CG351 Digital Signal Processing

Lecture 1

Dr. Tardi Rjahjadi, Room A301
http://www.eng.warwick.ac.uk/cg351

28 Lectures (Term 2), 3 revision classes (Term 3)

Examples classes (from week 13 in Term 2):

(Revision classes, held in CS1.04)
Tuesday 2:05 : CS, Maths, Physics, Philisophy
Friday 1:05 : CSE, EELEC, Electrical

Assesment:

Exam 80%
Assignment 20% [to be submitted before week 20]
-including computer programmng (cannot use Matlab or Mathematica) to be done in any high-level programming language of your choice

Course covers:

- Signal representation: Z transform & Fourier
- Filtering: FIR, IIR, matched filtering
- Applications: speeech, radar, image

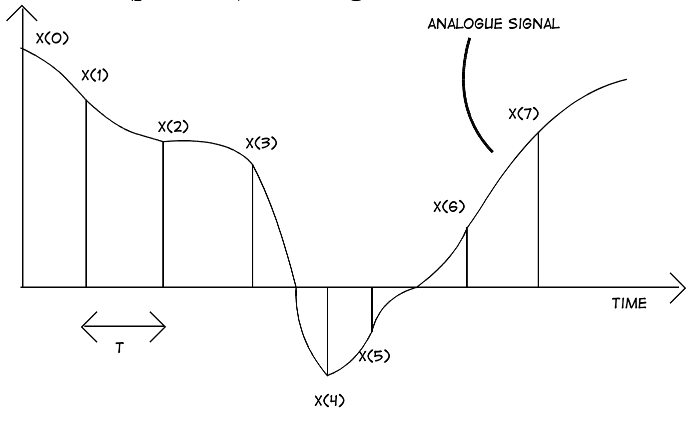
Useful reading (not required to own these books):

1. J.G Proadkis, D.G.Manolis, Introduction to Digital Signal Processing, MaxMillian, 1998
2. S.Mitre, Digital Signal Processing: A Computer-based Approach, McGraw-Hill, 1998

Some handours will exist, but if you miss one you can collect it from Dr. Tjahjadi's office – and **not** in subsequent lectures.

Time Sequence and their representation

Analogue signal – continuous in time (position) and magnitude
Digital signal – discrete in time (position) and magnitude



WHERE T IS SAMPLE INTERVAL, AND SET TO 1 FOR DISCRETE SIGNAL

A causal sequence

$$\{x(n)\}=\{x(0), x(1), x(2), \ldots, x(n), \ldots\}$$

where $x(n)=0$ for $n>0$ represents a sampled (digital) signal.

Z-Transform

The Z transform (ZT) of a sequence {x(n)} is

$$\begin{aligned} X(z) &= \sum_{n=0}^{\infty} x(n) z^{-n} \\ &= n(0).z^{0}+x(1)z^{-1}+x(2)z^{-2}+\ldots \end{aligned} \qquad \text{-(1.1)}$$

$$Eg: \{x(n)\}=\{r, r^{2,} r^{3,} r^{4,} \ldots, r^{n}, \ldots\}$$

$$\begin{aligned} X(z) &= \sum_{n=0}^{\infty} r^{n} z^{-n} \\ &= \sum_{n=0}^{\infty} (rz^{-1})^{n} \\ &= \frac{1}{(1-rz^{-1})} \qquad \text{(from Binomial Theorem)} \end{aligned}$$

