

Analyzing and Operating on Discrete Time Signals in MATLAB®

Aidan Sharpe & Elise Heim

February 23, 2024

Abstract

This lab includes a variety of exercises to enhance knowledge and understanding of digital signal responses. The first couple of exercises involved mathematically deducing the responses of signals. Then MATLAB[®] was utilized to visually confirm calculated behavior.

1 Introduction

MATLAB[®] is a very helpful tool for visualizing different signals and better understanding their behavior. The first two tasks include modulating amplitude, and determining periodicity mathematically. Then MATLAB[®] was used to plot different signals in order to verify their periodicity, and if applicable, their period. MATLAB[®] was also used to plot discrete time signals in order to better understand their behavior. In addition, the boundedness of a signal was checked using MATLAB[®] to plot the signal. The energy of an aperiodic signal was also observed using MATLAB[®], and plotted as a function of time. Finally, this lab utilized MATLAB[®] to explore the functionality of the modulo operator.

2 Results & Discussion

2.1 Analysis of Amplitude Modulation

A message, $m(t)$, with a bandwidth, $B = 2[\text{kHz}]$ modulates a cosine carrier with a frequency of $10[\text{kHz}]$. The combined signal is $s(t) = m(t) \cos(20000\pi t)$. Using a Fourier transform on $s(t)$ reveals a maximum frequency at $12[\text{kHz}]$. In fact, as seen in figure 1, by filling the band that $m(t)$ occupies with white noise, the Fourier transform of $s(t)$ contains white noise centered on the carrier frequency with twice the bandwidth of the original signal. The spike that occurs at $10[\text{kHz}]$ is the result of the original signal having a DC term and the carrier frequency having a value of $10[\text{kHz}]$.

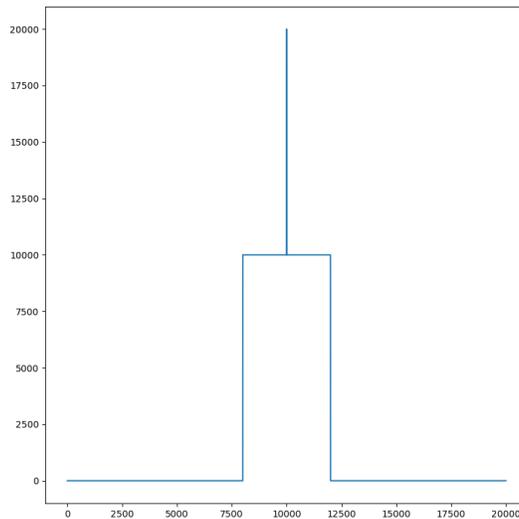


Figure 1: The Fourier transform of a white noise signal carried at $10[\text{kHz}]$

When instead, $m(t)$ had a triangular spectrum of amplitude 1, the spectrum of $s(t)$ was two triangles touching at the base at $10[\text{kHz}]$ as seen in figure 2.

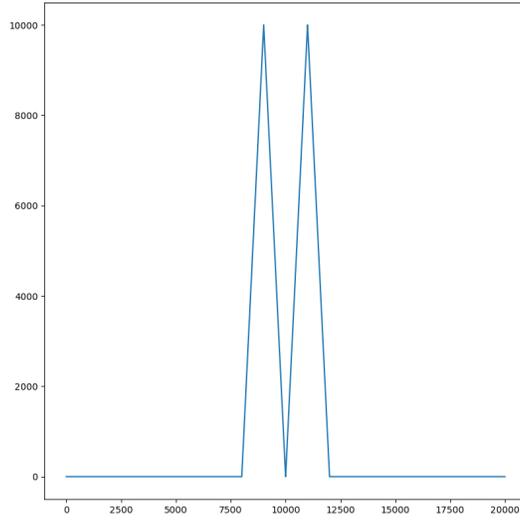


Figure 2: The Fourier transform of a signal with a triangular spectrum carried at 10[kHz]

