ECE 4270

Fundamentals of Digital Signal Processing

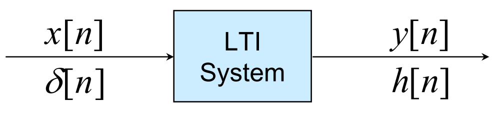
Lecture 25: Non-linear Signal Processing

School of Electrical and Computer Engineering
Georgia Institute of Technology
Summer, 2004

Outline

- Review of linear systems & their characteristics
  - Time invariant and linear
  - Spectral analysis and modification
- Need for non-linear filtering
  - A distribution matching scenario
  - A curve fitting scenario
  - Various kinds of noise and distortion
  - Translation of spectrum
  - Data dependency; decision circuitry

LTI Discrete-Time Systems



- Linearity (superposition):

$$T\{ax_1[n]+bx_2[n]\}=aT\{x_1[n]\}+bT\{x_2[n]\}$$

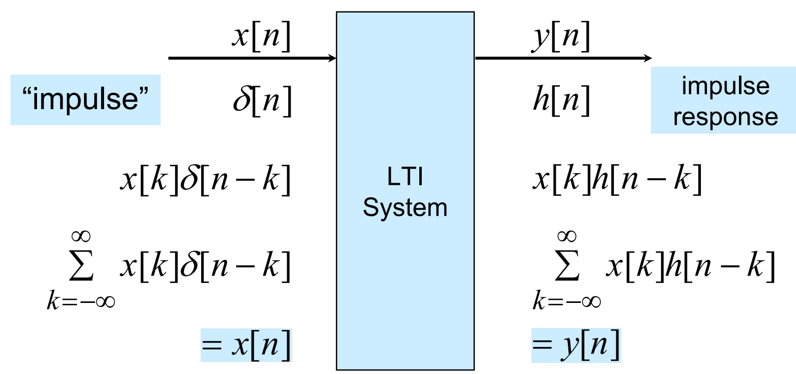
- Time-Invariance (shift-invariance):

$$x_1[n]=x[n-n_d] \;\Rightarrow\; y_1[n]=y[n-n_d]$$

- LTI implies discrete convolution:

$$y[n]=\sum_{k=-\infty}^{\infty} x[k]h[n-k]=x[n]*h[n]=h[n]*x[n]$$

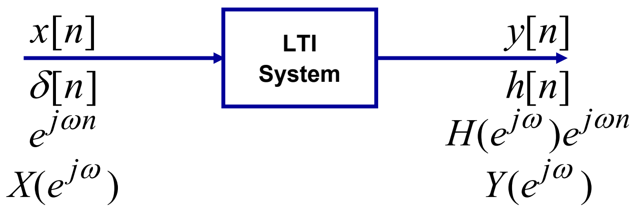
LTI Discrete-Time Systems



convolution sum
$$y[n]=\sum_{k=-\infty}^{\infty} x[k]h[n-k]=x[n]*h[n]=h[n]*x[n]$$

Convolution Theorem

$$y[n]=\sum_{k=-\infty}^{\infty} x[k]h[n-k] \Leftrightarrow Y(e^{j\omega})=X(e^{j\omega})H(e^{j\omega})$$



$$Y(e^{j\omega})=H(e^{j\omega})X(e^{j\omega})$$

$$\left|Y(e^{j\omega})\right|=\left|H(e^{j\omega})\right|\left|X(e^{j\omega})\right|$$

$$\angle Y(e^{j\omega})=\angle H(e^{j\omega})+\angle X(e^{j\omega})$$

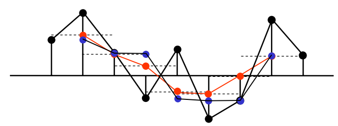
Non-linear Processing

- Need for non-linear processing
  - A distribution matching scenario
  - A curve fitting scenario
  - Various kinds of noise and distortion
  - Translation of spectrum
  - Data dependency; decision circuitry

Practical Nonlinear Processing Techniques

- Median filtering, rank order filtering
- Clipping, center-clipping
- Companding; μ-law, A-law
- Automatic gain control
- Various modulation systems – AM, FM
- Homomorphic processing
- Neural networks ⇒ symbolic processing, next lecture

