Summary
Signals and Systems 2020-2021 (WBAI016-05)

Bjorn Pijnacker

Contents

About this summary

This summary contains chapter 2-9 of the book Digital Signal Processing First (2nd ed.) by James H. McClellan, Ronald W. Schafer, and Mark A. Yoder. Any sections about MATLAB or examples are skipped. Sections I consider irrelevant will also be skipped. Section numbering will be consistent with the book (2nd ed.!) for the entire summary.

Chapter 2

Sinusoids

In signals and system, the basis for every (continuous) signal is a sinusoid. The general formula for a sinusoid is

$$x(t) = A\cos(\omega_0 t + \phi)$$

A, ω_0, and ϕ are fixed numbers for a particular cosine signal. Specifically, A is called the amplitude, ω_0 is called the radian frequency, and ϕ is the phase of the signal.

2.1 Tuning-Fork Experiment

The reason that cosine waves are so important is that many physical systems generate signals that can be modeled using cosine functions versus time. Consider a tuning fork. When struck, our tuning fork emits a 'pure' frequency, often of 440Hz. If we record the signal and look at the reconstructed sinusoidal wave, we should see an almost perfect sinusoidal shape.

2.2 Review of Sine and Cosine Functions

The sine and cosine functions are closely related. See Table 2.1 for some basic properties of the sine and cosine functions. Later on we will express sinusoidal signals by their complex phasor equivalents, so we do not discuss the trigonometric identities here, as those won't be useful.

2.3 Sinusoidal Signals

We have seen the general form of a cosine $x(t) = A\cos(\omega_0 t + \phi)$. We can also write

$$x(t) = A\cos(2\pi f_0 t = \phi)$$

Property	Equation
Equivalence	$\sin\theta = cos(\theta - \pi/2)$ or $\cos\theta = sin(\theta + \pi/2)$
Periodicity	$\forall_{k\in\mathbb{Z}} : \cos(\theta + 2k\pi) = \cos\theta$
Evenness of cosine	$\cos(-\theta) = \cos\theta$
Oddness of sine	$\sin(-\theta) = -\sin\theta$
Zeros of sine	$\forall_{k\in\mathbb{Z}} : \sin k\pi = 0$
Ones of cosine	$\forall_{k\in\mathbb{Z}} : \cos 2k\pi = 1$

Table 2.1: Basic properties of the sine and cosine functions

where $\omega_0 = 2\pi f_0$. We call f_0 the cyclic frequency, because it directly corresponds the cycles per second of the sinusoid.

2.3.1 Relation of Frequency to Periods

In a sinusoid we enter the frequency. In a plot, it's often much easier to find the period instead. The period T_0 is the amount of time that a single cycle of the sinusoid takes. It's related to the frequency with

$$f_0 = \frac{1}{T_0} \qquad \omega_0 = \frac{2\pi}{T_0}$$

A special case is when our frequency is 0. We treat this as there not being a period, because our signal will also be a constant value. We call this a DC signal.

2.3.2 Phase and Time Shift

The phase parameter ϕ determines (together with the frequency) the locations of the maxima and minima of a cosine wave, as well as the zero crossings in between. This introduces the concept of time shift, because we can change the phase to horizontally shift a signal. Whenever a signal can be represented in the form $x_1(t) = s(t - t_1)$ then we say that $x_1(t)$ is a time shifted version of $s(t)$. If t_1 is a positive number, then the shift is to the right and we say that the signal is delayed. When t_1 is negative, the signal goes to the left and we say it is advanced in time.

2.5 Complex Exponentials and Phasors

So far we've shown that cosine signals are useful mathematical representations for signals. It turns out, however, that analysis and manipulation of sinusoidal signals is greatly simplified by dealing with complex exponential signals instead. We will first give a refresher on complex numbers, then continue to complex exponential signals.

2.5.1 Review of Complex Numbers

A complex number is a pair of two real numbers. Complex numbers may be represented by the notation $z = a+jb$, where a and b are real numbers, and $j = \sqrt{-1}$. This notation is called the cartesian form of a complex number, which we can represent in the complex plane. In this plane, the horizontal axis plots a ($\Re\{z\}$), the vertical axis plots jb ($\Im\{z\}$). Because these numbers are in a plane, we can also represent them with a polar notation with length r and angle θ. Using some trigonometry and Pythagorean's theorem, we can derive a method for converting between these forms:

$$x = r\cos\theta \qquad y = r\sin\theta$$

and

$$r = \sqrt{x^2 + y^2} \qquad \theta = \arctan(x/y)$$

This $r\angle\theta$ notation is not the nicest, so we can use Euler's formula to convert between the complex exponential polar form using the r and θ factors. This exponential is given by

$$z = re^{j\theta} = rcos\theta + jr\sin\theta$$

As can be seen, this also immediately gives a nice conversion between polar and cartesian form.

2.5.2 Complex Exponential Signals

A complex exponential signal is defined as

$$z(t) = Ae^{j(\omega_0 t+\phi)}$$

Notice that the terms in here are exactly the same as in the previously defined general cosine function. Using Euler's equation, we can rewrite this complex exponential in cartesian form, yielding

$$z(t) = A\cos(\omega_0 t + \phi) + jA\sin(\omega_0 t + \phi)$$

If we were to plot this in the complex plane with $t \in [-\pi, \pi]$, and $A = 1$ we would see the unit circle. We can also use this complex exponential signal to show the trigonometric identities for sines and cosines, but we're not going to now.

