Chapter 3

Signal Processing Review

This chapter reviews continuous-time and discrete-time signal processing. If you haven't had these courses yet, you may wish to review a good signal processing textbook such as those by Munson & Kudeki, Oppenheim & Schafer, or Proakis & Menaloakis.

Continuous-time and discrete-time signals will be distinguished by the type of parentheses: $x(t)$ for continuous time, $x[n]$ for discrete time. ω is the frequency (in radians/sample) of a discrete-time signal; Ω is the frequency (in radians/second) of a continuous-time signal. Transforms are denoted by capital letters (e.g., $X(\omega)$), but phasors are denoted by lower-case boldface letters (e.g., $\mathbf{x}$).

Many properties of linear, time-invariant systems are true in both continuous time and discrete time. These properties will usually be presented in discrete time; the continuous time analog is inferred.

3.1 LTI Systems

A "signal" is any measurable function of time, including sound pressure, air particle velocity, electrical voltage or current, etc. A "system" is anything in the world that generates an output signal $y[n]$ when presented with an input signal $x[n]$, thus:

$$x[n] \rightarrow \boxed{h[n]} \rightarrow y[n] \tag{3.1}$$

What is an LTI System?

A system is "linear" if the following condition holds: $x_1[n] \rightarrow y_1[n]$ and $x_2[n] \rightarrow y_2[n]$ implies that $ax_1[n] + bx_2[n] \rightarrow ay_1[n] + by_2[n]$.

A system is "time-invariant" if $x[n] \rightarrow y[n]$ implies that $x[n-m] \rightarrow y[n-m]$.

A system is LTI if it is both linear and time-invariant.

Impulse Response

An LTI system is completely characterized by the impulse response $h[n]$. The output resulting from any input can be derived by convolution:

$$\textbf{(CT)} \quad y(t) = h(t) * x(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau \tag{3.2}$$

$$\textbf{(DT)} \quad y[n] = h[n] * x[n] = \sum_{m=-\infty}^{\infty} x[m]h[n-m] \tag{3.3}$$

$$\tag{3.4}$$

Eigenfunctions

Exponentials and sinusoids are the eigenfunctions of LTI systems, meaning that when the input to the system is an exponential, the output is an exponential with the same frequency:

$$x(t) = e^{at} \to y(t) = H(a)e^{at} \tag{3.5}$$

$$x[n] = a^n \to y[n] = H(a)a^n \tag{3.6}$$

$$x(t) = \cos(\Omega_0 t + \theta_0) \to y(t) = |H(j\Omega_0)| \cos(\Omega_0 t + \theta_0 + \angle H(j\Omega_0)) \tag{3.7}$$

$$x[n] = \cos(\omega_0 n + \theta_0) \to y[n] = |H(e^{j\omega_0})| \cos\left(\omega_0 n + \theta_0 + \angle H(e^{j\omega_0})\right) \tag{3.8}$$

In the equations above, $H(s)$ and $H(z)$ are the eigenvalues of the system at complex frequencies s and z, respectively. One of the remarkable findings of classical mathematics is that the eigenvalues of the system also turn out to be the Laplace transform and Z transform, respectively, of the impulse response.

3.2 Transforms

	Continuous Time ($s = \sigma + j\Omega$)	Discrete Time ($z = e^{-sT}$)
General Transform	Laplace Transform $X(s) = \int_{-\infty}^{\infty} x(t)e^{-st}dt$ $x(t) = \frac{1}{2\pi j}\int_{\sigma-j\infty}^{\sigma+j\infty} X(s)e^{st}ds$	Z Transform $X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$ $x[n] = \frac{1}{2\pi j}\oint_C X(z)z^{n-1}dz$
Evaluate at $s = j\Omega$ $z = e^{j\omega}$	Continuous Time Fourier Transform (CTFT) $X(\Omega) = \int_{-\infty}^{\infty} x(t)e^{-j\Omega t}dt$ $x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\Omega)e^{j\Omega t}d\Omega$	Discrete-Time Fourier Transform (DTFT) $X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$ $x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)e^{j\omega n}d\omega$
Signal is Periodic in Time $F_0 = \frac{1}{T_0} = \frac{1}{N_0 T}$	Fourier Series $X_k = \frac{1}{T_0}\int_0^{T_0} x(t)e^{-j2\pi kt/T_0}dt$ $x(t) = \sum_{k=-\infty}^{\infty} X_k e^{j2\pi kt/T_0}$	Discrete Fourier Series (DFS) $X_k = \frac{1}{N_0}\sum_{n=0}^{N_0-1} x[n]e^{-j2\pi kn/N_0}$ $x[n] = \sum_{k=0}^{N_0-1} X_k e^{j2\pi kn/N_0}$
Transform is Sampled in Frequency $N \geq \text{length}(x[n])$		Discrete Fourier Transform (DFT) $X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}$ $x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j2\pi kn/N}$

Let $u[n]$ be the unit step function. Then

$$\mathcal{Z}\left\{a^n u[n]\right\} = \begin{cases} \frac{1}{1-az^{-1}} & |z| > |a| \\ \text{undefined; diverges} & |z| \le |a| \end{cases} \tag{3.9}$$

$$\mathcal{Z}\left\{-a^n u(-n-1)\right\} = \begin{cases} \text{undefined; diverges} & |z| \ge |a| \\ \frac{1}{1-az^{-1}} & |z| < |a| \end{cases} \tag{3.10}$$

The DTFT is defined iff the Z transform converges on the unit circle ($x[n]$ absolutely summable).

The following transform pairs are worth proving, understanding, and memorizing:

$x(t)$	$X(\Omega)$	$x[n]$	$X(\omega)$
$\delta(t-t_0)$	$e^{-j\Omega t_0}$	$\delta[n-n_0]$	$e^{-j\omega n_0}$
$e^{j\Omega_0 t}$	$2\pi\delta(\Omega-\Omega_0)$	$e^{-j\omega_0 n}$	$2\pi\delta(\omega-\omega_0)$
$\cos(\Omega_0 t+\theta)$	$\pi e^{j\theta}\delta(\Omega-\Omega_0)+\pi e^{-j\theta}\delta(\Omega+\Omega_0)$	$\cos(\omega_0 t+\theta)$	$\pi e^{j\theta}\delta(\omega-\omega_0)+\pi e^{-j\theta}\delta(\omega+\omega_0)$
$u(t)-u(t-T)$	$Te^{-j\Omega T/2}\mathrm{sinc}(\Omega\mathrm{T}/2)$	$u[n]-u[n-N]$	$e^{-j\omega(N-1)/2}\mathrm{dsinc}(\omega\mathrm{N}/2)$
$\left(\frac{\Omega_c}{\pi}\right)\mathrm{sinc}(\Omega_c\mathrm{t})$	$u(\Omega+\Omega_c)-u(\Omega+\Omega_c)$	$\left(\frac{\omega_c}{\pi}\right)\mathrm{sinc}(\omega_c\mathrm{n})$	$u(\omega+\omega_c)-u(\omega+\omega_c)$

In the table above, the sinc and dsinc functions are defined as

$$\mathrm{sinc}(\mathrm{x}) = \frac{\sin \mathrm{x}}{\mathrm{x}} \tag{3.11}$$

$$\mathrm{dsinc}(\mathrm{x}) = \left(\frac{\omega}{2\mathrm{x}}\right)\frac{\sin(\mathrm{x})}{\sin(\omega/2)} \tag{3.12}$$

The dsinc function is only commonly used in the following construction:

$$\mathrm{dsinc}(\omega\mathrm{N}/2) = \left(\frac{1}{\mathrm{N}}\right)\frac{\sin(\omega\mathrm{N}/2)}{\sin(\omega/2)} \tag{3.13}$$

