



INSTITUTO TECNOLÓGICO DE MORELIA

DIVISIÓN DE ESTUDIOS PROFESIONALES

DEPARTAMENTO DE INGENIERIA ELECTRÓNICA

Control Digital

Practica No. 1: "Introducción a las Herramientas"

PROFESOR:

Febe Jocabed Zavala Mendoza

ALUMNO:

García Moreno Ángel Leonel (19121117)

Hernández Alvarado Andrea Joanellie (19121121)

Rojo Carrillo Pedro Eduardo (19121155)

Romero Manríquez Paavali Josue (19121156)

OBJETIVO:

Conocer y aplicar las herramientas básicas de MatLab y aprender a generar señales continuas y discretas con el entorno de programación de MatLab.

INTRODUCCION:

MatLab es un acrónimo de (Matrix Laboratory), es una herramienta de software que permite realizar cálculos, científicos y tecnológicos complejos a partir de una representación de valores basada en matrices. Integra un conjunto de librerías y comandos de alto nivel muy versátil y completo lo que hace idóneo para trabajar con simulaciones dentro de múltiples ámbitos científicos y tecnológicos [1].

Por lo tanto, en la siguiente práctica se observará y se aprenderá a generar señales continuas y discretas dentro del entorno de desarrollo de MatLab para en una posterior sesión de práctica utilizar los conocimientos necesarios para generarlas dentro de un microcontrolador.

Un microcontrolador es un dispositivo electrónico capaz de llevar a cabo procesos lógicos. Estos procesos acciones son programados en lenguaje ensamblador por el usuario, y son introducidos en este a través de un programador [2]. Por lo que se utilizará uno para generar las capturas de las señales continuas a hacia una discreta por medio de un ADC o generarlas con un DAC en su defecto.

Con las respectivas herramientas como lo son un ADC para hacer la captura de las señales continuas y convertirlo a señales digitales ya que los ADC varían en términos de resolución (la cantidad de bits utilizados para representar una señal), velocidad de muestreo (cuántas muestras por segundo pueden tomar) y otros parámetros, lo que les permite adaptarse a una amplia variedad de aplicaciones [3]. O con un DAC; para convertir una señal digital en una analógica. Se pueden usar varias técnicas. Estas se pueden agrupar en métodos con circuitos integrados y en métodos con componentes discretos pueden agrupar en métodos con circuitos integrados y en métodos con componentes discretos.

Para que estos datos sean capturados y llevados a la computadora es necesario un protocolo de comunicación lo cual uno de los más sencillos de utilizar es el protocolo UART (Universal Asynchronous Receiver-Transmitter) o USART (Universal Synchronous/Asynchronous Receiver-Transmitter) lo que permite enviar datos por una trama de bits que permiten realizar la comunicación utilizando únicamente dos líneas de datos [4]. Lo cual puede ser de utilidad para prácticas posteriores.

DESARROLLO:

a) Señales periódicas

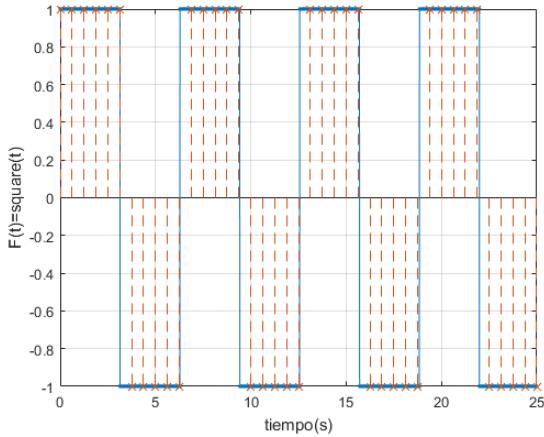


Fig. 1.1 Señal cuadrada en continua (color azul) Señal cuadrada en discreta (color rojo).

```
%SEÑAL CUADRADA
%
muestras = 10;
T = 6.25; %periodo señal Ft
Fs = muestras/T; %frec. muestreo: #muestras/ciclo
Ts = 1/Fs; %Periodo de muestreo
%SEÑAL CONTINUA
t = 0:0.0001:25;
Ft = square(t, 50);
plot(t,Ft,'.-');
xlabel('tiempo(s)') % Coloca titulo al eje x
ylabel('F(t)=square(t)') % Coloca titulo al eje y
grid on
%SEÑAL DISCRETA
kT = 0:Ts:25;
FkT = square(kT, 50);
hold on
stem(kT, FkT, '--x');
%}
```

