UNIVERSIDAD DEL NORTE
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
Subject: Signals and Systems. Teacher: Carlos Cárdenas.

Lab 2: Systems Modeling

Jessir Florez, Mateo Muñoz, Dylan Abuchaibe
{jessirf, mjnova, bastod} @uninorte.edu.co
October 13, 2022

***Abstract-* In this second lab, we continue learning about signal convolution, but this time in a different programming language than the one used in the first lab.**

1. INTRODUCTION

In this laboratory practice, unlike the previous one, we use MATLAB to execute and carry out the experience.

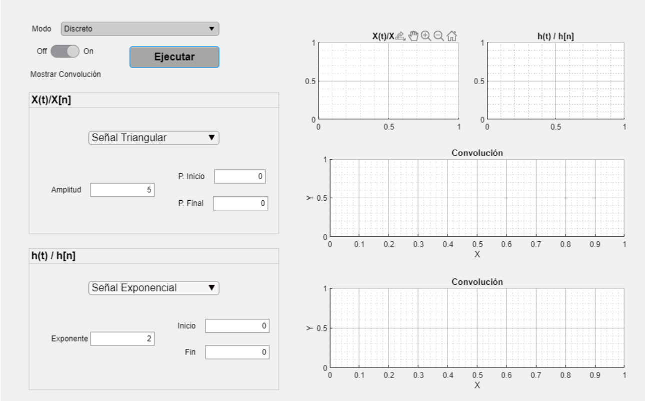
Following indications and the practice already carried out previously, work began in this new proposed laboratory to be able to convolve in continuous and discrete time with the indications proposed by the teacher in the laboratory guide.

2. PROCEDURES

As in the first laboratory, the interface was created so that the user can choose the type of signal and the variables to use. Which could be:

- Sine signal
- Exponential signal
- Rectangular signal
- Triangular signal
- Signal Ramp_1
- Signal Ramp_2
- Signal Ramp_3

In order for the user to be able to select the signal to generate, a drop-down menu was included in which they can select one of the seven available options. After having selected the signal to be plotted, the user will be asked to enter the necessary values to be able to plot the selected signal.



Modo: Discreto

X(t)/X[n]: Señal Triangular
Amplitud: 5
P. Inicio: 0
P. Final: 0

h(t) / h[n]: Señal Exponencial
Exponente: 2
Inicio: 0
Fin: 0

Figure 1. Interface in Matlab.

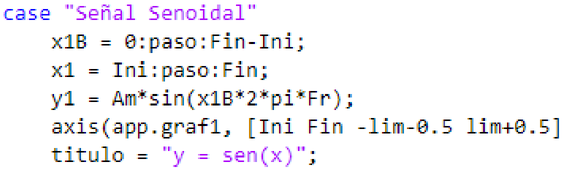
Matlab will work according to what is assigned to it and will respond to everything without any inconvenience thanks to conditionals that will act depending on a variable that will be responsible for storing each of the data entered by the user.

To be able to do the convolution part, almost the same methods that were used in the first laboratory practice were used and used. In this case, in Matlab it was necessary to use some new methods at the same time as it was related to procedures previously done and seen in class with the teacher's supervision.

Convolution will allow the user to see the effect the system will have on the input signal. You will determine the output of the system by knowing the input and the impulse response of the system. The code is fully capable of doing this process with any selected signal and will also allow you to observe in an animated way how it will be generated at each point.

For each type of signal, the different procedures required for the creation of each of them were carried out, which were:

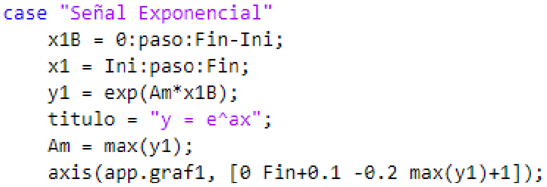
1. **Sine signal:** To create a sine signal, the user is prompted for the desired amplitude along with the start and end values. It will take all the values between the two extremes indicated as evidenced in the code.



```
case "Señal Senoidal"
    x1B = 0:paso:Fin-Ini;
    x1 = Ini:paso:Fin;
    y1 = Am*sin(x1B*2*pi*Fr);
    axis(app.graf1, [Ini Fin -lim-0.5 lim+0.5]
    titulo = "y = sen(x)";
```

Figure 2. Sine signal code.

