



Lab 1: Generación y Operación de Señales con MATLAB

Grupo: Sebastián Fernández, Carlos Rodriguez, Sergio Rodriguez
{sfernandeza, rodriguezmandoza, rodriguezaz} @uninorte.edu.co

15 de Septiembre de 2022

Resumen. Se presenta la propuesta de laboratorio en la que se orienta al estudiante en la generación y operación de señales básicas. Se aplican conceptos teóricos de escalamiento y desplazamiento de señales usando el software MATLAB.

1. INTRODUCCIÓN

Una señal se conoce como la función de una o mas variables que representan una cantidad física. Estas señales generalmente contienen algún tipo de información acerca algo estipulado, ya sea una medición de un instrumento eléctrico o el ciclo cardíaco de un humano. Existen varias maneras de identificar señales que se encuentran presentes en la vida cotidiana.

También existen diferentes tipos de señales. Tales como las continuas, discretas, las analógica. Existen clasificaciones de señales como las periódicas, las determinísticas, los pares e impares, etc.

La practica a realizar consiste en realizar operaciones básicas considerando diferentes tipos de señales. Esto se debe hacer utilizando una interfaz gráfica de usuario (GUI), en la que de manera interactiva, el usuario de manera libre pueda escoger que tipo de señal quiere con cualquier tipo de operación. Este laboratorio será realizado en MATLAB, ya que de antemano se acordó en utilizar esta herramienta; haciendo uso del “App Designer” que viene incluido en la aplicación.

2. OBJETIVO

- Desarrollar e implementar algoritmos para generar, desplazar y escalar señales usando MATLAB.

3. PROCEDIMIENTOS

El laboratorio consiste de dos partes. Para la primera parte del laboratorio se requiere generar diferentes tipos de señales. Estas señales son:

- Señal senoidal.
- Señal pulso.
- Señal cuadrática.
- Señal exponencial.
- Señal lineal.
- Señal triangular.
- Señal cuadrada.
- Secuencia de impulsos.

Recordemos que debemos hacer un GUI para que el usuario tenga la libertad de escoger los parámetros que esta persona quiera. Esto también incluye los valores que deben tomar estas señales. Con base a esto, entonces podemos decir que para la señal senoidal se requiere que el usuario ingrese los valores de la frecuencia y amplitud, y ademas la ecuación a graficar es $y(t)=A\sin(2\pi f t)$.

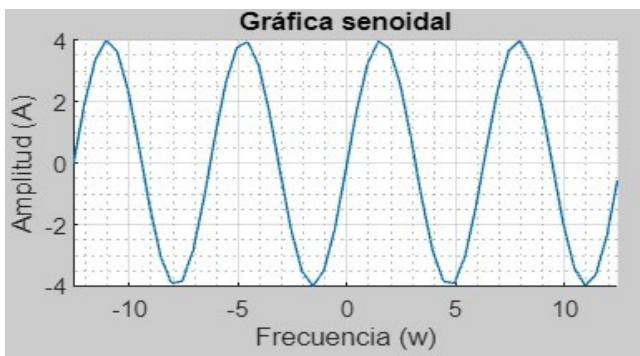


Figura 1. Señal senoidal.

De igual manera, las otras señales tienen valores los cuales el usuario debe ingresar. La señal pulso requiere que el usuario ingrese la amplitud y el ancho de la señal.

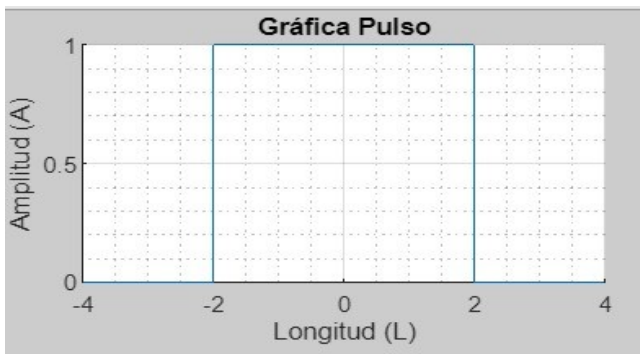


Figura 2. Señal pulso.

