Comparison of LUT and Taylor Series Sine Generation as Wav files

Roy P. Baldelomar, Ram P. Bruses, Denzel I. Tajo and Jerome Y. Tolibas
BSECE 4, ECE Department, Eastern Visayas State University
Tacloban City

I. Abstract

This paper aims to make a comparison between two methods of generating sinusoids in discrete systems, the Lookup table and the Taylor series approximation. Generating both as wave files then finding the MSE and RMSE, between the models and a reference sine in the time domain, and comparing their THD in the frequency domain. The Taylor series approximation provided more accurate results over the Lookup table in both the time and frequency domain.

II. Introduction

Sinusoidal wave is a special type of signal. It is preferred among other waveforms since it has its unique property which retains its wave shape even when added to another sine wave of the same frequency and arbitrary phase and magnitude. It has been said that sinusoids are arguably the most important type of signal use in any field including signal processing for it is the simplest form that can be observed in a real time system. Its development did a very high impact to science and engineering field and in digital signal processing, hence it is widely used in electronic analysis. [1] According to Fourier Series theory, any signal can be written only through sine and cosine signal, even a complex signal can be broken-down into a simple sine and cosine and mathematical analysis becomes easier. [2] All natural/analog signals are basically continuous sinusoidal signals. Fourier's discovery stated that the sinusoidal signal is a basic building block for other signals which implies that any signal can be expressed as a linear combination of sinusoids. [2] There are many uses and applications of generating a sine wave especially in the field of electronics. It plays an important role to circuit analysis by providing an approximation of circuit's input and output behavior. [1]

In generating a signal there are two circuits to be used, the analog and digital circuits. But they differ in output, digital circuits operate on digital signals while the other one is analog. [3] But in case of output signal quality using digital circuit is much more convenient. It's because analog circuit is prone to noise and signal attenuation. Nowadays most modern signal generators are based on digital since it is easier to transmit, can gives us more options, less errors to occur, less distortion in the original signal, more accurate and frequency stable. [4] But none of that would not be achieved without analog as its foundation.

III. Theoretical Background

A *sine wave/sinusoid* is one of the most basic building blocks of signals. They're easy to generate, transform, and combine, which makes them very useful for all kinds of purposes. [2] Several alternative algorithms have been developed to boost the performance of the sine-wave generation.

i. Sinusoidal Signal Using LUT

The lookup table approach symbolizes the opposite to the Taylor Expansion, in the sense that no calculations are performed in real time. The values of the sine function are pre-calculated at evenly spaced phase points in the intervals $0,2\pi$ and stored in memory during the initialization phase of the algorithm. The computation of the sine waves in real time entails approximating the input phase to one of the table indexes and retrieving the value stored at that index/address. [5] [6]

ii. Sinusoidal Signal using Taylor Series

$$\sin(\mathrm{x}) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} \pm \cdots$$

Taylor series are used as approximations of complicated functions by using polynomials. As the degree of the Taylor polynomial of function increases, the approximation becomes more and more accurate. [7] [8]

IV. Methodology

All the generation and tests were done using Python 3.7. Two methods of generation of sine would be used, however the built-in sine function (NumPy sine) of Python would also be used and would serve as reference. The comparison would be done by finding the Mean square error (MSE) and Root Mean Square Error (RMSE) in the time domain, and finding the Total harmonic distortion (THD) in the frequency domain.

i. Generation

For both methods, 1 cycle was used to generate the 55Hz wav file and 2 cycles for the 110Hz wav file. In order to make the 55Hz wave, each cycle should contain 800 data points and 400 data points per cycle for the 110Hz wave. This is because of the sampling rate used was 44100Hz which means in 1 sec the wav file would have 44100 data points. For the lookup table, 100 data points were pre calculated and stored in an array. The data points were then interpolated using numpy's *interp* function in order to attain the necessary number of data points per frequency. The 21^{st} order of the Taylor Series was selected, as it contained 2 cycles (-2pi to 2pi) and has a very high accuracy. It also had the same number of data points as the interpolated lookup table. Fig. 1 shows both methods, the interpolated LUT and the 2 cycles that was used to generate the wav file. [9]

