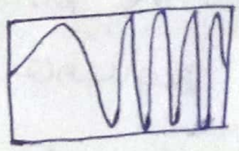
V APPLICATIONS OF DIGITAL SIGNAL PROCESSING

Spectral Analysis of Non-Stationary signals:

→ DFT can be employed for the spectral analysis of finite length signal composed of sinusoidal components as long as frequency, amplitude, phase of each component are stationary i.e., time-invarient & independent of signal length.

→ There are practical situations where the signal to be analysed is instead non stationary, for which there signal parameters are time varying.

chirp signal.



Ex: for chirp signal

$$x(n) = A\cos(\omega_0 n^2)$$

→ The instantaneous fq of $x(n)$ is given by $2\omega_0 n$, which is not a constant but increases linearly with time.

→ speech, radar, sonar are examples of some non-stationary signals.

→ Direct computation of DFT of such signals will provide misleading results.

→ So, To get around the time varying nature of signal parameters, an alternative approach would be to segment the sequence in to a set of sub sequences of short length, each subsequence centered at uniform intervals of time & its DFT is computed seperately.

→ If the subsequence length is reasonably small, it can be safely assumed to be stationary for all practical purposes.

→ As a result, the fq domain description of a long sequence is given by a set of short length DFTs, that is called time - dependent DFT.

→ To represent a non stationary signal $x(n)$ in terms of a set of short-length sequences, we can multiply with a window $w(n)$, that is stationary with respect to time and move the signal through the window.

→ The segments can be overlapping in time

→ The Discrete-time fourier transform of the short sequence obtained by windowing is called short-term time Fourier Transform, which is thus a f^n of location of window relative to original long sequence and fq.

Short-time Fourier Transform:

→ It is also known as time - dependent Fourier Transform of a sequence $x(n)$, and is defined by

$$X_{STFT}(e^{jw}, n) = \sum_{m=-\infty}^{\infty} x(n-m)\, w(m)\, e^{-jwm}$$

$w(n)$ → suitably chosen window sequence.

→ The f^n of the window is to extract a finite length portion of $x(n)$ such that the spectral characteristics of section extracted are approximately stationary over the duration of the window for practical purposes.

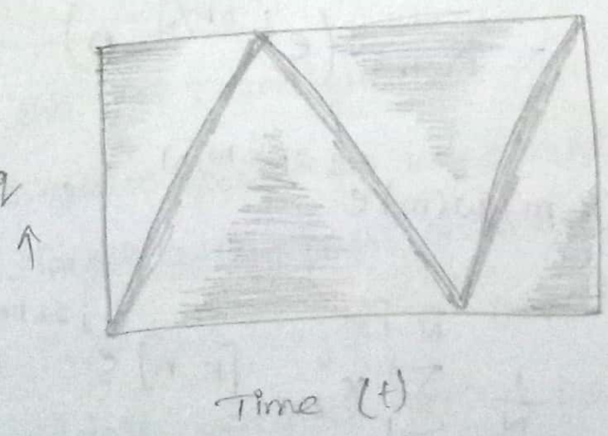
→ If $w(n) = 1$, the STFT → reduces to conventional DFT of $x(n)$.

→ STFT is a f^n of two variables ie, time (n), frequency (ω).

→ STFT is periodic f^n with respect to ω, with period 2π.

→ In most applications magnitude of STFT is of major interest.

→ The display of magnitude of STFT is usually refered to as spectrogram. Its display (plotting) requires 3 dimensions. Often it is plotted in 2 dimensions (time → x axis, fq → y axis) with magnitude represented by the darkness of the plot



Spectrogram of chirp signal

→ Here the white areas represent zero-valued magnitude, while gray areas represent non-zero magnitudes, with largest magnitude being shown in black

Sampling in Time and fq Dimensions :

→ Due to finite length of the windowed sequence, STFT is accurately represented by its fq samples as long as no. of fq samples is greater than window length.

→ The portion of sequence $x(n)$ inside the window can be fully recovered from fq samples of STFT

→ let R → be the window length defined in $0 \le m \le R-1$.

We sample $X_{STFT}(e^{j\omega}, n)$ at N equally spaced fqs $\omega_k = 2\pi k/N$, with $N \geq R$.

$$X_{STFT}[k,n] = X_{STFT}(e^{j\omega}, n)\Big|_{\omega = 2\pi k/N}$$

$$= X_{STFT}(e^{j2k/N}, n)$$

$$= \sum_{m=0}^{R-1} x(n-m)\, w(m)\, e^{-j2\pi km/N} \qquad 0 \le k \le N-1$$

$$\Rightarrow x(n-m)\, w(m) = \frac{1}{N} \sum_{k=0}^{N-1} X_{STFT}[k,n]\, e^{j2\pi km/N} \qquad 0 \le m \le N-1$$

$$\Rightarrow x(n-m) = \frac{1}{N\, w(m)} \sum_{k=0}^{N-1} X_{STFT}[k,n]\, e^{j2\pi km/N} \qquad 0 \le m \le R-1$$

→ To recover $x(n)$ from $X_{STFT}(k,n)$ → $N \geq R$.