2.2 Periodicity and Sampling Frequency

Consider the signal $x(t) = \cos(2\pi t/7)$. Given the standard forms, $\cos(2\pi ft)$ and $\cos(\omega t)$, where f is linear frequency and ω is angular frequency, $f = \frac{1}{7}$ and $\omega = \frac{2\pi}{7}$. Given sampling frequencies of 1[Hz], 2.5[Hz], 3[Hz], and π [Hz], the sampling theorem is satisfied. To determine if the sampled signal is periodic, the fundamental time period must be determined by

$$N = \frac{2\pi}{\omega} \quad (1)$$

where N is the fundamental time period and ω is the angular frequency of the signal. Given $\omega = \frac{2\pi}{7}$, the fundamental time period is 7[s]. With sampling frequencies of 1[Hz], 2.5[Hz], and 3[Hz], the discrete signal is periodic. However, with a sampling frequency of π [Hz], the discrete signal is aperiodic. Determining whether a sampled periodic signal is periodic can be done by satisfying the condition:

$$aN = bT_s \quad (2)$$

where a and b are integers, N is the fundamental time period, and T_s is the sampling period. The period of the discrete signal if the condition is satisfied is the product of a and b . Therefore, for $f_s = 1$ [Hz], there will be 7 samples per period, for $f_s = 2.5$ [Hz], there will be 35 samples per period, and for $f_s = 3$ [Hz], there will be 21 samples per period. Seen in figure 3, the periodicity and samples per period for $f_s = 1, 2.5,$ and 3 [Hz] can be determined simply by counting the samples. For $f_s = \pi$ [Hz], the aperiodic nature only becomes apparent after observing many samples.

2.3 Plotting Discrete Time Signals

MATLAB[®] can be very helpful for plotting complicated signals over specific intervals. Using the impulse, unit step, and the `stem` command, one can plot a variety of signals. One such signal is $\delta(n - 5) - 2\delta(n - 8) + 6\delta(n - 11)$ over the interval $0 < n < 12$, as shown in 4. This function comprises three impulse functions that have been shifted and scaled.

