EE482: DSP APPLICATIONS

CH2 DSP FUNDAMENTALS

http://www.ee.unlv.edu/~b1morris/ee482

DIGITAL SIGNALS AND SYSTEMS

CHAPTER 2.1

ELEMENTARY DIGITAL SIGNALS

- Digital signal
  - $x(n) \quad n \in \mathbb{Z}$
  - Deterministic – expressed mathematically (e.g. sinusoid)
  - Random – cannot be described exactly by equations (e.g. noise, speech)
- Unit impulse (Kronecker delta)
  - $\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$
  - Basic building block of all digital signals
- Unit step
  - $u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases} = \sum_{k=-\infty}^{n} \delta(k)$

SINUSOIDAL SIGNALS

- Continuous
  - $x(t) = A\sin(\Omega t + \phi) = A\ \sin(2\pi f t + \phi)$
- Sampled
  - $x(n) = A\sin(\Omega nT + \phi) = A\ \sin(2\pi f nT + \phi)$
    - $\Omega = 2\pi f$
  - $x(n) = A\sin(\omega n + \phi) = A\ \sin(\mathrm{F}\pi n + \phi)$
    - $\omega = \Omega T$

RELATIONSHIPS BETWEEN FREQ VARIABLES

Table 2.1

Variable	Units	Relationships	Ranges
Ω	rads/sec	$\Omega = 2\pi f$	$-\infty < \Omega < \infty$
f	cycles/sec (Hz)	$f = \frac{\Omega}{2\pi} = \frac{\omega f_s}{2\pi}$	$-\infty < f < \infty$
ω	rads/sample	$\omega = \Omega T = \frac{2\pi f}{f_s}$	$-\pi \leq \omega \leq \pi$
F	cycles/sample	$F = \frac{f}{f_s/2} = \frac{\omega}{2}$	$-1 \leq F \leq 1$

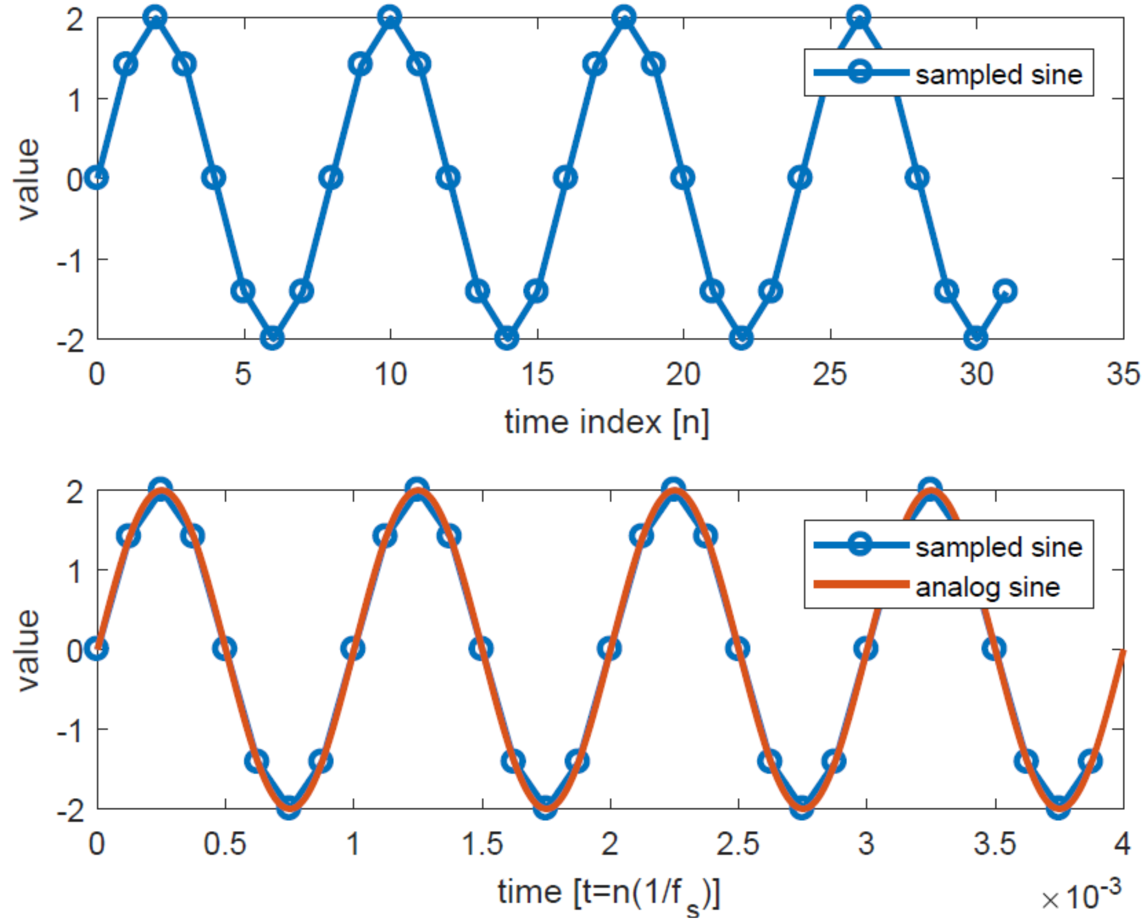
- Normalized frequency measures
- Note: max frequency for π or definition over a 2π interval
  - Consider $e^{j(\omega+2\pi k)}$

