Contribution à l'étude et au prototypage d'un banc d'essais didactique flexible d'instrumentation virtuelle et d'asservissement par ordinateur

Jean MBIHI

Groupe de Recherche en Informatique Industrielle et Automation (GR2IA)
École Normale Supérieure d'Enseignement Technique (ENSET),
B.P. 1872, Douala, Cameroun.
E-mail : mbihidr@yahoo.fr

Résumé

Le *banc d'essais didactique flexible* décrit dans cet article est constitué d'un ordinateur, d'une carte d'acquisition de données commerciale, d'un logiciel d'instrumentation et d'asservissement numérique développé en technologie de *programmation visuelle*, et des prototypes de processus dynamiques à commander. Il permet aux étudiants de niveau Master en IIA (Informatique Industrielle et Automation) et EEA (Électronique, Électrotechnique et Automatique), d'expérimenter rapidement les *asservissements numériques* par *régulateur PID* et par *retour d'état avec ou sans observateur*. Les techniques de mise en œuvre du banc d'essais sont dévoilées, puis les échantillons de résultats de travaux pratiques obtenus sont présentés.

Mots clés : Banc d'essais didactique flexible, programmation visuelle, asservissements numériques, régulateur PID, retour d'état, observateur.

1. Introduction

Une boucle d'asservissement numérique représentée à la figure 1, désigne un automatisme programmable contrôlé par ordinateur, conçu de sorte que la grandeur réelle de sortie puisse suivre la consigne fixée par l'opérateur. Ce type d'automatisme programmable est constitué d'un ordinateur contenant un comparateur et un contrôleur numérique, d'une carte *d'acquisition de données* (CAD) jouant le rôle d'interfaces matérielles de *conversion analogique-numérique* (CAN) et de *conversion numérique-analogique* (CNA) des signaux, d'un *processus* à régler et d'un *capteur* qui délivre l'image en tension de la grandeur physique à régler.

Le but du présent article consiste à présenter un prototype de banc d'essais à but didactique, mis au point en vue de faciliter la tenue des travaux pratiques de l'unité d'enseignement d'instrumentation et d'asservissement par ordinateur, dispensée aux étudiants de Master en IIA (*Informatique industrielle et automation*) et EEA (*électronique, électrotechnique et automatique*) de l'ENSET de Douala. Les processus cibles dédiés à ce banc

d'essais sont des systèmes dynamiques de premier et second ordre. La section 2 de l'article porte sur les choix technologiques des constituants du banc d'asservissement numérique présenté. Dans la section 3, le logiciel *d'instrumentation virtuelle et d'asservissement par ordinateur* (IVAO) développé pour le contrôle du banc d'essais sera présenté. Puis, les échantillons de résultats d'asservissement numérique des processus dynamiques considérés seront présentés à la section 4.

2. Constituants du banc d'essais étudié

Le banc d'essais proposé est dédié à l'étude expérimentale rapide de l'asservissement numérique des processus dynamiques mono-variables (une entrée et une sortie) d'ordre un et deux. Le prototype présenté dans cet article a été construit à l'aide des constituants matériels commerciaux qui sont : un ordinateur PC, une carte d'acquisition de données PCI-P1202 de ICP DAS, un servomécanisme didactique de Feedback avec tachymètre embarqué, et des processus électriques contrôlés en tension. Par ailleurs, l'outil d'instrumentation et de commande numérique par ordinateur est un logiciel

Figure 1 : Boucle d'asservissement numérique par ordinateur

propriétaire d'appoint, développé en Visual Basic et offrant de multiples possibilités d'asservissement numérique qui seront décrites dans la prochaine section. En effet, dans nombre de laboratoires de formation ([1-3]), l'emploi d'un logiciel d'asservissement numérique propriétaire dont le code source est disponible pour des améliorations futures, est le plus souvent privilégié par rapport à l'utilisation des produits commerciaux lourds et coûteux tels que les logiciels Labview et Matlab/Simulink.

