



Identification:un algorithme robuste et versatile

Alain Barraud, Suzanne Lesecq

► To cite this version:

Alain Barraud, Suzanne Lesecq. Identification:un algorithme robuste et versatile. Journal Européen des Systèmes Automatisés, 2006, 40 (6), pp.665-693. <hal-00121877>

HAL Id: hal-00121877

<https://hal.science/hal-00121877v1>

Submitted on 22 Dec 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Identification : un algorithme robuste et versatile

A. Barraud, S. Lesecq

Laboratoire d'Automatique de Grenoble
BP 46 38402 Saint Martin d'Hères, France
Alain.barraud@inpg.fr, suzanne.lesecq@inpg.fr

RÉSUMÉ. On présente une autre formulation de l'identification, en boucle ouverte, des modèles dynamiques linéaires stationnaires. Cette approche se différencie par une paramétrisation mixte des transferts consistant à prendre pour inconnues à estimer les pôles et non les coefficients du polynôme du dénominateur dont ils sont les racines. Cette solution présente des propriétés uniques en terme de précision sur la dynamique des modèles en présence de mesures bruitées. Le second point clé concerne la minimisation d'un critère « erreur de sortie » exploitant explicitement la propriété de « linéarité dans les paramètres » pour une partie des inconnues, réduisant ainsi considérablement la dimension de l'espace dans lequel on opère une optimisation non linéaire. Ces concepts s'appliquent aussi bien aux systèmes SISO que MIMO y compris pour les modèles d'états à paramètres structurés. L'identification d'un offset sur la sortie, des conditions initiales ou encore l'introduction de contraintes (transfert par transfert) telles la présence connue d'un intégrateur, du signe du gain statique, etc..., caractérisent la souplesse de la démarche tout en respectant le principe fondamental de « parcimonie ».

ABSTRACT. This paper deals with open loop linear time invariant linear systems identification. A new formulation is presented where the unknown dynamic parameters are directly the poles. This approach gives much better results in term of poles precision than computing poles from estimated polynomial coefficients as usually done. A second innovation is a systematic used of the "linear in the parameter" property of the other model parameters. An output error criterion is then minimized iteratively in a space of dimension equal to the system order. This criterion is itself the minimum of a quadratic problem coping with all the other model parameters. This approach works as well for siso, miso or mimo problems.

MOTS-CLÉS : Identification, hors ligne, méthode erreur de sortie, estimation directe des pôles, critère projeté, factorisation orthogonale, optimisation non linéaire..

KEYWORDS: Off line identification, output error method, direct poles estimation, projected criterion, orthogonal factorization, non linear optimization.

1. Introduction

Dans ce papier, on s'intéresse à l'identification hors ligne des procédés industriels. Dans ce contexte, on se limite aux modèles dynamiques linéaires stationnaires à temps discret. Par ailleurs, on suppose que l'acquisition des données se fait en boucle ouverte. On note $u_t \in \mathbb{R}^p$ les entrées et $y_t \in \mathbb{R}^q$ les sorties mesurées. Pour le cas des systèmes SISO (Single Input Single Output $p = q = 1$) et MISO (Multiple Inputs Single Output $p > 1, q = 1$) le modèle transfert en z est le seul identifié directement. Pour les systèmes MIMO (Multiple Inputs Multiple Outputs), l'alternative transfert – modèle d'états a été considérée. Ce cadre étant posé, il est clair qu'il existe d'innombrables publications répondant à ce sujet. Il n'est pas de notre propos d'en faire une synthèse qui serait nécessairement aussi inutile qu'incomplète. Citons simplement un grand classique du domaine (parmi d'autres), l'ouvrage de (Ljung, 1999), dans lequel on retrouve la plupart des fondamentaux en matière d'identification, référence que l'on pourra, par exemple, compléter par (Guidorzi, 1981) pour les modèles d'états canoniques et (Overschee, 1994) pour l'approche dite des « sous espace ». Ces dix dernières années ont bien sûr apporté leur lot de variantes ou améliorations mais ne contribue en rien vis-à-vis de l'argumentaire développé ici.

On peut diviser l'ensemble des algorithmes d'identification en deux catégories :

- ceux qui traitent de l'identification comme un problème de régression non linéaire, ce qui est la forme naturelle dans laquelle se présente habituellement la modélisation ;
- ceux qui imposent une structure d'erreur (dont une norme sera à minimiser) linéaire dans les paramètres à estimer.

Il est clair que ceux appartenant à la seconde catégorie sont, d'un point de vue conceptuel, beaucoup plus simple à traiter que les premiers (algorithmes du type moindres carrés, séquentiels ou non) mais souvent au prix de quelques inconvénients plus ou moins facilement contournables. Le biais fréquent de ces derniers, dans leurs versions premières (modèle ARX) est leur point faible historique. Mais ce qui nous paraît beaucoup plus contraignant, parce qu'incontournable, est leur sur paramétrage naturelle dans le cas MISO et a fortiori MIMO. Ils vont à l'encontre d'un principe premier reconnu de tous, celui de « parcimonie », et rendent difficile, voire impossible l'exploitation de connaissances a priori (en terme de propriétés dynamiques) transfert par transfert, autre pré requis d'une identification réussie. Ceci provient d'une mise au dénominateur commun forcée induisant fatalement des compensations de pôles et zéros.

Le but de cet article est de proposer une approche exploitant le meilleur de ces deux mondes (linéaire, non linéaire dans les paramètres) en introduisant deux idées clefs autour de la méthode « erreur de sortie » choisie pour son universalité :¹

– prendre pour paramètres de la dynamique directement les pôles du transfert et non les coefficients du polynôme dont ils sont les racines ;

– minimiser, par rapport à ces pôles, à l’aide d’une technique de programmation non linéaire appropriée, un critère du type erreur de sortie dont la valeur est déjà le minimum (explicite) par rapport aux inconnues « linéaires dans les paramètres » du problème d’identification (coefficients du numérateur, conditions initiales, offset sur la sortie, ...).

Cette approche possède des propriétés de robustesse et de versatilité qu’aucune autre méthode ne peut revendiquer.

2. La paramétrisation

Par simplicité, on prend pour contexte, dans ce paragraphe, le cas SISO. Soit donc le transfert $H(z)$ défini dans sa forme classique :

$$H(z) = z^{-k} \frac{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}} \quad [1]$$

où $k \geq 1$ est le retard exprimé en nombre de pas d’échantillonnage, $m \geq 0$ le nombre de zéros ou degré du numérateur et $n \geq 1$ le nombre de pôles ou degré du dénominateur. On notera par la suite :

$$\begin{cases} a = [a_1, a_2, \dots, a_n]^T \end{cases} \quad [2-a]$$

$$\begin{cases} b = [b_0, b_1, \dots, b_m]^T \end{cases} \quad [2-b]$$

Par opposition à ce paramétrage, on propose de remplacer les inconnues a par les paramètres :

$$\begin{cases} x_n = [r_1, \dots, r_{n_r}, (\rho_1, \theta_1), \dots, (\rho_{n_c}, \theta_{n_c})]; \\ 0 < r_k < 1, 0 < \rho_k < 1, 0 \leq \theta_k < \pi / 2 \\ n_r + 2n_c = n \end{cases} \quad [3]$$

où n_r est le nombre de pôles réels $z_k = r_k$ et n_c le nombre de paires de pôles complexes conjugués de la forme $z_k = \rho_k e^{\pm i\theta_k}$. Les $n + m + 1$ paramètres du

¹ S’applique à n’importe quel type de modèle dynamique, linéaire ou non linéaire, ou encore défini par des équations aux dérivées partielles.

transfert sont maintenant b et x_n . Au jeu de paramètres x_n est associé un polynôme $A(z)$ défini par :

$$A(z) = 1 + a_1 z^{-1} + \dots + a_n z^{-n} = \prod_{k=1}^n (1 - z_k z^{-1}) \quad [5]$$

De là, on peut établir la relation standard entrée – sortie, calculable à partir de x_n via [5]. On introduit à cet effet les notations suivantes en notant η_t la sortie du modèle à l'instant t :

$$\tilde{u}_t = [u_{t-k}, \dots, u_{t-k-m}]^T; \quad \tilde{\eta}_t = [\eta_{t-1}, \dots, \eta_{t-n}]^T; \quad [6]$$

Il vient alors :

$$\begin{cases} \eta_t = -\tilde{\eta}_t^T a + \tilde{u}_t^T b; & t = l+1, \dots, N \\ l = \max(k+m, n) \end{cases} \quad [7]$$

initialisé par $\tilde{\eta}_{l+1}$ les n conditions initiales à estimer également. N est le nombre de couples de données $\{u_t, y_t\}$ disponibles.

