

DSP Basics

DSP BASICS

CSJAXON

DSP Basics Copyright © by csjaxon. All Rights Reserved.

CONTENTS

This Text	1
-----------	---

Part I. Main Body

1. Introduction or Why?	3
2. Sampling and Its Effects	5
3. Z-transform and Processing Sampled Data.	32
4. Appendix A. Laplace and z Transform Properties.	37

Appendix	39
----------	----

This is a textbook that will focus on the most essential topics in Digital Signal Process (DSP). With the intent of focusing on those topics that have proven most helpful through out my career.

It will hopefully allow the students in DSP to build a deep understanding of the more basic topics in DSP. The intent is not to prepare the student for any problem they might encounter, but prepare the student to learn about a new technique when needed.

1.

This text is intended to cover the primary topics in Digital Signal Processing (DSP), Discrete Signal Processing. It is not intended to be complete or comprehensive, rather to focus on those topics that have proven most pertinent in using DSP to collect and analyze data. Although the data processed with these techniques vary, our primary goal is to work on the classic sampled data system as is shown in Figure 1 A).

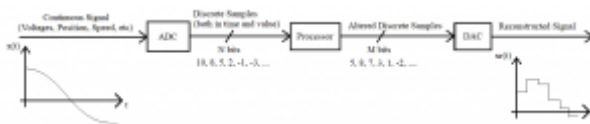


Figure 1.1
Sampling
System
Setup

In the above figure the input is a continuous signal or voltage, which might represent the output of a sensor detecting a position, speed, etc. For the computer to work with this signal it must be converted to digital numbers. This is achieved by the use of an Analog to Digital Converter (ADC), which sends out a stream of N-bit digital number at discrete points in time. This stream of numbers is depicted as a string of values which are discrete both in time and value. Once these samples have been processed, they are sent out as an altered stream of numbers. Note the output may be at a different time and value resolution, M-bit. Finally the output can be changed

into a voltage via a Digital to Analog Converter (DAC). The most notable feature of the reconstructed signal is its blocky nature.

The text will be organized in a fashion to address the various parts of this system, but not in a direct sequence since some background is needed to address some topics. Chapter 2 will be about the basic models for sampling and discuss techniques for characterizing the spectrum of a sampled signal. Chapter 3 progresses to the commonly used techniques to analyze and characterize the processing of the discrete data. Chapter 4 will describe the processing of data, or what is commonly called filtering of data, which will encompass classic filters, but also discuss some nonlinear and intuitive based filters. Chapter 5 will use the filtering techniques developed in Chapter 4 to address the reconstruction of the outputs. Chapter 6 will address the application and computation of the Discrete Fourier Transform (DFT). Chapter 7 will describe the application of adaptive filtering.

As stated previously this is far from a comprehensive text. It should however prepare the student to understand the basics of the field of DSP, be able apply DSP to the most basic systems and learn about any new DSP system they may encounter.

2.

SAMPLING AND ITS EFFECTS

Once people began using computers to analyze data, it became apparent that the data was incomplete, in that we only had the values at set time intervals. This process came to be called sampling. The original work on this was done by two engineers named Nyquist and Shannon and was based on the spectral content of a signal.

Section A) Nyquist and Shannon Sampling Model

The Nyquist Sampling Theorem is based on the following system or model of the sampling process. The analog signal, $x(t)$, that is to be analyzed is multiplied by a stream of impulses.

An impulse is a signal that is defined as

$$\delta(t) = \begin{cases} 0 & \text{for } t \neq 0 \\ \infty & \text{for } t = 0 \end{cases}$$

This is shown as a signal flow diagram here:

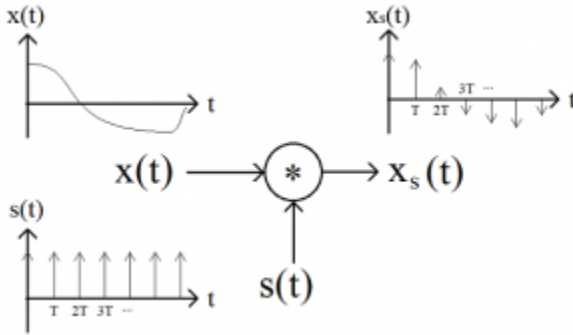


Figure 2.1 Nyquist Sampling Model

The main point of this system is that the output is a stream of δ functions with the area under each being based on the value of $x(t)$. To more completely analyze the effects of sampling, we first write an equation for the sampling function $s(t)$ as

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \quad (2.1)$$

then based on the system above we can see that the output $x_s(t)$ is

$$x_s(t) = x(t) * \sum_{n=-\infty}^{\infty} \delta(t - nT) \quad (2.2)$$

Provided that $x(t)$ is a continuous and properly conditioned function we can move it inside the summation

$$x_s(t) = \sum_{n=-\infty}^{\infty} \{x(t) * \delta(t - nT)\} \quad (2.3)$$

Please note that we will commonly disperse with

mathematical rigor in this text. Not because it is not important, but rather because it has a habit of derailing the reader and inhibits understanding. Similarly since the function $\delta(t - nT)$ is zero, except at $(t - nT) = 0$ or $t = nT$, we can rewrite equation 2.3 as

$$x_s(t) = \sum_{n=-\infty}^{\infty} \{x(nT) * \delta(t - nT)\} \quad (2.4)$$

With the primary result being that $x_s(t)$ is dependent only upon the value of $x(t)$ at the discrete times of (nT) .