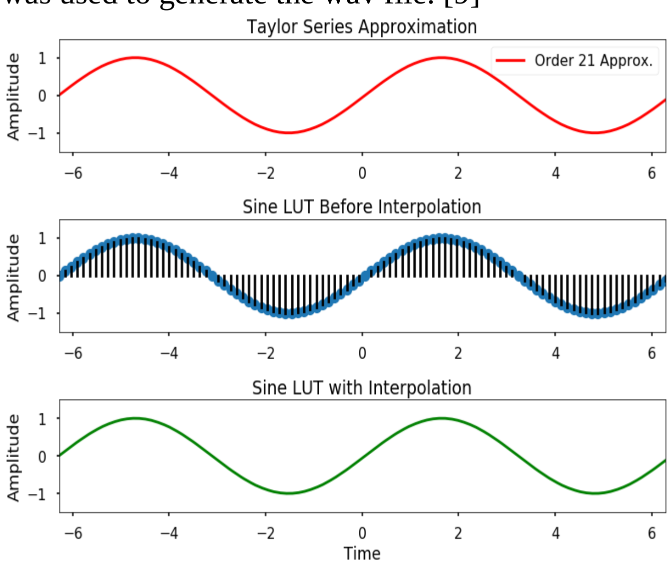

Fig. 1 Data points of the two methods, and the interpolated LUT

The *Wavio* module of python was used to generate the 16-bit PCM wav files. Each wav file has a sampling rate of 44100Hz and a duration of 1 sec. In order to get the duration of 1 sec, the cycle(s) used were stored in an array and are repeatedly appended until it reached the duration. The reference sine however didn't follow this procedure as it didn't use cycles, rather it calculated its data points corresponding to the frequency and duration. The Numpy sine would serve as an ideal generated sine wave and would be used in the comparison. Fig. 2 shows the first 2048 samples of the wav file, including numpy sine. [9]

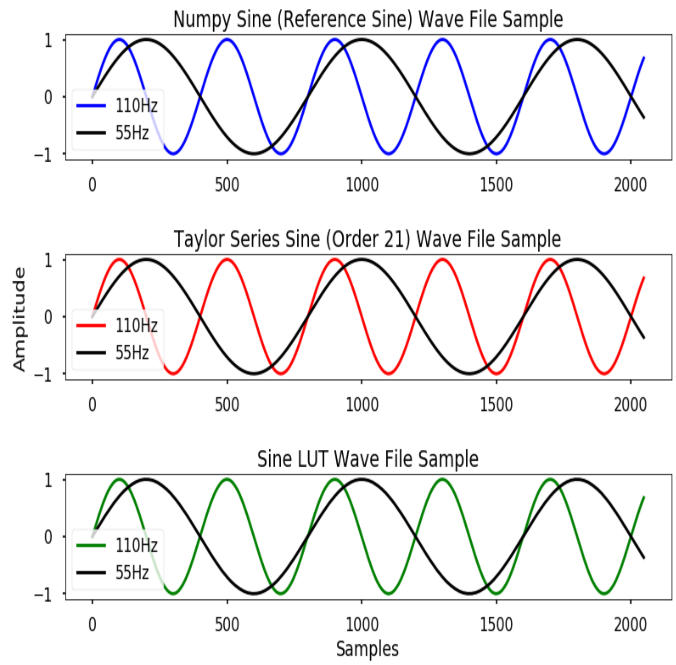

Fig. 2 Plot of wave samples, each graph displaying the 55Hz and 110Hz wave samples

ii. MSE and RMSE

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y})^2$$

Mean square error is a loss function. It is defined as the average of squares of the errors. The error is the difference between the attribute which is to be estimated (reference) and the estimator. It is a metric used to measure how accurate an estimator/model is. The MSE is the measure of the quality of the estimator, values closer to zero the better. [10] The estimators in this project are the 2 methods and the reference will be the numpy sine. The data points from the wav files will be compared to the wav file of numpy sine. RMSE is the square root of the MSE.