2.5.3 The Rotating Phasor Interpretation

If we define a complex number $X = Ae^{j\phi}$ then a complex exponential can be expressed as $Xe^{j\omega_0 t}$. Notice that the phase and amplitude components disappear from the complex exponential, as these are now encased in X. The $e^{j\omega_0 t}$ component now only contains the frequency of our signal. We call this X the complex amplitude or phasor.

If we plot our complex amplitude in the complex plane, we see a vector representing the initial direction and the magnitude of our signal. If we multiply this by $e^{j\omega_0 t}$ for a range of t-values, and then plot it again, we see that our phasor rotates around its origin. If we were to plot the horizontal (or vertical) component of this vector, we would see a sine or cosine signal appearing with the frequency as defined by $e^{j\omega_0 t}$.

2.5.4 Inverse Euler Formulas

The inverse Euler formulas are defined as

$$\cos\theta = \frac{e^{j\omega} + e^{-j\omega}}{2} \qquad \sin\theta = \frac{e^{j\omega} - e^{-j\omega}}{2j}$$

These formulas can be used to express a cosine or sine function as a sum of complex exponentials without needing to discard any real/imaginary components first.

2.6 Phasor Addition

In many cases we need to add multiple signals together. If all signals in a sum have the same frequency, then this addition greatly simplifies. If the frequency component is the same, then we only need to find out the new amplitude and phase; the frequency will be unchanged for the summed signal.

2.6.1 Addition of Complex Numbers

Two complex numbers $z_1 = a_1 + jb_1$ and $z_2 = a_2 + jb_2$ can be added as $z = a_1 + a_2 + j(b_1 + b_2)$

2.6.2 Phasor Addition Rule

The phasor representation of cosine signals can be used to show that the following sum of sinusoids

$$x(t) = \sum_{k=1}^{N} A_k \cos(\omega_0 t + \phi_k) = A\cos(\omega_0 t + \phi)$$

can be carried out by adding complex amplitudes to get A and ϕ:

$$\sum_{k=1}^{N} A_k e^{j\phi_k} = Ae^{j\phi}$$

To add these complex amplitudes, we convert them to cartesian form using Euler's formula, then add them together, and then convert them back to complex exponential form. The obtained values for A and ϕ can then be substituted in $x(t)$ to obtain the final function for our summed sinusoid.

Chapter 3

Spectrum Representation

In the previous chapter we have seen that a sinusoidal waveform can be represented by

$$x(t) = A\cos(2\pi f_0 + \phi) = \Re\{Xe^{j2\pi f_0 + \phi}\}$$

We also showed how to add multiple signals with the same frequency. In this chapter, we'll show how waveforms can be represented as sums of sinusoids having different frequencies and corresponding amplitudes using the spectrum representation.

3.1 The Spectrum of a Sum of Sinusoids

If we have a sum of multiple sinusoids

$$x(t) = X_0 + \sum_{k=1}^{N} A_k \cos(2\pi f_k t + \phi_k)$$

then we can represent it as a spectrum. We first rewrite the sum using Euler's formula, to obtain two components per cosine:

$$\begin{aligned} x(t) &= X_0 + \sum_{k=1}^{N} A_k \cos(2\pi f_k t + \phi_k) \\ &= X_0 + \sum_{k=1}^{N} \Re\{X_k e^{j2\pi f_k t}\} \\ &= X_0 + \sum_{k=1}^{N} \left\{ \frac{X_k e^{j2\pi f_k t}}{2} + \frac{X_k^* e^{j2\pi f_k t}}{2} \right\} \end{aligned}$$

This sum can be written as a spectrum. For each frequency f_k, we take it on the horizontal axis, and then on the vertical axis we plot the corresponding X_k component. In general, this gives the set of pairs

$$\left\{(0, X_0), (f_1, \tfrac{1}{2}X_1), (-f_1, \tfrac{1}{2}X_1^*), \ldots, (f_k, \tfrac{1}{2}X_k), (-f_k, \tfrac{1}{2}X_k^*)\right\}$$

3.1.1 Notation Change

Writing the entire $\frac{1}{2}X_k e^{j2\phi f_k t}$ component many times is a lot of work, so we define a_k:

$$a_k = \begin{cases} A_0 & k = 0 \\ \frac{1}{2}A_k e^{j\phi_k} & k \neq 0 \end{cases}$$

This allows us to write our spectrum as a set of (f_k, a_k) pairs, and it allows us to write the sum we obtained before simply as

$$x(t) = \sum_{k=-N}^{N} a_k e^{j2\pi f_k t}$$

3.1.3 Analysis vs Synthesis

In this context, synthesis is the process of using the frequency-domain representation (i.e. the spectrum) to recreate the original signal waveform $x(t)$. Analysis is the operation on the time-domain signal that produces the frequency-domain spectrum.

Up to this point, we have treated the special case that the signal of interest is a sum of a constant and finite number of sinusoids. This makes the analysis and synthesis operation pretty easy. We will later discuss the Fourier series that applies in general, and which can thus also be used to analyze more complex waveforms than those discussed up to this point.

3.2 Sinusoidal Amplitude Modulation

Amplitude modulation occurs when we multiply two sinusoidal signals together, instead of adding them. The amplitude of one signal is multiplied by the value of the other, basically.

