Music 270a: Introduction to Digital Filters

Tamara Smyth, trsmyth@ucsd.edu
Department of Music,
University of California, San Diego (UCSD)

February 25, 2019

Digital Filters

- Any medium through which a signal passes may be regarded as a filter.
- Typically however, a filter is viewed as something which modifies the signal in some way. Examples include:
  - stereo speakers
  - our vocal tract
  - our musical instruments
- A *digital* filter is a formula for going from one digital signal to another.

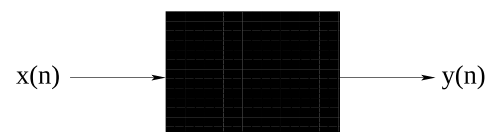


Figure 1: A black box filter.

Linearity and Time Invariance

- A filter is **linear** if it satisfies the following two properties:
  1. **Scaling:** the amplitude of the output is proportional to the amplitude of the input (i.e., scaling can be done either at the input or the output)
  $$\mathcal{L}\{gx(\cdot)\} = g\mathcal{L}\{x(\cdot)\}$$
  2. **Superposition:** the output due to a sum of input signals is equal to the sum of outputs due to each signal alone.
  $$\mathcal{L}\{x_1(\cdot) + x_2(\cdot)\} = \mathcal{L}\{x_1(\cdot)\} + \mathcal{L}\{x_2(\cdot)\}$$
- A filter is **time-invariant** if its behaviour is not dependent on time:
  - if the input signal is delayed by N samples, the output waveform is simply delayed by N samples:
  $$\mathcal{L}\{x(\cdot - N)\} = \mathcal{L}_{n-N}\{x(\cdot)\} = y(n-N)$$

Implications of Linear Time Invariance (LTI)

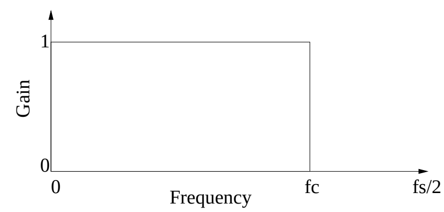
- LTI systems perform only 4 operations on a signal: copying, scaling, delaying, and adding.
- LTI systems, can, be reduced to a calculation involving only the sum of phasors, i.e., LTI filters
  - modify the amplitude and/or phase of the spectral components of an input signal;
  - are guaranteed to produce a sinusoid in response to a sinusoid, without altering the frequency;
  - may only attenuate, reject, amplify (boost) or leave unchanged, any *existing* frequency components of the input signal.

Filter Frequency Response

- The characteristic of a filter is described by its **frequency response** which consists of both a *magnitude* (*amplitude*) and a *phase* response:
  1. **magnitude (amplitude) response**
     - describes the *gain* of a filter at every frequency: a positive gain *boosts* the signal while a negative gain *attenuates* the signal.
     - determined by the **ratio** of the peak output amplitude to the peak input amplitude at a given frequency.
  2. **phase response**:
     - Determined by **subtracting** the input phase from the output phase at a particular frequency.
- A **frequency response** describes how the amplitude and phase of a sound's spectral components are changed by the filter.

The Ideal Low-Pass Filter

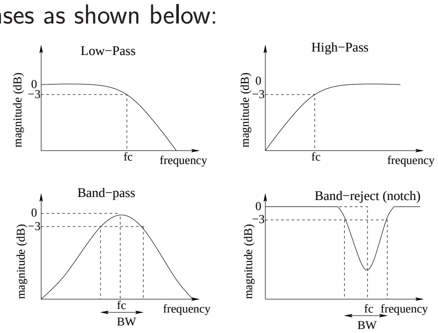
- A low-pass filter is one that allows low frequencies to pass while attenuating anything above a specified cutoff frequency f_c:
  - **pass band**: the region where frequencies are permitted to pass;
  - **stop band**: the region where frequencies are attenuated.
- The amplitude response of an ideal low-pass filter:



- This ideal amplitude response is not realizable in discrete time; how close we come depends on the complexity of the filter.

Magnitude Response of Basic Filters

- There are 4 main types of filters having amplitudes responses as shown below:



- Since no digital filter is *ideal*, they will always a have smooth (*gradual*) transition between the pass and stop bands:
  - the cutoff frequency f_c is typically defined as the frequency at which the **power transmitted drops by one half or 3 dB**;
  - for pass-band filters, a quality factor Q characterizes its bandwithdth BW:

$$Q = \frac{f_c}{\text{BW}}.$$

A Simple Low-Pass Filter

- The simplest low-pass filter (and therefore furthest from ideal) is given by the difference equation

$$y(n) = x(n) + x(n-1).$$

- The corresponding system diagram is given below:

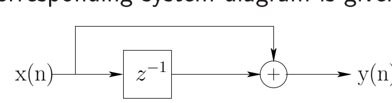


Figure 2: System diagram for the simple low-pass filter $y(n) = x(n) + x(n-1)$.

- This filter is really just a running averager (2-point averager) with a factor of two. That is, it takes the average of two adjacent samples.

Why is it a Low-Pass Filter?

Intuitive Analyis at Low Frequencies

- The running average of an input signal with little or no variation from sample to sample is very close to the input signal.
  - consider an input to this filter at the lowest possible frequency 0 Hz (DC), where the signal has a single value over time:

$$x_1(n) = [A, A, A, ...].$$

  - the output of the lowpass filter is

$$y(n) = x_1(n) + x_1(n-1) = [A, 2A, 2A, ...].$$

  - ignoring the first sample, it boosts DC (gain of 2).

