

# “MATLAB BÁSICO”

Informe N°2

Laboratorio de Comunicaciones Inalámbricas

1<sup>st</sup> Melanny Dávila

Ingeniería en Telecomunicaciones  
Facultad de Eléctrica y Electrónica  
Quito, Ecuador  
melanny.davila@epn.edu.ec

2<sup>nd</sup> Jonathan Álvarez

Ingeniería en Telecomunicaciones  
Facultad de Eléctrica y Electrónica  
Quito, Ecuador  
jonathan.alvarez@epn.edu.ec

**Abstract**—En el siguiente informe se va a recopilar la información de las actividades realizadas durante la práctica. Asimismo, se profundizará en temas de realización de gráficos en MatLab como usar ejes semilogarítmicos

**Index Terms**—Matrices, arreglos, dimensión, variables.

## I. INTRODUCCIÓN

La importancia del manejo de comandos simples en MATLAB será de gran importancia al tratar y presentar cálculos un manejo adecuado de las herramientas. MATLAB tiene muchas ventajas con respecto a otros lenguajes convencionales para resolver problemas técnicos debido a que es un sistema interactivo cuyo elemento básico de datos son los arreglos que no requieren ser dimensionados. Tiene rutinas poderosas que permiten gran variedad de computaciones así como comandos para graficar que brindan visualización de resultados de manera inmediata.

## II. OBJETIVOS

- Familiarizar al estudiante con la interfaz de MATLAB.
- Revisar comandos básicos utilizados en MATLAB.

## III. CUESTIONARIO

A. *Presentar los resultados obtenidos de los ejercicios propuestos en la práctica desarrollada.*

A continuación, se presenta el primer ejercicio desarrollado en clase, el mismo que consistía en la creación de dos matrices.

```
clc %Se limpia la consola
clear all %Se limpian las variables
close all %Se cierran todas las ventanas adicionales
% a) Cree una matriz A de tamaño 2x3 con números uniformemente distribuidos
% Luego cree una matriz B con unos de tamaño 3x2.
A=rand(2,3)
B=ones(3,2)
```

Fig. 1. Código implementado

Dando como resultado, las matrices mostradas en la figura 2.

A =

0.7060	0.2769	0.0971
0.0318	0.0462	0.8235

B =

1	1
1	1
1	1

Fig. 2. Resultados obtenidos

Posterior a eso, se obtuvo el producto de ambas matrices con el código mostrado en la figura 3.

```
% b) Multiplique la matriz A con la B
C=A*B
```

Fig. 3. Código implementado para el producto de matrices

Matriz 2x2 obtenida en base al producto de las matrices A y B.

C =

1.0801	1.0801
0.9015	0.9015

Fig. 4. Matriz resultante

En la figura 5 se presenta el segmento de código utilizado para la creación de una matriz identidad de dimensión 3x3 multiplicada por la matriz B y su resultado se muestra en la figura 6.

```
% c) Cree una matriz C identidad de tamaño 3x3 y multiplíquela por la matriz B
I=eye(3)*B
```

Fig. 5. Código utilizado para el producto de I\*B

```
I =

     1     1
     1     1
     1     1
```

Fig. 6. Resultados obtenido de I\*B

De igual manera, se calculó la suma de todos los elementos presentes en la fila 3 de la matriz B.

```
% d) Sume los elementos de la fila 3 de la matriz B.
S=sum(B(3,:))
```

Fig. 7. Código utilizado para la obtención de la suma de elementos

Donde el resultado es un escalar, cuyo valor es 2 debido a que los dos elementos presentes en dicha fila tienen valor de uno.

```
S =

     2
```

Fig. 8. Resultados obtenidos

De forma similar, se calculó la suma de todos los elementos que conforman la matriz A.

```
% e) Sume todos los elementos de la matriz A.
S1=sum(A(:))
```

Fig. 9. Código implementado

De esta forma se obtuvo un escalar como resultado de la suma de los seis elementos que conforman dicha matriz.

```
S1 =

    1.9816
```