Curve Fitting and Filtering



Curve fitting (0th order)

$$y[n]=\arg\min_{\mu}\sum_{i=-K}^{K}\delta(\mu-x[n+i])$$

1. Quadratic error (L2 norm): $\delta(\bullet)=\|\bullet\|^2$

- $y[n]=\frac{1}{2K+1}\sum_{i=-K}^{K}x[n+i]$ Moving average of equal weights

2. Absolute error (L1 norm): $\delta(\bullet)=|\bullet|$

- $y[n]=median\{x[n+i], i=-K,..,0,..,K\}$

$(2K+1)$-point median filter

Median Filter

Objective: minimize absolute (L1) error

$$y[n] = \arg\min_{\mu} \sum_{i=-K}^{K} |\mu - x[n+i]|$$

$$\text{Let } E = \sum_{i=-K}^{K} |\mu - x[n+i]|$$

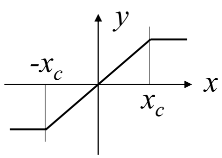
$$\frac{dE}{d\mu} = \sum_{i=-K}^{K} \text{sgn}(\mu - x[n+i]) \qquad \text{note}: \text{sgn}(0) = 0$$

$\frac{dE}{d\mu}$ attains 0 at $x[n+j]$ within the window such that there are equal number of values greater and less than $x[n+j]$, that is, at the median.

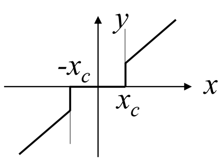
Use of Median Filter

- Often used in image and video processing
- Can preserve edges of an object while removing artifacts like speckles
- Effective in removing impulsive noise
- Also popular in smoothing estimated voicing parameters (or other discrete parameters)

Clipping



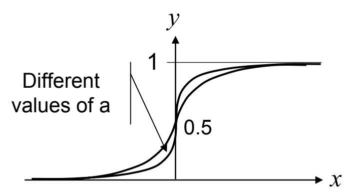
$$y[n] = \begin{cases} x[n], & -x_c \le x[n] \le x_c \\ x_c, & x[n] > x_c \\ -x_c, & x[n] < -x_c \end{cases}$$



$$y[n] = \begin{cases} 0, & -x_c \le x[n] \le x_c \\ x[n], & x[n] > x_c \text{ or } x[n] < -x_c \end{cases}$$

Center clipping is a primitive way to suppress low level "background" noise; x_c needs to be adaptively estimated. The function $y(x)$ is usually smooth in advanced methods.

Compression and Expansion



Examples of smooth 0-1 functions

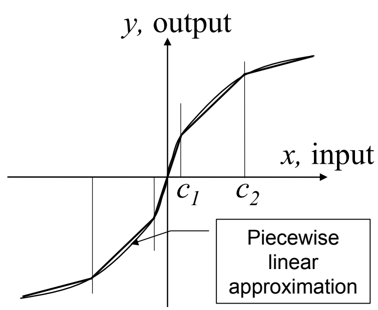
- Sigmoid functions:

$$y = 1/(1+e^{-ax})$$

$$dy/dx = ay(1-y)$$

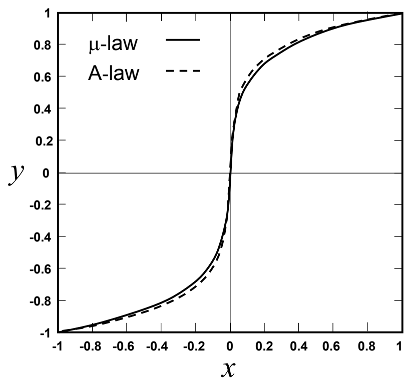
- Hypertangent function:

$$y = \tanh(x)$$



General companding

Instantaneous Companding for PCM



μ-law

$$y = \mathrm{sgn}(x) \bullet \frac{\ln(1+\mu|x|)}{\ln(1+\mu)}$$

μ=255, x is normalized

$$x = \mathrm{sgn}(y)\frac{(1+\mu)^{|y|}-1}{\mu}$$

A-law

$$y = \mathrm{sgn}(x)\frac{A|x|}{1+\ln A}, \qquad 0 \le |x| < \frac{1}{A}$$

$$= \mathrm{sgn}(x)\frac{1+\ln A|x|}{1+\ln A}, \qquad \frac{1}{A} \le |x| \le 1$$

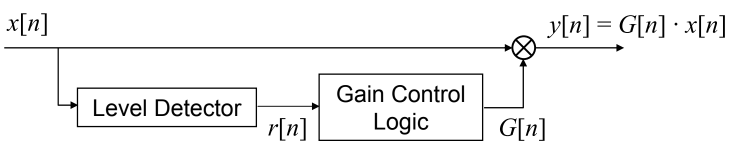
A=87.6

Companding schemes are used in digital telephony to get more dynamic range for voice before an "8"-bit PCM quantizer; the result is equivalent to 12-14 bits linear PCM without companding.

