

Laboratorio I (Automatica)

Silvio Simani

Dipartimento di Ingegneria
Università di Ferrara
Tel. 0532 293844
Fax. 0532 768602

E-mail: ssimani@ing.unife.it

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



Organizzazione delle Lezioni

Lunedì 8:30–11:30, Aula 9



Martedì 11:30–13:30, Aula 9



⇒ Teoria del Controllo Digitale

Giovedì e Venerdì 16:00–18:30, Aula 9 + Laboratorio di Informatica



⇒ Illustrazione degli Esercizi ed Esercitazioni (Guidate) al Calcolatore



Informazioni generali sul corso: Teoria



Sistemi di Controllo Digitale: Organizzazione delle lezioni

Parte Teorica

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 (Z -Trasformate, funzioni di trasferimento, Z -trasformate di segnali elementari, esempi di funzioni).
4. Z -Trasformate (proprietà, teoremi e metodi di antistrasformazione).
5. Campionamento e ricostruzione di segnali (dispositivi di campionamento, campionatore, 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).

Parte Applicativa

- Strumenti Software di Progettazione Assistita: Matlab[®], Simulink[®] e TFI (Transfer Function Interpreter).



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

Silvio Simani



Modalità d'esame



1: Raccolta di esperienze da portare all'esame

⇒ Testo esperienza, Strumenti utilizzati, Risultati ottenuti

⇒ Esercizi assegnati ad ogni lezione ⇒ Risoluzione in Laboratorio o a Casa



2: Prova Pratica al Calcolatore

⇒ Svolgimento di Esercizio individuale (simile a quelli assegnati a lezione)

Informazioni generali sul corso



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

Silvio Simani

Bibliografia e Strumenti Didattici

1. K.S igmaon, "MATLAB Primer". University of Florida,Florida,Second Edition ed.,1992. (Si scarica dalla rete).
2. The MathWorks, Inc., "Matlab, The Language of Technical Computing. Getting Started with MATLAB", Version 5.1 ed., May 1997. (In formato pdf su CD Matlab).
3. The MathWorks Inc., "Matlab User s Guide", 1993.
4. G.F. Franklin, J.D. Powell, and M. Workman, "Digital Control of Dynamic Systems". Addison-Wesley, Third Edition ed., 1998.
5. The MathWorks Inc., "Simulink User s Guide", 1995.
6. C. Bonivento, C. Melchiorri, and R.Zanasi, "Sistemi di Controllo Digitale". Bologna. Progetto Leonardo, Esculapio Ed., Marzo 1995.
7. M. Tibaldi, "Note introduttive a Matlab e Control System Toolbox". Bologna. Progetto Leonardo, 1993.
8. G.Marro, "TFI:insegnare ed apprendere i controlli automatici di base con Matlab". Bologna. Zanichelli ed., Ottobre 1998.



Introduzione a MATLAB

↩ **Linguaggio per risolvere problemi di calcolo numerico (MAThematical Laboratory)**

↩ **Marchio registrato da MathWorks 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*)**

⇒ **Contenuti in files (*m-files*)**



Elementi di base di Matlab

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

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

A =

1 2 3
4 5 6

>> 7*3+2

ans =

23



Istruzioni elementari

>> who

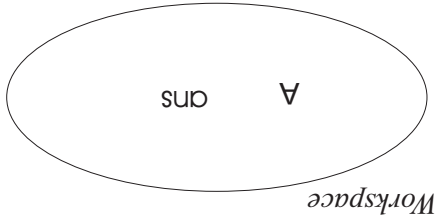
Your variables are:
A
ans

>> whos

Name Size Bytes Class

A 2x3 48 double array
ans 1x1 8 double array

Grand total is 7 elements using 56 bytes



Istruzioni "DOS-like"



Direttorio di lavoro: Z:\...\auto-?



>> **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)



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 $A^p = (A^T * A)^{-1} * A^T$



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 $V \cdot B \cdot V^T = D$

⇒ `>> B_i = 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)`



Risoluzione di sistemi lineari

Calcolare il valore di x, con $Ax = B$

⇒ `x = A\B`

⇒ `>> x = A \ B`

⇒ `>> x = inv(A) * B`

Calcolare il valore di x, con $xC = D$

⇒ `x = DC^-1`

⇒ `>> x = D / C`

⇒ `>> x = D * inv(C)`



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.
A = B * B;
B = A;
s = svd(A);
r = sum(s > 0);
```

Istruzioni di controllo

for
input
disp
while
end
if
...

Numeri complessi e formato dell'output

⇒ `>> x = [1, 3, 7.5 + 4*1i, 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.`



Generazione di matrici e polinomi

Matrice casuale

⇒ `>> A = rand(n,m)`

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

Rappresentazione di polinomi

⇒ $p(x) = x^3 - 6x + 3$

⇒ `p = [1, 0, -6, 3]`





Radici di un polinomio

$\Rightarrow \gg r = \text{roots}(p)$

$r =$

-2.6691
2.1451
0.5240



Prodotto c di due polinomi (coefficienti a e b)

$\Rightarrow \gg c = \text{conv}(a,b)$



Deconvoluzione (divisione) di polinomi

$\Rightarrow \gg [q,r] = \text{deconv}(a,b)$, con q, quoziente e r, resto



Sviluppo in fratti

$\Rightarrow \gg [r,p,k] = \text{residue}(n,d)$

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

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

$r =$
 $p =$
 $k =$

2
-3
1

-1
-2

Operazioni sui polinomi



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.

Esercizi Proposti



Costruzione della funzione Matlab *my-hankel()*

- Scrivere la funzione $H = \text{my-hankel}(X, N_rH, N_cH, \text{shift})$, in cui X è un vettore di L elementi, N_rH è il numero di righe di H , N_cH 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 con l'ipotesi che $L \geq \text{shift} + N_cH + N_rH - 1$,
$$H = \begin{bmatrix} X(1 + \text{shift}) & \dots & X(\text{shift} + N_rH) \\ \vdots & \ddots & \vdots \\ X(\text{shift} + N_cH) & \dots & X(\text{shift} + N_cH + N_rH - 1) \end{bmatrix}$$

Esercizi Proposti

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.



Risoluzione degli esercizi proposti

Costruzione della matrice Q

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

$>> Q = [C]; \% \text{ calcolo passo passo}$

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

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

$>> Q = [Q; C * A^{(n-1)}];$



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
```



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 , $\text{null}(H') = (\text{im}(H))^\perp$

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

Data K , se $T_1 = \text{orth}(K)$ e $T_2 = \text{null}(K')$, allora $T = [T_1 \ T_2]$ 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 $T = [T1 \ T2]$

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

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

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

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



Risoluzione degli esercizi proposti

Trasformazione delle matrici ($A^{(n \times n)}, B^{(n \times 1)}, C^{(1 \times n)}$)

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

