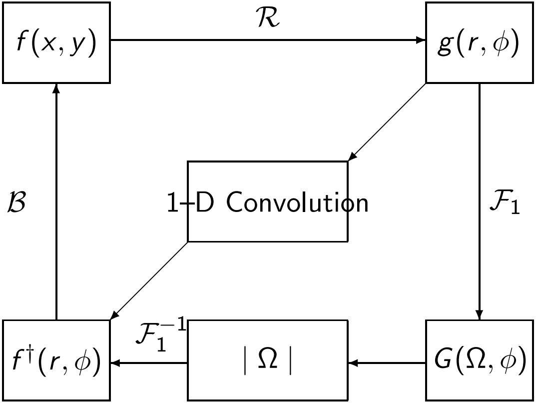
$$\mathcal{B}\left\{p\right\}(x, y) = \int_0^\pi p(\rho \cos(\theta - \phi), \phi) d\phi$$

$$p(r, \phi) = \mathcal{R}\{f(x, y)\}, \quad f_b(x, y) = \mathcal{B}\left\{p(r, \phi)\right\} = \mathcal{B}\left\{\mathcal{R}\{f\}\right\}$$

$$f_b(x, y) = f(x, y) * \frac{1}{|r|} = f(x, y) * \frac{1}{\sqrt{x^2 + y^2}}$$

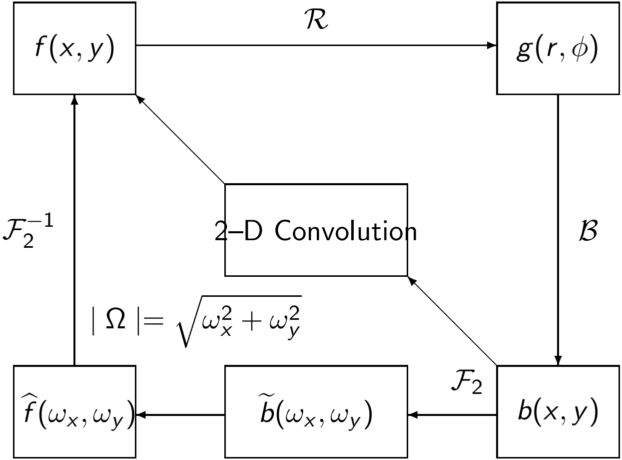
Inversion by Backprojection algorithms

$$f(x,y) = \mathcal{B}\left\{\mathcal{F}_1^{-1}\left\{|\Omega|\mathcal{F}_1\left\{g(r,\phi)\right\}\right\}\right\}$$



Inversion by Backprojection and filtering in 2D Fourier domaine

$$f(x,y) = \mathcal{F}_2^{-1}\left\{|\Omega|\mathcal{F}_2\left\{\mathcal{B}\left\{g(r,\phi)\right\}\right\}\right\}$$



$f(x,y)$

$\mathcal{R}$

$g(r,\phi)$

$\mathcal{F}_2^{-1}$

2–D Convolution

$\mathcal{B}$

$| \Omega |= \sqrt{\omega_x^2+\omega_y^2}$

$\mathcal{F}_2$

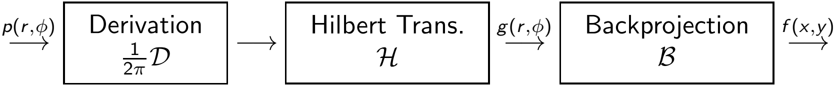
$\widehat{f}(\omega_x,\omega_y)$

$\widetilde{b}(\omega_x,\omega_y)$

$b(x,y)$

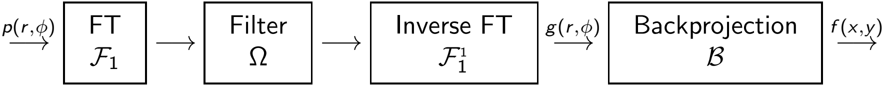
Inversion by Backprojection

- Direct Inversion of Radon Transform



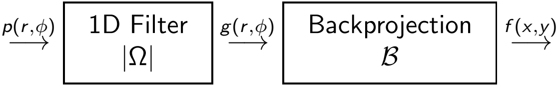
$\xrightarrow{p(r,\phi)}$ Derivation $\frac{1}{2\pi}\mathcal{D}$ $\longrightarrow$ Hilbert Trans. $\mathcal{H}$ $\xrightarrow{g(r,\phi)}$ Backprojection $\mathcal{B}$ $\xrightarrow{f(x,y)}$

- FBP:



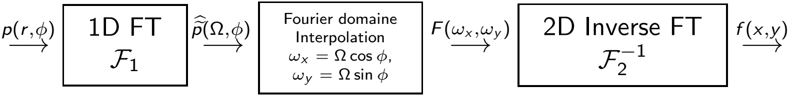
$\xrightarrow{p(r,\phi)}$ FT $\mathcal{F}_1$ $\longrightarrow$ Filter Ω $\longrightarrow$ Inverse FT $\mathcal{F}_1^1$ $\xrightarrow{g(r,\phi)}$ Backprojection $\mathcal{B}$ $\xrightarrow{f(x,y)}$

- Convolution–BackoProjection (CBP)



$\xrightarrow{p(r,\phi)}$ 1D Filter $|\Omega|$ $\xrightarrow{g(r,\phi)}$ Backprojection $\mathcal{B}$ $\xrightarrow{f(x,y)}$

- Fourier Domaine Inversion



$\xrightarrow{p(r,\phi)}$ 1D FT $\mathcal{F}_1$ $\xrightarrow{\widehat{p}(\Omega,\phi)}$ Fourier domaine Interpolation $\omega_x = \Omega\cos\phi$, $\omega_y = \Omega\sin\phi$ $\xrightarrow{F(\omega_x,\omega_y)}$ 2D Inverse FT $\mathcal{F}_2^{-1}$ $\xrightarrow{f(x,y)}$

Inversion by Backprojection

- Backprojection 2D Filtering

$$\xrightarrow{p(r,\phi)} \boxed{\begin{array}{c}\text{Backprojection}\\ \mathcal{B}\end{array}} \xrightarrow{b(x,y)} \boxed{\begin{array}{c}\text{2D FT}\\ \mathcal{F}_2\end{array}} \longrightarrow \boxed{\begin{array}{c}\text{2D Filter}\\ |\Omega|\end{array}} \longrightarrow \boxed{\begin{array}{c}\text{2D Inverse FT}\\ \mathcal{F}_2^{-1}\end{array}} \xrightarrow{f(x,y)}$$

- Backprojection and 2D Convolution

$$\xrightarrow{p(r,\phi)} \boxed{\begin{array}{c}\text{Backprojection}\\ \mathcal{B}\end{array}} \xrightarrow{b(x,y)} \longrightarrow \boxed{\begin{array}{c}\text{2D Filter}\\ |\Omega|\end{array}} \xrightarrow{f(x,y)}$$

Choice of the filter

$$H(\Omega) = |\Omega|$$

In practice:

$$H(\Omega) = |\Omega| W(\Omega)$$

- RAM–LAK
- Shepp–Logan
- Low pass Filter
- Hamming

Inversion par Backprojection: Implementation

convolution–Backprojection

$$\xrightarrow{p(r,\phi)} \boxed{\begin{array}{c}\text{Convolution}\\ h(r)\end{array}} \longrightarrow \xrightarrow{\overline{p}(r,\phi)} \boxed{\begin{array}{c}\text{Backprojection}\\ \mathcal{B}\end{array}} \xrightarrow{f(x,y)}$$

$$r = md, \quad -M/2 \leq m \leq M/2-1, \quad \phi = n\Delta$$

Discrete Convolution

$$\overline{p}(md, n\Delta) \simeq \overline{p}_n(m) \overset{\triangle}{=} \sum_{k=-M/2}^{M/2-1} p_n(k)h(m-k), \quad -M/2 \leq m \leq M/2-1$$

Linear Interpolation

$$\overline{p}(r, n\Delta) \simeq \overline{p}_n(m)+(r/d-m)[\overline{p}_n(m+1)-\overline{p}_n(m)], \; md \leq r \leq (m+1)d$$

Discrete Backprojection

$$f(x,y) \simeq \mathcal{B}_N\overline{p} \overset{\triangle}{=} \Delta \sum_{n=0}^{N-1} \overline{p}(r = x\cos n\Delta + y\sin n\Delta, n\Delta)$$

$$f(i\Delta x, j\Delta y) \simeq \Delta \sum_{n=0}^{N-1} \overline{p}(i\Delta x\cos n\Delta + j\Delta y\sin n\Delta, n\Delta)$$

Algebraic methods

- Discretization of the image

$$f(x,y) \simeq \sum_{i=1}^{I}\sum_{j=1}^{J} a_{i,j}\varphi_{i,j}(x,y)$$

- Discretization of the projections

$$p(r,\phi) \simeq \mathcal{R}f = \sum_{i=1}^{I}\sum_{j=1}^{J} a_{i,j}[\mathcal{R}\varphi_{i,j}] \overset{\triangle}{=} \sum_{i=1}^{I}\sum_{j=1}^{J} a_{i,j}h_{i,j}(r,\phi)$$

$$p(r_m,\phi_n) \simeq \sum_{i=1}^{I}\sum_{j=1}^{J} a_{i,j}h_{i,j}(r_m,\phi_n)$$

- Choice of the basis functions

$$\varphi_{i,j}(x,y) = \begin{cases} 1 & \text{inside the pixel} i,j \\ 0 & \text{elsewhere} \end{cases}$$

$$p(r_m,\phi_n) \simeq \sum_{i=1}^{I}\sum_{j=1}^{J} f_{i,j}h_{i,j}(r_m,\phi_n)$$

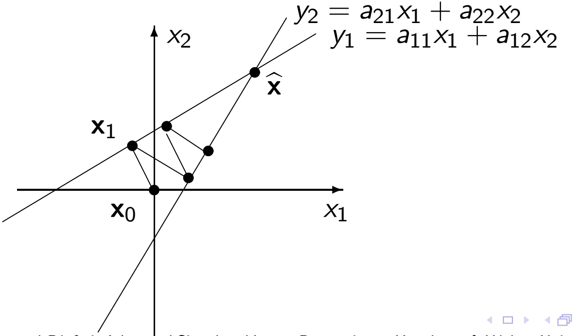
ART : Algebraic Reconstruction Techniques

Kaczmarz (Generalized inversion)

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{b}$$

$$\begin{cases} \mathbf{f}^0 = \mathbf{0} \\ \mathbf{f}_{(k+1)} = \mathbf{f}^{(k)} + \frac{g_i - \left\langle \mathbf{h}_i, \mathbf{f}^{(k)} \right\rangle}{\left\langle \mathbf{h}_i, \mathbf{h}_i \right\rangle} \mathbf{h}_i \quad i = 1, 2, \ldots, M \end{cases}$$

where $\mathbf{h}_i$ is the i-th ligne of the matrix $\mathbf{H}$.



