Chapter 3

The DFT and the Sampling Theorem

OBJECTIVE:

1. The principle objectives of this experiment is to introduce the DFT as a signal analysis tool and LTI filters as signal restoration tool. This labs involves designing filters based on analysis, using the DFT and other tools.

2. To study the effects of different sampling rates.
To understand the issues involved in sampling of a continuous-time signal.
To be able to use the filter design tools provided by MATLAB.
To get an introduction to multirate filtering.

Spectrum of a Signal

In chapter 2 of Oppenheim and Schaefer we learned how to express a signal in the frequency domain, using the Fourier Transform for Discrete Signals. The transform takes the following form:

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

This is a transform that must be done on pencil and paper. In class we learned of the Discrete Fourier Transform (DFT) which is similar to the above but can be done on a digital computer. The DFT is defined as:

$$X(k) = \sum_{n=0}^{N-1} x[n]W_N^{nk}$$

for $k = 0..N-1$ where $W_N \triangleq e^{-j\frac{2\pi}{N}}$.

In class we learned some of the differences between the Fourier Transform for Discrete signals and the DFT. We also learned that the Fast Fourier Transform referred to a method of evaluating the DFT that is more efficient.

The Matlab command fft will evaluate the DFT, using a fast algorithm if the length of the signal is a power of 2.

In lab 1 we used this command to investigate whether a system was LTI or not.

Look in the starup file that you copied to your disk in the first experiment. Go to the directory given there and look at the README file. It will tell you the names of several signals, give names of corruptions of these signals and give a short description of each of these. Read this file.

Now take the spectrum of each of the signals and their corruption. Provide **sketches** of these in your write-up (no need to waste printouts on these). Plot the signals in the time domain and provide sketches of these. Describe qualatatively what you see.

Objective Quality Measurement

The corrupted signals are degraded versions of the originals. We would like to quantify how much corruption has taken place. A common measure of degradation is Mean Squared Error(MSE). If we have a signal $x[n]$ and the same signal corrupted in some way called $xcorrupt[n]$, each of length N then the following is the MSE

$$MSE = \frac{1}{N}\sum_{n=1}^{N}(x[n] - xcorrupt[n])^2$$

Write Matlab code to determine the MSE for each of the signal pairs that your read about in the README file.

Give these results in your write-up.

Signal Restoration

Signal Restoration usually refers to enhancing or improving signals that have been degraded in some way.

For each of the two signal pairs given try to recover the signal by designing an LTI filter using Matlab tools as discussed in class. Your LTI filter will take the corrupted signal as input. The output should be as close to the original as possible. Use MSE as a quality measure. State the MSE that you were able to obtain with your filter. Give the details of your thinking in designing the filter. Give the filer itself. Limit your filter to an FIR filter of order 20.

Repeat the last step using an FIR filter of order 50.

Repeat the last step using an IIR filter of order 20.

Repeat the last step using an IIR filter of order 50.

Repeat the last step using any kind of processing you like (not necessarily LTI.) Your only restriction is that your only input is the corrupted signal.

NOTE: the MSE values that you obtained will be used to grade this lab. Be sure to give all your MSE values.

Sampling Issues

Signals bearing information may be available in either analog or discrete forms. An analog signal is one in which both amplitude and time vary continuously in the prescribed intervals, respectively. Speech signals or the electrical activity of the brain (EEG) are examples of these signals. Conversely, a discrete signal takes on digital values of the amplitude and time. Examples of which are computer data and telegraph signals.

A discrete signal can always be generated from an analog signal by the *Analog to Digital* (A/D) conversion. The operations involved in such a conversion are anti-alias filtering, sampling, and quantization as depicted in Figure 3.1.

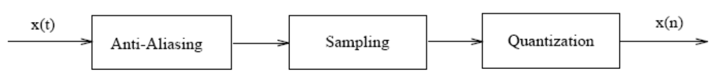

Figure 3.1: Analog to Digital Conversion

The key operation involved in the (A/D) process is sampling and is provided by what is known as the *Sampling Theorem*. It is this idea that we will explore in this section.

In the ideal sampling operation, only exact values of the continuous signal at uniformly spaced discrete intervals (nT_s, where T_s denotes the sampling period) are retained. A sampled signal can be generated by using a method referred to as sampling by modulation as shown in Figure 3.2.

