Parametric models

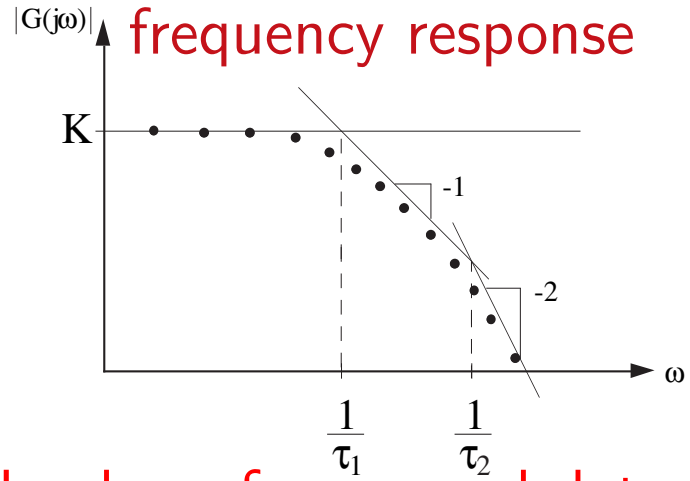
Transfer functions and state-space models (a structure is required) :

$$G(s) = \frac{K(s+1)}{(s+3)(s+5)} \quad ; \quad G(z) = \frac{z^2 + 2z + 3}{z^3 + 3z^2 + 3z + 8}$$

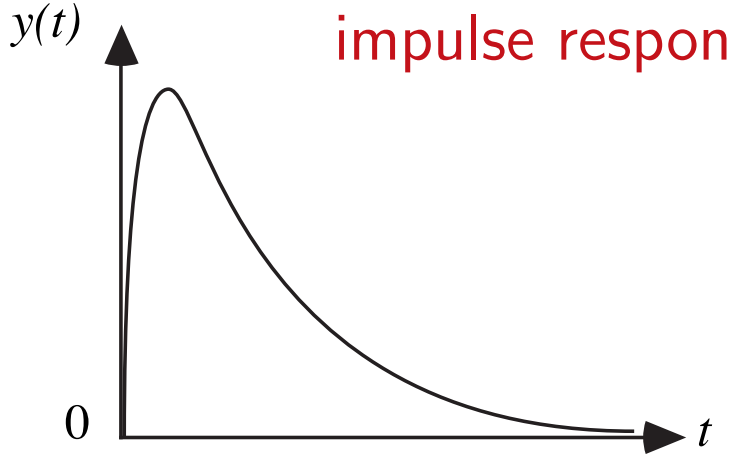
Number of parameters in model << Number of measured data
(redundancy leads to optimization approach)

Nonparametric models : Graphs, data, etc. (no structure is needed)

frequency response



impulse response



Number of measured data = Number of data in nonparametric model
algebraic approach (no redundancy)

Basic Ingredients

- A set of experimental data.

$$Z^N = \{(y(k), u(k)) \,|\, k = 1, \ldots, N\}$$

- A model structure : a mapping from the past data Z^{k-1} to the space of the model outputs. This model structure is used to define a parameterized predictor :

$$\hat{y}(k, \theta) = \mathcal{F}(\theta, Z^{k-1})$$

- A fit criterion that should be minimized :

$$J(\theta) = \sum_{k=1}^{N} [y(k) - \hat{y}(k, \theta)]^2$$

- Model validation

Basic Model Structures

We consider LTI discrete time model structures.

Finite Impulse Response (FIR)

The output of the system depends only on the past inputs :

$$y(k) = b_1 u(k-1) + b_2 u(k-2) + \cdots + b_m u(k-m) + e(k)$$

Autoregressive with external input (ARX)

The output depends on the past inputs and past outputs :

$$y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) = b_1 u(k-1) + \cdots + b_m u(k-m) + e(k)$$

State-space model

The output depends on the states and the inputs :

$$x(k+1) = Ax(k) + Bu(k) + w(k)$$
$$y(k) = Cx(k) + Du(k) + e(k)$$

FIR structure

FIR model : Suppose that the output of a system depends only on the present and past inputs, i.e. :

$$\begin{aligned} y(k) &= b_0 u(k) + b_1 u(k-1) + \cdots + b_m u(k-m) \\ &= [b_0 + b_1 q^{-1} + \cdots + b_m q^{-m}] u(k) = B(q^{-1}) u(k) \end{aligned}$$

The output computed by the convolution sum is :

$$y(k) = g(k) * u(k) = \sum_{j=0}^{\infty} g(j) u(k-j) = g(0)u(k) + g(1)u(k-1) + \cdots$$

The parameters of the FIR model are the first $m+1$ components of the impulse response : $g(k) = b_k$ for all $k \leq m$.

Time delay

Sampled discrete time systems have always some delay (at least one sampling delay). The number of leading coefficients of $B(q^{-1})$ that are equal to zero is called *time delay* and denoted by $d \geq 1$.

FIR structure

FIR model : The *true model* with a time delay d is supposed to be

$$y(k) = b_d^\circ u(k-d) + b_{d+1}^\circ u(k-d-1) + \cdots + b_m^\circ u(k-m) + e(k)$$

Parameterized predictor : $\hat{y}(k,\theta) = \phi^T(k)\theta$ where

$$\phi^T(k) = [u(k-d)\,,\, u(k-d-1)\,,\, \ldots\,,\, u(k-m)]$$
$$\theta^T = [b_d\,,\, b_{d+1}\,,\, \ldots\,,\, b_m]$$

Fit criterion :

$$J(\theta) = \sum_{k=1}^{N} [y(k) - \hat{y}(k,\theta)]^2 = \sum_{k=1}^{N} [y(k) - \phi^T(k)\theta]^2$$

Least squares solution :

$$\hat{\theta} = \left[\sum_{k=1}^{N} \phi(k)\phi^T(k)\right]^{-1} \sum_{k=1}^{N} \phi(k)y(k) = (\Phi^T\Phi)^{-1}\Phi^T Y$$

Relation with the correlation approach ?

FIR structure

Quality of the estimates : The *true model* is

$$y(k) = b_d^\circ u(k-d) + \cdots + b_m^\circ u(k-m) + e(k) = \phi^T(k)\theta_0 + e(k)$$

The least squares estimates are :

$$\hat{\theta} = \left[\sum_{k=1}^{N} \phi(k)\phi^T(k)\right]^{-1} \sum_{k=1}^{N} \phi(k)[\phi^T(k)\theta_0 + e(k)]$$

$$= \theta_0 + \left[\sum_{k=1}^{N} \phi(k)\phi^T(k)\right]^{-1} \sum_{k=1}^{N} \phi(k)e(k) = \theta_0 + (\Phi^T\Phi)^{-1}\Phi^T E$$

Biasedness : If $e(k)$ is zero mean and independent of $u(k)$ (and consequently of $\phi(k)$), then the estimates are unbiased, i.e. : $\mathbb{E}\{\hat{\theta}\} = \theta_0$.

Covariance of the estimates : If $e(k)$ is white with variance σ^2, then

$$\text{cov}[\hat{\theta}] = \mathbb{E}\left\{(\hat{\theta} - \theta_0)(\hat{\theta} - \theta_0)^T\right\} = \sigma^2 \left[\sum_{k=1}^{N} \phi(k)\phi^T(k)\right]^{-1} = \sigma^2(\Phi^T\Phi)^{-1}$$

FIR structure

Quality of the estimates : The covariance of the estimate can be rewritten as

$$\text{cov}[\hat{\theta}] = \frac{\sigma^2}{N}\left[\frac{1}{N}\sum_{k=1}^{N}\phi(k)\phi^T(k)\right]^{-1} = \frac{\sigma^2}{N}\hat{R}_{\phi\phi}^{-1}(0)$$

- The covariance of the parameter error decays like $1/N$.
- The covariance is proportional to noise-to-signal-ratio.
- The covariance does not depend on the specific shape of input or noise signal.
- A good experiment is the one in which the covariance matrix of the input signal, or the information matrix, is large.

Estimation of the noise variance : An unbiased estimate is given by

$$\widehat{\sigma^2} = \frac{1}{N-m}J(\hat{\theta})$$

FIR structure

FIR and basis functions : An FIR estimator can be seen as an estimator using a particular basis function :

$$\hat{y}(k,\theta) = \sum_{i=d}^{m} \theta_i q^{-i} u(k)$$

where $[q^{-d}, q^{-d-1}, \ldots, q^{-m}]u(k)$ can be considered as a vector of basis functions. When m goes to infinity, any linear model can be represented by this predictor.

General basis functions : By choosing other basis functions better predictors with smaller number of parameters can be obtained. For example :

$$\hat{y}(k,\theta) = \sum_{i=d}^{m} \theta_i \frac{q^{-i}}{A(q^{-1})} u(k)$$

where $A(q^{-1})$ is a known polynomial. If the roots of $A(q^{-1})$ includes all poles of the model, then with a few parameters the output can be estimated.
In practice, as the poles of the model are unknown, a good choice of $A(q^{-1})$ with the poles close to the dominant poles of the model can reduce significantly the number of parameter to estimate.

ARX structure

ARX model

The *true model* is supposed to be

$$y(k) + a_1^\circ y(k-1) + \cdots + a_n^\circ y(k-n) = b_d^\circ u(k-d) + \cdots + b_m^\circ u(k-m) + e(k)$$

Parameterized predictor

$$\begin{aligned}\hat{y}(k,\theta) &= -a_1 y(k-1) - \cdots - a_n y(k-n) \\ &\quad + b_d u(k-d) + \cdots + b_m u(k-m) = \phi^T(k)\theta\end{aligned}$$

where $\theta^T = [a_1, \ldots, a_n, b_d, \ldots, b_m]$ and

$$\phi^T(k) = [-y(k-1), \ldots, -y(k-n), u(k-d), \ldots, u(k-m)]$$

Fit criterion : $J(\theta) = \sum_{k=1}^{N} [y(k) - \hat{y}(k,\theta)]^2 = \sum_{k=1}^{N} [y(k) - \phi^T(k)\theta]^2$

Least squares solution : $\hat{\theta} = \left[\sum_{k=1}^{N} \phi(k)\phi^T(k)\right]^{-1} \sum_{k=1}^{N} \phi(k)y(k)$

ARX structure

Quality of the estimates : The *true model* is $y(k) = \phi^T(k)\theta_0 + e(k)$ and the least squares estimates are :

$$\hat{\theta} = \theta_0 + \left[\sum_{k=1}^{N} \phi(k)\phi^T(k)\right]^{-1} \sum_{k=1}^{N} \phi(k)e(k)$$

Biasedness : The parametric error is given by

$$\tilde{\theta} \equiv \hat{\theta} - \theta_0 = \left[\frac{1}{N}\sum_{k=1}^{N} \phi(k)\phi^T(k)\right]^{-1} \left[\frac{1}{N}\sum_{k=1}^{N} \phi(k)e(k)\right]$$

If the number of data N goes to infinity

$$\lim_{N\to\infty} (\hat{\theta} - \theta_0) = R_{\phi\phi}^{-1}(0)R_{\phi e}(0)$$

Therefore, the parameter estimates will be asymptotically unbiased if

1. $R_{\phi\phi}(0)$ is not singular,
2. $R_{\phi e}(0) = 0$. This condition is met only if $e(k)$ is white.

ARX structure

What is $e(k)$? It is an equation noise and not the output noise. Let's look at the ARX structure :

$$y(k) + a_1^\circ y(k-1) + \cdots + a_n^\circ y(k-n) = b_d^\circ u(k-d) + \cdots + b_m^\circ u(k-m) + e(k)$$

$$[1 + a_1^\circ q^{-1} + \cdots + a_n^\circ q^{-n}]y(k) = [b_d^\circ q^{-d} + \cdots + b_m^\circ q^{-m}]u(k) + e(k)$$

$$y(k) = \frac{B_0(q^{-1})}{A_0(q^{-1})}u(k) + \frac{1}{A_0(q^{-1})}e(k)$$

The output noise is $e(k)/A_0(q^{-1})$, so assuming $e(k)$ is white is not a reasonable assumption.

ARX structure typically gives biased estimates

Covariance of the estimates

If $e(k)$ is white with variance σ^2, then the asymptotic covariance of the parameters is :

$$\text{cov}(\hat{\theta}) = \mathbb{E}\{(\hat{\theta} - \theta_0)(\hat{\theta} - \theta_0)^T\} = \frac{\sigma^2}{N} R_{\phi\phi}^{-1}(0)$$

ARX structure

Instrumental Variables Method : Let's replace $\phi(k)$ in the LS estimates with $\phi_{iv}(k)$ and keep $\phi^T(k)$ unchanged

$$\hat{\theta}_{iv} = \left[\sum_{k=1}^{N} \phi_{iv}(k)\phi^T(k)\right]^{-1} \left[\sum_{k=1}^{N} \phi_{iv}(k)y(k)\right]$$

The parametric error becomes :

$$\begin{aligned}\tilde{\theta}_{iv} = \hat{\theta}_{iv} - \theta_0 &= \left[\sum_{k=1}^{N} \phi_{iv}(k)\phi^T(k)\right]^{-1} \sum_{k=1}^{N} \phi_{iv}(k)[\phi^T(k)\theta_0 + e(k)] - \theta_0 \\ &= \left[\sum_{k=1}^{N} \phi_{iv}(k)\phi^T(k)\right]^{-1} \sum_{k=1}^{N} \phi_{iv}(k)e(k)\end{aligned}$$

Therefore, the parameter estimates are asymptotically unbiased if :

1. $R_{\phi_{iv}\phi}(0)$ is not singular,
2. $R_{\phi_{iv}e}(0) = 0$.

Choose ϕ_{iv} uncorrelated with noise and correlated with ϕ.

ARX structure

Choice of Instrumental Variables

The instrumental variables should be

1. Uncorrelated with noise to have an asymptotically unbiased estimates.
2. Correlated as much as possible with $\phi(k)$ to make larger the information matrix and reduce the variance of the estimates.

$$\phi^T(k) = [-y(k-1), \ldots, -y(k-n), u(k-d), \ldots, u(k-m)]$$

Let's choose $\phi_{iv}(k)$ as a noiseless estimate of $\phi(k)$

IV based on auxiliary model

- Identify an auxiliary model $M(q^{-1})$ using the ARX or FIR structure.
- Compute : $y_M(k) = M(q^{-1})u(k)$
- Choose the vector of instrumental variables as :

$$\phi_{iv}^T(k) = [-y_M(k-1), \ldots, -y_M(k-n), u(k-d), \ldots, u(k-m)]$$

State-Space Model

State-space representation

An LTI discrete-time model can be represented in state-space form :

$$x(k+1) = Ax(k) + Bu(k) + w(k)$$
$$y(k) = Cx(k) + Du(k) + e(k)$$

where $w(k)$ and $e(k)$ are state and output noise with the covariance :

$$\mathbb{E}\left\{ \begin{bmatrix} w(k) \\ e(k) \end{bmatrix} \begin{bmatrix} w(k) & e(k) \end{bmatrix} \right\} = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}$$

State-space identification problem

Find A, B, C, D, Q, R, S and the system order n using the measured data.

