

Chapitre 6

IDENTIFICATION PARAMÉTRIQUE

Nous commençons ce chapitre par l'algorithme de base servant à estimer les paramètres de la transmittance d'un système dynamique linéaire invariant non perturbé. Différentes interprétations de cet algorithme ainsi que sa mise en œuvre sont aussi envisagées. Les algorithmes relatifs aux systèmes perturbés sont abordés plus loin et le chapitre se termine par la méthode du sous-espace permettant d'identifier les équations d'état d'un système.

En guise d'introduction, considérons un organe dont la fonction de transfert est

$$G(z) = \frac{Zy}{Zu} = \frac{bz}{z+a} \quad (6-1)$$

où a est un paramètre connu mais le paramètre b est inconnu et doit être identifié. Une méthode classique consiste à séparer cet organe du système auquel il est connecté, de lui appliquer un échelon de valeur k et de mesurer la valeur finale y_∞ de sa sortie en régime stationnaire. Sachant que

$$\begin{aligned} y_\infty &= \lim_{z \rightarrow 1} (1 - z^{-1})G(z)Zu \\ &= \lim_{z \rightarrow 1} (1 - z^{-1})G(z) \frac{k}{(1 - z^{-1})} = \frac{kb}{1 + a} \\ \Rightarrow b &= \frac{1 + a}{k} y_\infty. \end{aligned} \quad (6-2)$$

Cette méthode hors ligne, appelée en anglais *off-line identification*, a les inconvénients suivants :

1) La précision sur la valeur de b dépend de celle des autres paramètres. Par exemple, si Δk est l'incertitude sur la valeur k de l'entrée l'imprécision sur l'identification de b sera :

$$\Delta b = \frac{\partial b}{\partial k} \Delta k = -\frac{1+a}{k^2} y_\infty \Delta k = -\frac{b}{k} \Delta k. \quad (6-3)$$

On en déduit que Δb sera petit si l'amplitude k de l'entrée est grande. Ceci est général, l'action sur le système doit être grande dans une identification off-line ce qui n'est pas toujours toléré en pratique.

2) Le paramètre b peut varier durant le fonctionnement du système (variation de l'inertie, du frottement, ...) et il doit être continuellement ou périodiquement estimé afin d'adapter la politique de commande à sa variation. Dans ce cas fréquent, l'identification doit être nécessairement *on-line* c.à.d. durant le fonctionnement du système sans déconnecter du système l'organe à identifier.

La figure 6-1 montre le principe de l'identification on-line des paramètres d'un système.

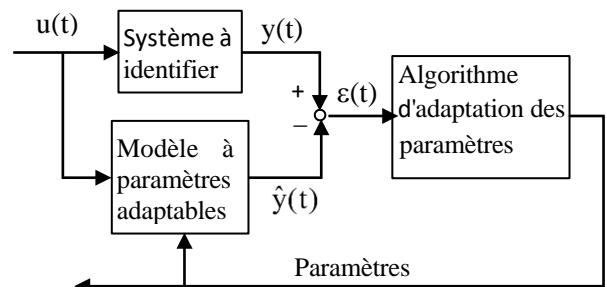


Fig. 6-1 Schéma montrant le déroulement de l'identification paramétrique

a) On associe au système à identifier, dont la transmittance $G(z)$ est inconnue, un modèle basé sur des considérations intuitives ou sur une analyse physique. En admettant que le système à identifier est linéaire, on propose un modèle dont la transmittance est de la forme :

$$\begin{aligned}\frac{Zy}{Zu} &= G(z, a_1, \dots, a_n, b_1, \dots, b_m) \\ &= \frac{b_1 z^{-1} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}}, \quad n \geq m.\end{aligned}$$

où les paramètres a_i et b_j sont à déterminer. À remarquer qu'on suppose que la réponse y est en retard par rapport à l'entrée u d'au moins une période ($Zy = z^{-1} \sum_{i=1}^n (b_i Zu - a_i Zy) z^{-(i-1)}$, $b_i = 0$ si $i > m$).

b) A chaque période d'échantillonnage, une entrée connue u est appliquée en même temps au système physique à identifier et au modèle proposé. On mesure à la fin de cette période la sortie y du système et on la compare avec la sortie \hat{y} *prédicté* par le modèle dont les paramètres sont estimés en se basant sur les mesures précédentes. Généralement un instrument à microprocesseur effectue ces estimations et les mémorise pour les exploiter à la période suivante.

c) En fonction de l'écart $\varepsilon = y - \hat{y}$, un *algorithme d'adaptation des paramètres* (AAP) réajuste les valeurs des paramètres du modèle pour mieux les adapter aux signaux u et y mesurés jusqu'à l'instant considéré. L'algorithme des moindres carrés récursifs, qui sera détaillé au paragraphe suivant, est le plus employé pour remplir cette tâche.

6-1 MOINDRES CARRÉS RÉCURSIFS

Dans la suite nous prenons comme unité du temps la période d'échantillonnage ($T = 1$). Durant une période t , c.à.d. entre $t - 1$ et t , le programme de prédiction (basé sur le modèle) associe aux paramètres du modèle des valeurs $\hat{a}_1(t), \dots, \hat{a}_n(t), \hat{b}_1(t), \dots, \hat{b}_m(t)$ et estime que la sortie à l'instant t sera

$$\begin{aligned}\hat{y}(t) &= -\hat{a}_1(t)y(t-1) - \dots - \hat{a}_n(t)y(t-n) \\ &\quad + \hat{b}_1(t)u(t-1) + \dots + \hat{b}_m(t)u(t-m), \quad (6-4) \\ y(t) &= 0 \quad \forall t \leq 0 \quad \text{et} \quad u(t) = 0 \quad \forall t < 0.\end{aligned}$$

Pour alléger l'écriture, on pose

$$\hat{\theta}^T(t) = [\hat{a}_1(t), \dots, \hat{a}_n(t), \hat{b}_1(t), \dots, \hat{b}_m(t)] \quad (6-5)$$

et

$$\varphi^T(t) = [-y(t-1), \dots, y(t-n), u(t-1), \dots, (t-m)]. \quad (6-6)$$

Le vecteur colonne $\hat{\theta}$ est appelé *vecteur des paramètres* et le vecteur colonne φ est appelé *vecteur de mesure ou d'observation*. Avec ces notations, le second membre de l'équation (6-4) s'écrit sous la forme d'un produit scalaire :

$$\hat{y}(t) = \hat{\theta}^T(t)\varphi(t). \quad (6-7)$$

En un instant t , l'écart entre la sortie mesurée $y(t)$ du système et la sortie prédicté $\hat{y}(t)$ est :

$$\varepsilon(t) = y(t) - \hat{y}(t) = y(t) - \hat{\theta}^T(t)\varphi(t). \quad (6-8)$$

Il est évident que $\varepsilon(t)$ dépend du choix du vecteur $\hat{\theta}(t)$, c.à.d. des valeurs associées aux paramètres du modèle durant la période t . Le modèle (6-7) représente correctement la dynamique du système à identifier si le vecteur $\hat{\theta}(t)$ vérifie les équations

$$y(i) - \hat{\theta}(t)\varphi(i) = 0. \quad \forall t \text{ et } \forall i = 1, 2, \dots, t. \quad (6-9)$$

Ceci signifie que si les paramètres estimés à l'instant t sont corrects, le modèle ayant ces paramètres doit produire les mêmes valeurs de la réponse réelle mesurées avant t . (6-9) est un système de t équations linéaires à $n + m$ inconnues (les paramètres). Mais, à cause des erreurs de mesure, ce système devient généralement incompatible quand $t \geq n + m$. Sa solution étant impossible, le mieux qu'on puisse faire consiste à déterminer à la période suivante le vecteur des paramètres $\hat{\theta}^T(t+1)$ qui minimise une certaine moyenne pondérée des $|\varepsilon(i)|$ ou des $\varepsilon^2(i)$. En d'autres termes, on associe à $\hat{\theta}^T(t+1)$ la valeur de θ qui minimise la fonction

$$J_t(\theta) = \sum_{i=1}^t \alpha_i \varepsilon^2(i) = \sum_{i=1}^t \alpha_i [y(i) - \theta^T \varphi(i)]^2 \quad (6-10)$$

où α_i est le poids associé au $i^{\text{ème}}$ écart $\varepsilon(i)$. La valeur de α_i dépend du degré de confiance qu'on accorde à

l'estimation à l'instant i et son choix sera discuté plus loin. Le vecteur des paramètres $\hat{\theta}^T(t+1)$ ainsi obtenu sert à calculer l'estimation $\hat{y}(t+1)$ de la sortie du système à l'instant $t+1$.

$J_t(\theta)$ est minimum au point où son gradient s'annule. D'où $\hat{\theta}^T(t+1)$ vérifie :

$$\frac{\delta J_t}{\delta \theta} = \left[\frac{\delta J_t}{\delta \theta_1} \dots \frac{\delta J_t}{\delta \theta_{n+m}} \right]^T = 0, \quad (6-11)$$

$$\theta_i = \begin{cases} a_i & \text{si } i \leq n \\ b_{i-n} & \text{si } i > n. \end{cases}$$

Or, pour $k = 1, \dots, n+m$, la dérivée de (6-10) par rapport à θ_k est :

$$\frac{\delta J_t(t)}{\delta \theta_k} = -2 \sum_{i=1}^t \alpha_i [y(i) - \theta^T \varphi(i)] \varphi_k(i) \quad (6-12)$$

où $\varphi_k(i)$ est la $k^{\text{ème}}$ composante de $\varphi(i)$. En écrivant les équations (6-12) sous une forme vectorielle, l'équation (6-11), que doit vérifier $\hat{\theta}^T(t+1)$, devient :

$$\sum_{i=1}^t \alpha_i [y(i) - \theta^T \varphi(i)] \varphi(i) = 0. \quad (6-13)$$

En remarquant que $\theta^T \cdot \varphi(i) \cdot \varphi(i) = \varphi(i) \cdot \varphi^T(i) \cdot \theta$ et en admettant que la matrice symétrique

$$P^{-1}(t) = \sum_{i=1}^t \alpha_i \varphi(i) \varphi^T(i) \quad (6-14)$$

est inversible, la solution de (6-13) est

$$\hat{\theta}(t+1) = P(t) \sum_{i=1}^t \alpha_i y(i) \varphi(i). \quad (6-15)$$

Selon cette relation, la détermination de $\hat{\theta}^T(t+1)$ nécessite la mémorisation de toutes les données $y(i)$ et $u(i)$ de $i = 0$ à $i = t$ ainsi que le calcul de l'inverse de la matrice (6-14). Pour réduire le temps de calcul ainsi que la dimension de la mémoire, il est avantageux de transformer (6-15) en des relations récursives liant $\hat{\theta}(t+1)$ à $\hat{\theta}(t)$ et $P(t)$ à $P(t-1)$ profitant ainsi des résultats obtenus à l'itération

précédente. L'équation (6-15) s'écrit :

$$\hat{\theta}(t+1) = P(t) \left[\sum_{i=1}^{t-1} \alpha_i y(i) \varphi(i) + \alpha_t y(t) \varphi(t) \right]$$

d'où, d'après (6-15) et (6-14),

$$\begin{aligned} \hat{\theta}(t+1) &= P(t) \left[P^{-1}(t-1) \hat{\theta}(t) + \alpha_t y(t) \varphi(t) \right] \\ &= P(t) \left\{ \left[P^{-1}(t) - \alpha_t \varphi(t) \varphi^T(t) \right] \hat{\theta}(t) + \alpha_t y(t) \varphi(t) \right\}. \end{aligned}$$

Tenant compte de l'expression (6-8) de l'erreur $\varepsilon(t)$, on obtient la relation récursive

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \alpha_t P(t) \varphi(t) \varepsilon(t). \quad (6-16)$$

où la matrice $P(t)$ doit être déterminée aussi d'une manière récursive. Or, d'après (6-14),

$$\begin{aligned} P^{-1}(t) &= P^{-1}(t-1) + \alpha_t \cdot \varphi(t) \cdot \varphi^T(t) \\ \Rightarrow P(t) &= [P^{-1}(t-1) + \alpha_t \cdot \varphi(t) \cdot \varphi^T(t)]^{-1}. \end{aligned} \quad (6-17)$$

Pour éliminer du second membre de (6-17) les opérations d'inversion qui consomment un long temps de calcul et cumulent les erreurs d'arrondi, on applique le lemme suivant :

Lemme

$$(A + B.C.D)^{-1} = A^{-1} - A^{-1}.B.(D.A^{-1}.B + C^{-1})^{-1}.D.A^{-1} \quad (6-18)$$

en supposant que les matrices A et C sont inversibles. Pour vérifier ce lemme, il suffit de montrer que la multiplication du second membre par l'inverse du premier est égale à la matrice unitaire I .

En remplaçant dans (6-18) les matrices A , B , C et D respectivement par $P^{-1}(t-1)$, $\varphi(t)$, α_t et $\varphi^T(t)$ et sachant que $[\varphi^T.P\varphi(t) + 1/\alpha_t]$ est un scalaire, l'équation (6-17) s'écrit :

$$P(t) = P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{1/\alpha_t + \varphi^T(t)P(t-1)\varphi(t)}. \quad (6-19)$$

Cette équation évite d'inverser $P^{-1}(t)$ à chaque itération et montre que si $P^{-1}(t-1)$ est inversible,

les matrices $P^{-1}(t)$, $P^{-1}(t+1)$, ... le sont aussi. L'algorithme des moindres carrés récursifs (MCR) se résume donc comme suit :

(MCR)

La matrice $P(t-1)$ et les vecteurs $\varphi(t)$ et $\hat{\theta}(t)$ obtenus durant l'étape t , sont mémorisés.

