

Identification de modèle ARMA à sortie quantifiée par la méthode de maximum de vraisemblances

Boubakeur LATRECHE ^a *, Lahcène MITICHE ^a

^a Département des sciences et de technologiques, Université de Djelfa, 17000, Algérie

ARTICLE INFO

Article history :

Received September 2014

Accepted January 2015

Keywords :

Identification ;

Modèle ARMA ;

Estimation ;

Quantification ;

Maximum de vraisemblance.

ABSTRACT

Dans cet article nous présentons une étude détaillée sur une méthode d'identification paramétrique de modèle ARMA (Autorégressive à moyenne ajusté) dans le cas où la sortie est quantifiée, nous allons utiliser la méthode de maximum de vraisemblance récursive pour estimer les paramètres de ce modèle (ARMA). Cette méthode est décomposée sur deux parties, l'algorithme d'expectation maximisation qui est connu d'essuyer d'un faible taux de convergence. A cette, l'estimation obtenue après quelques itérations EM peut être utilisée pour initialiser une méthode de quasi-Newton. La méthode de maximum de vraisemblance est basée sur la minimisation d'un critère quadratique. La simulation est faite sous le logiciel MATLAB

©2015 LESI. All right reserved.

1. Introduction

Depuis les années 60, le filtrage adaptatif a suscité un engouement et un développement sans précédent. Ce développement du filtrage adaptatif est né de l'essor du traitement numérique, de la croissance soutenue de la puissance des processeurs de traitement qui permettent la mise en œuvre en temps réel, d'algorithmes de plus en plus complexes et qui vont à des cadences de plus en plus élevées [1].

Suite à cet essor du filtrage adaptatif, de nombreuses applications telles que : l'identification des systèmes [2][3], l'égalisation de canaux, la modélisation des systèmes [4], la résolution de problème inverse [5] ont vu le jour. Des applications mentionnées plus haut, certaines comme l'identification des systèmes, sont un problème important dans des domaines aussi variés que : le traitement numérique du signal [6], le contrôle de procédé [7], ou encore les communications. L'identification de système est réalisée par l'ajustement de paramètres pour un modèle donné jusqu'à ce que ses sorties, pour une entrée particulière, coïncident autant que possible avec la sortie mesurée du système identifié pour la même entrée.

*Email : latreche_bakar@yahoo.fr

L'identification est aujourd'hui une composante fondamentale des sciences de l'ingénieur, Situé au traitement du signal numérique, cette discipline scientifique est liée au développement des moyens et des techniques de télécommunications [8], traitement de l'information [9], le codage et le décodage (compression) [10], la conversion analogique/numérique (CAN). Le développement continu de ces techniques à aider les chercheurs à mieux comprendre l'identification si les sorties (observations) et/ou les entrées sont quantifiées [3].

Notre travail consiste à l'identification d'un modèle ARMA (Autorégressive à Moyenne Ajustée) [11] à sortie quantifiée, en titre d'exemple, soit un signal d'entrée $u(t)$ est modélisé par un signal déterministe ou un processus aléatoire qui excité le modèle ARMA, la sortie est bruité par un bruit qui est une séquence indépendantes et identiquement distribuées (IID) de distribution normale $N(0, \sigma^2)$. La sortie bruyante $y(t)$ est quantifiée avec un quantificateur scalaire de Lloyd-Max ([Lloyd, 1957] [Max, 1960]) Q_t de K niveau variant dans le temps t .

Notre objectif est d'estimer les paramètres du modèle ARMA ? Nous le faisons en utilisant le critère du maximum de vraisemblance.

2. Principe de l'identification

L'identification consiste à déterminer les paramètres d'un modèle mathématique, dont la structure est établie selon un critère donné , les paramètres des modèles sont obtenus par la minimisation de l'erreur de prédiction entre le signal de sortie mesuré et le signal estimé suivant un critère d'optimalité par exemple : (moindres carrés , erreur quadratique moyenne, maximum de vraisemblance), nous nous intéressons plus particulièrement à la méthode qui est basée sur le blanchissement de l'erreur de prédiction. Formellement l'opération d'identification des paramètres du modèle peut se résumer par la figure ci-dessous

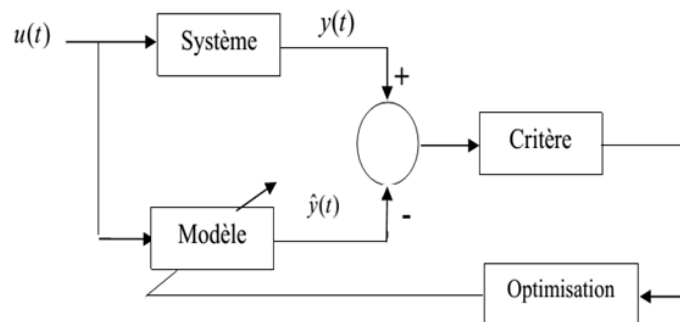


Fig. 1. Schéma d'identification paramétrique.

