ELEC 631 Digital Filters: II

Review of Prerequisite Material
Part III

Victoria, BC, Canada
Email: aantoniou@ieee.org

September 24, 2009

Review of the Prerequisite Material

- This presentation will deal with Part III of the review:
  - Spectral Interrelations
  - The Sampling Theorem

Review of the Prerequisite Material

- This presentation will deal with Part III of the review:
  - Spectral Interrelations
  - The Sampling Theorem
  - Processing of Continuous-Time Signals by Digital Filters
  - Introduction to the Design Process

Review of the Prerequisite Material

- This presentation will deal with Part III of the review:
  - Spectral Interrelations
  - The Sampling Theorem
  - Processing of Continuous-Time Signals by Digital Filters
  - Introduction to the Design Process
  - The Realization Process

- The Fourier transform of a continuous-time signal $x(t)$ is defined as

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\, dt$$

Spectral Interrelations

- The Fourier transform of a continuous-time signal $x(t)$ is defined as

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\, dt$$

- In general, $X(j\omega)$ is complex and can be written as

$$X(j\omega) = A(\omega)e^{j\phi(\omega)}$$

where

$$A(\omega) = |X(j\omega)| \quad \text{and} \quad \phi(\omega) = \arg X(j\omega)$$

Spectral Interrelations

- The Fourier transform of a continuous-time signal $x(t)$ is defined as

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\,dt$$

- In general, $X(j\omega)$ is complex and can be written as

$$X(j\omega) = A(\omega)e^{j\phi(\omega)}$$

where

$$A(\omega) = |X(j\omega)| \quad \text{and} \quad \phi(\omega) = \arg X(j\omega)$$

- Functions $A(\omega)$ and $\phi(\omega)$ are the *amplitude spectrum* and *phase spectrum* of the signal, respectively.

Spectral Interrelations

- The Fourier transform of a continuous-time signal $x(t)$ is defined as

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\, dt$$

- In general, $X(j\omega)$ is complex and can be written as

$$X(j\omega) = A(\omega)e^{j\phi(\omega)}$$

where

$$A(\omega) = |X(j\omega)| \quad \text{and} \quad \phi(\omega) = \arg X(j\omega)$$

- Functions $A(\omega)$ and $\phi(\omega)$ are the *amplitude spectrum* and *phase spectrum* of the signal, respectively.

- Together, the amplitude and phase spectrums constitute the *frequency spectrum*.

- Given a Fourier transform

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\,dt$$

signal $x(t)$ can be recovered from $X(j\omega)$ by using the *inverse Fourier transform* which is defined as

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\omega)e^{j\omega t}\,d\omega$$

Spectral Interrelations *Cont'd*

- Given a Fourier transform

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\,dt$$

signal $x(t)$ can be recovered from $X(j\omega)$ by using the *inverse Fourier transform* which is defined as

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\omega)e^{j\omega t}\,d\omega$$

- The Fourier transform and its inverse are often written in operator format as

$$X(j\omega) = \mathcal{F}x(t) \quad \text{and} \quad x(t) = \mathcal{F}^{-1}X(j\omega)$$

respectively.

- Given a Fourier transform

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\, dt$$

signal $x(t)$ can be recovered from $X(j\omega)$ by using the *inverse Fourier transform* which is defined as

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\omega)e^{j\omega t}\, d\omega$$

- The Fourier transform and its inverse are often written in operator format as

$$X(j\omega) = \mathcal{F}x(t) \quad \text{and} \quad x(t) = \mathcal{F}^{-1}X(j\omega)$$

respectively.

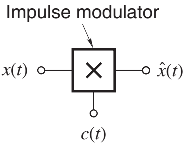
- An alternative shorthand notation used by Papoulis is

$$x(t) \leftrightarrow X(j\omega)$$

Spectral Interrelations *Cont'd*

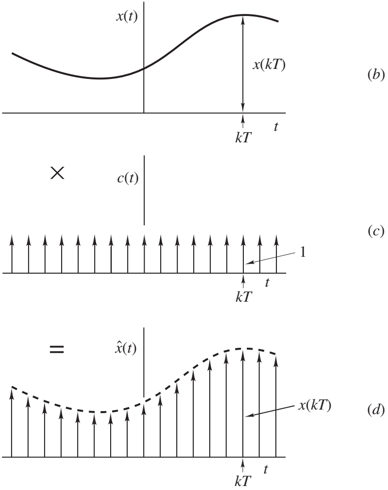
$x(t)$	$X(j\omega)$
$\delta(t)$	1
1	$2\pi\delta(\omega)$
$\delta(t-t_0)$	$e^{-j\omega t_0}$
$e^{j\omega_0 t}$	$2\pi\delta(\omega-\omega_0)$
$\cos\omega_0 t$	$\pi[\delta(\omega+\omega_0)+\delta(\omega-\omega_0)]$
$\sin\omega_0 t$	$j\pi[\delta(\omega+\omega_0)-\delta(\omega-\omega_0)]$

- An impulse-modulated signal, denoted as $\hat{x}(t)$, can be generated by sampling a continuous-time signal $x(t)$ using an ideal impulse modulator.



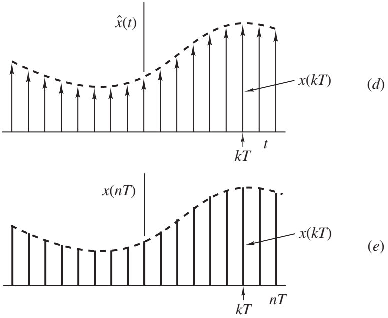
(a)

- The input and output of an impulse modulator are as follows:



Spectral Interrelations *Cont'd*

- Relation between impulse-modulated and discrete-time signals:



- An impulse-modulated signal is both a continuous-time as well as a sampled signal.

  This dual personality of impulse-modulated signals enables us to obtain spectral relationships between discrete-time and continuous-time signals.

Spectral Interrelations *Cont'd*

- An impulse-modulated signal is both a continuous-time as well as a sampled signal.

  This dual personality of impulse-modulated signals enables us to obtain spectral relationships between discrete-time and continuous-time signals.

