

REGULATION EN TEMPS REEL D'UN SYSTEME DE LEVITATION MAGNETIQUE AVEC xPC-TARGET

L. Benbaouche¹ & F. Khaber²

¹IUT Ecole Supérieure des Travaux Publics, 57, Bd Saint-Germain, 75005, France, benbaouche@estp.fr

²UFAS Département d'Electrotechnique, Faculté des Sciences de l'Ingénieur, 19000 Sétif Algérie, jfkahber@yahoo.fr

RESUME

Cet article traite la régulation en temps réel de la lévitation magnétique grâce au Toolbox xPC-Target de Matlab qui est un environnement de prototypage hôte-cible. Un observateur nous permet d'estimer les variables d'état non mesurables que sont l'intensité de courant et la vitesse de déplacement. A fin d'éliminer l'erreur de position engendrée, un algorithme PI est appliqué. Les gains du correcteur à retour d'état et de l'observateur sont déterminés par la méthode des moindres carrés. Le contrôleur est simulé sous l'environnement Matlab/Simulink et est implémenté au moyen de xPC-Target qui est une solution intéressante pour la commande de systèmes en temps réel. Les résultats expérimentaux obtenus montrent l'efficacité de l'approche adoptée et de l'algorithme implémenté.

ABSTRACT

In this paper, a real-time controller for magnetic levitation system is presented. It consists of a state observer and a PI controller for removing position error. The gains of the state feedback and of the observer are determined by the least squares method. The controller is simulated on Matlab/Simulink environment and implemented by means of xPC-Target which is a PC solution for targeting real time systems. The experimental results show the efficiency of the adopted approach and of the implemented algorithm.

MOTS-CLES

Régulation temps réel, Correcteur par retour d'état, Observateur d'état, xPC-Target, Carte PCI-DAS-12001, Lévitation magnétique, Matlab/Simulink.

1. INTRODUCTION

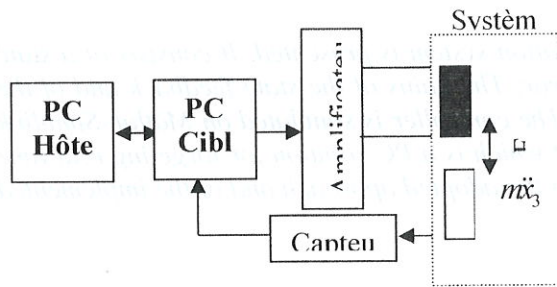
Le système de lévitation magnétique fait appel à des forces électromagnétiques asservies pour établir des liaisons sans contact. Un pendule pesant peut être maintenu à une certaine distance en commandant la tension d'alimentation de la bobine de l'électro-aimant. Les régulateurs analogiques et numériques ont été proposés dans le manuel d'AMIRA [1]. Dans cet article nous proposons la régulation en temps réel du système de lévitation magnétique, qui est flexible et efficace. La première étape est consacrée à l'étude et conception des correcteurs grâce à la simulation, réalisée par le logiciel Simulink, sur l'ordinateur « hôte ». La seconde étape est l'implémentation des algorithmes de régulation et l'exécution de l'application à partir de l'ordinateur « cible ». La simulation et la régulation sont donc réalisées dans l'espace d'état. Ce dernier est un modèle de spécification

du système. Il permet de représenter des calculs importants à plusieurs variables et présente aussi un avantage pour le traitement numérique. Quant au système de lévitation magnétique, un élément intégrateur est ajouté pour éliminer l'erreur de position en régime permanent. Le réglage d'état nécessite quant à lui la connaissance des grandeurs d'état que sont le courant électrique, la vitesse de déplacement de la tige, la hauteur et la grandeur de sortie de l'intégrateur. Les variables d'état, qui sont accessibles à la mesure sont la hauteur, le courant électrique la vitesse de déplacement et la grandeur de sortie de l'intégrateur, sont reconstitués grâce à un observateur. Un correcteur à retour d'état permet d'obtenir les performances dynamiques souhaitées. Le correcteur à action intégrale annule l'erreur du régime permanent. Afin d'éviter le retard, qui est dû à l'action intégrale du correcteur, un algorithme PI est utilisé. Grâce au logiciel

Simulink les vecteurs du gain de l'observateur et du correcteur par retour d'état sont déterminés en se basant sur la méthode des moindres carrés.