The modulator scheme uses a train of impulses as the carrier, $c(t)$,

$$c(t) = \sum_{n=-\infty}^{\infty} \delta(n - nT_s)$$

and the Fourier Transform (FT) representation of $c(t)$ is

$$c(t) = \sum_{n=-\infty}^{\infty} e^{j\omega_s tn}$$

where $\omega_s = 2\pi/T_s$. The sampled signal, $x_s(t)$ (where the subscript 's' refers to sampled) can now be represented as

$$x_s(t) = x(t) \sum_{n=-\infty}^{\infty} e^{j\omega_s tn}$$

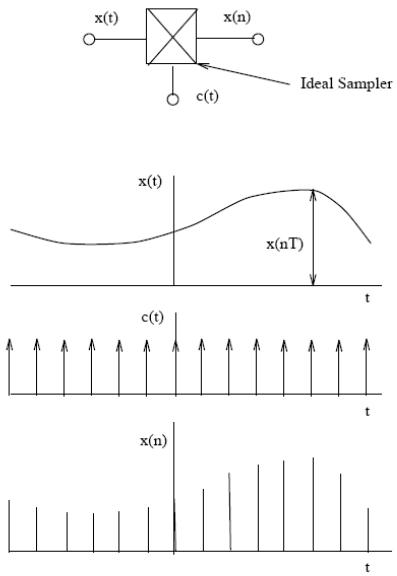

Figure 3.2: Ideal Sampling by Modulation

Taking the the Fourier Transform (FT) of $x_s(t)$, we obtain,

$$\begin{aligned} X_s(\omega) &= \mathcal{F}\left[x(t)\sum_{n=-\infty}^{\infty} e^{j\omega_s tn}\right] \\ &= \sum_{n=-\infty}^{\infty} \mathcal{F}\left[x(t)e^{j\omega_s tn}\right] \\ &= \sum_{n=-\infty}^{\infty} X(\omega - n\omega_s) \end{aligned}$$

The above analysis demonstrates how ideal sampling can be achieved. It also shows that the FT of the sampled signal is periodic with a period ω_s.

Sampling Theorem

In this section, one version of the sampling theorem is discussed, in which aliasing is avoided when all frequency components of a continuous-time signal are bandlimited to ω_{max}. From the FT of the ideally sampled signal $x(t)$, it is evident that as the sampling period T_s is decreased (ω_s increases) all replicas of $X(\omega)$ move farther apart, (see Figure 3.3b). On the other hand, if T_s increases (ω_s decreases) then the replicas of $X(\omega)$ move closer together. As T_s is continually increased, a point will be reached where the replicas will begin to overlap as shown in Figure 3.3c. This overlap of the frequency spectrum is known as aliasing or folding.. The minimum sampling period at which there is no aliasing (equivalently frequency folding) is attained when

$$T_s = \frac{\pi}{\omega_{max}} \qquad \text{or} \qquad \frac{\pi}{2\pi f_{max}} = \frac{1}{2 f_{max}} = f_s$$

where ω_{max} or f_{max} is the highest frequency component in $x(t)$. In the communication literature, ω_{max} or f_{max} is called the **Nyquist frequency** and $\omega_s = 2\omega_{max}$ or $f_s = 2f_{max}$ is called the **Nyquist rate**. Assuming ideal sampling, the lowpass version of the *sampling theorem* can be stated as follows:
Let $x(t)$ denote a continuous-time signal with FT $X(\omega)$ such that

$$X(\omega) = 0 \qquad \text{for} \quad |\omega| \leq \omega_{max} < \infty$$

then $x(t)$ can be expressed exactly in terms of its samples as

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT_s) \left(\frac{\sin(\omega_{max}(t - nT_s))}{\omega_{max}(t - nT_s)} \right)$$

where the sampling period is given by $T_s = \frac{\pi}{\omega_{max}}$ seconds.

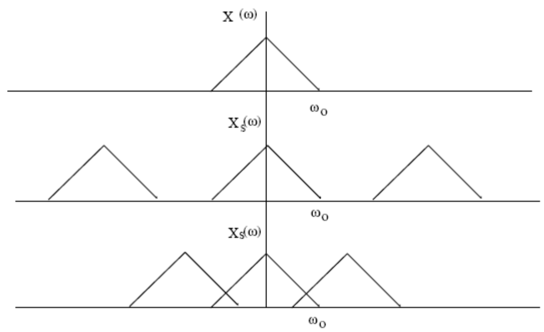

Figure 3.3: Fourier Transform of continuous time and and the sampled signal

Reconstruction

The lowpass version of the sampling theorem provide insight into the way a discrete-time signal can be reconstructed back into a continuous-time form. We note that in the above continuous-time signal representation, the discrete-time samples are convolved with an ideal lowpass filter specified by its *impulse response*

$$h_{lowpass}(nT_s) = \frac{\sin(0.5\omega_s nT_s)}{0.5\omega_s nT_s}$$

In practice the conversion of a sampled signal to a continuous-time form, referred to as the *digital to analog* (D/A) operation, involves two parts, namely a hold operation followed by lowpass filtering.

The most common form of hold operation is the zero-order hold or sample and hold. The impulse response for the sample and hold device is given by

$$h_{hold}(t) = \begin{cases} 1 & 0 \leq t < T_s \\ 0 & \text{otherwise} \end{cases}$$

The operation of the sample and hold is illustrated in Figure 3.4. Note that the lowpass filter operation which follows the sample and hold removes the higher order spectral components of the spectrum which are the translations of the baseband transform.

