

UNIVERSIDAD POLITÉCNICA DE VICTORIA
Integración de Sistemas Mecatrónicos
Ángel Arturo Ramírez Suárez IM 9-1

PRÁCTICA 1.

INSTRUCCIONES. Elabora un vector de 500 números aleatorios de una distribución normal con media 2 y desviación estándar de 5. Verifica la desviación estándar y la media utilizando (mean,std).

Código.

```
%Random numbers.
%Author: Arthur91x
%Universidad Politécnica de Victoria
%1.-Random variables. Make a vector of 500 random numbers from a Normal
distribution with
%mean 2 and standard deviation 5 (randn). After you generate the vector,
verify that the sample
%mean and standard deviation of the vector are close to 2 and 5
respectively (mean, std).

%Clean the Workspace, local variables and close open graphs.
clear
close all
clc

%Generate the vector with 500 random numbers, mean of 2 and std deviation
%of 5.
r = 2 + 5.*randn(500,1);

%Verify the mean and std.
dev = std(r)

med = mean(r)
```

Resultado.

```
dev =  
    5.1740  
  
med =  
    1.9886
```

Figura 1. Comprobación de media y desviación estándar.

PRÁCTICA 2.

INSTRUCCIONES. Elabora un script llamado `coinTest.m` que simule el modelo secuencial de una moneda que es lanzada 5000 veces. Registra cada vez que se obtiene 'heads' y grafica el estimado de probabilidad de obtener dicho valor con esta moneda. Grafica esta estimación junto con una línea horizontal en el valor esperado de 0.5.

Código.

```
%2.-Flipping a coin. Write a script called coinTest.m to simulate  
sequentially flipping a coin 5000  
%times. Keep track of every time you get 'heads' and plot the running  
estimate of the probability  
%of getting 'heads' with this coin. Plot this running estimate along with  
a horizontal line at the  
%expected value of 0.5, as below. This is most easily done without a loop  
(useful functions: rand,  
%round, cumsum).
```

```
%Clean all environment variables and close open graphs. Clean Workspace.  
clear  
close all  
clc
```

```
%Declare the number of coinflips.  
n=5000;
```

```
%  
r = randi([0,1],1,n);
```

```
%Get the cumulative values.  
res = cumsum(r);
```

```
%Plot the data.  
figure
```

```
%Divide the cumulative values individually by the number of points from 1
```

```

%to 5000.
plot(res ./ (1:n), 'r', 'lineWidth', 2);

%Retain the data so that it doesn't get erased by the second plot.
hold on

%Plot a line at 0.5 to indicate the desired value.
plot(0:4999, 0.5, 'b', 'lineWidth', 2)

%Add legend to the lines.
legend('Probability calculated.', 'Fair coin.')

%Add markings to the table.
title('Sample probability of Heads in a coin simulation.')
xlabel('Trials')
ylabel('Probability of heads')

%Add grid
grid on
axis([0 5000 0 1]) ;

```

Resultado.

