Functii utilizate:

**Plot(**t,u,t,y): t – timpul, u – comanda de tip treapta, y – raspunsul la comanda de tip treapta

**Legend**(‘Comanda de tip treapta’, ‘Raspunsul indicial’)

**Idinput**: pentru generarea semnalului SPAB

u = idinput(N)  
u = idinput(N,type,band,levels)  
[u,freqs] = idinput(N,'sine',band,levels,sinedata)

Argumente de intrare:

'rgs' — Gives a random, Gaussian signal.

'rbs' — Gives a random, binary signal. This is the default.

'prbs' — Gives a pseudorandom, binary signal.

'sine' — Gives a signal that is a sum of sinusoids.

band = [wlow, whigh]. Pentru ‘prbs’ band = [0 1]

levels = [minu, maxu]

Rutina apelata va fi de forma:

u = idinput(L,tip\_semnal,[freq\_min freq\_max],[u0-Du u0+Du])

**Iddata** pentru generarea unei structuri de date:

data = iddata(y,u,Te)

save('XXBY\_DateExper.mat','data') % data este structura in care s-a salvat experimentul,

Pentru identificarea modelului: armax, bj, levinson

sys = armax(data,[na nb nc nk]) : pentru modele de tip ARMA, ARMAX

sys = bj(data, [nb nc nd nf nk]) : pentru modele de tip BJ

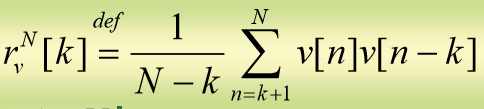
[a,e] = levinson(r,na) : pentru modele de tip AR

Data – structura de tip iddata

Na,nb,nc,nd,nf– gradele polinoamelor

Nk – timpul mort, daca exista

R – secventa de autocovarianta calculate cu formula:



Pentru filtrare se folosesc functiile: detrend, getdetrend,butter,filter

Eliminarea componentelor continue pentru comanda si iesire

y = detrend(x) removes the best straight-line fit from vector x and returns it in y. If x is a matrix, detrend removes the trend from each column.

y = detrend(x,'constant') removes the mean value from vector x or, if x is a matrix, from each column of the matrix.

y = detrend(x,'linear',bp) removes a continuous, piecewise linear trend from vector x or, if x is a matrix, from each column of the matrix. Vector bpcontains the indices of the breakpoints between adjacent linear segments. The breakpoint between two segments is defined as the data point that the two segments share.

Filtrarea zgomotelor (doar pentru procesul de presiune). Se foloseste un filtru trece jos tip buffer de ordin 1 si pulsatie pentru filtrarea iesirii:

hd = design(d,'butter')  
hd = design(d,'butter',designoption,value...)

[y](https://www.mathworks.com/help/matlab/ref/filter.html#outputarg_y) = filter([b](https://www.mathworks.com/help/matlab/ref/filter.html" \l "inputarg_b),[a](https://www.mathworks.com/help/matlab/ref/filter.html#inputarg_a),[x](https://www.mathworks.com/help/matlab/ref/filter.html#inputarg_x)) filters the input data x using a [rational transfer function](https://www.mathworks.com/help/matlab/ref/filter.html#buagwwg-2) defined by the numerator and denominator coefficients b and a.

If a(1) is not equal to 1, then filter normalizes the filter coefficients by a(1). Therefore, a(1) must be nonzero.

If x is a vector, then filter returns the filtered data as a vector of the same size as x.

If x is a matrix, then filter acts along the first dimension and returns the filtered data for each column.

If x is a multidimensional array, then filter acts along the first array dimension whose size does not equal 1.