EXAMPLE 2.1

- A=2; f=1000; fs = 8000;

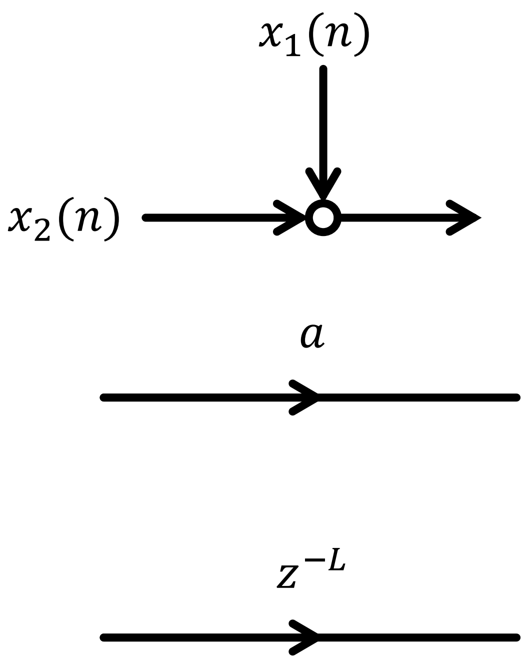
- n=0:31;
- w = 2*pi*f/fs;
- x = A*sin(w*n);

- h=figure;
- %plot sampled sine
- subplot(2,1,1)
- plot(n,x,'-o','linewidth',2);
- xlabel('time index [n]'); ylabel('value'); legend('sampled sine')
- %plot analog sine
- subplot(2,1,2)
- t=0:1e-5:4e-3;
- plot(n*(1/fs),x,'-o','linewidth',2);
- hold all;
- plot(t, A*sin(2*pi*f*t), 'linewidth',2);
- xlabel('time [t=n(1/f_s)]'); ylabel('value'); legend('sampled sine', 'analog sine')



BLOCK DIAGRAM REPRESENTATION

- Processing accomplished with 3 basic operations
- Addition
  - $y(n) = x_1(n) + x_2(n)$
- Multiplication
  - $y(n) = ax(n)$
- Time shift (delay)
  - $y(n) = x(n - L)$
  - Multiple delays can be implemented with a shift register (first-in, first-out buffer)(tapped delay line)



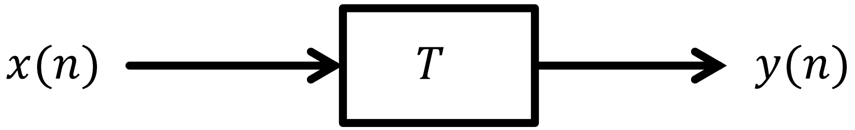
Multiplication in z domain

SYSTEM CONCEPTS

CHAPTER 2.2

SYSTEMS

- Generic system



- Linearity
  - Additive and homogeneity (scaling) properties
  - $T\{ax_1(n) + bx_2(n)\} = ay_1(n) + by_2(n)$

- Time invariance
  - Shift in input causes corresponding shift in output
  - $y(n - n_0) = T\{x(n - n_0)\}$
  - To test, check if $y_1(n) = y_2(n)$
    - $y_1(n) = y(n - n_0)$
      - Replace n by n_0
    - $y_2(n) = T\{x(n - n_0)\} = T\{g(n)\}$
      - Response of system to shifted input

LTI SYSTEMS

- Impulse response

  - Output of LTI system $y(n) = h(n)$ to input $x(n) = \delta(n)$
- Convolution
  - Input-output relationship of LTI system

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) = \sum_{k=-\infty}^{\infty} h(k)x(n-k)$$

GENERAL DIFFERENCE EQUATION SYSTEMS

$$\sum_{k=0}^{M} a_k y(n-k) = \sum_{k=0}^{L-1} b_k x(n-k) \qquad y(n) = \sum_{k=0}^{L-1} b_k x(n-k) - \sum_{k=1}^{M} a_k y(n-k)$$

