ECE 459 Project Report

Fall 2023

Project#1: Review of Signal Analysis

Jie Wang, Junhao Zhu, Junjie Ren, Ruiqi Zhao, Suhao Wang

Prof. Said Mikki
ZJU-UIUC Institue
Zhejiang University, Haining, Zhejiang, China
2023.10.15

1 Introduction

Our project analyzed a very simple and fundamental type of non-periodic signals, triangular pulse.

1.1 Definition of triangular pulse

A triangular pulse is a type of non-sinusoidal waveform that resembles a triangle. In the context of signals and systems, a triangular pulse usually starts at a minimum value (often zero), increases linearly to a peak value, and then decreases linearly back to the minimum value, all within a fixed period. Mathematically, it can be described using piece-wise functions. Let $x(t)$ be a triangular pulse with duration T_0 , then:

$$x(t) = \begin{cases} A(1 - \left| \frac{t}{a} \right|) & , -a < t < a \\ 0 & , otherwise \end{cases}$$

In this function:

- A is the amplitude of the pulse.
- a is any parameter that is not zero

1.2 Characteristics

- **Amplitude:** The peak value of the pulse.
- **Duration:** The width of the pulse from start to finish.
- **Symmetry:** A typical triangular pulse is symmetrical, meaning that the rise and fall times are equal.

1.3 Objective

The primary objectives of the project are:

1. **Generation and visualization** of the triangular pulse for different values of T_0 .
2. **Computation and analysis** of the Fourier Transform$X(f)$of the pulse.
3. **Discussion** on the impact of varying T_0 on the spectral shape of $X(f)$.

2 Methodology

2.1 Python Simulation

Triangular Pulse Generation A triangular pulse of duration T_0 can be mathematically defined as a piece-wise function[1]. Using Python, we can employ the **numpy** library to generate and plot the pulse for a given amplitude and T_0.

Fourier Transform Calculation Using the **numpy** library in Python, the Fourier Transform $X(f)$ of the generated triangular pulse is computed by fast Fourier Transform[2]. By varying T_0, the shape of the spectrum is observed.

2.2 Fourier Transform pair

$$x(t) = \begin{cases} 1 - \dfrac{2}{T_0}\Big| t \Big| & , -\dfrac{T_0}{2} < t < \dfrac{T_0}{2} \\ 0 & , otherwise \end{cases}$$

$$\begin{aligned}
X(w) &= \int_{-\infty}^{\infty} x(t)e^{jwt}dt \\
&= \int_{\frac{-1}{2}T_0}^{0} (1 + \frac{2}{T_0}t)e^{jwt}dt + \int_{0}^{\frac{1}{2}T_0} (1 - \frac{2}{T_0}t)e^{jwt}dt \left(set \quad a = \frac{1}{2}T_0 \right) \\
&= \frac{1 + ajw - e^{ajw}}{aw^2} + \frac{1 - ajw - e^{-ajw}}{aw^2} \\
&= a\frac{2 - e^{ajw} - e^{-ajw}}{a^2w^2} \\
&= -a\frac{e^{-ajw}\left(e^{\frac{1}{2}ajw}\left[e^{\frac{1}{2}ajw} - e^{\frac{-1}{2}ajw}\right]\right)^2}{a^2w^2} \\
&= a\left[\frac{sin(\frac{1}{2}aw)}{\frac{1}{2}aw}\right]^2 \\
&= \frac{T_0}{2}sinc^2\left[\frac{1}{4}T_0w\right]
\end{aligned}$$

2.3 Result

The larger T_0 is, the more the zeros of the function contract toward the center, and these peaks become narrower and higher.

