

Informazioni generali sul corso

↳ Organizzazione delle lezioni

1. Informazioni generali sul corso
2. Introduzione a *Matlab*[®]
3. Introduzione a *Simulink*[®]
4. Sistemi di Controllo Digitale (...)



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Sistemi di Controllo Digitale

1. Introduzione al controllo digitale (generalità sui sistemi di controllo digitale, dispositivi di interfaccia e anello di controllo digitale).
2. Introduzione agli strumenti matematici per l'analisi di sistemi a tempo discreto (equazioni alle differenze, esempi di controllo digitale e importanza del campionamento).
3. Strumenti matematici per l'analisi di sistemi a tempo discreto (\mathcal{Z} -Trasformate, funzioni di trasferimento, \mathcal{Z} -trasformate di segnali elementari, esempi di funzioni)
4. \mathcal{Z} -Trasformate (proprietà, teoremi e metodi di antitrasformazione).
5. Campionamento e ricostruzione di segnali (dispositivi di campionamento, campionario, ricostruttore, Teorema di Shannon, aliasing, esempi).
6. Sistemi a tempo discreto (funzione di trasferimento discreta, composizione di schemi a blocchi).
7. Stabilità dei sistemi discreti (definizione, piano complesso, luogo delle radici, piano w).
8. Specifiche di progetto di sistemi di controllo e progetto di regolatori (tecniche di discretizzazione, progetto con metodi analitici, progetto mediante luogo delle radici, progetto nel piano w , regolatori standard PID).



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Laboratorio (Automatica)

Silvio Simani

Dipartimento di Ingegneria
Università di Ferrara
Tel. 0532 97 4844
Fax. 0532 97 4870

E-mail: ssimani@ing.unife.it

URL: <http://www.ing.unife.it/simani>



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Organizzazione delle Lezioni

- ↳ **Mercoledì 14:00 – 17:00, Laboratorio Informatica**
- ⇒ Esercitazioni *Guidate* al Calcolatore
- ↳ **Giovedì , 14:00 – 16:00**
- ↳ **Venerdì , 10:30 – 12:30**
- ⇒ Teoria e Presentazione delle Esercitazioni



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Introduzione a MATLAB

- ⇨ Linguaggio per risolvere problemi di calcolo numerico (*MAThematical LABoratory*)
- ⇨ Marchio registrato da *Math Works Inc. (U.S.A.)*
- ⇨ Pacchetto software più diffuso tra progettisti e ricercatori
- ⇨ Può essere ampliato da pacchetti specifici (*toolbox*)
 - ⇒ Es. Control System Toolbox, Identification Toolbox, Simulink
- ⇨ È un *interprete in grado di eseguire*
 - ⇒ Istruzioni native (*build-in*, definite dal Matlab)
 - ⇒ Contenuti in files (*m-files, function-files, script-files*)



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Elementi di base di Matlab

⇨ L'elemento di base è la matrice (elementi interi, reali o complessi)

⇨ $>> \mathbf{A} = [1,2,3,4,5,6]$

⇨ $\mathbf{A} = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{matrix}$

⇨ $>> 7*3+2$

⇨ $\mathbf{ans} =$

23



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Informazioni generali sul corso

- ⇨ Modalità d'esame
- ⇨ Esercizi assegnati ad ogni lezione
 - ⇒ Risoluzione in Laboratorio
- ⇨ Raccolta di esperienze da portare all'esame
 - ⇒ Testo esperienza
 - ⇒ Strumenti utilizzati
 - ⇒ Risultati ottenuti



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Bibliografia essenziale e strumenti didattici

1. Dispense e Lucidi del Corso di Automatica I (Laboratorio).
2. Matlab, The Language of Technical Computing. Getting Started with MATLAB. Version 5.2 (In formato pdf su CD Matlab)
3. MATLAB Primer. Second Edition. Kermit Sigmon. Department of Mathematics. University of Florida.
4. The MathWorks Inc., Matlab User's Guide, 1993.
5. The MathWorks Inc., Simulink User's Guide, 1995.
6. G. Marro, TFI: insegnare e apprendere i controlli automatici di base con matlab. Bologna: Zanichelli, I ed., Ottobre 1998.