Fig. 10. Sumatoria obtenida

```
% f) Obtenga un vector con los elementos de la diagonal de la matriz C.
E=diag(C)
```

Fig. 11. Código implementado

```
E =

    1.0801
    0.9015
```

Fig. 12. Resultados obtenidos

```
% g) Obtenga el máximo valor de la matriz A.
M=max(A(:))
```

```
% h) Guarde los resultados usando el comando save.
save Preguntal.mat
```

Fig. 13. Código implementado

```
M =

    0.8235
```

Fig. 14. Resultados obtenidos

El objetivo del siguiente literal es reforzar conocimientos acerca de la creación de funciones y sus respectivos dominios de trabajo con el fin de poder obtener gráficas adecuadas y entendibles para el usuario. Es así como se presenta la creación del dominio de dos funciones trigonométricas (seno y coseno).

```
clc %Se limpia la consola
clear all %Se limpian las variables
close all %Se cierran todas las ventanas adicionales
% Grafique la función seno y coseno en el intervalo [-2Pi a 2Pi] con espaciamientos de
% 0.05 en una misma Figura. Añadir lo siguiente a la gráfica:
theta=-2*pi:0.05:2*pi;
f1=sin(theta);
f2=cos(theta);
```

Fig. 15. Código implementado para la creación de funciones

Mientras que en la figura 16, se presentan algunas características implementadas para mejorar la presentación de dichas funciones es así como se designó un color diferente a cada función, se puso el título de la gráfica, el nombre de los ejes y se activó la cuadrícula.

```
% a) Marcador tipo (+) de color verde a la función seno y un marcador tipo (x) de color
% rojo a la función coseno.
plot(theta,f1,'g+',theta,f2,'rx')
% b) Título a la gráfica
title('Funciones Seno y Coseno entre -2\pi y 2\pi')
% c) Leyenda a las curvas
legend('Seno','Coseno')
% d) Etiquetas en los ejes X y Y.
xlabel('Dominio')
ylabel('Rango')
% e) Activar la grilla
grid on
```

Fig. 16. Líneas de código para los detalles de los gráficos

El resultado obtenido en base a las figuras 15 y 16 es la siguiente gráfica:

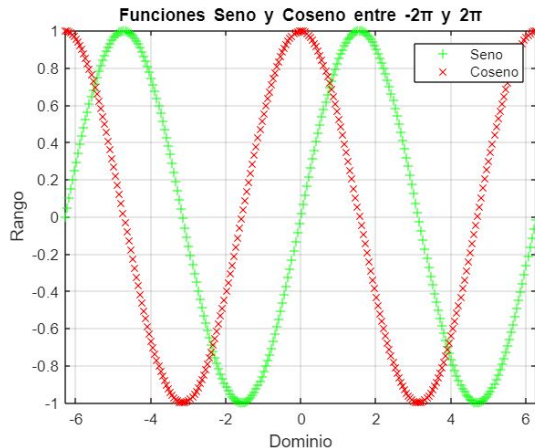


Fig. 17. Gráficas obtenidas

Finalmente, se trabajo nuevamente con operaciones aritméticas entre matrices. Para esto se creó una matriz mágica de dimensión 6x6, en la figura 18 se presenta el segmento de código que permite realizar eso.

```
clc %Se limpia la consola
clear all %Se limpian las variables
close all %Se cierran todas las ventanas adicionales
% Genere una matriz de tamaño 6x6 con el comando magic. Realice lo siguiente:
M=magic(6)
```

Fig. 18. Código de creación de matriz

A continuación se presenta el resultado dicha matriz:

```
M =
35     1     6    26    19    24
 3    32     7    21    23    25
31     9     2    22    27    20
 8    28    33    17    10    15
30     5    34    12    14    16
 4    36    29    13    18    11
```

Fig. 19. Matriz mágica

Luego de eso, se obtuvo el vector de la tercera columna mediante el código mostrado a continuación.

```
% a) Obtenga el vector de la 3 columna
C3=M(:,3)
```