En conséquence, l'identification de $H(z)$ à partir de ces N données dépend de $2n + m + 1$ paramètres : b, x_n et $\tilde{\eta}_{l+1}$. On notera :

$$x_l = [b^T, \tilde{\eta}_{l+1}^T]^T \in \mathfrak{R}^{n+m+1} \quad [8]$$

l'ensemble des paramètres intervenant linéairement dans l'erreur de sortie. Cette modélisation se fera, ici, en minimisant l'erreur de sortie $\varepsilon_t = (\eta_t - y_t)$ via le critère :

$$f(x_n, x_l) = \sum_{t=l+1}^N (\eta_t - y_t)^2 = \sum_{t=l+1}^N \varepsilon_t^2 \quad [9]$$

Le deuxième point clef est d'utiliser le fait que ε_t est non linéaire seulement en x_n et linéaire en x_l , de sorte qu'au lieu de minimiser f [9] par une procédure itérative d'optimisation non linéaire dans un espace à $2n + m + 1$ dimensions, cette minimisation itérative pourra être réalisée dans un espace réduit de dimension n . Soulignons qu'il s'agit de réaliser une seule optimisation non linéaire d'un critère lui-même obtenu comme un minimum et non une suite d'optimisations alternées portant tantôt sur x_n , tantôt sur x_l ! Cette optimisation imbriquée, appelée aussi critère projeté, n'est pas nouvelle en soi, (Barham, 1972), mais n'a semble-t-il pas beaucoup attiré l'attention de notre communauté, ou n'est que très rapidement mentionnée (en quelques lignes) comme dans (Walter, 1994). Elle s'écrit de la façon suivante :

$$\min_{x_n, x_l} f(x_n, x_l) = \min_{x_n} \left[\min_{x_l} f(x_n, x_l) \right] = \min_{x_n} \varphi(x_n) \quad [10]$$

sachant que $\varphi(x_n) = \min_{x_l} f(x_n, x_l)$ est un problème explicite calculable d'une manière numériquement stable par factorisation orthogonale (Lawson, 1974).

3. Evaluation du critère φ

A partir de [7], les $N-l$ valeurs de η_l auxquelles s'ajoutent les n conditions initiales $\tilde{\eta}_{l+1}$ peuvent être réécrites sous forme matricielle :

$$\eta = D(x_n)x_l = Dx_l \quad [11]$$

en notant :

$$\begin{cases} \eta = [\eta_{l+1-n}, \dots, \eta_l, \eta_{l+1}, \dots, \eta_N]^T \in \mathbb{R}^{N-l+n} \\ y = [y_{l+1-n}, \dots, y_l, y_{l+1}, \dots, y_N]^T \in \mathbb{R}^{N-l+n} \\ D^T = [d_{l+1-n}, \dots, d_l, d_{l+1}, \dots, d_N] \in \mathbb{R}^{(n+m+1) \times (N-l+n)} \end{cases} \quad [12]$$

les lignes de D étant calculées par la récurrence :

$$d_t = -[d_{t-1}, \dots, d_{t-n}]a + [\tilde{u}_t, 0, \dots, 0]; \quad t = l+1, \dots, N \quad [13-a]$$

avec a défini en [2-a]. Elles sont initialisées par :

$$\begin{bmatrix} d_{l+1-n}^T \\ \vdots \\ d_l^T \end{bmatrix} = \left\{ \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} 0 & & 1 \\ & \ddots & \\ 1 & & 0 \end{bmatrix} \right\} \in \mathbb{R}^{n \times (n+m+1)} \quad [13-b]$$

On en déduit l'expression du critère [9] :

$$\begin{cases} f(x_n, x_l) = f(x) = \|D(x_n)x_l - y\|_2^2 \\ x^T = [x_n^T, x_l^T] \in \mathbb{R}^{2n+m+1} \end{cases} \quad [14]$$

Posons $M = [D \ y]$. Soit Q une matrice orthogonale (obtenue par des transformations de Householder) (Lawson, 1974) telle que le produit QM soit triangulaire supérieure :

$$\tilde{M} = QM = \begin{bmatrix} U & c \\ 0 & \alpha \\ 0 \dots 0 \end{bmatrix}; U \in \mathbb{R}^{(n+m+1) \times (n+m+1)}, c \in \mathbb{R}^{n+m+1}, \alpha \in \mathbb{R} \quad [15]$$

Alors :

$$\varphi(x_n) = \alpha^2(x_n) \quad [16]$$

Les paramètres optimaux sont alors donnés par :

$$\begin{cases} \hat{x}_n = \arg \min_{x_n} \alpha^2(x_n) ; 0 < x_n < x_{\sup} \\ \hat{x}_l : U(\hat{x}_n) \hat{x}_l = c(\hat{x}_n) \end{cases} \quad [17]$$

la borne supérieure x_{\sup} étant donnée par [3]. La minimisation peut se faire par une procédure de type régression non linéaire (Coleman, 1996) ou par une technique de quasi Newton factorisée (Powell, 1975) dont la robustesse est bien connue. Dans le cas où un seul paramètre « non linéaire » intervient, on a fait appel à la méthode Brent parfaitement adaptée à ce contexte (Brent, 1973).

En résumé, l'équation [15] exprime le fait que l'on peut connaître le minimum de [14] par rapport à x_l indépendamment de \hat{x}_l lui-même. Le formalisme [17] caractérise donc un découplage complet du problème d'optimisation de [14] entre partie quadratique et non linéaire.

4. Optimisation et calculs post optimisation

4.1. Calcul du Jacobien

L'optimisation non linéaire introduire dans [17] passe par le calcul du Jacobien de la sortie calculée :

$$J_x = \begin{bmatrix} J_{x_n} & J_{x_l} \end{bmatrix} = \frac{\partial \eta}{\partial x} \in \mathfrak{R}^{(N-l+n) \times (2n+m+1)} \quad [18]$$

Pour des raisons de commodité, on a fait apparaître une partition en paramètres « non linéaires » et en paramètres « linéaires ». De [14], il vient :

$$\frac{\partial f}{\partial x} = 2(\eta - y)^T J_x \quad [19]$$

et de [11] on déduit :

$$J_{x_l} = D \quad [20]$$

Pour la partie non linéaire, il faut d'abord évaluer un Jacobien J_a par rapport au vecteur a . Posons $e_t^T = \partial \eta_t / \partial a$ et introduisons la matrice :

$$E^T = [e_{l+1-n}, \dots, e_l, e_{l+1}, \dots, e_N] \in \mathfrak{R}^{n \times (N-l+n)} \quad [21]$$

où les vecteurs e_t sont donnés par la récurrence suivante déduite de [7] :

$$e_t = -\tilde{\eta}_t - [e_{t-1}, \dots, e_{t-n}] a ; \quad t = l+1, \dots, N \quad [22]$$

initialisée par : $e_{l+1-n} = \dots = e_l = 0$. Alors, on a : $J_a = E$. Il reste à calculer la matrice de transformation $P = \partial a / \partial x_n$. Deux cas sont à considérer selon que l'on traite d'un pôle réel ou d'un pôle complexe. Soit $z = r$ un pôle réel. Notons :

$$A(z) = 1 + a_1 z^{-1} + \dots + a_n z^{-n} = (1 + \alpha_1 z^{-1} + \dots + \alpha_{n-1} z^{-(n-1)}) (1 - r z^{-1})$$

Alors, il vient :

$$\partial a / \partial r = -[1, \alpha_1, \dots, \alpha_{n-1}]^T = v_r^T \quad [23-a]$$

Pour un couple (ρ, θ) définissant une paire de pôles complexes conjugués, on a :

$$A(z) = 1 + a_1 z^{-1} + \dots + a_n z^{-n} = (1 + \alpha_1 z^{-1} + \dots + \alpha_{n-2} z^{-(n-2)}) (1 - 2\rho \cos \theta z^{-1} + \rho^2 z^{-2})$$

d'où on déduit :

$$\begin{cases} \partial a / \partial \rho = -2 \cos(\theta) [1, \alpha_1, \dots, \alpha_{n-2}, 0]^T + 2\rho [0, 1, \alpha_1, \dots, \alpha_{n-2}]^T = v_\rho^T \\ \partial a / \partial \theta = -2\rho \sin(\theta) [1, \alpha_1, \dots, \alpha_{n-2}, 0]^T = v_\theta^T \end{cases} \quad [23-b]$$

Compte tenu de la définition de x_n la matrice P s'écrit :

$$P = [v_r^1; \dots; v_r^{n_r}; v_\rho^1, v_\theta^1; \dots; v_\rho^{n_c}, v_\theta^{n_c}] \quad [24]$$

