

# Algorithme d'identification à erreur de sortie pour systèmes à temps continu en boucle fermée à partir de données échantillonnées

Arnaud BAYSSE, Francisco Javier CARRILLO, Abdallah HABBADI

Université de Toulouse, INP/ENIT, LGP,  
47 av. d'Azereix BP 1629 - 65016 Tarbes Cedex  
*arnaud.baysse@enit.fr, carrillo@enit.fr, habbadi@enit.fr*

*Résumé*— Cet article présente un algorithme d'identification hors ligne à erreur de sortie pour systèmes à temps continu en boucle fermée. Cet algorithme peut être considéré comme la version à temps continu de la méthode F-CLOE (Filtered Closed Loop Output Error) de Landau [6] pour les systèmes à temps discret. Des simulations et l'application de la méthode proposée sur un système expérimental sont aussi présentées.

*Mots-clés*— identification en boucle fermée, systèmes à temps continu, identification à erreur de sortie, CLOE, Moindres Carrés, Variable Instrumentale, système de régulation de niveau

## I. INTRODUCTION

L'identification des systèmes à temps continu est un sujet d'intérêt récent présentant de nombreux avantages. En effet, l'utilisation de modèles à temps continu permet un choix relativement indépendant de la période d'échantillonnage, et d'obtenir des paramètres estimés caractéristiques d'une réalité physique. Ainsi, un livre [4] et des articles récents [3], [5] montrent l'intérêt porté dans le domaine de l'identification de systèmes à temps continu. Il existe aussi une boîte à outils Matlab appelée CONTSID [2] (CONTinuous System IDentification) permettant l'identification de systèmes en boucle ouverte.

Dans le cadre de l'identification à erreur de sortie de systèmes en boucle fermée, un ensemble d'algorithmes, appelé CLOE (Closed Loop Output Error) est intéressant car ils permettent d'obtenir des paramètres non biaisés sous certaines conditions de convergence. Ces algorithmes, proposés par Landau [6], sont utilisés sur des systèmes à temps discret, et basés sur la théorie de la passivité pour assurer la convergence.

Dans cet article nous proposons une version continue de l'algorithme F-CLOE (Filtered Closed Loop Output Error). L'idée principale de cet article est inspirée de l'analyse de convergence proposée dans [10], utilisant les fonctions de sensibilités pour des algorithmes à erreur de sortie.

A la section II l'algorithme est décrit. Ensuite, à la section III la méthode proposée est appliquée à un système de régulation de niveau de liquide, en utilisant des simulations et des expérimentations. Enfin, la section IV conclut en résumant les résultats obtenus.

## II. DESCRIPTION DE L'ALGORITHME

Considérons un système de sortie  $y$ , d'entrée  $u$  et de bruit de sortie  $w$  tel que :

$$A(p)y(t) = B(p)u(t) + A(p)w(t) \quad (1)$$

Avec  $p$  l'opérateur de dérivation  $p = \frac{d}{dt}$ ;  $A(p)$  et  $B(p)$  des polynômes définis comme suit :

$$\begin{aligned} A(p) &= 1 + a_1p + \dots + a_np^n \\ B(p) &= b_0 + b_1p + \dots + b_mp^m; \quad m \leq n \end{aligned}$$

Le système fonctionne en boucle fermée, avec un régulateur linéaire à temps continu et à paramètres fixes  $C(p)$  décrit comme (cf. figure 1) :

$$u(t) = C(p)[r(t) - y(t)]; \quad C(p) = \frac{R(p)}{S(p)} \quad (2)$$

Avec  $r(t)$  la consigne en boucle fermée. On suppose que les signaux  $r(t)$ ,  $u(t)$ , et  $y(t)$  sont uniformément échantillonnés à la période d'échantillonnage  $T_s$ . On suppose de plus que la consigne  $r(t)$  et le régulateur  $C(p)$  sont connus.

Le problème de l'identification de modèle à temps continu en boucle fermée consiste à trouver un estimé du modèle  $\frac{B(p)}{A(p)}$  à partir de données discrètes  $\{u(t_k)\}_{k=1}^N$ ,  $\{y(t_k)\}_{k=1}^N$ ,  $\{r(t_k)\}_{k=1}^N$ , avec  $N$  le nombre de données utilisées et  $t_k = kT_s$ . On peut alors définir le modèle par les équations hybrides suivantes :

$$A(p)y(t_k) = B(p)u(t_k) + A(p)w(t_k) \quad (3)$$

Dans ce cas, on peut considérer  $w(t_k)$  comme un bruit blanc à temps discret, de moyenne nulle et de variance finie; on peut consulter [4] pour plus de détails sur cette hypothèse.

Dans la suite de l'article, pour faciliter la présentation, les signaux sont considérés en fonction du temps  $t$ , mais dans un but d'identification les signaux seront considérés à un temps d'échantillonnage  $t = t_k$ .