The sinc and dsinc functions have the following properties:

$$1 = \mathrm{sinc}(0) = \mathrm{dsinc}(0) \tag{3.14}$$

$$0 = \mathrm{sinc}(\mathrm{n}\pi) = \mathrm{dsinc}(\mathrm{n}\pi), \quad \mathrm{n} = \ldots, -3, -2, -1, 1, 2, 3, \ldots \tag{3.15}$$

Example 3.2.1: CTFT of a causal rectangular window

Consider the rectangular window

$$w(t) = u(t) - u(t-T) \tag{3.16}$$

where $u(t)$ is the unit step function. The Fourier transform of the rectangular window is given by

$$W(\Omega) = \int_{-\infty}^{\infty} w(t)e^{-j\Omega t}dt \tag{3.17}$$

$$= \int_0^T e^{-j\Omega t}dt \tag{3.18}$$

$$= \frac{1}{-j\Omega}e^{-j\Omega t}\Big|_0^T \tag{3.19}$$

$$= e^{-j\Omega T/2}\frac{e^{j\Omega T/2}-e^{-j\Omega T/2}}{-j\Omega} \tag{3.20}$$

$$= e^{-j\Omega T/2}\frac{-2j\sin(\Omega T/2)}{-j\Omega} \tag{3.21}$$

$$= Te^{-j\Omega T/2}\mathrm{sinc}(\Omega T/2) \tag{3.22}$$

Example 3.2.2: DTFT of a causal rectangular window

Consider the rectangular window

$$w[n] = u[n] - u[n-N] \tag{3.23}$$

Its Fourier transform is given by

$$\begin{aligned}
W(\omega) &= \sum_{n=-\infty}^{\infty} w[n]e^{-j\omega n} && (3.24)\\
&= \sum_{n=0}^{N-1} e^{-j\omega n} && (3.25)\\
&= \frac{1}{1-e^{-j\omega}} - \frac{e^{-j\omega N}}{1-e^{-j\omega}} && (3.26)\\
&= e^{-j\Omega(N-1)/2}\frac{e^{j\omega N/2}-e^{-j\omega N/2}}{e^{j\omega/2}-e^{-j\omega/2}} && (3.27)\\
&= e^{-j\omega(N-1)/2}\frac{-2j\sin(\omega N/2)}{-2j\sin(\omega/2)} && (3.28)\\
&= Ne^{-j\omega(N-1)/2}\mathrm{dsinc}(\omega N/2) && (3.29)
\end{aligned}$$

Example 3.2.3: Inverse Fourier transform of a lowpass filter

Consider the ideal lowpass filter, with continuous-time Fourier transform

$$H(\Omega) = \begin{cases} 1 & |\Omega| < \Omega_c \\ 0 & \text{otherwise} \end{cases} \tag{3.30}$$

and with discrete-time Fourier transform

$$H(\omega) = \begin{cases} 1 & |\omega| < \omega_c \\ 0 & \text{otherwise} \end{cases} \tag{3.31}$$

The inverse Fourier transforms are computed the same way in both continuous-time and discrete-time. Here are the computations for the discrete-time case:

$$\begin{aligned}
h(t) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} H(\omega)e^{j\omega n}d\omega && (3.32)\\
&= \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{j\omega n}d\omega && (3.33)\\
&= \frac{1}{j2\pi n}e^{j\omega n}\Big|_{\omega=-\omega_c}^{\omega_c} && (3.34)\\
&= \frac{e^{j\omega_c n}-e^{-j\omega_c n}}{j2\pi n} && (3.35)\\
&= \frac{-2j\sin(\omega_c n)}{j2\pi n} && (3.36)\\
&= \frac{\omega_c}{2\pi}\mathrm{sinc}(\omega_c \mathrm{n}) && (3.37)
\end{aligned}$$

3.3 The Dirac Delta

The Fourier transform of signal $x(t)$ is defined if the following conditions are true. These conditions are called the "Dirichlet conditions."

- $x(t)$ has a finite number of discontinuities.

- $x(t)$ is absolutely integrable ($x[n]$ is absolutely summable):

$$\int_{-\infty}^{\infty} |x(t)|dt < \infty, \qquad \sum_{n=-\infty}^{\infty} |x[n]| < \infty \tag{3.38}$$

Here's a surprise: the function $x[n] = \cos(n)$ does NOT satisfy the Dirichlet conditions. It's not absolutely integrable: $\sum_{-\infty}^{\infty} |\cos(n)| \rightarrow \infty$. That means that, technically, it doesn't have a Fourier transform.

In order to create meaningful Fourier transforms of cosines and sines, we need to define the Dirac Delta function, $\delta(t)$. The complete definitions of $\delta(t)$ and $\delta[n]$ are:

$$\text{For any } f(t), t_0 : \quad \int_a^b \delta(t-t_0)f(t)dt = \begin{cases} f(t_0) & a < t_0 < b \\ 0 & t_0 < a \text{ or } b < t_0 \\ \text{undefined} & t_0 = a \text{ or } t_0 = b \end{cases} \tag{3.39}$$

$$\text{For any } f[n], n_0 : \quad \sum_{n=a}^{b} \delta[n-n_0]f[n] = \begin{cases} f[n_0] & a \le n_0 \le b \\ 0 & \text{otherwise} \end{cases} \tag{3.40}$$

Equations 3.39 and 3.40 imply that:

$$\delta(t) = \begin{cases} \infty & t = 0 \\ 0 & t \neq 0 \end{cases}, \qquad \delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}, \tag{3.41}$$

The unit step is defined to be the integral of the Dirac delta:

$$u(t) = \int_{-\infty}^{t} \delta(\tau)d\tau = \begin{cases} 0 & t < 0 \\ 1 & t > 0 \\ \text{undefined} & t = 0 \end{cases} \tag{3.42}$$

$$u[n] = \sum_{m=-\infty}^{n} \delta[m] = \begin{cases} 0 & n < 0 \\ 1 & n \ge 0 \end{cases} \tag{3.43}$$

3.4 Transform Properties

	A=Time, B=Frequency	A=Frequency, B=Time
Real in A $\Leftrightarrow$ Conjugate Symmetric in B	$x[n] = \text{real}$ $\Leftrightarrow$ $X(\omega) = X^*(e^{-j\omega})$	$x[n] = x^*[-n]$ $\Leftrightarrow$ $X(\omega) = \text{real}$
Linearity $N = \max(N_1, N_2)$	$ax_1[n] + bx_2[n] \Leftrightarrow aX_1(\omega) + bX_2(\omega)$	
Convolve in A $\Leftrightarrow$ Multiply in B	$x_1[n] * x_2[n]$ $\Leftrightarrow$ $X_1(\omega)X_2(\omega)$	$x_1[n]x_2[n]$ $\Leftrightarrow$ $\frac{1}{2\pi}X_1(\omega) \circledast X_2(\omega)$
Shift in A $\Leftrightarrow$ Modulation in B	$x[n-m]$ $\Leftrightarrow$ $e^{-j\omega m}X(\omega)$	$e^{j\omega_0 n}x[n]$ $\Leftrightarrow$ $X(e^{j(\omega-\omega_0)})$
Sample in A $\Leftrightarrow$ Periodic in B	$x[n] \times \sum_r \delta[n - rM]$ $\Leftrightarrow$ $X(\omega) \circledast \frac{2\pi}{M}\sum_r \delta\left(\omega - \frac{2\pi r}{M}\right)$	$x[n] * \sum_r \delta(n - rN_0)$ $\Leftrightarrow$ $X(\omega) \times \frac{2\pi}{N_0}\sum_r \delta\left(\omega - \frac{2\pi r}{N_0}\right)$
Differentiate in A $\Leftrightarrow$ Multiply by B	$\frac{\partial x(t)}{\partial t}$ $\Leftrightarrow$ $j\Omega X(\Omega)$	$-jtx(t)$ $\Leftrightarrow$ $\frac{\partial X(\Omega)}{\partial \Omega}$
Differentiate in A $\Leftrightarrow$ Multiply by B	$x[n+1] - x[n-1]$ $\Leftrightarrow$ $2j\sin\omega X(\omega)$	$-jnx[n]$ $\Leftrightarrow$ $\frac{\partial X(\omega)}{\partial \omega}$