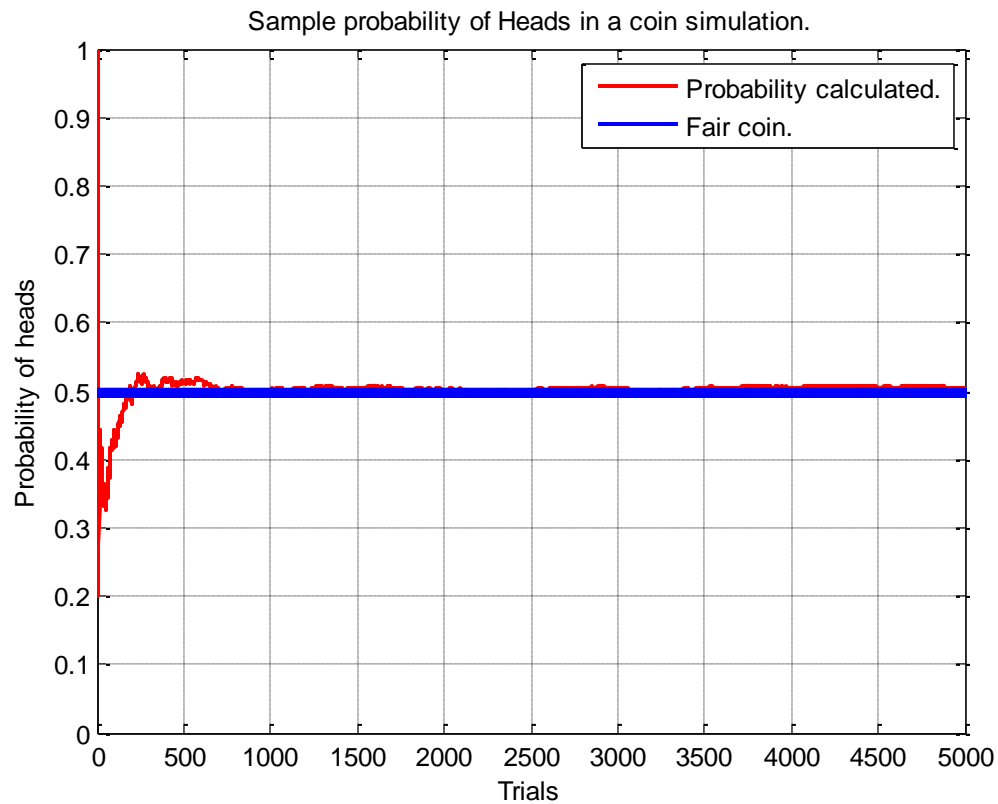


Figura 2. Modelo de probabilidad de obtener 'heads' en 5000 lanzamientos de moneda.

PRÁCTICA 3.

INSTRUCCIONES. Elabora una distribución de Poisson de 1000 números distribuidos con parámetro lambda 5.

Código.

```
%3.-Histogram. Generate 1000 Poisson distributed random numbers with
parameter lambda = 5
%(poissrnd). Get the histogram of the data and normalize the counts so
that the histogram sums
%to 1 (hist - the version that returns 2 outputs N and X, sum). Plot the
normalized histogram
%(which is now a probability mass function) as a bar graph (bar). Hold on
and also plot the actual
%Poisson probability mass function with lambda = 5 as a line (poisspdf). You
can try doing this with
%more than 1000 samples from the Poisson distribution to get better
agreement between the
%two.

%Clear Workspace, local variables and close open graphs.
close all
clear
clc

%Declare lambda value.
lambda = 5;

%n
n = 1:1000;

%Generate the Poisson distribution.
r = poissrnd(lambda,1000,1);

%Generate the histogram.
[x,c] = hist(r,13); %D is your data and 13 is number of bins.
h = x/sum(x); %Normalize to unit length. Sum of h now will be 1.

%Generate bar graphs with 13 values according to the number of analysis
%points.
bar(1:13,h);

%Keep the result along with the rest of the data from other plots.
hold on

%Generate the Poisson probability mass function.
p = poisspdf(1:13,5);
[a,b] = hist(p);

%Make a line plot of the real Poisson mass function.
plot(1:13,p,'r','lineWidth',2)

%Add grid, labels and tags.
```

```
grid on
title('Poisson distribution / mass function.')
xlabel('Samples.')
ylabel('Distribution values.')
legend('Distributed Poisson', 'Poisson mass function')
```

Resultado.