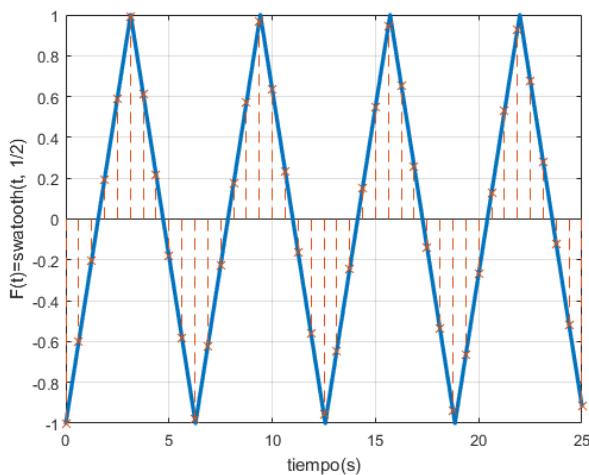


Fig. 1.2 Señal triangular en continua (color azul) Señal triangular en discreta (color rojo).

```

%SEÑAL TRIANGULAR}
%
muestras = 10;
T = 6.25; %periodo señal Ft
Fs = muestras/T; %frec. muestreo: #muestras/ciclo
Ts = 1/Fs; %Periodo de muestreo
%SEÑAL CONTINUA
t = 0:0.0001:25;
Ft = sawtooth(t, 1/2);
plot(t,Ft,'.-');
xlabel('tiempo(s)') % Coloca titulo al eje x
ylabel('F(t)=sawtooth(t, 1/2)') % Coloca titulo al eje y
grid on
%SEÑAL DISCRETA
kT = 0:Ts:25;
FkT = sawtooth(kT, 1/2);
hold on
stem(kT, FkT, '--x');
%

```

b) Señales exponenciales

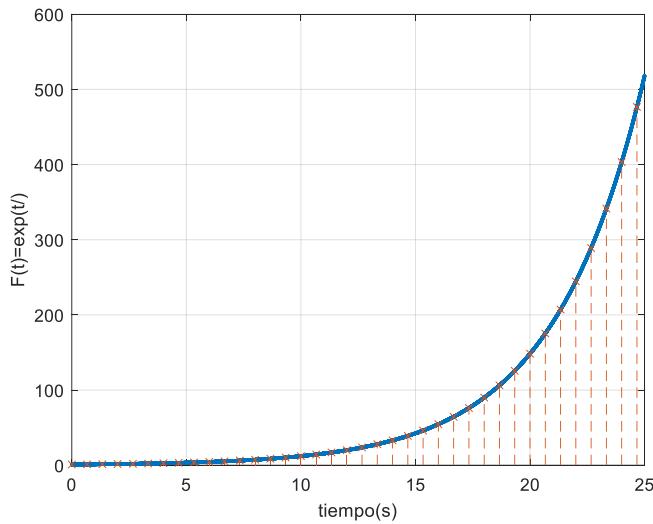


Fig. 2.1 Señal exponencial creciente en continua (color azul) Señal exponencial creciente en discreta (color rojo).

```

%SEÑAL EXPONENCIAL CRECIENTE
%
muestras = 10;
T = 6.25; %periodo señal Ft
Fs = muestras/T; %frec. muestreo: #muestras/ciclo
Ts = 1/Fs; %Periodo de muestreo
%SEÑAL CONTINUA
t = 0:0.001:25;
Ft = exp(t/4);
plot(t,Ft,'.-');
xlabel('tiempo(s)') % Coloca titulo al eje x
ylabel('F(t)=exp(t/4)') % Coloca titulo al eje y
grid on
%SEÑAL DISCRETA
kT = 0:Ts:25;
FkT = exp(kT/4);
hold on
stem(kT, FkT, '--x');
%
```

