# Apprentissage de modèles dynamiques

Marion Gilson
marion.gilson@univ-lorraine.fr

CRAN, Université de Lorraine – Polytech Nancy – France

# Outline

# Méthodes avancées d'identification des systèmes

## The players - Rappel

Système à identifier $G_0$


- $u(t)$ est l'entrée (à temps discret) qui peut être choisie librement (sauf dans des cas type environnement)

- $y(t)$ est la sortie (à temps discret) qui est mesurée et provient de :
    - une contribution de l'entrée $u(t)$, c'est à dire $G_0 u(t)$
    - une contribution indépendante de l'entrée $u(t)$, c'est à dire des perturbations $v(t)$.

## The players - Rappel

Système à identifier $G_0$


- $u(t)$ est l'entrée (à temps discret) qui peut être choisie librement (sauf dans des cas type environnement)

- $y(t)$ est la sortie (à temps discret) qui est mesurée et provient de :
  - une contribution de l'entrée $u(t)$, c'est à dire $G_0 u(t)$
  - une contribution indépendante de l'entrée $u(t)$, c'est à dire des perturbations $v(t)$.

## The players - Rappel

Système à identifier $G_0$


- Le signal $v(t)$ est une *perturbation inconnue* : bruit, perturbation, effets des entrées non mesurées,...
- $v(t)$ ne sera jamais identique dans le cas d'expériences répétées
  $\longrightarrow$ la meilleure estimation : processus stochastique à moyenne nulle
- *Challenge* de l'identification : gestion des perturbations inconnues
  - si $v(t) = 0$, alors pas de problème, juste un jeu algébrique pour trouver la relation entre $y(t)$ et $u(t)$
  - sinon l'identification fournit (généralement) un modèle du processus représenté par $G_0$ mais également un modèle des perturbations $v(t)$

# Méthodes avancées d'identification des systèmes

## Rappel

Avant de se lancer dans une méthode avancée (procédure complète), il est intéressant de connaître une approximation du système à étudier
$\longrightarrow$ Obtenir un modèle de comportement par les méthodes usuelles (Broïda, Second ordre...)

**Avantages :**

- simples à mettre en œuvre
- rapides
- obtention d'un modèle de comportement simplifié, permettant l'initialisation d'algorithmes plus avancés

**Inconvénients :**

- modèles simplistes (parfois trop)
- pas robustes aux perturbations (bruit sur les données, perturbations extérieures,. . .
- non prises en compte de toutes les données

Solution : utiliser des méthodes avancées d'identification de systèmes.

# Plan

# Choix du signal d'excitation

## Principes

Pour identifier à partir des données, il est nécessaire que l'information soit contenue dans les données !
$\longrightarrow$ excitation dans tout le spectre de fréquence susceptible de contenir les constantes de temps du système

- $sin(\omega t)$ : balayage en fréquence, parfait d'un point de vue spectre, mais pas toujours accepté par tous les systèmes
- $\delta(t)$ : parfait d'un point de vue théorique, difficile voire impossible à réaliser en pratique
- $b(t)$ : bruit blanc idéal d'un point de vue spectral mais réalisation pratique ?
- SBPA : signal binaire pseudo aléatoire : réalisation pratique d'un signal aléatoire (en anglais : Pseudo Random Binary Sequence, PRBS).
- parfois impossible d'exciter un système (systèmes environnementaux, machine en production par ex.) : utiulisation des commandes naturelles du système comme signal d'entrée.

# Choix du signal d'excitation

## Principes

Pour identifier à partir des données, il est nécessaire que l'information soit contenue dans les données !
$\longrightarrow$ excitation dans tout le spectre de fréquence susceptible de contenir les constantes de temps du système

- $sin(\omega t)$ : balayage en fréquence, parfait d'un point de vue spectre, mais pas toujours accepté par tous les systèmes
- $\delta(t)$ : parfait d'un point de vue théorique, difficile voire impossible à réaliser en pratique
- $b(t)$ : bruit blanc idéal d'un point de vue spectral mais réalisation pratique ?
- SBPA : signal binaire pseudo aléatoire : réalisation pratique d'un signal aléatoire (en anglais : Pseudo Random Binary Sequence, PRBS).
- parfois impossible d'exciter un système (systèmes environnementaux, machine en production par ex.) : utiulisation des commandes naturelles du système comme signal d'entrée.

# Choix du signal d'excitation

## Principes

Pour identifier à partir des données, il est nécessaire que l'information soit contenue dans les données !
$\longrightarrow$ excitation dans tout le spectre de fréquence susceptible de contenir les constantes de temps du système

- $sin(\omega t)$ : balayage en fréquence, parfait d'un point de vue spectre, mais pas toujours accepté par tous les systèmes
- $\delta(t)$ : parfait d'un point de vue théorique, difficile voire impossible à réaliser en pratique
- $b(t)$ : bruit blanc idéal d'un point de vue spectral mais réalisation pratique ?
- SBPA : signal binaire pseudo aléatoire : réalisation pratique d'un signal aléatoire (en anglais : Pseudo Random Binary Sequence, PRBS).
- parfois impossible d'exciter un système (systèmes environnementaux, machine en production par ex.) : utiulisation des commandes naturelles du système comme signal d'entrée.

# Choix du signal d'excitation

## Principes

Pour identifier à partir des données, il est nécessaire que l'information soit contenue dans les données !
$\longrightarrow$ excitation dans tout le spectre de fréquence susceptible de contenir les constantes de temps du système

- $sin(\omega t)$ : balayage en fréquence, parfait d'un point de vue spectre, mais pas toujours accepté par tous les systèmes
- $\delta(t)$ : parfait d'un point de vue théorique, difficile voire impossible à réaliser en pratique
- $b(t)$ : bruit blanc idéal d'un point de vue spectral mais réalisation pratique ?
- SBPA : signal binaire pseudo aléatoire : réalisation pratique d'un signal aléatoire (en anglais : Pseudo Random Binary Sequence, PRBS).
- parfois impossible d'exciter un système (systèmes environnementaux, machine en production par ex.) : utiulisation des commandes naturelles du système comme signal d'entrée.

# Choix du signal d'excitation

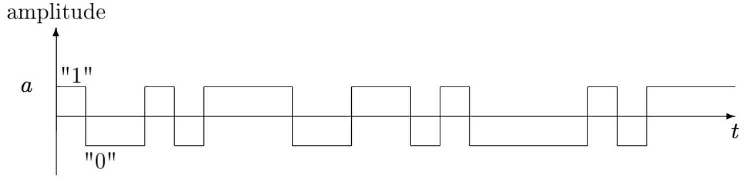
## Principes

Pour identifier à partir des données, il est nécessaire que l'information soit contenue dans les données !
$\longrightarrow$ excitation dans tout le spectre de fréquence susceptible de contenir les constantes de temps du système

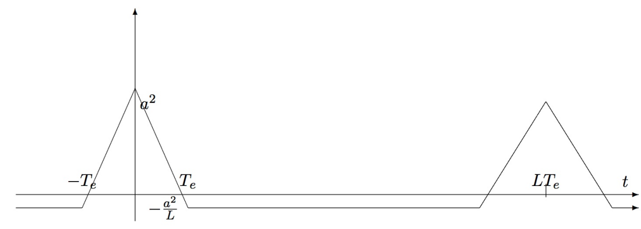
- $sin(\omega t)$ : balayage en fréquence, parfait d'un point de vue spectre, mais pas toujours accepté par tous les systèmes
- $\delta(t)$ : parfait d'un point de vue théorique, difficile voire impossible à réaliser en pratique
- $b(t)$ : bruit blanc idéal d'un point de vue spectral mais réalisation pratique ?
- SBPA : signal binaire pseudo aléatoire : réalisation pratique d'un signal aléatoire (en anglais : Pseudo Random Binary Sequence, PRBS).
- parfois impossible d'exciter un système (systèmes environnementaux, machine en production par ex.) : utiulisation des commandes naturelles du système comme signal d'entrée.

# Choix du signal d'excitation

## SBPA

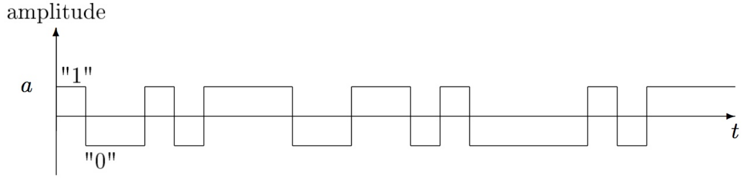



- Séquence de $N$ bits de **longueur** : $L = 2^N - 1$
- $E[s(t)] = \frac{a}{L}$
- **Fonction d'autocorrélation**

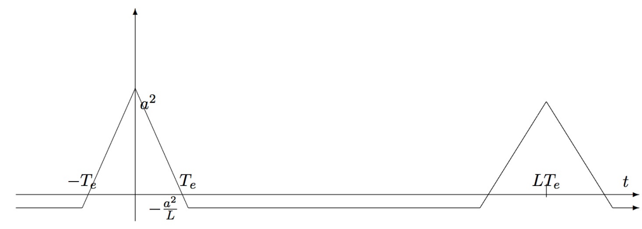


# Choix du signal d'excitation

## SBPA

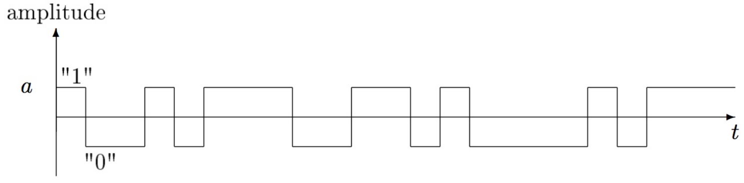



- Séquence de $N$ bits de **longueur** : $L = 2^N - 1$
- $E[s(t)] = \frac{a}{L}$
- **Fonction d'autocorrélation**



# Choix du signal d'excitation

## SBPA

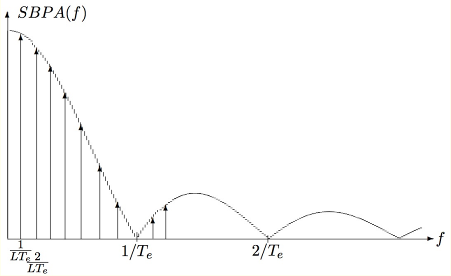



- Séquence de $N$ bits de **longueur** : $L = 2^N - 1$
- $E[s(t)] = \frac{a}{L}$
- **Fonction d'autocorrélation**

# Choix du signal d'excitation

## SBPA

- **Spectre de la SBPA** (TF de la fonction d'autocorrélation)




- **Choix des paramètres d'une SBPA**

Plus la séquence est longue ($L$) :

◇ plus il y a d'informations

◇ plus le transitoire est négligeable

◇ plus les organes "souffrent"

◇ plus les dérives s'accentuent

Deuxième compromis, plus $a$ est petit :

◇ plus le signal est noyé dans le bruit

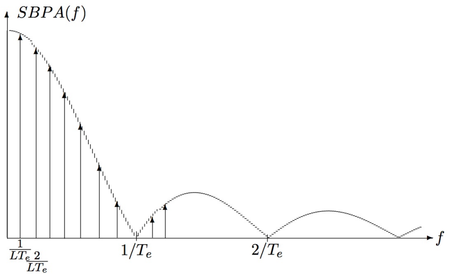
◇ moins les organes "souffrent"

◇ moins les dérives s'accentuent

# Choix du signal d'excitation

## SBPA

- **Spectre de la SBPA** (TF de la fonction d'autocorrélation)




- **Choix des paramètres d'une SBPA**

Plus la séquence est longue ($L$) :

- ◇ plus il y a d'informations
- ◇ plus le transitoire est négligeable
- ◇ plus les organes "souffrent"
- ◇ plus les dérives s'accentuent

Deuxième compromis, plus $a$ est petit :

- ◇ plus le signal est noyé dans le bruit
- ◇ moins les organes "souffrent"
- ◇ moins les dérives s'accentuent

# Choix du signal d'excitation

## SBPA

SBPA de longueur $L = 2^N - 1$ envoyée à la fréquence $1/T_e$ :

- plateau de longueur maximale : $NT_e$
- plus petit plateau de longueur : $T_e$

# Choix du signal d'excitation

## SBPA

SBPA de longueur $L = 2^N - 1$ envoyée à la fréquence $1/T_e$ :

- plateau de longueur maximale : $NT_e$
- plus petit plateau de longueur : $T_e$

Donc, **le choix de $N$ (et donc $L$) et de $T_e$** est un compromis entre :

- bonne identification du gain statique
- bonne excitation sur la bande de fréquence du système

# Choix du signal d'excitation

## SBPA

SBPA de longueur $L = 2^N - 1$ envoyée à la fréquence $1/T_e$ :

- plateau de longueur maximale : $NT_e$
- plus petit plateau de longueur : $T_e$

Donc, **le choix de $N$ (et donc $L$) et de $T_e$** est un compromis entre :

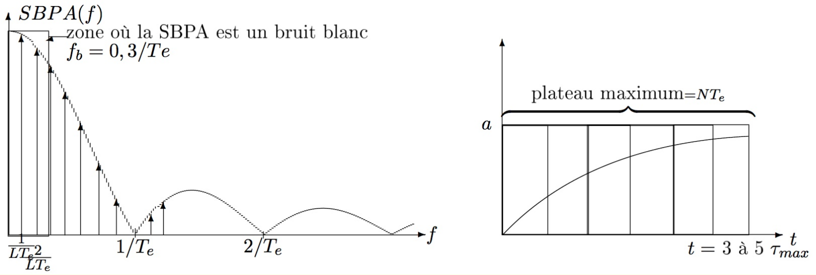
- bonne identification du gain statique
- bonne excitation sur la bande de fréquence du système

**Solution**

- pour gain statique : $NT_e = \mathbf{3\ à\ 5}\ T_{MAX}$ ($T_{MAX}$ = plus grande constante de temps du système)
- pour excitation : $0.3\frac{1}{T_e} = \frac{1}{2\pi T_{min}}$ ($T_{min}$ = plus petite constante de temps du système)

# Choix du signal d'excitation

## SBPA




Remarque : Si $T_{MAX} \gg T_{min}$, $N$ devient vite très grand donc $LT_e$ devient prohibitif !
$\longrightarrow$ Identification de systèmes à constantes de temps éloignées complexe.

# Plan

# Structure du modèle

## Modèles statiques et dynamiques

### Modèles statiques

- *dérivées* et *valeurs moyennes* non nulles
- composantes cycliques (périodiques)
- modèle du type
  - $y(t) = \varphi^T\theta + v(t)$
    où le vecteur de régression $\varphi(t)$ est déterministe (parfaitement connu), ne dépend pas des valeurs passées de la sortie
  - *exemple* : $y(t) = \varphi^T(t)\theta + v(t)$
    avec $\varphi^T(t) = [1 \quad t \quad t^2]$

# Avant de commencer : les opérateurs

## Modèles à temps continu – Opérateur dérivée

- Signal à temps continu : $y(t)$
- Transformée de Fourier : $Y(\omega) = \int_{-\infty}^{\infty} y(t)e^{-i\omega t}dt$
- Transformée de Laplace : $Y(s) = \int_{-\infty}^{\infty} y(t)e^{-st}dt$
- Système linéaire :

$$\begin{aligned} y(t) &= g * u(t) \\ Y(\omega) &= G(\omega)U(\omega) \\ Y(s) &= G(s)U(s) \end{aligned}$$

**Opérateur dérivée :** $pu(t) = \dot{u}(t)$ fonctionne comme la variable de Laplace $s$, mais dans le domaine temporel.

**Exemple :** Équivalence des écritures suivantes

$$y(t) = 0.5\dot{u}(t) + u(t) \tag{1}$$

$$y(t) = (0.5p + 1)u(t) \tag{2}$$

$$Y(s) = (0.5s + 1)U(s) \tag{3}$$

# Avant de commencer : les opérateurs

## Modèles à temps discret – Opérateur retard

- Signal à temps discret : $y(kh)$
- Transformée de Fourier : $Y^{(h)}(\omega) = h\sum_{k=-\infty}^{\infty} y(kh)e^{-i\omega kh}$
- Transformée en $z$ : $Y(z) = \sum_{k=-\infty}^{\infty} y(kh)z^{-k}$
- Système linéaire :

$$
\begin{aligned}
y(kh) &= g * u(kh) \\
Y^{(h)}(\omega) &= G_d(e^{i\omega h})U^{(h)}(\omega) \\
Y(z) &= G_d(z)U(z)
\end{aligned}
$$

**Opérateur retard :** $qu(kh) = u(kh+h)$ fonctionne comme la variable $z$, mais dans le domaine temporel.
**Exemple :** Équivalence des écritures suivantes

$$y(kh) = 0.5u(kh) + u(kh-h) \quad (4)$$

$$y(kh) = (0.5 + q^{-1})u(kh) \quad (5)$$

$$Y(z) = (0.5 + z^{-1})U(z) \quad (6)$$

# Structure du modèle

## Modèles statiques et dynamiques

### Modèles dynamiques – Représentation polynomiale

- Forme générale pour un modèle à temps discret

$$y(t_k) = G(q^{-1})u(t_k) + H(q^{-1})e(t_k)$$

ou

$$y(t_k) = \varphi^T(t_k)\theta + v(t_k)$$

  où le vecteur de régression $\varphi(t_k)$ dépend des valeurs passées de $y(t_k), u(t_k)$
- Structures classiques :
  - ◇ modèle ARX
  - ◇ modèle OE
  - ◇ modèle ARMAX
  - ◇ modèle BJ
- Ces modèles linéaires ne sont valables qu'autour d'un point de fonctionnement, avec une zone de variation assez étroite.