- As a continuous-time signal, an impulse-modulated signal has a Fourier transform given by

$$\hat{X}(j\omega) = \sum_{n=-\infty}^{\infty} x(nT)e^{-j\omega nT}$$

  which is the frequency spectrum of the signal.

Spectral Interrelations *Cont'd*

- The frequency spectrum of a discrete-time signal is defined as the complex function obtained by evaluating the z transform on the unit circle of the the z plane, i.e.,

$$X_D(e^{j\omega T}) = X_D(z)\Big|_{z=e^{j\omega T}} = \sum_{n=-\infty}^{\infty} x(nT)e^{-j\omega nT}$$

Spectral Interrelations *Cont'd*

- The frequency spectrum of a discrete-time signal is defined as the complex function obtained by evaluating the z transform on the unit circle of the the z plane, i.e.,

$$X_D(e^{j\omega T}) = X_D(z)\Big|_{z=e^{j\omega T}} = \sum_{n=-\infty}^{\infty} x(nT)e^{-j\omega nT}$$

- Therefore,

$$\hat{X}(j\omega) = X_D(e^{j\omega T}) = \sum_{n=-\infty}^{\infty} x(nT)e^{-j\omega nT}$$

Spectral Interrelations *Cont'd*

- The frequency spectrum of a discrete-time signal is defined as the complex function obtained by evaluating the z transform on the unit circle of the the z plane, i.e.,

$$X_D(e^{j\omega T}) = X_D(z)\Big|_{z=e^{j\omega T}} = \sum_{n=-\infty}^{\infty} x(nT)e^{-j\omega nT}$$

- Therefore,

$$\hat{X}(j\omega) = X_D(e^{j\omega T}) = \sum_{n=-\infty}^{\infty} x(nT)e^{-j\omega nT}$$

- In effect, the *Fourier transform* of impulse-modulated signal $\hat{x}(t)$ *is numerically equal to the z transform* of the corresponding discrete-time signal $x(nT)$ evaluated on the unit circle $|z| = 1$.

- Poisson's summation formula:

$$\sum_{n=-\infty}^{\infty} x(nT) = \begin{cases} \frac{x(0+)}{2} + \frac{1}{T}\sum_{n=-\infty}^{\infty} X(jn\omega_s) & \text{if } x(t) = 0 \text{ for } t < 0 \\ \frac{1}{T}\sum_{n=-\infty}^{\infty} X(jn\omega_s) & \text{otherwise} \end{cases}$$

- Poisson's summation formula:

$$\sum_{n=-\infty}^{\infty} x(nT) = \begin{cases} \frac{x(0+)}{2} + \frac{1}{T}\sum_{n=-\infty}^{\infty} X(jn\omega_s) & \text{if } x(t) = 0 \text{ for } t < 0 \\ \frac{1}{T}\sum_{n=-\infty}^{\infty} X(jn\omega_s) & \text{otherwise} \end{cases}$$

- In effect, *the sum of the values of a signal at $t = nT$ for $-\infty < n < \infty$ is related to the sum of the values of the spectrum of the signal at $\omega = n\omega_s$ for $-\infty < n < \infty$.*

Spectral Interrelations *Cont'd*

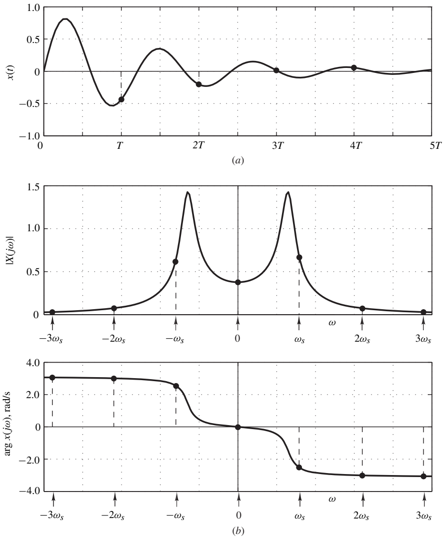
- Poisson's summation formula:

$$\sum_{n=-\infty}^{\infty} x(nT) = \begin{cases} \frac{x(0+)}{2} + \frac{1}{T}\sum_{n=-\infty}^{\infty} X(jn\omega_s) & \text{if } x(t) = 0 \text{ for } t < 0 \\ \frac{1}{T}\sum_{n=-\infty}^{\infty} X(jn\omega_s) & \text{otherwise} \end{cases}$$

- In effect, *the sum of the values of a signal at $t = nT$ for $-\infty < n < \infty$ is related to the sum of the values of the spectrum of the signal at $\omega = n\omega_s$ for $-\infty < n < \infty$*.

- Poisson's summation formula for the case of a right-sided signal is illustrated in the next slide.

Spectral Interrelations *Cont'd*



- Using Poisson's formula, the following formulas can be obtained:

  If $x(t) = 0$ for $t < 0$ (right-sided signal):

  $$\hat{X}(j\omega) = X_D(e^{j\omega T}) = \frac{x(0+)}{2} + \frac{1}{T}\sum_{n=-\infty}^{\infty} X(j\omega + jn\omega_s)$$

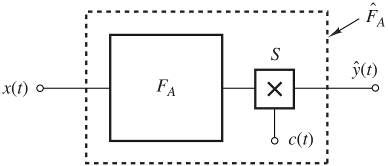
  Otherwise (two-sided signal):

  $$\hat{X}(j\omega) = X_D(e^{j\omega T}) = \frac{1}{T}\sum_{n=-\infty}^{\infty} X(j\omega + jn\omega_s)$$

- In effect, the frequency spectrum of the impulse-modulated signal $\hat{x}(t)$ is *numerically equal* to the frequency spectrum of discrete-time signal $x(nT)$ and the two can be *uniquely determined* from the frequency spectrum of the continuous-time signal $x(t)$, namely, $X(j\omega)$.

Spectral Interrelations *Cont'd*

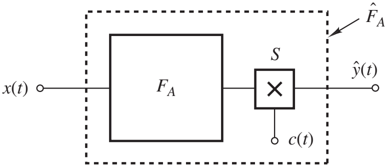
- Given an analog filter F_A with an impulse response $h_A(t)$ and transfer function $H_A(s)$, an *impulse-modulated filter*, designated as $\hat{F}_A$, can be constructed by using an impulse modulator as shown.