2. PRINCIPE DE LA REGULATION

La configuration matérielle de la régulation proposée est constituée essentiellement d'un ordinateur hôte, d'un ordinateur cible et d'un système de lévitation magnétique. L'ordinateur « hôte » permet d'une part, grâce à la simulation, l'étude du comportement physique du système de lévitation magnétique et la conception des correcteurs en utilisant les logiciels Matlab et Simulink et de générer le code exécutable avec Real-Time Workshop et de l'envoyer vers le PC cible. L'ordinateur « cible » exécute l'application sous le noyau temps réel xPC-Target. La communication entre le PC hôte et le PC cible se fait à travers la RS232.



Ordinateur hôte : PC exécutant Windows, Matlab, Simulink, Real-Time

xPC-Target.

Ordinateur cible : PC tournant sous le noyau xPC-Target.

Lévitation magnétique : Organe de commande, électro-aimant, tige et capteur.

Figure 1. Schéma synoptique de la régulation.

3. MODELISATION DU SYSTEME

Le processus est un pendule pesant pouvant se déplacer verticalement grâce à l'action des forces électromagnétiques générées par un électro-aimant parcourue par un courant électrique. L'ensemble est constitué d'un aimant et de deux bobines.

Le modèle mathématique continu qui décrit le système [1] est donc le suivant :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -86.95 & 0 & 0 \\ 652.18 & -43.48 & 0 \\ 0 & 43.48 & 43.48 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 8.69 \\ 0 \\ 0 \end{bmatrix} u$$

$$y = [0 \ 0 \ 4]x$$

où x_1 représente le courant, x_2 représente la vitesse de déplacement de la tige et x_3 désigne la hauteur.

La régulation étant numérique, il est nécessaire alors de discrétiser le modèle mathématique ($T_e = 0.001s$) pour déterminer les correcteurs. Le modèle mathématique discret (F, h, c^T) ainsi obtenu est le suivant:

$$x(k+1) = \begin{bmatrix} 0.92 & 0 & 0 \\ 0.61 & 0.96 & 0 \\ 0.01 & 0.04 & 1.04 \end{bmatrix} x(k) + \begin{bmatrix} 0.008 \\ 0.003 \\ 0 \end{bmatrix} u(k)$$

$$y(k) = [0 \ 0 \ 1]x(k)$$

L'action de l'intégrateur est décrite par l'équation aux différences suivante :

$$q(k+1) = q(k) + r(k) - y(k)$$

La variable décrivant l'action intégrale est insérée dans le vecteur d'état du processus. Le modèle mathématique global devient :

$$z(k+1) = \begin{bmatrix} x(k+1) \\ q(k+1) \end{bmatrix} = \begin{bmatrix} F & 0 \\ -c^T & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ q(k) \end{bmatrix} + \begin{bmatrix} h \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(k)$$

$$y(k) = [h \ 0] \begin{bmatrix} x(k) \\ q(k) \end{bmatrix}$$

4. SYNTHESE DE LA COMMANDE

L'utilisation de l'approche d'état pour la commande suppose la disponibilité de toutes les composantes du vecteur d'état. Dans le cas où une partie ou la totalité de ces composantes ne sont pas accessibles à la mesure, il est nécessaire de les reconstruire en utilisant un observateur d'état. La structure adoptée, représentée sur la figure 2, comporte essentiellement un contrôleur par retour d'état, un observateur et un correcteur intégrateur.

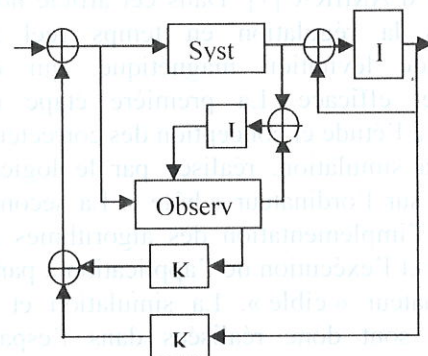


Figure 2. Synthèse de la loi de commande.