RMSE is the standard deviation of the errors. It is a measure of how spread out/concentrated the errors are around the line of best fit. [11]

iii. THD

$$\mathrm{THD} = \frac{\sqrt{\sum_{n=2}^{\infty} V_{n\_rms}^2}}{V_{fund\_rms}}$$

Total harmonic distortion is the ratio of all the RMS voltages of the harmonic frequencies (2nd harmonic onward) to the RMS voltage of the fundamental frequency. The THD is the percentage of all harmonic voltages present in the signal being compared with the level of the pure sine wave component. [12] For this comparison, *endolith's waveform_analysis* functions and scripts would be used to find the harmonic amplitudes. In the wav files, the amplitude of the first 14 harmonics will be computed. Fig. 3 and 4 shows the fft spectrum (Hann window) of the wave files. [9] The fundamental frequency will have the highest harmonic peak, followed by lower succeeding peaks. [12]

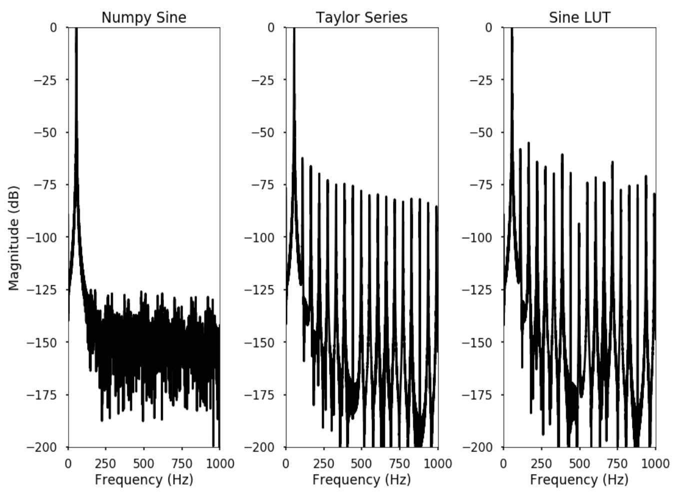

Fig. 3 The FFT spectrum of the 55hz wav files

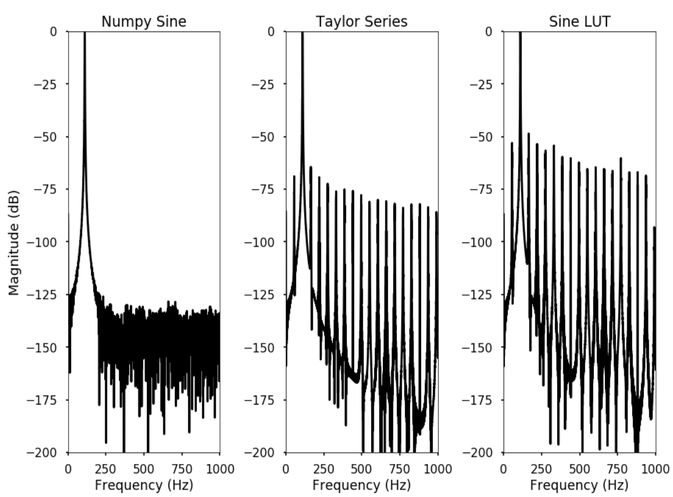

Fig. 4 The FFT spectrum of the 110hz wav files

V. Results

In the FFT plots (Fig 3 & 4), the harmonics present in both the generate wave files had greater harmonic peaks compared to the reference sine. Fig. 5 lists the first 14 harmonic peaks.

Harmonic Peaks 55Hz				Harmonic Peaks 110Hz			
No.	Taylor Series	Sine LUT	Numpy Sine	No.	Taylor Series	Sine LUT	Numpy Sine
1	22391.75	22367.35	22399.5	1	22397.22	22367.17	22399.5
2	21.365	37.413	0.003	2	6.066	37.414	0.003
3	11.495	42.589	0.002	3	3.457	42.589	0.001
4	7.851	14.876	0.003	4	2.477	14.878	0.001
5	5.95	11.688	0.001	5	1.944	11.69	0.002
6	4.791	9.669	0.002	6	1.632	9.672	0.008
7	4.007	20.714	0.001	7	1.379	20.712	0.002
8	3.455	6.927	0.002	8	1.205	6.93	0.006
9	3.026	0.479	0.004	9	1.077	0.47	0.002
10	2.703	5.77	0.002	10	0.965	5.774	0.001
11	2.431	6.318	0.003	11	0.888	6.321	0.001
12	2.21	4.858	0.002	12	0.801	4.863	0.009
13	2.043	15.419	0.003	13	0.743	15.41	0.003
14	1.9	3.835	0.001	14	0.701	3.84	0.001