The sampled STFT for a window defined in the region $0 \leq m \leq R-1$ is given by

$$X_{STFT}[k, lL] = X_{STFT}(e^{j2\pi k/N}, lL) \quad [\text{put } n = lL$$

$$= \sum_{m=0}^{R-1} x(lL-m)\, w(m)\, e^{-j2\pi km/N}$$

such that $-\infty < l < \infty$ & $0 \leq k \leq N-1$

→ It is possible to reconstruct the original signal from the sampled signal only when $N \geq R \geq L$.

Window selection:

→ In the STFT analysis of non-stationary signals, the window also plays an important role. window also plays an important role.

The length and shape of the window are main parameters.

→ The length and shape of the window are main parameters.

→ The window length R should be small for signals with widely varying spectral parameter which provides wide band spectrogram.

→ decrease in R increases the time-resolution property of STFT.

→ increase in R increases the fq-resolution property of STFT.

→ A longer window results in narrow-band spectrogram.

→ The two fq domain parameters characterising the fourier transform of a window are its main lobe width (Δ_{ML}) & relative side lobe Amplitude (A_{SL}).

→ Δ_{ML} determines the ability of the window to resolve signal components in the vicinity of each other.

→ A_{SL} controls the degree of leakage of one component into a ~~nearby~~ near by signal component.

→ So, window should be chosen to have a very small sidelobe amplitude.

<u>Analysis of speech signals using STFT</u>

→ STFT is often used in speech analysis, since they are generally non-stationary.

→ speech signals consist of voiced & unvoiced sounds.

→ speech signal segment over a small time interval can be considered as a stationary signal & its DFT can provide a reasonable representation of fq domain char of the speech in this time interval.

→ In order to provide a ~~ve~~ good estimate of changes in the vocal tract and the excitation, a short window ie, a wide band spectrogram is preferable. So the size of window selected should be close to 1 pitch period, for resolving formants (one maxima).

→ To resolve the harmonics of pitch fqs, a narrow-band spectrogram with a window size of several pitch periods is desirable.

Musical Sound Processing

Time Domain Operations :

Commonly used Time Domain operations are .

1) echo generation
2) Reverberation
3) flanging (to alternate a sound track)
4) Chorus generation
5) phasing.

→ for all these operations, the basic building block is 'delay'.

1. Echo generation :

a) Single Echo filter :

→ Echos are generated by delay units.

→ A single echo appearing R sampling periods later can be generated by the ~~IIR~~ FIR filter, which is characterized by the difference eqn.

$$y(n) = x(n) + \alpha x[n-R] \qquad |\alpha| < 1$$

or, by transfer f^n

$$\boxed{H(z) = 1 + \alpha z^{-R}}$$

R → time the sound wave takes to travel from the sound source to the listener after bouncing back from the reflecting wall, α → signal loss caused by propagation & Reflection.

→ The magnitude Response of the single echo filter shown in fig. (b). Because of its comb like structure of the response, it is also called comb filter.

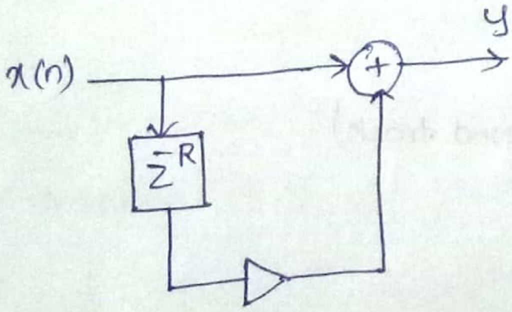
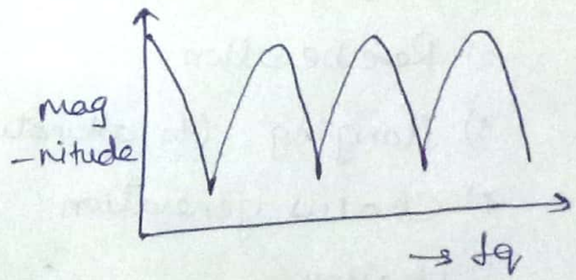


fig (a)
Single echo filter



(b)
magnitude Response.

b) Multiple echo filter:

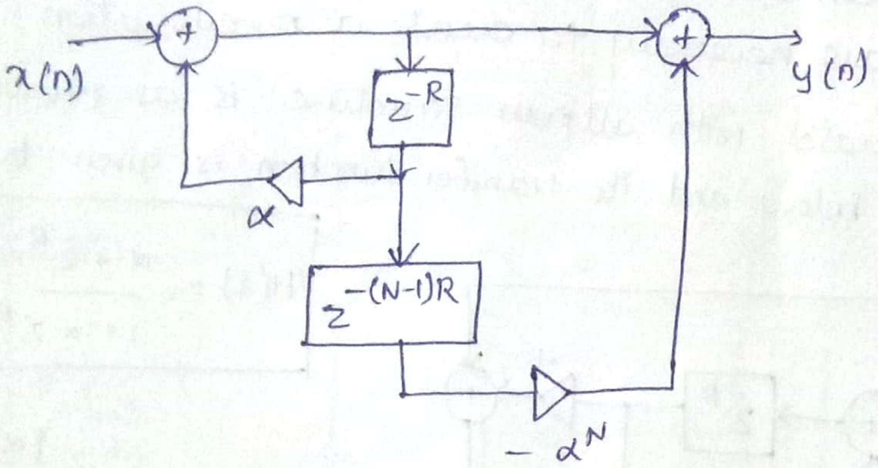
To generate a fixed no. of multiple echos spaced R sampling periods apart with exponentially decaying amplitudes, one can use FIR filter with a transfer f^n of the form

$$H(z) = 1 + \alpha z^{-R} + \alpha^2 z^{-2R} + \dots + \alpha^{N-1} z^{-(N-1)R}$$

$$\boxed{H(z) = \frac{1 - \alpha^N z^{-NR}}{1 - \alpha z^{-R}}}$$

Scanned by CamScanner

FIR filter realization of the above filter is shown below.