3.2.1 Multiplication of Sinusoids

To obtain the spectrum of the multiplication of two sinusoids, we must first rewrite is as a sum. We could use the reverse Euler's formula and obtain the sum each time, but we can also do this for the general case, after which we can deduct a simple formula for the multiplication of two sinusoids:

$$\cos\theta\cos\phi = \frac{\cos(\theta-\phi)+\cos(\theta+\phi)}{2}$$

We can do the same for other combinations of cos/sin multiplications (e.g. $\cos\theta\sin\phi$) but you can just look at the formula sheet for those instead.

3.2.2 Beat Note Waveform

A beat note is produced by adding two equal-amplitude sinusoids with nearly identical frequencies. The formula in the previous subsection indicates that such a beat note could be written as the product of two sinusoids. We derive a general relationship:

$$\begin{aligned}
&x(t) = \cos(2\pi f_1 t) + \cos(2\pi f_2 t) \\
&f_c = \tfrac{1}{2}(f_1 + f_2) && \text{(center frequency)} \\
&f_\Delta = \tfrac{1}{2}(f_1 - f_2) && \text{(deviation frequency)} \\
&x(t) = \cos(2\pi f_c t)\cos(2\pi f_\Delta t)
\end{aligned}$$

3.2.3 Amplitude Modulation

Multiplying frequencies is very common in communication systems where modulation is used to allow for better information transit. Amplitude modulation is the process of multiplying a high-frequency sinusoid signal by a low-frequency signal. The AM signal is of the form

$$x(t) = v(t)\cos(2\pi f_c t)$$

where it is assumed that the frequency of the cosine term is much higher than any of the frequencies in $v(t)$.

3.2.4 AM Spectrum

The spectrum of an AM signal is easy to describe. It consists of copies of the spectrum of $v(t)$ centered at $f = \pm f_c$. To derive the exact spectrum, we just write out $v(t)$ and the cosine term in complex exponentials, and perform the multiplication until we have a sum of complex exponentials that represent our spectrum.

3.2.5 The Concept of Bandwidth

The bandwidth of a spectrum (or signal) is the width between the highest and lowest frequency of the signal. We might also indicate where the center of this band is.

3.3 Operations on the Spectrum

In spectrum representation, operations that were difficult in the time-domain representation become much easier.

3.3.1 Scaling or Adding a Constant

Multiplying a signal by a scale factor (i.e. $\gamma x(t)$) multiplies all of the complex amplitudes in its spectrum by the same scale factor (γ) but leaves the frequencies unchanged.

Adding a constant c to the signal leaves everything unchanged except the complex amplitude at $f = 0$. The new value at f_0 becomes $a_0 + c$.

3.3.2 Adding Signals

Adding two signals can be done simply by merging their sets. If a frequency is present in both signals, then their complex amplitudes must be summed by phasor addition.

3.3.3 Time-Shifting $x(t)$ Multiplies a_k by a Complex Exponential

If we form a new signal $y(t)$ by time shifting $x(t)$ then the frequencies remain the same, but there is a phase change in the complex amplitudes of the spectrum. We multiply every complex exponential a_k as below.

$$y(t) = x(t - \tau_d) \quad \longleftrightarrow \quad b_k = a_k e^{-j2\pi f_k \tau_d}$$

3.4 Periodic Waveforms

A period signal satisfies the condition that $x(t+T_0) = x(t)$ for all t. The time interval T_0 is in this case the period of the signal $x(t)$. If we wish to find the fundamental frequency F_0 then we take the greatest common divisor of all frequencies of the signals that together sum up to make our signal. This is the fundamental frequency of the resulting signal after summing the sinusoids.

3.5 Fourier Series

With the Fourier series, we show that any waveform can be synthesized with a sum of complex exponential signals. This is expressed in general by using the Fourier synthesis summation:

$$x(t) = \sum_{j=-\infty}^{\infty} a_k e^{j(2\pi/T_0)kt}$$

3.5.1 Fourier Series: Analysis

The Fourier coefficients a_k for the harmonic sum can be derived from the signal $x(t)$ by using the Fourier analysis integral

$$a_k = \frac{1}{T_0} \int_0^{T_0} x(t) e^{-j(2\pi/T_0)kt} dt$$

Only one period of $x(t)$ is required for the integral, so T_0 should be the fundamental period of $x(t)$.

3.6 Time-Frequency Spectrum

So far we've made the assumption that the frequency of a signal is constant. This is not the case for most real life signals. We introduce the notion of a time-frequency spectrum, where the frequency of a signal is plotted against time. Most often we see this as a spectrogram, where the intensity of a certain frequency is shown with brightness.

3.7 Frequency Modulation: Chirp Signals

3.7.1 Chirp or Linearly Swept Frequency

If we want to create a signal that covers an interval of frequencies then we create a 'chirp' signal. A chirp signal's frequency changes linearly over time. We get a formula

$$x(t) = \Re\{Ae^{j\psi(t)} = Acos(\psi(t))$$

If we want to know the speed at which the frequency changes, we need to take the derivative of $\psi(t)$. We obtain a notion of the instantaneous frequency ω_i, defined as

$$\omega_i(t) = \frac{d}{dt}\psi(t)$$

The frequency variation produced by a time-varying angle function such as $\psi(t)$ is called frequency modulation, and signals of this class are called linear FM signals.

Chapter 4

Sampling and Aliasing

4.1 Sampling

So far we've only discussed continuous signals, where a signal is (modeled by) a mathematical function. If we want to process a signal with a computer, we need to first transform it to a discrete signal that the computer understands. A signal on a computer must be discrete. We sample the continuous waveform at some sample period T_s to obtain samples at points in the signal. These samples then digitally represent our signal. We call this a discrete-time signal, as opposed to a continuous-time signal. We denote values of a discrete-time signal by $x[n]$.

