Intepolating a Variable Sampled Signal

Tyler Olivieri; Devin Trejo; Jacob Esworthy

***Abstract*—Creating a variable sampled signal allows for conservation of bandwidth and power however it removes the ability to accurately find a Discrete Fourier Transform (DFT). A requirement of the DFT is that the signal itself by uniformly sampled. In this paper we show how it is possible to find the DFT using interpolation. Interpolation is a process of estimating intermediate sample values from the surrounding samples resulting in a constantly sampled signal. From an interpolated version of the signal we can once again find the DFT of the original signal.**

I. INTRODUCTION

A. *Signal Construction*

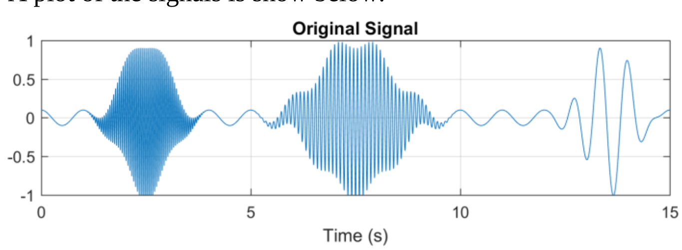
Processing a signal at a variable sampling rate allows for a lower sampling rate over portions of the signal where the instantaneous bandwidth is lower. Sampling at a variable rate requires less memory, consumes less power, and is less computationally expensive when applying a filter. These resources can be invaluable when programming embedded hardware. However, if one doesn't know the frequency of the input signal it is often safer to sample at a constant sampling rate and use an anti-aliasing filter to prevent further distortion. One might be inclined to sample at a constant rate first then change to a variable sampling rate afterwards for transmitting data where there will be less data to transmit.

This paper will explore taking a signal containing three bursts of different frequency cosines that are sampled at a constant sampling frequency above the Nyquist rate of the bandwidth of the signal. After we create a second version of the original signal using a variable sample rate.

We construct the two versions of the signal using from the following periodic signals:

$$x = 0.1\cos((2\pi)1t) + \begin{cases} \cos\big((2\pi)22t\big)\ 1 < t_{sec} < 4 \\ \cos\big((2\pi)11t\big)\ \ 5 < t_{sec} < 10 \\ \cos\big((2\pi)1.5t\big)\ \ 12 < t_{sec} < 15 \end{cases}$$

A plot of the signals is show below.



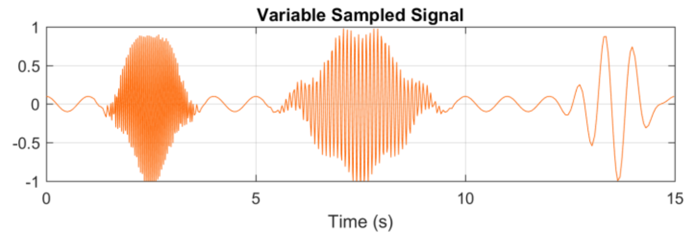


Figure 1: Original signal (top) Variable Sampled signal (bottom)

Observing the signals shows that they do not differ greatly. We do lose some resolution at the burst portions of the signal. The variable sampling frequency is still above the Nyquist rate of the instantaneous bandwidth of the signal so most of the signal content remains.

To see the difference between the two signals we use a stem plot to show how five samples of the variable signal compares to 4586 samples in the original signal. The plots below illustrate how many samples are being saved in the resampled signal while still maintaining most of the original signal definition.

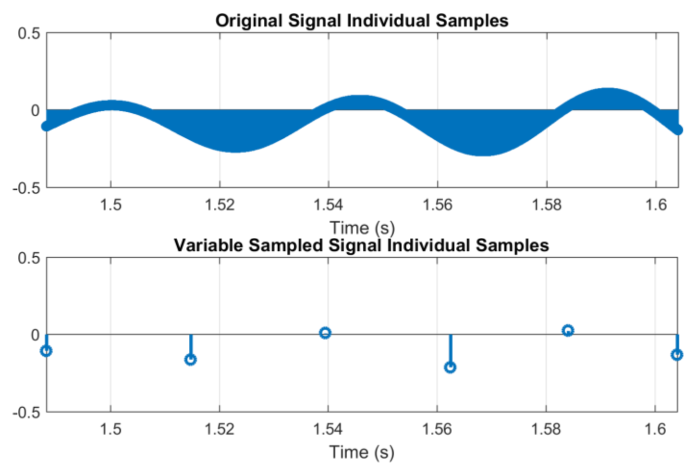


Figure 2: Original signal (top) Variable Sanmpled signal (bottom)

Looking at the period between samples we can see how the sample frequency changes from sample to sample. We will see how the varying sample rate will prevent us from finding an accurate DFT.