3. Identification du modèle ARMA

Considérons le modèle ARMA suivante :

$$x(t) = \frac{B(z)}{A(z)}u(t)$$

$$y(t) = x(t) + w(t)$$

$$Z(t) = Q_t [y(t)]$$

Avec :

$u(t)$ est le signal d'entrée modélisé par un signal déterministe ou un processus aléatoire, $A(z)$ et la partie autorégressive (AR) :

$$A(z) = \sum_{i=1}^n a_i z^{-i}, (a_0 = 1);$$

$B(z)$ et la partie moyenne ajusté (MA) :

$$B(z) = \sum_{i=0}^m b_i z^{-i}$$

$w(t)$ est une séquence indépendantes et identiquement distribuées (IID) de distribution normale $N(0, \sigma_2)$. La sortie bruyante $y(t)$ est quantifiée avec un quantificateur scalaire Q_t de K niveau variant dans le temps t

$$Q_t : R \rightarrow \{v_{t,1}, \dots, v_{t,k}\}, t \in Z$$

Défini par des pas de quantification :

$$[b_{t,k-1}, b_{t,k}] = Q^{-1} [v_{t,k}]; k = 1, \dots, k.$$

Avec $b_{t,0} = -\infty$ et $b_{t,k} = +\infty$ pour $t \in Z$.

Notre objectif est d'estimer θ_* , avec la connaissance de U_N et Z_N . Nous le faisons en utilisant la méthode du maximum de vraisemblance.

4. La méthode du maximum de vraisemblance

En utilisant le critère du maximum de vraisemblance [14] [15], la valeur estimé à temps N est obtenue par

$$\hat{\theta}_N \in \arg \max_{\theta} p_{\theta}(Z_N | U_N) = \arg \max_{\theta} l(\theta | U_N, Z_N), \tag{1}$$

Avec p_{θ} désigne la distribution de probabilité donnée, et

$$l(\theta | U_N, Z_N) = \log p_{\theta}(Z_N | U_N), \tag{2}$$

Est la fonction de log-likelihood de θ qui nous donne U_N et Z_N .

Pour résoudre Eq.(1), on utilise un algorithme récursif basé sur la méthode Expectation Maximisation (EM) [16]. L'avantage de cette méthode est que l'on peut dériver un algorithme qui ne nécessite pas une initialisation. Cependant, la méthode EM est connu d'essayer d'un faible taux de convergence. A cette, l'estimation obtenue après quelques itérations EM peut être utilisé pour initialiser une méthode de quasi-Newton pour recherche de gradient. Ainsi, on utilise un algorithme récursif basé sur la méthode quasi-Newton. Le but de l'utilisation de la méthode EM est de fournir une bonne estimation initiale pour la méthode quasi-Newton.

4.1. La méthode d'Expectation Maximisation(EM) récursive

Avant de dériver la méthode requise, nous donnons une brève explication intuitive de la méthode EM. Pour une présentation détaillée, voir [15]. Un inconvénient d'Eq. (1), est que $\log p_\theta(Z_N)$ est difficile à maximiser par rapport à en raison de l'induction d'un quantificateur non linéaire.

Supposons que la sortie $Y_N = \{y(t) : t = 1, \dots, N\}$ avant le quantificateur était connue. Notez que ce serait beaucoup plus facile de maximiser $\log p_\theta(Y_N)$. Tell que Y_N est indisponible, nous remplaçons $\log p_\theta(Z_N)$ par la moyenne de $\log p_\theta(Z_N, Y_N) = \log p_\theta(Z_N|Y_N) + \log p_\theta(Y_N)$ sur toutes les valeurs possibles de Y_N . Pour ce faire, nous utilisons la moyenne conditionnelle de distribution $p_{\hat{\theta}}(Y_N|Z_N)$ d' Y_N compte tenu des observations Z_N , et des estimations précédente $\hat{\theta}$ de θ . Cela conduit à la méthode EM, ce qui résout le problème d'Eq. (1), en utilisant la procédure itérative suivante :

$$\hat{\theta}_N^{(i)} \in \arg \max_{\theta} Q_N(\theta, \theta_N^{(i-1)}), \tag{3}$$

$$Q_N(\theta, \hat{\theta}) = \int \log p_\theta(Z_N, Y_N) p_{\hat{\theta}}(Y_N|Z_N) dY_N. \tag{4}$$