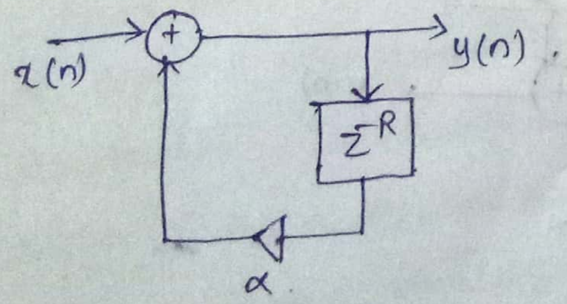
multiple echo filter

→ multiple echoes can be created by an IIR filter with transfer f^n of the form

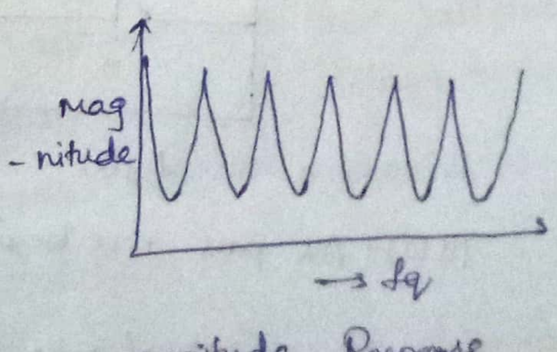
$$H(z) = 1 + \alpha z^{-R} + \alpha^2 z^{-2R} + \alpha^3 z^{-3R} + \cdots$$

$$\boxed{H(z) = \frac{1}{1-\alpha z^{-R}}}$$

The above filter can be realized as shown in fig below



IIR filter for infinite echos



magnitude Response of IIR filter

b) Reverberation :

→ Reverberation is composed of densely packed echoes.

→ It has been observed that approximately 1000 echoes per second are necessary to create a reverberation.

→ A reverberator with all pass structure is as shown in the figure below and its transfer function is given by

$$\boxed{H(z) = \frac{\alpha + z^{-R}}{1 + \alpha z^{-R}}}$$

$|\alpha| < 1$

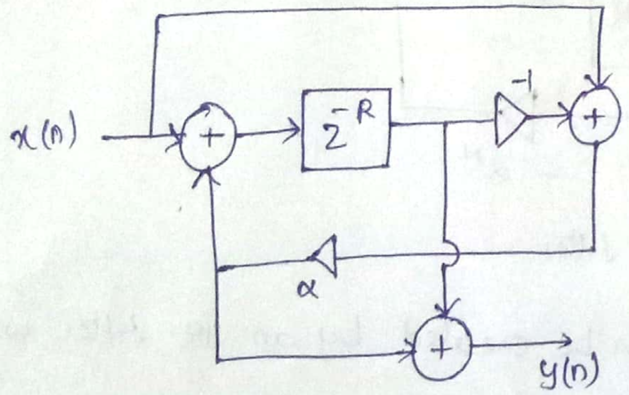


fig (i)

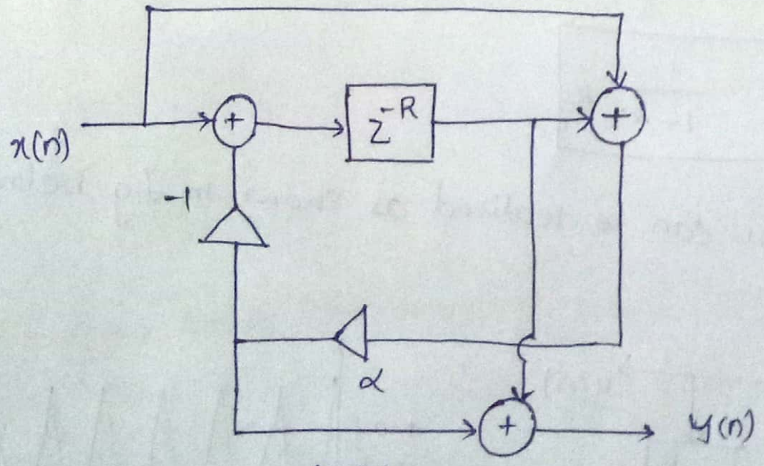


fig (2)

(i) (ii) : All pass reverberator.

→ IIR comb filter (echo filter) and all pass reverberator are basic units that are suitably interconnected to develop a natural sounding reverberation.

