

The DSP Cookbook

Pelle Juul Christensen

October 2017

Contents

1	Theory	2
1.1	Sine Wave	2
1.2	Phasors	3
1.3	Digital Signals	3
1.4	DSP Systems	4
1.5	Causal Systems	4
1.6	Difference Equations	4
1.7	Linear Time-Invariant System	4
1.8	Impulse Response	5
1.9	Convolution	6
1.10	Digital Frequency	6
1.11	Frequency Response	6
1.12	Z-transform	8
2	Recipes	9
2.1	Echo and Delay	9
2.2	Chorus	10
2.3	Flanging	10
2.4	Ring Modulation and Amplitude Modulation	10
2.5	Biquad Filters	11
2.5.1	Low-pass Filter	11
2.5.2	High-pass Filter	12
3	References	12

1 Theory

1.1 Sine Wave

The general equation of is

$$A \sin(2\pi t f + \theta), \quad (1)$$

where

A is the amplitude of the wave.

\sin is the sin function.

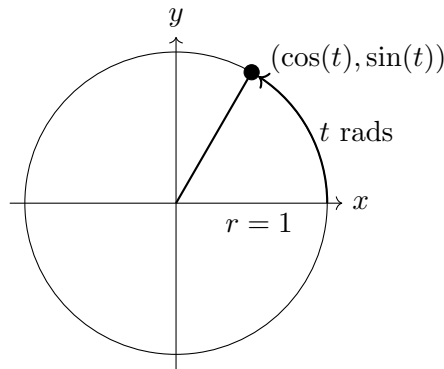
2π normalizes the period of the sin function to 1.

t is time.

f is the frequency of the wave.

θ is the phase offset of the wave.

The sin and cos functions are defined using the unit circle. Given a variable t we go t radians along the unit circle ¹. The point at t radians will have coordinates $(\cos(t), \sin(t))$ — $\cos(t)$ is the projection of the point onto the x-axis, $\sin(t)$ is the projection onto the y-axis.



Sinusoids have a period of 2π (a full arc), thus if we multiply t by 2π to get $\sin(2\pi t)$, we will get a period of 1. If t is time in seconds, we will have one period per second.

¹A unit circle is simply a circle with a radius of 1

To get more or less periods per second we multiply t by that amount. For example $\sin(2\pi t 440)$ has 440 periods per second.

Periods per second is measured in *Hertz* which has the unit $1/s$, which is also called the frequency of a wave. To calculate the period of a sine wave at a specific frequency f , you use $1/f$, e.g. a sine wave at $440Hz$ will have a period of $1/f = 0.002s$.

We can change the starting point of a wave by adding a phase component. A sin wave with a phase of $\pi/2$ is equivalent to a cos wave. The phase component is usually written as θ

A sine wave swings between -1 and 1 . To increase or decrease the amplitude, we multiply with a gain factor A .

1.2 Phasors

A *Phasor* is a complex sinusoid defined by Euler's equation

$$e^{jx} = \cos(x) + j \sin(x), \quad (2)$$

such that we have

$$\cos(x) = \text{Re}\{e^{jx}\} \quad (3)$$

$$\sin(x) = \text{Im}\{e^{jx}\} \quad (4)$$

1.3 Digital Signals

Unlike real-world *analog* signals, which are continuous, digital signals are represented by discrete numbers at specific points in time — a list of numbers. Each number is separated by some amount of time called the *sampling rate*.

If we wish to convert a analog signal to a digital one, we must sample it. This is done by measuring the analog signal one time each *sampling period* ($1/\text{samplerate}$). Given the analog signal $x(t)$ then we have the digital signal

$$x[n] = x(nT), \quad (5)$$

where T is the sampling period and n is the *sample index*. Notice that it is convention to use square brackets for discrete signals and parentheses for continuous signals.