Intuitive Analyis at High Frequencies

- The running average of an input signal with significant variation from sample to sample will be very different than the input signal.
  - consider a second input $x_2(n)$ at the highest possible frequency $f_s/2$ (the Nyquist limit), which swings more significantly from sample to sample:

$$x_2(n) = [A, -A, A, ...].$$

  - the output of the lowpass filter is

$$y(n) = x_2(n) + x_2(n-1) = [A, 0, 0, ...]$$

  - ignoring the first sample, it attenuates the Nyquist limit.

What about all the frequencies in between?

- This filter seems to boost low frequencies while rejecting components at higher frequencies.
- We may find the frequency response of the filter by checking the behaviour of the filter at every possible frequency between 0 and $f_s/2$ Hz (**sinewave analysis**).
- Alternatively, we can use an input signal that contains all of those frequencies, and then we only have to do the "checking" operation once.
- If we use an input signal with the broadest possible spectrum, i.e. an impulse, we will obtain an output from the filter, called an *impulse response*.
- The impulse response is a time domain description of the filter's response to all frequencies from DC to the Nyquist limit.
- The spectrum of the impulse response gives us the the frequency response of the filter.

Response at the Cutoff Frequency

- Look a little closer at the filter's response to $f_s/4$.

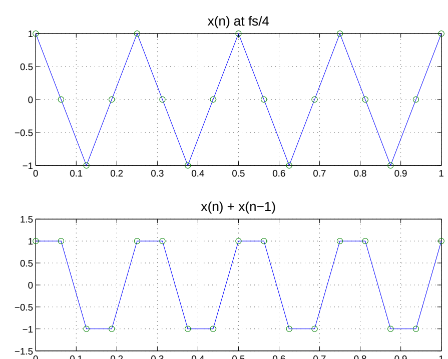


Figure 3: Filter behaviour at $f_s/4$.

Interpreting the Phase

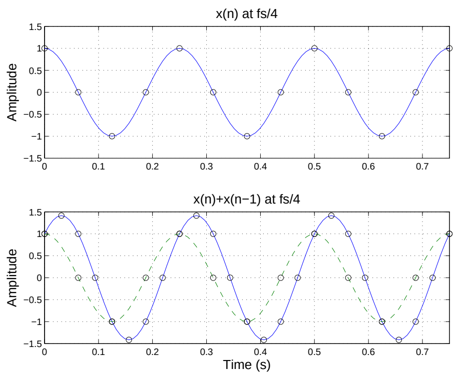


Figure 4: Filter behaviour at $f_s/4$.

- In addition to boosting the signal, this filter is *delaying* it by half a sample.
- In fact, this filter delays *all* frequencies by half a sample.

Linear Phase Filters

- Filters that delay all frequencies by the same amount are called *linear phase* filters.
- Linear phase filters have a symmetric impulse response;
  - the sample about which there is symmetry yields the delay (in samples) of the filter.

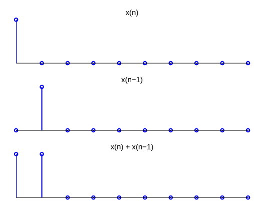


Figure 5: Filter impulse response.

- For this filter, the impulse response is symmetric about sample 0.5, which corresponds to a waveform delay of one-half sample at all frequencies.

Phase Delay

- The value about which the impulse response is symmetric is the *phase delay* of the filter.

> A "simple waveform delay" means the waveform will not change with a change in frequency.

- Linear phase is desirable because it delays all frequencies by the same number of samples and that means **no phase distortion**.

Low-pass Filter Implementation

```
N = 1024;              % signal length
x = [1 zeros(1, N-1)]; % impulse
y = [1 zeros(1, N-1)]; % output buffer

for n=2:N
  y(n) = x(n)+x(n-1);  % impulse response
end

Y = fft(y);           % frequency response
Y = abs(Y(1:N/2));    % amplitude response (positive frequencies)
fn = [0:N/2-1]/N;     % frequency axis

subplot(211); plot(fn, Y); grid;
title('Amplitude Response y(n) = x(n) + x(n-1)');
xlabel('Frequency (normalized)');
ylabel('Magnitude (linear)');
```

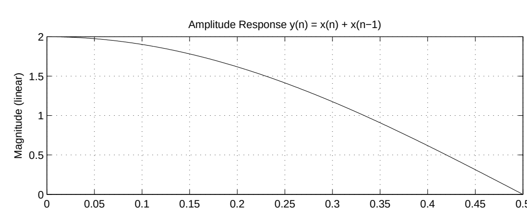


Figure 6: Magnitude response for the filter y(n) = x(n) + x(n-1).

Filter Frequency Response

- Let's test the simple lowpass filter's response at frequency f by setting it's input to the complex sinusoid
$$x(n) = Ae^{j(2\pi fnT+\phi)}$$
- Because of:
  1. **Time-invariance:** the frequency response will not depend on ϕ, so we may set it to 0;
  2. **Linearity:** we may set A to 1.
- The output of the filter in response to $x(n) = e^{j(2\pi fnT)}$ is given by
$$\begin{aligned} y(n) &= x(n) + x(n-1) \\ &= e^{j\omega nT} + e^{j\omega(n-1)T} \\ &= e^{j\omega nT} + e^{j\omega nT}e^{-j\omega T} \\ &= (1+e^{-j\omega T})e^{j\omega nT} \\ &= (1+e^{-j\omega T})x(n) \\ &\triangleq H(e^{j\omega T})x(n) \end{aligned}$$
where $H(e^{j\omega T}) = (1+e^{-j\omega T})$, a complex multiply, is the **frequency response** of the filter.