Finalement le Jacobien J_{x_n} relatif aux paramètres « non linéaires » est :

$$J_{x_n} = EP \quad [25]$$

4.2. Calcul de la matrice de covariance

Une fois le minimum $\hat{x} \in \Re^{2n+m+1}$ obtenu, le calcul de la matrice de covariance C_x des paramètres, en particulier celui des écarts type, représente un calcul post optimisation essentiel auquel on a ajouté ici celui du facteur de conditionnement des optimisations réalisées. Notons que les gradients de f et φ par rapport à x_n sont identiques mais leurs Hessiens respectifs diffèrent. Quoiqu'il en soit, on peut utiliser l'approximation classique du Hessian à l'aide du Jacobien seul, c'est-à-dire :

$$H_{xx} = 2J_x^T J_x = 2 \begin{bmatrix} P^T E^T \\ D^T \end{bmatrix} \begin{bmatrix} EP & D \end{bmatrix} \Rightarrow \quad [26]$$

$$C_x = \hat{f} H_{xx}^{-1} / \beta = \begin{bmatrix} C_n & C_{nl} \\ C_{nl}^T & C_l \end{bmatrix}$$

où β est le nombre de degrés de liberté, i.e. $\beta = N - l - n - m - 1 > 1$, contrainte qui impose donc un nombre minimum de données N . Le calcul de C doit être réalisé sans former explicitement H_{xx} dont le conditionnement peut être très élevé. Soit Q une matrice orthogonale (obtenue par transformations de Householder) telle que $Q[EP \ D] = U$ soit triangulaire supérieure. Soit \tilde{U} la matrice formée des $2n + m + 1$ premières lignes et colonnes de U . Soit V la matrice solution du

système triangulaire $\tilde{U}V = I$, I la matrice unité de dimension appropriée. Alors $H_{xx}^{-1} = 2V^T V$. Notons v_k les colonnes de V . On en déduit :

$$\sigma_x^2 = \text{diag}\{C_x\} = 2 \left[\|v_1\|_2^2, \dots, \|v_{2n+m+1}\|_2^2 \right] \hat{f}/\beta > 0 \quad [27]$$

Cet algorithme est le seul numériquement stable qui garantisse le caractère positif des variances à partir du Jacobien de la sortie modèle.

Une autre voie consiste à exploiter l'approximation du Hessien à l'optimum fourni par les méthodes d'optimisation non linéaire du type quasi newton, pour ce qui concerne les paramètres « non linéaires ». Cette approche jouit des mêmes propriétés numériques avec l'algorithme de Powell (Powell, 1975) qui travaille avec une forme factorisée du type LDL^T .

4.3. Calcul de la matrice de covariance

Le facteur de conditionnement du problème d'optimisation global est défini par : $\xi_f = \sqrt{\|H_{xx}\|_2 \|H_{xx}^{-1}\|_2}$. Compte tenu de la factorisation réalisée précédemment il vient :

$$\xi_f = \|V\|_2 \|\tilde{U}\|_2 \quad [28]$$

Ce nombre est généralement élevé compte tenu de la nature de l'ensemble des paramètres estimés. Une aide pour la détermination du nombre de zéros et de pôles vient directement de l'évaluation de leur écart type, mais celle-ci peut être efficacement complétée par le « degré d'indépendance linéaire » des colonnes du Jacobien relatifs aux sous ensembles de paramètres sélectionnés. Ceux-ci fournissent les facteurs de conditionnement des sous matrices (ou partie de) D et EP . En notant \tilde{U}_n et \tilde{U}_l les colonnes de \tilde{U} correspondant respectivement aux paramètres x_n et x_l (notations identiques pour la matrice V) alors les conditionnements associés à ces sous-ensemble de paramètres sont donnés par :

$$\xi_n = \|V_n\|_2 \|\tilde{U}_n\|_2 ; \xi_l = \|V_l\|_2 \|\tilde{U}_l\|_2 \quad [29]$$

On peut de la même manière estimer ξ_b en ne s'intéressant qu'aux colonnes de D relatives à l'estimation des coefficients du numérateur du transfert.

5. Validation

L'étape de validation consiste à tester la capacité du modèle identifié à simuler le comportement du procédé avec un jeu de données d'entrée – sorties autre que celui utilisé pour l'identification. Il est clair que les conditions initiales estimées simultanément avec les paramètres du transfert n'ont aucune raison d'être adaptées aux données de validation. Cette étape est donc plus qu'une simple simulation et

comporte la ré identification des conditions initiales appropriées à ces nouvelles données. Il s'agit d'une partie des paramètres notés x_l ce qui conduit à une nouvelle factorisation orthogonale de la partie correspondante de la matrice D . On sait que le critère à minimiser est $f(x) = \|D(x_n)x_l - y\|_2^2$ pour x_n et b connus. Partitionnons D en $\begin{bmatrix} D_b & D_y \end{bmatrix}$ conformément à la partition $x_l = [b^T, \tilde{\eta}_{l+1}^T]^T$. Il vient :

$$\begin{cases} f(x) = \|D(x_n)x_l - y\|_2^2 = \|D_b(x_n)b + D_y(x_n)\tilde{\eta}_{l+1} - y\|_2^2 \\ = \|D_y(x_n)\tilde{\eta}_{l+1} - w\|_2^2; \quad w = y - D_b(x_n)b \end{cases} \quad [30]$$

La minimisation de f par rapport à $\tilde{\eta}_{l+1}$ se fait par factorisation orthogonale de $\begin{bmatrix} D_y & z \end{bmatrix}$ de la même manière que pour le calcul de φ (15). Les nouvelles conditions initiales sont donc obtenues par résolution d'un simple système triangulaire et la valeur du critère erreur de sortie déduit du dernier élément de la matrice $\begin{bmatrix} D_y & z \end{bmatrix}$ dans sa version triangularisée. La simulation est conduite par le doublet x_n, b identifiés et $\tilde{\eta}_{l+1}$ estimé avec le fichier de validation.

6. Une propriété exclusive

Traditionnellement, on estime les coefficients du polynôme dénominateur à partir desquels on calcule les pôles du transfert, seuls paramètres caractérisant la dynamique du système. Ces coefficients a sont estimés avec une certaine incertitude mesurée généralement par leur écart type. Dans le meilleur des cas cette incertitude est de l'ordre du pour cent. Or cette incertitude, aussi petite soit elle, définit un ensemble de polynômes dont les racines peuvent varier considérablement et en particulier sortir du cercle unité. Autrement dit, même une petite incertitude sur a ne permet pas de tirer de conclusion sur la précision de la dynamique. A fortiori quand l'incertitude sur a dépasse 10% voire 20%, cela signifie que l'on ne sait rien de la dynamique du système. Pour s'en convaincre, voici un exemple avec un polynôme d'ordre 3 dont les coefficients sont ensuite faiblement perturbés (valeur relative de la perturbation : 1.00e-002), avec un conditionnement du Jacobien de 1.59e+003.

```
a = 1.00000e+000 -2.70850e+000 2.43919e+000 -7.30492e-001
racines = 8.37875e-001 8.80770e-001 9.89860e-001
a perturbé = 1.00000e+000 -2.72006e+000 2.42592e+000 -7.29756e-001
racines perturbées = (7.44697e-001-1.95964e-001i)
                    (7.44697e-001+1.95964e-001i) 1.23067e+000
ratio ||dr||/||da|| = 2.28543e+001
```

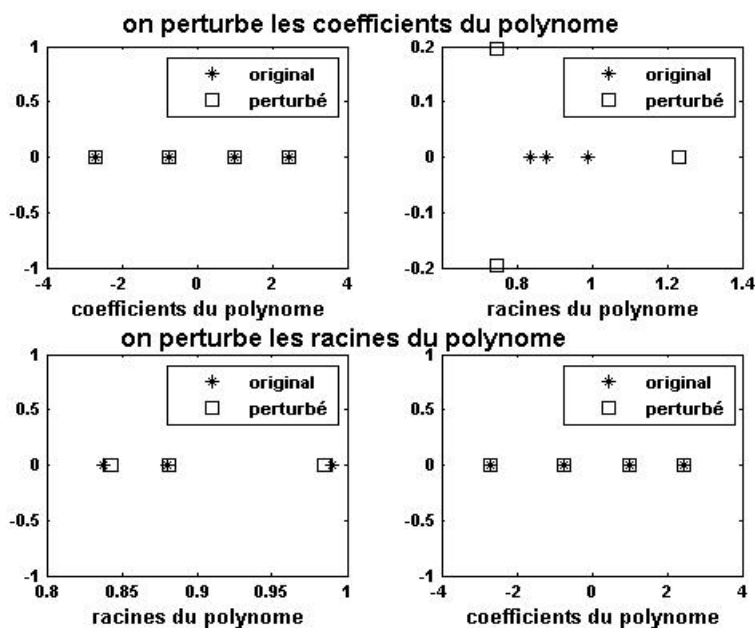
Ces résultats assez typiques parlent d'eux même. A contrario, si on perturbe les racines, les coefficients du nouveau polynôme varie dans les mêmes proportions :

```

racine = 8.37875e-001  8.80770e-001  9.89860e-001
a = 1.00000e+000 -2.70850e+000  2.43919e+000 -7.30492e-001
perturbation sur les racines = 5.29882e-003  3.88612e-004 -5.50544e-003
racine perturbées = 8.43173e-001  8.81158e-001  9.84354e-000
a perturbé = 1.00000e+000 -2.70869e+000  2.44032e+000 -7.31345e-001
perturbation sur a = 0.00000e+000 -1.81986e-004  1.13125e-003 -8.53682e-004
prévision via J = 0.00000e+000 -1.81986e-004  1.16050e-003 -8.79142e-004

```

Cette difficulté extraordinaire, bien connue des numériciens, a fait dire à Wilkinson en 1984 à propos de ses fameux polynômes « Speaking for myself I regard it as the most traumatic experience in my career as a numerical analyst ». Pour expliquer ce phénomène, au premier ordre et en supposant que la nature des racines reste individuellement inchangée, il suffit d'évaluer le Jacobien $\partial a / \partial x$ a étant les coefficients du polynôme et x ses racines. Celui-ci est le plus souvent très mal conditionné de sorte que numériquement il possède « presque » un noyau. En d'autres termes, cela signifie (au premier ordre) que l'on peut trouver une perturbation dx (de norme arbitraire) des racines telles que $da = [\partial a / \partial x] dx \approx 0$, c'est-à-dire qu'à une très petite perturbation des coefficients peut correspondre une grande variation des racines. Le schéma ci-dessous résume ces résultats.