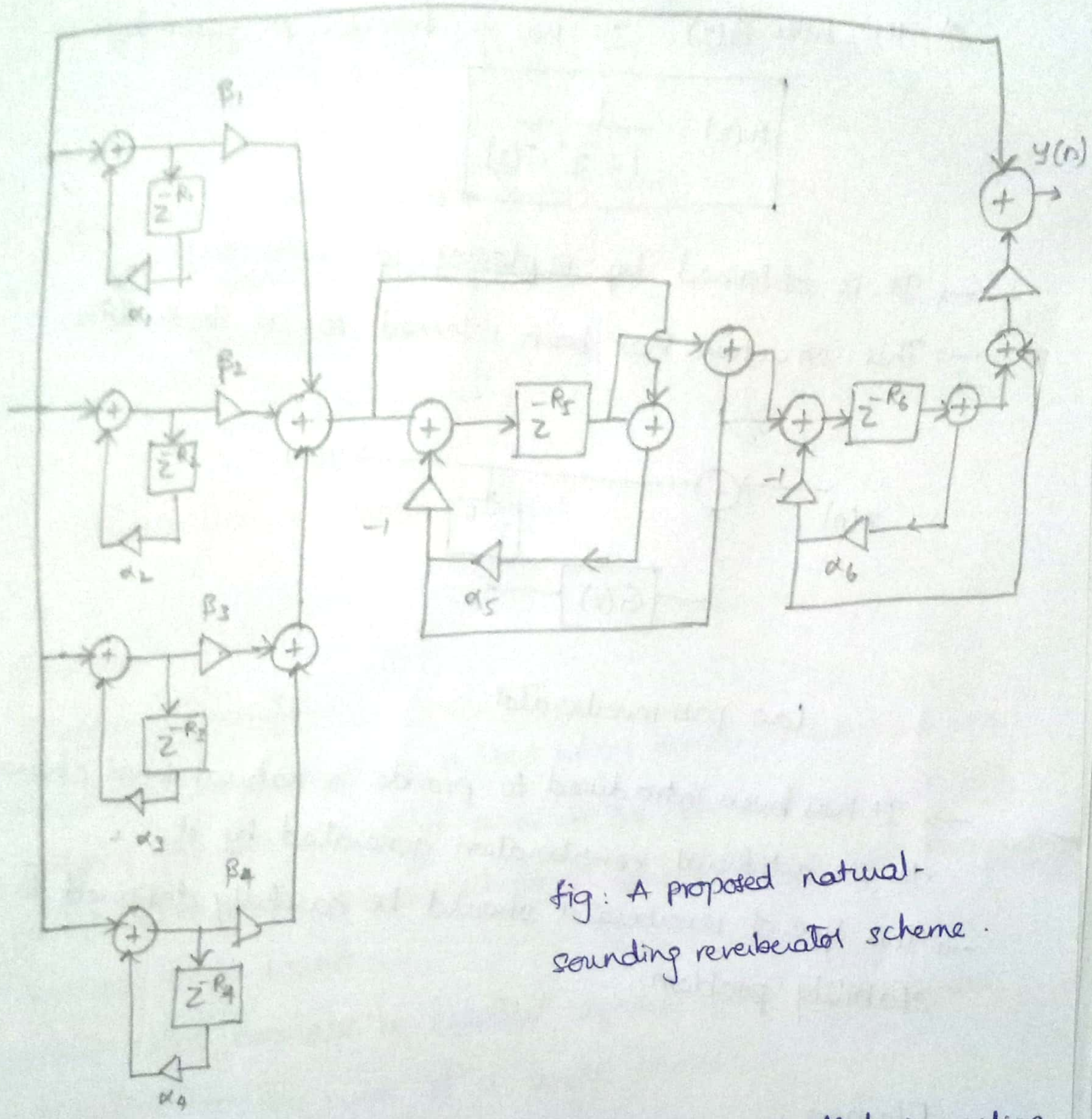


fig: A proposed natural-sounding reverberator scheme.

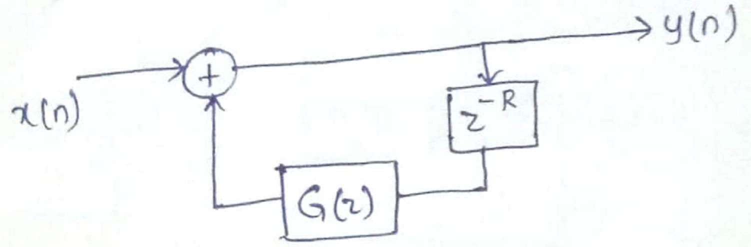
→ figure shows interconnection composed of a parallel connection of four IIR echo generators in cascade with two allpass reverberators. By choosing different values for delays in each section & the multiplier constants α_i, it is possible to arrive at pleasant sounding reverberation.

→ An interesting modification of the basic IIR comb filter is obtained by replacing the multiplier a with a lowpass FIR of IIR filter $G(z)$. It has a transfer f^n given by

$$\boxed{H(z) = \frac{1}{1 - z^{-R} G(z)}}$$

→ It is obtained by replacing α with $G(z)$.

→ This structure has been referred to as teeth filter



Low pass reverberator.

→ It has been introduced to provide a natural tonal character to the artificial reverberation generated by it.

→ This type of reverberator should be carefully designed to avoid stability problem.

