



UNIVERSIDAD DEL NORTE

DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA Y ELECTRÓNICA

Asignatura: Señales y Sistemas.

Docente: Carlos Cárdenas.

Lab 1: Generación y operación de señales en Matlab y Python

Jessir Florez, Mateo Muñoz, Dylan Abuchaibe

{jessirf, mjnova, bastod} @uninorte.edu.co

15 de septiembre del 2022

Abstract- The purpose of this first lab is to replicate everything learned in classes about signal creation and its operations. We can represent the signals through mathematical expressions with independent variables

1. INTRODUCCIÓN

El siguiente informe tiene como objetivo presentar el proceso que se llevó a cabo para desarrollar un software que permita afianzar de manera práctica los conocimientos adquiridos sobre operaciones con señales, temática abordada durante las primeras unidades del curso de señales y sistemas.

Inicialmente se explicarán los requerimientos y conceptos del laboratorio, los cuales han sido estudiados durante varias semanas durante clases. Se hablará también de las diferentes herramientas y softwares usados para la realización del algoritmo. El programa debe permitir al usuario seleccionar distintos tipos de señales estudiados en clase, debe permitir también la edición de las propiedades de cada señal, así como la opción de realizar operaciones de transformación sobre estas mismas.

2. PROCEDIMIENTOS

2.1 Actividad 1.

La actividad 1 consistió en crear un programa interactivo, en el cual el usuario pudiera crear a través de una lista de opciones sus propias

señales mientras él mismo va colocando las variables que considere.

Inicialmente y ya que el laboratorio se desarrolló en Python, se instalaron previamente algunas librerías como:

La librería **Streamlit** se utilizó para generar una interfaz gráfica (Figura 1) en línea que permita seleccionar y generar la señal con la que se desea operar, así como una serie de entradas a través de las cuales se puedan cambiar las propiedades de la señal seleccionada, y finalmente, permitir seleccionar una serie de transformaciones a aplicar.

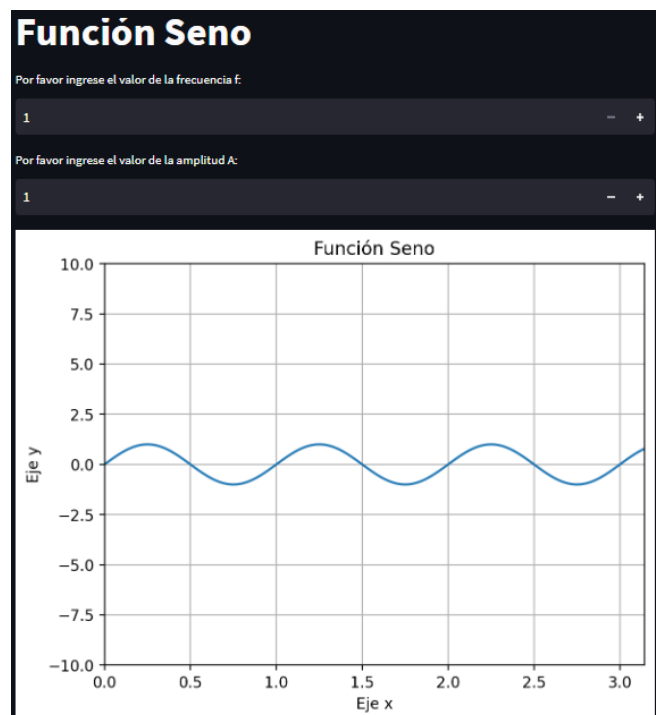


Figura 1. Interfaz gráfica generada con Streamlit

Para la selección de la señal que se desea generar, se implementó un menú desplegable con cada una de las 8 opciones disponibles. Una vez seleccionada la señal, se crea una ventana con las opciones para cambiar las propiedades de la señal, dependiendo del tipo seleccionado. En Python, para el proceso de selección de las señales, se utilizaron condicionales simples que actúan dependiendo de una variable encargada de almacenar la elección del usuario (opción).