ART : Algebraic Reconstruction Techniques

- At each iteration only one of equations (constraints) is used.
- One can also add other constraints. For example:

- without any constraint :

$$f_i(x) = x$$

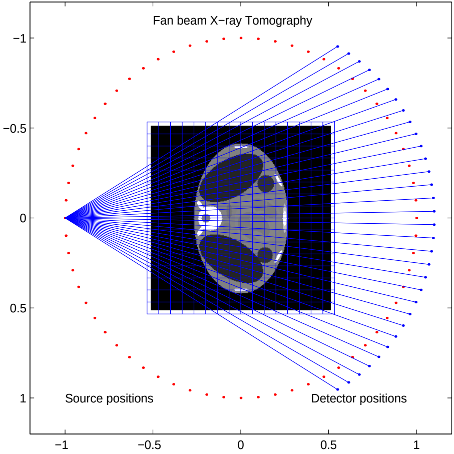
- Positivity :

$$f_i(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$

- Positivity and maximum value constraint:

$$f_i(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } 0 < x < 1 \\ 1 & \text{if } x > 1 \end{cases}$$

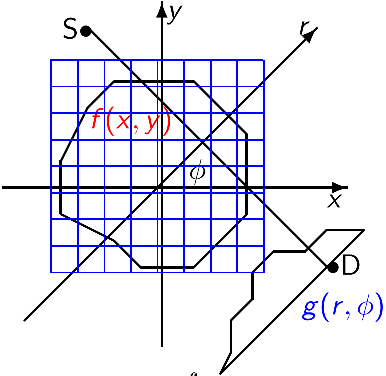
CT as a linear inverse problem

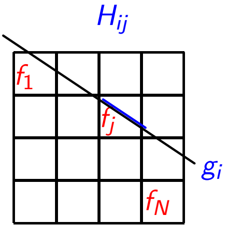


$$g(s_i) = \int_{L_i} f(\mathbf{r})\; \mathrm{d}l_i + \epsilon(s_i) \longrightarrow \text{Discretization} \longrightarrow \mathbf{g} = \mathbf{H}\mathbf{f} + \epsilon$$

- **g**, **f** and **H** are huge dimensional

Algebraic methods: Discretization





$$f(x,y) = \sum_j f_j \, b_j(x,y)$$

$$b_j(x,y) = \begin{cases} 1 & \text{if } (x,y) \in \text{ pixel } j \\ 0 & \text{else} \end{cases}$$

$$g(r,\phi) = \int_L f(x,y) \, \mathrm{d}l$$

$$g_i = \sum_{j=1}^{N} H_{ij} \, f_j + \epsilon_i$$

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \epsilon$$

Inversion: Deterministic methods

Data matching

- Observation model
  $g_i = h_i(\mathbf{f}) + \epsilon_i, \quad i = 1, \ldots, M \longrightarrow \mathbf{g} = \mathbf{H}(\mathbf{f}) + \boldsymbol{\epsilon}$
- Misatch between data and output of the model $\Delta(\mathbf{g}, \mathbf{H}(\mathbf{f}))$

$$\widehat{\mathbf{f}} = \arg\min_{\mathbf{f}} \{\Delta(\mathbf{g}, \mathbf{H}(\mathbf{f}))\}$$

- Examples:

– LS $$\Delta(\mathbf{g}, \mathbf{H}(\mathbf{f})) = \|\mathbf{g} - \mathbf{H}(\mathbf{f})\|^2 = \sum_i |g_i - h_i(\mathbf{f})|^2$$

– L_p $$\Delta(\mathbf{g}, \mathbf{H}(\mathbf{f})) = \|\mathbf{g} - \mathbf{H}(\mathbf{f})\|^p = \sum_i |g_i - h_i(\mathbf{f})|^p, \; 1 < p < 2$$

– KL $$\Delta(\mathbf{g}, \mathbf{H}(\mathbf{f})) = \sum_i g_i \ln \frac{g_i}{h_i(\mathbf{f})}$$

- In general, does not give satisfactory results for inverse problems.

Deterministic Inversion Algorithms

Least Squares Based Methods

$$\widehat{\mathbf{f}} = \arg\min_{\mathbf{f}} \{J(\mathbf{f})\} \quad \text{with} \quad J(\mathbf{f}) = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2$$

$$\nabla J(\mathbf{f}) = -2\mathbf{H}'(\mathbf{g} - \mathbf{H}\mathbf{f})$$

Gradient based algorithms:

- Initialize: $\mathbf{f}^{(0)}$
- Iterate: $\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} - \alpha \nabla J(\mathbf{f}^{(k)})$

At each iteration: $\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \alpha \mathbf{H}' \left(\mathbf{g} - \mathbf{H}\mathbf{f}^{(k)}\right)$

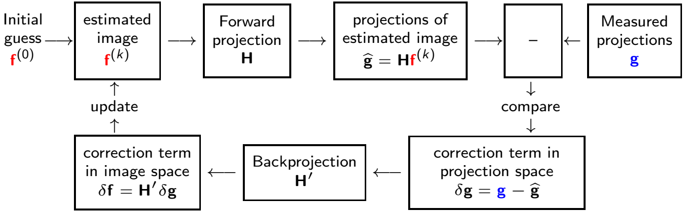
we have to do the following operations:

- Compute $\widehat{\mathbf{g}} = \mathbf{H}\mathbf{f}$ (Forward projection)
- Compute $\delta\mathbf{g} = \mathbf{g} - \widehat{\mathbf{g}}$ (Error or residual)
- Distribute $\delta\mathbf{f} = \mathbf{H}'\delta\mathbf{g}$ (Backprojection of error)
- Update $\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \delta\mathbf{f}$

Gradient based algorithms

Operations at each iteration: $\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \alpha \mathbf{H}' \left(\mathbf{g} - \mathbf{H}\mathbf{f}^{(k)} \right)$

- Compute $\widehat{\mathbf{g}} = \mathbf{H}\mathbf{f}$ (Forward projection)
- Compute $\delta\mathbf{g} = \mathbf{g} - \widehat{\mathbf{g}}$ (Error or residual)
- Distribute $\delta\mathbf{f} = \mathbf{H}'\delta\mathbf{g}$ (Backprojection of error)
- Update $\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \delta\mathbf{f}$



Initial guess $\mathbf{f}^{(0)}$ ⟶ estimated image $\mathbf{f}^{(k)}$ ⟶ Forward projection $\mathbf{H}$ ⟶ projections of estimated image $\widehat{\mathbf{g}} = \mathbf{H}\mathbf{f}^{(k)}$ ⟶ – ⟵ Measured projections $\mathbf{g}$

compare

correction term in image space $\delta\mathbf{f} = \mathbf{H}'\delta\mathbf{g}$ ⟵ Backprojection $\mathbf{H}'$ ⟵ correction term in projection space $\delta\mathbf{g} = \mathbf{g} - \widehat{\mathbf{g}}$

update

Gradient based algorithms

- Fixed step gradient:

$$\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \alpha \mathbf{H}' \left(\mathbf{g} - \mathbf{H}\mathbf{f}^{(k)} \right)$$

- Steepest descent gradient:

$$\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \alpha^{(k)} \mathbf{H}' \left(\mathbf{g} - \mathbf{H}\mathbf{f}^{(k)} \right)$$

  with $\alpha^{(k)} = \arg\min_{\alpha} \left\{ J(\mathbf{f} + \alpha \delta \mathbf{f}) \right\}$

- Conjugate Gradient

$$\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}$$

  The successive directions $\mathbf{d}^{(k)}$ have to be conjugate to each other.

Algebraic Reconstruction Techniques

- Main idea: Use the data as they arrive

$$\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \alpha^{(k)} [\mathbf{H}']_{i*} \left(g_i - [\mathbf{H}\mathbf{f}^{(k)}]_i \right)$$

which can also be written as:

$$\begin{aligned} \mathbf{f}^{(k+1)} &= \mathbf{f}^{(k)} + \frac{\left(g_i - [\mathbf{H}\mathbf{f}^{(k)}]_i \right)}{\mathbf{h}'_{i*}\mathbf{h}_{i*}} \mathbf{h}'_{i*} \\ &= \mathbf{f}^{(k)} + \sum_i \frac{\left(g_i - \sum_j H_{ij} f_j^{(k)} \right)}{\sum_i Hij^2} H_{ij} \end{aligned}$$

Algebraic Reconstruction Techniques

- Use the data as they arrive

$$\begin{aligned} \mathbf{f}^{(k+1)} &= \mathbf{f}^{(k)} + \frac{\left(g_i - [\mathbf{H}\mathbf{f}^{(k)}]_i\right)}{\mathbf{h}'_{i*}\mathbf{h}_{i*}}\mathbf{h}'_{i*} \\ &= \mathbf{f}^{(k)} + \sum_i \frac{\left(g_i - \sum_j H_{ij} f_j^{(k)}\right)}{\sum_i Hij^2} H_{ij} \end{aligned}$$

- Update each pixel at each time

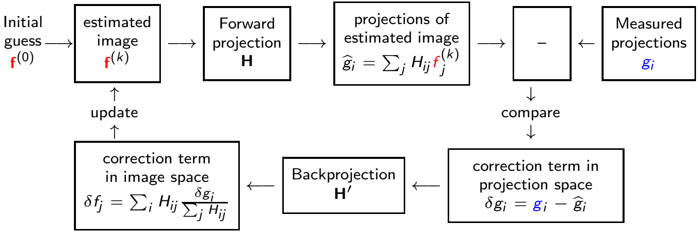
$$f_j^{(k+1)} = f_j^{(k)} + \frac{\left(g_i - \sum_j H_{ij} f_j^{(k)}\right)}{\sum_i Hij^2} H_{ij}$$

Algebraic Reconstruction Techniques (ART)

$$\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \sum_i \frac{\left(g_i - \sum_j H_{ij} f_j^{(k)}\right)}{\sum_i Hij^2} H_{ij}$$

or

$$f_j^{(k+1)} = f_j^{(k)} + \frac{\left(g_i - \sum_j H_{ij} f_j^{(k)}\right)}{\sum_i Hij^2} H_{ij}$$



