Laboratorio (Automatica)

Sergio Beghelli
Silvio Simani

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

E-mail: {sbeghelli,ssimani}@ing.unife.it

URL: http://www.ing.unife.it/~simani

Organizzazione delle Lezioni

- **Mercoledì 16:00–18:00, Aula 1**

  ⇒ Richiami di Teoria e Presentazione delle Esercitazioni

- **Venerdì 14:00–18:30, Laboratorio DU Meccanica**

  ⇒ Esercitazioni Guidate al Calcolatore

Informazioni generali sul corso

Organizzazione delle lezioni

1. Informazioni generali sul corso
2. Introduzione a Matlab
3. Simulazione di Sistemi Dinamici
4. Introduzione a Simulink
5. Osservatori e retroazione uscita-stato-ingresso
6. Modelli approssimati di sistemi dinamici
7. Progetto di Reti Correttrici
8. Identificazione di Sistemi Dinamici
9. Sintonizzazione di Controllori PID
10. Analisi di Sistemi a Dati Campionati

Informazioni generali sul corso

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

Bibliografia e strumenti didattici

1. Dispense del Corso di Laboratorio di Automatica. Sergio Beghelli, Cesare Fantuzzi, Silvio Simani. (Fotocopisteria, tutorato, www)
2. Matlab, The Language of Technical Computing. Getting Started with MATLAB. *Version 5.1* (In formato pdf su CD Matlab)
3. MATLAB Primer. Second Edition. Kermit Sigmon. Department of Methematics. University of Florida.
4. The MathWorks Inc., Matlab User's Guide, 1993.
5. L. F. Shampine and M. W. Reichel, "The Matlab Ode Suite", Tech. rep., The MathWorks, Inc, 1997. (Disponibile anche come file in formato pdf).
6. The MathWorks Inc., Simulink User's Guide, 1995.
7. B. C. Kuo, Automatic Control Systems. Englewood Cliffs, New Jersey: Prentice Hall, 7th ed., 1995.
8. P. Bolzern, R. Scattolini, and N. Schiavoni, Fondamenti di controlli automatici. Milano: McGraw- Hill, I ed., Marzo 1998.
9. G. Marro, TFI: insegnare e apprendere i controlli automatici di base con matlab. Bologna: Zanichelli, I ed., Ottobre 1998.
10. C. Fantuzzi, Controllori Standard PID. Versione 1.2, Appunti del Corso, 1a ed., Maggio 1997.

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 =**

$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{matrix}$$

- >> **7*3+2**
- **ans =**

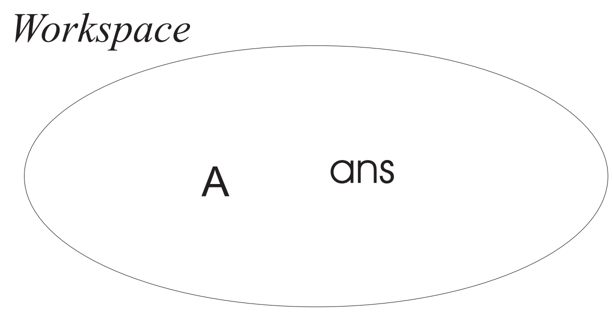
$$23$$

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_?**
  - ⇒ 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)*

Operazioni sulle matrici

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

$\Rightarrow$ `>> S = A + B`

$\Rightarrow$ `>> P = A * B`

$\Rightarrow$ `>> At = A'`

$\Rightarrow$ `>> Ai = inv(A)`

$\Rightarrow$ `>> Ap = pinv(A)`, con $Ap = \left(A^T * A\right)^{-1} * A^T$

Calcolo dei valori

- **Definita la matrice rettangolare** $\mathtt{A}_{m\times n}$ **e la matrice quadrata** $\mathtt{B}_{n\times n}$**, 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)`

Risoluzione di sistemi lineari

- **Calcolare il valore di $\mathbf{x}$, con $\mathrm{A}\mathbf{x} = \mathrm{B}$**

  $\Rightarrow$ $\mathbf{x} = \mathrm{A}^{-1}\mathrm{B}$

  $\Rightarrow$ `>> x = A \ B`

  $\Rightarrow$ `>> x = inv(A) * B`

- **Calcolare il valore di $\mathbf{x}$, con $\mathbf{x}\mathrm{C} = \mathrm{D}$**

  $\Rightarrow$ $\mathbf{x} = \mathrm{D}\mathrm{C}^{-1}$

  $\Rightarrow$ `>> x = D / C`

  $\Rightarrow$ `>> x = D * inv(C)`

Script files e function files

```
% File quad.m



A = B * B;

B = A;
```

```
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);
```

Istruzioni di controllo

- `for`
- `input`
- `disp`
- `while`
- `end`
- `if`
- `...`

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.

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]`

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)`

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 $\mathtt{n} = s^2 + 6s + 7$ e $\mathtt{d} = s^2 + 5s + 6$

  ⇒ 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
```

Esercizi Proposti

Costruzione della funzione Matlab *my_hankel()*

- Scrivere la funzione $H =$ *my_hankel(X,NrH,NcH,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

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

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

Esercizi Proposti

Costruzione di una funzione Matlab *obsv()*

- Scrivere un programma che, date le matrici $\mathrm{A}_{n\times n}$ e $\mathrm{C}_{m\times n}$, costruisca la matrice

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

Successivamente effettuare il test del rango.

Esercizi Proposti

Progetto di una trasformazione di matrici in ambiente Matlab

- Data la terna $(\mathrm{A}_{n\times n}, \mathrm{B}_{n\times 1}, \mathrm{C}_{1\times n})$, calcolare la matrice $P = [B,\ A * B,\ \ldots\ ,\ A^{n-1} * B]$. Successivamente calcolare le matrici `T1 = im(P)` e `T2`, con `T2` tale che `T = [T1, T2]` sia quadrata e invertibile. Si esegua la trasformazione `Ac = inv(T) * A * T`, `Bc = 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 $Q = [C;\ (C * A);\ \ldots\ ;\ C * A^{n-1}]$ 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,\ \ldots,\ {A^T}^{n-1} * C^T]^T = $
  $= [C;\ C * A;\ \ldots;\ C * A^{n-1}]$

```
>>Q = [C]; % 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 = orth(K)`, **ove** `H` **è una base ortonormale per l'immagine di** `K`
- `Z = null(V)` **ove** `Z` **è una base ortonormale per lo spazio nullo di** `V`**.**
- **Data una matrice reale** `H`**,** `null(H') =` $(im(H))^{\perp}$
- $\Re^n = \mathbf{null}(K') + \mathbf{im}(K)$
- **Data** `K`**, se** `T1 = orth(K)` **e** `T2 = 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 = obsv(A,C)`
  - ⇒ Controllare il rango di `Q` con `rank(Q)`
- **La matrice `T` di trasformazione risulta** `T=[T1 T2]`
  - ⇒ `T1=orth(Q')`, `T2=null(T1')`, [n,nc]=size(T1)
  - ⇒ `T = orth(Q',eye(n))`
- `A1 = inv(T) * A * T`, `B1 = 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** $(A_{(n\times n)}, B_{(n\times 1)}, C_{(1\times n)})$

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