converges only if $|rz^{1}|<1$

As a check, do the long divison

```
                1 + r.z^-1 + r2.z^-2 + ... + r^nz^-n + ...
         ____________________________________________
1-r.z^-1 / 1
~~~~~~~~~-(1-r.z^-1)
          ~~~~~~~~~~~
              r.z^-1
            -(r.z^-1 - r^2.z^-2)
            ~~~~~~~~~~~~~~~~~~~~
                     r^2.z^-2
                   -(r^2.z^-2 - r^3.z^-3)
                   ~~~~~~~~~~~~~~~~~~~~~~
                                r^3.z^-3
                                 -etc-
```

$$\text{i.e.} \quad X(z) = 1+rz^{-1}+r^{2}z^{-2}+\ldots+r^{n}z^{-n}+\ldots$$

$$\text{and} \quad \{x(n)\} = \{1, r, r^{2,} r^{3,} r^{4,} \ldots, r^{n}, \ldots\}$$

Shift property

Consider

$$\begin{aligned} Y(z) &= Z^{m} X(z) \\ &= \sum_{n=0}^{\infty} x(n) z^{-(n+m)} \\ &= \sum_{n=0}^{\infty} x(n'-m) z^{n'} \end{aligned} \qquad \text{-(1.2)}$$

where $n'=n+m$

$$\therefore \{Y(n)\}=\{\underbrace{x(n-m)}_{see\,note}\}$$

z^{-1} corresponds to a unit delay

(Note: $\{x(n)\}$ delayed by m sampled intervals (obtained by multiplying $X(z)$ with Z^{-m})

Lecture 2

Linear property

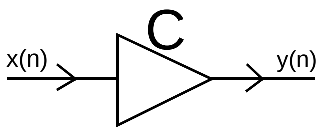
$$\begin{aligned}\text{Suppose}\{c(n)\} &= \{a(n)\}+\{b(n)\}\\ &= \{a(0)+b(0),a(1)+b(1),a(2)+b(2),+\ldots\}\\ &= \{a(n)+b(n)\}\end{aligned}$$

$$\begin{aligned}c(Z) &= \sum_{n=0}^{\infty}(a(n)+b(n))z^{-n}\\ &= A(z)+B(z)\end{aligned}$$

i.e. ZT is linear

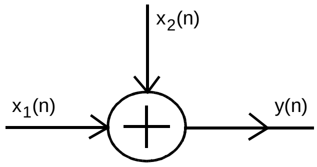
Linear Shift Invariant (LSI) or Linear Time Invariant (LTI) operators

1. Multiplier



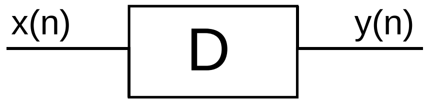
$$\begin{aligned}\{y(n)\} &= \{Cx(n)\}\\ &= C\{x(n)\}\end{aligned}$$

2. Adder



$$\begin{aligned} &= \{x_1(n)+x_2(n)\}\\ \{y(n)\} &= \{x_1(n)\}+\{x_2(n)\}\end{aligned}$$

3. Delay



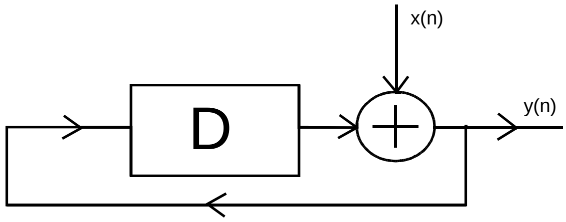
$$\{y(n)\}=D\{x(n)\}=\{x(n-1)\}$$

Delayed by one sample interval $D=Z^{-1}$

Examples

1. Integrator

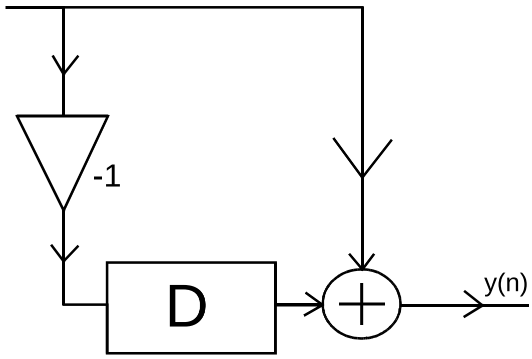
Utilising a feedback path



$$\begin{aligned}y(n) &= x(n)+y(n-1)\\ y(n)-y(n-1) &= x(n)\\ \text{or}(1-D)\{y(n)\} &= \{x(n)\}\end{aligned}$$

2. Differencer

Utilising a feed forward path



$$\begin{aligned} y(n) &= x(n) - x(n-1) \\ \text{or}\{y(n)\} &= (1-D)\{x(n)\} \end{aligned}$$

Nth order recursive filter

$$\{y(n)\} = \{x(n)\}\left[\sum_{m=0}^{N} a(m) D^m\right] + \{y(n)\}\left(\sum_{m=0}^{N} b(m) D^m\right)$$

$$\{y(n)\}(1 - \sum_{m=0}^{N} b(m) D^m) = \{x(n)\}(\sum_{m=0}^{N} a(m) D^m) \qquad \text{-(2.1)}$$

Recall $D^m\{x(n)\} = x(n-m) \leftrightarrow Z^{-m} X(Z)$

i.e: $D \leftrightarrow Z^{-1}$

Hence:

$$Y(Z)\left(1 - \sum_{m=1}^{N} b(m) Z^{-m}\right) = X(Z) \sum_{m=0}^{N} a(m) Z^{-m}$$

Transfer function (TF)

$$H(Z) = \frac{Y(Z)}{X(Z)} = \frac{\sum_{m=0}^{N} a(m) Z^{-m}}{1 - \sum_{m=1}^{N} b(m) Z^{-m}} \qquad \text{-(2.2)}$$

TF describes the behaviour of filter, it is defined as the Z-transform of the output function over the Z-transform of the input filter.

If $X(Z) = 1$ then $Y(Z) = H(Z)$ -(2.3)

Note $X(Z) = 1 \leftrightarrow \{x(n)\} = \underbrace{\{1,0,0,0,0,\ldots\}}_{\text{unit impulse sequence}}$ used to characterises a filter.

$$H(Z) = \sum_{n=0}^{\infty} h(n) Z^{-n}$$

$\{h(n)\}$ is Impulse Response (IT) and characterises the filter.

This is very important, as when we put in the sequence $\{1,0,0,0,\ldots\}$ the (filter) output describes the Z-Transform function of this filter.

Non-recursive filter (no feedback)

Finite number of non-zero elements in the sequence.

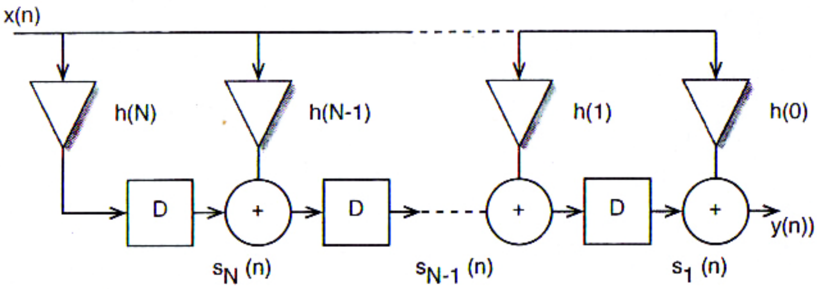
Finite Impulse Response (FIR) filtering $\{y(n)\}=\sum_{m=0}^{N} h(m) D^m \{x(n)\}$ -(2.4)

where $\{h(n)\}$ is IR

Z-transform transfer function $H(Z)=\sum_{m=0}^{N} h(m) Z^{-m}$ -(2.5)

$\{h(n)\}=\{h(0), h(1), h(2), \ldots\}$

Direct non-recursive implementation



- Cononical form, i.e. Number of delays = order of TF
- Finite memory.

$$\begin{aligned} if\ x(n) &= 0 \quad \forall n > M \\ y(n) &= 0 \quad \forall n > M+n \end{aligned}$$

i.e. Response is 0 after N time steps

- also called Moving Average (MA) filter

State Equations

States $s_i(n)$ respresents the past history of a filter:

- States & input at time *n* determine output at time *n* and state at time $(n+1)$.
- A state identifies with a delay.

(see last diagram from previous lecture)

$y(n)=h(0)x(n)+s_1(n)$ -(3.1)

(In the previous diagram) – the right most element is the first element from the input sequence, due to the presence of the delays.

$s_i(n+1)=s_i+1(n)+h(i)x(n)$ where $i<N$ -(3.2)

$S_N(n+1)=h(N)x(n)$ -(3.3)

(3.1) to (3.3) are state equations

(3.1 & (3.2) →

$$\begin{aligned} Y(z) &= h(0)x(z)+s_1(z) \\ zS_i(z) &= h(i)X(z)+S_{i+1}(z) \\ \text{or } S_i(z) &= h(i)z^{-1}X(z)+z^{-1}S_i+1(z) \end{aligned}$$ -(3.4)

From (3.4):

$$\begin{aligned} S_1(z) &= h(1)z^{-1}X(z)+z^{-1}S_2(z) \\ &= h(1)z^{-1}x(z)+z^{-1}[h(2)z^{-1}X(z)+z^{-1}S_3(z)] \\ &= h(1)z^{-1}X(z)+h(2)z^{-2}X(z)+z^{-2}[h(3)z^{-1}X(z)+z^{-1}S(2)] \\ \therefore\ Y(z) &= h(0)X(z)+h(1)z^{-1}X(z)+h(2)x^{-2}X(z)+\ldots+h(N)z^{-NX}(z) \\ &= \sum_{n=0}^{N} h(n)z^{-nX}(z) \end{aligned}$$

(ie same as (2.5))

$$H(z)=\sum_{n=0}^{N} h(n)z^{-n} \qquad \text{-(3.5)}$$

(refer to Fig 1)

Different implementation → different state equations but same TF (Transfer Function)

$$y(n)=h(0)x(n)+\sum_{m=1}^{N} h(m)s_m(n) \qquad \text{-(3.6)}$$

$$S_m(n+1)=s_m-1(n) \qquad m>1 \qquad \text{-(3.7)}$$

$$s_1(n+1)=x(n) \qquad \text{-(3.8)}$$

Zeros

Roots of $H(z)=0$

Multiplying (3.5) by Z^N gives

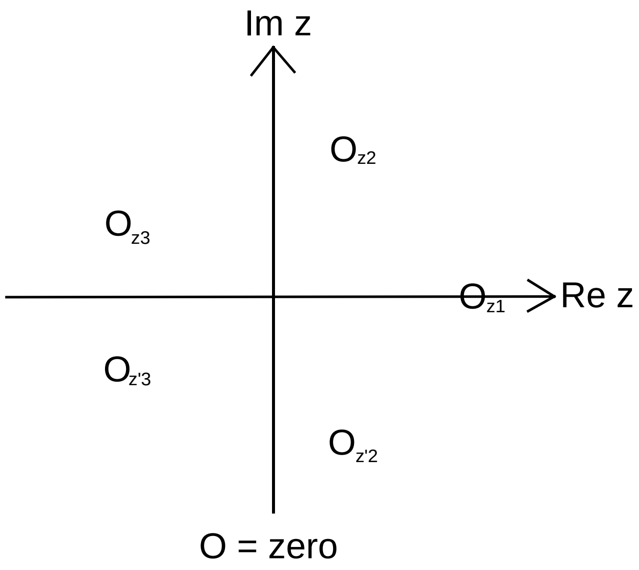
$$Z^N H(z)=\sum_{n=0}^{N} h(n)z^{N-n}$$

$$=\prod_{m=1}^{N}(z-z_m) \qquad \text{- (3.9)}$$

where $z_m=|z_m|\exp(j\arg[z_m])$ is the m-th zero of TF

Examples

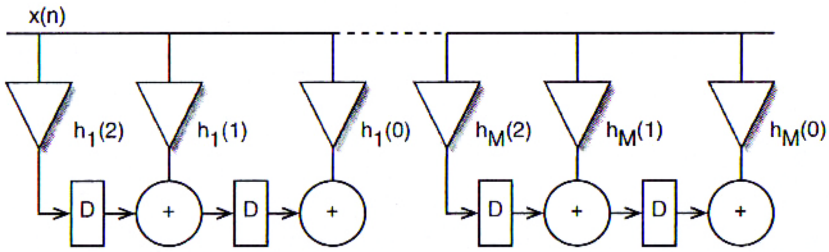
Z is a complex number, so it has a real part and an imaginary part.



This is a pictorial representation of a Z-plane,

$$(3.9) \Rightarrow H(z)=\prod_{m=1}^{M}(h_m(0)+h_m(1)Z^{-1}+h_m(2)Z^{-2}) \qquad \text{-(3.10)}$$

$\rightarrow$software modularity



This allows software modularity due to the break up of components, as shown in the diagram.

Stability of FIR filter

If $\{x(n)\}<X, \forall n$ gives rise to $\{y(n)\}<Y, \forall n$ then filter output is Bounded Input Bounded Output (BIBO) stable.

$|x(n)|<X \qquad 0 \le n \le \infty$

Since $y(n)=\sum_{m=0}^{N} h(m)x(n-m)$ $\qquad\leftarrow$convolution

$$|y(n)|=\left|\sum_{m=0}^{N} h(m)x(n-m)\right|<X\left|\sum_{m=0}^{N} h(m)\right|=HX$$

$\therefore$ FIR filters are BIBO stable

Recursive filters

$$(1-\sum_{m=1}^{N} b(m)D^m)\{y(n)\}=\sum_{m=0}^{N} a(M)D^m\{x(n)\}$$

where $a(m)$ and $b(m)$ are filter coefficients

TF $\qquad H(z)=\frac{Y(z)}{X(z)}=\frac{\sum_{m=0}^{N} a(m)z^{-m}}{1-\sum_{m=1}^{N} b(m)z^{-m}}$ $\qquad$ -(4.1) (same as 2.2)

$=\sum_{n=0}^{N} h(n)z^{-n}$

If $(1-\sum_{m=1}^{N} b(m)z^{-m})\neq 1$ then IR is infinite