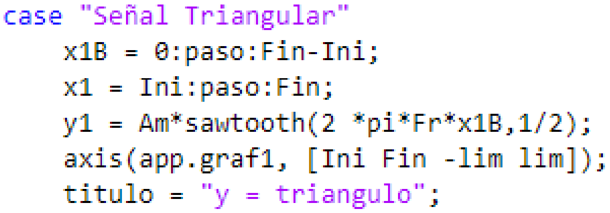
2. **Exponential signal:** In the case of the exponential signal, the user will be asked, as in the previous case, the limits required to delimit the signal and this time the value of the exponent to graph the desired signal.



```
case "Señal Exponencial"
    x1B = 0:paso:Fin-Ini;
    x1 = Ini:paso:Fin;
    y1 = exp(Am*x1B);
    titulo = "y = e^ax";
    Am = max(y1);
    axis(app.graf1, [0 Fin+0.1 -0.2 max(y1)+1]);
```

Figure 3. Exponential signal code.

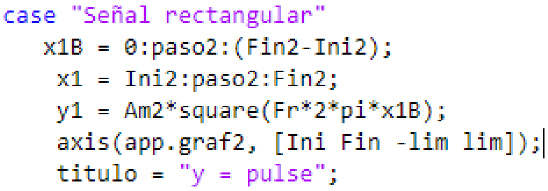
3. **Triangular signal:** For the triangular signal, the "sawtooth" command was used in order to correctly graph a wave with a triangular shape.



```
case "Señal Triangular"
    x1B = 0:paso:Fin-Ini;
    x1 = Ini:paso:Fin;
    y1 = Am*sawtooth(2 *pi*Fr*x1B,1/2);
    axis(app.graf1, [Ini Fin -lim lim]);
    titulo = "y = triangulo";
```

Figure 4. Triangular signal code.

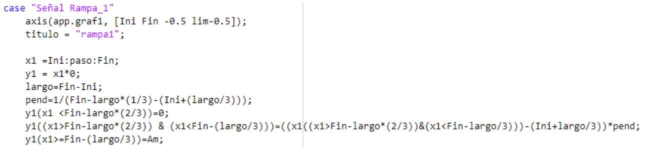
4. **Rectangular signal:** For the rectangular signal, it was necessary to use the "square" command so that the system generates a wave with a rectangular shape as requested by the user.



```
case "Señal rectangular"
  x1B = 0:paso2:(Fin2-Ini2);
   x1 = Ini2:paso2:Fin2;
   y1 = Am2*square(Fr*2*pi*x1B);
   axis(app.graf2, [Ini Fin -lim lim]);
   titulo = "y = pulse";
```

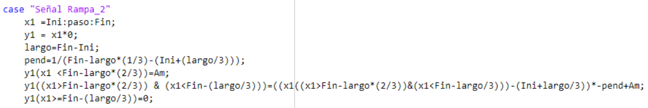
Figure 5. Rectangular signal code.

5. **Ramp signals (1-2-3):**For ramp signals, some of them are generated with vectors. While others, it creates a vector full of zeros and it will be completed with sections of ones or with the calculated slope.



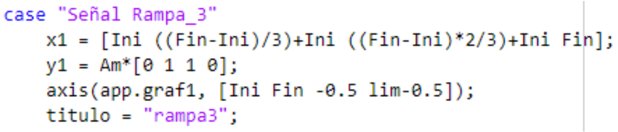
```
case "Señal Rampa_1"
    axis(app.graf1, [Ini Fin -0.5 lim-0.5]);
    titulo = "rampa1";

    x1 =Ini:paso:Fin;
    y1 = x1*0;
    largo=Fin-Ini;
    pend=1/(Fin-largo*(1/3)-(Ini+(largo/3)));
    y1(x1 <Fin-largo*(2/3))=0;
    y1((x1>Fin-largo*(2/3)) & (x1<Fin-(largo/3)))=((x1((x1>Fin-largo*(2/3))&(x1<Fin-largo/3)))-(Ini+largo/3))*pend;
    y1(x1>=Fin-(largo/3))=Am;
```