Pentru salvare: **save(‘XXBY\_IdentData’,’eData’,’vData’);**

Pentru estimarea complexitatii si a intarzierii: advice si delayest

advice(data)

nk = delayest(Data)

Pentru modele ARX. Se folosesc succesiv functiile: struc, arxstruc, selstruc

*nn* = struc(*na*,*nb*,*nk*)

V = arxstruc(ze,zv,nn)

Unde ze – structura de date estimate si zv – structura de date pentru validare

nn = selstruc(v)

Pentru validare: resid si compare

e = resid(m,vData);% m este modelul identificat; vData – modelul filtrat al datelor de validare;

plot(e);

compare(vData,m);

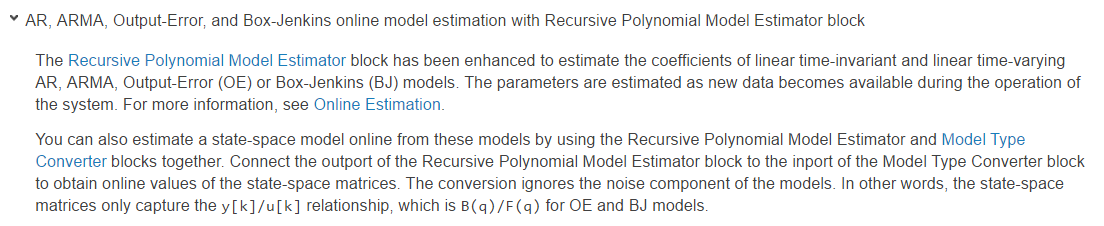
resid(m,vData,’corr’);

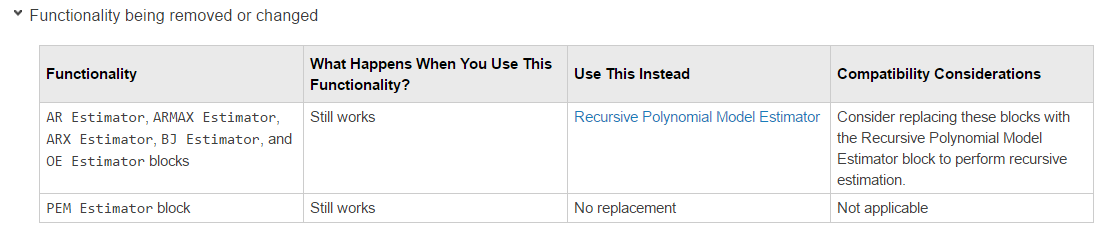
Pentru conversia din binar in decimal: d = bi2de(b). Inainte de efectuarea conversiei se ultilizeaza functia fliplr B = fliplr(A) astfel incat sa se tina cont de MSP(functia bi2de considera MSB ca fiind ultima valoare din secventa binara: ex: 1010 ->5, in realitate este 10, pentru asta se inverseaza secventa binara)

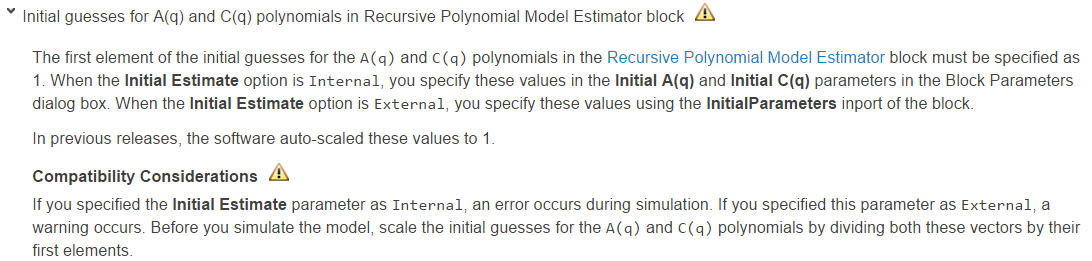
Harta polizerouri se obtine cu functia : h = pzplot(sys)

Compatibilitate intre diferite versiuni:

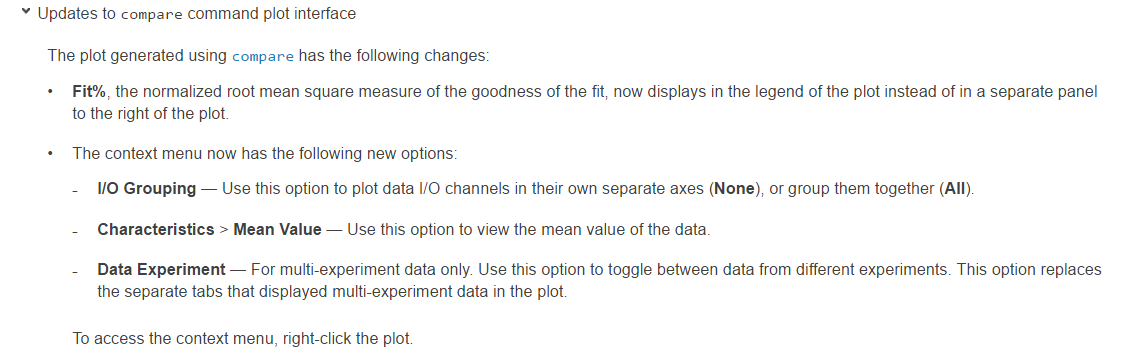
#### R2014a: plot command for iddata object enhanced

R2014b: 

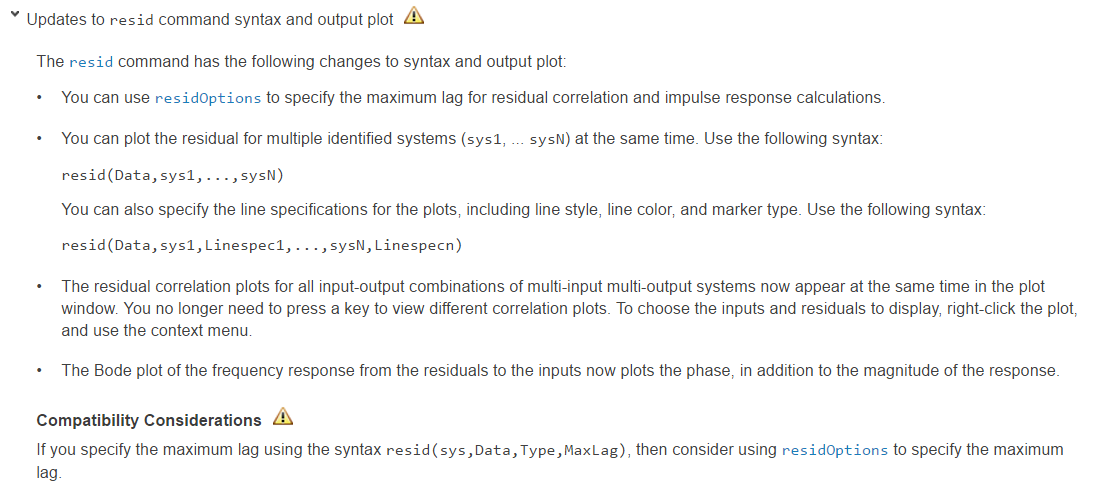


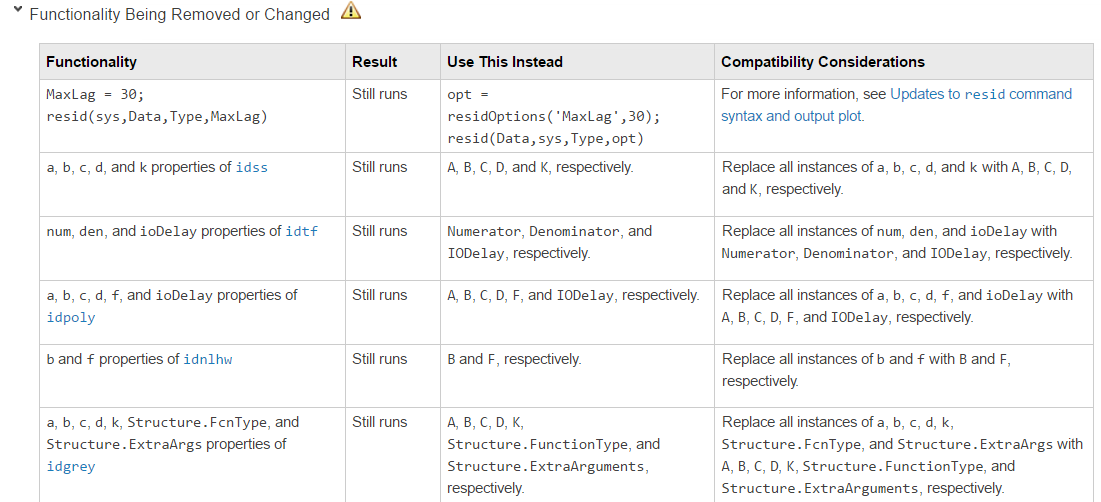


R2015b:

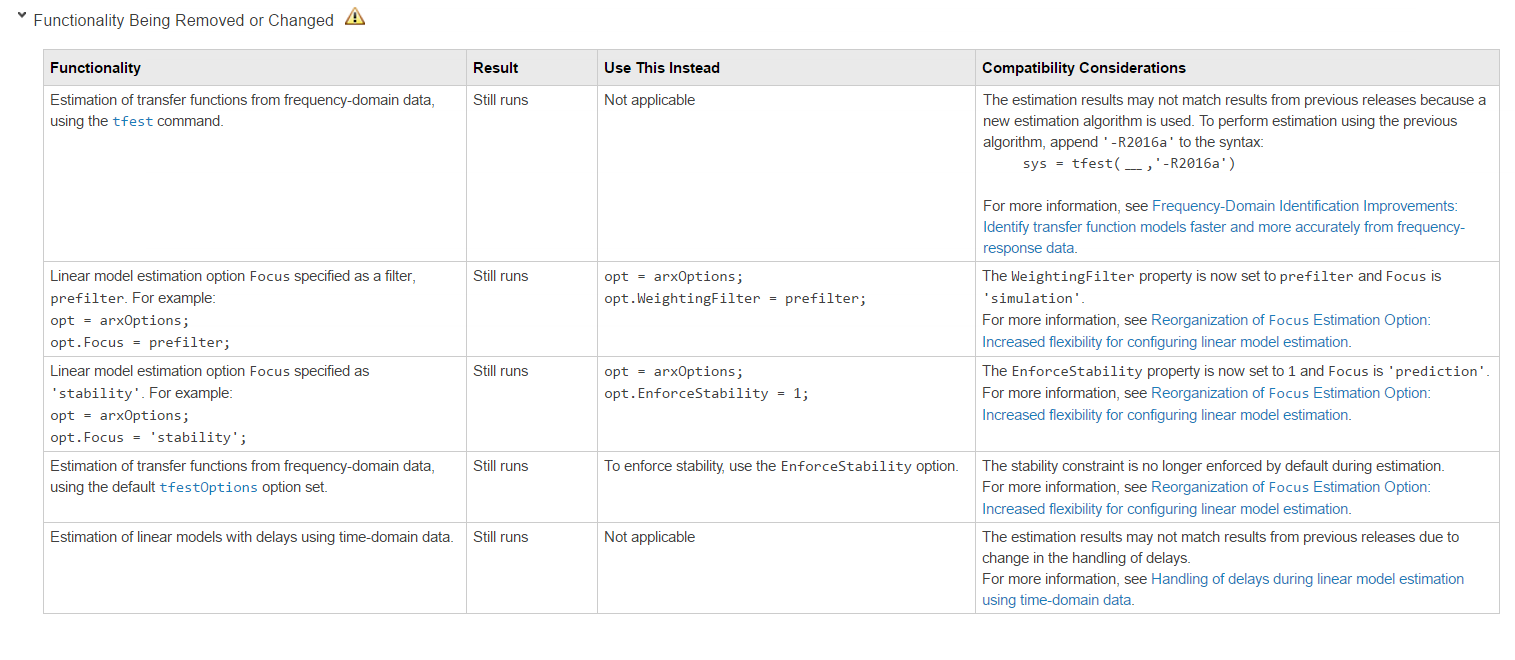


R2016a:





R2016b:



Pentru detalii suplimentare referitoare la compatibilitatea intre versiuni:

consultati:https://www.mathworks.com/help/ident/release-notes.html

Pentru detalii suplimentare referitoare la comenzi din Matlab se recomanda folosirea functiei help + nume\_comanda in Matlab sau google.com