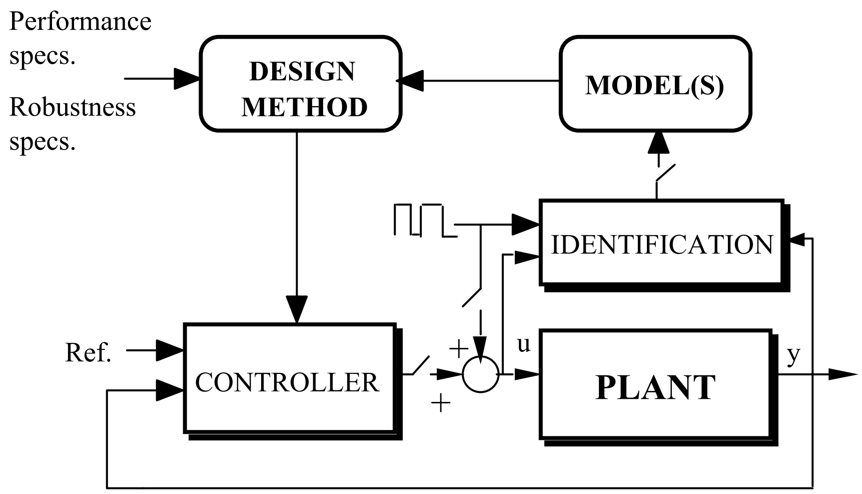
Chapter V

System Identification : The Bases

Version 1/25.11.2005

Chapter 5 System Identification: The Bases

Controller Design



In order to design and tune a controller correctly, one needs:

1. To specify the desired control loop performance and robustness
2. To know the dynamic model of the plant to be controlled (also known as the *control model*)
1. To possess of a suitable controller design method

Identification = determination of the *dynamic model*of a system from input/output measurements

Plant Models

Knowledge models

Use: Simulation et and plant design

Dynamic control models (input/output)

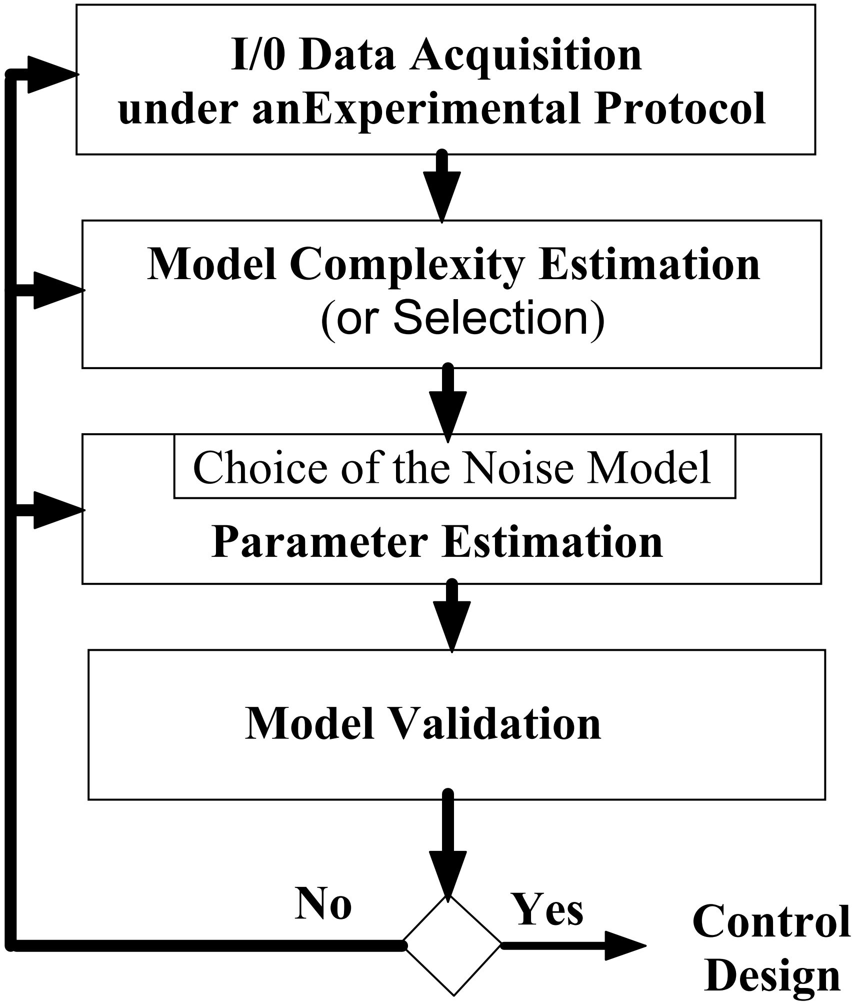
Use : design of the controllers

Dynamic models
- *non parametric* (ex: time or frequency response)
- *parametric* (ex: transfer function)

Parametric models
- *continuous*
- *Discrete time*

We will be interested in the identification of parametric discrete time dynamic models

System identification methodology



I/0 Data Acquisition under anExperimental Protocol

Model Complexity Estimation (or Selection)

Choice of the Noise Model

Parameter Estimation

Model Validation

No

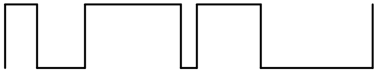
Yes

Control Design

System identification should be viewed as an iterative procedure whose objective is to find a valid «model »

Identification using a digital computer

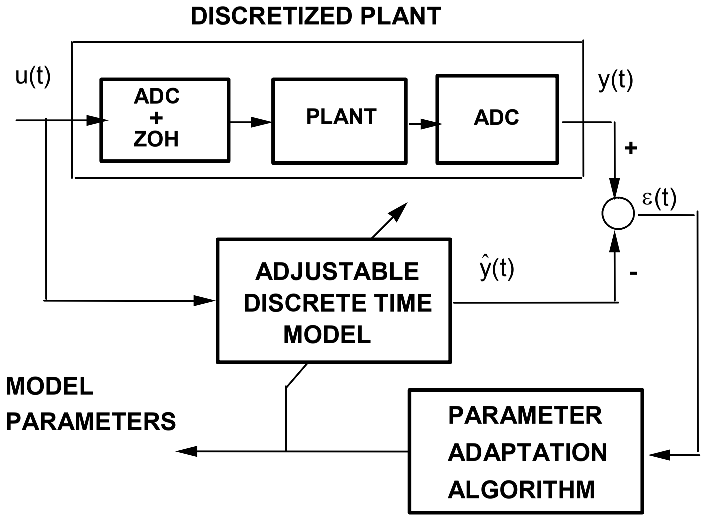
1) Data acquisition



input: low magnitude pseudo random binary sequence

2)Estimation of the model complexity (algorithms)

3)Estimation of the model parameters (algorithms)



4) Validation of the identified model

Statistical tests on ε(t) and ^y(t)

