

Trusa de instrumente IDENT din MATLAB

Ce este o trusă de instrumente din MATLAB?

O bibliotecă de rutine specializate pe un anumit domeniu de cercetare.

În particular, **IDENT** se referă la domeniul **IS** și conține rutine implementate în tehnologia **Programării Orientate pe Obiecte**.

2 p

Mai mult

Există și o interfață grafică prietenoasă cu utilizatorul (**GUI**), pentru efectuarea de experimente de identificare comparative.

Obiecte IDENT principale

IDDATA

IDMODEL

IDSS

Cîteva funcții IDENT frecvent utilizate

armax/arx

ident

n4sid

bj/oe

idarx

pem

compare

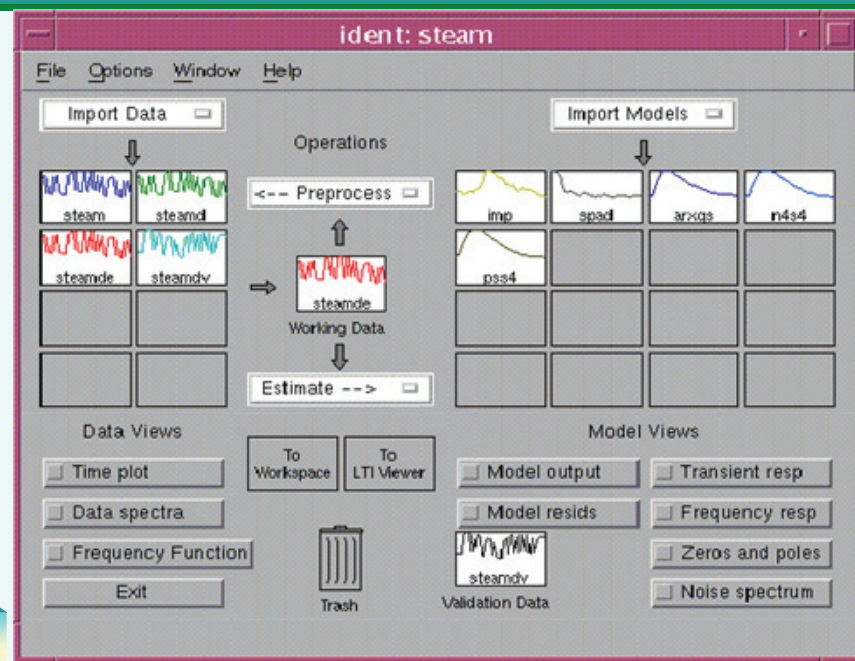
idpoly

predict

iddata

iv4

sim



• Lansată în execuție prin comanda:

`>> ident`

Trusa de instrumente IDENT din MATLAB

IDDATA

- D** Structură de date intrare-ieșire generate folosind metoda constructor `iddata` asociată obiectului `IDDATA` (date de identificare). Cele 2 matrici de date `u` (de intrare) și `y` (de ieșire) pot avea dimensiuni identice: `N` linii și `nr` coloane. Fiecare din cele `nr` perechi de coloane de date I/O este văzută ca un experiment econometric (sau o realizare). Dacă `y` are numai o coloană, atunci `nr=nu` și datele provin de la un model MISO. Datele se regăsesc în câmpurile `D.u` și `D.y`. Structura se compune din următoarele câmpuri:

```

Domain: 'Time'/'Frequency'
Name: 'String'
OutputData: [1x37 char]
    y: 'Same as OutputData' ← date de ieșire
OutputName: 'Ny-by-1 cell array of strings'
OutputUnit: 'Ny-by-1 cell array of strings'
InputData: [1x36 char]
    u: 'Same as InputData' ← date de intrare
InputName: 'Nu-by-1 cell array of strings'
InputUnit: 'Nu-by-1 cell array of strings'
    Period: [1x51 char] ← perioadă de eșantionare
InterSample: [1x36 char]
    Ts: [1x54 char]
Tstart: 'Scalar (Starting time)'
SamplingInstants: [1x51 char]
TimeUnit: 'String'
ExperimentName: [1x43 char]
Notes: 'Cell array of strings'
UserData: 'Arbitrary'

```

☞ Există blocuri de date în domeniul timpului sau în domeniul frecvenței.

➔ Pentru mai multe detalii:

```
>> idprops iddata
```

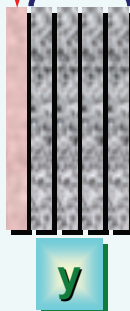
Trusa de instrumente IDENT din MATLAB

Cum se poate crea un bloc de date?

Este necesar ca, mai întâi, să se cunoască modul de reprezentare a datelor primare în mediul de programare MATLAB.

Date pe canale de măsură (fiecare coloană este asociată câte unui canal)

👉 Momente de eșantionare.
(eventual)



- Datele primare sunt reprezentate în diferite formate, dintre care MATLAB poate recunoaște următoarele, uzuale:

➔ **ASCII (.M)** (similare programelor principale)

➔ **ASCII (.TXT)** (cu datele precizate pe coloane)

➔ **Tabel (.XLS)** (accesibile cu MS Office - Excell)

➔ **Criptat (.MAT)** (create și salvate pe disc prin comanda **MATLAB save**)

- Pentru încărcarea lor în spațiul de lucru al mediului de programare MATLAB, se pot utiliza comenzile următoare:

```
>> nume_fisier_date ➔ ASCII (.M)
```

```
>> load nume_fisier_date.extensie ➔ ASCII (.TXT), Criptat (.MAT)
```

```
>> xlsread nume_fisier_date.xls ➔ Tabel (.XLS)
```

- Datele sunt memorate în variabile identificate prin numele fișierelor de proveniență sau, dacă este vorba despre un fișier **.MAT**, prin numele atribuite lor înainte de salvarea în acel fișier.

