Signals and Systems Laboratory Transformation of signals

Sebastian Lizarazo, Robinson Campo, and Kevin Muñoz
salizarazo@uninorte.edu.co, rlcampo@uninorte.edu.co, akmunoz@uninorte.edu.co

***Abstract*—This report presents the development of a computational program, made in Python, with the purpose of implementing the different topics given in the course on signal transformation. The transformation of signals in the eight different waves is presented, and the different methods of transformations that are performed.**
We start with the identification of each signal requested, that is, the way the wave of each signal is seen, followed by the function that gives as a specific description of each signal (the function that defines the type of signal), now with this we have its mathematical definition. All signals have a similar displacement and scaling since they share an almost equal wave transfer, this means that a pulse signal has a displacement similar to that of a square signal due to its analogous properties.

***Index Terms*—Matlab; Signal transformation; continuous; discrete.**

I. Introduction

THIS report presents the transformation of signals using the software of Python, prior to this, each of the signals to transform are identified. The classification of each signal depends on its parameters determined over time, these parameters are defined in whether the signal is continuous or discrete, if the signal tends to be periodic or not, or if it is simply an even or odd signal, each of these parameters detail the type of signal that would be simulated.
Now with a previous knowledge of how to identify each signal, it is passed concretely to the code created to analyze the transformation of signals for 8 different types of waves, some with kinship in terms of the origin of its form, the 8 types of signals can be shifted to see the behavior of the wave with a specific frequency and amplitude in a given time interval, this will also apply to the analysis of the scalar transformation.

September 15^{th}, 2022

II. Experimental framework and Results

A. Definition of variables and signals

To begin with, the variables required in the signals were defined. Then the functions are defined, such as the sinusoidal signal: y(t) = A·sin(2·pi·f·t).

Shifting: A signal is shifted when the variable ”t” in the function is accompanied by an addition or subtraction operation, mathematically defined as:

$$\text{Signal delay} = x(t - t_0)$$
$$\text{Signal Advance} = x(t + t_0)$$

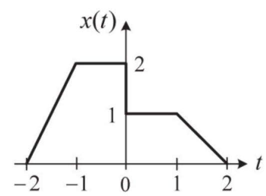

Figure 1. Representation of the original signal

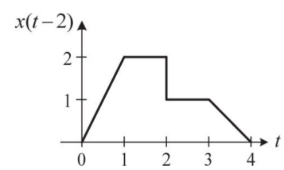

Figure 2. Representation of the shifted signal

Escalation: A signal is scaled when it is expanded or compressed, in the function if the variable ”t” is altered by a factor a, it means x(a*t) , this is explained as follows; if a ¿ 1 it is compressed and if a ¡ 1 it is expanded by that factor.

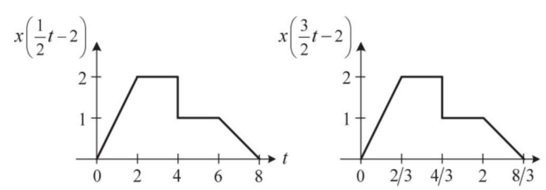

Figure 3. Representation of the original and escalated signal

B. User interface

In this part a friendly interface is created so that the user is able to choose which of the functions to transform, then there is another interface to choose the method to be developed and allows you to enter the various parameters to be used in the development.

C. Sine signal plotting

A sine wave is the definition of the mathematical sine function, the qualities of this wave are that the termination of the sine wave in the graph depends on its amplitude, this is justified with this formula:

y(t) = A·sin(2·pi·f·t)
The displacement of this wave is analogous to a continuous signal since the sine function is continuous and periodic, and also the scaling of this signal is directly associated with the amplitude and frequency at which you want to expand or compress the signal: For the realisation of the sinusoidal signal, the "sin" function was used together with the "numpy" library, delimiting its parameters by means of input variables, which are chosen by the user. For this signal there were no major problems to solve because it is a commonly used signal and pyton allows us to make optimal use of it. However, one of the obstacles that were presented in the creation of this signal is that when placing the variables of the signal very large it was difficult to see the transformation of the signal and the signal itself, so we proceeded to create a condition that made an adjustment to the display of the signal in order to observe this and its transformation in a good way.

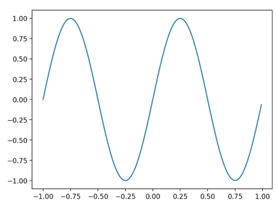

Figure 4. Representation of the snoidal signal

D. Pulse signal plotting

A pulse wave is a non-periodic, one-shot signal, it behaves as a rectangular signal and its function is analogous to that of a unitary step:

For the creation of this signal, the "arange" function was used together with the "numpy" library again. In this signal the user is the one who enters the values of amplitude and width of the signal.

An obstacle that arose in the creation of this signal and its transformation was the fact that the end user could visualise it correctly regardless of the parameters set in width and amplitude, i.e., that no matter how large these were, the entire signal could be visualised without any problem. To solve this, we proceeded to create a conditional which would adjust the signal in its axes to visualise it completely.