# Structure du modèle

## Modèle

### En pratique

- Les données expérimentales contiennent à la fois des composantes statiques et dynamiques

$$y(t_k) = \underbrace{y_s(t_k)}_{\text{composante statique}} + \underbrace{y_d(t_k)}_{\text{composante dynamique}} \tag{7}$$

- Approche habituelle
  - ◇ estimer la composante statique
  - ◇ enlever la composante statique avant d'estimer le modèle dynamique
    $\longrightarrow$ éliminer la composante moyenne des signaux d'E/S par exemple

# Structure du modèle

## Modèles paramétriques linéaires à temps discret

- *Fonction (opérateur) de transfert*

$$\underbrace{y(t_k) + a_1 y(t_{k-1}) + \ldots + a_{n_a} y(t - n_a)}_{A(q^{-1})y(t_k)} = \underbrace{b_1 u(t_{k-1}) + \ldots + b_{n_b} u(t - n_b)}_{B(q^{-1})u(t_k)}$$

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

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

$$q^{-1} u(t_k) = u(t_{k-1}) \quad \text{opérateur retard}$$

$\boxed{G(z) = \frac{Y(z)}{U(z)} = \frac{B(z)}{A(z)}}$, $z$ variable de la transformée en $Z$

- *Représentation d'état*

$$\begin{cases} x(t_{k+1}) = Ax(t_k) + Bu(t_k) \\ y(t_k) = Cx(t_k) + Du(t_k) \end{cases} \qquad \boxed{G(z) = C(zI - A)^{-1}B + D}$$

# Structure du modèle

## Modèles paramétriques linéaires à temps discret

- *Fonction (opérateur) de transfert*

$$\underbrace{y(t_k) + a_1 y(t_{k-1}) + \ldots + a_{n_a} y(t - n_a)}_{A(q^{-1})y(t_k)} = \underbrace{b_1 u(t_{k-1}) + \ldots + b_{n_b} u(t - n_b)}_{B(q^{-1})u(t_k)}$$

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

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

$$q^{-1} u(t_k) = u(t_{k-1}) \quad \text{opérateur retard}$$

$\boxed{G(z) = \frac{Y(z)}{U(z)} = \frac{B(z)}{A(z)}}$, $z$ variable de la transformée en $Z$

- *Représentation d'état*

$$\begin{cases} x(t_{k+1}) = Ax(t_k) + Bu(t_k) \\ y(t_k) = Cx(t_k) + Du(t_k) \end{cases} \qquad \boxed{G(z) = C(zI - A)^{-1}B + D}$$

# Structure du modèle

## Modèles paramétriques linéaires à temps continu

- *Fonction (opérateur) de transfert*

$$\underbrace{\frac{d^n y(t)}{dt^n} + a_1 \frac{d^{n-1} y(t)}{dt^{n-1}} + \ldots + a_n y(t)}_{A(p)y(t)} = \underbrace{b_0 \frac{d^m u(t)}{dt^m} + \ldots + b_m u(t)}_{B(p)u(t)}$$

$$A(p) = p^n + a_1 p^{n-1} + \ldots + a_n$$

$$B(p) = b_0 p^m + b_1 p^{m-1} + \ldots + b_m$$

$$p = \frac{d}{dt} \quad \text{opérateur différentiel}$$

$\boxed{G(s) = \frac{Y(s)}{U(s)} = \frac{B(s)}{A(s)}}$, $s$ variable de Laplace

- *Représentation d'état*

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \qquad G(s) = C(sI - A)^{-1}B + D$$

# Structure du modèle

## Modèles paramétriques linéaires à temps continu

- *Fonction (opérateur) de transfert*

$$\underbrace{\frac{d^n y(t)}{dt^n} + a_1 \frac{d^{n-1} y(t)}{dt^{n-1}} + \ldots + a_n y(t)}_{A(p)y(t)} = \underbrace{b_0 \frac{d^m u(t)}{dt^m} + \ldots + b_m u(t)}_{B(p)u(t)}$$

$$A(p) = p^n + a_1 p^{n-1} + \ldots + a_n$$

$$B(p) = b_0 p^m + b_1 p^{m-1} + \ldots + b_m$$

$$p = \frac{d}{dt} \quad \text{opérateur différentiel}$$

$\boxed{G(s) = \frac{Y(s)}{U(s)} = \frac{B(s)}{A(s)}}$, $s$ variable de Laplace

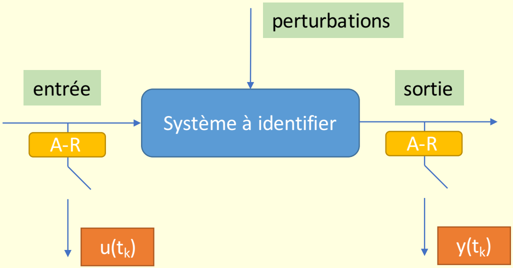
- *Représentation d'état*

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \qquad \boxed{G(s) = C(sI - A)^{-1}B + D}$$

# Structure du modèle

## Choix

**Objectif :** déterminer un modèle paramétrique à temps discret d'un système à partir de la mesure des signaux d'E/S



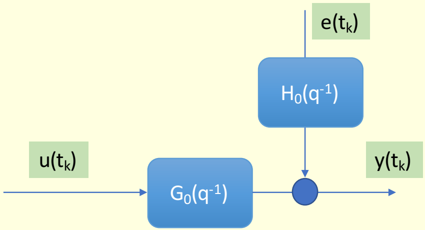

Une fois les signaux acquis, plusieurs choix doivent encore être effectués :

- quel modèle pour représenter l'effet des perturbations (bruits) ?
- quelle structure de modèles pour le système ?
- quelle méthode d'estimation des paramètres du modèle choisi ?

# Structure du modèle

## Perturbations

- bruit sur la sortie, entrée non mesurée, bruit de capteur,...
- souvent représentées comme des processus stochastiques stationnaires prenant la forme d'un terme additif sur la sortie



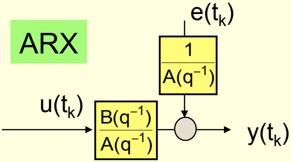

$$v(t_k) = H_0(q^{-1})e(t_k)$$
$$y(t_k) = G_0(q^{-1})u(t_k) + H_0(q^{-1})e(t_k)$$

- Hypothèse de travail : $e(t_k)$ est un bruit blanc gaussien à temps discret
- Densité de probabilité définie par ses 2 premiers moments (moyenne et variance)
- $e(t_k) \in \mathcal{N}(0, \lambda^2)$

# Structure du modèle

## Modèles dynamiques – Représentation polynomiale

Modèles dépendant de l'hypothèse sur les perturbations

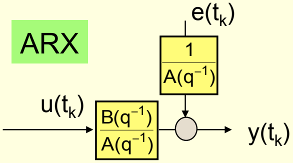



$$A(q^{-1})y(t_k) = B(q^{-1})u(t_k) + e(t_k)$$

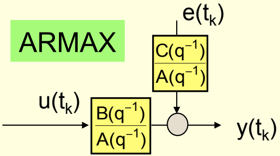
# Structure du modèle

## Modèles dynamiques – Représentation polynomiale

Modèles dépendant de l'hypothèse sur les perturbations




$$A(q^{-1})y(t_k) = B(q^{-1})u(t_k) + e(t_k)$$



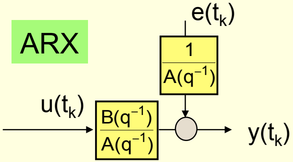

$$A(q^{-1})y(t_k) = B(q^{-1})u(t_k) + C(q^{-1})e(t_k)$$

# Structure du modèle

## Modèles dynamiques – Représentation polynomiale

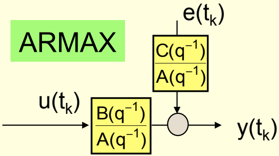
### Modèles dépendant de l'hypothèse sur les perturbations

**ARX**



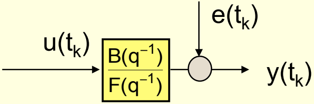

$$A(q^{-1})y(t_k) = B(q^{-1})u(t_k) + e(t_k)$$

**ARMAX**




$$A(q^{-1})y(t_k) = B(q^{-1})u(t_k) + C(q^{-1})e(t_k)$$

**OE**



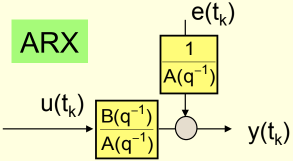

$$y(t_k) = \frac{B(q^{-1})}{F(q^{-1})}u(t_k) + e(t_k)$$

# Structure du modèle

## Modèles dynamiques – Représentation polynomiale

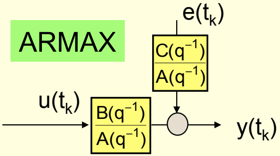
### Modèles dépendant de l'hypothèse sur les perturbations

**ARX**



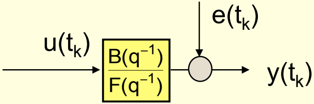

$$A(q^{-1})y(t_k) = B(q^{-1})u(t_k) + e(t_k)$$

**ARMAX**



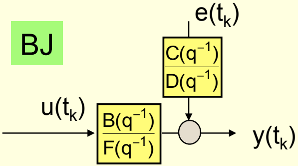

$$A(q^{-1})y(t_k) = B(q^{-1})u(t_k) + C(q^{-1})e(t_k)$$

**OE**




$$y(t_k) = \frac{B(q^{-1})}{F(q^{-1})}u(t_k) + e(t_k)$$

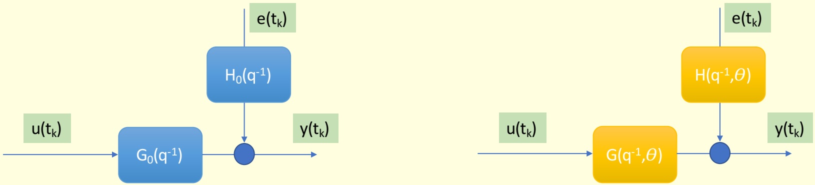
**BJ**




$$y(t_k) = \frac{B(q^{-1})}{F(q^{-1})}u(t_k) + \frac{C(q^{-1})}{D(q^{-1})}e(t_k)$$

# Structure du modèle

## Paramétrisation




Système vrai $\mathcal{S}$

$$y(t_k) = G_0(q^{-1})u(t_k) + H_0(q^{-1})e(t_k)$$

Modèle estimé $\mathcal{M}$

$$y(t_k) = G(q^{-1},\theta)u(t_k) + H_0(q^{-1},\theta)e(t_k)$$

- Le vecteur des paramètres $\theta$ rassemble les coeff de $G(q^{-1},\theta)$ et $H(q^{-1},\theta)$ **L'estimée de $\theta$ sera notée $\hat{\theta}$** et dépendra de l'estimateur utilisé (MC,. . . ).
- But de l'estimation : fournir le vecteur de paramètres le plus approprié en fonction du but recherché
- Ce cours : modèle de type fonction de transfert

# Structure du modèle

## Paramétrisation

### Paramétrisation des fonctions de transfert

- Paramétrisation des fonctions de transfert $G(q^{-1},\theta)$ et $H(q^{-1},\theta)$ (*paramétrisation utilisée au sein de la boîte à outils SID de Matlab*)

$$G(q^{-1},\theta) = q^{-n_k}\frac{B(q^{-1})}{F(q^{-1})} = q^{-n_k}\frac{b_0 + b_1(q^{-1}) + \cdots + b_{n_{b-1}}q^{-n_b+1}}{1 + f_1(q^{-1}) + \cdots + f_{n_f}q^{-n_f}}$$

$$H(q^{-1},\theta) = \frac{C(q^{-1})}{D(q^{-1})} = q^{-n_k}\frac{1 + c_1(q^{-1}) + \cdots + c_{n_c}q^{-n_c}}{1 + d_1(q^{-1}) + \cdots + d_{n_d}q^{-n_f}}$$

- Le vecteur des paramètres $\theta$ rassemble les coefficients $b_i, f_i, c_i, d_i$
- $n_k$ représente le retard (nombre d'échantillons de retard) entre la sortie et l'entrée
- $n_f, n_b, n_c$ et $n_d$ correspondent au nombre de paramètres à estimer des polynômes

# Structure du modèle

## Paramétrisation

### Paramétrisation des fonctions de transfert

- Paramétrisation des fonctions de transfert $G(q^{-1}, \theta)$ et $H(q^{-1}, \theta)$ (*paramétrisation utilisée au sein de la boîte à outils SID de Matlab*)

$$G(q^{-1}, \theta) = q^{-n_k} \frac{B(q^{-1})}{F(q^{-1})} = q^{-n_k} \frac{b_0 + b_1(q^{-1}) + \cdots + b_{n_{b-1}} q^{-n_b+1}}{1 + f_1(q^{-1}) + \cdots + f_{n_f} q^{-n_f}}$$

$$H(q^{-1}, \theta) = \frac{C(q^{-1})}{D(q^{-1})} = q^{-n_k} \frac{1 + c_1(q^{-1}) + \cdots + c_{n_c} q^{-n_c}}{1 + d_1(q^{-1}) + \cdots + d_{n_d} q^{-n_f}}$$

- Le vecteur des paramètres $\theta$ rassemble les coefficients $b_i, f_i, c_i, d_i$
- $n_k$ représente le retard (nombre d'échantillons de retard) entre la sortie et l'entrée
- $n_f, n_b, n_c$ et $n_d$ correspondent au nombre de paramètres à estimer des polynômes

# Structure du modèle

## Paramétrisation

### Paramétrisation des fonctions de transfert

- Paramétrisation des fonctions de transfert $G(q^{-1}, \theta)$ et $H(q^{-1}, \theta)$ (*paramétrisation utilisée au sein de la boîte à outils SID de Matlab*)

$$G(q^{-1}, \theta) = q^{-n_k} \frac{B(q^{-1})}{F(q^{-1})} = q^{-n_k} \frac{b_0 + b_1(q^{-1}) + \cdots + b_{n_{b-1}} q^{-n_b+1}}{1 + f_1(q^{-1}) + \cdots + f_{n_f} q^{-n_f}}$$

$$H(q^{-1}, \theta) = \frac{C(q^{-1})}{D(q^{-1})} = q^{-n_k} \frac{1 + c_1(q^{-1}) + \cdots + c_{n_c} q^{-n_c}}{1 + d_1(q^{-1}) + \cdots + d_{n_d} q^{-n_f}}$$

- Le vecteur des paramètres $\theta$ rassemble les coefficients $b_i, f_i, c_i, d_i$
- $n_k$ représente le retard (nombre d'échantillons de retard) entre la sortie et l'entrée
- $n_f, n_b, n_c$ et $n_d$ correspondent au nombre de paramètres à estimer des polynômes

# Structure du modèle

## Paramétrisation

### Paramétrisation des fonctions de transfert

- Paramétrisation des fonctions de transfert $G(q^{-1},\theta)$ et $H(q^{-1},\theta)$ (*paramétrisation utilisée au sein de la boîte à outils SID de Matlab*)

$$G(q^{-1},\theta) = q^{-n_k}\frac{B(q^{-1})}{F(q^{-1})} = q^{-n_k}\frac{b_0 + b_1(q^{-1}) + \cdots + b_{n_{b-1}}q^{-n_b+1}}{1 + f_1(q^{-1}) + \cdots + f_{n_f}q^{-n_f}}$$

$$H(q^{-1},\theta) = \frac{C(q^{-1})}{D(q^{-1})} = q^{-n_k}\frac{1 + c_1(q^{-1}) + \cdots + c_{n_c}q^{-n_c}}{1 + d_1(q^{-1}) + \cdots + d_{n_d}q^{-n_f}}$$

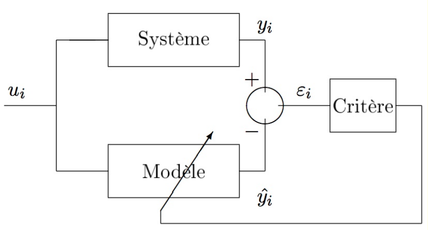
- Le vecteur des paramètres $\theta$ rassemble les coefficients $b_i, f_i, c_i, d_i$
- $n_k$ représente le retard (nombre d'échantillons de retard) entre la sortie et l'entrée
- $n_f, n_b, n_c$ et $n_d$ correspondent au nombre de paramètres à estimer des polynômes

# Plan

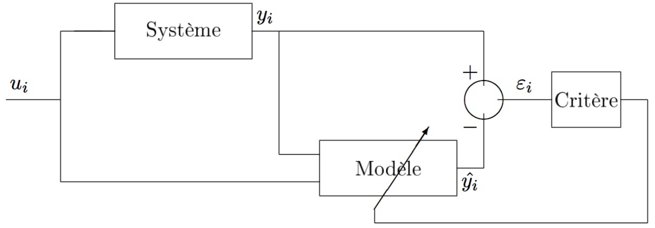
# Méthodes d'estimation

## Deux grandes méthodes

- **Minimisation de l'erreur de sortie**

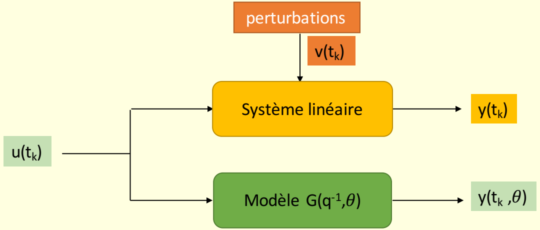



- **Minimisation de l'erreur de prédiction**



# Méthodes d'estimation

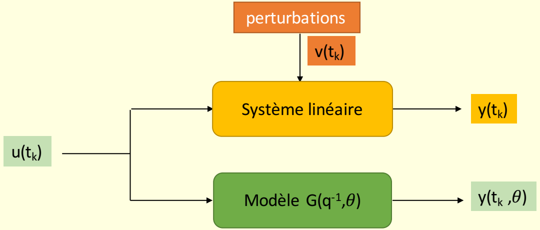
## Erreur de prédiction




- Pour un vecteur de paramètres donné $\theta$, le modèle permet de calculer la prédiction de la sortie notée $\hat{y}(t_k, \theta)$.
- Il faut alors déterminer $\theta$ tel que $\hat{y}(t_k, \theta)$ soit "la plus proche possible" de $y(t_k)$

# Méthodes d'estimation

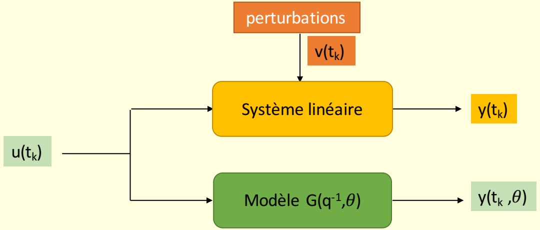
## Erreur de prédiction




- Pour un vecteur de paramètres donné $\theta$, le modèle permet de calculer la prédiction de la sortie notée $\hat{y}(t_k, \theta)$.
- Il faut alors déterminer $\theta$ tel que $\hat{y}(t_k, \theta)$ soit "la plus proche possible" de $y(t_k)$

# Méthodes d'estimation

## Erreur de prédiction

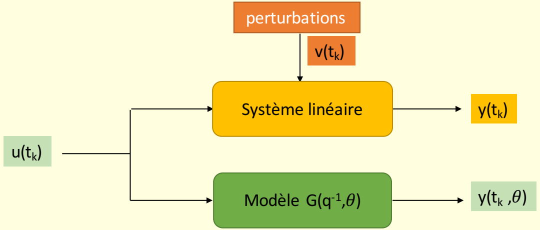



Pour résoudre ce problème d'estimation, il faut remarquer :

1. la prédiction de la sortie $\hat{y}(t_k, \theta)$ dépend de la structure de modèle choisie et donc du modèle de bruit
2. le concept "doit être la plus proche possible" doit être formulé mathématiquement

# Méthodes d'estimation

## Erreur de prédiction



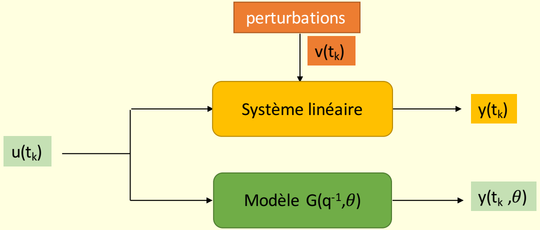

Pour résoudre ce problème d'estimation, il faut remarquer :

1. la prédiction de la sortie $\hat{y}(t_k, \theta)$ dépend de la structure de modèle choisie et donc du modéle de bruit

2. le concept "doit être la plus proche possible" doit être formulé mathématiquement

# Méthodes d'estimation

## Erreur de prédiction




Pour résoudre ce problème d'estimation, il faut remarquer :

1. la prédiction de la sortie $\hat{y}(t_k, \theta)$ dépend de la structure de modèle choisie et donc du modèle de bruit
2. le concept "doit être la plus proche possible" doit être formulé mathématiquement

# Méthodes d'estimation

## Erreur de prédiction – Cas général

Soit un ensemble $Z^N$ de données d'E/S

$$Z^N = \{u(t_1), y(t_1), u(t_2), y(t_2), \cdots, u(t_N), y(t_N)\}$$

1. Calculer la prédiction de la sortie à partir des données jusqu'à l'instant $t_k$
   $\hat{y}(t_k, \theta)$
2. Former l'erreur de prédiction $\varepsilon(t_k) = y(t_k) - \hat{y}(t_k, \theta)$
3. Construire la fonction de coût (ou critère)
   $$V(\theta, Z^N) = \frac{1}{N}\sum_{k=1}^{N}\varepsilon^2(t_k) = \frac{1}{N}\sum_{k=1}^{N}\left(y(t_k) - \hat{y}(t_k, \theta)\right)^2$$
4. Les paramètres optimaux sont ceux qui minimisent la fonction de coût
   $\hat{\theta} = \arg\min_\theta V(\theta, Z^N)$

