Digital Signal Processing: COSC360

Jordan Hanson
January 13, 2022
Whittier College Department of Physics and Astronomy

Unit 1.3 Outline

Previous lectures covered:

- Complex numbers 2: The Fourier series and Fourier transform (continuous and discrete)
- *Time-permitting*: The Laplace transform (continuous and discrete)

This lecture will cover: (Reading: **Chapter 2**)

- Statistics and probability: the normal distribution and other useful distributions
- Noise: digitization and sampling
- Noise: Spectral properties of noise, ADC and DAC

Statistics and Probability: The Normal Distribution

The *mean*, μ, and *standard deviation*, σ, of a data set $\{x_i\}$ are defined as

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{1}$$

$$\sigma^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu)^2 \tag{2}$$

Octave commands:

```
x = randn(100,1);
mean(x)
std(x)
```

One nice theorem: *The variance is the average of the squares minus the square of the average.* Let $\langle x \rangle$ represent the average of the quantity or expression x. We have

$$\sigma_x^2 = \langle x^2 \rangle - \langle x \rangle^2 \tag{3}$$

Proof: observe on board.

Note: There is a distinction between the *process or signal process* and the *the data*. Just because the data has a given μ and σ does not imply that the signal process has or will continue to have the exact same values of μ and σ. The underlying process could be *non-stationary*.

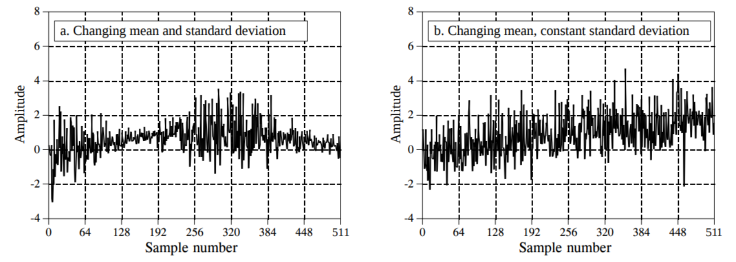


Figure 1: Signal processes in (a) and (b) are considered non-stationary because one or both of μ and σ depend on time.

A histogram is an object that represents the frequency[1] of particular values in a signal. For example, below is a histogram of 256,000 numbers drawn from a probability distribution:

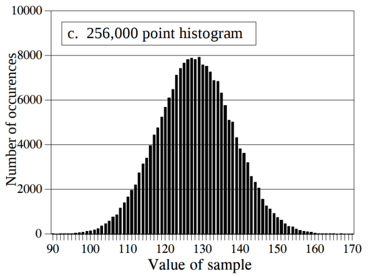


Figure 2: The histogram contains counts versus sample values.

[1] Careful: the word frequency refers to the number of occurences in the data, not a sinusoidal frequency.

The following octave code should reproduce something like Fig. 2 from the textbook:

```
x = randn(256000,1)*10.0+130.0;
[b,a] = hist(x,100);
plot(a,b,'o');
```

The function *randn(N,M)* draws $N \times M$ numbers from a normal distribution and returns them in the size the user desires. The function *hist(x,N)* creates N bins and sorts the data x_i into them.

For data that is appropriately stationary, we can use histograms to estimate μ and σ faster, since we only have to loop over bins rather than every data sample. Let H_i represent the counts in a given bin, and i represent the bin sample. We have:

$$\mu = \frac{1}{N} \sum_{i=1}^{M} i H_i \tag{4}$$

$$\sigma^2 = \frac{1}{N-1} \sum_{i=1}^{M} (i - \mu)^2 H_i \tag{5}$$

To obtain the mean in signal *amplitude*, you'll have to convert bin number to amplitude.