Frequency Response

- The filter's frequency response is acting as a complex multiply, which means a **gain scaling** and **phase shift** on the input signal.
- The complex filter gain in polar form is given by
$$H(e^{j\omega T}) = G(\omega)e^{j\Theta(\omega)}$$
where the amplitude and phase response are given by
$$G(\omega) \triangleq |H(e^{j\omega T})| \quad \text{and} \quad \Theta(\omega) \triangleq \angle H(e^{j\omega T}).$$
- To solve, we may "balance the exponents" to obtain
$$\begin{aligned} H(e^{j\omega T}) &= (1+e^{-j\omega T}) \\ &= (e^{j\omega T/2} + e^{-j\omega T/2})e^{-j\omega T/2} \\ &= 2\cos(\omega T/2)e^{-j\omega T/2}, \end{aligned}$$
where the amplitude response is then given by
$$\begin{aligned} G(\omega) &= \left|2\cos(\omega T/2)e^{-j\omega T/2}\right| \\ &= 2\cos(\pi fT), \qquad |f| \leq f_s/2, \end{aligned}$$
and the phase response is given by
$$\Theta(\omega) = -\frac{\omega T}{2} = -\pi\frac{f}{f_s}, \qquad |f| \leq f_s/2.$$

Interpreting the Phase Response

- What does it mean to have a phase response of
$$\Theta(\omega) = -\frac{\omega T}{2} = -\pi\frac{f}{f_s}, \qquad |f| \leq f_s/2?$$
- We see, again but perhaps more clearly, that the phase response is $\mathbf{linear}$ in frequency.
- In general, to obtain the phase delay (*time delay*) in seconds, divide by $-\omega$:
$$P(\omega) \triangleq -\frac{\Theta(\omega)}{\omega} \quad \text{seconds.}$$

Other simple non-recursive filters

- Test the following filter at DC and $f_s/2$:
$$y(n) = x(n) - x(n-1).$$
What type of filter is this?
- Verify that the following filter passes both DC and the Nyquist limit (1/2 the sampling rate):
$$y(n) = x(n) + x(n-2).$$
  – what about at a frequency in between, $f = fs/4$?
$$x(n) = [A, 0, -A, 0, A, 0, -A, \ldots],$$
  – at this frequency the filter produces no output; assume therefore a band-reject (notch) filter, with a notch at $f_s/4$.
- Verify that the following filter rejects both DC and the Nyquist limit, yes boosts a frequency at $f_s/4$.
$$y(n) = x(n) - x(n-2).$$
  – what kind of filter is it?

Plots of simple filters

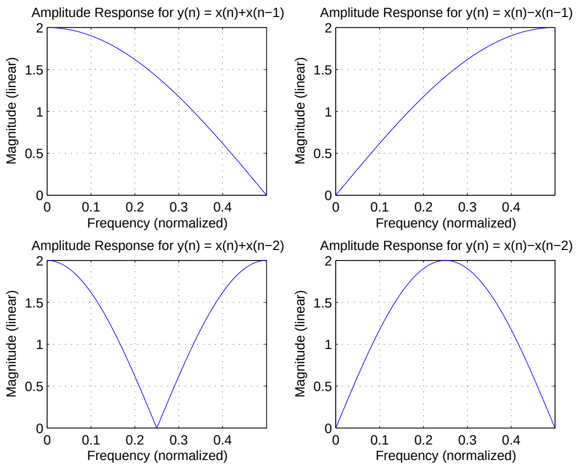


Figure 7: Amplitude Responses for simple filters

.

Generalized FIR

- Several different nonrecursive filters can be made by changing the delay (and thus order) and the coefficients of the filter terms.
- The general equation for an FIR (Finite Impulse Response) filter is given by

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

where M is the order of the filter.

- A filter can be defined simply by a set of coefficients. For example if

$$b_k = [1, 3, 3, 1]$$

the filter is third order, (i.e. $M = 3$), and can be expanded into the difference equation

$$y(n) = x(n) + 3x(n-1) + 3x(n-2) + x(n-3);$$

- When the input to the FIR filter is a unit impulse sequence, $\delta(n) = 1$ if $n = 1$ and $\delta(n) = 0$ otherwise, the output is the unit impulse response.

Matlab's `filter` function

- The Matlab implementation for the filter is most easily accomplished using the `filter` function

```
y = filter(B, A, x).
```

- The `filter` function takes three (3) arguments:
  1. feedforward coefficients B,
  2. feedback coefficients A,
  3. and the input signal x.
- If the filter doesn't have feedback coefficients, as is the case with an FIR filter, A = 1.
- Our simple low-pass filter, $y(n) = x(n) + x(n-1)$, is a first order filter with 2 feedforward coefficients (`B = [1, 1]`).
- In matlab:

```
B = [1 1];
A = 1;
y = filter(B, A, x);
```


Increasing the Filter Order

- Let's return now to the simple low-pass filter $y(n) = x(n) + x(n-1)$ which is just a two point running average (with a factor of 2).
- If we increase the number of samples averaged, i.e. increase the filter order,

$$y(n) = x(n) + x(n-1) + x(n-2),$$

the waveform will be smoothed (with a more gentle slope to zero), which corresponds to a lowered cutoff frequency

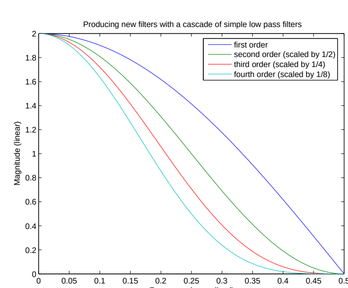


Figure 8: A Cascade of Simple Lowpass filters.