La señal cuadrática requiere graficar $y(t)=at^2+bt+c$, ingresando los valores de las constantes.

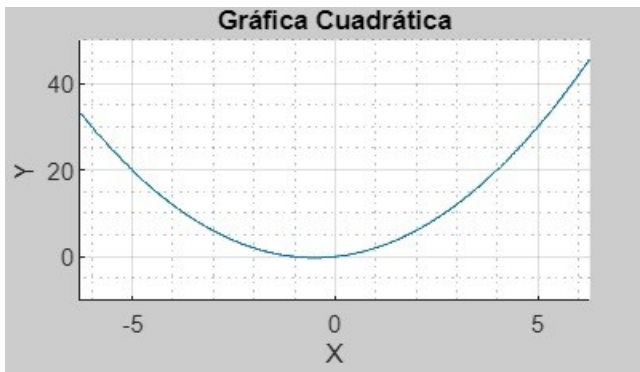


Figura 3. Señal cuadrática.

La señal exponencial debe graficar $y(t)=Ae^{-bt}$, y las constantes ingresadas por el usuario.



Figura 4. Señal exponencial.

La señal lineal debe cumplir $y(t)=mt + b$, con sus constantes digitadas por el usuario.

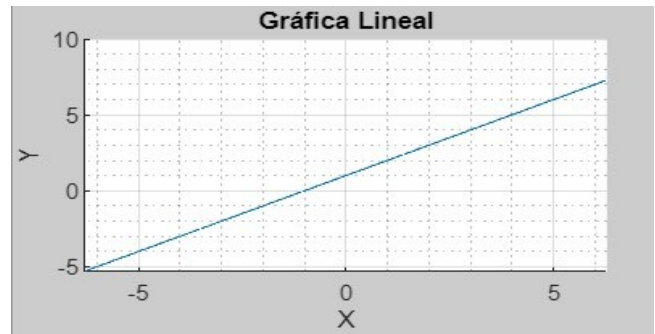


Figura 5. Señal lineal.

Para la señal triangular y la señal cuadrada, hay que hacer uso de los comandos “sawtooth” y “square” respectivamente.

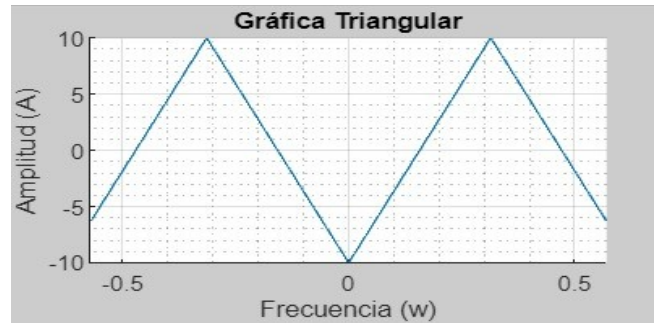


Figura 6. Señal triangular.

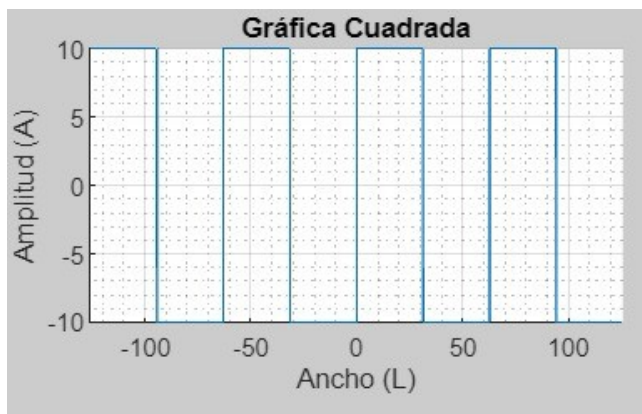


Figura 7. Señal cuadrada.

Y por ultimo la secuencia de impulsos debe permitir el ingreso de los valores de impulso para la construcción de una señal de tipo discreta.

Para la segunda parte del laboratorio se requiere realizar diferentes tipos de operaciones básicas con las señales. Se debe poder escoger entre tres tipos de operaciones: Escalamiento en la amplitud, escalamiento en el tiempo y desplazamiento en el tiempo.