3.1
-0.03
1.2
0.2
-0.7
-1.45
2.2
-0.05
0.93
0.21

Table 1: Using Eq. 4 and 5, find estimates of μ and σ for this data.

```
x = [...];
[b,a] = hist(x,4); %(How many bins?)
```

Some vocabulary:

- **normalization** - Total probability is 1.0. For pdf - the integral from $[-\infty, \infty]$ is 1.0. For pmf - the sum from $[-\infty, \infty]$ is 1.0.
- **pmf** - Probability mass function: A *normalized continuous function* that gives the probability of a value, given the value.
- **histogram** - Histograms are an attempted measurement of the pmf by breaking the data into discrete bins. Histograms can be *normalized* as well.
- **pdf** - Probability density function: A *normalized continuous function* that gives the probability density of a value, given the value. Integrating the *normalized* pdf between two values gives the probability of observing data between the given values.

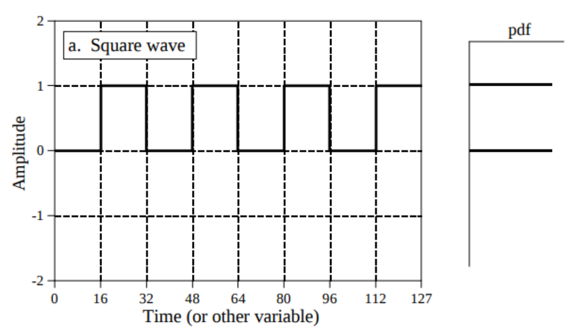


Figure 3: The square-wave signal spends equal time at 0.0 and 1.0, and the probability density function reflects that.

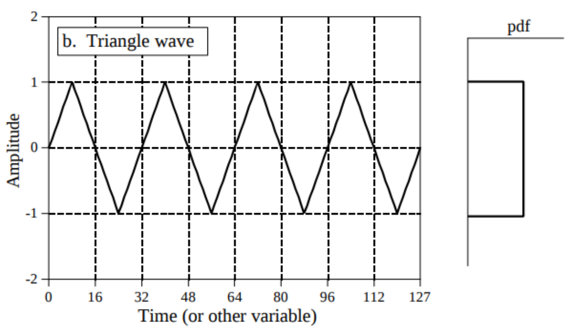


Figure 4: The triangle-wave signal spends equal time at all values *between* 0.0 and 1.0, and the probability density function reflects that.

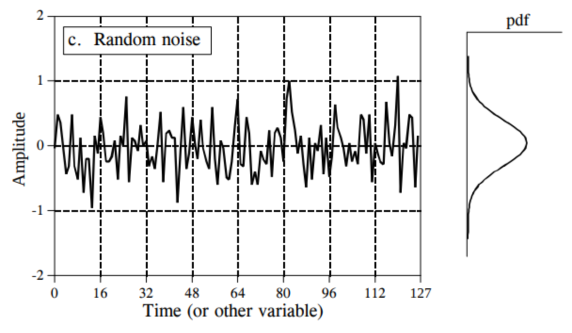


Figure 5: The random noise *usually* spends time near 0.0, but rarely it fluctuates to larger values.

Normally distributed data decreases in probability at a rate that is proportional (1) to the *distance from the mean*, and that is proportional (2) to the *probability itself.*

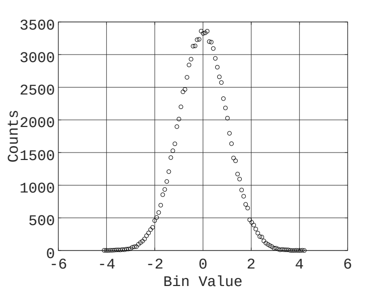


Figure 6: Normally distributed data counts decrease as measured further from the mean for *two reasons.*

Normal Distribution PDF

Let $p(x)$ be the PDF of normally distributed data x with mean μ. In order to obey conditions (1) and (2), the function $p(x)$ must be described by the following differential equation, where k is some constant.