Variable Signal Sample (N)	Sample time (sec)	Period Between Samples (n-(n-1)	Sample Frequency (Hz)
27	1.4883	0.0286	34.96503
28	1.5148	0.0265	37.73585
29	1.5395	0.0247	40.48583
30	1.5624	0.0229	43.66812
31	1.5841	0.0217	46.08295
32	1.604	0.0199	50.25126

B. *Signal FFT*

For this project we will be looking at the frequency content of our signals. To find the frequency spectra of the discrete time signal requires the use of the Discrete Fourier Transform (DFT). We learn that the Fourier Transform and the DFT are almost the same except for we change the continuous integral to a discrete summation.

$$X[F] = \sum_{n=0}^{N-1} x[n] e^{-j\Omega kn}$$

The DFT can be taken of both signals and one would clearly find that the DFT of the varying sampled signal to be incorrect. Compare the constant sampling rate signal with the varying sampled signal:

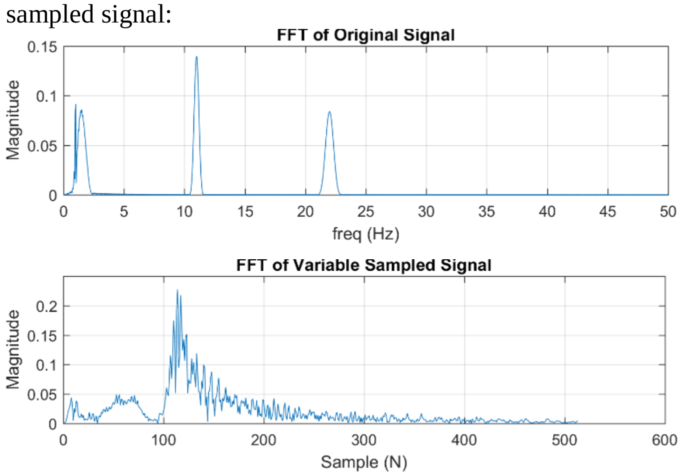


Figure 3: FFT of original (top) and resampled signal (bottom)

In the DFT of the original signal we can see the underlying frequencies that make up our signal. We have frequency content at the expected frequencies of 1, 1.5, 11, and 22 Hz. When we resample the signal, the sample rate is constantly changing and the resulting DFT is unrecognizable. A pre-requisite for taking the DFT is that the signal $x[n]$ is uniformly sampled.

This paper addresses the problems a variable sample signal introduces. In what scenarios is a uniformly sampled signal preferred over a variably sampled signal?

II. Approach

A. Interpolation

In general terms, interpolation is an estimation of a value between known values. In the context of digital signals, interpolation can be used to change the sampling rate of a signal from varying to constant as long as one can obtain two or more known data points. Interpolation can also be used to up-sample a signal that starts with a constant sampling frequency.

There are two methods of interpolation that are applicable to this project: linear interpolation and cubic spline interpolation. The main difference between the two types of interpolation is how they connect the samples of the signal.

Linear interpolation takes the derivative between two points to find a linear line equation. Imagine you have a particular interval $(x_k, x_{(k+1)})$. An equation for the line between these two points can be created with the point slope formula

$$y - x_k = \frac{(y_{k+1} - y_k)}{x_{k+1} - x_k}(x - x_k)$$

If there is a query point between $(x_k, x_{(k+1)})$ a value will be created at that query point from the equation of the linear line. The function of the line can also be represented by:

$$f = Af_k + Bf_{k+1} \qquad (1)$$

The coefficients A and B can be found by:

$$A = \frac{(x_{k+1} - x)}{x_{k+1} - x_k}$$
$$B = 1 - A$$

where x with no subscripts is the query point and the x with subscripts is the boundary points.

