1995

Using MATLAB to Illustrate the 'Phenomenon of Aliasing'

Sol Neeman, Ph.D.
Johnson & Wales University - Providence, sneeman@jwu.edu

USING MATLAB TO ILLUSTRATE THE PHENOMENON OF ALIASING

Sol Neeman
Johnson and Wales University

Abstract

The phenomenon of aliasing is important when sampling analog signals. In cases where the signal is bandlimited, one can avoid aliasing by ensuring that the sampling rate is higher than the Nyquist rate. But in cases where the signal is not bandlimited, aliasing is unavoidable if the signal is not filtered before it is sampled. It is then crucial to understand the phenomenon in order to estimate the distortion generated when the signal is reconstructed from its samples. Using the software package MATLAB by MathWorks, Inc., two examples are presented. The first is a pure sinusoid which is sampled at both higher and lower than the Nyquist rate, and the frequency spectrum of both sampled sinusoids are compared to illustrate the effect of aliasing. The second and more interesting case is a square wave which has an unlimited bandwidth. The square wave is a periodic wave that has Fourier expansion with odd harmonics only, the amplitudes of which drop as 1/n. A square wave is synthesized using MATLAB and its Fourier transform is presented graphically (The synthesized square wave inherently produces aliased components). The odd harmonics and the aliased components seen on the graph are analyzed and compared to the predicted theoretical results. Graphs generated by MATLAB accompany the analysis for both signals.

Introduction

Physical signals in nature are continuous, both in time and in amplitude. On digital computers, they can be represented by finite sequences with finite precision. To do this, the signal has to be sampled and the amplitude of the samples be approximated by a finite number of bits (quantization of the signal). Nyquist theorem states that the analog signal can be reconstructed from its samples, provided the sampling rate is greater than twice the highest frequency component in the analog signal. Thus if the highest frequency component is Ω_n (called the Nyquist frequency), the sampling frequency, Ω_s, has to satisfy:

$$\Omega_s > 2 \cdot \Omega_n$$

(the frequency $2 \cdot \Omega_n$ is called the Nyquist sampling rate)

If this condition is not satisfied, aliasing will occur, resulting in distortion in the reconstructed signal from its samples. When signals are not bandlimited, anti aliasing filters can be used to minimize the effect by attenuating the undesired high frequency components. In addition, wideband additive noise may fill in the high frequency range and then be aliased into the lower frequency range. In order to evaluate quantitatively the distortion in the reconstructed signal, it is necessary to understand why the high frequency components are aliased into the lower range.

Why and how Aliasing Occurs

To get an insight into this phenomenon, we analyze the Fourier transform of the sampled signal and compare it to the Fourier transform of the analog (continuous) signal[1]:

Let $x_a(t)$ represent an analog signal with the highest frequency component at Ω_n, and s(t) represent a periodic impulse train, with a period T, that is:

$$s(t) = \sum_n \delta(t - nT)$$

where δ(t) is the unit impulse function. If $x_a(t)$ is sampled at a period T, the sampled signal, $x_s(t)$, can be represented as the product of the two, that is:

$$x_s(t) = x_a(t) \cdot s(t) = x_a(t) \cdot \sum_n \delta(t - nT)$$

Since multiplication in the time domain translates into convolution in the frequency domain, the relationship between the Fourier transforms of the three functions is given by:

$$X_s(j\Omega) = \frac{1}{2\pi} X_a(j\Omega) * S(j\Omega)$$

where Ω represents analog frequency. The Fourier transform of $s(t)$, the periodic impulse train, is a periodic impulse train in the frequency domain, that is:

$$S(j\Omega) = \frac{2\pi}{T}\sum_k \delta(\Omega - \frac{k\cdot 2\pi}{T}) = \frac{2\pi}{T}\sum_k \delta(\Omega - k\Omega_s)$$

Carrying out the convolution in the previous equation, results in:

$$X_s(j\Omega) = \frac{1}{T}\sum_k X_a(j\Omega - kj\Omega_s)$$

This equation states that the Fourier transform of the sampled signal is a superposition of infinitely many shifted copies of $X_a(j\Omega)$, resulting in repeated copies of $X_a(j\Omega)$ in integer multiples of Ω_s, in both direction of the frequency axis. If the sampling rate is not greater than twice the Nyquist frequency, the copies, when superimposed, will overlap and higher frequencies will be aliased into the lower frequency range; therefore we would not be able to reconstruct the original signal from its samples by the use of a filter. On the other hand if the sampling rate is satisfied, no overlapping of the copies will happen and with the appropriate filter it is possible to reconstruct the original signal from its samples.

Fig. 1 represents the Fourier transform of $x_a(t)$ and $x_s(t)$ and the two cases described above.