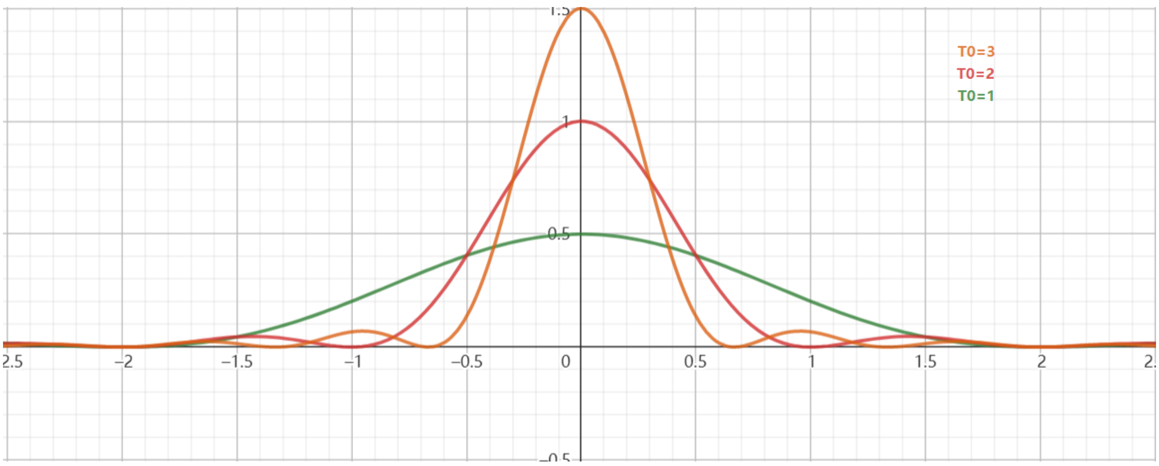

Figure 1: geogebra

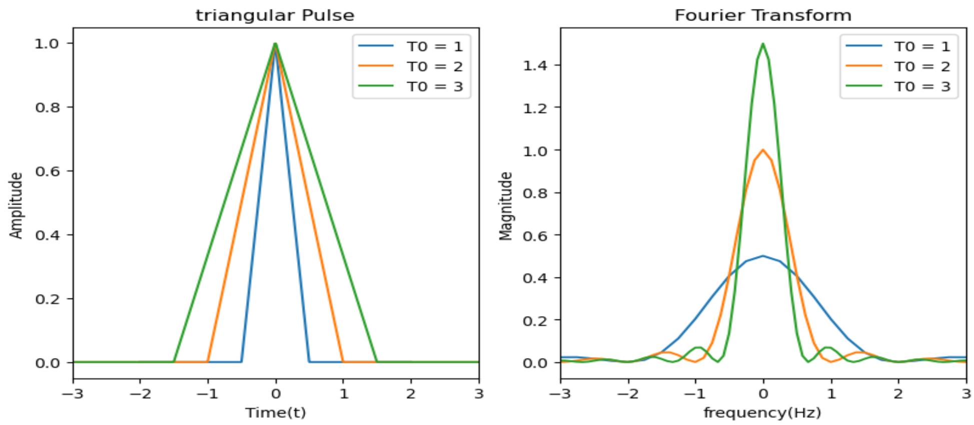

Figure 2: python

3 Discussion of the results

The results of our analysis are as follows:

3.1 Triangular Pulse Signal

The triangular pulse signal is a non-periodic waveform that resembles a triangle. The amplitude of the signal is unity within a specified duration T and falls to zero outside that range. The signal is symmetric with respect to its peak, which occurs at t = 0. As T varies, the shape and duration of the triangular pulse change significantly. Smaller values of T result in narrower and taller triangular pulses, while larger values of T create wider and shorter triangular pulses.

3.2 Fourier Transform of the Triangular Pulse

The Fourier transform of the pulse reveals the spectrum of the signal, with different T values causing variations in the spectrum's shape. A smaller T value results in a broader spectrum with higher-frequency components, while a larger T value narrows the spectrum with lower-frequency components. These observations demonstrate the fundamental relationship between the duration of the signal and its frequency content, which is a key concept in signal processing and Fourier analysis.

