Adaptive Control

Part 3: Open Loop System Identification - ***A Brief Review***

Why to review “open loop system identification” ?

• We need information upon the complexity of the plant model and its basic dynamical features for building an “adaptive control scheme”

•Recursive identification algorithms are used in indirect adaptive control

•It is an introduction to “identification in closed loop “ which is used for “Iterative identification in closed loop and controller redesign” (an adaptation technique)

OUTLINE

Open loop system identification

-Data acquisition

-Model complexity

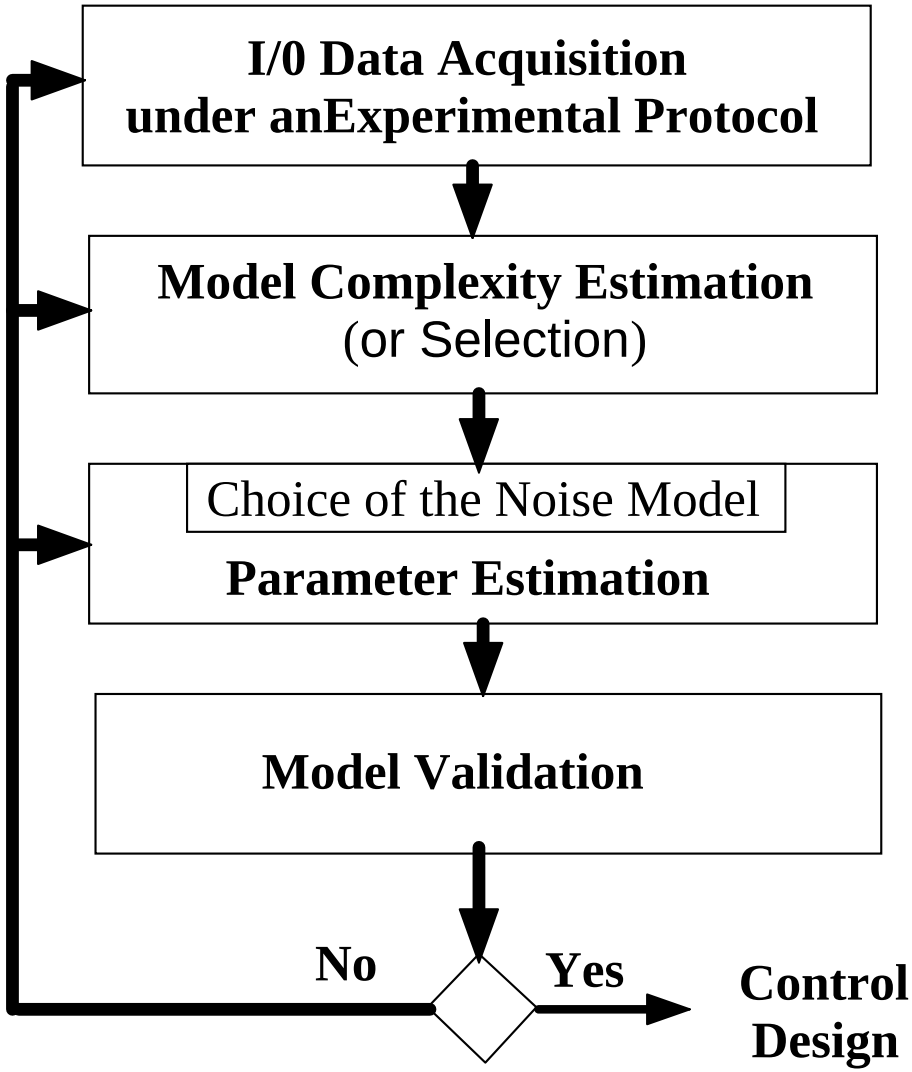
-Parameter estimation

-Validation

Objective of system identification (for control)

To extract from experimental data a dynamic model of the plant which will allow to design a controller in order to match the control specifications

System Identification Methodology



I/O Data Acqusition

Signal : *a P.R.B.S sequence*

Magnitude : *few % of the input operating point*

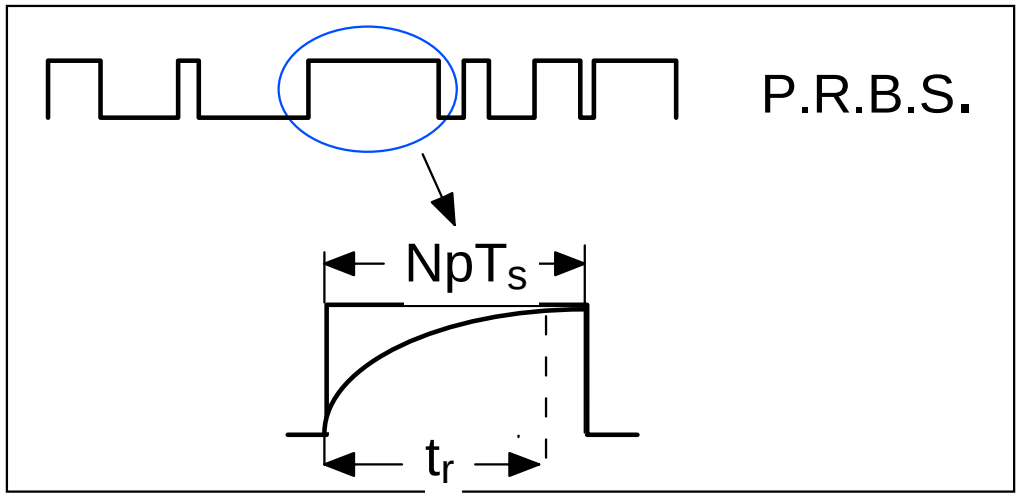
Clock frequency : $f_{clock} = (1/\ p)\, f_s\ ;\ p = 1, 2, 3\ (\ f_s = sampling\ \ frequency)$