$\Rightarrow$ infinite impulse response (IIR) filter.

If $a(m)=0$, $0<m\le N$

$\Rightarrow$ all pole filter or AutoRegressive (AR) filter

State equations:

$y(n)-s_1(n)+a(0)x(n)$ $\qquad$ -(4.2)

$S_i(n+1)=s_{i+1}(n)+a(i)x(n)+b(i)y(n), \quad i<N$ $\qquad$ -(4.3)

$S_N(n+1)=a(N)x(n)+b(N)y(n)$ -(4.4)

Substitution for $y(n)$ from (4.2) in (4.3):
$s_i(n+1)=S_{i+1}(n)+b(i)s_1(n)+[a(i)+a(0)b(i)]x(n)$ -(4.5)

Expressing the state equations in matrix form:
$\underline{S}(n+1)=\underline{A}\,\underline{S}(n)+\underline{B}\,x(n)$ -(4.6)
$y(n)=\underline{C}\,\underline{S}(n)+\underline{D}\,x(n)$ -(4.7)
where underlining denotes a matrix.

<u>A</u> is an $N\times N$ matrix:

$$\underline{A}=\begin{bmatrix} b(1) & 1 & 0 & 0 & \dots & 0 \\ b(2) & 0 & 1 & 0 & \dots & 0 \\ b(3) & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b(N-1) & 0 & 0 & 0 & \dots & 1 \\ b(N) & 0 & 0 & 0 & \dots & 0 \end{bmatrix}$$ -(4,8)

<u>S</u> is an $N\times 1$ matrix:

$$\underline{S}=\begin{bmatrix} s_1(n) \\ s_2(n) \\ s_3(n) \\ \vdots \\ s_N(n) \end{bmatrix}$$ -(4.9)

<u>B</u> is $N\times 1$ matrix

$$\underline{B}=\begin{bmatrix} a(1)+a(0)b(1) \\ a(2)+a(0)b(2) \\ \vdots \\ a(N)+a(0)b(N) \end{bmatrix}$$ -(4.10)

<u>C</u> is a $1\times N$ matrix:
$\underline{C}=(1\,0\,0\cdots 0)$ -(4.11)
$\underline{D}=(a(0))$

Examples:

(refer to fig)
using (4.8) to (4.11)

$$\Rightarrow \underline{S}(n+1)=\begin{bmatrix} 2 & 1 \\ -1 & 0 \end{bmatrix}\underline{S}(n)+\begin{bmatrix} 1 \\ -3 \end{bmatrix}x(n)$$

$$y(n)=[1 \quad 0]\underline{S}(n)+x(n)$$

From the filter block diagram and using (4.1)

$$H(z)=\frac{1-z^{-1}-2z^{-2}}{1-2z^{-1}+z^{-2}}$$

Poles & zeros of IIR filter

Writing (4.1) in factor form

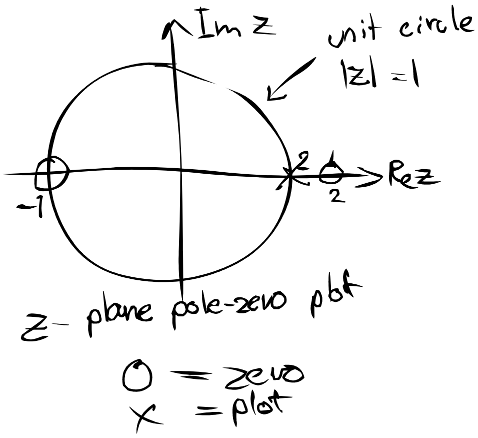
$$H(z)=\frac{a(z)}{B(z)}=\frac{\prod_{n=1}^{N}(z-\alpha_n)}{\prod_{n=1}^{N}(z-\beta_n)} \qquad \text{-(4.12)}$$

Where

α_n = zeros of TF
β_n = poles of TF

Example

$$H(z)=\frac{(z-2)(z+1)}{(z-1)^2}$$



A filter is BIBO stable is all pf ots poles are within the unit circle. i.e. $|\beta_m|<1$
Reason:

Suppose: $TF=H(z)=\frac{1}{1-z^{-1}\beta_m}\leftrightarrow\{h(n)\}=\{1,\beta_m,\beta_m^2,\ldots,\beta_m^n\}$

If $|\beta_m|<1$ the sequence converges to zero.
$|\beta_m|>1$ - unstable
$|\beta_m|=1$ - conditionally stable
FIR filters have no poles (no denominator is TF) – always BIBO stable

Poles and Eignvalues of state transition matrix

The eignvalues of the transition matrix $\underline{A}$ om (4.8) are found by solving the characteristic equation:
$det[\underline{A}-\lambda\underline{I}]=0$

Suppose $\underline{A}=\begin{bmatrix} b(1) & 1 & 0 \\ b(2) & 0 & 1 \\ b(3) & 0 & 0 \end{bmatrix}$

$$\underline{A}-\lambda\,\underline{I}=\begin{bmatrix} b(1)-\lambda & 1 & 0 \\ b(2) & -\lambda & 1 \\ b(3) & 0 & -\lambda \end{bmatrix}$$

$$det[\underline{A}-\lambda\,\underline{I}]=\lambda^3+\lambda^2 b(1)+\lambda\, b(2)+b(3)$$
$$=\lambda^3-(b(1)\lambda^{3-1}+b(2)\lambda^{3-2}+b(3)\lambda^{3-3})$$
$$=\lambda^3-\sum_{m=1}^{3} b(m)\lambda^{3-m}$$

For $N \times N$ matrix $\underline{A}$:

$$det[\underline{A}-\lambda\,\underline{I}]=\lambda^N-\sum_{m=1}^{N} b(m)\lambda^{N-m}=0$$

Substitute z for λ

$$z^n-\sum_{m=1}^{N} b(m)z^{N-m}=0$$
$$z^n-z^n\sum_{m=1}^{N} b(m)z^{-m}=0$$
$$z^n(1-\sum_{m=1}^{N} b(m)z^{-m})=0$$
$$z^n B(z)=\prod_{m=1}^{N}(z-z_m)=0$$

i.e. Eignvalues of $\underline{A}$ are poles of TF

ARMA realisation of IIR filter

Comprises an all-pole (AR) filter in cascade with ian FIR (MA) filter as in Fig.
State equations:

$s_i(n_1)=s_{i-1}(n) \qquad 1<i<N$ -(5.3)

$s_i(n+1)=y_1(n)$ -(5.4)

where $n_{¿}\, x(n)-\sum_{m=1}^{N} b(m)s_m(n)$ -(5.5)

$y_1 ¿$

$$y(n)=a(0)\,y_1(n)+\sum_{m=1}^{N} a(m)s+m(n) \qquad \text{-(5.6)}$$

$$=a(0)\,x(n)+\sum_{m=1}^{N}(a(m)+a(0)b(m))\,s_m(n) \qquad \text{-(5.7)}$$

In matrix from:

$\underline{s}(n+1)=\underline{A}\,\underline{s}(n)+\underline{B}\,x(n)$

$y(n)=\underline{C}\,\underline{s}(n)+\underline{D}\,x(n)$ -(5.9)

$$\text{where } \underline{A}=\begin{bmatrix} b(1) & b(2) & b(3) & \cdots & b(N) \\ 1 & 0 & 0 & \cdots & \cdots \\ 0 & 1 & 0 & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad \text{-(5.10)}$$

$\underline{B}=[1\,0\,0\ldots0]^T$ -(5.11)

$\underline{C}=[a(1)+a(0)b(1) \quad a(2)+a(0)b(2)\ldots \quad a(N)+a(0)b(N)]$ -(5.12)
$\underline{D}=[a(0)]$ -(5.13)
In the z domain

from (5.6), $Y(z)=\sum_{m=0}^{N} a(m)z^{-m}Y_1(z)$

from (5.5), $Y(z)=X(z)+\sum_{m=1}^{N} b(m)z^{-m}Y_1(z)$

$$\Rightarrow Y_1(z)=\frac{X(z)}{1-\sum_{m=1}^{N} b(m)z^{-m}}$$

$$TF=\frac{Y(z)}{X(z)}H(z)=\frac{\sum_{m=0}^{N} a(m)z^{-m}}{1-\sum_{m-1}^{N} b(m)z^{-m}}$$ same as (4.1)

Alternative realisations

Examples:

$$H(z)=\frac{1-z^{-1}-2x^{-2}}{1-2x^{-1}+z^{-2}}$$

c.f. (5.14)

$\Rightarrow a(0)=1, \quad a(1)=-1, \quad a(2)=2, \quad b(1)=2, \quad b(2)=-1$

Using (5.8) + (5.13)

$$\underline{s}(n+1)=\underbrace{\begin{bmatrix}2 & -1\\ 1 & 0\end{bmatrix}}_{\text{matrix A}}\underline{s}(n)+\underbrace{\begin{bmatrix}1\\ 0\end{bmatrix}}_{\text{matrix B}}x(n)$$

$$y(n)=\underbrace{[1 \quad 3]}_{\text{matrix C}}\underline{s}(n)+\underbrace{[1]}_{\text{matrix D}}x(n)$$

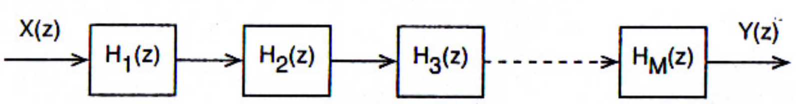
Serial and parallel implementaions

Serial implementation by factoring $H(z)$ into 2nd-order factors:

$$H(z)=\prod_{m=1}^{M} H_m(z) \qquad \text{-(6.1)}$$

where $$H_m(z)=\frac{\prod_{n=1}^{2}(z-\alpha_m(n))}{\prod_{n=1}^{2}(z-\beta_m(n))}$$

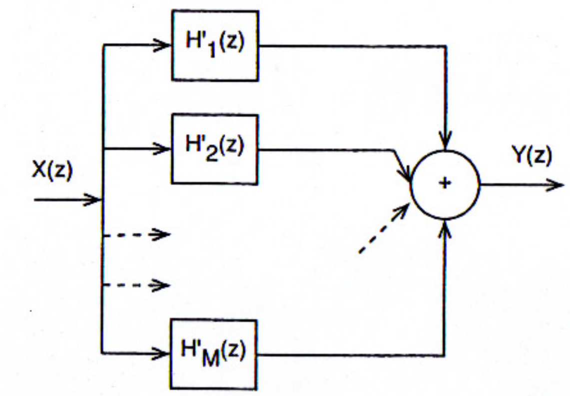
(6.1): You factor $H(z)$ into multiple smaller order functions $H_m(z)$. This also gives rise to software modularity when implemented in software.
The corresponding serial implementation is shown thus:



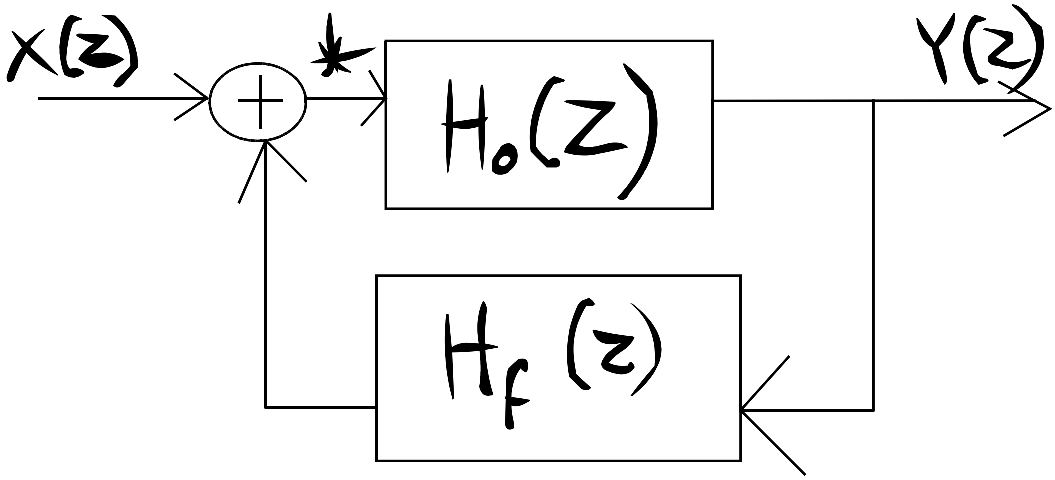
Parallel implementation by partial fraction expansion of $H(z)$:

$$H(z)=\sum_{m=1}^{M} H'_m(z) \qquad -(6.2)$$

where $H'(z)$ is 2nd order rational function.
Offers speed advantage (due to parallelism) but more complex.



Block diagram manipulation



$$Y(z)=H_0(z)\overbrace{(x(z)+H_f(z)Y(z))}^{*}$$
$$Y(z)(1-H_0(z)H_f(z))=H_0(z)X(z)$$
$$H(z)=\frac{Y(z)}{X(z)}=\frac{H_0(z)}{1-H_0(z)H_f(z)}$$

Frequency response (FR)

RF – filter's output amplitude and phase when the input is a sinusoid of the specified frequency and unit magnitude
Define $\{e_\omega(n)\}$ as the samples complex exponential signal

$$e_\omega(n)=\exp[jn\omega] \qquad -(6.3)$$

where $\omega=2\pi f$ is radial frequency
$j=\sqrt{-1}$
N.B.

$$e_{\omega+2\pi m}(n)=\exp[jn\omega+2\pi m]$$
$$=\exp[jn\omega]\exp[j\pi 2mn]$$
$$=\exp[jn\omega](-1)^{2mn}$$
$$=e_{\omega}(n) \qquad \text{-(6.4)}$$

i.e. Periodic

$$E_{\omega}(z)=\frac{1}{1-\exp[j\omega]z^{-1}} \qquad \text{-(6.5)}$$

i.e. Pole at $z=e^{j\omega}$

Suppose a filter with IR $\{h(n)\}$ and input is $\{s(n)\}$ then output

(* - convolution)

$$y(n)=h(n)*s(n) \qquad \text{-(6.6)}$$
