EE 123: Digital Signal Processing	**Spring 2007**

Lecture 3 — January 23

Lecturer: Prof. Anant Sahai *Scribe: Dominic Antonelli*

3.1 Outline

These notes cover the following topics:

- Eigenvectors and Eigenvalues
- Eigensignals
- Discrete-Time Fourier Transform
- Introduction to Sampling and Interpolation

3.2 Eigenvectors and Eigenvalues

3.2.1 Definition and Motivation

A matrix, A, maps a vector space to another vector space. The eigenvectors of A correspond to directions that are invariant to the transform defined by A, that is, if $\vec{v}$ is an eigenvector of A, then

$$A\vec{v} = \lambda\vec{v} \tag{3.1}$$

λ is called the eigenvalue corresponding to eigenvector $\vec{v}$. Note that if $\vec{v}$ is an eigenvector, then $a\vec{v} : a \neq 0$ is also an eigenvector (with the same corresponding eigenvalue) because

$$A(a\vec{v}) = \lambda(a\vec{v})$$

$$a(A\vec{v}) = a(\lambda\vec{v})$$

$$A\vec{v} = \lambda\vec{v}$$

Note also that the zero vector is never considered to be an eigenvector.

3.2.2 Calculating Eigenvectors/Eigenvalues

From Equation 3.1, we can derive a method for computing eigenvectors and eigenvalues of square matrices. Consider an n-by-n matrix A. First, we subtract $\lambda\vec{v}$ from both sides for Equation 3.1,

$$A\vec{v} - \lambda\vec{v} = 0$$

Then we introduce an identity matrix,

$$A\vec{v} - \lambda I\vec{v} = 0$$

and factor out $\vec{v}$,

$$(A - \lambda I)\vec{v} = 0$$

Now, if $(A - \lambda I)$ were invertible, then we could left-multiply both sides of this equation by the inverse, and get

$$I\vec{v} = \vec{v} = 0$$

However, $\vec{v} = 0$ is not a valid eigenvector, so the only way to have a valid solution to this equation is if $(A - \lambda I)$ is not invertible, which occurs iff $det(A - \lambda I) = 0$. Note that this determinant will be a polynomial in λ of degree n. Thus, it will have at least 1 and at most n distinct zeroes, though some of the zeroes may be complex (and complex zeroes always come in complex conjugate pairs). Note that a repeated root may have only a single distinct eigenvector, or it may have many, though we will ignore this technical detail. For our purposes, each repeat of a root will have a distinct eigenvector.

Once we know that λ_i is an eigenvalue, we can compute corresponding eigenvector(s) by simply solving the linear system of equations defined by

$$(A - \lambda_i I)\vec{v} = 0$$

Thus, the space of eigenvectors (the eigenspace) corresponding to eigenvalue λ_i is given by the nullspace of $(A - \lambda_i I)$. Since $(A - \lambda_i I)$ is not invertible (remember, we chose it such that $det(A - \lambda_i I) = 0$), the null space must have dimension ≥ 1. In fact, the dimension of the null space will equal the number of times that root was repeated, except in cases that we will not consider in this class.

3.3 Eigensignals

3.3.1 Analogy to Eigenvectors

Just as a matrix maps from a vector space to another vector space, a system (such as an LTI system) maps from a signal space to another signal space. If we look at it this way, we can get a concept of an eigensignal, that is, a signal, ϕ, that when presented as the input to a given system, T, produces a signal that is a scaled version of the input signal, $\lambda\phi$. That is, we can think of ϕ as an eigenvector, T as a square matrix, and λ as an eigenvalue (see Figure 3.1). In this case, λ gives us the gain or attenuation of the eigensignal.