7. Une approche versatile

Les extensions discutées ici sont sans doute classiques mais leur traitement n'en est pas moins spécifiques car reposant explicitement sur les deux points clefs que sont critère projeté et paramétrisation directe de la dynamique par les pôles. En particulier, l'estimation d'un offset ou de conditions initiales ne modifie pas la dimension du problème d'optimisation non linéaire puisque relevant de la partie « linéaire dans les paramètres ». La prise en compte de contraintes linéaires, égalités ou inégalités sur x_l reste d'une complexité modeste comparée à leur prise en compte dans un contexte général d'optimisation non linéaire. De même gérer des connaissances a priori du type intégrateur dans le modèle, ou imposer des pôles multiples devient trivial avec le choix paramétrique « pôle ».

7.1. Estimer un offset sur la sortie

Les points de fonctionnement autour desquels sont réalisées les identifications ne sont a priori jamais nuls. Pour s'y ramener, on retranche habituellement des mesures une constante dont l'évaluation n'est pas toujours aisée en particulier quand les débuts d'enregistrements se font à un moment où le procédé n'est pas stabilisé. Pour garantir une erreur de sortie centrée il suffit d'estimer cet offset en ajoutant au vecteur x_l un paramètre supplémentaire noté ici γ . Toute chose étant identique par ailleurs, la prise en compte de cette inconnue se résume donc à ajouter à la matrice D , voir les équations [11] et [12], une colonne de 1.

7.2. Ajouter des contraintes sur x_l

Le calcul de $\varphi(x_n)$ et de x_l est un problème de moindres carrés. Tout en maintenant l'approche factorisée pour sa stabilité numérique, il est aisé d'ajouter des contraintes linéaires égalité ou inégalité (Lawson, 1974), (Bjorck, 1996) sans pénaliser véritablement le coût calcul (même complexité, algorithme fini). Par exemple, on peut imposer une valeur au gain statique, ou son signe, ou encore celui de certains coefficients b_k . En revanche, si un paramètre b_k doit être nul, la contrainte est directement satisfaite en supprimant la colonne afférente de D .

7.3. Identifier des « intégrateurs »

Lorsqu'un ou plusieurs pôles estimés sont « très voisins » de 1, et si l'on sait a priori que la modélisation du couple entrée – sortie considéré nécessite la présence de pôle(s) valant exactement 1, alors il faut réduire la dimension de x_n en conséquence et introduire cette connaissance dans la structure du modèle. Soit donc

n_1 la multiplicité d'un pôle $z=1$. Le dénominateur de la fonction de transfert à estimer se présente alors comme suit :

$$\begin{cases} A(z) = (1 + a_1 z^{-1} + \dots + a_{n-n_1} z^{-n-n_1}) (1 - z^{-1})^{n_1} \\ = \prod_{k=1}^{n-n_1} (1 - z_k z^{-1}) \prod_{k=1}^{n_1} (1 - z^{-1}) \\ A(z) = (1 + \tilde{a}_1 z^{-1} + \dots + \tilde{a}_n z^{-n}) \end{cases} \quad [31]$$

L'optimisation non linéaire précédente est alors mise en œuvre en utilisant \tilde{a} en lieu et place de a . On introduit donc une fonction intermédiaire de plus $\tilde{a}[a(x_n)]$ nécessitant un Jacobien intermédiaire $\partial \tilde{a} / \partial a$ dont le calcul est trivial.

On a :

$$a^k = \begin{bmatrix} a^{k-1} \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ a^{k-1} \end{bmatrix}; a^0 = a; \tilde{a} = a^{n_1}$$

$$\frac{\partial \tilde{a}}{\partial a} = \frac{\partial a^{n_1}}{\partial a^{n_1-1}} \cdot \frac{\partial a^{n_1-1}}{\partial a^{n_1-2}} \dots \frac{\partial a^1}{\partial a} \cdot \frac{\partial a^k}{\partial a^{k-1}} = \begin{bmatrix} +1 & & & \\ -1 & \ddots & & \\ & \ddots & +1 & \\ & & -1 & \end{bmatrix} \in \Re^{(n+k) \times (n+k-1)}$$

7.4. Gérer la multiplicité des pôles

Trouver les racines multiples d'un polynôme est un problème numériquement indécidable. En revanche introduire comme connaissance a priori ou comme contrainte de modélisation la multiplicité des pôles inconnus est ici très simple dans la mesure où les inconnues à estimer sont directement les pôles du système. C'est, entre autre, un moyen de réduire la dimension de x_n et l'ordre de grandeur du conditionnement du problème d'optimisation, le cas extrême étant le modèle de Strejc dépendant d'une seule inconnue « non linéaire ». Désignons par $\mu_k, k=1, \dots, \nu$; la multiplicité des ν pôles inconnus (réel ou complexe). Il vient :

$$\begin{cases} A(z) = (1 + a_1 z^{-1} + \dots + a_n z^{-n}) = \prod_{k=1}^{\nu} (1 - z_k z^{-1})^{\mu_k} \\ n = \sum_{k=1}^{\nu} \mu_k \end{cases} \quad [32]$$

L'optimisation $\varphi(x_n)$ est mise en œuvre en utilisant la nouvelle fonction $a(x_n)$ définie par [32] à laquelle s'ajoute l'évaluation d'un nouveau Jacobien $\partial a / \partial x$. Pour un pôle réel α de multiplicité μ on obtient :

$$\left\{ \begin{array}{l} a^{k+1} = \begin{bmatrix} a^k \\ 0 \end{bmatrix} - \alpha \begin{bmatrix} 1 \\ a^k \end{bmatrix} \\ \partial a^{k+1} / \partial \alpha = - \begin{bmatrix} 1 \\ a^k \end{bmatrix} + \begin{bmatrix} \partial a^k / \partial \alpha \\ 0 \end{bmatrix} - \alpha \begin{bmatrix} 0 \\ \partial a^k / \partial \alpha \end{bmatrix} \end{array} \right. \quad \left\{ \begin{array}{l} a^0 = (a_1, \dots, a_n)^T \\ \partial a^0 / \partial \alpha = [0, \dots, 0]^T \end{array} \right.$$

$$k = 0, \dots, \mu - 1$$

La colonne correspondant du Jacobien est alors donnée par $\partial a^\mu / \partial \alpha$. Pour une paire de pôles complexes conjugués α et $\bar{\alpha}$ de module ρ et d'argument θ dont la multiplicité est μ on a les résultats similaires suivants :

$$\left\{ \begin{array}{l} a^{k+1} = \begin{bmatrix} a^k \\ 0 \\ 0 \end{bmatrix} - 2\rho \cos(\theta) \begin{bmatrix} 1 \\ a^k \\ 0 \end{bmatrix} + \rho^2 \begin{bmatrix} 0 \\ 1 \\ a^k \end{bmatrix} \\ \partial a^{k+1} / \partial \rho = -2 \cos(\theta) \begin{bmatrix} 1 \\ a^k \\ 0 \end{bmatrix} + 2\rho \begin{bmatrix} 0 \\ 1 \\ a^k \end{bmatrix} + \begin{bmatrix} \partial a^k / \partial \rho \\ 0 \\ 0 \end{bmatrix} - 2\rho \cos(\theta) \begin{bmatrix} 1 \\ \partial a^k / \partial \rho \\ 0 \end{bmatrix} + \dots \\ \partial a^{k+1} / \partial \theta = 2\rho \sin(\theta) \begin{bmatrix} 1 \\ a^k \\ 0 \end{bmatrix} + \begin{bmatrix} \partial a^k / \partial \theta \\ 0 \\ 0 \end{bmatrix} - 2\rho \cos(\theta) \begin{bmatrix} 1 \\ \partial a^k / \partial \theta \\ 0 \end{bmatrix} + \rho^2 \begin{bmatrix} 0 \\ 1 \\ \partial a^k / \partial \theta \end{bmatrix} \end{array} \right. \quad \rho^2 \begin{bmatrix} 0 \\ 1 \\ \partial a^k / \partial \rho \end{bmatrix}$$

$$k = 0, \dots, \mu - 1$$

initialisés par :

$$\left\{ \begin{array}{l} a^0 = (a_1, \dots, a_n)^T \\ \partial a^0 / \partial \rho = \partial a^0 / \partial \theta = [0, \dots, 0]^T \end{array} \right.$$