Length : $(2^{N-1} - 1)\, pT_s\ ;\ N = number\ of\ cells, T_s = 1/\ f_s$

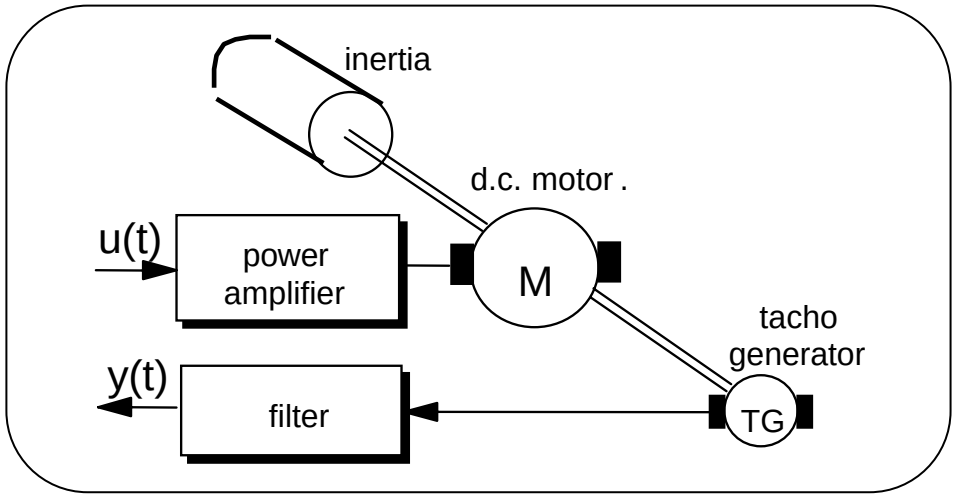
Largest pulse : NpT_s

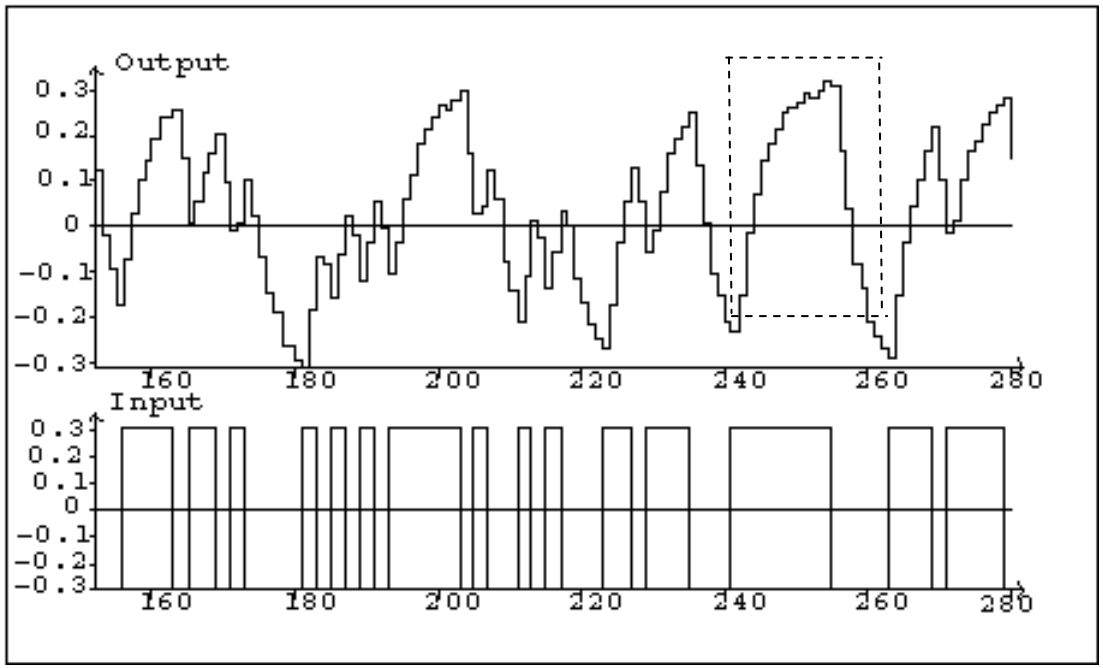
Length : < allowed duration for the experiment

Largest pulse : $\geq t_R$ (rise time)



An I/O File


inertia
d.c. motor .
u(t)
power amplifier
M
tacho generator
y(t)
filter
TG


Output
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
160
180
200
220
240
260
280
Input
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
160
180
200
220
240
260
280

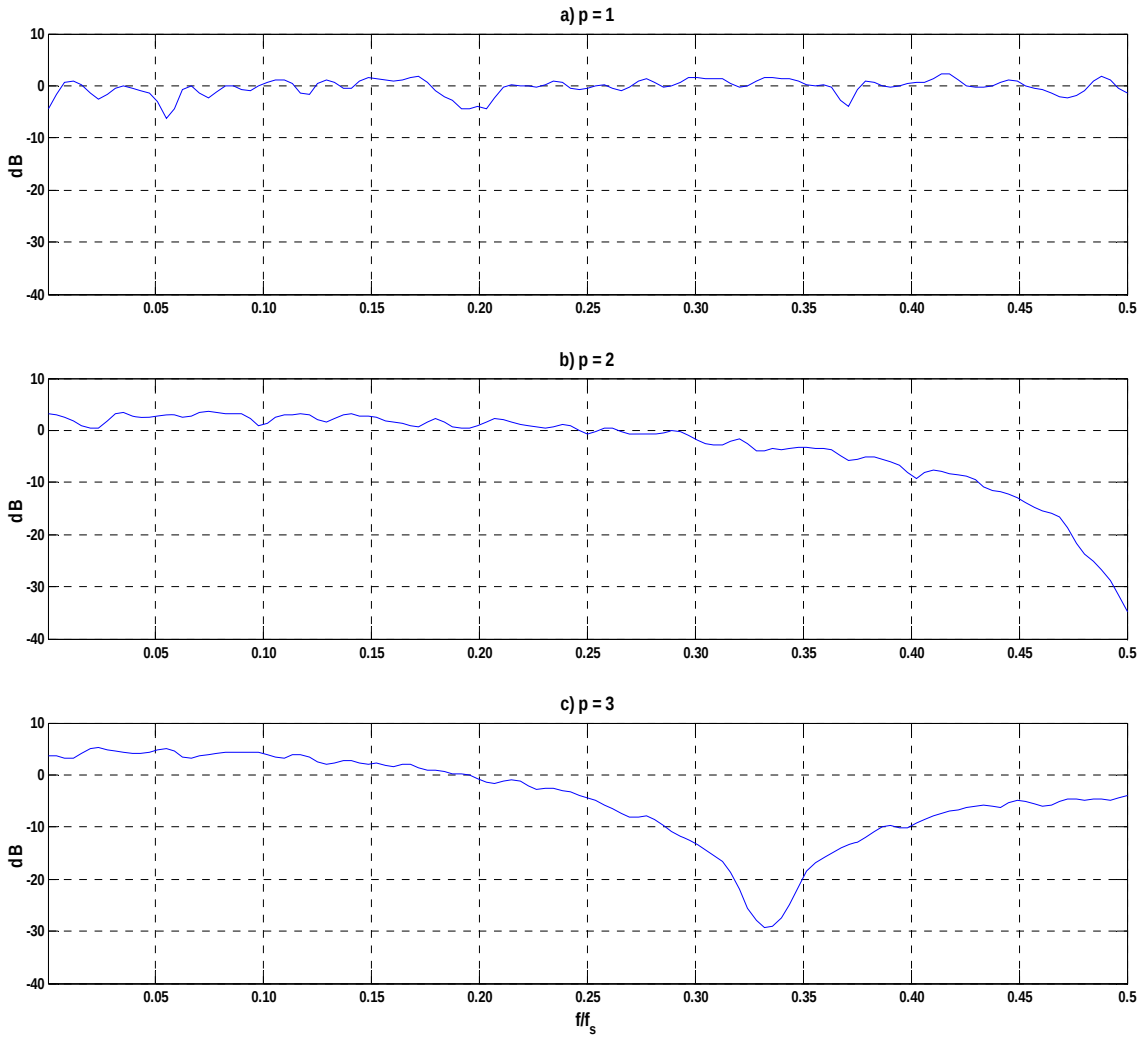
Spectral density of a P.R.B.S.

N=8

$p=1$
$f_{clock}=f_s$

$p=2$
$f_{clock}=f_s/2$

$p=3$
$f_{clock}=f_s/3$



Data pre-processing

The I/O data files should be centered

The use of non centered data files can cause serious errors

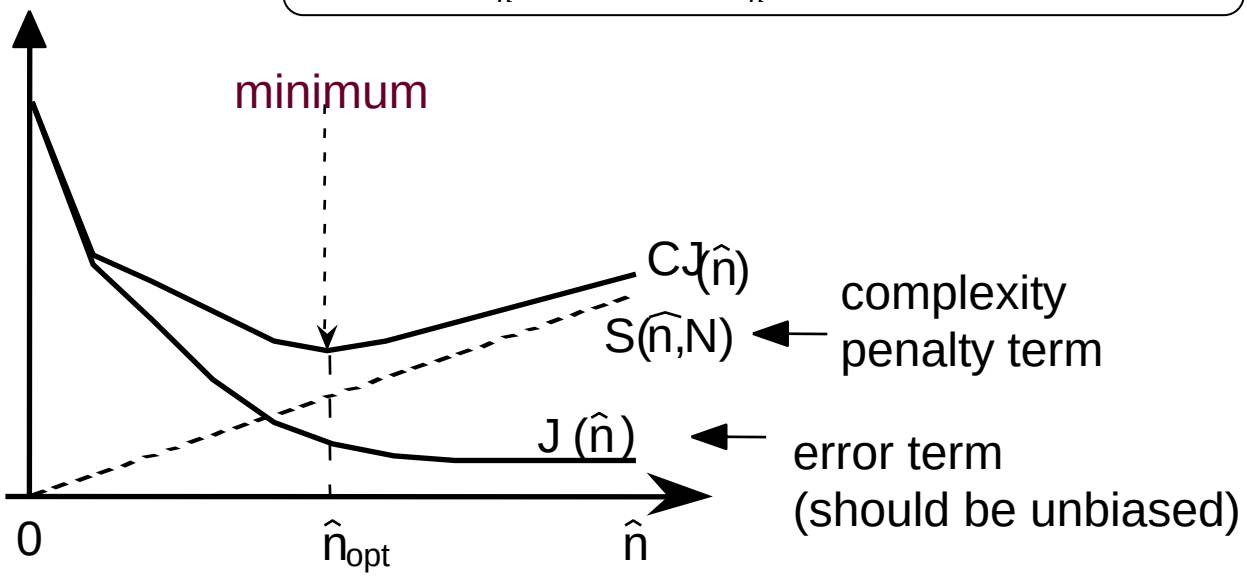
Complexity Estimation from I/O Data

Objective :