- Infinite impulse response (IIR)
  - $h(n)$ non-zero as $n \to \infty$
- Finite impulse response (FIR)
  - $h(n)$ defined over finite set of n
  - Special case of above with $a_k = 0$
  - This system only has zeroes and poles at $z = 0$
- Causality
  - Output only depends on previous input
  - $h(n) = 0, \qquad n < 0$
- Stability (BIBO)
  - $\sum |h(n)| < \infty$
  - Absolutely summable

Z-TRANSFORM

- Very useful computational tool for studying digital systems
- Definition $$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n}$$ $$z = re^{j\theta}$$ ← Complex variable
  - Has associated region of convergence (ROC)
    - Values of z where summation converges
- Useful summation formulas

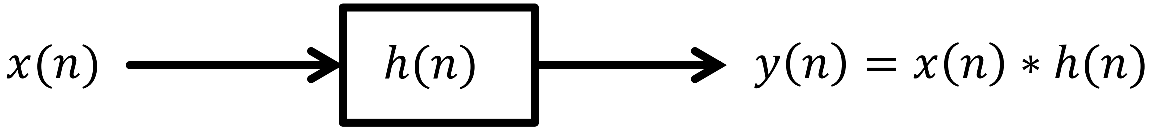
$$\sum_{n=0}^{N} \alpha^n = \frac{1-\alpha^{N+1}}{1-\alpha}$$

$$\sum_{n=0}^{\infty} \alpha^n = \frac{1}{1-\alpha}$$ ← $|\alpha| < 1$

Z-TRANSFORM PROPERTIES

- Linearity
  - $\mathcal{Z}\{ax_1(n) + bx_2(n)\} = aX_1(z) + bX_2(z)$
- Time shift
  - $\mathcal{Z}\{x(n-k)\} = z^{-k}X(z)$
- Convolution
  - $x(n) = x_1(n) * x_2(n) \rightarrow X(z) = X_1(z)X_2(z)$
    - ROC $= R_{x1} \cap R_{x2}$

TRANSFER FUNCTIONS



- Note: convolution in time is multiplication in Z-domain
- $Y(z) = X(z)H(z)$
- $H(z) = \frac{Y(z)}{X(z)}$

- General polynomial form from difference equation
  - Take Z-transform of both sides of general diff eq

$$H(z) = \frac{\sum_{k=0}^{L-1} b_k z^{-k}}{1 + \sum_{k=1}^{M} a_k z^{-k}}$$

POLES AND ZEROS

$$H(z) = \frac{\sum_{k=0}^{L-1} b_k z^{-k}}{1 + \sum_{k=1}^{M} a_k z^{-k}} \longrightarrow H(z) = b_0 \frac{\prod_{k=1}^{L-1}(z - z_k)}{\prod_{k=1}^{M}(z - p_k)} = b_0 \frac{(z - z_1)(z - z_2)\ldots}{(z - p_1)(z - p_2)\ldots}$$

- Zeros
  - Roots of the numerator polynomial
  - Locations in z-plane that make output zero

- Poles
  - Roots of the denominator polynomial
  - Locations in z-plane that make output infinity (unstable)
    - System is considered unstable if the ROC doesn't contain the unit circle (no DTFT exists)
    - Causal system → poles should be inside unit circle

EXAMPLE 2.10

- $H(z) = \frac{1}{L}\left[\frac{1-z^{-L}}{1-z^{-1}}\right]$
  - Notice this is a polynomial in z^{-1}
- Convert to polynomial in z to get all poles and zeros
- $H(z) = \frac{1}{L}\left[\frac{z^{L}-1}{z^{L}-z^{L-1}}\right] = \frac{1}{L}\left[\frac{z^{L}-1}{z^{L-1}(z-1)}\right]$
  - Poles
    - $(z-1) = 0 \rightarrow z = 1$
    - $z^{L-1} = 0 \rightarrow$ L-1 poles at $z = 0$
  - Zeros
    - $z^{L} - 1 = 0 \rightarrow z_l = e^{j\left(\frac{2\pi}{L}\right)l}$
    - L zeros even spaced around unit circle