To obtain a sample from a continuous-time signal we can sample a it at equally spaced time instants, $t_n = nT_s$. The samples are obtained by

$$x[n] = x(nT_s) \qquad -\infty < n < \infty$$

where $x(t)$ represents the signal we want so sample. We can also compute the values of a discrete-time signal directly from a formula.

4.1.1 Sampling Sinusoidal Signals

If we sample a sinusoid of the form $A\cos(\omega t = \phi)$, we obtain

$$\begin{aligned} x[n] &= x(nT_s) \\ &= A\cos(\omega n T_s + \phi) \\ &= A\cos(\hat{\omega} n + \phi) \end{aligned}$$

where we define $\hat{\omega}$ to be the normalized radian frequency

$$\hat{\omega} = \omega T_s = \frac{\omega}{f_s}$$

4.1.2 The Concept of Aliases

We introduce the concept of an alias, where two different discrete-time sinusoid formulas can define the same signal values. An easy to see alias is the one of the frequency $(\omega + 2)$. Because of the cyclic nature of the cosine, this defines the same signal as the original ω.

The identity $A\cos(2\pi - \omega) = \cos\omega$ also leads to aliases. For example, because of this identity, $\cos(1.6\pi n)$ is aliased by $\cos(0.4\pi n)$.

When talking about aliasing, the principle alias is defined to be the unique alias frequency in the interal $-\pi < \hat{\omega} < \pi$. In the above case, the principle alias is 0.4π.

4.1.3 Sampling and Aliasing

When sampling, aliasing can occur due to which we cannot reconstruct our signal. We need the normalized frequency $\hat{\omega}_0$ be the principle alias, that is,

$$-\pi < \hat{\omega}_0 = \omega_0 T_s < \pi$$

If this is not satisfied, we say that "aliasing has occurred". When we use the term aliasing in this context, we mean that when a signal is sampled, the resulting samples are identical to those obtained by sampling a lower frequency signal (corresponding to the principle alias).

4.1.4 Spectrum of a Discrete-Time Signal

When drawing a spectrum for a discrete-time signal, we draw a spectrum showing the alias of the signal. The first lines show the principle alias, and from there on outwards we show a line for every alias frequency that could have created our samples. In all cases this is an infinite spectrum, so we never draw it completely.

4.1.5 The Sampling Theorem

The aliasing naturally raises the question of how frequently we must sample in order to retain enough information to reconstruct the original continuous-time signal from its samples. This answer is given by the Shannon sampling theorem:

A continuous-time signal $x(t)$ with frequencies no higher than f_{max} can be reconstructed exactly from its samples $x[n] = x(nT_s)$, if the samples are taken at a rate $f_s = 1/T_s$ that is strictly greater than $2f_{max}$.

If we sample at less than the specified f_s, reconstruction becomes difficult or impossible. Section 4.2 shows this more in-depth.

4.1.6 Ideal Reconstruction

The sampling theorem suggests that there is a method to reconstruct a continuous-time signal from its samples. This reconstruction process would undo the C-to-D conversion, and as such we call it D-to-C conversion. Since the sampling process of the ideal C-to-D conversion is defined by the mathematical substitution $t = n/f_s$, we might expect D-to-C conversion to be

$$y(t) = y[n]\Big|_{n=f_s t} \qquad -\infty < t < \infty$$

but this is not the case. This substitution is not practical because it is only true when $y[n]$ is expressed as a sum of sinusoids. In theory, this would be a perfect D-to-C converter, but in practice we don't know the mathematical formula for $y[n]$. Instead, we are going to build a D-to-A converter that interpolates between the samples in $y[n]$. In section 4.3 we show a few different methods of achieving this.

Due to aliasing, if we convert from a discrete to a continuous signal, we have infinitely many frequencies to choose from for any component. As such, we define that we always use the principle alias frequency, which are guaranteed to lie in the range $-\pi < \hat{\omega} < \pi$. D-to-C converting from $\hat{\omega}$ to analog frequency will now always yield a signal with output frequencies between $\pm f_s/2$.

4.2 Spectrum View of Sampling and Reconstruction

4.2.1 Spectrum of a Discrete-Time Signal Obtained by Sampling

If we wish to plot a spectrum of a discrete-time sinusoid obtained by sampling then it is necessary to plot an infinite number of spectrum lines for a thorough representation of the spectrum, of course, this is not very practical. We can leave it at only plotting a few lines and indicating more exist in a predicable fashion.

Suppose we start with a continuous-time sinusoid $x(t) = A\cos(\omega_0 t + \phi)$ whose spectrum consists of two spectrum lines $\left\{(-w_0, \frac{1}{2}Ae^{-j\phi}), (w_0, \frac{1}{2}Ae^{j\phi})\right\}$. The spectrum of the sampled discrete-time signal $x[n] = A\cos((\omega_0/f_s)n + \phi)$ also has spectrum lines at $\hat{\omega} = \pm\omega_0/f_s$ and at all aliases at frequencies $\hat{\omega} = \pm\omega_0/f_s + 2\pi l$ with $l \in \mathbb{Z}$.

4.2.2 Over-Sampling

When we sample at a rate higher than twice the highest frequency to avoid problems with aliasing, we're over-sampling.

4.2.3 Aliasing Due to Under-Sampling

When our sampling rate is less than twice the highest frequency, our signal is under-sampled and we say that aliasing has occurred. In this case, if we reconstruct our signal, a principal alias exists that is lower than our original frequency, and thus our reconstructed signal will have a lower and different frequency from our original signal.