- The solution is not unique and depends on the state-space realization (the choice of states by a similarity transform).
- For given states or measured states the solution is trivial.
- How to find the states from the input/output measurements ?

State-Space Model

There are three methods to identify the state-space models

1) States are measured : The least squares algorithm is used to identify the state-space model.

Subspace Identification Methods

2) Subspace projection : Based on the input/output data a state estimator is constructed and the states are estimated. Then, the least squares algorithm is used to identify the state-space model.

3) Based on the observability matrix : The observability matrix is constructed using the input/output data. Then, the matrices C and A are identified from the observability matrix. Next, the other matrices, B and D are identified using the LS algorithm.

State-Space Model (states are measured)

Trivial solution : Suppose that the states $x(k)$ are measured. Then the state-space model becomes a linear regression :

$$Y(k) = \Theta\Phi(k) + E(k)$$

where

$$Y(k) = \begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix}, \Theta = \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \Phi(k) = \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}, E(k) = \begin{bmatrix} w(k) \\ e(k) \end{bmatrix}$$

Since the parameters are in a matrix, they can be estimated row by row using the LS algorithm :

$$\hat{\Theta}_i^T = \left[\sum_{k=1}^{N} \Phi(k)\Phi^T(k)\right]^{-1} \left[\sum_{k=1}^{N} \Phi(k)Y_i(k)\right]$$

where $\hat{\Theta}_i$ is the LS estimate of the i-th row of Θ and $Y_i(k)$ is the i-th row of $Y(k)$. The covariance of noise can be estimated based on the residuals :

$$\hat{E}(k) = Y(k) - \hat{\Theta}\Phi(k) \quad , \quad \begin{bmatrix} \hat{Q} & \hat{S} \\ \hat{S}^T & \hat{R} \end{bmatrix} = \frac{1}{N}\sum_{k=1}^{N} \hat{E}(k)\hat{E}^T(k)$$

Subspace method

Subspace Method : The method includes the following steps :

1. Estimate the **extended observability matrix** from the data.
2. Use this matrix to estimate the order of the model.
3. Estimate the matrices A and C from the observability matrix.
4. Estimate B and D.
5. Estimate Q, R and S.

This method is detailed in a reverse order.

Subspace method

Estimate B and D :

Suppose that $\hat{A}$ and $\hat{C}$ are known. Then, construct an output predictor, which is a linear regression :

$$\hat{y}(k) = \hat{C}(qI - \hat{A})^{-1}Bu(k) + Du(k) = [B^T \quad D] \begin{bmatrix} u_f^T(k) \\ u(k) \end{bmatrix} = \theta^T \phi(k)$$

where $u_f(k) = \hat{C}(qI - \hat{A})^{-1}u(k)$. Use LS algorithm. The estimates will be unbiased because $\phi(k)$ is not noisy.

For a SISO model with n states we have :
$\hat{C}(qI - \hat{A})^{-1} = \left[F_1(q^{-1}), \cdots, F_n(q^{-1})\right]$ therefore

$$\hat{y}(k) = b_1 F_1(q^{-1})u(k) + \cdots + b_n F_n(q^{-1})u(k) + Du(k) = \phi^T(k)\theta$$

where $\phi^T(k) = [u_{f_1}(k), u_{f_2}(k), \ldots, u_{f_n}(k), u(k)]$ and
$\theta^T = [b_1, b_2, \ldots, b_n, D] = [B^T, D]$, with $u_{f_i}(k) = F_i(q^{-1})u(k)$. So the vector B and the scalar D can be estimated with the classical LS algorithm. A similar procedure can be used for MIMO systems.

Subspace method

Estimate A and C :

Suppose that the extended observability matrix is given as (with $r > n$) :

$$O_r = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{r-1} \end{bmatrix}_{rn_y \times n}$$

- Then, C is the first n_y rows of O_r.
- A can be computed from :

  [the last $(r-1)n_y$ rows of O_r] = [the first $(r-1)n_y$ rows of O_r] $\times \hat{A}$

- Since the rank of O_r for observability is n, there will be a unique solution for $\hat{A}$.

Subspace method

Estimation of the order n :

We have the following facts

- The rank of O_r is n.
- $O_r T$ is also an extended observability matrix, where T is a similarity transform matrix.
  Proof : We have $\bar{C} = CT$ and $\bar{A} = T^{-1}AT$, therefore :

$$O_r T = \begin{bmatrix} CT \\ CAT \\ \vdots \\ CA^{r-1}T \end{bmatrix} = \begin{bmatrix} CT \\ CTT^{-1}AT \\ \vdots \\ CTT^{-1}A^{r-1}T \end{bmatrix} = \begin{bmatrix} \bar{C} \\ \bar{C}\bar{A} \\ \vdots \\ \bar{C}\bar{A}^{r-1} \end{bmatrix}$$

  Note that $T^{-1}A^2T = T^{-1}ATT^{-1}AT = \bar{A}^2$.
- Consider $Q = O_r\tilde{T}$ with $r > n$ columns and $\tilde{T}$ a full rank $n \times r$ matrix. Then the rank of Q is n.

Subspace method

Estimation of the order n :

Algorithm : Suppose that $Q = O_r \tilde{T}$ is available and n is unknown.

1. The order n can be estimated by computing the rank of Q using the singular value decomposition method.
2. Only n singular values of Q are strictly positive, the others are zero (singular values of Q are the square root of the eigenvalues of $Q^T Q$).
3. In the presence of noise the SVD can be applied to $\tilde{Q} = Q\Phi^T$, where Φ is a matrix of instrumental variables, correlated with the input/output data and uncorrelated with noise.
4. The first n columns of $\tilde{Q}$, corresponding to its largest singular values, give the extended observability matrix. This can be computed as the first n columns of $\mathbb{U}$, where $\tilde{Q} = \mathbb{U}\Sigma\mathbb{V}^T$.

Subspace method

Estimation of the observability matrix : The noise-free output of a state-space model is

$$\begin{aligned} y(k+i) &= Cx(k+i) + Du(k+i) \\ &= CAx(k+i-1) + CBu(k+i-1) + Du(k+i) \\ &= \dots \\ &= CA^i x(k) + CA^{i-1}Bu(k) + CA^{i-2}Bu(k+1) + \cdots \\ &\quad + CBu(k+i-1) + Du(k+i) \end{aligned}$$

$$\underbrace{\begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+r-1) \end{bmatrix}}_{Y_r(k)} = \underbrace{\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{r-1} \end{bmatrix}}_{O_r} x(k) + \underbrace{\begin{bmatrix} D & 0 & \cdots & 0 \\ CB & D & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{r-2}B & CA^{r-3}B & \cdots & D \end{bmatrix}}_{S_r} \underbrace{\begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+r-1) \end{bmatrix}}_{U_r(k)}$$

Which leads to

$$Y_r(k) = O_r x(k) + S_r U_r(k)$$

Subspace method

Let us assume that $N + r - 1$ data is available (with $r > n$) and define the following matrices (the dimensions are given for SISO systems) :

$$\boldsymbol{Y} = [Y_r(1)\,,\, Y_r(2)\,,\, \ldots\,,\, Y_r(N)]_{r\times N}$$
$$\boldsymbol{U} = [U_r(1)\,,\, U_r(2)\,,\, \ldots\,,\, U_r(N)]_{r\times N}$$
$$\boldsymbol{X} = [x(1)\,,\, x(2)\,,\, \ldots\,,\, x(N)]_{n\times N}$$

Then rewrite $Y_r(k) = O_r x(k) + S_r U_r(k)$ for $k = 1, \ldots, N$ as :

$$\boldsymbol{Y} = O_r \boldsymbol{X} + S_r \boldsymbol{U}$$

- In this equation only $\boldsymbol{Y}$ and $\boldsymbol{U}$ are available.
- Compute $\boldsymbol{U^\perp = I - U^T (U U^T)^{-1} U}$ which is orthogonal to $\boldsymbol{U}$.
- Multiply the above equation by $[\boldsymbol{U^\perp}]_{N\times N}$ to obtain $Q_{r\times N} \equiv \boldsymbol{Y U^\perp} = O_r \boldsymbol{X U^\perp}$, which is a double extended observability matrix (number of columns is $N > n$).
- Estimate the rank of Q to find n using SVD : $Q = \mathbb{U}\Sigma\mathbb{V}^T$.
- The extended observability matrix is the first n columns of $\mathbb{U}$.

Subspace method

Estimation of the observability matrix : In the presence of state and output noise ($w(k)$ and $e(k)$), we have :

$$\begin{aligned} y(k+i) &= Cx(k+i) + Du(k+i) + e(k+i) \\ &= CAx(k+i-1) + CBu(k+i-1) + Du(k+i) \\ &\quad + Cw(k+i-1) + e(k+i) \\ &= \dots \\ &= CA^i x(k) + CA^{i-1}Bu(k) + CA^{i-2}Bu(k+1) + \cdots \\ &\quad + CBu(k+i-1) + Du(k+i) \\ &\quad + CA^{i-1}w(k) + \cdots + Cw(k+i-1) + e(k+i) \end{aligned}$$

which leads to : $Y_r(k) = O_r x(k) + S_r U_r(k) + V_r(k)$,
where

$$V_r(k) = \begin{bmatrix} V(k) \\ V(k+1) \\ \vdots \\ V(k+r-1) \end{bmatrix} = \begin{bmatrix} e(k) \\ Cw(k) + e(k+1) \\ \vdots \\ CA^{r-2}w(k) + \cdots + Cw(k+r-2) + e(k+r-1) \end{bmatrix}$$

Subspace method

Defining $\boldsymbol{V} = [V_r(1), \ldots, V_r(N)]$, we obtain :

$$\boldsymbol{Y} = O_r \boldsymbol{X} + S_r \boldsymbol{U} + \boldsymbol{V}$$

Let's define : $\phi_r(k) = [\phi(k-1), \ldots, \phi(k-r)]^T$ not correlated with $V_r(k)$ and $\Phi = [\phi_r(1), \ldots, \phi_r(N)]$. Then multiply the equation by $\frac{1}{N}\boldsymbol{U}^{\perp}\Phi^T$ to obtain :

$$\tilde{Q} \equiv \frac{1}{N}\boldsymbol{Y}\boldsymbol{U}^{\perp}\Phi^T = O_r \frac{1}{N}\boldsymbol{X}\boldsymbol{U}^{\perp}\Phi^T + \frac{1}{N}\boldsymbol{V}\boldsymbol{U}^{\perp}\Phi^T = O_r \tilde{T}_N + V_N$$

Here $\tilde{T}_N$ is an $n \times r$ matrix. Suppose we can find $\phi_r(k)$ such that :

$$\lim_{N\to\infty} V_N = \lim_{N\to\infty} \frac{1}{N}\boldsymbol{V}\boldsymbol{U}^{\perp}\Phi^T = 0$$

$$\lim_{N\to\infty} \tilde{T}_N = \lim_{N\to\infty} \frac{1}{N}\boldsymbol{U}^{\perp}\Phi^T = \tilde{T} \quad \text{has full rank } n$$

Then the effect of noise will be asymptotically canceled. This can be achieved if the instrumental variable $\phi_r(k)$ is chosen as a function of $y(k-1)$ or $u(k-1)$.

Linear Black-Box Models

Structures without noise model (OE, FIR)

Assumption : noise is independent from input

$$y(k) = G_0(q^{-1})u(k) + n(k)$$

OE : $G_0(q^{-1}) = \dfrac{B_0(q^{-1})}{A_0(q^{-1})}$ **FIR :** $G_0(q^{-1}) = B_0(q^{-1})$

Structures with noise model (ARX, ARMAX, BJ)

Assumption : noise can be modeled by a filtered white noise

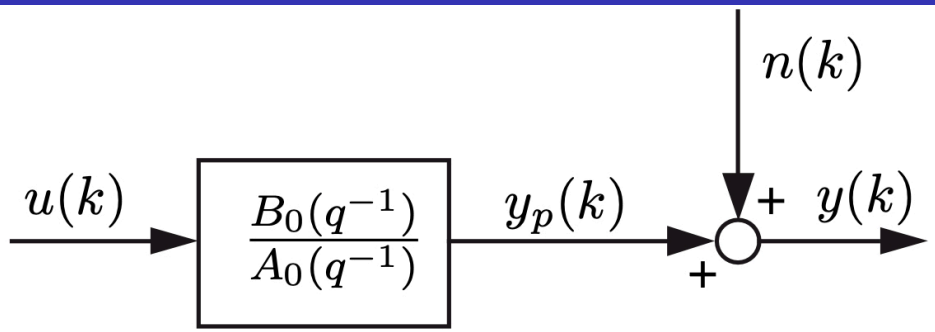
$$y(k) = \frac{B_0(q^{-1})}{A_0(q^{-1})}u(k) + H_0(q^{-1})e(k)$$

ARX : $H_0(q^{-1}) = \dfrac{1}{A_0(q^{-1})}$; **ARMAX :** $H_0(q^{-1}) = \dfrac{C_0(q^{-1})}{A_0(q^{-1})}$

BJ : $H_0(q^{-1}) = \dfrac{C_0(q^{-1})}{D_0(q^{-1})}$

Output error structure

True model



What is the ideal predictor (Suppose that $G_0(q^{-1})$ is known) ?

Answer : $\hat{y}(k) = G_0(q^{-1})u(k)$

What is the ideal prediction error ?

Answer :

$$\begin{aligned} \varepsilon(k) &= y(k) - \hat{y}(k) \\ &= G_0(q^{-1})u(k) + n(k) - G_0(q^{-1})u(k) = n(k) \end{aligned}$$

The ideal prediction error is not correlated with input signal

Output error structure

Parameterized predictor : Now, consider the parameterized predictor $\hat{y}(k,\theta) = G(q^{-1})u(k)$ with unknown parameter vector θ.
Parameterized prediction error :

$$\varepsilon(k,\theta) = y(k) - \hat{y}(k,\theta) = y(k) - G(q^{-1})u(k) = y(k) - \frac{B(q^{-1})}{A(q^{-1})}u(k)$$

- Prediction error is nonlinear w.r.t model parameters.
- Minimizing the identification criterion

$$J(\theta) = \sum_{k=1}^{N} \varepsilon^2(k,\theta)$$

  is a nonlinear least squares problem.
- The optimal solution $\hat{\theta}$ may be a local optimum.
- If $\hat{\theta} = \theta_0$, then the residual $\varepsilon(k,\hat{\theta})$ is not correlated with the input signal (validation test).