# Méthodes d'estimation

## Erreur de prédiction – Cas général

Soit un ensemble $Z^N$ de données d'E/S

$$Z^N = \{u(t_1), y(t_1), u(t_2), y(t_2), \cdots, u(t_N), y(t_N)\}$$

1. Calculer la prédiction de la sortie à partir des données jusqu'à l'instant $t_k$
   $\hat{y}(t_k, \theta)$
2. Former l'erreur de prédiction $\varepsilon(t_k) = y(t_k) - \hat{y}(t_k, \theta)$
3. Construire la fonction de coût (ou critère)
   $$V(\theta, Z^N) = \frac{1}{N}\sum_{k=1}^{N} \varepsilon^2(t_k) = \frac{1}{N}\sum_{k=1}^{N} \left(y(t_k) - \hat{y}(t_k, \theta)\right)^2$$
4. Les paramètres optimaux sont ceux qui minimisent la fonction de coût
   $\hat{\theta} = \arg\min_\theta V(\theta, Z^N)$

# Méthodes d'estimation

## Erreur de prédiction – Cas général

Soit un ensemble $Z^N$ de données d'E/S

$$Z^N = \{u(t_1), y(t_1), u(t_2), y(t_2), \cdots, u(t_N), y(t_N)\}$$

1. Calculer la prédiction de la sortie à partir des données jusqu'à l'instant $t_k$
   $\hat{y}(t_k, \theta)$
2. Former l'erreur de prédiction $\varepsilon(t_k) = y(t_k) - \hat{y}(t_k, \theta)$
3. Construire la fonction de coût (ou critère)

$$V(\theta, Z^N) = \frac{1}{N}\sum_{k=1}^{N} \varepsilon^2(t_k) = \frac{1}{N}\sum_{k=1}^{N} \big(y(t_k) - \hat{y}(t_k, \theta)\big)^2$$

4. Les paramètres optimaux sont ceux qui minimisent la fonction de coût
   $\hat{\theta} = \arg\min_\theta V(\theta, Z^N)$

# Méthodes d'estimation

## Erreur de prédiction – Cas général

Soit un ensemble $Z^N$ de données d'E/S

$$Z^N = \{u(t_1), y(t_1), u(t_2), y(t_2), \cdots, u(t_N), y(t_N)\}$$

1. Calculer la prédiction de la sortie à partir des données jusqu'à l'instant $t_k$
   $\hat{y}(t_k, \theta)$
2. Former l'erreur de prédiction $\varepsilon(t_k) = y(t_k) - \hat{y}(t_k, \theta)$
3. Construire la fonction de coût (ou critère)
   $$V(\theta, Z^N) = \frac{1}{N}\sum_{k=1}^{N} \varepsilon^2(t_k) = \frac{1}{N}\sum_{k=1}^{N} \big(y(t_k) - \hat{y}(t_k, \theta)\big)^2$$
4. Les paramètres optimaux sont ceux qui minimisent la fonction de coût
   $\hat{\theta} = \arg\min_\theta V(\theta, Z^N)$

# Méthodes d'estimation

## Erreur de prédiction – Cas général

- Soit le modèle général :

$$y(t_k) = G(q^{-1},\theta)u(t_k) + H(q^{-1},\theta)e(t_k)$$

- Multiplions par $H^{-1}(q^{-1},\theta)$ (pour "blanchir" le terme de bruit)

$$H^{-1}(q^{-1},\theta)y(t_k) = H^{-1}(q^{-1},\theta)G(q^{-1},\theta)u(t_k) + e(t_k)$$
$$y(t_k) - y(t_k) + H^{-1}(q^{-1},\theta)y(t_k) = H^{-1}(q^{-1},\theta)G(q^{-1},\theta)u(t_k) + e(t_k)$$
$$y(t_k) = \left(1 - H^{-1}(q^{-1},\theta)\right)y(t_k) + H^{-1}(q^{-1},\theta)G(q^{-1},\theta)u(t_k) + e(t_k)$$

- Puisque $e(t_k)$ est un bruit blanc, on en déduit la meilleure prédiction (cas général) :

$$\hat{y}(t_k,\theta) = \left(1 - H^{-1}(q^{-1},\theta)\right)y(t_k) + H^{-1}(q^{-1},\theta)G(q^{-1},\theta)u(t_k)$$

- Suivant les modèles de bruit et du système cette équation prend plusieurs formes différentes.
  Cas le plus simple (linéaire en les paramètres) : **modèle ARX**.

# Méthodes d'estimation

## Erreur de prédiction – Cas général

- Soit le modèle général :

$$y(t_k) = G(q^{-1},\theta)u(t_k) + H(q^{-1},\theta)e(t_k)$$

- Multiplions par $H^{-1}(q^{-1},\theta)$ (pour "blanchir" le terme de bruit)

$$H^{-1}(q^{-1},\theta)y(t_k) = H^{-1}(q^{-1},\theta)G(q^{-1},\theta)u(t_k) + e(t_k)$$
$$y(t_k) - y(t_k) + H^{-1}(q^{-1},\theta)y(t_k) = H^{-1}(q^{-1},\theta)G(q^{-1},\theta)u(t_k) + e(t_k)$$
$$y(t_k) = \left(1 - H^{-1}(q^{-1},\theta)\right)y(t_k) + H^{-1}(q^{-1},\theta)G(q^{-1},\theta)u(t_k) + e(t_k)$$

- Puisque $e(t_k)$ est un bruit blanc, on en déduit la meilleure prédiction (cas général) :

$$\hat{y}(t_k,\theta) = \left(1 - H^{-1}(q^{-1},\theta)\right)y(t_k) + H^{-1}(q^{-1},\theta)G(q^{-1},\theta)u(t_k)$$

- Suivant les modèles de bruit et du système cette équation prend plusieurs formes différentes.
Cas le plus simple (linéaire en les paramètres) : **modèle ARX**.

# Méthodes d'estimation

## Erreur de prédiction – Cas général

- Soit le modèle général :

$$y(t_k) = G(q^{-1}, \theta)u(t_k) + H(q^{-1}, \theta)e(t_k)$$

- Multiplions par $H^{-1}(q^{-1}, \theta)$ (pour "blanchir" le terme de bruit)

$$H^{-1}(q^{-1}, \theta)y(t_k) = H^{-1}(q^{-1}, \theta)G(q^{-1}, \theta)u(t_k) + e(t_k)$$
$$y(t_k) - y(t_k) + H^{-1}(q^{-1}, \theta)y(t_k) = H^{-1}(q^{-1}, \theta)G(q^{-1}, \theta)u(t_k) + e(t_k)$$
$$y(t_k) = \left(1 - H^{-1}(q^{-1}, \theta)\right)y(t_k) + H^{-1}(q^{-1}, \theta)G(q^{-1}, \theta)u(t_k) + e(t_k)$$

- Puisque $e(t_k)$ est un bruit blanc, on en déduit la meilleure prédiction (cas général) :

$$\hat{y}(t_k, \theta) = \left(1 - H^{-1}(q^{-1}, \theta)\right)y(t_k) + H^{-1}(q^{-1}, \theta)G(q^{-1}, \theta)u(t_k)$$

- Suivant les modèles de bruit et du système cette équation prend plusieurs formes différentes.
  Cas le plus simple (linéaire en les paramètres) : **modèle ARX**.

# Méthodes d'estimation

## Erreur de prédiction – Cas général

- Soit le modèle général :

$$y(t_k) = G(q^{-1}, \theta)u(t_k) + H(q^{-1}, \theta)e(t_k)$$

- Multiplions par $H^{-1}(q^{-1}, \theta)$ (pour "blanchir" le terme de bruit)

$$H^{-1}(q^{-1}, \theta)y(t_k) = H^{-1}(q^{-1}, \theta)G(q^{-1}, \theta)u(t_k) + e(t_k)$$
$$y(t_k) - y(t_k) + H^{-1}(q^{-1}, \theta)y(t_k) = H^{-1}(q^{-1}, \theta)G(q^{-1}, \theta)u(t_k) + e(t_k)$$
$$y(t_k) = \left(1 - H^{-1}(q^{-1}, \theta)\right)y(t_k) + H^{-1}(q^{-1}, \theta)G(q^{-1}, \theta)u(t_k) + e(t_k)$$

- Puisque $e(t_k)$ est un bruit blanc, on en déduit la meilleure prédiction (cas général) :

$$\hat{y}(t_k, \theta) = \left(1 - H^{-1}(q^{-1}, \theta)\right)y(t_k) + H^{-1}(q^{-1}, \theta)G(q^{-1}, \theta)u(t_k)$$

- Suivant les modèles de bruit et du système cette équation prend plusieurs formes différentes.
  Cas le plus simple (linéaire en les paramètres) : **modèle ARX**.

# Méthodes d'estimation

## Erreur de prédiction – Cas général

- Soit le modèle général :

$$y(t_k) = G(q^{-1}, \theta)u(t_k) + H(q^{-1}, \theta)e(t_k)$$

- Multiplions par $H^{-1}(q^{-1}, \theta)$ (pour "blanchir" le terme de bruit)

$$H^{-1}(q^{-1}, \theta)y(t_k) = H^{-1}(q^{-1}, \theta)G(q^{-1}, \theta)u(t_k) + e(t_k)$$
$$y(t_k) - y(t_k) + H^{-1}(q^{-1}, \theta)y(t_k) = H^{-1}(q^{-1}, \theta)G(q^{-1}, \theta)u(t_k) + e(t_k)$$
$$y(t_k) = \left(1 - H^{-1}(q^{-1}, \theta)\right)y(t_k) + H^{-1}(q^{-1}, \theta)G(q^{-1}, \theta)u(t_k) + e(t_k)$$

- Puisque $e(t_k)$ est un bruit blanc, on en déduit la meilleure prédiction (cas général) :

$$\boxed{\hat{y}(t_k, \theta) = \left(1 - H^{-1}(q^{-1}, \theta)\right)y(t_k) + H^{-1}(q^{-1}, \theta)G(q^{-1}, \theta)u(t_k)}$$

- Suivant les modèles de bruit et du système cette équation prend plusieurs formes différentes.
  Cas le plus simple (linéaire en les paramètres) : **modèle ARX**.

# Méthodes d'estimation

## Erreur de prédiction – Cas d'un modèle ARX

- Soit le modèle ARX

$$y(t_k) = q^{n_k}\frac{B(q^{-1},\theta)}{A(q^{-1},\theta)}u(t_k) + \frac{1}{A(q^{-1},\theta)}e(t_k) \tag{8}$$

- **Prédiction de la sortie**

$\hat{y}(t_k,\theta) = \big(1 - H^{-1}(q^{-1},\theta)\big)y(t_k) + H^{-1}(q^{-1},\theta)G(q^{-1},\theta)u(t_k)$

$\hat{y}(t_k,\theta) = \big(1 - A(q^{-1},\theta)\big)y(t_k) + q^{-n_k}B(q^{-1},\theta)u(t_k)$

$\hat{y}(t_k,\theta) = -a_1 y(t_{k-1}) - a_{n_a}y(t_{k-n_a}) + b_0 u(t_{k-n_k}) + \cdots + b_{n_{b-1}}u(t_{k-n_k-n_b+1})$

$\hat{y}(t_k,\theta) = \varphi^T(t_k)\theta$

$$\varphi^T(t_k) = \big[-y(t_{k-1}) \ \cdots - y(t_{k-n_a}) \ u(t_{k-n_k}) \ \cdots u(t_{k-n_k-n_b+1})\big]$$

$$\theta = \big[a_1 \ \cdots \ a_{n_a} \ b_0 \ \cdots \ b_{n_b-1}\big]$$

- Prédiction de la sortie = fonction linéaire en les paramètres, le problème est donc celui d'une régression linéaire et les paramètres peuvent être cherchés par des procédures de moindres carrés.

# Méthodes d'estimation

## Erreur de prédiction – Cas d'un modèle ARX

- Soit le modèle ARX

$$y(t_k) = q^{n_k} \frac{B(q^{-1},\theta)}{A(q^{-1},\theta)} u(t_k) + \frac{1}{A(q^{-1},\theta)} e(t_k) \qquad (8)$$

- **Prédiction de la sortie**

$$\hat{y}(t_k,\theta) = \left(1 - H^{-1}(q^{-1},\theta)\right) y(t_k) + H^{-1}(q^{-1},\theta) G(q^{-1},\theta) u(t_k)$$

$$\hat{y}(t_k,\theta) = \left(1 - A(q^{-1},\theta)\right) y(t_k) + q^{-n_k} B(q^{-1},\theta) u(t_k)$$

$$\hat{y}(t_k,\theta) = -a_1 y(t_{k-1}) - a_{n_a} y(t_{k-n_a}) + b_0 u(t_{k-n_k}) + \cdots + b_{n_{b-1}} u(t_{k-n_k-n_b+1})$$

$$\hat{y}(t_k,\theta) = \varphi^T(t_k)\theta$$

$$\varphi^T(t_k) = \left[ -y(t_{k-1}) \ \cdots - y(t_{k-n_a}) \ u(t_{k-n_k}) \ \cdots u(t_{k-n_k-n_b+1}) \right]$$

$$\theta = \left[ a_1 \ \cdots \ a_{n_a} \ b_0 \ \cdots \ b_{n_b-1} \right]$$

- Prédiction de la sortie = fonction linéaire en les paramètres, le problème est donc celui d'une régression linéaire et les paramètres peuvent être cherchés par des procédures de moindres carrés.

## Erreur de prédiction – Cas d'un modèle ARX

- Soit le modèle ARX

$$y(t_k) = q^{n_k}\frac{B(q^{-1},\theta)}{A(q^{-1},\theta)}u(t_k) + \frac{1}{A(q^{-1},\theta)}e(t_k) \tag{8}$$

- **Prédiction de la sortie**

$$\hat{y}(t_k,\theta) = \big(1 - H^{-1}(q^{-1},\theta)\big)y(t_k) + H^{-1}(q^{-1},\theta)G(q^{-1},\theta)u(t_k)$$

$$\hat{y}(t_k,\theta) = \big(1 - A(q^{-1},\theta)\big)y(t_k) + q^{-n_k}B(q^{-1},\theta)u(t_k)$$

$$\hat{y}(t_k,\theta) = -a_1 y(t_{k-1}) - a_{n_a}y(t_{k-n_a}) + b_0 u(t_{k-n_k}) + \cdots + b_{n_{b-1}}u(t_{k-n_k-n_b+1})$$

$$\hat{y}(t_k,\theta) = \varphi^T(t_k)\theta$$

$$\varphi^T(t_k) = \big[-y(t_{k-1}) \ \cdots - y(t_{k-n_a}) \ u(t_{k-n_k}) \ \cdots u(t_{k-n_k-n_b+1})\big]$$

$$\theta = \big[a_1 \ \cdots \ a_{n_a} \ b_0 \ \cdots \ b_{n_b-1}\big]$$

- Prédiction de la sortie = fonction linéaire en les paramètres, le problème est donc celui d'une régression linéaire et les paramètres peuvent être cherchés par des procédures de moindres carrés.

# Méthodes d'estimation

## Erreur de prédiction – Cas d'un modèle ARX

- Soit le modèle ARX

$$y(t_k) = q^{n_k}\frac{B(q^{-1},\theta)}{A(q^{-1},\theta)}u(t_k) + \frac{1}{A(q^{-1},\theta)}e(t_k) \quad (8)$$

- **Prédiction de la sortie**

$$\hat{y}(t_k,\theta) = \big(1 - H^{-1}(q^{-1},\theta)\big)y(t_k) + H^{-1}(q^{-1},\theta)G(q^{-1},\theta)u(t_k)$$

$$\hat{y}(t_k,\theta) = \big(1 - A(q^{-1},\theta)\big)y(t_k) + q^{-n_k}B(q^{-1},\theta)u(t_k)$$

$$\hat{y}(t_k,\theta) = -a_1 y(t_{k-1}) - a_{n_a}y(t_{k-n_a}) + b_0 u(t_{k-n_k}) + \cdots + b_{n_{b-1}}u(t_{k-n_k-n_b+1})$$

$$\hat{y}(t_k,\theta) = \varphi^T(t_k)\theta$$

$$\varphi^T(t_k) = \begin{bmatrix} -y(t_{k-1}) & \cdots - y(t_{k-n_a}) & u(t_{k-n_k}) & \cdots u(t_{k-n_k-n_b+1}) \end{bmatrix}$$

$$\theta = \begin{bmatrix} a_1 & \cdots & a_{n_a} & b_0 & \cdots & b_{n_b-1} \end{bmatrix}$$

- Prédiction de la sortie = fonction linéaire en les paramètres, le problème est donc celui d'une régression linéaire et les paramètres peuvent être cherchés par des procédures de moindres carrés.

# Méthodes d'estimation

## Erreur de prédiction – Cas d'un modèle ARX

- Soit le modèle ARX

$$y(t_k) = q^{n_k} \frac{B(q^{-1}, \theta)}{A(q^{-1}, \theta)} u(t_k) + \frac{1}{A(q^{-1}, \theta)} e(t_k) \qquad (8)$$

- **Prédiction de la sortie**

$$\hat{y}(t_k, \theta) = \big(1 - H^{-1}(q^{-1}, \theta)\big) y(t_k) + H^{-1}(q^{-1}, \theta) G(q^{-1}, \theta) u(t_k)$$
$$\hat{y}(t_k, \theta) = \big(1 - A(q^{-1}, \theta)\big) y(t_k) + q^{-n_k} B(q^{-1}, \theta) u(t_k)$$
$$\hat{y}(t_k, \theta) = -a_1 y(t_{k-1}) - a_{n_a} y(t_{k-n_a}) + b_0 u(t_{k-n_k}) + \cdots + b_{n_{b-1}} u(t_{k-n_k-n_b+1})$$
$$\hat{y}(t_k, \theta) = \varphi^T(t_k)\theta$$
$$\varphi^T(t_k) = \big[ -y(t_{k-1}) \ \cdots - y(t_{k-n_a}) \ u(t_{k-n_k}) \ \cdots u(t_{k-n_k-n_b+1}) \big]$$
$$\theta = \big[ a_1 \ \cdots \ a_{n_a} \ b_0 \ \cdots \ b_{n_b-1} \big]$$

- Prédiction de la sortie = fonction linéaire en les paramètres, le problème est donc celui d'une régression linéaire et les paramètres peuvent être cherchés par des procédures de moindres carrés.