Trusa de instrumente IDENT din MATLAB

Exemple de fișiere primare de date

→ ASCII (.M)

Y11.M - Notepad

File Edit Format View Help

```
%
%
% Data file Y11.M
%
% Collective conscience of humankind on Earth.
% Electro-magnetic measurements performed at Kings College London (UK),
% starting with Jul 2000, until Sep 2004.
%
%
% y = [0.629883 0.629883 0.620117 0.615234 0.620117 0.620117 0.620117 0.615234 ...
%       0.625000 0.615234 0.615234 0.610352 0.610352 0.600586 0.610352 0.615234 ...
%       0.610352 0.605469 0.610352 0.610352 0.605469 0.610352 0.615234 0.610352 ...
%       0.610352 0.595703 0.595703 0.595703 0.585938 0.590820 0.590820 0.595703 ...
%       0.600586 0.600586 0.590820 0.654297 0.737305 0.825195 0.634766 0.600586] ;
%
%                               ^ September 11, 2001 here.
%
% y = [y ...
%       0.581055 0.585938 0.590820 0.595703 0.600586 0.600586 0.595703 0.600586 ...
%       0.605469 0.600586 0.600586 0.605469 0.605469 0.610352 0.605469 0.600586 ...
%       0.595703 0.595703 0.590820 0.590820 0.600586 0.600586 0.600586 0.595703 ...
%       0.600586 0.595703 0.585938 0.585938 0.576172 0.566406 0.571289 0.595703 ...
%       0.581055 0.576172 0.590820 0.576172 0.595703 0.600586 0.585938 0.571289 ...
%       0.576172 0.576172 0.585938 0.581055 0.590820 0.585938 0.576172 0.576172 ...
%       0.566406 0.571289 0.561523 0.556641 0.546875 0.551758 0.551758 0.546875 ...
%       0.546875 0.551758 0.546875 0.541992 0.541992 0.556641 0.556641 0.556641 ...
%       0.551758 0.541992 0.541992 0.551758 0.551758 0.546875 0.556641 0.706875] ;
%
%                               March 11, 2004 here. ^
%
% y = [y ...
%       0.546875 0.537109 0.527344 0.532227 0.527344 0.532227 0.541992 0.546875 ...
%       0.551758 0.556641 0.551758 0.551758 0.546875 0.546875 0.537109 0.532227] ;
%
% y = 1-y ;
%
% Ts = 1 ;
% unit = 'Time [12 days interval]' ;
% ntime = 0:(length(y)-1) ;
% label = 'Collective conscience of humankind on Earth (2000-2004).' ;
% yunit = 'Electromagnetic intensity [mH]' ;
%
%
% Author: Dan Stefanoiu
% Date: 15.11.2004
% Updated: 25.07.2007
```

Observați maniera de atribuire a variabilei care conține datele.

Pot fi prezente și alte informații (auxiliare) privind datele.

Exemple de fișiere primare de date (final)

→ ASCII (.TXT)

Date pe canale de măsură

→ Tabel (.xls)

👉 **Momente de esantionare.**

👉 Nu este obligatoriu ca prima coloană să includă momente de eșantionare.

Trusa de instrumente IDENT din MATLAB

Și totuși... Cum se poate crea un bloc de date?

Odată încărcate în spațiul de lucru MATLAB, datele primare pot forma blocuri de date cu ajutorul comenzii:

>> iddata

Exemplu

- A fost proiectată funcția **make_DATA**, care crează și salvează pe disc un bloc de date corespunzător unor serii de timp măsurate simultan pe mai mult canale:

```

make_DATA.m - Notepad
File Edit Format View Help

%%
File MAKE_DATA.M
Function: MAKE_DATA
Synopsis: DATA = make_DATA(y) ;

Creates the IDDATA object from acquired data y. The first column of
y should be the sampling instants vector. The remaining columns are then
understood as measured data on different channels. If y is a vector
(row or column), then it is understood as measured data on one channel.
Once DATA being created, it is saved on disk in the current directory,
by means of MATLAB command SAVE. The DATA object can be loaded in MATLAB
workspace by means of dual command LOAD.

The user is also invited to input some auxiliary data (if any),
such as:

DATA.Name          = name of data block; the same name is used for the
                    file saved on disk; (char string);
DATA.Notes         = what the data set means (char string);
DATA.ExperimentName = name of the measuring experiment (char string);
DATA.TimeUnit      = unit of time (e.g. seconds, minutes, hours, days, etc.);
                    (char string);
DATA.Tstart        = integer that encodes the starting date of measurements;
                    can be created by:
                    datenum('dd-mmm-yyyy HH:MM:SS') ;
                    to retrieve the date, call:
                    datestr(DATA.Tstart) ;
                    the user is invited to insert the string: 'dd-mmm-yyyy HH:MM:SS';
DATA.Ts            = sampling period (scalar);
DATA.OutputName    = name of each measuring channel (cell array of char strings);
DATA.OutputUnit    = unit of measured data (cell array of char strings).

Example:
DATA.Name = 'T_Bucharest' ;
DATA.Notes = 'Systematic measurements on 2 channels' ;
DATA.ExperimentName = {'Daily min and max average temperatures in Bucharest'} ;
DATA.TimeUnit = 'day' ;
DATA.Tstart = datenum('29-Nov-2007 12:00:00') ;
DATA.Ts = 1 ;
DATA.OutputName = {'Minimum temperature' ;
                   'Maximum temperature'} ;
DATA.OutputUnit = {'°C' ;
                   '°C'} ;

Note: The field DATA.Tstart cannot be set unless the sampling is uniform.
Non uniform sampling leaves the field empty.
Shall the starting date of measurements has to be specified,
the field DATA.UserData can be used. For example:
DATA.UserData = '29-Nov-2007' ;

Empty input enforces empty output.

Uses:
VECTORIZE
WAR_ERR

