



# EBU4375: SIGNALS AND SYSTEMS

LAB1: MATLAB FOR REPRESENTING AND MANIPULATING SIGNALS



# ACKNOWLEDGMENT

These slides are partially from Labs prepared by  
Dr Jesus Raquena Carrion.

# YOUR TASKS

- BEFORE THE LAB:

- Read the slides carefully.
- Create a ID\_FS.txt file where ID is your QMUL ID number, F is the first letter of your forename and S is the first letter of your surname.
- **Type all the code** in a red frame in the **ID\_FS.txt file** and submit to the QMplus link.

- DURING THE LAB:

- Copy/paste the code from ID\_FS.txt into Matlab command window as required- indicated by 
- Take note of the results and your answers to questions indicated by 

- **Make sure you do the work yourself as there will be questions in the class tests and exam related to Matlab.**

# WHY MATLAB?

There exist several numerical computing environments that can be used for Signals and Systems, such as Matlab or Python. In this module, we will use Matlab:

- Matlab is a numerical computing environment that allows vector and matrix manipulations, representation of data and implementation of algorithms.
- Matlab is a natural platform for signal processing and includes a convenient toolbox for modeling in a graphical interface.

# DEFINING SIGNALS IN MATLAB

- Signal are a form of time-series data where the x-axes represents the time and the y-axes represents the value of the signal at each time instance.
- Values on both axes are presented in a vector form.

```
>> v = [1 2 3 4]
```

```
v =
```

```
    1    2    3    4
```

```
>> w = [0.1 0.2 0.3 0.4]
```

```
w =
```

```
    0.1000
```

```
    0.2000
```

```
    0.3000
```

```
    0.4000
```

# DISCRETE-TIME SIGNALS

- Once defined, a DT signal can be plotted by using the command **stem**.
- The following is a Matlab script that defines and represent the DT signal  $x[n] = 3 \times n$  in the time interval  $-4 \leq n \leq 4$ :

```
n = [-4 -3 -2 -1 0 1 2 3 4]; % Variable n denotes time
x = 3*n; % Variable x is the signal value

stem(n,x) % plots x against n
xlabel('n') % adds text below the X-axis
ylabel('x[n]') % adds text beside the Y-axis
```

# CONTINUOUS-TIME SIGNALS

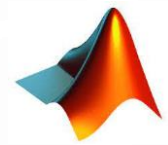
- Once defined, a CT signal can be plotted by using the command **plot**.
- BUT, it is impossible to have a vector that contains ALL values of time.
- INSTEAD, we only use the values at a finite number of time instants, which we call **samples**.
- For example, we define  $x(t) = 3 \times t$  by taking small steps in time, say  $\Delta t = 0.25$  as follows:

```
t = [-1 -0.75 -0.5 -0.25 0 0.25 0.5 0.75 1]; % Variable t ...  
           denotes time  
x = 3*t; % Variable x is the signal value  
  
plot(t,x) % plots x against t  
xlabel('t') % adds text below the X-axis  
ylabel('x(t)') % adds text beside the Y-axis
```

# EXAMPLES OF DT SIGNALS

- Before the LAB, type in your .txt file:
  - %Code from LAB1 Slide8 – QMULID= 191234567
  - Then the boxed code below.
  - Type your code after `%ADD THE CODE FOR PLOTTING x1, x2, x3r, x3i` referring to Slide 6
  - Define the value of `ne` as minimum 20 else equal to the last two digits of your QMUL ID number. If your ID is 191234567, `ne = 67`; if ID=191234501, `ne=20`.

```
ns=0;
ne=20;
n = ns:1:ne; % Time
x1 = exp(-0.2*n); % exponential signal
x2 = cos(2*pi*n/10); % sinusoidal signal
x3 = exp(j*2*pi*n/10); % complex exponential signal
x3r = real(x3); % real part of x3
x3i = imag(x3); % imaginary part of x3
x3a = abs(x3); % magnitude of x3
x3p = angle(x3); % phase of x3
figure
%ADD THE CODE FOR PLOTTING x1, x2, x3r, x3i
%remember to type figure before each
```



## Question:

How do you compare the plots of  $x2[n]$  and  $x3r[n]$ ?