# Méthodes d'estimation

## Erreur de prédiction – Cas d'un modèle ARX

- Soit le modèle ARX

$$y(t_k) = q^{n_k} \frac{B(q^{-1},\theta)}{A(q^{-1},\theta)} u(t_k) + \frac{1}{A(q^{-1},\theta)} e(t_k) \qquad (8)$$

- **Prédiction de la sortie**

$$\hat{y}(t_k,\theta) = \big(1 - H^{-1}(q^{-1},\theta)\big) y(t_k) + H^{-1}(q^{-1},\theta) G(q^{-1},\theta) u(t_k)$$
$$\hat{y}(t_k,\theta) = \big(1 - A(q^{-1},\theta)\big) y(t_k) + q^{-n_k} B(q^{-1},\theta) u(t_k)$$
$$\hat{y}(t_k,\theta) = -a_1 y(t_{k-1}) - a_{n_a} y(t_{k-n_a}) + b_0 u(t_{k-n_k}) + \cdots + b_{n_{b-1}} u(t_{k-n_k-n_b+1})$$
$$\hat{y}(t_k,\theta) = \varphi^T(t_k)\theta$$
$$\varphi^T(t_k) = \big[ -y(t_{k-1}) \; \cdots - y(t_{k-n_a}) \; u(t_{k-n_k}) \; \cdots u(t_{k-n_k-n_b+1}) \big]$$
$$\theta = \big[ a_1 \; \cdots \; a_{n_a} \; b_0 \; \cdots \; b_{n_b-1} \big]$$

- Prédiction de la sortie = fonction linéaire en les paramètres, le problème est donc celui d'une régression linéaire et les paramètres peuvent être cherchés par des procédures de moindres carrés.

# Méthodes d'estimation

## Erreur de prédiction – Cas d'un modèle ARX

Soit un ensemble $Z^N$ de données d'E/S

$$Z^N = \{u(t_1), y(t_1), u(t_2), y(t_2), \cdots, u(t_N), y(t_N)\}$$

1. Calculer la prédiction de la sortie à partir des données jusqu'à l'instant $t_{k-1}$ (*modèle de régression linéaire*) $\boxed{\hat{y}(t_k, \theta) = \varphi^T(t_k)\theta}$
2. Former l'erreur de prédiction $\varepsilon(t_k) = y(t_k) - \hat{y}(t_k, \theta) = y(t_k) - \varphi^T(t_k)\theta$
3. Construire la fonction de coût (ou critère)
$$V(\theta, Z^N) = \frac{1}{N}\sum_{k=1}^{N}\varepsilon^2(t_k) = \frac{1}{N}\sum_{k=1}^{N}\left(y(t_k) - \varphi^T(t_k)\theta\right)^2$$
4. Ce critère est une fonction quadratique de $\theta$. Donc au minimum du critère, sa dérivé première par rapport à $\theta$ est nulle.

# Méthodes d'estimation

## Erreur de prédiction – Cas d'un modèle ARX

Soit un ensemble $Z^N$ de données d'E/S

$$Z^N = \{u(t_1), y(t_1), u(t_2), y(t_2), \cdots, u(t_N), y(t_N)\}$$

1. Calculer la prédiction de la sortie à partir des données jusqu'à l'instant $t_{k-1}$ (*modèle de régression linéaire*) $\hat{y}(t_k, \theta) = \varphi^T(t_k)\theta$
2. Former l'erreur de prédiction $\varepsilon(t_k) = y(t_k) - \hat{y}(t_k, \theta) = y(t_k) - \varphi^T(t_k)\theta$
3. Construire la fonction de coût (ou critère)
$$V(\theta, Z^N) = \frac{1}{N}\sum_{k=1}^{N} \varepsilon^2(t_k) = \frac{1}{N}\sum_{k=1}^{N} \left(y(t_k) - \varphi^T(t_k)\theta\right)^2$$
4. Ce critère est une fonction quadratique de $\theta$. Donc au minimum du critère, sa dérivé première par rapport à $\theta$ est nulle.

# Méthodes d'estimation

## Erreur de prédiction – Cas d'un modèle ARX

Soit un ensemble $Z^N$ de données d'E/S

$$Z^N = \{u(t_1), y(t_1), u(t_2), y(t_2), \cdots, u(t_N), y(t_N)\}$$

1. Calculer la prédiction de la sortie à partir des données jusqu'à l'instant $t_{k-1}$ (*modèle de régression linéaire*) $\boxed{\hat{y}(t_k, \theta) = \varphi^T(t_k)\theta}$
2. Former l'erreur de prédiction $\boxed{\varepsilon(t_k) = y(t_k) - \hat{y}(t_k, \theta) = y(t_k) - \varphi^T(t_k)\theta}$
3. Construire la fonction de coût (ou critère)
$$\boxed{V(\theta, Z^N) = \frac{1}{N}\sum_{k=1}^{N} \varepsilon^2(t_k) = \frac{1}{N}\sum_{k=1}^{N} \left(y(t_k) - \varphi^T(t_k)\theta\right)^2}$$
4. Ce critère est une fonction quadratique de $\theta$. Donc au minimum du critère, sa dérivé première par rapport à $\theta$ est nulle.

# Méthodes d'estimation

## Erreur de prédiction – Cas d'un modèle ARX

Soit un ensemble $Z^N$ de données d'E/S

$$Z^N = \{u(t_1), y(t_1), u(t_2), y(t_2), \cdots, u(t_N), y(t_N)\}$$

1. Calculer la prédiction de la sortie à partir des données jusqu'à l'instant $t_{k-1}$ (*modèle de régression linéaire*) $\hat{y}(t_k, \theta) = \varphi^T(t_k)\theta$
2. Former l'erreur de prédiction $\varepsilon(t_k) = y(t_k) - \hat{y}(t_k, \theta) = y(t_k) - \varphi^T(t_k)\theta$
3. Construire la fonction de coût (ou critère)
$$V(\theta, Z^N) = \frac{1}{N}\sum_{k=1}^{N} \varepsilon^2(t_k) = \frac{1}{N}\sum_{k=1}^{N} \left(y(t_k) - \varphi^T(t_k)\theta\right)^2$$
4. Ce critère est une fonction quadratique de $\theta$. Donc au minimum du critère, sa dérivé première par rapport à $\theta$ est nulle.

# Méthodes d'estimation

## Erreur de prédiction – Cas d'un modèle ARX

5. Calcul de la dérivée

$$\frac{\partial V(\theta, Z^N)}{\partial \theta} = \frac{2}{N}\sum_{k=1}^{N} \varphi(t_k)\big(y(t_k) - \varphi^T(t_k)\theta\big) = 0$$

$$\frac{2}{N}\sum_{k=1}^{N} \big(\varphi(t_k)y(t_k) - \varphi(t_k)\varphi^T(t_k)\theta\big) = 0$$

6. Déduction : estimateur des moindres carrés

$$\hat{\theta}_{mc} = \left[\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right]^{-1} \sum_{k=1}^{N} \varphi(t_k)y(t_k)$$

à condition que l'inverse existe.

# Méthodes d'estimation

## Erreur de prédiction – Cas d'un modèle ARX

5. Calcul de la dérivée

$$\frac{\partial V(\theta, Z^N)}{\partial \theta} = \frac{2}{N}\sum_{k=1}^{N} \varphi(t_k)\big(y(t_k) - \varphi^T(t_k)\theta\big) = 0$$

$$\frac{2}{N}\sum_{k=1}^{N} \big(\varphi(t_k)y(t_k) - \varphi(t_k)\varphi^T(t_k)\theta\big) = 0$$

6. Déduction : estimateur des moindres carrés

$$\boxed{\hat{\theta}_{mc} = \left[\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right]^{-1} \sum_{k=1}^{N} \varphi(t_k)y(t_k)}$$

à condition que l'inverse existe.

# Méthodes d'estimation

## Erreur de prédiction – Cas d'un modèle ARX, formulation matricielle

- La forme matricielle est plus facilement utilisée. L'erreur de prédiction (ou résidus) s'écrit

$$\begin{bmatrix} \varepsilon(t_1) \\ \vdots \\ \varepsilon(t_N) \end{bmatrix} = \begin{bmatrix} y(t_1) \\ \vdots \\ y(t_N) \end{bmatrix} - \begin{bmatrix} \varphi^T(t_1) \\ \vdots \\ \varphi^T(t_N) \end{bmatrix} \theta$$

$$\varepsilon(t_k) = Y_N - \Phi_N \theta$$

- Le critère s'écrit :

$$V(\theta, Z^N) = \frac{1}{N} \sum_{k=1}^{N} \varepsilon^2(t_k) = \frac{1}{N} \varepsilon_N^T \varepsilon_N = \frac{1}{N} (Y_N - \Phi_N \theta)^T (Y_N - \Phi_N \theta)$$

- La condition de gradient nul conduit à :

$$\frac{\partial V(\theta, Z^N)}{\partial \theta} = -\Phi_N^T Y_N + \theta^T (\Phi_N^T \Phi_N) = 0$$

- Déduction de l'estimateur de moindres carrés sous forme matricielle

$$\hat{\theta}_{mc} = (\Phi_N^T \Phi_N)^{-1} \Phi_N^T Y_N$$

# Méthodes d'estimation

## Erreur de prédiction – Cas d'un modèle ARX, formulation matricielle

- La forme matricielle est plus facilement utilisée. L'erreur de prédiction (ou résidus) s'écrit

$$\begin{bmatrix} \varepsilon(t_1) \\ \vdots \\ \varepsilon(t_N) \end{bmatrix} = \begin{bmatrix} y(t_1) \\ \vdots \\ y(t_N) \end{bmatrix} - \begin{bmatrix} \varphi^T(t_1) \\ \vdots \\ \varphi^T(t_N) \end{bmatrix} \theta$$

$$\varepsilon(t_k) = Y_N - \Phi_N \theta$$

- Le critère s'écrit :

$$V(\theta, Z^N) = \frac{1}{N} \sum_{k=1}^{N} \varepsilon^2(t_k) = \frac{1}{N} \varepsilon_N^T \varepsilon_N = \frac{1}{N} \left(Y_N - \Phi_N \theta\right)^T \left(Y_N - \Phi_N \theta\right)$$

- La condition de gradient nul conduit à :

$$\frac{\partial V(\theta, Z^N)}{\partial \theta} = -\Phi_N^T Y_N + \theta^T \left(\Phi_N^T \Phi_N\right) = 0$$

- Déduction de l'estimateur de moindres carrés sous forme matricielle

$$\hat{\theta}_{mc} = \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T Y_N$$

# Méthodes d'estimation

## Erreur de prédiction – Cas d'un modèle ARX, formulation matricielle

- La forme matricielle est plus facilement utilisée. L'erreur de prédiction (ou résidus) s'écrit

$$\begin{bmatrix} \varepsilon(t_1) \\ \vdots \\ \varepsilon(t_N) \end{bmatrix} = \begin{bmatrix} y(t_1) \\ \vdots \\ y(t_N) \end{bmatrix} - \begin{bmatrix} \varphi^T(t_1) \\ \vdots \\ \varphi^T(t_N) \end{bmatrix} \theta$$

$$\varepsilon(t_k) = Y_N - \Phi_N \theta$$

- Le critère s'écrit :

$$V(\theta, Z^N) = \frac{1}{N} \sum_{k=1}^{N} \varepsilon^2(t_k) = \frac{1}{N} \varepsilon_N^T \varepsilon_N = \frac{1}{N} (Y_N - \Phi_N \theta)^T (Y_N - \Phi_N \theta)$$

- La condition de gradient nul conduit à :

$$\frac{\partial V(\theta, Z^N)}{\partial \theta} = -\Phi_N^T Y_N + \theta^T \left(\Phi_N^T \Phi_N\right) = 0$$

- Déduction de l'estimateur de moindres carrés sous forme matricielle

$$\hat{\theta}_{mc} = \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T Y_N$$

## Erreur de prédiction – Cas d'un modèle ARX, formulation matricielle

- La forme matricielle est plus facilement utilisée. L'erreur de prédiction (ou résidus) s'écrit

$$\begin{bmatrix} \varepsilon(t_1) \\ \vdots \\ \varepsilon(t_N) \end{bmatrix} = \begin{bmatrix} y(t_1) \\ \vdots \\ y(t_N) \end{bmatrix} - \begin{bmatrix} \varphi^T(t_1) \\ \vdots \\ \varphi^T(t_N) \end{bmatrix} \theta$$

$$\varepsilon(t_k) = Y_N - \Phi_N \theta$$

- Le critère s'écrit :

$$V(\theta, Z^N) = \frac{1}{N} \sum_{k=1}^{N} \varepsilon^2(t_k) = \frac{1}{N} \varepsilon_N^T \varepsilon_N = \frac{1}{N} \left(Y_N - \Phi_N \theta\right)^T \left(Y_N - \Phi_N \theta\right)$$

- La condition de gradient nul conduit à :

$$\frac{\partial V(\theta, Z^N)}{\partial \theta} = -\Phi_N^T Y_N + \theta^T \left(\Phi_N^T \Phi_N\right) = 0$$

- Déduction de l'estimateur de moindres carrés sous forme matricielle

$$\boxed{\hat{\theta}_{mc} = \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T Y_N}$$

# Méthodes d'estimation

## Erreur de prédiction – Cas d'un modèle ARX, aspects numériques

$$\hat{\theta}_{mc} = \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T Y_N$$

**Remarque sur l'implantation numérique :**

- Il est fortement *déconseillé* d'implémenter ces formules directement, l'inversion de la matrice $\left(\Phi_N^T \Phi_N\right)$ entraîne des erreurs numériques (dues à un mauvais conditionnement).
- *Alternatives* à l'inversion de matrice :
  - ◇ décomposition en valeurs singulières (SVD)
  - ◇ décomposition QR

  $\longrightarrow$ mieux adaptées pour les problèmes mal conditionnés (cad où la matrice à inverse est numériquement mal conditionnée)
- Sous Matlab : `pinv` ou

$$\hat{\theta}_{mc} = \Phi_N \backslash Y_N$$

# Méthodes d'estimation

## Erreur de prédiction – Cas d'un modèle ARX, aspects numériques

$$\hat{\theta}_{mc} = \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T Y_N$$

**Remarque sur l'implantation numérique :**

- Il est fortement *déconseillé* d'implémenter ces formules directement, l'inversion de la matrice $\left(\Phi_N^T \Phi_N\right)$ entraîne des erreurs numériques (dues à un mauvais conditionnement).

- *Alternatives* à l'inversion de matrice :
  - ◇ décomposition en valeurs singulières (SVD)
  - ◇ décomposition QR

  $\longrightarrow$ mieux adaptées pour les problèmes mal conditionnés (cad où la matrice à inverse est numériquement mal conditionnée)

- Sous Matlab : `pinv` ou

$$\hat{\theta}_{mc} = \Phi_N \backslash Y_N$$

# Méthodes d'estimation

## Erreur de prédiction – Cas d'un modèle ARX, aspects numériques

$$\hat{\theta}_{mc} = \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T Y_N$$

**Remarque sur l'implantation numérique :**

- Il est fortement *déconseillé* d'implémenter ces formules directement, l'inversion de la matrice $\left(\Phi_N^T \Phi_N\right)$ entraîne des erreurs numériques (dues à un mauvais conditionnement).

- *Alternatives* à l'inversion de matrice :
  - ◇ décomposition en valeurs singulières (SVD)
  - ◇ décomposition QR

  $\longrightarrow$ mieux adaptées pour les problèmes mal conditionnés (cad où la matrice à inverse est numériquement mal conditionnée)

- Sous Matlab : `pinv` ou

$$\hat{\theta}_{mc} = \Phi_N \backslash Y_N$$

# Méthodes d'estimation

## Exemple – Estimation d'un modèle ARX d'ordre 1

- On cherche à identifier un système du premier ordre représenté par :

$$G(q^{-1},\theta) = \frac{B(q^{-1},\theta)}{A(q^{-1},\theta)} = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}}$$

- Hypothèse : structure de modèle de type ARX

$$y(t_k,\theta) = \frac{B(q^{-1},\theta)}{A(q^{-1},\theta)} u(t_k,\theta) + \frac{1}{A(q^{-1},\theta)} e(t_k,\theta)$$

$$y(t_k,\theta) = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}} u(t_k,\theta) + \frac{1}{1 + a_1 q^{-1}} e(t_k,\theta)$$

$$(1 + a_1 q^{-1}) y(t_k,\theta) = b_1 q^{-1} u(t_k,\theta) + e(t_k,\theta)$$

$$y(t_k,\theta) = -a_1 q^{-1} y(t_k,\theta) + b_1 q^{-1} u(t_k,\theta) + e(t_k,\theta)$$

$$y(t_k,\theta) = -a_1 y(t_{k-1},\theta) + b_1 u(t_{k-1},\theta) + e(t_k,\theta)$$

$$y(t_k,\theta) = \begin{bmatrix} -y(t_{k-1},\theta) & u(t_{k-1},\theta) \end{bmatrix} \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} + e(t_k,\theta)$$

$$y(t_k,\theta) = \varphi^T(t_k)\theta + e(t_k,\theta)$$

# Méthodes d'estimation

## Exemple – Estimation d'un modèle ARX d'ordre 1

- On cherche à identifier un système du premier ordre représenté par :

$$G(q^{-1},\theta) = \frac{B(q^{-1},\theta)}{A(q^{-1},\theta)} = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}}$$

- Hypothèse : structure de modèle de type ARX

$$y(t_k,\theta) = \frac{B(q^{-1},\theta)}{A(q^{-1},\theta)} u(t_k,\theta) + \frac{1}{A(q^{-1},\theta)} e(t_k,\theta)$$

$$\begin{aligned}
y(t_k,\theta) &= \frac{b_1 q^{-1}}{1 + a_1 q^{-1}} u(t_k,\theta) + \frac{1}{1 + a_1 q^{-1}} e(t_k,\theta) \\
(1 + a_1 q^{-1}) y(t_k,\theta) &= b_1 q^{-1} u(t_k,\theta) + e(t_k,\theta) \\
y(t_k,\theta) &= -a_1 q^{-1} y(t_k,\theta) + b_1 q^{-1} u(t_k,\theta) + e(t_k,\theta) \\
y(t_k,\theta) &= -a_1 y(t_{k-1},\theta) + b_1 u(t_{k-1},\theta) + e(t_k,\theta) \\
y(t_k,\theta) &= \begin{bmatrix} -y(t_{k-1},\theta) & u(t_{k-1},\theta) \end{bmatrix} \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} + e(t_k,\theta) \\
y(t_k,\theta) &= \varphi^T(t_k)\theta + e(t_k,\theta)
\end{aligned}$$

# Méthodes d'estimation

## Exemple – Estimation d'un modèle ARX d'ordre 1

- On cherche à identifier un système du premier ordre représenté par :

$$G(q^{-1},\theta) = \frac{B(q^{-1},\theta)}{A(q^{-1},\theta)} = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}}$$

- Hypothèse : structure de modèle de type ARX

$$y(t_k,\theta) = \frac{B(q^{-1},\theta)}{A(q^{-1},\theta)} u(t_k,\theta) + \frac{1}{A(q^{-1},\theta)} e(t_k,\theta)$$

$$y(t_k,\theta) = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}} u(t_k,\theta) + \frac{1}{1 + a_1 q^{-1}} e(t_k,\theta)$$

$$(1 + a_1 q^{-1}) y(t_k,\theta) = b_1 q^{-1} u(t_k,\theta) + e(t_k,\theta)$$

$$y(t_k,\theta) = -a_1 q^{-1} y(t_k,\theta) + b_1 q^{-1} u(t_k,\theta) + e(t_k,\theta)$$

$$y(t_k,\theta) = -a_1 y(t_{k-1},\theta) + b_1 u(t_{k-1},\theta) + e(t_k,\theta)$$

$$y(t_k,\theta) = [-y(t_{k-1},\theta) \quad u(t_{k-1},\theta)] \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} + e(t_k,\theta)$$

$$y(t_k,\theta) = \varphi^T(t_k)\theta + e(t_k,\theta)$$

# Méthodes d'estimation

## Exemple – Estimation d'un modèle ARX d'ordre 1

- On cherche à identifier un système du premier ordre représenté par :

$$G(q^{-1},\theta) = \frac{B(q^{-1},\theta)}{A(q^{-1},\theta)} = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}}$$