Fig. 20. Segmento de código

El vector resultante se muestra en la figura 21.

```
C3 =
6
7
2
33
34
29
```

Fig. 21. Vector resultante

De manera similar, se busco obtener la fila 3 de la matriz creada.

```
% b) Obtenga el vector de la fila 3
F3=M(3,:)
```

Fig. 22. Segmento de código

El vector resultante de dicha fila se muestra en la figura 23.

```
F3 =
31     9     2    22    27    20
```

Fig. 23. Vector resultante

Mediante el uso del comando “trace”, se obtuvo de una manera rápida y sencilla la suma de los elementos de la diagonal principal.

```
% c) Obtenga la suma de los elementos de la diagonal principal
S=trace(M)
```

Fig. 24. Segmento de código

A continuación, se presenta el resultado de dicha suma.

S =

111

Fig. 25. Suma resultante

De igual manera, se calculó la suma entre el vector de la columna 3 y la fila 2 de la matriz M, para etso se transpuso al vector de la fila 2; tal y como se presenta en a figura 26.

```
% d) Obtenga el vector resultante de la suma entre la columna 3 y la fila 2
F2=M(2,:);
S1=C3+F2'
```

Fig. 26. Segmento de código para la suma de vectores

El vector resultante de la suma se muestra en la figura 27.

S1 =

9

39

9

54

57

54

Fig. 27. Vector resultante

Finalmente, se obtuvo la transpuesta de la matriz M.

```
% e) Obtenga la transpuesta de la matriz
Mt=M'
```

Fig. 28. Segmento de código para transponer a la matriz

El vector resultante se muestra en la figura 21.

Mt =

35	3	31	8	30	4
1	32	9	28	5	36
6	7	2	33	34	29
26	21	22	17	12	13
19	23	27	10	14	18
24	25	20	15	16	11

Fig. 29. Matriz transpuesta

B. Consulte los comandos *semilogy* y *semilogx*. Luego, implemente en MATLAB una gráfica de ejemplo con cada comando. Cada gráfica debe incluir un título, etiquetas de ejes y tener activa la grilla.

1) *semilogy(X,Y)*: gráfica las coordenadas X y Y usando la escala lineal en el eje X y usa escala logarítmica base 10 en el eje Y. Para graficar múltiples sistemas de coordenadas formadas por segmentos, se debe especificar X e Y como vectores de la misma longitud. Para dibujar múltiples set de coordenadas en los mismos ejes, se debe especificar al menos uno de las matrices X e Y.

- *semilogy(X,Y,LineStyle)* Genera la gráfica usando el estilo de línea, marcador y color especificado.
- *semilog(X,Y,...,Xn,Yn)* gráfica múltiples pares de coordenadas X e Y en los mismos ejes.
- *semilog(Y)* gráfica Y en contra de un set implícito de coordenadas X

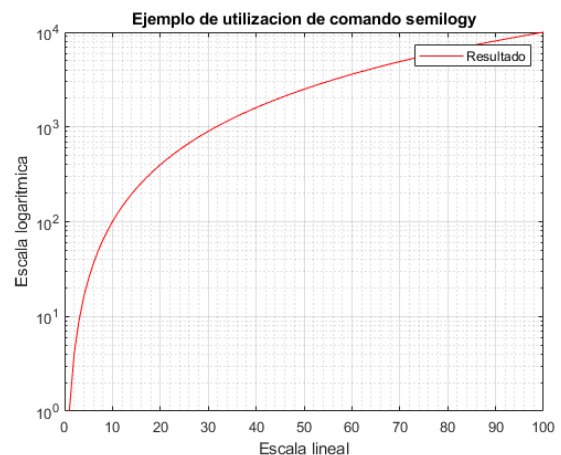


Fig. 30. Ejemplo de gráfica con el comando *semilogy*

2) *semilogx(X,Y)*: Gráfica un set de coordenadas con eje logarítmico en base 10 para el eje X y en base lineal para el eje Y. Para graficar un set de coordenadas conectadas por segmentos de línea, especifique X e Y como vectores de la misma longitud. Para graficar múltiples sets de coordenadas en los mismos ejes, especifique al menos X o Y como matriz