$$=\sum_{m=0}^{n} h(m)s(n-m) \qquad \text{-(6.7)}$$
$$=\sum_{m=0}^{n} h(m)\exp[j(n-m)\omega]$$
$$=e_{\omega}(n)\sum_{m=0}^{n} h(m)\exp[-jm\omega]$$

Let $n\to\infty$

$$y_{\infty}(n)=e_{\omega}(n)\sum_{m=0}^{\infty} h(m)\exp[-jm\omega]$$
$$=s(n)H[\exp[-j\omega]]$$

i.e. Input weighted by a complex number

If you imagine that $z=\exp[-j\omega]$, then in fact we have a z-transform for the IR.

$$H(\omega)\overset{\circ}{=}H[\exp[-j\omega]]=\sum_{m=0}^{\infty} h(m)\exp[-jm\omega] \qquad \text{-(6.8)}$$
$$=s(n)\underbrace{H[\exp(j\omega)]}_{H(\omega)}$$

or more generally

FR $\quad H(\omega)=\sum_{m=-\infty}^{\infty} h(m)\exp[-jm\omega]$ -(6.9)

Filter must be stable to have FR

Normally

$$H(\omega)=|H(\omega)|\exp[jn\omega] \qquad \text{-(7.1)}$$

i.e. Magnitude response $|H(\omega)|$ and phase response $\exp[jn\omega]$

$F(\omega)$ is periodic, i.e. $H(\omega+2\pi m)=H(\omega)$, where $m=0,1,2,\dots$ -(7.2)

N.B. RHS of (7.1) is Fourier series of $H(\omega)$

$$\Rightarrow h(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi} H(\omega)\exp[jn\omega]d\omega \qquad \text{-(7.3)}$$

Substitute (7.1) into RHS of (7.3) and use

$$\int_{-\pi}^{\pi} \exp[j(n-\omega)\omega]d\omega = 2\pi\delta mn \qquad \text{-(7.4)}$$

to show (7.3)

FR, poles and zeros

Substituting $z = \exp[j\omega]$ in (4.12)

$$\text{FR} \rightarrow H(\omega) = \frac{\prod_{n=1}^{N}(\exp[j\omega] - \alpha_n)}{\prod_{n=1}^{N}(\exp[j\omega] - \beta_n)} \qquad \text{-(7.5)}$$

$$= \frac{\prod_{n=1}^{N} A_n(\omega)}{\prod_{n=1}^{N} B_n(\omega)} \qquad \text{-(7.6)}$$

Where α_n - nth zero

β_n – nth pole

i.e. FR depends on poles & zeros

Writing α_n = $|\alpha_n|\exp[j\theta n]$, the nth zero makes a contribution with a magnitude

$$|A_n(\omega)|^2 = 1 + |\alpha_n|^2 - 2|\alpha_n|\cos[\omega - \theta_n] \qquad \text{-(7.7)}$$

minimum at $\omega = \theta_n$, maximum at

$\omega = \theta_n + \pi$

Similarly, the contribution of a pole at the angle of pole and minimum at the opposite angle.

i.e. A pole “amplifies” reponse while a zero “attenuates” response

Useful to calculate $H(\omega)$ at frequencies $\omega = 0$, $\frac{\pi}{2}$, π.

The contribution of a pole to the repsonse grows as $|\beta_n| \rightarrow 1$ (where magnitude is measured from origin of pole-zero plot)

A zero on the unit circle implies a null response at that frequency at the zero.

Example

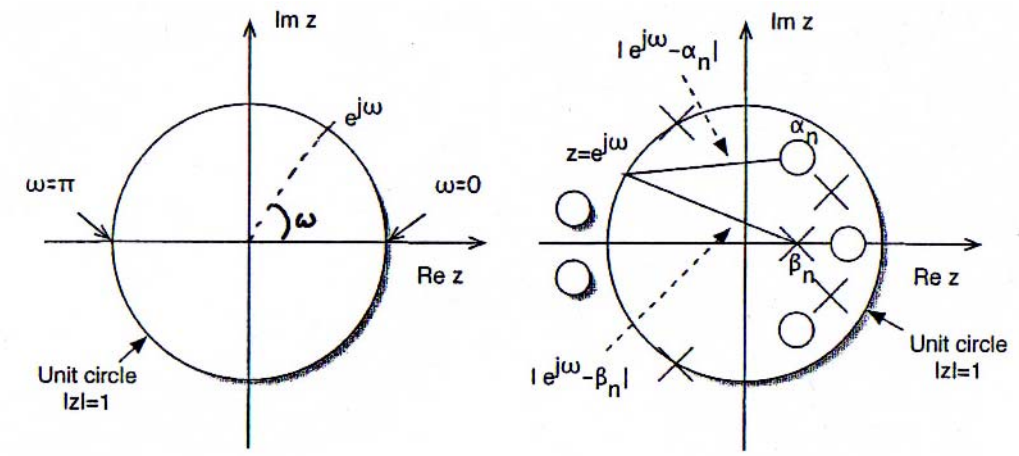
$$G(z) = \frac{1 + 0.1z^{-1} - 0.72z^{-2}}{1 + 0.92z^{-1} + 0.81z^{-2}} \quad \left(\times \frac{z^2}{z^2}\right)$$

$$\Rightarrow G(\omega) = \frac{1 + 0.1e^{-j\omega} - 0.72e^{-2j\omega}}{1 + 0.9e^{-j\omega} + +0.81e^{-2j\omega}}$$

$$G(z) = \frac{z^2 + 0.1z - 0.72}{z^2 + 0.8z + 0.81}$$

$$zeros: z = -0.9, 0.8$$

$$poles: z = -0.45 \pm j0.779$$



Near the zeros we expect little reponse, but as we go around the circumference (frequencies) and approach a pole, the reponse increases.

$\omega=0 \quad G(\omega)=0.14$

$\omega=\pi \quad G(\omega)=0.198$

$\omega=\frac{\pi}{2} \quad G(\omega)=1.87$

$\omega=\frac{\pi}{4} \quad G(\omega)=0.57$

$\omega=\frac{3\pi}{4} \quad G(\omega)=3.03$

$\omega=2.1 \quad G(\omega)=9.05$

Filter Types

1. Lowpass (LP)

(see diag L8-3-1)

ω_c – cutoff frequency
0 to ω_c – passband
ω_c to π – stopband

Example:

(see diag L8-3-2)

$$H(z)=0.103\times\frac{1+1.22z^{-1}+0.52^{-2}}{1-1.22z^{-1}+0.52^{-2}}$$

poles at z = $\frac{1}{\sqrt{x}}e^{\pm j\frac{\pi}{6}}$, zeros at $z=\frac{1}{\sqrt{2}}e^{\pm j\frac{5\pi}{6}}$

2. Highpass (HP)

(see diag L8-3-3)

Obtained from a lowpass filter by $H_{hp}(z)=1-H_{lp}(z)$ in $H_{lp}(z)$

$\Rightarrow H(\omega-5\,pi)=H(e^{j(\omega-5\,pi)})$ (since $e^{j\pi}=-1$)

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

Basically we shift a lowpass filter by π.

Example:

$$H(z)=0.103\times\frac{1-1.22z^{-1}+0.52z^{-2}}{1+1.22z^{-1}+0.52z^{-2}}$$

(Lecture 8-1 Bottom)

3. Bandpass

(see diag L8-4-1)

$\omega_1-\omega_0$is passband

$\frac{(\omega_1-\omega_0)}{2}$ = centre frequency

$\frac{(\omega_1-\omega_0)}{(\omega_1+\omega_0)}$= fractional bandwidth

Example:

(see diag L8-4-2)

$$H(z)=0.103\times\frac{1-0.81z^{-2}}{1+0.81z^{-2}}$$

4. Bandstop

(see diag L8-4-3)

Combine a lowpass and highpass frequencies response, or shifting a bandpass frquency response.

Bandstop is the incerse of a bandpass filter.

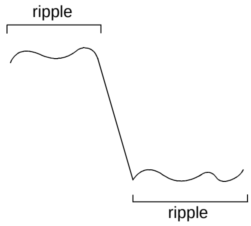
Example:

$$H(z)=0.103\times\frac{1+0.81z^{-2}}{1-0.81z^{-2}}$$

(Lecture 8-2 Bottom)

Filter Design

1. A recursive filter of a given order is generally better than a non-recursive; but at the expense of non-linear phase response.
   Not only does this cause delay – it also distorts some of the frequencies.
2. Finite stopband attentuation – will not have a null response to frequencies outsite its passband.
3. Finite transition bandwidth no discontinuties response
4. Non-uniform response in the passband – passband ripple



Processing analogue signals

(see diag Lecture 9-1 Top)
Define frequency content of a continuous signal by its Fourier transform (FT)

$$X(\omega)=\int_{-\infty}^{\infty} x(t)\exp[-j\omega t]\, ft \qquad \text{-(9.1)}$$

with inverse

$$X(t)=\int_{-\infty}^{\infty} \exp[j\omega t]\, d\omega \qquad \text{-(9.2)}$$

N.B $X(\omega)$ is non-periodic

FT or spectrum of a samples signal $\{x_1(n)\}$ is periodic in frequency with period $\frac{2\pi}{T}$

$$X_1(\omega)=\sum_{n=-\infty}^{\infty} x_1(n)\exp[-jn\omega T] \qquad \text{-(9.3)}$$

where T is the sampling interval.
T and m are usually interoperable, and are both intended to represent the sampling interval. We use T to be consistent with the rest of the module.
(see diag lecture 9-1 bottom)
I.E. $X_1(\omega)$ and $X_2(\omega)$cannot approximate a non-periodic $X(\omega)$ exept within one period
(see diag lecture 9-2 Top)

For a given T, $X(\omega)$ outsite $|\omega|\le\frac{\pi}{T}$ is fixed by T, so choose samples to minimise.

$$E=\frac{1}{2\pi}\int_{\frac{-\pi}{T}}^{\frac{\pi}{T}} |X(\omega)-X_1(\omega)|^2\, d\omega \qquad \text{-(9.4)}$$

$$E=\frac{1}{\pi}\int_{\frac{-\pi}{T}}^{\frac{\pi}{T}}\left[|X(\omega)|^2-x(\omega)\sum_{n=-\infty}^{\infty} x_1(n)e^{jn\omega T}-X*(\omega)\sum_{m=-\infty}^{\infty} x_1(m)e^{-jm\omega T}+\sum_{m,n=-\infty}^{\infty} x_1(n)x_1(m)e^{j(n-m)\omega T}\right]d\omega$$

Minimise E w.r.t $x_1(k)\Rightarrow$

$$\frac{2E}{2x_1(K)}=\frac{2}{T}=x_1(K)-\frac{1}{\pi}\int_{\frac{-\pi}{T}}^{\frac{\pi}{T}} x(\omega)e^{-jK\omega T}\, d\omega \qquad \text{-(9.6)}$$

using $\int_{\frac{-\pi}{T}}^{\frac{\pi}{T}} e^{-j(m-n)T}\, d\omega=\frac{2\pi}{T}\delta mn$

and $X*(\omega)=X(-\omega)$
Since RHS of (9.6) = 0 for turning point.

$$X_1(k)=\frac{T}{2\pi}\int_{\frac{-\pi}{T}}^{\frac{\pi}{T}} x(\omega)e^{jK\omega T}\,d\omega \qquad \text{-(9.7)}$$

i.e. Signal bandlimited to $|\omega|\le\frac{\pi}{T}$