Now the question is, “How can we analyze this signal to interpret and predict the effects of sampling?” As is common, we begin by looking at the spectrum or Fourier transform (FT) of the signal. For this we will start with the transform of equation 2.2

$$X_s(f) = \int_{-\infty}^{\infty} \left(x(t) * \sum_{n=-\infty}^{\infty} \{\delta(t - nT)\} \right) e^{j2\pi f t} dt \quad (2.5)$$

We will now bring in a Fourier transform identity that is not commonly used. It is common to use the identity that the product of two FT's is the convolution of two time functions. By the duality of FT's, the product of two time functions is the convolution of their FT's. In equation form,

$$X_s(f_o) = \int_{-\infty}^{\infty} (X(f) * S(f_o - f)) df \quad (2.6)$$

Now the spectrum of the input is given, characterizing the

signal we are analyzing, so the question has to be what is $S(f)$? For this an intuitive description is used in the following video.



One or more interactive elements has been excluded from this version of the text. You can view them online here:

<https://kstatelibraries.pressbooks.pub/dsp-basics/?p=28#oembed-1>

Thus the FT of the $s(t)$ (a stream of time impulses) is a stream of frequency impulses.

$$\begin{aligned} S(f) &= \int_{-\infty}^{\infty} s(t) * e^{j\omega_o t} dt \\ &= \sum_{n=-\infty}^{\infty} f_s \delta(f - f_o - n f_s) \end{aligned} \quad (2.7)$$

Applying this to the convolution described above we have

$$\begin{aligned} X_s(f_o) &= \int_{-\infty}^{\infty} X(f) * \delta(f_o - f + n f_s) df \\ &= \sum_{n=-\infty}^{\infty} X(f_o - n f_s) \end{aligned} \quad (2.8)$$

As described previously δ functions have a sifting property, which means that the only points that survive are when $(f_o - f + n f_s) = 0$ or $f = (f_o + n f_s)$. This sifting also has the effect of changing the integral into a summation at these frequencies are $f_o + n f_s$

$$\begin{aligned} X_s(f) &= \sum_{n=-\infty}^{\infty} X(f_o + n f_s) \\ &= \sum_{n=-\infty}^{\infty} X(f_o + n f_s) \end{aligned} \quad (2.9)$$

Note this also did a change of variable from f_o to f .

The real result of this is that portions of $X_s(f)$ will translate or alias to appear at other frequencies. For example, consider the case of $n = 1$ and note that $X(0) = X_s(-f_s)$. This effect is known as aliasing, since a component of the incoming signal is shifted to another frequency. A more critical affect of this is shown in the following video.



One or more interactive elements has been excluded from this version of the text. You

can view them online here:

<https://kstatelibraries.pressbooks.pub/dsp-basics/?p=28#oembed-2>

One of the more basic things we can find from the aliasing is that as long as the maximum frequency in our incoming signal is less than $\frac{1}{2}$ of the sampling frequency ($f_s = \frac{1}{T}$) the spectrum of the incoming signal can be preserved. In equational form we would have

$$f_{max} \leq \frac{f_s}{2} \quad (2.10)$$

or

$$f_s \geq 2f_{max} \quad (2.11)$$

Thus $2 * f_{max}$ is commonly called the Nyquist sampling

rate. It is easy to remember this since the sampling needs to be at least at the peaks and valleys, or faster.

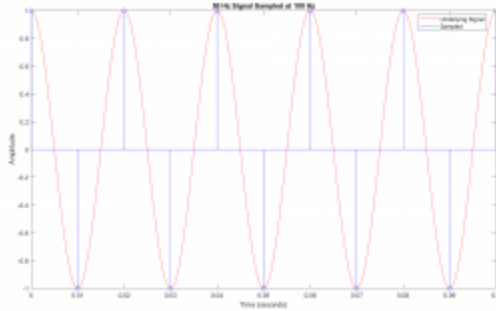


Figure 2.2 Sampling at the Edge of the Nyquist Rate.

Example of Aliasing via MatLab

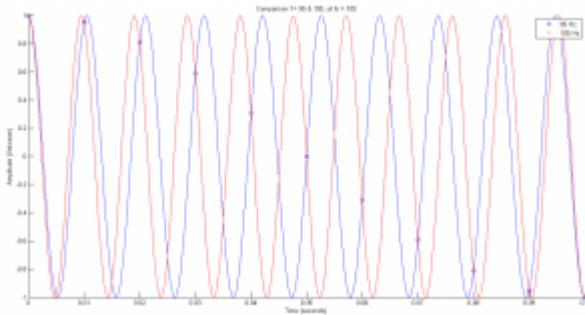
```
% First generate a fine sampling of signals to be sampled.
T = 1/100; % 100 Hz sampling rate.
xF = [0:T/50:0.1]; % Fine sampling rate of 5 KHz (T / 50 = 1/
5000 )
CF1 = cos( 2*pi*95*xF ); % One signal is at 95 Hz
CF2 = cos( 2*pi*105*xF ); % and the second is at 105 Hz.
% plot the fine sampling, showing underlying continuous
signals.
plot( xF, CF1, 'b', xF, CF2, 'r' );
pause();
% Now sample at 100 Hz sampling rate
xC = [0:T:0.1];
```



```

CC1 = cos( 2*pi*95*xC );
CC2 = cos( 2*pi*105*xC );
hold on % plotting on top of continuous signals.
plot( xC, CC1, 'bo', xC, CC2, 'rx' );
% Annotate graph.
xlabel( 'Time (seconds)' );
ylabel( 'Amplitude (Unknown)' );
title( 'Comparison f = 95 & 105, at fs = 100' );
legend( '95 Hz', '105 Hz', 'Samp 95 Hz', 'Samp 105 Hz' );

```



Note how the sampled points, marked with an “o” and “x” align, although the source signal is at a different frequency. This is the basic concept of aliasing in that signals at different frequencies appear the same once sampled.