- 1) A l'instant t , on lit les valeurs de $y(t)$ et de $u(t)$.
- 2) Durant l'étape $t+1$, c'est-à-dire entre t et $t+1$, on calcule successivement :

a) $\hat{y}(t) = \hat{\theta}(t) \cdot \varphi(t)$,

b) $\varepsilon(t) = y(t) - \hat{y}(t)$,

c) $P(t) = P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{1/\alpha_t + \varphi^T(t)P(t-1)\varphi(t)}$

d) $\hat{\theta}(t+1) = \hat{\theta}(t) + \alpha_t P(t)\varphi(t)\varepsilon(t)$

e) $\varphi^T(t+1) = [-y(t), \varphi_1(t), \dots, \varphi_{(n-1)}(t),$

$u(t), \varphi_{(n+1)}(t), \dots, \varphi_{(n+m-1)}(t)]$.

Initialisation. L'algorithme ci-dessus n'est valable qu'à partir d'un instant t_0 où les matrices

$$P^{-1}(t-1) = \sum_{i=1}^{t-1} \alpha_i \varphi(i) \varphi^T(i), \quad t = t_0, t_0 + 1, \dots,$$

deviennent inversibles. Par conséquent, avant de démarrer l'algorithme, il faut effectuer des tests de régularité sur les matrices $P^{-1}(t-1)$ afin de détecter l'instant de démarrage t_0 . En pratique, on évite ces tests et on démarre l'algorithme dès la première période en posant $\hat{\theta}(1) = \varphi(1) = 0$ et en remplaçant $P(1)$ (qui n'existe pas) par la matrice diagonale $P_1 = c.I$ où c est un réel positif. Dans ces conditions et tenant compte de (6-14), au lieu de (6-15), les premières estimations du vecteur des paramètres seront données par

$$\hat{\theta}(t+1) = [P_1^{-1} + \sum_{i=2}^t \alpha_i \varphi(i) \varphi^T(i)]^{-1} \cdot \sum_{i=1}^t \alpha_i y(i) \varphi(i) \quad (6-15')$$

À noter que, d'après (6-19), la matrice entre crochets étant inversible pour $t = 1$ (car égale à P_1^{-1}), elle reste inversible $\forall t > 1$. D'autre part, les éléments diagonaux des matrices $\alpha_i \cdot \varphi(i) \cdot \varphi^T(i)$ étant positifs, leur somme S augmente avec t et, pour t suffisamment grand, les éléments diagonaux de $P_1^{-1} = \delta \cdot I$, $\delta = 1/c$, deviennent négligeables devant ceux de S . En

choisisant δ petit (typiquement, $\delta = 0.001$ c.à.d. $c = 1000$) l'équation (6-15)' tend rapidement vers (6-15).

6-2 INTERPRÉTATIONS DU MCR ET MISE EN ŒUVRE

Afin de montrer certaines propriétés concernant le comportement de l'algorithme MCR, deux interprétations différentes de cet algorithme sont données dans cette section. La première interprète MCR par le filtre de Kalman et la deuxième par la descente du gradient.

Filtre de Kalman.

Les équations d'état d'un système linéaire strictement propre ($m < n$) discret perturbé sont de la forme :

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + w(t), \\ y(t) &= Cx(t) + v(t). \end{aligned} \quad (6-20)$$

où $u(t) \in \mathbb{R}^p$ est l'entrée du système et $y(t) \in \mathbb{R}^q$ est sa sortie. Le vecteur $x(t) \in \mathbb{R}^n$ caractérise l'état interne du système à l'instant t mais il est généralement inaccessible entièrement à la mesure. Les vecteurs $w(t)$ et $v(t)$ sont des perturbations aléatoires qui affectent respectivement x et y . On suppose que ces perturbations vérifient les relations suivantes :

$$\begin{aligned} \mathbb{E}[w(t)] &= 0, \quad \mathbb{E}[w(t)w^T(t+\tau)] = \delta(\tau)R_1, \\ \mathbb{E}[v(t)] &= 0, \quad \mathbb{E}[v(t)v^T(t+\tau)] = \delta(\tau)R_2, \\ \mathbb{E}[w(t)v^T(t+\tau)] &= \delta(\tau)R_{12}. \end{aligned}$$

R_1 , R_2 et R_{12} sont des matrices constantes représentant respectivement en tout instant t les corrélations entre les composantes de w , les composantes de v et entre les composantes de w et celles de v . Ces corrélations sont nulles en deux instants différents (bruits blancs).

Connaissant l'état initial $x(0)$, Kalman calcule successivement l'estimation $\hat{x}(t+1)$ de $x(t+1)$, $t = 0, 1, 2, \dots$, en mesurant l'entrée $u(t)$ et la sortie $y(t)$ et en appliquant la relation

$$\hat{x}(t+1) = A \cdot \hat{x}(t) + B \cdot u(t) + K(t)[y(t) - C \cdot \hat{x}(t)]. \quad (6-21)$$

Le dernier terme corrige l'estimation en fonction de l'écart entre la valeur mesurée et la valeur estimée de y . La matrice $K(t)$ est obtenue par les relations récursives suivantes :

$$K(t) = [AP_1(t)C^T + R_{12}][CP_1(t)C^T + R_2]^{-1} \quad (6-22)$$

avec

$$P_1(0) = 0,$$

$$P_1(t+1) = [AP_1(t)A^T + R_1] - K(t)[CP_1(t)A^T + R_{12}^T]. \quad (6-23)$$

Dans ces équations, $P_1(t) = E[\varepsilon(t)\varepsilon^T(t)]$ est la matrice d'autocorrélation du vecteur d'écart $\varepsilon(t) = x(t) - \hat{x}(t)$. En minimisant la moyenne quadratique de $\varepsilon(t)$, $E[\|\varepsilon(t)\|^2] = E[\varepsilon^T(t)\varepsilon(t)]$, Kalman a démontré que, par ce choix de $K(t)$, $E[\|\varepsilon(t)\|^2] = \text{tr}[P_1(t)] \rightarrow 0$ quand t tend vers l'infini.

Estimation de θ par un filtre de Kalman.

Considérons de nouveau le système linéaire (6-3) dont les paramètres $a_1, \dots, a_n, b_1, \dots, b_m$ forment le vecteur à estimer θ . Ce système étant supposé invariant, on peut écrire :

$$\theta(t+1) = \theta(t) = \theta. \quad (6-24)$$

D'autre part, comme ci-dessus, on désigne par $\varphi(t)$ le vecteur de mesure défini par (6-6) et on écrit la sortie à l'instant t sous la forme :

$$y(t) = \varphi^T(t).\theta(t) + \beta_t.v(t) \quad (6-25)$$

où $\beta_t.v(t)$ représente une perturbation provenant de deux sources : de l'imprécision du modèle, des bruits ou des erreurs de mesure. Nous supposerons que $v(t)$ est un signal aléatoire de moyenne nulle et de variance σ^2 et β_t est un facteur qui diminue au fur et à mesure que l'identification du système se précise.

Maintenant, (6-24) et (6-25) sont des équations d'état de la forme (6-20) avec

$$A = I, \quad B = 0, \quad w(t) = 0, \quad C = \varphi^T(t) \text{ et } v(t) = \beta_t.v(t). \quad (6-26)$$

En admettant que, pour deux instants distincts t_1 et t_2 , les perturbations $v(t_1)$ et $v(t_2)$ ne sont pas corrélés, on peut estimer θ en appliquant l'algorithme de Kalman défini par (6-21), (6-22) et (6-23). Tenant compte de (6-26) et sachant que $R_1 = 0$, $R_{12} = 0$ et $R_2 = \beta_t^2\sigma^2$, on obtient :

$$\hat{\theta}(t+1) = \hat{\theta}(t) + K(t)[y(t) - \varphi^T(t).\hat{\theta}(t)], \quad (6-27)$$

$$K(t) = P_1(t).\varphi(t).[\varphi^T(t).P_1(t).\varphi(t) + \beta_t^2.\sigma^2]^{-1} \\ = \frac{P_1(t).\varphi(t)}{[\varphi^T(t).P_1(t).\varphi(t) + \beta_t^2.\sigma^2]} \quad (6-28)$$

$$P_1(t+1) = P_1(t) - K(t).\varphi^T(t).P_1(t). \quad (6-29)$$

En remplaçant dans (6-27) et (6-29) $K(t)$ par son expression (6-28), en désignant par $\varepsilon(t)$ l'erreur d'estimation $y(t) - \varphi^T(t).\hat{\theta}(t)$ et en posant

$$P(t) = P_1(t) / \sigma^2 \quad \text{et} \quad \alpha_t = 1 / \beta_t^2, \quad (6-30)$$

on retrouve les relations récursives de l'algorithme MCR :

$$P(t+1) = P(t) - \frac{P(t).\varphi(t)\varphi^T(t).P(t)}{1/\alpha_t + \varphi^T(t).P(t)\varphi(t)} \quad (6-31)$$

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \frac{P(t).\varphi(t)\varepsilon(t)}{1/\alpha_t + \varphi^T(t).P(t)\varphi(t)} \\ = \hat{\theta}(t) + \alpha_t P(t+1)\varphi(t)\varepsilon(t). \quad (6-32)$$

D'après Kalman, l'estimation $\hat{\theta}(t)$ tend vers le vrai vecteur des paramètres $\theta(t) = \theta$ quand t tend vers l'infini. Par conséquent, l'algorithme MCR converge vers le vrai vecteur des paramètres d'autant plus rapidement que la décroissance vers 0 de $\text{tr}[P(t)] = \sigma^2 E[\|\varepsilon(t)\|^2]$ est rapide. D'autre part, la relation $\alpha_t = 1 / \beta_t^2$ montre que le facteur de pondération α_t à la période t doit être d'autant plus grand que la prédiction $\varphi^T(t).\hat{\theta}(t)$ de la réponse y à cette période est précise c.à.d. quand t croît.

Estimation de θ par la descente du gradient.

Rappelons que l'estimation $\hat{\theta}(t+1)$ par l'algorithme MCR est déduite en minimisant la fonction

$$J_t(\theta) = \sum_{i=1}^t \alpha_i [y(i) - \varphi(i)^T \theta]^2 \quad \alpha_i > 0 \quad \forall i. \quad (6-32)$$

L'approximation de cette fonction par son développement de Taylor de second ordre autour de l'estimation $\hat{\theta}(t)$ à la période précédente est :

$$\begin{aligned} \tilde{J}_t(\theta) &= J_t(\hat{\theta}(t)) + \left(\frac{\partial J_t}{\partial \theta} \right)_{\hat{\theta}(t)} \cdot [\theta - \hat{\theta}(t)] \\ &+ \frac{1}{2} [\theta - \hat{\theta}(t)]^T \left(\frac{\partial^2 J_t}{\partial \theta^2} \right)_{\hat{\theta}(t)} [\theta - \hat{\theta}(t)] \end{aligned} \quad (6-33)$$

où

$\psi_t = \left(\frac{\partial J_t}{\partial \theta} \right)_{\hat{\theta}(t)}$ est le vecteur gradient ψ_t de J_t au point $\hat{\theta}(t)$,

$S_t = \left(\frac{\partial^2 J_t}{\partial \theta^2} \right)_{\hat{\theta}(t)}$ est la matrice Hessienne S_t de J_t au point $\hat{\theta}(t)$.

$\tilde{J}_t(\theta)$ est minimum quand son gradient s'annule, c'est-à-dire, d'après (6-33), au point $\tilde{\theta}(t+1)$ qui vérifie

$$\begin{aligned} \frac{\partial \tilde{J}_t}{\partial \theta} &= \left(\frac{\partial J_t}{\partial \theta} \right)_{\hat{\theta}(t)} + \left(\frac{\partial^2 J_t}{\partial \theta^2} \right)_{\hat{\theta}(t)} [\theta - \hat{\theta}(t)] \\ &= \psi_t + S_t [\theta - \hat{\theta}(t)] = 0. \end{aligned}$$

D'où

$$\tilde{\theta}(t+1) = \hat{\theta}(t) - S_t^{-1} \psi_t \quad (6-34)$$

en admettant que le Hessien S_t est inversible. Or, d'après (6-32),

$$\begin{aligned} \psi_t &= \left(\frac{\partial J_t}{\partial \theta} \right)_{\hat{\theta}(t)} = -2 \sum_{i=1}^t \alpha_i [y(i) - \varphi(i)^T \hat{\theta}(t)] \varphi(i) \\ &= \left(\frac{\partial J_{t-1}}{\partial \theta} \right)_{\hat{\theta}(t)} - 2 \alpha_t \varphi(t) \varepsilon(t) \end{aligned} \quad (6-35)$$

où $\varepsilon(t)$ désigne toujours l'écart $y(t) - \varphi^T(t) \hat{\theta}(t)$.