$$\frac{dp}{dx} = -k(x - \mu)p(x) \tag{6}$$

Normal distribution

Rearranging Eq. 6, we have

$$\frac{dp}{p} = -k(x-\mu)dx \tag{7}$$

Integrating both sides gives

$$\ln(p) = -\frac{1}{2}k(x-\mu)^2 + C_0 \tag{8}$$

Exponentiating,

$$p(x) = C_1 \exp\left(-\frac{1}{2}k(x-\mu)^2\right) \tag{9}$$

Ensuring that the PDF is *normalized* requires

$$\int_{-\infty}^{\infty} p(x)dx = 1 \tag{10}$$

Normal distribution

But how do we integrate Eq. 9? First, a change of variables. Let $s = \sqrt{k/2}(x - \mu)$, so $ds = \sqrt{k/2}dx$. Then, we have

$$C_1\sqrt{\frac{2}{k}}\int_{-\infty}^{\infty} \exp(-s^2)ds = 1 \tag{11}$$

Squaring both sides, we have

$$C_1^2\frac{2}{k}\left(\int_{-\infty}^{\infty} \exp(-s^2)ds\right)^2 = 1 \tag{12}$$

Let's pretend the two factors of the integral involve different variables:

$$C_1^2 \frac{2}{k} \left(\int_{-\infty}^{\infty} \exp(-x^2) dx \right) \left(\int_{-\infty}^{\infty} \exp(-y^2) dy \right) = 1 \qquad (13)$$

Now we have

$$C_1^2 \frac{2}{k} \int_{-\infty}^{\infty} \exp(-(x^2 + y^2)) dx dy = 1 \qquad (14)$$

Change to polar coordinates ($x^2 + y^2 = r^2$)

$$C_1^2 \frac{2}{k} \int_0^{\infty} \int_0^{2\pi} r \exp(-r^2) dr d\phi = 1 \qquad (15)$$

Normal distribution

One more substitution: $u = r^2$, and $du = 2rdr$:

$$-\frac{C_1^2}{k}\int_0^\infty \int_0^{2\pi} \exp(-u) du d\phi = 1 \tag{16}$$

Solving for C_1, we find

$$C_1 = \sqrt{\frac{k}{2\pi}} \tag{17}$$

Thus the pdf of normally distributed data is

$$p(x) = \sqrt{\frac{k}{2\pi}} \exp\left(-\frac{1}{2}k(x-\mu)^2\right) \tag{18}$$

Let's defined $k = \frac{1}{\sigma_x^2}$ so that it's clear the exponent has the proper ratio of units:

$$\boxed{p(x) = \sqrt{\frac{1}{2\pi\sigma_x^2}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma_s}\right)^2\right)} \tag{19}$$

Statistics and Probability: Programming with Octave

More on the *hist* function in octave[2]

```
pkg install -forge io
pkg install -forge statistics
pkg load statistics
pkg help histfit
histfit(randn(1000,1))
histfit(rand(1000,1))
```

Let's work out the σ of a *flat* distribution between $[0,1]$. What is it for a flat distribution between $[-1,1]$? (We can derive this by hand as well if we cannot access statistics package).

[2] I hope this works, but if not, it's ok.

Some interesting notation for normal distributions:

$$N(\mu, \sigma) = \sqrt{\frac{1}{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \tag{20}$$

Let's write a function NGaus.m that produces the Gaussian probability given μ and σ:

```
function ret = NGaus(mu,sigma,x)
    ...
endfunction
```

Now let's write a function *NRand* that sums N uniformly-distributed (flat) random variables x:

```
function ret = NRand(n)
    ret = sum(rand(n,1));
endfunction
```

Create a histogram of a few hundred outputs of *NRand*. What do you notice about the pmf? Let's plot *NGaus* on the same axes as the histogram of *NRand*. How do they compare?

We are on our way to producing $N(0,1)$ distributed numbers, and therefore our first **noise** signals...

The Box-Muller method for $N(0,1)$ distruted numbers:

$$X_1 = \sqrt{-2\ln(U)}\cos(2\pi V) \tag{21}$$

$$X_2 = \sqrt{-2\ln(U)}\sin(2\pi V) \tag{22}$$

Try this in octave... More vocabulary:

- **cdf** - Cumulative distribution function: Probability that a continuous random variable X is less than some value x. For a given pdf, the cdf $\Phi(X)$ is the integral of the total probability on $[-\infty, x]$. The derivative of the pdf is related to the pdf via the fundamental theorem of calculus.