Section B) Aliasing Diagram

As we observed in the previous section, the affects of sampling can be rather confusing and hard to follow. Thus a process that can help us visualize these affects would be helpful. The Aliasing or Folding Diagram is a graphical technique for estimating the spectral shape of a signal after

sampling. Since it is graphical in nature, it is primarily a tool to help estimate and visualize the affects of frequency aliasing.

Aliasing Diagram Procedure

Based on the sampling interval, the aliasing diagram can be laid out as a folded line of frequencies as shown in Figure 2.3. Here the frequency axis is folded back at $f_s/2$, f_s , $3f_s/2$, ...

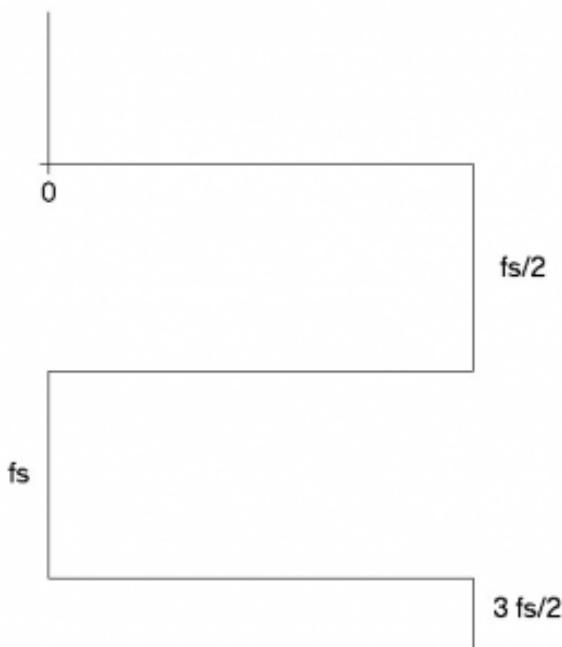


Figure 2.3 Frequency Axis Layout for Aliasing Diagram.

On the diagram that has been laid out, you should sketch the positive portion of the spectrum of the signal to be sampled. Figure 2 shows the diagram with an example positive spectrum sketched in.

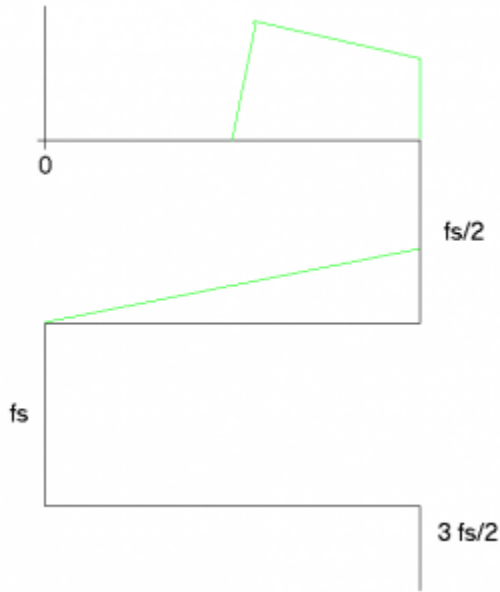


Figure 2.4 Second Step of Drawing in the Positive Spectrum.

Once the spectrum is laid out, a “Resultant Spectrum” is found by adding up all the spectrum’s from all the various folds. The Resultant is shown in red on the 0 to $fs/2$ portion of the frequency axis in Figure 2.5.

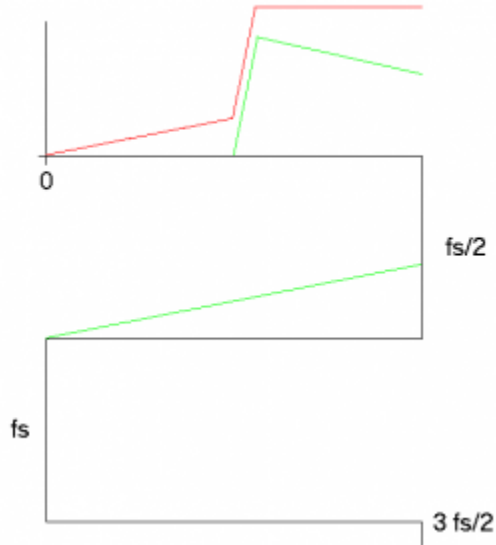


Figure 2.5 Resultant Spectrum Drawn on 0 to $fs/2$.

The Resultant is then copied to all the various folded frequency axis'. The copied spectrum's are shown in blue in Figure 2.6.

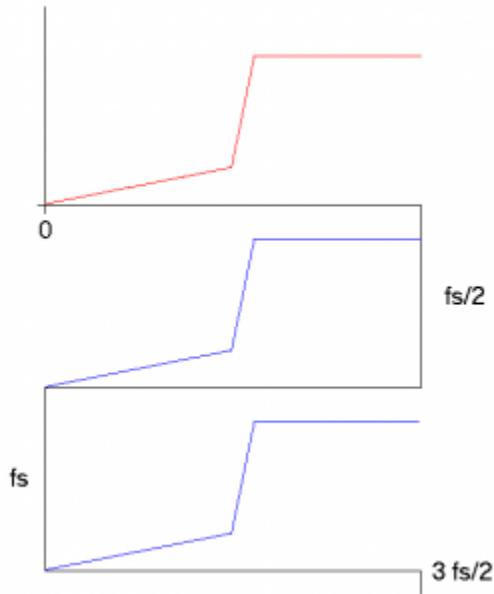


Figure 2.6 Resultant Spectrum Copied into Upper Frequencies.