7.5. Identification à partir de plusieurs fichiers de données

Cette extension répond à deux situations clefs. Les données expérimentales sont rarement exemptes de mesures aberrantes nécessitant des coupures dans les fichiers. Le problème est alors comment exploiter « au mieux » les données restantes. En outre, lors de campagnes d'essais la richesse des entrées peut varier considérablement en fonction d'éventuelles contraintes d'exploitation et on peut souhaiter trouver le modèle ajustant au mieux l'ensemble des essais dédiés à l'identification. L'algorithme proposé incluant par défaut l'estimation des conditions initiales, la solution proposée consiste à utiliser toutes les données disponibles en concaténant les différents fichiers de données en un seul et en estimant autant de jeux de conditions initiales qu'il y a de fichiers. Soit n_f le nombre de fichiers considérés que l'on concatène en un seul fichier dans lequel $t_k, k = 1, \dots, n_f + 1$ désignent les numéros d'échantillons auxquels commence chaque fichier. Les cas limites sont donnés par $(t_1 = 1)$, le dernier fichier se terminant à l'instant $(t_{n_f+1} - 1)$. Le nombre de données utiles pour chaque fichier est donc $N_k = t_{k+1} - t_k$. Rappelons que pour un fichier la matrice D peut être partitionnée en $\begin{bmatrix} D_b & D_y \end{bmatrix}$. La construction de la matrice pour l'ensemble des fichiers s'obtient comme suit, l'indice supérieur désignant le numéro du fichier utilisé :

$$D = \begin{bmatrix} D_b^1 & D_y^1 & 0 & 0 \\ D_b^2 & 0 & D_y^2 & 0 \\ \vdots & \vdots & \ddots & \\ D_b^{n_f} & 0 & & D_y^{n_f} \end{bmatrix}; x_l^T = \begin{bmatrix} b^T, (\eta_{l+1}^1)^T, \dots, (\eta_{l+1}^{n_f})^T \end{bmatrix} \quad [33]$$

Le reste de la procédure reste inchangé. Ceci signifie notamment que la dimension du problème d'optimisation non linéaire n'est pas affectée par cette stratégie.

7.6. Systèmes MISO

Pour un problème ayant p entrées et une sortie, chaque entrée est modélisée par un transfert $H(z)$ tel que celui défini en [1] de sorte que la relation entrées – sortie s'écrit :

$$\eta(z) = H_1(z)u_1(z) + \dots + H_p(z)u_p(z) = \eta^1(z) + \dots + \eta^p(z) \quad [34]$$

Chaque sous sortie $\eta^k, k = 1, \dots, p$ est calculée comme dans le cas SISO selon une équation du type [11] $\eta^k = D_k x_l$ dans laquelle l'indice k ajouté porte sur le numéro de l'entrée considérée. Pour pouvoir implémenter [34] dans le domaine

temporel, il faut que les matrices D_k relatives à chaque entrée comportent le même nombre de ligne. Une fois chaque calcul du type SISO réalisé seules les lignes d'indice supérieur à $\max_k(l_k - n_k)$ seront conservées pour réaliser la concaténation des matrices D_k . On a noté par n_k est le nombre de pôles associé à l'entrée k et l_k la valeur de l relatif à l'entrée k . L'équation [11] devient :

$$\eta = \begin{bmatrix} D_1(x_n^1) & \cdots & D_p(x_n^p) \end{bmatrix} \begin{bmatrix} x_l^1 \\ \vdots \\ x_l^p \end{bmatrix} \quad [35]$$

Le traitement est ensuite identique au cas SISO.

8. Systèmes MIMO et modèle d'état

Le modèle est maintenant défini par un modèle d'état à temps discret :

$$\begin{cases} x_{t+1} = Ax_t + Bu_t; \eta_t = Cx_t; \\ A \in \mathfrak{R}^{n \times n}, B \in \mathfrak{R}^{n \times p}; C \in \mathfrak{R}^{q \times n} \end{cases} \quad [36]$$

Les paramètres sont tout ou partie des éléments des matrices A , B , C et de l'état initial x_1 .

8.1. Extension du cas MISO

Le problème MIMO peut être traité comme q problèmes MISO. Ce modèle est obtenu par une association classique des modèles d'état de chaque transfert. Pour un transfert identifié [1] entre l'entrée i et la sortie j deux cas de figures sont à considérer selon que $l = k + m$ ou $l = n$:

$$\begin{aligned}
A_{ij} &= \begin{bmatrix} \alpha_1 & 1 & & \\ \vdots & & \ddots & \\ \alpha_{l-1} & & & 1 \\ \alpha_l & 0 & \cdots & 0 \end{bmatrix}; b_{ij} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_l \end{bmatrix}; \\
\left\{ \begin{array}{l} \alpha^T = [-a_1, \dots, -a_n, 0, \dots, 0] \\ \beta^T = \left[\underbrace{0, \dots, 0}_{k-1}, b_0, \dots, b_m \right] \end{array} \right\} l = k + m > n \\
\left\{ \begin{array}{l} \alpha^T = [-a_1, \dots, -a_n] \\ \beta^T = \left[\underbrace{0, \dots, 0, b_0, \dots, b_m}_{k+m}, 0, \dots, 0 \right] \end{array} \right\} l = n \geq k + m \\
c_{ij}^T = [1 \quad 0 \quad \cdots \quad 0]
\end{aligned} \quad [37-a]$$

A l'issue d'une identification MISO, on peut construire le modèle d'état :

$$A^j = \begin{bmatrix} A_{1j} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & A_{pj} \end{bmatrix}; B^j = \begin{bmatrix} b_{1j} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & b_{pj} \end{bmatrix}; C^j = [c_{1j}^T \quad \cdots \quad c_{pj}^T] \quad [37-b]$$

puis concaténer q structures de la forme [37-b] :

$$A = \begin{bmatrix} A^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & A^q \end{bmatrix}; B = \begin{bmatrix} B^1 \\ \vdots \\ B^q \end{bmatrix}; C = \begin{bmatrix} C^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & C^q \end{bmatrix} \quad [37-c]$$

Habituellement, il en résulte, pour des données simulées sans bruit, un modèle d'état non minimal. Pour des données réelles, ou des données simulées et bruitées on ne peut plus parler de formes d'état minimales ou non, car on est dans le domaine de l'approximation au sens ici du critère erreur de sortie. Bien que l'on respecte le principe de parcimonie au niveau de chaque transfert, à qualité de fit identique, on peut, en général, construire des modèles d'état de dimension inférieure.

8.2. Approche directe

On peut considérer de nouveau la minimisation du critère d'erreur de sortie $\sum_t \|y_t - \eta_t\|_2^2$ par rapport aux paramètres du modèle [36]. Enoncé tel quel, le

problème est numériquement insoluble pour cause de sur paramétrage (conditionnement arbitrairement grand). Les approches du type sous espace évitent en partie la difficulté numérique en imposant une erreur linéaire dans les paramètres au détriment d'un critère n'ayant plus un sens physique et au risque d'une estimation biaisée. De surcroît, le sur paramétrage considérable rend non significatif toute notion de précision sur les paramètres.

Sauf paramétrage spécifique à une application donnée que l'on ne traite pas ici en raison de la spécificité de chaque problème, on propose un algorithme respectant le principe de parcimonie et le paramétrage exploité pour les transferts. Il est conçu de telle sorte que l'on puisse distinguer de nouveau des paramètres « linéaires », la partie des éléments de B à estimer, certains éléments de C et l'état initial x_1 et des paramètres « non linéaires », les valeurs propres des blocs A_{ij} dont sont déduits les éléments de A_{ij} à estimer.

L'algorithme proposé opère sortie par sortie selon les étapes suivantes :

– Sortie k=1

-Réaliser une identification MISO via p transferts en z selon l'algorithme d'optimisation décrit précédemment.

-Convertir en une forme d'état MISO [37-b].

Estimer l'état initial x_1 par moindres carrés, i.e. :

$$\hat{x}_1 = \arg \min_{x_1} \left\| \begin{bmatrix} y^k - (CX)^T \end{bmatrix} - Mx_1 \right\|_2^2 \quad [38]$$

$y^k \in \mathfrak{R}^N$ étant le vecteur de tous les échantillons de la sortie k=1. Les matrices M et $X = X^1$ sont données par les récurrences :

$$\begin{cases} M = \begin{bmatrix} v_1^T, \dots, v_N^T \end{bmatrix}^T ; v_t = v_{t-1}A, t = 2, \dots, N; v_1 = C; \\ X^1 = \begin{bmatrix} \tilde{x}_1, \dots, \tilde{x}_N \end{bmatrix} ; \tilde{x}_t = A\tilde{x}_{t-1} + Bu_{t-1}, t = 2, \dots, N; \tilde{x}_1 = 0; \end{cases} \quad [39]$$

– Sortie k = 2, ..., q

-Calculer $X^{k-1} = [\hat{x}_1, x_2, \dots, x_N]$ $x_t, t = 2, \dots, N$ étant les états du modèle d'état obtenu pour la sortie $k-1$, initialisés par \hat{x}_1 , et construire $X = [X^1, \dots, X^{k-1}]$.