Mais, comme $\hat{\theta}(t)$ minimise J_{t-1} , il annule le gradient $(\partial J_{t-1} / \partial \theta)_{\hat{\theta}(t)}$ et (6-35) devient :

$$\psi_t = -2 \alpha_t \varphi(t) \varepsilon(t). \quad (6-36)$$

D'autre part, en dérivant deux fois (6-32) par rapport à θ et tenant compte de (6-14), on obtient :

$$\begin{aligned} S_t &= \left(\frac{\partial^2 J_t}{\partial \theta^2} \right)_{\hat{\theta}(t)} = 2 \sum_{i=1}^t \alpha_i \varphi^T(i) \varphi(i) \\ &= 2 P^{-1}(t). \end{aligned} \quad (6-37)$$

Enfin, en remplaçant les expressions de ψ_t et de S_t dans (6-34), on trouve :

$$\tilde{\theta}(t+1) = \hat{\theta}(t) + \alpha_t P(t) \varphi(t) \varepsilon(t). \quad (6-38)$$

La comparaison de (6-38) avec (6-16) montre que $\tilde{\theta}(t+1) = \hat{\theta}(t+1)$ c.à.d. la descente du gradient conduit au même résultat que le MCR. D'autre part, d'après (6-34), on obtient $\hat{\theta}(t+1)$ à partir de $\hat{\theta}(t)$ en ajoutant à ce dernier le vecteur $-S_t^{-1} \psi_t$. Rappelons que l'opposé $-\psi_t$ du gradient au point $\hat{\theta}(t)$ a la direction de la plus forte pente vers les valeurs inférieures de J_t (convergence rapide de ε vers 0). Mais, comme cette direction ne mène pas généralement vers le minimum de J_t , elle est ajustée par la transformation S_t^{-1} (fig. 6-2).

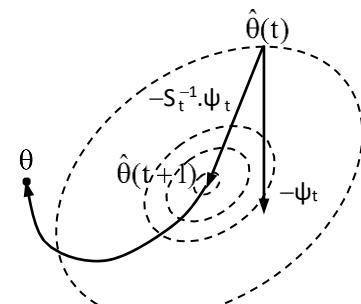


Fig. 6-2 Descente du gradient

Ainsi, quand t croît, le point minimum $\hat{\theta}(t+1)$ de la fonction $J_t(\theta)$ (ou de $\tilde{J}_t(\theta)$) décrit une trajectoire qui tend vers le vrai vecteur θ des paramètres comme nous l'avons démontré ci-dessus.

Choix des facteurs de pondération α_i .

Généralement le choix du facteur de pondération α_i , $i = 1, 2, \dots$ de la fonction à minimiser,

$$J_t(\theta) = \sum_{i=1}^t \alpha_i \varepsilon^2(i) = \sum_{i=1}^t \alpha_i [y(i) - \theta^T \varphi(i)]^2,$$

est d'autant plus grand que la valeur de $\varepsilon^2(i)$ est fiable. Quand le système est invariant, il est rationnel d'associer à toutes les périodes le même facteur de pondération $\alpha_i = 1$. Cependant, comme durant les premières périodes la matrice $P(t)$ est encore affectée par la matrice d'initialisation arbitraire $P_1 = c \cdot I$ et que l'estimation $\hat{\theta}$ est encore loin du vrai vecteur θ , il est judicieux d'accorder moins d'importance aux $\varepsilon^2(i)$ relatives à ces périodes. Ceci peut se faire en associant à la période i le facteur de pondération

$$\alpha_i = 1 - \lambda^i (1 - \alpha_0). \quad (6-40)$$

Typiquement, on choisit λ et α_0 entre 0.9 et 1 d'où, au départ, $\alpha_i < 1$ mais à partir d'une certaine période tous les α_i auront presque la même valeur 1. La constante λ est appelée *facteur d'oubli*.

Choix du signal d'entrée. Soit

$$G(s) = K \frac{s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m}{s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n}, \quad n \geq m,$$

la fonction de transfert d'un système comportant $n+m+1$ paramètres, a_1, \dots, a_n , b_1, \dots, b_m et K à déterminer. La réponse stationnaire de ce système à une entrée ayant p harmoniques,

$$u(t) = u_0 + \sum_{i=1}^p u_i \sin(\omega_i t), \quad t \geq 0,$$

a pour expression (voir section 5-3)

$$y(t) = K \frac{b_m}{a_n} u_0 + \sum_{i=1}^p |G(j\omega_i)| u_i \sin[\omega_i t + \phi(\omega_i)]$$

où $\phi(\omega_i)$ est l'argument de $G(j\omega_i)$. Imaginons des filtres qui isolent les harmoniques de la sortie $y(t)$ l'un de l'autre au lieu d'être superposés. On obtient à la sortie de ces filtres un signal constant y_0 et des signaux sinusoïdaux $y_i(t)$, $i = 1, 2, \dots, p$, vérifiant les $2p + 1$ équations suivantes :

$$y_0 = K \frac{b_m}{a_n} u_0, \\ |y_i(t)| = |G(j\omega_i)u_i|, \quad \arg[y_i(t)] = \phi(\omega_i), \quad i \geq 1,$$

dont les inconnues sont les $n + m + 1$ paramètres de $G(s)$. Par conséquent, pour estimer ces paramètres d'une manière suffisamment précise l'entrée $u(t)$ doit comporter, en plus de la constante u_0 , un nombre d'harmoniques p au moins égal à $(n + m)/2$. Plus p est grand plus l'estimation des paramètres sera meilleure et on dit que $u(t)$ doit être "riche en fréquences". Le signal le plus riche en fréquences est le bruit blanc car, son spectre étant constante sur $(-\infty, +\infty)$, il contient toutes les fréquences. Mais, comme le bruit blanc n'est qu'un signal théorique impossible à le réaliser, on le remplace en pratique par une approximation dont le spectre est constant sur une large bande de fréquences (band-limited white noise).

6-3 SIMULATION PAR MATLAB

Le toolbox Simulink est muni d'un bloc appelé « S-function » dans lequel on peut incréer un M-file dont la structure est décrite dans le « help » de Matlab. Ce M-file détermine la relation entre les entrées de ce bloc et ses sorties. En général, cette relation n'est pas une simple expression mathématique mais elle se définit par un programme. Voici, à titre d'exemple, un programme nommé "mcri" associé à un bloc « S-function » pour simuler l'algorithme MCR.

```
function [sys,x0,str,ts]
    = mcri(t,x,v,flag,n,m,lambda,dt)
```

% MCR est une fonction-S pour estimer les paramètres d'un système linéaire invariant non perturbé

% Les arguments t , x , v et $flag$ sont introduits automatiquement par simulink à chaque itération.

% t = instant du simulateur.

% $v = [u(t), y(t)]'$ = entrée et sortie à l'instant t .

% $x = [\theta ; P(:) ; \phi ; \alpha]$ = vecteur colonne appelé état, constitué à l'instant t des estimations des paramètres, des éléments de la matrice P , du vecteur d'observation ϕ et de la pondération α .

% $flag$ = code qui indique ce que la fonction doit

associer à sys.

% Les autres arguments de la fonction mcri sont :
 % n et m = respectivement les degrés du dénominateur et du numérateur.
 % lamda = le facteur d'oubli
 % dt = la période d'échantillonnage.
 % Voir SFUNTMPL pour la signification des valeurs de flag.

Le programme mcri est le suivant.

```
r= n+m ;

if flag == 2
% La fonction mcri associe à sys le nouvel état
% x(t+1).
  theta = x(1 : r);
  P = zeros(r, r);
  P(:) = x(r + 1 : r + r^2);
  phi = x(r + r^2 + 1 : 2*r + r^2);
  alpha = x(2*r + r^2 + 1);
  u = v(1); y = v(2);
  epsilon = y - theta'*phi;
  den = (1/alpha) + phi'*P*phi;
  nvP = P - (P*phi*phi'*P)/den;
  nvtheta = theta + alpha*nvP*phi*epsilon;
  nvphi = [-y; phi(1 : n-1)]; u; phi(n+1:r1)];
  nvalpha = 1 - lamda*(1 - alpha);
  sys = [nvtheta; nvP(:); nvphi; nvalpha];

elseif flag == 0
% sys = les dimensions et les conditions initiales.

% Dimensions et mode de programmation.
  sys(1) = 0;
% Pas d'états continus.
  sys(2) = 2*r+(r^2)+1;
% Dimension de x.
  sys(3) = r+1;
% Dimension de la sortie: theta + trace.
  sys(4) = 2;
% Dimension de l'entrée v = [u, y].
  sys(5) = 0;
% Pas de racines à déterminer.
  sys(6) = 0;
% L'entrée ne sera pas lue durant flag 3.
  sys(7) = 1;
% Période d'échantillonnage constante.

% Initialisation
  P = 1000*eye(r);
```

```
x0 = [zeros(r,1); P(:); zeros(r,1); 0.98];
str = [];
% c'est toujours vide.
ts = [dt 0];
% dt = période T ; pas de décalage (= 0).

elseif flag == 3
% définition de la sortie.
  P = zeros(r, r);
  P(:) = x(r+1 : r+(r^2));
  tr = trace(P);
  sys = [x(1:r);tr];
  % c.à.d. theta et la trace de P.

else
  sys = [];
  % les autres flags n'ont pas de rôle ici.
end
```

Pour enrichir en fréquences le signal agissant sur le système à identifier on lui ajoute un bruit blanc dont la période d'échantillonnage est d'environ 0.01 fois la constante de temps du système. Sans trop affecter le fonctionnement du système, la puissance de ce bruit doit être suffisante pour que son effet soit détectable à la sortie. Admettons que la transmittance inconnue du système à identifier est

$$G(z) = \frac{0.2553z - 0.06266}{z^2 - 1.414z + 0.6065}.$$

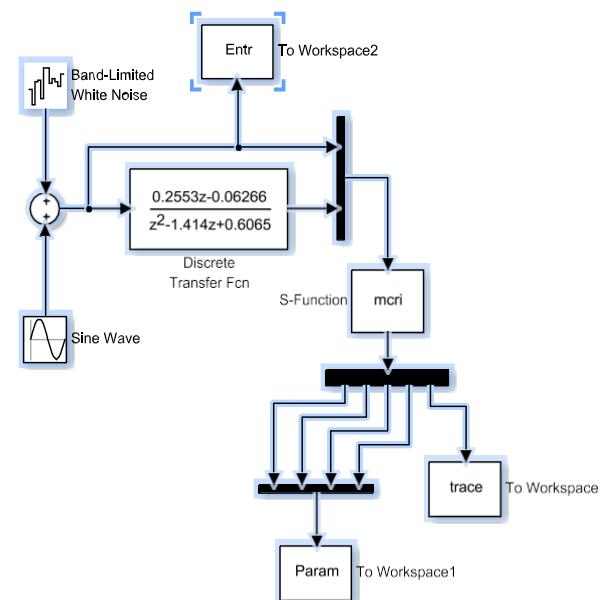
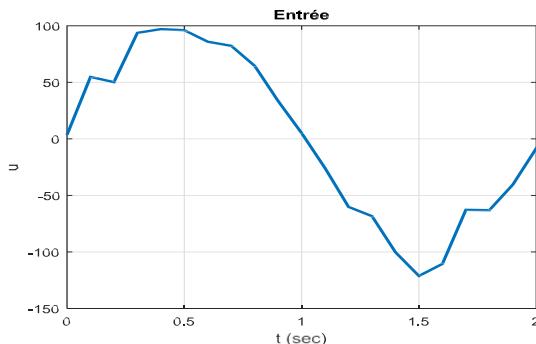


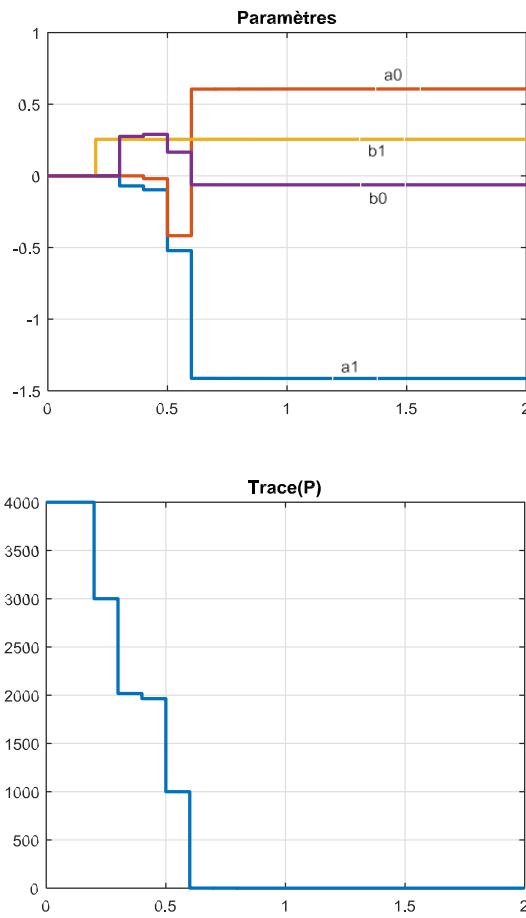
Fig. 6-3 Le simulateur MCR

En ajoutant le bruit blanc au signal d'action u (supposé ici sinusoïdal), et en introduisant l'entrée



résultante et la sortie du système à identifier au bloc S-function muni de l'algorithme «mcri» ci-dessus (fig. 6-3), on obtient les résultats suivants :

Cette entrée est la somme d'une fonction sinusoïdale et d'un bruit blanc.



Les valeurs des paramètres à la fin de la simulation (qu'on peut lire à l'aide du curseur sur la fenêtre

graphique) sont presque égales aux vrais paramètres du système :

$$a_1 = -1.4140, \quad a_0 = 0.6065, \\ b_1 = 0.2553, \quad b_0 = -0.0627$$

et la valeur finale de la trace qui mesure la moyenne quadratique de l'erreur n'est que 0.0049.

Identification des systèmes perturbés.