As a final step, the spectrum of the sampled signal can be drawn by “unfolding” the frequency axis. In Figure 2.7, this unfolded plot shows the resultant in red and the unfolded upper frequency in blue.

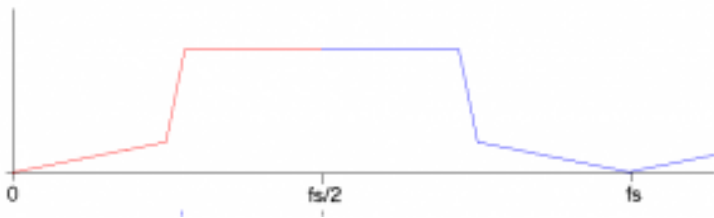


Figure 2.7 Final Spectrum, Unfolded.

EXAMPLE

A signal is sample every 200 micro seconds, giving a sampling frequency of 5 K Hz. The positive spectrum of the signal is shown in Figure 2.8.

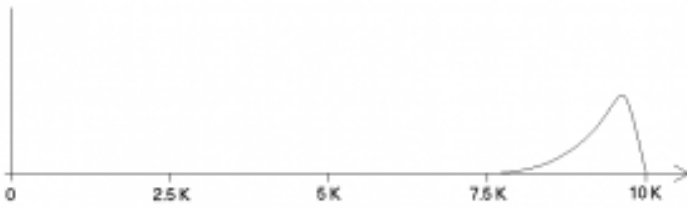


Figure 2.8 Input Signals Spectrum.

The Aliasing Diagram is shown in Figure 2.9, with the original spectrum is in green, the resultant is in red and the other reflected spectrums are in blue.

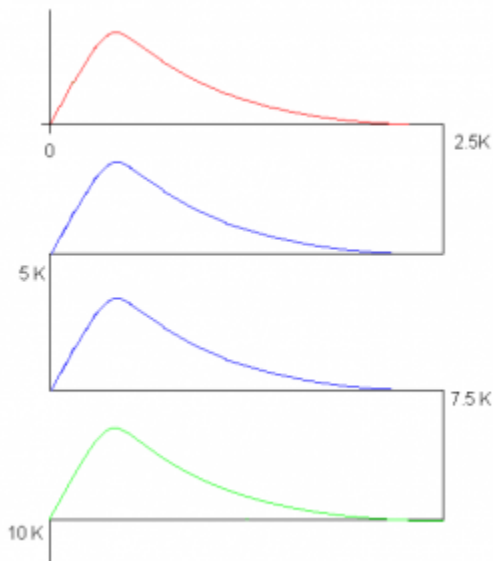


Figure 2.9 Final Aliasing Diagram for Example signal.

With the original, resultant and reflected spectrums we can unfold the spectrum to get the resultant spectrum that is shown in Figure 2.10.

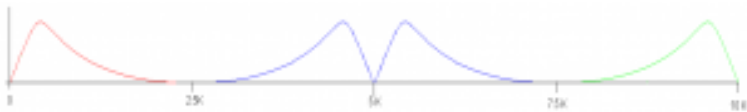


Figure 2.10 Final Spectrum out to 2 fs.

An important aspect of this example is the fact that the original spectrum is still present in the spectrum after sampling. Since the spectrum is not distorted, we will be creating an approach to recover this signal.

The following video will further exemplify the use of the aliasing diagram to interpret the effects of sampling.



One or more interactive elements has been excluded from this version of the text. You can view them online here:

<https://kstatelibraries.pressbooks.pub/dsp-basics/?p=28#oembed-3>

Section C) Digital to Analog Converter (DAC) Reconstruction

Once the sampled data has been filtered by the micro, refer to Figure 1.1, the output sequence is sent to a DAC. The DAC will create an analog voltage that matches the number from the micro. Since the numbers are spaced out at set time intervals, the resulting signal will appear as a blocky (stairstep) waveform. The plot that exemplifies this form is shown in Figure 2.11.

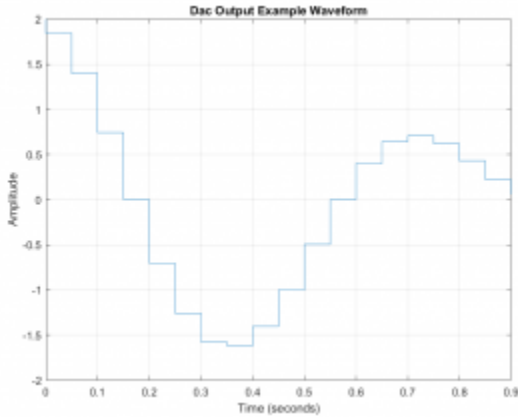


Figure 2.11. Characteristic DAC Output Waveform

Much like the Nyquist model for sampling, we will need to create a system that emulates the shape of this blocky reconstruction. A system like that shown in Figure 2.12 will perform this action.