Example 3.4.1: CTFT of zero-phase rectangular window

Recall the following transform pair from example 3.2:

$$w_R(t) = u(t) - u(t-T) \leftrightarrow W_R(\Omega) = Te^{-j\Omega T/2}\text{sinc}(\Omega\text{T}/2) \tag{3.44}$$

The "zero-phase" rectangular window is defined as

$$w_0(t) = w_R(t + T/2) = u(t + T/2) - u(t - T/2) \tag{3.45}$$

or, equivalently,

$$w_0(t) = \begin{cases} 1 & |t| < T/2 \\ 0 & \text{otherwise} \end{cases} \tag{3.46}$$

From the time-shift property of the Fourier transform, the CTFT of $w_0(t)$ is shown to be

$$W_0(\Omega) = T\text{sinc}(\Omega\text{T}/2) \tag{3.47}$$

Example 3.4.2: DTFT of zero-phase rectangular window

Recall the following transform pair from example 3.2:

$$w_R[n] = u[n] - u[n-N] \leftrightarrow W_R(\omega) = Ne^{-j\omega(N-1)/2}\text{dsinc}(\omega\text{N}/2) \tag{3.48}$$

If and only if N is an odd number, it is possible to define a "zero-phase" discrete-time rectangular window as

$$w_0[n] = w_R(n + (N-1)/2) = u[n + (N-1)/2] - u[n - (N-1)/2] \tag{3.49}$$

or, equivalently,

$$w_0[n] = \begin{cases} 1 & |n| \leq (N-1)/2 \\ 0 & \text{otherwise} \end{cases} \tag{3.50}$$

From the time-shift property of the Fourier transform, the DTFT of $w_0[n]$ is shown to be

$$W_0(\omega) = N\text{dsinc}(\omega\text{N}/2) \tag{3.51}$$

3.5 Sampling, Periodicity, and Circular Convolution

Sampling in Time = Periodic Repetition in Frequency

$$x(t) = \sum_n x_d[n]\delta(t - nT) \tag{3.52}$$

$$\Leftrightarrow \tag{3.53}$$

$$X(\Omega) = X(j(\Omega + 2\pi F_s)) \tag{3.54}$$

Sampling frequency $F_s = 1/T$. Choose $F_s \geq 2F_{max}$. $2F_{max}$ is called the Nyquist frequency.

$$x(t) \rightarrow \boxed{\text{LPF at } F_{max}} \rightarrow \boxed{x_d[n] = x(nT)} \rightarrow x_d[n] \tag{3.55}$$

$$x_d[n] \rightarrow \boxed{x(t) = \textstyle\sum_n x_d[n]\delta(t - nT)} \rightarrow \boxed{\text{LPF at } 2\pi F_{max}} \rightarrow x(t) \tag{3.56}$$

Periodic Repetition in Time = Sampling in Frequency

$$x(t) = x(t + T_0) \tag{3.57}$$

$$\Leftrightarrow \tag{3.58}$$

$$X(\Omega) = \sum_k C_k\delta(j\Omega - jk\Omega_0) \tag{3.59}$$

Fundamental frequency $\Omega_0 = 2\pi/T_0$.

The DTFT is always periodic, regardless of the characteristics of $x[n]$ or of the problem setup: $X(e^{j(\omega+2\pi)}) = X(\omega)$.

The DFT is just the sampled DTFT of a finite-length signal. For this reason the DFT is also always periodic: $X[k+N] = X[k]$.

Notice that the DFT has exactly the same form as the discrete Fourier series. In fact, the DFT is equivalent to the following process: 1) repeat $x[n]$ periodically with period N, 2) compute the DFS of this new periodic signal, 3) multiply by N.

Multiplication of the Fourier series, discrete Fourier series, or DFT implies circular convolution of the two periodic time-domain signals:

$$Z_k = X_k Y_k \Leftrightarrow z[n] = x[n] \circledast y[n] \tag{3.60}$$

Circular convolution is similar to regular convolution, except that 1) the sum is computed over a single period, instead of being computed for all infinity, and 2) the shifted argument is circularly-shifted (shifted modulo N, written $y[[n-m]]_N$) instead of being linearly shifted. The second requirement is optional if the two arguments are periodic (as in multiplication of DFS coefficients), but required if the two arguments are finite in length (as in multiplication of DFT coefficients):

$$x[n] \circledast y[n] = \sum_{m=0}^{N-1} x[m] y[[n-m]]_N \tag{3.61}$$

Circular convolution and linear convolution result in the same output only if both $x[n]$ and $y[n]$ are sufficiently zero-padded:

$$x_1[n] \circledast x_2[n] = x_1[n] * x_2[n] \quad \text{iff} \quad N \geq \text{length}(x_1[n]) + \text{length}(x_2[n]) - 1 \tag{3.62}$$

3.6 Phasor Notation

The study of spatially distributed acoustic systems (e.g., rooms) often requires us to analyze complicated vectors that depend on both frequency and spatial coordinates. Rather than writing $F(x, y, z, \Omega)$ all the time (the explicit Fourier transform of the signal $f(x, y, z, t)$), we can save some notation by dropping the explicit Ω dependence. Don't be fooled by this trick: the phasor is still a function of Ω, we have only dropped the Ω notation in order to reduce the number of variables in each equation!

It is only possible to write a phasor if the signal $f(x, y, z, t)$ is a sinusoid at frequency Ω. If $f(x, y, z, t)$ is a sinusoid, then the definition of the phasor is

$$f(x, y, z, t) = \Re\left\{\mathbf{f}(x, y, z) e^{j\Omega t}\right\} \tag{3.63}$$

In other words,

$$f(x, y, z, t) = A(x, y, z) \cos(\Omega t + \theta(x, y, z)) \Leftrightarrow \mathbf{f}(x, y, z) = A(x, y, z) e^{j\theta(x,y,z)} \tag{3.64}$$

In order to further simplify notation, position is often denoted using the vector $\vec{r} = [x, y, z]^T$, thus

$$f(\vec{r}, t) = \Re\left\{\mathbf{f}(\vec{r}) e^{j\Omega t}\right\} \tag{3.65}$$

Although $\mathbf{f}(\vec{r})$ is only strictly defined if $f(\vec{r}, t)$ is a sinusoid, phasor analysis is often useful when the signals of interest are more complicated things (like speech or music). Phasor analysis is useful as long as the system of interest is LTI. If the system is LTI, then the system output, $\mathbf{g}(\vec{r}_2)$, in response to phasor input $\mathbf{f}(\vec{r}_1)$, is given by

$$\mathbf{g}(\vec{r}_2) = \mathbf{f}(\vec{r}_1) H(j\Omega, \vec{r}_1, \vec{r}_2) \tag{3.66}$$

where $H(j\Omega, \vec{r}_1, \vec{r}_2)$ is the eigenvalue of the system at frequency Ω operating from position $\vec{r}_1$ to position $\vec{r}_2$. But remember that the eigenvalues of a linear system are also the Laplace transform of its impulse response; therefore the output in response to a signal with arbitrary Fourier transform $F(j\Omega, \vec{r}_1)$ is given by

$$G(j\Omega, \vec{r}_2) = H(j\Omega, \vec{r}_1, \vec{r}_2) F(j\Omega, \vec{r}_1) \tag{3.67}$$

The whole point of phasor notation is to shrink Eq. 3.67 down to Eq. 3.66, so that we don't have to write as much in every step of every derivation.

