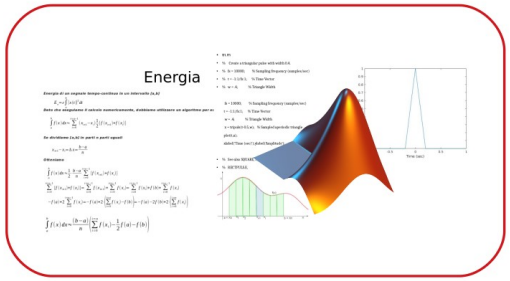
UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Programmazione Informatica

Ingegneria Civile e Ambientale
Ingegneria Navale

## Matlab 7 – Spettro di un segnale campionato. Generazione di report. Esercitazione di gruppo n.2




Antonio Fiumara
antonio.fiumara@dia.units.it

A. A. 2025-26

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Struttura del Corso

Gli argomenti trattati nel corso saranno:
Lezione Teoria 1 – Introduzione, presentazione del corso, codifica dell'informazione
Lezione Teoria 2 – Rappresentazione dell'informazione in un computer
Lezione Teoria 3 – Algebra booleana, introduzione e applicazioni nella programmazione
Lezione Teoria 4 – Architettura dei calcolatori. HW, SW, Sistemi Operativi
Lezione Teoria 5 – Algoritmi e strutture dati - Implementazione di algoritmi notevoli (ordinamento , ricerca, ecc)
Lezione Programmazione - Matlab 1 – Introduzione, installazione, comandi di base, matrici
Lezione Programmazione - Matlab 2 – Input/Output (I/O), strutture di controllo
Lezione Programmazione - Matlab 3 – Funzioni, ciclo for
Lezione Programmazione - Matlab 4 – Grafici. Sistemi Lineari, Algebra lineare, Analisi matematica
Lezione Programmazione - Matlab 5 – Strutture dati avanzate. Assegnazione esercitazione di gruppo n.1.
Lezione Programmazione - Matlab 6 – Funzioni nel dominio del tempo e della frequenza. Analisi ed elaborazione dei segnali.
**Lezione Programmazione - Matlab 7 – Spettro di un segnale campionato. Generazione di report. Esercitazione di gruppo n.2**
Lezione Programmazione - Matlab 8 – Correzione esercitazioni di gruppo. Ripasso e approfondimenti
Esonero esame Programmazione Matlab (3/12/25)

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Elaborazione dei segnali – Modulazione DSB

**DSB Dual Side Band**

$$x_M(t) = x(t)\cos(2\pi f_M t) \qquad X_M(f) = \frac{1}{2}\cdot[X(f-f_M) + X(f+f_M)]$$

- SSB Single Side Band
  Come DSB ma con una delle due "bande laterali" soppressa (con un filtro passa banda)

- AM Amplitude Modulation
  Mudulazione d'ampiezza con portante trasmessa

$$x_M(t) = A\cdot(1 + m_a x(t))\cdot\cos(2\pi f_M t);\quad m_a < 1 \rightarrow indice\ di\ mdulazione$$

- ***QAM Quadrature*** Amplitude Modulation, si possono modulare due segnali distinti utilizzando una sola portante $f_M$, purché per la modulazione si utilizzino segnali ortogonali, come, ad esempio, seno e coseno sincroni e aventi la stessa frequenza.

$$x_M(t) = x_1(t)\sin(2\pi f_M t) + x_2(t)\cos(2\pi f_M t)$$

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Elaborazione dei segnali – Demodulazione DSB

Il modo più semplice per **demodulare** un segnale DSB $x_M(T)$ è quello di eseguire nuovamente il prodotto del per **$\cos(2\pi f_M t)$**, cioè si esegue di nuovo una modulazione DSB utilizzando la stessa portante (demodulazione coerente). Per **ricostruire** il segnale originale in maniera corretta occorre applicare un **filtro passabasso** avente una frequenza di taglio $f_H$ maggiore della banda di x(t) e minore della metà della frequenza di modulazione $f_M$

$x(t)=A\cos(2\pi f_1 t)$ Segnale

$x_M(t)=x(t)\cos(2\pi f_M t)$ Segnale modulato DSB

$x_d(t)=x_M(t)\cos(2\pi f_M t)$ Segnale demodulato (coincide con un'ulteriore modulazione DSB)

$x_R(t)=lowpass(x_d(t),f_H)$ Ricostruzione del segnale x(t) in banda base

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Modulazione DSB – Esercizio

Si consideri un segnale x(t) cosinusoidale di ampiezza A=1 V e frequenza $f_1$=500 Hz

$$x(t)=A\cos(2\pi f_1 t) \qquad X(f)=\frac{1}{2}\cdot[\delta(f-f_1)+\delta(f+f_1)]$$

1) Calcolare lo spettro del segnale x(t)
2) Eseguire la modulazione DSB di x(t) con una portante cosinusoidale di ampiezza 1 e frequenza $f_M$ =10 kHz.

$$x_M(t)=x(t)\cos(2\pi f_M t) \qquad X_M(f)=\frac{1}{2}\cdot[X(f-f_M)+X(f+f_M)]$$

3) Calcolare lo spettro del segnale $x_M(t)$
4) Eseguire la demodulazione di $x_M(t)$ ed estrarre il segnale ricostruito in banda base $x_R(t)$ applicando un filtro passa basso di frequenza di taglio $f_H$ opportuna

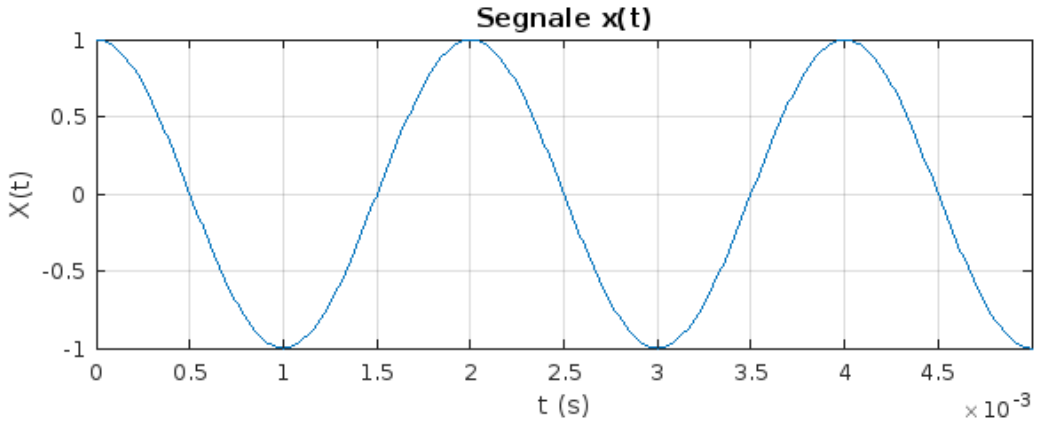
$$x_d(t)=x_M(t)\cos(2\pi f_M t) \qquad X_d(f)=\frac{1}{2}\cdot[X_M(f-f_M)+X_M(f+f_M)]$$