Coefficients as Impulse Response

- If we look at the response of the filter to an impulse

```
>> x(1:10)

ans =

     1 0 0 0 0 0 0 0 0 0

>> y(1:10)

ans =

     1 1 0 0 0 0 0 0 0 0
```

  we see the impulse response y is equivalent to coefficients of our FIR filter.
- This can be expressed using the general FIR equation, with an input of $x(n) = \delta(n)$:

$$h(n) = \sum_{k=0}^{M} b_k \delta(n-k) = b_n.$$

Cascade Connection

- An order can be increased by using a filter cascade.
- Each time the previous output y is input to the filter, a new impulse response representing is created.

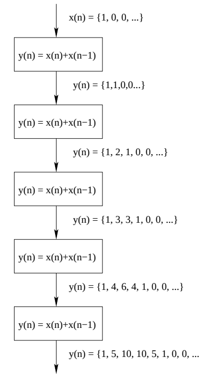


Figure 9: A cascade of simple low-pass filters.

- The new output always has a finite number of non-zero components and a *finite impulse response*.
- For each first-order filter in the cascade, the impulse response increases by one.

Matlab LPF Cascade Implementation

```
x = [1 zeros(1, N-1)];
```

Cascade...

```
B = [1 1];
A = [1];
y1 = filter(B, A, x);
y2 = filter(B, A, y1);
y3 = filter(B, A, y2);
y4 = filter(B, A, y3);
```

Or Simply...

```
B = [1 5 10 10 5 1];
A = [1];
y = filter(B, A, x);
```


Recursive Filters

- Using FIR filters often required significant computation and coefficients to reproduce a desired frequency response.
- It is often possible to reduce the number of feedforward coefficients needed to obtain a frequency response by introducing feedback coefficients.
- A simple example of a first order recursive low-pass filter is given by

$$y(n) = b_0 x(n) + a_1 y(n-1)$$

- The general difference equation for LTI filter therefore, is given by

$$\begin{aligned} y(n) = b_0 x(n) \; &+ \; b_1 x(n-1) + \cdots + b_M x(n-m) \\ &- \; a_1 y(n-1) - \cdots - a_N y(n-N) \end{aligned}$$

FIR vs IIR Simple Low-Pass Filters

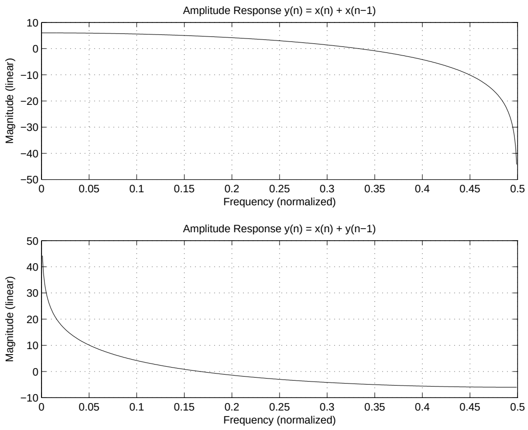


Figure 10: Simple Non-recursive low-pass $y(n) = x(n)+x(n-1)$ (top) and Recursive low-pass $y(n) = x(n) + y(n-1)$ (bottom).

Z Transforms

- The *bilateral* z transform of the discrete-time signal $x(n)$ is given by

$$X(z) \triangleq \sum_{n=-\infty}^{\infty} x(n)z^{-n}.$$

- Since most signals we'll be using are *causal*, we will typically see the *unilateral* z transform given by

$$X(z) \triangleq \sum_{n=0}^{\infty} x(n)z^{-n}.$$

- The z transform is a generalization of the DTFT (the DFT in the limit as the number of its samples approaches infinity).
  - The carrier term is a *generalized* sampled complex sinusoid $z^n = e^{(\sigma+j\omega)t}$ with an exponential envelope rather than a constant modulus.
  - The DTFT equals the z transform evaluated on the *unit circle* in the z plane.
- The z transform of a signal x is given by

$$\mathcal{Z}\{x(\cdot)\} = X(z) \quad \text{also notated} \quad \mathcal{Z}\{x(n)\} = X(z).$$

Z Transform Properties

- Two (2) important properties of z transforms:

1. The z transform $\mathcal{Z}\{\cdot\}$ is a *linear operator* which means, by definition

$$\begin{aligned}\mathcal{Z}\{\alpha x_1(n) + \beta x_2(n)\} &= \alpha\mathcal{Z}\{x_1(n)\} + \beta\mathcal{Z}\{x_2(n)\} \\ &\triangleq \alpha X_1(z) + \beta X_2(z).\end{aligned}$$

2. From the *shift theorem* for z transforms, the z transform of a signal delayed by M samples is given by

$$\mathcal{Z}\{x(n-M)\} = z^{-M}X(z).$$

- Shift Theorem:

A **delay of M samples** in the time domain corresponds to a **multiplication by z^{-M}** in the frequency domain.