$$X_1(omerga)=B_{\frac{\pi}{T}}(\omega)\,X(\omega) \qquad \text{-(9.8)}$$

where $B_{\frac{\pi}{T}}(\omega) \quad =1 \quad |\omega|\le\frac{\pi}{T}$

$\quad =0 \quad \text{else}$

$B_{\frac{\pi}{T}}(\omega)$ - a LP filter with ω_c of half the sampling frequency $\left(2\frac{\pi}{T}\right)$

- called anti-aliasing filter.

Bandlimiting error $E_b=\frac{1}{\pi}\int_{\frac{\pi}{T}}^{\infty}|X(\omega)|^2\,d\omega$ -(9.9)

-quite significant if sampling frequency is not high enough.

So, $\{x_1(n)\}$ is obtained by:

1. Bandlimiting $x(t)$ to half the sampling frequency $\left(\frac{2\pi}{T}\right)$
2. Sampling $x(t)$ at $\frac{2\pi}{T}$

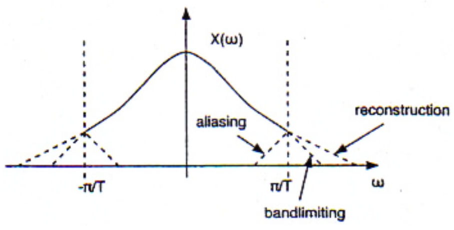
It can be shown that if a signal is not badnlimited then its spectrum comprises the original spectrum $X(\omega)$ plus aliasing spectrum

$$\sum_{\substack{m=-\infty\\ m\neq 0}}^{\infty} X\left(\omega-\frac{2\pi m}{T}\right)$$

Aliasing creates error in reconstruction (see diag lecture 9-2-middle)

Perfect reconstruction is not possible due to imperfect bandlimiting of signal.
User oversampling to minimise problem:

e.g. 5MHz for audio instead of 50kHz



Sub-band Filters and Wavet Transform

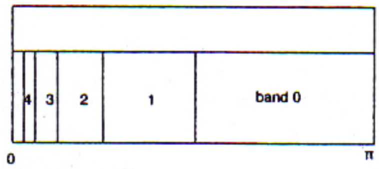
Divide frequency domain into a number of non-overlapping frequency bands.

Applications:

- Data compression of audio and images

- Computer vision

Logarithmic frequency scale:
- Bands of constat size in log f, i.e. Constant octave bandwidth.



Octave sub-band filters

Reducing bandwidth by half with each split ⇒ halving the sampling rate with each split.For a sequence $\{x(n)\}$ retain every second sample to get

$$x_1(n)=\begin{cases} x(n) & n=2m \\ 0 & n=2m-1 \end{cases} \qquad \text{-(10.1)}$$

$$\rightarrow\{x_1(n)\}=\frac{1}{2}\left(\{x(n)\}+\{(-1)^n x(n)\}\right) \qquad \text{-(10.2)}$$

$$X_1(z)=\frac{1}{2}(X(z)-X(-z)) \qquad \text{-(10.3)}$$

$X(-z)$ centred at sampling frequency $\omega=\pi$, causes alaising.
Let $H(z)$ – Low Pass (LP) filter Transfer Function (TF)
& $C(z)$ – input

$$C_1(z)=\frac{1}{2}(H(z)C(z)+H(-z)C(-z)) \qquad \text{-(10.4)}$$

Interpolate $C_1(z)$ to give the same sampling rate as original signal to generate a bandpass signal, and then generate

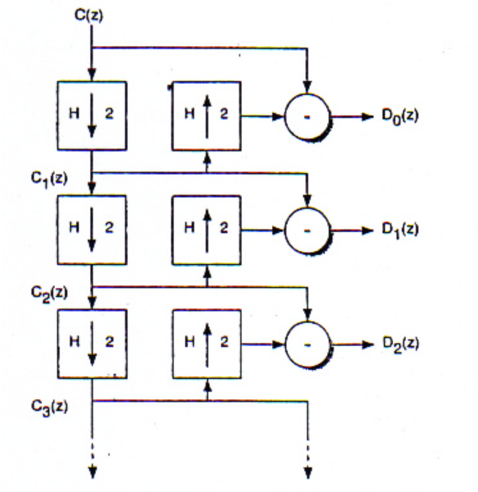
$$D_0(z)=C(z)-2H(z)C_1(z) \qquad \text{-(10.5)}$$

where the factor is for normalisation.

$$C(z)=D_0(z)+2H(z)C_1(z) \qquad \text{-(10.6)}$$

$D_0(z)$ removes aliasing.

Tree scheme for actave sub-band decomposition



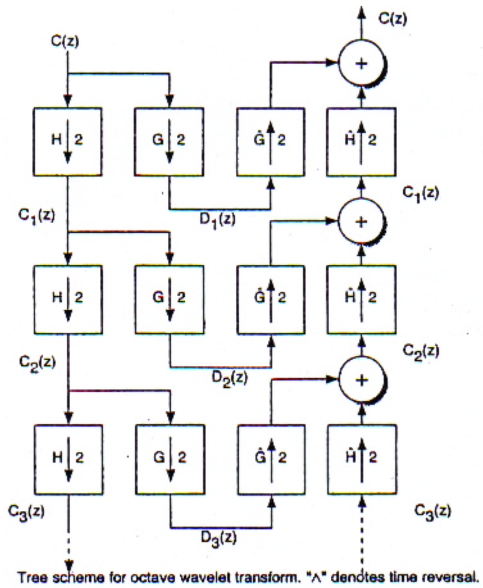
Tree scheme for octave sub-band decomposition. Downward arrows denote decimation; upwards arrows interpolation.

Key: $H(z)$ = LP TF; ↓ – subsampling; ↑ - interpolation
General stage: $D_i(z)=C_i(z)-2H(z)C_{i+1}(z)$ -(10.7)

Example:

Laplacian pyramid using $\{h(n)\}=\{0.05, 0.25, 0.4, 0.25, 0.05\}$
-such schemes are scale space:
higher levels of pyramid (or tree) contain signals which are scaled versions of the original
-based on subsampling by α, i.e. Dyadic

Tree scheme for octave wavelet transform



Tree scheme for octave wavelet transform. "^" denotes time reversal.

Input and output at stage i are $C_i(z)$
$H(z)$ indicates Low Pass (LP) filter; $G(z)$ indicated High Pass (HP) filter
$C_0(z)=C(z)$ – input to system

At stage *i*: $C_{i+1}(z)=\frac{1}{2}(H(z)C_i(z)+H(-z)C_i(-z))$ -(10.8)

$D_{i+1}(z)=\frac{1}{2}(G(z)C_i(z)+G(-z)C_i(-z))$ -(10.9)

Interpolating with the corresponding time-reverse filters

$E_i(z)=2H(z^{-1})C_{i+1}(z), \quad F_i(z)=2G(z^{-1})D_{i+1}(z)$ -(10.10)

Output at state *i*

$G_i(z)=E_i(z)+F_i(z)$

$=C_i(z)(H(z)H(z^{-1})+G(z)G(z^{-1}))+C_i(-z)(H(-z)H(z^{-1})+G(-z)G(z^{-1}))$ -(10.11)

For $G_i(z)=C-i(z)$ independently of $C_i(z)$

$$\left.\begin{array}{r} H(z)H(z^{-1})+G(z)G(z^{-1})=1 \\ H(-z)H(z^{-1})+G(-z)G(z^{-1})=0 \end{array}\right\}$$ -(11.1)

which is satisfied using

$G(z)=(-z)^{1-N}H(z^{-1})$ -(11.2)

(conjugate quadrature filters or quadrature mirror filters)

Substituting (11.2) into 1st equation of (11.1) gives

$H(z)H(z^{-1})+H(-z)H(-z^{-1})=1$

E.g. Daubechies: $\{h(n)\}=\frac{1}{8}\{1+\sqrt{3},\ \ 3+\sqrt{3},\ \ 1-\sqrt{3}\}$

$H(z)$ should be zero at π to minimise aliasing for image compression.

Matched Filtering

To detect known signal against a background of distrubances.

Let signal $\{x(n)\}$ & filter IR $\{h(n)\}$.

Filter output $y(m)=\sum_{n=-\infty}^{\infty} h(n)x(m-n)$ -(11.3)

Maximise $y(0)=\sum_{n=-\infty}^{\infty} h(n)x(-n)$ -(11.4)

We want to maximise the response of our filter in the presence of a known signal.

Constrain filter coefficients to have finite energy

$\sum_{n=-\infty}^{\infty} h^2(n)=1$ -(11.5)

Hence we have a problem of maximisation.

Use Lagrange multiplier λ to maximise

$E(h)=y(0)-\lambda(\sum_{n=-\infty}^{\infty} h^2(n)-1)$ -(11.6)

The bracketed part above is a contrainst

$\frac{\delta E(n)}{\delta h(n)}=x(-n)-2\lambda h(n)=0$

$\rightarrow h(n)=\frac{x(-n)}{2\lambda}$ -(11.7)

We differentiate (11.4) and (11.5) in order to obtain an equation which allows us to maximise.

Using (11.5)

$$\Rightarrow \lambda \frac{1}{2}\sqrt{\sum_{n=-\infty}^{\infty} x^2(n)}$$

Using (11.7)

$$\Rightarrow h(n) = \frac{x(n-1)}{\sqrt{\sum_{n=-\infty}^{\infty} x^2(n)}}$$

$$= \epsilon\, x(-n) \qquad \text{-(11.8)}$$

where $$\epsilon = \frac{1}{\sqrt{\sum_{n=-\infty}^{\infty} x^2(n)}} \qquad \text{-(11.9)}$$

I.e., there matched filter IR is the time-reverse of the signal sequence.

For this $h(n)$,

$$y(0) = \epsilon \sum_{n=-\infty}^{\infty} x^2(n)$$

$$= \frac{1}{\epsilon} \qquad \text{-(11.10)}$$

Let normalising constant $\epsilon = 1$ then using (11.8) and (11.3)

$$y(n) = \sum_{m=-\infty}^{\infty} x(-m)\, x(n-m)$$

$$= \sum_{m=-\infty}^{\infty} x(m)\, x(m+n)$$

The autocorrelation sequence of signal $\{x(n)\}$ is $r_x(n) = \sum_{m=-\infty}^{\infty} x(m)\, x(m+n)$ -(11.12)

From property 2

$$r_x(n) = \sum_{m=-\infty}^{\infty} x(m)\, x(m-n) \qquad \text{-(11.2A)}$$

Properties of $r_x(n)$.

1. $r_x(0) > r_x(n), \quad n \neq 0$
2. $r_x(-n) = r_x(n)$

It is maximum at time 0

It is symmetrical about the time 0

Note:
$$\{x(N-n)\} \;\leftrightarrow\; \sum_{n=0}^{N} x(N-n)\, z^{-n}$$

$$= \sum_{n=0}^{N} x(n)\, z^{m-N}$$

$$= z^{-n} X(z^{-1})$$

i.e. $\{x(-n)\} \leftrightarrow X(z^{-1})$ (when $N=0$)

N.B. ZT of $(x(n) * y(n)) = X(z)Y(z)$

$\therefore R_x(z)=X(z)X(z^{-1})$ -(11.3)

Define cross-correlation of $\{x(n)\}$ and $\{y(n)\}$ as

$$r_{xy}(n)=\sum_{m=-\infty}^{\infty} x(m)y(m-n) \qquad \text{-(12.1)}$$

$$r_{xy}(-n)=r_{yx}(n) \qquad \text{-(12.2)}$$

$$R_{xy}(z)=\sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} x(m)z^{-m}y(m-n)z^{m-n}$$

$$=\sum_{m=-\infty}^{\infty} x(m)z^{-m}\sum_{m'=-\infty}^{\infty} y(m')z^{m'}$$

$$=X(z)Y(z^{-1}) \qquad \text{-(12.3)}$$

$$r_{xy}(n)=\sum_{m=-\infty}^{\infty} x(m)y(m-n)$$