Spectral Interrelations *Cont'd*

- Given an analog filter F_A with an impulse response $h_A(t)$ and transfer function $H_A(s)$, an *impulse-modulated filter*, designated as $\hat{F}_A$, can be constructed by using an impulse modulator as shown.

- The impulse response of such a filter, designated as $\hat{h}_A(t)$, is an impulse-modulated signal.



Spectral Interrelations *Cont'd*

- Analog filters are causal systems and as a result

$$h(t) = \hat{h}(t) = 0 \quad \text{for} \quad t < 0$$

- Analog filters are causal systems and as a result

$$h(t) = \hat{h}(t) = 0 \quad \text{for} \quad t < 0$$

- Using Poisson's summation formula, the spectrum of the impulse response can be expressed as

$$\hat{H}(j\omega) = H_D(e^{j\omega T}) = \frac{h(0+)}{2} + \frac{1}{T} \sum_{n=-\infty}^{\infty} H(j\omega + jn\omega_s)$$

where $h(0+)$ is the initial value of the impulse response of the analog filter.

• • •

$$\hat{H}(j\omega) = H_D(e^{j\omega T}) = \frac{h(0+)}{2} + \frac{1}{T}\sum_{n=-\infty}^{\infty} H(j\omega + jn\omega_s)$$

- The frequency spectrum of the impulse response of a filter is numerically equal to the frequency response of the filter.

...

$$\hat{H}(j\omega) = H_D(e^{j\omega T}) = \frac{h(0+)}{2} + \frac{1}{T}\sum_{n=-\infty}^{\infty} H(j\omega + jn\omega_s)$$

- The frequency spectrum of the impulse response of a filter is numerically equal to the frequency response of the filter.

- If we let $j\omega = s$ and $e^{j\omega T} = z$ in the above equation, then we get the important relation

$$\hat{H}(s) = H_D(z) = \frac{h(0+)}{2} + \frac{1}{T}\sum_{n=-\infty}^{\infty} H(s + jn\omega_s)$$

In effect, an impulse-modulated filter can be represented by *a continuous-time transfer function $\hat{H}(s)$* just like an analog filter and by *a discrete-time transfer function* $H(z)$ just like a digital filter.

Spectral Interrelations *Cont'd*

...

$$\hat{H}(s) = H_D(z) = \frac{h(0+)}{2} + \frac{1}{T}\sum_{n=-\infty}^{\infty} H(s + jn\omega_s)$$

- Because of this *dual personality* of impulse-modulated filters, given an analog filter with a continuous-time transfer function $H(s)$, a digital filter with a discrete-time transfer function $H(z)$ can be deduced.

...

$$\hat{H}(s) = H_D(z) = \frac{h(0+)}{2} + \frac{1}{T}\sum_{n=-\infty}^{\infty} H(s + jn\omega_s)$$

- Because of this *dual personality* of impulse-modulated filters, given an analog filter with a continuous-time transfer function $H(s)$, a digital filter with a discrete-time transfer function $H(z)$ can be deduced.
- Under certain conditions, the frequency response of the derived digital filter is approximately the same as that of the analog filter to within a constant.

• • •

$$\hat{H}(s) = H_D(z) = \frac{h(0+)}{2} + \frac{1}{T}\sum_{n=-\infty}^{\infty} H(s + jn\omega_s)$$

- Because of this *dual personality* of impulse-modulated filters, given an analog filter with a continuous-time transfer function $H(s)$, a digital filter with a discrete-time transfer function $H(z)$ can be deduced.
- Under certain conditions, the frequency response of the derived digital filter is approximately the same as that of the analog filter to within a constant.
- Therefore, analog filters can be used to design digital filters, as will be shown in Chap. 11.

The Sampling Theorem

- The sampling theorem states:

  A bandlimited signal $x(t)$ for which

  $$X(j\omega) = 0 \quad \text{for} \quad |\omega| \geq \frac{\omega_s}{2}$$

  where $\omega_s = 2\pi/T$, can be uniquely determined from its values $x(nT)$.

- By virtue of Poison's summation formula, i.e.,

$$\hat{X}(j\omega) = \frac{1}{T} \sum_{n=-\infty}^{\infty} X(j\omega + jn\omega_s)$$

impulse modulation, which is essentially a sampling process, will produce sidebands.

The Sampling Theorem *Cont'd*

- By virtue of Poison's summation formula, i.e.,

$$\hat{X}(j\omega) = \frac{1}{T} \sum_{n=-\infty}^{\infty} X(j\omega + jn\omega_s)$$

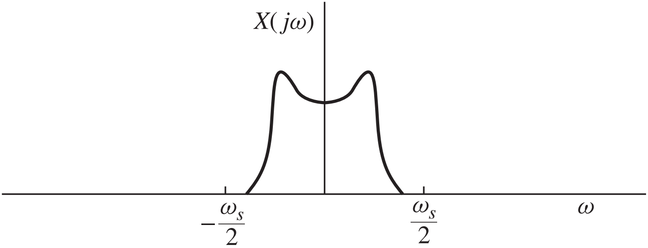
  impulse modulation, which is essentially a sampling process, will produce sidebands.

- If the signal is bandlimited such that

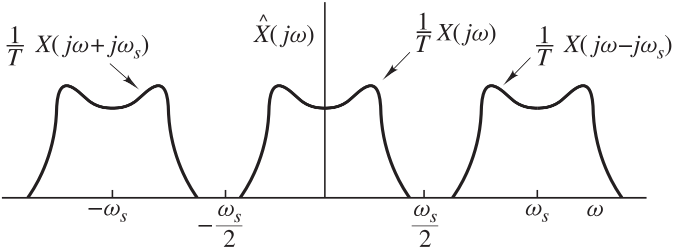
$$X(j\omega) = 0 \quad \text{for} \quad |\omega| \geq \frac{\omega_s}{2}$$

  then the sidebands will not overlap as shown in the next slide.