Flanging :

→ It was created by feeding the same musical piece to two tape recorders and then combining their delayed outputs while varying the difference Δt between their delay times.

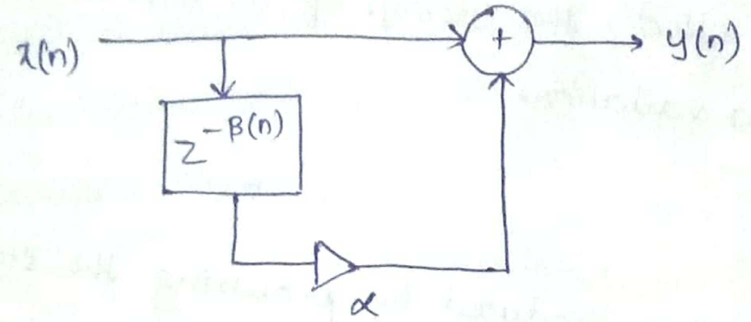
→ FIR comb filter can be modified to create flanging effect.

→ The corresponding i/p - o/p relation is given by

$$y(n) = x(n) + \alpha\, x(n - \beta(n))$$

$\beta(n)$ → periodically varying delay between 0 & R with fq ω_0

$$\beta(n) = \frac{R}{2}\left(1 - \cos(\omega_0 n)\right)$$



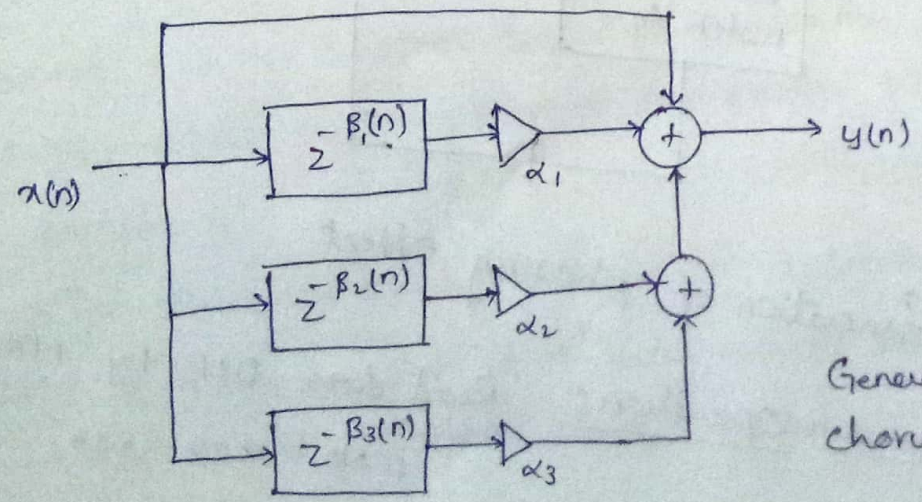
Generation of flanging effect.

~~Chorus~~

Chorus Generator:

→ The chorus effect is achied when several musicians are playing the same musical piece at the same time but with small changes in the amplitude & small timing differences between their sounds.

→ Such effect can also be created synthetically by a chorus generator from the music of a single musician.



Generation of chorus effect

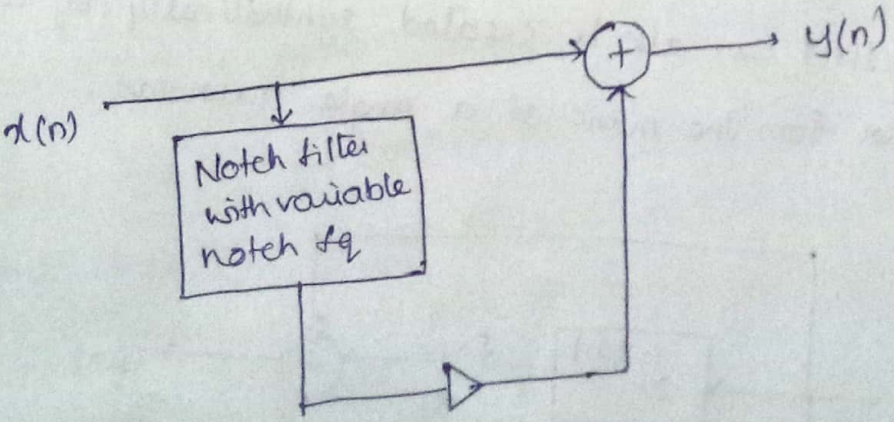
→ The above figure can effectively create a chorus of four musicians from the music of a single musician.

→ To achieve this effect, the delays $\beta_i(n)$ are randomly varied with slow variations.

Phasing effect

→ The phasing effect is produced by processing the signal through a narrowband notch filter with variable notch characteristics and adding a scaled portion of the notch filter output to the original signal

→ The phase of the signal at the notch filter output can dramatically alter the phase of the combined signal, particularly around the notch fq when it is varied slowly.