- Hypothèse : structure de modèle de type ARX

$$y(t_k,\theta) = \frac{B(q^{-1},\theta)}{A(q^{-1},\theta)} u(t_k,\theta) + \frac{1}{A(q^{-1},\theta)} e(t_k,\theta)$$

$$y(t_k,\theta) = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}} u(t_k,\theta) + \frac{1}{1 + a_1 q^{-1}} e(t_k,\theta)$$

$$(1 + a_1 q^{-1}) y(t_k,\theta) = b_1 q^{-1} u(t_k,\theta) + e(t_k,\theta)$$

$$y(t_k,\theta) = -a_1 q^{-1} y(t_k,\theta) + b_1 q^{-1} u(t_k,\theta) + e(t_k,\theta)$$

$$y(t_k,\theta) = -a_1 y(t_{k-1},\theta) + b_1 u(t_{k-1},\theta) + e(t_k,\theta)$$

$$y(t_k,\theta) = \left[-y(t_{k-1},\theta) \quad u(t_{k-1},\theta)\right] \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} + e(t_k,\theta)$$

$$y(t_k,\theta) = \varphi^T(t_k)\theta + e(t_k,\theta)$$

## Exemple – Estimation d'un modèle ARX d'ordre 1

- On cherche à identifier un système du premier ordre représenté par :

$$G(q^{-1},\theta) = \frac{B(q^{-1},\theta)}{A(q^{-1},\theta)} = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}}$$

- Hypothèse : structure de modèle de type ARX

$$\begin{aligned}
y(t_k,\theta) &= \frac{B(q^{-1},\theta)}{A(q^{-1},\theta)} u(t_k,\theta) + \frac{1}{A(q^{-1},\theta)} e(t_k,\theta) \\
y(t_k,\theta) &= \frac{b_1 q^{-1}}{1 + a_1 q^{-1}} u(t_k,\theta) + \frac{1}{1 + a_1 q^{-1}} e(t_k,\theta) \\
(1 + a_1 q^{-1}) y(t_k,\theta) &= b_1 q^{-1} u(t_k,\theta) + e(t_k,\theta) \\
y(t_k,\theta) &= -a_1 q^{-1} y(t_k,\theta) + b_1 q^{-1} u(t_k,\theta) + e(t_k,\theta) \\
y(t_k,\theta) &= -a_1 y(t_{k-1},\theta) + b_1 u(t_{k-1},\theta) + e(t_k,\theta) \\
y(t_k,\theta) &= \left[-y(t_{k-1},\theta) \quad u(t_{k-1},\theta)\right] \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} + e(t_k,\theta) \\
y(t_k,\theta) &= \varphi^T(t_k)\theta + e(t_k,\theta)
\end{aligned}$$

# Méthodes d'estimation

## Exemple – Estimation d'un modèle ARX d'ordre 1

- On cherche à identifier un système du premier ordre représenté par :

$$G(q^{-1},\theta) = \frac{B(q^{-1},\theta)}{A(q^{-1},\theta)} = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}}$$

- Hypothèse : structure de modèle de type ARX

$$\begin{aligned}
y(t_k,\theta) &= \frac{B(q^{-1},\theta)}{A(q^{-1},\theta)} u(t_k,\theta) + \frac{1}{A(q^{-1},\theta)} e(t_k,\theta) \\
y(t_k,\theta) &= \frac{b_1 q^{-1}}{1 + a_1 q^{-1}} u(t_k,\theta) + \frac{1}{1 + a_1 q^{-1}} e(t_k,\theta) \\
(1 + a_1 q^{-1}) y(t_k,\theta) &= b_1 q^{-1} u(t_k,\theta) + e(t_k,\theta) \\
y(t_k,\theta) &= -a_1 q^{-1} y(t_k,\theta) + b_1 q^{-1} u(t_k,\theta) + e(t_k,\theta) \\
y(t_k,\theta) &= -a_1 y(t_{k-1},\theta) + b_1 u(t_{k-1},\theta) + e(t_k,\theta)
\end{aligned}$$

$$y(t_k,\theta) = \begin{bmatrix} -y(t_{k-1},\theta) & u(t_{k-1},\theta) \end{bmatrix} \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} + e(t_k,\theta)$$

$$y(t_k,\theta) = \varphi^T(t_k)\theta + e(t_k,\theta)$$

# Méthodes d'estimation

## Exemple – Estimation d'un modèle ARX d'ordre 1

- On cherche à identifier un système du premier ordre représenté par :

$$G(q^{-1},\theta) = \frac{B(q^{-1},\theta)}{A(q^{-1},\theta)} = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}}$$

- Hypothèse : structure de modèle de type ARX

$$\begin{aligned}
y(t_k,\theta) &= \frac{B(q^{-1},\theta)}{A(q^{-1},\theta)} u(t_k,\theta) + \frac{1}{A(q^{-1},\theta)} e(t_k,\theta) \\
y(t_k,\theta) &= \frac{b_1 q^{-1}}{1 + a_1 q^{-1}} u(t_k,\theta) + \frac{1}{1 + a_1 q^{-1}} e(t_k,\theta) \\
(1 + a_1 q^{-1}) y(t_k,\theta) &= b_1 q^{-1} u(t_k,\theta) + e(t_k,\theta) \\
y(t_k,\theta) &= -a_1 q^{-1} y(t_k,\theta) + b_1 q^{-1} u(t_k,\theta) + e(t_k,\theta) \\
y(t_k,\theta) &= -a_1 y(t_{k-1},\theta) + b_1 u(t_{k-1},\theta) + e(t_k,\theta) \\
y(t_k,\theta) &= [-y(t_{k-1},\theta) \quad u(t_{k-1},\theta)] \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} + e(t_k,\theta) \\
y(t_k,\theta) &= \varphi^T(t_k)\theta + e(t_k,\theta)
\end{aligned}$$

# Méthodes d'estimation

## Exemple – Estimation d'un modèle ARX d'ordre 1

- Régression linéaire en les paramètres :

$$
\begin{aligned}
y(t_k,\theta) &= \varphi^T(t_k)\theta + e(t_k,\theta)\\
\text{avec : } \ \varphi^T(t_k) &= [-y(t_{k-1},\theta) \ \ u(t_{k-1},\theta)]\\
\text{et : } \ \theta &= \begin{pmatrix} a_1 \\ b_1 \end{pmatrix}
\end{aligned}
$$

Pour $N$ données ($k=1$ à $N$), on obtient l'écriture matricielle suivante :

$$
\begin{bmatrix} y(t_2) \\ \vdots \\ y(t_N) \end{bmatrix} = \begin{bmatrix} -y(t_1) & u(t_1) \\ \vdots & \vdots \\ -y(t_{N-1}) & u(t_{N-1}) \end{bmatrix} \theta + \begin{bmatrix} e(t_2) \\ \vdots \\ e(t_N) \end{bmatrix}
$$

- On en déduit :

$$
\hat{\theta}_{mc} = \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T Y_N
$$

# Méthodes d'estimation

## Exemple – Estimation d'un modèle ARX d'ordre 1

- Régression linéaire en les paramètres :

$$y(t_k,\theta) = \varphi^T(t_k)\theta + e(t_k,\theta)$$
$$\text{avec : } \varphi^T(t_k) = [-y(t_{k-1},\theta) \;\; u(t_{k-1},\theta)]$$
$$\text{et : } \theta = \begin{pmatrix} a_1 \\ b_1 \end{pmatrix}$$

Pour $N$ données ($k = 1$ à $N$), on obtient l'écriture matricielle suivante :

$$\begin{bmatrix} y(t_2) \\ \vdots \\ y(t_N) \end{bmatrix} = \begin{bmatrix} -y(t_1) & u(t_1) \\ \vdots & \vdots \\ -y(t_{N-1}) & u(t_{N-1}) \end{bmatrix} \theta + \begin{bmatrix} e(t_2) \\ \vdots \\ e(t_N) \end{bmatrix}$$

- On en déduit :

$$\hat{\theta}_{mc} = \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T Y_N$$

# Méthodes d'estimation

## Exemple – Estimation d'un modèle ARX d'ordre 1

- Remarque : sous Matlab l'expression

$$\begin{bmatrix} y(t_2) \\ \vdots \\ y(t_N) \end{bmatrix} = \begin{bmatrix} -y(t_1) & u(t_1) \\ \vdots & \vdots \\ -y(t_{N-1}) & u(t_{N-1}) \end{bmatrix} \theta + \begin{bmatrix} e(t_2) \\ \vdots \\ e(t_N) \end{bmatrix}$$

$$\hat{\theta}_{mc} = \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T Y_N$$

se code :

```
Phi = [-y(1:N-1) u(1:N-1)];
Y = y(2:N);
theta = Phi \ Y
```

## Test d'un estimateur

Comment tester le bon fonctionnement d'un estimateur sous Matlab ?

- Etape 1 : tester l'estimateur sous un exemple académique
- Etape 2 : si le test est concluant, alors utilisation de l'estimateur sur des données réelles

Etapes à suivre pour le test d'un estimateur sur un exemple

1. choix d'un *système "vrai"* à identifier, par exemple : $G(q^{-1}, \theta) = \frac{b_1 q^{-1}}{1+a_1 q^{-1}}$
2. choix d'une *structure* de modèle, par exemple ARX
3. choix du *signal d'excitation*, par exemple SBPA ou bruit blanc

$\longrightarrow$ On obtient alors un jeu de données d'E/S $[y(t) \;\; u(t)]$ sur lequel tester l'estimateur :

4. former $u(t)$, simuler le système "vrai" pour obtenir $y(t)$, ajouter du bruit $e(t)$
5. représenter les données à l'écran pour vérification visuelle
6. former les matrices $\Phi_N$ et $Y_N$
7. calculer $\hat{\theta}_{mc}$
8. comparer aux paramètres du modèle "vrai" pour *validation*.

# Test d'un estimateur

Comment tester le bon fonctionnement d'un estimateur sous Matlab ?

- Etape 1 : tester l'estimateur sous un exemple académique
- Etape 2 : si le test est concluant, alors utilisation de l'estimateur sur des données réelles

Etapes à suivre pour le test d'un estimateur sur un exemple

1. choix d'un *système "vrai"* à identifier, par exemple : $G(q^{-1}, \theta) = \frac{b_1 q^{-1}}{1+a_1 q^{-1}}$
2. choix d'une *structure* de modèle, par exemple ARX
3. choix du *signal d'excitation*, par exemple SBPA ou bruit blanc

$\longrightarrow$ On obtient alors un jeu de données d'E/S $[y(t) \quad u(t)]$ sur lequel tester l'estimateur :

4. former $u(t)$, simuler le système "vrai" pour obtenir $y(t)$, ajouter du bruit $e(t)$
5. représenter les données à l'écran pour vérification visuelle
6. former les matrices $\Phi_N$ et $Y_N$
7. calculer $\hat{\theta}_{mc}$
8. comparer aux paramètres du modèle "vrai" pour *validation*.

## Test d'un estimateur

Comment tester le bon fonctionnement d'un estimateur sous Matlab ?

- Etape 1 : tester l'estimateur sous un exemple académique
- Etape 2 : si le test est concluant, alors utilisation de l'estimateur sur des données réelles

Etapes à suivre pour le test d'un estimateur sur un exemple

1. choix d'un *système "vrai"* à identifier, par exemple : $G(q^{-1}, \theta) = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}}$
2. choix d'une *structure* de modèle, par exemple ARX
3. choix du *signal d'excitation*, par exemple SBPA ou bruit blanc

$\longrightarrow$ On obtient alors un jeu de données d'E/S $[y(t) \quad u(t)]$ sur lequel tester l'estimateur :

4. former $u(t)$, simuler le système "vrai" pour obtenir $y(t)$, ajouter du bruit $e(t)$
5. représenter les données à l'écran pour vérification visuelle
6. former les matrices $\Phi_N$ et $Y_N$
7. calculer $\hat{\theta}_{mc}$
8. comparer aux paramètres du modèle "vrai" pour *validation*.

# Test d'un estimateur

Pour aller plus loin

- Test de l'efficacité face au bruit :
  - ajouter du bruit en sortie sous une structure ARX et tester l'estimateur sur les données bruitées
- Test de l'efficacité face à une erreur de structure de modèle :
  - changer la structure du modèle "vrai" pour observer les performances de l'estimateur ARX face à un non-respect de l'hypothèse de structure
  - exemple : choisir une structure de modèle OE pour le modèle "vrai".

Exercice : créez vos données puis testez l'estimateur des moindres carrés sur l'exemple du 1er ordre suivant :

$$G(q^{-1},\theta) = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}} = \frac{0.2q^{-1}}{1 - 0.8q^{-1}}$$

avec une structure ARX puis OE pour le système "vrai".

# Test d'un estimateur

Pour aller plus loin

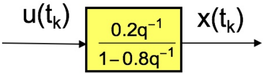
- Test de l'efficacité face au bruit :
  - ◇ ajouter du bruit en sortie sous une structure ARX et tester l'estimateur sur les données bruitées
- Test de l'efficacité face à une erreur de structure de modèle :
  - ◇ changer la structure du modèle "vrai" pour observer les performances de l'estimateur ARX face à un non-respect de l'hypothèse de structure
  - ◇ exemple : choisir une structure de modèle OE pour le modèle "vrai".

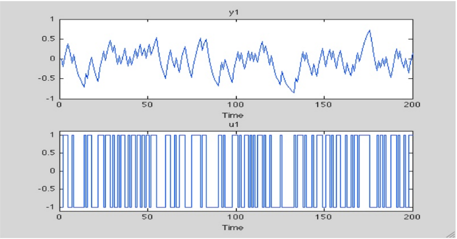
Exercice : créez vos données puis testez l'estimateur des moindres carrés sur l'exemple du 1er ordre suivant :

$$G(q^{-1}, \theta) = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}} = \frac{0.2 q^{-1}}{1 - 0.8 q^{-1}}$$

avec une structure ARX puis OE pour le système "vrai".

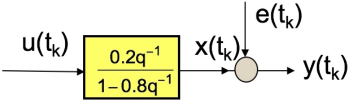
# Implémentation sous Matlab, exemple 1er ordre



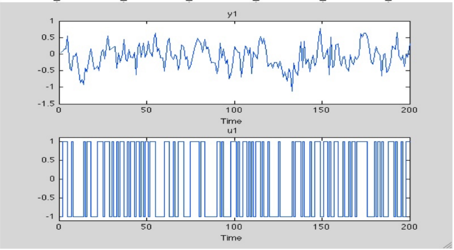




```
% Simulation du système non bruité
B=[0 0.2];
A=[1 -0.8];
N=200;
S = idpoly(1,B,1,1,A);            %y=B/Au
u=sign(randn(N,1));
x=sim(S,u);                       % simulation de la sortie non bruitée
% Estimation des paramètres par MC
Phi=[-x(1:N-1) u(1:N-1)];         % Matrice de régression
Y=ydet(2:N);                      % Régresseur
theta_mc=(Phi\Y)'                 % Estimation par MC
ans   -0.8000   0.2000            % On estime parfaitement bien les 2 paramètres
                                  % Normal résolution d'un système sur-déterminé
                                  % dans un contexte non bruité
% voir aussi la fonction arx de la SID
% datadet=iddata(x,u); idplot(datadet);
%Marx=arx(datadet,[1 1 1])
```

# Implémentation sous Matlab, exemple 1er ordre




$$y(t_k) = \frac{B(q^{-1})}{A(q^{-1})} u(t_k) + e(t_k)$$




```
% Simulation du système bruité
randn('state',sum(100*clock));
e=0.2*randn(N,1);
y=sim(S,[u e]);        % simulation de la sortie bruitée
dataS= iddata(y,u); % Objet de données iddata avec comme sortie et u comme entrée
idplot(dataS)
% Estimation des paramètres par MC
Phi=[-y(1:N-1) u(1:N-1)];      % Matrice de régression
Y=y(2:N);                      % Régresseur
theta_mc=(Phi\Y)';             % Estimation par MC
ans   -0.4675   0.1898         % Les paramètres sont erronés

% voir aussi la fonction arx de la SID
% Marx=arx(dataS,[1 1 1])
```

# Plan

# Prédicteur pour un modèle OE

- Cas général :

$$\hat{y}(t_k,\theta) = \big(1 - H^{-1}(q^{-1},\theta)\big)y(t_k) + H^{-1}(q^{-1},\theta)G(q^{-1},\theta)u(t_k)$$

- Structure de modèle OE :

$$G(q^{-1},\theta) = q^{-n_k}\frac{B(q^{-1},\theta)}{F(q^{-1},\theta)} \quad \text{et} \quad H(q^{-1},\theta) = 1$$

- Prédicteur, cas OE :

$$\hat{y}(t_k,\theta) = q^{-n_k}\frac{B(q^{-1},\theta)}{F(q^{-1},\theta)}u(t_k)$$

$$\hat{y}(t_k,\theta) = \varphi^T(t_k)\theta$$

Expression non linéaire en $\theta$
avec :

$$\varphi^T(t_k) = \begin{bmatrix} -\hat{y}(t_{k-1}) & \cdots & -\hat{y}(t_{k-n_f}) & u(t_{k-n_k}) & \cdots & u(t_{k-n_k-n_b+1}) \end{bmatrix} \tag{9}$$

$$\theta = \begin{pmatrix} f_1 & \cdots & f_{n_f} & b_0 & \cdots & b_{n_b-1} \end{pmatrix}^T \tag{10}$$

## Prédicteur pour un modèle OE

- Cas général :

$$\hat{y}(t_k,\theta) = \big(1 - H^{-1}(q^{-1},\theta)\big)y(t_k) + H^{-1}(q^{-1},\theta)G(q^{-1},\theta)u(t_k)$$

- Structure de modèle OE :

$$G(q^{-1},\theta) = q^{-n_k}\frac{B(q^{-1},\theta)}{F(q^{-1},\theta)} \quad \text{et} \quad H(q^{-1},\theta) = 1$$

- Prédicteur, cas OE :

$$\hat{y}(t_k,\theta) = q^{-n_k}\frac{B(q^{-1},\theta)}{F(q^{-1},\theta)}u(t_k)$$

$$\hat{y}(t_k,\theta) = \varphi^T(t_k)\theta$$

Expression non linéaire en $\theta$
avec :

$$\varphi^T(t_k) = \left[-\hat{y}(t_{k-1}) \ \cdots - \hat{y}(t_{k-n_f}) \ u(t_{k-n_k}) \ \cdots u(t_{k-n_k-n_b+1})\right] \tag{9}$$

$$\theta = \left(f_1 \ \cdots \ f_{n_f} \ b_0 \ \cdots \ b_{n_b-1}\right)^T \tag{10}$$

## Prédicteur pour un modèle OE

- Cas général :

$$\hat{y}(t_k,\theta) = \big(1 - H^{-1}(q^{-1},\theta)\big)y(t_k) + H^{-1}(q^{-1},\theta)G(q^{-1},\theta)u(t_k)$$

- Structure de modèle OE :

$$G(q^{-1},\theta) = q^{-n_k}\frac{B(q^{-1},\theta)}{F(q^{-1},\theta)} \quad \text{et} \quad H(q^{-1},\theta) = 1$$

- Prédicteur, cas OE :

$$\hat{y}(t_k,\theta) = q^{-n_k}\frac{B(q^{-1},\theta)}{F(q^{-1},\theta)}u(t_k)$$
$$\hat{y}(t_k,\theta) = \varphi^T(t_k)\theta$$