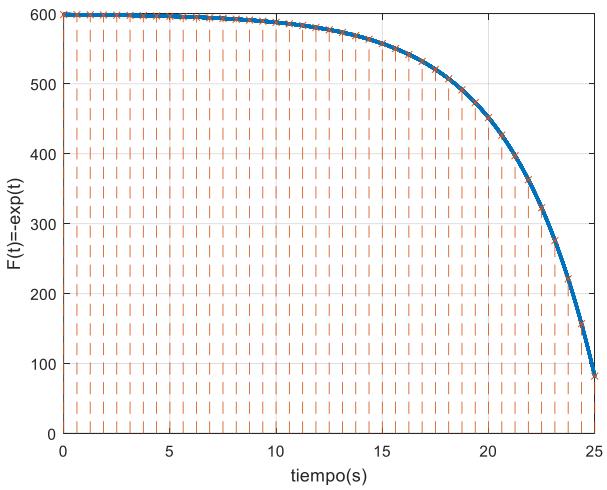


Fig. 2.2 Señal exponencial decreciente en continua (color azul) Señal exponencial decreciente en discreta (color rojo).

```
%SEÑAL EXPONENCIAL DECRECIENTE
%
muestras = 10;
T = 6.25;
Fs = muestras/T;
Ts = 1/Fs;
%SEÑAL CONTINUA
t = 0:0.001:25;
Ft = 600-exp(t/4);
plot(t,Ft,'.-');
xlabel('tiempo(s)') % Coloca titulo al eje x
ylabel('F(t)=-exp(t)') % Coloca titulo al eje y
grid on
%SEÑAL DISCRETA
kT = 0:Ts:25;
FkT = 600-exp(kT/4);
hold on
stem(kT, FkT, '--x');
%}
```

c) Señales senoidales

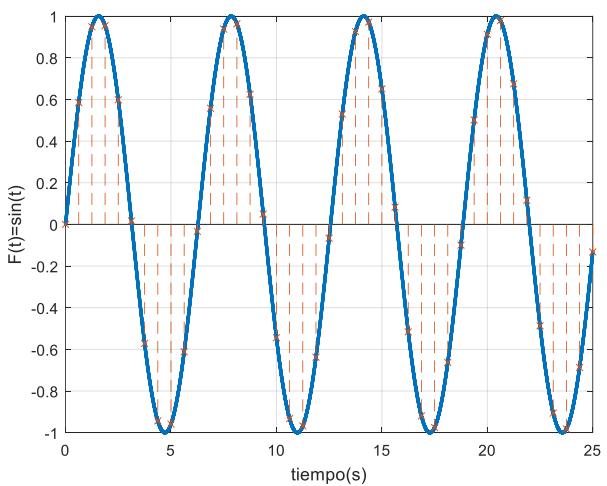


Fig. 3.1 Señal senoidal en continua (color azul) Señal senoidal en discreta (color rojo).

```
%SEÑAL SENO
%
muestras = 10;
T = 6.25;
Fs = muestras/T;
Ts = 1/Fs;
%SEÑAL CONTINUA
t = 0:0.001:25;
Ft = sin(t);
plot(t,Ft,'.-');
%plot(t,Ft);
xlabel('tiempo(s)') % Coloca titulo al eje x
ylabel('F(t)=sin(t)') % Coloca titulo al eje y
grid on
%SEÑAL DISCRETA
kT = 0:Ts:25;
FkT = sin(kT);
hold on
stem(kT, FkT, '---x');
%)
```