Contrairement à la plupart de bancs d'essais didactiques à processus cible unique [1-3], une grande variété de processus est supportée par le banc d'essais proposé, comme on le verra plus tard dans la section 4. Le banc d'essais didactique présenté étant dédié à la mise en œuvre d'une seule boucle d'asservissement, les ressources nécessaires de la carte PCI-P1202 sont les suivantes : a) canal d'entrée analogique CAN0; b) canal de sortie analogique CNA0; c) Timer embarqué (générateur de période d'échantillonnage); d) driver P1200.DLL fourni par le fabricant du produit pour la programmation en temps réel avec Visual Basic, Visual C++ et C#.

3. Logiciel d'asservissement numérique

3.1 Fonctionnalités du logiciel IVAO

Le logiciel d'asservissement numérique décrit dans cette section a été développé avec Visual Basic pour le système d'exploitation cible Windows 2000/XP. C'est une application Windows très légère qui occupe 250 Ko d'espace disque après l'installation. Les composants visuels d'entrées-sorties associés aux multiples fonctionnalités qu'il offre, sont facilement reconnaissables sur un panneau de contrôle virtuel facile à utiliser par tout apprenant ayant bien suivi le cours théorique de l'unité d'enseignement d'instrumentation et asservissement par ordinateur.

Une copie d'écran du panneau de contrôle virtuel obtenu après le démarrage du logiciel, est présentée à la figure 2. On y reconnaît les fonctionnalités suivantes :

- 1) Commande en boucle ouverte,
- 2) Régulation par PID avec paramètres de consigne u_r réglables par curseur.
- 3) Commande numérique par retour d'état d'état avec observateur complet et partiel.
- 4) Saisie du modèle du processus.

K_p → Gain proportionnel ; T_i et T_d → Constantes de temps intégrale et différentielle.
 $\{A, B, C, D\}$ → Paramètres du modèle d'état continu (ou discret).
 $[K_1 \ K_2]$ → Gains de retour d'état ; $[L_1 \ L_2]$ → Gains de l'observateur.
 K_v et K_{vx2} → Gains statiques des capteurs.
 T → Période d'échantillonnage (ms).
 N → Nombre d'échantillons

Figure 2 : Panneau de contrôle virtuel du logiciel d'asservissement numérique

- 5) Saisie des gains statiques de capteurs et des unités de grandeurs physiques à contrôler.
- 6) Affichage numérique et graphique des résultats obtenues pour un temps final $T_f = N T$ (voir figure 2 pour les notations N et T), avec échelles des temps et des valeurs réglables par curseurs.
- 7) Disponibilité des boutons de contrôle.

PCI-P1202; b) commande en boucle ouverte; c) commande numérique par PID ; d) calcul du modèle d'état discret; e) calcul de l'observateur d'état (complet ou partiel); f) représentation et sauvegarde des résultats de travaux pratiques.

3.2.1 Fonction nécessaires du driver de la carte PCI-P1202

3.2 Modules constitutifs du logiciel

Le travail de mise en œuvre logiciel a été morcelé en plusieurs modules dont les fonctions respectives sont : a) déclaration des fonctions nécessaires du driver de la carte

Le tableau 1 présente la syntaxe Visual Basic des principales fonctions du driver P1202.DLL, utilisées pour l'accès aux adresses d'entrées-sorties matérielles de la carte PCI-1202 [4]. Chaque ligne d'instructions est précédée d'un

commentaire permettant de spécifier la fonction de l'instruction. Par ailleurs, les fonctions du driver utilisées sont écrites uniquement pour des raisons de clarté.

<i>Fonctions utiles du driver de la carte PCI-1202 L/H</i>
<pre> ' Nombre de cartes PCI installé etat = P1202_DriverInit(TotalBoards) ' Initialisation de la carte PCI #0 etat = P1202_ActiveBoard(0) ' Configuration ADC0 Actif, plage +/- 5 V etat = P1202_SetChannelConfig(0, 0) ' Conversion N/A etat = P1202_Da(0, Val_ucNum) ' Conversion A/N avec résultat dans val_ys etat = P1202_AdPolling(val_ys): ' Temporisation de Nusec microsecondes etat = P1202_DelayUs(Nusec) </pre>

Tableau 1 : Fonctions du driver P1200.DLL

3.2.2 Commande en boucle ouverte

La commande numérique d'un processus cible en boucle ouverte ainsi que la lecture de la réponse obtenue sont effectuées à chaque période d'échantillonnage de durée T à l'aide du code Visual Basic du tableau 2.