A continuación, se explicará el procedimiento que se siguió la generación de cada tipo de onda:

1. **Señal senoidal:** para esta señal, se le pide al usuario la frecuencia(f) y la amplitud para la onda. Se genera un vector para X que tendrá valores que irán desde 0 a $2/f$. Luego se llena otro vector para el eje y con valores generados a partir de pasar el vector x por una función senoidal. Dado que la función solo se desplaza en Y de -1 a 1, se multiplica el vector por la amplitud para obtener finalmente la señal con los valores de amplitud deseados.
2. **Señal Pulso:** Para la señal pulso, el usuario debe ingresar los valores deseados de amplitud y ancho del pulso. Para este caso, se utilizó la librería *scikit-dsp-comm*, usada para el procesamiento de señales. De esta librería se usó el módulo *sigsys.py*. Por defecto, el pulso generado tiene una amplitud de 1, por lo que nuevamente se multiplica el vector del eje Y por la amplitud deseada.
3. **Señal cuadrática:** La generación de la onda cuadrática es mucho más intuitiva para este caso. Primero se deben ingresar los valores a, b y c deseados. Para este caso, el vector generado para el eje X va simplemente de -10 a 10 con un paso de 0,1. Luego se generan directamente los valores para el eje Y con la ecuación y los valores previamente recolectados.
4. **Señal exponencial:** para esta señal solamente se requieren de dos entradas (A y B). Para el eje X se genera un vector que

va de 0 a 2π , y para el eje Y se generan los valores a partir de la ecuación:

$$y(t) = Ae^{-bt}$$

5. **Señal lineal:** para este caso también se requiere solo de dos variables (m y b), un vector para x que va de 0 a 10, y posteriormente se pasan estos valores por la ecuación

$$y(t) = mt + b$$

6. **Señal triangular:** para esta señal, se le solicita al usuario los valores deseados para la amplitud y la frecuencia de la onda. El vector para X en este caso va desde 0 hasta $2/f$, lo cual permite visualizar $3/2$ de la onda generada. El vector generado se llena con un paso de 0,1. Para el eje Y se llena un vector utilizando el comando Square.
7. **Señal cuadrada:** para esta señal, se utilizó como ayuda el comando *square* para poder generar una señal cuadrada.
8. **Secuencia de impulsos:** para esta sección se debía tener en cuenta el tamaño del vector del tiempo. El usuario ingresará los valores de impulso para la generación de una señal discreta.

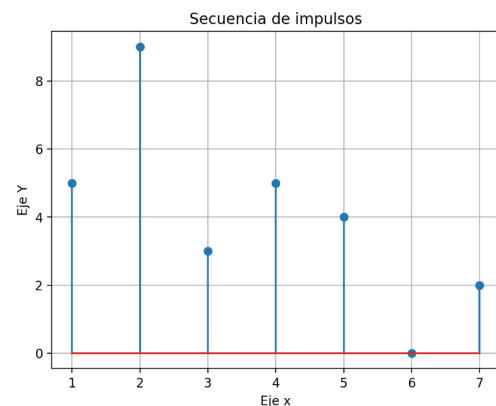


Figura 2. Secuencia de impulsos generada

2.2 Actividad 2.

Para la segunda parte de este laboratorio, se

pidió que dentro de la misma interfaz gráfica el usuario pueda hacer operaciones con las señales introducidas, tales como desplazamiento y escalamiento. El usuario podrá tomar el tipo de forma que quiere generar y también podrá elegir el tipo de operación a realizar.

Para cada uno de los casos podrá ver si se escala en el tiempo, cómo se comprime o expande la señal. Para escalamiento de amplitud, cómo se atenúa o se amplifica la señal original. Y para el desplazamiento en el tiempo se mostrará el efecto de desplazamiento ya sea para la derecha o la izquierda.

3. ANÁLISIS DE RESULTADOS

Para la primera parte del laboratorio se logró de manera satisfactoria realizar una interfaz gráfica de generación de señales, el cual se puede evidenciar en la **Figura 1** y **Figura 3**, ya que se obtuvo la gráfica solicitada, así como los valores de amplitud y frecuencia suministrados .

En segunda parte del laboratorio se llegó de igual manera al objetivo propuesto el cual era transformar las señales generadas anteriormente.

En la **Figura 4** podemos evidenciar el resultado del “desplazamiento” a través del tiempo generado en un señal triangular, en la **Figura 5** se analiza el resultado de realizar la transformación “Escalamiento en amplitud” sobre una señal cuadrada. Finalmente en la **Figura 6** se logra evidenciar el resultado del “Escalamiento en el tiempo” de una secuencia de pulsos, cumpliendo así satisfactoriamente con las transformaciones solicitadas en la guía de laboratorio.