ADC) If we are measuring a real-world signal, we will use an *Analog to Digital Converter (ADC)*. Since computers only have a finite amount of memory, we will need to choose how precise our measurements will be. If

we represent our sample with b bits then we will have 2^b levels to work with. The process of rounding our sample to the nearest representable number is called *quantization*.

1.4 DSP Systems

A DSP system (or just "a system") is a computer program — an algorithm — which takes one or more digital signal and returns one or more digital signals.

1.5 Causal Systems

In systems with memory, we need to agree on what should happen when $n < 0$. Usually we will use $x[n] = 0$ for $n < 0$. Such a system is said to be *causal*.

If we have a system where $x[n] \neq 0$ for $n < 0$, then the system is said to be *non-causal*.

1.6 Difference Equations

The general form of a difference equation is

$$y[n] = a_1 y[n-1] + a_2 y[n-2] + \cdots + a_N y[n-N] + b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + \cdots + b_L x[n-L]. \quad (6)$$

Also written as

$$y[n] = \sum_{k=1}^N a_k y[n-k] + \sum_{k=0}^L b_k x[n-k]. \quad (7)$$

Where the $y[\cdot]$ components determine the recursive (feedback) characteristic, and the $x[\cdot]$ components the non-recursive characteristic.

1.7 Linear Time-Invariant System

A system $T(x[n])$ is *linear time-invariant* (LTI) if it is both linear and time-invariant.

A system is linear if it supports the scalability property

$$T(\alpha x[n]) = \alpha T(x[n]). \quad (8)$$

As well as the superposition property

$$T(x_3[n]) = T(x_1[n] + x_2[n]) = T(x_1[n]) + T(x_2[n]), \quad (9)$$

where $x_3[n] = x_1[n] + x_2[n]$.

A system is time-invariant if it meets the condition

$$x[n - L] = y[n - L] \quad (10)$$

for some delay L — a delay in the input signal will cause a corresponding delay in the output signal.

A system is LTI if all of its subcomponents are LTIs.

1.8 Impulse Response

An impulse is a short spike in amplitude. The ideal impulse signal is defined by

$$\delta[n] = \begin{cases} 1, & \text{if } n = 0 \\ 0, & \text{otherwise} \end{cases}. \quad (11)$$

The *impulse response* of a system is how it reacts to an input of $x[n] = \delta[n]$. In this case the output of the system is normally denoted $h[n]$.

LTI systems can be described completely by their impulse responses. To obtain the output of a system based on an impulse response, one should convolute the input signal with the impulse response

$$T(x[n]) = x[n] * h_T[n]. \quad (12)$$

A system has *finite impulse response* (FIR) if it does not contain any recursive components. In that case the impulse response is determined by

$$h[n] = T(\delta[n]) = \sum_{k=0}^L b_k \delta[n - k]. \quad (13)$$

The impulse response of FIR systems will always converge to zero.

A system has *infinite impulse response* (IIR) if it contains recursive components. In that case the impulse response is determined by

$$h[n] = T(\delta[n]) = \sum_{k=1}^N a_k h[n - k] + \sum_{k=0}^L b_k \delta[n - k]. \quad (14)$$

Some IIR systems will converge to a state of near-zero. Such systems are said to be *stable*. IIR systems can also be *unstable*, which means that their output will be non-convergent (i.e. blow up).

1.9 Convolution

The *convolution* of two signals $x[n]$ and $h[n]$ is defined as

$$x[n] * h[n] = \sum_{m=-\infty}^{m=\infty} x[n-m] h[m]. \quad (15)$$

In reality m will be bounded by the sizes of the signals. If $x[n]$ has size N and $h[n]$ size M , then m should be within 0 and M , since no samples is defined before 0, and no samples in $h[x]$ are defined after M . The equation is then

$$x[n] * h[n] = \sum_{m=0}^M x[n-m] h[m]. \quad (16)$$

The result will be defined for n in $0 \leq n < N + M$.