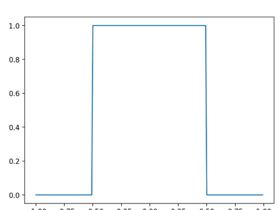

Figure 5. Representation of the pulse signal

The function of this signal is: y(t) = U(t+1) - U(t-1)

E. Quadratic signal plotting

The quadratic wave is characterized as a parabola, defined by the function: $y(t) = at^2 + bt + c$. On displacement these signals will move to the right or left depending on the sign that is adding or subtracting to "t".

Figure 6. Representation of the quadratic signal

The creation of this signal it is used the equation stipulated in the guide, in which the user must enter the desired values to be plotted. It is worth mentioning that in the creation of this signal there were no problems to be mentioned as they did not arise.

F. Exponential signal plotting

An exponential wave is commonly defined as a graph that has an upward or downward termination depending on whether "b" is positive or negative: y(t) = Ae^{-bt}

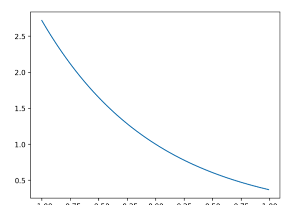

Figure 7. Representation of the exponential signal

For the creation of this signal, the "exp" function was used together with the always reliable "numpy" library, which in turn requires the input of the amplitude variable "A" and the exponent constant "b". In addition, it is worth mentioning that there were no problems in the creation of the signal in question and it was successfully created without any setbacks.

G. Lineal signal plotting

In the creation of this signal we simply applied the equation stipulated in the laboratory guide together with the constants that the user had to enter (m and b), in addition we used the libraries mentioned above, then we proceeded to apply the time scaling and time shifting necessary in the code in question. Fortunately, there were no problems to solve in the creation and processing of this signal either.

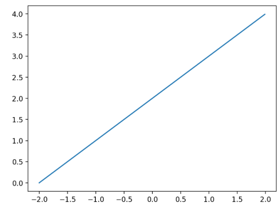

Figure 8. Representation of the lineal signal

H. Triangular signal plotting

It is characterized as a periodic wave with a symmetrical slew rate in most of the functions, it has a kinship with the sinusoidal signal for its low harmonic content, the behavior of the wave is known as ramp:

$$x(t) = \sum_{i=0}^{N} (-1)^i n^{-2} (sin[nt])$$

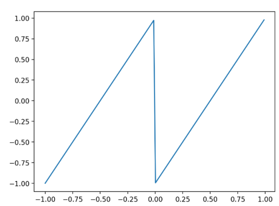

Figure 9. Representation of the triangular signal

The creation and processing of this signal used the ”SAW-TOOTH” function together with the ”numpy” library, which allow us to create the triangular signal without any problem since Python has this functionality by default and saves time and errors, in short, there were no problems to solve in this signal.

I. Square signal plotting

In a series of harmonics a square signal is the harmonic increment of a sine wave, the behavior of the square wave is similar to that of a voltage rising and falling at regular intervals.

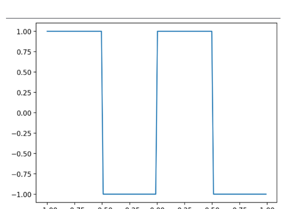

Figure 10. Representation of the square signal

In the creation of the square signal, the ”square” function was used, which allowed to perform both the creation and the processing of this signal optimally and without errors, which allows us to say that the processes with this signal were successfully carried out.

J. Pulse sequence signal plotting

This type of signal is described as an infinitely short pulse in time which tends to establish its area, a pulse sequence is a signal that has a peak of infinitely high amplitude in an extremely small time, this type of waves is defined by the Dirac delta function:

$$x(n) = \sum_{k} x[k]\delta[n-k]$$

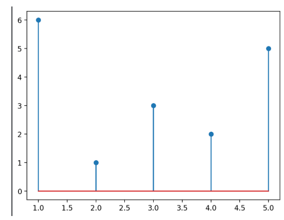

Figure 11. Representation of the sequence of pulses signal

Lo que sigue del procedimiento es la repetición del mismo proceso relatado anteriormente para cada señal discreta.

Para Mayor profundización en la lógica establecida en el código, referirse al archivo `Florez_Camargo_Muñoz_Lab1.m`

III. Conclusions

In this laboratory, eitgh signals were create in a graphic interface, each one of them with their specific inputs. The axis of the periodic signals was create based in their period, wich was obteined with the frequency, and the non periodic ones axis was created with their specific inputs.

References

[1] J. Tello *Introducción a las señales y sistemas*, 1er ed. Barranquilla, Colombia: Universidad del Norte, 2017.
[2] F. Zapata. ”Onda senoidal: características, partes, cálculo, ejemplos”. lifeder. https://www.lifeder.com/onda-senoidal/
[3] ”Señal Exponencial”. Universidad Nacional Experimental del Táchira. http://www.unet.edu.ve/aula10c/Asenales/Unid01/terce09.htm: :text=Estas
[4] B. Contreras. ”¿Cuáles son los tipos de señales de onda? – Blog Logicbus”. Blog Logicbus. https://www.logicbus.com.mx/blog/cuales-son-los-tipos-de-senales-de-onda/