<i>Commande en boucle ouverte du processus et lecture de la réponse en Visual basic</i>
<pre> ' Commande en boucle ouverte = consigne uc(n) = vyr ' vyr = Image en tension de la consigne val_uc = vyr ' Image en tension de la consigne Val_ucNum = Val(207.637 * val_uc + 2048) ' Valeur entière du mot de code etat = P1202_Da(0, Val_ucNum) ' Conversion N/A au canal0 etat = P1202_AdPolling(val_ys): ' Conversion A/N vys(n) = val_ys ' Image en tension de la grandeur de sortie ys(n) = (1 / Kv) * vys(n) ' Kv = gain en tension du capteur </pre>

Tableau 2 : Section de code Visual Basic pour la commande en boucle ouverte

3.2.3 Commande numérique par PID

La mise en œuvre du contrôleur numérique PID se déroule en trois étapes suivantes : a) lecture dans le cadre des consignes des données requises (valeurs de consigne, unités de la grandeur à régler, gain des capteurs, période d'échantillonnage, nombre d'échantillons); b) lecture dans le cadre de contrôleurs du gain proportionnel K_p et des constantes de temps T_i (intégrale) et T_d (différentielle); c) calcul de l'erreur en tension $e(n) = u_{yr} - u_y(n)$; d) calcul et application de la loi de commande numérique obtenue par la méthode de discrétisation

suivante $p \rightarrow \frac{z-1}{Tz}$ [5] :

$$\begin{cases}
 P \Rightarrow u_c(n) = K_p e(n) \\
 PD \Rightarrow u_c(n) = K_p (1 + Td/T) e(n) - K_p (Td/T) e(n-1) \\
 PI \Rightarrow u_c(n) = u_c(n-1) + K_p (1 + T/T_i) e(n) - K_p e(n-1) \\
 PID \Rightarrow u_c(n) = u_c(n-1) + a_0 e(n) + a_1 e(n-1) + a_2 e(n-2)
 \end{cases} \quad (1)$$

$$\text{avec } \begin{cases}
 a_2 = K_p \left(1 + \frac{Td}{T} + \frac{T}{T_i} \right), \\
 a_1 = -K_p \left(1 + 2 \frac{Td}{T} \right), \\
 a_0 = K_p \left(\frac{Td}{T} \right)
 \end{cases} \quad (2)$$

3.2.4 Calcul du modèle d'état discret

Le module de commande par rétroaction d'état est applicable aux processus du second ordre dont la grandeur de sortie correspond à l'état x_r . Si le modèle d'état d'entrée est de type continu, les paramètres $\{A_c, B_c\}$ du modèle discret équivalent sont calculés à partir des paramètres $\{A, B\}$ du modèle continu, par approximation de Taylor d'ordre N , définie par [6]:

$$A = e^{A_c T} \approx I + \psi T A_c, \text{ avec } \psi = \sum_{k=0}^N \frac{A_c^k T^k}{(k+1)!} \quad (3)$$

$$B = T \psi B_c$$

Dans le cas contraire le modèle d'état d'entrée est considéré comme discret. La relation (3) a


```

Private Sub CmdSauvegarde_Click()
Dim NomFichier As String
NomFichier = InputBox("Nom du Fichier de Données MATLAB à Sauver (*.M)", "Sauvegarde de données  
au format MATLAB", "IvaoData.M")
Open NomFichier For Output As #1
For i = 0 To (Nech - 2)
If i = 0 Then 'Header of Matlab file
Print #1, "' IV/AOPos.M - Tab=Tableau de données'", Chr$(13) + Chr$(10)
Print #1, "' Tab=[n, n.T(s), uc(Volts), x1=Ys(Volts) x2s]", Chr$(13) + Chr$(10)
ElseIf i = 1 Then
Print #1, "Tab=[", i - 1, Str(temps(i - 1)), Str(uc(i - 1)), Str(ys(i - 1)), Str(x2s(i - 1)),  
Chr$(13) + Chr$(10)
Else
Print #1, " ", i - 1, Str(temps(i - 1)), Str(uc(i - 1)), Str(ys(i - 1)), Str(x2s(i - 1)), Chr$(13) + Chr$(10)
End If
Next i
Print #1, " ", Nech - 1, Str(temps(Nech - 1)), Str(uc(Nech - 1)), Str(ys(Nech - 1)), Str(x2s(i - 1)), " ]",  
Chr$(13) + Chr$(10)
Close #1
MessageTexte.Visible = False
MsgBox ("Fin de Sauvegarde: Cliquez Ok ")
End Sub