- Matlab
  - `fvtool([1 0 0 0 0 0 0 0 -1], [1 -1]);`

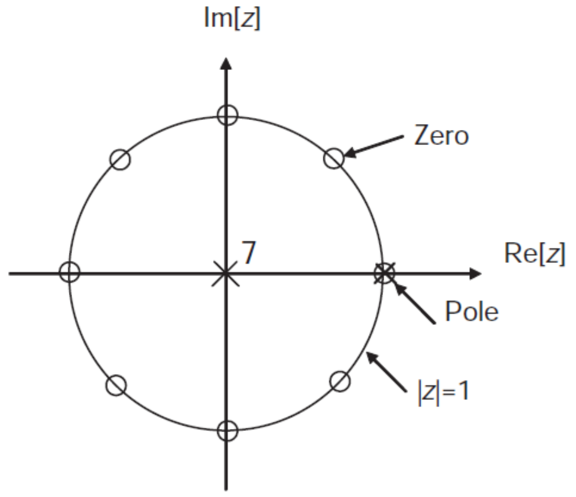


Figure 2.12 Pole–zero diagram of the moving-averaging filter, $L = 8$

FREQUENCY RESPONSE

- Discrete-time Fourier transform (DTFT)

$$H(\omega) = H(z)|_{z=e^{j\omega}} = \sum_{n=-\infty}^{\infty} h(n)e^{-j\omega n}$$

  - Evaluate transfer function along the unit circle $|z| = |e^{j\omega}|$

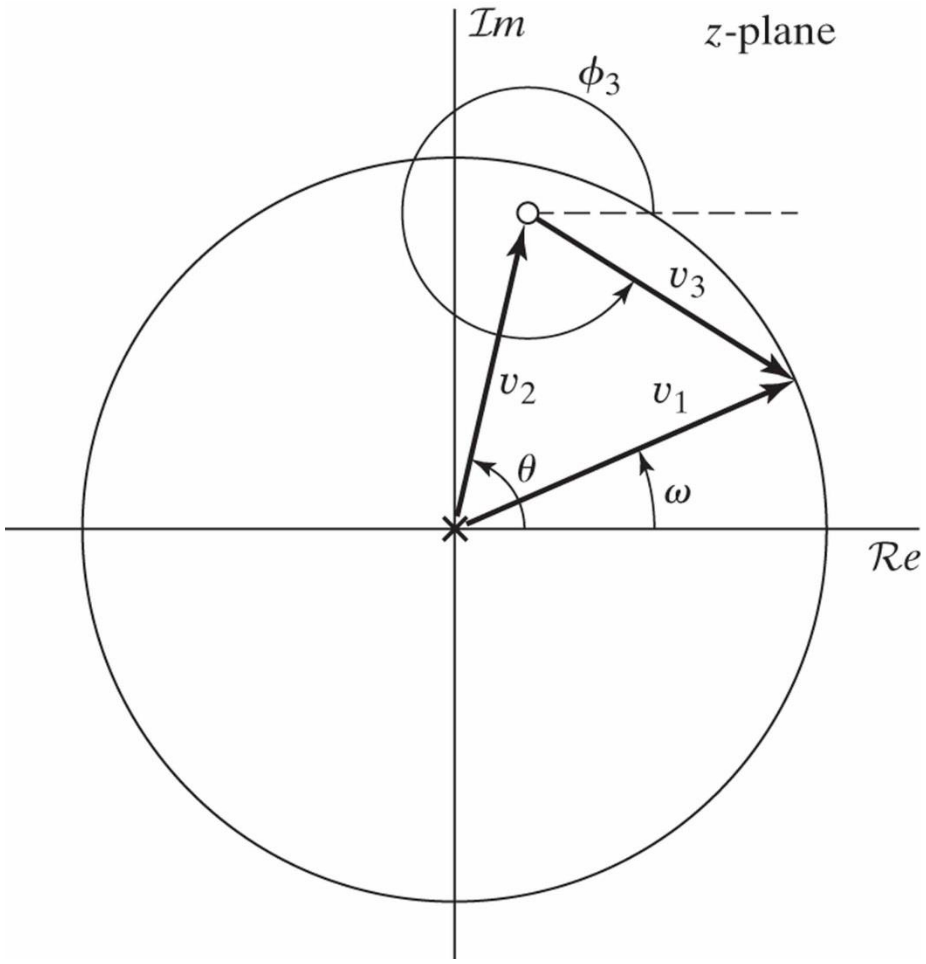
$$H(\omega) = |H(\omega)|e^{\angle H(\omega)}$$

$$|H(\omega)| = \sqrt{H(\omega)H^*(\omega)} \quad \angle H(\omega) = \arctan\left(\frac{\operatorname{Im} H(\omega)}{\operatorname{Re} H(\omega)}\right)$$

- Frequency response is periodic in 2π interval and symmetric
  - Only $[0, \pi]$ interval is required for evaluation

GRAPHICAL DTFT INTERPRETATION

- Poles
  - $|H(\omega)|$ gets larger closer to θ
- Zeros
  - $|H(\omega)|$ gets smaller closer to θ



- What does a highpass filter look like?
- What does a lowpass filter look like?