If the pdf follows $f(x)$, then

$$\Phi(X \leq x) = \int_{-\infty}^{x} f(x)dx \tag{23}$$

The cdf of $N(0,1)$ has an expected shape, but can't be expressed with elementary functions.

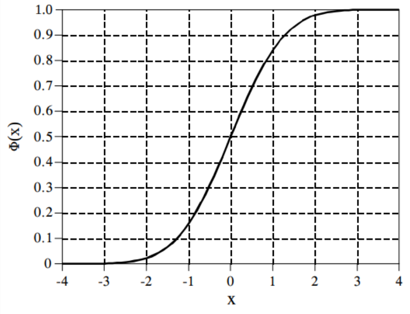


Figure 7: The cumulative distribution of the normal distribution. Although we can plot it, it's hard to write. We will discuss the *erf* and *erfc* functions in the near future.

Statistics and Probability: Other Useful Distributions

Statistics and Probability: Other Useful Distributions

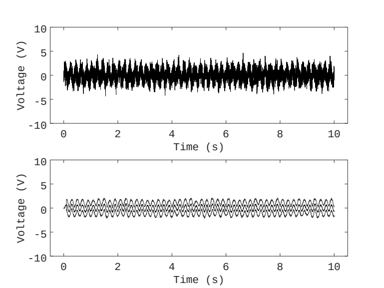
We now know how to obtain random uniform numbers (**rand**) in octave, and have algorithms (Box-Muller) and functions (**randn**) in octave for $N(0,1)$[3]. What if we require a *different pdf*? One technique is to use *inverse transform sampling:*

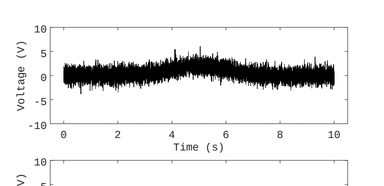
1. For the pdf $p(x)$, work out the cdf $\Phi(x)$.
2. Generate a sample of uniform random numbers $u_i \in [0,1]$.
3. Call $\Phi^{-1}(u_i)$, that is, invert the cdf and plug in the list u_i to the dependent variable.

Write an octave script that generates exponentially-distributed numbers, e.g. pdf $\propto \exp(-x)$.

[3] This can be scaled to any μ and σ values we need.

Octave Programming: The scripts **meanStdDev...m** on Moodle demonstrate different digitized signals. Examine the effect of changing the pdf of the noise from Gaussian to exponential.





Octave has many pre-programmed distributions. Although system noise is usually normally distributed, it's good to know these:
`https://octave.org/doc/v4.2.0/Distributions.html`

Useful video on inverse transform sampling:
`https://www.youtube.com/watch?v=rnBbYsysPaU&t=373s`

Supplement: Damped driven harmonic oscillator

Supplement: Damped driven harmonic oscillator

Let the defining equation for the signal $x(t)$ be

$$x'' + bx' + cx = A\cos(\omega_0 t) \tag{24}$$

Take the Fourier transform of both sides:

$$\left((j\omega)^2 + j\omega b + c\right) X(\omega) = A \int_{-\infty}^{\infty} \cos(\omega_0 t) \exp(-j\omega t) dt \tag{25}$$

The right-hand side is a pair of delta-functions:

$$F\{\cos(\omega_0 t)\} = \frac{1}{2}\left(\delta(\omega - \omega_0) + \delta(\omega + \omega_0)\right) \tag{26}$$

Supplement: Damped driven harmonic oscillator

Solving for $X(\omega)$ gives

$$X(\omega) = -\frac{A}{2}\frac{\delta(\omega-\omega_0)+\delta(\omega+\omega_0)}{\omega^2+j\omega b+c} \tag{27}$$

Taking the inverse Fourier transform gives

$$x(t) = -\frac{A}{4\pi}\left(\frac{\exp(-j\omega_0 t)}{\omega_0^2+j\omega_0 b+c}+\frac{\exp(j\omega_0 t)}{\omega_0^2+j\omega_0 b+c}\right) \tag{28}$$

Let $k^2 = \omega_0^2 - c$. (Finish on board). Are there any special conditions you see (resonance, total damping)? Can we plot this in Octave for varying damping and c parameters?

