DSP Introduction

J. Shima

Signal processing is the study of modifying, analyzing, modeling, and representing real-world signals. Specifically, digital signal processing (DSP) is the methodology of converting real (analog) signals into discrete-time representations that are easily stored and manipulated using a computer processor.

DSP includes a large breadth of disciplines within its field, ranging from digital filtering, digital communications, image processing, to neural networks. Every discipline has a common background: Processing discrete-time samples from a real-world signal to obtain a desired result. In the past this was often done with analog processing. But some DSP techniques prove to be not only far superior to analog processing, but some techniques are not even possible with analog circuitry. In these situations DSP begins to demonstrate its usefulness. And with the advent of fast computer processors, DSP only gets more popular. It is used extensively in every cell phone, modem, television, music player, GPS unit, and most anything electronic that uses a real-world signal.

The history of signal processing is peppered with unique discoveries that kick started the field. Newton's use of finite-difference calculus has direct ties to discrete-time differentiation and integration used today. Gauss hand computed the first Fast Fourier Transform (FFT) well before the continuous-time Fourier transform concept was published by Jean Baptiste Fourier. Leonhard Euler's influence, as with everything in mathematics, is indeliably etched into signal processing theory. The best known example is Euler's identity: which is well-used in signal processing and is considered by some to be the most elegant mathematical equation ever.

In the 1950-60s, several important discoveries were made that solidified the importance of DSP. The FFT method was re-introduced by Cooley and Tukey in a seminal paper in 1965. During the Kennedy administration, Tukey attended a talk about how to detect Russian nuclear explosions via spectral analysis. After the meeting, he derived the FFT algorithm and worked with Cooley to implement it on a computer. It has been said that Tukey was reticent to publish the FFT derivation since he did not believe it was any sort of important breakthrough.

In 1960 Bernard Widrow and his student Ted Hoff developed the Widrow-Hoff least-means square (LMS) algorithm, making adaptive signal processing a viable and prospering field in DSP. The LMS method allowed simple implementation of adaptive filters using a computer. One could even implement an iterative version of the prolific Wiener filter using the LMS algorithm.

The use of LMS in neural networks would become historical. Before then, training multi-layer perceptron neural networks was next to impossible. This bottleneck caused a marked decline in neural network research. A surge in neural network interest would have to wait for the discovery of the backpropagation algorithm by Werbos in the 1970s. The backpropagation algorithm was a major breakthrough in neural network training, and it used the LMS algorithm as its foundation. From that point on, the field of neural processing gained substantial foothold and future research exploded.

The crux of digital signal processing theory is the ability to represent, without ambiguity, an analog signal from a discrete set of digital samples. That is, if we take periodic "snapshots" of an analog signal, can we store off these samples on a digital computer and reconstruct the analog signal from these discrete values. This is akin to giving someone a handful of discrete numbers, i.e. [0, 0.9511, 0.5878, -0.5878, -0.9511, -0.0000, 0.9511, 0.5878, -0.5878, -0.9511] and asking them if they are able to tell you the equation of the sinusoid (i.e. sin(wt)) which generated those sample values.

One can imagine that this concept would be quite beneficial for processing data on a computer, which can only store discrete binary numbers. By processing discrete numbers, there is no need to store off the original analog signal. This is the perfect job for a computer processor.

The ability to represent an analog signal with discrete values was discovered by Claude Shannon in 1949. Shannon proved what is now referred to as the ***Nyquist-Shannon sampling theorem***. His work leveraged the initial research by Harry Nyquist in 1928, where Nyquist was studying telegraph communications.

Simply stated, the sampling theorem says: *An analog signal can be unambiguously represented by its digital sample values if the analog signal is sampled greater than twice its analog bandwidth.*

Ok, maybe that wasn't simple! Put another way, if we can take samples (snapshots) of an analog signal at a rate of at least 2x the bandwidth of the analog signal, we can uniquely represent that analog signal using discrete-time digital values. So all we need to store off are some digital values in order to fully characterize an analog signal. Pretty impressive stuff!

As an example, say we have an analog sinusoid at frequency F=100Hz. In order to properly sample this signal, we would have to choose a sampling frequency (Fs) greater than 200Hz. Thus we would be storing off digital samples every T = 5ms. Here is T is the Nyquist rate (1/Fs). From these digital samples we could reconstruct the analog signal without ambiguity.