```

Tableau 3 : Code source Visual Basic du module de sauvegarde des résultats

(contrôle FlexGrid de Visual Basic), alors que les résultats d'essais sont représentés graphiquement sur un *grapheur* (contrôle standard *Image* de Visual Basic). Ces deux composants visuels (tableur et grapheur) sont des modules logiciels récupérés dans le code source d'un projet Visual Basic développé dans le cadre d'une recherche antérieure [7].

3.2.7 Sauvegarde des résultats

Le bouton de commande *Sauve* permet d'enregistrer les résultats obtenus dans un fichier Matlab. La structure du tableau de données enregistré est la même que celle du tableur. Le code source de la procédure événementielle correspondante est un constituant développé dans [8], puis réutilisé

après quelques modifications mineures. Le tableau 3 décrit la structure du code source de ce constituant logiciel. L'implémentation des modules logiciels décrits dans cette section, a été facilitée en suivant les bases de programmation système détaillées dans [8] et [9].

4. Asservissements numériques

4.1 Asservissements électriques

Le processus considéré est un circuit électrique du 2^e ordre représenté à la figure 4. Sachant que $R_1 = 2,2 \text{ k}\Omega$, $R_2 = 10 \text{ k}\Omega$, $L = 51,5 \text{ mH}$ et $C = 1 \text{ uF}$, alors la fonction de transfert $G_c(p) = V_c(p)/U_c(p)$ qui en résulte en boucle ouverte s'écrit :

$$G(p) = \frac{\frac{R_2}{R_1 + R_2}}{\frac{R_2 LC}{R_1 + R_2} p^2 + \frac{L + R_1 R_2 C}{R_1 + R_2} p + 1} \quad (4)$$

L'application numérique donne :

$$G_c(p) = \frac{0.8197}{4,221 (10)^{-8} p^2 + 0,001808 (10)^{-8} p + 1} \quad (5)$$

Figure 4 : Circuit électrique du 2^e ordre

Figure 5: Réponse indicielle

Par ailleurs, avec le choix du vecteur d'état $x = [x_1 = u_c \ x_2 = i_L]^T$ et de la grandeur de sortie $y = u$, on obtient le modèle d'état :

$$\begin{cases} \frac{dx(t)}{dt} = \begin{bmatrix} -100 & 100 \times 10^4 \\ -19.417 & -427.18 \times 10^4 \end{bmatrix} x(t) \\ \quad + \begin{bmatrix} 0 \\ 19.417 \end{bmatrix} u_c(t) \\ y(t) = [1 \ 0] x(t) \end{cases} \quad (6)$$

4.1.1 Asservissement de tension par PID

Le régulateur PID à utiliser est calculé préalablement avec Matlab. Le modèle discret équivalent est calculé par le logiciel en utilisant les relations (1) et (2).

La figure 5 présente les résultats de la réponse indicielle obtenus par simulation avec Matlab (figure 5(a)) et expérimentalement (figure 5(b)) en considérant les paramètres $K_p = 2$, $T_i = 0,0145$ s et $T_d = 0,000035$ s. On constate que pour le même signal de

consigne, le logiciel d'asservissement reproduit bien les réponses prédites du circuit étudié.

4.1.2 Asservissement numérique de tension et du courant par rétroaction d'état avec observateur

En discrétisant le modèle d'état (6) du processus à l'aide de la primitive *c2d* de Matlab avec une période d'échantillonnage $T = 0.1$ ms, on obtient le modèle discret équivalent suivant :

$$x(k+1) = \begin{bmatrix} 0,95577 & 22,324 \\ -0,00043348 & 0,0043314 \end{bmatrix} x(k) + \begin{bmatrix} 0,034428 \\ 0,00043314 \end{bmatrix} u_c(k) \quad (7)$$

Puisque la mesure du courant $x_2 = i_L$ n'est pas disponible, il est nécessaire de le générer à l'aide d'un observateur d'état d'ordre partiel admettant pour grandeurs d'entrée u_c (tension de commande) et y (tension de sortie), et pour grandeur de sortie l'estimé de la grandeur inconnue i_L .