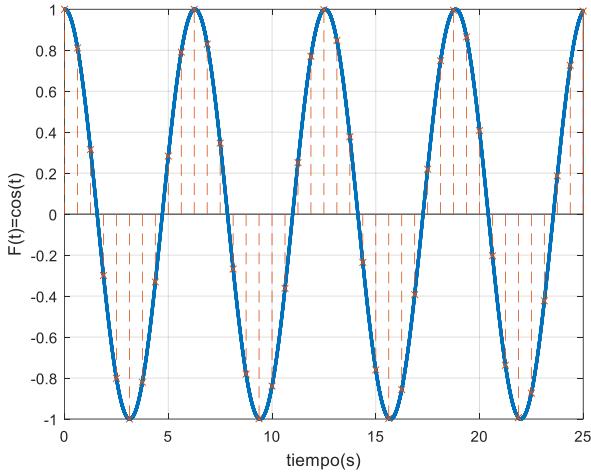


Fig. 3.2 Señal cosenoide en continua (color azul) Señal cosenoide en discreta (color rojo).

```
%SEÑAL COSENO
%
muestras = 10;
T = 6.25;
Fs = muestras/T;
Ts = 1/Fs;
%SEÑAL CONTINUA
t = 0:0.001:25
Ft = cos(t);
plot(t,Ft,'.-');
%plot(t,Ft);
xlabel('tiempo(s)') % Coloca titulo al eje x
ylabel('F(t)=cos(t)') % Coloca titulo al eje y
grid on
%SEÑAL DISCRETA
kT = 0:Ts:25;
FkT = cos(kT);
hold on
stem(kT, FkT, '---x');
%)
```

d) Señales senoidales amortiguadas exponencialmente

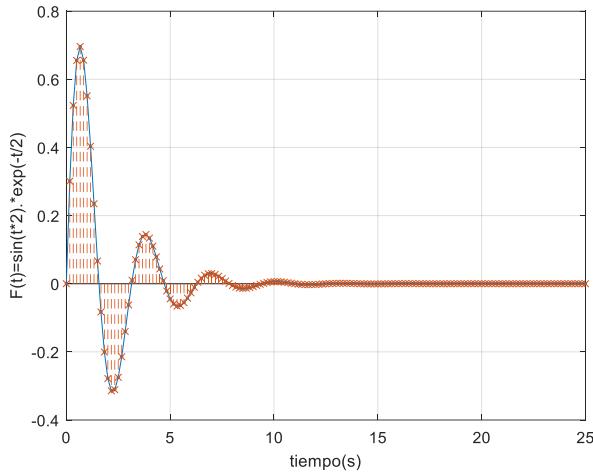


Fig. 4.1 Señal senoidales amortiguadas exponencialmente en continua (color azul) Señal senoidales amortiguadas exponencialmente en discreta (color rojo).

```
%SEÑAL SENOIDAL AMORTIGUADA EXPONENCIALMENTE
%
muestras = 10;
T = 6.25; %periodo señal Ft
Fs = muestras/T; %frec. muestreo: #muestras/ciclo
Ts = 1/Fs; %Periodo de muestreo
%SEÑAL CONTINUA
t = 0:Ts/4:25;
%t = linspace(0,20)';
Ft = sin(t.^2).*exp(-t/2);
plot(t,Ft,'.-');
%plot(t,Ft);
xlabel('tiempo(s)') % Coloca titulo al eje x
ylabel('F(t)=sin(t^2).*exp(-t/2)') % Coloca titulo al eje y
grid on
%SEÑAL DISCRETA
kT = 0:Ts/4:25;
FkT = sin(kT.^2).*exp(-t/2);
hold on
stem(kT, FkT, '--x');
%
```