L'algorithme d'identification par moindres carrés récursifs (MCR) sert à identifier des systèmes déterministes dont la perturbation à sa sortie peut être considéré négligeable. Or, dans la plupart des cas pratiques, ces systèmes sont soumis à des perturbations aléatoires (variation de la tension d'alimentation, de la charge, des conditions ambiantes, des erreurs de mesure, ...). Pour tenir compte de cette réalité, au lieu de $y(t) = \varphi^T(t)\theta$, nous considérons dans ce qui suit des systèmes dont le modèle est de la forme

$$y(t) = \varphi^T(t)\theta + v(t) \quad (6-41)$$

où $v(t)$ est une variable aléatoire de moyenne nulle représentant l'effet de l'ensemble des perturbations sur la sortie du système. Comme ci-dessus, θ et $\varphi(t)$ désignent respectivement le vecteur des paramètres et le vecteur des observations. Rappelons que si la transmittance du système est écrite sous la forme

$$G(s) = \frac{Zy}{Zu} = \frac{b_1 z^{-1} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}},$$

les vecteurs θ et $\varphi(t)$, $t = 1, 2, \dots$, sont définis par

$$\theta^T = (a_1, \dots, a_n, b_1, \dots, b_m),$$

$$\varphi^T(t) = [-y(t-1), \dots, -y(t-n), u(t-1), \dots, u(t-m)].$$

Rappelons aussi que l'algorithme des moindres carrés récursifs (MCR) découle de l'estimation à la période $(t + 1)$ du vecteur θ des paramètres par application des relations (6-14) et (6-15) :

$$\hat{\theta}(t+1) = P(t) \sum_{i=1}^t \alpha_i y(i) \cdot \varphi(i), \quad (6-42)$$

$$P^{-1}(t) = \sum_{i=1}^t \alpha_i \varphi(i) \cdot \varphi^T(i). \quad (6-43)$$

Cette estimation est la meilleure selon le critère des moindres carrés qui consiste à minimiser la fonction

$$J_t(\theta) = \sum_{i=1}^t \alpha_i \varepsilon^2(i) = \sum_{i=1}^t \alpha_i [y(i) - \theta^T \varphi(i)]^2 \quad (6-44)$$

Nous savons de la section précédente que si le système à identifier est déterministe ($v(t) = 0 \forall t$), l'estimation (6-42) converge, en moyenne quadratique, vers le vrai vecteur θ des paramètres quand t tend vers l'infini. Voyons si c'est le cas en présence des perturbations. En remplaçant la nouvelle expression (6-41) de $y(t)$ dans la relation (6-42), on obtient :

$$\begin{aligned} \hat{\theta}(t+1) &= P(t) \sum_{i=1}^t \alpha_i [\varphi^T(i)\theta + v(i)]\varphi(i), \\ &= P(t) \underbrace{\sum_{i=1}^t \alpha_i \varphi(i)\varphi^T(i)}_{P^{-1}(t)} \theta + P(t) \sum_{i=1}^t \alpha_i v(i)\varphi(i). \\ &\quad (\text{car } \varphi^T(i)\theta\varphi(i) = \varphi(i)\varphi^T(i)\theta). \\ \Rightarrow \hat{\theta}(t+1) &= \theta + P(t) \sum_{i=1}^t \alpha_i v(i)\varphi(i) \\ \text{ou } \hat{\theta}(t+1) &= \theta + [tP(t)] \left[\frac{1}{t} \sum_{i=1}^t \alpha_i v(i)\varphi(i) \right]. \quad (6-45) \end{aligned}$$

Or, en établissant l'algorithme MCR, nous avons vu que la matrice $P(t)$ (ou $[t.P(t)]$) est inversible pour tout t et, en modélisant la perturbation aléatoire, nous avons montré au chapitre 5 précédent qu'elle constitue un processus ergodique. Par conséquent, d'après (6-45), l'estimation $\hat{\theta}(t+1)$ ne tendra vers le vrai vecteur des paramètres θ que si

$$\lim_{t \rightarrow \infty} \left[\frac{1}{t} \sum_{i=1}^t \alpha_i v(i)\varphi(i) \right]_{\text{ps}} = E[\alpha_i v(t)\varphi(t)] = 0. \quad (6-46)$$

Cette condition signifie qu'en tout instant t , la perturbation $v(t)$ ne doit pas se corrélérer avec le vecteur $\varphi(t)$. Or, la $k^{\text{ème}}$ composante $-y(t-k)$, $k = 1, \dots, n$, de $\varphi(t)$ est corrélé avec $v(t-k)$, $v(t-k-1)$, ..., d'où $\varphi(t)$ est en corrélation avec $v(t-1)$, $v(t-2)$, Comme en pratique la perturbation $v(t)$ à l'instant t est souvent corrélée avec les perturbations précédentes $v(t-k)$, $k = 1, 2, \dots$, elle l'est aussi avec $\varphi(t)$. Par conséquent, en présence des perturbations,

l'estimation de θ par l'algorithme MCR comporte généralement un *biais* (une erreur) sauf si $v(t)$ est un bruit blanc comme dans le cas du modèle ARX défini par $A(q^{-1})y(t) = B(q^{-1})u(t) + e(t)$.

Dans ce qui suit, nous décrirons les principales méthodes qui permettent de modifier l'algorithme MCR afin d'éliminer le biais d'estimation. Ces méthodes peuvent se classer en deux catégories :

- Par blanchiment de la perturbation.
- Par décorrélation entre la perturbation et le vecteur des mesures.

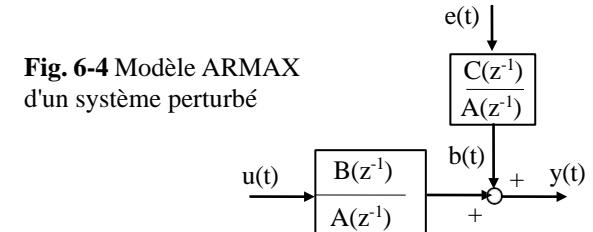
Dans la première catégorie, on transforme le modèle (6-41) de sorte que la perturbation soit un bruit blanc $e(t)$ et dans la deuxième catégorie, on transforme le modèle de sorte que la perturbation et le vecteur des mesures soient non corrélés.

6-4 METHODE DE PRÉDICTION RECURSIVE DE L'ERREUR (RPEM)

C'est une méthode par blanchiment de la perturbation $v(t)$. Elle consiste à transformer le modèle (6-41) sous la forme :

$$y(t) = \varphi_1^T(t) \cdot \theta_1 + e(t) \quad (6-47)$$

où $e(t)$ est un bruit blanc, θ_1 et $\varphi_1(t)$ sont des vecteurs modifiés des paramètres et des observations.



Nous avons vu au chapitre 5 précédent qu'un système perturbé peut se ramener à un processus ARMAX comme le montre la figure 6-4 où $e(t)$ est un bruit blanc et

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \cdots + a_n q^{-n}, \\ B(q^{-1}) &= b_1 q^{-1} + \cdots + b_m q^{-m}, \\ C(q^{-1}) &= 1 + c_1 q^{-1} + \cdots + c_p q^{-p}. \end{aligned}$$

On rappelle que les racines des polynômes A et C sont intérieures au cercle unité et la figure 6-4 est une représentation de l'équation

$$A(q^{-1}) \cdot y(t) = B(q^{-1}) \cdot u(t) + C(q^{-1}) \cdot e(t).$$

La forme détaillée de cette relation est

$$\begin{aligned} y(t) &= -a_1 \cdot y(t-1) - \cdots - a_n \cdot y(t-n) \\ &\quad + b_1 \cdot u(t-1) + \cdots + b_m \cdot u(t-m) \\ &\quad + c_1 \cdot e(t-1) + \cdots + c_p \cdot e(t-p) + e(t). \end{aligned} \quad (6-48)$$

En posant maintenant

$$\varphi_1^T(t) = [-y(t-1), \dots, -y(t-n), u(t-1), \dots, u(t-m), e(t-1), \dots, e(t-p)] \quad (6-49)$$

et

$$\theta_1^T = [a_1, \dots, a_n, b_1, \dots, b_m, c_1, \dots, c_p], \quad (6-50)$$

l'équation (6-48) prend la forme (6-47). A noter que le nombre des paramètres à estimer est devenu $n+m+p$ au lieu de $n+m$ dans l'algorithme MCR.

Malheureusement, le vecteur $\varphi_1(t)$ ne peut être utilisé tel qu'il est défini par (6-49) puisque ses composantes $e(t-1), \dots, e(t-p)$ ne sont pas mesurables. Mais, d'après (6-48), on remédie à cette difficulté en estimant $e(t)$ par

$$\begin{aligned} \varepsilon_1(t) &= y(t) + \hat{a}_1(t)y(t-1) + \cdots + \hat{a}_n(t)y(t-n) \\ &\quad - \hat{b}_1(t)u(t-1) - \cdots - \hat{b}_m(t)u(t-m) \\ &\quad - \hat{c}_1(t)\varepsilon_1(t-1) - \cdots - \hat{c}_p(t)\varepsilon_1(t-p) \end{aligned}$$

où les $\hat{a}_i(t), \hat{b}_i(t), \hat{c}_i(t)$ sont les estimations des paramètres du "procédé + perturbations" à l'instant t considéré. Pour alléger l'expression de $\varepsilon_1(t)$, on pose :

$$\hat{\theta}_1^T(t) = [\hat{a}_1, \dots, \hat{a}_n, \hat{b}_1, \dots, \hat{b}_m, \hat{c}_1, \dots, \hat{c}_p](t) \quad (6-51)$$

et

$$\begin{aligned} \psi^T(t) &= [-y(t-1), \dots, -y(t-n) \\ &\quad u(t-1), \dots, u(t-m), \\ &\quad \varepsilon(t-1), \dots, \varepsilon(t-p)] \end{aligned} \quad (6-52)$$

d'où

$$\varepsilon_1(t) = y(t) - \psi^T(t) \cdot \hat{\theta}_1(t). \quad (6-53)$$

En remplaçant dans (6-48) les $e(t-i)$ par $\varepsilon(t-i)$, $i = 1, \dots, p$, on obtient le modèle approximatif :

$$y(t) = \psi^T(t) \cdot \theta_1(t) + e(t). \quad (6-54)$$

Comme le bruit blanc $e(t)$ est indépendant de tout ce qui se passe avant l'instant t, il n'est pas corrélé avec $\psi(t)$ défini par (6-51) et la condition (6-46) qui devient $E[\alpha_i e(t) \psi(t)] = 0$ est maintenant satisfaite.

La méthode de la prédiction récursive de l'erreur (RPEM) consiste à appliquer l'algorithme MCR au modèle approximatif (6-54), c'est-à-dire à procéder de la manière suivante :

(RPEM)

La matrice $P(t-1)$ et les vecteurs $\psi(t)$ et $\hat{\theta}_1(t)$, obtenus durant l'étape t, sont stockés dans la mémoire.

- 1) A l'instant t, on lit les valeurs de $y(t)$ et de $u(t)$.
- 2) Durant l'étape $t+1$, c'est-à-dire entre t et $t+1$, on calcule successivement :

- a) $\hat{y}(t) = \psi^T(t) \cdot \hat{\theta}_1(t)$
- b) $\varepsilon_1(t) = y(t) - \hat{y}(t)$
- c) $P(t) = P(t-1) - \frac{P(t-1) \cdot \psi(t) \cdot \psi^T(t) \cdot P(t-1)}{1/\alpha_t + \psi^T(t) \cdot P(t-1) \cdot \psi(t)}$
- d) $\hat{\theta}_1(t+1) = \hat{\theta}_1(t) + \alpha_t \cdot P(t) \cdot \psi(t) \cdot \varepsilon_1(t)$
- e) $\psi_{i+1}(t+1) = \psi_{i-1}(t) \quad \forall i \text{ sauf}$
 $\psi_1(t+1) = y(t), \psi_{(n+1)}(t+1) = u(t), \psi_{(n+m+1)}(t+1) = \varepsilon_1(t).$

Mais, l'approximation (6-54), qui remplace la vraie équation (6-48) du modèle, n'est-elle pas une cause pour produire de nouveau un biais d'estimation ? Pour répondre à cette question, remplaçons dans (6-42) et (6-43) de l'algorithme MCR $\varphi(i)$ par $\psi(i)$ afin d'obtenir les expressions de l'estimation de θ_1 par l'algorithme RPEM :

$$\hat{\theta}_1(t+1) = P(t) \sum_{i=1}^t \alpha_i y(i) \cdot \psi(i), \quad (6-55)$$

$$P^{-1}(t) = \sum_{i=1}^t \alpha_i \psi(i) \cdot \psi^T(i). \quad (6-56)$$

Comme il est supposé que le modèle (6-47) représente correctement le système perturbé, l'équation (6-55) s'écrit :

$$\hat{\theta}_1(t+1) = P(t) \sum_{i=1}^t \alpha_i [\varphi_i^T(i) \theta_1 + e(i)] \cdot \psi(i),$$

En retranchant $\psi^T(i)$ de $\varphi_i^T(i)$ et en lui ajoutant $\psi^T(i)$, tenant compte de (6-56) et sachant que $\psi^T(i)\theta_1\psi(i) = \psi(i)\psi^T(i)\theta_1$, on obtient

$$\begin{aligned} \hat{\theta}_1(t+1) &= \theta_1 \\ &+ P(t) \sum_{i=1}^t \alpha_i \psi(i) [\varphi_i^T(i) - \psi^T(i)] \theta_1 \quad (6-57) \\ &+ P(t) \sum_{i=1}^t \alpha_i \psi(i) e(i). \end{aligned}$$

Le troisième terme du second membre de (6-57) s'annule presque sûrement quand $t \rightarrow \infty$ car le bruit blanc $e(i)$ est non corrélé avec $\psi(i)$.