Expression non linéaire en $\theta$
avec :

$$\varphi^T(t_k) = \big[-\hat{y}(t_{k-1}) \ \cdots - \hat{y}(t_{k-n_f}) \ u(t_{k-n_k}) \ \cdots u(t_{k-n_k-n_b+1})\big] \tag{9}$$
$$\theta = \big(f_1 \ \cdots \ f_{n_f} \ b_0 \ \cdots \ b_{n_b-1}\big)^T \tag{10}$$

## Prédicteur pour un modèle OE

- Cas général :

$$\hat{y}(t_k, \theta) = \left(1 - H^{-1}(q^{-1}, \theta)\right) y(t_k) + H^{-1}(q^{-1}, \theta) G(q^{-1}, \theta) u(t_k)$$

- Structure de modèle OE :

$$G(q^{-1}, \theta) = q^{-n_k} \frac{B(q^{-1}, \theta)}{F(q^{-1}, \theta)} \quad \text{et} \quad H(q^{-1}, \theta) = 1$$

- Prédicteur, cas OE :

$$\hat{y}(t_k, \theta) = q^{-n_k} \frac{B(q^{-1}, \theta)}{F(q^{-1}, \theta)} u(t_k)$$
$$\hat{y}(t_k, \theta) = \varphi^T(t_k) \theta$$

Expression non linéaire en $\theta$
avec :

$$\varphi^T(t_k) = \left[ -\hat{y}(t_{k-1}) \ \cdots - \hat{y}(t_{k-n_f}) \ u(t_{k-n_k}) \ \cdots u(t_{k-n_k-n_b+1}) \right] \tag{9}$$
$$\theta = \left( f_1 \ \cdots \ f_{n_f} \ b_0 \ \cdots \ b_{n_b-1} \right)^T \tag{10}$$

# Prédicteur pour un modèle OE

- L'estimée $\hat{\theta}$ est solution de :

$$\hat{\theta} = \arg\min_{\theta} \underbrace{\frac{1}{N}\sum_{t=1}^{N}\left(y(t_k - \varphi^T(t_k,\theta)\right)^2}_{V(\theta, Z^N)}$$

- Difficulté : le minimum du critère est atteint lorsque la dérivée s'annule :

$$\left.\frac{\partial V_N(\theta, Z^N)}{\partial \theta}\right|_{\theta=\hat{\theta}} = 0$$

  or la dérivée est une expression compliquée non linéaire en $\theta$
  $\Longrightarrow$ Présence de plusieurs minima locaux (la dérivée s'annule pour plusieurs valeurs de $\theta$)

- Conséquence : la solution $\hat{\theta}$ est calculée par des méthodes itériatives (Gauss Newton, Gradient,...)
- Problème majeur : estimer un minimum local (*initialisation très importante*)

# Prédicteur pour un modèle OE

- L'estimée $\hat{\theta}$ est solution de :

$$\hat{\theta} = \arg\min_{\theta} \underbrace{\frac{1}{N}\sum_{t=1}^{N}\left(y(t_k - \varphi^T(t_k,\theta)\right)^2}_{V(\theta, Z^N)}$$

- Difficulté : le minimum du critère est atteint lorsque la dérivée s'annule :

$$\left.\frac{\partial V_N(\theta, Z^N)}{\partial\theta}\right|_{\theta=\hat{\theta}} = 0$$

  or la dérivée est une expression compliquée non linéaire en $\theta$
  $\Longrightarrow$ Présence de plusieurs minima locaux (la dérivée s'annule pour plusieurs valeurs de $\theta$)

- Conséquence : la solution $\hat{\theta}$ est calculée par des méthodes itériatives (Gauss Newton, Gradient,...)
- Problème majeur : estimer un minimum local (*initialisation très importante*)

## Prédicteur pour un modèle OE

- L'estimée $\hat{\theta}$ est solution de :

$$\hat{\theta} = \arg\min_{\theta} \underbrace{\frac{1}{N}\sum_{t=1}^{N}\left(y(t_k - \varphi^T(t_k,\theta)\right)^2}_{V(\theta, Z^N)}$$

- Difficulté : le minimum du critère est atteint lorsque la dérivée s'annule :

$$\left.\frac{\partial V_N(\theta, Z^N)}{\partial\theta}\right|_{\theta=\hat{\theta}} = 0$$

  or la dérivée est une expression compliquée non linéaire en $\theta$
  $\Longrightarrow$ Présence de plusieurs minima locaux (la dérivée s'annule pour plusieurs valeurs de $\theta$)
- Conséquence : la solution $\hat{\theta}$ est calculée par des méthodes itériatives (Gauss Newton, Gradient,...)
- Problème majeur : estimer un minimum local (*initialisation très importante*)

## Prédicteur pour un modèle OE

- L'estimée $\hat{\theta}$ est solution de :

$$\hat{\theta} = \arg\min_{\theta} \underbrace{\frac{1}{N}\sum_{t=1}^{N}\left(y(t_k - \varphi^T(t_k,\theta)\right)^2}_{V(\theta,Z^N)}$$

- Difficulté : le minimum du critère est atteint lorsque la dérivée s'annule :

$$\left.\frac{\partial V_N(\theta, Z^N)}{\partial\theta}\right|_{\theta=\hat{\theta}} = 0$$

  or la dérivée est une expression compliquée non linéaire en $\theta$
  $\Longrightarrow$ Présence de plusieurs minima locaux (la dérivée s'annule pour plusieurs valeurs de $\theta$)

- Conséquence : la solution $\hat{\theta}$ est calculée par des méthodes itériatives (Gauss Newton, Gradient,...)
- Problème majeur : estimer un minimum local (*initialisation très importante*)

# Plan

## Propriétés statistiques des estimateurs

- Etablir les propriétés asymptotiques pour un nombre de données $N$ fini est un problème difficile et complexe
- Solution simplificatrice retenue : écriture des propriétés asymptotiques de $\hat{\theta}_N$ lorsque $N$ tend vers l'infini, $\longrightarrow$ propriétés asymptotiques
- Du fait de la présence du bruit stochastique $v(t_k)$ sur les données d'E/S $Z^N$, le vecteur des paramètres estimés $\theta_N$ est une variable aléatoire.
- Conséquence : l'estimée $\theta_N$ est différente d'un jeu de données à l'autre (suivant la réalisation du bruit présent sur les données)
- L'objectif de l'estimateur : estimer les paramètres les plus "proches" possibles des paramètres vrais, $\forall$ la réalisation du bruit ,

## Propriétés statistiques des estimateurs

- Etablir les propriétés asymptotiques pour un nombre de données $N$ fini est un problème difficile et complexe
- Solution simplificatrice retenue : écriture des propriétés asymptotiques de $\hat{\theta}_N$ lorsque $N$ tend vers l'infini, $\longrightarrow$ propriétés asymptotiques
- Du fait de la présence du bruit stochastique $v(t_k)$ sur les données d'E/S $Z^N$, le vecteur des paramètres estimés $\theta_N$ est une variable aléatoire.
- Conséquence : l'estimée $\theta_N$ est différente d'un jeu de données à l'autre (suivant la réalisation du bruit présent sur les données)
- L'objectif de l'estimateur : estimer les paramètres les plus "proches" possibles des paramètres vrais, $\forall$ la réalisation du bruit ,

## Propriétés statistiques des estimateurs

- Etablir les propriétés asymptotiques pour un nombre de données $N$ fini est un problème difficile et complexe
- Solution simplificatrice retenue : écriture des propriétés asymptotiques de $\hat{\theta}_N$ lorsque $N$ tend vers l'infini, $\longrightarrow$ propriétés asymptotiques
- Du fait de la présence du bruit stochastique $v(t_k)$ sur les données d'E/S $Z^N$, le vecteur des paramètres estimés $\theta_N$ est une variable aléatoire.
- Conséquence : l'estimée $\theta_N$ est différente d'un jeu de données à l'autre (suivant la réalisation du bruit présent sur les données)
- L'objectif de l'estimateur : estimer les paramètres les plus "proches" possibles des paramètres vrais, $\forall$ la réalisation du bruit ,

## Propriétés statistiques des estimateurs

- Etablir les propriétés asymptotiques pour un nombre de données $N$ fini est un problème difficile et complexe
- Solution simplificatrice retenue : écriture des propriétés asymptotiques de $\hat{\theta}_N$ lorsque $N$ tend vers l'infini, $\longrightarrow$ propriétés asymptotiques
- Du fait de la présence du bruit stochastique $v(t_k)$ sur les données d'E/S $Z^N$, le vecteur des paramètres estimés $\theta_N$ est une variable aléatoire.
- Conséquence : l'estimée $\theta_N$ est différente d'un jeu de données à l'autre (suivant la réalisation du bruit présent sur les données)
- L'objectif de l'estimateur : estimer les paramètres les plus "proches" possibles des paramètres vrais, $\forall$ la réalisation du bruit ,

## Propriétés statistiques des estimateurs

- Etablir les propriétés asymptotiques pour un nombre de données $N$ fini est un problème difficile et complexe

- Solution simplificatrice retenue : écriture des propriétés asymptotiques de $\hat{\theta}_N$ lorsque $N$ tend vers l'infini, $\longrightarrow$ propriétés asymptotiques

- Du fait de la présence du bruit stochastique $v(t_k)$ sur les données d'E/S $Z^N$, le vecteur des paramètres estimés $\theta_N$ est une variable aléatoire.

- Conséquence : l'estimée $\theta_N$ est différente d'un jeu de données à l'autre (suivant la réalisation du bruit présent sur les données)

- L'objectif de l'estimateur : estimer les paramètres les plus "proches" possibles des paramètres vrais, $\forall$ la réalisation du bruit ,

## Propriétés statistiques des estimateurs

Objectif : définir un estimateur présentant les propriétés suivantes :

- Estimateur *non biaisé* (unbiased) :

$$E(\hat{\theta}_N) = \theta_0, \hat{\theta}_N \sim \mathcal{N}(\theta_0, P_\theta)$$

  où $P_\theta$ est la matrice de covariance

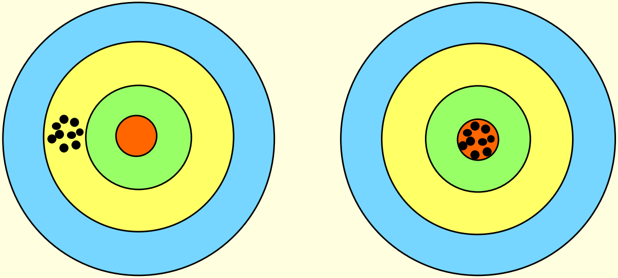
- Estimateur *convergent* (consistent) : $\hat{\theta}_N$ est consistent si
  - ◇ $Pr\left[\lim_{N\to\infty} \hat{\theta}_N = \theta_0\right] = 1$
  - ◇ $\hat{\theta}_N \longrightarrow \theta_0$ avec probabilité 1 pour $N \longrightarrow \infty$

- *Variance :* (minimale si possible)

$$cov(\hat{\theta}_N) = E(\hat{\theta}_N - E(\hat{\theta}_N))(\hat{\theta}_N - E(\hat{\theta}_N))^T$$

## Illustration du biais d'un estimateur

- Comparaison du biais de deux estimateurs



(a) Estimateur A (b) Estimateur B

- La moyenne des tirs pour l'estimateur A n'est pas au cœur de la cible (erreur systématique = biais) contrairement à l'estimateur B pour lequel la moyenne des tirs correspond au centre de la cible (estimateur B sans biais).

# Interprétation de $\hat{\theta}_N \sim \mathcal{N}(\theta_0, P_\theta)$

- Supposons que l'on puisse répéter $p$ expériences de $N$ données sur le système, donnant lieu à $p$ estimées différentes $\hat{\theta}_N^{(i)}$, pour $i \in [1, p]$, alors

$$\hat{\theta}_N \sim \mathcal{N}(\theta_0, P_\theta) \tag{11}$$
$$\Downarrow \tag{12}$$
$$P_\theta = E\big((\hat{\theta}_N - \theta_0)(\hat{\theta}_N - \theta_0)^T\big) \tag{13}$$

- La matrice de covariance des paramètres $P_\theta$ :
  - est une mesure des fluctuations des estimées autour de la valeur vraie
  - dépend du signal d'entrée $u(t_k)$ et du nombre $N$ de données

- Elle permet d'évaluer la variance de l'estimateur
  - un estimateur à variance minimale est dit précis

- Objectifs recherchés :
  - **estimateur sans biais et à variance minimale**

    $\Longrightarrow$ **estimateur optimal**

## Interprétation de $\hat{\theta}_N \sim \mathcal{N}(\theta_0, P_\theta)$

- Supposons que l'on puisse répéter $p$ expériences de $N$ données sur le système, donnant lieu à $p$ estimées différentes $\hat{\theta}_N^{(i)}$, pour $i \in [1, p]$, alors

$$\hat{\theta}_N \sim \mathcal{N}(\theta_0, P_\theta) \tag{11}$$

$$\Downarrow \tag{12}$$

$$P_\theta = E\big((\hat{\theta}_N - \theta_0)(\hat{\theta}_N - \theta_0)^T\big) \tag{13}$$

- La matrice de covariance des paramètres $P_\theta$ :
  - est une mesure des fluctuations des estimées autour de la valeur vraie
  - dépend du signal d'entrée $u(t_k)$ et du nombre $N$ de données

- Elle permet d'évaluer la variance de l'estimateur
  - un estimateur à variance minimale est dit précis

- Objectifs recherchés :
  - **estimateur sans biais et à variance minimale**

    $\Longrightarrow$ **estimateur optimal**

## Interprétation de $\hat{\theta}_N \sim \mathcal{N}(\theta_0, P_\theta)$

- Supposons que l'on puisse répéter $p$ expériences de $N$ données sur le système, donnant lieu à $p$ estimées différentes $\hat{\theta}_N^{(i)}$, pour $i \in [1, p]$, alors

$$\hat{\theta}_N \sim \mathcal{N}(\theta_0, P_\theta) \tag{11}$$

$$\Downarrow \tag{12}$$

$$P_\theta = E\left((\hat{\theta}_N - \theta_0)(\hat{\theta}_N - \theta_0)^T\right) \tag{13}$$

- La matrice de covariance des paramètres $P_\theta$ :
  - ◇ est une mesure des fluctuations des estimées autour de la valeur vraie
  - ◇ dépend du signal d'entrée $u(t_k)$ et du nombre $N$ de données

- Elle permet d'évaluer la variance de l'estimateur
  - ◇ un estimateur à variance minimale est dit précis

- Objectifs recherchés :
  - ◇ **estimateur sans biais et à variance minimale**

    $\Longrightarrow$ **estimateur optimal**

## Interprétation de $\hat{\theta}_N \sim \mathcal{N}(\theta_0, P_\theta)$

- Supposons que l'on puisse répéter $p$ expériences de $N$ données sur le système, donnant lieu à $p$ estimées différentes $\hat{\theta}_N^{(i)}$, pour $i \in [1, p]$, alors

$$\hat{\theta}_N \sim \mathcal{N}(\theta_0, P_\theta) \tag{11}$$
$$\Downarrow \tag{12}$$
$$P_\theta = E\left((\hat{\theta}_N - \theta_0)(\hat{\theta}_N - \theta_0)^T\right) \tag{13}$$

- La matrice de covariance des paramètres $P_\theta$ :
  - ◇ est une mesure des fluctuations des estimées autour de la valeur vraie
  - ◇ dépend du signal d'entrée $u(t_k)$ et du nombre $N$ de données

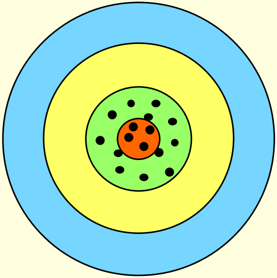
- Elle permet d'évaluer la variance de l'estimateur
  - ◇ un estimateur à variance minimale est dit précis

- Objectifs recherchés :
  - ◇ **estimateur sans biais et à variance minimale**

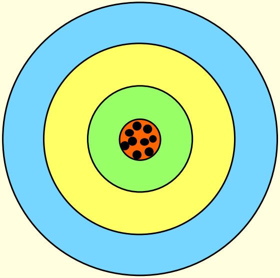
    $\Longrightarrow$ **estimateur optimal**

## Illustration de la variance d'un estimateur

- Comparaison de la variance de deux estimateurs



(c) Estimateur A

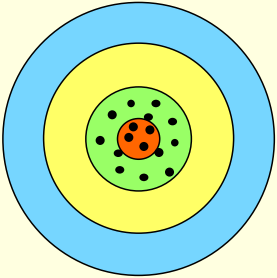


(d) Estimateur B

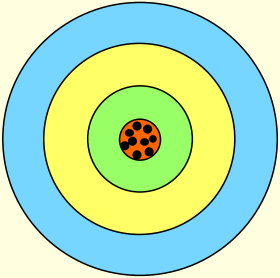
- Les deux estimateurs sont sans biais (moyenne des tirs au cœur de la cible)
- L'estimateur B est plus précis (variance plus faible) que l'estimateur A
$\Longrightarrow$ **estimateur B = estimateur optimal**

## Illustration de la variance d'un estimateur

- Comparaison de la variance de deux estimateurs



(e) Estimateur A



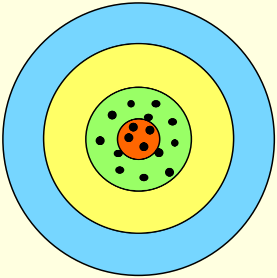
(f) Estimateur B

- Les deux estimateurs sont sans biais (moyenne des tirs au cœur de la cible)

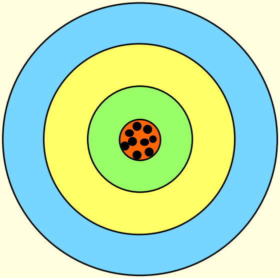
- L'estimateur B est plus précis (variance plus faible) que l'estimateur A
  $\Longrightarrow$ **estimateur B = estimateur optimal**

## Illustration de la variance d'un estimateur

- Comparaison de la variance de deux estimateurs



(g) Estimateur A



(h) Estimateur B

- Les deux estimateurs sont sans biais (moyenne des tirs au cœur de la cible)
- L'estimateur B est plus précis (variance plus faible) que l'estimateur A
  $\Longrightarrow$ **estimateur B = estimateur optimal**

## Interprétation de $Pr\left[\lim_{N\to\infty}\hat{\theta}_N = \theta_0\right] = 1$

- $Pr\left[\lim_{N\to\infty}\hat{\theta}_N = \theta_0\right] = 1$ s'écrit aussi :
  - $p\lim_{N\to\infty}\hat{\theta}_N = \theta_0$
  - ou encore $\lim\limits_{N\to\infty} P(\|\hat{\theta}_N - \theta_0\| < \varepsilon) = 0$

  L'estimateur est **convergent** en probabilité

- Si l'on pouvait acquérir une infinité de données ($N \to \infty$), le vecteur des paramètres estimés aurait la distribution suivante :

$$\hat{\theta}_{N\to\infty} \sim \mathcal{N}(\theta_0, P_\theta) \text{ avec } P_\theta = 0$$

- Remarque
  - si le biais et la variance de l'estimateur tendent asymptotiquement (pour $N \to \infty$) vers 0, alors l'estimateur est **convergent**

## Interprétation de $Pr\left[\lim_{N\to\infty}\hat{\theta}_N = \theta_0\right] = 1$

- $Pr\left[\lim_{N\to\infty}\hat{\theta}_N = \theta_0\right] = 1$ s'écrit aussi :
  - $p\lim_{N\to\infty}\hat{\theta}_N = \theta_0$
  - ou encore $\lim\limits_{N\to\infty} P(\|\hat{\theta}_N - \theta_0\| < \varepsilon) = 0$