Spline interpolation uses a polynomial function when connecting samples of the signal, therefore giving it a smooth rounded curve. It is important that cubic spline interpolation be continuous in both the first and second derivative. However, the linear line between two points does not have a continuous second derivative. Also, the first and second derivatives are also not continuous at boundary points of two adjacent intervals. A new variable can be introduced, f_k'', that varies linearly from a value at f_k'' to f_{k+1}''. This value can be added on the right side of eq. 1.

$$f = Af_k + Bf_{k+1} + Cf_k'' + Df_{k+1}'' \qquad (2)$$

Where A and B are defined above and then coefficients C and D are found as

$$C = \frac{1}{6}(A^3 - A)(x_{k+1} - x_k)^2$$
$$D = \frac{1}{6}(B^3 - B)(x_{k+1} - x_k)^2$$

To check if f'' is the second derivative of the interpolated equation, the derivative of the generic eq. 2 is taken in respect to x, using the definitions of A, B, C, and D to compute: $\frac{dA}{dx}$, $\frac{dB}{dx}$, $\frac{dC}{dx}$ and $\frac{dD}{dx}$.

$$\frac{df}{dx} = \frac{f_{k+1} - f_k}{x_{k+1} - x_k} - \frac{3A^2 - 1}{6}(x_{k+1} - x_k)f_k'' + \frac{3B^2 - 1}{6}(x_{k+1} - x_k)f_{k+1}'' \qquad (3)$$

For cubic spline interpolation to be valid the values of f_k'' must be known, which requires eq. 3 be continuous across the boundary interval. Now x is evaluated at x_k in the interval $(x_k, x_{(k+1)})$ and rearrange the equation we can obtain

$$\frac{(x_k - x_{k-1})}{6} f_{k-1}'' + \frac{(x_{k+1} - x_{k-1})}{3} f_k'' + \frac{x_{k+1} - x_k}{6} f_{k+1}'' = \frac{f_{k+1} - f_k}{x_{k+1} - x_k} - \frac{f_k - f_{k-1}}{x_k - x_{k-1}}$$

Linear interpolation is simpler computationally but does create more rigid signals. One can see the difference between linear and cubic spline interpolation in Figure 5.

As stated above, we chose to use and compare both linear interpolation and cubic spline interpolation. Once we interpolate the signal, we now have data at a constant sampling rate. Using this new data at a constant sampling rate we can now take the Fourier transform of the signal to determine the frequency content of the signal.

III. Results

A. Interpolating the Varying Signal

Using the interpolation technique we were successful in reconstructing our variable sampled signal to a constant sampled signal. In MATLAB we performed both a linear and cubic spline interpolation to see how accurately the two methods reconstruct our original signal. In our reconstruction process we choose a relatively high sample rate to ensure we capture the original signal. Choosing a correct higher resample rate is more of a guess and check process since theoretically one does not know original signal's contents. In our case we chose a sampling rate of 8000 samples per second. 8000Hz is higher than the required Nyquist sample of criteria $2 * f_{max}$ which we know to be $2 * (22Hz) = 44Hz$. The resampled signals are shown below:

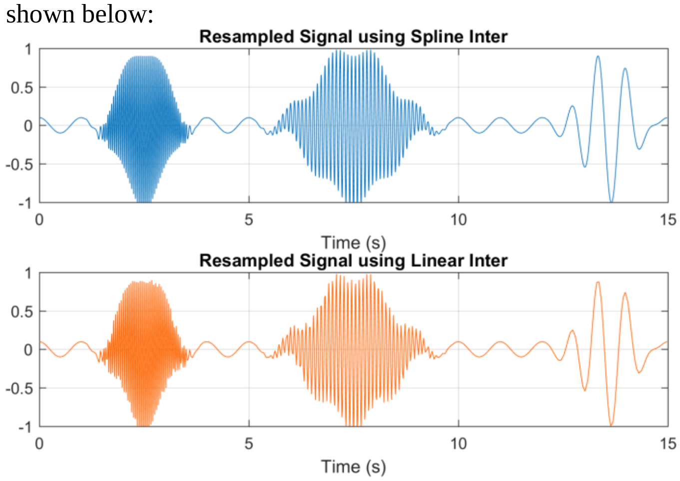


Figure 4: Linear and Spline Interpolated Signals

Comparing the interpolated signals to the variable sampled signal it appears that the interpolation did not change much. The difference is not apparent until we zoom into a portion of the signal.