3.7 Windowing for Filter Design

An easy way to create a discrete-time filter is to window the ideal impulse response of the filter. Suppose that $h_d[n]$ is a desired impulse response, aligned so that it is symmetric around the time $n = M$ (M may be an integer, or a half-integer). For example, the delayed version of an ideal lowpass filter is given by

$$H_d(\omega) = e^{-j\omega M} H_0(\omega) \tag{3.68}$$

$$H_0(\omega) = \begin{cases} 1 & |\omega| \le \omega_c \\ 0 & \omega_c < |\omega| \le \pi \end{cases} \tag{3.69}$$

$$h_d[n] = h_0[n-M] = \frac{\omega_c}{\pi}\mathrm{sinc}(\omega_c(\mathrm{n}-\mathrm{M})) \tag{3.70}$$

Suppose we also take a window centered at time M, and with a length of $N = 2M+1$. For example, the rectangular window is

$$w[n] = w_0[n-m] = \begin{cases} 1 & 0 \le n \le N-1 \\ 0 & \text{otherwise} \end{cases} \tag{3.71}$$

$$W(\omega) = e^{-j\omega M} W_0(\omega) \tag{3.72}$$

$$W_0(\omega) = \frac{\sin(\omega N/2)}{\sin(\omega/2)} \tag{3.73}$$

Then the impulse response of the windowed filter is $h[n] = w[n]h_d[n]$, and its frequency response is

$$H(\omega) = e^{-j\omega M}\left(\frac{1}{2\pi}H_0(\omega) \circledast W_0(\omega)\right) \tag{3.74}$$

For example, an ideal lowpass filter, windowed by a rectangular window, produces the filter

$$h[n] = \begin{cases} \frac{\omega_c}{\pi}\mathrm{sinc}(\omega_c(\mathrm{n}-\mathrm{M})) & 0 \le n \le N-1 \\ 0 & \text{otherwise} \end{cases} \tag{3.75}$$

with a frequency response $H(\omega)$ that has linear phase ($\angle H(\omega) = -\omega M$), and that has a magnitude response kind of like an ideal lowpass filter with ripples.

Provided that $\omega_0 > \omega_c$, an ideal bandpass filter with bandwidth $2\omega_c$ and center frequency ω_0 can be constructed by modulating the lowpass filter, i.e.

$$h_0[n] = \frac{2\omega_c}{\pi}\cos(\omega_0 n)\mathrm{sinc}(\omega_c \mathrm{n}) \tag{3.76}$$

An ideal highpass filter can be constructed by modulating the lowpass filter by $\cos(\pi n) = (-1)^n$, i.e.,

$$h_0[n] = (-1)^n \frac{\omega_c}{\pi}\mathrm{sinc}(\omega_c \mathrm{n}) \tag{3.77}$$

Windowing in time means convolving in frequency. Thus

$$y[n] = x[n]w[n] \tag{3.78}$$

Corresponds to

$$Y(\omega) = \frac{1}{2\pi}X(\omega) \circledast W(\omega) \tag{3.79}$$

In order to avoid "smearing" the spectrum, it is desirable to use a window whose spectrum has a narrow main lobe, and very low amplitude sidelobes. Three common windows are the rectangular window, the triangular window, and the Hanning window.

The rectangular window is defined as:

$$w_R[n] = u[n] - u[n-N] \tag{3.80}$$

where $u[n]$ is the unit step. The transform of $w_R[n]$ is

$$W_R(\omega) = Ne^{-j\omega(N-1)/2}\mathrm{dsinc}(\omega N/2) \tag{3.81}$$

where

$$\mathrm{dsinc}(\omega N/2) \equiv \frac{\sin(\omega N/2)}{N \sin \omega/2} \tag{3.82}$$

The triangular window of length N is proportional to the convolution of two rectangular windows of length $N/2$. The triangular window is

$$w_T[n] = \frac{1}{N}(1 - |1 - 2n|)w_R[n] \tag{3.83}$$

The Hanning window of length N is

$$w_H[n] = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi n}{N}\right)\right)w_R[n] \tag{3.84}$$

Both the triangular window and the Hanning window are more selective in time than the rectangular window (meaning that they emphasize the part of the frame near the window center, and de-emphasize other parts). Both are less selective in frequency than the rectangular window (meaning that the main lobe is about twice as wide), but have less long-distance smearing (meaning that the sidelobes are lower).

3.8 Short-Time Fourier Transform (STFT)

The short-time Fourier transform (STFT) is a way of taking a "snapshot" of the local frequency content of some signal, $x[n]$, in the vicinity of time $n = m$. The STFT is defined to be the Fourier transform of the "local signal" $x_m[n]$

$$x_m[n] = x[n+m]w[n] \tag{3.85}$$

or

$$x_m[n-m] = x[n]w[n-m] \tag{3.86}$$

where $w[n]$ is some window that is non-zero only for a finite range, usually $0 \le n \le N-1$.

The short-time Fourier transform (STFT) is the transform of $x_m[n]$:

$$X_m(e^{j\psi}) = \sum_{n=-\infty}^{\infty} x_m[n]e^{-j\psi n} = \sum_{n=0}^{N-1} x_m[n]e^{-j\psi n} \tag{3.87}$$

Using the variable substitution $n \to (n-m)$, Eq. 3.87 can also be written as

$$X_m(e^{j\psi}) = \sum_{n=m}^{m+N-1} x[n]w[n-m]e^{-j\psi(n-m)} \tag{3.88}$$

Here is an interesting property: the STFT can also be written as a sub-band filtered signal. Eq. 3.88 is the same as

$$X_m(e^{j\psi}) = \sum_{n=m}^{m+N-1} x[n]h_\psi[n-m] \tag{3.89}$$

where the filter is

$$h_\psi[n] = w[n]e^{-j\psi n} \tag{3.90}$$

or

$$H_\psi(\omega) = W(e^{j(\omega+\psi)}) \tag{3.91}$$

Thus the signal $X_m(e^{j\psi})$ can also be thought of as a time-domain signal, with samples m, equal to a sub-band filtered copy of $x[n]$, filtered with center frequency $-\psi$.

The STFT is completely determined by its N frequency samples:

$$X_m[k] = \sum_{n=0}^{N-1} x_m[n] e^{-j2\pi kn/N} \tag{3.92}$$

The STFT is conjugate symmetric, i.e. $X_m(\omega) = X_m^*(e^{-j\omega})$.

Notice that the windowed signals satisfy

$$\sum_{m=-\infty}^{\infty} x_m[n-m] = x[n] \sum_{m=-\infty}^{\infty} w[n-m] \tag{3.93}$$

Therefore the inverse STFT works as follows: use an inverse Fourier transform to compute the local signals $x_m[n]$. Then add them up, and normalize:

$$x[n] = \frac{\sum_{m=-\infty}^{\infty} x_m[n-m]}{\sum_{m=-\infty}^{\infty} w[n-m]} \tag{3.94}$$

The STFT is usually not computed for every sample; instead, it is usually only computed once every M samples, for some downsampling factor M. The downsampling factor M is also sometimes called the "frame skip" parameter, or the "frame spacing." If the window length is N samples, then $N - M$ is the "overlap" between adjacent frames. With a frame skip parameter of M, the STFT of frame number f is given by

$$X_{fM}(e^{j\psi}) = \sum_{n=0}^{N-1} x_{fM}[n] e^{-j\psi n} = \sum_{n=fM}^{fM+N-1} x[n] w[n-fM] e^{-j\psi(n-fM)} \tag{3.95}$$

and the inverse transform is

$$x[n] = \frac{\sum_{f=-\infty}^{\infty} x_{fM}[n-fM]}{\sum_{f=-\infty}^{\infty} w[n-fM]} \tag{3.96}$$

Notice that the denominator in Eq. 3.96 doesn't depend on the signal: it only depends on the window, and the frame spacing. It is almost always a good idea to choose the window and M so that

$$\sum_{f=-\infty}^{\infty} w[n-fM] = 1 \tag{3.97}$$

Eq. 3.97 is called the "overlap-add" condition. Under this condition, the inverse STFT is computed by just overlapping the local signals, and adding them together. Examples of windows that work include the Hanning window (with $M = N/2$), the rectangular window (with N equal to any integer multiple of M), and the triangular window.