Après vérification de la commandabilité et de l'observabilité du système, les vecteurs de gain du contrôleur par retour d'état et celui de l'observateur sont déterminés grâce au principe de séparation tout en minimisant le critère d'optimalité quadratique [2] suivant :

$$J_N = x^T(N)Qx(N) + \sum_{k=0}^{N-1} x^T(k)Qx(k) + ru^2(k)$$

où N est un nombre fini d'échantillons, Q est une matrice de pondération de l'état intermédiaire et r est un coefficient de pondération du signal d'entrée de commande.

3.1. Observateur d'état

La reconstruction des variables d'état $x_1(k)$, $x_2(k)$ et $x_3(k)$ nécessite un observateur. L'équation d'état de l'observateur est :

$$\hat{x}(k+1) = F\hat{x}(k) + hu(k) + L[y(k) - \hat{y}(k)]$$

L'erreur de l'observateur est :

$$\tilde{x}(k+1) = x(k+1) - \hat{x}(k+1) = [F - Lc^T] \tilde{x}(k)$$

La vitesse avec laquelle les variables d'état de l'observateur $\hat{x}(k)$ tend vers $x(k)$ est déterminée par les valeurs propres de la matrice $[F - Lc^T]$. L'observateur doit répondre plus rapidement que le système corrigé, ceci nous impose de proposer des valeurs propres inférieures à celles du système corrigé.

3.2. Correcteur par retour d'état

L'influence de la grandeur d'entrée $r(k)$ est considérée comme de perturbation. Pour la conception du correcteur, elle ne sera pas prise en compte. Etant donné la matrice du processus avec intégrateur F_z et le vecteur d'entrée h_z , le correcteur par retour d'état $K_z^T = [K^T \ K_I]$ peut donc être déterminé à l'aide de la méthode des moindres carrés. La loi de commande est alors donnée par :

$$u(k) = -K^T \hat{x}(k) - K_I q(k) = -u_{com} - u_I$$

Afin d'éviter le retard d'une période d'échantillonnage qui est dû à l'action intégrale lorsque la grandeur d'entrée varie, l'algorithme

dont est menu l'intégrateur est remplacé par un algorithme type PI de la forme suivante :

$$u_I(k) = K_I [\gamma q(k+1) + (1-\gamma)q(k)] \text{ , où } \gamma \leq 1$$

3.3. Algorithme

L'algorithme appliqué pour déterminer les vecteurs de gain du retour d'état et d'observation est le suivant :

a) Initialisation de Q et de r ($Q = P_N$).

b) Détermination des gains des vecteurs de retour d'état et de l'observateur.

$$K^T(N-j) = [h^T P(N-j+1)h + r]^{-1} h^T P(N-j+1)F$$

c) Détermination des matrices $P(N-j+1)$

$$P(N-j) = Q + F^T P(N-j+1)F$$

$$[I - h[h^T P(N-j+1)h + r]^{-1} h^T P(N-j+1)]F$$

d) Détermination des commandes

$$u(N-j) = -K^T(N-j)x(N-j)$$

5. SIMULATION ET IMPLEMENTATION

La simulation a été réalisée dans l'environnement Matlab/Simulink.

L'implémentation a été réalisée par les logiciels Real-Time WorkShop et xPC-Target conformément à la configuration donnée sur la figure suivante :

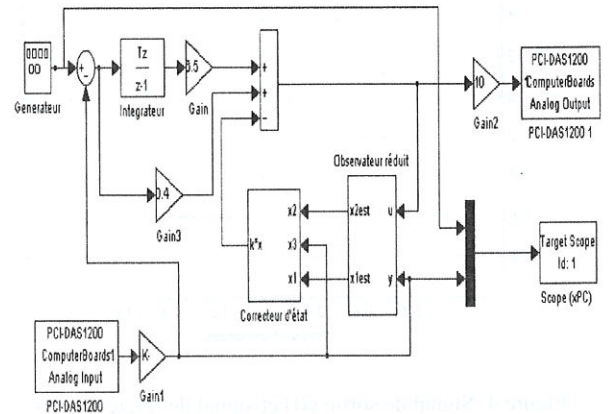


Figure 3. Schéma de principe de l'implémentation.