Initial guess $\mathbf{f}^{(0)}$ ⟶

estimated image $\mathbf{f}^{(k)}$ ⟶

Forward projection $\mathbf{H}$ ⟶

projections of estimated image $\widehat{g}_i = \sum_j H_{ij} f_j^{(k)}$ ⟶

$-$ ⟵ Measured projections g_i

↓ compare ↓

correction term in projection space $\delta g_i = g_i - \widehat{g}_i$ ⟵

Backprojection $\mathbf{H}'$ ⟵

correction term in image space $\delta f_j = \sum_i H_{ij} \frac{\delta g_i}{\sum_j H_{ij}}$

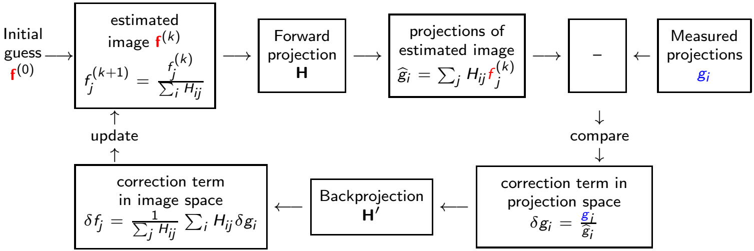
↑ update ↑

Algebraic Reconstruction using KL distance

- $\widehat{\mathbf{f}} = \arg\min_{\mathbf{f}} \{J(\mathbf{f})\}$ with $J(\mathbf{f}) = \sum_i g_i \ln \frac{g_i}{\sum_j H_{ij} f_j}$

$$f_j^{(k+1)} = \frac{f_j^{(k)}}{\sum_i H_{ij}} \sum_i H_{ij} \frac{g_i}{\sum_j H_{ij} f_j^{(k)}}$$

Interestingly, this is the OSEM (Ordered subset Expectation-Maximization) algorithm which is based on Maximum Likelihood and proposed first by Shepp & Vardi.



Initial guess $\mathbf{f}^{(0)}$

estimated image $\mathbf{f}^{(k)}$

$f_j^{(k+1)} = \frac{f_j^{(k)}}{\sum_i H_{ij}}$

Forward projection **H**

projections of estimated image

$\widehat{g}_i = \sum_j H_{ij} f_j^{(k)}$

–

Measured projections g_i

compare

correction term in projection space

$\delta g_i = \frac{g_i}{\widehat{g}_i}$

Backprojection $\mathbf{H}'$

correction term in image space

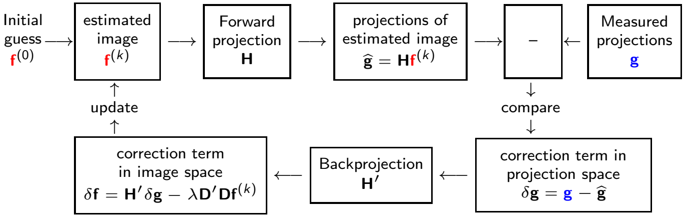
$\delta f_j = \frac{1}{\sum_j H_{ij}} \sum_i H_{ij} \delta g_i$

update

Gradient based algorithms

$$\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + \alpha \left[\mathbf{H}' \left(\mathbf{g} - \mathbf{H}\mathbf{f}^{(k)} \right) - \lambda \mathbf{D}'\mathbf{D}\mathbf{f}^{(k)} \right]$$

- Compute $\widehat{\mathbf{g}} = \mathbf{H}\mathbf{f}$ (Forward projection)
- Compute $\delta\mathbf{g} = \mathbf{g} - \widehat{\mathbf{g}}$ (Error or residual)
- Compute $\delta\mathbf{f}_1 = \mathbf{H}'\delta\mathbf{g}$ (Backprojection of error)
- Compute $\delta\mathbf{f}_2 = -\mathbf{D}'\mathbf{D}\mathbf{f}$ (Correction due to regularization)
- Update $\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + [\delta\mathbf{f}_1 + \delta\mathbf{f}_2]$



Initial guess $\mathbf{f}^{(0)}$ ⟶

estimated image $\mathbf{f}^{(k)}$ ⟶

Forward projection $\mathbf{H}$ ⟶

projections of estimated image $\widehat{\mathbf{g}} = \mathbf{H}\mathbf{f}^{(k)}$ ⟶

$-$ ⟵

Measured projections $\mathbf{g}$

↓ compare ↓

correction term in projection space $\delta\mathbf{g} = \mathbf{g} - \widehat{\mathbf{g}}$ ⟵

Backprojection $\mathbf{H}'$ ⟵

correction term in image space $\delta\mathbf{f} = \mathbf{H}'\delta\mathbf{g} - \lambda\mathbf{D}'\mathbf{D}\mathbf{f}^{(k)}$

↑ update ↑

Content

1. Introduction: Signals and Images, Linear transformations (Convolution, Fourier, Laplace, Hilbert, Radon, ..., Discrete convolution, Z transform, DFT, FFT, ...)
2. Modeling: parametric and non-parametric, MA, AR and ARMA models, Linear prediction
3. Deconvolution and Parameter Estimation: Deterministic (LS, Regularization) and Probabilistic methods (Wiener Filtering)
4. Inverse problems and Bayesian estimation
5. Kalman Filtering and smoothing
6. Case study: Signal deconvolution
7. Case study: Image restoration
8. Case study: Image reconstruction and Computed Tomography

7. Case study: Image restoration

- Image restoration
- Classical methods: Wiener filtering in 2D
- Bayesian approach to deconvolution
- Simple and Blind Deconvolution
- Practical issues and computational costs

Full Bayesian approach

$$\mathcal{M}: \qquad \mathbf{g} = \mathbf{H}\mathbf{f} + \epsilon$$

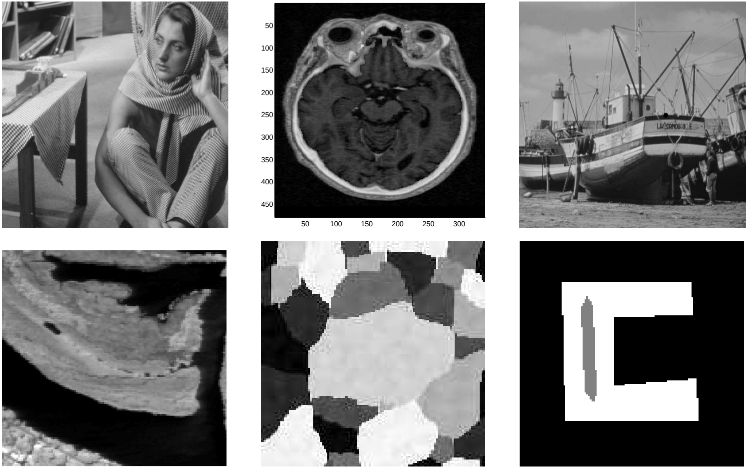
- Forward & errors model: $\longrightarrow p(\mathbf{g}|\mathbf{f}, \boldsymbol{\theta}_1; \mathcal{M})$
- Prior models $\longrightarrow p(\mathbf{f}|\boldsymbol{\theta}_2; \mathcal{M})$
- Hyperparameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \longrightarrow p(\boldsymbol{\theta}|\mathcal{M})$
- Bayes: $\longrightarrow p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g}; \mathcal{M}) = \frac{p(\mathbf{g}|\mathbf{f},\boldsymbol{\theta};\mathcal{M})\; p(\mathbf{f}|\boldsymbol{\theta};\mathcal{M})\; p(\boldsymbol{\theta}|\mathcal{M})}{p(\mathbf{g}|\mathcal{M})}$
- Joint MAP: $\qquad (\widehat{\mathbf{f}}, \widehat{\boldsymbol{\theta}}) = \arg\max_{(\mathbf{f},\boldsymbol{\theta})} \{p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g}; \mathcal{M})\}$
- Marginalization: $\left\{ \begin{array}{ll} p(\mathbf{f}|\mathbf{g}; \mathcal{M}) & = \int p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g}; \mathcal{M})\; \mathrm{d}\mathbf{f} \\ p(\boldsymbol{\theta}|\mathbf{g}; \mathcal{M}) & = \int p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g}; \mathcal{M})\; \mathrm{d}\boldsymbol{\theta} \end{array} \right.$
- Posterior means: $\left\{ \begin{array}{ll} \widehat{\mathbf{f}} & = \int \mathbf{f}\; p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g}; \mathcal{M})\; \mathrm{d}\mathbf{f}\; \mathrm{d}\boldsymbol{\theta} \\ \widehat{\boldsymbol{\theta}} & = \int \boldsymbol{\theta}\; p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g}; \mathcal{M})\; \mathrm{d}\mathbf{f}\; \mathrm{d}\boldsymbol{\theta} \end{array} \right.$
- Evidence of the model:

$$p(\mathbf{g}|\mathcal{M}) = \iint p(\mathbf{g}|\mathbf{f}, \boldsymbol{\theta}; \mathcal{M}) p(\mathbf{f}|\boldsymbol{\theta}; \mathcal{M}) p(\boldsymbol{\theta}|\mathcal{M})\; \mathrm{d}\mathbf{f}\; \mathrm{d}\boldsymbol{\theta}$$

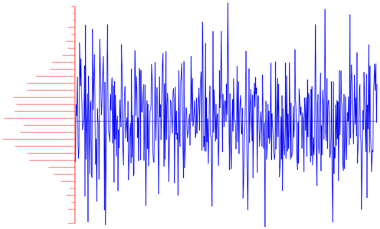
Two main steps in the Bayesian approach

- Prior modeling
  - Separable:
    Gaussian, Generalized Gaussian, Gamma,
    mixture of Gaussians, mixture of Gammas, ...
  - Markovian: Gauss-Markov, GGM, ...
  - Separable or Markovian with hidden variables (contours, region labels)
- Choice of the estimator and computational aspects
  - MAP, Posterior mean, Marginal MAP
  - MAP needs optimization algorithms
  - Posterior mean needs integration methods
  - Marginal MAP needs integration and optimization
  - Approximations:
    - Gaussian approximation (Laplace)
    - Numerical exploration MCMC
    - Variational Bayes (Separable approximation)

Which images I am looking for?

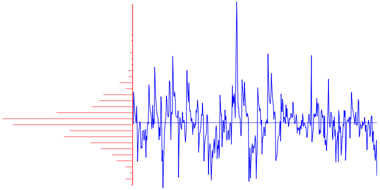


Which signals I am looking for?



