ECE 459 Project Report

Fall 2023

Project#5: Carson's Rule

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

Prof. Said Mikki
ZJU-UIUC Institute
Zhejiang University, Haining, Zhejiang, China
Dec.7, 2023

1 Introduction

1.1 problem statement

As communication systems continue to advance, the need for effective modulation schemes becomes paramount. The project addresses the following key challenges:

1. **Bandwidth Estimation:** Accurate estimation of the bandwidth is essential for efficient spectrum utilization and signal fidelity.
2. **Theoretical vs. Numerical Analysis:** Analyzing the theoretical expectations against numerical results helps validate theoretical models and understand real-world system behaviors.

1.2 objective

The primary objectives of this project are as follows:

1. **Implement FM Signal Generation:** Develop a Python-based implementation for generating FM signals using the Direct Method, with a focus on the characteristics of the message signal.
2. **Bandwidth Estimation:** Explore numerical methods for estimating the bandwidth of both the message signal and the FM-modulated signal, comparing the results against theoretical expectations.
3. **Carson's Rule Application:** Apply Carson's Rule to theoretically estimate the bandwidth of the FM-modulated signal and compare it with the numerically estimated bandwidth.

1.3 characteristic

The project will specifically investigate the following characteristics:

1. **rect signal:** $\text{rect}(t)$ is the rectangular function, defined as:
$$\text{rect}(t) = \begin{cases} 1, & \text{if } |t| \leq \frac{T}{2} \\ 0, & \text{otherwise} \end{cases}$$
2. **Message Signal:** The project employs a rectangular pulse signal as the information-bearing message signal, simulating real-world scenarios where signals may not be ideal sinusoidal waveforms.
3. **Frequency Modulation (FM):** FM signals are generated using the Direct Method, allowing for a comprehensive analysis of the frequency deviation and sensitivity parameters.
4. **Bandwidth Analysis:** Both numerical and theoretical methods are utilized to estimate the bandwidth of the message signal and the FM-modulated signal.

1.4 Experiment Setup

The project will specifically investigate the following characteristics:

1. **Python Implementation:** The project utilizes Python programming for signal generation, analysis, and visualization.
2. **Numerical Bandwidth Estimation:** Employ numerical methods, including the Fast Fourier Transform (FFT), to estimate the bandwidth of the message and FM signals.
3. **Theoretical Bandwidth Estimation:** Apply Carson's Rule to theoretically estimate the bandwidth of the FM-modulated signal based on the frequency deviation and message signal characteristics.

2 Methodology

2.1 Generate FM Signal

2.1.1 Mathematical Representation

The mathematical representation of an FM signal can be expressed by:

$$x(t) = A\cos\left(2\pi f_c t + 2\pi k_f \int_{-\infty}^{t} m(\tau)d\tau\right) \tag{1}$$

where f_c is the carrier frequency, and the k_f is the frequency-sensitivity factor.

2.1.2 Method description

There are two main methods for generating Frequency Modulation (FM) signals: the direct method and the indirect method.

Direct Method: In the direct method, the carrier frequency is directly modified according to the message signal by using a sinusoidal oscillator. The instantaneous amplitude of the message signal determines the frequency deviation applied to the carrier. By directly altering the carrier frequency, the resulting FM signal exhibits continuous frequency changes that match the variations in the message signal. This method is commonly used in analog FM systems.

Indirect Method: The indirect method of generating FM signals involves using a mixer, multiplier, and integrator. This method is commonly used in digital FM systems for precise frequency control. In this method, the narrow-band FM signal is produced using a highly precise oscillator, ensuring accuracy in terms of the carrier frequency and modulation index. Subsequently, the narrow-band FM signal is transformed into a wide-band FM signal through the utilization of a multiplier.

In the context of Python programming, the direct method refers to the generation of a wide-band or narrow-band signal through direct integration.

2.1.3 integral achievement

The code utilizes the *cumsum* function in *numpy* to accomplish the required integration for the generation of the frequency-modulated (FM) signal.

The cumulative sum operation employed within the code effectively performs numerical integration on the message signal. It accumulates the values of the signal over time, mimicking the process of integrating the signal. The division by the sampling frequency (Fs) ensures accurate integration with respect to the sampling rate.