To get a good estimation of the model complexity (n_A, n_B, d) directly from noisy data

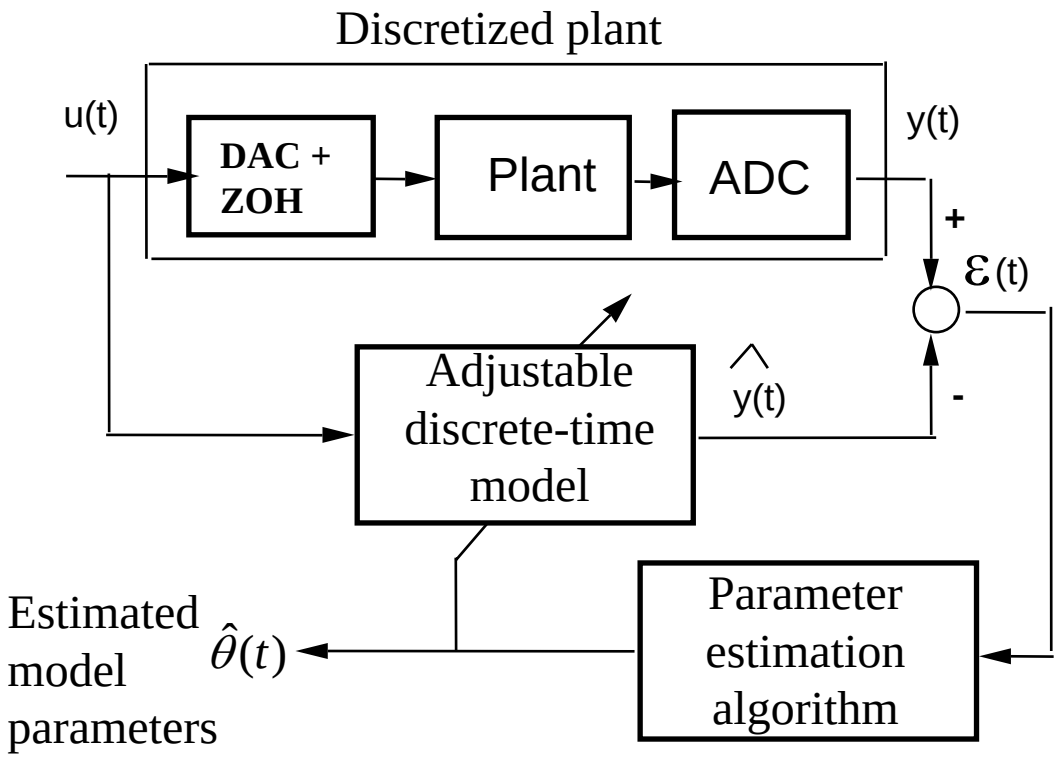
$$n = \max(n_A, n_B + d)$$

$$\hat{n}_{opt} = \min_{\hat{n}} CJ = \min_{\hat{n}} \left[J(\hat{n}) + S(\hat{n}, N) \right]$$



To get a good order estimation, J should tend to the value for noisy free data when $N \to \infty$ *(use of instrumental variables)*

Parameter Estimation



Parameter estimation alg.: → batch (non recursive)
→ recursive

It does not exists a unique algorithm providing good results in all the situations encountered in practice

Plant Model

$$G(q^{-1}) = \frac{q^{-d}B(q^{-1})}{A(q^{-1})} = \frac{q^{-d-1}B^*(q^{-1})}{A(q^{-1})}$$

$$A(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_{n_A} q^{-n_A} = 1 + q^{-1}A^*(q^{-1})$$

$$B(q^{-1}) = b_1 q^{-1} + \ldots + b_{n_B} q^{-n_B} = q^{-1}B^*(q^{-1})$$

$$y(t+1) = -A^*(q^{-1})y(t) + B^*(q^{-1})u(t-d) = \theta^T \psi(t)$$

$$\theta^T = \left[a_1, \ldots a_{n_A}, b_1, \ldots, b_{n_B}\right]$$

$$\psi(t)^T = \left[-y(t) \ldots -y(t-n_A+1), u(t-d) \ldots u(t-d-n_B+1)\right]$$

Recursive Parameter Estimation Methods

Plant Model

$$y(t+1) = -A^*(q^{-1})y(t) + B^*(q^{-1})u(t-d) = \theta^T \psi(t)$$

$$\theta - \textit{parameter vector}\,;\ \psi - \textit{measurement vector}$$

Estimated model

$$\hat{y}^\circ(t+1) = \hat{\theta}^T(t)\phi(t)$$

$$\hat{\theta} - \textit{estimated parameter vector}\,;\ \phi - \textit{observation vector}$$

Prediction error (a priori)

$$\varepsilon^0(t+1) = y(t+1) - \hat{\theta}^T(t)\Phi(t) = y(t+1) - \hat{y}^0(t+1)$$

Parameter adaptation algorithm (P.A.A.)

$$\hat{\theta}(t+1) = \hat{\theta}(t) + F(t+1)\Phi(t)\varepsilon^0(t+1)$$

$$F^{-1}(t+1) = \lambda_1(t)F^{-1}(t) + \lambda_2(t)\Phi(t)\Phi^T(t)$$

$$0 < \lambda_1(t) \le 1\,;\ 0 \le \lambda_2(t) < 2$$

$$\Phi(t) = f\left[\phi(t)\right] \qquad \textit{regressor vector}$$

Recursive Least Squares

Plant Model

$$y(t+1) = -A^*(q^{-1})y(t) + B^*(q^{-1})u(t-d) = \theta^T \psi(t)$$

$$\theta - \textit{parameter vector}\,;\ \psi - \textit{measurement vector}$$

Estimated model

$$\hat{y}^\circ(t+1) = \hat{\theta}^T(t)\phi(t)$$

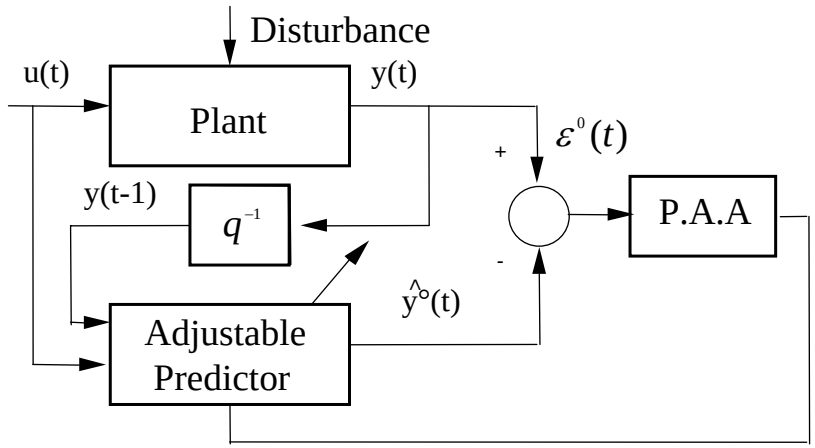
$$\hat{\theta} - \textit{estimated parameter vector}\,;\ \phi - \textit{observation vector}$$

$$\theta^T = \left[a_1, \ldots a_{n_A}, b_1, \ldots, b_{n_B}\right]$$

$$\phi(t)^T = \psi(t)^T \doteq \left[-y(t) \ldots -y(t-n_A+1), u(t-d) \ldots u(t-d-n_B+1)\right]$$