En utilisant l'approche à erreur de sortie, la sortie estimée est donnée par (voir la figure 1) :

$$\hat{y}(t) = \frac{\hat{B}(p)}{\hat{A}(p)}\hat{u}(t); \quad \hat{u}(t) = C(p)[r(t) - \hat{y}(t)] \quad (4)$$

La sortie estimée peut s'écrire sous la forme suivante :

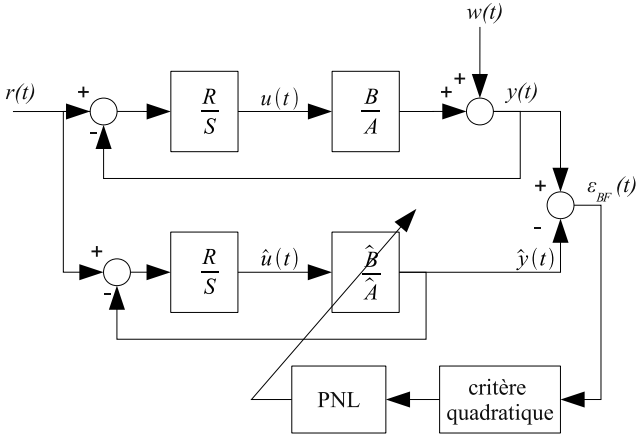


Fig. 1: Approche d'identification à erreur de sortie

$$\begin{aligned}
\hat{y}(t) &= -\hat{a}_1 p \hat{y}(t) - \hat{a}_2 p^2 \hat{y}(t) - \dots - \hat{a}_n p^n \hat{y}(t) \\
&\quad + \hat{b}_0 \hat{u}(t) + \hat{b}_1 p \hat{u}(t) + \dots + \hat{b}_m p^m \hat{u}(t) \\
&= \hat{\theta}^T \hat{\phi}(t); \\
\hat{\theta}^T &= \begin{bmatrix} \hat{a}_1 & \hat{a}_2 & \dots & \hat{a}_n & \hat{b}_0 & \hat{b}_1 & \dots & \hat{b}_m \end{bmatrix}; \\
\hat{\phi}^T(t) &= \begin{bmatrix} -p \hat{y}(t) & -p^2 \hat{y}(t) & \dots & -p^n \hat{y}(t) \\ \hat{u}(t) & p \hat{u}(t) & \dots & p^m \hat{u}(t) \end{bmatrix}
\end{aligned}$$

Avec  $\hat{\theta}$  le vecteur des paramètres estimés et  $\hat{\phi}(t)$  le vecteur des observations.

L'approche à erreur de sortie, illustrée sur la figure 1, consiste à estimer les paramètres du modèle en minimisant l'erreur de sortie  $\varepsilon_{BF}(t) = y(t) - \hat{y}(t)$  par Programmation Non Linéaire (PNL).

#### A. Algorithme d'adaptation paramétrique hors-ligne

Afin d'identifier le système à partir de la méthode hors-ligne, la sortie du modèle doit être estimée pour une période d'échantillonnage donnée :  $\hat{y}_k = f(\hat{\theta}, \hat{u}_k, \hat{y}_k)$ . Ce signal est le résultat d'une simulation d'un modèle non linéaire dans les paramètres, connaissant  $\hat{u}_k$  et  $\hat{\theta}$ . Le critère quadratique suivant est alors minimisé :

$$J = \sum_{k=1}^N (y(t_k) - \hat{y}(t_k))^2 = \sum_{k=1}^N \varepsilon_{BF}^2(t_k) \quad (5)$$

La minimisation de ce critère est effectuée par programmation non-linéaire. Les trois algorithmes principaux utilisés sont l'algorithme du gradient, l'algorithme de Gauss-Newton et enfin l'algorithme de Levenberg-Marquardt, décrits respectivement par :

$$\begin{aligned}
\hat{\theta}_{j+1} &= \hat{\theta}_j - J'_\theta \Big|_{\hat{\theta}=\hat{\theta}_j} = \hat{\theta}_j - \frac{\partial J}{\partial \hat{\theta}} \Big|_{\hat{\theta}=\hat{\theta}_j} \\
\hat{\theta}_{j+1} &= \hat{\theta}_j - \left[ J''_\theta \right]^{-1} J'_\theta \Big|_{\hat{\theta}=\hat{\theta}_j} \\
\hat{\theta}_{j+1} &= \hat{\theta}_j - \left\{ \left[ J''_\theta + \mu I \right]^{-1} J'_\theta \right\} \Big|_{\hat{\theta}=\hat{\theta}_j} \quad (6)
\end{aligned}$$

Avec  $j$  le pas d'itération,  $J'_\theta$  et  $J''_\theta$  respectivement le gradient et le hessien du critère  $J$ . Ces deux fonctions sont calculées à partir des fonctions de sensibilité.

Pour un paramètre  $\hat{\theta}_l$ , on définit la fonction de sensibilité par  $\sigma_{\hat{\theta}_l} = \frac{\partial \hat{y}(t)}{\partial \hat{\theta}_l}$ . On peut alors définir le vecteur des fonctions de sensibilité comme

$$\sigma_\theta^T(t) = \begin{bmatrix} \frac{\partial \hat{y}(t)}{\partial \hat{\theta}_1} & \dots & \frac{\partial \hat{y}(t)}{\partial \hat{\theta}_l} & \dots & \frac{\partial \hat{y}(t)}{\partial \hat{\theta}_{m+n+1}} \end{bmatrix} \quad (7)$$

Le gradient et le hessien sont calculés en utilisant les formulations suivantes :

$$J'_\theta = -2 \sum_{k=1}^N \varepsilon_{BF}(t_k) \sigma_\theta^T(t_k); \quad J''_\theta \simeq 2 \sum_{k=1}^N \sigma_\theta^T(t_k) \sigma_\theta^T(t_k) \quad (8)$$

Il est important de noter que pour calculer ces fonctions de sensibilité  $\sigma_\theta^T(t_k)$ , le vecteur des paramètres  $\hat{\theta}$  doit être constant sur l'horizon de minimisation  $N$ . De plus, pour une estimation correcte avec l'algorithme, une bonne initialisation des paramètres est nécessaire [9], [10].