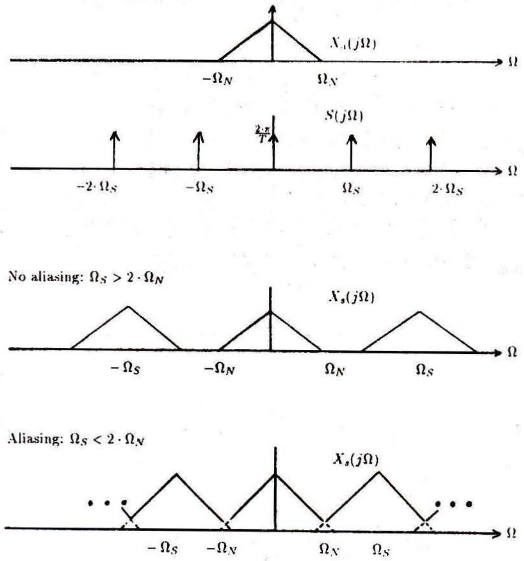

Fig.1 Fourier Transform of an analog signal, $x_a(t)$ and its sampled form, $x_s(t)$.

Aliasing can also be demonstrated in a simpler way, when we analyze the relationship between the sampled versions of a group of pure sinusoids[2]. Consider the continuous sinusoid of frequency w_0 :

$$x(t) = sin(w_0 \cdot t)$$

When sampled at a sampling period T, the resulting sampled sequence is:

$$x(n) = sin(w_0 \cdot n \cdot T) \qquad for \;\; n = 0, \pm1, \pm2, ...$$

Now consider the group of sinusoids of frequency w, where :

$$w = w_0 + \frac{2\pi k}{T} \qquad for \;\; k = 0, \pm1, \pm2, ...$$

When sampled at a sampling period T, any sinusoid:

$$y(t) = sin[(w_0 + \frac{2\pi k}{T}) \cdot t]$$

from this group, will result in a sequence which is identical to the sampled sequence of frequency w_0. To see that, we observe:

$$\begin{aligned} y(n) &= sin[(w_0 + \tfrac{2\pi k}{T}) \cdot nT] \\ &= sin[(w_0 \cdot nT + 2\pi kn)] \\ &= sin[w_0 \cdot nT] \end{aligned}$$

Thus $x(t)$ and any of the sinusoids represented by $y(t)$, cannot be distinguished when sampled at a sampling period T. Note that w_0 is considered in tandem with $-w_0$.

Bin# and Actual Frequency Relationship in FFT Computation

When a sampled signal, $x_s(t)$ is represented by a sequence $x(n)$, the Fourier transform of the sequence, $X(e^{jw})$, is a frequency scaled version of $X_s(j\Omega)$ with the scaling $w = \Omega T$. Thus the analog frequency $\Omega = \Omega_s$ (where $\Omega_s = 2\pi/T$), is normalized to $w = 2\pi$. The FFT of a sequence of length N, produces N points (referred to as bin #'s). In MATLAB, the first $N/2 + 1$ points correspond to the frequency range DC to $f_s/2$ (Nyquist frequency). The rest of the points correspond to negative frequencies (which will be truncated in the following examples).

Thus, to translate the values of the bin #'s to the actual frequency f, we use the equation:

$$f = (bin\# - 1) \cdot f_s/N$$

Graphing FFT Values Versus the Actual Frequency, in Hz[3,4]

To graph the FFT values of a signal versus the actual frequency in Hz rather than versus the bin #, we have to generate a frequency vector of length $N/2+1$, evenly spaced, which represents the frequency range: DC to $f_s/2$. For example, let $X(n)$ represent N points FFT with N even and a sampling frequency f_s; In MATLAB, we can produce a frequency vector Hz via:

$$Hz = (f_s/2) * (0 : N/2)/(N/2);$$

Since $X(n)$ is of length N and we need only the first $N/2+1$ elements, we have to truncate the portion that corresponds to negative frequencies. This is done via:

$$X(N/2+2 : N) = [\,];$$

Then we can plot $|X(n)|$ versus Hz.

Aliasing When Undersampling a pure sinusoid

To illustrate aliasing when a pure sinusoid is undersampled, we use MATLAB to synthesize a sinusoid of frequency $550Hz$, then represent it by two sequences:

1)A sequence corresponding to a sampling frequency of $f_s = 2,000Hz$, thus satisfying the sampling rate in Nyquist theorem.

2)A sequence corresponding to a sampling frequency of $f_s = 1,000Hz$, a sampling rate lower than the Nyquist rate.