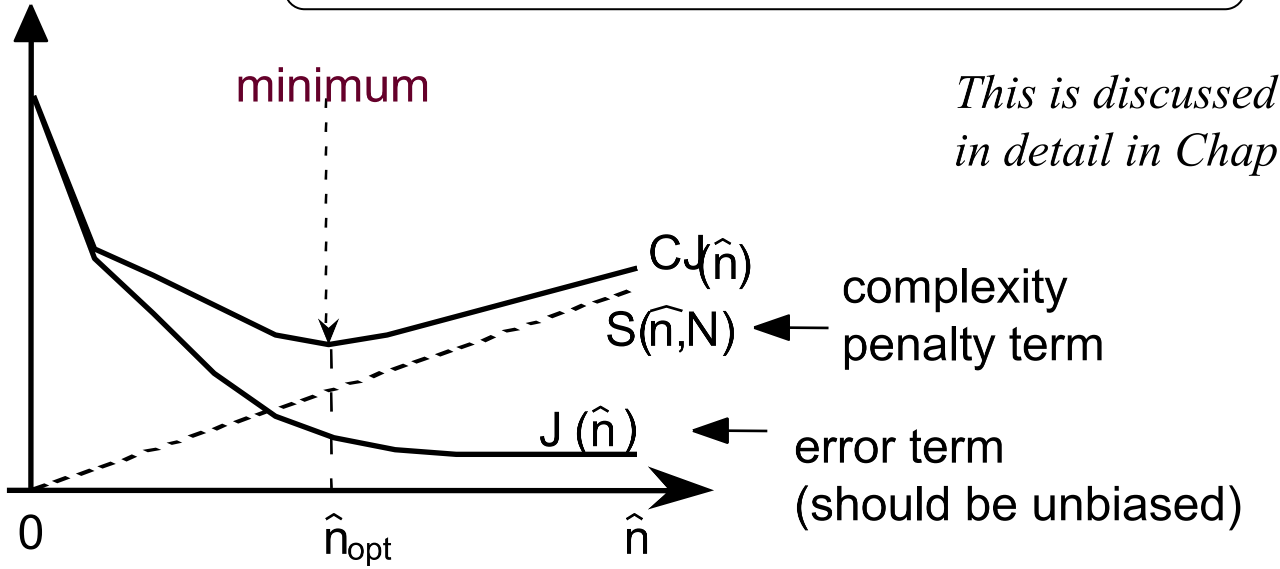
Complexity Estimation from I/O Data

Objective :

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

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

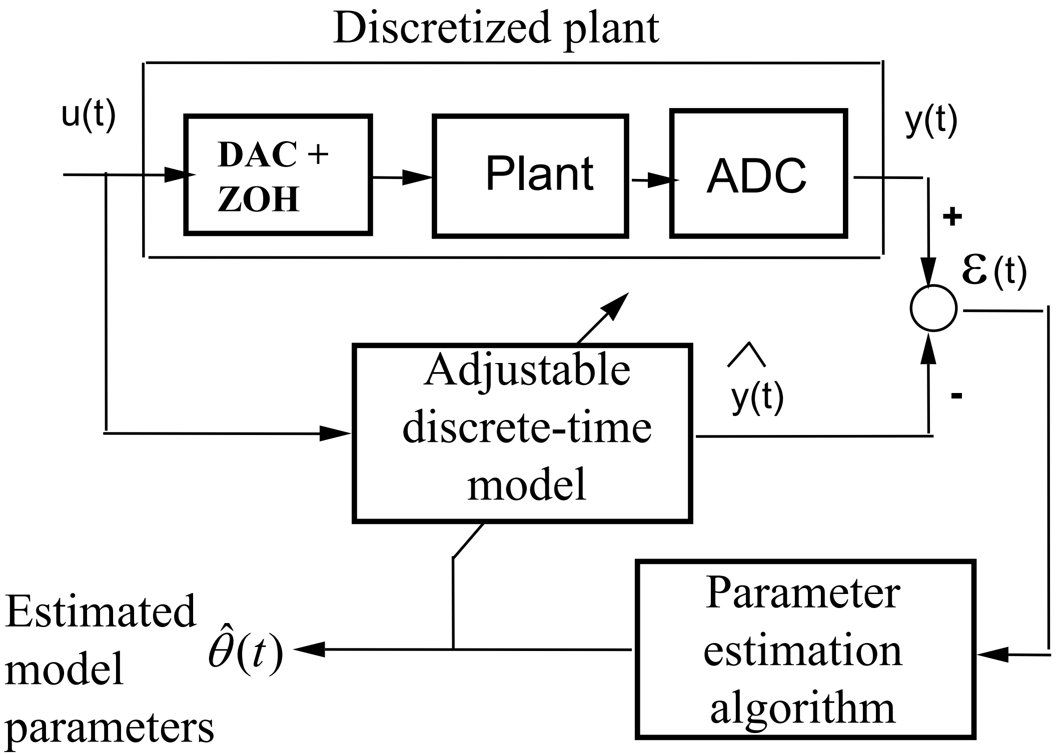
$$\boxed{\hat{n}_{opt} = \min_{\hat{n}} CJ = \min_{\hat{n}}\left[J(\hat{n}) + S(\hat{n}, N)\right]}$$



This is discussed in detail in Chapter 6

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

Parameter Estimation



Discretized plant

u(t)

DAC +
ZOH

Plant

ADC

y(t)

+

$\varepsilon(t)$

$\hat{y}(t)$

-

Adjustable
discrete-time
model

Parameter
estimation
algorithm

Estimated
model
parameters

$\hat{\theta}(t)$

Parameter estimation alg.: → batch (non recursive)

→ recursive

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

Caracteristics of identification by means of a digital computer

- Use low magnitude testing signals
- Excellent precision for the identified parameters
- Objective criteria for validation
- Tracking of the variations of the system parameters in real time allowing retuning of controllers during operation
- Identification of disturbances models

Other applications of system identification:

- Modeling of the transducer noises in view of their elimination
- Detection and measurement of vibration frequencies
- Spectral analysis of the signals

Parametric identification (estimation)

Non recursive parameter estimation:

Process as a one block the input/output data files obtained over a time horizon
Parameters are not estimated during data acquisition or file reading

Recursive parameter estimation:

One pair of input/output data is processed at each sampling time (during data acquisition (real time) or when reading the data file).
Estimation of parameters is provided at each sampling instant

Advantages of recursive parameter estimation

- Obtaining an estimated model as the system evolves
- Considerable data compression, since the recursive algorithms process at each instant only one input/output pair instead of the whole input/output data set
- Much lower requirements in terms of memory and CPU power
- Easy implementation on microcomputers
- Possibility to implement real-time identification systems
- Possibility to track the parameters of time variable systems

Recursive paramter identification (estimation)
Parametric adaptation algorithm (PAA)

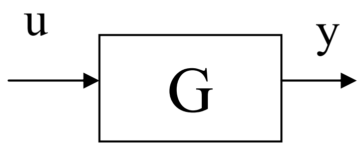
Parameter vector = contains all the parameters of the model

$$\begin{bmatrix}\text{New parameters}\\ \text{estimation}\\ \text{(vector)}\end{bmatrix} = \begin{bmatrix}\text{Old parameters}\\ \text{estimation}\\ \text{(vector)}\end{bmatrix} +$$

$$\begin{bmatrix}\text{Adaptation}\\ \text{Gain}\\ \text{(matrix)}\end{bmatrix} \times \underbrace{\begin{bmatrix}\text{Measurement}\\ \text{function}\\ \text{(vector)}\end{bmatrix}}_{\text{Observation vector}} \times \begin{bmatrix}\text{Error prediction}\\ \text{function}\\ \text{(scalar)}\end{bmatrix}$$