```


Trusa de instrumente IDENT din MATLAB

Exemplu

make_DATA

make_DATA.m - Notepad

File Edit Format View Help
function DATA = make_DATA(y)

```
%
%% BEGIN
%% Messages
% ~~~~~
warning('off','MATLAB:dispatcher:InexactMatch');
FN = 'MAKE_DATA';
E1 = [FN 'Missing or empty input data. Empty output. Exit.'];
BL = [blanks(3) 'Insert.'];
EN = (ENTER means none);
I1 = [BL 'the data name block [ENTER means ''DATA'']: '];
I2 = [BL 'data notes' EN];
I3 = [BL 'the experiment name' EN];
I4 = [BL 'the time unit' EN];
I5 = [BL 'the starting date in format <dd-mmm-yyyy HH:MM:SS> (ENTER means NOW): '];
I6 = [BL 'user info (such as the starting date) as a string: '];
I7 = [BL 'data name on channel %d' EN];
I8 = [BL 'unit on channel %d' EN];
S = [FN 'Data saved in file <%s.MAT>.'];

% Faults preventing
% ~~~~~
DATA = iddata; % aici se crează obiectul DATA vid, care va fi completat ulterior cu informație
if (isempty(y))
    war_err(E1);
    return;
end;
if (isempty(y))
    war_err(E1);
    return;
end;

% Building the DATA object
% ~~~~~
if (isscalar(y) || isvector(y))
    DATA.y = vectorize(y); % Storing the data ...
    DATA.Ts = 1; % the sampling period ...
else
    DATA.y = y(:,2:end); % and the sampling instants (if any).
    DATA.SamplingInstants = y(:,1);
end;
war_err(FN);
DATA.Name = input(I1,'s'); % Setting the name of data block.
if (isempty(DATA.Name))
    DATA.Name = 'DATA';
end;
DATA.Notes = input(I2,'s'); % Setting the notes on data (what they mean).
DATA.ExperimentName = {input(I3,'s')}; % Setting the name of experiment or supplementary information.
DATA.TimeUnit = input(I4,'s'); % Setting the time unit (e.g. ms, s, hours, days, etc.)
if (~isempty(DATA.Ts))
    DATA.Tstart = now; % Setting the starting date and/or time
    FN = input(I5,'s'); % (only allowed for uniform sampling).
    if (isempty(FN))
        DATA.Tstart = datenum(FN);
    end;
else
    DATA.Tstart = datenum(FN);
end;
DATA.UserData = input(I6,'s'); % Here the starting date can be specified as a string
% in a preferred format (such as 'dd-Mmm-yyyy').
EN = size(DATA.y,2);
FN = input(sprintf(I7,1),'s'); % Setting the name of each output channel.
BL = input(sprintf(I8,1),'s'); % Setting the unit of each output channel.
for (n=2:EN)
    FN = [FN ; {input(sprintf(I7,n),'s')}];
    BL = [BL ; {input(sprintf(I8,n),'s')}];
end;
DATA.OutputName = FN;
DATA.OutputUnit = BL;
DATA.Domain = 'Time'; % Data are in time domain.
Y = DATA;
eval(['save ', DATA.Name '.mat Y']); % Save the DATA object.
war_err(sprintf(S,DATA.Name));

% END
```

⚡ Dacă sunt mai mult de 2 coloane, se consideră că prima include momente de eșantionare.

Trusa de instrumente IDENT din MATLAB

IDMODEL

M Obiectul **IDMODEL** (model de identificare I/O) conține câmpurile:

polinoamele modelului \rightarrow (A,B,...,F)

```
a: 'A-polynomial (row vector)'
b: 'B-polynomial (row vector)'
c: 'C-polynomial (row vector)'
d: 'D-polynomial (row vector)'
f: 'F-polynomial (row vector)'
```

RIO

$$A(q^{-1})y[n] = \frac{B(q^{-1})}{F(q^{-1})}u[n] + \frac{C(q^{-1})}{D(q^{-1})}e[n]$$

$\forall n \in \mathbb{N}^*$

indicii structurali \rightarrow (na,nb,...,nf)

```
da: 'standard deviation of a (scalar)'
db: 'standard deviation of b (scalar)'
dc: 'standard deviation of c (scalar)'
dd: 'standard deviation of d (scalar)'
df: 'standard deviation of f (scalar)'
```

timpul mort \rightarrow

```
na: 'order of A-polynomial (scalar)'
nb: 'order of B-polynomial (scalar)'
nc: 'order of C-polynomial (scalar)'
nd: 'order of D-polynomial (scalar)'
nf: 'order of F-polynomial (scalar)'
nk: 'delay of B-polynomial (scalar)'
```

InitialState: [1x45 char]

Name: 'string'

perioada de eșantionare \rightarrow

Ts: 'sample time in seconds (scalar)'

InputName: 'Nu-by-1 cell array of strings'

InputUnit: 'Nu-by-1 cell array of strings'

OutputName: 'Ny-by-1 cell array of strings'

OutputUnit: 'Ny-by-1 cell array of strings'

TimeUnit: 'string'

ParameterVector: 'Np-by-1 vector'

PName: 'Np-by-1 cell array of strings'

CovarianceMatrix: 'Np-by-Np matrix'

NoiseVariance: 'Ny-by-Ny matrix'

InputDelay: 'Nu-by-1 vector'

Algorithm: [1x38 char]

EstimationInfo: [1x39 char]

Notes: 'Array or cell array of strings'

UserData: 'Arbitrary'

\rightarrow Pentru mai multe detalii:

>> idprops idmodel

Evident, polinoamele A și B se regăsesc în câmpurile: **M.a**, respectiv **M.b**. În **M.b** sunt salvați atât coeficienții nenuli cât și cei nuli (datorați întârzierii intrinseci) ai polinomului B. Ordinele polinoamelor sunt memorate în **M.na**, respectiv **M.nb**, iar întârzierea intrinsecă – în **M.nk**.

Trusa de instrumente IDENT din MATLAB

IDSS

 → Pentru mai multe detalii: `>> idprops idss`

S Obiectul IDSS (model de identificare pe stare) conține câmpurile:

matricile modelului →

(A,B,...,K)

starea inițială →

$$\begin{cases} x[n+1] \equiv A(\theta)x[n] + B(\theta)u[n] + K(\theta)e[n] \\ y[n] \equiv C(\theta)x[n] + D(\theta)u[n] + e[n] \end{cases}$$