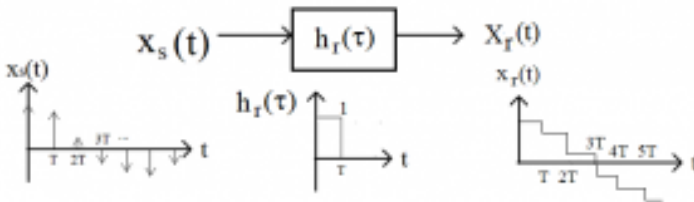


Figure 2.12 Model for Analyzing the DAC Reconstruction

In this model we have taken our input to be a stream of impulses, much like the Nyquist sampled signal, then filtered this stream of impulses with a filter that has an impulse of

response that is a square pulse. The following video will show this process in a graphical form.



One or more interactive elements has been excluded from this version of the text. You can view them online here:

<https://kstatelibraries.pressbooks.pub/dsp-basics/?p=28#oembed-4>

As with the sampling model, we will use the FT to analyze of the input and system to identify the prominent effects of this reconstruction.

We begin by establishing what the FT of a square pulse looks like.

$$H(f) = \int_{-\infty}^{\infty} (h(t) * e^{j\omega t}) dt \quad (2.12)$$

Noting that $h(t)$ is one from $t = 0$ to T , and zero otherwise, we would have

$$H(f) = \int_0^T (1 * e^{j\omega t}) dt \quad (2.13)$$

Now $e^{j\omega t}$ has an easy integral which is equal to $\frac{e^{j\omega t}}{j\omega}$ and thus

$$H(f) = \frac{(e^{j\omega T} - e^{j\omega 0})}{j\omega} \quad (2.14)$$

In order to simplify equation 2.13, we will factor out $e^{j\omega * \frac{T}{2}}$ from the numerator, resulting in

$$H(f) = e^{j\omega \frac{T}{2}} \frac{(e^{j\omega \frac{T}{2}} - e^{-j\omega \frac{T}{2}})}{j\omega} \quad (2.15)$$

Then employing one of the Euler's Identities we have

$$H(f) = e^{j\omega \frac{T}{2}} \frac{2j \sin(\omega \frac{T}{2})}{j\omega} \quad (2.16)$$

Cancelling out the j in the numerator and denominator and multiplying the numerator and denominator by $\frac{T}{2}$ we have

$$H(f) = T e^{j\omega \frac{T}{2}} \frac{\sin(\omega \frac{T}{2})}{\omega \frac{T}{2}} \quad (2.17)$$

This last step of multiplying the numerator and denominator by $\frac{T}{2}$ was done to have the ratio on the right hand side of 2.16, appear as $\frac{\sin(x)}{x}$ which is commonly called $\text{sinc}(x)$.

$$H(f) = T e^{j\omega * \frac{T}{2}} \text{sinc}(\omega * \frac{T}{2}) \quad (2.18)$$

To properly interpret equation 2.18, we need to recognize that the first part, $e^{j\omega * \frac{T}{2}}$ has a magnitude of 1 for all ω and only represents a phase shift. Thus the magnitude response is set by the sinc function, a plot of which is included in Figure 2.13, but the following

alterations to the equation is needed to make the plot more understandable. First, we translate from radial to plain frequency by replacing ω with $2\pi f$ and also replacing T with $\frac{1}{f_s}$

$$H(f) = \frac{1}{f_s} e^{j\pi \frac{f}{f_s}} \text{sinc}\left(\pi \frac{f}{f_s}\right) \quad (2.18)$$

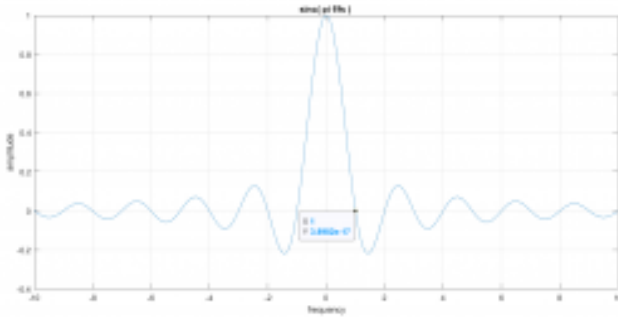


Figure 2.13 Plot of $\text{sinc}\left(\pi * \frac{f}{f_s}\right)$, Normalized to $T = 1$
Second ($f_s = 1$).

Then the output of the system in Figure 2.12 should be the product of the inputs spectrum with the sinc shown above. In equational form we have

$$X_r(f) = X_s(f) * \text{sinc}\left(\pi * \frac{f}{f_s}\right) \quad (2.18)$$

We will now work on a Matlab emulation of this process as a way to visualize the effect of the DAC reconstruction.

Matlab Based Emulation

We start with two support functions, `ins_replicas` and `ins_zeros`, which emulate the DAC output and Nyquist sampled signals.

```
function xn = ins_replicas( x, l );
%
% xn = ins_replicas( x, l );
%
n = length( x );
l = l+1;
% Create an array that is has l*n samples
representing
xn = zeros( 1, l*n ); % an upsampling.
% Loop through the samples in the original
signal.
for k = 1:n
    for m = 1:l % loop to fill in all the samples
        % in the upsampled signal.
        xn((k-1)*l+m) = x(k); % x(k) is thus replicated
    end
end
return
```

```
function xn = ins_zeros( x, l )  
%  
% xn = ins_zeros( x, l );  
%  
n = length( x );  
l = l+1;  
% Create an array that is has l*n samples  
% representing an upsampling.  
xn = zeros( 1, l*n );  
  
% Loop through the samples in the original  
signal.  
for k = 1:n  
    xn(k*l) = x(k); % place x(k) at start of each  
    sample  
end % leaving the other at zero.  
return
```