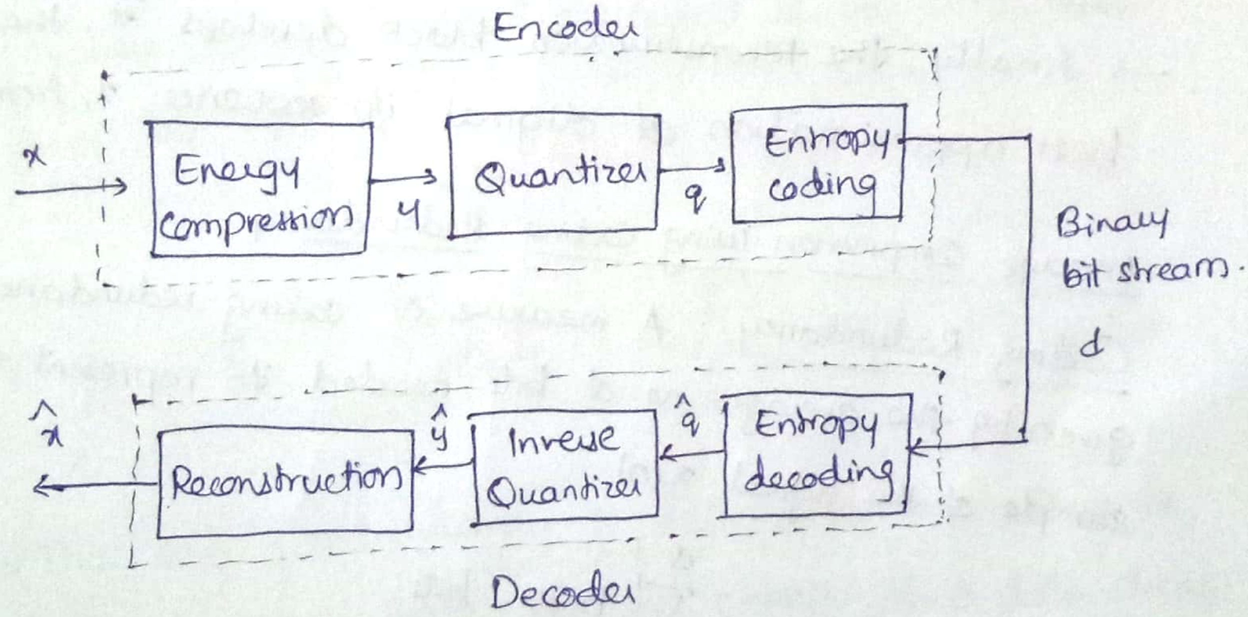
Generation of phasing effect

fq Domain Operations : Read from DSP by Mitra

P. No - 827 - 834

Signal Compression.

→ Signal compression is concerned with the reduction of the amount of data, while preserving the information content of the signal with some acceptable fidelity. [clarity of the reproduced signal].



Signal compression system- Block diagram.

→ A signal coding system consists of encoder & decoder.

Encoder: The energy compression block transforms the i/p sequence x in to another sequence y with the same total energy, while packing most of the energy in very few of samples y.

→ The quantizer block develops an approximate representation of y in the form of integer valued sequence q by y.

→ The entropy coding block encodes the integers in the sequence q into binary bitstream d.

Decoder: → The entropy decoding block regenerates the integer-valued sequence q from the binary bit stream d.

→ The inverse quantizer develops $\hat{y}$, a replica of y from q.

→ finally, the reconstruction block develops $\hat{x}$, the best approximation of original i/p sequence $\dot{x}$, from $\hat{y}$.

Image compression using coding Redundancy

Coding Redundancy: A measure of coding redundancy is given by the average no. of bits needed to represent each sample of the signal $x(n)$

$$B_{av} = \sum_{i=0}^{Q-1} b_i p_i \text{ bits}$$

total no. of bits required to represent a signal N.

B_{av} → total no. of bits required

b_i → length of i_{th} code word i.e, total no. of bits required to represent value of random variable r_i

p_i → probability of ~~of~~ each symbol r_i, given by

$$p_i = \frac{m_i}{N}, \quad 0 \le i \le Q-1.$$

m_i → total no. of samples taking the value r_i.

Q → no. of distinct probability values, such that

$$\sum_{i=0}^{Q-1} p_i = 1$$

Entropy : A measure of average information content of the signal $x(n)$ is given by Entropy $[H_x]$.

$$H_x = \sum_{i=0}^{Q-1} P_i I_i$$

where $I_i \rightarrow$ information content of the i^{th} symbol r_i

given by

$$I_i = -\log_2 P_i .$$

$$\Rightarrow \quad H_x = \sum_{i=0}^{Q-1} P_i I_i = -\sum_{i=0}^{Q-1} P_i \log_2 P_i \text{ bits/symbol.}$$

→ The coding redundancy is defined as the difference between actual data rate and entropy of a data stream

Oversampling A/D converter. [Sigma delta $(\Sigma\Delta)$ A/D converter]

→ In DSP, we accept continuous time signal, then passing through sample & hold ckt, we convert it into digital form by means of A/D converter

→ If sampling theorem $(f_s \geq 2f_m)$ is not followed, one cannot recover original signal due to aliasing.