2.1.4 Modulator Parameters

- **carrier frequency:** $f_c = 10Hz$
- **duration of rect:** $T = 1s$
- **amplitude of carrier:** $A_c = 1$
- **frequency-sensitivity factor:** $k_f = 10$.
- **sampling rate:** $F_s = 2000Hz$. For the digital representation, it should be at least twice the maximum frequency present in the message signal to satisfy the Nyquist criterion.
- **amplitude of m(t):** $A_m = 1$.

2.1.5 result

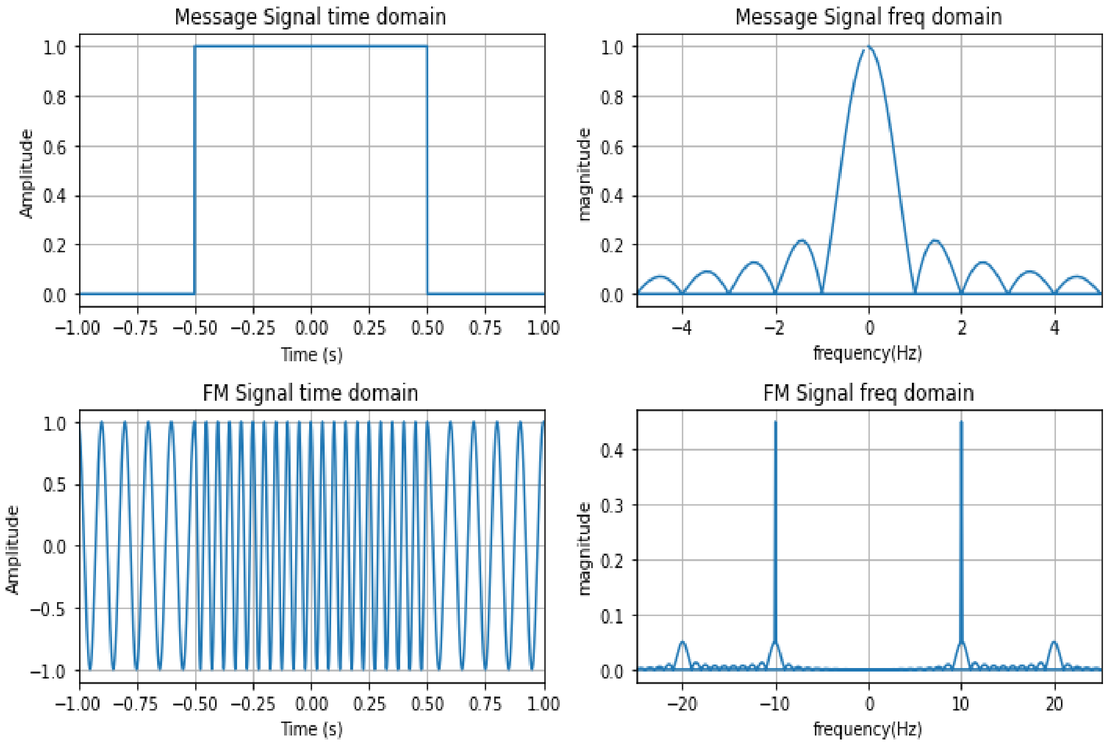

Figure 1: Generation of FM signal

2.2 bandwidth calculation

2.2.1 Numerical Approach for Bandwidth Estimation

In this study, we employed a numerical approach to estimate the bandwidth of a communication system. The algorithm involved numerical integration of the frequency spectrum obtained through the Fast Fourier Transform (FFT). The key steps of the methodology include:

1. **Initialization:**
   - Define the frequency range and spectrum of the signal.
   - Calculate the minimum step size based on the frequency range.
2. **Numerical Integration:**
   - Initialize an array to store cumulative energy values.
   - Perform numerical integration on the frequency spectrum using a loop.
   - Set a threshold for bandwidth energy.
3. **Bandwidth Calculation:**
   - Convert the index to an estimate of bandwidth, considering the step size and symmetry of the frequency spectrum.