Plant Model

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



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

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

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

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

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

Algorithms for parameter estimation

Discrete time plant model (unknown parameters))

$$y(t+1) = -a_1 y(t) + b_1(t)u(t) = \theta^T \phi(t)$$

Measurement vector

$$\theta^T = [a_1, b_1] \longleftarrow \text{Parameter vector} \quad ; \quad \phi(t)^T = [-y(t), u(t)]$$

Adjustable prediction model (à priori)

$$\hat{y}^o(t+1) = \hat{y}(t+1|\hat{\theta}(t)) = -\hat{a}_1(t)y(t) + \hat{b}_1(t)u(t) = \hat{\theta}(t)^T \phi(t)$$

$$\theta(t)^T = [\hat{a}_1(t), \hat{b}_1(t)] \longleftarrow \text{Vector of adjustable parameters}$$

Prediction error (à priori)

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

Criterion to be minimized (objective):

$$J(t+1) = \left[\varepsilon^o(t+1)\right]^2 = \left[\varepsilon^o(t+1, \hat{\theta}(t))\right]^2$$

?

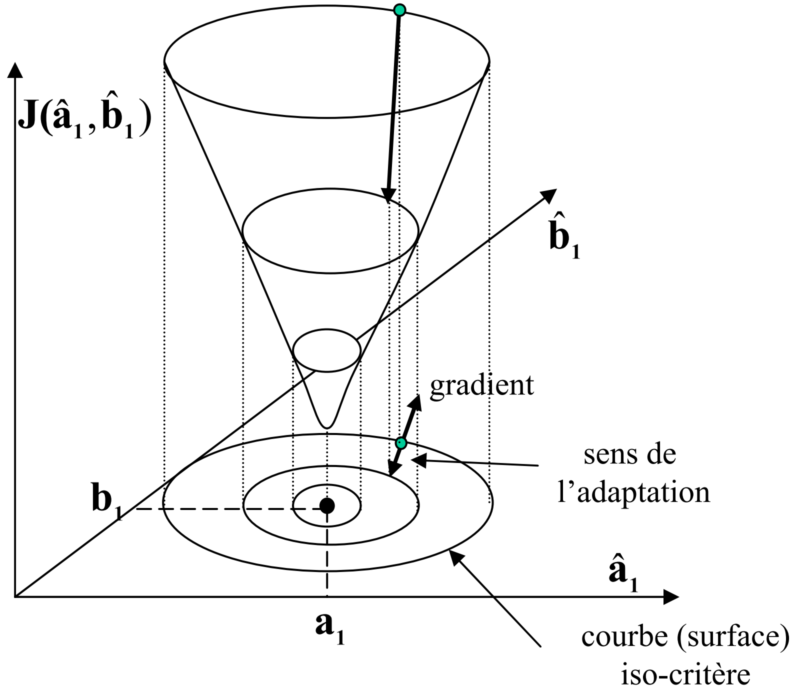
Parameter adaptation algorithm

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \Delta\hat{\theta}(t+1) = \hat{\theta}(t) + f\left(\hat{\theta}(t), \phi(t), \varepsilon^o(t+1)\right)$$

PAA – Gradient algorithm

Criterion to be minimized (objective):

$$\min_{\hat{\theta}(t)} J(t+1) = \left[\varepsilon^{o}(t+1)\right]^{2}$$



gradient of the criterion

$$\hat{\theta}(t+1) = \hat{\theta}(t) - F\frac{\partial J(t+1)}{\partial \hat{\theta}(t)}$$

$$F = \alpha I \quad (\alpha > 0)$$ **(*I* = unit matrix)**

$$\frac{1}{2}\frac{\partial J(t+1)}{\partial \hat{\theta}(t)} = \frac{\partial \varepsilon^{o}(t+1)}{\partial \hat{\theta}(t)}\varepsilon^{o}(t+1)$$

$$\varepsilon^{o}(t+1) = y(t+1) - \hat{y}^{o}(t+1) = y(t+1) - \hat{\theta}(t)^{T}\phi(t) \quad \Rightarrow \quad \frac{\partial \varepsilon^{o}(t+1)}{\partial \hat{\theta}(t)} = -\phi(t)$$

$$\boxed{\hat{\theta}(t+1) = \hat{\theta}(t) + F\phi(t)\varepsilon^{o}(t+1)}$$

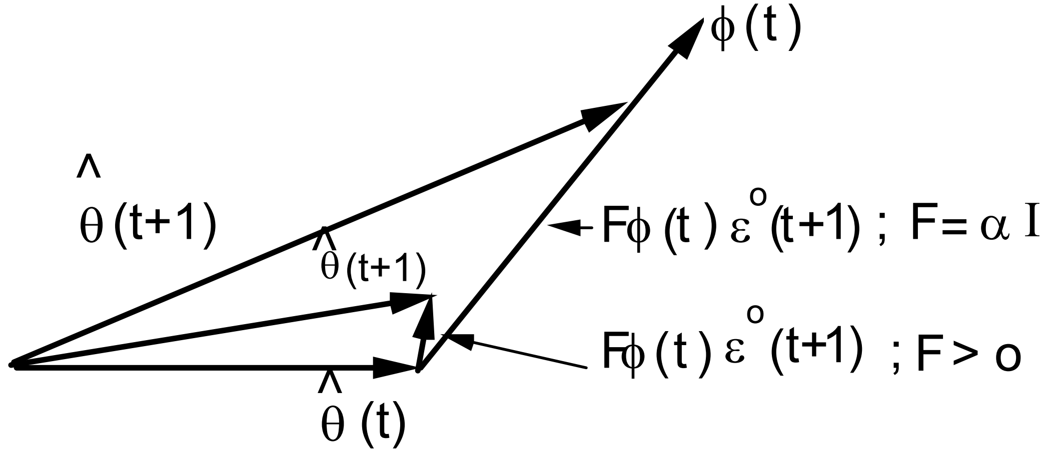
PAA – Gradient algorithm

$$\boxed{\hat{\theta}(t+1) = \hat{\theta}(t) + F\phi(t)\varepsilon^{o}(t+1)}$$

Adaptation gain

$$\begin{cases} F = \alpha I & (\alpha > 0) \\ F > 0 & \textit{Positive definite matrix} \end{cases}$$

Geometrical interpretation →



Attention: Instability risk if $F(\alpha)$ is large !!

