Musical Sound Processing

- Almost all musical programs are produced basically in two stages:
- (1) Sound from each individual intrument is recorded in an acoustically inert studio on a single track of a multitrack tape recorder
- (2) Signals from each track are manipulated by the sound engineer by adding special audio effects and then combined in a mix-down system to generate the final stereo recording on a two-track tape recorder

1

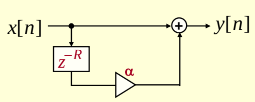
Musical Sound Processing

- The special audio effects are either generated using time-domain operations or frequency-domain operations
- Commonly used time-domain operations are echo generation, reverberation, flanging, chorus generation, and phasing
- Commonly used frequency-domain operations are filters and equalizers

2

Time-Domain Operations

- **Single Echo Filter -**
- Echoes are simply generated by delay units
- For example, the direct sound and a single echo appearing after R sample periods later can be generated by the FIR filter shown below:



3

Time-Domain Operations

- The FIR filter of the previous slide is characterized in the time-domain by the difference equation

$$y[n] = x[n] + \alpha\, x[n-R], \quad |\alpha| < 1$$

- Equivalently, in the frequency-domain it is characterized by the transfer function

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

4

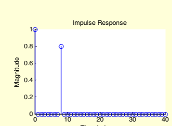
Time-Domain Operations

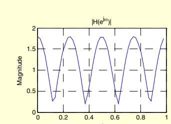
- The delay parameter R in the previous two equations denote the time sound wave takes to travel from the sound source to the listener after bouncing back from the reflecting wall
- The parameter α, with $|\alpha| < 1$, represents signal loss caused by propagation and reflection

5

Time-Domain Operations

- The impulse response and the magnitude response of the single echo filter for $\alpha = 0.8$ and $R = 8$ are shown below:



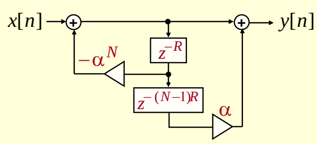


- Program 15_6.m can be used to investigate the effect of a single echo

6

Time-Domain Operations

- **Multiple Echo Filter –**
- The FIR filter shown below generates a fixed number of multiple echoes spaced R sampling periods apart with exponentially delaying amplitudes

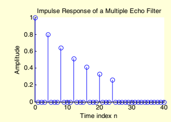


Time-Domain Operations

- The transfer function of the multiple echo filter of the previous slide is given by

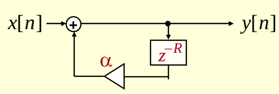
$$H(z) = \frac{1-\alpha^N z^{-NR}}{1-\alpha z^{-R}}, \qquad |\alpha| < 1$$

- The impulse response of a multiple echo filter with $\alpha = 0.8$, $N = 6$, and $R = 4$, is shown below:



Time-Domain Operations

- An infinite number of echoes spaced R sample periods apart with exponentially decaying amplitudes can be generated by the IIR filter shown below:

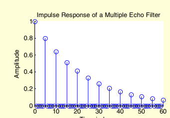


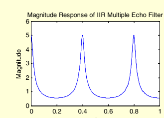
Time-Domain Operations

- The transfer function of the IIR multiple echo filter of the previous slide is given by

$$H(z) = \frac{1}{1-\alpha z^{-R}}, \qquad |\alpha| < 1$$

- Its impulse and magnitude responses for $R = 4$ and $\alpha = 0.8$ are shown below:



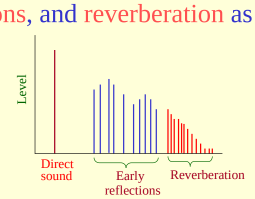


Time-Domain Operations

- The fundamental repetition frequency of the IIR multiple echo filter is given by $F_R = F_T / R$ Hz, where F_T is the sampling frequency
- In practice, the repetition frequency F_R is often locked to the fundamental frequency of an accompanying musical instrument, such as the drum beat
- Program 15_7.m can be used to investigate the effect of multiple echoes

Time-Domain Operations

- **Reverberation –**
- The sound reaching a listener in a closed space, such as a concert hall, consists of several components: direct sound, early reflections, and reverberation as shown below:



Time-Domain Operations

- Early reflections are composed of several closely spaced echoes that are basically delayed and attenuated copies of the direct sound
- Reverberation is composed of densely packed delayed and attenuated echoes
- Sound recorded in an inert studio is different from that recorded in a closed space