5.1. Mise en œuvre de l'algorithme

En choisissant la matrice de pondération $Q = \text{diag}\{100, 100, 100\}$ et le coefficient de pondération du signal d'entrée $r=1$, nous obtenons pour le régime permanent, correspondant à la solution de l'équation de

Riccati, le vecteur de gain du retour d'état suivant : $K_{ob} = [0.08068 \ 0.533 \ 0.178]$

Les valeurs propres imposées à l'observateur sont $\{0.94, 0.83 \pm j0.13\}$:

Les valeurs propres du système corrigé sont : $\{0.97, 0.92 \pm j0.27\}$.

Nous remarquons que ces valeurs propres sont bien supérieures à celles de l'observateur et se trouvent à l'intérieur du cercle unité, donc le système corrigé est stable et bien amortie.

5.2. Résultats de simulation

La grandeur de sortie qui suit le signal de référence avec une erreur de position nulle montre que le système en boucle fermée est bien stable (figure 4). Les variables d'état estimées du système (courant et vitesse) reproduisent fidèlement les variables réelles du système comme montré sur les figures 5 et 6. La loi de commande synthétisée est représentée sur la figure 7. Le choix judicieux des matrices de pondération justifie l'amplitude assez faible de cette commande.

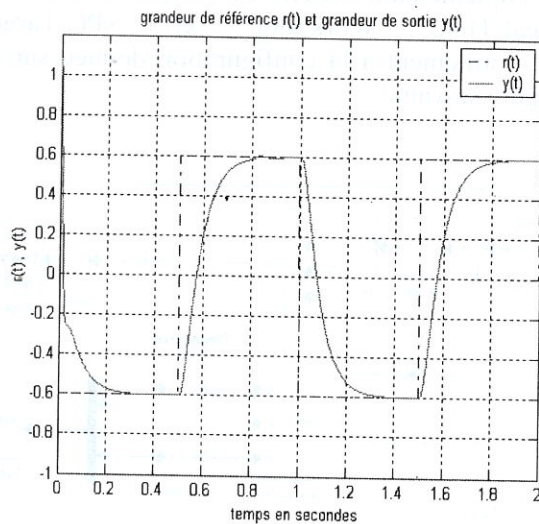


Figure 4. Signal de sortie $y(t)$ et signal de référence $r(t)$.

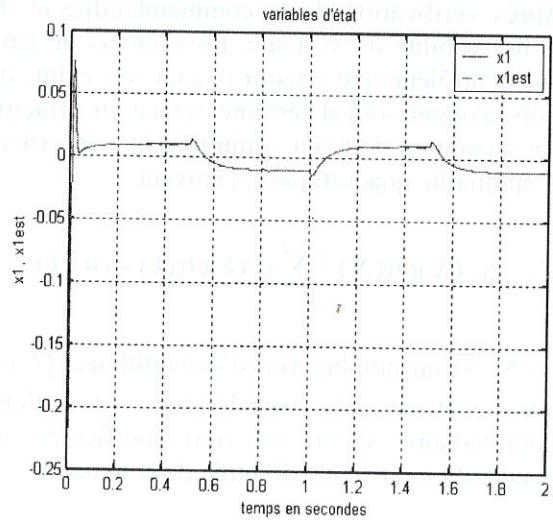


Figure 5. Etat $x1(t)$ et son estimée $x1est(t)$.

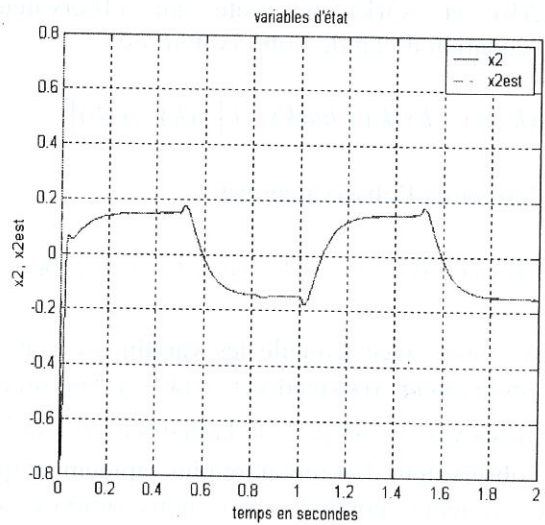


Figure 6. Etat $x2(t)$ et son estimée $x2est(t)$.