Structures with noise model

$$\text{True model :} \qquad y(k) = G_0(q^{-1})u(k) + H_0(q^{-1})e(k)$$

Ideal output predictor : Suppose that $G_0(q^{-1})$ and $H_0(q^{-1})$ are known and

$$\begin{aligned} H_0(q^{-1})e(k) &= \Big[1 + \overbrace{h_1 q^{-1} + \ldots + h_{n_h} q^{-n_h}}^{H_0(q^{-1})-1}\Big]e(k) \\ &= \underbrace{e(k)}_{\text{unpredictable}} + \underbrace{[H_0(q^{-1}) - 1]e(k)}_{\text{known at } k-1} \end{aligned}$$

Ideal predictor $\qquad \hat{y}(k, \theta_0) = G_0(q^{-1})u(k) + [H_0(q^{-1}) - 1]e(k)$

The prediction error for the ideal predictor ?

$$\varepsilon(k, \theta_0) = y(k) - \hat{y}(k, \theta_0) = e(k)$$

The prediction error for the ideal predictor is white

Structures with noise model

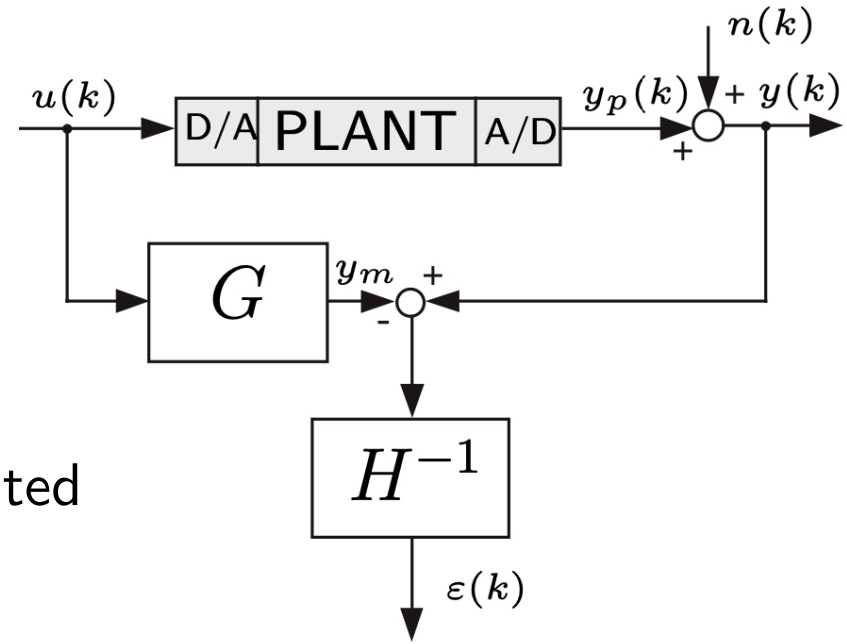
Parameterized output predictor : Since G_0 and H_0 are unknown and $e(k-1)$ is not measureable, the following predictor is proposed.

$$\hat{y}(k,\theta) = G(q^{-1})u(k) + [H(q^{-1}) - 1]\varepsilon(k,\theta)$$

which leads to the following prediction error :

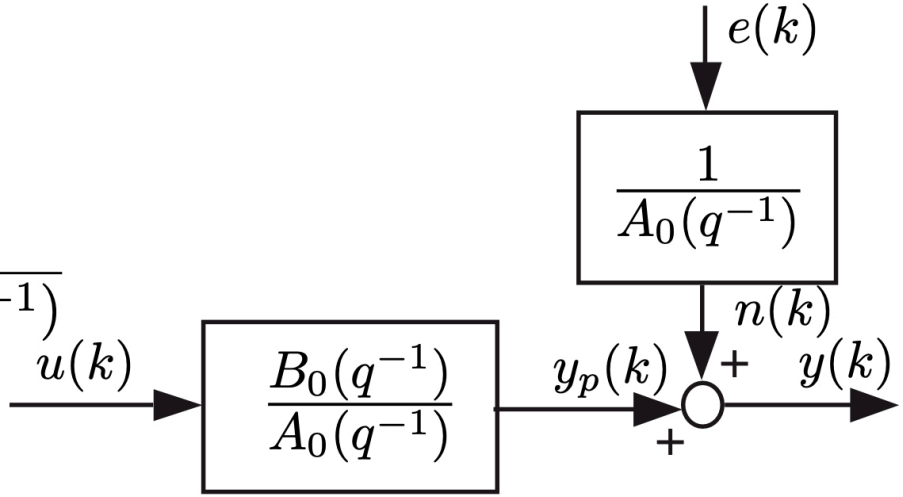
$$\begin{aligned}\varepsilon(k,\theta) &= y(k) - \hat{y}(k,\theta) = y(k) - G(q^{-1})u(k) - [H(q^{-1}) - 1]\varepsilon(k,\theta) \\ &= H^{-1}(q^{-1})[y(k) - G(q^{-1})u(k)]\end{aligned}$$

Important : If G and H have the same structure as G_0 and H_0, and if $G = G_0$ and $H = H_0$, then the residual is white. If $G = G_0$ but $H \neq H_0$, then the residual is uncorrelated with the input.



ARX structure

True model : $H_0 = \dfrac{1}{A_0(q^{-1})}$



General predictor : $\hat{y}(k,\theta) = G(q^{-1})u(k) + [H(q^{-1}) - 1]\varepsilon(k,\theta)$

General prediction error : $\varepsilon(k,\theta) = H^{-1}(q^{-1})[y(k) - G(q^{-1})u(k)]$

ARX predictor : $\hat{y}(k,\theta) = \dfrac{B(q^{-1})}{A(q^{-1})}u(k) + \Big[\dfrac{1}{A(q^{-1})} - 1\Big]\varepsilon(k,\theta)$

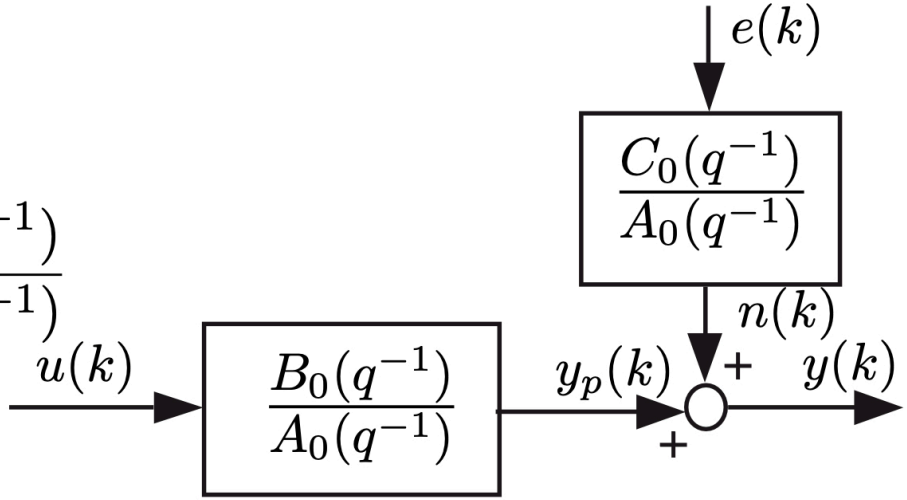
Prediction error for ARX structure :

$$\varepsilon(k,\theta) = A(q^{-1})[y(k) - \frac{B(q^{-1})}{A(q^{-1})}u(k)] = A(q^{-1})y(k) - B(q^{-1})u(k)$$

The prediction error is linear w.r.t the model parameters

ARMAX structure

True model : $H_0 = \dfrac{C_0(q^{-1})}{A_0(q^{-1})}$



General predictor : $\hat{y}(k,\theta) = G(q^{-1})u(k) + [H(q^{-1}) - 1]\varepsilon(k,\theta)$

General prediction error : $\varepsilon(k,\theta) = H^{-1}(q^{-1})[y(k) - G(q^{-1})u(k)]$

ARMAX predictor : $\hat{y}(k,\theta) = \dfrac{B(q^{-1})}{A(q^{-1})}u(k) + \Big[\dfrac{C(q^{-1})}{A(q^{-1})} - 1\Big]\varepsilon(k,\theta)$

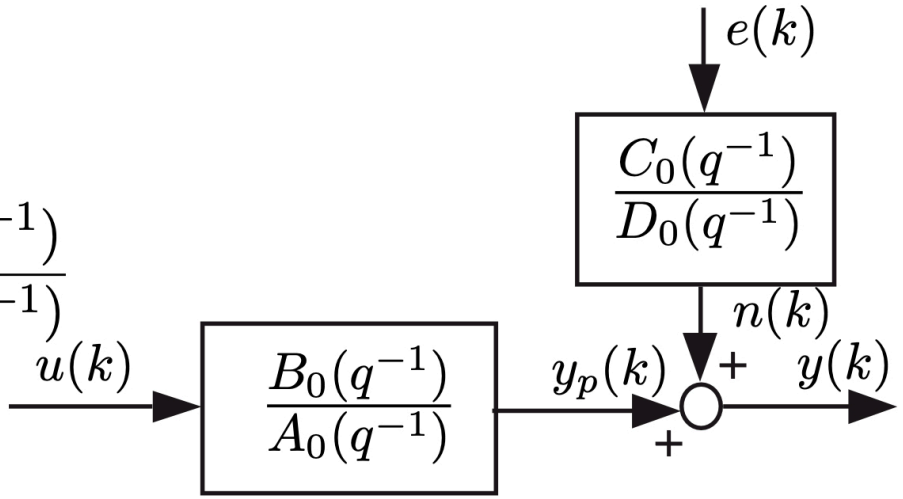
Prediction error for ARMAX structure :

$$\varepsilon(k,\theta) = \frac{A(q^{-1})}{C(q^{-1})}\Big[y(k) - \frac{B(q^{-1})}{A(q^{-1})}u(k)\Big] = \frac{1}{C(q^{-1})}\Big[A(q^{-1})y(k) - B(q^{-1})u(k)\Big]$$

The prediction error is nonlinear w.r.t the model parameters

Box-Jenkins structure

True model : $H_0 = \dfrac{C_0(q^{-1})}{D_0(q^{-1})}$



General predictor : $\hat{y}(k,\theta) = G(q^{-1})u(k) + [H(q^{-1}) - 1]\varepsilon(k,\theta)$

General prediction error : $\varepsilon(k,\theta) = H^{-1}(q^{-1})[y(k) - G(q^{-1})u(k)]$

BJ predictor : $\hat{y}(k,\theta) = \dfrac{B(q^{-1})}{A(q^{-1})}u(k) + \Big[\dfrac{C(q^{-1})}{D(q^{-1})} - 1\Big]\varepsilon(k,\theta)$

Prediction error for BJ structure :

$$\varepsilon(k,\theta) = \frac{D(q^{-1})}{C(q^{-1})}\Big[y(k) - \frac{B(q^{-1})}{A(q^{-1})}u(k)\Big]$$

The prediction error is nonlinear w.r.t the model parameters

Bias distribution in the frequency domain

Parseval's relation

$$R_{\varepsilon\varepsilon}(0) = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \varepsilon^2(k,\theta) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_{\varepsilon\varepsilon}(\omega) d\omega$$

We recall the following relations : if $y(k) = G(q^{-1})u(k)$, then

$$\left.\begin{array}{l} \Phi_{yu}(\omega) = G(e^{j\omega})\Phi_{uu}(\omega) \\ \Phi_{yy}(\omega) = G(e^{j\omega})\Phi_{uy}(\omega) \\ \Phi_{yu}(\omega) = \Phi_{uy}(-\omega) \end{array}\right\} \Rightarrow \Phi_{yy}(\omega) = |G(e^{j\omega})|^2 \Phi_{uu}(\omega)$$

Spectrum of the prediction error for the OE structure

$$\varepsilon(k,\theta) = [G_0(q^{-1}) - G(q^{-1})]u(k) + n(k)$$

$$\Phi_{\varepsilon\varepsilon}(\omega) = |G_0(e^{j\omega}) - G(e^{j\omega})|^2 \Phi_{uu}(\omega) + \Phi_{nn}(\omega)$$

Bias distribution in the frequency domain

Bias distribution for the OE structure

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2\pi} \int_{-\pi}^{\pi} \left[|G_0(e^{j\omega}) - G(e^{j\omega}, \theta)|^2 \Phi_{uu}(\omega) + \Phi_{nn}(\omega) \right] d\omega$$

- If G and G_0 have the same structure the minimum of the criterion is obtained for $\hat{\theta} = \theta_0$.
- If G and G_0 have different structure, a good approximation of G_0 is obtained where the spectrum of u is large.
- To have a better model in the frequency where $|F(e^{j\omega})|$ is large, the input and output of the plant can be filtered by $F(q^{-1})$ before minimizing the criterion.

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2\pi} \int_{-\pi}^{\pi} |F(e^{j\omega})|^2 \left[|G_0(e^{j\omega}) - G(e^{j\omega}, \theta)|^2 \Phi_{uu}(\omega) + \Phi_{nn}(\omega) \right] d\omega$$

Bias distribution in the frequency domain

Bias distribution for the structures with noise model

$$\varepsilon(k,\theta) = H^{-1}(q^{-1})\left\{\left[G_0(q^{-1}) - G(q^{-1})\right]u(k) + \left[H_0(q^{-1}) - H(q^{-1})\right]e(k)\right\} + e(k)$$

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2\pi}\int_{-\pi}^{\pi} |H^{-1}(e^{j\omega},\theta)|^2 [|G_0(e^{j\omega}) - G(e^{j\omega},\theta)|^2 \Phi_{uu}(\omega) + |H_0(e^{j\omega}) - H(e^{j\omega},\theta)|^2 \Phi_{ee}(\omega)] d\omega$$

- If G and H have the same structure and order as G_0 and H_0, an asymptotically unbiased estimate is obtained.
- If G and G_0 have different structure, a good approximation is obtained where the spectrum of u and $|H^{-1}(e^{j\omega})|$ are large. For example for the ARX structure with $H^{-1} = A$ a good approximation in HF will be obtained.

Frequency analysis of PE methods

Bias distribution for the structures with noise model

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2\pi} \int_{-\pi}^{\pi} |H^{-1}(e^{j\omega}, \theta)|^2 [|G_0(e^{j\omega}) - G(e^{j\omega}, \theta)|^2 \Phi_{uu}(\omega) + |H_0(e^{j\omega}) - H(e^{j\omega}, \theta)|^2 \Phi_{ee}(\omega)] d\omega$$

- If G and G_0 have the same structure but H and H_0 have different structure, the parameters of G are asymptotically unbiased if there is no common parameters between G and H (BJ structure). For ARX and ARMAX the bias in noise model makes the parameters of the plant model biased.
- If G and H are different with the structure of G_0 and H_0, the plant model G is typically better identified than the noise model H because the spectrum of u is larger than the spectrum of noise.

Comparison of different structures

ARX : simple structure, LS algorithm, questionable noise model, typically gives biased estimates (asymptotically unbiased estimates only when $n(k) = e(k)/A$ or IV are used).