Automatic Gain Control

- Purpose: properly scale input signal power level without adversely affecting the dynamics of the signal; for example
  - In A/D, maintain a proper match in dynamic range between the signal and the quantizer
  - In receivers, maintain a certain output level regardless of the strength of the received signal
- Gain Control has two major adjustment phases:
  - Attack: to reduce the signal dynamic range by attenuating high amplitude signals
  - Release: to increase the signal dynamic range by amplifying low amplitude signals
- Automatic Gain Control: "instantaneously" and "automatically" controls the signal dynamic range upon "detection" of the signal level

Automatic Gain Control Block Diagram

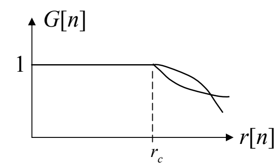


- Level $r[n]$ can be defined as a low-pass filtered version of $|x[n]|$, envelope of $x[n]$, or RMS value of $x[n]$, etc., e.g. $r[n] = \alpha r[n-1] + (1-\alpha)|x[n]|$
- The Gain Control Logic determines
  - attack time constant: how quickly the gain is reduced when input signal amplitude rises
  - release time constant: how quickly the gain is increased when input signal amplitude falls

Gain Control (cont.)

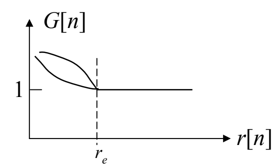
- Gain controller function
  - Attack time, $r[n] > \xi r[n\text{-}1]$, ρ>0

$$G(n) = \begin{cases} \left(r[n]/r_c\right)^{-\rho}, & \text{if } r[n] > r_c \\ 1, & \text{if } r[n] \le r_c \end{cases}$$



  - Release time, $r[n] \le \xi r[n\text{-}1]$, θ>0

$$G(n) = \begin{cases} 1, & \text{if } r[n] \ge r_e \\ \left(r[n]/r_e\right)^{-\theta}, & \text{if } r[n] < r_e \end{cases}$$



Modulation

- Amplitude modulation

$$y[n] = x[n] \bullet A\cos(\omega_c n)$$

$$y[n] = x[n] \bullet (-1)^n \quad \text{i.e. when } \omega_c = \pi$$

Amplitude modulation causes translation of spectrum.

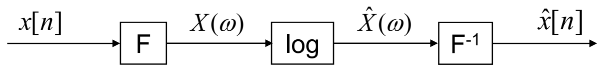
- Frequency modulation

$$y(t) = A\cos[\phi_{FM}(t)] = A\cos\left(\omega_c t + \Delta_\omega \int_0^t x(\tau)d\tau + \theta\right)$$

$$\frac{d\phi_{FM}(t)}{dt} = \omega_c + \Delta_\omega x(t)$$

Frequency modulation causes translation and expansion of spectrum and needs care if implemented digitally.

Complex Cepstrum



$$\log X(\omega) = \log|X(\omega)| + j\angle X(\omega)$$

$$X(\omega) = X_1(\omega)X_2(\omega)$$

$$\log X(\omega) = \log X_1(\omega) + \log X_2(\omega)$$

$$= \underbrace{(\log|X_1(\omega)| + \log|X_2(\omega)|)}_{\log|X(\omega)|} + j\underbrace{(\angle X_1(\omega) + \angle X_2(\omega))}_{\angle X(\omega) \mod 2\pi}$$

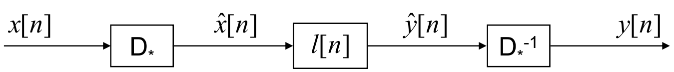
$$\hat{x}[n] = \int_{-\pi}^{\pi} \log[X(\omega)]e^{j\omega n}d\omega \Longrightarrow$$ complex cepstrum

$$\log|X(\omega)| \overset{\mathcal{F}}{\Longleftrightarrow} c[n] = (\hat{x}[n] + \hat{x}[-n])/2 \Longrightarrow$$ real cepstrum