$\forall n \in \mathbb{N}^*$

RSS

```

A: 'A-matrix (Nx-by-Nx matrix)'
B: 'B-matrix (Nx-by-Nu matrix)'
C: 'C-matrix (Ny-by-Nx matrix)'
D: 'D-matrix (Nu-by-Ny matrix)'
K: 'K-matrix (Nx-by-Ny matrix)'
X0: 'Initial states (Nx-by-1 matrix)'
dA: 'Std deviation of A-matrix'
dB: 'Std deviation of B-matrix'
dC: 'Std deviation of C-matrix'
dD: 'Std deviation of D-matrix'
dK: 'Std deviation of K-matrix'
dX0: 'Std deviation of X0-matrix'
SSParameterization: '['Free'|'Canonical'|'Structured']'
As: 'Structure matrix for A'
Bs: 'Structure matrix for B'
Cs: 'Structure matrix for C'
Ds: 'Structure matrix for D'
Ks: 'Structure matrix for K'
X0s: 'Structure matrix for X0'
StateName: 'Nx-by-1 cell array of strings'
InitialState: '['Auto'|'Backcast'|'Estimate'|
               'Zero'|'Fixed']'
nk: 'Model delays in response from u to y
     (nu-by-1 vector)'
DisturbanceModel: '['Estimate'|'None'|'Fixed']'
CanonicalIndices: 'Row vector or 'Auto''
Name: 'string'
Ts: 'Scalar (sample time in seconds)'
InputName: 'Nu-by-1 cell array of strings'
InputUnit: 'Nu-by-1 cell array of strings'
OutputName: 'Ny-by-1 cell array of strings'
OutputUnit: 'Ny-by-1 cell array of strings'
TimeUnit: 'string'
ParameterVector: 'Np-by-1 vector'
PName: 'Np-by-1 cell array of strings'
CovarianceMatrix: 'Np-by-Np matrix'
NoiseVariance: 'Ny-by-Ny matrix'
InputDelay: 'Nu-by-1 vector'
Algorithm: 'Structure: algorithm details'
EstimationInfo: 'Structure: estimation results'
Notes: 'Array or cell array of strings'
UserData: 'Arbitrary'
  
```

perioada de eșantionare →

Ev ident, matricile A, B, C, D, K se regăsesc în câmpurile: S.A, S.B, S.C, S.D, respectiv M.K. În S.X0 se află vectorul stărilor inițiale ale sistemului.

Trusa de instrumente IDENT din MATLAB

Ce legătură există între obiectele IDMODEL și IDSS?

Orice model **RSISO** poate fi convertit într-un model cu reprezentare pe stare (**MRS**) și reciproc, prin intermediul unor algoritmi specifici.

MIO

$$A(q^{-1})y[n] = \frac{B(q^{-1})}{F(q^{-1})}u[n] + \frac{C(q^{-1})}{D(q^{-1})}e[n] \quad \forall n \in \mathbb{N}^*$$

obiect IDSS

obiect IDMODEL

```
>> MRS = idss(MIO) ;
```

```
>> MIO = idpoly(MRS) ;
```

MRS

$$\begin{cases} \mathbf{x}[n+1] \equiv \mathbf{A}(\theta)\mathbf{x}[n] + \mathbf{B}(\theta)\mathbf{u}[n] + \mathbf{K}(\theta)\mathbf{e}[n] \\ \mathbf{y}[n] \equiv \mathbf{C}(\theta)\mathbf{x}[n] + \mathbf{D}(\theta)\mathbf{u}[n] + \mathbf{e}[n] \end{cases} \quad \forall n \in \mathbb{N}^*$$

Modelele pe stare în timp discret pot fi văzute și ca **modele în timp continuu**, prin precizarea perioadei de eșantionare (implicit unitară).

⚡ Nu poate fi controlat numărul de stări (care, de regulă, este ridicat).

```
>> MRS.Ts = Ts ;
```

① Trusa de instrumente IDENT din MATLAB

Cum se poate crea un obiect de tip model de identificare?

Există două posibilități uzuale:

① Direct, prin utilizarea uneia dintre comenzile:

```
>> idpoly
```

```
>> idarx
```

```
>> idss
```

```
>> idgrey
```

→ pentru modele I/O liniare polinomiale

⚡ A doua – numai pentru modele ARX.

→ pentru modele liniare pe stare

② Indirect, prin utilizarea uneia dintre rutinele de identificare menționate.

⚡ Se va reveni asupra acestui subiect.

• Apeluri întrebuintate frecvent:

```
>> M = idpoly ; → Crează modelul grosier (A=1, B=0, C=1, D=1, F=1) ...
```

```
>> M.a = A ; → ... apoi atribuie polinoamele.
```

```
>> M.b = B ;
```

```
>> M.c = C ;
```

```
>> M.d = D ;
```

```
>> M.f = F ;
```

$$\begin{cases} A = [1 & a_1 & a_2 & \dots & a_{na}] \\ B = [0 & \dots & 0 & b_1 & b_2 & \dots & b_{nb}] \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ F = [1 & f_1 & f_2 & \dots & f_{nf}] \end{cases}$$

$nk \geq 1$ (întârzierea intrinsecă)

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \dots + a_{na} q^{-na} \\ B(q^{-1}) &= q^{-nk} (b_1 q^{-1} + \dots + b_{nb} q^{-nb}) \\ C(q^{-1}) &= 1 + c_1 q^{-1} + \dots + c_{nc} q^{-nc} \\ D(q^{-1}) &= 1 + d_1 q^{-1} + \dots + d_{nd} q^{-nd} \\ F(q^{-1}) &= 1 + f_1 q^{-1} + \dots + f_{nf} q^{-nf} \end{aligned}$$

```
>> M = idpoly(A,B,C,D,F) ; → Crează direct modelul dorit.
```

⚡ Ultimele polinoame pot lipsi.

⚡ Unele polinoame pot fi vide.

Trusa de instrumente IDENT din MATLAB

Principalele rutine de identificare (1/9)

armax/arx

pem

bj/oe

iv4

levinson

n4sid

armax

- Identifică modele din clasa **ARMAX**, folosind date I/O măsurate.
- Apeluri tipice:

indici structurali

```
>> M = armax(DATA, [na nb nc nk]);
```

întârzierea intrinsecă (implicit 1)

```
>> M = armax(DATA, 'na', na, 'nb', nb, 'nc', nc, 'nk', nk);
```

obiect IDMODEL

obiect IDDATA

☞ Variantă care permite omiterea unora dintre indicii structurali.