Les itérations (3) et (4) permettent de calculer $\hat{\theta}_N$, pour un nombre fixe de N . Pour obtenir un algorithme récursif, nous pouvons calculer une itération pour chaque nouvel échantillon disponible. Ce faisant, nous obtenons l'algorithme itératif suivant :

$$\hat{\theta}_N \in \arg \max_{\theta} Q_N(\theta, \hat{\theta}_{N-1}) \tag{5}$$

Lemme1

La fonction $Q_N(., .)$ dans (4) est donnée par :

$$Q_N(\theta, \hat{\theta}) = -\frac{N}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{t=1}^N (\bar{y}(t, \hat{\theta}) - x(t, \theta))^2 - \frac{1}{2\sigma^2} \sum_{t=1}^N (\overline{y^2}(t, \hat{\theta}) - \bar{y}^2(t, \hat{\theta})), \tag{6}$$

Ou

$$\overline{y^2}(t, \theta) = \varepsilon_\theta \{y^2(t) | z(t)\} \tag{7}$$

Et

$$\bar{y}(t, \theta) = \varepsilon_{\theta} \{y(t) | z(t)\} \tag{8}$$

Si a et b sont des variables aléatoires, et $f(\cdot)$ est une fonction, expectation conditionnel $\varepsilon_{\theta} \{y(t) | z(t)\}$ de $f(a)$ qui nous donne b est définie par :

$$\varepsilon_{\theta} \{f(a) | b\} = \int f(a) p_{\theta} \{a | b\} da.$$

Remarque : Notez que (6) peut être calculé en

$$\begin{aligned} \bar{y}(t, \theta) &= \int_{Q_t^{-1}[z(t)]} y(t) p_{\theta}(y(t)) dy(t) \\ &= \frac{1}{2} \left[erf \left(\frac{b(t) - x(t, \theta)}{\sqrt{2\sigma^2}} \right) - erf \left(\frac{a(t) - x(t, \theta)}{\sqrt{2\sigma^2}} \right) \right] \end{aligned}$$

Où

$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$ désigne la fonction d'erreur,

Et

$[a(t), b(t)] = Q_t^{-1}[z(t)]$ désigne le pas de quantification correspondant à l'instant (t) .

En combinant (5) avec le lemme 1, on obtient que :

$$\hat{\theta} \in \arg \min_{\theta} \sum_{t=1}^N (\bar{y}(t, \hat{\theta}_{N-1}) - x(t, \theta))^2. \tag{9}$$

$$\hat{\theta} \in \arg \min_{\theta} \sum_{t=1}^N \left(A(z, \theta) \frac{\bar{y}(t, \hat{\theta}_{N-1})}{A(z, \hat{\theta}_{N-1})} - B(z, \theta) \frac{u(t)}{A(z, \hat{\theta}_{N-1})} \right)^2.$$

Après, on calcule $\hat{\theta}_N$ comme suivant :

$$\hat{\theta}_N = \hat{\theta}_{N-1} + L_N (\tilde{y}(N, \hat{\theta}_{N-1}) - \tilde{\varphi}^T(N, \hat{\theta}_{N-1}) \hat{\theta}_{N-1}) \tag{10}$$

Avec

$$\tilde{y}(t, \theta) = \frac{\bar{y}(t, \theta)}{A(z, \theta)}, \tag{11}$$

$$\tilde{\varphi}(t, \theta) = \frac{1}{A(z, \theta)} [u(t), \dots, u(t-n), -\bar{y}(t-1, \theta), \dots, -\bar{y}(t-m, \theta)]^T, \tag{12}$$

$$L_N = \frac{P_{N-1} \tilde{\varphi}(N, \hat{\theta}_{N-1})}{1 + \tilde{\varphi}^T(N, \hat{\theta}_{N-1}) P_{N-1} \tilde{\varphi}(N, \hat{\theta}_{N-1})} \tag{13}$$

$$P_N = P_{N-1} - \frac{P_{N-1} \tilde{\varphi}(N, \hat{\theta}_{N-1}) \tilde{\varphi}^T(N, \hat{\theta}_{N-1}) P_{N-1}}{1 + \tilde{\varphi}^T(N, \hat{\theta}_{N-1}) P_{N-1} \tilde{\varphi}(N, \hat{\theta}_{N-1})} \tag{14}$$