Time-Domain Operations

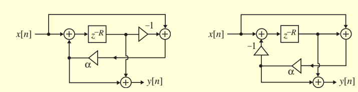
- Digital filtering can be employed to convert the sound recorded in an inert studio into a natural-sounding one by artificially creating the echoes and adding them to the original signal
- The IIR multiple echo filter of Slide 9 by itself does not provide natural-sounding reverberations for two reasons

Time-Domain Operations

- 1) Its magnitude response is not constant at all frequencies, resulting in a "coloration" of many musical sounds that are often unplesant for listening purposes
- 2) The output echo density, given by the number of echoes per second, generated by an unit impulse input, is much lower than that observed in a real room, thus causing a "fluttering" of the composite sound

Time-Domain Operations

- It has been observed that approximately 1000 echoes per second are necessary to create a reverberation that is free of flutter
- A more realistic reverberation, a reverberator with an allpass structure as shown below can be used



Time-Domain Operations

- The transfer function of the allpass reverberator is given by

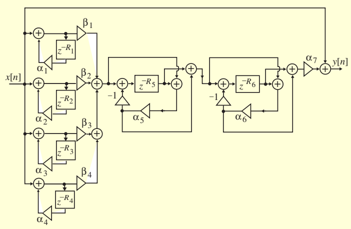
$$H(z) = \frac{\alpha + z^{-R}}{1 + \alpha z^{-R}}, \qquad |\alpha| < 1$$

- In the steady state, the spectral balance of the sound signal remains unchanged due to unity magnitude response of the allpass reverberator
- Program 15_8.m can be used to investigate the effect of the allpass reverberator

Time-Domain Operations

- The IIR multiple echo filter of Slide 9 and the allpass reverberator of Slide 16 are basic units that are suitably connected to develop natural-souding reverberation
- One such interconnection scheme is shown in the next slide
- By choosing different values for the delays in each section (obtained by adjusting R_i) and the multiplier constants α_i, it is possible to arrive at a plesant sounding reverberation

Time-Domain Operations



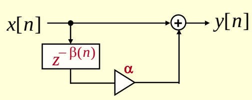
- Program 15_9.m can be used to investigate the effect of the above reverberator

Time-Domain Operations

- **Flanging –**
- This operation was originally 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
- One way of varying Δt is to slow down one of the tape recorders by placing one's thumb on the flange of the feed reel, which led to the name flanging

Time-Domain Operations

- The digital filter structure shown below can be used to create the flanging effect



- Here, the delay unit develops time-varying delay $\beta(n)$

Time-Domain Operations

- By periodically varying the delay between 0 and R with a low frequency ω_o such as

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

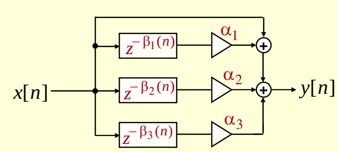
generates a flanging effect
- Since $\beta(n)$ at an instant n has a noninteger value, $y[n]$ should be computed using some type of interpolation method described in Section 13.5
- Program 15_10.m can be used to investigate the flanging effect

Time-Domain Operations

- **Chorus Generator –**
- The chorus effect is achieved when several musicians are playig the same musical piece at the same time but with small changes in the amplitudes and small timing differences between their sounds
- Such an effect can be created synthetically by a chorus generator from the music of a single musician

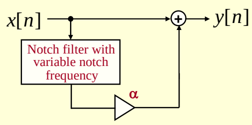
Time-Domain Operations

- The digital filter structure shown below can effectively create a chorus of 4 musicians from the music of a single musician
- To achieve this effect, the delays $\beta_i(n)$ are randomly varied with very slow variations



Time-Domain Operations

- **Phasing Effect Generator –**
- The phasing effect is produced by processing the signal through a narrowband notch filter with variable notch chacteristics and adding a portion of the notch filter output to the original signal as indicated below:



Frequency-Domain Operations

- The frequency responses of individually recorded instruments or musical sounds of performers are often modified by the sound engineer during the mix-down process
- These effect are achieved by passing the original signals through an equalizer

Frequency-Domain Operations

- The equalizer provides "presence" by peaking the mid-frequency components in the range 1.5 to 3 kHz and modifies the bass-treble relationships by providing "boost" or "cut" to components outside this range
- The equalizer is usually formed by cascading first-order and second-order filters with adjustable frequency responses