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Operazioni sulle matrici

⇒ Date le matrici A e B di dimensioni opportune, si possono definire le seguenti operazioni:

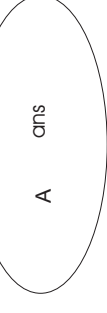
```
⇒ >> S = A + B
⇒ >> P = A * B
⇒ >> At = A'
⇒ >> Ai = inv(A)
⇒ >> Ap = pinv(A), con Ap = (AT * A)-1 * AT
```



Istruzioni elementari

```
⇒ >> who
⇒ >> Your variables are:
A ans
⇒ >> whos
```

Workspace



```
⇒ >> Name Size Bytes Class
A 2x3 48 double array
ans 1x1 8 double array
```

Grand total is 7 elements using 56 bytes



Calcolo dei valori

⇒ Definita la matrice rettangolare $A_{m \times n}$ e la matrice quadrata $B_{n \times n}$, Matlab definisce le seguenti funzioni:

```
⇒ >> det(B)
⇒ >> rank(A)
⇒ >> [V,D] = eig(B), con V*B*V' = D
⇒ >> Bi = inv(B)
⇒ >> [m,n] = size(A)
```

⇒ Selezione degli elementi della matrice A

```
⇒ A(i,j), A(1:2,2:3), A(1,:), A(:,3:5)
```



Istruzioni "DOS-like"

⇒ Direttore di lavoro: Z:\...\auto_?

```
⇒ >> dir
⇒ >> type
⇒ >> delete, ...
⇒ >> help <topic>
```

⇒ Creazione del file *pippo.mat* che contiene la matrice A

```
⇒ >> save pippo A (salva la matrice A nel file "pippo.mat")
⇒ >> clear A (rimuove dal workspace la matrice A)
⇒ >> load pippo (carica da file la matrice A)
```



Istruzioni di controllo

```

⇒ for
⇒ input
⇒ disp
⇒ while
⇒ end
⇒ if
⇒ ...

```



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Risoluzione di sistemi lineari

```

⇒ Calcolare il valore di x, con  $Ax = B$ 
⇒  $x = A^{-1}B$ 
⇒  $>> x = A \setminus B$ 
⇒  $>> x = \text{inv}(A) * B$ 
⇒ Calcolare il valore di x, con  $xC = D$ 
⇒  $x = DC^{-1}$ 
⇒  $>> x = D / C$ 
⇒  $>> x = D * \text{inv}(C)$ 

```



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Numeri complessi e formato dell'output

```

⇒  $>> x = [1, 3, 7.5 + 4*i, 6.3]$ 

```

x =

```

1 3 7.5 + 4i 6.3

```

```

⇒ Formato dell'output

```

```

⇒ format short: 5 cifre
⇒ format long: 15 cifre
⇒ format exe: formato esadecimale
⇒ format long e: floating point format with 15 digits.

```



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Script files e function files

```

% File quad.m
function r = rank(A)
%RANK Matrix rank.
% RANK(A) provides an estimate
% of the number of linearly
% independent rows or columns
% of a matrix A.
s = svd(A);
r = sum(s > 0);

```



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Operazioni sui polinomi

Deconvoluzione (divisione) di polinomi

⇒ `>> [q,r] = deconv(a,b)`, con q, quoziente e r, resto

Sviluppo in fratti

⇒ `>> [r,p,k] = residue(n,d)`

⇒ con $n = s^2 + 6s + 7$ e $d = s^2 + 5s + 6$

⇒ Sviluppo in fratti: $\frac{s^2+6s+7}{s^2+5s+6} = \frac{2}{s+3} - \frac{1}{s+2} + 1$

r = p = k =
 2 -3 1
 -1 -2



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Esercizi Proposti

Costruzione della funzione Matlab `my_hankel()`

- Scrivere la funzione $H = \text{my_hankel}(X, NrH, NcH, \text{shift})$, in cui X è un vettore di L elementi, NrH è il numero di righe di H, NcH è il numero di colonne di H, e shift è un intero maggiore od uguale a 0. La matrice H deve essere costruita in modo tale che

$$H = \begin{bmatrix} X(1 + \text{shift}) & \dots & X(\text{shift} + NcH) \\ \vdots & \ddots & \vdots \\ X(\text{shift} + NrH) & \dots & X(\text{shift} + NcH + NrH - 1) \end{bmatrix}$$

con l'ipotesi che $L \geq \text{shift} + NcH + NrH - 1$.