Noise: Digitization and Sampling, theory and examples

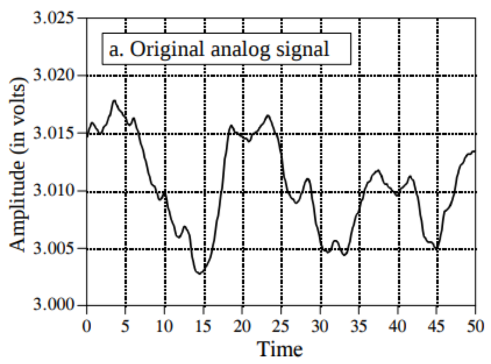


Figure 9: An example of analogue data from chapter 2 of the text. Both the dependent and independent axes are continuous.

Noise: Digitization and Sampling

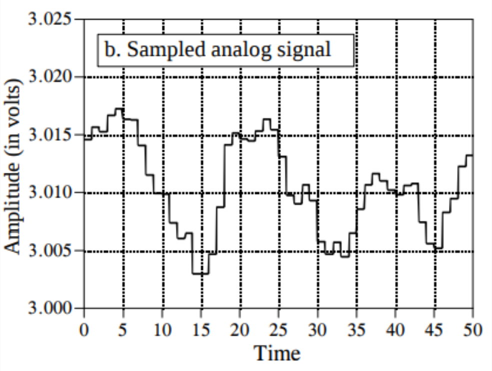


Figure 10: The same signal from Fig. 9, except a *sample-and-hold* action has been applied to the independent variable.

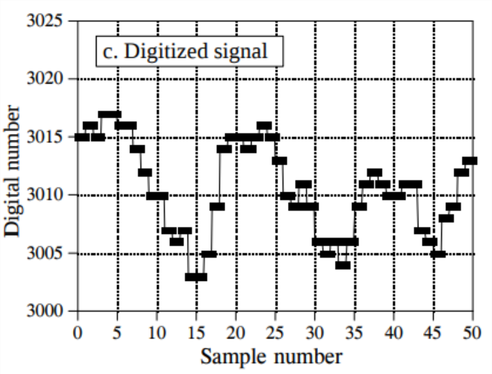


Figure 11: The same signal from Fig. 10, except a *digitization* action has been applied to the dependent variable.

Noise: Digitization and Sampling

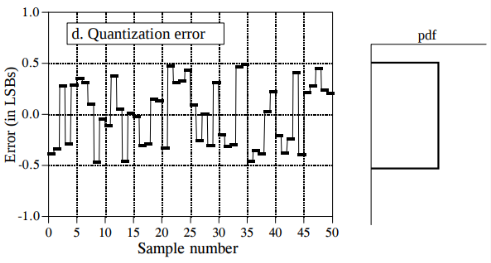


Figure 12: The error incurred by the *digitization* action from Fig. 11. The y-axis is expressed in units of LSB (least significant bit - more in a second). Turns out we know the σ of this error: $LSB/sqrt12$.

A model for a particular value in Fig. 10, the sample-and-hold action[4] is

$$s_n(t) = f(n\Delta t) square(t - n\Delta t) \tag{29}$$

where $f(n\Delta t)$ is the function or data value, and

$$square(t) = 1, \quad |t| \leq T/2 \tag{30}$$

The entire N-sample data-set or signal is

$$s(t) = \sum_{n=0}^{N-1} f(n\Delta t) square(t - n\Delta t) \tag{31}$$

[4]Technically, this is the 0^{th}-order hold, and there are other (much) less common choices.

Sample/Hold Signal Model

$$s(t) = \sum_{n=0}^{N-1} square(t - n\Delta t) \tag{32}$$

Several important questions:

1. What is $S(\omega)$?
2. What are the important relationships between Δt, N, and the frequencies present in the data?
3. How precisely does $s(t)$ represent the data?