Regressor vector

$$\Phi(t) = \phi(t) = \psi(t)$$



See functions: ***rls.sci(.m)*** on the book website

Effect of stochastic disturbances (noise measurement)

•Identification algorithms operates at low signal to noise ratio (in order to disturb as little as possible a plant under operation)

•This often causes an error on estimated parameters called “bias”

•The reason for the existence of many identification algorithms is that *it does not exist an unique algorithm which gives unbiased estimates in all practical situations*

Non recursive least squares

$$\hat{\theta}(N) = F(N)\sum_{i=1}^{N} y(i)\phi(i-1) \qquad (*)$$

$$F(N)^{-1} = \sum_{i=1}^{N} \phi(i-1)\phi(i-1)^{T}$$

See functions: ***nrls.sci(.m)*** on the book website

Bias in Least Squares Parameter Estimation

In the presence of measurement noise the estimation of parameters is "biased" when using least squares algorithm

Plant output in the presence of noise: $y(t+1) = \theta^T \psi(t) + w(t+1) = \theta^T \phi(t) + w(t+1)$ **(**)**

Bias for the least squares algorithm (replace *y* in (*) by (**) *)*:

$$\hat{\theta}(N) = \theta + \left[\frac{1}{N}\sum_{t=1}^{N}\phi(t-1)\phi(t-1)^T\right]^{-1}\left[\frac{1}{N}\sum_{t=1}^{N}\phi(t-1)w(t)\right]$$

Condition for asymptotic unbiased estimation

$$\lim_{N\to\infty}\left[\frac{1}{N}\sum_{i=1}^{N-1}\phi(t-1)w(t)\right] = E\{\phi(t-1)w(t)\} = 0 \qquad \textbf{(***)}$$

regressor (observation) vector

noise

It is necessary that $\phi(t-1)$*(the regressor)* and *w(t) be uncorrelated*

For the least squares this implies : w(t) = e(t) (white noise).

For all the other cases the estimated parameters will be biased

Unbiased estimation in the presence of noise

Suppose : $\hat{\theta} = \theta$ *and we want that the algorithm leaves unchanged this value*

$$\hat{y}(t+1|\theta) = \theta^T \phi(t) \longrightarrow \varepsilon(t+1|\theta) = y(t+1) - \hat{y}(t+1|\theta) = w(t+1)$$

Necessary condition for unbiased estimation:

$$\textbf{(****)} \longrightarrow \lim_{N\to\infty}\left[\frac{1}{N}\sum_{i=1}^{N-1}\phi(t-1,\theta)\varepsilon(t,\theta)\right] = E\left\{\phi(t-1,\theta)\varepsilon(t,\theta)\right\} = 0$$

To eliminate the bias : $\boxed{E\left\{\phi(t)\varepsilon(t+1)\right\} = 0 \quad for \quad \hat{\theta} \equiv \theta}$ necessary condition

One modifies the LS algorithm in order to obtain:

$\varepsilon(t+1)$ as a white noise for: $\hat{\theta} = \theta$

or:

uncorrelated $\phi(t)$ and $\varepsilon(t+1)$ for: $\hat{\theta} = \theta$

Parameter Estimation Methods

I- *Based on the asymptotic whitening of the prediction error*
(Recursive Least Squares, Extended Least Squares, Recursive Max. Likelihood, O.E. with Extended Prediction Model)

II- *Based on the asymptotic decorrelation between the prediction error and the observation vector*
(Output Error, Instrumental Variable)

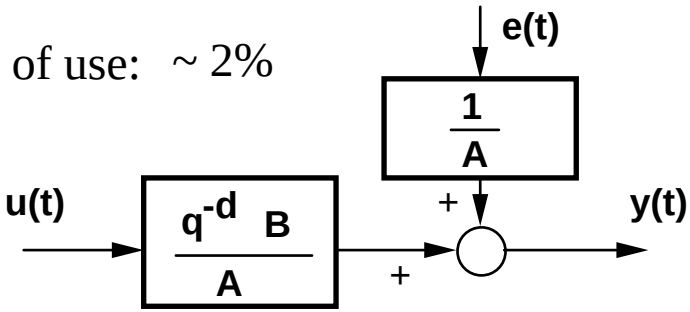
Remark:

One makes assumptions on the "noise"
and
One constructs the appropriate algorithm

«Plant + Noise » Models

$$S1: A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + e(t)$$

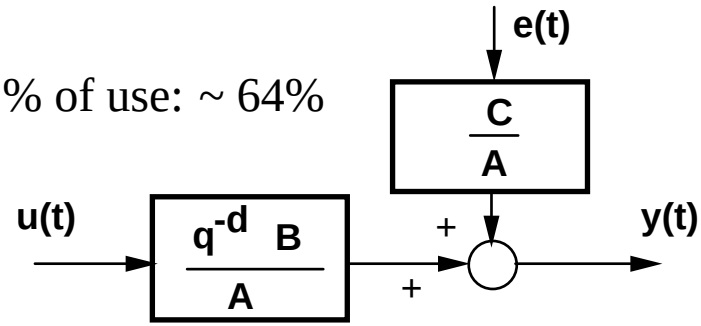
% of use: ~ 2%



(Recursive) Least Squares

$$S3: A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + C(q^{-1})e(t)$$

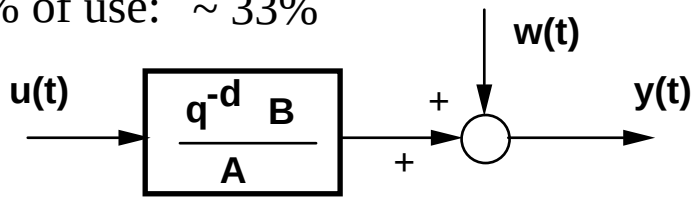
% of use: ~ 64%



Extended Least Squares
O.E. with Extended Prediction Model
(Recursive) Maximum Likelihood

$$S2: A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + A(q^{-1})w(t)$$

% of use: ~ 33%

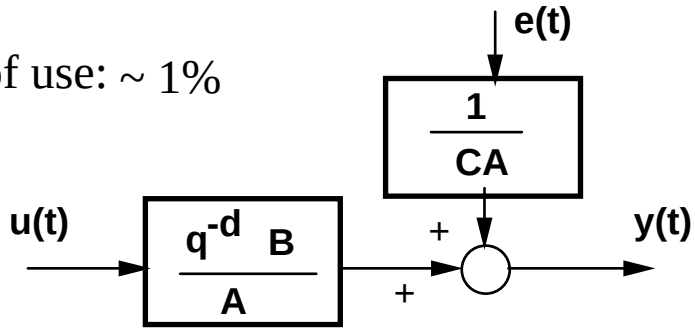


u and w are independent

Ouput Error(O.E.)
Instrumental Variable…

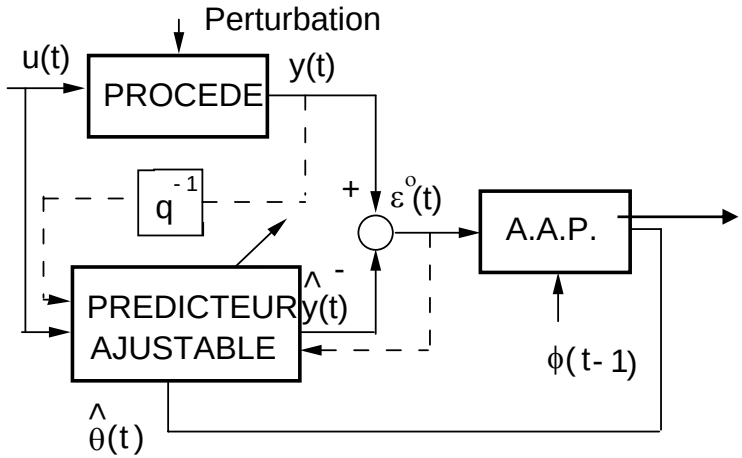
$$S4: A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + \left[1/C(q^{-1})\right]e(t)$$