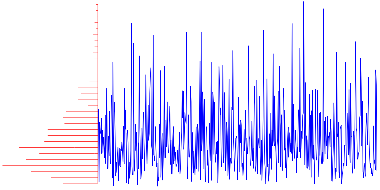
Gaussian

$$p(f_j) \propto \exp\left\{-\alpha|f_j|^2\right\}$$



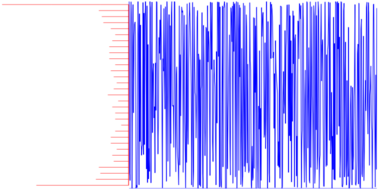
Generalized Gaussian

$$p(f_j) \propto \exp\left\{-\alpha|f_j|^p\right\}, \quad 1 \leq p \leq 2$$



Gamma

$$p(f_j) \propto f_j^\alpha \exp\left\{-\beta f_j\right\}$$



Beta

$$p(f_j) \propto f_j^\alpha (1-f_j)^\beta$$

Different prior models for signals and images

- Separable $p(\mathbf{f}) = \prod_j p_j(f_j) \propto \exp\left\{-\beta \sum_j \phi(f_j)\right\}$

$$p(\mathbf{f}) \propto \exp\left\{-\beta \sum_{\mathbf{r}\in\mathcal{R}} \phi(f(\mathbf{r}))\right\}$$

- Markoviens (simple) $p(f_j|f_{j-1}) \propto \exp\left\{-\beta\phi(f_j - f_{j-1})\right\}$

$$p(\mathbf{f}) \propto \exp\left\{-\beta \sum_{\mathbf{r}\in\mathcal{R}} \sum_{\mathbf{r}'\in\mathcal{V}(\mathbf{r})} \phi(f(\mathbf{r}), f(\mathbf{r}'))\right\}$$

- Markovien with hidden variables
  $z(\mathbf{r})$ (lines, contours, regions)

$$p(\mathbf{f}|\mathbf{z}) \propto \exp\left\{-\beta \sum_{\mathbf{r}\in\mathcal{R}} \sum_{\mathbf{r}'\in\mathcal{V}(\mathbf{r})} \phi(f(\mathbf{r}), f(\mathbf{r}'), z(\mathbf{r}), z(\mathbf{r}'))\right\}$$

Different prior models for images: Separable

- Gaussian:

$$p(f_j) \propto \exp\left\{-\alpha|f_j|^2\right\} \longrightarrow \quad \Omega(\mathbf{f}) = \alpha \sum_j |f_j|^2$$

- Generalized Gaussian (GG):

$$p(f_j) \propto \exp\left\{-\alpha|f_j|^p\right\}, \quad 1 \leq p \leq 2 \longrightarrow \quad \Phi(\mathbf{f}) = \alpha \sum_j |f_j|^p,$$

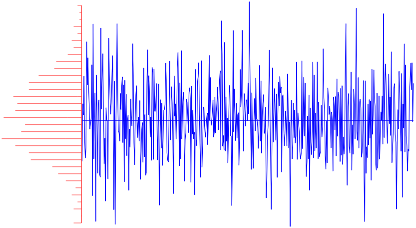
- Gamma: $f_j > 0$

$$p(f_j) \propto f_j^\alpha \exp\left\{-\beta f_j\right\} \longrightarrow \quad \Omega(\mathbf{f}) = \alpha \sum_j \ln f_j + \beta \sum_j f_j,$$

- Beta: $1 > f_j > 0$

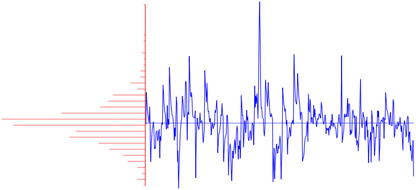
$$p(f_j) \propto f_j^\alpha (1-f_j)^\beta \longrightarrow \quad \Omega(\mathbf{f}) = \alpha \sum_j \ln f_j + \beta \sum_j \ln(1-f_j),$$

Different prior models for images: Separable



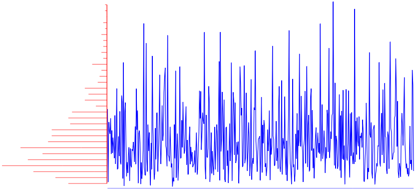
Gaussian

$$p(f_j) \propto \exp\left\{-\alpha|f_j|^2\right\}$$



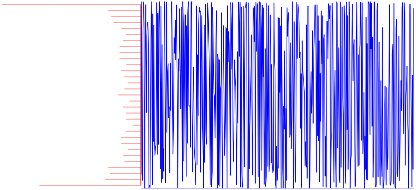
Generalized Gaussian

$$p(f_j) \propto \exp\left\{-\alpha|f_j|^p\right\}, \quad 1 \leq p \leq 2$$



Gamma

$$p(f_j) \propto f_j^\alpha \exp\left\{-\beta f_j\right\}$$



Beta

$$p(f_j) \propto f_j^\alpha (1-f_j)^\beta$$

Different prior models: Simple Markovian

$$p(f_j|\mathbf{f}) \propto \exp\left\{-\alpha \sum_{i\in v_j} \phi(f_j, f_i)\right\} \longrightarrow \quad \Phi(\mathbf{f}) = \alpha \sum_j \sum_{i\in V_j} \phi(f_j, f_i)$$

- 1D case and one neigbor $V_j = j-1$:

$$\Phi(\mathbf{f}) = \alpha \sum_j \phi(f_j - f_{j-1})$$

- 1D Case and two neighbors $V_j = \{j-1, j+1\}$:

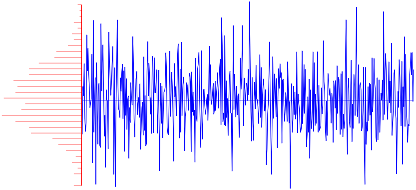
$$\Phi(\mathbf{f}) = \alpha \sum_j \phi\left(f_j - \beta(f_{j-1} + f_{j-1})\right)$$

- 2D case with 4 neighbors:

$$\Phi(\mathbf{f}) = \alpha \sum_{\mathbf{r}\in\mathcal{R}} \phi\left(f(\mathbf{r}) - \beta \sum_{\mathbf{r}'\in\mathcal{V}(\mathbf{r})} f(\mathbf{r}')\right)$$

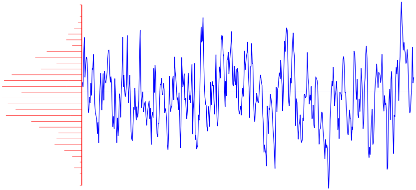
- $\phi(t) = |t|^\gamma$: Generalized Gaussian

Different prior models: Simple Markovian



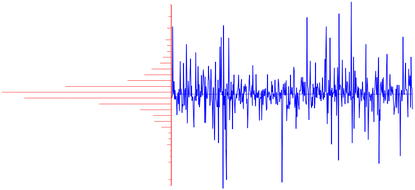
IID Gaussian

$p(f_j) \propto \exp\left\{-\alpha|f_j|^2\right\}$



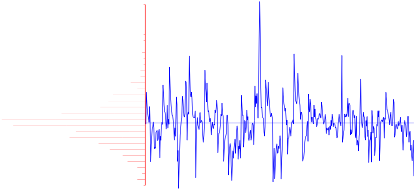
Gauss-Markov

$p(f_j|f_{j-1}) \propto \exp\left\{-\alpha|f_j - f_{j-1}|^2\right\}$



IID GG

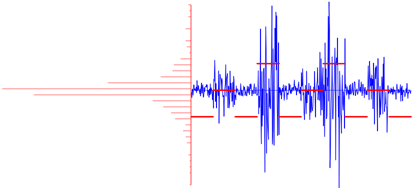
$p(f_j) \propto \exp\left\{-\alpha|f_j|^p\right\}$



Markovian GG

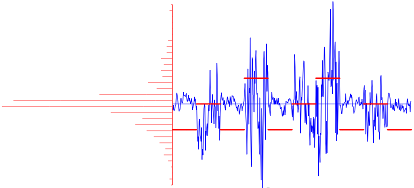
$p(f_j|f_{j-1}) \propto \exp\left\{-\alpha|f_j - f_{j-1}|^p\right\}$

Different prior models: Non-stationnary signals



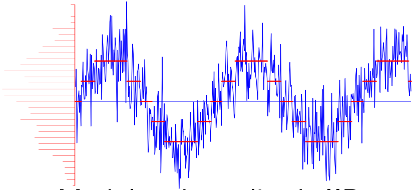
Modulated Variances IID

$p(f_j|z_j) = \mathcal{N}(0, v(z_j))$



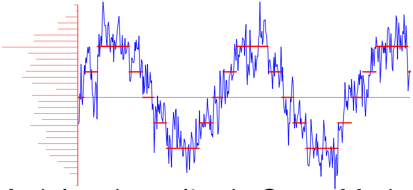
Modulated Variances Gauss-Markov

$p(f_j|f_{j-1}, z_j) = \mathcal{N}(f_{j-1}, v(z_j))$



Modulated amplituds IID

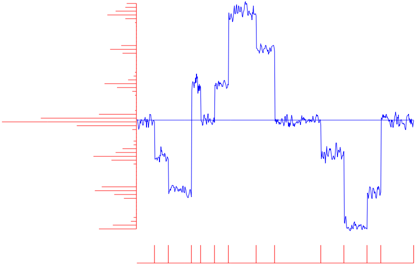
$p(f_j|z_j) = \mathcal{N}(a(z_j), 1)$



Modulated amplituds Gauss-Markov

$p(f_j|f_{j-1}, z_j) = \mathcal{N}(a(f_{j-1}, z_j), 1)$

Different prior models: Markovian with hidden variables

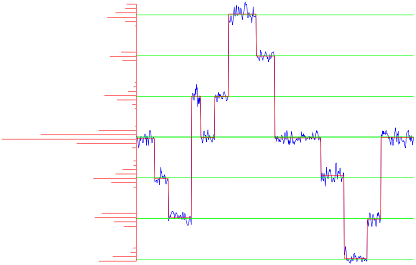


Piecewise Gaussians

(contours hidden variables)

$$p(f_j|q_j,f_{j-1}) = \mathcal{N}\left((1-q_j)f_{j-1},\sigma_f^2\right)$$

$$p(\mathbf{f}|\mathbf{q}) \propto \exp\left\{-\alpha\sum_j \left|f_j-(1-q_j)f_{j-1}\right|^2\right\}$$



Mixture of Gaussians (MoG)

(regions labels hidden variables)

$$p(f_j|z_j=k) = \mathcal{N}\left(m_k,\sigma_k^2\right) \text{ \& } z_j \text{ markovian}$$

$$p(\mathbf{f}|\mathbf{z}) \propto \exp\left\{-\alpha\sum_k\sum_{j\in\mathcal{R}_k}\left(\frac{f_j-m_k}{\sigma_k}\right)^2\right\}$$

