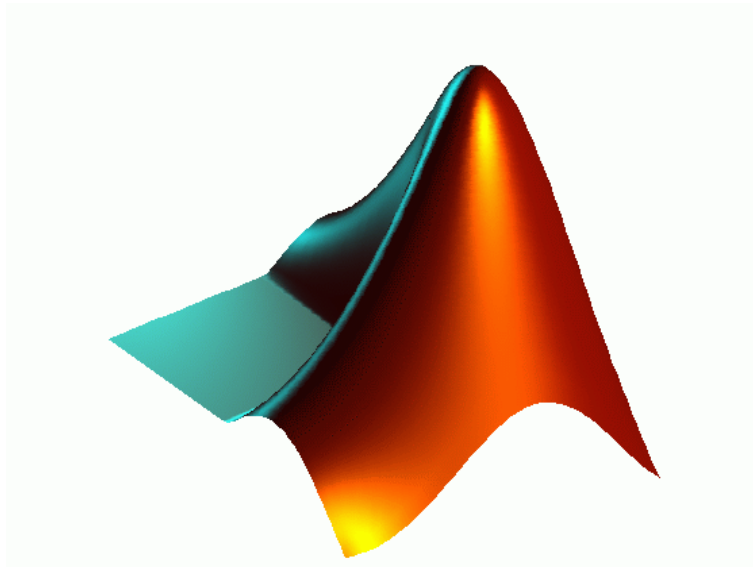


Laboratorio Matlab di Segnali e Sistemi

Mattia Zorzi

15 maggio 2012



Indice

1	Introduzione	5
2	Funzionalità basilari di Matlab	7
2.1	Come si presenta Matlab	7
2.2	Command Window	8
2.2.1	Definizione di oggetti	9
2.2.2	Operazioni di base sui vettori	12
2.2.3	Funzioni Matlab	14
2.2.4	Altre istruzioni matlab	18
2.3	Gestione delle variabili	18
2.4	m-files	20
2.5	Strutture di controllo	22
2.6	Grafici	24
3	Segnali, convoluzioni e serie di Fourier	29
3.1	Segnali pari e dispari, segnali complessi	29
3.2	Area ed energia di un segnale	30
3.3	Convoluzione	34
3.4	Serie di Fourier	39
4	Trasformata di Fourier	43
4.1	Trasformata di <i>Fourier</i> per segnali a tempo continuo	43
4.2	Trasformata di <i>Fourier</i> per segnali a tempo discreto	45
4.3	Aliasing	47

5	Sistemi LTI discreti	49
5.1	Sistemi LTI discreti	49
5.2	Risposta impulsiva	51
5.3	Risposta in frequenza	54
5.4	Autofunzioni dei filtri	55
5.5	Stima della periodicità dell'attività solare	58
6	Sistemi LTI continui	61
6.1	Equazioni differenziali in Matlab	61
6.2	Sistemi LTI continui	62

Capitolo 1

Introduzione

Questa dispensa contiene il materiale fornito agli studenti per il laboratorio del corso di Segnali e Sistemi per ingegneria dell'Informazione svolto all'università di Padova nell'a.a. 2011/2012. L'obiettivo principale del laboratorio è fornire una certa familiarità con lo strumento di calcolo *Matlab*, il quale è ampiamente usato in contesti ingegneristici e applicativi. L'obbiettivo secondario, ma non per questo meno importante, è permettere agli studenti di assimilare meglio i concetti fondamentali riguardanti la teoria dei segnali e dei sistemi visti a lezione.

Il termine *Matlab* deriva dall'abbreviazione di MATrix LABoratory. Fu creato alla fine degli anni '70 da Cleve Moler, il presidente del dipartimento di scienze informatiche dell'Università del Nuovo Messico. Egli creò Matlab per dare ai suoi studenti accesso alle librerie *linpack* e ad *eispack* senza che essi dovessero conoscere il *Fortran*. Presto si diffuse nelle altre Università e trovò un grande pubblico tra la comunità dei matematici applicati. Jack Little, un ingegnere, conobbe il programma durante una visita a Moler all'Università di Stanford nel 1983. Riconoscendo il suo potenziale commerciale, si unì con Moler e Steve Bangert. Essi riscrissero *Matlab* in C e fondarono la The MathWorks nel 1984 per continuare il suo sviluppo.

Matlab perciò è un ambiente per il calcolo numerico e l'analisi statistica che comprende anche l'omonimo linguaggio di programmazione creato dalla MathWorks. Questo software funziona su diversi sistemi operativi, tra cui *Windows*, *Mac OS*, *GNU/Linux* e *Unix*. Attualmente in commercio è disponibile la versione R2012a.

Infine è doveroso citare, in alternativa a *Matlab*, il software gratuito *Oc-*

tave. Il suo vantaggio è che è compatibile quasi completamente con Matlab.

Capitolo 2

Funzionalità basilari di Matlab

2.1 Come si presenta Matlab

Quando il programma viene richiamato appare la schermata di Figura 2.1:

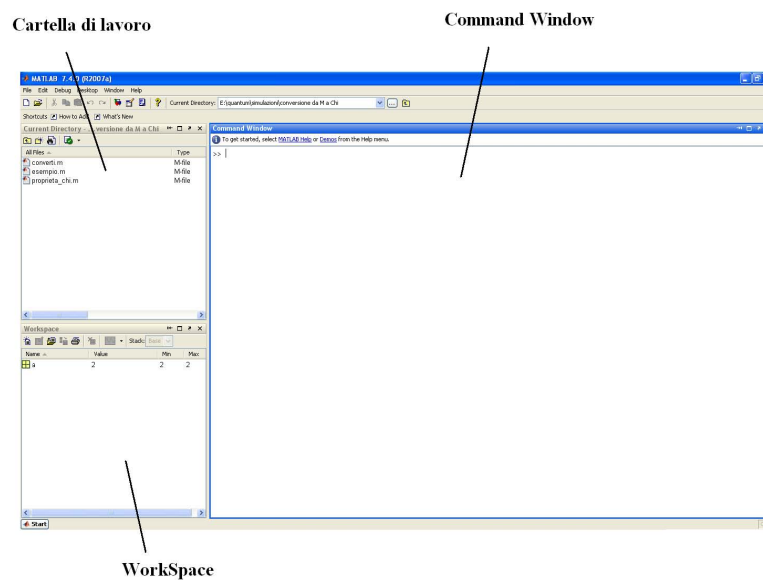


Figura 2.1: Schermata di Matlab

- Command Window è la sotto-finestra in cui vengono trasmesse le istruzioni desiderate a Matlab
- Workspace visualizza le variabili attualmente create dalle istruzioni fornite.
- La cartella di lavoro è la sotto-finestra che visualizza tutti i file contenuti nella cartella in cui ha accesso Matlab.

In Figura 2.2 è mostrata la barra dei pulsanti di Matlab. Il primo tasto serve per creare un file Matlab che chiameremo *m-file*. Il secondo tasto apre la cartella di lavoro. Il quartultimo apre il sottoprogramma *Simulink* che vedremo in seguito. L'ultimo tasto permettere di accedere alla guida in linea di Matlab. Affianco è possibile selezionare la cartella di lavoro desiderata.

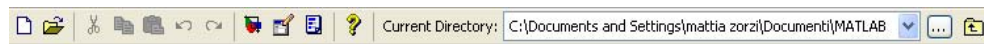


Figura 2.2: Barra dei pulsanti di Matlab

2.2 Command Window

In questa finestra appare un simbolo di attesa seguito da un cursore

```
>> _
```

Per fornire una istruzione a Matlab è sufficiente scrivere l'istruzione e premere invio. Se scriviamo un'istruzione inesistente, Matlab segnala, attraverso una frase in rosso, che il comando non esiste.

```
>> fhsj
??? Undefined function or variable 'fhsj'.
>>
```

Quando Matlab deve segnalare un errore utilizza sempre il colore rosso, per gli altri messaggi utilizza il nero. I comandi più semplici e intuitivi riguardano le operazioni di somma, sottrazione, moltiplicazione, divisione ed elevamento a potenza di numeri reali

```
>> 6.2+2-3.02^2*3/2

ans =

    -5.4806

>>
```

La variabile *ans* contiene il risultato dell'operazione. Per non visualizzare l'echo di un'istruzione è sufficiente scrivere punto e virgola dopo l'istruzione

```
>> 6.2+2-3.02^2*3/2;
>>
```

2.2.1 Definizione di oggetti

- Definizione di una variabile reale

```
a=3;
```

- Definizione di un numero complesso

```
>> z=7+i*9

z =

    7.0000 + 9.0000i

>>
```

Va osservato che i è una variabile predefinita come $\sqrt{-1}$ e in quanto tale può essere sovrascritta. È buona norma perciò non sovrascrivere i . Nota bene: anche j è considerata un'unità immaginaria.

- Definizione della variabile stringa s che contiene la parola ciao

```
>> s='ciao'

s =
```

```
ciao  
  
>>
```

- Definizione del vettore riga

$$v = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix} \quad (2.1)$$

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

- Definizione del vettore colonna

$$v = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad (2.2)$$

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

Il punto e virgola serve per indicare la fine di una riga.

- Definizione della matrice

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \quad (2.3)$$

```
>> A=[1 2 3; 4 5 6]

A =

     1     2     3
     4     5     6

>>
```

- Generazione del vettore

$$v = \left(5 \ 6 \ 7 \ 8 \ 9 \ 10 \right) \quad (2.4)$$

i cui elementi sono equispaziati di 1 in maniera crescente

```
>> v=5:10

v =

     5     6     7     8     9    10

>>
```

- Generazione del vettore

$$v = \left(0 \ 0.1 \ 0.2 \ 0.3 \ 0.4 \ 0.5 \ 0.6 \right) \quad (2.5)$$

i cui elementi sono equispaziati di 0.1 in maniera crescente

```
>> v=0:0.1:0.6

v =

Columns 1 through 4

     0    0.1000    0.2000    0.3000

Columns 5 through 7
```

```

0.4000    0.5000    0.6000
>>

```

2.2.2 Operazioni di base sui vettori

- Somma tra due vettori

$$u = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad v = \begin{pmatrix} 3 \\ 4 \end{pmatrix} \quad (2.6)$$

```

>> u=[1; 2];
>> v=[3; 4];
>> w=u+v

w =

     4
     6

>>

```

- Calcolo del prodotto interno $\langle u, v \rangle = u^T v$

```

>> u'*v

ans =

    11

>>

```

- Prodotto tra i vettori u e v componente per componente

```

>> v.*u

ans =

```

```

      3
      8

>>

```

Esiste anche l'analogo per la divisione (`./`) e l'elevamento a potenza (`.^`).

