Notiztitel 17.10.2008

7.5.2014

# Gabor-Transformation

$$y(t) = \sum_{m=-\infty}^{\infty}{}' \sum_{n=-\infty}^{\infty}{}' C_{nm}\, g_{nm}(t)$$

$$g_{nm}(t) = s(t - m\tau_0)\, e^{j\Omega n t}$$

Gauß $= \quad s(t) = \left(\frac{\sqrt{2}}{\sigma}\right)^{1/2} \cdot e^{-\pi t^2/\sigma^2}$

with $\Omega \leq \frac{2\pi}{\tau_0}$

Critical sampling if above equation is fulfilled with the equality sign, otherwise, we have over-sampling

Previous definition for $\Omega$ was

$$\Omega = \frac{2\pi}{N \cdot T}$$

how : $\Omega = \frac{2\pi}{\tau_0} = \frac{2\pi}{N \cdot T}$

Combined basis function is of type :

$$\psi_{nm}(t) = e^{\left(-\pi\frac{t^2}{\sigma^2} + j\omega t\right)}$$

(very similar Morlet Wavelet

$$\psi(t) = e^{-\frac{\pi t^2}{\sigma^2}}\left(e^{j\omega t} - e^{-\frac{\omega^2}{2}}\right)$$

$\Rightarrow$ It can be proved that there (Gaussian-like) basis functions minimize the product of time and frequency resolution $\Delta t \cdot \Delta \omega$ (see 2.22).

## The discrete GT :

discretization of time : $t = k \cdot T$

$$e^{j\Omega n t} = e^{j\frac{2\pi}{NT} n t}$$

$$= e^{j\frac{2\pi}{\cancel{NT}} n\, k\cancel{T}} = e^{j\overbrace{\left(\frac{2\pi}{N}\right)}^{\Omega} n \cdot k}$$

$$= e^{j\Omega n k}$$

$$\Rightarrow y(k) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} c_{nm}\, g_{nm}(k)$$

$$g_{nm}(k) = \delta(k - mN)\, e^{j\Omega n k}$$

$\Rightarrow$ This corresponds perfectly to critical sampling

For over-sampling: $\Omega \leq \frac{2\pi}{N}$

$= \frac{2\pi}{N^*}$ with $N^* > N$

which would then correspond to $M \cdot N^*$ Gabor coefficients, Where $M \cdot N^{\dagger} > M \cdot N \sim K$

too many Gabor coefficients

# Computation of Gabor-Coefficients

usually we have now no more a orthogonal function system

$\Rightarrow$ alternative methods for

# Coefficient Computation

## 1) Least Squares Method

$$y(k) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} c_{mn} \cdot g_{mn}(k)$$

$$= c_{00} \cdot g_{00}(k) + c_{01} \cdot g_{01}(k) + \dots$$

$$c_{0N-1}\, g_{0N-1}(k) + c_{10} \cdot g_{10}(k) +$$

$$c_{11} \cdot g_{11}(k) + \cdots \; c_{N-1\,M-1} \cdot g_{N-1\,M-1}(k))$$

Then write this equation also for other time samples $y(k+1)$, $y(k+2)$

Finally:

$$\begin{pmatrix} y(1) \\ y(2) \\ \vdots \\ y(K) \end{pmatrix} = K\left\{ \begin{pmatrix} g_{00}(1) & g_{01}(1) & \cdots & g_{N-1\,M-1}(1) \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ g_{00}(K) & \cdots & \cdots & g_{N-1\,M-1}(K) \end{pmatrix} \right. \cdot \begin{pmatrix} c_{00} \\ c_{01} \\ \vdots \\ c_{0\,M-1} \\ c_{10} \\ c_{11} \\ \vdots \\ c_{N-1\,M-1} \end{pmatrix}$$

$$\overbrace{\underline{y}} = \overbrace{G}^{M \cdot N} \cdot \overbrace{\underline{c}}$$

$$\Rightarrow \underline{y} = G \cdot \underline{c} \quad \Rightarrow \underline{c} = G^{-1} \cdot \underline{y}$$

can be accomplished in case of critical sampling: $M \cdot N = K$

in case of oversampling:

\# coefficients $>$ \# samples

$\rightarrow$ no solution

Sub-sampling: more equations available than coefficients

$\rightarrow$ over-determined equation system:

error $\underline{e} = \underline{y} - G \cdot \underline{c}$

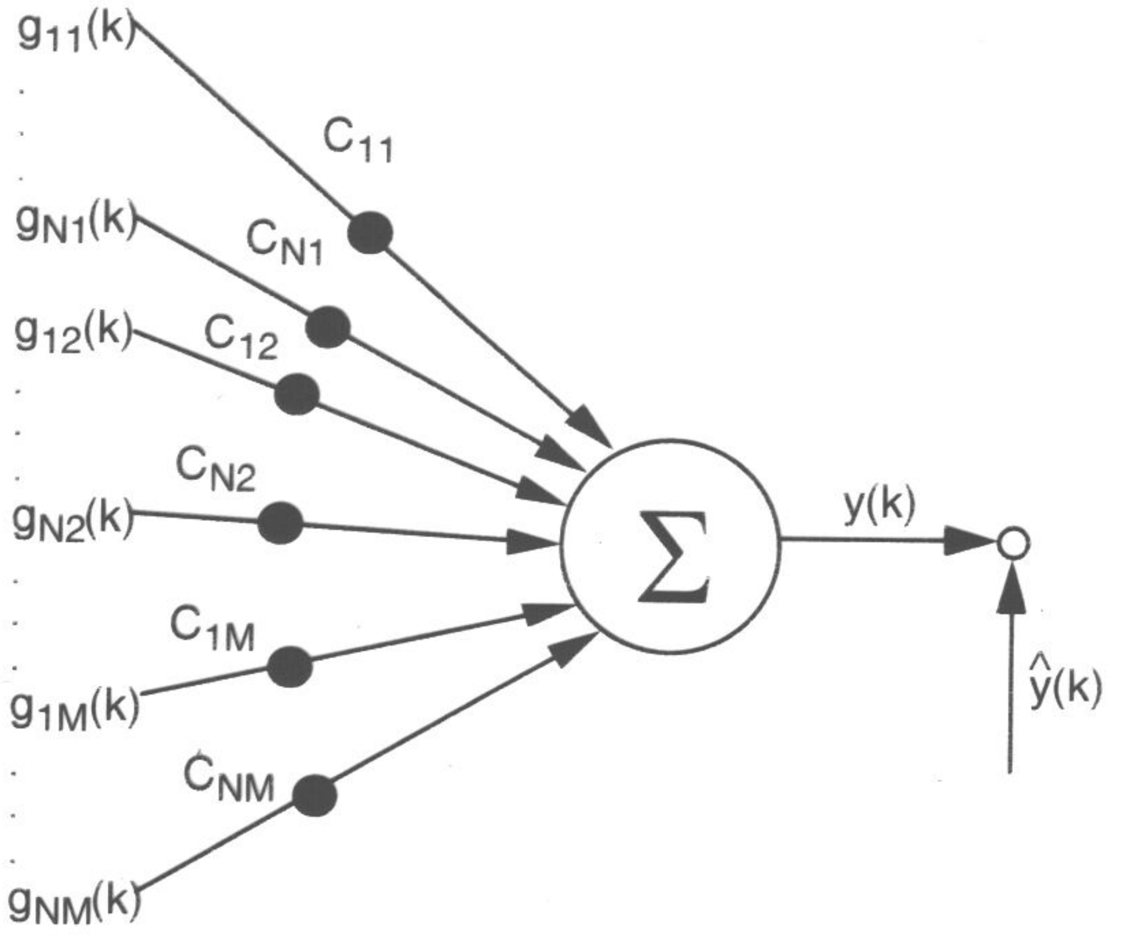
minimize error: $\frac{d}{d\underline{c}}\left(\underline{e}^T \cdot \underline{e}\right)$

$= \frac{d}{d\underline{c}} \sum e_i^2$

the solution is:

$$\underline{c} = \left(G^T \cdot G\right)^{-1} \cdot G^T \cdot y$$

another alternative: Neural Network



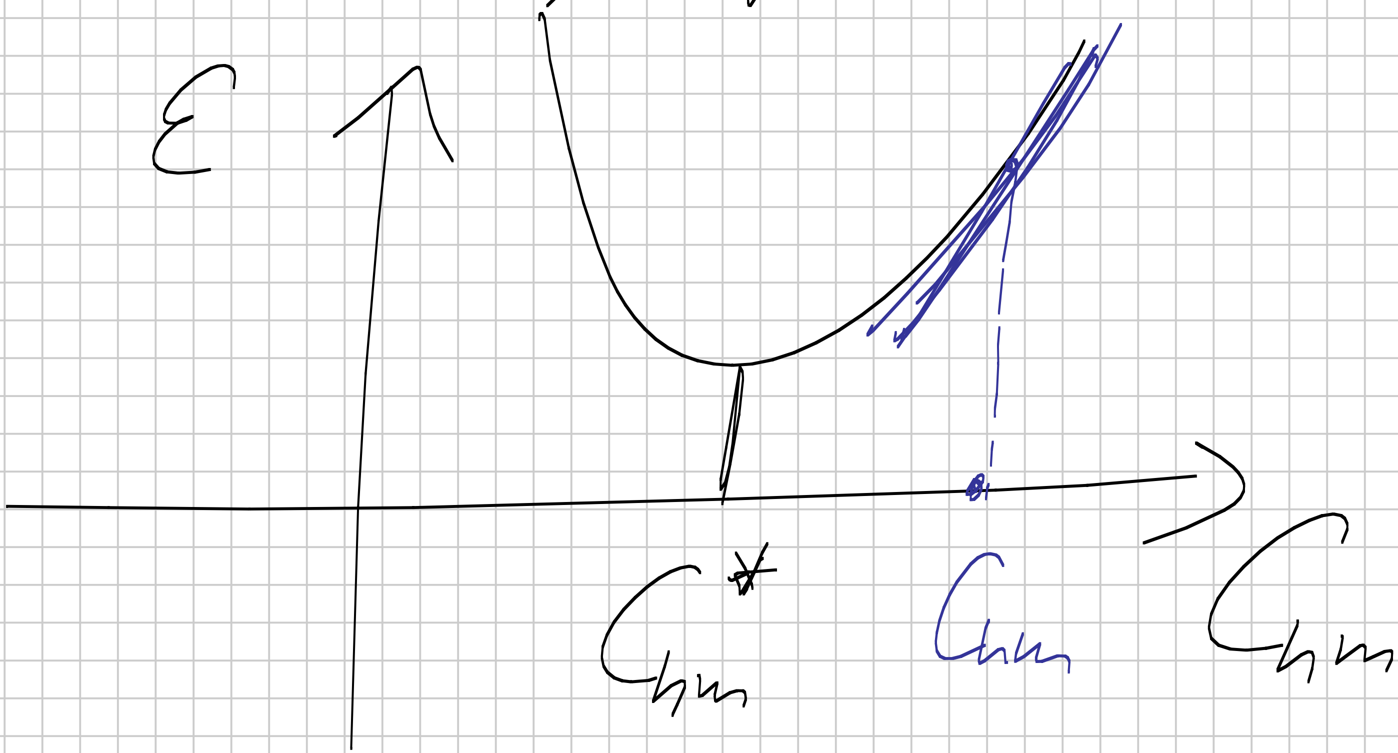
Minimize the error at the output of the neuron:

$$\varepsilon(k) = \frac{1}{2}\Big(\underbrace{\hat{y}(k)}_{\text{Correct time sample}} - \underbrace{y(k)}_{\text{synthesized sample of neuron}}\Big)^2$$

$$= \frac{1}{2}\left(\hat{y}(k) - \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} c_{mn}\, g_{mn}(k)\right)^2$$

Learning rule for neuron:

$$c_{hm}(k) = c_{hm}(k-1) - \beta \frac{\partial \varepsilon(k)}{\partial c_{hm}}$$



$$\frac{\partial \mathcal{E}(k)}{\partial c_{nm}} = -\left[\hat{y}(k) - \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} c_{nm}\, g_{nm}(k)\right] \cdot g_{nm}(k)$$

$\Rightarrow$ Final learning rule:

$$c_{nm}(k) = c_{nm}(k-1) + \beta \cdot \left[\hat{y}(k) - \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} c_{nm}\, g_{nm}(k)\right] \cdot g_{nm}(k)$$

## 2.3 Special Transformations in the time domain

### 2.3.1 Principal Component Analysis

PCA, KLT

Consider: $K$ of different pattern vectors $\underline{x}(k)$

$k = 1, \dots K$

Major statistical properties:

average vector:

$$\underline{m}_x = E\{\underline{x}\} = \frac{1}{K} \cdot \sum_{k=1}^{K} \underline{x}(k)$$

Covariance matrix:

$$C_x = E\{(\underline{x} - \underline{m}_x)(\underline{x} - \underline{m}_x)^T\}$$

$$= \frac{1}{K} \sum_{k=1}^{K} \left( \underline{x}(k) \cdot \underline{x}(k)^T \right) - \underline{m}_x \cdot \underline{m}_x^T$$

<u>next step</u>: Compute Eigenvalues + Eigenvectors of $C_x$

$$C_x \cdot \underline{e}_n = \lambda_n \cdot \underline{e}_n$$

$n = 1, \dots N$ (dimension of vector $\underline{x}$)