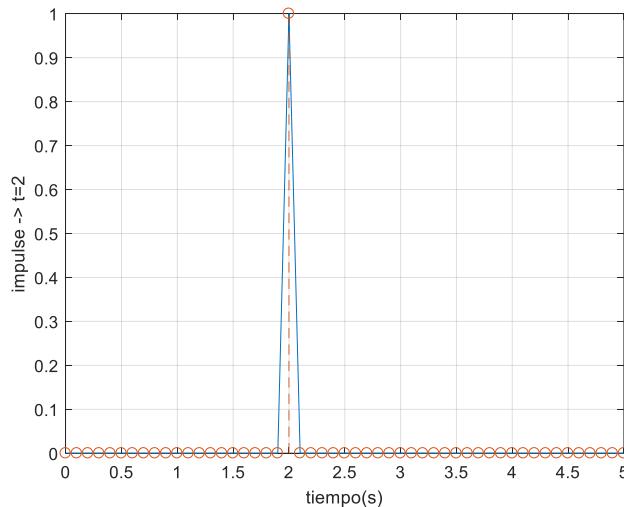


Fig. 4.2 Señal impulso en continua (color azul) Señal impulso en discreta (color rojo).

```
%FUNCIÓN IMPULSO
%
muestras = 10;
T = 6.25; %periodo señal Ft
Fs = muestras/T; %frec. muestreo: #muestras/ciclo
Ts = 1/Fs; %Periodo de muestreo
%SEÑAL CONTINUA
t = 0:0.1:5;
impulse = t==2;
Ft = 2;
plot(t,impulse);
xlabel('tiempo(s)') % Coloca titulo al eje x
ylabel('impulse -> t=2') % Coloca titulo al eje y
grid on
%SEÑAL DISCRETA
kT = (0:0.1:5);
impulsekT = kT==2;
hold on
stem(kT, impulsekT, '--');
%
```

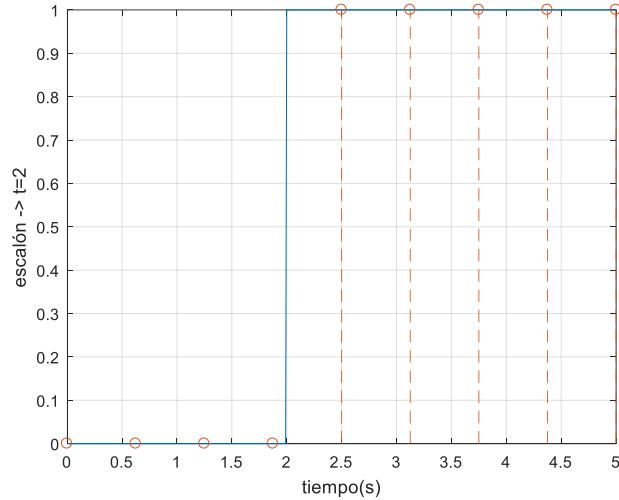


Fig. 4.3 Señal escalón en continua (color azul) Señal escalón en discreta (color rojo).

```
%FUNCIÓN ESCALÓN
%
muestras = 10;
T = 6.25; %periodo señal Ft
Fs = muestras/T; %frec. muestreo: #muestras/ciclo
Ts = 1/Fs; %Periodo de muestreo
%SEÑAL CONTINUA
t = 0:0.01:5;
unitstep = t>=2;
plot(t,unitstep);
xlabel('tiempo(s)') % Coloca titulo al eje x
ylabel('escalón -> t=2') % Coloca titulo al eje y
grid on
%SEÑAL DISCRETA
kT = 0:Ts:5;
unitstepkT = kT>=2;
hold on
stem(kT, unitstepkT, '--');
%
```

