Lecture Notes in Speech Production, Speech Coding, and Speech Recognition

Mark Hasegawa-Johnson
University of Illinois at Urbana-Champaign

February 17, 2000

Chapter 1

Basics of Digital Signal Processing

1.1 LTI Systems

$$x(n) \rightarrow \boxed{h(n)} \rightarrow y(n) \tag{1.1}$$

1.1.1 What is an LTI System?

- **Linearity**
  If $x_1(n) \rightarrow y_1(n)$ and $x_2(n) \rightarrow y_2(n)$, then
  $$ax_1(n) + bx_2(n) \rightarrow ay_1(n) + by_2(n) \tag{1.2}$$
- **Time-Invariance**
  $$\text{If} \quad x(n) \rightarrow y(n) \quad \text{then} \quad x(n-m) \rightarrow y(n-m) \tag{1.3}$$

1.1.2 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:

$$\textbf{(CT)} \qquad y(t) = h(t) * x(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau \tag{1.4}$$

$$\textbf{(DT)} \qquad y(n) = h(n) * x(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m) \tag{1.5}$$

$$\tag{1.6}$$

1.1.3 Eigenfunctions

Exponentials and sinusoids are the eigenfunctions of LTI systems, meaning that the following rule holds:

$$\boxed{\text{Input is Sinusoid at } \omega_1 \Rightarrow \text{Output is Sinusoid at } \omega_1 \text{ (different phase, different amplitude)}} \tag{1.7}$$

For example, if $H(s) = A(s)e^{j\Phi(s)}$, then

$$x(t) = e^{-s_1 t} \rightarrow \boxed{h(t)} \rightarrow y(t) = H(s_1)e^{-s_1 t} = A(s_1)e^{-s_1 t + j\Phi(s_1)} \tag{1.8}$$

1.2 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 $X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n}$ $x(n) = \frac{1}{2\pi j}\int_C X(z)z^{n-1}dz$
Evaluate at $s = j2\pi f$ (1.9) $z = e^{j\omega}$ (1.10)	Fourier Transform $X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt$ $x(t) = \int_{-\infty}^{\infty} X(f)e^{j2\pi ft}df$	DTFT $X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}$ $x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n}d\omega$
Signal is Periodic in Time $F_0 = \frac{1}{T_0} = \frac{1}{N_0 T}$ (1.11)	Fourier Series $C_k = \frac{1}{T_0}\int_{T_0} x(t)e^{-j2\pi kF_0 t}dt$ $x(t) = \sum_{k=-\infty}^{\infty} C_k e^{j2\pi kF_0 t}$	DFS $c_k = \frac{1}{N_0}\sum_{n=0}^{N_0-1} x(n)e^{-j2\pi kn/N_0}$ $x(n) = \sum_{k=0}^{N_0-1} c_k e^{j2\pi kn/N_0}$
Transform is Sampled in Frequency $N \geq \text{length}(x(n))$ (1.12)		DFT $X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}$ $x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N}$

1.2.1 Fourier Series: Dirichlet's Conditions

Fourier series is defined if:

- $x(t)$ has a finite number of discontinuities.
- $x(t)$ has a finite number of minima and maxima.
- $x(t)$ is absolutely integrable ($x(n)$ is absolutely summable):

$$\int_{T_0} |x(t)|dt < \infty, \qquad \sum_{n=1}^{N_0-1} |x(n)| < \infty \tag{1.13}$$

1.2.2 Z Transform and DFT: Region of Convergence

$$\mathcal{Z}\{a^n u(n)\} = \begin{cases} \frac{1}{1-az^{-1}} & |z| > |a| \\ \text{undefined; diverges} & |z| \leq |a| \end{cases} \tag{1.14}$$

$$\mathcal{Z}\{-a^n u(-n-1)\} = \begin{cases} \text{undefined; diverges} & |z| \geq |a| \\ \frac{1}{1-az^{-1}} & |z| < |a| \end{cases} \tag{1.15}$$

DFT defined iff Z transform converges on the unit circle ($x(n)$ absolutely summable).