Then we compute the FFT of these sequences and plot the results. Clearly, in the second case the frequency of $550Hz$ will be aliased into the range $0 - 500Hz$ (Note that when sampling at a rate of f_s, the FFT values will correspond to the range of frequencies: DC to $f_s/2$). Thus the FFT values for the sequence representing a sampling rate of $1000Hz$ will correspond to the frequency range DC to 500 Hz and the FFT values for the sequence representing a sampling rate of $2000Hz$ will correspond to the frequency range DC to $1000Hz$.

We will define two time indices, $t1$ and $t2$, of 1024 points each and two frequency vectors $Hz1$ and $Hz2$:

$t1$ runs from 0.00 *to* 0.5115 seconds, in steps of $.5ms$, equivalent to a sampling rate of $2,000Hz$.

$t2$ runs from 0.00 *to* 1.023 seconds, in steps of $1ms$, equivalent to a sampling rate of $1,000Hz$.

$Hz1$, the first frequency vector, runs from DC to $1,000Hz$.

$Hz2$, the second frequency vector, runs from DC to $500Hz$.

The next lines are the commands in MATLAB needed to produce both sequences, compute their FFT and plot the results [3]:

```
t1=0:.0005:.5115;
Hz1=(2000/2)*(0:1024/2)/(1024/2);
x1=sin(2*pi*550*t1);
X1=abs(fft(x1));
X1(514:1024)=[ ];
subplot(211);
plot(t1(1:64),x1(1:64));
subplot(212);
plot(Hz1,X1);
(Fig. 2 shows MATLAB plots for x1 and X1)

t2=0:.001:1.023;
Hz2=(1000/2)*(0:1024/2)/(1024/2);
x2=sin(2*pi*550*t2);
X2=abs(fft(x2));
X2(514:1024)=[ ];
subplot(211);
plot(t2(1:256),x2(1:256));
subplot(212);
plot(Hz2,X2);
(Fig. 3 shows MATLAB plots for x2 and X2)
```

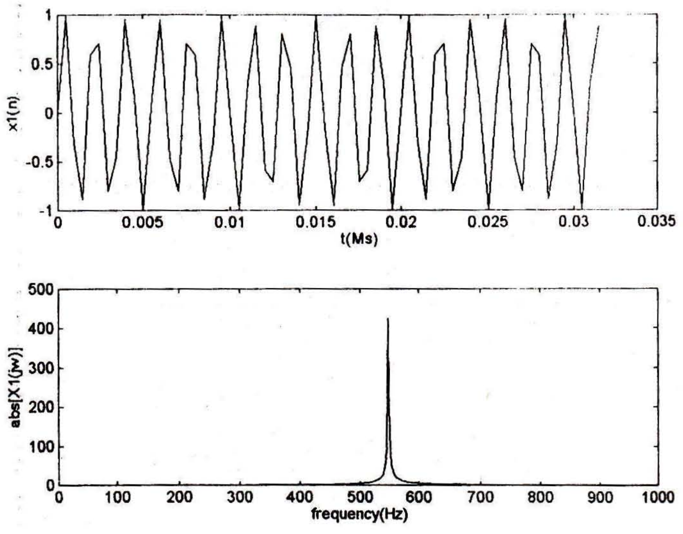

Fig.2 $550Hz$ sine wave sampled at $f_s = 2,000Hz$ and its FFT.

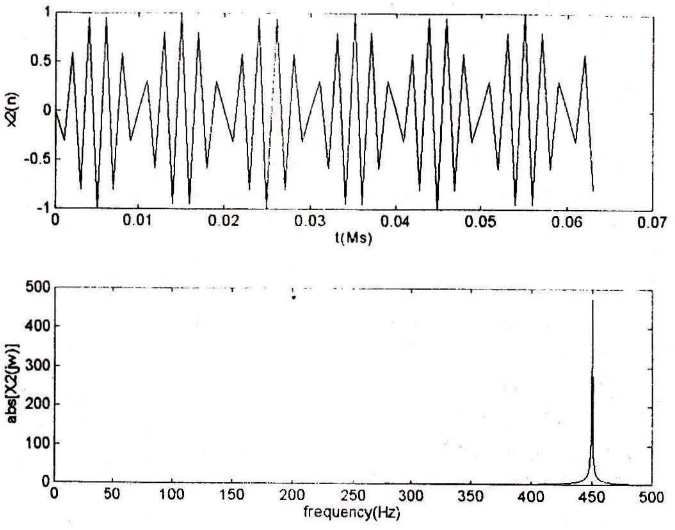

Fig.3 $550Hz$ sine wave sampled at $f_s = 1,000Hz$ and its FFT.

Notice that X1 in the upper graph of Fig. 2 correctly represents the frequency component of $550Hz$. The plot of X2, on the other hand, shows that the frequency of $550Hz$ is aliased into the frequency $450Hz$. This is in accordance with the previous analysis we have since when the spectrum of $x(t)$, the continuous signal, is copied (in integer multiples of $1000Hz$, on both sides of the frequency axis), the frequency $550Hz$ will be aliased into the frequency $450Hz$.

