❶ 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** şi conţine rutine implementate în tehnologia **Programării Orientate pe Obiecte**.

2 p

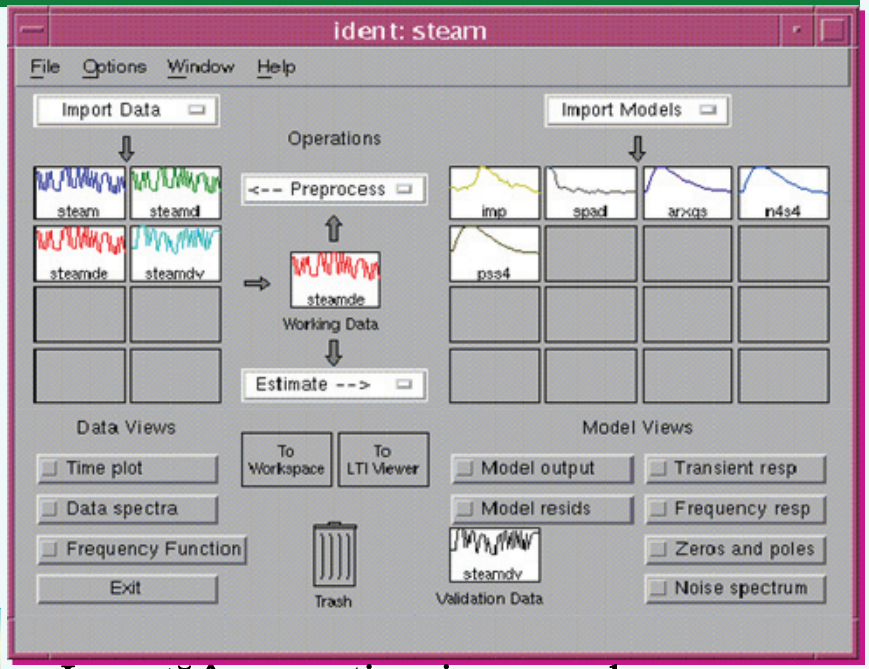
Mai mult ⇒ Există şi o interfaţă grafică prietenoasă cu utilizatorul (GUI), pentru efectuarea de experimente de identificare comparative.

Obiecte IDENT principale

- IDDATA
- IDMODEL
- IDSS

Cîteva funcţii IDENT frecvent utilizate

- armax/arx
- bj/oe
- compare
- iddata
- ident
- idarx
- idpoly
- iv4
- n4sid
- pem
- predict
- sim



- Lansată în execuţie prin comanda:

```
>> ident
```

❶ Trusa de instrumente IDENT din MATLAB

IDDATA

D Structură de date intrare-ieşire generate folosind metoda constructor **iddata** asociată obiectului **IDDATA** (date de identificare). Cele 2 matrici de date **u** (de intrare) şi **y** (de ieşire) pot avea dimensiuni identice: **N** linii şi **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** şi datele provin de la un model MISO. Datele se regăsesc în cîmpurile **D.u** şi **D.y**. Structura se compune din următoarele cîmpuri:

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

☝ Există blocuri de date în domeniul timpului sau în domeniul frecvenţei.

← date de ieşire

← date de intrare

← perioadă de eşantionare

➔ Pentru mai multe detalii:

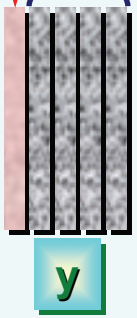
```
>> idprops iddata
```

❶ Trusa de instrumente IDENT din MATLAB

Cum se poate crea un bloc de date?

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

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



☝ Momente de eşantionare. (eventual)

- Datele primare sunt reprezentate în diferite formate, dintre care MATLAB poate recunoaşte următoarele, uzuale:
  - ➔ **ASCII (`.M`)** (similare programelor principale)
  - ➔ **ASCII (`.TXT`)** (cu datele precizate pe coloane)
  - ➔ **Tabel (`.XLS`)** (accesibile cu MS Office - Excell)
  - ➔ **Criptat (`.MAT`)** (create şi 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 fişierelor de proveniență sau, dacă este vorba despre un fişier **`.MAT`**, prin numele atribuite lor înainte de salvarea în acel fişier.