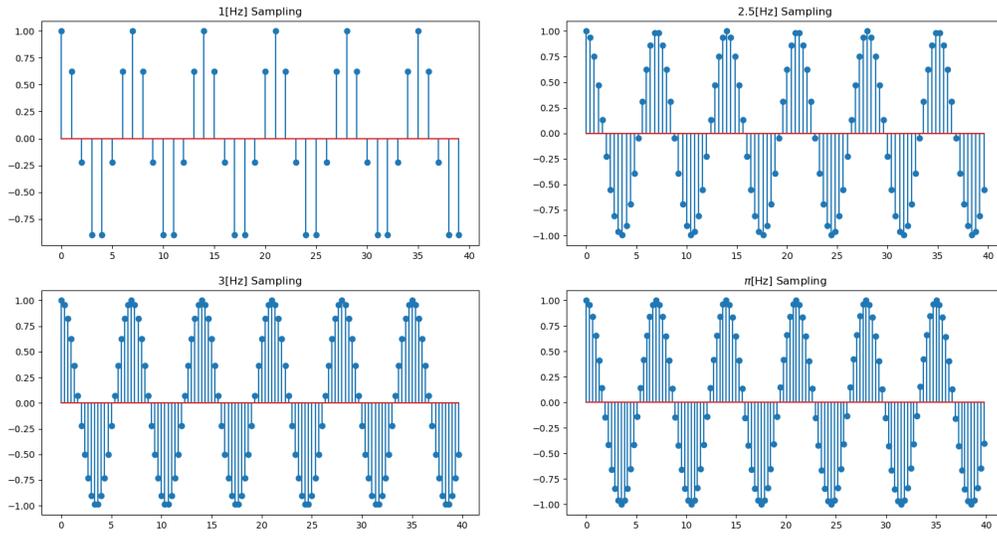


Figure 3: Sampling a periodic signal at different frequencies

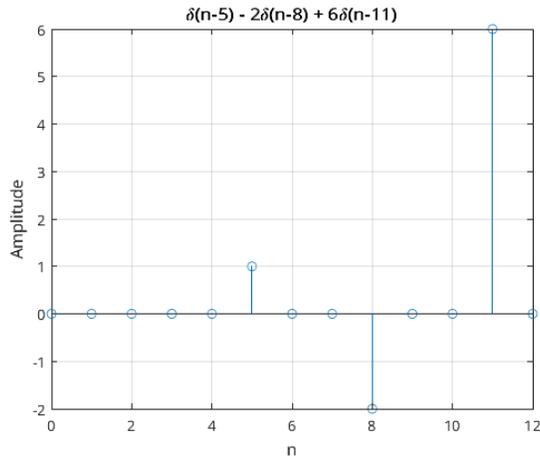


Figure 4: Shifted and scaled impulses

The signal $u(n - 3) - 2u(n - 8) + u(n - 12)$ across $0 < n < 15$ is seen in figure 5. This signal consists of three unit steps that have been shifted and scaled.

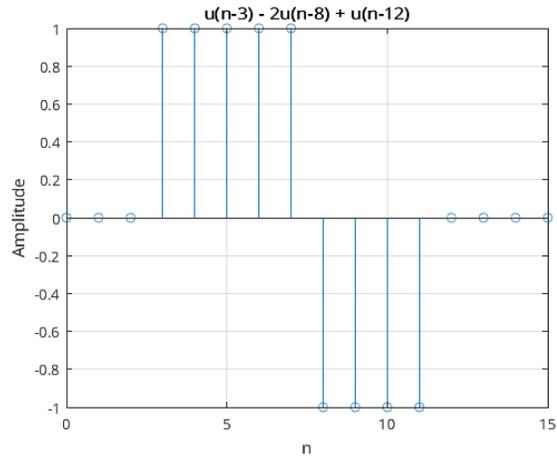


Figure 5: Shifted and scaled unit steps

Plotting the signal $nu(n)$ over the interval $0 < n < 20$ reveals the discrete ramp function seen in figure 6.

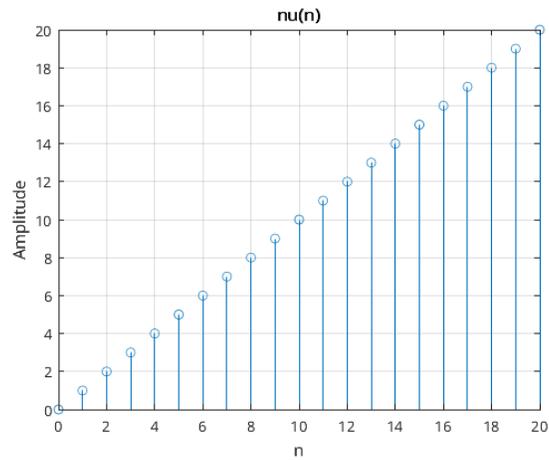


Figure 6: A discrete ramp function

Plotting the signal $n^2u(n)$ on the interval, $0 < n < 10$ reveals a quadratic regulated by the unit step seen in figure 7.

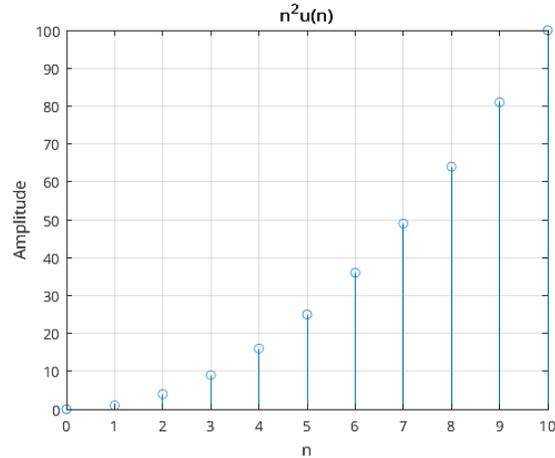


Figure 7: Discrete quadratic signal

MATLAB[®] is not limited to these simple signals. The signal $3(n-3)^2 e^{-0.3n} \sin\left(\frac{2n}{3}\right) u(n)$ was plotted from $0 < n < 30$ as seen in figure 8. This signal resembles an under-damped second order response due to the combination of exponential decay and sinusoidal terms.