Noise: Digitization and Sampling

The Fourier transform of $s(t)$ may be obtained using a combination of properties of the Fourier transform, plus the result obtained for $F\{square(t)\}$. Let $x = \omega\Delta t/2$. The result is (observe on board):

$$S(\omega) = sync(x)\sum_{n=0}^{N-1} f(n\Delta t)\exp(-j\omega n\Delta t)\Delta t \tag{33}$$

The factor at right is a discrete version of the Fourier Transform. Let the DFT represent the discrete Fourier transform on the right. Equation 33 may be written

$$S(\omega) = DFT\{f(t)\}sync(x) \tag{34}$$

The spectrum of a sampled signal is the **convolution** of the discrete Fourier transform of the signal and the sync function with a period of the time between samples.

The convolution of two functions $f(t)$ and $g(t)$ is

$$(f \circ g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{35}$$

Convolution theorem: The Fourier transform of the convolution of two functions $f \circ g$ is

$$F\{f \circ g\} = F(\omega)G(\omega) \tag{36}$$

The DFT can contain only have a finite number of frequencies, since it is discrete. What are the limits of this?

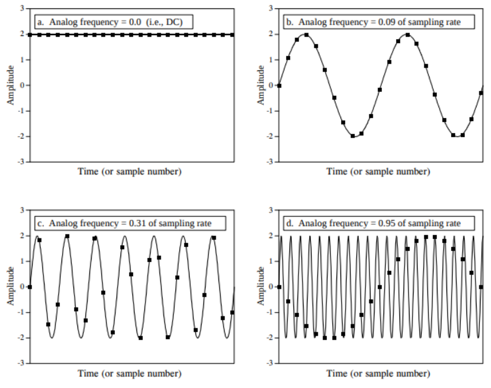


Figure 13: Various degrees of sampling.

Notice that the sync function has a zero, which occurs at $x = \pi$, for some frequency f_s. This implies that

$$\pi = \frac{\omega \Delta t}{2} \tag{37}$$

$$\pi = \frac{2\pi f_s \Delta t}{2} \tag{38}$$

$$f_s = \frac{1}{\Delta t} \tag{39}$$

The frequency f_s is known as the *sampling frequency*.

We have finally arrived at the *sampling theorem:*

Sampling Theorem

A signal containing frequencies less than or equal to $f_{crit} = f_s/2$ can be perfectly reconstructed.

Let's go back and think about Fig. 13.

The DFT can contain only have a finite number of frequencies, since it is discrete. What are the limits of this?

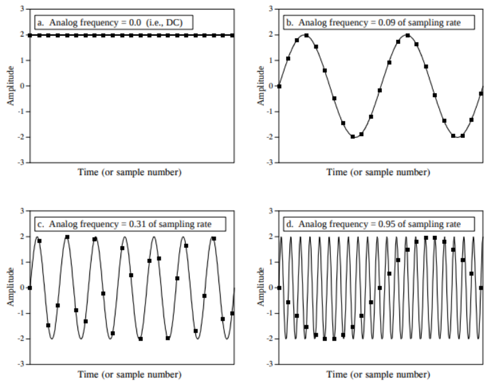


Figure 14: What if the sine wave had a frequency of f_c?. (Professor draw on board).

Equation 33 contained a form of the DFT. Let $h(k\Delta t) = h_k$, with $f_s = 1/\Delta t$ and N time samples. The discrete Fourier transform is defined as

$$H_n \approx \Delta t \sum_{k=0}^{N-1} h_k \exp\left(-2\pi j k \frac{n}{N}\right) \tag{40}$$

The integral is approximated at frequencies $f_n = \frac{n}{n\Delta t}$. The inverse DFT is

$$h_k \approx \frac{\Delta f}{N} \sum_{n=0}^{N-1} H_n \exp\left(2\pi j k \frac{n}{N}\right) \tag{41}$$

What is Δf? There are $N/2$ *independent frequencies* for real data, so $\Delta f = f_c/(N/2) = T^{-1}$.