- Trasposizione del vettore u (solo nel caso in cui u è definito nel campo reale)

```

>> u'

ans =

      1      2

>>

```

- Selezione dell'elemento j -esimo del vettore

$$v = \begin{pmatrix} 3 & 4 & 5 & 6 \end{pmatrix}^T \quad (2.7)$$

```

>> v=[3; 4; 5; 6];
>> j=2;
>> v(j)

ans =

      4

>>

```

- Selezione di una sottosequenza del vettore

$$v = \begin{pmatrix} 1.1 & 2.2 & 3.3 & 4.4 & 5.5 & 6.6 \end{pmatrix} \quad (2.8)$$

secondo gli indici di posizione j, k

```
>> v=[1.1 2.2 3.3 4.4 5.5 6.6];  
>> j=2;  
>> k=5;  
>> v(j:k)  
  
ans =  
  
    2.2000    3.3000    4.4000    5.5000  
  
>>
```

- Polinomi: in matlab un polinomio è rappresentato da un vettore (riga o colonna). Per creare un polinomio è sufficiente immettere suoi coefficienti in un vettore, che chiameremo p , in ordine decrescente. Per esempio, consideriamo

$$x^5 + 10x^4 + 0.6x^3 + 12x^2 - 7x + 5 \quad (2.9)$$

```
>> p=[1 10 0.6 12 -7 5]  
  
p =  
  
Columns 1 through 4  
  
    1.0000   10.0000    0.6000   12.0000  
  
Columns 5 through 6  
  
   -7.0000    5.0000  
  
>>
```

2.2.3 Funzioni Matlab

Matlab fornisce molte funzioni all'utente. L'argomento delle funzioni, se presente, è compreso tra parentesi tonde.

Esempio 1. Vogliamo valutare la seguente funzione polinomiale per $x = 2$

$$p(x) = x^5 - 7x + 5. \quad (2.10)$$

In questo caso basta utilizzare la funzione *polyval* con argomenti di ingresso il corrispondente polinomio e il valore 2. Tale funzione restituisce $p(2)$.

```
>> p=[1 0 0 0 -7 5];  
>> valore=polyval(p',2)  
  
valore =  
  
    23  
  
>>
```

Inoltre non è necessario salvare il valore che restituisce la funzione

```
>> p=[1 0 0 0 -7 5];  
>> polyval(p',2)  
  
ans =  
  
    23  
  
>>
```

Per avere informazioni su una funzione Matlab è sufficiente scrivere *help* seguito dal nome della funzione

```
>> help polyval  
POLYVAL Evaluate polynomial.  
Y = POLYVAL(P,X) returns the value of a  
polynomial P evaluated at X. P is a vector of  
length N+1 whose elements are the coefficients  
of the polynomial in descending powers.  
  
    Y = P(1)*X^N + P(2)*X^(N-1) + ... + P(N)*X  
    + P(N+1)  
  
If X is a matrix or vector, the polynomial is
```

evaluated at all points in X . See POLYVALM for evaluation in a matrix sense.

`[Y,DELTA] = POLYVAL(P,X,S)` uses the optional output structure S created by POLYFIT to generate prediction error estimates DELTA. DELTA is an estimate of the standard deviation of the error in predicting a future observation at X by $P(X)$.

If the coefficients in P are least squares estimates computed by POLYFIT, and the errors in the data input to POLYFIT are independent, normal, with constant variance, then $Y \pm \text{DELTA}$ will contain at least 50\% of future observations at X .

`Y = POLYVAL(P,X,[],MU)` or `[Y,DELTA] = POLYVAL(P,X,S,MU)` uses $XHAT = (X - MU(1))/MU(2)$ in place of X . The centering and scaling parameters MU are optional output computed by POLYFIT.

Class support for inputs P, X, S, MU :
float: double, single

See also `polyfit`, `polyvalm`.

Overloaded methods:
`gf/polyval`

Reference page in Help browser
`doc polyval`

>>

Esempio 2. Vogliamo calcolare le radici del polinomio

$$x^5 - 7x + 5. \quad (2.11)$$

La funzione Matlab che dobbiamo utilizzare è *roots*. Il parametro d'ingresso è il polinomio e il parametro di uscita è un vettore contenente le radici del polinomio.

```
>> p=[1 0 0 0 -7 5];
>> roots(p)

ans =

    -1.7704
   -0.1618 + 1.6671i
   -0.1618 - 1.6671i
    1.3464
    0.7477

>>
```

Riportiamo alcune funzioni Matlab di interesse.

Funzioni costanti	
Funzione	Descrizione
pi	π
exp(1)	numero di Nepero
eps	Precisione di Matlab

Funzioni per numeri reali	
Funzione	Descrizione
round(x)	Restituisce l'approssimazione intera del numero reale x
rem(x,y)	Restituisce il resto della divisione $x \div y$
sign(x)	Restituisce il segno di x
sqrt(x)	Restituisce la radice quadrata di x
sin(x)	Restituisce il seno di x
cos(x)	Restituisce il coseno di x
tan(x)	Restituisce la tangente di x
asin(x)	Restituisce l'arcoseno di x
acos(x)	Restituisce l'arcocoseno di x
atan(x)	Restituisce l'arcotangente di x
log(x)	Restituisce il logaritmo di x
log10(x)	Restituisce il logaritmo in base dieci di x

Funzioni per numeri complessi

Funzione	Descrizione
<code>real(z)</code>	Restituisce la parte reale del numero complesso z
<code>imag(z)</code>	Restituisce la parte immaginaria del numero complesso z
<code>abs(z)</code>	Restituisce il modulo del numero complesso z
<code>phase(z)</code>	Restituisce la fase di z
<code>conj(z)</code>	Restituisce il complesso coniugato di z
<code>exp(z)</code>	Restituisce e^z

Funzioni per vettori e matrici

Funzione	Descrizione
<code>length(v)</code>	Restituisce la lunghezza del vettore v
<code>norm(v)</code>	Restituisce la norma euclidea di v
<code>sum(v)</code>	Restituisce la somma degli elementi di v
<code>wrev(v)</code>	Restituisce il vettore che ha gli elementi di v invertiti di ordine
<code>min(v)</code>	Restituisce il valore e la posizione dell'elemento più piccolo di v
<code>max(v)</code>	Restituisce il valore e la posizione dell'elemento più grande di v
<code>zeros(m,1)</code>	Genera un vettore di zeri di dimensione m
<code>ones(m,1)</code>	Genera un vettore di uno di dimensione m
<code>polyval(p,k)</code>	Restituisce il valore $p(k)$ dove $p(x)$ è un polinomio
<code>roots(p)</code>	Restituisce le radici del polinomio $p(x)$
<code>conv(p,q)</code>	Restituisce il polinomio risultante dal prodotto dei polinomi $p(x), q(x)$

2.2.4 Altre istruzioni matlab

Istruzione	Descrizione
<code>disp(s)</code>	Visualizza l'oggetto s nel Command Window
<code>clc</code>	Pulisce la finestra Command Window
<code>num2str(x)</code>	Restituisce la variabile stringa che corrisponde alla variabile reale/complessa x
<code>pause</code>	Arresta l'inserimento di istruzioni nel Command Window, l'effetto termina premendo INVIO
<code>%</code>	Simbolo per i commenti

2.3 Gestione delle variabili

Ogni variabile definita nel Command Window viene memorizzata nel Work-Space. L'istruzione *whos* permette la visualizzazione delle variabili del Work-Space

```
>> whos
```

Name	Size	Bytes	Class	Attributes
A	2x4	64	double	
ans	1x1	8	double	
v	1x7	56	double	

>>

L'istruzione *clear* permette la cancellazione di una o tutte le variabili del Workspace

>> whos				
Name	Size	Bytes	Class	Attributes
A	2x4	64	double	
ans	1x1	8	double	
v	1x7	56	double	
>> clear A				
>> whos				
Name	Size	Bytes	Class	Attributes
ans	1x1	8	double	
v	1x7	56	double	
>> clear all				
>> whos				
>>				

Chiudendo Matlab tutte le variabili del Workspace vengono perse. Per salvare tutte le variabili del Workspace si ricorre all'istruzione *save* seguita dal nome del file su cui verranno salvate

```
>> save risultati
>>
```

dopo questa istruzione Matlab crea il file **risultati.mat** nella cartella di lavoro. In questa maniera possiamo uscire dall'ambiente Matlab senza perdere i nostri dati. Per ricaricare tutte le variabili salvate nel file *.mat* si utilizza il comando *load*

```
>> load risultati
```

```
>>
```

2.4 m-files

Finora abbiamo visto come sia possibile utilizzare Matlab tramite la riga di comando. Può succedere però che si vogliano scrivere veri e propri programmi, cioè una successione di comandi da eseguire uno dopo l'altro in un ordine fissato. In tal caso la successione di comandi deve essere memorizzata in un file chiamato *m-file*. Per creare un *m-file* è sufficiente selezionare il primo tasto della barra dei pulsanti di Matlab, Figura 2.2. Si aprirà un editor, Figura 2.3, dove è possibile scrivere la sequenza di istruzioni. Terminata questa fase

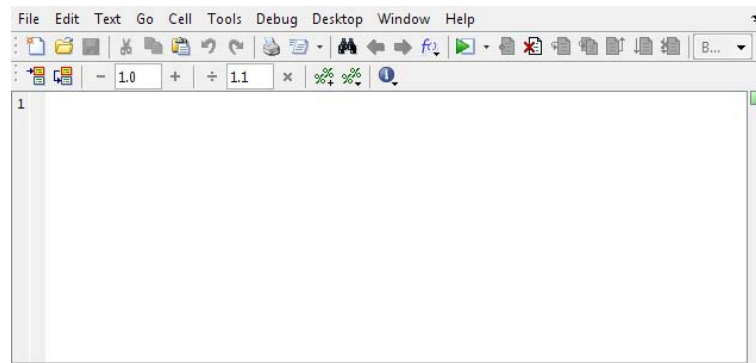


Figura 2.3: Editor per gli *m-file*.

dobbiamo salvare il lavoro. Supponiamo di chiamarlo **prova**, allora Matlab creerà un file *prova.m* nella cartella di lavoro. Infine per eseguire la sequenza di istruzioni salvate in *prova.m* è sufficiente scrivere il nome del file, senza estensione, nel Command Window

```
>>prova
>>
```

Esempio 3. Vogliamo scrivere il programma **es3.m** che calcola la media e la deviazione standard del vettore

$$x = \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix}^T. \quad (2.12)$$

```

1 %%%%%%%%%%
2 %  es3.m  %
3 %%%%%%%%%%
4
5 clear all % cancello il vecchio Workspace
6 clc% pulisci lo schermo
7 x=[1 2 3 4]';
8 media=sum(x)/length(x);
9 diff=x-media; % vettore differenza
10 dev_std=sqrt(sum(diff.^2)/length(x));
11 disp(['Media ' num2str(media)]) %stampo risultati
12 disp(['Deviazione standard ' num2str(dev_std)])

```

Eseguendo `es3.m` otteniamo

```

Media 2.5 Deviazione standard 1.118
>>

```

Esempio 4. Vogliamo creare il programma `es4.m` che salva i due vettori

$$u = \begin{pmatrix} 4 & 7 & 9 \end{pmatrix}^T, \quad v = \begin{pmatrix} 0 & 1 & 3 \end{pmatrix}^T \quad (2.13)$$

nel file `dati.mat`

```

1 %%%%%%%%%%
2 %  es4.m  %
3 %%%%%%%%%%
4
5 clear all
6 u=[4 7 9]'; %genero i vettori u,v
7 v=[0; 1; 3];
8 save dati % salvo le variabili del Workspace

```

Esempio 5. Utilizzando i vettori dell'esercizio precedente, calcolare $u^T v$

```

1 %%%%%%%%%%
2 %  es5.m  %
3 %%%%%%%%%%
4
5 clear all %cancello tutte le variabili Workspace
6 load dati %carico il vecchio Workspace
7 u'*v

```

2.5 Strutture di controllo

Oltre agli operatori aritmetici esistono gli operatori relazionali $<$, $>$, $==$, $<=$, $>=$, \sim che corrispondono ai simboli matematici $<$, $>$, $=$, \leq , \geq , \neq . Mediante questi operatori si possono scrivere delle relazioni che possono essere vere o false. Il valore vero viene indicato da Matlab con 1, mentre il valore falso viene indicato con 0.