% of use: ~ 1%



Generalized Least Squares

Structure of recursive identification methods



$$\hat{\theta}(t+1) = \hat{\theta}(t) + F(t+1)\Phi(t)\varepsilon^{0}(t+1)$$

$$F^{-1}(t+1) = \lambda_1(t)F(t) + \lambda_2\Phi(t)\Phi(t)^T$$

$$0 < \lambda_1(t) \le 1 \; ; \; 0 \le \lambda_2(t) < 2 \; ; \; F(0) > 0$$

(#)

Characteristic elements:

- predictor structure
- signals used for the observation (ϕ) and regressor vectors (Φ)
- dimension of the vector of estimated parameters $\hat{\theta}$ and Φ
- generation of the prediction error (ε)
- **all use the same parameter adaptation algorithm**

Types of identification methods:

I) *Based on the asymptotic whitening of the prediction error* (ε)

II) *Based on the asymptotic decorrelation of* Φ and ε

Extended Least Square (ELS)

Idea : *Identification of the plant model and of the disturbance (ARMAX) in order to obtain a white prediction error*

Plant + disturbance (ARMAX):

$$y(t+1) = -a_1 y(t) + b_1 u(t) + c_1 e(t) + e(t+1)$$

Optimal predictor (known parameters)

$$\hat{y}(t+1) = -a_1 y(t) + b_1 u(t) + c_1 e(t)$$

One replaces *e(t)* par *ε(t)*

Predicition error (known parameters) :

$$\varepsilon(t+1) = y(t+1) - \hat{y}(t+1) = e(t+1)$$

Adjustable predictor (unknown parameters):

$$\hat{y}^o(t+1) = -\hat{a}_1(t) y(t) + \hat{b}_1(t) u(t) + \hat{c}_1(t)\varepsilon(t) = \hat{\theta}(t)^T \phi(t) \qquad \text{(a priori)}$$

$$\hat{\theta}(t)^T = \left[\hat{a}_1(t), \hat{b}_1(t), \hat{c}_1(t)\right] \quad ; \quad \phi(t)^T = \left[-y(t), u(t), \varepsilon(t)\right]$$

$$\hat{y}(t+1) = -\hat{a}_1(t+1) y(t) + \hat{b}_1(t+1) u(t) + \hat{c}_1(t+1)\varepsilon(t) = \hat{\theta}(t+1)^T \phi(t)^T \quad \text{(a posteriori)}$$

Regressor: $\Phi(t) = \phi(t)$

Extended Least Square (ELS)

Prediction error (unknown parameters)

$$\varepsilon^o(t+1) = y(t+1) - \hat{y}^o(t+1) \quad ; \quad \varepsilon(t+1) = y(t+1) - \hat{y}(t+1)$$

PAA: One uses (#)

Rem.: *The size of* $\hat{\theta}$ *and* Φ *grows with respect to the least squares*

General case :

$$\hat{\theta}(t)^T = \left[\hat{a}_1(t)...\hat{a}_{n_A}, \hat{b}_1(t)...\hat{b}_{n_B}(t), \hat{c}_1(t)...\hat{c}_{n_C}(t)\right]$$

$$\Phi(t)^T = \left[-y(t)...-y(t-n_A+1), u(t-d)...u(t-d-n_B+1), \varepsilon(t)...\varepsilon(t-n_C+1)\right]$$

Properties:

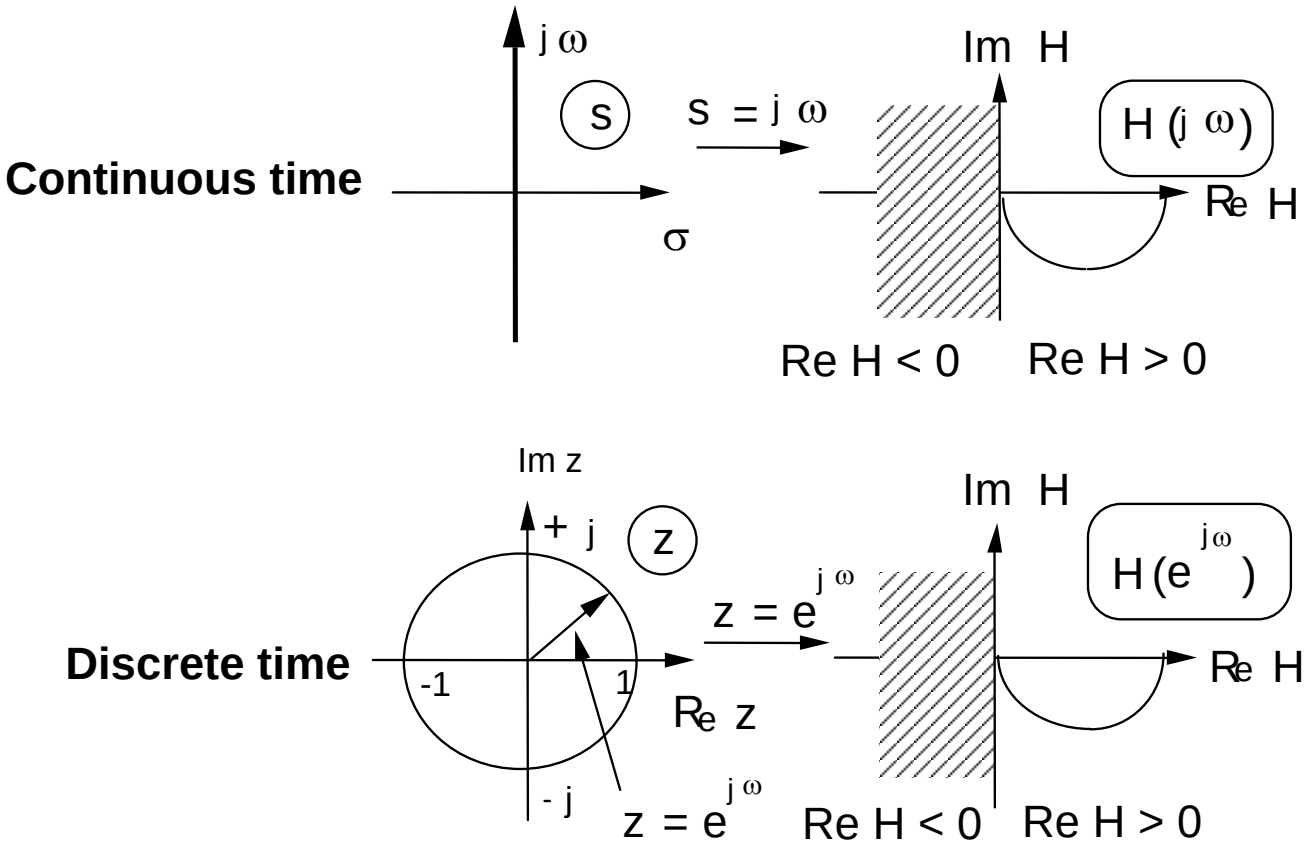
- *$\varepsilon(t)$ tends asymptotically towards a white noise (unbiased parameter estimation if in addition the input is persistently exciting)*
- *Sufficient convergence condition:* $\left(\frac{1}{C(z^{-1})} - \frac{\lambda_2}{2}\right)$; $2 > \lambda_2 \geq \max \lambda_2(t)$

Can explain the non convergence for some noise

Strictly positive real transfer function (SPR)

See function: ***rels.sci(.m)*** on the book web site

Strictly Positive Real Transfer Function (SRP)



- asymptotically stable
- $\mathrm{Re}\, H(e^{j\omega}) > 0$ *for all* $|e^{j\omega}| = 1, (0 < \omega < \pi)$ *(discrete case)*

An SPR transfer fct. introduces a phase lag less than 90° at all frequencies

Output error with extended prediction model (XOLOE)

An extension of output error methods for ARMAX models.