The Sampling Theorem *Cont'd*



(*a*)



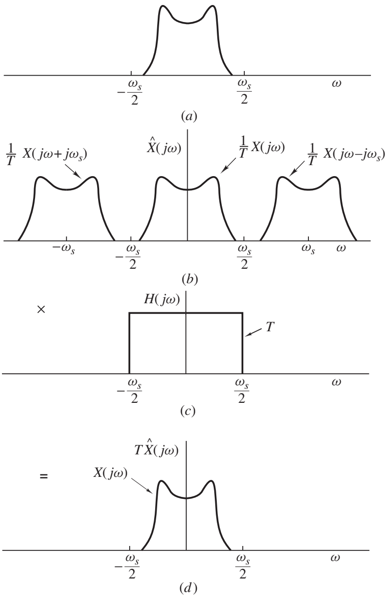
(*b*)

- Therefore, if the signal is filtered with an ideal lowpass filter that rejects all the frequencies outside the baseband, the original signal will be recovered.

The Sampling Theorem *Cont'd*

- Therefore, if the signal is filtered with an ideal lowpass filter that rejects all the frequencies outside the baseband, the original signal will be recovered.
- Such a filter would have a frequency response

$$H_{LP}(j\omega) = \begin{cases} T & \text{if } |\omega| < \omega_s/2 \\ 0 & \text{otherwise} \end{cases}$$

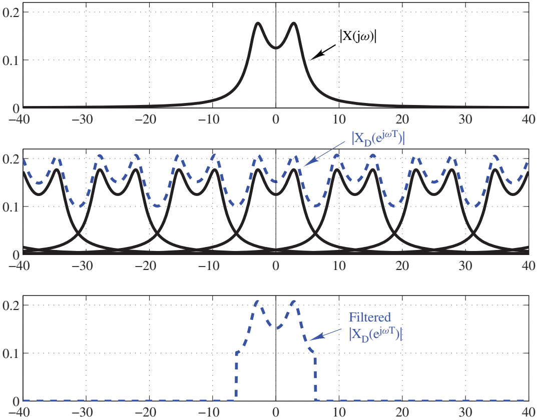


$-\frac{\omega_s}{2}$
$\frac{\omega_s}{2}$
ω
(a)
$\frac{1}{T}X(j\omega+j\omega_s)$
$\hat{X}(j\omega)$
$\frac{1}{T}X(j\omega)$
$\frac{1}{T}X(j\omega-j\omega_s)$
$-\omega_s$
$-\frac{\omega_s}{2}$
$\frac{\omega_s}{2}$
ω_s
ω
(b)
$\times$
$H(j\omega)$
T
$-\frac{\omega_s}{2}$
$\frac{\omega_s}{2}$
ω
(c)
$T\hat{X}(j\omega)$
=
$X(j\omega)$
$-\frac{\omega_s}{2}$
$\frac{\omega_s}{2}$
ω
(d)

- If the continuous-time signal does not satisfy the required condition of the sampling theorem, i.e., if the spectrum of the signal is not zero at frequencies equal to or greater than $\omega_s/2$, then so-called *aliasing distortion* will be introduced as illustrated in the next three slides.

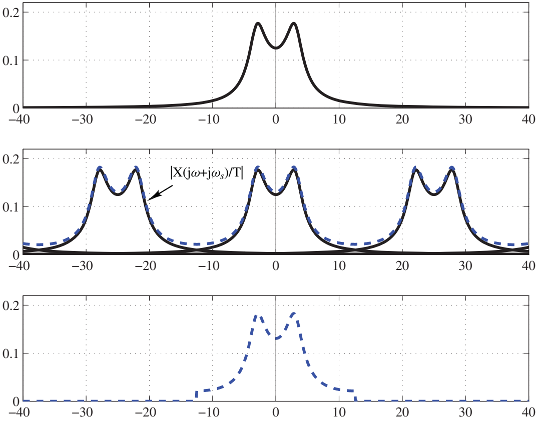
The Sampling Theorem *Cont'd*

- With a sampling frequency of 12.5 rad/s, the discrepancy between the solid and dashed curves in the figure is quite large.



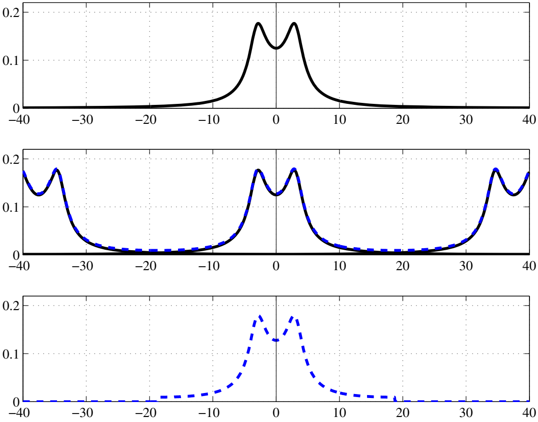
The Sampling Theorem *Cont'd*

- As the sampling frequency is increased to 25, the sidebands will move away from the baseband and the discrepancy between the solid and dashed curves will decrease quite a bit as shown.

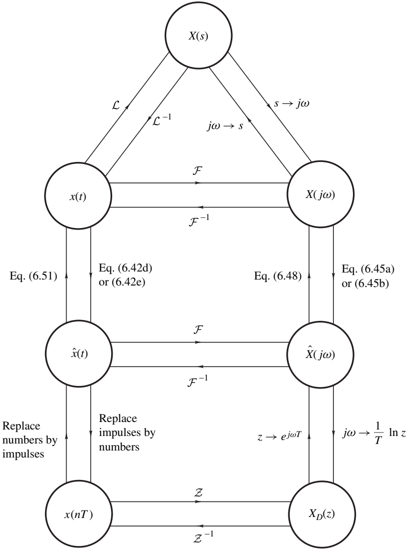


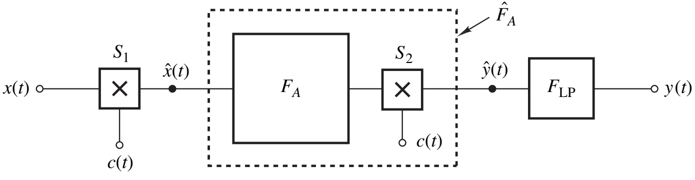
The Sampling Theorem *Cont'd*

- A further increase in the sampling frequency to 40 rad/s will render the discrepancy between the solid and dashed curves almost negligible as can be seen.