- Struttura *if*

```

1  if [condizione]
2      [istruzioni]
3  else
4      [istruzioni]
5  end

```

- Struttura *for*

```

1  for j=min:max
2      [istruzioni]
3  end

```

- Struttura *while*

```

1  while [condizione]
2      [istruzioni]
3  end

```

Esempio 6. Scrivere un programma che somma due polinomi $p(x)$, $q(x)$. Tenere conto del fatto che i due polinomi possono avere grado diverso (e di conseguenza i corrispondenti vettori possono avere lunghezze diverse). Testare il programma con i seguenti polinomi

$$\begin{aligned}
 p(x) &= x^3 + 1 \\
 q(x) &= x^2 + 7x - 3.
 \end{aligned}
 \tag{2.14}$$

```

1  %%%%%%%%%%%
2  %  es6.m  %
3  %%%%%%%%%%%

```

```

4
5 clear all
6 clc %pulisci lo schermo
7 p=[1 0 0 1]; %polinomi da sommare
8 q=[2 7 -3];
9 if length(p)<length(q)
10     corto=p;
11     lungo=q;
12 else
13     corto=q;
14     lungo=p;
15 end
16 diff=length(lungo)-length(corto);
17 if diff>0
18     somma=[zeros(1,diff) corto]+lungo; %aggiungi zeri al polinomio
19                                         %piu' corto
20 else
21     somma=corto+lungo;
22 end
23 disp('Polinomio somma')
24 somma

```

Esempio 7. Scrivere un programma che stampa i primi dieci numeri interi che sono il quadrato di un numero intero. Fare in modo che ogni numero venga stampato dopo che l'utente ha digitato INVIO.

```

1 %%%%%%%%%%%
2 %  es7.m  %
3 %%%%%%%%%%%
4
5 clear all
6 clc %pulisci finestra Command Window
7 for n=1:10
8     disp(n^2)
9     pause
10 end

```

Esempio 8. Scrivere lo stesso programma dell'Esempio 7 utilizzando il controllo *while*.

```

1 %%%%%%%%%%%
2 %  es8.m  %
3 %%%%%%%%%%%
4
5 clear all
6 clc %pulisci finestra Command Window
7 n=1;
8 while n<=10
9     disp(n^2)
10    n=n+1;
11    pause
12 end

```

2.6 Grafici

Matlab fornisce anche funzioni grafiche. Consideriamo un segnale definito su un dominio $D \subset \mathbb{R}$

$$y = f(t), \quad t \in D. \quad (2.15)$$

Per disegnare tale segnale su carta:

- prendiamo un numero finito di punti t_1, \dots, t_n contenuti in D , equispaziati secondo una prescelta quantità T
- calcoliamo i corrispondenti y_1, \dots, y_n con $y_i = f(t_i)$
- disegniamo i punti sul piano cartesiano e infine li uniamo secondo un opportuno criterio.

Detto in maniera più formale: campioniamo il segnale con passo di campionamento pari a T e poi lo interpoliamo linearmente. Il concetto è lo stesso per stampare in Matlab tale segnale. Consideriamo il vettore

$$t = \begin{pmatrix} t_1 & \dots & t_n \end{pmatrix}^T, \quad t_i \in D, \quad (2.16)$$

calcoliamo

$$y = \begin{pmatrix} f(t_1) & \dots & f(t_n) \end{pmatrix}^T. \quad (2.17)$$

Infine utilizziamo la funzione *plot* passandole i vettori t e y .

Esempio 9. Stampare il segnale

$$f(t) = \sin\left(\frac{2\pi}{5}t - 0.1\right), \quad t \in [0, 10] \quad (2.18)$$

con passo di campionamento pari a $T = 0.02$.

```

1 %%%%%%%%%%
2 %  es9.m  %
3 %%%%%%%%%%
4
5 clear all
6 T=0.02; %passo di campionamento
7 t=0:T:10; % crea il vettore t
8 for j=1:length(t) % crea il vettore y
9     y(j)=sin(2*pi/5*t(j)-0.1);
10 end
11 plot(t,y)
12 xlabel('t') % nome asse x
13 ylabel('f(t)') % nome asse y
14 axis([0 10, -1 1]) % assi del grafico
15 grid on % griglia grafico

```

Eseguendo il programma otteniamo il grafico di Figura 2.4. Matlab oltre alla figura fornisce una serie utility grafiche: salva figura, zoom grafico, esplora grafico, datatip. Osserviamo che in questo *m-file* ci sono delle nuove funzioni di Matlab. Nella seguente tabella ne riportiamo la descrizione.

Funzione	Descrizione
figure	Apre una nuova finestra per stampare grafici
close all	Chiude tutte le finestre dei grafici aperte
grid on	Mostra la griglia nel grafico
grid off	Non mostra la griglia nel grafico (default)
xlabel(s)	Chiama l'asse x con la variabile stringa <i>s</i>
ylabel(s)	Chiama l'asse y con la variabile stringa <i>s</i>
legend()	Istruzione per la legenda del grafico
title(s)	Intitola il grafico usando la variabile stringa <i>s</i>
axis([x1 x2,y1 y3])	Imposta gli assi del grafico secondo (x_1, x_2) e (y_1, y_2)
hold on	Permette di sovrapporre più grafici nella stessa finestra
hold off	Permette di stampare solo un grafico nella finestra (default)
stem	Istruzione per stampare segnali a tempo discreto

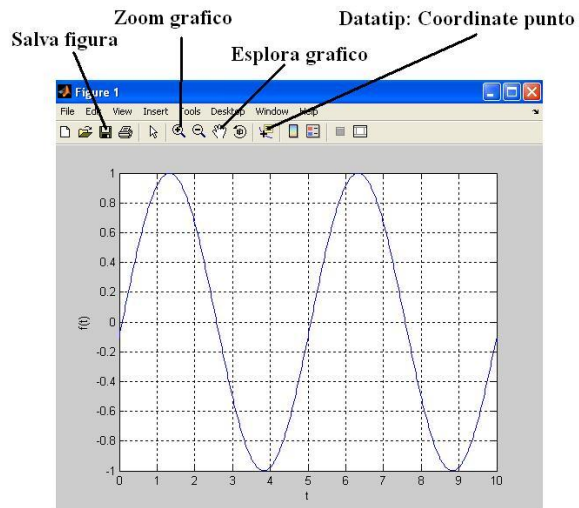


Figura 2.4: Grafico dell'Esempio 9

Esercizio 1. Consideriamo il segnale dell'Esempio 9. Ripetere l'esempio con $T = 0.8$ e $T = 2$. Commentare i grafici trovati.

Esercizio 2. Stampare il segnale

$$f(t) = e^{-0.01t} \sin(0.4t), \quad t \in [0, 100]. \quad (2.19)$$

Scegliere in maniera opportuna il passo di campionamento.

Esempio 10. Stampare su due figure differenti i segnali

$$\begin{aligned} f(t) &= \cos(2t), \quad t \in [0, 50] \\ g(t) &= e^{-0.1t} \cos(2t) \end{aligned} \quad (2.20)$$

con passo di campionamento pari a $T = 0.02$.

```

1 %%%%%%%%%%%
2 %  es10.m  %
3 %%%%%%%%%%%
4
5 clear all
6 close all % chiudo tutte le finestre aperte

```

```

7  T=0.02; %passo di campionamento
8  t=0:T:50; % crea il vettore t
9  y=cos(2*t); % crea il vettore per f
10 z=exp(-0.1*t).*cos(2*t); % crea il vettore per g
11 plot(t,y) %stampo il primo grafico
12 xlabel('t') % nome asse x
13 ylabel('f(t)') % nome asse y
14 title('primo grafico') % titolo grafico
15 figure %apro una nuova figura
16 plot(t,z,'r') %stampo il secondo grafico
17 xlabel('t') % nome asse x
18 ylabel('g(t)') % nome asse y
19 title('secondo grafico') % titolo grafico

```

In questo caso abbiamo utilizzato una scrittura molto più compatta e veloce per creare il vettore y . La funzione `cos` accetta come parametro di ingresso anche un vettore e restituisce il corrispondente vettore y . Vale la stessa cosa per `exp` e tutte le altre funzioni analoghe di Matlab. Osserviamo che l'istruzione `plot` prevede un terzo parametro opzionale per specificare il colore del segnale: `r`, `b`, `c`, `k`, `g` corrispondono a rosso, blu, ciano, nero, verde.

Esercizio 3. Facendo riferimento all'Esempio 10, vedere cosa succede quando:

- tolgo la riga quindici
- sostituisco la riga quindici con `hold on`.

Capitolo 3

Segnali, convoluzioni e serie di Fourier

3.1 Segnali pari e dispari, segnali complessi

Iniziamo ad applicare le funzionalità di Matlab calcolando la parte pari e dispari di un segnale.

Esempio 11. Scrivere un programma che stampa il segnale

$$f(t) = e^{\sin(2t)}, \quad t \in [-5, 5], \quad (3.1)$$

la sua parte pari e dispari (il tutto sullo stesso grafico).

```
1 %%%%%%%%%%%
2 %  es11.m  %
3 %%%%%%%%%%%
4
5 clear all
6 close all
7 clc
8 T=0.02; %passo di campionamento
9 t=-5:T:5; % crea il vettore t
10 y=exp(sin(2*t));
11 y_rib=wrev(y); % creo il vettore per la funzione
12               % ribaltata
13 y_pari=0.5*(y+y_rib); %vettore parte pari
14 y_disp=0.5*(y-y_rib); %vettore parte dispari
```

```

15 plot(t,y) % grafici
16 hold on
17 plot(t,y_pari,'r')
18 plot(t,y_disp,'g')
19 xlabel('t')
20 legend('f(t)','parte pari','parte dispari') %legenda
21 grid on

```

In Figura 3.1 è mostrato il grafico che stampa Matlab.

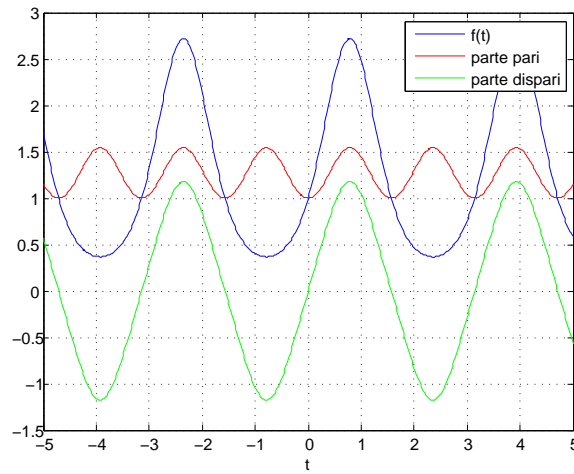


Figura 3.1: Grafico dell'Esempio 11

Esercizio 4. Stampare la parte reale e immaginaria del segnale complesso

$$f(t) = e^{j2\pi\bar{f}t}, \quad t \in [0, 0.5] \quad (3.2)$$

con $\bar{f} = 0.1$.