If we calculate the aliases for a certain sample then we can obtain the principal alias that is the frequency of what would be the reconstructed signal from our sample.

4.2.4 Folding Due to Under-Sampling

Under-sampling can lead to a type of aliasing called folding. This occurs when due to aliasing, the positive principal alias component is an alias of the negative original frequency component, and vice versa. When this happens, phase of the reconstructed signal changes sign. This means that when we sample a 100Hz signal at 125 samples/s, we get the same samples that we would have gotten by sampling a 25Hz sinusoid, but with opposite phase.

4.2.5 Maximum Reconstructed Frequency

If we sample, then reconstruct, a linear FM chirp signal, we get a good demonstration of aliasing and folding. When we look at the output frequency against the input frequency, we see that due to aliasing and folding, the output frequency goes up and down in a triangle-wave pattern. This occurs because the output cannot have a higher frequency than $f_s/2$.

4.3 Discrete-to-Continuous Conversion

4.3.1 Interpolation with Pulses

A general formula that describes a broad class of D-to-C converters is given by

$$y(t) = \sum_{n=-\infty}^{\infty} y[n]p(t - nT_s)$$

where $p(t)$ is the characteristic pulse shape of the converter. The equation states that the output signal is produced by superimposing scaled and time-shifted pulses. We need to choose a pulse shape $p(t)$ to reconstruct our signal. We have a few relatively simple choices, each with their own shortcomings.

4.3.2 Zero-Order Hold Interpolation

The simplest $p(t)$ that we can choose is a symmetric square pulse of the form

$$p(t) = \begin{cases} 1 & -\frac{1}{2}T_s < t \leq \frac{1}{2}T_s \\ 0 & \text{otherwise} \end{cases}$$

With this, the space between samples get filled with a continuous-time waveform like we wanted, though any samples is just 'stretched' horizontally to fill the space, which leads to a stair-step effect between samples. An upside of this conversion is that it is really fast; the only thing it needs to know is the current sample and the previous sample. Because the interpolation waveform is a constant, it is called zero-order interpolation.

4.3.3 Linear Interpolation

Linear interpolation uses first-order polynomial (straight-line) segments connecting adjacent samples. The waveform $p(t)$ is defined as

$$p(t) = \begin{cases} 1 - |t|/T_s & -T_s < t \leq T_s \\ 0 & \text{otherwise} \end{cases}$$

While this is a significant improvement over the zero-order hold interpolation, it still has a significant reconstruction error.

4.3.4 Cubic-Spline Interpolation

The pulse for this pulse shape consists of four cubic spline (third-order polynomial) segments. The duration is twice that of the triangular pulse and four times that of the square pulse. The reconstructed signal is at any point dependent on the two signals preceding and the two signals following the current time instant.

4.3.6 Ideal Bandlimited Interpolation

The pulse shape that gives the "ideal D-to-C conversion" is given by the following equation:

$$p(t) = \text{sinc}(t/T_s) = \frac{\sin(\pi t/T_s)}{\pi t/T_s} \quad \text{for} -\infty < t < \infty$$

THe infinite length of this pulse implies that to reconstruct a signal at time t exactly from its samples requires all the samples, and not just those near t.

Chapter 5

FIR Filters

5.1 Discrete-Time Systems

A discrete-time system is a computational process for transforming one sequence, called the input signal, into another sequence called the output signal. Systems are often depicted by block diagrams. In systems such as these, the input and output are discrete. These systems can as such be implemented completely digitally. In general we represent the operation of a system by the notation

$$x[n] \quad \stackrel{\mathcal{T}}{\longmapsto} \quad y[n]$$

which states that the output sequence y is related to x by the mapping $\mathcal{T}$.

An example of such a system is $y[n] = \max\{x[n], x[n-1], x[n-2]\}$. In this case, the output value depends on three input values. The output value is the max value of the last three values in the input sequence.

5.2 The Running-Average Filter

A simple but often-used system is the computation of a running average of multiple consecutive values of the input sequence. This is often used for smoothing out data prior to interpretation. The general form of an L-point running averager is

$$y[n] = \frac{1}{L}\sum_{k=0}^{L} x[n-k]$$

We can visualize this as a sliding window of width L over the last samples. We then take the average in this window, and this dictates our current sample value.

The given formula above is causal, because it uses only values from the past and the present. When a FIR filter uses values from the future, it becomes noncausal.

We prefer causal filters, because these run in real-time and are thus more suitable for signal processing applications that depend on fast processing.

5.3 The General FIR Filter

The general difference equation for a FIR filter is

$$y[n] = \sum_{k=0}^{M} b_k x[n-k]$$

where the coefficients b_k are fixed numbers. Usually the coefficients aren't all the same, but in the case of a running average they are, for example, $1/(M+1)$.

If we have a signal of finite length and we pass it through a FIR filter, we can see that the output can only be defined for locations where all of the components are defined. Usually we consider the undefined portions of a signal to just be zero, so that we can get the complete output. The output of a FIR filter is always the same length, or longer, than the original signal.

5.4 The Unit Impulse Response and Convolution

5.4.1 Unit Impulse Sequence

The unit impulse sequence is by far the easiest sequence. The mathematical notation of it is the Kronecker delta function

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

The shifted impulse is a concept that is very useful in representing signals and systems. For example, the formula $x[n] = 2\delta[n] + 4\delta[n-1] + 6\delta[n-2] + 4\delta[n-3] + 2\delta[n-4]$ spells out a signal that is $\{2, 4, 6, 4, 2\}$ in a more mathematical sense.