4.2. La méthode de quasi-Newton récursive

Pour un nombre fixe de N , le problème Eq. (1), peut être résolu en utilisant un algorithme de recherche de gradient. A cet effet, nous utilisons la méthode de quasi-Newton. Comme ci-dessus, pour obtenir un algorithme récursif, nous calculons une itération quasi-Newton pour chaque nouvel échantillon disponible. Cela donne les itérations suivantes :

$$\hat{\theta}_{N+1} = \hat{\theta}_N - \mu_N T_N g_N \tag{15}$$

où le scalaire μ_N désigne le pas de taille à l'itération N , la matrice T_N dénote une approximation de l'inverse de la Hessian de $l(\theta | Z_N)$ à $\hat{\theta}_N$, et le vecteur g_N désigne le gradient de $l(\theta | Z_N)$ à $\hat{\theta}_N$, c'est à dire ,

$$g_N = \frac{\partial}{\partial \theta} l(\theta | Z_N) |_{\hat{\theta}_N} \tag{16}$$

Nous avons choisi de calculer T_N en utilisant la formule de Broyden-Fletcher-Goldfarb-Shanno (BFGS) [13], qui est initialisé par $T_{N_0} = I$ (N_0 désigne le premier échantillon après les itérations EM passer à itérations quasi-Newton) et se déroule comme suit :

$$T_{N+1} = T_N + \left(1 + \frac{q_N^T T_N q_N}{s_N^T q_N} \right) \frac{s_N s_N^T}{s_N^T q_N} - \frac{s_N q_N^T T_N + T_N q_N s_N^T}{s_N^T q_N} \tag{17}$$

Avec $s_N = \theta_{N+1} - \theta_N$ et $q_N = g_{N+1} - g_N$.

De plus, le paramètre μ_N est obtenu à partir d'un algorithme de recherche récursive. Nous utilisons l'algorithme de retour arrière est décrit dans [17], qui est formée par les sous itérations des itérations principales Eq. 13, Soit $\alpha = 0.01$ et $\beta = 0.5$. Au sous- itération, le pas μ_N est mis à jour en utilisant $\mu_{N,i} = \beta \mu_{N,i-1}$, à partir de la valeur initiale de $\mu_{N,i} = 1$. Les sous-itérations sont arrêtées lorsque

$$l\left(\hat{\theta}_N - \mu_{N,i} T_N g_N \mid Z_N\right) < l\left(\hat{\theta}_N \mid Z_N\right) + \alpha \mu_{N,i} g_N^T s_N \tag{18}$$

Pour implémenter les itérations Eq.(13), nous devons fournir des expressions pour $l(\theta | Z_N)$ et son gradient ;

$$l(\theta | Z_N) = \sum_{t=1}^N \log p_{\theta}(z(t)) \tag{19}$$

Lemme 2

Le gradient de la fonction de log-likelihood $l(\theta | Z_N)$ est donnée par :

$$\frac{\partial}{\partial \theta} l(\theta | Z_N) = \frac{1}{\sigma^2} \sum_{t=1}^N (\bar{y}(t, \theta) - x(t, \theta)) \dot{x}(t, \theta)$$

Où

$$\dot{x}(t, \theta) \hat{=} \frac{\partial}{\partial \theta} x(t, \theta) = \varphi(z, \theta)u(t) \tag{20}$$

$$\varphi(z, \theta) = \left[\frac{\Omega_n^T(z)}{A(z, \theta)}, \frac{z^{-1}B(z, \theta)\Omega_{m-1}^T(z)}{A^2(z, \theta)} \right]^T \tag{21}$$

et $\Omega_n(z) = [1, z^{-1}, \dots, z^{-n}]^T$.

5. Mise en œuvre de l’algorithme d’identification

L’algorithme d’identification décrit ci-dessus peut être résumé dans ce paragraphe. Choisissez une θ_0 estimation initiale de sorte que $B(z, \hat{\theta}_0) = A(z, \hat{\theta}_0) = 1$, et choisir un nombre N_0 pour que l’algorithme d’estimation récursive passe de la méthode fondée sur EM à la méthode fondée sur quasi-Newton. Puis :

Si $N < N_0$:

- Utilisation de Eq.(6), pour calculer $\bar{y}(N, \hat{\theta}_{N-1})$;
- Utilisation de Eq.(9), et Eq.(10), pour calculer $\tilde{y}(N, \hat{\theta}_{N-1})$ et $\tilde{\phi}(N, \hat{\theta}_{N-1})$;
- Utilisation de Eq.(8), Eq.(11), et Eq.(12), pour calculer $\hat{\theta}_N$

Si non, si $N > N_0$, définissons la valeur initiale des itérations Eq.(15), comme $T_{N_0} = I$.