#### B. Les fonctions de sensibilité

Afin d'utiliser l'approche à erreur de sortie, il est nécessaire de calculer les fonctions de sensibilité suivantes pour les paramètres du numérateur :

$$\begin{aligned}
\sigma_{\hat{b}_m} &= \frac{\partial \hat{y}(t)}{\partial \hat{b}_m} = \frac{\partial}{\partial \hat{b}_m} \left[ \frac{\hat{B}(p)}{\hat{A}(p)} \hat{u}(t) \right] \\
&= \frac{\frac{\partial}{\partial \hat{b}_m} \hat{B}(p)}{\hat{A}(p)} \hat{u}(t) + \frac{\hat{B}(p)}{\hat{A}(p)} \frac{\partial}{\partial \hat{b}_m} \hat{u}(t)
\end{aligned}$$

En utilisant l'équation (4), on obtient :

$$\sigma_{\hat{b}_m} = \frac{p^m \hat{u}(t)}{\hat{A}(p)} - \frac{\hat{B}(p) R(p)}{\hat{A}(p) S(p)} \frac{\partial \hat{y}(t)}{\partial \hat{b}_m}$$

Cela donne

$$\sigma_{\hat{b}_m} \left[ 1 + \frac{\hat{B}(p) R(p)}{\hat{A}(p) S(p)} \right] = \frac{p^m \hat{u}(t)}{\hat{A}(p)}$$

et

$$\sigma_{\hat{b}_m} = \frac{p^m \hat{u}(t)}{\hat{A}(p)} \frac{1}{\left( 1 + \frac{\hat{B}(p) R(p)}{\hat{A}(p) S(p)} \right)} = \frac{S(p) p^m \hat{u}(t)}{\hat{A}(p) S(p) + \hat{B}(p) R(p)}$$

$$\sigma_{\hat{b}_m} = \frac{S(p) p^m \hat{u}(t)}{\hat{P}(p)}; \quad (9)$$

$$\hat{P}(p) = \hat{A}(p) S(p) + \hat{B}(p) R(p) \quad (10)$$

Pour obtenir les paramètres du dénominateur, les fonctions de sensibilité suivantes sont calculées :

$$\sigma_{\hat{a}_n} = \frac{\partial \hat{y}(t)}{\partial \hat{a}_n} = \frac{\partial}{\partial \hat{a}_n} \left[ \frac{\hat{B}(p)}{\hat{A}(p)} \hat{u}(t) \right]$$

$$\sigma_{\hat{a}_n} = \frac{-\hat{B}(p)}{\hat{A}^2(p)} p^n \hat{u}(t) + \frac{\hat{B}(p)}{\hat{A}(p)} \frac{\partial}{\partial \hat{a}_n} \hat{u}(t)$$

En prenant en compte l'expression de  $\hat{u}(t)$  donné dans l'équation (4), on obtient le résultat suivant :

$$\sigma_{\hat{a}_n} = -\frac{p^n \hat{y}(t)}{\hat{A}(p)} - \frac{\hat{B}(p) R(p)}{\hat{A}(p) S(p)} \frac{\partial \hat{y}(t)}{\partial \hat{a}_n}$$

alors

$$\sigma_{\hat{a}_n} \left[ 1 + \frac{\hat{B}(p) R(p)}{\hat{A}(p) S(p)} \right] = -\frac{p^n \hat{y}(t)}{\hat{A}(p)}$$

pour obtenir finalement que :

$$\begin{aligned} \sigma_{\hat{a}_n} &= -\frac{p^n \hat{y}(t)}{\hat{A}(p)} \frac{1}{\left(1 + \frac{\hat{B}(p) R(p)}{\hat{A}(p) S(p)}\right)} = \frac{-S(p) p^n \hat{y}(t)}{\hat{A}(p) S(p) + \hat{B}(p) R(p)} \\ \sigma_{\hat{a}_n} &= \frac{-S(p) p^n \hat{y}(t)}{\hat{P}(p)} \end{aligned} \quad (11)$$

Avec  $\hat{P}(p)$  donné par l'équation (10). Les équations (9) et (11) impliquent que les fonctions de sensibilité sont naturellement filtrées par un transfert qui prend en compte le fait que le système est en boucle fermée. On peut alors écrire :

$$\sigma_{\hat{\theta}}(t) = \frac{\partial \hat{y}(t)}{\partial \hat{\theta}} = \frac{S(p)}{\hat{P}(p)} \hat{\phi}(t) \quad (12)$$

### C. Lien avec l'approche F-CLOE

Il a été montré dans un article précédent [1] qu'en posant  $(\theta - \hat{\theta}) = \tilde{\theta}$ , on obtient comme expression de l'erreur en boucle fermée :

$$\varepsilon_{BF} = \frac{S}{P} [\tilde{\theta}^T \hat{\phi} + Aw] = \frac{\hat{P}}{P} \tilde{\theta}^T \sigma + \frac{AS}{P} w \quad (13)$$

L'expression ainsi obtenue est la même que celle obtenue dans le cas discret pour l'algorithme F-CLOE (Filtered Closed-Loop Output Error) proposé par Landau [7]. La différence avec notre approche est que l'équation (13) est maintenant valide pour les modèles à temps continu. De plus, ce résultat a été trouvé en utilisant les fonctions de sensibilité et non l'approche de la passivité. L'analogie entre l'algorithme proposé et l'algorithme F-CLOE permet d'obtenir les mêmes conditions de stabilité. Ainsi, pour obtenir une estimation non biaisée des paramètres en utilisant l'algorithme récursif de Gauss-Newton [1], la condition de convergence est la suivante :

$$Re \left( \frac{\hat{P}}{P} - \frac{1}{2} \right) > 0 \quad (14)$$

On peut remarquer que si  $\hat{P} \approx P$ , la condition de convergence est facilement vérifiée.