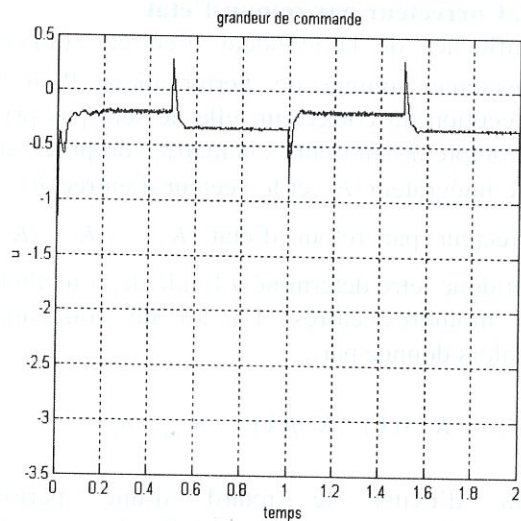


Figure 7. Signal de commande $u(t)$.

5.3. Implémentation

Après exécution de l'application générée par xPC-Target sur l'ordinateur « cible », la commande du système se fait en temps réel indépendamment de l'ordinateur « hôte ». Les résultats obtenus (réels) montrent que la sortie du système suit correctement le signal de référence (figure 8). Une comparaison de la sortie réelle avec celle obtenue par simulation (figure 4) montre l'efficacité de l'approche adoptée et de l'algorithme implémenté.

6. CONCLUSION

Les résultats obtenus permettent de valider les algorithmes de régulation implémentés. La programmation dynamique, conçu par Bellman [2], a permis de trouver la commande optimale qui a stabilisé le système. Les gains du correcteur par retour d'état et de l'observateur se déterminent automatiquement grâce à l'algorithme implémenté. L'utilisation du logiciel xPC-Target confère une interactivité et une flexibilité dans la commande temps réel.

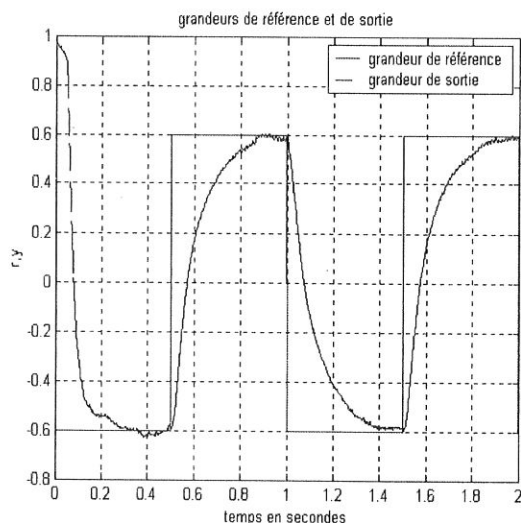


Figure 8. Grandeurs d'entrée et de sortie réelles.

7. REFERENCES

- [1] AMIRA, *Laboratory Setup Magnetic Suspension*, Copyright amira GmbH, 1998, Duisburg.
- [2] R. Bellman, Special issue, *IEEE Trans. Aut. Control*, vol. AC-16, 26 Oct. 1981.
- [3] D.G. Luenberger, An introduction to observers, *IEEE Trans. Aut. Control*, vol. 16, pp.596-602, 1971.
- [4] H. Tolle, *Mehrgrößen-Regelkreissynthese*

Band II : Entwurf im Zustandsraum, Oldenbourg Verlag, München, 1985.

[5] *xPC Target For Use with RealTime Workshop*, 2001, the Mathworks

[6] *Simulink, Model-Based and System-Based Design*, Version 4, the Mathworks

[7] *Real-Time Workshop*, Version 3, the Mathworks

[8] W. Lin, Global asymptotic stabilization of general nonlinear system with stable free dynamics via passivity and bounded feedback, *Automatica*, Vol. 32 (1996), 915-924.

[9] M. Krstic, I. Kanellakopoulos, and P.V. Kokotovic, *Nonlinear and Adaptive Control Design*, John Wiley, 1995.