Using these functions we will now emulate the effect from the DAC reconstruction.

```

% Clear out prior to test.
close all;
clear
% Create index array
n = 0:2047;
% and the test signal that sweeps through
frequencies.
x = cos( pi/4096 * n.*n );
fs = 44e3; % (Sampling frequency assumed to be
44 KHz)
% Simulation of DAC reconstruction by inserting
replicas of samples.
xr = ins_replicas( x, 7 );
% Emulation of Nyquist sampled signal by
inserting zeros.
y = ins_zeros( x,7);

%Fourier Transform of all the data.
X = fft( x );
Y = fft(y);.
XR = fft( xr/8 ); % divide by 8 needed since
inserted replicas add to magnitude.

% plot Signals.
subplot( 3,1,1 ),plot( x );
ylim( [-1.1 1.1] );

```

```

title( 'Original Signal' );
subplot( 3,1,2 ),plot( y );
ylim( [-1.1 1.1] );
title( 'Signal with Inserted Zeros' );
subplot( 3,1,3 ),plot( xr );
ylim( [-1.1 1.1] );
title( 'Signal with Samples Replicated' );

% plot zoomed view of signals.
figure
subplot( 3,1,1 ),plot( x );
xlim( [0 100] );
ylim( [-1.1 1.1] );
title( 'Original Signal' );
subplot( 3,1,2 ),plot( y );
xlim( [0 800] );
ylim( [-1.1 1.1] );
title( 'Signal with Inserted Zeros' );
subplot( 3,1,3 ),plot( xr );
xlim( [0 800] );
ylim( [-1.1 1.1] );
title( 'Signal with Samples Replicated' );

%Plot the spectrums.
figure
subplot( 3,1,1 ),plot( 22e3*[0:length(X(1:end/

```



```

2))-1]/length(X(1:end/2)), abs( X(1:end/2) ) );
% Note that 22e3 is one half of fs = 44K
title( 'Reconstruction Spectrum' );
grid;
subplot( 3,1,2 ),plot( 176e3*[0:length(Y(1:end/
2))-1]/length(Y(1:end/2)), abs( Y(1:end/2) ) );
% And again 176e3 is one half of 8*fs = 352K
title( 'Inserting Zeros' );
grid;
subplot( 3,1,3 ),plot( 176e3*[0:length(XR(1:end/
2))-1]/length(XR(1:end/2)),abs(XR(1:end/2)) );
title( 'Spectrum of DAC Reconstruction' );
grid;

```

The following Figure is the first generated by the above MatLab script.

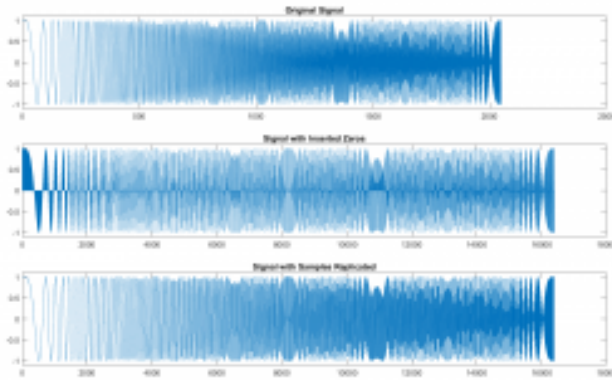


Figure 2.14 Signals for Testing The Effect of a DAC on a Signals Reproduction.

The first plot, at top of Figure 2.14, is the input signal and is a cosine wave who's frequency is continually increasing. This type of signal will be used repeatedly to allow for a visual display of the frequency response of systems. The second plot shows the same signal with zeros inserted between each sample. This is done to emphasis how the Nyquist sampling model is actually representing the sampled signal. Finally the third plot is signal will each sample replicated 4 times, which react similar to the blocky DAC signal we described previously. To display these signals and their shape more clearly The same signals are plotted here, but zoomed in on the first section of the plots.

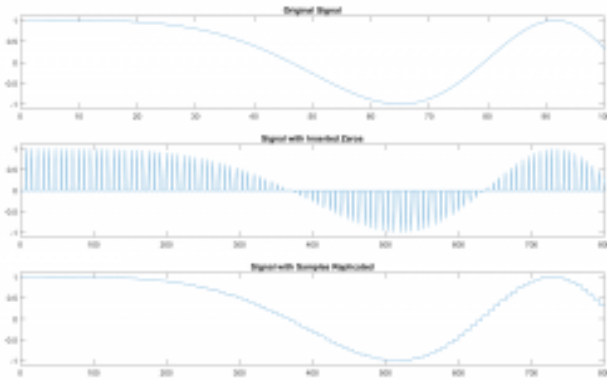


Figure 2.15 First Few Samples of The Signals for Testing The Effect of a DAC on a Signals Reproduction.

Figure 2.16 displays the spectrums of the three signals. This top plot shows the spectrum of the “input” signal, for which it should be noted that it is very wide band. If we could do a true spectrum, feel of the effects of our finite time window it would be flat across all the frequencies. Now to create the second signal, we inserted seven zeros between each of the samples, making it look like the Nyquist sampled signal only sampled at a rate 8 times faster. Note that the spectrum of the zero inserted signal shows 8 copies of the spectrum shown in the top plot. On the other hand, the third plot, which is the replicated signal, shows a distinct roll off of the spectrum. This roll off is the “sinc” that we spoke of previously.