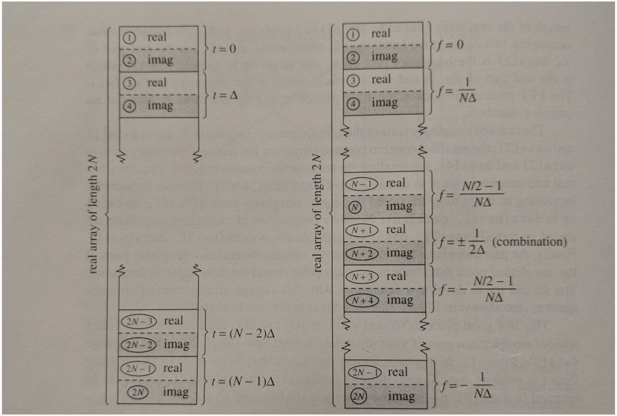


Figure 15: The FFT must conserve degrees of freedom. The data are organized to optimize speed and efficiency. (Left) Time samples. (Right) Frequency samples.

Noise: Digitization and Sampling

Therefore in some implementations (**not** including octave) transforming forward and then backward incurs a factor of N.

Looking at Fig. 15, let's do a few things:

1. Convince ourselves that $\Delta f = 1/T$.
2. Convince ourselves that degrees of freedom are conserved.
3. Convince ourselves that the data $\pm 1/2\Delta t$ should be the same.
4. Ensure we understand the order of the negative frequencies.

Parseval's theorem works like this for the discrete quantities:

$$\sum_{k=0}^{N-1} |h_k|^2 = \frac{1}{N} \sum_{n=0}^{N-1} |H_n|^2 \tag{42}$$

Noise: Digitization and Sampling, Octave coding example

Noise: Digitization and Sampling

How do we determine if data has been *properly sampled* or *properly sampled*? Aliasing.[5] On Moodle, obtain the Aliasing.m script.

1. Activity: run for the first time, and take a few moments to understand the output. What are the units of the axes? What is the lower panel showing?
2. Add noise by boosting the standard deviation of the pdf of the noise distribution by increasing the parameter noise_sigma.
3. What happens to the amplitude when you increase the number of modes in the Fourier series?
4. Push the Fourier modes way above the sampling rate. What happens to the amplitude?[6] *Probably best to minimize noise.*

[5] This is apart from the trivial case when we know f_s and f_{max}.
[6] Do you see the Gibb's phenomenon disappear? Why?

Aliasing. Building off of the Aliasing.m script, do the following:

1. *Filter the data* according to the transfer function of the single-pole low-pass filter.
2. Use this technique to get rid of any aliasing, and explore the effect on the Fourier modes and noise.
3. Limit the Fourier series to ≈ 25 terms, and use $f_0 \approx 1$ Hz. Plot the signal while varying f_s. What do you notice?

Looks like adding noise in frequency-space where there is no signal just distorts the signal. Does this make sense?

Now we return to the problem of precision due to *digitization* in the *dependent variable*, as opposed to *sampling* in the independent variable.

Recall how the **standard deviation** in the digitized dependent variable is $LSB/\sqrt{12}$. Let's begin calling the least-significant bit (LSB) the quantum (Q). Now imagine we are digitizing analog sinusoids between $-V/2$ and $V/2$ with N quanta. The standard deviation of data distributed like a sinusoid is $V/(2\sqrt{2})$. Taking the ratio of the two standard deviations will give us the signal to noise ratio.