Aliasing When Representing a Square Wave By a Discrete Sequence

As was mentioned before, when an analog signal contains frequency components higher than $\Omega_s/2$, these components will be aliased into the range DC to $\Omega_s/2$. A square wave is an interesting example to illustrate this effect. Consider a periodic square wave with period $T = 2\pi/w_0$ whose Fourier series is given by:

$$X(t) = \frac{4}{\pi}\sum_{k=0}^{\infty}\frac{\sin(2k+1)w_0t}{(2k+1)}$$

The Fourier expansion contains only odd harmonics, the amplitude of which drop as $1/n$ (where n=harmonic #) and is infinite. This means that inherently any representation of a square wave by a discrete sequence will result in aliasing.

To illustrate this using MATLAB, we synthesize a square wave, compute and plot the magnitude of its FFT and compare the results to the predicted aliased components. We will synthesize a $30Hz$ square wave with 50% duty cycle, consisting of 1024 points, sampled at a frequency of $1000Hz$. The following are the commands in MATLAB needed to synthesize a square wave, compute its FFT and plot the results:

```
x=square(2*pi*30*t2,50);
X=abs(fft(x));
X(514:1024)=[ ];
plot(Hz2,X);
```

where t2 and Hz2 are those generated in the previous example and "square" is a command from the Signal Processing Toolbox.

Fig 4 shows the graphs generated by MATLAB.

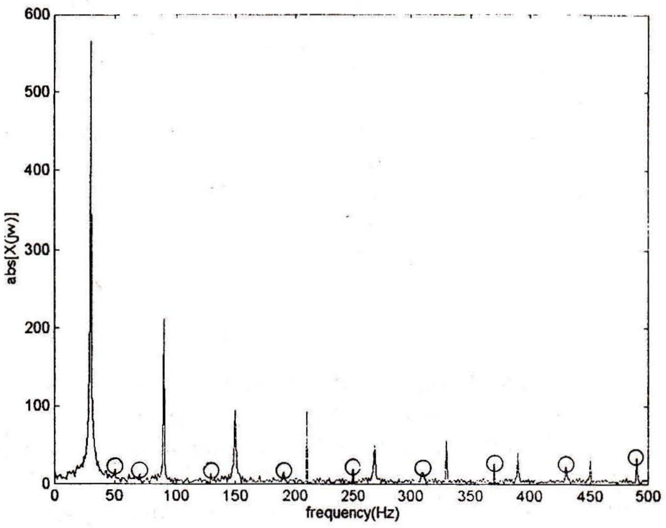

Fig. 4 FFT of a $30Hz$ square wave.

First we will analyze the predicted aliasing effect.

The first harmonic of the square wave is $30Hz$, the 2nd is $90Hz$, the 3rd $150Hz$, etc. Since the sampling rate is $f_s = 1000Hz$, we should expect that all the harmonics above $f_s/2 = 500Hz$ will be aliased into the range DC *to* $500Hz$.

The following is a list of the odd harmonics up to the 35th, the corresponding frequencies and for the frequencies above $500Hz$, the aliased frequency:

Harmonic #	*frequency*	*Expected to be aliased into the frequency*
1	30	
3	90	
5	150	
7	210	
9	270	
11	330	
13	390	
15	450	
17	510	490
19	570	430
21	630	370
23	690	310
25	750	250
27	810	190
29	870	130
31	930	70
33	990	10
35	1050	50
.	.	.
.	.	.

The aliased components are indicated in Fig 4 and they are compatible with the values in the predicted list. Of course there are infinitely many odd harmonics aliased into the range $DC\ to\ 500Hz$. In this example, since their amplitude drops as $1/n$, we can have an estimate of the distortion produced by each component in the reconstructed signal.

Summary

When sampling analog signals below the Nyquist rate, the frequency components above $f_s/2$, where f_s is the sampling frequency, are aliased into the range $DC\ to\ f_s/2$. Illustrating the aliasing effect using synthesized signals can provide better insight and understanding into this phenomenon which can in turn help to estimate or find bounds to the distortion in the reconstructed signal.

References

[1] Alan V. Oppenheim and Ronald W. Schafer, "Discrete-Time Signal Processing", Prentice Hall, 1989, pp.(82-87).

[2] Johnny R. Johnson, "Introduction to Digital Signal Processing", Prentice Hall, 1989, pp.(34-37).

[3] The MathWorks, Inc., "MATLAB Reference Guide", 1992.

[4] The MathWorks, Inc., "Signal Processing Toolbox User's Guide", 1992.