First-Order Digital Filters and Equalizers

Low-frequency Filters and Equalizers –

- The transfer function of a first-order low-frequency shelving filter for boost is given by

$$G_{LP}^{(B)}(z) = \tfrac{K}{2}[1-\mathcal{A}_B(z)] + \tfrac{1}{2}[1+\mathcal{A}_B(z)]$$

where

$$\mathcal{A}_B(z) = \frac{\alpha_B - z^{-1}}{1-\alpha_B z^{-1}}$$

First-Order Digital Filters and Equalizers

- The tuning parameter α_B is given by

$$\alpha_B = \frac{1-\tan(\omega_c T/2)}{1-\tan(\omega_c T/2)}$$

where ω_c is the cutoff frequency and T is the sampling period

First-Order Digital Filters and Equalizers

- The transfer function of a first-order low-frequency shelving filter for cut is given by

$$G_{LP}^{(C)}(z) = \tfrac{K}{2}[1-\mathcal{A}_C(z)] + \tfrac{1}{2}[1+\mathcal{A}_C(z)]$$

where

$$\mathcal{A}_C(z) = \frac{\alpha_c - z^{-1}}{1-\alpha_C z^{-1}}$$

First-Order Digital Filters and Equalizers

- The tuning parameter α_C is given by

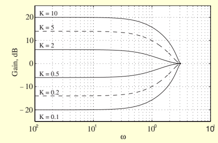
$$\alpha_C = \frac{K - \tan(\omega_c T/2)}{K - \tan(\omega_c T/2)}$$

where ω_c is the cutoff frequency and T is the sampling period

- The gain responses of the first-order lowpass shelving filter are shown in the next slide for various values of the tuning parameters

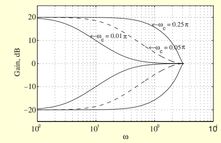
First-Order Digital Filters and Equalizers

- Gain responses of the low-frequency shelving filter for boost and cut are shown below



$\omega_c = 0.25\pi$

$T = 1$



$T = 1$ and $K = 10$ for boost

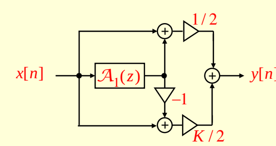
$T = 10$ and $K = 0.1$ for cut

First-Order Digital Filters and Equalizers

- Note: (1) The parameter K controls the amount of boost or cut at low frequencies
- (2) The parameter α_B controls the boost bandwidth, while the parameter α_C controls the cut bandwidth

First-Order Digital Filters and Equalizers

- A realization of the low-frequency shelving filter is shown below where $\mathcal{A}_1(z) = \mathcal{A}_B(z)$ for boost and $\mathcal{A}_1(z) = \mathcal{A}_C(z)$ for cut



First-Order Digital Filters and Equalizers

High-frequency Filters and Equalizers –

- The transfer function of a first-order high-frequency shelving filter for boost is given by

$$G_{HP}^{(B)}(z) = \tfrac{1}{2}[1 - \mathcal{A}_B(z)] + \tfrac{K}{2}[1 + \mathcal{A}_B(z)]$$

where

$$\mathcal{A}_B(z) = \frac{\alpha_B - z^{-1}}{1 - \alpha_B z^{-1}}$$

First-Order Digital Filters and Equalizers

- The tuning parameter α_B is given by

$$\alpha_B = \frac{1 - \tan(\omega_c T/2)}{1 - \tan(\omega_c T/2)}$$

where ω_c is the cutoff frequency and T is the sampling period

First-Order Digital Filters and Equalizers

- The transfer function of a first-order high-frequency shelving filter for cut is given by

$$G_{HP}^{(C)}(z) = \tfrac{1}{2}[1 - \mathcal{A}_C(z)] + \tfrac{K}{2}[1 + \mathcal{A}_C(z)]$$

where

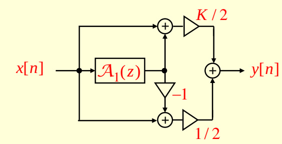
$$\mathcal{A}_C(z) = \frac{\alpha_c - z^{-1}}{1 - \alpha_C z^{-1}}$$

- The multiplier constant α_C here is given by

$$\alpha_C = \frac{1 - K\tan(\omega_c T/2)}{1 + K\tan(\omega_c T/2)}$$

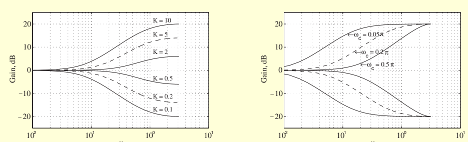
First-Order Digital Filters and Equalizers