Particular case of Gauss-Markov models

$$\left\{\begin{array}{l} \mathbf{g}=\mathbf{H}\mathbf{f}+\boldsymbol{\epsilon} \quad \text{with} \\ \mathbf{f}\sim\mathcal{N}\left(0,\sigma_f^2(\mathbf{D}'\mathbf{D})^{-1}\right) \end{array}\right. = \left\{\begin{array}{l} \mathbf{g}=\mathbf{H}\mathbf{f}+\boldsymbol{\epsilon} \\ \mathbf{f}=\mathbf{C}\mathbf{f}+\mathbf{z} \text{ with } \mathbf{z}\sim\mathcal{N}(0,\sigma_f^2\mathbf{I}) \\ \text{and} \quad \mathbf{D}=(\mathbf{I}-\mathbf{C}) \end{array}\right.$$

$$\mathbf{f}|\mathbf{g}\sim\mathcal{N}(\widehat{\mathbf{f}},\widehat{\mathbf{P}}) \quad \text{with} \quad \widehat{\mathbf{f}}=\widehat{\mathbf{P}}\mathbf{H}'\mathbf{g}, \quad \widehat{\mathbf{P}}=\left(\mathbf{H}'\mathbf{H}+\lambda\mathbf{D}'\mathbf{D}\right)^{-1}$$

$$\widehat{\mathbf{f}}=\arg\min_{\mathbf{f}}\left\{J(\mathbf{f})=\|\mathbf{g}-\mathbf{H}\mathbf{f}\|^2+\lambda\|\mathbf{D}\mathbf{f}\|^2\right\}$$

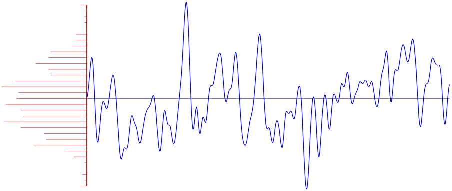
$$\left\{\begin{array}{l} \mathbf{g}=\mathbf{H}\mathbf{f}+\boldsymbol{\epsilon} \\ \text{with} \quad \mathbf{f}\sim\mathcal{N}\left(0,\sigma_f^2(\mathbf{D}\mathbf{D}')\right) \end{array}\right. = \left\{\begin{array}{l} \mathbf{g}=\mathbf{H}\mathbf{f}+\boldsymbol{\epsilon} \\ \mathbf{f}=\mathbf{D}\mathbf{z} \quad \text{with} \quad \mathbf{z}\sim\mathcal{N}(0,\sigma_f^2\mathbf{I}) \end{array}\right.$$

$$\mathbf{z}|\mathbf{g}\sim\mathcal{N}(\widehat{\mathbf{z}},\widehat{\mathbf{P}}) \quad \text{with} \quad \widehat{\mathbf{z}}=\widehat{\mathbf{P}}\mathbf{D}'\mathbf{H}'\mathbf{g}, \quad \widehat{\mathbf{P}}=\left(\mathbf{D}'\mathbf{H}'\mathbf{H}\mathbf{D}+\lambda\mathbf{I}\right)^{-1}$$

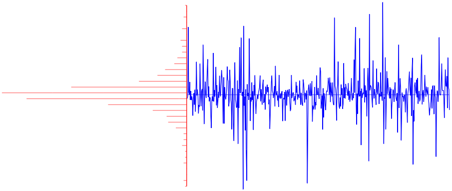
$$\widehat{\mathbf{z}}=\arg\min_{\mathbf{z}}\left\{J(\mathbf{z})=\|\mathbf{g}-\mathbf{H}\mathbf{D}\mathbf{z}\|^2+\lambda\|\mathbf{z}\|^2\right\}\longrightarrow\widehat{\mathbf{f}}=\mathbf{D}\widehat{\mathbf{z}}$$

$\mathbf{z}$ Decomposition coeff on a basis (column of $\mathbf{D}$)

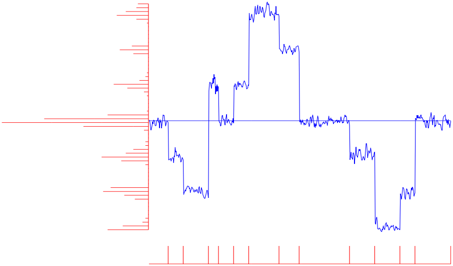
Which images I am looking for?



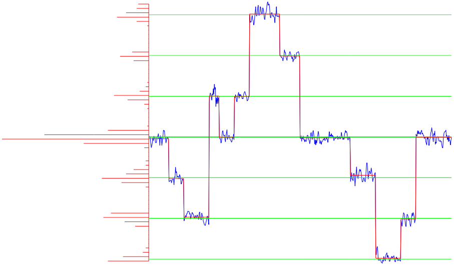
Gauss-Markov



Generalized GM



Piecewize Gaussian



Mixture of GM

Markovien prior models for images

$$\Omega(\mathbf{f}) = \sum_j \phi(f_j - f_{j-1})$$

- Gauss-Markov : $\phi(t) = |t|^2$
- Generalized Gauss-Markov : $\phi(t) = |t|^\alpha$
- Picewize Gauss-Markov or GGM : $\phi(t) = \begin{cases} t^2 & |t| \leq T \\ T^2 & |t| > T \end{cases}$

  or equivalently :

$$\Omega(\mathbf{f}|\mathbf{q}) = \sum_j (1 - q_j)\phi(f_j - f_{j-1})$$

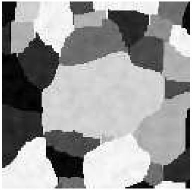
  $\mathbf{q}$ line process (contours)

- Mixture of Gaussians :

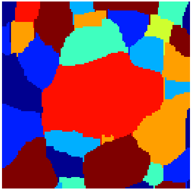
$$\Omega(\mathbf{f}|\mathbf{z}) = \sum_k \sum_{\{j:z_j=k\}} \left(\frac{f_j - m_k}{v_k}\right)^2$$

  $\mathbf{z}$ region labels process.

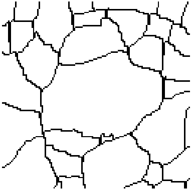
Gauss-Markov-Potts prior models for images



$f(\mathbf{r})$



$z(\mathbf{r})$



$c(\mathbf{r}) = 1 - \delta(z(\mathbf{r}) - z(\mathbf{r}'))$

$$p(f(\mathbf{r})|z(\mathbf{r}) = k, m_k, v_k) = \mathcal{N}(m_k, v_k)$$

$$p(f(\mathbf{r})) = \sum_k P(z(\mathbf{r}) = k)\,\mathcal{N}(m_k, v_k) \quad \text{Mixture of Gaussians}$$

- Separable iid hidden variables: $p(\mathbf{z}) = \prod_{\mathbf{r}} p(z(\mathbf{r}))$
- Markovian hidden variables: $p(\mathbf{z})$ Potts-Markov:

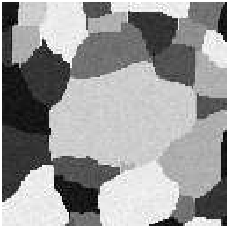
$$p(z(\mathbf{r})|z(\mathbf{r}'), \mathbf{r}' \in \mathcal{V}(\mathbf{r})) \propto \exp\left\{\gamma \sum_{\mathbf{r}'\in\mathcal{V}(\mathbf{r})} \delta(z(\mathbf{r}) - z(\mathbf{r}'))\right\}$$

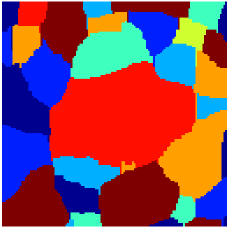
$$p(\mathbf{z}) \propto \exp\left\{\gamma \sum_{\mathbf{r}\in\mathcal{R}} \sum_{\mathbf{r}'\in\mathcal{V}(\mathbf{r})} \delta(z(\mathbf{r}) - z(\mathbf{r}'))\right\}$$

Four different cases

To each pixel of the image is associated 2 variables $f(\mathbf{r})$ and $z(\mathbf{r})$

- $\mathbf{f}|\mathbf{z}$ Gaussian iid, $\mathbf{z}$ iid :
  Mixture of Gaussians
- $\mathbf{f}|\mathbf{z}$ Gauss-Markov, $\mathbf{z}$ iid :
  Mixture of Gauss-Markov
- $\mathbf{f}|\mathbf{z}$ Gaussian iid, $\mathbf{z}$ Potts-Markov :
  Mixture of Independent Gaussians
  (MIG with Hidden Potts)
- $\mathbf{f}|\mathbf{z}$ Markov, $\mathbf{z}$ Potts-Markov :
  Mixture of Gauss-Markov
  (MGM with hidden Potts)


$f(\mathbf{r})$


$z(\mathbf{r})$

Case 1: $\mathbf{f}|\mathbf{z}$ Gaussian iid, $\mathbf{z}$ iid

Independent Mixture of Independent Gaussiens (IMIG):

$$p(f(\mathbf{r})|z(\mathbf{r})=k)=\mathcal{N}(m_k,v_k),\quad \forall\mathbf{r}\in\mathcal{R}$$
$$p(f(\mathbf{r}))=\sum_{k=1}^K \alpha_k\mathcal{N}(m_k,v_k), \text{with } \sum_k \alpha_k=1.$$

$$p(\mathbf{z})=\prod_{\mathbf{r}} p(z(\mathbf{r})=k)=\prod_{\mathbf{r}}\alpha_k=\prod_k \alpha_k^{n_k}$$

Noting

$$m_z(\mathbf{r})=m_k, v_z(\mathbf{r})=v_k, \alpha_z(\mathbf{r})=\alpha_k, \forall\mathbf{r}\in\mathcal{R}_k$$

we have:

$$\begin{aligned} p(\mathbf{f}|\mathbf{z}) &= \prod_{\mathbf{r}\in\mathcal{R}} \mathcal{N}(m_z(\mathbf{r}), v_z(\mathbf{r})) \\ p(\mathbf{z}) &= \prod_{\mathbf{r}} \alpha_z(\mathbf{r}) = \prod_k \alpha_k^{\sum_{\mathbf{r}\in\mathcal{R}}\delta(z(\mathbf{r})-k)} = \prod_k \alpha_k^{n_k} \end{aligned}$$