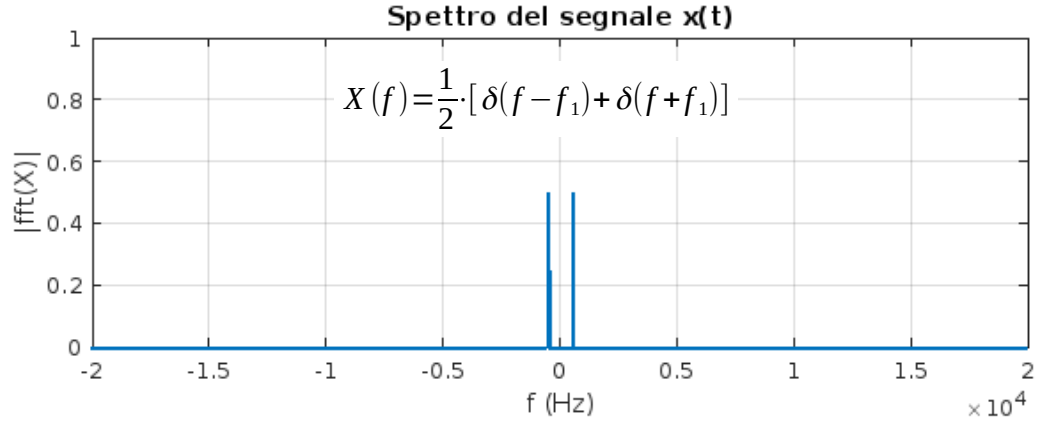
5) Eseguire i grafici dei segnali sia nel dominio del tempo sia nel dominio della frequenza