❶ Trusa de instrumente IDENT din MATLAB

Exemple de fişiere 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
%
```

☝ Observaţi maniera de atribuire a variabilei care conţine datele.

☝ Pot fi prezente şi alte informaţii (auxiliare) privind datele.

❶ Trusa de instrumente IDENT din MATLAB

Exemple de fişiere primare de date (final)

➔ ASCII (.TXT)

➔ Tabel (.XLS)

Date pe canale de măsură

hxstep.txt

File Edit Format View Help

0.0000	0.0060
0.0033	0.0110
0.0097	0.2968
0.0137	0.0053
0.0169	0.2899
0.0187	0.2308
0.1098	0.3231
0.2578	0.3131
0.4264	0.3148
0.6011	0.3182
0.7660	0.3029
0.9166	0.2979
1.0496	0.2994
1.1610	0.2792
1.2532	0.3094
1.3281	0.2948
1.3873	0.3045
1.4408	0.3122
1.4853	0.2585
1.5255	0.2994
1.5577	0.3071
1.5743	0.2916
1.5927	0.3074
1.6113	0.2850
1.6266	0.2916
1.6408	0.2962
1.6409	0.2725
1.6426	0.3065
1.6462	0.3051
1.6464	0.3005
1.6485	0.3245
1.6489	0.2620
1.6504	0.3131
1.6520	0.2902
1.6529	0.2936
1.6538	0.2527
1.6543	0.2884
1.6583	0.2890
1.6587	0.3034
1.6588	0.3154
1.6600	0.2766
1.6627	0.3100
1.6639	0.2879
1.6643	0.2937
1.6644	0.2962
1.6650	0.2412
1.6726	0.2884
1.6737	0.2913

Momente de eşantionare.

Microsoft Excel - umiditate ambient.xls

File Edit View Insert Format Tools Data Window

This copy of Office is not genuine. Click here to learn more o

Calibri 11

A40 05-03-2010 10:00:00

	A	B	C	D	E
1	result_time	humidity	humidity_min	humidity_max	
2	03-03-10 20:00	40.1144065	39.895107	40.245899	
3	03-03-10 21:00	39.846647	39.763084	39.963188	
4	03-03-10 22:00	39.6537675	39.595299	39.762035	
5	03-03-10 23:00	39.79450625	39.695988	39.892998	
6	04-03-10 00:00	39.846541	39.79137	39.924404	
7	04-03-10 01:00	39.9021015	39.821743	39.980854	
8	04-03-10 02:00	40.0906145	40.01223	40.170033	
9	04-03-10 03:00	40.23862075	40.166851	40.325493	
10	04-03-10 04:00	40.4025915	40.323364	40.481796	
11	04-03-10 05:00	40.6002005	40.513027	40.670139	
12	04-03-10 06:00	40.735178	40.701321	40.766888	
13	04-03-10 07:00	40.16945	39.885624	40.453754	
14	04-03-10 08:00	39.210968	38.718647	39.635151	
15	04-03-10 09:00	38.07919025	37.790737	38.465534	
16	04-03-10 10:00	37.50352633	37.466496	37.54155	
17	04-03-10 11:00	37.70030025	37.426128	38.050858	
18	04-03-10 12:00	38.01164825	37.920845	38.14402	
19	04-03-10 13:00	38.3958565	38.051025	38.740688	
20	04-03-10 14:00	38.12345675	38.054138	38.20322	
21	04-03-10 15:00	37.450326	35.383911	38.167187	
22	04-03-10 16:00	38.950432	38.082409	40.065468	
23	04-03-10 17:00	41.439331	40.486614	42.772568	
24	04-03-10 18:00	42.8757745	41.518456	44.302143	
25	04-03-10 19:00	43.74478667	43.135494	44.297234	
26	04-03-10 20:00	43.94111	42.92429	44.758427	
27	04-03-10 21:00	44.7561475	42.663506	45.717289	
28	04-03-10 22:00	40.656144	40.228954	41.288315	
29	04-03-10 23:00	39.9255685	39.822906	40.048161	
30	05-03-10 00:00	39.754987	39.708542	39.798798	
31	05-03-10 01:00	39.63815567	39.614006	39.682274	
32	05-03-10 02:00	39.555626	39.492001	39.622368	
33	05-03-10 03:00	39.59315875	39.492001	39.761013	
34	05-03-10 04:00	39.58655175	39.460434	39.830311	

Draw

Ready

Nu este obligatoriu ca prima coloană să includă momente de eşantionare.

❶ Trusa de instrumente IDENT din MATLAB

Şi totuşi... 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 şi 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 sould 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 fiels DATA.UserData can be used. For example:
%             DATA.UserData = '29-Nov-2007' ;
%
%       Empty input enforces empty output.
%
%       Uses:     VECTORIZE
%                 WAR_ERR
```