Case 2: $\mathbf{f}|\mathbf{z}$ Gauss-Markov, $\mathbf{z}$ iid

Independent Mixture of Gauss-Markov (IMGM):

$$p(f(\mathbf{r})|z(\mathbf{r}), z(\mathbf{r}'), f(\mathbf{r}'), \mathbf{r}' \in \mathcal{V}(\mathbf{r})) = \mathcal{N}(\mu_z(\mathbf{r}), v_z(\mathbf{r})), \forall \mathbf{r} \in \mathcal{R}$$

$$\begin{array}{ll} \mu_z(\mathbf{r}) & = \frac{1}{|\mathcal{V}(\mathbf{r})|} \sum_{\mathbf{r}' \in \mathcal{V}(\mathbf{r})} \mu_z^*(\mathbf{r}') \\ \mu_z^*(\mathbf{r}') & = \delta(z(\mathbf{r}') - z(\mathbf{r}))\ f(\mathbf{r}') + (1 - \delta(z(\mathbf{r}') - z(\mathbf{r}))\ m_z(\mathbf{r}') \\ & = \quad (1 - c(\mathbf{r}')) \quad f(\mathbf{r}')\ + \qquad\quad c(\mathbf{r}') \qquad\qquad m_z(\mathbf{r}') \end{array}$$

$$\begin{array}{lll} p(\mathbf{f}|\mathbf{z}) & \propto \prod_{\mathbf{r}} \mathcal{N}(\mu_z(\mathbf{r}), v_z(\mathbf{r})) & \propto \prod_k \alpha_k \mathcal{N}(m_k 1, \mathbf{\Sigma}_k) \\ p(\mathbf{z}) & = \prod_{\mathbf{r}} v_z(\mathbf{r}) & = \prod_k \alpha_k^{n_k} \end{array}$$

with $1_k = 1, \forall \mathbf{r} \in \mathcal{R}_k$ and $\mathbf{\Sigma}_k$ a covariance matrix $(n_k \times n_k)$.

Case 3: $\mathbf{f}|\mathbf{z}$ Gauss iid, $\mathbf{z}$ Potts

Gauss iid as in Case 1:

$$p(\mathbf{f}|\mathbf{z}) = \prod_{\mathbf{r}\in\mathcal{R}} \mathcal{N}(m_z(\mathbf{r}), v_z(\mathbf{r})) = \prod_k \prod_{\mathbf{r}\in\mathcal{R}_k} \mathcal{N}(m_k, v_k)$$

Potts-Markov

$$p(z(\mathbf{r})|z(\mathbf{r}'), \mathbf{r}' \in \mathcal{V}(\mathbf{r})) \propto \exp\left\{\gamma \sum_{\mathbf{r}'\in\mathcal{V}(\mathbf{r})} \delta(z(\mathbf{r}) - z(\mathbf{r}'))\right\}$$

$$p(\mathbf{z}) \propto \exp\left\{\gamma \sum_{\mathbf{r}\in\mathcal{R}} \sum_{\mathbf{r}'\in\mathcal{V}(\mathbf{r})} \delta(z(\mathbf{r}) - z(\mathbf{r}'))\right\}$$

Case 4: $\mathbf{f}|\mathbf{z}$ Gauss-Markov, $\mathbf{z}$ Potts

Gauss-Markov as in Case 2:

$$p(f(\mathbf{r})|z(\mathbf{r}), z(\mathbf{r}'), f(\mathbf{r}'), \mathbf{r}' \in \mathcal{V}(\mathbf{r})) = \mathcal{N}(\mu_z(\mathbf{r}), v_z(\mathbf{r})), \forall \mathbf{r} \in \mathcal{R}$$

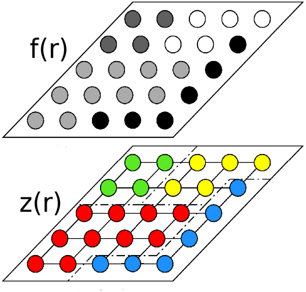
$$\begin{array}{ll} \mu_z(\mathbf{r}) & = \frac{1}{|\mathcal{V}(\mathbf{r})|} \sum_{\mathbf{r}' \in \mathcal{V}(\mathbf{r})} \mu_z^*(\mathbf{r}') \\ \mu_z^*(\mathbf{r}') & = \delta(z(\mathbf{r}') - z(\mathbf{r}))\; f(\mathbf{r}') + (1 - \delta(z(\mathbf{r}') - z(\mathbf{r}))\; m_z(\mathbf{r}') \end{array}$$

$$p(\mathbf{f}|\mathbf{z}) \quad \propto \prod_{\mathbf{r}} \mathcal{N}(\mu_z(\mathbf{r}), v_z(\mathbf{r})) \propto \prod_k \alpha_k \mathcal{N}(m_k 1, \mathbf{\Sigma}_k)$$

Potts-Markov as in Case 3:

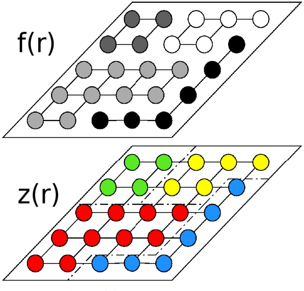
$$p(\mathbf{z}) \propto \exp\left\{ \gamma \sum_{\mathbf{r} \in \mathcal{R}} \sum_{\mathbf{r}' \in \mathcal{V}(\mathbf{r})} \delta(z(\mathbf{r}) - z(\mathbf{r}')) \right\}$$

Summary of the two proposed models



f|**z** Gaussian iid
z Potts-Markov

(MIG with Hidden Potts)



f|**z** Markov
z Potts-Markov

(MGM with hidden Potts)

Bayesian Computation

$$p(\mathbf{f},\mathbf{z},\boldsymbol{\theta}|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f},\mathbf{z},v_\epsilon)\, p(\mathbf{f}|\mathbf{z},\mathbf{m},\mathbf{v})\, p(\mathbf{z}|\gamma,\boldsymbol{\alpha})\, p(\boldsymbol{\theta})$$

$\boldsymbol{\theta} = \{v_\epsilon, (\alpha_k, m_k, v_k), k = 1, \cdot, K\}$ $\quad p(\boldsymbol{\theta})$ Conjugate priors

- Direct computation and use of $p(\mathbf{f},\mathbf{z},\boldsymbol{\theta}|\mathbf{g};\mathcal{M})$ is too complex
- Possible approximations :
  - Gauss-Laplace (Gaussian approximation)
  - Exploration (Sampling) using MCMC methods
  - Separable approximation (Variational techniques)
- Main idea in Variational Bayesian methods:
  Approximate
  $$p(\mathbf{f},\mathbf{z},\boldsymbol{\theta}|\mathbf{g};\mathcal{M}) \quad \text{by} \quad q(\mathbf{f},\mathbf{z},\boldsymbol{\theta}) = q_1(\mathbf{f})\; q_2(\mathbf{z})\; q_3(\boldsymbol{\theta})$$
  - Choice of approximation criterion : $KL(q : p)$
  - Choice of appropriate families of probability laws for $q_1(\mathbf{f})$, $q_2(\mathbf{z})$ and $q_3(\boldsymbol{\theta})$

MCMC based algorithm

$$p(\mathbf{f},\mathbf{z},\boldsymbol{\theta}|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f},\mathbf{z},\boldsymbol{\theta})\, p(\mathbf{f}|\mathbf{z},\boldsymbol{\theta})\, p(\mathbf{z})\, p(\boldsymbol{\theta})$$

General scheme:

$$\widehat{\mathbf{f}} \sim p(\mathbf{f}|\widehat{\mathbf{z}},\widehat{\boldsymbol{\theta}},\mathbf{g}) \longrightarrow \widehat{\mathbf{z}} \sim p(\mathbf{z}|\widehat{\mathbf{f}},\widehat{\boldsymbol{\theta}},\mathbf{g}) \longrightarrow \widehat{\boldsymbol{\theta}} \sim (\boldsymbol{\theta}|\widehat{\mathbf{f}},\widehat{\mathbf{z}},\mathbf{g})$$

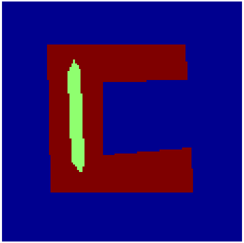
- Estimate $\mathbf{f}$ using $p(\mathbf{f}|\widehat{\mathbf{z}},\widehat{\boldsymbol{\theta}},\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f},\boldsymbol{\theta})\, p(\mathbf{f}|\widehat{\mathbf{z}},\widehat{\boldsymbol{\theta}})$
  Needs optimisation of a quadratic criterion.

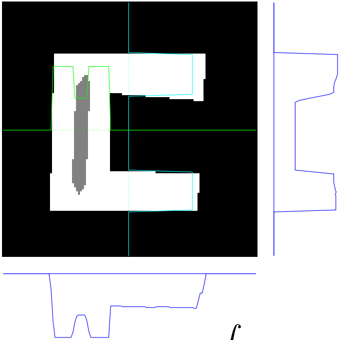
- Estimate $\mathbf{z}$ using $p(\mathbf{z}|\widehat{\mathbf{f}},\widehat{\boldsymbol{\theta}},\mathbf{g}) \propto p(\mathbf{g}|\widehat{\mathbf{f}},\widehat{\mathbf{z}},\widehat{\boldsymbol{\theta}})\, p(\mathbf{z})$
  Needs sampling of a Potts Markov field.

- Estimate $\boldsymbol{\theta}$ using
  $p(\boldsymbol{\theta}|\widehat{\mathbf{f}},\widehat{\mathbf{z}},\mathbf{g}) \propto p(\mathbf{g}|\widehat{\mathbf{f}},\sigma_\epsilon^2\mathbf{I})\, p(\widehat{\mathbf{f}}|\widehat{\mathbf{z}},(m_k,v_k))\, p(\boldsymbol{\theta})$
  Conjugate priors ⟶ analytical expressions.