DISCRETE FOURIER TRANSFORM

- Notice the DTFT is a continuous function of ω
  - Requires an infinite number of samples to compute (infinite sum)
- DFT is a sampled version of the DTFT
  - Samples are taken at N equally spaced frequencies along unit circle
    - $\omega_k = \frac{2\pi k}{N}, \ k = 0,1,\ldots,N-1$

$$X(k) = X(\omega)\big|_{\omega=\frac{2\pi k}{N}} = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi k}{N}n}$$

  - n – time index
  - k – frequency index

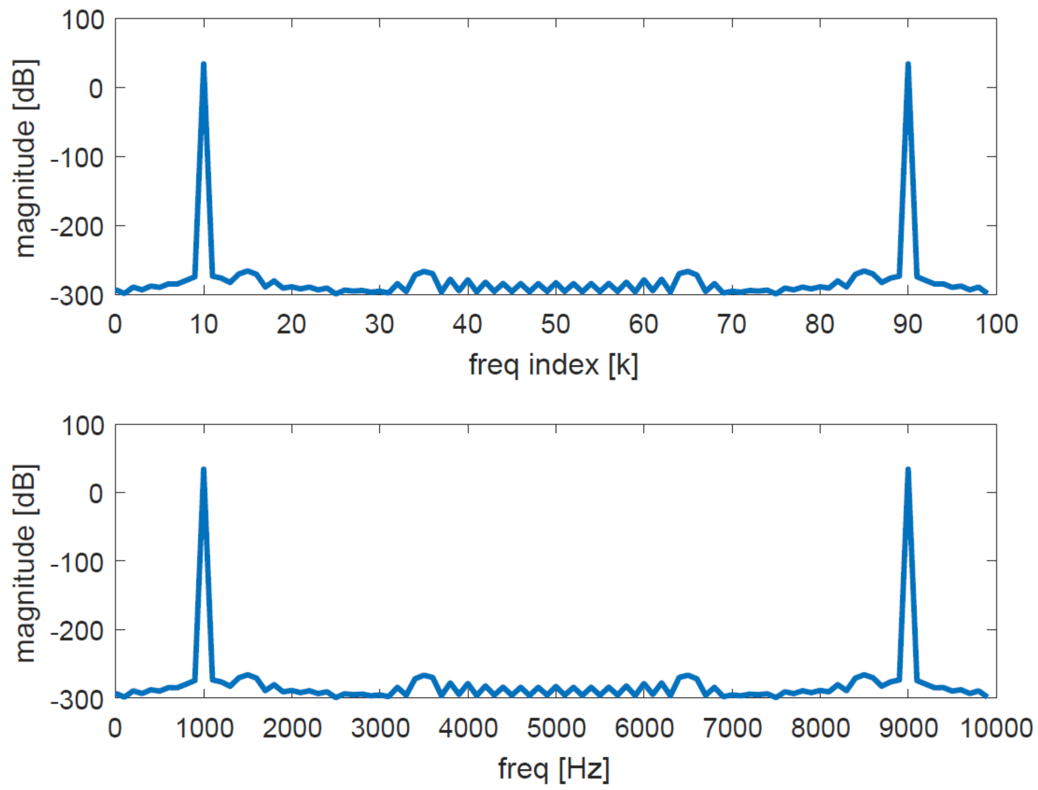
DFT IMPLEMENTATION

$$X(k) = X(\omega)\big|_{\omega=\frac{2\pi k}{N}} = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi k}{N}n}$$

- DFT can be computed very efficiently with the fast Fourier transform (FFT)
- Frequency resolution of DFT
  - $\Delta_\omega = \frac{2\pi}{N}, \quad \Delta_f = \frac{f_s}{N}$
- Analog frequency mapping
  - $f_k = k\Delta_f = \frac{kf_s}{N}, \quad k = 0,1,\ldots,N-1$
  - Nyquist frequency $\frac{f_s}{2}$ corresponds to $k = \frac{N}{2}$