FIR : simple structure (no denominator), LS algorithm, no noise model, unbiased estimate, OE is minimized, needs too many parameters.

OE : nonlinear optimization (GN algorithm), no noise model, asymptotically unbiased estimate, OE is minimized, variance of the estimated parameters is not optimal.

ARMAX : nonlinear optimization (GN algorithm), noise model has a common denominator with the plant model, asymptotically unbiased estimates.

BJ : nonlinear optimization (GN algorithm), noise model independent from plant model, asymptotically unbiased estimates, more parameters to estimate.

Closed loop identification

Why are we interested in closed-loop identification ?

- There are systems that are unstable in open-loop operation.
- The output drift occurs in some systems in open-loop operation.
- It is always better to identify an accurate model in the frequency zone interesting for control.

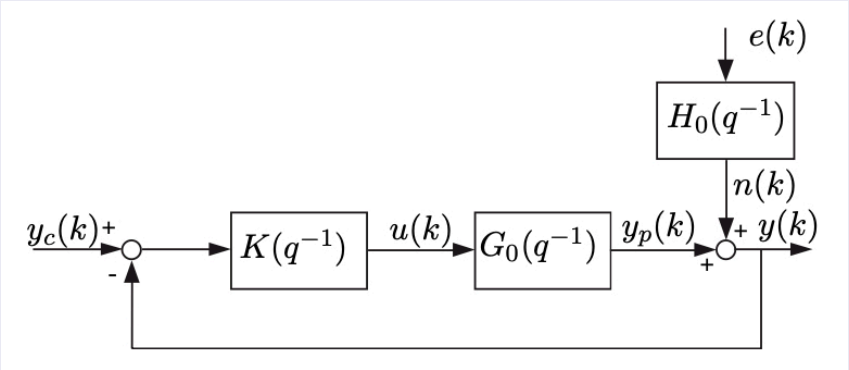
Problems :

- u is correlated with noise via feedback, then the main assumption for the structures without noise model is not met.
- Without an external excitation the information matrix may become singular if the controller is of low order. For example, for a proportional controller, the input and output of the system are linearly dependent $(u(k) = -Ky(k))$ and the information matrix becomes singular. This problem can be solved by changing the controller parameters during data acquisition.

Closed loop identification

Direct methods

The closed loop system is excited with an external signal and a PE method using the structures with a noise model is employed for model identification (ARX, ARMAX or BJ).



A necessary condition to obtain a consistent estimate for G is that not only G and G_0 should have the same structure but also H and H_0

Instrumental Variables Method

$$\varphi_{IV}^T(k) = [-y_M(k-1)\ldots - y_M(k-n)\ ,\ u_M(k-d)\ldots u_M(k-m)]$$

where : $y_M(k) = M_1(q^{-1})y_c(k), u_M(k) = M_2(q^{-1})y_c(k).$

Closed loop identification

Indirect method

First a model for the closed-loop system is identified then, knowing the controller, the plant model is computed :

$$\mathcal{T}(q^{-1}) = \frac{y(k)}{y_c(k)} = \frac{K(q^{-1})G_0(q^{-1})}{1 + K(q^{-1})G_0(q^{-1})}$$

$$G(q^{-1}) = \frac{\mathcal{T}(q^{-1})}{K(q^{-1})[1 - \mathcal{T}(q^{-1})]}$$

Properties :

- Order of G is typically too large.
  (order of $G >$ order of $\mathcal{T} >$ order of G_0)
- For the reason stated above, there is a possibility of pole/zero cancelation in G, so a model order reduction should be done.
- Controller should be known.

Model Validation

1. Validation w.r.t the objective (control, simulation, prediction). If the objective is met, the model is validated.
2. Validation based on a new set of experimental data.

   Time domain validation : Compare the model output (identified from the first set of data) with the measured output (from the validation data).

   Frequency domain validation : Compare the computed frequency response of the model with the identified nonparametric model (obtained by spectral analysis) in the Bode diagram.
3. Validation by statistical methods (verification of assumptions).

   Whiteness of residuals : For ARX, ARMAX and BJ.

   Independence of residuals and past data : For all structures.

Model Validation

Whiteness test

For the structures with noise model (ARX, ARMAX and BJ) the residual $\varepsilon(k, \hat{\theta})$ should be white, if the estimated parameters (for the plant model and the noise model) are equal to the true parameters $\hat{\theta} = \theta_0$.

How can we test the whiteness of the residuals ?

We can compute the autocorrelation of $\varepsilon(k, \hat{\theta})$. If $R_{\varepsilon\varepsilon}(h) = 0 \quad \forall h \neq 0$ then $\varepsilon(k, \hat{\theta})$ is white.

However, this condition is never satisfied for a finite number of data.

- For finite number of data $\hat{R}_{\varepsilon\varepsilon}(h) \quad \forall h \neq 0$ is a zero-mean random variable if the estimate of the correlation function is unbiased.
- We can compute a confidence interval around zero, if we know the probability density function of $\hat{R}_{\varepsilon\varepsilon}(h) \quad \forall h \neq 0$.

What is the pdf of $\hat{R}_{\varepsilon\varepsilon}(h)$?

Model Validation (Whiteness Test)

Theorem (Central limit Theorem)

Consider N independent random variables $x_1, x_2, \ldots, x_N$ with mean μ and finite variance σ^2 (with unknown distribution). Let $\bar{x}_N = \frac{1}{N}\sum_{n=1}^{N} x_n$, then $\sqrt{N}\bar{x}_N$ (for large N) converges in distribution to $\mathcal{N}(\mu\sqrt{N}, \sigma^2)$.

Take $x_h = \varepsilon(k)\varepsilon(k-h)$ with $h \neq 0$ as a random variable and assume that $\varepsilon(k)$ is white. Then $\mu = \mathbb{E}\{x_h\} = 0$ and

$$\sigma^2 = \mathbb{E}\{\varepsilon^2(k)\varepsilon^2(k-h)\} = R^2_{\varepsilon\varepsilon}(0)$$

Therefore, according to the central limit Theorem :

$$\sqrt{N-h}\hat{R}_{\varepsilon\varepsilon}(h) = \sqrt{N-h}\frac{1}{N-h}\sum_{k=h}^{N-1}\varepsilon(k)\varepsilon(k-h) \qquad \text{for} \quad h > 0$$

converges in distribution to a Gaussian distribution with $\mathcal{N}\left(0, R^2_{\varepsilon\varepsilon}(0)\right)$.

Model Validation (Whiteness Test)

- Knowing the distribution of $\hat{R}_{\varepsilon\varepsilon}(h)$, we can compute a confidence interval with a given probability for it.
- It is clear that

$$r(h) = \frac{\sqrt{N-h}\hat{R}_{\varepsilon\varepsilon}(h)}{\hat{R}_{\varepsilon\varepsilon}(0)}$$

  has a normal distribution $\mathcal{N}(0,1)$
- The central limit theorem is Valid for large $N-h$. In practice we should have $N > 100$ and estimate the correlation functions for $h < 25$.
- Therefore, If $\varepsilon(k,\hat{\theta})$ is white, then :

$$-2 \leq \frac{\sqrt{N-h}\hat{R}_{\varepsilon\varepsilon}(h)}{\hat{R}_{\varepsilon\varepsilon}(0)} \leq 2 \qquad \text{for } 1 \leq h < 25, N > 100$$

  for a probability of 0.95.
- It does not mean that the probability of whiteness of $\varepsilon(k,\hat{\theta})$ is 0.95.

Model Validation

Cross-correlation test

For all structures the residual $\varepsilon(k, \hat{\theta})$ should be independent of the past inputs. A correlation between the reiduals and input shows that there exists some information about the true system in the residuals, which has not been captured by the model.

Confidence interval

If the output error is not correlated with the input signal then :

$$-2 \leq \frac{\sqrt{N-|h|}\hat{R}_{\varepsilon u}(h)}{\sqrt{\hat{R}_{\varepsilon\varepsilon}(0)\hat{R}_{uu}(0)}} \leq 2 \qquad \text{for } -25 \leq h \leq 25, N > 100$$

with a probability of 0.95.

It does not mean that the probability of the independence of $\varepsilon(k, \hat{\theta})$ and $u(k)$ is 0.95.

Model Validation

Statistical validation tests

After a parametric identification based on the prediction error method, two statistical validation tests are carried out in MATLAB :

- Whiteness test for the residuals.
- Cross-correlation test between the residuals and the past inputs.

- If the cross-correlation test is satisfied then the plant model is validated.
- If both tests are satisfied then the plant and noise model are validated for structures with noise model (ARX, ARMAX, BJ).
- For the OE and FIR structures, the whiteness test is irrelevant. The model is validated if the cross-correlation test is satisfied.

To validate a model, several types of validation tests (statistical, time-domain and frequency-domain) should be performed.

Practical aspects of identification

A successful system identification depends on certain choices :

Sampling period : Choice of sampling period, anti-aliasing filter, numerical problems.

Input design : Choice of excitation signal (step, impulse, sum of sinusoid signals, PRBS, white noise, filtered white noise, ...), choice of magnitude, signal conditioning (scaling, high and low frequency filtering).

Model structure : Choice of linear, nonlinear, state-space, input/output (ARX, ARMAX, etc), and the order selection for the plant and noise model (numerator, denominator, time delay).

Sampling Period

1. The condition of Shannon Theorem must be respected :

$$\omega_s > 2\omega_{\max} \quad \Rightarrow \quad T_s < \frac{\pi}{\omega_{\max}}$$

to avoid the aliasing effect. For physical system $\omega_{\max}$ is infinity. In practice, there is always the aliasing effect. This effect can be reduced by an anti-aliasing filter.

2. The sampling period should not be too small for two reasons :
   - Numerical problems for computing the controller (all poles and zeros of the model goes to 1).
   - Implementation problem (control computation time $\ll T_s$).

3. In practice, if the model is identified for designing a controller, the sampling frequency (or sampling time) is chosen as :

$$20\omega_b < \omega_s < 30\omega_b \qquad \text{or} \qquad \frac{T_r}{10} < T_s < \frac{T_r}{5}$$

where ω_b is the desired closed-loop bandwidth and T_r is the rise time of the step response of the system. If ω_b is not given, it is chosen equal to or slightly greater than the open-loop bandwidth.

Input design

How to design a PRBS :

1. Choice of magnitude with a trade-off between signal to noise ratio and system nonlinearity.
2. Number of data N should be large enough to filter out the noise (200 $<$ N).
3. The length of shift register n is related to the number of parameters in model and desired frequency resolution. n is chosen greater than 6 to have at least 32 frequency points excited. If system contains very low-damped modes a greater n should be chosen.
4. Number of periods is chosen such that 2 is satisfied.
5. A PRBS can be enriched in low frequencies by using a frequency divider (the clock frequency of the shift register is divided by D_f). A rule of thumb is :

$$\text{length of the largest pulse in PRBS} > \text{settling time of the system}$$

$$nD_fT_s > T_{set}$$

This rule should be used with caution because a large value of D_f reduces the frequency contents of PRBS in high frequencies.

Input design

Signal conditioning :

- Input and output should be scaled to have approximately the same magnitude to avoid the numerical problems (otherwise the information matrix or Hessian may be ill-conditioned).
- Aberrant points should be detected and removed.
- The mean value of input and output should be removed :

$$u(k) = u_e(k) - \frac{1}{N}\sum_{k=1}^{N} u_e(k) \qquad y(k) = y_e(k) - \frac{1}{N}\sum_{k=1}^{N} y_e(k)$$

- Low-and high frequency disturbances should be removed from data by appropriate data filter or by appropriate choice of structure of noise model.

Low-pass filter $L(q^{-1}) = \dfrac{1-q^{-1}}{1-\alpha q^{-1}}$ with $\alpha = e^{-T/\tau_f}$

High-pass filter $L(q^{-1}) = \dfrac{1-\alpha}{1-\alpha q^{-1}}$ with $\alpha = e^{-T/\tau_f}$

Estimate the structure (m, n, d) of a system using data

Notation

m : is the degree of $B(q^{-1}) = b_0 + b_1 q^{-1} + \cdots + b_m q^{-m}$.

n : is the degree of $A(q^{-1}) = 1 + a_1 q^{-1} + \cdots + a_n q^{-n}$.

d : is the time between the application of the input and the first "significant" reaction of the system. d is the number of zero leading coefficients of $B(q^{-1})$ $(d \geq 1)$.

δ : is the order of the model $G(z^{-1}) = B(z^{-1})/A(z^{-1})$, $\delta = \max(m, n)$.

$\delta_{\max}$: is an upper bound on the model order.

n_A : is the number of parameters in $A(q^{-1})$ $(n_A = n)$.

n_B : is the number of parameters in $B(q^{-1})$ $(n_B = m - d + 1)$.

n_θ : is the number of parameters to be estimated $(n_\theta = n_A + n_B)$.

Structure selection

Estimation of model order δ

Nonparametric Methods :

- An oscillation in the step or impulse response corresponds to $\delta \geq n \geq 2$, two distinct oscillations give $\delta \geq n \geq 4$.
- Each slope of -20 dB per decade in Bode diagram corresponds to one simple pole of the plant model. Each resonance mode corresponds to a pair of complex poles ($\delta \geq$ number of evident poles).
- The rank of $Q = \boldsymbol{Y}\boldsymbol{U}^{\perp}$ in the subspace method ($\delta =$ rank Q).

Parametric Methods : By over-parameterization

- Zero/Pole cancellation
- Loss function evolution

Model order estimation

Zero/Pole Cancellation

- If δ is chosen too large (the model is over parameterized), there will be zero/pole cancellations in the model.
- We choose ARMAX structure with $d = 1$ and $\delta = n = m = n_c$. Then, for $\delta = 1, \ldots, \delta_{\max}$ we identify a set of models.
- For $\delta > \delta_0$ (the true model order), there will be a common factor M between A, B and C in the ARMAX identified models. Because if A_0, B_0 and C_0 are the solutions of

$$A_0(q^{-1})y(k) = B_0(q^{-1})u(k) + C_0(q^{-1})e(k)$$

  then $A = MA_0, B = MB_0$ and $C = MC_0$ will be also a solution of the ARMAX equation.
- Can we use ARX structure to verify zero/pole cancellation ?

Model order selection

Zero/Pole Cancellation

- The zero/pole map of each model can be inspected for possible existence of zero/pole cancellation.
- Each zero/pole cancellation indicates that the order of the model is overestimated by 1.
- In the presence of noise, the variance of the poles and zero and a confidence interval around each pole and zero should be computed.
- An intersection between the confidence interval of a pole and that of a zero indicates an overestimation of the model order.
- The model order is the maximum value of δ for which there is no zero/pole cancellation.

Question : Can we avoid cancellation of true poles and zeros which are close to each other (like resonance and anti-resonance modes in mechatronic systems) ?