Application of CT in NDT

Reconstruction from only 2 projections



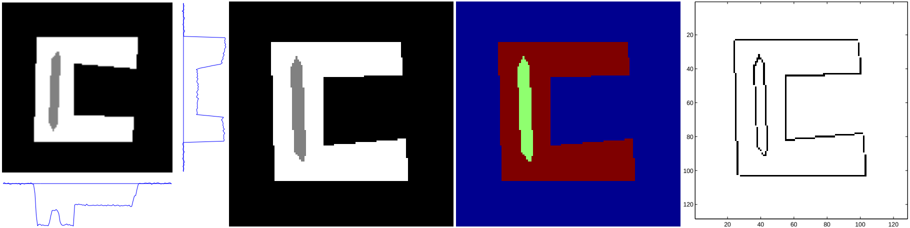


$$g_1(x) = \int f(x,y)\,\mathrm{d}y, \qquad g_2(y) = \int f(x,y)\,\mathrm{d}x$$

- Given the marginals $g_1(x)$ and $g_2(y)$ find the joint distribution $f(x,y)$.
- Infinite number of solutions : $f(x,y) = g_1(x)\,g_2(y)\,\Omega(x,y)$
  $\Omega(x,y)$ is a Copula:

$$\int \Omega(x,y)\,\mathrm{d}x = 1 \quad \text{and} \quad \int \Omega(x,y)\,\mathrm{d}y = 1$$

Application in CT



$\mathbf{g}\mid\mathbf{f}$	$\mathbf{f}\mid\mathbf{z}$	$\mathbf{z}$	$\mathbf{c}$
$\mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon}$	iid Gaussian	iid	$c(\mathbf{r}) \in \{0,1\}$
$\mathbf{g}\mid\mathbf{f} \sim \mathcal{N}(\mathbf{H}\mathbf{f}, \sigma_\epsilon^2\mathbf{I})$	or	or	$1 - \delta(z(\mathbf{r}) - z(\mathbf{r}'))$
Gaussian	Gauss-Markov	Potts	binary

Proposed algorithm

$$p(\mathbf{f}, \mathbf{z}, \boldsymbol{\theta}|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f}, \mathbf{z}, \boldsymbol{\theta})\, p(\mathbf{f}|\mathbf{z}, \boldsymbol{\theta})\, p(\boldsymbol{\theta})$$

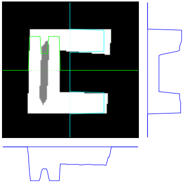
General scheme:

$$\widehat{\mathbf{f}} \sim p(\mathbf{f}|\widehat{\mathbf{z}}, \widehat{\boldsymbol{\theta}}, \mathbf{g}) \longrightarrow \widehat{\mathbf{z}} \sim p(\mathbf{z}|\widehat{\mathbf{f}}, \widehat{\boldsymbol{\theta}}, \mathbf{g}) \longrightarrow \widehat{\boldsymbol{\theta}} \sim (\boldsymbol{\theta}|\widehat{\mathbf{f}}, \widehat{\mathbf{z}}, \mathbf{g})$$

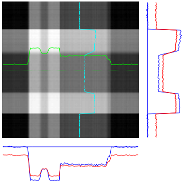
Iterative algorithme:

- Estimate $\mathbf{f}$ using $p(\mathbf{f}|\widehat{\mathbf{z}}, \widehat{\boldsymbol{\theta}}, \mathbf{g}) \propto p(\mathbf{g}|\mathbf{f}, \boldsymbol{\theta})\, p(\mathbf{f}|\widehat{\mathbf{z}}, \widehat{\boldsymbol{\theta}})$
  Needs optimisation of a quadratic criterion.
- Estimate $\mathbf{z}$ using $p(\mathbf{z}|\widehat{\mathbf{f}}, \widehat{\boldsymbol{\theta}}, \mathbf{g}) \propto p(\mathbf{g}|\widehat{\mathbf{f}}, \widehat{\mathbf{z}}, \widehat{\boldsymbol{\theta}})\, p(\mathbf{z})$
  Needs sampling of a Potts Markov field.
- Estimate $\boldsymbol{\theta}$ using
  $p(\boldsymbol{\theta}|\widehat{\mathbf{f}}, \widehat{\mathbf{z}}, \mathbf{g}) \propto p(\mathbf{g}|\widehat{\mathbf{f}}, \sigma_\epsilon^2 \mathbf{I})\, p(\widehat{\mathbf{f}}|\widehat{\mathbf{z}}, (m_k, v_k))\, p(\boldsymbol{\theta})$
  Conjugate priors ⟶ analytical expressions.

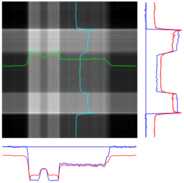
Results



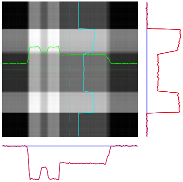
Original



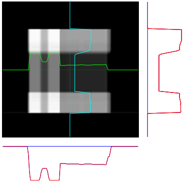
Backprojection



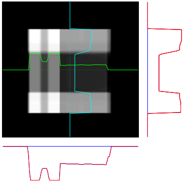
Filtered BP



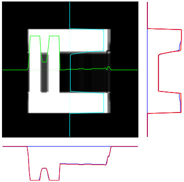
LS



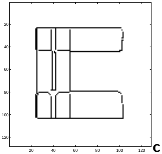
Gauss-Markov+pos

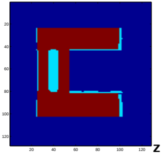


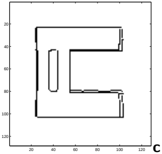
GM+Line process



GM+Label process





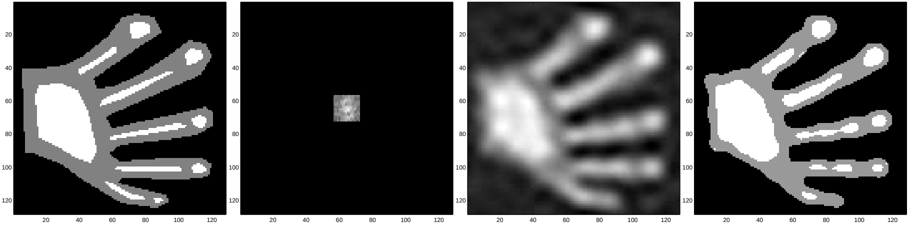


Application in Microwave imaging

$$g(\omega) = \int f(\mathbf{r}) \exp\{-j(\omega.\mathbf{r})\} \, \mathrm{d}\mathbf{r} + \epsilon(\omega)$$

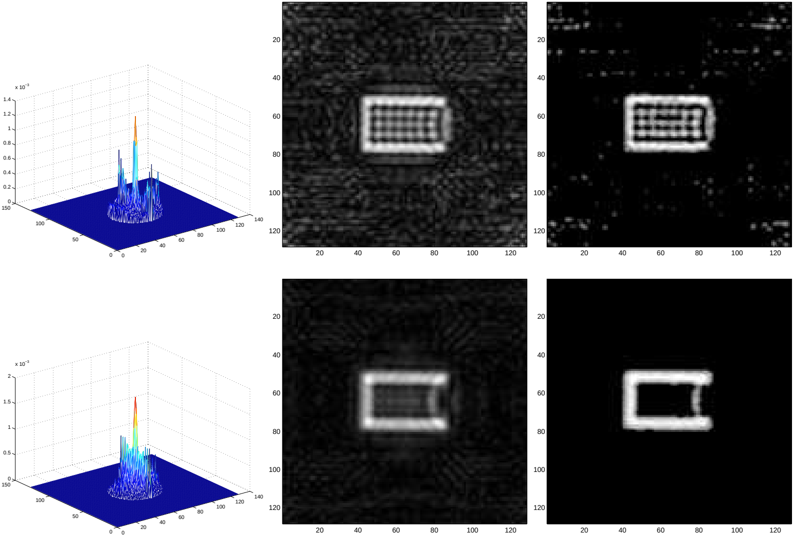
$$g(u,v) = \iint f(x,y) \exp\{-j(ux+vy)\} \, \mathrm{d}x \, \mathrm{d}y + \epsilon(u,v)$$

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon}$$



$f(x,y)$ $g(u,v)$ $\widehat{\mathbf{f}}$ IFT $\widehat{\mathbf{f}}$ Proposed method

Application in Microwave imaging



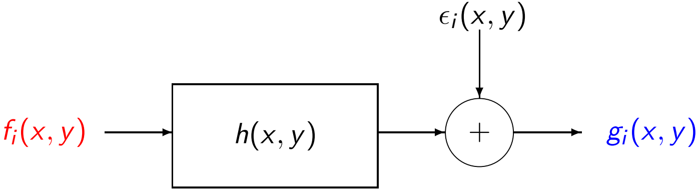
Conclusions

- Bayesian Inference for inverse problems
- Approximations (Laplace, MCMC, Variational)
- Gauss-Markov-Potts are useful prior models for images incorporating regions and contours
- Separable approximations for Joint posterior with Gauss-Markov-Potts priors
- Application in different CT (X ray, US, Microwaves, PET, SPECT)

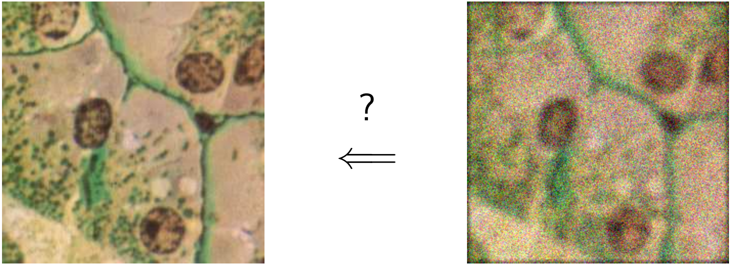
Perspectives :

- Efficient implementation in 2D and 3D cases
- Evaluation of performances and comparison with MCMC methods
- Application to other linear and non linear inverse problems: (PET, SPECT or ultrasound and microwave imaging)

Color (Multi-spectral) image deconvolution



Observation model : $\mathbf{g}_i = \mathbf{H}\mathbf{f}_i + \boldsymbol{\epsilon}_i, \quad i = 1, 2, 3$

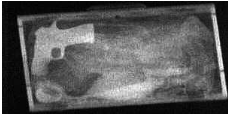


Images fusion and joint segmentation

(with O. Féron)

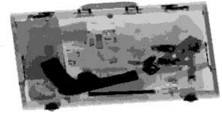
$$\left\{\begin{array}{l} g_i(\mathbf{r}) = f_i(\mathbf{r}) + \epsilon_i(\mathbf{r}) \\ p(f_i(\mathbf{r})|z(\mathbf{r}) = k) = \mathcal{N}(m_{i\,k}, \sigma_{i\,k}^2) \\ p(\underline{\mathbf{f}}|\mathbf{z}) = \prod_i p(\mathbf{f}_i|\mathbf{z}) \end{array}\right.$$