2.2.2 Carson's Rule for Theoretical Bandwidth Estimation

To provide a benchmark for comparison, Carson's rule was applied to estimate the theoretical bandwidth. Carson's rule is expressed as:

$$B_{\text{Carson}} = 2 \cdot (\Delta f + f_m)$$

Where:

- B_{Carson} is the theoretical bandwidth.
- Δf is the frequency deviation of the modulating signal.
- f_m is the highest frequency component in the modulating signal.

2.2.3 Comparison of Numerical and Theoretical Calculation

Table 1: change T in rect(t/T)

T	Carson's Rule	Cal_Bandwidth	Error(%)
1	38.6	21.0	45.6
2	28.4	20.6	27.5
3	25.0	20.8	16.8
4	23.6	20.8	11.9
5	22.6	20.6	8.8
6	22.2	20.8	6.3
7	21.6	20.8	3.7
8	21.4	20.6	3.7
9	21.2	21.0	0.9
10	20.2	20.0	1.0
11	20.2	20.0	1.0
12	20.2	20.0	1.0
13	20.2	20.0	1.0
14	20.2	20.0	1.0
15	20.2	20.0	1.0
16	20.2	20.0	1.0
17	20.2	20.0	1.0
18	20.2	20.0	1.0
19	20.2	20.0	1.0
20	20.2	20.0	1.0
21	20.2	20.0	1.0
22	20.2	20.0	1.0
23	20.2	20.0	1.0
24	20.2	20.0	1.0
25	20.2	20.0	1.0
$\vdots$	$\vdots$	$\vdots$	$\vdots$
96	20.2	20.0	1.0
97	20.2	20.0	1.0
98	20.2	20.0	1.0
99	20.2	20.0	1.0

Carson's Rule: It can be observed that the bandwidth calculated using Carson's Rule demonstrates a decreasing trend as the duration progresses, ranging from T=1 to T=99.

Bandwidth Calculation Results: The calculated bandwidth values exhibit a relatively stable pattern, hovering around an average value of 20.0 with minor fluctuations. These values do not exhibit a clear upward or downward trend as the duration changes.

Percentage Error Analysis: When the duration T is large, the percentage error remains relatively constant at 1.0% for the majority of the time points. However, in cases where T is small, the percentage error tends to be comparatively larger.

Table 2: change kf

kf	Carson's Rule	Cal_Bandwidth	Error(%)
1	5.6	2.4	57.1
2	7.6	4.6	39.5
3	9.6	6.6	31.2
4	11.6	8.8	24.1
5	13.6	10.8	20.6
6	15.6	12.8	17.9
7	17.6	14.8	15.9
8	19.6	16.8	14.3
9	21.6	18.8	13.0
10	23.6	20.8	11.9
11	25.6	22.8	10.9
12	27.6	24.8	10.1
13	29.6	26.8	9.5
14	31.6	28.8	8.9
15	33.6	30.8	8.3
16	35.6	32.8	7.9
17	37.6	34.8	7.4
18	39.6	36.8	7.1
19	41.6	38.8	6.7
20	43.6	40.8	6.4
21	45.6	42.8	6.1
22	47.6	44.8	5.9
23	49.6	46.8	5.6
24	51.6	48.8	5.4
25	53.6	50.8	5.2
⋮	⋮	⋮	⋮
96	195.6	192.8	1.4
97	197.6	194.8	1.4
98	199.6	196.8	1.4
99	201.6	198.8	1.4

Carson's Rule: It can be observed that the bandwidth calculated using Carson's Rule demonstrates an increasing trend as the k_f progresses. This trend can be observed by analyzing the formula of FM signal.

Percentage Error Analysis: When the modulation index k_f is large, the percentage error remains relatively constant at 1.4% for the majority of the time points. However, in cases where k_f is small, the percentage error tends to be comparatively larger.

3 Discussion

3.1 choice of frequency-sensitive factor

The frequency sensitivity (k_f) determines the amount of frequency deviation in the FM signal for a given amplitude of the message signal. Selecting an optimal value for k_f ensures that the frequency variations adequately represent the desired modulation depth.

3.2 Choice of integral method

1. Trapezoidal Rule (*numpy*):

- Approximates the integral by dividing the area under the curve into trapezoids.
- *numpy* function: `numpy.trapz`
- Reasonable accuracy for smooth functions but may underestimate the integral for rapidly changing or oscillatory functions.