5.4.2 Unit Impose Response Sequence

The output of a filter is often called the response to the input. When the input is the unit impulse $\delta[n]$, the output is aptly names the unit impulse response $h[n]$. We can now restate the general formula for a FIR filter in terms of this:

$$h[n] = \sum_{k=0}^{M} b_k \delta[n-k]$$

As can be seen, the impulse response sequence is identical to the sequence of difference equation coefficients. Because of this, a FIR filter is completely defined by its impulse response.

5.4.3 FIR Filters and Convolution

A general expression for the output of a FIR filter in terms of the impulse response is

$$y[n] = \sum_{k=0}^{M} h[k]x[n-k]$$

This sum is called a finite convolution sum, and we say that the output is obtained by convolving the sequences $h[n]$ and $x[n]$. We use a star ($*$) to represent this operation and as such we write it as

$$y[n] = h[n] * x[n]$$

The Length of a Convolution

The length L_y of a convolution is defined in terms of the length of $x[n]$: L_x and the length of $h[n]$: L_h as

$$L_y = L_x + L_h - 1$$

Filtering the Unit-Step Signal

An input signal to a FIR system can be of infinite duration, so is the case with the unit step signal $u[n]$ which is defined as

$$u[n] = \begin{cases} 0 & n < 0 \\ 1 & n \geq 0 \end{cases}$$

Of course we can never find the filter result for the entire sequence, seeing as it is infinite, but we can do enough of the convolution to discover a general pattern, seeing as the signal is the same for all $n < 0$ and the same for all $n \geq 0$. The only part of interest will most likely be in the step where the signal goes from 0 to 1.

5.5 Implementation of FIR Filters

5.5.1 Building Blocks

Multiplier

This system multiplies a signal by a scalar. The output is given by the rule

$$y[n] = \beta x[n]$$

where the coefficient β is our scalar. This system is also a FIR filter, with impulse response $h[n] = \beta\delta[n]$

Adder

This system performs the addition of two signals. This system has two inputs and one output, and is not a FIR filter. When passed through this system, the two signals are summed pointwise e.g. the values with the same index are summed to create a new value at that index for the output.

Unit-Delay

A unit delay system performs a delay by one unit of time. It is a FIR filter with impulse response $h[n] = \delta[n-1]$.

5.6 Linear Time-Invariant (LTI) Systems

5.6.1 Time Invariance

A system is said to be time invariant if when an input is delayed by n_0, so is the output. This is, for example, not the case for a system where $x[n]$ is multiplied by n.

5.6.2 Linearity

Linear systems have the property that if $x_1[n] \mapsto y_1[n]$ and $x_2[n] \mapsto y_2[n]$ then

$$x[n] = \alpha x_1[n] + \beta x_2[n] \longmapsto y[n] = \alpha y_1[n] + \beta y_2[n]$$

The linearity condition is also called the principle of superposition. If the input is the sum (superposition) of two or more scaled sequences, then we can find the output due to each sequence acting alone and then add (superimpose) the separate scaled outputs.

5.6.3 The FIR Case

FIR systems satisfy both the linearity and time invariance conditions. A system that satisfied both of these is called an LTI system. Not all LTI systems are FIR systems, but all FIR systems *are* LTI systems.

Chapter 6

Frequency Response of FIR Filters

6.1 Sinusoidal Response of FIR Filters

LTI systems behave in a simple way when the input is a discrete-time complex exponential. We consider an example:

$$\begin{aligned}
x[n] &= Ae^{j\phi}e^{j\hat{\omega}n} && -\infty < n < \infty \\
y[n] &= \sum_{k=0}^{M} b_k A e^{j\phi} e^{j\hat{\omega}(n-k)} \\
&= \left(\sum_{k=0}^{M} b_k A e^{-j\hat{\omega}k}\right) Ae^{j\hat{\omega}}e^{j\hat{\omega}n} \\
&= \mathcal{H}(\hat{\omega})Ae^{j\phi}e^{j\hat{\omega}n} && -\infty < n < \infty
\end{aligned}$$

where

$$\mathcal{H}(\hat{\omega}) = \sum_{k=0}^{M} b_k A e^{-j\hat{\omega}k}$$

We see that the output of our system is the original input signal multiplied by some $\mathcal{H}(\hat{\omega})$. We name the quantity $\mathcal{H}(\hat{\omega})$ the frequency-response function for our system, or shortened, the frequency response.

An issue of notation arises for consistency with the z-transform in chapter 9. The righthand side of our $\mathcal{H}(\hat{\omega})$ contains powers of the complex exponential $e^{j\hat{\omega}}$. This is true for many expressions for the frequency response. We choose to write the frequency response as $H(e^{j\hat{\omega}})$ instead of $\mathcal{H}(\hat{\omega})$ to emphasize the ubiquity of $e^{j\hat{\omega}}$.

If we have a system with input $x[n] = Ae^{j\phi}e^{j\hat{\omega}n}$, then using the polar form of $H(e^{j\hat{\omega}})$ gives the result

$$y[n] = (|H(e^{j\hat{\omega}})| \cdot A)e^{j(\angle H(e^{j\hat{\omega}})+\phi)}e^{j\hat{\omega}n}$$

6.2 Superposition and the Frequency Response

The principle of superposition makes it very easy to find the output of an LTI system if the input is a sum of complex exponential signals. If, as an input, we have a signal that is the sum of multiple sinusoids, then we can first write it as a sum of complex exponentials, then multiply each of the terms by $H(e^{j\hat{\omega}})$ with the appropriate $\hat{\omega}$ filled in. We can then use the property of polarity of $H(e^{j\hat{\omega}})$ to find the result.