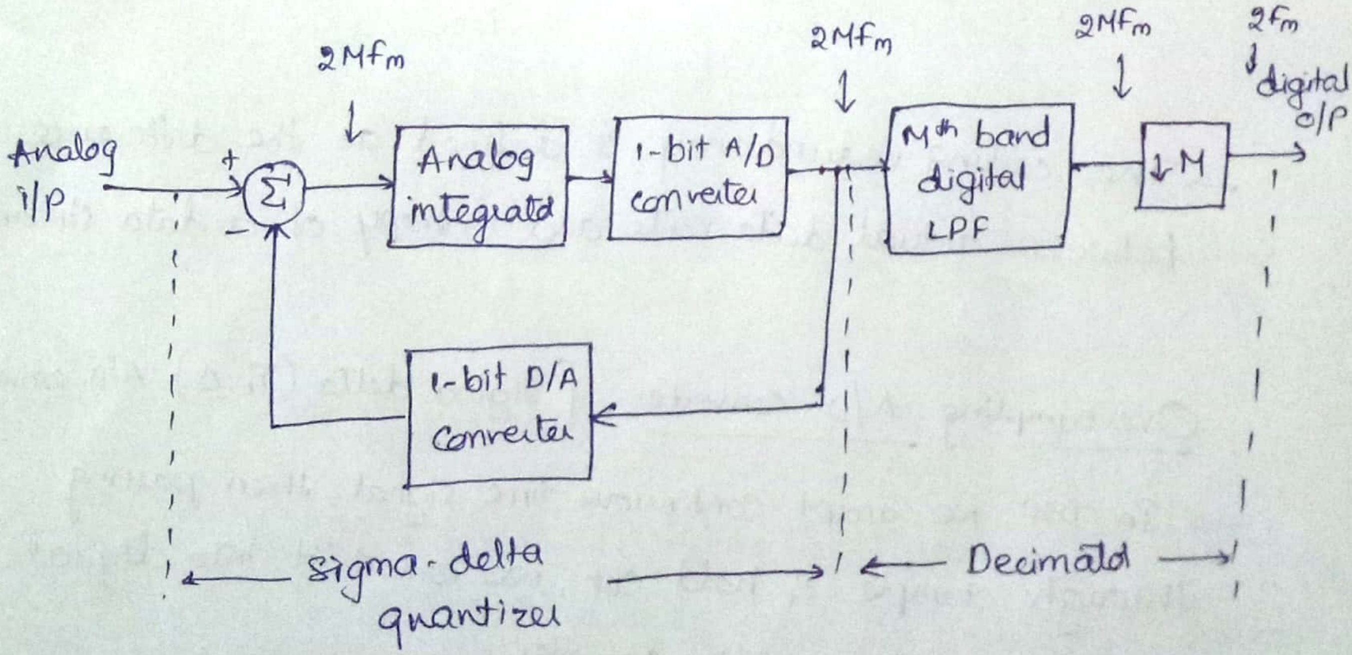
→ To prevent aliasing, we normally prefer anti-aliasing filter, before sampling.

→ But the problem is, the design of ~~of~~ anti-aliasing filter becomes very critical & of high order, if we consider sampling frequency (f_s) as twice the message signal fq (f_m)

i.e., $f_s = 2f_m$.

→ It is usually difficult to implement such a filter in VLSI technology. Such a filter also introduces distortion in its output.

→ ∴ we have to sample analog signal at a rate much higher than the nyquist rate. This will release sharp cut-off requirement of antialiasing filter, thus making the filter constructions simpler.



Over sampling ΣΔ A/D Converter

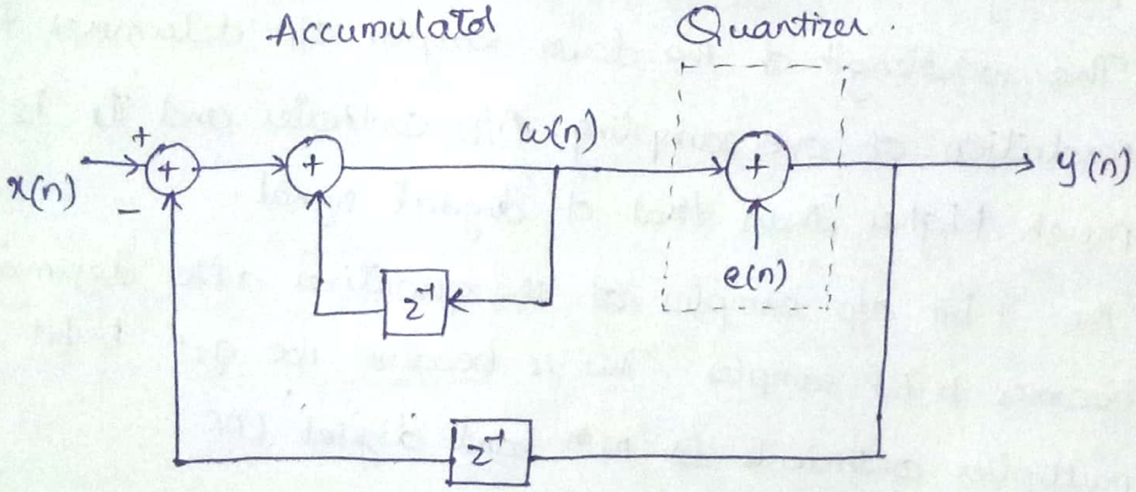
Over sampling approach to i/p signal is application of multirate digital signal processing.

→ We know that A/D converter introduces quantization noise.