❶ 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 ;
        if (nargin < 1)
            war_err(E1) ;
            return ;
        end ;
        if (isempty(y))
            war_err(E1) ;
            return ;
        end ;
%
% Building the DATA object
% ~~~~~~~~~~~~~~~~~~~~~~~~
%
        if (isscalar(y) || isvector(y))          % Storing the data ...
            DATA.y = vectorize(y).' ;
            DATA.Ts = 1 ;                        % the sampling period ...
        else
            DATA.y = y(:,2:end) ;
            DATA.SamplingInstants = y(:,1) ;     % and the sampling instants (if any).
        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))                   % Setting the starting date and/or time
            FN = input(I5,'s') ;                 % (only allowed for uniform sampling).
            if (isempty(FN))
                DATA.Tstart = now ;
            else
                DATA.Tstart = datenum(FN) ;
            end ;
        else                                     % Here the starting date can be specified as a string
            DATA.UserData = input(I6,'s') ;      % in a preferred format (such as 'dd-Mmm-yyyy').
        end ;
        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
%
```

aici se crează obiectul **DATA** vid, care va fi completat ulterior cu informaţie

☝ Dacă sunt mai mult de 2 coloane, se consideră că prima include momente de eşantionare.

❶ Trusa de instrumente IDENT din MATLAB

IDMODEL

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

polinoamele modelului → (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

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

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

```
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)'
```

indicii structurali → (na,nb,…,nf)

```
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)'
```

timpul mort →

```
nk:  'delay of B-polynomial (scalar)'
InitialState:  [1x45 char]
         Name:  'string'
```

perioada de eşantionare →

```
           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'
```

→ Pentru mai multe detalii:

```
>> idprops idmodel
```

Evident, polinoamele A şi B se regăsesc în cîmpurile: M.a, respectiv M.b. În M.b sunt salvaţi atît coeficienţii nenuli cît şi cei nuli (datoraţi întîrzierii intrinseci) ai polinomului B. Ordinele polinoamelor sunt memorate în M.na, respectiv M.nb, iar întîrzierea intrinsecă – în M.nk.

❶ Trusa de instrumente IDENT din MATLAB

IDSS

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

`s` Obiectul `IDSS` (model de identificare pe stare) conține cîmpurile:

matricile modelului ➔ (A,B,…,K)

```
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)'
```

starea inițială ➔

```
X0:  'Initial states (Nx-by-1 matrix)'
```

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

RSS

```
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'
```

perioada de eșantionare ➔

```
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'
```

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

❶ Trusa de instrumente IDENT din MATLAB

Ce legătură există între obiectele `IDMODEL` şi `IDSS`?

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

MIO

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

obiect `IDSS` obiect `IDMODEL`

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

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

MRS

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

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

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

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

❶ 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
```
➔ pentru modele I/O liniare polinomiale

☝ A doua – numai pentru modele ARX.

```
>> idss
>> idgrey
```
➔ pentru modele liniare pe stare

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

☝ Se va reveni asupra acestui subiect.

- Apeluri întrebuințate frecvent:

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

```
>> M.a = A ;
>> M.b = B ;
>> M.c = C ;
>> M.d = D ;
>> M.f = F ;
```
➔ ... apoi atribuie polinoamele.

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

$B = [\underbrace{0 \; \dots \; 0}_{nk \geq 1} \; b_1 \; b_2 \; \dots \; b_{nb}]$ (întîrzierea intrinsecă)

$\dots$

$F = [1 \; f_1 \; f_2 \; \dots \; f_{nf}]$

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

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

☝ Ultimele polinoame pot lipsi.

☝ Unele polinoame pot fi vide.

❶ 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:

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

indici structurali

întîrzierea intrinsecă (implicit 1)

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 şi u de tip date preliminare (vectori/matrici de date aşezate pe coloane).

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

☝ În cazul seriilor de timp.

Metoda de identificare

Minimizarea Erorii de Predicție (MMEP)

❶ 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

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

Matrici de polinoame

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

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

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

Semnale ➔ $\mathbf{y} \in \mathbb{R}^{ny}$ $\mathbf{u} \in \mathbb{R}^{nu}$ $\mathbf{e} \in \mathbb{R}^{ny}$ — aceeaşi dimensiune

☝ Există tot atîtea perturbaţii cîte canale de măsură.

Ecuaţia generică ARMAX-MIMO

$$\sum_{j=1}^{ny} \mathrm{A}_{l,j}\left(q^{-1}\right) y_j[n] = \sum_{i=1}^{nu} \mathrm{B}_{l,i}\left(q^{-1}\right) u_i[n] + \sum_{j=1}^{ny} \mathrm{C}_{l,j}\left(q^{-1}\right) e_j[n] \quad \forall n \in \mathbb{N}, \ \forall l \in \overline{1,ny}$$

polinoame de grad na | polinoame de grad nb | polinoame de grad nc

☝ Avînd coeficienţi necunoscuţi.