3.2 Area ed energia di un segnale

In questa sezione vediamo come calcolare l'area e l'energia (e quindi anche il valore medio e la potenza) di un dato segnale. Iniziamo con i segnali a tempo continuo.

Esempio 12. Scrivere un programma che stampa il segnale

$$f(t) = \text{sinc}(t), \quad t \in \mathbb{R} \quad (3.3)$$

e ne calcola l'area.

Chiaramente possiamo solo stampare il segnale su un sotto-intervallo limitato di \mathbb{R} , per esempio su $[-M, M]$ con M sufficientemente grande. Scegliamo $M = 1000$ e $T = 0.02$. Sapendo che $\int_{-\infty}^{\infty} \text{sinc}(t)dt$ esiste, è lecito considerare la seguente approssimazione

$$\int_{-\infty}^{\infty} \text{sinc}(t)dt \simeq \sum_{k=-\frac{M}{T}}^{\frac{M}{T}} T \text{sinc}(kT), \quad (k \in \mathbb{Z}). \quad (3.4)$$

Infine utilizziamo la funzione *sinc* di Matlab.

```

1  %%%%%%%%%%
2  %  es12.m  %
3  %%%%%%%%%%
4
5  clear all
6  close all
7  clc
8  T=0.02; %passo di campionamento
9  M=1000;
10 t=-M:T:M; % crea il vettore t
11 y=sinc(t);
12 plot(t,y) %stampa grafico
13 xlabel('t')
14 ylabel('f(t)')
15 grid on
16 area=T*sum(y);%calcola area
17 disp(['Area del sinc: ' num2str(area)])

```

Esercizio 5. Consideriamo il segnale dell'Esempio 12. Calcolare il valore medio l'energia e la potenza di f .

Esempio 13. Scrivere un programma che stampa il segnale periodico (su un periodo)

$$f(t) = \sin^3(t), \quad t \in \mathbb{R} \quad (3.5)$$

e ne calcola l'energia.

La periodicità del segnale è $T_p = 2\pi$. Sapendo che $\int_0^{2\pi} |f(t)|^2 dt$ esiste ed è finito, per calcolare l'energia è lecita l'approssimazione

$$\int_0^{T_p} |f(t)|^2 dt \simeq \sum_{k=0}^{\lfloor \frac{T_p}{T} \rfloor} T |f(kT)|^2, \quad k \in \mathbb{Z} \quad (3.6)$$

dove T è il passo di campionamento.

```

1 %%%%%%%%%%%
2 %  es13.m  %
3 %%%%%%%%%%%
4
5 clear all
6 close all
7 clc
8 T=0.02; %passo di campionamento
9 Tp=2*pi; %periodicita'
10 t=0:T:Tp; % crea il vettore t
11 y=sin(t).^3;
12 plot(t,y) %stampa grafico
13 xlabel('t')
14 ylabel('f(t)')
15 grid on
16 E=T*sum(y.^2); %calcola area
17 disp(['Energia della funzione: ' num2str(E)])

```

Esercizio 6. Consideriamo il segnale dell'Esempio 13. Calcolare l'area, il valore medio e la potenza di $f(t)$.

Ora prendiamo in considerazione i segnali a tempo discreto.

Esempio 14. Scrivere un programma che stampa il segnale a dominio discreto

$$f(nT) = e^{-0.02nT} \sin(nT), \quad n \geq 0 \quad (3.7)$$

con $T = 1$ e ne calcola l'energia.

```

1  %%%%%%%%%%%
2  %   es14.m   %
3  %%%%%%%%%%%
4
5  clear all
6  close all
7  clc
8  T=1; %passo di campionamento
9  M=200;
10 n=0:M; % crea il vettore n
11 y=exp(-0.02*n*T).*sin(n*T);
12 stem(n,y) %stampa grafico
13 xlabel('n')
14 ylabel('f(nT)')
15 E=T*sum(y.^2);%calcola energia
16 disp(['Energia del segnale: ' num2str(E)])

```

Esercizio 7. Consideriamo il segnale dell'Esempio 14. Calcolare l'area, il valore medio e la potenza di f .

Esempio 15. Consideriamo il segnale periodico a dominio discreto

$$f(nT) = \sin\left(\frac{2\pi nT}{T_p}\right), \quad n \in \mathbb{Z} \quad (3.8)$$

con $T = 0.05$ e $T_p = 2$. Stampare f su un periodo e calcolarne l'area.

```

1  %%%%%%%%%%%
2  %   es15.m   %
3  %%%%%%%%%%%
4
5  clear all
6  close all
7  clc
8  T=0.05; %passo di campionamento
9  Tp=2; %periodicita'
10 n=0:Tp/T; % crea il vettore n
11 y=sin(2*pi*n*T/Tp);
12 stem(n,y) %stampa grafico

```

```

13 xlabel('n')
14 ylabel('f(nT)')
15 A=T*sum(y);% calcolo area
16 disp(['Area del segnale: ' num2str(A)])

```

In Figura 3.2 è mostrato il grafico che stampa Matlab.

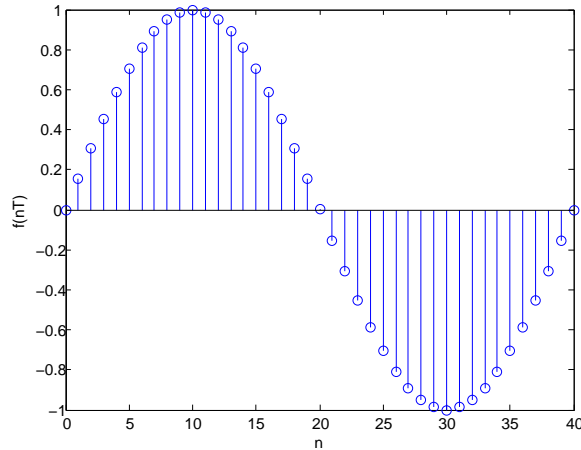


Figura 3.2: Grafico dell'Esempio 15

Esercizio 8. Consideriamo il segnale dell'Esempio 15. Calcolare l'energia di f .

3.3 Convoluzione

In questa sezione vediamo come calcolare la convoluzione tra segnali. Per tale motivo prendiamo in considerazione la risposta y di un sistema lineare (tempo invariante) ad un ingresso generico x . Dalla teoria sappiamo che y è la convoluzione tra risposta impulsiva h e l'ingresso x .

Esempio 16. Consideriamo un filtro lineare, causale con risposta impulsiva

$$h(t) = e^{\alpha t} + e^{\bar{\alpha} t}, \quad t \geq 0. \quad (3.9)$$

Supponiamo di alimentare il filtro con un gradino unitario x all'istante zero. Stampare la risposta impulsiva e la risposta y al gradino unitario per i

seguenti valori di α :

$$\begin{aligned}\alpha &= -0.4 + j0.6 \\ \alpha &= j0.6 \\ \alpha &= 0.04 + j0.6.\end{aligned}\tag{3.10}$$

Guardando le risposte impulsive e le corrispondenti risposte al gradino, discutere la stabilità dei sistemi.

Sappiamo che

$$y(t) = \int_{-\infty}^{\infty} h(t - \tau)x(\tau)d\tau = \int_0^t h(t - \tau)x(\tau)d\tau.\tag{3.11}$$

Sotto ipotesi simili a quelle dell'Esempio 12, fissando un T sufficientemente piccolo abbiamo la seguente approssimazione

$$y(t) \simeq T \sum_{k=0}^{\lfloor \frac{t}{T} \rfloor} h(t - kT)x(kT).\tag{3.12}$$

In altre parole possiamo approssimare y come il risultato di una convoluzione discreta. In Matlab la funzione di convoluzione discreta è *conv(x,h)*.

```

1 %%%%%%%%%%%
2 %  es16.m  %
3 %%%%%%%%%%%
4
5 clear all
6 close all
7 clc
8 T=0.001;
9 alpha=-0.4+j*0.6;
10 % alpha=j*0.6;
11 % alpha=0.04+j*0.6;
12 t=0:T:30;
13 h=exp(alpha*t)+exp(conj(alpha)*t);
14 plot(t,h) % stampa risposta impulsiva
15 title('risposta impulsiva')
16 x=ones(1,length(t)); %ingresso
17 y=T*conv(x,h); %uscita del filtro

```

```

18 y=[y(1:length(t))];%riaggiustamento del vettore
19 figure
20 plot(t,y)
21 title('uscita del filtro')

```

Nella Figura 3.3 sono mostrate le risposte al gradino per i diversi α . La valutazione e i commenti riguardanti la stabilità vengono lasciati allo studente.

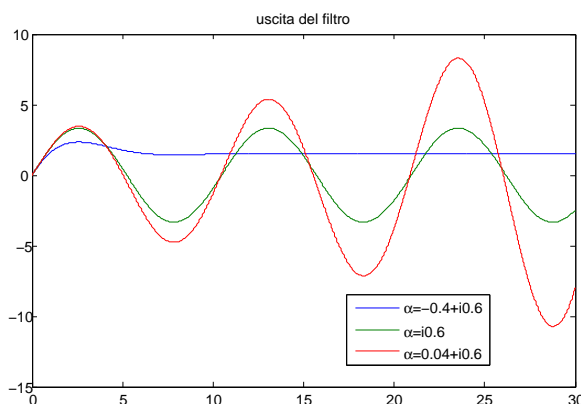


Figura 3.3: Risposte al gradino unitario del filtro dell'Esempio 16

Esercizio 9. Consideriamo un filtro lineare, causale con risposta impulsiva

$$h(t) = e^{\alpha_1 t} + e^{\bar{\alpha}_1 t} + e^{\alpha_2 t} + e^{\bar{\alpha}_2 t}, \quad t \geq 0. \quad (3.13)$$

Supponiamo di alimentare il filtro con un gradino unitario x all'istante zero. Calcolare la risposta impulsiva e la risposta y al gradino unitario per i seguenti valori di α_1 e α_2 :

$$\begin{aligned}
 \alpha_1 &= -0.4 + j0.6, \quad \alpha_2 = -0.4 + j0.6 \\
 \alpha_1 &= -0.4 + j0.6, \quad \alpha_2 = j0.6 \\
 \alpha_1 &= -0.4 + j0.6, \quad \alpha_2 = 0.04 + j0.6 \\
 \alpha_1 &= j0.6, \quad \alpha_2 = j0.6 \\
 \alpha_1 &= j0.6, \quad \alpha_2 = 0.04 + j0.6 \\
 \alpha_1 &= 0.04 + j0.6, \quad \alpha_2 = 0.04 + j0.6.
 \end{aligned} \quad (3.14)$$

Guardando le risposte impulsive e le corrispondenti risposte al gradino, discutere la stabilità dei sistemi.

Esempio 17. Consideriamo un filtro lineare discreto, causale con risposta impulsiva

$$h(n) = \alpha^n + \bar{\alpha}^n, \quad n \geq 0. \quad (3.15)$$

Supponiamo di alimentare il filtro con un gradino unitario discreto x all'istante zero. Calcolare la risposta impulsiva e la risposta y al gradino x per i seguenti valori di α :

$$\begin{aligned} \alpha &= 0.8 + j0.4 \\ \alpha &= 0.8 + j0.6 \\ \alpha &= 0.8 + j0.63. \end{aligned} \quad (3.16)$$

Guardando le risposte impulsive e le corrispondenti risposte al gradino, discutere la stabilità dei sistemi.