3.3.2 Eigensignals of LTI Systems

Now, if T is an LTI system, then the complex exponentials (functions of the form $e^{j\omega n}$) are eigensignals. To prove this, we look at what happens when we put $x[n] = e^{j\omega n}$ into an

Figure 3.1. Diagram of an eigensignal for a system T

arbitrary LTI system with impulse response, $h[n]$:

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

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]e^{j\omega(n-k)}$$

$$y[n] = e^{j\omega n} \sum_{k=-\infty}^{\infty} h[k]e^{-j\omega k}$$

$$y[n] = x[n] \sum_{k=-\infty}^{\infty} h[k]e^{-j\omega k}$$

If the system is BIBO stable, then this sum will converge to a constant in n, thus we have

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

where

$$\lambda = \sum_{k=-\infty}^{\infty} h[k]e^{-j\omega k}$$

is the eigenvalue (or gain) for eigensignal, $x[n] = e^{j\omega n}$. Note that λ depends on the value of ω. We call the function

$$H(e^{j\omega}) = \sum_{k=-\infty}^{\infty} h[k]e^{-j\omega k} \tag{3.2}$$

the frequency response of system T defined by impulse response $h[n]$. This transformation (from $h[n]$ to $H(e^{j\omega})$) is called the Discrete Time Fourier Transform (DTFT). Note that this function is periodic in ω with period 2π (and also sometimes less). Also note that the DTFT of a signal is meaningful:

$$X(e^{j\omega}) = \sum_{k=-\infty}^{\infty} x[k]e^{-j\omega k}$$

3.3.3 Inverting the DTFT

An interesting property of the DTFT is that it can be inverted - that is, the DTFT of a signal completely determines the signal itself (likewise for the impulse response of an LTI system). The inverse DTFT is defined by

$$h[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\omega})e^{j\omega n}d\omega$$

We can verify that this correctly inverts the DTFT by plugging in the RHS of Equation 3.2 for $H(e^{j\omega})$,

$$h[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left(\sum_{k=-\infty}^{\infty} h[k]e^{-j\omega k}\right)e^{j\omega n}d\omega$$

$$h[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left(\sum_{k=-\infty}^{\infty} h[k]e^{j\omega(n-k)}\right)d\omega$$

Assuming this converges, we can flip the order of the summation and integration,

$$h[n] = \frac{1}{2\pi}\left(\sum_{k=-\infty}^{\infty}\int_{-\pi}^{\pi} h[k]e^{j\omega(n-k)}d\omega\right)$$

and factor out $h[k]$,

$$h[n] = \frac{1}{2\pi}\left(\sum_{k=-\infty}^{\infty} h[k]\int_{-\pi}^{\pi} e^{j\omega(n-k)}d\omega\right)$$

Next, we note that since we integrating over one entire period of $e^{j\omega(n-k)}$, the integral is 0 when $n \neq k$. When $n = k$, the integral is

$$\int_{-\pi}^{\pi} d\omega = 2\pi$$

Thus, we have

$$h[n] = \frac{1}{2\pi}\left(\sum_{k=-\infty}^{\infty} h[k]2\pi\delta(n-k)\right)$$

$$h[n] = \sum_{k=-\infty}^{\infty} h[k]\delta(n-k)$$

$$h[n] = h[n]$$

3.4 Sampling

3.4.1 Problem Statement

Suppose there is a continuous-time signal, $x(t)$ being transmitted, and we receive a noisy version, $y(t)$, of this signal. Then, we sample it at some frequency, ω_s, to get a discrete-time signal, $y[n]$. Our goal, in general, is to gain some information about $x(t)$ from the discrete-time samples in $y[n]$. For example, suppose that there is no noise, but that the signal $y(t)$ is a delayed version of $x(t)$, that is

$$y(t) = x(t - d)$$

where d is the delay. We wish to compute an estimate, $\hat{d}$, of d. Suppose further that we know $x(t)$ is the simple unit step function, $u(t)$. Then, in the absence of noise, $y(t)$ will be a delayed step function, $y(t) = u(t - d)$, and the samples we get will be $y[n] = u(n - d)$. Note that if we sample at a rate of 1 sample per second, then we will be able to figure out d to within a range of $1s$, but if we sample at a rate of 1000 samples per second, then we will be able to figure out d to within a range of $1ms$. However, there is something very strange about the unit step function - it consists of signals of infinity frequency. In general, for signals that have no components with frequency higher than some fixed frequency ω_0, sampling at a frequency higher than $\omega_s = 2\omega_0$ is not necessary, but may be useful for saving processing resources.