☝ Necunoscută.

Matrice de covarianţă a zgomotelor

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

☝ Toate polinoamele unei matrici au acelaşi grad.

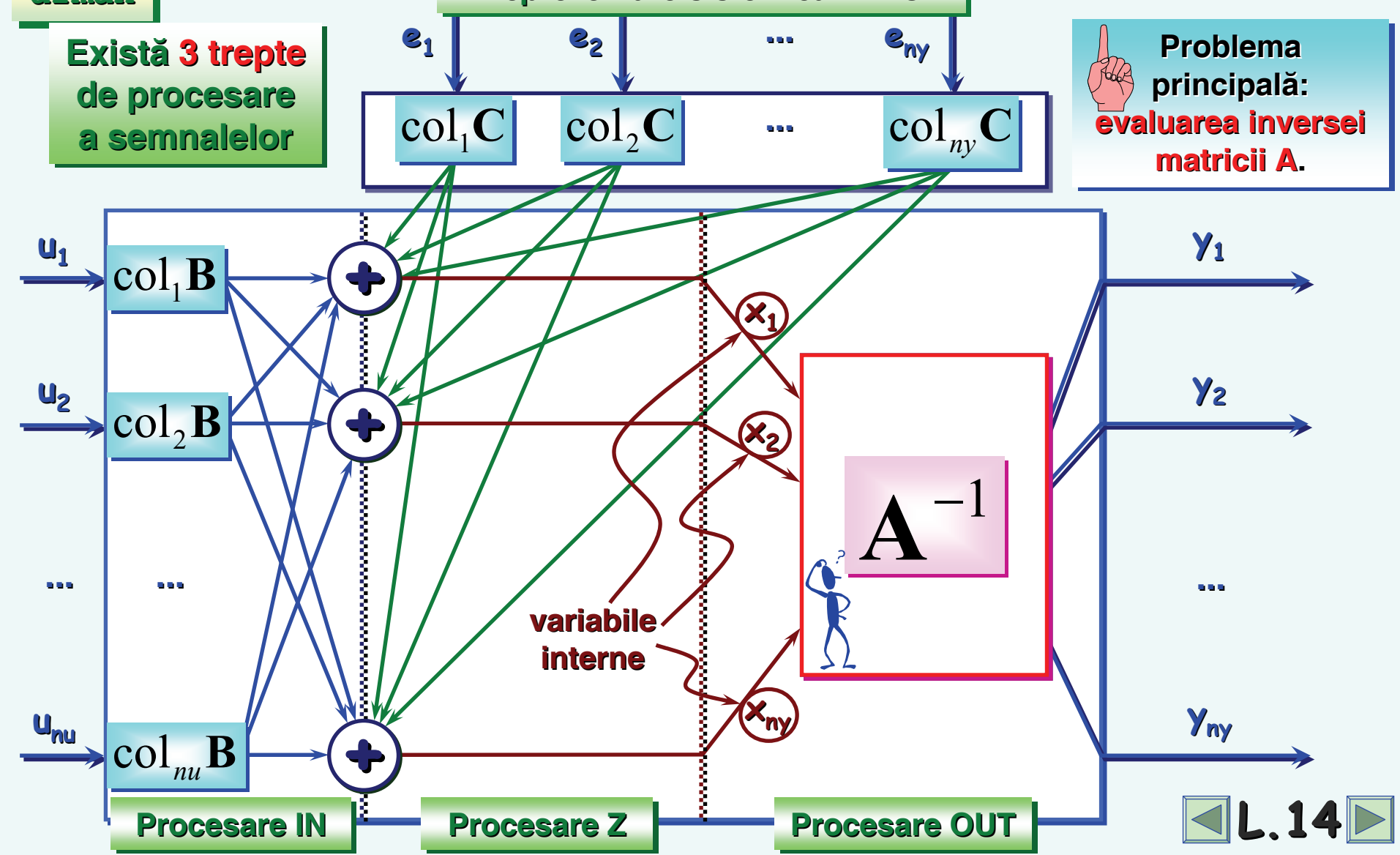
❶ Trusa de instrumente IDENT din MATLAB

Principalele rutine de identificare (3/9)

armax

Există 3 trepte de procesare a semnalelor

Reprezentare sistemică MIMO



Problema principală: evaluarea inversei matricii A.

❶ Trusa de instrumente IDENT din MATLAB

Principalele rutine de identificare (4/9)

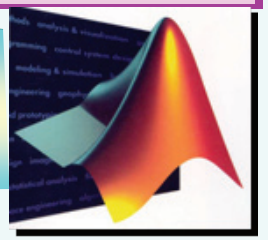
armax

Cum ar putea fi inversată matricea de polinoame A?