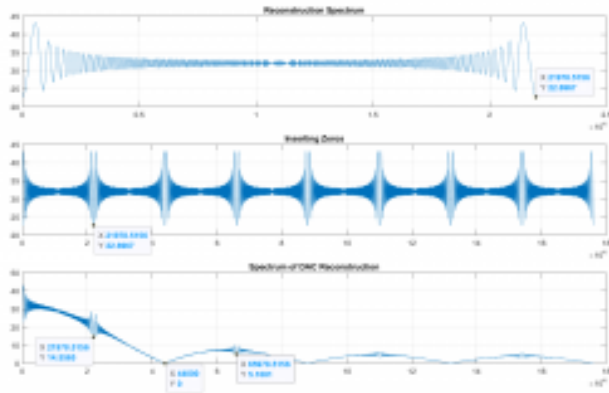


Figure 2.16 Spectrum of The Signals, Showing The Effect of a DAC on a Signals Reproduction.

Looking at the points that have been tagged in the spectrums we can quantify some important facts. First the highest frequency in the input or original signal, at the end of the top plot we have a magnitude of 22.6967, and this is the same as the magnitude in the second spectrum (zero inserted) at the same frequency, indicating that we really do have copies of the original spectrum. On the other hand, the spectrum of the replicated signal shows the roll off. And if we look at the magnitude at the same frequency as shown in the second plot, we see a decrease of $14.5565 / 22.6967 = 0.6413$ which is very close to $\text{sinc}(f / f_s)$ at $f = f_s/2$ or $\text{sinc}(1 / 2) = 0.6366$.

Thus it can be seen that the effect of the DAC on the output signal can be a problem, especially for signals that contain components that approach $f_s/2$. As we develop more techniques to analyze and filter signals, we will return to this

problem and demonstrate ways to correct these frequency components and better recover the signal.

Conclusion

In this section we have looked at how the sampling of a signal can cause a variety of effects, such as causing some frequency components to appear to be a different frequency than they originally were, known as aliasing. We also established that if we are to avoid aliasing of frequency components we will need to sample faster than the Nyquist rate $f_s \geq f_{max}$. Finally we showed that sending samples out to a DAC had some unusual effects.

Probably the most important take away from this chapter is how what appears as a purely theoretical model of a process can be used to explain its effects. Such as sampling being modelled as a stream of impulses being multiplied with a signal and explaining aliasing. Or a stream of impulses being filtered by a filter with a square impulse response can identify a frequency roll off.

3.

Z-TRANSFORM AND PROCESSING SAMPLED DATA.

In the previous chapter we looked what happens to a signal when we reduce it to a sequence of number via an ADC. We will now move on to the next step in our original system in which we will develop the algorithms we will be using in the processor to convert the samples coming in from the ADC into the sequence we send out to the DAC.

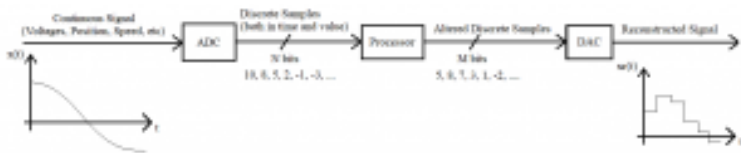


Figure 3.1 Basic Discrete Data System Layout.

Section A) The Z Transform and Its More Important Properties.

In chapter 2, we showed how effective the Nyquist model was at characterizing data that is a sequence of samples, and that the Fourier Transform (FT) of that modelled data

elucidated some very interesting properties. We will now take that sampled data and employ the Laplace transform. Equation 3.1 is the Laplace transform of the input $x(t)$ times the stream of impulses that create the Nyquist sampled signal.

$$X_s(s) = \int_{-\infty}^{\infty} \left(x(t) * \sum_{n=-\infty}^{\infty} \delta(t - nT) \right) e^{-st} dt \quad (3.1)$$

We will move $x(t)$ under the sum and then based on the sifting property of impulse functions, we replace $x(t)$ with $x(nT)$, since $t = nT$ is the only place that $\delta(t - nT)$ is not zero.

$$X_s(s) = \int_{-\infty}^{\infty} \left(\sum_{n=-\infty}^{\infty} x(nT) \delta(t - nT) \right) e^{-st} dt \quad (3.2)$$

As was the case with the FT, we will somewhat dispense with mathematical rigor and assume the proper conditioning on the functions and interchange the summation and integral. Do understand, this is done not to dismiss rigor, but to aid in understanding the process.

$$X_s(s) = \sum_{n=-\infty}^{\infty} x(nT) \int_{-\infty}^{\infty} \delta(t - nT) e^{-st} dt \quad (3.3)$$

Employing the sifting property of the δ function, the integral disappears and it replaced with the summation of the values at $t = nT$. Resulting in

$$X_s(s) = \sum_{n=-\infty}^{\infty} x(nT) e^{-snT} \quad (3.4)$$

Then we will substitute in $z = e^{sT}$ and remove the redundant T from $x(nT)$ and we will have equation 3.5

$$X_s(z) = \sum_{n=0}^{\infty} (x(n) * z^{-n}) \quad (3.5)$$

This summation equation (3.5) is known as the z transform, and much like the Laplace transform it will be our tool to understand and design discrete systems. Since the z transform is the Laplace of a sampled signal, it will provide us with similar properties and results as the Laplace, but will subtle differences.

