Sampling and Aliasing

Matlab Case Study for Signals and Systems (Draft)

# Introduction

There will be situations where we will want to analyze an analog (continuous) signal in a digital environment such as MATLAB. In these cases, it is important to remain aware of the differences between continuous and discrete signals and how sample rates can affect the transition between them.

In this mini-case study, you will:

* Observe the effects of aliasing on various signals
* Use the Nyquist Sampling Theorem to interpolate a signal from a lower sample rate to a higher one.

When you’re finished you will have a greater awareness of the possible risks of discretizing a continuous signal and a greater understanding of digital interpolation methods.

# Sampling

When we want to represent a continuous signal in a discrete environment, we do so by taking samples of the signal at regular intervals. Blah blah blah todo

# Aliasing

To have an alias means to be known by another name. Similarly, a signal experiences “aliasing” when it is sampled at such a low rate that it is indistinguishable from another signal.

[visual aid: two signals that become identical due to aliasing]

When aliasing occurs in our sample, we have essentially “lost” information. While we can try to interpolate our signal to get a higher sample rate,

# Interpolation

Sometimes once we have a discrete version of a signal, we might find that its sample rate is too low to be accurate for our purposes. However, there are a variety of techniques we can use to “interpolate” a discrete signal – deriving a continuous signal, or a new discrete signal with a higher sample rate.

A picture containing table, different, group, light

Description automatically generated

Figure 1: A discretely-sampled signal can be interpolated to a higher sample rate

There are many interpolation techniques. The simplest is linear interpolation – drawing a straight line between each point on the graph and using it to approximate intermediate values. However, this process is obviously not perfect. Anything could be happening between two adjacent samples of a signal, and a straight line might not be the best way to represent it.

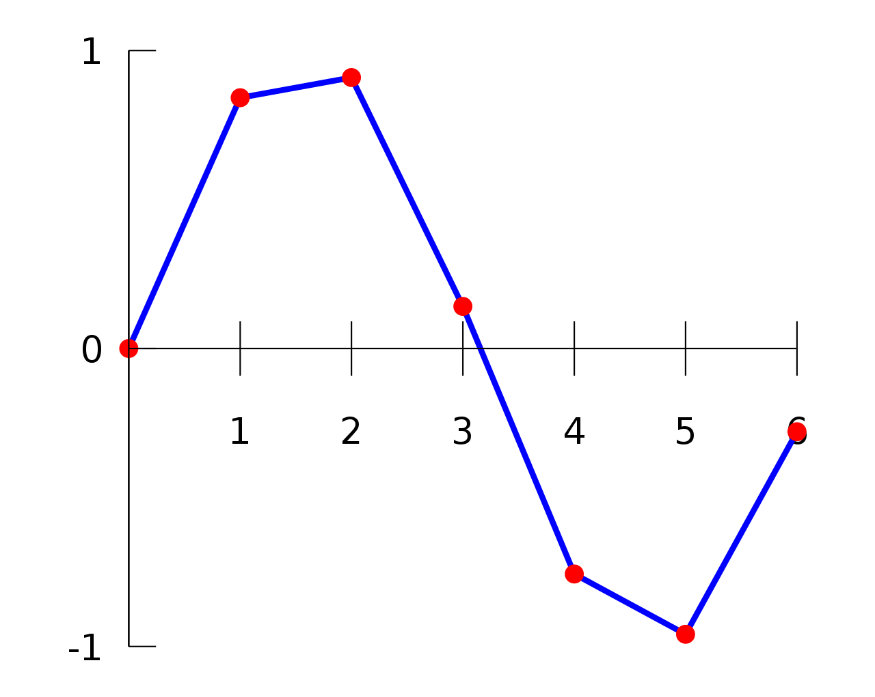


Figure 2: Linear interpolation

Under the right conditions, however, some techniques allow us to interpolate a signal perfectly, recreating the original. One of these conditions is that the signal is band-limited. This means that if we were to look at all the frequencies present in the signal, it would contain no frequencies above a certain bandwith B.

[Visual aid: Band-limited function]

For instance, the function below is band-limited: it is a sum of sine waves that have a frequency of 10 Hz or below, so its bandwidth is 10 Hz.

The second condition is that the sample rate is at least twice the bandwidth. If a signal contains a 50 Hz wave, the sample rate must be at least 100. This sample rate of twice the bandwidth is called the Nyquist Sample Rate. As long as we sample at this rate or higher, we can always reconstruct our signal perfectly.

If both of these conditions are met, we can reproduce the original signal perfectly from a low-sample-rate version of itself using the following formula:

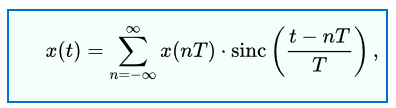


Figure 3: Whittaker-Shannon Interpolation Formula

Where T is the sample period. Note that x(nT) corresponds to the nth sample of the discrete signal.

# Case Study

* **Sampling and Aliasing**
  + Run the provided code and examine the resulting plot. The left side shows a discretely sampled version of several signals. The right side shows the same signals at a much higher sample rate (such that it is essentially continuous).
  + Which of the signals experience aliasing?
  + Experiment with changing the sampling frequency. What is the highest frequency necessary to prevent aliasing on all of the plots?
* **Interpolation**
  + The variable “lowsample” in the workspace is a timeseries signal with a bandwidth of 10 Hz – meaning that it can be perfectly represented as a sum of sine waves that are 10 Hz or less in frequency. It is currently sampled at a rate of 20 Hz. Use the interpolation formula shown in [reference needed] to reproduce the signal at a higher sample rate of your choice. Compare this to the timeseries signal “highsample” to see if you were successful.