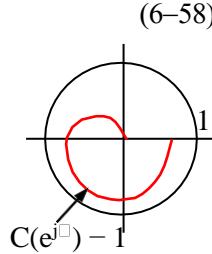
D'autre part,

$$\begin{aligned} [\varphi_i^T(i) - \psi^T(i)] \theta_1 &= \sum_{k=1}^p c_k [e(i-k) - \varepsilon_i(i-k)] \\ &= [C(q^{-1}) - 1] [e(i-k) - \varepsilon_i(i-k)]. \end{aligned}$$

D'après le théorème du petit gain, qui sera démontré dans la partie IV section 3-5, si

$$|C(e^{j\omega}) - 1| < 1 \quad \forall \omega \quad (6-58)$$

c.à.d. si la courbe $C(e^{j\omega}) - 1$, dite courbe de Nyquist de $C^*(z) = C(z) - 1$ reste à l'intérieur du cercle unité quand ω varie de 0 à l'infini, le deuxième terme de (6-57) tend vers zéro. D'où, sous la condition (6-58), $\hat{\theta}_1(t+1) \rightarrow \theta_1$ mais cette condition n'étant pas nécessaire, la convergence sans



biais du RPEM vers θ_1 peut se réaliser sans que (6-58) soit satisfaite.

6-5 METHODE DU MAXIMUM DE VRAISEMBLANCE (ML)

Contrairement à l'algorithme RPEM, la convergence vers θ_1 par la méthode du maximum de vraisemblance est assurée sans condition. Considérons de nouveau l'équation liant $y(t)$ à $u(t)$ et $e(t)$:

$$A(q^{-1}) \cdot y(t) = B(q^{-1}) \cdot u(t) + C(q^{-1}) \cdot e(t). \quad (6-59)$$

En posant

$$A^*(q^{-1}) = A(q^{-1}) - 1 = a_1 \cdot q^{-1} + \dots + a_n \cdot q^{-n}, \quad (6-60)$$

$$C^*(q^{-1}) = C(q^{-1}) - 1 = c_1 \cdot q^{-1} + \dots + c_p \cdot q^{-p}, \quad (6-61)$$

l'équation (6-59) devient

$$\begin{aligned} y(t) &= \quad (6-62) \\ &- A^*(q^{-1}) \cdot y(t) + B(q^{-1}) \cdot u(t) + C^*(q^{-1}) \cdot e(t) + e(t) \end{aligned}$$

qui s'écrit

$$y(t) = z(t) + e(t) \quad (6-63)$$

avec

$$\begin{aligned} z(t) &= \quad (6-64) \\ &- A^*(q^{-1}) \cdot y(t) + B(q^{-1}) \cdot u(t) + C^*(q^{-1}) \cdot e(t). \end{aligned}$$

$z(t)$ n'est autre que $\varphi_1^T(t) \theta_1$ de l'équation (6-47).

Pour déterminer son expression, remplaçons dans l'équation précédente $e(t)$ par $y(t) - z(t)$ et arrangeons les termes, on obtient :

$$\begin{aligned} z(t) &= \quad (6-65) \\ &- A^*(q^{-1}) \cdot \frac{y(t)}{C(q^{-1})} + B(q^{-1}) \cdot \frac{u(t)}{C(q^{-1})} + C^*(q^{-1}) \cdot \frac{y(t)}{C(q^{-1})}. \end{aligned}$$

On estime donc $y(t)$ par

$$\hat{y}(t) = \hat{\varphi}_1(t) \hat{\theta}_1(t) = \hat{z}(t) \quad (6-66)$$

$$= -\hat{A}^*(q^{-1}) \cdot \frac{y(t)}{\hat{C}(q^{-1})} + \hat{B}(q^{-1}) \cdot \frac{u(t)}{\hat{C}(q^{-1})} + \hat{C}^*(q^{-1}) \cdot \frac{y(t)}{\hat{C}(q^{-1})}.$$

En retranchant $[\hat{C}^*(q^{-1}) / \hat{C}(q^{-1})] \hat{y}(t)$ des deux membres de l'équation précédente et en désignant par $\varepsilon(t)$ l'erreur de prédiction $[y(t) - \hat{y}(t)]$, on obtient :

$$\hat{y}_f(t) = -\hat{A}^*(q^{-1})y_f(t) + \hat{B}(q^{-1})u_f(t) + \hat{C}^*(q^{-1})\varepsilon_f(t) \quad (6-67)$$

où $x_f(t) = x(t)/\hat{C}(q^{-1})$ est le filtrage d'un signal $x(t)$ par $1/\hat{C}(q^{-1})$. En d'autres termes,

$$x_f(t) = x(t) - \hat{c}_1 x_f(t-1) - \dots - \hat{c}_p x_f(t-p). \quad (6-68)$$

Ainsi, tenant compte de (6-62), l'équation (6-67) peut s'écrire sous la forme

$$\hat{y}_f(t) = \varphi_f^T(t) \hat{\theta}_1(t) \quad (6-69)$$

avec

$$\varphi_f^T(t) = [-y_f(t-1), \dots, -y_f(t-n), u_f(t-1), \dots, u_f(t-m), \varepsilon_f(t-1), \dots, \varepsilon_f(t-p)].$$

On retrouve donc la même forme d'estimation que le RPEM à part que les signaux u et y sont maintenant filtrés par $1/\hat{C}(q^{-1})$ et que les composantes $\varepsilon_f(t-i)$ de $\varphi_f(t)$ ne sont plus des approximations comme c'était le cas des composantes $\varepsilon_i(t-i)$ de $\psi(t)$. En appliquant la méthode RPEM aux signaux u_f , y_f et ε_f , on obtient l'algorithme du maximum de vraisemblance récursif. Cet algorithme a l'avantage de converger vers le vrai vecteur des paramètres $\hat{\theta}_1$ sans être soumis à la condition (6-58). Néanmoins, si les pôles de $1/\hat{C}(z^{-1})$ ne sont pas à l'intérieur du cercle unité à chaque période, les signaux y_f , u_f et ε_f risquent de diverger. Pour remédier à cette difficulté, on prend la précaution de remplacer

$$\hat{C}(q^{-1}) = 1 + \hat{c}_1(t)q^{-1} + \dots + \hat{c}_p(t)q^{-p}$$

par

$$\hat{C}_s(q^{-1}) = 1 + \gamma(t)\hat{c}_1(t)q^{-1} + \dots + \gamma^p(t)\hat{c}_p(t)q^{-p} \quad (6-70)$$

où

$$\gamma(t) = (1 - \gamma_0^t), \quad 0.5 \leq \gamma_0 \leq 0.99.$$

Comme $\gamma(t) < 1$ mais tend vers 1 quand $t \rightarrow \infty$, son rôle consiste à rapprocher les zéros de $\hat{C}(z^{-1})$ de l'origine durant les premières périodes afin que $1/\hat{C}_s(z^{-1})$ soit stable quand il est encore loin de $1/C(z^{-1})$ (fig. 6-5). Pour les grandes valeurs de t le

problème de stabilité ne se pose plus puisque, les $\hat{c}_i(t)$ devenant voisins des $c_i(t)$, $1/\hat{C}_s(z^{-1})$ sera voisin de $1/C(z^{-1})$ qui est par construction stable.

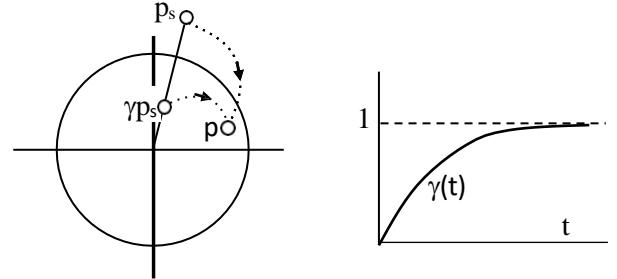


Fig. 6-5 Contraction des pôles

Au lieu de (6-68), la contraction des racines de $\hat{C}(z^{-1})$ nous conduit à définir le filtrage d'un signal $x(t)$ (y , u et ε) par

$$x_f(t) = x(t) - [\gamma(t)\hat{c}_1 x_f(t-1) + \dots + \gamma^p(t)\hat{c}_p x_f(t-p)].$$

En définitive, la méthode du maximum de vraisemblance se résume comme suit :

(ML)

La matrice $P(t-1)$, les vecteurs $\varphi_f(t)$, $\hat{\theta}_1(t)$ et le scalaire $\gamma(t-1)$ obtenus durant l'étape t , sont stockés dans la mémoire.

- 1) A l'instant t , on lit les valeurs de $y(t)$ et de $u(t)$.
- 2) Durant l'étape $t+1$, c'est-à-dire entre t et $t+1$, on calcule successivement :
 - a) $\gamma(t) = \gamma_0 \cdot \gamma(t-1) + (1 - \gamma_0)$, $(\gamma(0) = 0)$,
 - b) $y_f(t) = y(t) - [\gamma(t)\hat{c}_1 y_f(t-1) + \dots + \gamma^p(t)\hat{c}_p y_f(t-p)]$,
 - c) $u_f(t) = u(t) - [\gamma(t)\hat{c}_1 u_f(t-1) + \dots + \gamma^p(t)\hat{c}_p u_f(t-p)]$,
 - d) $\hat{y}_f(t) = \varphi_f^T(t) \hat{\theta}_1(t)$,
 - e) $\varepsilon_f(t) = y_f(t) - \hat{y}_f(t)$,
 - f) $P(t) = P(t-1) - \frac{P(t-1) \cdot \varphi_f(t) \cdot \varphi_f^T(t) \cdot P(t-1)}{1/\alpha_t + \varphi_f^T(t) \cdot P(t-1) \cdot \varphi_f(t)}$,
 - g) $\hat{\theta}_1(t+1) = \hat{\theta}_1(t) + \alpha_t P(t) \cdot \varphi_f(t) \cdot \varepsilon_f(t)$,
 - h) $\varphi_{f_i}(t+1) = \varphi_{f_{(i-1)}}(t)$ $\forall i$ sauf $\varphi_{f_1}(t+1) = y_f(t)$, $\varphi_{f_{(n+1)}}(t+1) = u_f(t)$, $\varphi_{f_{(n+m+1)}}(t+1) = \varepsilon_f(t)$.

Comme nous l'avons fait pour le MCR, nous pouvons introduire l'algorithme précédent dans un bloc S-function et simuler son fonctionnement par Simulink (fig. 6-6).

Cet algorithme est appliqué à un système de transmittance

$$G(z) = \frac{0z^{-1} + 0.2553z^{-2} - 0.06266z^{-3}}{1 - 1.414z^{-1} + 0.6065z^{-2}}$$

dont la sortie est perturbée par la réponse à un bruit blanc de la fonction de transfert

$$G_b(z) = \frac{1 - 0.312z^{-1} + 0.46z^{-2}}{1 - 1.414z^{-1} + 0.6065z^{-2}}.$$

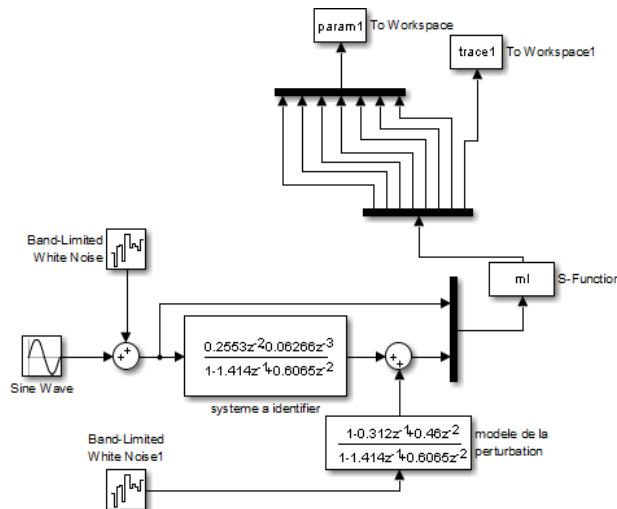
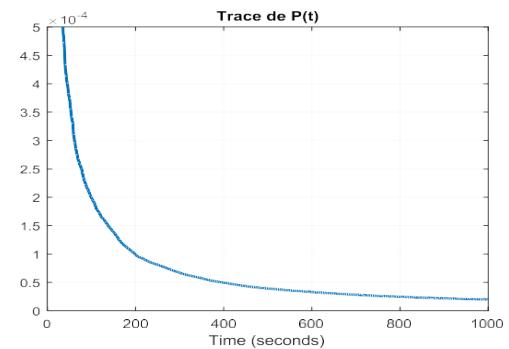
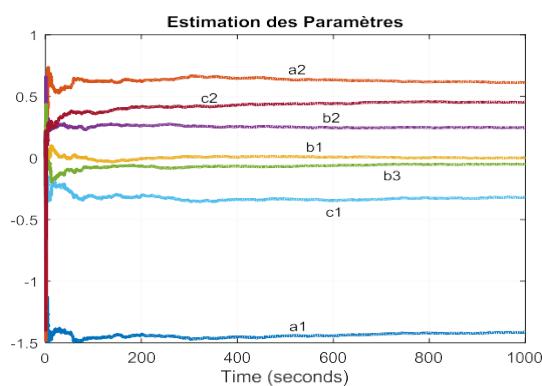


Fig. 6-6 Simulateur de l'algorithme ML

Les paramètres de G et G_b sont supposés inconnus et les figures suivantes montrent leur estimation récursive par l'algorithme ML.



Matlab fournit les estimations des paramètres suivants obtenus après 1000 secondes :

$$\begin{aligned} a_1 &= -1.4148, a_2 = 0.6114, \\ b_1 &= 0.0024, b_2 = 0.2459, b_3 = -0.0513 \\ c_1 &= -0.3187, c_2 = 0.4531. \end{aligned}$$

La trace de $P(t)$ qui mesure la moyenne de l'erreur quadratique est presque nulle.