Can be viewed as a modification of RELS

Plant + disturbance (ARMAX):

$$y(t+1) = -a_1 y(t) + b_1 u(t) + c_1 e(t) + e(t+1)$$

Adjustable predictor for RELS:

$$\hat{y}^o(t+1) = -\hat{a}_1(t) y(t) + \hat{b}_1(t) u(t) + \hat{c}_1(t)\varepsilon(t) \pm \hat{a}_1(t)\hat{y}(t)$$

Adjustable predictor for XOLOE:

$$\hat{y}^o(t+1) = -\hat{a}_1(t)\hat{y}(t) + \hat{b}_1(t) u(t) + \hat{h}_1(t)\varepsilon(t) = \hat{\theta}(t)^T \phi(t) \quad ; \quad \hat{h}_1(t) = \hat{c}_1(t) - \hat{a}_1(t)$$

$$\hat{\theta}(t)^T = \left[\hat{a}_1(t), \hat{b}_1(t), \hat{h}_1(t)\right] \quad ; \quad \Phi(t) = \phi(t)^T = \left[-\hat{y}(t), u(t), \varepsilon(t)\right]$$

Instead of *y(t)* in RELS

Prediction error (a priori): $\varepsilon^o(t+1) = y(t+1) - \hat{y}^o(t+1)$

PAA: One uses (#)

See function: ***xoloe.sci(.m)*** on the book web site

Recursive output error (OLOE)

Plant Model

$$y(t+1) = -A^*(q^{-1})y(t) + B^*(q^{-1})u(t-d) = \theta^T \psi(t)$$

$$\theta - \text{parameter vector}; \psi - \text{measurement vector}$$

Estimated model

$$\hat{y}^0(t+1) = -\hat{A}^*(t,q^{-1})\hat{y}(t) + \hat{B}^*(t,q^{-1})u(t-d) = \hat{\theta}^T(t)\phi(t) \quad \textit{a priori}$$

$$\hat{y}(t+1) = \hat{\theta}^T(t+1)\phi(t) \quad \textit{a posteriori}$$

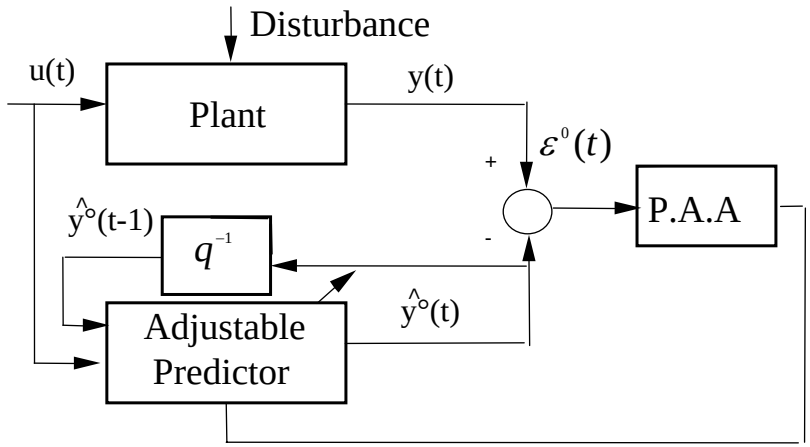
$$\hat{\theta} - \text{estimated parameter vector}; \phi - \text{observation vector}$$

$$\theta^T = \left[a_1,...a_{n_A}, b_1,...,b_{n_B}\right] \qquad \hat{\theta}^T = \left[\hat{a}_1,...\hat{a}_{n_A}, \hat{b}_1,...,\hat{b}_{n_B}\right]$$

$$\phi(t)^T = \left[-\hat{y}(t)...-\hat{y}(t-n_A+1), u(t-d)...u(t-d-n_B+1)\right]$$

Regressor vector

$$\Phi(t) = \phi(t)$$



Recursive output error

Prediction error (unknown parameters)

$$\varepsilon^o(t+1) = y(t+1) - \hat{y}^o(t+1) \quad ; \quad \varepsilon(t+1) = y(t+1) - \hat{y}(t+1)$$

PAA: One uses (#)

Properties:

- *Unbiased estimation of the plant parameters without identifiyng the noise model (useful for non gaussian noise)*

- *Sufficient convergence condition:* $\left(\frac{1}{A(z^{-1})} - \frac{\lambda_2}{2} \right)$; $2 > \lambda_2 \geq \max \lambda_2(t)$

Strictly positive real transfer function (SPR)

Remark:

The SPR condition can be relaxed by filtering the prediction error or the regressor

See function ***oloe.sci (.m)*** on the web site of the book

Output error with filtered observations (OEFO)

Adjustable predictor (output error):

$$\hat{y}^o(t+1) = \hat{\theta}(t)^T \phi(t)$$

$$\hat{\theta}^T = \left[\hat{a}_1,...\hat{a}_{n_A},\hat{b}_1,...,\hat{b}_{n_B}\right] \qquad \phi(t)^T = \left[-\hat{y}(t)...-\hat{y}(t-n_A+1),u(t-d)...u(t-d-n_B+1)\right]$$

$$\hat{y}(t+1) = \hat{\theta}(t+1)^T \phi(t) \Rightarrow \hat{y}(t) = \hat{\theta}(t)^T \phi(t-1)$$

Prediction error:

$$\varepsilon^o(t+1) = y(t+1) - \hat{y}^o(t+1) \qquad ; \qquad \varepsilon(t+1) = y(t+1) - \hat{y}(t+1)$$

Filtering the observations:

Filter: $L(q^{-1}) = \hat{A}(q^{-1})$ ← **An estimation of the polynomial A(q-1)**

Regressor: $\Phi(t) = \phi_f(t) = \dfrac{1}{\hat{A}(q^{-1})}\phi(t)$

PAA: One uses (#) with : $\Phi(t) = \phi_f(t)$

• *Sufficient convergence condition :* $\left(\dfrac{\hat{A}(z^{-1})}{A(z^{-1})} - \dfrac{\lambda_2}{2}\right)$; $2 > \lambda_2 \geq \max \lambda_2(t)$

Strictly positive real transfer function (SPR)

See function ***foloe.sci(.m)*** on the book web site

Output error with adaptive filtered observations (OEAFO)

Uses an adaptive filter on the observations instead of a fixed filter
(takes advantage of the improvement of the estimation of A as times goes)

Filtering the observations:

Filtre: $L(t,q^{-1}) = \hat{A}(t,q^{-1})$ ← **Estimation of polynomial $A(q^{-1})$ provided by the algorithm itself**

$$\Phi(t) = \phi_f(t) = \frac{1}{\hat{A}(t,q^{-1})}\phi(t)$$

Initialization:

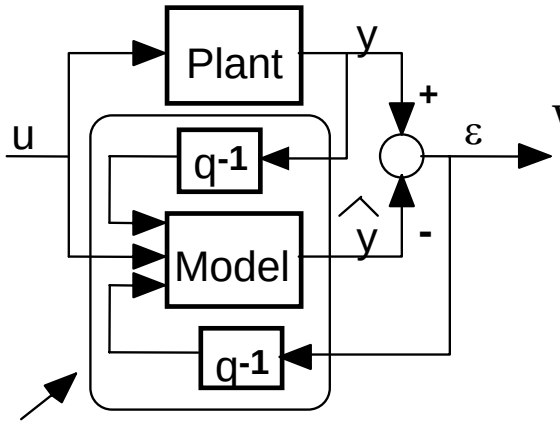
$\hat{A}(0,q^{-1}) = \hat{A}_0(q^{-1})$ or: $\hat{A}(0,q^{-1}) = 1$ (simpler and more efficient)