## III. APPLICATION À UN SYSTÈME DE RÉGULATION DE NIVEAU DE LIQUIDE

On choisit d'appliquer l'algorithme d'identification proposé sur un système de régulation de niveau. Ce système est un équipement de laboratoire très connu appelé DTS200, "système trois cuves" [11]. Ce système est composé de trois

cuves et de deux pompes. Chaque cuve possède un robinet pour le réglage manuel du débit de sortie. Dans les essais illustrés dans cet article, une seule cuve est utilisée ainsi qu'une seule pompe.

Les équations suivantes décrivent le système, sans capteur ni actionneur :

$$S_c \frac{dH}{dt} = Q_{in} - Q_{out}; \quad Q_{out} = \alpha \sqrt{H} \quad (15)$$

$S_c$  est la surface du cylindre,  $H$  est le niveau d'eau,  $Q_{in}$  et  $Q_{out}$  sont respectivement les débits d'entrée et de sortie, et  $\alpha$  est un coefficient qui représente l'ouverture du robinet.

Ce système est non linéaire, à cause de la racine carrée. Afin de pouvoir tester l'algorithme proposé, nous considérerons un système linéarisé autour d'un point de fonctionnement. En effet, si on utilise de faibles variations de la consigne autour d'un point de fonctionnement, un modèle linéaire peut approcher le fonctionnement du système.

Dans la suite de l'article, des simulations du modèle linéarisé, puis des expérimentations sur le système DTS200 seront présentées.

### A. Linéarisation

En utilisant les deux premiers termes de la série de Taylor, une fonction non linéaire  $f(x)$  peut être approximée par :

$$f(x) \approx f(x_0) + \left. \frac{\partial f}{\partial x} \right|_{x=x_0} (x - x_0) \quad (16)$$

En notant  $Q_{in} = Q_0 + q$ ,  $H = H_0 + h$  et en appliquant le développement limité (16), on obtient :

$$S_c \frac{dh}{dt} = Q_0 + q - \alpha \left( \sqrt{H_0} + \frac{1}{2\sqrt{H_0}} (H_0 + h - H_0) \right)$$

A l'équilibre, on a  $Q_{in} = Q_{out}$ ; ainsi  $Q_0 = \alpha \sqrt{H_0}$  et :

$$S_c \frac{dh}{dt} = q - ah; \quad a = \frac{\alpha}{2\sqrt{H_0}} \quad (17)$$

En utilisant  $p = \frac{d}{dt}$ , la fonction de transfert suivante décrivant le système linéarisé est obtenue :

$$\begin{aligned} h(t) &= F_0(p)q(t); \\ F_0(p) &= \frac{1}{S_c p + a} = \frac{\frac{1}{a}}{1 + \frac{S_c}{a}p} = \frac{K_0}{1 + \tau p} \end{aligned} \quad (18)$$

### B. Modèle complet

Le système utilise une pompe comme actionneur et un capteur résistif piezo-électrique de pression pour mesurer la hauteur de liquide. Comme la dynamique de la cuve est très lente, comparée à celles de l'actionneur et du capteur, ces derniers peuvent être modélisés par des gains  $K_1$  et  $K_2$ .

$$u(t) = K_1 q(t); \quad y(t) = K_2 h(t) \quad (19)$$

En utilisant les équations (18) et (19), le système linéaire complet suivant est obtenu :

$$\begin{cases} y(t) = F(p)u(t); \\ F(p) = \frac{K}{1 + \tau p}; \quad K = K_0 K_1 K_2 \end{cases} \quad (20)$$

### C. Initialisation des paramètres

Afin d'utiliser l'algorithme proposé, il est nécessaire d'initialiser les paramètres estimés. Une bonne initialisation permet une convergence plus rapide vers la solution. L'initialisation peut être effectuée grâce à de nombreuses méthodes. Dans cet article, nous choisissons d'utiliser les méthodes hors lignes des Moindres Carrés (MC) et de la Variable Instrumentale (VI) avec Filtre à Variable d'Etat (FVE) [4]. L'approche FVE est décrite brièvement dans la suite.

On peut écrire le modèle (20) sous la forme régresseur suivante :

$$\dot{y}(t) = -\frac{1}{\tau}y(t) + \frac{K}{\tau}u(t) = \theta^T \phi(t); \quad (21)$$

$$\theta^T = \left[ \frac{1}{\tau} \quad \frac{K}{\tau} \right]; \quad \phi^T(t) = \left[ -y(t) \quad u(t) \right]$$

$\theta$  est le vecteur des paramètres, et  $\phi(t)$  le vecteur de régression. Comme la dérivée de la sortie n'est pas mesurable, elle est recréée en utilisant des filtres. Dans notre cas, on utilise un filtre de premier ordre avec gain statique unitaire ayant la forme :

$$y_f(t) = \frac{\lambda}{p+\lambda}y(t) \quad ; \quad u_f(t) = \frac{\lambda}{p+\lambda}u(t) \quad (22)$$