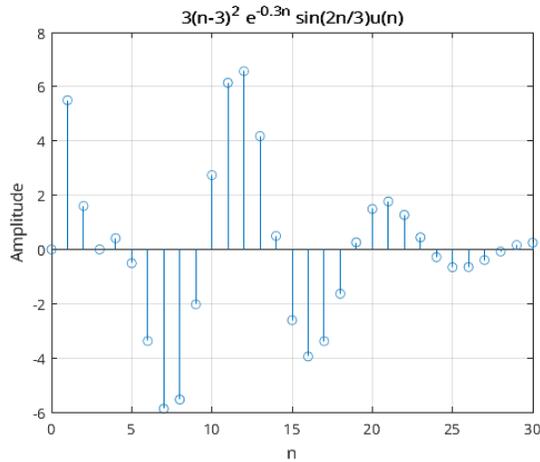


Figure 8: Oscillating exponential decay

2.4 Bounded Signals

A bounded signal has an amplitude that does not exceed a certain value. Using MATLAB[®], the signal $x(n) = 5 \cos\left(\frac{\pi}{4}n\right) u(n)$ where $a = \frac{\pi}{4}$ is verified to be bounded as clearly seen in figure 9. It is bounded since the amplitude of the signal never has an absolute value greater than 5.

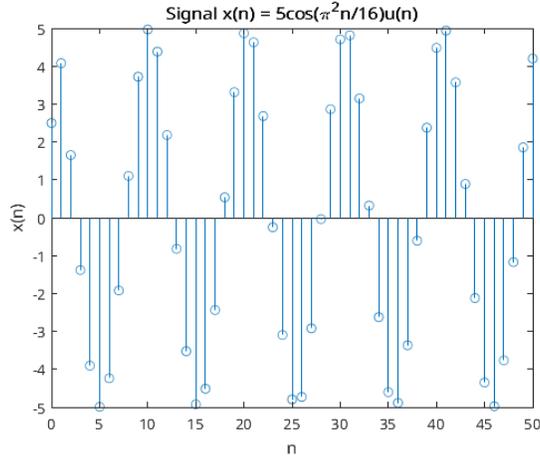


Figure 9: Boundedness of a cosine

The boundedness of $x(n) = A|b|^n u(n)$ where $A = 5$ and $b = \frac{1+j}{2}$ is seen in figure 10. Since the signal is defined only for non-negative integer values of n , and $|b| < 1$, it is bounded by $A|b|^0$, which is 5 in this case.

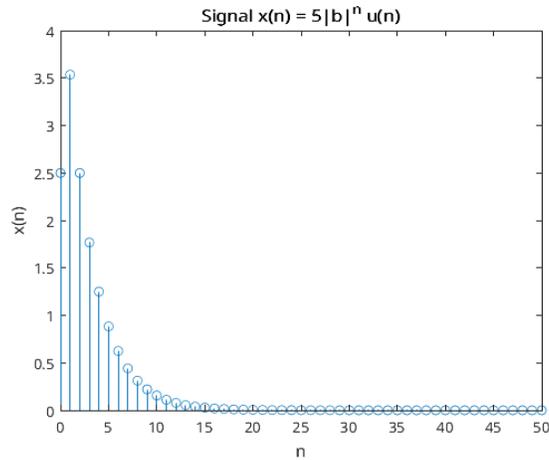


Figure 10: More complex boundedness

2.5 Energy of an Aperiodic Signal

Given a signal $x(n)$ its energy can be calculated by evaluating the discrete sum:

$$E_x = \sum_n |x[n]|^2. \quad (3)$$

Given a periodic signal, the energy will always be infinite, however, some aperiodic signals have finite energy. For example, the signal, $x[n] = \frac{1}{n}u[n-1]$ will have a total energy of $\frac{\pi^2}{6}$. This is easily seen in figure 11, where the energy, in red, approaches $\frac{\pi^2}{6}$ in only the first 100 samples.