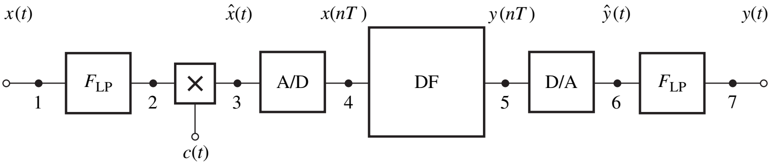


- In other words, the less the overlap between side bands, the less the aliasing distortion.


$X(s)$
$\mathcal{L}$
$\mathcal{L}^{-1}$
$j\omega \to s$
$s \to j\omega$
$x(t)$
$\mathcal{F}$
$\mathcal{F}^{-1}$
$X(j\omega)$
Eq. (6.51)
Eq. (6.42d) or (6.42e)
Eq. (6.48)
Eq. (6.45a) or (6.45b)
$\hat{x}(t)$
$\mathcal{F}$
$\mathcal{F}^{-1}$
$\hat{X}(j\omega)$
Replace numbers by impulses
Replace impulses by numbers
$z \to e^{j\omega T}$
$j\omega \to \frac{1}{T} \ln z$
$x(nT)$
$\mathcal{Z}$
$\mathcal{Z}^{-1}$
$X_D(z)$


S_1
$x(t)$
$\hat{x}(t)$
$c(t)$
F_A
S_2
$c(t)$
$\hat{F}_A$
$\hat{y}(t)$
F_{LP}
$y(t)$

(a)


$x(t)$
$\hat{x}(t)$
$x(nT)$
$y(nT)$
$\hat{y}(t)$
$y(t)$
1
F_{LP}
2
$c(t)$
3
A/D
4
DF
5
D/A
6
F_{LP}
7

(b)

Introduction to the Design Process

- In general the design of digital filters involves the following steps:

- In general the design of digital filters involves the following steps:
  - Approximation

Introduction to the Design Process

- In general the design of digital filters involves the following steps:
  - Approximation
  - Realization

Introduction to the Design Process

- In general the design of digital filters involves the following steps:
  - Approximation
  - Realization
  - Study of arithmetic errors

Introduction to the Design Process

- In general the design of digital filters involves the following steps:
  - Approximation
  - Realization
  - Study of arithmetic errors
  - Implementation

- The *approximation step* is the process of generating a transfer function that would satisfy the desired specifications which may concern the amplitude or phase response or even the time-domain response of the filter.

Introduction to the Design Process *Cont'd*

- The *approximation step* is the process of generating a transfer function that would satisfy the desired specifications which may concern the amplitude or phase response or even the time-domain response of the filter.

- The available methods for the solution of the approximation problem can be classified as *direct* or *indirect*.

Introduction to the Design Process *Cont'd*

- The *approximation step* is the process of generating a transfer function that would satisfy the desired specifications which may concern the amplitude or phase response or even the time-domain response of the filter.

- The available methods for the solution of the approximation problem can be classified as *direct* or *indirect*.

- In direct methods, the problem is solved directly in the z domain.

Introduction to the Design Process *Cont'd*

- The *approximation step* is the process of generating a transfer function that would satisfy the desired specifications which may concern the amplitude or phase response or even the time-domain response of the filter.

- The available methods for the solution of the approximation problem can be classified as *direct* or *indirect*.

- In direct methods, the problem is solved directly in the z domain.

- In indirect methods, a continuous-time transfer function is first obtained and then converted into a corresponding discrete-time transfer function.

Introduction to the Design Process *Cont'd*

- The *approximation step* is the process of generating a transfer function that would satisfy the desired specifications which may concern the amplitude or phase response or even the time-domain response of the filter.
- The available methods for the solution of the approximation problem can be classified as *direct* or *indirect*.
- In direct methods, the problem is solved directly in the z domain.
- In indirect methods, a continuous-time transfer function is first obtained and then converted into a corresponding discrete-time transfer function.
- Nonrecursive filters are always designed through direct methods whereas recursive filters can be designed either through direct or indirect methods.

- Approximation methods can also be classified as *closed-form* or *iterative*.

Introduction to the Design Process *Cont'd*

- Approximation methods can also be classified as *closed-form* or *iterative*.
- In closed-form methods, the problem is solved through a small number of design steps using a set of closed-form formulas.

Introduction to the Design Process *Cont'd*

- Approximation methods can also be classified as *closed-form* or *iterative*.
- In closed-form methods, the problem is solved through a small number of design steps using a set of closed-form formulas.
- In iterative methods, an initial solution is assumed and through the application of optimization methods a series of progressively improved solutions are obtained until some design criterion is satisfied.

- In general, the designer is interested in approximation methods that

- In general, the designer is interested in approximation methods that
  - are simple,

- In general, the designer is interested in approximation methods that
  - are simple,
  - are reliable,

Introduction to the Design Process *Cont'd*

- In general, the designer is interested in approximation methods that
  - are simple,
  - are reliable,
  - yield precise designs,

- In general, the designer is interested in approximation methods that
  - are simple,
  - are reliable,
  - yield precise designs,
  - require minimal computation effort, and so on.

- The *realization* or *synthesis* of a digital filter is the process of generating a digital-filter network or structure from the transfer function or some other characterization of the filter.

Introduction to the Design Process *Cont'd*

- The *realization* or *synthesis* of a digital filter is the process of generating a digital-filter network or structure from the transfer function or some other characterization of the filter.
- The network obtained is said to be the *realization* of the transfer function.

Introduction to the Design Process *Cont'd*

- The *realization* or *synthesis* of a digital filter is the process of generating a digital-filter network or structure from the transfer function or some other characterization of the filter.
- The network obtained is said to be the *realization* of the transfer function.
- As for approximation methods, realization methods can be classified as *direct* or *indirect*.

Introduction to the Design Process *Cont'd*

- The *realization* or *synthesis* of a digital filter is the process of generating a digital-filter network or structure from the transfer function or some other characterization of the filter.

- The network obtained is said to be the *realization* of the transfer function.

- As for approximation methods, realization methods can be classified as *direct* or *indirect*.