Les équations (22) nous permettent d'écrire :

$$\dot{y}_f(t) + \lambda y_f(t) = \lambda y(t) \Leftrightarrow \dot{y}_f(t) = \lambda y(t) - \lambda y_f(t) \quad (23)$$

Ainsi la dérivée filtrée de la sortie  $\dot{y}_f(t)$  peut être exprimée comme une combinaison linéaire de la sortie  $y(t)$  et de la sortie filtrée  $y_f(t)$ . En remplaçant dans l'équation (21) les signaux  $y(t)$  et  $u(t)$  par les signaux filtrés  $y_f(t)$  et  $u_f(t)$ , on obtient :

$$\begin{aligned} \dot{y}_f(t) &= -\frac{1}{\tau}y_f(t) + \frac{K}{\tau}u_f(t) \\ \lambda y(t) - \lambda y_f(t) &= -\frac{1}{\tau}y_f(t) + \frac{K}{\tau}u_f(t) \\ Y_f(t) &= \theta^T \phi_f(t) \quad \phi_f^T(t) = \left[ -y_f(t) \quad u_f(t) \right] \end{aligned}$$

$Y_f(t)$  est la dérivée filtrée de la sortie et  $\phi_f(t)$  est le vecteur de régression filtré.

L'algorithme des MC peut ainsi être utilisé pour calculer le vecteur des paramètres estimés :

$$\hat{\theta}_{MC} = \left[ \sum_{k=1}^N \phi_f(t_k) \phi_f^T(t_k) \right]^{-1} \sum_{k=1}^N \phi_f(t_k) Y_f(t_k) \quad (24)$$

### D. La méthode de la variable instrumentale

Dans le but d'initialiser l'algorithme, la méthode de la variable instrumentale est utilisée. Le modèle estimé, sous sa forme régresseur, s'écrit :

$$\begin{aligned} \dot{\hat{y}}(t) &= -\frac{1}{\hat{\tau}}\hat{y}(t) + \frac{\hat{K}}{\hat{\tau}}u(t) = \hat{\theta}_{MC}^T \hat{\phi}(t); \quad (25) \\ \hat{\theta}_{MC}^T &= \left[ \frac{1}{\hat{\tau}} \quad \frac{\hat{K}}{\hat{\tau}} \right]; \quad \hat{\phi}^T(t) = \left[ -\hat{y}(t) \quad u(t) \right] \end{aligned}$$

Avec  $\hat{\theta}_{MC}$  le vecteur des paramètres estimés par la méthode des MC de l'équation (24). Si on utilise le même filtre que celui utilisé dans la section III-C, les signaux filtrés s'écrivent :

$$\hat{y}_f(t) = \frac{\lambda}{p+\lambda}\hat{y}(t) \quad u_f(t) = \frac{\lambda}{p+\lambda}u(t)$$

La sortie estimée filtrée est donnée par :

$$\begin{aligned} \dot{\hat{y}}_f(t) &= -\frac{1}{\hat{\tau}}\hat{y}_f(t) + \frac{\hat{K}}{\hat{\tau}}u_f(t) \\ \lambda \hat{y}(t) - \lambda \hat{y}_f(t) &= -\frac{1}{\hat{\tau}}\hat{y}_f(t) + \frac{\hat{K}}{\hat{\tau}}u_f(t) \\ \hat{Y}_f &= \hat{\theta}_{MC}^T \hat{\phi}_f \quad \hat{\phi}_f^T(t) = \left[ -\hat{y}_f(t) \quad u_f(t) \right] \end{aligned}$$

Comme dans la section III-C, la sortie dérivée est créée en utilisant le filtre proposé. Le vecteur des paramètres  $\hat{\theta}_{VI}$  est alors calculé de la manière suivante :

$$\hat{\theta}_{VI} = \left[ \sum_{k=1}^N \hat{\phi}_f(t_k) \hat{\phi}_f^T(t_k) \right]^{-1} \sum_{k=1}^N \hat{\phi}_f(t_k) Y_f(t_k) \quad (26)$$

L'estimation des paramètres de l'équation (26) est non biaisée en boucle ouverte mais biaisée en boucle fermée à cause de la corrélation entre l'entrée  $u(t)$  et le bruit. Cependant, cette méthode donne une meilleure estimation des paramètres que la méthode des moindres carrés de la section III-C et est utilisée pour initialiser l'algorithme proposé. On peut noter qu'il est possible d'obtenir un estimé non biaisé des paramètres en boucle fermée en utilisant la variable instrumentale [4], [5].

### E. Implémentation de l'algorithme proposé

On choisit d'utiliser l'algorithme hors-ligne avec la PNL de Levenberg-Marquardt, décrit précédemment avec l'équation (6). Ce type d'optimisation utilise le gradient  $J'_\theta$  et le hessien  $J''_\theta$  de la fonction  $J$  à optimiser ainsi qu'un paramètre de convergence  $\mu$ . Au début de l'algorithme, on choisit un  $\mu$  très grand pour pouvoir converger plus rapidement ; on se rapproche alors de la méthode du gradient ( $\mu \gg 1$ ). A l'approche de la solution, on diminue le paramètre  $\mu$  et on se rapproche alors de la méthode de Gauss-Newton ( $\mu \ll 1$ ), qui permet une convergence plus rapide à proximité de l'optimum.

Les paramètres que l'on souhaite estimer peuvent présenter des ordres de grandeur différents, générant une désensibilisation de l'espace paramétrique. Les méthodes de PNL rencontrent alors des difficultés de convergence. Une solution possible consiste en la normalisation des fonctions de sensibilité par un changement de paramètres [8].