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Generazione di matrici e polinomi

Matrice casuale

⇒ `>> A = rand(n,m)`

⇒ Generazione di una matrice ad elementi casuali secondo alcuni parametri definibili dall'utente (distribuzione, valore medio, varianza, seme)

Rappresentazione di polinomi

⇒ $p(x) = x^3 - 6x + 3$

⇒ `>> p = [1, 0, -6, 3]`



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Operazioni sui polinomi

Radici di un polinomio

⇒ `>> r = roots(p)`

r =
 -2.6691
 2.1451
 0.5240

Prodotto c di due polinomi (coefficienti a e b)

⇒ `>> c = conv(a,b)`



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Risoluzione degli esercizi proposti

⇒ **Costruzione della matrice Q**

⇒ **Calcolare** $Q = [C^T, A^T * C^T, \dots, A^{T^{n-1}} * C^T]^T =$
 $= [C; C * A; \dots; C * A^{n-1}]$

```
>>Q = [C]; % calcolo passo passo
```

```
>>Q = [Q; C*A];
```

```
>>Q = [Q; C * A^2];
```

```
.
```

```
.
```

```
.
```

```
>>Q = [Q; C * A^(n-1)];
```



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Risoluzione degli esercizi proposti

⇒ **Costruzione della matrice Q in maniera automatica**

⇒ **Utilizzo della funzione** `obsv(A,C)`

```
function Q = obsv(A,C)
```

```
n = size(A,1);
```

```
Q = C;
```

```
for i=1:n-1
```

```
    Q = [C; Q*A];
```

```
end
```



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Esercizi Proposti

⇒ **Costruzione di una funzione Matlab** `obsv()`

- Scrivere un programma che, date le matrici $A_{n \times n}$ e $C_{m \times n}$, costruisca la matrice

$$Q = [C^T, A^T * C^T, \dots, A^{T^{n-1}} * C^T]^T.$$

Successivamente effettuare il test del rango.



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Esercizi Proposti

⇒ **Progetto di una trasformazione di matrici in ambiente Matlab**

- Data la terna $(A_{n \times n}, B_{n \times 1}, C_{1 \times n})$, calcolare la matrice $P = [B, A * B, \dots, A^{n-1} * B]$. Successivamente calcolare le matrici $T1 = \text{im}(P)$ e $T2$, con $T2$ tale che $T = [T1, T2]$ sia quadrata e invertibile. Si esegua la trasformazione $Ac = \text{inv}(T) * A * T$, $Bc = \text{inv}(T) * B$ e $Cc = C * T$. Infine, detto n_c il numero di colonne di $T1$, estrarre le matrici $Ac1$, avente le prime n_c righe e colonne di Ac , $Bc1$ dalle prime n_c righe di Bc e $Cc1$, le prime n_c colonne di Cc . In maniera analoga, calcolare la matrice $Q = [C; (C * A); \dots; C * A^{n-1}]$ e effettuare la trasformazione T ricavata, come in precedenza, dall'immagine di Q' e dal suo complemento ortogonale.



Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Risoluzione degli esercizi proposti

⇒ **Trasformazione delle matrici** $(A_{(n \times n)}, B_{(n \times 1)}, C_{(1 \times n)})$

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 2 & -2 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & -1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, C = [0 \ 0 \ 0 \ 0 \ 1] .$$



Risoluzione degli esercizi proposti

⇒ **Richiami di geometria**

⇒ $H = \text{orth}(K)$, ove H è una base ortonormale per l'immagine di K

⇒ $Z = \text{null}(V)$ ove Z è una base ortonormale per lo spazio nullo di V .

⇒ **Data una matrice reale** H , $\text{null}(H') = (\text{im}(H))^\perp$

⇒ $\mathcal{R}^n = \text{null}(K') + \text{im}(K)$

⇒ **Data** K , se $T1 = \text{orth}(K)$ e $T2 = \text{null}(K')$, allora $T = [T1 \ T2]$ è una base ortonormale per K



Risoluzione degli esercizi proposti

⇒ **Trasformazione delle matrici** $(A_{(n \times n)}, B_{(n \times 1)}, C_{(1 \times n)})$

⇒ **Calcolare la matrice** $Q = \text{obsv}(A, C)$

⇒ Controllare il rango di Q con $\text{rank}(Q)$

⇒ **La matrice** T di trasformazione risulta $T = [T1 \ T2]$

⇒ $T1 = \text{orth}(Q')$, $T2 = \text{null}(T1')$, $[n, nc] = \text{size}(T1)$

⇒ $T = \text{orth}(Q', \text{eye}(n))$

⇒ $A1 = \text{inv}(T) * A * T$, $B1 = \text{inv}(T) * B$, $C1 = C * T$

⇒ $Ao = A1(1:nc, 1:nc)$, $Bo = B1(1:nc)$, $Co = C1(1:nc)$