Pour faciliter notre tâche, le «Toolbox Identification» de Matlab contient différents algorithmes d'identification pour différents modèles. Taper «`iddemo`» ou lire l'aide relative à cette boîte à outils pour connaître les syntaxes de ces algorithmes et apprendre à les utiliser.

Le PREM de Matlab est applicable au modèle le plus général,

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

et à ses cas particuliers :

BOX-JENKENS : $A(q^{-1}) = 1$ ($d^0 A = 0$),
 ARMAX : $D(q^{-1}) = F(q^{-1}) = 1$,
 ARX : $D(q^{-1}) = F(q^{-1}) = C(q^{-1}) = 1$,
 OE : $D(q^{-1}) = F(q^{-1}) = 1$, $C(q^{-1}) = A(q^{-1})$ (OE : output error).

Donnons des exemples.

Exemple 6-1

Soit $G(z) = \frac{Zy}{Zu} = \frac{2z^{-1} + 0.5z^{-2}}{1 - 1.2z^{-1} + 0.6z^{-2}}$

la transmittance du système à identifier dont la sortie y est perturbée par la réponse de la fonction de transfert

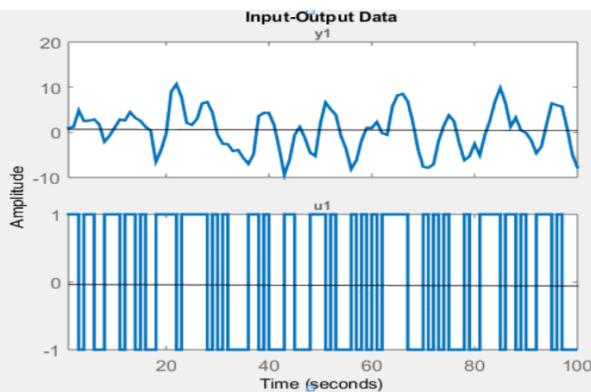
$$G_b(z) = \frac{1 - z^{-1} + 0.4z^{-2}}{1 - 1.2z^{-1} + 0.6z^{-2}}.$$

au signal $e(t) = wgn(100,1,1)$ constitué d'une colonne de 100 valeurs d'un bruit blanc gaussien de puissance $\sigma^2 = 1$.

Par l'algorithme RPEM appliqué à un échantillon de 100 mesures expérimentales de l'entrée u et de la sortie y , on estime les paramètres de G et G_b supposés inconnus. Dans cet exemple, faute de données expérimentales, on effectue une simulation de la réponse y à une suite d'impulsions égales aux signes de valeurs aléatoires ayant une distribution normale : $u = \text{sign}(\text{randn}(100,1))$.

Simulation

```
>> A = [1 -1.2 0.6]; B = [0 2 0.5]; C = [1 -1 0.4];
>> u = sign(randn(100,1)); e = wgn(100,1,1);
>> th0 = idpoly(A,B,C);
>> y = sim(th0,[u,e]);
>> z = iddata(y,u);
>> plot(z)
```



Pour représenter la convergence de l'estimation $\hat{\theta}$ vers le vecteur des paramètres θ , on commence par obtenir les estimations de θ , y , P , ϕ et ψ à l'instant $t = 1$ pour déduire récursivement ensuite ces estimations aux instants $t = 2, 3, \dots$

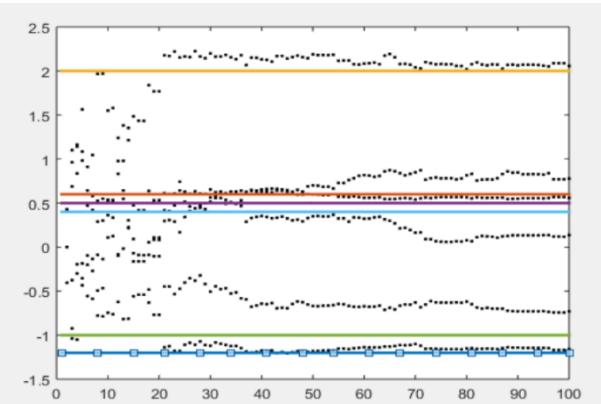
Dans les instructions suivantes, les composantes du vecteur $[2 2 0 0 1]$ sont respectivement les degrés des polynômes A , B , C , D , F et du retard d . D'autre

part, l'écriture «'ff',0.98» signifie que le facteur d'oubli est $\lambda = 0.98$. La dernière ligne représente les droites horizontales dont les ordonnées sont les vrais paramètres. Elles sont introduites pour montrer la convergence des estimations vers ces paramètres

Estimation récursive

```
>> [th,yp,p,phi,psi] ...
= rpem(z(1,:),[2 2 0 0 1],'ff',0.98);
>> for k = 2:100
[th,yp,p,phi,psi] ...
= rpem(z(k,:),[2 2 0 0 1], 'ff',0.98,th',p,phi,psi);
>> plot(k,th,'.')
>> hold on
>> end
>> plot(ones(100,1)*[A(2:3) B(2:3) C(2:3)])
```

On obtient le même résultat par l'algorithme RARMAX au lieu de RPEM mais dans ce cas les degrés des polynômes n , m , p et le retard d se définissent par $[2 2 2 1]$ au lieu de $[2 2 2 0 0 1]$.



En employant les fonctions « pem » ou « armax » (sans la première lettre r) Matlab fournit directement le modèle sans passer par une représentation graphique.

```
>> marmx = armax(z,[2 2 2 1])
```

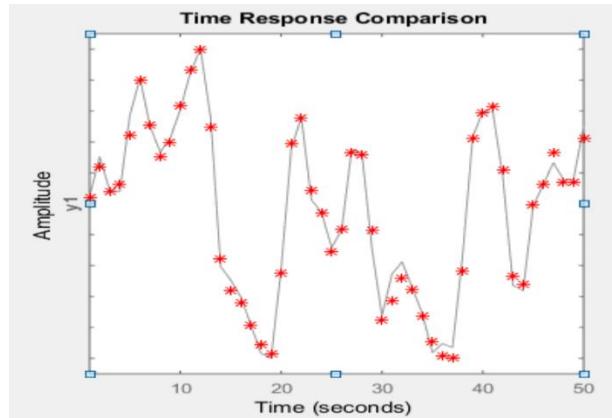
marmx =

Discrete-time ARMAX model:

$$\begin{aligned} A(z)y(t) &= B(z)u(t) + C(z)e(t) \\ A(z) &= 1 - 1.165 z^{-1} + 0.5832 z^{-2} \\ B(z) &= 1.95 z^{-1} + 0.6293 z^{-2} \\ C(z) &= 1 - 1.226 z^{-1} + 0.7169 z^{-2} \\ \text{FPE: } &1.15 \text{ (final prediction error).} \end{aligned}$$

Pour s'assurer du modèle obtenu, on peut appliquer de nouvelles entrées u et e au système réel et au modèle et comparer leurs réponses.

```
>> ut = sign(randn(50,1)); et = 0.4*randn(50,1);
>> yt = sim(th0,[ut,et]);
>> zt = iddata(yt,ut);
>> compare(zt,marmx,'r*')
```

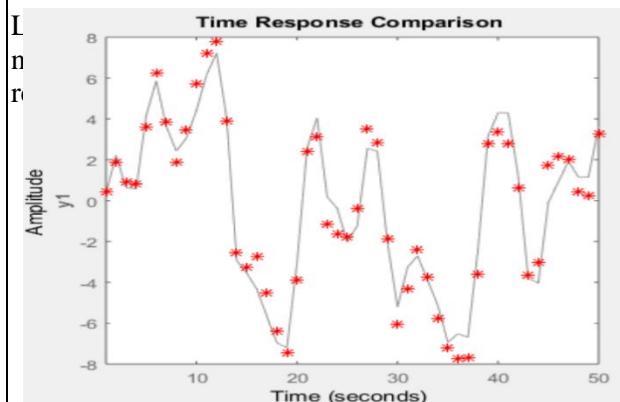


Matlab indique que les réponses du système réel et du modèle marmx aux nouvelles entrées u_t et e_t coïncident en 87.57 % des points.

Voyons si un modèle ARX, représente aussi bien le système.

```
>> marx = arx(z,[2 2 1])
marx =
Discrete-time ARX model:
A(z)y(t) = B(z)u(t) + e(t)
A(z) = 1 - 0.9466 z^-1 + 0.3911 z^-2
B(z) = 1.919 z^-1 + 1.259 z^-2
FPE: 2.26
```

```
>> compare(zt, marx,'r*')
```



6-6 METHODE DE LA VARIABLE INSTRUMENTALE (IV)

Cette méthode consiste à remplacer le vecteur d'observation φ par un autre ψ non corrélé avec la perturbation v c.à.d.

$$E[\alpha_t \psi(t).v(t)] = 0.$$

Elle ne modélise pas la perturbation mais seulement la fonction de transfert. Il s'agit donc d'estimer les paramètres du modèle :

$$\begin{aligned} y(t) &= -a_1 y(t-1) - \cdots - a_n y(t-n) \\ &\quad + b_0 u(t) + b_1 u(t-1) + \cdots + b_m u(t-m) \\ &\quad + v(t) \\ &= \theta^T \cdot \varphi(t) + v(t) \end{aligned}$$

où $v(t)$ n'est pas nécessairement blanc.

Comme pour l'algorithme MCR, on estime le vecteur des paramètres à l'instant $t+1$ en minimisant la fonction

$$J_t = \sum_{i=1}^t \alpha_i [y(i) - \theta^T \cdot \varphi(i)]^2$$

c.à.d. en résolvant l'équation

$$\frac{\delta J_t}{\delta \theta} = -2 \sum_{i=1}^t \alpha_i [y(i) - \theta^T \cdot \varphi(i)] \frac{\delta [\theta^T \cdot \varphi(i)]}{\delta \theta}.$$

Le terme $\delta [\theta^T \cdot \varphi(i)] / \delta \theta$ qui n'est autre que

$\varphi(i) = [-y(i-1), \dots, -y(i-n), u(i), \dots, u(i-m)]^T$. sera remplacé dans la méthode de la variable instrumentale par

$$\psi(i) = [-y_M(i-1), \dots, -y_M(i-n), u(i-1), \dots, u(i-m)]^T.$$

où y_M est la réponse d'un modèle ayant la même entrée u que le système réel. La sortie y_M de ce modèle, électronique ou numérique, n'est pas affectée par la perturbation v (fig. 6-7).

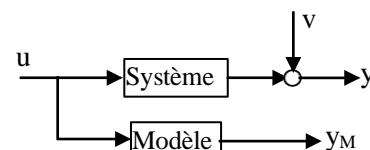


Fig. 6-7 Représentation de y et y_M

Avec cette substitution, on estime θ à l'instant $t + 1$ en résolvant

$$\sum_{i=1}^t \alpha_i [y(i) - \theta^T \cdot \varphi(i)] \psi(i) = 0.$$

Cette estimation de θ n'est valable que si ψ est voisin de φ c.à.d. que si y_M est voisin de y . Sous cette condition, la solution de l'équation précédente est

$$\begin{aligned} \hat{\theta}(t+1) &= \left[\sum_{i=1}^t \alpha_i \cdot \psi(i) \cdot \varphi^T(i) \right]^{-1} \sum_{i=1}^t \alpha_i y(i) \psi(i), \\ &= \left[\sum_{i=1}^t \alpha_i \cdot \psi(i) \cdot \varphi^T(i) \right]^{-1} \sum_{i=1}^t \alpha_i [\theta^T \varphi(i) + v(i)] \psi(i) \\ &= \theta + t \left[\sum_{i=1}^t \alpha_i \cdot \psi(i) \cdot \varphi^T(i) \right]^{-1} \left[\frac{1}{t} \sum_{i=1}^t \alpha_i v(i) \right] \psi(i). \end{aligned}$$

Comme la sortie du modèle y_M est indépendante de v , il en est de même de ψ et on a :

$$\lim_{t \rightarrow \infty} \frac{1}{t} \sum_{i=1}^t \alpha_i v(i) \psi(i) = E[\alpha_i v(t)] \psi(t) = 0$$

d'où, en posant $P(t) = \left[\sum_{i=1}^t \alpha_i \cdot \psi(i) \cdot \varphi^T(i) \right]^{-1}$

$\theta(t+1) \rightarrow \theta$ si la matrice $[t.P(t)]$ reste finie.

Par les mêmes arrangements algébriques que pour l'algorithme MCR, on arrive aux relations récursives suivantes :

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \alpha_t P(t) \psi(t) \varepsilon(t),$$

avec $\varepsilon(t) = y(t) - \varphi^T(t) \hat{\theta}(t)$

$$\begin{aligned} P(t) &= P(t-1) - \frac{P(t-1) \psi(t) \varphi^T(t) P(t-1)}{1/\alpha_t + \varphi(t) P(t-1) \psi(t)} \quad (6-72) \\ &= P(t-1) - \Delta, \end{aligned}$$

$$\Delta = \frac{P(t-1) \psi(t) \varphi^T(t) P(t-1)}{1/\alpha_t + \varphi(t) P(t-1) \psi(t)}. \quad (6-73)$$

Or, l'équation du modèle est

$$\begin{aligned} y_M(t) &= -\hat{a}_1 y_M(t-1) - \cdots - \hat{a}_n y_M(t-n) \\ &\quad + \hat{b}_0 u(t) + \hat{b}_1 u(t-1) + \cdots + \hat{b}_m u(t-m) \\ &= \hat{\theta}^T(t) \psi(t). \end{aligned}$$

Durant les premières périodes où $\hat{\theta}(t)$ est encore loin du vrai vecteur des paramètres θ , le modèle produit des valeurs pour y_M qui ne sont pas voisins des valeurs de y d'où ψ ne sera pas voisin de φ . Dans ce cas, la trace de la matrice Δ définie en (6-73) peut être négative d'où $\text{tr}[P(t)]$ qui mesure l'erreur d'estimation ε peut croître, sa convergence vers zéro ne sera pas garantie et l'algorithme pourra produire un biais ou même diverger. Pour remédier à cette difficulté, on commence par une étape d'initialisation en appliquant l'algorithme MCR sur un nombre N de périodes c.à.d. en supposant que durant ces périodes l'erreur due à la perturbation est négligeable devant l'erreur d'estimation. On prend généralement $N \approx 8.(n_A + n_B)$, soit 8 fois le nombre des paramètres à estimer. Au bout de ces périodes, le vecteur $\hat{\theta}(t)$ devient assez voisin de θ et presque indépendant des perturbations ce qui permet de passer aux relations (6-71) et (6-72). L'algorithme de la variable instrumentale IV se résume donc comme suit.