Considérons le vecteur de paramètres  $\hat{\theta}^T = \left[ \hat{\theta}_1 \cdots \hat{\theta}_l \cdots \hat{\theta}_{m+n+1} \right]$  où les  $\hat{\theta}_i$ ,  $i = 1, \dots, l, \dots, m+n+1$ , sont les paramètres définis en section II. On peut définir sa valeur initiale  $\hat{\theta}_0^T = \left[ \hat{\theta}_{1,0} \cdots \hat{\theta}_{l,0} \cdots \hat{\theta}_{m+n+1,0} \right]$ , obtenue par une première identification par exemple. Pour un paramètre  $\hat{\theta}_l$ , on peut écrire :

$$\hat{\theta}_l = \hat{\theta}_{l,0} + \Delta \hat{\theta}_{l,0} = \left( 1 + \hat{\delta}_l \right) \hat{\theta}_{l,0} \quad (27)$$

$\Delta\hat{\theta}_l$  est la variation de  $\hat{\theta}_l$  et  $\hat{\delta}_l$  le nouveau paramètre normalisé. La fonction de sensibilité par rapport à  $\hat{\theta}_l$  s'écrit :

$$\sigma_{\hat{\theta}_l}(t) = \frac{\partial\hat{y}(t)}{\partial\hat{\theta}_l} = \frac{\partial\hat{y}(t)}{\partial(\hat{\theta}_{l,0} + \hat{\delta}_l\hat{\theta}_{l,0})} = \frac{1}{\hat{\theta}_{l,0}} \frac{\partial\hat{y}(t)}{\partial\hat{\delta}_l} = \frac{1}{\hat{\theta}_{l,0}} \sigma_{\hat{\delta}_l}(t) \quad (28)$$

On obtient ainsi la nouvelle fonction de sensibilité  $\sigma_{\hat{\delta}_l}(t) = \frac{\partial\hat{y}(t)}{\partial\hat{\delta}_l}$  normalisée. On effectue la même opération (28) sur tous les paramètres. En définissant  $\hat{\delta}^T = [\hat{\delta}_1 \cdots \hat{\delta}_l \cdots \hat{\delta}_{m+n+1}]$  le vecteur des paramètres normalisés, on obtient le vecteur des fonctions de sensibilité normalisées suivant :

$$\sigma_{\hat{\delta}}^T(t) = \begin{bmatrix} \frac{\partial\hat{y}(t)}{\partial\hat{\delta}_1} & \cdots & \frac{\partial\hat{y}(t)}{\partial\hat{\delta}_l} & \cdots & \frac{\partial\hat{y}(t)}{\partial\hat{\delta}_{m+n+1}} \end{bmatrix} \quad (29)$$

On utilise alors l'algorithme de Levenberg-Marquardt pour identifier le vecteur de paramètres  $\hat{\delta}$  en utilisant la forme :

$$\hat{\delta}_{j+1} = \hat{\delta}_j - \left\{ \left[ J_{\hat{\delta}}'' + \mu I \right]^{-1} J_{\hat{\delta}}' \right\}_{\hat{\delta}=\hat{\delta}_j} \quad (30)$$

avec :

$$J_{\hat{\delta}}' = -2 \sum_{k=1}^N \varepsilon_{BF}(t_k) \sigma_{\hat{\delta}}(t_k) ; \quad J_{\hat{\delta}}'' \simeq 2 \sum_{k=1}^N \sigma_{\hat{\delta}}(t_k) \sigma_{\hat{\delta}}^T(t_k) \quad (31)$$

Finalement, le vecteur des paramètres réels  $\hat{\theta}$  à l'itération  $j+1$  est calculé en utilisant la formule  $\hat{\theta}_{j+1} = \hat{\theta}_0 (1 + \hat{\delta}_{j+1})$ .

De plus, la formulation  $[J_{\hat{\delta}}'' + \mu I]^{-1}$  est remplacé par  $[J_{\hat{\delta}}'' + \mu \text{diag}(J_{\hat{\delta}}'')]^{-1}$ , permettant une meilleure mise à l'échelle du problème.

#### F. Simulation du système linéaire en boucle fermée

Les simulations ont été effectuées en utilisant l'approche hors-ligne. Le système choisi est celui présenté à la section III utilisé directement dans une forme linéaire, comme présenté à l'équation (20). Un bruit de mesure  $w(t_k)$  est utilisé pour simuler un bruit capteur. Celui-ci est un bruit blanc gaussien, de moyenne nulle, et choisi pour avoir un rapport signal à bruit de 20 dB. Le signal de référence  $r(t)$  est un Signal Binaire Pseudo Aléatoire (SBPA), très utilisé dans le cadre de l'identification paramétrique.

Les paramètres du modèle sont  $K = 6$  and  $\tau = 110$ s. Le régulateur PI est réglé en utilisant la méthode directe, décrite dans [1], avec une constante de temps  $\tau_d = 80$ s. Cela donne un régulateur de la forme  $C(p) = 0,2292 + \frac{0,0021}{p}$ . La période d'échantillonnage est  $T_s = 1$ s.