(see book pg. 213 – 214 for details)

PAA – Improved gradient algorithm

a posteriori output of the adjustable predictor

$$\hat{y}(t+1) = \hat{y}(t+1|\hat{\theta}(t+1)) = -\hat{a}_1(t+1)y(t) + \hat{b}_1(t+1)u(t) = \hat{\theta}(t+1)^T \phi(t)$$

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

Criterion to be minimized (objective): $\min\limits_{\hat{\theta}(t+1)} J(t+1) = \left[\varepsilon(t+1)\right]^2$

Gradient technique: $\hat{\theta}(t+1) = \hat{\theta}(t) - F\dfrac{\partial J(t+1)}{\partial \hat{\theta}(t)}$

$$\frac{1}{2}\frac{\partial J(t+1)}{\partial \hat{\theta}(t+1)} = \frac{\partial \varepsilon(t+1)}{\partial \hat{\theta}(t+1)}\varepsilon(t+1)$$

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

$$\hat{\theta}(t+1) = \hat{\theta}(t) + F\phi(t)\varepsilon(t+1)$$

For the implementation one should express: $\varepsilon(t+1) = f(\hat{\theta}(t), \phi(t), \varepsilon^0(t+1))$

PAA – Improved gradient algorithm

$$\varepsilon(t+1) = \underbrace{y(t+1)-\hat{\theta}(t)^T\phi(t)}_{\varepsilon^0(t+1)} - \left[\hat{\theta}(t+1)-\hat{\theta}(t)\right]^T\phi(t)$$

$$\hat{\theta}(t+1) = \hat{\theta}(t) + F\phi(t)\varepsilon(t+1) \longrightarrow \hat{\theta}(t+1)-\hat{\theta}(t) = F\phi(t)\varepsilon(t+1)$$

$$\varepsilon(t+1) = \varepsilon^0(t+1) - \phi(t)F\phi(t)\varepsilon(t+1) \longrightarrow \varepsilon(t+1) = \frac{\varepsilon^o(t+1)}{1+\phi(t)^T F\phi(t)}$$

$$\boxed{\hat{\theta}(t+1) = \hat{\theta}(t) + \frac{F\phi(t)\varepsilon^o(t+1)}{1+\phi(t)^T F\phi(t)}}$$

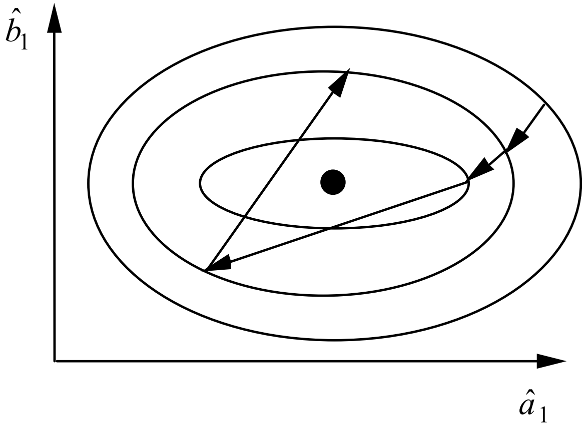
Stable for any *F* > *0*

Implementation:

1. Before *t+1* one has: $u(t), u(t-1), .. y(t), y(t-1), \phi(t), \hat{\theta}(t), F$
2. Before *t+1* one computes : $F\phi(t)/(1+\phi(t)^T F\phi(t)),\ \hat{y}^0(t+1)$
3. At *t+1* acquisition of *y(t+1)* and *u(t+1) is sent*
4. Running of the AAP
   (computation of : $\varepsilon^0(t+1), \hat{\theta}(t+1)$)

Least squares algorithm

Minimization at each step of $\varepsilon^2(t+1)$ does not impliy necessarily la minimisation de $\sum \varepsilon^2(i+1)$ over a horizon of t steps.



To get a good result one needs (intuitively):

- high adaptation gain at the beginning
- low adaptation gain at the end

The least squares algorithm has these properties

Discrete time plant model (unknown parameters))

$$y(t+1) = -a_1 y(t) + b_1(t)u(t) = \theta^T \phi(t)$$

Measurement vector

$$\theta^T = [a_1, b_1] \longleftarrow \text{Parameter vector} \quad ; \quad \phi(t)^T = [-y(t), u(t)]$$

Adjustable prediction model (à priori)

$$\hat{y}^o(t+1) = \hat{y}(t+1|\hat{\theta}(t)) = -\hat{a}_1(t)y(t) + \hat{b}_1(t)u(t) = \hat{\theta}(t)^T \phi(t)$$

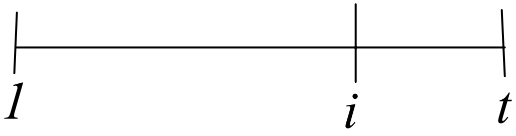
$$\theta(t)^T = [\hat{a}_1(t), \hat{b}_1(t)] \longleftarrow \text{Vector of adjustable parameters}$$

Least squares algorithm

Least squares criterion (objective) :

$$\min_{\hat{\theta}(t)} J(t) = \frac{1}{t}\sum_{i=1}^{t}\left[y(i) - \hat{\theta}(t)^T \phi(i-1)\right]^2 = \frac{1}{t}\sum_{i=1}^{t}\varepsilon^2\left(i, \hat{\theta}(t)\right)$$

$$\hat{\theta}(t)^T \phi(i-1) = -\hat{a}_1(t)y(i-1) + \hat{b}_1(t)u(i-1) = \hat{y}\left(i \middle| \hat{\theta}(t)\right)$$



Output prediction at instant *i (i ≤ t)* based on the estimated parameters at instant *t*

Solution: $\frac{\partial J(t)}{\partial \hat{\theta}(t)} = 0$

(see details in the book pg.216 -218)

$$\boxed{\hat{\theta}(t) = F(t)\sum_{i=1}^{t} y(i)\phi(i-1)}$$

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

Non recursive algorithm!

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

Recursive least squares

$\hat{\theta}(t) \rightarrow \hat{\theta}(t+1)$ without doing all the calculations

$$\hat{\theta}(t) = F(t)\sum_{i=1}^{t} y(i)\phi(i-1) \qquad ; \qquad F(t)^{-1} = \sum_{i=1}^{t} \phi(i-1)\phi(i-1)^T$$

$$\hat{\theta}(t+1) = F(t+1)\sum_{i=1}^{t+1} y(i)\phi(i-1) = \hat{\theta}(t) + \Delta\hat{\theta}(t+1) \xleftarrow{\quad} \textbf{?}$$

$$F(t+1)^{-1} = \sum_{i=1}^{t+1} \phi(i-1)\phi(i-1)^T = F(t)^{-1} + \phi(t)\phi(t)^T \qquad (*)$$

$\downarrow$ *(See book pg.218-220)*

Version I

$$\boxed{\begin{aligned} \hat{\theta}(t+1) &= \hat{\theta}(t) + F(t+1)\phi(t)\varepsilon^{o}(t+1) && (**) \\ F(t+1) &= F(t) - \frac{F(t)\phi(t)\phi(t)^T F(t)}{1+\phi(t)^T F(t)\phi(t)} && (***) \\ \varepsilon^{o}(t+1) &= y(t+1) - \hat{\theta}(t)^T \phi(t) \end{aligned}}$$