Convolution is commutative, meaning that

$$x[n] * h[n] = h[n] * x[n]. \quad (17)$$

It is also associative, which means that

$$g[x] * (x[n] * h[n]) = (g[x] * x[n]) * h[n], \quad (18)$$

as well as distributive

$$g[x] * (x[n] + h[n]) = (g[x] * x[n]) + (g[x] * h[n]). \quad (19)$$

1.10 Digital Frequency

Digital frequency is defined as

$$\theta = 2\pi fT, \quad (20)$$

where f is the frequency in Hz and T is the sampling period. This means that $\theta = 0$ denotes DC, $\theta = \pi$ the Nyquist frequency ($1/2 * \text{samplerate}$), and $\theta = 2\pi$ the sampling frequency.

1.11 Frequency Response

The frequency response of a system describes how it responds to signals of a specific frequency, in terms of magnitude and phase. The frequency response of a system is usually shown as a plot with frequency on the x-axis

and amplitude on the y-axis. If the plot is presented with a logarithmic frequency scale, it is sometimes called a *Bode* diagram.

Using the equation

$$H(e^{j\theta}) = \frac{\sum_{k=0}^L b_k e^{-j\theta k}}{\sum_{k=1}^N a_k e^{-j\theta k}}, \quad (21)$$

we get a complex number which describes the phase and magnitude response for a signal at the digital frequency θ . The magnitude $\|H(e^{j\theta})\|$ is the gain at that frequency. The angle $\angle H(e^{j\theta})$ is the *phase shift* at that frequency.

The $H(e^{j\theta})$ term is derived by first selecting the input signal

$$x[n] = e^{j\theta n}, \quad (22)$$

and then convolving with the impulse response of our system

$$y[n] = \sum_{m=-\infty}^{\infty} h[m] x[n-m] = \sum_{m=-\infty}^{\infty} h[m] e^{j\theta(n-m)} \quad (23)$$

$$= e^{j\theta n} \sum_{m=-\infty}^{\infty} h[m] e^{j\theta m} = x[n] \sum_{m=-\infty}^{\infty} h[m] e^{j\theta m} \quad (24)$$

$$= x[n] H(e^{j\theta}), \quad (25)$$

so we have

$$H(e^{j\theta}) = \sum_{m=-\infty}^{\infty} h[m] e^{j\theta m}, \quad (26)$$

and since

$$y[n] = x[n] H(e^{j\theta}), \quad (27)$$

we have

$$H(e^{j\theta}) = \frac{y[n]}{x[n]} \quad (28)$$

To get to our final definition of $H(e^{j\theta})$ we first note that

$$y[n] = \sum_{m=-\infty}^{\infty} h[m] x[n-m] \quad (29)$$

$$y[n] = x[n] H(e^{j\theta}) = e^{j\theta n} H(e^{j\theta}) \quad (30)$$

And then rearranging

$$e^{j\theta n} H(e^{j\theta}) = \sum_{k=1}^N a_k e^{j\theta(n-k)} H(e^{j\theta}) + \sum_{k=0}^L b_k e^{j\theta(n-k)} \quad (31)$$

$$e^{j\theta n} H(e^{j\theta}) = e^{j\theta n} \left(\sum_{k=1}^N a_k e^{-j\theta k} H(e^{j\theta}) + \sum_{k=0}^L b_k e^{-j\theta k} \right) \quad (32)$$

$$H(e^{j\theta}) = \sum_{k=1}^N a_k e^{-j\theta k} H(e^{j\theta}) + \sum_{k=0}^L b_k e^{-j\theta k} \quad (33)$$

$$1 = \sum_{k=1}^n a_k e^{-j\theta k} + \frac{\sum_{k=0}^L b_k e^{-j\theta k}}{H(e^{j\theta})} \quad (34)$$

$$1 - \sum_{k=1}^n a_k e^{-j\theta k} = \frac{\sum_{k=0}^L b_k e^{-j\theta k}}{H(e^{j\theta})} \quad (35)$$

$$\frac{1 - \sum_{k=1}^n a_k e^{-j\theta k}}{\sum_{k=0}^L b_k e^{-j\theta k}} = \frac{1}{H(e^{j\theta})} \quad (36)$$

$$H(e^{j\theta}) = \frac{\sum_{k=0}^L b_k e^{-j\theta k}}{1 - \sum_{k=1}^n a_k e^{-j\theta k}} \quad (37)$$