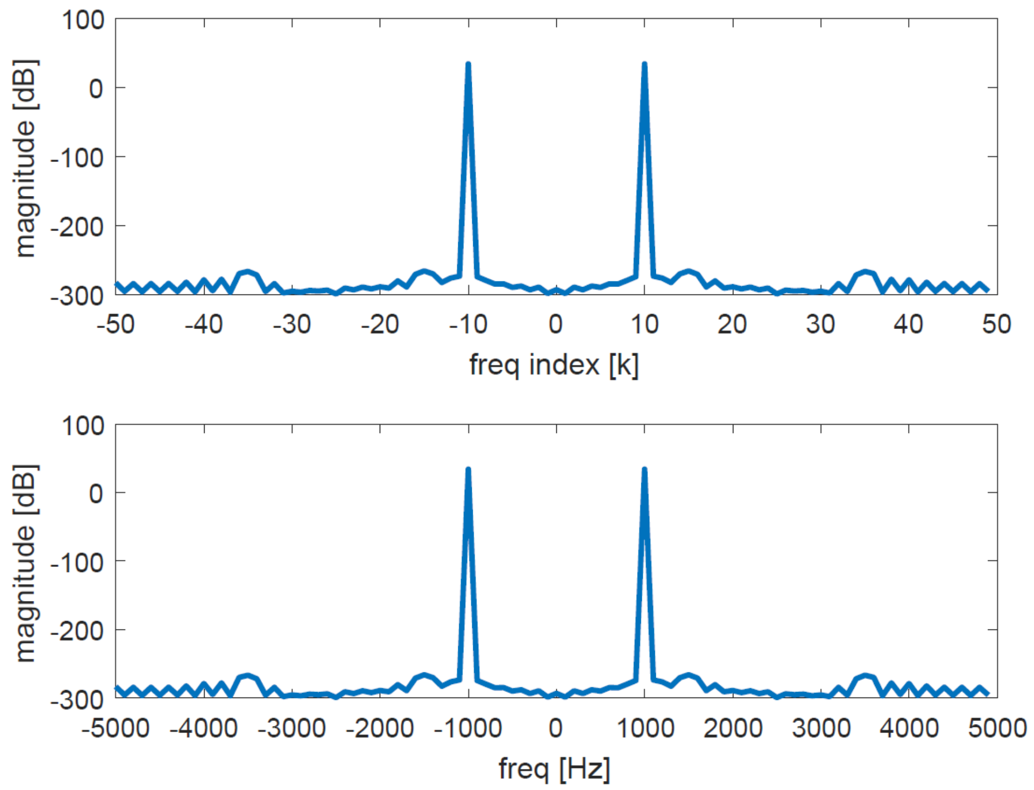
EXAMPLE 2.16

```
N = 100; A = 1; f=1000; fs = 10000;
n=0:N-1;
w = 2*pi*f/fs;

x = sin(w*n);
X = fft(x);
K = length(X);

h=figure;
subplot(2,1,1)
plot(0:K-1, 20*log10(abs(X)), 'linewidth', 2);
xlabel('freq index [k]'); ylabel('magnitude [dB]');
subplot(2,1,2)
%convert index to freq
f = (0:K-1) * fs/N;
plot(f, 20*log10(abs(X)), 'linewidth', 2);
xlabel('freq [Hz]'); ylabel('magnitude [dB]');
```



EXAMPLE 2.16 – SHIFTED FREQUENCIES

- ind = -K/2:K/2-1;
- Xs = fftshift(X)
- h=figure;
- subplot(2,1,1)
- plot(ind, 20*log10(abs(Xs)), 'linewidth', 2);
- xlabel('freq index [k]');
- ylabel('magnitude [dB]');
- subplot(2,1,2)
- %convert index to freq
- f = ind * fs/N;
- plot(f, 20*log10(abs(Xs)), 'linewidth', 2);
- xlabel('freq [Hz]');
- ylabel('magnitude [dB]');
-



INTRO TO RANDOM VARIABLES

CHAPTER 2.3

RANDOM VARIABLES

- Function that maps from a sample space to a real value
  - $x: S \to \mathbb{R}$
    - x – random variable (does not have a value)
    - S – sample space
- Cumulative distribution function (CDF)
  - $F(X) = P(x \leq X)$
    - E.g. probability $\{x \leq X\}$
- Probability density function (PDF)
  - $f(X) = \frac{dF(X)}{dX}$
    - $\int_{-\infty}^{\infty} f(X)dX = 1$
    - $P(X_1 < x \leq X_2) = F(X_2) - F(X_1)$

      $$P(X_1 < x \leq X_2) = \int_{X_1}^{X_2} f(X)dX$$
- Probability mass function (PMF)
  - For discrete x, takes values $X_i,\ i = 1, 2, 3, \ldots$
  - $p_i = P(x = X_i)$

UNIFORM RANDOM VARIABLE

- Variable takes on value in a range with equal probability
- $f(X) = \begin{cases} \frac{1}{X_2 - X_1} & X_1 \leq x \leq X_2 \\ 0 & else \end{cases}$
- Be sure you can calculate mean and variance
- Be aware that the book is a little sloppy in notation
  - RV x vs X