$\mathbf{g}_1$

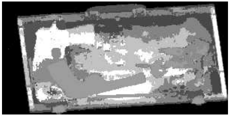


$\mathbf{g}_2$

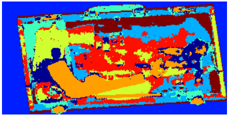
$\longrightarrow$



$\widehat{\mathbf{f}}_1$



$\widehat{\mathbf{f}}_2$

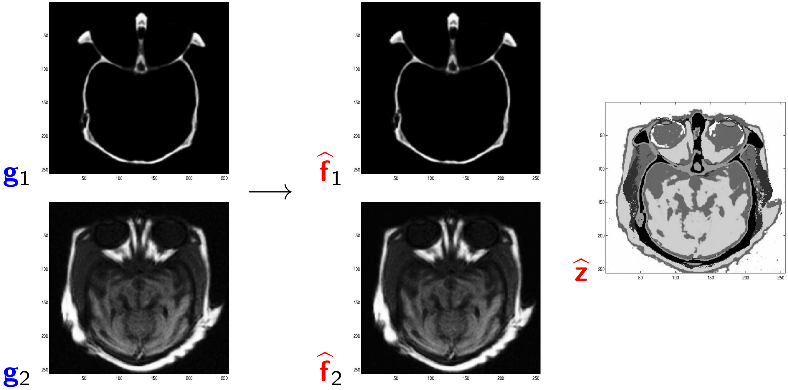


$\widehat{\mathbf{z}}$

Data fusion in medical imaging

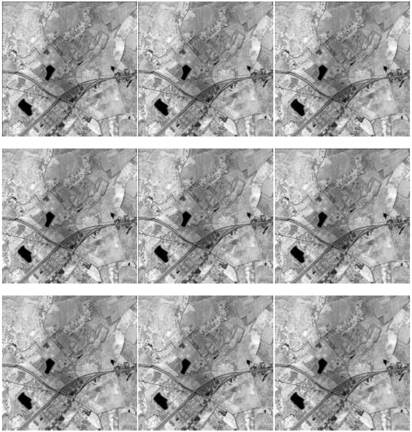
(with O. Féron)

$$\begin{cases} g_i(\mathbf{r}) = f_i(\mathbf{r}) + \epsilon_i(\mathbf{r}) \\ p(f_i(\mathbf{r})|z(\mathbf{r}) = k) = \mathcal{N}(m_{i\,k}, \sigma_{i\,k}^2) \\ p(\underline{\mathbf{f}}|\mathbf{z}) = \prod_i p(\mathbf{f}_i|\mathbf{z}) \end{cases}$$



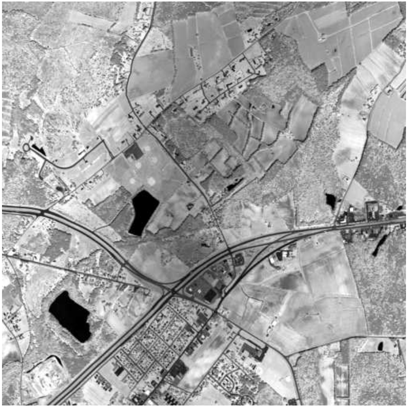
Super-Resolution

(with **F. Humblot**)



Low Resolution images

$\overset{?}{\Longrightarrow}$

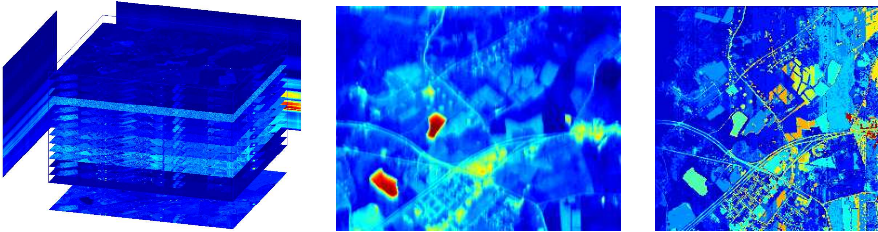


High Resolution image

Joint segmentation of hyper-spectral images

(with N. Bali & A. Mohammadpour)

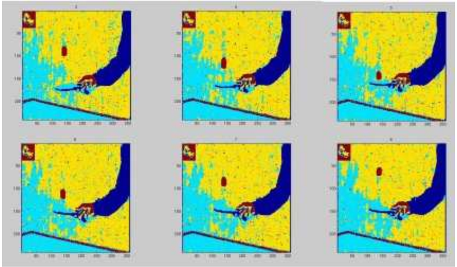
$$\begin{cases} g_i(\mathbf{r}) = f_i(\mathbf{r}) + \epsilon_i(\mathbf{r}) \\ p(f_i(\mathbf{r})|z(\mathbf{r}) = k) = \mathcal{N}(m_{i\,k}, \sigma_{i\,k}^2), \quad k = 1, \cdots, K \\ p(\underline{\mathbf{f}}|\mathbf{z}) = \prod_i p(\mathbf{f}_i|\mathbf{z}) \\ m_{i\,k} \;\; \text{follow a Markovian model along the index} \;\; i \end{cases}$$



Segmentation of a video sequence of images

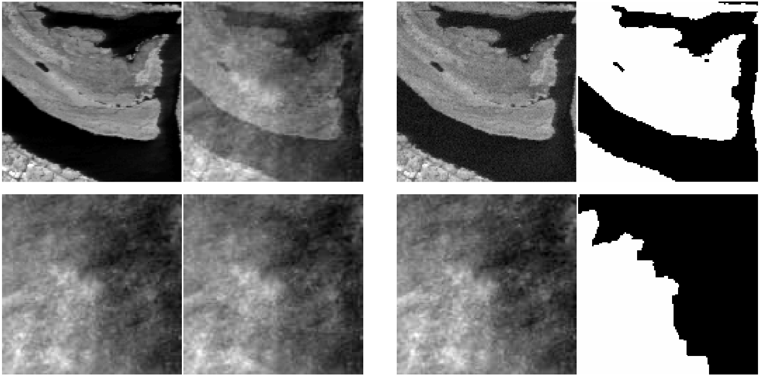
(with P. Brault)

$$\begin{cases} g_i(\mathbf{r}) = f_i(\mathbf{r}) + \epsilon_i(\mathbf{r}) \\ p(f_i(\mathbf{r})|z_i(\mathbf{r}) = k) = \mathcal{N}(m_{i\,k}, \sigma^2_{i\,k}), \quad k = 1, \cdots, K \\ p(\underline{\mathbf{f}}|\mathbf{z}) = \prod_i p(\mathbf{f}_i|\mathbf{z}_i) \\ z_i(\mathbf{r}) \;\; \text{follow a Markovian model along the index} \;\; i \end{cases}$$



Source separation

(with H. Snoussi & M. Ichir) $\begin{cases} g_i(\mathbf{r}) = \sum_{j=1}^{N} A_{ij} f_j(\mathbf{r}) + \epsilon_i(\mathbf{r}) \\ p(f_j(\mathbf{r}) | z_j(\mathbf{r}) = k) = \mathcal{N}(m_{j\,k}, \sigma^2_{j\,k}) \\ p(A_{ij}) = \mathcal{N}(A_{0\,ij}, \sigma^2_{0\,ij}) \end{cases}$



$\underline{\mathbf{f}}$ $\underline{\mathbf{g}}$ $\underline{\widehat{\mathbf{f}}}$ $\underline{\widehat{\mathbf{z}}}$

Some references

- A. Mohammad-Djafari (Ed.) Problèmes inverses en imagerie and en vision (Vol. 1 and 2), Hermes-Lavoisier, Traité Signal and Image, IC2, 2009,
- A. Mohammad-Djafari (Ed.) Inverse Problems in Vision and 3D Tomography, ISTE, Wiley and sons, ISBN: 9781848211728, December 2009, Hardback, 480 pp.
- H. Ayasso and Ali Mohammad-Djafari Joint NDT Image Restoration and Segmentation using Gauss-Markov-Potts Prior Models and Variational Bayesian Computation, To appear in IEEE Trans. on Image Processing, TIP-04815-2009.R2, 2010.
- H. Ayasso, B. Duchene and A. Mohammad-Djafari, Bayesian Inversion for Optical Diffraction Tomography Journal of Modern Optics, 2008.
- A. Mohammad-Djafari, Gauss-Markov-Potts Priors for Images in Computer Tomography Resulting to Joint Optimal Reconstruction and segmentation, International Journal of Tomography & Statistics 11: W09. 76-92, 2008.
- A Mohammad-Djafari, Super-Resolution : A short review, a new method based on hidden Markov modeling of HR image and future challenges, The Computer Journal doi:10,1093/comjnl/bxn005:, 2008.
- O. Féron, B. Duchène and A. Mohammad-Djafari, Microwave imaging of inhomogeneous objects made of a finite number of dielectric and conductive materials from experimental data, Inverse Problems, 21(6):95-115, Dec 2005.
- M. Ichir and A. Mohammad-Djafari, Hidden markov models for blind source separation, IEEE Trans. on Signal Processing, 15(7):1887-1899, Jul 2006.
- F. Humblot and A. Mohammad-Djafari, Super-Resolution using Hidden Markov Model and Bayesian Detection Estimation Framework, EURASIP Journal on Applied Signal Processing, Special number on Super-Resolution Imaging: Analysis, Algorithms, and Applications:ID 36971, 16 pages, 2006.
- O. Féron and A. Mohammad-Djafari, Image fusion and joint segmentation using an MCMC algorithm, Journal of Electronic Imaging, 14(2):paper no. 023014, Apr 2005.
- H. Snoussi and A. Mohammad-Djafari, Fast joint separation and segmentation of mixed images, Journal of Electronic Imaging, 13(2):349-361, April 2004.
- A. Mohammad-Djafari, J.F. Giovannelli, G. Demoment and J. Idier, Regularization, maximum entropy and probabilistic methods in mass spectrometry data processing problems, Int. Journal of Mass Spectrometry, 215(1-3):175-193, April 2002.

Thanks, Questions and Discussions

Thanks to:

My graduated PhD students:

- H. Snoussi, M. Ichir, (Sources separation)
- F. Humblot (Super-resolution)
- H. Carfantan, O. Féron (Microwave Tomography)
- S. Fékih-Salem (3D X ray Tomography)

My present PhD students:

- H. Ayasso (Optical Tomography, Variational Bayes)
- D. Pougaza (Tomography and Copula)
- ———————
- Sh. Zhu (SAR Imaging)
- D. Fall (Emission Positon Tomography, Non Parametric Bayesian)

My colleages in GPI (L2S) & collaborators in other instituts:

- B. Duchêne & A. Joisel (Inverse scattering and Microwave Imaging)
- N. Gac & A. Rabanal (GPU Implementation)
- Th. Rodet (Tomography)
- ———————
- A. Vabre & S. Legoupil (CEA-LIST), (3D X ray Tomography)
- E. Barat (CEA-LIST) (Positon Emission Tomography, Non Parametric Bayesian)
- C. Comtat (SHFJ, CEA)(PET, Spatio-Temporal Brain activity)

Questions and Discussions