Model order selection

Loss-function evolution :

Loss function is the mean value of the optimization criterion evaluated at the estimated value.

$$L_f(\delta, N) = \frac{1}{N} \sum_{k=1}^{N} \varepsilon^2(k, \hat{\theta})$$

- For ARX structure $L_f(\delta, N)$ is a monotonically nonincreasing function with respect to δ (why ?).
- Take $d = 1$ and $\delta = n = m$. Then, for $\delta = 1, \ldots, \delta_{\max}$ we identify a set of models.
- For $\delta > \delta_0$ (the true model order), the loss function will not change significantly. The over parameterization is used for modelling the realization of noise and not the model behaviour.
- By inspecting the evolution of the loss function, we can find a rough estimate of the model order at which the decrease of the loss function is not significant.

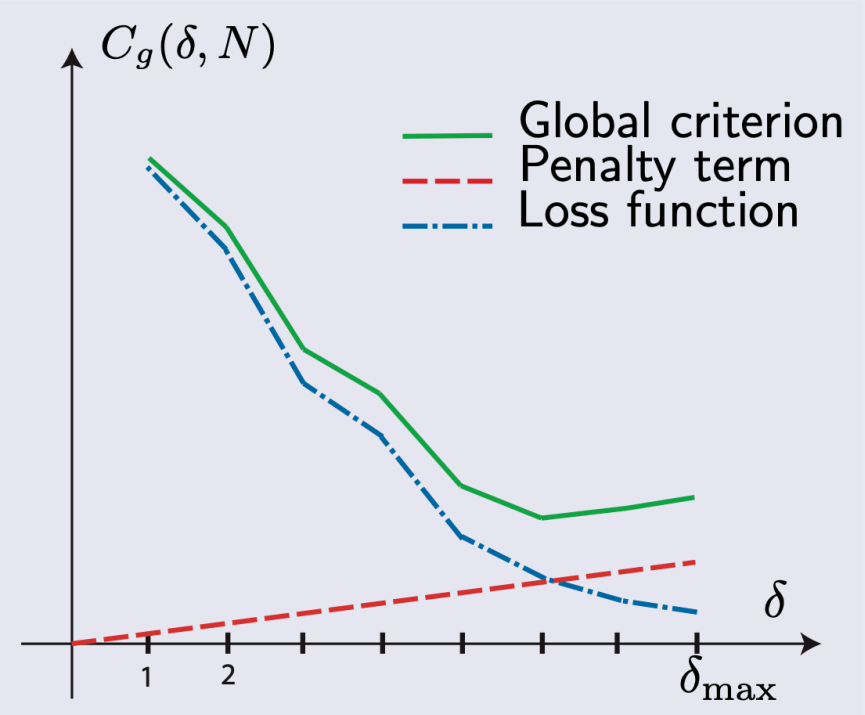
Model order selection

Loss-function evolution :

In order to have a quantitative criterion for model order selection, a penalty term can be added to the loss function :

$$C_g(\delta, N) = L_f(\delta, N) + S(\delta, N)$$

$$\begin{aligned} AIC &= L_f(\delta, N) + \frac{2\delta}{N} J(n) \\ BIC &= L_f(\delta, N) + \frac{\delta \log(N)}{N} \\ CIC &= L_f(\delta, N) + \frac{\delta \log^2(N)}{N} \\ FPE &= L_f(\delta, N) \frac{1 + \delta/N}{1 - \delta/N} \end{aligned}$$



These methods usually give an over estimation of the model order !

Structure selection

Estimation of time delay d :

- Time-delay can be estimated by an FIR model with $d = 0$ (or $d = 1$ since we know that $b_0 = 0$) and $m = m_{\max}$.
- The first coefficients of $B(q^{-1})$ which are close to zero (considering their standard deviations) represent the time delay. If b_k is zero, we have the following property with a probability of 0.95 :

$$0 \in [b_k - 2\sigma_k \, , \, b_k + 2\sigma_k]$$

- FIR is preferred because it is unbiased and use LS algorithm (global optimal solution).
- For oscillatory systems, where m is large, the variance of the parameters will be large so FIR will not be a good choice.
- Other structures like OE or ARMAX with order δ can also be used and the first coefficients of $B(q^{-1})$ inspected.

Structure selection

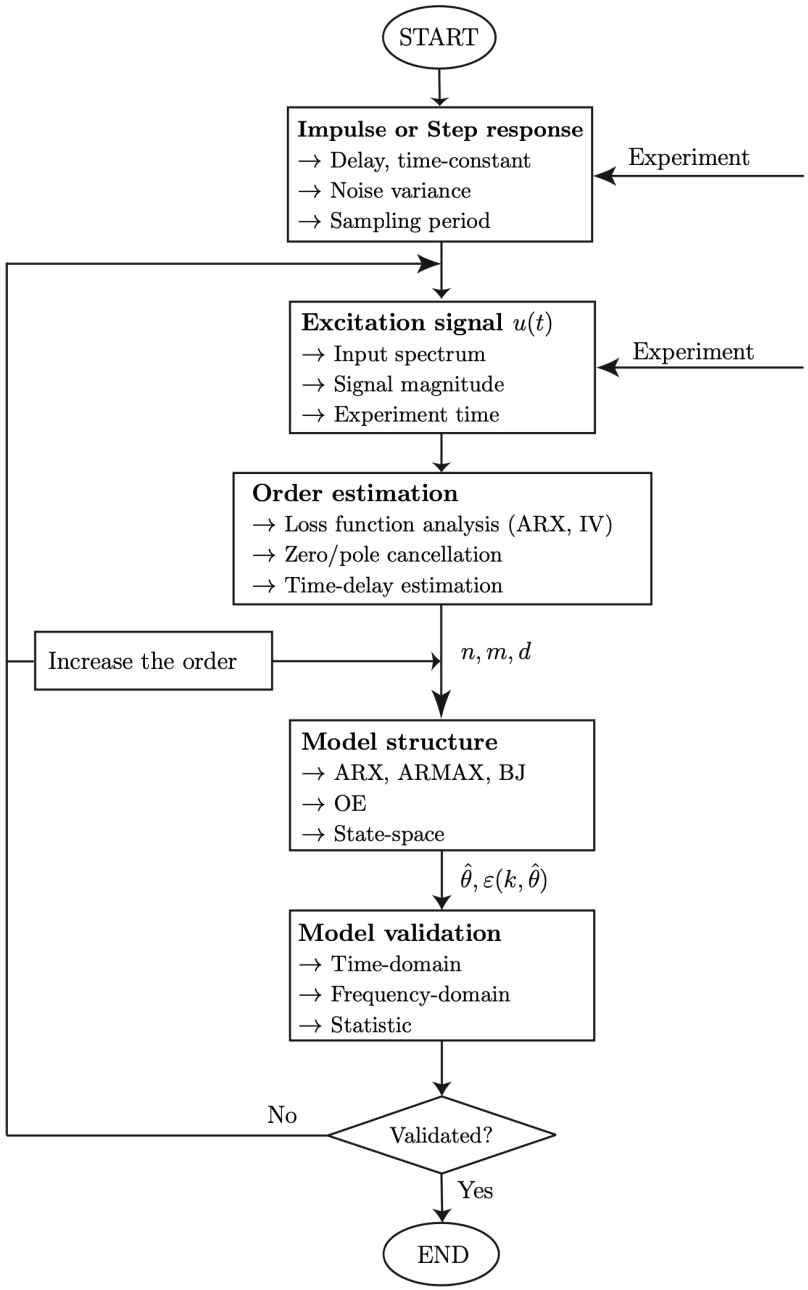
Estimation of n and m :

- Take $m = \delta$ and d equal to the estimated value and identify ARX models for $n = 1, \ldots, \delta$, and use the loss function method to estimate n.
- Take n and d equal to the estimated values and identify ARX models for $m = d, \ldots, \delta$, and use the loss function method to estimate m.
- Check the variance of a_n and b_m and compute $a_n \pm 2\sigma$ and $b_n \pm 2\sigma$. If zero belongs to these intervals, it shows an overestimation of n and m.
- In Matlab, n_A, n_B and d are estimated all together using the loss function method. First the following intervals are chosen :

$$n_A \in [n_{A_{\min}}\,,\, n_{A_{\max}}] \quad , \quad n_B \in [n_{B_{\min}}\,,\, n_{B_{\max}}] \quad , \quad d \in [d_{\min}\,,\, d_{\max}]$$

Then, a set of models concerning all combinations of values in the intervals are identified using the ARX structure. The number of parameters will increase from $n_{A\min} + n_{B_{\min}}$ to $n_{A_{\max}} + n_{B_{\max}}$. Next, the lowest loss function for each number of parameters is plotted versus the number of parameters in the model. From the evolution of the loss function, the number of parameters of numerator, denominator and d will be selected.

Identification Procedure



Optimization Algorithms

How to minimize the fit criterion ?

$$\hat{\theta} = \arg\min_{\theta} J(\theta) = \sum_{k=1}^{N} \varepsilon^2(k,\theta) = \sum_{k=1}^{N} [y(k) - \hat{y}(k,\theta)]^2$$

For FIR and ARX structures this leads to the linear LS algorithm. For other structures, nonlinear optimization algorithms should be used.

- **Pseudo linear regression algorithm :** Reformulate the output predictor as a pseudo linear regression (i.e. $\hat{y}(k) = \varphi^T(k,\theta)\theta$) and solve it iteratively.
- **Gauss-Newton algorithm :** Initialize θ with LS algorithm. Compute the gradient J' and Hessian J'' of the criterion. Use the following algorithm :

$$\hat{\theta}_{i+1} = \hat{\theta}_i - [J''(\hat{\theta}_i)]^{-1} J'(\hat{\theta}_i)$$

- **Recursive algorithms :** Solve the LS algorithm by a recursive formula. Appropriate for on-line parameter estimation.

Pseudo linear regression algorithm

Example (OE structure)

The output predictor is given by : $\hat{y}(k,\theta) = \dfrac{B(q^{-1})}{A(q^{-1})}u(k).$

It can be rewritten as :

$$\begin{aligned}\hat{y}(k,\theta) &= -a_1\hat{y}(k-1,\theta) - \cdots - a_n\hat{y}(k-n,\theta) \\ &\quad + b_d u(k-d) + \cdots + b_m u(k-m) = \phi^T(k,\theta)\theta\end{aligned}$$

where $\theta^T = [a_1, \ldots, a_n, b_d, \ldots, b_m]$ and

$$\phi^T(k,\theta) = [-\hat{y}(k-1,\theta), \ldots, -\hat{y}(k-n,\theta), u(k-d), \ldots, u(k-m)]$$

Then $$\hat{\theta}_{i+1} = \left[\sum_{k=1}^{N} \phi(k,\hat{\theta}_i)\phi^T(k,\hat{\theta}_i)\right]^{-1} \left[\sum_{k=1}^{N} \phi(k,\hat{\theta}_i)y(k)\right]$$

$$\hat{\theta}_i \quad \Rightarrow \quad \hat{y}(k,\hat{\theta}_i) = \frac{B(q^{-1},\hat{\theta}_i)}{A(q^{-1},\hat{\theta}_i)}u(k) \quad \Rightarrow \quad \phi(k,\hat{\theta}_i) \quad \Rightarrow \quad \hat{\theta}_{i+1}$$

Pseudo linear regression algorithm

Example (ARMAX structure)

The prediction error for the ARMAX structure can be rewritten as :

$$\begin{aligned}\varepsilon(k,\theta) &= \frac{1}{C(q^{-1})}\left[A(q^{-1})y(k) - B(q^{-1})u(k)\right] \\ &= y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) - b_d u(k-d) - \cdots \\ &\quad -b_m u(k-m) - c_1\varepsilon(k-1,\theta) - \cdots - c_{n_c}\varepsilon(k-n_c,\theta) \\ &= y(k) - \phi_x^T(k,\theta)\theta\end{aligned}$$

where $\theta^T = [a_1, \ldots, a_n, b_d, \ldots, b_m, c_1, \ldots, c_{n_c}]$ and

$$\begin{aligned}\phi_x^T(k,\theta) = [&-y(k-1), \ldots, -y(k-n), u(k-d), \ldots, u(k-m), \\ &\varepsilon(k-1,\theta), \ldots, \varepsilon(k-n_c,\theta)]\end{aligned}$$

$$\Rightarrow \hat{\theta}_{i+1} = \left[\sum_{k=1}^{N} \phi_x(k,\hat{\theta}_i)\phi_x^T(k,\hat{\theta}_i)\right]^{-1}\left[\sum_{k=1}^{N} \phi_x(k,\hat{\theta}_i)y(k)\right]$$

Optimization algorithms

Gauss-Newton algorithm

$$J(\theta) = \sum_{k=1}^{N} \varepsilon^2(k,\theta) = \sum_{k=1}^{N} [y(k) - \hat{y}(k,\theta)]^2 \quad \Rightarrow \quad \hat{\theta}_{i+1} = \hat{\theta}_i - [J''(\hat{\theta}_i)]^{-1} J'(\hat{\theta}_i)$$

Computing the gradient :

$$J'(\theta) = \frac{\partial J}{\partial \theta} = -2 \sum_{k=1}^{N} \frac{\partial \hat{y}}{\partial \theta} \varepsilon(k,\theta) = -2 \sum_{k=1}^{N} \psi(k,\theta) \varepsilon(k,\theta)$$

Computing the Hessian :

$$J''(\theta) = \frac{\partial^2 J}{\partial \theta \partial \theta^T} = 2 \sum_{k=1}^{N} \left[\psi(k,\theta)\psi^T(k,\theta) - \frac{\partial \psi}{\partial \theta} \varepsilon(k,\theta) \right]$$

$$\approx 2 \sum_{k=1}^{N} \psi(k,\theta)\psi^T(k,\theta)$$

where $\psi(k,\theta) \equiv \partial \hat{y} / \partial \theta$ should be computed for each model structure.

Gauss-Newton algorithm

Example (Compute $\psi(k,\theta)$ for the OE structure)

The output predictor is given by :

$$\hat{y}(k,\theta) = \frac{B(q^{-1})}{A(q^{-1})}u(k) = \frac{b_d q^{-d} + \cdots + b_m q^{-m}}{1 + a_1 q^{-1} + \cdots + a_n q^{-n}}u(k)$$

$$\frac{\partial \hat{y}}{\partial b_i} = \frac{q^{-i}}{A(q^{-1})}u(k) = \frac{1}{A(q^{-1})}u(k-i) \qquad i = d, \ldots, m$$

$$\frac{\partial \hat{y}}{\partial a_i} = \frac{-q^{-i}B(q^{-1})}{A^2(q^{-1})}u(k) = \frac{-1}{A(q^{-1})}\hat{y}(k-i) \qquad i = 1, \ldots, n$$

$$\psi^T(k,\theta) = \frac{1}{A(q^{-1})}[-\hat{y}(k-1,\theta), \ldots, -\hat{y}(k-n,\theta),$$
$$u(k-d), \ldots, u(k-m)] = \frac{1}{A(q^{-1})}\phi^T(k,\theta)$$

Asymptotic covariance of the parameter estimates

What is the covariance of the parameter estimates ?