Exemple

```
>> M = armax(DATA, 'na', 3, 'nb', 7) ; ➔ Identifică modelul ARX[3,7].
```

```
>> M = armax(DATA, 'na', 1, 'nc', 6) ; ➔ Identifică modelul ARMA[1,6].
```

- Apeluri alternative rapide:

```
>> M = armax(iddata(y,u),indici_structurali);
```

☞ Cu y și u de tip date preliminare (vectori/matrici de date așezate pe coloane).

```
>> M = armax(iddata(y,[]),indici_structurali);
```

☞ În cazul seriilor de timp.



Metoda de identificare

Minimizarea Erorii de Predicție (MMEP)

Trusa de instrumente IDENT din MATLAB

Principalele rutine de identificare (2/9)

armax

- Modelele identificate pot fi multi-dimensionale.

Dar nu de tip MIMO

Modelul **ARMAX-MIMO**

$$\begin{cases} \mathbf{A}(q^{-1})\mathbf{y}[n] = \mathbf{B}(q^{-1})\mathbf{u}[n] + \mathbf{C}(q^{-1})\mathbf{e}[n] \\ E\{\mathbf{e}[n]\mathbf{e}^T[m]\} = \Lambda\delta_0[n-m] \end{cases} \quad \forall n \in \mathbb{N}$$

Matrici de polinoame

$$\mathbf{A} \in \mathbb{R}^{ny \times ny}(q^{-1})$$

$$\mathbf{B} \in \mathbb{R}^{ny \times nu}(q^{-1})$$

$$\mathbf{C} \in \mathbb{R}^{ny \times ny}(q^{-1})$$

Semnale \rightarrow

$$\mathbf{y} \in \mathbb{R}^{ny}$$

$$\mathbf{u} \in \mathbb{R}^{nu}$$

$$\mathbf{e} \in \mathbb{R}^{ny}$$

aceeași dimensiune

Există tot atâtea perturbații câte canale de măsură.

Ecuatia generică **ARMAX-MIMO**

$$\sum_{j=1}^{ny} A_{l,j}(q^{-1})y_j[n] = \sum_{i=1}^{nu} B_{l,i}(q^{-1})u_i[n] + \sum_{j=1}^{ny} C_{l,j}(q^{-1})e_j[n] \quad \forall n \in \mathbb{N}$$

polinoame de grad

$$na$$

polinoame de grad

$$nb$$

polinoame de grad

$$nc$$

$$\forall l \in \overline{1, ny}$$

Avînd coeficienți necunoscuți.

Matrice de covarianță a zgomotelor

$$\Lambda \in \mathbb{R}^{ny \times ny}$$

Toate polinoamele unei matrici au același grad.

Trusa de instrumente IDENT din MATLAB

Principalele rutine de identificare (3/9)

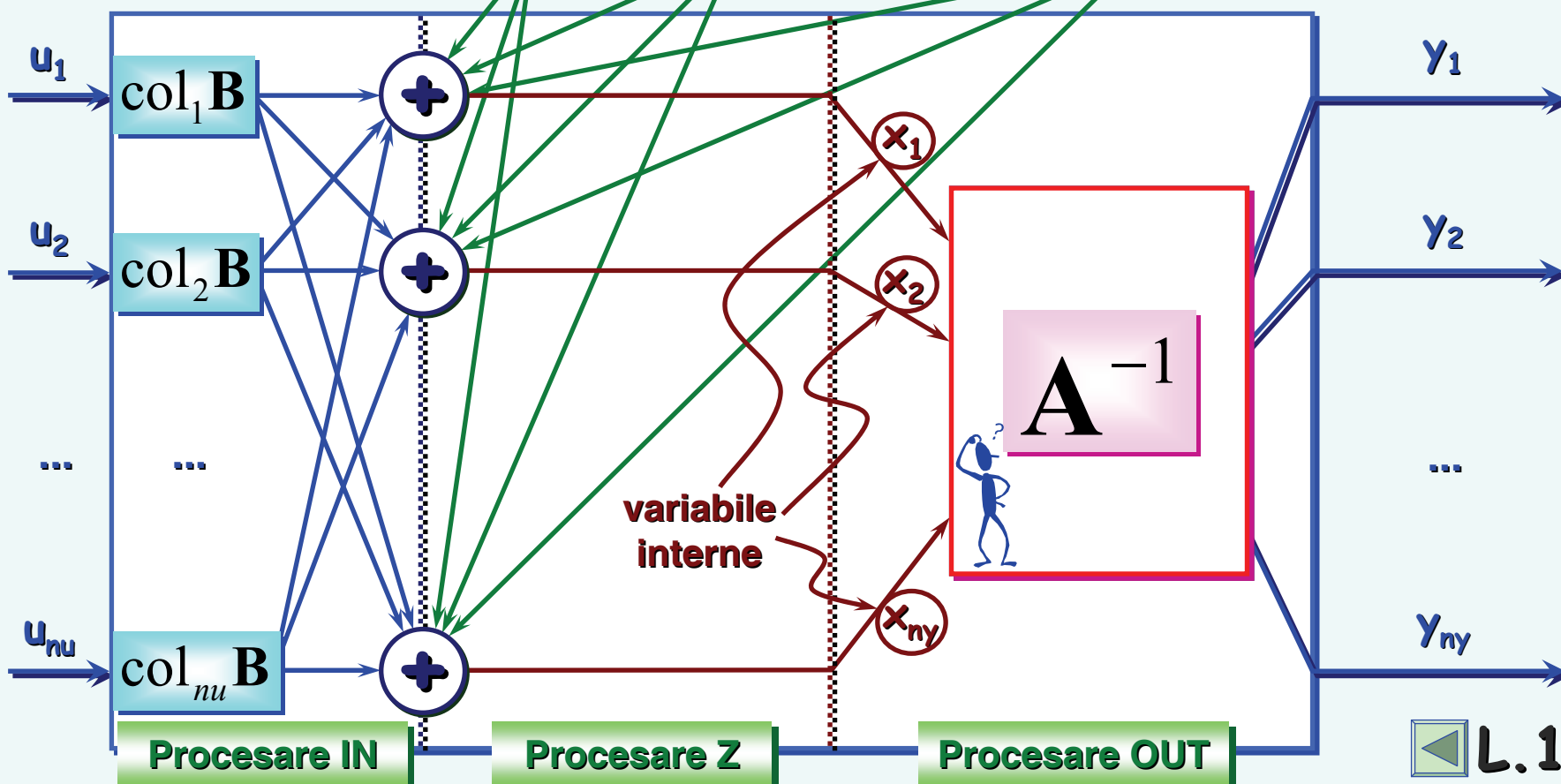
armax

Există 3 trepte
de procesare
a semnalelor

Reprezentare sistemică MIMO



Problema principală:
**evaluarea inversei
matricii A.**



Trusa de instrumente IDENT din MATLAB

Principalele rutine de identificare (4/9)

armax

Cum ar putea fi inversată matricea de polinoame A ?