1.3 Transform Properties

	A=Time, B=Frequency	A=Frequency, B=Time
Real in A $\Leftrightarrow$ Conjugate Symmetric in B	$x(n) = \text{real}$ $\Leftrightarrow$ $X(k) = X^*(-k)$	$x(n) = x^*(-n)$ $\Leftrightarrow$ $X(k) = \text{real}$
Linearity $N = \max(N_1, N_2)$ (1.16)	$ax_1(n) + bx_2(n) \Leftrightarrow aX_1(k) + bX_2(k)$	
Convolve in A $\Leftrightarrow$ Multiply in B	$x_1(n) \circledast x_2(n)$ $\Leftrightarrow$ $X_1(k)X_2(k)$	$x_1(n)x_2(n)$ $\Leftrightarrow$ $\frac{1}{N}X_1(k) \circledast X_2(k)$
Shift in A $\Leftrightarrow$ Modulation in B	$x((n-m))_N$ $\Leftrightarrow$ $e^{-\frac{j2\pi km}{N}}X(k)$	$e^{\frac{j2\pi in}{N}}x(n)$ $\Leftrightarrow$ $X((k-i))_N$
Sample in A $\Leftrightarrow$ Periodic in B	$x(n) \times \sum_r \delta(n - rM)$ $\Leftrightarrow$ $X(e^{j\omega}) \circledast \frac{1}{M}\sum_r \delta\left(\omega - \frac{2\pi r}{M}\right)$	$x(n) * \sum_r \delta(n - rN_0)$ $\Leftrightarrow$ $X(e^{j\omega}) \times \frac{1}{N_0}\sum_r \delta\left(\omega - \frac{2\pi r}{N_0}\right)$

$$x_1(n) \circledast x_2(n) = x_1(n) * x_2(n) \quad \text{iff} \quad N \geq \text{length}(x_1(n)) + \text{length}(x_2(n)) - 1 \tag{1.17}$$

1.4 Sampling Theorem

Sampling in Time = Periodic Repetition in Frequency

$$x(t) = \sum_n x_d(n)\delta(t - nT) \tag{1.18}$$

$$\Leftrightarrow \tag{1.19}$$

$$X(f) = X(f + F_s) \tag{1.20}$$

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

$$x(t) \rightarrow \boxed{\text{LPF at } F_{max}} \rightarrow \boxed{x_d(n) = x(nT)} \rightarrow x_d(n) \tag{1.21}$$

$$x_d(n) \rightarrow \boxed{x(nT) = x_d(n)} \rightarrow \boxed{\text{LPF at } F_{max}} \rightarrow x(t) \tag{1.22}$$

Periodic Repetition in Time = Sampling in Frequency

$$
\begin{aligned}
x(t) &= x(t+T_0) && (1.23)\\
&\Leftrightarrow && (1.24)\\
X(f) &= \sum_k C_k \delta(f-kF_0) && (1.25)
\end{aligned}
$$

Fundamental frequency $F_0 = 1/T_0$.

1.5 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:

$$x_d(n) = x(n) \times \sum_r \delta(n - rM) = \begin{cases} x(n) & n = \ldots, M, 2M, 3M, \ldots \\ 0 & \text{else} \end{cases} \tag{1.26}$$

2. Change the axis labels:

$$y(n) = x_d(nM) \qquad Y(e^{j\omega}) = X_d(e^{j\omega/M}) \tag{1.27}$$

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

$$X_d(e^{j\omega}) = X(e^{j\omega}) * \frac{1}{M}\sum_k \delta\left(\omega - \frac{2\pi k}{M}\right) = \frac{1}{M}\sum_k X(e^{j(\omega - \frac{2\pi k}{M})}) \tag{1.28}$$

So we get aliasing unless

$$X_d(e^{j\omega}) = 0 \text{ for } \omega > \frac{\pi}{M} \tag{1.29}$$

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:

$$x(n) \rightarrow \boxed{\text{LPF at } \pi/M} \rightarrow \boxed{y(n) = x(nM)} \rightarrow y(n) \tag{1.30}$$

1.6 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:

$$v_u(n) = \begin{cases} x(n/L) & n = \ldots, -L, 0, L, 2L, \ldots \\ 0 & \text{else} \end{cases}, \qquad V_u(e^{j\omega}) = X(e^{jL\omega}) \tag{1.31}$$

$V_u(e^{j\omega})$ 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:

$$x(n) \rightarrow \boxed{v_u(n) = x(n/L)} \rightarrow \boxed{\text{LPF at } \pi/L} \rightarrow v(n) \tag{1.32}$$

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

$$h(n) = \frac{\sin(\pi n/L)}{\pi n} \tag{1.33}$$

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:

$$x(n) \rightarrow \boxed{v_u(n) = x(n/L)} \rightarrow \boxed{\text{LPF at } \pi/L} \rightarrow \boxed{\text{Multiply by } L} \rightarrow v(n) \tag{1.34}$$

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

$$Lh(n) = \frac{\sin(\pi n/L)}{\pi n/L} \tag{1.35}$$

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

$$v(n) = \begin{cases} x(n/L) & n = \ldots, -L, 0, L, 2L, \ldots \\ \text{interpolated values} & \text{else} \end{cases} \tag{1.36}$$