To pass from (*) to (***) one uses the « Matrix inversion lemma »

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

Recursive least squares

Replacin *F(t+1)* in (**) by (***) one gets:

$$F(t+1)\phi(t)\varepsilon^{o}(t+1) = F(t)\phi(t)\frac{\varepsilon^{o}(t+1)}{1+\phi(t)^{T}F(t)\phi(t)}$$

On the other hand :

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

Version II
(for analysis)

$$\hat{\theta}(t+1) = \hat{\theta}(t) + F(t)\phi(t)\varepsilon(t+1)$$

$$F(t+1)^{-1} = F(t)^{-1} + \phi(t)\phi(t)^{T}$$

$$F(t+1) = F(t) - \frac{F(t)\phi(t)\phi(t)^{T}F(t)}{1+\phi(t)^{T}F(t)\phi(t)}$$

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

Recursive least squares

Initial adaptation gain: $F(0) = \frac{1}{\delta} I = (GI)I \quad ; \quad 0 < \delta << 1$

Typical value: *GI = 1000* *Stable algorithm for any F(0)>0*

It is a decreasing gain algorithm !!

Consder the case of a single parameter. F et ϕ are scalar.
In this case one gets:

$$F(t+1) = \frac{F(t)}{1 + \phi(t)^2 F(t)} \leq F(t)$$

Rem.: the algorithmn can not follow variations of the parameters

Recursive least squares

The algorithm can be generalized for any dimension.

Model to be identified:

$$y(t) = \frac{q^{-d}B(q^{-1})}{A(q^{-1})}u(t) \qquad A(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_{n_A} q^{-n_A}$$

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

$$y(t+1) = -\sum_{i=1}^{n_A} a_i y(t+1-i) + \sum_{i=1}^{n_B} b_i u(t-d-i+1) = \theta^T \phi(t)$$

$$\theta^T = \left[a_1,\ldots a_{n_A}, b_1,\ldots, b_{n_B}\right];\ \phi(t)^T = \left[-y(t)\ldots - y(t-n_A+1), u(t-d)\ldots u(t-d-n_B+1)\right]$$

a priori adjustable predictor:

$$\hat{y}^o(t+1) = -\sum_{i=1}^{n_A} \hat{a}_i(t) y(t+1-i) + \sum_{i=1}^{n_B} \hat{b}_i(t) u(t-d-i+1) = \hat{\theta}(t)^T \phi(t)$$

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

a priori predicition error:

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

One uses the least squares algorithm given previously

Choice of the adaptation gain *F(t)*

General form:

$$\boxed{\begin{array}{l} F(t+1)^{-1} = \lambda_1(t)F(t)^{-1} + \lambda_2(t)\phi(t)\phi(t)^T \\ 0 < \lambda_1(t) \leq 1 \ ; \ 0 \leq \lambda_2(t) < 2 \ ; \ F(0) > 0 \end{array}}$$

$$F(t+1) = \frac{1}{\lambda_1(t)}\left[F(t) - \frac{F(t)\phi(t)\phi(t)^T F(t)}{\frac{\lambda_1(t)}{\lambda_2(t)} + \phi(t)^T F(t)\phi(t)}\right]$$

A.1 Decreasing gain (RLS): $\lambda_1(t) = \lambda_1 = 1 \ ; \ \lambda_2(t) = 1$ $\boxed{t \nearrow F(t)^{-1} \nearrow F(t) \searrow}$

Identification of stationary systems (constant parameters)

Forgetting factor

A.2 Facteur d'oubli fixe: $\lambda_1(t) = \lambda_1 \ ; \ 0 < \lambda_1 < 1 \ ; \ \lambda_2(t) = \lambda_2 = 1$

Typical values for λ_1: $\lambda_1 = 0.95,...,0.99$

Minimized criterion: $J(t) = \sum_{i=1}^{t} \lambda_1^{(t-i)} \left[y(i) - \hat{\theta}(t)^T \phi(i-1)\right]^2$

Identification of slowly time varying systems

Choice of the adaptation gain

A.3 Variable forgetting factor: $\lambda_1(t) = \lambda_0 \lambda_1(t-1) + 1 - \lambda_0 \quad ; \quad 0 < \lambda_0 < 1$

$$\lambda_2(t) = \lambda_2 = 1$$

Typical values: $\lambda_1(0) = 0.95,...,0.99 \quad ; \quad \lambda_0 = 0.95,...,0.99$

Minimized criterion:

$$J(t) = \sum_{i=1}^{t} \left[\prod_{j=1}^{t-1} \lambda_1(j-i) \right] \left[y(i) - \hat{\theta}(t)^T \phi(i-1) \right]^2$$

Since $\lambda_1(t)$ tends towards *1* for large *i*, one forgets only initial data

Reccomended for the identification of staionary systems.
Offers in general better performances than A.1

Choice of the adaption gain

A.4 Constant trace: $trF(t+1) = trF(t) = trF(0) = nGI$

$$F(0) = \begin{bmatrix} GI & & 0 \\ & \cdot & \\ & & \cdot \\ 0 & & GI \end{bmatrix}$$

n = number of parameters

$GI = (0.01)0.1 \text{ to } 4$

$$trF(t+1) = \frac{1}{\lambda_1(t)} tr\left[F(t) - \frac{F(t)\phi(t)\phi(t)^T F(t)}{\alpha(t) + \phi(t)^T F(t)\phi(t)} \right] = trF(t)$$

$$\Big\downarrow$$

One computes: $\lambda_1(t)$, $\quad for\ \alpha(t) = \lambda_1(t)/\lambda_2(t)\ fixed$

Identification of time varying systems

A.5 Decreasing gain + constant trace

One switches from A.1 to A.4 when: $trF(t) \leq nG \quad ; \quad G = (0.01)0.1 \text{ to } 4$

Identification of time varuing systems in the absence of initial information upon the parameters

Choice of the adaptation gain

A.6 Variable forgetting factor + constant trace

One switches from A.3 to A.4 when: $trF(t) \leq nG \quad ; \quad G = (0.01)0.1 \text{ to } 4$

Identification of time varuing systems in the absence of initial information upon the parameters

A.7 Constant gain (improved gradient algorithm)

$$\lambda_1(t) = \lambda_1 = 1 \quad ; \quad \lambda_2(t) = \lambda_2 = 0 \quad \longrightarrow \quad F(t+1) = F(t) = F(0)$$

Identification of systems with few parameters (≤ 3) *and low noise level.*
Simple implementation but performance inferior to A.1, A.2, A.3 and A.4

Choice of the initial adaptation gain *F(0)*

$$F(0) = \frac{1}{\delta} I = (GI)I$$

The adaption gain can be interpreted as a measure of the Parametric error (precision of the estimation).

Without initial information upon the parameters: $GI = 1000\,;\hat{\theta}(0) = 0$

Initial parameter estimation is available: $GI =<< 1\,;\hat{\theta}(0) = \hat{\theta}_0$

The trace of the gain matrix is a measure of the « value » of the adaptation gain

Remark:

If the trace of F(t) does not decrease significantly, in general the parameter estimatin is bad.