- Assume that $\hat{\theta}$ converges to θ^* when N goes to infinity.
- Taylor expansion of $J'(\hat{\theta}) = 0$ around θ^* gives :

$$J'(\hat{\theta}) \approx J'(\theta^*) + J''(\theta^*)(\hat{\theta} - \theta^*) = 0$$

$$\Rightarrow \hat{\theta} - \theta^* = -[J''(\theta^*)]^{-1} J'(\theta^*)$$

- Then, $\text{cov}(\hat{\theta}) = \mathbb{E}\{(\hat{\theta} - \theta^*)(\hat{\theta} - \theta^*)^T\}$ is given by :

$$\text{cov}(\hat{\theta}) = \mathbb{E}\left\{[J''(\theta^*)]^{-1} J'(\theta^*) J'^T(\theta^*) [J''(\theta^*)]^{-1}\right\}$$

- If $\varepsilon(k, \theta^*)$ is white with variance σ_e^2, then under some mild assumptions $J'(\theta^*)J'^T(\theta^*) = \sigma_e^2 J''(\theta^*)$. Replacing θ^* with $\hat{\theta}$:

$$\text{cov}(\hat{\theta}) \approx \sigma_e^2 \left[\sum_{k=1}^{N} \psi(k, \hat{\theta}) \psi^T(k, \hat{\theta})\right]^{-1}$$

- Using the central limit theorem $\sqrt{N}(\hat{\theta} - \theta^*)$ has a zero-mean Gaussian distribution (for large N).

Recursive least squares algorithm

For on-line identification of time-varying systems

Parameter estimates at instant k :

$$\hat{\theta}_k = \left[\sum_{i=1}^{k} \phi(i)\phi^T(i)\right]^{-1} \sum_{i=1}^{k} \phi(i)y(i)$$

Problem : Too much computation at each sampling interval

Solution : Using recursive algorithm (compute $\hat{\theta}_{k+1}$ as a function of $\hat{\theta}_k$)

$$\hat{\theta}_k = P_k \sum_{i=1}^{k} \phi(i)y(i) \quad \text{where} \quad P_k = \left[\sum_{i=1}^{k} \phi(i)\phi^T(i)\right]^{-1}$$

$$\begin{aligned} P_{k+1}^{-1} &= \sum_{i=1}^{k+1} \phi(i)\phi^T(i) = \sum_{i=1}^{k} \phi(i)\phi^T(i) + \phi(k+1)\phi^T(k+1) \\ &= P_k^{-1} + \phi(k+1)\phi^T(k+1) \end{aligned}$$

Recursive least squares algorithm

$$
\begin{aligned}
\hat{\theta}_{k+1} &= P_{k+1}\sum_{i=1}^{k+1}\phi(i)y(i) = P_{k+1}\left[\sum_{i=1}^{k}\phi(i)y(i) + \phi(k+1)y(k+1)\right] \\
&= P_{k+1}\left[P_k^{-1}\hat{\theta}_k + \phi(k+1)y(k+1)\right] \\
&= P_{k+1}[P_{k+1}^{-1} - \phi(k+1)\phi^T(k+1)]\hat{\theta}_k + P_{k+1}\phi(k+1)y(k+1) \\
&= \hat{\theta}_k + P_{k+1}\phi(k+1)\left[y(k+1) - \phi^T(k+1)\hat{\theta}_k\right] \\
&= \hat{\theta}_k + P_{k+1}\phi(k+1)\varepsilon(k+1)
\end{aligned}
$$

Matrix inversion lemma

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B[C^{-1} + DA^{-1}B]^{-1}DA^{-1}$$

To find the inverse of $P_{k+1}^{-1} = P_k^{-1} + \phi(k+1)\phi^T(k+1)$, take $A = P_k^{-1}$, $B = \phi(k+1)$, $C = 1$, $D = \phi^T(k+1)$ which leads to :

$$P_{k+1} = P_k - \frac{P_k\phi(k+1)\phi^T(k+1)P_k}{1 + \phi^T(k+1)P_k\phi(k+1)}$$

Weighted least squares algorithm

Suppose that errors in different instants have different importance (e.g. old errors has less importance in time-varying systems).
Weighted error is defined as :

$$\mathcal{E}_W \equiv W[Y - \Phi\theta]$$

W is a weighting matrix (usually diagonal). The criterion becomes :

$$J(\theta) = \mathcal{E}_W^T \mathcal{E}_W = \mathcal{E}^T W^T W \mathcal{E} = [Y - \Phi\theta]^T W^T W [Y - \Phi\theta]$$

and the vector of parameters :

$$\hat{\theta} = (\Phi^T W^T W \Phi)^{-1} \Phi^T W^T W Y$$

$$W^T W = \text{diag}(\lambda^{N-1}, \lambda^{N-2}, \ldots, \lambda^1, \lambda^0) \qquad 0.9 \leq \lambda \leq 0.99$$

The last error is weighted by $\lambda^0 = 1$ and the first error by $\lambda^{N-1} \approx 0$.

λ is called forgetting factor (facteur d'oubli).

Recursive weighted least squares

$$\hat{\theta}_k = \left[\sum_{i=1}^{k} \phi(i)\lambda^{k-i}\phi^T(i)\right]^{-1} \sum_{i=1}^{k} \phi(i)\lambda^{k-i}y(i) = P_k \sum_{i=1}^{k} \phi(i)\lambda^{k-i}y(i)$$

$$P_{k+1}^{-1} = \lambda P_k^{-1} + \phi(k+1)\phi^T(k+1)$$

- For $\lambda = 1$ (without forgetting factor) the trace of adaptation gain P_{k+1} goes to zero when k goes to infinity (the algorithm becomes insensitive to parameter variations).
- For $\lambda < 1$, the trace of P_{k+1} does not converge to zero and the algorithm remains alive w.r.t parameter variations.

The recursive algorithm using the matrix inversion lemma is given by :

$$\begin{aligned} P_{k+1} &= \frac{1}{\lambda}\left[P_k - \frac{P_k\phi(k+1)\phi^T(k+1)P_k}{\lambda + \phi^T(k+1)P_k\phi(k+1)}\right] \\ \hat{\theta}_{k+1} &= \hat{\theta}_k + P_{k+1}\phi(k+1)[y(k+1) - \phi^T(k+1)\hat{\theta}_k] \end{aligned}$$

Recursive least squares algorithm

Initialization : There are two ways for initializing the algorithm :

1. The initial values are fixed a priori. In general, $\hat{\theta}_0 = 0$ and $P_0 = \alpha I$, where α is a large value multiplied by the number of parameters p, say $\alpha = 1000p$, and I is the unity matrix. Because P_0 is an initial estimate of the covariance matrix of the parameters. Since the initial value is far from the true one, a large covariance matrix is chosen.
2. The recursion starts after p sampling period. At iteration p, $\hat{\theta}_p$ is estimated by solving a system of linear equation as :

$$\hat{\theta}_p = \Phi_p^{-1} Y_p \quad ; \quad P_p = [\Phi_p^T \Phi_p]^{-1}$$

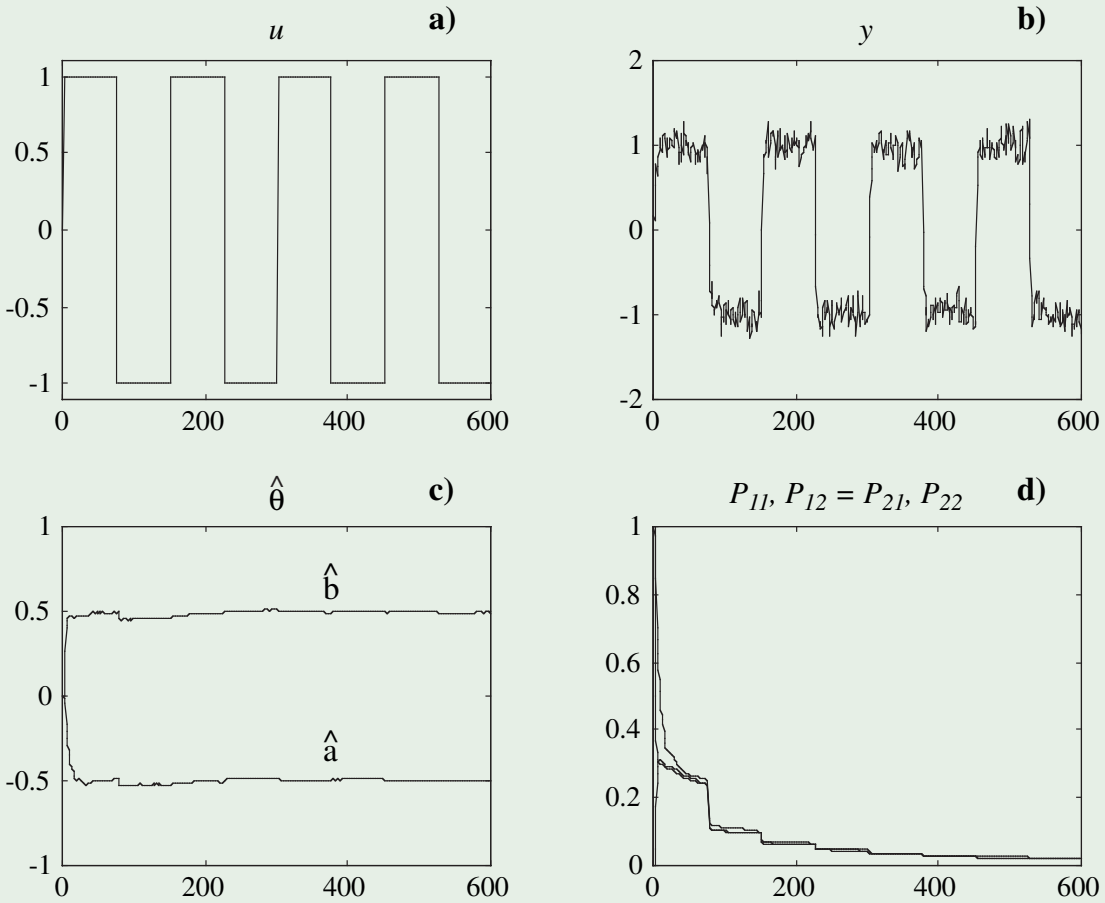
where

$$\Phi_p = \begin{bmatrix} \phi^T(1) \\ \vdots \\ \phi^T(p) \end{bmatrix}$$

and $Y_p = [y(1), \ldots, y(p)]^T$.

Example (Time invariant system)

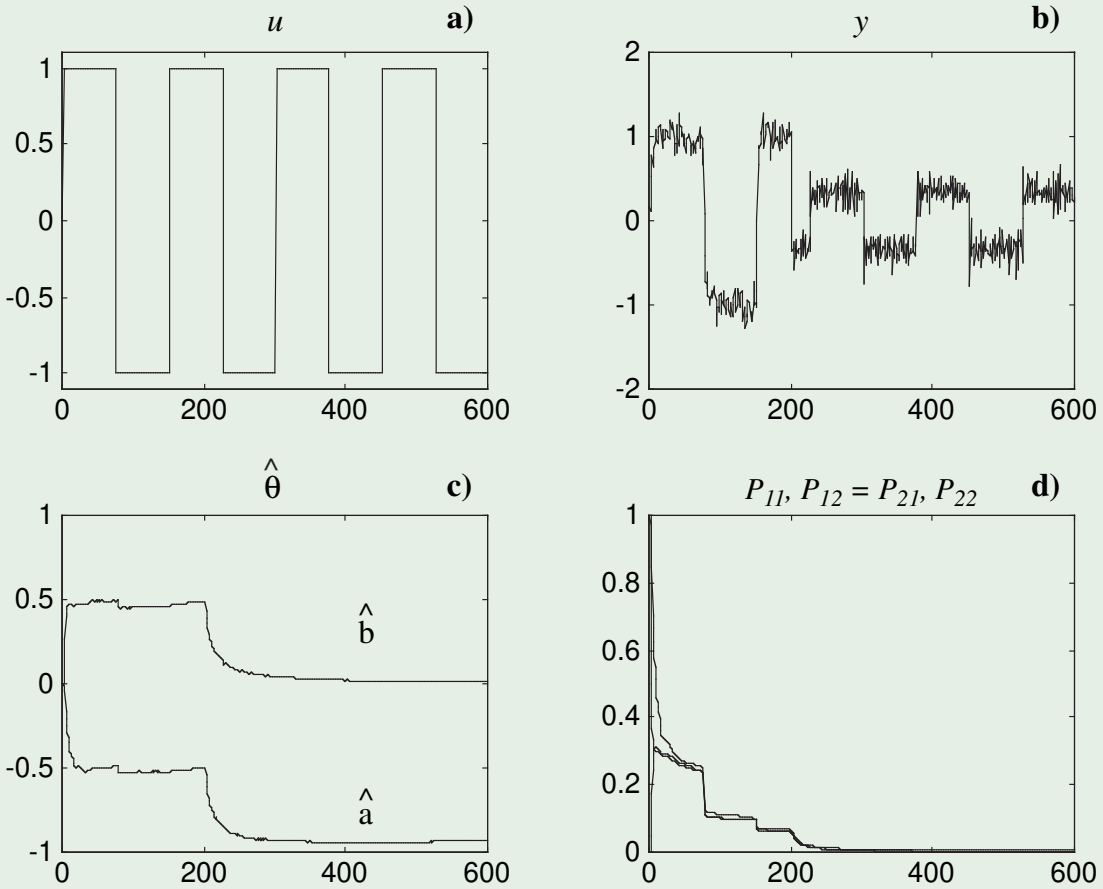
$$y(k) + a_1^\circ y(k-1) = b_1^\circ u(k-1) + e_0(k) \qquad \theta_0 = [a_1^\circ \; b_1^\circ]^T = [-0.5 \; 0.5]^T$$



Recursive least squares

Example (Time variant system (without forgetting factor))