Puis :

- Usage Eq.(14), et le lemme 2 pour calculer le de gradient g_N ;
- Usage Eq.(16), et Eq.(17), pour calculer μ_N ;
- Usage Eq.(13), calculer $\hat{\theta}_N$;
- Usage Eq.(15), calculer T_N .

6. Exemples de simulation

Considérons le système physique stable ayant la fonction de transfert suivante :

$$H(z) = \frac{B(z)}{A(z)} = \frac{1 + 0.62z^{-1}}{1 + 0.73z^{-1} - 0.24z^{-2}}$$

Avec : $b_0 = 0; b_1 = 1; b_2 = 0.62; a_0 = 1; a_1 = 0.73; a_2 = -0.24$.

L’implémentation de l’algorithme de maximum de vraisemblance sur PC est donnée les résultats suivantes :

Tableau 1. Influence du nombre d’échantillons N, ($\sigma^2 = 0.25$).

| Nombre d’échantillons | \hat{a}_1 | \hat{a}_2 | \hat{b}_1 | \hat{b}_2 | Moyenne du Biais | Moyenne de l’EQM |
|-----------------------|-------------|-------------|-------------|-------------|------------------|------------------|
| 256 | 0.6988 | -0.2556 | 0.9745 | 0.5879 | 0.0849 | 0.0278 |
| 512 | 0.7104 | -0.2450 | 0.9854 | 0.6157 | 0.0476 | 0.0140 |
| 1024 | 0.7238 | -0.2449 | 1.0023 | 0.6143 | 0.0276 | 0.0070 |
| 2048 | 0.7272 | -0.2393 | 1.0007 | 0.6195 | 0.0149 | 0.0035 |
| 4096 | 0.7284 | -0.2410 | 1.0007 | 0.6167 | 0.0089 | 0.0018 |

Tableau 2. Influence de la variance σ^2 , ($N=1024$).

| Variance (σ^2) | \hat{a}_1 | \hat{a}_2 | \hat{b}_1 | \hat{b}_2 | Moyenne du Biais | Moyenne de l'EQM |
|-------------------------|-------------|-------------|-------------|-------------|------------------|------------------|
| 1 | 0.7094 | -0.2667 | 1.0116 | 0.6039 | 0.0501 | 0.0117 |
| 0.25 | 0.7238 | -0.2449 | 1.0023 | 0.6143 | 0.0276 | 0.0070 |
| 0.04 | 0.7265 | -0.2416 | 0.9995 | 0.6161 | 0.0196 | 0.0060 |
| 0.01 | 0.7274 | -0.2414 | 0.9992 | 0.6169 | 0.0181 | 0.0056 |

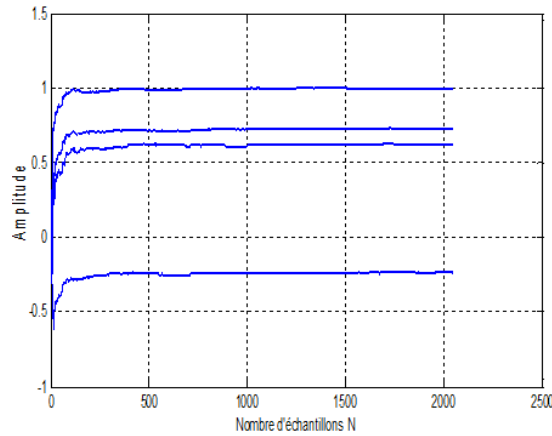


Fig. 2. Evolution des paramètres ($N = 2048$), ($\sigma^2 = 0.25$)