- In direct methods, the filter structure is obtained directly from a given discrete-time transfer function whereas in indirect realizations it is obtained indirectly from an equivalent continuous-time transfer function.

- Many realization methods have been proposed in the past that lead to digital-filter structures of varying complexity and properties.

Introduction to the Design Process *Cont'd*

- Many realization methods have been proposed in the past that lead to digital-filter structures of varying complexity and properties.
- The designer is usually interested in realizations that

- Many realization methods have been proposed in the past that lead to digital-filter structures of varying complexity and properties.
- The designer is usually interested in realizations that
  - are easy to implement in very-large-scale integrated (VLSI) circuit form,

- Many realization methods have been proposed in the past that lead to digital-filter structures of varying complexity and properties.
- The designer is usually interested in realizations that
  - are easy to implement in very-large-scale integrated (VLSI) circuit form,
  - require the minimum number of unit delays, adders, and multipliers,

- Many realization methods have been proposed in the past that lead to digital-filter structures of varying complexity and properties.
- The designer is usually interested in realizations that
  - are easy to implement in very-large-scale integrated (VLSI) circuit form,
  - require the minimum number of unit delays, adders, and multipliers,
  - are not seriously affected by the use of finite-precision arithmetic in the implementation, and so on.

Introduction to the Design Process *Cont'd*

- Designs of all types from that of a refrigerator to an electrical drill entail *imperfections* of various sorts brought about by modeling inaccuracies, component tolerances, unusual or unexpected nonlinear effects, and so on.

Introduction to the Design Process *Cont'd*

- Designs of all types from that of a refrigerator to an electrical drill entail *imperfections* of various sorts brought about by modeling inaccuracies, component tolerances, unusual or unexpected nonlinear effects, and so on.
- A design will be approved to the extent that design imperfections do not violate the desired specifications.

Introduction to the Design Process *Cont'd*

- Designs of all types from that of a refrigerator to an electrical drill entail *imperfections* of various sorts brought about by modeling inaccuracies, component tolerances, unusual or unexpected nonlinear effects, and so on.
- A design will be approved to the extent that design imperfections do not violate the desired specifications.
- In digital filters and digital systems in general, most imperfections are caused by *numerical imprecision* of some form.

  It is, therefore, important to study the ways in which numerical imprecision can manifest itself.

- During the approximation step, the coefficients of the transfer function are determined to a high degree of precision.

- During the approximation step, the coefficients of the transfer function are determined to a high degree of precision.
- In practice, however, digital hardware have finite precision that depends

- During the approximation step, the coefficients of the transfer function are determined to a high degree of precision.
- In practice, however, digital hardware have finite precision that depends
  - on the length of registers used to store numbers

- During the approximation step, the coefficients of the transfer function are determined to a high degree of precision.
- In practice, however, digital hardware have finite precision that depends
  - on the length of registers used to store numbers
  - the type of number system used (e.g., signed-magnitude, two's complement)

Introduction to the Design Process *Cont'd*

- During the approximation step, the coefficients of the transfer function are determined to a high degree of precision.
- In practice, however, digital hardware have finite precision that depends
  - on the length of registers used to store numbers
  - the type of number system used (e.g., signed-magnitude, two's complement)
  - the type of arithmetic used (e.g., fixed-point or floating-point), etc.

Introduction to the Design Process *Cont'd*

- In order to accommodate filter coefficients in registers, they must be quantized (e.g., rounded or truncated).

- In order to accommodate filter coefficients in registers, they must be quantized (e.g., rounded or truncated).
- When the transfer function coefficients are quantized, errors are introduced in the amplitude and phase responses of the filter, which are commonly referred to as *quantization errors*.

Introduction to the Design Process *Cont'd*

- Quantization errors can cause a digital filter to violate the required specifications and in extreme cases even to become unstable.

Introduction to the Design Process *Cont'd*

- Quantization errors can cause a digital filter to violate the required specifications and in extreme cases even to become unstable.
- Like the filter coefficients, the signals to be processed as well as the internal signals of a digital filter (e.g., the products generated by multipliers) must also be quantized.

Introduction to the Design Process *Cont'd*

- Quantization errors can cause a digital filter to violate the required specifications and in extreme cases even to become unstable.
- Like the filter coefficients, the signals to be processed as well as the internal signals of a digital filter (e.g., the products generated by multipliers) must also be quantized.
- Errors introduced by the quantization of signals are actually *noise sources* and, as a consequence, they can have a dramatic effect on the processed signal.

Introduction to the Design Process *Cont'd*

- Quantization errors can cause a digital filter to violate the required specifications and in extreme cases even to become unstable.
- Like the filter coefficients, the signals to be processed as well as the internal signals of a digital filter (e.g., the products generated by multipliers) must also be quantized.
- Errors introduced by the quantization of signals are actually *noise sources* and, as a consequence, they can have a dramatic effect on the processed signal.
- In short, the *effects of arithmetic errors* on the performance of the filter must be investigated and ways must be found to mitigate any problems associated with numerical imprecision.

Introduction to the Design Process *Cont'd*

- The *implementation* of a digital filter can assume two forms, namely, *software* or *hardware*.

Introduction to the Design Process *Cont'd*

- The *implementation* of a digital filter can assume two forms, namely, *software* or *hardware*.
- A software implementation involves the simulation of the filter network or difference equation on a general-purpose digital computer, workstation, or DSP chip.

Introduction to the Design Process *Cont'd*

- The *implementation* of a digital filter can assume two forms, namely, *software* or *hardware*.
- A software implementation involves the simulation of the filter network or difference equation on a general-purpose digital computer, workstation, or DSP chip.
- A hardware implementation involves the conversion of the filter network into a dedicated piece of hardware.

Introduction to the Design Process *Cont'd*

- The choice of implementation is usually critically dependent on the application at hand.

Introduction to the Design Process *Cont'd*

- The choice of implementation is usually critically dependent on the application at hand.
- In *nonreal-time* applications where a record of the data to be processed is available in a database, a software implementation would be entirely satisfactory.

Introduction to the Design Process *Cont'd*

- The choice of implementation is usually critically dependent on the application at hand.
- In *nonreal-time* applications where a record of the data to be processed is available in a database, a software implementation would be entirely satisfactory.
- In *real-time* applications, however, where data must be processed at a very high rate, e.g., in communication systems, a hardware implementation is mandatory.