For example, let's consider the effect of delaying the sampled sequence by one sample. We will first note that the z transform is a summation and looks like

$$\sum_{n=0}^{\infty} (x(n) * z^{-n}) = x(0) + x(1)z^{-1} + x(2)z^{-2} + x(3)z^{-3} + \dots \quad (3.6)$$

The z transform of the delayed version, $x(n-1)$, is then

$$\bar{X}(z) = \sum_{n=0}^{\infty} x(n-1)z^{-n} \quad (3.7)$$

Writing out the summation we would have

$$\begin{aligned} \bar{X}(z) &= x(-1) + x(0)z^{-1} + x(1)z^{-2} + x(2)z^{-3} + \\ &\dots \end{aligned} \quad (3.8)$$

Note the ... indicates that the sequence continues on in the manner shown. Next we set out $x(-1)$ and then factor out a single z^{-1} for the remainder.

$$\bar{X}(z) = x(-1) + z^{-1}(x(0) + x(1)z^{-1} + x(2)z^{-2} + \dots) \quad (3.9)$$

We can now observe that sum on the right, in parenthesis, is the z transform of $x(n)$ or $X(z)$.

$$\bar{X}(z) = x(-1) + z^{-1} X(z) \quad (3.10)$$

An appendix that relates the z transform properties with the Laplace transform properties is at the end of this text.

This property is analogous to the Laplace transform of the derivative of a time function.

Let $X(s)$ be the Laplace transform of the function of $x(t)$ then

$$L\left\{ \frac{dx}{dt} \right\} = s X(s) + x(0) \quad (3.11)$$

As a reminder of what the purpose of the Laplace transform is, consider the following circuit

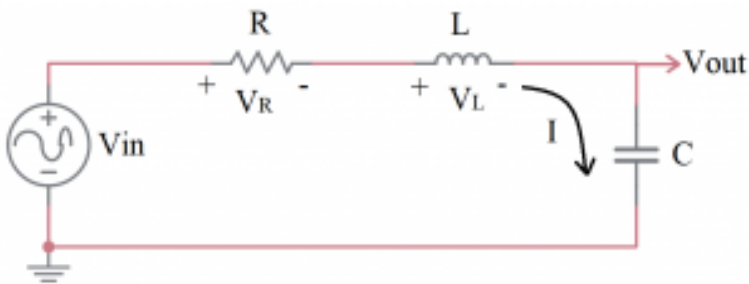


Figure 3.2 Circuit Used to Demonstrate The Purpose of The Laplace Transform.

Applying the Kirchhoff Voltage Law (KVL) around the circuit, we have

$$-V_{in} + V_R + V_L + V_{out} = 0 \quad (3.12)$$

Recalling from physics, the voltage on the resistor (R) and inductor (L) are $I * R$ and $L \frac{dI}{dt}$ respectively. Also that

$$I = C * \frac{V_{out}}{dt} \text{ or } \frac{dI}{dt} = C * \frac{V_{out}^2}{d^2 t}.$$

Applying this to equation 3.12 and solving for V_{in} we have

$$V_{in} = L \frac{dI}{dt} + I * R + V_{out} \quad (3.13)$$

Note that in this circuit, $I = C * \frac{V_{out}}{dt}$ and thus

$$\frac{dI}{dt} = C * \frac{V_{out}^2}{d^2 t} \text{ resulting in}$$

$$V_{in} = L C \frac{\{V_{out}\}^2}{d^2 t} + R C \frac{V_{out}}{dt} + V_{out} \quad (3.14)$$

4.

APPENDIX A. LAPLACE AND Z TRANSFORM PROPERTIES.

This property is analogous to the Laplace transform of the derivative of a time function.

Let $X(s)$ be the Laplace transform of the function of $x(t)$ then

$$L\left\{ \frac{dx}{dt} \right\} = sX(s) + x(0)$$

(3.11)

As a reminder of what the purpose of the Laplace transform is, consider the following circuit

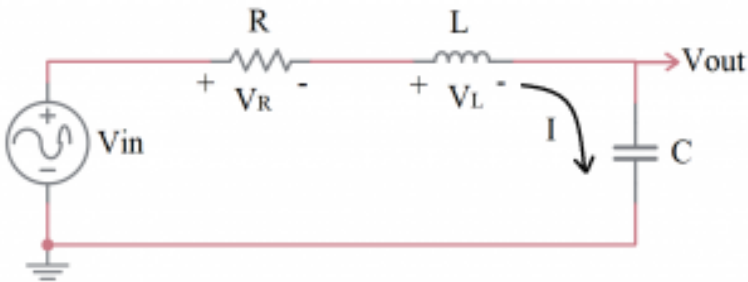


Figure 3.2 Circuit Used to Demonstrate The Purpose of
The Laplace Transform.

Applying the Kirchhoff Voltage Law (KVL) around the circuit, we have

$$-V_{in} + V_R + V_L + V_{out} = 0 \quad (3.12)$$

Recalling from physics, the voltage on the resistor (R) and inductor (L) are $I * R$ and $L \frac{dI}{dt}$ respectively. Also that

$$I = C * \frac{V_{out}}{dt} \text{ or } \frac{dI}{dt} = C * \frac{V_{out}^2}{d^2 t}.$$

Applying this to equation 3.12 and solving for V_{in} we have

$$V_{in} = L \frac{dI}{dt} + I * R + V_{out} \quad (3.13)$$

Note that in this circuit, $I = C * \frac{V_{out}}{dt}$ and thus

$$\frac{dI}{dt} = C * \frac{V_{out}^2}{d^2 t} \text{ resulting in}$$

$$V_{in} = L C \frac{\{V_{out}\}^2}{d^2 t} + R C \frac{V_{out}}{dt} + V_{out} \quad (3.14)$$

This is where you can add appendices or other back matter.