În general, este **difficil**, dacă nu **imposibil**.

Este necesară o simplificare a modelului, care va permite fie **evitarea inversării**, fie **inversarea cu ușurință**.

Modelul va pierde însă din precizie.

Adoptarea unui model **AR-MO**

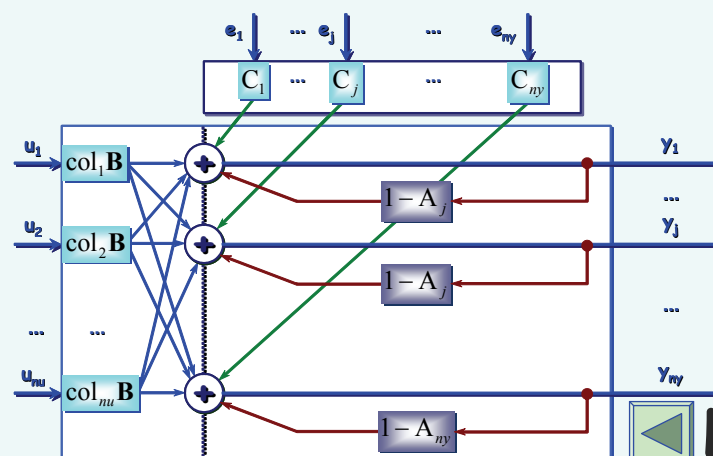
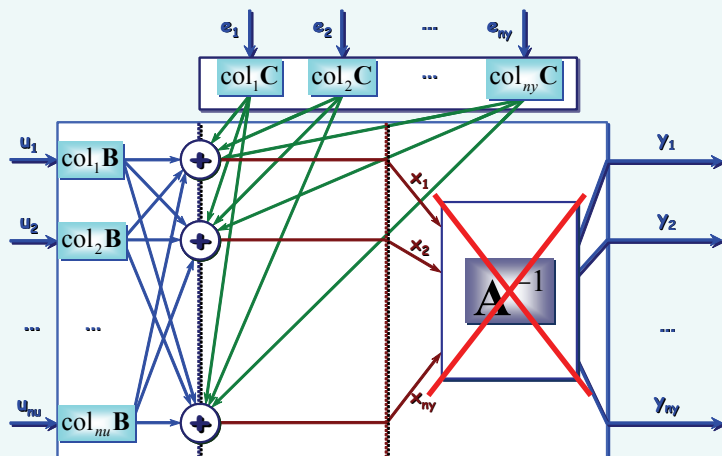
Dacă nu există date de intrare.

Ce principiu?

Adoptarea principiului din **MATLAB**

Modelul **MIMO** este văzut ca o colecție de n_y modele **MISO**.

Fiecare model **MISO** codifică influența pe care o au **toate** intrările asupra unei **anumite ieșiri**, corupte doar de zgomotul canalului său (nu și de zgomotele celorlalte canale).



Trusa de instrumente IDENT din MATLAB

armax

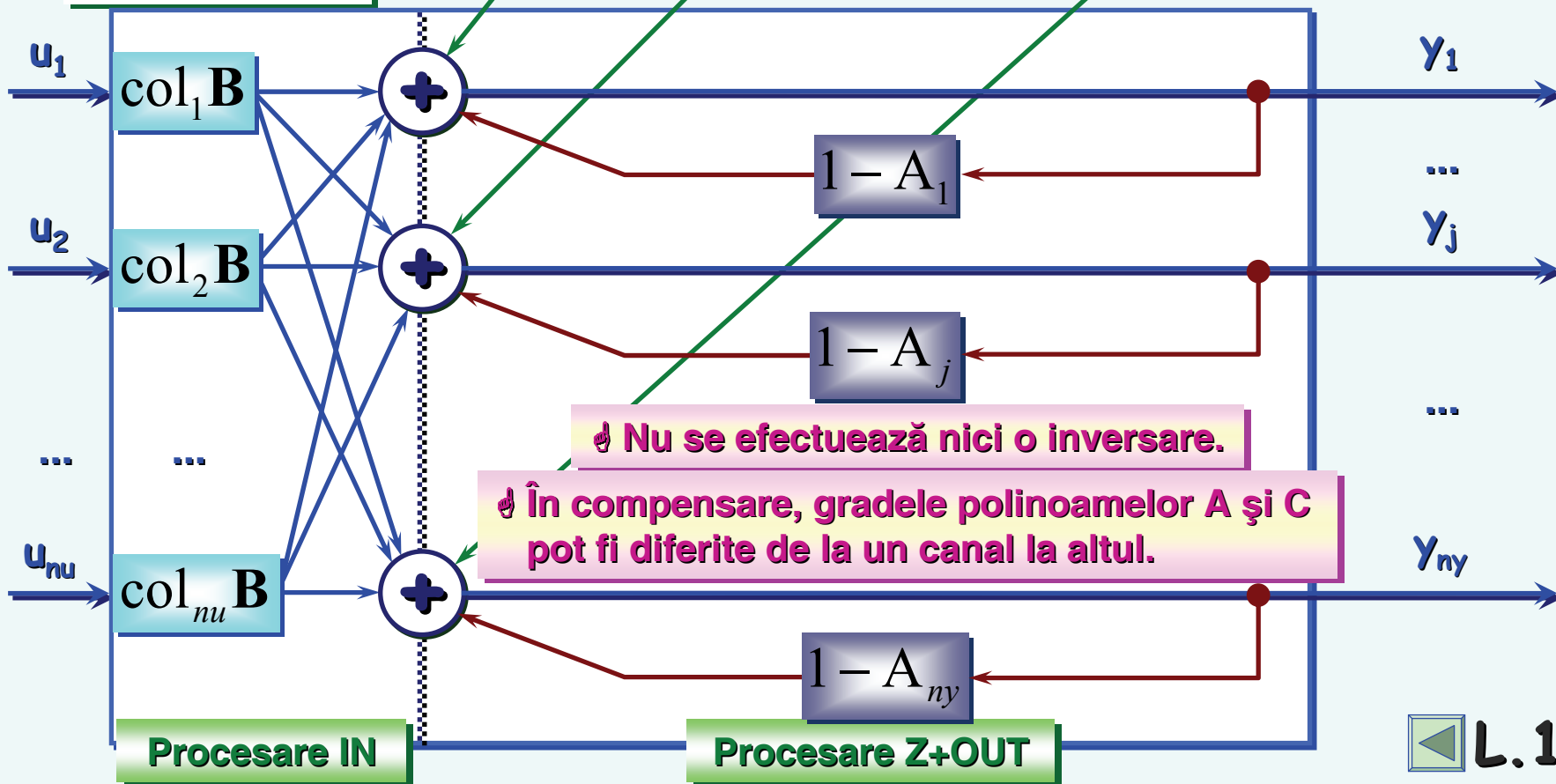
Principalele rutine de identificare (5/9)