$$x_R(t)=lowpass(x_d(t),f_H) \qquad X_d(f)=?$$

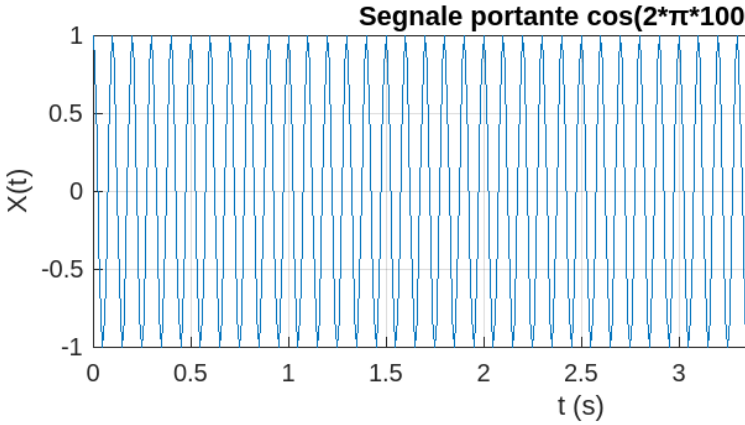
# Modulazione DSB – Esercizio

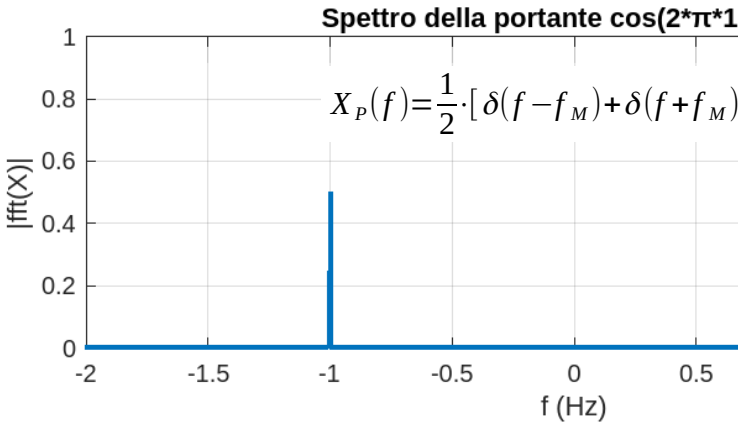
$$x(t) = A\cos(2\pi f_1 t)$$




$$x_P(t) = \cos(2\pi f_M t)$$



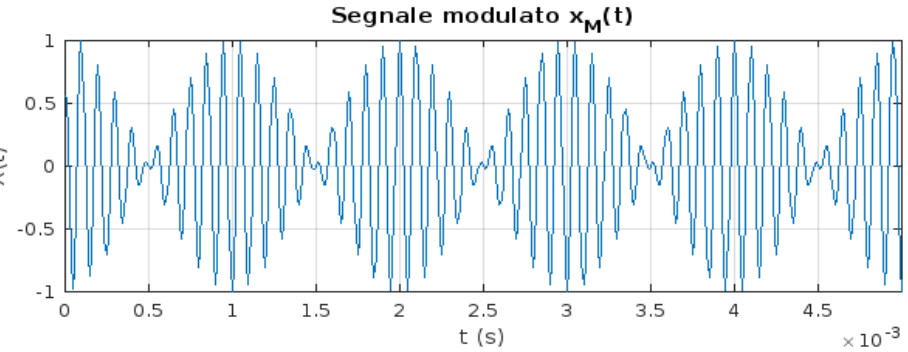



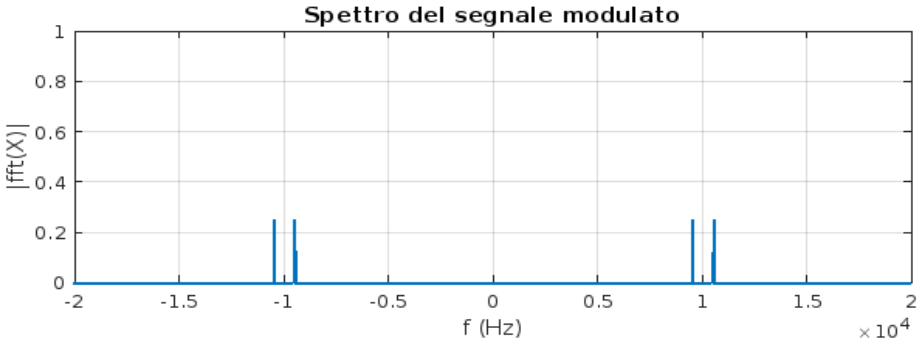
UNIVERSITÀ DEGLI STUDI DI TRIESTE

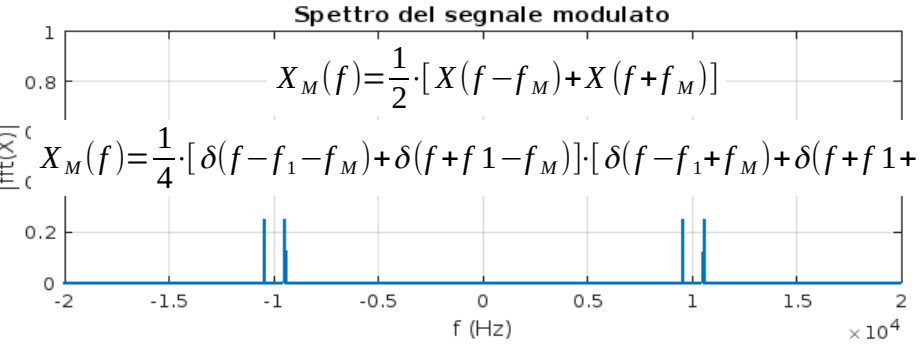
# Modulazione DSB – Esercizio

$$x_M(t) = x(t)\cos(2\pi f_M t)$$

$$X_M(f) = \frac{1}{2}\cdot[X(f-f_M)+X(f+f_M)]$$

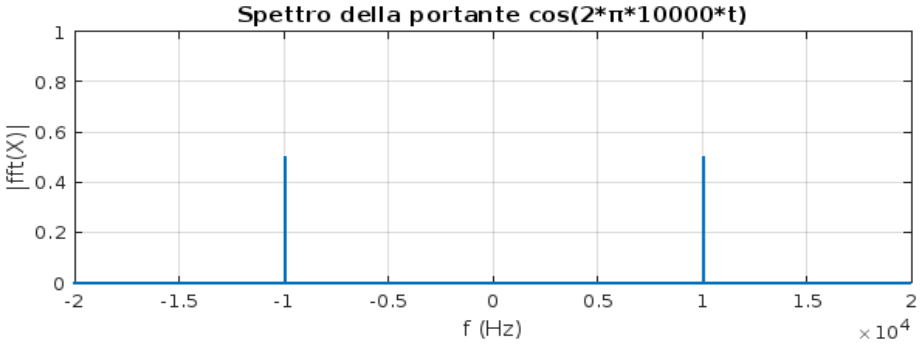



$$X_M(f) = \frac{1}{2}\cdot[X(f-f_M)+X(f+f_M)]$$

$$X_M(f) = \frac{1}{4}\cdot[\delta(f-f_1-f_M)+\delta(f+f1-f_M)]\cdot[\delta(f-f_1+f_M)+\delta(f+f1+f_M)]$$