2. Simpson's Rule (*scipy*):

- Uses quadratic interpolation between points to compute the integral.
- *scipy* function: `scipy.integrate.simps`
- Higher accuracy than the trapezoidal rule for smooth functions by considering curvature.

Above are several prevalent numerical integration methods commonly employed in Python. However, ultimately, we have opted to employ the cumsum method from the NumPy library. The cumulative sum operation performed by `numpy.cumsum` can be considered as an approximation of the integration process for the message signal. While it may not possess the same level of accuracy or sophistication as specialized numerical integration techniques such as Simpson's rule or Gaussian quadrature, it serves as a straightforward and practical approach to obtain an integrated representation of the message signal.

3.3 bandwidth

3.3.1 Numerical vs. Theoretical Bandwidth

The results obtained from the numerical approach were compared with the theoretical estimates using Carson's rule. The following observations were made:

- **Advantages of Numerical Approach:**
  - Takes into account the actual frequency content of the signal.
  - Provides a more accurate estimate for signals with complex frequency spectra.
- **Limitations of Carson's Rule:**
  - Assumes a sinusoidal modulating signal, which may not represent real-world signals accurately.
  - May provide overestimates or underestimates for signals with non-sinusoidal components.

3.3.2 Percentage Error Analysis

The percentage error between the numerical and theoretical bandwidth estimates was calculated as:

$$\text{Percentage Error} = \frac{|\text{Numerical Bandwidth} - \text{Theoretical Bandwidth}|}{\text{Theoretical Bandwidth}} \times 100\%$$

The percentage error metric helps assess the accuracy of the numerical approach relative to the theoretical estimation.

4 Conclusion

4.1 Summary

In conclusion, the project focused on generating a frequency-modulated (FM) signal using Python. The FM signal was generated by modulating a carrier signal with a message signal. The message signal, represented as an array, was integrated using the *cumsum* function from the *numpy* library. The cumulative sum operation simulated the process of integrating the message signal, providing an approximation of the integral.

4.1.1 bandwidth

- The numerical approach provides a more flexible and accurate method for bandwidth estimation, especially for signals with complex frequency content.
- Carson's rule, while widely used, may yield less accurate results for signals with non-sinusoidal components.

4.2 Implications and Recommendations

- The choice between numerical and theoretical methods depends on the nature of the signal and the desired level of accuracy.
- Further research could explore enhancements to the numerical approach or alternative theoretical models.

4.3 Application

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

1. **FM Signal:** Frequency Modulation (FM) signals find extensive applications in various fields. They are widely utilized in telecommunications, broadcasting, wireless communication systems, radar systems, audio synthesis, and frequency synthesis [1].
2. **Direct Method of FM Signal Generation:** One prevalent approach for generating FM signals is the direct method, where the message signal directly modulates the carrier signal's frequency. This method involves integrating the message signal and using the resulting integral to vary the instantaneous frequency of the carrier signal, thereby achieving frequency modulation [2]
3. **Carson's Rule:** Carson's Rule, introduced by John R. Carson, provides an empirical formula for estimating the bandwidth of FM signals. It states that the bandwidth of an FM signal is approximately equal to twice the sum of the maximum frequency deviation and the highest frequency component present in the message signal [3]. This rule aids in determining the necessary bandwidth allocation for FM transmission systems and ensures efficient spectrum utilization.

5 Appendix

5.1 References

[1] Proakis, J. G., & Salehi, M. (2007). Communication Systems Engineering. Pearson Education.

[2] Rappaport, T. S. (2009). Wireless Communications: Principles and Practice. Prentice Hall.

[3] Carson, J. R. (1922). Factors Affecting Selectivity in Radio Signaling Systems. Proceedings of the IRE, 10(4), 57-64.

5.2 Python Realization

The experiment is recorded in the attached Jupyter notebook in an interactive way.

5.3 Used Functions Documentation

- **numpy.arange():** 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.
- **numpy.cumsum():** calculates the cumulative sum of elements in an array.

5.4 Task Separation

Task 1: FM: By Suhao Wang

1. All the task that is related to generation of FM sigmal

Task 2: bandwidth: By Junjie Ren

1. All the task that is related to the bandwidth calculation