Sappiamo che

$$y(n) = \sum_{k=-\infty}^{\infty} h(n-k)x(k) = \sum_{k=0}^n h(n-k)x(k). \quad (3.17)$$

Anche in questo caso utilizziamo la funzione *conv*.

```

1  %%%%%%%%%%%
2  %   es17.m   %
3  %%%%%%%%%%%
4
5  clear all
6  close all
7  clc
8  alpha=0.8+j*0.4;
9  % alpha=0.8+j*0.6;
10 % alpha=0.8+j*0.63;
11 n=0:1:80;
12 h=alpha.^n+conj(alpha).^n;
13 % stem(t,h) % stampa risposta impulsiva
14 % title('risposta impulsiva')
15 x=ones(1,length(n)); %ingresso
16 y=conv(x,h); %uscita del filtro

```

```

17 y=[y(1:length(n))];%riaggiustamento del vettore
18 % figure
19 stem(n,y)
20 title('uscita del filtro')

```

Nella Figura 3.4 sono mostrate le risposte al gradino per i diversi α . La valutazione e i commenti riguardanti la stabilità vengono lasciati allo studente.

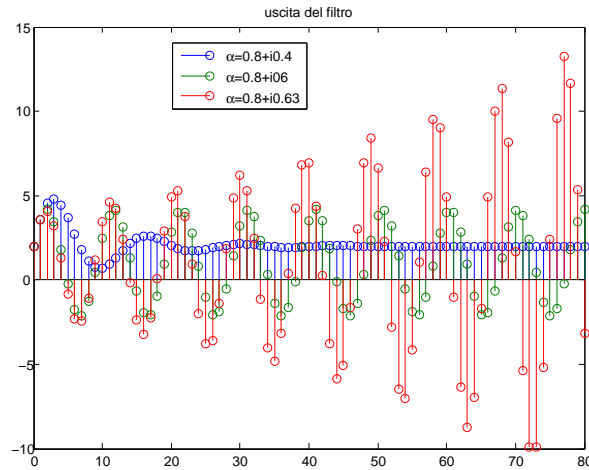


Figura 3.4: Risposte al gradino unitario del filtro dell'Esempio 17

Esercizio 10. Consideriamo un filtro lineare discreto, causale con risposta impulsiva

$$h(n) = \alpha_1^n + \bar{\alpha}_1^n + \alpha_2^n + \bar{\alpha}_2^n, \quad n \geq 0. \quad (3.18)$$

Supponiamo di alimentare il filtro con un gradino unitario discreto x all'istante zero. Calcolare la risposta impulsiva e la risposta y al gradino x per i seguenti valori di α_1 e α_2 :

$$\begin{aligned}
 \alpha_1 &= 0.8 + j0.4, \quad \alpha_2 = 0.8 + j0.4 \\
 \alpha_1 &= 0.8 + j0.4, \quad \alpha_2 = 0.8 + j0.6 \\
 \alpha_1 &= 0.8 + j0.4, \quad \alpha_2 = 0.8 + j0.63 \\
 \alpha_1 &= 0.8 + j0.6, \quad \alpha_2 = 0.8 + j0.6 \\
 \alpha_1 &= 0.8 + j0.6, \quad \alpha_2 = 0.8 + j0.63 \\
 \alpha_1 &= 0.8 + j0.63, \quad \alpha_2 = 0.8 + j0.63.
 \end{aligned} \quad (3.19)$$

Guardando le risposte impulsive e le corrispondenti risposte al gradino, discutere la stabilità dei sistemi.

3.4 Serie di Fourier

In questa sezione parleremo della serie di *Fourier*. Dato un segnale periodico $x(t)$ di periodo T_p a energia finita sul periodo, possiamo sempre esprimerlo come

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega_0 t} \quad (3.20)$$

dove

$$a_k = \frac{1}{T_p} \int_0^{T_p} x(t) e^{-jk\omega_0 t} dt, \quad \omega_0 = \frac{2\pi}{T_p}. \quad (3.21)$$

Esempio 18. Illustrare l'effetto Gibbs nello sviluppo della serie di Fourier troncata di un'onda quadra

$$x(t) = A_0 \text{rep}_{T_p} \text{rect} \left(\frac{t}{dT_p} \right) \quad (3.22)$$

con duty cycle $d = 0.3$, periodo $T_p = 2$ e ampiezza $A_0 = 7$. Troncare lo sviluppo a $N = 10, 30, 50$ armoniche.

In questo caso è facile vedere che

$$a_k = A_0 d \text{sinc}(kd). \quad (3.23)$$

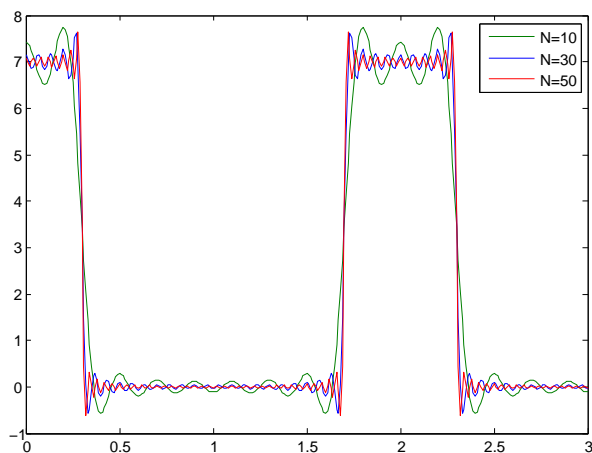
```

1 %%%%%%%%%%
2 %  es18.m  %
3 %%%%%%%%%%
4
5 clear all
6 close all
7 Tp=2;
8 d=0.3;
9 A0=7;
10 N=10;
11 % N=30;
12 % N=50;
```

```

13
14 %calcolo dei coefficienti di Fourier
15 a0=A0*d*sinc(0);
16 for k=1:N
17     a_piu(k)=A0*d*sinc(k*d); % a_n per n>0
18     a_meno(k)=A0*d*sinc(-k*d); % a_n per n<0
19 end
20 omega0=2*pi/Tp;
21 %calcolo della serie di Fourier
22 t=0:0.01:3;
23 x=a0*ones(1,length(t));
24 for k=1:N
25     x=x+a_meno(k)*exp(-j*k*omega0*t)+a_piu(k)*exp(j*k*omega0*t);
26 end
27 %stampa segnale
28 plot(t,x)

```

Figura 3.5: Serie di *Fourier* troncate dell'Esempio 18

In Figura 3.5 sono mostrate le corrispondenti serie troncate.

Esempio 19. Consideriamo la serie di *Fourier*

$$x(t) = 2 \sum_{k=1}^{\infty} (-1)^{k-1} \frac{1}{k} \sin(kt), \quad t \in \mathbb{R}. \quad (3.24)$$

Stampare la corrispondente serie di *Fourier* troncata per $N = 10, 30, 50$.

```

1  %%%%%%%%%%%
2  %  es19.m  %
3  %%%%%%%%%%%
4
5  clear all
6  close all
7  N=10;
8  % N=30;
9  % N=50;
10 %calcolo della serie di Fourier
11 t=0:0.01:20;
12 x=zeros(1,length(t));
13 for k=1:N
14     x=x+2*(-1)^(k-1)/k*sin(k*t);
15 end
16 %stampa segnale
17 plot(t,x)

```

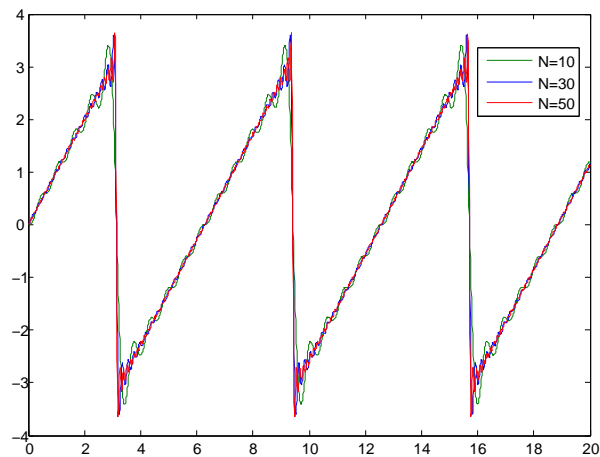


Figura 3.6: Serie di *Fourier* troncate dell'Esempio 19

In Figura 3.6 sono mostrate le corrispondenti serie troncate.

Esercizio 11. Consideriamo il segnale, periodico di periodo $T_p = 2$, così definito per $t \in [0, 2)$:

$$x(t) = \begin{cases} 2(1-t), & 0 \leq t < 1 \\ 2, & 1 \leq t < 2. \end{cases} \quad (3.25)$$

Stampare la corrispondente serie di *Fourier* troncata per $N = 10, 30, 50$.

Esercizio 11 Bis. Consideriamo la serie di *Fourier*

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jkt} \quad (3.26)$$

dove

$$a_k = \begin{cases} 0, & k, \text{ pari} \\ -\frac{2}{\pi k^2}, & k \text{ dispari.} \end{cases} \quad (3.27)$$

Stampare la corrispondente serie di *Fourier* troncata per $N = 1, 2, 3, 10, 30$.

Capitolo 4

Trasformata di Fourier

In questo capitolo affrontiamo il problema del calcolo delle trasformate di *Fourier* in Matlab.

4.1 Trasformata di *Fourier* per segnali a tempo continuo

Dato un segnale $x(t)$ a tempo continuo, la corrispondente trasformata di *Fourier* è definita come

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt, \quad \omega \in \mathbb{R}. \quad (4.1)$$

Se il precedente integrale esiste ed è finito, allora è lecita la seguente approssimazione

$$X(\omega) \simeq T \sum_{k=-M}^M x(kT)e^{-j\omega kT}, \quad \omega \in \mathbb{R} \quad (4.2)$$

dove T è il passo di campionamento e M è un numero sufficientemente elevato. Inoltre per stampare $X(\omega)$ dobbiamo anche campionare la pulsazione ω , quindi otteniamo

$$X(n\Omega) \simeq T \sum_{k=-M}^M x(kT)e^{-jkn\Omega T}, \quad n \in \mathbb{Z} \quad (4.3)$$

dove Ω è una quantità sufficientemente piccola.

Esempio 20. Stampare la trasformata di *Fourier* del segnale

$$x(t) = \frac{2}{\pi} \operatorname{sinc}\left(\frac{2}{\pi}t\right), \quad t \in \mathbb{R}. \quad (4.4)$$

```

1 %%%%%%%%%%%
2 %  es20.m  %
3 %%%%%%%%%%%
4
5 clear all
6 close all
7 M=100;
8 T=0.05; %passo di campionamento di x
9 Omega=0.05; %passo di campionamento di X
10 t=-M:T:M; %segnale
11 x=2/pi*sinc(2/pi*t);
12 omega=-10:Omega:10; %trasformata di Fourier
13 X=zeros(1,length(omega));
14 for k=1:length(t)
15     X=X+T*x(k)*exp(-j*omega*t(k));
16 end
17 %stampa trasformata
18 plot(omega,real(X))

```

Esercizio 12. Ripetere l'Esempio 20 con $T = 0.01$ e $\Omega = 0.01$. Quale fenomeno è possibile osservare?