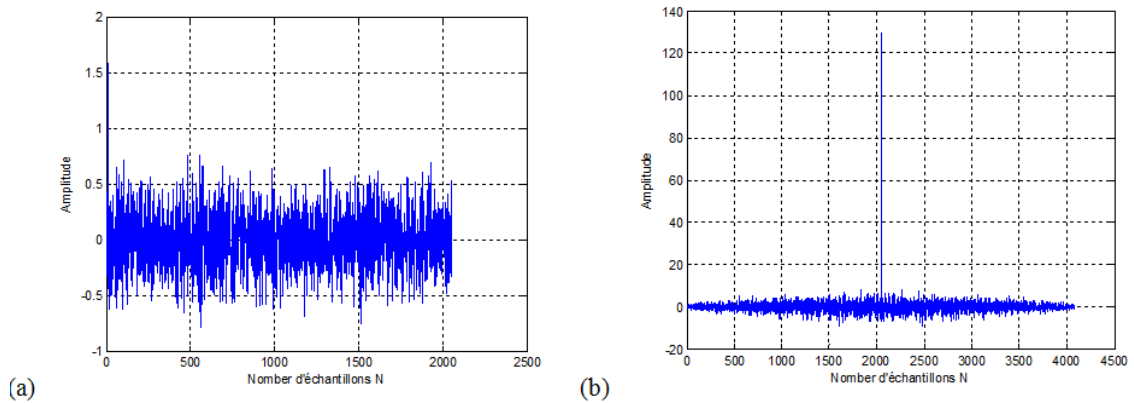


Fig. 3. L'illustration (a) représente l'erreur de prédiction et (b) sa fonction d'autocorrélation.

7. Conclusion

D'après les résultats de simulation, la méthode de maximum de vraisemblance qui a été présentée en détail pour les observations quantifiées, donne des estimations non biaisées, si la valeur moyenne du bruit est nulle, si non est d'estimation biaisée. Les valeurs estimées sont plus proches aux valeurs réelles tant que le nombre d'échantillons plus grands. D'autre part l'erreur de prédiction est asymptotiquement blanche.

REFERENCES

- [1] F. Mi chaut, M. Bellanger, *Filtrage adaptif, théorie et algorithmes*, Paris : Lavoisier, 2005.
- [2] L. Ljung, *System Identification : Theory for the User*, Second edition, Englewood Cliffs, NJ : Prentice-Hall, 1999.
- [3] L. Y. Wang, G. G. Yin, J. F. Zhang, Y. Zhao, *System Identification with Quantized Observations*, Birkhauser, Boston, 2010.
- [4] F. Van Der Heijden, R. P. W. Duin, D. de Ridder, *Parameter Estimation and State Estimation, An Engineering Approach using MATLAB*, Wiley, 2004.
- [5] Y-P. Sheng, L-S. Wang, *The Least Squares Solutions of Inverse Problem for Symmetric Orthogonal Anti-symmetric Matrices*, IEEE Conference on Machine learning and Cybematics, pp. 631-634, 2006.
- [6] T. Soderstrom, P. Stoica, *On some system identification techniques for adaptive filtering*, IEEE Trans. Circuits Syst, vol. 35, pp. 457-461, 1988.
- [7] D. R. Coughanowr, *Process systems analysis and control*, McGraw-Hill, Inc. New York, 1991.
- [8] J. Frang, H. Li, *Distributed adaptive quantization for wireless sensor networks : From delta modulation to maximum likelihood*, IEEE Transaction on Signal Processing, 56(10) : 5246-5257, 2008.
- [9] H. C. Papadopoulos, G .W. Wornell, A. V. Oppenheim, *Sequential signal encoding from noisy measurements using quantizers with dynamic bias control*, IEEE Transactions on Information Theory, 47(3) : 978-1002, 2002.
- [10] J. Xiao, A. Ribeiro, Z. Luo, G. Giannakis, *Ditributed compression-estimation izing wireless sensor networks*, IEEE Signal Processing Magazine, 23(4) : 27-41, 2006.
- [11] B. Choi, *ARMA Model Identification*, Springer, New York, 1992.
- [12] B. Picinbono, *Signaux aléatoires et bases du traitement statistique du signal*, Tome3, Dunod, Paris, 1993.
- [13] R. Fletcher, *Partial Methods of optimization*, Jhon Wiley et Sons, Inc, 1987.
- [14] J. Schoukens, R. Pintelon, Y. Rolain, *Mastering System Identification in 100 Exercises*, IEEE Pres et Wiley, Canada, 2012.
- [15] G. J. Mclachlan, T. Krishnan, *The EM Algorithm and Extensions*, Wiley-Interscience, 2 edition, 2008.
- [16] D. Cappé, E. Moulines, T. Rydén, *Inference in Hidden Markov Models*, Springer, 2005.
- [17] S. Boyd, L. Vandenberghe, *Convex optimisation*, Cambridge Univ, 2004.