-Puis calculer un lot de paramètres $\hat{\alpha}$ par

$$\hat{\alpha} = \arg \min_{\alpha} \|y^k - X\alpha\|_2^2 \Rightarrow \hat{y}^k = X\hat{\alpha}; \tilde{y}^k = y - \hat{y}^k; \quad [40]$$

$$\alpha^T = [\alpha_k^1, \dots, \alpha_k^{k-1}]$$

-Réaliser une identification MISO avec la sortie virtuelle \tilde{y}^k via les transferts puis une conversion en modèle d'état [37-b]. Le modèle d'état pour les k premières sorties est :

$$A = \begin{bmatrix} A^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & A^k \end{bmatrix}; B = \begin{bmatrix} B^1 \\ \vdots \\ B^k \end{bmatrix}; C = \begin{bmatrix} c_1^T & 0 & 0 \\ \vdots & \ddots & 0 \\ \alpha_k^1 & \alpha_k^{k-1} & c_k^T \end{bmatrix} \quad [41]$$

-Calculer l'état relatif aux états introduits par la sortie k , par moindres carrés, démarche similaire à [38].

Cet algorithme est un compromis entre parcimonie, robustesse numérique, et optimisation globale du critère erreur de sortie. Il n'est pas optimal en ce sens que le résultat obtenu n'est pas invariant par rapport à l'ordre dans lequel sont traitées les sorties.

Remarque : La connaissance des conditions initiales sur les sous sorties modèles en [34], sauf situation particulière, ne permet pas de remonter à l'état initial. En conséquence cet état initial est re-estimer par [38], le terme Mx_1 étant la sortie due à l'état initial seul et $(CX)^T$ celle due à l'entrée, à état initial nul, [39].

9. Exemples

9.1. Validation d'un modèle

Définition : On considère qu'un modèle est valide si à la fois ses paramètres ont une bonne précision (inférieure à un seuil donné), l'erreur entre les sorties simulées et mesurées est cohérente avec le rapport signal sur bruit et enfin qu'aucune partie des mesures de sortie ne soit significativement moins bien prédite par la sortie simulée, comparativement à l'ensemble des données.

Cette définition s'inspire clairement des résultats présentés en (Leseq, 2005) avec toutefois deux différences. La troisième condition introduite ici n'apparaît pas dans (Leseq, 2005) et on ne tient pas compte de la richesse de l'entrée, supposée ici correctement choisie pour l'identification. Cette suppression arrive naturellement dans la mesure où les autres conditions sont (presque) toujours plus coercitives. Quantitativement cette validation est exprimée, pour une sortie donnée, par un degré de validité $\kappa : 0 \leq \kappa \leq 1$ défini par :

$$\begin{cases} \kappa = 1 - \min \left[1, \min \left\{ \sigma(\varepsilon) / \sigma(y), \sigma(x) ./ |x|, \sigma(\varepsilon_t) / \sigma(y_t) \right\} \right] \\ \varepsilon = \eta - y \\ I : |\varepsilon_t| > 2\sigma(\varepsilon), t \in I \end{cases} \quad [42]$$

Dans cette expression l'ensemble des indices I peut être vide. La notation style matlab « ./ » exprime une division composante par composante qui n'a lieu que pour des diviseurs non nuls. Le vecteur x désigne les paramètres du ou des transferts relatifs à une sortie. Le rôle du troisième terme est de signaler une erreur éventuelle de modèle lorsque la sortie est en moyenne quadratique bien ajustée mais au détriment de certaines zones. On suppose bien sûr que les mesures aberrantes ont été éliminées, voir le paragraphe 7.5 .

9.2. Contexte SISO

On a simulé sur 250 points une réponse indicielle avec une large plage mettant en évidence le point de fonctionnement du système. Le niveau de bruit (rapport des écarts types sortie/bruit additif) est de 5%.

La figure 1 et la table 1 illustrent le rôle de chacun des termes constituant le degré de validité introduit en [42]. Il s'agit de la réponse à un échelon d'un second ordre aperiodique présentant un non minimum de phase. Le modèle « 1 » est un premier ordre retardé dont la qualité du fit est de près de 94%. Avec la véritable structure, modèle « 2 » on obtient à peine 95%. Sur la base de ce seul critère la plus grande complexité du véritable modèle n'est pas du tout justifiée. En intégrant la précision des paramètres le choix serait plutôt en faveur du modèle « 1 ». En revanche, le troisième terme prend en compte le mauvais fit du rebond caractéristique du non minimum phase, marginal en valeur relative sur l'ensemble de l'ajustement des données mais caractéristique d'une mauvaise structure avec le modèle « 1 ». Cette analyse est typiquement prise en compte dans [42] qui conduit à un indice de validité de 0.79, modèle « 1 » et 0.88 pour le modèle « 2 ».

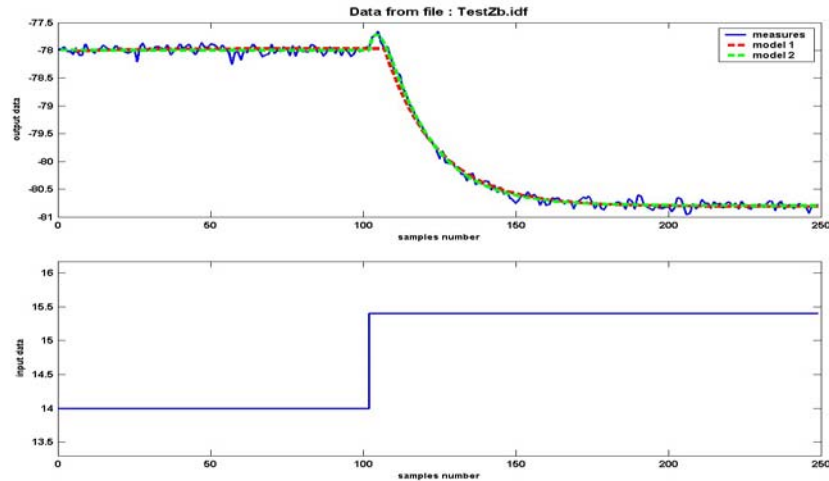


Figure 1. *Contexte SISO*

Model number : 1

Identified Transfert Function

Least Squares optimization conditioning : 1.3007e+004

Non linear parameters optimization conditioning : 1.0000e+000

Sample(s) Delay : 6

Numerator coefficient(s) : -1.1489e-001

Standard deviation : 5.0704e-004

Real pole(s) : 9.4343e-001

Std deviation : 9.3724e-004

Denominator coefficient(s) : 1.0000e+000 -9.4343e-001

Standard deviation : 0.0000e+000 9.3724e-004

Output Offset : -4.9539e+001

Output Initial Values : -7.8021e+001

Relative minimum output error : 6.1571e-002

Model Validity : 0.79

Model number : 2

Identified Transfert Function

Least Squares optimization conditioning : 5.6723e+004

Non linear parameters optimization conditioning : 2.7316e+001

Sample(s) Delay : 1

Numerator coefficient(s) : 1.4415e-001 -1.8181e-001

Standard deviation : 6.0589e-003 6.0443e-003

Real pole(s) : 9.3587e-001 7.0646e-001

Std deviation : 1.6844e-003 2.8719e-002

Denominator coefficient(s) :	1.0000e+000	-1.6423e+000	6.6115e-001
Standard deviation	: 0.0000e+000	2.1594e-002	2.8175e-002
Output Offset :	-4.9990e+001		
Output Initial Values :	-7.7987e+001-7.7989e+001		
Relative minimum output error :	5.1269e-002		
Model Validity :	0.88		

Table 1. *Contexte SISO*

9.3. Un exemple multi fichier

On dispose de deux enregistrements, une réponse indicielle et la réponse à une séquence binaire pseudo aléatoire (PRBS) choisie pour sa richesse spectrale. Chacun de ces deux contextes est caractérisé par un enregistrement de 250 points. Le niveau de bruit est de nouveau de 5%. L'objectif est d'identifier un modèle moyen ajustant au mieux les deux fichiers, i.e. les 500 mesures disponibles. La rupture entre les deux jeux de donnée est figurée par un trait vertical sur la figure 2. Les résultats, table 2 font apparaître, en toute logique, deux jeux de conditions initiales. On notera que le conditionnement du problème d'optimisation non linéaire a diminué, de même que celui des moindres carrés malgré un accroissement du nombre d'inconnues (le deuxième jeu de conditions initiales). On observe ici l'impact direct d'une entrée enrichie par la présence de la PRBS.