Para escribir líneas de código en el App Designer de MATLAB, primero se debe ingresar los componentes requeridos para los puntos a completar. Se requiere una tabla con ejes para la visualización de graficas, distintas cajas de edición de numero para poder ingresar los valores para cada grafica, una lista desplegable en la que podemos elegir el tipo de señal que deseamos, botones de opciones para elegir el tipo de operación de la señal y un botón final para generar las gráficas.

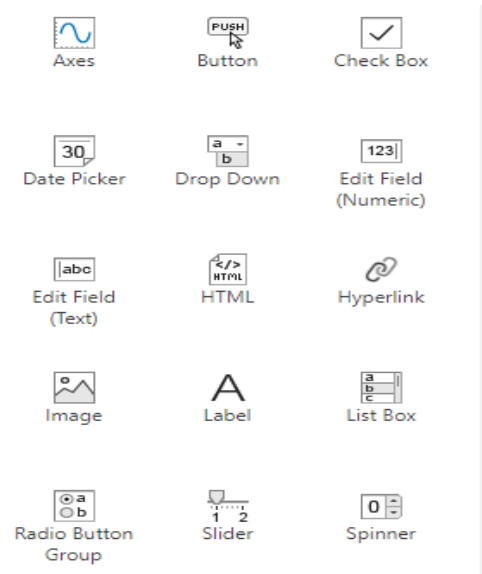


Figura 8. Librería de componentes.

Ya que se esta haciendo uso del App Designer de MATLAB, escribir códigos y operaciones varia un poco en comparación a la escritura estándar de MATLAB. Primero, se escribe el código en referencia a la presión del botón. Esto significa que al presionar el botón va a correr todo lo que esta en esa sección del código.

Entonces, para hacer que el usuario ingrese un valor de una variable se utiliza el `app.FrecuenciaEditField.Value`, donde "Frecuencia" puede ser reemplazado por cualquier palabra siempre y cuando sea igual al nombre de la caja de edición de numero. De esta misma manera, para poder graficar una onda se utiliza `plot(app.UIAxes,t,y)`. Esto hace que utilice la tabla que ingresamos previamente y no genere una pestaña aparte con la gráfica generada. El resto es simplemente escribir las ecuaciones y generar la gráfica.

Hay varias cosas que valen la pena resaltar del código. Primero, se utilizo el comando "switch" en dos distintas ocasiones para simplificar y fácilmente organizar cada opción. Lo segundo es que la variable `t` que se tiene definida en el código varia entre cada señal. Esto es simplemente para mejor visualización o por conveniencia alguna. El resto del código es básicamente escribir la ecuación proporcionada y verificar que se cumpla correctamente.

4. ANÁLISIS DE RESULTADOS

A continuación, se estudiarán detalladamente los resultados y otras cosas de cada práctica realizada, obtenidos mediante los procedimientos descritos previamente.

Primero, en la creación del código rápidamente se pudo ver que se requerían distintas cajas de edición. Esto es debido a que varias señales no usan los mismos parámetros que las otras, entonces se tuvieron que añadir unas cuantas más. Sin embargo esto generó otro problema, esta es que ahora es posible ingresar valores en lugares donde no corresponden (Figura 8).

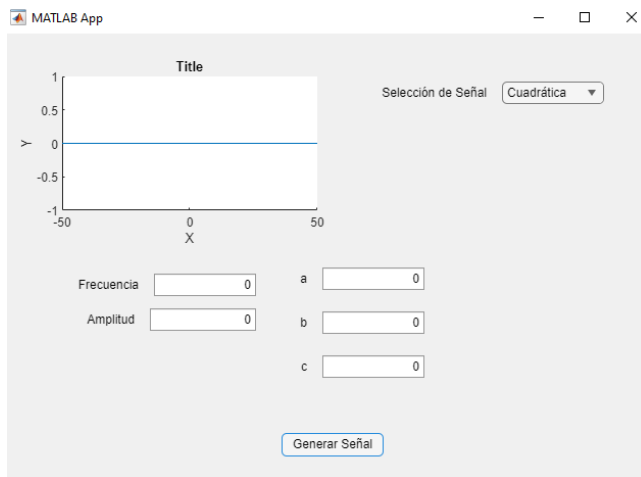


Figura 9. Visualización de error en los parámetros a ingresar.