Z Transform of the Difference Equation

- Given the general difference equation for LTI filters,

$$\begin{aligned}y(n) = b_0x(n) \;&+\; b_1x(n-1) + \cdots + b_Mx(n-m) \\ &-\; a_1y(n-1) - \cdots - a_Ny(n-N),\end{aligned}$$

taking the z transform of both sides yields

$$\begin{aligned}\mathcal{Z}\{y(\cdot)\} &= \mathcal{Z}\{b_0x(n) + b_1x(n-1) + \cdots + b_Mx(n-m) \\ &\quad -a_1y(n-1) - \cdots - a_Ny(n-N)\}.\end{aligned}$$

- Apply the **linearity property** to obtain

$$\begin{aligned}\mathcal{Z}\{y(\cdot)\} &= b_0\mathcal{Z}\{x(n)\} + b_1\mathcal{Z}\{x(n-1)\} \\ &\quad + \cdots + b_M\mathcal{Z}\{x(n-M)\} \\ &\quad -a_1\mathcal{Z}\{y(n-1)\} - \cdots - a_N\mathcal{Z}\{x(n-N)\},\end{aligned}$$

and the **shift theorem** to obtain

$$\begin{aligned}Y(z) &= b_0X(z) + b_1z^{-1}X(z) + \cdots + b_Mz^{-M}X(z) \\ &\quad -a_1z^{-1}Y(z) - \cdots - a_Nz^{-N}Y(z).\end{aligned}$$

Transfer Function

- The transfer function is the ratio of the output to the input.
- Group the $Y(z)$ terms together on the left hand side and factor out common terms $X(z)$ and $Y(z)$:
$$Y(z)[1 + a_1 z^{-1} + \cdots + a_N z^{-N}] = X(z)[b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}]$$
- In defining the following polynomials:
$$A(z) \triangleq 1 + a_1 z^{-1} + \cdots + z_N z^{-N}$$
$$B(z) \triangleq b_0 + b_1 z^{-1} + \cdots + b_M z^{-M},$$
the z transform of the difference equation becomes
$$A(z)Y(z) = B(z)X(z).$$
- In solving for $Y(z)/X(z)$ we obtain the transfer function
$$H(z) = \frac{Y(z)}{X(z)} = \frac{B(z)}{A(z)},$$
where $X(z)$ and $Y(z)$ are the z transforms of the input and output signal, respectively.
- $H(z)$ is the z transform of the impulse response describing the filter.

Transfer Function Summary

- A transfer function provides an algebraic representation of a LTI filter in the frequency domain.
- Recall, that we can determine the frequency response by observing the effects of the filter at different frequencies, a technique called *sine-wave analysis*.
- The gain or *amplitude response* of the filter at a given frequency is determined by the **ratio** of the the peak output amplitude to the peak input amplitude at this frequency.
- The *phase response* of the of the filter at a given frequency is determined by the **difference** between the output and input phases at a given frequency.
- The transfer function of an LTI filter is given by
$$H(z) = \frac{Y(z)}{X(z)}$$
where $Y(z)$ is the z transform of the output signal $y(n)$ and $X(z)$ is the z transform of the input signal $x(n)$.
- The *transfer function* is equal to the z transform of the impulse response.

Pole Zero Analysis

- The transfer function for the recursive LTI digital filter
$$H(z) = g\frac{1 + \beta_1 z^{-1} + \cdots + \beta_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}}$$
- We can factor the numerator and denominator to obtain
$$H(z) = g\frac{(1 - q_1 z^{-1})(1 - q_2 z^{-1}) \cdots (1 - q_M z^{-1})}{(1 - p_1 z^{-1})(1 - p_2 z^{-1}) \cdots (1 - p_N z^{-1})}$$

Every polynomial can be characterized by its roots plus a scale factor.

- We may therefore characterize a transfer function, by its
  1. Numerator roots, called the **zeros** of the filter
  2. Denominator roots, called **poles** of the filter
  3. Constant gain factor

Poles and Zeros cont.

- The factored transfer function is given by
$$H(z) = g\frac{(1 - q_1 z^{-1})(1 - q_2 z^{-1}) \cdots (1 - q_M z^{-1})}{(1 - p_1 z^{-1})(1 - p_2 z^{-1}) \cdots (1 - p_N z^{-1})}$$

Zeros

- The *roots* of the numerator polynomial are given by
$$\{q_1, q_2, \ldots, q_M\}.$$

When z takes on any of these values, the transfer function evaluates to zero and thus they are called the *zeros* of the filter.

Poles

- The *roots* of the denominator polynomial are given by
$$\{p_1, p_2, \ldots, p_M\}$$

When z approaches any of these values, the transfer function becomes larger and larger, approaching infinity. Thus these are called the *poles* of the filter.

Imagining Poles and Zeros

- Recall, the magnitude $H(z)$ is a function of z:
  - Since z is complex, it lies in the z plane.
  - Since the magnitude $H(z)$ is real, it can be represented a distance above the z plane. The plot appears as an infinitely thin surface spanning in all directions over the z plane.
  - The zeros are the points where the surface dips down to touch the z plane.
  - The poles look just like "poles" that rise forever from the z plane, getting thinner the higher they go.
- Notice $M+1$ feedforward coefficients gives rise to M zeros while N feedback coefficients gives rise to N poles.

> The filter order is given by the maximum of the numerator and denominator polynomial orders.