(IV)

La matrice $P(t-1)$ et les vecteurs $\varphi(t)$, $\psi(t)$ et $\hat{\theta}(t)$ obtenus durant l'étape t , sont stockés dans la mémoire

- 1) A l'instant t , on lit les valeurs de $y(t)$ et de $u(t)$.
- 2) Durant l'étape $t + 1$, c'est-à-dire entre t et $t + 1$, on calcule successivement :

a) $\hat{y}(t) = \hat{\theta}^T(t) \cdot \varphi(t)$,

b) $y_M(t) = \hat{\theta}^T(t) \psi(t)$,

c) $\varepsilon(t) = y(t) - \hat{y}(t)$,

d) $P(t) = P(t-1) - \frac{P(t-1) \psi(t) \varphi^T(t) P(t-1)}{1/\alpha_t + \varphi(t) P(t-1) \psi(t)}$

e) $\hat{\theta}(t+1) = \hat{\theta}(t) + \alpha_t P(t) \psi(t) \varepsilon(t)$,

f) $\varphi^T(t+1) = [-y(t), \varphi_1(t), \dots, \varphi_{(n-1)}(t), u(t), \varphi_{(n+1)}(t), \dots, \varphi_{(n+m-1)}(t)]$.

g) Si $t \leq N$, $\psi(t) = \varphi(t)$.

Si $t > N$,

$\psi^T(t+1) = [-y_M(t), \psi_1(t), \dots, \psi_{(n-1)}(t), u(t), \psi_{(n+1)}(t), \dots, \psi_{(n+m-1)}(t)]$.

L'implantation de cet algorithme dans un bloc S-function de Simulink est semblable à celle du MCR. La fonction «iv4» de Matlab applique une variante de cet algorithme

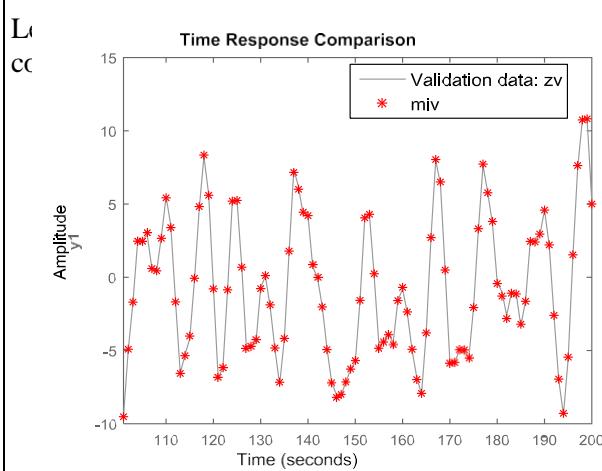
Exemple 6-2

On effectue 200 mesures de l'entrée u et de la sortie y d'un système $m_0 = A/B$ perturbé par $v = (C/A)e$. Les 100 premières mesures z_e sont employées pour identifier le système m_0 par un modèle miv et, afin de valider ce modèle, les 100 dernières mesures $z_v = (u_v, y_v)$ sont employées pour comparer y_v à la réponse de miv à u_v

```
>> A = [1 -1.2 0.6]; B = [0 2 0.5]; C = [1 -1 0.4];
>> m0 = idpoly(A,B,C);
>> u = sign(randn(200,1)); e = 0.1*randn(200,1);
>> y = sim(m0,[u,e]);
>> z = iddata(y,u);
>> ze = z(1:100); zv = z(101:200);
>> miv = iv4(ze,[2 2 1])
```

```
min =
Discrete-time ARX model :
A(z)y(t) = B(z)u(t) + e(t)
A(z) = 1 - 1.204 z^-1 + 0.6037 z^-2
B(z) = 2.011 z^-1 + 0.4793 z^-2
FPE: 0.02676
```

```
>> compare(zv,miv,'r*')
```



La coïncidence des réponses indiquée par Matlab est de 97.94 % ce qui est presque parfait.

Identification d'une matrice de transfert.

Un système linéaire multivariable perturbé se définit par la relation

$$y(t) = G(q^{-1})u(t) + v(t) \quad (6-74)$$

où $u \in \mathbb{R}^p$, $y \in \mathbb{R}^q$, $v \in \mathbb{R}^q$ et $G(q^{-1})$ est une matrice dont l'élément $G_{ij}(q^{-1})$ est la fonction de transfert entre la $j^{\text{ème}}$ entrée et la $i^{\text{ème}}$ sortie c.à.d

$$y_i(t) = G_{ij}(q^{-1})u_j(t) + v_i(t).$$

Par superposition, l'effet de toutes les entrées u_j sur la sortie y_j est

$$y_i(t) = \sum_{j=1}^p G_{ij}(q^{-1})u_j(t) + v_i(t). \quad (6-75)$$

En désignant par $A_i(q^{-1})$ le dénominateur commun des $G_{ij}(q^{-1})$ et par $B_{ij}(q^{-1})$ les numérateurs correspondants, l'équation (6-75) s'écrit sous la forme

$$A_i(q^{-1})y_i(t) = \sum_{j=1}^p B_{ij}(q^{-1})u_j(t) + w_i(t)$$

$$\text{ou } A_i(q^{-1})y_i(t) = B_i^T(q^{-1})u(t) + w_i(t) \quad (6-76)$$

avec

$$B_i^T(q^{-1}) = [B_{i1}(q^{-1}) \ B_{i2}(q^{-1}) \ \dots \ B_{ip}(q^{-1})]$$

$$\text{et } w_i(t) = A_i(q^{-1})v_i(t).$$

La relation (6-76) a la même forme que dans le cas monovariable à part que $B_i(q^{-1})$ et $u(t)$ sont maintenant des vecteurs à p composantes au lieu d'une. Pour cette raison, avec de légères modifications, les algorithmes précédents s'étendent aux systèmes multivariables.

Exemple 6-3

Soit un système à deux entrées et une sortie ayant pour fonction de transfert

$$G(q^{-1}) = \begin{pmatrix} \frac{q^{-1}(0.5 + 0.3q^{-2})}{1 - 1.5q^{-1} + 0.7q^{-2}} & \frac{q^{-2}}{1 - 1.5q^{-1} + 0.7q^{-2}} \end{pmatrix},$$

perturbé par la réponse $v(t)$ de

$$G_b(q^{-1}) = \frac{1 - q^{-1} + 0.5q^{-2}}{1 - 1.5q^{-1} + 0.7q^{-2}}.$$

à $e(t) = 0.1 * \text{randn}(200,1)$.

Faute de données expérimentales, on commence par simuler la réponse y du système en lui appliquant les entrées

$$u_1 = \text{randn}(200,1) \quad \text{et} \quad u_2 = 2 * \text{randn}(200,1).$$

Les 100 premières lectures z_e de l'entrée $u = [u_1 \ u_2]$ et de la sortie y seront employées pour l'identification du système et la modélisation de la perturbation et les 100 dernières lectures z_v seront employées pour la validation de ces résultats.

Simulation

```
>> A = [1 -1.5 0.7]; B = [0 0.5 0 0.3; 0 0 1 0];
>> C = [1 -1 0.5];
>> m0 = idpoly(A, B, C);
>> u1 = sign(randn(200,1));
>> u2 = 2*sign(randn(200,1));
>> e = 0.1*randn(200,1);
>> y = sim(m0,[[u1,u2],e]);
>> z = iddata(y,[u1,u2],0.5); % T = 0.5 sec.
>> ze = z(1:100); zv = z(101:200);
```

Identification

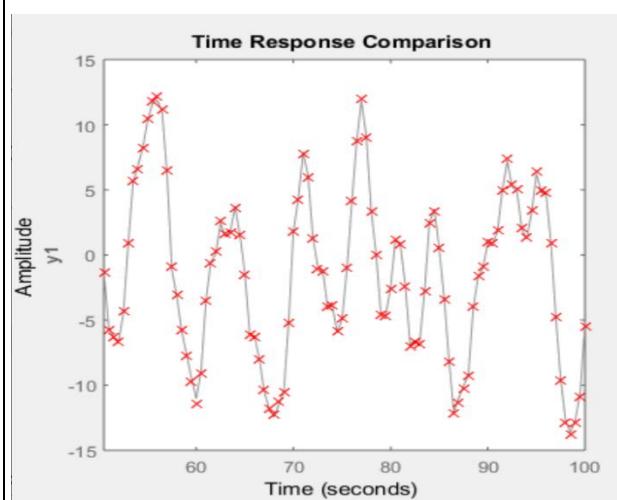
```
>> ml = armax(ze,'na',2,'nb',[3 3],'nc',2,'nk',[1 2])
(nk : retards de y par rapport à u1 et u2)
```

Discrete-time ARMAX model :

$$\begin{aligned} A(z)y(t) &= B(z)u(t) + C(z)e(t) \\ A(z) &= 1 - 1.502 z^{-1} + 0.7017 z^{-2} \\ B(z) &= 0.4593 z^{-1} + 0.02447 z^{-2} + 0.3011 z^{-3} \\ B2(z) &= 0.9905 z^{-2} + 0.003379 z^{-3} \\ C(z) &= 1 - 0.9824 z^{-1} + 0.3259 z^{-2} \\ \text{FPE: } &0.01222 \end{aligned}$$

Validation

```
>> compare(zv,ml,'rx')
```



Coïncidence : 97.55 %

Une certaine erreur peut apparaître au départ car le modèle part d'un état initial nul tandis que z_v ne l'est pas en cet instant.

6-7 METHODE DU SOUS-ESPACE

Nous commençons par généraliser le problème du moindre carré au cas multivariable puis nous l'appliquons aux équations d'état d'un système linéaire invariant ayant plusieurs entrées et plusieurs sorties.

Généralisation du problème de moindre carré. Considérons l'équation linéaire

$$y(t) = \Theta x(t) + v(t) \quad (6-77)$$

où Θ est une matrice constante $y(t)$ et $x(t)$ sont des vecteurs connus et $v(t)$ est un vecteur de moyenne nulle indépendant de $x(t)$. Il s'agit d'estimer la matrice Θ à partir des valeurs de x et de y mesurés aux instants $t = 1, 2, \dots, N$ où N est un entier suffisamment grand. En d'autres termes, il s'agit de déterminer la matrice Θ qui vérifie le mieux la relation matricielle

$$Y = \Theta X + V \quad (6-78)$$

où les matrices Y , X et V sont respectivement constituées des vecteurs colonnes $[x(1); \dots; x(N)]$, $[y(1); \dots; y(N)]$ et $[v(1); \dots; v(N)]$. L'estimation $\hat{\Theta}$ de Θ doit minimiser $J = \| (Y - \Theta X - V) X^T z \|^2$ pour tout vecteur $z \in \mathbb{R}^N$, c.à.d. elle doit vérifier

$$\begin{aligned} \frac{d}{d\Theta} [z^T X (Y - \Theta X - V)^T (Y - \Theta X - V) X^T z] \\ = -2z^T X X^T (Y - \Theta X - V) X^T z = 0, \quad \forall z. \end{aligned}$$

En admettant que la matrice XX^T est inversible, on déduit que Θ doit vérifier

$$(Y - \Theta X - V) X^T = 0.$$

Comme, pour tout t , $v(t)$ est de moyenne nulle et indépendant de $x(t)$, la matrice $E(vx^T) \approx VX^T/N \rightarrow 0$ quand $N \rightarrow \infty$. D'où l'estimation de Θ est

$$\hat{\Theta} = YX^T(XX^T)^{-1} \quad (6-79)$$

et celle de Y est

$$\hat{Y} = \hat{\Theta}X = Y\Pi_X \quad (6-80)$$

$$\text{où } \Pi_X = X^T(XX^T)^{-1}X. \quad (6-81)$$

L'équation (6-80) montre que chaque ligne de \hat{Y} est une combinaison linéaire des lignes de X c.à.d. que l'espace engendré par les lignes de \hat{Y} est inclus dans l'espace E_X engendré par les lignes de X. D'autre part, en remarquant que $\Pi_X^T\Pi_X = \Pi_X$, on déduit que

$$\hat{Y}\Pi_X = \hat{Y} \quad \text{et} \quad (Y - \hat{Y})\Pi_X = 0. \quad (6-82)$$

Les égalités (6-80) et (6-82) montrent que Π_X est la matrice de projection orthogonale sur l'espace E_X et que $\Pi_X^\perp = I - \Pi_X$ est la matrice de projection sur l'espace complémentaire de E_X comme le schématise la figure 6-8.

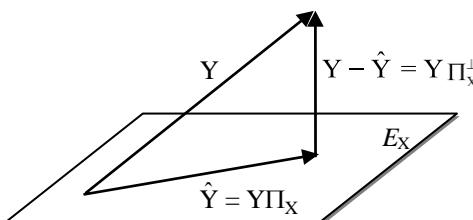


Fig. 6-8 Projections sur E_X et sur son complémentaire.