Esercizio 13. Stampare la trasformata di *Fourier* del segnale

$$x(t) = e^{-2|t|}, \quad t \in \mathbb{R}. \quad (4.5)$$

Esercizio 13 Bis. Stampare il modulo della trasformata di *Fourier* per $-10 \leq \omega \leq 10$ dei seguenti segnali

$$x_1(t) = e^{-0.5|t|}, \quad t \in \mathbb{R} \quad (4.6)$$

$$x_2(t) = \cos(7t)e^{-0.5|t|}, \quad t \in \mathbb{R}. \quad (4.7)$$

Come si differenziano le due trasformate?

4.2 Trasformata di *Fourier* per segnali a tempo discreto

Consideriamo un segnale a tempo discreto

$$x(n), \quad n \in \mathbb{Z}. \quad (4.8)$$

La corrispondente trasformata di Fourier a tempo discreto è

$$X(e^{j\theta}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\theta n} \simeq \sum_{n=-M}^M x(n)e^{-j\theta n}, \quad \theta \in [-\pi, \pi) \quad (4.9)$$

dove M è un numero sufficientemente grande. Anche in questo caso dobbiamo campionare θ

$$X(e^{jk\Theta}) = \sum_{n=-M}^M x(n)e^{-jkn\Theta}, \quad k \in \mathbb{Z} \quad (4.10)$$

dove Θ è scelto sufficientemente piccolo.

Esempio 21. Stampare la trasformata di *Fourier* del segnale a dominio discreto

$$x(n) = \begin{cases} 1, & |n| \leq 10 \\ 0, & |n| > 10 \end{cases}. \quad (4.11)$$

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %   es21.m   %
3  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4
5  clear all
6  close all
7  clc
8  n=-10:10; %segnale
9  x=ones(1,length(n));
10 Theta=0.01; %trasformata di Fourier
11 theta=-pi:Theta:pi;
12 X=zeros(1,length(theta));
13 for k=1:length(n)
14     X=X+x(k)*exp(-j*theta*n(k));
15 end

```

```

16 plot(theta,real(X))%stampa trasformata
17 axis([-3.5 3.5,-5 25])

```

Esercizio 14. Stampare la trasformata di *Fourier* del segnale a dominio discreto

$$x(n) = \text{sinc}\left(\frac{3n}{\pi}\right), \quad n \in \mathbb{Z}. \quad (4.12)$$

Esempio 22. Si consideri il segnale dell'Esempio 21 e il segnale periodico

$$y(n) = \text{rep}_N x(n), \quad N = 40. \quad (4.13)$$

Calcolare i coefficienti di *Fourier* a_k , $-\frac{N}{2} \leq k \leq \frac{N}{2}$, di $y(n)$ e stampare sul grafico dell'Esempio 21 il segnale a tempo discreto

$$\frac{2\pi}{N}k \mapsto Na_k, \quad -\frac{N}{2} \leq k \leq \frac{N}{2}. \quad (4.14)$$

Cosa si può osservare dal grafico?

In questo caso

$$a_k = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} y(n) e^{-j\frac{2\pi}{N}kn}. \quad (4.15)$$

```

1 %%%%%%%%%%%
2 %  es22.m  %
3 %%%%%%%%%%%
4
5 % parte invariata dell'Esempio 21
6 clear all
7 close all
8 clc
9 n=-10:10; %segnale
10 x=ones(length(n));
11 Theta=0.01; %trasformata di Fourier
12 theta=-pi:Theta:pi;
13 X=zeros(1,length(theta));
14 for k=1:length(n)
15     X=X+x(k)*exp(-j*theta*n(k));
16 end

```

```

17 plot(theta,real(X))%stampa trasformata
18 axis([-3.5 3.5,-5 25])
19 % coefficienti di Fourier di y(n)
20 y=ones(1,length(n));
21 N=40;
22 k=-20:20;
23 for l=1:length(k)
24     a(l)=1/N*exp(-j*2*pi/N*k(l)*n)*y';
25 end
26 hold on
27 stem(2*pi/N*k,N*a,'r')

```

Esercizio 15. Ripetere l'Esempio 22 ponendo $\Theta = \frac{\pi}{20}$. Cosa succede? Dare una spiegazione teorica del risultato trovato.

4.3 Aliasing

Esempio 23. Consideriamo il segnale dell'Esempio 20. Dopo aver calcolato la banda (in maniera teorica) del segnale, ripetere l'esempio campionando $x(t)$ con i tre seguenti tempi di campionamento: $T = 0.5$, $T = 1$ e $T = 2$. Nota: Stampare $X(\omega)$ nell'intervallo $[-10, 10]$. Motivare, conoscendo la banda del segnale, i risultati trovati.

```

1 %%%%%%%%%%%
2 %  es23.m  %
3 %%%%%%%%%%%
4
5 clear all
6 close all
7 M=300;
8 T=0.5; %passo di campionamento di x
9 %T=1;
10 %T=2;
11 Omega=0.05; %passo di campionamento di X
12 t=-M:T:M; %segnale
13 x=2/pi*sinc(2/pi*t);
14 omega=-10:Omega:10; %trasformata di Fourier
15 X=zeros(1,length(omega));

```

```
16 for k=1:length(t)
17     X=X+T*x(k)*exp(-j*omega*t(k));
18 end
19 %stampa trasformata
20 plot(omega,real(X))
```

Esercizio 16. Consideriamo il segnale a banda infinita dell'Esercizio 13. Ripetere l'esercizio con $T = 0.1$, $T = 0.5$ e $T = 1$. Nota: Stampare $X(\omega)$ nell'intervallo $[-10, 10]$. Motivare, conoscendo la banda del segnale, i risultati trovati.

Capitolo 5

Sistemi LTI discreti

5.1 Sistemi LTI discreti

In questa sezione parliamo dei sistemi lineari discreti di tipo causale attraverso un esempio di sistema che descrive la dinamica di una certa popolazione. Sia $x(n)$ il numero di individui che costituiscono la popolazione all'istante n . In caso di risorse nulle la specie tende ad estinguersi con un tasso di mortalità pari ad m secondo la seguente equazione alle differenze

$$x(n) = mx(n-1). \quad (5.1)$$

Esempio 24. Si consideri una popolazione con $m = 0.8$ e sia $x(n)$ il numero di individui costituenti la popolazione all'istante n . Supponendo che attualmente il numero di individui sia pari a 50 ($x(0) = 50$), calcolare la dinamica della popolazione dall'istante 0 a 20.

```
1 %%%%%%%%%%%
2 %  es24.m  %
3 %%%%%%%%%%%
4
5 clear all
6 close all
7 clc
8 m=0.8;
9 x(1)=50; % condizione iniziale
10 for n=2:21
```

```

11      x(n)=m*x(n-1);
12  end
13  stem(0:20,x) %stampa grafico
14  xlabel('n')
15  ylabel('x(n)')

```

Ora supponiamo che ci siano delle risorse disponibili costanti che possono sostenere una popolazione di numerosità R . In questa situazione c'è un tasso di natalità dipendente dalla differenza fra la popolazione sostenibile dall'ambiente R e quella presente

$$x(n) = mx(n-1) + (R - x(n-1)). \quad (5.2)$$

Esempio 25. Si consideri una popolazione con $m = 0.8$, $R = 20$ e sia $x(n)$ il numero di individui costituenti la popolazione all'istante n . Supponendo che attualmente il numero di individui sia pari a 50, calcolare la dinamica della popolazione dall'istante 0 a 20.

```

1  %%%%%%%%%%%
2  %  es25.m  %
3  %%%%%%%%%%%
4
5  clear all
6  close all
7  clc
8  m=0.8;
9  R=20;
10 x(1)=50; %condizione iniziale
11 for n=2:21
12     x(n)=m*x(n-1)+(R-x(n-1));
13 end
14 stem(0:20,x) %stampa grafico
15 xlabel('n')
16 ylabel('x(n)')

```

Esercizio 17. Ora supponiamo che il numero di individui nati all'istante n dipenda dalla differenza fra la popolazione sostenibile dall'ambiente R e quella presente all'istante $n-2$ per un tasso di fertilità t . Il nuovo sistema diventa:

$$x(n) = mx(n-1) + (R - tx(n-2)). \quad (5.3)$$

Calcolare la dinamica della popolazione dall'istante 0 a 20 per i seguenti casi

$$\begin{aligned} x(-1) &= 10, \quad x(0) = 20, \quad m = 0.8, \quad t = 0.4, \quad R = 20 \\ x(-1) &= 10, \quad x(0) = 20, \quad m = 0.8, \quad t = 1, \quad R = 20. \end{aligned} \quad (5.4)$$

Guardando i grafici, a quale classe di esseri viventi può essere associata la dinamica del primo e del secondo caso rispettivamente?

5.2 Risposta impulsiva

Consideriamo il modello (5.3) della dinamica di una popolazione. Tale modello può essere visto come un sistema LTI discreto con ingresso costante $u(n) = R$:

$$x(n) = mx(n-1) - tx(n-2) + u(n). \quad (5.5)$$

Perciò, la risposta impulsiva $h(n)$ di tale modello si può ottenere imponendo $u(n) = \delta(n)$ con $x(-2) = 0$ e $x(-1) = 0$.

Esempio 26. Stampare la risposta impulsiva del modello (5.5) con

1. $m = 0.8$ e $t = 0.4$
2. $m = 0.8$ e $t = 1$.

Guardando i grafici prodotti, dire se i corrispondenti sistemi sono stabili o meno e verificare che le conclusioni trovate siano conformi ai risultati trovati nell'Esercizio 17.

```

1 %%%%%%%%%%%
2 %  es26.m  %
3 %%%%%%%%%%%
4
5 clear all
6 close all
7 clc
8 m=0.8;
9 t=0.4;
10 % t=1;
11 u=[0 0 1 zeros(1,20)]; %ingresso delta(n)
12 h(1)=0; % condizione iniziali nulle

```

```

13 h(2)=0;
14 for n=3:23
15     h(n)=m*h(n-1)+(u(n)-t*h(n-2));
16 end
17 stem(0:20,h(3:23)) %stampa grafico
18 xlabel('n')
19 ylabel('x(n)')

```

Esercizio 18. Consideriamo un modello a tempo discreto con ingresso x e uscita y che esegue la media mobile su tre punti:

$$y(n) = \frac{1}{3}(x(n) + x(n-1) + x(n-2)). \quad (5.6)$$

Calcolare la risposta impulsiva.

Ora supponiamo di voler calcolare la risposta x di un sistema lineare (causale) ad un ingresso generico u partendo da condizioni iniziali nulle. In questa situazione potremo applicare l'approccio di forza bruta dell'Esempio 25 oppure, avendo a disposizione la risposta impulsiva h , calcolare x tramite la convoluzione tra u e h .

Esempio 27. Consideriamo il modello (5.5), supponendo che le risorse disponibili varino secondo un bioritmo (ad esempio le stagioni):

$$u(n) = R \frac{\sin(\frac{n}{N_0})}{2} + S_0. \quad (5.7)$$

Calcolare la dinamica della popolazione dall'istante 0 a 60 partendo da condizioni iniziali nulle con $m = 0.6$, $t = 0.2$, $R = 5$, $N_0 = 2$ e $S_0 = 10$.