Relating Amplitude Response to Poles and Zeros

- The frequency response of the transfer function (factored form) is given by

$$H(e^{j\omega t}) = g\frac{(1-q_1e^{-j\omega T})(1-q_2e^{-j\omega T})\cdots(1-q_Me^{-j\omega T})}{(1-p_1e^{-j\omega T})(1-p_2e^{-j\omega T})\cdots(1-p_Ne^{-j\omega T})}$$

- Consider the amplitude response $G(\omega) \triangleq |H(e^{j\omega T})|$

$$\begin{aligned} G(\omega) &= |g|\frac{|1-q_1e^{-j\omega T}|\cdot|1-q_2e^{-j\omega T}|\cdots|1-q_Me^{-j\omega T}|}{|1-p_1e^{-j\omega T}|\cdot|1-p_2e^{-j\omega T}|\cdots|1-p_Ne^{-j\omega T}|} \\ &= |g|\frac{|e^{-jM\omega T}|\cdot|e^{j\omega T}-q_1|\cdots|e^{j\omega T}-q_M|}{|e^{-jN\omega T}|\cdot|e^{j\omega T}-p_1|\cdots|e^{j\omega T}-p_N|} \\ &= |g|\frac{|e^{j\omega T}-q_1|\cdot|e^{j\omega T}-q_2|\cdots|e^{j\omega T}-q_M|}{|e^{j\omega T}-p_1|\cdot|e^{j\omega T}-p_2|\cdots|e^{j\omega T}-p_N|}. \end{aligned}$$

> The amplitude response is the product of the difference between two complex numbers.

Difference of Two Complex Numbers

- In the complex plane, the number $z = x + jy$ is plotted at the coordinates (x, y). The difference of two vectors $u = x_1 + jy_1$ and $v = x_2 + jy_2$ is given by

$$u - v = (x_1 - x_2) + j(y_1 - y_2).$$

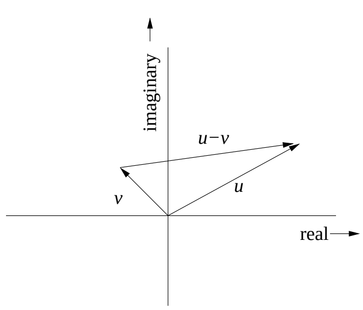


Figure 11: Treatment of complex numbers as vectors in a plane.

Poles and Zeros and the Unit Circle

- Recall the amplitude response

$$G(\omega) = |g|\frac{|e^{j\omega T}-q_1|\cdot|e^{j\omega T}-q_2|\cdots|e^{j\omega T}-q_M|}{|e^{j\omega T}-p_1|\cdot|e^{j\omega T}-p_2|\cdots|e^{j\omega T}-p_N|}.$$

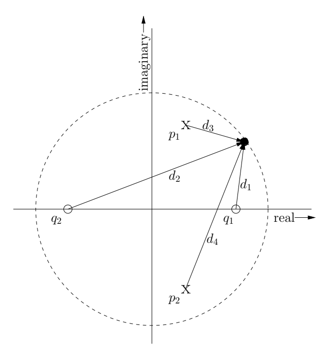


Figure 12: Measurement of amplitude response from a pole-zero diagram (a bi-quad section).

- Thus the term $e^{j\omega T} - q_i$ may be drawn as an arrow from the ith zero to the point $e^{j\omega T}$ on the unit circle, and $e^{j\omega T} - p_i$ is an arrow from the ith pole.
- The amplitude response at frequency ω is given by

$$G(\omega) = \frac{d_1 d_2}{d_3 d_4}$$

Stability

- A filter is said to be *stable* if its impulse response $h(n)$ decays to 0 as n goes to infinity.
- Recall, the transfer function is the z transform of the impulse response.
$$H(z) \triangleq \mathcal{Z}\{h(n)\}$$
- Consider a *causal* impulse response of the form
$$h(n) = R^n e^{j\omega nT}, n = 0, 1, 2, \ldots$$
which is a damped complex sinusoid when $0 < R < 1$ and is exponentially increasing when $R > 1$.
- The signal $h(n)$ has the z transform
$$\begin{aligned} H(z) &= \sum_{n=0}^{\infty} R^n e^{j\omega nT} z^{-n} \\ &= \sum_{n=0}^{\infty} \left(Re^{j\omega T} z^{-1}\right)^n \\ &= \frac{1}{1 - Re^{j\omega T} z^{-1}} \end{aligned}$$
where the last step is a *closed form* representation of a geometric series and holds for $|Re^{j\omega T} z^{-1}| < 1$, which is true whenever $R < |z|$.

Stability cont.

- The transfer function
$$H(z) = \frac{1}{1 - Re^{j\omega T} z^{-1}}$$
has a single pole at $Re^{j\omega T}$.
- If $R > 1$ the pole of $H(z)$ moves outside the unit circle, and the impulse response $h(n)$ has an exponentially increasing amplitude ($|h(n)| = R^n$). That is, it is *unstable*.
- A pole lying on the unit circle is considered *marginally stable*, that is, it neither decays nor grows in amplitude. In physically modelling synthesis, marginally stable poles often occur in *lossless* systems such as *ideal vibrating strings*.
- Therefore, to verify stability, we may find the roots of the denominator polynomial and ensure their roots are less than 1.

For the stability of any finite order LTI filter, all its poles must lie strictly inside the unit circle.

The Biquad Section

- The term "biquad" is short for "bi-quadratic", and is a common name for a two-pole, two-zero digital filter. The transfer function of the biquad can be defined as
$$H(z) = g\frac{1 + \beta_1 z^{-1} + \beta_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$
- When the coefficients a_1 and a_2 are real (as we typically assume), the poles must be either
  1. real (when $(a_1/2)^2 \geq a_2$)
  2. form a complex conjugate pair (when $(a_1/2)^2 < a_2$).
- When the poles form a complex pair, we may express them in polar form as
$$\begin{aligned} p_1 &= Re^{j\theta_c} \\ p_2 &= Re^{-j\theta_c} \end{aligned}$$
R is the pole radius, or distance from the origin in the z-plane.