Enfin, l'équation suivante représente la relation supposée linéaire entre les entrées, les sorties et l'état mesurés durant N périodes.

$$Y = \Gamma X + HU + V. \quad (6-83)$$

Il s'agit d'estimer les matrices des paramètres Γ et H en présence des perturbations affectant les sorties et groupées dans la matrice V .

Cette estimation se ramène au problème de moindre carré précédent (celui de (6-78)) en posant

$$\Theta' = [\Gamma \ H] \quad \text{et} \quad X' = \begin{bmatrix} X \\ U \end{bmatrix}.$$

Identification des équations d'état.

a) Formulation. Les équations d'état d'un système linéaire invariant perturbé sont de la forme :

$$x(t+1) = Ax(t) + Bu(t) + Kv(t)$$

$$y(t) = Cx(t) + Du(t) + v(t), \quad (6-84)$$

$$x(t) \in \mathcal{R}^n, \quad u(t) \in \mathcal{R}^p, \quad y(t) \in \mathcal{R}^q$$

Les vecteurs Kv et v sont les effets des perturbations aléatoires sur l'état et la sortie, K étant la matrice de Kalman.. On admet que la perturbation v est indépendante de l'entrée u .

La sortie à l'instant $t + k$ dépend de l'état à l'instant t , des entrées entre t et $t + k$ et de la perturbation à l'instant $t + k$. Pour obtenir cette relation, remplaçons $x(i)$ par son expression pour $i = t + k, \dots, t$.

$$\begin{aligned} y(t+k) &= CAx(t+k-1) \\ &\quad + CBu(t+k-1) + Du(t+k) \\ &\quad + CKv(t+k-1) + v(t+k) \\ &= CA^2x(t+k-2) \\ &\quad + CABu(t+k-2) + CBu(t+k-1) + Du(t+k) \\ &\quad + CAKv(t+k-2) + CKv(t+k-1) + v(t+k) \\ &\quad \vdots \end{aligned}$$

Donc

$$\begin{aligned} y(t+k) &= CA^kx(t) \\ &\quad + \sum_{i=1}^k CA^{k-i}Bu(t+i-1) + Du(t+k) \quad (6-85) \\ &\quad + \sum_{i=1}^k CA^{k-i}Kv(t+i-1) + v(t+k). \end{aligned}$$

Pour $k = 0, \dots, r-1$, l'équation matricielle suivante regroupe les r équations (6-85) :

$$Y_r(t) = \Gamma_r x(t) + H_r U_r(t) + S_r(t) V_r(t) \quad (6-86)$$

où

$$Y_r(t) = [y(t) \ y(t+1) \ \dots \ y(t+r-1)]^T,$$

$$U_r(t) = [u(t) \ u(t+1) \ \dots \ u(t+r-1)]^T,$$

$$V_r(t) = [v(t) \ v(t+1) \ \dots \ v(t+r-1)]^T.$$

et

$$\Gamma_r = [C \ CA \ \dots \ CA^{r-1}]^T, \quad (6-87)$$

$$H_r = \begin{bmatrix} D & 0 & \cdots & 0 & 0 \\ CB & D & \cdots & 0 & 0 \\ CAB & CB & \cdots & 0 & \vdots \\ \vdots & \vdots & \ddots & D & 0 \\ CA^{r-2}B & CA^{r-3}B & \cdots & CB & D \end{bmatrix}, \quad (6-88)$$

$$S_r = \begin{bmatrix} I & 0 & \cdots & 0 & 0 \\ CK & I & \cdots & \vdots & \vdots \\ CAK & CK & \cdots & 0 & \vdots \\ \vdots & \vdots & \ddots & I & 0 \\ CA^{r-2}K & CA^{r-3}K & \cdots & CK & I \end{bmatrix}. \quad (6-89)$$

Supposons que l'entrée et les perturbations s'annulent à partir de l'instant t . Dans ce cas, l'équation (6-86) se réduit à $Y_r(t) = \Gamma_r x(t)$. Si le rang de Γ_r est égal à l'ordre n du système (au nombre des variables d'état), on peut déterminer d'une manière unique l'état $x(t)$ à l'instant t en mesurant la sortie y aux instants $t, t+1, \dots, t+r-1$ et on dit que l'état du système est observable de la sortie.

Les matrices H_r et S_r sont dites de Toeplitz ; elles sont triangulaires et les sous-matrices des sous-diagonales sont égales.

Groupons les matrices $Y_r(t), \dots, Y_r(t+N-1)$ l'une à côté de la précédente pour constituer la matrice suivante dont la forme est dite block de Hankel :

$$Y = [Y_r(t) \ Y_r(t+1) \ \cdots \ Y_r(t+N-1)] \\ = \begin{bmatrix} y(t) & y(t+1) & \cdots & y(t+N-1) \\ y(t+1) & y(t+2) & \cdots & y(t+N) \\ \vdots & \vdots & \ddots & \vdots \\ y(t+r-1) & y(t+r) & \cdots & y(t+r+N-2) \end{bmatrix}.$$

De la même manière, on construit les matrices

$$U = [U_r(t) \ U_r(t+1) \ \cdots \ U_r(t+N-1)], \\ V = [V_r(t) \ V_r(t+1) \ \cdots \ V_r(t+N-1)].$$

Avec ces notations, les équations (6-86) relatives aux instants $t, t+1, \dots, t+N-1$ peuvent se grouper pour former l'équation

$$Y = \Gamma_r X + H_r U + S_r V \quad (6-90)$$

où $X = [x(t) \ x(t+1) \ \cdots \ x(t+N-1)]$.

Généralement, la matrice X n'est pas complètement accessible à la mesure mais, en fonction de données passées

$U_p = [u(t-1), u(t-2), \dots]$ et $Y_p = [y(t-1), y(t-2), \dots]$,

elle peut être estimée par un filtre de Kalman qui produit une matrice L telle que

$$X = LZ \quad \text{avec} \quad Z = [U_p \ Y_p]^T.$$

Enfin, l'équation (6-90) devient :

$$Y = \Gamma_r LZ + H_r U + S_r V. \quad (6-91)$$

Pour estimer A et C à partir de (6-87), on commence par estimer Γ_r en éliminant le terme en U . Ayant A, C et LZ on estime H_r et par suite B et D par moindre carré.

Estimation des matrices d'état. Pour éliminer le terme en U de l'équation (6-91), il suffit de la projeter sur le complémentaire de l'espace E_U engendré par les lignes de U . On obtient :

$$V\Pi_U^\perp = \Gamma_r LZ\Pi_U^\perp + S_r V\Pi_U^\perp. \quad (6-92)$$

Or, comme les perturbations V sont indépendantes des entrées U , on a en moyenne $VU^T = 0$ d'où, tenant compte de (6-81),

$$V\Pi_U^\perp = V(I - \Pi_U) = V[I - U^T(UU^T)^{-1}U] = V.$$

D'autre part, comme les perturbations actuelles et futures sont indépendantes des données passées Z , on a $VZ^T = 0$ d'où, en remplaçant dans (6-92) $V\Pi_U^\perp$ par V et en la multipliant par Z^T , elle devient :

$$Y\Pi_U^\perp Z^T = \Gamma_r LZ\Pi_U^\perp Z^T. \quad (6-93)$$

La solution de cette équation qui comporte deux matrices inconnues, Γ_r et L , n'est pas unique. Mais nous démontrons au chapitre 3 de la quatrième partie de ces notes qu'une matrice peut être remplacée par un produit de 3 matrices, F , Σ et G^T , Σ étant diagonale à éléments positifs et les deux autres unitaires ($FF^T = GG^T = I$). Cette décomposition est dite à valeurs singulières (SVD). En l'appliquant au premier membre, l'équation (6-93) devient :

$$F\Sigma G^T = \Gamma_r L Z \Pi_U^\perp Z^T.$$

Une solution équilibrée de cette équation consiste à choisir

$$\Gamma_r = F\Sigma^{1/2} \quad \text{et} \quad L = \Sigma^{1/2} G^T (Z \Pi_U^\perp Z^T)^{-1}. \quad (6-94)$$

Ayant $\Gamma_r = [C \ C A \ \dots \ C A^{r-1}]^T$, la matrice C est ses q premières lignes. Les autres lignes forment la matrice

$$\Gamma = \Gamma_{r-1} A \quad \text{ou} \quad \Gamma^T = A^T \Gamma_{r-1}^T, \quad (6-95)$$

D'après (6-79), la matrice A qui vérifie le mieux l'équation précédente selon le critère du moindre carré est

$$A = (\Gamma_{r-1} \Gamma_{r-1}^T)^{-1} \Gamma_{r-1} \Gamma. \quad (6-96)$$

Il reste à estimer B et D. À part le terme $S_r V$ des perturbations, toutes les matrices de l'équation (6-91) sont maintenant connues sauf H_r qu'on estime par moindre carré en appliquant (6-79), ce qui donne

$$H_r = (Y - \Gamma_r X) U^T (U U^T)^{-1}. \quad (6-97)$$

En se référant à l'expression (6-88) de H_r , on estime D par l'une des sous-matrices de la diagonale et, connaissant C, on estime B en appliquant le moindre carré à $M = CB$, M étant une sous-matrice de la sous-diagonale de H_r c.à.d.

$$B = (C C^T)^{-1} C M. \quad (6-98)$$

Matlab exécute cet algorithme par l'instruction «n4sid(données, ordre)».

Exemple 6-4

On considère un système à 2 entrées et 2 sorties défini par les équations d'état

$$\begin{aligned} \dot{x} &= \begin{pmatrix} -0.1 & 0 & 0.2 \\ 1 & -0.2 & 0 \\ 0 & 1 & -0.5 \end{pmatrix} x + \begin{pmatrix} 1 & 0 \\ -0.5 & 2 \\ 0 & -1 \end{pmatrix} u \\ y &= \begin{pmatrix} 2 & 0 & -1 \\ 0 & 1 & 0.5 \end{pmatrix} x. \end{aligned} \quad (6-99)$$

La sortie est perturbée par la matrice à deux colonnes identiques $v = 0.1 \text{randn}(1000,2)$ et son état par Kv où la matrice K est le gain de Kalman. Cette matrice ainsi que les matrices d'état A, B et C sont supposées inconnues et il s'agit de les estimer en mesurant l'entrée u et la sortie y aux instants $t = 0, 1, 2, \dots$.

Faute de données expérimentales, on commence par simuler la réponse y du système en lui appliquant les entrées

$$u_1 = \text{randn}(1000,1) \quad \text{et} \quad u_2 = 2 * \text{randn}(1000,1)$$

Les 500 premières données z_e de l'entrée $u = [u_1 \ u_2]$ et de la sortie $y = [y_1 \ y_2]$ seront employées pour l'identification du système et 50 des dernières données seront employées pour la validation de ces résultats.

```
>> A = [-0.1 0 0.2; 1 -0.2 0; 0 1 -0.5];
>> B = [1 0; -0.5 2; 0 -1];
>> C = [2 0 -1; 0 1 0.5]; D = [0 0; 0 0];
>> mo = idss(A, B, C, D);
>> u = randn(1000,1)*[1 2];
>> v = 0.1*randn(1000,2);
>> y = sim(mo,[u v]);
>> z = iddata(y,u);
>> ze = z(1:500); zv = z(600:650);
>> ml = n4sid(ze,3)
```

ml =

Discrete-time identified state-space model :
 $x(t+Ts) = A x(t) + B u(t) + K e(t)$
 $y(t) = C x(t) + D u(t) + e(t)$

A =

$$\begin{matrix} & x1 & x2 & x3 \\ x1 & -0.6443 & -0.3134 & -0.06781 \\ x2 & 0.7625 & -0.3178 & -0.1286 \\ x3 & -0.05338 & -0.8885 & 0.1652 \end{matrix}$$

B =

$$\begin{matrix} & u1 & u2 \\ x1 & 0.006073 & 0.01215 \\ x2 & 0.005549 & 0.0111 \\ x3 & -0.00379 & -0.007581 \end{matrix}$$

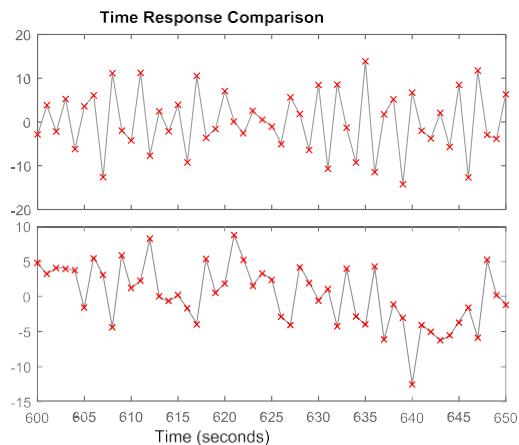
```
C =
  x1   x2   x3
y1 177.4 -54.96 -7.165
y2 -12.11  84.35 -27.76
```

```
K =
  y1      y2
x1 0.0001197 0.0001579
x2 -0.0001654 -0.0003445
x3 0.001006 0.0004549
```

Sample time : 1 seconds

Les matrices obtenues sont complètement différentes des matrices (6-99) par lesquelles nous avons défini le système. Ceci est normal car, comme nous l'avons indiqué dans les chapitres précédents, il existe une infinité d'équations d'état qui peuvent représenter un même système. La fonction « compare » permet de constater que l'identification précédente du système est presque parfaite.

```
>> compare(zv, ml,'rx')
```



Matlab indique que la coïncidence est de 98.65 % pour y_1 et de 97.72 % pour y_2