Para arreglar esto, iniciamos el selector de señal y los parámetros a ingresar en un valor en la que ambas correspondan, y hacer que las que no se usan inicien en 0, o en otras palabras, que aparezcan invisibles inicialmente.

Luego si queremos cambiar de señal hay que hacer que las anteriores cajas de texto desaparezcan y que aparezcan las necesarias. Para esto, se creó un segundo “switch”, en la que se configuran dos propiedades de la caja de edición: “Visible” y “Label.Visible”. Si, por ejemplo, queremos hacer que el ingreso de valores para la señal cuadrática no aparezcan al seleccionar la opción de señal lineal, entonces se utiliza:

```
app.aEditField.Visible = 'off'
y
app.aEditFieldLabel.Visible = 'off'.
```

De igual manera con las diferentes variables. A pesar de esto, es importante hacer esto mismo pero con las variables que se requieren. Esto es ya que si solo tenemos variables apagadas respectivamente, lo que pasaría es que, por ejemplo, si cambiamos la señal de lineal a cuadrática, todas las cajas de edición habrán desaparecido (Figura 9).

```
switch app.SeleccioneSealDropDown.Value
case 'Senoidal'
    app.FrecuenciaEditField.Visible = 'on';
    app.FrecuenciaEditFieldLabel.Visible = 'on';
    app.AmplitudEditField.Visible = 'on';
    app.AmplitudEditFieldLabel.Visible = 'on';

    app.aEditField.Visible = 'off';
    app.aEditFieldLabel.Visible = 'off';
    app.bEditField.Visible = 'off';
    app.bEditFieldLabel.Visible = 'off';
    app.cEditField.Visible = 'off';
    app.cEditFieldLabel.Visible = 'off';
    app.mEditField.Visible = 'off';
    app.mEditFieldLabel.Visible = 'off';
```

Figura 10. Código ejemplo de como activar y desactivar una caja de edición.

Para las operaciones con las señales, esto se hace cambiando los parámetros del comando “plot”. Es decir, si para hacer un escalamiento en la amplitud, la línea de código sería `plot(app.UIAxes,t,A*y)`, para el escalamiento en el tiempo sería `plot(app.UIAxes,t/A,y)`, y para el desplazamiento en el tiempo sería `plot(app.UIAxes,t+A,y)`. Estos inicialmente aparecen de manera oculta. Para activarlo, se presiona una casilla, y esta mostrará cuatro opciones, las tres operaciones de señales y una adicional de no hacer ninguna operación, la cual propósito incluye no hacer una operación y hacer que la opción predeterminada para la generación de la onda no incluya una operación.

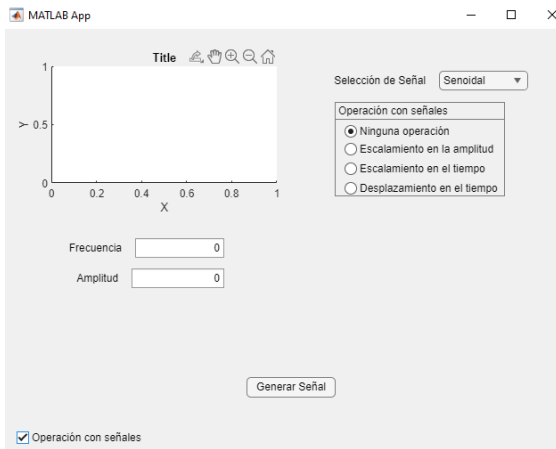


Figura 11. Casilla para activar la visualización de la operación con señales..

4. CONCLUSIONES

A lo largo de todas las prácticas realizadas, se pudo analizar y comprender el comportamiento de las distintas señales dentro del “App Designer” de MATLAB. También se pudo comprender diferentes interacciones de líneas de código y lógica de esta misma y se pudieron realizar las distintas operaciones de señales.

Se puede concluir que el laboratorio fue de mayor parte exitoso y se cumplieron con todos los objetivos propuestos. Hubieron aspectos que pudieron ser mejores al momento de la creación del código, sin embargo cumple con lo básico de lo que se necesitaba lograr.