În general, este dificil, dacă nu imposibil.

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

☝ Modelul va pierde însă din precizie.



Adoptarea unui model AR-MO

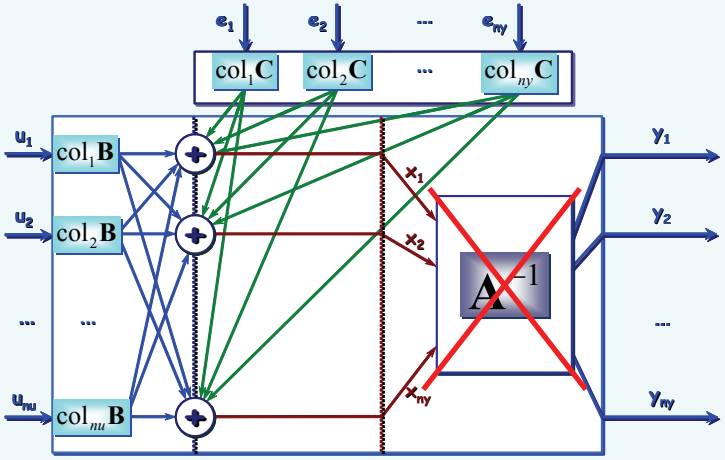
☝ Dacă nu există date de intrare.

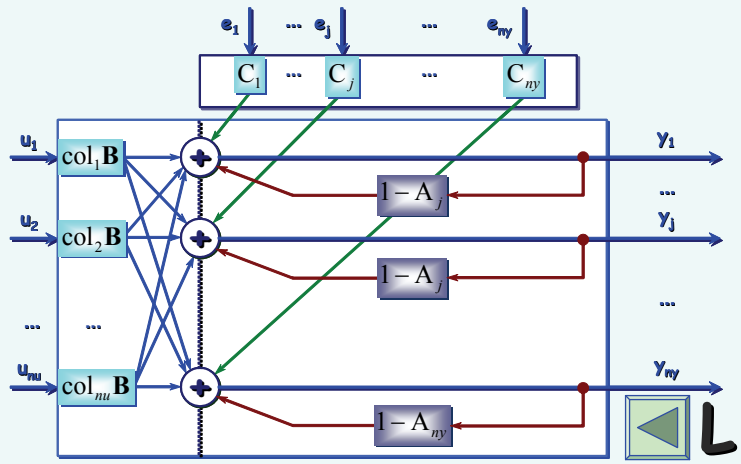
Ce principiu?

Adoptarea principiului din MATLAB

Modelul MIMO este văzut ca o colecţie de ny modele MISO.

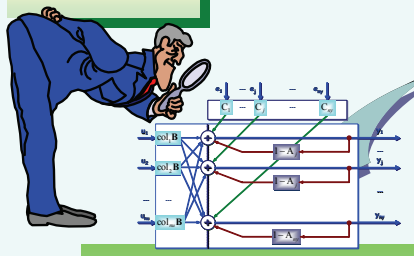
Fiecare model MISO codifică influenţa pe care o au toate intrările asupra unei anumite ieşiri, corupte doar de zgomotul canalului său (nu şi de zgomotele celorlalte canale).