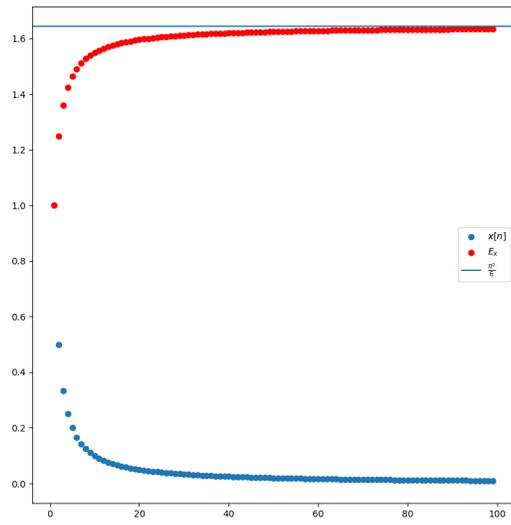


Figure 11: First 100 samples of $x[n]$ and its energy

2.6 The Modulo Operation

The modulo operation is the central operation in modular arithmetic. Essentially, the operation bundles the set of integers into a subset from zero up to, but not including the modulus by wrapping back to zero. This is similar to how a clock starts and ends at 12 o'clock, yet time remains continuous. The modulo operation is a helpful tool for manipulating the order of the terms of a discrete signal. For example, given the signal, $x[n] = [1 \ -5 \ 4 \ -8 \ 6 \ -3 \ -1]$ on the interval $0 \leq n \leq 6$, the order of the terms can easily be manipulated using 7 as the modulus. By using 7 as the modulus, it is ensured that n remains on the defined interval between 0 and 6. The signal $x[n]$ was manipulated using $x[\langle n + 3 \rangle_7]$, $x[\langle n - 4 \rangle_7]$, and $x[\langle -n + 2 \rangle_7]$, where the angle brackets indicate the modulo operation with a modulus of the subscript. The result of the modulo operation on the sampling times of $x[n]$ are seen in figure 12.

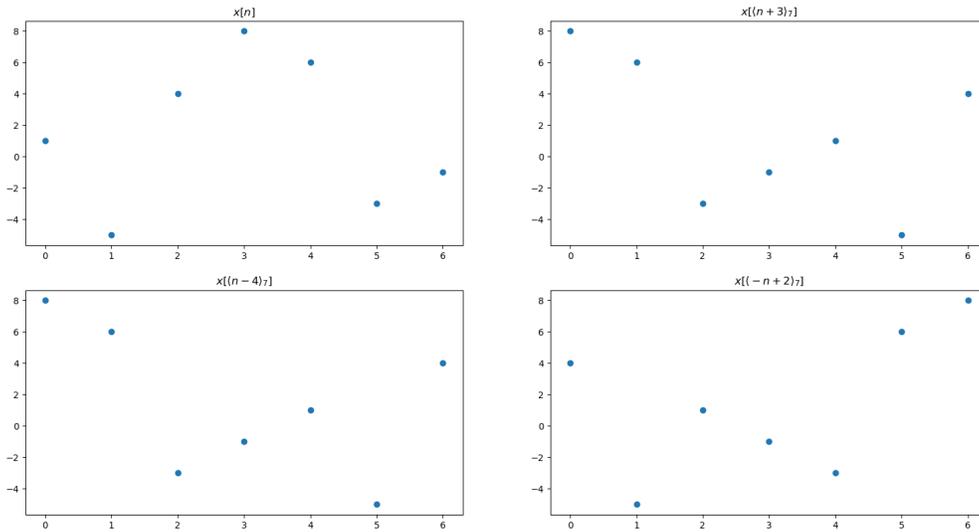


Figure 12: Applying the modulo operation to $x[n]$.

3 Conclusions

In performing various operations in isolation and in conjunction with other operations on discrete and continuous signals, the manipulation of such signals becomes more intuitive. For example, by visualizing the Fourier transform of amplitude modulated signal, it becomes clear that the bandwidth is centered around the carrier frequency and doubled. Again, by visualizing the sampling of a continuous time signal with various sampling frequencies, it the condition to produce a periodic discrete time signal becomes quite clear. The same level of intuition can be reached for boundedness, energy, and the effects of the modulo operation, simply by programatically creating a visual representation. While these simulations are obvious simplifications of reality, having a strong understanding of simple operations is crucial to understanding much more complex systems.