La simulation préalable par Matlab du système asservi par rétroaction d'état de gain $K = [1,834 \ -931,784]$ avec observateur d'état d'ordre partiel de gain $L_o = -0,03$ a donné des résultats théoriques et pratiques présentés sur la figure 6. On constate sur la figure 6(b) que le logiciel *IVAO* reproduit bien les signaux prédits par simulation (figure 6(b)). Ainsi, une analyse

Figure 6 : Résultats de rétroaction d'état avec observateur d'ordre réduit

expérimentale précise d'un circuit électrique quelconque est possible.

4.2 Asservissement de vitesse par PID

L'asservissement de vitesse considéré à la figure 7 est constitué d'un amplificateur de puissance, d'un servomoteur Feedback équipé d'un tachymètre de gain statique α , et d'un frein électromagnétique. Le schéma d'implantation des constituants correspond à la figure 7(a) et le schéma bloc associé est représenté à la figure 7(b). La fonction de transfert $M(p)=Y(p)/U_c(p)$ de ce système en boucle ouverte s'écrit :

$$M(p) = \frac{0,375}{1 + 0,29 p} \quad (8)$$

Un régulateur numérique de vitesse par PID dont les paramètres $K_p = 2,5$; $T_i = 0,3$ et $T_d = 0,2$ ms a été préalablement conçu à l'aide de Matlab, puis testé sur le banc d'essais. La figure 8 présente les résultats d'essais (figure 8(a)) et de simulation (figure 8(b)) obtenus. Remarquons que l'unité de la grandeur manipulée est en tr/mn pour un gain en tension $\alpha = 0,05$ ($V.mn/tr$) du capteur.

a) Schéma d'implantation du banc d'asservissement numérique de vitesse

b) Schéma bloc de l'asservissement numérique de vitesse

Figure 7: Banc d'asservissement numérique de vitesse

Figure 8: Résultats d'asservissement numérique de vitesse par PID :
(a) Expérimentation (b) Simulation

4.3 Asservissement numérique de position par rétroaction d'état avec observateur

Bien que la vitesse de sortie y de l'asservissement soit mesurable à l'aide d'un tachymètre, il n'existe pas de capteur de position. C'est ce qui justifie ici la nécessité de rétroaction d'état de l'observateur d'ordre partiel dont le gain d'observation correspond à L_o .

En considérant comme vecteur d'état noté $x = (\nu \ \theta)^T$, le processus en boucle ouverte est décrit par le modèle d'état continu suivant :

$$\begin{cases} \frac{dx_1(t)}{dt} = \begin{bmatrix} -\frac{1}{\tau} & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} \frac{K_m}{\tau} \\ 0 \end{bmatrix} u_c(t) \\ y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \end{cases} \quad (9)$$

avec $K_m = 0,375$, $\tau = 0,29$ s

La simulation préalable par Matlab du système asservi avec un gain de d'état noté $K = [2,2 \ 1,6235]$ en présence d'un observateur d'état d'ordre partiel de gain $L_o = -0,95$, a donné des résultats théoriques

Figure 9 : Résultats d'asservissement numérique de position par retour d'état et observateur

satisfaisants qui ont permis de valider les résultats expérimentaux. La figure 9 présente les résultats de simulation par Matlab (figure 9(a)) et ceux obtenus expérimentalement (figure 9(b)) à l'aide du logiciel . Ces résultats montrent une fois de plus que le banc d'essais présenté reproduit assez bien le comportement théorique d'un système asservi de position avec retour d'état de l'observateur.