1.12 Z-transform

The z -transform is a way of analysing the frequency response of a system. It is defined by

$$Z(x[n]) = X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}, \quad (38)$$

and has the shifting property

$$Z(x[n-m]) = z^{-m} \sum_{n=-\infty}^{\infty} x[n] z^{-n} = z^{-m} X(z). \quad (39)$$

If we apply the z -transform to the general difference equation we get

$$y[n] = \sum_{k=1}^N a_k y[n-k] + \sum_{k=0}^L b_k x[n-k] \quad (40)$$

$$Y(z) = \sum_{k=1}^N a_k z^{-k} Y(z) + \sum_{k=0}^L b_k z^{-k} X(z) \quad (41)$$

$$(42)$$

from which we arrive at

$$\frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^L b_k z^{-k}}{1 - \sum_{k=1}^N a_k z^{-k}}, \quad (43)$$

which is called the *transfer function*. Note that setting $z = e^{j\theta}$ makes the above equation equal to the frequency response (Eq. 37).

We use the above equation to solve for *poles* and *zeros* of the system. The zeros are found by solving the equation

$$\sum_{k=0}^L b_k z^{-k} = 0, \quad (44)$$

which is the numerator of Eq. 43 — when this is zero, the system response will be zero.

To find the poles, we solve for the denominator like

$$1 - \sum_{k=1}^N a_k z^{-k} = 0, \quad (45)$$

which gives us the places where the ratio explodes into infinity.

In reality the poles and zeros will cancel each other out by different amounts based on the input frequency. Therefore, the response will seldom actually be zero or infinity.

2 Recipes

2.1 Echo and Delay

An *echo* effect, in its simplest form, is achieved by mixing a signal with a delayed version of itself

$$y[n] = x[n] + b_m x[n - m], \quad (46)$$

where m is the delay amount in samples and b_m the amplitude of the echo.

The above equation will only give one reflection, if you want multiple reflections, you will need a feedback component:

$$y[n] = x[n] + a_m y[n - m], \quad (47)$$

You can get creative with delay effects by adding more delay taps, or applying effects to the delayed components — a low-pass filter can be used to achieve a dampening effect.

2.2 Chorus

The chorus effect is achieved by mixing a signal with a delayed version of itself, while varying the delay. The delay usually varies between 20 and 50 ms. This causes a doubling effect, like when multiple instruments are playing together.

The basic effect looks like

$$y[n] = x[n] + x[n - g[n]] \quad (48)$$

where $g[n]$ is the function determining the delay for sample n .

You can easily get creative with the chorus effect by using feedback instead of delay, and by adding multiple delays.

2.3 Flanging

The flanger effect is achieved by mixing a signal with a delayed version of itself, while varying the delay. The delay is usually below 20 ms. This creates constructive and destructive interference at specific frequencies, resulting in a swooshing sound.

The basic effect looks like

$$y[n] = x[n] + x[n - g[n]] \quad (49)$$

where $g[n]$ is the function determining the delay for sample n .

You can easily get creative with the flanger effect by using feedback instead of delay, and by adding multiple delays.

2.4 Ring Modulation and Amplitude Modulation

Ring modulation is a simple but powerful effect achieved by multiplying two signals,

$$y[n] = x_m[n] \cdot x_c[n], \quad (50)$$

where $x_m[n]$ is the modulator signal, $x_c[n]$ is the carrier signal.

Amplitude modulation (AM) is a related effect, achieved by mixing some of the modulated signal back into the output,

$$y[n] = A_m \cdot x_m[n] \cdot x_c[n] + A_c \cdot x_c[n], \quad (51)$$

where A_m is the amplitude of the modulated signal, and A_c is the amplitude of the carrier signal.