3.9 Room Response Measurement

The foundation of many audio engineering algorithms is a linear, time-invariant model of a speaker-room-microphone system. A speaker is an RLC circuit driving a spring-mass mechanical system, thus it is "mostly" linear. Rooms filter a signal by adding echoes to it; the addition of echoes to a signal is a linear time-invariant process. A microphone is a mechanical system driving an RLC system, thus a microphone is also "mostly" linear. We therefore have the following model:

$$x(t) \rightarrow \boxed{H(\Omega)} \rightarrow y(t) \tag{3.98}$$

The first step in many audio engineering applications, therefore, is to measure $H(\Omega)$. Three approaches are commonly used: tone response, impulse response, and pseudo-noise response.

Tone Response Method

Play a cosine wave through the loudspeaker,

$$x(t) = \cos(\Omega t) \tag{3.99}$$

Then

$$y(t) = |H(\Omega)| \cos(\Omega t + \angle H(\Omega)) \tag{3.100}$$

Repeat this process for frequencies at third-octave intervals, e.g. for the frequencies 32Hz, 40Hz, 51Hz, 64Hz, ...

Problems with this method:

- Lots of measurements: for a seven-octave response, need to measure the response to 22 tones.
- Phase ambiguity. $\cos(\Omega t + \Phi) = \cos(\Omega t + \Phi + 2\pi k)$ for any integer k; thus the phase estimated based on measurements may not be the true system phase.

Impulse Response

Suppose we could find an excitation signal $x(t)$ that is approximately an impulse — say, a hand clap, or a balloon pop, or the blast of a starter pistol. Then

$$y(t) = h(t) * x(t) \approx h(t) * \delta(t) = h(t) \tag{3.101}$$

Thus $y(t)$ is itself $h(t)$, and its transform is $H(\Omega)$.

The problem with this method is that no physically realizable signal is exactly an impulse. The spectrum of an impulse is flat at all frequencies,

$$\mathcal{F}\left\{\delta(t)\right\} = 1 \tag{3.102}$$

The spectrum of a starter pistol, for example, often has a resonance peak at the frequency of the barrel resonance, and a rolloff above about 10kHz possibly caused by the fact that it takes about $50\mu s$ for the gunpowder blast to claim all of the gunpowder.

Pseudo-Noise

It is relatively easy to create a digital signal $v[n]$ of length N with the characteristic that $|V(k)|$, the N-point DFT of $v[n]$, is equal to 1 over the entire Nyquist band. A signal so constructed may be played back periodically through a loudspeaker, i.e.,

$$x(nT) = x[n], \quad x[(n)_N] = v[n] \tag{3.103}$$

Then

$$y[n] = h[n] * x[n] = h[n] \circledast v[n], \quad 0 \le n \le N-1 \tag{3.104}$$

where

$$H(\omega) \approx H(j\Omega T), \quad -\pi < \omega < \pi \tag{3.105}$$

Given $y[n]$, the impulse response of the room may be computed as

$$\begin{aligned} \hat{h}[n] &= y[n] * w[n] \\ &= h[n] * (v[n] \circledast w[n]) \end{aligned}$$

If $v[n] \circledast w[n] = \delta[n]$, then $\hat{h}[n] = h[n]$.

Suppose we use the constraint that $w[n] = v[-n]$. Then $v[n]$ must be chosen so that $r_v[n] = 0$, where

$$r_v[n] = v[n] \circledast v[-n] = \sum_{m=0}^{N-1} v[m]v[(n+m)_N] \tag{3.106}$$

1. **Pseudo-Noise**

   Construct $v[n]$ as a series of linearly independent zero-mean, unit variance random variables. Then

$$E\left(r_v[0]\right) = E\left(\sum_m v^2[m]\right) = 1 \tag{3.107}$$

   Also, because of the linear independence of $v[m]$ and $v[n+m]$, we can write

$$\begin{aligned} E\left(r_v[n]\right) &= E\left(\sum_m v[m]v[n+m]\right), \quad n \neq 0 \\ &= E(v[m])E(v[n+m]) \\ &= E(v)^2 = 0 \end{aligned}$$

   Putting these two equations together, we find that

$$E\left(r_v[n]\right) = \delta[n] \tag{3.108}$$

   Unfortunately, it is only the expected value of $r_v[n]$ that equals an impulse. The autocorrelation of any particular finite-length sequence $v[n]$ is not exactly an impulse; the error will show up as noise in the output measurement.

2. **Maximum-Length Sequence**

   The maximum-length sequence of order n (Schroeder, 1979) is a sequence $v[n]$ of length $N = 2^n - 1$ such that

$$r_v[n] = \begin{cases} 1 & n = 0 \\ -\frac{1}{N} & n \neq 0 \end{cases} \tag{3.109}$$

   Suppose that $v[n]$ is repeated periodically on a loudspeaker. Then

$$y[n] = x[n] * h[n] = v[n] \circledast h[n] \quad 0 \leq n \leq N-1 \tag{3.110}$$

   Construct the sequence

$$q[n] = y[n] * v[-n] \tag{3.111}$$

   then this sequence is equal to

$$q[n] = h[n] \circledast r_v[n] = \sum_{k=-\infty}^{\infty} \hat{h}[n-kN] \tag{3.112}$$

   where $\hat{h}[n] \approx h[n]$. The difference between $\hat{h}[n]$ and $h[n]$ is caused by several things. First, the constant offset term in $r_v[n]$ causes a constant offset term in $\hat{h}[n]$, but this term goes to zero as N gets very large. Second, $\hat{h}[n]$ is aliased. Third, background noise in the room adds error to $\hat{h}[n]$; this error may be reduced by adding together multiple repetitions of $\hat{h}[n]$.

Consider that

$$\begin{aligned} \mathcal{F}\left\{v[-n]\right\} &= \sum_{n=-(N-1)}^{0} v[-n]e^{-j\omega n} \\ &= \sum_{n=0}^{N-1} v[n]e^{j\omega n} \\ &= V(e^{-j\omega}) = V^*(\omega) \end{aligned}$$

Let $R_v(\omega)$ be the Fourier transform of $r_v[n]$. Its value is

$$R_v(\omega) = \mathcal{F}\left\{v[n] \circledast v[-n]\right\} = V(\omega)V^*(\omega) = |V(\omega)|^2 \tag{3.113}$$

But if $r_v[n] \approx \delta[n]$, then $R_v(\omega) \approx 1$, and therefore $|V(\omega)| = 1$.