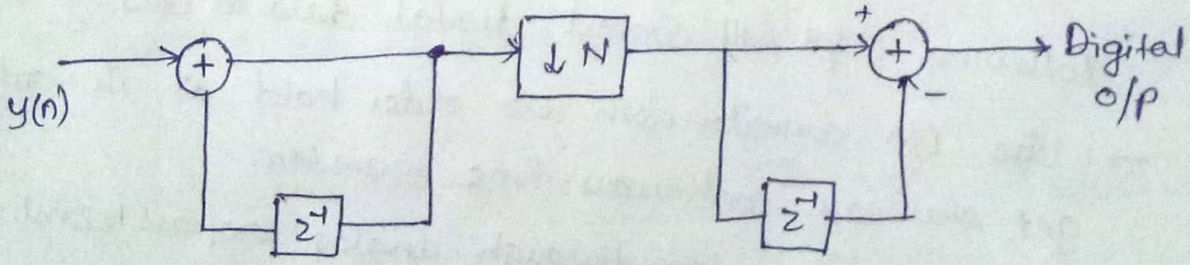
→ Due to oversampling, quantization noise power reduces in the signal.

→ One can implement sigma-delta ($\Sigma\Delta$) quantization scheme to improve noise performance of over sampling A/D converter.

→ The fig below shows the $\Sigma\Delta$ quantizer realization



Structure of $\Sigma\Delta$ quantizer.



Structure of N-pt decimator.

→ Since analog signal is sampled at a rate much higher than the Nyquist rate, we will get closely spaced samples.

→ ∴ The difference between the amplitude of two consecutive samples is very small.

→ This allows us to represent it in digital form using very few bits, usually 1-bit.

→ The sampling rate can be reduced by passing digital signal through an anti-aliasing low pass into band pass digital filter to reduce its bandwidth to π/m, and then passing it through a factor of M down sampler.

→ The wordlength of the down sampler o/p determines the resolution of oversampling A/D converter and its is much higher than that of digital signal.

→ The 1-bit o/p samples of the quantizer after decimation becomes b-bit samples. This is because we get b-bit multiplier coefficients for m^{th} band digital LPF.

Oversampling D/A ($\Sigma\Delta$) Converter

Following steps will convert digital data to analog signal:

→ 1. Use D/A converter with zero order hold at its output & get staircase continuous-time waveform.

→ 2. Pass the waveform through analog lowpass reconstruction filter.

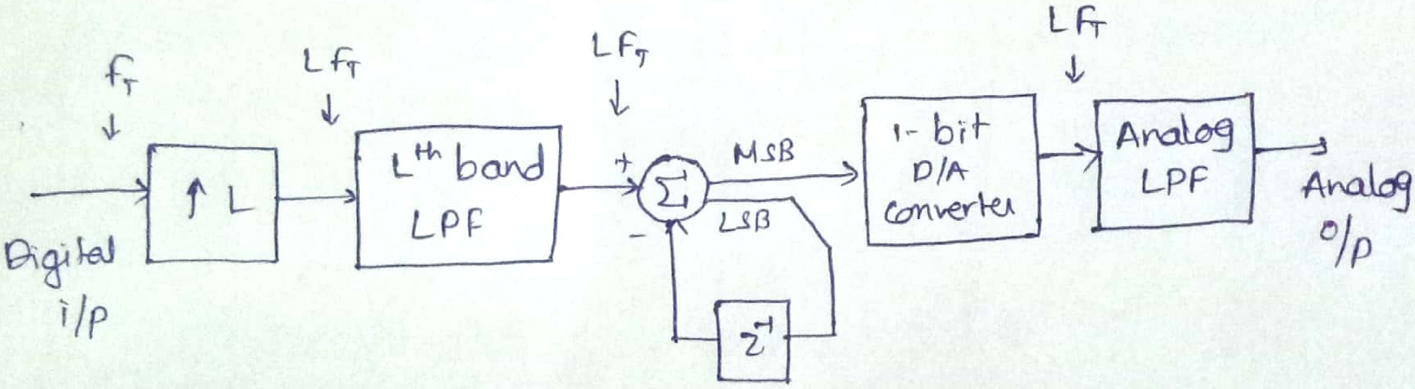
→ The problem is, if we follow Nyquist rate of sampling, we require analog low pass reconstruction filter with very sharp cut off in its fq response.

→ This leads to high precision analog ckt components.

→ To overcome this problem, we use over sampling approach, which leads to low precision analog ckt components for analog reconstruction filter.

→ One can still improve the performance of an oversampling D/A converter by employing a digital $\Sigma\Delta$ 1-bit quantizer at the o/p of ditial interpolator.

→ The fig below shows the oversampling $\Sigma\Delta$ D/A converter



Oversampling sigma-delta D/A Converter.

→ As shown in the fig, the digital i/p signal is upsampled first.

→ Then the upsampled o/p is passed through L^{th} band LPF.

→ o/p of LPF is given to sigma-delta quantizer.

→ Quantizer extracts the MSB from the i/p.

→ Remaining LSB (ie., quantization noise) is subtracted from its i/p.

→ The MSB o/p is fed to 1 bit D/A converter and finally passed through analog lowpass reconstruction filter to remove all fq components beyond signal band.