  L'estimateur est **convergent** en probabilité

- Si l'on pouvait acquérir une infinité de données ($N \to \infty$), le vecteur des paramètres estimés aurait la distribution suivante :

$$\hat{\theta}_{N\to\infty} \sim \mathcal{N}(\theta_0, P_\theta) \text{ avec } P_\theta = 0$$

- Remarque
  - si le biais et la variance de l'estimateur tendent asymptotiquement (pour $N \to \infty$) vers 0, alors l'estimateur est **convergent**

## Interprétation de $Pr\left[\lim_{N\to\infty}\hat{\theta}_N = \theta_0\right] = 1$

- $Pr\left[\lim_{N\to\infty}\hat{\theta}_N = \theta_0\right] = 1$ s'écrit aussi :
  - $p\lim_{N\to\infty}\hat{\theta}_N = \theta_0$
  - ou encore $\lim\limits_{N\to\infty} P(\|\hat{\theta}_N - \theta_0\| < \varepsilon) = 0$

  L'estimateur est **convergent** en probabilité

- Si l'on pouvait acquérir une infinité de données ($N \to \infty$), le vecteur des paramètres estimés aurait la distribution suivante :

$$\hat{\theta}_{N\to\infty} \sim \mathcal{N}(\theta_0, P_\theta) \text{ avec } P_\theta = 0$$

- Remarque
  - si le biais et la variance de l'estimateur tendent asymptotiquement (pour $N \to \infty$) vers 0, alors l'estimateur est **convergent**

# Plan

# Propriétés statistiques des moindres carrés

- Pour établir les propriétés asymptotiques (biais, variance) de l'estimateur des MC, il faut formuler une hypothèse sur le système "vraï" ayant généré les données. On suppose que :

$$\boxed{\mathcal{S}: \; y(t_k) = \varphi^T(t_k)\theta_0 + v(t_k)}$$

- Le vecteur des paramètres estimés par MC (modèle) s'écrit :

$$\hat{\theta}_{mc} = \left[\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right]^{-1} \sum_{k=1}^{N} \varphi(t_k)y(t_k) \tag{14}$$

$$\hat{\theta}_{mc} = \left[\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right]^{-1} \sum_{k=1}^{N} \varphi(t_k)(\varphi(t_k)^T\theta_0 + v(t_k)) \tag{15}$$

$$\hat{\theta}_{mc} = \theta_0 + \left[\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right]^{-1} \sum_{k=1}^{N} \varphi(t_k)v(t_k)) \tag{16}$$

- Lorsque $N \to \infty$, $\hat{\theta}_{mc} - \theta_0 = E\left[\varphi(t_k)\varphi^T(t_k)\right]^{-1} E\left[\varphi(t_k)v(t_k)\right]$

## Propriétés statistiques des moindres carrés

- Pour établir les propriétés asymptotiques (biais, variance) de l'estimateur des MC, il faut formuler une hypothèse sur le système “vrai” ayant généré les données. On suppose que :

$$\boxed{\mathcal{S}: \; y(t_k) = \varphi^T(t_k)\theta_0 + v(t_k)}$$

- Le vecteur des paramètres estimés par MC (modèle) s'écrit :

$$\hat{\theta}_{mc} = \left[\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right]^{-1} \sum_{k=1}^{N} \varphi(t_k)y(t_k) \tag{14}$$

$$\hat{\theta}_{mc} = \left[\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right]^{-1} \sum_{k=1}^{N} \varphi(t_k)(\varphi(t_k)^T\theta_0 + v(t_k)) \tag{15}$$

$$\hat{\theta}_{mc} = \theta_0 + \left[\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right]^{-1} \sum_{k=1}^{N} \varphi(t_k)v(t_k)) \tag{16}$$

- Lorsque $N \to \infty$, $\hat{\theta}_{mc} - \theta_0 = E\left[\varphi(t_k)\varphi^T(t_k)\right]^{-1} E\left[\varphi(t_k)v(t_k)\right]$

## Propriétés statistiques des moindres carrés

- Pour établir les propriétés asymptotiques (biais, variance) de l'estimateur des MC, il faut formuler une hypothèse sur le système "vrai" ayant généré les données. On suppose que :

$$\boxed{\mathcal{S}:\ y(t_k) = \varphi^T(t_k)\theta_0 + v(t_k)}$$

- Le vecteur des paramètres estimés par MC (modèle) s'écrit :

$$\hat{\theta}_{mc} = \left[\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right]^{-1} \sum_{k=1}^{N} \varphi(t_k)y(t_k) \tag{14}$$

$$\hat{\theta}_{mc} = \left[\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right]^{-1} \sum_{k=1}^{N} \varphi(t_k)(\varphi(t_k)^T\theta_0 + v(t_k)) \tag{15}$$

$$\hat{\theta}_{mc} = \theta_0 + \left[\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right]^{-1} \sum_{k=1}^{N} \varphi(t_k)v(t_k)) \tag{16}$$

- Lorsque $N \to \infty$, $\hat{\theta}_{mc} - \theta_0 = E\left[\varphi(t_k)\varphi^T(t_k)\right]^{-1} E\left[\varphi(t_k)v(t_k)\right]$

# Propriétés statistiques des moindres carrés

- Pour établir les propriétés asymptotiques (biais, variance) de l'estimateur des MC, il faut formuler une hypothèse sur le système "vrai" ayant généré les données. On suppose que :

$$\boxed{\mathcal{S}: \; y(t_k) = \varphi^T(t_k)\theta_0 + v(t_k)}$$

- Le vecteur des paramètres estimés par MC (modèle) s'écrit :

$$\hat{\theta}_{mc} = \left[\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right]^{-1} \sum_{k=1}^{N} \varphi(t_k)y(t_k) \quad (14)$$

$$\hat{\theta}_{mc} = \left[\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right]^{-1} \sum_{k=1}^{N} \varphi(t_k)(\varphi(t_k)^T\theta_0 + v(t_k)) \quad (15)$$

$$\hat{\theta}_{mc} = \theta_0 + \left[\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right]^{-1} \sum_{k=1}^{N} \varphi(t_k)v(t_k)) \quad (16)$$

- Lorsque $N \to \infty$, $\hat{\theta}_{mc} - \theta_0 = E\left[\varphi(t_k)\varphi^T(t_k)\right]^{-1} E\left[\varphi(t_k)v(t_k)\right]$

## Propriétés statistiques des moindres carrés

- Pour établir les propriétés asymptotiques (biais, variance) de l'estimateur des MC, il faut formuler une hypothèse sur le système "vrai" ayant généré les données. On suppose que :

$$\boxed{\mathcal{S}: \; y(t_k) = \varphi^T(t_k)\theta_0 + v(t_k)}$$

- Le vecteur des paramètres estimés par MC (modèle) s'écrit :

$$\hat{\theta}_{mc} = \left[\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right]^{-1} \sum_{k=1}^{N} \varphi(t_k)y(t_k) \tag{14}$$

$$\hat{\theta}_{mc} = \left[\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right]^{-1} \sum_{k=1}^{N} \varphi(t_k)(\varphi(t_k)^T\theta_0 + v(t_k)) \tag{15}$$

$$\hat{\theta}_{mc} = \theta_0 + \left[\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right]^{-1} \sum_{k=1}^{N} \varphi(t_k)v(t_k)) \tag{16}$$

- Lorsque $N \to \infty$, $\boxed{\hat{\theta}_{mc} - \theta_0 = E\left[\varphi(t_k)\varphi^T(t_k)\right]^{-1} E\left[\varphi(t_k)v(t_k)\right]}$

## Biais de l'estimateur des moindres carrés

- Lorsque $N \to \infty$, $\boxed{\hat{\theta}_{mc} - \theta_0 = E\left[\varphi(t_k)\varphi^T(t_k)\right]^{-1} E\left[\varphi(t_k)v(t_k)\right]}$

- L'estimateur des moindres carrés sera convergent ssi
  1. $E\left[\varphi(t_k)\varphi^T(t_k)\right]$ est non singulière
  2. $E\left[\varphi(t_k)v(t_k)\right] = 0$

  ◇ La 1ère condition est souvent vérifiée (sous condition de signal d'excitation d'ordre suffisant et période d'échantillonnage adéquate)

  ◇ La 2nde condition n'est quasi jamais vérifiée !

SAUF si $v(t_k) = e(t_k)$ est un bruit blanc,
donc si le modèle recherché est de structure
ARX !

- En général, l'estimateur des moindres carrés n'est pas convergent (toujours biaisé), sauf si le bruit $v(t_k)$ est un bruit blanc (structure ARX).

## Biais de l'estimateur des moindres carrés

- Lorsque $N \to \infty$, $\boxed{\hat{\theta}_{mc} - \theta_0 = E\left[\varphi(t_k)\varphi^T(t_k)\right]^{-1} E\left[\varphi(t_k)v(t_k)\right]}$

- L'estimateur des moindres carrés sera convergent ssi
  1. $E\left[\varphi(t_k)\varphi^T(t_k)\right]$ est non singulière
  2. $E\left[\varphi(t_k)v(t_k)\right] = 0$

  ◇ La 1ère condition est souvent vérifiée (sous condition de signal d'excitation d'ordre suffisant et période d'échantillonnage adéquate)

  ◇ La 2nde condition n'est quasi jamais vérifiée !

SAUF si $v(t_k) = e(t_k)$ est un bruit blanc,
donc si le modèle recherché est de structure
ARX !

- En général, l'estimateur des moindres carrés n'est pas convergent (toujours biaisé), sauf si le bruit $v(t_k)$ est un bruit blanc (structure ARX).

## Biais de l'estimateur des moindres carrés

- Lorsque $N \to \infty$, $\boxed{\hat{\theta}_{mc} - \theta_0 = E\left[\varphi(t_k)\varphi^T(t_k)\right]^{-1} E\left[\varphi(t_k)v(t_k)\right]}$

- L'estimateur des moindres carrés sera convergent ssi
  1. $E\left[\varphi(t_k)\varphi^T(t_k)\right]$ est non singulière
  2. $E\left[\varphi(t_k)v(t_k)\right] = 0$

  ◇ La 1ère condition est souvent vérifiée (sous condition de signal d'excitation d'ordre suffisant et période d'échantillonnage adéquate)

  ◇ La 2nde condition n'est quasi jamais vérifiée !

SAUF si $v(t_k) = e(t_k)$ est un bruit blanc,
donc si le modèle recherché est de structure
ARX !

- En général, l'estimateur des moindres carrés n'est pas convergent (toujours biaisé), sauf si le bruit $v(t_k)$ est un bruit blanc (structure ARX).

# Biais de l'estimateur des moindres carrés

- Lorsque $N \to \infty$, $\boxed{\hat{\theta}_{mc} - \theta_0 = E\left[\varphi(t_k)\varphi^T(t_k)\right]^{-1} E\left[\varphi(t_k)v(t_k)\right]}$

- L'estimateur des moindres carrés sera convergent ssi
  1. $E\left[\varphi(t_k)\varphi^T(t_k)\right]$ est non singulière
  2. $E\left[\varphi(t_k)v(t_k)\right] = 0$

    ◇ La 1ère condition est souvent vérifiée (sous condition de signal d'excitation d'ordre suffisant et période d'échantillonnage adéquate)

    ◇ La 2nde condition n'est quasi jamais vérifée !

SAUF si $v(t_k) = e(t_k)$ est un bruit blanc,
donc si le modèle recherché est de structure
ARX !

- En général, l'estimateur des moindres carrés n'est pas convergent (toujours biaisé), sauf si le bruit $v(t_k)$ est un bruit blanc (structure ARX).

# Biais de l'estimateur des moindres carrés

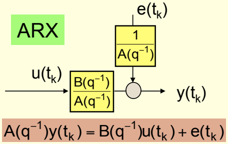
- Lorsque $N \to \infty$, $\hat{\theta}_{mc} - \theta_0 = E\left[\varphi(t_k)\varphi^T(t_k)\right]^{-1} E\left[\varphi(t_k)v(t_k)\right]$

- L'estimateur des moindres carrés sera convergent ssi
  1. $E\left[\varphi(t_k)\varphi^T(t_k)\right]$ est non singulière
  2. $E\left[\varphi(t_k)v(t_k)\right] = 0$

  ◇ La 1ère condition est souvent vérifiée (sous condition de signal d'excitation d'ordre suffisant et période d'échantillonnage adéquate)

  ◇ La 2nde condition n'est quasi jamais vérifiée !

SAUF si $v(t_k) = e(t_k)$ est un bruit blanc, donc si le modèle recherché est de structure ARX !




- En général, l'estimateur des moindres carrés n'est pas convergent (toujours biaisé), sauf si le bruit $v(t_k)$ est un bruit blanc (structure ARX).

# Variance de l'estimateur des moindres carrés

- Matrice de covariance des paramètres :
  - ◇ mesure des fluctuations des estimés autour de la valeur vraie.

$$cov(\hat{\theta}_N) = P_{\hat{\theta}_N} = P\left[\left(\hat{\theta}_N - \theta_0\right)\left(\hat{\theta}_N - \theta_0\right)^T\right]$$

  - ◇ Calcul de la variance à partir de l'écriture matricielle. Soit

$$\hat{\theta}_N = \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T Y_N$$

# Variance de l'estimateur des moindres carrés

- Matrice de covariance des paramètres :
  - ◇ mesure des fluctuations des estimés autour de la valeur vraie.

    $$cov(\hat{\theta}_N) = P_{\hat{\theta}_N} = P\left[\left(\hat{\theta}_N - \theta_0\right)\left(\hat{\theta}_N - \theta_0\right)^T\right]$$

  - ◇ Calcul de la variance à partir de l'écriture matricielle. Soit

$$\hat{\theta}_N = \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T Y_N$$

# Variance de l'estimateur des moindres carrés

- Matrice de covariance des paramètres :
  - ◇ mesure des fluctuations des estimés autour de la valeur vraie.

    $$cov(\hat{\theta}_N) = P_{\hat{\theta}_N} = P\left[\left(\hat{\theta}_N - \theta_0\right)\left(\hat{\theta}_N - \theta_0\right)^T\right]$$

  - ◇ Calcul de la variance à partir de l'écriture matricielle. Soit

    $$\hat{\theta}_N = \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T Y_N$$

    On en déduit :

    $$\begin{aligned}\hat{\theta}_N - \theta_0 &= \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T Y_N - \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T \Phi_N \theta_0 \\ &= \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T \left(Y_N - \Phi_N \theta_0\right) \\ &= \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T \varepsilon_0\end{aligned}$$

# Variance de l'estimateur des moindres carrés

- Matrice de covariance des paramètres :
  - ◇ mesure des fluctuations des estimés autour de la valeur vraie.

$$cov(\hat{\theta}_N) = P_{\hat{\theta}_N} = P\left[\left(\hat{\theta}_N - \theta_0\right)\left(\hat{\theta}_N - \theta_0\right)^T\right]$$

  - ◇ Calcul de la variance à partir de l'écriture matricielle. Soit

$$\hat{\theta}_N = \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T Y_N$$

On en déduit :

$$\begin{aligned}
\hat{\theta}_N - \theta_0 &= \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T Y_N - \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T \Phi_N \theta_0 \\
&= \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T \left(Y_N - \Phi_N \theta_0\right) \\
&= \left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T \varepsilon_0
\end{aligned}$$

On reporte dans l'équation de la variance :

$$cov(\hat{\theta}_N) = E\left[\left(\left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T \varepsilon_0\right)\left(\left(\Phi_N^T \Phi_N\right)^{-1} \Phi_N^T \varepsilon_0\right)^T\right]$$

# Variance de l'estimateur des moindres carrés

- Matrice de covariance des paramètres (suite) :
  - ◇ Suite du calcul

$$
\begin{aligned}
cov(\hat{\theta}_N) &= E\left[\left(\left(\Phi_N^T\Phi_N\right)^{-1}\Phi_N^T\varepsilon_0\right)\left(\left(\Phi_N^T\Phi_N\right)^{-1}\Phi_N^T\varepsilon_0\right)^T\right] \\
&= E\left[\left(\Phi_N^T\Phi_N\right)^{-1}\Phi_N^T\varepsilon_0\varepsilon_0^T\Phi_N\left(\Phi_N^T\Phi_N\right)^{-1}\right] \\
&= \sigma_0^2 E\left[\left(\Phi_N^T\Phi_N\right)^{-1}\Phi_N^T\Phi_N\left(\Phi_N^T\Phi_N\right)^{-1}\right] \text{ car } E\left[\varepsilon_0\varepsilon_0^T\right] = \sigma_0^2 I \\
&= \sigma_0^2 E\left[\left(\Phi_N^T\Phi_N\right)^{-1}\right]
\end{aligned}
$$

  - Expression développée

$$
cov(\hat{\theta}_N) = \sigma_0^2 E\left[\left(\frac{1}{N}\sum_{k=1}^{N}\varphi(t_k)\varphi^T(t_k)\right)^{-1}\right] \tag{17}
$$

  - En conclusion

$$
\boxed{cov(\hat{\theta}_N) = \sigma_0^2 E\left[\left(\Phi_N^T\Phi_N\right)^{-1}\right] = \sigma_0^2 E\left[\left(\frac{1}{N}\sum_{k=1}^{N}\varphi(t_k)\varphi^T(t_k)\right)^{-1}\right]}
$$

## Variance de l'estimateur des moindres carrés

$$cov(\hat{\theta}_N) = \sigma_0^2 E\left[\left(\Phi_N^T \Phi_N\right)^{-1}\right] = \sigma_0^2 E\left[\left(\frac{1}{N}\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right)^{-1}\right]$$

- Lorsque $N \to \infty$, alors : $cov(\hat{\theta}_N) = \dfrac{\sigma_0^2}{N} R_{\varphi\varphi}^{-1}(0)$

- Interprétation : la variance des paramètres estimés par MC
  - est proportionnelle à la variance du bruit $e_0$
  - diminue avec le nombre de données
  - le signal d'excitation $u$ doit être "riche" afin que la matrice d'information $R_{\varphi\varphi}(0)$ soit importante

- En pratique, $\sigma_0^2$ est inconnu. Une estimée (non biaisée) peut être calculée en sachant que :
  - $\sigma_\varepsilon^2 = E\left[\varepsilon^2(t_k) - E\left[\varepsilon(t_k)\right]^2\right] = E\left[\varepsilon^2(t_k)\right]$ car $E\left[\varepsilon(t_k)\right] = 0$
  - D'où : $\sigma_\varepsilon^2 = \dfrac{1}{N - dim(\hat{\theta})} \sum_{k=1}^{N} \varepsilon^2(t_k) \quad et \quad cov(\hat{\theta}_N) = \dfrac{\hat{\sigma}_\varepsilon^2}{N} R_{\varphi\varphi}^{-1}(0)$

## Variance de l'estimateur des moindres carrés

$$cov(\hat{\theta}_N) = \sigma_0^2 E\left[\left(\Phi_N^T \Phi_N\right)^{-1}\right] = \sigma_0^2 E\left[\left(\frac{1}{N}\sum_{k=1}^N \varphi(t_k)\varphi^T(t_k)\right)^{-1}\right]$$