Les fonctions de sensibilité appliquées à ce modèle sont définies par les équations suivantes :

$$\sigma_{\hat{K}} = \frac{S(p)}{\hat{P}(p)} \hat{u} \quad ; \quad \sigma_{\hat{\tau}} = -\frac{S(p)}{\hat{P}(p)} p \hat{y} \quad (32)$$

avec :

TABLE I: Résultats obtenus en simulation

	Valeurs réelles	VI	Algorithme proposé sans normalisation	Algorithme proposé avec normalisation
$\bar{K}$	6	5,91416	6,0044	6,0010
$\hat{\sigma}_{\bar{K}}^2$		$4,84 \cdot 10^{-4}$	$2,33 \cdot 10^{-4}$	$1,57 \cdot 10^{-4}$
$\bar{\tau}$	110	90,3202	110,0262	110,0067
$\hat{\sigma}_{\bar{\tau}}^2$		2,08	$8,69 \cdot 10^{-1}$	$4,82 \cdot 10^{-1}$
D		196,89	9,30	8,23

$$\begin{cases} C(p) = \frac{R(p)}{S(p)} = \frac{K_p p + K_i}{p} \\ \frac{S(p)}{\hat{P}(p)} = \frac{S}{\hat{A}S + \hat{B}R} = \frac{p}{\hat{\tau}p^2 + (1 + \hat{K}K_p)p + K_i \hat{K}} \end{cases} \quad (33)$$

On peut noter que les fonctions de sensibilité (32) sont physiquement réalisables. L'initialisation des paramètres  $\hat{K}$  et  $\hat{\tau}$  est effectuée d'abord par la méthode des MC puis celle de la VI avec FVE, présentées aux sections III-C et III-D. Le paramètre  $\lambda$  utilisé pour le filtrage est choisit pour être  $\lambda = 40 \frac{1}{\tau} = 0,3636$ . Le paramètre initial de Levenberg-Marquardt est  $\mu = 0,1$ .

Afin d'évaluer la robustesse de la méthode proposée, on utilise 100 simulations de Monte Carlo. On calcule ensuite les moyennes  $\bar{K}$  et  $\bar{\tau}$  ainsi que les variances  $\hat{\sigma}_{\bar{K}}^2$  et  $\hat{\sigma}_{\bar{\tau}}^2$  des paramètres identifiés. On calcule aussi la distance paramétrique comme suit :

$$D = \sqrt{\sum_i^N (K - \hat{K}_i)^2 + \sum_i^N (\tau - \hat{\tau}_i)^2} \quad (34)$$

Avec  $N$  le nombre de réalisations de Monte Carlo, et  $i$  représentant l'indice de la  $i$ ème simulation. Le tableau I présente les résultats trouvés pour les méthodes de la variable instrumentale (VI), ainsi que la méthode proposée dans cet article, avec et sans normalisation des fonctions de sensibilité.

On constate que les valeurs moyennes des paramètres estimés par la méthode proposée sont très proches des paramètres réels. Les variances d'estimation sont pour leur part très faibles. Les résultats obtenus avec normalisation des fonctions de sensibilité sont très proches des résultats obtenus sans normalisation. Cependant, la normalisation permet de diminuer le nombre moyen d'itérations, en passant de 4,6 itérations dans le cas non normalisé à 2,8 dans le cas normalisé.

#### G. Dispositif expérimental

Des tests expérimentaux ont été effectués sur le système présenté en section III. Les acquisitions des données ont été faites à l'aide d'un système dSpace DS1104. Celui-ci possède des convertisseurs analogique/numérique et numérique/analogique avec une résolution de 16 bits, et des niveau de tension entre -10 et +10V. Sur le système expérimental, le capteur de hauteur de liquide délivre une tension entre -10 et +10V et la pompe est commandée par

TABLE II: Résultats obtenus avec le dispositif expérimental

	VI	Algorithme proposé sans normalisation	Algorithme proposé avec normalisation
$\hat{K}$	2,60	6,90	6,93
$\hat{\sigma}_K^2$	$2,11 \cdot 10^{-3}$	$1,84 \cdot 10^{-4}$	$1,80 \cdot 10^{-4}$
$\hat{\tau}$	23,5	107,73	108,12
$\hat{\sigma}_\tau^2$	$1,43 \cdot 10^{-1}$	$5,08 \cdot 10^{-2}$	$4,70 \cdot 10^{-2}$

un signal entre -10 et +10 V. Afin d'obtenir un niveau de commande de la pompe entre 0 et 10V et une mesure de la hauteur de liquide directement en centimètres, on effectue une mise à l'échelle en ajoutant un gain ainsi qu'un offset. Un premier test d'identification nous permet de définir les paramètres du régulateur PI, ainsi que le paramètre  $\lambda$ . On obtient ainsi  $C(p) = 0,5515 + \frac{0,074}{p}$  et  $\lambda = 0,5372$ . De plus, comme la mesure du capteur est très bruitée, on ajoute un filtre de premier ordre de paramètre  $\zeta$  défini comme  $\zeta = 1,2\lambda = 0,6618$ . Le paramètre initial de Levenberg-Marquardt est choisit plus grand que pour la simulation, à cause d'une moins bonne initialisation par la VI. On a ainsi  $\mu = 1$ . Enfin, la période d'échantillonnage choisie est  $T_s = 1$ . La figure 2 résume les modifications apportées.

Comme dans la partie simulation, le signal de consigne est un SBPA. Le point de fonctionnement choisi est  $H_0 = 20$ cm. On effectue une première initialisation avec la VI présentée en section III-D.

Le tableau II présente les résultats obtenus avec les méthodes de la variable instrumentale (VI) ainsi qu'avec l'algorithme proposé, avec et sans normalisation des fonctions de sensibilité. Pour les deux paramètres, les valeurs identifiées  $\hat{K}$  et  $\hat{\tau}$  ainsi que les variances d'estimation  $\hat{\sigma}_K^2$  et  $\hat{\sigma}_\tau^2$  sont données. La figure 3 présente la comparaison des sorties réelle et simulées.