The sampling theorem is the brick and mortar of all DSP theory. Several important concepts derived from the sampling theorem are key to understanding the mechanism of sampling. These include the following:

1) If we have a band-limited analog signal x(t) with a Fourier transform X(f), then sampling the analog signal every T seconds results in discretized values x[nT] of the original analog signal.

2) The resultant Fourier transform X(theta) of the sampled signal x[nT] contains copies of the analog Fourier transform X(f) spaced every multiple of the sampling frequency. This is typically referred to as the Discrete-Time Fourier Transform.
3) The critical sampling frequency is referred to as the Nyquist frequency, and is typically written as Fs > Fmax/2.
4) If the Nyquist frequency is lower than 2x the analog signal bandwidth, then the resultant digital values no longer uniquely represent the sampled analog signal. This is referred to as "aliasing" in DSP. It refers to the phenomenon where a higher-frequency signal shows up in the digital values mimicking a lower-frequency signal. Thus it is an "alias" of its true frequency. The problem with aliasing is we are unable to discern the true frequency. That is the price paid for violating the Nyquist criteria.

Now, sampling in the real world is typically done by analog-digital converters (ADCs). This mixed-signal device takes in an analog signal, digitizes it at a given rate (Nyquist frequency), and passes the digitized values to a signal processor for further analysis. A sample clock (Fs) is fed to the ADC to tell it when to take a snapshot of the incoming analog signal. The ADC is the heart of the sampling operation.

Figure 1. shows a typical DSP circuit.

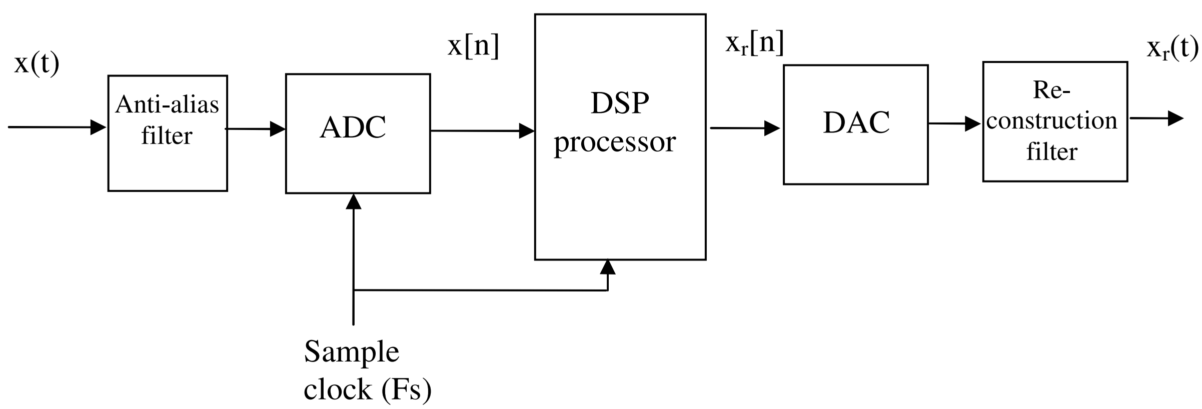

Figure 1.

The anti-alias filter shown in Figure 1. is an analog low-pass filter that band limits the incoming signal x(t) to frequencies below Fs/2. Recall if the input frequencies are above Fs/2, then we will see aliasing in the digital samples. So this analog filter is a necessity, unless you guarantee the incoming signal is already bandlimited (which is not the case in most real-world systems).

Once the ADC digitizes the analog signal, the samples are sent over to a processor for further analysis. If we want to reconstruct the original analog signal, or a modified version of it, we can do so from the digital samples inside the processor. By sending the samples to a digital-analog converter (DAC), each digital value is turned into an analog voltage over each sample period T=1/Fs. This resembles a choppy analog signal with "steps" in it every T seconds. This is due to the fact that the DAC takes in a digital sample every T seconds and outputs a corresponding analog value for it.

After the DAC is an analog reconstruction filter. The reconstruction filter is a low-pass filter that "smooths" out the analog step values from the DAC, recreating the original analog signal. If the processor just passed values from the ADC to the DAC (and we had perfect components), then $x(t) = x_r(t)$

More to come….