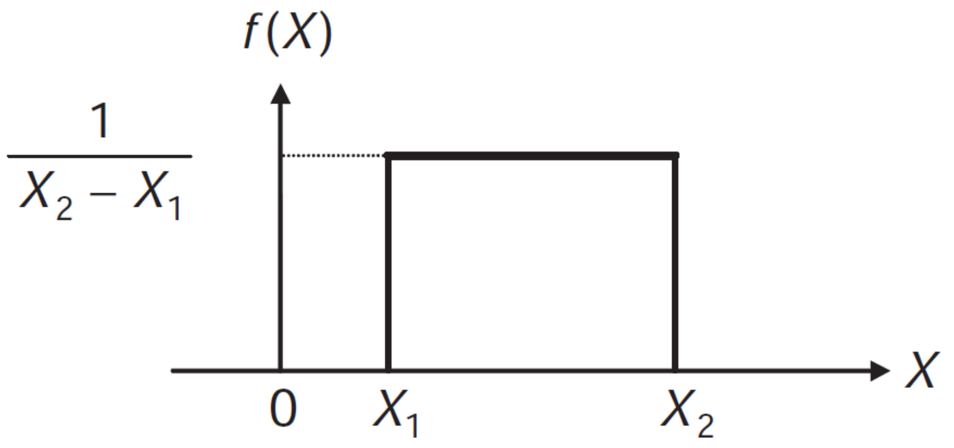


Figure 2.17 The uniform density function

STATISTICS OF RANDOM VARIABLES

- Expected value (mean)
  - $m_x = E[x]$ expectation operator
    - $m_x = \int_{-\infty}^{\infty} X f(X) dX$ continuous
    - $m_x = \sum_i X_i p_i$ discrete
- Can be can computed with `mean.m` and `var.m`
  - Read help for info on finite sample versions

- Variance (spread around mean)
  - $\sigma_x^2 = E[(x - m_x)^2] = E[x^2] - m_x^2$
  - Continuous
    - $\sigma_x^2 = \int_{-\infty}^{\infty} (X - m_x)^2 f(X) dX$
  - Discrete
    - $\sigma_x^2 = \sum_i p_i (X_i - m_x)^2$
  - For $m_x = 0$,
    - $\sigma_x^2 = E[x^2] = P_x$
      - Second moment, power

FIXED-POINT REPRESENTATION AND QUANTIZATION EFFECTS

CHAPTER 2.4

FIXED-POINT NUMERICAL EFFECTS

- Fractional numbers are represented in 2's complement with $B = M + 1$ bits
- $x = b_0 . b_1 b_2 \dots b_{M-1} b_M$

sign bit binary point msb lsb

  - $b_0 = \begin{cases} 0 & x \geq 0 \quad \text{positive} \\ 1 & x < 0 \quad \text{negative} \end{cases}$
  - Value $= -b_0 + \sum_{m=1}^{M} b_m 2^{-m}$
    - $-1 \leq x \leq (1 - 2^{-M})$
    - Unbalanced range with more negative than positive numbers

GENERAL FRACTIONAL FORMAT Qn.m

- $x = b_0 b_1 b_2 \ldots b_n . b_1 b_2 \ldots b_M$

  sign bit — integer — binary point — fractional

- Q format
  - Qn.m = Q#integer.#fraction
  - Larger n increases dynamic range but at cost of reduced precision (smallest fractional resolution)
  - b_0 is not counted as part of integer just as a sign-bit

- Example 2.25
- $x = 0100\ 1000\ 0001\ 1000b$

  $= 0x4818$
- Q0.15
  - $x = 2^{-1} + 2^{-4} + 2^{-11} + 2^{-12} = 0.56323$
- Q2.13
  - $x = 2^{1} + 2^{-2} + 2^{-9} + 2^{-10} = 2.25293$
- Q5.10
  - $x = 2^{4} + 2^{1} + 2^{-6} + 2^{-7} = 18.02344$

FINITE WORD LENGTH EFFECTS

1. Quantization errors
   - Signal quantization
   - Coefficient quantization
2. Arithmetic errors
   - Roundoff (truncation)
   - Overflow

SIGNAL QUANTIZATION

- ADC conversion of sampled signals to fixed levels
- Using Q15 and B bits
  - Dynamic range $-1 \le x < 1$
  - Quantization step
    - $\Delta = \frac{2}{2^B} = 2^{-B+1} = 2^{-M}$
- Quantization error
  - $e(n) = x(n) - x_B(n)$
    - $x_B(n) = Q[x(n)]$
  - $|e(n)| \le \frac{\Delta}{2} = 2^{-B}$ (rounding)
    - Error dependent on word length B
    - More bits for better resolution, less error (noise)
- Signal to quantization noise (SQNR)
  - $SQNR = \frac{\sigma_x^2}{\sigma_e^2} = 3.2^{2B}\sigma_x^2$
  - $SQNR = 4.77 + 6.02B + 10\log_{10}\sigma_x^2 \ dB$