Fig. 5 Harmonic peaks present in the wave files

From the table, the sine LUT had higher harmonic peaks compared to the Taylor series (Order 21) in both the frequency generated. The THD was then computed using its formula. Fig. 6 shows the THD, MSE and RMSE of the wav files.

	55Hz			110Hz		
	Taylor Series	Lookup Table	Numpy Sine	Taylor Series	Lookup Table	Numpy Sine
MSE	4.00E-06	1.47E-05		2.42E-06	2.83E-05	
RMSE	2.00E-03	3.84E-03		1.56E-03	5.32E-03	
THD	0.3270310%	0.8072220%	0.0001260%	0.1041800%	0.807269%	0.0001780%

Fig. 6 Table of Results

The Taylor series scored high in the MSE and RMSE in both frequencies compared to the LUT. Both methods produced substantial amplitudes of harmonic, making them have higher values of THD compared to the reference sine.

VI. Conclusion

The Taylor series approximate produced better accuracy in the time domain and lower THD compared to the Lookup table. Despite this, the lookup table still provides an easy and fast method in generating sine wave. Based on the results, it doesn't stray too far from the Taylor series approximation.

VII. Acknowledgement

Special thanks to endolith (https://github.com/endolith) for the signal processing functions and scripts, to Warren Weckesser (https://github.com/WarrenWeckesser) for the wavio module and to our instructor Engr. Ryan Reas for guiding and introducing us to signal processing.

VIII. References

[1] P. John M. Santiago Jr., "Chapter 11: Making Waves with Funky Functions," in *Circuit Analysis For Dummies*, John Wiley & Sons, Inc., 2013, pp. 175-191.

[2] R. N. Bracewell, "Chapter 2 Groundwork," in *The Fourier Transform and Its Applications 3rd Ed.*, McGraw-Hill International Editions, 2000, pp. 5-18.

[3] L. E. F. Jr., "Types of Electronic Communication," in *Principles of Electronic Communication Systems 4th Ed.*, McGraw-Hill Education, 2016, pp. 6-8.

[4] L. E. F. Jr., "Digital Transmission of Data," in *Principles of Electronic Communication Systems 4th Ed.*, McGraw-Hill Education, 2016, pp. 193-194.

[5] V. Skanda, "Sine Wave Generator Using Numerically Controlled Oscillator Module," Microchip Technology Inc., 2013.

[6] J. Vankka, "Methods of Mapping from Phase to Sine Amplitude in Digital Synthesis," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. Vol. 44, no. No. 2, pp. 526-534, March 1997.

[7] G. P. a. M. Kampik, "Low-Spur Numerically Controlled Oscillator Using Taylor Series Approximation," in *XI International PhD Workshop*, Gliwice, Poland, October 2009.

[8] M. J. a. A. Alfalou, "Direct Digital Frequency Synthesizer with CORDIC Algorithm and Taylor," *European Journal of Scientific Research, EuroJournals*, vol. 30, no. 4, pp. 542-553, 2009.

[9] J. Glover and V. a. T. J. Lazzarini, "Python for Audio Signal Processing," The Sound and Digital Music Research Group, National University of Ireland, Maynooth, Ireland.

[10] A. H. Murphy, "Skill Scores Based on the Mean Square Error and Their Relationships to the Correlation Coefficient," *Monthly Weather Review*, vol. Volume 116, pp. 2417-2424, December 1988.

[11] "Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature," Copernicus Publications on behalf of the European Geosciences Union., College Park, MD 20740, USA, 2014.

[12] Microsemi, "Basic Total Harmonic Distortion (THD) Measurement," Microsemi Corporation, January 2015.