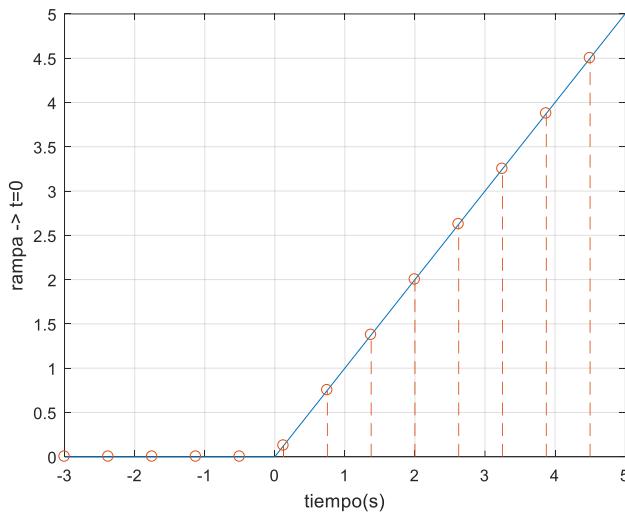


Fig. 4.4 Señal rampa (color azul) Señal rampa (color rojo).

```
%FUNCIÓN RAMPA
%
muestras = 10;
T = 6.25;
Fs = muestras/T;
Ts = 1/Fs;
%SEÑAL CONTINUA
t = -3:0.0001:5;
unitstep = t>=0;
ramp = t.*unitstep;
Ft = 2;
plot(t,ramp);
xlabel('tiempo(s)') % Coloca titulo al eje x
ylabel('rampa -> t=0') % Coloca titulo al eje y
grid on
%SEÑAL DISCRETA
kT = -3:Ts:5; %pasos de T=0.01
unitstepkT = kT>=0;
rampkT = kT.*unitstepkT;
hold on
stem(kT, rampkT, '--');
%}
```

Arquitectura del microcontrolador

En la presente practica y practicas futuras se utilizará el microcontrolador STM32F103C8, en la siguiente tabla se presenta un resumen de los puertos que dispone el microcontrolador además de sus respectivas velocidades. Toda la información fue extraída del siguiente documento [5].



Fig. 5.1 Placa de desarrollo Blue Pill con el microcontrolador STM32F103C8

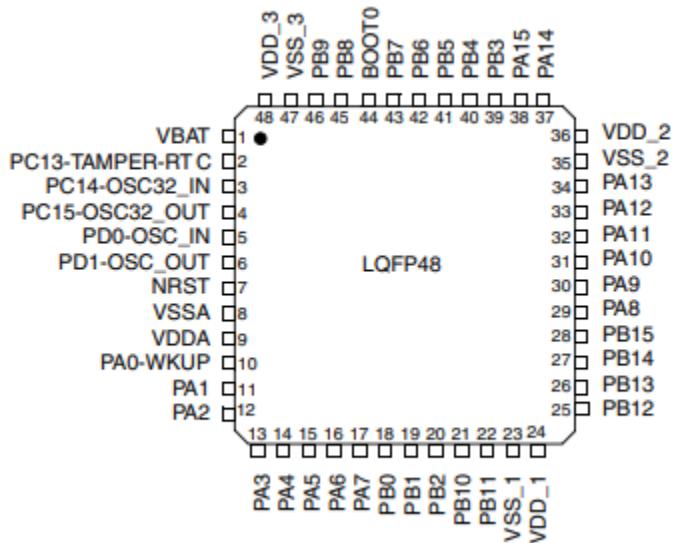


Fig. 5.2 Pin-Out del microcontrolador STM32F103C8

- 1. Puertos I/O (GPIO):** El STM32F103C8 cuenta con hasta 37 pines de E/S (GPIO) que se pueden configurar como entradas o salidas digitales según sea necesario (PA0-PA15, PB0-PB15, PC13-PC15 y PD0 junto con PD1).
- 2. Convertidores Analógico-Digital (ADC):** El dispositivo incluye un convertidor analógico a digital (ADC) de 12 bits. Puede tener hasta 10 canales de entrada analógica para la conversión de señales analógicas a digitales. (PA0-PA7, PB0 y PB1)

3. Generador de Señales: Los temporizadores incluyen temporizadores básicos (TIM2, TIM3, TIM4), temporizadores avanzados (TIM1) y temporizadores básicos avanzados (TIM8).

4. Puertos PWM (Modulación por Ancho de Pulso):

- **TIM1 y TIM8:** Estos son temporizadores avanzados que generalmente tienen más canales PWM disponibles. Cada uno de ellos puede tener múltiples canales PWM, a menudo más de 4 canales. (Para TIM1 PA8-PA12)
- **TIM2, TIM3 y TIM4:** Estos son temporizadores básicos y su número de canales PWM disponibles varía según la configuración, pero generalmente tienen al menos 4 canales PWM cada uno. (Para TIM2: PA0-PA3, para TIM3: PB4, PB5, PB0 y PB1 y por último TIM4: PB6, PB7 y los 2 restantes no permiten la salida del uC).