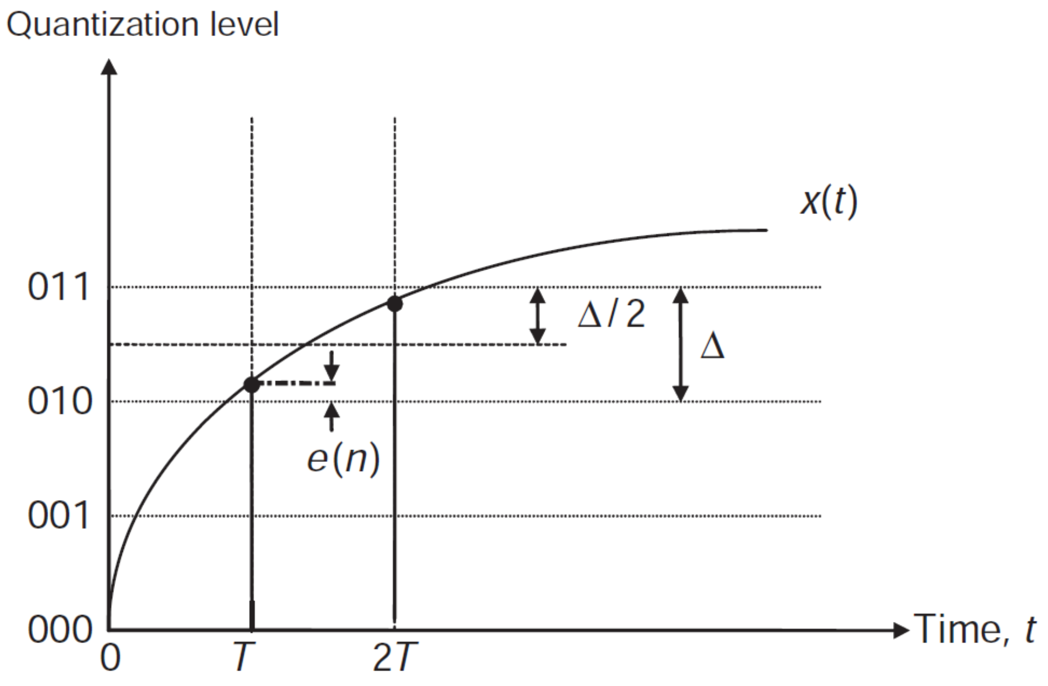


Figure 2.21 Quantization process related to a 3-bit ADC

COEFFICIENT QUANTIZATION

- Same error issues as for signals
- Results in movement of the locations of poles/zeros
  - Changes system function polynomials
  - Can lead to instability if poles go outside the unit circle
    - Generally, more a problem with IIR filters
- Can limit coefficient quantization effects by using lower-order filters
  - Use of cascade and parallel filter structures

ROUNDOFF NOISE

- A product must be represented in B bits by rounding (truncation)
  - $y(n) = \alpha x(n)$

    $2B$ bits　　B bits　　B bits

- Error model
  - $y(n) = Q[\alpha x(n)] = \alpha x(n) + e(n)$
    - $e(n)$ is uniformly distributed zero mean noise (rounding)

OVERFLOW AND SOLUTIONS

CHAPTER 2.5

OVERFLOW

- $y(n) = x_1(n) + x_2(n)$
  - $-1 \le x_i(n) < 1$
  - $-1 \le y(n) < 1$
- Overflow occurs when the sum cannot fit in the word container
- Signals need to be scaled to prevent overflow

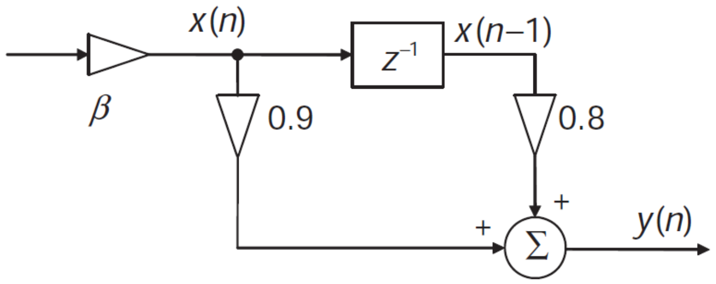


Figure 2.24 Block diagram of simple FIR filter with scaling factor β

- Notice: this reduces the SQNR
  - $SQNR = 10\log_{10}(\frac{\beta^2\sigma_x^2}{\sigma_e^2})$
  - $SQNR = 4.77 + 6.02B + 10\log_{10}\sigma_x^2 + \underbrace{20\log_{10}\beta}_{\text{Negative since } \beta < 1}\ dB$