5. Conclusion

La qualité des résultats théoriques et pratiques présentés dans cet article montre que le banc d'essais présenté, est un outil

didactique d'appoint flexible, permettant aux étudiants de Master en IIA et EEA de l'ENSET de Douala, d'expérimenter rapidement et rigoureusement les lois de commande numérique étudiées dans l'unité d'enseignement *d'instrumentation virtuelle et asservissement par ordinateur*. La flexibilité du banc d'essais est surtout due au fait que seuls quelques réglages simples en *temps d'exécution* suffisent pour passer de l'asservissement numérique d'un processus donné à celui d'un autre processus cible différent.

Les limitations de la version actuelle du logiciel IVAO proposé sont : a) manque de

simulateur virtuel qui permettrait de concevoir le régulateur numérique d'un processus sans utiliser Matlab; b) absence de base de données des processus dynamiques cibles à asservir ; c) non prise en compte du comportement dynamique du capteur; d) limitation aux processus de second ordre. Ces limitations suscitent de nouvelles perspectives de recherches pédagogiques futures.

Références bibliographiques

[1] R. M. DE SANTIS, Cours 3.520 : Asservissements de position angulaire avec un lien de transmission élastique : Travaux pratiques", Ecole Polytechnique de Montréal, 1991.

[2] E. DUQUENOY, "Plateforme pour l'étude et la programmation des asservissements échantillonnés", Colloque sur l'enseignement des technologies et des sciences de l'information et des systèmes (CETISIS), 13-14 Novembre 2003, Toulouse.

[3] V. SILVIA, V. CARVALHO, R. M. VASCONCELOS, F. SAORES, "Remote PID control of a DC motor", International Journal of Online Engineering", Vol. 3, No. 3, 2007.

[4] ICP DAS, "PCI- P1202/1602/1800/1802 Hardware User's manual – Version 3.4", 2003, ICP DAS.

[5] A. DELEXURIEUX et M. RAMI, "Cours d'électronique numérique et échantillonnage : BTS, IUT, écoles d'ingénieurs", Eyrolles, 1991.

[6] G. F. FRANKLIN, J. D. POWELL et M. L. WORKMAN, "Digital control of dynamic systems – Second Edition", Addison-Wesley, 1990.

[7] J. MBIHI, N. KELAKO, et M. L. NGO NOUCK, "Didacticiel d'instrumentation virtuelle à coût réduit", Journal sur l'enseignement des sciences et des techniques de l'information et des systèmes, *J3EA* - Vol. 7, No. 2 (2008), © EDP Sciences.

[8] J. MBIHI, "Informatique et Automatisation : Automatismes Programmables Contrôlés par Ordinateur". Editions Publibook Universités, Paris, 2006.

[9] MICROSOFT, "Visual Basic 5.0 for 32 bits Windows development", 1997.

Sites Internet utiles

[Int-1] www.cetsis.org/ "Site du colloque sur l'enseignement des technologies et des sciences de l'information et des systèmes".

[Int-2] www.icpdas.com/catalog/catalog.htm, "Carte d'acquisition de données pour bus PCI".

[Int-3] www.icpdas.com/news/new_events/2006/20061121.htm, "C# and VB.Net sample programs for PCI DAQ cards are released".

Remerciements

L'auteur tient à remercier sincèrement les deux rapporteurs anonymes de J3EA qui ont examiné en profondeur le premier manuscrit de cet article. Leurs remarques et suggestions très constructives, ont beaucoup contribué à l'amélioration aussi bien de la forme que du fond du contenu de la version finale.

Biographie de l'auteur

J. MBIHI a obtenu le Ph. D. (1999) et le MSc. A. (1992) en génie électrique à l'École Polytechnique de Montréal (Québec, Canada). Il est actuellement Maître de conférence, responsable du *Groupe de Recherche en Informatique Industrielle et Automatisation* (GR2IA), et chef du département d'Industrie Textile et de l'Habillement à l'ENSET de Douala (Cameroun). Il est membre du comité scientifique de l'unité de formation doctorale en Physique et Sciences de l'Ingénieur de l'Université de Douala, et membre du Comité Consultatif des Institutions Universitaires du Cameroun. Il est auteur de l'ouvrage "*Informatique et Automatisation : Automatismes programmables contrôlés par ordinateur*", publié aux éditions Publibook de Paris en 2006. Ses recherches actuelles portent sur les nouvelles technologies d'instrumentation virtuelle et de commande par ordinateur des processus à distance via Internet.