- Lorsque $N \to \infty$, alors : $cov(\hat{\theta}_N) = \dfrac{\sigma_0^2}{N} R_{\varphi\varphi}^{-1}(0)$
- Interprétation : la variance des paramètres estimés par MC
  - ◇ est proportionnelle à la variance du bruit $e_0$
  - ◇ diminue avec le nombre de données
  - ◇ le signal d'excitation $u$ doit être "riche" afin que la matrice d'information $R_{\varphi\varphi}(0)$ soit importante
- En pratique, $\sigma_0^2$ est inconnu. Une estimée (non biaisée) peut être calculée en sachant que :
  - ◇ $\sigma_\varepsilon^2 = E\left[\varepsilon^2(t_k) - E\left[\varepsilon(t_k)\right]^2\right] = E\left[\varepsilon^2(t_k)\right]$ car $E\left[\varepsilon(t_k)\right] = 0$
  - ◇ D'où : $\hat{\sigma}_\varepsilon^2 = \dfrac{1}{N - dim(\hat{\theta})} \sum_{k=1}^{N} \varepsilon^2(t_k)$ et $cov(\hat{\theta}_N) = \dfrac{\hat{\sigma}_\varepsilon^2}{N} R_{\varphi\varphi}^{-1}(0)$

# Variance de l'estimateur des moindres carrés

$$cov(\hat{\theta}_N) = \sigma_0^2 E\left[\left(\Phi_N^T \Phi_N\right)^{-1}\right] = \sigma_0^2 E\left[\left(\frac{1}{N}\sum_{k=1}^N \varphi(t_k)\varphi^T(t_k)\right)^{-1}\right]$$

- Lorsque $N \to \infty$, alors : $cov(\hat{\theta}_N) = \dfrac{\sigma_0^2}{N} R_{\varphi\varphi}^{-1}(0)$
- Interprétation : la variance des paramètres estimés par MC
  - ◇ est proportionnelle à la variance du bruit $e_0$
  - ◇ diminue avec le nombre de données
  - ◇ le signal d'excitation $u$ doit être "riche" afin que la matrice d'information $R_{\varphi\varphi}(0)$ soit importante
- En pratique, $\sigma_0^2$ est inconnu. Une estimée (non biaisée) peut être calculée en sachant que :
  - ◇ $\sigma_\varepsilon^2 = E\left[\varepsilon^2(t_k) - E\left[\varepsilon(t_k)\right]^2\right] = E\left[\varepsilon^2(t_k)\right]$ car $E\left[\varepsilon(t_k)\right] = 0$
  - ◇ D'où : $\sigma_\varepsilon^2 = \dfrac{1}{N - dim(\hat{\theta})}\sum_{k=1}^N \varepsilon^2(t_k) \quad et \quad cov(\hat{\theta}_N) = \dfrac{\hat{\sigma}_\varepsilon^2}{N} R_{\varphi\varphi}^{-1}(0)$

## Variance de l'estimateur des moindres carrés

$$cov(\hat{\theta}_N) = \sigma_0^2 E\left[\left(\Phi_N^T \Phi_N\right)^{-1}\right] = \sigma_0^2 E\left[\left(\frac{1}{N}\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right)^{-1}\right]$$

- Lorsque $N \to \infty$, alors : $cov(\hat{\theta}_N) = \dfrac{\sigma_0^2}{N} R_{\varphi\varphi}^{-1}(0)$
- Interprétation : la variance des paramètres estimés par MC
  - ◇ est proportionnelle à la variance du bruit $e_0$
  - ◇ diminue avec le nombre de données
  - ◇ le signal d'excitation $u$ doit être "riche" afin que la matrice d'information $R_{\varphi\varphi}(0)$ soit importante
- En pratique, $\sigma_0^2$ est inconnu. Une estimée (non biaisée) peut être calculée en sachant que :
  - ◇ $\sigma_\varepsilon^2 = E\left[\varepsilon^2(t_k) - E\left[\varepsilon(t_k)\right]^2\right] = E\left[\varepsilon^2(t_k)\right]$ car $E\left[\varepsilon(t_k)\right] = 0$
  - ◇ D'où : $\hat{\sigma}_\varepsilon^2 = \dfrac{1}{N - dim(\hat{\theta})} \displaystyle\sum_{k=1}^{N} \varepsilon^2(t_k) \quad et \quad cov(\hat{\theta}_N) = \dfrac{\hat{\sigma}_\varepsilon^2}{N} R_{\varphi\varphi}^{-1}(0)$

# Variance de l'estimateur des moindres carrés

$$cov(\hat{\theta}_N) = \sigma_0^2 E\left[\left(\Phi_N^T \Phi_N\right)^{-1}\right] = \sigma_0^2 E\left[\left(\frac{1}{N}\sum_{k=1}^N \varphi(t_k)\varphi^T(t_k)\right)^{-1}\right]$$

- Lorsque $N \to \infty$, alors : $cov(\hat{\theta}_N) = \dfrac{\sigma_0^2}{N} R_{\varphi\varphi}^{-1}(0)$
- Interprétation : la variance des paramètres estimés par MC
  - est proportionnelle à la variance du bruit $e_0$
  - diminue avec le nombre de données
  - le signal d'excitation $u$ doit être "riche" afin que la matrice d'information $R_{\varphi\varphi}(0)$ soit importante
- En pratique, $\sigma_0^2$ est inconnu. Une estimée (non biaisée) peut être calculée en sachant que :
  - $\sigma_\varepsilon^2 = E\left[\varepsilon^2(t_k) - E\left[\varepsilon(t_k)\right]^2\right] = E\left[\varepsilon^2(t_k)\right]$ car $E\left[\varepsilon(t_k)\right] = 0$
  - D'où : $\hat{\sigma}_\varepsilon^2 = \dfrac{1}{N - dim(\hat{\theta})} \displaystyle\sum_{k=1}^{N} \varepsilon^2(t_k)$ $\quad et \quad$ $cov(\hat{\theta}_N) = \dfrac{\hat{\sigma}_\varepsilon^2}{N} R_{\varphi\varphi}^{-1}(0)$

# Variance de l'estimateur des moindres carrés

$$cov(\hat{\theta}_N) = \sigma_0^2 E\left[\left(\Phi_N^T \Phi_N\right)^{-1}\right] = \sigma_0^2 E\left[\left(\frac{1}{N}\sum_{k=1}^{N} \varphi(t_k)\varphi^T(t_k)\right)^{-1}\right]$$

- Lorsque $N \to \infty$, alors : $cov(\hat{\theta}_N) = \dfrac{\sigma_0^2}{N} R_{\varphi\varphi}^{-1}(0)$
- Interprétation : la variance des paramètres estimés par MC
  - ◇ est proportionnelle à la variance du bruit $e_0$
  - ◇ diminue avec le nombre de données
  - ◇ le signal d'excitation $u$ doit être "riche" afin que la matrice d'information $R_{\varphi\varphi}(0)$ soit importante
- En pratique, $\sigma_0^2$ est inconnu. Une estimée (non biaisée) peut être calculée en sachant que :
  - ◇ $\sigma_\varepsilon^2 = E\left[\varepsilon^2(t_k) - E\left[\varepsilon(t_k)\right]^2\right] = E\left[\varepsilon^2(t_k)\right]$ car $E\left[\varepsilon(t_k)\right] = 0$
  - ◇ D'où : $\sigma_\varepsilon^2 = \dfrac{1}{N - dim(\hat{\theta})} \displaystyle\sum_{k=1}^{N} \varepsilon^2(t_k)$ et $cov(\hat{\theta}_N) = \dfrac{\hat{\sigma}_\varepsilon^2}{N} R_{\varphi\varphi}^{-1}(0)$

# Variance de l'estimateur des moindres carrés

$$cov(\hat{\theta}_N) = \sigma_0^2 E\left[\left(\Phi_N^T \Phi_N\right)^{-1}\right] = \sigma_0^2 E\left[\left(\frac{1}{N}\sum_{k=1}^N \varphi(t_k)\varphi^T(t_k)\right)^{-1}\right]$$

- Lorsque $N \to \infty$, alors : $cov(\hat{\theta}_N) = \dfrac{\sigma_0^2}{N} R_{\varphi\varphi}^{-1}(0)$
- Interprétation : la variance des paramètres estimés par MC
  - est proportionnelle à la variance du bruit $e_0$
  - diminue avec le nombre de données
  - le signal d'excitation $u$ doit être "riche" afin que la matrice d'information $R_{\varphi\varphi}(0)$ soit importante
- En pratique, $\sigma_0^2$ est inconnu. Une estimée (non biaisée) peut être calculée en sachant que :
  - $\sigma_\varepsilon^2 = E\left[\varepsilon^2(t_k) - E\left[\varepsilon(t_k)\right]^2\right] = E\left[\varepsilon^2(t_k)\right]$ car $E\left[\varepsilon(t_k)\right] = 0$
  - D'où : $\sigma_\varepsilon^2 = \dfrac{1}{N - dim(\hat{\theta})} \displaystyle\sum_{k=1}^N \varepsilon^2(t_k)$ *et* $cov(\hat{\theta}_N) = \dfrac{\hat{\sigma}_\varepsilon^2}{N} R_{\varphi\varphi}^{-1}(0)$

# Analyse des performances de l'estimateur des MC par simulation de Monte Carlo

```
B=[0 0.2];              % B(q)=0.2q-1
A=[1 -0.8];             % A(q)=1-0.8q-1
N=500;                  % Nombre d'échantillons
S = idpoly(1,B,1,1,A);            %y=B/Au+e  - Modèle OE et non un modèle ARX – MC biaisé
u=sign(randn(N,1)); % signal d'entrée
Matrice_param_arx=[];
 for i=1:200            % 200 réalisations de bruit
        randn('state',sum(100*clock));
        e=0.2*randn(N,1);
        y=sim(S,[u e]); % simulation de la sortie bruitée
        dataS= iddata(y,u); % Objet de données ayant y comme sortie et u comme entrée
        Marx=arx(dataS,[1 1 1]); % Estimation par MC des paramètres du modèle ARX
        Matrice_param_arx=[Matrice_param_arx;  [Marx.B(2:end)  Marx.A(2:end)]];
end
% Valeur moyenne des paramètres estimés par MC
mean(Matrice_param_arx)
0.1994     -0.5888   % Paramètre â1 estimé : biaisé – Estimateur des MC : biaisé
% Ecart-type des paramètres estimés par MC
std(Matrice_param_arx)
0.0106      0.0222
```

# Plan

# Estimation par l'erreur de prédiction – Bilan

Deux cas de figures doivent être distingués :

1. Le prédicteur est une fonction **linéaire en les paramètres**
   - Cas d'un modèle ARX (et FIR)
   - La prédiction de la sortie s'écrit sous forme de régression linéaire

$$\hat{y}(t_k, \theta) = \varphi^T(t_k)\theta$$

   - Une solution analytique existe !
   - L'estimateur des moindres carrés est obtenu analytiquement, à partir des données, sans problème d'initialisation
   - Propriétés statistiques de l'estimateur : estimées biaisées et asymptotiquement biaisées en général (hypothèse sur le bruit irréaliste)
   - Cependant, même si les estimées par moindres carrés ne sont pas les meilleures, elle peuvent servir à initialiser des méthodes itératives

## Estimation par l'erreur de prédiction – Bilan

Deux cas de figures doivent être distingués :

1. Le prédicteur est une fonction **linéaire en les paramètres**
   - Cas d'un modèle ARX (et FIR)
   - La prédiction de la sortie s'écrit sous forme de régression linéaire

$$\hat{y}(t_k, \theta) = \varphi^T(t_k)\theta$$

   - Une solution analytique existe !
   - L'estimateur des moindres carrés est obtenu analytiquement, à partir des données, sans problème d'initialisation
   - Propriétés statistiques de l'estimateur : estimées biaisées et asymptotiquement biaisées en général (hypothèse sur le bruit irréaliste)
   - Cependant, même si les estimées par moindres carrés ne sont pas les meilleures, elle peuvent servir à initialiser des méthodes itératives

## Estimation par l'erreur de prédiction – Bilan

Deux cas de figures doivent être distingués :

1. Le prédicteur est une fonction **linéaire en les paramètres**
   - Cas d'un modèle ARX (et FIR)
   - La prédiction de la sortie s'écrit sous forme de régression linéaire

     $$\hat{y}(t_k, \theta) = \varphi^T(t_k)\theta$$

   - Une solution analytique existe !
   - L'estimateur des moindres carrés est obtenu analytiquement, à partir des données, sans problème d'initialisation
   - Propriétés statistiques de l'estimateur : estimées biaisées et asymptotiquement biaisées en général (hypothèse sur le bruit irréaliste)
   - Cependant, même si les estimées par moindres carrés ne sont pas les meilleures, elle peuvent servir à initialiser des méthodes itératives

## Estimation par l'erreur de prédiction – Bilan

Deux cas de figures doivent être distingués :

1. Le prédicteur est une fonction **linéaire en les paramètres**
   - Cas d'un modèle ARX (et FIR)
   - La prédiction de la sortie s'écrit sous forme de régression linéaire

$$\hat{y}(t_k, \theta) = \varphi^T(t_k)\theta$$

   - Une solution analytique existe !
   - L'estimateur des moindres carrés est obtenu analytiquement, à partir des données, sans problème d'initialisation
   - Propriétés statistiques de l'estimateur : estimées biaisées et asymptotiquement biaisées en général (hypothèse sur le bruit irréaliste)
   - Cependant, même si les estimées par moindres carrés ne sont pas les meilleures, elle peuvent servir à initialiser des méthodes itératives

## Estimation par l'erreur de prédiction – Bilan

Deux cas de figures doivent être distingués :

1. Le prédicteur est une fonction **linéaire en les paramètres**
   - Cas d'un modèle ARX (et FIR)
   - La prédiction de la sortie s'écrit sous forme de régression linéaire

     $$\hat{y}(t_k, \theta) = \varphi^T(t_k)\theta$$

   - Une solution analytique existe !
   - L'estimateur des moindres carrés est obtenu analytiquement, à partir des données, sans problème d'initialisation
   - Propriétés statistiques de l'estimateur : estimées biaisées et asymptotiquement biaisées en général (hypothèse sur le bruit irréaliste)
   - Cependant, même si les estimées par moindres carrés ne sont pas les meilleures, elle peuvent servir à initialiser des méthodes itératives

# Estimation par l'erreur de prédiction – Bilan

Deux cas de figures doivent être distingués :

1. Le prédicteur est une fonction **linéaire en les paramètres**
   - Cas d'un modèle ARX (et FIR)
   - La prédiction de la sortie s'écrit sous forme de régression linéaire

     $$\hat{y}(t_k, \theta) = \varphi^T(t_k)\theta$$

   - Une solution analytique existe !
   - L'estimateur des moindres carrés est obtenu analytiquement, à partir des données, sans problème d'initialisation
   - Propriétés statistiques de l'estimateur : estimées biaisées et asymptotiquement biaisées en général (hypothèse sur le bruit irréaliste)
   - Cependant, même si les estimées par moindres carrés ne sont pas les meilleures, elle peuvent servir à initialiser des méthodes itératives

## Estimation par l'erreur de prédiction – Bilan

Deux cas de figures doivent être distingués (suite) :

5. Le prédicteur **n'est pas une fonction linéaire en les paramètres**
   - Cas des modèles ARMAX, OE, BJ
   - La prédiction de la sortie ne s'écrit pas sous forme de régression linéaire
   - Une solution analytique n'existe pas !
   - Minimisation du critère effectuée à l'aide d'algorithmes numériques itératifs : gradient, Gauss-Newton, Levenberg-Marquardt,. . .
   - Prop. stat. des estimateurs : asymptotiquement non biaisés en général
   - Problèmes :
     - si le critère possède plusieurs minima, la méthode peut converger vers un minimum local et non vers le minimum global
     - Algorithmes gourmands en calculs
     - La convergence peut être très sensible à l'initialisation de l'algorithme

## Estimation par l'erreur de prédiction – Bilan

Deux cas de figures doivent être distingués (suite) :

5. Le prédicteur **n'est pas une fonction linéaire en les paramètres**
   - ◇ Cas des modèles ARMAX, OE, BJ
   - ◇ La prédiction de la sortie ne s'écrit pas sous forme de régression linéaire
   - ◇ Une solution analytique n'existe pas !
   - ◇ Minimisation du critère effectuée à l'aide d'algorithmes numériques itératifs : gradient, Gauss-Newton, Levenberg-Marquardt,. . .
   - ◇ Prop. stat. des estimateurs : asymptotiquement non biaisés en général
   - ◇ Problèmes :
     - ◇ si le critère possède plusieurs minima, la méthode peut converger vers un minimum local et non vers le minimum global
     - ◇ Algorithmes gourmands en calculs
     - ◇ La convergence peut être très sensible à l'initialisation de l'algorithme

## Estimation par l'erreur de prédiction – Bilan

Deux cas de figures doivent être distingués (suite) :

5. Le prédicteur **n'est pas une fonction linéaire en les paramètres**
   - Cas des modèles ARMAX, OE, BJ
   - La prédiction de la sortie ne s'écrit pas sous forme de régression linéaire
   - Une solution analytique n'existe pas !
   - Minimisation du critère effectuée à l'aide d'algorithmes numériques itératifs : gradient, Gauss-Newton, Levenberg-Marquardt,...
   - Prop. stat. des estimateurs : asymptotiquement non biaisés en général
   - Problèmes :
     - si le critère possède plusieurs minima, la méthode peut converger vers un minimum local et non vers le minimum global
     - Algorithmes gourmands en calculs
     - La convergence peut être très sensible à l'initialisation de l'algorithme

## Estimation par l'erreur de prédiction – Bilan

Deux cas de figures doivent être distingués (suite) :

5. Le prédicteur **n'est pas une fonction linéaire en les paramètres**
   - Cas des modèles ARMAX, OE, BJ
   - La prédiction de la sortie ne s'écrit pas sous forme de régression linéaire
   - Une solution analytique n'existe pas !
   - Minimisation du critère effectuée à l'aide d'algorithmes numériques itératifs : gradient, Gauss-Newton, Levenberg-Marquardt,. . .
   - Prop. stat. des estimateurs : asymptotiquement non biaisés en général
   - Problèmes :
     - si le critère possède plusieurs minima, la méthode peut converger vers un minimum local et non vers le minimum global
     - Algorithmes gourmands en calculs
     - La convergence peut être très sensible à l'initialisation de l'algorithme

## Estimation par l'erreur de prédiction – Bilan

Deux cas de figures doivent être distingués (suite) :

5. Le prédicteur **n'est pas une fonction linéaire en les paramètres**
   - Cas des modèles ARMAX, OE, BJ
   - La prédiction de la sortie ne s'écrit pas sous forme de régression linéaire
   - Une solution analytique n'existe pas !
   - Minimisation du critère effectuée à l'aide d'algorithmes numériques itératifs : gradient, Gauss-Newton, Levenberg-Marquardt,...
   - Prop. stat. des estimateurs : asymptotiquement non biaisés en général
   - Problèmes :
     - si le critère possède plusieurs minima, la méthode peut converger vers un minimum local et non vers le minimum global
     - Algorithmes gourmands en calculs
     - La convergence peut être très sensible à l'initialisation de l'algorithme

## Estimation par l'erreur de prédiction – Bilan

Deux cas de figures doivent être distingués (suite) :

5. Le prédicteur **n'est pas une fonction linéaire en les paramètres**
   - Cas des modèles ARMAX, OE, BJ
   - La prédiction de la sortie ne s'écrit pas sous forme de régression linéaire
   - Une solution analytique n'existe pas !
   - Minimisation du critère effectuée à l'aide d'algorithmes numériques itératifs : gradient, Gauss-Newton, Levenberg-Marquardt,. . .
   - Prop. stat. des estimateurs : asymptotiquement non biaisés en général
   - Problèmes :
     - si le critère possède plusieurs minima, la méthode peut converger vers un minimum local et non vers le minimum global
     - Algorithmes gourmands en calculs
     - La convergence peut être très sensible à l'initialisation de l'algorithme