```

1 %%%%%%%%%%%
2 %  es27.m  %
3 %%%%%%%%%%%
4
5 clear all
6 close all
7 clc
8 m=0.6;
9 t=0.2;
10 R=5;

```

```

11  N0=2;
12  S0=10;
13  % calcolo della risposta impulsiva
14  u=[0 0 1 zeros(1,60)]; %ingresso delta(n)
15  h(1)=0; % condizione iniziali
16  h(2)=0;
17  for n=3:63
18      h(n)=m*h(n-1)-t*h(n-2)+u(n);
19  end
20  h=h(3:end); % risposta impulsiva dall'istante zero
21  %calcolo della risposta del sistema
22  n=0:60;
23  u=R*sin(n/N0)/2+S0;
24  x=conv(h,u);
25  x=x(1:61); % riaggiustamento del vettore
26  stem(0:60,x) %stampa grafico
27  xlabel('n')
28  ylabel('x(n)')

```

Esercizio 19. Consideriamo una popolazione di pesci da allevamento la cui dinamica è descritta dal modello (5.5) con $m = 0.6$, $t = 0.2$, $x(-2) = 0$ e $x(-1) = 0$. Inizialmente, dall'istante 0 a 39, i proprietari dell'allevamento decidono di non intervenire per modificare le risorse naturali che offre il laghetto che ospita questi pesci

$$u(n) = 5 \frac{\sin(\frac{n}{2})}{2} + 10, \quad 0 \leq n < 40. \quad (5.8)$$

Tuttavia, gli allevatori non sono soddisfatti del numero di pesci che producono e dall'istante $n = 40$ immettono nel lago una risorsa (che si aggiunge a quella naturale) che decresce di una unità all'aumentare di n

$$u(n) = 5 \frac{\sin(\frac{n}{N_0})}{2} + 10 + 80 - n, \quad 40 \leq n < 80. \quad (5.9)$$

Dopo questa decisione gli allevatori sono contenti, ma vorrebbero che la popolazione di questi pesci fosse costante nel tempo. Decidono perciò di costruire un impianto sofisticato che fornisce risorse costanti nel tempo alla popolazione di pesci

$$u(n) = 30, \quad n \geq 80. \quad (5.10)$$

Stampare u e la corrispondente dinamica della popolazione.

5.3 Risposta in frequenza

In questa sezione ci occupiamo del calcolo della risposta in frequenza di un sistema LTI causale discreto.

Esempio 28. Consideriamo il sistema LTI dell'Esempio 26. Vogliamo calcolare la risposta in frequenza per i due casi considerati. Stampare modulo e fase delle risposte in frequenza.

```

1  %%%%%%%%%%%
2  %   es28.m   %
3  %%%%%%%%%%%
4
5  clear all
6  close all
7  clc
8  m=0.8;
9  t=0.4;
10 %   t=1;
11 % calcolo della risposta impulsiva
12 u=[0 0 1 zeros(1,60)]; %ingresso delta(n)
13 h(1)=0; % condizioni iniziali
14 h(2)=0;
15 for k=3:63
16     h(k)=m*h(k-1)-t*h(k-2)+u(k);
17 end
18 h=h(3:end); %risposta impulsiva dall'istante 0
19 % risposta in frequenza
20 n=0:60;
21 Theta=0.01; %trasformata di Fourier
22 theta=-pi:Theta:pi;
23 H=zeros(1,length(theta));
24 for k=1:length(n)
25     H=H+h(k)*exp(-j*theta*n(k));
26 end
27 %stampa modulo risposta in frequenza
28 figure
29 plot(theta,abs(H))
30 xlabel('theta')
```

```

31 ylabel(' |H| ')
32 %stampa fase risposta in frequenza
33 figure
34 plot(theta, phase(H))
35 xlabel('theta')
36 ylabel('fase di H')

```

Esercizio 20. Si consideri il sistema dell'Esercizio 18. Stampare modulo e fase della risposta in frequenza $H(e^{j\theta})$.

5.4 Autofunzioni dei filtri

Consideriamo un sistema LTI causale BIBO-stabile con risposta impulsiva $h(n)$:

$$x(n) = h \star u(n). \quad (5.11)$$

Se $u(n) = \sin(\theta_0 n)$ e $h(n)$ è a valori reali, allora

$$x(n) = |H(e^{j\theta_0})| \sin(\theta_0 n + \arg[H(e^{j\theta_0})]). \quad (5.12)$$

Osservazione 5.1. *Tipicamente nelle applicazioni si prende in considerazione l'ingresso $\tilde{u}(n) = \sin(\theta_0 n)$ con $n > 0$ (e quindi $\tilde{u}(n) = 0$ per $n < 0$). In questo caso si può dimostrare che dopo un transitorio iniziale la risposta del sistema tende ad assomigliare a (5.12).*

Esempio 29. Consideriamo il sistema (5.5) con $m = 0.6$ e $t = 0.2$. Alimentiamo il sistema con la sinusoide di ampiezza unitaria e $\theta_0 = 1$

$$u(n) = \sin(\theta_0 n), \quad n \geq 0. \quad (5.13)$$

Stampare i tre seguenti grafici:

1. $u(n)$ e $x(n)$ (scegliamo $0 \leq n \leq 600$)
2. $|U(e^{j\theta})|$ e $|X(e^{j\theta})|$
3. $|H(e^{j\theta})|$

Verificare dai grafici che

$$|H(e^{j\theta_0})| \simeq \frac{|X(e^{j\theta_0})|}{|U(e^{j\theta_0})|}. \quad (5.14)$$

```

1  %%%%%%%%%%%
2  %  es29.m  %
3  %%%%%%%%%%%
4
5  clear all
6  close all
7  clc
8  m=0.6;
9  t=0.2;
10 n=0:600;
11 % segnale di ingresso u
12 theta0=1;
13 u=sin(theta0*n);
14 Theta=0.01; %trasformata di Fourier
15 theta=-pi:Theta:pi;
16 U=zeros(1,length(theta));
17 for k=1:length(n)
18     U=U+u(k)*exp(-j*theta*n(k));
19 end
20
21 %risposta impulsiva
22 d=[0 0 1 zeros(1,600)]; %ingresso delta(n)
23 h(1)=0; % condizioni iniziali
24 h(2)=0;
25 for k=3:n(end)+3
26     h(k)=m*h(k-1)-t*h(k-2)+d(k);
27 end
28 h=h(3:end); % risposta impulsiva dall'istante 0
29 %calcolo della risposta alla sinusoide u
30 x=conv(u,h);
31 Theta=0.01; %trasformata di Fourier
32 theta=-pi:Theta:pi;
33 X=zeros(1,length(theta));
34 for k=1:length(n)
35     X=X+x(k)*exp(-j*theta*n(k));
36 end
37
38 % risposta in frequenza

```

```

39 Theta=0.01; %trasformata di Fourier
40 theta=-pi:Theta:pi;
41 H=zeros(1,length(theta));
42 for k=1:length(n)
43     H=H+h(k)*exp(-j*theta*n(k));
44 end
45
46 % grafici di x e u
47 figure
48 stem(n,x(1:length(n)),'r')
49 hold on
50 stem(n,u,'b')
51 xlabel('n')
52 legend('x','u')
53
54 % grafici |X| e |U|
55 figure
56 plot(theta,abs(X),'r')
57 hold on
58 plot(theta,abs(U),'b')
59 xlabel('theta')
60 legend('|X|','|U|')
61
62 % grafico |H|
63 figure
64 plot(theta,abs(H),'b')
65 xlabel('theta')
66 legend('|H|')

```

Esercizio 21. Consideriamo l'Esempio 29. Stampare la fase di $H(e^{j\theta})$ e calcolare $\arg[H(e^{j\theta_0})]$. Verificare poi che nel primo grafico x e u sono sfasate tra loro di tale quantità.

Suggerimento: stampare sul primo grafico il segnale continuo $u(t) = \sin(\theta_0 t), t \geq 0, \dots$

5.5 Stima della periodicità dell'attività solare

Una macchia solare è una regione della superficie del Sole (la fotosfera) che è distinta da una temperatura minore dell'ambiente circostante, e da forte attività magnetica. Anche se in realtà le macchie solari sono estremamente luminose, perché hanno una temperatura di circa 5000 kelvin, il contrasto con le regioni circostanti, ancora più luminose grazie ad una temperatura di 6000 kelvin, le rende chiaramente visibili come macchie scure.

Esempio 30. A partire dal 1749, mensilmente viene misurato il numero di macchie che appaiono sulla superficie del Sole. Tali dati sono resi disponibili dal *Royal Observatory of Belgium*. Il file `Sdata.mat` contiene il vettore x di tutte queste misure e il corrispondente vettore n che tiene conto dei corrispondenti mesi (contati a partire dal partire dal 1749). Stampare $x(n)$ e il modulo della sua trasformata di Fourier $X(e^{j\theta})$.

In questo caso il vettore dei dati x può essere visto come un segnale $x(n)$ con supporto compatto: $x(n) \neq 0$ per ogni n tale che $1 \leq n \leq M$ dove M è la lunghezza del vettore x .

```

1  %%%%%%%%%%%
2  %   es30.m   %
3  %%%%%%%%%%%
4
5  clear all
6  close all
7  clc
8  load Sdata
9  stem(n,x) % grafico del numero di macchie solari
10 xlabel('mesi')
11 ylabel('numero di macchie solari')
12 axis([1 3136, 0 260])
13 Theta=0.01; %X(e^jtheta)
14 theta=-pi:Theta:pi;
15 X=zeros(1,length(theta));
16 for k=1:length(n)
17     X=X+x(k)*exp(-j*theta*n(k));
18 end
19 figure
20 plot(theta,abs(X))

```

```

21 xlabel('theta')
22 ylabel('|X|')

```

Dal primo grafico si nota che $x(n)$ aumenta e diminuisce periodicamente. Andando a vedere $|X(e^{j\theta})|$ si notano delle componenti periodiche per $\theta \simeq 0$. Queste, tuttavia, sono mascherate dalla componente continua $X(0)$.

Esempio 31. Consideriamo l'Esempio 30. Vogliamo stimare la componente periodica che abbiamo osservato in precedenza. Come osservato, la componente continua maschera la componente periodica. Per tale motivo eliminiamo la componente continua e consideriamo il segnale

$$y(n) = x(n) - \frac{1}{M} \sum_{k=1}^M x(k). \quad (5.15)$$