Figure 6. Ramp_1 signal code



```
case "Señal Rampa_2"
    x1 =Ini:paso:Fin;
    y1 = x1*0;
    largo=Fin-Ini;
    pend=1/(Fin-largo*(1/3)-(Ini+(largo/3)));
    y1(x1 <Fin-largo*(2/3))=Am;
    y1((x1>Fin-largo*(2/3)) & (x1<Fin-(largo/3)))=((x1((x1>Fin-largo*(2/3))&(x1<Fin-largo/3)))-(Ini+largo/3))*-pend+Am;
    y1(x1>=Fin-(largo/3))=0;
```

Figure 7. Ramp_2 signal code



```
case "Señal Rampa_3"
    x1 = [Ini ((Fin-Ini)/3)+Ini ((Fin-Ini)*2/3)+Ini Fin];
    y1 = Am*[0 1 1 0];
    axis(app.graf1, [Ini Fin -0.5 lim-0.5]);
    titulo = "rampa3";
```

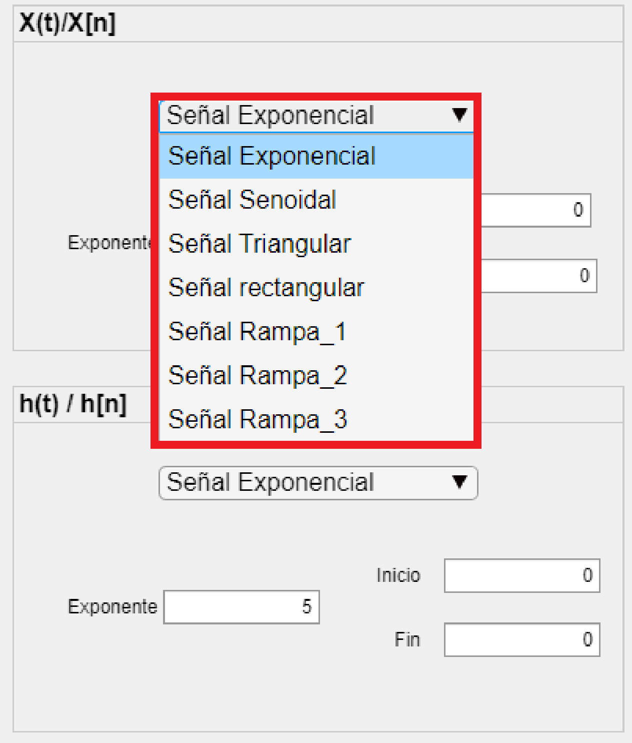
Figure 8. Ramp_3 signal code

3. ANALYSIS OF RESULTS

The laboratory was completed despite the different problems that arose from using or managing a new program such as MATLAB. It has a complete interface as evidenced in **Figure 1**.

The user will be able to implement each of the signals proposed in the drop-down menu as shown in **Figure 9**.

For the convolution part, a switch was added where the user selects if he needs the system to create the convolution or only graph the indicated signals. **Figure 10.**
As long as the button is on, the system will be able to generate the convolution process with each of the signals available for the user's choice within the drop-down menu.
The code is designed in such a way that a dynamic process is seen with each of the convolutions that the user wants to perform.



X(t)/X[n]

Señal Exponencial

Exponente

0

0

h(t) / h[n]

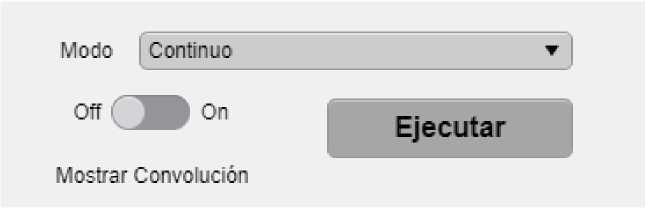
Señal Exponencial

Exponente 5

Inicio 0

Fin 0

Figure 9. Dropdown Menu



Modo Continuo

Off On

Mostrar Convolución

Figure 10. Switch for convolutions

4. CONCLUSIONS

In this second laboratory practice, we continued working with the signals and the power to convolve them. This time we worked with a new tool as it was in MATLAB, the previous laboratory was worked in PYTHON and we found some similarities in code functionalities.

As it was concluded in the previous laboratory, it came with more practice and it was a little easier to get along in this new laboratory and more bases are taken for the next ones to come.

5. REFERENCES

[1] J. Tello. Introduction to signs and Systems, Barranquilla, Colombia: Universidad del Norte, 2017.