- *semilogx(Y,X,LineStyle)* crea un plot usando el estilo de línea, marcador y color especificados

- `semilogx(X1,Y1,...,Xn,Yn,)` Gráfica múltiples pares de coordenadas X e Y en los mismos ejes, esta sintaxis se usa como alternativa a la utilización de matrices
- `semilog(Y)` Gráfica Y contra un set implícito de coordenadas x, si y es un vector, las coordenadas irán desde 1 a la longitud de Y. Si Y es una matriz, el gráfico contendrá una línea por cada columna en Y. Las coordenadas en X variarán para el número de filas en Y.

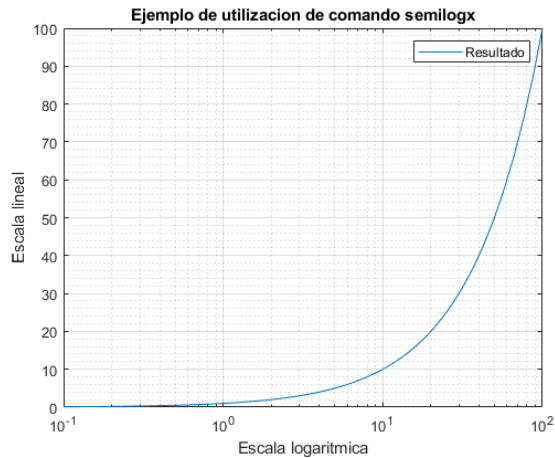


Fig. 31. Ejemplo de gráfica con el comando `semilogx`

```
informe.m  X  +
1  % Informe 2 Dávila, Alvarez
2  clc
3  clear all
4  close all
5
6  x = 1:100;
7  y = x.^2;
8  semilogy(x,y,'-r')
9  title('Ejemplo de utilizacion de comando semilogy')
10 xlabel('Escala lineal')
11 ylabel('Escala logarítmica')
12 legend('Resultado')
13 grid on; grid minor
14
15 figure
16
17 x = logspace(-1,2);
18 y = x;
19 semilogx(x,y)
20 title('Ejemplo de utilizacion de comando semilogx')
21 xlabel('Escala logarítmica')
22 ylabel('Escala lineal')
23 legend('Resultado')
24 grid on; grid minor
25
26
```

Fig. 32. Código implementado para los ejemplos de utilización de `semilogx` y `semilogy`

- El software de simulación permite manejar información de manera fácil lo que comúnmente tomaría bastante tiempo con el uso de otro programa.
- MATLAB tiene comandos que nos permiten visualizar resultados de manera inmediata con múltiples leyendas y títulos que brindan buena presentación a los gráficos

#### D. Recomendaciones:

- Se recomienda revisar las dimensiones de los arreglos previo a la implementación de alguna operación con el fin de evitar problemas en la ejecución de los diferentes scripts creados.
- Es importante colocar leyendas en los gráficos con el fin de distinguir a las diferentes gráficas mostradas y de esta manera evitar confusiones.

#### REFERENCES

- [1] Semilog plot (y-axis has log scale) - MATLAB `semilogy`. <https://www.mathworks.com/help/matlab/ref/semilogy.html> (accedido jun. 13, 2021).
- [2] Semilog plot (x-axis has log scale) - MATLAB `semilogx`. <https://www.mathworks.com/help/matlab/ref/semilogx.html> (accedido jun. 13, 2021).

#### C. Conclusiones:

- Mediante esta práctica de laboratorio se pudo reforzar conocimientos aprendidos previamente. Asimismo, se pudo aprender nuevos comandos que facilitarán la realización de diferentes operaciones aritméticas entre matrices.