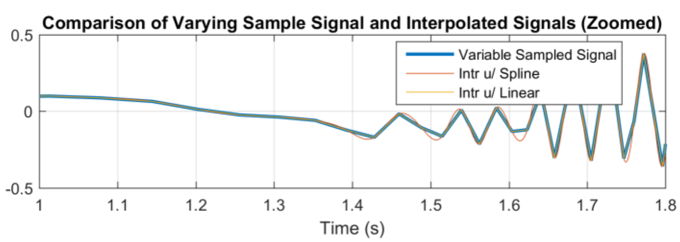


Figure 5: Sampled Signal Compared to Interpolated Signal

The spline interpolation creates the expected smooth version of the signal which we say better relates to the original signal. Meanwhile the linear interpolated signal better follows the variable sampled signal. Note that the linear interpolated signal and the spline interpolated signal contain the same number of samples.

B. DFT of the Interpolated Signal

Now that we have the signal with a varied sample rate, we can find its DFT. Again we take the DFT of both the linear and spline interpolated signals to see if one performs better than the other.

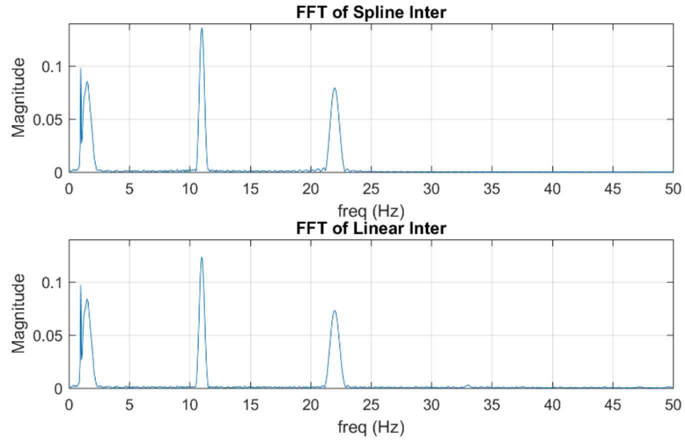


Figure 6: Discrete Fourier Transform of Interpolated Signals

Comparing the DFT of the interpolated signals (Figure 6) to the original signal (Figure 3) we see similar resemblance. The important aspects of the correct DFT are the harmonics of 1, 1.5, 11, and 22 Hz. Both the methods create accurate results.

Compare the linear interpolation technique to the spline. The spline interpreted signal DFT has less noise across the spectra since it has a smoother interpretation of the signal. Note the small resonating frequency in the linear interpolated DFT around 33Hz. The resonance is an artifact of compounding noise from the misaligned representation of the original signal. The better interpolation method depends on what type of signal you are trying to interpolate. Since our signals consisted of a sum of sine waves a spline interpolation will create a better representation. In a case where a signal contains sharp edges such as in a pulse train a linear interpolation will create the better output signal. Overall we conclude that both interpolated signals allow us to find an accurate representation of frequency spectra.

IV. Conclusion

In conclusion, the DFT assumes evenly spaced points in its calculation, which creates problems when a DFT is taken of a signal sampled at a varied rate. An easy work around is to resample the signal at a constant rate and take the DFT. We showed how using interpolation creates a clean uniformly sampled signal. We explored two types of interpolation in this paper: linear and cubic spline. The DFT of the signal after interpolated was accurate for both cases. In Figure 5 it can be seen that the cubic spline represents the true nature of the signal better but is computationally more intensive.

V. References

[1] S. Y. Liu, "Spline-Interpolation," [Online]. Available: http://www.geos.ed.ac.uk/~yliu23/docs/lect_spline.pdf. [Accessed 22 October 2015].

VI. APPENDIX

Source code can be found on Github:
https://github.com/dtrejod/myece5514/tree/master/proj1