Le tableau II montre des valeurs très différents entre la VI et la méthode proposée. Malgré cela, en observant la figure 3, on peut dire que la sortie estimée par la méthode proposée est très proche de la sortie mesurée. Ainsi, même si l'initialisation est mauvaise, l'algorithme arrive à converger vers la solution.

Remarque : les valeurs identifiées avec la méthode proposée avec et sans normalisation sont très proches. Pour garder une figure lisible, seule l'identification avec norma-

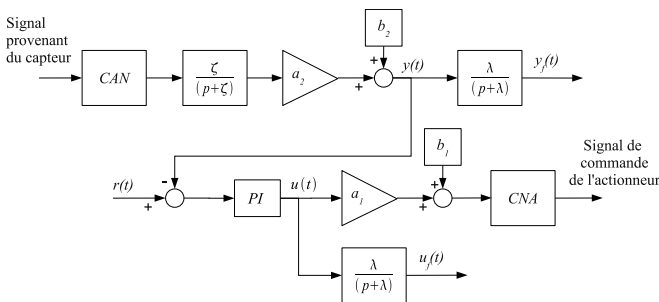


Fig. 2: Mise à l'échelle des signaux et asservissement du système

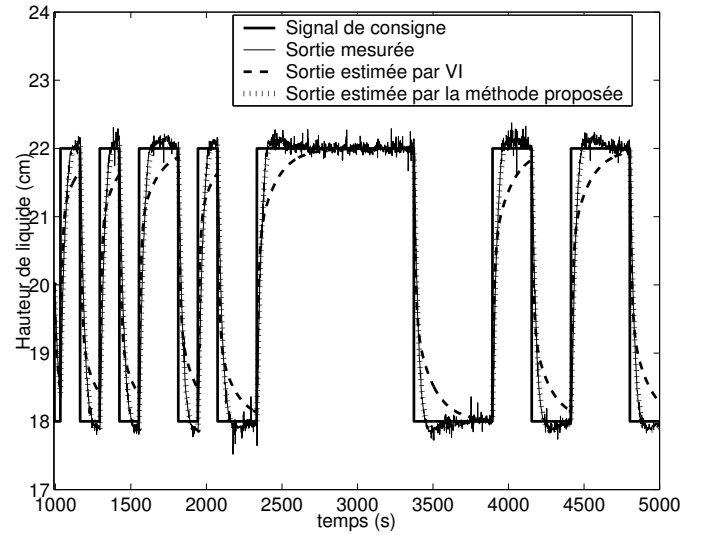


Fig. 3: Comparaison des sorties mesurée et estimées

lisation est montrée sur la figure 3.

#### IV. CONCLUSION

Dans cet article, une méthode d'identification à erreur de sortie de systèmes linéaires en boucle fermée à temps continu a été présentée. Cette méthode peut être considérée comme une version continue de l'algorithme discret F-CLOE de Landau.

Des simulations et une expérimentation sur un système de régulation de hauteur de liquide montrent de très bons résultats obtenus grâce à cette méthode.

#### RÉFÉRENCES

- [1] Carrillo F.J., Baysse A. et Habbadi A. Output error identification algorithms for continuous-time systems operating in closed-loop, 15th IFAC Symposium on System Identification, SYSID'09, St-Malo, France, 6-8 juillet 2009.
- [2] Garnier H., Gilson M., Bastogne T. et Zbali H. Contsid : un outil logiciel pour l'identification de modèles paramétriques à temps continu à partir de données expérimentales, Journées d'Identification et Modélisation Expérimentale, JIME'06, Poitiers, France, 16-17 novembre 2006.
- [3] Garnier H., Gilson M., Young P.C. et Huselstein E. An optimal IV technique for identifying continuous-time transfer function model of multiple input systems, *Control Engineering Practice*, vol. 15, n 4, 2007.
- [4] Garnier H., Wang L., (Eds.). *Identification of Continuous-time Models from Sampled Data*. Springer, Londres, 2008.
- [5] Gilson M., Garnier H. et Van den Hof P. Instrumental variable methods for continuous-time model identification in closed loop, Proceedings of the American Control Conference, Boston, U.S.A., 2004.
- [6] Landau I.D. et Karimi A. An output error recursive algorithm for unbiased identification in closed loop. *Automatica*, vol. 33, n 5, pp. 933-938, 1997.
- [7] Landau, I.D. Identification in closed loop : a powerful design tool (better designs models, simpler controllers). *Control Engineering Practice*, vol. 9, n 1 pp 51-65, 2001.
- [8] Richalet J. *Pratique de l'identification*. Hermès, Paris, 1991.
- [9] Tohme E., Ouvrard R., Abche A., Trigeassou J.C., Pointot T. et Mercere G. A methodology to enhance the convergence of output error identification algorithms. European Control Conference ECC'07, Kos, Greece, 2007.
- [10] Trigeassou J.C., Carrillo F.J., Pointot T. et Grospeud O. Convergence des algorithmes à erreur de sortie hors-ligne et récursifs. *APII Journal Européen des Systèmes Automatisés*. vol. 36, pp 397-415, 2002.
- [11] Amira, Laboratory Experiment DTS200 "Three-Tank-System", [http://www.amira.de/dts200\\_engl.html](http://www.amira.de/dts200_engl.html)