$$=\sum_{m=-\infty}^{\infty} x(m)\acute{y}(n-m)$$

$$=x(n)*\acute{y}(n) \quad where\ \acute{y}=y(-n)$$

i.e. Cross-correlation is convolution with a time-reversed sequence.

In practice $h(n)=\epsilon x(N-m)$ -(12.4)

(fot finite sequences) and maximum output at time N

$$r_x(N-n)=\sum_{m=0}^{N} x(m)x(m+N-n) \quad where\ \epsilon=1$$

$$=\frac{1}{\epsilon}\sum_{m=0}^{N} x(m)x(m+N-n) \quad where\ \epsilon\neq 1 \qquad \text{-(12.5)}$$

Example: signal $\{x(n)\}=\{1,2,3\}$; matched filter $\{h(n)\}=\{3,2,1\}$

fitler output: $y(n)=\sum_{m=0}^{2} h(m)x(n-m)$

3	2	1	
1			3
2	1		8
3	2	1	14
	3	2	8
		3	3

i.e. $\{y(n)\}=\{3,8,14,8,3\}$

Characterising random signals

A noise sequence $\{v(n)\}$ is a sample from a stochastic process, a process which generates a set of sequences in such a way that their statistical properties are consistent across the set and, in all cases of practical interest, directly related to averages which can be measured on an individual sequence. Such processes are ergodic.

First order statistics:

$$\left.\begin{array}{l} \text{probability density } p_v(u)\geq 0 \quad -\infty<u<\infty \\ \int_{-\infty}^{\infty} p_v(u)\,du=1 \end{array}\right\} \qquad \text{-(12.6)}$$

Assume stationary process:

$p_v(u)$ is independent of the time origin and hence $\underset{\wedge}{is}$ not a function of n. The expectation of mean of $v(n)$ is

$$\mu_v \triangleq E[v(n)] = \int_{-\infty}^{\infty} u p_v(u)\, du \qquad \text{-(12.7)}$$

($\triangleq$ seems to me "equivalent to")

$$E[v(n)] = E[v(m)], \forall m, n \qquad \text{-(12.8)}$$

If process is erogdic.

$$E[v(n)] = \bar{v} = \frac{1}{N}\sum_{m=0}^{N-1} v(n-m) \qquad \text{-(12.9)}$$

Variance $\sigma \triangleq E\left[v^2(n) - E^2[v(n)]\right]$

$$= \int_{-\infty}^{\infty} (u-\mu)^2 p_v(u)\, du \qquad \text{-(12.10)}$$

- indicates spread of density about its mean

Example: Uniform density.

$$\begin{aligned} p_v(u) &= \frac{1}{2W} && |u| \leq W \\ &= 0 && else \end{aligned}$$

$\mu_v = 0$

$$\sigma_v^2 = \int_{-W}^{W} (u-\mu_v)^2 p_v(u)\, du$$

$$= \int_{-W}^{W} u^2 \frac{1}{2W} du$$

$$= \frac{W^2}{3}$$

2nd order statistics: Autocorrelation sequence $\{R_v(n)\}$ is

$$R_v(n) = E[v(m)v(m-n)] \qquad \text{-(13.1)}$$

$\{R_v(n)\}$ is symmetrical and a maximum at the origin.

ZT of $\{R_v(n)\}$ is

$$S_v(z) = \sum_{n=-\infty}^{\infty} R_v(n) z^{-n}$$

Autocorrelation of a process descibes how similar it is to its time shift. If the process has non-zero mean, however, a large autocorrelation value will result even if the variation of the signal is completely random. Therefore remove mean before calculating correlation to give the atuocovariance:

$$C_v(n) = E[(v(m)-\mu_v)(v(m-n)-\mu_v)] = R_v(n) - \mu_v^2 \qquad \text{-(13.2)}$$

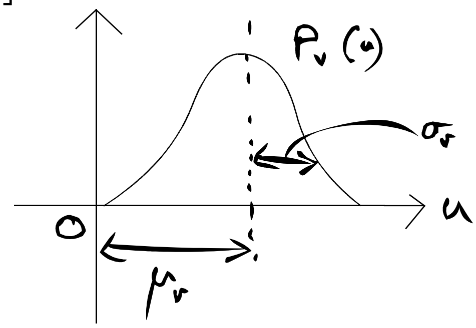
If $\{v(n)\}$is a ample from a zero mean stationary white noise process with variance σ^2, then it satisfies:

$$\mu_v = 0 \quad , \quad \underbrace{R_v(n)}_{\text{see note}} = \sigma^2 \delta_{no} \qquad \text{-(13.3)}$$

Note: i.e. its autocorrelation sequence is an impulse at the origin.
Its Fourier Transform $S_v(\omega) \triangleq S_v(\exp[j\omega]) = \sigma^2$ - a constant.

Gaussian or normal density

$$p_v(u) = \frac{1}{\sqrt{2\pi}\,\sigma_v} \exp\left[\frac{-(u-\mu_v^2)}{2\sigma_v^2}\right]$$



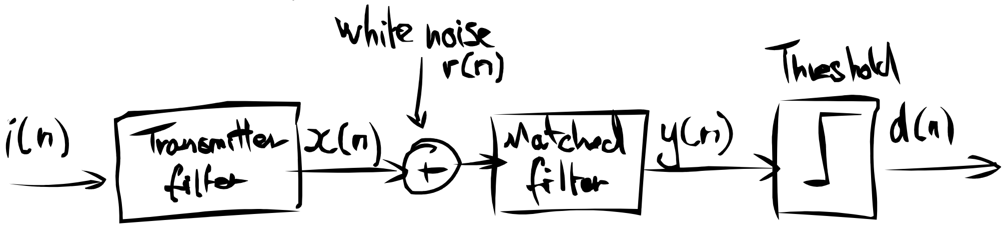
Detection of known signals in noise

Note: If u,v,x,y are random values &
a,b,c,d are arbitary, but known constants
then

$$E[(au+bv)(cx+dy)] = acE[ux] + adE[uy] + bcE[vx] + bdE[vy] \qquad \text{-(13.4)}$$

A noisy signal $\omega(n) = x(n) + v(n)$ where
$\{x(n)\}$ is signal and
$\{v(n)\}$ is zero-mean white noise with

$$\left.\begin{aligned} E[v(n)x(m)] &= x(m)E[v(n)] = 0 \\ E[v(n)v(m)] &= \sigma^2 \delta_{mn} \end{aligned}\right\} \qquad \text{-(13.5)}$$



Model of a typical signal detection application (e.g. Radar & communication)
$\{i(n)\}$ - sequence of impulses
$\{d(n)\}$ - binary decision sequence

Matched filter IR is $\{\epsilon\, x(N-n)\}$

Filter output at time 0 has

$$E[y(n)] = E[\sum_{n=0}^{N} h(n)x(N-n) + \sum_{n=0}^{N} h(n)v(N-n)]$$

using (13.4) ⇒

$$= \sum_{n=0}^{N} h(n)x(N-n) + \overbrace{\sum_{n=0}^{N} h(n)E[v(N-n)]}^{\text{tends to 0}}$$

(using (12.4) ⇒)

$$=\epsilon\sum_{n=0}^{N}x^2(n) \qquad \text{-(13.6)}$$

and

$$E[y^2(N)]=\sum_{m=0}^{N}\sum_{n=0}^{N}h(m)h(n)E[(x(N-m)+v(N-m))(x(N-n)+v(N-n))]$$

using (13.4) & (12.4)

$$E[y^2(N)]=\left(\epsilon\sum_{n=0}^{N}x^2(n)\right)^2+\sigma^2\sum_{n=0}^{N}h^2(n) \qquad \text{-(13.7)}$$

using (13.6), (13.7) & (12.4) output variance is

$$E[y^2(N)]-E^2[y(N)]$$
$$=\sigma^2\sum_{n=0}^{N}h^2(n)$$
$$=\epsilon^2\sigma^2\sum_{n=0}^{N}x^2(n) \qquad \text{-(13.8)}$$

Detection performance in terms of signal-noise ratio (SNR)

$$\rho_0=\frac{E^2[y(n)]}{E[y^2(N)]-E^2[y(N)]} \qquad \text{-(14.1)}$$

Using (13.6) & (13.8) ⇒

$$\rho_o=\sum_{n=0}^{N}\frac{x^2(n)}{\delta^2} \qquad \text{-(14.2)}$$

Total signal energy

$$e_x=\frac{1}{N+1}\sum_{n=0}^{N}x^2(n) \qquad \text{-(14.3)}$$

Input SNR

$$\rho_i=\frac{e_x}{\delta^2} \qquad \text{-(14.4)}$$

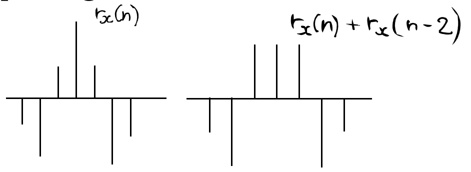
SNR gain of matched fitler

$$G=\frac{\rho_o}{\rho_i}=N+1 \qquad \text{-(14.5)}$$

⇒ the longer the signal sequence the more noise can be tolerated but at the expense of rate of detection.

Example

In radar: If the sequence is too long then range of radar will be greater, increasing the risk of overlapping echoes due to multiple targets.



Interference between adjacent 'target' responses depends on the signal autocorrelation $\{r_x(n)\}$

Abiguity depends on $R_x(n)$

Example

In a radar system, 2 echoes separated by *m* time intervals give

$$y_2(N)=y(N)+y(N-m)$$
$$=\epsilon(r_x(0)+r_x(m))$$

$\Rightarrow$ false alarm (or false dismissal) if $r_x(m)$ are large for $m\neq 0$

$\therefore$ Use signals with large peak-sidelobe ratio.

$$x=\frac{r_x(0)}{\max[|r_x(n)|]}$$

Modelling signals

Suppose $\{\omega(n)\}$, a sample of a random process, is input to a filter with IR $\{h(n)\}$, and $\{x(n)\}$ is a filter output with

$$R_x(n)=\sum_{l=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}h(l)h(m)E[\omega(k-l)\omega(k-n-m)] \qquad \text{-(14.6)}$$

(k is a constant which is used to shift the noise from the impulse response.)

Using (13.1) this simplifies to

$$R_X(n)=\sum_{l=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}h(l)h(m)R_\omega(n+m-l)$$

or

$$R_x(n)=\sum_{l=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}h(m)h(l+m)R_\omega(n-l)$$

i.e. $R_X(n)=\{r_h(n)\}*\{R_\omega(n)\}$ -(14.7)

Where $\{r_h(n)\}$ is autocorrelation of $\{h(n)\}$

Taking ZT and using (11.13)

$$S_x(z)=H(z)h(z^{-1})S_\omega(z) \qquad \text{-(14.8)}$$

where $R_x(z)=S_x(z)$ and $R_\omega(z)=S_\omega(z)$.

(We use S because we want the spectrum)