Remove in most of the case the problems related to the SPR condition

See function: ***afoloe.sci(.m)*** on the book web site

Validation of Identified Models

Statistical Validation



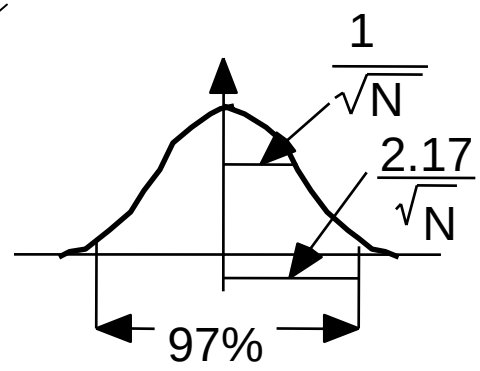
ARMAX (ARARX) predictor

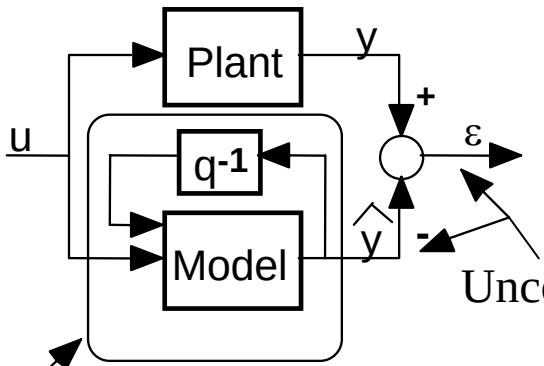
Whiteness Test

$$\left|RN(i)\right| \le \frac{2.17}{\sqrt{N}};\ i \ge 1$$

Normalized autocorrelations or crosscorelations

number of data





Output Error Predictor

Uncorrelation Test

$$N = 256 \rightarrow \left|RN(i)\right| \le 0.136$$

pratical value : $\left|RN(i)\right| \le 0.15$

«Whiteness » test

$\{\varepsilon(t)\}$: centered sequence of residual prediction errors

One computes:

$$R(0) = \frac{1}{N}\sum_{t=1}^{N}\varepsilon^2(t) \quad ; \quad RN(0) = \frac{R(0)}{R(0)} = 1$$

$$R(i) = \frac{1}{N}\sum_{t=1}^{N}\varepsilon(t)\varepsilon(t-i) \quad ; \quad RN(i) = \frac{R(i)}{R(0)} \quad ; \quad i = 1,2,3...i_{\max} \; ;$$

Theoretical values: $RN(i) = 0;\ i = 1, 2 \ldots i_{max}$

Real situation:

- Finite number of data
- Residual structural errors (orders, nonlinearities, noise)
- Objective: to obtain « good » simple models

Validation criterion (N = number of data):

$$\left|RN(i)\right| \le \frac{2.17}{\sqrt{N}} \quad ; \quad i \ge 1$$

or: $\left|RN(i)\right| \le 0.15 \; ; \; i = 1,...,i_{\max}$

« Uncorrelation » test

$\{\varepsilon(t)\}, \{\hat{y}(t)\}$: centered sequences of residual prediction errors and predictions

Output error predictor: $\hat{A}(q^{-1})\hat{y}(t) = q^{-d}\hat{B}(q^{-1})u(t)$

One computes:

$$R(i) = \frac{1}{N}\sum_{t=1}^{N}\varepsilon(t)\hat{y}(t-i) \;\; ; \;\; i = 0,1,2,...,i_{\max} \qquad ; \qquad i_{\max} = \max(n_A, n_B + d)$$

$$RN(i) = \frac{R(i)}{\left[\left(\frac{1}{N}\sum_{t=1}^{N}\hat{y}^2(t)\right)\left(\frac{1}{N}\sum_{t=1}^{N}\varepsilon^2(t)\right)\right]^{1/2}} \;\; ; \;\; i = 0,1,2,...,i_{\max}$$

Remark: $RN(0) \neq 1$

Theoretical values: $RN(i) = 0;\ i = 1, 2 \ldots i_{max}$

Real situation:

- Finite number of data
- Residual structural errors (orders, nonlinearities, noise)
- Objective: to obtain « good » simple models

Validation criterion (N = number of data):

$$|RN(i)| \leq \frac{2.17}{\sqrt{N}} \;\; ; \;\; i \geq 1$$

or: $|RN(i)| \leq 0.15 \; ; \; i = 1,...,i_{\max}$

Matlab/Scilab routines for
Open Loop System identification

To be downloaded from the web site:
http//:landau-bookic.lag.ensieg.inpg.fr

- function files(.m and .sci)
- data(.mat)

List of functions for open loop identification

Scilab Functions	*Matlab functions*	*Description*
estorderls.sci	estorderls.m	Order estimation with the least squares criterion
estorderiv.sci	estorderiv.m	Order estimation with the instrumental variable criterion
nrls.sci	nrls .m	Non recursive Least squares
rls.sci	rls.m	Recursive least squares
rels.sci	rels.m	Recursive extended least squares
oloe.sci	oloe.sci	Output Error(recursive)
foloe.sci	foloe.m	Output error with filtered observations
afoloe.sci	afoloe.m	Output error with adaptive filtered observations
xoloe.sci	xoloe.m	Output error with extended prediction model
	olvalid.m	Validation of plant models identified in open loop

Estorderiv – Order estimation using Instrumental Variable Method

Order of a discrete time system: $n = \max(n_A, n_B + d)$

>> help estorderiv

Upper bound for the order

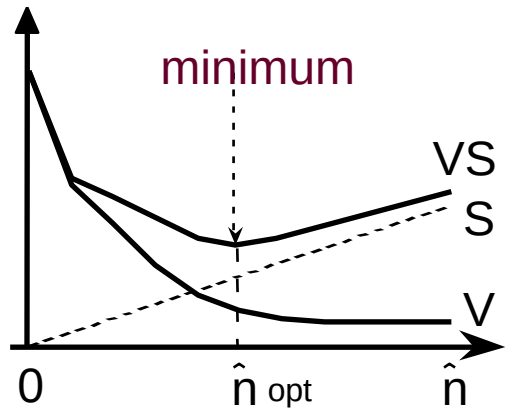
[V,S,VS]=estorderiv(y,u,nmax)

Measured output

Excitation

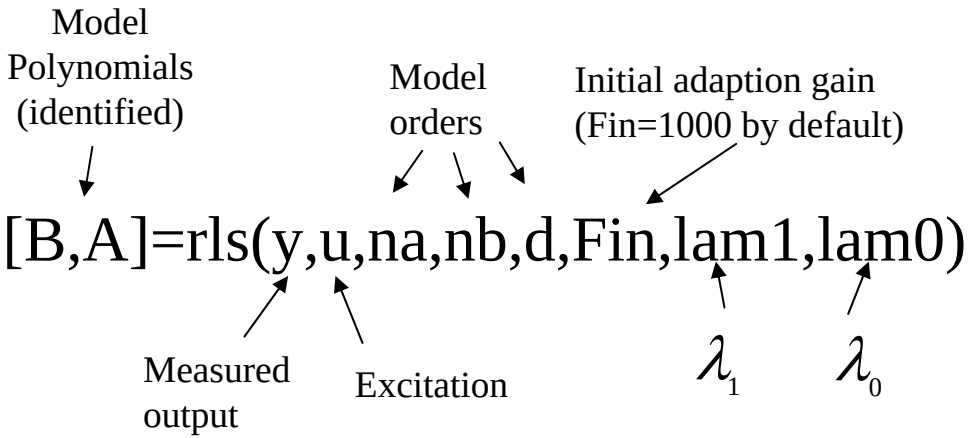
V - vector of the error criterion V=[V(0) V(1) V(2) ….]^T,where V(0)=1
S – vector of penalty coefficients. Model complexity n is penalized
VS – normalized vector of penalized criterion VS=V+S

n corresponds to the minimum of VS(n)
(attn. for the n axis : should start with 0)