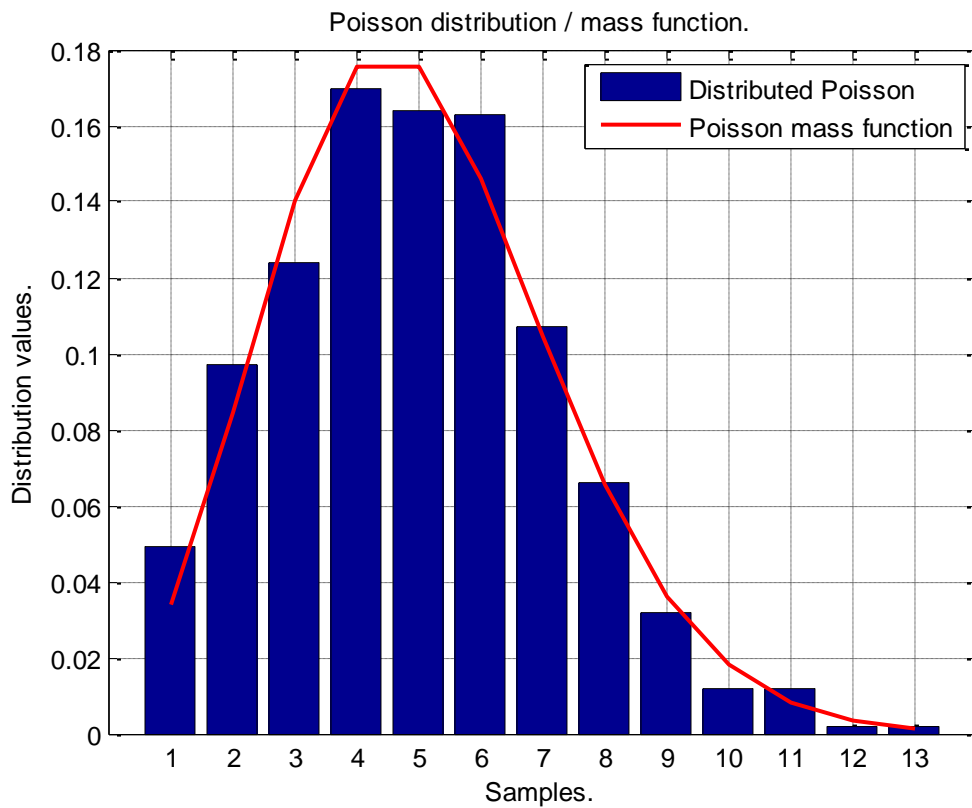


Figura 3. Gráfica de la distribución de Poisson.

PRÁCTICA 4.

INSTRUCCIONES. Práctica con celdas.

- Elabora una celda de 3x3 donde la primera columna contenga los nombres 'Joe', 'Sarah' y 'Pat'; la segunda columna 'Smith', 'Brown' y 'Jackson' y la tercera columna contenga sus salarios \$30,000.00, \$150,000.00 y \$120,000.00. Muestra los resultados utilizando disp.
- Sarah se casa y cambia su apellido a 'Meyers'. Realiza este cambio en la celda que elaboraste.
- Pat recibe una promoción y tiene un aumento de \$50,000.00. Cambia su salario añadiendo esta cantidad al valor correspondiente en la celda.

Código.

```
%Practice with cells. Usually, cells are most useful for storing strings,
because the length of each
%string can be unique.
%a. Make a 3x3 cell where the first column contains the names: 'Joe',
'Sarah', and 'Pat', the
%second column contains their last names: 'Smith', 'Brown', 'Jackson',
and the third
%column contains their salaries: $30,000, $150,000, and $120,000. Display
the cell using
%disp.
%b. Sarah gets married and changes her last name to 'Meyers'. Make this
change in the cell
%you made in a. Display the cell using disp.
%c. Pat gets promoted and gets a raise of $50,000. Change his salary by
adding this amount
%to his current salary. Display the cell using disp.

%Clean the Workspace, local variables and close open graphs.
clear
close all
clc

%Title.
disp('cellProblem')

%Declare the cell values.
cell = {'Joe', 'Smith', 30000; 'Sarah', 'Brown', 150000; 'Pat',
'Jackson', 120000};

%Display it using disp even though it can be done automagically, as asked
in
%the exercise.
disp(cell)

%Change the values of element 2,2 of the cell from 'Brown' to 'Meyers'.
cell{2,2} = 'Meyers';
```

```

%Diplay it using disp even though it can be done automagically, as asked
in
%the exercise.
disp(cell)

%Add the raise to Pat's salary.
cell{3,3} = 120000+50000;

%Diplay it using disp even though it can be done automagically, as asked
in
%the exercise.
disp(cell)

```

Resultado.

```

cellProblem
    'Joe'      'Smith'      [ 30000]
    'Sarah'    'Brown'      [150000]
    'Pat'      'Jackson'    [120000]

    'Joe'      'Smith'      [ 30000]
    'Sarah'    'Meyers'     [150000]
    'Pat'      'Jackson'    [120000]

    'Joe'      'Smith'      [ 30000]
    'Sarah'    'Meyers'     [150000]
    'Pat'      'Jackson'    [170000]

>> |

```

Figura 4. Elaboración y modificación de celdas.

PRÁCTICA 5.

INSTRUCCIONES. Uso de estructuras.

- Obtén el contenido de tu directorio actual. `a` es un arreglo estructura. ¿Cuál es su tamaño? ¿Cuáles son los nombres de los campos en `a`?
- Elabora un ciclo para acceder a todos los elementos de `a` y, si no son un directorio, muestra la siguiente frase: 'File filename contains X bytes' donde filename es el nombre del archivo y X el número de bytes de éste.
- Elabora una función llamada `displayDir.m` que mostrará los tamaños de los archivos del directorio en el cual es ejecutada.