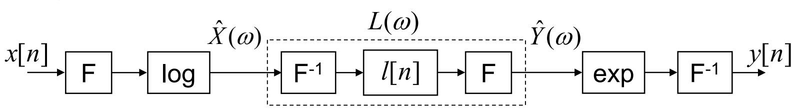
n is called quefrency

Homomorphic Processing

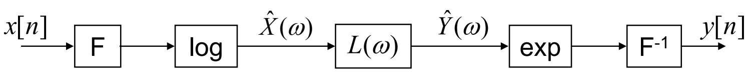
Quefrency domain implementation



Expansion



Spectral domain implementation



Terminology in Homomorphic Processing

frequency ⟺ quefrency

Spectrum ⟺ Cepstrum

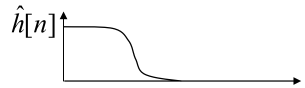
$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \qquad \hat{x}[n] = \int_{-\pi}^{\pi} \log[X(\omega)]e^{j\omega n}d\omega$$

Filter(ing) ⟺ Lifter(ing)

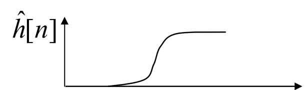
$$y[n] = x[n] * h[n] \qquad \hat{y}[n] = \hat{x}[n] + \hat{h}[n]$$

$$Y(\omega) = X(\omega)H(\omega) \qquad \log Y(\omega) = \log X(\omega) + \log H(\omega)$$

$$\Longrightarrow \tilde{y}[n] = \hat{x}[n] \bullet \hat{h}[n] \Longleftarrow$$ Somewhat confusing



Low-quefrency liftering



High-quefrency liftering

Complex Cepstrum of Sequences with Rational z-transforms

$$X(z) = Az^{-r} \frac{\prod_{k=1}^{M_i}(1-a_k z^{-1})\prod_{k=1}^{M_o}(1-b_k z)}{\prod_{k=1}^{N_i}(1-c_k z^{-1})\prod_{k=1}^{N_o}(1-d_k z)}$$

i = inside u.c.
o = outside u.c.

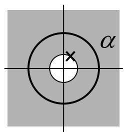
$$\hat{X}(z) = \log X(z) = \log A + \sum_{k=1}^{M_i}\log(1-a_k z^{-1}) + \sum_{k=1}^{M_o}\log(1-b_k z)$$

z^{-r} removed

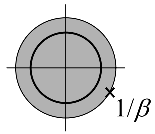
$$+\sum_{k=1}^{M_i}\log(1-a_k z^{-1}) + \sum_{k=1}^{M_o}\log(1-b_k z)$$

ROCs

$$\log(1-\alpha z^{-1}) = -\sum_{n=1}^{\infty}\frac{\alpha^n}{n}z^{-n}, \quad |\alpha z^{-1}|<1$$



$$\log(1-\beta z) = -\sum_{n=1}^{\infty}\frac{\beta^n}{n}z^{n}, \quad |\beta z|<1$$



Complex Spectrum as Root Power Sum

$$\hat{x}[n] = \log(A)\delta[n]$$ Cepstrum at 0 quefrency

$$-\left[\sum_{k=1}^{M_i}\frac{a_k^n}{n} - \sum_{k=1}^{N_i}\frac{c_k^n}{n}\right]u[n-1]$$ Right-side cepstrum

$$+\left[\sum_{k=1}^{M_o}\frac{b_k^{-n}}{n} - \sum_{k=1}^{N_o}\frac{d_k^{-n}}{n}\right]u[-n+1]$$ Left-side cepstrum

Example: a train of uniformly spaced samples with varying amplitude

$$p[n] = \sum_{l=0}^{L}\lambda_l\delta[n-lN]$$

$$P(z) = \sum_{l=0}^{L}\lambda_l z^{-lN} = \sum_{l=0}^{L}\lambda_l(z^N)^{-l} = \prod_{l=0}^{L}[1-a_l(z^N)^{-1}]$$

$$\log P(z) = \sum_{l=0}^{L}\log[1-a_l(z^N)^{-1}] = \sum_{l=0}^{L}\left[-\sum_{k=1}^{\infty}\frac{a_l^k}{k}(z^N)^{-k}\right]$$

A Linear System Excited by A Pulse Train

$$p[n] = \sum_{l=0}^{L}\lambda_l\delta[n-lN]$$

$$\log P(z) = \sum_{l=0}^{L}\log[1-a_l(z^N)^{-1}] = \sum_{l=0}^{L}\left[-\sum_{k=1}^{\infty}\frac{a_l^k}{k}(z^N)^{-k}\right]$$

$$y[n] = h[n] * p[n] \qquad \hat{y}[n] = \hat{h}[n] + \hat{p}[n]$$