4 Conclusion

4.1 Summary

This project successfully demonstrates the relationship between a time-domain triangular pulse and its frequency-domain representation. The impact of the pulse duration T_0 on the Fourier spectrum showcases the fundamental principles of time-frequency duality. Based on the analysis above, we conclude that the larger T_0 is, the more the zeros of the function contract toward the center, and these peaks become narrower and higher.

4.2 Application

The conclusion of this project can be applied to various fields:

- **Signal processing:** When analyzing signals in the frequency domain, the choice of window function affects the trade-off between time and frequency precision. A narrow window in the time domain results in a broader spectrum in the frequency domain, and vice versa.
- **Communication systems:** In digital communication, the choice of pulse shaping affects the bandwidth of transmitted signals. A narrow pulse in the time domain requires a wider bandwidth in the frequency domain.
- **Medical imaging:** In medical imaging, the trade-off between time and frequency precision is crucial. MRI scans, for example, use techniques to balance the time it takes to collect data and the frequency information related to tissue properties.
- **Radar and sonar systems:** Radar and sonar systems use pulse compression techniques to obtain both time and frequency information about the target. Narrow pulses are transmitted to achieve better range resolution, while the frequency spread provides information about the target's characteristics.

5 Appendix

5.1 References

[1]Kaufmann, T., Keller, T. J., Franck, J. M., Barnes, R. P., Glaser, S. J., Martinis, J. M., & Han, S. (2013). DAC-board based X-band EPR spectrometer with arbitrary waveform control. *Journal of Magnetic Resonance*, *235*, 95-108.
[2]Brigham, E. O. (1988). *The fast Fourier transform and its applications.* Prentice-Hall, Inc..

5.2 Python Code

```
import numpy as np
import matplotlib.pyplot as plt
from numpy.fft import fft, fftfreq, fftshift

def triangular_pulse(t, T0, amplitude=1.0):
    return np.where((t >= -T0/2) & (t <= T0/2), amplitude * (1 -(2/T0)*np.
        abs(t)), 0)

def generate_fft(T0):

    length = 2*T0
    sample_number = 1000 # need to adjust to ensure sample_freq>2*1/T0
    sample_freq = sample_number/(2*length) #left axis and right axis

    t = np.linspace(-length, length, sample_number)
    x_t = triangular_pulse(t, T0)

    X_f = fft(x_t)
    f = fftfreq(len(x_t), d=1 / sample_freq)
    amp = abs(X_f)/sample_freq  # could not abs(X_f)/N
    return t,x_t,f,amp


t1,x1,f1,amp1 = generate_fft(1)
t2,x2,f2,amp2 = generate_fft(2)
t3,x3,f3,amp3 = generate_fft(3)
# draw function
plt.subplot(1, 2, 1)
plt.plot(t1, x1,label='T0 = 1')
plt.plot(t2, x2,label='T0 = 2')
```

```
plt.plot(t3, x3,label='T0 = 3')
plt.title('triangular Pulse')
plt.xlabel('Time(t)')
plt.ylabel('Amplitude')
plt.legend()
plt.xlim(-3, 3)  #adjust to show graph clearly

    # draw fft
plt.subplot(1, 2, 2)
plt.plot(fftshift(f1), fftshift(amp1),label='T0 = 1')
plt.plot(fftshift(f2), fftshift(amp2),label='T0 = 2')
plt.plot(fftshift(f3), fftshift(amp3),label='T0 = 3')
plt.title('Fourier Transform')
plt.xlabel('frequency(Hz)')
plt.ylabel('Magnitude')
plt.legend()
plt.xlim(-3, 3)  #adjust to show graph clearly
```

Listing 1: Python code

5.3 Used Functions Documentation

- **numpy.linspace():** Generates evenly spaced numbers over a specified interval.
- **numpy.where():** Returns elements based on conditional expression.
- **numpy.fft.fft():** Computes the one-dimensional n-point discrete Fourier Transform.
- **numpy.fft.fftfreq():** Returns the discrete Fourier Transform sample frequencies.

5.4 Used Application

Geogebra: Generate figure 1 and figure 2.https://www.geogebra.org/