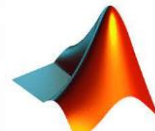
- Time shifted
- Amplified
- Not correlated
- Similar



# EXAMPLES OF CT SIGNALS

- Before the LAB, type in your .txt file:
  - %Code from LAB1 Slide9 – QMULID= 191234567
  - Then the boxed code below.
  - Type your code after `%ADD THE CODE FOR PLOTTING x1, x2, x3r, x3i` referring to Slide 7
  - Define the value of *te* as minimum 20 else equal to the last two digits of your QMUL ID number. If your ID is 191234567, *te* =67; if ID=191234501, *te*=20.

```
ts=0;
te=20;
dt=0.001;
t = ts:dt:te; % Time
x1 = exp(-0.2*t); % exponential signal
x2 = cos(2*pi*t/10); % sinusoidal signal
x3 = exp(j*2*pi*t/10); % complex exponential signal
x3r = real(x3); % real part of x3
x3i = imag(x3); % imaginary part of x3
x3a = abs(x3); % magnitude of x3
x3p = angle(x3); % angle of x3
figure
%ADD THE CODE FOR PLOTTING x1, x2, x3r, x3i
%remember to type figure before each
```



## Question:

How do you compare the plots of  $x3i(t)$  and  $x3r(t)$ ?

- Time shifted
- Amplified
- Not correlated
- Similar

# BASIC OPERATIONS WITH SIGNALS IN MATLAB

Operating with signals in Matlab means operating with the vectors that represent them. Mathematically, signals extend from  $-\infty$  to  $\infty$ . However, in Matlab we can only represent a finite number of samples.

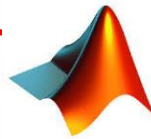
- The following are some examples of operations with DT signals in Matlab:

```
n = -10:1:10; % definition of n
x = ones(size(n)); % x is a vector with the same size as n ...
    and all ones
y = 2*x; % y is a scaled version of x
z = x + y; % z is the sum of x and y
v = y.*z ; % v is the product of x and y, DO NOT FORGET THE DOT!
w = zeros(size(n)); % w is a vector with the same size as n ...
    and all zeros
w(11:end) = 1; % The samples 11 to 21 of w are set to 1
```

# BASIC OPERATIONS WITH SIGNALS IN MATLAB

- Calculating the area, average value, energy and mean power of CT signals involves integration. In Matlab, we only represent a finite number of samples and the integral will be approximated by a sum. Given a CT signal  $x$  in Matlab and a time step between  $dt$ , area, average value, energy and mean power can be calculated as shown below.
- Before the LAB, type in your .txt file:
  - %Code from LAB1Slide11 – QMULID= 191234567
  - Then the boxed code below.
- For signals  $x1$ ,  $x2$ , and  $x3r$  defined in Slide 8, enter the values  $A_{rx}$ ,  $A_{cx}$ ,  $E_x$ ,  $P_x$  in the lab sheet on QMplus after the lab session.

```
Arx = sum(x)*dt; % Area of x
Avx = (sum(x)*dt)/(length(x)*dt); %
Average value of x
Ex = sum(x.^2)*dt; % Energy of x
Px = (sum(x.^2)*dt)/(length(x)*dt); %
Average value of x
```



Question:  
Fill in the values

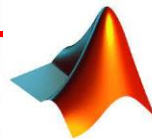


ne	Axr	Axv	Ex	Px
x1				
x2				
x3r				

# BASIC OPERATIONS WITH SIGNALS IN MATLAB

- Calculating the area, average value, energy and mean power of DT signals involves adding samples or the square of the samples.
- In Matlab, we use `sum` to add up the samples in a vector, `length` to obtain the number of samples in a vector and `x.^2` to square each sample in `x`.
- Before the LAB, type in your `.txt` file:
  - `%Code from LAB1Slide12 – QMULID= 191234567`
  - Then the boxed code below.
- For signals `x1`, `x2`, and `x3r` defined in Slide 9, enter the values `Arx`, `Acx`, `Ex`, `Px` in the lab sheet on QMplus after the lab session.

```
Arx= sum(x); % Area of x
Avx = sum(x)/length(x); % Average
value of x
Ex = sum(x.^2); % Energy of x, DO NOT
FORGET THE DOT!
Px = sum(x.^2)/length(x); %Mean Power
of x, DO NOT FORGET THE DOT!
```



Question: Fill in the values

te	Axr	Axv	Ex	Px
x1				
x2				
x3r				



# TIPS FOR THE LAB

- Prepare well and upload your .txt file before coming to the LAB.
- Open you .txt file and copy/paste into the *Command Window* the following:
  - Slide 8 code: Take some time to understand the plots.
  - Slide 11 code: Take note of all results as you need those for Lab-sheet submission
  - Slide 9 code: Take some time to understand the plots.
  - Slide 12 code: Take note of all results as you need those for Lab-sheet submission
- If you have any questions during the preparation, please post them on the Qmplus forum.
- If you have questions during the LAB please ask the supervisors.