Reprezentarea sistemică multi-MISO

 $e_1 \quad \dots \quad e_j \quad \dots \quad e_{ny}$ $C_1 \quad \dots \quad C_j \quad \dots \quad C_{ny}$

Acum, există doar 2 trepte.

Practic, matricile A și C sunt diagonale.



Nu se efectuează nici o inversare.

În compensare, gradele polinoamelor A și C pot fi diferite de la un canal la altul.

Trusa de instrumente IDENT din MATLAB

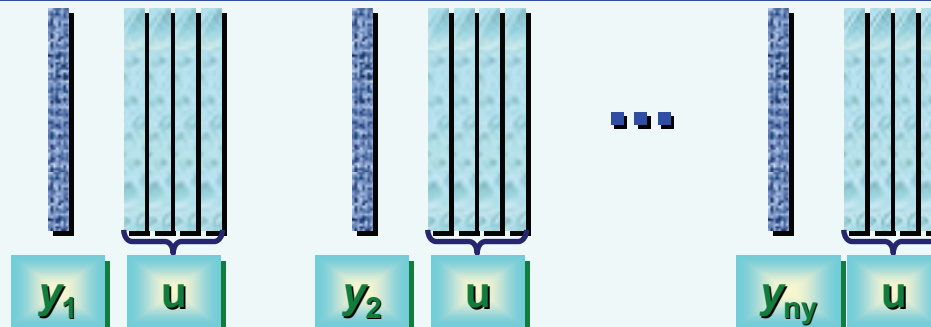
Principalele rutine de identificare (6/9)

armax

În concluzie

Modelul **ARMAX-MIMO**, este înlocuit de o **colecție de modele ARMAX-MISO**, identificate din blocuri de date de forma:

Se apelează **armax** de n_y ori.



- O funcție înrudită tentantă pentru Automatică este și: **arx** (pentru identificarea modelelor **ARX**)

Totuși

Funcția **arx** utilizează **Metoda Celor Mai Mici Pătrate (MCMMP)**, fiind mai puțin precisă decât **armax**.

pem

Prediction Error Minimization

- Funcție aflată la baza celorlalte rutine de identificare a modelelor I/O liniare.
- De regulă, ea **nu este utilizată direct**, ci **indirect**, prin intermediul celorlalte rutine (care o apelează în interiorul lor).
- Totuși, dacă se dorește, ea poate fi apelată ca în cazul funcției **armax**:

```
>> M = pem(DATA, ms);
```

structura modelului: obiect de tip **IDMODEL**, **IDSS** sau direct **indicii structurali**.

Trusa de instrumente IDENT din MATLAB

Principalele rutine de identificare (7/9)

bj

- Identifică modele din clasa **Box-Jenkins**, folosind date I/O măsurate.
- Apeluri tipice:

```
>> M = bj (DATA, [nb nc nd nf nk]);
```

```
>> M = bj (DATA, 'nb', nb, 'nc', nc, 'nd', nd, 'nf', nf, 'nk', nk);
```

obiect IDMODEL

obiect IDDATA

Variantă care permite omiterea unora dintre indicii structurali.

- Modelul **ARMAX** fiind un caz particular de model **Box-Jenkins**, el se poate identifica accidental și cu ajutorul funcției **bj**.

Funcția **bj** este însă mai lentă decât funcția **armax**.

- O altă funcție care identifică un model particular este și:
- Modelele **multi-MISO** se pot identifica de asemenea, folosind funcția **bj**.

Exemplu

```
>> M = bj (DATA, 'nb', [3 4], 'nc', 7, ...
           'nd', 2, 'nf', [5 6]);
```

→ Identifică modelul
BJ[(3,4),7,2,(5,6)]:

$$y[n] = \frac{B_1(q^{-1})}{F_1(q^{-1})} u_1[n] + \frac{B_2(q^{-1})}{F_2(q^{-1})} u_2[n] + \frac{C(q^{-1})}{D(q^{-1})} e[n] \quad \forall n \in \mathbb{N}$$

$$y[n] = \frac{B(q^{-1})}{F(q^{-1})} u[n] + e[n] \quad \forall n \in \mathbb{N}^*$$

$$e[n] = y[n] - \frac{B(q^{-1})}{F(q^{-1})} u[n] \quad \forall n \in \mathbb{N}^*$$

(pentru identificarea modelelor de tip
Eroare de Ieșire – Output Error)

Trusa de instrumente IDENT din MATLAB

Principalele rutine de identificare (8/9)

iv4

- Identifică modele din clasa **ARX**, prin: **Metoda Variabilelor Instrumentale (MVI)**

- Apeluri tipice:

```
>> M = iv4(DATA,[na nb nk]);
```

```
>> M = iv4(DATA,'na',na,'nb',nb,'nk',nk);
```

☞ Cu aceleași notații ca în cazul funcției **armax**.

levinson

- Identifică modele din clasa **AR**, folosind date I/O măsurate, pe baza:

$$A(q^{-1})y[n] = e[n] \quad \forall n \in \mathbb{N}^*$$

Algoritmului Levinson-Durbin (ALD)

- Apel tipic:

```
>> A = levinson(ry,na);
```

indice structural (implicit: **length(ry) - 1**)

☞ Care este și valoarea maximă posibilă.

vectorul
parametrilor

vectorul de auto-covarianță
a datelor

$$A = [1 \ a_1 \ a_2 \ \dots \ a_{na}]$$

- ALD** este una dintre cele mai eficiente proceduri de identificare, dar se referă la un model de complexitate redusă.