Schroeder cites this as a specific strong point of the MLS method: it is not necessary to do any post-processing at all in order to get an estimate of $|H(\omega)|$, because it is already given by the measured signal $y[n]$:

$$|Y(\omega)| = |H(\omega)||V(\omega)| = |H(\omega)| \tag{3.114}$$

Note that the same is not true of the phase terms:

$$\angle Y(\omega) = \angle H(\omega) + \angle V(\omega) \tag{3.115}$$

The phase of the maximum likelihood sequence, $\angle V(\omega)$, is uniformly distributed between $-\pi$ and π. It can only be canceled by the post-processing described above:

$$q[n] = y[n] * v[-n], \quad Q(\omega) = Y(\omega)V^*(\omega) \tag{3.116}$$

$$\angle Q(\omega) = \angle Y(\omega) - \angle V(\omega) = \angle H(\omega) \tag{3.117}$$

3.10 Downsampling

We can reduce the amount of information stored on the computer with an algorithm like this:

1. Throw away $M-1$ out of every M samples:

$$x_d[n] = x[n] \times \sum_r \delta(n - rM) = \begin{cases} x[n] & n = \ldots, M, 2M, 3M, \ldots \\ 0 & \text{else} \end{cases} \tag{3.118}$$

2. Change the axis labels:

$$y[n] = x_d[nM] \qquad Y(\omega) = X_d(e^{j\omega/M}) \tag{3.119}$$

The second step doesn't change the information in the signal, but the first step does. Remember that sampling in time equals periodic repetition in frequency, so

$$X_d(\omega) = \frac{1}{2\pi} X(\omega) * \frac{2\pi}{M} \sum_{k=0}^{M-1} \delta\left(\omega - \frac{2\pi k}{M}\right) = \frac{1}{M} \sum_{k=0}^{M-1} X(e^{j(\omega - \frac{2\pi k}{M})}) \tag{3.120}$$

So we get aliasing unless

$$X_d(\omega) = 0 \text{ for } \omega > \frac{\pi}{M} \tag{3.121}$$

This is just like resampling the signal at $F_s(\text{new}) = F_s(\text{old})/M$. The solution is just like an A/D, except that both input and output are digital:

$$x[n] \rightarrow \boxed{\text{LPF at } \pi/M} \rightarrow \boxed{y[n] = x[nM]} \rightarrow y[n] \tag{3.122}$$

3.11 Upsampling

Suppose we want to increase the sampling rate: $F_s(\text{new}) = L \times F_s(\text{old})$. We start by relabeling the time and frequency axes, like so:

$$v_u[n] = \begin{cases} x[n/L] & n = \ldots, -L, 0, L, 2L, \ldots \\ 0 & \text{else} \end{cases}, \qquad V_u(\omega) = X(e^{jL\omega}) \tag{3.123}$$

$V_u(\omega)$ is periodic with period $2\pi/L$. To get rid of the extra copies of the spectrum, so it is periodic with period 2π, we LPF:

$$x[n] \rightarrow \boxed{v_u[n] = x(n/L)} \rightarrow \boxed{\text{LPF at } \pi/L} \rightarrow v[n] \tag{3.124}$$

But remember that the transform of an ideal LPF at frequency π/L is

$$h[n] = \frac{1}{L}\mathrm{sinc}(\pi \mathrm{n}/\mathrm{L}) = \frac{\sin(\pi \mathrm{n}/\mathrm{L})}{\pi \mathrm{n}} \tag{3.125}$$

The maximum value of this filter is $h[0] = 1/L$, which means that if we use this filter we will wind up multiplying the entire signal by $1/L$!! Therefore, we must throw in a scaling factor of L:

$$x[n] \rightarrow \boxed{v_u[n] = x(n/L)} \rightarrow \boxed{\text{LPF at } \pi/L} \rightarrow \boxed{\text{Multiply by } L} \rightarrow v[n] \tag{3.126}$$

The inverse transform of an ideal LPF, multiplied by L, is

$$Lh[n] = \frac{\sin(\pi n/L)}{\pi n/L} \tag{3.127}$$

So if we use the algorithm given above, we get an output signal $v[n]$ which looks like

$$v[n] = \begin{cases} x[n/L] & n = \ldots, -L, 0, L, 2L, \ldots \\ \text{interpolated values} & \text{else} \end{cases} \tag{3.128}$$

3.12 Problems

Problem 3.1

Use the definition of the Fourier transform to prove the continuous-time Fourier transform pairs listed in Sec. 3.2. For the first two transforms, use the definition of the Dirac delta (Eq. 3.39). Use the second transform, plus the linearity property of Fourier transforms, to prove the third. The fourth and fifth transforms can be solved using integration with variable substitution (Eq. 1.29).

Problem 3.2

Use the definition of the DTFT to prove the discrete-time Fourier transform pairs listed in section 3.2. For the first two transforms, use the definition of the delta functions (Eqs. 3.39 and 3.40). Use the second transform, plus the linearity property of Fourier transforms, to prove the third. To prove the fourth transform, use Eq. 1.32. The fifth transforms can be solved using integration with variable substitution (Eq. 1.29).

Problem 3.3

The modulation and windowing properties of the DTFT are

$$\text{Modulation property:} \quad \mathcal{F}\left\{e^{j\omega_0 n}x[n]\right\} = X(e^{j(\omega-\omega_0)}) \tag{3.129}$$

$$\text{Windowing property:} \quad \mathcal{F}\left\{x[n]w[n]\right\} = \frac{1}{2\pi}X(e^{j\omega}) \circledast W(e^{j\omega}) \tag{3.130}$$

a. Use the definition of the DTFT, with appropriate variable substitution, to prove Eq. 3.129.

b. Demonstrate that the modulation property of the DTFT is just a special case of the windowing property.

Problem 3.4

Consider the following non-causal triangular window:

$$w_t[n] = \frac{1}{N}w_r[n] * w_r[-n] \tag{3.131}$$

where

$$w_r[n] = u[n] - u[n-N] \tag{3.132}$$

a. Sketch $w_t[n]$.

b. Use the conjugate symmetry and convolution properties of the Fourier transform to show that

$$W_t(e^{j\omega}) = \frac{1}{N}|W_r(e^{j\omega})|^2 \tag{3.133}$$

Sketch $W_t(e^{j\omega})$.

c. Now consider the following triangular window:

$$w_t[n] = \frac{N-|n|}{N}\left(u[n] - u[n-2N]\right) \tag{3.134}$$

Find $W_t(e^{j\omega})$.

Problem 3.5

Two of the most commonly used DSP windows—the Hanning window and Hamming window—can be written in the following form. In this equation, N must be odd, and B is the filter design parameter: $B = 0.5$ for a Hanning window, and $B = 0.46$ for a Hamming window.

$$w[n] = r[n]\left((1-B) + B\cos(2\pi n/N)\right)$$

where $r[n]$ is the zero-centered rectangular window

$$r[n] = \begin{cases} 1 & |n| \le (N-1)/2 \\ 0 & |n| > (N-1)/2 \end{cases}$$

a. What is the DTFT $R(\omega)$ of $r[n]$? At what frequency is the first null of $R(\omega)$? What is the amplitude of the first sidelobe of $R(\omega)$?

b. Express $W(\omega)$ as the sum of three scaled and frequency-shifted copies of $R(\omega)$. At what frequency is the first null of $W(\omega)$?

c. In terms of B, what is the amplitude of the first sidelobe of $W(\omega)$? Find the value of B which minimizes the amplitude of the first sidelobe, and say what that minimum amplitude turns out to be.

d. Sketch $H(\omega)$, the DTFT of the following digital filter. Label the amplitude, peak frequencies, and the frequencies of one or two zero crossings.

$$h[n] = \cos(\pi n/3)w[n]$$

e. Sketch $G(\omega)$, the DTFT of the following digital filter. Label the amplitude and cutoff frequencies.

$$g[n] = 0.25\text{sinc}(\pi n/4)\cos(\pi n/3)w[n]$$

Problem 3.6

Remember that a filter characteristic $H(e^{j\omega})$ is defined by its magnitude $|H(\omega)|$ and phase $\angle H(e^{j\omega})$. One particularly useful representation of the phase of $H(e^{j\omega})$ is the group delay $\tau_H(\omega)$, defined as

$$\tau_H(\omega) = -\frac{d\angle H(e^{j\omega})}{d\omega} \tag{3.135}$$

In general, a linear phase filter has a constant group delay—and the constant is equal to the amount by which the output is delayed with respect to the input. If $\tau(\omega)$ is not constant, components of the input $x[n]$ at different frequencies will be delayed by different amounts. If the differences are large, the result is a sort of reverberated sound.

a. In order to understand group delay, consider the filter

$$h_1[n] = \delta[n-D]$$

Suppose that

$$y_1[n] = h_1[n] * x[n]$$

Find a simple representation of $y_1[n]$ in terms of $x[n]$.

b. What is the DTFT of this filter, $H_1(e^{j\omega})$? What is the group delay? How is the group delay related to your answer to part a?

c. Consider $h_2[n]$, given by

$$h_2[n] = w[n]\left(\frac{\omega_c}{\pi}\right)\text{sinc}\left(\omega_c(\text{n} - \frac{\text{N}-1}{2})\right) \tag{3.136}$$

where

$$w[n] = u[n] - u[n-N] \tag{3.137}$$

Find the magnitude response, $|H_2(e^{j\omega})|$, and the group delay $\tau_2(\omega)$.