Introduction to the Design Process *Cont'd*

- The choice of implementation is usually critically dependent on the application at hand.
- In *nonreal-time* applications where a record of the data to be processed is available in a database, a software implementation would be entirely satisfactory.
- In *real-time* applications, however, where data must be processed at a very high rate, e.g., in communication systems, a hardware implementation is mandatory.
- Often the best engineering solution might be partially in terms of software and partially in terms of hardware since software and hardware are highly exchangeable nowadays.

- The natural order of the four basic design steps is as stated in the preceding discussion, namely, approximation, realization, study of imperfections, and implementation.

Introduction to the Design Process *Cont'd*

- The natural order of the four basic design steps is as stated in the preceding discussion, namely, approximation, realization, study of imperfections, and implementation.
- However, realization is the easiest to learn and for this reason it was treated in Digital Filters: I.

Introduction to the Design Process *Cont'd*

- The natural order of the four basic design steps is as stated in the preceding discussion, namely, approximation, realization, study of imperfections, and implementation.
- However, realization is the easiest to learn and for this reason it was treated in Digital Filters: I.
- The approximation step, study of errors, and other design considerations will be discussed in detail in later presentations as part of Digital Filters: II.

Introduction to the Design Process *Cont'd*

- The natural order of the four basic design steps is as stated in the preceding discussion, namely, approximation, realization, study of imperfections, and implementation.
- However, realization is the easiest to learn and for this reason it was treated in Digital Filters: I.
- The approximation step, study of errors, and other design considerations will be discussed in detail in later presentations as part of Digital Filters: II.
- A discussion on the implementation of digital filters can be found at the end of Chap. 8.

Realization

- A great collection of direct and indirect realization methods have evolved over the past 40 years.

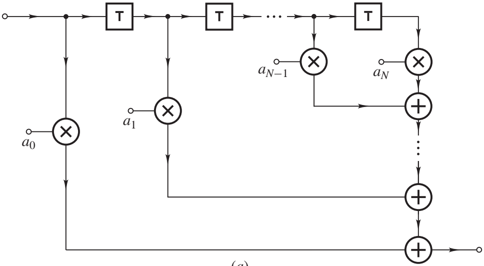
Realization

- A great collection of direct and indirect realization methods have evolved over the past 40 years.
- In direct methods, the transfer function is put in some form that enables the identification of an interconnection of elemental digital-filter subnetworks.

Realization

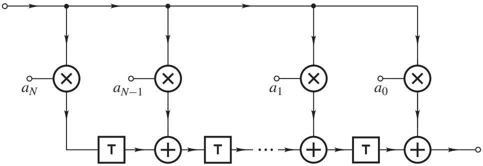
- A great collection of direct and indirect realization methods have evolved over the past 40 years.
- In direct methods, the transfer function is put in some form that enables the identification of an interconnection of elemental digital-filter subnetworks.
- Some of these methods are as follows:
  - Direct
  - Direct canonic
  - State-space
  - Lattice
  - Parallel
  - Cascade

- Nonrecursive realizations:



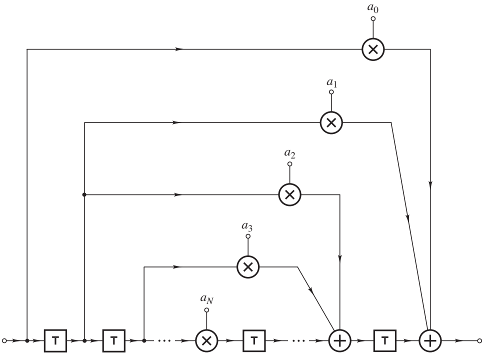
(a)

Realization *Cont'd*



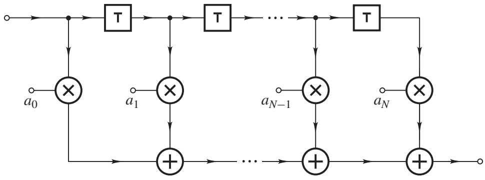
(b)

Realization *Cont'd*



(c)

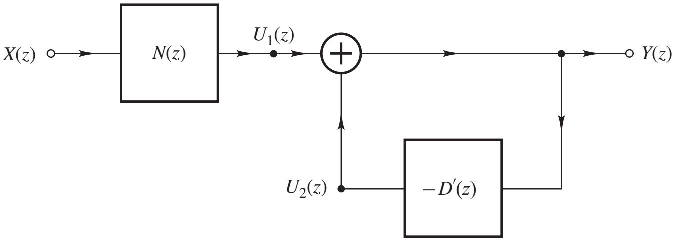
Realization *Cont'd*



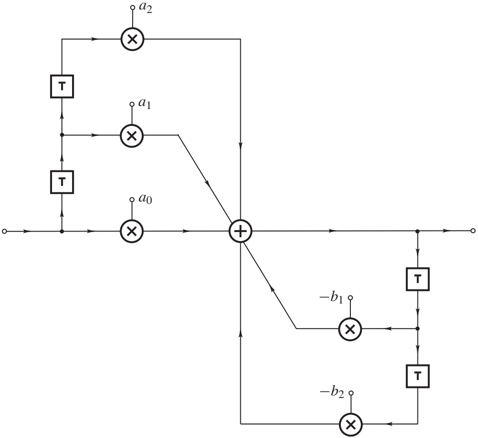
(*d*)

Realization *Cont'd*

- Recursive realizations:

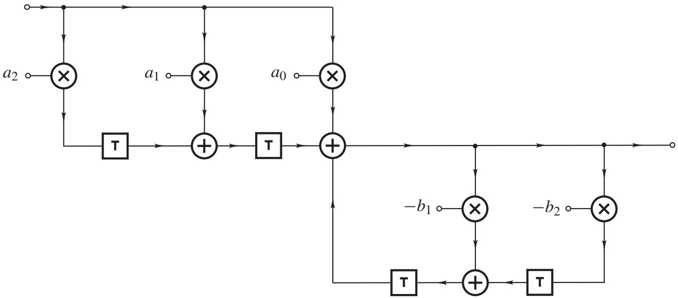


- Second-order direct recursive realization:



(*a*)

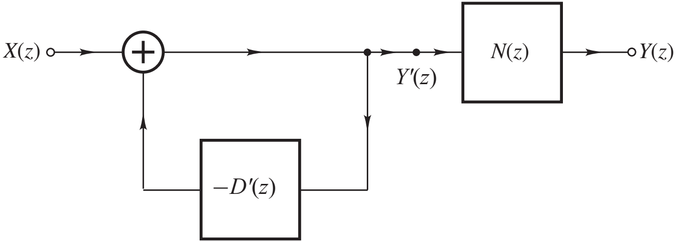
- Another second-order direct recursive realization:



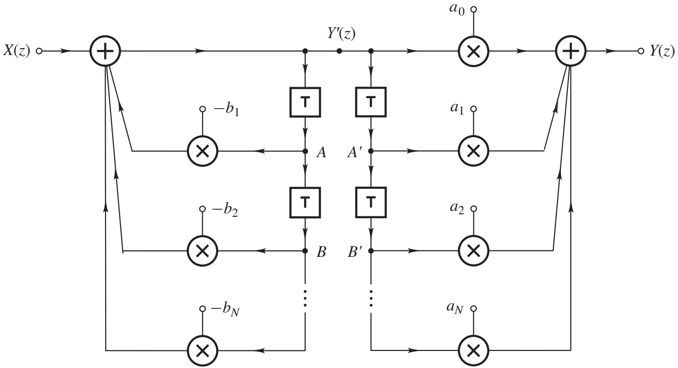
(*b*)

Realization *Cont'd*

- Direct Canonic Realizations:

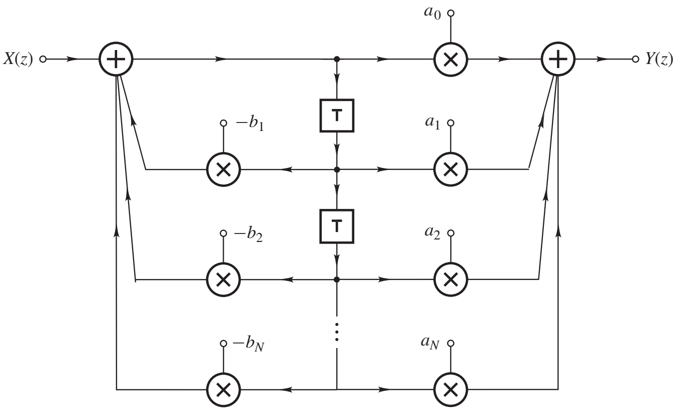


(a)



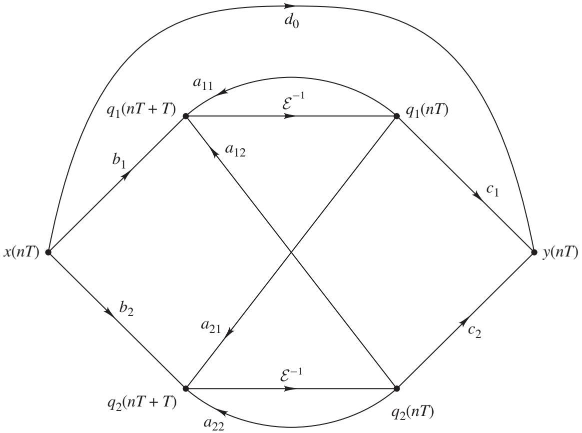
(b)

Realization *Cont'd*

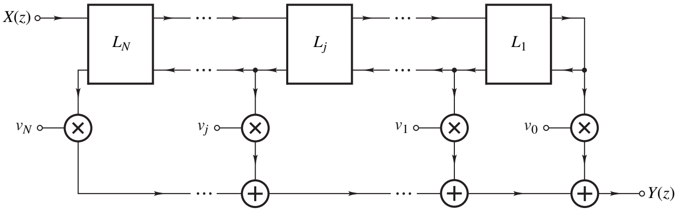


Realization *Cont'd*

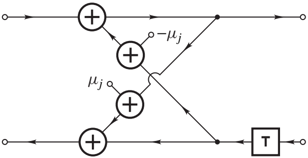
- State-space realization:



- Lattice Realization:

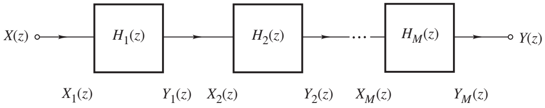


(*a*)

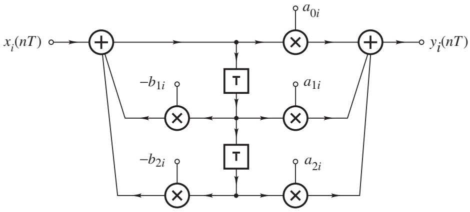


(*b*)

- Cascade realization:



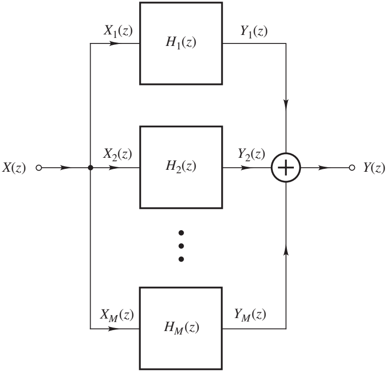
(a)



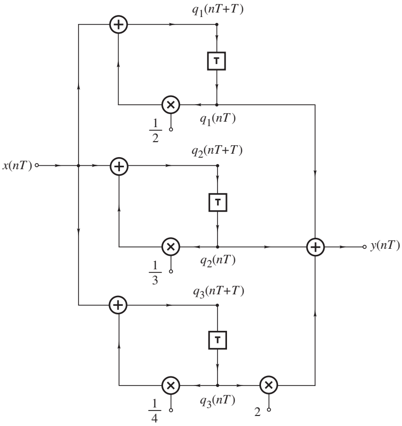
(b)

Realization *Cont'd*

- Parallel realization:



- A third-order digital filter made up of three first-order filters connected in parallel:



This slide concludes Part III of the review.
Thank you for your attention.