Using the frequency response, we see that we do not have to deal with the the time-domain description such as the difference equation or impulse response when the input is a complex exponential signal.

6.3 Steady-State and Transient Response

For a complex exponential signal to be multiplied by the frequency response we need the complex exponential to exist over $[-\infty, \infty]$. We can slightly relax this condition and still take advantage of the convenience of the frequency response. We see this by considering a suddenly applied signal from $n = 0$.

We see the following property. When a complex signal is suddenly applied, the output can be considered to be defined over three distinct regions. In the first region, $n < 0$, the input is zero and therefore so is the corresponding output. The second region is a transition region whose length is M samples. In this region, the complex multiplier of $e^{j\hat{\omega}n}$ depends upon n. This region is often called the transient part. In the third region, $n \geq M$, the output is identical to the output that would be obtained if the input were defined over the doubly infinite interval. This part of the output is generally called the steady-state part.

6.4 Properties of the Frequency Response

6.4.1 Relation to Impulse Response and Difference Equation

$H(e^{j\hat{\omega}})$ can be calculated directly from the filter coefficients $\{b_k\}$. We show this by writing the correspondence

$$h[n] = \sum_{k=0}^{M} h[k]\delta[n-k] \longleftrightarrow H(e^{j\hat{\omega}}) = \sum_{k=0}^{M} h[k]e^{-j\hat{\omega}k}$$

The process of going from the FIR difference equation or impulse response to the frequency response is straightforward. The reverse process is also simple. If we express $H(e^{j\hat{\omega}})$ in terms of powers of $e^{-j\hat{\omega}}$ we can simply extract the coefficients. To go to the frequency response, we simply replace write the powers of $e^{-j\hat{\omega}}$ for each coefficient.

6.4.2 Periodicity of $H(e^{j\hat{\omega}})$

An important property of a discrete-time LTI system is that its frequency response $H(e^{j\hat{\omega}})$ is always a periodic function with period 2π. This means that two complex exponential signals with frequencies differing by 2π have identical time samples. For this reason, it is always sufficient to specify the frequency response only over an interval of one period, usually $-\pi < \hat{\omega} < \pi$.

6.4.3 Conjugate Symmetry

The frequency response for $H(e^{j\hat{\omega}})$ is complex valued, but usually has a symmetry in its magnitude and phase that allows us to focus on just the interval $0 \leq \hat{\omega} \leq \pi$ when plotting. This is the property of conjugate symmetry

$$H(e^{-j\hat{\omega}}) = H^*(ej\hat{\omega})$$

6.6 Cascaded LTI Systems

Convolution in the time-domain is equal to multiplication in the frequency-domain, as such we can easily cascade systems. If we have systems $H_1(e^{j\hat{\omega}})$ and $H_2(e^{j\hat{\omega}})$, their cascaded version $H(e^{j\hat{\omega}})$ is simply

$$H(e^{j\hat{\omega}}) = H_1(e^{j\hat{\omega}})H_2(e^{j\hat{\omega}})$$

This illustrates that convolving two impulse responses is equivalent to multiplying their frequency response. For FIR systems the frequency response is a polynomial in variable $e^{-j\hat{\omega}}$. Multiplying them thus requires polynomial multiplication.

6.7 Running-Sum Filtering

Another common LTI system is the L-point running sum defined by the difference equation

$$y[n] = \sum_{k=0}^{L-1} x[n-k]$$

The frequency response of this filter is

$$H(e^{j\hat{\omega}}) = \sum_{k=0}^{L-1} e^{-j\hat{\omega}k}$$

Using a property of geometric series, we see this can be rewritten in a different form

$$\begin{aligned}
H(e^{j\hat{\omega}}) = \sum_{k=0}^{L-1} e^{-j\hat{\omega}k} &= \left(\frac{1 - e^{-j\hat{\omega}L}}{1 - e^{-j\hat{\omega}}}\right) \\
&= \left(\frac{e^{-j\hat{\omega}L/2}(e^{j\hat{\omega}L/2} - e^{-j\hat{\omega}L/2})}{e^{-j\hat{\omega}/2}(e^{j\hat{\omega}/2} - e^{-j\hat{\omega}/2})}\right) \\
&= \left(\frac{\sin(\hat{\omega}L/2)}{\sin(\hat{\omega}/2)}\right) e^{-j\hat{\omega}(L-1)/2}
\end{aligned}$$

We often express this in the form

$$H(e^{j\hat{\omega}}) = D_L(\hat{\omega})e^{-j\hat{\omega}(L-1)/2} \qquad D_L(\hat{\omega}) = \frac{\sin(\hat{\omega}L/2)}{\sin(\hat{\omega}/2)}$$

We refer to the function $D_L(\hat{\omega})$ as a Dirichlet form, where the subscript L indicates it comes from an L-point running sum.

Chapter 9

z-Transforms

9.1 Definition of the z-Transform

A finite-length signal $x[n]$ consists of a set of signal values $\{x[0], x[1], \ldots, x[L-1]\}$ that can be represented by

$$x[k] = \sum_{k=0}^{L-1} x[k]\delta[n-k]$$

The z-transform of the signal is defined by the formula

$$X(z) = \sum_{K=0}^{L-1} x[k]z^{-k}$$

where z is a complex number. $X(z)$ is a polynomial of degree $L-1$ in the variable z^{-1}.