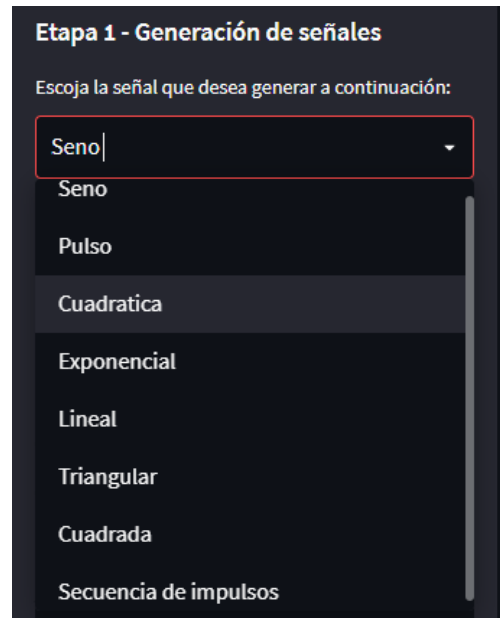


Figura 3. Seleccionador de señales

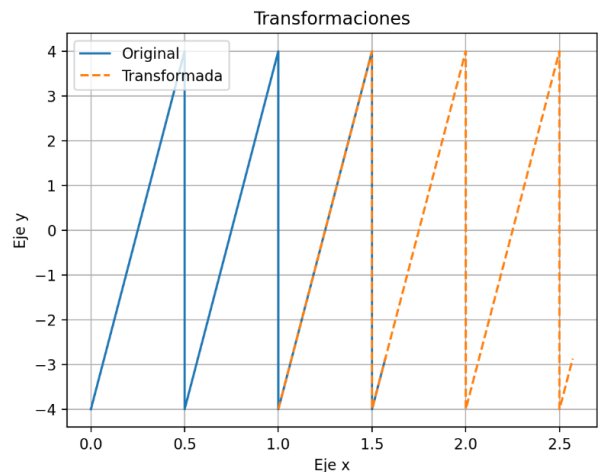


Figura 4. Desplazamiento en el tiempo de una señal triangular

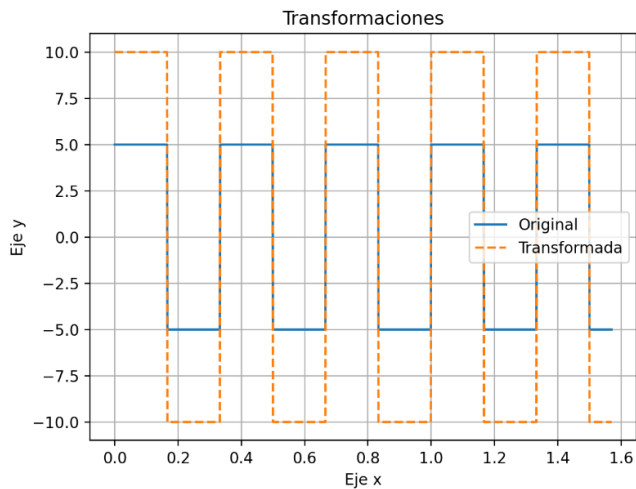


Figura 5. Escalamiento en amplitud de una función

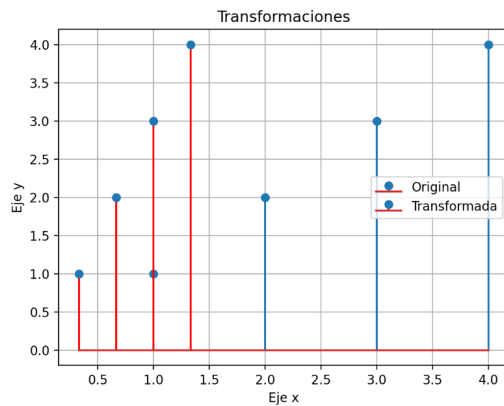


Figura 6. Escalamiento en el tiempo de una señal

4. CONCLUSIONES

En este laboratorio pudimos observar el comportamiento de distintas señales generadas a partir de una lista de opciones y cómo se podía trabajar con ellas con diferentes operaciones empleando diferentes funciones y métodos de empleo en la plataforma de Python.

Se aprendió a poder tener una mejor resolución de problemas ante los diferentes errores que se iban presentando en la creación del código. Llegamos a un nivel más avanzado al siguiente laboratorio ya que tenemos esta experiencia para poder aplicarla en un software distinto como lo es Matlab.

5. REFERENCIAS

[1] J. Tello. Introducción a señales y Sistemas, Barranquilla, Colombia: Universidad del Norte, 2017.