(can happens when the excitation signals are not appropriate)

Choice of the input sequence for identification

« null prediction error » does not impliesin all the cases
« estimation of the true parameters »!!

Plant model: $y(t+1) = -a_1 y(t) + b_1 u(t)$

Estimated model: $\hat{y}(t+1) = -\hat{a}_1 y(t) + \hat{b}_1 u(t)$

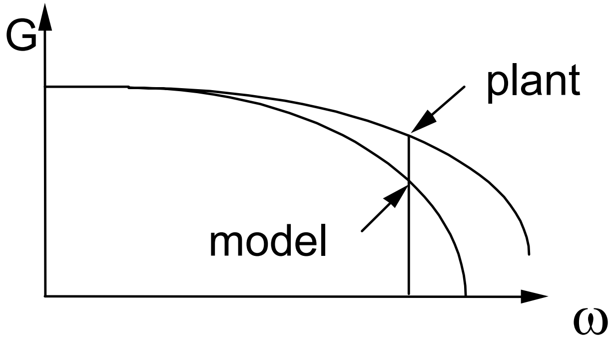
$u(t) = const.$

$$\frac{b_1}{1+a_1} = \frac{\hat{b}_1}{1+\hat{a}_1}$$

The two models have the same static gain but $\hat{a}_1 \neq a_1 \,;\, \hat{b}_1 \neq b_1$

$$y(t+1) = y(t) = \frac{b_1}{1+a_1} u \quad \text{et} \quad \hat{y}(t+1) = \hat{y}(t) = \frac{\hat{b}_1}{1+\hat{a}_1} u$$

$$\varepsilon(t+1) = y(t+1) - \hat{y}(t+1) = 0 \quad \text{for } u(t) = const. \,;\, \hat{a}_1 \neq a_1 \,;\, \hat{b}_1 \neq b_1$$



If we would like to distinguish the two models one should apply: $u(t) = \sin(\omega t) \,;\, \omega \neq 0$

Analysis

$$\varepsilon(t+1) = y(t+1) - \hat{y}(t+1) = -[a_1 - \hat{a}_1]y(t) + [b_1 - \hat{b}_1]u(t) = 0$$

$$\uparrow$$

$$y(t) = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}} u(t)$$

$$\varepsilon(t+1) = \left[(\hat{a}_1 - a_1) b_1 q^{-1} + (b_1 - \hat{b}_1)(1 + a_1 q^{-1}) \right] u(t) = 0$$

$$\varepsilon(t+1) = \left[(b_1 - \hat{b}_1) + q^{-1}(b_1 \hat{a}_1 - a_1 \hat{b}_1) \right] u(t) = (\alpha_0 + \alpha_1 q^{-1}) u(t) = 0 \qquad \textbf{(*)}$$

$\nearrow$ Solution of the recursive equation: $u(t) = z^t = e^{sT_e t}$

$$(\alpha_0 + z^{-1}\alpha_1) z^t = 0 \quad \longrightarrow \quad z = -\frac{\alpha_1}{\alpha_0} = e^{\sigma T_e} \; ; \sigma = r\acute{e}el \qquad \boxed{u(t) = e^{\sigma T_e t}}$$

$$u(t) = const \Rightarrow \sigma = 0 \Rightarrow z = 1 \Rightarrow -\alpha_1 = \alpha_0 \Rightarrow b_1 - \hat{b}_1 = a_1 \hat{b}_1 - b_1 \hat{a}_1 \Rightarrow \frac{b_1}{1 + a_1} = \frac{\hat{b}_1}{1 + \hat{a}_1}$$

Problem : find *u(t)* such that: $\varepsilon = 0 \Rightarrow \hat{a}_1 = a_1 ; \hat{b}_1 = b_1$

Answer: *u(t)* should not be a solution of (*).

An example : $u(t) = e^{j\omega T_e t} \quad or \quad e^{-j\omega T_e t} \; or \sin \omega T_e t, \omega \neq 0$

General case – choice of the excitation signal

Structure of the model to be indentified:

$$y(t) = -\sum_{i=1}^{n_A} a_i y(t-i) + \sum_{i=1}^{n_B} b_i u(t-d-i)$$

Number of the parameters to be identified: $= n_A + n_B$

Excitation signal: $u(t) = -\sum_{i=1}^{p} \sin \omega_i T_e t$

One should choose *p* such that *u(t)* can not be a solution of the recursive equation for ε whch features the parametr errors.

$$\left.\begin{array}{ll} n_A + n_B = even & p \geq \dfrac{n_A + n_B}{2} \\ n_A + n_B = odd & p \geq \dfrac{n_A + n_B + 1}{2} \end{array}\right\}$$

In order to identify correctly one should use an input « rich » in frequencies.

Standard solution: *Pseudo Random Binary Sequence (PRBS)*

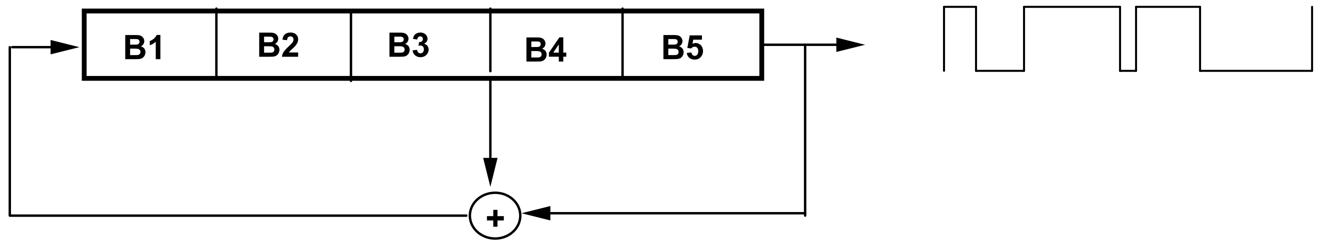
Pseudo random binary sequence (PRBS)

Sequence of rectangular pulses, modulated in width

(rich in frequencies – almost uniform spectral density from 0 to $0.5f_s$)

Generation: *shift registers connected in feedbck*

Example : generation of a PRBS of length $31=2^5-1$



(addition modulo 2)

Length of the sequence **: gives its periodicity.**

Random variation of the pulse width within a sequence

Characteristic parameters:
- number of cells (N)
- Maximum pulse duration ($t_{im}=Nt_e$)
- length of the sequence ($L=2^N-1$)

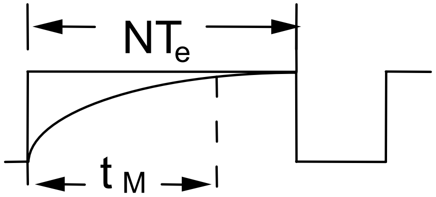
C++ code and .m file for generation of PRBS: see the book website

Sizing the PRBS

For a correct identification od the stedy state gain:

Approach 1 1 (choice of N)