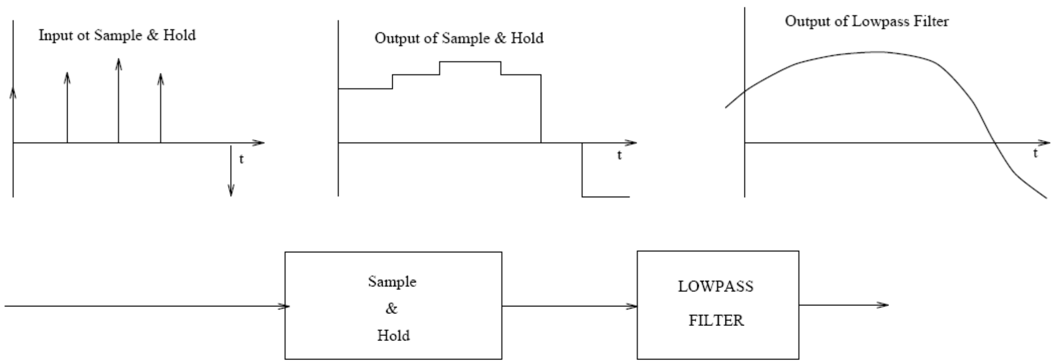

Figure 3.4: Reconstruction of continuous time signal

Problem 1

Consider a sinusoidal signal,

$$x(t) = \sin \Omega_0 t$$

where $\Omega_0 = 20\ \pi\ rad/sec$

1. Using a sampling rate 20 times the **Nyquist rate**, simulate the continuous time signal, $x(t)$. Using the MATLAB **PLOT** routine generate the graph over the time interval $0 \leq t \leq 1$ seconds. Ensure that the graph has proper time scale.

2. Sample $x(t)$ at the **Nyquist rate** and overlay the resulting sequence on the above graph using the MATLAB **STEM** routine. See also **HOLD, AXIS ...**

   The sampled sequence $x[n]$ above can be generated by using the functions exp2_p1, which has been generated for this experiment. For example, given the sampling rate f_s and the length of the time interval, $T_f = 1$ seconds, the sampled sequence $x[n]$ can be generated as $[t, x] =$ exp2_p1(f_s,T_f); where t is a time vector representing the sampling instants and x is the sampled sequence, $x[n]$.

Problem 2

The signal of interest is

$$y(t) = \frac{1}{2}\sin(12\pi t) + \frac{1}{3}\sin(16\pi t) + \frac{1}{5}\sin(22\pi t) + \frac{1}{7}\sin(28\pi t)$$

1. Theoretically calculate the two-sided FT of $y(t)$.

2. Sample $y(t)$ on the interval $0 \leq t \leq 20$ at 20 times the **Nyquist rate**. Evaluate its two-sided spectrum using *discrete Fourier Transform* (DFT). The details of this will be discussed in the future experiments. Here, we are only concerned with using the transform to derive the spectrum.

   The DFT is performed in Matlab by using the routine **fft**. The output of the transform, which we denote by X, is a complex sequence of length N samples representing the phase and magnitude information contained in the spectrum. The important point to note here is that the samples of the DFT spectrum are spaced in normalized frequency by

   $$\omega_k = \frac{2\pi k}{N} \quad \text{for } k = 0, 1, \ldots, N-1$$

   This means that the spectral width (frequency separation) of each DFT sample is

   $$\frac{2\pi}{N} \;\; \texttt{rad} \setminus \texttt{sec}$$

   in normalized frequency, implying that a sinusoid with normalized frequency ω_o will appear in the DFT spectrum at the k^{th} DFT sample if $\omega_k \leq \omega_o < \omega_{k+1}$. **Normalized frequency can be changed to the absolute scale by multiplying by the sampling rate.** Use the MATLAB function named, exp2_p2 that has been written to give the required sequence directly, e.g,
   $[F, Y] = exp2\_p2(fs, T_f)$;
   where Y is the DFT of $y(t)$ sampled at the rate $= f_s$ over the interval $0 \leq t \leq T_f$ and F is the vector containing frequency spacing of the spectrum in Y.

3. Plot the two-sided spectrum evaluated in (2) using the MATLAB **Stem** routine.

4. Repeat 2 and 3 with the following sampling frequencies:

   - 3/2 time the **Nyquist rate**
   - 5/6 times the **Nyquist rate**
   - 13/30 times the **Nyquist rate**

Plot all graphs with same axes.

Q1. Explain the differences in the spectrum using the sampling theorem and the idea of aliasing? Why are the resulting frequency components in the locations that they are in? What are these frequency locations? Use the idea of frequency folding around $f = f_s/2$.
Q2. Given that the sampling rate for $y(t)$ is fixed to be 20 Hz., how would it be possible to ensure that the spectrum of the sampled signal $y[n]$, for the frequencies $0 \leq f \leq 10$ is a true representation. Draw a block diagram of the procedure to meet such an objective.