Código.

```
%5.-Using Structs. Structs are useful in many situations when dealing
with diverse data. For
%example, get the contents of your current directory by typing a=dir;
%a. ais a struct array. What is its size? What are the names of the
fields in a?
%b. write a loop to go through all the elements of a, and if the element
is not a directory,
%display the following sentence 'File filename contains X bytes', where
filename is the
%name of the file and X is the number of bytes.
%c. Write a function called displayDir.m, which will display the sizes
of the files in the
%current directory when run, as below:
```

```
%Clean Workspace, local variables and close open graphs.
close all
clear
clc
```

```
%Check what's in the actual dir.
a = dir
```

```
%Iterate to check is the data is a directory or not.
ct = 0;
x = 0;
for i = 1: length(a)
    if(a(i)..isdir == 0)
        %If a(i) isn't a directory, indicate it and add to the count.
        disp('Archivo encontrado. Contador ++');
        ct = ct + 1;

        if(x == 0)
            x = 1;
            y = i;
        end
        %Get the number of bytes and filename.
        b(i).name = a(i).name;

        %Get file size.
```



```

        b(i).size = a(i).bytes;
    end
end

%Display number of files found.
fprintf('\n\n');
fprintf('Se encontraron %d archivos.\n\n',ct);

%Display name and file size.
for i=y : length(b)
    fprintf('El archivo %s contiene %d bytes.\n',b(i).name, b(i).size);
end

```

Resultado.

- a. Is a struct array of 27x1 with the fields: name, date, byte, isdir & datenum.

The screenshot shows the MATLAB command window on the left and the Variable Browser on the right. The command window displays the definition of the structure array 'a' with fields: name, date, bytes, isdir, and datenum. The Variable Browser shows 'a' as a 27x1 struct.

```

a =
27x1 struct array with fields:
    name
    date
    bytes
    isdir
    datenum
>>

```

Name	Value	M
a	<27x1 struct>	

Figura 5. Elementos de la estructura a.

- b. c.

```

Se encontraron 25 archivos.

El archivo 01-Indexado_de_matrices.m contiene 313 bytes.
El archivo 03-Visualizacion_de_matrices.m contiene 564 bytes.
El archivo GUI1.fig contiene 4754 bytes.
El archivo GUI1.m contiene 4972 bytes.
El archivo ISM - Tarea_I.pdf contiene 189669 bytes.
El archivo ISM - Tarea_II.pdf contiene 576930 bytes.
El archivo ISM - Tarea_II.rar contiene 209902 bytes.
El archivo ISM - Tarea_III.pdf contiene 452163 bytes.
El archivo ISM - Tarea_III.zip contiene 54708 bytes.
El archivo contiene bytes.
El archivo Options.m contiene 810 bytes.
El archivo Plotsin (2).m contiene 354 bytes.
El archivo Poisson.m contiene 1473 bytes.
El archivo UPV.m contiene 35 bytes.
El archivo UPV2.m contiene 56 bytes.
El archivo cellWork.m contiene 1414 bytes.
El archivo coinTest.m contiene 1225 bytes.
El archivo displayDir.asv contiene 1406 bytes.
El archivo displayDir.m contiene 1491 bytes.
El archivo plotSin.m contiene 401 bytes.
El archivo potSin.m contiene 183 bytes.
El archivo randomNumbers.m contiene 431 bytes.
El archivo shotProblems.m contiene 1310 bytes.
El archivo twoLinePlot.m contiene 240 bytes.
El archivo untitled.fig contiene 6201 bytes.
El archivo untitled.m contiene 3503 bytes.

```

Figura 6. Cálculo de tamaño de archivos automático utilizando los campos isadir y byte de la estructura a.

PRÁCTICA 6.

INSTRUCCIONES. Uso de handles.

- a. **Elabora un script llamado handlesPractice.m**
- b. **Elabora una variable x de 0 a 2π y obtén $y=\sin(x)$.**
- c. **Elabora una nueva figura y grafica (x,y,'r').**
- d. **Establece el límite de x de 0 a pi.**
- e. **Establece la propiedad xtick en el eje para que vaya de los valores [0 pi 2pi] y la etiqueta xticklabel de {'0', '1' '2'}.**
- f. **Establece la propiedad ytick con los valores -1:.5:1.**
- g. **Activa la gradilla.**
- h. **Cambia el color de los indicadores en el eje y a verde y de los indicadores x a azul. Cambia el fondo a negro.**
- i. **Cambia la propiedad de color de la figura a gris oscuro.**
- j. **Añade título 'One sine wave from 0 to 2' con tamaño de letra 14, en negrita y color blanco.**
- k. **Añade las etiquetas x & y adecuadas con color cyan y verde respectivamente, tamaño de letra 12.**