RLS –recursive least squares identification function

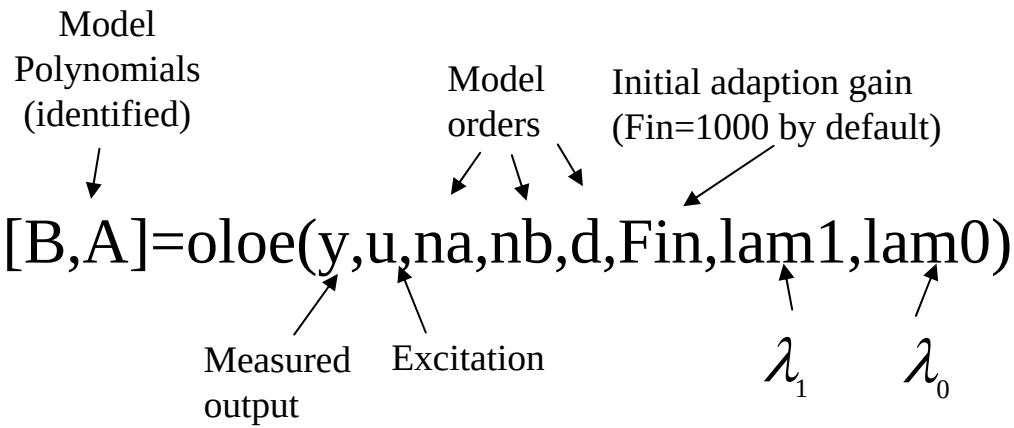
>> help rls



lam1=1;lam0=1 : decreasing gain (default algorithm)
0.95<lam1<1; lam0=1 : decreasing gain with fixed forgetting factor
0.95<lam1,lam0<1 : decreasing gain with variable forgetting factor (typical value :0.97)

OLOE – open loop output error identification function

>> help oloe



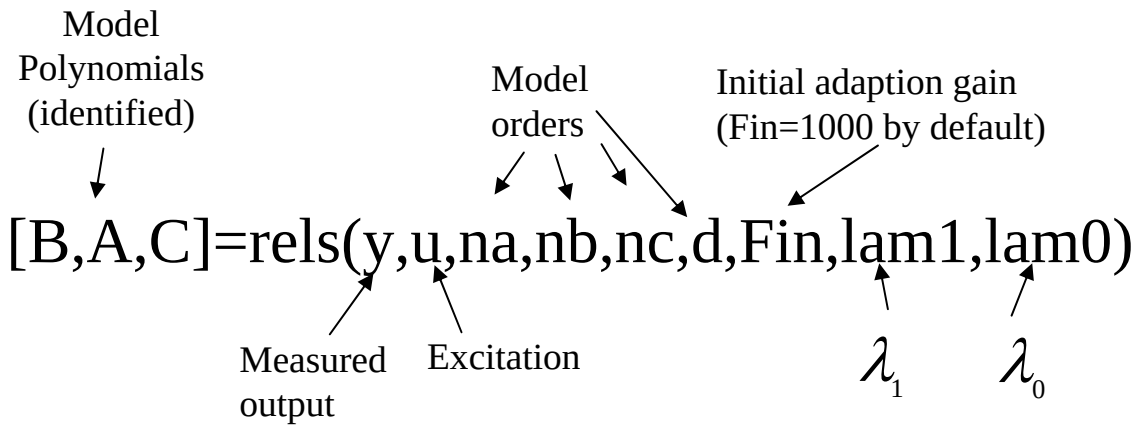
lam1=1;lam0=1 : decreasing gain (default algorithm)

0.95<lam1<1; lam0=1 : decreasing gain with fixed forgetting factor

0.95<lam1,lam0<1 : decreasing gain with variable forgetting factor (typical value :0.97)

RELS – Extended least squares identification function
XOLOE- Output error with extended prediction model

>> help rels



>> help xoloe

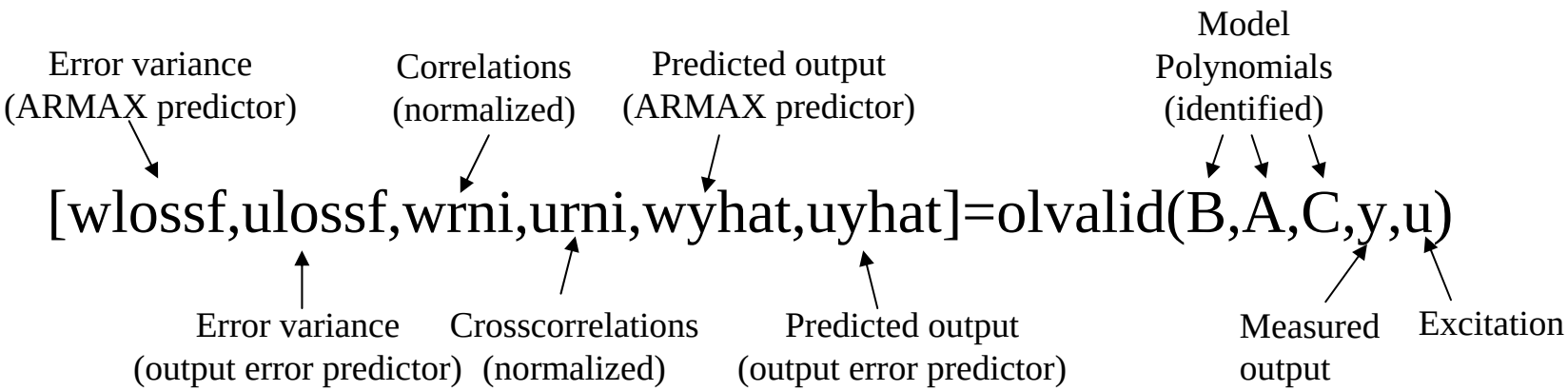
[B,A,C]=xoloe(y,u,na,nb,nc,d,Fin,lam1,lam0)

lam1=1;lam0=1 : decreasing gain (default algorithm)
0.95<lam1<1; lam0=1 : decreasing gain with fixed forgetting factor
0.95<lam1,lam0<1 : decreasing gain with variable forgetting factor (typical value :0.97)

OLVALID – Validation of plant models identified in open loop

Performs “whiteness test”(w) and “uncorrelation test”(u)

>> help olvalid



Attention: If polynomial C is not available enter [1]

Data files for open loop identification

- T0.mat: simulated example without noise

$$d = 1;\ A = 1 - 1.5q^{-1} + 0.7q^{-1};\ B = 1q^{-1} + 0.5q^{-1}$$

- T1.mat: same simulated example with noise

- poulbo1c.mat: flexible transmission

$$d = 2;\ n_A = 4;\ n_B = 2$$

- aeroc.mat : air heater

- mot3c.mat: D.C. motor

- rob2.mat: flexible robot arm with two vibration modes

Open loop system identification - references

More details can be found in :

I.D. Landau, G. Zito, *Digital Control Systems – Design, identification and implementation* , Springer, London, 2005

I.D. Landau,R. Lozano,M.M'Saad, *Adaptive Control*, Springer, London, 1997

I.D. Landau, *Commande des systèmes – conception, identification, mise en œuvre*, Hermes, Paris, 2002.

and

http://landau-bookic.lag.ensieg.inpg.fr

Free routines (matlab, scilab) and slides can be downloaded

General references:

T. Söderstrom, P. Stoica : *System identification*, Prentice Hall, UK, 1989

L. Ljung : *System identification – Theory for the user*, Prentice Hall, NJ, 2nd edition, 2002