- A realization of the high-frequency shelving filter is shown below where $\mathcal{A}_1(z) = \mathcal{A}_B(z)$ for boost and $\mathcal{A}_1(z) = \mathcal{A}_C(z)$ for cut



First-Order Digital Filters and Equalizers

- Gain responses of the high-frequency shelving filter for boost and cut are shown below



$\omega_c = 0.5\pi$

$T = 1$

$T = 1$ and $K = 10$ for boost

$T = 10$ and $K = 0.1$ for cut

First-Order Digital Filters and Equalizers

- Note: (1) The parameter K controls the amount of boost or cut at high frequencies
- (2) The parameter α_B controls the boost bandwidth, while the parameter α_C controls the cut bandwidth

Second-Order Digital Filters and Equalizers

- The transfer function of the second-order peak filter for boost is given by

$$G_{BP}^{(B)}(z) = \tfrac{K}{2}[1 - \mathcal{A}_{2B}(z)] + \tfrac{1}{2}[1 + \mathcal{A}_{2B}(z)]$$

where $\beta = \cos(\omega_o)$ controls the center angular frequency ω_o where the bandpass response peaks and

$$\mathcal{A}_{2B}(z) = \frac{\alpha_B - \beta(1+\alpha_B)z^{-1} + z^{-2}}{1 - \beta(1+\alpha_B)z^{-1} + \alpha_B z^{-2}}$$

Second-Order Digital Filters and Equalizers

- Here, the parameter α_B is related to the 3-dB bandwidth B_w of the bandpass response through

$$\alpha_B = \frac{1 - \tan(B_w T/2)}{1 + \tan(B_w T/2)}$$

- Likewise, the transfer function of the second-order peak filter for cut is given by

$$G_{BP}^{(C)}(z) = \tfrac{K}{2}[1 - \mathcal{A}_{2C}(z)] + \tfrac{1}{2}[1 + \mathcal{A}_{2C}(z)]$$

Second-Order Digital Filters and Equalizers

- In the previous expression, the center angular frequency ω_o, where the bandstop response dips, is related to the parameter β through

$$\beta = \cos(\omega_o)$$

and

$$\mathcal{A}_{2C}(z) = \frac{\alpha_C - \beta(1+\alpha_C)z^{-1} + z^{-2}}{1 - \beta(1+\alpha_C)z^{-1} + \alpha_C z^{-2}}$$

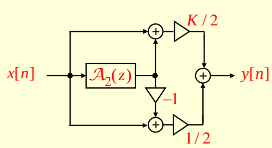
Second-Order Digital Filters and Equalizers

- Here, the parameter α_C is related to the 3-dB bandwidth B_w of the bandpass response through

$$\alpha_C = \frac{K - \tan(B_w T/2)}{K + \tan(B_w T/2)}$$

Second-Order Digital Filters and Equalizers

- Since both $G_{BP}^{(B)}(z)$ and $G_{BP}^{(C)}(z)$ are identical in form, the structure shown below can be employed for boost if $\mathcal{A}_2(z) = \mathcal{A}_{2B}(z)$ and for cut if $\mathcal{A}_2(z) = \mathcal{A}_{2C}(z)$



Second-Order Digital Filters and Equalizers

- Note: The peak or the dip of the gain response occurs at the frequency ω_o which is independently controlled by the parameter β
- Note: The 3-dB bandwidth B_w of the gain response is solely determined by the parameter α_B for boost or by the parameter α_C for cut

Second-Order Digital Filters and Equalizers

- The height of the peak of the magnitude response for boost is given by

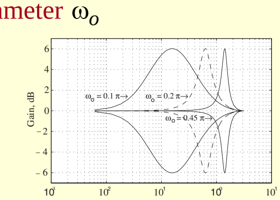
$$K = G_{BP}^{(B)}(e^{j\omega_o})$$

- The height of the dip of the magnitude response for cut is given by

$$K = G_{BP}^{(C)}(e^{j\omega_o})$$

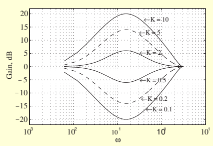
Second-Order Digital Filters and Equalizers

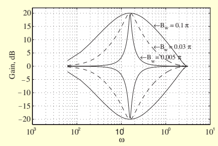
- Figure below show the gain responses of the second-order peak filter obtained by varying the parameter ω_o