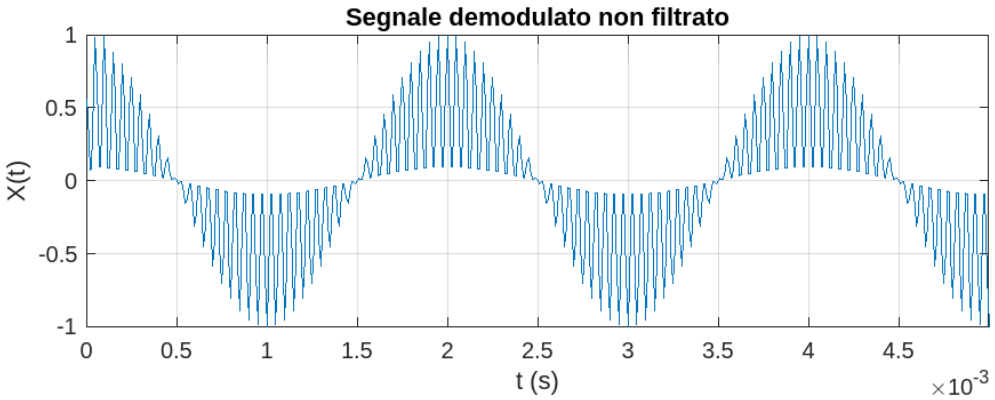


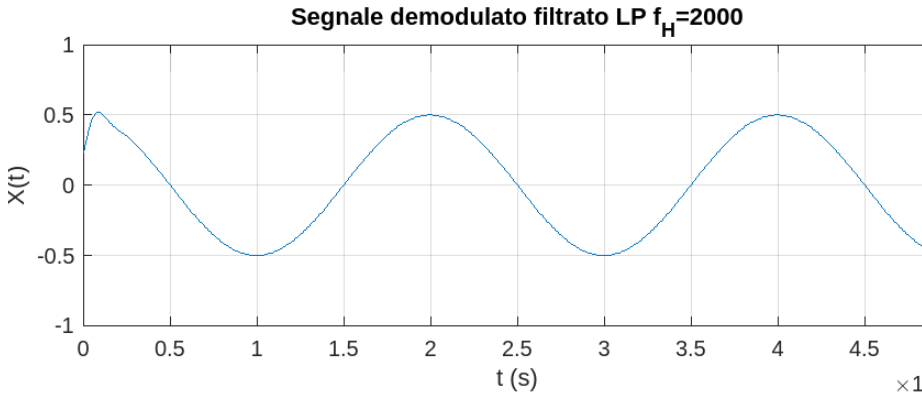

$x_R(t) = lowpass(x_d(t), f_H)$ $X_d(f) = ?$

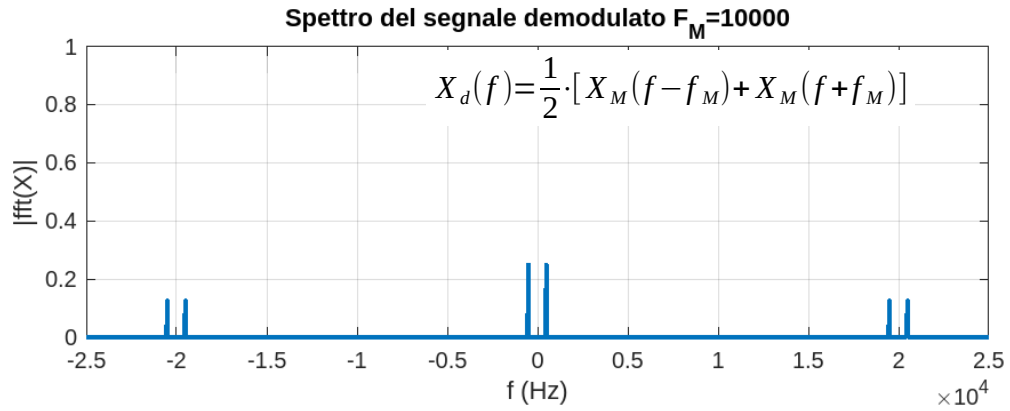
# Modulazione DSB – Esercizio

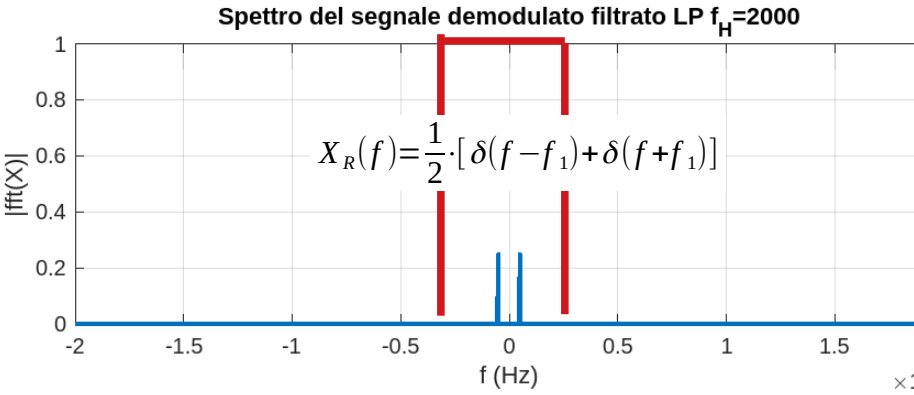
$$x_d(t) = x_M(t)\cos(2\pi f_M t)$$

$$x_R(t) = lowpass(x_d(t), f_H)$$



UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Modulazione DSB – Esercizio

```
clear all;                  %cancella tutte le variabili
close all;                  %chiude tutti i grafici
Fm=10000;                   %frequenza di modulazione (portante)
Fs=100000;                  %frequenza di campionamento
Ts=1./Fs;                   %Periodo di campionamento
L=Fs;                       %numero di campioni, abbiamo scelto L=Fs;
t=(0:L-1)*Ts;               %crea l'array tempo t  da 0 a (L-1)*Ts con passo Ts
f=Fs/L*(-L/2:L/2-1)         %crea l'array frequenza f da -Fs/2 a Fs/2 – 1 con passo
Fs/L (1 Hz nel nostro caso)
Fpass=2000;                 %frequenza di taglio del filtro passa basso, utile per
lowpass()

A1=1;                       %segnale: ampiezza
f1=500;                     %segnale: frequenza Hz
x=A1*cos(2*pi*f1*t);        %crea il segnale x(t)
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Modulazione DSB – Esercizio

```
figure(1)                                   % crea il grafico 1 (vuoto)
subplot(2,1,1)                              % predispone il plot in 2 sottografici –
primo grafico
plot(t,x)                                   % plot segnale nel dominio del tempo
title("Segnale x(t)")                       % stampa il titolo
xlabel("t (s)")                             % stampa l'etichetta per l'asse x
ylabel("X(t)")                              % stampa l'etichetta per l'asse y
axis([min(t) max(t)/200 min(x) max(x)]);    % definisce le dimensioni degli assi
grid on                                     % attiva la griglia

%sound(x,Fs)                                % riproduzione audio di x, campionato alla
frequenza Fs
%pause                                      % sospende l'esecuzione (space per
proseguire)

Y=fft(x,L)/L;                               % calcola Y(f), la trasformata di Fourier di
x(t)
Y=abs(fftshift(Y))                          % trasla Y(f) (freq negative e positive) ed
esegue |Y(f)|
figure(1)                                   % passa al grafico 1
subplot(2,1,2)                              % predispone il plot in 2 sottografici –
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Modulazione DSB – Esercizio

```
figure(2)
subplot(2,1,1)
x_lp=lowpass(x, fpass, Fs);                % applica filtro passa basso
x_m=x_lp.*cos(2*pi*Fm*t);                  % modulazione
Y=fft(x_m,L)/L;                            % trasformata FFT – spettro del segnale modu
plot(f,abs(fftshift(Y)),"LineWidth",2)
title("Spettro del segnale modulato")
axis([-2*Fm 2*Fm 0 1]);
grid on;
xlabel("f (Hz)")
ylabel("|fft(X)|")

figure(3)
subplot(2,1,2)
plot(f,abs(fftshift(Y)),"LineWidth",2)
title("Spettro del segnale modulato")
axis([-2*Fm 2*Fm 0 1]);
grid on;
xlabel("f (Hz)")
ylabel("|fft(X)|")
```

# Modulazione DSB – Esercizio

```
figure(4)
subplot(2,1,2)
x_d=x_m.*cos(2*pi*Fm*t);              % demodulazione
Y=fft(x_d,L)/L;                       % trasformata FFT del segnale demodulato
plot(f,abs(fftshift(Y)),"LineWidth",2)
title(sprintf("Spettro del segnale demodulato F_M=%d",Fm))
axis([-2.5*Fm 2.5*Fm 0 1]);
grid on;
xlabel("f (Hz)")
ylabel("|fft(X)|")

figure(5)
subplot(2,1,2)
x_r=lowpass(x_d, fpass, Fs);          % applica filtro passa basso
Y=fft(x_r,L)/L;                       % trasformata FFT del segnale ricostruito in banda
plot(f,abs(fftshift(Y)),"LineWidth",2)
title(sprintf("Spettro del segnale demodulato filtrato LP f_H=%d",fpass))
axis([-2*Fm 2*Fm 0 1]);
grid on;
xlabel("f (Hz)")
ylabel("|fft(X)|")
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Modulazione DSB – Esercizio

```
figure(3)
subplot(2,1,1)
plot(t,x_m)                                  % plot del segnale modulato nel dominio d
tempo
title("Segnale modulato x_M(t)")
axis([min(t) max(t)/200 min(x) max(x)]);
xlabel("t (s)")
ylabel("X(t)")
grid on;

figure(4)
subplot(2,1,1)
plot(t,x_d)                                  % plot del segnale demodulato prima del f
title("Segnale demodulato non filtrato")
axis([min(t) max(t)/200 min(x) max(x)]);
xlabel("t (s)")
ylabel("X(t)")
grid on;
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Modulazione DSB – Esercizio

```
figure(5)
subplot(2,1,1)
plot(t,x_r)                                          % plot del segnale x_R(t) nel dom
tempo
title(sprintf("Segnale demodulato filtrato LP f_H=%d",fpass))
axis([min(t) max(t)/200 min(x) max(x)]);
grid on
xlabel("t (s)")
ylabel("X(t)")

x_p=cos(2*pi*Fm*t);                                  % per comodità definiamo il segn
figure(6)
subplot(2,1,1)
plot(t,x_p)                                          % plot del segnale portante
title(sprintf("Segnale portante cos(2*π*%d*t)",Fm))
axis([min(t) max(t)/200 min(x) max(x)]);
grid on
xlabel("t (s)")
ylabel("X(t)")
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Modulazione DSB – Esercizio

```
figure(2)
subplot(2,1,2)