5. Puertos de Comunicación: Tiene hasta 9 interfaces de comunicación (I2C, USART, ISO 7816 interface, LIN, IrDA capability, modem control, SPI, CAN y USB 2.0 full-speed).

| | |
|---|--|
| I2C, SMBUS | PB7(SDA1), PB6(SCL1) y PB10(SCL2), PB11(SDA2). |
| SPI | PB3(MOSI1), PB4(MISO1), PB5(CLK1) y PB15(MOSI2), PB14(MISO2), PB13(CLK2) |
| USART, LIN, IrDA, SmartCard y Multiprocesor Comunication | PA10(RX1), PA9(TX1), PA2(TX2), PA3(RX2) y PB10(TX3), PB11(RX3) |
| USB | PA12(DP) y PA11(DM) |

6. Velocidades de Programación: La frecuencia del microcontrolador es de 0.015625MHz hasta 72 MHz

CONCLUSIONES:

García Moreno Ángel Leonel

Como conclusión para esta práctica se entiende que se tomaron los primeros pasos para la discretización utilizando la herramienta Matlab para simular diferentes tipos de señales, con el propósito de conocer algunas propiedades visuales de la discretización, la que más destaco fue el muestreo de señales siendo este un factor muy importante ya que utilizando diferentes frecuencias de muestro se pudo observar más claramente las ventajas que tiene que la frecuencia de muestreo sea mucho mayor a la frecuencia de la señal continua ya que mientras más muestras se tomaban la figura de la señal original resaltaba como la representación del área bajo la curva de una función utilizando la integral.

Hernández Alvarado Andrea Joanellie

La práctica consistió en la graficación de diferentes tipos de señales en continua y discreta. Se realizó cada una de ellas mediante el software de Matlab y todas se llevaron a cabo de manera sencilla, únicamente en la señal de decaimiento exponencial en un inicio está se graficaba de 0 a -600 en Y, sin embargo, se solucionó restando esta cifra a la función y se logró obtener la señal en continua y discreta de manera correcta.

Rojo Carrillo Pedro Eduardo

Con la realización de la práctica el alumno comprendió cabalmente como es el proceso para graficar y muestrear funciones de distintos tipos, por ejemplo, cuadradas, triangulares, etc. También se pudo observar los efectos que la frecuencia de muestreo ocasionaba al ver lo partidas que estaban las señales comparadas con el esquema continuo; el único inconveniente que se presentó fue el caso donde se tenía que ver una función exponencial en caída, para lo cual hubo que utilizar un pequeño truco matemático y se tuvo que sumar una cantidad de offset a la señal, logrando así que se obtuviera el resultado esperado.

Romero Manríquez Paavali Josue

Se concluye que la práctica se aplicaron los conocimientos sobre MatLab para graficar las señales continuas y discretas de sus respectivos tipos de función. Con la finalidad de observar las replicas discretas a una cantidad de diez muestras con una frecuencia para cuatro ciclos de la continua. También un propósito de que la generación de las señales sea en un simulador, es para observar que la generación de las discretas, aunque tengan el mismo tipo de muestreo, la información recopilada es distinta, ya sea perdiendo información como la señal impulso o con la información necesaria para replicarla como la senoidal amortiguada exponencialmente.

BIBLIOGRAFIA:

- [1] Ó. R. García, L. M. J. García, L. P. Castelló, A. G. Aparicio, y A. P. Vidal, *MATLAB: conceptos básicos y descripción gráfica*. Universidad Miguel Hernández, 2018.
- [2] P. Aguayo S., *INTRODUCCIÓN AL MICROCONTROLADOR*, Paul Aguayo S. 2004.
- [3] P. Ares Paredes, *Generador de señales usando microcontroladores*. Cartagena, 2020. [En línea]. Disponible en: <https://repositorio.upct.es/bitstream/handle/10317/9073/tfg-are-gen.pdf?sequence=1&isAllowed=y>
- [4] D. S. Dawoud y P. Dawoud, *Serial Communication Protocols and Standards*. CRC Press, 2022.
- [5] M. ST, «STM32F103x8». 29 de marzo de 2022. [En línea]. Disponible en: <https://www.st.com/resource/en/datasheet/stm32f103c8.pdf>