Second-Order Digital Filters and Equalizers

- Figure below show the gain responses of the second-order peak filter obtained by varying the parameters K and B_w

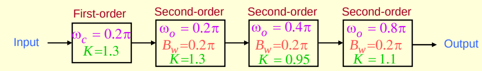




49

Higher-Order Equalizers

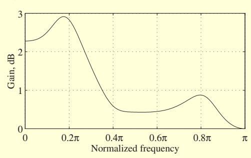
- A graphic equalizer with tunable gain responses can be built using a cascade of first-order and second-order equalizers with external control of the maximum gain values of each section in the cascade
- Figure below shows the block diagram of a typical graphic equalizer



50

Higher-Order Equalizers

- Figure below shows the gain response of the equalizer shown in the previous slide for some typical values of the parameter K (maximum gain values) of the individual sections



51

Musical Sound Processing

- There are basically four methods of musical sound synthesis:

  (1) Wavetable synthesis

  (2) Spectral modeling synthesis,

  (3) Nonlinear synthesis, and

  (4) Physical modeling synthesis

52

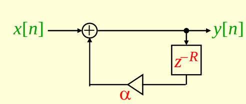
Wavetable Synthesis

- We outline next a simple wavetable synthesis-based method for generating the sounds of plucked string instruments
- The basic idea behid the wavetable synthesis method is to store one period of a desired musical tone and peat it over and over again to generate a periodic signal

53

Wavetable Synthesis

- Such a signal can be generated by running the IIR digital filter structure shown below with specified initial conditions, called wavetable, stored in the delay register z^{-R} and with no input



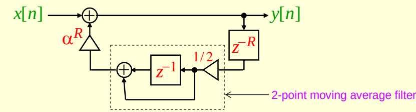
54

Wavetable Synthesis

- Mathematically, the generated periodic note can be expressed as
$$y[n] = y[n-R]$$
where R, called the wavetable length, is the period
- The frequency of the tone is F_T / R, where F_T is the sampling frequency
- Usually, samples of simple waveforms are used as initial conditions

Wavetable Synthesis

- A simple modification of the algorithm has been used to generate plucked-string tones
- The modified algorithm is given by
$$y[n] = \frac{\alpha^R}{2}(y[n-R] + y[n-R-1])$$
- The corresponding structure is shown below



Basic plucked-string filter structure

Wavetable Synthesis

- Note: The structure inside the dashed box is a lowpass filter $G(z)$ consisting of a 2-point moving average filter
- The initial sound of a plucked huitar string contains many high-frequency components
- To simulate this effect, the plucked-string structure is run with zero input and with zero-mean random number stored in the delay block z^{-R}

Wavetable Synthesis

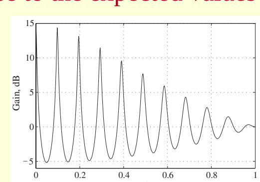
- The high-frequency components of the stored data get repeatedly lowpass filtered by $G(z)$ as they circulate around the feedback loop of the plucked-string filter structure and decay faster than the low-frequency components
- Since the 2-point moving average filter has a group delay of $\frac{1}{2}$ samples, the pitch period of the tone is $R + \frac{1}{2}$

Wavetable Synthesis

- The transfer function of the plucked-string filter structure is given by
$$H(z) = \frac{1}{1 - \frac{\alpha^R}{2}(1 + z^{-1})z^{-R}}$$
- As the loop delay, for $R = 20$, is 20.5 samples, the resonance frequencies are expected to occur at integer multiples of the pitch frequency $F_T / 20.5$, where F_T is the sampling frequency

Wavetable Synthesis

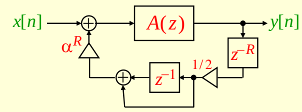
- It can be seen from the gain response given below that for $R = 20$ and $\alpha = 0.99$, the resonance frequencies occur at frequencies very close to the expected values



Wavetable Synthesis

- In addition, the amplitudes of the peaks decrease with increasing frequency, as desired
- Moreover, the widths of the resonance peaks increase with increasing frequency, as expected
- For better control of the pitch frequency, an allpass filter $A(z)$ is inserted in the feedback loop, as indicated in the next slide

Wavetable Synthesis



Modified plucked-string filter structure

- The fractional group delay of the allpass filter can be adjusted to tune the overall loop delay of the modified structure