If input is unit variance white noise process, then $S_\omega(z)=1$, and

$$S_x(z)=H(z)H(z^{-1}) \qquad \text{-(14.9)}$$
$$=|H(\omega)|^2 \qquad \Leftarrow \text{power spectrum}$$

For compression of image and audio, use

$$H(z)=\frac{1}{1-rz^{-1}} \qquad \text{-(15.1)}$$

with power spectrum (lowpass)

$$S_x(\omega)=\frac{1}{(1+r^2-2r\cos\omega)} \qquad \text{-(15.2)}$$

Data Compression

To effeciently code signals.
Irreversible or lossy coding – not possible to reconstruct original signal.

Transformed coding

Discrete cosine transform (DCT)

$$C_{ij}=\begin{cases}\frac{1}{\sqrt{N}} & u=0, 0\le j<N\\ \frac{1}{\sqrt{N}}\cos\frac{\pi(2j+1)i}{2N} & 1\le i<N, 0\le j<N\end{cases} \qquad \text{-(15.3)}$$

real, orthonormal transform, ie:

$$C^{-1}=C^T \qquad \text{-(15.4)}$$

transformed nth block of samples $x(n)$ (each block of N sample)

$$\underline{u}(n)=\underline{C}\,\underline{x}(n)$$

output $\underline{y}(n)=C^T(\underline{u}(n)+\underline{q}(n))=\underline{x}(n)+\underline{C^T}\,\underline{q}(n)$

where $\underline{q}(n)$ - quantisation error.

From (15.2) and (15.3), energy in ith DCT coefficient $\left(u_i^2(n)\right)$

$$E[u_i^2(n)]=E[u_0^2(n)]\frac{(1-r)^2}{1+r^2-2r\cos\left[\frac{\pi i}{2N}\right]}$$

$$S_0 \quad \frac{E[u_i^2(n)]}{E[u_0^2(n)]}\rightarrow_{i\rightarrow N}\frac{(1-r)^2}{(1+r^2)} \qquad \text{-(15.6)}$$

For $r=0.96$, this ratio is $<10^{-3}$, high frequency (i.e. Large i) coefficients require up to 10 bits/sample less than the lowest frequency ($i=0$).

Threshold coding for quantisation, i.e.

$$Q(u)=\begin{cases}u+q & |u|>t_q\\ 0 & |u|\le t_q\end{cases} \qquad \text{-(15.7)}$$

where t_q is a theshold value.

Run-length coding

Includes:
- Length of stretch of zeros
- Values of non-zero samples
- An indicator of another stretch of zeros

Joint Picture Experts Group (JPEG)

- Based on 2D DCT, uses 8x8 block & treshold quantisation
- Compression up to 30:1

Sub-coding approach (Disc)rete Wavelet Transform (DWT) transform coding)

(refer to diagram (of lecture 15))
$G(z)$ & $H(z)$ satisfy (11.1) & (11.2), e.g. Daubechies filter.

Discrete Fourier Transform (DFT)

- Computes Fourer specturm
- The DFT pair of a periodic sequence $\{x_p(n)\}$ with period of N sample is

$$X_p(K)=\sum_{n=0}^{N-1} x_p(n)e^{-j(\frac{2\pi}{N})nk} \qquad \text{-(16.1)}$$

$$x_p(n)=\frac{1}{N}\sum_{k=0}^{N-1} X_p(k)e^{j(\frac{2\pi}{N})nk} \qquad \text{-(16.2)}$$

These are known as the "1-D DFT Pair". A DFT is concerned with a sequence which is periodic (hence the subscript p). Specifically, it has a period on n. (16.2) is the inverse of (16.1) and reconstructs the original signal.
[see topleft from diag for 16/17]
(Didn't really understand it, but the graphs are paired by column)

Example

If

$$\begin{aligned} x_p(n) &= a^n \qquad 0 \le n \le N-1 \\ x_p(n+mN) &= x_p(n) \end{aligned}$$

where $a=0.9$, $N=16$

$$X_p(K)=\sum_{n=0}^{N-1} a^n e^{-j(\frac{2\pi}{N})nk}$$

$$=\sum_{n=0}^{N-1}\left[ae^{-j(\frac{2\pi}{N})nk}\right]^n$$

$$=\frac{1-a^N}{1-ae^{-j(\frac{2\pi}{N})nk}} \qquad 0 \le k \le N-1$$

[see bottom left diag of 16/17]
(Note that the middle and lower portion are repeated both left and right)

Z Transform and DFT

$$X(z)=\sum_{n=0}^{N-1} x(n)z^{-1}$$

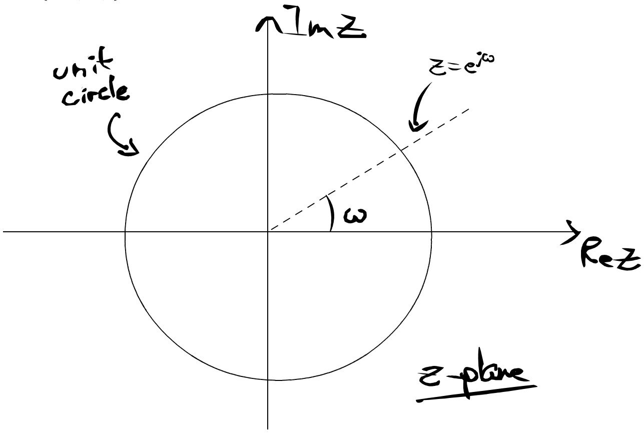
let $z=re^{j\omega}$ where r is magnitude, $\omega=\frac{2\pi k}{N}$ is angle.

$$X(re^{j\omega})=\sum_{n=0}^{N-1}(x(n)r^{-n})e^{-jn\omega}$$

$$X(z)|_{z=e^{j\omega}} = X(e^{j\omega}) \overset{\circ}{=} X(\omega) \quad (\text{for } r=1)$$

$$= \sum_{n=0}^{N-1} X(n) e^{-jn\omega} \qquad \text{-(16.3)}$$

$$= DFT\{x(n)\}$$



Also,

$$X(z) = \sum_{n=0}^{N-1} x_p(n) z^{-n}$$

$$= \sum_{n=0}^{N-1} . \frac{1}{N} \sum_{k=0}^{N-1} x_p(t) e^{j(\frac{2\pi}{N})nk} z^{-n} \qquad \text{(using (16.2))}$$

$$= \sum_{k=0}^{N-1} . X_p(k) \frac{1}{N} \sum_{n=0}^{N-1} \left(e^{j(\frac{2\pi}{N})k} z^{-1} \right)^n$$

$$= \sum_{k=0}^{N-1} . \frac{X_p(k)}{N} . \left(\frac{1-z^{-N}}{1-z^{-1} e^{j(\frac{2\pi}{N})k}} \right) \qquad \text{-(16.4)}$$

i.e. Z tranform in terms of DFT coefficients

$$\frac{2-D\,DFT}{X_p(k_1, k_2)} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_p(n_1, n_2) e^{-j(\frac{2\pi}{n_1}k_1)} e^{-j(\frac{2\pi}{n_2}k_2)} \qquad \text{-(16.5)}$$

$$x_p(n_1, n_2) = \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} X_p(k_1, k_2) e^{j(\frac{2\pi}{n_1}k_1)} e^{j(\frac{2\pi}{n_2}k_2)} \qquad \text{-(16.6)}$$

where N_1 – period along n_1
N_2 – period along n_2

If $N_1 = N_2$

$$X_p(k_1, k_2) = \frac{1}{N_1} . \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_1-1} x_p(n_1, n_2) e^{-j2\pi \frac{(n_1 k_1 + n_2 k_2)}{N_1}} \qquad \text{-(16.7)}$$

$$x_p(n_1, n_2) = \frac{1}{N_1} . \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_1-1} X_p(k_1, k_2) e^{j2\pi \frac{n_1 k_1 + n_2 k_2}{N}} \qquad \text{-(16.8)}$$

Properties of DFT

1. Seperability

(16.7) & (16.8) $\rightarrow_{N-1}$

$$X_p(k_{1,}k_2)=\frac{1}{N}.\sum_{n_1=0}^{N-1}e^{-j(\frac{2\pi}{N})n_1k_1}\sum_{n_2=0}^{N-1}x_p(n_{1,}n_2)e^{-j(\frac{2\pi}{N})n_2k_2} \quad \text{-(16.9)}$$

for $k_{1,}k_2=0,1,\ldots,N-1$

$$x_p(n_{1,}n_2)=\frac{1}{N}.\sum_{k_1=0}^{N-1}e^{j\frac{(2\pi)}{N}n_{1,}k_1}\sum_{k_2=0}^{N-1}X_p(k_{1,}k_2)e^{j(\frac{2\pi}{N})n_2k_2} \quad \text{-(16.10)}$$

for $n_{1,}n_2=0,1,\ldots,N-1$

Advantage:

(16.9) $\rightarrow$

$$X_p(k_{1,}k_2)=\frac{1}{N}.\sum_{n_1=0}^{N-1}X_p(n_{1,}k_2)e^{-j(\frac{2\pi}{N})n_1k_1} \quad \text{-(16.11)}$$

(fix k_1)

where

$$X_p(n_{1,}k_2)=N\left[\frac{1}{N}.\sum_{n_2=0}^{N01}x_p(n_{1,}n_2)e^{-j(\frac{2\pi}{N})n_{2,}k_2}\right] \quad \text{-(16.12)}$$

(fix k_2)

Similarly, for Inverse DFT (IDFT)

$n_1 \rightarrow \text{row}$

$n_2 \rightarrow \text{column}$

Above we have defined a 2-D DFT using 2x 1-D DFTs.

$\Rightarrow$ 2-D DFT can be obtained in 2 steps by successive application of 1-D DFT

Similarly for 2-D IDFT

2. Translation

$$x_p(n_{1,}n_2)e^{j(\frac{2\pi}{N})(k_{10}n_1+k_{20}n_2)}\Leftrightarrow X_p(k_1-k_{10,}k_2-k_{20}) \quad \text{-(17.1)}$$

(i.e. Shift $(k_{1,}k_2)$ to $(k_{10,}k_{20})$)

$$x_p(n_1-n_{10,}n_2-n_{20})\Leftrightarrow X_P(k_{1,}k_2)e^{-j(\frac{2\pi}{N})(k_1n_{10}+k_2n_{20})} \quad \text{-(17.2)}$$

(i.e. Shift $(n_{1,}n_2)$ to $(n_{10,}n_{20})$)

(17.2) States that if you multiply the spectrum (the sequence in the frequency domain) the effect is to shift the origin in the time domain.

Example

Let $k_{10}=k_{20}=\frac{N}{2}$

$$e^{j(\frac{2\pi}{N})(k_{10}n_2+k_{20}n_2)}=e^{j\pi(n_1+n_2)}=(-1)^{(n_1+n_2)}$$

and

$$x_p(n_1,n_2)(-1)^{(n_1+n_2)}\Leftrightarrow X_p\left(k_1-\frac{N}{2,}k_2=\frac{N}{2}\right) \qquad \text{-(17.3)}$$

[see diag 16 & 17, bottom right quadrant of page, left digram (A simple image)]

3. Periodicity & Conjugate symmetry

$$X_p(k)=X_p(k+N) \qquad \text{-(17.4)}$$