2.5 Biquad Filters

Biquad filters are a general form of second-order two pole, two zeros filters. They have the difference equation

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}. \quad (52)$$

If we multiply this by z^2/z^2 we get

$$H(z) = \frac{b_0 z^2 + b_1 z + b_2}{z^2 + a_1 z + a_2}, \quad (53)$$

and see that both the numerator and denominator are quadratic equations, hence the name.

Biquad filters can be used to implement many common types of filter including low-pass, high-pass, bandpass, band-reject and all-pass.

For calculating the coefficients of these filters we will need

$$K = \tan(\pi f_c / f_s), \quad (54)$$

where f_c is the filter center frequency and f_s is the sample rate.

2.5.1 Low-pass Filter

Low-pass biquad filters are designed by selecting a cutoff frequency f_c , and a Q factor which controls the gain around f_c . When $Q = 1/\sqrt{2}$ the filter will be flat around f_c — values above or below will cause amplification or attenuation.

The coefficients are then calculated by:

$$b_0 = \frac{K^2 Q}{K^2 Q + K + Q} \quad (55)$$

$$b_1 = \frac{2K^2 Q}{K^2 Q + K + Q} \quad (56)$$

$$b_2 = b_0 \quad (57)$$

$$a_1 = \frac{2Q(K^2 - 1)}{K^2 Q + K + Q} \quad (58)$$

$$a_2 = \frac{K^2 Q - K + Q}{K^2 Q + K + Q} \quad (59)$$

[Zolzer, 2011]

2.5.2 High-pass Filter

High-pass biquad filters are designed by selecting a cutoff frequency f_c , and a Q factor which controls the gain around f_c . When $Q = 1/\sqrt{2}$ the filter will be flat around f_c — values above or below will cause amplification or attenuation.

The coefficients are then calculated by:

$$b_0 = \frac{K^2 Q}{K^2 Q + K + Q} \quad (60)$$

$$b_1 = -\frac{2Q}{K^2 Q + K + Q} \quad (61)$$

$$b_2 = \frac{Q}{K^2 Q + K + Q} \quad (62)$$

$$a_1 = \frac{2Q(K^2 - 1)}{K^2 Q + K + Q} \quad (63)$$

$$a_2 = \frac{K^2 Q - K + Q}{K^2 Q + K + Q} \quad (64)$$

[Zolzer, 2011]

3 References

References

[Zolzer, 2011] Zolzer, U. (2011). *DAFX: Digital Audio Effects*. Wiley Publishing, 2nd edition.

Index

- (, 3
- AM, *see* Amplitude modulation
- Amplitude, 3
- Amplitude modulation, 10
- Biquad filter, 11
- Bode diagram, 6
- Chorus effect, 10
- Convolution, 6
- Cos, 2
- Dampening, 9
- Delay effect, 9
- Digital frequency, 6
- Digital signals, 3
- Echo effect, 9
- Finite impulse response, 5
- FIR, *see* Finite impulse response
- Flanging, 10
- Frequency, 3
- Frequency response, 6
- Hertz, 3
- High-pass filter (biquad), 12
- Hz, *see* Hertz
- IIR, *see* Infinite impulse response
- Impulse response, 5
- Infinite impulse response, 5
- Linear Time-Invariant System, 4
- Linearity property, 4
- Low-pass filter (biquad), 11
- LTI, *see* Linear Time-Invariant System
- period, 2
- Phase, 3
- Phasor, 3
- Poles, 9
- Quantization, 3
- Ring modulation, 10
- Sampling period, 3
- Sampling rate, 3
- Second order filter, 11
- Sin, 2
- Sine waves, 2
- Stability, 5
- Stable, 5
- Superposition property, 5
- System, 4
- Time-invariant, 5
- Transfer function, 9
- Unstable, 5
- Z-transform, 8
- Zeros, 9