Another related problem is that of reconstructing $x(t)$ or $y(t)$ from $y[n]$. There are many different methods of doing this - we discuss three of them below.

3.4.2 Methods of interpolation

Constant Interpolation

The quality of estimates of $y(t)$ or $x(t)$ depend on four different factors: frequency content of the transmitted signal, sampling rate, method of interpolation, and noise. Suppose, for example, that we use constant interpolation between samples - that is, if we received a sample, $y[n]$ at time n, we assume that $y(t)$ is equal to that value from time $t = n$ until the time of the next sample. See Figure 3.2 for an example. This requires a trivial amount of processing, but accuracy depends heavily on the sampling rate and frequency content of the transmitted signal - unless the sample frequency is significantly higher than the highest frequency present in the signal, this will result in large error.

Linear Interpolation

Likewise, we can use simple linear interpolation - assume that the signal is a straight line between each pair of adjacent samples. See Figure 3.3 for an example. Again, this requires the sampling frequency to be significantly higher than the highest frequency present in the signal. For example, both of these methods have significant error for the signal shown in

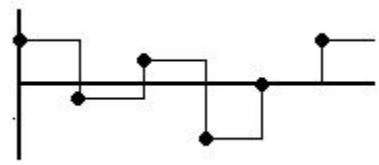

Figure 3.2. Example of constant interpolation. The dots represent samples, and the lines connecting them represent the reconstructed signal

Figure 3.4. Also, both of these methods of interpolation have difficulty dealing with delays - they can give drastically different results for the signal $y(t)$ compared to the results for signal $y(t-d)$ if the delay, d, is not an integer multiple of the sampling rate.

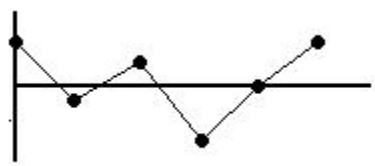

Figure 3.3. Example of linear interpolation. The dots represent samples, and the lines connecting them represent the reconstructed signal

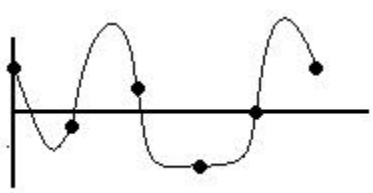

Figure 3.4. Example of a continuous-time signal with the same sample values.

Sinc Interpolation

In order to handle such delays, we can try to use an interpolation method that is LTI (we can think of our interpolator as a system that takes in $y[n]$ and produces an estimate $\hat{y}(t)$ of $y(t)$). Because we want an LTI interpolator, we should consider what happens to the output of our interpolator on input of a complex exponential, $e^{j\omega n}$, since we know that such signals are eigensignals of LTI systems. In the case of a delay by τ, on input of

$$y[n] = e^{j\omega n}$$

the output is

$$y(n-\tau) = e^{j\omega(n-\tau)} = e^{j\omega n}e^{-j\omega\tau} = y[n]e^{-j\omega\tau}$$

Thus, we can see that this has frequency response,

$$H(e^{j\omega}) = e^{-j\omega\tau}$$

and inverting the DTFT yields,

$$h[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{-j\omega\tau}e^{j\omega n}d\omega$$

$$h[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{j\omega(n-\tau)}d\omega$$

$$h[n] = \frac{\sin(\pi(n-\tau))}{\pi(n-\tau)}$$

This function is a form of the normalized *sinc* function, $h[n] = sinc(n-\tau)$. Using this idea, we can, under certain conditions, exactly reconstruct the signal from our samples using the following formula,

$$y(t) = \sum_{n=-\infty}^{\infty} y[n]\frac{\sin[\pi(t-nT)/T]}{\pi(t-nT)/T} = \sum_{n=-\infty}^{\infty} y[n]sinc(\pi(t-nT)/T)$$

where T is the sampling rate. This exactly reconstructs the original signal if the following two conditions are met:

1. The original signal must be bandlimited. That is, the DTFT of the original signal must be 0 for all frequencies $|\omega| > \omega_0$ for some constant ω_0.

2. The sampling frequency, ω_s must be at least $2\omega_0$.

If the second condition does not hold, then aliasing occurs - frequencies above ω_0 are incorrectly reconstructed.