☞ Nu poate identifica modele AR-MO.

$$r_y[k] = \frac{1}{N-k} \sum_{n=k+1}^N y[n]y[n-k] \quad \forall k \in \overline{0, K}$$



$$K \geq na$$

... care se evaluează astfel:

```
>> [ry,k] = xcorr(y,K);
```

```
>> ry(k<0) = [];
```

Trusa de instrumente IDENT din MATLAB

Principalele rutine de identificare (9/9)

n4sid

- Identifică modele **cu reprezentare pe stare**, folosind date I/O măsurate, pe baza:

Metodei Subspațiilor de Stare (MSS)

$$\begin{cases} \mathbf{x}[n+1] \equiv \mathbf{A}(\theta)\mathbf{x}[n] + \mathbf{B}(\theta)\mathbf{u}[n] + \mathbf{F}(\theta)\mathbf{e}[n] \\ \mathbf{y}[n] \equiv \mathbf{C}(\theta)\mathbf{x}[n] + \mathbf{D}(\theta)\mathbf{u}[n] + \mathbf{e}[n] \\ E\{\mathbf{e}[n]\mathbf{e}^T[m]\} = \Lambda(\theta)\delta_0[n-m] \end{cases} \quad \forall n, m \in \mathbb{N}$$

Matrici de parametri

$$\mathbf{A} \in \mathbb{R}^{n_x \times n_x} \quad \mathbf{B} \in \mathbb{R}^{n_x \times n_u} \quad \mathbf{F} \in \mathbb{R}^{n_x \times n_y}$$

$$\mathbf{C} \in \mathbb{R}^{n_y \times n_x} \quad \mathbf{D} \in \mathbb{R}^{n_y \times n_u}$$

Matrice de covarianță a zgomotelor (diagonală)

$$\Lambda \in \mathbb{R}^{n_y \times n_y}$$

- Apel tipic:

```
>> S = n4sid(DATA, nx);
```

obiect IDSS

obiect IDDATA

numărul dorit de stări

```
>> F = S.K;
```

Exemplu

```
>> S = n4sid(DATA, 'best', 'InitialState', 'Estimate');
```

→ Identifică modelul pe stare cu număr optimal de stări în gama 1:10 și transmite modelului inclusiv starea inițială estimată.

S.x0

⚡ Toate, cu coeficienți necunoscuți.

Se va detecta cel mai bun din gama 1:10.

- Dacă nu se cunoaște, se poate preciza, în loc, 'best'.
- Pentru fiecare pas de întârziere dintre cele n_k , se adaugă câte o stare virtuală.

⚡ Rutina are uneori un comportament neașteptat de slab (pentru anumite valori n_x).

Trusa de instrumente IDENT din MATLAB

Interfața grafică IDENT

Permite compararea diferitelor modele de identificare.

- Lansată în execuție prin comanda:

>> ident

The screenshot displays the MATLAB IDENT software interface, which is used for system identification. It consists of several main windows:

- ident: Untitled**: The main workspace. It features a central area with a 'Preprocess' button and a 'Working Data' plot. To the right, there is a grid of 'Model Views' showing various ARX models (e.g., arx441, arx444, arx244, arx222, etc.). Below the grid, there are checkboxes for 'Time plot', 'Data spectra', 'Frequency function', 'Model output', 'Transient resp', 'Frequency resp', 'Zeros and poles', and 'Noise spectrum'. At the bottom, there is a 'Trash' icon and a 'Validation Data' plot.
- Model Output: temperature**: A window showing a plot of 'Measured and simulated model output' over 'Time'. The plot displays multiple colored lines representing different model fits. To the right of the plot, a 'Best Fits' list shows the results for various models, including their R-squared values (e.g., arx5601: 90.6, arx5501: 90.29, etc.).
- Parametric Models**: A window for configuring the parametric model. It includes fields for 'Structure' (ARX: [na nb nk]), 'Orders' (5 100 1), 'Equation' (Ay=Bu+e), 'Method' (ARX selected), and 'Name' (arx51001). It also has sections for 'Focus' (Simulation), 'Dist.model' (Estimate), 'Initial state' (Auto), and 'Covariance' (Estimate). At the bottom, there are buttons for 'Order Selection', 'Order Editor...', 'Estimate', 'Close', and 'Help'.

- Interfața este convivială și poate fi descoperită pas cu pas de către utilizator.

① Trusa de instrumente IDENT din MATLAB

🔑 Probleme de simulare

Problema 1.1

Utilizați comenzile **help** și **helpwin** pentru a obține informații despre funcțiile MATLAB prezentate. Apoi, elaborați mici programe, prin care să testați funcționarea acestor funcții. Datele se pot genera pseudo-aleator cu ajutorul funcțiilor **rand** și/sau **randn**. O funcție interesantă pentru generarea de semnale de stimul este și **idinput**. (Obiectivul este acela de a vă obișnui cu maniera de apelare a acestor funcții și nu neapărat de a verifica precizia rezultatelor oferite de ele.)

Problema 1.2

1. După modelul funcției **make_DATA** proiectați o rutină care să construiască un obiect de tip **IDDATA** pentru modele MIMO complete. (Cu alte cuvinte, adăugați și intrările achiziționate în blocul de date). Testați corectitudinea rutinei.
2. Proiectați o rutină care să construiască un obiect de tip **IDSS**.

Problema 1.3

Lansați în execuție interfața grafică de identificare **IDENT** și testați toate facilitățile acesteia, plecând de la **Import data > Example**.