Problem 3.7

The process of voiced speech production (e.g., sung vowels) can be modeled as a linear filter,

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

where $x[n]$ is a periodic excitation signal modeling the volume velocity coming through the singer's vocal folds,

$$x[n] = \sum_{r=-\infty}^{\infty} x_0[n + rN]$$

and $h[n]$ is an infinite-length impulse response modeling the frequency response of the mouth. Suppose it were possible to estimate the signal $x[n]$ in some way; then an approximation of $h[n]$ would be given by

$$\hat{h}[n] = \frac{1}{N}\sum_{k=0}^{N-1} \frac{Y(2\pi k/N)}{X(2\pi k/N)} e^{\frac{j2\pi kn}{N}}$$

where $Y(2\pi kn/N)$ and $X(2\pi kn/N)$ are the N-point DFT of any single period of $x[n]$ and $y[n]$, respectively, and N is the fundamental pitch period of $x[n]$. What is the relationship between $\hat{h}[n]$ and $h[n]$? Justify your answer.

Problem 3.8

Suppose that $x[n]$ is a cosine, given by

$$x[n] = \cos(\omega_0 n)$$

a. Suppose that the STFT $X_m(e^{j\psi})$ is computed using a rectangular window of length N. Find $X_m(e^{j\psi})$.

b. Suppose that the STFT is only computed once per M samples. Find $X_{fM}(e^{j\psi})$, the STFT of frame number f.

Problem 3.9

a. Suppose that the input to a room response is a periodic signal $x[n] = v[(n)_N]$, where the $()_N$ notation means "modulo N," i.e.

$$x[n] = \begin{cases} v[n+N] & -N \le n \le -1 \\ v[n] & 0 \le n \le N-1 \\ v[n-N] & N \le n \le 2N-1 \\ \vdots & \end{cases}$$

$x[n]$ is played through a speaker-room-microphone system with impulse response $h[n]$, so that $y[n] = h[n] * x[n]$ is the linear convolution of $h[n]$ and $x[n]$. Show that $y[n]$ is therefore the circular convolution of $v[n]$ and $h[n]$, repeated periodically with period N. Argue that therefore $Y(k) = X(k)H(k)$, where $Y(k)$ is the N-point DFT of $y[n]$.

b. Define the circular autocorrelation $r_v[n]$ and the estimated impulse response $q[n]$ as in the following equation

$$r_v[n] = \sum_{m=0}^{N-1} v[m]v[(m+n)_N], \quad q[n] = y[n] * v[-n] \tag{3.138}$$

Show that $q[n]$ is a time-aliased periodic repetition of $\hat{h}[n] = r_v[n] \circledast h[n]$, i.e.

$$q[n] = \sum_{k=-\infty}^{\infty} \hat{h}[n-kN], \quad \hat{h}[n] = r_v[n] \circledast h[n]$$

where $\circledast$ denotes circular convolution. Argue that therefore, if $r_v[n] = \delta[n]$, then $q[n]$ is a time-aliased periodic repetition of the true room response $h[n]$.

Problem 3.10

Consider the problem of estimating the phase response $\angle H(j\Omega)$ of a loudspeaker. Assume that the phase response is "mostly linear," meaning that for some small nonlinear term $f(j\Omega)$,

$$\angle H(j\Omega) = -D\Omega + f(j\Omega) \tag{3.139}$$

a. Consider a speaker-microphone system with frequency response $H(j\Omega)$. If the input (the sound played from the speaker) is $x(t) = \cos(\Omega_1 t)$, then the output (recorded at the microphone) is

$$y(t) = |H(j\Omega_1)| \cos\left(\Omega_1(t - \tau(\Omega_1))\right) \tag{3.140}$$

where $\tau(\Omega_1)$ is the phase delay in seconds—that is, $\tau(\Omega_1)$ is the number of seconds by which a cosine wave at frequency Ω_1 gets delayed by the speaker. If the phase is as given in equation 3.139, what is $\tau(\Omega_1)$?

b. Notice that the measured phase of the speaker is ambiguous: in general, it is only possible to measure the principal value of the phase, defined as

$$\Phi_{PV}(\Omega) = (\angle H(j\Omega))_{2\pi} = \angle H(j\Omega) + 2\pi k(\Omega) \quad \text{for some unknown integer } k(\Omega)$$

Using phase unwrapping techniques (e.g. the `unwrap` function in matlab), it is possible to eliminate all phase ambiguity except a frequency-independent but unknown phase shift $2\pi k$, i.e.

$$\hat{\Phi}(\Omega) = \begin{cases} \angle H(j\Omega) + 2\pi k & |H(j\Omega)| > 0 \\ \text{Undefined} & |H(j\Omega)| \approx 0 \end{cases} \tag{3.141}$$

If $|H(0)| > 0$, it is possible to get rid of the $2\pi k$ term by setting $\hat{\Phi}(\Omega) = 0$. Unfortunately, most audio systems have a bandpass characteristic, meaning that $|H(0)| = 0$. Therefore, the best we can do is to create a smooth but possibly incorrect phase estimate $\hat{\Phi}$ as shown in equation 3.141.

Why does phase ambiguity make it difficult to measure the phase delay? Consider what would happen to your formula from part (a) if you used the estimated phase instead of the true phase.

c. Consider the following two delay estimators, sometimes called the phase delay and the group delay:

$$\hat{\tau}_P(\Omega) = -\frac{\hat{\Phi}(\Omega)}{\Omega}$$

$$\hat{\tau}_G(\Omega) = -\frac{\partial\hat{\Phi}(\Omega)}{\partial\Omega}$$

Find the values of both of these estimators, in terms of D, $f(\Omega)$, and the unknown constant k, if the phase is as given in equation 3.139. Discuss the relative merits of these two phase estimators.

Matlab Exercises

Before you begin the matlab exercises, make sure that you understand the connection between matlab and your sound card. Find a PC with a microphone, and record your own voice saying anything you like. Play back the file, and make sure that your voice sounds reasonable. Repeat the same procedure with several different sampling rates: most PC sound cards can record and play at F_s =8kHz, 11.025kHz, 22.05kHz, and 44.1kHz, and many sound cards can use an unlimited variety of sampling rates. IMPORTANT NOTE ABOUT PLOTS: A plot is considered INCORRECT if the abscissa is not labeled. Use the two-argument form of the `plot` function to specify the time or frequency at which each point should be plotted, then use the `xlabel` to specify the units. Acceptable units for the xlabel of a time-domain signal are samples, seconds, and milliseconds. Be careful even when using samples: element number "1" of vector x is usually $x[0]$, not $x[1]$. For example, for part (a) of this problem, you could type

```
plot([0:6],v); xlabel('Time (samples)');
```

Acceptable units for the xlabel of a spectral plot are Hz, kHz, radians/second, cycles, or radians; the words "cycles" and "radians" are shorthand for "cycles/sample" and "radians/sample." For example, for subplot (b) of this problem, you could type

```
plot([0:6]/7,abs(fft(v))); xlabel('Freq (cycles)');
```

Problem 3.11

Audio Playback

Create a 1000Hz tone, 500ms long, using the `cosine` function in matlab. Play the tone back to ensure that you can hear it. Make sure that you are playing it back using the right sampling frequency.

Use the `enframe` function (available on the course website) to break your tone into "local" signals $x_{fM}[n]$, with a frame skip parameter of about 5ms, and a window length of about 10ms. Make sure that your window length N is exactly equal to a power of two.