Código.

```
%6.-Handles. We'll use handles to set various properties of a figure in
%order to make it look cool.
%a. Do all the following in a script named handlesPractice.m
%b. First, make a variable x that goes from 0 to 2? , and then make
y=sin(x).
%c. Make a new figureand do plot(x,y,'r')
%d. Set the x limit to go from 0 to 2? (xlim)
%e. Set the xtick property of the axis to be just the values [0 pi
2*pi], and set
%xticklabelto be {'0','1','2'}. Use set and gca
%f. Set the ytickproperty of the axis to be just the values -1:.5:1. Use
set and gca
%g. Turn on the grid by doing grid on.
%h. Set the ycolor property of the axis to green, the xcolor property to
cyan, and the
%colorproperty to black (use set and gca)
%i. Set the colorproperty of the figure to a dark gray (I used [.3 .3
.3]). Use set and(gcf)
%j. Add a title that says 'One sine wave from 0 to 2?' with fontsize 14,
fontweight
%bold, and color white. Hint: to get the ? to display properly, use \pi
in your string.
%Matlab uses a Tex or Latex interpreter in xlabel, ylabel, and title. You
can do all this just
%by using title, no need for handles.
%k. Add the appropriate x and y labels (make sure the ? shows up that
way in the x label)
```

```

%using a fontsize of 12 and color cyan for x and green for y. Use xlabel
and ylabel
%1. Before you copy the figure to paste it into word, look at copy
options (in the figure's Edit
%menu) and under 'figure background color' select 'use figure color'.

%Clean the Workspace, local variables and close open graphs.
clear
close all
clc

%Create the x & y variables where x goes from 0 to 2*pi and y is the sin
of
%such function.
x = 0:0.1:2*pi;
y = sin(x);

%Plot the information with red outline.
plot(x,y,'r')

%Set xlim.
xlim([0 2*pi])

%Set the handle h.
set(gca,'XTick',[0 pi 2*pi])
set(gca,'XTickLabel',{'0','1','2'})
set(gca,'YTick',[-1:0.5:1])

%Activate the grid.
grid on

%Change color for X and Y axis.
set(gca,'Xcolor','c')
set(gca,'Ycolor','g')

%Change background color.
set(gca,'Color','k')

%Set title.
title('One sine wave from 0 to
2\pi.','fontSize',14,'fontWeight','bold','color','white')

%Set xlabel and ylabel.
xlabel('x values in term of \pi','fontSize',12,'color','c')
ylabel('sin(x)','fontSize',12,'color','c')

```

Resultado.

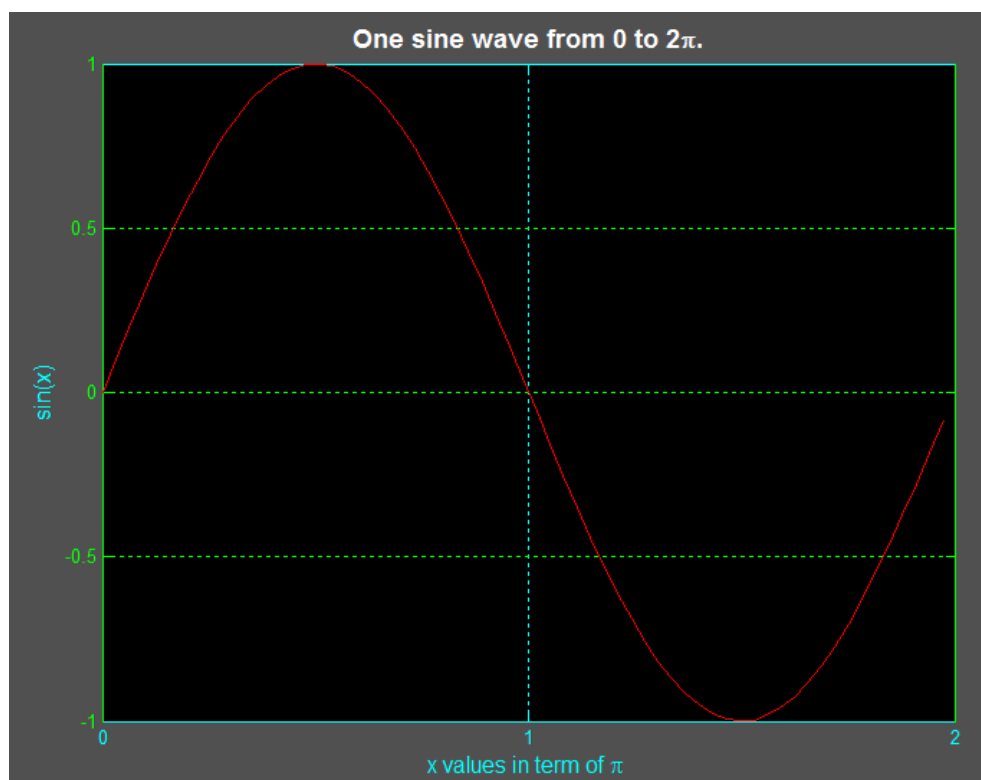


Figura 7. Gráfica generada y modificada utilizando handles.