Y=fft(x_p,L)/L;                                          % spettro della portante
plot(f,abs(fftshift(Y)),"LineWidth",2)
title(sprintf("Spettro della portante cos(2*π*%d*t)",Fm))
axis([-2*Fm 2*Fm 0 1]);
grid on;
xlabel("f (Hz)")
ylabel("|fft(X)|")

figure(6)
subplot(2,1,2)
plot(f,abs(fftshift(Y)),"LineWidth",2)
title(sprintf("Spettro della portante cos(2*π*%d*t)",Fm))
axis([-2*Fm 2*Fm 0 1]);
grid on;
xlabel("f (Hz)")
ylabel("|fft(X)|")
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Note esplicative

**Y=fft(x,L)/L;** % calcola la FFT di x utilizzando L campioni. Occorre dividere per il numero di campioni

**fftshift(Y)**

Xpari = [1 2 3 4 5 6];
Fftshift(Xpari) → 4 5 6 1 2 3
Xdispari = [1 2 3 4 5 6 7];
fftshift(Xdispari) → 5 6 7 1 2 3 4

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Note esplicative

title(**sprintf**("Spettro della portante cos(2*π***%d***t)",**Fm**))

str = **sprintf**(formatSpec,A1,...,An)
A = 1/eps;  %eps  è Floating-point relative accuracy, costante di sistema e vale $2^{-52}$

str_e = sprintf('%0.5e',A) → str_e = '4.50360e+15'

str_f = sprintf('%0.5f',A) →  str_f ='4503599627370496.00000'

str_g = sprintf('%0.5g',A) → str_g ='4.5036e+15'

Fm=10000;
str_d=sprintf("Spettro della portante cos(2*π***%d***t)",**Fm**) → 'Spettro della portante cos(2*π***10000***t)'

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Note esplicative

axis([-2*Fm 2*Fm 0 1]);
**axis( [ $x_{min}$ $x_{max}$ $y_{min}$ $y_{max}$ ] );**

subplot(2,1,2)
**subplot (m,n,p)** divide la figura corrente in una griglia m x n e crea gli assi nella posizione specificata da p. MATLAB® numera le posizioni del sub-plot per riga. Il primo sub-plot è la prima colonna della prima riga, il secondo sub-plot è la seconda colonna della prima riga e così via.

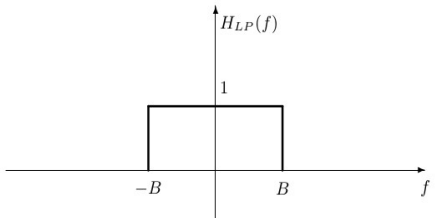
**sound(x,Fs)** % riproduzione audio di x, campionato alla frequenza Fs
**pause** % sospende l'esecuzione (space per proseguire)

UNIVERSITÀ DEGLI STUDI DI TRIESTE

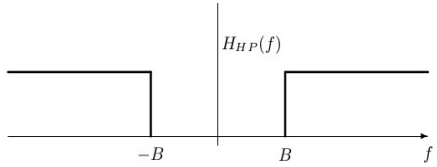
# Filtri ideali in MATLAB

I filtri ideali consentono di rimuovere perfettamente un disturbo dal segnale utile, fornendo in uscita una versione non distorta del segnale in ingresso. In base alle caratteristiche di selettività del filtro è possibile effettuare la seguente classificazione:

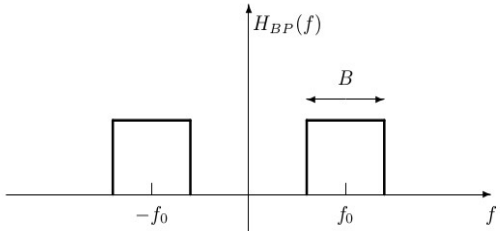
- Passa basso (Low Pass Filter)



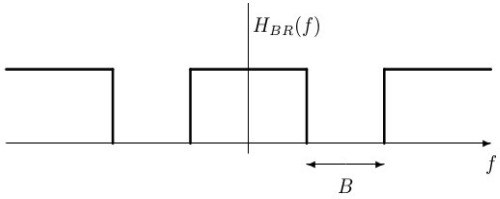

- Passa alto (High Pass Filter)




- Passa banda (Band Pass Filter)

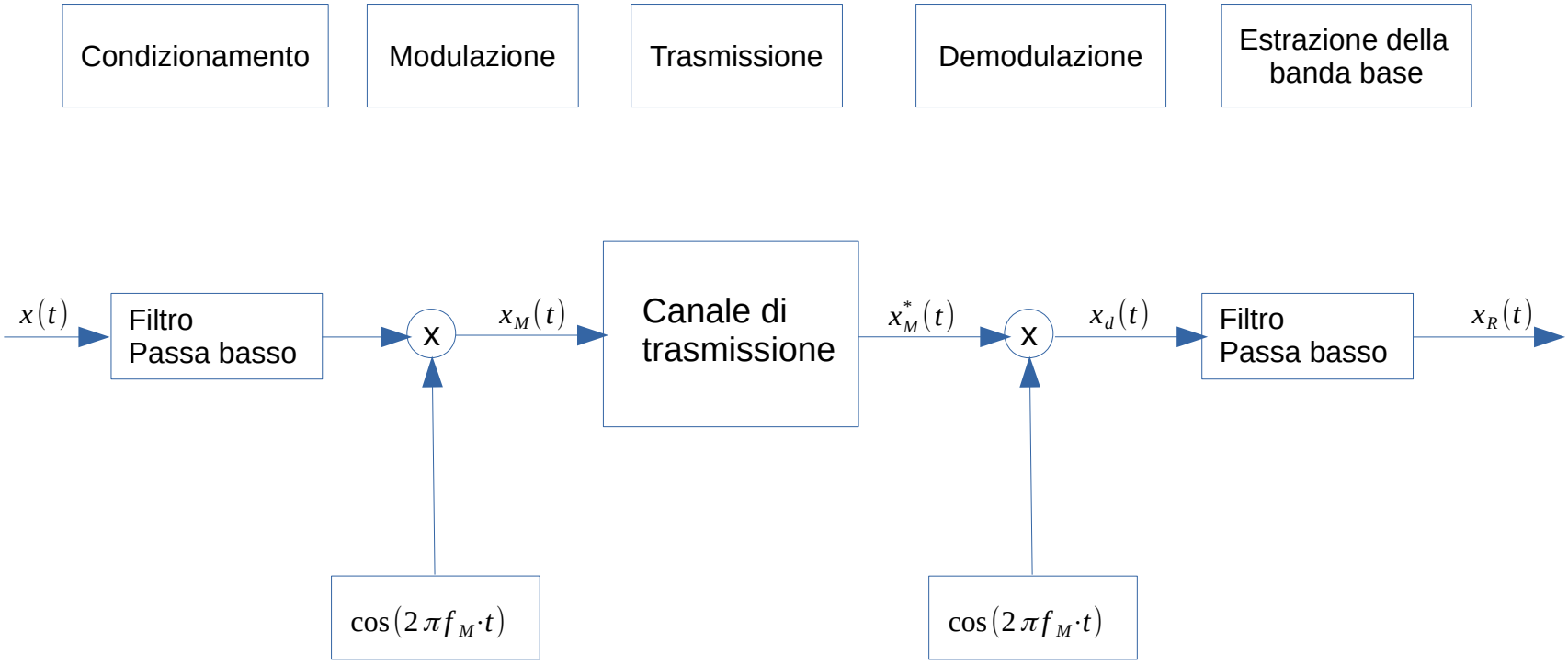



- Elimina banda  (Band Reject Filter)



UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Note esplicative



UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Classi e oggetti in MATLAB

- Le **classi** in MATLAB sono costrutti della programmazione orientata agli oggetti che fungono da modelli per creare **oggetti**, definite con la parola chiave classdef.