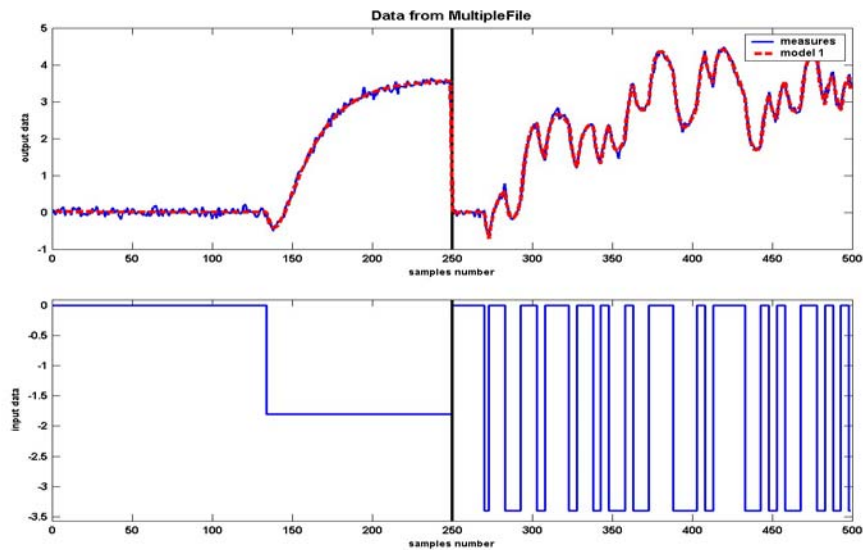


Figure 2. *Multi fichier*

Identified Transfert Function
Input conditioning : 6.1723e+000
Least Squares optimization conditioning : 4.4430e+003
Non linear parameters optimization conditioning : 1.0088e+001
Sample(s) Delay : 1
Numerator coefficient(s) : 1.0087e-001 -1.1375e-001
Standard deviation : 8.9923e-004 8.9697e-004
Real pole(s) : 9.5622e-001 8.5280e-001
Std deviation : 5.1892e-004 2.4131e-003
Denominator coefficient(s) : 1.0000e+000 -1.8090e+000 8.1546e-001
Standard deviation : 0.0000e+000 2.5092e-003 2.7573e-003
Output Offset : 3.6457e-004
Batch number : 1, Output Initial Values : 1.8733e-002 1.9078e-002
Batch number : 2, Output Initial Values : -2.3412e-002 -1.8385e-002
Relative minimum output error : 4.8443e-002
Model Validity : 0.87

Table 2. *Multi fichier***9.4. Contexte MISO**

On a simulé un système avec 3 entrées caractérisé par respectivement un premier ordre, un second ordre oscillant retardé (de 5 pas d'échantillonnage), et un troisième ordre (une constante de temps et deux pôles très peu amortis) retardé (de 2 pas d'échantillonnage).

Le premier jeu de données correspond à 3 entrées en échelon pour créer un contexte peu riche en terme d'excitation « persistante », difficulté à laquelle on a ajouté un niveau de bruit élevé (20%). On obtient néanmoins une « bonne » confiance dans les paramètres. Le degré global de validité est dominé par un fit chaotique sur certaines portions du signal de sortie, dû ici au niveau important de bruit, voir la figure 3 et la table 3.

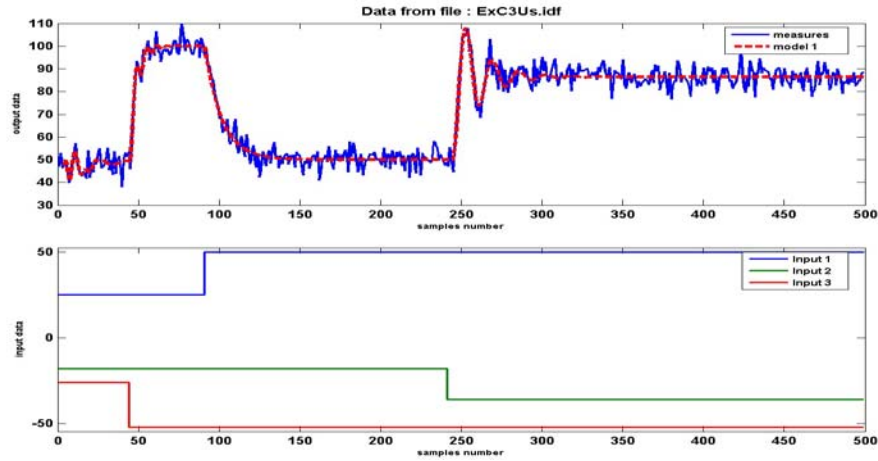


Figure 3. *ContexteMISO*

Identified Transfert Function

Least Squares optimization conditioning : 1.9629e+005

Non linear parameters optimization conditioning : 1.0775e+001

Input 1

Sample(s) Delay : 1

Numerator coefficient(s) : -1.8256e-001

Standard deviation : 2.3973e-003

Real pole(s) : 9.0862e-001

Std deviation : 4.5574e-003

Denominator coefficient(s) : 1.0000e+000 -9.0862e-001

Standard deviation : 0.0000e+000 4.5574e-003

Input 2

Sample(s) Delay : 5

Numerator coefficient(s) : -2.8334e-001

Standard deviation : 3.0301e-003

Cplx pole(s) {rho, theta} : 9.3685e-001 3.8346e-001

Standard deviation : 4.4975e-003 4.5783e-003

Denominator coefficient(s) : 1.0000e+000 -1.7376e+000 8.7770e-001

Standard deviation : 0.0000e+000 1.2976e-002 3.1528e-003

Input 3

Sample(s) Delay : 2

Numerator coefficient(s) : -3.3971e-001

Standard deviation : 7.8742e-003

Real pole(s) : 7.8390e-001

Std deviation : 1.6460e-002

Cplx pole(s) {rho, theta} : 8.6317e-001 9.9294e-001

Standard deviation : 3.3737e-002 3.3943e-002

Denominator coefficient(s) :	1.0000e+000	-1.7269e+000	1.4843e+000	-
5.8405e-001				
Standard deviation	:	0.0000e+000	4.0709e-002	9.8351e-002
3.3807e-002				
Output Offset :	1.1587e+001			
Output Initial Values :	1.9009e+001			
Relative minimum output error :	1.9359e-001			
Model Validity :	0.58			

Table 3. *ContexteMISO*

A l’opposé la figure 4 illustre une situation bien plus favorable obtenue avec trois entrées du type « PRBS », et un bruit additif de faible niveau (5%). Pour comparaison, les paramètres utilisés pour la simulation ont été ajoutés table 5.

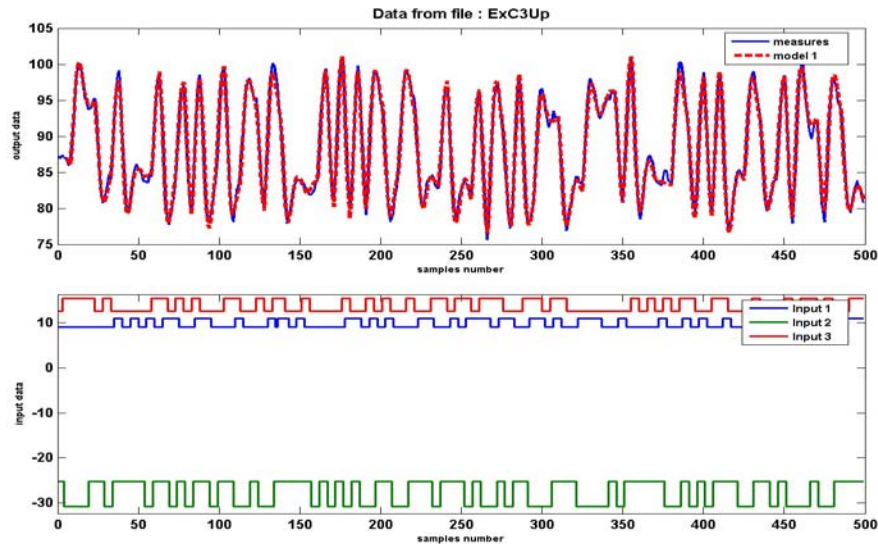


Figure 4. *ContexteMISO PRBS*

Identified Transfert Function	
Input conditioning :	9.8100e+001
Least Squares optimization conditioning :	6.1538e+010
Non linear parameters optimization conditioning :	1.4440e+001
Input 1	
Sample(s) Delay :	1
Numerator coefficient(s) :	-1.8778e-001
Standard deviation	: 1.2278e-002
Real pole(s)	: 7.5142e-001