Bi-quadratic Resonant Filter

- For a biquad resonant (bandpass) filter,
$$H(z) = g\frac{1 + \beta_1 z^{-1} + \beta_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}},$$
the filter coefficients are given by
$$\beta_1 = 0, \quad \beta_2 = -R$$
and
$$a_1 = -2R\cos(2\pi f_c T), \quad a_2 = R^2,$$
where f_c is the resonant (or center) frequency, and R, is set according to the desired bandwidth of the resonator
$$R = e^{-\pi B_w T},$$
where B_w is the bandwidth at -3dB in Hz given by
$$B_w = \frac{f_c}{Q},$$
and T is the sampling period.
- This function is often called *bi-quadratic* or simply a *biquad* because both the numerator and denominator of it's transfer function are quadratic polynomials.

Matlab Implementation of the Biquad

- The two control parameters for this filter are
  1. the center frequency f_c
  2. the quality factor Q (or bandwidth).

```
T = 1/44100;
Q = 20;
fc = 400;
Bw = fc/Q;

R = exp(-pi*Bw*T);

B = [1 0 -R];
A = [1 -2*R*cos(2*pi*fc*T) R*R];

[h, w] = freqz(B,A);
```

- Use `freqz`, `tf2zp` and `zplane` to see how the spectrum corresponds to the plot of the poles and zeros in the z-plane.
- Parameters can be changed in real-time, making this an efficient and ideal implementation for performance situations.

FIR/Convolution

- Since the feedforward coefficient's of the FIR filter are the same as the non-zero elements of the impulse response, a general expression for the FIR filter's output can also be given by

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k),$$

where $h(\cdot)$ is the impulse responses and replaces the coefficients b_k.

- When the relation between the input and the output of the FIR filter is expressed in terms of the input and impulse response, we say the the output is obtained by *convolving* the sequences $x(n)$ and $h(n)$.
- Note that this is *circular* or *cyclic* convolution. To simulate acyclic convolution (as we do when simulating sampled continuous-time systems), we need to zero-pad sufficiently $(N+M-1)$ so that non-zero samples do not "wrap around" as a resulting of shifting x.

Visulalizing the convolution sum

- The convolution sum is given by

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k).$$

n	0	1	2	3	4	5
x(n)	1	1	0	0	0	0
h(n)	1	3	3	1	0	0
h(0)x(n-0)	1	1	0	0	0	0
h(1)x(n-1)		3	3	0	0	0
h(2)x(n-2)			3	3	0	0
h(3)x(n-3)				1	1	0
y(n)	1	4	6	4	1	0

The Delay Line

- The delay line models *acoustic propagation delay*.

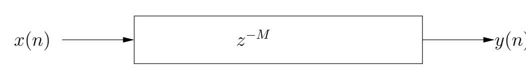


Figure 13: The M-sample delay line.

- The delay line introduces a time delay corresponding to M samples between its input and output

$$y(n) = x(n-M), \quad n = 0, 1, 2, \ldots$$

- It is linear phase:
  1. a phase delay of M samples
  2. delays all frequencies by this amount
  3. IR is symmetric about M^{th} sample

The Feedforward Comb Filter

- Increasing the delay on the second term of the "simple lowpass filter" can be seen as replacing that term with a delay line, yielding

$$y(n) = x(n) + gx(n - M),$$

where g is the coefficient multiplying the delay.

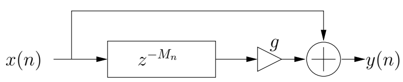


Figure 14: A feedforward comb filter.

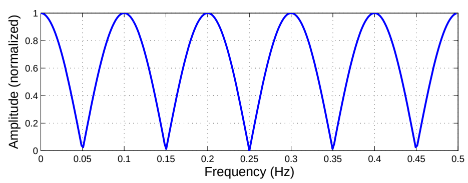


Figure 15: The comb filter magnitude response.

- The name of the filter comes from the fact that the magnitude response resembles the teeth of a comb, and because there are not feedback terms.

The Delay Line Coefficent g

- The feedforward coefficient g controls the proportion of the delay signal in the output.

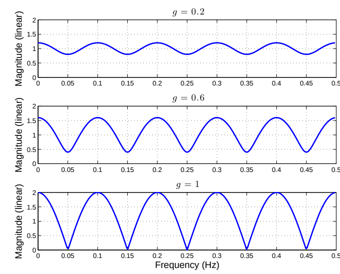


Figure 16: Delayline coefficient controlls notch attenuation and peak gain.

- Often referred to as the DEPTH parameter, setting
  - the amount of **gain** at the maxima,
  - the amount of **attenuation** at the minima,

  that is, the *depth* from the peaks to the notches.
- Has a range from 0 to 1 (1 corresponds to maximum depth).

Why Spectral Notches?

- Notches occur in the spectrum as a result of **destructive interference**.
- Recall that delaying a sine tone 180 degrees (1/2 a cycle) and summing with the original will cause the signal to disappear at the output.

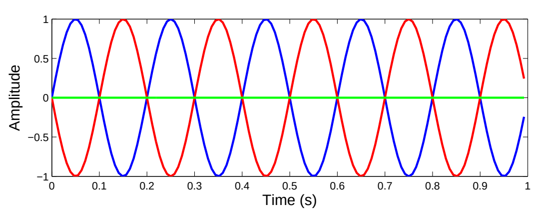


Figure 17: Complete destructive interference.