1 Trusa de instrumente IDENT din MATLAB

armax

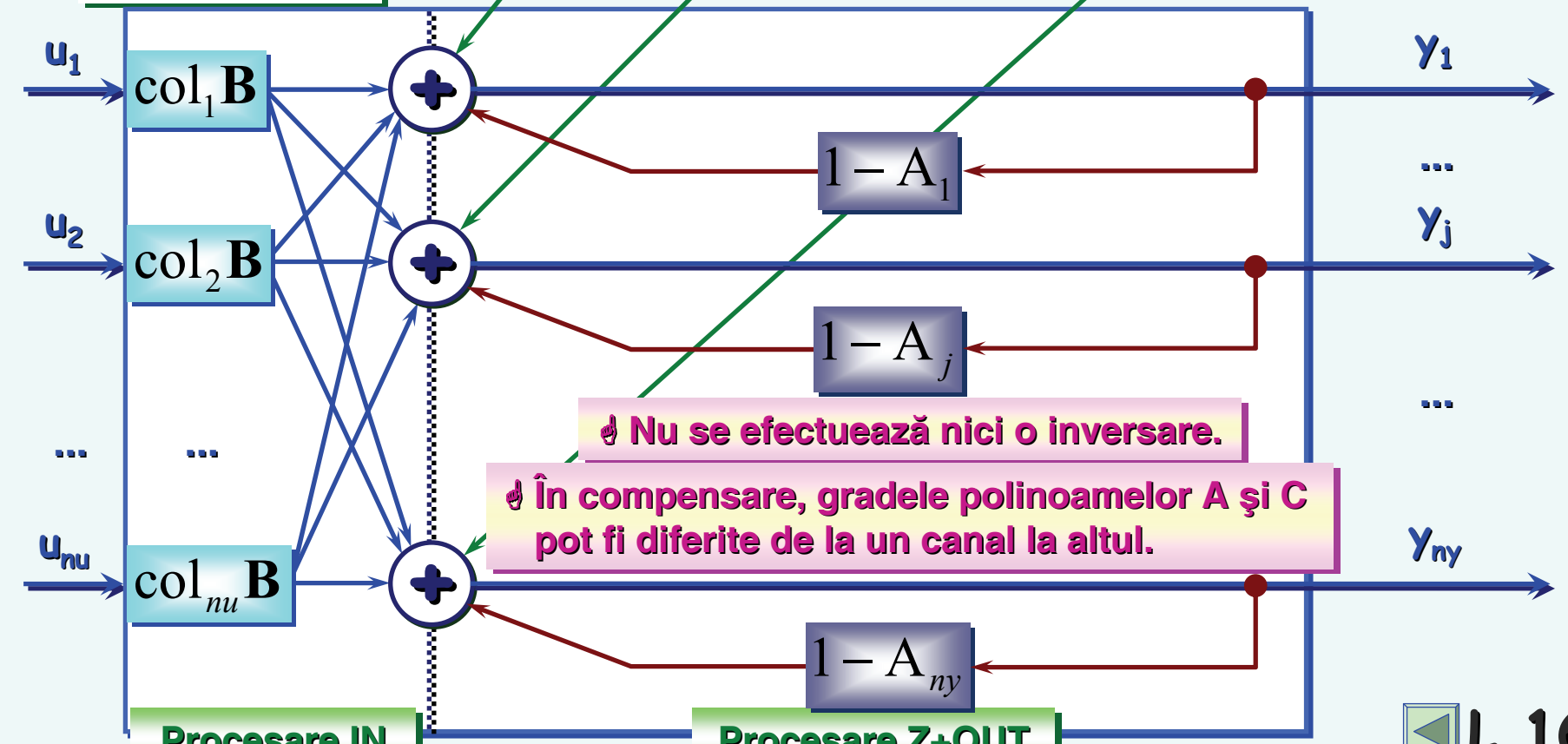


Principalele rutine de identificare (5/9)

Reprezentarea sistemică multi-MISO

Acum, există doar 2 trepte.

Practic, matricile A şi C sunt diagonale.



Nu se efectuează nici o inversare.

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

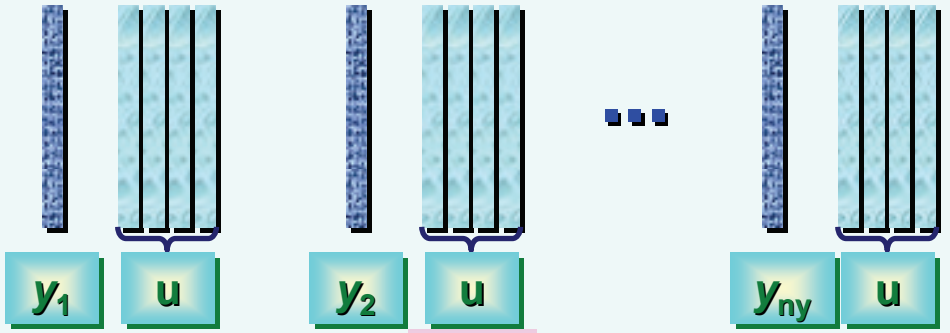
❶ Trusa de instrumente IDENT din MATLAB

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 ny ori.



- O funcție înrudită tentantă pentru Automatică este şi: `arx` (pentru identificarea modelelor ARX)

Totuşi ⇒ Funcția `arx` utilizează **Metoda Celor Mai Mici Pătrate** (**MCMMP**), fiind mai puțin precisă decît `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).
- Totuşi, dacă se doreşte, ea poate fi apelată ca în cazul funcției `armax`:

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

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

❶ 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:

RIO cu A=1 — indici structurali — întîrzierea intrinsecă (implicit 1)

```
>> 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 şi cu ajutorul funcției **bj**.

☝ Funcția **bj** este însă mai lentă decît funcția **armax**.

- O altă funcție care identifică un model particular este şi: **oe** (pentru identificarea modelelor de tip Eroare de Ieşire – Output Error)
- Modelele multi-MISO se pot identifica de asemenea, folosind funcția **bj**.

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

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

Exemplu

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

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

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

❶ 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 aceleaşi notaţii ca în cazul funcţiei **armax**.

levinson

- Identifică modele din clasa **AR**, folosind doar date de ieşire măsurate, pe baza:

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

Algoritmului Levinson-Durbin (ALD)

- Apel tipic:

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

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

☝ Care este şi valoarea maximă posibilă.

vectorul parametrilor

vectorul de auto-covarianţă a datelor

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

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

$K \geq na$

- **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.

… care se evaluează astfel:

```
>> [ry,k] = xcorr(y,K);
>> ry(k<0) = [];
```

❶ 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)

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

Matrici de parametri

$\mathbf{A} \in \mathbb{R}^{nx \times nx}$ $\mathbf{B} \in \mathbb{R}^{nx \times nu}$ $\mathbf{F} \in \mathbb{R}^{nx \times ny}$

$\mathbf{C} \in \mathbb{R}^{ny \times nx}$ $\mathbf{D} \in \mathbb{R}^{ny \times nu}$

Matrice de covarianță a zgomotelor (diagonală) $\boldsymbol{\Lambda} \in \mathbb{R}^{ny \times ny}$

☝ Toate, cu coeficienți necunoscuți.

- Apel tipic:

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

obiect IDSS — obiect IDDATA — numărul dorit de stări

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

- Dacă nu se cunoaşte, se poate preciza, în loc, `'best'`.
- Pentru fiecare pas de întîrziere dintre cele `nk`, se adaugă cîte o stare virtuală.

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

Exemplu

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

➔ Identifică modelul pe stare cu număr optimal de stări în gama 1:10 şi transmite modelului inclusiv starea inițială estimată. `S.X0`

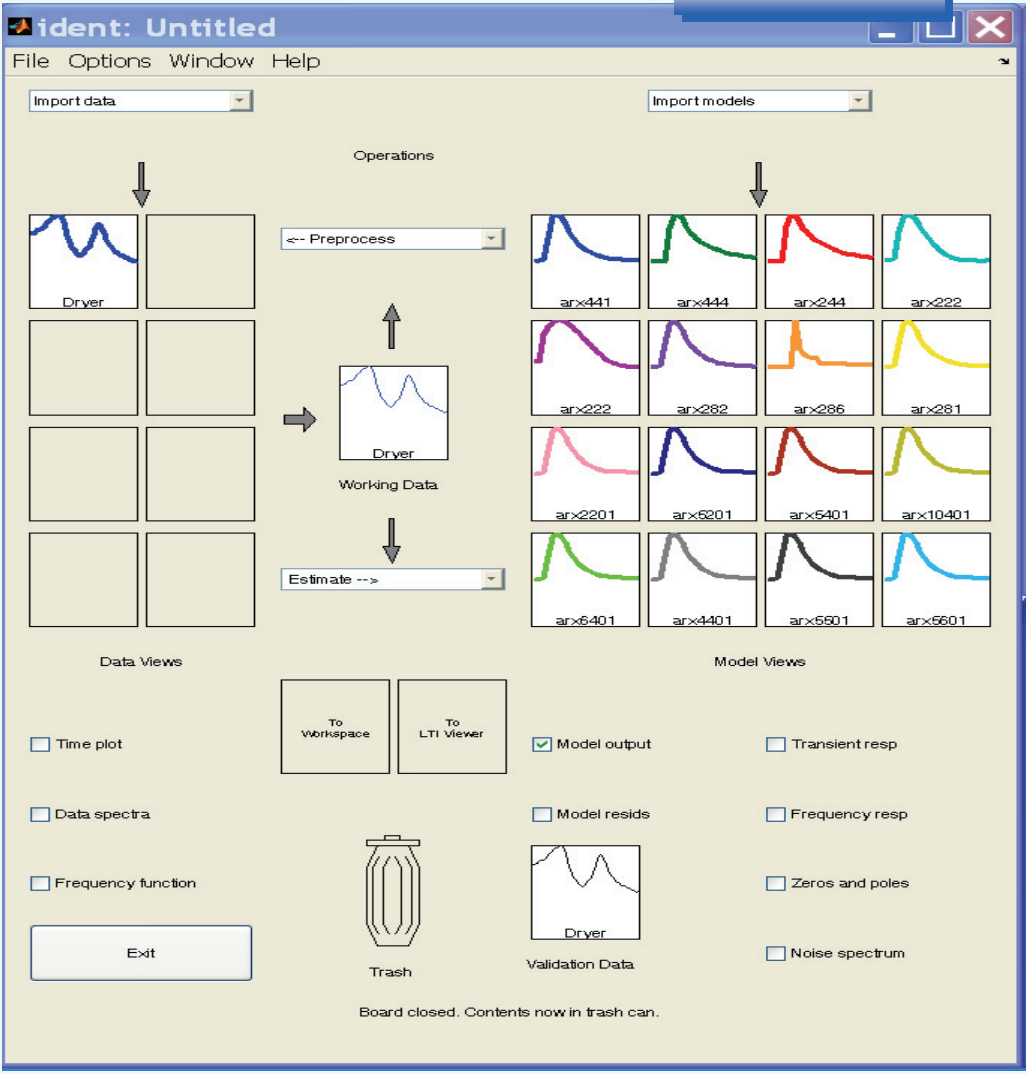
☝ Rutina are uneori un comportament neaşteptat de slab (pentru anumite valori `nx`).

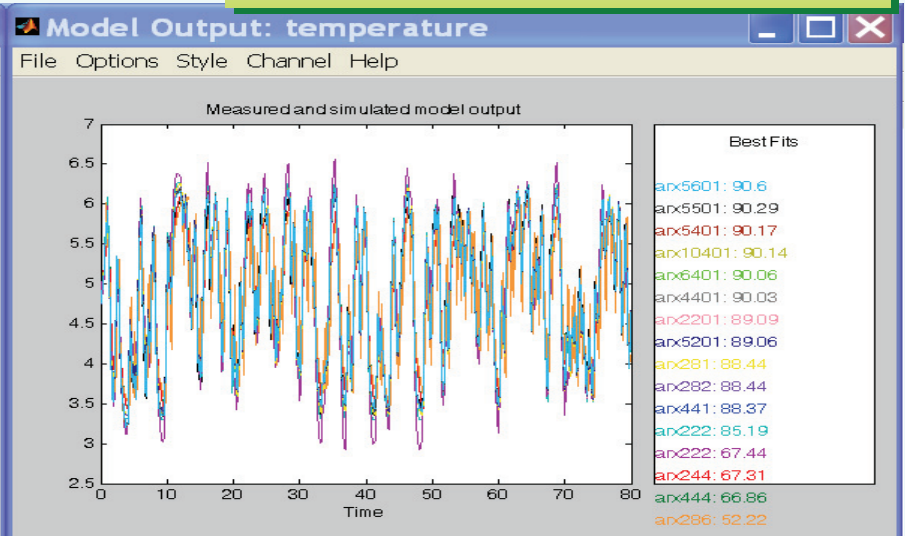
❶ Trusa de instrumente IDENT din MATLAB

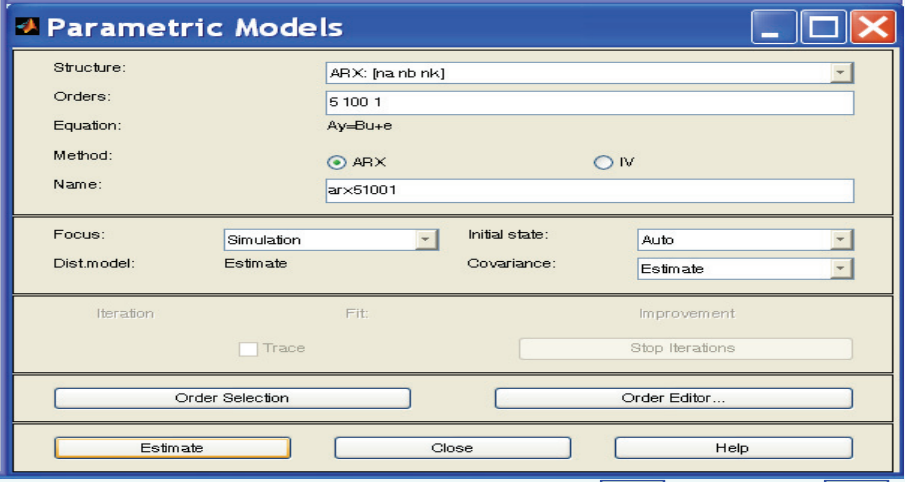
Interfața grafică IDENT

Permite compararea diferitelor modele de identificare.

- Lansată în execuție prin comanda: `>> ident`







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

❶ Trusa de instrumente IDENT din MATLAB

☞ Probleme de simulare

Problema 1.1

Utilizați comenzile **help** şi **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** şi/sau **randn**. O funcție inetersantă pentru generarea de semnale de stimul este şi **idinput**. (Obiectivul este acela de a vă obişnui cu maniera de apelare a acestor funcții şi nu neapărat de a verifica precizia rezultatelor oferite de ele.)

Problema 1.2

1.5p

0.5p

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 şi 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** şi testați toate facilitățile acesteia, plecînd de la **Import data > Example**.