- Una classe è composta da **proprietà** (che memorizzano i dati di un oggetto) e **metodi** (che definiscono le operazioni eseguibili su tali dati), e può anche includere **eventi** per segnalare modifiche.

- MATLAB supporta classi definite dall'utente per creare tipi di dati personalizzati e include classi predefinite per tipi fondamentali come numeri, stringhe e logici

- L’accesso ai dati o ai metodi ricorda quello utilizzato per le **struct** (notazione a punti)

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Classi e oggetti in MATLAB - metodi

- I **metodi** sono le operazioni definite da una classe.
- I metodi possono richiamare le funzioni di MATLAB® per eseguire le operazioni sugli oggetti della classe.
- MATLAB determina quale metodo o funzione chiamare in base all'argomento dominante.
- I metodi dei **costruttori** di classe creano oggetti della classe e devono seguire regole specifiche.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Classi e oggetti in MATLAB - proprietà

- Le **proprietà** contengono i dati dell'oggetto.
- Le classi definiscono le stesse proprietà per tutti gli oggetti, ma ogni oggetto può avere valori di dati unici.
- Gli **attributi** delle proprietà controllano quali funzioni o metodi possono accedere alla proprietà.
- È possibile definire funzioni che vengono eseguite ogni volta che vengono impostati o interrogati i valori della proprietà.
- Le proprietà possono attivare **eventi** quando il codice accede ai loro valori.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Classi e oggetti in MATLAB - Costruttore

- Classes can define a special method to create objects of the class, called a **constructor**.
- Constructor methods enable you to pass arguments to the constructor, which you can assign as property values.
- The BasicClass Value property restricts its possible values using the mustBeNumeric function.
- Here is a constructor for the BasicClass class. When you call the constructor with an input argument, it is assigned to the Value property.
- If you call the constructor without an input argument, the Value property has a default value of empty ([]).

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Classi e oggetti in MATLAB - Costruttore

```
methods
    function obj = BasicClass(val)
        if nargin == 1
            obj.Value = val;
        end
    end
end
```

By adding this constructor to the class definition, you can create an object and set the property value in one step:

```
a = BasicClass(pi/3)

a =
  BasicClass with properties:
    Value: 1.0472
```

The constructor can perform other operations related to creating objects of the class.
For information on constructors, see Class Constructor Methods.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Classi e oggetti in MATLAB - esempio

- The basic purpose of a class is to define an object that encapsulates data and the operations performed on that data. For example, BasicClass defines a property and two methods that operate on the data in that property:

- Value — Property that contains the numeric data stored in an object of the class

- roundOff — Method that rounds the value of the property to two decimal places

  multiplyBy — Method that multiplies the value of the property by the specified number

- Start a class definition with a classdef ClassName...end block, and then define the class properties and methods inside that block. Here is the definition of BasicClass:

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Classi e oggetti in MATLAB - esempio

```
classdef BasicClass
   properties
      Value {mustBeNumeric}
   end
   methods
      function r = roundOff(obj)
         r = round([obj.Value],2);
      end
      function r = multiplyBy(obj,n)
         r = [obj.Value]*n;
      end
   end
end
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Classi e oggetti in MATLAB - esempio

- To use the class:
- Save the class definition in a .m file with the same name as the class.
- Create an object of the class.
- Access the properties to assign data.
- Call methods to perform operation on the data.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Classi e oggetti in MATLAB - esempio

- Creare un oggetto:

Create an object of the class using the class name:

```
a = BasicClass
```

```
a =

  BasicClass with properties:

    Value: []
```

Initially, the property value is empty.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Classi e oggetti in MATLAB - esempio

- Accedere alle proprietà

Assign a value to the `Value` property using the object variable and a dot before the property name:

```
a.Value = pi/3;
```

To return a property value, use dot notation without the assignment:

```
a.Value
```

```
ans =

    1.0472
```

For information on class properties, see Property Syntax.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Classi e oggetti in MATLAB - esempio

- Richiamare un metodo

Call the `roundOff` method on object a:

```
roundOff(a)
```

```
ans =

    1.0500
```

Pass the object as the first argument to a method that takes multiple arguments, as in this call to the `multiplyBy` method:

```
multiplyBy(a,3)
```

```
ans =

    3.1416
```

You can also call a method using dot notation:

```
a.multiplyBy(3)
```

Passing the object as an explicit argument is not necessary when using dot notation. The notation uses the object to the left of the dot.

For information on class methods, see Method Syntax.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Importare una sequenza di file

- To import or export multiple files, create a control loop to process one file at a time. When constructing the loop:
- To build sequential file names, use **sprintf**.
- To find files that match a pattern, use **dir**.
- Use function syntax to pass the name of the file to the import or export function.

https://it.mathworks.com/help/matlab/import_export/process-a-sequence-of-files.html?s_tid=srchtitle_support_results_2_file

https://it.mathworks.com/help/matlab/import_export/supported-file-formats-for-import-and-export.html

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Importare una sequenza di file

For example, to read files named **file1.txt** through **file20.txt** with **importdata**:

```
numfiles = 20;
mydata = cell(1, numfiles);

for k = 1:numfiles
  myfilename = sprintf('file%d.txt', k);
  mydata{k} = importdata(myfilename);
end
```

To read all files that match ***.jpg** with imread:

```
jpegFiles = dir('*.jpg');
numfiles = length(jpegFiles);
mydata = cell(1, numfiles);

for k = 1:numfiles
    mydata{k} = imread(jpegFiles(k).name);
end
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Importare una sequenza di file

To read all files that match ***.jpg** with **imread**:

```
jpegFiles = dir('*.jpg');
numfiles = length(jpegFiles);
mydata = cell(1, numfiles);

for k = 1:numfiles
    mydata{k} = imread(jpegFiles(k).name);
end
```

https://it.mathworks.com/help/matlab/ref/imread.html?s_tid=srchtitle_support_results_1_imread

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Generazione automatica di report in MATLAB

Esistono molti modi per generare report automatizzati in Matlab che integrino **dati** elaborati con Matlab

1) E' possibile utilizzare le funzioni di formattazione di stringa come, ad esempio, **sprintf**() e funzioni di scrittura su file, **fprintf**(), per creare un file **html** che contenga **dati** elaborati con Matlab