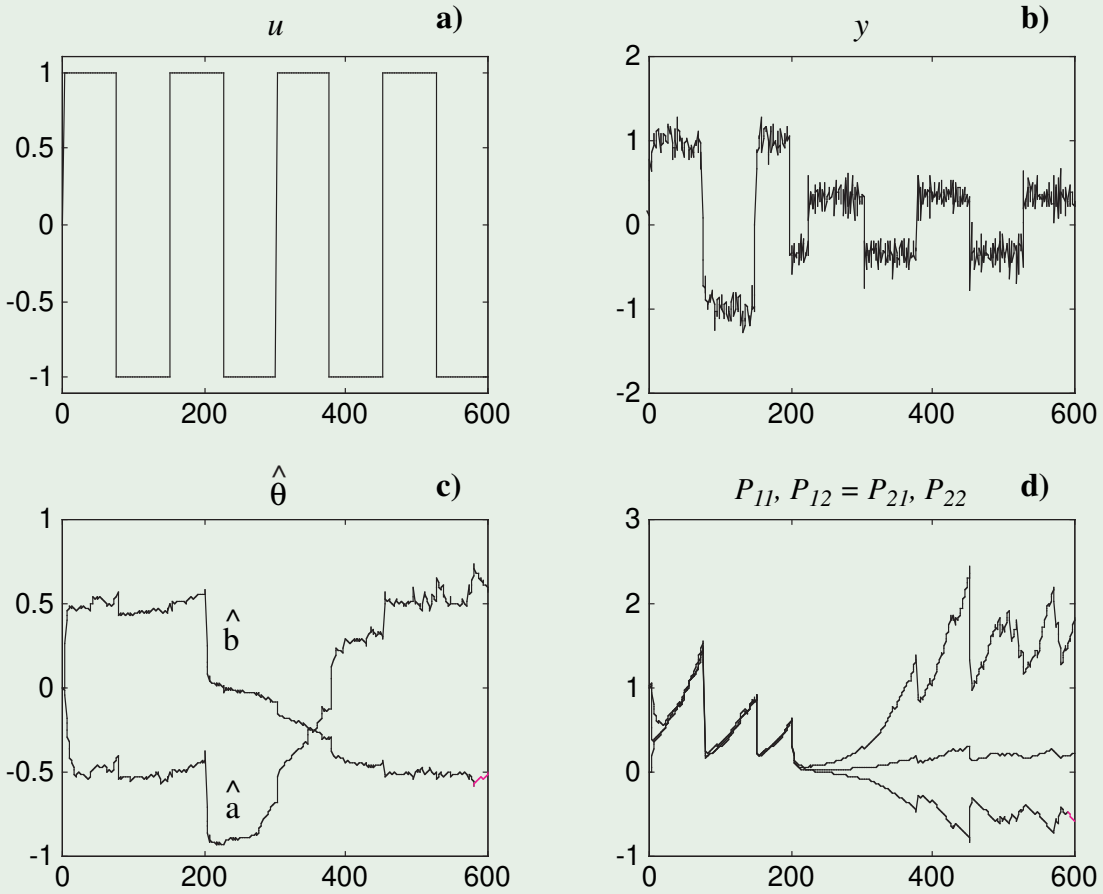
$$\theta_0 = \begin{cases} [-0.5\ 0.5]^T & k < 200 \\ [0.5\ -0.5]^T & k \geq 200 \end{cases} \quad \text{pour}$$



Recursive weighted least squares

Example (Time variant system (with forgetting factor $\lambda = 0.97$))

$$\theta_0 = \begin{cases} [-0.5 \ 0.5]^T & \text{pour} \quad k < 200 \\ [0.5 \ -0.5]^T & \text{pour} \quad k \geq 200 \end{cases}$$



Identification of Nonlinear Systems

Grey-box Identification

In this approach, some physical insight is available (e.g. a first principle model). Then the model parameters are identified by minimizing the prediction error. The prediction error may be linear in parameters that leads to a LS problem or it is nonlinear that leads to a nonlinear optimization method.

Black-box Identification

In this approach, no physical insight is available. Then we choose a model structure which has good flexibility to cover approximately a large class of nonlinear behaviour. A typical choice is a linear combination of basis functions that can approximate any nonlinear function when the number of bases increases.

Grey-box Identification

Example (Linear regression problem)

consider a nonlinear model given by :

$$y(k) = \alpha_0 u(k-1)y(k-1) + \beta_0 y^2(k-2) + n(k)$$

The output predictor is given by : $\hat{y}(k) = \phi^T(k)\theta$ where

$$\phi^T(k) = [u(k-1)y(k-1) \quad y^2(k-2)] \quad , \quad \theta^T = [\alpha \quad \beta]$$

Then the system parameters can be identified by the least squares method

- If $\hat{\theta} = \theta_0$, the residuals will be equal to $n(k)$ (uncorrelated with $u(k)$). So the uncorrelation of the residuals and $u(k)$ can be used as a validation test.
- A noise model can be considered with $n(k) = H(q^{-1})e(k)$. In this case the predictor will be

$$\hat{y}(k) = \alpha u(k-1)y(k-1) + \beta y^2(k-2) + [H(q^{-1}) - 1]\varepsilon(k)$$

and whiteness of the residuals can be used for the noise model validation.

Grey-box Identification

Identification of robotic arms : The model of a robotic arm using the Euler-Lagrange method is given by :

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau$$

n	Number of joints
$q \in \mathbb{R}^{n\times 1}$	Joint angles
$M(q) \in \mathbb{R}^{n\times n}$	Inertia matrix
$C(q,\dot{q}) \in \mathbb{R}^{n\times n}$	Coriolis matrix
$G(q) \in \mathbb{R}^{n\times 1}$	Gravity vector
$\tau \in \mathbb{R}^{n\times 1}$	Torque vector

Procedure : The excitation signals are added to the joint torques and the joint angles q, joint speeds $\dot{q}$ and joint accelerations $\ddot{q}$ are measured. Then the parameters of the matrices M, C and G can be identified with the least squares algorithm.

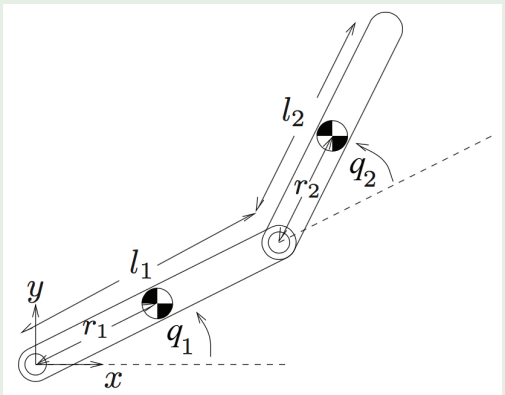
Example (Identification of a two-link planar robot)

The dynamic model of the system can be obtained as :

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

$$M_{11} = I_1 + I_2 + m_1 r_1^2 + m_2(l_1^2 + r_2^2) + 2m_2 l_1 r_2 \cos(q_2)$$
$$M_{12} = M_{21} = I_2 + m_2 r_2^2 + m_2 l_1 r_2 \cos(q_2)$$
$$M_{22} = I_2 + m_2 r_2^2$$

$$C_{11} = -\dot{q}_2 m_2 l_1 r_2 \sin(q_2)$$
$$C_{12} = -(\dot{q}_1 + \dot{q}_2) m_2 l_1 r_2 \sin(q_2)$$
$$C_{21} = \dot{q}_1 m_2 l_1 r_2 \sin(q_2)$$
$$C_{22} = 0$$



$$\alpha = I_1 + I_2 + m_1 r_1^2 + m_2(l_1^2 + r_2^2), \qquad \beta = m_2 l_1 r_2, \qquad \gamma = I_2 + m_2 r_2^2$$

Grey-box Identification

Example (Identification of a two-link planar robot)

Therefore, the model in linear regression form will be :

$$\begin{bmatrix} \phi_{11}(k) & \phi_{12}(k) & \phi_{13}(k) \\ 0 & \phi_{22}(k) & \phi_{23}(k) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} \tau_1(k) \\ \tau_2(k) \end{bmatrix}$$

where for each instant k :

$$\phi_{11} = \ddot{q}_1 \qquad \phi_{12} = [2\ddot{q}_1 + \ddot{q}_2]\cos(q_2) - [\dot{q}_2\dot{q}_1 + \dot{q}_1^2 + \dot{q}_2^2]\sin(q_2)$$

$$\phi_{13} = \ddot{q}_2 \qquad \phi_{22} = \ddot{q}_1\cos(q_2) + \dot{q}_1^2\sin(q_2) \qquad \phi_{23} = \ddot{q}_1 + \ddot{q}_2$$

Remarks :

- LS algorithm can be used to identify α, β and γ and to compute M and C that can be used in simulation or for controller design.
- The physical parameters are not identifiable. However, if we know some of them we can find the others.

Black-box Identification

For any nonlinear system the output predictor can be given by :

$$\hat{y}(k, \theta) = F(\phi(k), \theta)$$

where F is some nonlinear function of θ and $\phi(k)$.

Similar to the black-box models for linear systems, we can define :

NFIR : The regressor vector will use only past inputs $u(k-l), l > 0$.

NARX : The regressor will use past inputs $u(k-l)$ and past outputs $y(k-l)$.

NOE : The regressor will use past inputs $u(k-l)$ and past predicted output $\hat{y}(k-l, \theta)$.

NARMAX : The regressor will use past inputs $u(k-l)$, past outputs $y(k-l)$ and past prediction errors $\varepsilon(k-l, \theta)$.

Black-box Identification

Basis functions : $F(\phi(k), \theta)$ can be well approximated using some basis functions :

$$F(\phi(k), \theta) = \sum_{i=1}^{n} \theta_i F_i(\phi(k))$$

Bias-Variance Trade-off : When $n \to \infty$ any nonlinear function is approximated with the basis functions but increasing n increases the variance of the parameters.

Choice of F_i : It can be chosen using a single parameterized mother basis function denoted by $P(x)$.

Mother basis functions : Typical functions are

- piecewise-constant pulse function
- Gaussian function
- Sigmoid function

Black-box Identification

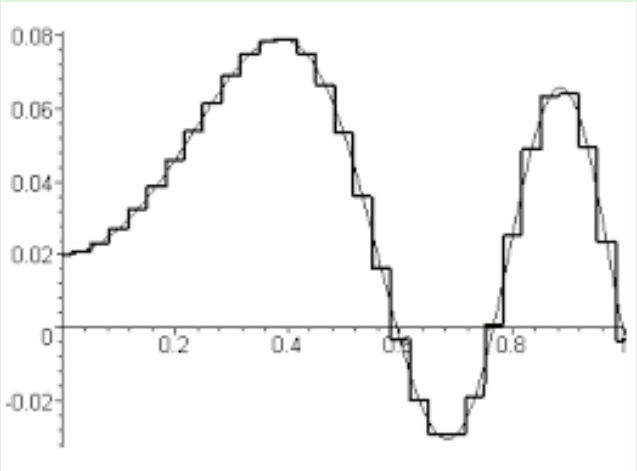
Example (Piecewise-constant pulse function)

$$P(x) = \begin{cases} 1 & \text{for } 0 \leq x < \Delta \\ 0 & \text{otherwise} \end{cases}$$

Any nonlinear function $F(x)$ can be approximated by :

$$F(x) = \sum_{i=0}^{\infty} \theta_i P(x - i\Delta)$$

where $\theta_i = F(i\Delta)$ and $F_i(x) = P(x - i\Delta)$.



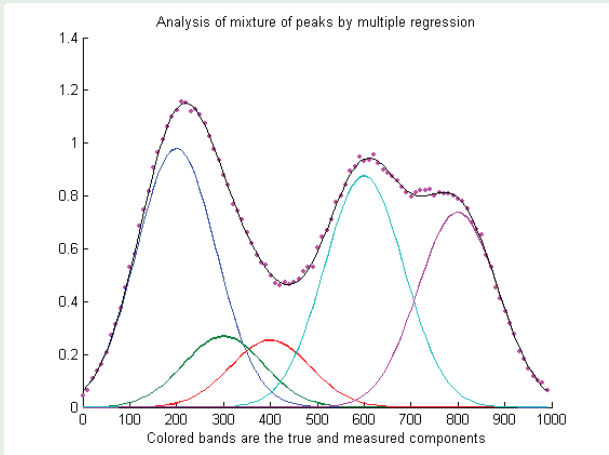
Example (Gaussian function)

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Any nonlinear function $F(x)$ can be approximated by :

$$F(x) = \sum_{i=1}^{n} \theta_i P(x - \mu_i)$$

The parameters of the Gaussian function $\beta_i = [\mu_i \quad \sigma_i]$ should be chosen a priori or be optimized together with θ.



Black-box Identification

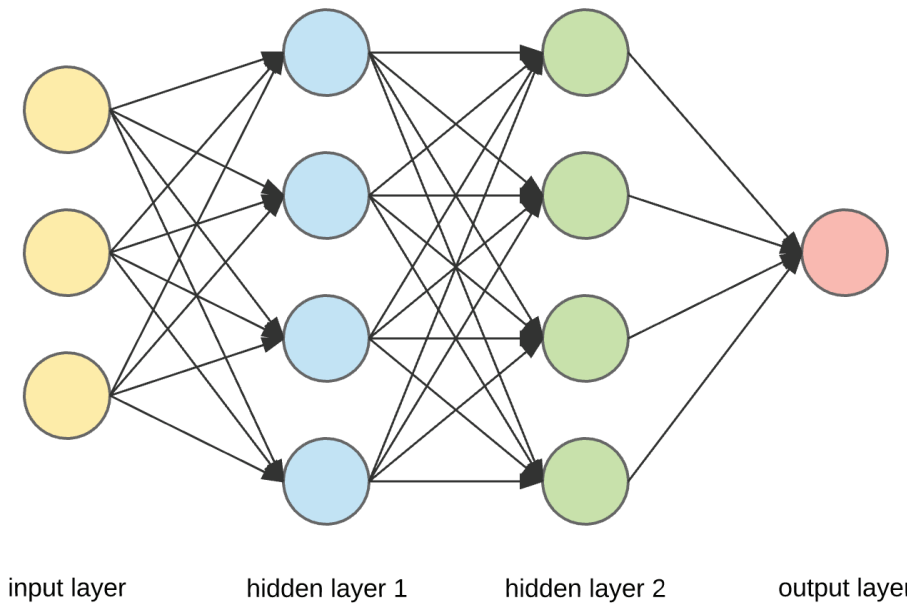
- The model parameters are obtained as :

$$\theta = \arg\min_{\theta,\beta} \sum_{k=1}^{N} \|y(k) - \sum_{i=1}^{n} \theta_i F_i(\phi(k), \beta)\|$$

- $\phi(k)$ is chosen based on some a priori knowledge about the system. If no information is available a high dimension $\phi(k)$ may be considered that complicates the optimization problem.
- A basis function like Gaussian function, wavelet, sigmoid, etc should be chosen.
- If the parameters of the basis function are fixed (defined by user), the optimization becomes a least squares problem, otherwise a nonlinear, gradient-based, numerical optimization should be solved.
- In order to consider the bias-variance trade-off a new term $\lambda\|\theta\|$ can be added to the fit criterion, such that bias and variance are minimized together.

Black-box Identification

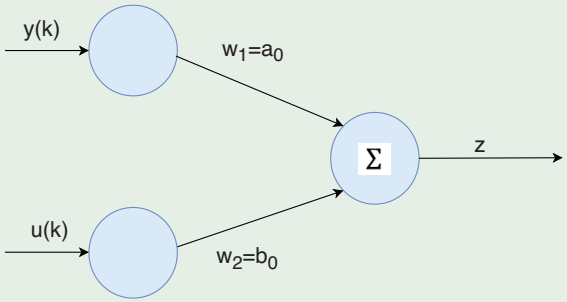
Neural Network models : Neural network is a very general black-box model structure based on convolving basis function expansions. It consists of input layer, hidden layers and output layer. Each layer includes some nodes. The output of each node is a function (called *activation function*) of the sum of its inputs.