- If a sinusoid has a frequency of f_0, then a delay of 1/2 cycle corresponds to a delay of

$$\tau = \frac{1}{2f_0} \text{ seconds}$$

or

$$M = \frac{f_s}{2f_0} \text{ samples.}$$

Delay Parameter M

- A delay of $M = f_s/(2f_0)$ samples in the comb filter will yield a *notch* (complete cancelation) at

$$f_0 = \frac{f_s}{2M} \text{ Hz},$$

as well as notches at **odd harmonics** of f_0.

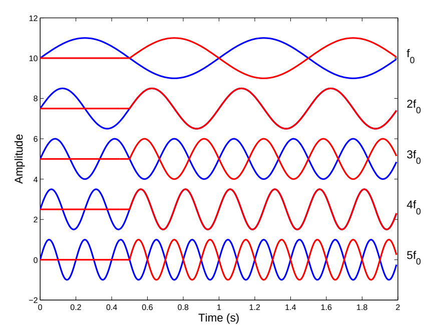


Figure 18: Destructive interference occurs at odd harmonics of the fundamental frequency.

The Simple Feedback Comb Filter

- What happens when we multiply the output of a delay line by a gain factor g then feed it back to the input?

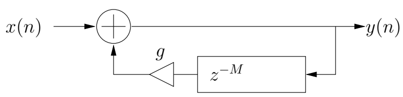


Figure 19: The signal flow diagram of a comb filter.

- The difference equation for this filter is

$$y(n) = x(n) + gy(n - M),$$

- If the input to the filter is an impulse

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

the output (impulse response) will be ...

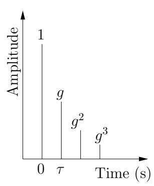


Figure 20: Impulse response for filter $y(n) = x(n) + gy(n - M)$.

Effect of Feedback Delay

- Since the pulses are equally spaced in time at an interval equal to the loop time $\tau = M/f_s$ seconds, it is periodic and will sound at the frequency $f_0 = 1/\tau$.

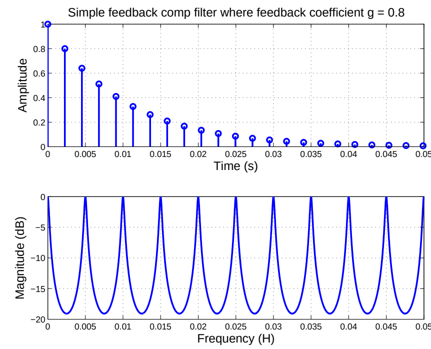


Figure 21: Impulse and magnitude response of a comb filter with feedback g = 0.8.

- Like in the feedforward case, the spacing between the maxima of the "teeth" is equal to the natural frequency.

Effect of the Feedback coefficient g

- The depth of the minima and height of the maxima are set by the choice of g, where values closer to 1 yield more extreme maxima and minima.

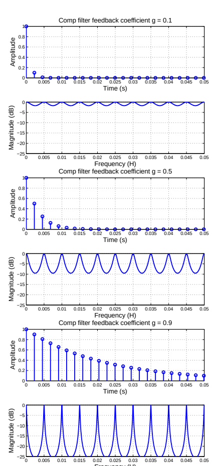


Figure 22: Impulse and Magnitude Response with increasing feedback coefficient.

Comb Filter Decay Rate

- The response decays exponentially as determined by the loop time and gain factor g.
- Values of g nearest 1 yield the longest decay times.
- To obtain a desired decay time, g may be approximated by

$$g = 0.001^{\tau/T}$$

where

$$\begin{aligned} \tau &= \text{the loop time} \\ T60 &= \text{the time to decay by 60dB} \end{aligned}$$

and 0.001 is the level of the signal at 60dB down.

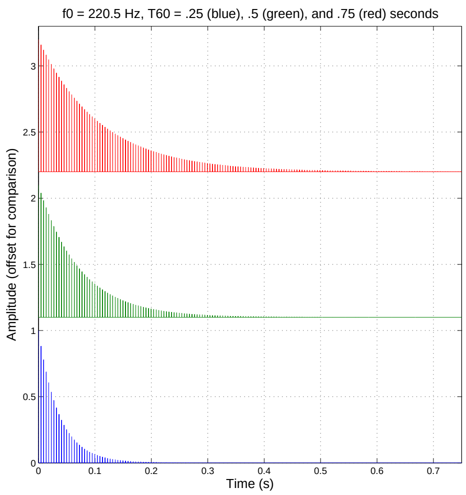


Figure 23: Comb filter impulse responses with a changing the decay rate.

General Comb Filter

- Consider now, adding to the filter a delay element which delays the input by M_1 samples, with some gain g_1.
- The general comb filter is given by the difference equation

$$y(n) = x(n) + g_1 x(n - M_1) - g_2 y(n - M_2)$$

where g_1 and g_2 are the feedforward and feedback coefficients, respectively.

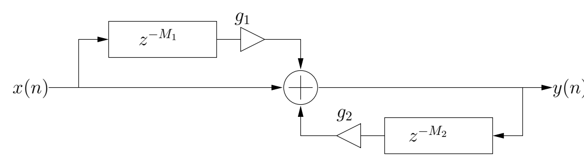


Figure 24: Signal flow diagram for digital comb filters.