Use the `fft` function to compute the short-time Fourier transform of your audio signal. Set the FFT length exactly equal to the window length N (this should be the default). Use `fft` to compute another STFT of the signal, but this time, use an FFT length equal to four times N.

Use `subplot` to create a plot with three subplots. In the top plot, show one frame as a function of time (abscissa labeled in milliseconds). In the second plot, show the log magnitude FFT of the same frame, in decibels (computed, for example, using `20*log10(abs(fft(FRAMES)))`), computed using an FFT length equal to the window length. In the third plot, show the FFT of the same frame, but this time, make sure that the FFT length is set equal to four times the window length. Hand in a copy of this plot. Explain the difference between subplots 2 and 3. Explain, also, why the maximum height in plots 2 and 3 is not equal to 0dB.

Use `imagesc` to create an image plot from one of your log-magnitude-STFT matrices. Make sure that the abscissa is labeled in milliseconds, from left to right, and the ordinate is labeled in Hertz, from bottom to top. In order to label the ordinate from bottom to top, you may need to use `flipud`, `fliplr`, or `transform` on the matrix before you plot it. Hand in a copy of this plot. Explain why the plot appears as it does.

Problem 3.12

Wideband and Narrowband Spectrograms

Record your own voice, saying your name (or somebody else's). Play back the signal to make sure that it was recorded correctly.

Use the `enframe` function (available on the course website) to break your tone into "local" signals $x_{fM}[n]$, with a frame skip parameter of about 2ms, and a window length of about 4ms. Use `20*log10(abs(fft(X)))` to compute the log magnitude STFT, in decibels, of your voice.

Create a second log-magnitude STFT with the identical procedure, but this time, use a frame skip parameter of about 15ms, and a window length of about 30ms.

Create a plot with four subplots. In the top plot, show one frame of your voice, as computed with a short window (abscissa labeled in milliseconds) — specifically, you should plot the frame with the largest peak amplitude. In the second plot, show the log-magnitude FFT of the same frame. In the third plot, show the long-window frame with the largest peak amplitude. In the fourth plot, show the FFT of the same frame. Comment on the differences between the second and fourth subfigures.

Use `imagesc` to create image plots from both of your log-magnitude-STFT matrices. Make sure that the abscissa is labeled in milliseconds, from left to right, and the ordinate is labeled in Hertz, from bottom to top. In order to label the ordinate from bottom to top, you may need to use `flipud`, `fliplr`, or `transform` on the matrix before you plot it. Hand in copies of these plots. Explain the differences between them.

Problem 3.13

Measuring the Impulse Response of a Room: Balloon Pop

Record a 30-second audio sequence directly into your PC. During the recording, pop a balloon; the balloon pop should be at least 2-3 meters away from the microphone. After the recording finishes, excise the recorded pop, and the first second after the pop. Plot the waveform, and zoom in on the first 10-100ms. Can you see the direct sound? (Roughly) how many discrete echoes can you see? How many milliseconds elapse before the series of echoes disappears into the background noise? How many decibels above the background noise is the direct sound? From your answers to these two questions, estimate how long it would take the impulse response of the room to decay to -60dB relative to the direct sound.

Call your recorded impulse response $h[n]$. Clip $h[n]$ to N samples in such a way that $h[0]$ is the impulse corresponding to the direct sound, and $h[N-1]$ is the last echo apparently louder than the background noise. Scale $h[n]$ so that $h[0] = 1$. Take any speech, music, or audio clip, and call it $x[n]$. Compute $y[n] = h[n] * x[n]$. Can you hear the difference between $y[n]$ and $x[n]$? If not, why not, do you suppose? Compute $h[n] * y[n]$, $h[n] * h[n] * y[n]$, and so on. How many times must you reverberate the sound before the reverberation starts to become annoying? How do you suppose your answer would change if you were recording in a large concert hall? What if you were recording in a padded recording studio?

Plot $h[n]$ (with the abscissa in milliseconds), and hand in a copy of this plot.

Problem 3.14

Measuring the Impulse Response of a Room: Maximum length sequences

Write a matlab function to create a maximum length sequence $v[n]$ of length $N = 2^n - 1$, where the order n should be one of the arguments of your function. Create an MLS with length $N = 7$. Create a plot with three sub-plots: the first sub-plot showing $v[n]$, the second showing $|V(\omega)|$, the third showing $\angle V(\omega)$. Submit a copy of this plot with your homework.

Compute $r_v[n] = v[n] \circledast v[-n]$. Use the `fliplr` command to create $v[-n]$, and the `conv` command to create the linear convolution $v[n] * v[-n]$, then zero-pad this vector to make its length even, and add together the two halves (time alias!) in order to create $v[n] \circledast v[-n]$. Be careful about the time indices—the first element of a vector may not be the $n = 0$ element. Create a plot with four sub-plots. Plot the linear convolution $v[n] * v[-n]$ in the first plot, the circular convolution $r_v[n]$ in the second plot, $|R_v(\omega)|$ in the third plot, and $\angle R_v(\omega)$ in the fourth plot. Send me a copy of this plot.

Create a maximum-length sequence (MLS) $v[n]$ long enough to last about 500ms when played at some reasonable sampling rate. Create a periodic signal $x[n]$ containing K repetitions of $v[n]$, where K is 5-10. Play $x[n]$. What does it sound like?

Record the MLS on a CD, tape recorder, or digital audio recorder. Play it back from the tape recorder, and record the result on your PC; what you record should be the result of filtering the MLS through the sequential impulse responses of the tape recorder's loudspeaker, the room, and the microphone.

Create the signal $q[n] = y[n] * v[-n]$.

Notice that, because of the arbitrary timing of your recording start, the $n = 0$ sample (the sample at which the first copy of $\hat{h}[n]$ starts) is probably not the first element in the vector $q[n]$. Plot the signal $q[n]$, and zoom in around the first big impulse response; try to figure out exactly which element of the vector should be called the $n = 0$ element, so that

$$q[n] = \sum_{k=0}^{K-1} \hat{h}_k[n - kN]$$

where the estimated impulse responses $\hat{h}_k[n]$ are all almost but not exactly identical. Create a plot with three sub-plots. In the first two sub-plots, show any two of the estimated impulse responses $\hat{h}_k[n]$ and $\hat{h}_j[n]$ for $j \neq k$. In the third sub-plot, show the difference $\hat{h}_j[n] - \hat{h}_k[n]$. Why is the difference not zero?

Create an averaged room response estimate

$$\bar{h}[n] = \frac{1}{K} \sum_{k=0}^{K-1} q[n + kN], \quad 0 \leq n \leq N - 1$$

Create also an average of the unfiltered recording

$$\bar{y}[n] = \frac{1}{K} \sum_{k=0}^{K-1} y[n + kN], \quad 0 \leq n \leq N - 1$$

Create a plot with two sub-plots. Plot $|\bar{H}(f)|$ and $|\bar{Y}(f)|$, with the frequency axes labeled in either Hertz or kHz. Comment on the similarity between these two plots. Comment also on the magnitude response you observe: over what frequencies does the speaker have a constant response? What seems to be the high-frequency cutoff? What seems to be the low-frequency cutoff? Are there significant resonant peaks?

Plot the group delay in milliseconds,

$$\tau_G(f) = -1000 \left. \frac{\partial \angle \bar{H}(j\Omega)}{\partial \Omega} \right|_{\Omega = 2\pi f}$$

where, for some appropriate vectors `Hbar` and `Omega`, the derivative can be estimated as

```
diff(Hbar)./diff(Omega).
```

Label the X axis in Hertz or kHz, and label the Y axis in milliseconds. What is the group delay from one speaker to the other? Is the group delay constant over any range of frequencies? (Notice that you may need to zoom the plot in order to see the regions of interest). How does the group delay response relate to the magnitude response?