$t_{im} = T_e.N > t_M$



$$\text{But : } (2^N - 1)T_e = L \leq \text{allowed duration of the experiment}$$

$$\boxed{N \rightarrow NT_e \geq t_M \; ; (2^N - 1)T_e \leq \text{exp}\mathit{eriment\ duration}}$$

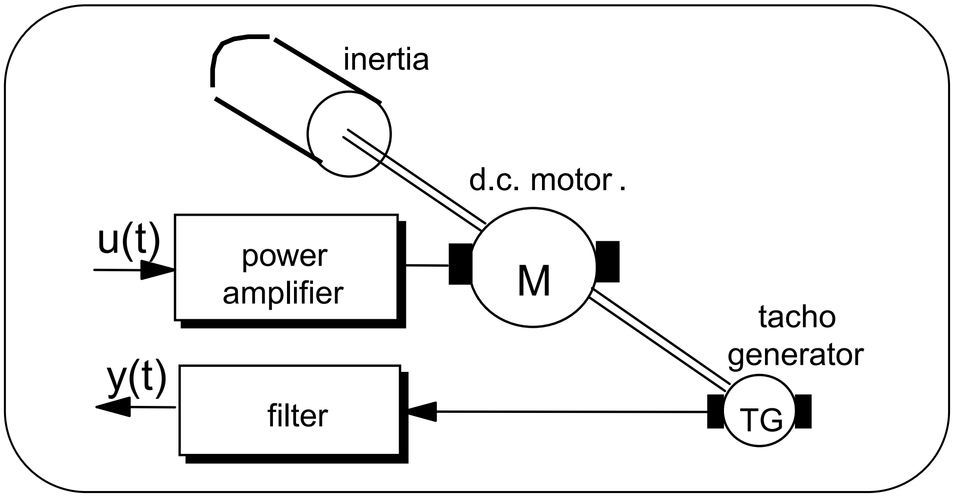
Approach 2 (choice of N and of the PRBS clock frequency f_{PRBS})

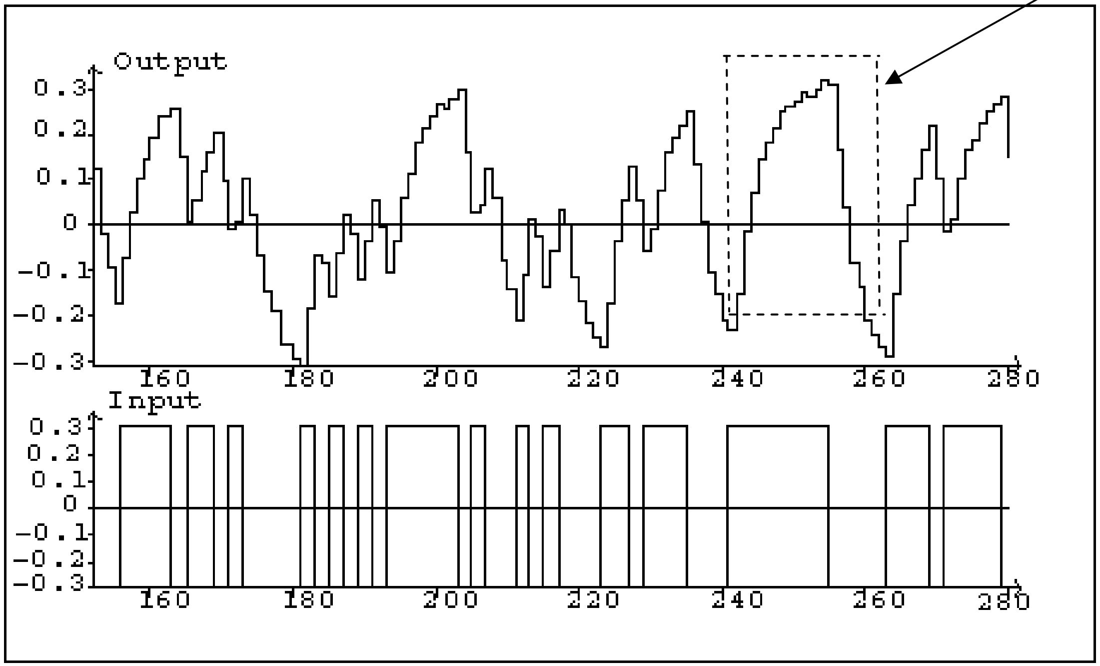
$$f_{PRBS} = \frac{f_e}{p} \quad ; \quad p = 1,2,3,...$$

$$\boxed{N, p \rightarrow pNT_e > t_M \; ; p(2^N - 1)T_e \leq \mathit{dur\acute{e}e\ de\ l'\acute{e}ssai}}$$

Approach 2 allows to get a larger maximum pulse width for the same duration of the experiment (see detailed comparison in the book pg.231-233**)**

An I/O File





One gets the full time response for the largest pulse

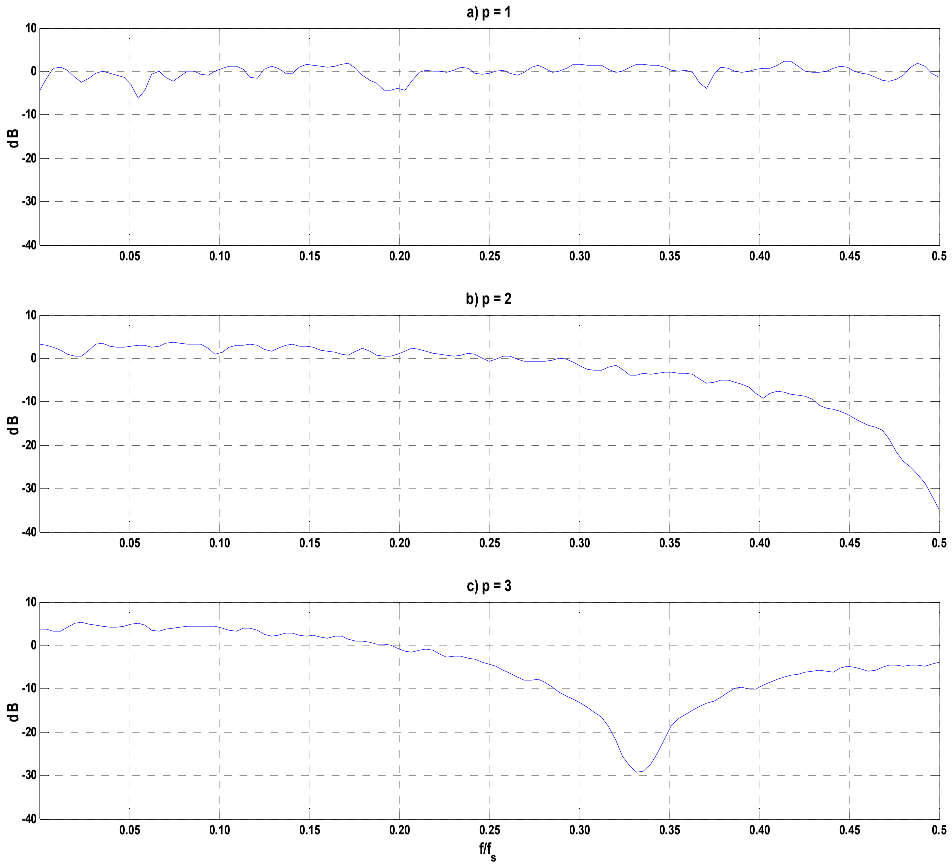
Spectral density of a PRBS.