2) Oppure utilizzae la funzione **publish** e il relativo tool

3) MATLAB® Report Generator™ fornisce funzioni e API (Application Programming Interface ) che integrano funzionalità di reporting nelle applicazioni MATLAB.

https://it.mathworks.com/help/rptgen/report-generator-creation.html

https://it.mathworks.com/help/rptgen/?s_tid=srchbrcm

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# MATLAB Report Generator

- MATLAB® Report Generator™ fornisce funzioni e API (Application Programming Interface )che integrano funzionalità di reporting nelle applicazioni MATLAB.
- È possibile sviluppare programmi che generano report in formato PDF, Microsoft® Word, Microsoft PowerPoint® e HTML.
- MATLAB Report Generator consente di acquisire dinamicamente risultati e cifre dal codice MATLAB e di documentarli in un unico report.
- È possibile utilizzare modelli predefiniti (template) e personalizzabili in formato Word e HTML oppure progettare report basati sui modelli e sugli standard aziendali.

https://it.mathworks.com/help/rptgen/report-generator-creation.html
https://it.mathworks.com/help/rptgen/?s_tid=srchbrcm

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Generazione automatica di report in MATLAB

```
%genera html
contenuto=sprintf("<!DOCTYPE html> \n <html> <!DOCTYPE html>")
contenuto=contenuto+sprintf("<head> \n <style>")
contenuto=contenuto+sprintf("table { \n font-family: arial, sans-serif;\n border-collapse: collapse;\n width: 100%;}")
contenuto=contenuto+sprintf(" td, th {\n border: 1px solid #dddddd;\n text-align: left;\n padding: 8px;\n}")
contenuto=contenuto+sprintf("tr:nth-child(even) {\n background-color: #dddddd;\n }")
contenuto=contenuto+sprintf("</style> \n </head>\n")
contenuto=contenuto+sprintf("<body> <h2>HTML Table</h2> <table> <tr> <th>Company</th> <th>Contact</th> <th>Country</th> </tr>")
contenuto=contenuto+sprintf(" <tr> <td>Alfreds Futterkiste</td> <td>Maria Anders</td> <td>Germany</td> </tr>")
contenuto=contenuto+sprintf("<tr> <td>Centro comercial Moctezuma, via Roma %d </td> <td>Mexico</td> </tr>",10)
contenuto=contenuto+sprintf("<tr> <td>Ernst Handel</td> <td>Roland Mendel</td> <td>Austria</td> </tr>")
contenuto=contenuto+sprintf("</table> </body> </html>")
fid=fopen ("tabella.html","w")
fprintf(fid, '%s', contenuto);
fclose(fid);
```

https://www.w3schools.com/html/html_tables.asp

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Generazione automatica di report in MATLAB

```
%genera html altro modo
fid=fopen ("tabella.html","w")
fprintf(fid, "<!DOCTYPE html> \n <html> <!DOCTYPE html>\n")
fprintf(fid, "<head> \n <style>")
fprintf(fid, "table { \n font-family: arial, sans-serif;\n border-collapse: collapse;\n width: 100%;}")
fprintf(fid, " td, th {\n border: 1px solid #dddddd;\n text-align: left;\n padding: 8px;\n}")
fprintf(fid, "tr:nth-child(even) {\n background-color: #dddddd;\n }")
fprintf(fid, "</style> \n </head>\n")
fprintf(fid, "<body> <h2>HTML Table</h2> <table> <tr> <th>Company</th> <th>Contact</th> <th>Country</
th> </tr>")
fprintf(fid, " <tr> <td>Alfreds Futterkiste</td> <td>Maria Anders</td> <td>Germany</td> </tr>")
fprintf(fid, "<tr> <td>Centro comercial Moctezuma, via Roma %d </td> <td>Mexico</td> </tr>",10)
fprintf(fid, "<tr> <td>Ernst Handel</td> <td>Roland Mendel</td> <td>Austria</td> </tr>")
fprintf(fid, "</table> </body> </html>")
fclose(fid);
```

https://www.w3schools.com/html/html_tables.asp

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# MATLAB Report Generator

## Classes

### Report Containers and Base Classes

| | |
|---|---|
| `mlreportgen.report.Report` | Report container |
| `mlreportgen.dom.Document` | Document container |
| `mlreportgen.dom.Container` | Container of document objects |
| `mlreportgen.report.Reporter` | Superclass for MATLAB reporters |
| `mlreportgen.finder.Finder` | Base class for finders |
| `mlreportgen.finder.Result` | Base class for finder results |

https://it.mathworks.com/help/rptgen/report-generator-creation.html

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# MATLAB Report Generator

**Functions**

**Report Display and Conversion**

| | |
|---|---|
| `rptview` | Display report or presentation |
| `docview` | View or perform operations on Word document |
| `rptconvert` | Convert DocBook XML files generated by Report Explorer to formatted reports |

setedit(nomefile) è il comando per avviare l'app Report Explorer

https://it.mathworks.com/help/rptgen/report-generator-creation.html

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# MATLAB Report Generator - esempio

```
% 1. Import the Report and DOM namespace so that you do not have to use fully qualified class names.
import mlreportgen.report.*
import mlreportgen.dom.*
% 2. Create the report object. Use "magic" as the file name and "html" as the report type. To customize properties that apply to the whole report, see mlreportgen.report.Report.
rpt = Report("magic","pdf");
% 3. Create a title page and specify the title, subtitle and author. Then, add the title page to the report. To customize additional title page properties, see mlreportgen.report.TitlePage.
tp = TitlePage;
tp.Title = "Quadrato Magico";
tp.Subtitle = "Colonne, Righe, Diagonali: forniscono tutte la stessa somma";
tp.Author = "Nome dell'autore";
append(rpt,tp);
% 4.Add a default table of contents object to the report. To customize the table of contents, see mlreportgen.report.TableOfContents.
append(rpt,TableOfContents);
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# MATLAB Report Generator

% 5. Create a chapter object for the introduction and specify the chapter title. Add a section, add a paragraph to that section, and add that section to the chapter. Create another section and add a paragraph to it. For information on customizing chapters and sections, see mlreportgen.report.Chapter and mlreportgen.report.Section respectively.