Noise: Digitization and Sampling

Written in equation form, these ideas translate as follows:

$$\sigma_S = \frac{V}{2\sqrt{2}} = \frac{2^N Q}{2\sqrt{2}} \tag{43}$$

$$\sigma_Q = \frac{Q}{\sqrt{12}} \tag{44}$$

$$\frac{\sigma_S}{\sigma_Q} = \frac{\frac{2^N Q}{2\sqrt{2}}}{\frac{Q}{\sqrt{12}}} \tag{45}$$

$$\frac{\sigma_S}{\sigma_Q} = \frac{2^N \sqrt{12}}{2\sqrt{2}} \tag{46}$$

$$\tag{47}$$

Often SNR is quoted in decibels. The formula for decibel is

$$P_{dB} = 10 \log(P_2/P_1) \tag{48}$$

Since $P \propto V^2$,

$$P_{dB} = 20 \log(V_2/V_1) \tag{49}$$

Noise: Digitization and Sampling

Applying the definition of decibel to Eq. 46:

$$SNR_{dB} = 20\log\left(\frac{2^N\sqrt{12}}{2\sqrt{2}}\right) \tag{50}$$

$$SNR_{dB} = 20N\log(2) + 20\log(\sqrt{6}/2) \tag{51}$$

$$SNR_{dB} = 6.02N + 1.76 \tag{52}$$

Example: in the presence of *just* quantization noise, if we digitize with 8 bits, we expect an SNR $\approx$ 50 dB. What if we drop to 4 bits? Answer: $\approx$ 26 dB. What if there is *analogue* noise in addition to the *quantization* noise? Suppose it has standard deviation of σ_N.

$$\frac{\sigma_S}{\sigma_Q + \sigma_N} = \frac{\frac{2^N Q}{2\sqrt{2}}}{\frac{Q}{\sqrt{12}} + \sigma_N} \tag{53}$$

$$SNR_{dB} = 6.02N + 1.76 - 20\log\left(1 + \sqrt{12}\frac{\sigma_N}{Q}\right) \tag{54}$$

Recalling that $\sigma_Q = Q/\sqrt{12}$, we have

$$SNR_{dB} = 6.02N + 1.76 - 20\log\left(1 + \frac{\sigma_N}{\sigma_Q}\right) \tag{55}$$

Example: suppose we have $\sigma_N = 20$ mV, and we have $N = 8$ bits over a 2.56 V range. This means $\sigma_Q = 2.56/256/\sqrt{12} = 1/(100\sqrt{12}) \approx 2.89$ mV. Thus, $\sigma_N/\sigma_Q = 20/2.89 \approx 6.92$. The final answer is then $SNR_{dB} = 32$. This is the maximum achievable SNR (in decibels) in this system.[7]

[7] More realistically, the noises add in quadrature, but in practice the system is designed so that σ_N dominates over σ_Q.

What are the maximum SNR values in dB acheivable on the following systems?

1. 16 bits, 6.55V, with 50 mV of analogue noise.
2. 12 bits, 4.096V, with 50 mV of analogue noise.
3. 12 bits, 4.096V, with 5 mV of analogue noise.

Noise: Digitization and Sampling

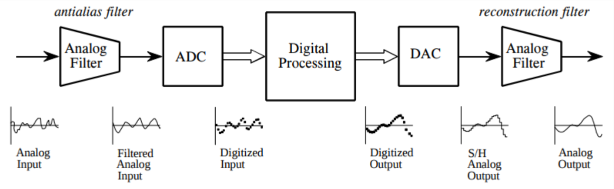


Figure 16: The complete (basic) system for sampling and digitization, reconstruction.

Noise: Digitization and Sampling

We already know what the reconstruction filter should be: the inverse of the sync function encountered in the derivation of the sampling theorem ideas.

What type of anti-aliasing filter should be chosen? Chapter 3 of the text covers three:

1. Butterworth
2. Bessel
3. Chebyshev

But this begs the question. How do filters work in general? Besides the single-pole examples we've already seen, what other types are there?
We have now reached **Unit 2!**

Conclusion

Unit 1.3 Outline

Previous lectures covered:

- Complex numbers 2: The Fourier series and Fourier transform (continuous and discrete)
- *Time-permitting*: The Laplace transform (continuous and discrete)

This lecture will cover: (Reading: **Chapter 2**)

- Statistics and probability: the normal distribution and other useful distributions
- Noise: digitization and sampling
- Noise: Spectral properties of noise, ADC and DAC