N=8

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

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

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



The use of a frequency divider for the clock frequency of the PRBS raises
The spectral density at low frequencies and reduces it at high frequencies

Choice of the magnitude for the PRBS

- Low magnitude, but higher than the level of the residual noise
- Low signal/noise ratio requires a longer experiment
- Augmenting the level of the PRBS is not a good ideea (nonlinear phenomena in the process may appear)

Typical values :

0.5%(0.15%) to 10% (5%) of the steady state input (operating point)

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: $y(t+1)=\theta^T\psi(t)+w(t)=\theta^T\phi(t)+w(t)$

Bias for the least squares algorithm:

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

Condition for asymptotic unbiased estimation

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

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*

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

Necessary condition for unbiased estimation:

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

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

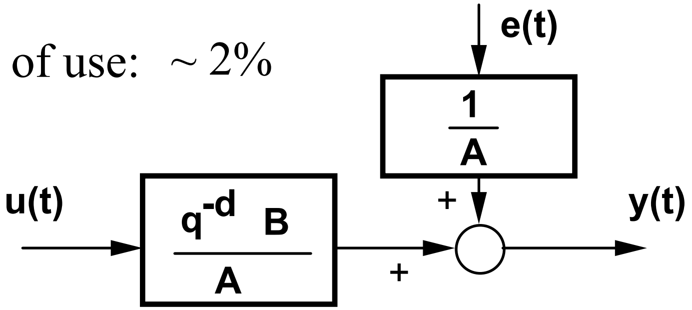
Remark:

One makes assumptions on the “noise”
and
One constructs the appropriate algorithm

«Plant + Distrurbance » Models

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

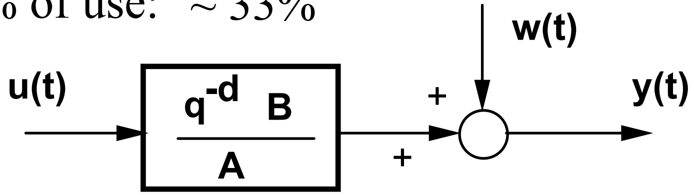
% of use: ~ 2%



(Recursive) Least Squares

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

% of use: ~ 33%

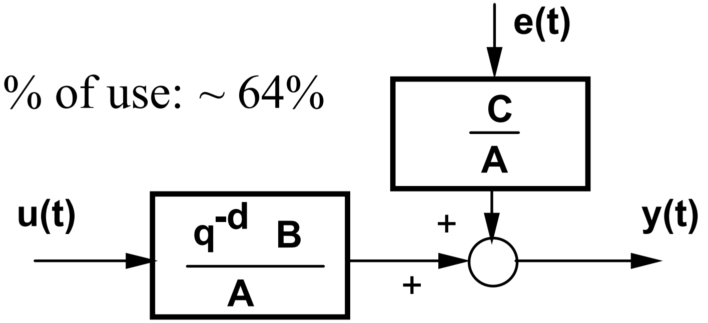


u and w are independent

Ouput Error(O.E.)
Instrumental Variable…

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

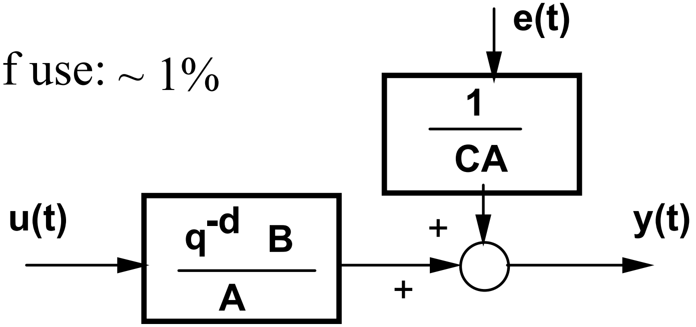
% of use: ~ 64%



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

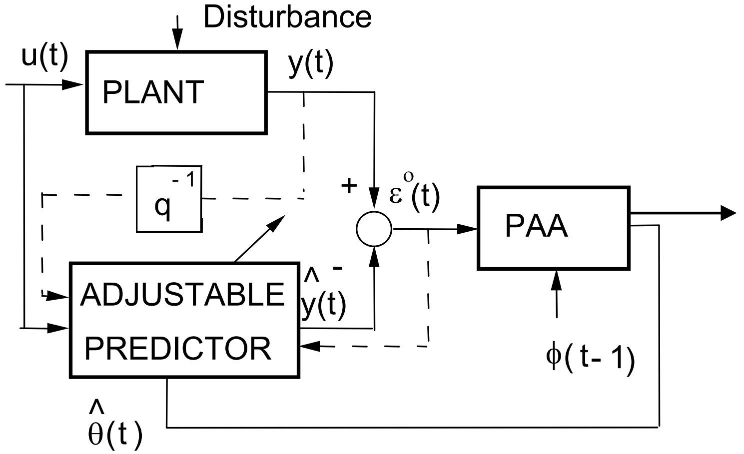
$$S4: A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + \left[1/C(q^{-1})\right]e(t)$$

% of use: ~ 1%



Generalized Least Squares

Structure of recursive identification methods



$$\hat{\theta}(t+1)=\hat{\theta}(t)+F(t+1)\Phi(t)\varepsilon^{0}(t+1)$$
$$F^{-1}(t+1)=\lambda_1(t)F(t)+\lambda_2\Phi(t)\Phi(t)^T$$
$$0<\lambda_1(t)\leq 1\ ;\ 0\leq\lambda_2(t)<2\ ;\ F(0)>0$$

(*)

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 parmeter adaption 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

Identification methods

- There are several structures « plant + noise »
- It does not exist a unique method usable for the structures

Consequence:

In practice one needs an interactive system for identification

It should provide:

- Different structures for the “plant + disturbance”
- Different identification methods and PAA
- Validation methods for the identified models
- A system for input/output data acquisition and processing (including the generation of a PRBS)
- Tools for the model analysis
- Tools for visualising different graphs and plots

Example: WinPIM (Adaptech). See www.adaptech.com

Routines for system identification (Scilab and Matlab) can be downloaded from the book website (www.landau-bookic.lag.ensieg.inpg.fr)

Some concluding remarks

-Identification includes 4 steps (*I/O acquisition, model complexity estimation, parameter estimation, model validation*)

- Parameter estimation can be done recursively or non recursively
- Unicity of identified parameters depends upon the input used
- Standard input for identification :

Pseudo Random Binary Sequence (PRBS)

- Disturbances and noise can lead to parameter estimation error (*bias*)
- It does not exist a unique structure« plant + disturbance » which describe all the situations encountered in practice
- It does not exist a unique parmater identification method which gives good results for all the« plant + disturbance » structures
- One needs an interactive system for doing *identification*