```

1 %%%%%%%%%%%
2 %  es31.m  %
3 %%%%%%%%%%%
4
5 clear all
6 close all
7 clc
8 load Sdata
9 y=x-1/length(x)*sum(x);
10 Theta=0.01; %Y(e^jtheta)
11 theta=-pi:Theta:pi;
12 Y=zeros(1,length(theta));
13 for k=1:length(n)
14     Y=Y+y(k)*exp(-j*theta*n(k));
15 end
16 figure
17 plot(theta,abs(Y))
18 xlabel('theta')
19 ylabel('|Y|')

```

Dal grafico di $|Y(e^{j\theta})|$ è facile vedere che l'armonica con potenza maggiore si ha per $\omega_{MAX} = \frac{2\pi}{T_{MAX}} \simeq 0.0484$. Quindi $T_{MAX} = 129,82$ mesi, in quanto ogni misura viene effettuata mensilmente. Concludiamo che questo fenomeno delle macchie solari avviene con una periodicità pari ad 10,82 anni (circa).

Capitolo 6

Sistemi LTI continui

6.1 Equazioni differenziali in Matlab

Consideriamo l'equazione differenziale del secondo ordine a parametri costanti

$$a \frac{d^2}{dt^2} y(t) + b_1 \frac{d}{dt} y(t) + b_0 y(t) = x(t) \quad (6.1)$$

dove $a \neq 0$ e $x(t)$ noto. Poniamo

$$v_1(t) = y(t), \quad v_2(t) = \frac{d}{dt} y(t) \quad (6.2)$$

quindi (6.1) diventa

$$\begin{cases} \frac{d}{dt} v_1(t) = v_2(t) \\ \frac{d}{dt} v_2(t) = -\frac{b_1}{a} v_2(t) - \frac{b_0}{a} v_1(t) + \frac{1}{a} x(t) \end{cases} \quad (6.3)$$

Esprimendo (6.3) in forma vettoriale

$$\begin{pmatrix} \frac{d}{dt} v_1(t) \\ \frac{d}{dt} v_2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\frac{b_0}{a} & -\frac{b_1}{a} \end{pmatrix} \begin{pmatrix} v_1(t) \\ v_2(t) \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{a} \end{pmatrix} x(t). \quad (6.4)$$

Infine, definendo

$$v(t) = \begin{pmatrix} v_1(t) \\ v_2(t) \end{pmatrix} \quad (6.5)$$

e

$$A = \begin{pmatrix} 0 & 1 \\ -\frac{b_0}{a} & -\frac{b_1}{a} \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ \frac{1}{a} \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad D = 0 \quad (6.6)$$

otteniamo il sistema

$$\begin{cases} \frac{d}{dt}v(t) = Av(t) + Bx(t) \\ y(t) = Cv(t) + Dx(t) \end{cases} \quad (6.7)$$

Osserviamo che la (6.7) è una forma equivalente di (6.1). Perciò, le matrici A, B, C, D descrivono in maniera univoca l'equazione differenziale (6.1).

Osservazione 6.1. Per l'equazione differenziale del primo ordine

$$a \frac{d}{dt}y(t) + b_0 y(t) = x(t), \quad a \neq 0 \quad (6.8)$$

abbiamo $v(t) = y(t)$, $A = -\frac{b_0}{a}$, $B = \frac{1}{a}$, $C = 1$, $D = 0$.

Esercizio 22. Si consideri l'equazione differenziale di ordine n descritta da

$$a \frac{d^n}{dt^n}y(t) + \sum_{k=0}^{n-1} b_k \frac{d^k}{dt^k}y(t) = cx(t), \quad a \neq 0. \quad (6.9)$$

Verificare che le corrispondenti matrici sono

$$A = \begin{pmatrix} 0 & 1 & 0 & \dots \\ & \ddots & \ddots & \\ & & 0 & 1 \\ -\frac{b_0}{a} & \dots & \dots & -\frac{b_{n-1}}{a} \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ \vdots \\ \vdots \\ \frac{1}{a} \end{pmatrix}, \quad C = \begin{pmatrix} 0 & \dots & 0 & 1 \end{pmatrix}, \quad D = 0. \quad (6.10)$$

In Matlab, per calcolare la soluzione ai valori iniziali $y(t)$ utilizziamo la funzione `lsim`. I parametri di ingresso sono: **A,B,C,D**, il vettore dei tempi t , il vettore x e il vettore delle condizioni iniziali $\begin{pmatrix} y(0) & \frac{d}{dt}y(0) \end{pmatrix}^T$. Nel seguito vedremo in dettaglio l'utilizzo di tale funzione con degli esempi.

6.2 Sistemi LTI continui

In questa sezione prendiamo in esame l'equazione differenziale che descrive la dinamica di un oscillatore armonico costituito da una molla di costante elastica k collegata ad una massa m , Figura 6.1. Sia $y(t)$ la posizione di un prefissato punto dell'oscillatore all'istante t . Se ad esso non è applicata nessuna forza esterna, il moto del punto è descritto dalla seguente equazione differenziale

$$m \frac{d^2}{dt^2}y(t) = -ky(t). \quad (6.11)$$

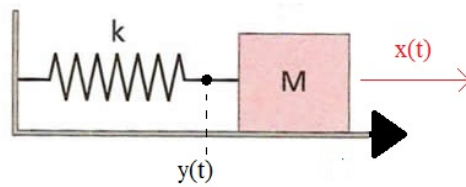


Figura 6.1: Modello molla-massa.

Esempio 32. Si consideri un oscillatore armonico con $m = 0.1\text{Kg}$ e $k = 1.6\frac{\text{N}}{\text{m}}$. Supponendo che il punto prefissato dell'oscillatore sia a 10m dalla posizione di riposo ($y(0) = 10$, $\frac{d}{dt}y(0) = 0$), calcolare la dinamica di tale punto dall'istante 0 a 10.

In questo caso

$$A = \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & 0 \end{pmatrix}, B = \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix}, C = \begin{pmatrix} 1 & 0 \end{pmatrix}, D = 0 \quad (6.12)$$

e $x(t) = 0$ in quanto l'oscillatore non è soggetto a nessuna forza.

```

1  %%%%%%%%%%%
2  % es32.m %
3  %%%%%%%%%%%
4
5  close all
6  clear all
7  clc
8  m=0.1;
9  k=1.6;
10 y0=[10; 0]; % condizioni iniziali
11 T=0.01;
12 t=0:T:10; % vettore dei tempi
13 x=zeros(1,length(t));% ingresso nullo
14 A=[0 1; -k/m 0];% definizione matrici
15 B=[0;1/m];
16 C=[1 0];
17 D=0;
18 y=lsim(A,B,C,D,x,t,y0); % dinamica modello
19 figure

```

20 `plot(t,y)`

Esercizio 23. Ripetere l'Esempio 32 supponendo che la posizione iniziale del punto prefissato sia 20m. Che cosa si può concludere da questi due esempi?

Ora supponiamo che sia presente l'attrito viscoso λ . L'equazione dell'oscillatore armonico smorzato è

$$m \frac{d^2}{dt^2} y(t) = -\lambda \frac{d}{dt} y(t) - ky(t). \quad (6.13)$$

Esempio 33. Si consideri un oscillatore armonico smorzato. Supponendo che il punto prefissato dell'oscillatore sia a 10m dalla posizione di riposo ($y(0) = 10$, $\frac{d}{dt}y(0) = 0$), calcolare la dinamica di tale punto dall'istante 0 a 10 per i due seguenti casi:

- $m = 0.1\text{Kg}$, $\lambda = 0.1 \frac{\text{Kg}}{\text{s}}$, $k = 1.6 \frac{\text{N}}{\text{m}}$.
- $m = 0.1\text{Kg}$, $\lambda = 0.9 \frac{\text{Kg}}{\text{s}}$, $k = 1.6 \frac{\text{N}}{\text{m}}$.

Cosa differenzia i due casi precedenti?

In questo caso

$$A = \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{\lambda}{m} \end{pmatrix}, B = \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix}, C = \begin{pmatrix} 1 & 0 \end{pmatrix}, D = 0 \quad (6.14)$$

e $x(t) = 0$ in quanto l'oscillatore non è soggetto a nessuna forza.

```

1 %%%%%%%%%%%
2 % es33.m %
3 %%%%%%%%%%%
4
5 close all
6 clear all
7 clc
8 m=0.1;
9 k=1.6;
10 lambda=0.1;
11 % lambda=0.9;
12 y0=[10; 0]; % condizioni iniziali
13 T=0.01; % vettore dei tempi

```

```

14 t=0:T:10;
15 x=zeros(1,length(t)); % ingresso nullo
16 A=[0 1; -k/m -lambda/m];% definizione matrici
17 B=[0; 1/m];
18 C=[1 0];
19 D=0;
20 y=lsim(A,B,C,D,x,t,y0); % dinamica modello
21 figure
22 plot(t,y)

```

Esercizio 24. Ripetere l'Esempio 33 supponendo che la posizione iniziale del punto prefissato sia 20m. Che cosa si può concludere da questi due esempi?

Ora supponiamo che sull'oscillatore agisca una forza costante dovuta alla presenza della gravità. In questo caso l'equazione dell'oscillatore armonico diventa

$$m \frac{d^2}{dt^2} y(t) = -\lambda \frac{d}{dt} y(t) - ky(t) + mg. \quad (6.15)$$

Esempio 34. Si consideri un modello molla-massa verticale in cui c'è la presenza della forza di gravità. Sia $m = 0.1\text{Kg}$, $\lambda = 0.1 \frac{\text{Kg}}{\text{s}}$, $k = 1.6 \frac{\text{N}}{\text{m}}$. Supponendo che il punto prefissato dell'oscillatore sia a 10m dalla posizione di riposo ($y(0) = 10$, $\frac{d}{dt}y(0) = 0$), calcolare la dinamica di tale punto dall'istante 0 a 20.

In questo caso

$$A = \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{\lambda}{m} \end{pmatrix}, B = \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix}, C = \begin{pmatrix} 1 & 0 \end{pmatrix}, D = 0. \quad (6.16)$$

```

1 %%%%%%%%%%%
2 % es34.m %
3 %%%%%%%%%%%
4
5 close all
6 clear all
7 clc
8 m=0.1;
9 k=1.6;
10 lambda=0.1;

```

```

11 g=9.8;
12 y0=[10; 0]; % condizioni iniziali
13 T=0.01; % vettori dei tempi
14 t=0:T:20;
15 x=m*g*ones(1,length(t)); % ingresso
16 A=[0 1; -k/m -lambda/m]; % definizione matrici
17 B=[0;1/m];
18 C=[1 0];
19 D=0;
20 y=lsim(A,B,C,D,x,t,y0); % dinamica modello
21 figure
22 plot(t,y)

```

Esercizio 25. Ripetere l'Esempio 34 con $\lambda = 0.9 \frac{\text{Kg}}{\text{s}}$. Perchè la dinamica cambia in maniera rilevante?

Esempio 35. Consideriamo il modello

$$m \frac{d^2}{dt^2} y(t) = -\lambda \frac{d}{dt} y(t) - ky(t) + x(t). \quad (6.17)$$

con $m = 0.1 \text{Kg}$, $\lambda = 0.1 \frac{\text{Kg}}{\text{s}}$, $k = 1.6 \frac{\text{N}}{\text{m}}$. Supponiamo che all'oscillatore sia applicata una forza sinusoidale:

$$x(t) = F_0 \sin(\bar{\omega}t), \quad F_0 = 5\text{N}. \quad (6.18)$$

Stampare $x(t)$ e la dinamica dell'oscillatore dall'istante 0 a 20 partendo da condizioni iniziali nulle per $\bar{\omega} = 2$, $\bar{\omega} = 4$ e $\bar{\omega} = 7$.

```

1 %%%%%%%%%%%
2 % es35.m %
3 %%%%%%%%%%%
4
5 close all
6 clear all
7 clc
8 m=0.1;
9 k=1.6;
10 lambda=0.1;
11 y0=[0; 0];

```

```
12 T=0.01; % vettore dei tempi
13 t=0:T:20;
14 A=[0 1; -k/m -lambda/m];% definizione matrici
15 B=[0;1/m];
16 C=[1 0];
17 D=0;
18 F0=5; % segnale di ingresso x
19 omg=2;
20 % omg=4;
21 % omg=7;
22 x=F0*sin(omg*t);
23 y=lsim(A,B,C,D,x,t,y0); % dinamica modello
24 figure
25 plot(t,x)
26 hold on
27 plot(t,y,'r')
28 legend('x','y')
29 grid on
```

Esercizio 26. Ripetere l'Esempio 35 supponendo che l'oscillatore smorzato all'istante $t = 0$ sia a 10m dalla posizione di riposo.