$$X_p(k)=X^*_p(-k) \qquad \text{-(17.5)}$$
$$|X_p(k)|=|x_p(-k)| \qquad \text{-(17.6)}$$

4. Rotation

Let $n_1=r\cos\Theta,\quad k_1=R\cos\phi$
$n_2=r\sin\Theta,\quad k_2=R\sin\phi$

then $x_p(n_{1,}n_2)\Rightarrow x_p(r,\Theta)$
$X_p(k_{1,}k_2)\Rightarrow X_p(R,\phi)$

$$x_p(r,\Theta+\Theta_0)\Leftrightarrow X_p(R,\phi+\phi_0) \qquad \text{-(17.7)}$$

[see diag 16 & 17, very bottom right]

5. Distributivity & Scaling

$$DFT\{x_1(n)+x_2(n)\}=DFT\{x_1(n)\}+DFT\{x_2(n)\} \qquad \text{-(17.8)}$$
$$DFT\{x_1(n)\cdot x_2(n)\}\neq DFT\{x_2(n)\}\cdot DFT\{x_2(n)\} \qquad \text{-(17.9)}$$
$$ax(n)\Leftrightarrow aX(k) \qquad \text{-(17.10)}$$

where a is a scalar constant

Circular Convolution

- Provides a link between time domain and frequency domain.
- The circular or periodic convolution of 2 periodic sequences of period N is

$$y_p(n)=\sum_{l=0}^{N-1}x_p(l)h_p(n-l) \qquad \text{-(17.11)}$$

- Filtering operation

[see diag 17&18, top left, top left]

The top two sequences are periodic, and are $x_p(l)$ and $h_p(n-l)$ respectivly, the others are derivations ending finally on the definition of (17.11)

$$\left.\begin{aligned} Y_p(k) &= \sum_{n=0}^{N-1}\left[\sum_{l=0}^{N-1} x_p(l)h_p(n-l)\right]e^{-j(\frac{2\pi}{N})nk} \\ &= \sum_{l=0}^{N-1} x_p(l)\left[\sum_{n=0}^{N-1} h_p(n-l)e^{-j(\frac{2\pi}{N})(n-l)k}\right]e^{-j(\frac{2\pi}{N})lk} \\ &= H_p(k)\sum_{l=0}^{N-1} x_p(l)e^{-j(\frac{2\pi}{N})lk} = H_p(k)X_p(k) \end{aligned}\right\} \qquad \text{-(17.12)}$$

Linear Convolution

In practice h and x won't tbe of the same length, we use a linear/aperiodic convolution to over come this.

The linear or aperiodic convolution of seqences:

$x_p(n)$ non-zero in $0 \le n \le N_1 - 1$

$h_p(x)$ non-zero in $0 \le n \le N_2 - 1$

is:

$$y_p(n) = \sum_{m=0}^{N_1+N_2-1} h_p(n)x_p(n-m) \qquad \text{-(18.1)}$$

- The appropriate number of zero-valued samples are appended to both: $x_p(n)$ and $h_p(n)$ to make them (N_1+N_2-1)-point sequences.
- Otherwise the two periods will overlap → wraparound error.

[see diag 17&18 top left middle]

Append 3 to upper, append 6 to lower, reflect h, multiply all, shift right, multiply, repeat 10 times. (10 is the period)

Sectioned Convolution

If $N_1 \gg N_2$ or $N_2 \gg N_1$ then it is inefficient and impractical to use N_1+N_2-1 as the period of convolution.

Because:

The entire longer sequence must be available before convolution.

- Long delays before any output is generated
- If N_1+N_2-1 is too large, it is impractical to use DFT

  Instead, we can use sectioned convolution.

Overlap-all

Note that the subscript p (denoting periodic sequences) has been ommited from this section, but it must be assumed.

If $x(n)$ is of infinitie duration and $h(n)$ is of N_2 samples

Make $x(n) = \sum_{k=0}^{\infty} x_k(n)$

where $x_k(n)\begin{cases} x(n) & kN_3 \le n \le (k+1)N_3-1 \\ 0 & \text{otherwise} \end{cases}$

N_3 same order as N_2

[see diag 17&18 bottom left left]

The linear convolution:

$$y(n)=\sum_{m=0}^{n} h(m)\underbrace{\sum_{k=0}^{\infty} x_k(n-m)}_{\text{circular convolution}}$$

$$=\sum_{k=0}^{\infty} h(n)*x_k(n) \quad (\text{* denotes convolution})$$

$$=\sum_{k=0}^{\infty} y_k(n)$$

where $y_k(n)$ is convolution of $h(n)$ with $x_k(n)$

Duration of each convolution = N_3+N_2-1 (to avoid wraparound error) ⇒ overlap between k^{th} and $(k+1)^{th}$ convolution $=N_2-1$

Need to combine sum of convolution

[see diag 17&18 bottom left middle]

Need to wait for output of $y_1(n)$ to add to $y_0(n)$ before outputting result.

Fast Fourier Tranforms (FFT)

DFT pair:

$$X(k)=\sum_{n=0}^{N-1} x(n)\omega_N^{nk} \qquad \text{-(18.2)}$$

$$x(n)=\frac{1}{N}\sum_{k=0}^{N-1} Z(k)\omega_N^{-nk} \qquad \text{-(18.3)}$$

where: $n=0,1,\ldots,N-1$

$K=0,1,\ldots,N-1$

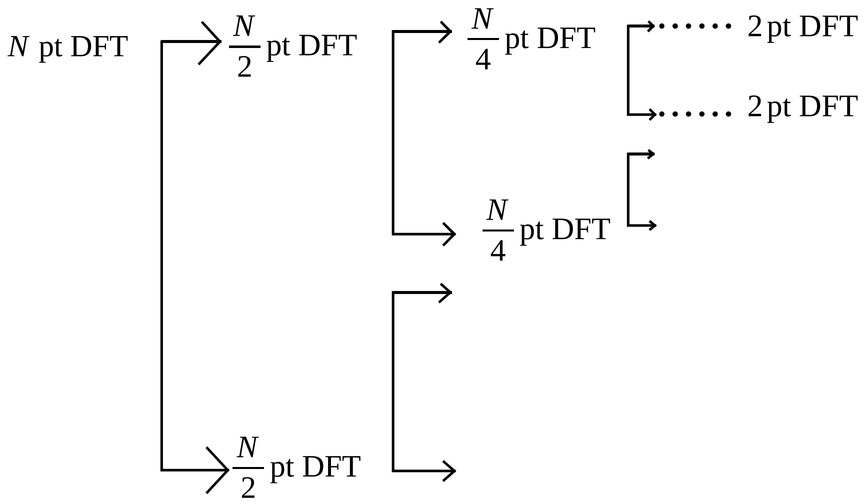
$\omega_N=e^{-j\frac{2\pi}{N}}$ (requires cos & sin, computationally **slow**)

N is a power of 2

FFT reduces computational complexity of (18.2) and (18.3)

Fast Fourier Transform (FFT) reduces time for computing DFT from N^2 to $Nlog_2 N$ complex multiplications and additions.

Basic idea



Use $\omega_N^k = (\omega_N^{k-L})\,\omega_N^L$ -(18.4)

only ω_N^L involves cosine and sine.

Computing IDFT using FFT

IDFT $x(n) = \frac{1}{N} \underbrace{\sum_{k=0}^{N-1} X(k)\,\omega_N^{-nk}}_{\text{DFT of } \{X^*(k)\}}$

where $\omega_N = e^{-j(\frac{2\pi}{N})}$

Complex conjugate & multiply by $N \rightarrow$

$$Nx^*(n) = \sum_{k=0}^{N-1} X^*(k)\,\omega_N^{nk}$$

Complex conjugate and divide by N.

$$x(n) = \frac{1}{N}\left[\sum_{k=0}^{N-1} X^*(k)\,\omega_n^{nk}\right]^*$$

$\Rightarrow$ use FFT for both DFT and IDFT

Speech processing

- Speech – a continuously changing sound pressure wave which links the speaker's mout to the listener's ears
  [see diag 19 top]
- Decompose the sound wave into component elements & use these elements to recognise speech or to create new messages.

The human speech production mechanism

[see diag 19 lower left]

- Larynx consists of carilage held together by ligaments.
- Vocal cords – a pair of elastic bands of muscles and mucous membrane
- Trachea – air passage from larynx to bronchi
- Pharynx cavity – connect mouth to stomach
- Acoustic filter – vocal tract & 2 nasal tracts
- Vocal tract:
  - From vocal cords to lips including the various articulators (i.e. Lips, jaw, tongue, velum)
  - Length ~17cm
  - Corss-sectional area – 0 to 20cm^2
- Nasal tract:
  - Velum to nostril
  - Length ~ 12cm
  - Coupled to vocal tract by velum
- Chest cavity – expands & contracts to force air from lungs
- Vocal cords:
  - Vibrate and modeulate air into discrete puffs of broad-spectrum pulses when tensed
  - Air is unaffected when spread apart

Voice generation methods

1. Voiced sounds

- Elevate air pressure in lungs and adjust tension of vocal cords
  - to cause a flow of air
  - to cause vocal cords to vibrate
  - periodic interruption to generate discrete pulses
- Sounds depends on shape of vocal tract
- E.g., /I/ in “six”

[see diag 19 middle right]

- Fundamental period T_0:
  - Time between successive vocal cord “open” phases
  - Proportional to size of vocal cords
  - Inversely propertional to tension exerted on vocal cords
- Fundamental frequency or pitch $F_0 = \frac{1}{T_0}$ is rate of vibration
- Pitch range:
  - 50-250Hz (man)
  - 120-500Hz (woman)

2. Unvoiced sound

- Articulators form a constriction along the vocal tract & force air through it at high Reynold's number to cause turbulance
  $\rightarrow$ to create a noise source
- E.g. /s/ in “six”

3. Mixed (voiced & unvoiced)

- E.g. /z/ in “three zebras”

4. Plosive sound

- Form a complete closure of vocal tract and then abruptly release the built-up pressure
  $\rightarrow$silence + unvoiced, e.g. /t/ in “pat”
  or
  $\rightarrow$silence + voiced, e.g. /b/ in “boot”