9.2 Basic z-Transform Properties

9.2.1 Linearity Property of the z-Transform

The z-transform is a linear transformation because it satisfies the superposition property. This means that if we take the z-transform of a linear combination of signals in the n-domain, then we can take the z-transform of all the signals and then perform the linear combination on the result.

9.2.2 Time-Delay Property of the z-Transform

The z-transform of the signal $y[x] = x[n-1]$ (unit delay) is simply z^{-1}. If we multiple the z-polynomial by this factor, we thus delay the signal by 1. More generally, if we want to delay a signal by n_0, we multiply the z-polynomial by z^{-n_0}.

9.2.3 A General z-Transform formula

The general z-transform formula is defined as

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

In some cases this some will not converge, and in some cases it will. The region of convergence is not discussed further in this course/book.

9.3 The z-Transform and Linear Systems

9.3.1 Unit-Delay System

The unit-delay FIR filter is defined by

$$y[n] = x[n-1]$$

If we transform this to the z-domain, we get

$$Y(z) = (z^{-1})X(z)$$

which is the polynomial product of two z-transforms.

9.3.3 The z-Transform of an FIR Filter

If we want to take the z-transform of a FIR filter, we can start with the difference equation and then give the z-transform of that by taking the z-transform of both sides of the difference equation.

9.3.4 z-Transform of the Impulse Response

We can also take the z-transform of a FIR filter by taking the z-transform of the impulse response. This operation is defined simply as

$$H(z) = \sum_{k=0}^{M} h[k]z^{-k}$$

9.3.5 Roots of a z-Transform Polynomial

The z-transform yields a polynomial. This means we can find the roots of a polynomial that represents a signal. When $H(z)$ is expressed in terms of its roots then it can be written as the product of first-order factors that completely define the polynomial to within a multiplicative constant, that is,

$$H(z) = G \prod_{k=1}^{M} (1 - z_k z^{-1}) = G \prod_{k=1}^{M} \frac{z - z_k}{z}$$

where G is a constant. The first-degree term $(z - z_k)$ makes it clear that these roots are also zeros of $H(z)$.

9.4 Convolution and the z-Transform

The convolution property of the z-transform states that polynomial multiplication of z-transforms is equivalent to time-domain convolution. The formal property states

$$y[n] = h[n] * x[n] \quad \overset{z}{\longleftrightarrow} \quad Y(z) = H(z)X(z)$$

9.4.1 Cascading Systems

We have seen that in the time-domain, cascading systems is done by convolution. Naturally thus follows that in the z-domain, this is done by polynomial multiplication.

9.4.2 Factoring z-Polynomials

If we can multiple z-transforms to get higher order systems, then we can also factor z-transform polynomials to break down a large system into smaller modules. The factors of a high-order polynomial $H(z)$ represent components that make up $H(z)$ in a cascade connection.

9.5 Relationship Between the z-Domain and the $\hat{\omega}$-Domain

If we wish to go from the z-domain to the $\hat{\omega}$-domain, we have to substitute z for the complex exponential $e^{j\hat{\omega}}$. The other way around is also possible, by doing the reverse substitution.

9.5.1 The z-Plane and the Unit Circle

The z-transform is a complex-valued function of the complex variable $z = x + jy$. A plot of the magnitude $|X(z)|$ would be a three-dimensional plot of magnitude versus x and y. The values of $z = e^{j\hat{\omega}}$ define a circle of radius one in the z-plane, and the DTFT is the z-transform evaluated on the unit circle.

9.6 The Zeros and Poles of $H(z)$

The roots of the z-polynomials play a key role in simplifying the use of $H(z)$ for problem solving. The product form of $H(z)$ show that it is a product of first order factors $1 - z_k z^{-1}$. If we rewrite such a sum into a fraction, we get a certain power of z in the denominator. The power n of this z represents a n-order pole at $z = 0$.

9.6.1 Pole-Zero Plot

A pole-zero plot is a graphical method to represent poles and zeros of $H(z)$. Each zero location is denoted by a small circle, and the poles at $z = 0$ are indicated with a single cross with a numeral beside it indicating its order.

9.6.2 Significance of the Zeros of $H(z)$

If we know at which values of z the function $H(z) = 0$, then we can calculate the frequencies at which the filter with this system function will null that frequency. This is only the case for zeros that lie on the unit circle. The angle of the complex exponential that is z represents the frequency that it nulls.

9.6.3 Nulling Filters

We can also specifically make a filter that nulls a certain frequency. If we wish to null a frequency $\hat{\omega}_0$ then we need two filters, one to null $e^{j\hat{\omega}_0}$ and one to null $e^{-j\hat{\omega}_0}$. We take the system functions

$$H_1(z) = 1 - z_1 z^{-1} \qquad H_2(z) = 1 - z_2 z^{-1}$$

. We then multiply this together to get a system function to null the frequency $\hat{\omega}_0$.

9.7 Simple Filters

9.7.1 Generalize the L-point Running-Sum Filter

The general form of the system function for the L-point running-sum filter is

$$H(z) = \sum_{k=0}^{L-1} a^k z^{-k} = \frac{1 - a^L z^{-L}}{1 - az^{-1}} = \frac{z^L - a^L}{z^{L-1}(z-a)}$$

From this we see that it has roots $z = ae^{j2\pi k/L}$.

Sadly, the rest of this chapter remains unfinished in this summary.