Std deviation : 1.3708e-003
 Denominator coefficient(s) : 1.0000e+000 -7.5142e-001
 Standard deviation : 0.0000e+000 1.3708e-003
 Input 2
 Sample(s) Delay : 5
 Numerator coefficient(s) : -5.2735e-001
 Standard deviation : 2.4446e-003
 Cplx pole(s) {rho, theta} : 8.2367e-001 5.4035e-001
 Standard deviation : 1.5904e-003 9.9466e-004
 Denominator coefficient(s) : 1.0000e+000 -1.4126e+000 6.7844e-001
 Standard deviation : 0.0000e+000 3.2904e-003 1.3478e-003
 Input 3
 Sample(s) Delay : 2
 Numerator coefficient(s) : -1.1463e-001
 Standard deviation : 4.0872e-003
 Real pole(s) : 9.6596e-001
 Std deviation : 3.0854e-004
 Cplx pole(s) {rho, theta} : 1.0398e+000 1.1236e+000
 Standard deviation : 6.2430e-004 5.0448e-004
 Denominator coefficient(s) : 1.0000e+000 -1.8653e+000 1.9499e+000 -
 1.0444e+000
 Standard deviation : 0.0000e+000 1.1246e-003 2.6585e-003 1.5544e-
 003
 Output Offset : 7.9802e+001
 Output Initial Values : 8.2724e+001
 Relative minimum output error : 9.9445e-002
 Model Validity : 0.78

Table 4. *ContexteMISO, PRBS*

Theoretical parameters
 input 1
 Sample(s) Delay : 1
 Numerator coefficient(s) : -1.4633e-001
 Denominator coefficient(s) : 1.0000e+000 -9.2683e-001
 Pole(s) : 9.2683e-001
 input 2
 Sample(s) Delay : 5
 Numerator coefficient(s) : -5.3015e-001
 Denominator coefficient(s) : 1.0000e+000 -1.4142e+000 6.7930e-001
 Pole(s) : 7.0711e-001-4.2343e-001i 7.0711e-001+4.2343e-001i
 input 3
 Sample(s) Delay : 2

Numerator coefficient(s) : -4.4106e-002
Denominator coefficient(s) : 1.0000e+000 -2.4148e+000 2.2882e+000 -
8.5137e-001
Pole(s) : 9.4887e-001+0.0000e+000i 7.3296e-001-6.0002e-001i 7.3296e-
001+6.0002e-001i
Output Offset : 8.0000e+001
Output Initial Values : 8.7026e+001

Table 4. *ContexteMISO Paramètres exacts*

9.5. Performance comparée

On a comparé le comportement de notre approche à celle offerte par la méthode du modèle implémenté dans matlab, commande « oe » et la méthode des moindres carrés basique, commande « arx ». Dans les deux cas on a fixé l'option « état initial » à « estimé ». Pour que l'on puisse opérer des comparaisons on s'est limité à des fichiers de données ne contenant qu'un jeu de test, par opposition à l'aspect « multi fichiers » introduit ici. Par ailleurs ne pouvant contraindre un pôle à être multiple, on a éliminé des données simulées dans de telles conditions.

En premier lieu, il est évident que la commande « arx » ne peut rivaliser en raison du biais introduit par le bruit additif en sortie.

Deuxièmement, la présence d'offset sur les mesures est très mal prise en compte malgré l'option « detrend », sans laquelle le problème d'identification est vu insoluble (matrice de covariance singulière). L'estimation des conditions initiales ne suffit pas dans ce contexte telle qu'il est classiquement fait..

Troisièmement, la présence d'un intégrateur met en échec la procédure « oe » (et a fortiori « arx ») incapable de retrouver un pôle à 1 et donc la dynamique restante du système.

A titre d'illustration on a repris le premier test présenté précédemment, en notant « mm » notre méthode du modèle. On a les résultats suivants :

	« arx »	« oe »	référence	« mm »
Pôles	0.958	0.988	0.937	0.936
	-0.315	0.967	0.713	0.706
zéros	1.980	0.991	1.258	1.261

Il faut souligner que la précision affichée par « oe » en terme d'écart type est excellente (largement inférieure au centième) et donc pas en rapport avec la réalité. Ce biais résulte ici exclusivement de la mauvaise estimation des conditions initiales due à un offset inconnu et estimé comme devant être la moyenne des signaux entrée

et sortie, voir figure 5 à comparer avec la figure 1. On notera le pseudo centrage des entrées sorties induit par « detrend » sans lequel il y a, par ailleurs, échec complet de la procédure d'identification.

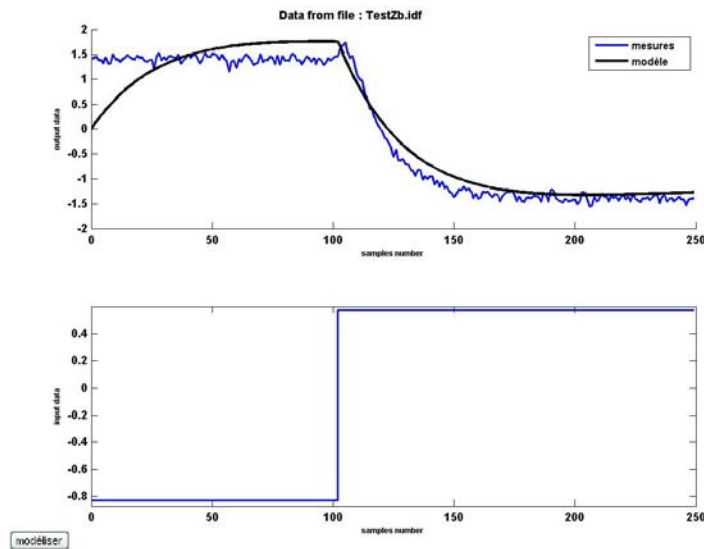


Figure 5. méthode « oe »

10. Conclusion

Le choix paramétrique « pôles » exclut toute approche du type erreur de modèle linéaire dans les paramètres. En revanche, l'optimisation non linéaire induite est plus robuste que celle opérant sur les coefficients du polynôme, car il est trivial d'imposer des contraintes linéaires (bornes) garantissant la stabilité dynamique du modèle identifié. Par ailleurs, cette démarche évite l'usage de solveurs de polynômes. Enfin et surtout, la précision sur les pôles, directement en rapport avec le niveau d'erreur résiduelle de modélisation, exprime véritablement une confiance sur la dynamique identifiée, ce qu'aucune autre méthode n'offre. Ces bonnes propriétés ont bien sûr un coût, au moins en développement algorithmique. En effet si on initialise la recherche de la structure d'une façon erronée, par exemple on déclare deux pôles réels alors qu'il faudrait choisir une paire de pôles complexe conjuguée, l'optimisation conduit à quasiment deux pôles identiques, représentatifs de la partie réelle d'un complexe dont la partie imaginaire aurait été imposée nulle. Cette situation est détectée automatiquement dans l'environnement développé et la procédure d'optimisation commute sur la structure ad hoc à degré identique du dénominateur.

Cette approche devrait néanmoins permettre d'aborder sous un nouvel angle les stratégies de détermination automatique de la structure d'un transfert, qui, jusqu'à présent, n'ont pas conduit au succès espéré. Soulignons que cette technique s'applique à toutes les configurations de modèles dynamiques linéaires stationnaires, mono ou multivariables et offre toute la souplesse habituellement nécessaire au traitement de données industrielles (contrainte éventuelle sur le gain statique, offset, intégrateur, données aberrantes, etc...). La technique de validation, avec ré-estimation des conditions initiales, permet de s'affranchir de toute variation du point de fonctionnement. Enfin, les informations, rarement évaluées, concernant le conditionnement des problèmes d'optimisation, viennent compléter significativement les indications habituelles portant sur les écarts types, et les différents critères d'Akaike (Ljung, 1999) (AIC et FPE notamment) relatives à l'appréciation de la complexité du modèle.

12. Bibliographie

- Barham R., Drane W, "An algorithm for least squares estimation of nonlinear parameters when some of the parameters are linear", *Technometrics*, 14, 757-766, 1972.
- Bjorck A, "Numerical method for least squares problems", Siam, 1996.
- Brent R., "Minimization without derivatives", Prentice hall, 1973.
- Coleman T., Li Y., "An interior trust region approach for non linear minimisation subject to bounds", *Siam Journal on optimisation*, 6, 418-445, 1996.
- Guidorzi R.P., "Invariants and Canonical Forms for Systems and Structural Parameter Identification", *Automatica*, 17, 117-133, 1981.
- Lawson C, Hanson R. "Solving least squares problems", Prentice Hall, 1974.
- Lesecq S., Gentil S., Barraud A. «Diagnostic Symptom Generation Aggregating Model Validity and Residual Evaluation», ECC, Seville, Dec. 2005.
- Ljung L. "System Identification – Theory for the user", Prentice Hall , 2nd edition, 1999.
- Overschee P., De Moor B. "N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems", *Automatica*, 30, 75-93, 1994.
- Powell M. "A factorized quasi Newton algorithm", Report CSS 15, AERE, Harwell, 1975.
- Walter E., Pronzato L., « Identification de modèles paramétriques à partir de données expérimentales », Masson, 1994.
- Zeng Z, "What are the roots of the Wilkinson polynomial?", Midwest Numerical Analysis Day", May 12, 2001.