```
ch1 = Chapter;
ch1.Title = "Introduction";
sec1 = Section;
sec1.Title = "What is a Magic Square?";
para = Paragraph(join(["A magic square is an N-by-N matrix " ...
"constructed from the integers 1 through N^2 " ...
"with equal row, column, and diagonal sums."]));
append(sec1,para)
append(ch1,sec1)
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# MATLAB Report Generator

```
sec2=Section;
sec2.Title = "Albrecht Durer and the Magic Square";
para = Paragraph(join([ ...
"The German artist Albrecht Durer (1471-1528) created "...
"many woodcuts and prints with religious and "...
"scientific symbolism. One of his most famous works, "...
"Melancholia I, explores the depressed state of mind "...
"which opposes inspiration and expression. "...
"Renaissance astrologers believed that the Jupiter "...
"magic square (shown in the upper right portion of "...
"the image) could aid in the cure of melancholy. The "...
"engraving's date (1514) can be found in the "...
"lower row of numbers in the square."]));
append(sec2,para)
append(ch1,sec2)
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# MATLAB Report Generator

```
% 6. Create an image of Durer in a figure window. Create the image in a MATLAB
figure. Add the figure to the second section of introduction chapter and then, add the
chapter to the report. For more information on figures, see mlreportgen.report.Figure.
For more information on images, see mlreportgen.report.FormalImage.
durerImage=load(which("durer.mat"),"-mat");
figure("Units","Pixels","Position",...
[200 200 size(durerImage.X,2)*.5 ...
size(durerImage.X,1)*.5 ]);
image(durerImage.X);
colormap(durerImage.map);
axis("image");
set(gca,"Xtick",[],"Ytick",[],...
"Units","normal","Position",[0 0 1 1]);
append(sec2,Figure)
append(rpt,ch1)
close gcf
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# MATLAB Report Generator

%7. Add another chapter object and specify the title. Specify the MATLAB code to create a 10-by-10 magic square. Add the results to a table and set these table properties:

Row and column separators

Table border

Alignment of table entries

% Then, add the table to the chapter and the chapter to the report. For more information on tables, see mlreportgen.dom.Table.

```
ch2 = Chapter();
ch2.Title = sprintf("10 x 10 Magic Square");
square = magic(10);
tbl = Table(square);
tbl.Style = {...
RowSep("solid","black","1px"),...
ColSep("solid","black","1px"),};
tbl.Border = "double";
tbl.TableEntriesStyle = {HAlign("center")};
append(ch2,tbl);
append(rpt,ch2);
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# MATLAB Report Generator

```
ch3 = Chapter();
ch3.Title = sprintf("25 x 25 Magic Square");
square = magic(25);
clf;
imagesc(square)
set(gca,"Ydir","normal")
axis equal
axis tight
fig = Figure(gcf);
fig.Snapshot.Height = "4in";
fig.Snapshot.Width = "6in";
fig.Snapshot.Caption = sprintf("25 x 25 Magic Square");
append(ch3,fig);
append(rpt,ch3);
delete(gcf)
close(rpt)
rptview(rpt)
```

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Esercitazione di gruppo n.2

Consegna entro il 28/11/2025
Valutazione: fino a 3 punti sul voto finale

Dato un segnale audio x(t),
1) eseguire l'analisi in frequenza di x(t)
2) applicare un filtro passa basso "avente frequenza di taglio adeguata"
3) modulare il segnale con una funzione coseno di frequenza $f_M$
4) applicare un rumore additivo di ampiezza $N_0$ ,
5) demodulare e applicare un filtro passa basso.
6) Generare, da Matlab, un report in PDF che sintetizzi il lavoro eseguito

Valori dei parametri: per ognuno dei gruppi, utilizzare il valore k calcolato nella esercitazione n.1

$f_M = k / 10$ kHz

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Energia

***Energia di un segnale tempo-continuo in un intervallo [a,b]***

$$E_x = \varepsilon \int_a^b |x(t)|^2 dt$$

***Dato che eseguiamo il calcolo numericamente, dobbiamo utilizzare un algoritmo per e:***

$$\int_a^b f(x)dx \approx \sum_{i=0}^{i=n-1} (x_{i+1}-x_i)\frac{1}{2}[f(x_{i+1})+f(x_i)]$$

***Se dividiamo [a,b] in parti n parti uguali***

$$x_{i+1}-x_i = \Delta x = \frac{b-a}{n}$$

***Otteniamo***

$$\int_a^b f(x)dx \approx \frac{1}{2}\cdot\frac{b-a}{n}\sum_{i=0}^{i=n-1}[f(x_{i+1})+f(x_i)]$$

$$\sum_{i=0}^{i=n-1}[f(x_{i+1})+f(x_i)] = \sum_{i=0}^{i=n-1} f(x_{i+1}) + \sum_{i=0}^{i=n-1} f(x_i) = \sum_{i=0}^{i=n-1} f(x_i) + f(b) + \sum_{i=0}^{i=n-1} f(x_i)$$

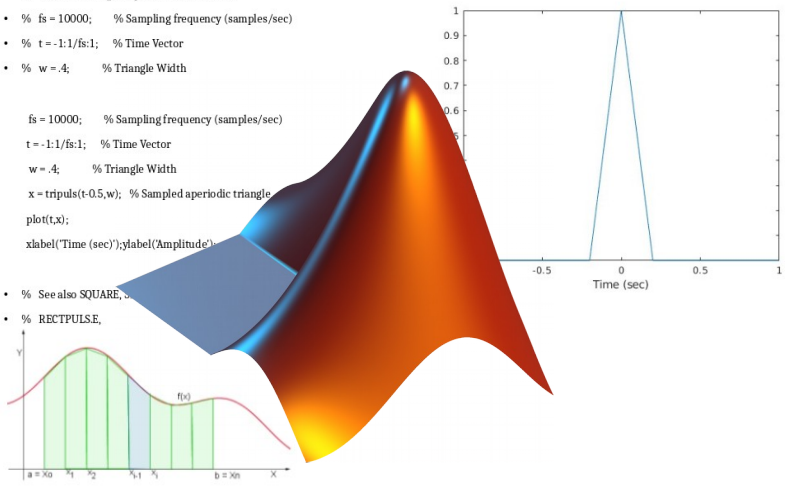
$$-f(a)+2\sum_{i=0}^{i=n-1} f(x_i) = -f(a)+2\left(\sum_{i=0}^{i=n} f(x_i) - f(b)\right) = -f(a)-2f(b)+2\left(\sum_{i=0}^{i=n} f(x_i)\right)$$

$$\int_a^b f(x)dx \approx \frac{(b-a)}{n}\left(\sum_{i=0}^{i=n} f(x_i) - \frac{1}{2}f(a) - f(b)\right)$$

- tri.m
- % Create a triangular pulse with width 0.4.
- % fs = 10000; % Sampling frequency (samples/sec)
- % t = -1:1/fs:1; % Time Vector
- % w = .4; % Triangle Width

```
fs = 10000;      % Sampling frequency (samples/sec)
t = -1:1/fs:1;   % Time Vector
w = .4;          % Triangle Width
x = tripuls(t-0.5,w);  % Sampled aperiodic triangle
plot(t,x);
xlabel('Time (sec)');ylabel('Amplitude')
```

- % See also SQUARE, S
- % RECTPULSE,




https://it.mathworks.com/videos/matlab-overview-61923.html