Neural Network models can be used for modelling of very complex nonlinear systems.

Example (ARX model)

Consider the following simple neural network model with no hidden layer :



The output of the NN model is :

$$z = w_1 y(k) + w_2 u(k)$$

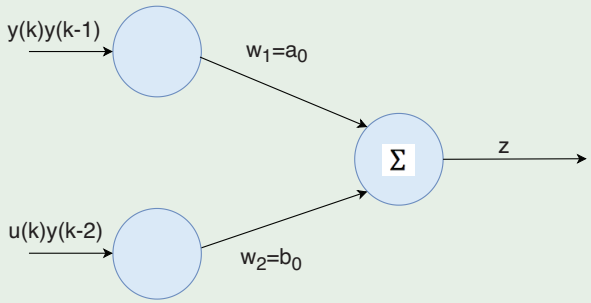
which corresponds to a simple ARX model with

$$y(k+1) = a_0 y(k) + b_0 u(k)$$

if we take $z = y(k+1), w_1 = a_0$ and $w_2 = b_0$.

Example (NARX model)

Consider the following simple neural network model with no hidden layer :



The output of the NN model is :

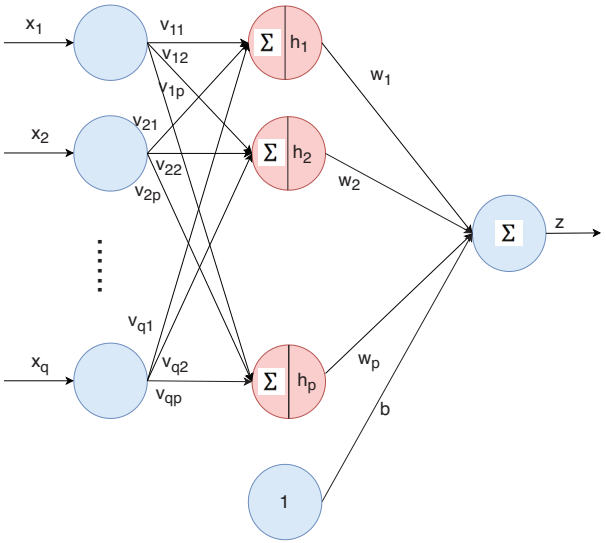
$$z = w_1 y(k)y(k-1) + w_2 u(k)y(k-2)$$

which corresponds to a simple NARX model with

$$y(k+1) = a_0 y(k)y(k-1) + b_0 u(k)y(k-2)$$

if we take $z = y(k+1), w_1 = a_0$ and $w_2 = b_0$.

Feedforward Neural Network :



$$x_i \in \{y(k), \cdots, y(k-n), u(k), \cdots, u(k-m), y(k)^2, \cdots, y(k-n)^2, u(k)^2, \cdots, u(k-m)^2, y(k)u(k), \cdots, y(k-n)u(k-m)\}$$

$$s_i = \sum_{j=1}^{q} x_j v_{ji} \quad (v_{ji} \text{ are known weights})$$

$$W = [w_1, \cdots, w_p, b]^T$$

$$H = [h_1, \cdots, h_p, 1]^T$$

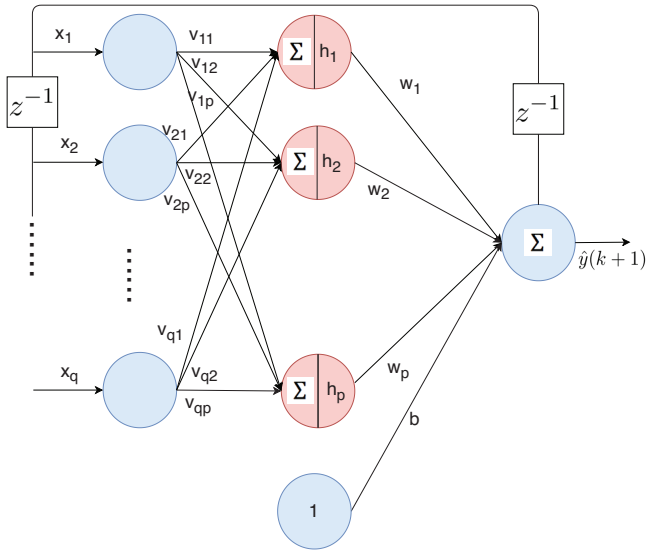
$$z = \sum_{i=1}^{p} w_i h_i(s_i) + b = W^T H$$

h_i is the basis function and

$$W = \left(\sum_{k=1}^{N} H_k H_k^T \right)^{-1} \sum_{k=1}^{N} H_k y(k+1)$$

Neural Network Models

Recurrent Neural Network : Instead of using input and output data (u, y) as in feedforward NN, input and predicted output $(u, \hat{y})$ data are used in NN :



$$
\begin{aligned}
x_i \in \{&\hat{y}(k), \cdots, \hat{y}(k-n), u(k), \cdots, \\
&u(k-m), \hat{y}(k)^2, \cdots, \hat{y}(k-n)^2, \\
&u(k)^2, \cdots, u(k-m)^2, \hat{y}(k)u(k), \\
&\cdots, \hat{y}(k-n)u(k-m)\}
\end{aligned}
$$

$$\hat{y}(k+1) = W^T H_k(W), \text{NOE !}$$

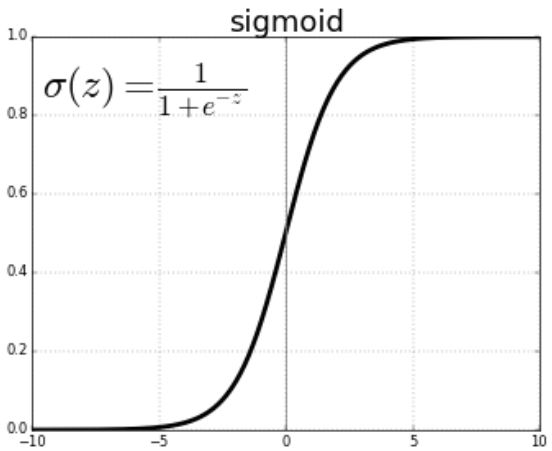
Iterative Gauss-Newton solution :

$$\hat{W}_{i+1} = \hat{W}_i + \gamma \left(\sum_{k=1}^{N} H_k(\hat{W}_i) H_k^T(\hat{W}_i) \right)^{-1} \sum_{k=1}^{N} H_k(\hat{W}_i) \varepsilon_k$$

Activation Functions : Typical activation functions are :

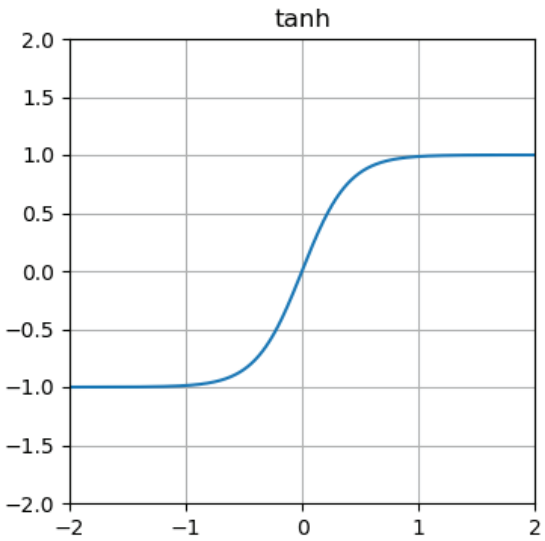
sigmoid function

$$f_{\mathsf{s}} = \frac{1}{1+e^{-x}}$$



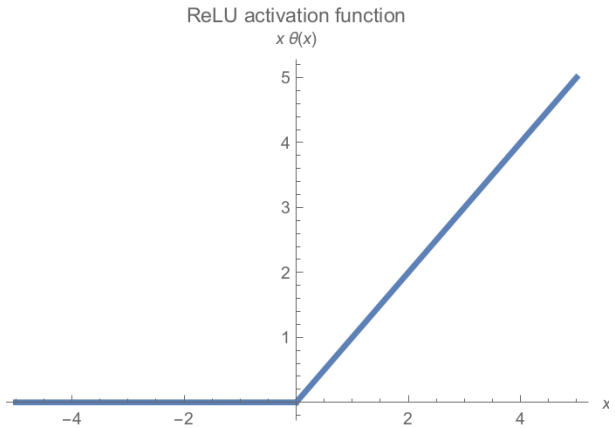
tanh function

$$f_{\mathsf{t}} = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

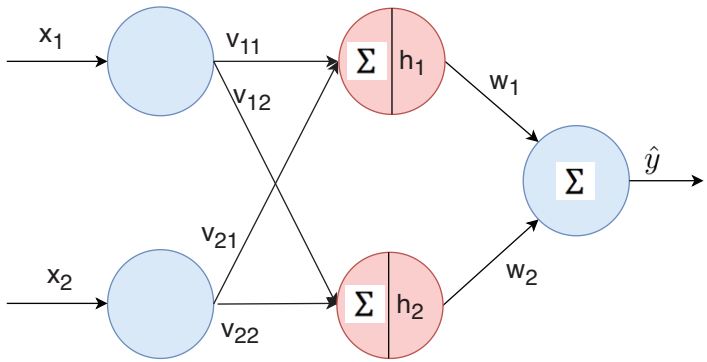


Rectified linear activation unit (ReLU)

$$f_{\mathsf{R}} = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{if } x \geq 0 \end{cases}$$



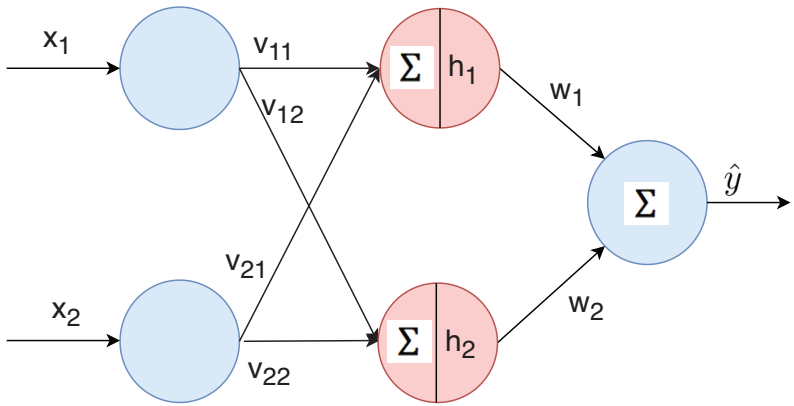
NN training using back propagation (gradient descent) algorithm : Optimizing W



$$s_1 = \hat{v}_{11}x_1 + \hat{v}_{21}x_2$$
$$s_2 = \hat{v}_{12}x_1 + \hat{v}_{22}x_2$$
$$\hat{y} = \hat{w}_1 h_1(s_1) + \hat{w}_2 h_2(s_2) = \hat{W}^T H$$
$$h(s) = \frac{1}{1+e^{-s}}, \frac{\partial h}{\partial s} = h(s)(1-h(s))$$

$$E = \frac{1}{2}\varepsilon^2 = \frac{1}{2}(y-\hat{y})^2$$
$$\frac{\partial E}{\partial \hat{y}} = -\varepsilon, \frac{\partial \hat{y}}{\partial \hat{w}_1} = h_1(s_1)$$
$$\frac{\partial E}{\partial \hat{w}_1} = \frac{\partial E}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial \hat{w}_1} = -h_1(s_1)\varepsilon$$
$$\frac{\partial E}{\partial \hat{w}_2} = -h_2(s_2)\varepsilon$$
$$\frac{\partial E}{\partial \hat{W}} = -H\varepsilon$$
$$\hat{W}_{i+1} = \hat{W}_i - \gamma\frac{\partial E}{\partial \hat{W}} = \hat{W}_i + \gamma H\varepsilon$$

NN training using back propagation (gradient descent) algorithm : Optimizing V



$$s_1 = \hat{v}_{11}x_1 + \hat{v}_{21}x_2$$
$$s_2 = \hat{v}_{12}x_1 + \hat{v}_{22}x_2$$
$$\hat{y} = \hat{w}_1 h_1(s_1) + \hat{w}_2 h_2(s_2) = \hat{W}^T H$$
$$V = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}, D_H = \begin{bmatrix} \frac{\partial h_1}{\partial s_1} & \frac{\partial h_2}{\partial s_2} \end{bmatrix}^T$$

$$E = \frac{1}{2}\varepsilon^2 = \frac{1}{2}(y - \hat{y})^2$$
$$\frac{\partial E}{\partial \hat{y}} = -\varepsilon, \frac{\partial \hat{y}}{\partial h_1} = \hat{w}_1, \frac{\partial s_1}{\partial \hat{v}_{11}} = x_1$$
$$\frac{\partial E}{\partial \hat{v}_{11}} = \frac{\partial E}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial h_1}\frac{\partial h_1}{\partial s_1}\frac{\partial s_1}{\partial \hat{v}_{11}} = -x_1\hat{w}_1(\partial h_1/\partial s_1)\varepsilon$$
$$\partial E/\partial \hat{v}_{21} = -x_2\hat{w}_1(\partial h_1/\partial s_1)\varepsilon$$
$$\partial E/\partial \hat{v}_{12} = -x_1\hat{w}_2(\partial h_2/\partial s_2)\varepsilon$$
$$\partial E/\partial \hat{v}_{22} = -x_2\hat{w}_2(\partial h_2/\partial s_2)\varepsilon$$
$$\partial E/\partial V = -X(\hat{W} \circ D_H)^T\varepsilon$$
$$V_{i+1} = V_i - \gamma\frac{\partial E}{\partial V} = V_i + \gamma X(\hat{W} \circ D_H)^T\varepsilon$$

Neural Network Models

Some basic terminology : Given the number of input/out data N, the prediction error criterion can be defined as :

$$E = \frac{1}{2}\sum_{k=1}^{m}\varepsilon_k^2 = \frac{1}{2}\sum_{k=1}^{m}(y(k) - \hat{y}(k))^2$$

- An epoch is one complete training pass over the whole data set (all weights are converged).
- Batch : When divide data set into number of sets or parts, each set or part is a batch ; A batch is used for one gradient update.
- Iteration An iteration consists of updating the gradients on a single batch of data ;
- Batch size : Total number of data, m, present in a single batch ;
  - Batch gradient descent : $m = N$, smooth convergence ;
  - Stochastic gradient descent (SGD) : $m = 1$, fast speed for large data ;
  - Mini-batch gradient descent : $1 < m < N$;